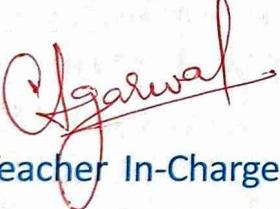


**Thadomal Shahani Engineering College**  
Bandra (W.), Mumbai- 400 050.

**❖ CERTIFICATE ❖**

Certify that Mr./Miss Niranjan Rajesh Joshi  
of IT Department, Semester IV with  
Roll No. 51 has completed a course of the necessary  
experiments in the subject Advance 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.
17	To study and perform setup of AWS EC2 service and launch an EC2 Instance		18/07/23	
27	To study and perform the setup of AWS Cloud9 service and launch a python program in Cloud9		25/07/23	
37	To study AWS S3 service and create bucket for hosting static web application		01/08/23	<i>P. J. Agarwal 20/10/23</i>
47	To study AWS CodePipeline and deploy web application using AWS CodePipeline		08/08/23	
57	To understand Kubernetes cluster Architecture, install and Spinup kubernetes cluster on Linux machines / cloud platform		13/08/23	
67	To understand terraform lifecycle and to build, change and destroy AWS infrastructure using terraform		22/08/23	
77	To perform static analysis on python programs and analyse SAST process		29/08/23	
87	To understand continuous monitoring using Nagios		12/09/23	
97	To understand AWS Lambda function and create a Lambda function using Python to log "An image has been added" message once file is added to S3 bucket		05/09/23	<i>P. J. Agarwal 20/10/23</i>
107	To create Lambda function using python for adding data to Dynamo DB database		05/09/23	
117	Assignment 1		01/08/23	
127	Assignment 2		13/10/23	

**Name :-Niranjan Rajesh Joshi**  
**Roll No:- 2105051**  
**Batch:- T13**  
**Date Of Performance :- 25/07/2023**

## **Experiment 1**

**Aim:-**Study and create AWS EC2 instance

**LO Mapped:- L01**

**Theory:-**

EC2 stands for **Amazon Elastic Compute Cloud**. Amazon EC2 is a **web service** that provides **resizable compute capacity in the cloud**. Amazon EC2 reduces the time required to obtain and **boot new user instances** to minutes rather than in older days, if you need a server then you had to put a purchase order, and cabling is done to get a new server which is a very time-consuming process. Now, Amazon has provided an EC2 which is a **virtual machine** in the cloud that completely changes the industry.

**Steps to create AWS account :-**

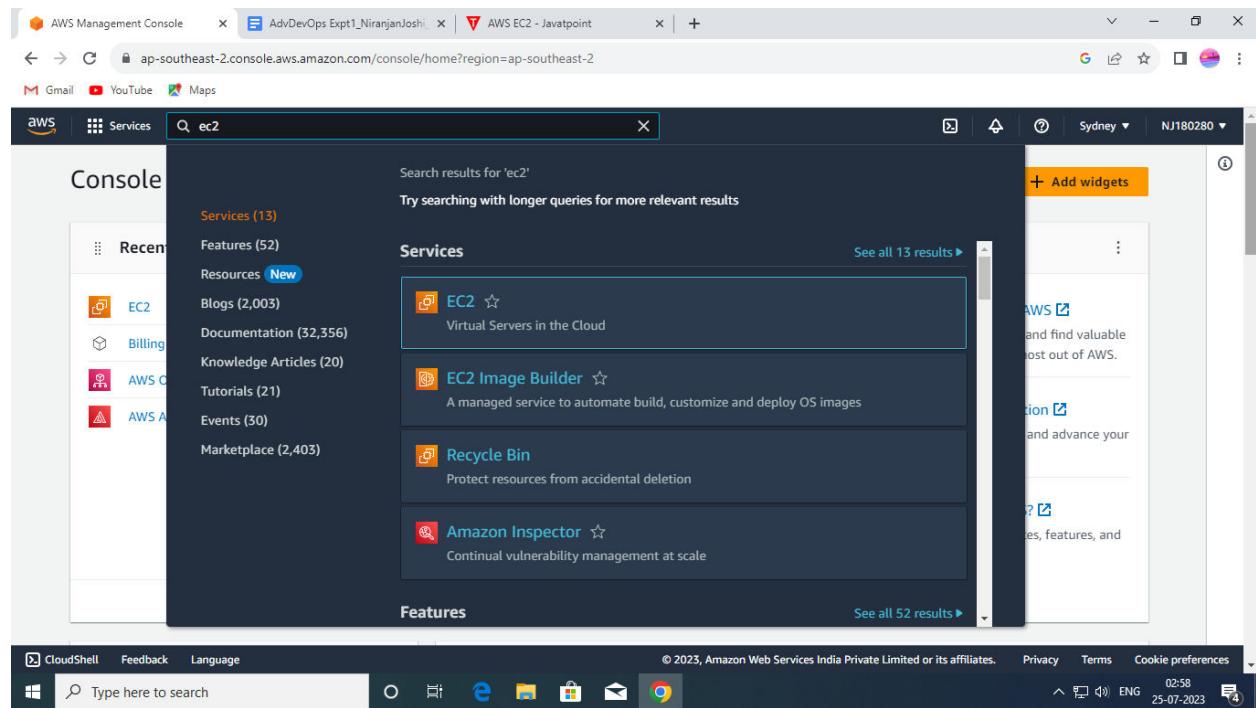
- 1)** Open the Amazon Web Services home page .
- 2)** Choose Create an AWS account.
- 3)** Enter your account information, and then choose Verify email address. This will send a verification code to your specified email address.
- 4)** Enter your verification code, and then choose Verify.

- 5)** Enter a strong password for your root user, confirm it, and then choose Continue. AWS requires that your password meet the following conditions:
- 6)** It must have a minimum of 8 characters and a maximum of 128 characters. It must include a minimum of three of the following mix of character types: uppercase, lowercase, numbers, and ! @ # \$ % ^ & \* () <> [] {} | \_+= symbols. It must not be identical to your AWS account name or email address.
- 7)** Choose Business or Personal. Personal accounts and business accounts have the same features and functions. Enter your company or personal information. Read and accept the AWS Customer Agreement. Be sure that you read and understand the terms of the AWS Customer Agreement.
- 8)** Choose Continue. At this point, you'll receive an email message to confirm that your AWS account is ready to use. You can sign in to your new account by using the email address and password you provided during sign up. However, you can't use any AWS services until you finish activating your account.
- 9)** Enter the information about your payment method, and then choose Verify and Continue. If you want to use a different billing address for your AWS billing information, choose Use a new address.
- 10)** Enter your country or region code from the list, and then enter a phone number where you can be reached in the next few minutes.
- 11)** When the automated system contacts you, enter the PIN you receive and then submit. Select one of the available AWS Support plans. For a description of the available Support plans and their benefits, see Compare AWS Support plans.
- 12)** Choose Complete sign up. A confirmation page appears that indicates that your account is being activated. Check your email and spam folder for an email message that confirms your account was activated.

## Steps (to create an EC2 instance) :-

1) Sign in to AWS account

2) In above search bar, search for EC2 service as shown below and then click on EC2 :



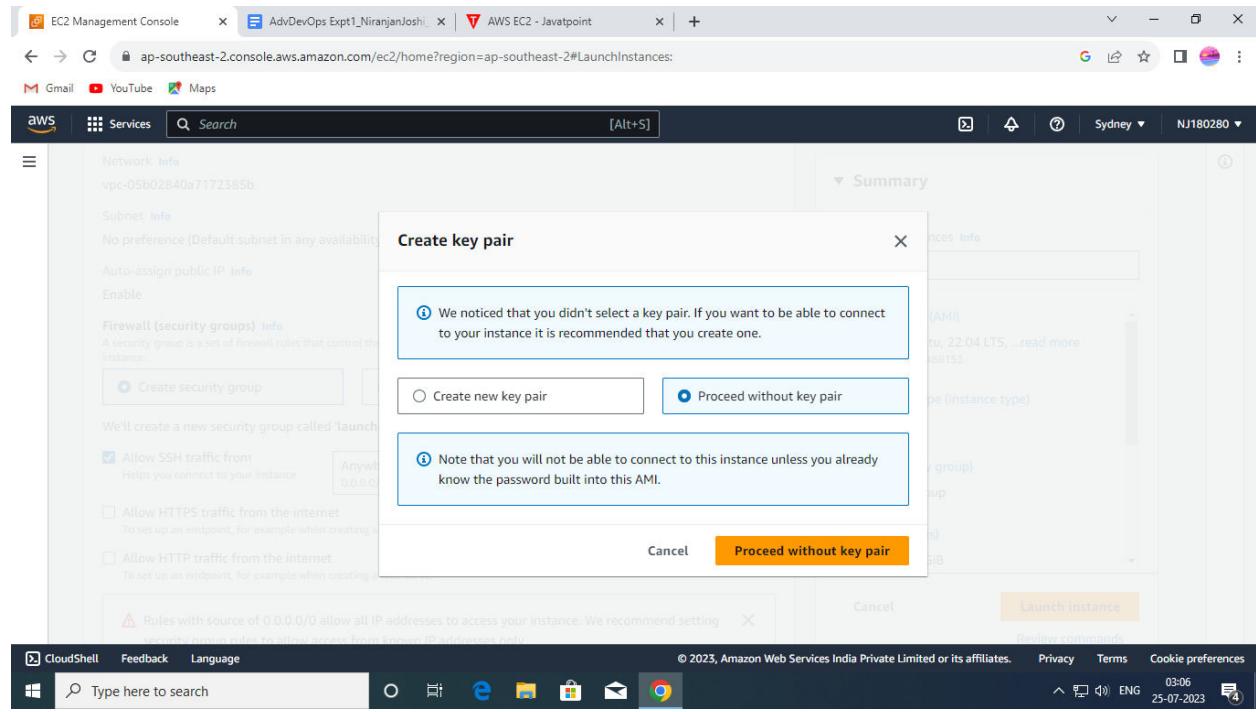
3) Now EC2 dashboard will be shown and then scroll below and click on Launch Instance to create a new instance

The screenshot shows the AWS EC2 Management Console dashboard. On the left, a sidebar lists various EC2-related services: Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area has three main sections: 'Launch instance' (with a 'Launch instance' button and 'Migrate a server' link), 'Service health' (showing the region as Asia Pacific (Sydney) and the status as 'This service is operating normally'), and 'Scheduled events' (listing 'Asia Pacific (Sydney)' with 'No scheduled events'). A sidebar on the right provides information about T4g instances and 10 things to do to reduce AWS costs, along with links to the AWS Health Dashboard and Amazon GuardDuty Malware Protection. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

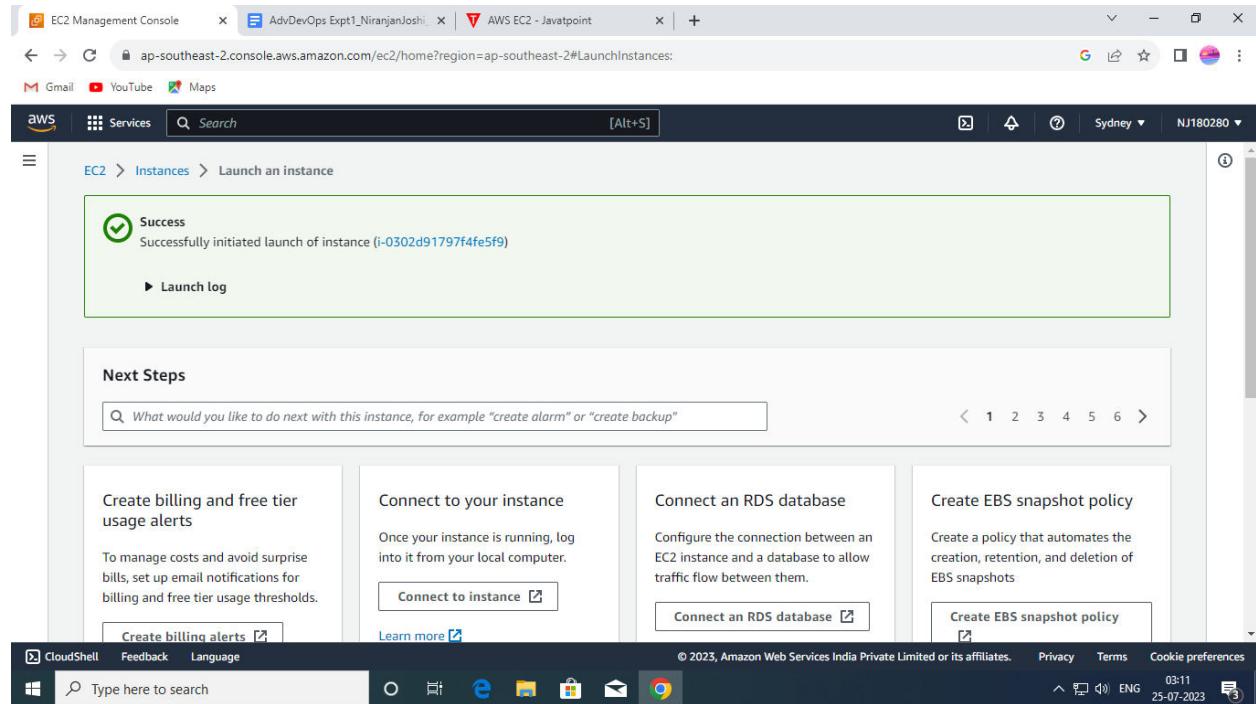
4) After clicking on launch instance new window will open and then give a name to instance and also select OS/AMI to work with

The screenshot shows the 'Launch instances' wizard, step 1: Application and OS Images (Amazon Machine Image). The instance name is set to 'Ubuntuinstance1'. The 'Quick Start' section shows various AMI options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar is available to search for full catalog. The 'Summary' section on the right shows the following configuration: Number of instances (1), Software Image (AMI) set to Canonical, Ubuntu, 22.04 LTS, Virtual server type (instance type) set to t2.micro, Firewall (security group) set to New security group, and Storage (volumes) set to 1 volume(s) - 8 GiB. The 'Launch instance' button is prominently displayed at the bottom right.

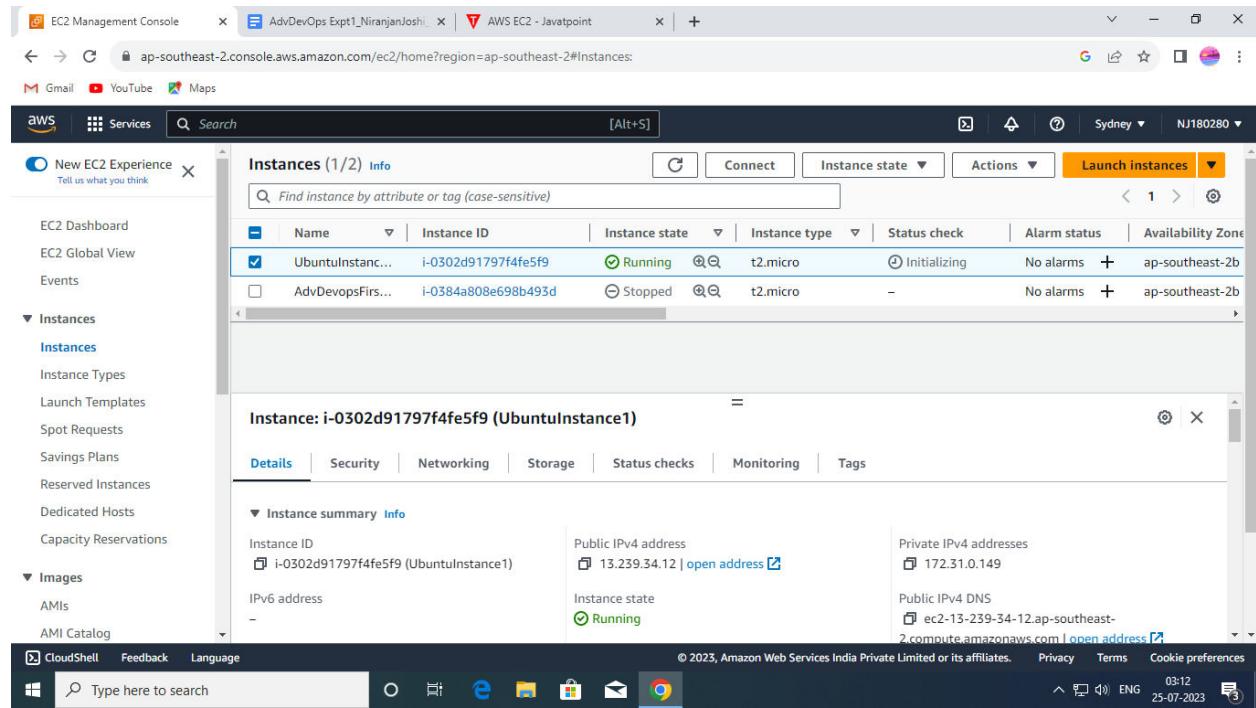
5) Do the necessary configurations and then click on Launch instance button and after clicking it will ask for key value pair , if key is not present then select proceed without key pair button



6) Now click on launch instance and then instance will be successfully created



7) Then go to EC2 dashboard and click on instances , then below page appears



EC2 Management Console

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
UbuntuInstance1	i-0302d91797f4fe5f9	Running	t2.micro	Initializing	No alarms	ap-southeast-2b
AdvDevOpsFirst	i-0384a808e698b493d	Stopped	t2.micro	-	No alarms	ap-southeast-2b

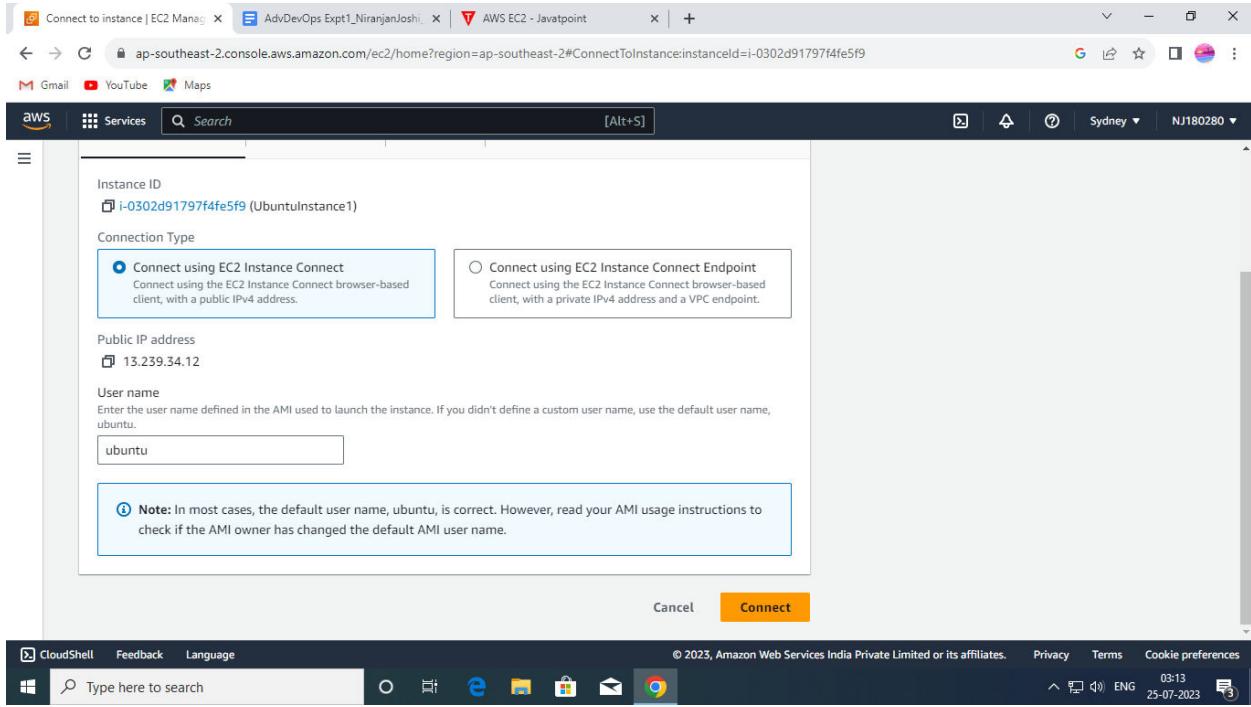
Instance: i-0302d91797f4fe5f9 (UbuntuInstance1)

Details Security Networking Storage Status checks Monitoring Tags

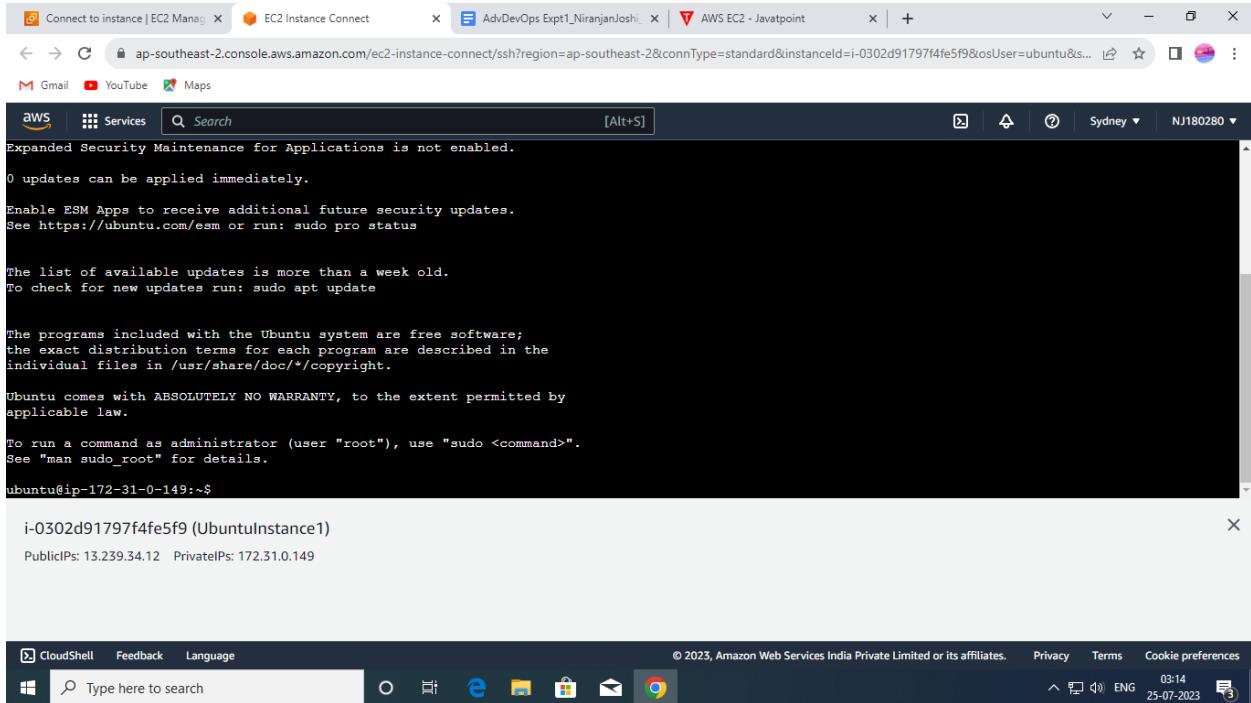
Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0302d91797f4fe5f9 (UbuntuInstance1)	13.239.34.12   open address	172.31.0.149
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-13-239-34-12.ap-southeast-2.compute.amazonaws.com   open address

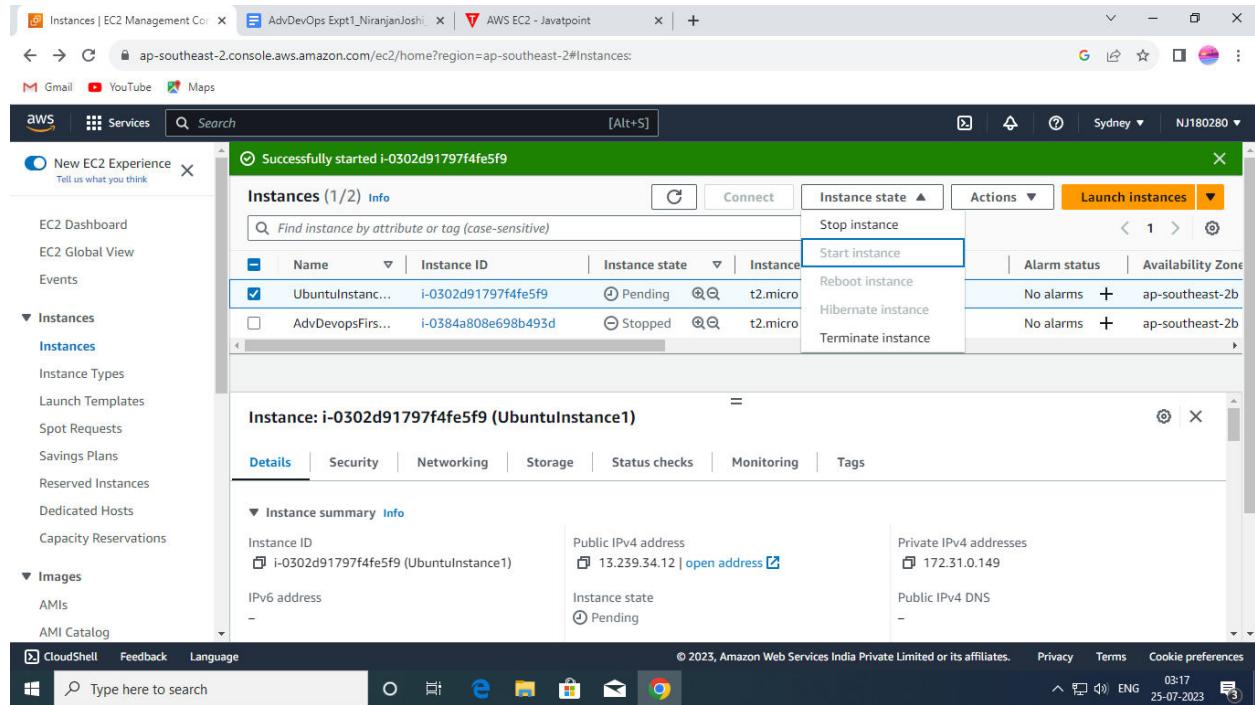
8) Then click on connect and then again on connect button in yellow color



9) After connecting instance will be created and shown on screen



10) After that go to instances page and select instance , then select instance state and click on stop instance



**Conclusion:-** Learned about creation of AWS account from scratch and creation of ubuntu environment instance in EC2 machine on AWS following all necessary steps .

**Name :- Niranjan Raejsh Joshi**

**RollNo :- 2105051**

**Batch :- T13**

**Date Of Performance :- 01/08/2023**

## **Assignment 2**

**Aim :-** Study and create AWS cloud9 IDE service

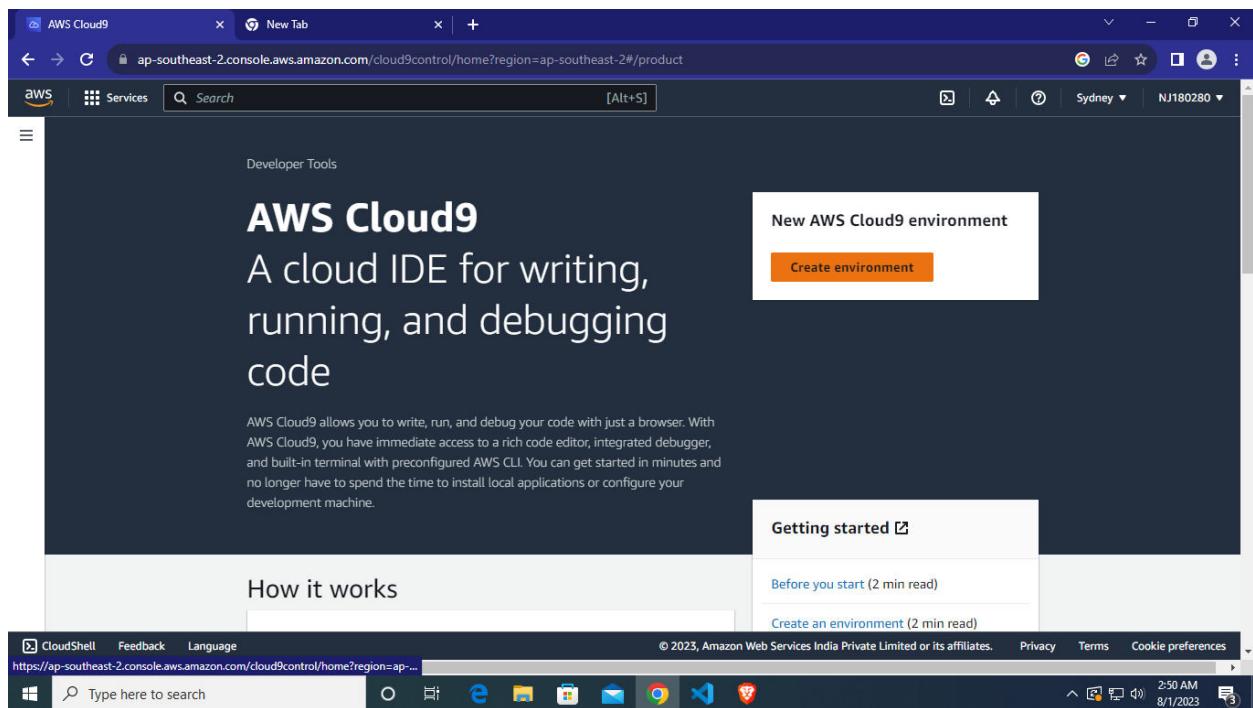
### **Theory :-**

AWS Cloud9 is an integrated development environment (IDE) offered by Amazon Web Services (AWS). It provides a cloud-based platform for software development, allowing developers to write, run, and debug code from within a web browser. Here are some key features and information about AWS Cloud9:

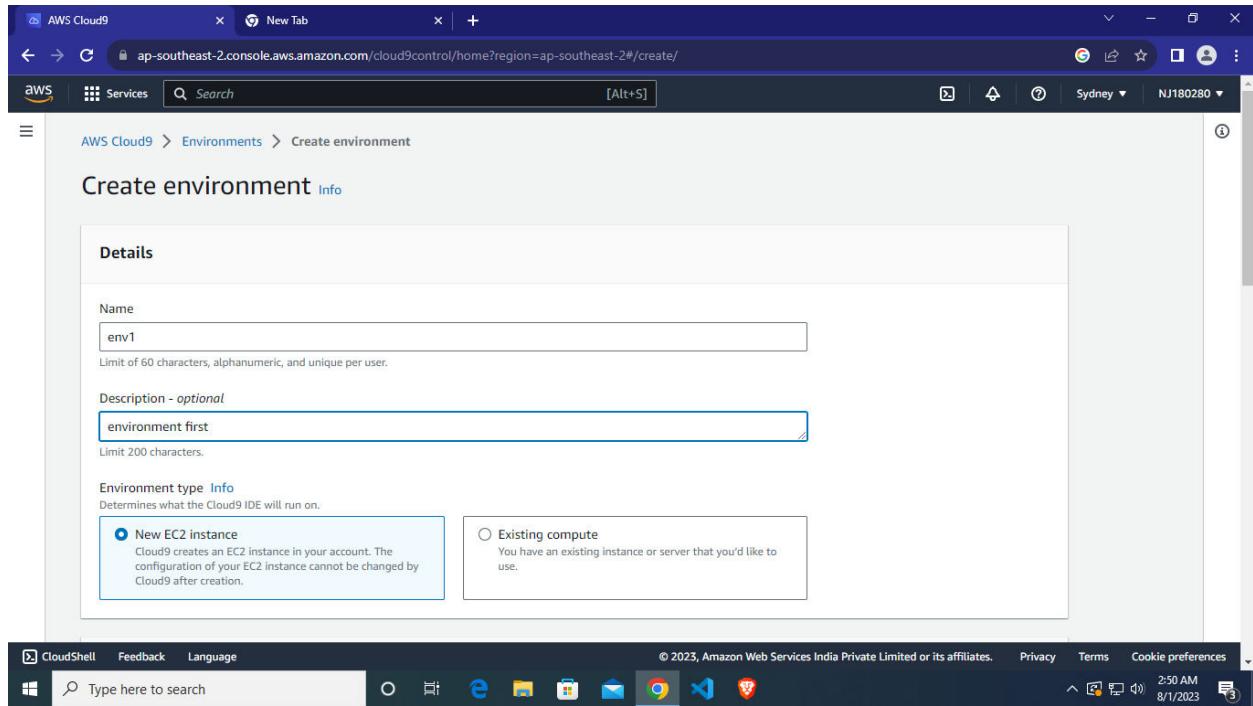
- 1) **Browser-Based IDE:** AWS Cloud9 allows developers to access their development environment through a web browser. This means you can code from anywhere with an internet connection without needing to set up your development environment on a local machine.
- 2) **Collaboration:** Cloud9 supports collaborative coding by allowing multiple users to work on the same code in real-time. This can be particularly useful for team projects and pair programming.
- 3) **Built-in Terminal:** Cloud9 includes a built-in terminal that lets you execute commands and run scripts directly from the IDE. This eliminates the need to switch between different tools for coding and terminal access.
- 4) **Preconfigured Development Environments:** AWS Cloud9 supports various programming languages and runtimes. It provides preconfigured environments for popular languages like JavaScript, Python, Java, and more. You can also create custom environments tailored to your needs.
- 5) **AWS Integration:** Cloud9 is tightly integrated with other AWS services, making it easy to deploy and test applications on AWS infrastructure. You can access resources like Amazon EC2 instances, Lambda functions, and databases directly from within the IDE.
- 6) **Code Debugging:** Cloud9 offers debugging capabilities, allowing developers to set breakpoints, inspect variables, and step through code to identify and fix issues.
- 7) **Version Control Integration:** Cloud9 can integrate with version control systems like Git, making it easier to manage code repositories and collaborate with team members.

## ✓ Steps to create AWS cloud 9 IDE service :-

- 1) Log in to your AWS account by entering appropriate credentials
- 2) In search bar present on dashboard search for cloud9 service by entering “cloud9”



- 3) Then click on create environment option and enter environment name and environment description , select EC2 instance in environment type



AWS Cloud9 > Environments > Create environment

### Create environment Info

**Details**

Name  Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional*  Limit 200 characters.

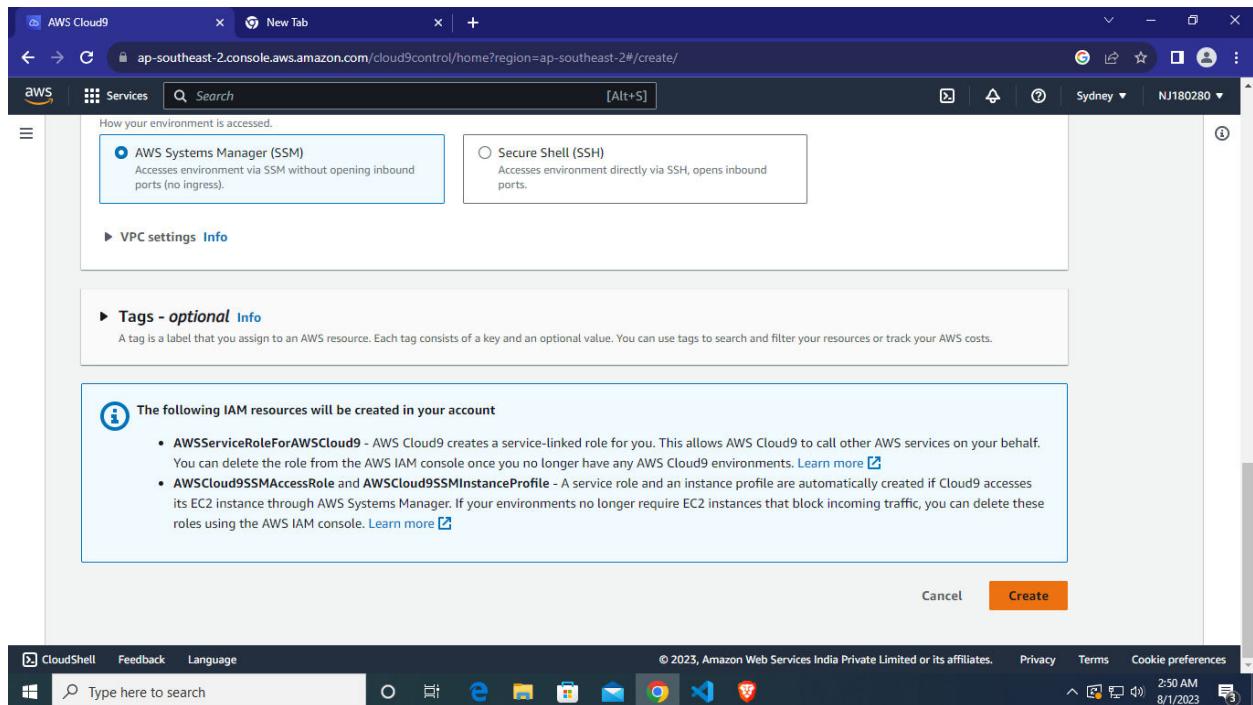
Environment type Info  
Determines what the Cloud9 IDE will run on.

**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.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 2:50 AM 8/1/2023

4) Then at bottom right corner create button is present , click it



How your environment is accessed.

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

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

► VPC settings Info

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

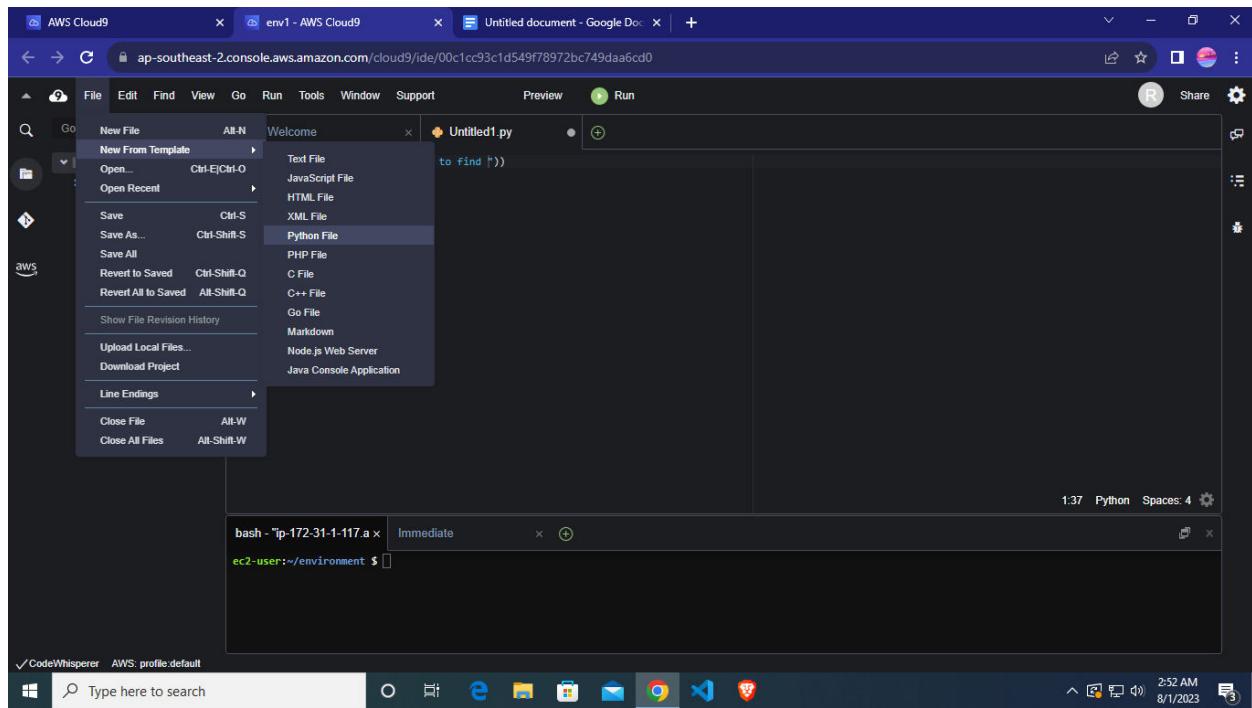
**Info** The following IAM resources will be created in your account

- **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](#)

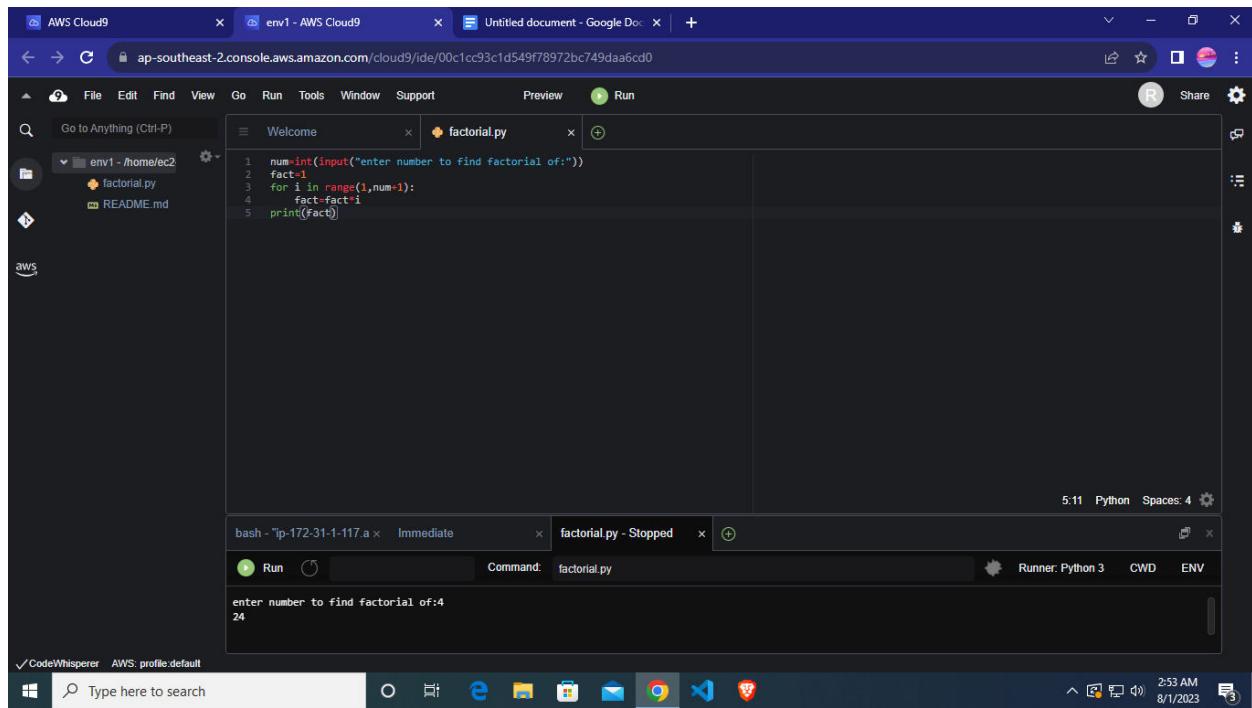
Cancel **Create**

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 2:50 AM 8/1/2023

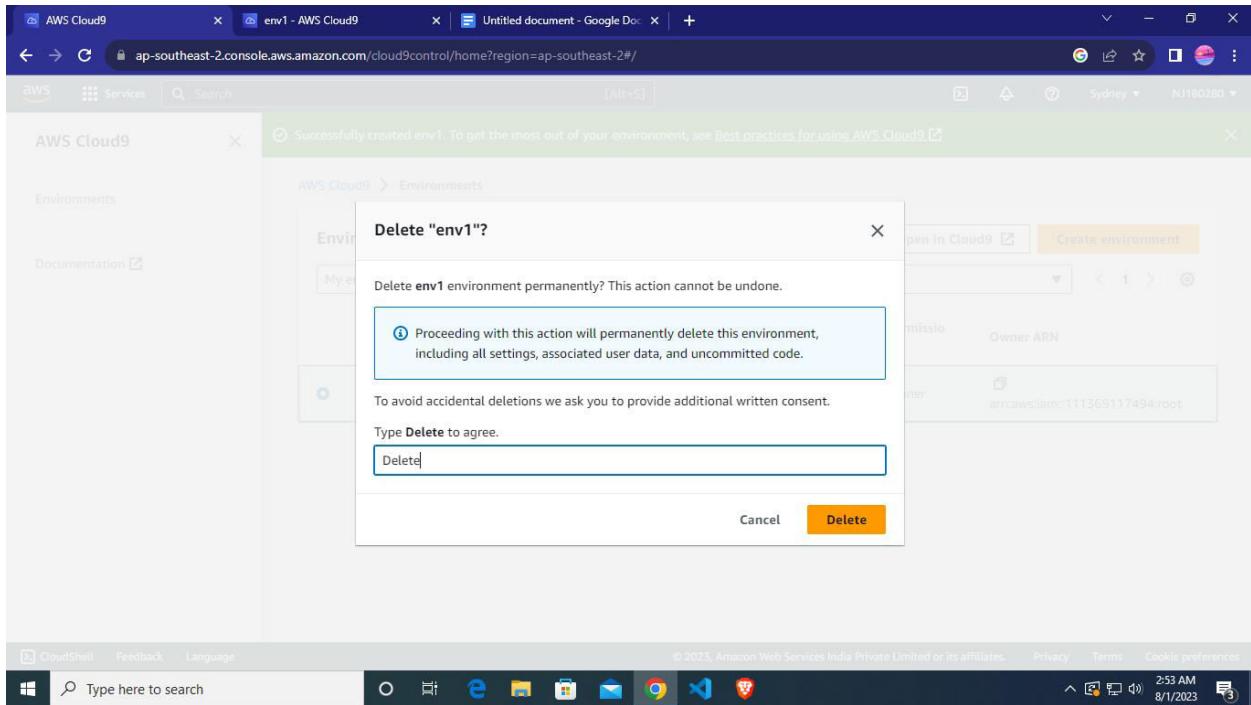
5) Then cloud9 instance will be in running state , then go to file and click on python file to cerate a new python file



6) Then type code for any program , in this case code is to find factorial of a number and python is used as programming language and after completing writing of code , click on run button above to execute the written code and output is visible at bottom in console window



7) After using the environment, we have to delete the environment, come to dashboard of cloud9 and see for created environment, click on delete option and confirm the deletion by typing delete in textbox given, this will delete cloud9 instance which was running



**Conclusion :-** learnt about AWS cloud9 service, its use-case and how it simplifies task, configured a AWS cloud9 IDE, ran a python program to calculate factorial of a number in cloud9 IDE and then deleted the environment after use

**Name :- Niranjan Rajesh Joshi**  
**RollNo :- 2105051**  
**Batch :- T13**  
**Date Of Performance :- 01/08/2023**

## **Experiment 3**

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

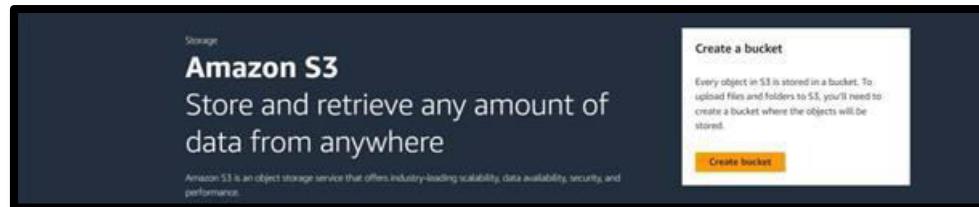
### **THEORY-**

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.

#### ***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

General configuration

Bucket name: `awsbuckets3`  
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region: Europe (Stockholm) `eu-north-1`

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

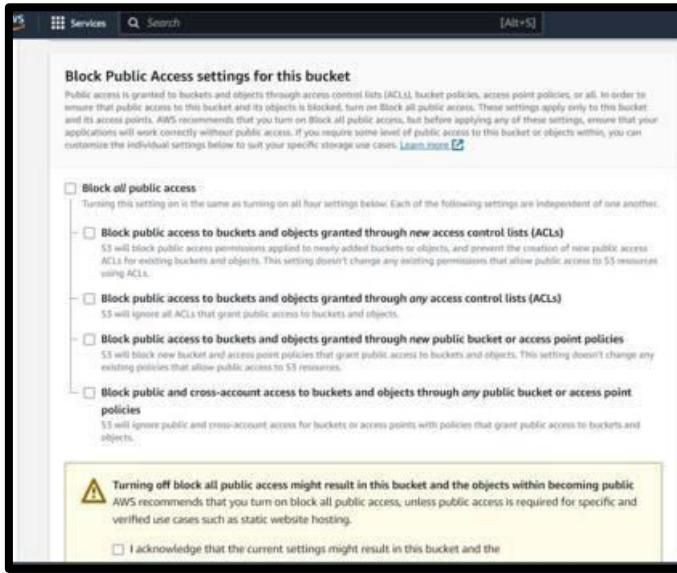
**Object Ownership** [Info](#)  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**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.

## 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.

Upload succeeded

View details below

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

**Summary**

Destination	Succeeded	Failed
s3://hash-my-aws-bucket	2 files, 3.5 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** Configuration

**Files and folders (2 Total, 3.5 KB)**

Name	Folder	Type	Size	Status	Error
aws.png	-	image/png	3.1 kB	Successed	-
main.html	-	text/html	359.0 B	Successed	-

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

- Block public access
- Object Ownership
- Make public Object

Amazon S3 > Buckets > hash-my-aws-bucket

hash-my-aws-bucket [Info](#)

Objects Properties Permissions Metrics Management Access Points

**Objects (2)**

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](#)

<input checked="" type="checkbox"/>	Name	Type	Last modified	Actions	Create folder	Upload
<input checked="" type="checkbox"/>	aws.png	png	October 3, 2018 (UTC+05:30)	<a href="#">Copy</a>	<a href="#">Move</a>	<a href="#">Edit actions</a>
<input checked="" type="checkbox"/>	main.html	html	October 3, 2018 (UTC+05:30)	<a href="#">Initiate restore</a>	<a href="#">Calculate total size</a>	<a href="#">Storage class</a>

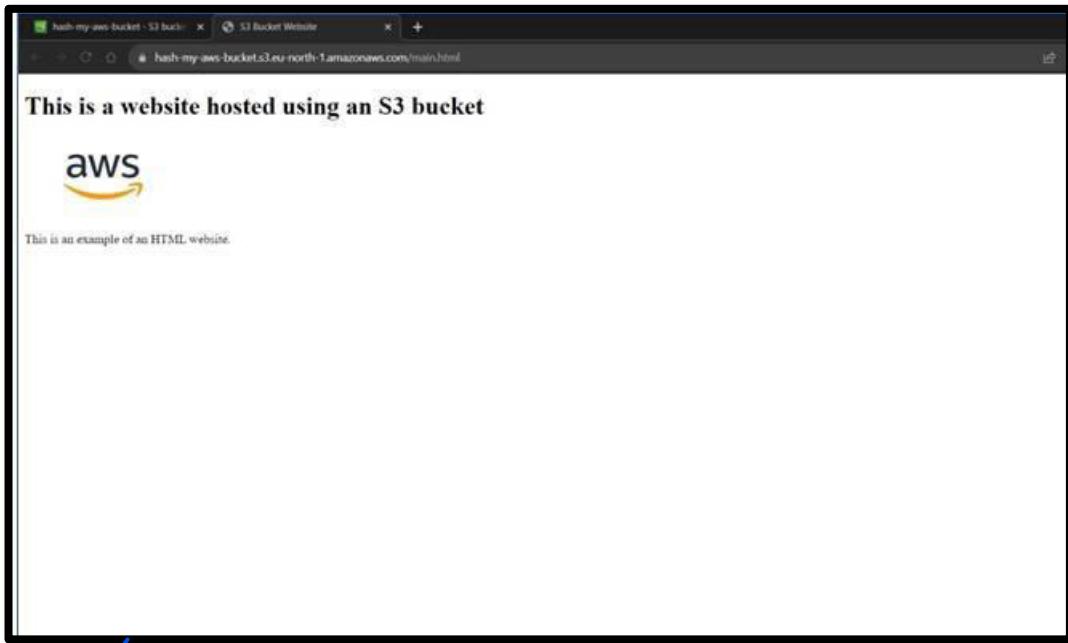
**Step 6: Copy your Object URL**

Now, click on your HTML File Object Name.

Copy the Object URL.

## **Step 7: 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 is a powerful and scalable cloud storage solution for hosting static web applications. It offers low latency, high reliability, and cost-effectiveness, making it a vital tool for modern web development and deployment.

XRCM

**Name :-Niranjan Rajesh Joshi**

**Roll No:- 2105051**

**Batch:- T13**

**Date Of Performance :- 08/08/2023**

## Experiment 4

**Aim:-**To study AWS code pipeline and deploy web application using code pipeline

### **Theory:-**

AWS CodePipeline is a **fully managed continuous delivery service** that helps automate the release pipelines for fast and reliable application and infrastructure updates. **It allows you to model, visualize, and automate the steps required to release your software.** Here are some key points about AWS CodePipeline:

**Automation of Software Release Process:** AWS CodePipeline automates the build, test, and deployment phases of your release process every time there is a code change, based on the release model you define.

**Integration with Different Services:** It integrates with a variety of third-party services and AWS services such as AWS CodeBuild, AWS CodeDeploy, and AWS CloudFormation, enabling you to have a fully automated release process for your applications.

**Customizable Pipeline:** CodePipeline allows you to build custom release workflows with multiple stages and actions. Each stage can have one or more actions, and you can define the actions to be performed at each stage, such as source code versioning, building, testing, and deployment.

**Visual Workflow:** It provides a visual representation of your release process, allowing you to see the stages and actions in the pipeline and monitor the progress of each release.

**Integration with Third-Party Tools:** It supports integration with a wide range of third-party tools and services through its extensible architecture, enabling you to incorporate your favorite tools into the release process.

**Flexibility and Control:** CodePipeline provides flexibility and control over the release process, allowing you to define custom rules for the execution of each action and the transition between stages.

**Security:** It integrates with AWS Identity and Access Management (IAM) to control access to your pipelines, ensuring that only authorized users have the necessary permissions to view or modify the pipelines.

**Monitoring and Logging:** AWS CodePipeline provides monitoring and logging capabilities, allowing you to track the execution of each action and stage in the pipeline and quickly identify any issues or failures.

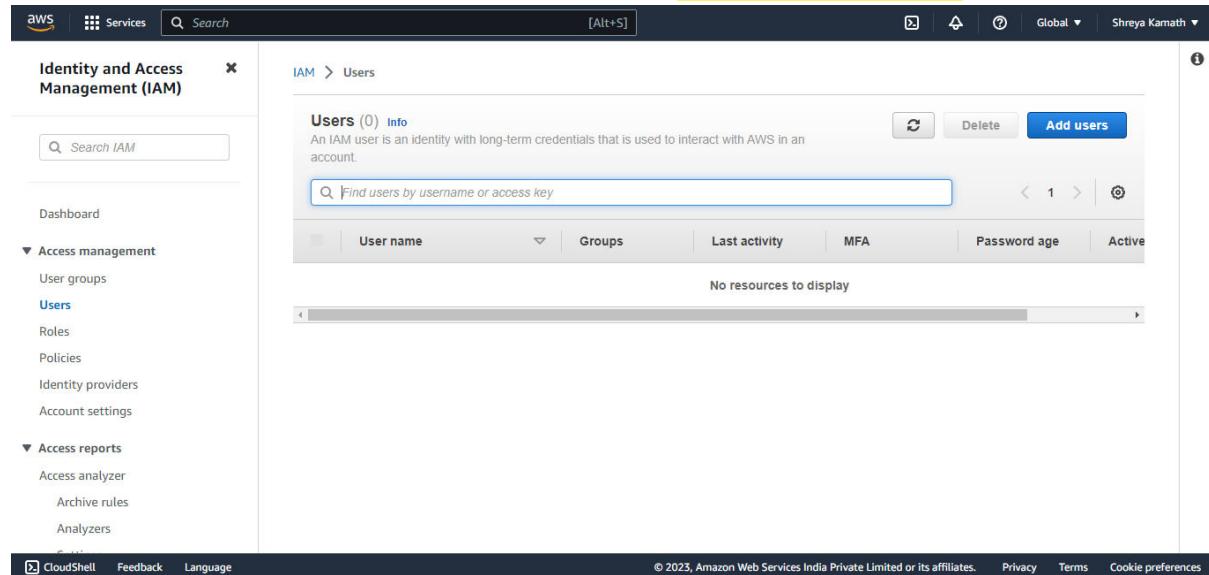
**Scalability and Availability:** As a fully managed service, AWS CodePipeline offers scalability and high availability, ensuring that your release pipelines can handle any workload and are always accessible.

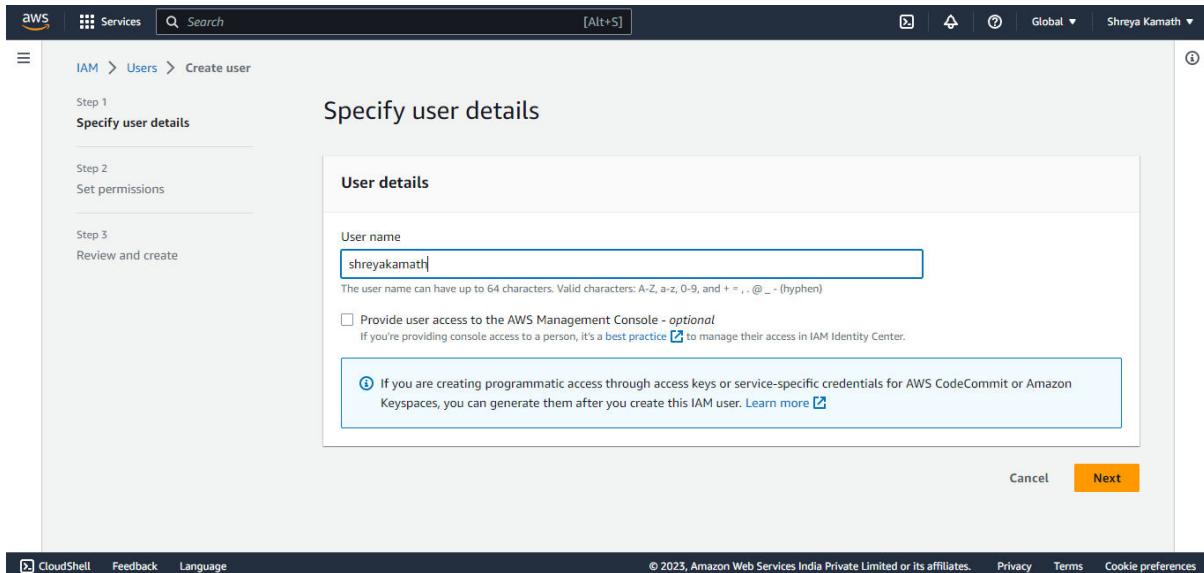
**Cost-Effective:** With a pay-as-you-go pricing model, AWS CodePipeline helps you optimize costs by charging only for the resources you use.

## Steps :-

1) Login to AWS account and in search bar search **IAM** and click on it

2) Dashboard of IAM user open and then **create a new user**





Specify user details

User details

User name: shreyakamath

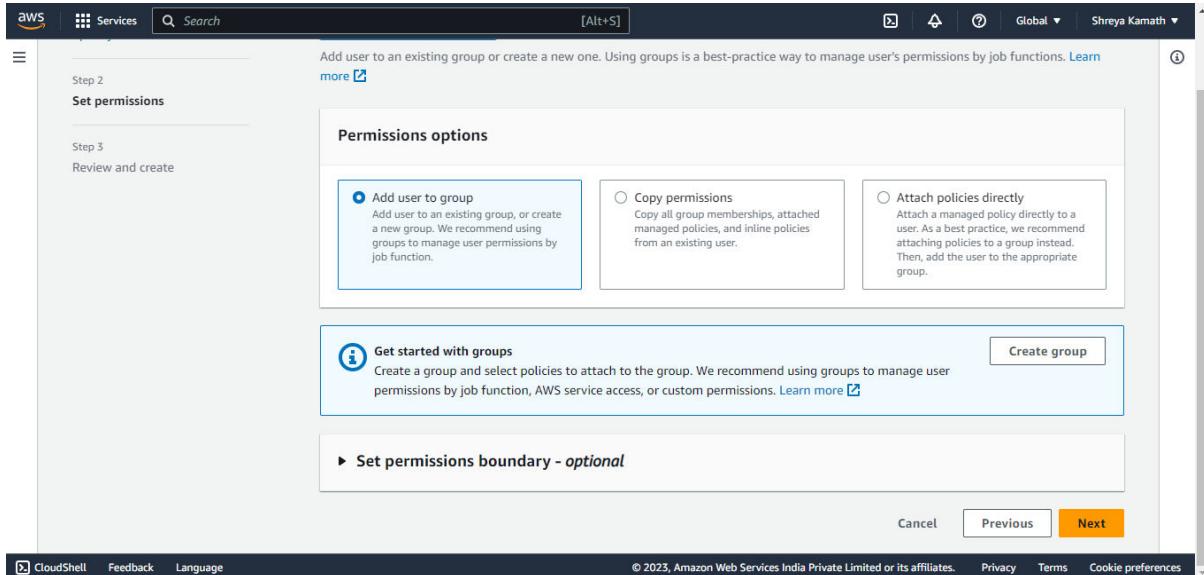
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . \_ - (hyphen)

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

[Learn more](#)

Cancel Next

## 2) Set the permissions for the new user



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

Permissions options

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

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

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

[Create group](#)

[Get started with groups](#)

[Learn more](#)

Set permissions boundary - *optional*

Cancel Previous Next

## 3) Create user and user name , usergroup will be visible on dashboard

The screenshot shows the AWS IAM 'Create user group' wizard. The current step is 'Permissions policies (871)'. The user has selected the 'AdministratorAccess' policy. The 'Create user group' button is highlighted in orange. A success message at the bottom indicates 't13shreya user group created.' The 'User groups (1)' table shows the newly created group 't13shreya'.

**Create user group**

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

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

t13shreya

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

**Permissions policies (871)**

Filter by Type

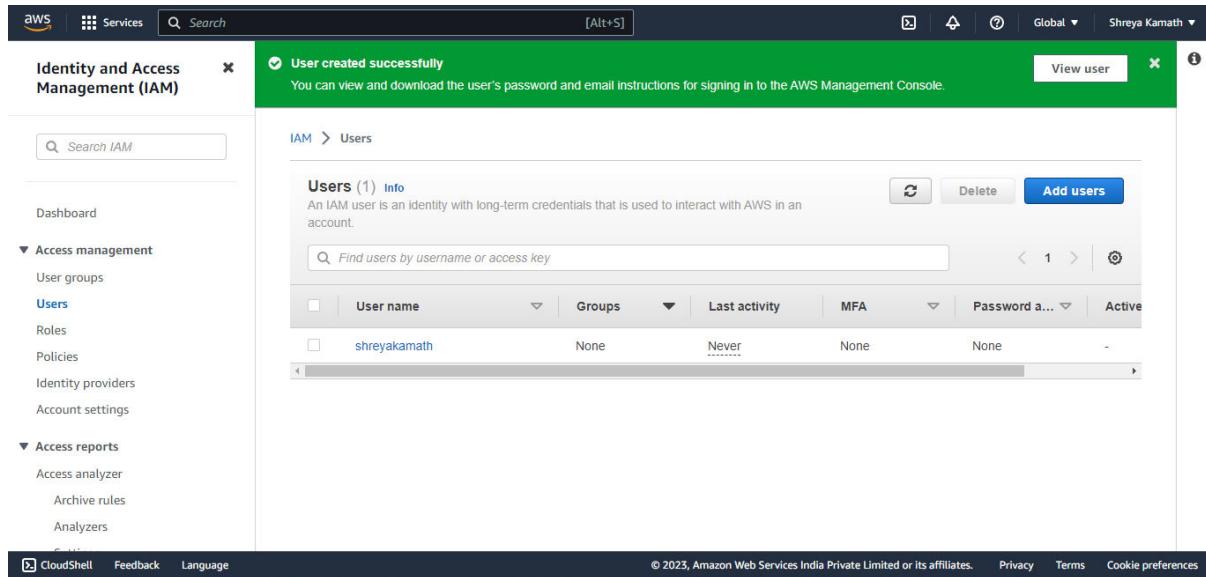
Policy name	Type	Use...	Description
AdministratorAccess	AWS managed	None	Provides full access to AWS services
AdministratorAccess	AWS managed	None	Grants account administrative permissions
AdministratorAccess	AWS managed	None	Grants account administrative permissions
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to Alexa

**t13shreya user group created.**

**User groups (1)**

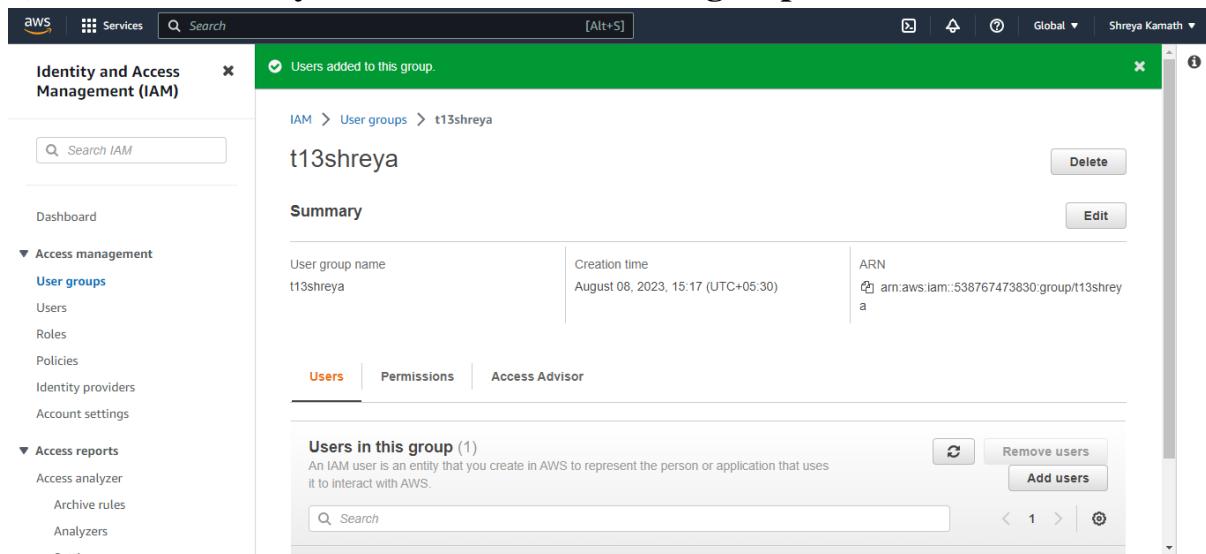
Group name	Users	Attached policies	Created
t13shreya	0	-	2023-08-08 (Now)

**Set permissions boundary - optional**



The screenshot shows the AWS IAM Users page. A green header bar at the top indicates that a user has been created successfully. The main content area shows a table of users with one entry: 'shreyakamath'. The table includes columns for User name, Groups, Last activity, MFA, Password last used, and Active status. The 'shreyakamath' user is listed with 'None' in all columns except 'Active' which is marked with a checkmark. The page also includes a search bar, navigation buttons, and standard AWS navigation elements like CloudShell, Feedback, Language, and a footer with copyright information.

#### 4) See the summary of created user and usergroup



The screenshot shows the AWS IAM User Groups page. A green header bar at the top indicates that a user group has been added. The main content area shows a table of user groups with one entry: 't13shreya'. The table includes columns for User group name, Creation time, and ARN. The 't13shreya' user group is listed with 't13shreya' in the User group name column, 'August 08, 2023, 15:17 (UTC+05:30)' in the Creation time column, and 'arn:aws:iam::538767473830:group/t13shreya' in the ARN column. The page also includes a search bar, navigation buttons, and standard AWS navigation elements like CloudShell, Feedback, Language, and a footer with copyright information.

The image shows two screenshots of the AWS IAM (Identity and Access Management) service. The top screenshot displays the 'Users' list, showing a single user named 'shreyakamath'. The bottom screenshot shows the detailed 'Summary' page for the user 'shreyakamath', including ARN, console access status, and access key details. Both screenshots include the AWS navigation bar and the IAM service header.

**Users (1) Info**  
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password a...	Active
shreyakamath	t13shreya	Never	None	None	-

**shreyakamath Info**

Summary		
ARN arn:aws:iam::53876747530:user/shreyakamath	Console access Disabled	Access key 1 Not enabled
Created August 08, 2023, 15:17 (UTC+05:30)	Last console sign-in -	Access key 2 Not enabled

**Permissions policies (0)**  
Permissions are defined by policies attached to the user directly or through groups.

**5) Go to security credentials tab and see for console password and other details , take screenshot of these details**

The screenshot shows the AWS IAM console for a user named 'Shreya Kamath'. The 'Security credentials' tab is selected. It displays the ARN of the user, console access status, and access keys. Below this, the 'Console sign-in' section shows a sign-in link and password status. The 'Multi-factor authentication (MFA)' section indicates 0 MFA devices assigned. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Language.

A modal window titled 'Console password' is displayed, showing a success message: 'You have successfully enabled the user's new password. This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.' Below the message, it shows the 'Console sign-in URL' as <https://538767473830.signin.aws.amazon.com/console>, the 'User name' as 'shreyakamath', and the 'Console password' as a masked string. At the bottom are 'Download .csv file' and 'Close' buttons.

6) Go to dashboard and search codecommit in search bar

The image shows two screenshots of the AWS console. The top screenshot is the 'Console Home' page, featuring a 'Recently visited' section with 'Cloud9' and 'EC2' links, and a 'Welcome to AWS' section with links for 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'. The bottom screenshot shows the 'User groups' creation step in the AWS IAM service. It displays 'Permissions options' with the 'Add user to group' radio button selected. Below this, a table lists a single user group named 'shreyat13' with 0 users and the policy 'AdministratorAccess' attached, created on 2023-08-08 (Now). The 'Create group' button is highlighted with a blue border.

7) Go to repositories option and create a new repository , give a name to it

The image shows two screenshots of the AWS CodeCommit interface. The top screenshot displays the 'Repositories' list, showing a single entry: 'No results' with the subtext 'There are no results to display.' The bottom screenshot shows the 'Repository settings' page for a repository named 'adv devops'. The settings include a description field, a tag named 'repo1' with value 'repo1', and an optional checkbox for 'Enable Amazon CodeGuru Reviewer for Java and Python'. The repository settings page also includes a sidebar with various AWS services and links.

Developer Tools > CodeCommit > Repositories

Repositories

No results

There are no results to display.

Repository settings

Repository name: adv devops

Description - optional

Tags

Key: Name, Value: repo1

Enable Amazon CodeGuru Reviewer for Java and Python - optional

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

## 8) Go to code option in left sidebar and select https there

**Repository settings**

Repository name: adv devops  
100 characters maximum. Other limits apply.

Description - optional:

Tags:

Key: Name	Value - optional: repo1	<input type="button" value="Remove tag"/>
<input type="button" value="Add tag"/>		

Enable Amazon CodeGuru Reviewer for Java and Python - optional  
Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.  
A service-linked role will be created in IAM on your behalf if it does not exist.

**Success**  
Repository successfully created

Developer Tools > CodeCommit > Repositories > advdevops

**Connection steps**

**Step 1: Prerequisites**  
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. [Learn how to add team members to an AWS CodeStar Project](#).

**Step 2: Git credentials**  
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

**Step 1: Prerequisites**  
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. [Learn how to add team members to an AWS CodeStar Project](#).

**Step 2: Git credentials**  
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

**Step 3: Clone the repository**  
Clone your repository to your local computer and start working on code. Run the following command:

```
git clone https://git-codecommit.eu-north-1.amazonaws.com/v1/repos/advdevops
```

**Additional details**  
You can find more detailed instructions in the documentation. [View documentation](#)

## 9) Create a new application and give name to it , create a deployment group for deployment of application created

The image shows two sequential screenshots from the AWS CodeDeploy console.

**Top Screenshot (Create application):**

- Application configuration:**
  - Application name:** shreyaT13
  - Compute platform:** EC2/On-premises
  - Tags:** A tag named "webapp1" is added.
- Buttons:** Cancel, Create application

**Bottom Screenshot (Create deployment group):**

- Deployment group name:** webappdeploygroup1
- Service role:** A service role named "service1" is selected.
- Deployment type:** In-place (selected)

The screenshot shows the AWS CodeDeploy console. On the left, a sidebar menu is open with the following items: Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Getting started, Deployments, Applications (Application, Settings, Deployment configurations, On-premises instances), Pipeline (CodePipeline), and Settings. The Deploy (CodeDeploy) item is expanded. In the main content area, a green banner at the top says "Success" and "Deployment group created". Below it, the path is "Developer Tools > CodeDeploy > Applications > shreyaT13 > webappdeploygroup1". The title "webappdeploygroup1" is displayed. Below the title are "Edit", "Delete", and "Create deployment" buttons. A "Deployment group details" section shows the following configuration: Deployment group name (webappdeploygroup1), Application name (shreyaT13), Compute platform (EC2/On-premises); Deployment type (In-place), Service role ARN (arn:aws:iam::538767473830:role/deployrole), Deployment configuration (CodeDeployDefault.AllAtOnce); Rollback enabled (False), Agent update scheduler (Learn to schedule update in AWS Systems Manager), and Environment configuration: Amazon EC2 instances. At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

## 10) Select the source provider , artifact

The screenshot shows the AWS CodePipeline configuration interface. On the left, a sidebar shows steps: Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main area is titled "Source provider" and says "This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details." A dropdown menu is open, showing "AWS CodeCommit". Below it, "Repository name" is set to "advdevops", "Branch name" is set to "master", and "Change detection options" are set to "Amazon CloudWatch Events (recommended)". "Output artifact format" is set to "CodePipeline default". At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

## 11) Review the application and then click on deploy , after deploying success message is displayed on screen

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar menu for 'CodePipeline' lists various stages and actions: Source (CodeCommit, CodeArtifact), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), and Pipeline (CodePipeline). The 'Pipeline' section is expanded, showing 'Getting started', 'Pipelines', and 'Settings'. Below the sidebar are links for 'Go to resource' and 'Feedback'.

The main content area is titled 'Review' and shows the 'Pipeline settings' for the new pipeline 'webapppipeline1'. The pipeline name is 'webapppipeline1', the artifact location is 'A new Amazon S3 bucket will be created as the default artifact store for your pipeline', and the service role name is 'AWSCodePipelineServiceRole-eu-north-1-webapppipeline1'.

Below the pipeline settings, the 'Step 2: Add source stage' section is visible, showing the 'Source action provider' for the pipeline.

At the top of the main content area, there is a success message: 'Success' and 'Congratulations! The pipeline webapppipeline1 has been created.' with a 'Create a notification rule for this pipeline' button.

The pipeline summary for 'webapppipeline1' shows the 'Source' stage (AWS CodeCommit) is 'In progress' (Pipeline execution ID: 2a0b4a9a-b288-4b51-a87b-14ba15c1cf2) and the 'Deploy' stage (CodeDeploy) has 'Didn't Run'.

## Conclusion:-

Learnt about creation and deployment of web application using AWS codepipeline , hence learnt about basic components of codepipeline in AWS such as codebuild , codecommit , codedeploy and built , committed and deployed a web application using codepipeline

Name :-Niranjan Rajesh Joshi  
Roll No:- 2105051  
Batch:- T13  
Date Of Performance :- 13/10/2023

X from

~~Experiment 5~~

#### Aim:-

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

#### Theory:-

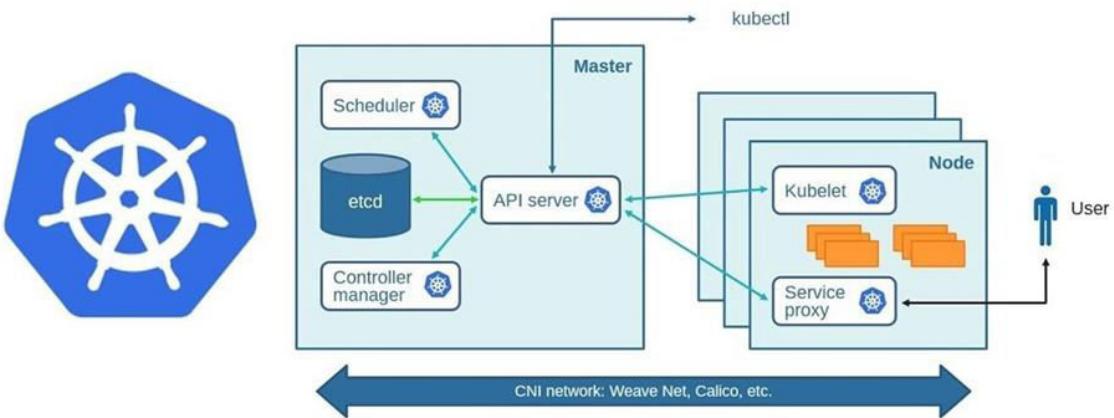
Kubernetes is an open-source container management tool that automates container deployment, scaling & load balancing.

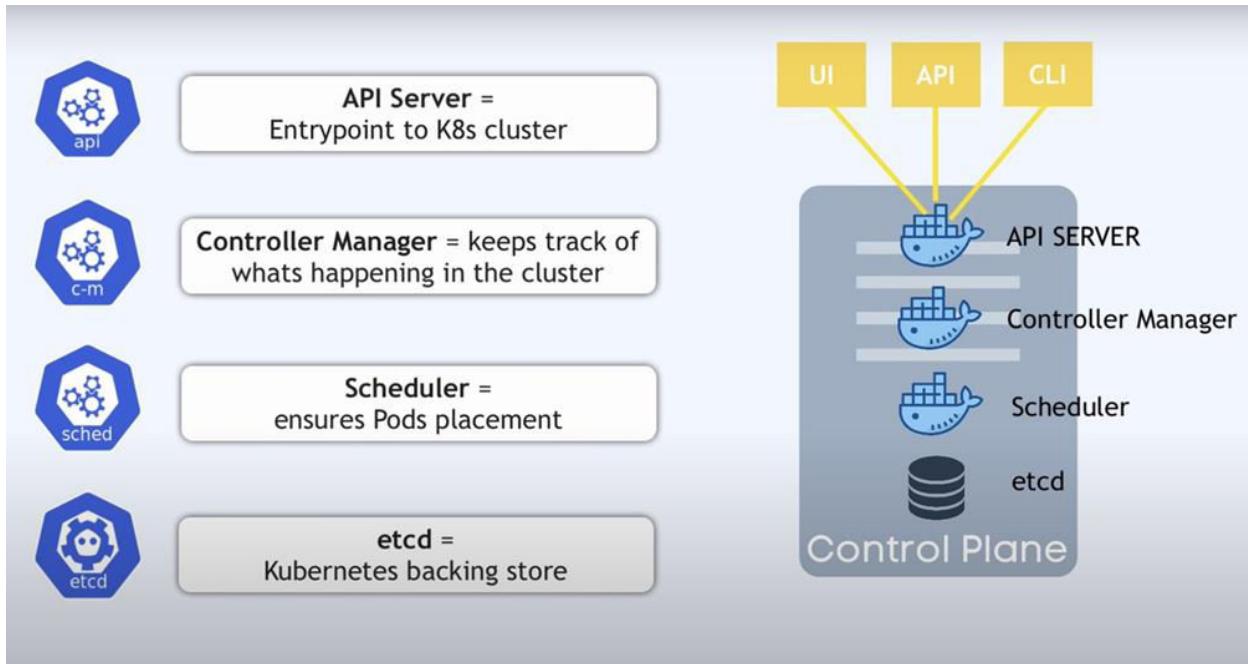
It schedules, runs, and manages isolated containers that are running on virtual/physical/cloud machines.

All top cloud providers support Kubernetes. One popular name for Kubernetes is K8s.

## ARCHITECTURE

# Kubernetes





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

In a Kubernetes cluster, several services run on each node to enable the proper functioning of the cluster. These services play crucial roles in managing, scheduling, and maintaining the various components of the cluster. Some of the essential Kubernetes services running on nodes include:

**kubelet:** It is the primary node agent that runs on each node in the cluster and ensures that the containers are running in a Pod. Kubelet communicates with the Kubernetes control plane and manages the Pods and their containers according to the instructions received.

**kube-proxy:** Kube-proxy is responsible for network proxying on the node and performs the necessary network address translation and load balancing to route the traffic to the appropriate Pod. It maintains network rules on the node and enables communication between different Pods and services in the cluster.

**container runtime:** Kubernetes supports various container runtimes such as Docker, containerd, and CRI-O. The container runtime is responsible for pulling the container images from the container registry, creating the container from those images, and managing the container lifecycle, including starting, stopping, and deleting containers.

**kube-dns/coredns:** These are the DNS services in Kubernetes that provide service discovery and enable communication between different services within the cluster. They ensure that the Pods

can locate each other by their DNS names and facilitate communication between different components.

**kube-scheduler:** The kube-scheduler is responsible for making decisions about which nodes should run specific Pods based on resource requirements, hardware constraints, and other policies. It assigns the Pods to nodes that can accommodate them and meet the specified criteria.

**kube-controller-manager:** This component includes several controllers that handle different aspects of the cluster, such as node and endpoint controllers, which perform tasks like node monitoring and managing endpoints. It ensures that the desired state of the cluster is maintained and takes necessary actions to reconcile any discrepancies.

**etcd:** While not strictly a service running on the node, etcd is a consistent and highly available key-value store used by Kubernetes to store all of its data. It maintains the state of the cluster, including configuration data, metadata, and the current state of all objects in the system.

Each of these services plays a critical role in ensuring the proper functioning of a Kubernetes cluster, enabling efficient management and orchestration of containerized applications.

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

A Pod Disruption Budget (PDB) is a Kubernetes feature that allows user to control the number of Pods that are down simultaneously in a given deployment or replica set. It ensures high availability by specifying the minimum number of Pods that must be available at any given time during voluntary disruptions, such as maintenance or scaling events.

PDBs help to prevent situations where too many Pods are terminated simultaneously, leading to potential service disruption or downtime. They allow user to set constraints on the disruption budget, ensuring that a minimum number of Pods remain available even when nodes or underlying hardware experience failures or planned maintenance activities.

By defining a Pod Disruption Budget, user can maintain a desired level of availability and reliability for users applications, especially in scenarios where user need to perform updates, upgrades, or scaling operations without causing unnecessary downtime or service interruptions.

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

In Kubernetes, a load balancer serves the crucial role of distributing incoming network traffic across multiple backend services or Pods. Its primary functions include:

**Traffic distribution:** It evenly distributes incoming network traffic across multiple instances of an application or service running in the Kubernetes cluster. This ensures that no single Pod or node is overwhelmed by an excessive amount of traffic.

**High availability:** Load balancing helps ensure high availability and reliability of applications by preventing any single point of failure. If one Pod or node becomes unavailable, the load balancer redirects traffic to other available instances, thereby minimizing downtime and service disruptions.

**Scalability:** Load balancing facilitates horizontal scaling by dynamically adding or removing Pods to handle varying levels of traffic. This enables the system to adapt to changes in demand and maintain consistent performance even during peak usage periods.

**Optimized resource utilization:** By efficiently distributing traffic, a load balancer helps in optimizing resource utilization across the cluster. It ensures that resources are utilized effectively without overloading any specific node or Pod.

Overall, the load balancer plays a critical role in ensuring the reliable and efficient operation of applications in a Kubernetes cluster, enabling seamless traffic management, high availability, and scalability to meet varying demands.

## INSTALLATION:

## 1. Install Docker

## 2) Install minikube using following commands

```

Oct 14 22:27 • prasad@prasad-VirtualBox:~ prasad@prasad-VirtualBox:~

prasad@prasad-VirtualBox:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
prasad@prasad-VirtualBox:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
[sudo] password for prasad:
prasad@prasad-VirtualBox:~$ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration

  Exiting due to PROVIDER_DOCKER_NGRCP: "docker version --format <no value>--<no value>--<no value>" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: connect: permission denied
  Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
  Documentation: https://docs.docker.com/engine/install/linux-postinstall/

prasad@prasad-VirtualBox:~$ sudo usermod -aG docker $USER && newgrp docker
prasad@prasad-VirtualBox:~$ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration

  The requested memory allocation of 1971MB does not leave room for system overhead (total system memory: 1971MB). You may face stability issues.
  Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1971mb'

  Using Docker driver with root privileges
  Starting control plane node minikube in cluster minikube
  Pulling base image ...
  Downloading Kubernetes v1.27.4 preload ...
  > preloaded-images-k8s-v18-v1...: 393.21 MB / 393.21 MB 100.00% 2.85 MiB
  > gcr.io/k8s-minikube/kicbase...: 447.62 MiB / 447.62 MiB 100.00% 2.99 MiB
  Creating docker container (CPU=2, Memory=1971MB) ...

  Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
  Suggestion:

  Try one or more of the following to free up space on the device:
  1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
  2. Increase the storage allocated to Docker for Desktop by clicking on:
  Docker icon > Preferences > Resources > Disk Image Size
  3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
  Related issue: https://github.com/kubernetes/minikube/issues/9024

  This container is having trouble accessing https://registry.k8s.io
  To pull new external images, you may need to configure a proxy: https://minikube.stgs.k8s.io/docs/reference/networking/proxy/
  ■ Generating certificates and keys ...

Oct 14 22:27 • prasad@prasad-VirtualBox:~ prasad@prasad-VirtualBox:~

prasad@prasad-VirtualBox:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
prasad@prasad-VirtualBox:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
[sudo] password for prasad:
prasad@prasad-VirtualBox:~$ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration

  The requested memory allocation of 1971MB does not leave room for system overhead (total system memory: 1971MB). You may face stability issues.
  Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1971mb'

  Using Docker driver with root privileges
  Starting control plane node minikube in cluster minikube
  Pulling base image ...
  Downloading Kubernetes v1.27.4 preload ...
  > preloaded-images-k8s-v18-v1...: 393.21 MB / 393.21 MB 100.00% 2.85 MiB
  > gcr.io/k8s-minikube/kicbase...: 447.62 MiB / 447.62 MiB 100.00% 2.99 MiB
  Creating docker container (CPU=2, Memory=1971MB) ...

  Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
  Suggestion:

  Try one or more of the following to free up space on the device:
  1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
  2. Increase the storage allocated to Docker for Desktop by clicking on:
  Docker icon > Preferences > Resources > Disk Image Size
  3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
  Related issue: https://github.com/kubernetes/minikube/issues/9024

  This container is having trouble accessing https://registry.k8s.io
  To pull new external images, you may need to configure a proxy: https://minikube.stgs.k8s.io/docs/reference/networking/proxy/
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC ...
  ■ Configuring Bridge CNI (Container Network Interface) ...
  ■ Using Image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Verifying Kubernetes components...
  Enabled addons: default-storageclass, storage-provisioner
  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasad@prasad-VirtualBox:~$ kubectl get pods -A
prasad@prasad-VirtualBox:~$ 

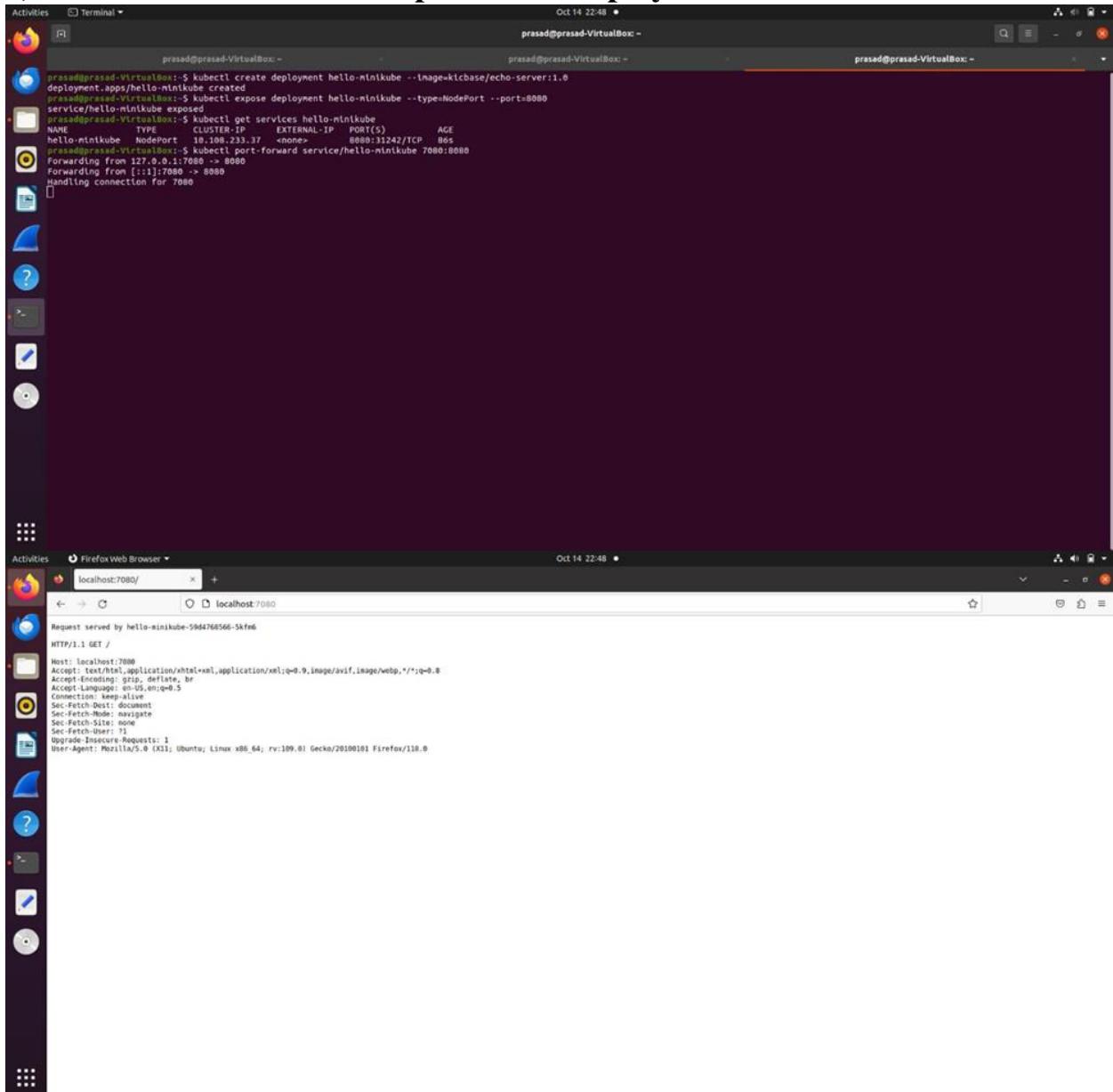
```

### 3) Install kubectl

Activities Terminal Oct 14 22:36 • prasad@prasad-VirtualBox: ~

```
prasad@prasad-VirtualBox: ~
Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
...
Downloaded Kubernetes v1.27.4 preloaded ...
> preloaded-images-k8s-v18-v1: 393.21 MiB / 393.21 MiB 100.00% 2.85 MiB
> gcr.io/k8s-minikube/kicbase...: 447.62 MiB / 447.62 MiB 100.00% 2.99 MiB
Creating docker container ((CPU=2, Memory=1971MiB) ...
Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
Suggestion:
Try one or more of the following to free up space on the device:
1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
2. Increase the storage allocated to Docker for Docker by clicking on:
Docker icon > Preferences > Resources > Disk Image Size
3. Run "minikube ssh -- docker system prune" if using the docker container runtime
Related issue: https://github.com/kubernetes/minikube/issues/9024
TLS container is having trouble accessing https://registry.k8s.io
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.27.4 on Docker 24.0.4 ...
■ Generating certificates and keys ...
■ Booting up control plane ...
■ Configuring RBAC rules ...
Configuring Bridge CNI (Container Networking Interface) ...
■ Building CNI configuration for minikube/storage-provisioner v5
Verifying Kubernetes components...
Enabled addons: default-storageclass, storage-provisioner
kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasad@prasad-VirtualBox: ~> <ctrl-d>
prasad@prasad-VirtualBox: ~$ sudo snap install kubectl --classic
[sudo] password for prasad:
kubectl 1.28.2 from Canonical** installed
prasad@prasad-VirtualBox: ~$ kubectl get po -A
prasad@prasad-VirtualBox: ~$ kubectl get po -A
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-system coredns-5d78c969d-6wdgp 1/1 Running 0 16m
kube-system kube-dns-minikube 1/1 Running 0 16m
kube-system kube-apiserver-minikube 1/1 Running 0 16m
kube-system kube-controller-manager-minikube 1/1 Running 0 17m
kube-system kube-proxy-snjnt 1/1 Running 0 16m
kube-system kube-scheduler-minikube 1/1 Running 0 16m
kube-system storage-provisioner 1/1 Running 1 (16m ago) 16m
*** prasad@prasad-VirtualBox: ~
```

## 4) Create Sample Deployment



```
prasad@prasad-VirtualBox:~$ kubectl create deployment hello-minikube --image=kicbase/echo-server:1.6
deployment.apps/hello-minikube created
prasad@prasad-VirtualBox:~$ kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
prasad@prasad-VirtualBox:~$ kubectl get services hello-minikube
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort    10.106.233.37  <none>        8080:31242/TCP   86s
prasad@prasad-VirtualBox:~$ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::]:7080 -> 8080
Handling connection for 7080
```

Activities Terminal Oct 14 22:48 prasad@prasad-VirtualBox:~

Firefox Web Browser Oct 14 22:48 localhost:7080

Request served by hello-minikube-59d4768566-5kfm6

HTTP/1.1.1 GET /

Host: localhost:7080

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.5

Cache-Control: max-age=0

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Sec-Fetch-User: ?1

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/110.0

## Conclusion:-

Here we studied Kubernetes cluster architecture in detail. Also we installed Kubernetes in ubuntu machine and created a sample deployment.

**Name :-Niranjan Rajesh Joshi**

**Roll No:- 2105051**

**Batch:- T13**

**Date Of Performance :- 22/08/2023**



## **Experiment 6**

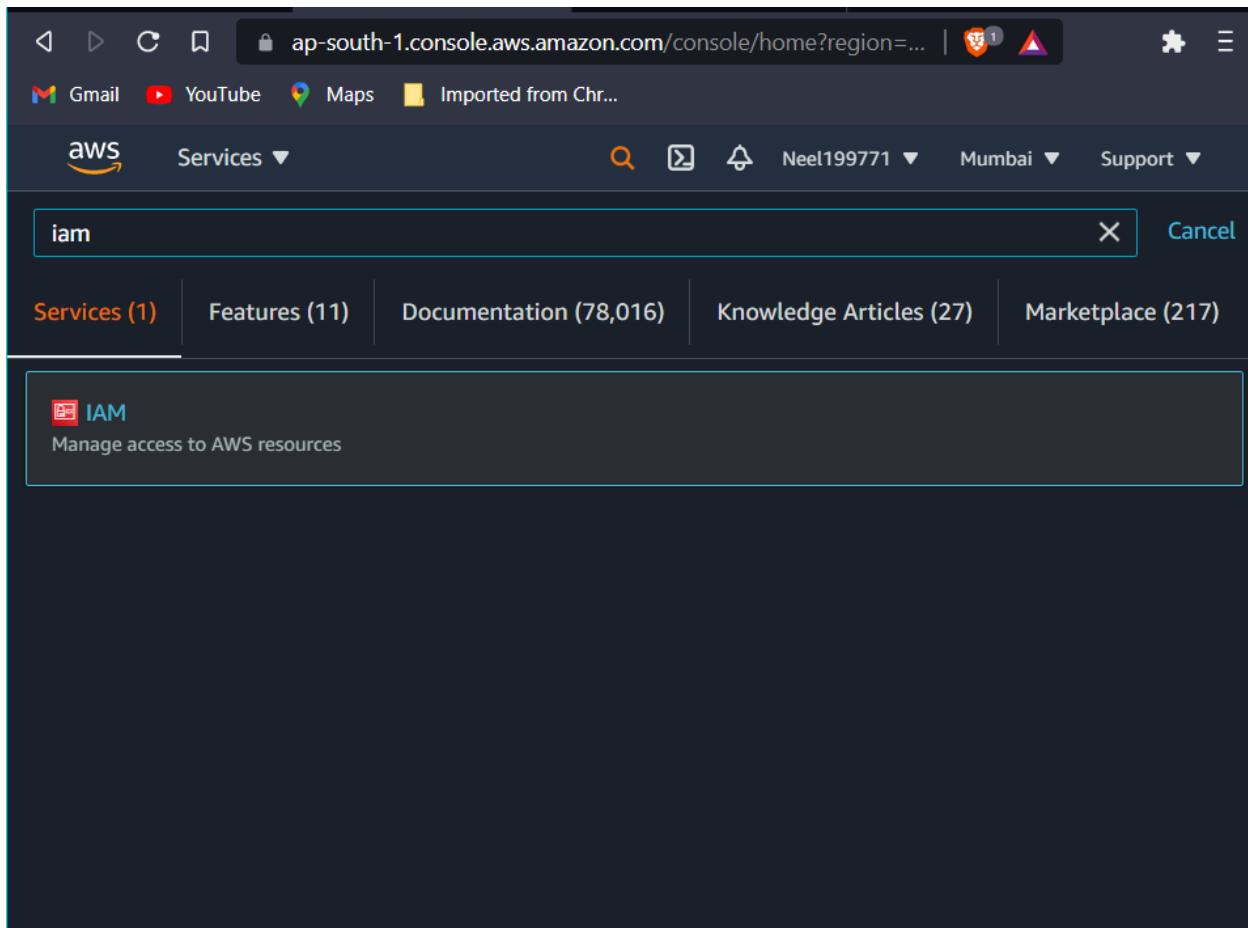
### **Aim :-**

To understand terraform lifecycle and to build, change , and destroy AWS infrastructure using Terraform

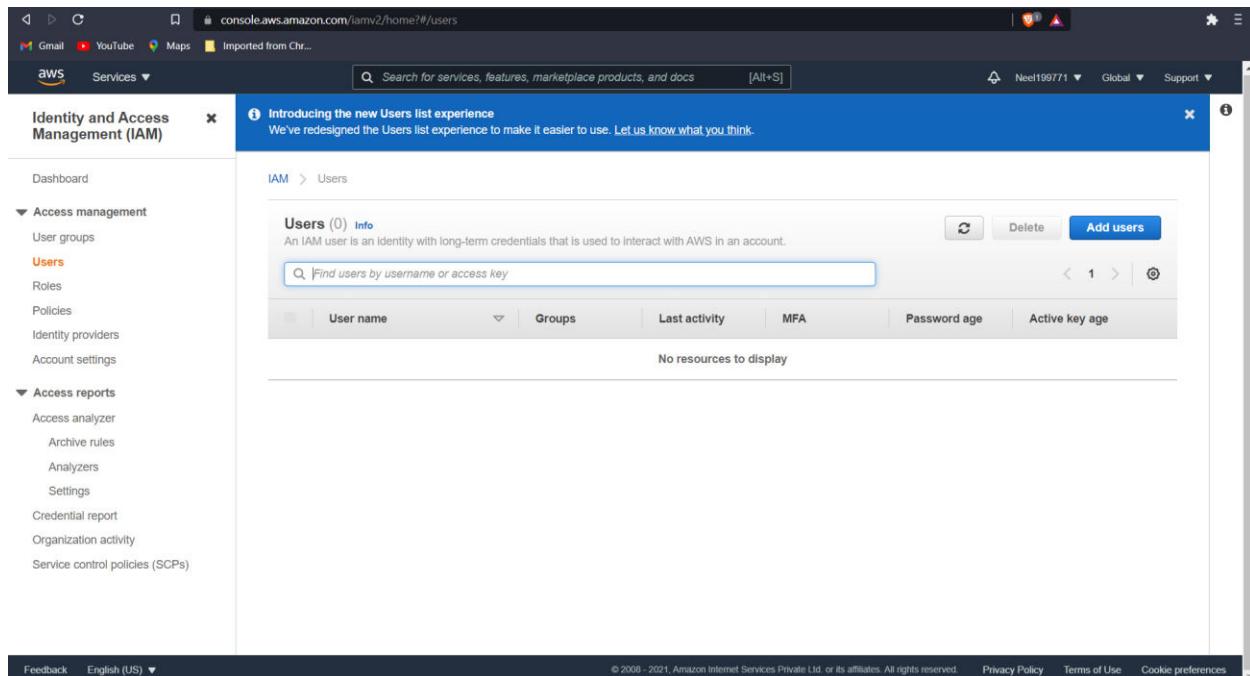
### **Steps :-**

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently.

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



2) Now click on Add Users in The User Section as shown in the image



Identity and Access Management (IAM)

Introducing the new Users list experience

We've redesigned the Users list experience to make it easier to use. Let us know what you think.

IAM > Users

Users (0) Info

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

Find users by username or access key

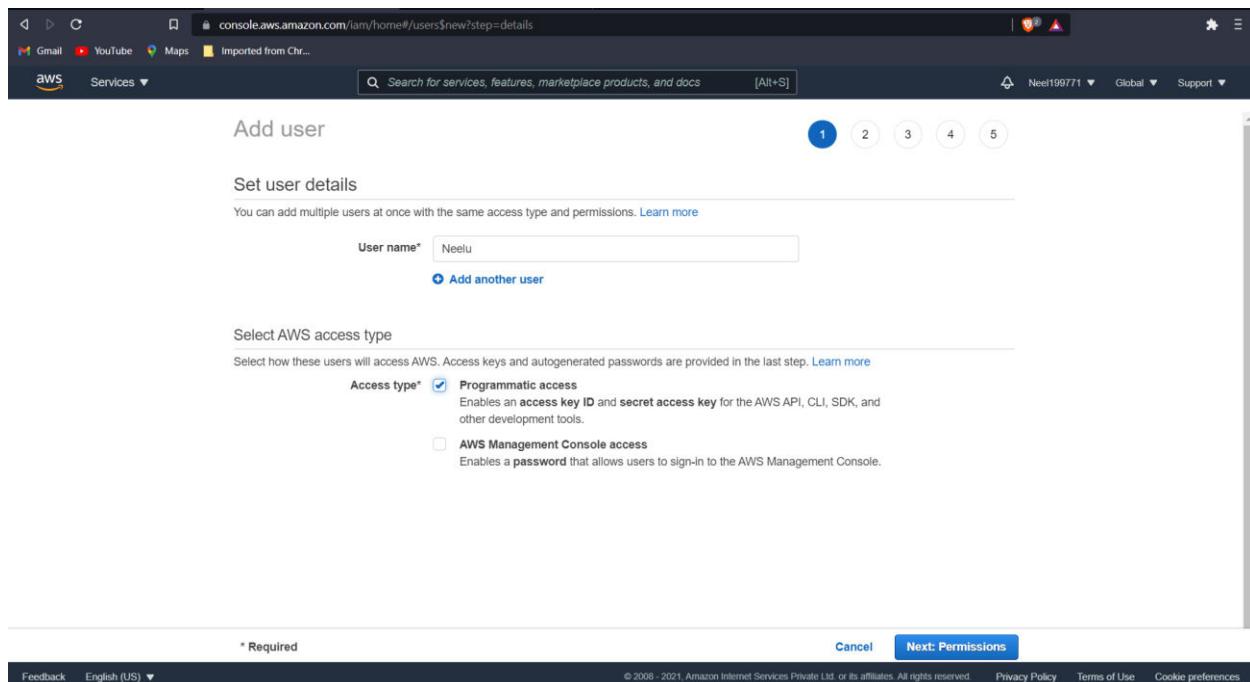
User name Groups Last activity MFA Password age Active key age

No resources to display

Add users

Feedback English (US) © 2006 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

### 3) Now give any name For username And Check The Programmatic Access field shown below



Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel **Next: Permissions**

Feedback English (US) © 2006 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

### 4) Add Group name and Check the first Policy Name

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions.

Learn more

Group name: AdvDevops

Create policy Refresh

Filter policies Search Showing 669 results

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> AdministratorAccess	Job function	None	Provides full access to AWS services and resources.
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct access to AWS services.
<input type="checkbox"/> AdministratorAccess-AWSElastic...	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and administrators to manage AWS services.
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	None	Provides device setup access to AlexaForBusiness services.
<input type="checkbox"/> AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to related AWS services.

Create group

## 5) Don't add tags

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Neelu
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	AdvDevops

Tags

No tags were added.

Create user

## 6) Now Download .csv file

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://99616344093.signin.aws.amazon.com/console>

[Download .csv](#)

User	Access key ID	Secret access key
Neelu	AKIA6P37X2EDTWPP417L	***** Show

## 7) Go to services and Ec2

Waiting for console.aws.amazon.com...

[new\\_user\\_credentials.csv](#)

## 8) Again google search the following terms

Google search results for "creating ec2 instances using terform". The search bar shows the query. Below it, a snippet from a ThinkStack blog post is displayed, followed by a link to the full article and a "File folder" icon.

creating ec2 instances using terform

All Videos Images News Shopping More

About 5,13,000 results (0.56 seconds)

Showing results for **creating ec2 *instance* using *terraform***  
Search instead for [creating ec2 instances using terform](#)

**Using Terraform to Create an EC2 Instance**

1. Create an EC2 Instance.
2. Automatically look up the latest Windows Server 2019 AMI for the EC2 instance.
3. Create and attach a additional drive.
4. Create a Cloudwatch Alarm Metric to monitor CPU.

<https://www.thinkstack.co/blog/using-terraform-to-create-an-ec2-instance>

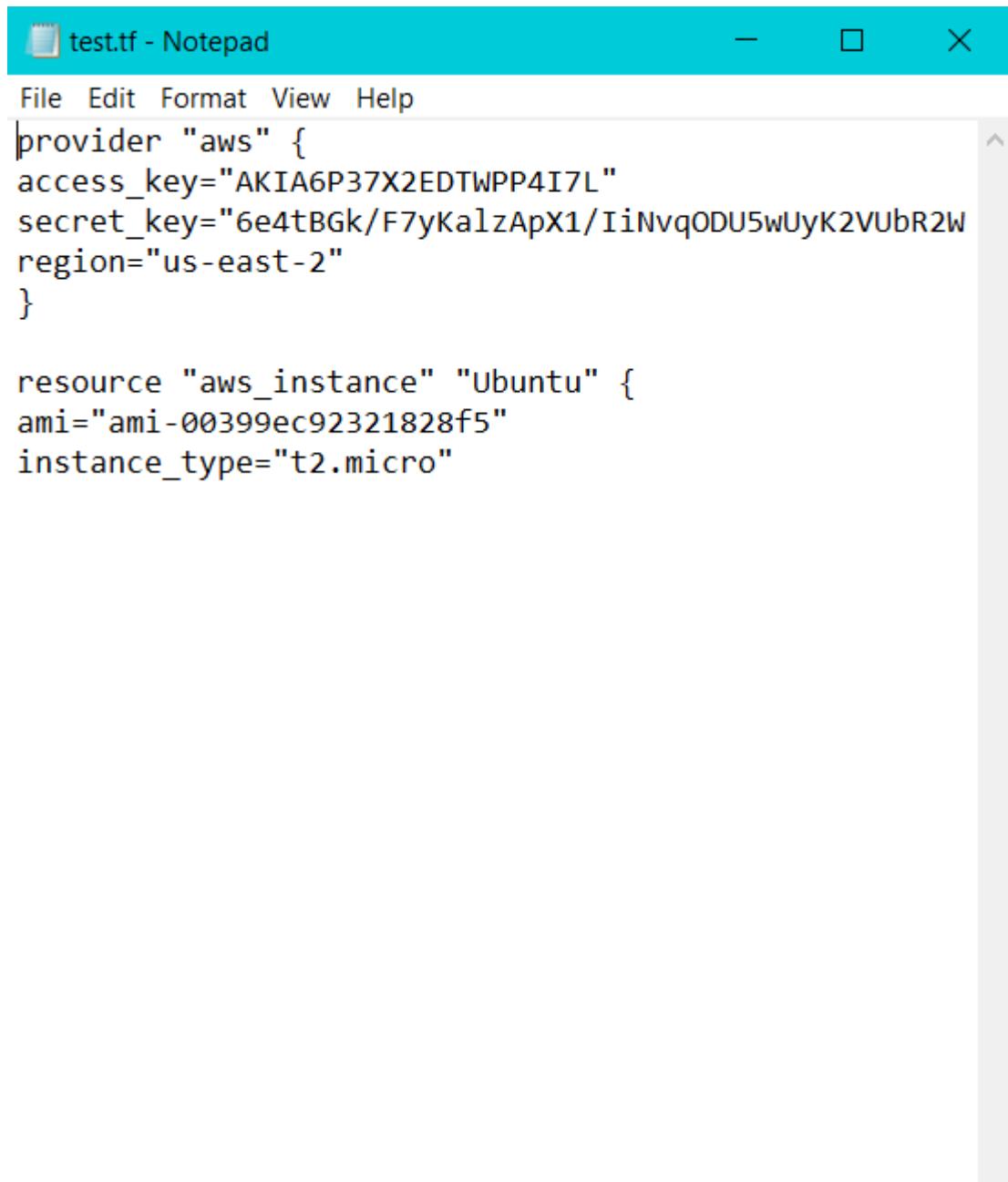
**Using Terraform to Create an EC2 Instance - ThinkStack**

About featured snippet

Created a folder Name Terraform Scripts in the C drive where the AdvDevops folder was created



**9) Now Go to note pad And Type the below Details properly But before It Just Change the ACCESS KEY AND SECRECT KEY TO THE ONE IN YOUR .csv File . Set region same as below if you want MUMBAI as your region.**

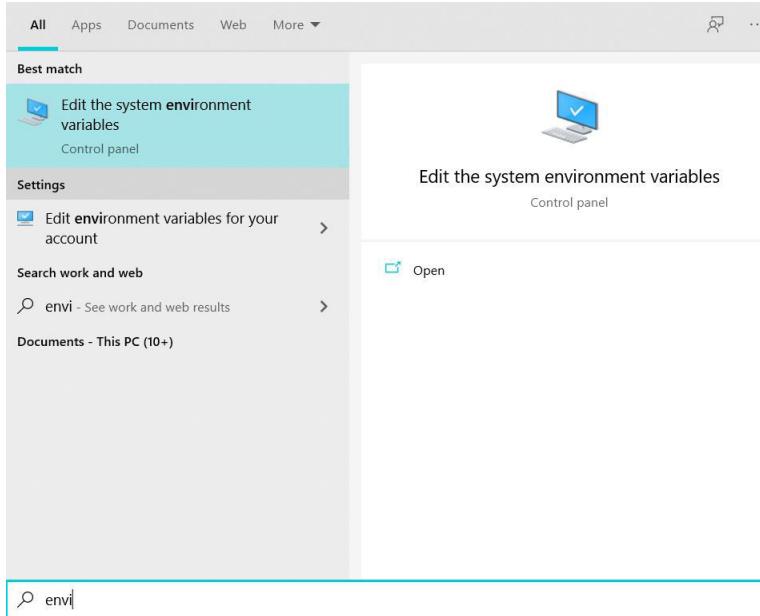


```
test.tf - Notepad
File Edit Format View Help
provider "aws" {
  access_key="AKIA6P37X2EDTWPP4I7L"
  secret_key="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
  region="us-east-2"
}

resource "aws_instance" "Ubuntu" {
  ami="ami-00399ec92321828f5"
  instance_type="t2.micro"
```

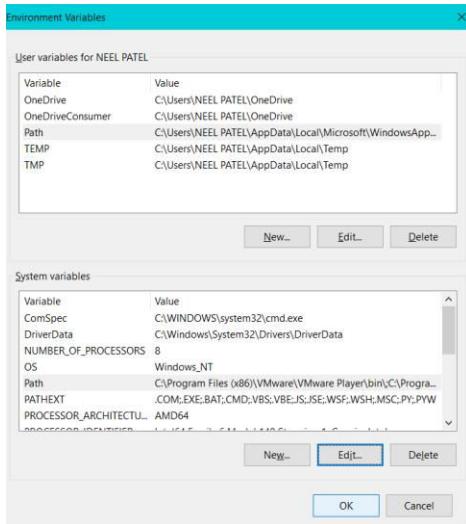
**10) Now search EDIT THE SYSTEM ENVIRONMENT VARIABLES in your windows search.**

**Open it**



**11) 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.**



**12) Now Open Command Prompt and then pate the path of Terraform script**

**Eg. CD C:\Terraform Script as shown below**

**Now type Terraform Init command**

```
C:\Terraform Script>terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.52.0...
- Installed hashicorp/aws v3.52.0 (signed by HashiCorp)

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

Terraform has been successfully initialized!

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

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

Then if there are no errors type **Terraform Plan** as shown below (type YES when command prompt ask)

```
C:\Terraform Script>terraform plan
```

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

+ create

Terraform will perform the following actions:

```
+ resource "aws_instance" "Ubuntu" {
  + ami                               = "ami-0c1a7f89451184c8b"
  + arn                               = (known after apply)
  + associate_public_ip_address       = (known after apply)
  + availability_zone                 = (known after apply)
  + cpu_core_count                   = (known after apply)
  + cpu_threads_per_core            = (known after apply)
  + disable_api_termination        = (known after apply)
  + ebs_optimized                   = (known after apply)
  + get_password_data               = false
  + host_id                          = (known after apply)
```

Now Finally Type **Terraform Apply**

```
C:\Terraform Script>terraform apply
```

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

+ create

Terraform will perform the following actions:

```
+ resource "aws_instance" "Ubuntu" {
  + ami                               = "ami-0c1a7f89451184c8b"
  + arn                               = (known after apply)
  + associate_public_ip_address       = (known after apply)
  + availability_zone                 = (known after apply)
  + cpu_core_count                   = (known after apply)
```

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

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

Enter a value: yes

aws_instance.Ubuntu: Creating...
aws_instance.Ubuntu: Still creating... [10s elapsed]
aws_instance.Ubuntu: Still creating... [20s elapsed]
aws_instance.Ubuntu: Still creating... [30s elapsed]
aws_instance.Ubuntu: Creation complete after 31s [id=i-0eac948a456860494]

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

**13) Now go to EC2 and check that is an instance created by the name of UBUNTU and is it in running status or not If it is in Running Status then Come back to Command prompt And Terminate the Instance by**

**Typing - Terraform destroy**

```
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.Ubuntu: Destroying... [id=i-0eac948a456860494]
aws_instance.Ubuntu: Still destroying... [id=i-0eac948a456860494, 10s elapsed]
aws_instance.Ubuntu: Still destroying... [id=i-0eac948a456860494, 20s elapsed]
aws_instance.Ubuntu: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.

C:\Teraform Script>
```

**14) Now go back to EC2 if the instance is terminated, if yes then logout of the Aws Console. And close the command prompt!**

**Conclusion :-**

Terraform is a powerful Infrastructure as Code (IaC) tool that automates the provisioning, management, and destruction of AWS infrastructure. It can help you to save time, reduce errors, and improve the consistency of your infrastructure.

**Name :-Niranjan Rajesh Joshi**

**Roll No:- 2105051**

**Batch:- T13**

**Date Of Performance :- 29/08/2023**

## **Experiment 7**

**Aim :-**

**to perform static analysis on python programs using sonarqube SAST process**

**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.

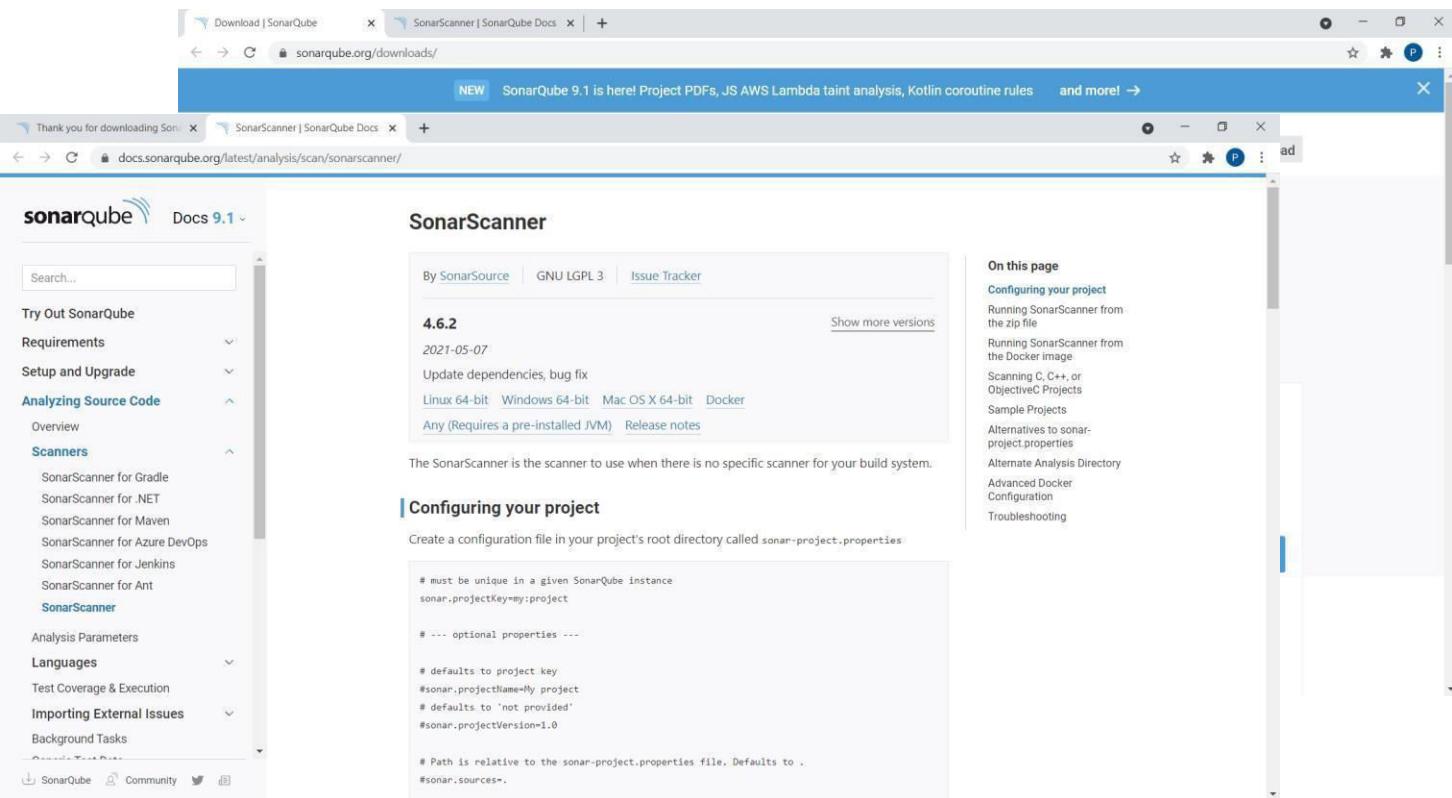
To avoid these issues in code, developers should always follow the good coding practice, but sometimes it is not possible to follow the rules and maintain the good quality as there may be many reasons.

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 :-

### 1) Download sonarqube



The screenshot shows a web browser window with the URL [sonarqube.org/downloads/](https://sonarqube.org/downloads/). The main content is the 'SonarScanner' documentation page for SonarQube 9.1. It features a sidebar with links to 'Scanners', 'Languages', and 'Importing External Issues'. The main content area has a header 'SonarScanner' and a sub-header '4.6.2'. It includes a code block for `sonar-project.properties` and a 'On this page' sidebar with various configuration and troubleshooting links.

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

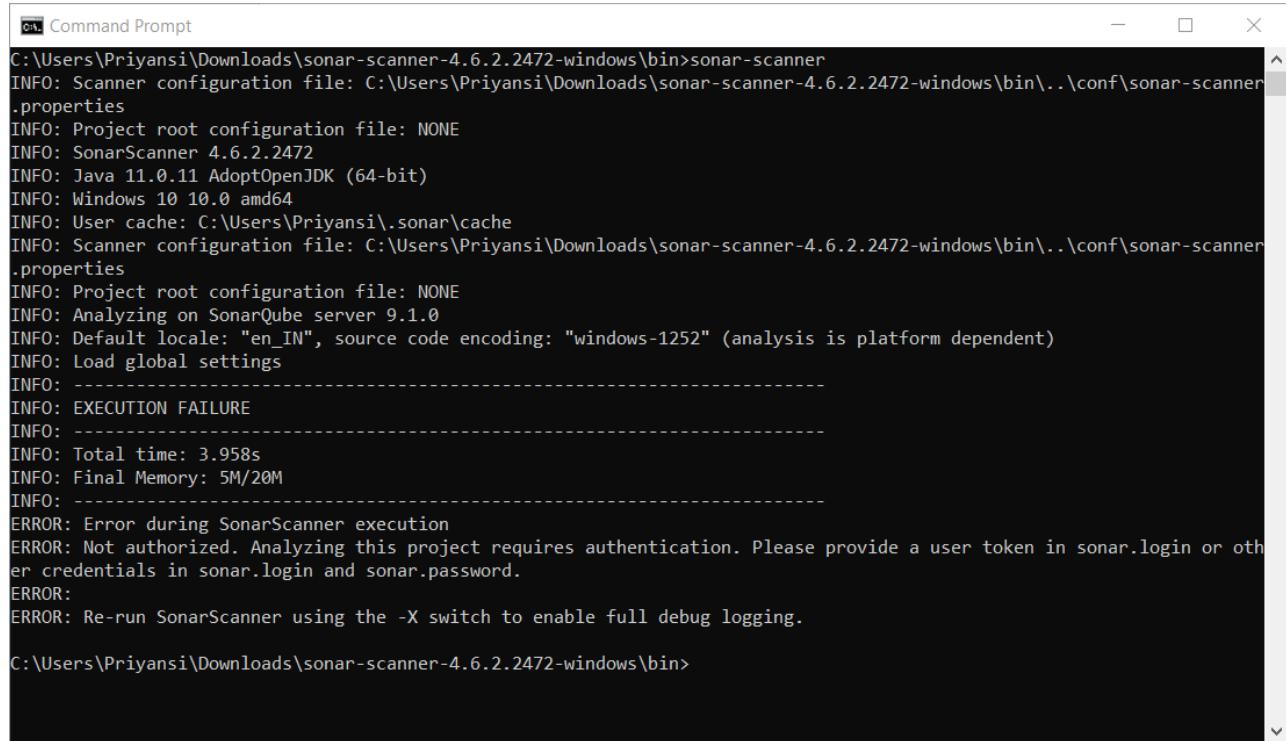
### 3) Open command prompt. Run commands:

- `cd "sonarqube-9.1.0.47736\bin\windows-x86-64"`
  - `StartSonar.bat`

```
jvm 1      at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1702)
jvm 1      at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1672)
jvm 1      at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:119)
jvm 1      at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
jvm 1      at org.sonar.application.process.EsManagedProcess.checkHttpStatus(EsManagedProcess.java:90)
jvm 1      at org.sonar.application.process.EsManagedProcess.checkOperational(EsManagedProcess.java:75)
jvm 1      at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:60)
jvm 1      at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1      at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1  Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
jvm 1      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1      at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1      at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1      ... 10 common frames omitted
jvm 1  Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1      at org.apache.http.nio.pool.RouteproxySpecificPool.timeout(RouteproxySpecificPool.java:160)
jvm 1      at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimeout(AbstractNIOConnPool.java:628)
jvm 1      at org.apache.http.nio.pool.AbstractNIOConnPool$1.timeout(AbstractNIOConnPool.java:894)
jvm 1      at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
jvm 1      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1      at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1      at org.apache.http.impl.nio.com.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.java:221)
jvm 1      at org.apache.http.impl.nio.client.CloseableHttpSyncClientBase$1.run(CloseableHttpSyncClientBase.java:64)
jvm 1      at java.base/java.lang.Thread.run(Thread.java:834)
jvm 1  2021.09.29 13:50:50 INFO [app][o.s.a.ProcessLauncherImpl] Process(es) is up
jvm 1  2021.09.29 13:50:50 INFO [app][o.s.a.ProcessLauncherImpl] Launch process[[key="web", ipcIndex=2, logfilenamePrefix=web]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\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/jdk.internal.management=ALL-UNNAMED --add-opens=java.base/jdk.management.internal=ALL-UNNAMED -Xms512m -Xms128m -XX:HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*[:1]] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.server.web.AppServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process1779451691724410119\properties
jvm 1  | 2021.09.29 13:51:42 INFO [app][o.s.a.SchedulerImpl] Process(es) is up
jvm 1  | 2021.09.29 13:51:42 INFO [app][o.s.a.ProcessLauncherImpl] Launch process[[key="ce", ipcIndex=3, logfilenamePrefix=ce]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=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.management=ALL-UNNAMED --add-opens=java.base/jdk.management.internal=ALL-UNNAMED -Xms128m -XX:HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*[:1]] -cp ./lib/sonar-application-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process394448741319563\properties
jvm 1  2021.09.29 13:51:42 WARN app[] [startup] ######
jvm 1  2021.09.29 13:51:42 WARN app[] [startup] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
jvm 1  2021.09.29 13:51:42 WARN app[] [startup] ######
jvm 1  2021.09.29 13:51:46 INFO [app][o.s.a.SchedulerImpl] Process(es) is up
jvm 1  2021.09.29 13:51:46 INFO [app][o.s.a.SchedulerImpl] SonarQube is up
```

**4) Open another command prompt. Run command:**

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

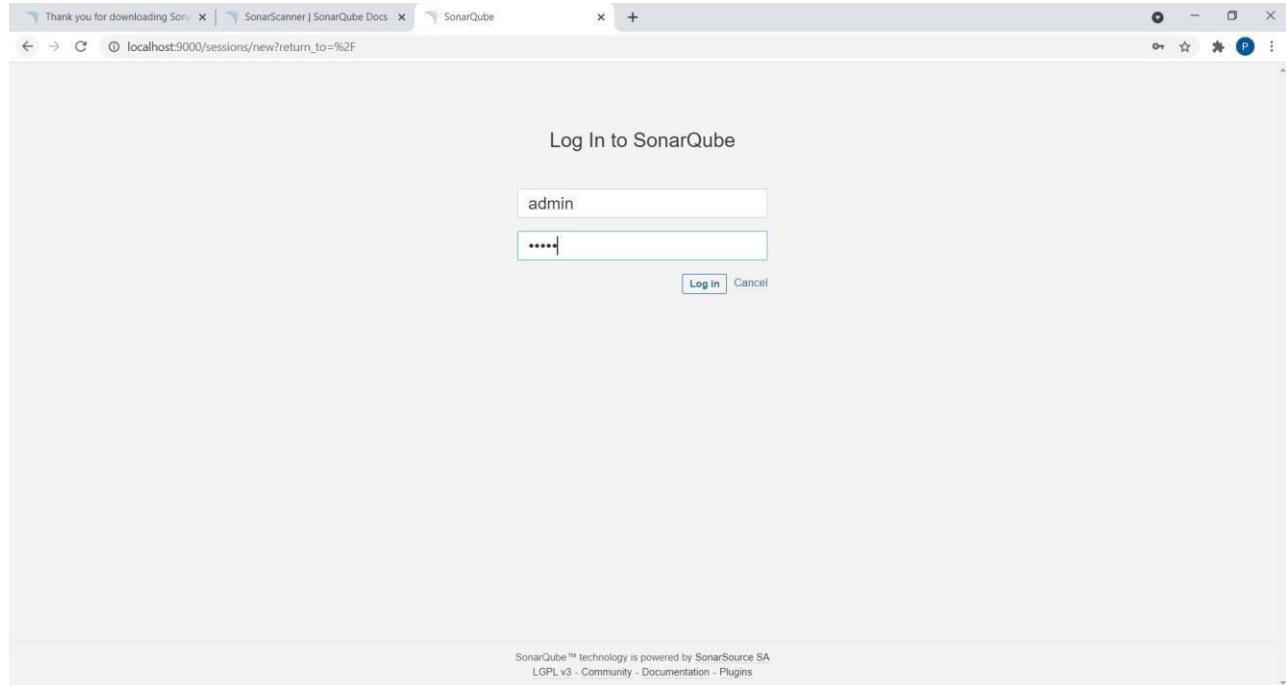


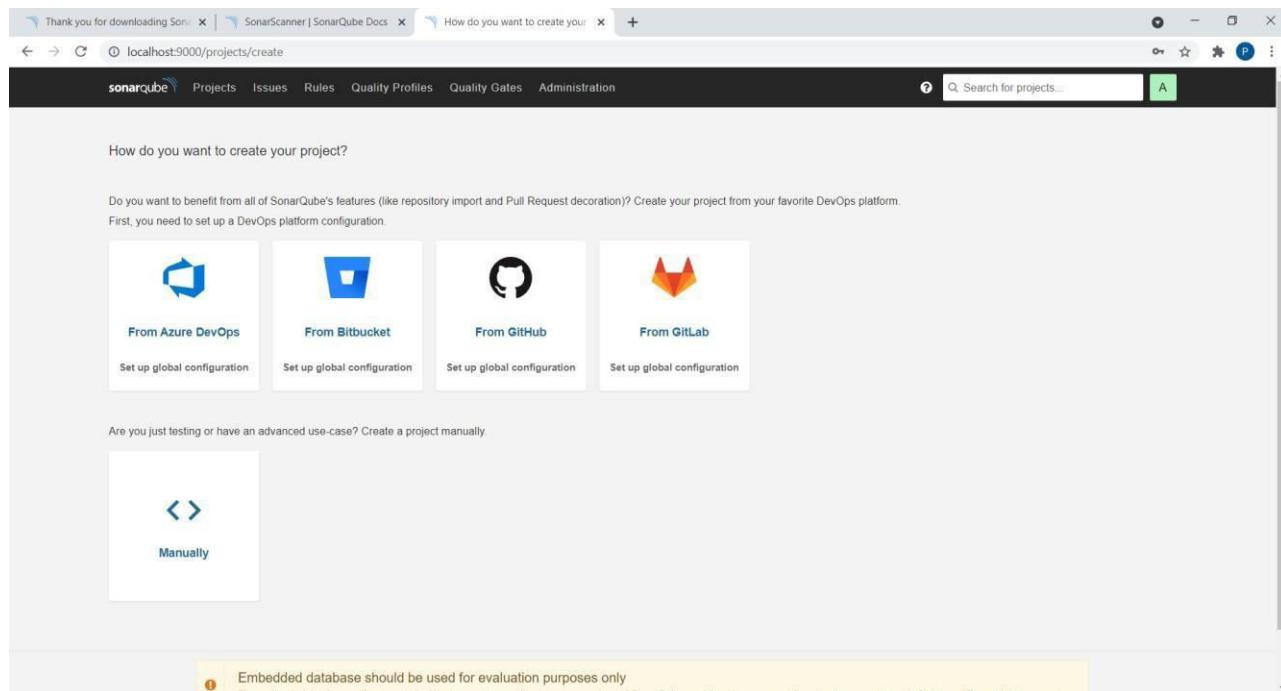
```
Command Prompt
C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>sonar-scanner
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
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>
```

**5) 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

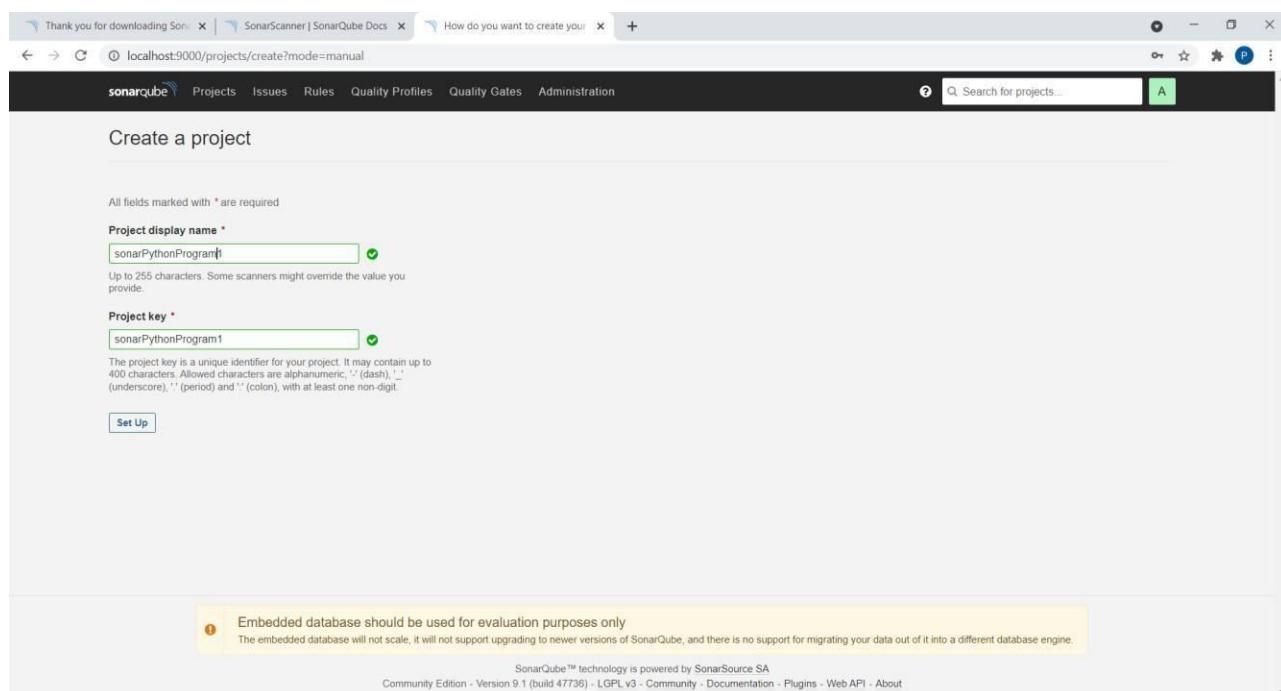
Set up global configuration      Set up global configuration      Set up global configuration      Set up global configuration

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

Manually

**Embedded database should be used for evaluation purposes only**  
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.

## 6) Click on Create a project Manually.



Create a project

All fields marked with \* are required

**Project display name \***  
sonarPythonProgram1

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

**Project key \***  
sonarPythonProgram1

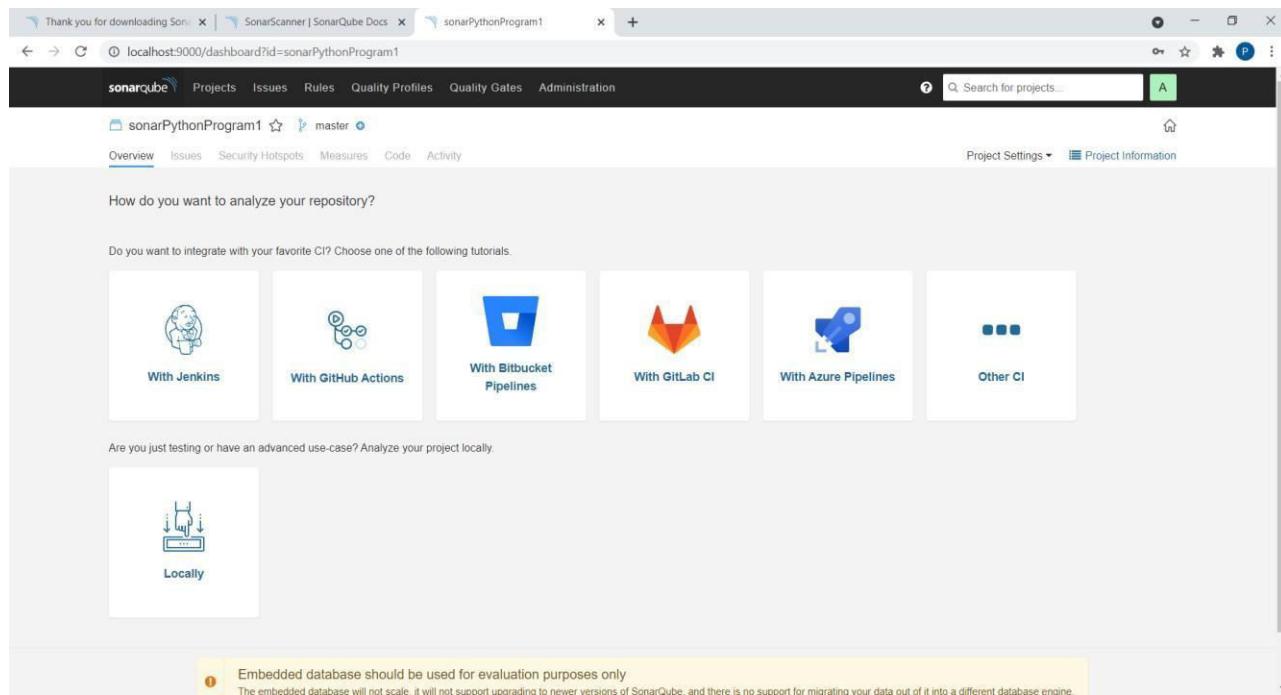
The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '\_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

**Set Up**

**Embedded database should be used for evaluation purposes only**  
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.

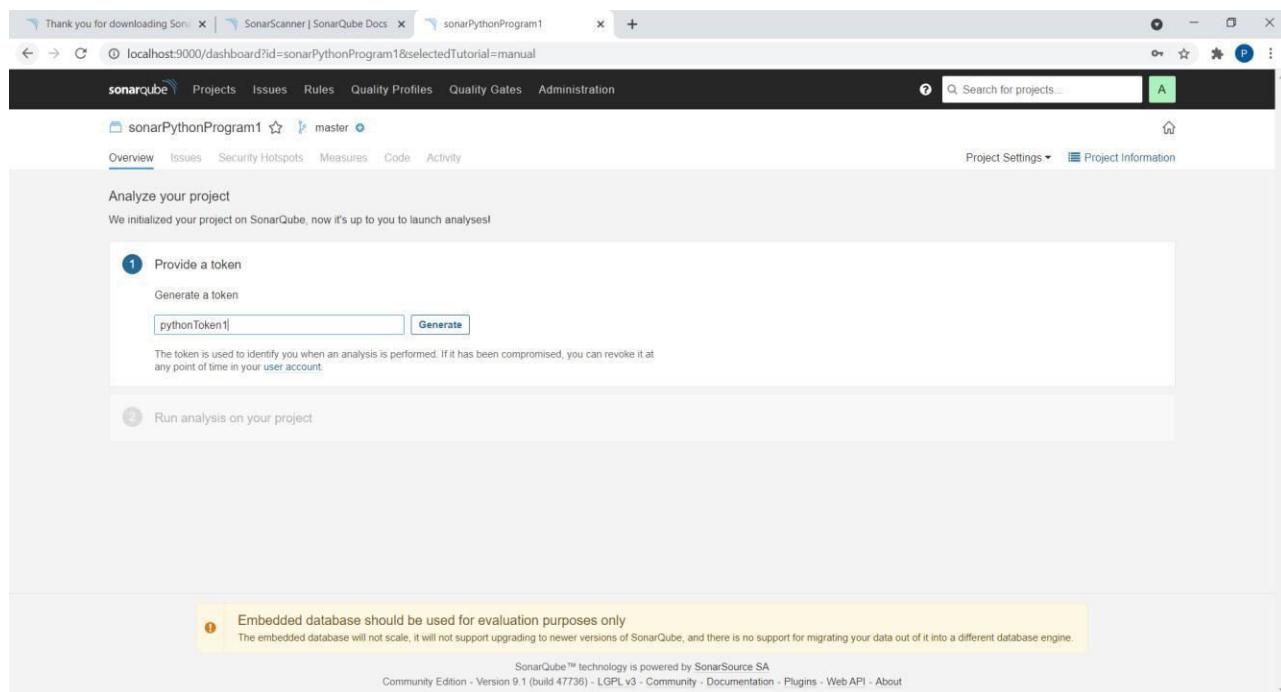
SonarQube™ technology is powered by SonarSource SA  
Community Edition - Version 9.1 (build 47736) - LGPL v3 - Community - Documentation - Plugins - Web API - About

## 7) 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 'sonarPythonProgram1'. The 'sonarPythonProgram1' tab is active, showing the project's status: 'sonarPythonProgram1' (master branch). The dashboard features a search bar and navigation links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. Below the navigation, there are tabs for 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. A 'Project Settings' dropdown and a 'Project Information' link are also present. The main content area is titled 'How do you want to analyze your repository?' and includes a section for CI integration with Jenkins, GitHub Actions, Bitbucket Pipelines, GitLab CI, Azure Pipelines, and other CI options. Another section for local analysis is shown with a 'Locally' button. 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'. The URL in the address bar includes '&selectedTutorial=manual'. The dashboard layout is identical to the previous screenshot, with tabs for 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. 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.' The main content area is titled 'Analyze your project' and includes a note: 'We initialized your project on SonarQube, now it's up to you to launch analyses!' Below this, a step-by-step guide is shown: '1 Provide a token' with a 'Generate token' button and a note about token usage, and '2 Run analysis on your project'. The bottom of the page includes the SonarQube footer: 'SonarQube™ technology is powered by SonarSource SA', 'Community Edition - Version 9.1 (build 47736) - LGPL v3 - Community - Documentation - Plugins - Web API - About'.

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

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

1 Provide a token  
pythonToken1: 41740dddf269d68dfda1ec55f28cd250be46d48f   
The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

2 Run analysis on your project

Embedded database should be used for evaluation purposes only  
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

**Click on Continue.**

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

1 Provide a token  
pythonToken1: 41740dddf269d68dfda1ec55f28cd250be46d48f   
2 Run analysis on your project  
What option best describes your build?  
 Maven  Gradle  .NET  Other (for JS, TS, Go, Python, PHP, ...)  
What is your OS?  
 Linux  Windows  macOS  
Download and unzip the Scanner for Windows  
Visit the official documentation of the Scanner to download the latest version, and add the bin directory to the %PATH% environment variable  
Execute the Scanner  
Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.  
sonar-scanner.bat -Dsonar.projectKey=sonarPythonProgram1 -Dsonar.sources= -Dsonar.host.url=http://localhost:9000 -Dsonar.login=41740dddf269d68dfda1ec55f28cd250be46d48f   
Please visit the official documentation of the Scanner for more details.

(IN SONAR SCANNER FOLDER)

**10) 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("CDXL
III"))

```

**11) 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=41740dddf269d68dfda1ec55f28cd250be46d48f"**

```

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=41740dddf269d68dfda1ec55f28cd250be46d48f"
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 10.0 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=29ms
INFO: Load global settings (done) | time=29ms
INFO: Service id: 0f41a1f2-48e9-4b8e-215c
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: Process project properties (done) | time=20ms
INFO: Execute project builders
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: 'sonarPythonProgram1'
INFO: Load project settings for component key: 'sonarPythonProgram1' (done) | time=40ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=20ms
INFO: Load active rules
INFO: Load active rules (done) | time=4452ms
WARN: SCM provider not detected 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: Sensor Python Sensor [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

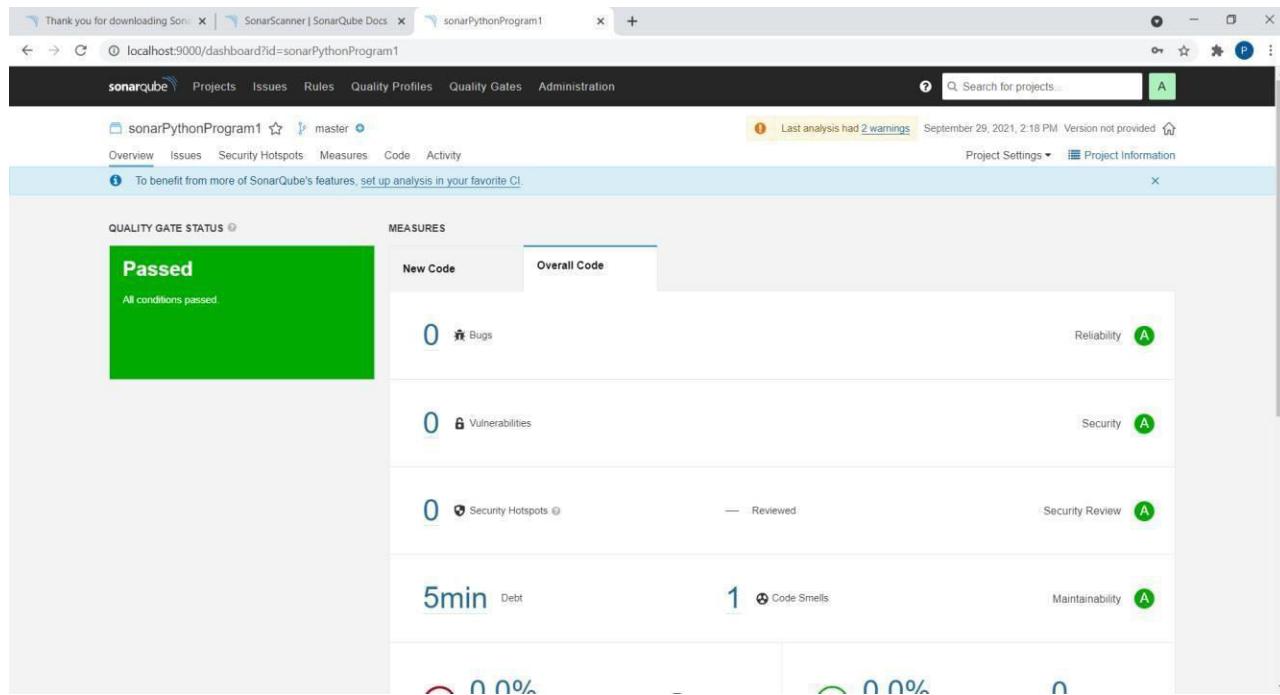
```

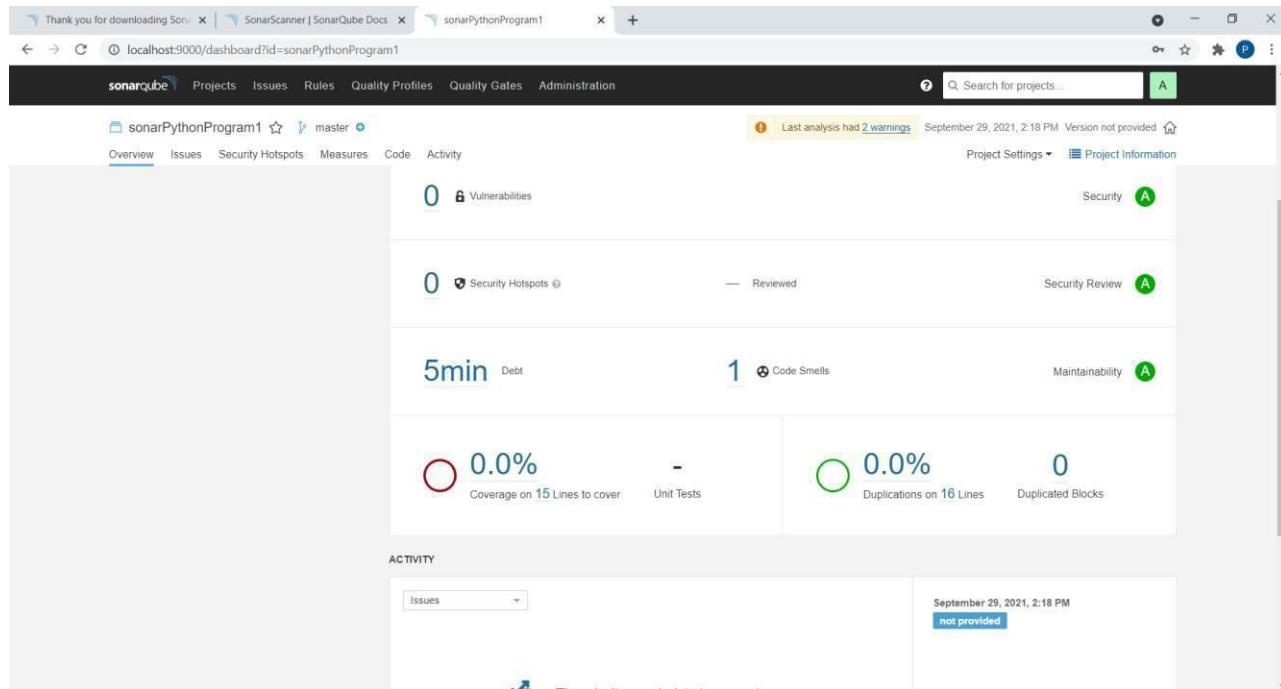
```

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: Sensor SCM Publisher [vbnet] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=12ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor Calculating CPD for 1 file
INFO: CPD Executor CPD calculation finished (done) | time=10ms
INFO: Analysis report generated in 59ms, dir size=103.9 kB
INFO: Analysis report compressed in 19ms, zip size=14.7 kB
INFO: Analysis report uploaded in 76ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=sonarPythonProgram1
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AxwwV1hx91b8xeZLXH1
INFO: Analysis total time: 7.502 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total Time: 10.887s
INFO: Final Memory: 7H/30M
INFO: -----
C:\Users\Priyansi\Documents\SonarExps>

```

**13) 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.**





sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

sonarPythonProgram1 master

Overview Issues Security Hotspots Measures Code Activity

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

Project Settings Project Information

0 Vulnerabilities

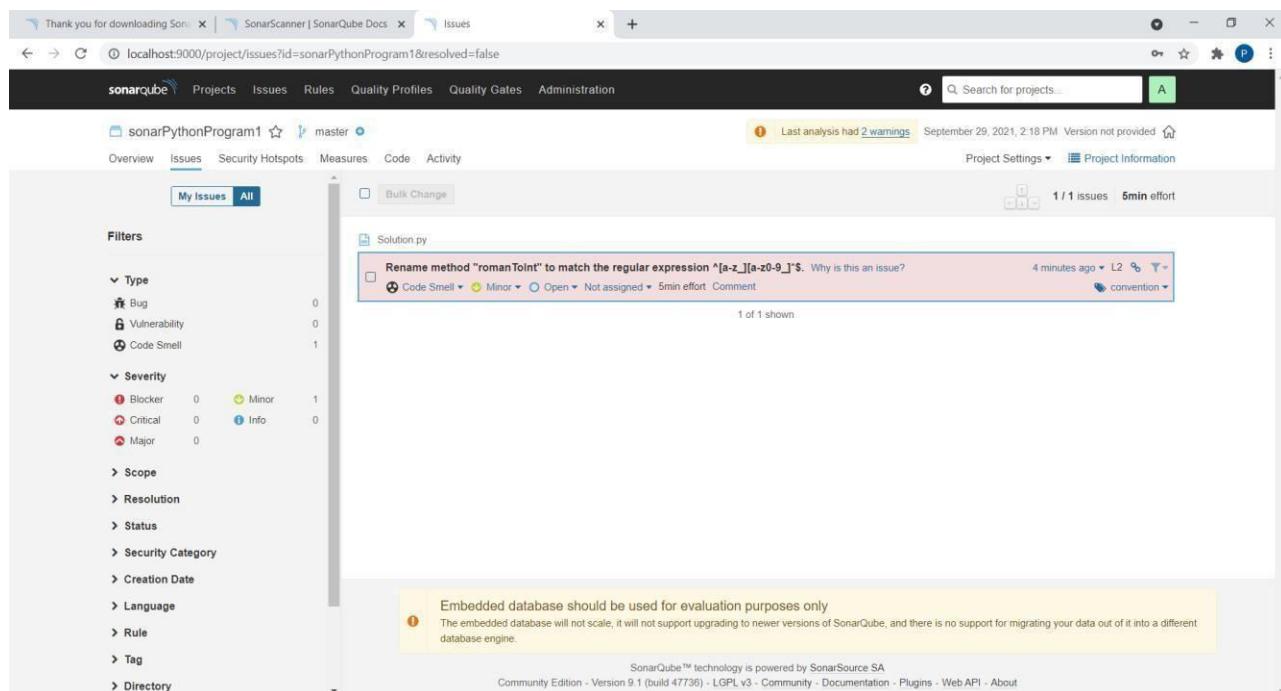
0 Security Hotspots Reviewed Security Review

5min Debt 1 Code Smells Maintainability

0.0% Coverage on 15 Lines to cover Unit Tests 0.0% Duplications on 16 Lines Duplicated Blocks

ACTIVITY

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



sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

sonarPythonProgram1 master

Overview Issues Security Hotspots Measures Code Activity

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

Project Settings Project Information

My Issues All

Bulk Change 1 / 1 issues 5min effort

**Solution.py**

Rename method "romanToInt" to match the regular expression ^[a-z\_][a-z0-9\_]\*\$. Why is this an issue? 4 minutes ago L2

Code Smell  Minor  Open  Not assigned 5min effort Comment convention

1 of 1 shown

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

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

Press "Ctrl + C" to stop the server.

```

xx 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:228)
jvm 1 | at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1 | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1 | at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
jvm 1 | at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1 | at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1 | at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1 | ... 10 common frames omitted
jvm 1 | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1 | at org.apache.http.nio.pool.RouteSpecificPool$1.route(RouteSpecificPool.java:169)
jvm 1 | at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimeout(AbstractNIOConnPool.java:528)
jvm 1 | at org.apache.http.nio.pool.AbstractNIOConnPool$InternalSessionRequestCallback.timeout(AbstractNIOConnPool.java:894)
jvm 1 | at org.apache.http.impl.nio.reactor.SessionRequestImpl.timeout(SessionRequestImpl.java:184)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1 | at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1 | at org.apache.http.impl.nio.conn.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.java:221)
jvm 1 | at org.apache.http.impl.nio.client.CloseableHttpAsyncClient$Base$1.run(CloseableHttpAsyncClient$Base.java:64)
jvm 1 | at java.base/java.lang.Thread.run(Thread.java:834)
jvm 1 | [2021-09-29 13:50:59 INFO app[]|o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | [2021-09-29 13:50:59 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.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.0.0.1::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.server.app.WebServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process177945169172441019\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 INFO app[]|o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-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.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.0.0.1::1] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp\sq-process3944487414319503\sq.properties
jvm 1 | [2021-09-29 13:51:42 WARN app[]|o.s.a.SchedulerImpl] [Startup] #####
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] [Startup] #####
jvm 1 | [2021-09-29 13:51:46 INFO app[]|o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | [2021-09-29 13:51:46 INFO app[]|o.s.a.SchedulerImpl] SonarQube is up
wrapper | CTRL-C trapped. Shutting down.
jvm 1 | [2021-09-29 14:38:57 INFO app[]|o.s.a.SchedulerImpl] Stopping SonarQube
jvm 1 | [2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl] Process[ce] is stopped
jvm 1 | [2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl] Process[web] is stopped
jvm 1 | [2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl] Process[es] is stopped
jvm 1 | [2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl] SonarQube is stopped
wrapper | <-- Wrapper Stopped
Terminate batch job (Y/N)? y

```

## Conclusion :-

The SonarQube SAST process is a powerful tool for performing static analysis on Python programs. It can help you to identify and fix security vulnerabilities and code quality issues, improving the overall security and quality of your software.

**Name :-Niranjan Rajesh Joshi**

**Roll No:- 2105051**

**Batch:- T13**

**Date Of Performance :- 12/09/2023**

**Experiment 8**

**Aim:-**

To understand continuous monitoring using Nagios

**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.

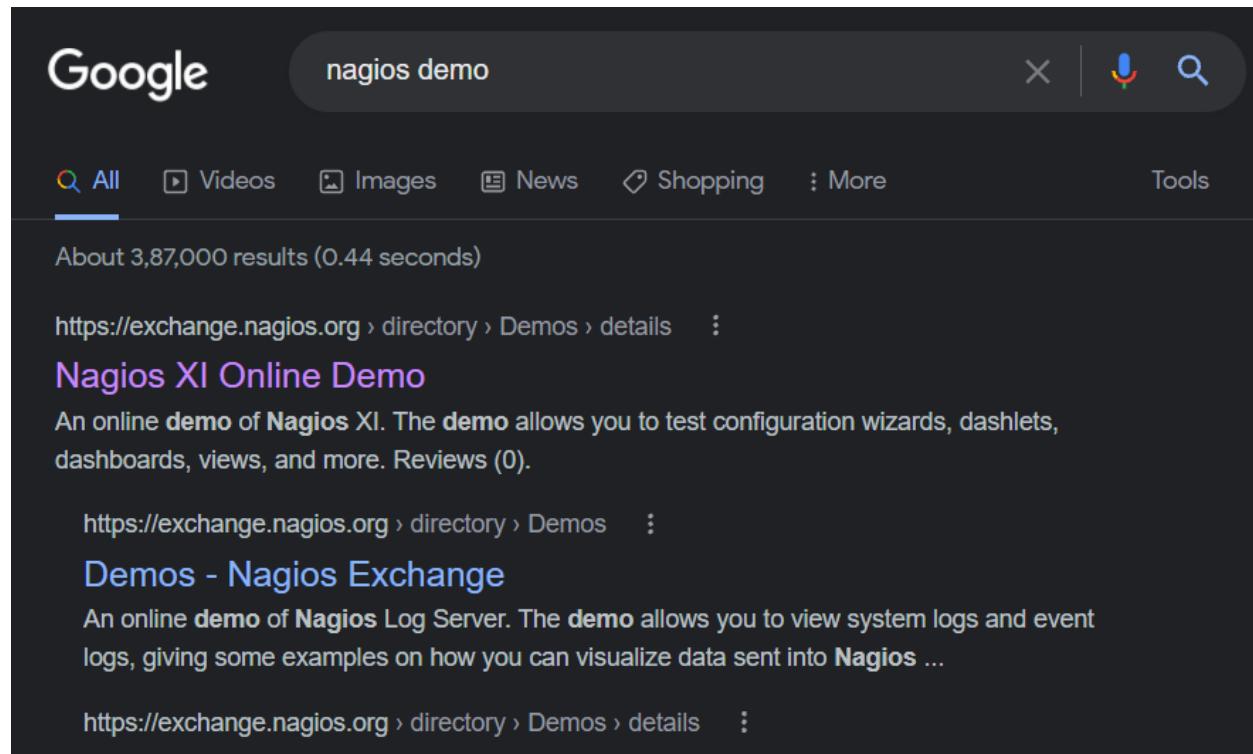
**Centralized Monitoring:** Nagios provides a centralized view of the entire IT infrastructure, allowing administrators to have a comprehensive overview of the health and performance of all monitored resources from a single location.

**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



2) Now click on the website-

## 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...

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 a web browser window with the following details:

- Address Bar:** nagiosxi.demos.nagios.com/nagiosxi/login.php
- Page Headers:** Nagios XI Login
- Left Panel (Login):**
  - Login Form:** Username, Password, Login button.
  - Forgot your password?** link.
  - Select Language:** dropdown menu with various flags.
- Right Panel (Demo System):**
  - Nagios XI Demo System** header.
  - Demo Account Options:** You can access the demo with different accounts to get a different view of the monitoring system.
  - Administrator Access:** Username: nagiosadmin, Password: nagiosadmin. [Log in as Administrator](#)
  - Read-Only User Access:** Username: readonly, Password: readonly. [Log in as Read-Only User](#)
  - Advanced User Access:** Username: advanced, Password: advanced. [Log in as Advanced User](#)
  - Normal User Access:** Username: jdoe, Password: jdoe. [Log in as Normal User](#)
  - Administrator Access (dark theme):** Username: darktheme, Password: darktheme. [Log in as Administrator](#)
- Bottom:** Demo Notes

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 web interface. The top navigation bar includes links for Home, Views, Dashboards, Reports, Configure, Tools, Help, Admin, and a user dropdown for 'nagiosadmin'. The left sidebar contains a 'Quick View' section with links to Home Dashboard, Tactical Overview, Birdseye, Operations Center, Operations Screen, Open Service Problems, Open Host Problems, All Service Problems, All Host Problems, and Network Outages. Below this are sections for 'Details' (Service Status, Host Status, Hostgroup Summary, Hostgroup Overview, Hostgroup Grid, Servicegroup Summary, Servicegroup Overview, Servicegroup Grid, BPI), 'Metrics' (Metrics), 'Graphs' (Performance Graphs, Graph Explorer), 'Maps' (World Map, B2Bmap, Hypermap, Minimap, NagVis, Network Status Map), and 'Incident Management' (Incident Management). The main content area features three summary cards: 'Host Group Status' (Summary View, showing 50 Up, 0 Down, 0 Unreachable, 6 Pending, 3 Unhandled, 3 Problems, 59 All), 'Host Status Summary' (Up: 50, Down: 0, Unreachable: 0, Pending: 6, Unhandled: 3, Problems: 3, All: 59), and 'Service Status Summary' (Ok: 63, Warning: 99, Unknown: 5, Critical: 130, Pending: 7, Unhandled: 239, Problems: 239, All: 1007). The bottom status bar indicates 'Nagios XI 5.7.2 • Check for Updates' and shows the copyright notice 'Copyright © 2008-2021 Nagios Enterprises, LLC'.

## 6) Now we click on BBMap

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

The screenshot shows the Nagios XI web interface with the BBMap Status Grid. The grid displays the status of various hosts and services across multiple columns, including CPU Usage, Disk I/O, and Network metrics. A legend on the left identifies host icons: green for up, red for down, and yellow for warning. The interface includes a top navigation bar with Home, Views, Dashboards, Reports, Configure, Tools, Help, Admin, and a search bar. On the left, a sidebar provides links for service status, host status, and various monitoring tools like BBMap, BIMap, and Network Status Map.

## 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.

**Name :-Niranjan Rajesh Joshi**  
**Roll No:- 2105051**  
**Batch:- T13**  
**Date Of Performance :- 05/09/2023**

## **Experiment 9**

### **Aim:-**

To understand AWS lambda functions and create a lambda function using python to log “an image has been added” message , once the file has been added in S3 bucket

### **Theory:-**

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS) that allows you to run code without provisioning or managing servers. With AWS Lambda, you can upload your code and the service will automatically run and scale your code based on the incoming requests or events. Here are some key features and aspects of AWS Lambda:

**Event-Driven Computing:** AWS Lambda allows you to execute your code in response to events such as changes to data in an Amazon S3 bucket, updates to a DynamoDB table, HTTP requests via Amazon API Gateway, or custom events from various AWS services.

**Serverless Architecture:** It enables you to build applications and services without the need to manage infrastructure. You are charged only for the compute time you consume, with no charge when your code is not running.

**Support for Multiple Languages:** AWS Lambda supports multiple programming languages including Node.js, Python, Java, Go, Ruby, and .NET Core, allowing you to choose the language that best suits your application.

**Automatic Scaling:** Lambda automatically scales your application by running code in response to each trigger. It can handle a few requests per day or thousands of requests per second.

**Microservices and Backend Services:** Lambda is often used to build scalable and cost-effective back-end services for mobile, web, and other applications. It is also commonly used in the development of microservices-based architectures.

**Integration with Other AWS Services:** Lambda seamlessly integrates with other AWS services, enabling you to create powerful applications using a combination of AWS services without managing servers.

**Security and Access Control:** AWS Lambda provides built-in security features, allowing you to control the execution role and access permissions for your Lambda functions through AWS Identity and Access Management (IAM).

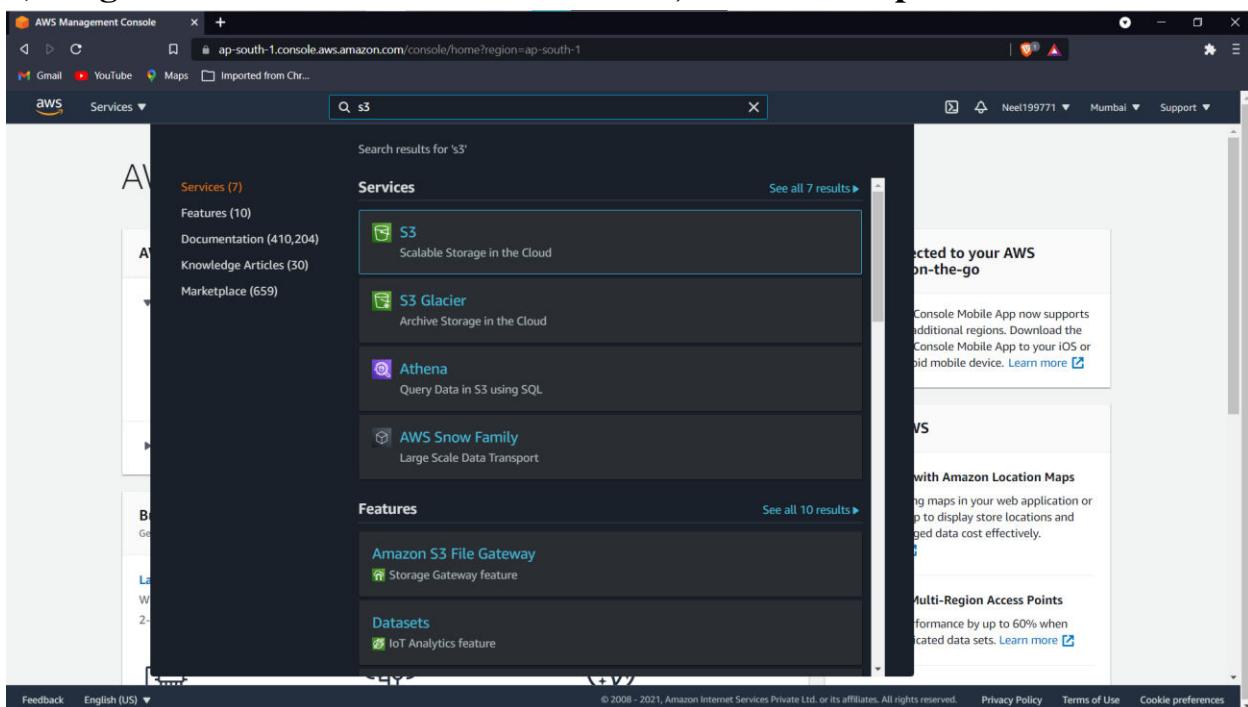
**Easy Deployment and Management:** Lambda makes it easy to deploy and manage your code. You can update your code without downtime, and the service handles all the operational and administrative activities, such as capacity provisioning, monitoring, and logging.

**High Availability and Fault Tolerance:** AWS Lambda automatically replicates your code across multiple availability zones to ensure high availability and fault tolerance.

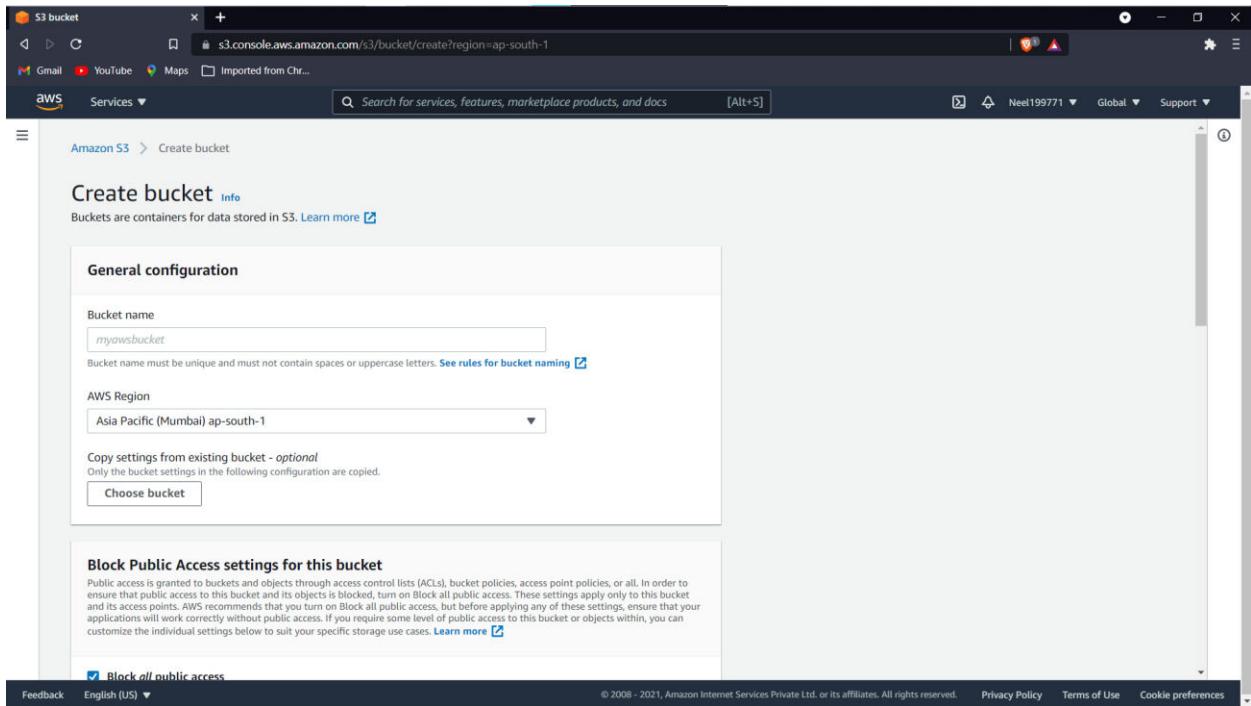
**Monitoring and Logging:** Lambda provides detailed monitoring and logging through Amazon CloudWatch, allowing you to monitor performance metrics, set alarms, and troubleshoot issues for your Lambda functions.

## Steps :-

### 1) Login to AWS account then Search S3 ,click on the option below shown-



### 2) Create an S3 bucket by giving it a name



Amazon S3 > Create bucket

**Create bucket** Info

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

**General configuration**

Bucket name  Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

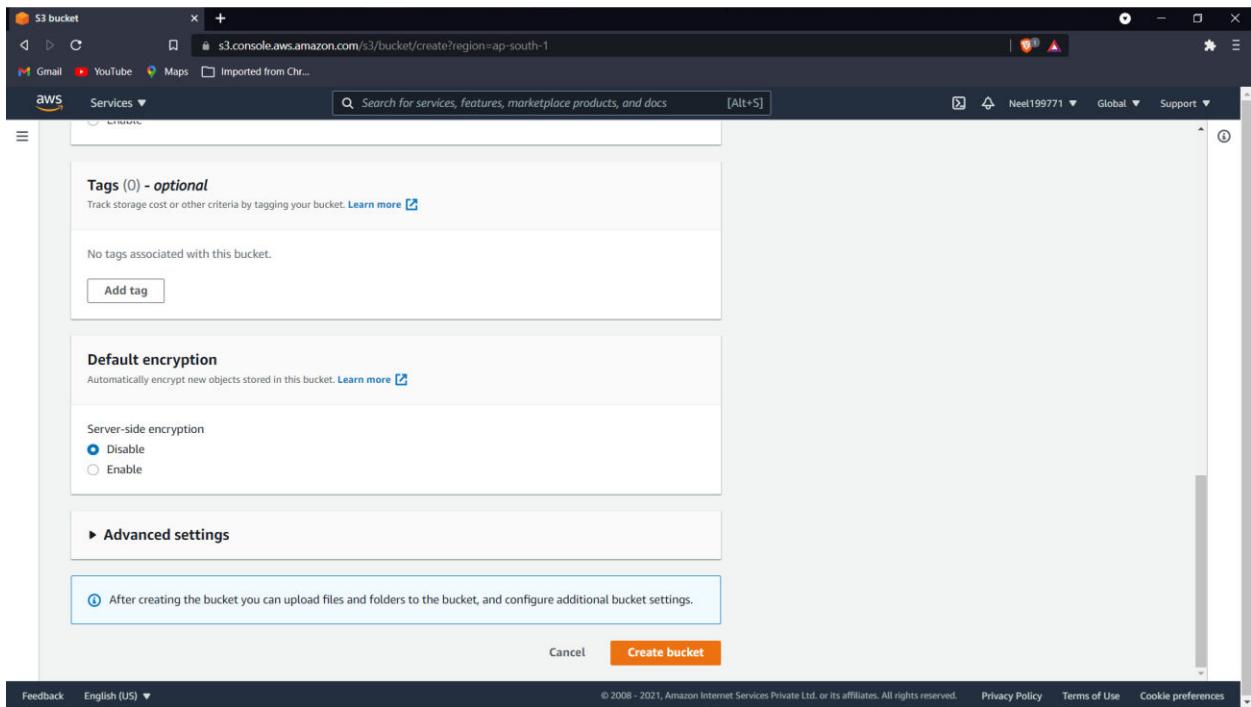
Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

**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

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



Amazon S3 > Create bucket

**Tags (0) - optional**

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

**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.

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

**3) Click on upload button after the s3 bucket is created in the object section**

The image shows two screenshots of the AWS S3 console. The top screenshot displays the 'Objects' list for the bucket 'neel-patel-t21-82'. It shows two objects, with buttons for Actions (Copy S3 URI, Copy URL, Download, Open, Delete), Create folder, and Upload. The bottom screenshot shows the 'Upload' interface, where files can be dragged and dropped or selected via 'Add files' or 'Add folder'. The 'Files and folders (0)' table shows a single entry for 'Destination'. The 'Destination' section is empty, indicating no files have been uploaded.

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

neel-patel-t21-82

Objects (2)

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

## 5) Now search lambda

Search results for 'lambda'

Services (5)

- Lambda
- CodeBuild
- AWS Signer
- Amazon Lex

Features

- Local processing
- Target groups

## 6) Click create function , click on below options and click on configure

Functions - Lambda

aws Services

AWS Lambda

Lambda > Functions

Functions (1)

Last fetched 10 seconds ago

Actions

Create function

Lambda

Services

Lambda > Functions > Create function

Create function

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

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.

Blueprints

Filter by tags and attributes or search by keyword

Export

1 2 3 4 5

kinesis-firehose-syslog-to-json

An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON.

nodejs12.x · kinesis-firehose

s3-get-object-python

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

python3.7 · s3

config-rule-change-triggered

An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types.

nodejs12.x · config

lex-book-trip-python

Book details of a visit, using Amazon Lex to perform natural language understanding

python · lex

dynamodb-process-stream

An Amazon DynamoDB trigger that logs the

microservice-http-endpoint

A simple backend read/write to

kinesis-analytics-output

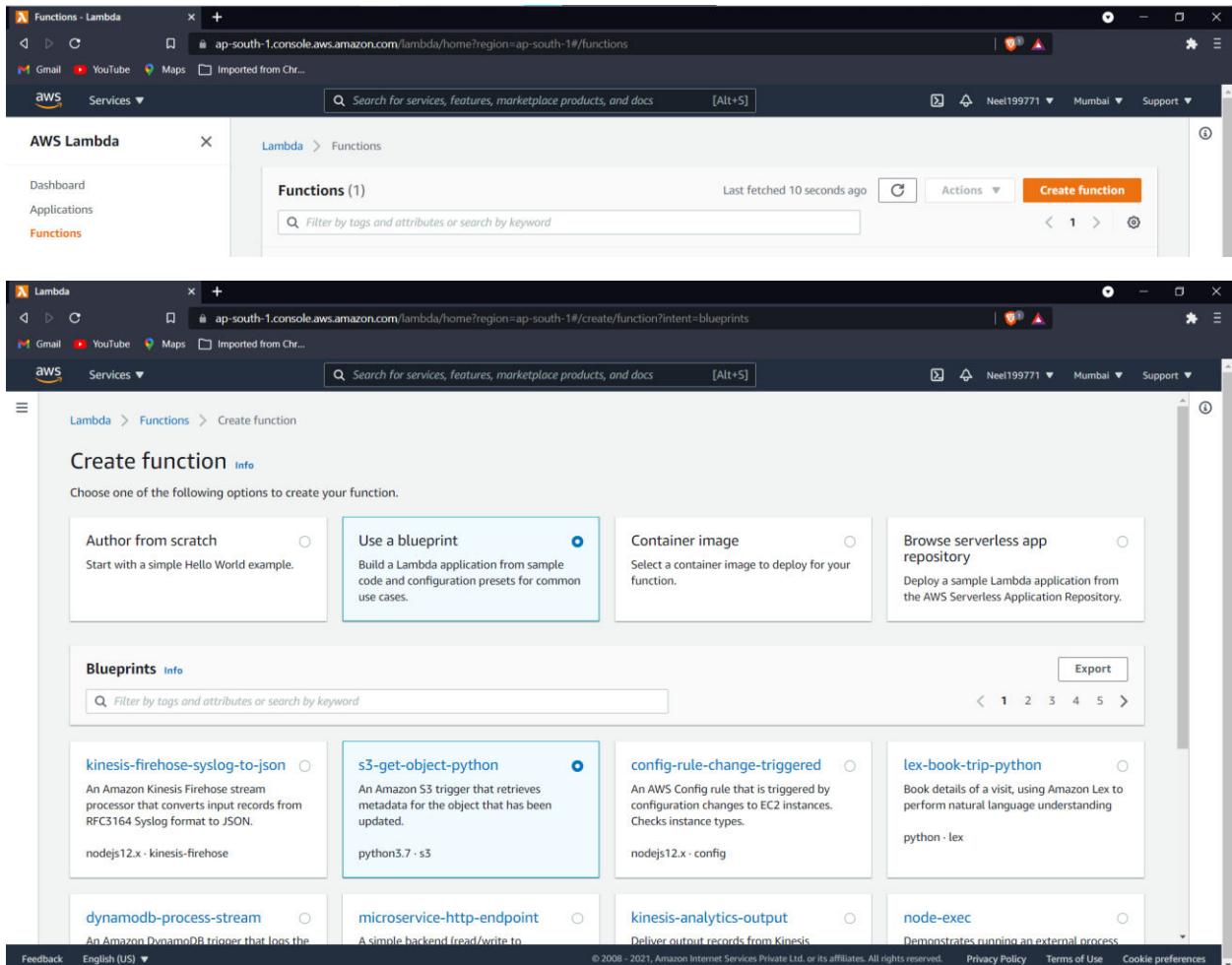
Deliver output records from Kinesis

node-exec

Demonstrates running an external process

Feedback English (US)

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



Lambda

Services

Lambda > Functions > Create function > Configure blueprint s3-get-object-python

Basic information

Function name

Neelt21

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name

Enter a name for your new role.

**FILL IT**

Use only letters, numbers, hyphens, or underscores with no spaces.

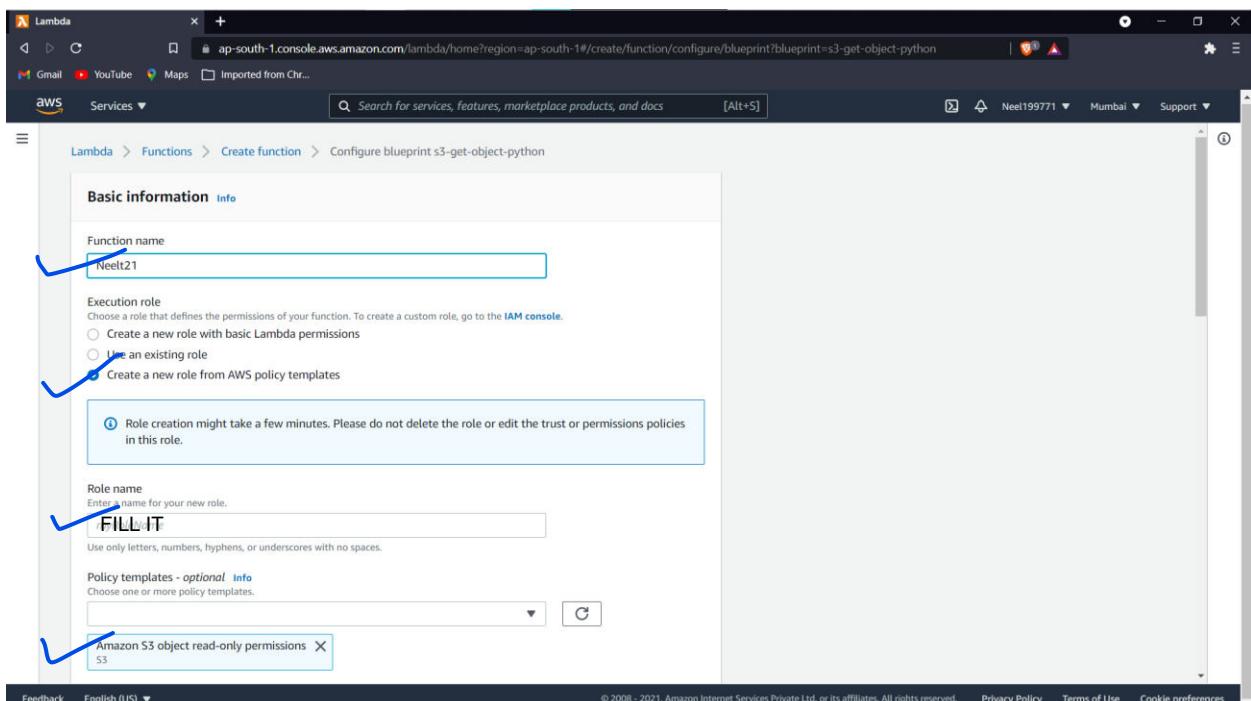
Policy templates - optional

Choose one or more policy templates.

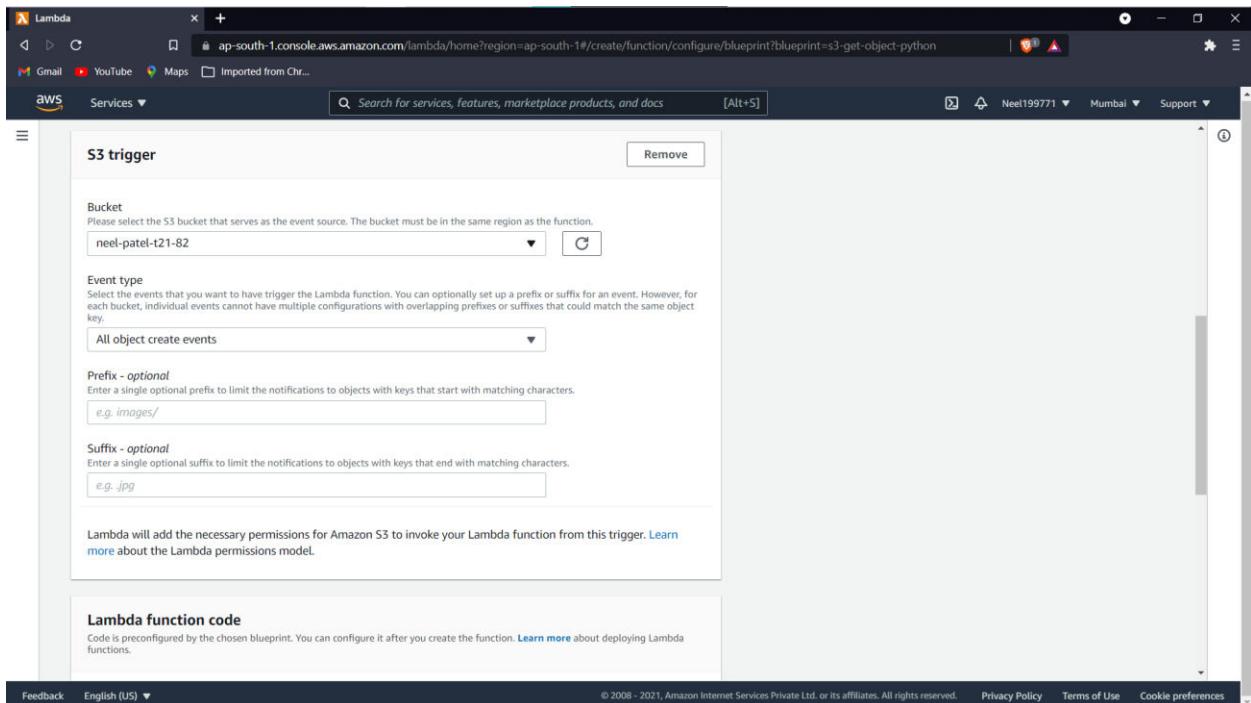
Amazon S3 object read-only permissions

Feedback English (US)

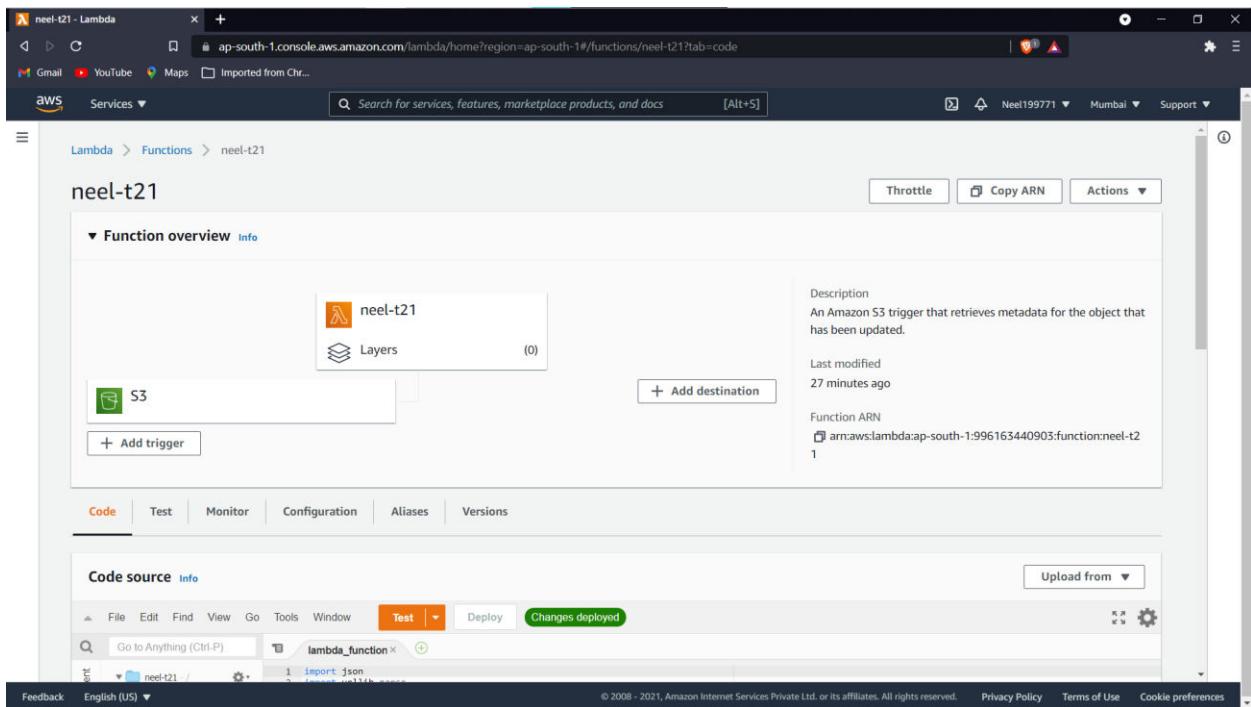
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences



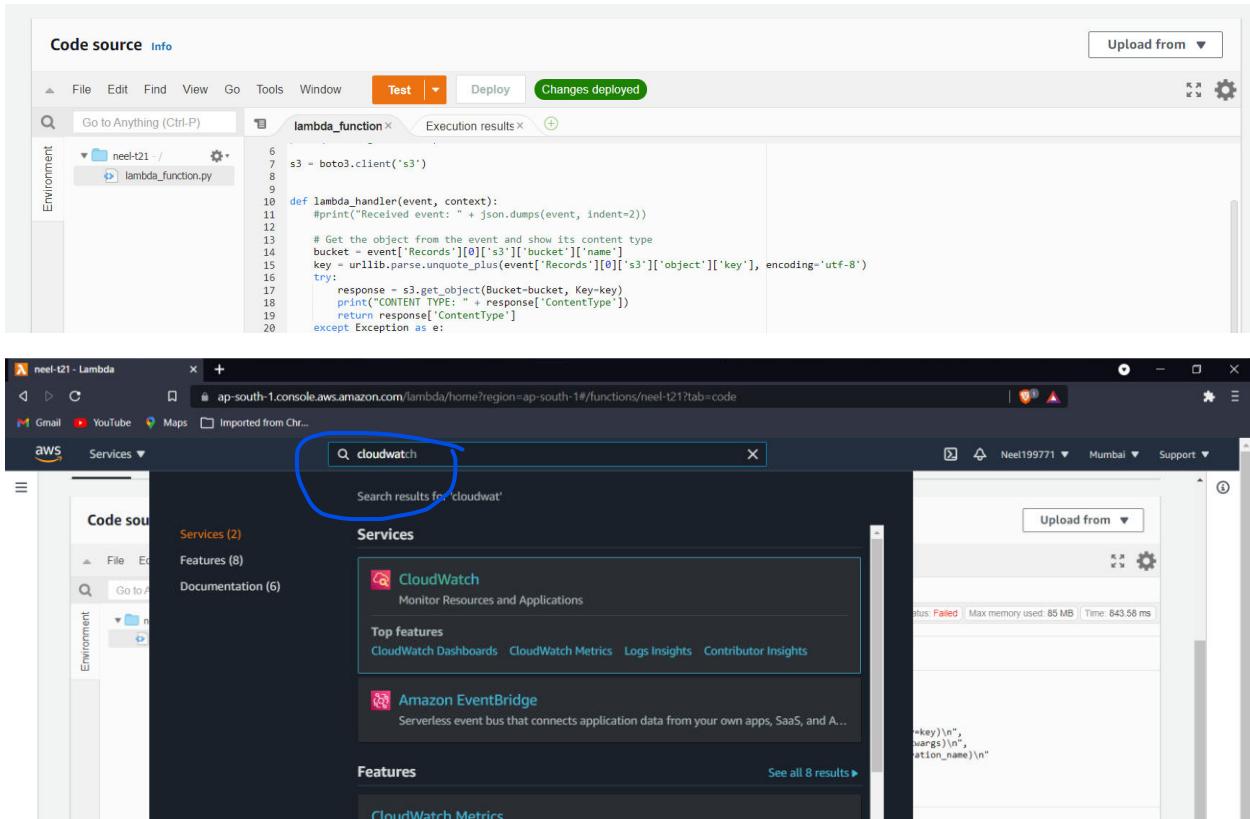
## 7)Select the bucket created and create trigger ,click on create function



## 8) Check the given trigger is created



## 9) Click on the orange test button



## 10) Check the logs , “an image has been added” message will be printed in logs

CloudWatch Management Console

CloudWatch > Log groups > /aws/lambda/neel-t21

Actions View in Logs Insights Search log group

Log group details

Retention	Creation time	Stored bytes	ARN
Never expire	21 minutes ago	-	arn:aws:logs:ap-south-1:996163440903:log-group:/aws/lambda/neel-t21:1
KMS key ID	Metric filters	Subscription filters	Contributor Insights rules
-	0	0	-

Log streams (2)

Log stream	Last event time
2021-09-07/[\$LATEST]je51215ab14be44f8555e7bff287da1d	2021-09-07 15:47:48 (UTC+05:30)
2021-09-07/[\$LATEST]842026ddeba34f8baea274d87a7b9793	2021-09-07 15:32:47 (UTC+05:30)

## 11) terminate by clicking on delete function

neel-t21 - Lambda

Lambda > Functions > neel-t21

neel-t21

Function overview

Actions

- Throttle
- Copy ARN
- Actions
- Publish new version
- Create alias
- Export function
- Delete function

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

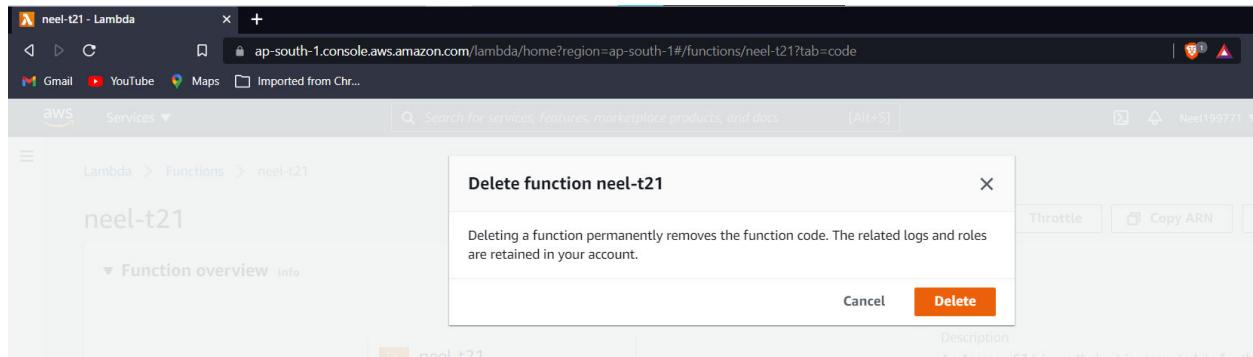
Code source

Code source

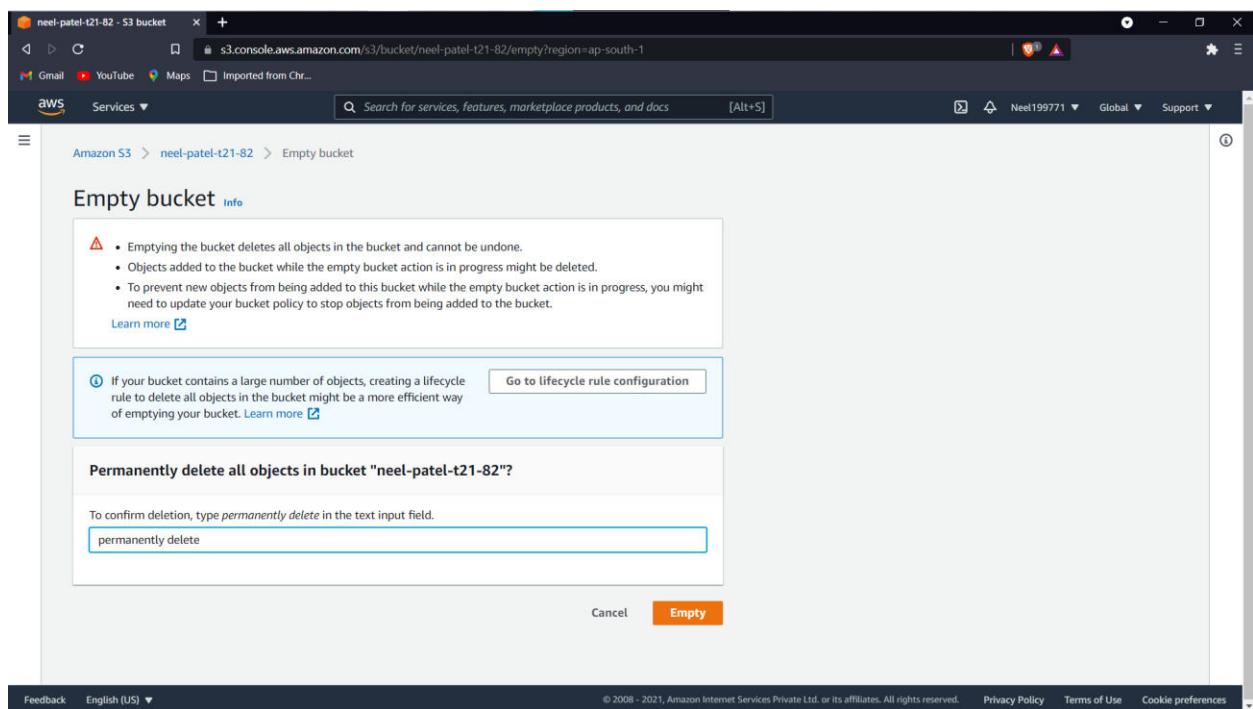
File Edit Find View Go Tools Window Test Deploy Changes deployed

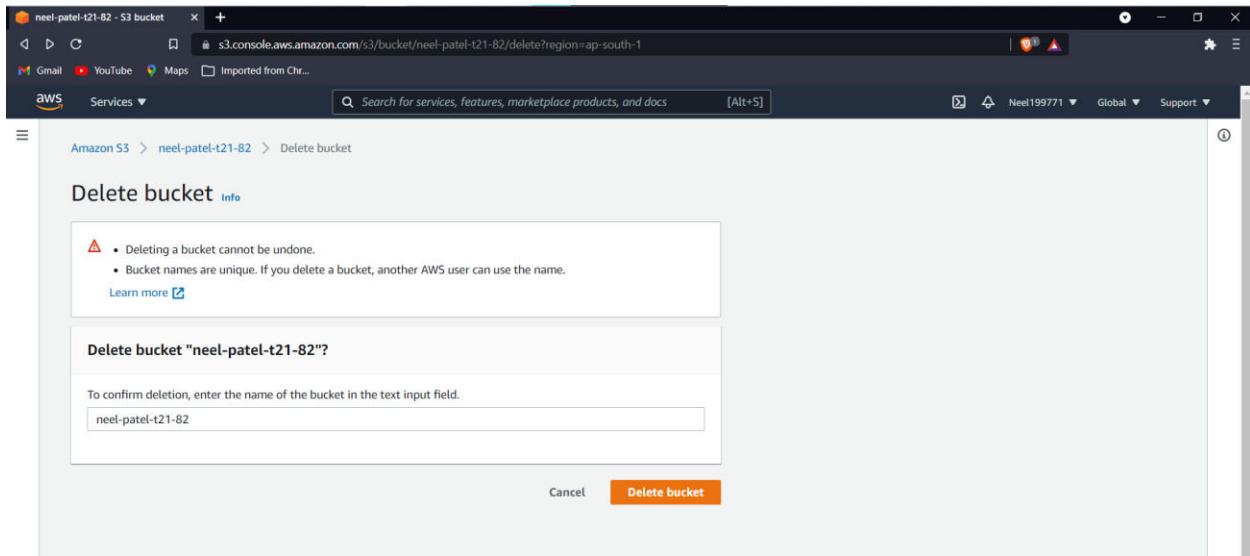
Execution results

Feedback English (US)



## 12) Empty and delete the bucket





## Conclusion:-

Learnt about AWS Lambda function their applications in software industry and created a Lambda function using python to add image in a S3 bucket and verified if image is added or not by printing message after message is added

**Name :-Niranjan Rajesh Joshi**  
**Roll No:- 2105051**  
**Batch:- T13**  
**Date Of Performance :- 05/09/2023**

## **Experiment 10**

### **Aim:-**

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

### **Theory:-**

#### **DYNAMO DB**

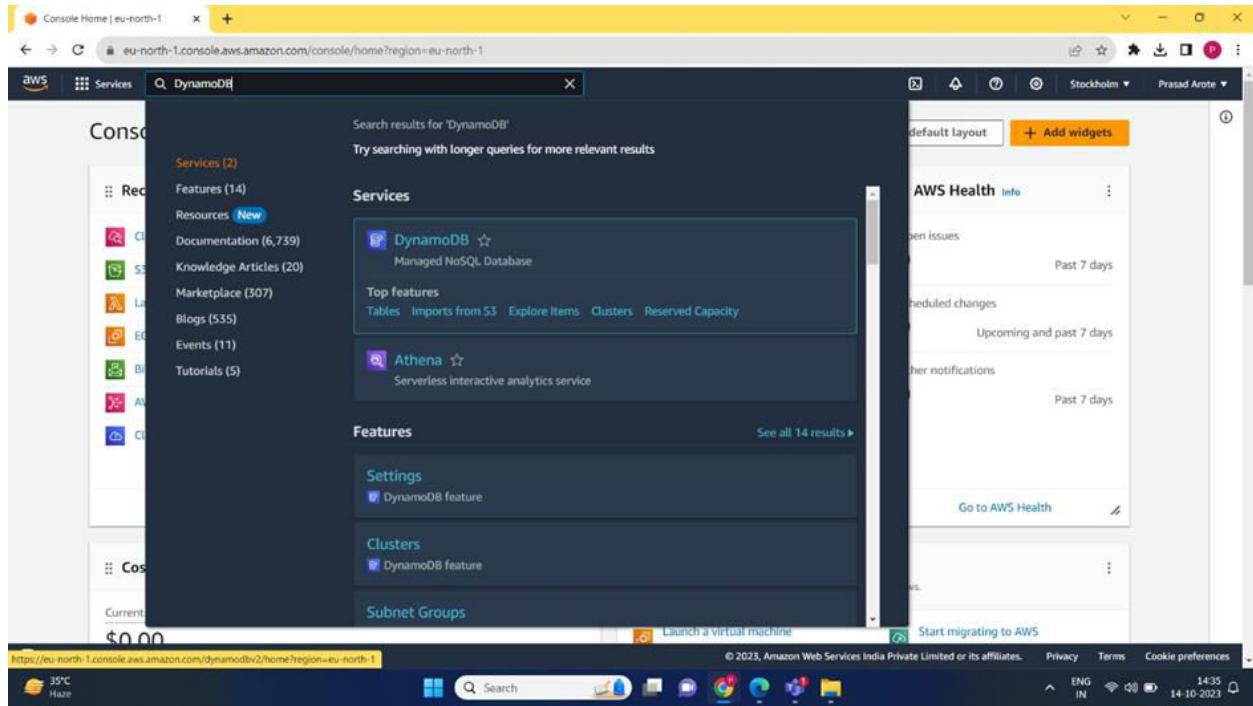
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and 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

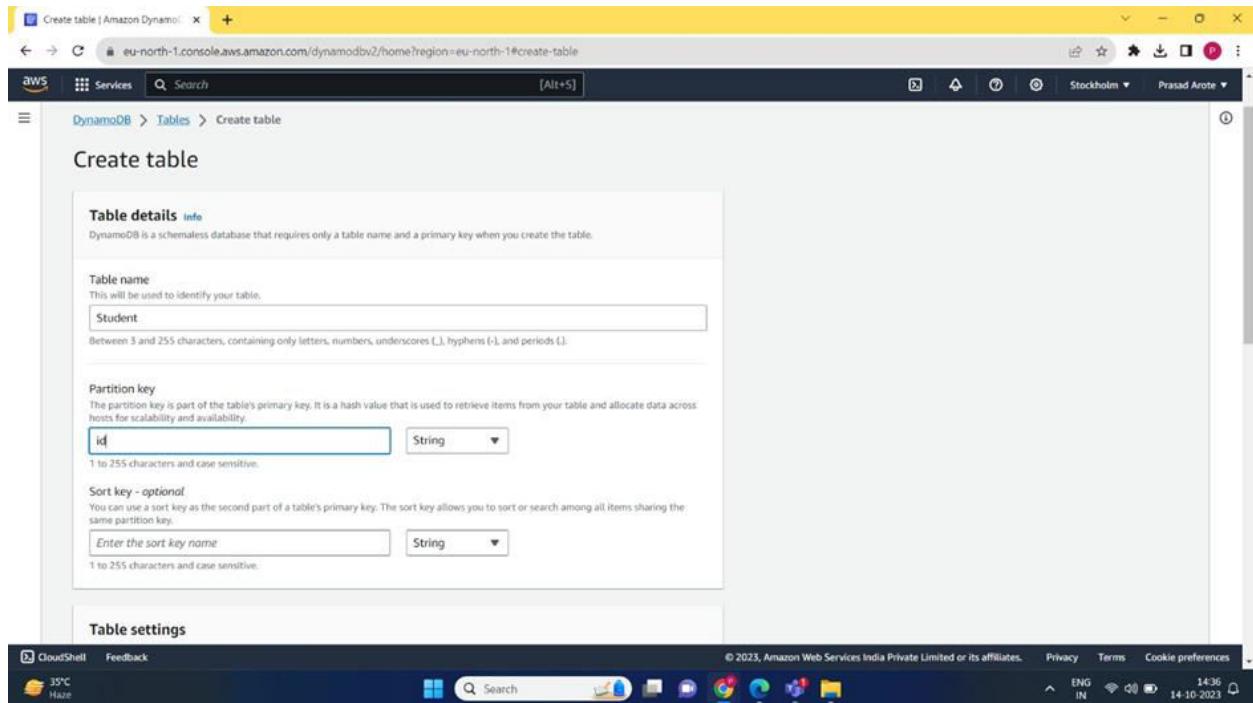
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

### **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



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

Screenshot of the AWS IAM 'Create role' wizard, Step 1: Select trusted entity.

**Select trusted entity**

**Trusted entity type**

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

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

**Service or use case**  
Lambda

**Choose a use case for the specified service.**

Screenshot of the AWS IAM 'Create role' wizard, Step 2: Add permissions.

**Add permissions**

**Permissions policies (1/887)**

Choose one or more policies to attach to your new role.

Filter by Type: All types, 4 matches

Policy name	Type	Description
<input checked="" type="checkbox"/>  AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB...
<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...

**Set permissions boundary - optional**

Cancel Previous Next

Role name: prasad\_admin

Description: 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      "service:lambda.amazonaws.com"
11    ]
12  }

```

Identity and Access Management (IAM) - Roles (9) info

Role name	Trusted entities	Last activity
AWSCloud9SSMAccessRole	AWS Service: ec2, and 1 more	75 days ago
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application	-
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked)	75 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
lambdafunc1-role-11c5j6u	AWS Service: lambda	1 hour ago
prasad_admin	AWS Service: lambda	-
PyRole	AWS Service: lambda	68 days ago
Runpython	AWS Service: lambda	68 days ago

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

Screenshot of the AWS Lambda console search results and a detailed view of the 'Create function' wizard.

**Search Results (Top):**

- Services** (7):
  - Features (3): Documentation (9,951), Knowledge Articles (20), Marketplace (664)
  - Resources (New):
    - Documentation (9,951)
    - CloudWatch Metrics (20)
    - Logs (1,017)
    - Events (13)
    - Tutorials (6)
- Lambda**: Run code without thinking about servers.
- CodeBuild**: Build and Test Code.
- AWS Signer**: Ensuring trust and integrity of your code.
- Amazon Inspector**: Continual vulnerability management at scale.

**Create function (Bottom):**

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**: addstudentdata

**Runtime**: Python 3.9

**Architecture**: x86\_64

**Permissions**: (Info)

The screenshots show the 'Create function' wizard in the AWS Lambda console. The user is on the 'Permissions' step. In the first two screenshots, a blue arrow points to the 'Use an existing role' radio button and the 'prasad\_admin' dropdown. In the third screenshot, a blue arrow points to the 'Advanced settings' section.

**Function name:** addstudentdata

**Runtime:** Python 3.9

**Architecture:** x86\_64

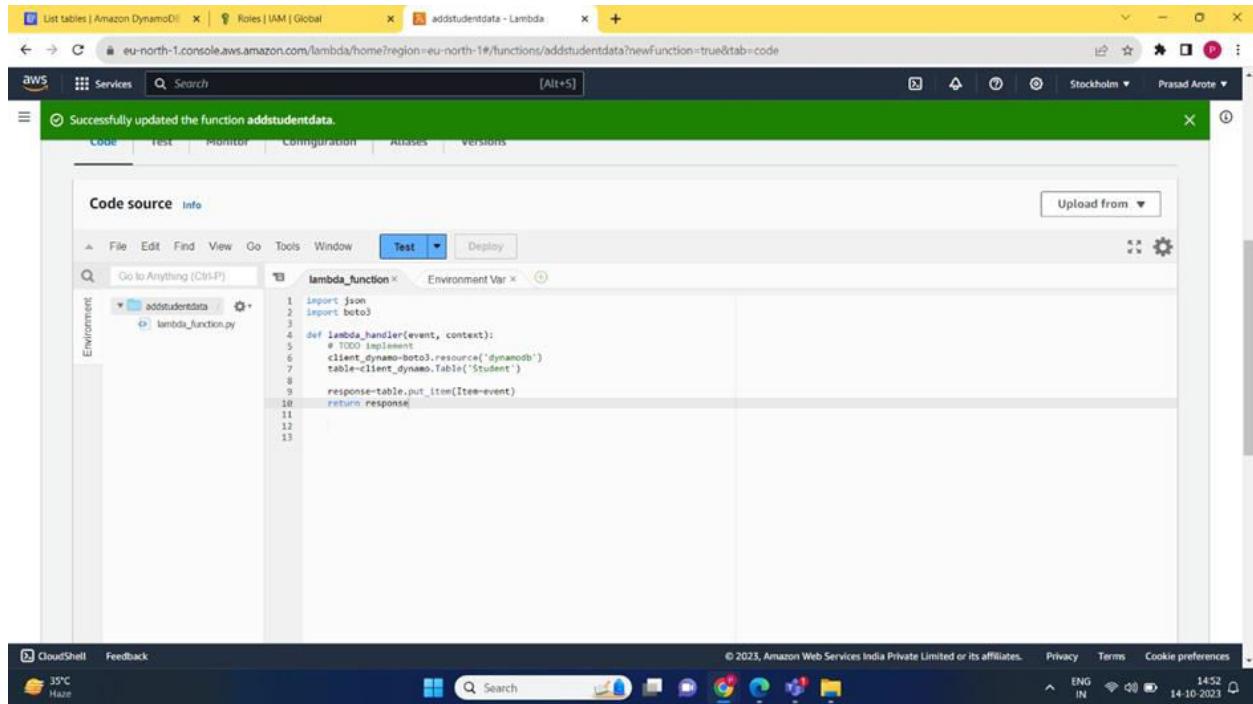
**Permissions:**

- Execution role:** Use an existing role (radio button selected)
- Existing role:** prasad\_admin

**Advanced settings:**

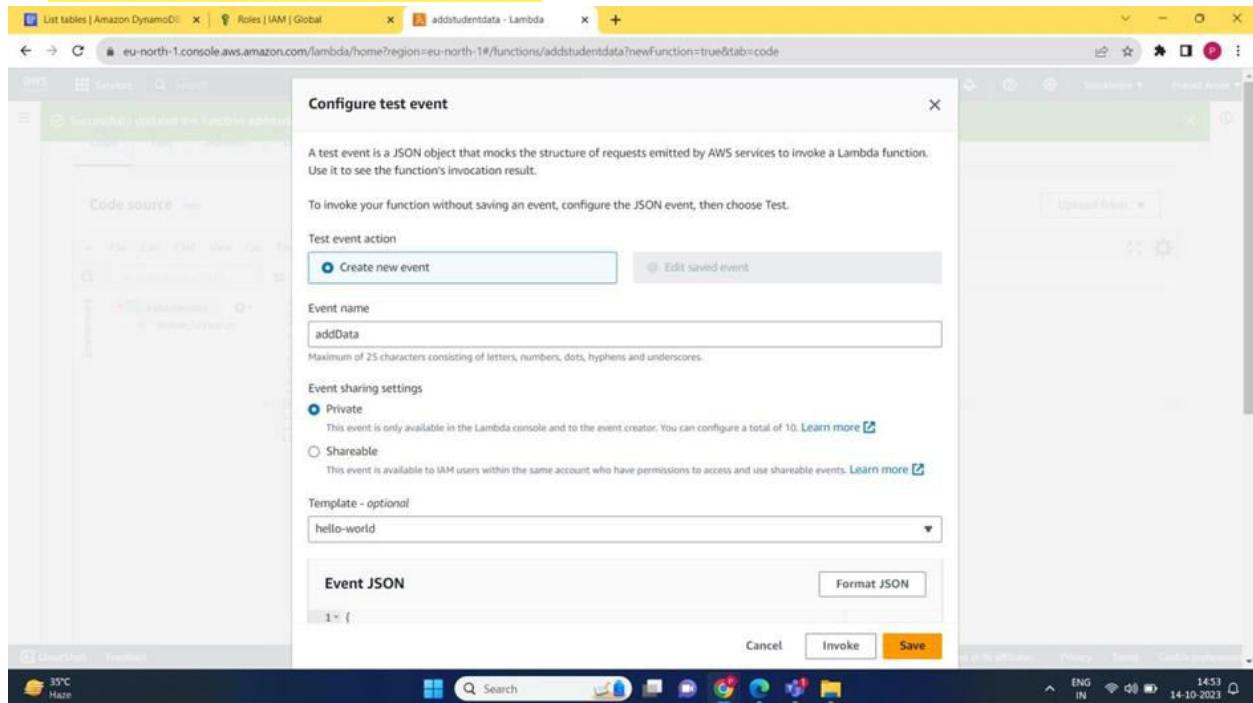
- Enable Code signing
- Enable function URL
- Enable tags
- Enable VPC

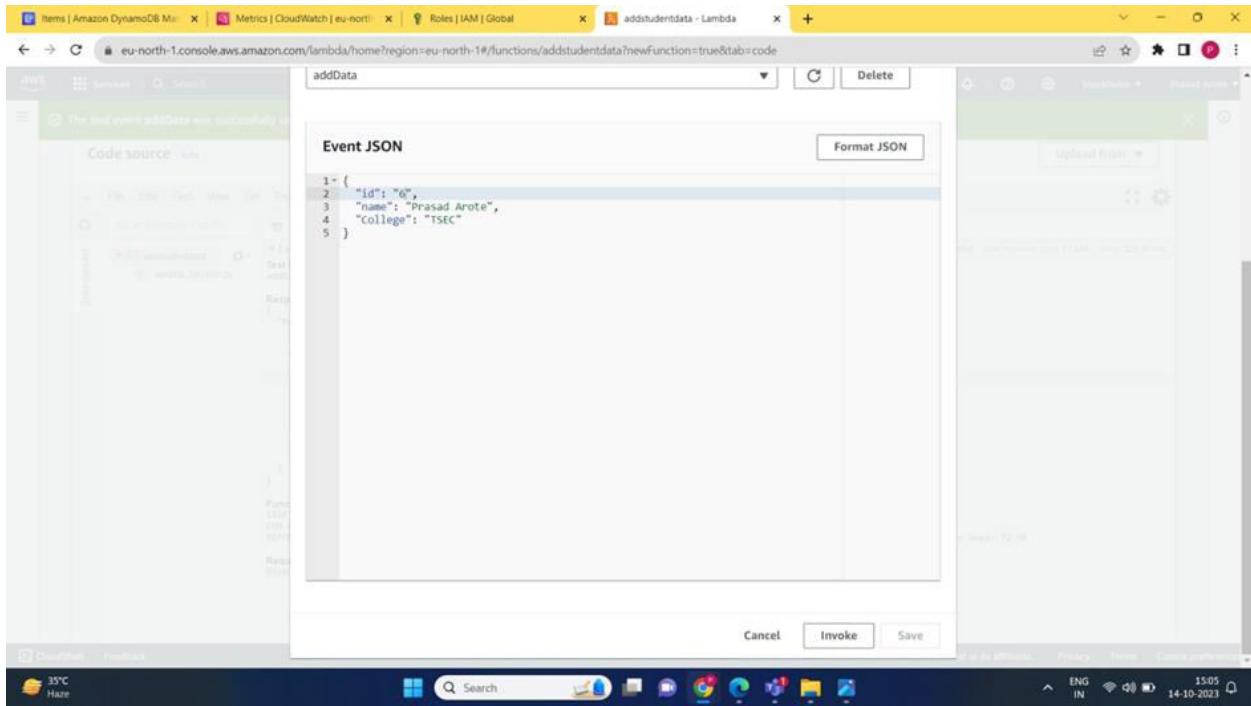
## 5) Write the following code in code source



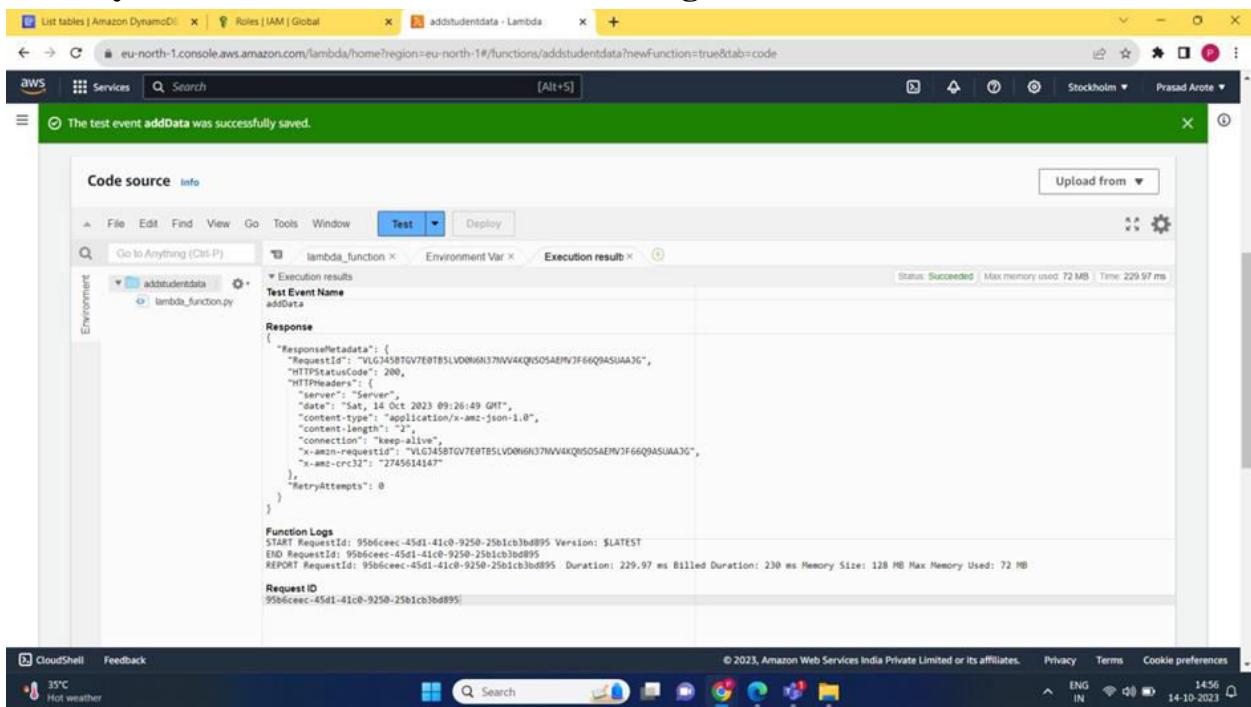
```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # TODO implement
6     client_dynamo=boto3.resource('dynamodb')
7     table=client_dynamo.Table('Student')
8
9     response=table.put_item(Item=event)
10
11     return response
12
13
```

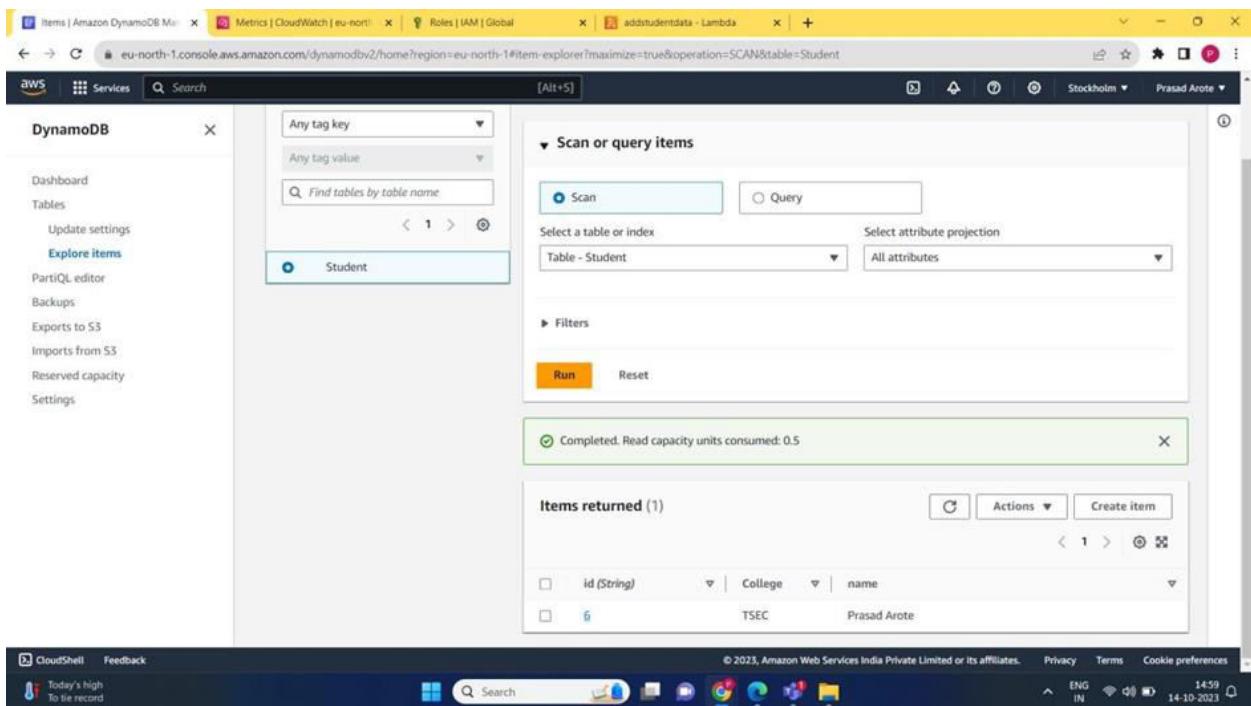
## 6) Configure the test event and save





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





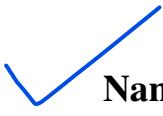
## Conclusion:-

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

Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure. Traditionally, to run any application on your computer, you had to install the version that matched your machine's operating system

Orchestration is the coordination and management of multiple computer systems, applications and/or services, stringing together multiple tasks in order to execute a larger workflow or process. These processes can consist of multiple tasks that are automated and can involve multiple systems.

The goal of orchestration is to streamline and optimize the execution of frequent, repeatable processes and thus to help data teams more easily manage complex tasks and workflows. Anytime a process is repeatable, and its tasks can be automated, orchestration can be used to save time, increase efficiency, and eliminate redundancies. For example, you can simplify data and machine learning with jobs orchestration.

 **Name :- Niranjan Rajesh Joshi**

**RollNo :- 2105051**

**Batch :- T13**

**Date Of Performance :- 01/08/2023**

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

### **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.

**Name :- Niranjan Rajesh Joshi**

**RollNo :- 2105051**

**Batch :- T13**

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



## **Written Assignment 2**

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

To deploy a Lambda function on AWS, user can follow these general steps:

#### **1) Create a Lambda Function:**

- a) Log in to the AWS Management Console.
- b) Open the Lambda console.
- c) Click on "Create function."
- d) Choose the method for creating the function (Author from scratch, Use a blueprint, or Browse serverless app repository).
- e) Configure the function details such as the function name, runtime, and permissions.

#### **2) Write and Upload Code:**

- a) Write your function code in the selected runtime environment.
- b) Upload the code either directly in the Lambda console or through an S3 bucket.

#### **3) Configure Function:**

- a) Set up the function's environment variables, execution role, memory allocation, timeout, and other settings.

#### **4) Configure Triggers:**

- a) You can configure triggers for your Lambda function such as API Gateway, S3, DynamoDB, etc., depending on the use case.

#### **5) Test the Function:**

- a) Use the test events provided by Lambda or create your own custom test event to verify the function's behavior.

#### **6) Deploy the Function:**

- a) Once the function is configured and tested, you can deploy it by clicking the "Deploy" button in the Lambda console.

## 7) **Monitor the Function:**

- a) Monitor the function's performance and any potential errors using AWS CloudWatch logs and metrics.

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

AWS Lambda supports various deployment options that cater to different use cases and development preferences. Some of the key deployment options for AWS Lambda include:

**AWS Management Console:** Users can create, configure, and deploy Lambda functions directly through the AWS Management Console. This web-based interface allows users to set up and manage Lambda functions without the need for any additional tools or services.

**AWS Command Line Interface (CLI):** The AWS CLI provides a command-line interface for managing AWS services, including Lambda. Users can use the AWS CLI to create, deploy, update, and manage Lambda functions, which is particularly useful for automating deployment processes and integrating Lambda into scripts and workflows.

**AWS Software Development Kits (SDKs):** AWS SDKs are available for multiple programming languages, allowing users to integrate Lambda into their applications directly. Using the SDKs, users can deploy Lambda functions programmatically, providing flexibility and customization for the deployment process.

**AWS CloudFormation:** AWS CloudFormation enables users to define and deploy infrastructure as code. Users can use CloudFormation templates to provision and manage Lambda functions along with other AWS resources in a declarative manner, facilitating consistent and repeatable deployments.

**AWS Serverless Application Model (SAM):** SAM is an open-source framework that extends CloudFormation to simplify the deployment of serverless applications. It provides a simplified syntax for defining serverless applications, including Lambda functions, APIs, and other resources, making it easier to deploy complex serverless applications.

**Integrated Development Environments (IDEs):** Several IDEs, such as AWS Toolkit for Visual Studio, AWS Toolkit for PyCharm, and AWS Toolkit for IntelliJ, offer tools and features for developing, debugging, and deploying Lambda functions directly from the IDE environment.

These deployment options provide developers with various tools and methods to create, manage, and deploy Lambda functions on AWS, catering to diverse development workflows and preferences.

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

AWS provides three main deployment modes that users can leverage for their applications:

#### 1) **EC2 Deployment:**

**Description:** EC2 (Elastic Compute Cloud) is a web service that provides resizable compute capacity in the cloud. It allows users to launch virtual servers (EC2 instances) and provides full control over the operating system, network, and other aspects of the infrastructure.

**Use Case:** EC2 deployment is suitable for users who require complete control and customization over their computing resources. It is ideal for applications that demand a high level of configurability and flexibility.

#### 2) **Elastic Beanstalk Deployment:**

**Description:** AWS Elastic Beanstalk is a platform as a service (PaaS) that simplifies the process of deploying and scaling web applications and services. It automatically handles the details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

**Use Case:** Elastic Beanstalk is well-suited for developers who want to deploy applications without managing the underlying infrastructure. It is an excellent choice for web developers who want to focus on writing code rather than managing the infrastructure.

#### 3) **Serverless Deployment with AWS Lambda:**

**Description:** AWS Lambda is a serverless computing service that allows users to run code without provisioning or managing servers. It automatically scales applications by running code in response to specific events or triggers.

**Use Case:** Lambda is suitable for building serverless applications and for executing code in response to various events, such as changes to data in an Amazon S3 bucket, DynamoDB table updates, or HTTP requests via Amazon API Gateway. It is particularly beneficial for event-driven and microservices architectures where the focus is on individual functions rather than entire applications.

Each deployment mode provides distinct advantages and is designed to meet different application requirements, allowing users to choose the best fit for their specific use case and development needs.

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

Three main components of AWS Lambda include:

### **Lambda Function:**

The Lambda function is the heart of the AWS Lambda service. It is the code that runs in response to events. Users can upload their code or write it directly in the AWS Management Console, using one of the supported programming languages such as Node.js, Python, Java, Go, and others. The function code is designed to be stateless and can be triggered by a variety of events or requests. Users can define the function's configuration, including memory size, timeouts, and environment variables.

### **Event Sources:**

Event sources are the triggers that invoke the Lambda function. AWS Lambda can be triggered by various AWS services such as Amazon S3, Amazon DynamoDB, Amazon Kinesis, Amazon SQS, and others. Additionally, it can be integrated with custom event sources through Amazon API Gateway or other AWS services. These event sources enable users to create custom event-driven architectures that respond to specific actions or changes within their AWS environment.

### **Lambda Execution Environment:**

The Lambda execution environment consists of the infrastructure and resources necessary to run the Lambda function. AWS manages the execution environment and automatically provisions and scales the infrastructure based on the incoming request volume. It includes the compute resources, networking, and security configurations needed to execute the function code securely and reliably. AWS Lambda handles the complexities of infrastructure management, allowing users to focus solely on writing the function code and defining the event sources.