

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

**CERTIFICATE**

Certify that Mr./Miss Altuf Alam  
of I7 Department, Semester IV with  
Roll No. 02 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 20 23 - 20 24

Teacher In-Charge

Head of the Department

Date 20/10/23

Principal

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	To understand the benefit of cloud infrastructure and setup Aws cloud9.	17/7/23		7
2.	To build your application using Aws		24/7/23	
codebuild and Deploy on S3.				
3.	Creating on S3 bucket using AWS	3,17/23		
Service and uploading a file into it.				
4.	To understand the installation		7/8/23	
process of Terraform				
5.	To understand the concept of terraform	21/8/23		
and use it to create and instance				
using limit , plan , apply &				
destroy command				4
6.	To understand the concept of	28/8/23		
Sonar cube & server scanner				
7.	To learn how to use lambda	4/09/23		
to run a simple program from S3 bucket				
8.	To learn how to use lambda to	11/09/23		
find the content type of object				
uploaded in S3 bucket.				
9.	To understand and install	18/09/23		
Nagios monitor host using nagios				
10.	To complete study of	28/10/23		
Kubernetes				
11.	Theory Assignment 1			
12.	Theory Assignment 2			

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

## Assignment No 1

**AIM** - To make a EC2 Machine in AWS.

### **THEORY** -

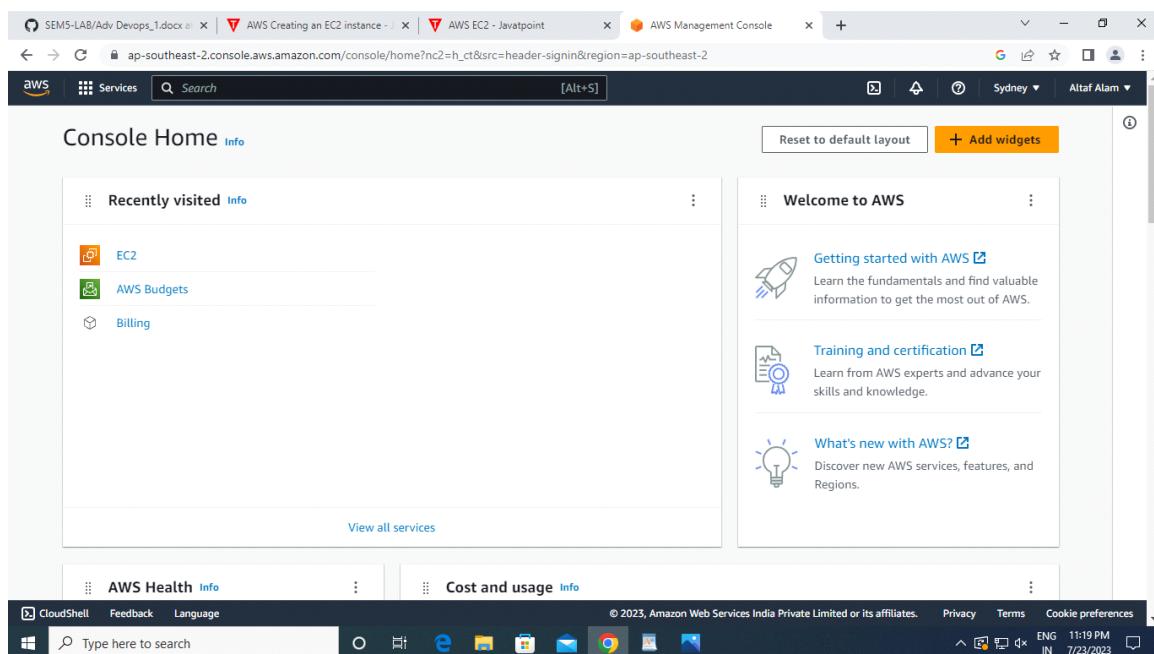
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time required to obtain and boot new user instances to minutes rather than in older days, if you need a server then you had to put a purchase order, and cabling is done to get a new server which is a very time-consuming process.

### **STEPS** -

LOGIN TO AWS

ACCOUNT, THEN SEACH

EC2.



Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

NOW CLICK ON LAUNCH / CREATE NEW INSTANCES.

The screenshot shows the AWS EC2 Management Console in the Sydney region. The left sidebar includes options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main 'Resources' section displays various Amazon EC2 metrics: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. A callout box highlights the 'Launch instance' button. The right sidebar shows account attributes such as Supported platforms (VPC), Default VPC (vpc-057865ca5797d5990), and Settings (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments). The bottom right corner shows the date and time as 7/23/2023, 10:46 PM.

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

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the name 'Altaf' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Ubuntu' AMI is selected. A callout box provides information about the Free tier: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' The final step shows a summary with 1 instance and 1 volume (8 GiB) before reaching the 'Launch instance' button.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

Click on T2 micro (free tier one)

The screenshot shows the AWS EC2 Instance Types comparison page. The 't2.micro' instance is selected and highlighted with a blue border. The table provides detailed specifications for each instance type:

Instance type	vCPUs	Architecture	Memory (GiB)	Storage (GB)	Storage type	Network performance
t1.micro	1	i386, x86_64	0.612	-	-	Very Low
t2.nano	1	i386, x86_64	0.5	-	-	Low to Moderate
<b>t2.micro</b>	<b>1</b>	<b>i386, x86_64</b>	<b>1</b>	<b>-</b>	<b>-</b>	<b>Low to Moderate</b>
t2.small	1	i386, x86_64	2	-	-	Low to Moderate
t2.medium	2	i386, x86_64	4	-	-	Low to Moderate
t2.large	2	x86_64	8	-	-	Low to Moderate

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

Now add security group ALL TRAFFIC ,

PROTOCOL – ALL,

SOURCE- ANYWHERE.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

The screenshot shows the AWS EC2 Launch Instance wizard. In the 'Network settings' section, it shows a VPC (vpc-057865ca5797d5990) and a subnet (No preference). Under 'Firewall (security groups)', there is a note about creating a new security group named 'launch-wizard-1'. It lists three rules: 'Allow SSH traffic from Anywhere', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. The 'Launch instance' button is highlighted.

## Configure Storage

The screenshot shows the 'Configure storage' section of the EC2 wizard. It specifies 1x 8 GiB gp2 volume for the root volume (Not encrypted). A note indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. The 'Advanced' tab is selected, showing options like purchasing options and domain join directory. A tooltip for the 'Free tier' is visible, detailing usage limits. The 'Launch instance' button is highlighted.

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

The screenshot shows the AWS EC2 'Launch an instance' success page. It displays a green checkmark icon and the message 'Successfully initiated launch of instance (i-005492fe4b082ff1f)'. Below this, there is a link to 'Launch log'. The main content area is titled 'Next Steps' with a search bar. It includes four cards: 'Create billing and free tier usage alerts', 'Connect to your instance', 'Connect an RDS database', and 'Create EBS snapshot policy'. The bottom of the screen shows the Windows taskbar with various pinned icons.

Once Check is complete click on launch instances.

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances' (selected), 'Images', and 'Elastic Block Store'. The main content area shows a table of instances with two entries:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Altaf	i-0282e3d4188fbe037	Terminated	t2.micro	-	No alarms	ap-southeast-2c
Altaf	i-005492fe4b082ff1f	Running	t2.micro	Initializing	No alarms	ap-southeast-2c

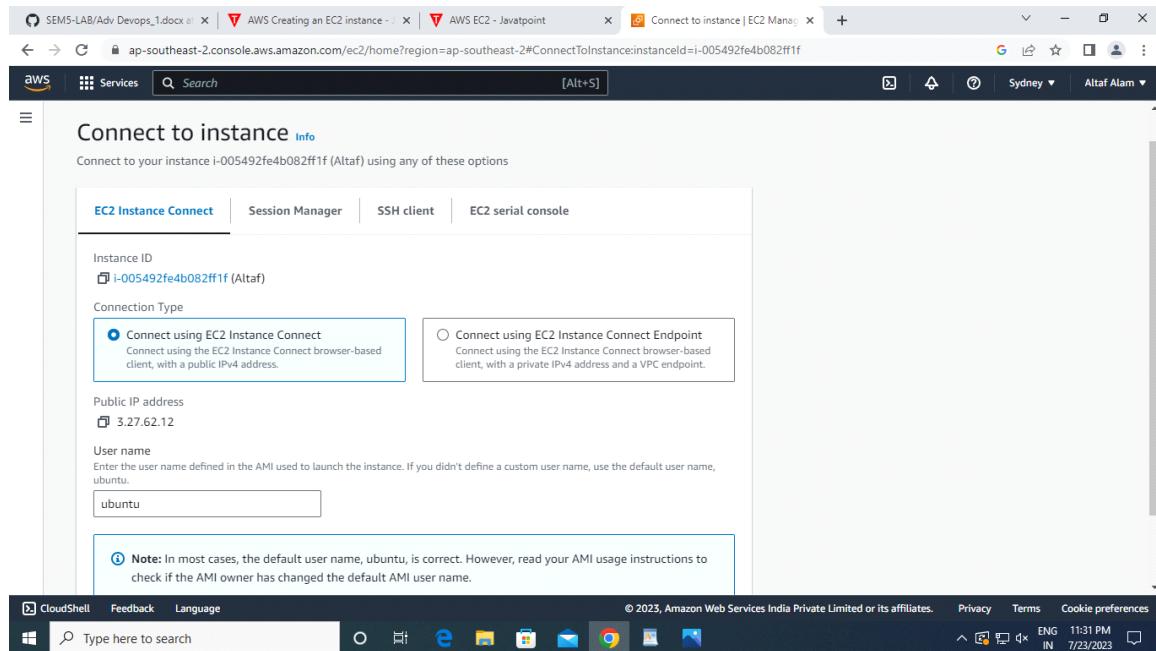
A modal window titled 'Select an instance' is open at the bottom, showing the same two instances. The bottom of the screen shows the Windows taskbar.

Connecting to Instance :

Altaf Alam , 02 , T11

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23



## FOLLOW SOME BASIC LINUX COMMANDS AS SHOWN BELOW-

A screenshot of a terminal window titled "EC2 Instance Connect" showing a terminal session on an Ubuntu system. The session starts with a message about free software distribution terms. It then shows the user running commands like "ls", "cat", and "sudo". The user types "hello" and "cat hello", then "ctrl+C" to interrupt. Finally, they run "3+4" and "122" which are not found. The terminal ends with "ubuntu@ip-172-31-23-145:~\$". Below the terminal, the instance details are shown: "i-005492fe4b082ff1f (Altaf)", "PublicIPs: 3.27.62.12", and "PrivateIPs: 172.31.23.145". The browser toolbar at the bottom includes CloudShell, Feedback, Language, and a search bar.

## CONCLUSION:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 24/07/23

Hence learned and implemented the steps of create an ec2 machine.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 31/07/23

## Experiment No 2

Aim : To create a Cloud9 Environment.

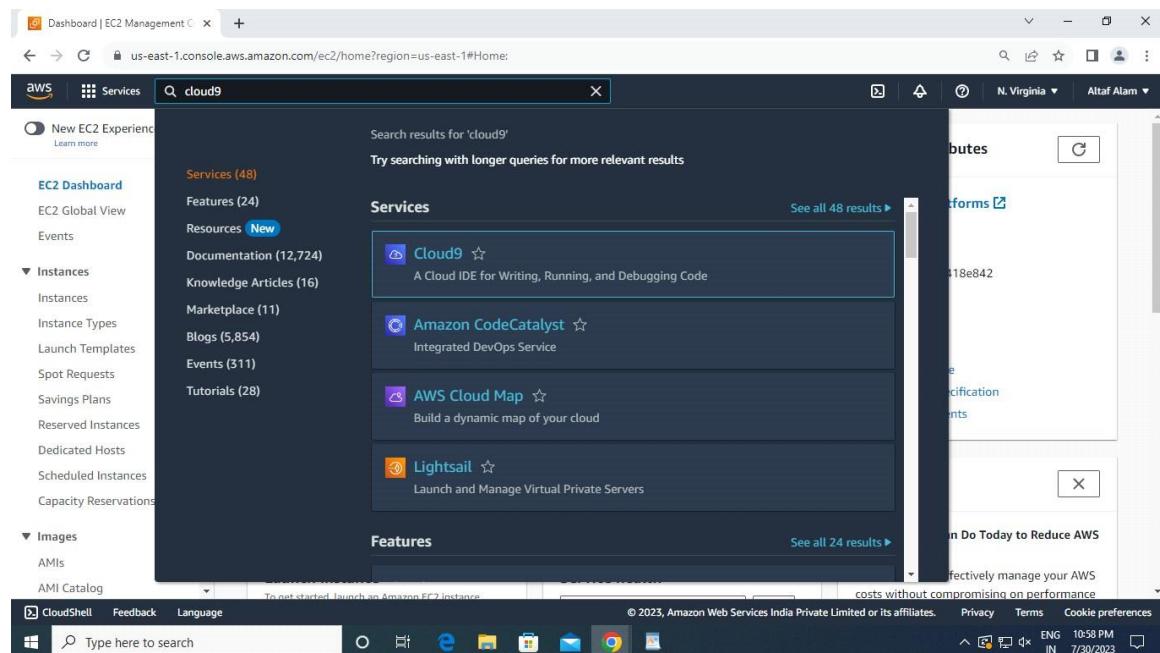
### Theory :

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

### STEPS-

LOG IN TO YOUR AWS ACCOUNT,

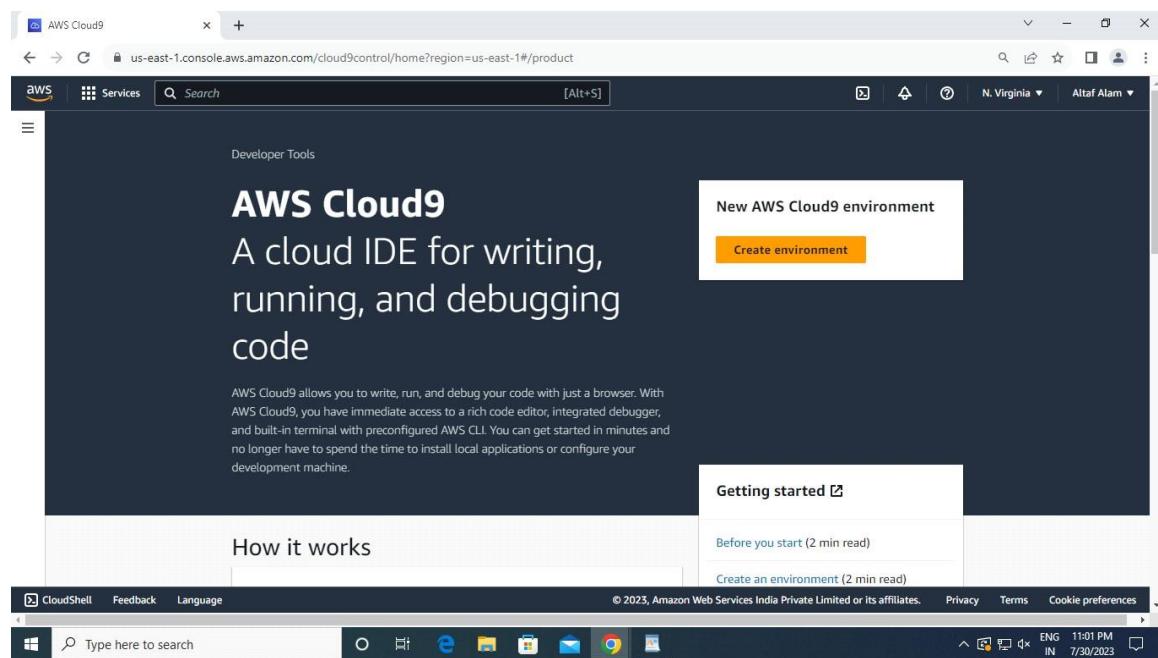
SEARCH FOR CLOUD 9 IN THE SEARCH BAR



CLICK ON CLOUD 9.

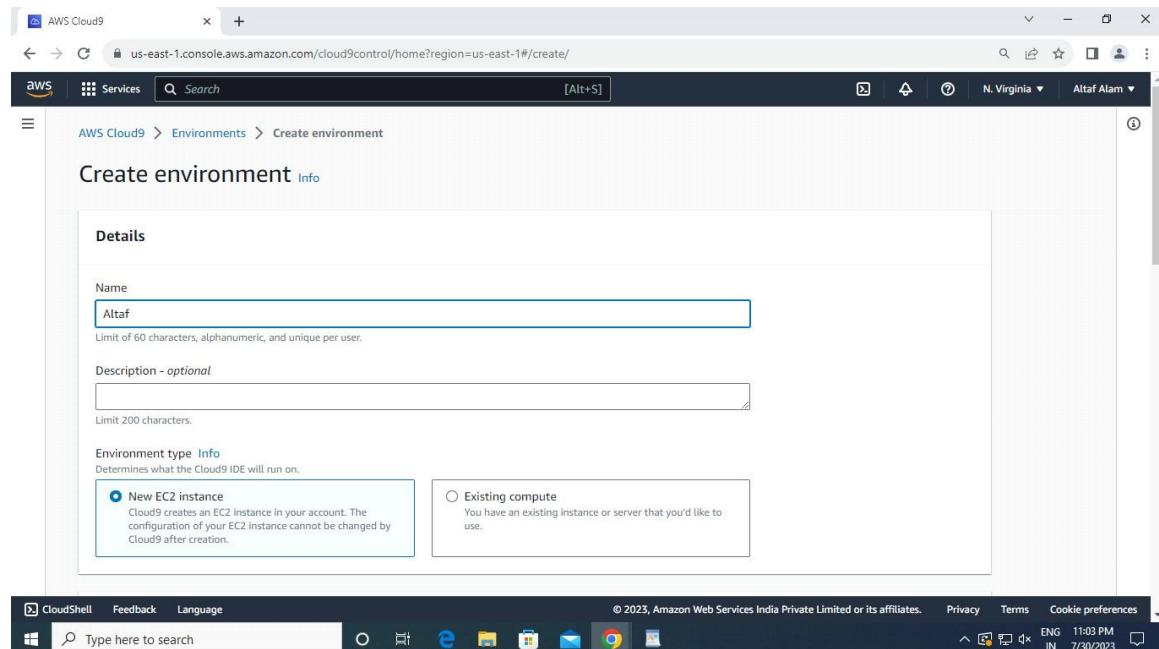
Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 31/07/23



CLICK ON CREATE ENVIRONMNET,

NAME THE ENVIRONMENT



CHOOSE FREE TIER INSTANCE.

Altaf Alam , 02 , T11

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 31/07/23

The screenshot shows the 'New EC2 instance' configuration page in the AWS Cloud9 console. It includes sections for 'Instance type', 'Platform', 'Timeout', and 'Network settings'. The 'Instance type' section lists options like t2.micro, t3.small, m5.large, and additional types. The 'Platform' section shows 'Amazon Linux 2' selected. The 'Timeout' section shows '30 minutes' selected. The 'Network settings' section is partially visible.

CLICK ON CREATE BUTTON .

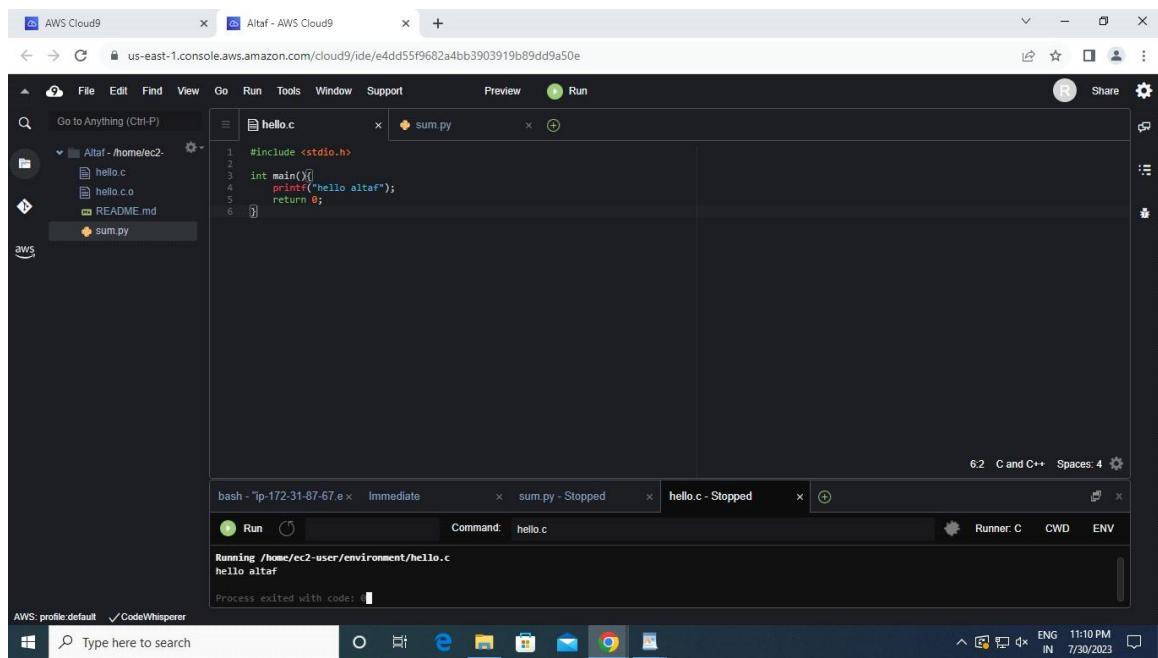
The screenshot shows the 'Create' confirmation page after clicking the 'Create' button. It displays a summary of the resources being created, including AWS Systems Manager (SSM) and Secure Shell (SSH). It also lists optional tags and a note about IAM roles. At the bottom, there are 'Cancel' and 'Create' buttons, with 'Create' being highlighted.

CLOUD9 CREATED. CLICK ON OPEN OPTION.

Altaf Alam , 02 , T11

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 31/07/23

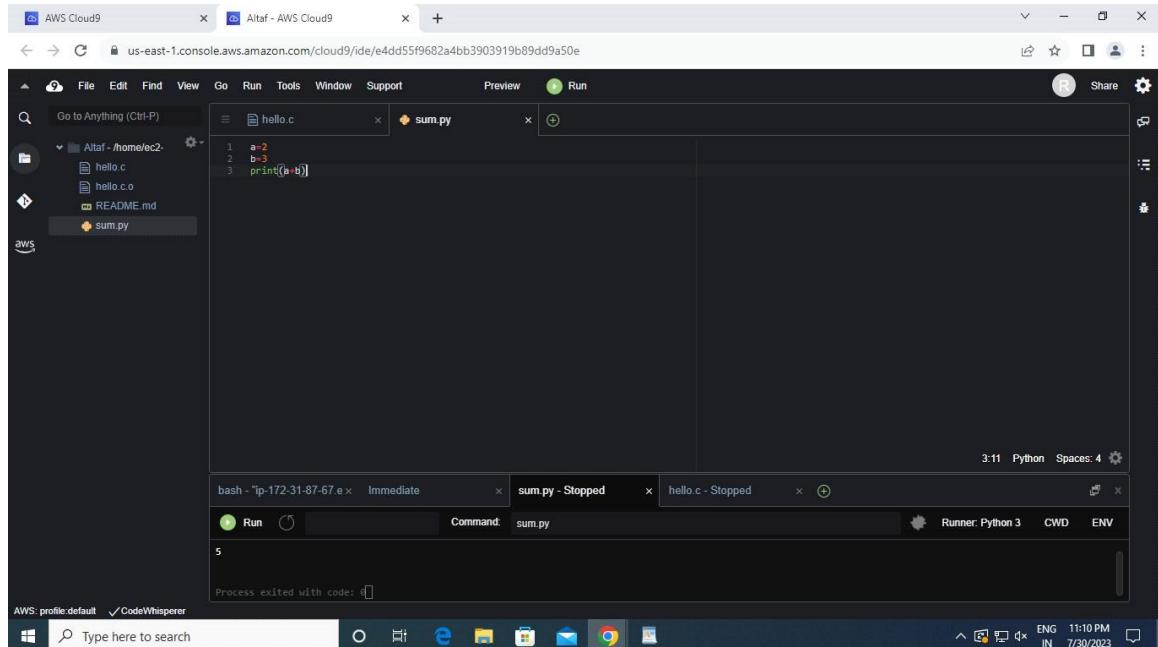


The screenshot shows the AWS Cloud9 IDE interface. In the top navigation bar, there are tabs for 'AWS Cloud9' and 'Altaf - AWS Cloud9'. The URL is 'us-east-1.console.aws.amazon.com/cloud9/ide/e4dd55f9682a4bb3903919b89dd9a50e'. The main workspace displays two code files: 'hello.c' and 'sum.py'. The 'hello.c' file contains the following C code:

```
#include <stdio.h>
int main(){
    printf("Hello Altaf");
    return 0;
}
```

The 'Run' tab at the bottom indicates the command is 'hello.c' and the runner is 'C'. The output window shows the program's output: 'Hello Altaf'. The status bar at the bottom right shows the date and time as '7/30/2023 11:10 PM'.

NOW YOU CAN SELECT ANY CODING LANGUAGE WRITE CODE AND RUN THE PROGRAM ON CLOUD.



The screenshot shows the AWS Cloud9 IDE interface. In the top navigation bar, there are tabs for 'AWS Cloud9' and 'Altaf - AWS Cloud9'. The URL is 'us-east-1.console.aws.amazon.com/cloud9/ide/e4dd55f9682a4bb3903919b89dd9a50e'. The main workspace displays two code files: 'hello.c' and 'sum.py'. The 'sum.py' file contains the following Python code:

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

The 'Run' tab at the bottom indicates the command is 'sum.py' and the runner is 'Python 3'. The output window shows the program's output: '5'. The status bar at the bottom right shows the date and time as '7/30/2023 11:10 PM'.

## Conclusion :

Hence learned and implemented steps to Create an Cloud9 environment.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 31/07/23

## LAB ASSIGNMENT 3

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

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

### THEORY:

#### 1. Create a S3 bucket.

The screenshot shows the AWS S3 Management Console. At the top, there's a banner for 'Amazon S3' with the tagline 'Store and retrieve any amount of data from anywhere'. Below the banner, a section titled 'How it works' shows a video thumbnail for 'Introduction to Amazon S3'. To the right, there are sections for 'Create a bucket', 'Pricing', and 'Resources'. The main area is titled 'Create bucket' with a sub-section 'General configuration'. Under 'Bucket name', the input field contains 'prasad-website'. Under 'AWS Region', the dropdown is set to 'Europe (Stockholm) eu-north-1'. There's also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. At the bottom, there's an 'Object Ownership' section with two radio buttons: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'.

The screenshot shows the AWS S3 Bucket creation interface. In the top section, under 'Block all public access', several options are listed:

- Block all public access**: Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**: S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**: S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**: S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

A warning message in a yellow box states: "Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting." A checkbox below it is checked: "I acknowledge that the current settings might result in this bucket and the objects within becoming public."

In the middle section, under 'Default encryption', the 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' option is selected. It also mentions 'Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)' and provides a link to the 'Amazon S3 pricing page'.

Under 'Bucket Key', it says: "Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS." Options for 'Disable' and 'Enable' are shown, with 'Enable' selected.

A 'Create bucket' button is prominently displayed at the bottom right.

## 2. Upload the files of web application.

S3 Management Console

Services Search [Alt+S]

Amazon S3 > Buckets > prasad-website > Upload

### Upload

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

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

**Files and folders (0)**

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
No files or folders			
You have not chosen any files or folders to upload.			

**Destination**

Destination

CloudShell Feedback 29°C Haze © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 02:01 15-10-2023

S3 Management Console

Services Search [Alt+S]

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

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

**Files and folders (24 Total, 89.5 KB)**

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
Bun 1.svg	-	image/svg+xml	865.0 B
Bun 1@2x.png	-	image/png	8.6 KB
Cheese.svg	-	image/svg+xml	619.0 B
Cheese@2x.png	-	image/png	1.4 KB
Lettuce.svg	-	image/svg+xml	629.0 B
Lettuce@2x.png	-	image/png	2.4 KB
Onion.svg	-	image/svg+xml	831.0 B
Onion@2x.png	-	image/png	2.8 KB
Patty.svg	-	image/svg+xml	639.0 B
Patty@2x.png	-	image/png	3.9 KB

CloudShell Feedback 29°C Haze © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 02:09 15-10-2023

S3 Management Console

Services Search [Alt+S]

Upload succeeded

View details below.

**Upload: status**

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

**Summary**

Destination	Succeeded	Failed
s3://prasad-website	24 files, 89.5 KB (100.00%)	0 files, 0 B (0%)

**Files and folders (24 Total, 89.5 KB)**

Name	Folder	Type	Size	Status	Error
No files or folders					

CloudShell Feedback 29°C Haze © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 02:10 15-10-2023

### 3. Enable Static website hosting

The screenshot shows the AWS S3 console with the 'Edit static website hosting' page for the 'prasad-website' bucket. The 'Static website hosting' section is active, with 'Enable' selected for both the 'Static website hosting' and 'Hosting type' options. Under 'Index document', 'index.html' is specified. A note about public readability is present. Under 'Error document - optional', 'error.html' is specified. The bottom section, 'Redirection rules - optional', is currently empty.

This screenshot is identical to the one above, showing the 'Edit static website hosting' configuration page. The red box highlights the note: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access'.

prasad-website - S3 bucket

Successfully edited static website hosting.

Amazon S3 > Buckets > prasad-website

prasad-website [Info](#)

Objects [Properties](#) Permissions Metrics Management Access Points

**Bucket overview**

AWS Region	Amazon Resource Name (ARN)	Creation date
Europe (Stockholm) eu-north-1	arn:aws:s3:::prasad-website	October 15, 2023, 02:01:26 (UTC+05:30)

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

Bucket Versioning  
Disabled  
Multi-factor authentication (MFA) delete  
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API.

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 02:11 15-10-2023 ENG IN

## 5. Change the Bucket Policy

prasad-website - S3 bucket

s3.console.aws.amazon.com/s3/bucket/prasad-website/property/edit?region=eu-north-1

Amazon S3 > Buckets > prasad-website > Edit bucket policy

Edit bucket policy [Info](#)

**Bucket policy**

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

[Policy examples](#) [Policy generator](#)

Bucket ARN  
arn:aws:s3:::prasad-website

**Policy**

1

[Edit statement](#)

Select a statement  
Select an existing statement in the policy or add a new statement.  
+ Add new statement

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 02:11 15-10-2023 ENG IN

**AWS Policy Generator**

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

**Step 1: Select Policy Type**

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy

**Step 2: Add Statement(s)**

A statement is the formal description of a single permission. See [a description of elements that you can use in statements](#).

Effect  Allow  Deny

Principal

AWS Service   All Services (\*)

Actions   All Actions (\*)

Amazon Resource Name (ARN)

Add Conditions (Optional)

**Add Statement**

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
*	Allow	s3:GetObject	arn:aws:s3:::prasad-website	None

**Step 3: Generate Policy**

A **policy** is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

**Generate Policy** **Start Over**

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided **as is** without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.  
An **amazon.com** company

Screenshot of the AWS Policy Generator tool showing a JSON policy document for an S3 bucket.

**AWS Service:** Amazon S3

**Actions:** All Actions (\*)

**Policy JSON Document:**

```
{
  "Id": "Policy1697316791653",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1697316788348",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::prasad-website/*",
      "Principal": "*"
    }
  ]
}
```

**Step 3: Review and Save**

A policy is a set of rules that define what actions are allowed or denied for specific users or groups. Policies can be applied at the bucket level or at the object level.

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided **as is** without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.  
An [amazon.com](#) company

**Screenshot of the AWS Management Console showing the policy editor for the S3 bucket.**

**Services:** Search [Alt+S]

**Edit statement**

Select a statement  
Select an existing statement in the policy or add a new statement.

+ Add new statement

**Code View:**

```
1 ▾ {
2   "Id": "Policy1697316791653",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1697316788348",
7       "Action": [
8         "s3:GetObject"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:s3:::prasad-website/*",
12      "Principal": "*"
13    }
14  ]
15 }
```

**Add new statement**

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 02:24 15-10-2023

6. Now open the link (given in the bucket below) in browser and you can see the static website hosted.

## CONCLUSION:

Here we studied to host a static website on S3 bucket.



Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

### **Assignment No 3**

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

**LAB OUTCOME** –

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

LO5: To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

### **THEORY** -

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc.

Terraform can manage infrastructure on multiple cloud platforms. Terraform's state allows you to track resource changes throughout your deployments. You can commit your configurations to version control to safely collaborate on infrastructure. Terraform plugins called providers let Terraform interact with cloud platform and other services via their application programming interfaces (APIs).

#### **A) Installation and Configuration of Terraform in Windows**

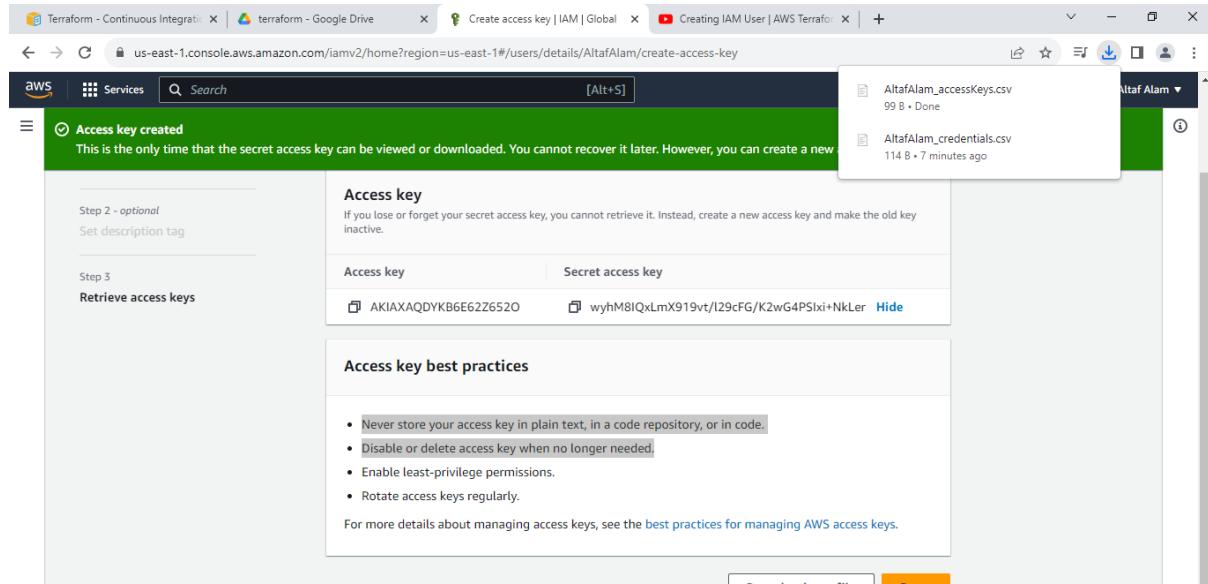
Step 1: Download terraform

To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website  
website: <https://www.terraform.io/downloads.html>

Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23



Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key.

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

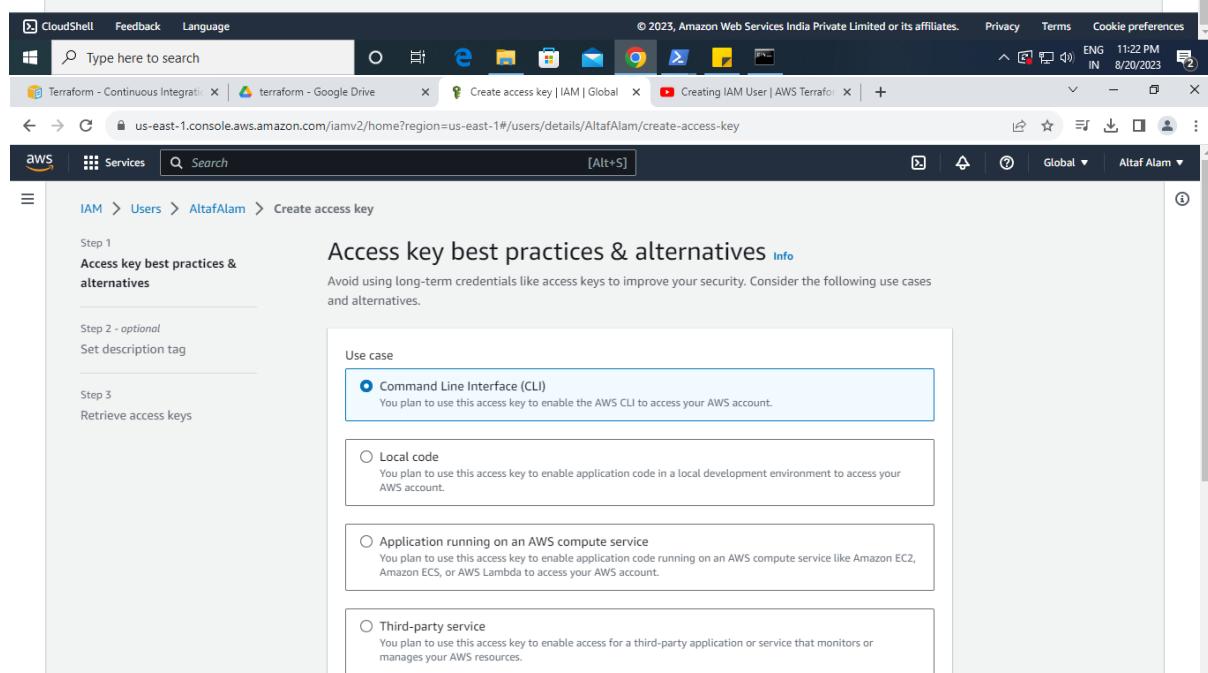
Access key	Secret access key
AKIAQ... <a href="#">View details</a>	wyhM8I... <a href="#">View details</a> <a href="#">Hide</a>

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file | Done



Access key best practices & alternatives [Info](#)

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

Use case

**Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.

**Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.

**Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

**Third-party service**  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG 11:22 PM IN 8/20/2023

Type here to search

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG 11:21 PM IN 8/20/2023

Type here to search

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS Identity and Access Management (IAM) service in the AWS Management Console. The user is creating a new IAM user named 'AltafAlam'. The 'Access keys' section is visible, showing that no access keys have been created. The 'SSH public keys for AWS CodeCommit' section also shows that no SSH public keys have been uploaded. The 'Security credentials' tab is selected, displaying the ARN of the user, their console access status (Enabled without MFA), and their creation date (August 20, 2023, 23:13 UTC-07:00). The 'Console sign-in' section shows the console sign-in link (<https://482117963900.signin.aws.amazon.com/console>) and the last console sign-in date (Never).

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM User Details page for the user 'AltafAlam'. The 'Summary' section displays the ARN (arn:aws:iam::482117963900:user/AltafAlam), Console access status (Enabled without MFA), and creation date (August 20, 2023, 23:13 (UTC-07:00)). The 'Permissions' tab is selected, showing two attached policies: 'Permissions policies (2)'. The 'Console sign-in details' section provides the console sign-in URL (<https://482117963900.sigin.aws.amazon.com/console>), user name (AltafAlam), and console password (l(BK984) - Hide). A 'View user' button is visible in the top right.

Identity and Access Management (IAM)

AltafAlam Info

Summary

ARN: arn:aws:iam::482117963900:user/AltafAlam

Console access: Enabled without MFA

Created: August 20, 2023, 23:13 (UTC-07:00)

Last console sign-in: Never

Permissions Groups (1) Tags Security credentials Access Advisor

Permissions policies (2)

Filter by Type: All types

Console sign-in details

Email sign-in instructions

Console sign-in URL: https://482117963900.sigin.aws.amazon.com/console

User name: AltafAlam

Console password: l(BK984) Hide

View user

CloudShell Feedback Language

Terraform - Continuous Integrati... terraform - Google Drive Create user | IAM | Global Creating IAM User | AWS Terrafo...

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/AltafAlam?section=permissions

Services Search [Alt+S]

Global Altaf Al... Delete

CloudShell Feedback Language

Terraform - Continuous Integrati... terraform - Google Drive Create user | IAM | Global Creating IAM User | AWS Terrafo...

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

Services Search [Alt+S]

Global Altaf Al... Delete

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Retrieve password

Console sign-in details

Email sign-in instructions

Console sign-in URL: https://482117963900.sigin.aws.amazon.com/console

User name: AltafAlam

Console password: l(BK984) Hide

Cancel Download .csv file Return to users list

CloudShell Feedback Language

Terraform - Continuous Integrati... terraform - Google Drive Create user | IAM | Global Creating IAM User | AWS Terrafo...

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/AltafAlam?section=permissions

Services Search [Alt+S]

Global Altaf Al... Delete

CloudShell Feedback Language

Terraform - Continuous Integrati... terraform - Google Drive Create user | IAM | Global Creating IAM User | AWS Terrafo...

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

Services Search [Alt+S]

Global Altaf Al... Delete

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM User creation process in three stages:

- Step 1: Specify user details**: Shows a success message: "User created successfully". It includes a link to "View user".
- Step 2: Set permissions**: Shows the "Console sign-in details" section with the following information:
  - Console sign-in URL: <https://482117963900.signin.aws.amazon.com/console>
  - User name: AltafAlam
  - Console password: \*\*\*\*\* (with a "Show" link)
- Step 4: Retrieve password**: Shows the "Permissions summary" table:

Name	Type	Used as
AdvDevops	Group	Permissions group
IAMUserChangePassword	AWS managed	Permissions policy

It also shows the "Tags - optional" section, which is currently empty.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM User creation interface. The user has been created with the name 'AltafAlam'. The 'Permissions summary' section shows the user is part of the 'AdvDevops' group and has an attached policy named 'IAMUserChangePassword'. There are no tags associated with the user.

Name	Type	Used as
AdvDevops	Group	Permissions group
IAMUserChangePassword	AWS managed	Permissions policy

The screenshot also shows the 'Create user group' interface. A new user group named 'AdvDevOps' is being created. Under the 'Permissions policies' section, the 'AdministratorAccess' policy is selected. The 'Create user group' button is highlighted in orange.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM User creation process at Step 3: Set permissions. It displays three options: Add user to group (selected), Copy permissions, and Attach policies directly. Below this, a table lists the 'User groups (1/1)' assigned to the user, which includes the 'AdvDevops' group with 'AdministratorAccess' and a creation date of 2023-08-20 (Now). A section for setting a permissions boundary is also visible.

**Set permissions**

**Permissions options**

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

**User groups (1/1)**

Group name	Users	Attached policies	Created
AdvDevops	0	AdministratorAccess	2023-08-20 (Now)

**Set permissions boundary - optional**

**Create IAM User**

**Console password**

Autogenerated password  
You can view the password after you create the user.

Custom password  
Enter a custom password for the user.

Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

**Note:** If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next Step**

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM User Creation Wizard and the resulting IAM Users list.

**Step 4: Retrieve password**

User name: AltafAlam

Provide user access to the AWS Management Console - *optional*

**Are you providing console access to a person?**

Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

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

**Next**

**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.

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

**Create user**

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

**IAM Dashboard**

**Security recommendations**

- Add MFA for root user
- Root user has no active access keys

**IAM resources**

User groups	Users	Roles	Policies	Identity providers
0	0	7	5	0

**AWS Account**

Account ID: 482117963900  
Account Alias: Create  
Sign-in URL for IAM users in this account: https://482117963900.sigin.aws.amazon.com/console

**Quick Links**

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

**User "AltafAlam" deleted.**

**Delete AdvDevops?**

Delete **AdvDevops** permanently? All the users in this group will lose the group permissions.  
This action cannot be undone.

To confirm deletion, enter the group name in the text input field.

**Cancel** **Delete**

**User groups**

**Create group**

**CloudShell** **Feedback** **Language**

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

ENG IN 10:44 PM 8/20/2023

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS IAM (Identity and Access Management) console. A modal dialog box titled "Delete AltafAlam?" is open, asking if the user wants to permanently delete the user "AltafAlam". It states that this action cannot be undone and will delete all user data, security credentials, and inline policies. The user name "AltafAlam" and last activity ("5 minutes ago") are displayed. Below the modal, the main IAM dashboard shows a list of users, roles, and policies.

**Delete AltafAlam?**

Delete AltafAlam permanently? This will also delete all its user data, security credentials and inline policies.

User name	Last activity
AltafAlam	5 minutes ago

Note: Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature. [Learn more](#)

This action cannot be undone.

To confirm deletion, enter the user name in the text input field.

[Cancel](#) [Delete user](#)

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

The screenshot shows the AWS EC2 (Elastic Compute Cloud) Instances page. It displays a table with one instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
-	i-02823f9368819c40d	Terminated	t2.micro	-	No alarms	us-east-1c

**Instances (1) Info**

**Select an instance**

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows two instances of Visual Studio Code side-by-side. Both instances have the same main.tf file open, which defines an AWS instance named 'AltafInstance'.

**Top Instance (Screenshot 1):**

- Terminal Output:**

```
main.tf - terraform script - Visual Studio Code
Release Notes: 1.81.1
main.tf
8   resource aws_instance AltafInstance {
9     ami           = "ami-053b0d53c279acc90"
10    instance_type = "t2.micro"

  - root_block_device {
    - delete_on_termination = true -> null
    - device_name          = "/dev/sda1" -> null
    - encrypted             = false -> null
    - iops                  = 100 -> null
    - tags                  = {} -> null
    - throughput            = 0 -> null
    - volume_id              = "vol-0fa9a246d5e7be52fc" -> null
    - volume_size            = 8 -> null
    - volume_type             = "gp2" -> null
  }
}

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

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

Enter a value: yes

aws_instance.AltafInstance: Destroying... [id=i-02823f9368819c40d]
aws_instance.AltafInstance: Still destroying... [id=i-02823f9368819c40d, 10s elapsed]
aws_instance.AltafInstance: Still destroying... [id=i-02823f9368819c40d, 20s elapsed]
aws_instance.AltafInstance: Still destroying... [id=i-02823f9368819c40d, 30s elapsed]
aws_instance.AltafInstance: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.
PS C:\terraform script>
```
- Status Bar:** Ln 11, Col 2 | Spaces:4 | UTF-8 | Plain Text | Go Live | IN 8/20/2023

**Bottom Instance (Screenshot 2):**

- Terminal Output:**

```
main.tf - terraform script - Visual Studio Code
Release Notes: 1.81.1
main.tf
8   resource aws_instance AltafInstance {
9     ami           = "ami-053b0d53c279acc90"
10    instance_type = "t2.micro"

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\terraform script> terraform destroy
aws_instance.AltafInstance: Refreshing state... [id=i-02823f9368819c40d]

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

Terraform will perform the following actions:

# aws_instance.AltafInstance will be destroyed
- resource "aws_instance" "AltafInstance" {
    - ami           = "ami-053b0d53c279acc90" -> null
    - arm          = "arn:aws:ec2:us-east-1:482117963900:instance/i-02823f9368819c40d" -> null
    - associate_public_ip_address = true -> null
    - availability_zone          = "us-east-1c" -> null
    - cpu_core_count            = 1 -> null
    - cpu_threads_per_core      = 1 -> null
    - disable_api_stop          = false -> null
    - disable_api_termination    = false -> null
    - ebs_optimized             = false -> null
    - get_password_data         = false -> null
    - hibernation               = false -> null
    - id                      = "i-02823f9368819c40d" -> null
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state             = "running" -> null
    - instance_type              = "t2.micro" -> null
    - ipv6_address_count        = 0 -> null
    - ipv6_addresses             = [] -> null
    - monitoring                = false -> null
    - placement_partition_number = 0 -> null
}
```
- Status Bar:** Ln 11, Col 2 | Spaces:4 | UTF-8 | Plain Text | Go Live | IN 8/20/2023

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS EC2 Instances page. The main table displays one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
-	i-02823f9368819c40d	Running	t2.micro	Initializing	No alarms	us-east-1c

A modal window titled "Select an instance" is open, indicating that an action is about to be performed on the selected instance.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows two instances of Visual Studio Code side-by-side, both displaying a Terraform configuration file named `main.tf`. The code defines an AWS instance resource named "AltafInstance".

```
resource "aws_instance" "AltafInstance" {  
    ami = "ami-053bd53c279acc90"  
    ...  
}
```

In the top instance, the terminal output shows the Terraform plan and apply process:

```
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.AltafInstance: Creating...  
aws_instance.AltafInstance: Still creating... [10s elapsed]  
aws_instance.AltafInstance: Still creating... [20s elapsed]  
aws_instance.AltafInstance: Still creating... [30s elapsed]  
aws_instance.AltafInstance: Creation complete after 40s [id=i-02823f9368819c40d]  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  
PS C:\terraform script>
```

In the bottom instance, the terminal output shows the validation and execution results:

```
Success! The configuration is valid.  
PS 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:  
# aws_instance.AltafInstance will be created  
+ resource "aws_instance" "AltafInstance" {  
    ami = "ami-053bd53c279acc90"  
    ...  
}
```

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows two instances of Visual Studio Code side-by-side. Both instances have the same workspace structure:

- EXPLORER**: Shows files: .terraform, .terraform.lock.hcl, and main.tf.
- TERMINAL**: Shows the command "PS C:\terraform script> terraform plan".
- OUTPUT**: Shows the execution plan output.
- DEBUG CONSOLE**: Not visible.
- TERMINAL**: Active tab.

**Terminal Output (top instance):**

```
main.tf - terraform script - Visual Studio Code
Release Notes: 1.81.1
main.tf x

main.tf
6 }
7
8 resource "aws_instance" "AltafInstance" {
    + monitoring          = (known after apply)
    + outpost_arn          = (known after apply)
    + password_data        = (known after apply)
    + placement_group       = (known after apply)
    + placement_partition_number = (known after apply)
    + primary_network_interface_id = (known after apply)
    + private_dns           = (known after apply)
    + private_ip             = (known after apply)
    + public_dns             = (known after apply)
    + public_ip              = (known after apply)
    + secondary_private_ips = (known after apply)
    + security_groups        = (known after apply)
    + source_dest_check     = true
    + spot_instance_request_id = (known after apply)
    + subnet_id              = (known after apply)
    + tags_all               = (known after apply)
    + tenancy                = (known after apply)
    + user_data              = (known after apply)
    + user_data_base64        = (known after apply)
    + user_data_replace_on_change = false
    + vpc_security_group_ids = (known after apply)
}

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\terraform script>
```

**Terminal Output (bottom instance):**

```
main.tf - terraform script - Visual Studio Code
Release Notes: 1.81.1
main.tf x

main.tf
6 }
7
8 resource "aws_instance" "AltafInstance" {
    + create

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.
PS 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:

# aws_instance.AltafInstance will be created
+ resource "aws_instance" "AltafInstance" {
    + ami                      = "ami-053b0d53c279acc90"
    + arn                      = (known after apply)
    + associate_public_ip_address = (known after apply)
    + availability_zone         = (known after apply)
    + cpu_core_count            = (known after apply)
    + cpu_threads_per_core      = (known after apply)
    + disable_api_stop          = (known after apply)
    + disable_api_termination   = (known after apply)
    + ebs_optimized             = (known after apply)
    + get_password_data         = false
    + host_id                  = (known after apply)
    + host_resource_group_arn   = (known after apply)
    + iam_instance_profile      = (known after apply)
    + id                        = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle        = (known after apply)
}
```

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot displays a dual-monitor setup. The left monitor shows a Windows desktop with several open browser tabs, including 'Terraform - Continuous Integration', 'terraform - Google Drive', 'AltafAlam | IAM | Global', 'Launch an instance | EC2 Manager', and 'Day #9: How to create an EC2...'. Below these tabs is a search bar and a navigation bar with icons for AWS services like Lambda, S3, and CloudWatch. The main workspace is a Visual Studio Code window titled 'main.tf - terraform script - Visual Studio Code'. It shows a 'TERRAFORM SCRIPT' folder structure with files '.terraform', '.terraform.lock.hcl', and 'main.tf'. The 'TERMINAL' tab is active, displaying the command-line output of running Terraform commands: 'terraform fmt', 'terraform init', and 'Terraform has been successfully initialized!'. The right monitor shows the AWS CloudFormation console. A search bar at the top finds 'ubuntu'. Under the 'Quick Start' section, there are cards for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. An 'ubuntu' card is selected, showing its details: 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type', AMI ID 'ami-053b0d53c279acc90', and a note that it's 'Free tier eligible'. To the right, a 'Summary' section shows 'Number of instances Info' (1) and a 'Browse more AMIs' link. A modal dialog box is open, stating 'Free tier: In your first year' and containing 'Cancel', 'Launch instance', and 'Review commands' buttons.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows the AWS Cloud Console interface for launching an EC2 instance. The top navigation bar includes tabs for 'Terraform - Continuous Intell.', 'terraform - Google Drive', 'AltafAlam | IAM | Global', 'Launch an instance | EC2 Main', 'Day #9: How to create an EC...', and a '+' button. The main title is 'Launch an instance'. The 'Summary' section on the right shows the following configuration:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023.1.2... (ami-08a52ddb321b52a8c)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A 'Free tier: In your first year' message is displayed. At the bottom are 'Cancel' and 'Launch instance' buttons.

The left sidebar shows the navigation path: EC2 > Instances > Launch an instance. The main content area contains sections for 'Name and tags' and 'Application and OS Images (Amazon Machine Image)'. The 'Name and tags' section has a 'Name' field containing 'AltafInstance'. The 'Application and OS Images' section includes a search bar with the placeholder 'Search our full catalog including 1000s of application and OS images'.

The browser status bar at the bottom indicates the URL 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:', the date '8/20/2023', and the time '11:42 PM'. The interface is in English (ENG) and India (IN).

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

The screenshot shows two views of the AWS EC2 Management console:

- Instances Info:** This view displays a table with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. A search bar at the top allows filtering by attribute or tag. A large message in the center states "No instances" and "You do not have any instances in this region". A prominent orange "Launch instances" button is located at the bottom right.
- EC2 Dashboard:** This view provides an overview of various Amazon EC2 resources in the US East (N. Virginia) Region. It includes counts for Instances (running), Dedicated Hosts, Auto Scaling Groups, Elastic IPs, Key pairs, Placement groups, Security groups, Snapshots, and Volumes. A callout box highlights the "Launch instance" feature, which is described as easily sizing, configuring, and deploying Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Below this, there's a "Service health" section.

## CONCLUSION :

Altaf Alam , 02 , T11

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

Hence , installed and configure terraform . Also created Iamuser in terraform and learn about how to use it.

## LAB Assignment 5

**AIM:** To Build, change, and destroy AWS infrastructure Using Terraform.

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

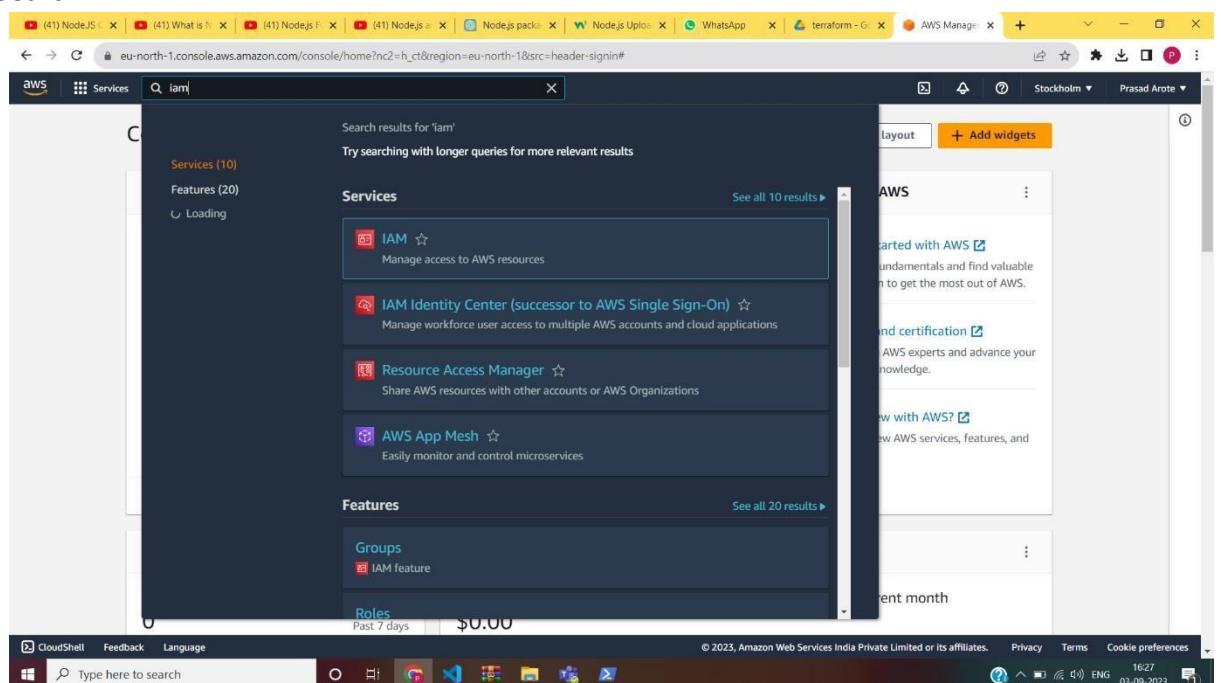
**LO5:** To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

### Theory:

- 1) Make dir Terraform Scripts Open [aws.amazon.com](https://aws.amazon.com)

Login to your account

Search IAM



- 2) Click on Users ( on the LHS )

## T11 Altaf Alam 04

The screenshot shows the AWS IAM service interface. On the left, a sidebar menu includes 'Dashboard', 'Access management' (with 'User groups', 'Users' selected, 'Roles', 'Policies', 'Identity providers', and 'Account settings'), 'Access reports' (with 'Access analyzer', 'Archive rules', 'Analyzers', and 'Settings'), and 'Credential report' (with 'Organization activity'). The main content area is titled 'Users (0) Info' and contains the message 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar and a table header ('User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', 'Console last sign-in') are visible. Below the table, it says 'No resources to display'. At the top right, there are 'Delete' and 'Create user' buttons. The bottom of the screen shows the Windows taskbar with various pinned icons.

### 3) Click Add users

The screenshot shows the 'Create user' wizard, Step 1: Specify user details. The left sidebar lists 'Step 1: Specify user details', 'Step 2: Set permissions', and 'Step 3: Review and create'. The main area is titled 'Specify user details' and contains a 'User details' section. In the 'User name' field, 'prasad' is entered. A note below states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +-=.,@\_- (hyphen)' and 'Provide user access to the AWS Management Console - optional'. A note at the bottom says: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)'.

### 4) Set Permissions -> AmazonEC2FullAccess

## T11 Altaf Alam 04

The screenshot shows the AWS IAM 'Create user' wizard at the 'Set permissions' step. The 'Attach policies directly' option is selected. In the 'Permissions policies' section, 'AmazonEC2FullAccess' is selected.

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess	AWS managed - job function	0
AdministratorAccess-Amplify	AWS managed	0
AmazonEC2ContainerServiceforEC2...	AWS managed	0
AmazonEC2ContainerServiceRole	AWS managed	0
<b>AmazonEC2FullAccess</b>	AWS managed	0
AmazonEC2ReadOnlyAccess	AWS managed	0
AmazonEC2RoleforAWSCodeDeploy	AWS managed	0
AmazonEC2RoleforAWSCodeDeploy...	AWS managed	0
AmazonEC2RoleforDataPipelineRole	AWS managed	0
AmazonEC2RoleforSSM	AWS managed	0
AmazonEC2RolePolicyForLaunchWi...	AWS managed	0
AmazonEC2SpotFleetAutoscaleRole	AWS managed	0

## 5) Create User

## T11 Altaf Alam 04

The screenshot shows the AWS IAM console with a success message: "User created successfully". The message indicates that a new IAM user named "prasad" has been created. The left sidebar shows navigation options like Dashboard, Access management, and Access reports. The main area displays a table of users, with "prasad" listed as the first entry.

### 6) Create access key

The screenshot shows the "Create access key" step in the AWS IAM console. It displays a list of use cases for creating access keys. The "Command Line Interface (CLI)" option is selected. Other options include Local code, Application running on an AWS compute service, Third-party service, and Application running outside AWS. The left sidebar shows the navigation path: IAM > Users > prasad > Create access key.

### 7) Download the .csv file

The screenshot shows the AWS IAM 'Create access key' page. A green banner at the top says 'Access key created' with the note: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, the 'Retrieve access keys' section shows the generated access key details:

Access key	Secret access key
AKIAQTTZTO7NGUOFNMJP	***** Show

Below the table, the 'Access key best practices' section lists several guidelines:

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

At the bottom right, there are 'Download .csv file' and 'Done' buttons.

8) Open EC2 Instances and Copy the AMI ID of any one of them.

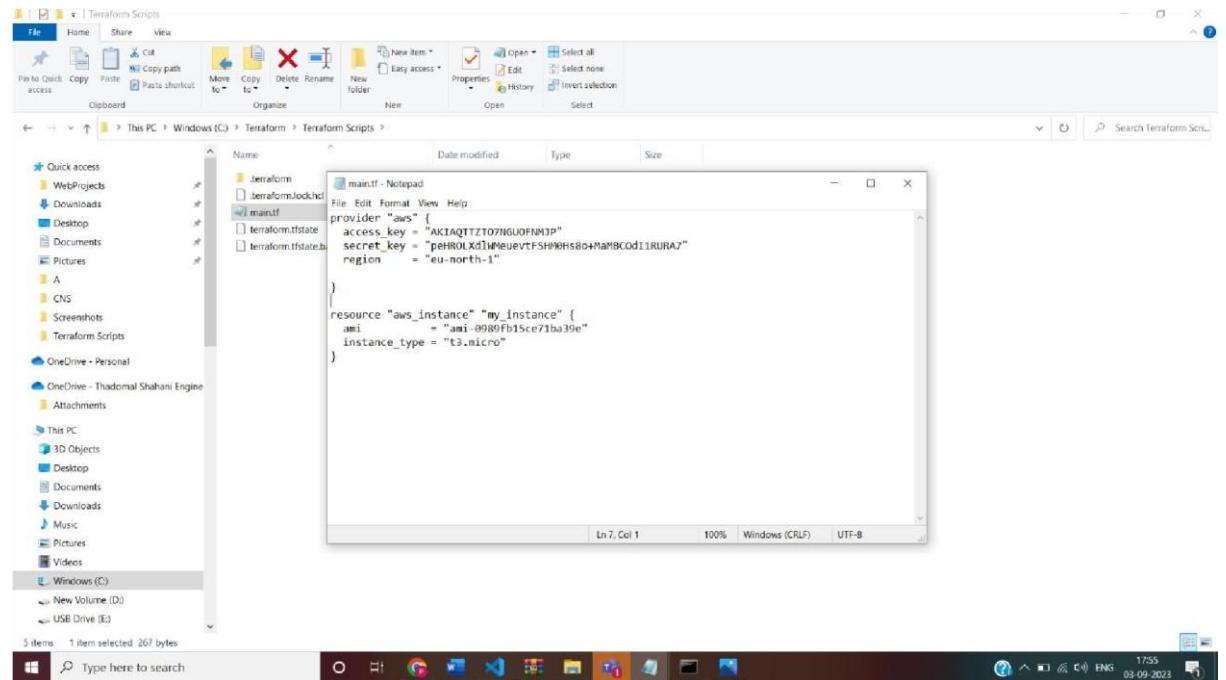
The screenshot shows the AWS EC2 'Launch instances' page. In the 'Quick Start' section, an AMI is selected: 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type'. The 'AMI ID' is listed as 'ami-0989fb15ce71ba39e'. The 'Summary' section shows the following details:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 22.04 LTS, ...read more  
ami-0989fb15ce71ba39e
- Virtual server type (instance type):** t3.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

A tooltip for the 'Free tier' button provides information: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.'

At the bottom right, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

9) Configure the main.tf file



- 10) Run the commands in cmd -> terraform init , terraform validate , terraform plan, terraform apply to create EC2 instance using terraform.

```

C:\Windows\System32\cmd.exe
C:\Terraform\Terraform Scripts>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.15.0...
- Installed hashicorp/aws v5.15.0 (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,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Terraform\Terraform Scripts>terraform validate
Success! The configuration is valid.

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

Terraform will perform the following actions:

# aws_instance.my_instance will be created
+ resource "aws_instance" "my_instance" {
    + ami           = "ami-0089fh15ce71ba39e"
    + arn           = (known after apply)
    + associate_public_ip_address = (known after apply)
    + availability_zone      = (known after apply)
    + cpu_core_count        = (known after apply)
    + cpu_threads_per_core = (known after apply)
    + disable_api_stop      = (known after apply)
    + direct_connect_termination = (known after apply)
    + ebs_optimized         = (known after apply)
    + encrypted_password_data = (known after apply)
    + host_id              = (known after apply)
    + host_resource_group_arn = (known after apply)
}
```

T11 Altaf Alam 04

```
C:\Windows\System32\cmd.exe
if you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Terraform\Terraform Scripts>terraform validate
Success! The configuration is valid.

C:\Terraform\Terraform Scripts>terraform plan

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

Terraform will perform the following actions:

# aws_instance.my_instance will be created
+ resource "aws_instance" "my_instance" {
    + ami                                = "ami-0989f1b5ce71ba39e"
    + ami_id                             = "(known after apply)"
    + associate_public_ip_address        = "(known after apply)"
    + availability_zone                  = "(known after apply)"
    + cpu_core_count                     = "(known after apply)"
    + cpu_threads_per_core              = "(known after apply)"
    + disable_api_stop                  = "(known after apply)"
    + disable_api_termination           = "(known after apply)"
    + ebs_optimized                      = "(known after apply)"
    + get_password_data                 = false
    + host_id                            = "(known after apply)"
    + host_resource_group_arn            = "(known after apply)"
    + iam_instance_profile               = "(known after apply)"
    + id                                 = "(known after apply)"
    + instance_initiated_shutdown_behavior = "(known after apply)"
    + instance_lifecycle                = "(known after apply)"
    + instance_state                     = "(known after apply)"
    + instance_type                      = "t3.micro"
    + ipv6_address_count                = "(known after apply)"
    + ipv6_addresses                     = "(known after apply)"
    + key_name                           = "(known after apply)"
    + monitoring                         = "(known after apply)"
    + outpost_arn                        = "(known after apply)"
    + password_data                      = "(known after apply)"
    + placement_group                   = "(known after apply)"
    + placement_partition_number         = "(known after apply)"
    + primary_network_interface_id      = "(known after apply)"
    + private_dns                        = "(known after apply)"
    + private_ip                         = "(known after apply)"
    + public_dns                          = "(known after apply)"
    + public_ip                           = "(known after apply)"
    + secondary_private_ips             = "(known after apply)"
    + security_groups                    = "(known after apply)"
```

```
C:\>get_password_data
+ host_id = false
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t3.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)
+ network_interface = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)
+ private_ip = (known after apply)
+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags_all = (known after apply)
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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.my_instance: Creating...
aws_instance.my_instance: Still creating... [10s elapsed]
aws_instance.my_instance: Creation complete after 16s [id=i-073badc5b739e9ea]

apply complete! Resources: 1 added, 0 changed, 0 destroyed.

E:\Terraform\Terraform Scripts>
```

## T11 Altaf Alam 04

The screenshot shows the AWS EC2 Instances page. A single instance is listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-073badc5b739e9ea0	Running	t3.micro	Initializing	No alarms	eu-north-1b	ec2-51-20-3-21.e

A modal window titled "Select an instance" is open, prompting the user to choose which instance to terminate.

### 11) Destroy the instance using terraform destroy.

```
cd\Windows\System32\cmd.exe
aws_instance {
  maintenance_options {
    - auto_recovery = "default" -> null
  }
  metadata_options {
    http_endpoint      = "enabled" -> null
    http_protocol_ipv6 = "disabled" -> null
    http_put_response_hop_limit = 1 -> null
    http_tokens        = "optional" -> null
    instance_metadata_tags = "disabled" -> null
  }
  private_dns_name_options {
    enable_resource_name_dns_a_record   = false -> null
    enable_resource_name_dns_aaaa_record = false -> null
    hostname_type                      = "ip-name" -> null
  }
  root_block_device {
    delete_on_termination = true -> null
    device_name          = "/dev/sda1" -> null
    encrypted            = false -> null
    iops                 = 100 -> null
    kms_id               = "" -> null
    throughput           = 0 -> null
    volume_id            = "vol-070b9163befc9c2f1" -> null
    volume_size          = 8 -> null
    volume_type          = "gp2" -> null
  }
}
Plan: 0 to add, 0 to change, 1 to destroy.

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

Enter a value: yes
aws_instance.my_instance: Destroying... [id=i-073badc5b739e9ea0]
aws_instance.my_instance: Still destroying... [id=i-073badc5b739e9ea0, 10s elapsed]
aws_instance.my_instance: Still destroying... [id=i-073badc5b739e9ea0, 20s elapsed]
aws_instance.my_instance: Still destroying... [id=i-073badc5b739e9ea0, 30s elapsed]
aws_instance.my_instance: Still destroying... [id=i-073badc5b739e9ea0, 40s elapsed]
aws_instance.my_instance: Destruction complete after 42s

Destroy complete! Resources: 1 destroyed.
C:\Terraform\Terraform Scripts>
```

**CONCLUSION:** Here, we understood the use of terraform and we have successfully created a EC2 instances and destroyed it using terraform.

## **Lab Assignment 6**

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

**LO4:** To identify and remediate application vulnerabilities earlier and help integrate security in the development process using SAST Techniques.

### **THEORY:**

SonarQube:

Overview: SonarQube is an open-source platform for continuous inspection of code quality. It is used to analyze and measure code quality and security issues in a codebase.

Features:

Static Code Analysis: SonarQube scans source code to identify bugs, code smells, and security vulnerabilities.

Continuous Integration: It integrates seamlessly with CI/CD pipelines, providing automated code analysis during the development process.

Security Analysis: While it primarily focuses on code quality, it also has some security rules to catch common security issues.

Maintainability Metrics: SonarQube provides maintainability metrics and helps teams understand code complexity and maintainability.

Dashboard and Reporting: It offers dashboards and reports for tracking code quality and issues over time.

Use Case: SonarQube is used for improving code quality, maintainability, and to catch some common code security issues. It's more about general code quality and development best practices.

SAST (Static Application Security Testing):

Overview: SAST is a security testing method that analyzes source code, bytecode, or binary code for vulnerabilities without executing the application. It is primarily focused on identifying security issues and vulnerabilities in the code.

Features:

Code Scanning: SAST tools examine the source code or compiled code to identify potential security vulnerabilities, such as SQL injection, cross-site scripting, and more.

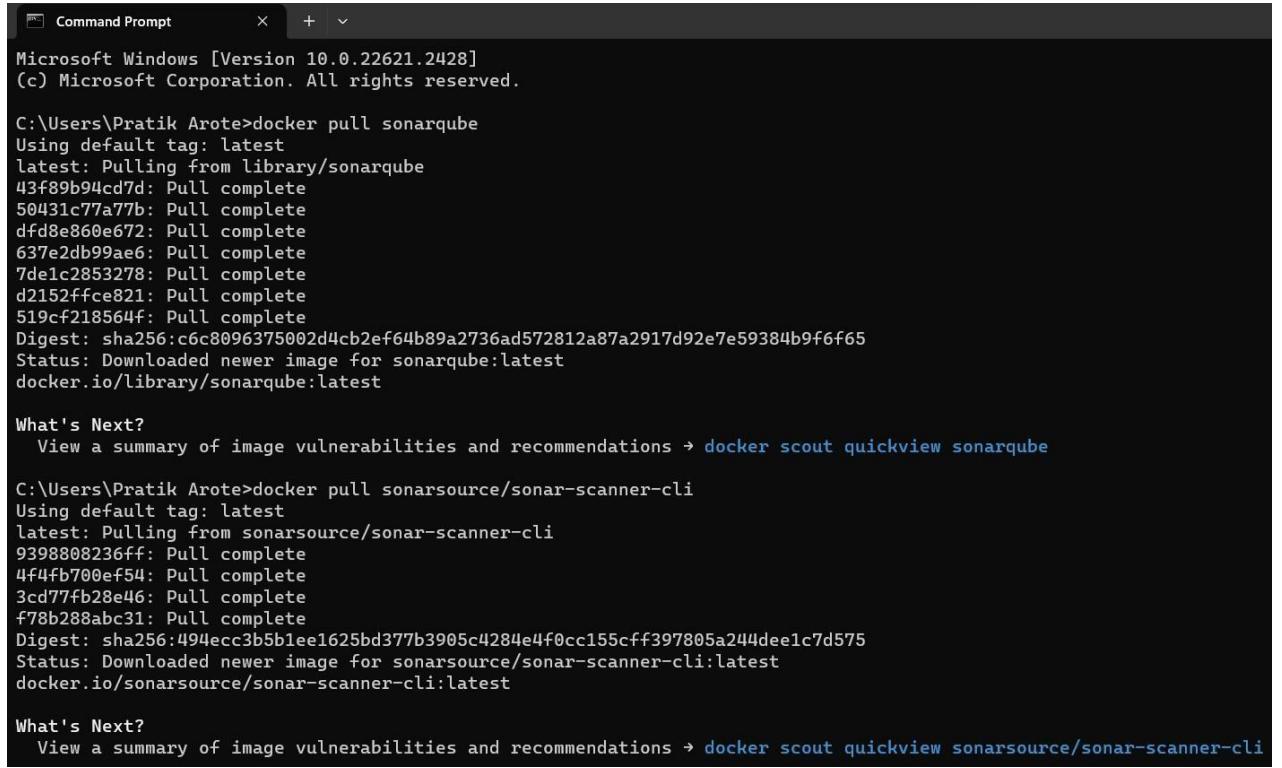
Early Detection: SAST is used early in the development process to find security issues before they can be exploited.

Language Support: SAST tools support various programming languages and frameworks.

Integration: They can be integrated into CI/CD pipelines to automatically scan code before deployment.

Use Case: SAST is used for finding and fixing security vulnerabilities in code. It helps secure applications by identifying potential security threats early in the development lifecycle.

1. INSTALL sonarqube (docker images) and sonarscanner zip file from <https://docs.sonarsource.com/sonarqube/latest/analyzingsourcecode/scanners/sonarscanner/> and set up config file as given in docs.



```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

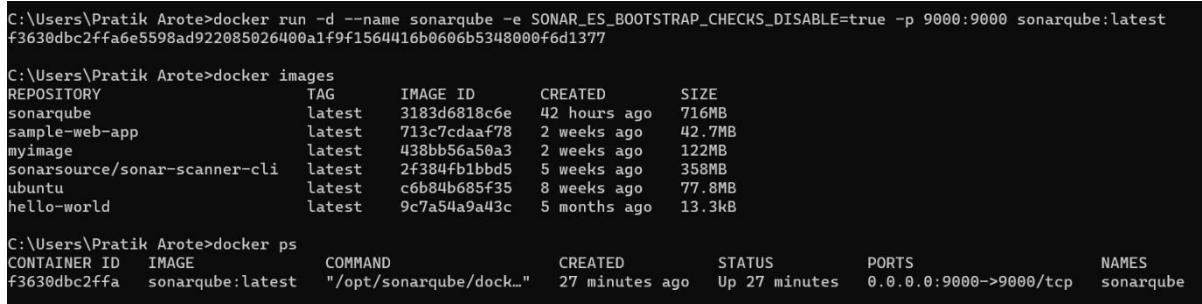
C:\Users\Pratik Arote>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
43f89b94cd7d: Pull complete
50431c77a77b: Pull complete
dfd8e860e672: Pull complete
637e2db99ae6: Pull complete
7de1c2853278: Pull complete
d2152ffce821: Pull complete
519cf218564f: Pull complete
Digest: sha256:c6c8096375002d4cb2ef64b89a2736ad572812a87a2917d92e7e59384b9f6f65
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube

C:\Users\Pratik Arote>docker pull sonarsource/sonar-scanner-clí
Using default tag: latest
latest: Pulling from sonarsource/sonar-scanner-clí
9398808236ff: Pull complete
4f4fb700ef54: Pull complete
3cd77fb28e46: Pull complete
f78b288abc31: Pull complete
Digest: sha256:494ecc3b5b1ee1625bd377b3905c4284e4f0cc155cff397805a244dee1c7d575
Status: Downloaded newer image for sonarsource/sonar-scanner-clí:latest
docker.io/sonarsource/sonar-scanner-clí:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview sonarsource/sonar-scanner-clí
```

2. Spin up the container

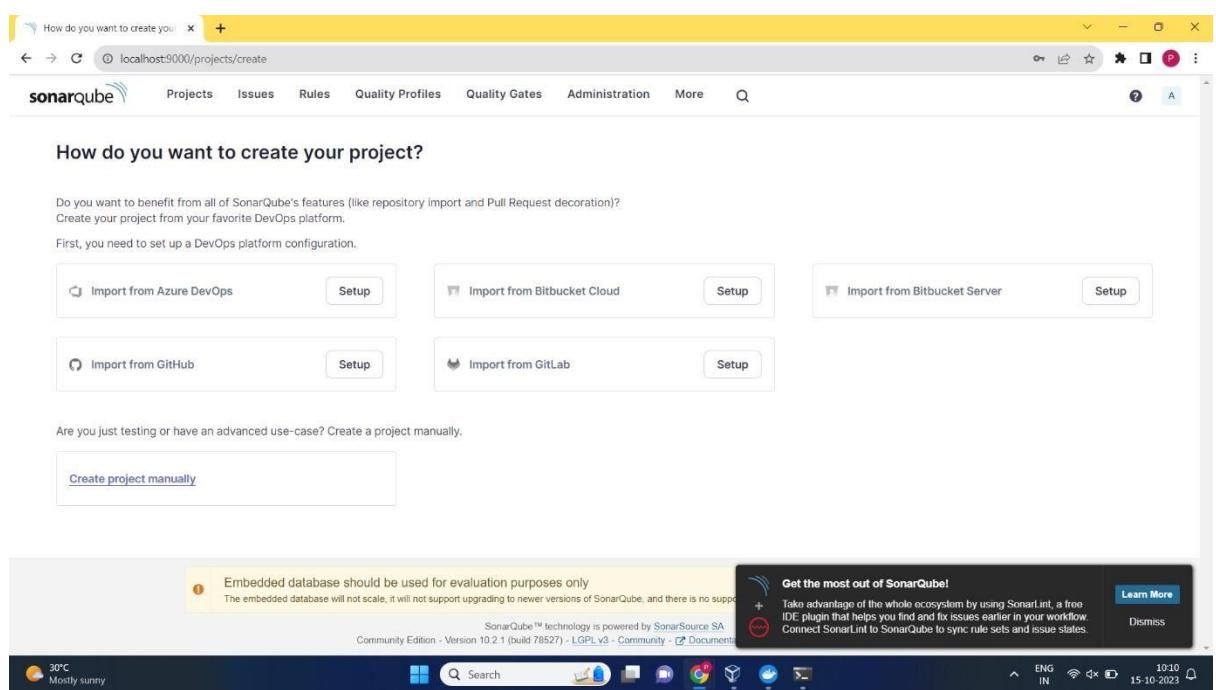
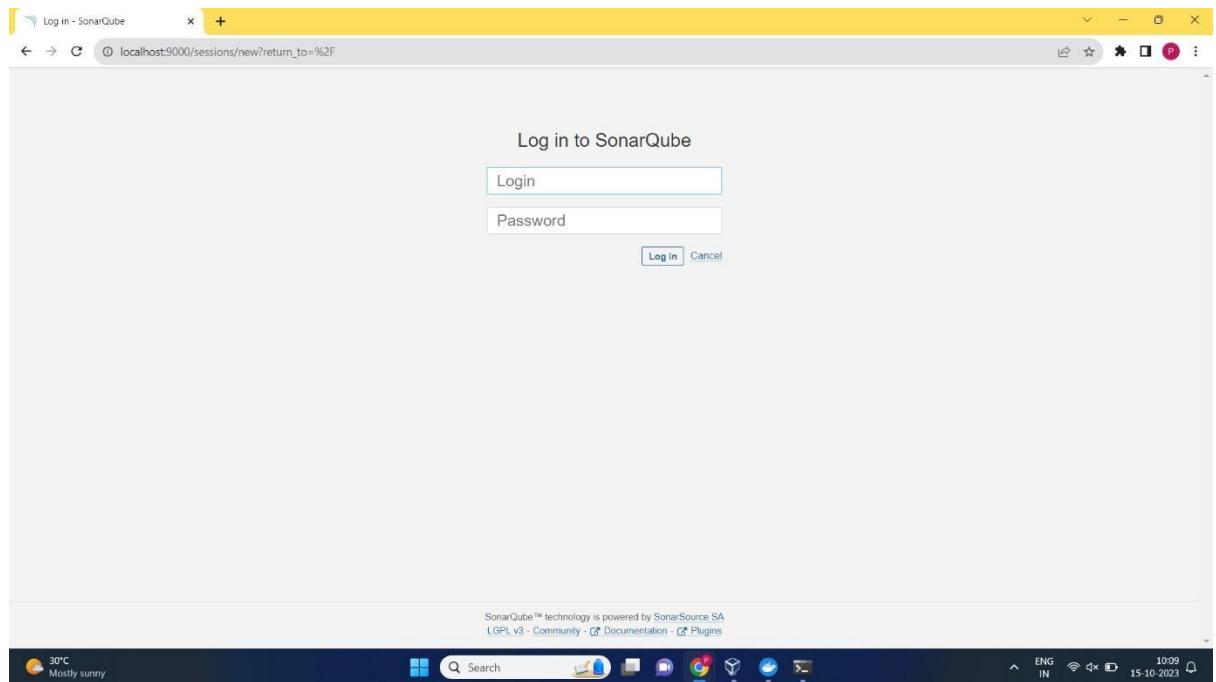


```
C:\Users\Pratik Arote>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
f3630dbc2ffa6e5598ad922085026400a1f9f1564416b0606b5348000f6d1377

C:\Users\Pratik Arote>docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
sonarqube           latest   3183d6818c6e  42 hours ago  716MB
sample-web-app      latest   713c7cdaf78   2 weeks ago   42.7MB
myimage              latest   438bb56a50a3  2 weeks ago   122MB
sonarsource/sonar-scanner-clí  latest   2f384fb1bbd5  5 weeks ago   358MB
ubuntu               latest   c6b84b685f35  8 weeks ago   77.8MB
hello-world          latest   9c7a54a9a43c  5 months ago  13.3kB

C:\Users\Pratik Arote>docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
f3630dbc2ffa        sonarqube:latest   "/opt/sonarqube/dock..."   27 minutes ago    Up 27 minutes   0.0.0.0:9000->9000/tcp   sonarqube
```

3. Open <http://localhost:9000> on the browser. Enter login and password both as “admin” and then set up new password.



#### 4. Create a project

Create a project

localhost:9000/projects/create?mode=manual

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

**Create a project**

Project display name \*

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

Project key \*

 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.

Main branch name \*

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

**Next**

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 external databases.

Get the most out of SonarQube!  
SonarQube™ technology is powered by SonarSource SA  
Community Edition - Version 10.2.1 (build 78527) - LGPL v3 - Community - [Documents](#)

30°C Mostly sunny

Create a project

localhost:9000/projects/create?mode=manual&setncd=true

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

**Set up project for Clean as You Code**

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

Choose the baseline for new code for this project

Use the global setting

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

Define a specific setting for this project

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

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

Get the most out of SonarQube!  
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue stats.

ENG IN 10:22 15-10-2023

The screenshot shows the SonarQube interface for setting up a new project. At the top, a banner says "Congratulations! Your project has been created." Below it, there are several options for CI integration:

- With Jenkins**
- With GitHub Actions**
- With Bitbucket Pipelines**
- With GitLab CI**
- With Azure Pipelines**
- Other CI**: A note states "SonarQube integrates with your workflow no matter which CI tool you're using."
- Locally**: A note says "Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment."

A yellow warning box at the bottom left says "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 it." To the right, a "Get the most out of SonarQube!" sidebar offers SonarLint integration.

## 5. Provide token

The screenshot shows the SonarQube interface after a token has been generated. The banner now says "We initialized your project on SonarQube, now it's up to you to launch analyses!"

**1 Provide a token**

A token is displayed: `pythonToken: sqp_c8922aed03df90f4cc0dfb26200772ff42a9032b`. A link to revoke it is shown.

**2 Run analysis on your project**

A yellow warning box at the bottom left says "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 it." To the right, a "Get the most out of SonarQube!" sidebar offers SonarLint integration.

**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 `bins` directory to the `%PATH%` environment variable

Execute the Scanner

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

```
sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_c8922aed03df90f4cc0dfb26200772ff42a9032b"
```

Please visit the [official documentation of the Scanner](#) for more details.

## 6. Enter the following command

```
C:\Windows\System32\cmd.exe x + 
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\sonar-scanner-5.0.1.3006-windows\bin>sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=C:\Users\Pratik Arote\Desktop\sastPython" -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_c8922aed03df90f4cc0dfb26200772ff42a9032b" -D"sonar.projectBaseDir=C:\Users\Pratik Arote\Desktop\sastPython"
INFO: Scanner configuration file: C:\Program Files\sonar-scanner-5.0.1.3006-windows\bin..\conf\sonar-scanner.properties
INFO: Project root configuration file: C:\Users\Pratik Arote\Desktop\sastPython\sonarPythonProgram1\sonarPythonProgram1\src\main\java\com\pratik\arote\sonar\ScannerTest.java
INFO: SonarScanner 5.0.1.3006
INFO: Java 17.0.7 Eclipse Adoptium (64-bit)
INFO: Windows 11 10.0 amd64
INFO: User cache: C:\Users\Pratik Arote\.sonar\cache
INFO: Analyzing on SonarQube server 10.2.1.78527
INFO: Default locale: "en_IN", source code encoding: "windows-1252" (analysis is platform dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=58ms
INFO: Server id: 147B411E-AYsxoFDzQL-rufd2_SS
INFO: User cache: C:\Users\Pratik Arote\.sonar\cache
INFO: Load/download plugins
INFO: Load/plugins index
INFO: Load/plugins index (done) | time=338ms
INFO: Load/download plugins (done) | time=8251ms
INFO: Process project properties
INFO: Process project properties (done) | time=40ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=7ms
INFO: Project key: sonarPythonProgram1
INFO: Base dir: C:\Users\Pratik Arote\Desktop\sastPython
INFO: Working dir: C:\Users\Pratik Arote\Desktop\sastPython\scannerwork
INFO: Load project settings for component key: 'sonarPythonProgram1'
INFO: Load project settings for component key: 'sonarPythonProgram1' (done) | time=122ms
WARN: SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=597ms
INFO: Load active rules
INFO: Load active rules (done) | time=7984ms
INFO: Load analysis cache
INFO: Load analysis cache (404) | time=60ms
INFO: Load project repositories
INFO: Load project repositories (done) | time=295ms
```

```

C:\Windows\System32\cmd.exe x + v
INFO: Sensor VB.NET Properties [vbnet] (done) | time=2ms
INFO: Sensor IaC Docker Sensor [iac]
INFO: 0 source files to be analyzed
INFO: 0/0 source files have been analyzed
INFO: Sensor IaC Docker Sensor [iac] (done) | time=206ms
INFO: -----
INFO: Run sensors on project
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=7ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=47ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor 1 file had no CPD blocks
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 253ms, dir size=136.5 kB
INFO: Analysis report compressed in 48ms, zip size=17.5 kB
INFO: Analysis report uploaded in 201ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: 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=Aysx47EpoQL-ruFd3M3Y
INFO: Analysis total time: 22.756 s
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=Aysx47EpoQL-ruFd3M3Y
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 35.565s
INFO: Final Memory: 23M/77M
INFO: -----
C:\Program Files\sonar-scanner-5.0.1.3006-windows\bin>

```

## 7. See the result of the test

The screenshot shows the SonarQube interface for the 'sonarPythonProgram1' project. The main dashboard features a large green 'Passed' badge with a checkmark, indicating the quality gate status. Below this, there's a decorative icon of a computer monitor with code snippets and a checkmark. A message at the bottom encourages users to set up analysis in their favorite CI. To the right, there are several performance measures displayed in boxes, each with an 'A' grade: Reliability (0 Bugs), Maintainability (0 Code Smells), Security (0 Vulnerabilities), Security Review (0 Security Hotspots), Coverage (0.0% Coverage), and Duplications (0.0% Duplications).

## CONCLUSION:

Here we have successfully performed static analysis of python programs.

## Lab Assignment 7

**AIM:** To understand AWS Lambda functions and create a Lambda function using Python to log “An Image has been added” message, once a file is added to a S3 bucket.

**LO6:** To engineer a composition of nano services using AWS Lambda and Step Functions with the Serverless Framework.

### THEORY:

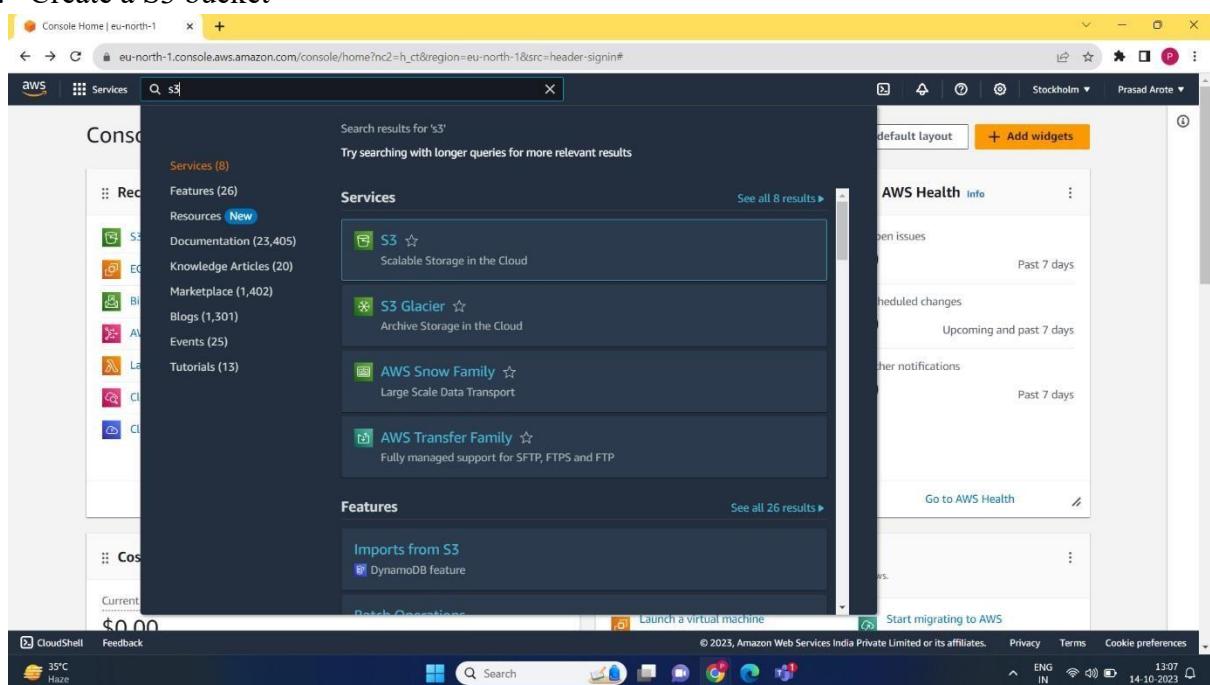
#### LAMBDA FUNCTION

AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you use.

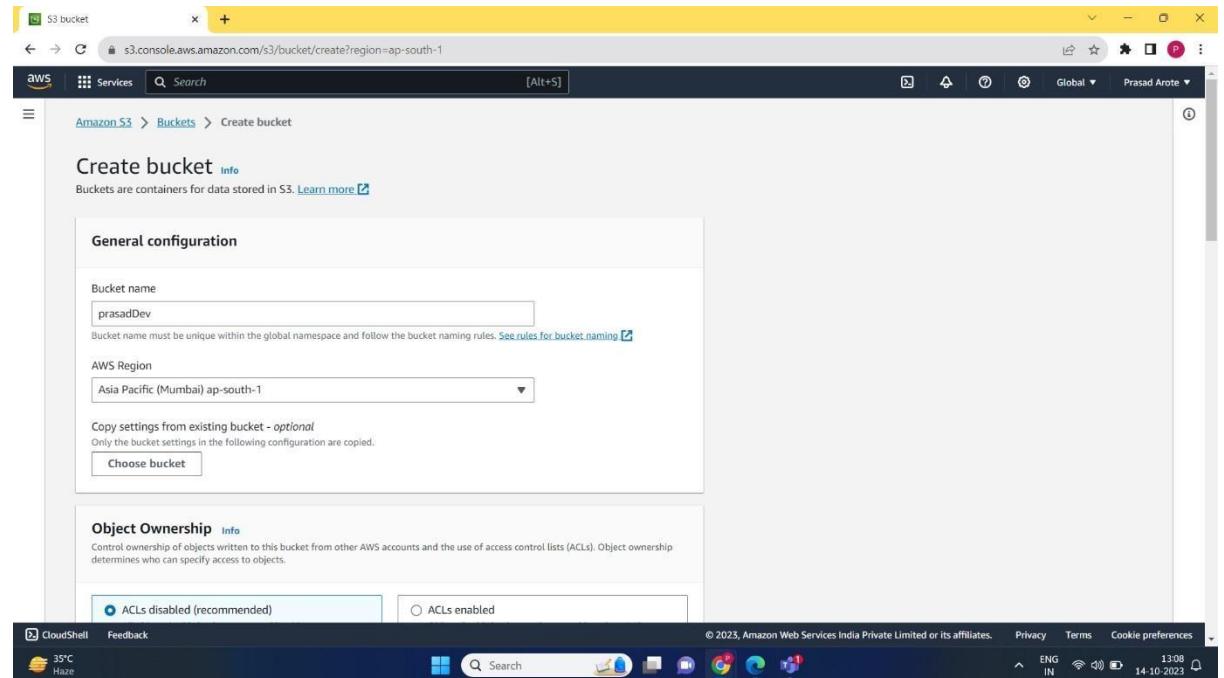


### Installation:

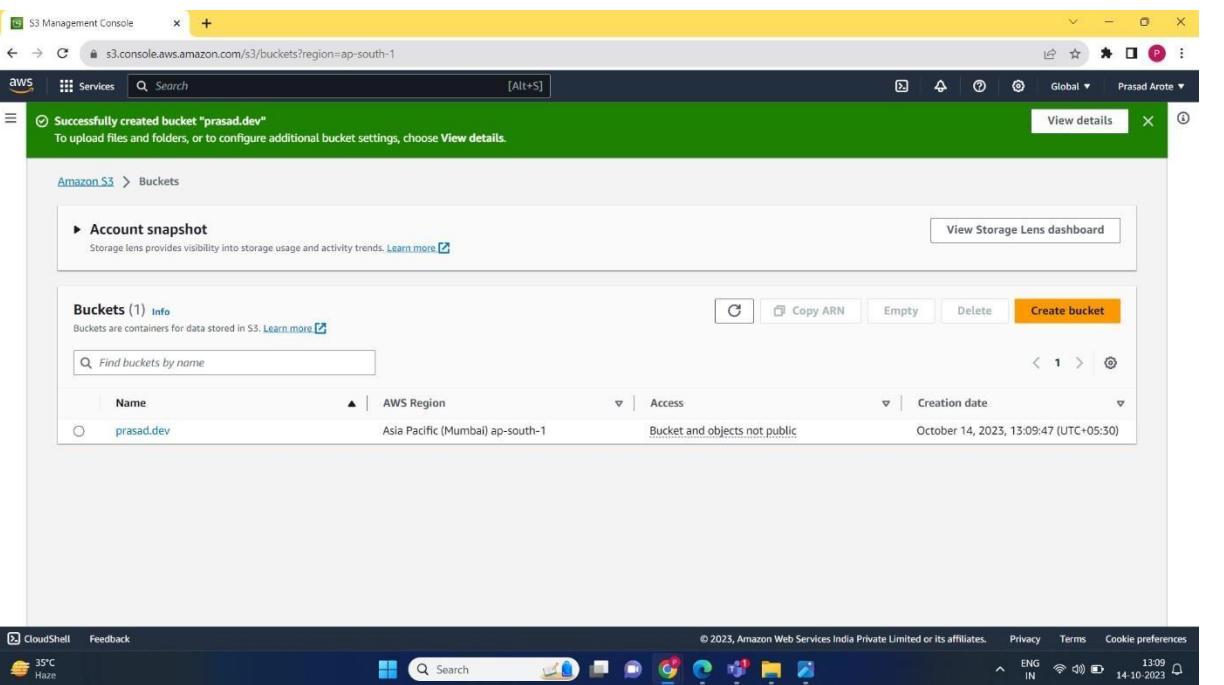
#### 1. Create a S3 bucket



## T11 ALTAF ALAM 02



The screenshot shows the 'Create bucket' page in the AWS S3 console. In the 'General configuration' section, the 'Bucket name' is set to 'prasadDev' and the 'AWS Region' is set to 'Asia Pacific (Mumbai) ap-south-1'. Under 'Object Ownership', 'ACLs disabled (recommended)' is selected. The status bar at the bottom indicates 'Successfully created bucket "prasad.dev"'.



The screenshot shows the 'Buckets' list in the AWS S3 Management Console. A green banner at the top confirms the successful creation of the bucket 'prasad.dev'. The table below lists the bucket details:

Name	AWS Region	Access	Creation date
prasad.dev	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	October 14, 2023, 13:09:47 (UTC+05:30)

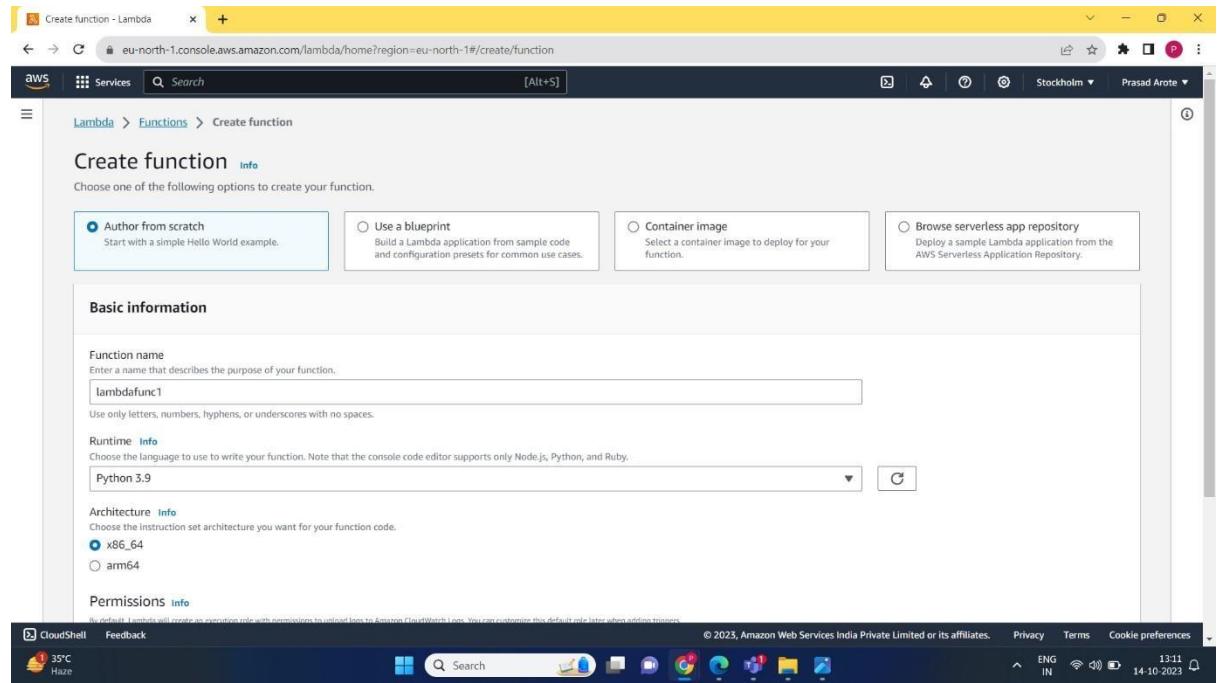
## 2. Create a Lambda function.

## T11 ALTAF ALAM 02

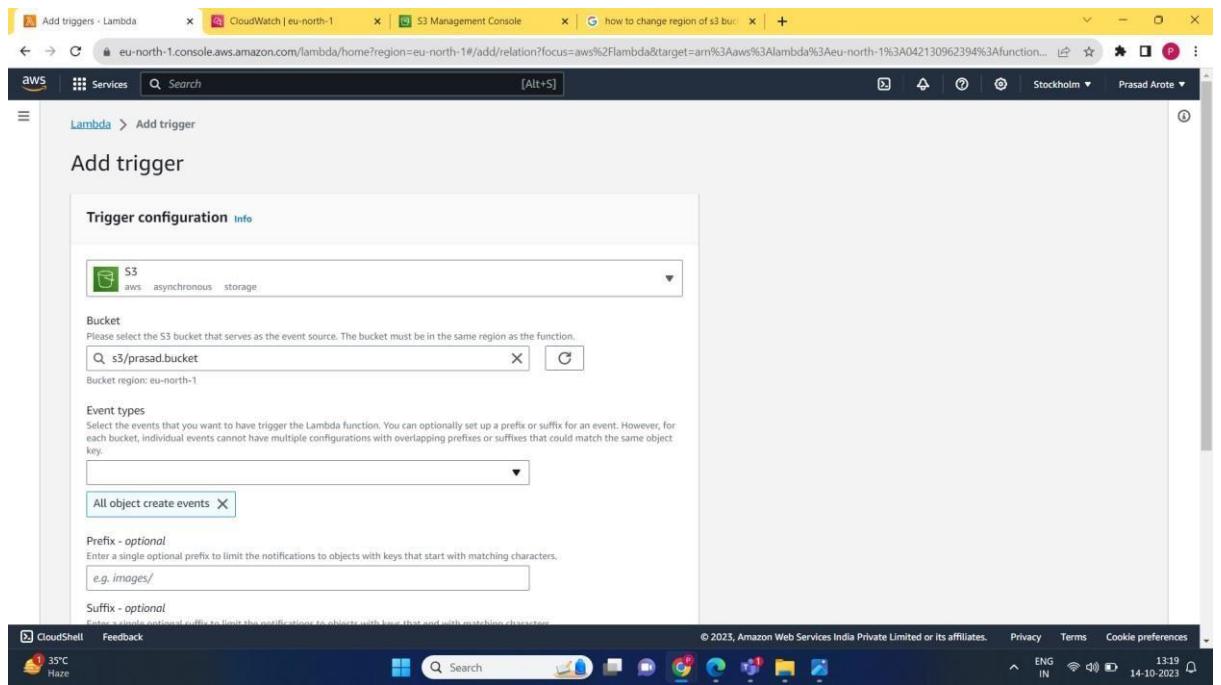
The screenshot shows the AWS S3 Management Console. A search bar at the top right contains the query 'lambda'. Below it, a sidebar lists services like Amazon S3, Features, Resources, and Buckets. The main content area displays search results under 'Services' and 'Features'. Under 'Services', 'Lambda' is listed with the description 'Run code without thinking about servers'. Other services shown include CodeBuild, AWS Signer, and Amazon Inspector. Under 'Features', 'Local processing' is listed as an IoT Core feature.

The screenshot shows the AWS Lambda console. A success message states 'Successfully created the function lambdafunc1. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' The function name 'lambdafunc1' is displayed. The 'Function overview' section shows the function icon, layers (0), and options to add triggers and destinations. On the right, there is a 'Description' panel with details like 'Last modified 3 seconds ago', 'Function ARN arn:aws:lambda:eu-north-1:042130962394:function:lambdafunc1', and 'Function URL'. Below the overview, tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions' are visible. The bottom navigation bar includes CloudShell, Feedback, and standard browser controls.

## T11 ALTAF ALAM 02



### 3. Create a trigger



## T11 ALTAF ALAM 02

The image consists of three vertically stacked screenshots from the AWS Lambda console.

**Screenshot 1: Adding a trigger for a Lambda function.**

This screenshot shows the "Add triggers - Lambda" page. It includes fields for "Event types" (set to "All object create events"), "Prefix - optional" (set to "e.g. images/"), and "Suffix - optional" (set to "e.g. jpg"). A checkbox for "Recursive invocation" is checked, with a note explaining it's not recommended. A note at the bottom states that Lambda will add necessary permissions for S3 to invoke the function. Buttons for "Cancel" and "Add" are at the bottom.

**Screenshot 2: Configuration of the lambdafunc1 function.**

This screenshot shows the "lambdafunc1 - Lambda" configuration page. It displays the "Function overview" section, which lists the trigger "prasad.bucket" successfully added to the function. The "Configuration" tab is selected. Other tabs include "Code", "Test", "Monitor", "Aliases", and "Versions". On the right, there are sections for "Description", "Last modified" (8 minutes ago), "Function ARN" (arn:aws:lambda:eu-north-1:042130962394:function:lambdafunc1), and "Function URL".

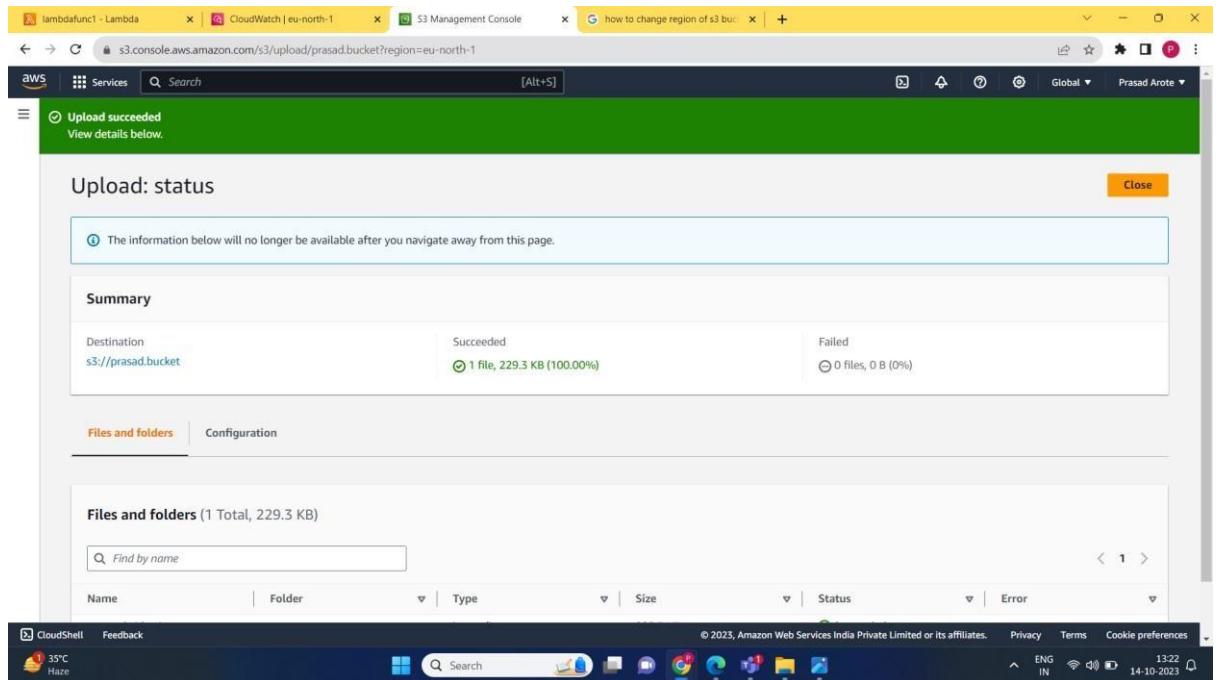
**Screenshot 3: Another view of the lambdafunc1 configuration.**

This screenshot shows a similar view of the lambdafunc1 configuration page, likely a different tab or a refresh. The "Configuration" tab is again selected, and the "Function overview" section shows the successful addition of the trigger. The right sidebar contains the same information as the previous screenshot.

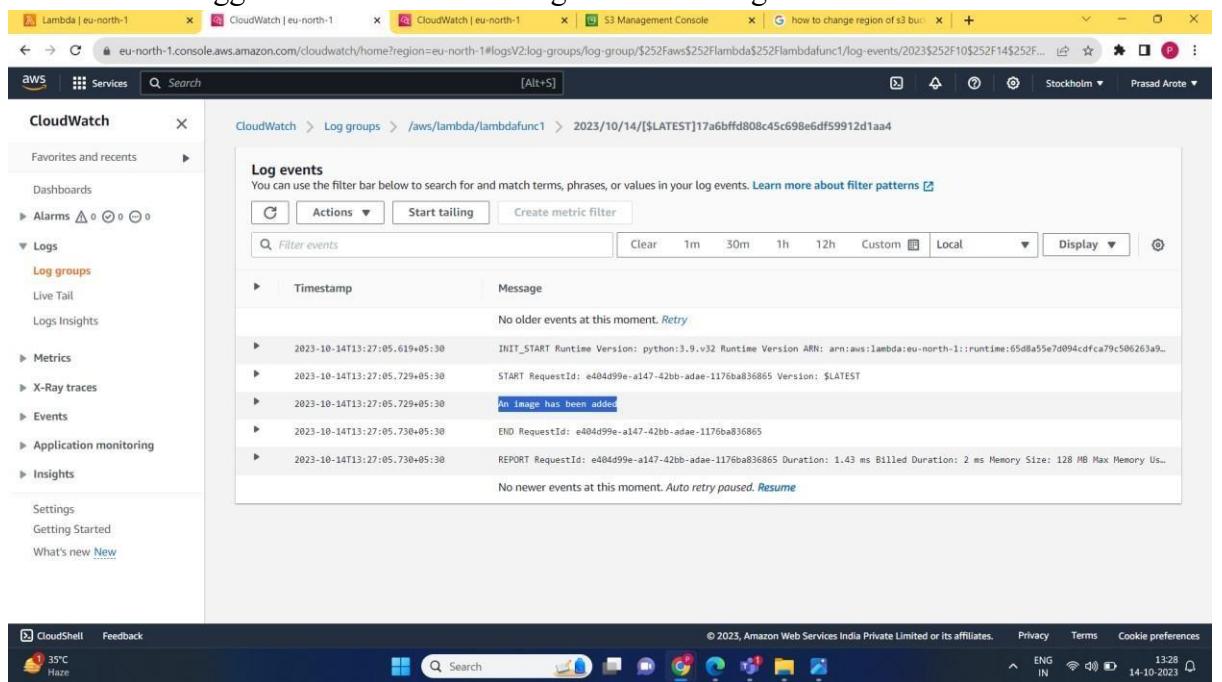
## T11 ALTAF ALAM 02

The screenshot shows the AWS S3 Management Console interface for uploading files to a bucket named 'prasad.bucket'. The top navigation bar includes tabs for Lambda, CloudWatch, and S3 Management Console. The main area displays the 'Upload' screen with a central message: 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a 'Files and folders (0)' section with a table header for Name, Folder, Type, and Size. A search bar and pagination controls are also present. The 'Destination' section shows the target bucket as 's3://prasad.bucket'. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.' The 'Permissions' section allows granting public access. The 'Properties' section lets users specify storage class, encryption settings, and tags. At the bottom right are 'Cancel' and 'Upload' buttons.

## T11 ALTAF ALAM 02



4. Thus we have triggered the function that logs when an image is added to S3 Bucket.



**Conclusion:** We have successfully created an lambda functions that logs when an image is added in S3 bucket.

T11 ALTAF ALAM 02

## Lab Assignment 8

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

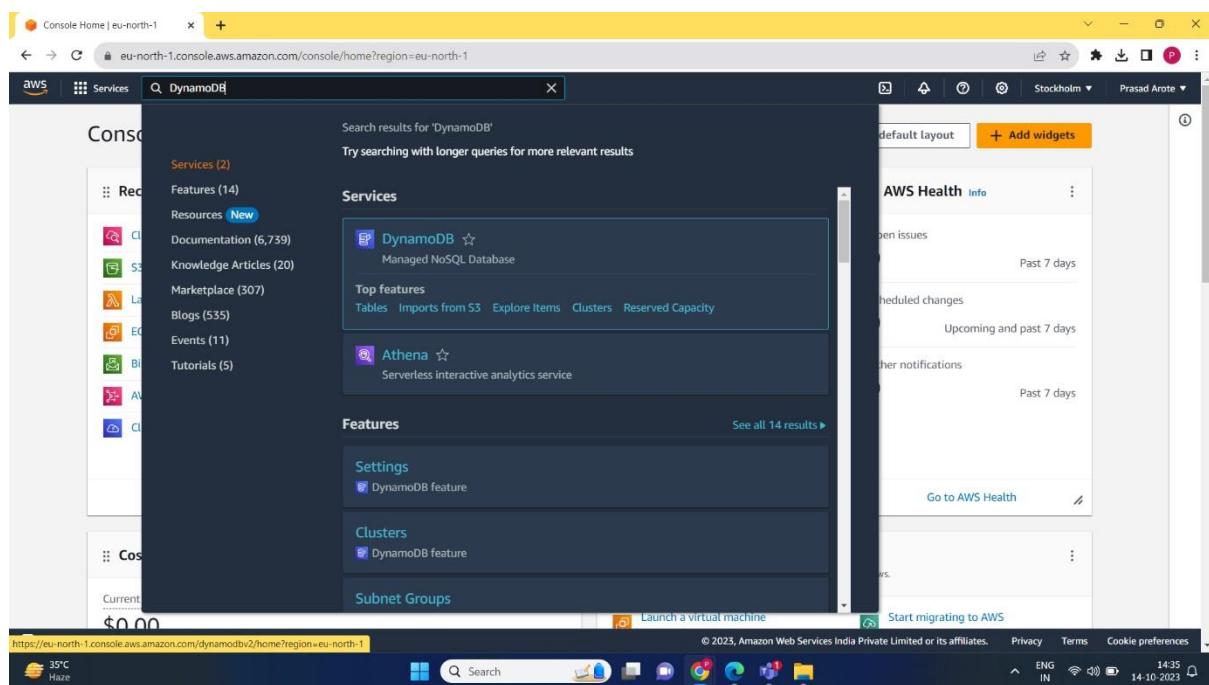
**LO6:** To engineer a composition of nano services using AWS Lambda and Step Functions with the serverless framework.

### 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.

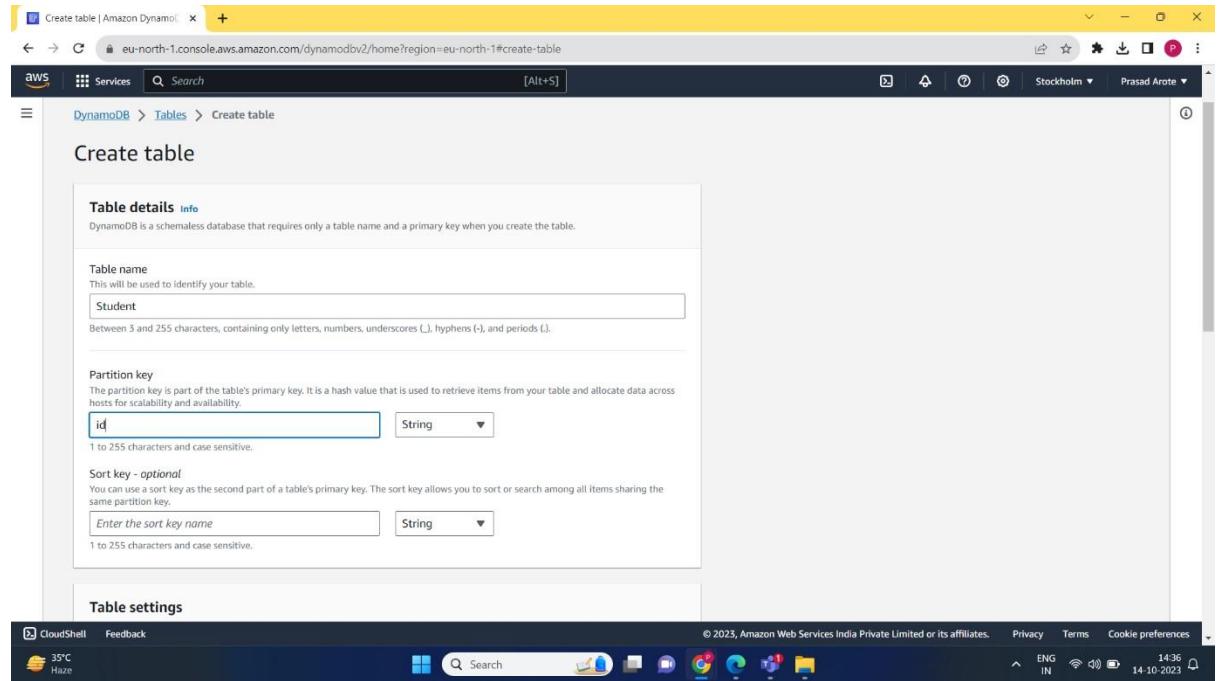
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource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



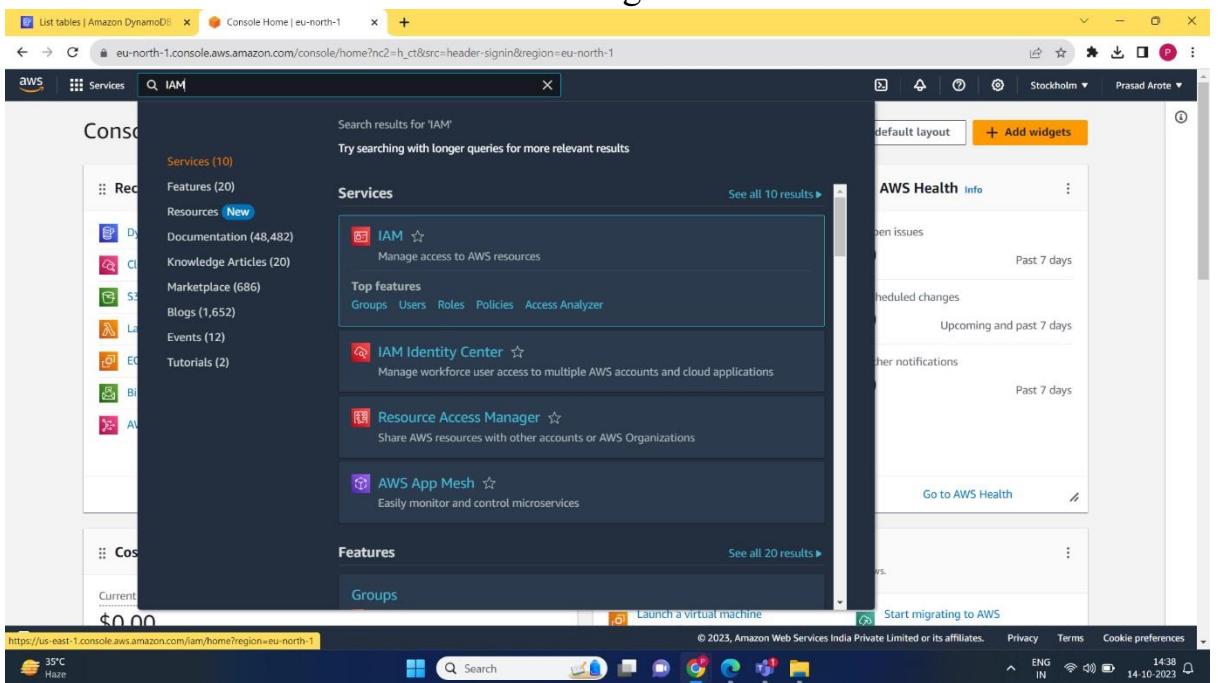
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

**STEPS:**

**1. Create a table**



**2. Create a role using IAM**



**3. Add permissions – AmazonDynamoFullAccess**

## T11 ALTAF ALAM 02

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

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.
- 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.

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Add permissions

Permissions policies (1/887)

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

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

## T11 ALTAF ALAM 02

The screenshot shows the AWS IAM console interface for creating a new role. The title bar indicates "List tables | Amazon DynamoDB" and "Create role | IAM | Global". The main area is titled "Role details" with the sub-section "Step 2: Add permissions". The "Role name" field contains "prasad\_admin". The "Description" field contains "Allows Lambda functions to call AWS services on your behalf.". Below this, the "Step 1: Select trusted entities" section is shown, with a "Trust policy" code block:

```
1 ~ [ {  
2 ~ "Version": "2012-10-17",  
3 ~ "Statement": [  
4 ~ {  
5 ~ "Effect": "allow",  
6 ~ "Action": [  
7 ~ "sts:AssumeRole"  
],  
8 ~ "Principal": [  
9 ~ {  
10 ~ "Service": [  
11 ~ "lambda.amazonaws.com"  
]  
}]  
}]
```

The status bar at the bottom shows "CloudShell Feedback", "35°C Haze", and the date "14-10-2023".

The screenshot shows the AWS IAM console interface with the title bar "List tables | Amazon DynamoDB" and "Roles | IAM | Global". The main area displays a list of roles under "Identity and Access Management (IAM)". A green banner at the top right says "Role prasad\_admin created." The "Create role" button is highlighted. The table lists the following roles:

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)	-
lambdafunc1-role-1t5lj6u	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

The status bar at the bottom shows "CloudShell Feedback", "35°C Haze", and the date "14-10-2023".

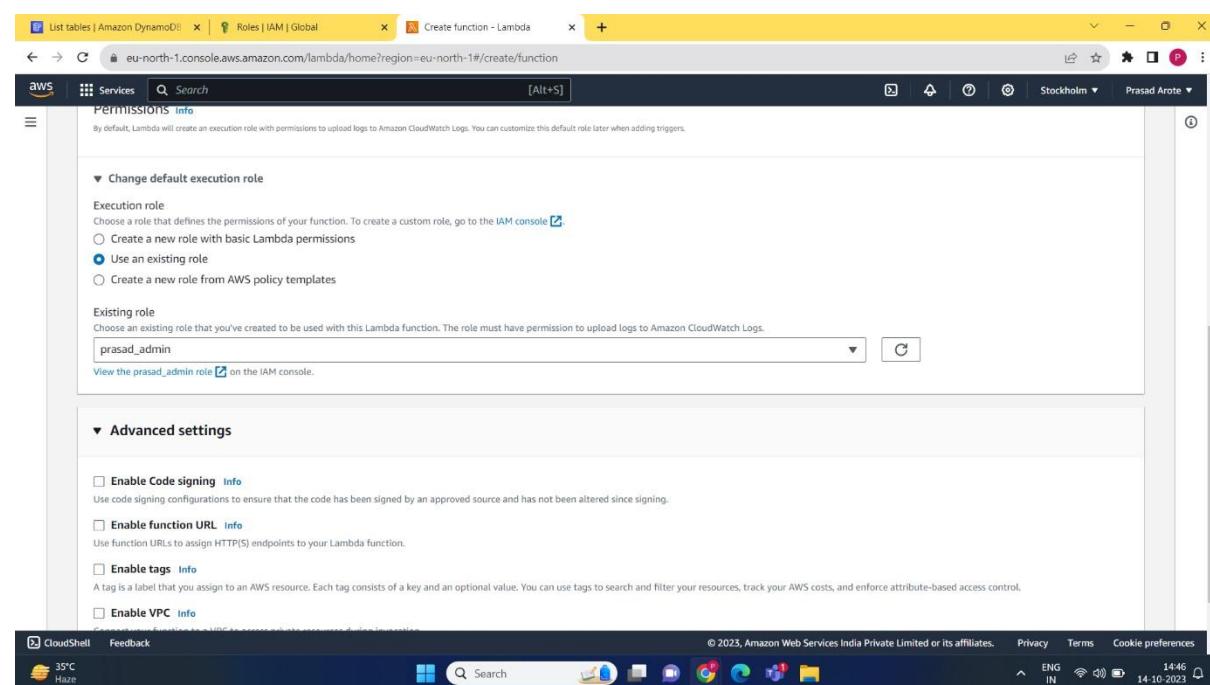
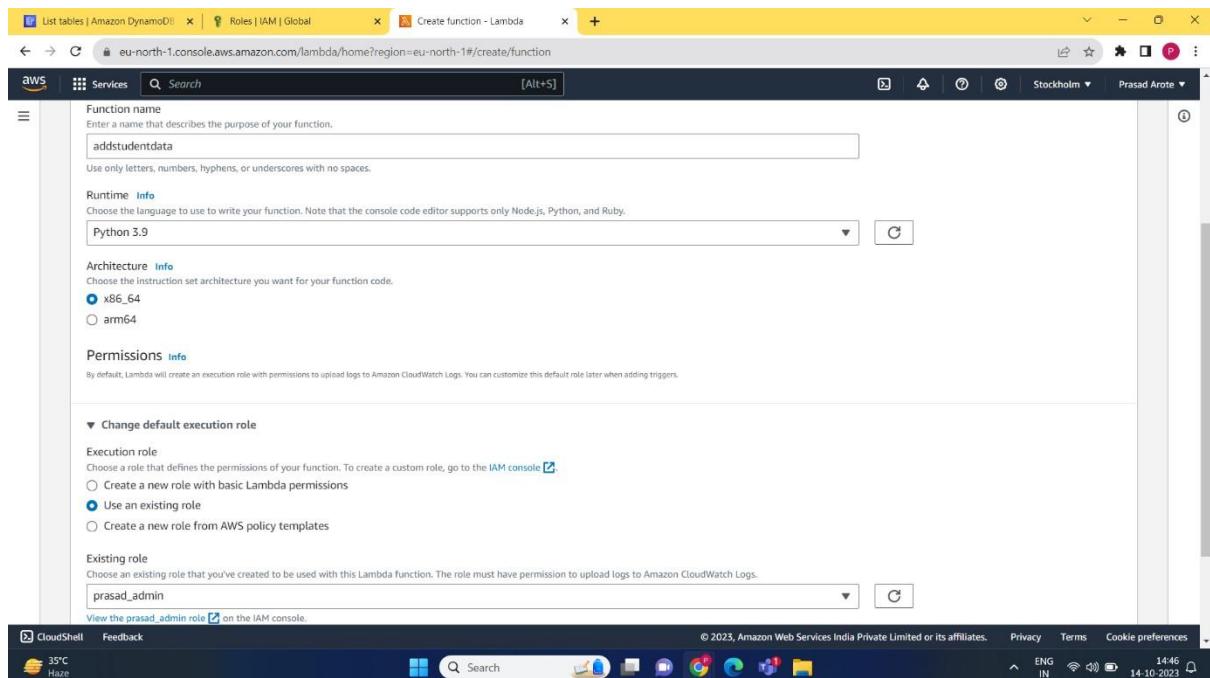
## 4. Create a Lambda Function

## T11 ALTAF ALAM 02

The screenshot shows the AWS Lambda search results page. The search bar at the top contains the query 'lambda'. Below the search bar, there is a message: 'Search results for 'lambda''. A note below it says 'Try searching with longer queries for more relevant results'. The main section is titled 'Services' and lists several services: Lambda, CodeBuild, AWS Signer, and Amazon Inspector. To the right of the services, there is a sidebar titled 'AWS Health Info' which displays various status metrics and links to 'Go to AWS Health'.

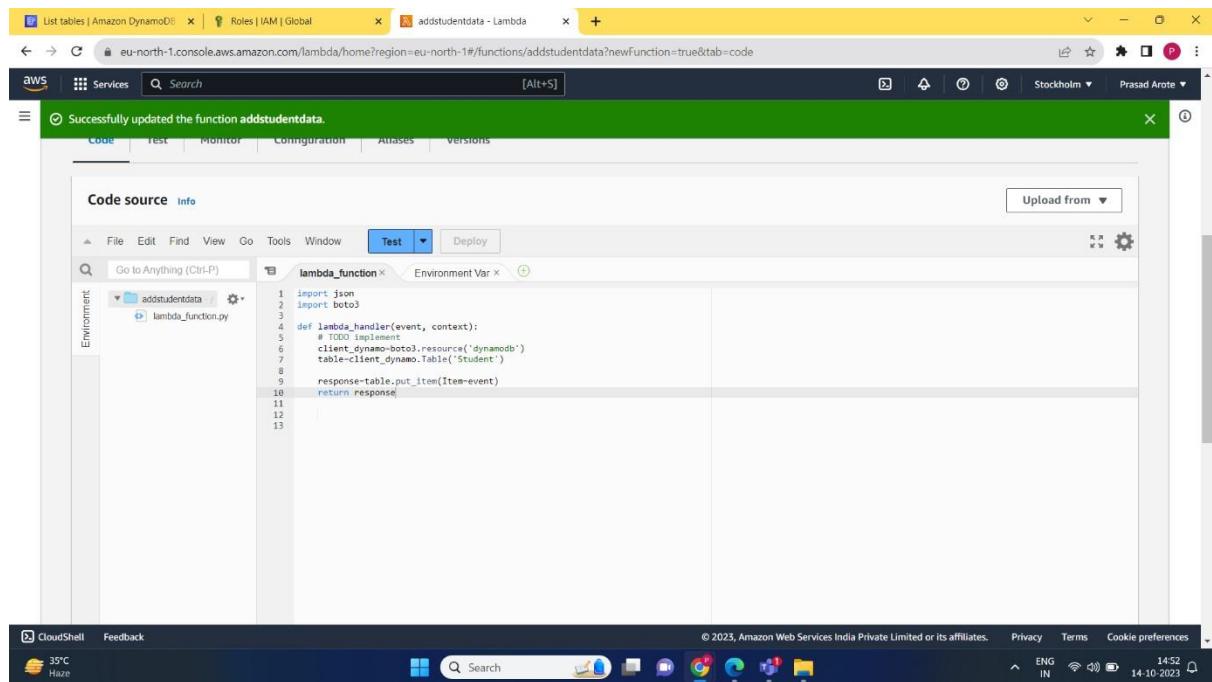
The screenshot shows the 'Create function' wizard. At the top, there are four options: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. Below these, the 'Basic information' section is visible, containing fields for 'Function name' (set to 'addstudentdata'), 'Runtime' (set to 'Python 3.9'), 'Architecture' (set to 'x86\_64'), and 'Permissions' (info link). The bottom of the screen shows the standard AWS navigation bar with links like CloudShell, Feedback, and the AWS logo.

## T11 ALTAF ALAM 02

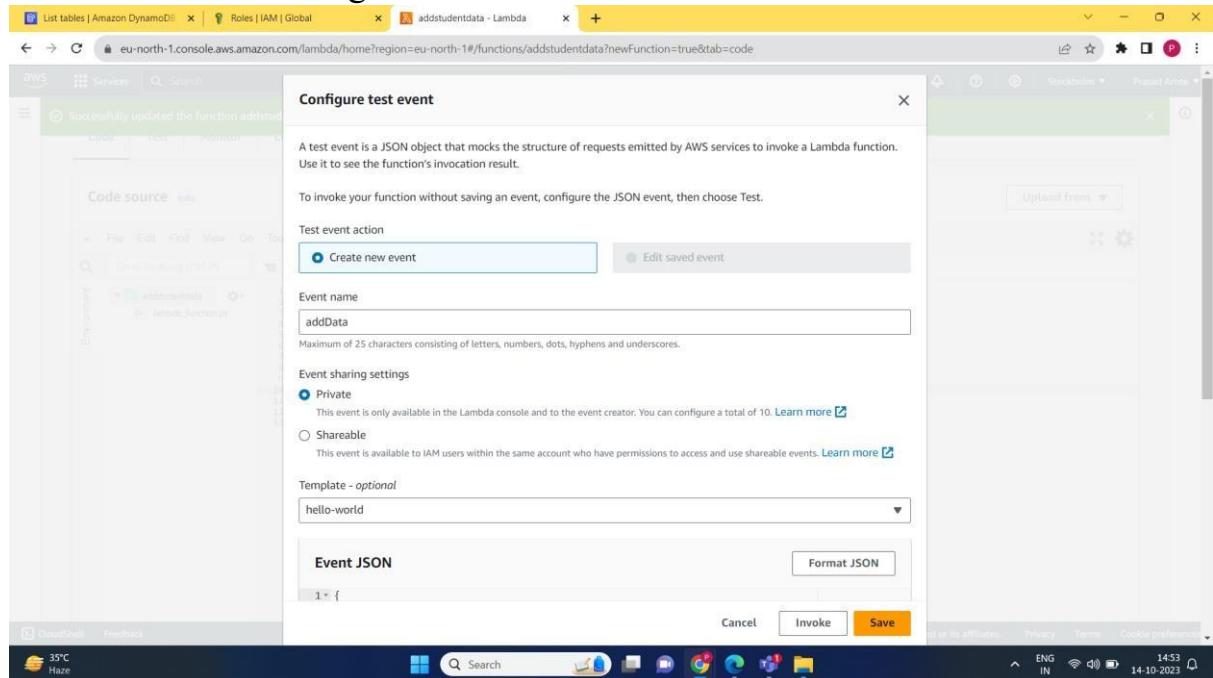


5. Write the following code

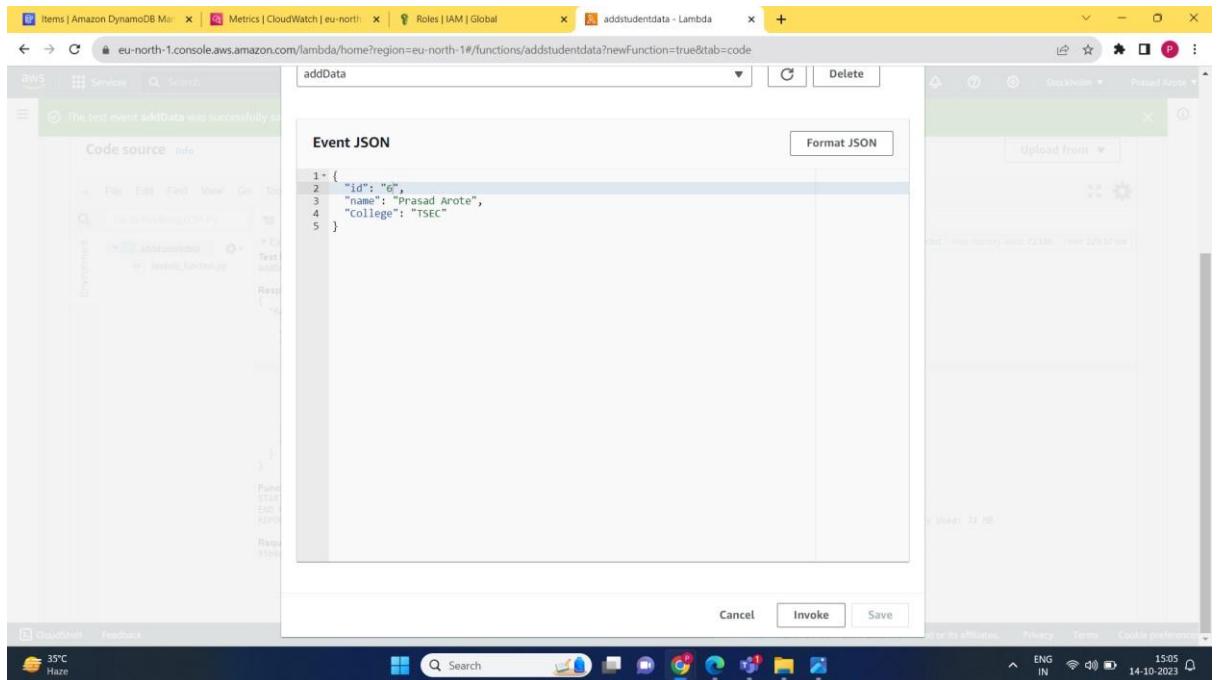
## T11 ALTAF ALAM 02



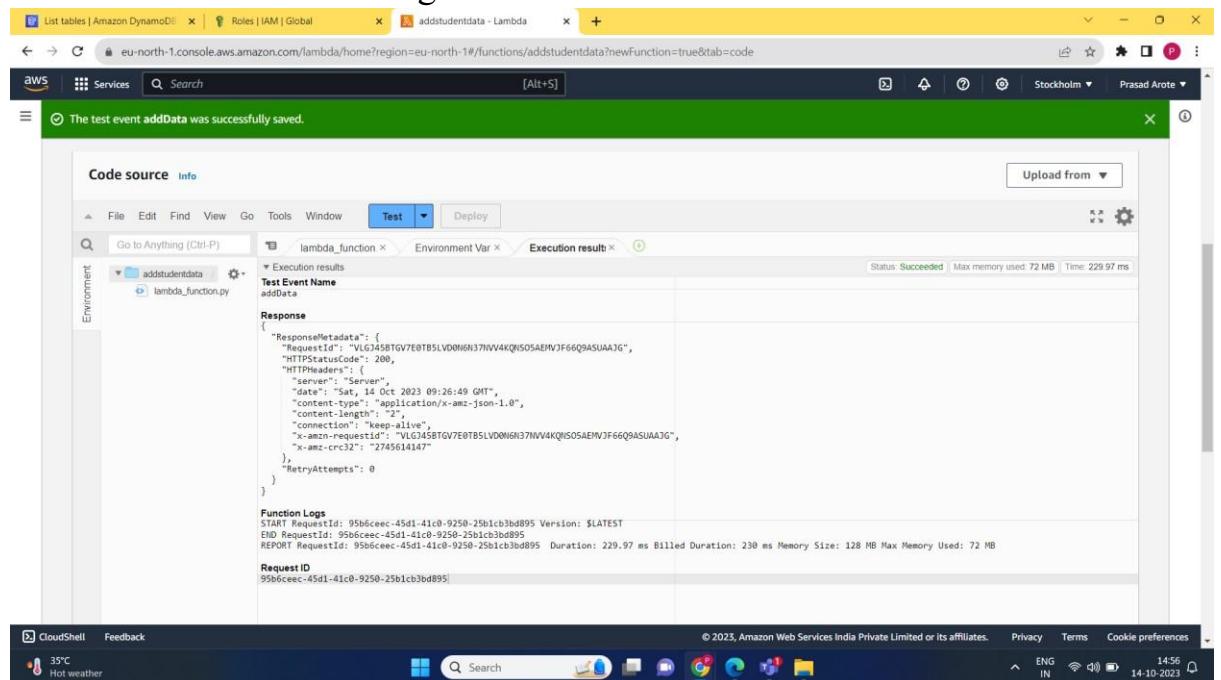
## 6. Configure test event and Save

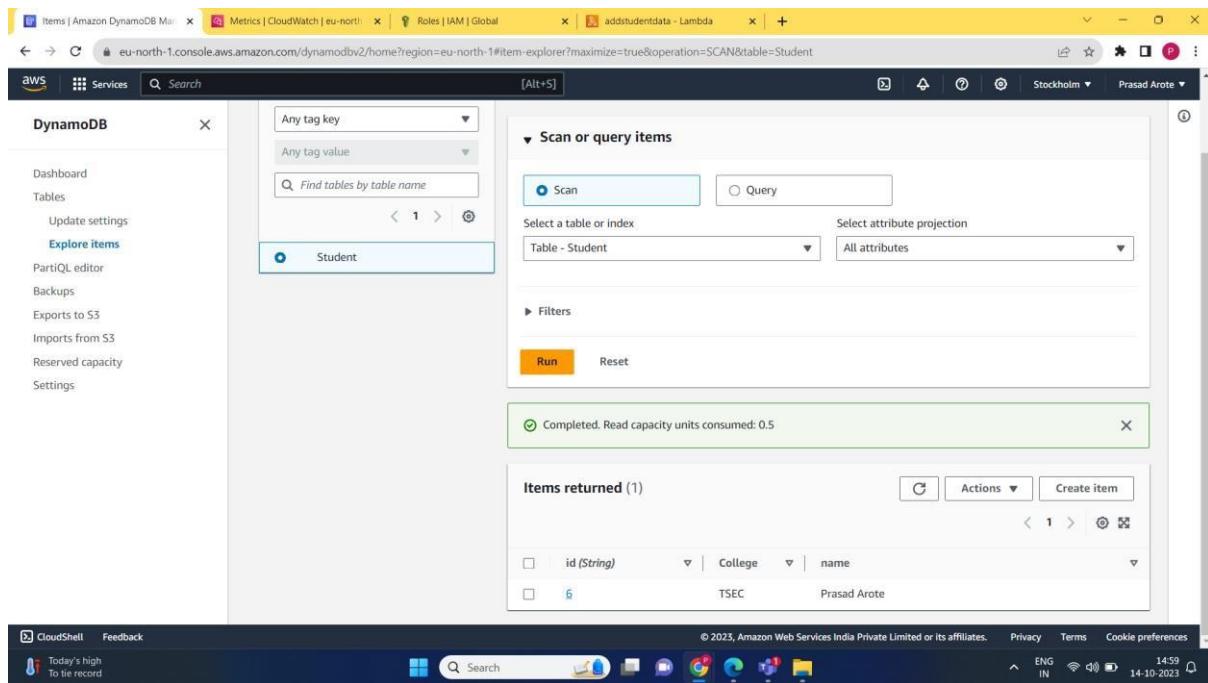


## T11 ALTAF ALAM 02



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





## CONCLUSION:

Thus, we have successfully inserted data in DynamoDB by using a Lambda function.

## Assignment No 9

**Aim:** To understand demo of Nagios and open-source tools.

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

**LO5:** To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

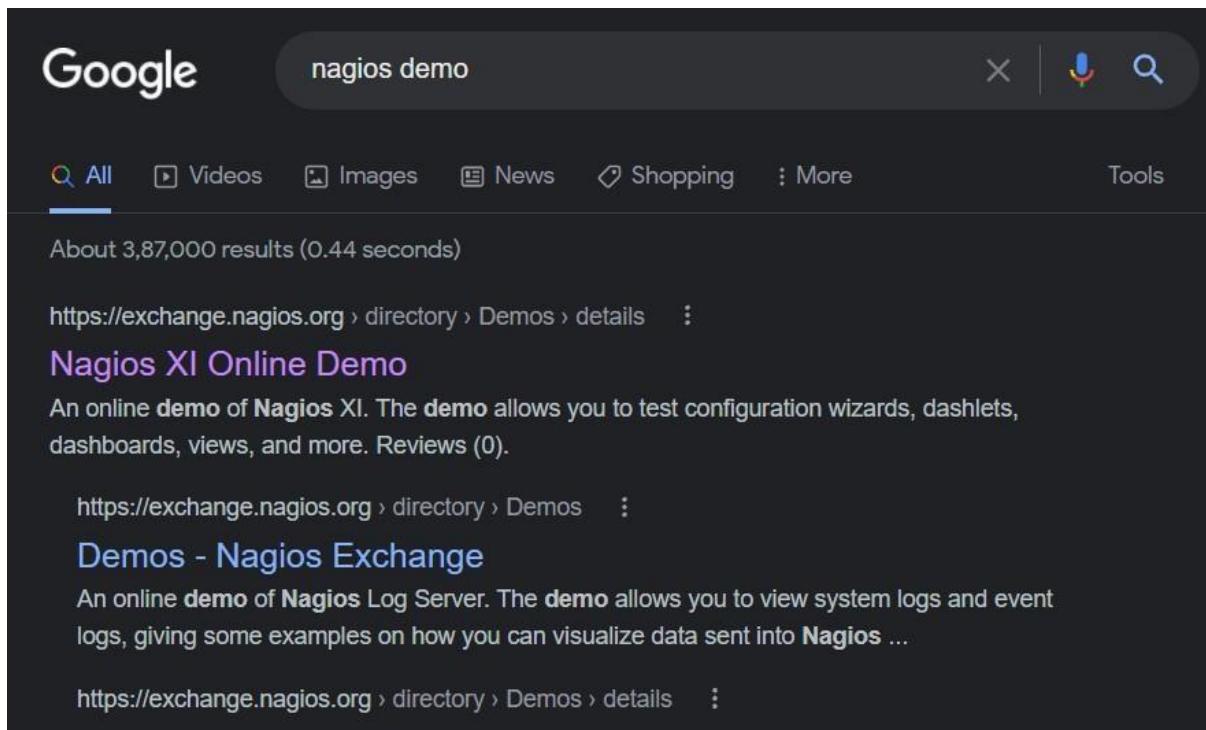
### Theory:

#### **What is Nagios and how it works?**

Nagios is an open source monitoring system for computer systems. ... Nagios software runs periodic checks on critical parameters of application, network and server resources. For example, Nagios can monitor memory usage, disk usage, microprocessor load, the number of currently running processes and log files.

### Steps-

#### **Go to google.com, Search Nagios Demo**



Now click on the website-

The screenshot shows the Nagios XI Online Demo page. At the top, there's a navigation bar with links for Network, Enterprise, Support, Library, Project, and Exchange. Below that is a secondary navigation bar with Home, Directory (which is highlighted), and About. The main content area has a breadcrumb trail: Home > Directory > Demos > Nagios XI Online Demo. On the left, there's a "Directory Tree" section. The central part features the title "Nagios XI Online Demo" in orange. Below it are buttons for Submit review, Recommend, Print, Visit, and Claim. A rating section shows 5 stars and 0 votes. It also displays the owner as "egalstad" and the website as "nagiosxi.demos.nagios.com". The hit count is listed as 141800. To the right, there are two search boxes: "Search Exchange" and "Search All Sites", both with "Go" buttons. A section for "Nagios Live Webinars" encourages users to learn how Nagios can help their organization.

Now click on login as administrator

The screenshot shows a browser window with the address bar showing "nagiosxi.demos.nagios.com/nagiosxi/login.php?redirect=/nagiosxi/index.php%3f&noauth=1". The main content is a "Login" form with fields for Username and Password, and a "Login" button. Below the form is a "Forgot your password?" link. To the right, there's a "Demo Account Options" section titled "Nagios XI Demo System". It lists five types of accounts with their respective log-in options and credentials:

- Administrator Access**: Log In as Administrator (Username: nagiosadmin, Password: nagiosadmin)
- Read-Only User Access**: Log In as Read-Only User (Username: readonly, Password: readonly)
- Advanced User Access**: Log In as Advanced User (Username: advanced, Password: advanced)
- Normal User Access**: Log In as Normal User (Username: jdoe, Password: jdoe)
- Administrator Access - showing the dark theme**: Log In as Administrator (Username: darktheme, Password: darktheme)

At the bottom, there's a "Demo Notes" section and a taskbar with various icons and system status information.

It will have interface like this

The screenshot shows the Nagios XI Home Dashboard. On the left, there's a sidebar with navigation links for Home Dashboard, Tactical Overview, Birdseye, Operations Center, Operations Screen, Open Service Problems, Open Host Problems, All Service Problems, All Host Problems, Network Outages, Details (Service Status, Host Status, Hostgroup Summary, Hostgroup Overview, Hostgroup Grid, Servicegroup Summary, Servicegroup Overview, Servicegroup Grid, BPI, Metrics), Graphs (Performance Graphs, Graph Explorer), and Maps (World Map, Bimap, Hypermap, Minemap, NagVis, Network Status Map). The main content area has several sections: 'Getting Started Guide' with common tasks like changing account settings and notifications; 'Host Status Summary' showing 132 Up, 1 Down, 1 Unreachable, and 0 Pending hosts; 'Service Status Summary' showing 1267 Ok, 29 Warning, 97 Unknown, 1 Critical, and 0 Pending services; 'Administrative Tasks' with initial setup tasks like configuring mail settings; and 'Start Monitoring' with options for Run a Config Wizard and Run Auto-Discovery. A 'We're Here To Help!' section features a photo of a support team member and links to Support Forum, Help Resources, Customer Ticket Support Center, and Customer Phone Support. At the bottom, it says 'Nagios XI 5.8.9 • Check for Updates' and shows system status: About | Legal | Copyright © 2008-2023 Nagios Enterprises, LLC, 11:14, 28-08-2023.

Now click on Host status-

The screenshot shows the Nagios XI Host Status page. The sidebar is identical to the Home Dashboard. The main content shows a table of hosts with columns: Host, Status, Duration, Attempt, Last Check, and Status Information. The table lists various hosts like 102.68.102.38, www.google.com, localhost, dns.google, bbqsnmp, 10.0.0.0, 9a34-181-162-10-91.sa.ngrok.io, pascalene.com, monster01, www.emot.cl, test, voip1.traci.net, trello.com, and siga.uem.mz. Each host row includes icons for status (green for up, red for down, yellow for warning), duration, attempt count, last check time, and detailed status information. The table shows 242 total records. At the bottom, it says 'Nagios XI 5.8.9 • Check for Updates' and shows system status: About | Legal | Copyright © 2008-2023 Nagios Enterprises, LLC, 11:09, 28-08-2023.

In the above image one can see Host Status Summary and Service Status

Summary also how many host are up, down and also errors in detail Now click on Host Group Status.

**Host Group Status**

Host Group	Hosts	Services
Monitoring Servers (Monitoring Servers)	7 Up	138 OK 5 Warning 1 Unknown 3 Critical
Hostgroup Two (hg2)	2 Up	41 OK 3 Warning 10 Unknown 3 Critical
Some Other Hostgroup (hg3)	2 Up	19 OK 1 Unknown 3 Warning
Linux Servers (linux-servers)	11 Up	292 OK 3 Warning 1 Unknown 3 Critical
Network Devices (network-devices)	8 Up	320 OK

**Service Status Summary**

Status	Count
Ok	1307
Warning	29
Unknown	37
Critical	3
Pending	0

Here we can see Status Summary for All Host Groups

Now we click on BBMap

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

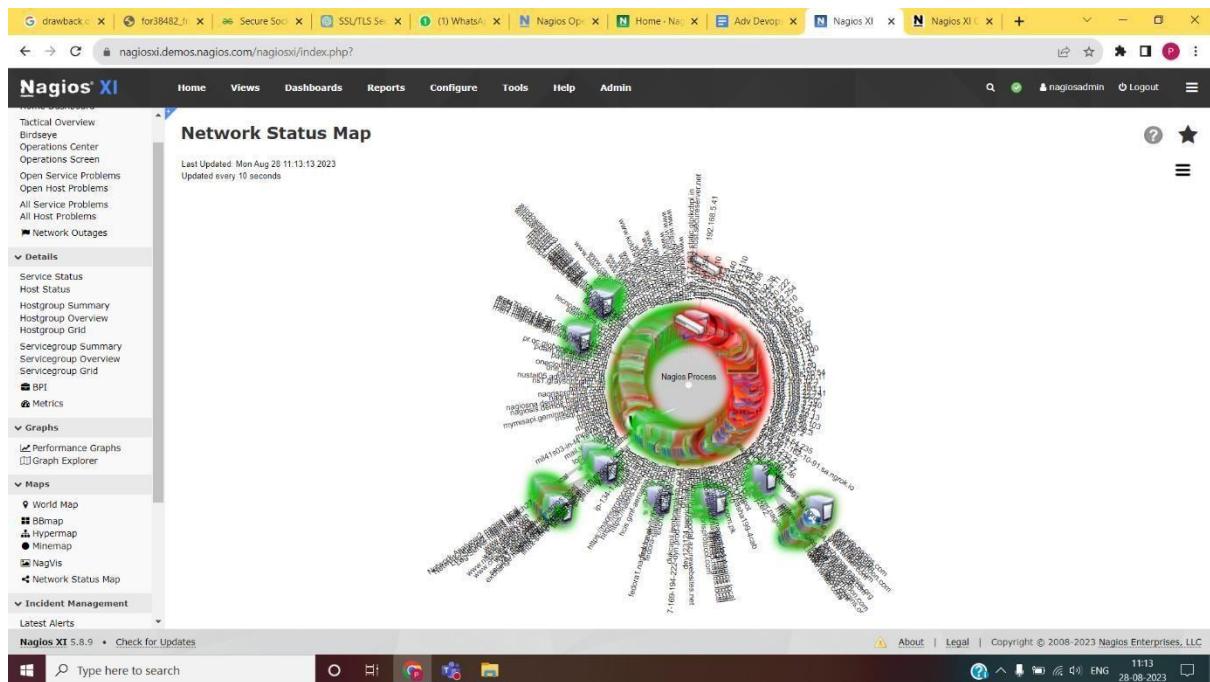
**BBMap**

The BBmap shows the current status of hosts/services in an icon grid layout. When a host has been acknowledged or is in downtime, the services will display semi-transparent. If a service is acknowledged or in downtime it will display the wrench icon.

**Status Grid**

Host	Service	Status
1.156.177.103.static.gtplckcbpl.in	/Disk Usage	OK
1.200.169.192.host.secureserver.net	/Boot Disk Usage	OK
10.0.0.0.0	404 Errors	OK
10.0.0.0.1	Active Connections	OK
10.0.0.254.254	Active Directory Server	OK
10.0.70.98	Apache 404 Errors	OK
10.10.1.7	Apache 404 Errors	OK
10.10.10.10	Apache Web Server	OK
10.10.2.34	Auroral Activity	OK
10.10.30.1	Argus/Geode/Statuary	OK
10.11.1.25	Argus/Geode/Statuary	OK
10.11.32.140	Apache/MySQL/PHP	OK
10.12.4.61	Bandwidth Spike	OK
10.159.129.110	Baldrick Status	OK
	CPU Credit Balance	OK
	CPU Credit Usage	OK
	CPU Stats	OK
	CPU Usage	OK
	CPU Usage for vHost	OK
	CPU Utilization	OK
	Copilot Planning - HOST	OK
	Crash_404 URLs Status	OK
	Cisco VPN Sessions	OK
	Cron Scheduling Daemon	OK
	Current Load	OK
	Current Users	OK
	DHCP	OK
	DNS IP Hatch - 3004.dae	OK
	DNS IP Hatch - 10th.com	OK
	DNS IP Hatch - ns1.alexa.org	OK
	DNS IP Hatch - pdm.snowball	OK
	DNS IP Hatch - trolio.com	OK

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



## CONCLUSION:

Hence, we understood Nagios. It is a powerful monitoring tool, provided valuable insights into its capabilities and benefits for effective system monitoring and management.

## Lab Assignment 10

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

**LO2.** To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes.

### THEORY:

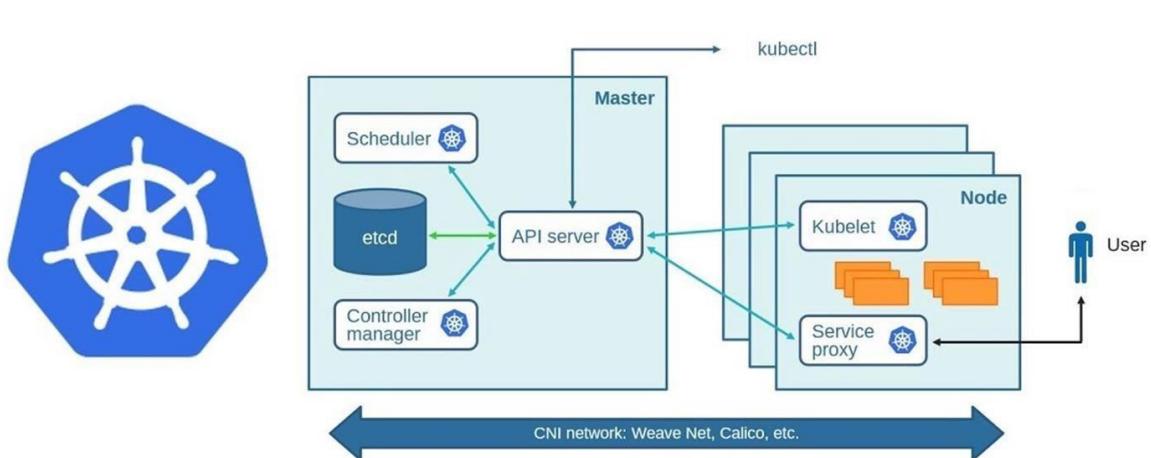
Kubernetes is an open-source container management tool that automates container deployment, scaling & load balancing.

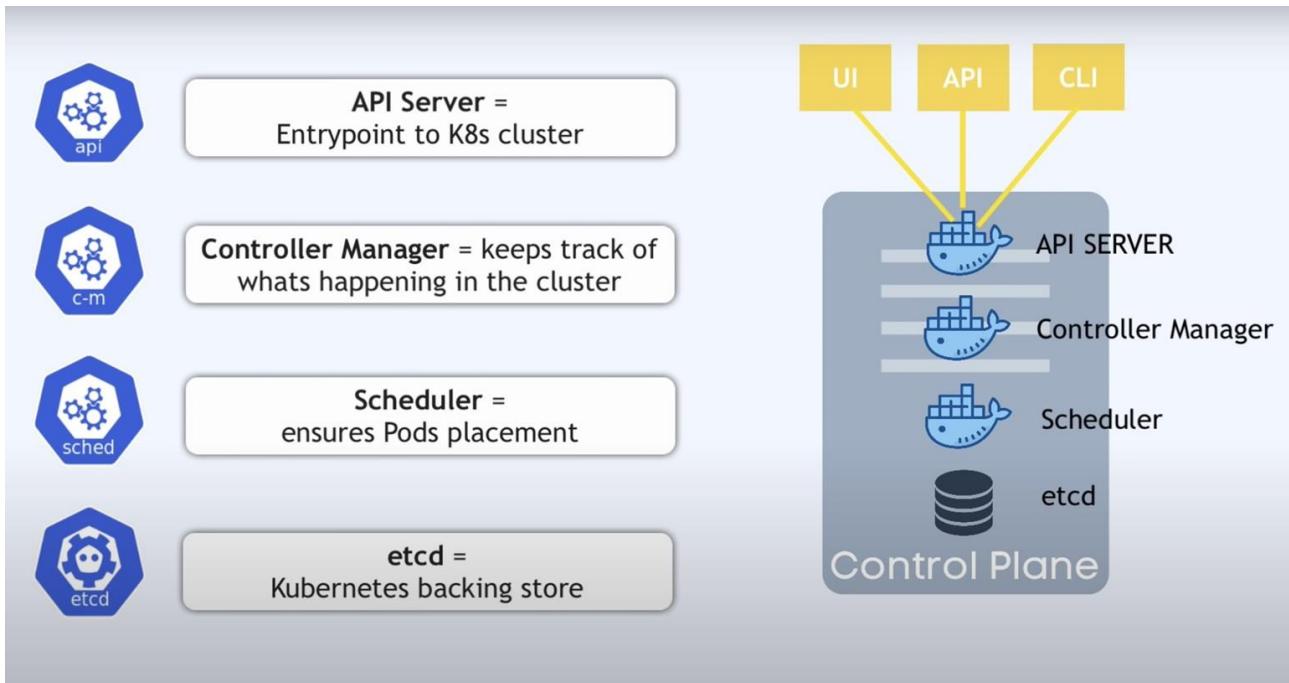
It schedules, runs, and manages isolated containers that are running on virtual/physical/cloud machines.

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

### ARCHITECTURE

# Kubernetes





## Working with Kubernetes

- We create a Manifest (.yml) file
- Apply those to cluster (to master) to bring it into the desired state.
- POD runs on a node, which is controlled by the master.

## ● Role of Master Node

- Kubernetes cluster contains containers running on Bare Metal / VM instances/cloud instances/ all mix.
- Kubernetes designates one or more of these as masters and all others as workers.
- The master is now going to run a set of K8s processes. These processes will ensure the smooth functioning of the cluster. These processes are called the 'Control Plane'.
- Can be Multi-Master for high availability.
- Master runs control plane to run cluster smoothly.

## ● Components of Control Plane

### ■ Kube-api-server → (For all communications)

- This api-server interacts directly with the user (i.e we apply .yml or .json manifest to kubeapi-server)
- This kube-api-server is meant to scale automatically as per load.
- Kube-api-server is the front end of the control plane.

### ■ etcd

- Stores metadata and status of the cluster.
- etcd is a consistent and high-available store (key-value-store)

- Source of truth for cluster state (info about the state of the cluster)

→ **etcd has the following features**

1. Fully Replicated → The entire state is available on every node in the cluster.
2. Secure → Implements automatic TLS with optional client-certificate authentication.
3. Fast → Benchmarked at 10,000 writes per second.

■ **Kube-scheduler (action)**

- When users request the creation & management of Pods, Kube-scheduler is going to take action on these requests.
- Handles POD creation and Management.
- Kube-scheduler match/assign any node to create and run pods.
- A scheduler watches for newly created pods that have no node assigned. For every pod that the scheduler discovers, the scheduler becomes responsible for finding the best node for that pod to run.
- The scheduler gets the information for hardware configuration from configuration files and schedules the Pods on nodes accordingly.

■ **Controller-Manager**

- Make sure the actual state of the cluster matches the desired state.

→ Two possible choices for controller manager—

1. If K8s is on the cloud, then it will be a cloud controller manager.
2. If K8s is on non-cloud, then it will be kube-controller-manager.

**Components on the master that runs the controller**

**Node Controller** → For checking the cloud provider to determine if a node has been detected in the cloud after it stops responding.

**Route-Controller** → Responsible for setting up a network, and routes on your cloud.

**Service-Controller** → Responsible for load Balancers on your cloud against services of type Load Balancer.

**Volume-Controller** → For creating, attaching, and mounting volumes and interacting with the cloud provider to orchestrate volume.

■ **Nodes (Kubelet and Container Engine)**

- Node is going to run 3 important pieces of software/process.

**Kubelet**

- The agent running on the node.
- Listens to Kubernetes master (eg- Pod creation request).
- Use port 10255.

T11 ALTAF ALAM 02

- Send success/Fail reports to master.

## Container Engine

- Works with kubelet
  - Pulling images
  - Start/Stop Containers
  - Exposing containers on ports specified in the manifest.

# Kube-Proxy

- Assign IP to each pod.
  - It is required to assign IP addresses to Pods (dynamic)
  - Kube-proxy runs on each node & this makes sure that each pod will get its unique IP Address.
  - These 3 components collectively consist of ‘node’.

## **INSTALLATION:**

## 1. Install Docker

T11 ALTAF ALAM 02

```
Activities Terminal Oct 14 22:11 prasad@prasad-VirtualBox: ~
update      Update configuration of one or more containers
walt       Block until one or more containers stop, then print their exit codes

Global Options:
--config string   Location of client config files (default "/home/prasad/.docker")
--context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug      Enable debug mode
-H, --host list  Daemon socket to connect to
-l, --log-level string Set the logging level (debug, info, warn, error, fatal) (default "info")
--tls            Use TLS; implied by -tlsverify
--tlscacert string Trust certs signed only by this CA (default "/home/prasad/.docker/ca.pem")
--tlscert string  Path to TLS certificate file (default "/home/prasad/.docker/cert.pem")
--tlskey string   Path to TLS key file (default "/home/prasad/.docker/key.pem")
--tlsverify      Use TLS and Verify the remote
-v, --version     Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/
prasad@prasad-VirtualBox: ~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
7f3a233935d4: Pull complete
Digest: sha256:88ec0aca9a3ec19dd5b7eaf7358bf458c25f9d3af59ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
  (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -t ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
prasad@prasad-VirtualBox: ~
```

2. Install minikube using following commands

## T11 ALTAF ALAM 02

The image shows two side-by-side terminal windows on a Linux desktop environment. Both terminals are running on a host system with the user 'prasad'. The top terminal window is titled 'Activities Terminal' and shows the command:

```
prasad@prasad-VirtualBox:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

Output of the curl command:

```
% Total % Received % Xferd Average Speed Time Time Current  
Dload Upload Total Spent Left Speed  
100 82.4K 100 82.4M 0 0 5315K 0 0:00:15 0:00:15 --:--:-- 5916K  
[sudo] password for prasad:  
[sudo] password for prasad:  
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)  
Using the docker driver based on user configuration  
💡 Exiting due to PROVIDER DOCKER NEWGRP: "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. Get "http://<2>%var%2Frun%2Fdocker.sock/v1.24/version": dial 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/
```

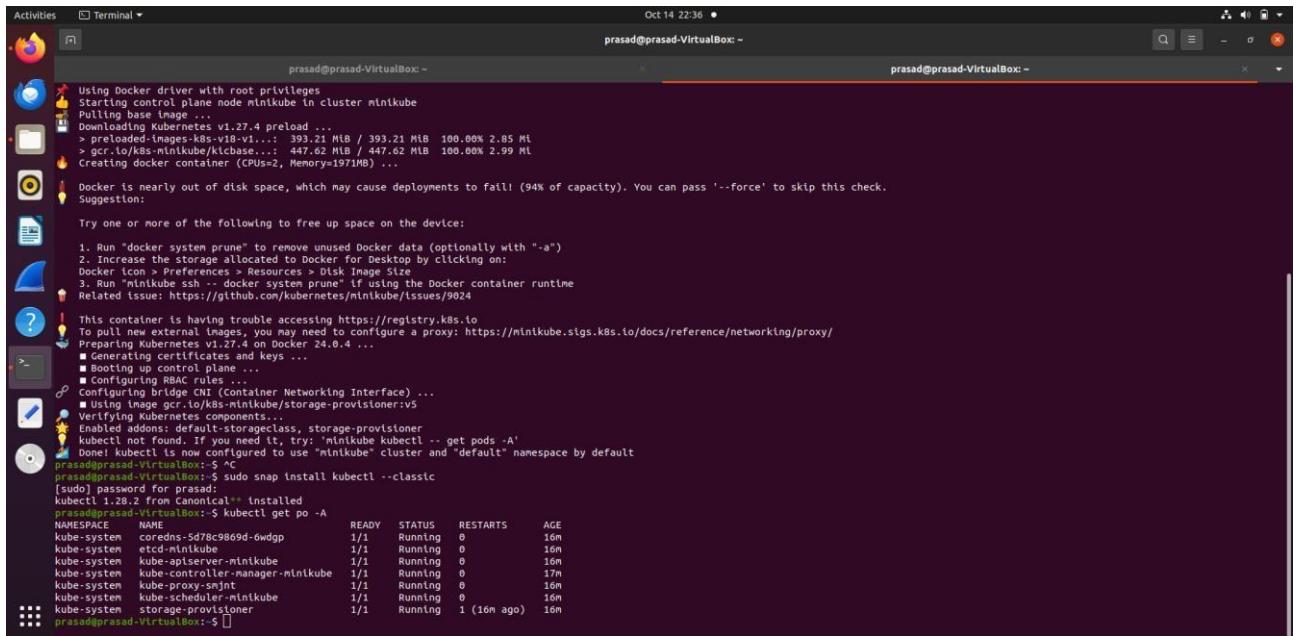
The bottom terminal window is also titled 'Activities Terminal' and shows the command:

```
prasad@prasad-VirtualBox:~$ sudo usermod -aG docker $USER && newgrp docker
```

Output of the sudo command:

```
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 MB / 447.62 MB 100.00% 2.99 MiB  
Creating docker container (CPUs=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.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) ...  
Using bridge.gcr.io/minikube/storage-provisioner:v5  
Verifying kubelet certificates configuration  
Enabled addons: default-storageclass, storage-provisioner  
kubelet 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
```

### 3. Install kubectl



```

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

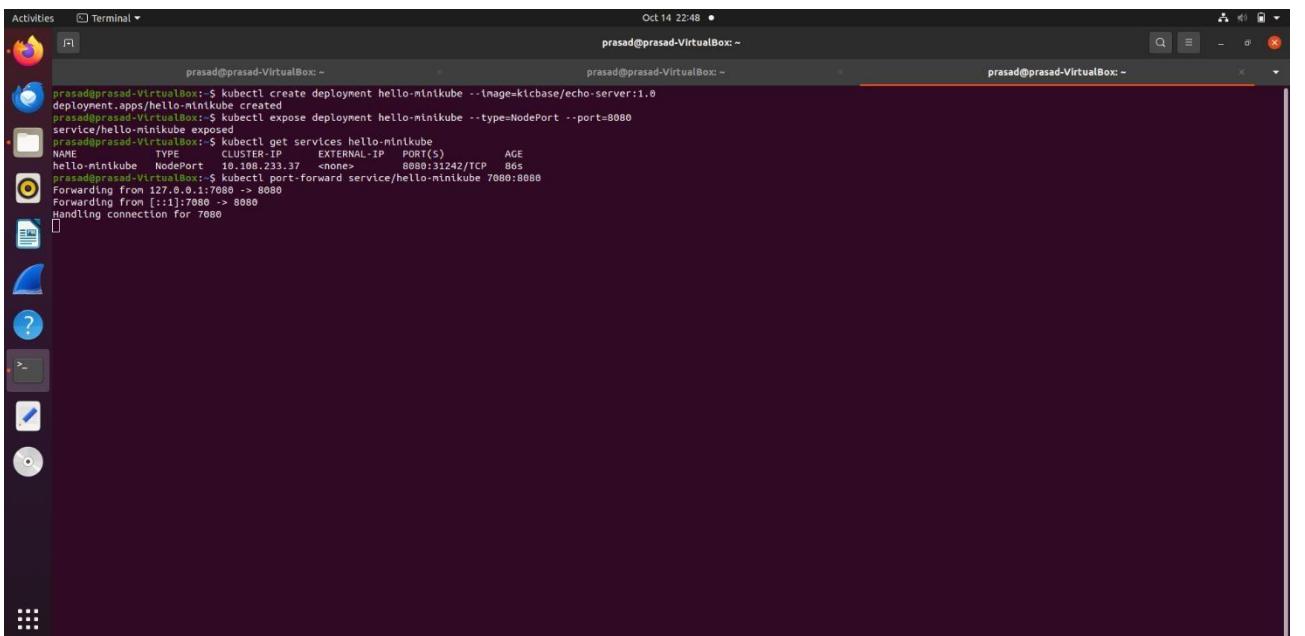
Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image
  Downloading kubernetes v1.27.4 preload ...
    > gcr.io/k8s-minikube/kichache...: 393.21 MiB / 393.21 MiB 100.00% 2.85 MiB
    > gcr.io/k8s-minikube/kichache...: 447.62 MiB / 447.62 MiB 100.00% 2.99 MiB
Creating docker container (CPUs=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.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 ...
  ○ Container networking interface (Container Networking 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'
prasad@prasad-VirtualBox: ~$ kubectl config set-cluster minikube
[sudo] password for prasad:
kubectl 1.28.2 from Canonical** installed
prasad@prasad-VirtualBox: ~$ kubectl get po -A
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-system coredns-5d78c9809d-6wdgp 1/1 Running 0 16m
kube-system etcd-minikube 1/1 Running 0 16m
kube-system kube-scheduler-minikube 1/1 Running 0 16m
kube-system kube-controller-manager-minikube 1/1 Running 0 17m
kube-system kube-proxy-smjnt 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 a sample deployment.

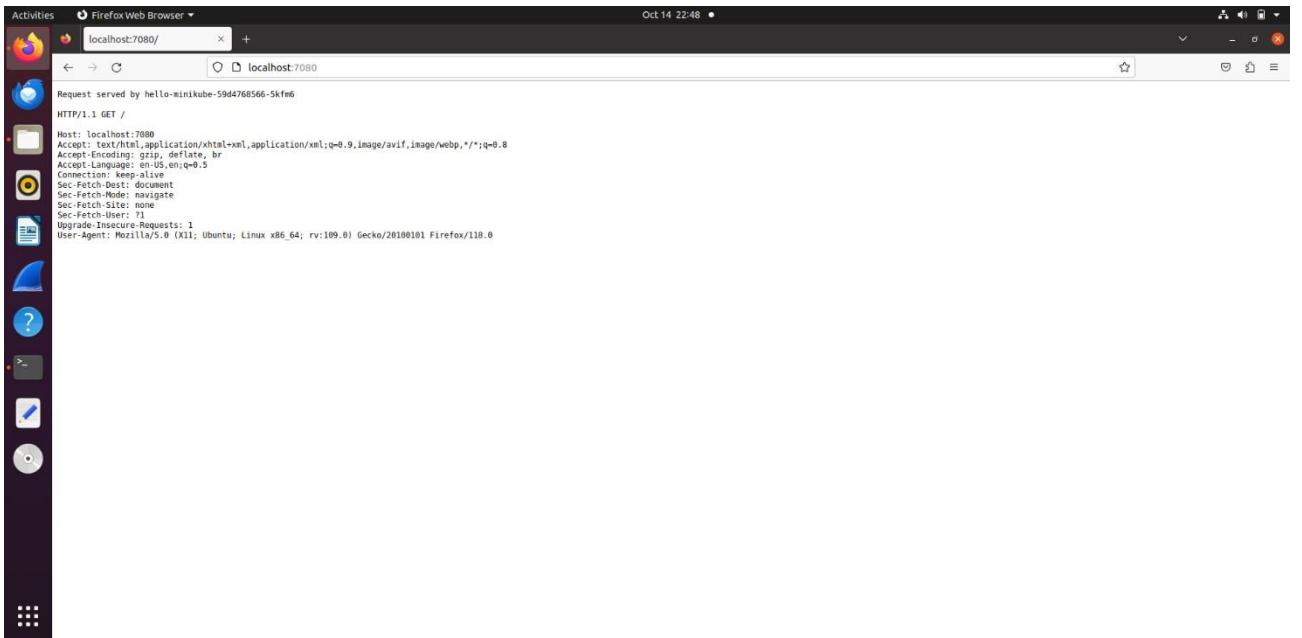


```

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

prasad@prasad-VirtualBox: ~$ kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
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 10.108.233.37 <none> 8080:31242/TCP 86s
prasad@prasad-VirtualBox: ~$ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from [::]:7080 -> 8080
Handling connection for [::]:7080

```



## CONCLUSION:

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

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

## **Written Assignment No 1**

### **Q.1 What are the best security measures that you can take while using Kubernetes?**

Kubernetes, a powerful container orchestration platform, has become increasingly popular for deploying and managing containerized applications. However, it's essential to ensure the security of your Kubernetes clusters. Here are some best security measures to consider:

1. Role-Based Access Control (RBAC): Implement RBAC to control who can perform actions within the cluster. Define roles and permissions to restrict access to critical resources, minimizing the risk of unauthorized actions.
2. API Server Access Control: Limit access to the Kubernetes API server by defining network policies and using technologies like firewalls and API gateways. Restricting API server access reduces the attack surface.
3. Pod Security Policies: Enforce security policies at the pod level to control actions like running as a privileged user, accessing the host network, or using a host's PID namespace.
4. Resource Quotas and Limit Ranges: Define resource quotas to limit the number of pods or resource consumption per namespace. Limit ranges can restrict resource requests and limits at a namespace level.
5. Container Security: Employ secure container images, regularly update them, and scan for vulnerabilities. Implement runtime protection mechanisms like AppArmor or SELinux to isolate containers.
6. Network Policies: Use network policies to control network traffic between pods. Limit communication to necessary ports and protocols, minimizing the risk of lateral movement in case of a breach.
7. Secrets Management: Use Kubernetes Secrets for sensitive data like API keys and passwords. Avoid storing secrets in plaintext in YAML files or in environment variables.
8. Pod Identity and Service Accounts: Utilize Service Accounts to manage access to Kubernetes resources. Implement Pod Identity to ensure that pods can access only the resources they need.
9. Audit Logging and Monitoring: Enable Kubernetes audit logging to track all API server requests. Use a robust monitoring and alerting system to detect and respond to security incidents.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

10. Security Updates: Stay vigilant about security updates for Kubernetes components, the host OS, and container runtimes. Regularly apply patches to address vulnerabilities.

11. Image Signing and Verification: Implement digital signatures and verification for container images to ensure that you're running authorized and unaltered software.

12. Network Segmentation: Isolate critical workloads in separate network segments or clusters to minimize the impact of potential breaches.

13. External Authentication Providers: Implement external authentication providers like LDAP or OIDC to enhance authentication and authorization mechanisms.

14. Backup and Disaster Recovery: Regularly back up your cluster data and configuration. Have a disaster recovery plan in place to ensure minimal downtime in case of an incident.

15. Employee Training: Educate your team about best security practices and conduct security drills to prepare for potential threats.

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

Protecting data is paramount in today's digital world, and there are several security techniques to safeguard sensitive information:

1. Encryption: Encryption is the process of converting data into a code to prevent unauthorized access. There are two main types:

- Data-at-rest Encryption: This encrypts data stored on disk or in databases, ensuring that even if physical storage devices are compromised, the data remains secure.
- Data-in-transit Encryption: This secures data as it's transmitted over networks. Techniques like SSL/TLS encrypt data during transmission, preventing eavesdropping.

2. Access Control: Access control is about controlling who can access data and what they can do with it. Role-based access control (RBAC) and attribute-based access control (ABAC) are common methods. RBAC defines roles and their permissions, while ABAC uses attributes of the user or data to make access decisions.

3. Data Masking and Redaction: Data masking involves hiding original data with fictional data while maintaining the format. Redaction, on the other hand, permanently removes or replaces sensitive data. These techniques ensure that sensitive information is not exposed to unauthorized users.

Each of these techniques plays a crucial role in data security, and combining them can provide a robust defense against data breaches and unauthorized access.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

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

In Kubernetes, you can expose a service using Ingress, which is an API object that manages external access to services within the cluster. Here are the steps to expose a service using Ingress:

1. Install an Ingress Controller: Before using Ingress, you need to have an Ingress controller running in your cluster. Popular choices include Nginx Ingress Controller and Traefik. Install the desired controller based on your requirements.
2. Define an Ingress Resource: Create an Ingress resource, which defines the rules for routing external traffic to your service. Here's an example of an Ingress resource:

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

In this example, the Ingress resource specifies that traffic with the hostname "example.com" and the path "/app" should be directed to the "my-service" service on port 80.

3. DNS Configuration: Ensure that the DNS record for the host specified in the Ingress resource (e.g., "example.com") points to the IP address of the Ingress controller's load balancer.
4. SSL/TLS Configuration (Optional): If you need to secure your Ingress with SSL/TLS, create a Kubernetes Secret containing the SSL certificate and key, and reference it in the Ingress resource.
5. Apply the Ingress Resource: Use the `kubectl apply -f` command to apply the Ingress resource to your cluster:

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

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

6. Verify and Test: After applying the Ingress resource, verify that it's working by accessing your service via the defined host and path (e.g., <http://example.com/app>). Test various paths and configurations to ensure proper routing.

Exposing a service using Ingress provides a flexible and powerful way to manage external access to your applications within a Kubernetes cluster.

#### **Q.4 Which service protocols does Kubernetes ingress expose?**

Kubernetes Ingress is designed to expose HTTP and HTTPS services. In other words, it primarily supports HTTP/HTTPS traffic routing. The Ingress resource uses rules and paths to route incoming HTTP requests to specific services and paths within the cluster.

To expose other service protocols, such as TCP or UDP, you would typically need to use different resources or controllers. For example:

- TCP and UDP Services: To expose services that rely on the TCP or UDP protocol, you can use a Service resource with the 'LoadBalancer' type or use a different controller designed for those protocols.
- WebSocket: If you need to support WebSocket connections, some Ingress controllers, like Nginx Ingress, have WebSocket support. You can specify WebSocket routes in your Ingress resource.

Keep in mind that while Kubernetes Ingress primarily focuses on HTTP/HTTPS traffic, it can be extended or complemented with other resources or controllers to support a broader range of service protocols.

## **Written Assignment No 1**

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

Deploying a Lambda function on AWS is a fundamental step in building serverless applications. AWS Lambda allows you to run code without provisioning or managing servers. Here's a high-level overview of how to deploy a Lambda function on AWS:

1. AWS Account Setup: Before you can deploy a Lambda function, you need an AWS account. If you don't have one, create an AWS account by visiting the AWS Management Console.
2. Create a Lambda Function: Access the AWS Lambda service through the AWS Management Console. Click on "Create Function" and choose one of the following ways to create your function:
  - Author from Scratch: Write your code directly in the Lambda function editor.
  - Use a Blueprint: Start with a predefined blueprint, which can be particularly useful for common use cases like data processing, image resizing, or API endpoints.
  - Upload a .zip File: Package your code and any dependencies into a .zip file and upload it.
  - Upload a .jar File: For Java-based Lambda functions, you can upload a .jar file.
3. Configure the Function: Set the function's name, runtime (e.g., Node.js, Python, Java, etc.), execution role (permissions), and other optional settings like environment variables and VPC configuration.
4. Write the Function Code: If you haven't already written your code, use the Lambda function editor to write your function's code. Ensure that your code follows the AWS Lambda function handler signature for your chosen runtime.
5. Test the Function: You can test the function within the Lambda console. Input test data to see how your function behaves. Make sure it produces the expected results.
6. Configure the Triggers: Lambda functions can be triggered by various AWS services (e.g., S3, API Gateway, SNS, etc.). You can configure triggers within the Lambda console, specifying how your function is invoked.
7. Deploy the Function: Once you are satisfied with your Lambda function's code, configuration, and testing, click the "Create Function" button to deploy it.
8. Monitoring and Logging: Use AWS CloudWatch to set up monitoring and logging for your Lambda function. This helps you track and troubleshoot the function's performance.

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

9. Integration and Scaling: Integrate your Lambda function with other AWS services as needed. Lambda automatically scales based on the incoming workload, ensuring that your application can handle varying levels of traffic.

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

AWS Lambda offers several deployment options to suit different development and deployment scenarios:

1. Console Deployment: You can deploy Lambda functions directly through the AWS Management Console by writing code in the Lambda editor or by uploading deployment packages.
2. AWS Command Line Interface (CLI): The AWS CLI allows you to create, update, and deploy Lambda functions using commands in your terminal. This is particularly useful for automating deployments and integrating Lambda functions into scripts or CI/CD pipelines.
3. AWS SDKs: AWS provides SDKs for various programming languages, such as Python, Node.js, Java, etc. These SDKs allow you to interact with AWS services, including Lambda, programmatically.
4. Serverless Framework: The Serverless Framework is an open-source framework that simplifies the deployment of serverless applications, including Lambda functions. It provides a declarative configuration and supports multiple runtimes.
5. AWS SAM (Serverless Application Model): AWS SAM is an open-source framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way of defining the Amazon API Gateway APIs, AWS Lambda functions, and Amazon DynamoDB tables needed by your serverless application.
6. Third-Party Tools: There are various third-party tools and platforms that offer integrations with AWS Lambda for deployment and management, such as AWS CodePipeline, Jenkins, and Travis CI.

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

AWS offers several deployment modes, and while you mentioned "full deployment modes," I'll discuss the three primary deployment modes in the context of AWS Lambda:

1. Full Copy Deployment (Blue/Green Deployment): In a full copy deployment, you create a complete, independent copy of your application environment, often referred to as a "blue"

Name : Altaf Alam

Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 19/10/23

environment. You then deploy the new version of your application, known as the "green" environment, alongside the existing version. After thorough testing and verification, you switch traffic from the blue environment to the green environment. This mode is often used for AWS Elastic Beanstalk, AWS OpsWorks, and EC2 deployments.

2. Rolling Deployment: In a rolling deployment, the new version of your application is deployed in stages across your existing infrastructure. This mode allows for a gradual rollout and provides the ability to roll back if issues are detected. It's a common approach for services like EC2 Auto Scaling groups.

3. Canary Deployment: A canary deployment is a phased deployment strategy in which the new version is released to a subset of users or traffic. This approach allows you to closely monitor the performance and stability of the new version and gradually expand the deployment if everything is functioning as expected. AWS CodeDeploy supports canary deployments.

These deployment modes are not specific to AWS Lambda but can be applied to various AWS services and application deployment scenarios.

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

AWS Lambda functions are composed of several components:

1. Function Code: This is the heart of the Lambda function. It contains the actual code that performs the desired functionality. The code is typically written in a supported runtime, such as Node.js, Python, Java, or others. You can package the code and any required dependencies into a deployment package, which is then uploaded to AWS Lambda.

2. Execution Role: An execution role is an AWS Identity and Access Management (IAM) role that grants permissions to the Lambda function. It defines what AWS resources the function can access and what actions it can perform. The execution role is crucial for interacting with other AWS services and resources securely.

3. Event Source: Lambda functions are typically triggered by specific events, such as changes to an S3 bucket, incoming data to an Amazon Kinesis stream, or API Gateway requests. The event source is responsible for invoking the Lambda function when these events occur. AWS services like S3, DynamoDB, SNS, and others can act as event sources.

These three components work together to define the behavior and execution of an AWS Lambda function. The function code specifies what the function does, the execution role defines what it can access, and the event source triggers the function when required events take place.