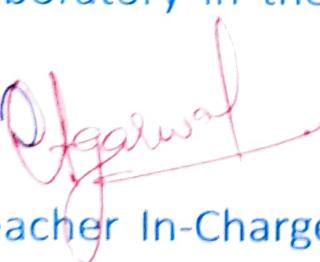


# Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

## © CERTIFICATE ©

Certify that Mr./Miss Aditya Naik  
of IT Department, Semester V with  
Roll No. 80 has completed a course of the necessary  
experiments in the subject Advance Devops under my  
supervision in the **Thadomal Shahani Engineering College**  
Laboratory in the year 2023 - 2024

  
Teacher In-Charge

Head of the Department

Date 21/10/23

Principal

## CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	To study and perform the setup of AWS EC2 service and launch an EC2 instance		19/07/23	
2.	To study and perform the setup of AWS cloud9 service and launch a python program in cloud9.		19/07/23	
3.	To study AWS s3 service, and to create a bucket for housing static web application.		26/07/23	
4.	To study AWS Cloud pipeline and deploy web application using cloud pipeline		27/07/23	<i>Agarwal 21/10/23</i>
5.	To understand Kubernetes cluster and architecture		14/10/23	
6.	To understand terraform life cycle and to build, change and destroy AWS using Terraform		23/09/23	
7.	To prepare static analysis for python program using SonarQube SAST process		13/09/23	
8.	To understand continuous testing using Nagios		02/08/23	
9.	To understand AWS Lambda functions and convert a Lambda function using python to log "an image that has been added" measure once		27/09/23	

## CONTENTS

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

### Theory:

Creating an Amazon Web Services (AWS) EC2 (Elastic Compute Cloud) instance involves several steps. EC2 instances are virtual machines that can run a variety of operating systems. Here are the basic steps to study and create an AWS EC2 instance:

#### 1. Sign in or create an AWS Account:

- If you don't already have an AWS account, go to the [AWS website](<https://aws.amazon.com/>) and sign up for an account.

#### 2. Access the AWS Management Console:

- Log in to the AWS Management Console using your credentials.

#### 3. Navigate to the EC2 Dashboard:

- In the AWS Management Console, go to the EC2 Dashboard. You can find it under the "Compute" section or search for "EC2."

#### 4. Choose an AWS Region:

- Select the AWS region where you want to launch your EC2 instance. AWS has regions worldwide; choose one closest to your target audience or for lower latency.

#### 5. Launch an EC2 Instance:

- Click on the "Instances" link on the EC2 Dashboard to create a new instance.

#### 6. Select an Amazon Machine Image (AMI):

- An AMI is a pre-configured image of an operating system. Choose an AMI that suits your needs, such as a Linux distribution, Windows Server, or a specific application image.

#### 7. Choose an Instance Type:

- Select the instance type that best fits your requirements in terms of CPU, memory, storage, and network performance. AWS offers various instance types optimized for different workloads.

#### 8. Configure Instance Details:

- Customize the instance settings, including the number of instances, VPC (Virtual Private Cloud), subnet, and other networking options.

#### 9. Add Storage:

- Specify the storage capacity for your instance. You can also add additional EBS (Elastic Block Store) volumes if needed.

#### 10. Add Tags:

- Assign tags to your EC2 instance to make it easier to organize and identify resources.

#### 11. Configure Security Groups:

- Security groups act as virtual firewalls for your instance. Define inbound and outbound rules to control traffic access.

#### 12. Review and Launch:

- Review your instance's configuration to ensure it's accurate. Then, click "Launch."

#### 13. Create a Key Pair:

- If you haven't created a key pair before, you'll be prompted to create one. This key pair is essential for securely connecting to your EC2 instance.

#### 14. Launch the Instance:

- Select the key pair you created and click "Launch Instances." This action will initiate the creation of your EC2 instance.

#### 15. View Instance Details:

- After your instance is launched, you can view its details, including its public IP address and other information, on the EC2 Dashboard.

#### 16. Connect to Your EC2 Instance:

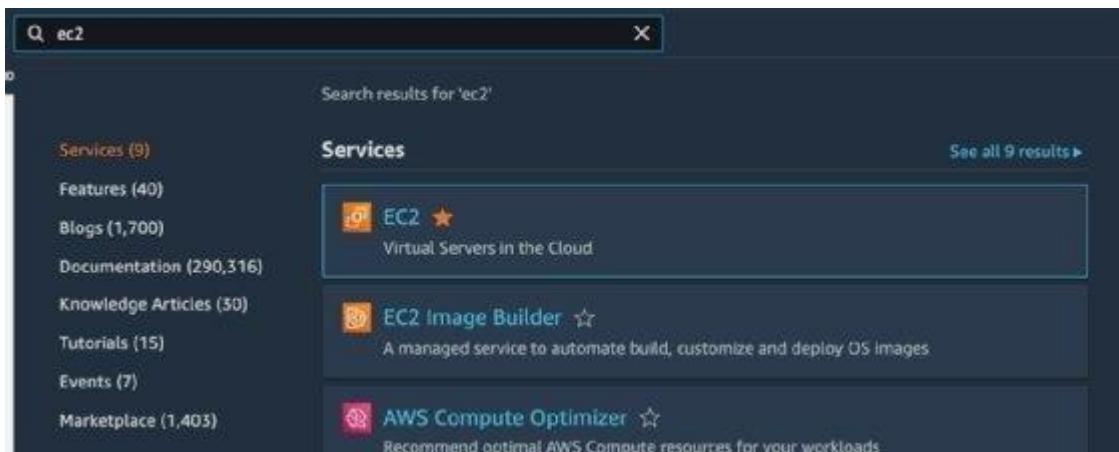
- You can connect to your EC2 instance using SSH (for Linux) or Remote Desktop (for Windows). Use the key pair you created to authenticate.

## 17. Configure Your Instance:

- Once connected, you can configure your EC2 instance according to your project or application requirements.

The screenshot shows the AWS EC2 'Launch an instance' wizard. The top navigation bar includes links for 'Launch an instance | EC2 Manager', 'Gmail', and 'YouTube'. The main content area is titled 'Launch an instance' with a sub-section 'Name and tags'. A green box highlights the 'Name' field, which contains the value 'myec2'. To the right of the field is a link 'Add additional tags'. Below this section is a collapsed panel titled 'Application and OS Images (Amazon Machine Image)'. At the bottom of the page is a search bar with the placeholder 'Search our full catalog including 1000s of application and OS images'.





**Conclusion:** These steps will help us create a basic AWS EC2 instance. It's important to understand the various instance types, AMIs, and configuration options to tailor your EC2 instance to your specific needs. Additionally, remember to manage your instances effectively, including stopping or terminating them when they are no longer needed to avoid unnecessary charges.

## Assignment 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 CREATE ENVIRONMNET,

## NAME THE ENVIRONMNET

**Details**

Name  
prasadCloud9

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

Description - optional

Limit 200 characters.

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

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

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

**New EC2 instance**

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

Now click on next step.

Cloud9 after creation.

**New EC2 instance**

Instance type **Info**  
The memory and CPU of the EC2 Instance that will be created for Cloud9 to run on.

t3.micro (1 GiB RAM + 2 vCPU)  
Free-tier eligible. Ideal for educational users and exploration.

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

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

Additional instance types  
Explore additional instances to fit your need.

Platform **Info**  
This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Amazon Linux 2

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

The screenshot shows the 'Network settings' section of the AWS Cloud9 configuration. It includes a 'Connection' dropdown set to 'AWS Systems Manager (SSM)', which allows access via SSM without opening inbound ports. There is also an option for 'Secure Shell (SSH)' which opens inbound ports. Other sections like 'Tags - optional' and 'VPC settings' are also visible.

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

Again, click on Next Step,  
Now click on Create Environment.

This screenshot shows the final step of creating an AWS Cloud9 environment. It displays a summary of IAM resources being created: 'AWSserviceRoleForAWSCloud9' and 'AWSCloud9SSMAccessRole' and 'AWSCloud9SSMInstanceProfile'. It includes a note that these can be deleted from the AWS IAM console if no longer needed. At the bottom are 'Cancel' and 'Create' buttons.

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

The screenshot shows the AWS Cloud9 interface. On the left, there's a sidebar with 'AWS Cloud9' at the top, followed by 'Environments' and 'Documentation'. The main area has a header 'AWS Cloud9 > Environments' with tabs for 'Environments (1)', 'Delete', 'View details', 'Open in Cloud9', and 'Create environment'. Below this is a search bar with 'My environments' and a dropdown arrow. A table lists one environment: 'prasadCloud9' (Name), 'Cloud9 IDE' (Cloud9 IDE), 'EC2 instance' (Environment type), 'AWS Systems Manager (SSM)' (Connection), 'Owner' (Permission), and 'arn:aws:iam:042130962394:root' (Owner ARN). At the bottom, there are links for 'CloudShell', 'Feedback', 'Language', and 'Cookie preferences'.

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
prasadCloud9	<a href="#">Open</a>	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:iam:042130962394:root

Now select any coding language and perform any operation via a code. Shown below

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes tabs for GitHub, AWS Cloud9, and prasadCloud9 - AWS Cloud9. The main workspace displays a terminal window with the command "bash -c ip-172-31-9-211.e" and an editor window titled "hello.c" containing the following code:

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

The terminal below shows the output "Hello World". At the bottom, the status bar indicates "AWS profile default" and "CodeWhisperer".

**Conclusion** – In conclusion, the use of AWS Cloud9 as an integrated development environment (IDE) has proven to be a highly efficient and flexible solution for our development needs. Throughout the course of this experiment, several key observations and benefits have become apparent.

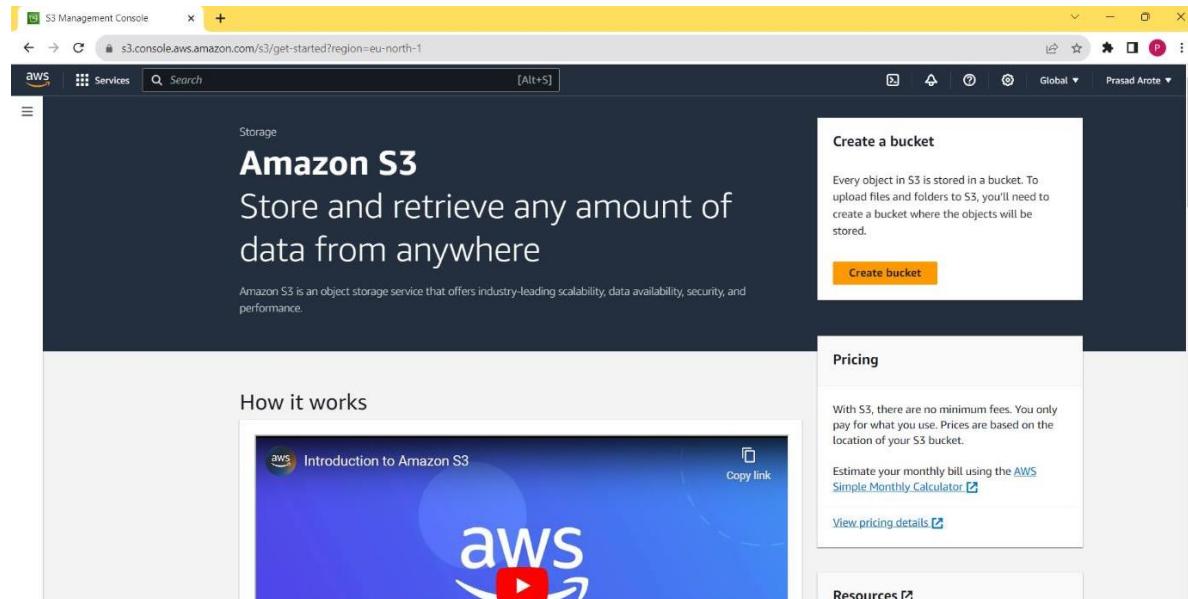
## LAB ASSIGNMENT 3

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

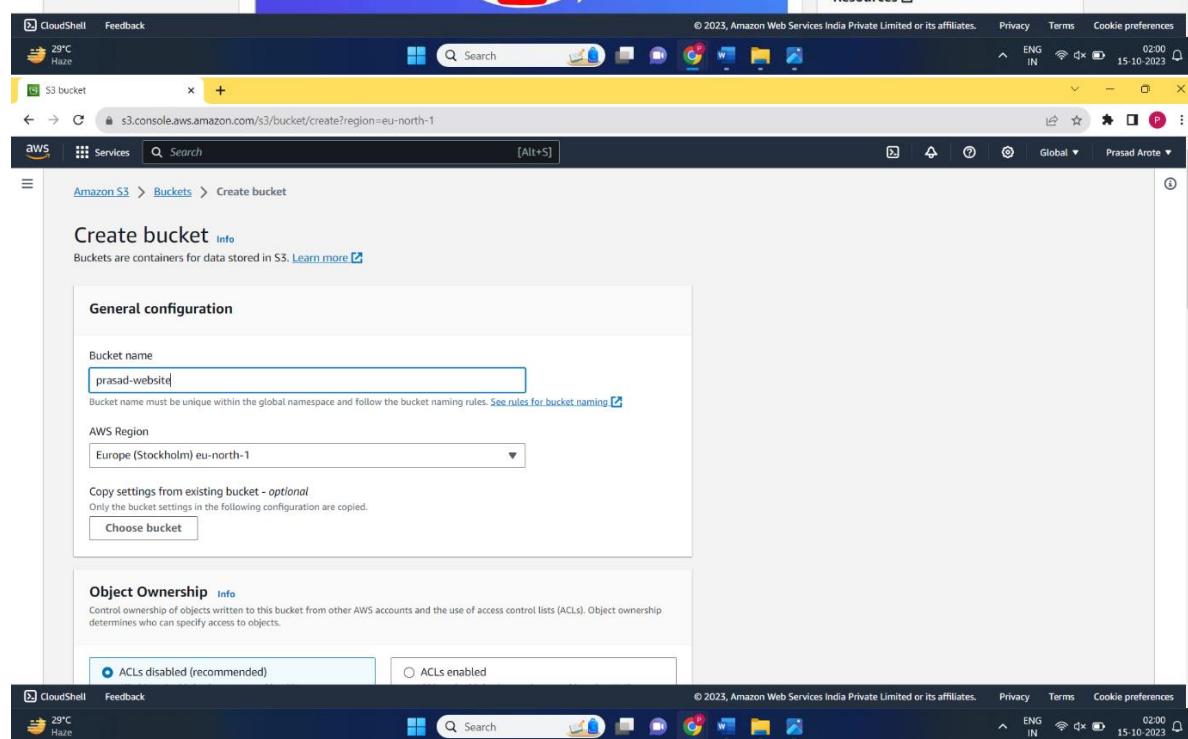
**LO1:** To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.

### **THEORY:**

#### 1. Create a S3 bucket.



The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with 'Services' and 'CloudShell'. The main area has a dark header with 'Amazon S3' and a sub-header 'Store and retrieve any amount of data from anywhere'. Below this, a paragraph explains that Amazon S3 is an object storage service. To the right, a large white box titled 'Create a bucket' contains instructions about buckets and a prominent orange 'Create bucket' button. Below this, a 'Pricing' section discusses costs and provides links to calculators and details. At the bottom, a 'Resources' section is partially visible.

The screenshot shows the 'Create bucket' configuration page. The URL is <https://s3.console.aws.amazon.com/s3/bucket/create?region=eu-north-1>. The page has a header with 'CloudShell', 'Feedback', and the user 'Prasad Arote'. It shows the navigation path 'Amazon S3 > Buckets > Create bucket'. The main form is titled 'Create bucket' with an 'Info' link. It asks for a 'Bucket name' (input: 'prasad-website'), 'AWS Region' (selected: 'Europe (Stockholm) eu-north-1'), and 'Copy settings from existing bucket - optional' (button: 'Choose bucket'). Below this is the 'Object Ownership' section with two radio buttons: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The footer includes standard AWS links like 'Privacy', 'Terms', and 'Cookie preferences', along with system status icons.

The screenshots show the AWS S3 Bucket creation process:

- Screenshot 1: Public Access Settings**  
Shows the "Block all public access" checkbox being selected. A warning message 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 for acknowledging this is checked.
- Screenshot 2: Default Encryption**  
Shows the "Server-side encryption with Amazon S3 managed keys (SSE-S3)" radio button selected. Other options include "Server-side encryption with AWS Key Management Service keys (SSE-KMS)" and "Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)". A note 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." A "Bucket Key" section has "Disable" and "Enable" options, with "Enable" selected.
- Screenshot 3: Advanced Settings**  
Shows a note: "After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings." At the bottom are "Cancel" and "Create bucket" buttons.

## 2. Upload the files of web application.

Screenshot of the AWS S3 Management Console showing the upload process for a website.

**Upload Progress:**

- Step 1: Initial upload screen showing a large dashed box for file/directory selection.
- Step 2: Files and folders listed (24 Total, 89.5 KB). Includes:
  - Bun 1.svg
  - Bun 1@2x.png
  - Cheese.svg
  - Cheese@2x.png
  - Lettuce.svg
  - Lettuce@2x.png
  - Onion.svg
  - Onion@2x.png
  - Patty.svg
  - Patty@2x.png
- Step 3: Confirmation message: "Upload succeeded".

**Summary:**

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

**Files and folders:**

Name	Folder	Type	Size	Status	Error
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		

### 3. Enable Static website hosting

The screenshot shows the AWS S3 console with the URL [s3.console.aws.amazon.com/s3/bucket/prasad-website/property/website/edit?region=eu-north-1](https://s3.console.aws.amazon.com/s3/bucket/prasad-website/property/website/edit?region=eu-north-1). The page is titled 'Edit static website hosting'. It has two main sections: 'Static website hosting' and 'Hosting type'. Under 'Static website hosting', 'Enable' is selected. Under 'Hosting type', 'Host a static website' is selected. A note at the bottom of this section states: '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](#)'. Below these, there are fields for 'Index document' (set to 'index.html') and 'Error document - optional' (set to 'error.html'). The status bar at the bottom shows 'CloudShell Feedback' and the date '15-10-2023'.

This screenshot is identical to the one above, except the 'Error document - optional' field is set to 'error.html' instead of 'index.html'. The rest of the configuration and the status bar are the same.

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 ENG IN 02:21 15-10-2023

## 5. Change the Bucket Policy

prasad-website - S3 bucket

s3.console.aws.amazon.com/s3/bucket/prasad-website/property/policy/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](#)

**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 ENG IN 02:21 15-10-2023

**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  Use a comma to separate multiple values.

AWS Service  All Services (\*)

Actions  1 Action(s) Selected  All Actions (\*)

Amazon Resource Name (ARN)  ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}. Use a comma to separate multiple values.

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 1: AWS Policy Generator**

The screenshot shows the AWS Policy Generator interface. At the top, there are tabs for 'prasad-website - S3 bucket', 'AWS Policy Generator', and 'prasad-website.s3.eu-north-1.amazonaws.com'. The main area is titled 'Policy JSON Document' and contains the following JSON code:

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

Below the code, a note states: 'Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will **not be reflected in the policy generator tool**'. A 'Close' button is at the bottom.

**Screenshot 2: AWS S3 Bucket Properties - Policy Tab**

This screenshot shows the 'Policy' tab of the AWS S3 Bucket Properties dialog. It displays the same JSON policy as the generator. On the right side, there's a sidebar titled 'Edit statement' with the sub-section 'Select a statement' and a button '+ Add new statement'.

**Screenshot 3: AWS CloudShell**

The screenshot shows the AWS CloudShell terminal with the following command executed:

```
aws s3api put-bucket-policy --bucket prasad-website --policy file://policy.json
```

The terminal output shows the policy has been successfully applied to the bucket.

Successfully edited bucket policy.

Amazon S3 > Buckets > prasad-website

**prasad-website** Info

Objects Properties **Permissions** Metrics Management Access Points

**Permissions overview**

Access  
Objects can be public

**Block public access (bucket settings)**

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

**Edit**

**Block all public access**  
⚠ Off  
► Individual Block Public Access settings for this bucket

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.

Not secure | prasad-website.s3-website.eu-north-1.amazonaws.com

**BRRRGRRR**

HUNGRY? GRAB A BRRRGRRR

**Ingredients**

- Patty
- Cheese
- Tomatoes
- Onions
- Lettuce

**PRICES**

Whole wheat bun .....	20
Patty .....	80
Onions .....	20
Tomatoes .....	20
Lettuce .....	20
Cheese slice .....	10

**Choose what goes into your burger**

Patty Cheese Tomatoes Onions Lettuce

**Current Order Total**  
**INR 170**  
To Pay

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 02:25 15-10-2023

## CONCLUSION:

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



## **Lab Assignment 4**

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

### **THEORY:**

AWS CodePipeline is a fully managed continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS). It enables you to automate the building, testing, and deployment of your applications or code changes, facilitating the release of software changes more quickly and reliably.

AWS CodePipeline works by creating a series of stages in a pipeline, each of which can perform various actions, such as source code retrieval, building, testing, and deployment. It can integrate with

various AWS services and other tools, providing a flexible and extensible way to set up your CI/CD workflows.

To deploy a static web application using AWS CodePipeline, you can follow these high-level steps:

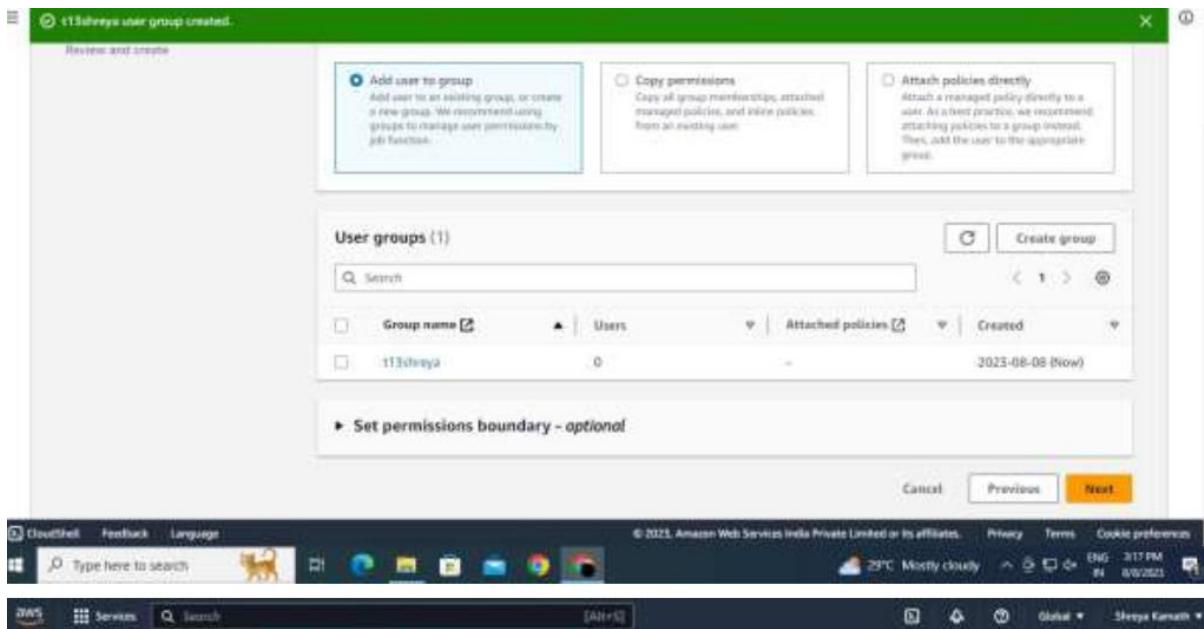
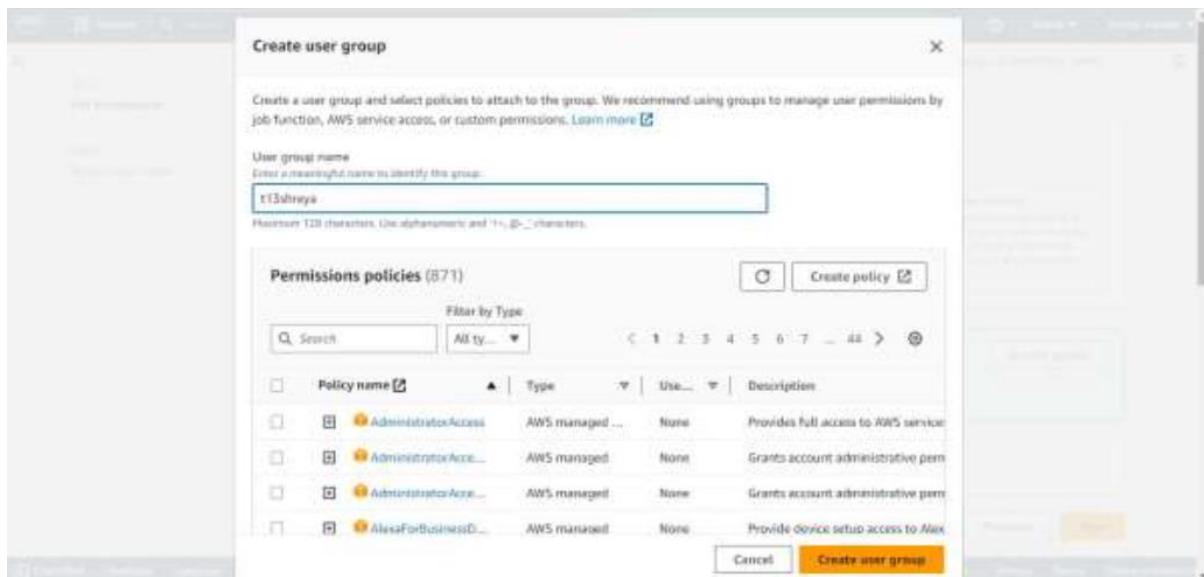
1. Create an Amazon S3 bucket to host your web app files.
2. Set up an AWS CodeCommit or other source code repository.
3. Create an AWS CodePipeline.
4. Define the source provider (CodeCommit, GitHub, S3).
5. Configure a build stage using AWS CodeBuild or similar services.
6. Set up optional testing stages.
7. Configure deployment stages for the S3 bucket.
8. Grant CodePipeline permissions to access the S3 bucket.
9. Review and validate your pipeline configuration.
10. Trigger a manual or automatic pipeline execution for testing.
11. Monitor pipeline executions for success or failures.
12. Enhance your pipeline as needed for testing, monitoring, or notifications.
13. Optionally, set up a custom domain with Amazon Route 53 and CloudFront.

## SCREENSHOTS:

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with various options like Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, and Analytics. The main area is titled "Users (0) info" and contains a sub-header: "An IAM user is an identity with long-term credentials that is used to interact with AWS or an account." Below this is a search bar labeled "Find users by username or access key". A table header includes columns for "User name", "Groups", "Last activity", "MFA", "Password age", and "Active". A message at the bottom of the table says "No resources to display".

This screenshot shows the "Specify user details" step of the "Create user" wizard. It's part of a three-step process: Step 1 (Specify user details), Step 2 (Set permissions), and Step 3 (Review and create). The main form is titled "User details" and contains a "User name" field with the value "shreyakamath". A note below the field specifies character limits and allowed symbols. There's also an optional checkbox for "Provide user access to the AWS Management Console - optional". A callout box provides instructions for generating programmatic access keys. At the bottom right are "Cancel" and "Next" buttons.

[Feedback](#) [Language](#) © 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)



Screenshot of the AWS IAM User Groups page showing the creation of a new user group named "t13shreya".

**Identity and Access Management (IAM)**

**User groups**

**t13shreya**

**Summary**

User group name: t13shreya	Creation time: August 08, 2023, 15:17 (UTC+05:30)	ARN: arn:aws:iam:538767473830:group/t13shreya
----------------------------	---	---

**Users** **Permissions** **Access Advisor**

**Users in this group (1)**

An IAM user is an entity that can create an AWS resource to represent the person or application that uses it to interact with AWS.

**Add users**

Screenshot of the AWS IAM Users page showing the creation of a new user named "shreyakamath".

**Identity and Access Management (IAM)**

**Users**

**shreyakamath**

**Summary**

Username: shreyakamath	Groups: t13shreya	Last activity: Never	MFA: None	Password last used: -	Active
------------------------	-------------------	----------------------	-----------	-----------------------	--------

**Add users**

Screenshot of the AWS IAM User details page for the user "shreyakamath".

**Identity and Access Management (IAM)**

**Users**

**shreyakamath**

**Summary**

ARN: arn:aws:iam:538767473830:user/shreyakamath	Console access: Disabled	Access key 1: Not enabled
Created: August 08, 2023, 15:17 (UTC+05:30)	Last console sign-in: -	Access key 2: Not enabled

**Permissions** **Groups (0)** **Tags** **Security credentials** **Access Advisor**

**Permissions policies (0)**

Permissions are defined by policies attached to the user directly or through groups.

**Add permissions**

SNS Services Search [Alt+F] Stockfish shreyakanth@5387-6747-5650

Console Home [Info](#)

Recently visited [Info](#)

Cloud9 [Edit](#)

View all services

Welcome to AWS

Getting started with AWS [Edit](#)  
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification [Edit](#)  
Learn from AWS experts and advance your skills and knowledge.

What's new with AWS? [Edit](#)  
Discover new AWS services, features, and Regions.

AWS Health [Info](#)

CloudShell Feedback Language Type here to search  © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Mostly cloudy 8:28 PM 8/8/2021

Developer Tools [X](#)

CodeCommit

Source + CodeCommit

- Getting started
- Repositories
- Code**
- Pull requests
- Commits
- Branches
- Git tags
- Settings
- Approval rule templates

Artifacts + CodeArtifact

Build + CodeBuild

Deploy + CodeDeploy

Pipeline + CodePipeline

Step 1: Prerequisites

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

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

Step 2: Git credentials

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

Step 3: Clone the repository

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

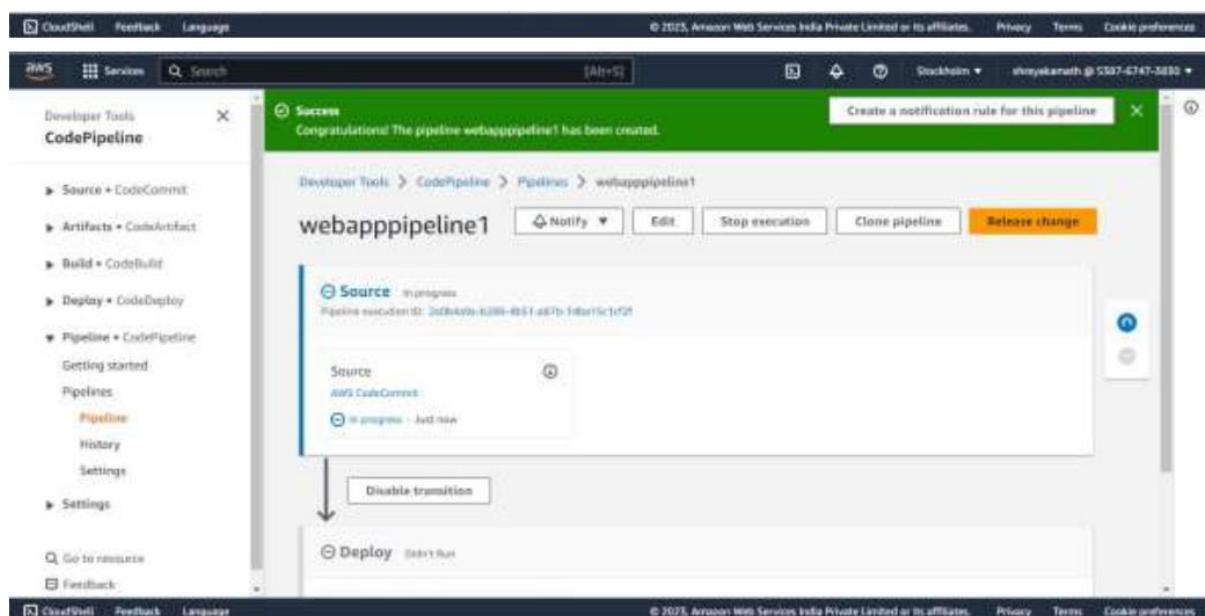
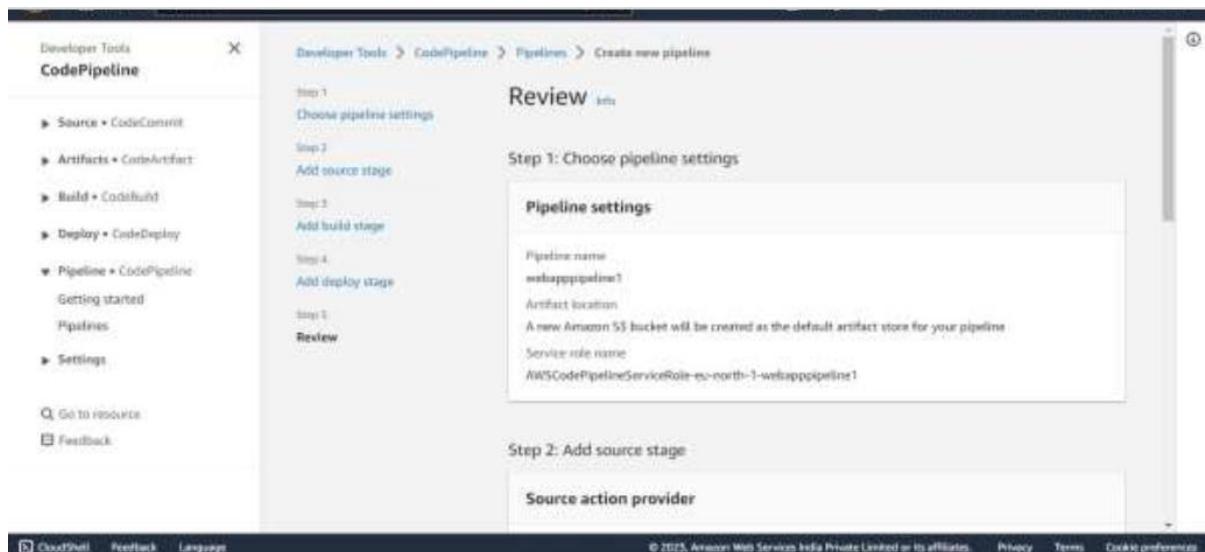
```
git clone https://git-codecommit.eu-north-1.amazonaws.com/v1/repos/ad-devops
```

Copy [Edit](#)

Additional details

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

CloudShell Feedback Language [Alt+F] Stockfish shreyakanth@5387-6747-5650



## CONCLUSION:

Hence, I have learnt the concept of Continuous Delivery/Continuous Integration and successfully deployed a static web application using AWS CodePipeline.

## Lab Assignment 05

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

**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.

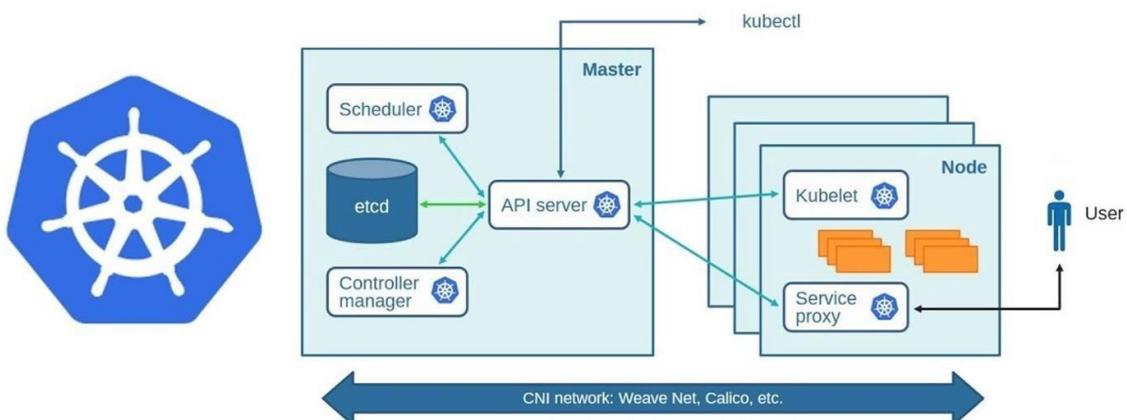
It schedules, runs, and manages isolated containers that are running on virtual/physical/cloud machines.

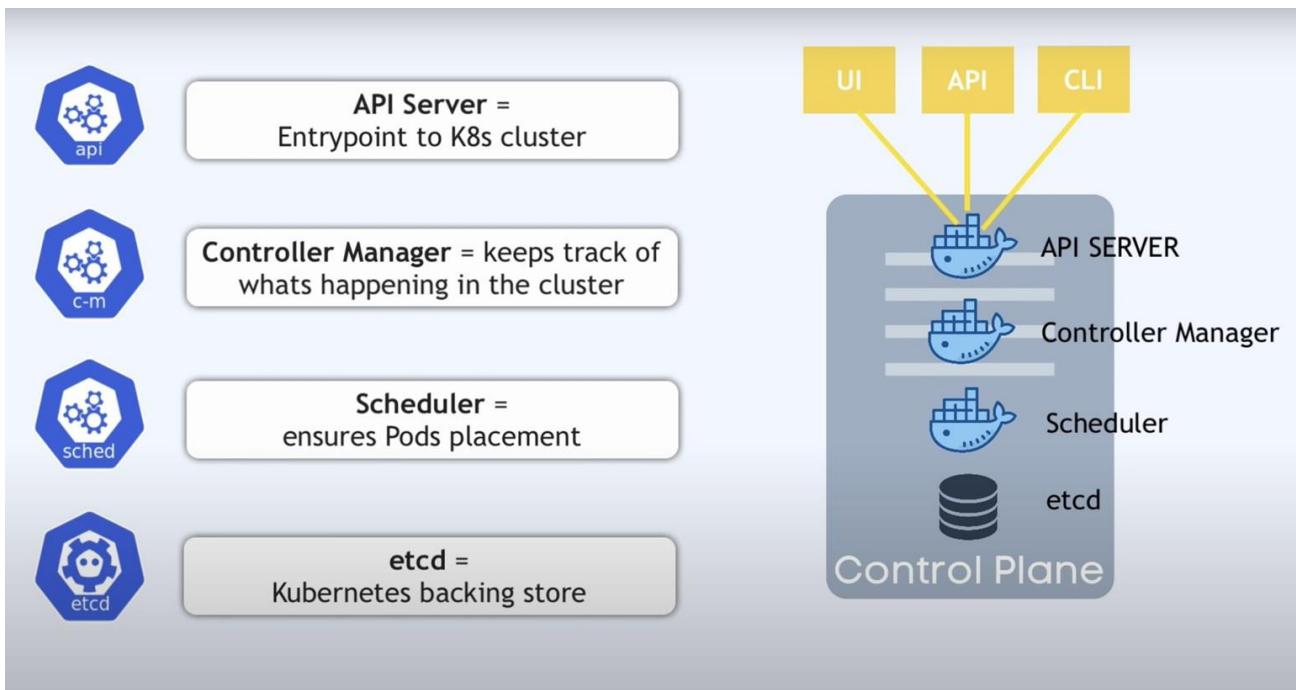
All top cloud providers support Kubernetes.

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 or 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 kube-api-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.
- 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

```

Activities Terminal Oct 14 22:11 • prasad@prasad-VirtualBox:-
prasad@prasad-VirtualBox:~$ docker run hello-world
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.
    (and)
 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 -it 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:~$ 

```

```

Activities Terminal Oct 14 22:11 • prasad@prasad-VirtualBox:-
prasad@prasad-VirtualBox:~$ curl -L https://get.docker.com | sh
  % Total    % Received ==========[>]=====
  0     0    0     0    0     0      0      0 --:--:--:--:--:-- 
  100   100    0     0    0     0      0      0 --:--:--:--:--:-- 
Unpacking curl (7.68.0-ubuntu2.20) ...
Setting up curl (7.68.0-ubuntu2.20) ...
Processing triggers for man-db (2.9.1-1) ...
prasad@prasad-VirtualBox:~$ sudo install -n 0755 -d /etc/apt/keyrings
prasad@prasad-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
prasad@prasad-VirtualBox:~$ echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
> sudo tee /etc/apt/sources.list.d/docker.list >/dev/null
prasad@prasad-VirtualBox:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [33.3 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 91.0 kB in 2s (59.6 kB/s)
Reading package lists...
Done
prasad@prasad-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists...
Reading package lists...
Building dependency tree
Reading state information...
The following packages were automatically installed and are no longer required:
  adium-theme-ubuntu-command-not-found-data cpp-7 difftstat enhanceman example-content fwupdatd gcc-7-base gcc-8-base gettext gir1.2-goa-1.0 gir1.2-gtksource-3.0 gir1.2-mutter-2 gnome-software-common
  gnome-user-guide guile-2.0-libs ifupdown intltool-debian iptutils-arping libappstream-glib8 libapt-pkg-perl libarchive-zip-perl libargon2-0 libart-2.0-2 libasync-mergepoint-perl
  libbb-hooks-endoscope perl-libbb-hooks-op-check perl-time-timer perl libboost-filesystem-1.65.1 libboost-iostreams-1.65.1 libboost-system-1.65.1 libboost-thread-1.65.1
  libcalipso9 libcamerawriter libcapnp libccache libcdio libcdio-paranoia libcdio-tracklib libcdio-tracklib libcurl libcurl4 libcurl4-openssl-dev libcurl4-openssl4 libcurl4-openssl4-dev libcurl4-openssl4v4 libcurl4-openssl4v4-dev libcurl4-openssl4v4v4 libcurl4-openssl4v4v4-dev libcurl4-openssl4v4v4v4 libcurl4-openssl4v4v4v4-dev libcurl4-openssl4v4v4v4v4 libcurl4-openssl4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4-dev libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4 libcurl4-openssl4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v4v44

```

## 2. Install minikube using following commands

```

Oct 14 22:27 ●
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: $ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
     0 82.4M     0 82.4M      0      0  0:00:15  0:00:15  --:-- 5916k
prasad@prasad-VirtualBox: $ sudo install minikube-linux-amd64 /usr/local/bin/minikube
[sudo] password for prasad:
prasad@prasad-VirtualBox: $ minikube start --driver=docker
minikube v1.31.2 on Ubuntu 20.04 (vbox/amd64)
Using the docker driver based on user configuration

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

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

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

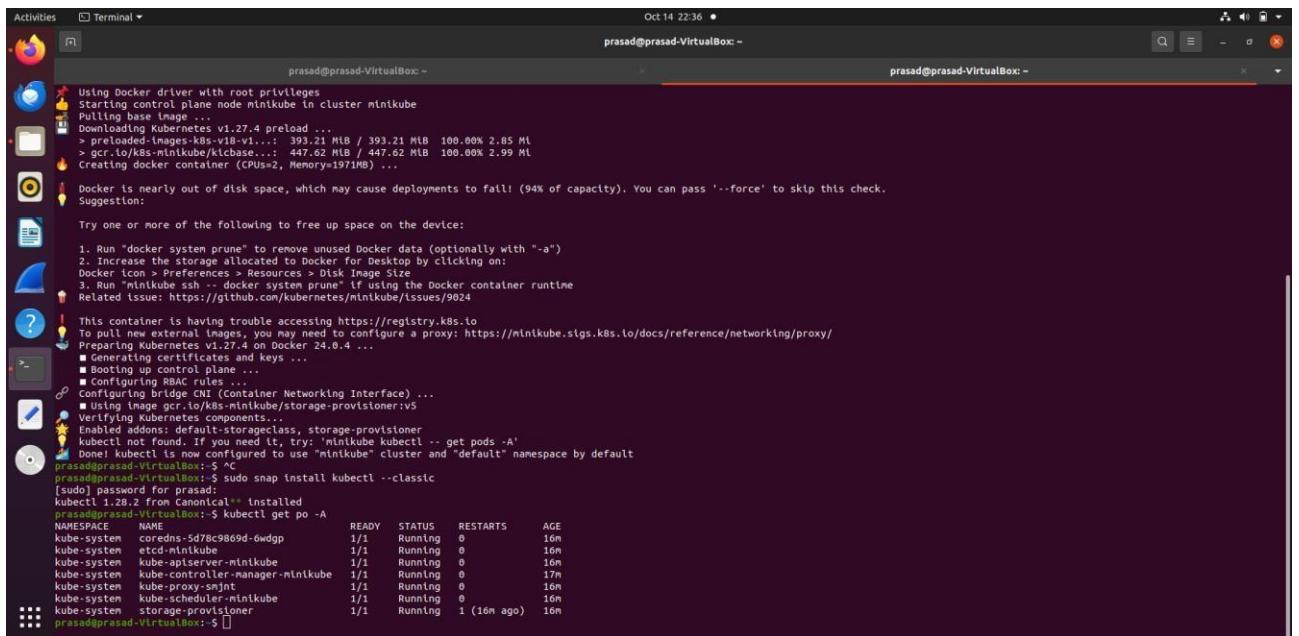
Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
Suggestion:

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 bridge CNI (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'
  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasad@prasad-VirtualBox: $ ls
prasad@prasad-VirtualBox: ~

```

### 3. Install kubectl



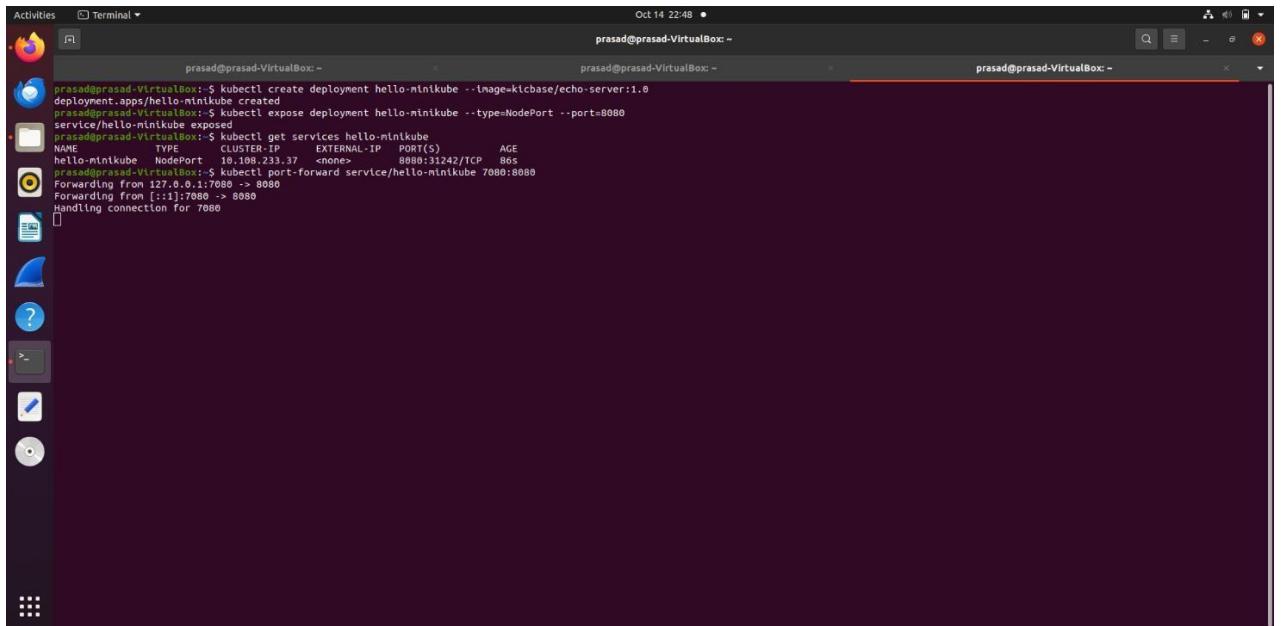
```
Activities Terminal Oct 14 22:36 ● prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~
prasad@prasad-VirtualBox: ~

Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image ...
  Downloading Kubernetes v1.27.4 preload ...
    > 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=1971MiB) ...
Docker is nearly out of disk space, which may cause deployments to fail! (94% of capacity). You can pass '--force' to skip this check.
Suggestion:

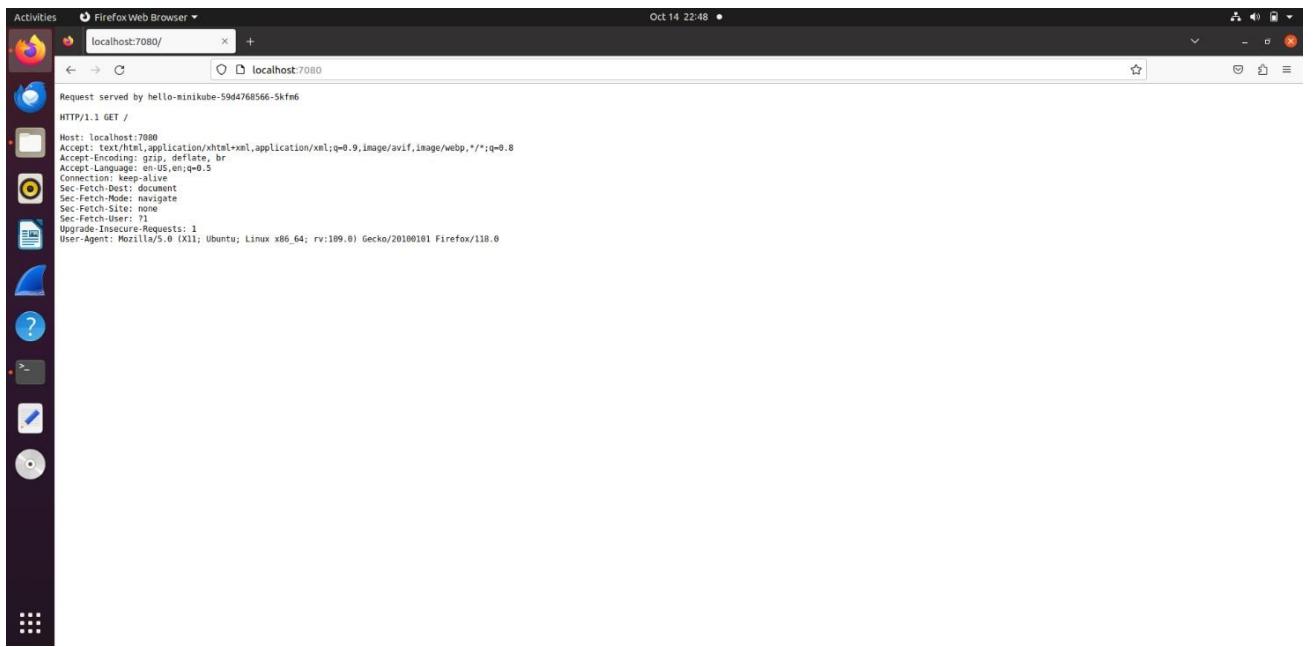
Try one or more of the following to free up space on the device:
1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
2. Increase the storage allocated to Docker for Desktop by clicking on:
Docker icon > Preferences > Resources > Disk Image Size
3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
Related issue: https://github.com/kubernetes/minikube/issues/9624

This container is having trouble accessing https://registry.k8s.io
To resolve this 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 image gcr.io/k8s-minikube/storage-provisioner:v5
  ○ Verifying Kubernetes components ...
  ■ Publishing default port forwarder class, storage-provisioner ...
kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasad@prasad-VirtualBox: ~ $ ~
prasad@prasad-VirtualBox: ~ $ sudo snap install kubectl --classic
[sudo] password for prasad:
kubectl 1.28.2 from Canonical * installed
prasad@prasad-VirtualBox: ~ $ kubectl get po -A
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-apiserver-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)
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   10.1.0.10   23.37.14.17   8080:31242/TCP   86s
prasad@prasad-VirtualBox: ~ $ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::]:7080 -> 8080
Handling connection for 7080
```



## CONCLUSION:

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

## LAB ASSIGNMENT 6

**AIM:** To understand terraform lifecycle, core concepts/terminologies and install it.

**LO3:** To apply best practices for managing infrastructure as code environments and use terraform to define and deploy cloud infrastructure.

### THEORY:

Terraform is one of the most popular Infrastructure-as-code (IaC) tool, used by DevOps teams to automate infrastructure tasks. It is used to automate the provisioning of your cloud resources. Terraform is an open-source, cloud-agnostic provisioning tool developed by HashiCorp and written in GO language.

Benefits of Terraform:

- Does orchestration, not just configuration management.
- Supports multiple providers such as AWS, Azure, Oracle, GCP, and many more.
- Provide immutable infrastructure where configuration changes smoothly.
- Uses easy to understand language, HCL (HashiCorp configuration language).
- Easily portable to any other provider.

### TERRAFORM LIFECYCLE

Terraform lifecycle consists of – **init**, **plan**, **apply**, and **destroy**.



1. **Terraform init** initializes the (local) Terraform environment. Usually executed only once per session.
2. **Terraform plan** compares the Terraform state with the as-is state in the cloud, build and display an execution plan. This does not change the deployment (read-only).
3. **Terraform apply** executes the plan. This potentially changes the deployment.
4. **Terraform destroy** deletes all resources that are governed by this specific terraform environment.

### TERRAFORM CORE CONCEPTS/TERMINOLOGIES

1. Variables: Terraform has input and output variables, it is a key-value pair. Input variables are used as parameters to input values at run time to customize our deployments. Output variables are return values of a terraform module that can be used by other configurations.
2. Provider: Terraform users provision their infrastructure on the major cloud providers such as AWS, Azure, OCI, and others. A provider is a plugin that interacts with the various APIs required to create, update, and delete various resources.
3. Module: Any set of Terraform configuration files in a folder is a module. Every Terraform configuration has at least one module, known as its root module.
4. State: Terraform records information about what infrastructure is created in a Terraform state file. With the state file, Terraform is able to find the resources it created previously, supposed to manage and update them accordingly.
5. Resources: Cloud Providers provides various services in their offerings, they are referenced as Resources in Terraform. Terraform resources can be anything from compute instances, virtual networks to higher-level components such as DNS records. Each resource has its own attributes to define that resource.
6. Data Source: Data source performs a read-only operation. It allows data to be fetched or computed from resources/entities that are not defined or managed by Terraform or the current Terraform configuration.
7. Plan: It is one of the stages in the Terraform lifecycle where it determines what needs to be created, updated, or destroyed to move from the real/current state of the infrastructure to the desired state.
8. Apply: It is one of the stages in the Terraform lifecycle where it applies the changes real/current state of the infrastructure in order to achieve the desired state.

## INSTALLATION:

### 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>

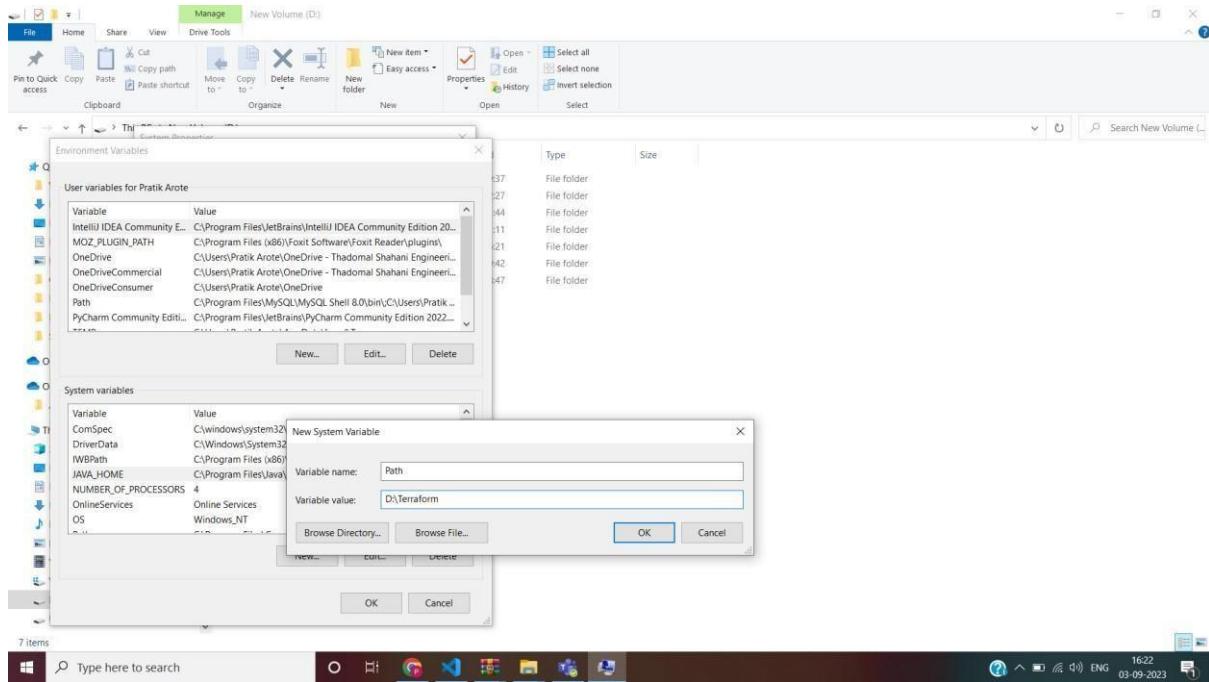
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.

The screenshot shows the HashiCorp Terraform website's download page. At the top, it says "Install or update to v1.5.6 (latest version) of Terraform to get started." Below this, there are tabs for Operating System (macOS, Windows, Linux, FreeBSD, OpenBSD, Solaris), with Windows selected. Under "Binary download for Windows", there are two options: "386" (Version: 1.5.6) and "AMD64" (Version: 1.5.6), each with a "Download" button. Below these, under "Release information", is a "Changelog" section with a "GitHub" link. On the right side, there are boxes for "About Terraform", "Featured docs" (Introduction to Terraform, Configuration Language, Terraform CLI, Terraform Cloud, Provider Use), and "Terraform Cloud" (Automate your infrastructure provisioning at any scale). At the bottom, there is a cookie consent message and an "ACCEPT" button.

## 2. Extract the downloaded setup file Terraform.exe in C:\Terraform Directory

The screenshot shows a Windows File Explorer window with a ZIP file named "terraform\_1.5.6\_windows\_amd64.zip" selected. A context menu is open, and a "Extraction path and options" dialog box is overlaid. In the dialog, the "Destination path (will be created if does not exist)" is set to "C:\". The "Update mode" is set to "Extract and replace files". Other options include "Overwrite mode" (set to "Ask before overwirite"), "Miscellaneous" (checkboxes for "Keep broken files" and "Display files in Explorer"), and a "Save settings" button. The background shows a desktop with a taskbar at the bottom.

## 3. Set the System path for Terraform in Environment Variables.



#### 4. Open PowerShell with Admin Access. Open Terraform in PowerShell and check its functionality.

```

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Pratik Arote> cd D:
PS D:\> Terraform
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console   Try Terraform expressions at an interactive command prompt
  fmt       Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get       Install or upgrade remote Terraform modules
  graph    Generate a graph of the steps in an operation
  import   Associate existing resources with Terraform
  login    Obtain and save credentials for a remote host
  logout   Remove locally-stored credentials for a remote host
  metadata Metadata related commands
  output   Show output values from your root module
  provider Show providers used for this configuration
  refresh  Update the state to match remote systems
  show    Show the current state or a saved plan
  state   Advanced state management
  taint   Mark a resource instance as no longer functional
  test    Execute tests for individual integration testing
  untaint Remove the "tainted" state from a resource instance
  version Show the current Terraform version
  workspace Workspace management

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

PS D:\>

```

## CONCLUSION:

Here, we studied the about the terraform lifecycle and terminologies/concepts of terraform and installed it on our system.

## Lab Assignment 7

**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/analyzing-sourcecode/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-cli
Using default tag: latest
latest: Pulling from sonarsource/sonar-scanner-cli
9398808236ff: Pull complete
4f4fb700ef54: Pull complete
3cd77fb28e46: Pull complete
f78b288abc31: Pull complete
Digest: sha256:494ecc3b5b1ee1625bd377b3905c4284e4f0cc155cff397805a244dee1c7d575
Status: Downloaded newer image for sonarsource/sonar-scanner-cli:latest
docker.io/sonarsource/sonar-scanner-cli:latest

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

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   713c7cdaaf78  2 weeks ago   42.7MB
myimage              latest   438bb56a50a3  2 weeks ago   122MB
sonarsource/sonar-scanner-cli  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.

The screenshots illustrate the SonarQube web interface. The first screenshot shows the login screen with fields for 'Login' and 'Password', and buttons for 'Log In' and 'Cancel'. The second screenshot shows the 'How do you want to create your project?' page, which lists several import options: 'Import from Azure DevOps' (Setup), 'Import from Bitbucket Cloud' (Setup), 'Import from Bitbucket Server' (Setup), 'Import from GitHub' (Setup), and 'Import from GitLab' (Setup). The third screenshot shows the same page with a tooltip about using an embedded database for evaluation purposes, and a 'Get the most out of SonarQube!' sidebar with information about SonarLint.

#### 4. Create a project

The screenshot shows the 'Create a project' page in SonarQube. The project display name is 'sonarPythonProgram'. The project key is also 'sonarPythonProgram'. The main branch name is 'main'. A note at the bottom says 'The name of your project's default branch [Learn More](#)'. A 'Next' button is visible.

**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 states.

[Learn More](#) [Dismiss](#)

The screenshot shows the 'Set up project for Clean as You Code' page. It asks to choose a baseline for new code. The 'Use the global setting' option is selected, which is 'Previous version'. A note says 'Any code that has changed since the previous version is considered new code.' Another note says 'Recommended for projects following regular versions or releases.' Other options shown are 'Define a specific setting for this project' (with 'Previous version' and 'Number of days' sub-options), 'Continuous delivery' (with 'Last x days' and 'Recommended for projects following continuous delivery' notes), and 'Definition of new code' (with 'Definition of new code' and 'Definition of new code' notes).

**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 states.

[Learn More](#) [Dismiss](#)

Congratulations! Your project has been created.

How do you want to analyze your repository?

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

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.

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 states.

## 5. Provide token

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

1 Provide a token

pythonToken: sqp\_c8922aed03df90f4cc0dfb26200772ff42a9032b

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

Continue

2 Run analysis on your project

Embedded database should be used for evaluation purposes only  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support.

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 states.

**2 Run analysis on your project**

What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

What is your OS?

Linux Windows macOS

Download and unzip the Scanner for Windows

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `%PATH%` environment variable.

Execute the Scanner

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

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

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

## 6. Enter the following command

```
C:\Windows\System32\cmd.exe > 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: NONE
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: 1478411F-AYsxFDZoQL-ruFd2_S5
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
```

```

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 Imports [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 a project named 'sonarPythonProgram1'. The main dashboard features a green 'Passed' status for the Quality Gate. Below this, there are several cards providing metrics: Reliability (0 Bugs, A grade), Maintainability (0 Code Smells, A grade), Security (0 Vulnerabilities, A grade), Security Review (0 Security Hotspots, A grade), Coverage (0.0% Coverage), and Duplications (0.0% Duplications). The navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

## CONCLUSION:

Here we have successfully performed static analysis of python programs using SonarQube.

## Assignment No 8

**Aim:** To understand continuous monitoring using Nagios.

**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**

Google

nagios demo

All Videos Images News Shopping More Tools

About 3,87,000 results (0.44 seconds)

<https://exchange.nagios.org/directory/Demos/details> ::

**Nagios XI Online Demo**

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

<https://exchange.nagios.org/directory/Demos> ::

**Demos - Nagios Exchange**

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

<https://exchange.nagios.org/directory/Demos/details> ::

Now click on the website-

The screenshot shows the Nagios XI Online Demo page. At the top, there's a navigation bar with links for Home, Directory (which is underlined), and About. Below the navigation is a breadcrumb trail: Home | Directory | Demos | Nagios XI Online Demo. The main content area features a "Directory Tree" section with a heading "Nagios XI Online Demo". Below this, there's a review summary: "Submit review | Recommend | Print | Visit | Claim | Rating: ★★★★☆ | Favoured: 0 | 0 votes | Owner: egalstad | Website: nagiosxi.demos.nagios.com | Hits: 141800". To the right, there are two search boxes: "Search Exchange" and "Search All Sites", each with a "Go" button. A sidebar on the right is titled "Nagios Live Webinars" with the subtext "Let our experts show you how Nagios can help your organization."

Now click on login as administrator

The screenshot shows a web browser window with two tabs: "Nagios XI Online Demo - Nagios XI" and "Login - Nagios XI". The main content is the "Login" page for the "Nagios XI Demo System". It includes fields for "Username" and "Password", a "Login" button, and a link "Forgot your password?". Below the login form is a "Select Language:" dropdown with various flags. To the right, there's a "Demo Account Options" section with a heading "Nagios XI Demo System". It describes how users can access the demo with different accounts. It lists five account types with their respective log-in buttons and credentials:

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

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

It will have interface like this

The screenshot shows the Nagios XI Home Dashboard. On the left, there's a sidebar with sections like Quick View, Details, Graphs, and Maps. The main area has several cards: 'Getting Started Guide' with common tasks, 'Host Status Summary' (Up: 132, Down: 1, Unreachable: 1, Pending: 0), 'Service Status Summary' (Ok: 1267, Warning: 29, Unknown: 97, Critical: 1, Pending: 0), and 'Administrative Tasks'. To the right, there's a 'We're Here To Help!' section with a photo of a support team member and links to Support Forum, Customer Support, Help Resources, and Customer Ticket Support Center. At the bottom, there are buttons for 'Start Monitoring' (Run a Config Wizard, Run Auto-Discovery, Advanced Config), and the footer includes copyright information and system status.

Now click on Host status-

The screenshot shows the Nagios XI Host Status page. The left sidebar is identical to the previous dashboard. The main area features two summary tables: 'Host Status Summary' (Up: 132, Down: 1, Unreachable: 1, Pending: 0) and 'Service Status Summary' (Ok: 1267, Warning: 29, Unknown: 97, Critical: 1, Pending: 0). Below these are two large tables: 'All hosts' (showing 1-242 of 242 total records) and 'Showing 1-242 of 242 total records'. The 'All hosts' table includes columns for Host, Status, Duration, Attempt, Last Check, and Status Information. A search bar and pagination controls are at the top of the host list table. The footer includes standard navigation links and system status.

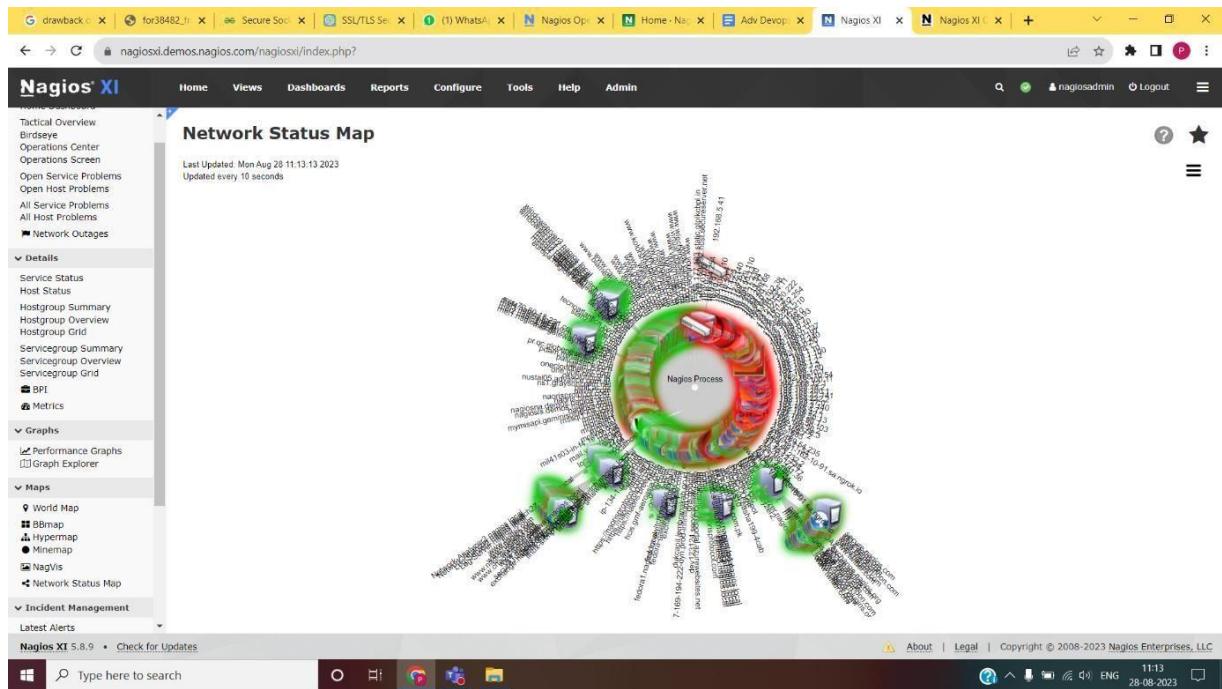
In the above image one can see Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.

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-

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 9

**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:

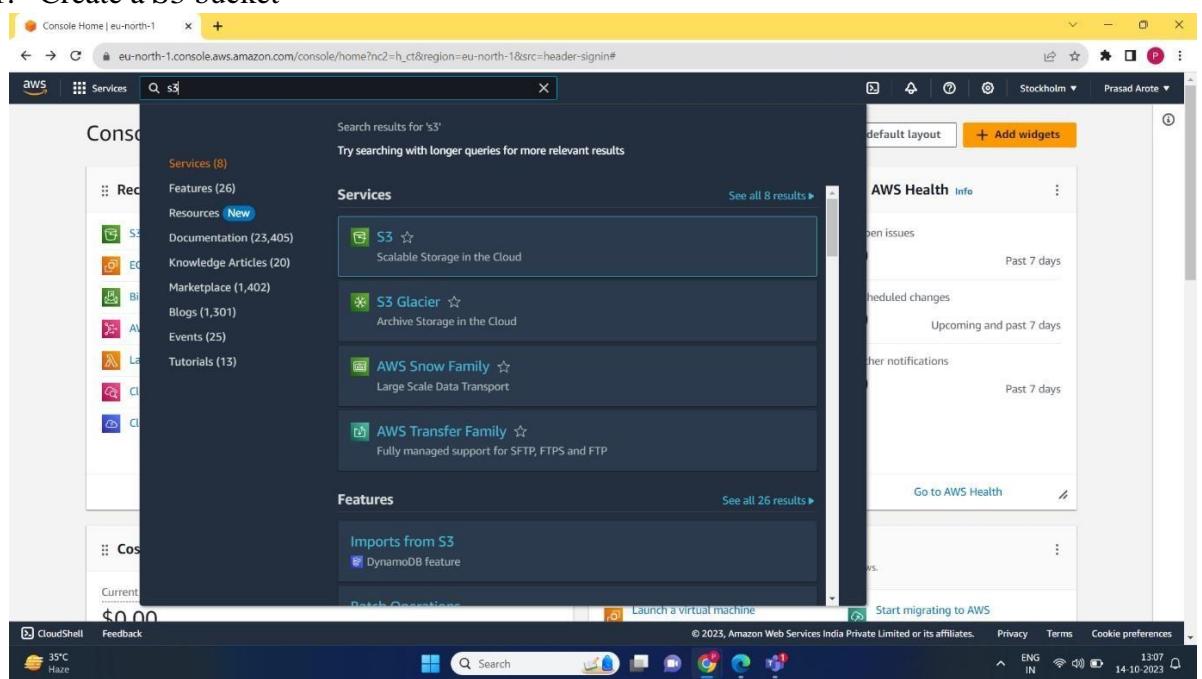
#### LAMBDA FUNCTION

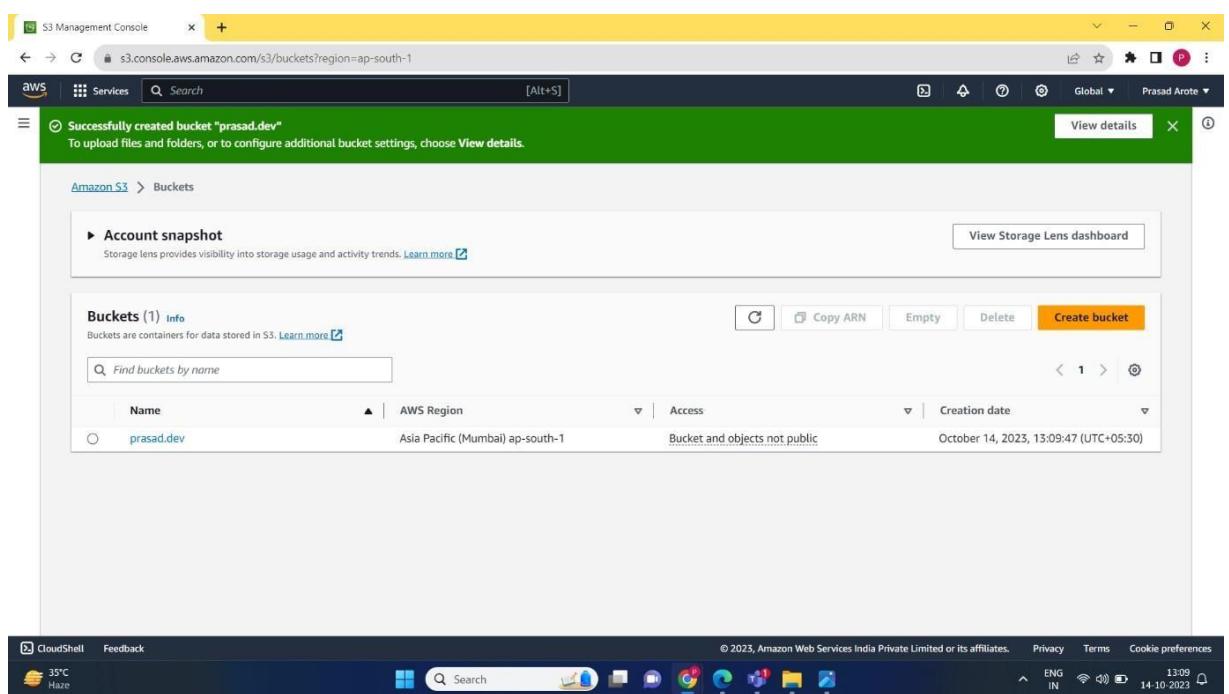
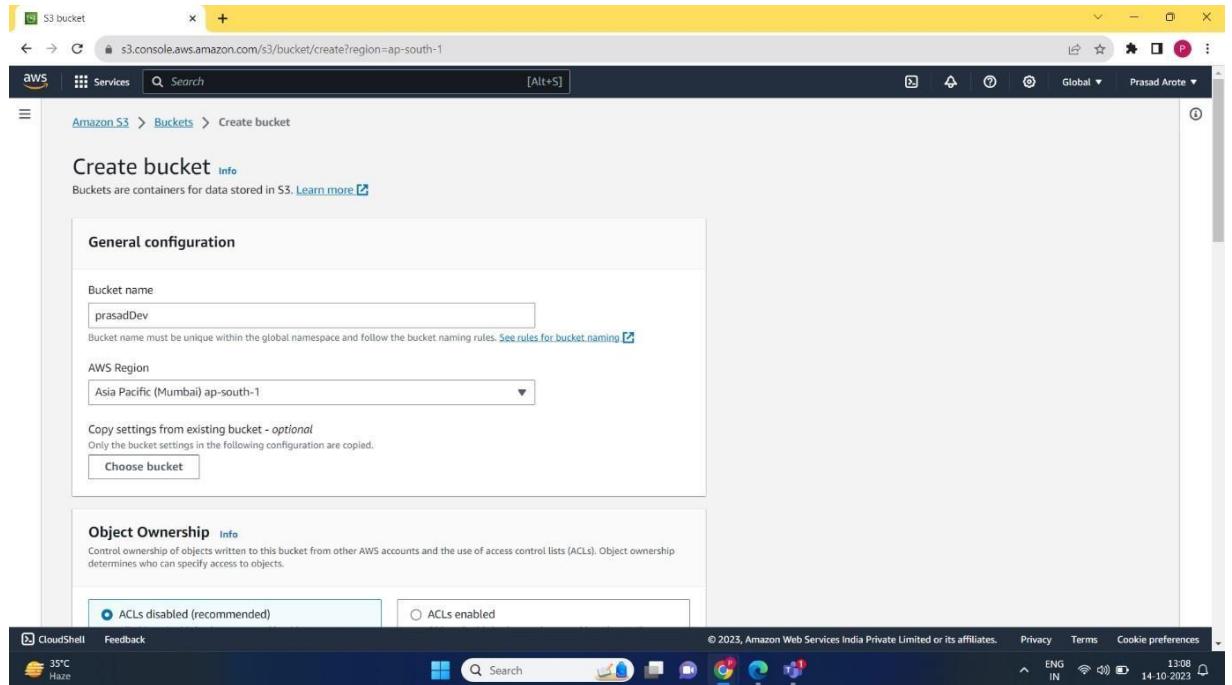
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you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



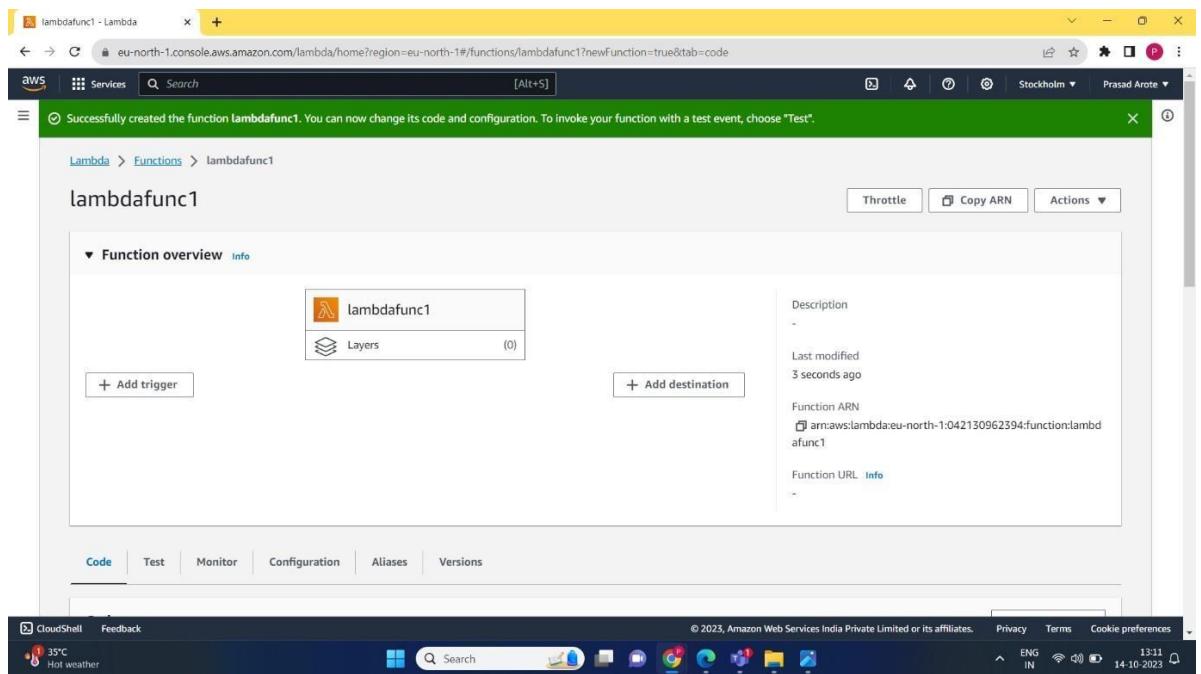
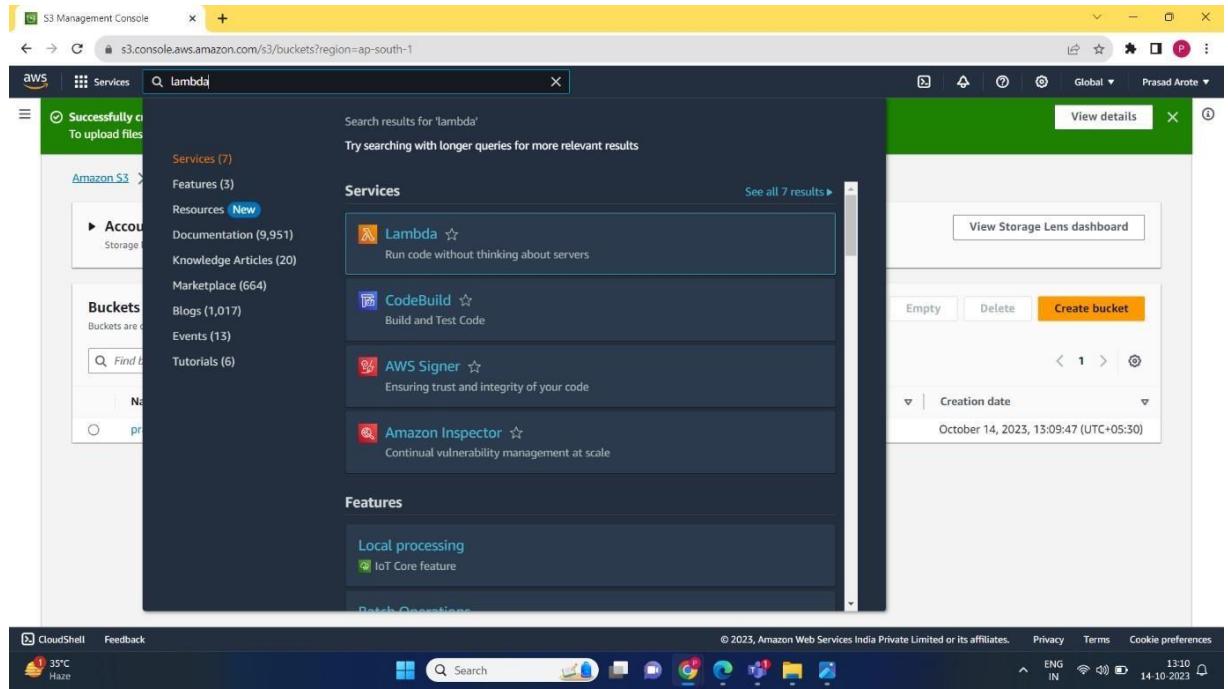
### Installation:

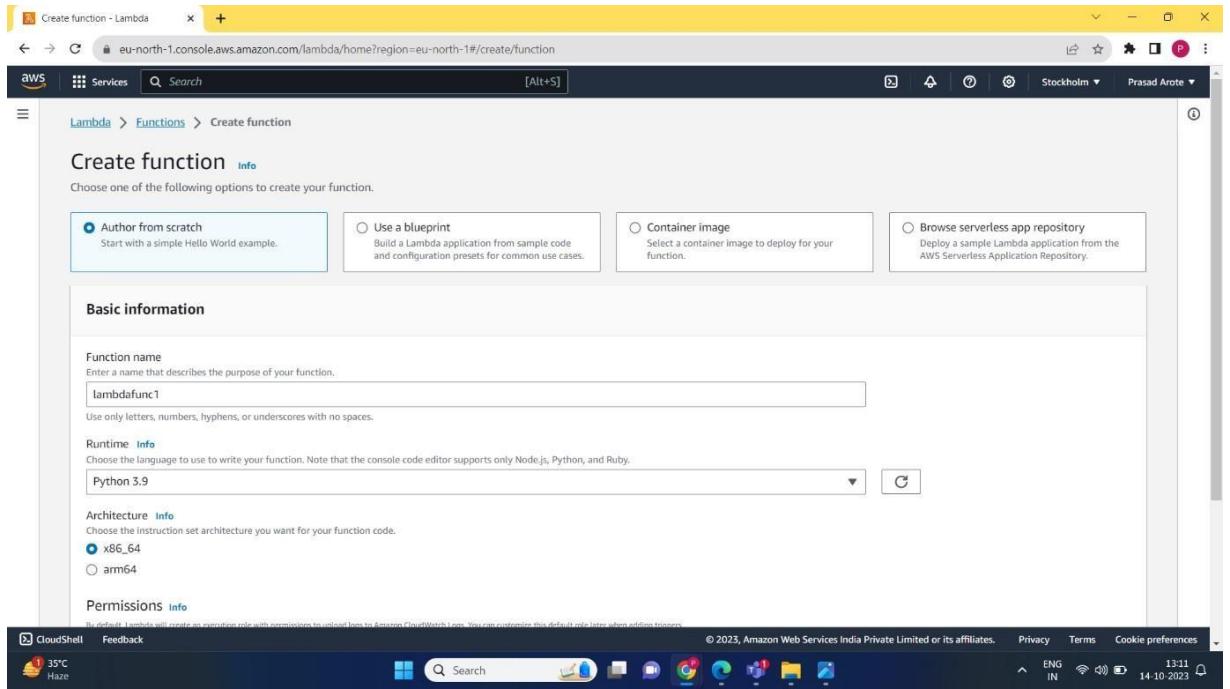
#### 1. Create a S3 bucket



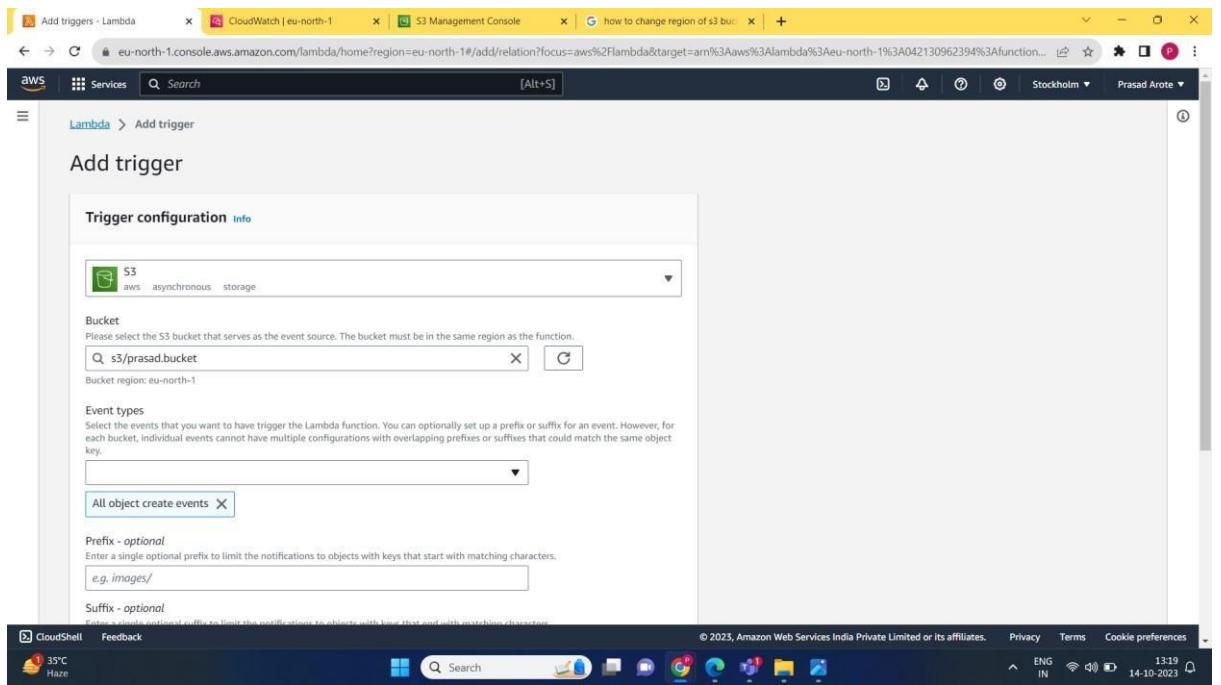


## 2. Create a Lambda function.





### 3. Create a trigger



Screenshot of the AWS Lambda trigger configuration page:

**Event types**: All object create events.

**Prefix - optional**: e.g. images/

**Suffix - optional**: e.g. jpg

**Recursive invocation**: Acknowledged that using the same S3 bucket for both input and output is not recommended and can lead to increased Lambda usage and costs.

**Lambda permissions**: Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger.

**Buttons**: Cancel, Add.

Screenshot of the AWS Lambda function configuration page for "lambdafunc1":

**Trigger**: The trigger "prasad.bucket" was successfully added to function "lambdafunc1".

**Function overview** (Info):

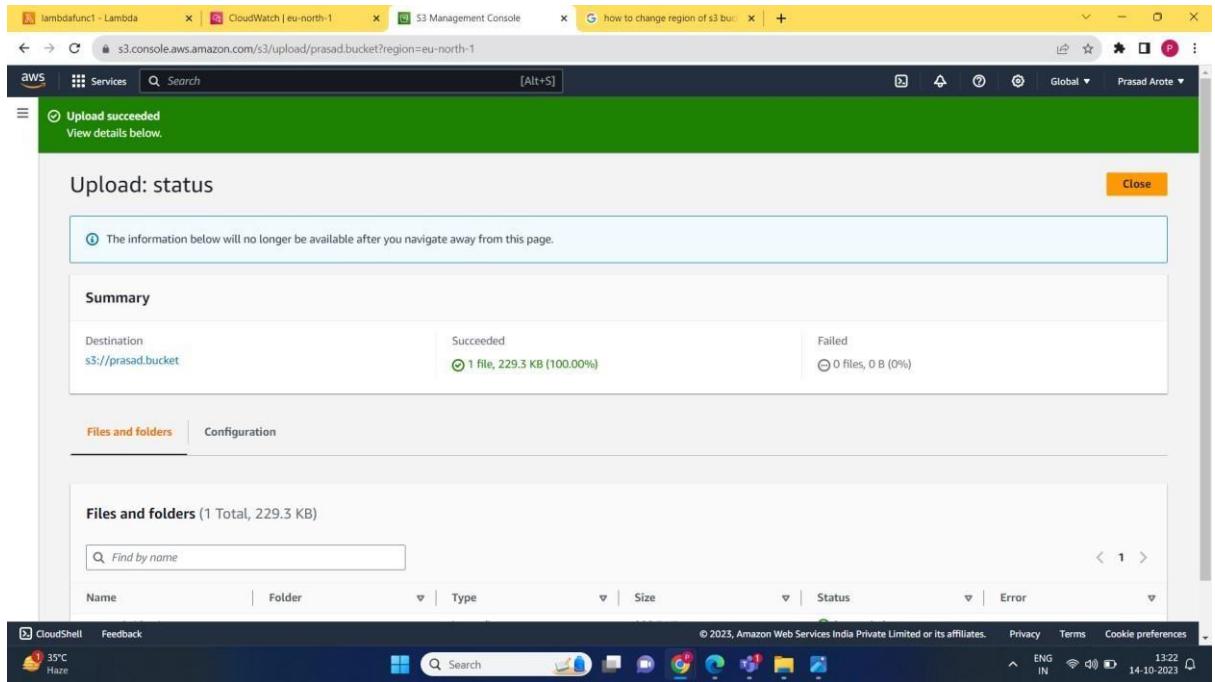
- Function Name**: lambdafunc1
- Description**: (empty)
- Last modified**: 8 minutes ago
- Function ARN**: arn:aws:lambda:eu-north-1:042130962394:function:lambdafunc1
- Function URL**: (Info)

**Destinations**: S3 (selected), + Add destination, + Add trigger.

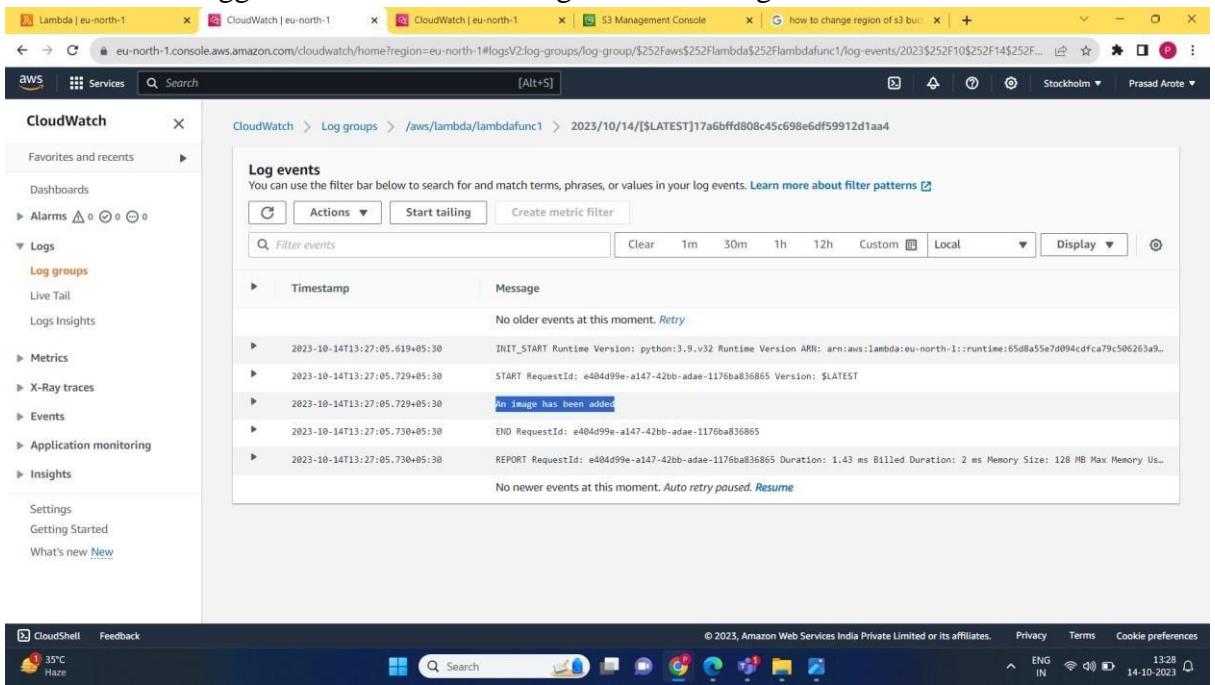
**Configuration tabs**: Code, Test, Monitor, Configuration (selected), Aliases, Versions.

The screenshot shows the AWS S3 Management Console interface for uploading files to a bucket named 'prasad.bucket'. The process is divided into several steps:

- Upload Info:** A large text area at the top allows users to drag and drop files or choose 'Add files' or 'Add folder'. It includes a note about uploading files larger than 160GB.
- Files and folders (0):** A table where users can manage uploaded files. It has columns for Name, Folder, Type, and Size. A search bar and sorting options are available.
- No files or folders:** A message indicating that no files have been chosen for upload.
- Destination:** A section where users can specify the destination bucket. In this step, the bucket 's3://prasad.bucket' is selected.
- Destination details:** A sub-section showing bucket settings for new objects stored in the specified destination.
- Permissions:** An optional section for granting public access or access to other AWS accounts.
- Properties:** An optional section for specifying storage class, encryption settings, and tags.
- Cancel** and **Upload** buttons at the bottom right of the form.



- 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.



## Assignment 10

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

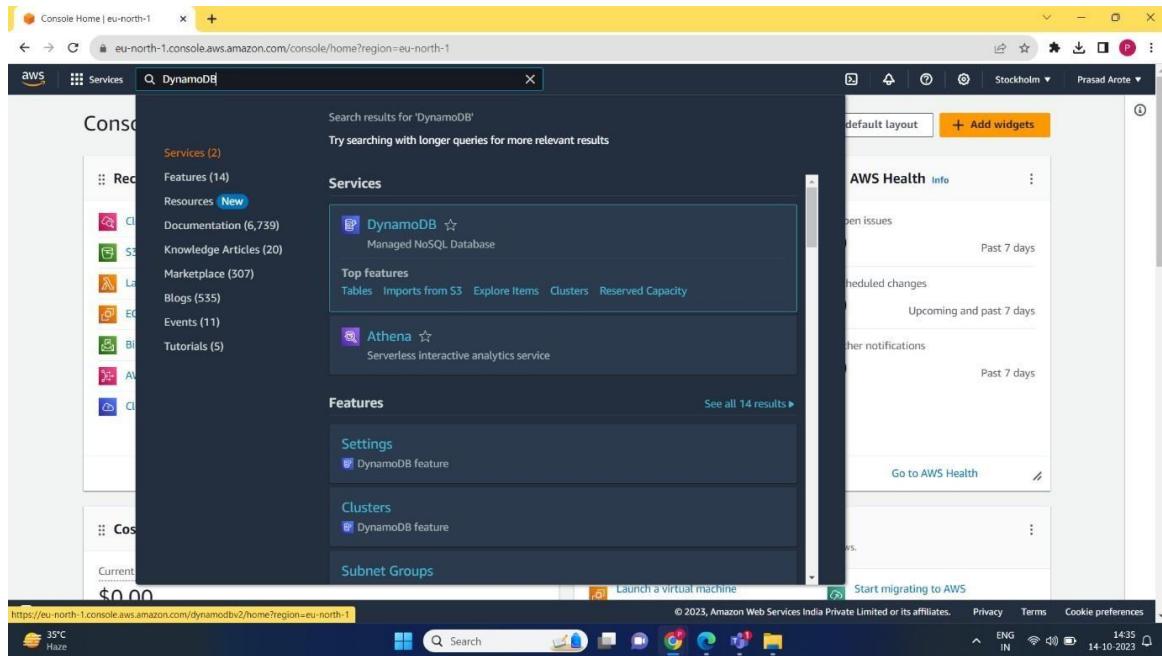
**LO6:** To engineer a composition of nano services using AWS Lambda and Step Functions with the serverless framework.

### **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.

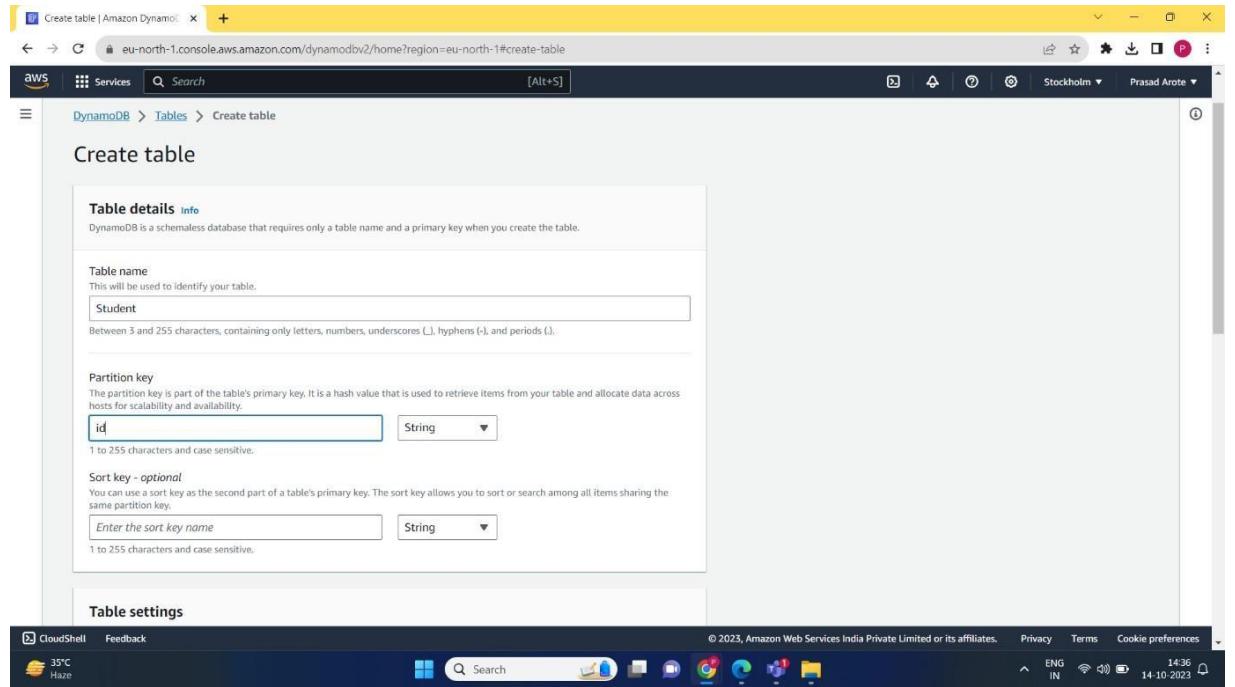
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource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



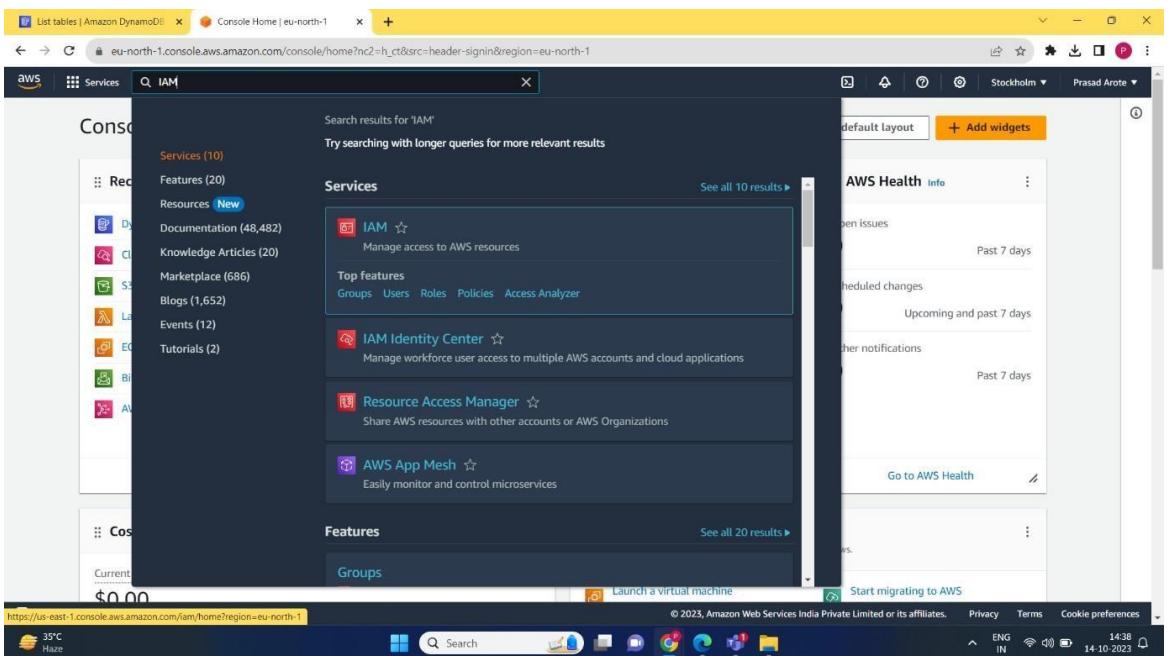
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

## STEPS:

### 1. Create a table



### 2. Create a role using IAM



### 3. Add permissions – AmazonDynamoFullAccess

**Select trusted entity**

**Trusted entity type**

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

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

Service or use case  
Lambda

**Add permissions**

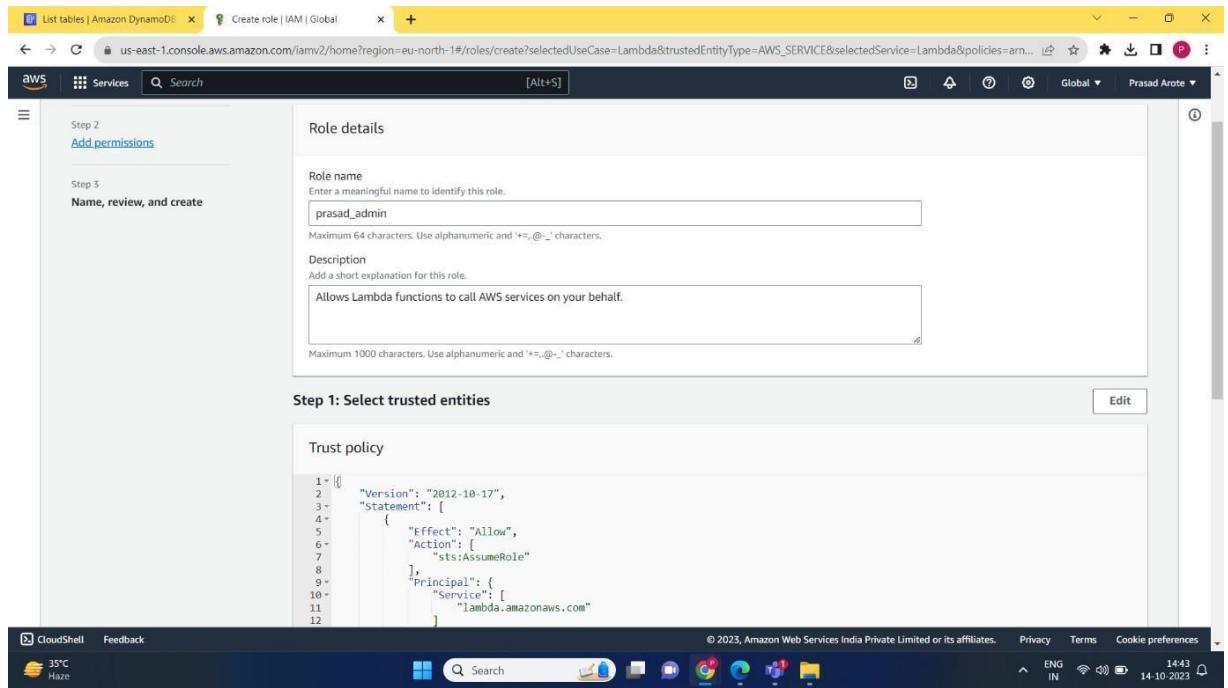
**Permissions policies (1/887)**

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

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

**Set permissions boundary - optional**

Cancel Previous Next



**Identity and Access Management (IAM)**

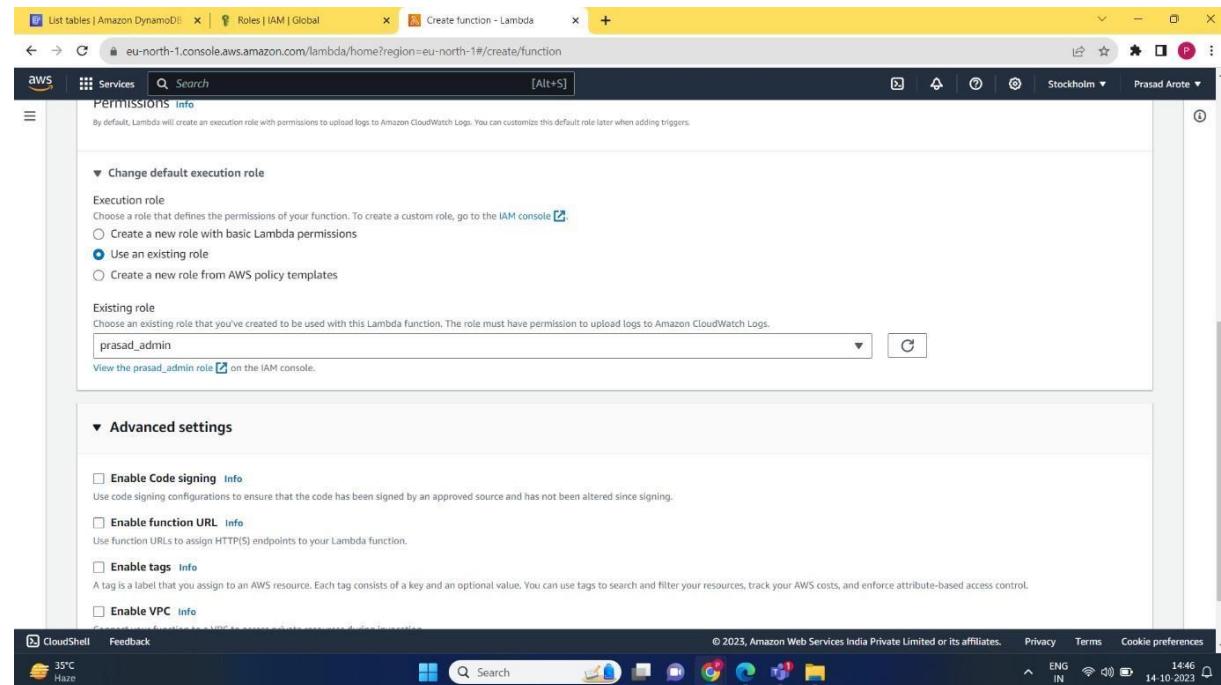
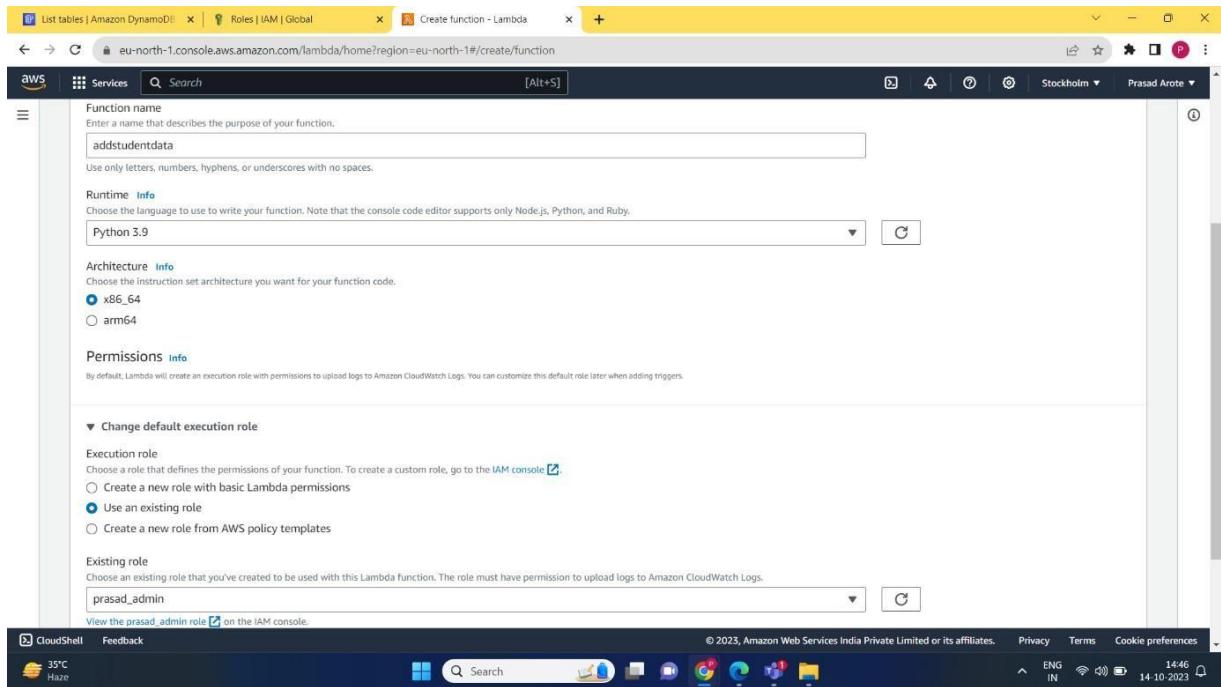
**Roles (9) Info**  
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSCloud9SSMAccessRole	AWS Service: ec2, and 1 more.	75 days ago
<input type="checkbox"/>	AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-	-
<input type="checkbox"/>	AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked)	75 days ago
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
<input type="checkbox"/>	lambdafunc1-role-11c5lj6u	AWS Service: lambda	1 hour ago
<input type="checkbox"/>	prasad_admin	AWS Service: lambda	-
<input type="checkbox"/>	PyRole	AWS Service: lambda	68 days ago
<input type="checkbox"/>	Runpython	AWS Service: lambda	68 days ago

## 4. Create a Lambda Function

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" Try searching with longer queries for more relevant results'. The results are categorized under 'Services' and 'Features'. Under 'Services', the 'Lambda' service is listed first, followed by 'CodeBuild', 'AWS Signer', and 'Amazon Inspector'. Under 'Features', 'Local processing' is listed. On the right side of the page, there is a sidebar titled 'AWS Health Info' which displays 'Open issues', 'Scheduled changes', and 'Other notifications' for the past 7 days. At the bottom of the page, there is a link to 'Go to AWS Health'.

The screenshot shows the 'Create function' wizard. The title bar says 'Create function - Lambda'. The main heading is 'Create function' with an 'Info' link. Below it, there is a sub-instruction: 'Choose one of the following options to create your function.' There are four options: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. The 'Author from scratch' option has a sub-instruction: 'Start with a simple Hello World example.' The 'Basic information' section includes fields for 'Function name' (containing 'addstudentdata') and 'Runtime' (set to 'Python 3.9'). Other sections include 'Architecture' (set to 'x86\_64') and 'Permissions' (with a note about creating an execution role). At the bottom, there are links for 'CloudShell' and 'Feedback', along with the standard AWS footer.



5. Write the following code

The screenshot shows the AWS Lambda function editor for the 'addstudentdata' function. The code source tab is selected, displaying the following Python code:

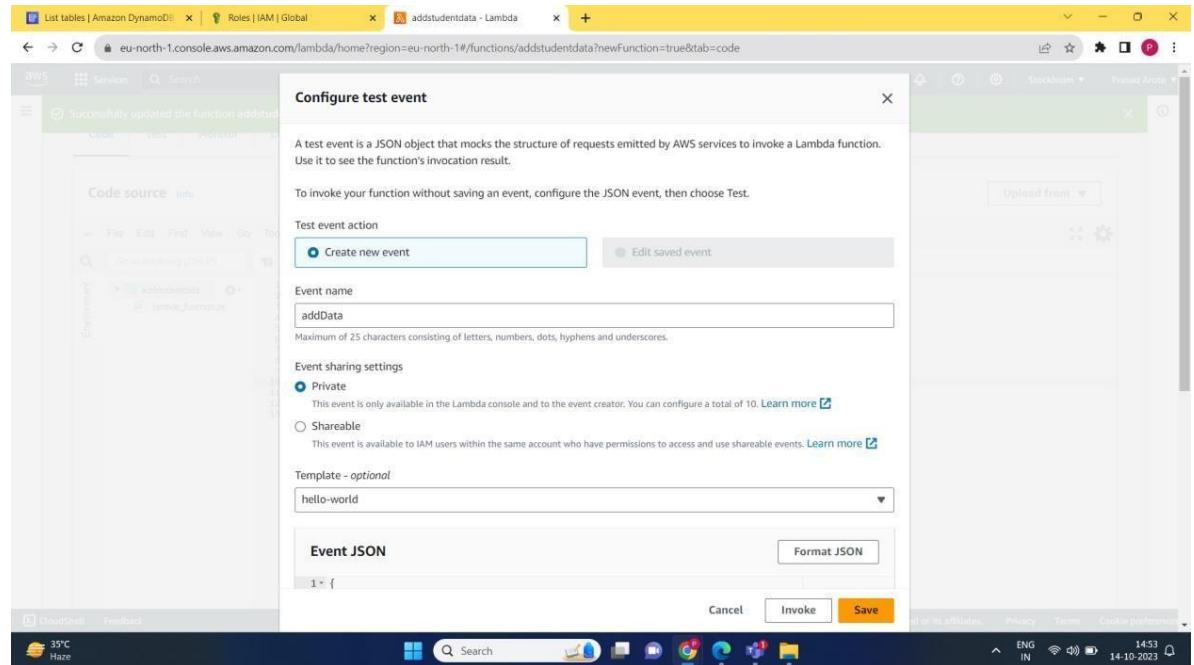
```

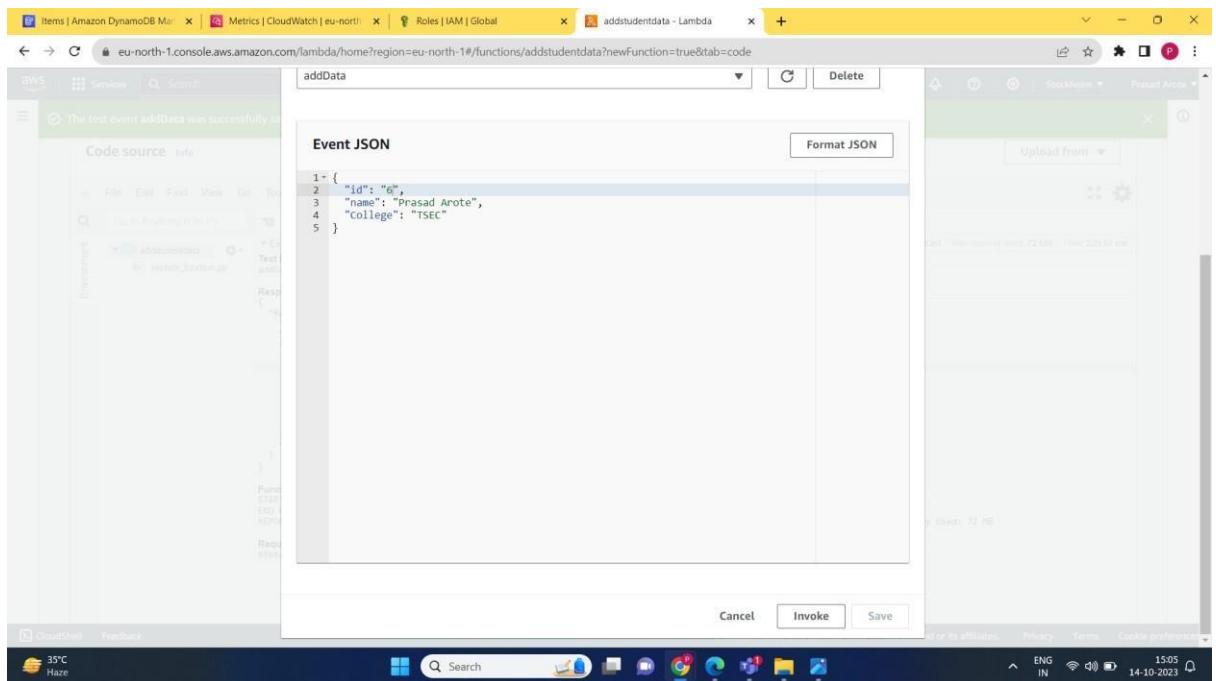
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # TODO implement
6     client_dynamo = boto3.resource('dynamodb')
7     table = client_dynamo.Table('Student')
8
9     response = table.put_item(Item=event)
10
11     return response
12
13

```

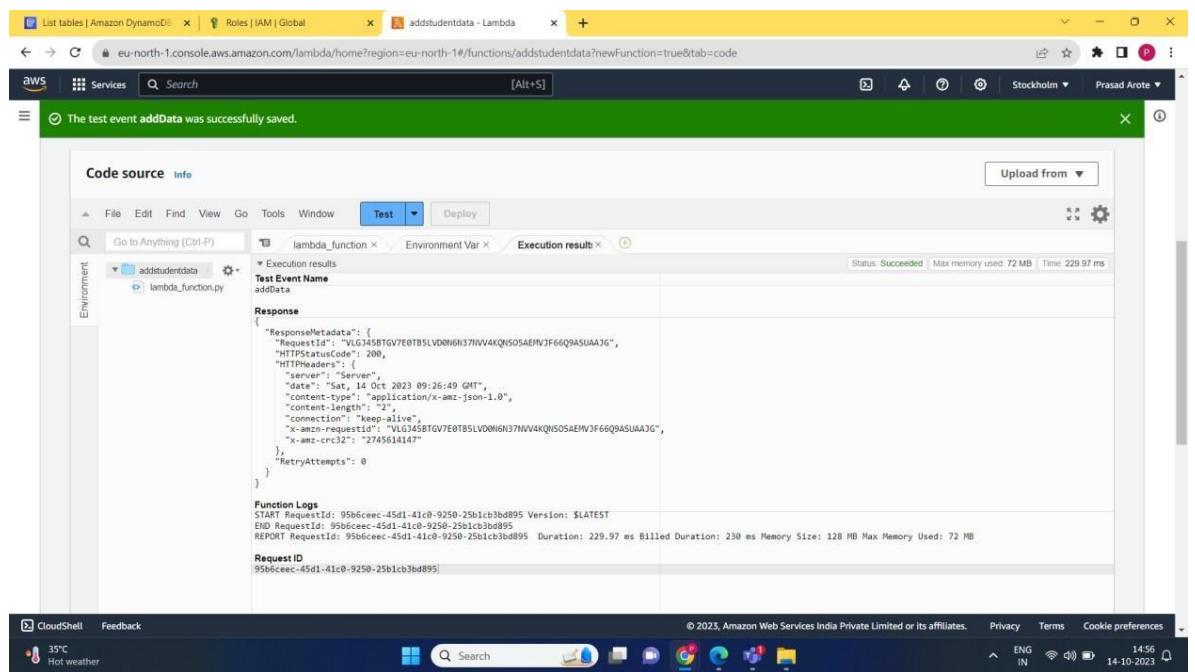
The interface includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Test tab is currently active. The configuration sidebar on the left shows the environment variables and the path 'addstudentdata / lambda\_function.py'. The bottom status bar indicates the date and time as 14-10-2023.

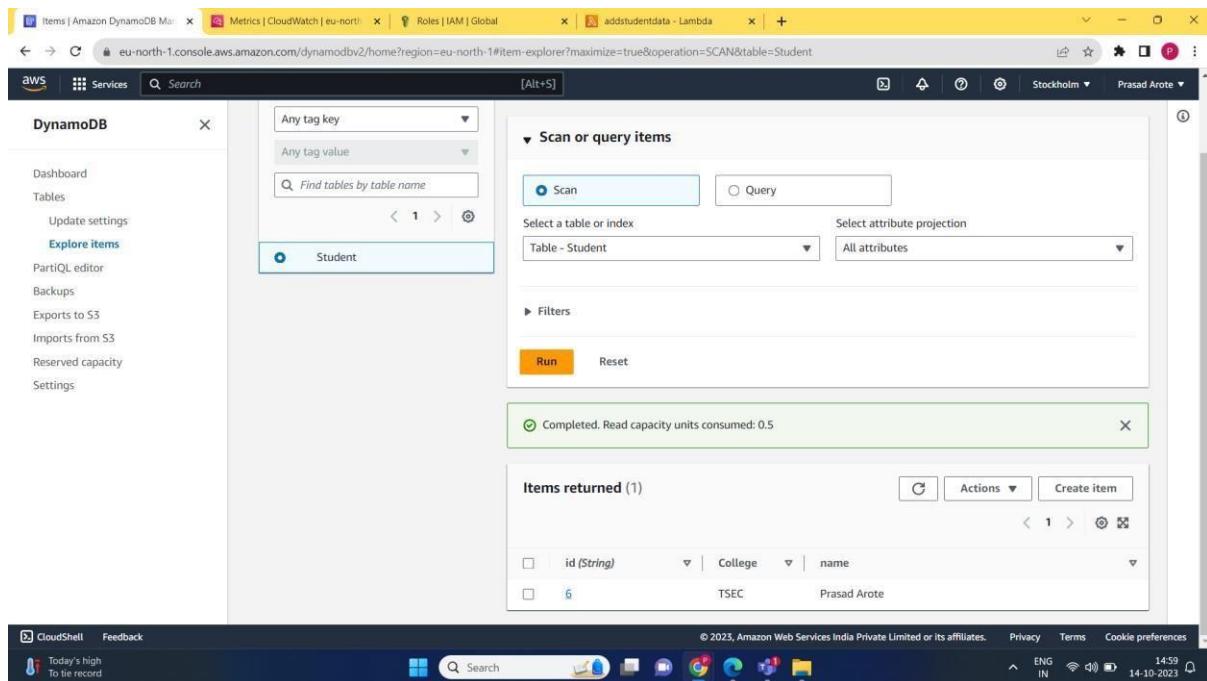
## 6. Configure test event and Save





- 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.

## **Adv. DevOps Written Assignment : 01**

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

1. Role-Based Access Control (RBAC): RBAC restricts who can perform actions within a Kubernetes cluster. It defines roles and role bindings to specify what resources and operations users or service accounts can access. This prevents unauthorized access and actions within the cluster.
2. Regular Updates: Keeping Kubernetes and its components up to date is crucial. New releases often include security patches. Regular updates help mitigate known vulnerabilities and ensure your cluster remains secure.
3. Network Policies: Network policies allow you to define rules for communication between pods. By specifying which pods can communicate with each other, you can limit the attack surface and prevent unauthorized access.
4. Container Security Tools: Employ container security tools like

vulnerability scanners to assess the security of container images. These tools can identify and remediate vulnerabilities in the containerized applications before they are deployed.

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

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

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

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

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

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

Combining these measures into a comprehensive security strategy is essential for safeguarding your Kubernetes cluster and the applications running within it. It's important to stay informed about best practices and evolving security threats in the Kubernetes ecosystem.

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

Three security techniques commonly used to protect data are:

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

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

in a database.

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

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

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

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

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

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

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

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

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

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

```
```bash
```

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

```
```
```

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

```
```yaml
```

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

```
kind: Ingress
```

```
metadata:
```

```
  name: my-ingress
```

```
spec:
```

rules:

- host: example.com

  http:

    paths:

- path: /path

      pathType: Prefix

    backend:

      service:

        name: your-service

      port:

        number: 80

  ...

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

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

```bash

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

...

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

Ingress controller.

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

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

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

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

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

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

**4. UDP:** While less common, some Ingress controllers support UDP services. UDP is a connectionless protocol used for various purposes, including DNS and VoIP. Exposing UDP services may require specific controller support.

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

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

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

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

## **Assignment 02**

**Q1:** How to deploy a Lambda function on AWS?

AWS Console: Log in to the AWS Management Console.

Lambda Service: Navigate to the Lambda service from the AWS Console.

Create Function: Click the "Create Function" button.

Author from Scratch: Choose the "Author from scratch" option to create a new Lambda function.

Function Name: Provide a unique name for your function.

Runtime: Select the runtime for your function (e.g., Python, Node.js, Java).

Execution Role: Choose an existing role or create a new one that grants the necessary permissions to your function.

Function Code: Upload your code package or write code directly in the inline code editor.

Configuration: Set function-specific configurations like memory, timeout, and VPC settings.

Create Function: Click the "Create Function" button to deploy your Lambda function on AWS.

**Q2:** What are the deployment options for AWS Lambda?

Manual Deployment: You can manually create a Lambda function through the AWS Management Console by specifying function details and uploading code.

CLI Deployment: Use the AWS Command Line Interface (CLI) to deploy Lambda functions by running commands like `aws lambda create-function`.

Deployment Packages: You can package your function code and its dependencies into a deployment package (ZIP file) and upload it to AWS Lambda.

Integration with AWS Services: Deploy Lambda functions in response to various AWS events using integrations like S3 triggers, API Gateway, or Amazon SNS.

Continuous Integration/Continuous Deployment (CI/CD): Integrate Lambda deployments into your CI/CD pipeline using tools like AWS CodePipeline or AWS SAM (Serverless Application Model).

Infrastructure as Code (IaC): Define Lambda functions and their configurations in infrastructure as code templates using AWS CloudFormation or the AWS Serverless Application Model (SAM).

**Serverless Framework:** Use frameworks like the Serverless Framework to simplify the deployment of Lambda functions and their associated resources.

**AWS SAM:** AWS Serverless Application Model (SAM) is a framework specifically designed for serverless application deployment, including Lambda functions.

**AWS CDK:** The AWS Cloud Development Kit (CDK) allows developers to define Lambda functions using familiar programming languages and then deploy them programmatically.

**Third-Party Tools:** Various third-party tools and services offer deployment options for Lambda, such as the Serverless Framework, Terraform, and more.

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

**Full Copy Deployment Mode:** In this mode, all the resources in your AWS Lambda function, including code, configuration, and dependencies, are copied to a new version or alias of the function. This ensures that the deployed function is isolated from the original function and won't be affected by changes made to the original.

**Immutable Deployment Mode:** Immutable deployments involve creating a new version of your Lambda function with the updated code and configurations, leaving the original function unchanged. This ensures that the existing function remains stable while the new version is deployed. If any issues arise, you can quickly roll back to the previous version.

**Traffic Split Deployment Mode:** In this mode, traffic is gradually shifted from one version of your Lambda function to another. You can specify the percentage of traffic to route to the new version, allowing you to perform canary deployments and monitor the new version's performance before fully deploying it.

**Q4:** What are the 3 components of AWS Lambda?

**1. Function Code:** This is the code that defines the logic and behaviour of your AWS Lambda function. You can write your function code in programming languages such as Node.js, Python, Java,

C#, Ruby, Go, or use custom runtimes. The function code is packaged and uploaded as a deployment package, and it's what gets executed when the Lambda function is invoked.

**2. Event Source:** An event source is a trigger that invokes your Lambda function. AWS Lambda

can be triggered by various events, such as changes in an S3 bucket, incoming HTTP requests via API Gateway, database updates in DynamoDB, scheduled events from CloudWatch, and more. The event source determines when and why your Lambda function should run.

3. Execution Role: An execution role is an AWS Identity and Access Management (IAM) role that grants permissions to your Lambda function. This role defines what AWS resources your Lambda

function can interact with and what actions it can perform. It's essential for security and access control.

These three components work together to create a functional AWS Lambda function. When an event occurs that matches the trigger defined by the event source, AWS Lambda executes your function

code, using the execution role to access other AWS resources as needed.