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

Agarwal

Teacher In-Charge

Head of the Department

Date 21/10/23

Principal

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

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 hosting static web application.		26/07/23	
4.	To study AWS cloud pipeline and deploy web application using cloud pipeline		2/8/23	Agarwal 27/10/23
5.	To understand Kubernetes cluster and architecture		16/8/23	
6.	To understand terraform life cycle and to build, change and destroy AWS using Terraform.		23/08/23	
7.	To prepare static analysis on python program using donar cube AST process		13/09/23	
8	To understand continuous hosting using Nagios		02/08/23	
9.	To understand AWS lambda function and write a lambda function using python to log "an image has been added" when one a		27/09/23	

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
9.	Once a file added to S3 bucket		27/09/23	
10.	To create a lambda function using python for adding data to dynamic databases.		27/09/23	Agarwal 21/10/23
11.	Assignment 1		26/07/23	
12.	Assignment 2		11/10/23	

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:

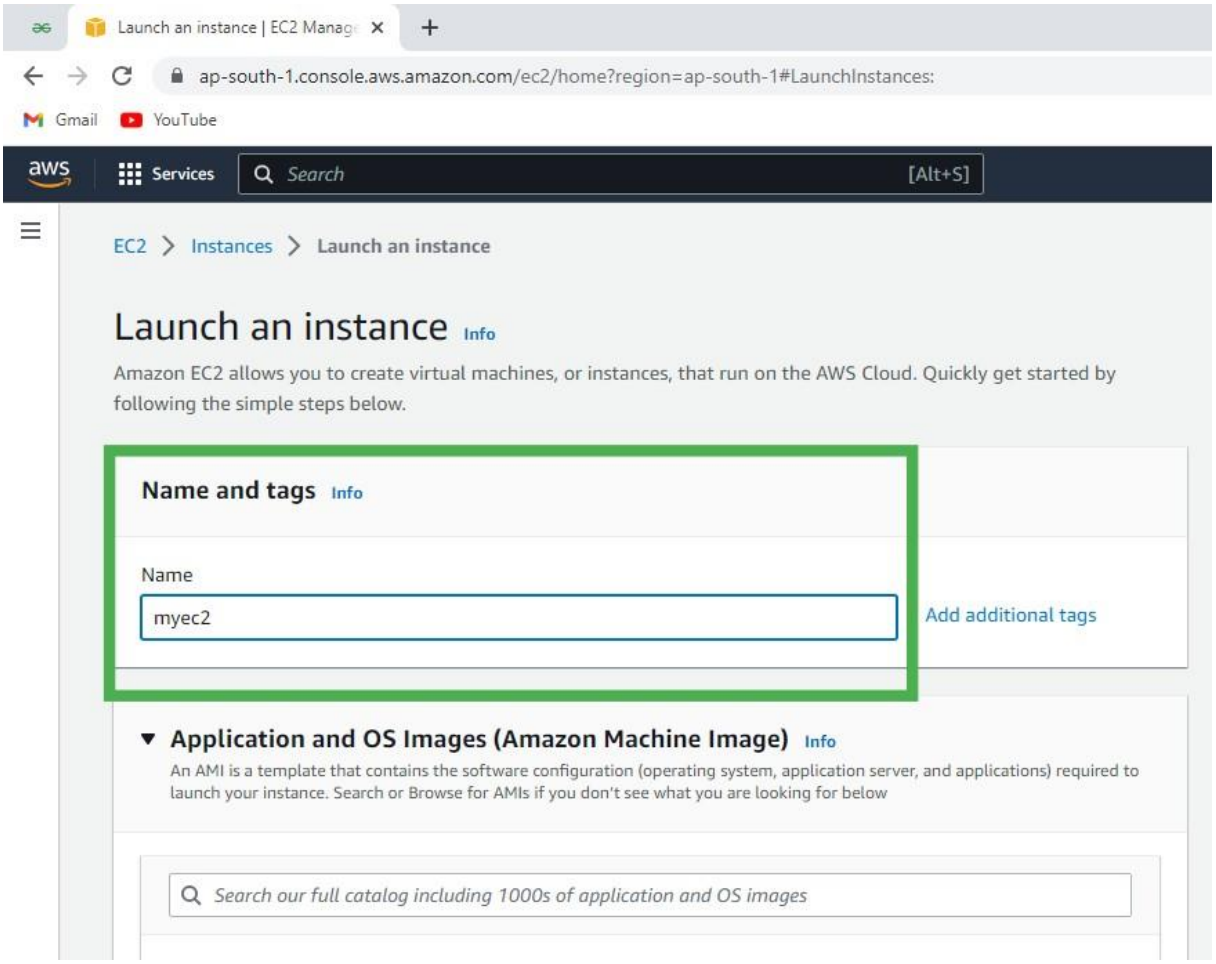
- After your instance is launched, you can view its details, including its public IP address and other information, on the EC2 Dashboard.

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.



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.

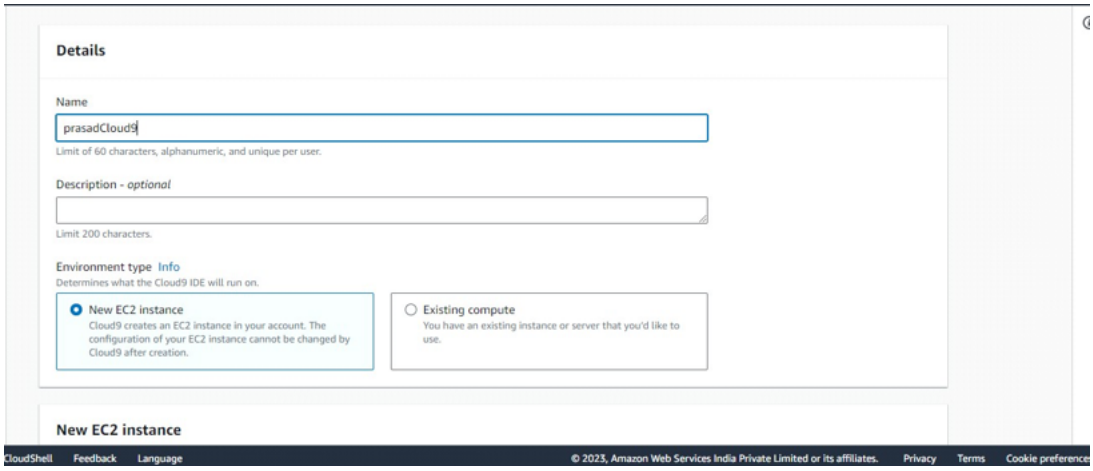
STEPS-

LOG IN TO YOUR AWS ACCOUNT,

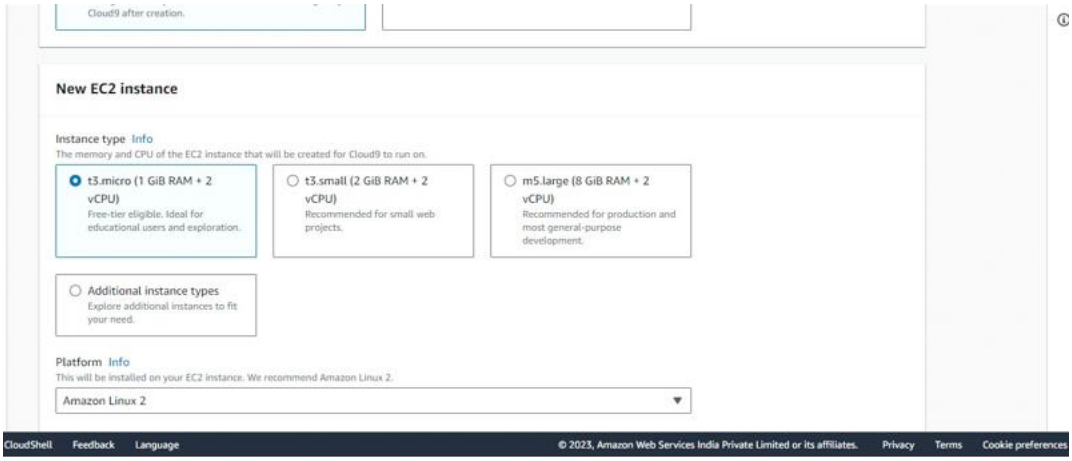
SEARCH FOR CLOUD 9 IN THE SEARCH BAR

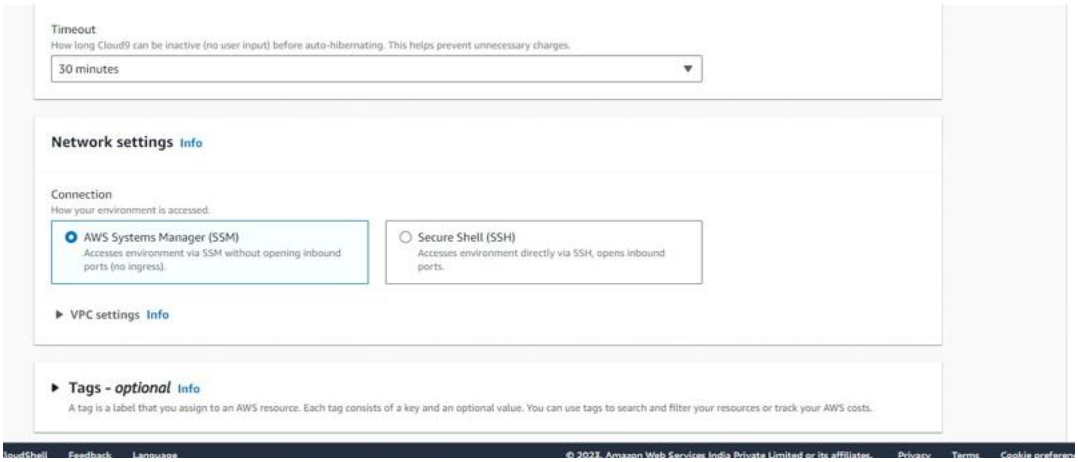
CLICK ON CREATE ENVIRONMNET,

NAME THE ENVIRONMNET

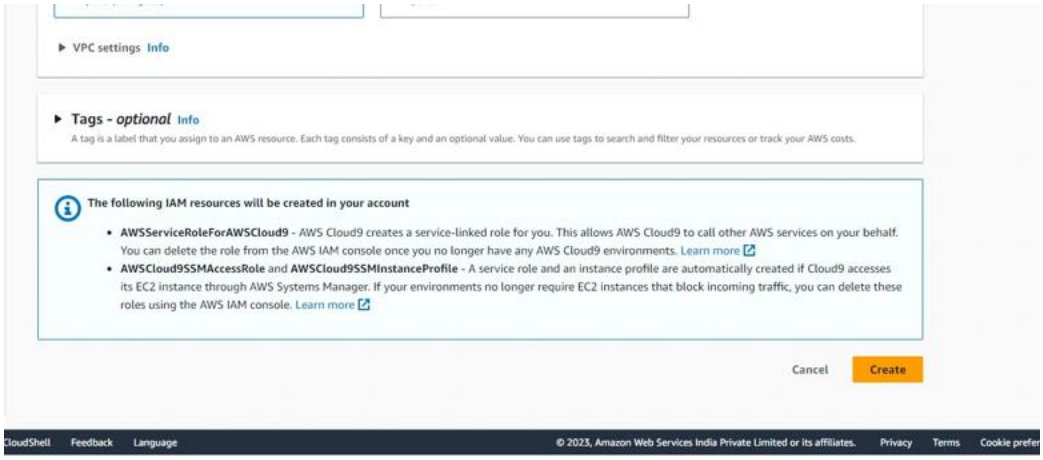


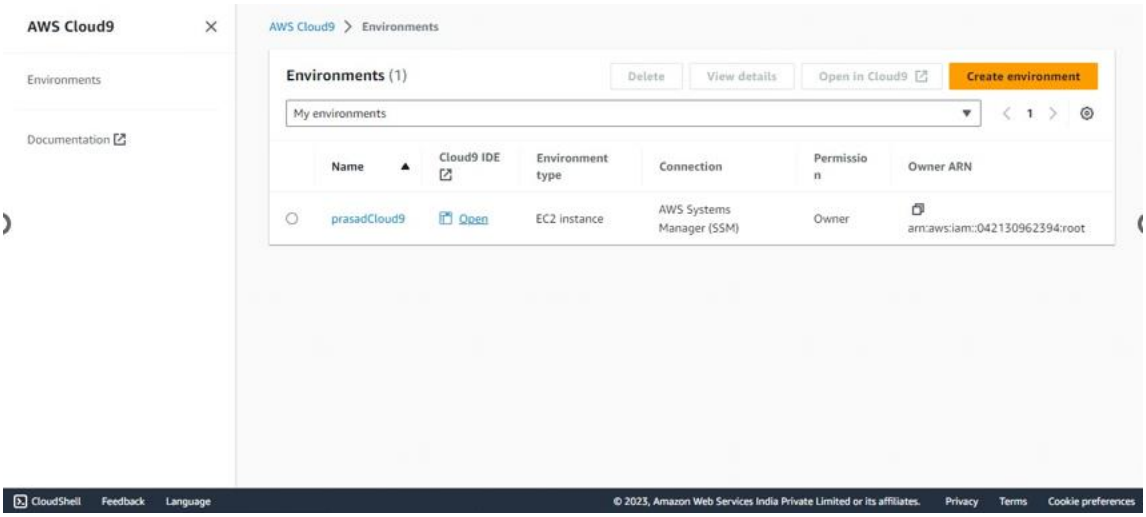
Now click on next step.



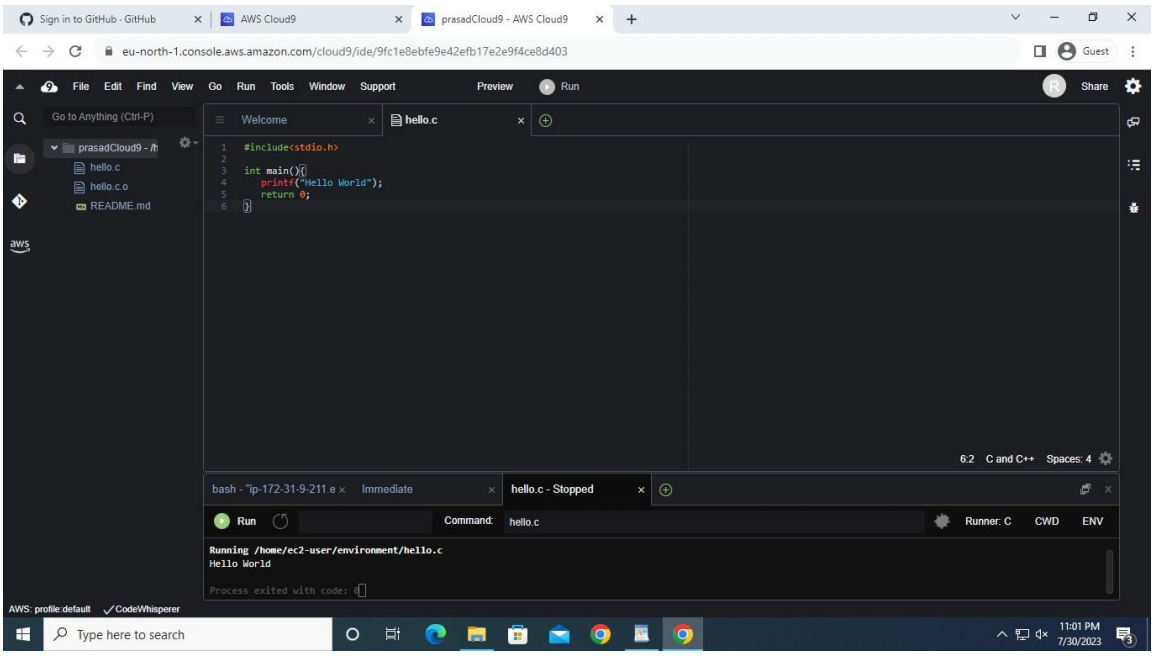


Again, click on Next Step,
Now click on Create Environment.





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



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.

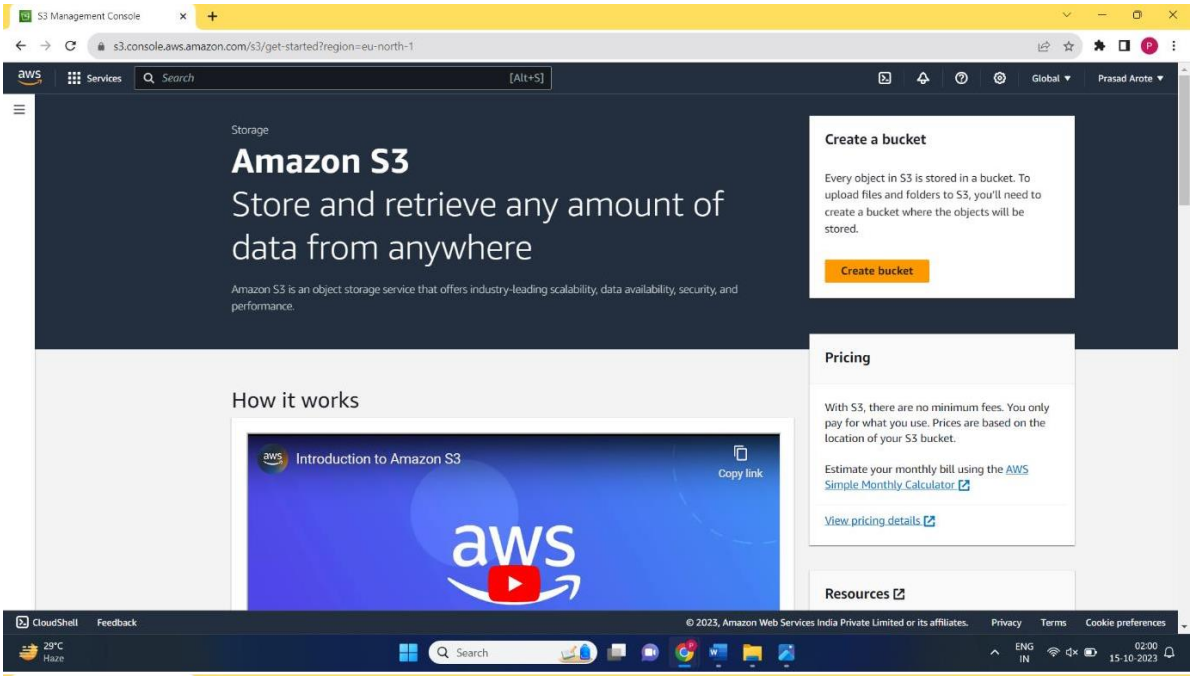
LAB ASSIGNMENT 3

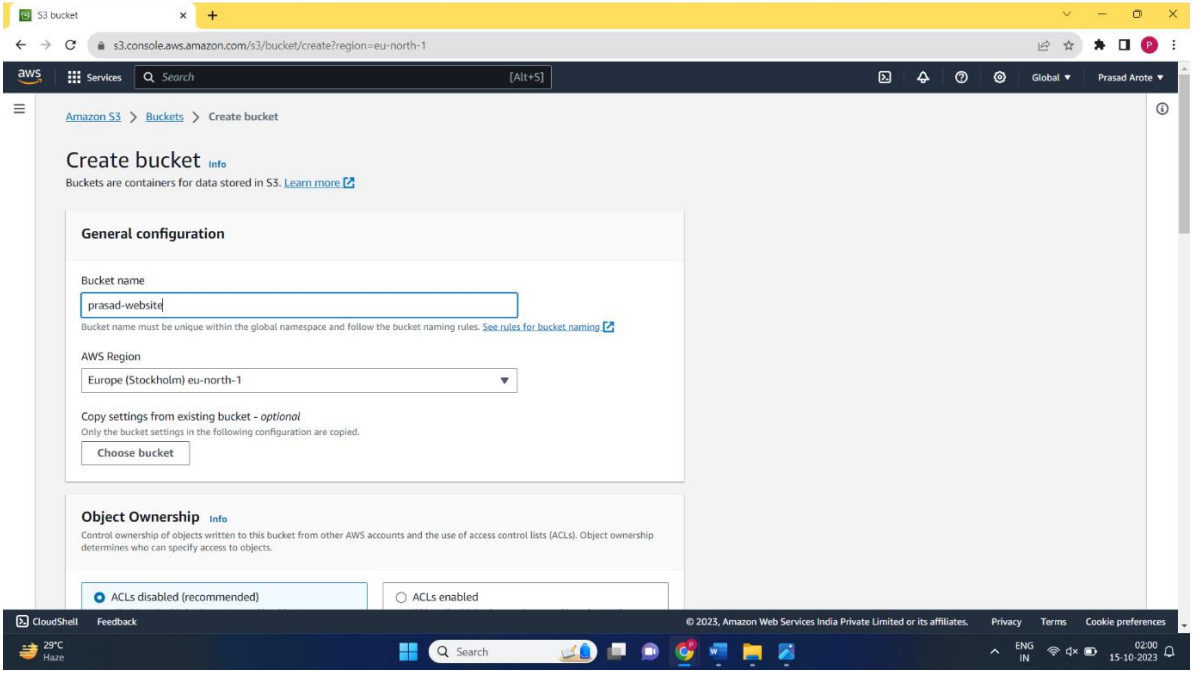
AIM: To study AWS S3 service and create a bucket for hosting static web application.

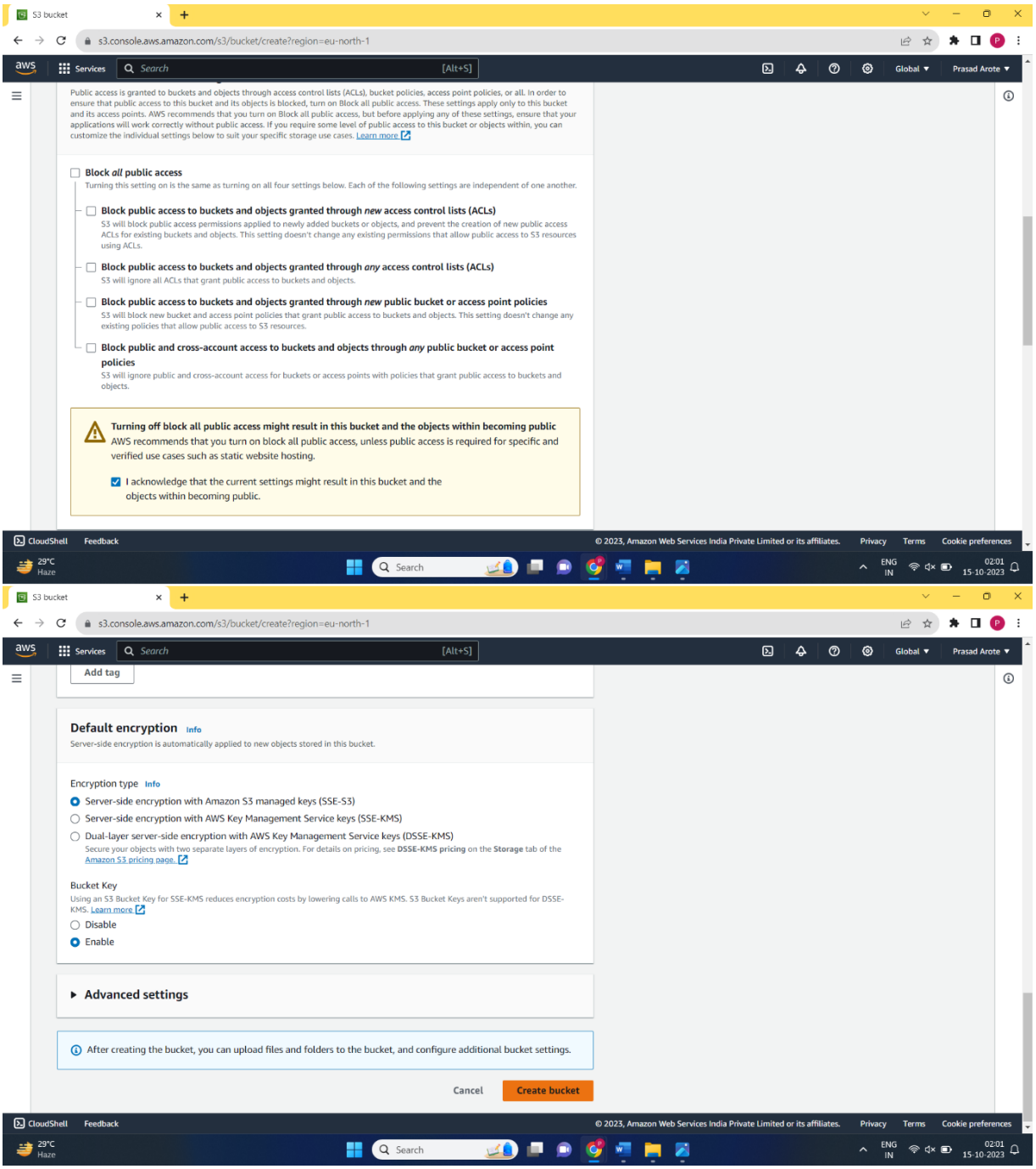
LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.

THEORY:

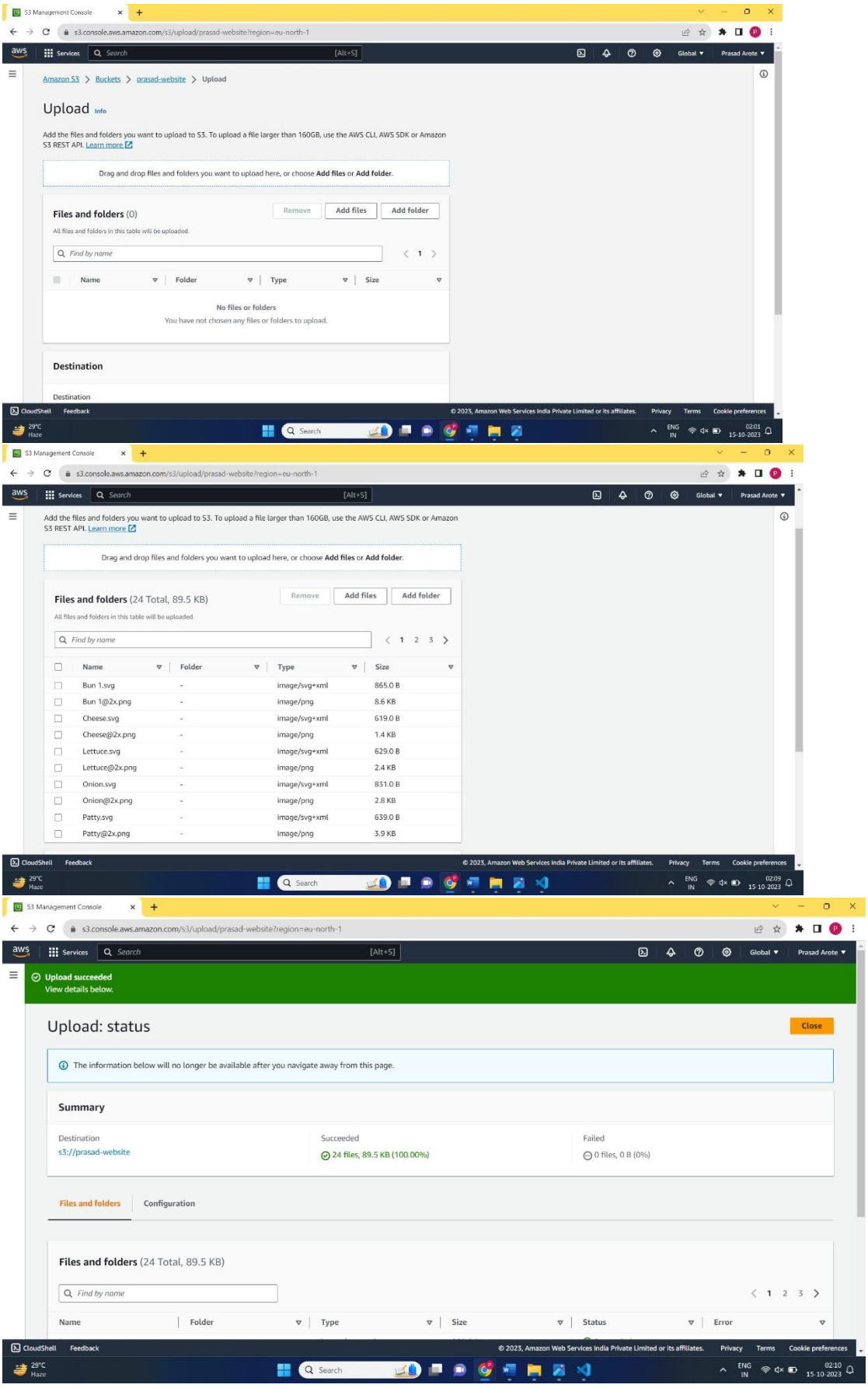
1. Create a S3 bucket.



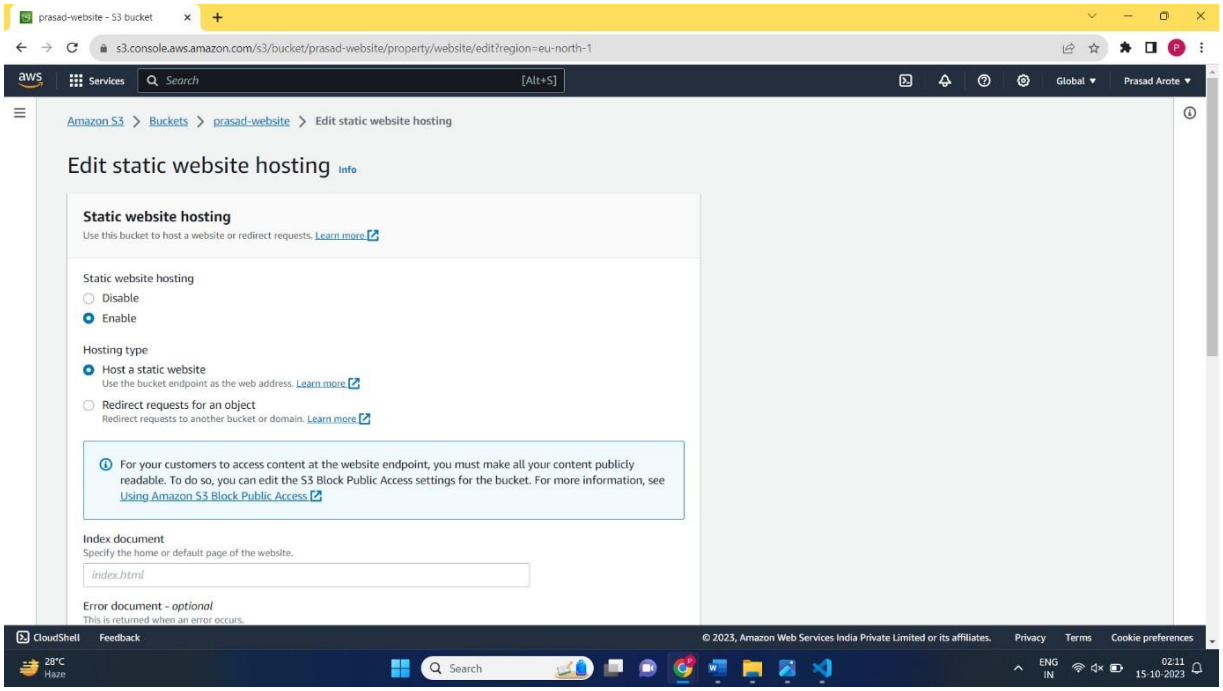


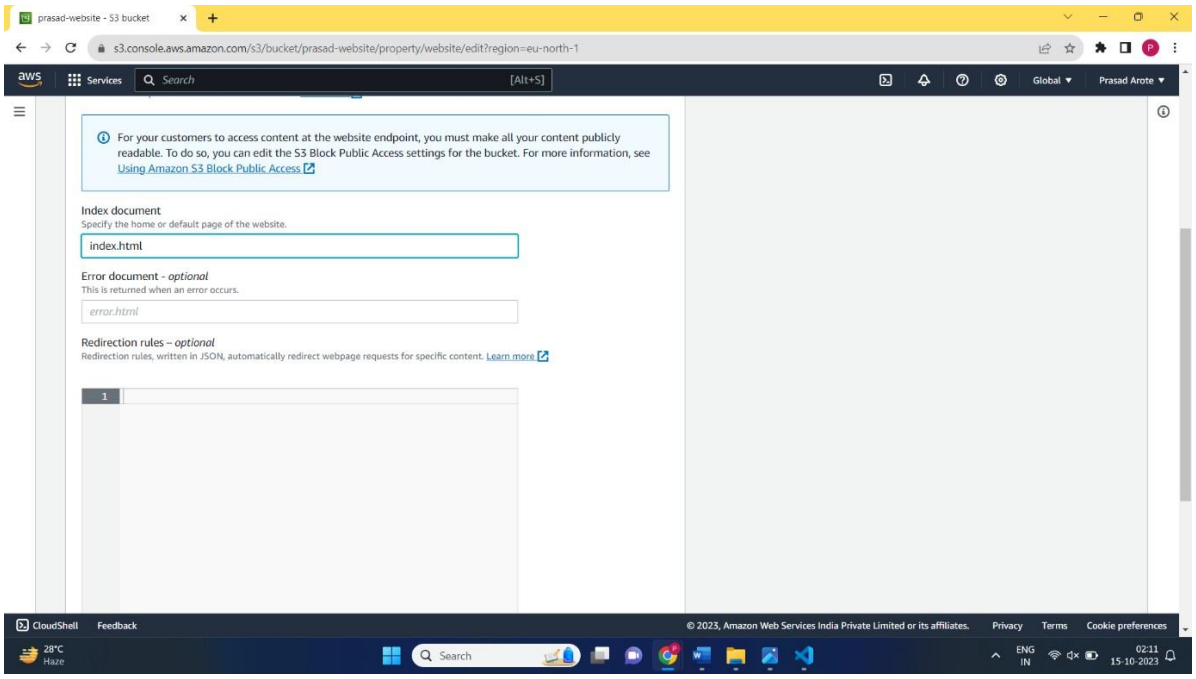


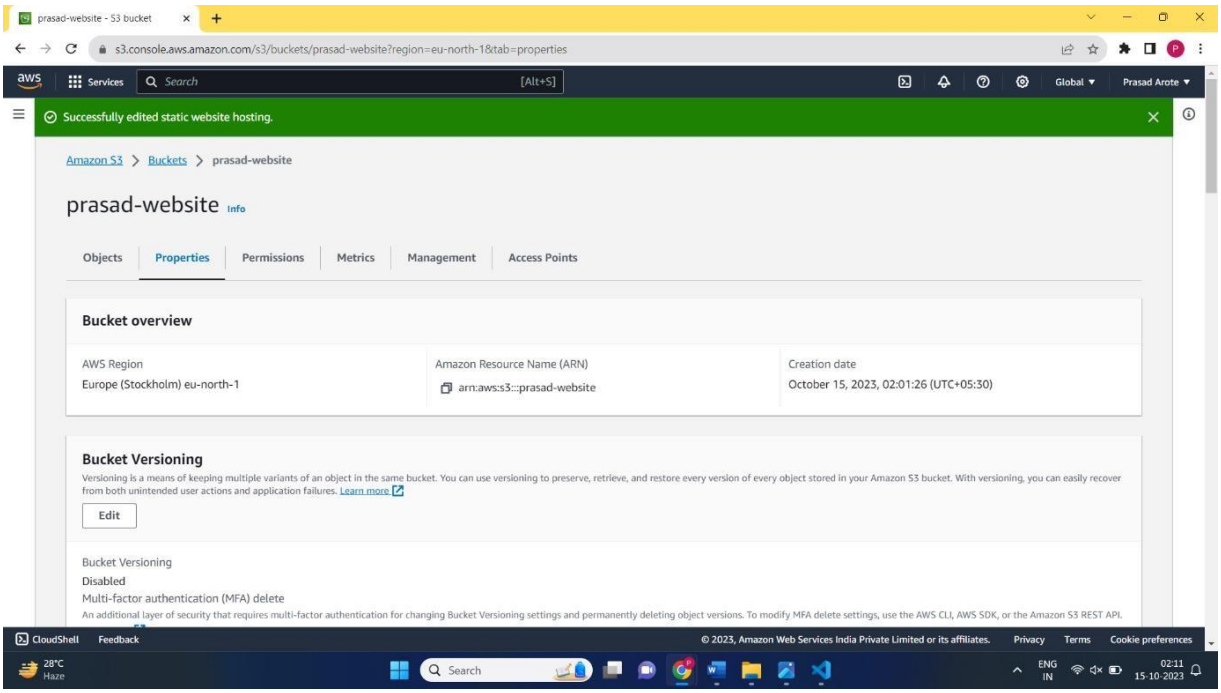
2. Upload the files of web application.


S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
Amazon S3 > Buckets > prasad-website > Upload
Upload
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (0)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
Name
Folder
Type
Size
No files or folders
You have not chosen any files or folders to upload.
Destination
Destination
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Files and folders (24 Total, 89.5 KB)
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
Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://prasad-website
Succeeded
24 files, 89.5 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Status
Error

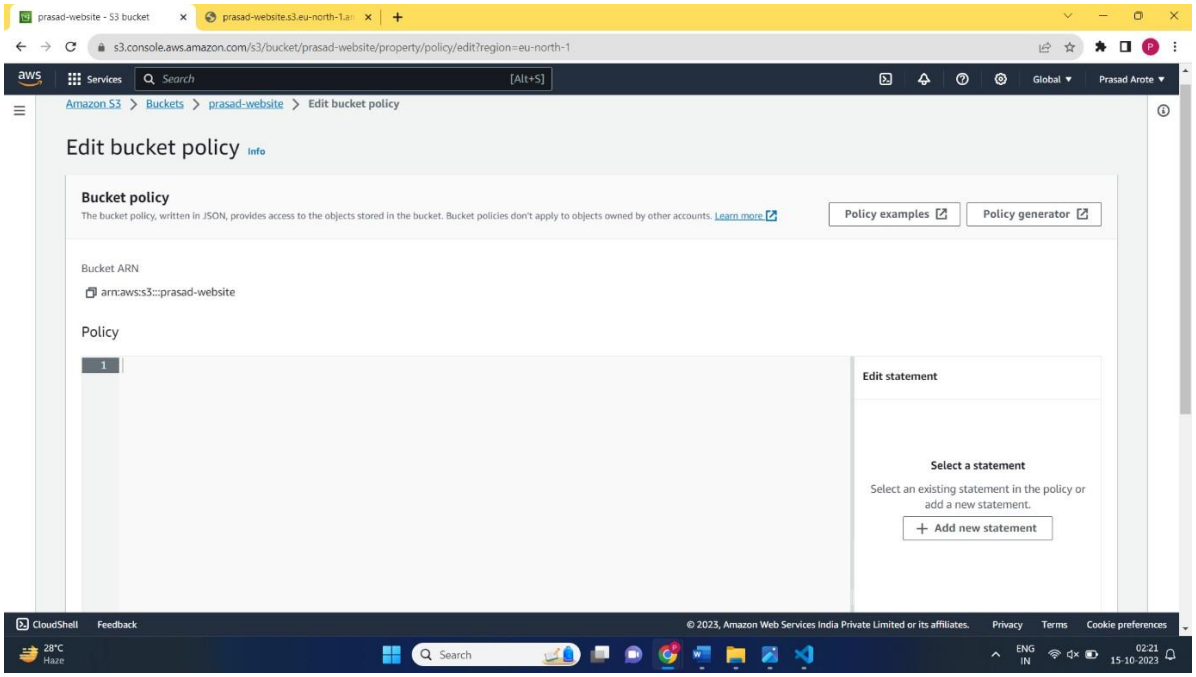
3. Enable Static website hosting

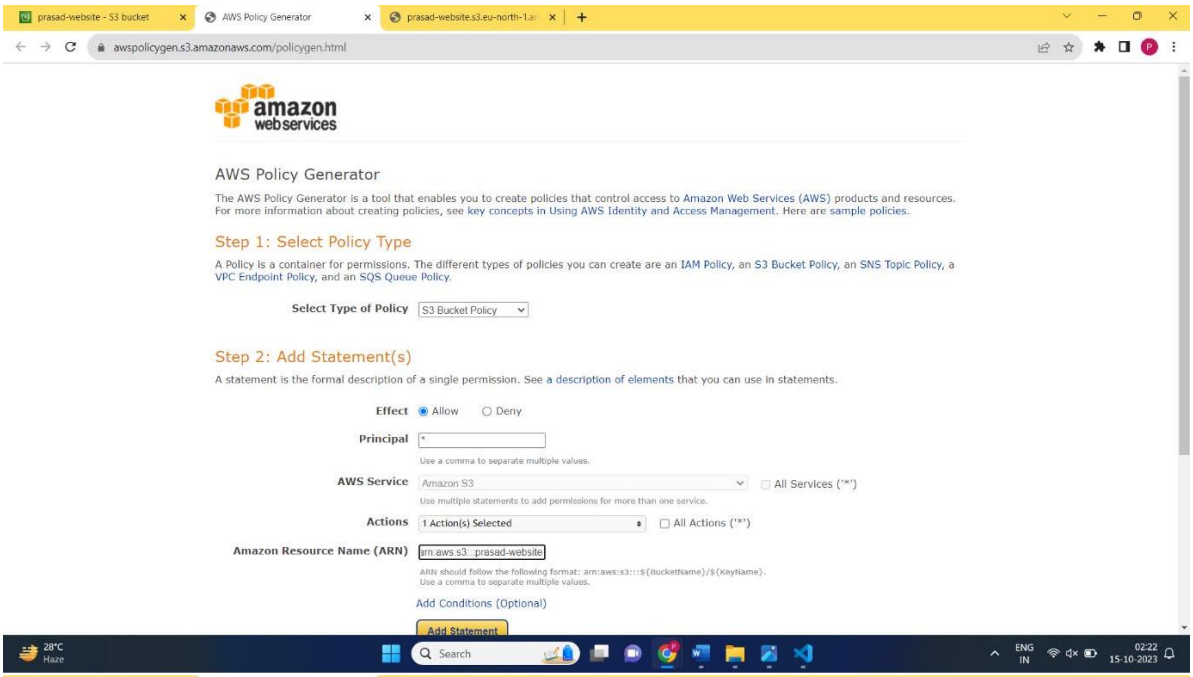


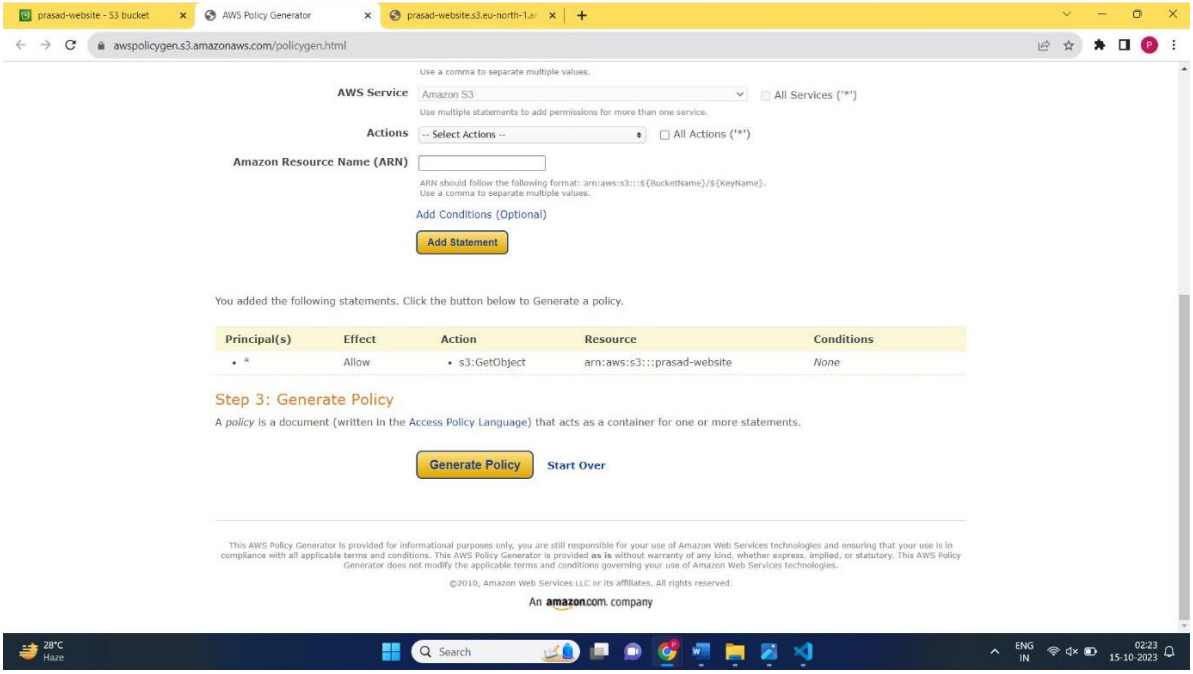


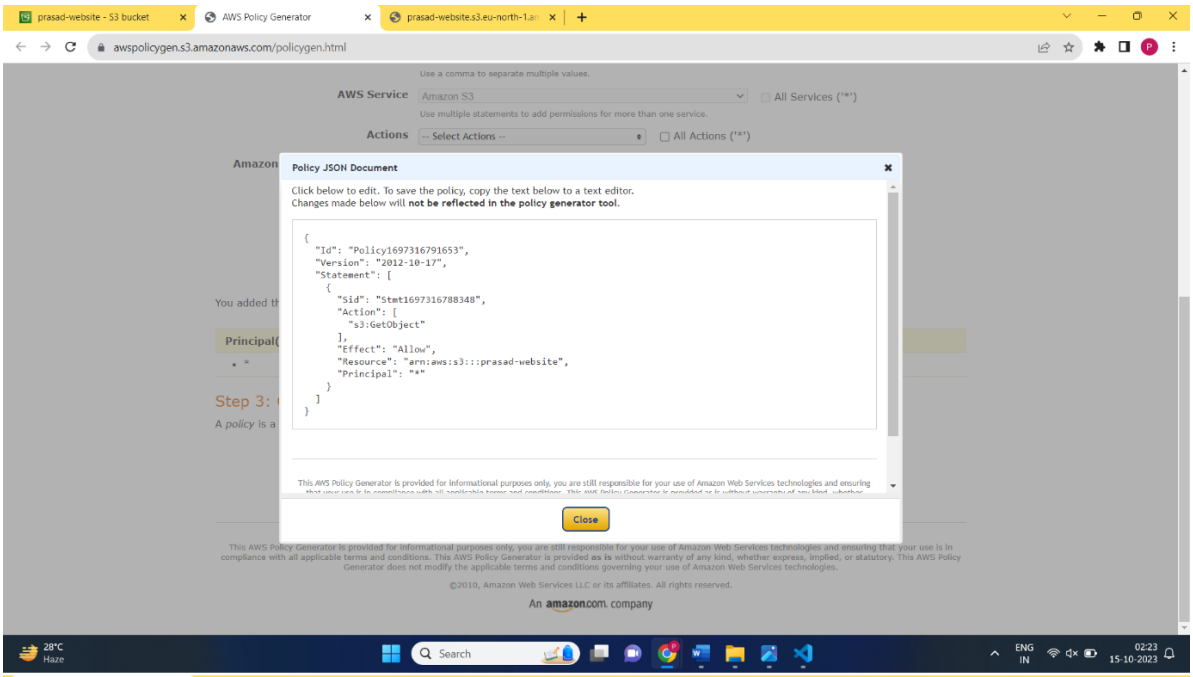


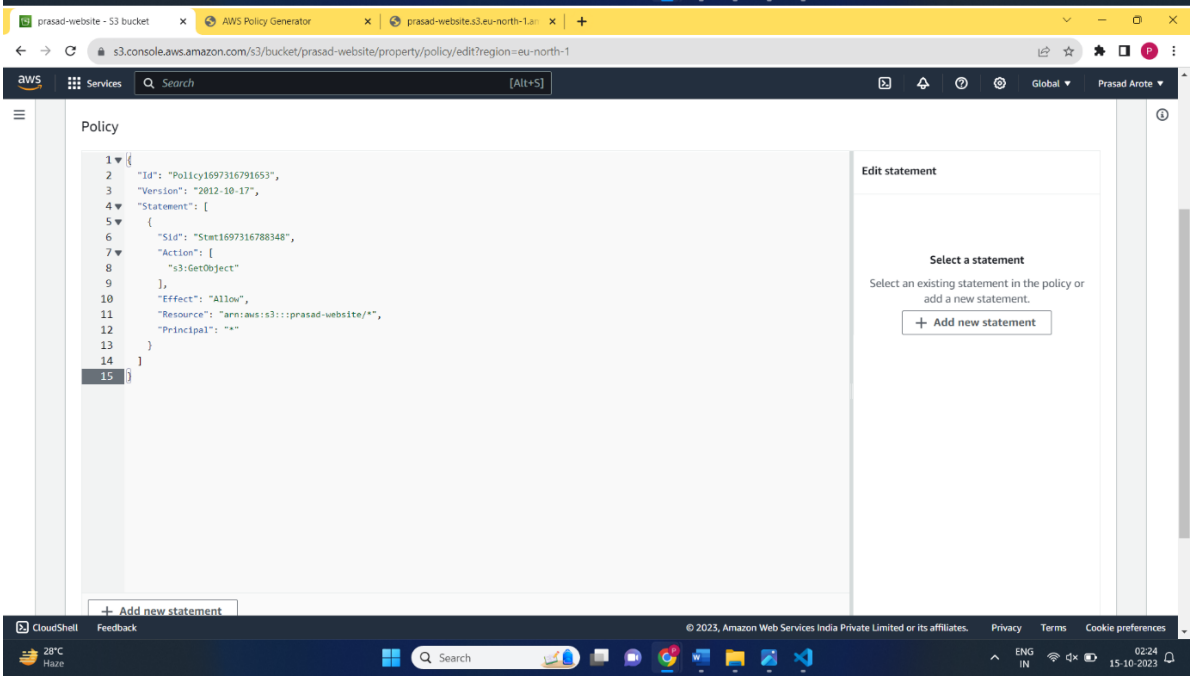
5. Change the Bucket Policy

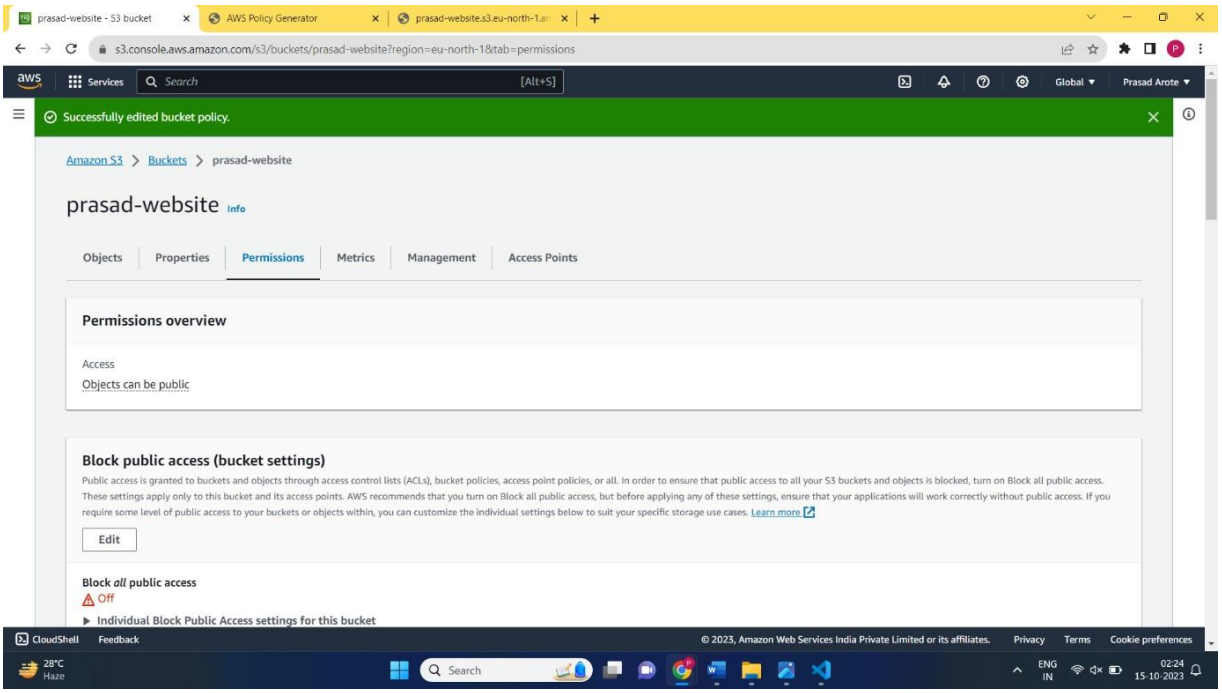



prasad-website - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
amazon web services
AWS Policy Generator
The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about 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
S3 Bucket 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
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
1 Action(s) Selected
All Actions ('*')
Amazon Resource Name (ARN)
arn:aws:s3:::prasad-website
ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.
Add Conditions (Optional)
Add Statement
Search

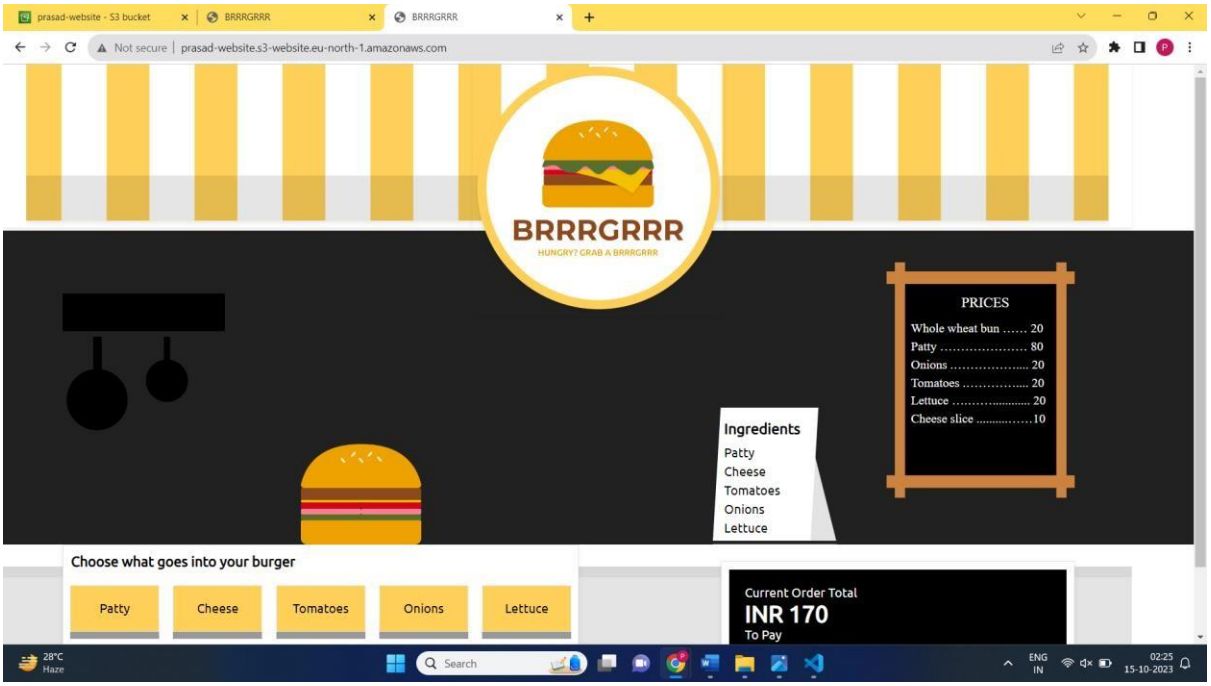

prasad-website - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
Use a comma to separate multiple values.
AWS Service
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
-- Select Actions --
All Actions ('*')
Amazon Resource 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
Search


prasad-website - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
AWS Service
Amazon S3
All Services ('*')
Actions
-- Select Actions --
All Actions ('*')
Policy JSON Document
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.
{
"Id": "Policy1697316791653",
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Stmt1697316788348",
"Action": [
"s3:GetObject"
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::prasad-website",
"Principal": "*"
}
]
}
Close
Step 3:
This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring
©2010, Amazon Web Services LLC or its affiliates. All rights reserved.
An amazon.com company
28°C
Haze
Search
ENG
IN
02:23
15-10-2023


prasad-website - S3 bucket
AWS Policy Generator
s3.console.aws.amazon.com/s3/bucket/prasad-website/property/policy/edit?region=eu-north-1
aws
Services
Search
[Alt+S]
Global
Prasad Arote
Policy
{
"Id": "Policy1697316791653",
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Stmt1697316788348",
"Action": [
"s3:GetObject"
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::prasad-website/*",
"Principal": "*"
}
]
}
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
+ Add new statement
+ Add new statement
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Haze
Search
ENG
IN
02:24
15-10-2023



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



CONCLUSION:

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

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.

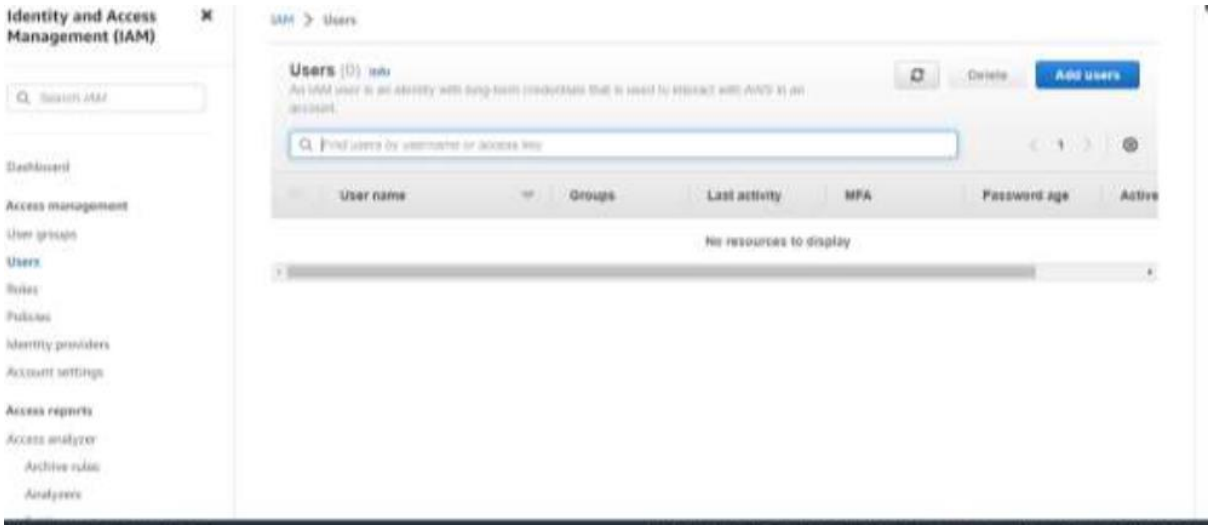
10. Trigger a manual or automatic pipeline execution for testing.

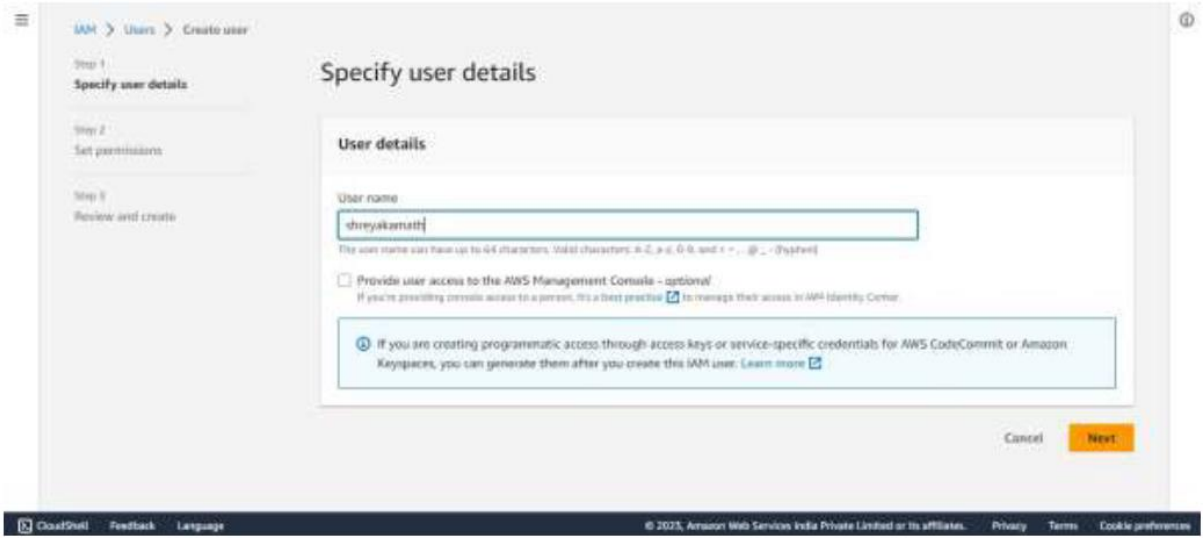
11. Monitor pipeline executions for success or failures.

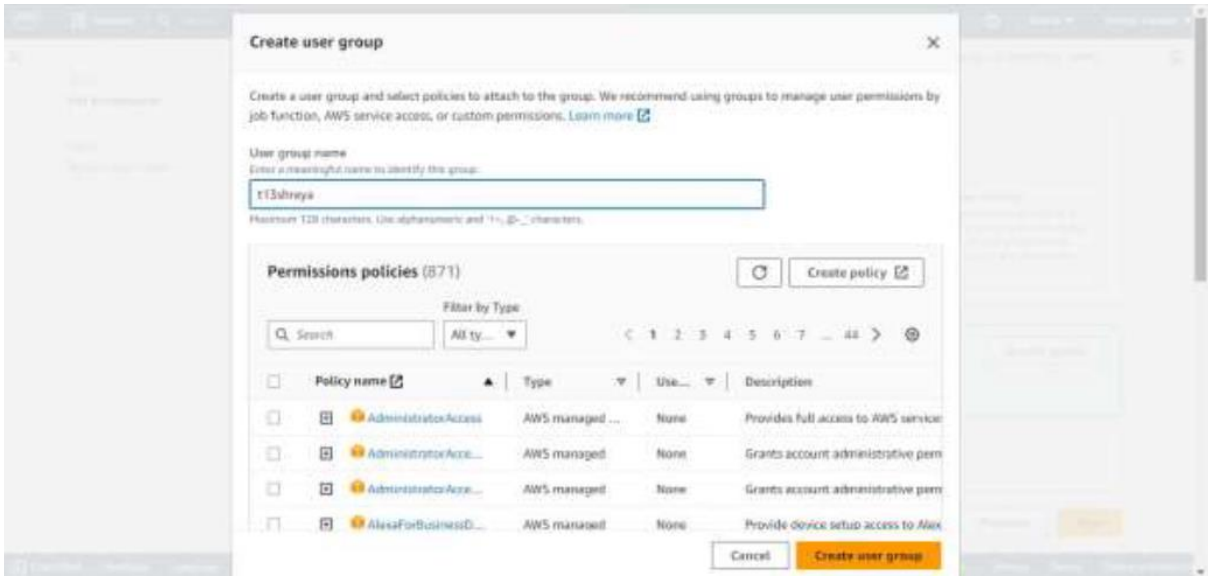
12. Enhance your pipeline as needed for testing, monitoring, or notifications.

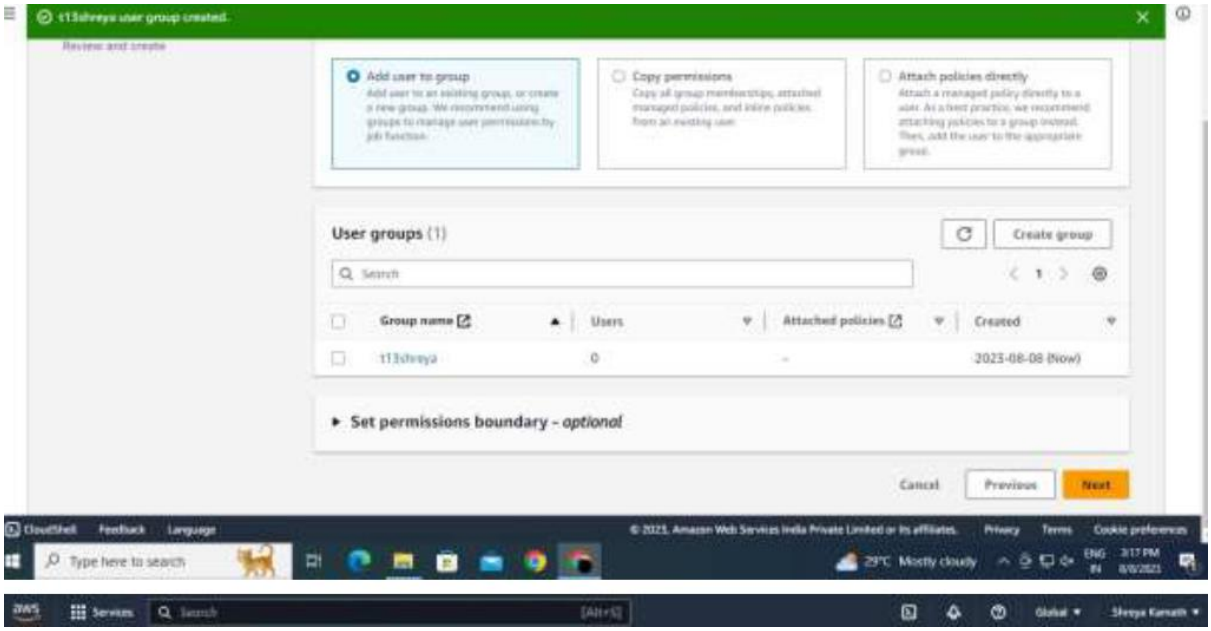
13. Optionally, set up a custom domain with Amazon Route 53 and CloudFront.

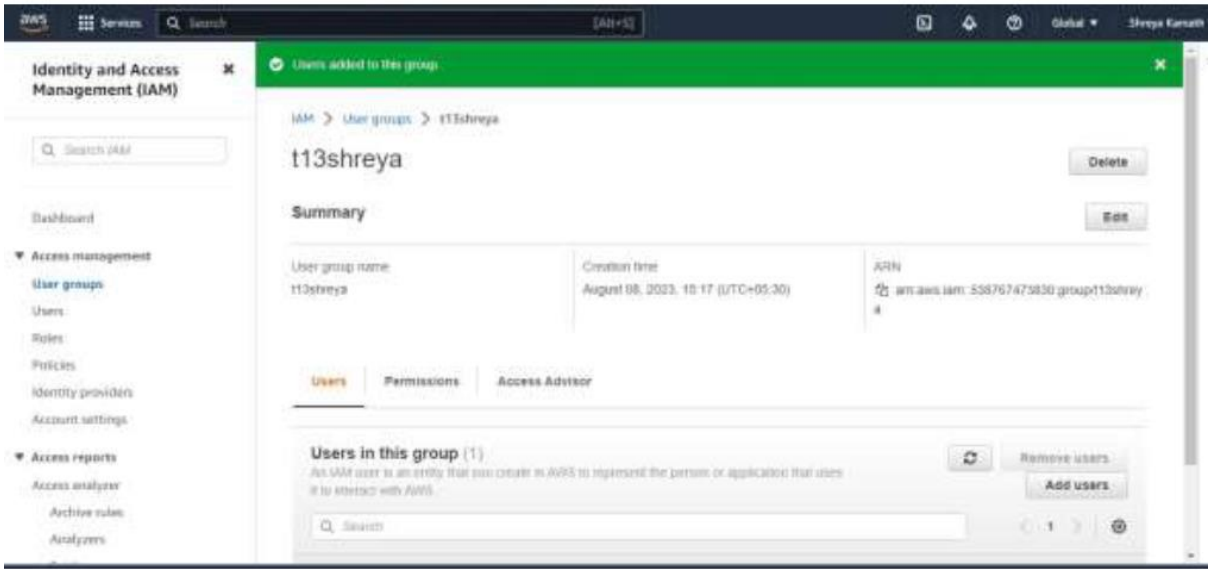
SCREENSHOTS:

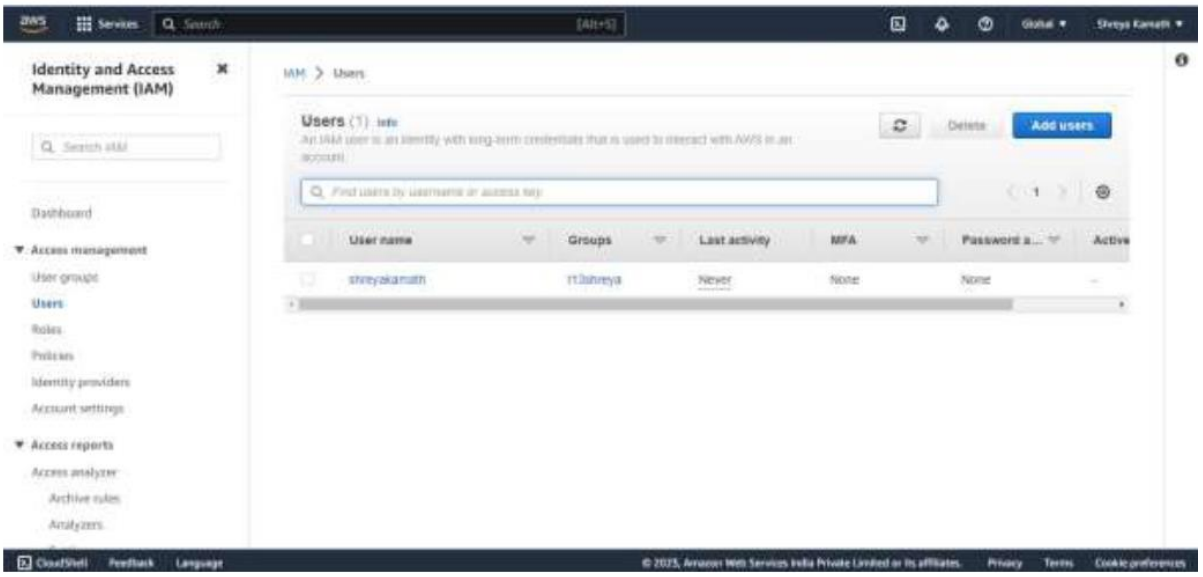


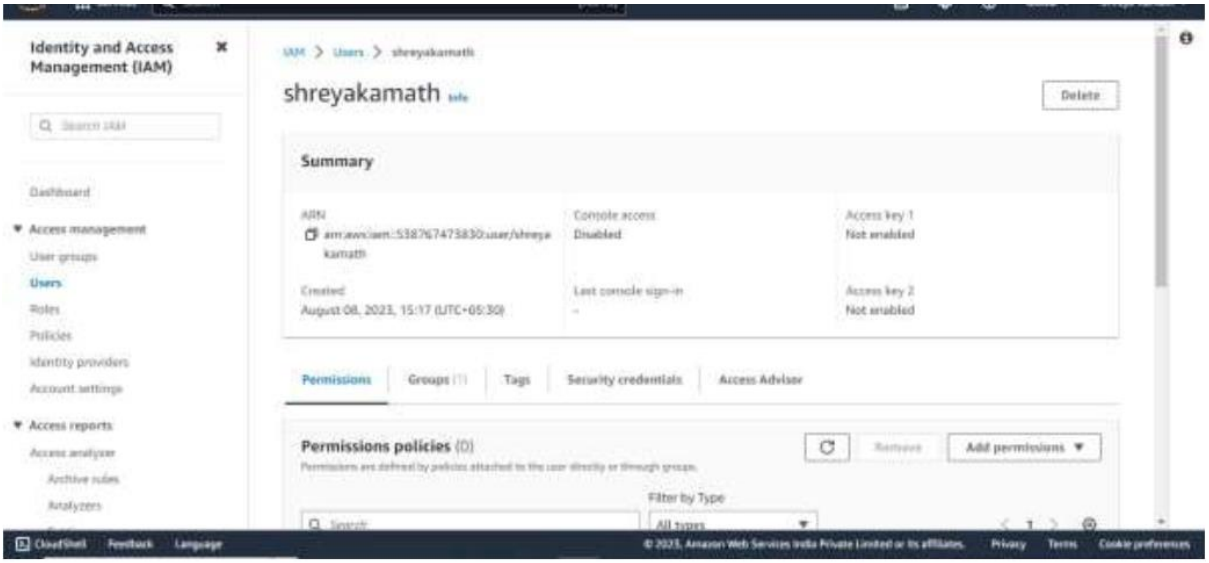


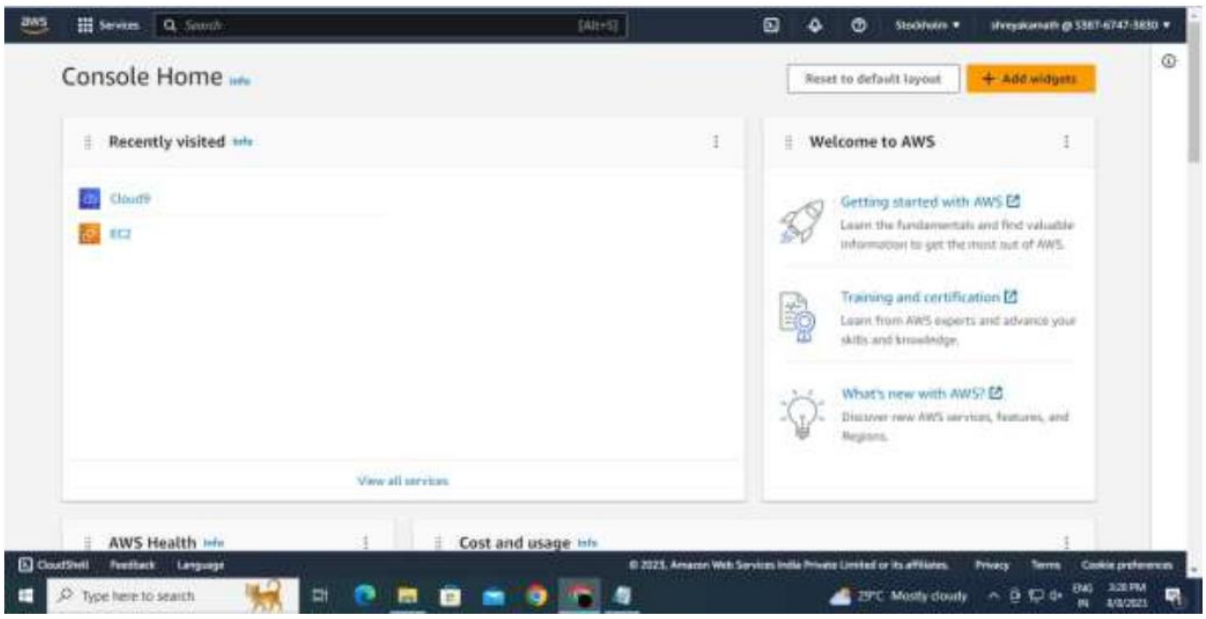

Create user group
Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more
User group name
t13shreya
Permissions policies (871)
Create policy
Filter by Type
Search
All ty...
Policy name
Type
Use...
Description
AdministratorAccess
AWS managed ...
None
Provides full access to AWS service
AWS managed
Grants account administrative perm
Cancel
Create user group

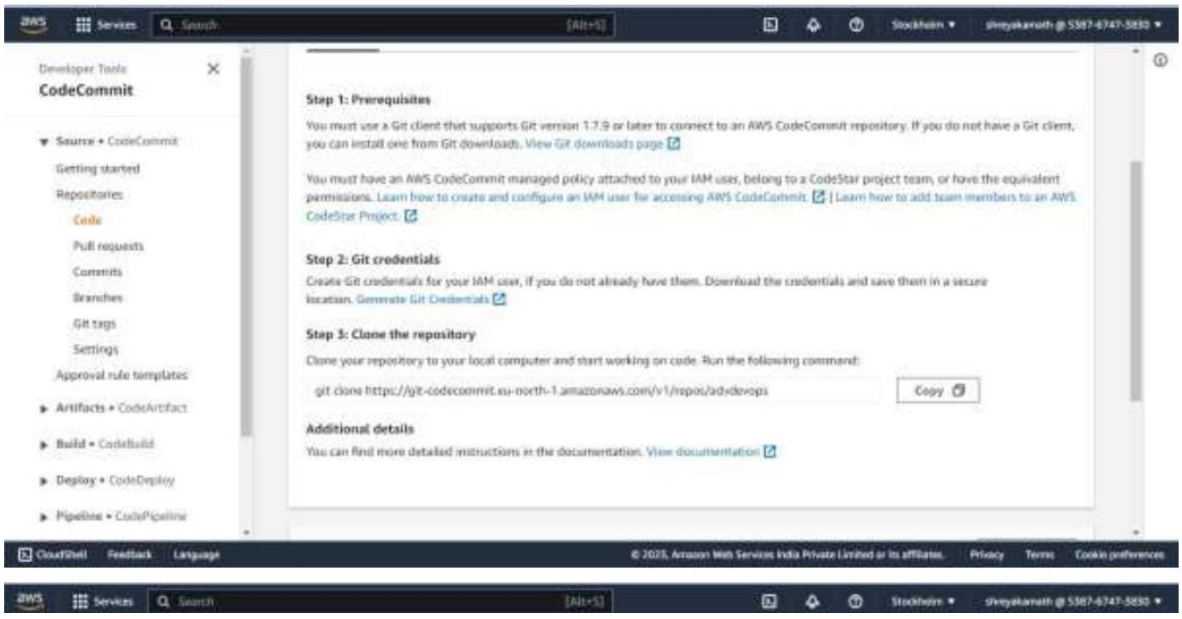

t13shreya user group created.
Review and create
Add user to group
Copy permissions
Attach policies directly
User groups (1)
Create group
Search
Group name
Users
Attached policies
Created
t13shreya
0
2023-08-08 (Now)
Set permissions boundary - optional
Cancel
Previous
Next
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
29°C Mostly cloudy
ENG
IN
3:17 PM
8/8/2023
aws
Services
Search
[Alt+S]
Global
Shreya Kamath

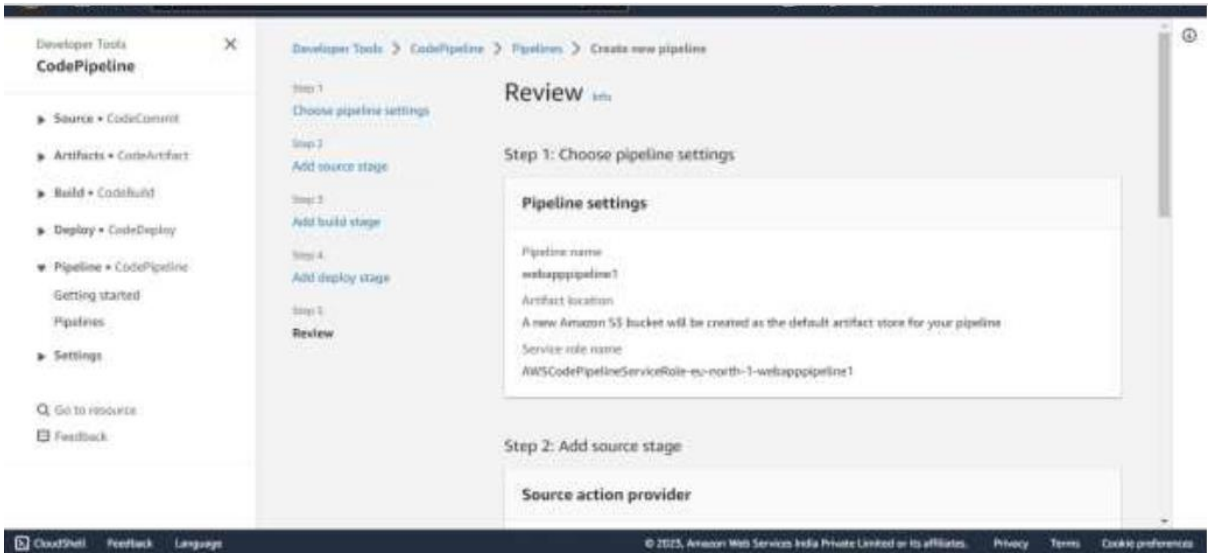

Identity and Access Management (IAM)
Users added to the group
IAM > User groups > t13shreya
t13shreya
Delete
Summary
Edit
User group name
t13shreya
Creation time
August 08, 2023, 15:17 (UTC+05:30)
ARN
Users
Permissions
Access Advisor
Users in this group (1)
Remove users
Add users
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers

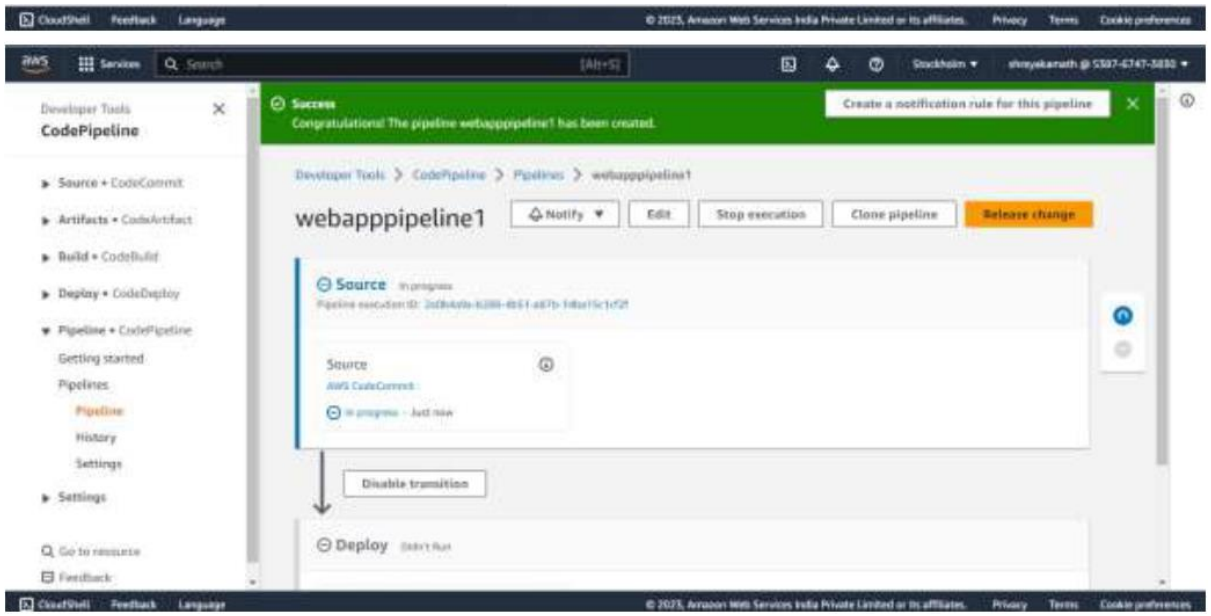

Identity and Access Management (IAM)
IAM > Users
Users (1) info
Delete
Add users
User name
Groups
Last activity
MFA
Password a...
shreyakamath
t13shreya
Never
None
None
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Identity and Access Management (IAM)
IAM > Users > shreyakamath
shreyakamath info
Delete
Summary
ARN
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 (1)
Tags
Security credentials
Access Advisor
Permissions policies (0)
Add permissions
Filter by Type
All types
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Console Home
Reset to default layout
+ Add widgets
Recently visited
Cloud9
EC2
View all services
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.
AWS Health
Cost and usage
CloudShell
Feedback
Language
Privacy
Terms
Cookie preferences
Type here to search
29°C Mostly cloudy


Developer Tools
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
You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit | Learn how to add team members to an AWS CodeStar Project
Step 2: Git credentials
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. Generate Git Credentials
Step 3: Clone the repository
Clone your repository to your local computer and start working on code. Run the following command:
git clone https://git-codecommit.eu-north-1.amazonaws.com/v1/repos/adxdevops
Copy
Additional details
You can find more detailed instructions in the documentation. View documentation
CloudShell
Feedback
Language
Privacy
Terms
Cookie preferences





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.

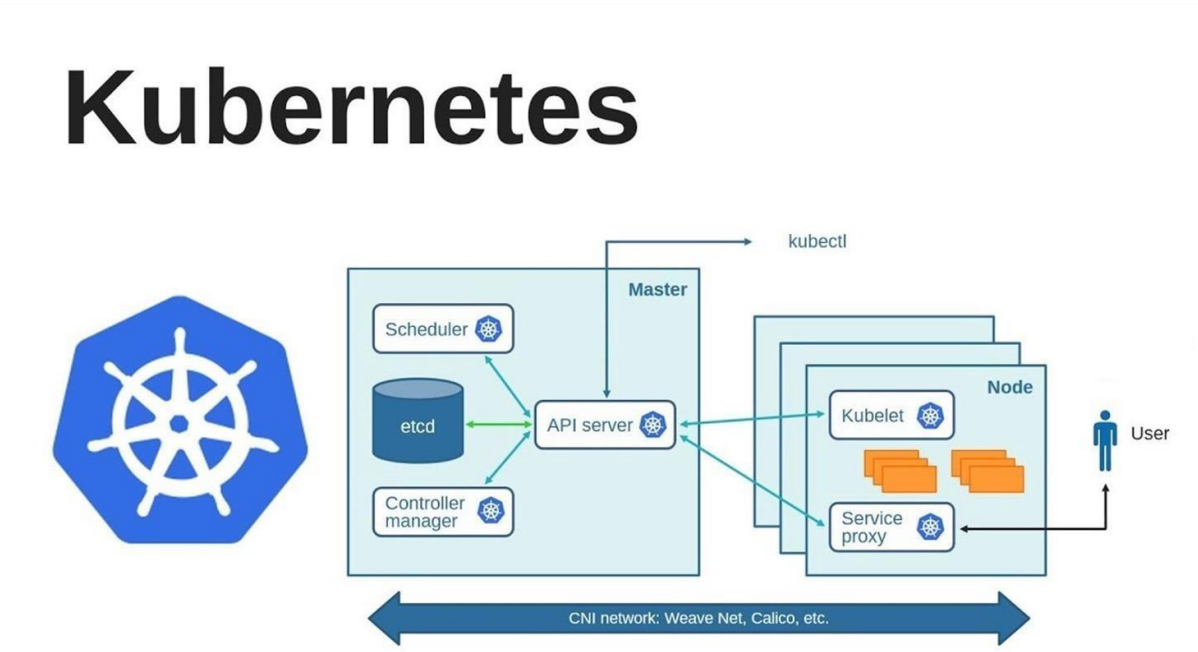
LO2. To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes.

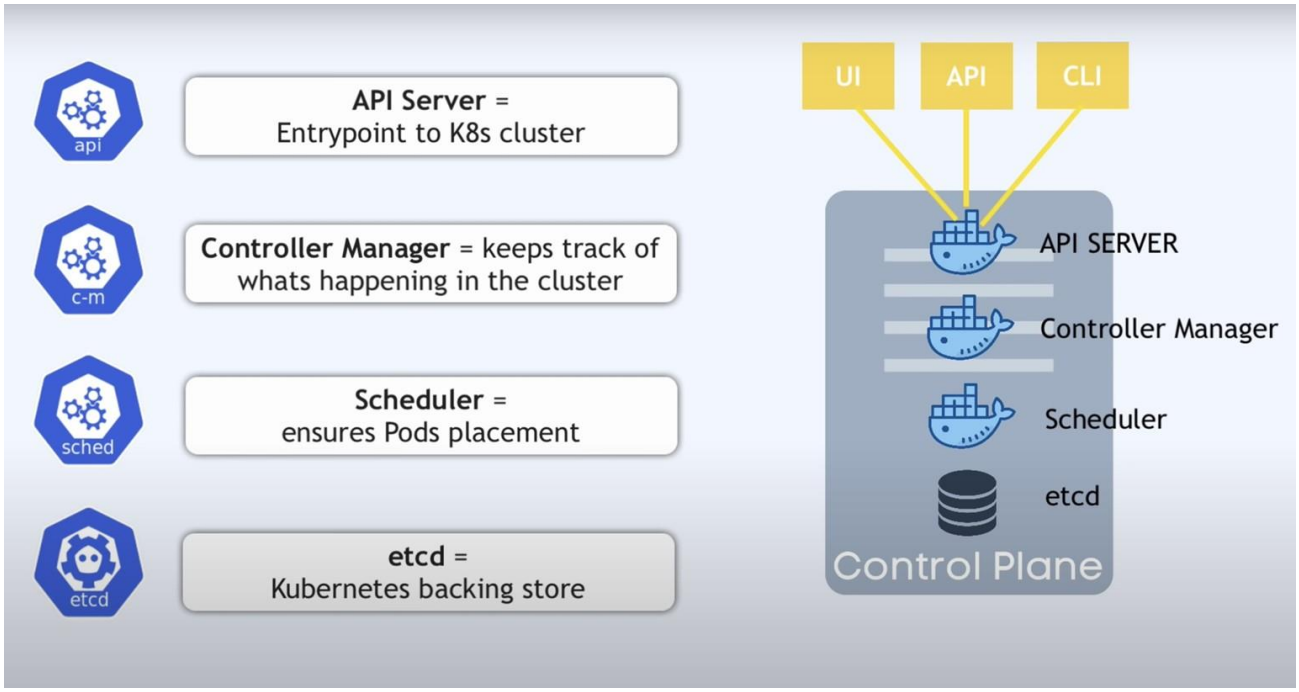
THEORY:

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





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 touch 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-schedular (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

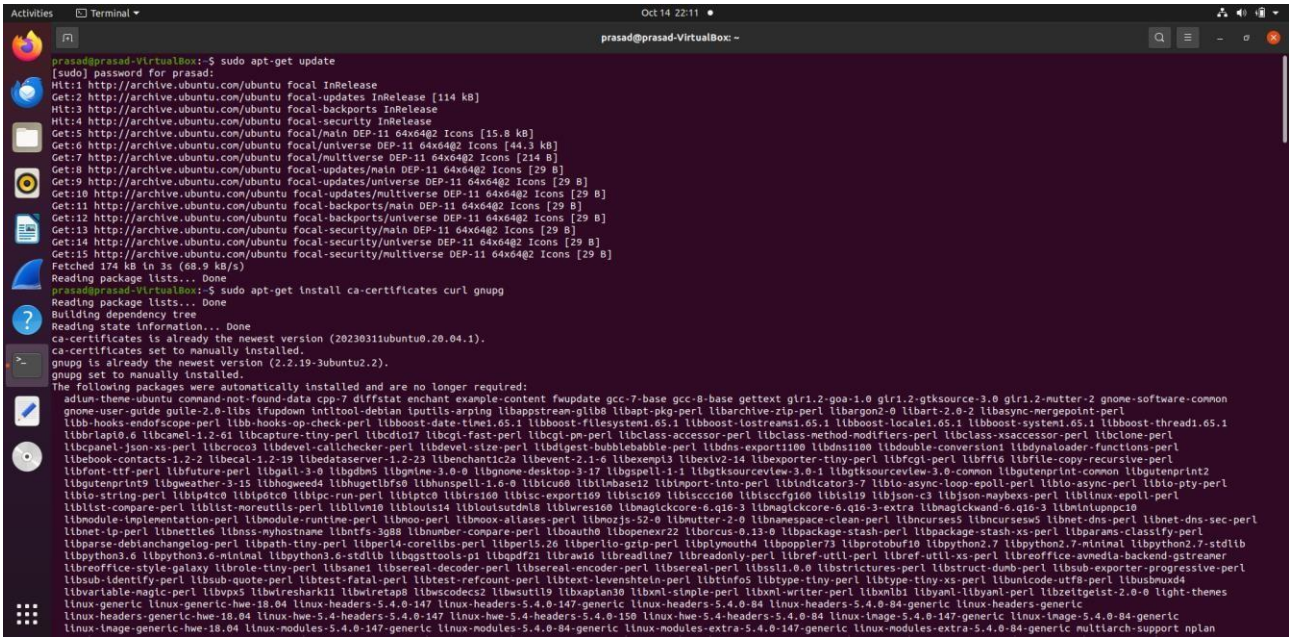
- Works with kubelet
- Pulling images
- Start/Stop Containers
- Exposing containers on ports specified in the manifest.

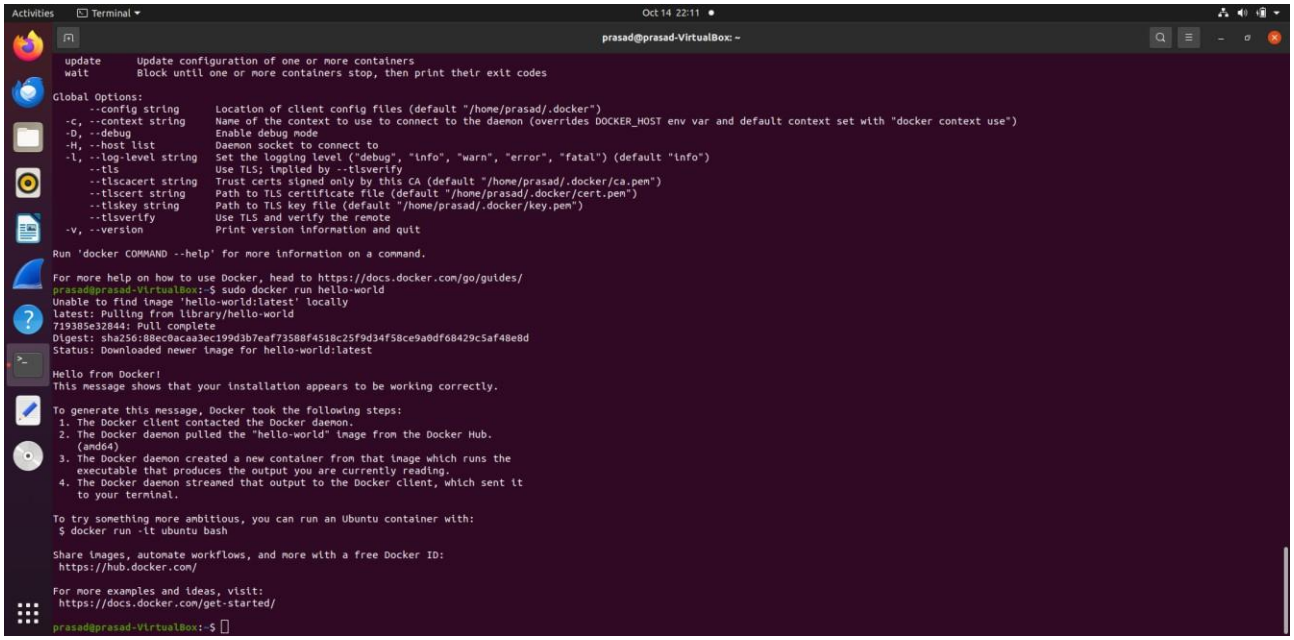
Kube-Proxy

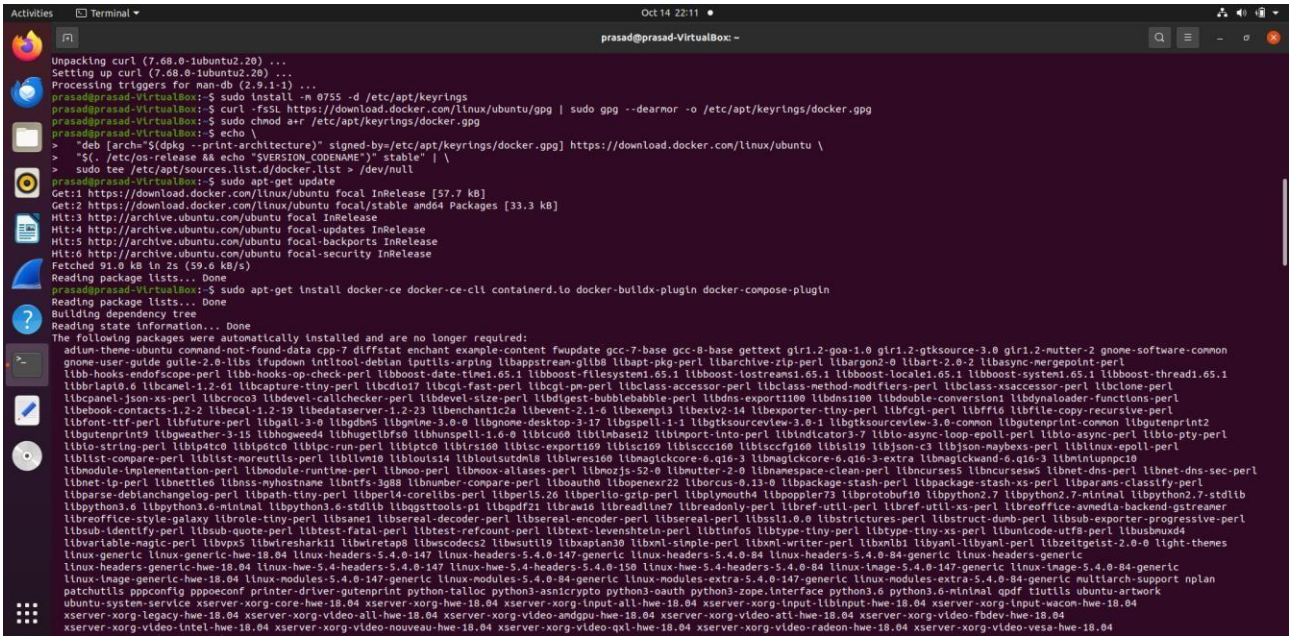
- Assign IP to each pod.
- It is required to assign IP addresses to Pods (dynamic)
- Kube-proxy runs on each node & this makes sure that each pod will get its unique IP Address.
- These 3 components collectively consist of ‘node’.

INSTALLATION:

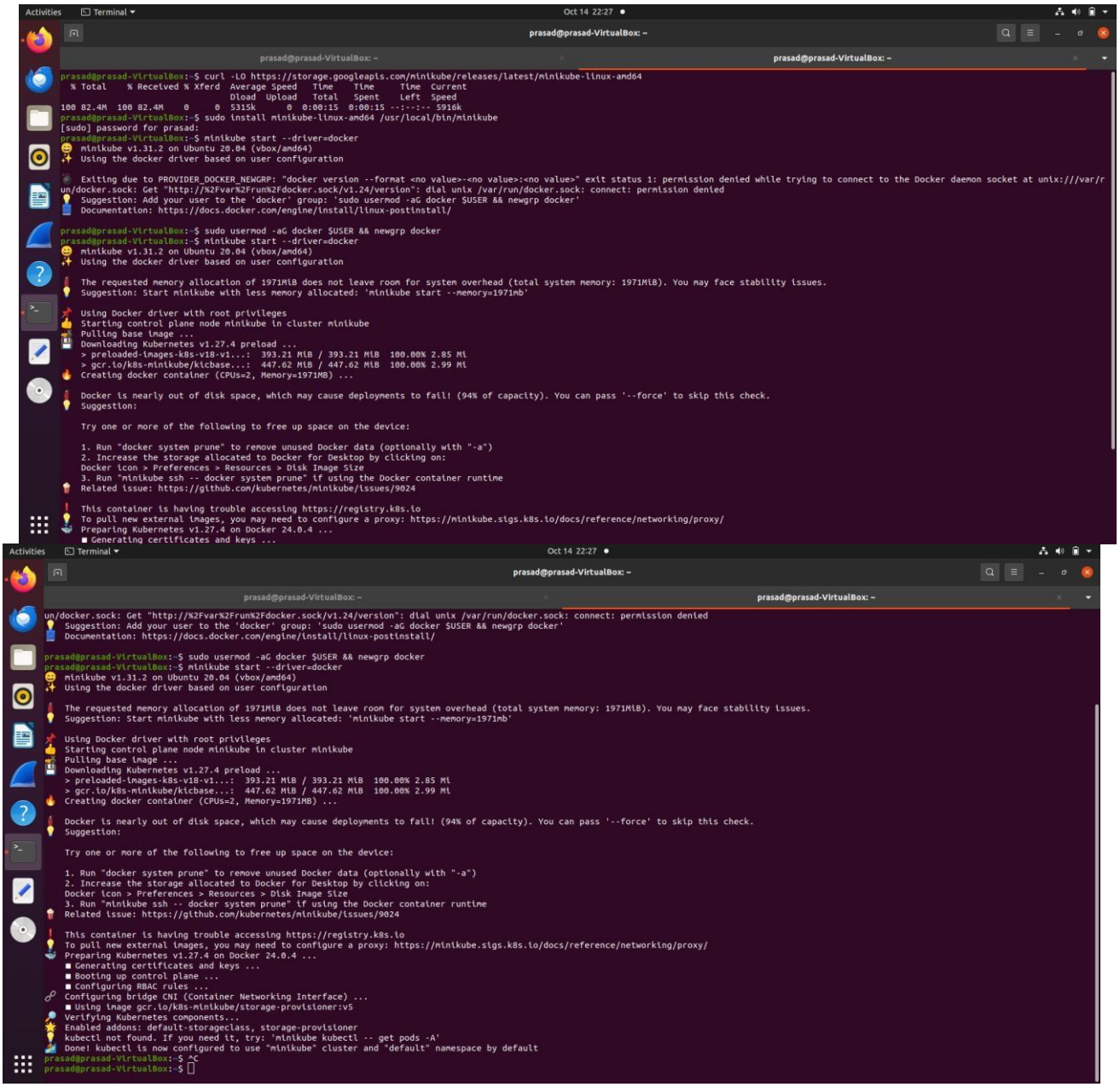
1. Install Docker



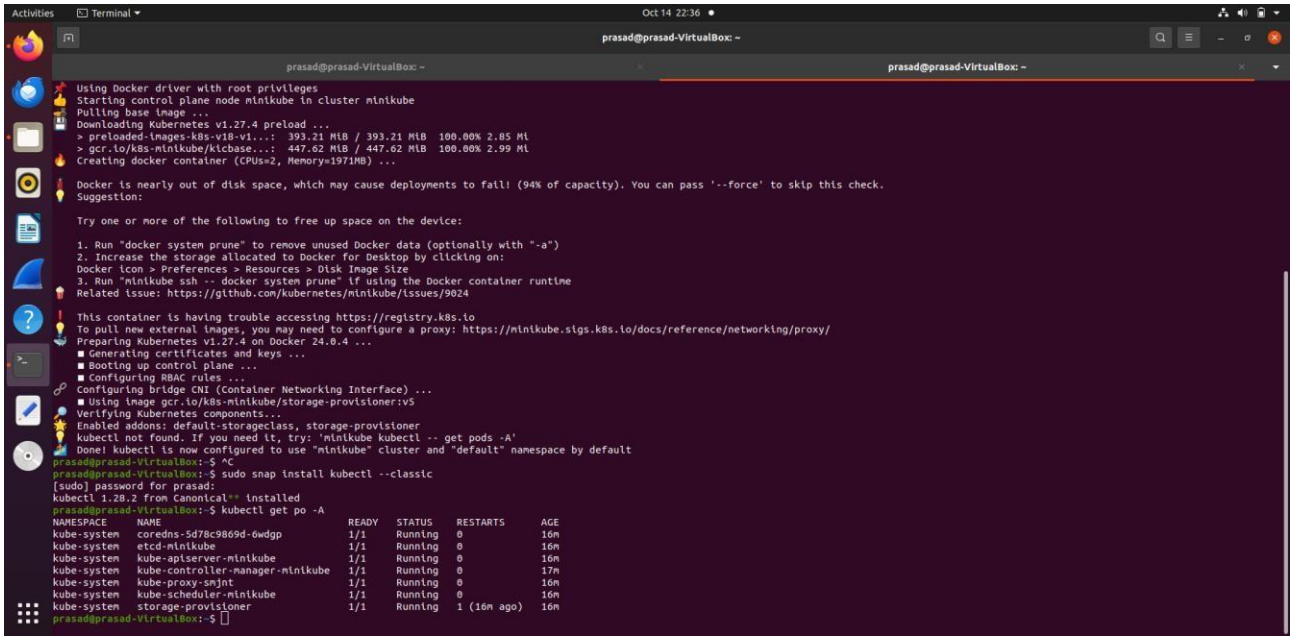




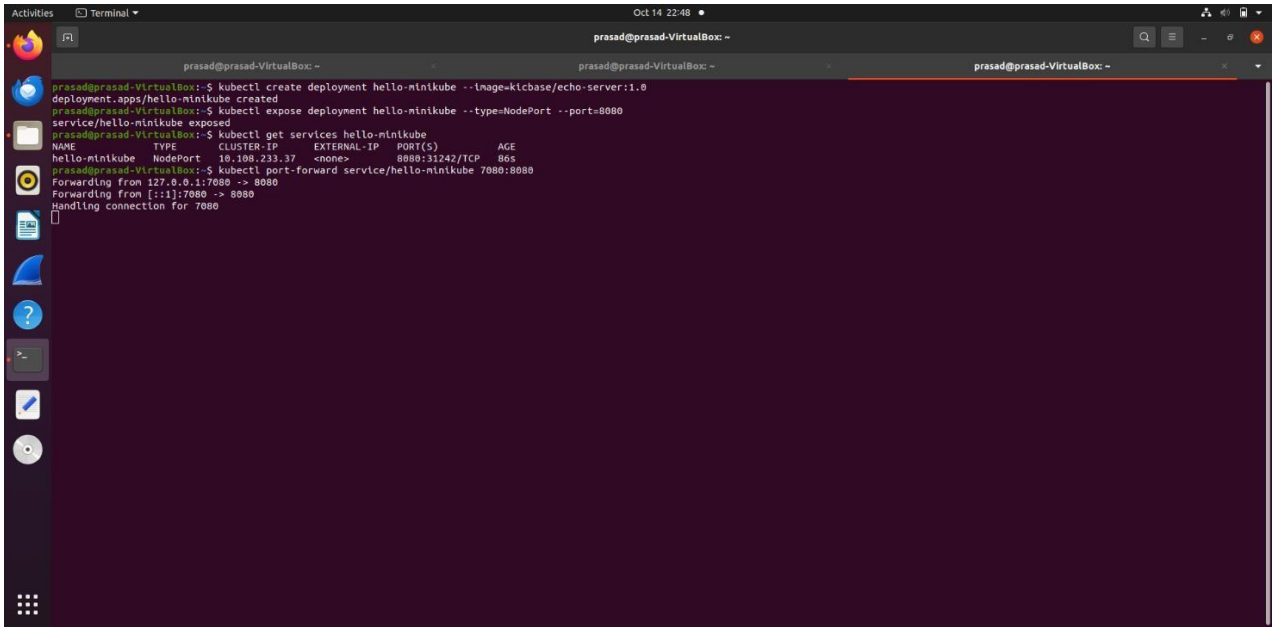
2. Install minikube using following commands

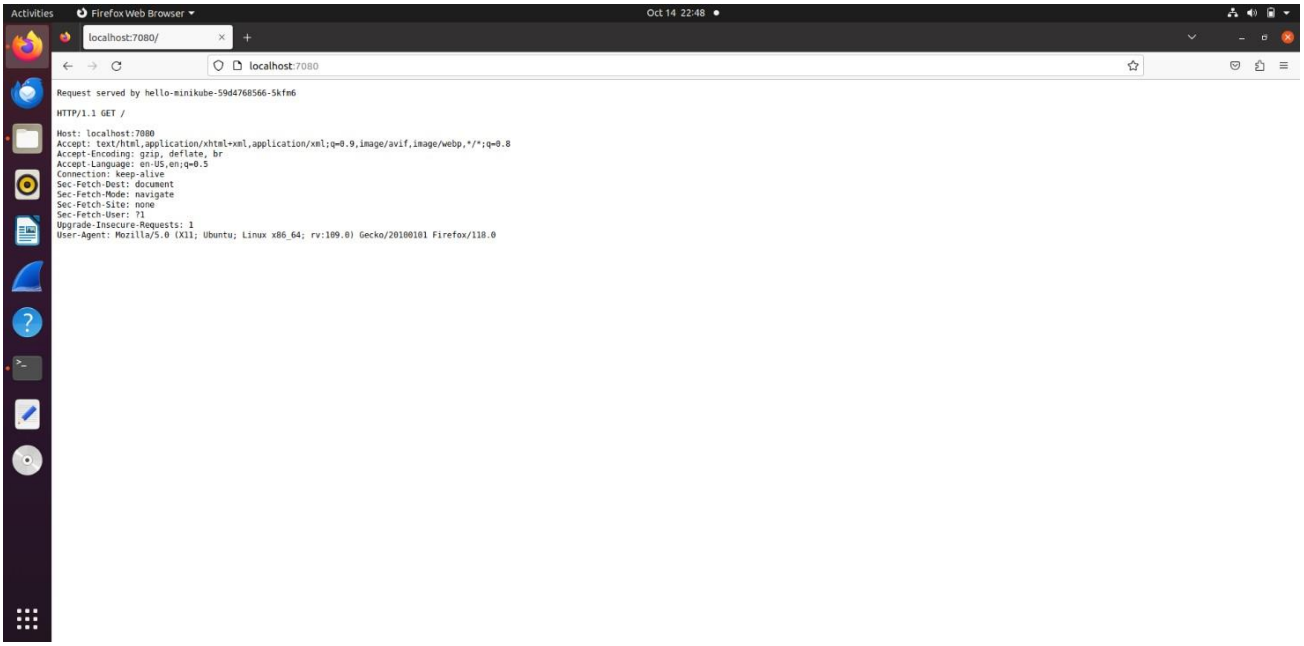


3. Install kubectl



4. Create a sample deployment.





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:

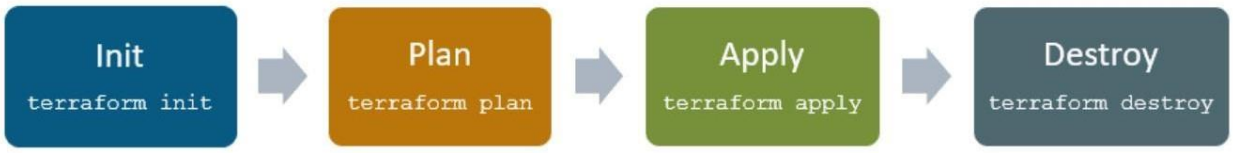
Terraform is one of the most popular Infrastructure-as-code (IaC) tool, used by DevOps teams to automate infrastructure tasks. It is used to automate the provisioning of your cloud resources. Terraform is an open-source, cloud-agnostic provisioning tool developed by HashiCorp and written in GO language.

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.

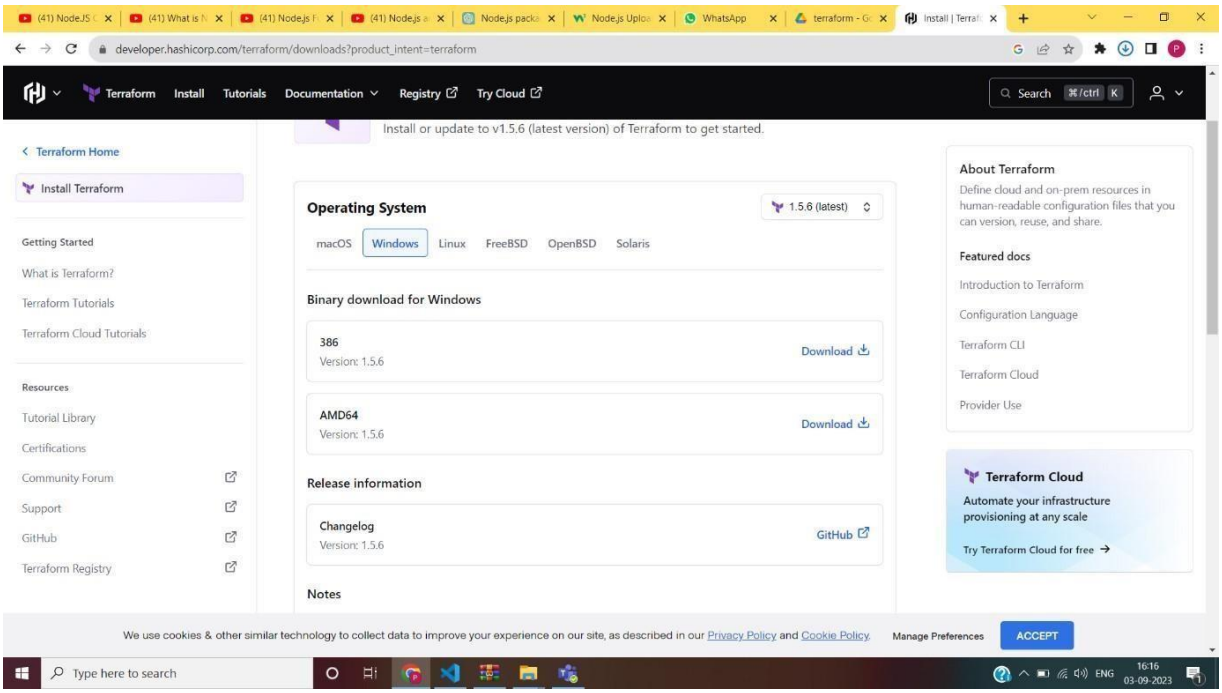
INSTALLATION:

1) Download Terraform

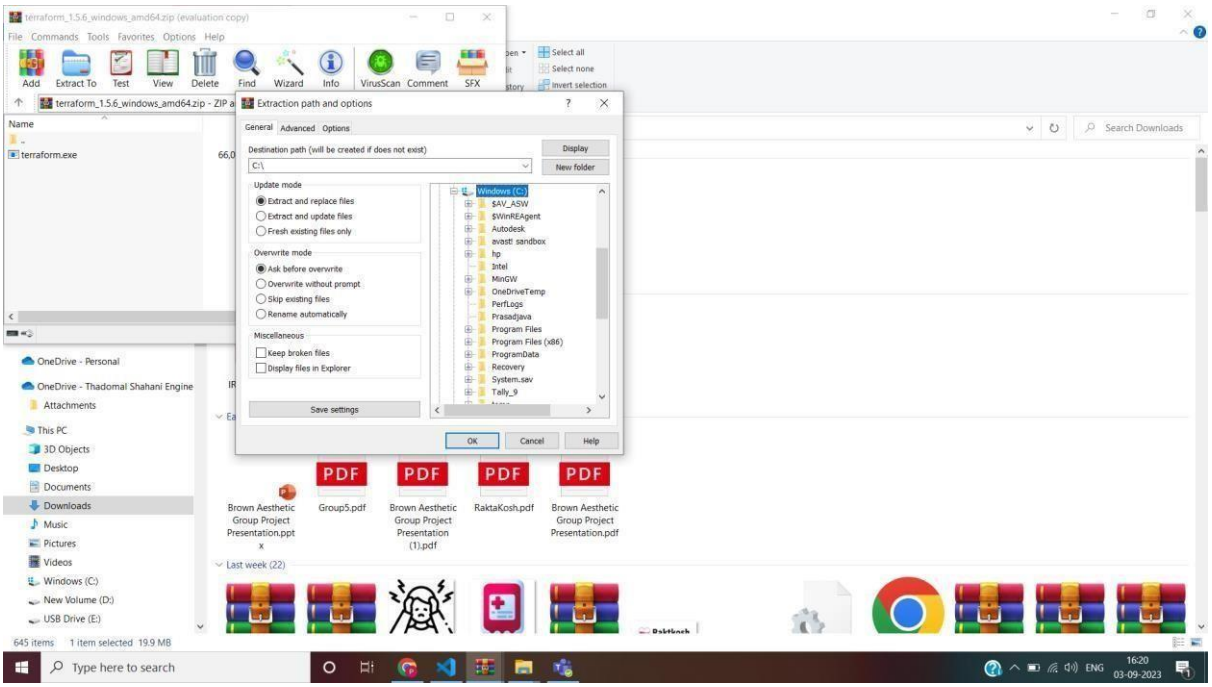
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

website: https://www.terraform.io/downloads.html

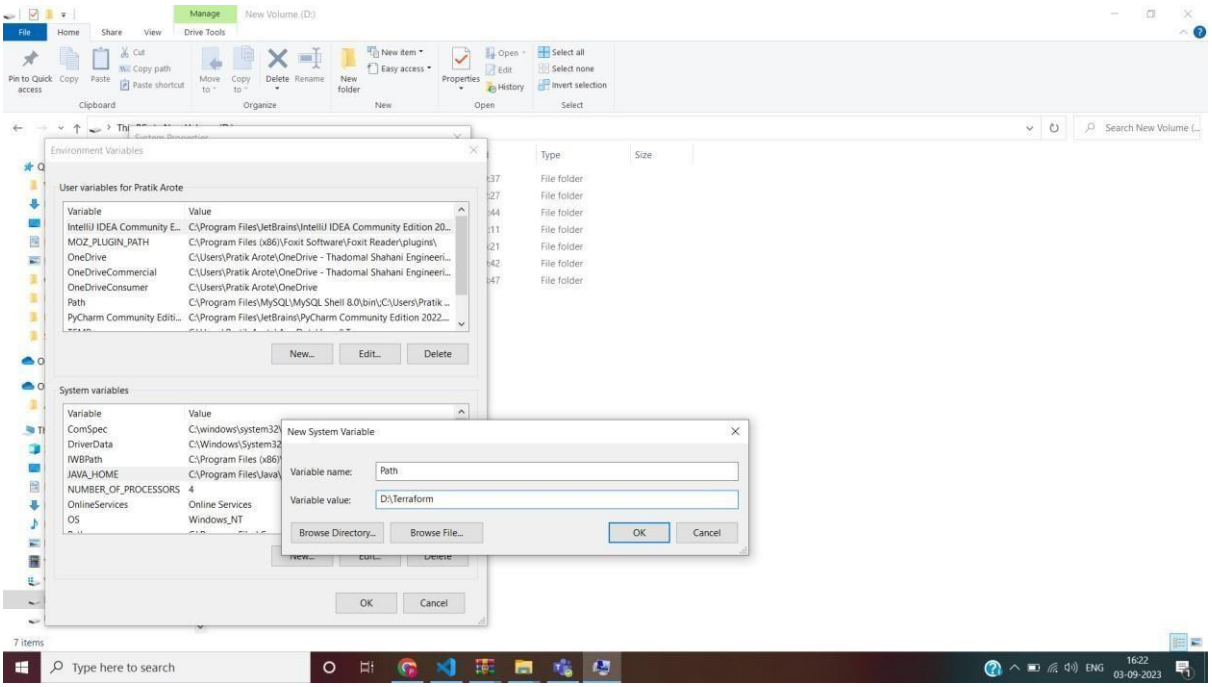
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.



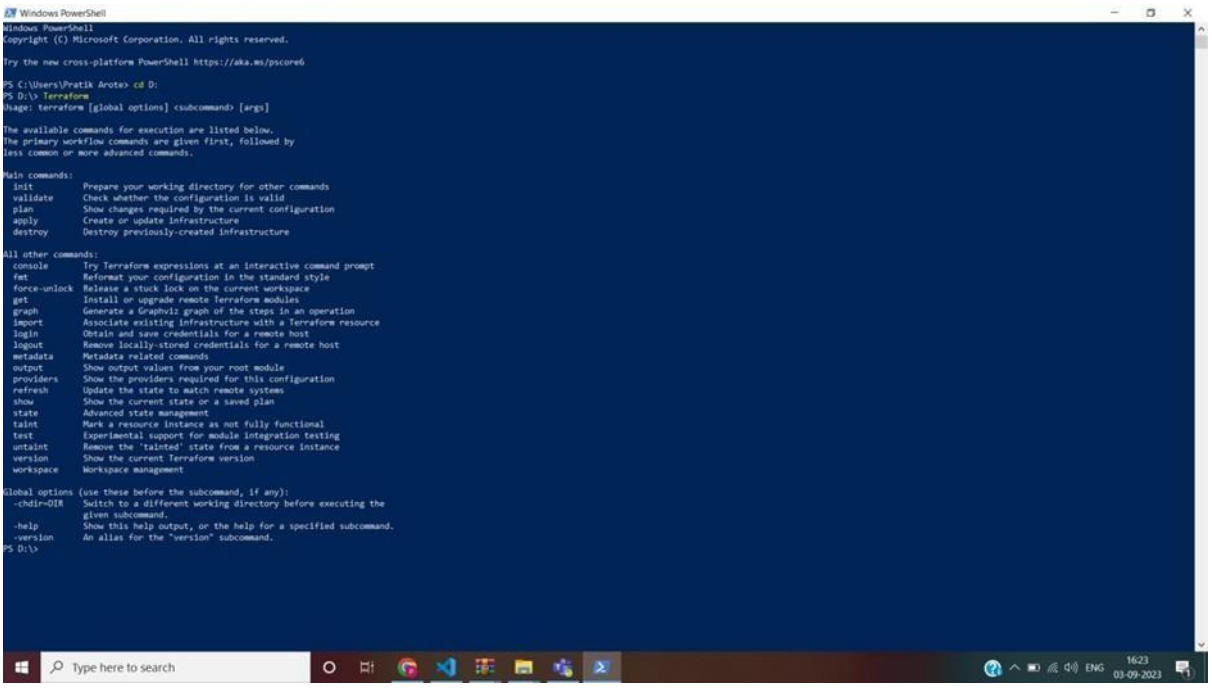
2. Extract the downloaded setup file Terraform.exe in C:\Terraform Directory



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



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



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.

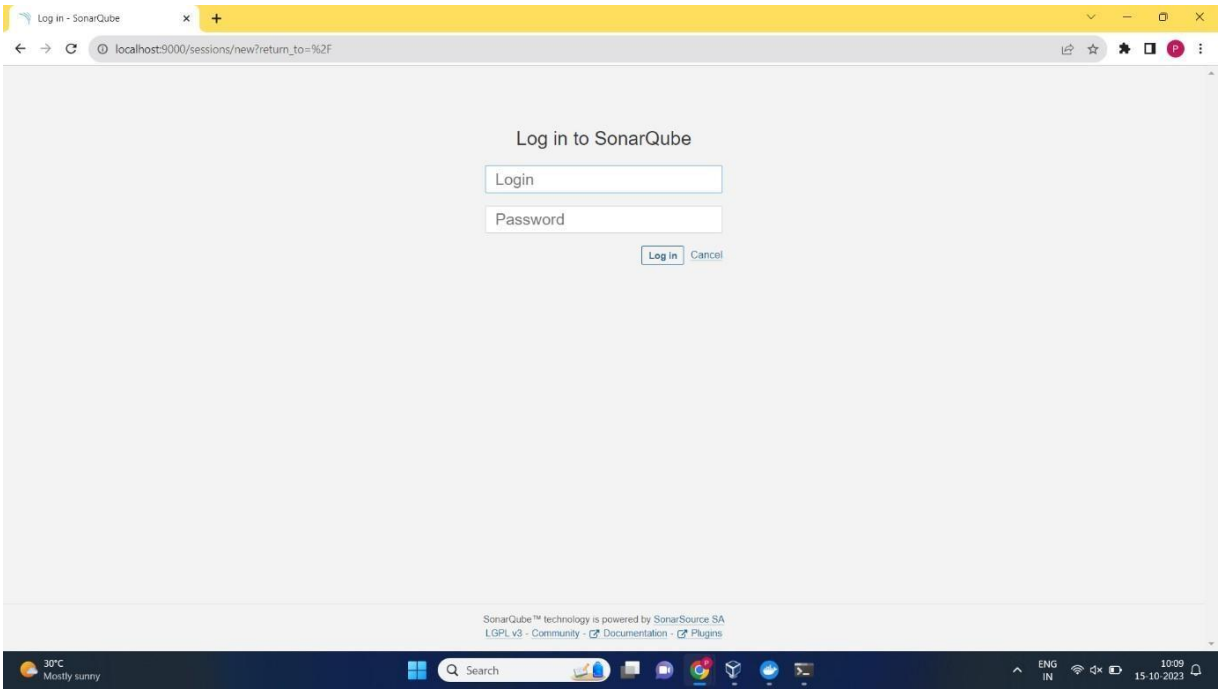
1. INSTALL sonarqube (docker images) and sonarscanner zip file from https://docs.sonarsource.com/sonarqube/latest/analyzing-sourcecode/scanners/sonarscanner/ and set up config file as given in docs.

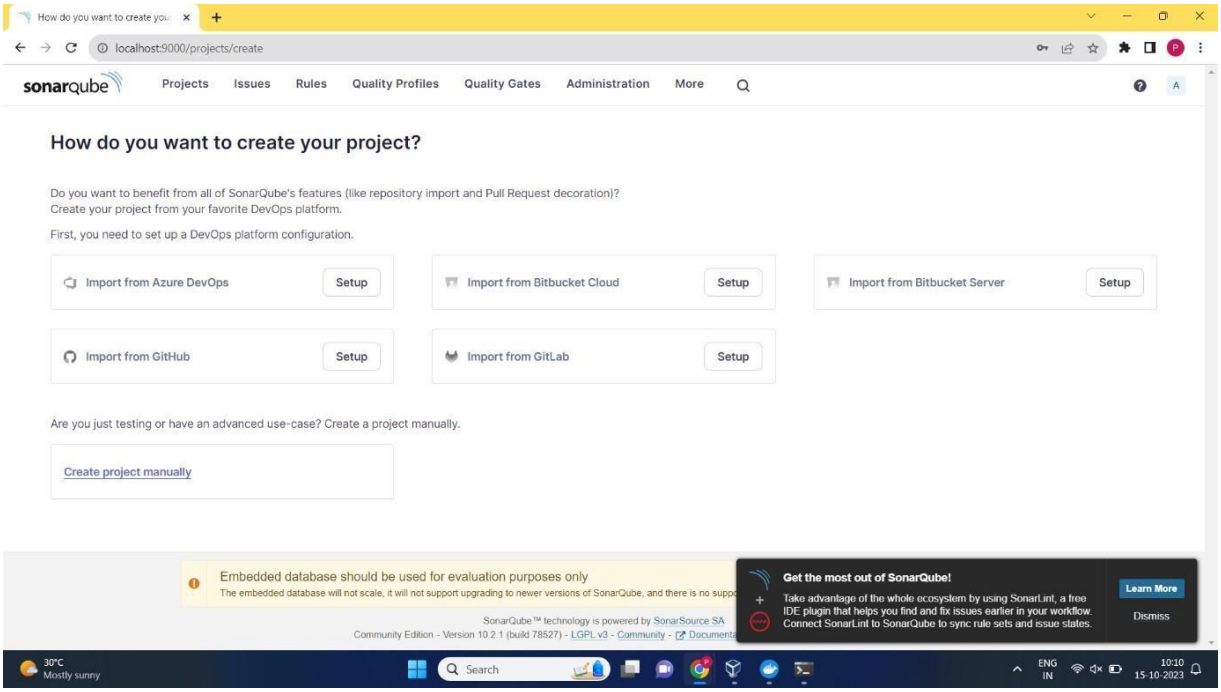
```
Command Prompt
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

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

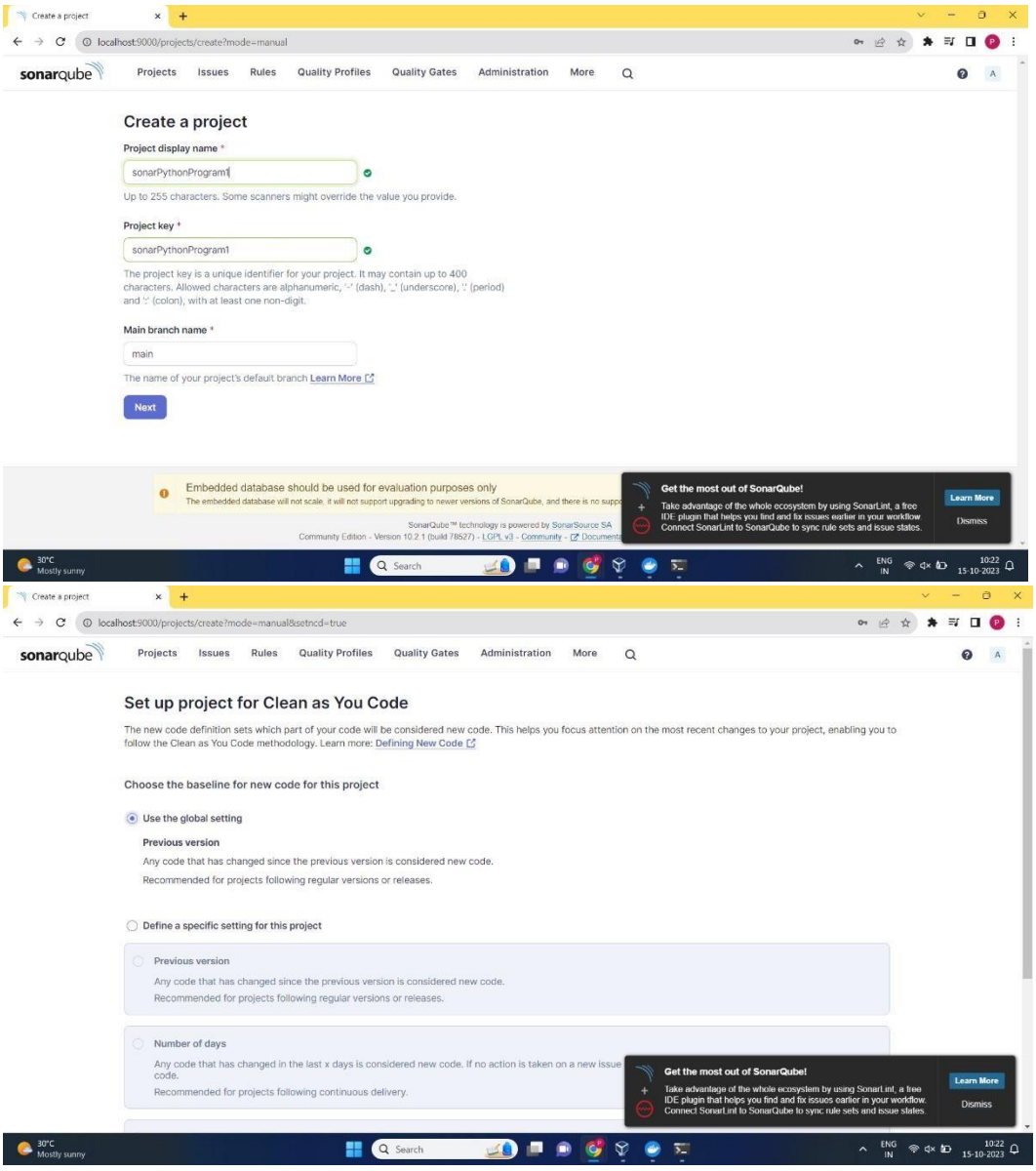
```
C:\Users\Pratik Arote>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
f3630dbc2ffa6e5598ad922085026400a1f9f1564416b0606b5348000f6d1377

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
```

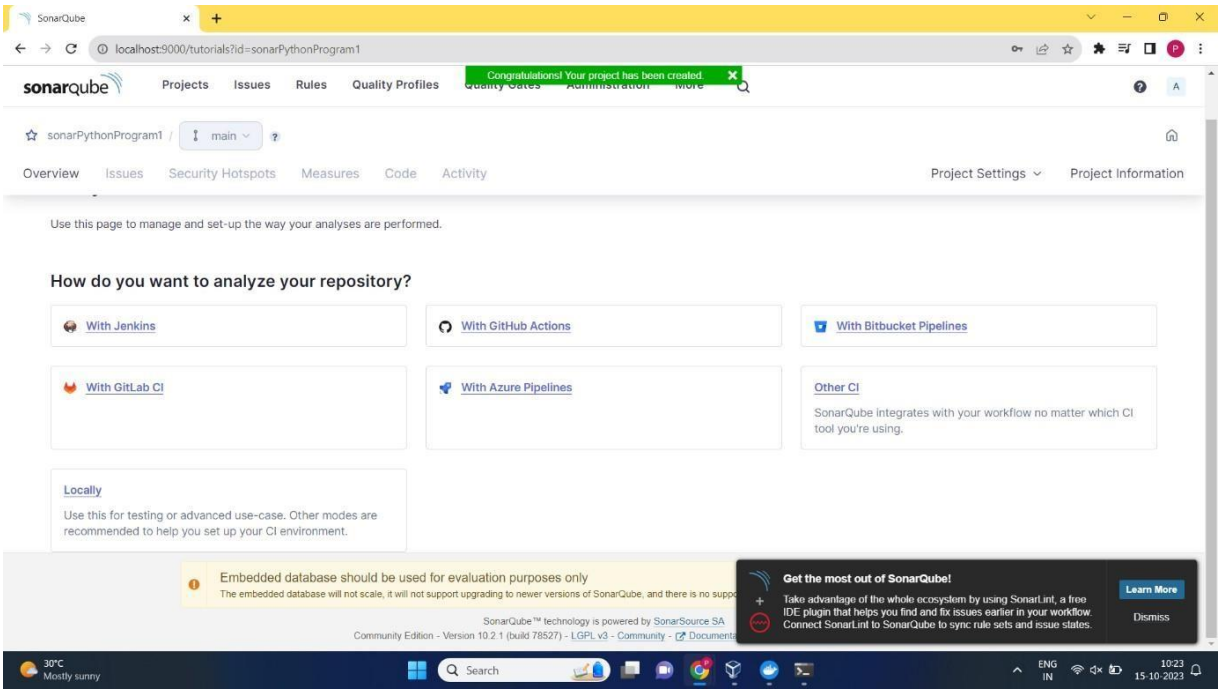
3. Open http://localhost:9000 on the browser. Enter login and password both as “admin” and then set up new password.



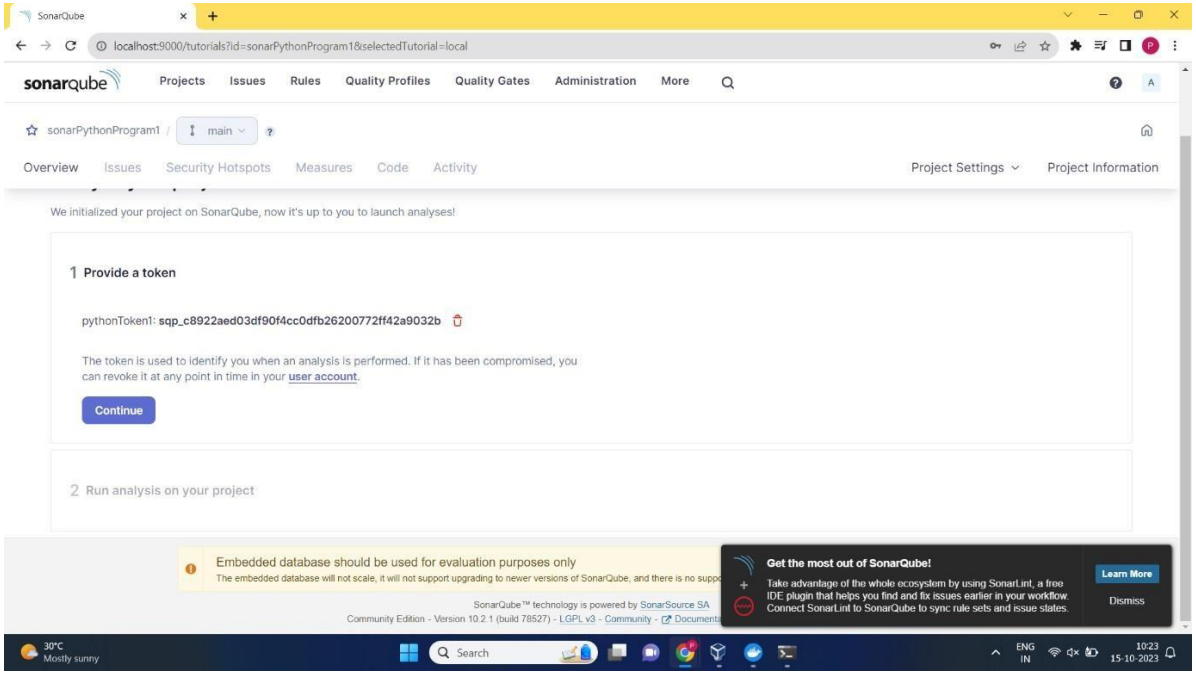


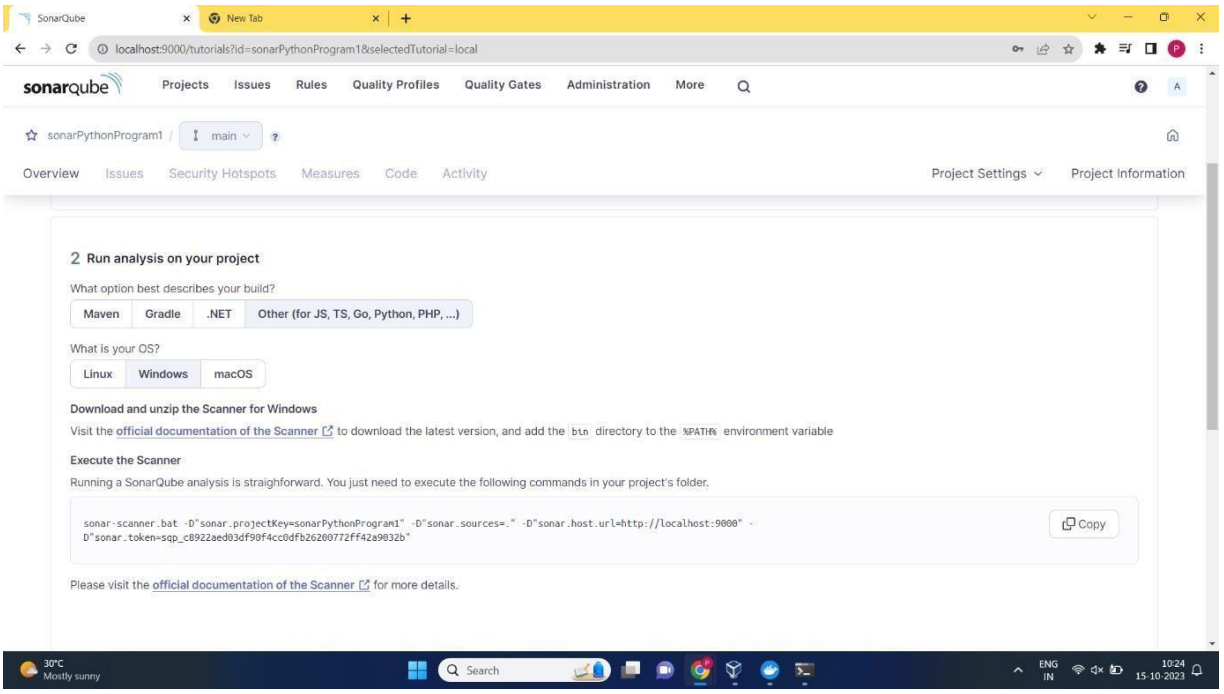
4. Create a project


Create a project
localhost:9000/projects/create?mode=manual
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Create a project
Project display name *
sonarPythonProgram1
Up to 255 characters. Some scanners might override the value you provide.
Project key *
sonarPythonProgram1
The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.
Main branch name *
main
The name of your project's default branch Learn More
Next
Embedded database should be used for evaluation purposes only
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
Search
Create a project
localhost:9000/projects/create?mode=manual&setncd=true
Set up project for Clean as You Code
The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: Defining New Code
Choose the baseline for new code for this project
Use the global setting
Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
Define a specific setting for this project
Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue
code.
Recommended for projects following continuous delivery.
Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
Learn More
Dismiss
Search

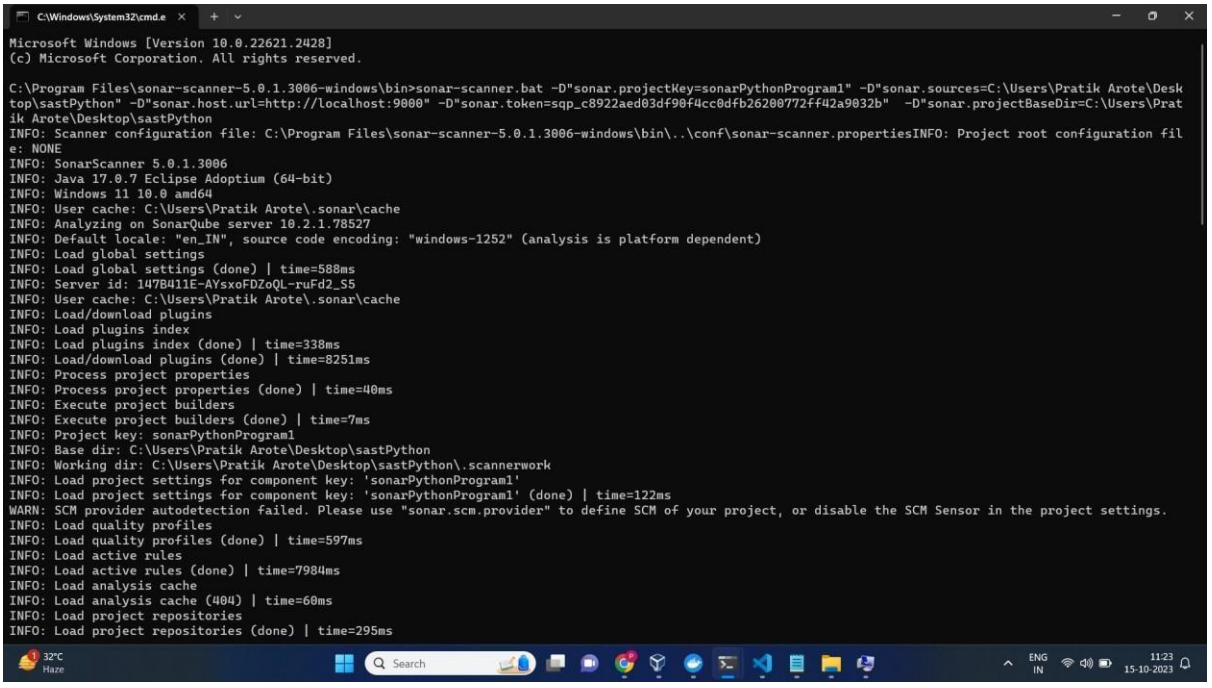


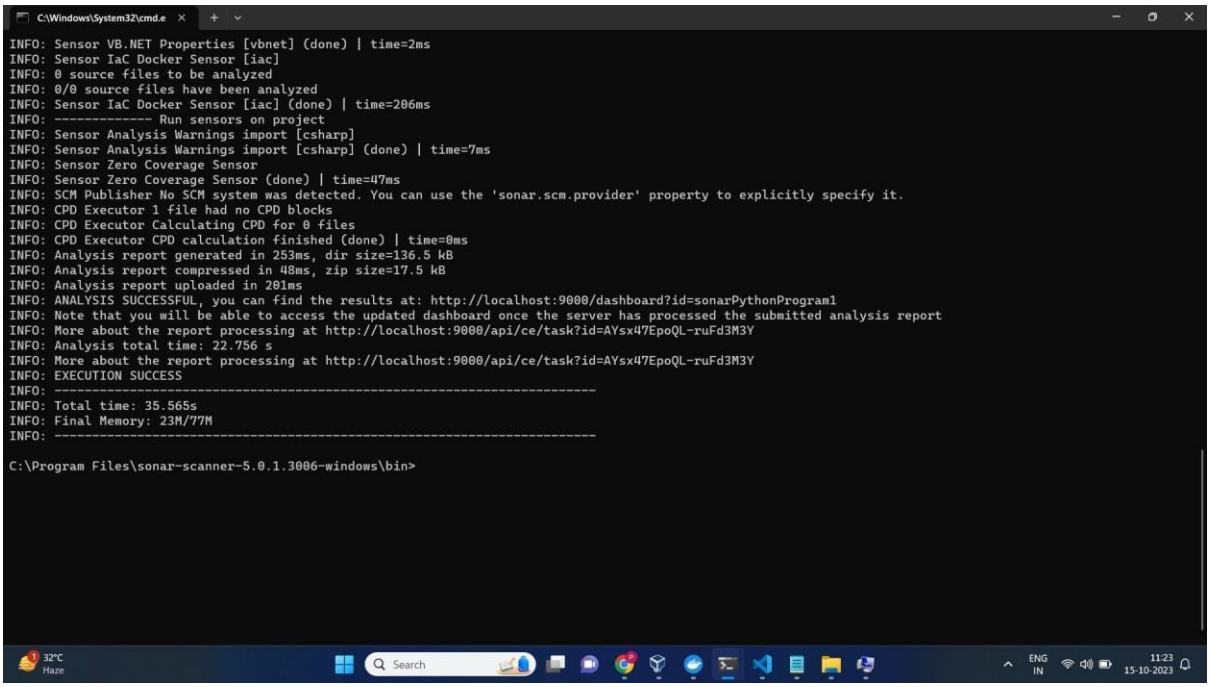
5. Provide token



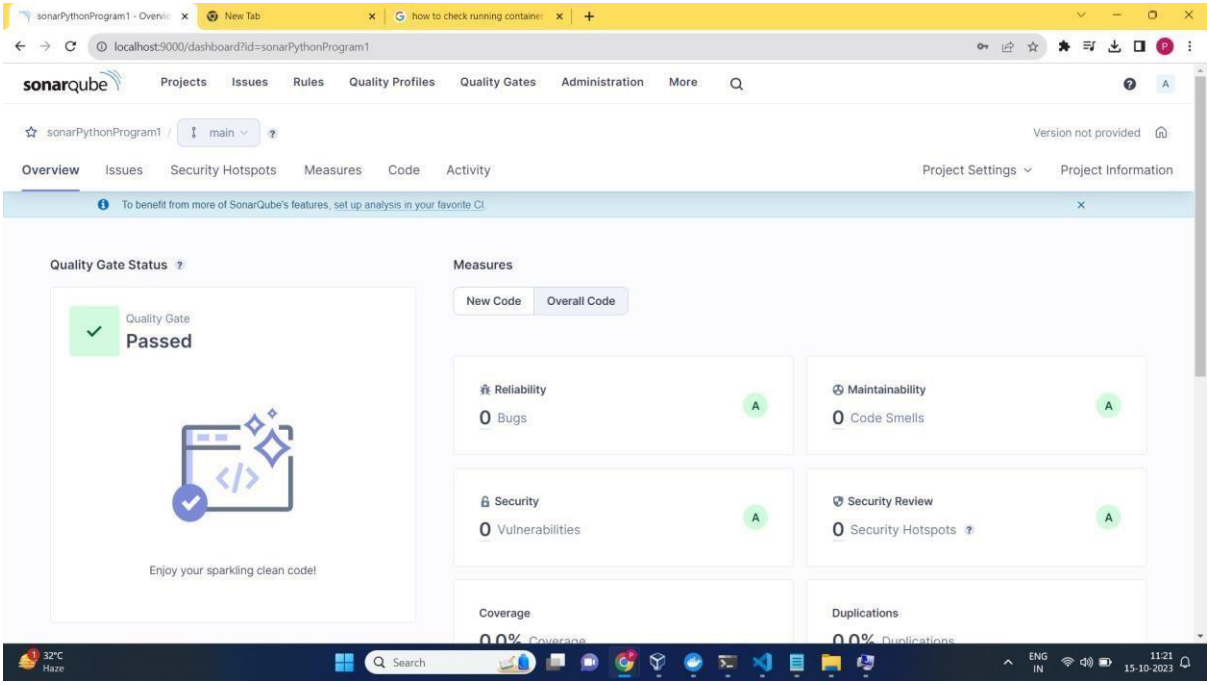


6. Enter the following command





7. See the result of the test



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.

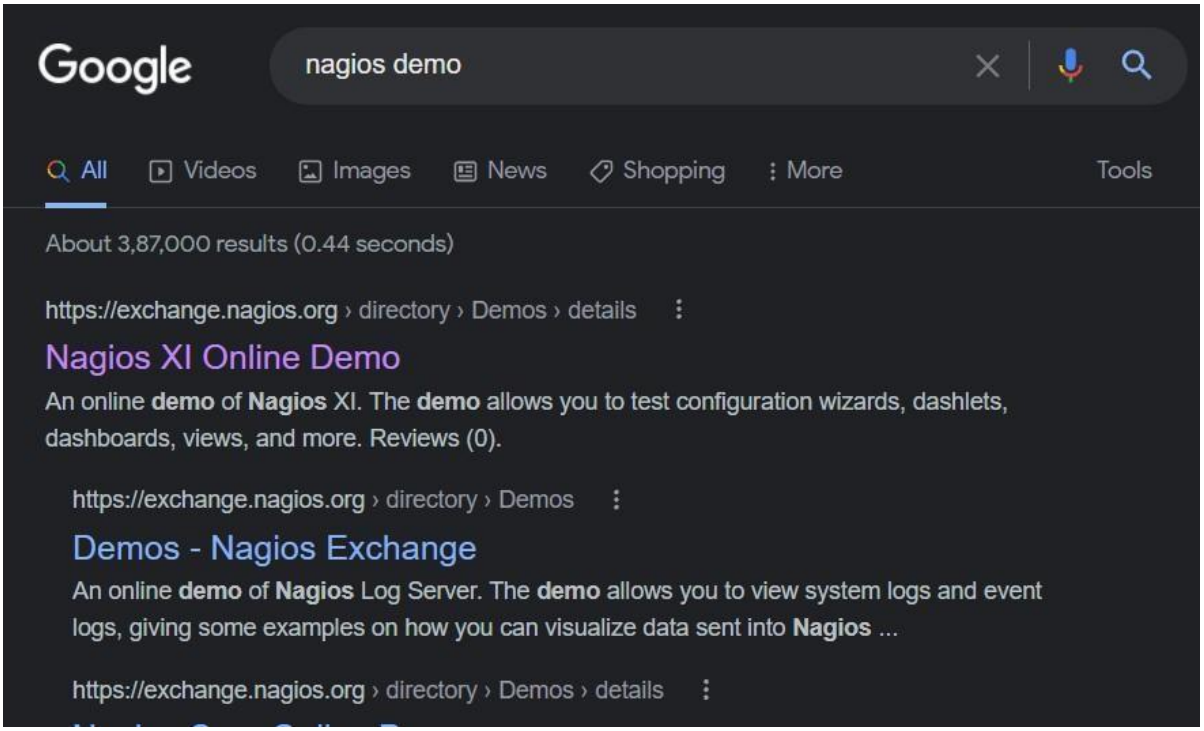
Theory:

What is Nagios and how it works?

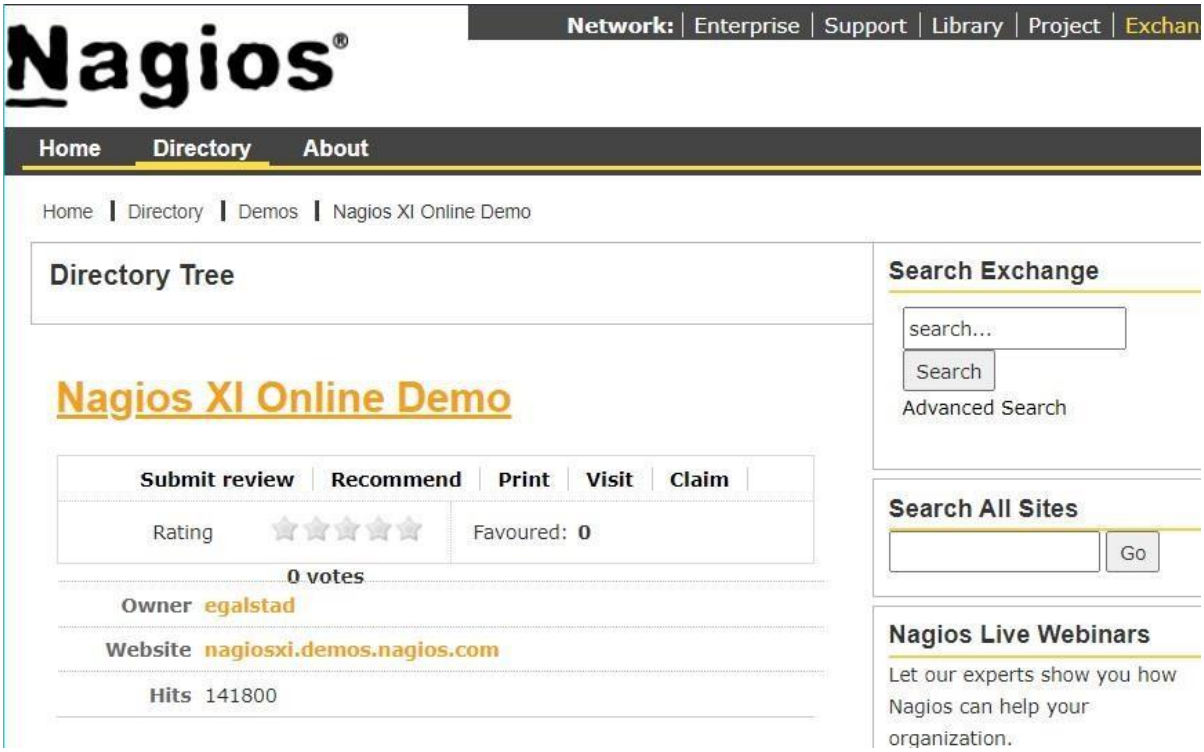
Nagios is an open source monitoring system for computer systems ..... Nagios software runs periodic checks on critical parameters of application, network and server resources. For example, Nagios can monitor memory usage, disk usage, microprocessor load, the number of currently running processes and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

Steps-

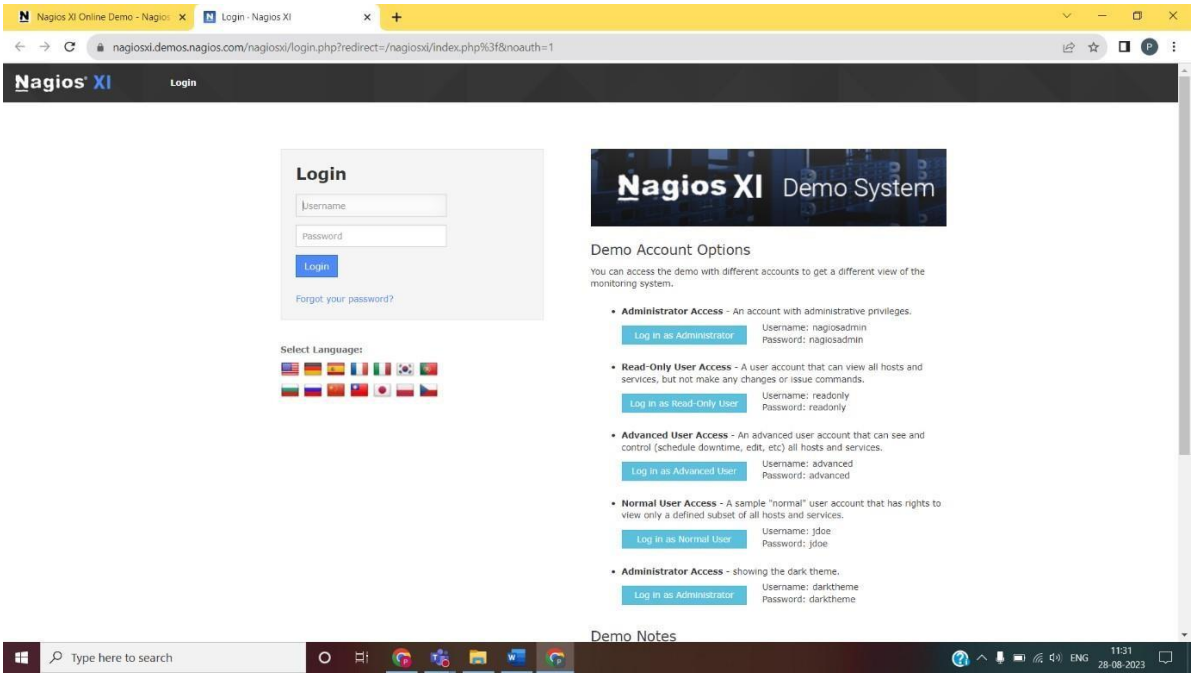
Go to google.com, Search Nagios Demo



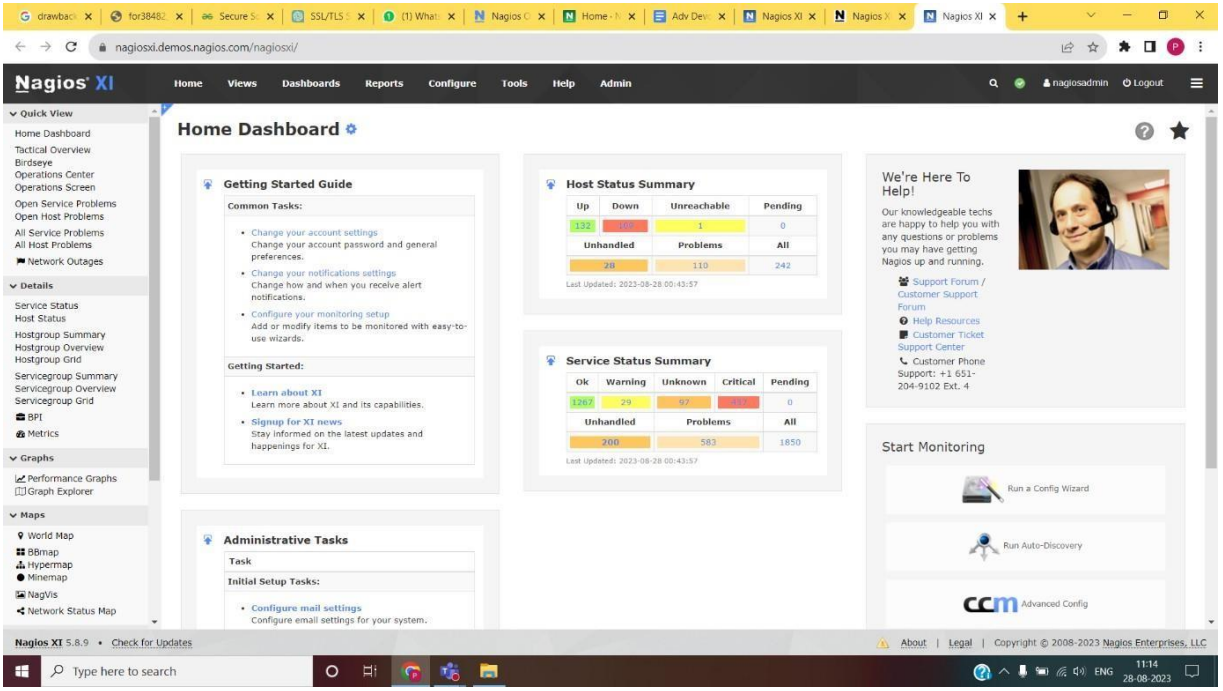
Now click on the website-



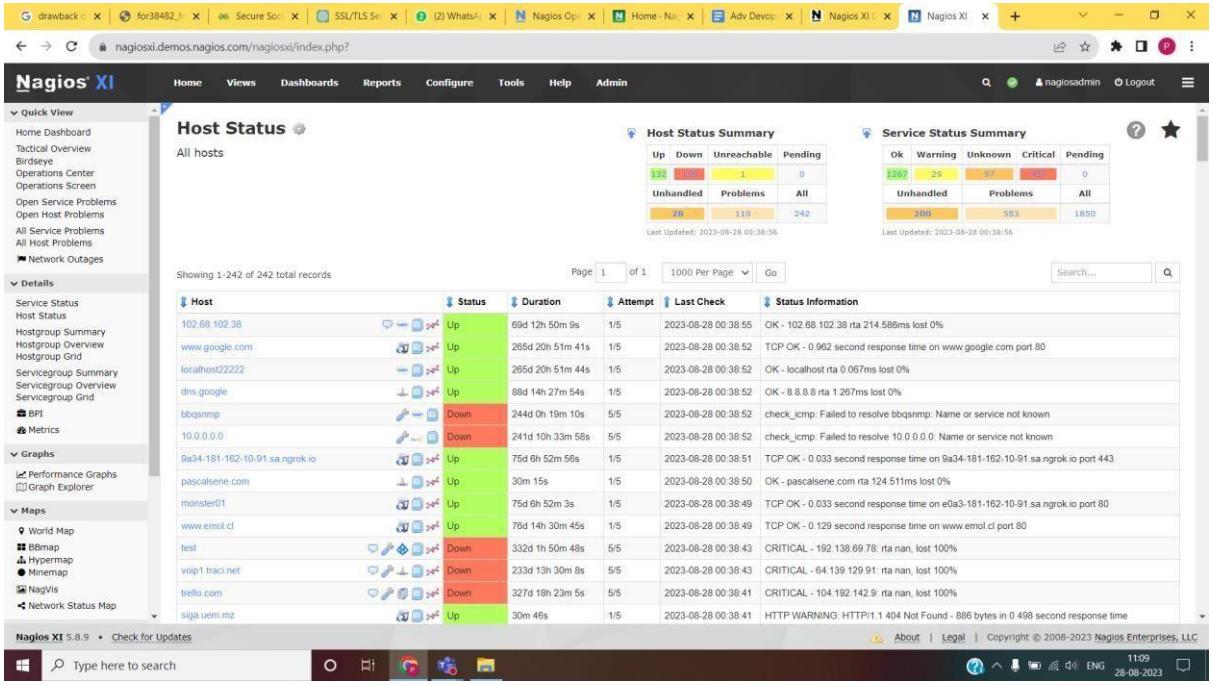
Now click on login as administrator



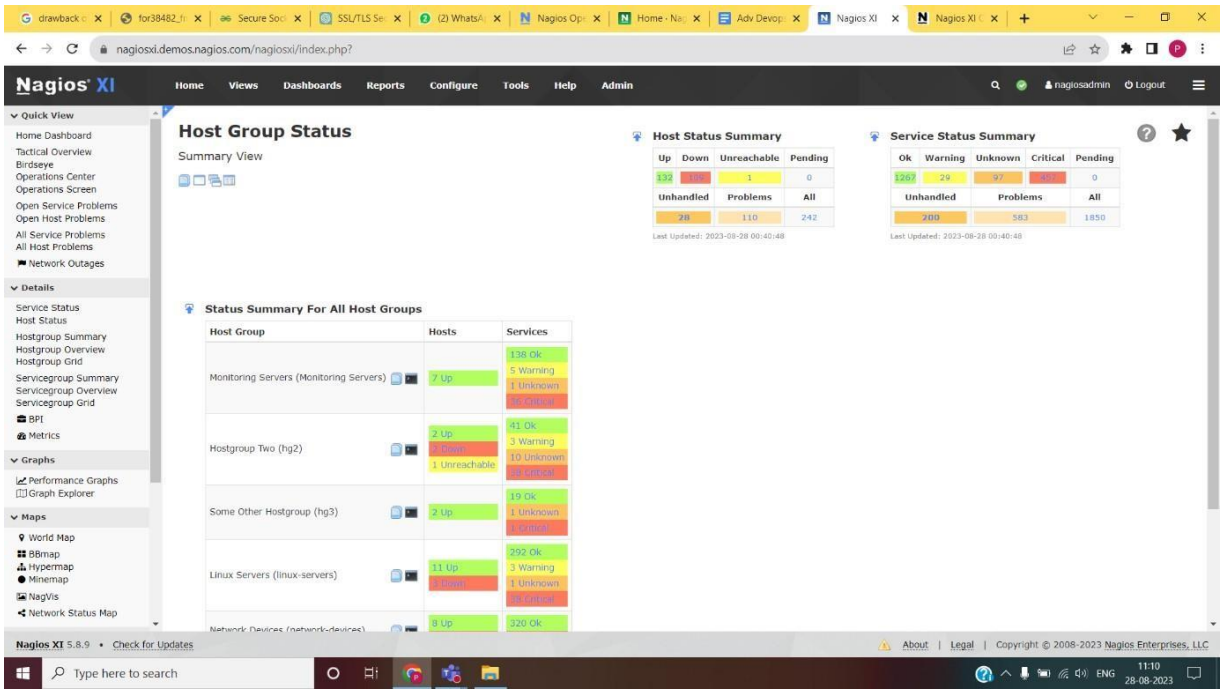
It will have interface like this



Now click on Host status-



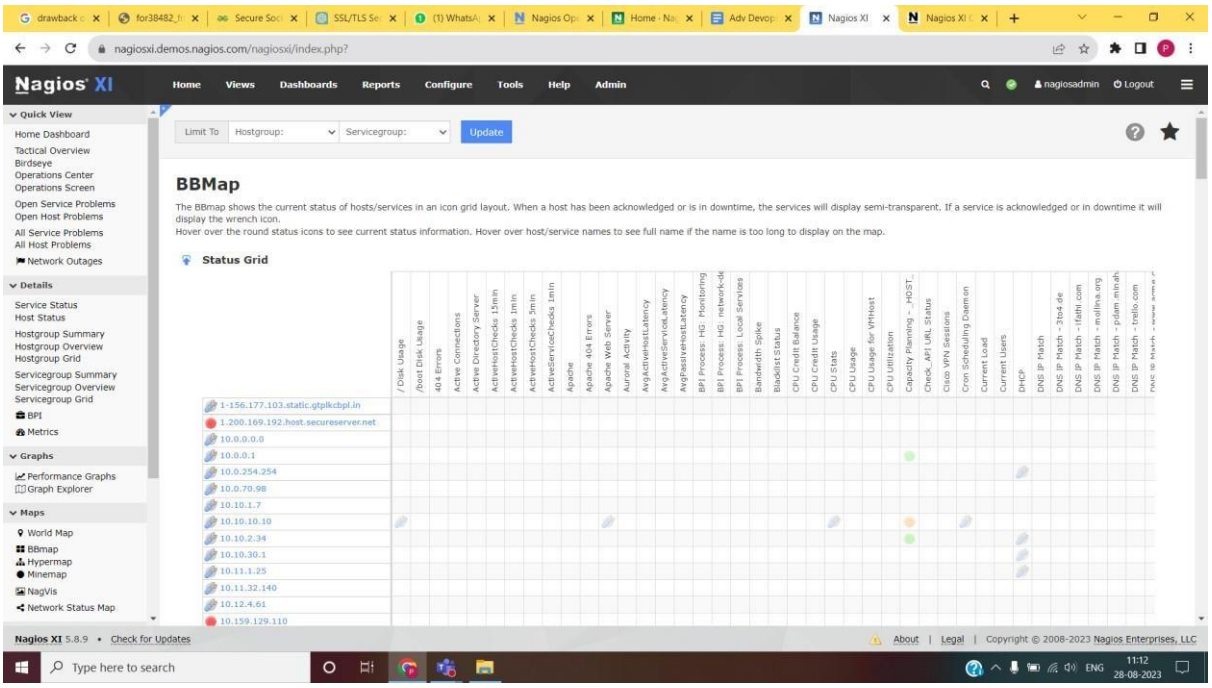
In the above image one can see Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.



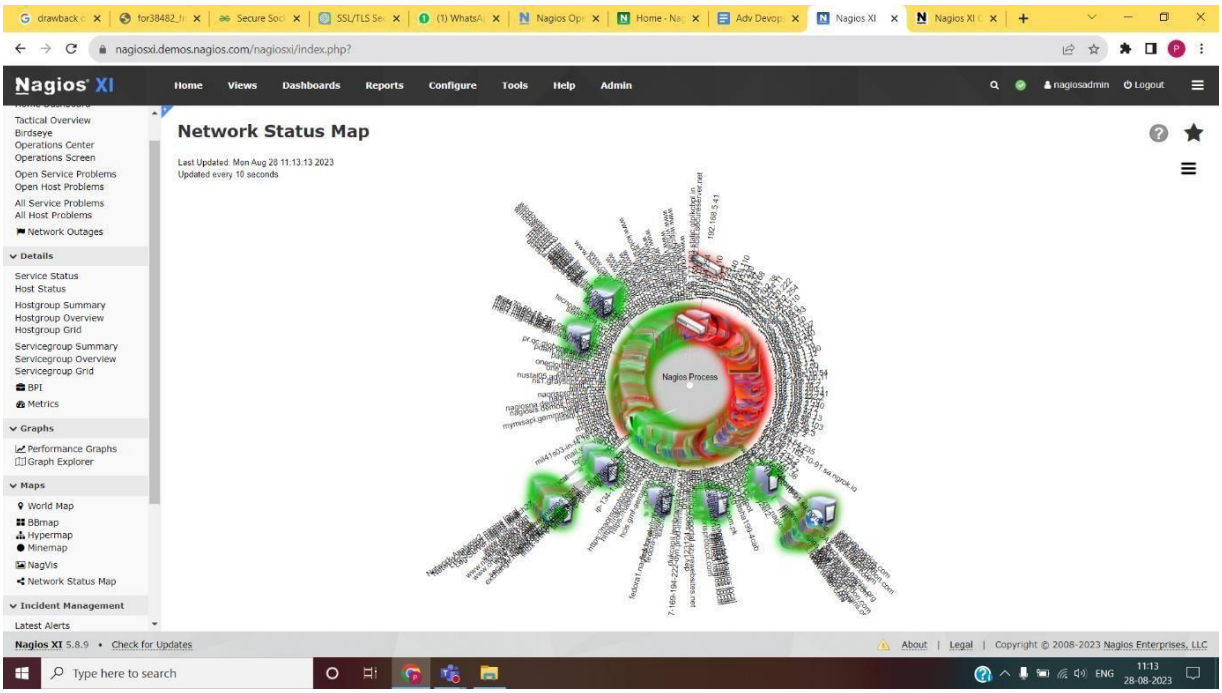
Here we can see Status Summary for All Host Groups

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.

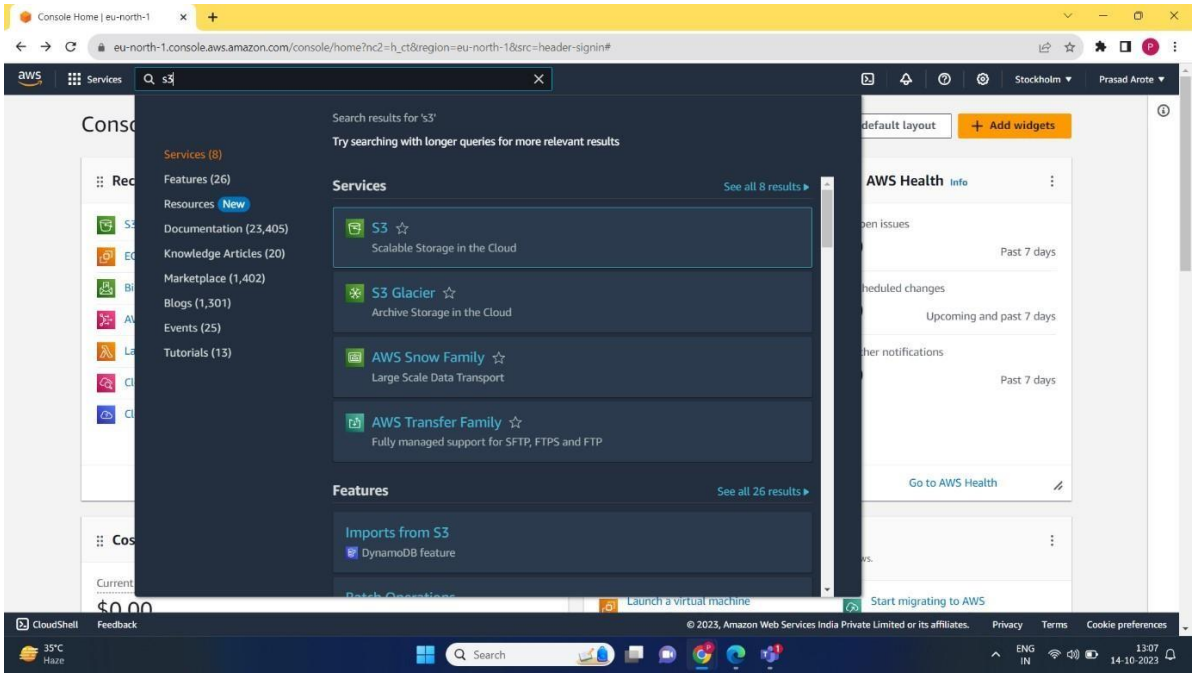
THEORY:

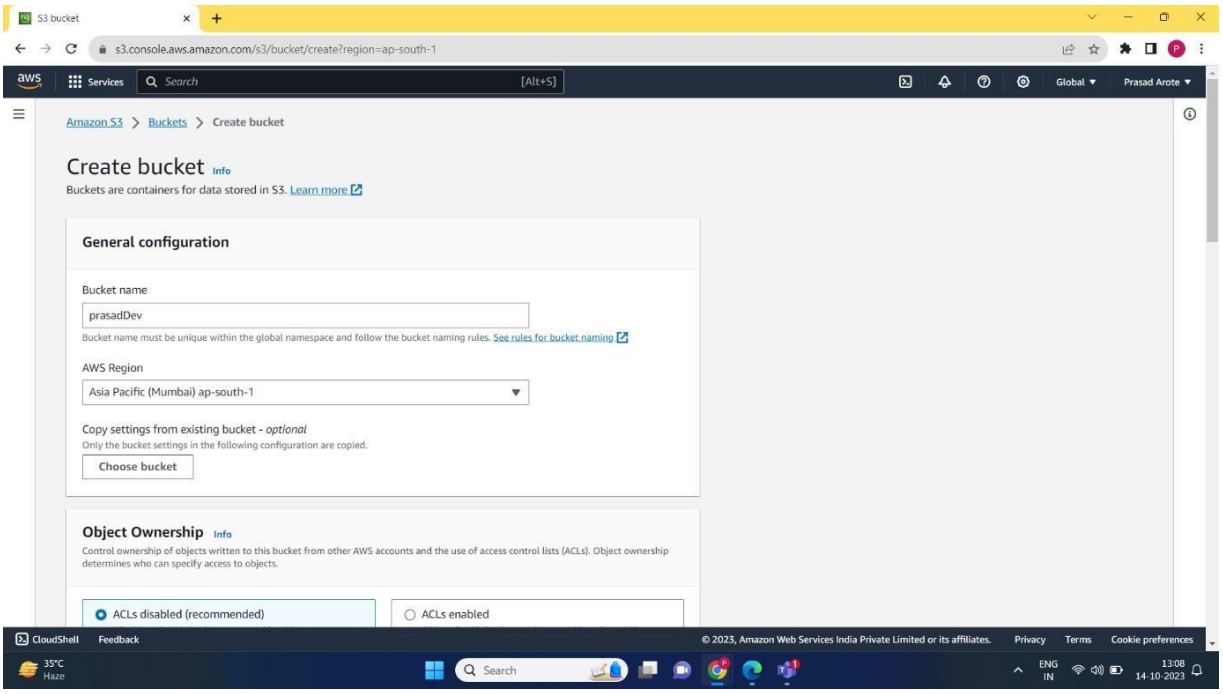
LAMBDA FUNCTION

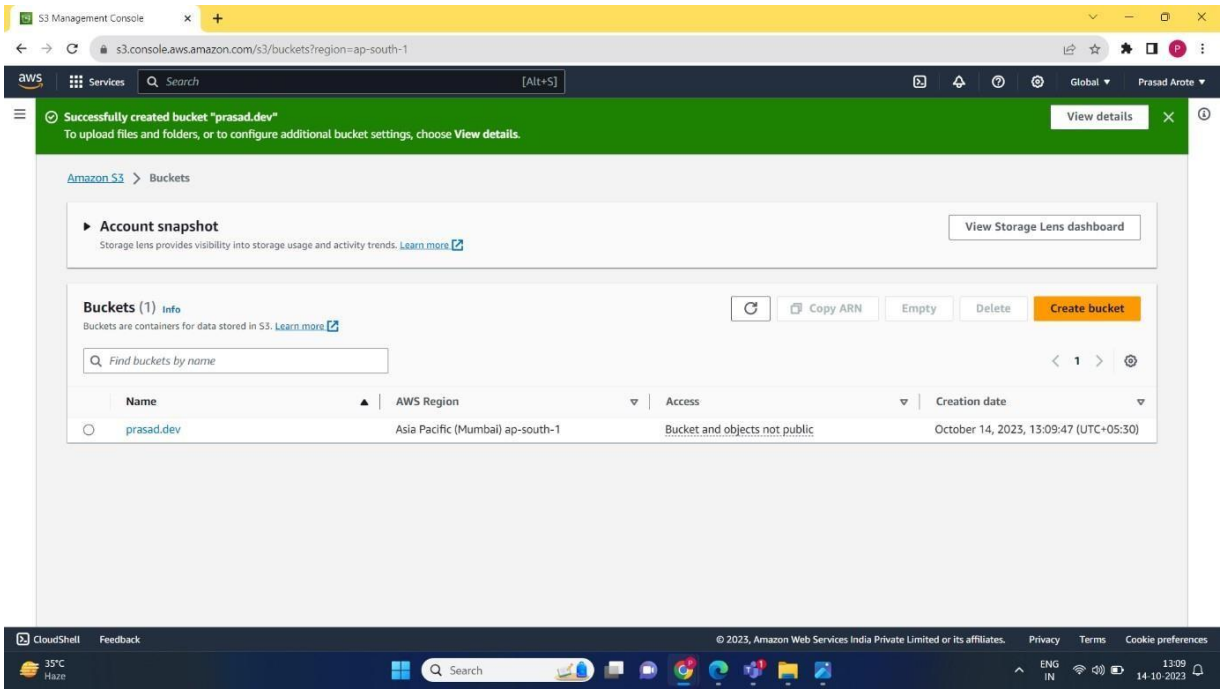
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you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

Installation:

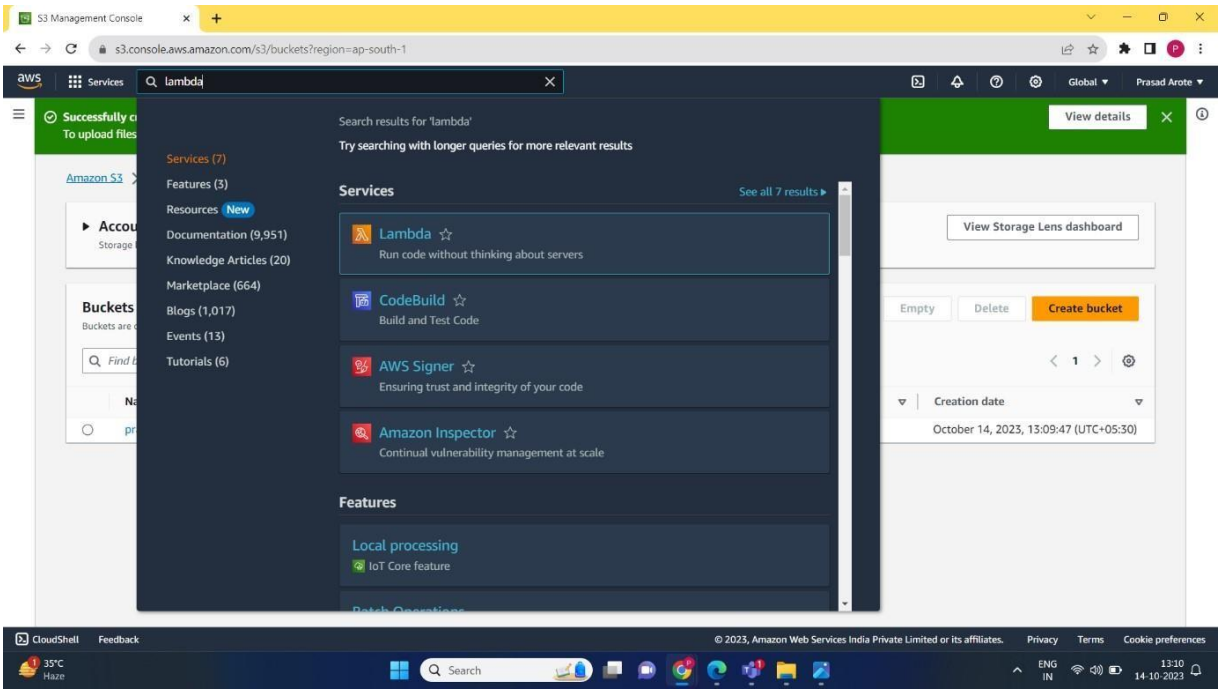
1. Create a S3 bucket

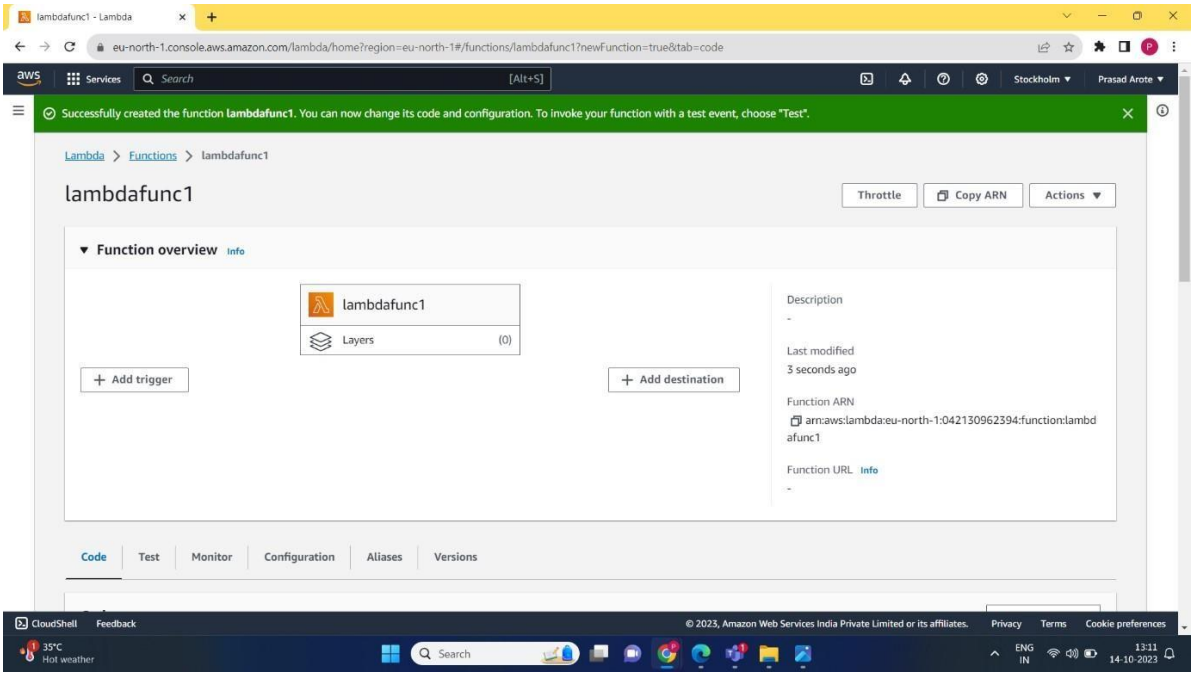


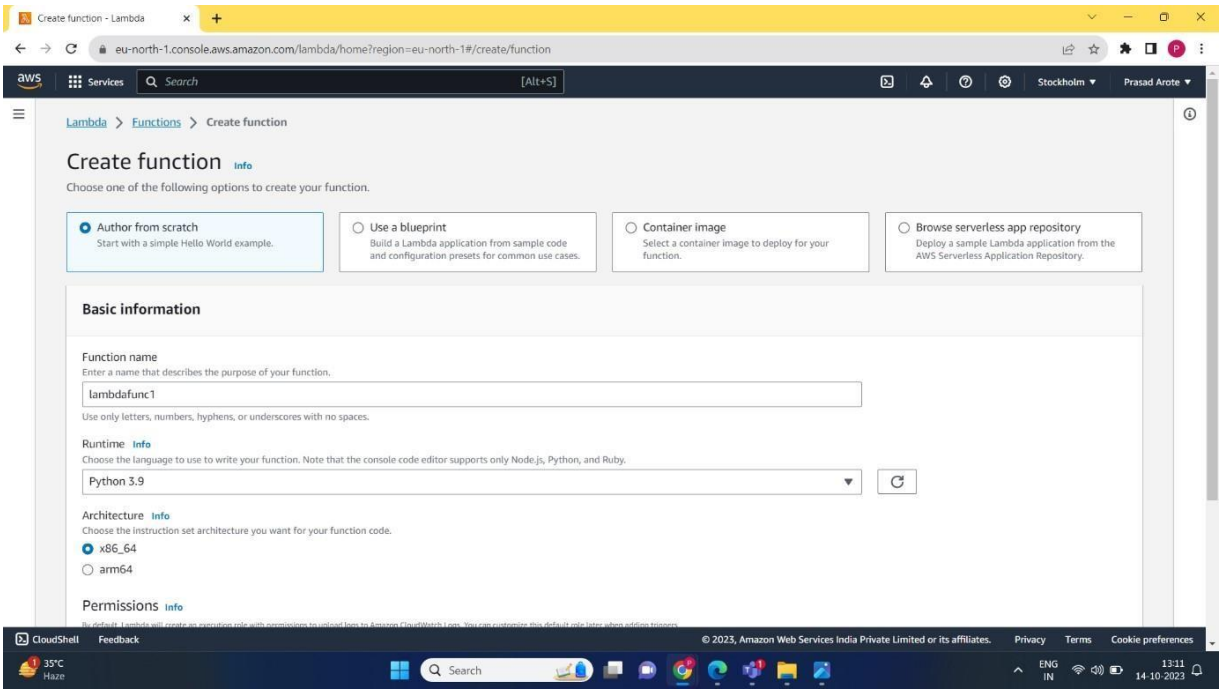




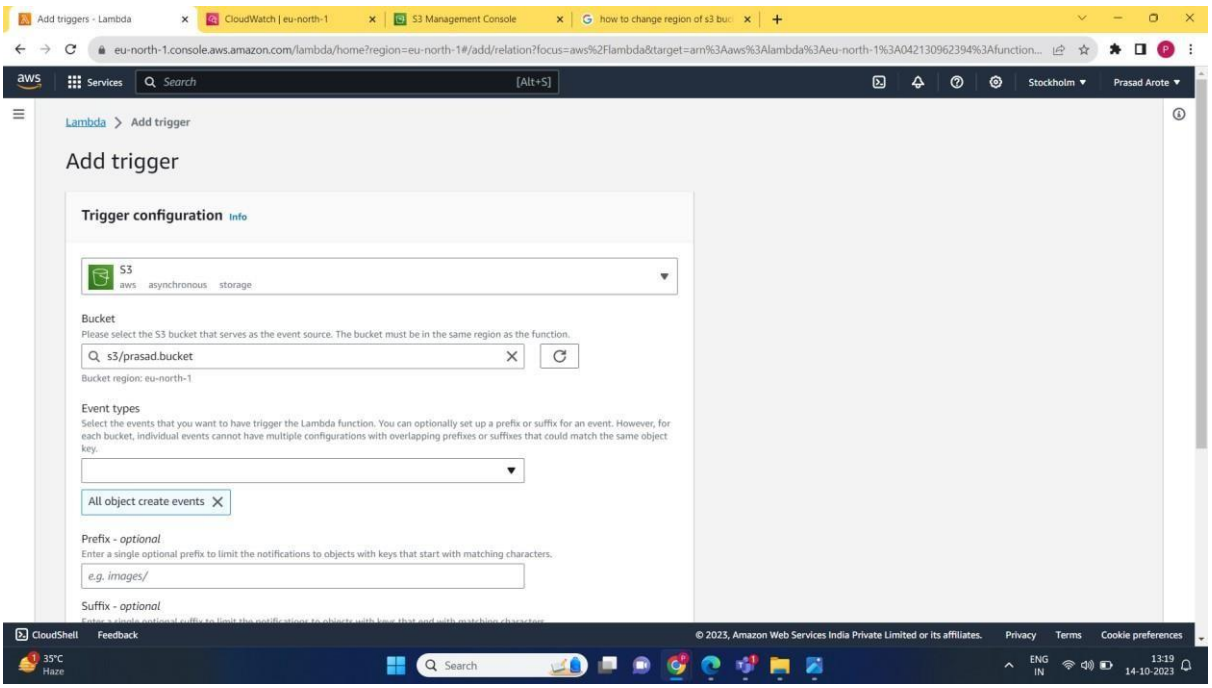
2. Create a Lambda function.

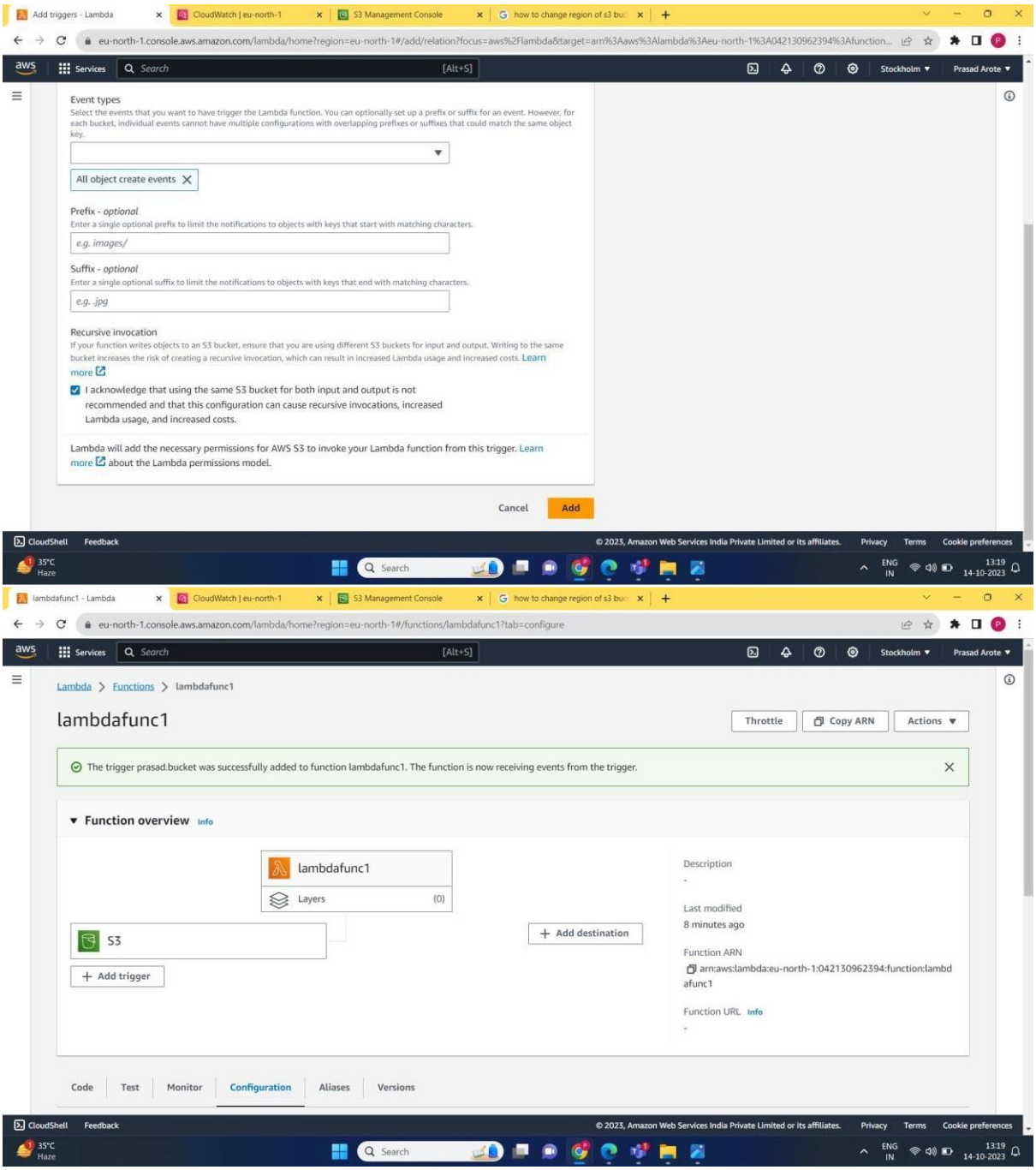

S3 Management Console
s3.console.aws.amazon.com/s3/buckets?region=ap-south-1
Services
lambda
Global
Prasad Arote
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services (7)
Features (3)
Resources New
Documentation (9,951)
Knowledge Articles (20)
Marketplace (664)
Blogs (1,017)
Events (13)
Tutorials (6)
Services
See all 7 results
Lambda
Run code without thinking about servers
CodeBuild
Build and Test Code
AWS Signer
Ensuring trust and integrity of your code
Amazon Inspector
Continual vulnerability management at scale
Features
Local processing
IoT Core feature
View details
Amazon S3
View Storage Lens dashboard
Buckets
Empty
Delete
Create bucket
Creation date
October 14, 2023, 13:09:47 (UTC+05:30)
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
13:10
14-10-2023


lambdafunc1 - Lambda
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc1?newFunction=true&tab=code
Services
Search
[Alt+S]
Stockholm
Prasad Arote
Successfully created the function lambdafunc1. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > lambdafunc1
lambdafunc1
Throttle
Copy ARN
Actions
Function overview Info
lambdafunc1
Layers (0)
Add trigger
Add destination
Description
-
Last modified
3 seconds ago
Function ARN
arn:aws:lambda:eu-north-1:042130962394:function:lambdafunc1
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Hot weather
Search
ENG
IN
13:11
14-10-2023

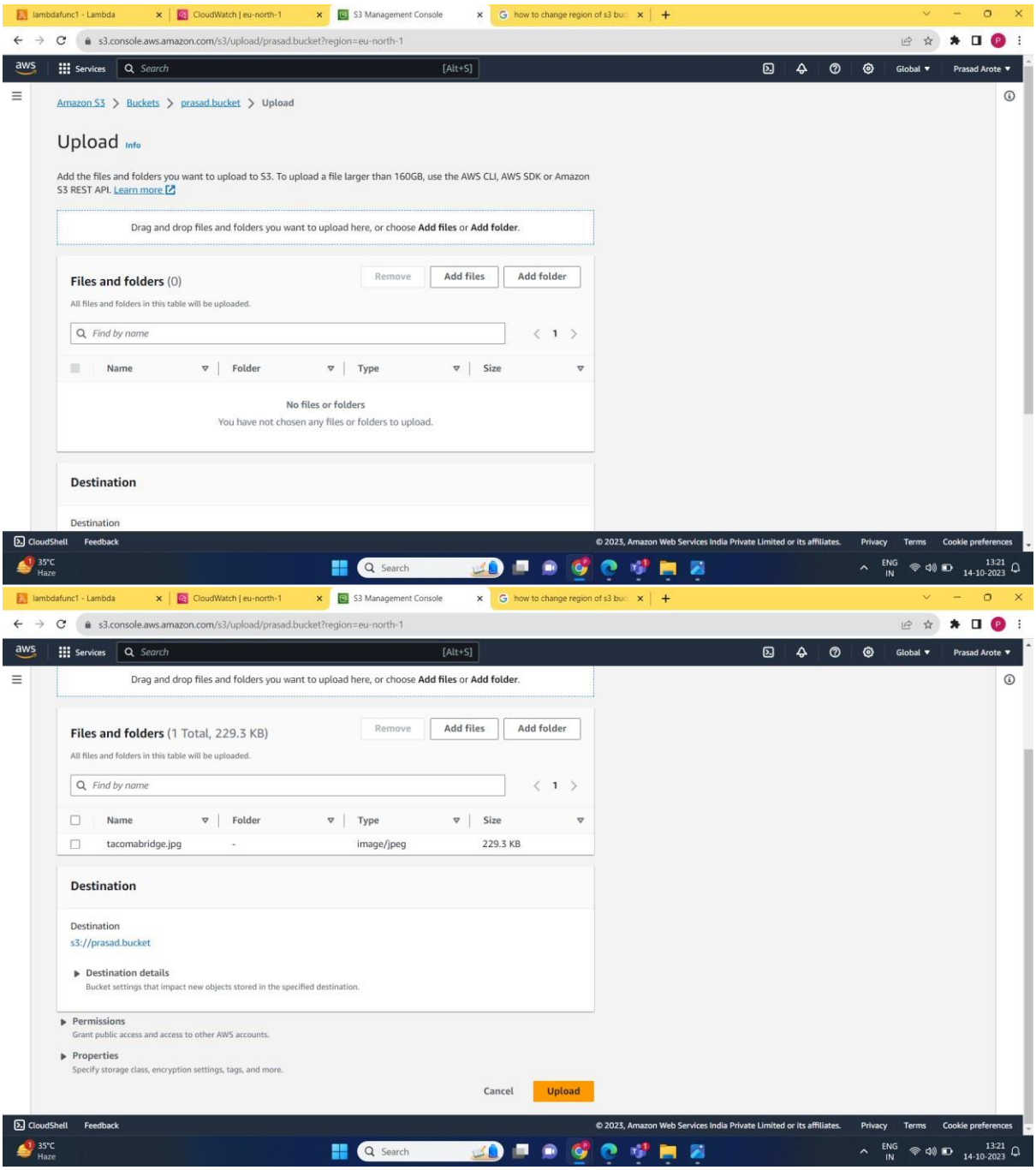


3. Create a trigger

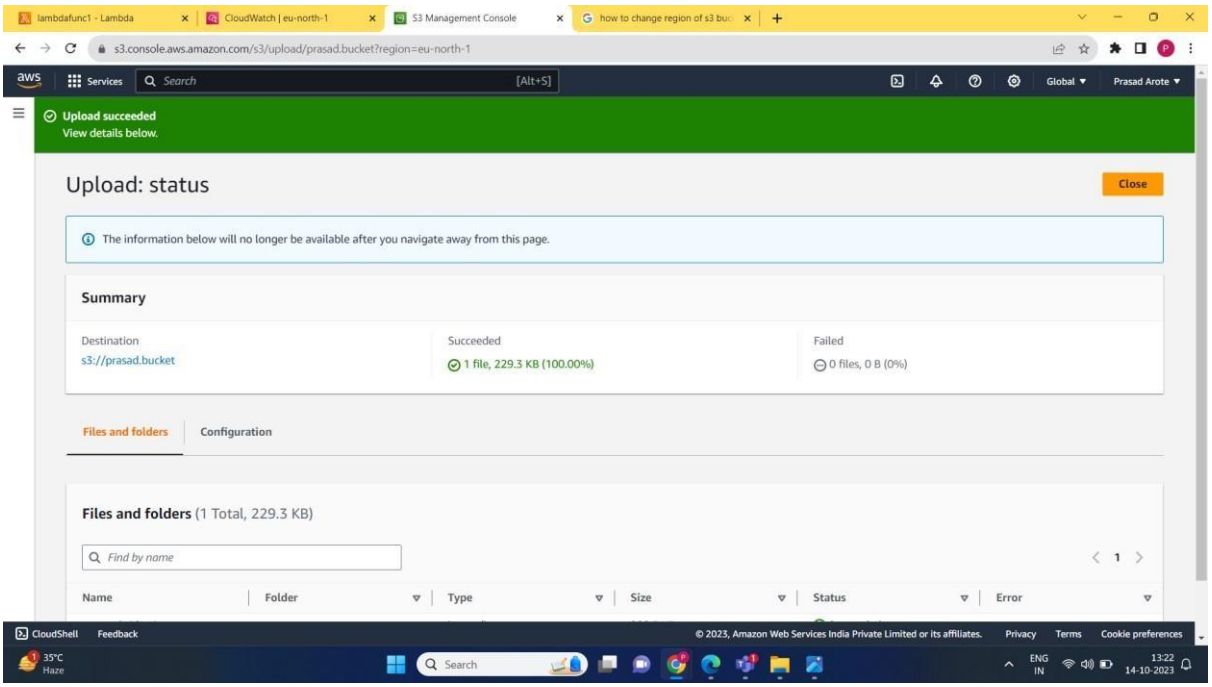



Add triggers - Lambda
CloudWatch | eu-north-1
S3 Management Console
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aeu-north-1%3A042130962394%3Afunction...
Services
Search
[Alt+S]
Stockholm
Prasad Arote
Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/
Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.
e.g. .jpg
Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more
I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.
Cancel
Add
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
13:19
14-10-2023

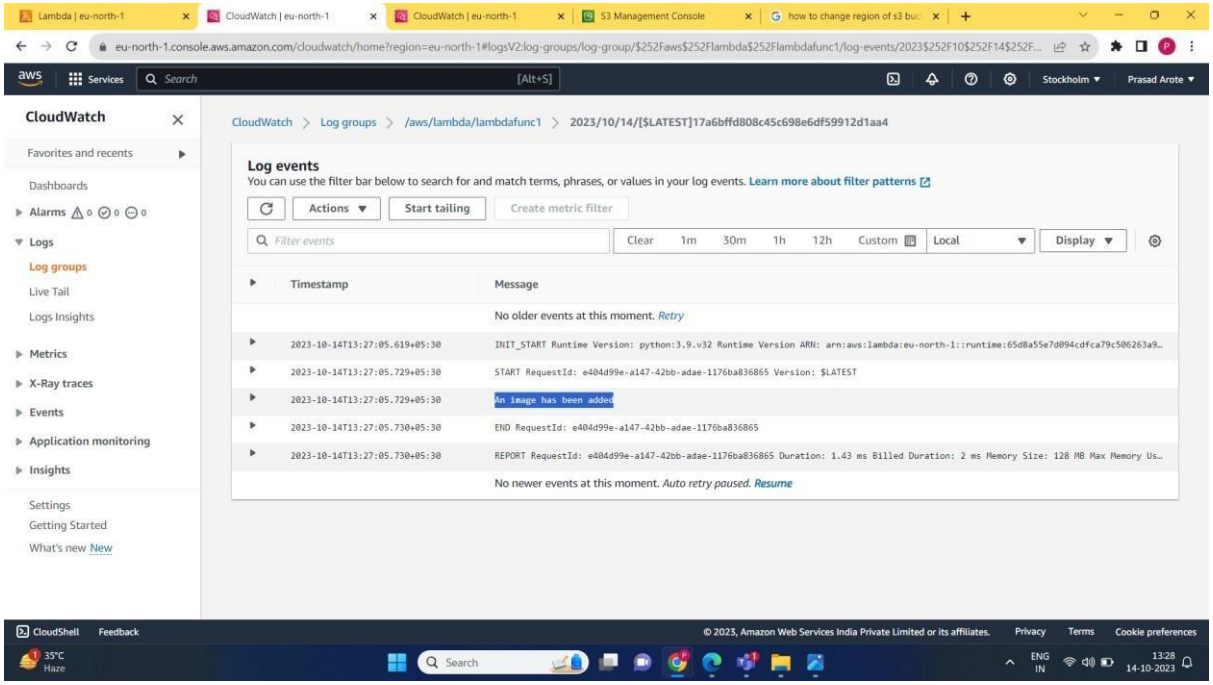
lambdafunc1 - Lambda
CloudWatch | eu-north-1
S3 Management Console
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc1?tab=configure
Services
Search
[Alt+S]
Stockholm
Prasad Arote
Lambda > Functions > lambdafunc1
lambdafunc1
Throttle
Copy ARN
Actions
The trigger prasad.bucket was successfully added to function lambdafunc1. The function is now receiving events from the trigger.
Function overview Info
lambdafunc1
Layers (0)
S3
Add trigger
Add destination
Description
-
Last modified
8 minutes ago
Function ARN
arn:aws:lambda:eu-north-1:042130962394:function:lambdafunc1
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
13:19
14-10-2023


lambdafunc1 - Lambda
CloudWatch | eu-north-1
S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad.bucket?region=eu-north-1
Services
Search
[Alt+S]
Global
Prasad Arote
Amazon S3 > Buckets > prasad.bucket > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (0)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
Name
Folder
Type
Size
No files or folders
You have not chosen any files or folders to upload.
Destination
Destination
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
13:21
14-10-2023

lambdafunc1 - Lambda
CloudWatch | eu-north-1
S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad.bucket?region=eu-north-1
Services
Search
[Alt+S]
Global
Prasad Arote
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 229.3 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
Name
Folder
Type
Size
tacomabridge.jpg
-
image/jpeg
229.3 KB
Destination
Destination
s3://prasad.bucket
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.
Properties
Specify storage class, encryption settings, tags, and more.
Cancel
Upload
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
13:21
14-10-2023



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



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

Assignment 10

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

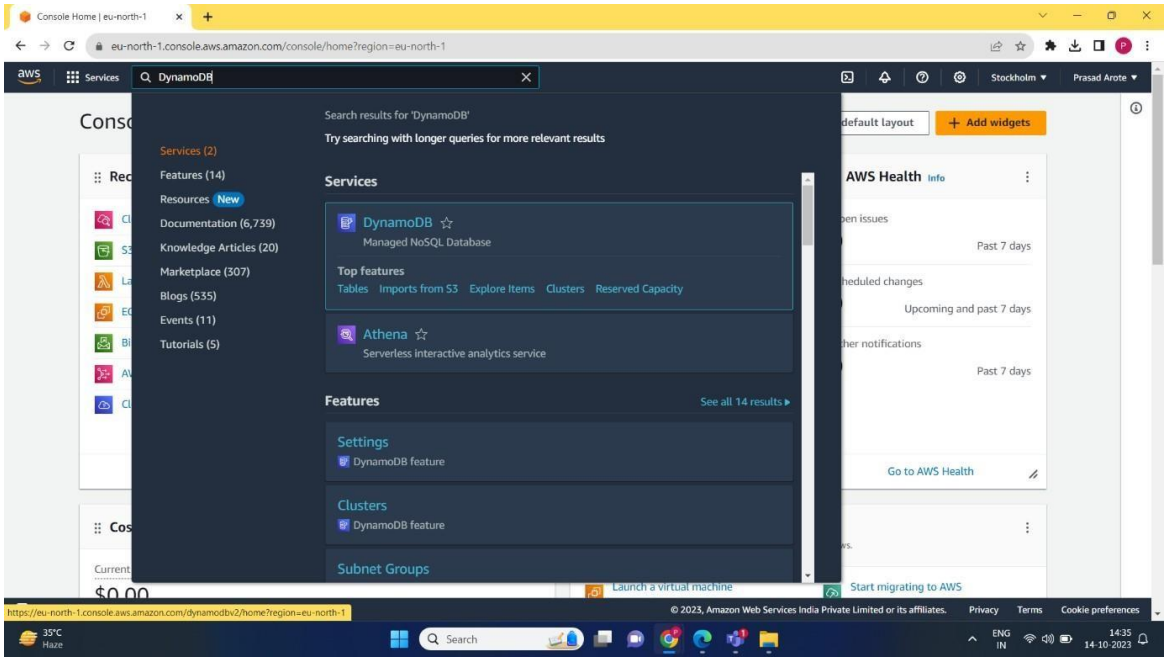
LO6: To engineer a composition of nano services using AWS Lambda and Step Functions with the serverless framework.

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.

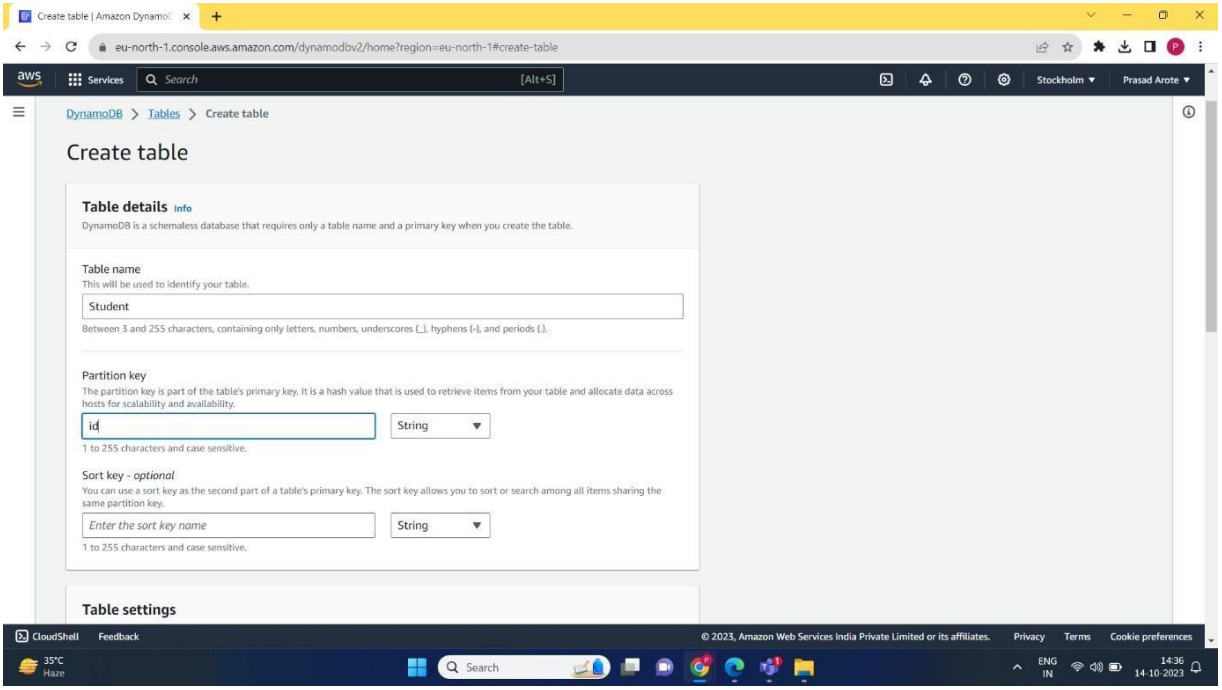
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource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



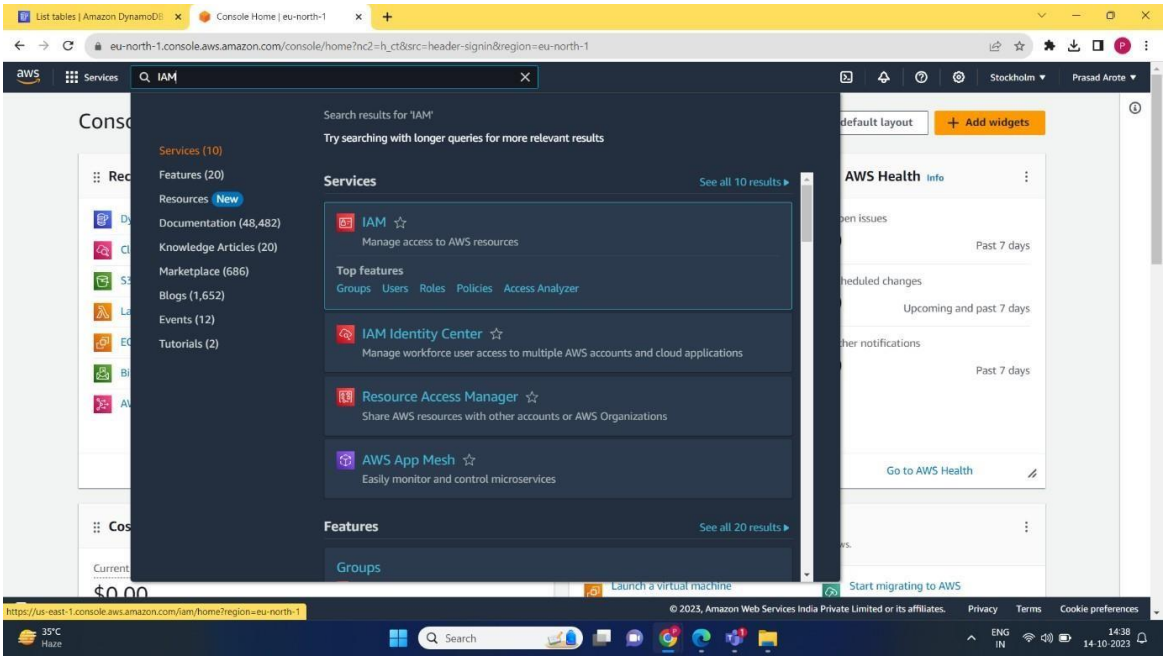
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

STEPS:

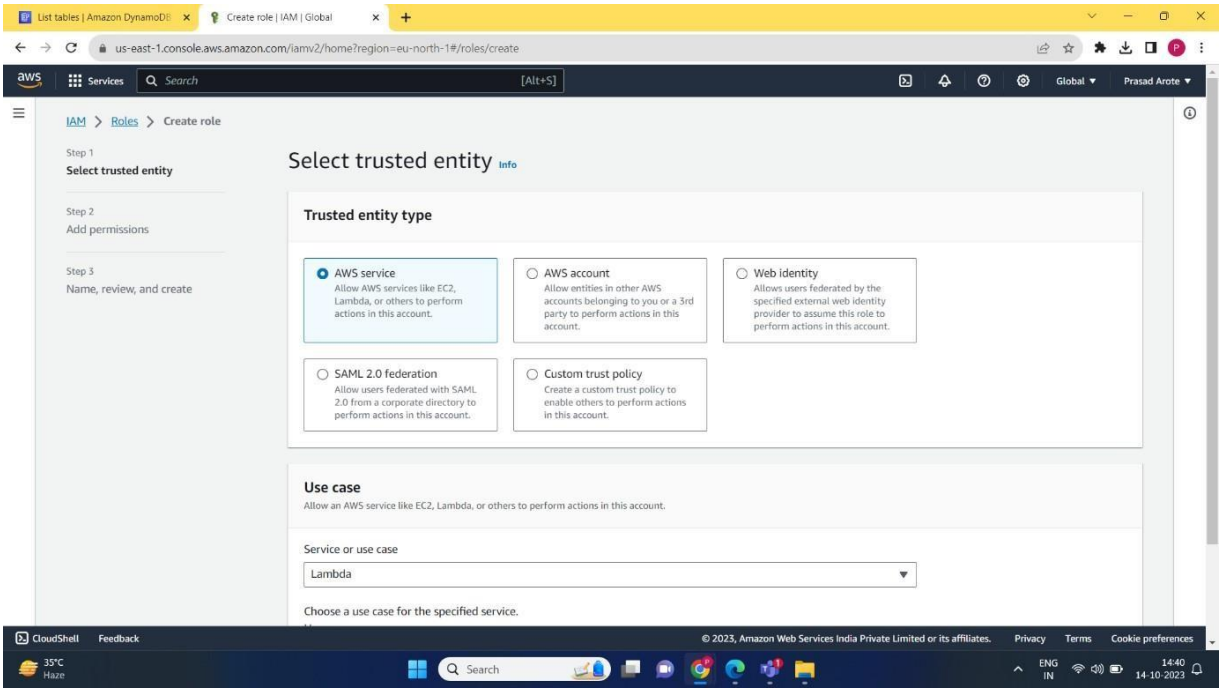
1. Create a table

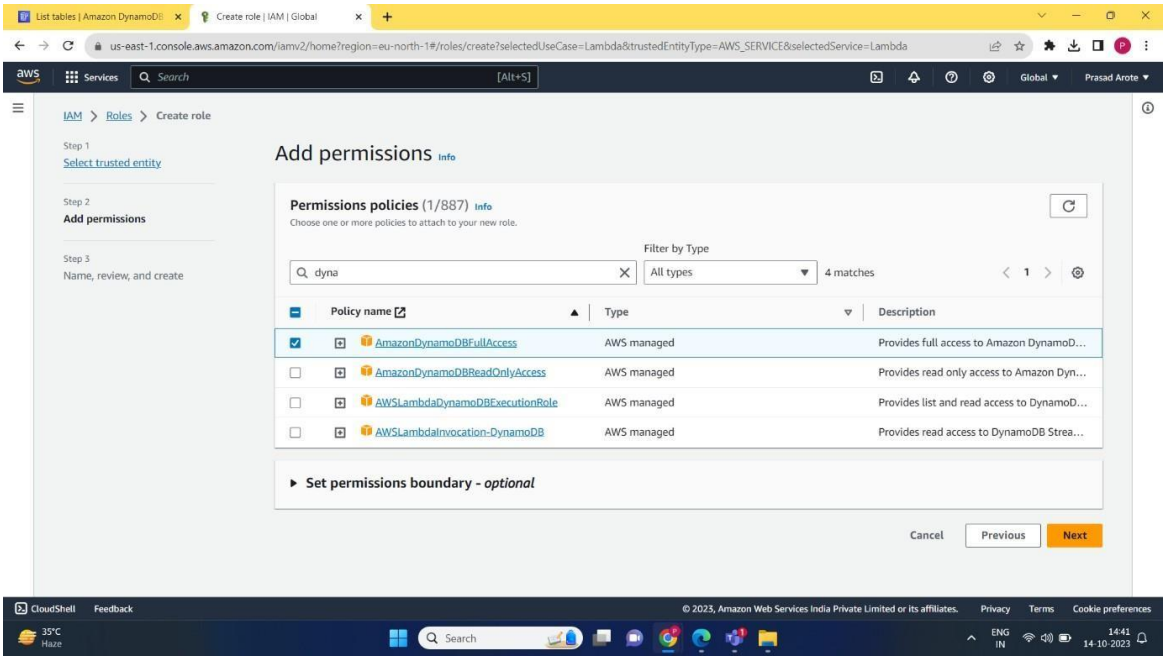


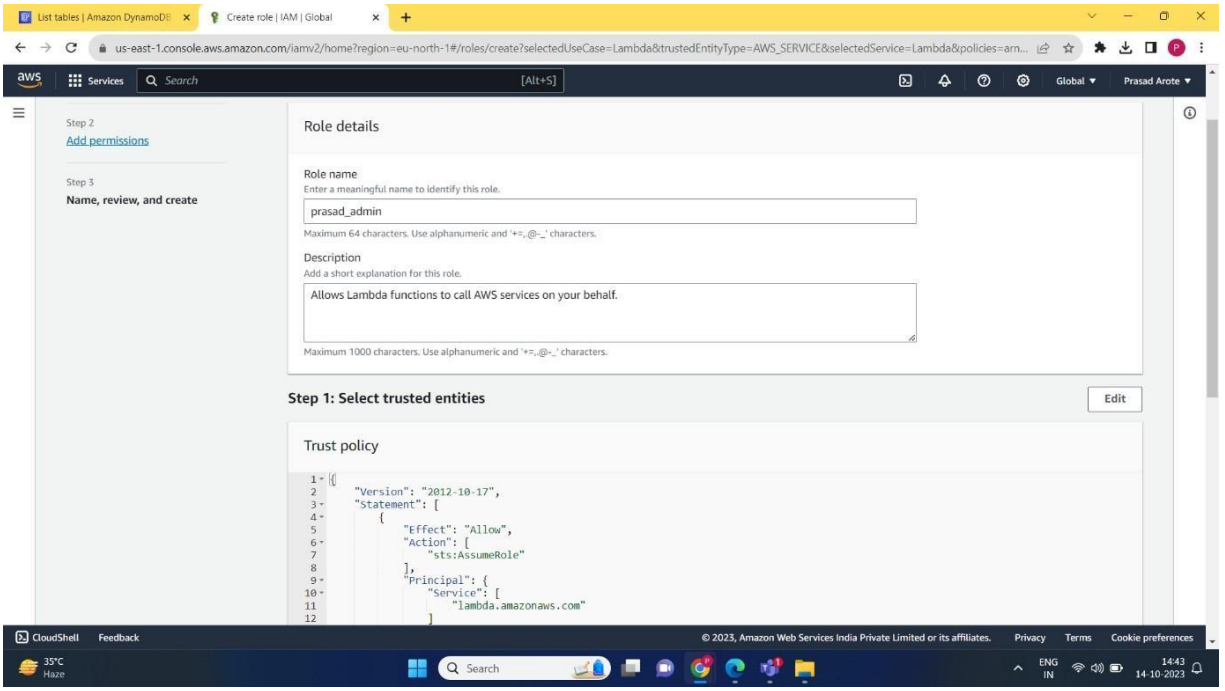
2. Create a role using IAM

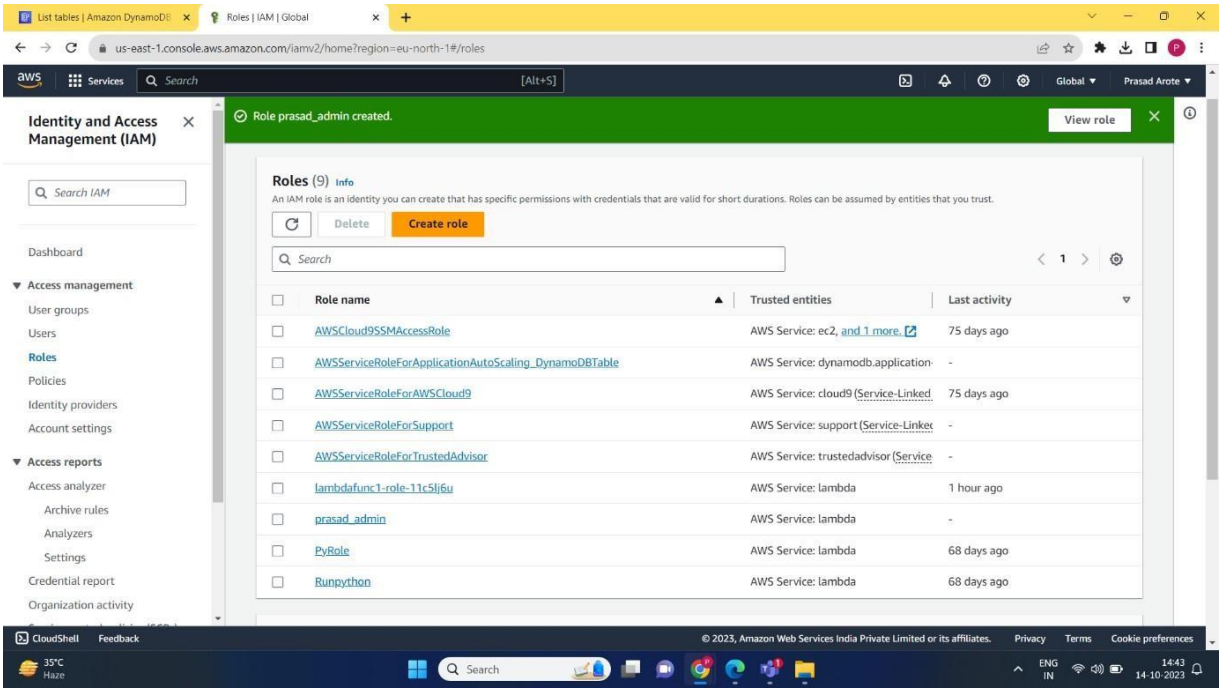


3. Add permissions – AmazonDynamoFullAccess

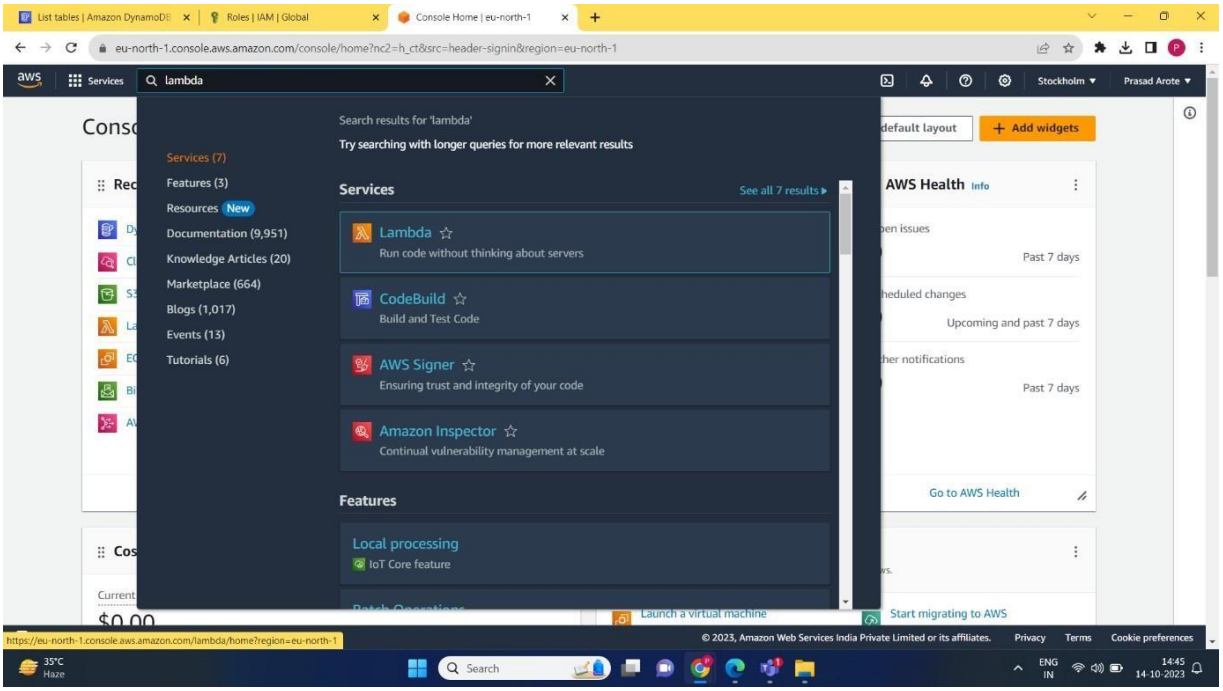


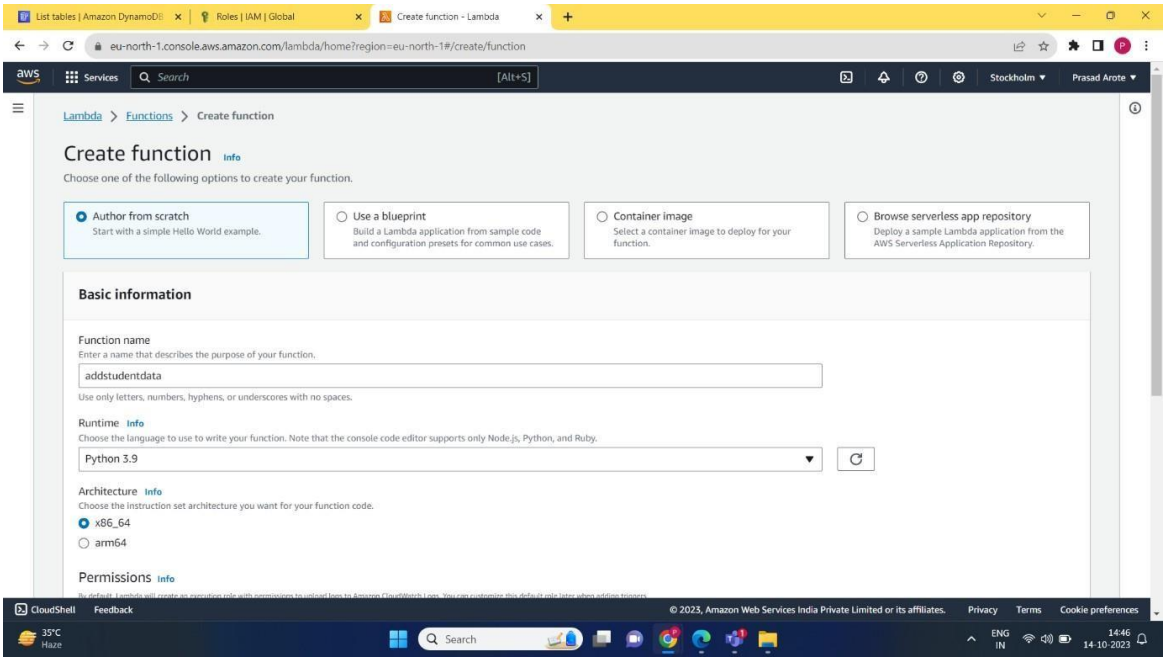


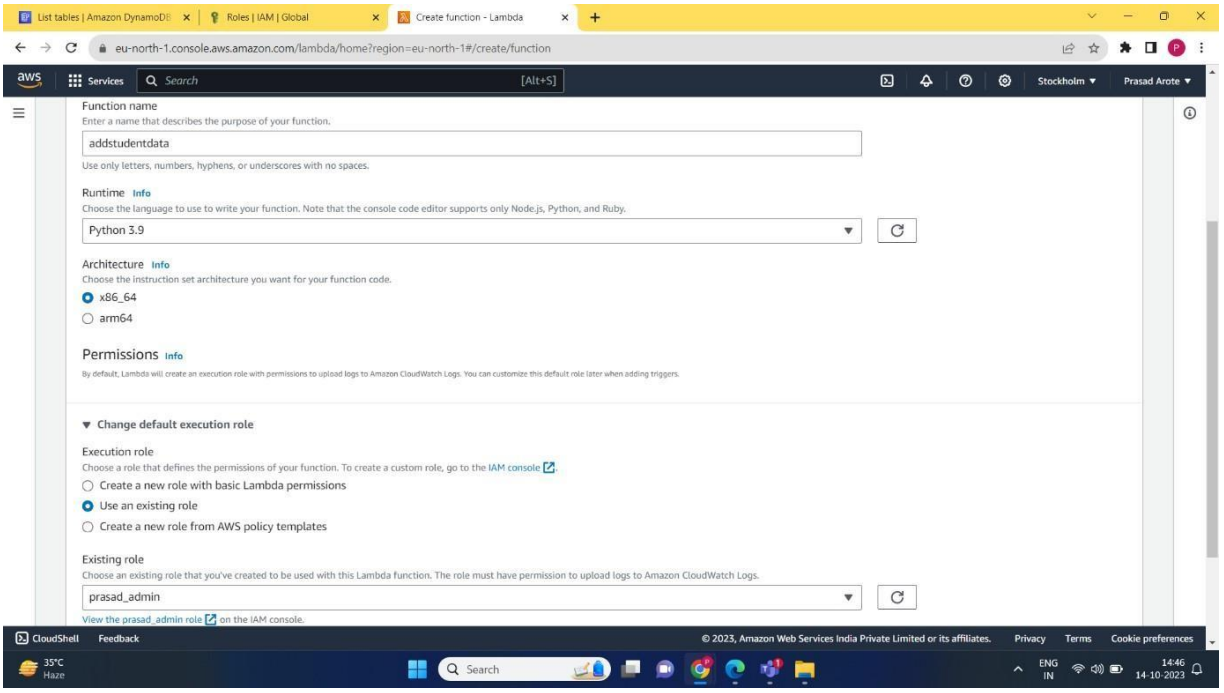


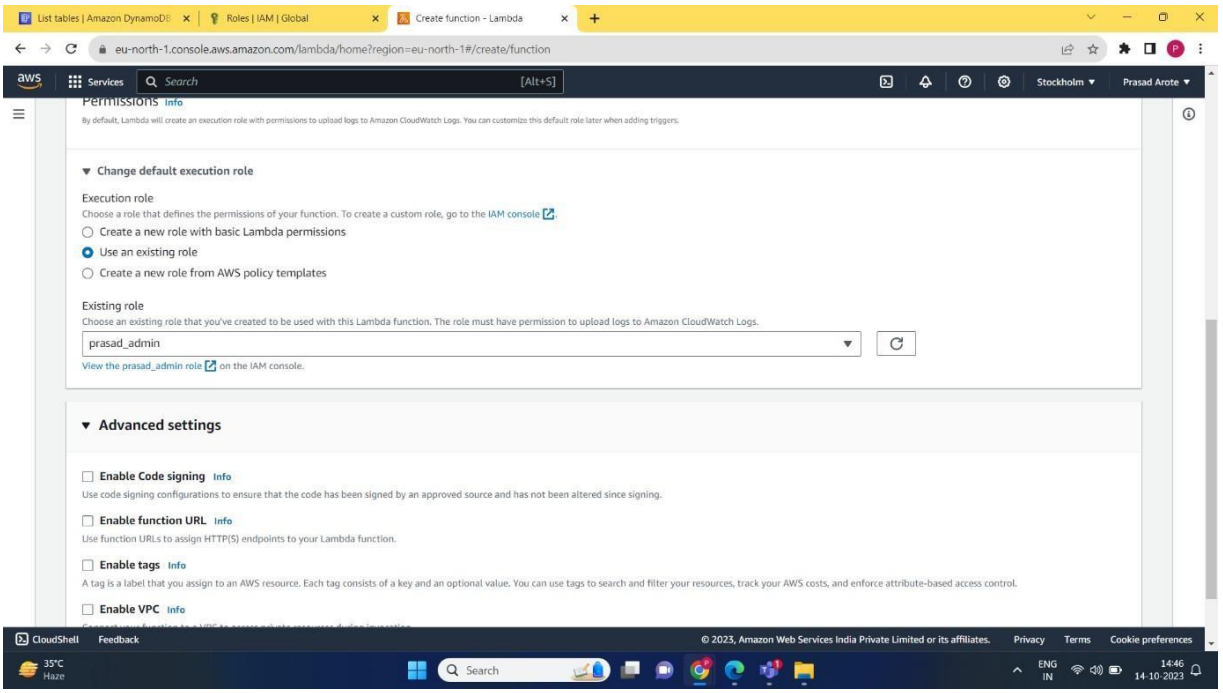


4. Create a Lambda Function

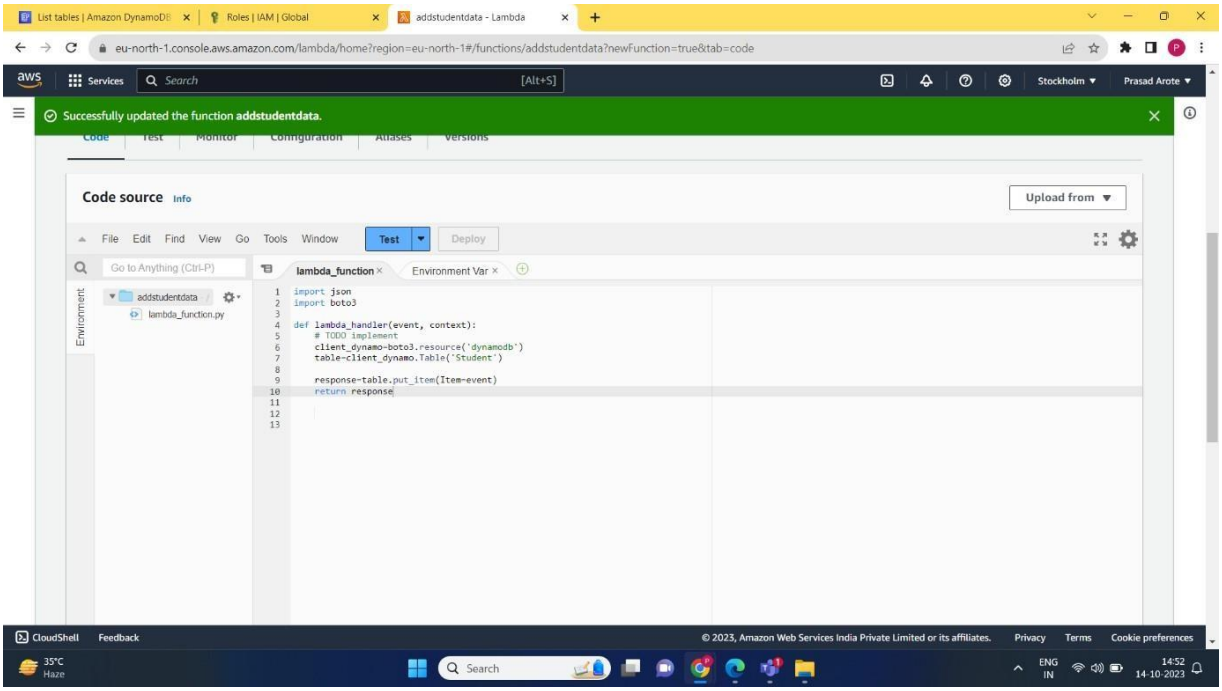

List tables | Amazon DynamoDB
Roles | IAM | Global
Console Home | eu-north-1
eu-north-1.console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin&region=eu-north-1
Services
lambda
Stockholm
Prasad Arote
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services (7)
Features (3)
Resources New
Documentation (9,951)
Knowledge Articles (20)
Marketplace (664)
Blogs (1,017)
Events (13)
Tutorials (6)
Services
See all 7 results
Lambda
Run code without thinking about servers
CodeBuild
Build and Test Code
AWS Signer
Ensuring trust and integrity of your code
Amazon Inspector
Continual vulnerability management at scale
Features
Local processing
IoT Core feature
default layout
Add widgets
AWS Health Info
Past 7 days
Upcoming and past 7 days
Go to AWS Health
Launch a virtual machine
Start migrating to AWS
https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
14:45
14-10-2023


List tables | Amazon DynamoDB
Roles | IAM | Global
Create function - Lambda
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/create/function
Services
Search
[Alt+S]
Stockholm
Prasad Arote
Lambda > Functions > Create function
Create function Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
Container image
Select a container image to deploy for your function.
Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.
Basic information
Function name
Enter a name that describes the purpose of your function.
addstudentdata
Use only letters, numbers, hyphens, or underscores with no spaces.
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.9
Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Permissions Info
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
14:46
14-10-2023

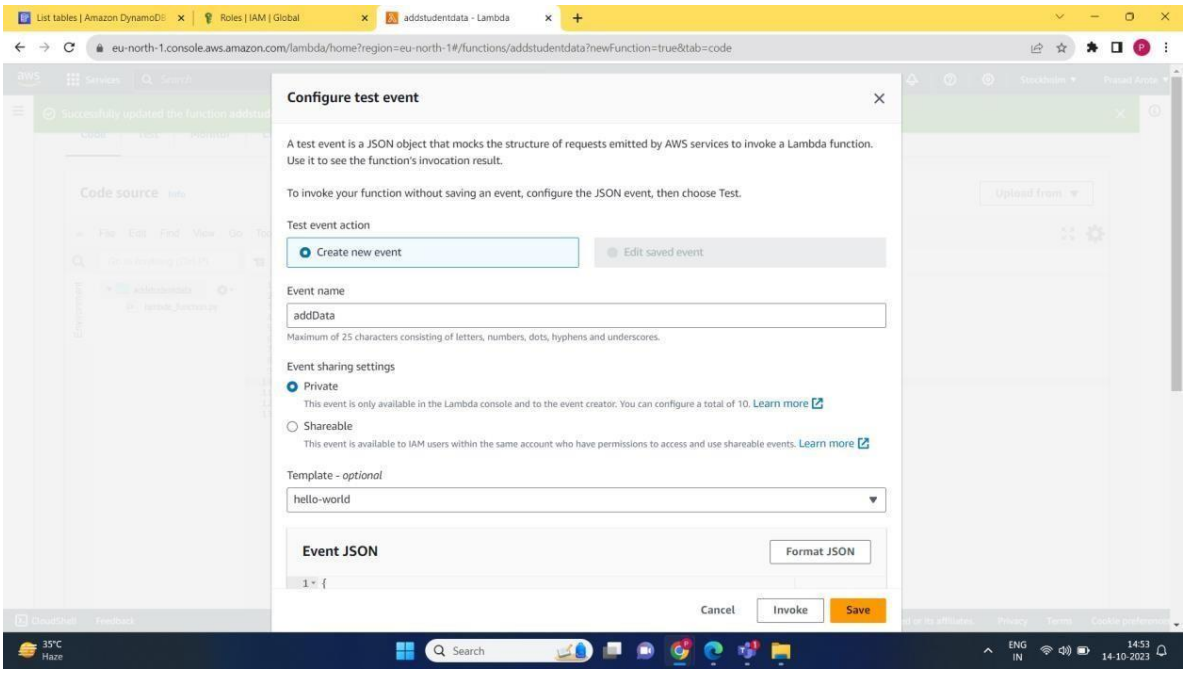


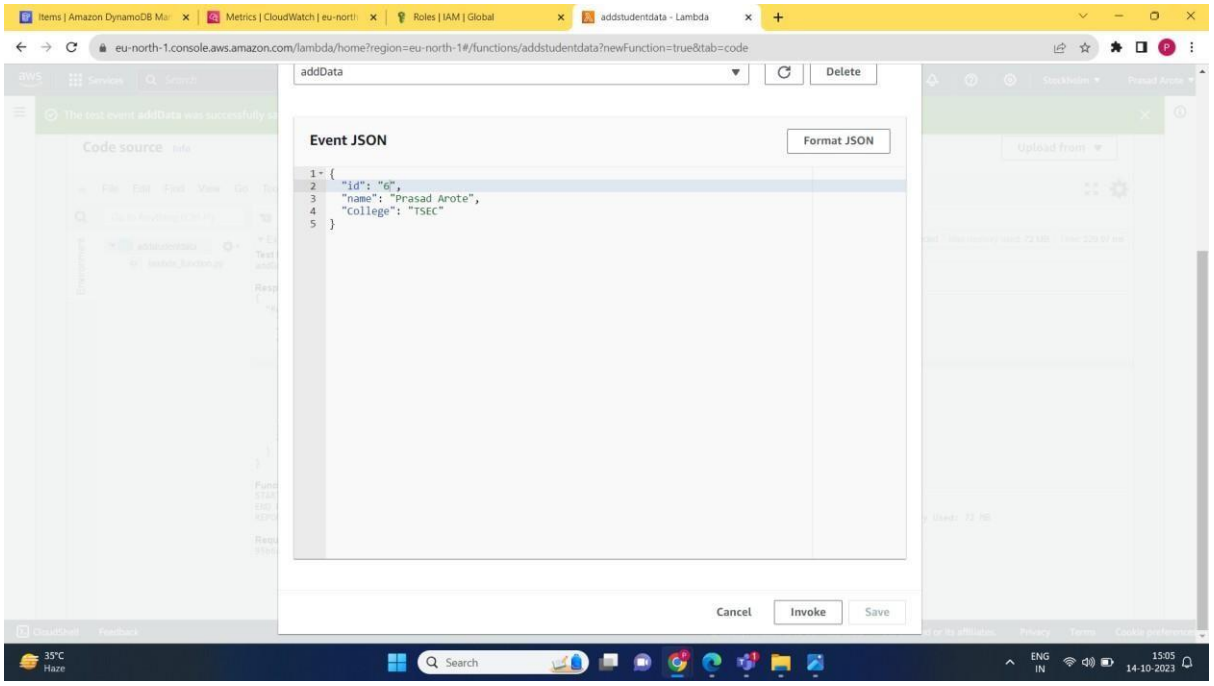


5. Write the following code

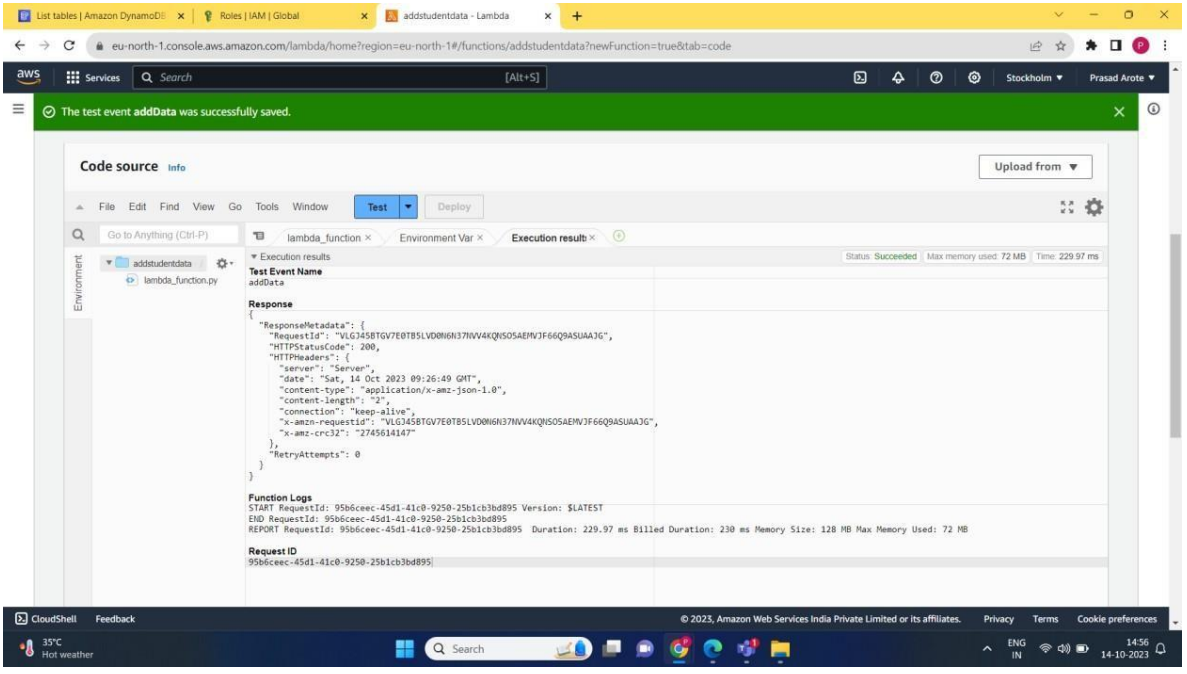


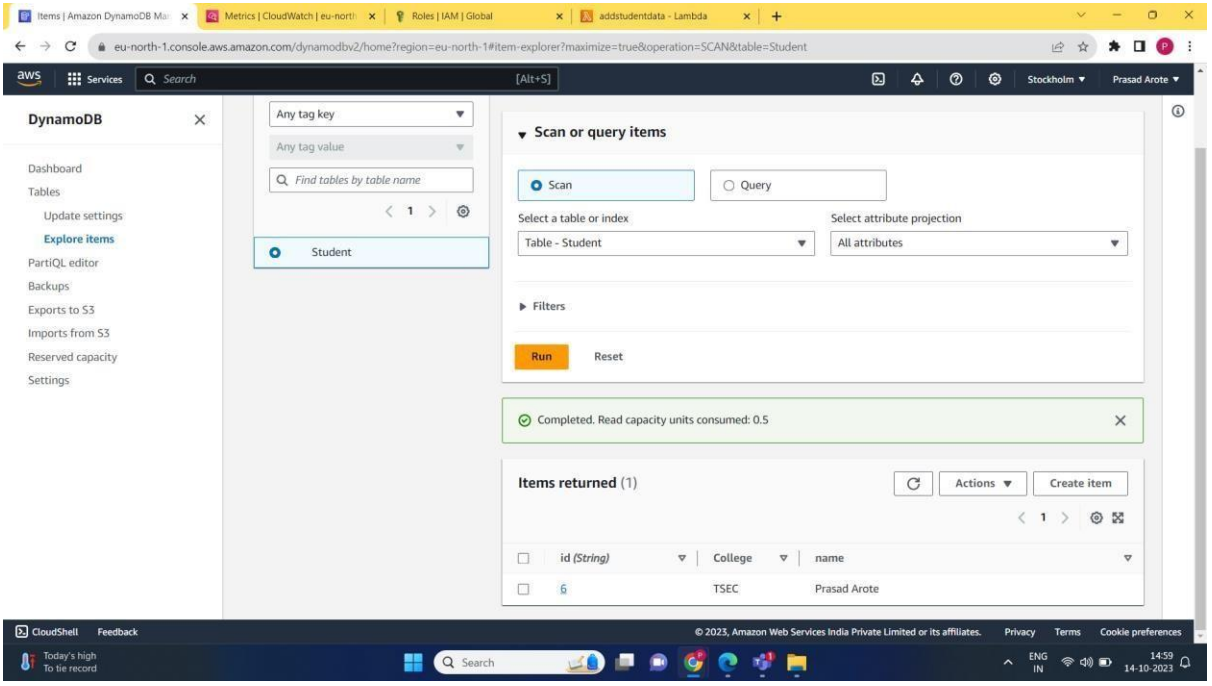
6. Configure test event and Save





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





CONCLUSION:

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.