Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

☙ CERTIFICATE ❧

Certify that Mr./Miss Soumil Vivek Salvi

of IT Department, Semester V with

Roll No. 104 has completed a course of the necessary

experiments in the subject Advanced Dev Ops under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

[illegible]

Teacher In- Charge

Head of the Department

Date 20/10/23

Principal

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

CONTENTS

ASSIGNMENT-1

AIM : Study and create AWS EC2 instance

LO MAPPED : LO1

THEORY :

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

Features of Amazon EC2

Amazon EC2 provides the following high-level features:

- Instances :

Virtual servers.

- Amazon Machine Images (AMIs):

Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).

- Instance types :

Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.

- Key pairs :

Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.

- Instance store volumes :

Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.

- Amazon EBS volumes :

Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).

- Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones :

Multiple physical locations for your resources, such as instances and Amazon EBS volumes.

- Security groups :

A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances, and the destination IP ranges to which your instances can connect.

- Elastic IP addresses :

Static IPv4 addresses for dynamic cloud computing.

- Tags :

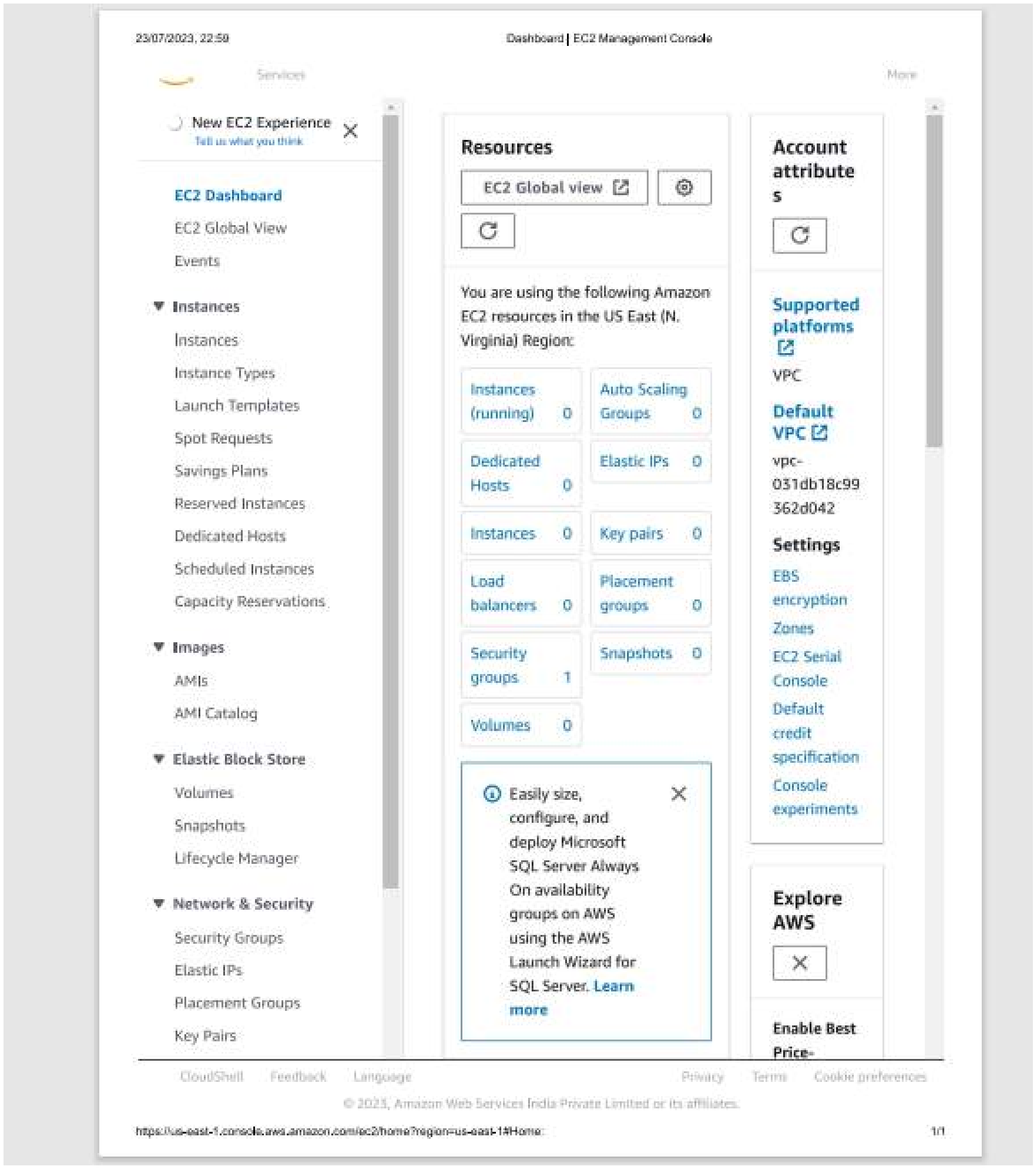
Metadata that you can create and assign to your Amazon EC2 resources.

- Virtual private clouds (VPCs) :

Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your own network.

STEPS-

1. LOGIN TO AWS ACCOUNT,
THEN SEARCH EC2.


23/07/2023, 22:59 Dashboard | EC2 Management Console

Services More

New EC2 Experience
Tell us what you think

EC2 Dashboard
EC2 Global View
Events
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
▼ Images
AMIs
AMI Catalog
▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
▼ Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs

Resources

EC2 Global view

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running) 0	Auto Scaling Groups 0
Dedicated Hosts 0	Elastic IPs 0
Instances 0	Key pairs 0
Load balancers 0	Placement groups 0
Security groups 1	Snapshots 0
Volumes 0	

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Account attributes

Supported platforms
VPC
Default VPC
vpc-031db18c99362d042
Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Explore AWS

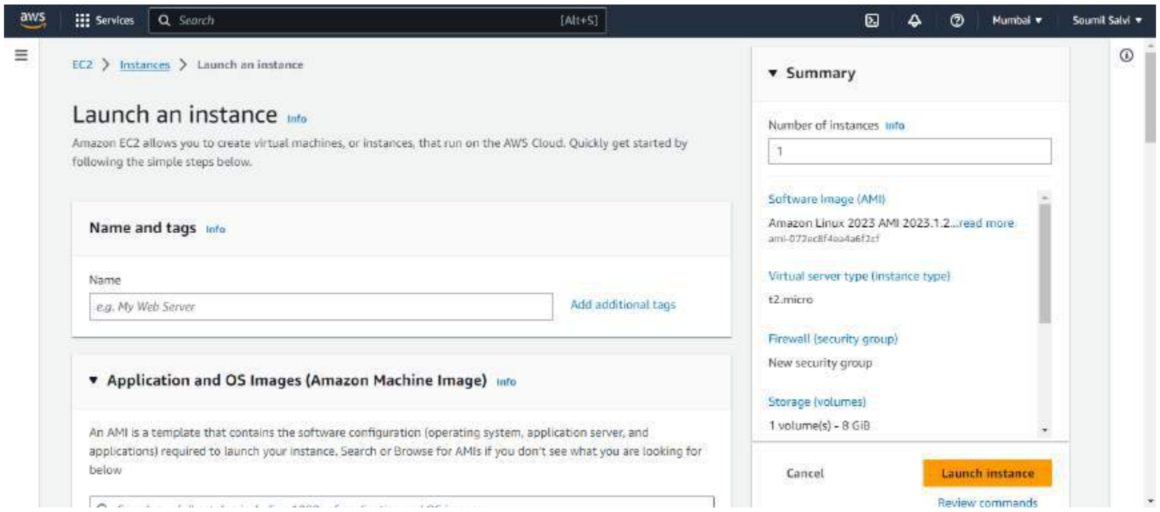
Enable Best Price-

CloudShell Feedback Language Privacy Terms Cookie preferences

© 2023, Amazon Web Services India Private Limited or its affiliates.

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home: 1/1

2. NOW CLICK ON LAUNCH / CREATE NEW INSTANCES.



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: e.g. My Web Server

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

Summary

Number of instances: 1

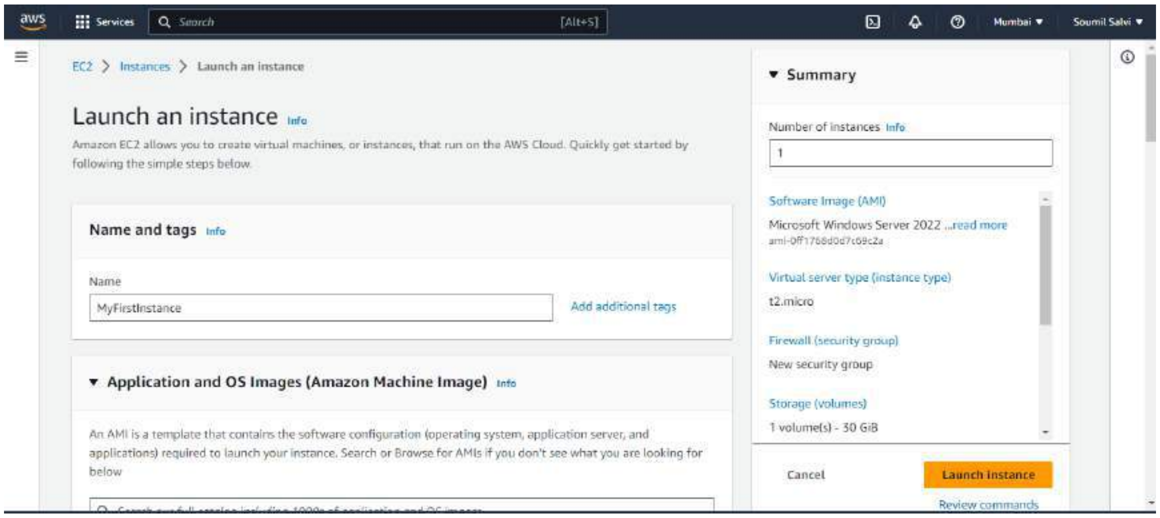
Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more ami-072ec8f4ea4a6f2cf

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

3. Choose any machine you want to create here I am creating UBUNTU(free tier).



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: MyFirstInstance

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

Summary

Number of instances: 1

Software Image (AMI): Microsoft Windows Server 2022 ...read more ami-0ff1766d0d7c69c2a

Virtual server type (instance type): t2.micro

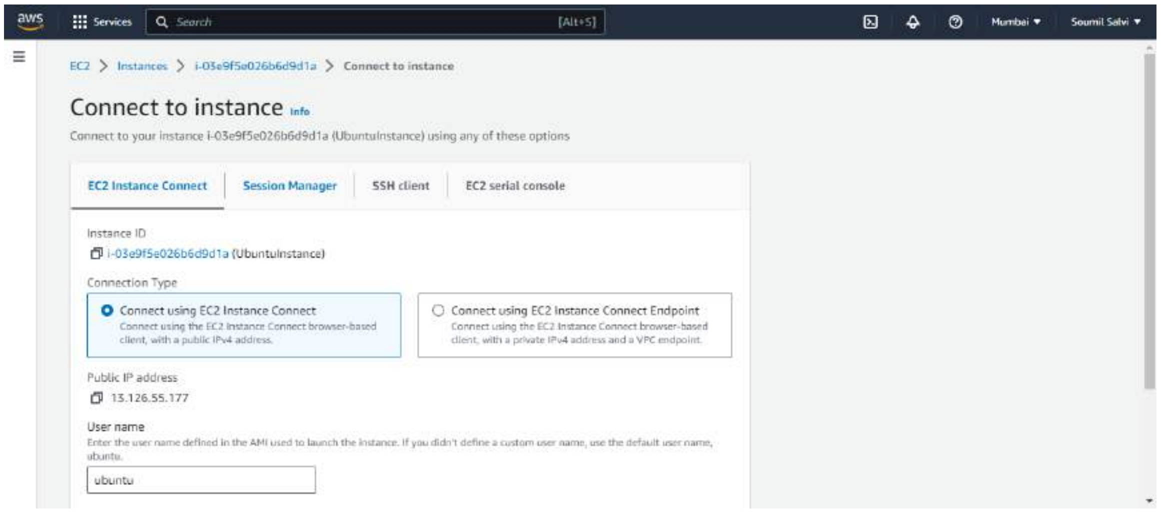
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

4. Now add security group ALL TRAFFIC ,

PROTOCOL – ALL,

SOURCE- ANYWHERE.



Connect to instance Info

Connect to your instance i-03e9f5e026b6d9d1a (UbuntuInstance) using any of these options

Instance ID
i-03e9f5e026b6d9d1a (UbuntuInstance)

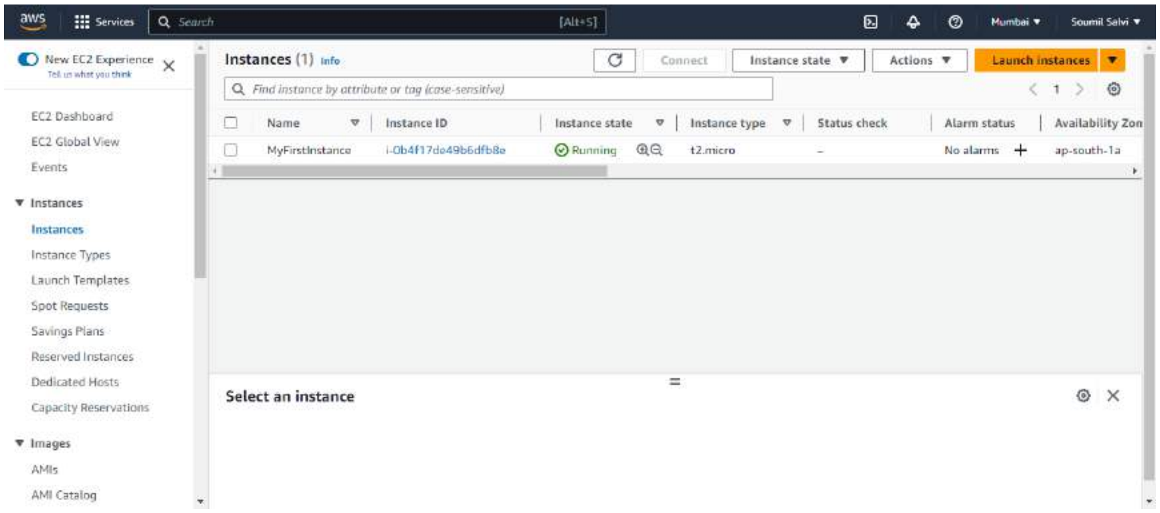
Connection Type
Connect using EC2 Instance Connect — Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint — Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
13.126.55.177

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.
ubuntu

5.NOW WAIT TILL THE STATUS CHECK IS 2/2 and Instance is running.

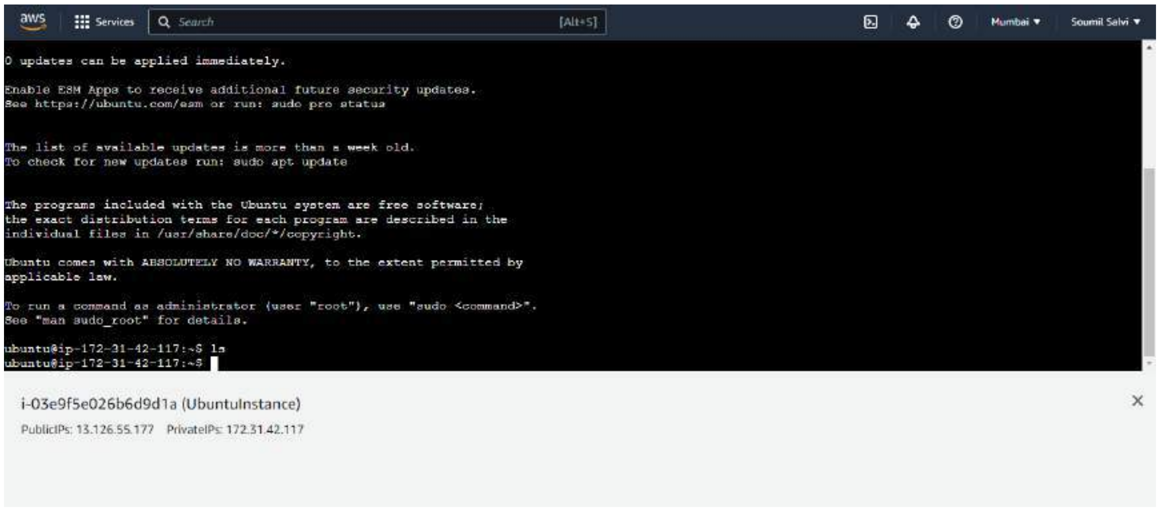
Once Check is complete click on launch instances.



Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
MyFirstInstance	i-0b4f17de49b6dfb8e	Running	t2.micro	–	No alarms	ap-south-1a

6. FOLLOW SOME BASIC LINUX COMMANDS AS SHOWN BELOW-



```
0 updates can be applied immediately.

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


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


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

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

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

ubuntu@ip-172-31-42-117:~$ ls
ubuntu@ip-172-31-42-117:~$
```

i-03e9f5e026b6d9d1a (UbuntuInstance)

PublicIPs: 13.126.55.177 PrivateIPs: 172.31.42.117

CONCLUSION : Hence learned and implemented the steps to create an EC2 machine.

ASSIGNMENT - 2

AIM : To understand the benefits of cloud Infrastructure and setup AWS Cloud 9 IDE, launch AWS Cloud9 IDE and perform collaboration demonstration.

LO MAPPED : LO1

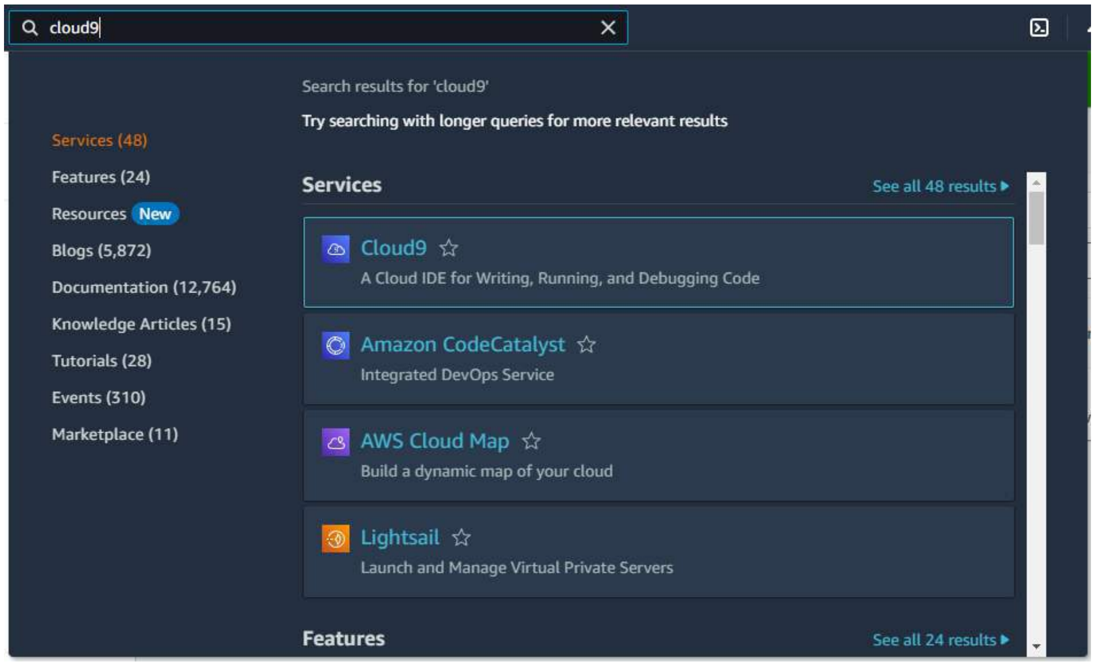
THEORY : AWS Cloud9 is an integrated development environment, or IDE.

The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

You access the AWS Cloud9 IDE through a web browser. You can configure the IDE to your preferences. You can switch color themes, bind shortcut keys, enable programming language-specific syntax coloring and code formatting, and more.

STEPS:

1) Search AWS Cloud in the AWS Dashboard.



Search results for 'cloud9'
Try searching with longer queries for more relevant results

Services — See all 48 results

Cloud9 — A Cloud IDE for Writing, Running, and Debugging Code

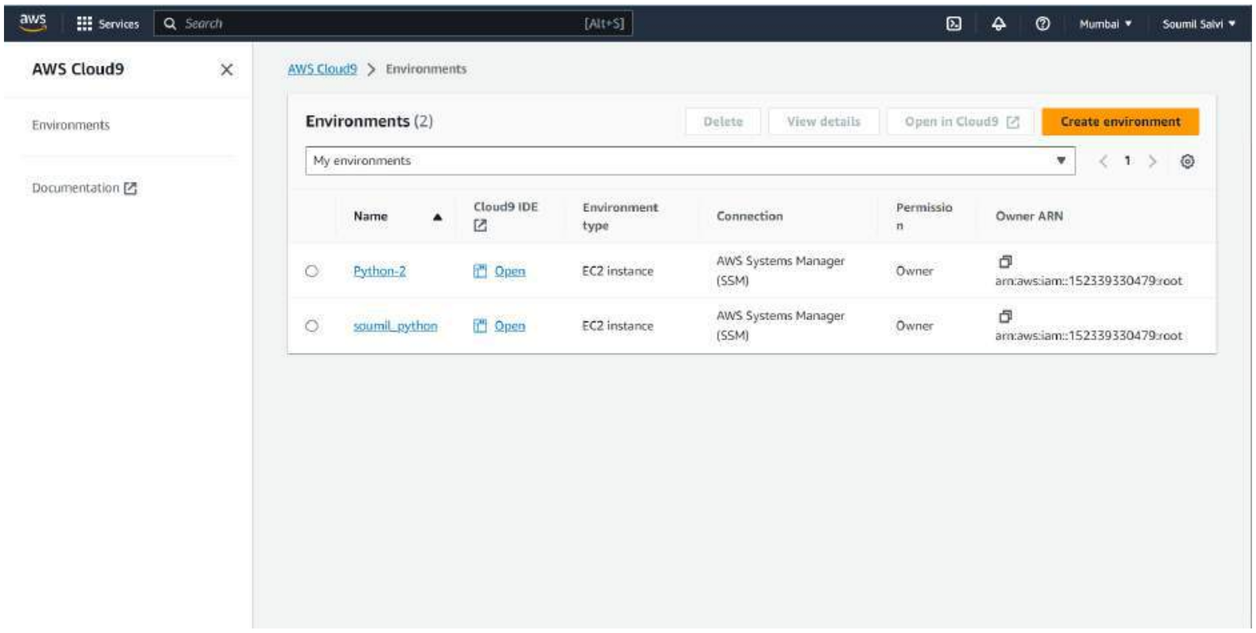
Amazon CodeCatalyst — Integrated DevOps Service

AWS Cloud Map — Build a dynamic map of your cloud

Lightsail — Launch and Manage Virtual Private Servers

Features — See all 24 results

2) Click on Cloud9, Enter your name and create your Cloud9 instance.



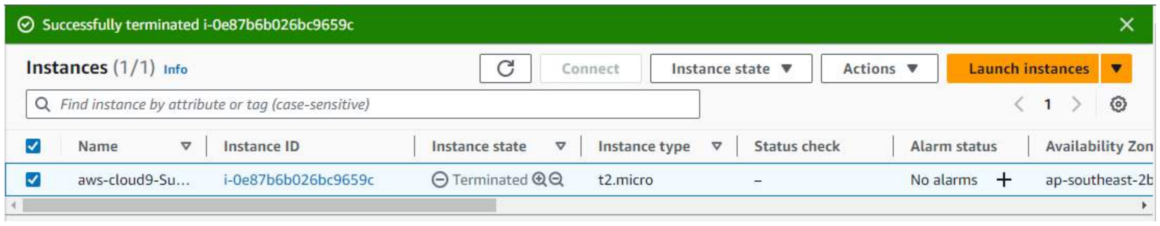
AWS Cloud9 > Environments

Environments (2)

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Python-2	Open	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:iam::152339330479:root
soumil_python	Open	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:iam::152339330479:root

3) Click on open. This opens the Terminal. Start coding.

4) Finally, terminate the instance.



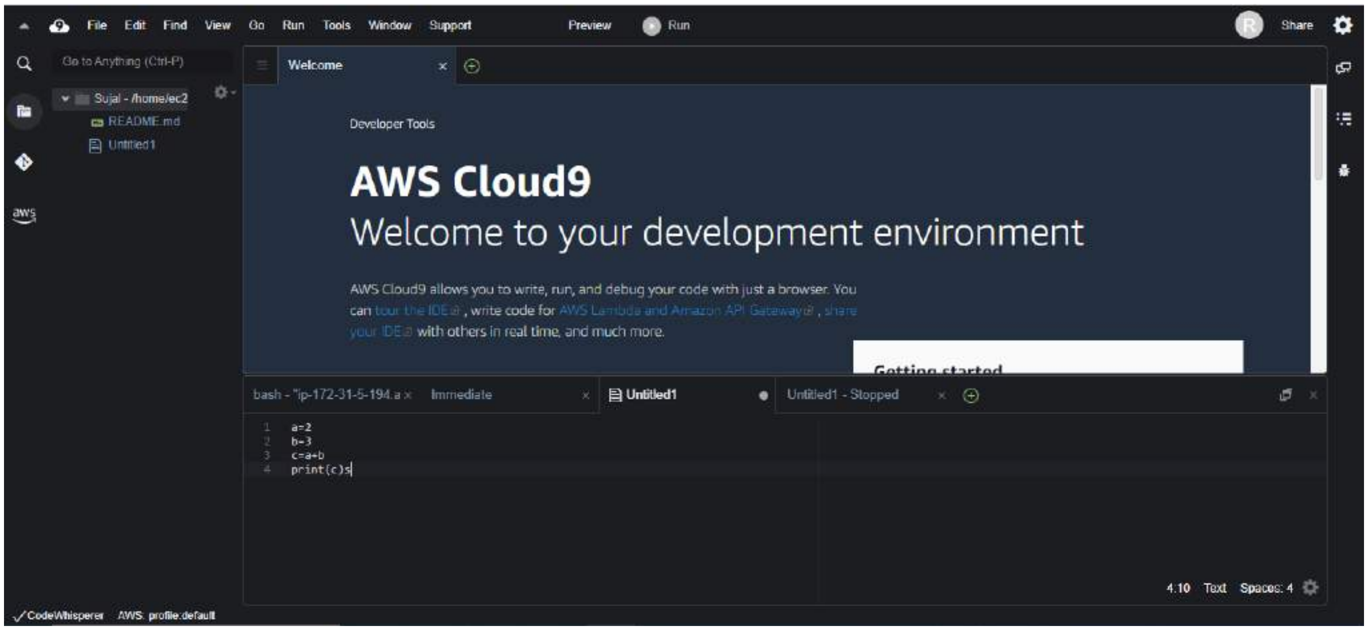
Successfully terminated i-0e87b6b026bc9659c

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
aws-cloud9-Su...	i-0e87b6b026bc9659c	Terminated	t2.micro	–	No alarms	ap-southeast-2b

OUTPUT:

PYTHON:

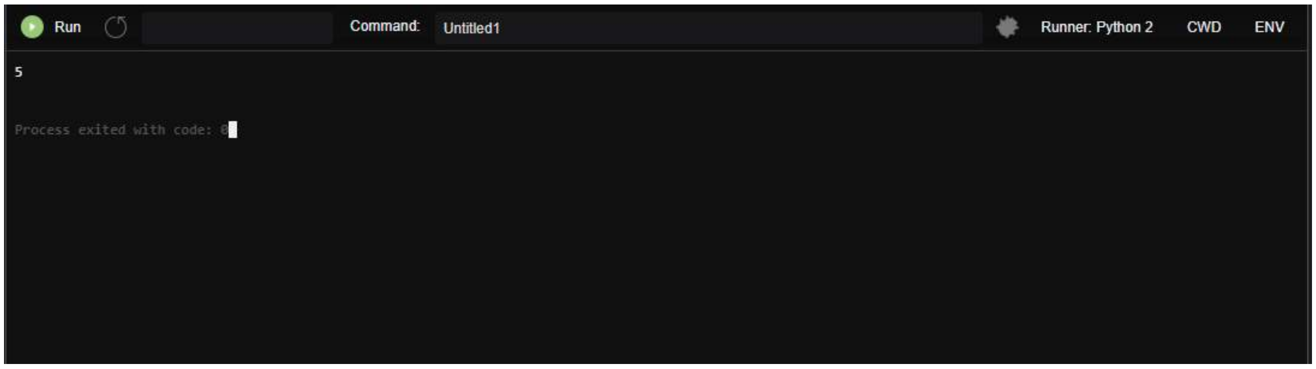


AWS Cloud9

Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

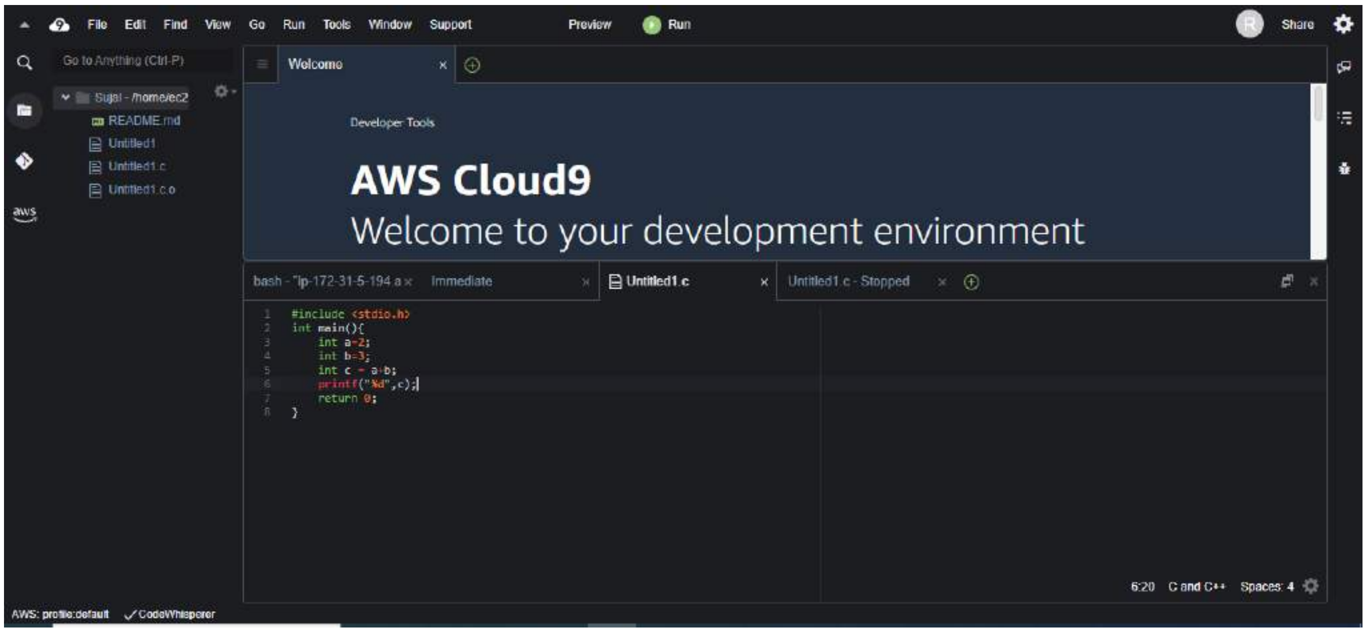
```
a=2
b=3
c=a+b
print(c)s
```



Command: Untitled1 Runner: Python 2

```
5

Process exited with code: 0
```

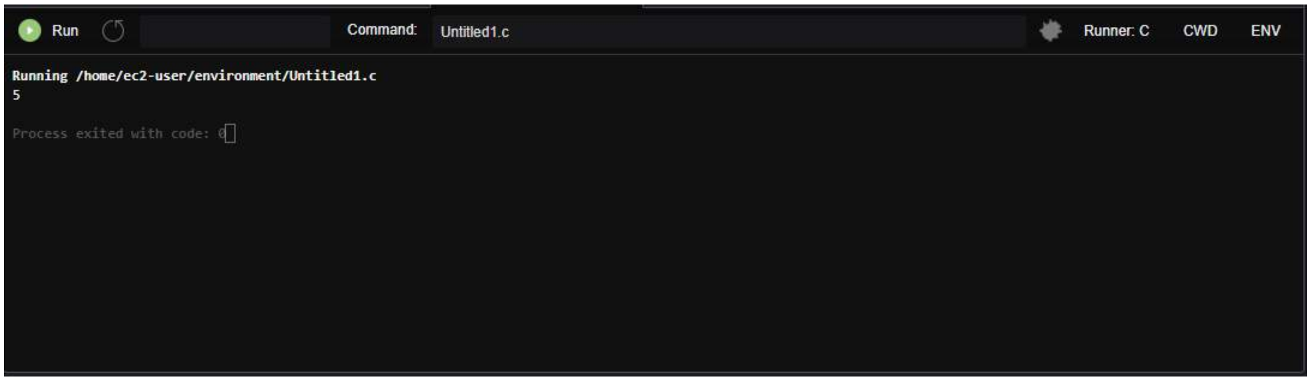
C:



AWS Cloud9

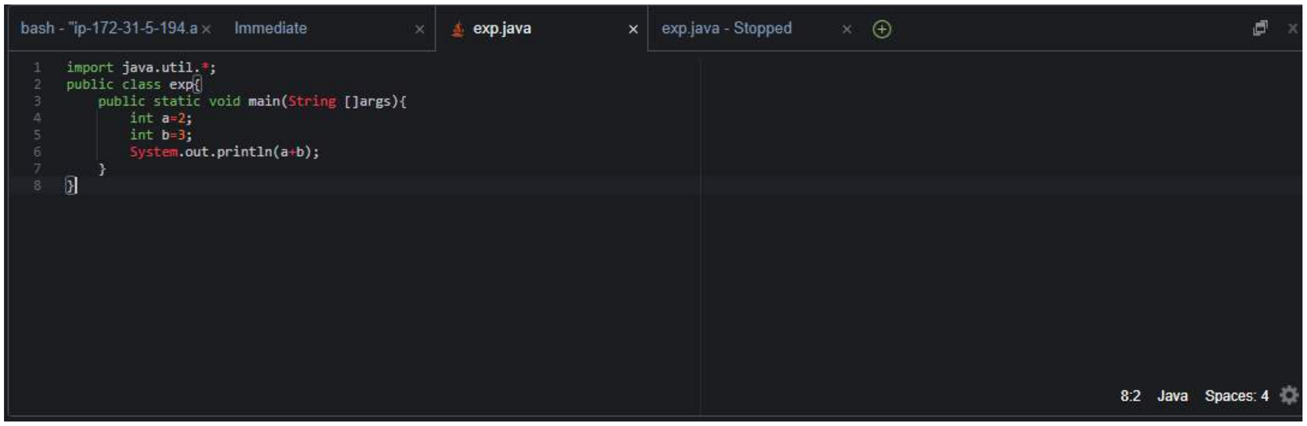
Welcome to your development environment

```
#include <stdio.h>
int main(){
    int a=2;
    int b=3;
    int c = a+b;
    printf("%d",c);
    return 0;
}
```

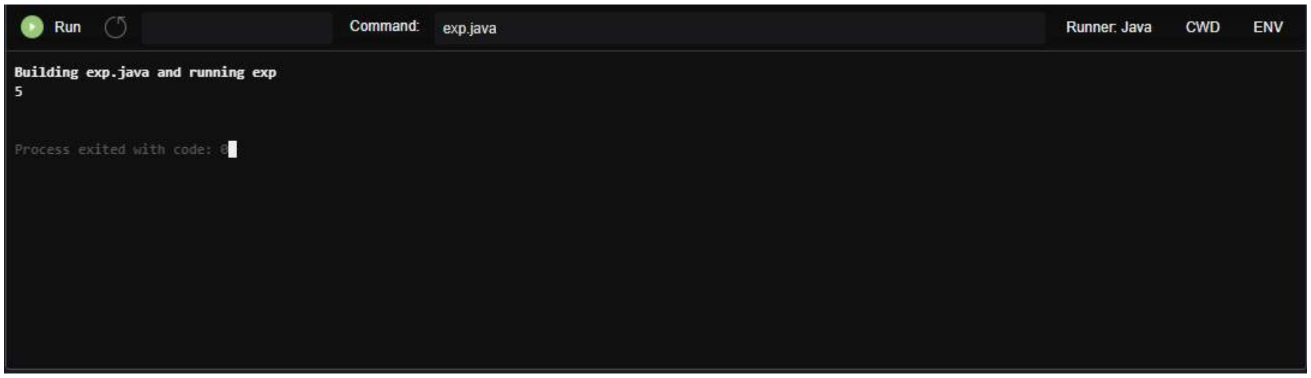


```
Running /home/ec2-user/environment/Untitled1.c
5

Process exited with code: 0
```

JAVA:

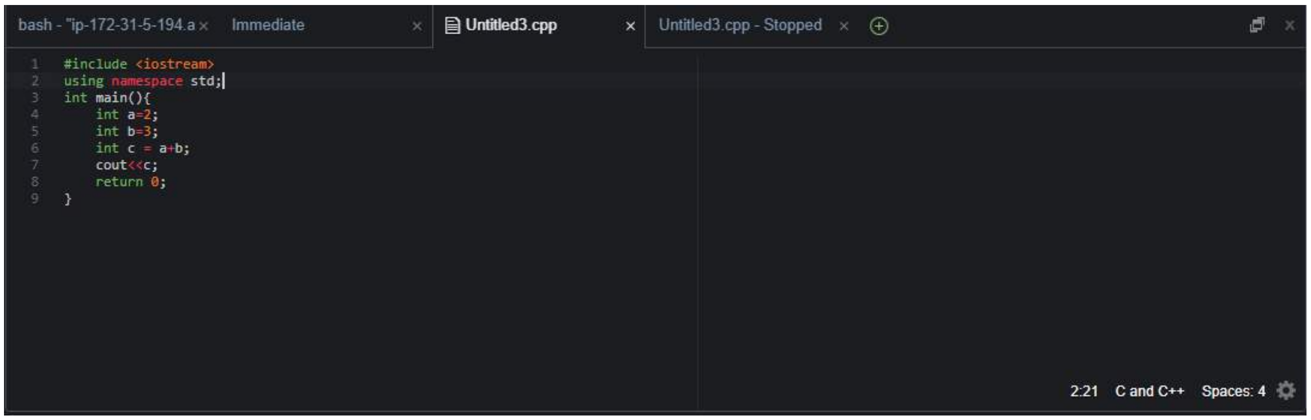


```java
import java.util.*;
public class exp{
    public static void main(String []args){
        int a=2;
        int b=3;
        System.out.println(a+b);
    }
}
```

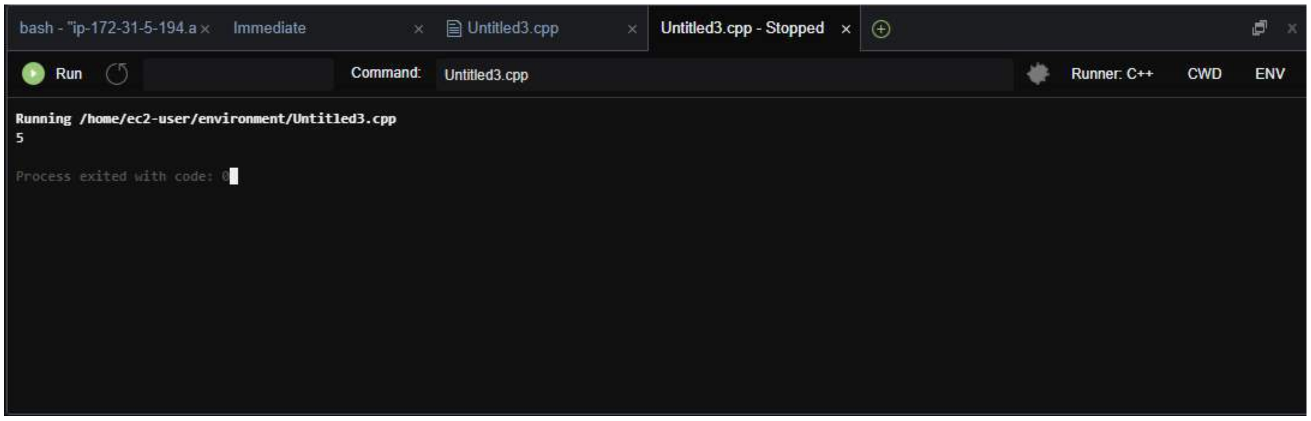


```
Building exp.java and running exp
5

Process exited with code: 0
```

C++:



```
#include <iostream>
using namespace std;
int main(){
    int a=2;
    int b=3;
    int c = a+b;
    cout<<c;
    return 0;
}
```



Command: Untitled3.cpp

```
Running /home/ec2-user/environment/Untitled3.cpp
5

Process exited with code: 0
```

<u>CONCLUSION:</u> AWS Cloud9 instance was created and Python, C, C++ and Java programs were performed using Cloud9 IDE.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 3

AIM : To study AWS S3 service and create a bucket for hosing static web application.

LO MAPPED : LO1,LO2

THEORY :

Amazon Simple Storage Service (Amazon S3) is a scalable object storage service provided by Amazon Web Services (AWS). It is designed to store and retrieve any amount of data from anywhere on the web. To understand the theory behind AWS S3 buckets, let's break down some key concepts:

i. Bucket: A bucket is the fundamental container in S3. It's like a top-level folder or directory where you store your data. Each bucket has a globally unique name, meaning no two buckets can have the same name across all of AWS. For example, you might have a bucket named "mycompanydata."

ii. Objects: Objects are the data files stored within S3 buckets. These can be anything from text files and images to videos and databases. Each object in a bucket has a unique key that distinguishes it from other objects in the same bucket. The combination of the bucket name and object key forms a unique identifier for the object within S3. For example, an object might have a key like "user-profiles/user123.jpg."

iii. Data Consistency: S3 provides strong read-after-write consistency for all objects in your bucket. This means that once you write an object to an S3 bucket, you can immediately read it, and you will always get the latest version of the object.

iv. Data Durability: S3 is designed to provide 99.999999999% (11 nines) durability, which means that your data is highly resilient and protected against data loss. This durability is achieved through data replication across multiple Availability Zones within a region.

v. Data Availability: S3 is designed for high availability, ensuring that your data is accessible when you need it. Availability is typically in the range of 99.99%, and it can be even higher depending on how you configure your bucket.

vi. Storage Classes: S3 offers various storage classes, each with different characteristics and pricing. These storage classes include Standard, Intelligent-Tiering, Glacier, and more. You can choose the appropriate storage class for your specific use case based on factors like data access frequency and retrieval time requirements.

vii. Access Control: You can control access to your S3 buckets and objects using AWS Identity and Access Management (IAM) policies and Access Control Lists (ACLs). This allows you to specify who can upload, download, or delete objects within your buckets.

viii. Data Lifecycle Management: S3 allows you to define data lifecycle policies to automatically transition objects between storage classes or delete them after a certain period. This can help you optimize storage costs.

ix. Versioning: S3 supports versioning, which means you can keep multiple versions of an object in a bucket. This is useful for data recovery and maintaining a version history.

x. Security: S3 offers features like encryption at rest and in transit, which help protect your data. You can also configure bucket policies to restrict access to specific IP addresses or VPCs.

xi. Event Notifications: You can set up event notifications on S3 buckets to trigger actions (e.g., invoking a Lambda function) when specific events occur, such as object creation or deletion.

xii. Cross-Region Replication: You can replicate your S3 data to other AWS regions for disaster recovery, compliance, or low-latency access to data in different geographical locations.

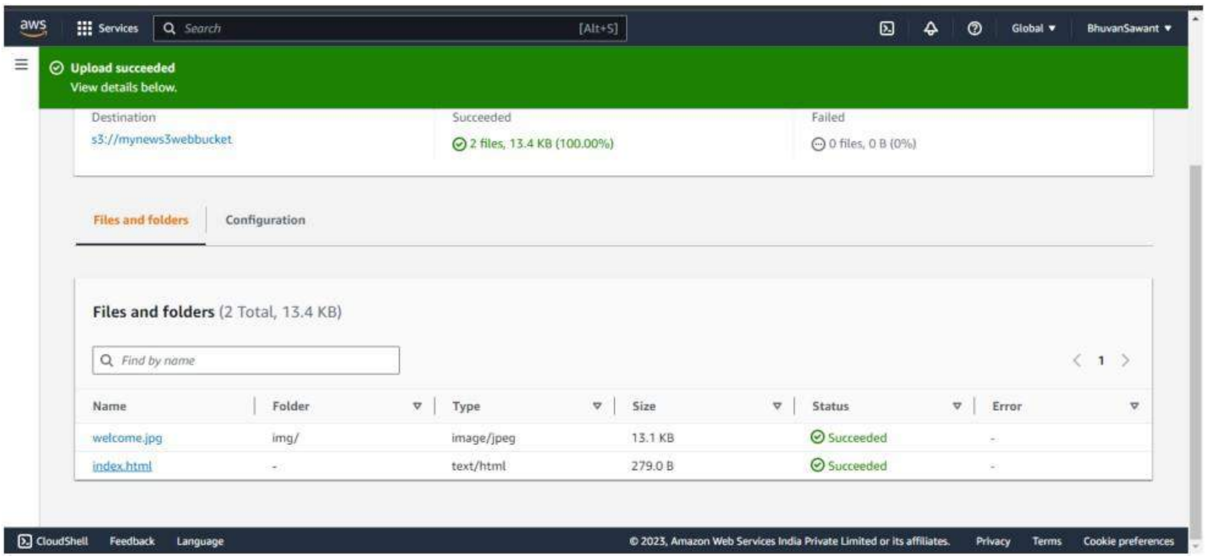
In summary, AWS S3 buckets are versatile, highly durable, and scalable containers for storing objects of various types. They provide robust data management capabilities, security features, and access control mechanisms, making them a fundamental component of many cloud-based applications and data storage solutions. Understanding these concepts is essential for effectively using S3 to store and manage your data in the AWS cloud.

Use Cases:

i. Data Storage: S3 is primarily used for storing and archiving data. It can store virtually any type of data, including documents, images, videos, backups, logs, and databases.
ii. Static Website Hosting: You can host static websites directly on S3 by configuring it to serve HTML, CSS, JavaScript, and other web assets. This is a cost-effective way to host websites with high availability and scalability.
iii. Data Backup and Recovery: Many organizations use S3 as a backup and disaster recovery solution. It offers high durability and can be configured to automatically backup data from on-premises servers or other AWS resources.
iv. Data Lakes: S3 is often used as a foundation for building data lakes, which are centralized repositories for storing and analyzing large volumes of structured and unstructured data. It can integrate with various analytics and data processing services in AWS.
v. Content Delivery: Amazon S3 can be integrated with Amazon CloudFront (AWS's content delivery network) to deliver content like images, videos, and other media files to users around the world with low latency and high data transfer speeds.
vi. Big Data and Analytics: S3 is commonly used as a data store for big data and analytics platforms like Amazon EMR, Amazon Redshift, and AWS Glue. It allows you to store vast amounts of data and access it for analysis.
vii. IoT Data Storage: Internet of Things (IoT) devices can send data directly to S3 for storage and analysis, enabling IoT applications to collect and process large volumes of sensor data.
viii. Log Storage and Analysis: Many organizations use S3 to store log files generated by applications and infrastructure. Services like Amazon Athena can be used to analyze these logs directly in S3.

Steps :-

i. First login to your AWS account.



Upload succeeded
View details below.

Destination	Succeeded	Failed
s3://mynews3webbucket	2 files, 13.4 KB (100.00%)	0 files, 0 B (0%)

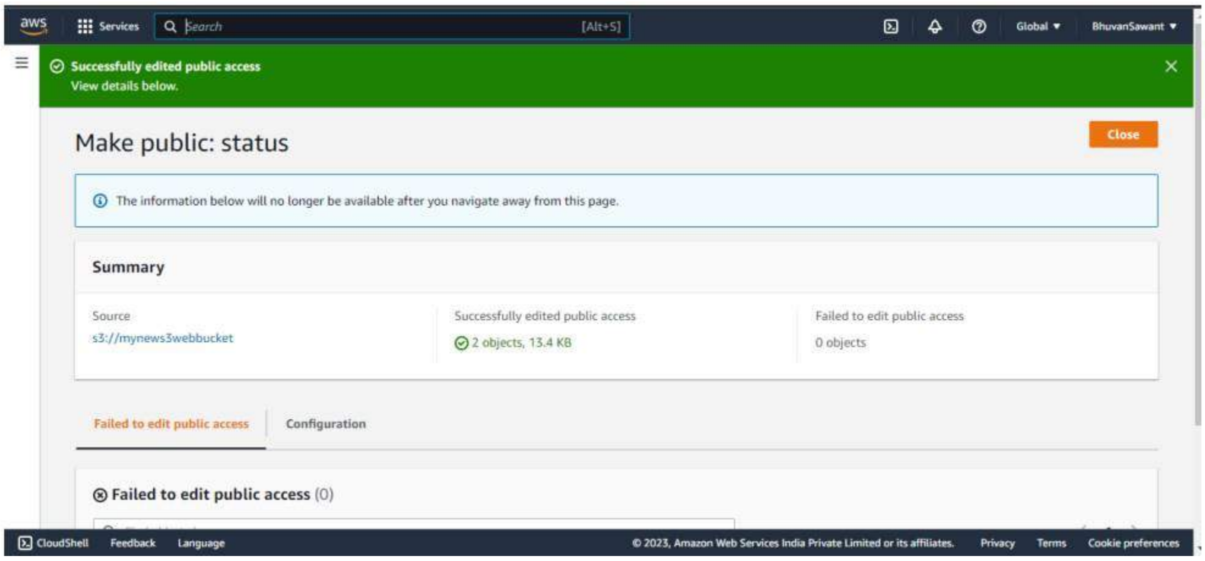
Files and folders | Configuration

Files and folders (2 Total, 13.4 KB)

Name	Folder	Type	Size	Status	Error
welcome.jpg	img/	image/jpeg	13.1 KB	Succeeded	-
index.html	-	text/html	279.0 B	Succeeded	-

© 2023, Amazon Web Services India Private Limited or its affiliates.

ii. Then create a new S3 bucket.



Successfully edited public access
View details below.

Make public: status

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

Summary

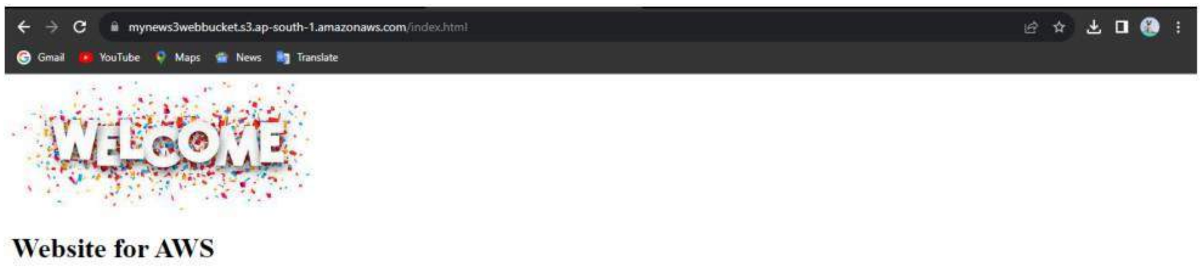
Source	Successfully edited public access	Failed to edit public access
s3://mynews3webbucket	2 objects, 13.4 KB	0 objects

Failed to edit public access | Configuration

Failed to edit public access (0)

© 2023, Amazon Web Services India Private Limited or its affiliates.

iii. After that host a static web page using the S3 bucket.



mynews3webbucket.s3.ap-south-1.amazonaws.com/index.html

Website for AWS

CONCLUSION : In this assignment we learnt about AWS S3 bucket and hosted a static web page using the S3 bucket.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 4

AIM : To study AWS Code Pipeline and deploy web application using Code Pipeline.

LO MAPPED : LO1,LO2

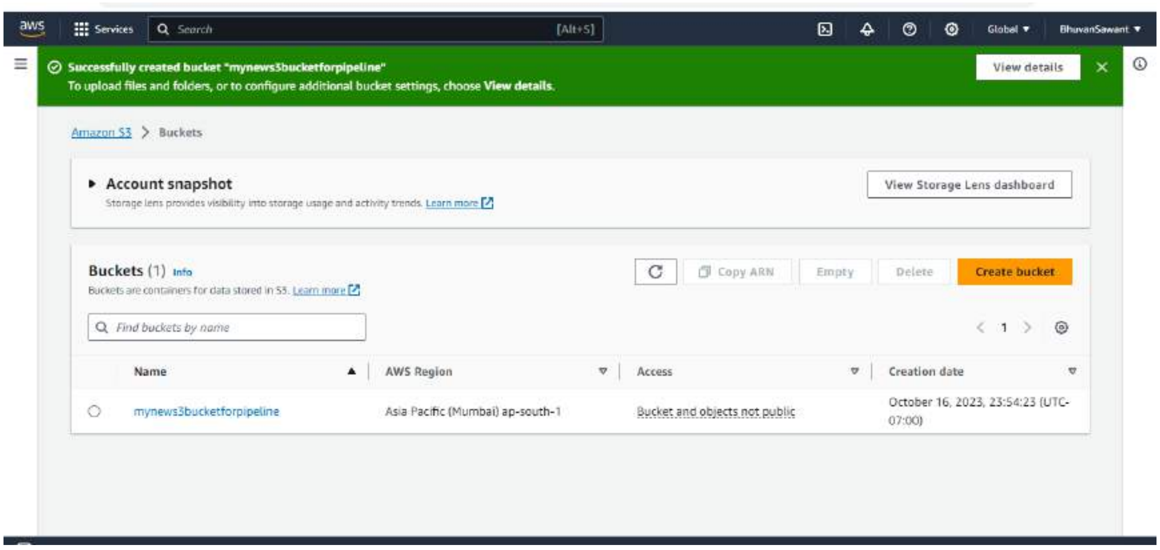
THEORY :
AWS CodePipeline is a continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS).

The key aspects of AWS CodePipeline:

Overview:

1. CI/CD Workflow:

    - AWS CodePipeline facilitates the automation of the build, test, and deployment phases of the release process. It allows you to define a series of stages, each of which can represent a phase in your release pipeline.



Successfully created bucket "mynews3bucketforpipeline"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Amazon S3 > Buckets

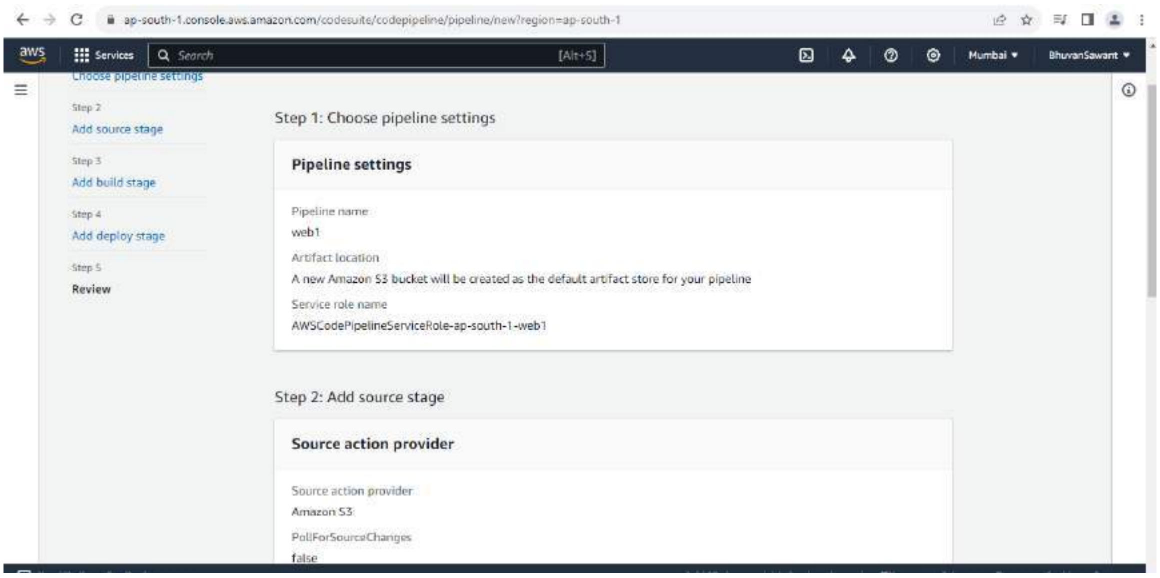
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
mynews3bucketforpipeline	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	October 16, 2023, 23:54:23 (UTC-07:00)

2. Integration with Other AWS Services:

    - CodePipeline integrates with various AWS services, such as AWS CodeBuild for building applications, AWS CodeDeploy for automating deployments, and AWS Lambda for running custom actions.



Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
web1

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

Service role name
AWSCodePipelineServiceRole-ap-south-1-web1

Step 2: Add source stage

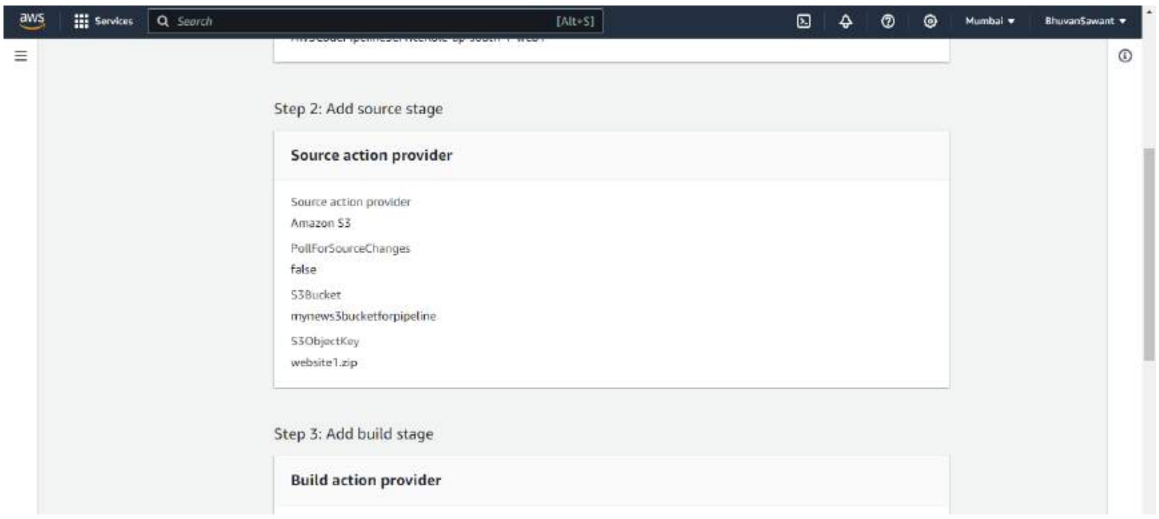
Source action provider

Source action provider
Amazon S3

PollForSourceChanges
false

3. Pipeline Execution:

   - Pipelines consist of a series of stages, and each stage can have one or more actions. Actions represent a task, such as source code retrieval or deployment to a specific environment.



Step 2: Add source stage

Source action provider

Source action provider
Amazon S3

PollForSourceChanges
false

S3Bucket
mynews3bucketforpipeline

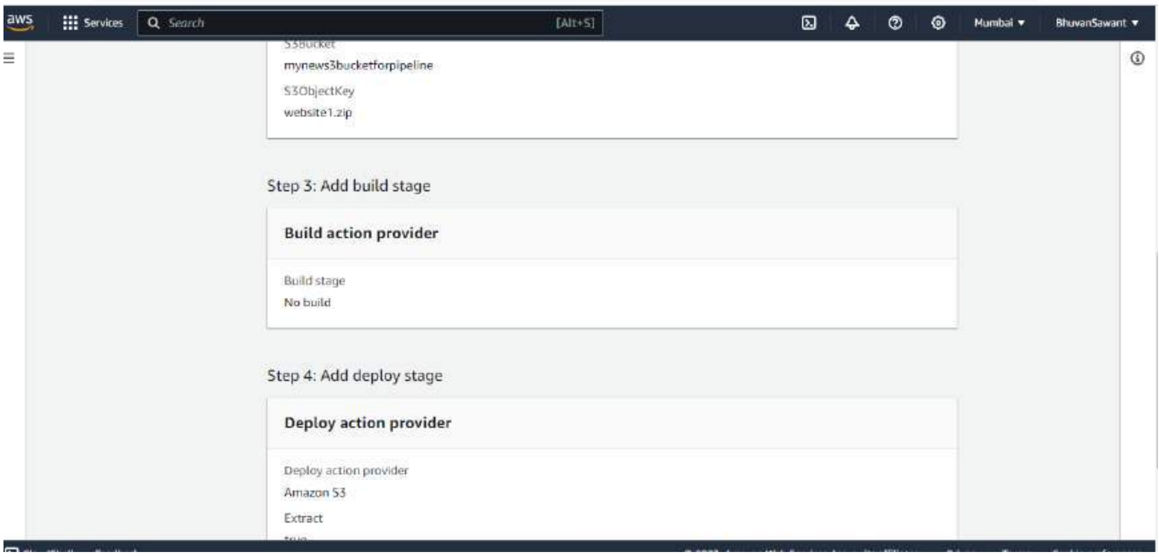
S3ObjectKey
website1.zip

Step 3: Add build stage

Build action provider

4. Source Providers:

   - CodePipeline supports integration with various source code repositories, including AWS CodeCommit, GitHub, and Amazon S3.



S3Bucket
mynews3bucketforpipeline
S3ObjectKey
website1.zip

Step 3: Add build stage

Build action provider

Build stage
No build

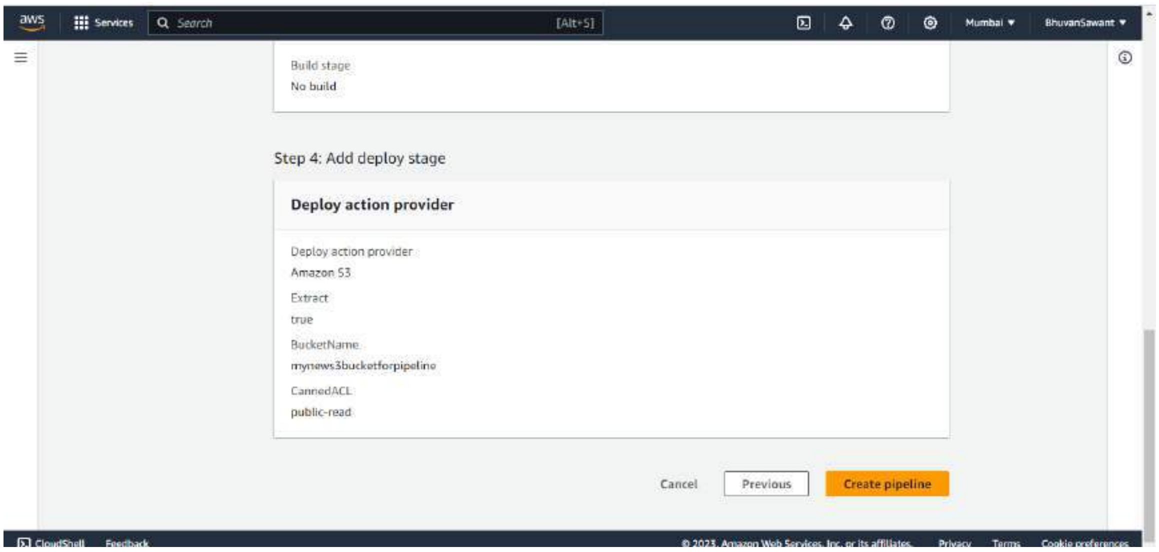
Step 4: Add deploy stage

Deploy action provider

Deploy action provider
Amazon S3
Extract

5. Artifact Management:

    - CodePipeline uses artifacts to store the files and data needed for each action in a pipeline. Artifacts can be passed between stages to ensure consistency in the deployment process.



Build stage
No build

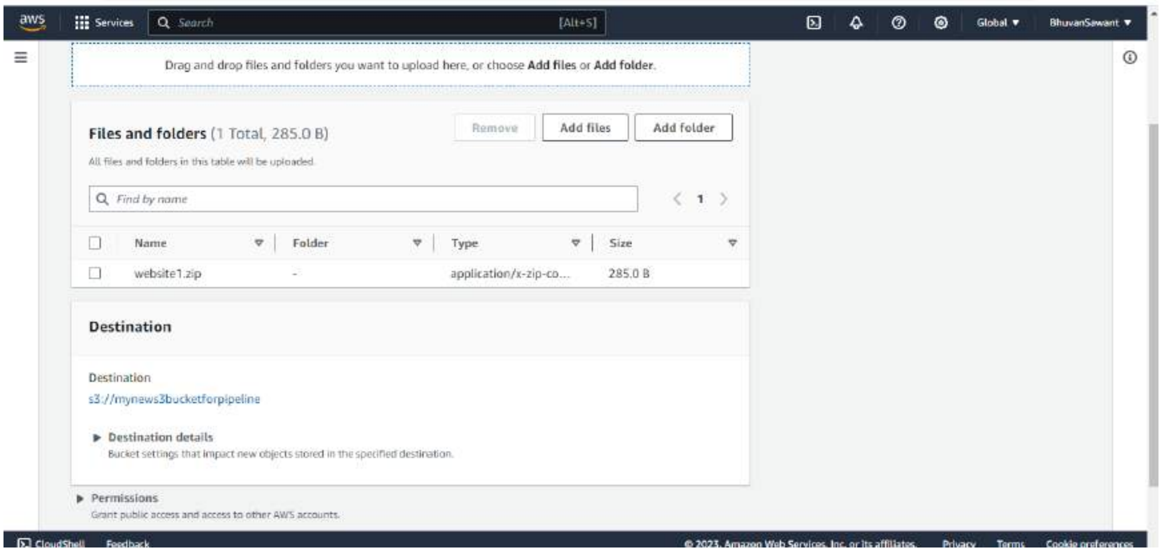
Step 4: Add deploy stage

Deploy action provider

Deploy action provider
Amazon S3
Extract
true
BucketName
mynews3bucketforpipeline
CannedACL
public-read

6. Integration with Third-Party Tools:

    - Besides AWS services, CodePipeline supports integration with third-party tools. This is achieved through custom actions, which allow you to use external tools and scripts in your pipeline.


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

Files and folders (1 Total, 285.0 B)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
website1.zip	-	application/x-zip-co...	285.0 B

Destination

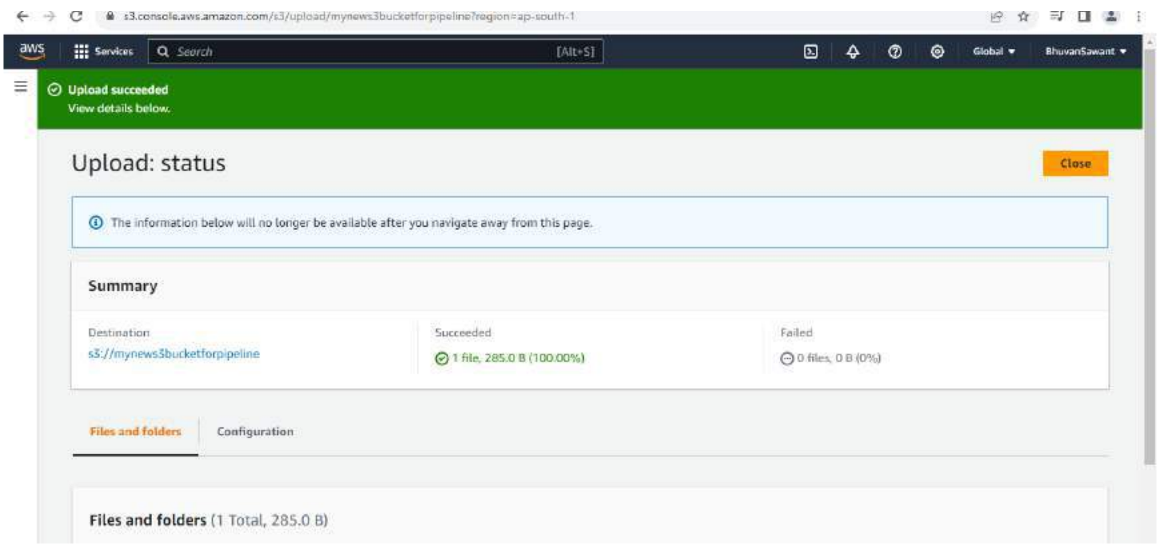
Destination
s3://mynews3bucketforpipeline

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

Permissions
Grant public access and access to other AWS accounts.

7. Pipeline Visualizations:

   - CodePipeline provides a visual representation of your release process, making it easy to understand and monitor the status of each stage and action.


s3.console.aws.amazon.com/s3/upload/mynews3bucketforpipeline?region=ap-south-1

Upload succeeded
View details below.

Upload: status

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

Summary

Destination	Succeeded	Failed
s3://mynews3bucketforpipeline	1 file, 285.0 B (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 285.0 B)

Key Concepts:

1. Pipeline:

   - A pipeline is a series of stages that represents your release process. Each stage can contain one or more actions.

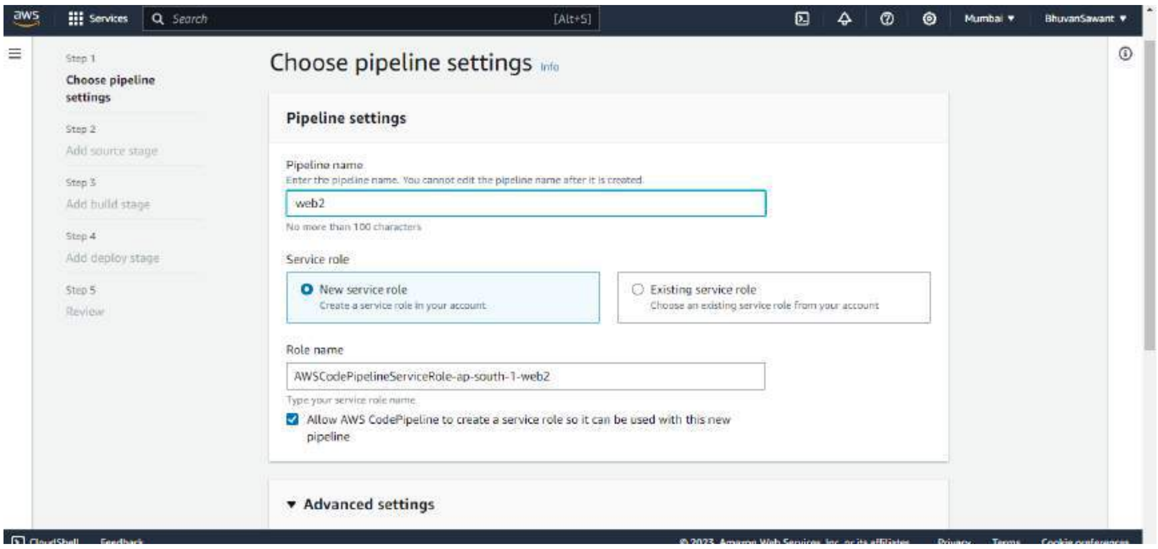
2. Stage:

   - A stage is a logical unit in a pipeline, representing a phase in the release process. Stages are executed sequentially.

3. Action:
    - An action represents a task within a stage. Actions can include tasks such as building code, deploying to a test environment, or running tests.
4. Artifact:
    - Artifacts are the files and data that are produced as a result of an action. They are used to pass information between stages in a pipeline.

To deploy web application using CodePipeline here are the following steps to be followed:

1. Set Up Source Stage:
    - Configure a source stage in AWS CodePipeline, linking to your version control system (e.g., CodeCommit, GitHub).



Choose pipeline settings

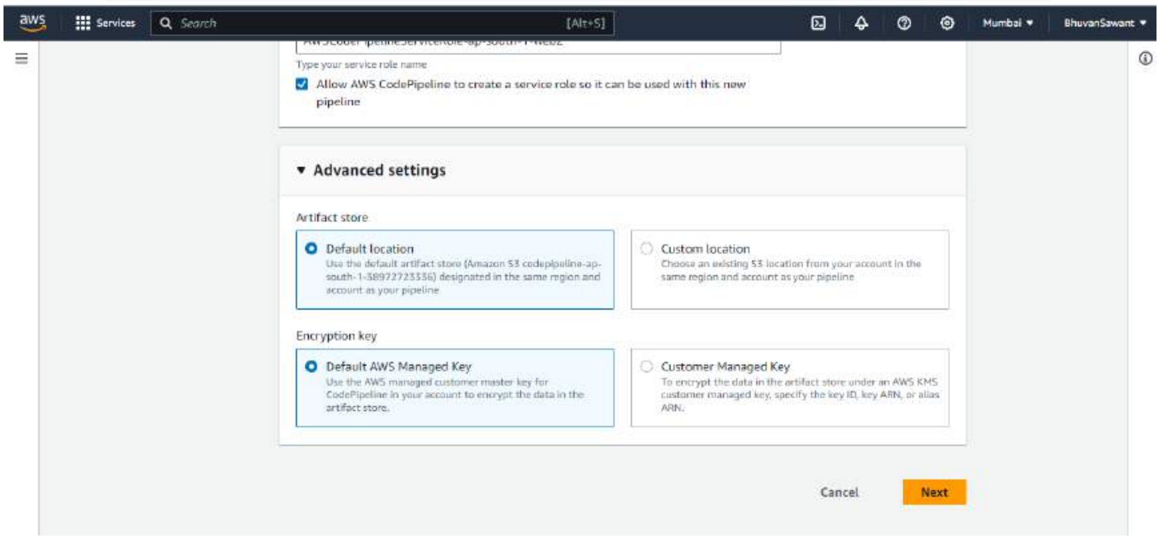
Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
web2
No more than 100 characters

Service role
[✓ New service role — Create a service role in your account] Existing service role — Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-ap-south-1-web2
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

2. Configure Build Stage:
    - Set up a build stage using AWS CodeBuild to compile, test, and package your web application.



Type your service role name

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

Advanced settings

Artifact store

[✓ Default location] Use the default artifact store (Amazon S3 codepipeline-ap-south-1-38972723336) designated in the same region and account as your pipeline

Custom location: Choose an existing S3 location from your account in the same region and account as your pipeline

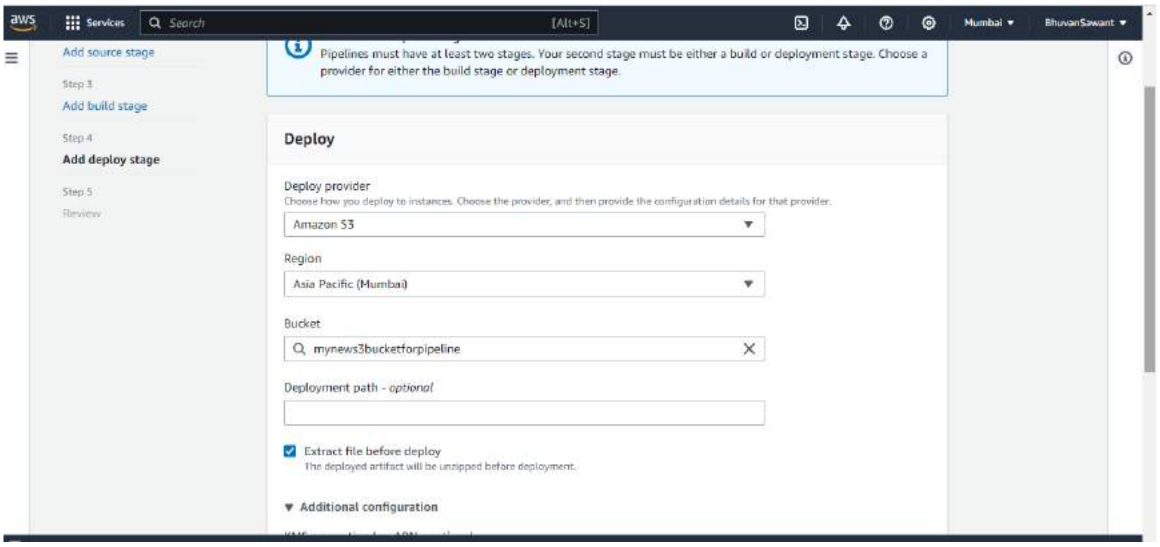
Encryption key

[✓ Default AWS Managed Key] Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

Customer Managed Key: To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

3. Define Deployment Stage:

    - Create a deployment stage using AWS CodeDeploy or another deployment provider to deploy your application to target environments.



Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
Amazon S3

Region
Asia Pacific (Mumbai)

Bucket
mynews3bucketforpipeline

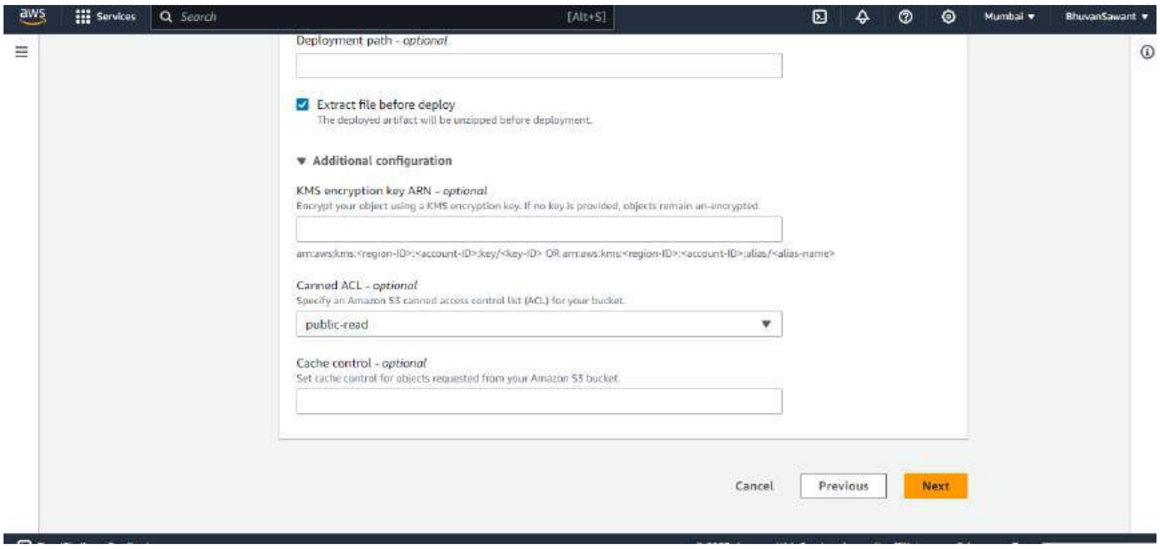
Deployment path - *optional*

☑ Extract file before deploy
The deployed artifact will be unzipped before deployment.

Additional configuration

4. Configure Approval (Optional):

    - Optionally, add a manual approval stage to review and approve deployments before proceeding to the next stage.



Deployment path - *optional*

☑ Extract file before deploy
The deployed artifact will be unzipped before deployment.

▼ Additional configuration

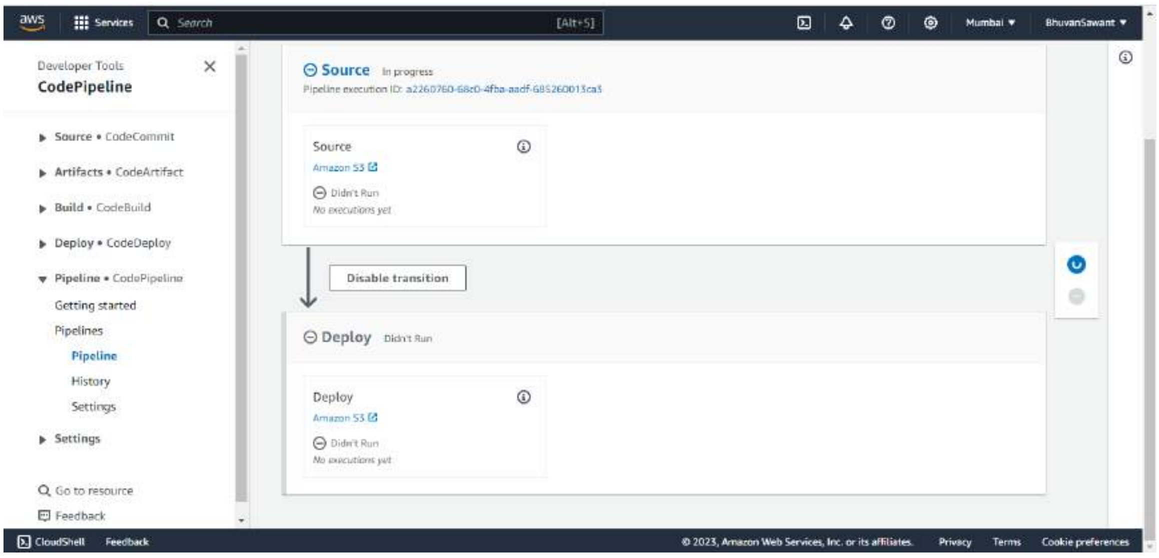
KMS encryption key ARN - *optional*
Encrypt your object using a KMS encryption key. If no key is provided, objects remain un-encrypted.

arn:aws:kms:<region-ID>:<account-ID>:key/<key-ID> OR arn:aws:kms:<region-ID>:<account-ID>:alias/<alias-name>

Canned ACL - *optional*
Specify an Amazon S3 canned access control list (ACL) for your bucket.
public-read

Cache control - *optional*
Set cache control for objects requested from your Amazon S3 bucket.

5. Artifact Passing:
   - Ensure proper passing of artifacts between stages to maintain consistency in the deployment process.



Source In progress
Pipeline execution ID: a2260760-68c0-4fba-aadf-685260013ca5

Source
Amazon S3
Didn't Run
No executions yet

Deploy Didn't Run

Deploy
Amazon S3
Didn't Run
No executions yet

6. Add Monitoring (Optional):
   - Integrate monitoring tools (e.g., AWS CloudWatch) to track the performance and health of your application during and after deployment.
7. Configure Notifications (Optional):
   - Set up notifications using AWS SNS or other services to receive alerts about pipeline events and status changes.
8. Test and Validate:

- Test the pipeline by triggering a build, ensuring that each stage executes successfully, and the application deploys as expected.

9. Modify Pipeline as Needed:
    - Make adjustments to the pipeline configuration based on the specific requirements of your web application and deployment process.
10. Continuous Improvement:
    - Implement continuous improvement practices, such as monitoring feedback, optimizing build and deployment scripts, and iterating on the pipeline structure.

CONCLUSION : In this assignment we learned how to host static web page through codepipeline and deploy web application using Code Pipeline.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 5

AIM : To understand the Kubernetes Cluster Architecture.

LO MAPPED : LO1,LO5

THEORY :
Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Let's take a look at why Kubernetes is so useful by going back in time.

- Traditional deployment era: Early on, organizations ran applications on physical servers. There was no way to define resource boundaries for applications in a physical server, and this caused resource allocation issues. For example, if multiple applications run on a physical server, there can be instances where one application would take up most of the resources, and as a result, the other applications would underperform. A solution for this would be to run each application on a different physical server. But this did not scale as resources were underutilized, and it was expensive for organizations to maintain many physical servers.
- Virtualized deployment era: As a solution, virtualization was introduced. It allows you to run multiple Virtual Machines (VMs) on a single physical server's CPU. Virtualization allows applications to be isolated between VMs and provides a level of security as the information of one application cannot be freely accessed by another application.
- Container deployment era: Containers are similar to VMs, but they have relaxed isolation properties to share the Operating System (OS) among the applications. Therefore, containers are considered lightweight. Similar to a VM, a container has its own filesystem, share of CPU, memory, process space, and more. As they are decoupled

from the underlying infrastructure, they are portable across clouds and OS distributions.

Why you need Kubernetes and what it can do

Kubernetes provides you with:

• Service discovery and load balancing Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.

• Storage orchestration Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.

• Automated rollouts and rollbacks You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.

• Automatic bin packing You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.

• Self-healing Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

• Secret and configuration management Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Q1)What are the various Kubernetes services running on nodes? Describe the role of each service.

Here are the key services running on Kubernetes nodes and their roles:

- Container Runtime: Docker (or another container runtime): The container runtime is responsible for running and managing containers. Docker is a common choice, but Kubernetes can also work with container runtimes like containerd and CRI-O. Its role is to create, start, stop, and manage containers based on container images.
- Kubelet: The Kubelet is an agent running on each node that communicates with the Kubernetes control plane (master node). It ensures that containers are running in a Pod (the smallest deployable unit in Kubernetes). Kubelet pulls container images and starts, stops, and restarts containers as needed.
- Kube Proxy: Kube Proxy is responsible for network connectivity within the cluster. It maintains network rules on each node to enable communication between different Pods. It sets up and manages network routes, ensuring that services and Pods can connect to each other.
- Container Network Interface (CNI): CNI plugins provide the network abstraction needed for connecting containers within a Pod and across Pods on different nodes. Kubernetes allows various CNI plugins like Calico, Flannel, and Weave, which implement different network policies and features.
- Node Status: The Node Status service periodically reports the node's condition to the control plane. This information helps the control plane make scheduling decisions. It communicates the node's available resources, capacity, and overall health.

Q2) What is Pod Disruption Budget (PDB)?

A Pod Disruption Budget (PDB) is a policy object in Kubernetes that allows you to define and control the allowable disruption or unavailability of a set of pods during voluntary disruptions, such as node draining or maintenance activities. PDBs are used to ensure that critical workloads have a certain level of availability and are not unduly disrupted when nodes need to be updated, scaled down, or otherwise modified.

Key aspects of a Pod Disruption Budget (PDB) include:

- Targeted Pods: PDBs are applied to a specific set of pods by specifying a set of labels and selectors. This defines which pods the PDB applies to.
- MaxUnavailable: The most crucial aspect of a PDB is specifying the maxUnavailable field. This field determines the maximum number of pods that can be unavailable during a disruption event. It can be specified as an absolute number (e.g., 1) or a percentage (e.g., 10%). This ensures that a minimum number of pods within the defined set remain available.
- Selector and Labels: The PDB uses selector rules to select pods based on their labels. Only pods matching the selector rules will be subject to the disruption budget.
- Availability Constraints: PDBs can be used to enforce constraints on the availability of pods, helping to maintain a desired level of redundancy and fault tolerance for important workloads.
- Preventing Disruptions: When the disruption budget is violated, Kubernetes will prevent any action that would cause more pods to become unavailable than specified by the PDB. This can include scaling down replicas, draining nodes, or terminating pods.

Q3) What is the role of Load Balance in Kubernetes?

Load balancing in Kubernetes plays a critical role in ensuring the availability, scalability, and reliability of applications and services running within the cluster. It involves distributing incoming network traffic or requests across multiple pods or replicas of an application, thereby optimizing resource utilization and improving fault tolerance. Here are the key roles and benefits of load balancing in Kubernetes:

- Distribute Traffic: Load balancers evenly distribute incoming traffic or requests across a set of backend pods or services. This distribution prevents overloading individual pods and ensures efficient utilization of resources.
- High Availability: Load balancing helps ensure high availability of services by distributing traffic across multiple pods. In the event of a

pod failure, the load balancer redirects traffic to healthy pods, minimizing service disruption.

- Scaling: As your application traffic increases, you can scale your application horizontally by adding more pods or replicas. Load balancers automatically adapt to the increased capacity, spreading traffic across the expanded set of pods.
- Service Discovery: Load balancers provide a stable endpoint (e.g., a virtual IP address) for clients to access a service. The actual pods behind the service can change (e.g., due to scaling, updates, or failures), but clients don't need to be aware of these changes.
- Health Checking: Load balancers can perform health checks on backend pods to verify their availability and responsiveness. If a pod becomes unhealthy, the load balancer stops sending traffic to it, ensuring that clients only connect to healthy instances.
-

CONCLUSION : Understood about Kubernetes and Kubernetes cluster architecture

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 6

AIM : To understand terraform lifecycle, core concepts/ terminologies and install it on a linux machine.

LO MAPPED : LO1,LO5

THEORY :
Terraform, developed by HashiCorp, is an open-source infrastructure as code (IaC) software tool that allows users to define and provision data center infrastructure using a declarative configuration language. In simpler terms, it lets you codify your infrastructure, enabling consistent and reproducible deployments. It is used for defining, provisioning, and managing cloud infrastructure using a declarative language.

Core Concepts:

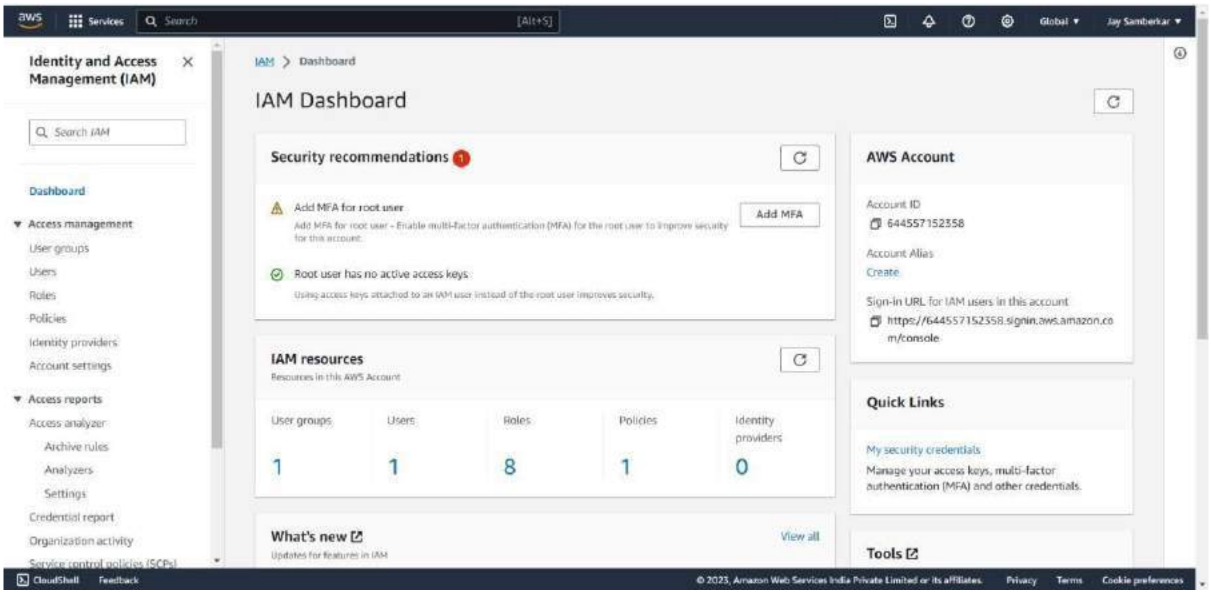
- Providers: Terraform uses providers to interact with cloud services. Examples include AWS, Azure, Google Cloud, and many others. Each provider offers resource types that can be managed.
- Resources: These are the primary components in Terraform. A resource might be a physical component such as an EC2 instance in AWS or a database in Azure.
- State: Terraform maintains a state file that maps real-world resources to your configuration. This state is used to determine what Terraform will do on the next apply.
- Modules: These are containers for multiple resources that are used together. They provide a way to group resources, and they can be used to create reusable infrastructure components.
- Variables and Outputs: Variables allow for parameterization of the Terraform configuration, making it more dynamic and flexible.

Outputs are a way to get information about the infrastructure, like IP addresses or DNS names.

- Provisioners: While not always recommended due to their imperative nature, provisioners can be used to model specific actions on the local machine or on a remote machine, like executing scripts.

Steps :

1. Login and Search IAM on AWS Console and select the first option.



Security recommendations 1

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

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources
Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
1	1	8	1	0

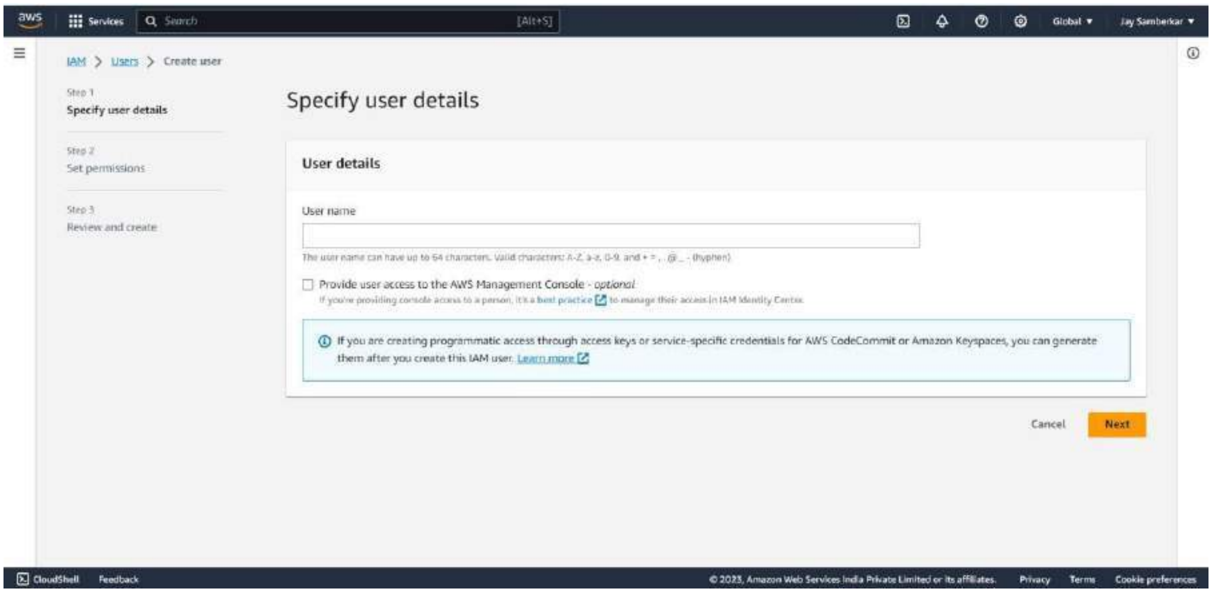
What's new
Updates for features in IAM

AWS Account
Account ID
644557152358
Account Alias
Create
Sign-in URL for IAM users in this account
https://644557152358.signin.aws.amazon.com/console

Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

© 2023, Amazon Web Services India Private Limited or its affiliates.

2. Head to users and Click on Create User



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

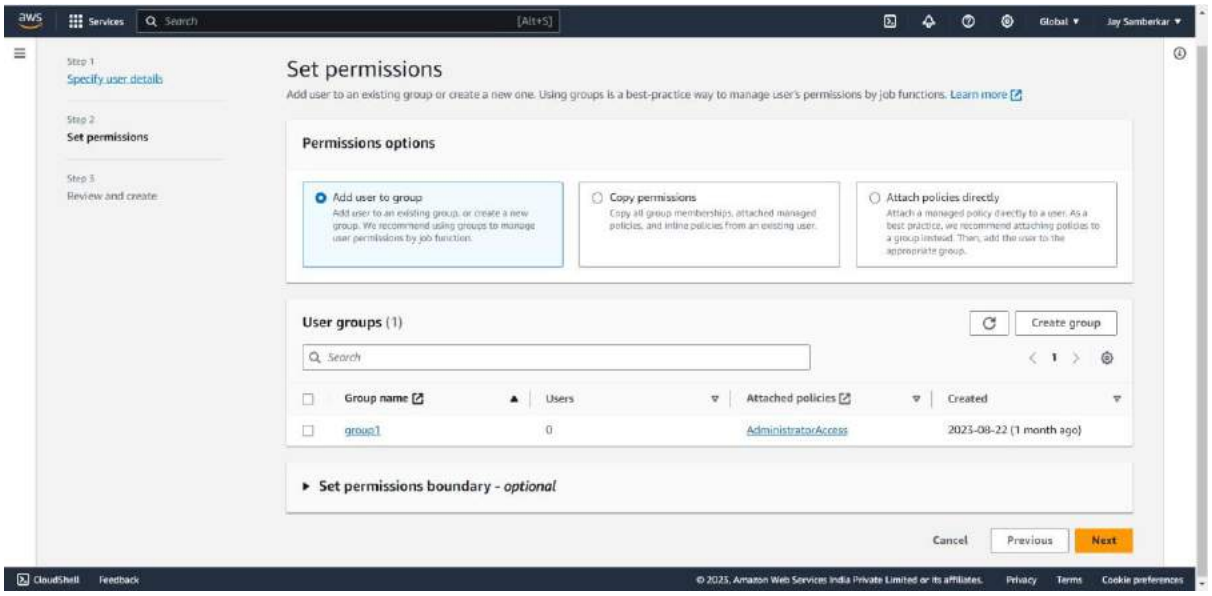
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

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

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

© 2023, Amazon Web Services India Private Limited or its affiliates.

3. Enter a new username and click on next
4. Click on create group



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

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

Permissions options

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

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

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

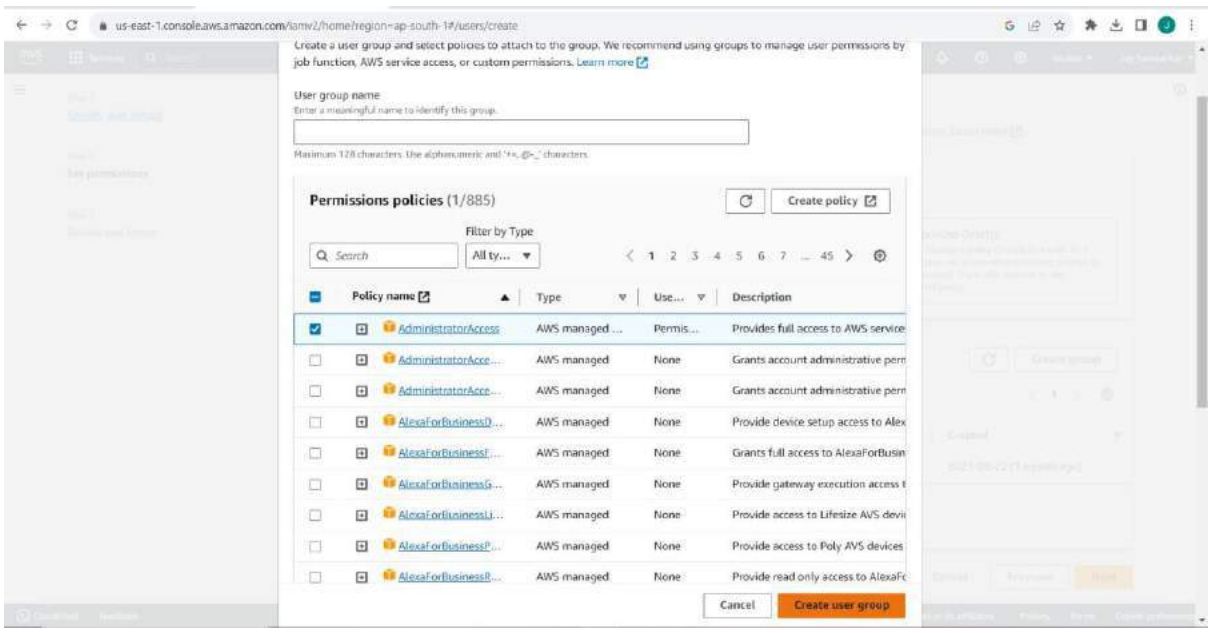
User groups (1)

	Group name	Users	Attached policies	Created
☐	group1	0	AdministratorAccess	2023-08-22 (1 month ago)

▸ Set permissions boundary - *optional*

© 2023, Amazon Web Services India Private Limited or its affiliates.

5. Select the first Policy will full access and enter a group name and create the group.



us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/create

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

User group name
Enter a meaningful name to identify this group.

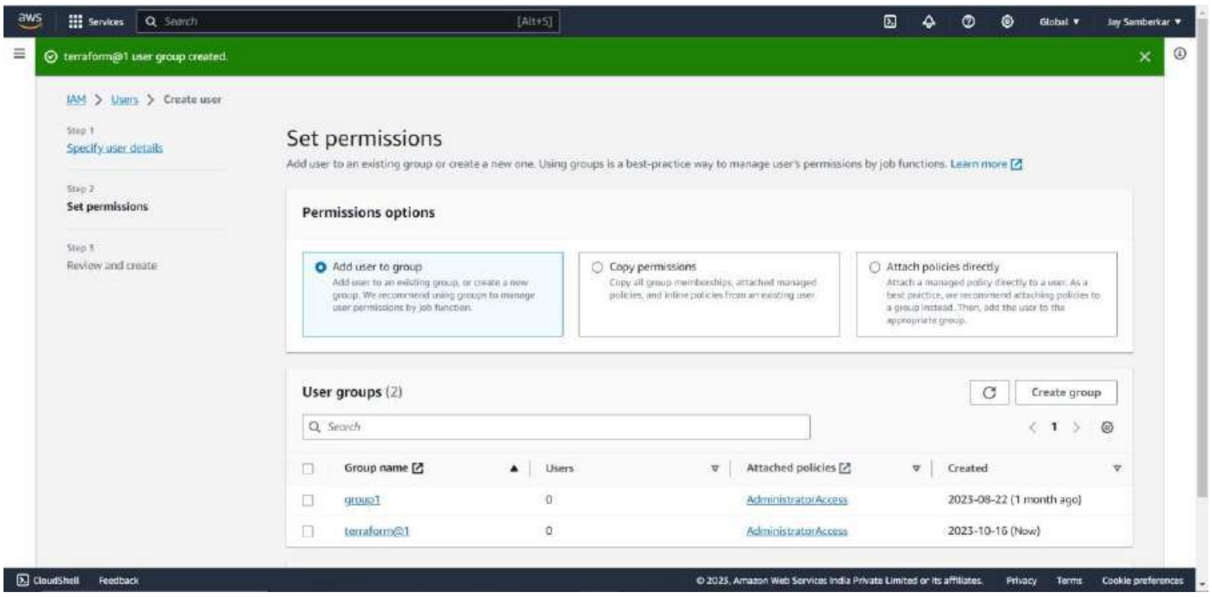
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions policies (1/885)

Filter by Type: All ty...

	Policy name	Type	Use...	Description
✓	AdministratorAccess	AWS managed ...	Permis...	Provides full access to AWS service
	AdministratorAcce...	AWS managed	None	Grants account administrative pern
	AdministratorAcce...	AWS managed	None	Grants account administrative pern
	AlexaForBusinessD...	AWS managed	None	Provide device setup access to Alex
	AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaForBusin
	AlexaForBusinessG...	AWS managed	None	Provide gateway execution access t
	AlexaForBusinessLi...	AWS managed	None	Provide access to Lifesize AVS devi
	AlexaForBusinessP...	AWS managed	None	Provide access to Poly AVS devices
	AlexaForBusinessR...	AWS managed	None	Provide read only access to AlexaFc

6. Then click on next, select the policy and create user.



terraform@1 user group created.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

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

Permissions options

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

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

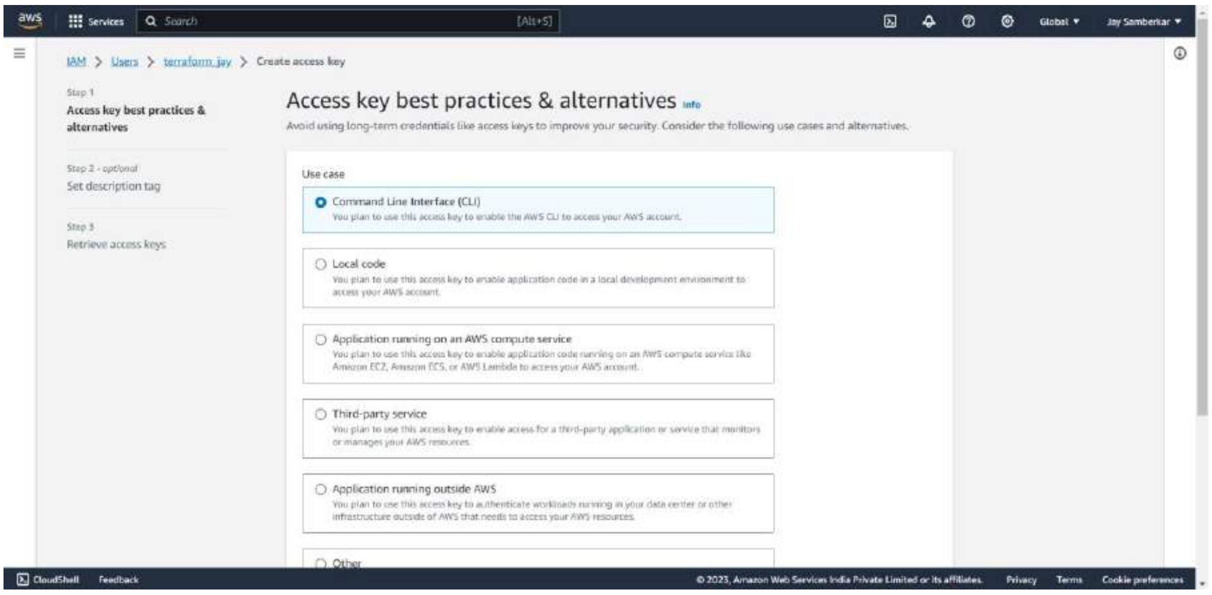
Attach policies directly: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (2)

	Group name	Users	Attached policies	Created
	group1	0	AdministratorAccess	2023-08-22 (1 month ago)
	terraform@1	0	AdministratorAccess	2023-10-16 (Now)

© 2023, Amazon Web Services India Private Limited or its affiliates.

7. Then create an access key with CLI.



IAM > Users > terraform_jay > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Access key best practices & alternatives Info

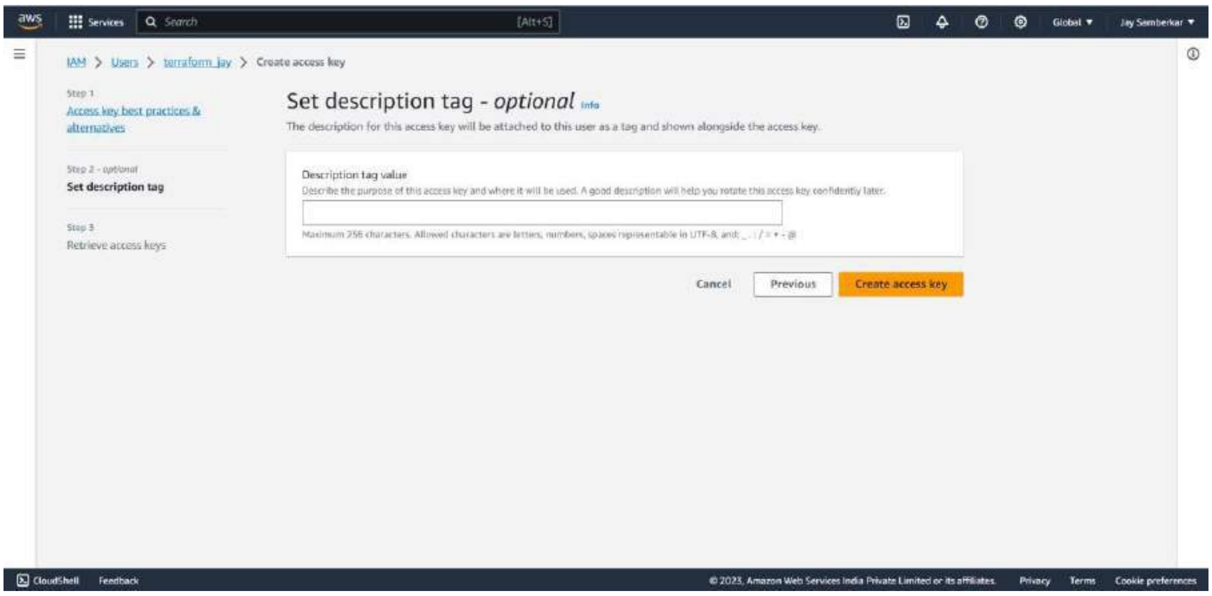
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- [✓ Command Line Interface (CLI)]
  You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code
  You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service
  You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service
  You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS
  You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other

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

8. Leave the description field empty and create a key. Download the access key in the .csv file.



IAM > Users > terraform_jay > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Set description tag - *optional* Info

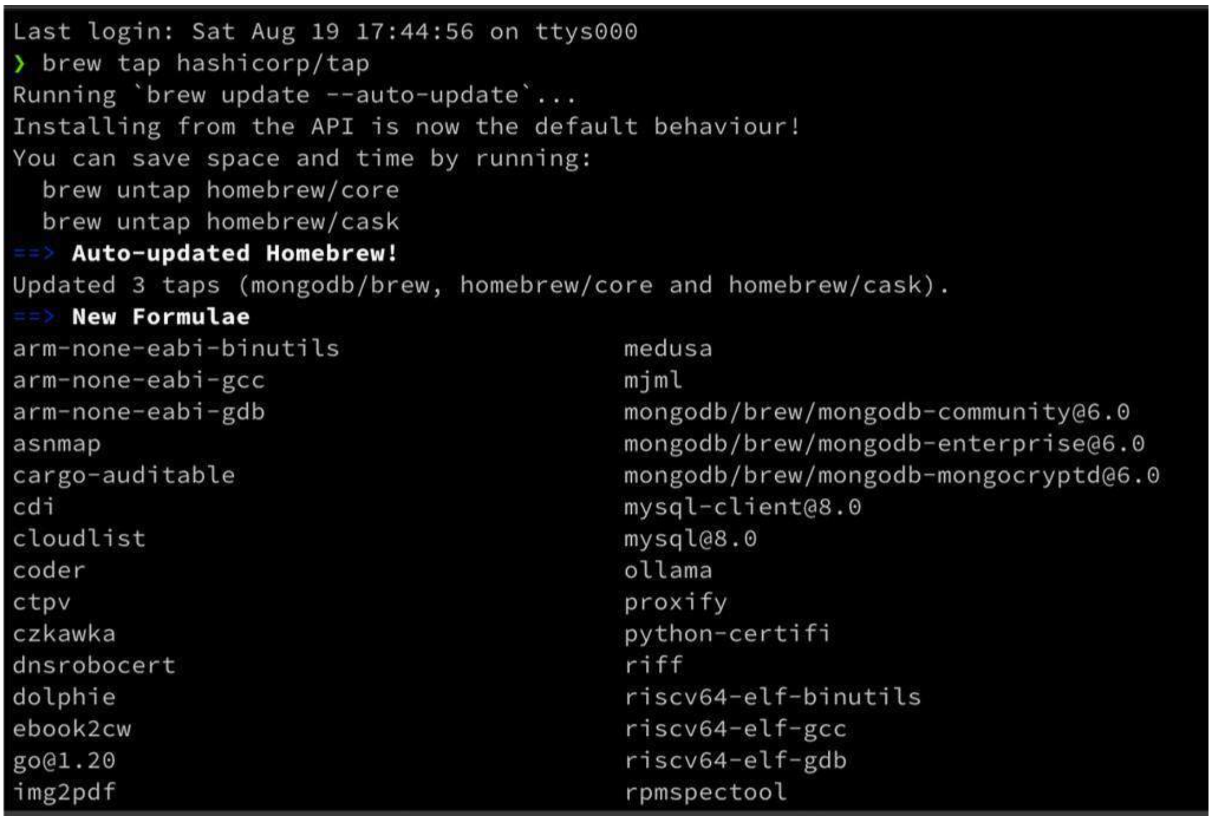
The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

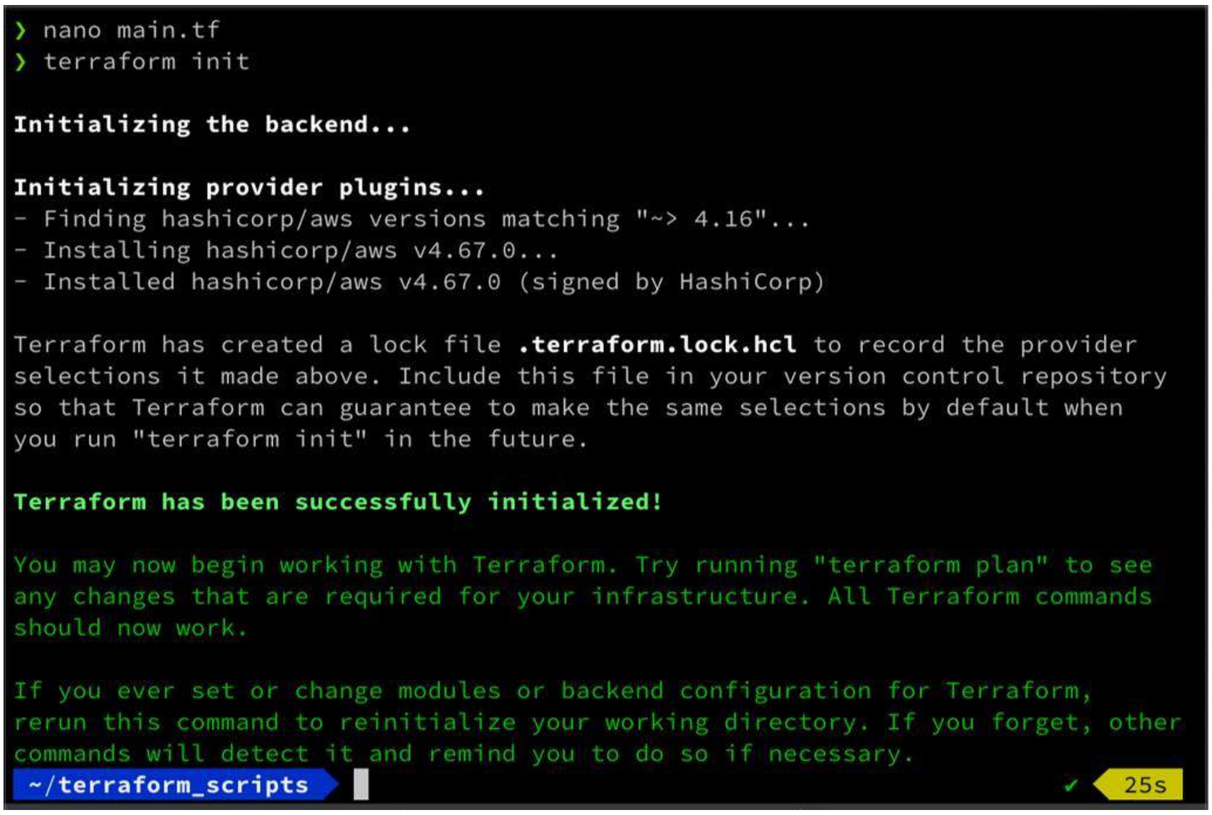
Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

9. Install terraform.



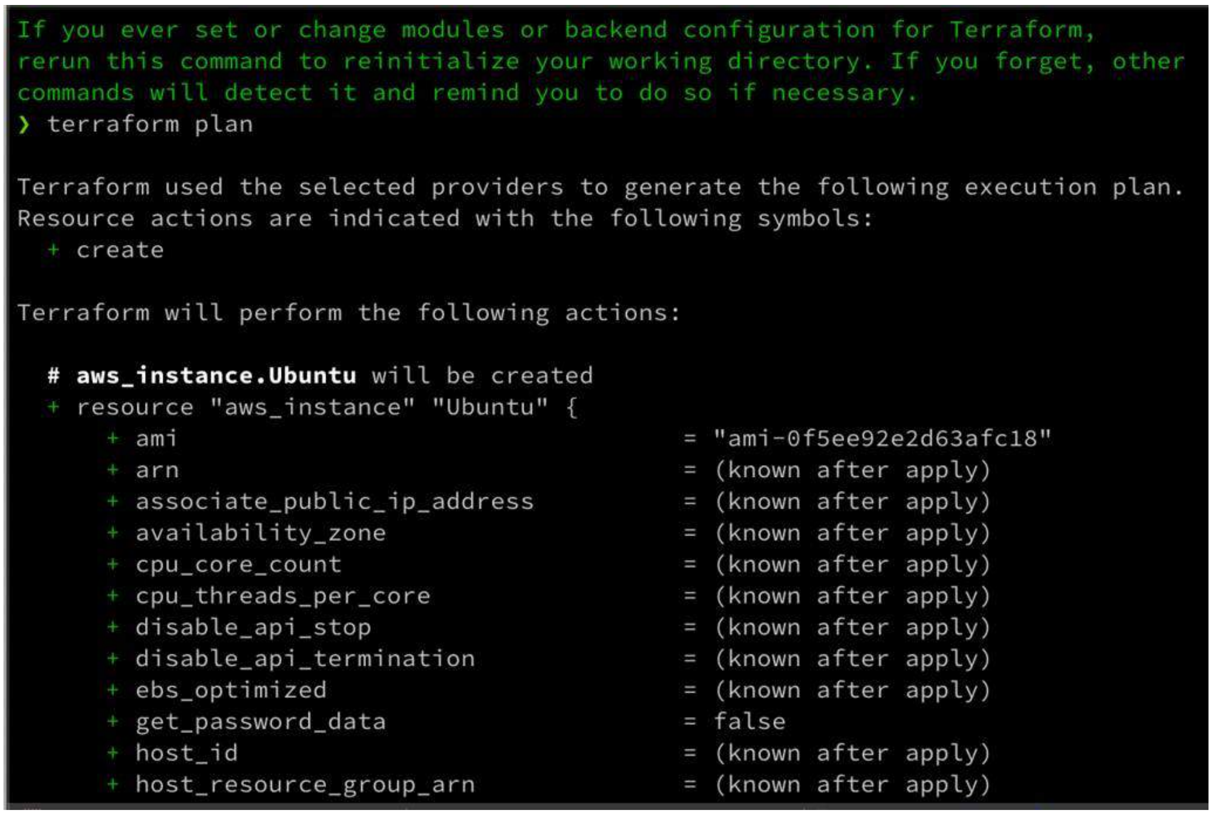
```
Last login: Sat Aug 19 17:44:56 on ttys000
❯ brew tap hashicorp/tap
Running `brew update --auto-update`...
Installing from the API is now the default behaviour!
You can save space and time by running:
  brew untap homebrew/core
  brew untap homebrew/cask
==> Auto-updated Homebrew!
Updated 3 taps (mongodb/brew, homebrew/core and homebrew/cask).
==> New Formulae
arm-none-eabi-binutils                  medusa
arm-none-eabi-gcc                       mjml
arm-none-eabi-gdb                       mongodb/brew/mongodb-community@6.0
asnmap                                  mongodb/brew/mongodb-enterprise@6.0
cargo-auditable                         mongodb/brew/mongodb-mongocryptd@6.0
cdi                                     mysql-client@8.0
cloudlist                               mysql@8.0
coder                                   ollama
ctpv                                    proxify
czkawka                                 python-certifi
dnsrobocert                             riff
dolphie                                 riscv64-elf-binutils
ebook2cw                                riscv64-elf-gcc
go@1.20                                 riscv64-elf-gdb
img2pdf                                 rpmspectool
```

10. Create the terraform config file main.tf with required configurations.



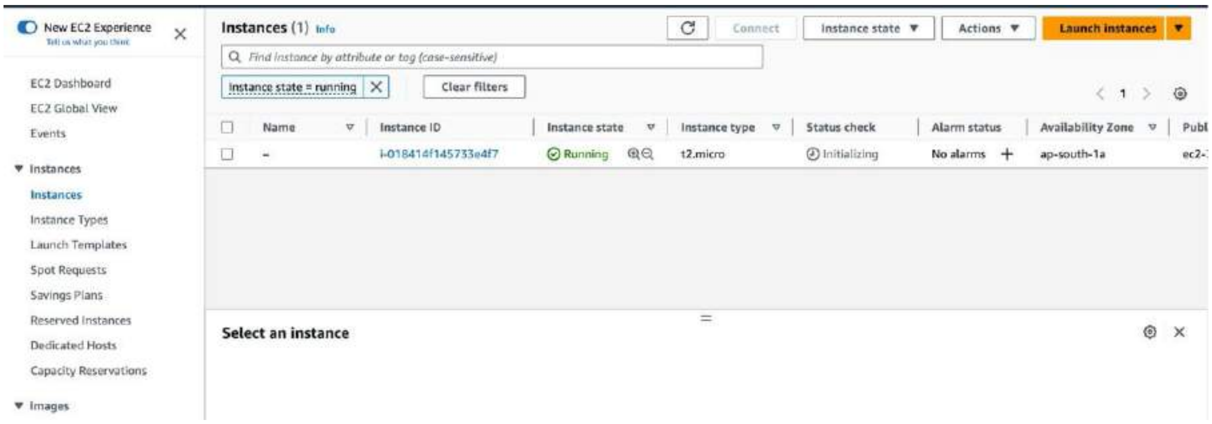
```
❯ nano main.tf
❯ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.67.0...
- Installed hashicorp/aws v4.67.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
 ~/terraform_scripts                                              ✔  25s
```

11. Run the command terraform plan and enter yes



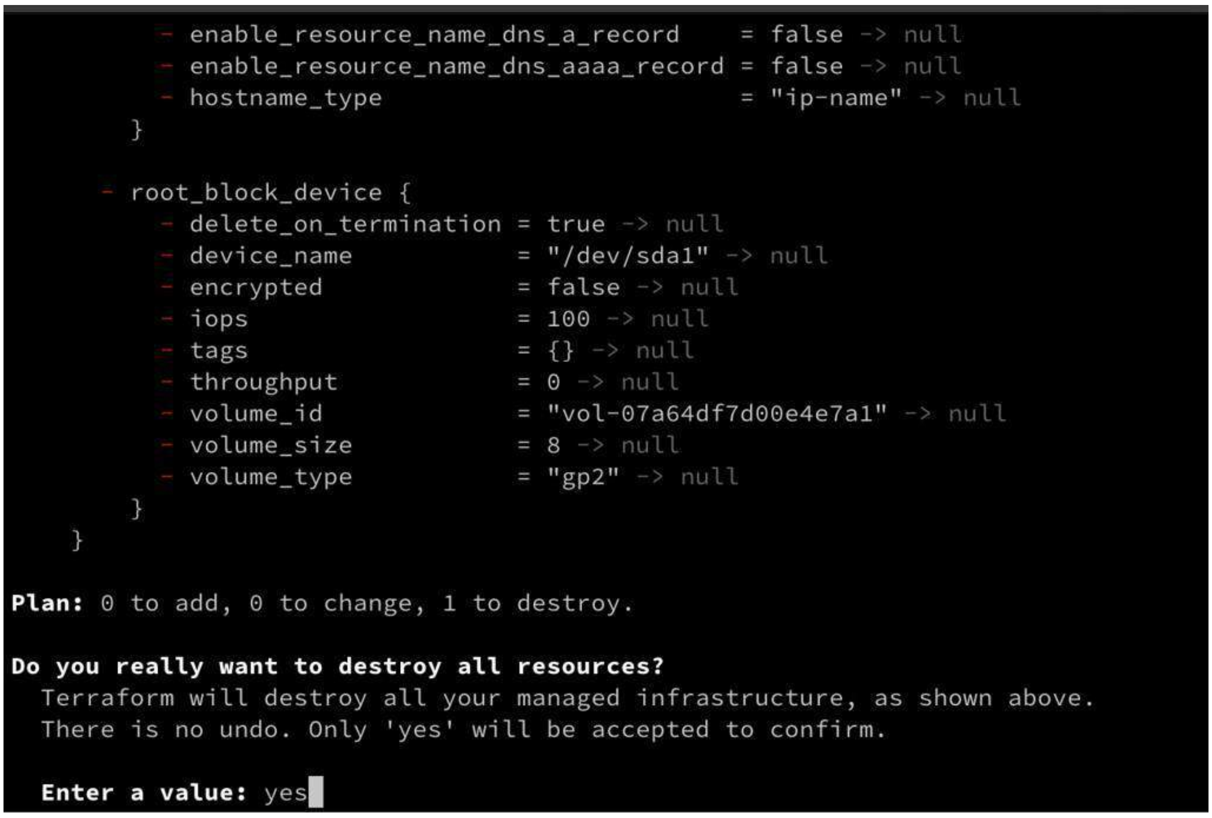
```
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
❯ terraform plan

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

Terraform will perform the following actions:

  # aws_instance.Ubuntu will be created
  + resource "aws_instance" "Ubuntu" {
      + ami                                  = "ami-0f5ee92e2d63afc18"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
```

12. Check EC2 instances on AWS, you'll find an instance running started by terraform.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
-	i-018414f145733e4f7	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-

13. Now run terraform destroy to terminate the instance.



```
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/sda1" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-07a64df7d00e4e7a1" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp2" -> null
        }
    }

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

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

  Enter a value: yes
```

CONCLUSION : We understood the workings of Terraform to create and manage EC2 instances in AWS which provides a highly automated and reproducible infrastructure-as-code solution. The ability to effortlessly spin up and tear down instances not only enhances operational agility but also ensures optimal resource utilization.

NAME : SOUMIL SALVI

ROLL NO : 104

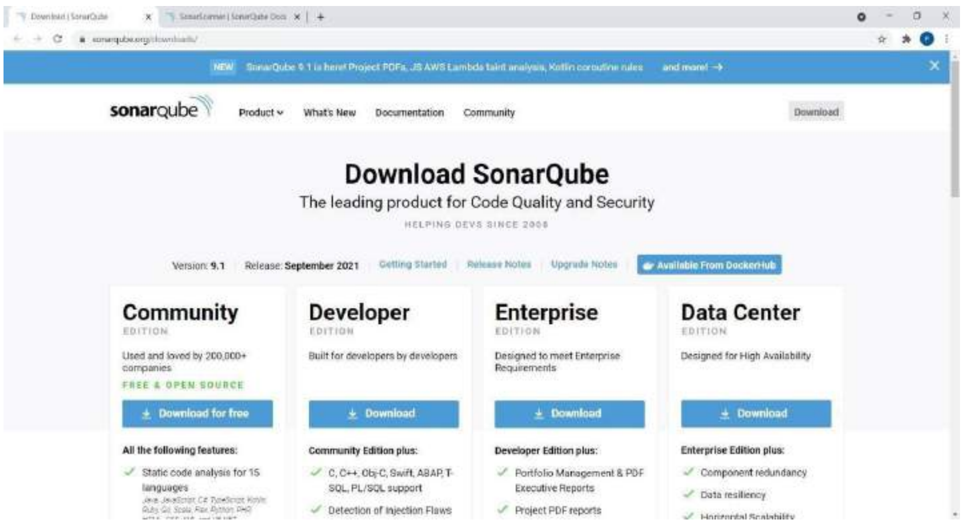
ASSIGNMENT 7

AIM : To perform static analysis on python programs using SonarCube SAST processes

LO MAPPED : LO4

THEORY :

Download SonarQube and Sonar Scanner



NEW SonarQube 9.1 is here! Project PDFs, JS AWS Lambda taint analysis, Kotlin coroutine rules and more! →

sonarqube Product What's New Documentation Community Download

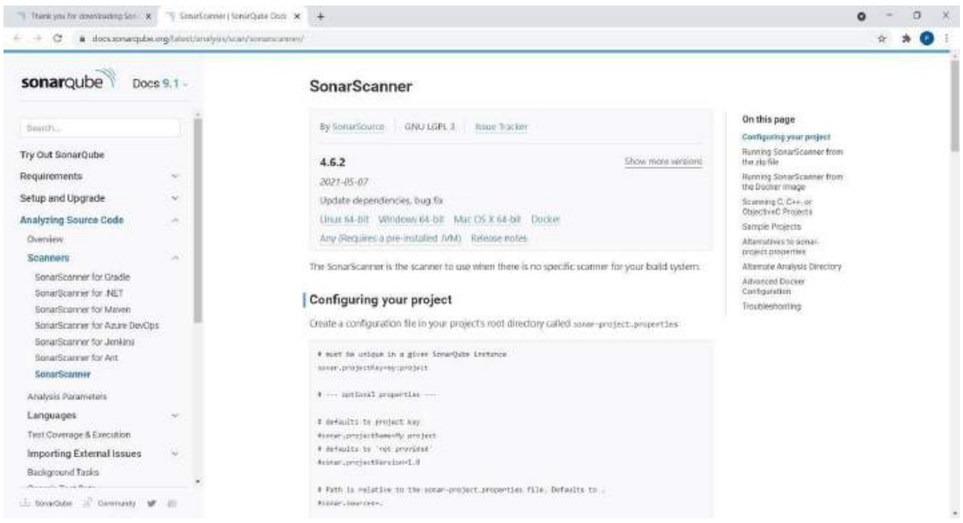
Download SonarQube

The leading product for Code Quality and Security

HELPING DEVS SINCE 2008

Version: 9.1 | Release: September 2021 | Getting Started | Release Notes | Upgrade Notes | Available From DockerHub

Community EDITION	Developer EDITION	Enterprise EDITION	Data Center EDITION
Used and loved by 200,000+ companies FREE & OPEN SOURCE	Built for developers by developers	Designed to meet Enterprise Requirements	Designed for High Availability
Download for free	Download	Download	Download
All the following features:	Community Edition plus:	Developer Edition plus:	Enterprise Edition plus:
✓ Static code analysis for 15 languages	✓ C, C++, Obj-C, Swift, ABAP, T-SQL, PL/SQL support	✓ Portfolio Management & PDF Executive Reports	✓ Component redundancy
	✓ Detection of Injection Flaws	✓ Project PDF reports	✓ Data resiliency
			✓ Horizontal Scalability



sonarqube Docs 9.1

Try Out SonarQube
Requirements
Setup and Upgrade
Analyzing Source Code
Overview
Scanners
SonarScanner for Gradle
SonarScanner for .NET
SonarScanner for Maven
SonarScanner for Azure DevOps
SonarScanner for Jenkins
SonarScanner for Ant
SonarScanner
Analysis Parameters
Languages
Test Coverage & Execution
Importing External Issues
Background Tasks

SonarScanner

By SonarSource | GNU LGPL 3 | Issue Tracker

4.6.2 Show more versions

2021-05-07

Update dependencies, bug fix

Linux 64-bit Windows 64-bit Mac OS X 64-bit Docker

Any (Requires a pre-installed JVM) Release notes

The SonarScanner is the scanner to use when there is no specific scanner for your build system.

Configuring your project

Create a configuration file in your projects root directory called sonar-project.properties

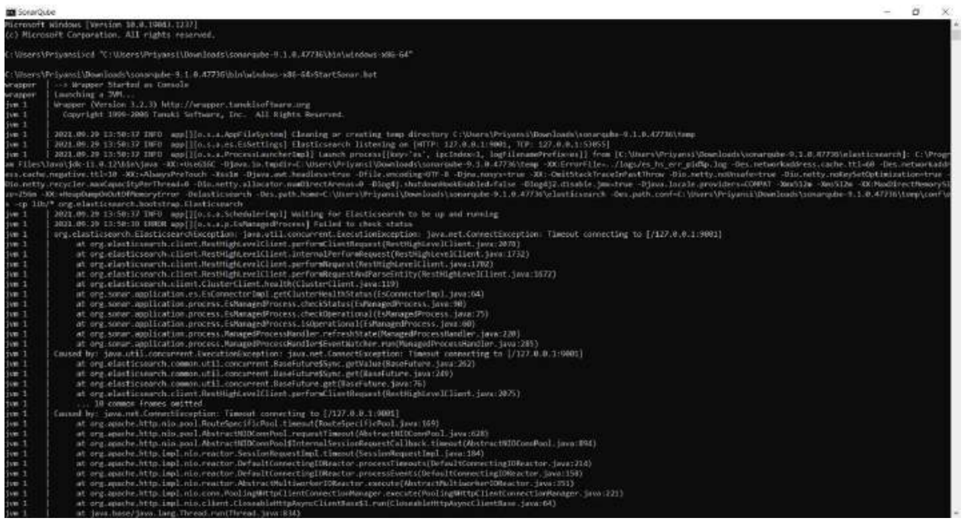
```
# must be unique in a given SonarQube instance
sonar.projectKey=my:project

# --- optional properties ---

# defaults to project key
#sonar.projectName=My project
# defaults to 'not provided'
#sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Defaults to .
#sonar.sources=.
```

On this page
Configuring your project
Running SonarScanner from the zip file
Running SonarScanner from the Docker image
Scanning C, C++, or ObjectiveC Projects
Sample Projects
Alternatives to sonar-project.properties
Alternate Analysis Directory
Advanced Docker Configuration
Troubleshooting

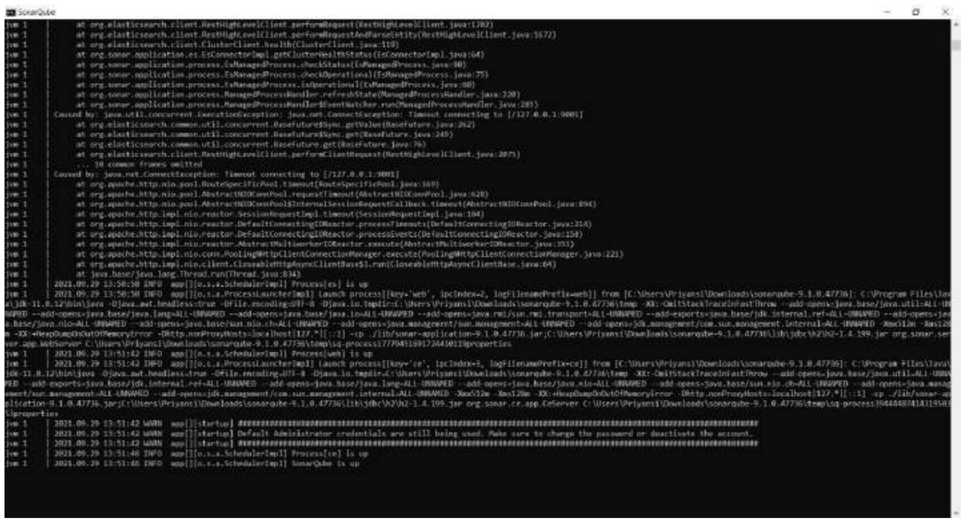
After downloading, set Environment Variables. Add “sonarqube-9.1.0.47736\bin” to Path.

Open command prompt. Run commands:

- cd “sonarqube-9.1.0.47736\bin\windows -x86 -64”
- StartSonar.bat



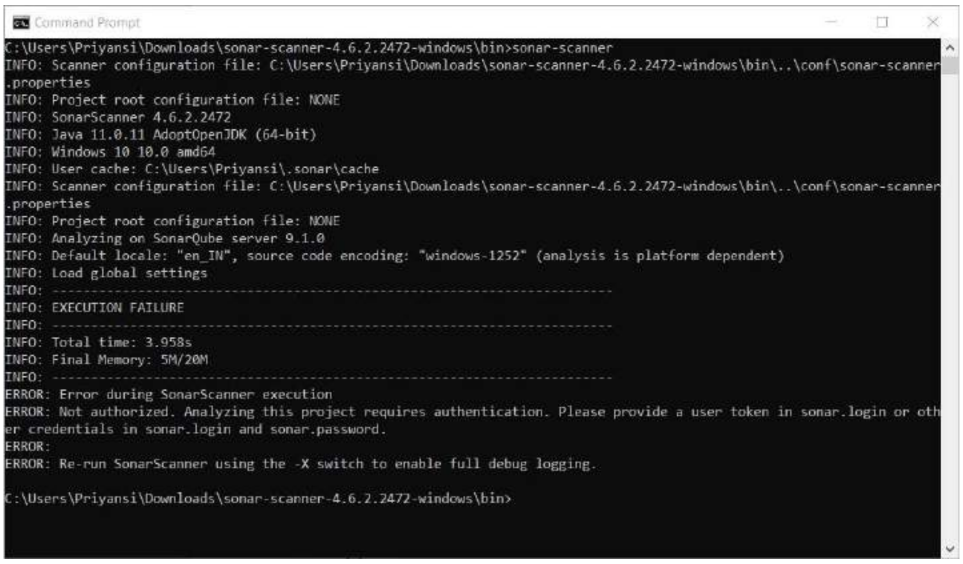
```
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Priyansi>cd "C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\bin\windows-x86-64"

C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\bin\windows-x86-64>StartSonar.bat
wrapper  | --> Wrapper Started as Console
wrapper  | Launching a JVM...
jvm 1    | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1    |   Copyright 1999-2006 Tanuki Software, Inc.  All Rights Reserved.
jvm 1    |
jvm 1    | 2021.09.29 13:50:37 INFO  app[][o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp
jvm 1    | 2021.09.29 13:50:37 INFO  app[][o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:53055]
jvm 1    | 2021.09.29 13:50:37 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenamePrefix=es]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\elasticsearch]: C:\Progr
am Files\Java\jdk-11.0.12\bin\java -XX:+UseG1GC -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:ErrorFile=../logs/es_hs_err_pid%p.log -Des.networkaddress.cache.ttl=60 -Des.networkadd
ress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -
Dio.netty.recycler.maxCapacityPerThread=0 -Dio.netty.allocator.numDirectArenas=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.locale.providers=COMPAT -Xmx512m -Xms512m -XX:MaxDirectMemorySi
ze=256m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\elasticsearch -Des.path.conf=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\conf\e
s -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1    | 2021.09.29 13:50:37 INFO  app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1    | 2021.09.29 13:50:38 ERROR app[][o.s.a.p.EsManagedProcess] Failed to check status
jvm 1    | org.elasticsearch.ElasticsearchException: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2078)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.internalPerformRequest(RestHighLevelClient.java:1732)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1702)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1672)
jvm 1    |         at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:119)
jvm 1    |         at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.checkStatus(EsManagedProcess.java:90)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.checkOperational(EsManagedProcess.java:75)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:60)
jvm 1    |         at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1    |         at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1    | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1    |         ... 10 common frames omitted
jvm 1    | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |         at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
jvm 1    |         at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimeout(AbstractNIOConnPool.java:628)
jvm 1    |         at org.apache.http.nio.pool.AbstractNIOConnPool$InternalSessionRequestCallback.timeout(AbstractNIOConnPool.java:894)
jvm 1    |         at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
jvm 1    |         at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1    |         at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1    |         at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1    |         at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
jvm 1    |         at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1    |         at java.base/java.lang.Thread.run(Thread.java:834)
```



```
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1702)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1672)
jvm 1    |         at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:119)
jvm 1    |         at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.checkStatus(EsManagedProcess.java:90)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.checkOperational(EsManagedProcess.java:75)
jvm 1    |         at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:60)
jvm 1    |         at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1    |         at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1    | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1    |         at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1    |         at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1    |         ... 10 common frames omitted
jvm 1    | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |         at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
jvm 1    |         at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimeout(AbstractNIOConnPool.java:628)
jvm 1    |         at org.apache.http.nio.pool.AbstractNIOConnPool$InternalSessionRequestCallback.timeout(AbstractNIOConnPool.java:894)
jvm 1    |         at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
jvm 1    |         at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1    |         at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1    |         at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1    |         at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
jvm 1    |         at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1    |         at java.base/java.lang.Thread.run(Thread.java:834)
jvm 1    | 2021.09.29 13:50:50 INFO  app[][o.s.a.SchedulerImpl] Process[es] is up
jvm 1    | 2021.09.29 13:50:50 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Jav
a\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UN
NAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=jav
a.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128
m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ser
ver.app.WebServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process17770451691724410119properties
jvm 1    | 2021.09.29 13:51:42 INFO  app[][o.s.a.SchedulerImpl] Process[web] is up
jvm 1    | 2021.09.29 13:51:42 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\
jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNA
MED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.manag
ement/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-ap
plication-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process39444487414319583
53properties
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] ##############################################################################
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] ##############################################################################
jvm 1    | 2021.09.29 13:51:46 INFO  app[][o.s.a.SchedulerImpl] Process[ce] is up
jvm 1    | 2021.09.29 13:51:46 INFO  app[][o.s.a.SchedulerImpl] SonarQube is up
```

Open another command prompt. Run command:

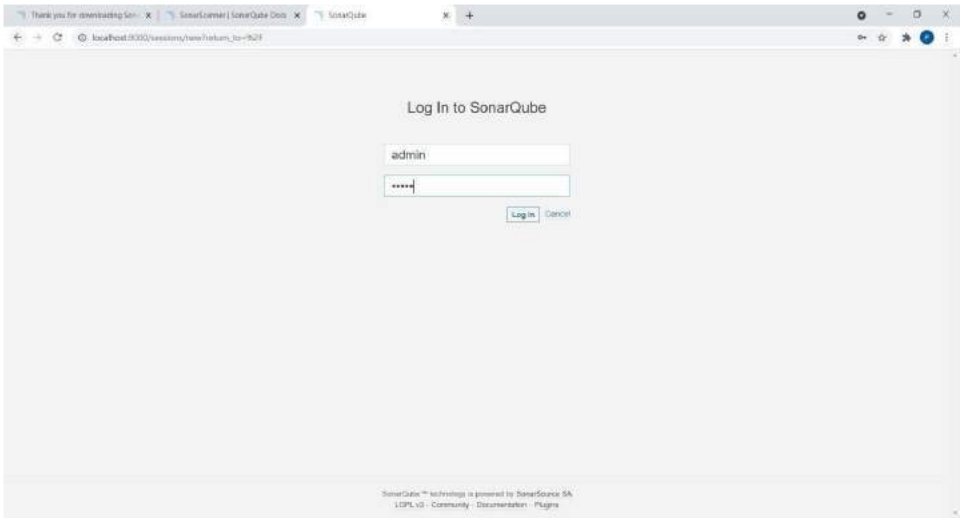
- cd “sonar-scanner-4.6.2.2472-windows\bin”
- sonar-scanner

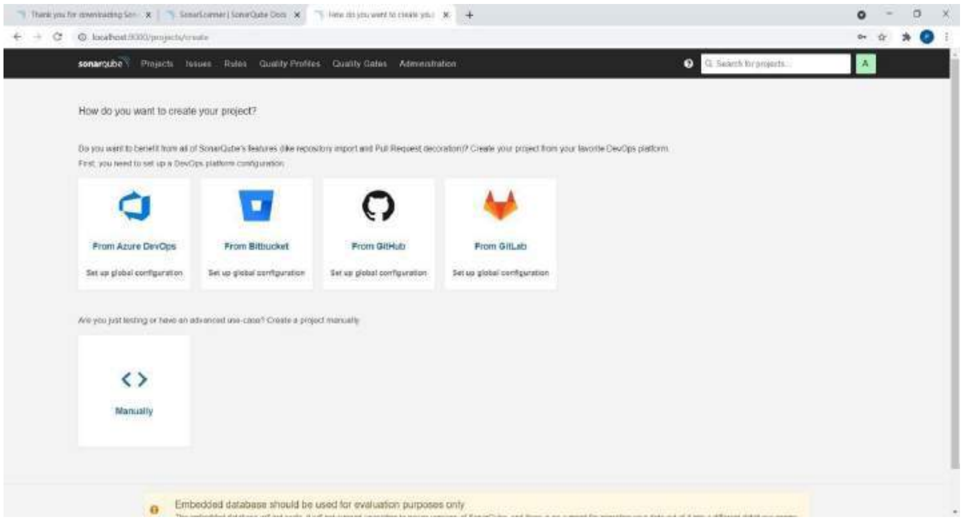


```
C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>sonar-scanner
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin\..\conf\sonar-scanner
.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 AdoptOpenJDK (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\Priyansi\.sonar\cache
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin\..\conf\sonar-scanner
.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 9.1.0
INFO: Default locale: "en_IN", source code encoding: "windows-1252" (analysis is platform dependent)
INFO: Load global settings
INFO: ------------------------------------------------------------------------
INFO: EXECUTION FAILURE
INFO: ------------------------------------------------------------------------
INFO: Total time: 3.958s
INFO: Final Memory: 5M/20M
INFO: ------------------------------------------------------------------------
ERROR: Error during SonarScanner execution
ERROR: Not authorized. Analyzing this project requires authentication. Please provide a user token in sonar.login or oth
er credentials in sonar.login and sonar.password.
ERROR:
ERROR: Re-run SonarScanner using the -X switch to enable full debug logging.

C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>
```

Server up and running on localhost:9000

Login using credentials as User: admin and Password: admin and Set a new password





How do you want to create your project?

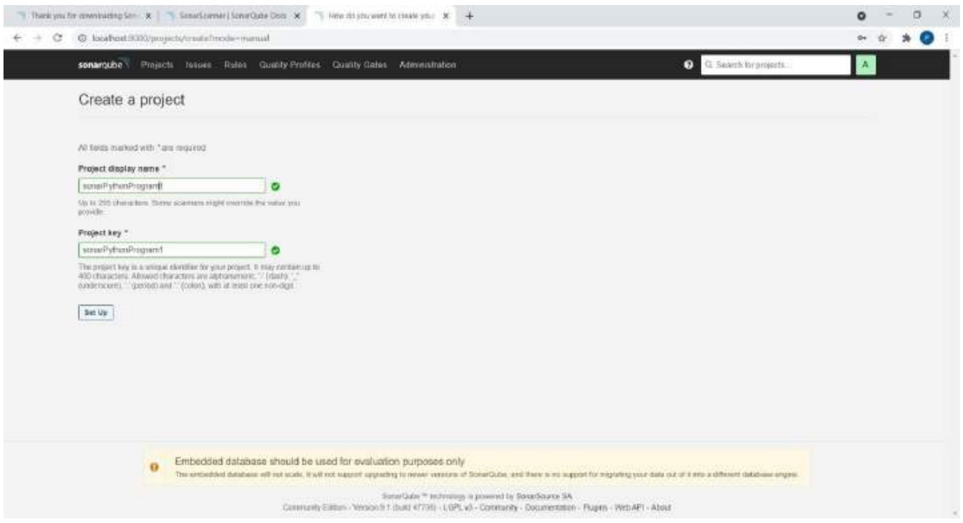
Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.
First, you need to set up a DevOps platform configuration.

From Azure DevOps
Set up global configuration
From Bitbucket
Set up global configuration
From GitHub
Set up global configuration
From GitLab
Set up global configuration

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

Embedded database should be used for evaluation purposes only

Click on Create a project Manually



Create a project

All fields marked with * are required

Project display name *

sonarPythonProgram1

Up to 255 characters. Some scanners might override the value you provide.

Project key *

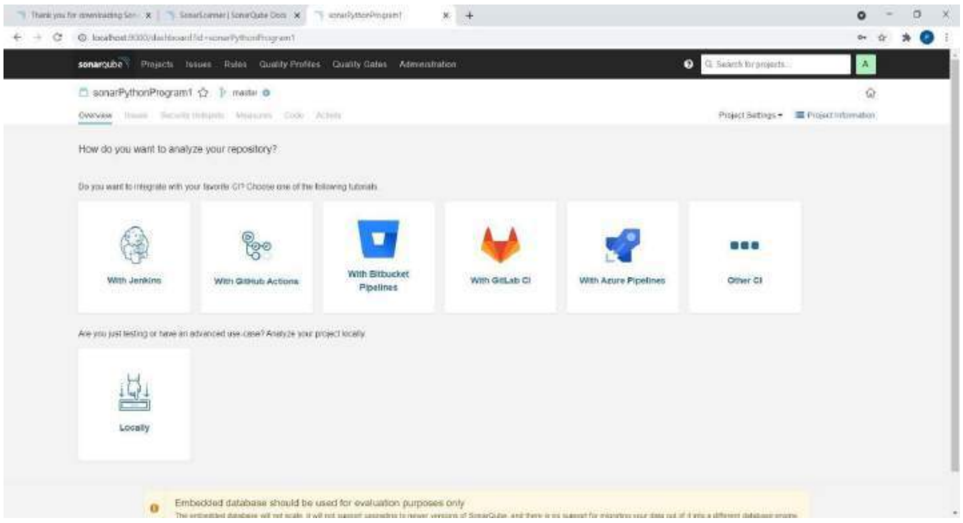
sonarPythonProgram1

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Set Up

Embedded database should be used for evaluation purposes only

Give any Project display name.



sonarPythonProgram1 master

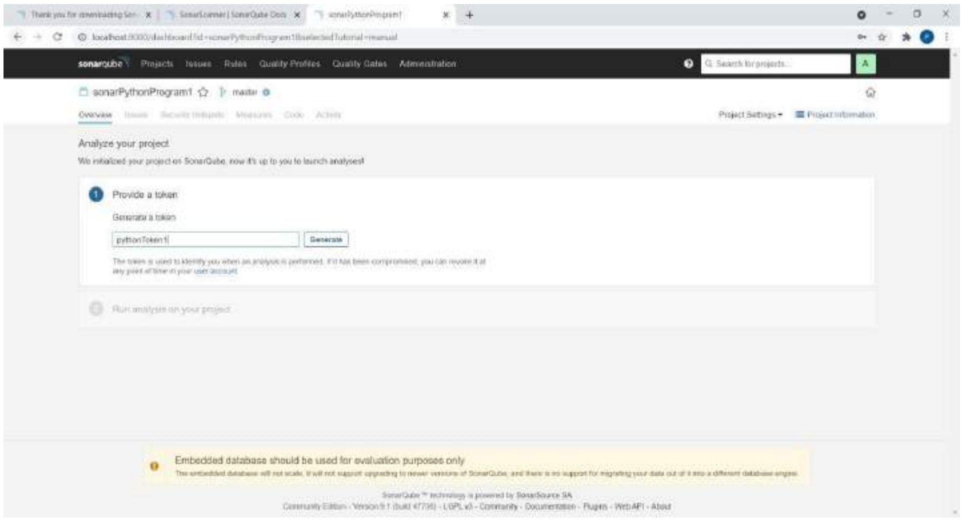
How do you want to analyze your repository?

Do you want to integrate with your favorite CI? Choose one of the following tutorials.

Are you just testing or have an advanced use-case? Analyze your project locally.

Embedded database should be used for evaluation purposes only

Click on Locally



sonarPythonProgram1 master

Analyze your project

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

1 Provide a token

Generate a token

pythonToken1

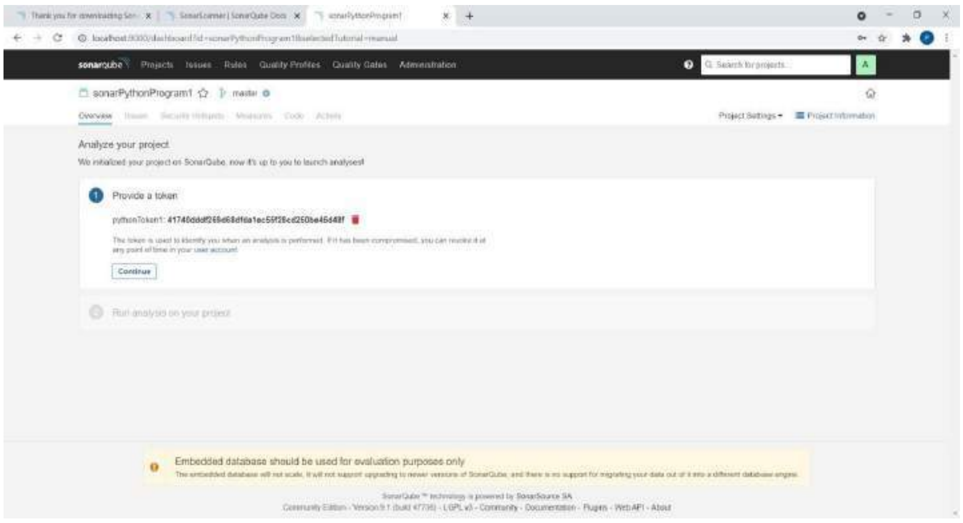
Generate

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

2 Run analysis on your project

Embedded database should be used for evaluation purposes only

Give any name to token and click on Generate.



Analyze your project

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

1 Provide a token

pythonToken1: 41740dddf269d68dfda1ec55f28cd250be45d48f

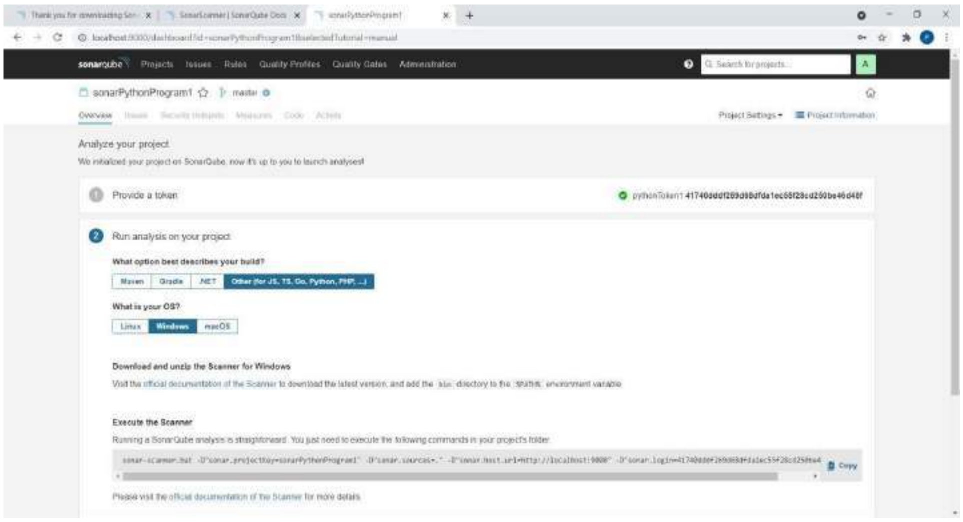
The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

Continue

2 Run analysis on your project

Embedded database should be used for evaluation purposes only

Click on Continue.



Analyze your project

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

1 Provide a token — pythonToken1: 41740dddf269d68dfda1ec55f28cd250be46d48f

2 Run analysis on your project

What option best describes your build?

Maven | Gradle | .NET | [✓ Other (for JS, TS, Go, Python, PHP, ...)]

What is your OS?

Linux | [✓ Windows] | macOS

Download and unzip the Scanner for Windows

Visit the official documentation of the Scanner to download the latest version, and add the bin directory to the %PATH% environment variable

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=41740ddf269d68dfda1ec55f28cd250be4
```

Please visit the official documentation of the Scanner for more details

Save a Python program in a folder.

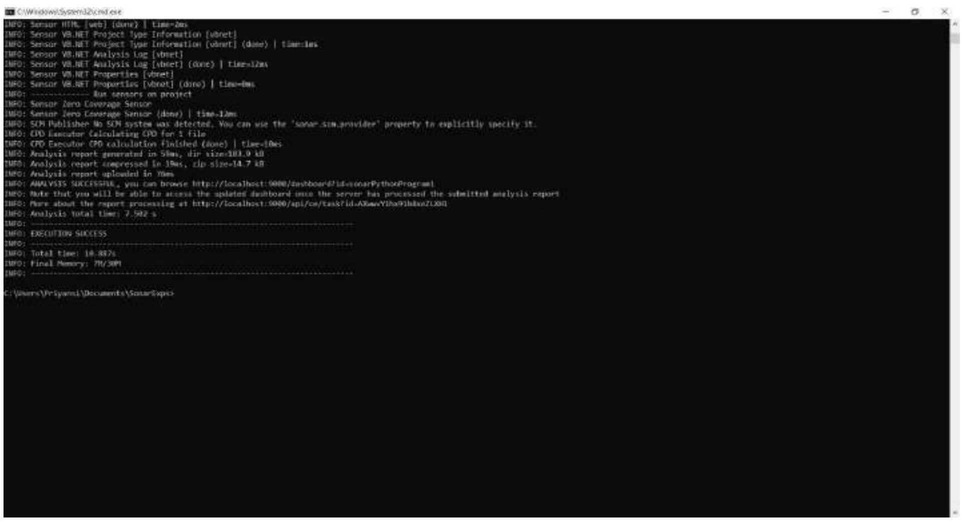
```
class Solution(object):

def romanToInt(self, s):

roman =

{'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000,'IV':4,'IX':9,'XL':40,'XC':90,'CD':40
0,'CM':900} i = 0 num = " " while i < len(s):

if i+1<len(s) and s[i:i+2] in roman:

num+=roman[s[i:i+2]] i+=2

else: #print(i)
```

num+=roman[s[i]] i+=1

return num

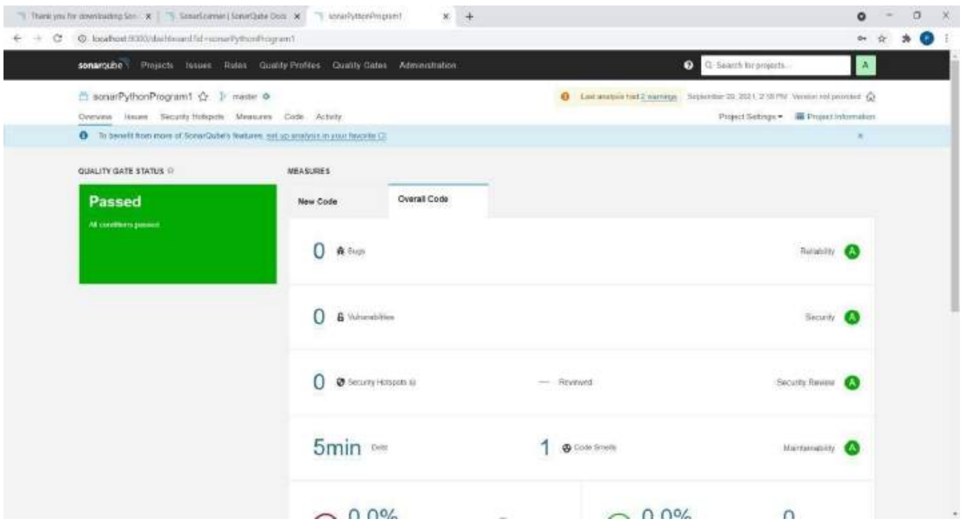
ob1 = Solution() print(ob1.romanToInt("III"))   print(ob1.romanToInt("CDXLIII")

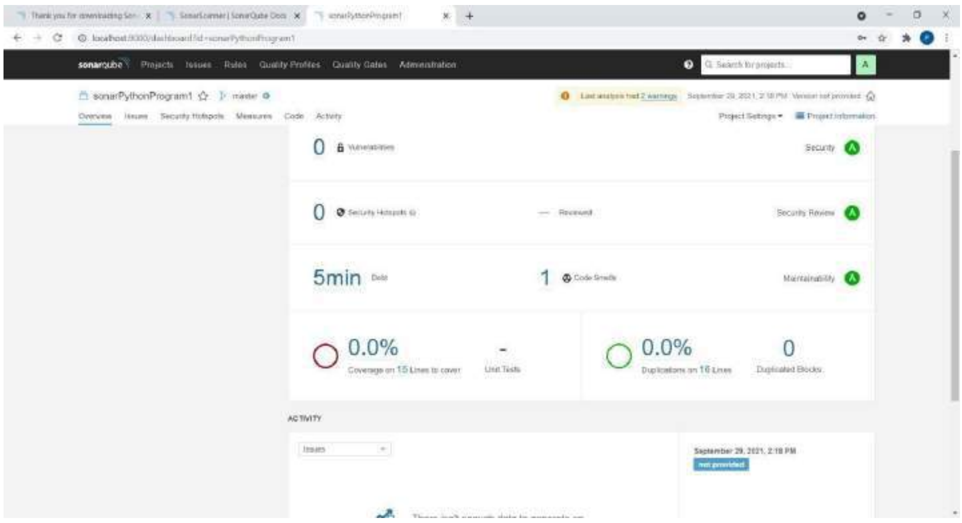
Open command prompt in this folder and Run program using copied command.
sonar-scanner.bat -D"sonar.projectKey=<YourDisplayName>"
D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" –

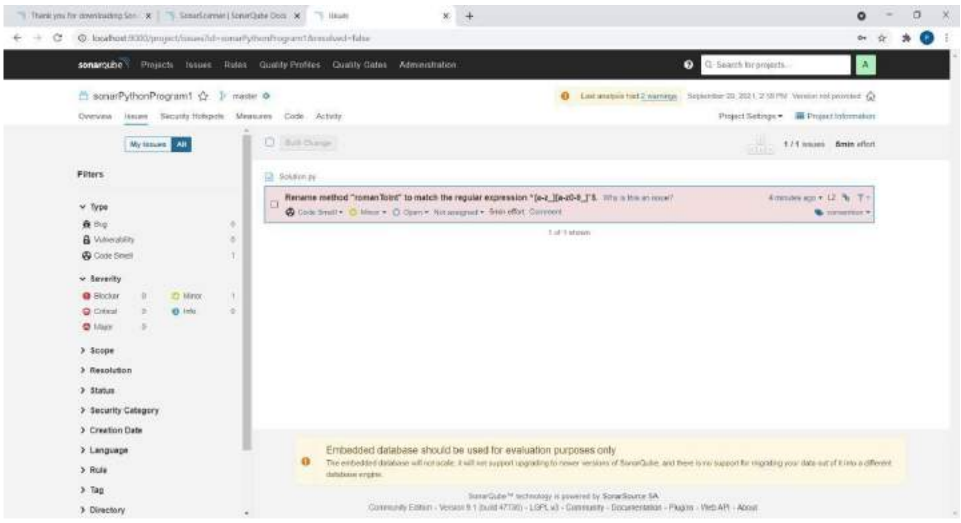


```
C:\Windows\System32\cmd.exe
INFO: Sensor HTML [web] (done) | time=2ms
INFO: Sensor VB.NET Project Type Information [vbnet]
INFO: Sensor VB.NET Project Type Information [vbnet] (done) | time=1ms
INFO: Sensor VB.NET Analysis Log [vbnet]
INFO: Sensor VB.NET Analysis Log [vbnet] (done) | time=12ms
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: ------------- Run sensors on project
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=12ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor Calculating CPD for 1 file
INFO: CPD Executor CPD calculation finished (done) | time=10ms
INFO: Analysis report generated in 59ms, dir size=103.9 kB
INFO: Analysis report compressed in 19ms, zip size=14.7 kB
INFO: Analysis report uploaded in 70ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=sonarPythonProgram1
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AXwwY1ha91b8xnZLX0I
INFO: Analysis total time: 7.502 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 10.887s
INFO: Final Memory: 7M/30M
INFO: ------------------------------------------------------------------------

C:\Users\Priyansi\Documents\SonarExps>
```

Given below is the inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

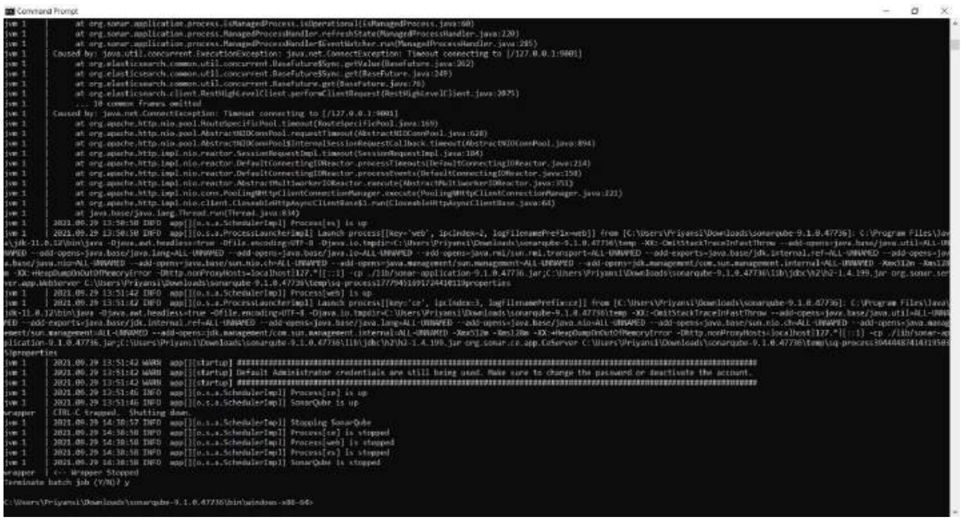






localhost:9000/project/issues?id=sonarPythonProgram1&resolved=false
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
sonarPythonProgram1
master
Last analysis had 2 warnings
My Issues
All
Bulk Change
1 / 1 issues
5min effort
Filters
Type
Bug 0
Vulnerability 0
Code Smell 1
Severity
Blocker 0
Minor 1
Critical 0
Info 0
Major 0
Scope
Resolution
Status
Security Category
Creation Date
Language
Rule
Tag
Directory
Solution.py
Rename method "romanToInt" to match the regular expression ^[a-z_][a-z0-9_]*$.
Code Smell
Minor
Open
Not assigned
5min effort
Comment
convention
1 of 1 shown
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.
SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.1 (build 47736) - LGPL v3 - Community - Documentation - Plugins - Web API - About

Press “Ctrl + C” to stop the server.



```
Command Prompt
jvm 1    |      at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:60)
jvm 1    |      at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1    |      at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1    | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
jvm 1    |      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1    |      at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1    |      at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1    |      ... 10 common frames omitted
jvm 1    | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1    |      at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
jvm 1    |      at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimeout(AbstractNIOConnPool.java:628)
jvm 1    |      at org.apache.http.nio.pool.AbstractNIOConnPool$InternalSessionRequestCallback.timeout(AbstractNIOConnPool.java:894)
jvm 1    |      at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
jvm 1    |      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1    |      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1    |      at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1    |      at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
jvm 1    |      at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1    |      at java.base/java.lang.Thread.run(Thread.java:834)
jvm 1    | 2021.09.29 13:50:58 INFO  app[][o.s.a.SchedulerImpl] Process[es] is up
jvm 1    | 2021.09.29 13:50:58 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Jav
a\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UN
NAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=jav
a.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128
m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ser
ver.app.WebServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process17779451691724410119properties
jvm 1    | 2021.09.29 13:51:42 INFO  app[][o.s.a.SchedulerImpl] Process[web] is up
jvm 1    | 2021.09.29 13:51:42 INFO  app[][o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\
jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNA
MED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.manag
ement/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-ap
plication-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process3944448741431958
5properties
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] ####################################################################################################
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
jvm 1    | 2021.09.29 13:51:42 WARN  app[][startup] ####################################################################################################
jvm 1    | 2021.09.29 13:51:46 INFO  app[][o.s.a.SchedulerImpl] Process[ce] is up
jvm 1    | 2021.09.29 13:51:46 INFO  app[][o.s.a.SchedulerImpl] SonarQube is up
wrapper  | CTRL-C trapped.  Shutting down.
jvm 1    | 2021.09.29 14:38:57 INFO  app[][o.s.a.SchedulerImpl] Stopping SonarQube
jvm 1    | 2021.09.29 14:38:58 INFO  app[][o.s.a.SchedulerImpl] Process[ce] is stopped
jvm 1    | 2021.09.29 14:38:58 INFO  app[][o.s.a.SchedulerImpl] Process[web] is stopped
jvm 1    | 2021.09.29 14:38:58 INFO  app[][o.s.a.SchedulerImpl] Process[es] is stopped
jvm 1    | 2021.09.29 14:38:58 INFO  app[][o.s.a.SchedulerImpl] SonarQube is stopped
wrapper  | <-- Wrapper Stopped
Terminate batch job (Y/N)? y

C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\bin\windows-x86-64>
```

CONCLUSION : In this assignment we implemented static analysis on python programs using SonarCube SAST processes.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 8

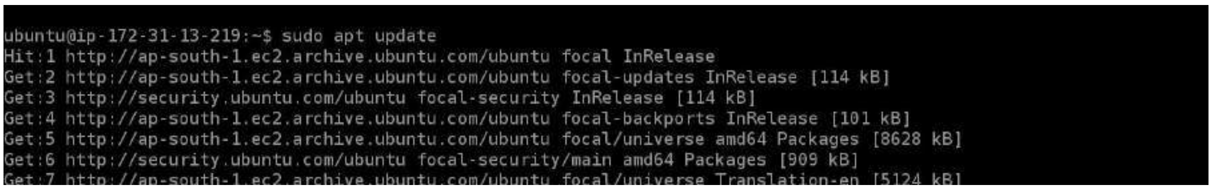
AIM : To understand continuous monitoring and Installation and configuration of Nagios on Linux Machine

LO MAPPED : LO1,LO5

THEORY :
Login to Aws.

Create Ec2 instances on Aws account of any linux os

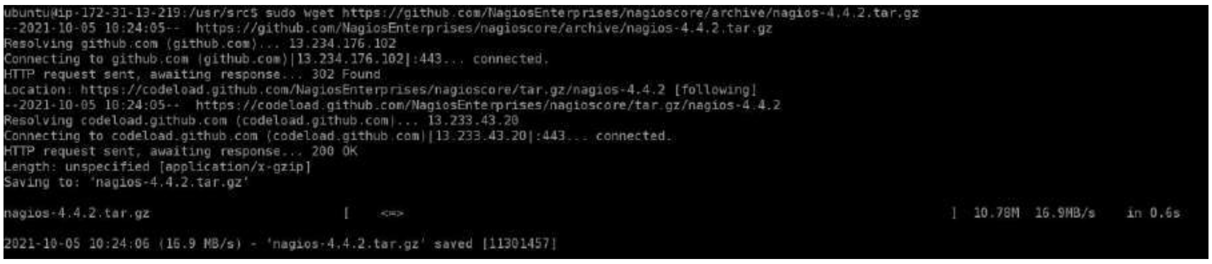
Then Run the following command in SS



```
ubuntu@ip-172-31-13-219:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [909 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
```

cd /usr/src/

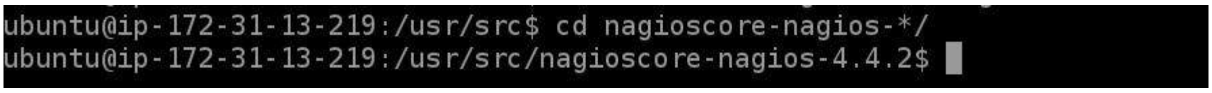
sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz



```
ubuntu@ip-172-31-13-219:/usr/src$ sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
--2021-10-05 10:24:05--  https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2 [following]
--2021-10-05 10:24:05--  https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2
Resolving codeload.github.com (codeload.github.com)... 13.233.43.20
Connecting to codeload.github.com (codeload.github.com)|13.233.43.20|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'nagios-4.4.2.tar.gz'

nagios-4.4.2.tar.gz                 [    <=>                                         ]  10.78M  16.9MB/s    in 0.6s

2021-10-05 10:24:06 (16.9 MB/s) - 'nagios-4.4.2.tar.gz' saved [11301457]
```

sudo tar zxf nagios-*.tar.gz cd

nagioscore-nagios-*/



```
ubuntu@ip-172-31-13-219:/usr/src$ cd nagioscore-nagios-*/
ubuntu@ip-172-31-13-219:/usr/src/nagioscore-nagios-4.4.2$ █
```

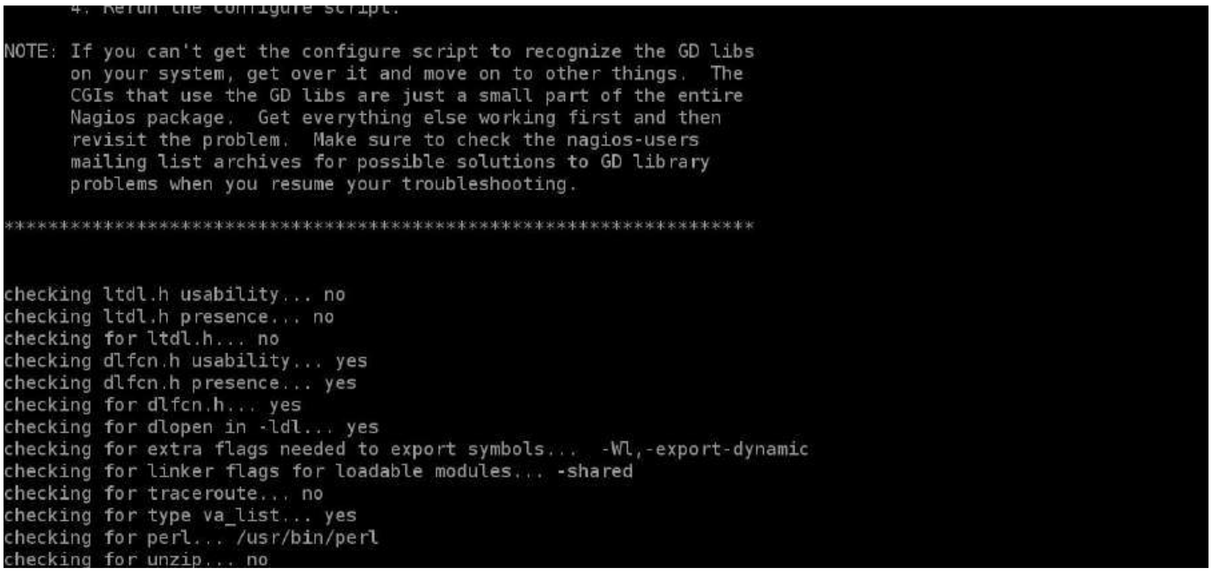
Now finally run the following command

sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled

if error comes install c compiler on the linux by

following this link

https://linuxize.com/post/how-to-install-gcc-compiler-on-ubuntu-18-04/ Finally-



```
     4. Rerun the configure script.

NOTE: If you can't get the configure script to recognize the GD libs
      on your system, get over it and move on to other things.  The
      CGIs that use the GD libs are just a small part of the entire
      Nagios package.  Get everything else working first and then
      revisit the problem.  Make sure to check the nagios-users
      mailing list archives for possible solutions to GD library
      problems when you resume your troubleshooting.

*********************************************************************


checking ltdl.h usability... no
checking ltdl.h presence... no
checking for ltdl.h... no
checking dlfcn.h usability... yes
checking dlfcn.h presence... yes
checking for dlfcn.h... yes
checking for dlopen in -ldl... yes
checking for extra flags needed to export symbols...  -Wl,-export-dynamic
checking for linker flags for loadable modules... -shared
checking for traceroute... no
checking for type va_list... yes
checking for perl... /usr/bin/perl
checking for unzip... no
```

Now let us install plugins by

sudo wget -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagiosplugins/archive/release-2.2.1.tar.g

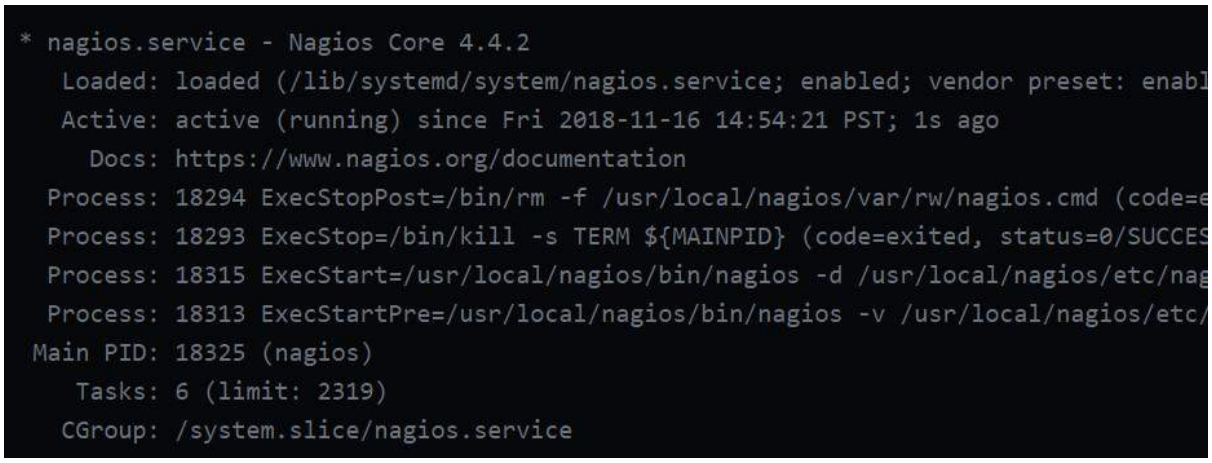
then

sudo tar zxf nagios-plugins.tar.gz then

cd nagios-plugins-release-2.2.1

finally start and then check status of nagi os sudo systemctl start nagios sudo

systemctl status nagios

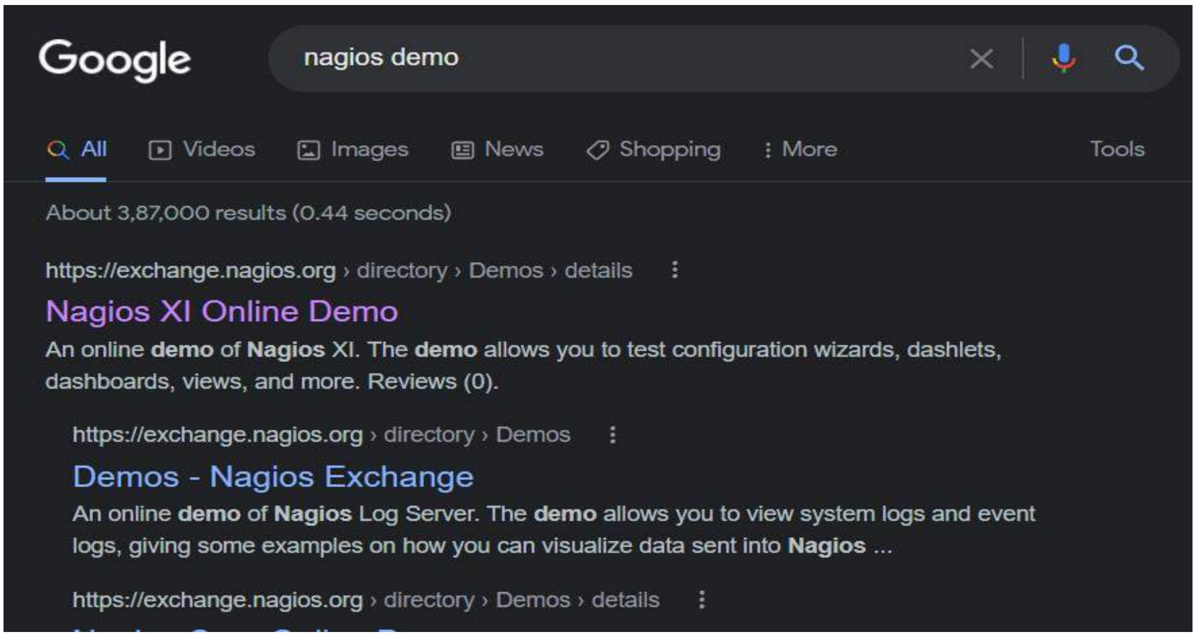


```
* nagios.service - Nagios Core 4.4.2
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabl
   Active: active (running) since Fri 2018-11-16 14:54:21 PST; 1s ago
     Docs: https://www.nagios.org/documentation
  Process: 18294 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (code=e
  Process: 18293 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/SUCCES
  Process: 18315 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nag
  Process: 18313 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/
 Main PID: 18325 (nagios)
    Tasks: 6 (limit: 2319)
   CGroup: /system.slice/nagios.service
```

Steps-

Go to google.com, Search NagiOs Demo

Click on the first link shown below



About 3,87,000 results (0.44 seconds)

https://exchange.nagios.org › directory › Demos › details

Nagios XI Online Demo

An online **demo** of **Nagios** XI. The **demo** allows you to test configuration wizards, dashlets, dashboards, views, and more. Reviews (0).

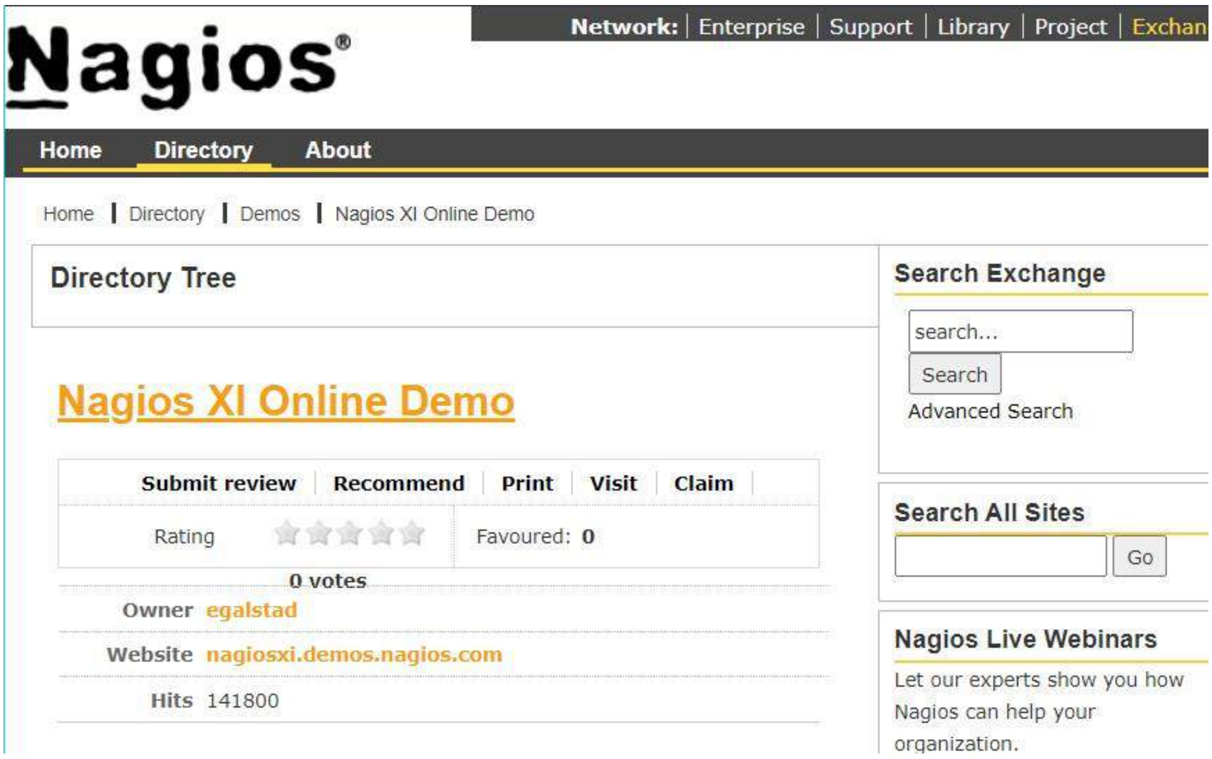
https://exchange.nagios.org › directory › Demos

Demos - Nagios Exchange

An online **demo** of **Nagios** Log Server. The **demo** allows you to view system logs and event logs, giving some examples on how you can visualize data sent into **Nagios** ...

https://exchange.nagios.org › directory › Demos › details

Now click on the website-


Network: | Enterprise | Support | Library | Project | Exchan

Home Directory About

Home | Directory | Demos | Nagios XI Online Demo

Directory Tree

Nagios XI Online Demo

Submit review | Recommend | Print | Visit | Claim

Rating	Favoured: 0

0 votes

Owner	egalstad
Website	nagiosxi.demos.nagios.com
Hits	141800

Search Exchange

search...

Search

Advanced Search

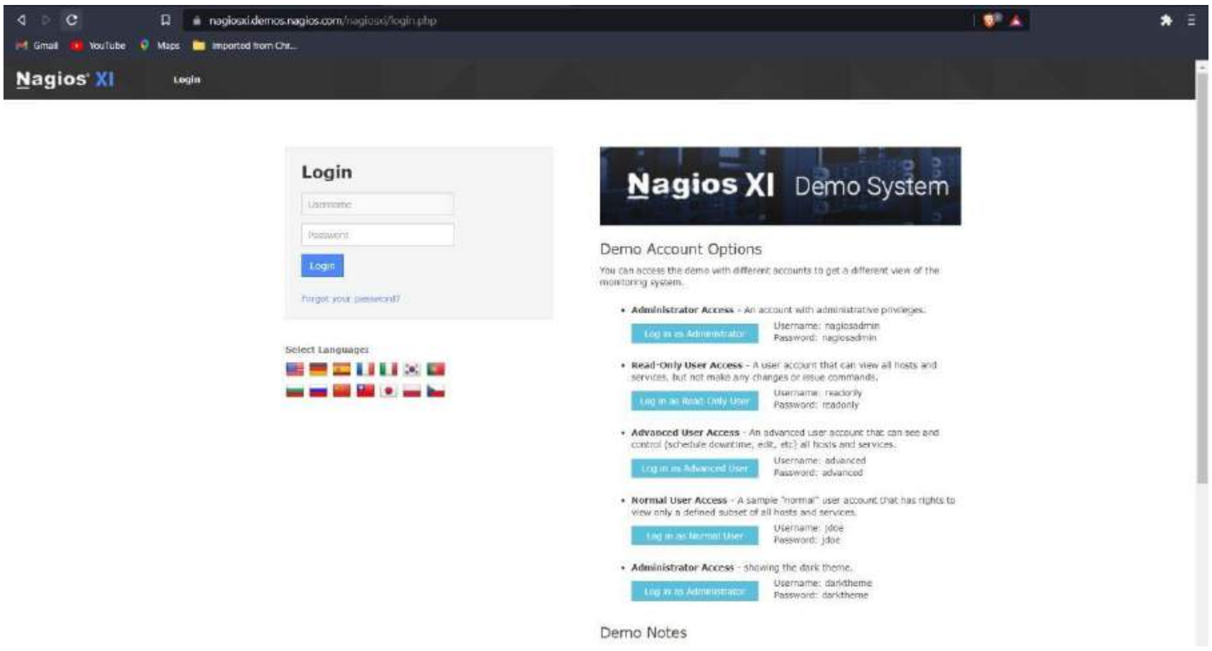
Search All Sites

Go

Nagios Live Webinars

Let our experts show you how Nagios can help your organization.

Now click on login as administrator


nagiosxi.demos.nagios.com/nagiosxi/login.php

Login

Username

Password

Login

Forgot your password?

Select Languages:

Nagios XI Demo System

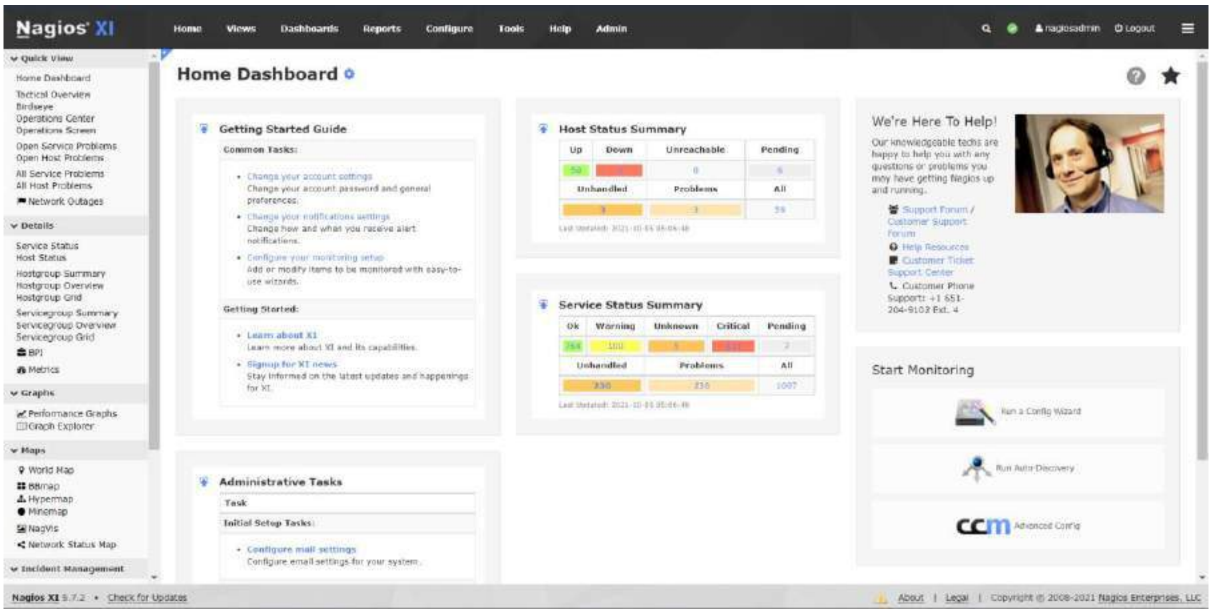
Demo Account Options

You can access the demo with different accounts to get a different view of the monitoring system.

- **Administrator Access** - An account with administrative privileges.
  Log in as Administrator
  Username: nagiosadmin
  Password: nagiosadmin
- **Read-Only User Access** - A user account that can view all hosts and services, but not make any changes or issue commands.
  Log in as Read Only User
  Username: readonly
  Password: readonly
- **Advanced User Access** - An advanced user account that can see and control (schedule downtime, edit, etc) all hosts and services.
  Log in as Advanced User
  Username: advanced
  Password: advanced
- **Normal User Access** - A sample "normal" user account that has rights to view only a defined subset of all hosts and services.
  Log in as Normal User
  Username: jdoe
  Password: jdoe
- **Administrator Access** - showing the dark theme.
  Log in as Administrator
  Username: darktheme
  Password: darktheme

Demo Notes

It will have interface like this



Home Dashboard

Getting Started Guide

Common Tasks:

- Change your account settings
  Change your account password and general preferences.
- Change your notifications settings
  Change how and when you receive alert notifications.
- Configure your monitoring setup
  Add or modify items to be monitored with easy-to-use wizards.

Getting Started:

- Learn about XI
  Learn more about XI and its capabilities.
- Signup for XI news
  Stay informed on the latest updates and happenings for XI.

Administrative Tasks

Task

Initial Setup Tasks:

- Configure mail settings
  Configure email settings for your system.

Host Status Summary

Up	Down	Unreachable	Pending
50	3	0	6
Unhandled	Problems	All	
3	3	59	

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
764	100	5	131	7
Unhandled	Problems	All		
230	236	1007		

We're Here To Help!

Our knowledgeable techs are happy to help you with any questions or problems you may have getting Nagios up and running.

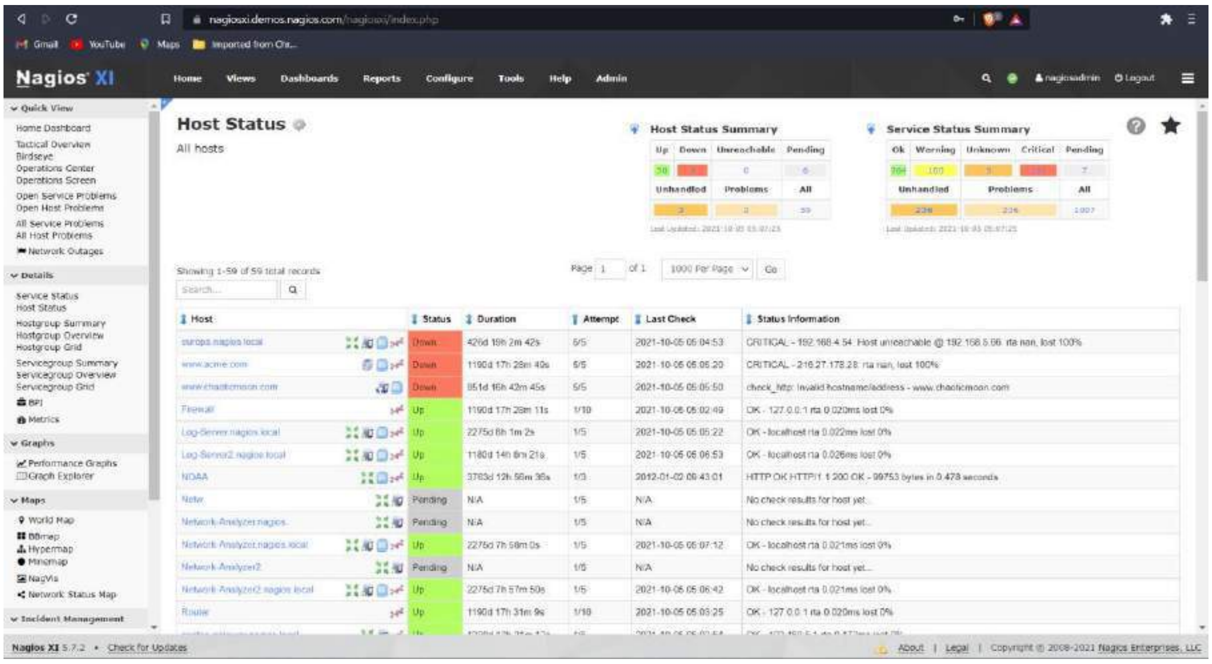
Support Forum / Customer Support Forum
Help Resources
Customer Ticket Support Center
Customer Phone Support: +1 651-204-9102 Ext. 4

Start Monitoring

Run a Config Wizard
Run Auto-Discovery
Advanced Config

Nagios XI 5.7.2 • Check for Updates | About | Legal | Copyright © 2008-2021 Nagios Enterprises, LLC

Now click on Host status-



Host Status

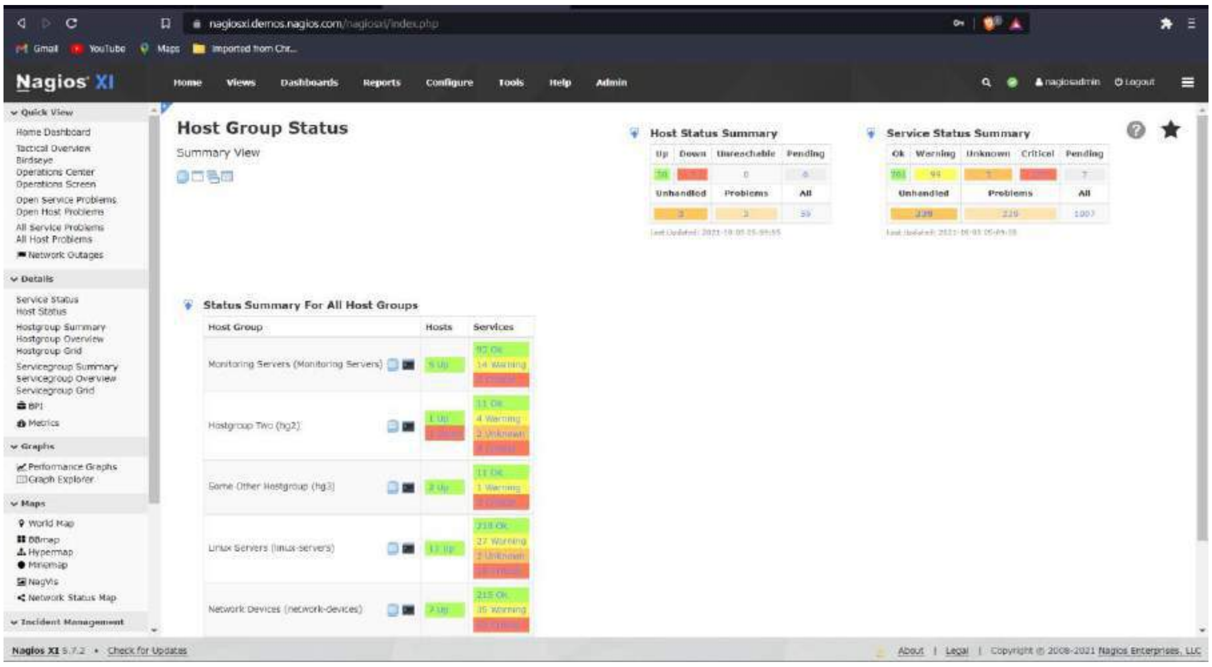
All hosts

Showing 1-59 of 59 total records

Page 1 of 1 | 1000 Per Page | Go

Host	Status	Duration	Attempt	Last Check	Status Information
europa.nagios.local	Down	426d 19h 2m 42s	5/5	2021-10-05 05:04:53	CRITICAL - 192.168.4.54: Host unreachable @ 192.168.5.66. rta nan, lost 100%
www.acme.com	Down	1190d 17h 28m 40s	5/5	2021-10-05 05:05:20	CRITICAL - 216.27.178.28: rta nan, lost 100%
www.chaoticmoon.com	Down	851d 16h 42m 45s	5/5	2021-10-05 05:05:50	check_http: Invalid hostname/address - www.chaoticmoon.com
Firewall	Up	1190d 17h 28m 11s	1/10	2021-10-05 05:02:49	OK - 127.0.0.1 rta 0.020ms lost 0%
Log-Server.nagios.local	Up	2275d 8h 1m 2s	1/5	2021-10-05 05:05:22	OK - localhost rta 0.022ms lost 0%
Log-Server2.nagios.local	Up	1180d 14h 8m 21s	1/5	2021-10-05 05:06:53	OK - localhost rta 0.026ms lost 0%
NOAA	Up	3783d 12h 56m 36s	1/3	2012-01-02 09:43:01	HTTP OK HTTP/1.1 200 OK - 99753 bytes in 0.478 seconds
Netw	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer.nagios	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer.nagios.local	Up	2275d 7h 58m 0s	1/5	2021-10-05 05:07:12	OK - localhost rta 0.021ms lost 0%
Network-Analyzer2	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer2.nagios.local	Up	2275d 7h 57m 50s	1/5	2021-10-05 05:06:42	OK - localhost rta 0.021ms lost 0%
Router	Up	1190d 17h 31m 9s	1/10	2021-10-05 05:03:25	OK - 127.0.0.1 rta 0.020ms lost 0%

In the above image one can see Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.



Host Group Status

Summary View

Host Status Summary

Up	Down	Unreachable	Pending
58	[illegible]	0	0
Unhandled	Problems		All
3	3		59

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
761	99	5	[illegible]	7
Unhandled		Problems		All
239		229		1007

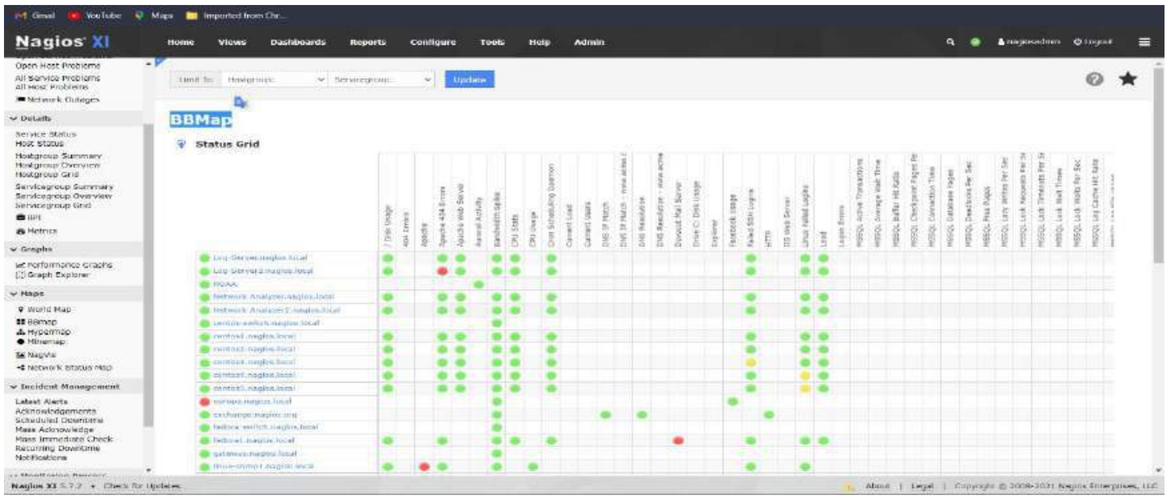
Status Summary For All Host Groups

Host Group	Hosts	Services
Monitoring Servers (Monitoring Servers)	5 Up	92 Ok 14 Warning [illegible] Critical
Hostgroup Two (hg2)	1 Up [illegible]	11 Ok 4 Warning 2 Unknown [illegible] Critical
Some Other Hostgroup (hg3)	2 Up	11 Ok 1 Warning [illegible] Critical
Linux Servers (linux-servers)	11 Up	218 Ok 27 Warning 2 Unknown [illegible] Critical
Network Devices (network-devices)	7 Up	215 Ok 35 Warning [illegible] Critical

Here we can see Status Summary For All Host Groups

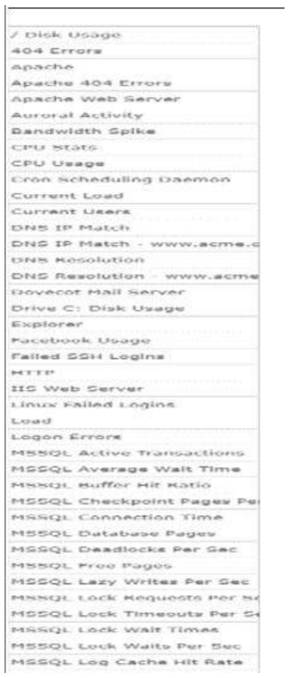
Now we click on BBMap

In this we can see status of following stuff in each host-



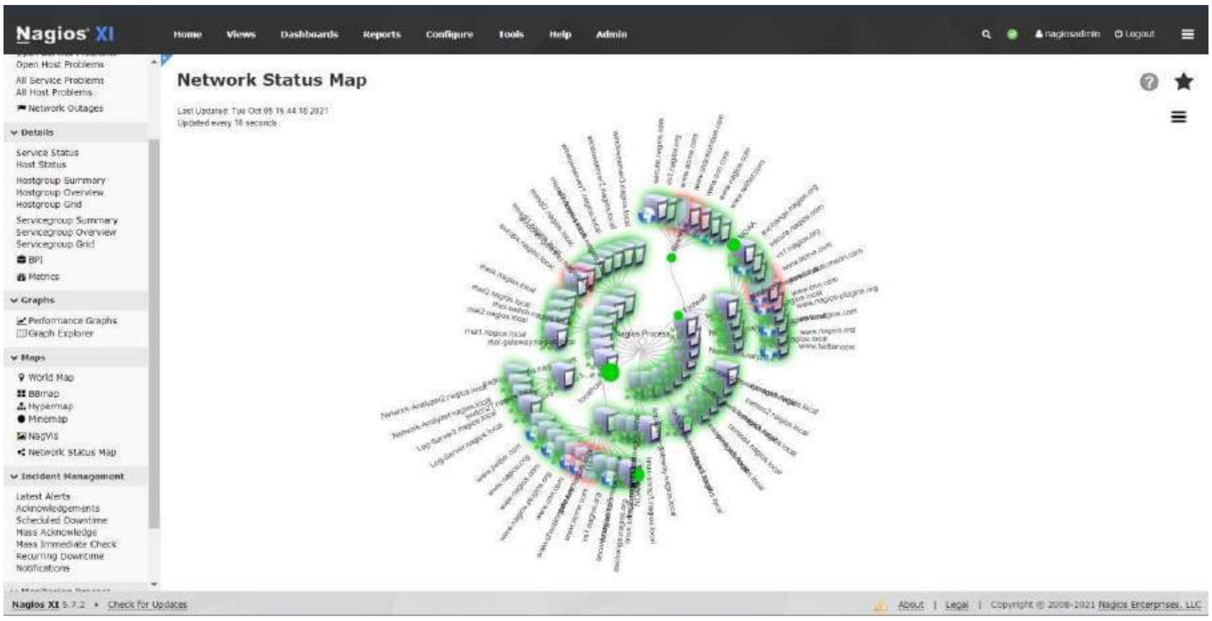
BBMap

Status Grid



/ Disk Usage
404 Errors
Apache
Apache 404 Errors
Apache Web Server
Auroral Activity
Bandwidth Spike
CPU Stats
CPU Usage
Cron Scheduling Daemon
Current Load
Current Users
DNS IP Match
DNS IP Match - www.acme.c
DNS Resolution
DNS Resolution - www.acme
Dovecot Mail Server
Drive C: Disk Usage
Explorer
Facebook Usage
Failed SSH Logins
HTTP
IIS Web Server
Linux Failed Logins
Load
Logon Errors
MSSQL Active Transactions
MSSQL Average Wait Time
MSSQL Buffer Hit Ratio
MSSQL Checkpoint Pages Pe
MSSQL Connection Time
MSSQL Database Pages
MSSQL Deadlocks Per Sec
MSSQL Free Pages
MSSQL Lazy Writes Per Sec
MSSQL Lock Requests Per S
MSSQL Lock Timeouts Per S
MSSQL Lock Wait Times
MSSQL Lock Waits Per Sec
MSSQL Log Cache Hit Rate

Now we have Network status map which is graphical representation of the network status



Network Status Map

Last Updated: Tue Oct 05 16:44:16 2021
Updated every 10 seconds

CONCLUSION : In this assignment we learned about how to continuously monitor Nagios commands and Installation and configuration of Nagios on Linux Machine.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 9

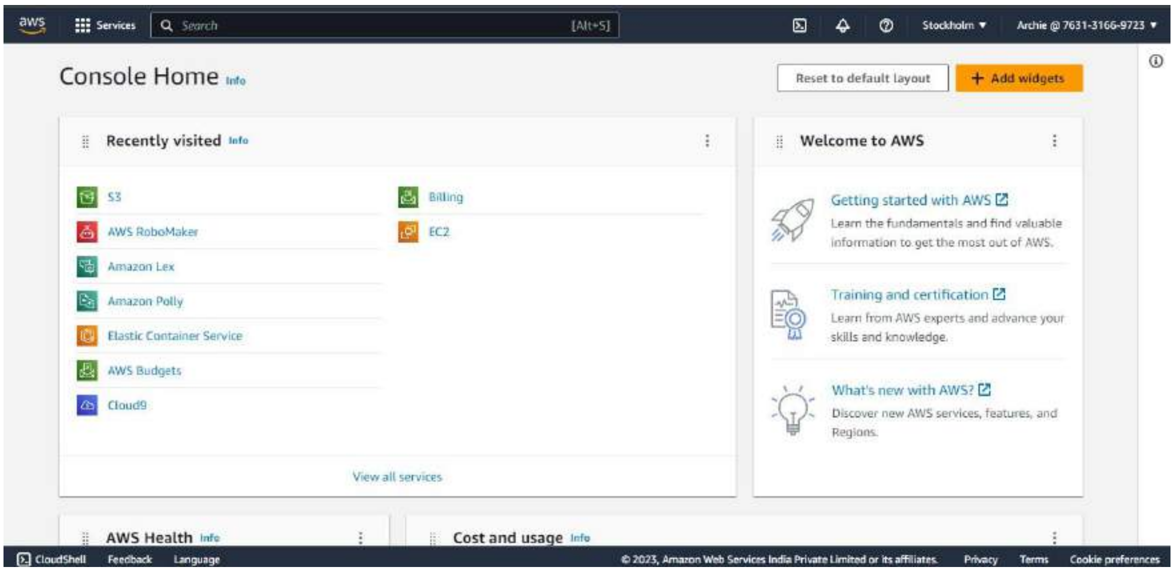
AIM : To understand Lambda Function and create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.

LO MAPPED : LO1,LO5

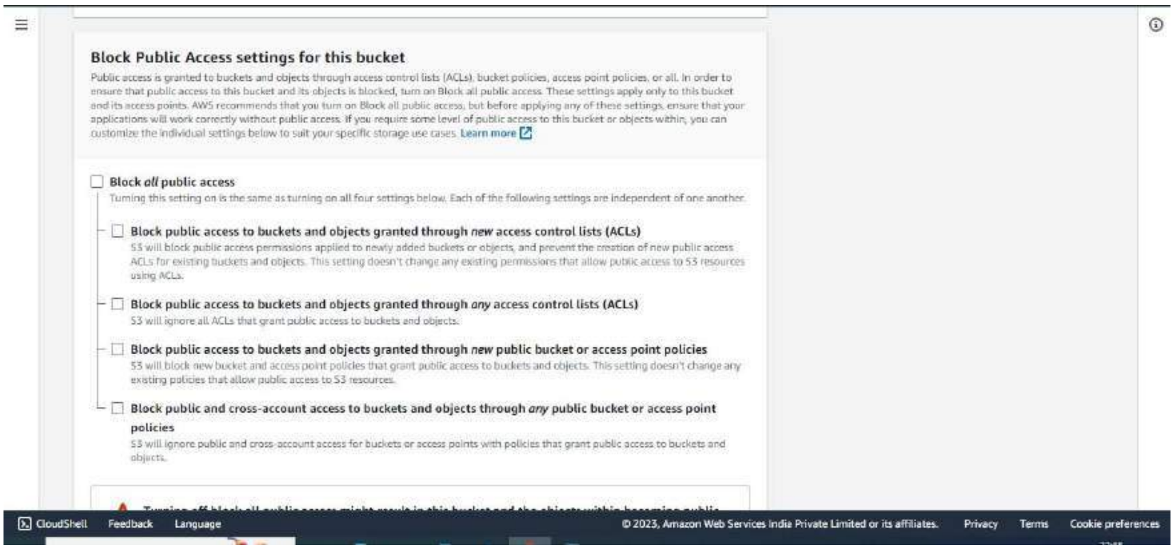
THEORY :
You can use Lambda to process event notifications from Amazon Simple Storage Service. Amazon S3 can send an event to a Lambda function when an object is created or deleted. You configure notification settings on a bucket, and grant Amazon S3 permission to invoke a function on the function's resource-based permissions policy.

STEPS TO FOLLOW:

1. Log in as IAM User



2. Create a S3 bucket and Enable the "Block all public access"



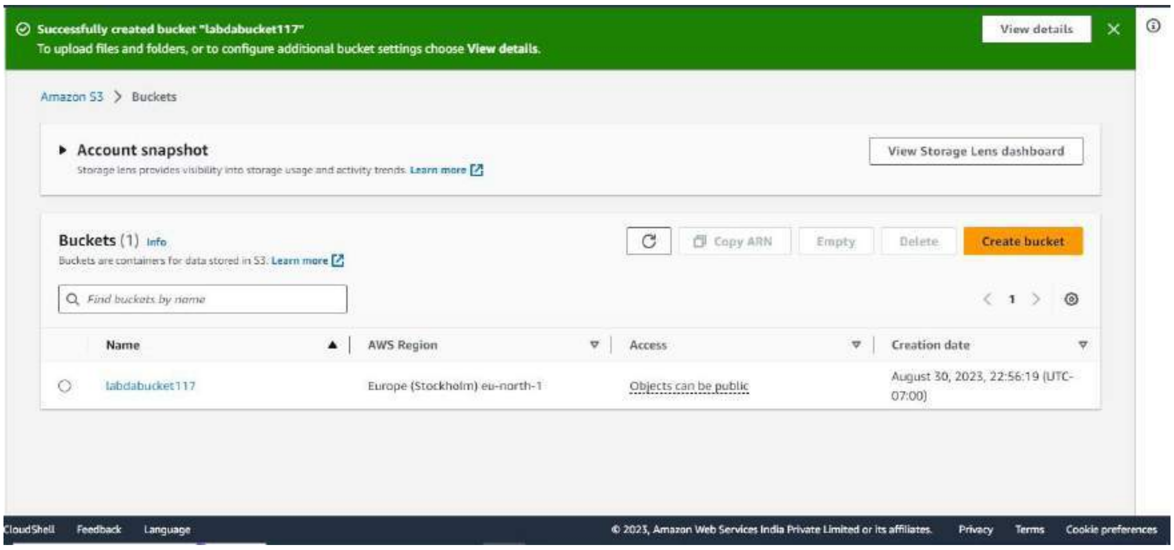
Block Public Access settings for this bucket

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

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



Successfully created bucket "labdabucket117"
To upload files and folders, or to configure additional bucket settings choose **View details**.

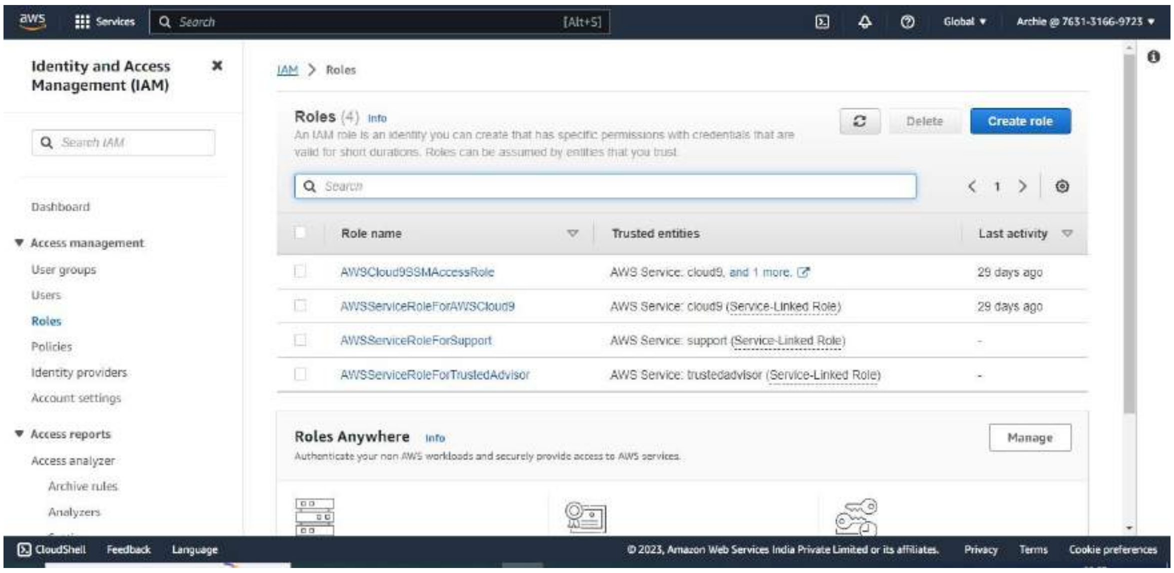
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
labdabucket117	Europe (Stockholm) eu-north-1	Objects can be public	August 30, 2023, 22:56:19 (UTC-07:00)

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

3. Search IAM on the console and go to "Roles". Click on Create Roles



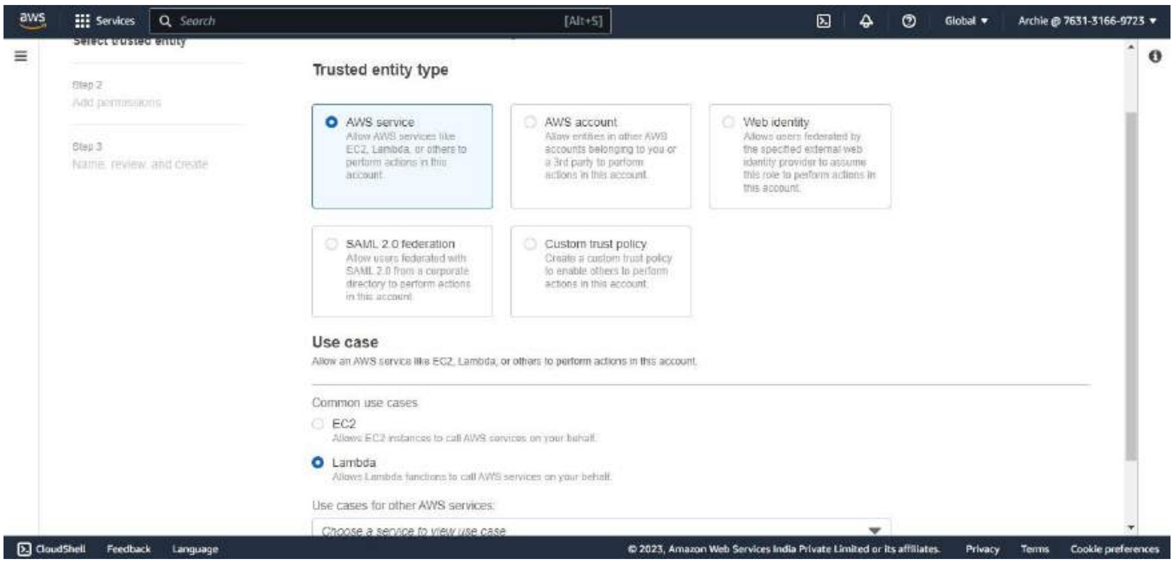
Roles (4) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSCloud9SSMAccessRole	AWS Service: cloud9, and 1 more.	29 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)	29 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-

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

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

4. Select the options of "AWS service" and "lambda"



Trusted entity type

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

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

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

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

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

Use case

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

Common use cases

EC2: Allows EC2 instances to call AWS services on your behalf.

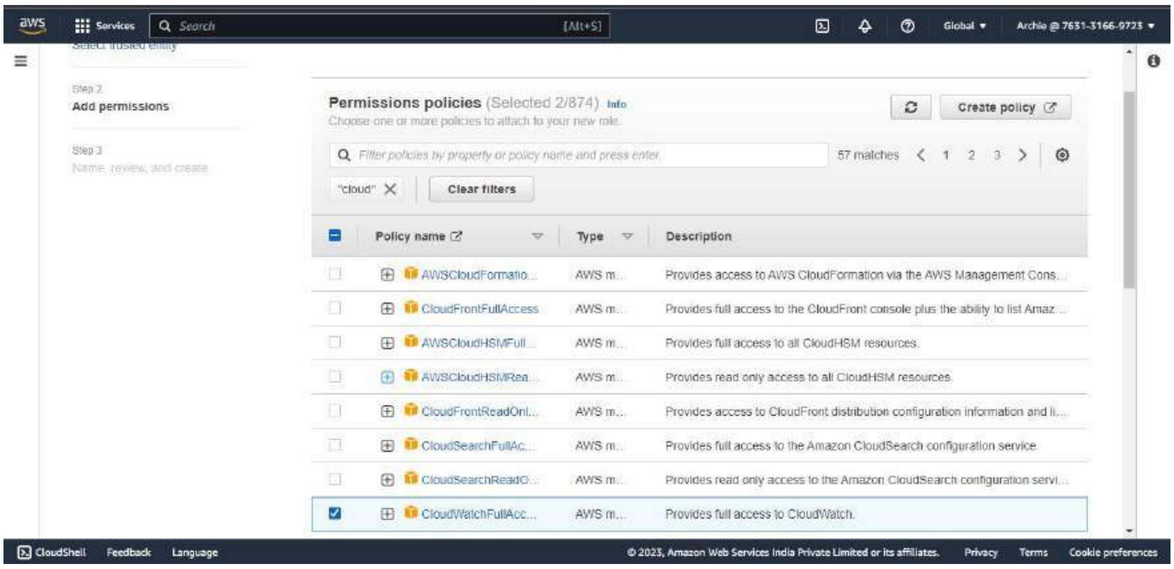
[✓ Lambda] Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

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

5. Enable the "CloudWatchFullAccess" and "AmazonS3FullAccess"


Step 2
Add permissions

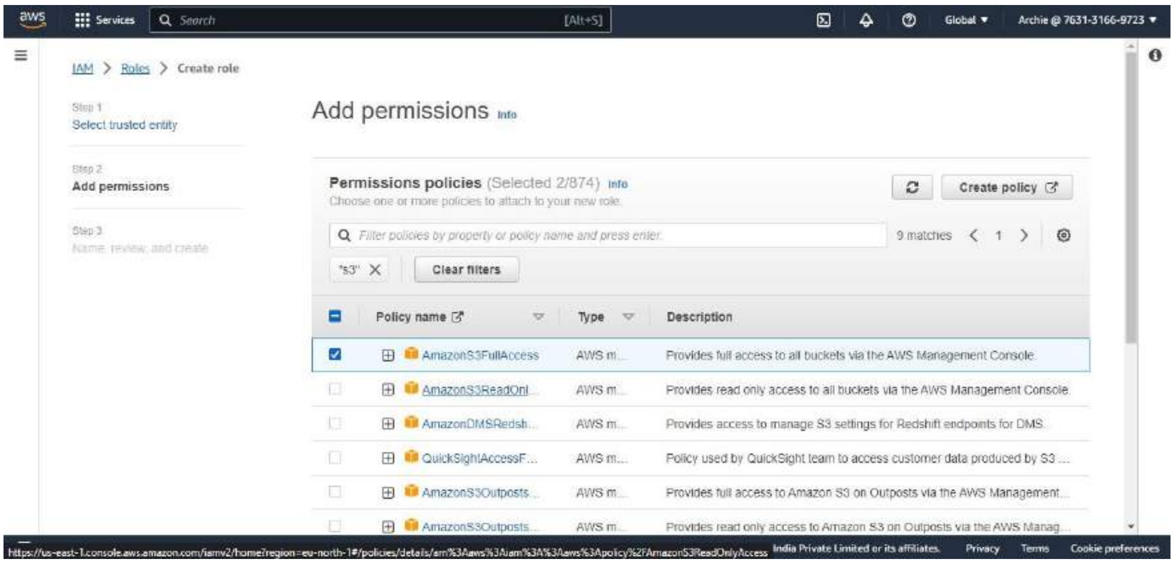
Step 3
Name, review, and create

Permissions policies (Selected 2/874) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 57 matches

"cloud" Clear filters

	Policy name	Type	Description
☐	AWSCloudFormatio...	AWS m...	Provides access to AWS CloudFormation via the AWS Management Cons...
☐	CloudFrontFullAccess	AWS m...	Provides full access to the CloudFront console plus the ability to list Amaz...
☐	AWSCloudHSMFull...	AWS m...	Provides full access to all CloudHSM resources.
☐	AWSCloudHSMRea...	AWS m...	Provides read only access to all CloudHSM resources.
☐	CloudFrontReadOnl...	AWS m...	Provides access to CloudFront distribution configuration information and li...
☐	CloudSearchFullAc...	AWS m...	Provides full access to the Amazon CloudSearch configuration service
☐	CloudSearchReadO...	AWS m...	Provides read only access to the Amazon CloudSearch configuration servi...
☑	CloudWatchFullAcc...	AWS m...	Provides full access to CloudWatch.


IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

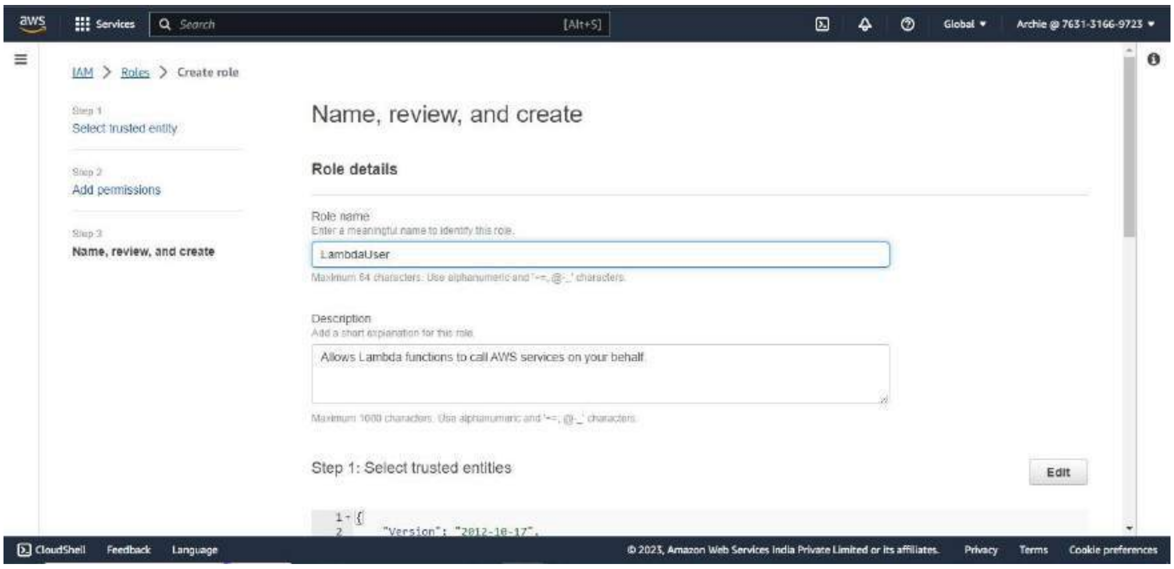
Permissions policies (Selected 2/874) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 9 matches

"s3" Clear filters

	Policy name	Type	Description
☑	AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.
☐	AmazonS3ReadOnl...	AWS m...	Provides read only access to all buckets via the AWS Management Console.
☐	AmazonDMSRedsh...	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
☐	QuickSightAccessF...	AWS m...	Policy used by QuickSight team to access customer data produced by S3 ...
☐	AmazonS3Outposts...	AWS m...	Provides full access to Amazon S3 on Outposts via the AWS Management...
☐	AmazonS3Outposts...	AWS m...	Provides read only access to Amazon S3 on Outposts via the AWS Manag...

6. Click on Create Role and you will be redirected to this dashboard. Give the Role a Name and Click on Done.


IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

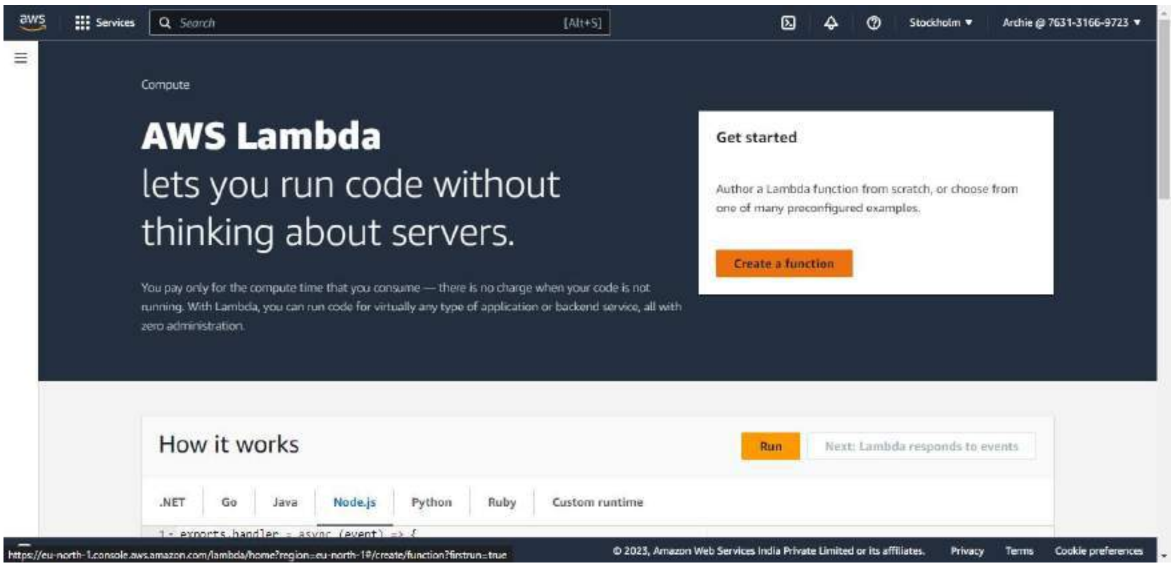
Role name
Enter a meaningful name to identify this role.
LambdaUser
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities Edit

```
{
    "Version": "2012-10-17",
```

7. Search for Lambda in the console and click on Create Function.



Compute

AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

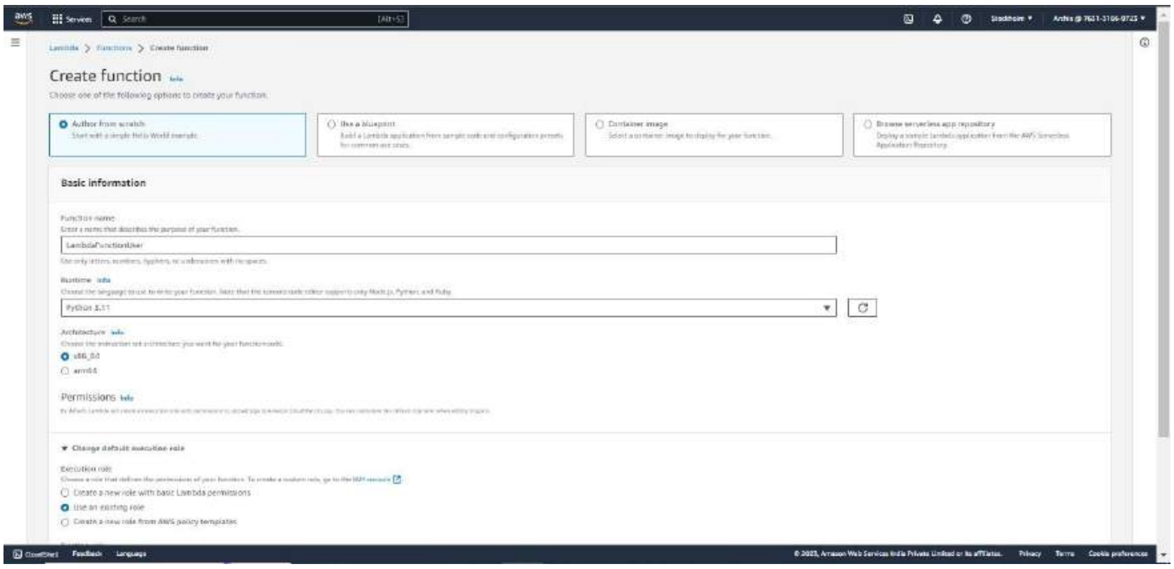
How it works

```
exports.handler = async (event) => {
```

https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/create/function?firstrun=true

© 2023, Amazon Web Services India Private Limited or its affiliates.

8. Change the settings of the Lambda Function.



Create function

Choose one of the following options to create your function.

[✓ Author from scratch] Start with a simple Hello World example.
Use a blueprint
Container image
Browse serverless app repository

Basic information

Function name: LambdaFunctionUser

Runtime: Python 3.11

Architecture: [✓ x86_64] arm64

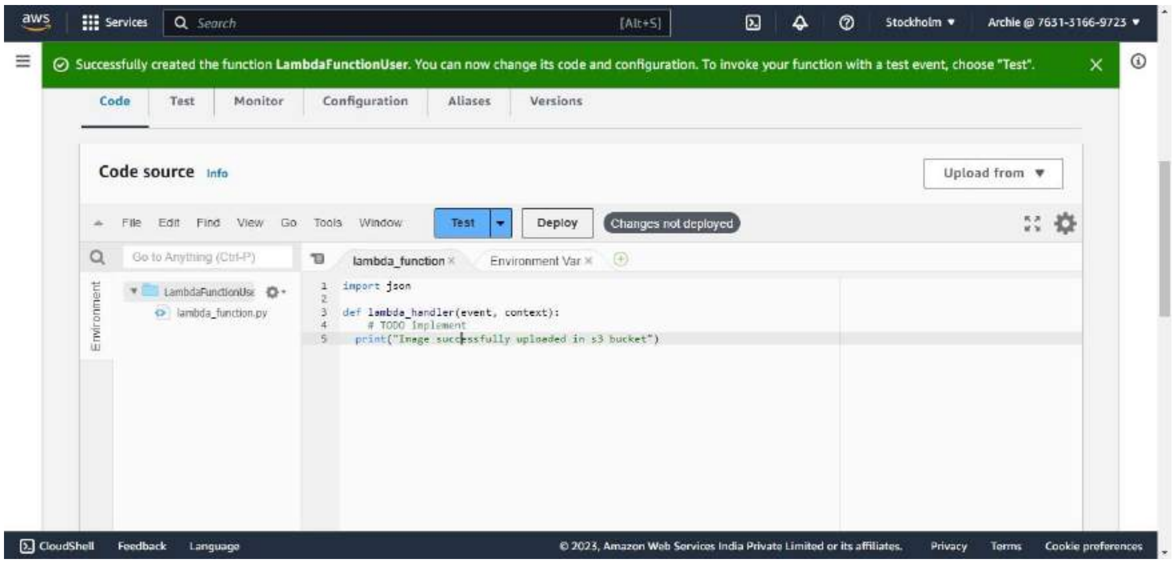
Permissions

Change default execution role

Execution role
Create a new role with basic Lambda permissions
[✓ Use an existing role]
Create a new role from AWS policy templates

© 2023, Amazon Web Services India Private Limited or its affiliates.

9. Add the python code and click on deploy to save the changes.



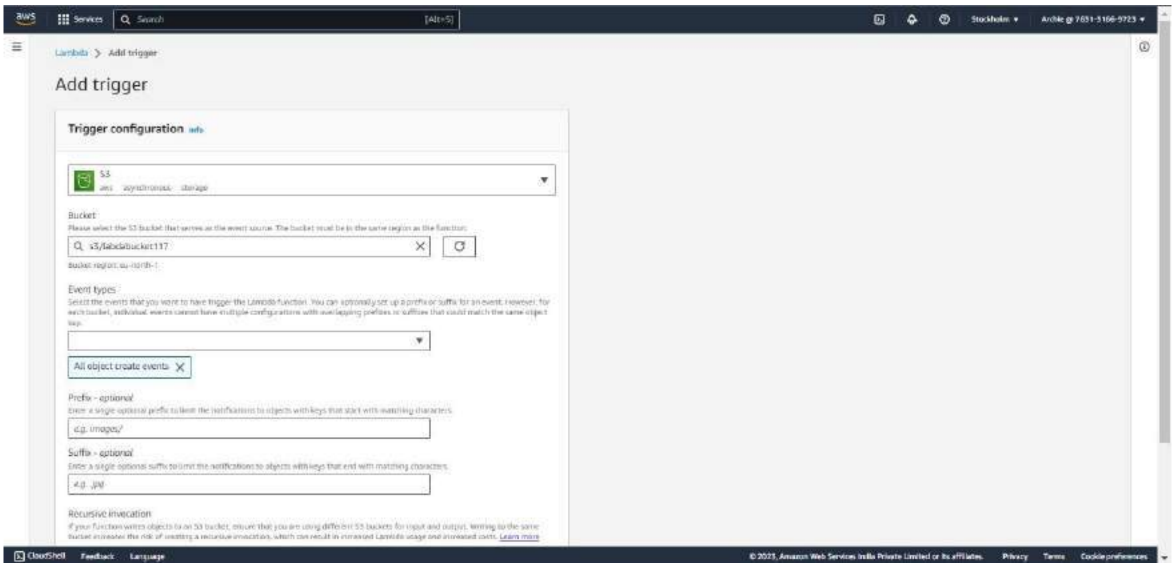
Successfully created the function LambdaFunctionUser. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code source

```python
import json

def lambda_handler(event, context):
    # TODO Implement
  print("Image successfully uploaded in s3 bucket")
```

© 2023, Amazon Web Services India Private Limited or its affiliates.

10. Scroll above and click on ADD TRIGGER. Select the following options and click on Done.



Add trigger

Trigger configuration

S3

Bucket: s3/labdabucket117

Bucket region: eu-north-1

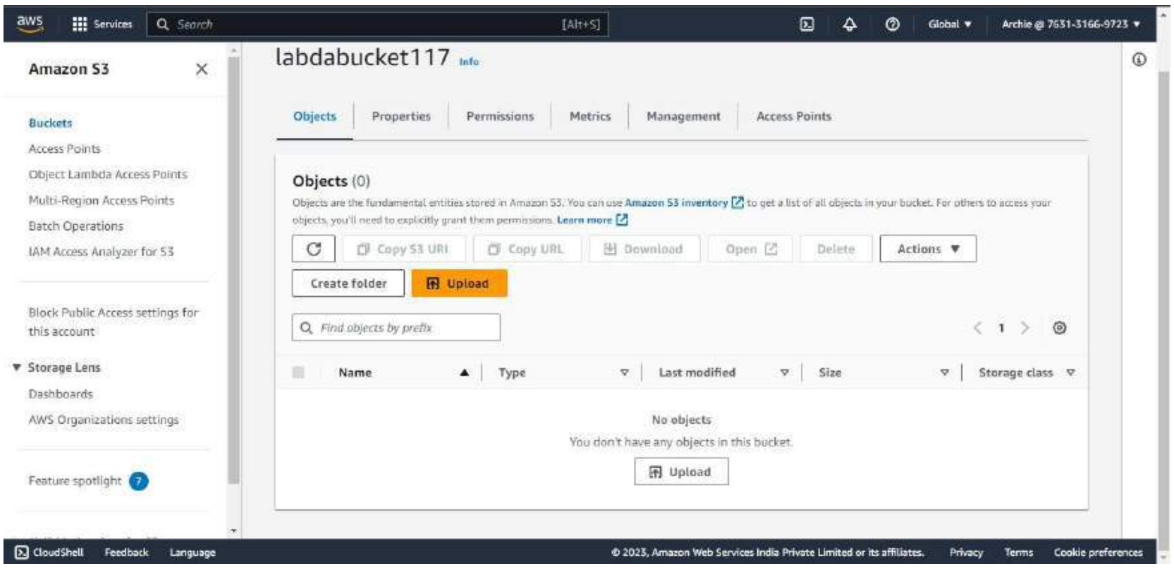
Event types: All object create events

Prefix - optional: e.g. images/

Suffix - optional: e.g. .jpg

Recursive invocation

11. Go to S3 bucket and click on Add files. Select a image and click on Upload.



Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

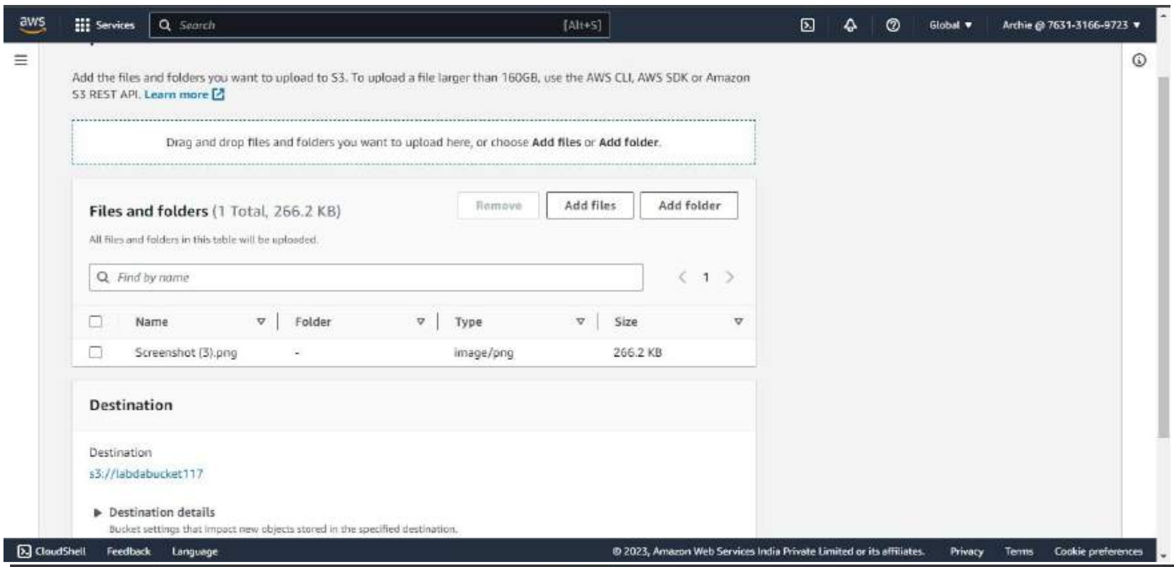
Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

Upload

© 2023, Amazon Web Services India Private Limited or its affiliates.



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

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

Files and folders (1 Total, 266.2 KB)

All files and folders in this table will be uploaded.

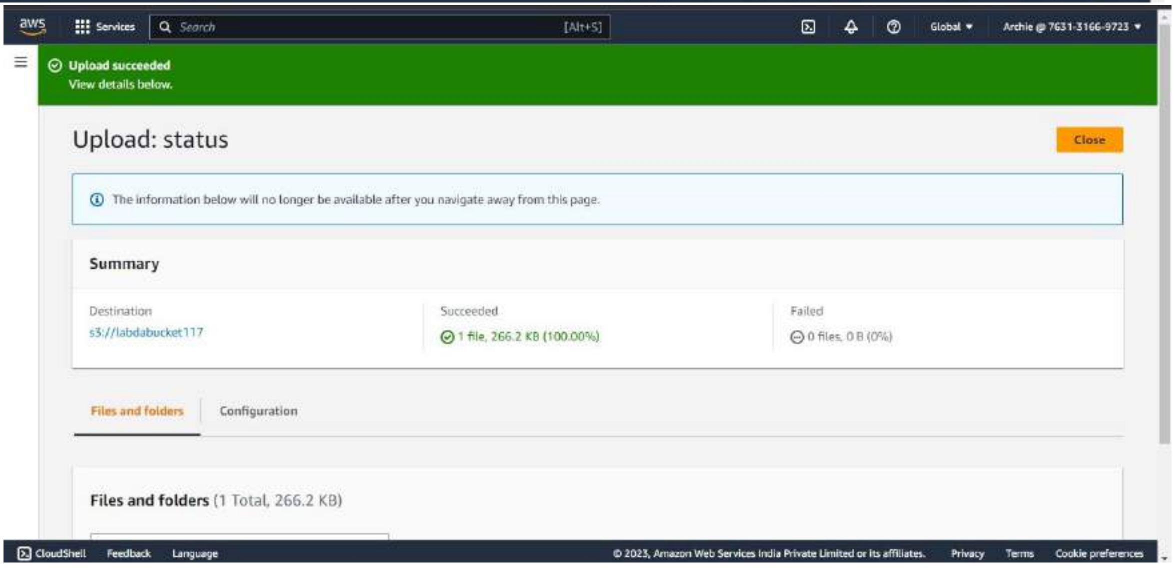
Name	Folder	Type	Size
Screenshot (3).png	-	image/png	266.2 KB

Destination

Destination
s3://labdabucket117

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

© 2023, Amazon Web Services India Private Limited or its affiliates.



Upload succeeded
View details below.

Upload: status

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

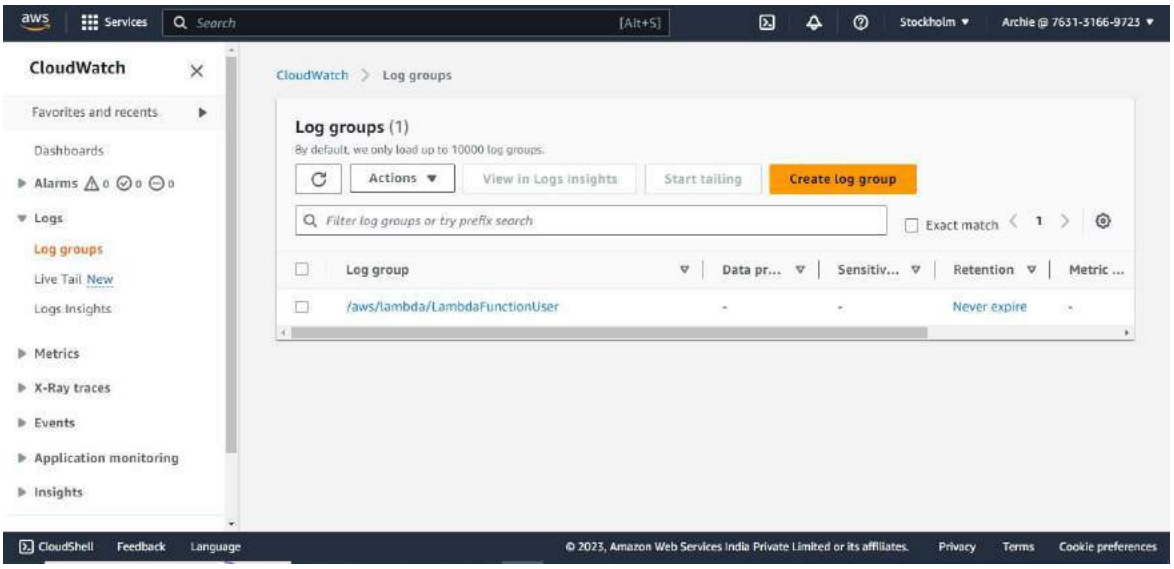
Summary

Destination	Succeeded	Failed
s3://labdabucket117	1 file, 266.2 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 266.2 KB)

© 2023, Amazon Web Services India Private Limited or its affiliates.

12. Search CloudWatch and go to Log groups. Select the existing Log Group.

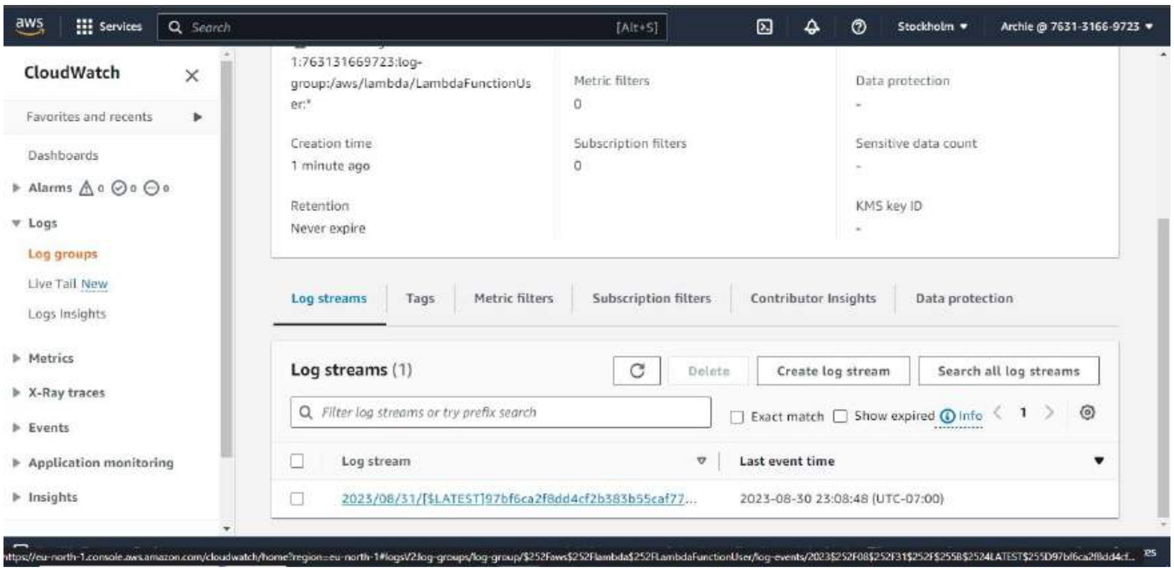

CloudWatch > Log groups

Log groups (1)

By default, we only load up to 10000 log groups.

Log group	Data pr...	Sensitiv...	Retention	Metric ...
/aws/lambda/LambdaFunctionUser	-	-	Never expire	-

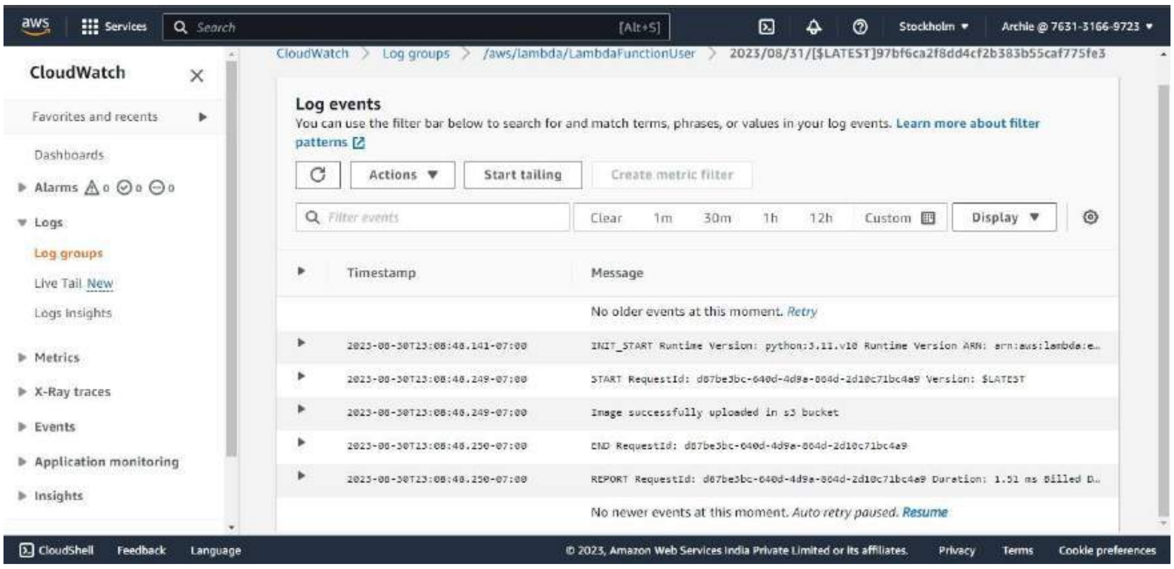
13. Click on the link provided and then you will see the message displayed.


1:763131669723:log-group:/aws/lambda/LambdaFunctionUser:*

Creation time: 1 minute ago	Metric filters: 0	Data protection: -
Retention: Never expire	Subscription filters: 0	Sensitive data count: -
		KMS key ID: -

Log streams (1)

Log stream	Last event time
2023/08/31/[$LATEST]97bf6ca2f8dd4cf2b383b55caf77...	2023-08-30 23:08:48 (UTC-07:00)


CloudWatch > Log groups > /aws/lambda/LambdaFunctionUser > 2023/08/31/[$LATEST]97bf6ca2f8dd4cf2b383b55caf775fe3

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                        Message
                                 No older events at this moment. Retry
2023-08-30T23:08:48.141-07:00    INIT_START Runtime Version: python:3.11.v10 Runtime Version ARN: arn:aws:lambda:e...
2023-08-30T23:08:48.249-07:00    START RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Version: $LATEST
2023-08-30T23:08:48.249-07:00    Image successfully uploaded in s3 bucket
2023-08-30T23:08:48.250-07:00    END RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9
2023-08-30T23:08:48.250-07:00    REPORT RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Duration: 1.51 ms Billed D...
                                 No newer events at this moment. Auto retry paused. Resume
```

CONCLUSION : In this assignment, we learnt how to create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.

NAME : SOUMIL SALVI

ROLL NO : 104

ASSIGNMENT 10

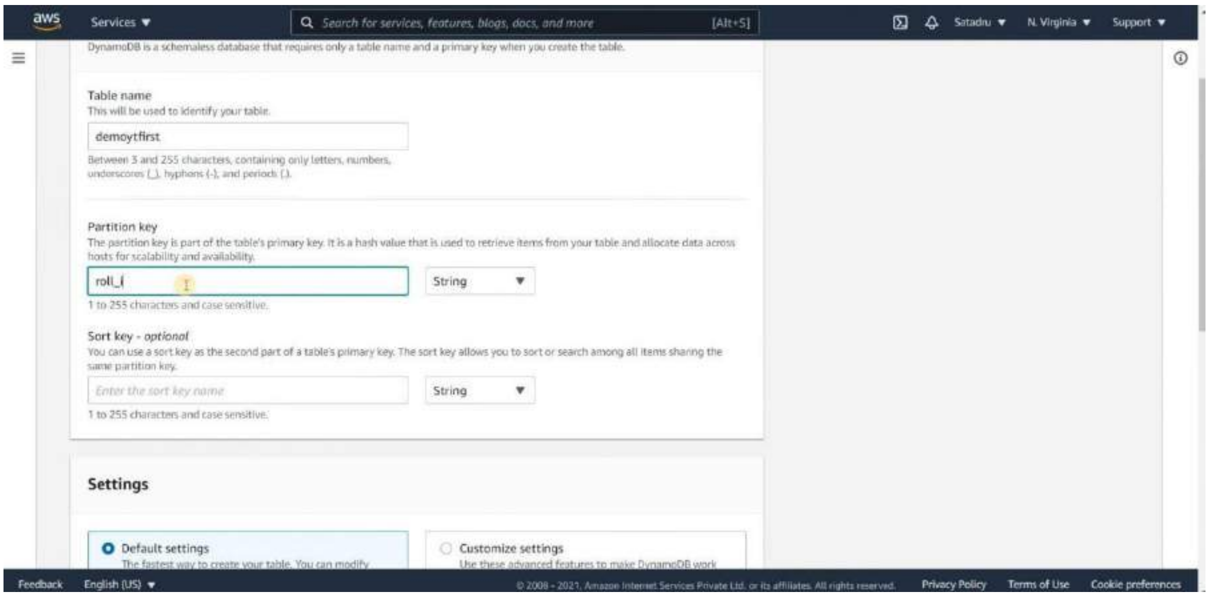
AIM : To create a Lambda function using Python for adding data to Dynamo DB database.

LO MAPPED : LO1,LO6

THEORY :

Step 1: Set Up AWS Environment

1. Create an AWS Account: If you don't have an AWS account, sign up for one at AWS Console.
2. Access AWS Lambda Console:


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Satadru
N. Virginia
Support

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
demoytfirst
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
roll_i String
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name String
1 to 255 characters and case sensitive.

Settings

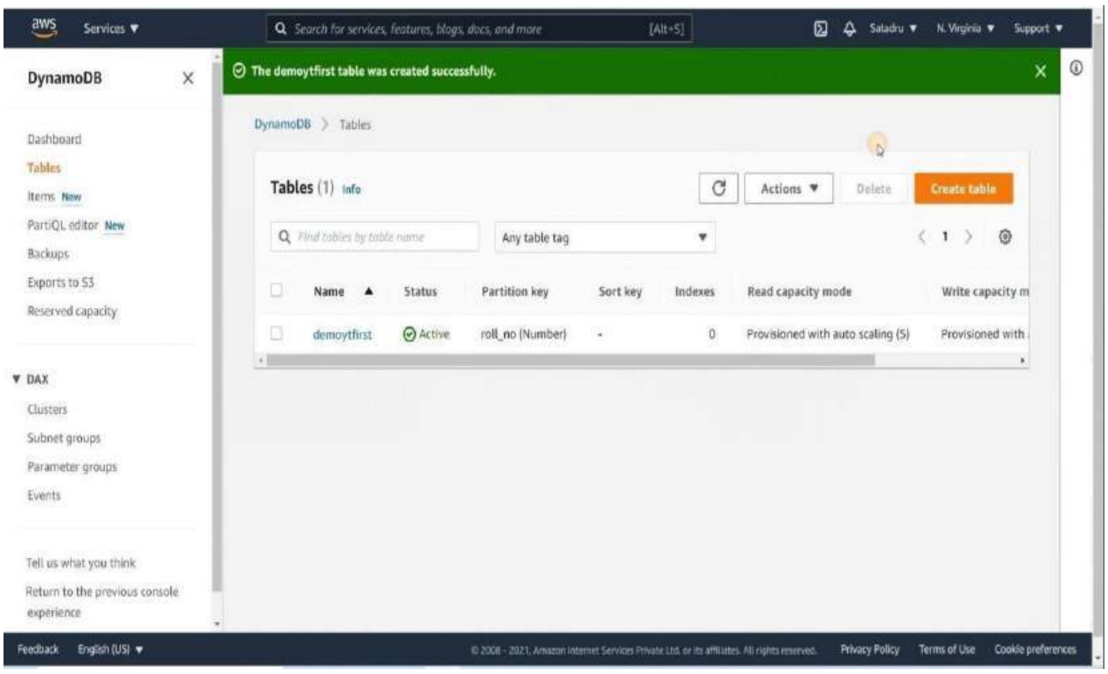
Default settings
The fastest way to create your table. You can modify

Customize settings
Use these advanced features to make DynamoDB work

Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

- Go to the AWS Lambda Console.
- Click on "Create function."

3. Create a New Lambda Function



The demoytfirst table was created successfully.

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity m
demoytfirst	Active	roll_no (Number)	-	0	Provisioned with auto scaling (5)	Provisioned with

- Choose "Author from scratch."
- Enter a name for your function (e.g., AddToDynamoDBFunction).
- Choose "Python" as the runtime.

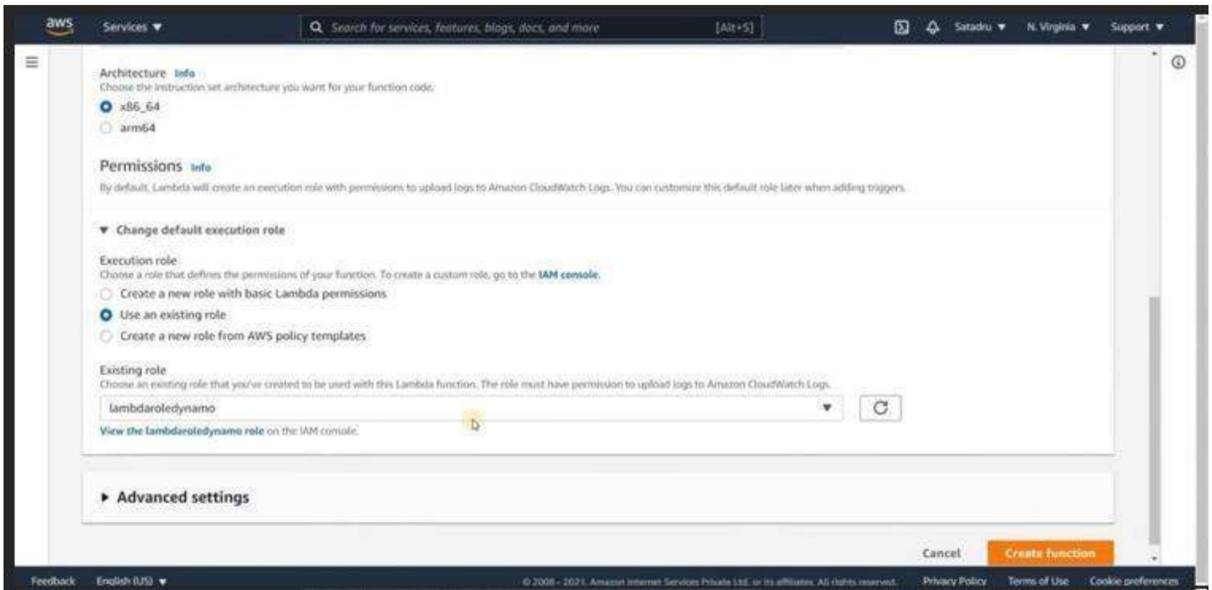
4. Configure Execution Role

- Create a new role with basic Lambda permissions and additional permissions to interact with DynamoDB.
- Attach the role to your Lambda function.

Step 2: Set Up DynamoDB

1. Access DynamoDB Console:

    - Go to the AWS DynamoDB Console.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Satadru
N. Virginia
Support
Cancel
Create function
Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

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

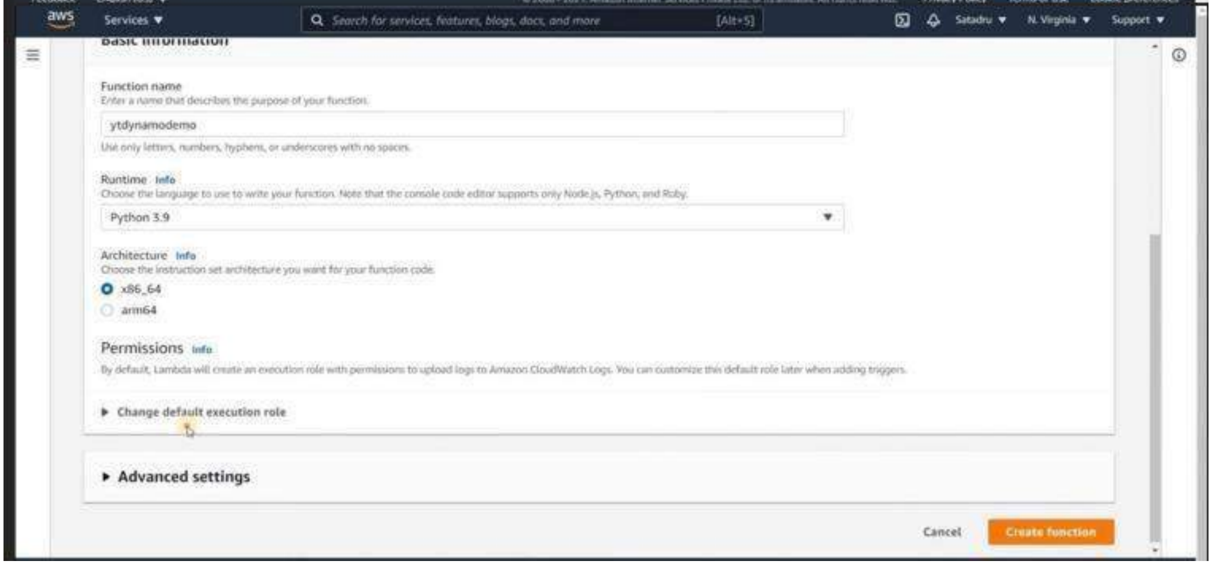
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
[✓ Use an existing role]
Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
lambdaroledynamo
View the lambdaroledynamo role on the IAM console.

▶ Advanced settings


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Satadru
N. Virginia
Support
Cancel
Create function

Basic information

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

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.9

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

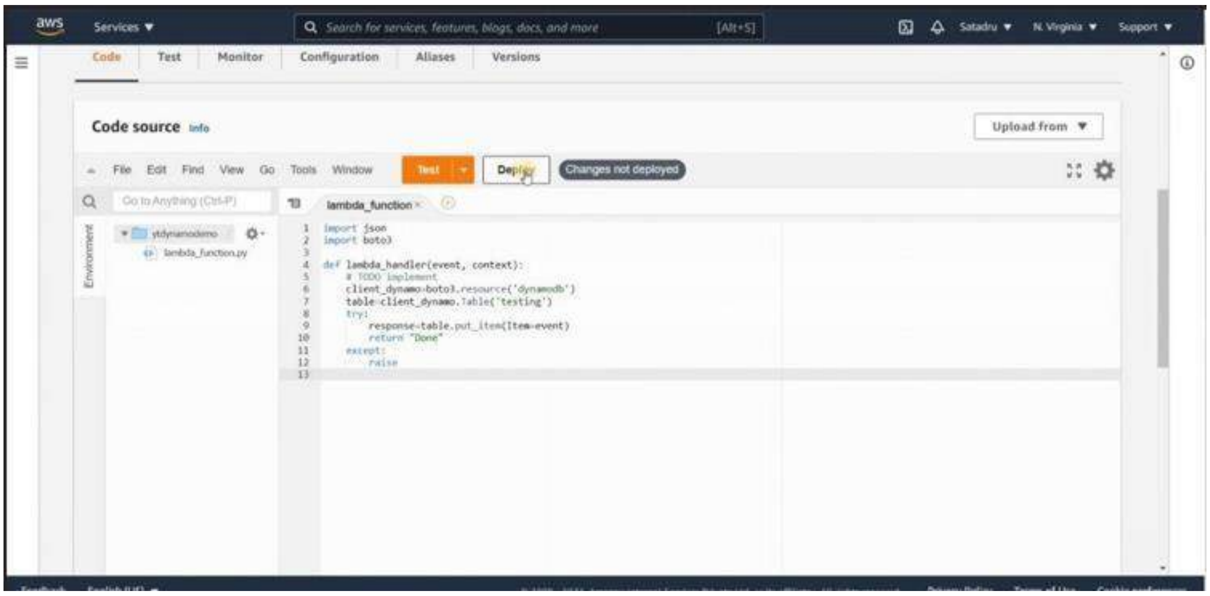
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

▶ Advanced settings

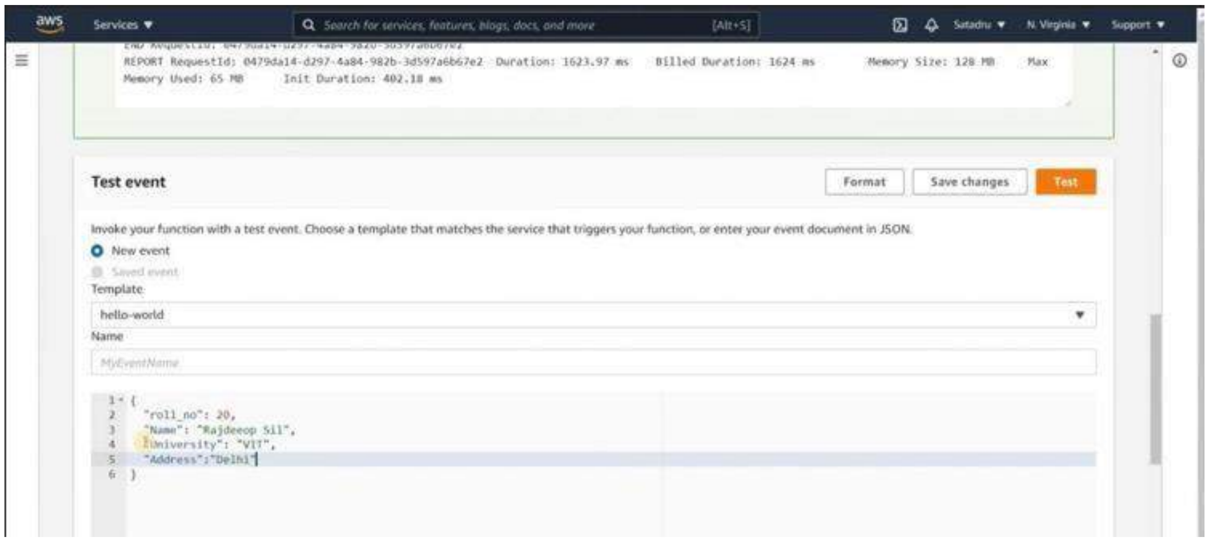
2. Create a DynamoDB Table:

- Click on "Create table."
- Enter a table name and a primary key (e.g., ID).
- Configure other settings as needed and create the table.



```
import json
import boto3

def lambda_handler(event, context):
    # TODO implement
    client_dynamo=boto3.resource('dynamodb')
    table=client_dynamo.Table('testing')
    try:
        response=table.put_item(Item=event)
        return "Done"
    except:
        raise
```

Step 3: Write Lambda Function Code



```
REPORT RequestId: 0479da14-d297-4a84-982b-3d597a6b67e2  Duration: 1623.97 ms  Billed Duration: 1624 ms   Memory Size: 128 MB   Max
Memory Used: 65 MB      Init Duration: 402.18 ms
```

Test event

Invoke your function with a test event. Choose a template that matches the service that triggers your function, or enter your event document in JSON.

● New event

○ Saved event

Template: hello-world

Name: MyEventName

```
{
  "roll_no": 20,
  "Name": "Rajdeeop Sil",
  "University": "VIT",
  "Address":"Delhi"
}
```

. Configure Lambda Handler:

- In the Lambda function configuration, set the handler to filename.lambda_handler (e.g., yourfilename.lambda_handler).

Step 4: Configure Environment Variables

1. Add DynamoDB Table Name:

- In the Lambda function configuration, add an environment variable (e.g., DYNAMODB_TABLE) with the value set to your DynamoDB table name.

Step 5: Test Locally

1. Test Lambda Function Locally:

- Create a test event with sample data.
- Test the Lambda function using the "Test" button in the Lambda console.

Step 6: Deploy Lambda Function

1. Deploy the Lambda Function:

    - Click on the "Deploy" button in the Lambda console.

Step 7: Test in AWS Lambda Console

1. Test in AWS Lambda Console:

- Use the Lambda console to test the function by creating a test event with sample data.
- Verify that the function executes successfully.

Step 8: Monitor and Troubleshoot

1. Check CloudWatch Logs:

- Use CloudWatch Logs to check logs generated by your Lambda function for troubleshooting.

Step 9: Documentation

1. Create Documentation:

- Document the purpose, input parameters, and expected output of your Lambda function.
- Explain any challenges faced during development and how they were resolved.

By following these steps, you can create a Lambda function using Python to add data to a DynamoDB database in the AWS environment.

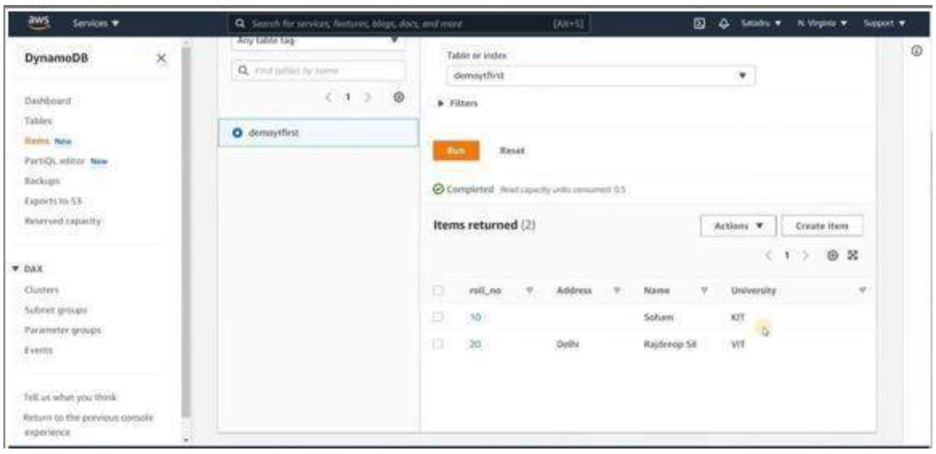


Table or index: demoytfirst

Completed Read capacity units consumed 0.5

Items returned (2)

roll_no	Address	Name	University
10		Soham	KIT
20	Delhi	Rajdeeop Sil	VIT

CONCLUSION : By this assignment we learn how to create a Lambda function using Python for adding data to Dynamo DB database.

NAME : SOUMIL SALVI

ROLL NO : 104

WRITTEN ASSIGNMENT - 1

***1*. what security measures can be taken while using Kubernetes?**

i. Role-Based Access Control (RBAC): RBAC restricts who can perform actions within a Kubernetes cluster. It defines roles and role bindings to specify what resources and operations users or service accounts can access. This prevents unauthorized access and actions within the cluster.

ii. Regular Updates: Keeping Kubernetes and its components up to date is crucial. New releases often include security patches. Regular updates help mitigate known vulnerabilities and ensure your cluster remains secure.

iii. Network Policies: Network policies allow you to define rules for communication between pods. By specifying which pods can communicate with each other, you can limit the attack surface and prevent unauthorized access.

iv. Container Security Tools: Employ container security tools like vulnerability scanners to assess the security of container images. These tools can identify and remediate vulnerabilities in the containerized applications before they are deployed.

v. Monitoring and Audit: Implement monitoring and auditing solutions to track cluster activity. This helps detect and respond to suspicious or unauthorized behavior. Tools like Prometheus and Grafana can be used for monitoring, while audit logs provide insights into cluster activity.

vi. Secrets Management: Sensitive data like API keys, passwords, and certificates should be stored securely using Kubernetes secrets or external vaults. This prevents sensitive information from being exposed within containers or configuration files.

vii. PodSecurityPolicies (PSP): PSP is a Kubernetes feature that enforces security policies at the pod level. It allows you to define restrictions on privilege escalation, host access, and other security-sensitive configurations for pods.

viii. Namespaces: Use Kubernetes namespaces to logically isolate workloads. This provides a level of separation between different applications or teams, reducing the risk of unauthorized access or interference between them.

ix. Admission Controllers: Admission controllers are webhook plugins that intercept and validate requests to the Kubernetes API server. You can use them to enforce custom policies and ensure that only compliant resources are admitted to the cluster.

x. Container Runtime Security: Implement container runtime security solutions like Docker Security Scanning or container runtime protection tools. These tools monitor containers at runtime for abnormal behavior, helping to detect and respond to potential threats.

***2*. What are the three security techniques that can be used to protect data?**

Three security techniques commonly used to protect data are:

i. Encryption: Encryption is the process of converting data into a secure format that can only be read by someone with the decryption key. It ensures that even if unauthorized parties access the data, they cannot understand it without the correct key. Two common types of encryption are:

   a. Data-at-rest Encryption: Protects data when it's stored on disk or in a database.

   b. Data-in-transit Encryption: Secures data as it's transmitted between systems over networks.

ii. Access Control: Access control mechanisms regulate who can access data and what actions they can perform on it. This involves setting permissions, roles, and policies to ensure that only authorized users or applications can access and manipulate data. Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) are commonly used access control models.

iii. Data Masking/Redaction: Data masking or redaction involves obscuring or replacing sensitive data with fictitious or scrambled values. This is often used in non-production environments or when sharing data with third parties. It ensures that even if someone gains access to the data, they cannot see the actual sensitive information.

These techniques are often used in combination to create a layered approach to data security, providing multiple levels of protection to safeguard sensitive information from unauthorized access and disclosure.

3. How do you expose a service using ingress in Kubernetes?

To expose a service using Ingress in Kubernetes, you need to follow these steps:

i. Set up Kubernetes: Ensure you have a Kubernetes cluster up and running, and you have the `kubectl` command-line tool configured to communicate with the cluster.

ii. Deploy Your Application: Deploy your application as a Kubernetes Deployment or a Pod, and create a Kubernetes Service to expose it internally within the cluster. This Service will be the target for the Ingress.

iii. Install an Ingress Controller: You need to have an Ingress controller installed in your cluster. Some popular options include Nginx Ingress Controller, Traefik, or HAProxy Ingress. The controller will manage the Ingress resources and configure the load balancer.

For example, to install the Nginx Ingress Controller, you can use:

```bash
```

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.0.0/deploy/static/provider/cloud/deploy.yaml
```

iv. Create an Ingress Resource: Define an Ingress resource that specifies the rules for routing traffic to your service. Here's an example Ingress resource manifest:

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
    - host: example.com
      http:
        paths:
          - path: /path
            pathType: Prefix
            backend:
              service:
                name: your-service
                port:
                  number: 80
```

In this example, traffic for `example.com/path` will be routed to `your-service`.

v. Apply the Ingress Resource: Use `kubectl apply` to create the Ingress resource in your cluster:

```bash
kubectl apply -f your-ingress.yaml
```

vi. Configure DNS: Ensure that the DNS records for the specified hostname (e.g., `example.com`) point to the external IP address of your Ingress controller.

vii. Access Your Service: After DNS propagation, you should be able to access your service externally via the hostname and path you defined in the Ingress resource.

_4._ Which service protocols does Kubernetes ingress expose?

Kubernetes Ingress is primarily designed to expose HTTP and HTTPS services, making it suitable for routing and load balancing web traffic. However, with the evolution of Kubernetes and Ingress controllers, it has expanded to support additional protocols and features:

i. **HTTP:** Ingress is commonly used to expose HTTP services. You can define routing rules based on URL paths, hostnames, and other HTTP attributes.

ii. **HTTPS:** Secure HTTP services can be exposed through Ingress by configuring TLS certificates. This allows you to terminate SSL/TLS encryption at the Ingress controller and route decrypted traffic to your services.

iii. **TCP:** Some Ingress controllers, like Nginx Ingress, support TCP services. This enables you to expose non-HTTP services such as databases or custom protocols. TCP-based routing typically relies on port numbers.

iv. **UDP:** While less common, some Ingress controllers support UDP services.

UDP is a connectionless protocol used for various purposes, including DNS and VoIP. Exposing UDP services may require specific controller support.

v. **gRPC:** If your services use the gRPC protocol, you can configure Ingress resources to handle gRPC traffic. gRPC is a high-performance RPC (Remote Procedure Call) framework often used for communication between microservices.

vi. **WebSocket:** Ingress controllers can be configured to support WebSocket connections. WebSocket is a protocol that enables full-duplex communication over a single TCP connection and is used for real-time applications.

vii. **Custom Protocols:** In some cases, you may need to expose services using custom or proprietary protocols. Depending on your Ingress controller and its capabilities, you might be able to configure it to handle these custom protocols.

Additionally, Ingress controllers often evolve, so it's essential to refer to the documentation and features of the specific controller you plan to use to ensure compatibility with your service protocols.

NAME : SOUMIL SALVI

ROLL NO : 104

WRITTEN ASSIGNMENT 2

Q1. How to deploy lambda function on AWS?

Deploying a Lambda function on AWS involves several steps. Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Here's a step-by-step guide on how to deploy a Lambda function:

Step 1: Create a Lambda Function

1. Log in to the AWS Management Console.

2. Navigate to the Lambda service by searching for it in the AWS Services section.

3. Click the "Create function" button.

Step 2: Configure Your Function

4. Choose "Author from scratch."

5. Fill in the basic information for your function, including the name, runtime, and execution role.

6. For "Execution role," you can create a new role from a template or use an existing role if you have one with the necessary permissions.

7. Click the "Create function" button.

Step 3: Upload Your Code

8. In the "Function code" section, you can upload your code either directly (ZIP file or folder) or by specifying a repository from AWS CodeCommit, Amazon S3, or other options.

9. Configure the handler if needed. The handler is the method that AWS Lambda invokes.

10. Set environment variables if your code requires them.

11. Adjust the runtime settings if necessary.

12. Click the "Deploy" button to upload your code.

Step 4: Define the Trigger

13. In the "Add triggers" section, you can specify event sources that will trigger your Lambda function. This can be an API Gateway, an S3 bucket, an SNS topic, etc.

14. Configure the trigger according to your needs and grant permissions as required.

Step 5: Test Your Function

15. You can test your Lambda function by creating a test event or using a sample test event provided by AWS.

16. Click the "Test" button to run your function and see the results.

Step 6: Monitor and Troubleshoot (Optional)

17. AWS Lambda provides various monitoring and logging options. You can configure CloudWatch alarms, inspect logs, and set up notifications to keep an eye on your function's performance.

Step 7: Save and Deploy the Function

18. After you've configured everything to your satisfaction, click the "Save" button to save your changes.

19. Click the "Deploy" button to make your function live and ready to respond to triggers.

Step 8: Invocation and Scaling

Your Lambda function is now deployed and ready to be invoked by the trigger you configured. AWS Lambda handles the scaling of resources based on the incoming workload automatically.

Q2. What are the deployment options for AWS Lambda?

AWS Lambda offers several deployment options:

1. Code Upload: You can directly upload your function code as a ZIP file or a deployment package when creating or updating your Lambda function.

2. AWS Lambda Layers: You can use Lambda Layers to separate your function code from its dependencies. This allows you to manage and version common libraries independently and share them across multiple functions.

3. **AWS SAM (Serverless Application Model):** AWS SAM is a framework for building serverless applications. You can define your Lambda functions and their associated resources in a SAM template file, then deploy the entire application using AWS SAM CLI.

4. AWS CloudFormation: You can use AWS CloudFormation templates to define and deploy Lambda functions along with other AWS resources. This enables infrastructure as code (IaC) for your serverless applications.

5. AWS Serverless Application Repository: You can publish and share serverless applications and Lambda functions using the AWS Serverless Application Repository. Users can deploy your applications directly from the repository.

6. AWS CodePipeline: You can integrate AWS Lambda deployment into a continuous integration/continuous deployment (CI/CD) pipeline using AWS CodePipeline. This automates the building, testing, and deployment of your Lambda functions.

7. Container Image Support: AWS Lambda now supports deploying functions as container images, allowing you to package your function and dependencies in a Docker container and deploy them as a Lambda function.

These deployment options provide flexibility and enable you to choose the one that best fits your application architecture and development workflow.

Q3 What are the 3 full deployment modes that can be used for AWS?

In the context of AWS Lambda, there are three primary deployment modes:

1. Automatic Deployment: AWS Lambda provides automatic deployment when you directly upload your function code as a ZIP file or specify a deployment package. This is the most common and straightforward deployment method, and it's suitable for most use cases.

2. Infrastructure as Code (IaC) with CloudFormation: AWS CloudFormation is a service that allows you to define and provision AWS infrastructure and resources in a template. You can use CloudFormation to define your Lambda functions, their associated resources, and any other AWS resources your application needs. When you create or update the CloudFormation stack, it deploys the Lambda functions and other resources defined in the template. This approach is more suitable for complex serverless applications and infrastructure orchestration.

3. Serverless Application Model (AWS SAM): AWS SAM is an open-source framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way to define serverless resources, including Lambda functions. You can use AWS SAM to define your functions and related resources in a SAM template, and then deploy the entire application using AWS SAM CLI. This approach is a balance between the simplicity of direct deployment and the power of CloudFormation for more complex applications.

These deployment modes offer different levels of control and abstraction, allowing you to choose the one that best matches your deployment needs and development workflow.

Q4. What are the 3 components of AWS Lambda?

AWS Lambda consists of three primary components:

1. Function Code: This is the core component of AWS Lambda. It's the code that you want to run in response to events. You can write your code in various supported programming languages (e.g., Node.js, Python, Java, C#, etc.). You can package your code along with its dependencies into a deployment package, which you upload to Lambda.

2. Event Sources: Event sources are triggers that invoke your Lambda function. AWS Lambda supports various event sources, such as Amazon S3, Amazon DynamoDB, Amazon SNS, AWS API Gateway, and more. When an event occurs in one of these services, it triggers the execution of your Lambda function.

3. Execution Environment: AWS Lambda manages the execution environment for your function. It automatically scales and provisions the necessary infrastructure to run your code. You don't need to worry about server provisioning or resource management. AWS Lambda also provides runtime support for various programming languages, so your code runs in an isolated environment with the necessary resources to execute.