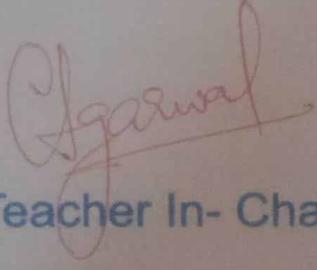


**Thadomal Shahani Engineering College**  
Bandra (W.), Mumbai - 400 050.

**CERTIFICATE**

Certify that Mr./Miss Piyush Hingorani  
of I.T. (T1) Department, Semester V with  
Roll No. 42 has completed a course of the necessary  
experiments in the subject Adv. DevOps LAB under my  
supervision in the **Thadomal Shahani Engineering College**  
Laboratory in the year 2023 - 2024

  
Teacher In-Charge

Head of the Department

Date 20/10/23

Principal

## CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1)	To study and perform setup of AWS EC2 service and launch EC2 Instance		18/7/23	
2)	To Study and perform the setup of AWS Cloud9 service and launch a python program in Cloud9		25/7/23	
3)	To study AWS S3 service and create bucket for hosting static web app.		18/8/23	
4)	To study AWS codepipeline and deploy web application using AWS codePipeline		8/8/23	
5)	To understand Kubernetes Cluster Architecture, install and Spinup Kubernetes cluster on Linux Machines/ Cloud Platform		13/10/23	
6)	To understand terraform lifecycle and to build, change and destroy AWS infrastructure using terraform.		22/8/23	✓ Futura 20/10/23
7)	To perform static analysis on python programs and sonarQube SAST process		29/8/23	
8)	To understand Continuous Monitoring using Nagios		12/9/23	
9)	To understand AWS Lambda function and Create a Lambda function using Python to log "An image has been added" message file is added to s3 bucket		5/9/23	
10)	To create lambda function using python for adding data to Dynamodb		5/9/23	
11)	Assignment -1		11/8/23	
12)	Assignment -2		13/10/23	

ROLL NUMBER: 42(T13)

NAME: PIYUSH HINGORANI

DATE: 18/07/23

## **ASSIGNMENT -1 (EC2)**

**AIM:** To Study and create AWS EC2 instance

**LO MAPPED:** LO1- To understand the fundamentals of Cloud Computing to be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.

### **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.

Here's a detailed explanation of an AWS EC2 instance:

#### 1. Virtual Server:

An EC2 instance is a virtualized computing environment that mimics a physical server. It runs an operating system of your choice and supports a wide range of applications, databases, and services.

#### 2. Scalability:

One of the key benefits of EC2 is its scalability. You can easily scale your instances horizontally (adding more instances) or vertically (resizing an instance) to meet changing workloads. This elasticity helps you optimize performance while controlling costs.

#### 3. Variety of Instance Types:

EC2 provides a variety of instance types optimized for different use cases. These instance types vary in terms of CPU, memory, storage, and network capabilities, allowing you to choose the best fit for your application's requirements.

#### **4. Amazon Machine Image (AMI):**

An AMI is a pre-configured template that contains the information required to launch an instance. It includes the operating system, application software, and any additional configuration you've applied. You can choose from a wide range of public AMIs or create your own custom AMIs.

#### **5. Networking and Security:**

Each EC2 instance is associated with a security group, which acts as a virtual firewall controlling inbound and outbound traffic. You can also assign Elastic IP addresses for consistent IP assignments, create Virtual Private Clouds (VPCs) for isolated networking, and configure network settings like subnets, routing tables, and network access control lists.

#### **6. Cost Flexibility:**

EC2 instances are available in different pricing models, including On-Demand (pay-as-you-go), Reserved Instances (upfront payment for long-term usage), and Spot Instances (bid-based pricing). This allows you to choose the most cost-effective option for your workload.

#### **STEPS:-**

LOGIN TO AWS ACCOUNT,

THEN SEARCH EC2.

The screenshot shows the AWS Console Home page. At the top, there's a search bar and navigation links for Services, Mumbai, and Piyush.Hingorani. Below the header, there are two main sections: 'Recently visited' (listing EC2, Billing, and AWS Budgets) and 'Welcome to AWS' (with links to Getting started with AWS, Training and certification, and What's new with AWS). In the middle, there's a summary card for AWS Health showing 0 open issues over the past 7 days. To the right, there's a 'Cost and usage' summary showing current month costs at \$0.00 and top costs for the current month. At the bottom, there are links for CloudShell, Feedback, Language, and cookie preferences.

NOW CLICK ON LAUNCH / CREATE NEW INSTANCES.

The screenshot shows the AWS EC2 Management Dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area has a 'Resources' section showing counts for various EC2 components like Instances (running), Auto Scaling Groups, Dedicated Hosts, etc. It also features a callout for launching Microsoft SQL Server Always On availability groups. Below this is a 'Launch instance' section with a prominent orange 'Launch instance' button and a 'Service health' section indicating normal status. On the right, there's an 'Account attributes' panel for the Default VPC and Settings, and an 'Explore AWS' panel with sections for price performance, cost reduction tips, and more.

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

**Name and tags**

Name: Piyush H

**Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group  
Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per

**Launch instance**

Click on T2 micro (free tier one)

**Application and OS Images (Amazon Machine Image)**

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

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

**Quick Start**

Amazon, macOS, Ubuntu, Windows, Red Hat, SUSE Li

**Amazon Machine Image (AMI)**

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
ami-0f5ee92e2d63afc18 (64-bit (x86)) / ami-077053fb4029de92f (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

**Description**

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

Architecture: 64-bit (x86) AMI ID: ami-0f5ee92e2d63afc18 Verified provider

**Summary**

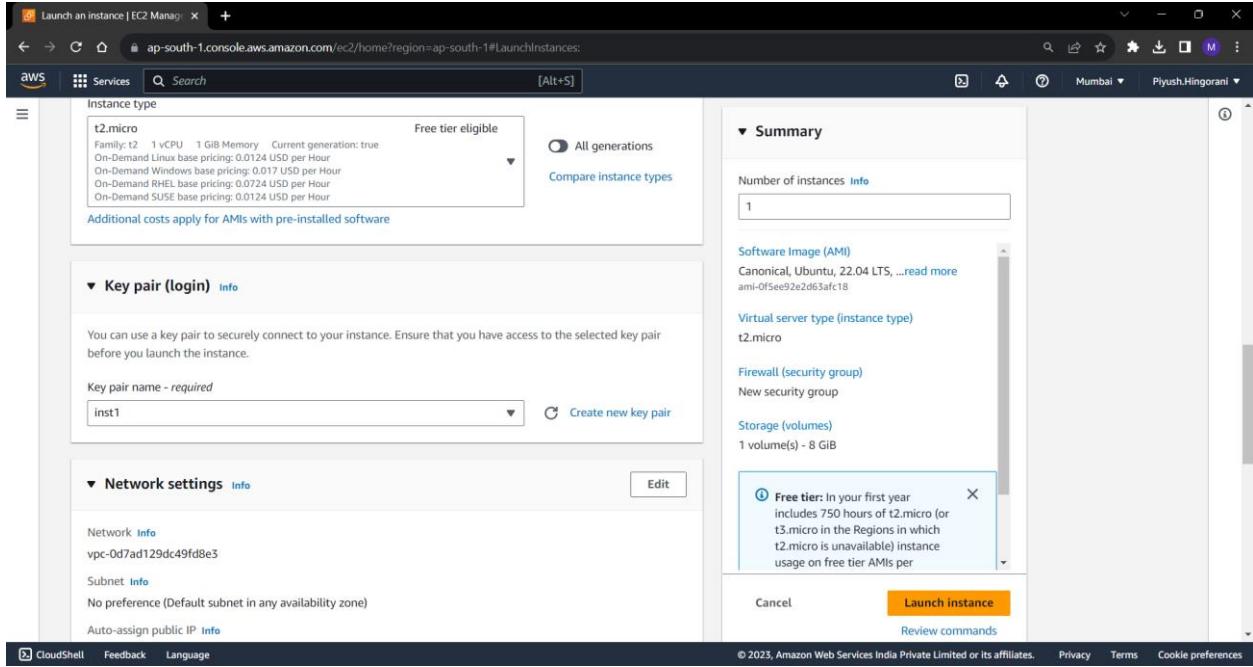
Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group  
Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per

**Launch instance**

Click on NEXT, then Again Click Next.

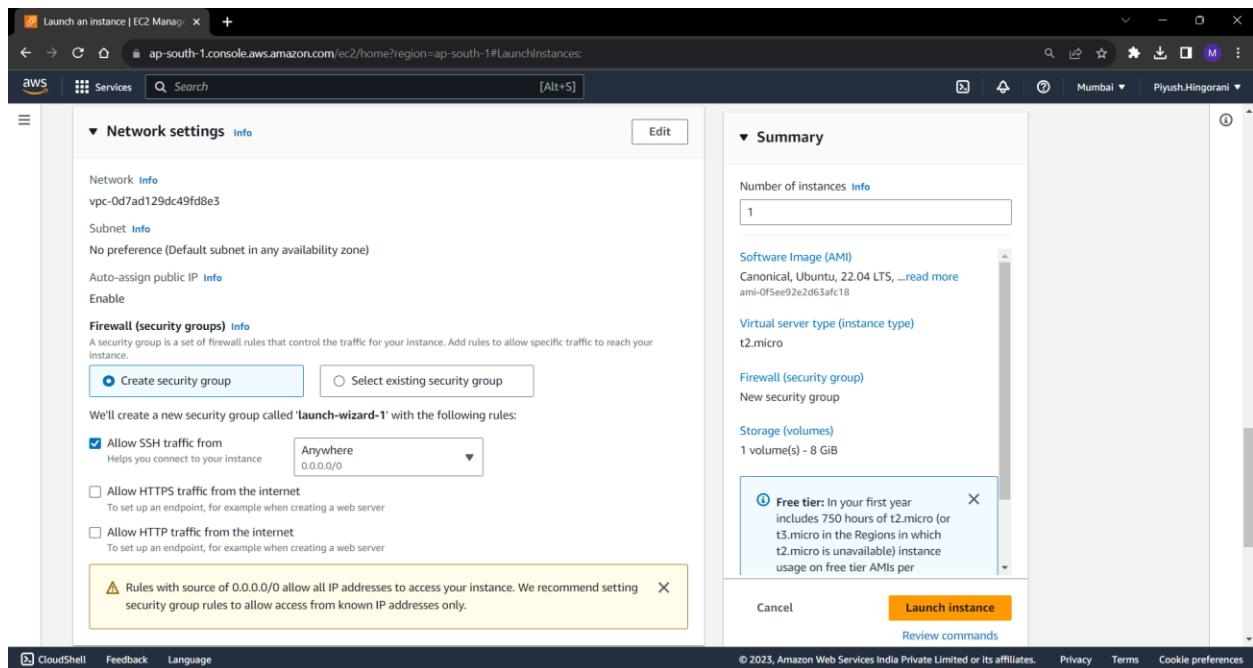


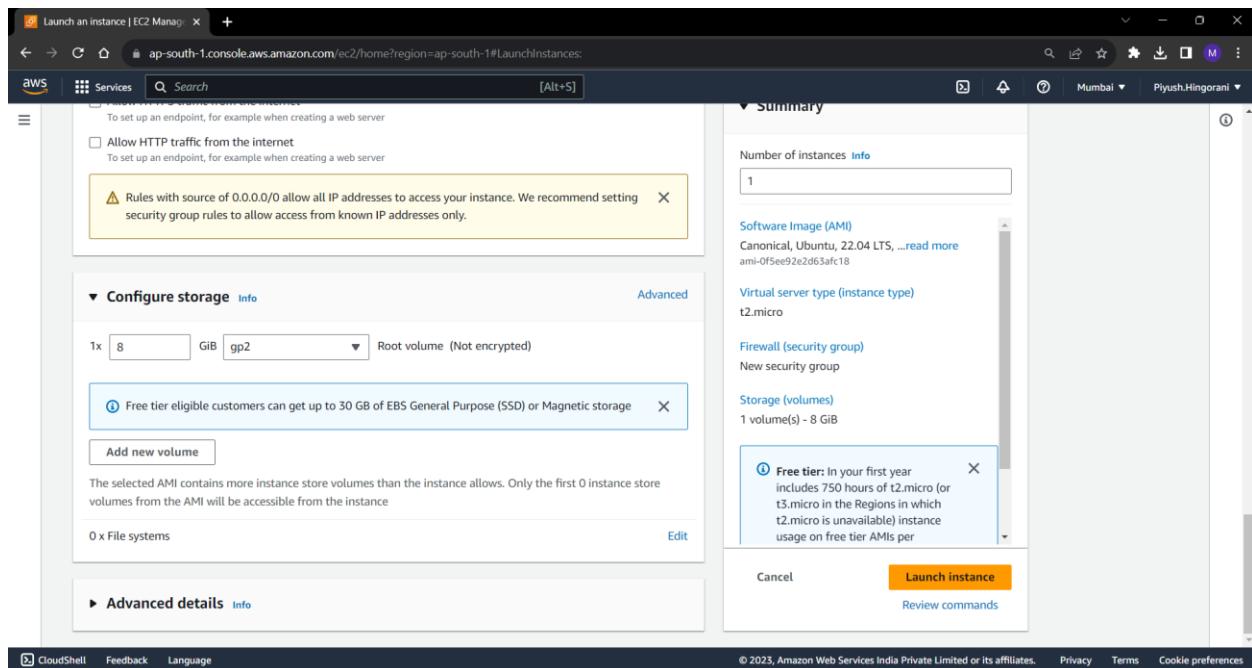
THEN CREATE A KEY PAIR BY ANY NAME AND DOWNLOAD IT. THEN CLICK NEXT

Now add security group ALL TRAFFIC ,

PROTOCOL – ALL,

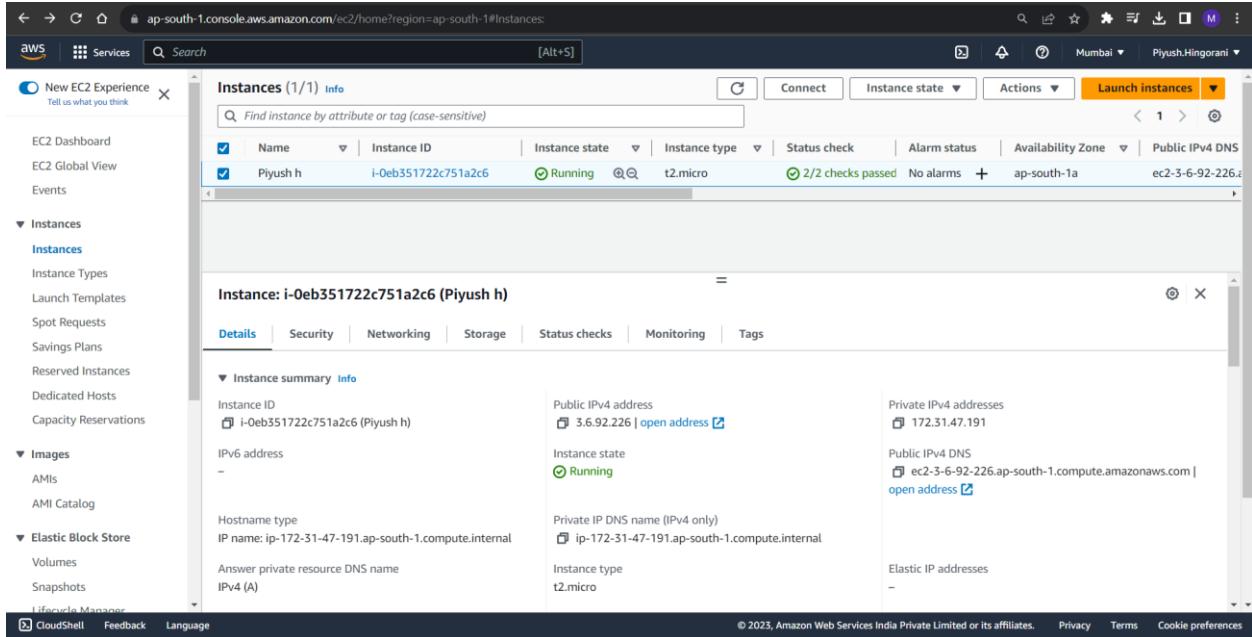
SOURCE- ANYWHERE.



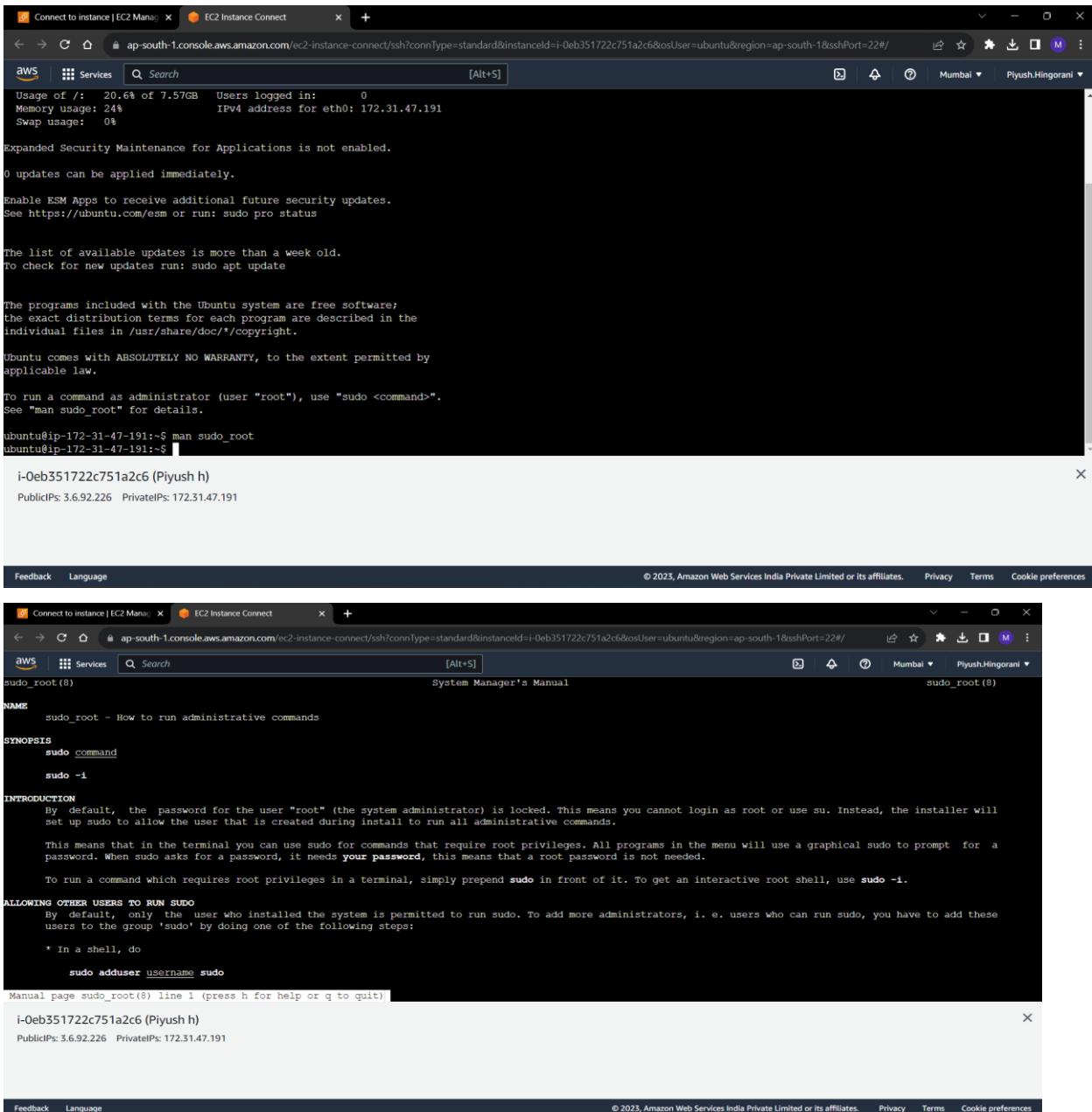


NOW WAIT TILL THE STATUS CHECK IS 2/2 and Instance is running.

Once Check is complete click on launch instances.



## FOLLOW SOME BASIC LINUX COMMANDS AS SHOWN BELOW-



The screenshot shows two terminal windows from an EC2 Instance Connect session. The top window displays system status and update information:

```
Usage of /: 20.6% of 7.57GB Users logged in: 0
Memory usage: 24% IPv4 address for eth0: 172.31.47.191
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
see https://ubuntu.com/esm or run: sudo pro status

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-172-31-47-191:~$ man sudo_root
ubuntu@ip-172-31-47-191:~$
```

The bottom window shows the output of the `man sudo_root` command:

```
i-0eb351722c751a2c6 (Piyush h)
PublicIPs: 3.6.92.226 PrivateIPs: 172.31.47.191

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

```
sudo_root(8)                               System Manager's Manual
sudo_root - How to run administrative commands

NAME
    sudo_root - How to run administrative commands

SYNOPSIS
    sudo command
    sudo -i

INTRODUCTION
    By default, the password for the user "root" (the system administrator) is locked. This means you cannot login as root or use su. Instead, the installer will set up sudo to allow the user that is created during install to run all administrative commands.

    This means that in the terminal you can use sudo for commands that require root privileges. All programs in the menu will use a graphical sudo to prompt for a password. When sudo asks for a password, it needs your password, this means that a root password is not needed.

    To run a command which requires root privileges in a terminal, simply prepend sudo in front of it. To get an interactive root shell, use sudo -i.

ALLOWING OTHER USERS TO RUN SUDO
    By default, only the user who installed the system is permitted to run sudo. To add more administrators, i. e. users who can run sudo, you have to add these users to the group 'sudo' by doing one of the following steps:
        * In a shell, do
            sudo adduser username sudo

Manual page sudo_root(8) line 1 (press h for help or q to quit)
i-0eb351722c751a2c6 (Piyush h)
PublicIPs: 3.6.92.226 PrivateIPs: 172.31.47.191

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

## CONCLUSION:

In summary, Amazon EC2 instances provide flexible and scalable compute capacity in the cloud. They empower developers and businesses to deploy, manage, and scale applications without the need for physical hardware, offering agility, cost-efficiency, and a wide range of configuration options. Hence, learned and implemented the steps to create an EC2 machine.

**ROLL NO: 42 (T13)**

**NAME: PIYUSH HINGORANI**

**DATE: 25/07/23**

### **ASSIGNMENT-2 (CLOUD9)**

**AIM:** To Study and create AWS Cloud 9 IDE service.

**LO MAPPED:** LO1

#### **THEORY:**

Cloud9 IDE is an Online IDE, published as open source from version 2.0, until version 3.0. It supports multiple programming languages, including C, C++, PHP, Ruby, Perl, Python, JavaScript with Node.js, and Go. It is written almost entirely in JavaScript, and uses Node.js on the back-end.

#### **Features and Capabilities:**

- Code Editing: Cloud9 provides a feature-rich code editor with syntax highlighting, auto-completion, and code folding for multiple programming languages.
- Collaborative Coding: Developers can collaborate in real-time by sharing their Cloud9 environment with teammates, making it easy to work on projects together.
- Code Debugging: Integrated debugging tools help identify and fix issues in your code.
- Terminal Access: Cloud9 includes a terminal for running command-line tasks within the IDE.
- Version Control Integration: It seamlessly integrates with version control systems like Git, allowing for efficient code version management.
- Built-in Runners: Developers can run applications directly from the IDE using built-in runners for various languages.

- **Integrations:** Cloud9 can be integrated with other AWS services like Lambda, EC2, and more, making it a powerful tool for cloud development.
- **Extensibility:** You can install and configure additional software, libraries, and tools to customize your development environment.

**Accessibility and Availability:** Since Cloud9 is web-based, developers can access their environments from anywhere using a web browser. This accessibility is valuable for remote work, team collaboration, and on-the-go development.

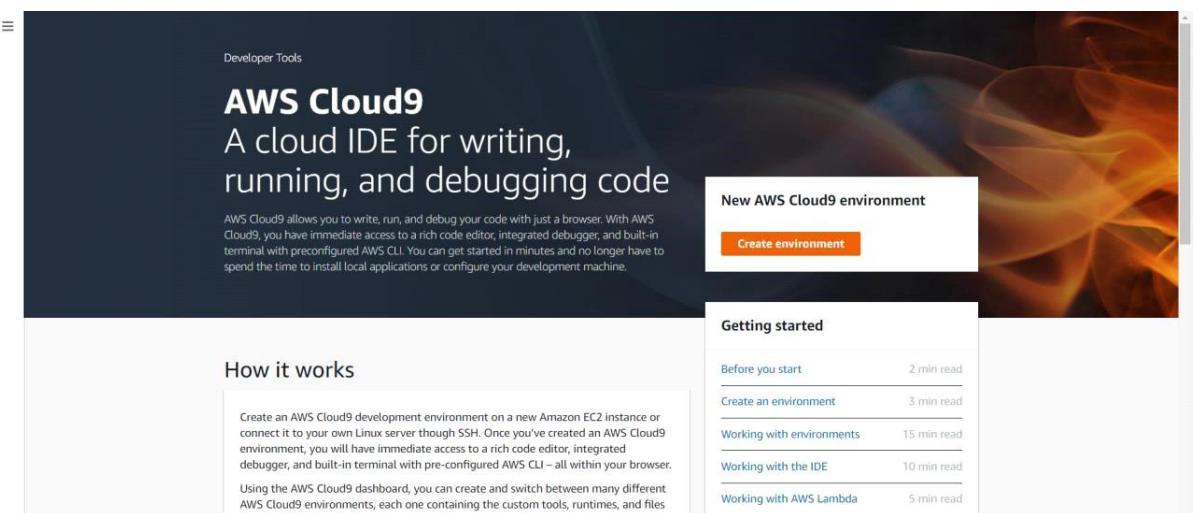
**AWS Integration:** Cloud9 is tightly integrated with other AWS services, enabling seamless development and deployment workflows. Developers can easily connect to AWS resources, such as databases, Lambda functions, and S3 buckets, directly from the IDE.

**On-Demand Resources:** Cloud9 provides on-demand resources, so developers don't need to worry about configuring or maintaining the underlying infrastructure.

### **STEPS:-**

1)LOG IN TO YOUR AWS ACCOUNT,

SEARCH FOR CLOUD 9 IN THE SEARCH BAR



## 2)CLICK ON CREATE ENVIRONMNET,

NAME THE ENVIRONMNET

Create environment [Info](#)

**Details**

Name  
Piyush.Hingorani

Description - optional  
my first cloud environment

Environment type [Info](#)  
Determines what the Cloud9 IDE will run on.

New EC2 instance  
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute  
You have an existing instance or server that you'd like to use.

New EC2 instance

Instance type [Info](#)  
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)  
Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)  
Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)  
Recommended for production and most general-purpose development.

Platform [Info](#)  
This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Amazon Linux 2

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

30 minutes

### 3) NOW CLICK ON CREATE ENVIRONMENT

The screenshot shows the 'Network settings' step of the AWS Cloud9 'Create Environment' wizard. It includes sections for 'Connection' (AWS Systems Manager (SSM) selected), 'Tags - optional', and a note about IAM resource creation. Buttons for 'Cancel' and 'Create' are at the bottom.

**Connection**  
How your environment is accessed.

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

Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

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

**The following IAM resources will be created in your account**

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

Cancel **Create**

The screenshot shows the AWS Cloud9 environment list after creating a new environment named 'Piyush.Hingorani'. The table lists the environment details, including Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN.

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Piyush.Hingorani	<a href="#">Open</a>	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:iam::644228993852:root

4) NOW SELECT ANY CODING LANGUAGE AND PERFORM ANY OPERATION via A CODE. SHOWN BELOW

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. The title bar indicates the URL is ap-south-1.console.aws.amazon.com/cloud9/ide/0e15f93cf8a4ebf8ece8fc7c0f493f4. The left sidebar shows a file tree with Pyush Hingorani, Armstrong.py, and README.md. The main editor window displays the following Python code:

```
print("Enter the Number: ")
num = int(input())
temp = num
noOfDigit = 0
res = 0
while num > 0:
    num = int(num/10)
    noOfDigit = noOfDigit+1
num = temp
while num > 0:
    rem = num%10
    i = 1
    i = 0
    while i<noOfDigit:
        pow = pow*rem
        i = i+1
    res = res+pow
    num = int(num/10)
if res==temp:
    print("\nThe number is an Armstrong Number")
else:
    print("\nThe number is not an Armstrong Number")
```

The bottom terminal window shows the output of running the script with the input 370, which correctly identifies it as an Armstrong number.

## CONCLUSION:

In summary, AWS Cloud9 IDE offers a cloud-based, feature-rich, and flexible environment for coding, debugging, and collaborating on projects. Its integration with AWS service makes it a powerful tool for developing and deploying cloud applications. Hence, learned and implemented steps to Create an Cloud9 environment.

**Roll No: 42**

**Batch: T13**

**Name: Piyush Hingorani**

**Date: 01/08/2023**

### **ASSIGNMENT-3**

**AIM-** To study AWS S3 service and create a bucket for housing static web application.

#### **THEORY-**

AWS Simple Storage Service (S3) from the aforementioned list, S3, is the object storage service provided by AWS. It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services.

An Amazon S3 bucket can be set up to operate similarly to a website. This section illustrates how to host a website using Amazon S3. There are mainly 7 steps to hosting a static website using Amazon Web Service(AWS) S3.

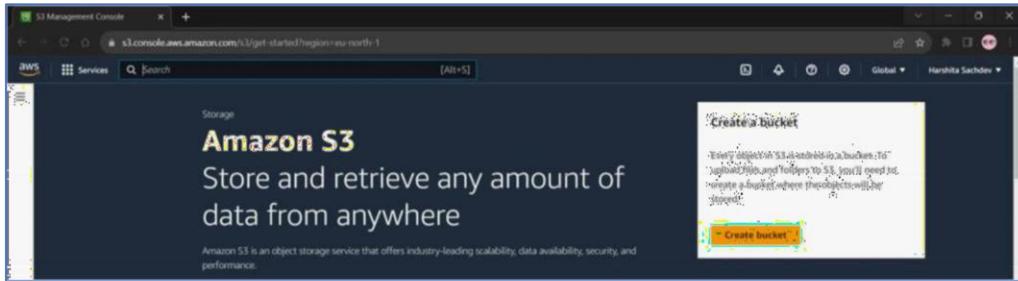
#### **STEPS:**

##### **Step 1: Creating a Bucket**

1. First, we have to launch our S3 instance. Follow these steps for creating a Bucket
2. Open the Amazon S3 console by logging into the AWS Management Console at

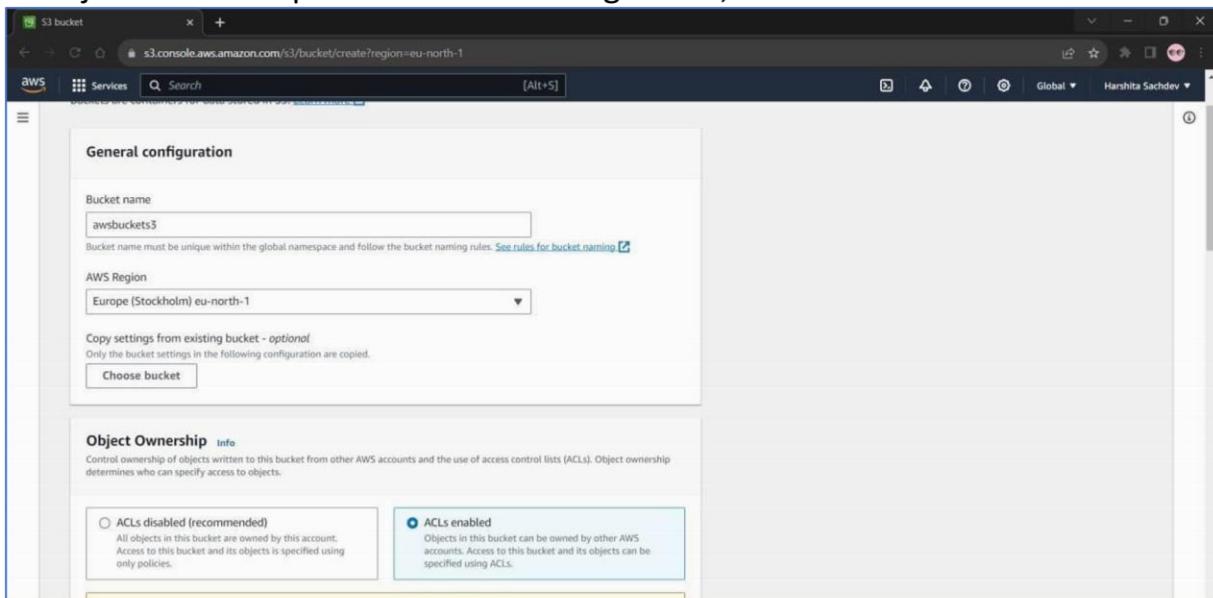
<https://console.aws.amazon.com/s3/>.

3. Click on Create Bucket.



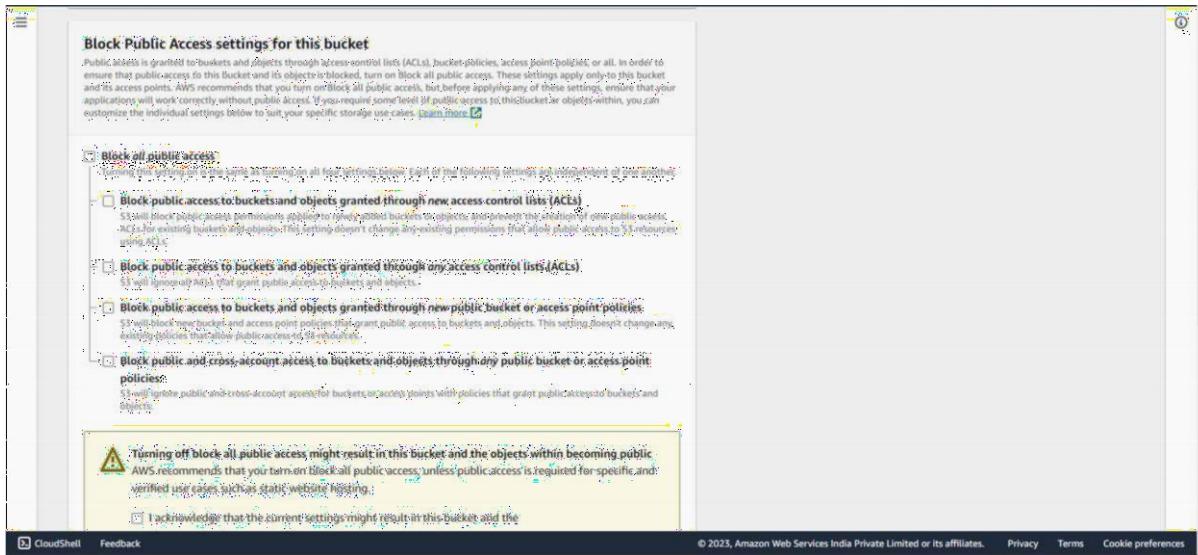
4. Choose Bucket Name – Bucket Name Should be Unique

5. Object Ownership – Enable for making Public, Otherwise disable



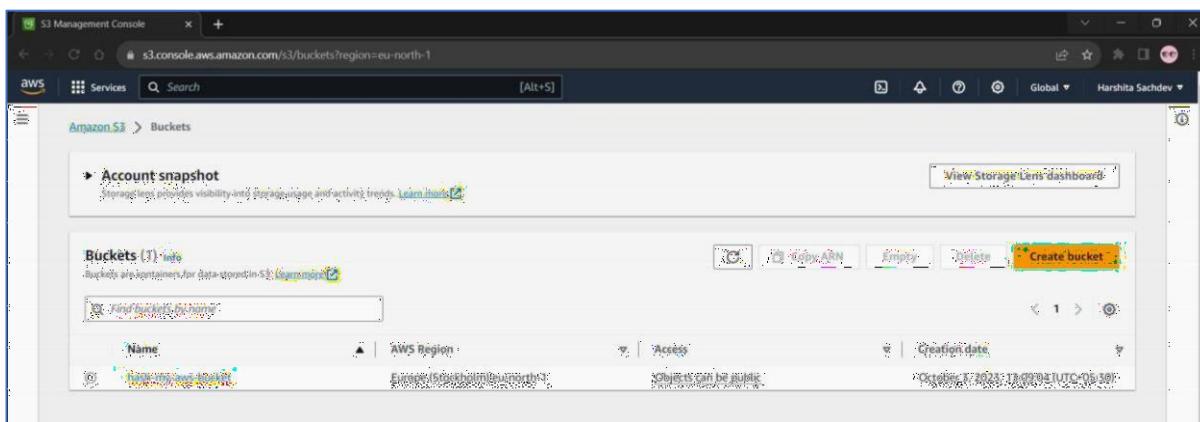
## Step 2: Block Public Access settings for the bucket

1. Uncheck (Block all public access) for the public, otherwise set default. If you uncheck (Block all public keys).



2. Now click on create bucket

3. Bucket is created



### Step 3: Now upload code files

Select Bucket and Click your Bucket Name.

Now, click on upload (then click add File/folder) and select your HTML code file from your PC/Laptop.

The screenshot shows the AWS S3 Management Console with a green header bar indicating 'Upload succeeded'. Below it, a summary table shows the destination bucket 's3://hash-my-aws-bucket' with 'Succeeded' status for 2 files (3.5 KB) and 'Failed' status for 0 files (0 B). The 'Files and folders' tab is selected, displaying a table with two items: 'aws.png' (image/png, 3.1 KB, Succeeded) and 'main.html' (text/html, 359.0 B, Succeeded).

#### **Step 4: Once the Files are uploaded successfully, click on Permissions and now follow this Process –**

- Block public access
- Object Ownership
- Make public Object

The screenshot shows the AWS S3 Bucket 'hash-my-aws-bucket' with two objects listed: 'aws.png' (image/png, 3.1 KB, Standard storage class) and 'main.html' (text/html, 359.0 B, Standard storage class). The 'Actions' menu for 'aws.png' is open, showing options like 'Copy', 'Move', 'Get object URL', 'Edit actions', and 'Make public using ACL'.

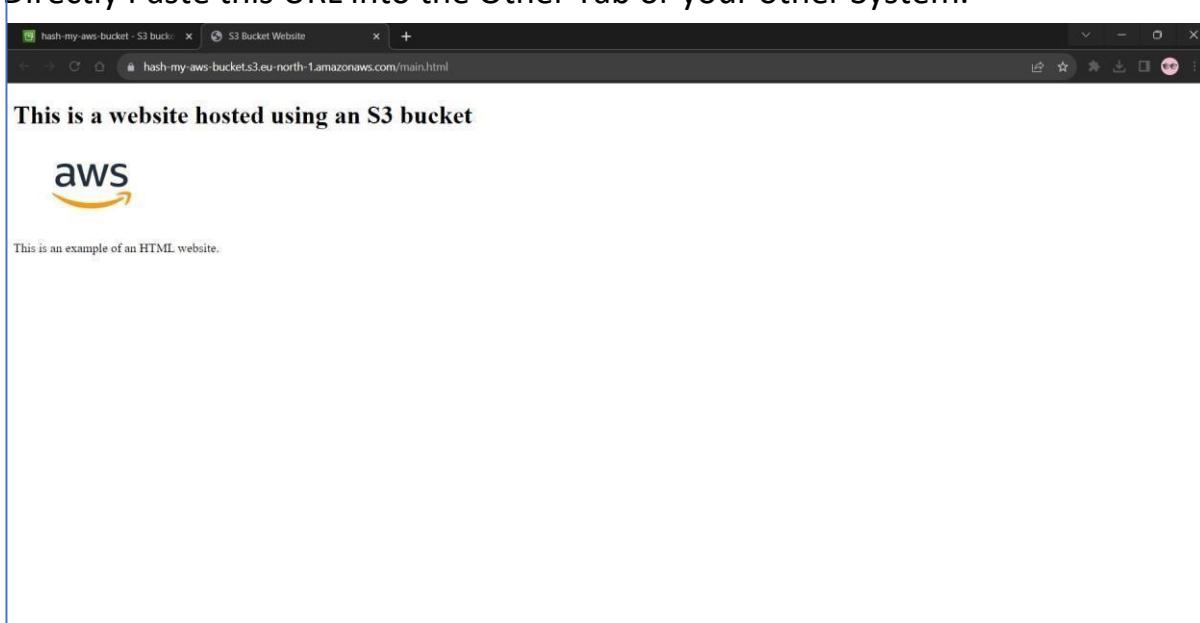
## **Step 5: Copy your Object URL**

Now, click on your HTML File Object Name.

Copy the Object URL.

## **Step 6: Check out your Website!**

Directly Paste this URL into the Other Tab or your other System.



**CONCLUSION:** This experiment demonstrated how to utilize AWS S3, a powerful and Scalable cloud storage solution, to host a static web application. S3's ability to serve static content with low latency and high reliability makes it a suitable choice for hosting static websites. By completing this experiment, we gained practical insights into leveraging cloud services for web hosting, which is vital for modern web development and deployment. AWS S3 offers a costeffective and efficient solution for hosting various types of web applications, contributing to the agility and scalability of web development projects.

**Roll No:-42**  
**Batch:- T13**  
**Name :-Piyush Hingorani**  
**Date :- 08/08/2023**

## **ASSIGNMENT-4**

**AIM:** To study AWS CodePipeline and deploy web application using Code

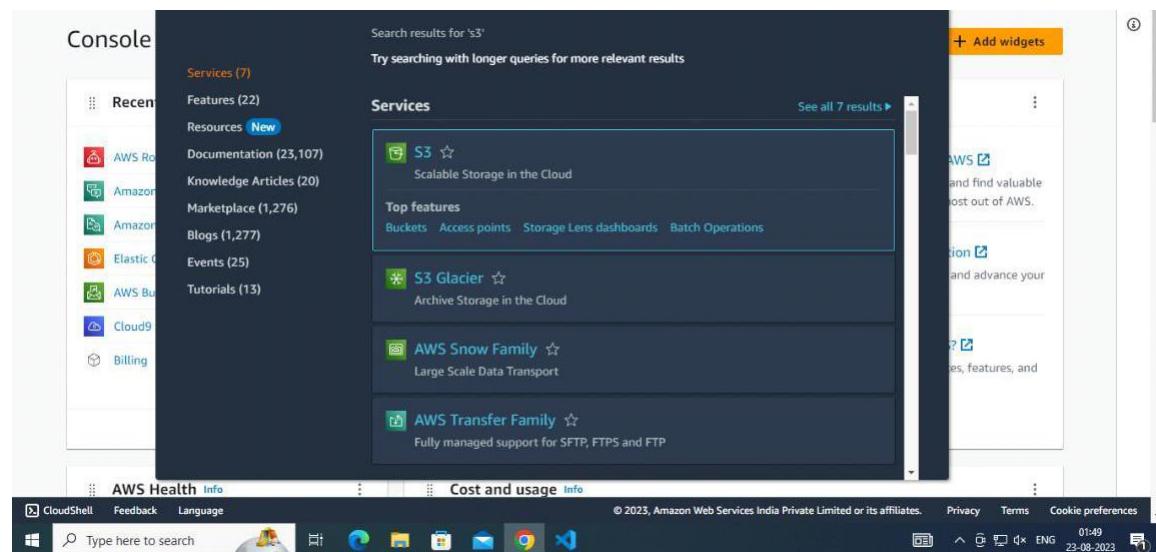
**LO MAPPED: LO1 , LO2**

### **THEORY:**

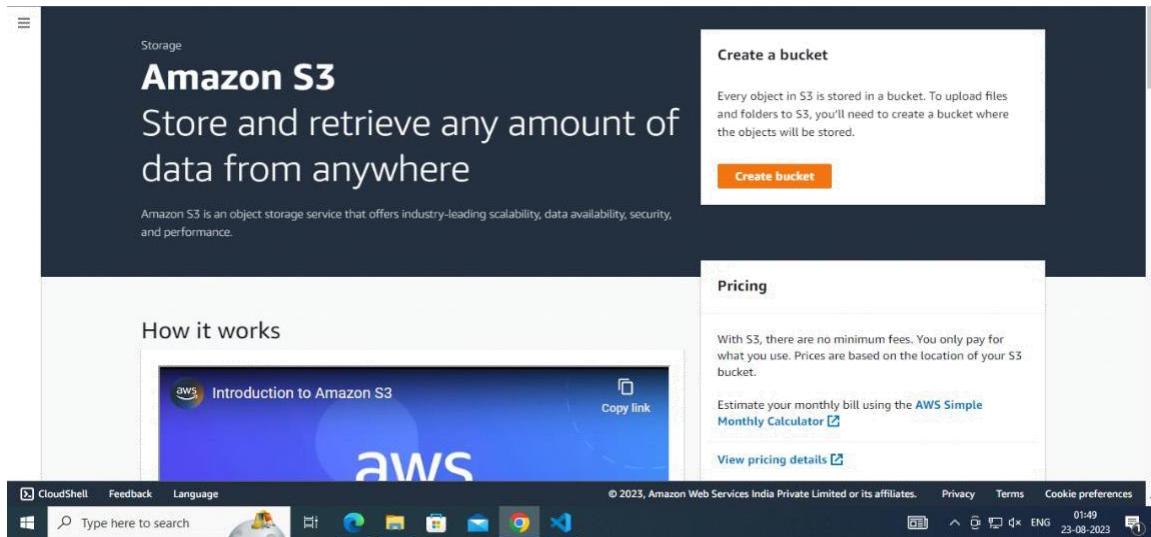
Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.

### **STEPS:**

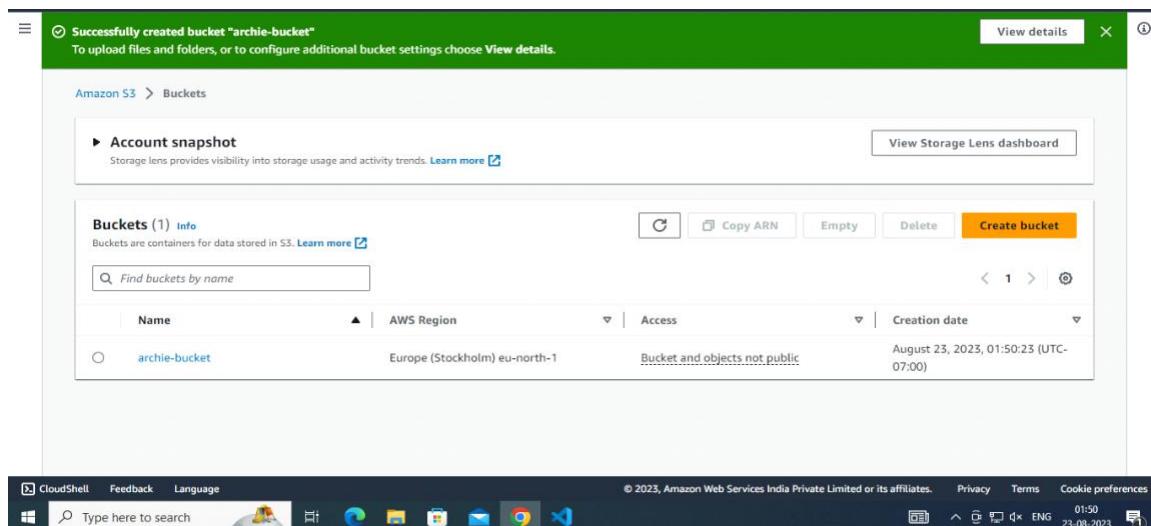
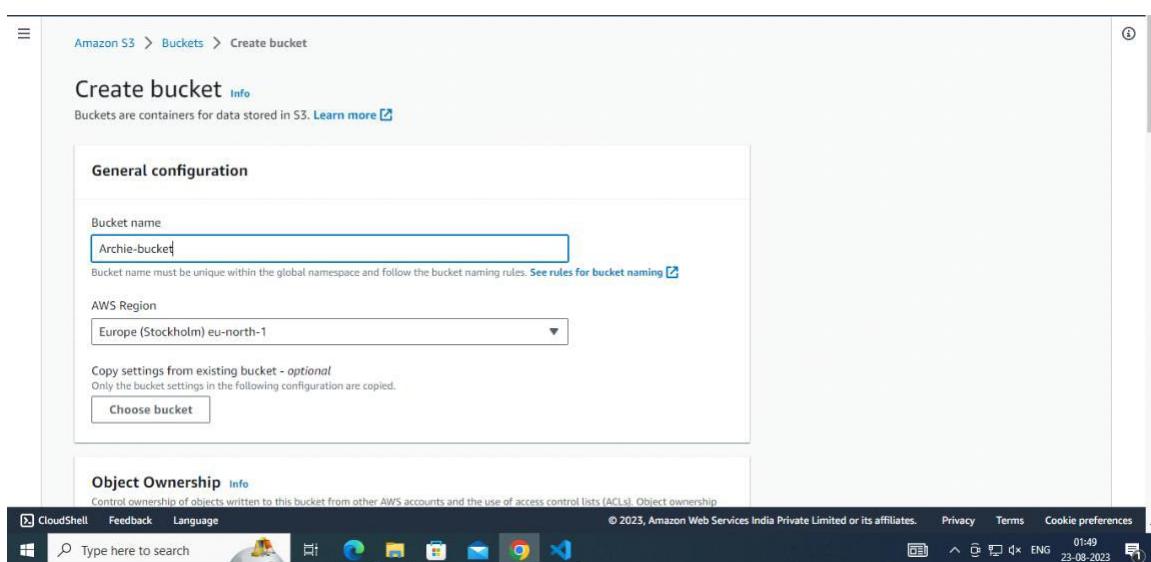
#### **1. Log in as IAM User. Search for S3 in console**



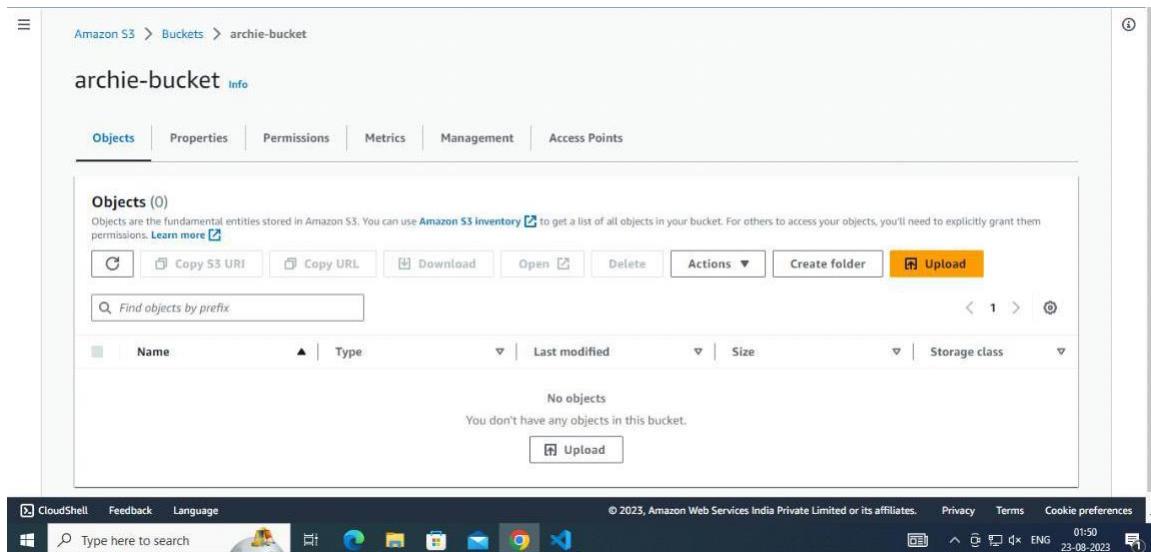
## 2. Create a new bucket by clicking Create Bucket Button



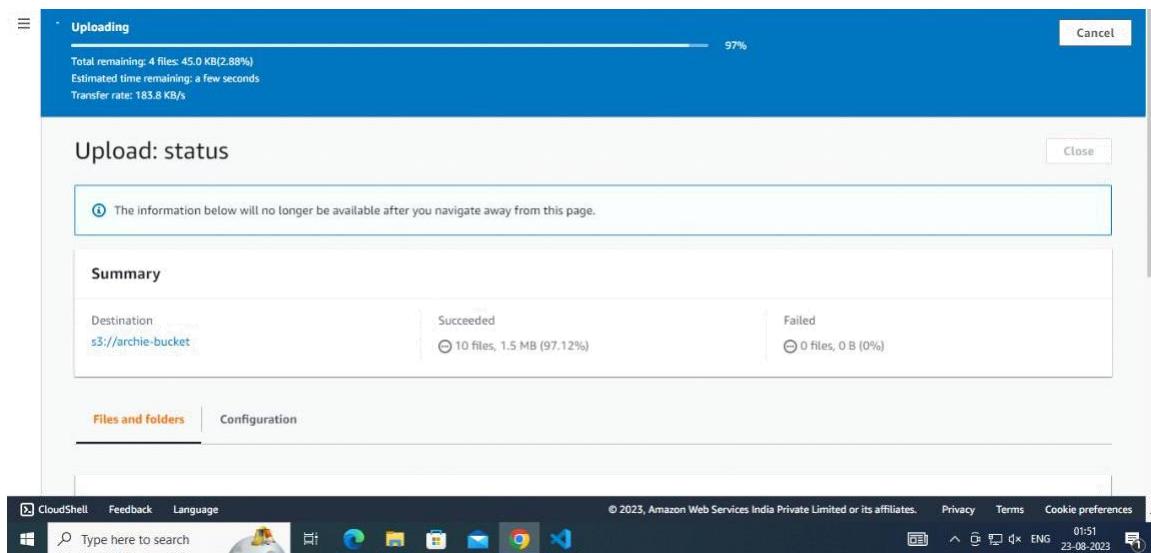
## 3. Give the bucket a name and click on Create Bucket to make a new bucket



#### 4. Click on Upload button to add files and folders of your projects.



The screenshot shows the Amazon S3 console interface. At the top, the navigation bar includes 'Amazon S3 > Buckets > archie-bucket'. Below it, the bucket name 'archie-bucket' is displayed with a 'Info' link. A horizontal menu bar contains 'Objects' (which is selected), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The main area is titled 'Objects (0)' and contains a message: '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](#)'. Below this are several action buttons: 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and the 'Upload' button, which is highlighted with a yellow background. A search bar 'Find objects by prefix' is present, along with a page navigation bar showing '1' of '1' pages. A table header row includes columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message 'No objects' indicates there are no files in the bucket. At the bottom of the main area is another 'Upload' button. The footer of the browser window shows standard links like CloudShell, Feedback, Language, and a search bar. On the right side of the footer, it says '© 2023, Amazon Web Services India Private Limited or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'. The date '23-08-2023' and time '01:50' are also shown.



The screenshot shows an 'Uploading' progress dialog. At the top, it displays 'Uploading' with a progress bar at 97%. Below the progress bar, status information is provided: 'Total remaining: 4 files: 45.0 KB(2.88%)', 'Estimated time remaining: a few seconds', and 'Transfer rate: 183.8 KB/s'. In the center, a summary table for the upload is shown:

Destination	Succeeded	Failed
s3://archie-bucket	10 files, 1.5 MB (97.12%)	0 files, 0 B (0%)

At the bottom of the dialog, there are tabs for 'Files and folders' (which is selected) and 'Configuration'. The footer of the browser window is identical to the one in the previous screenshot, showing the same links and system status.

**5.Go to the index.html folder and copy the Object URL. Copy this URL to another web page, it will show error.**

The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with navigation links like 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. Below that is a section for 'Block Public Access settings for this account'. Under 'Storage Lens', there are 'Dashboards' and 'AWS Organizations settings'. A 'Feature spotlight' section is also present. The main area shows a breadcrumb path: 'Amazon S3 > Buckets > archie-bucket > index.html'. The object name 'index.html' is highlighted in blue. Below the path are buttons for 'Copy S3 URI', 'Download', 'Open', and 'Object actions'. A 'Properties' tab is selected, showing details under 'Object overview'. The object details are as follows:

Attribute	Value
Owner	69c0afe2e8cc19c68d4eb4e4b802a26ff33f8e21d09f76c6bb33316053ffbc02
AWS Region	Europe (Stockholm) eu-north-1
Last modified	August 23, 2023, 01:51:09 (UTC-07:00)
Size	5.5 KB
Type	
S3 URI	<a href="s3://archie-bucket/index.html">s3://archie-bucket/index.html</a>
Amazon Resource Name (ARN)	<a href="#">arn:aws:s3:::archie-bucket/index.html</a>
Entity tag (Etag)	<a href="#">70440bb05fb044e018fb20b1af497f8</a>
Object URL	<a href="https://archie-bucket.s3.eu-north-1.amazonaws.com/index.html">https://archie-bucket.s3.eu-north-1.amazonaws.com/index.html</a>

The bottom of the screen shows a Windows taskbar with icons for CloudShell, Feedback, Language, a search bar, and several open application windows. The status bar at the bottom right indicates the time as 01:51 and the date as 23-08-2023.

## 6. Now we need to enable permissions. Click on Permission tab.

The screenshot shows the AWS S3 console with the 'Objects' tab selected. A green banner at the top indicates 'Upload succeeded'. Below the banner, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Permissions' tab is highlighted. The main area displays 'Objects (3)'. A table lists the objects: 'images/' (Folder), 'index.html' (html), and 'style.css' (css). The table includes columns for Name, Type, Last modified, Size, and Storage class. At the bottom of the table, there are navigation arrows and a refresh icon.

## 7. Change the public access permission.

The screenshot shows the AWS S3 console with the 'Permissions' tab selected. A green banner at the top indicates 'Upload succeeded'. Below the banner, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Permissions' tab is highlighted. The main area displays 'Permissions overview' and 'Access' settings, which show 'Bucket and objects not public'. Below this, the 'Block public access (bucket settings)' section is expanded. It contains a note about blocking public access and a 'Edit' button. Under 'Block all public access', the status is set to 'On'. There is also a link to 'Individual Block Public Access settings for this bucket'. At the bottom of the page, there is a navigation bar with links for CloudShell, Feedback, Language, and search, along with system status information.

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

**Block public access (bucket settings)**

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

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

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

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

Windows Type here to search ⓘ 01:52 23-08-2023

## 8. Go to the Object Ownership and Enable the ACLs

☰ Successfully edited Block Public Access settings for this bucket. ⓘ

**Object Ownership** [Info](#) ⓘ

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

**Object Ownership**  
**Bucket owner enforced**  
ACLs are disabled. All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**Edit**

**Access control list (ACL)** ⓘ

Grant basic read/write permissions to other AWS accounts. [Learn more](#) ⓘ

**Info** This bucket has the bucket owner enforced setting applied for Object Ownership  
When bucket owner enforced is applied, use bucket policies to control access. [Learn more](#) ⓘ

Grantee Objects Bucket ACL ⓘ

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

Windows Type here to search ⓘ 01:53 23-08-2023

**Object Ownership**

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

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

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

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

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

I acknowledge that ACLs will be restored.

Object Ownership

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 01:53 23-08-2023

## 9.Finally select all the files and folders and click on Actions and click on 'Make public using ACL'.

Amazon S3 > Buckets > archie-bucket

**archie-bucket** [Info](#)

Objects (3)

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](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Find objects by prefix

<input checked="" type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/>	images/	Folder	-	-	-
<input checked="" type="checkbox"/>	index.html	html	August 23, 2023, 01:51:09 (UTC-07:00)	5.5 KB	Standard
<input checked="" type="checkbox"/>	style.css	css	August 23, 2023, 01:51:09 (UTC-07:00)	904.0 B	Standard

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 01:54 23-08-2023

Amazon S3 > Buckets > archie-bucket

archie-bucket [Info](#)

Objects (3)

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](#)

Actions ▾ [Create folder](#) [Upload](#)

Name	Type	Last modified
images/	Folder	-
index.html	html	August 23, 2023, 01:51:09 (UTC-07:00)
style.css	css	August 23, 2023, 01:51:09 (UTC-07:00)

Storage class

5 KB Standard

1.0 B Standard

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

Type here to search 01:54 23-08-2023

## 10. Finally click on the 'Make Public' button and reload the web page

Make public [Info](#)

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

⚠ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

This action applies to all objects within the specified folders. Objects added to these folders while the action is in progress might be affected.

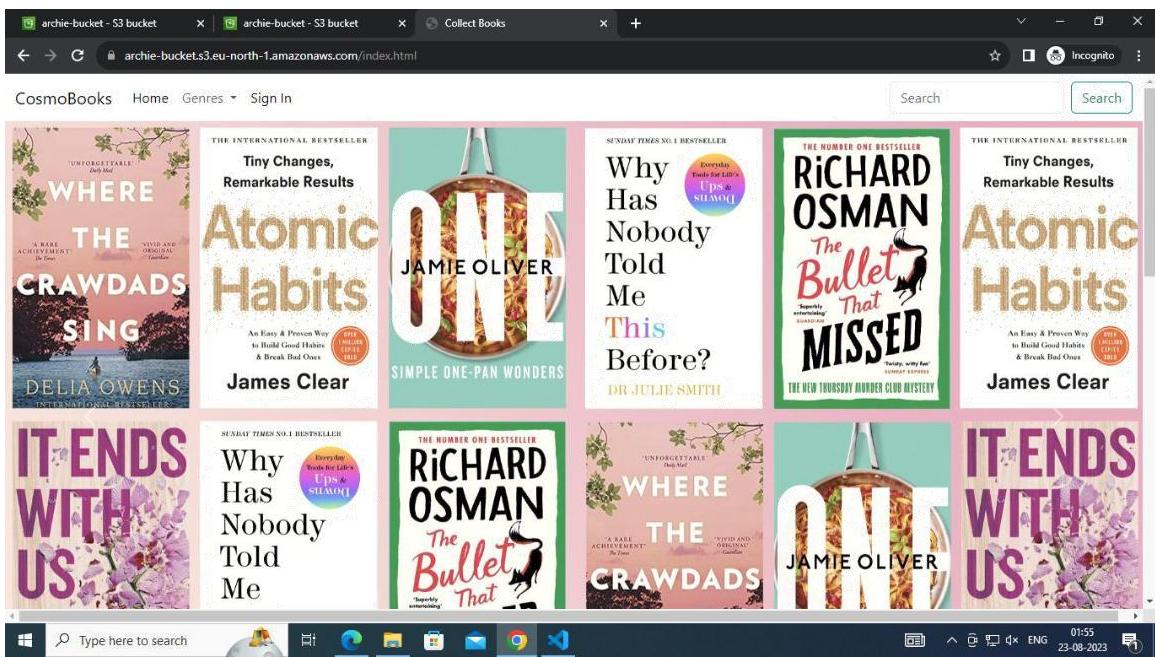
Specified objects

Name	Type	Last modified	Size
images/	Folder	-	-
index.html	html	August 23, 2023, 01:51:09 (UTC-07:00)	5.5 KB
style.css	css	August 23, 2023, 01:51:09 (UTC-07:00)	904.0 B

Cancel [Make public](#)

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

Type here to search 01:55 23-08-2023



## CONCLUSION:-

In this assignment, we learnt how to build our application using AWS Code Build and Deploy on S3 using AWS Code Pipeline.

**Roll No:-42**  
**Batch:- T13**  
**Name :-Piyush Hingorani**

## **ASSIGNMENT-5**

**AIM:** To understand the Kubernetes Cluster Architecture.

**LO MAPPED:** LO1 , LO3

### **THEORY:**

**Q.1 What are the various Kubernetes services running on nodes? Describe the role of each service.**

In a Kubernetes cluster, there are several essential services that run on nodes. These services are critical for the proper functioning of the cluster. Below, I'll describe the role of each service in detail:

#### **kubelet:**

The kubelet is responsible for managing containers on a node. It ensures that the containers in a Pod are running and healthy. It communicates with the control plane to receive Pod specifications and takes actions to make sure the containers match the desired state. For example, if a Pod specification indicates that it should run three containers, the kubelet ensures that those containers are up and running. If a container fails, the kubelet restarts it.

**Example:** Let's say you have a Pod with three containers, and one of them crashes due to a software issue. The kubelet will detect the failure and restart the failed container to maintain the desired state.

#### **kube-proxy:**

Kube-proxy is responsible for managing network connectivity to and from Pods. It maintains network rules on the host to enable communication between Pods and external networks. It sets up routes, handles load balancing, and ensures that network traffic is properly directed to the correct Pod.

**Example:** If you have a service in your cluster that needs to load balance traffic to a set of Pods, kube-proxy manages this load balancing by configuring network rules and routes, directing traffic to the appropriate Pods.

### **Container Runtime:**

The container runtime is responsible for running containers within Pods. Kubernetes supports various container runtimes, such as Docker, containerd, and CRI-O. These runtimes are responsible for pulling container images, creating containers, and managing their lifecycle.

**Example:** If you define a Pod that runs a Docker container with a specific image, the container runtime (e.g., Docker) pulls the image from a container registry and runs the container as specified.

### **cAdvisor (Container Advisor):**

cAdvisor is responsible for collecting and exposing resource usage and performance data for containers. It provides valuable information about CPU, memory, network, and disk usage of running containers.

**Example:** You can use cAdvisor to monitor the resource consumption of your containers. For instance, it can help you identify a container that is consuming an unusually high amount of CPU or memory, indicating a potential performance issue.

### **Node Problem Detector (Optional):**

The Node Problem Detector is responsible for detecting and reporting hardware and system failures on the node. It helps in identifying and isolating issues with nodes, such as hardware errors, kernel panics, or out-of-memory conditions.

**Example:** If a node experiences a hardware issue, such as a failing disk drive, the Node Problem Detector can detect this problem and report it, allowing administrators to take action and potentially drain the node to prevent further issues.

### **Device Plugins (Optional):**

Device plugins are used to expose and manage specialized hardware resources on the node, such as GPUs, FPGAs, or hardware accelerators. They enable Pods to use these resources when required.

**Example:** If you have GPUs on your nodes and want to run machine learning workloads that require GPU acceleration, you can use a GPU device plugin to

expose these GPUs to your Pods. Pods that need GPU resources can request them in their specifications.

### **OS Services (e.g., SSH, NTP):**

These services, including SSH for remote access and NTP for time synchronization, are essential for maintaining the health and reliability of the node. SSH provides administrative access for troubleshooting and maintenance, while NTP ensures the node's clock is synchronized with the cluster, preventing time-related issues.

**Example:** You can use SSH to log in to a node for troubleshooting or updates. NTP ensures that all nodes in the cluster have synchronized clocks, which is crucial for maintaining consistency in distributed systems.

### **Kubelet Container:**

The kubelet itself runs in its own container on the node. It is responsible for interacting with the container runtime, managing container logs, and performing garbage collection to reclaim disk space from unused container images.

**Example:** If you examine a running node, you'll find the kubelet running in its own container. It manages container-related tasks on the node, such as cleaning up old container images to free up storage space.

These roles collectively ensure the smooth operation of a Kubernetes node and the containers within it, making it possible to run and manage containerized applications in a distributed environment.

## **Q.2 What is Pod Disruption Budget (PDB)?**

A **Pod Disruption Budget (PDB)** is a Kubernetes resource that allows you to control the disruption or eviction of Pods during voluntary disruptions (e.g., maintenance) and involuntary disruptions (e.g., hardware failures). PDBs define the minimum availability requirements for Pods in a set of related Pods, such as those belonging to a Deployment or StatefulSet. They ensure that a certain number of Pods are available at all times, helping to maintain the stability and availability of your applications in a Kubernetes cluster.

Here's a detailed explanation of PDBs and an example to illustrate their use:

### **Components of a Pod Disruption Budget (PDB):**

**minAvailable:** This field specifies the minimum number of Pods that must be kept running in the group (e.g., a Deployment or StatefulSet). This ensures that a minimum number of replicas remain available during disruptions.

**maxUnavailable:** This field specifies the maximum number of Pods that can be unavailable during disruptions. It's complementary to minAvailable. You can choose to define one or the other, but not both. It provides a way to limit the maximum unavailability of Pods.

**selector:** PDBs are associated with Pods using label selectors. The selector is used to match Pods in the group that the PDB applies to.

### **Use Cases for PDBs:**

**Rolling Updates:** When performing rolling updates of applications using Deployments or StatefulSets, PDBs can be used to ensure that a certain number of Pods remain available during the update process, preventing unintended disruptions to the application.

**Node Drains:** When nodes need maintenance or are being drained, PDBs can prevent the simultaneous eviction of too many Pods, ensuring that the application maintains its desired level of availability.

**High Availability:** PDBs can be used to enforce high availability requirements for critical components of an application. For example, a database cluster may require a certain number of replicas to be available at all times to prevent data loss.

### **Example of a Pod Disruption Budget:**

Let's say you have a Deployment managing a web application with a replica count of 5. You want to ensure that at least 3 replicas of the web application are available at all times during updates or node maintenance.

Here's how you would define a PDB for this use case:

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: web-app-pdb
```

```
spec:  
  minAvailable: 3  
  selector:  
    matchLabels:  
      app: web-app
```

In this example:

`minAvailable: 3` specifies that at least 3 replicas of Pods with the label `app: web-app` must remain available during disruptions.

`selector` specifies that this PDB applies to Pods with the label `app: web-app`.

With this PDB in place, if you perform a rolling update of the Deployment or if nodes need maintenance, Kubernetes will ensure that at least 3 Pods of the `web-app` Deployment remain operational during these events, thereby meeting the specified availability requirements.

PDBs are a powerful tool for maintaining application stability and availability in Kubernetes clusters, particularly when handling planned or unplanned disruptions to your workloads.

### **Q.3 What is the role of Load Balance in Kubernetes?**

In Kubernetes, a Load Balancer is a critical component that helps distribute network traffic evenly across a set of Pods or Services. Load balancing is essential for ensuring high availability, scaling applications, and maintaining stable network connections. Here's a detailed explanation of the role of Load Balancers in Kubernetes, along with an example:

#### **Role of Load Balancers in Kubernetes:**

**Distributing Traffic:** Load Balancers evenly distribute incoming network traffic across multiple Pods or Services. This ensures that no single Pod or Service becomes overwhelmed, improving the responsiveness and availability of the application.

**High Availability:** Load Balancers are typically configured with health checks to monitor the status of Pods or Services. If a Pod or Service becomes unhealthy or unresponsive, the Load Balancer can automatically route traffic away from it, ensuring the application remains available.

**Scaling:** As your application grows and you need to add more instances (Pods) to handle increased traffic, Load Balancers can seamlessly adapt to include these new instances in the traffic distribution. This makes it easier to scale your application horizontally.

**Session Persistence:** Some Load Balancers support session persistence or sticky sessions, which ensure that requests from the same client are consistently routed to the same backend Pod. This is useful for stateful applications that rely on session data.

**External Access:** Load Balancers often act as a point of entry for external traffic into your cluster. They can route traffic to the appropriate Services within the cluster based on the configuration and rules you define.

#### **Types of Load Balancers in Kubernetes:**

**Service Type:** LoadBalancer: Kubernetes provides a native LoadBalancer Service type. When you define a Service of type LoadBalancer, the Kubernetes cluster provisions an external Load Balancer, typically provided by the cloud provider, to distribute traffic to the Service. For example:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  type: LoadBalancer
```

**Ingress Controllers:** Ingress controllers, such as Nginx Ingress or HAProxy, are used to manage external access to Services within a cluster. Ingress controllers

provide more advanced routing and traffic management capabilities than the basic LoadBalancer Service type.

**Example of Load Balancer in Kubernetes:**

Let's say you have a web application deployed as a set of Pods and you want to make it accessible to external users. You can create a LoadBalancer Service to achieve this:

```
apiVersion: v1
kind: Service
metadata:
  name: web-service
spec:
  selector:
    app: web-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: LoadBalancer
```

In this example:

metadata.name is the name of the Service.

spec.selector specifies the Pods to which the traffic should be load balanced.

spec.ports define the ports to which the Load Balancer should forward traffic.

type: LoadBalancer indicates that you want to provision an external Load Balancer for this Service.

Once this configuration is applied, Kubernetes (or the cloud provider) will provision an external Load Balancer and assign it an IP address. Users can then access your web application using this IP address, and the Load Balancer will distribute incoming requests across the Pods running your web application.

Load Balancers are a fundamental component for ensuring the availability, scalability, and external accessibility of applications in a Kubernetes cluster. They play a crucial role in maintaining a stable and responsive environment for your applications.

**CONCLUSION:** Hence, In this assignment we learned what is the Kubernetes Cluster Architecture.

**Roll No: 42**

**Batch: T13**

**Name: Piyush Hingorani**

**Date: 22/08/2023**

## **ASSIGNMENT - 06**

**AIM-** To understand terraform lifecycle and to Build, change, and destroy AWS infrastructure Using Terraform.

### **LO MAPPED – LO1,LO3**

#### **THEORY-**

Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently. Terraform can manage existing and popular service providers as well as custom in-house solutions.

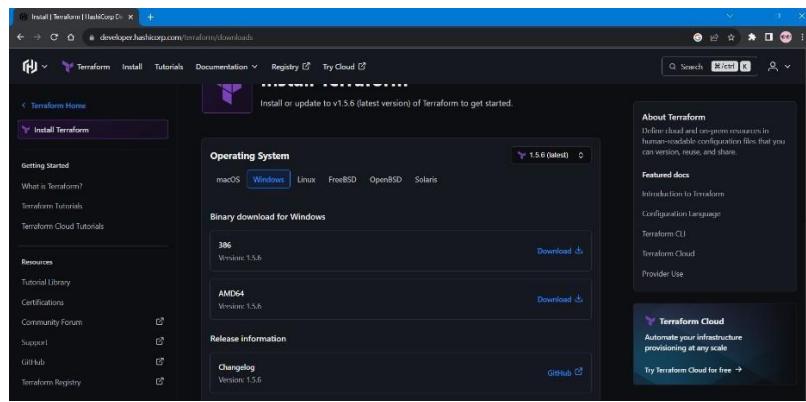
The key features of Terraform are:

- **Infrastructure as Code:** Infrastructure is described using a high-level configuration syntax. This allows a blueprint of your datacenter to be versioned and treated as you would any other code. Additionally, infrastructure can be shared and re-used.
- **Execution Plans:** Terraform has a "planning" step where it generates an execution plan. The execution plan shows what Terraform will do when you call apply. This lets you avoid any surprises when Terraform manipulates infrastructure.
- **Resource Graph:** Terraform builds a graph of all your resources, and parallelizes the creation and modification of any non-dependent resources. Because of this, Terraform builds infrastructure as efficiently as possible, and operators get insight into dependencies in their infrastructure.

- **Change Automation:** Complex changesets can be applied to your infrastructure with minimal human interaction. With the previously mentioned execution plan and resource graph, you know exactly what Terraform will change and in what order, avoiding many possible human errors.

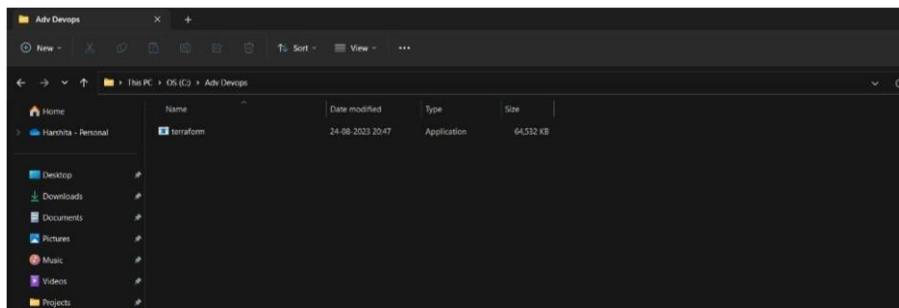
## STEPS-

### Google Search – Terraform DOWNLOAD



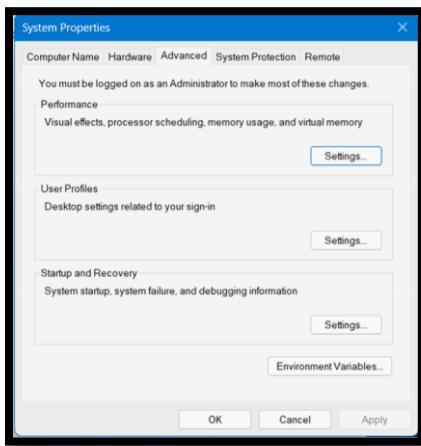
**Create a Folder in Your C drive Named AdvDevops**

**Then Extract The downloaded file in The C drive Folder Named AdvDevops  
Shown Below**



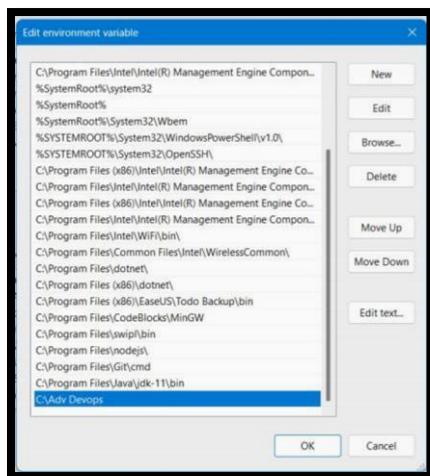
**Now search EDIT THE SYSTEM ENVIRONMENT VARIABLES in your windows search.**

Open it

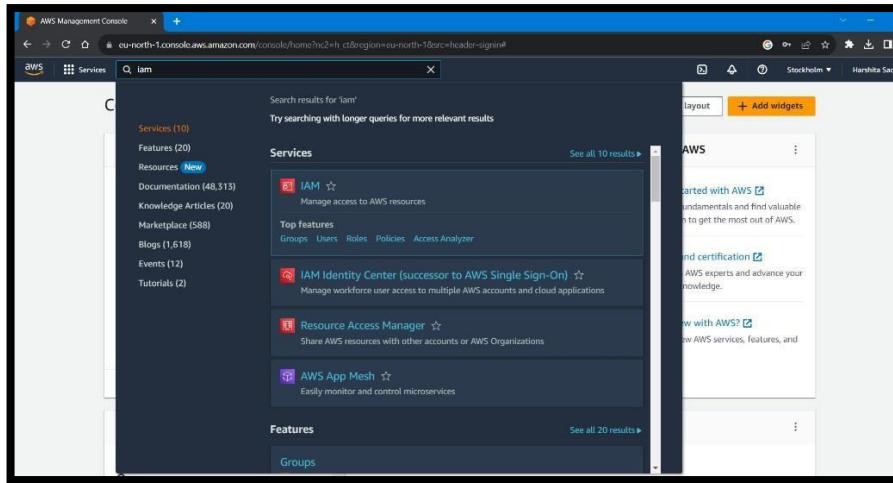


**Now click on PATH OF USER VARIABLES, then click on Edit option**

**Now go to edit and then add new path C:\AdvDevOps**



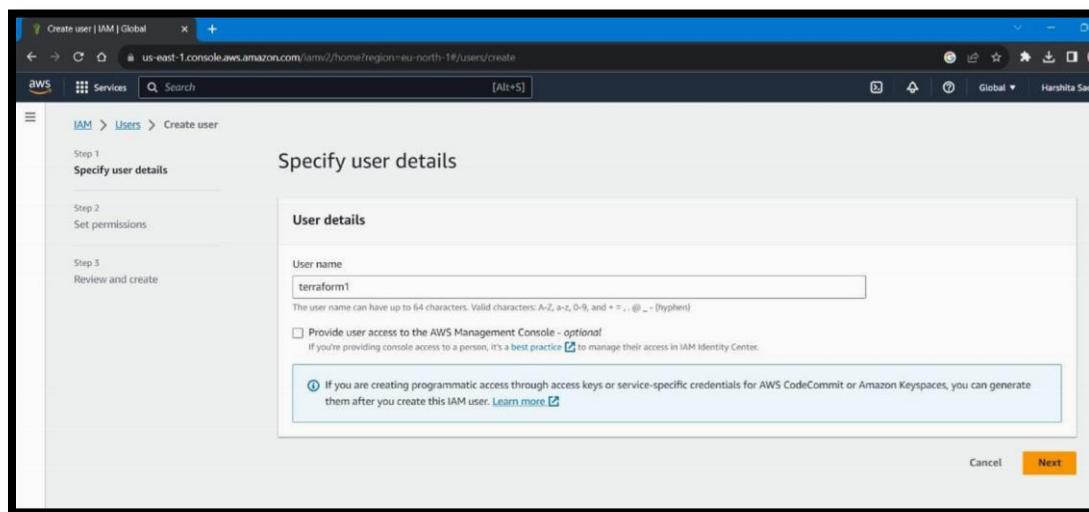
**Repeat same procedure for system variables.**



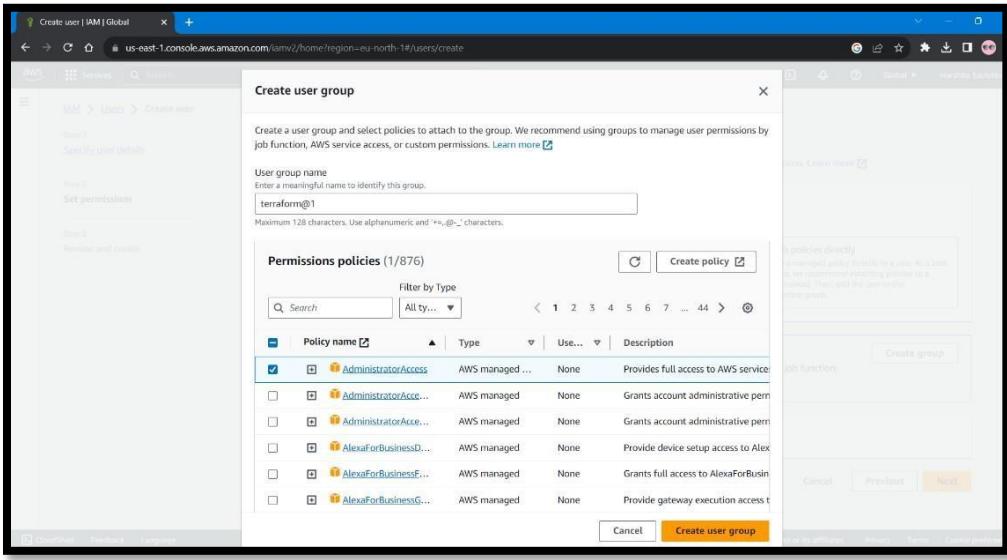
Open And Login to your AWS console- And search IAM and click on it

**Now click on Add Users in The User Section as shown in the image and add the user**

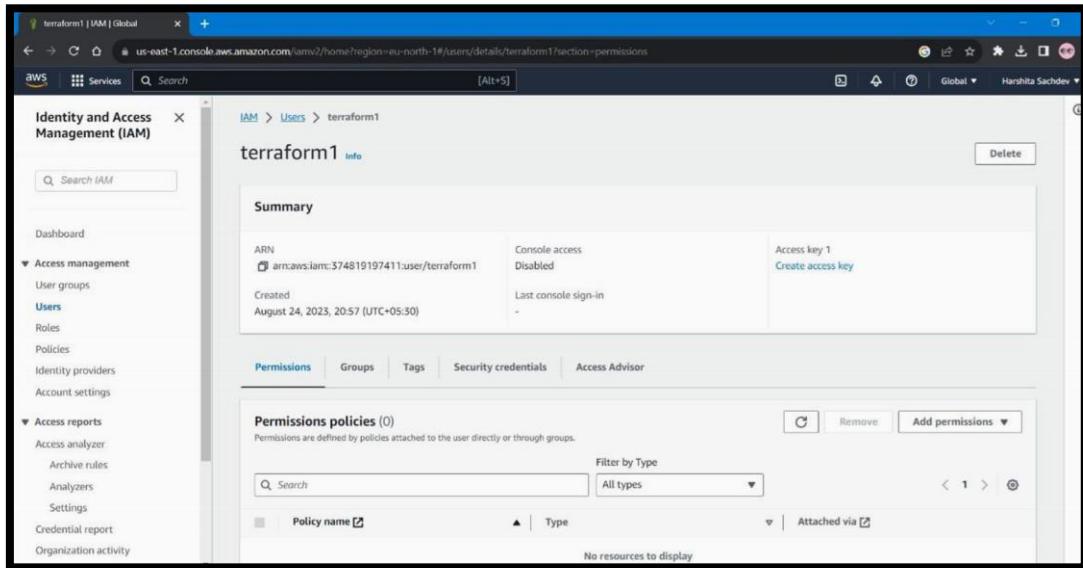
name

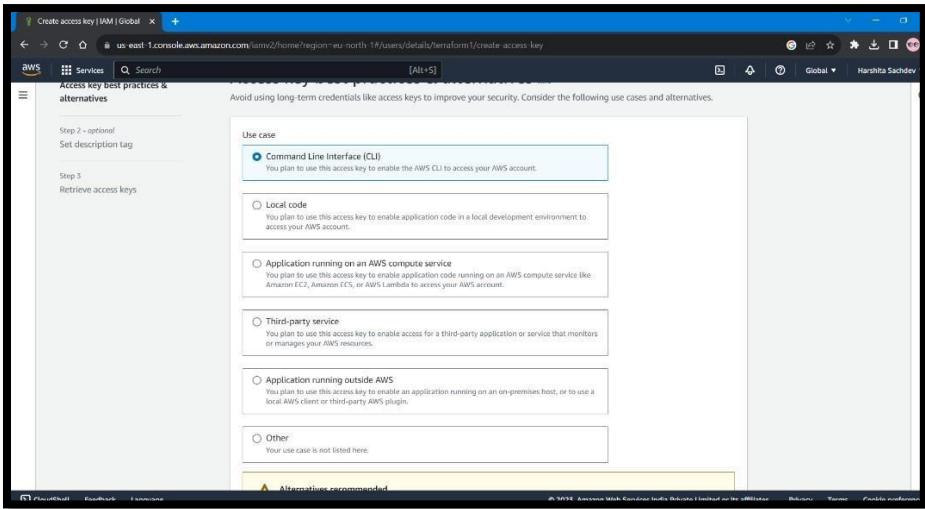


## Add Group name and Check the first Policy Name



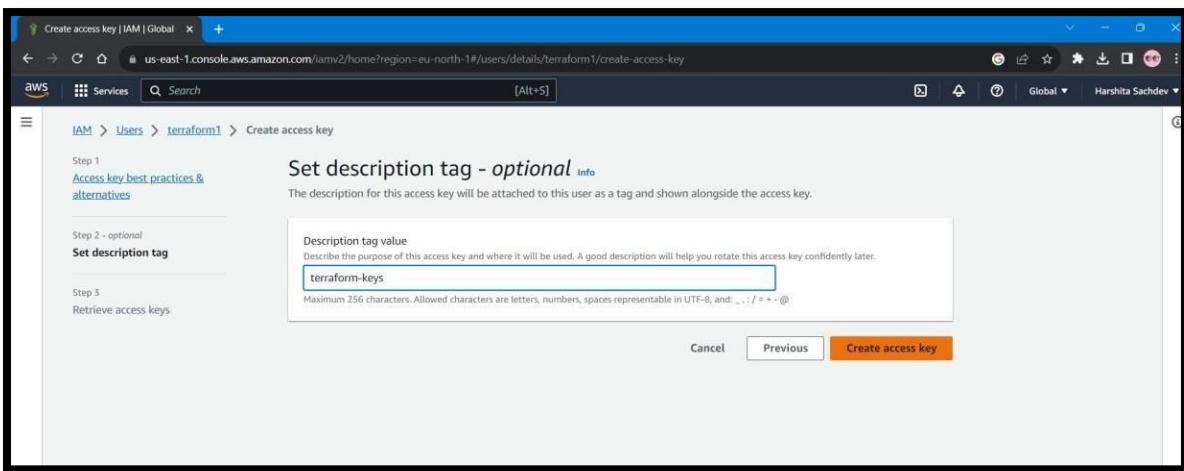
Now, Click on create access key



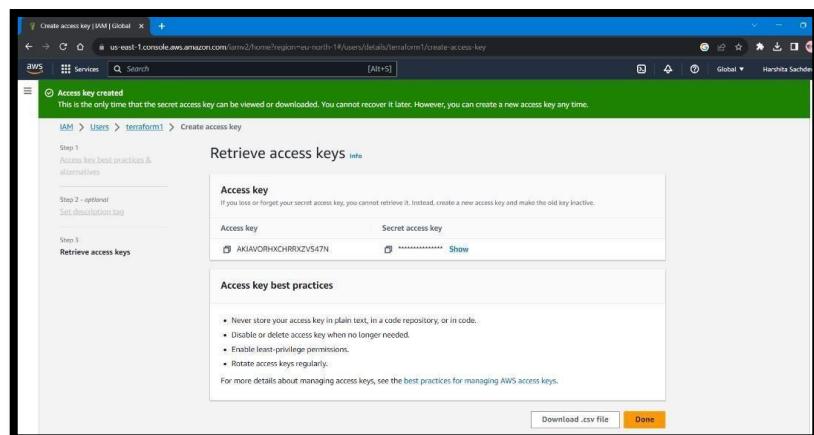


Now select Command Line Interface and click on next

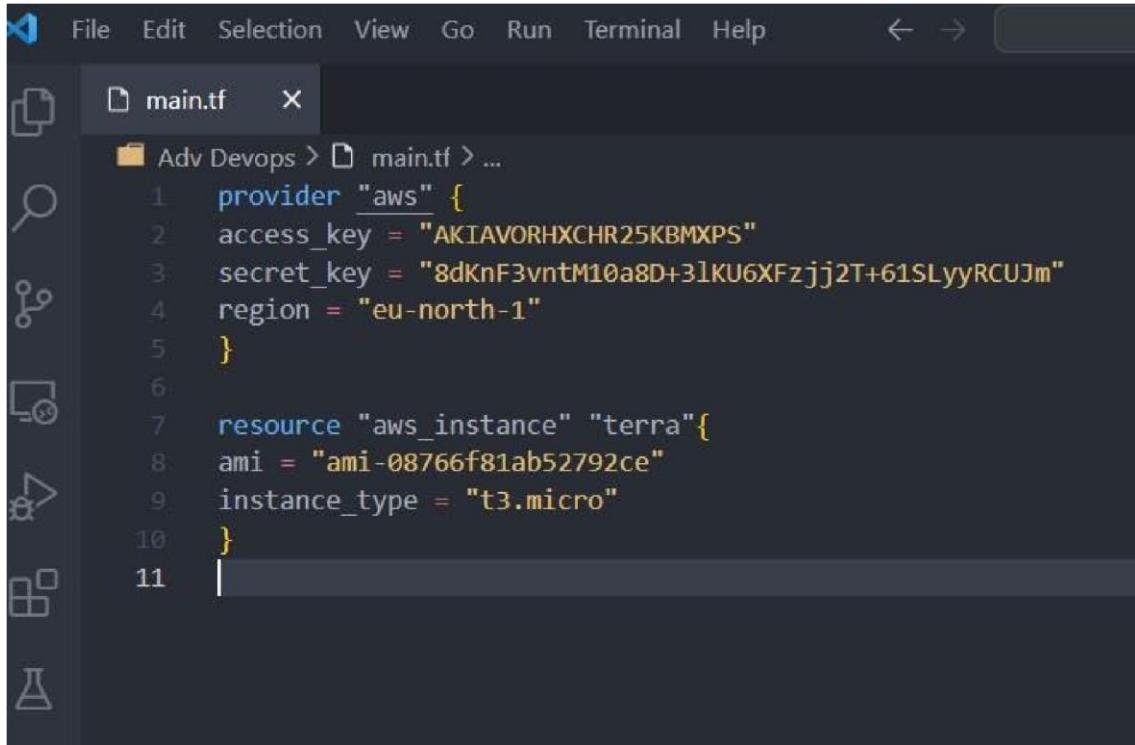
Give tag name and click on create access key



Access key is generated download .csv file

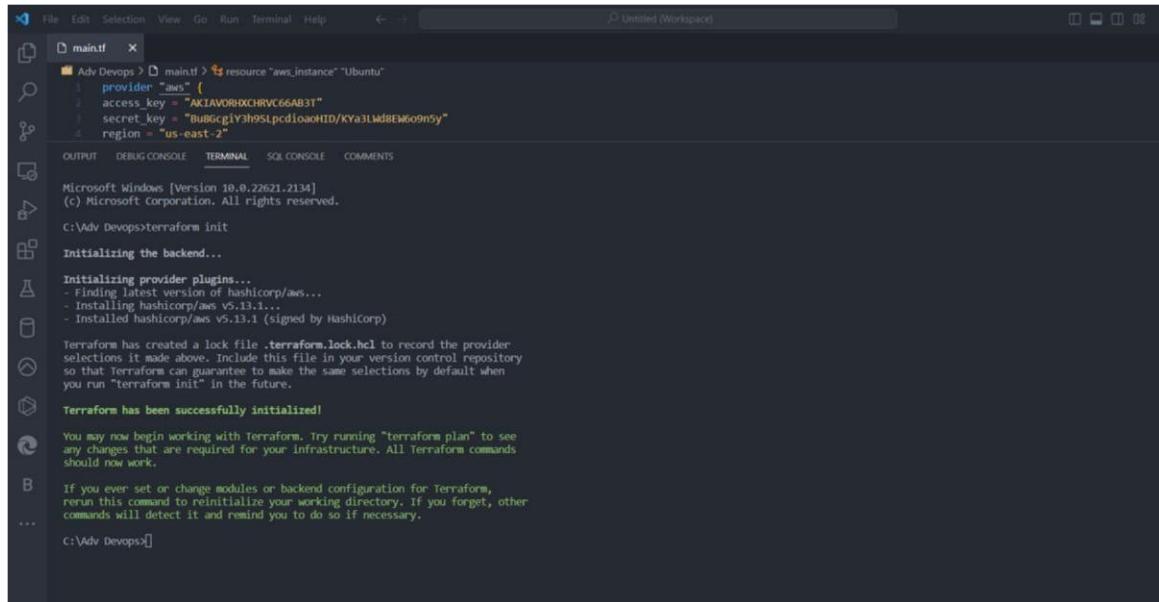


write this script and save it in Terraform Script folder. use the acces keys and secret key present in the csv files.



```
File Edit Selection View Go Run Terminal Help ← → 
main.tf x
Adv Devops > main.tf > ...
provider "aws" {
  access_key = "AKIAVORHXCHR25KBMXPS"
  secret_key = "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"
  region = "eu-north-1"
}
resource "aws_instance" "terra"{
  ami = "ami-08766f81ab52792ce"
  instance_type = "t3.micro"
}
```

open cmd and choose the path where you have stored the script and run the following commands init, plan, apply and destroy.



```
File Edit Selection View Go Run Terminal Help ← → 
main.tf x
Adv Devops > main.tf > resource "aws_instance" "Ubuntu"
provider "aws" {
  access_key = "AKIAVORHXCHRVC66AB31"
  secret_key = "BubScgjY3h951pcdioaoHID/KYa3LWd8EW6o9m5y"
  region = "us-east-2"
}
OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE COMMENTS
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Adv Devops>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.13.1...
- Installed hashicorp/aws v5.13.1 (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.

C:\Adv Devops>
```

```
resource "aws_lambda_function" "lambda" {
  function_name = "lambda"
  runtime       = "nodejs12.x"
  handler       = "index.handler"
  role          = "arn:aws:iam::9089fb15ce7ba39e:lambdaLayerLambdaLayer"
  memory_size   = 128
  timeout       = 3
  layers        = [aws_lambda_layer_lambda_layer.id]
}

resource "aws_lambda_layer_version" "lambda_layer" {
  layer_name    = "lambda-layer"
  content       = filebase64("lambda-layer.zip")
}
```

By running the apply command the instance will be running

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the file structure with `main.tf` selected.
- Code Editor:** Displays the `main.tf` file content:

```
provider "aws" {
  access_key = "AKIAWDRHXCHR25K8BMXPS"
  secret_key = "8dk0f3vntM10a8D+3lKU6XFzjj2T+615LyRCUJm"
  region     = "eu-north-1"
}

resource "aws_instance" "terra"
```
- Terminal:** Shows the Terraform plan output:

```
+ spot_instance_request_id      = (known after apply)
+ subnet_id                     = (known after apply)
+ tags.all                      = (known after apply)
+ tenancy                       = (known after apply)
+ user_data                     = (known after apply)
+ user_data.base64              = (known after apply)
+ user_data.replace_on_change   = false
+ vpc_security_group_ids        = (known after apply)
```
- Status Bar:** Shows the message "Plan: 1 to add, 0 to change, 0 to destroy."
- Bottom Status Bar:** Shows "Do you want to perform these actions?".
- Bottom Output Panel:** Shows the Terraform command and its progress:

```
aws_instance.terra: Creating...
aws_instance.terra: Still creating... [10s elapsed]
aws_instance.terra: Creation complete after 15s [id:i-09d1e4dec65500ae]
```

We can see the instance is running

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays the following table:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Myfirstinstance	i-02631b2a25a6e47d5	Stopped	t3.micro	-	No alarms	+ eu-north-1b	-
<input type="checkbox"/>	-	i-09d1e4dec65500aaee	Running	t3.micro	Initializing	No alarms	+ eu-north-1b	ec2-16-171-134-

By running the destroy command the instance will be terminated

```
File Edit Selection View Go Run Terminal Help <- > Untitled (Workspace)

main.tf x
provider "aws" {
  access_key = "AKIAV0B0XCHR25KBWYDS"
  secret_key = "8d0nfjvnrh10a8D+31KU6XFzjj2T+61StyyRCUJm"
  region     = "eu-north-1"
}

resource "aws_instance" "terra"

OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE COMMENTS + ... Code cmd

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-03e9fe83ffee5cc0ad" -> null
  - volume_size            = 8 -> null
  - volume_type            = "gp2" -> null
}

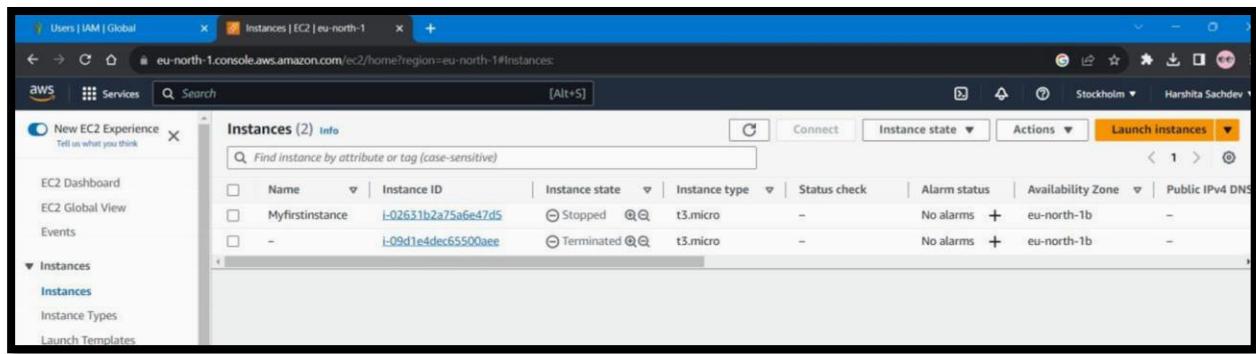
Plan: 0 to add, 0 to change, 1 to destroy.

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

aws_instance.terra: Destroying... [id:i-09d1e4dec65500ae]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 10s elapsed]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 20s elapsed]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 30s elapsed]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 40s elapsed]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 50s elapsed]
aws_instance.terra: Still destroying... [id:i-09d1e4dec65500ae, 60s elapsed]
aws_instance.terra: Destruction complete after 2m2s

Destroy complete! Resources: 1 destroyed.
```



## Delete all security keys and csv files.

## Delete users and user groups

## CONCLUSION –

In this assignment we learnt about the terraform lifecycle, core concepts/terminologies and installation of terraform in windows and creating/destroying an EC2 instance.

**Roll No:-42**  
**Batch:- T13**  
**Name :-Piyush Hingorani**  
**Date :- 12/09/2023**

## **ASSIGNMENT-7**

**AIM:** To perform static analysis on Python programs using SonarQube SAST process.

**LO MAPPED: LO4**

### **THEORY:**

SonarQube is a universal tool for static code analysis that has become more or less the industry standard. Keeping code clean, simple, and easy to read is also a lot easier with SonarQube.

#### **What is SonarQube?**

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications. It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

#### **Benefits of SonarQube**

Sustainability - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications. Increase productivity - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code

- Quality code - Code quality control is an inseparable part of the process of software development.
- Detect Errors - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- Increase consistency - Determines where the code criteria are breached and enhances the quality
- Business scaling - No restriction on the number of projects to be evaluated
- Enhance developer skills - Regular feedback on quality problems helps developers to improve their coding skills

#### **Why SonarQube?**

Developers working with hard deadlines to deliver the required functionality to the customer. It is so important for developers that many times they compromise with the code quality, potential bugs, code duplications, and bad distribution of complexity.

Additionally, they tend to leave unused variables, methods, etc. In this scenario, the code would work in the desired way.

To avoid these issues in code, developers should always follow the good coding practice, but sometimes it is not possible to follow the rules and maintain the good quality as there may be many reasons.

In order to achieve continuous code integration and deployment, developers need a tool that not only works once to check and tell them the problems in the code but also to track and control the code to check continuous code quality. To satisfy all these requirements, here comes SonarQube in the picture.

## **STEPS:**

### Download SonarQube and Sonar Scanner

The screenshot shows the SonarQube download page. At the top, there's a banner for SonarQube 9.1. Below it, the main heading is "Download SonarQube" with the subtitle "The leading product for Code Quality and Security". A sub-subtitle "HELPING DEVS SINCE 2008" is present. The page features four main sections representing different editions:

- Community EDITION**: Used and loved by 200,000+ companies. It's described as "FREE & OPEN SOURCE". A blue "Download for free" button is available. Features listed include static code analysis for 15 languages (Java, JavaScript, C#, TypeScript, Kotlin, Ruby, Go, Scala, Flex, Python, PHP, etc.).
- Developer EDITION**: Built for developers by developers. It includes "Community Edition plus:" which adds C, C++, Obj-C, Swift, ABAP, T-SQL, PL/SQL support, and Detection of Injection Flaws.
- Enterprise EDITION**: Designed to meet Enterprise Requirements. It includes "Developer Edition plus:" which adds Portfolio Management & PDF Executive Reports, and Project PDF reports.
- Data Center EDITION**: Designed for High Availability. It includes "Enterprise Edition plus:" which adds Component redundancy, Data resiliency, and Horizontal Scalability.

At the bottom of the page, there are links for "Version: 9.1", "Release: September 2021", "Getting Started", "Release Notes", "Upgrade Notes", and "Available From DockerHub".

The screenshot shows the SonarScanner documentation page. The left sidebar has a search bar and navigation links for "Try Out SonarQube", "Requirements", "Setup and Upgrade", "Analyzing Source Code", "Scanners", "Analysis Parameters", "Languages", "Test Coverage & Execution", "Importing External Issues", and "Background Tasks".

The main content area is titled "SonarScanner". It shows the version "4.6.2" from "2021-05-07" with a link to "Show more versions". Below this, there's a section titled "Configuring your project" with the sub-instruction "Create a configuration file in your project's root directory called sonar-project.properties". It includes a code snippet example:

```
# 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 the right side, there's a sidebar titled "On this page" with links to "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", and "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>StartSonar.bat
-> Wrapper Started as Console
Wrapper
Launching a JVM...
Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.

2021.09.29 13:56:37 INFO app[]|o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp
2021.09.29 13:56:37 INFO app[]|o.s.a.es.Settings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:53055]
2021.09.29 13:56:37 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:\Program 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&log -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dio.nettyallocatorArena=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j.disable.jmx=true -Djava.locale.providers=COMPAT -Xmx512m -Xms512m -XX:MaxDirectMemorySize=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\es\conf.lib* org.elasticsearch.bootstrap.Bootstrap.ESearch
2021.09.29 13:56:37 INFO app[]|o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
2021.09.29 13:56:37 ERROR app[]|o.s.a.p.ESManagedProcess] Failed to check status
org.elasticsearch.ElasticsearchException: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
at org.elasticsearch.client.RestHighLevelClient.innerPerformRequest(RestHighLevelClient.java:2078)
at org.elasticsearch.client.RestHighLevelClient.innerPerformRequest(RestHighLevelClient.java:1732)
at org.elasticsearch.client.RestHighLevelClient.innerPerformRequest(RestHighLevelClient.java:1702)
at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1672)
at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:19)
at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
at org.sonar.application.process.ESManagedProcess.checkStatus(ESManagedProcess.java:90)
at org.sonar.application.process.ESManagedProcess.checkOperational(ESManagedProcess.java:75)
at org.sonar.application.process.ESManagedProcess.isOperational(ESManagedProcess.java:60)
at org.sonar.application.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
at org.sonar.application.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
... 10 common frames omitted
Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
at org.apache.http.nio.pool.AbstractIOConnPool.requestTimeout(AbstractIOConnPool.java:628)
at org.apache.http.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.process$ timeouts(DefaultConnectingIOReactor.java:214)
at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
at org.apache.http.impl.nio.conn.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.java:221)
at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
at java.base/java.lang.Thread.run(Thread.java:834)
```

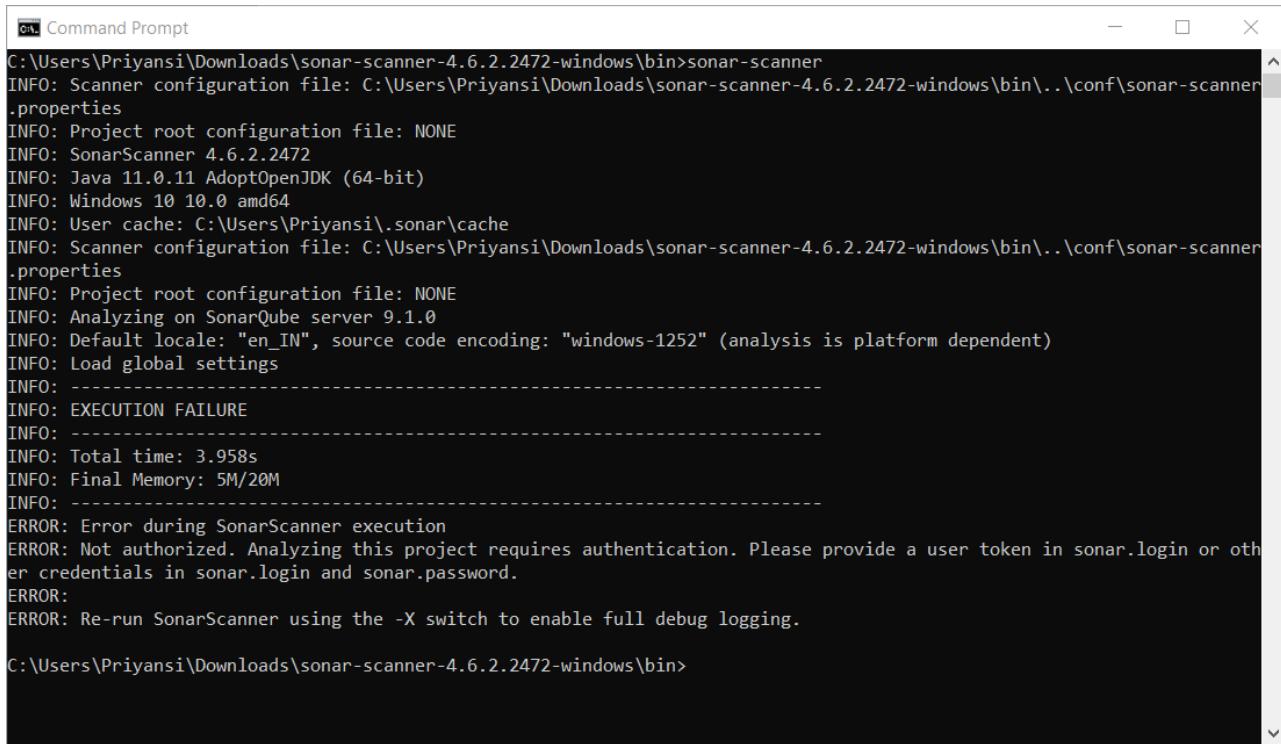
```
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Priyansi>
at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1702)
at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:19)
at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
at org.sonar.application.process.ESManagedProcess.checkStatus(ESManagedProcess.java:90)
at org.sonar.application.process.ESManagedProcess.checkOperational(ESManagedProcess.java:75)
at org.sonar.application.process.ESManagedProcess.isOperational(ESManagedProcess.java:60)
at org.sonar.application.process.ESManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
at org.sonar.application.process.ESManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
... 10 common frames omitted
Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
at org.apache.http.nio.pool.AbstractIOConnPool.requestTimeout(AbstractIOConnPool.java:628)
at org.apache.http.nio.pool.AbstractIOConnPool$InternalSessionRequestCallback.timeout(AbstractIOConnPool.java:894)
at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.process$ timeouts(DefaultConnectingIOReactor.java:214)
at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
at org.apache.http.impl.nio.conn.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.java:221)
at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
at java.base/java.lang.Thread.run(Thread.java:834)

2021.09.29 13:56:50 INFO app[]|o.s.a.ProcessLauncherImpl] Process[es] is up
2021.09.29 13:56:50 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\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-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi.transport=ALL-UNNAMED --add-exports=java.base/dk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=dk.management/com.sun.management.internal=ALL-UNNAMED -Xms512m -Xmx512m -XX:MaxDirectMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*::1] -Dhttp.nonProxyHosts=localhost[127.*::1] -cp .\lib\sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process3944487143195035.properties
2021.09.29 13:51:42 WARN app[]|startUp] #####
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.
2021.09.29 13:51:42 WARN app[]|startUp] #####
2021.09.29 13:51:46 INFO app[]|o.s.a.SchedulerImpl] Process[ce] is up
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

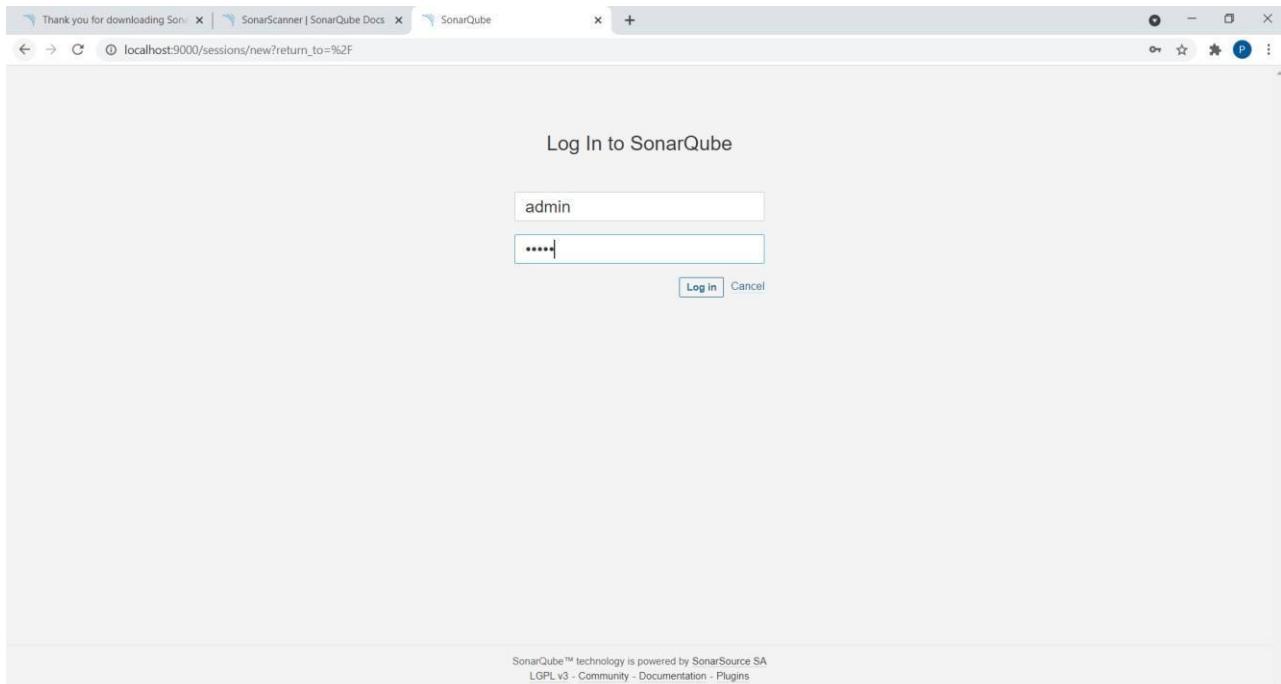


```
C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>sonar-scanner
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 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 other 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      From Bitbucket      From GitHub      From GitLab

Set up global configuration    Set up global configuration    Set up global configuration    Set up global configuration

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

Manually

Embedded database should be used for evaluation purposes only

## Click on Create a project Manually.

Project display name \*

sonarPythonProgram

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

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 47730) - LGPL v3 - Community - Documentation - Plugins - Web API - About

## Give any Project display name.

The screenshot shows the SonarQube dashboard for the project 'sonarPythonProgram1'. At the top, there are three tabs: 'Thank you for downloading SonarQube!', 'SonarScanner | SonarQube Docs', and the active tab 'sonarPythonProgram1'. The URL in the address bar is 'localhost:9000/dashboard?id=sonarPythonProgram1'. The main header includes the SonarQube logo, navigation links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and a search bar 'Search for projects...'. Below the header, the project details are shown: 'sonarPythonProgram1' with a star icon, 'master' branch, and a 'Code' status indicator. The 'Overview' tab is selected. On the right, there are 'Project Settings' and 'Project Information' dropdowns. The main content area asks 'How do you want to analyze your repository?' and provides options for integrating with CI systems: 'With Jenkins', 'With GitHub Actions', 'With Bitbucket Pipelines', 'With GitLab CI', 'With Azure Pipelines', and 'Other CI'. It also offers a local analysis option: 'Locally'. A yellow warning box at the bottom states: '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.'

## Click on Locally

The screenshot shows the SonarQube dashboard for the project 'sonarPythonProgram1' after clicking the 'Locally' button. The URL is now 'localhost:9000/dashboard?id=sonarPythonProgram1&selectedTutorial=manual'. The main content area has changed to 'Analyze your project' with the sub-instruction 'We initialized your project on SonarQube, now it's up to you to launch analyses!'. It shows two steps: 1. Provide a token, which includes a 'Generate a token' button and a text input field containing 'pythonToken1'. A note below says: '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. A yellow warning box at the bottom states: '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.' At the bottom, the footer information includes: 'SonarQube™ technology is powered by SonarSource SA', 'Community Edition - Version 9.1 (build 47736) - LGPL v3 - Community - Documentation - Plugins - Web API - About'.

Give any name to token and click on **Generate**.

The screenshot shows the SonarQube dashboard for the project 'sonarPythonProgram1'. At the top, there are three tabs: 'Thank you for downloading SonarQube', 'SonarScanner | SonarQube Docs', and 'sonarPythonProgram1'. The main content area has a header 'Analyze your project' with the sub-header 'We initialized your project on SonarQube, now it's up to you to launch analyses!'. Below this, there are two numbered steps: 1. Provide a token and 2. Run analysis on your project. Step 1 is active, showing a token value 'pythonToken1: 41740ddd289d68dfda1ec55f28cd250be46d48f' with a copy icon. A note below says '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.' A 'Continue' button is present. Step 2 is shown below with a note: '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.' At the bottom, there is footer text: 'SonarQube™ technology is powered by SonarSource SA', 'Community Edition - Version 9.1 (build 47736) - LGPL v3 - Community - Documentation - Plugins - Web API - About'.

Click on **Continue**.

The screenshot shows the SonarQube dashboard for the project 'sonarPythonProgram1'. The interface is identical to the previous screenshot, but the second step 'Run analysis on your project' is now active. It includes fields for 'What option best describes your build?' (with 'Other (for JS, TS, Go, Python, PHP, ...)' selected), 'What is your OS?' (with 'Windows' selected), and a 'Download and unzip the Scanner for Windows' section with instructions and a command-line example. The command-line example is: 'sonar-scanner.bat -Dsonar.projectKey=sonarPythonProgram1 -Dsonar.sources= -Dsonar.host.url=http://localhost:9000'.

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':400,'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("CD
XLIII"))

```

Open command prompt in this folder and Run program using copied command.

“sonar-scanner.bat

```
D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -
D"sonar.host.url=http://localhost:9000" -
D"sonar.login=41740ddf269d68fdfa1ec55f28cd250be46d48f"
```

```

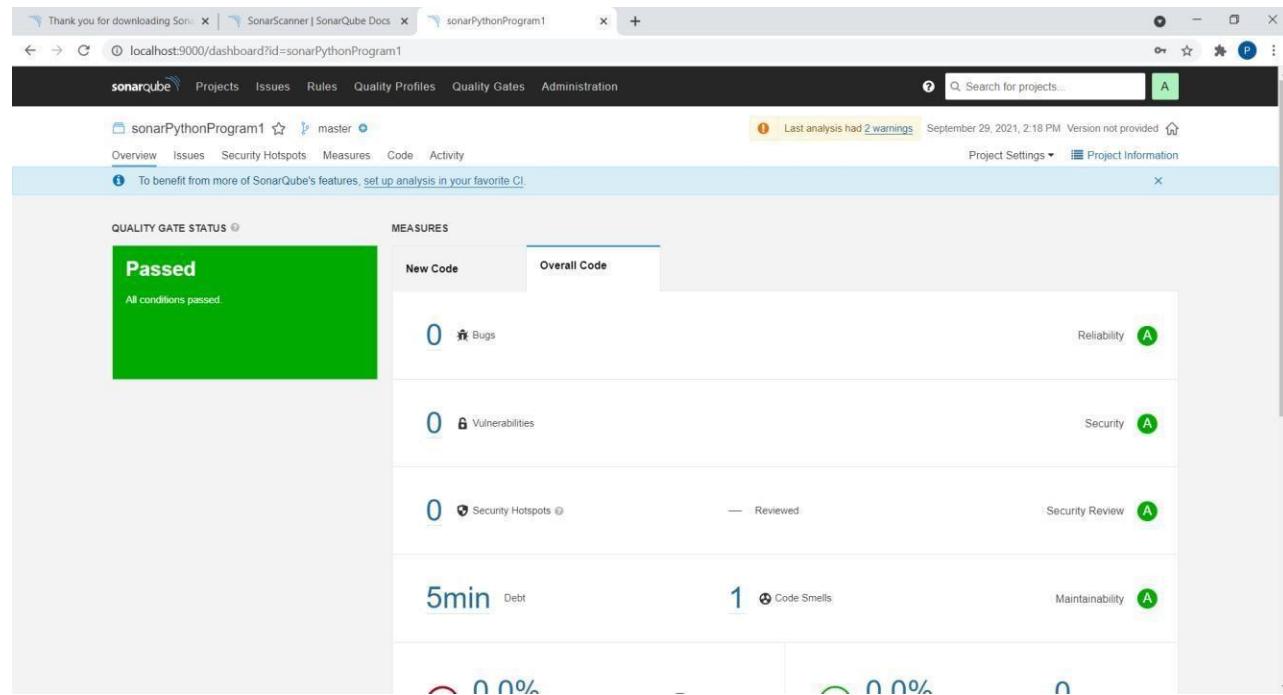
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Priyansi\Documents\SonarExps>sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=41740ddf269d68fdfa1ec55f28cd250be46d48f"
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: Load global settings (done) | time=20ms
INFO: Server id: BF41A1F2-AXwphP#4x91b8xeZLScU
INFO: User cache: C:\Users\Priyansi\sonar\cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=102ms
INFO: Load/download plugins (done) | time=1674ms
INFO: Process project properties
INFO: Load project properties (done) | time=20ms
INFO: Execute project builder
INFO: Execute project builders (done) | time=2ms
INFO: Project key: sonarPythonProgram1
INFO: Base dir: C:\Users\Priyansi\Documents\SonarExps
INFO: Working dir: C:\Users\Priyansi\Documents\SonarExps\scannerwork
INFO: Load project settings for component key: 'sonarPythonProgram'
INFO: Load project settings for component key: 'sonarPythonProgram1' (done) | time=40ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=201ms
INFO: Load active rules
INFO: Load active rules (done) | time=4452ms
WARN: SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
INFO: Indexing files...
INFO: Project configuration:
INFO: 1 file indexed
INFO: Quality profile for py: Sonar way
INFO: ----- Run sensors on module sonarPythonProgram1
INFO: Load metrics repository
INFO: Load metrics repository (done) | time=37ms
INFO: Load Python metrics [python]
WARN: Your code is analyzed as compatible with python 2 and 3 by default. This will prevent the detection of issues specific to python 2 or python 3. You can get a more precise analysis by setting a python version in your configuration via the parameter "sonar.python.version"
INFO: Starting global symbols computation
INFO: 1 source file to be analyzed
INFO: Load project repositories

```

```
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: CPD 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 50ms, dir size=103.9 kB
INFO: Analysis report compressed in 19ms, zip size=14.7 kB
INFO: Analysis report uploaded in 76ms
ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=sonarPythonProgram1
Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
More about the report processing at http://localhost:9000/api/ce/task?id=AxwwwY1hx91b8xeZLXH1
Analysis total time: 7.502 s
-----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 10.887s
INFO: Final Memory: 7M/30M
INFO: -----
C:\Users\Priyanshi\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.



Thank you for downloading SonarScanner | SonarQube Docs | sonarPythonProgram1

localhost:9000/dashboard?id=sonarPythonProgram1

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

sonarPythonProgram1 master

Last analysis had 2 warnings September 29, 2021, 2:18 PM Version not provided

Search for projects... A

Overview Issues Security Hotspots Measures Code Activity

Vulnerabilities: 0

Security: A

Security Hotspots: 0 Reviewed: Security Review A

Debt: 5min

Code Smells: 1

Maintainability: A

Coverage: 0.0% Coverage on 15 Lines to cover Unit Tests

Duplications: 0.0% Duplications on 16 Lines Duplicated Blocks

ACTIVITY

Issues: September 29, 2021, 2:18 PM not provided

There isn't enough data to generate an activity report.

Thank you for downloading SonarScanner | SonarQube Docs | Issues

localhost:9000/project/issues?id=sonarPythonProgram1&resolved=false

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

sonarPythonProgram1 master

Last analysis had 2 warnings September 29, 2021, 2:18 PM Version not provided

Search for projects... A

Overview Issues Security Hotspots Measures Code Activity

My Issues All

Bulk Change

1 / 1 issues 5min effort

**Filters**

Type: Bug 0, Vulnerability 0, Code Smell 1

Severity: Blocker 0, Critical 0, Major 0, Minor 1, Info 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\_]\*\$. Why is this an issue?

4 minutes ago L2 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.ManagedProcessHandlerEventWatcher.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.BaseFutureSync.getValue(BaseFuture.java:262)
[jvm 1 |     at org.elasticsearch.common.util.concurrent.BaseFutureSync.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.util.concurrent.ExecutionException: 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.requestCallback.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.execute(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.client.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.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 13:50:00 INFO app[]|o.s.a.SchedulerImpl] Process[es] is up
[jvm 1 | 2021.09.29 13:50:00 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\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-UNNAMED --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=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED --add-opens=java.base/java.nio.charset=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process17779451691724101819\properties
[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 WARN app[]|o.s.a.SchedulerImpl] 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[]|o.s.a.SchedulerImpl] #####
[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
```

**CONCLUSION:** In this assignment, we learnt analysis of using sonarqube. The goal of SonarQube is to empower developers first and to grow an open community around the quality and security of code.

**Roll No:-42**  
**Batch:- T13**  
**Name :-Piyush Hingorani**  
**Date :- 12/09/2023**

## **ASSIGNMENT-8**

**AIM:-**To understand continuous monitoring using Nagios

**LO MAPPED:- LO1, LO5**

### **THEORY:-**

Nagios is an open-source monitoring system that provides monitoring of services, applications, and network resources. It is designed to alert system administrators about potential issues before they become critical problems. Nagios allows you to monitor the entire IT infrastructure, including servers, switches, applications, and services. It provides a comprehensive monitoring solution for both small and large organizations. Some key features and capabilities of Nagios include:

**Monitoring Capabilities:** Nagios can monitor a wide variety of network services including SMTP, POP3, HTTP, NNTP, ICMP, SNMP, FTP, SSH, and many more.

**Alerting and Notification:** It provides alerting and notification functionalities to notify system administrators when something goes wrong. Nagios can send alerts via email, SMS, or other methods to ensure that the right people are notified in real-time.

**Plugin Architecture:** Nagios has a modular architecture that allows users to develop their plugins and addons to monitor specific devices and services that are not covered by default.

**Customizable Dashboards and Reports:** Nagios offers customizable dashboards and reporting capabilities that provide insights into the performance and health of the monitored resources.

**Scalability and Flexibility:** Nagios can scale to monitor complex, large-scale IT infrastructures. It is highly flexible and can be customized to meet specific monitoring and alerting requirements.

**Extensibility:** Nagios can be extended through various addons and plugins, allowing it to integrate with other tools and services, and enabling the monitoring of a wide range of devices and applications.

**Historical Monitoring and Trend Analysis:** Nagios can store historical data and provide trend analysis, allowing system administrators to identify patterns and plan for future infrastructure needs.

**Community Support and Active Development:** Being an open-source project, Nagios has a vibrant community that contributes to its development and support. This community-driven approach ensures that the software remains updated and robust.

**Centralized Monitoring:** Nagios provides a centralized view of the entire IT infrastructure, allowing administrators to have a comprehensive overview of the health and performance of all monitored resources from a single location.

**Integration with Third-Party Tools:** Nagios can integrate with various third-party tools and services, making it a versatile monitoring solution that can fit into different IT ecosystems and workflows.

#### **STEPS:-**

**1) Go to google.com, Search Nagios Demo**

**Click on the first link shown below**

Google search results for "nagios demo":

- [Nagios XI Online Demo](https://exchange.nagios.org/directory/Demos/details)  
An online **demo** of Nagios XI. The **demo** allows you to test configuration wizards, dashlets, dashboards, views, and more. Reviews (0).
- [Demos - Nagios Exchange](https://exchange.nagios.org/directory/Demos)  
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 ...
- [Nagios XI Online Demo](https://exchange.nagios.org/directory/Demos/details)

## 2) Now click on the website-

Nagios®

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](http://nagiosxi.demos.nagios.com)

Hits 141800

Search Exchange

search...  Advanced Search

Search All Sites

Nagios Live Webinars

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

### 3) Now click on login as administrator

The screenshot shows the Nagios XI login interface. On the left, there's a "Login" form with fields for "Username" and "Password", a "Login" button, and a "Forgot your password?" link. Below the form is a "Select Language:" dropdown with various flags representing different languages. To the right, a large banner reads "Nagios XI Demo System". Underneath the banner, the text "Demo Account Options" is displayed, followed by a list of account types with their respective log-in links and credentials:

- 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

Below this section, the text "Demo Notes" is visible.

The screenshot shows the Nagios XI Home Dashboard. The top navigation bar includes links for Home, Views, Dashboards, Reports, Configure, Tools, Help, Admin, and a Logout button. The dashboard features several cards:

- Getting Started Guide**: Includes sections for "Common Tasks" (Change your account settings, Change your notifications settings, Configure your monitoring setup) and "Getting Started" (Learn about XI, Signup for XI news).
- Host Status Summary**: Shows a grid with columns for Up, Down, Unreachable, Pending, and a summary section for Unhandled, Problems, and All. Last Updated: 2021-10-05 05:06:48.
- Service Status Summary**: Shows a grid with columns for Ok, Warning, Unknown, Critical, and Pending. Last Updated: 2021-10-05 05:06:48.
- Administrative Tasks**: Includes a "Task" section and "Initial Setup Tasks" (Configure mail settings).
- We're Here To Help!**: Features a photo of a support technician wearing a headset and a list of support resources: Support Forum / Customer Support Forum, Help Resources, Customer Ticket Support Center, Customer Phone Support, and a phone number +1 651-204-9102 Ext. 4.
- Start Monitoring**: Includes links for Run a Config Wizard, Run Auto-Discovery, and Advanced Config.

On the left sidebar, there are several collapsed sections: Quick View, Details, Graphs, Maps, and Incident Management. At the bottom of the sidebar, it says "Nagios XI 5.7.2 • Check for Updates".

The screenshot shows the Nagios XI interface. On the left, a sidebar contains navigation links for Home Dashboard, Quick View, Details, Graphs, Maps, and Incident Management. The main content area displays the 'Host Status' summary for all hosts. It includes two summary tables: 'Host Status Summary' and 'Service Status Summary'. The 'Host Status Summary' table shows counts for Up, Down, Unreachable, Pending, Unhandled, Problems, and All. The 'Service Status Summary' table shows counts for Ok, Warning, Unknown, Critical, Pending, Unhandled, Problems, and All. Below these are detailed tables for 'Host' and 'Status Information'.

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 49s	5/5	2021-10-05 05:05:20	CRITICAL - 216.27.178.28: ita 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	3763d 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

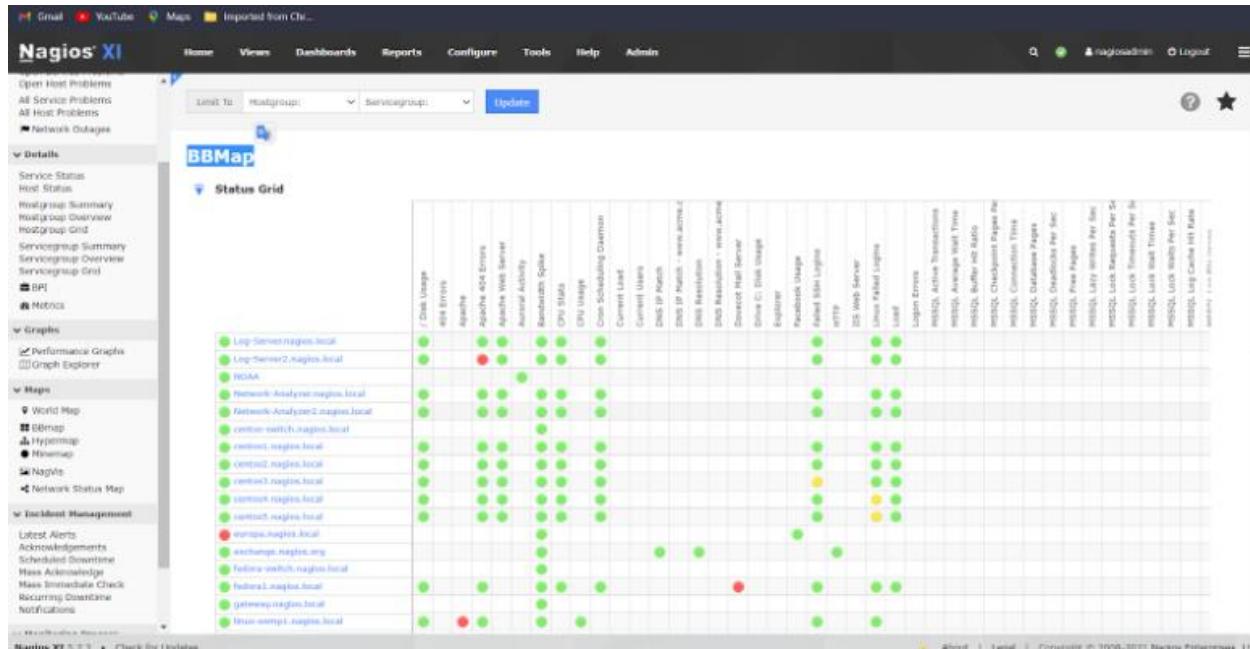
## 5) Now click on Host Group Status.

The screenshot shows the Nagios XI interface. The main content area displays the 'Host Group Status' summary. It includes two summary tables: 'Host Status Summary' and 'Service Status Summary'. The 'Host Status Summary' table shows counts for Up, Down, Unreachable, Pending, Unhandled, Problems, and All. The 'Service Status Summary' table shows counts for Ok, Warning, Unknown, Critical, Pending, Unhandled, Problems, and All. Below these are detailed tables for 'Status Summary For All Host Groups'.

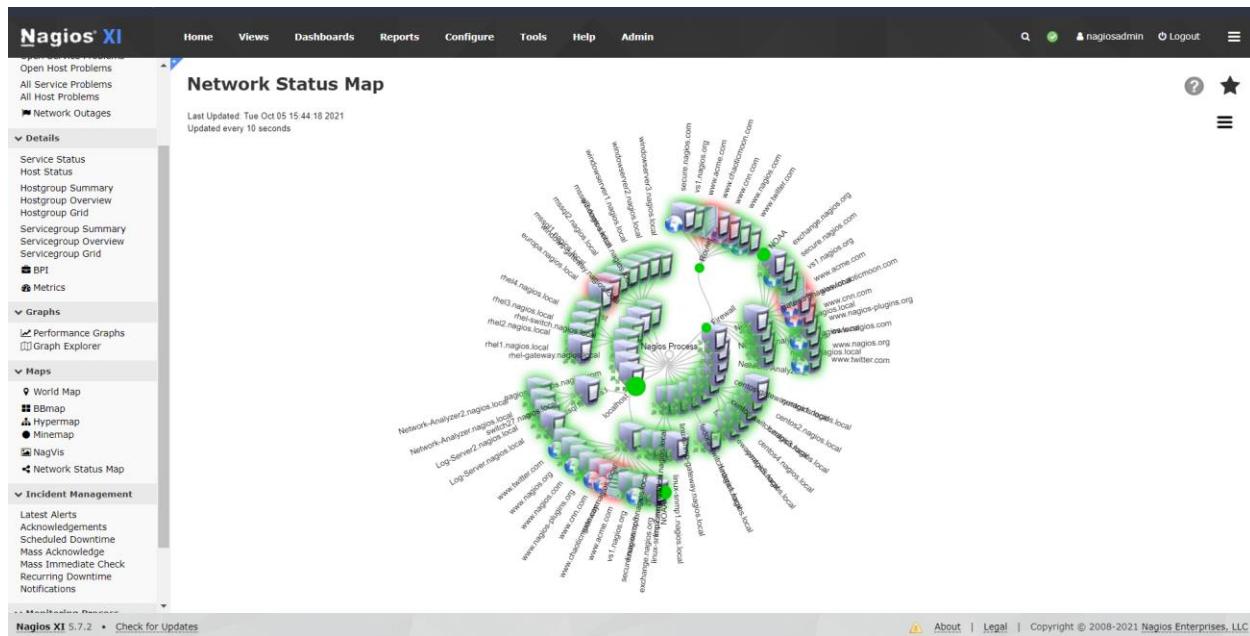
Host Group	Hosts	Services
Monitoring Servers (Monitoring Servers)	5 Up	93 Ok, 14 Warning, 2 Critical
Hostgroup Two (hg2)	1 Up, 1 Down	11 Ok, 4 Warning, 2 Unknown, 4 Critical
Some Other Hostgroup (hg3)	2 Up	11 Ok, 1 Warning, 2 Critical
Linux Servers (linux-servers)	11 Up	218 Ok, 27 Warning, 2 Unknown, 15 Critical
Network Devices (network-devices)	7 Up	215 Ok, 35 Warning, 6 Critical

## 6) Now we click on BBMap

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



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



## **CONCLUSION:-**

Continuous monitoring with Nagios enables proactive detection of system issues, ensuring minimal downtime and enhanced operational efficiency. Through customizable alerts and comprehensive reporting, Nagios empowers administrators to maintain optimal performance across diverse IT environments. Its scalable architecture and robust community support make it an invaluable tool for streamlined and centralized monitoring.

**Roll No.:42**  
**Name: Piyush Hingorani**  
**Batch: T13**  
**DATE: 13/10/23**

## **ASSIGNMENT-9**

**AIM:- To understand AWS Lambda functions**

**LO MAPPED: LO1, LO5**

### **THEORY:**

AWS Lambda is a serverless compute service provided by Amazon Web Services (AWS) that enables you to run code in response to various events without the need to manage servers. It's a key component of AWS's serverless computing offerings. To understand the theory behind AWS Lambda functions, let's break it down into its core concepts:

#### **Serverless Computing:**

Serverless computing is a cloud computing model where cloud providers (like AWS) manage the infrastructure for you, allowing you to focus solely on your code.

#### **Lambda Function:**

A Lambda function is the fundamental unit of execution in AWS Lambda. It is a piece of code that can be executed in response to events such as HTTP requests, file uploads, database changes, etc. Lambda supports multiple programming languages, including Node.js, Python, Java, and more.

#### **Event Sources:**

Lambda functions are triggered by events. These events can come from various AWS services, like Amazon S3, Amazon DynamoDB, Amazon API Gateway, or custom events from your applications. Lambda listens to these event sources and automatically executes the code you've configured.

#### **Stateless Execution:**

Lambda functions are stateless, meaning they don't maintain any server-specific state between invocations. Each invocation of a Lambda function is independent and isolated from the others.

### **Scaling and Concurrency:**

AWS Lambda automatically scales based on the number of incoming events. If there are more events, AWS will create more instances of your Lambda function to handle the load, and if there are fewer events, it will scale down accordingly. You pay only for the compute time your code consumes.

### **Execution Environment:**

Each Lambda function runs in an execution environment provided by AWS. You can't control or manage this environment, but you can specify its configuration, including the amount of memory allocated to the function.

### **Function Versioning and Aliases:**

You can create multiple versions of a Lambda function. This is useful for deploying and managing different versions of your code. You can also create aliases to point to specific versions, allowing you to easily switch between them.

### **IAM Roles:**

Lambda functions can assume AWS Identity and Access Management (IAM) roles. These roles define what AWS services and resources the function can interact with, ensuring proper security and access control.

### **Logging and Monitoring:**

AWS Lambda provides built-in logging to capture function execution details. You can also integrate it with AWS CloudWatch for monitoring and creating custom metrics.

### **Triggers and Destinations:**

Lambda can be triggered by various event sources and can send the results to destinations such as other AWS services, like S3, DynamoDB, SNS, and more.

## **STEPS:**

**Login to Aws account-**

**Search S3 ,click on the option below shown-**

The screenshot shows the AWS Management Console search results for 's3'. The search bar at the top contains 's3'. Below it, the results are categorized into 'Services' and 'Features'.

- Services** (7):
  - S3: Scalable Storage in the Cloud
  - S3 Glacier: Archive Storage in the Cloud
  - Athena: Query Data in S3 using SQL
  - AWS Snow Family: Large Scale Data Transport
- Features** (10):
  - Amazon S3 File Gateway: Storage Gateway feature
  - Datasets: IoT Analytics feature

On the right side of the search results, there is a sidebar with the following information:

- Connected to your AWS on-the-go
- Console Mobile App now supports additional regions. Download the Console Mobile App to your iOS or Android mobile device. [Learn more](#)
- with Amazon Location Maps
- Multi-Region Access Points

**Create an S3 bucket by giving it a name**

The screenshot shows the 'Create bucket' wizard in the AWS S3 service. The steps are as follows:

- General configuration**:
  - Bucket name**: myawsbucket
  - AWS Region**: Asia Pacific (Mumbai) ap-south-1
  - Copy settings from existing bucket - optional**: Only the bucket settings in the following configuration are copied.  
Choose bucket
- 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

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

**Tags (0) - optional**  
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

**Add tag**

**Default encryption**  
Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

Disable  
 Enable

**Advanced settings**

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

**Create bucket**

## Click on upload button after the s3 bucket is created in the object section

Amazon S3 > neel-patel-t21-82 > Upload

**Upload** [Info](#)

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

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

	Name	Folder	Type	Size

**Files and folders (0)**

All files and folders in this table will be uploaded.

**Find by name**

**No files or folders**

You have not chosen any files or folders to upload.

**Destination**

Destination

**Feedback** English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms](#)

Add any .py or .java extenstion file and click on upload

The screenshot shows the AWS S3 console interface. At the top, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The Objects tab is selected. Below the tabs, there's a header bar with actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown), Create folder, and Upload. A search bar says "Find objects by prefix". A table lists the objects:

Name	Type	Last modified	Size	Storage class
Sum1.py	py	September 7, 2021, 15:16:21 (UTC+05:30)	150.0 B	Standard
T11.jpg	jpg	September 7, 2021, 15:31:45 (UTC+05:30)	130.1 KB	Standard

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

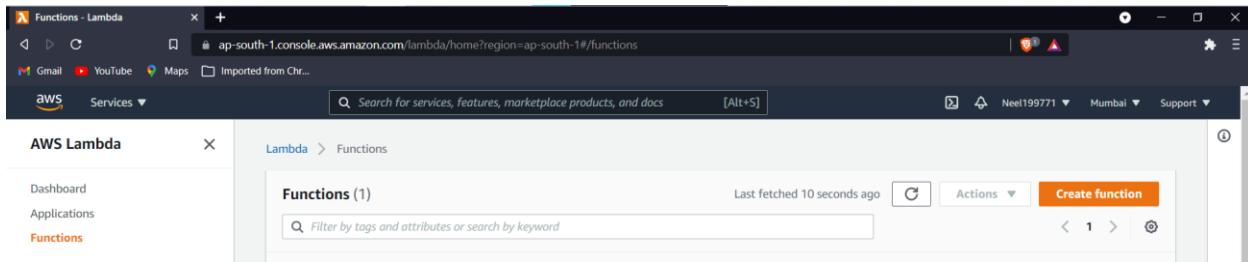
Now search Lamda

The screenshot shows the AWS Lambda search results. The search bar at the top contains the query "lambda". The results are categorized into Services and Features:

- Services**:
  - Lambda: Run Code without Thinking about Servers
  - CodeBuild: Build and Test Code
  - AWS Signer: Ensuring trust and integrity of your code
  - Amazon Lex: Build Voice and Text Chatbots
- Features**:
  - Local processing: IoT Core feature
  - Target groups: EC2 feature

At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

## Click create function

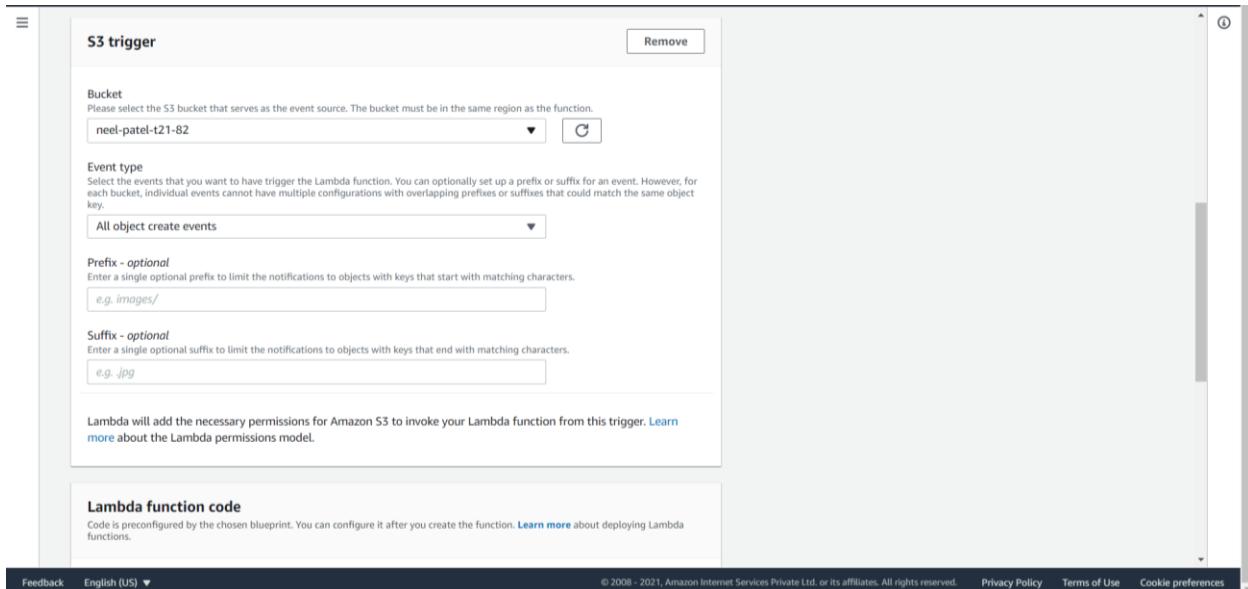


## Click on below options and click on configure

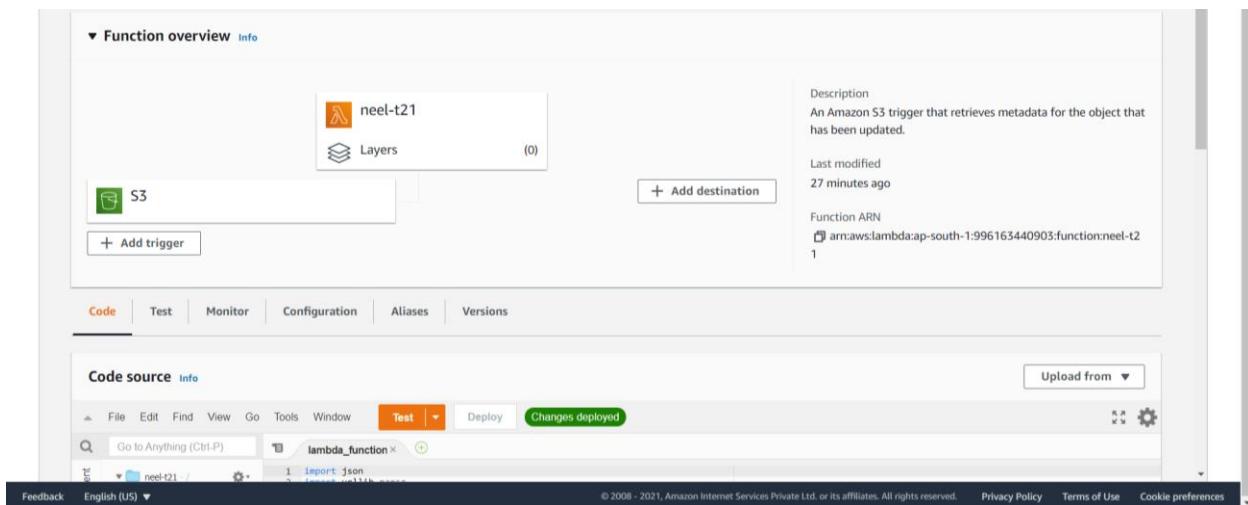
A screenshot of the 'Create function' blueprint selection screen. It shows four main options: 'Author from scratch', 'Use a blueprint' (which is selected), 'Container image', and 'Browse serverless app repository'. Below this, there is a section titled 'Blueprints' with a search bar and a grid of blueprint cards. One card, 's3-get-object-python', is highlighted with a blue border. Other cards include 'kinesis-firehose-syslog-to-json', 'config-rule-change-triggered', 'lex-book-trip-python', 'dynamodb-process-stream', 'microservice-http-endpoint', and 'node-exec'. At the bottom, there are links for 'Feedback', 'English (US)', and copyright information.

A screenshot of the 'Configure blueprint s3-get-object-python' configuration screen. It includes sections for 'Basic information', 'Execution role', 'Role name', and 'Policy templates - optional'. In the 'Execution role' section, 'Create a new role with basic Lambda permissions' is selected. A note says 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.' In the 'Role name' section, 'myRoleName' is entered. In the 'Policy templates - optional' section, 'Amazon S3 object read-only permissions' is selected. The footer contains standard links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

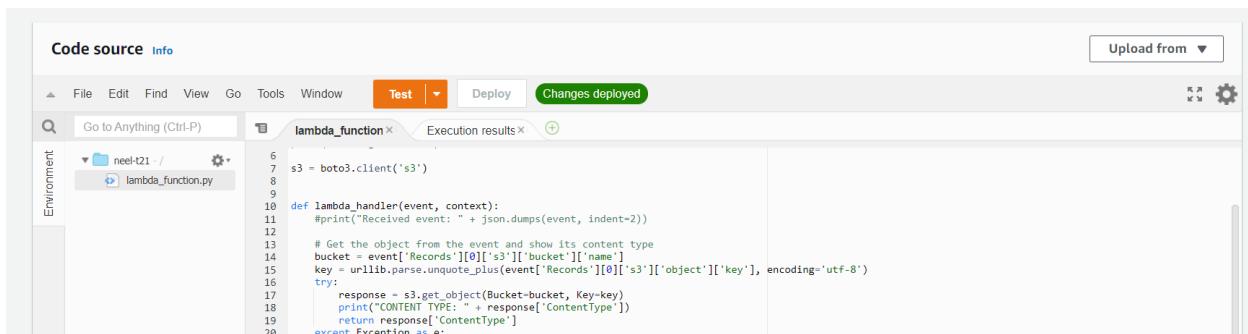
Select the bucket created and create trigger ,click on create function-



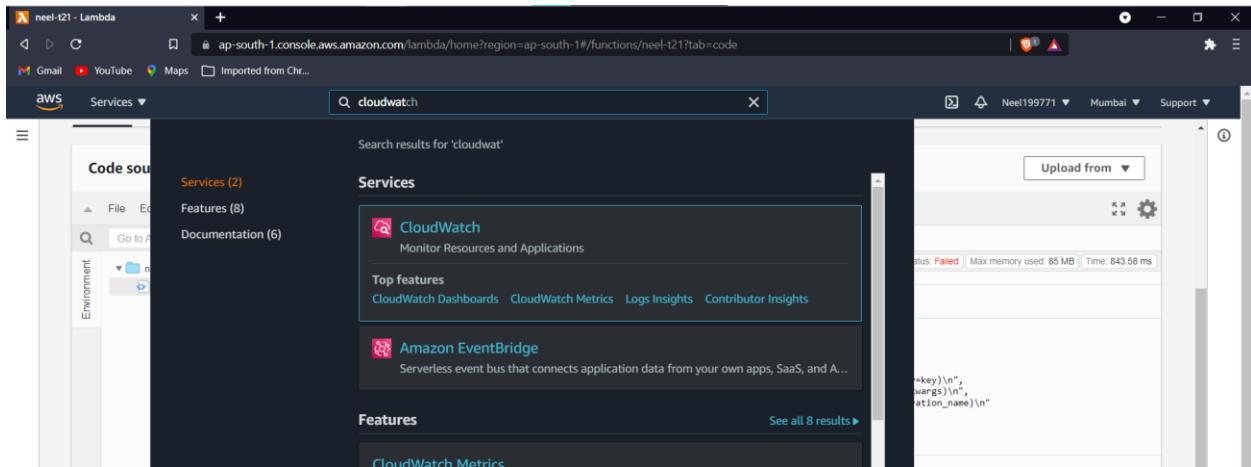
Check the given trigger is created



Click on the orange test button-



**Now,**



**Check the logs of the test-**

A screenshot of the AWS CloudWatch Logs interface. The URL is /aws/lambda/neel-t21. The left sidebar shows 'Log groups' (highlighted in orange) and 'Logs Insights'. The main content area shows 'Log group details' for the group '/aws/lambda/neel-t21'. It includes fields for Retention (Never expire), Creation time (21 minutes ago), Stored bytes (0), ARN (arn:aws:logs:ap-south-1:996163440903:log-group:/aws/lambda/neel-t21:\*), KMS key ID (-), Metric filters (0), Subscription filters (0), and Contributor Insights rules (-). Below this, the 'Log streams' tab is selected, showing two log streams: '2021/09/07/[LATEST]e51215ab14be44f8555e7bf287da1d' (Last event time: 2021-09-07 15:47:48 UTC+05:30) and '2021/09/07/[LATEST]842026ddeba34f8bea274d87a7b9793' (Last event time: 2021-09-07 15:52:47 UTC+05:30). There are also buttons for 'Create log stream' and 'Search all'.

**Now terminate-**

**Click on delete function.**

**Function overview** [Info](#)

**neel-t21**

S3

Layers (0)

+ Add destination

+ Add trigger

Description  
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Last modified  
31 minutes ago

Function ARN  
arn:aws:lambda:ap-south-1:996163440903:function:neel-t21

Code | Test | Monitor | Configuration | Aliases | Versions

**Code source** [Info](#)

File Edit Find View Go Tools Window **Test** | Deployment Changes deployed

Go to Anything (Ctrl-P) lambda\_function Execution result

Execution results

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

**Delete function neel-t21**

Deleting a function permanently removes the function code. The related logs and roles are retained in your account.

Cancel **Delete**

## Empty bucket

**Empty bucket** [Info](#)

⚠ Emptying the bucket deletes all objects in the bucket and cannot be undone.  
Objects added to the bucket while the empty bucket action is in progress might be deleted.  
To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

Learn more [\[?\]](#)

If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. Learn more [\[?\]](#)

Permanently delete all objects in bucket "neel-patel-t21-82"?

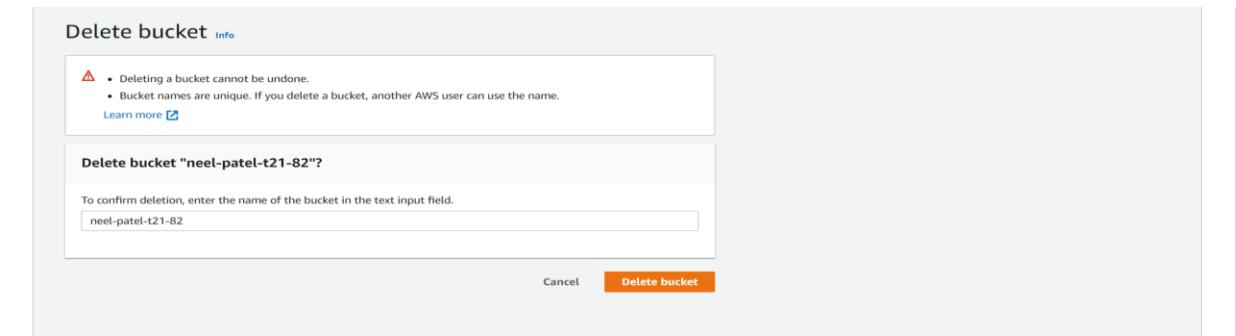
To confirm deletion, type *permanently delete* in the text input field.

permanently delete

Cancel **Empty**

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

## Delete bucket-



## CONCLUSION:

In this assignment, we covered the core concepts and terminologies of AWS Lambda, explored its practical applications, and gained an understanding of how it integrates with various AWS services.

**Roll No:- 42**

**Name :-Piyush Hingorani**

**Batch:- T13**

**Date:- 05/09/2023**

## **ASSIGNMENT-10**

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

**LO MAPPED: LO1, LO5**

**THEORY:-**

### **DYNAMO DB**

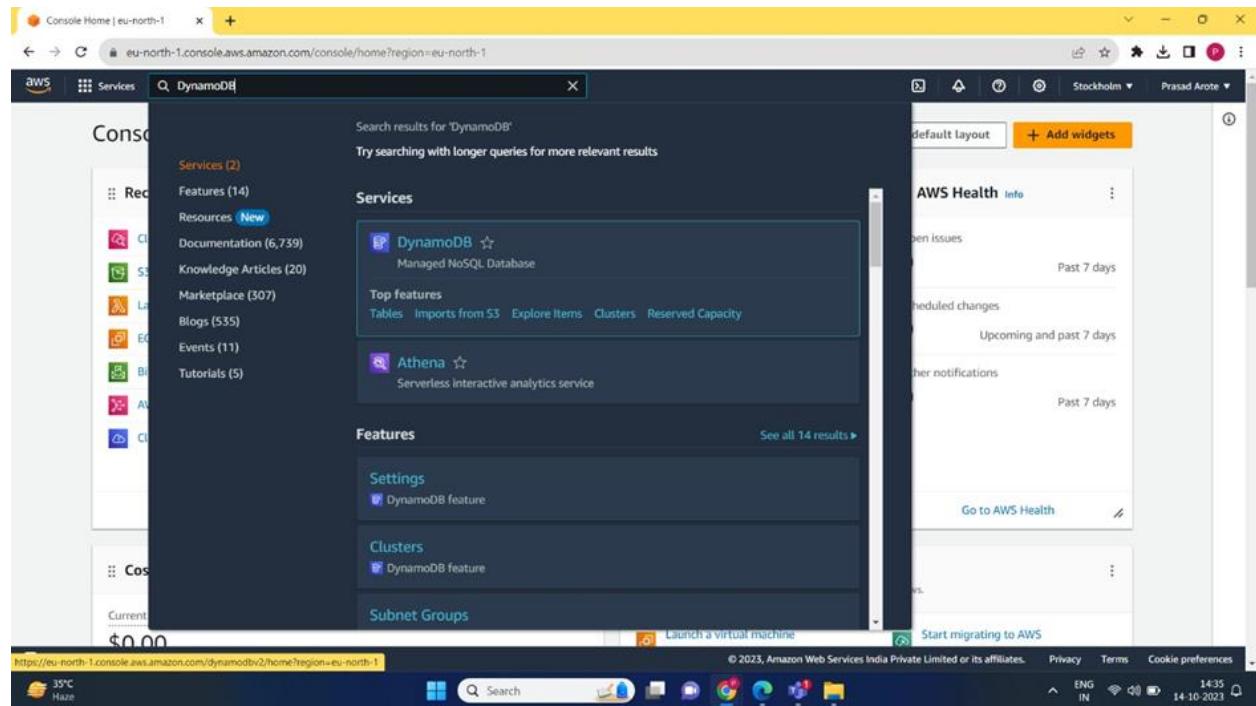
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics

DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

## Steps :-

### 1) Login to AWS account and search for DynamoDB in search bar



### 2) Click on DynamoDB option shown above and then click on create table

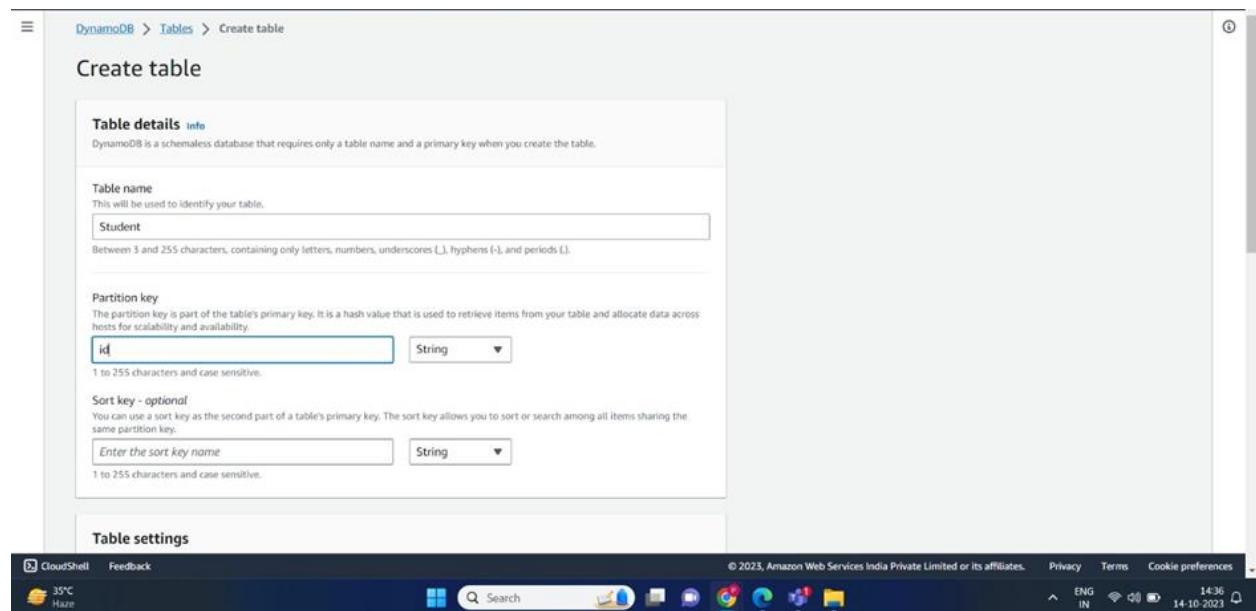


Table details info  
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name  
This will be used to identify your table.  
 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.

String

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.

String

Table settings

**3)Then search IAM in the search box above and create a new role , give AmazonDynamoFullAccess permission to created user**

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Select trusted entity Info

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

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

Service or use case Info  
Lambda

Choose a use case for the specified service.

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

CloudShell Feedback 35°C Haze Search ENG IN 14:40 14-10-2023

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Add permissions Info

Permissions policies (1/887) Info  
Choose one or more policies to attach to your new role.

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon Dynamo...
<input type="checkbox"/> AmazonDynamoDBReadOnlyAccess	AWS managed	Provides read only access to Amazon Dyn...
<input type="checkbox"/> AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to DynamoD...
<input type="checkbox"/> AWSLambdaInvocation-DynamoDB	AWS managed	Provides read access to DynamoDB Strea...

Filter by Type  
Q dyna All types 4 matches

Set permissions boundary - optional

Cancel Previous Next

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

CloudShell Feedback 35°C Haze Search ENG IN 14:41 14-10-2023

The screenshot shows two overlapping browser windows. The top window is titled 'Role details' and is on 'Step 2: Add permissions'. It shows a role named 'prasad\_admin' with a trust policy allowing Lambda to assume it. The bottom window is titled 'Identity and Access Management (IAM)' and shows a list of roles. The 'prasad\_admin' role is listed, created 75 days ago, with AWS Lambda as the trusted entity.

**Role details**

**Step 2: Add permissions**

**Step 3: Name, review, and create**

**Role name**  
Enter a meaningful name to identify this role.  
**prasad\_admin**

**Description**  
Add a short explanation for this role.  
Allows Lambda functions to call AWS services on your behalf.

**Step 1: Select trusted entities**

**Trust policy**

```

1+ [
2+   "Version": "2012-10-17",
3+   "Statement": [
4+     {
5+       "Effect": "Allow",
6+       "Action": [
7+         "sts:AssumeRole"
8+       ],
9+       "Principal": [
10+         {
11+           "Service": [
12+             "lambda.amazonaws.com"
13+           ]
14+         }
15+       ]
16+     }
17+   ]
18+
19+ ]

```

**Identity and Access Management (IAM)**

**Role prasad\_admin created.**

**Roles (9) Info**

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

<input type="checkbox"/> Role name	Trusted entities	Last activity
<a href="#">AWSCloud9SSMAccessRole</a>	AWS Service: ec2, <a href="#">and 1 more</a>	75 days ago
<a href="#">AWSServiceRoleForApplicationAutoScaling_DynamoDBTable</a>	AWS Service: dynamodb.application	-
<a href="#">AWSServiceRoleForAWSCloud9</a>	AWS Service: cloud9 <a href"="">(Service-Linked)</a>	75 days ago
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support <a href="">(Service-Linked)</a>	-
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor <a href="">(Service-Linked)</a>	-
<a href="#">lambdafunc1-role-11c5fj6u</a>	AWS Service: lambda	1 hour ago
<a href="#">prasad_admin</a>	AWS Service: lambda	-
<a href="#">PyRole</a>	AWS Service: lambda	68 days ago
<a href="#">Runpython</a>	AWS Service: lambda	68 days ago

#### 4) Search Lambda in search box and click on it , then create a new lambda function

The screenshot shows the AWS Lambda console interface. At the top, a search bar displays "Search results for 'lambda'" and "Try searching with longer queries for more relevant results". Below the search bar, there are two main sections: "Services" and "Features".

**Services** (7)

- Documentation (9,951)
- Knowledge Articles (20)
- Marketplace (664)
- Blogs (1,017)
- Events (13)
- Tutorials (6)

**Features**

- Local processing

On the right side, there is a sidebar titled "AWS Health" with a "Info" button. It displays various status metrics:

- Open issues: Past 7 days
- Scheduled changes: Upcoming and past 7 days
- Other notifications: Past 7 days

At the bottom of the page, there is a navigation bar with links for "CloudShell", "Feedback", "Privacy", "Terms", and "Cookie preferences". The URL in the address bar is <https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1>.

**Create function** Info

Choose one of the following options to create your function.

Author from scratch  
Start with a simple Hello World example.

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

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

Browse serverless app repository  
Deploy a sample Lambda application from the AWS Serverless Application Repository.

**Basic information**

Function name  
Enter a name that describes the purpose of your function.  
addstudentdata

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 invoke functions. You can customize this default role later when adding triggers.

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 14:46 14-10-2023

**Function name**  
Enter a name that describes the purpose of your function.  
**addstudentdata**  
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**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
**Use an existing role**   
 Create a new role with basic Lambda permissions  
 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.  
**prasad\_admin**

[View the prasad\\_admin role](#) on the IAM console.

**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](#).  
**Use an existing role**   
 Create a new role with basic Lambda permissions  
 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.  
**prasad\_admin**

[View the prasad\\_admin role](#) on the IAM console.

**Advanced settings**

**Enable Code signing** [Info](#)  
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

**Enable function URL** [Info](#)  
Use function URLs to assign HTTP(S) endpoints to your Lambda function.

**Enable tags** [Info](#)  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

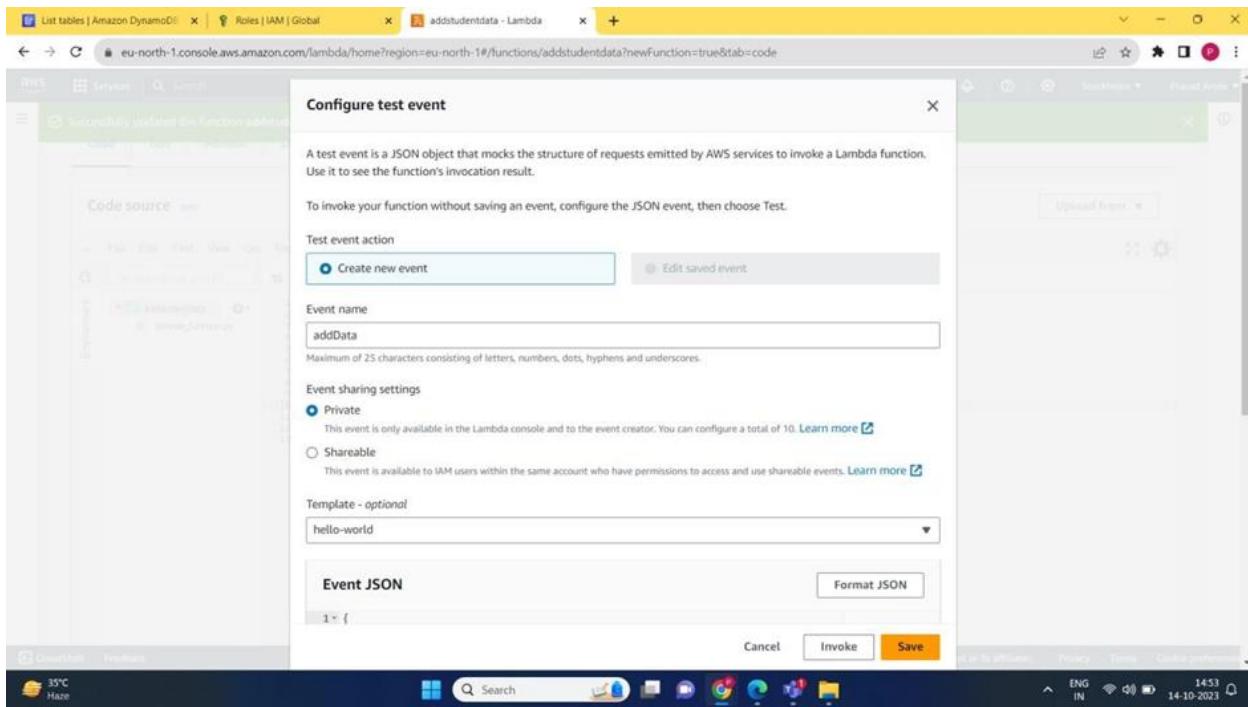
**Enable VPC** [Info](#)

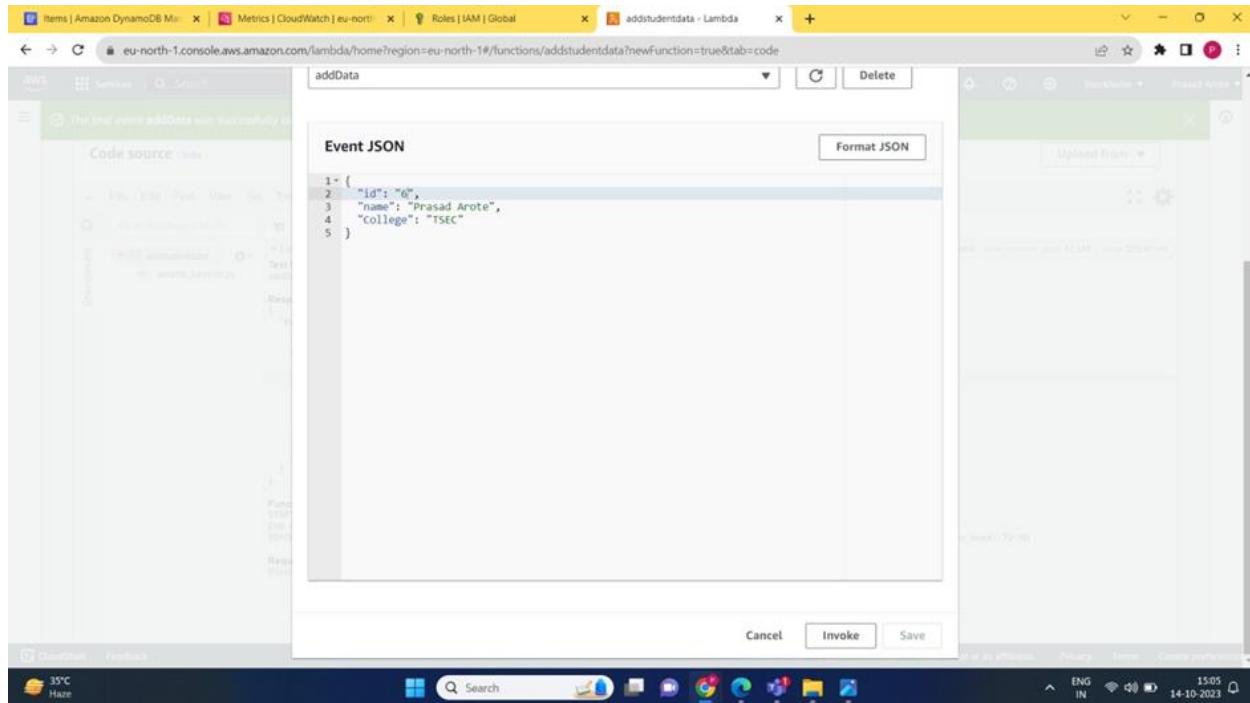
## 5) Write the following code in code source

The screenshot shows the AWS Lambda function editor. The top bar displays a green success message: "Successfully updated the function addstudentdata." Below the bar, the navigation menu includes Code, Test, Monitor, Configuration, Aliases, and Versions. The main area is titled "Code source" with tabs for Info, Code, Test, and Deploy. The "Test" tab is selected. On the left, there's an "Environment" sidebar with "addstudentdata" and "lambda\_function" listed. The central code editor contains the following Python code:

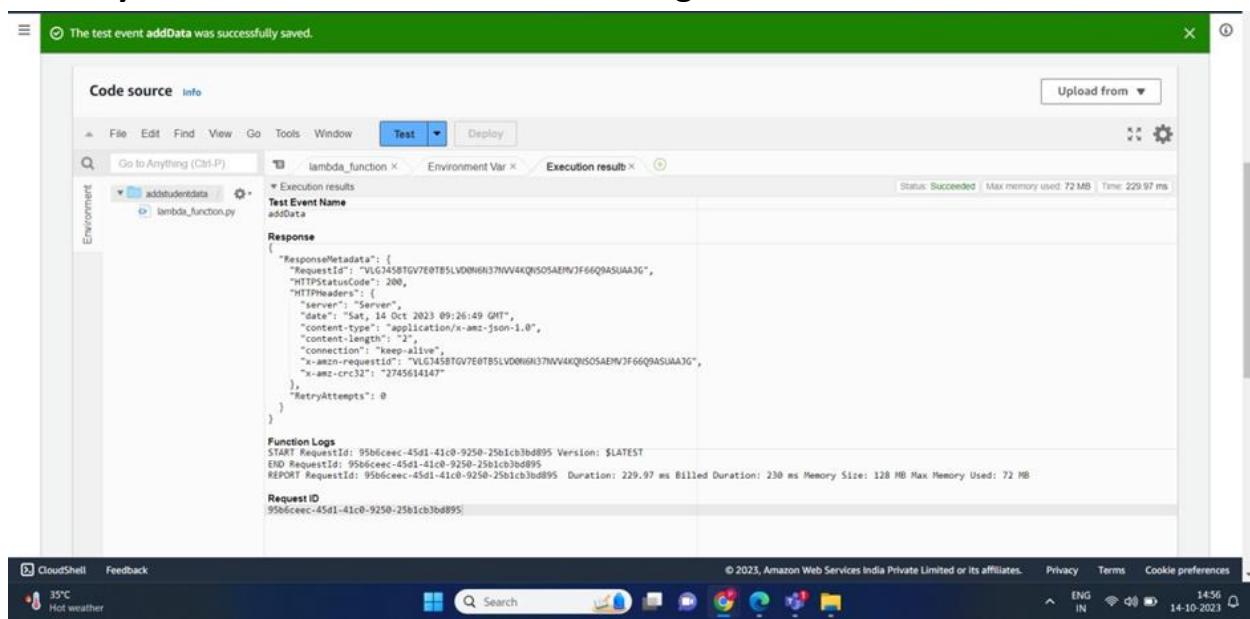
```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # TODO implement
6     client_dynamo=boto3.resource('dynamodb')
7     table=client_dynamo.Table('Student')
8
9     response=table.put_item(Item=event)
10
11     return response
12
13
```

## 6) Configure the test event and save





**7) Run the test and afterwards go to the DynamoDB>Explore items> Student where you can see the record inserted using lambda function.**



The screenshot shows the Amazon DynamoDB console. On the left, a sidebar menu includes options like Dashboard, Tables, Update settings, Explore items (which is selected), PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, and Settings. The main area is titled 'Scan or query items' with tabs for 'Scan' (selected) and 'Query'. It shows a table named 'Student' with 'All attributes' selected. A success message at the bottom says 'Completed. Read capacity units consumed: 0.5'. The results table has columns for id (String), College, and name. One row is shown: id is 6, College is TSEC, and name is Prasad Arote.

	id (String)	College	name
6	TSEC	Prasad Arote	

## CONCLUSION:-

Learnt about Amazon DynamoDB database service and inserted data into DynamoDB database by creating a new user , granting him permissions and then using a lambda function

**RollNo :- 42**

**Name :- Piyush Hingorani**

**Batch :- T13**

**Date:- 13/10/2023**

### **Written Assignment 1:Study of Kubernetes**

#### **Q1) What security measures can be taken while using kubernetes ?**

There are a number of security measures that can be taken while using Kubernetes. Some of the most important include:

**1)Use Role-Based Access Control (RBAC):** RBAC allows you to define who has access to the Kubernetes API and what permissions they have. This is essential for preventing unauthorized access to your cluster and its resources.

**2)Use third-party authentication for the API server:** This allows you to integrate Kubernetes with an existing identity provider, such as GitHub or Okta. This can make it easier to manage user accounts and permissions.

**3)Protect etcd with TLS and a firewall:** etcd is a distributed key-value store that stores the state of your Kubernetes cluster. It is a critical component, so it is important to protect it from unauthorized access.

**4)Isolate Kubernetes nodes:** Kubernetes nodes are the machines that run your containerized applications. It is important to isolate them from the rest of your network to prevent attackers from gaining access to your cluster.

**5)Monitor network traffic to limit communications:** Kubernetes workloads can communicate with each other over the network. It is important to monitor this traffic and limit it to only the necessary communication paths.

**6)Use process whitelisting:** Process whitelisting allows you to define which processes are allowed to run on your Kubernetes nodes. This can help to prevent attackers from running malicious code on your cluster.

**7)Turn on audit logging:** Audit logging records all activity on your Kubernetes cluster. This can help you to detect and investigate security incidents.

**8)Keep Kubernetes up to date:** The Kubernetes team regularly releases security updates. It is important to keep your Kubernetes cluster up to date to protect against known vulnerabilities.

**9)Lock down the Kubelet:** The Kubelet is a service that runs on each Kubernetes node and is responsible for managing pods. It is important to lock down the Kubelet to prevent attackers from gaining control of your nodes.

In addition to these general security measures, there are a number of other specific things you can do to secure your Kubernetes cluster, such as:

**10)Use Kubernetes namespaces to isolate your workloads:** Namespaces allow you to group your workloads together and isolate them from each other. This can help to prevent attackers from moving laterally between your workloads.

**11)Use Pod Security Policies (PSPs) to restrict the privileges of your pods:** PSPs allow you to define what resources and privileges your pods are allowed to use. This can help to prevent attackers from gaining access to sensitive data or running malicious code on your cluster.

**12)Use a service mesh to manage network traffic between your workloads:** A service mesh can help you to secure and manage the network traffic between your workloads. This can help to prevent attackers from communicating with your workloads or eavesdropping on their traffic.

**13) Use a security scanner to scan your container images for vulnerabilities:** A security scanner can help you to identify and fix vulnerabilities in your container images before they are deployed to your cluster.

**14) Implement a security incident response plan:** A security incident response plan will help you to respond to security incidents in a timely and effective manner.

By following these security measures, you can help to protect your Kubernetes cluster and its workloads from attack.

## **Q2) What are the three security techniques used to protect data ?**

The three most important security techniques used to protect data are:

- 1) Encryption**
- 2) Access control**
- 3) Backup and recovery**

Encryption is the process of converting data into a format that cannot be read without a secret key. This makes data unreadable to unauthorized individuals, even if they have access to it. Encryption can be used to protect data at rest, in transit, and in use.

Access control is the process of restricting access to data to authorized individuals. This can be done using a variety of methods, such as passwords, multi-factor authentication, and role-based access control (RBAC). Access control is important for preventing unauthorized individuals from accessing and modifying data.

Backup and recovery is the process of creating copies of data and storing them in a secure location. This is important for protecting data from loss or corruption. Backup and recovery plans should be regularly tested to ensure that they are working properly.

In addition to these three core security techniques, there are a number of other security measures that can be used to protect data, such as network security, physical security, and security awareness training.

Here are some examples of how these three security techniques can be used to protect data:

**Encryption:** A company can encrypt its customer database to protect it from unauthorized access, even if the database is compromised.

**Access control:** A hospital can use RBAC to restrict access to patient medical records to authorized personnel, such as doctors and nurses.

**Backup and recovery:** A government agency can regularly back up its financial data to a secure location in case of a cyberattack or natural disaster.

By using these security techniques, organizations can help to protect their data from unauthorized access, modification, and loss.

### **Q3) How do you expose a service using ingress in Kubernetes?**

To expose a service using Ingress in Kubernetes, you need to create an **Ingress resource**. An Ingress resource specifies the rules for routing traffic to your services.

To create an Ingress resource, you can use the following command:

```
kubectl create ingress <ingress-name>
```

The Ingress resource must specify the following:

- 1) The rules for routing traffic to your services.**
- 2) The hostname or IP address that traffic will be routed to.**
- 3) The port that traffic will be routed to.**

For example, the following Ingress resource exposes a service named my-service on port 80:

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
    - http:
        paths:
          - path: /
            pathType: Prefix
        backend:
          service:
            name: my-service
            port: 80
```

Once you have created the Ingress resource, you can access the service at the hostname or IP address specified in the Ingress resource. For example, if the Ingress resource specifies the hostname my-service.example.com, you can access the service at my-service.example.com on port 80.

You can also use Ingress to expose multiple services on the same hostname or IP address. To do this, you can specify multiple rules in the Ingress resource. For example, the following Ingress resource exposes two services, my-service and my-other-service, on port 80:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
    - http:
        paths:
          - path: /
```

```
pathType: Prefix
backend:
service:
name: my-service
port: 80
- http:
  paths:
  - path: /other
pathType: Prefix
backend:
service:
name: my-other-service
port: 80
```

Ingress is a powerful tool for exposing services in Kubernetes. It allows you to expose services on a specific hostname or IP address, and to expose multiple services on the same hostname or IP address.

#### **Q4) Which service protocol does Kubernetes ingress expose ?**

Ingress is a Kubernetes resource that allows you to expose services running in a cluster to external traffic. **Ingress can expose services through either HTTP or HTTPS.**

To expose a service using Ingress, you need to create an Ingress resource. An Ingress resource specifies the rules for routing traffic to your services.

When you create an Ingress resource, you need to specify the following:

- 1) The rules for routing traffic to your services.**
- 2) The hostname or IP address that traffic will be routed to.**
- 3) The port that traffic will be routed to.**

The Ingress resource also specifies the service protocol, which can be either HTTP or HTTPS.

To expose a service using HTTP, you need to specify the http protocol in the Ingress resource. For example, the following Ingress resource exposes a service named my-service on port 80 using HTTP:

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

To expose a service using HTTPS, you need to specify the https protocol in the Ingress resource. For example, the following Ingress resource exposes a service named my-service on port 443 using HTTPS:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  tls:
  - hosts:
```

```
- my-service.example.com
secretName: my-tls-secret

rules:
- http:
  paths:
    - path: /
  pathType: Prefix
  backend:
    service:
      name: my-service
      port: 443
```

Once you have created the Ingress resource, you can access the service at the hostname or IP address specified in the Ingress resource. For example, if the Ingress resource specifies the hostname my-service.example.com, you can access the service at my-service.example.com on port 80.

You can also use Ingress to expose multiple services on the same hostname or IP address. To do this, you can specify multiple rules in the Ingress resource.

**Roll No:-42**  
**Batch:- T13**  
**Name :-Piyush Hingorani**

## **WRITTEN ASSIGNMENT-2**

### **Q.1 How to deploy Lambda function on AWS?**

=>Deploying a Lambda function on AWS involves several steps. Here's a detailed guide on how to deploy a Lambda function:

#### **Prerequisites:**

An AWS account.

The AWS Command Line Interface (CLI) installed and configured with appropriate permissions.

Your Lambda function code packaged as a ZIP archive or uploaded to an Amazon S3 bucket.

Familiarity with the programming language and runtime you're using for your Lambda function (e.g., Node.js, Python, Java, etc.).

#### **Step-by-Step Guide to Deploying a Lambda Function:**

Create or Prepare Your Lambda Function Code:

Write your Lambda function code or prepare it if you haven't already. Ensure it follows the AWS Lambda function structure, including the handler function.

Package Your Code:

If your function code consists of multiple files or dependencies, package it as a ZIP archive. Make sure that the primary function handler is at the top level of the archive.

Create an Execution Role (if needed):

Lambda functions often need permissions to interact with other AWS services. Create an AWS Identity and Access Management (IAM) role that grants the necessary permissions to your Lambda function. The role should have policies attached that allow access to AWS resources, like S3, DynamoDB, or others.

Upload Code to Amazon S3 (if needed):

If your deployment package is larger than 3 MB, you will need to upload it to an Amazon S3 bucket. Make sure your Lambda function has permissions to access this bucket.

#### **Deploy the Lambda Function:**

You can deploy a Lambda function using the AWS Management Console, AWS CLI, AWS SDKs, or AWS CloudFormation. Here's how to deploy using the AWS CLI:

Open a terminal and run the following AWS CLI command to create your Lambda function:

```
aws lambda create-function --function-name MyFunctionName --runtime nodejs14.x --role arn:aws:iam::123456789012:role/MyRole --handler index.handler --zip-file fileb://function.zip
```

Replace MyFunctionName with your desired function name.

Specify the correct runtime for your function (e.g., nodejs14.x for Node.js 14).

Use the --role option with the ARN of the IAM role you created.

Specify the --handler option with the name of your handler function.

Use --zip-file to reference your deployment package.

### **Test Your Lambda Function:**

After creating your Lambda function, you can test it using the AWS Management Console, AWS CLI, or an SDK. Make sure it works as expected.

### **Configure Event Sources (if needed):**

If your Lambda function is triggered by events from other AWS services (e.g., S3, SNS, API Gateway), configure these event sources in the AWS Management Console.

### **Set Up Environment Variables (if needed):**

If your function relies on environment variables, configure these in the Lambda function's configuration.

### **Deploy Updates (if needed):**

If you make changes to your function code or configuration, you can update the Lambda function by uploading a new deployment package, and the changes will be applied.

### **Monitor and Troubleshoot:**

Use AWS CloudWatch and other monitoring tools to track the performance and behavior of your Lambda function. You can also view logs and error messages in CloudWatch Logs to troubleshoot issues.

### **Scale and Manage Your Function:**

AWS Lambda automatically scales your function to handle incoming requests. You can configure concurrency limits, timeout settings, and other properties to manage its behavior.

### **Cost Management:**

Keep an eye on the costs associated with your Lambda function, as you'll be billed based on the number of requests and duration of execution. Utilize cost management tools and set up billing alerts to avoid unexpected charges.

Deploying a Lambda function on AWS is a straightforward process, but it's important to pay attention to configuration, permissions, and monitoring to ensure your function operates as expected in a production environment.

## **Q.2 What are the deployment options for AWS Lambda?**

AWS Lambda offers several deployment options, each suited for different use cases and development workflows. Here are the primary deployment options for AWS Lambda, explained in detail:

### **Upload Deployment Package:**

This is the simplest deployment option. You create a ZIP archive that contains your function code and dependencies. Then, you manually upload it when creating or updating your Lambda function. This approach is suitable for small functions or when you need full control over your deployment process.

Steps:

Create a ZIP archive containing your function code and dependencies.

Use the AWS Management Console, AWS CLI, or AWS SDKs to create or update your Lambda function, providing the ZIP archive as the deployment package.

### **S3 Bucket Deployment:**

For larger deployment packages or for separating the deployment process from the Lambda function creation or update, you can store your deployment package in an Amazon S3 bucket. When you create or update a Lambda function, you specify the S3 bucket location for your deployment package.

Steps:

Upload your deployment package to an S3 bucket.

Use the AWS Management Console, AWS CLI, or AWS SDKs to create or update your Lambda function, specifying the S3 bucket location.

### **AWS Serverless Application Model (SAM):**

SAM is an open-source framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way to define the Amazon API Gateway APIs, AWS Lambda functions, and Amazon DynamoDB tables needed by your serverless application. You can define your Lambda function configurations and deployment details in a template.yaml file, which SAM uses to create and deploy the function.

Steps:

Create a template.yaml file that defines your Lambda function and its dependencies.

Use the AWS SAM CLI to package and deploy your serverless application to AWS. This CLI tool simplifies packaging and deployment, including the creation of Amazon S3 buckets for deployment.

### **Lambda Layers:**

Lambda Layers allow you to separate your function code from its dependencies. You can create a custom Layer with your dependencies and then reference it in your

Lambda function. When updating the dependencies, you only need to update the Layer, reducing the size and complexity of the deployment package.

Steps:

Create a Lambda Layer containing your dependencies.

Reference the Layer in your Lambda function configuration.

When updating dependencies, update the Layer without changing your function code.

Container Images (AWS Lambda for Containers):

AWS Lambda added support for running functions in container images. With this option, you build a container image with your function code, dependencies, and any runtime environment you need. You can use your preferred container registry to store the image. Lambda manages the container execution, scaling, and resource allocation.

Steps:

Build a Docker container image with your function code and dependencies.

Push the image to a container registry (e.g., Amazon ECR, Docker Hub).

Create a Lambda function using the image from your container registry.

Continuous Deployment Tools:

You can integrate AWS Lambda deployment into your continuous deployment pipelines using tools like AWS CodePipeline, AWS CodeBuild, Jenkins, or any other CI/CD solution. This approach automates the deployment process, allowing you to push changes to your function code in a version-controlled manner.

Steps:

Set up a CI/CD pipeline that monitors your code repository.

Configure the pipeline to build and deploy your Lambda function when changes are detected.

Each of these deployment options has its own advantages and is suitable for different scenarios. The choice depends on factors such as the size and complexity of your function, your development workflow, and whether you require more granular control over your deployments.

### **Q.3 What are the 3 full deployment modes that can be used for AWS?**

In the context of AWS, there are three primary full deployment modes, each offering a distinct approach to deploying and managing applications. These modes are designed to accommodate different use cases and operational requirements. Let's explore each of them in detail:

#### **EC2-Based Deployment:**

##### **Description:**

EC2-based deployment is the traditional deployment mode in AWS where you provision and manage virtual machines (EC2 instances) to run your applications. In this mode,

you have full control over the underlying infrastructure, including the choice of instance types, operating systems, and configuration. This mode is ideal for applications that require a high degree of customization or when you have legacy systems that need to run in a traditional virtualized environment.

#### **Use Cases:**

Running legacy applications or software that can't be containerized.  
Applications with complex network configurations or specific hardware requirements.  
When you need to manage the entire stack from the operating system up.

#### **Key Components:**

Amazon Elastic Compute Cloud (EC2) instances.  
Amazon Virtual Private Cloud (VPC) for network isolation.  
Elastic Load Balancers for distributing traffic.  
Amazon RDS or other database services for data storage.

#### **Benefits:**

Complete control over infrastructure.  
Compatibility with a wide range of software.  
Ability to run non-containerized or legacy applications.

#### **Challenges:**

Manual scaling and management of EC2 instances.  
More operational overhead compared to serverless or container-based solutions.  
Limited automation compared to other deployment modes.

#### **Serverless Deployment:**

##### **Description:**

Serverless deployment is a modern cloud computing paradigm in which you focus solely on writing code (usually in the form of functions) and let the cloud provider manage all the underlying infrastructure. AWS Lambda is a key component of this approach, allowing you to run code in response to events without provisioning or managing servers. Serverless computing is highly scalable and event-driven.

#### **Use Cases:**

Building scalable, event-driven applications.  
Microservices architecture.  
Real-time data processing and analysis.  
Reducing operational overhead by offloading infrastructure management.

#### **Key Components:**

AWS Lambda for running code in response to events.  
Amazon API Gateway for exposing APIs.  
Various AWS services for data storage and processing.  
Amazon EventBridge or Amazon S3 event triggers for event-driven applications.

**Benefits:**

Auto-scaling and high availability.  
Minimal operational overhead.  
Pay-per-use pricing.  
Easy integration with other AWS services.

**Challenges:**

Stateless execution, which may require workarounds for stateful applications.  
Limited runtime options compared to EC2 instances.  
Function duration and resource limits.

**Container-Based Deployment:****Description:**

Container-based deployment leverages containerization technology to package applications and their dependencies into a consistent and portable format. AWS offers Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS) for managing containers in a scalable, automated, and highly available manner. This deployment mode is ideal for containerized applications, microservices, and orchestrating container workloads.

**Use Cases:**

Microservices architecture.  
Porting and running containerized applications.  
Managing applications that need to scale and have dependencies isolated in containers.

**Key Components:**

Amazon ECS or Amazon EKS for managing containers.  
Amazon ECR for container registry.  
Docker for building and running containers.  
Kubernetes for container orchestration (EKS).

**Benefits:**

Scalability and flexibility of containerization.  
Portability and consistency of containers.  
Advanced orchestration and management features in Kubernetes.

**Challenges:**

Container management complexity.  
Learning curve for orchestrators like Kubernetes.  
Ongoing operational overhead.

These three full deployment modes represent different approaches to deploying and managing applications in AWS. Your choice should be based on your specific requirements, including the nature of your applications, your scalability needs, and your desired level of operational control. It's not uncommon for organizations to use a

combination of these deployment modes, depending on their application portfolio and use cases.

#### **Q.4 What are the 3 components of AWS Lambda?**

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS) that allows you to run code in response to events without the need to manage servers. AWS Lambda has three core components that work together to enable serverless compute capabilities:

##### **Lambda Function:**

**Description:** A Lambda function is the core unit of execution in AWS Lambda. It represents your code, which can be written in various programming languages such as Node.js, Python, Java, Go, and more. A Lambda function is a small, self-contained piece of code that can perform a specific task when triggered by an event. It can be as simple as a few lines of code or more complex, and it typically follows a specific structure, including a handler function that AWS Lambda invokes when an event occurs.

**Use Cases:** Lambda functions are used for a wide range of purposes, including data processing, automation, real-time file processing, API endpoints, and more. They are particularly well-suited for building serverless applications and microservices.

##### **Key Characteristics:**

Small, single-purpose code.

Stateless (no persistent storage of data between invocations).

Event-driven execution.

Automatic scaling and resource allocation.

##### **Event Source:**

**Description:** Event sources are triggers that initiate the execution of a Lambda function. These sources can be various AWS services or external systems that generate events. When an event occurs, AWS Lambda is automatically invoked, and the event data is passed to the Lambda function for processing. AWS Lambda supports a wide range of event sources, including AWS services like Amazon S3, Amazon DynamoDB, AWS SNS, and custom event sources using AWS Step Functions.

**Use Cases:** Event sources enable Lambda functions to respond to changes in data, incoming requests, system events, and more. This makes them suitable for building event-driven applications and automating workflows.

##### **Key Characteristics:**

Diverse sources, including AWS services and custom events.

Real-time event triggering.

Integration with various AWS services.

### **Execution Environment:**

**Description:** The execution environment is the runtime environment where Lambda functions run. AWS Lambda manages and provisions these environments dynamically as needed, and it abstracts the underlying infrastructure from developers. The execution environment includes the compute resources (CPU, memory) and the network configuration necessary for the function's execution. The environment automatically scales with incoming event load.

**Use Cases:** The execution environment is responsible for ensuring that Lambda functions can run in a scalable and highly available manner without the need for manual provisioning or management. It enables the on-demand execution of code.

### **Key Characteristics:**

Automatic provisioning and scaling.

Abstracts infrastructure management.

Resource allocation (memory and CPU) defined per function.

Isolation between concurrent executions.

Together, these three components form the foundation of AWS Lambda. A Lambda function processes events from various event sources within an execution environment provided by AWS Lambda. The serverless nature of Lambda, where you focus on code rather than infrastructure, allows you to build applications that are highly scalable and responsive to real-time events with minimal operational overhead. This serverless model is particularly valuable for organizations looking to optimize resource utilization and reduce infrastructure management complexities.