Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./Miss Saish Bavalekar

of IT Department, Semester IV with

Roll No. 10 has completed a course of the necessary

experiments in the subject Adv Devops Lab under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

Teacher In- Charge

Head of the Department

Date 21/10/23

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	To understand the benefit of cloud infrastructure and setup aws cloud9.		17/7/23	
2.	To build your application using AWS codebuild deploy on S3		24/7/23	
3.	To creating on S3 bucket using AWS service and uploading a file into it		31/7/23	
4.	To understand the installation proces of Terraform		7/8/23	
5.	To understand the concept of terraform and use it to create and instance using limit, plan, apply and destroy command		21/8/23	
6.	To understand the concept of sonar qube & server scanner.		28/8/23	
7.	To learn how to use lambda to run a simple program from S3 bucket		4/9/23	
8.	To learn how to use lambda to find the content type of object uploaded in S3 bucket		11/9/23	
9.	To understand and install Nagios, monitor host using Nagios		18/9/23	
10	To complete study of Kubernetes		2/10/23	
11.	Theory Assignment 1			
12.	Theory Assignment 2			

Roll number:10

Name: Saish Bavalekar

Date: 23/7/23

ASSIGNMENT 1

AIM: To make an EC2 instance in AWS

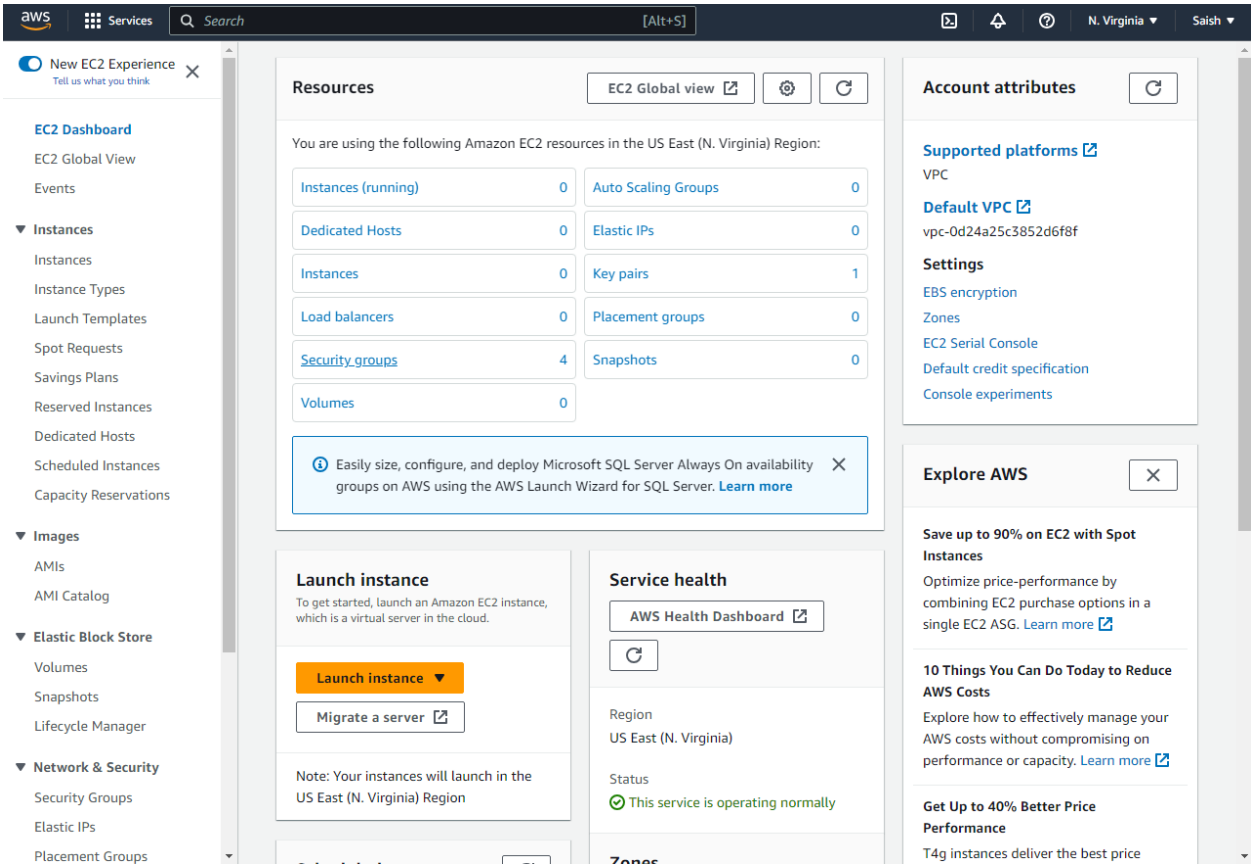
THEORY:

EC2

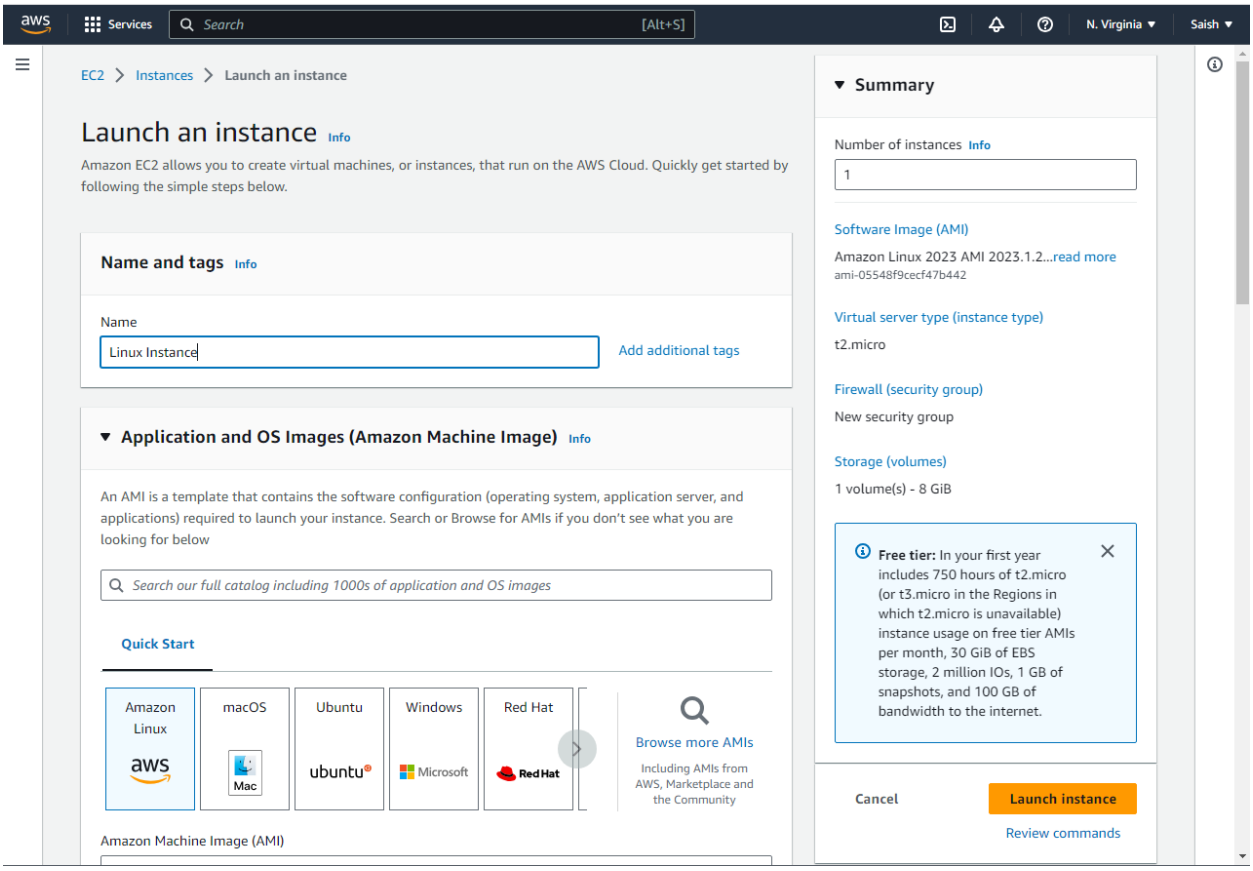
- EC2 stands for Amazon Elastic Compute Cloud.
- Amazon EC2 is a web service that provides resizable compute capacity in the cloud.
- Amazon EC2 reduces the time required to obtain and boot new user instances to minutes rather than in older days, if you need a server then you had to put a purchase order,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

STEPS TO CREATE AN EC2 INSTANCE ON AWS:

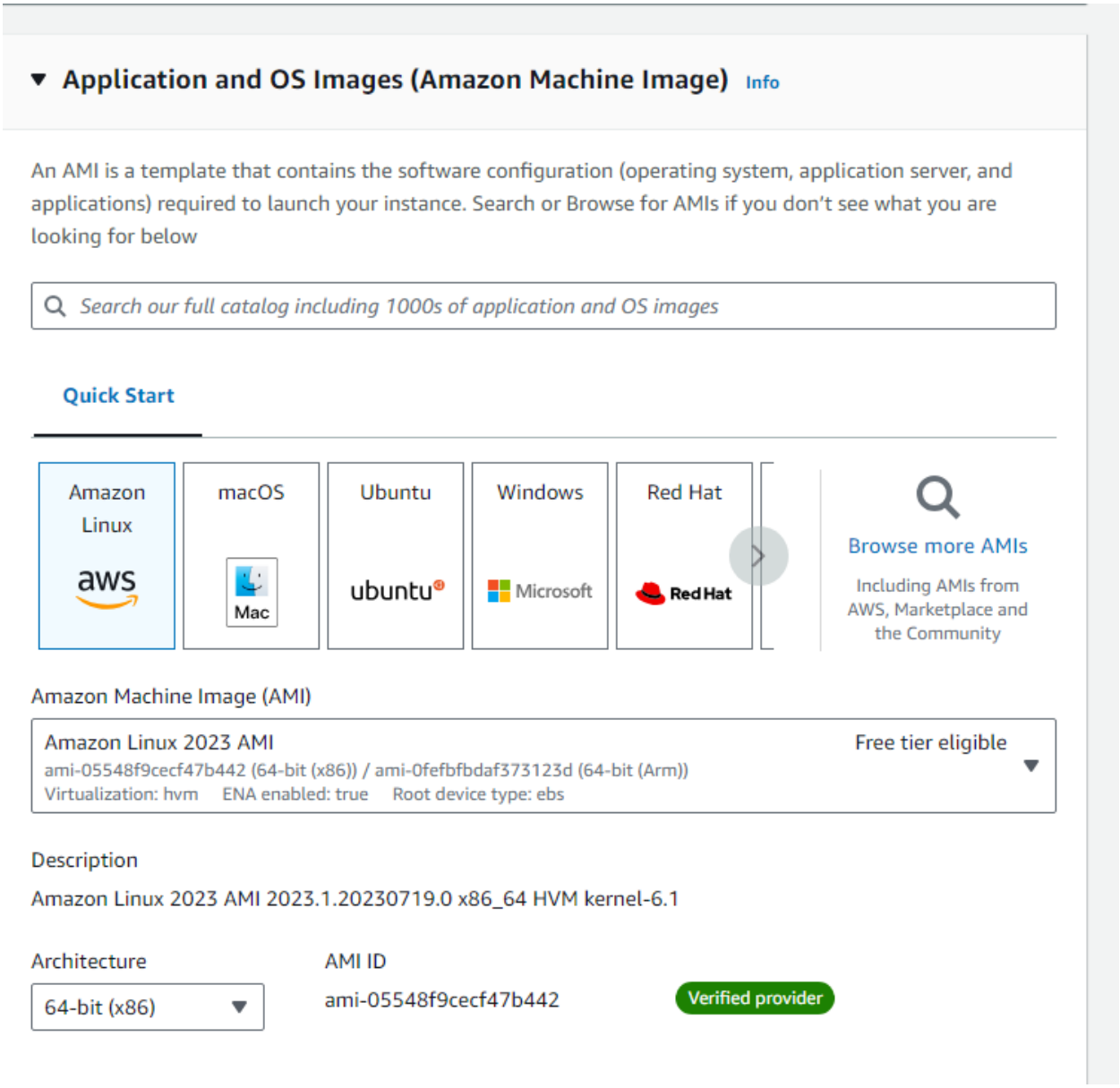
1. Login to your aws console and go to EC2 service and click on Launch Instance



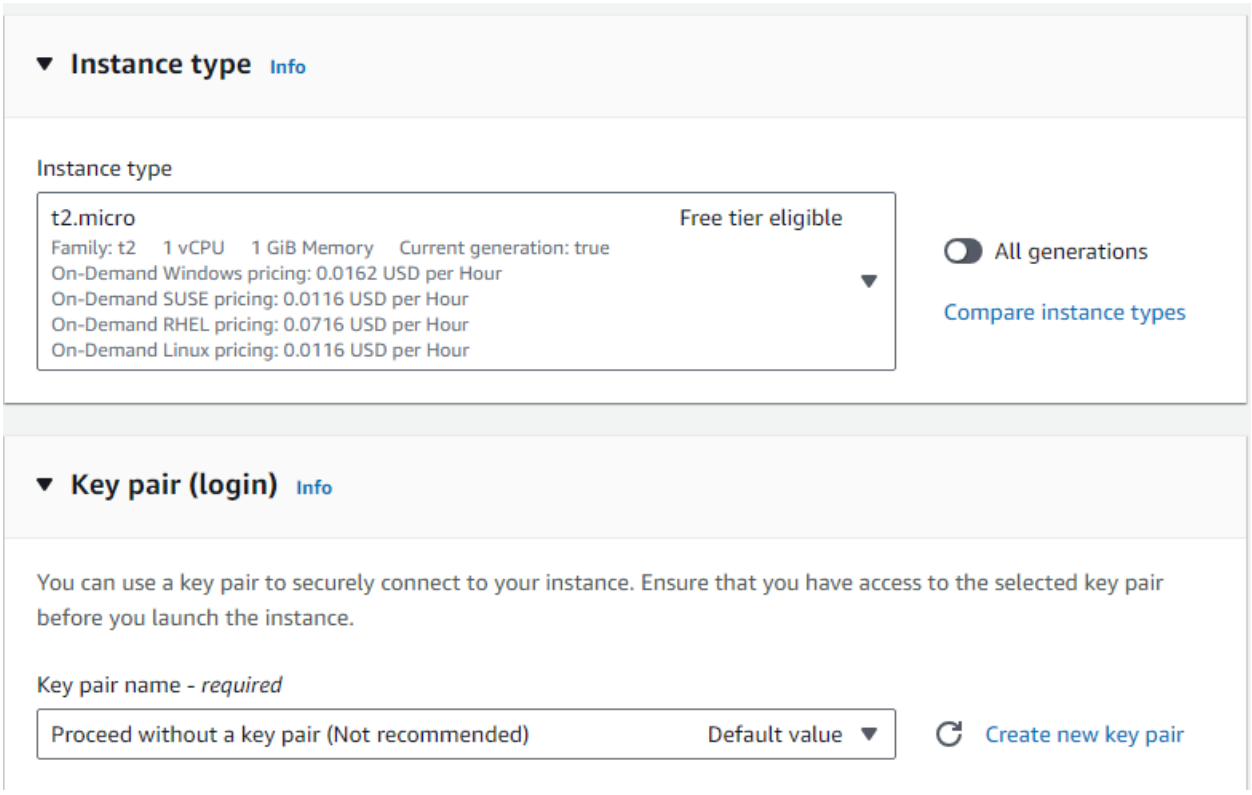
2. Name your EC2 instance



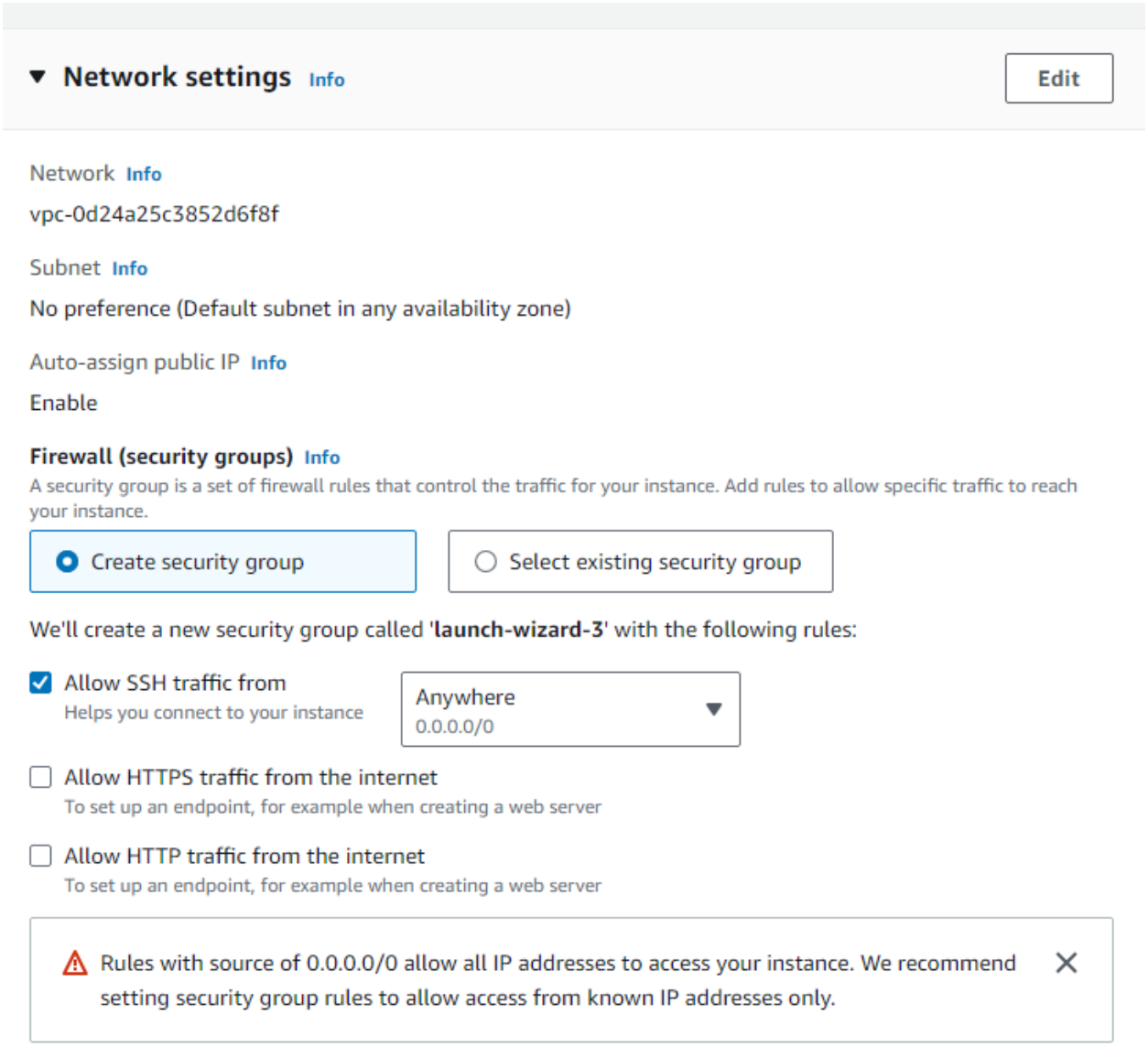
3. Select AMI(Amazon Machine Image) you wish to proceed with. Here I've selected Amazon Linux image. Make sure the AMI is free tier.



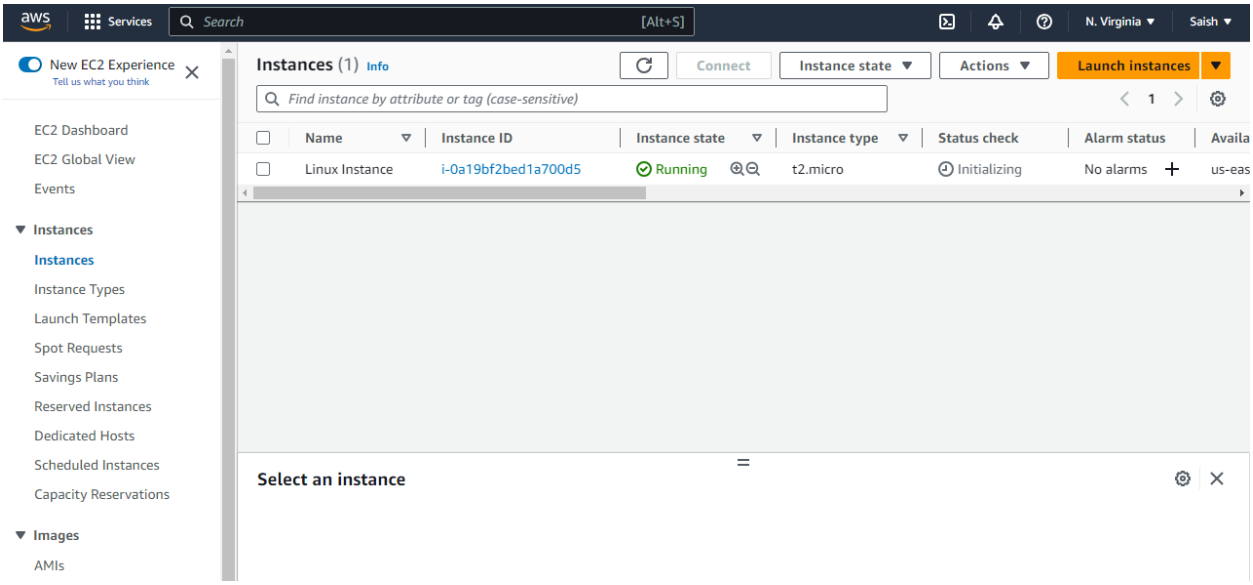
4. Select the instance type. Here we have selected t2.micro which is free tier eligible. We are proceeding without using a key pair name. Create one for more security.



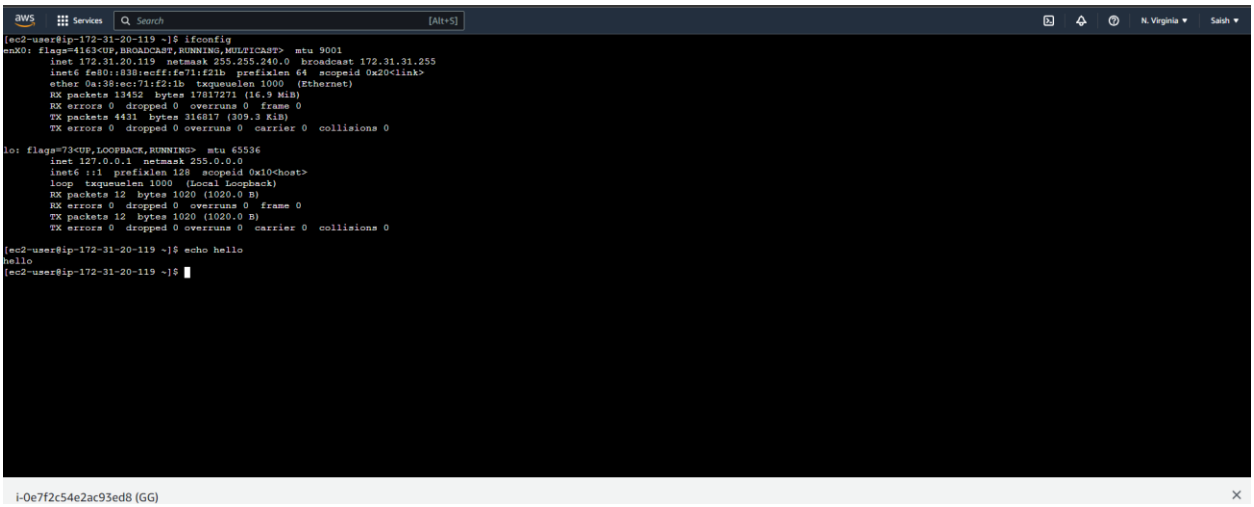
5. Add a security group or use a previously created security group and create the EC2 instance.



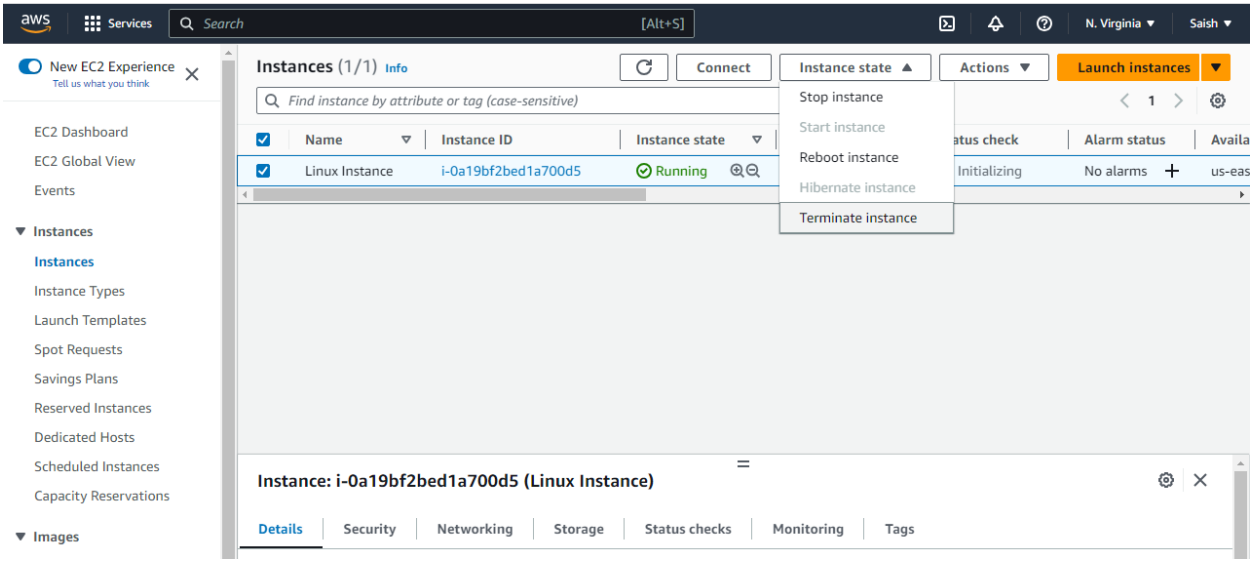
6. Launch the EC2 instance that has been created.



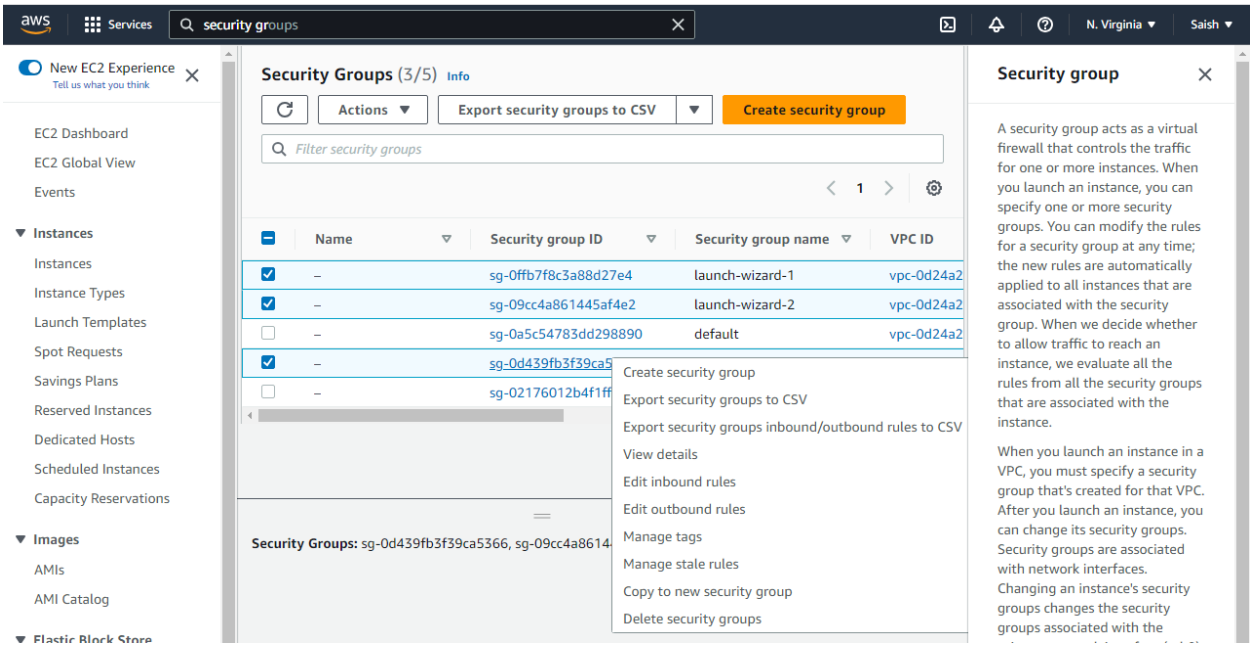
7. With the virtual linux machine at our service, we can run linux commands on the terminal.



8. Ensure to terminate the instance after use.



9. After the instance has been terminated also delete the unnecessary security groups.



LAB OUTCOME:

LO1 - To understand the benefits of Cloud9 infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Saish Bavalekar

Roll Number:10

30/7/23

ASSIGNMENT 2

AIM: To create an AWS Cloud9 Environment.

THEORY:

AWS Cloud9

Cloud9 is virtual compiler provided by AWS. We can, code, build, test, debug and release software. AWS Cloud9 environment is a place where we can store our project files and where we run the tools to develop our application.

With Cloud9 Environment we can:

-Store your project files locally on the instance or server.

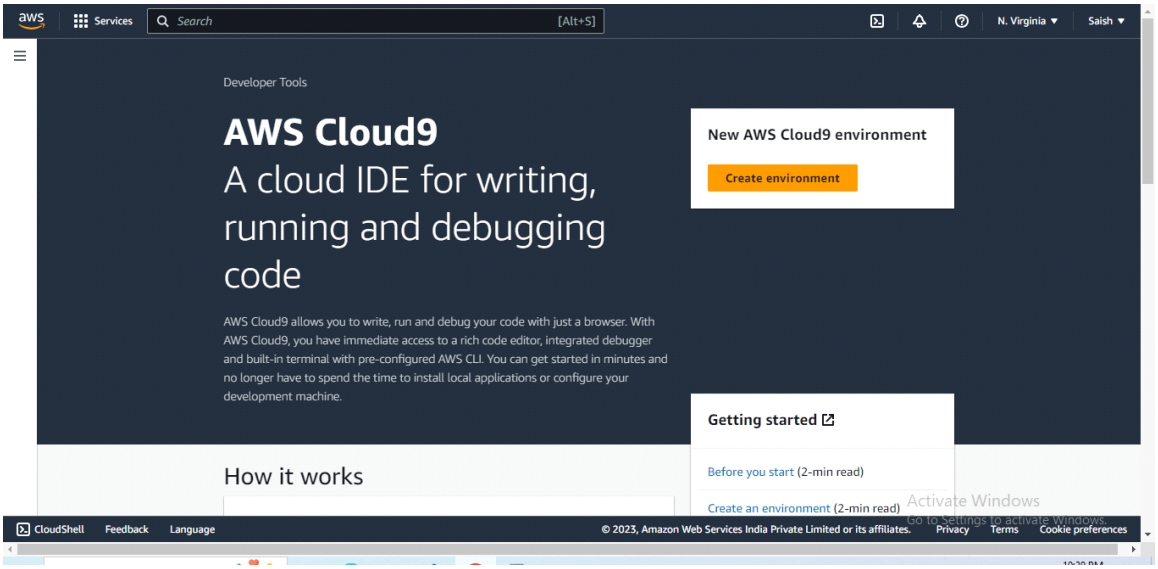
-Clone remote code repository into our environment.

-Work with a combination of local and cloned files in the environment.

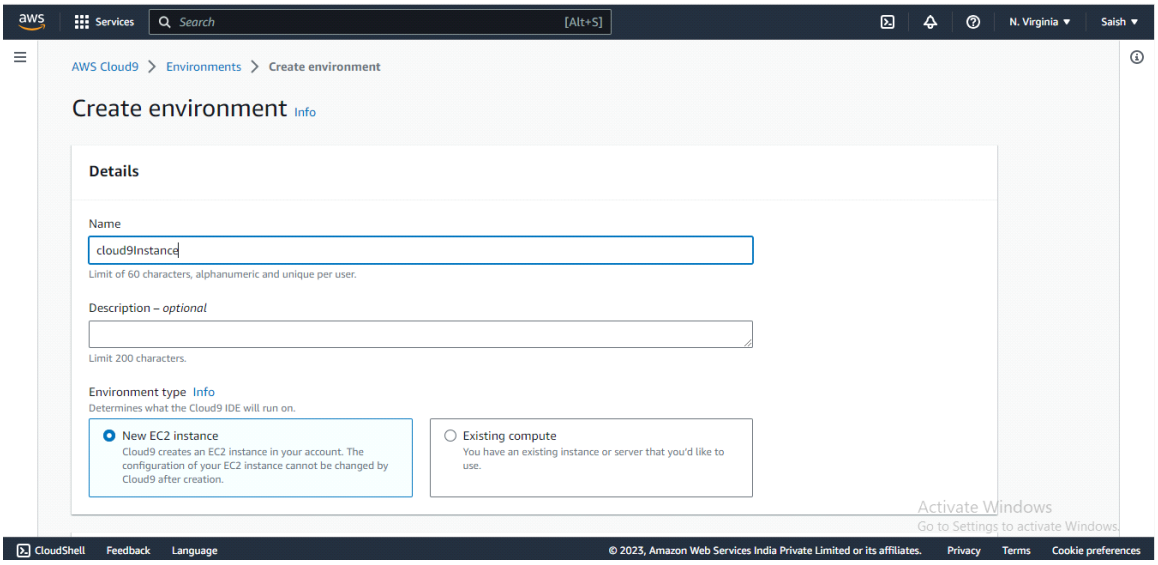
We can create and switch between multiple environments, with each environment set up for a specific development project. By storing the projects on cloud, our projects no longer need to be tied to a single computer or server setup. This enables you to do things such as easily switch between computers and more quickly onboard developers to our team.

Steps to Setup AWS Cloud9 Environment:

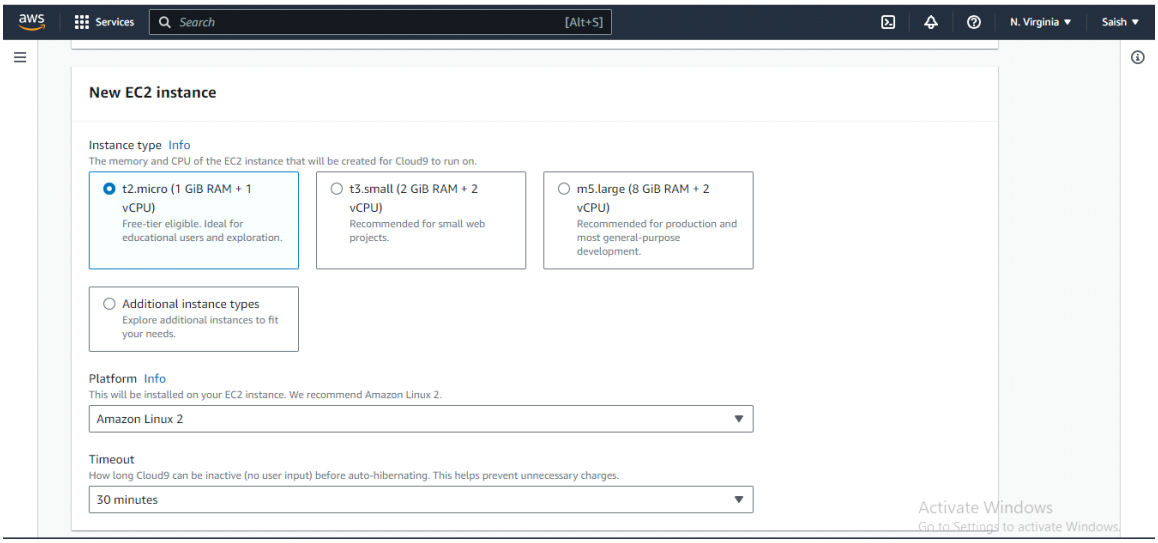
1. **Login to your AWS console and search for Cloud9 and click on create environment.**



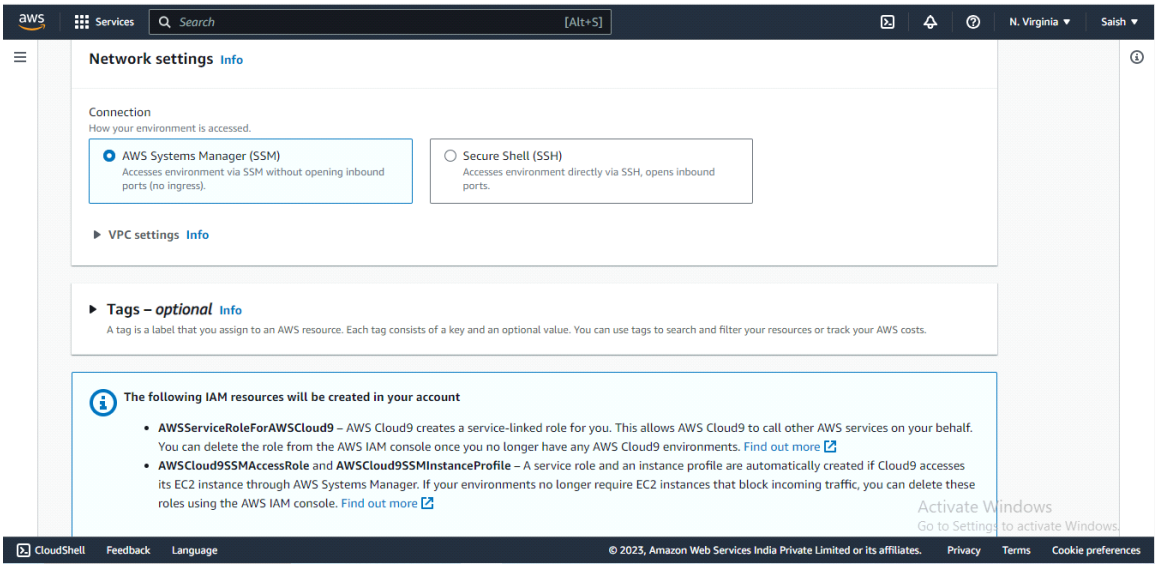
2. **Name your environment and select the environment type. We can opt to use an existing EC2 instance or create a new one.**



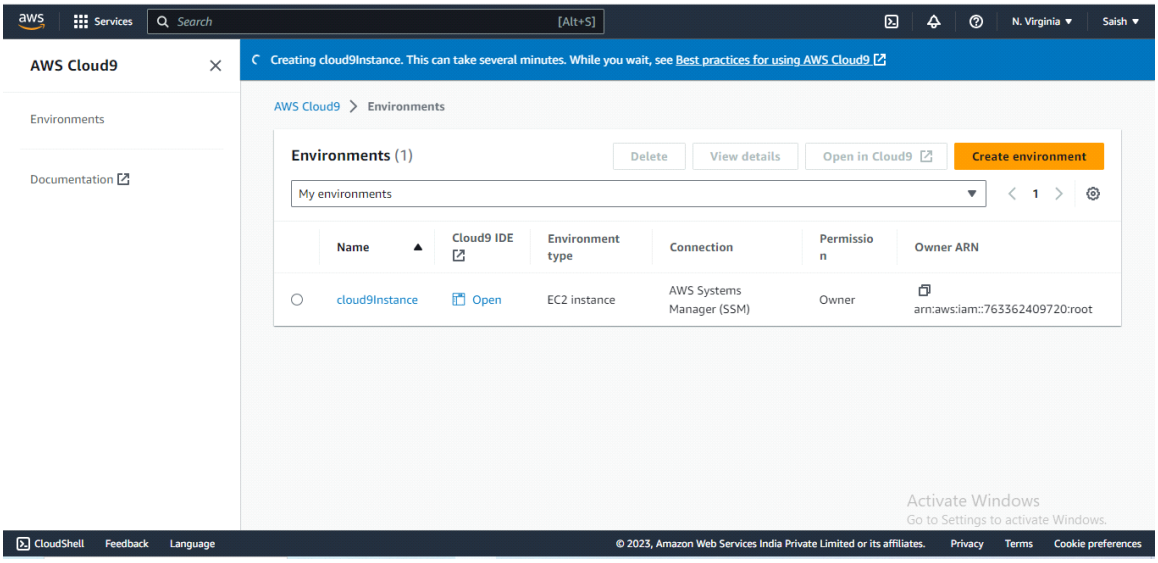
3. **Select instance type as t2.micro which comes under free tier.**



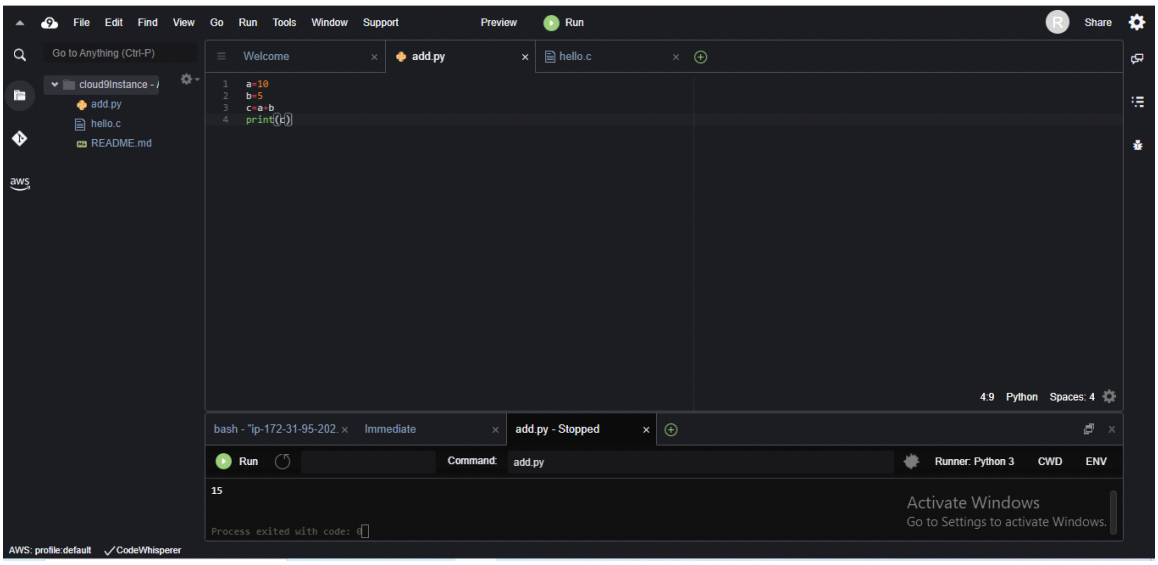
4. **Configure the network settings as per need. We'll proceed with the default settings.**

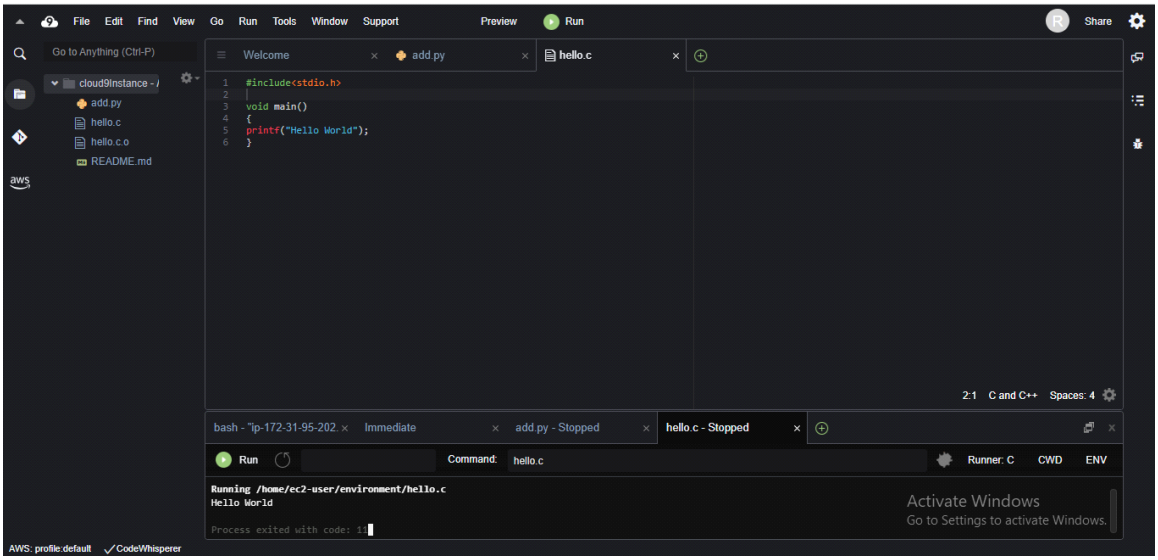


5. **After Cloud9 environment has been created Open it from the Cloud9 Dashboard.**

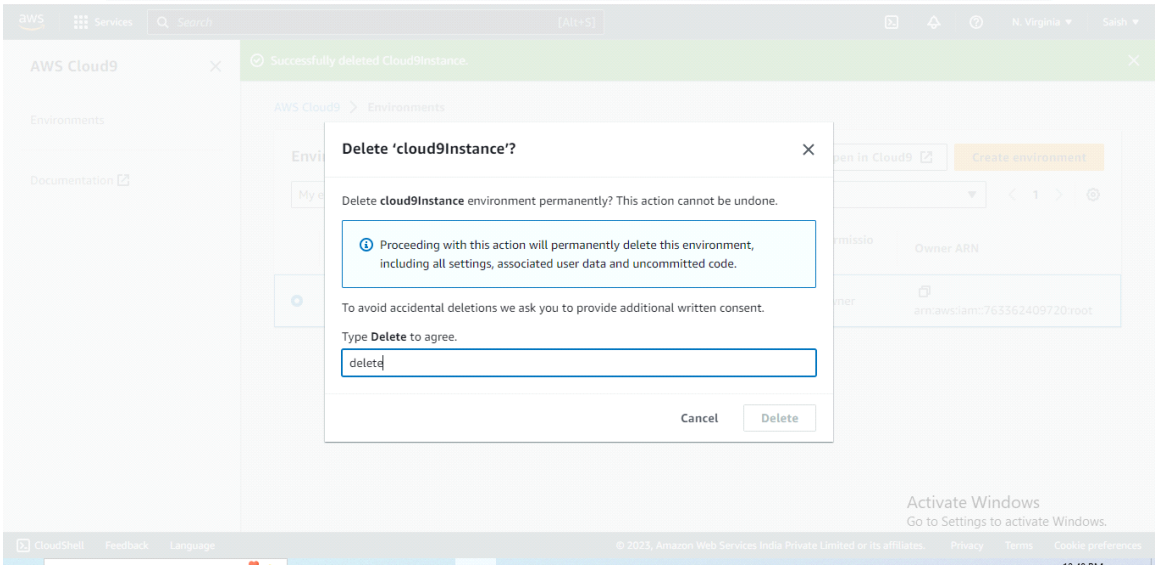


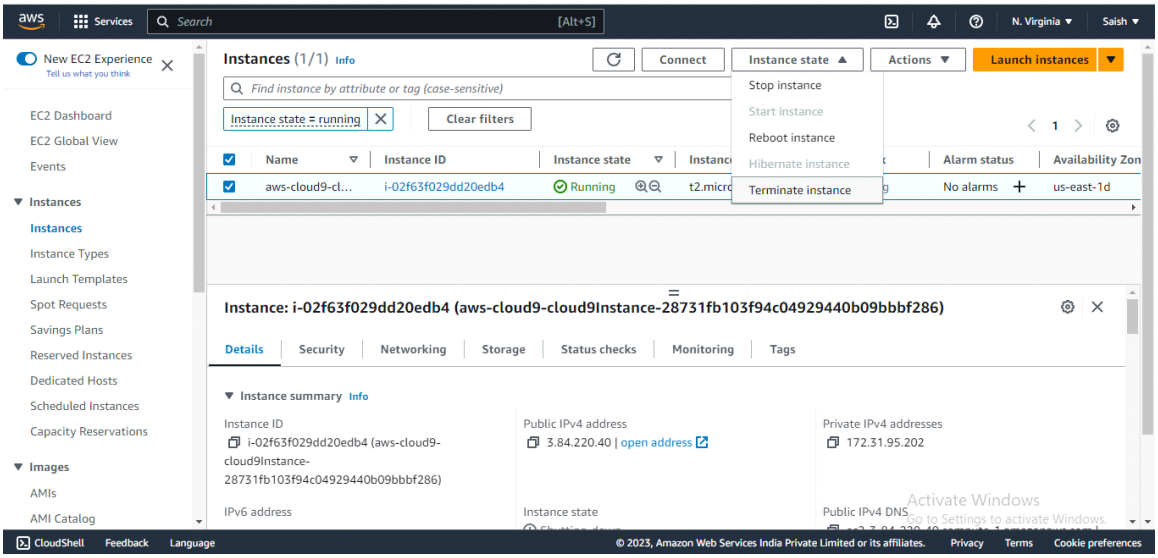
6. **Cloud9 Environment acts as a virtual compiler. Using the Cloud9 Environment we have executed some basic programs in Python and C.**





7. **Delete the Cloud9 Environment. Also ensure to terminate the EC2 instance after use.**





LAB OUTCOME:

L01-To understand the benefits of Cloud9 infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Roll No:10

Saish Bavalekar

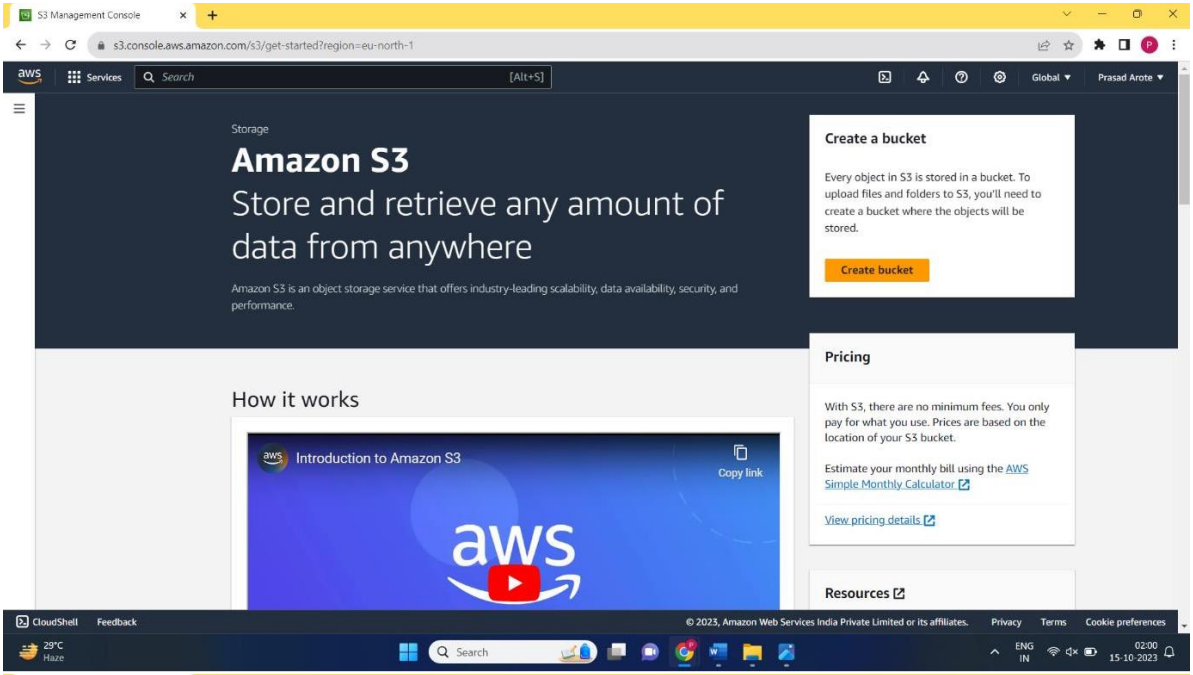
ASSIGNMENT 3

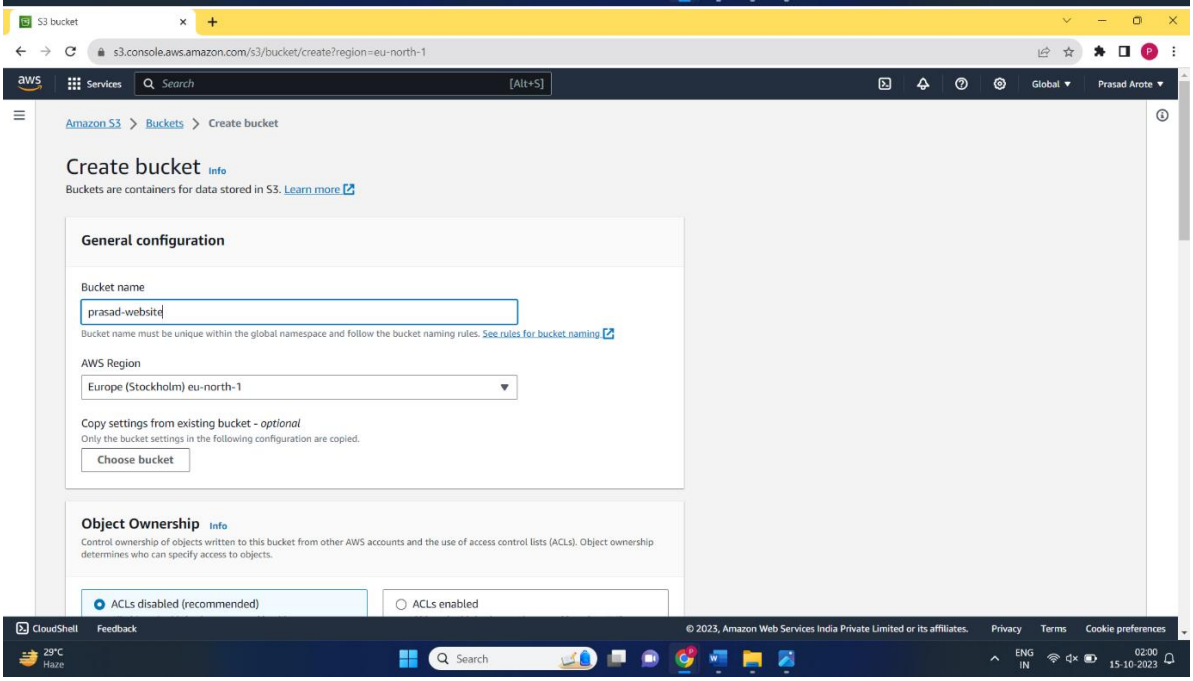
AIM: To study AWS S3 service and create a bucket for hosting static web application.

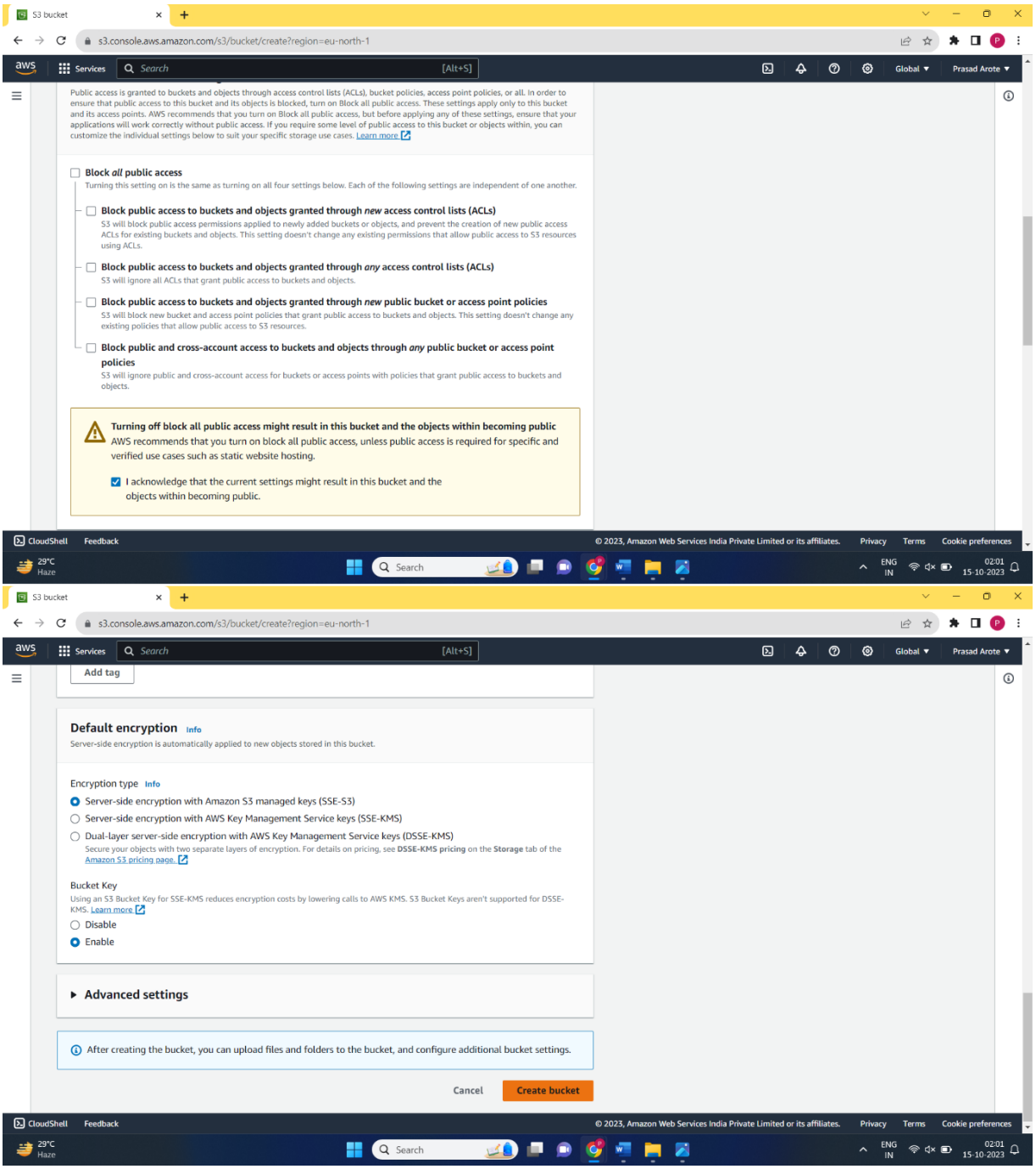
LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.

THEORY:

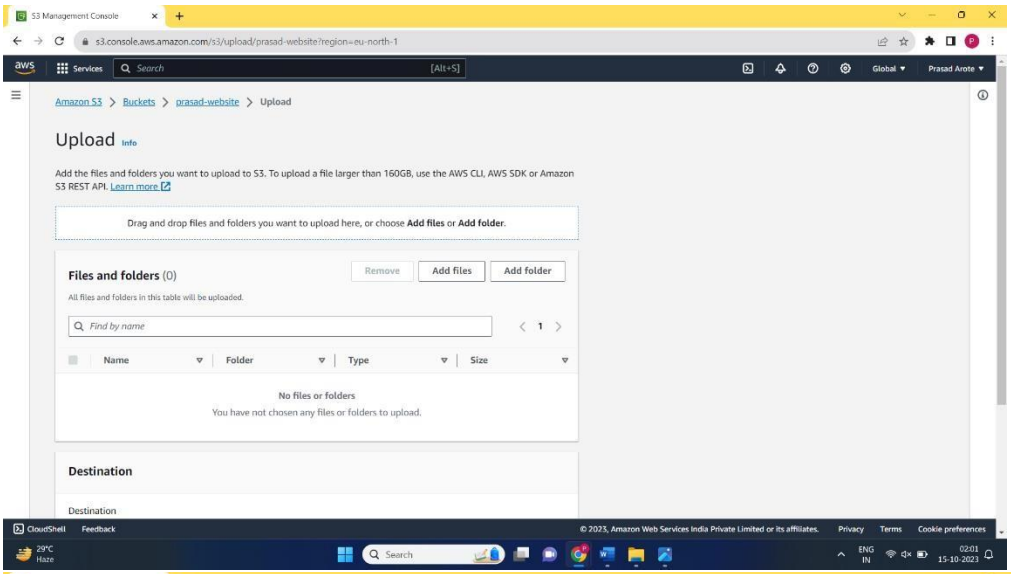
1. Create a S3 bucket.

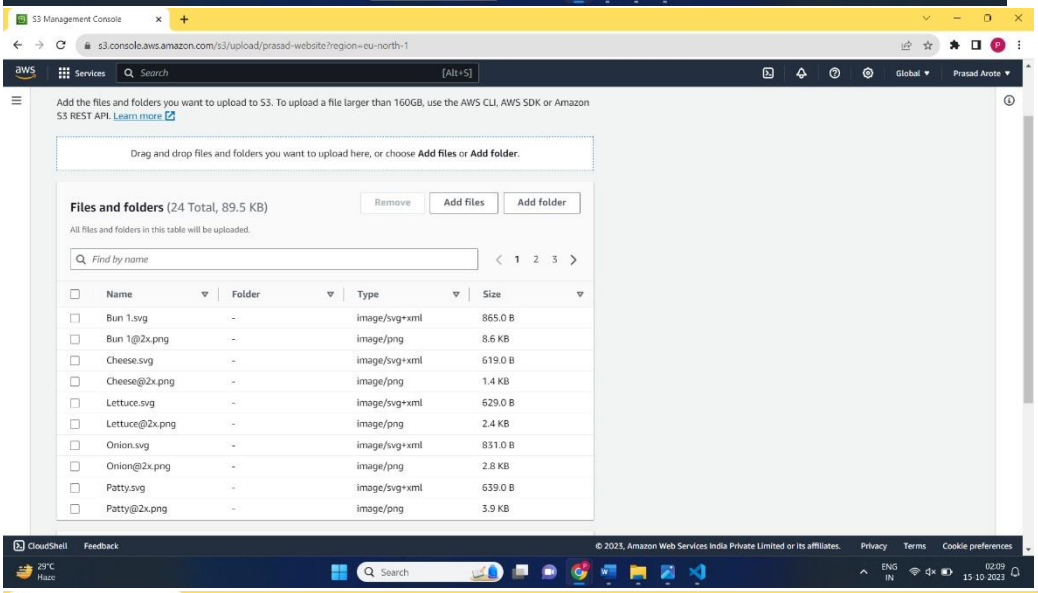


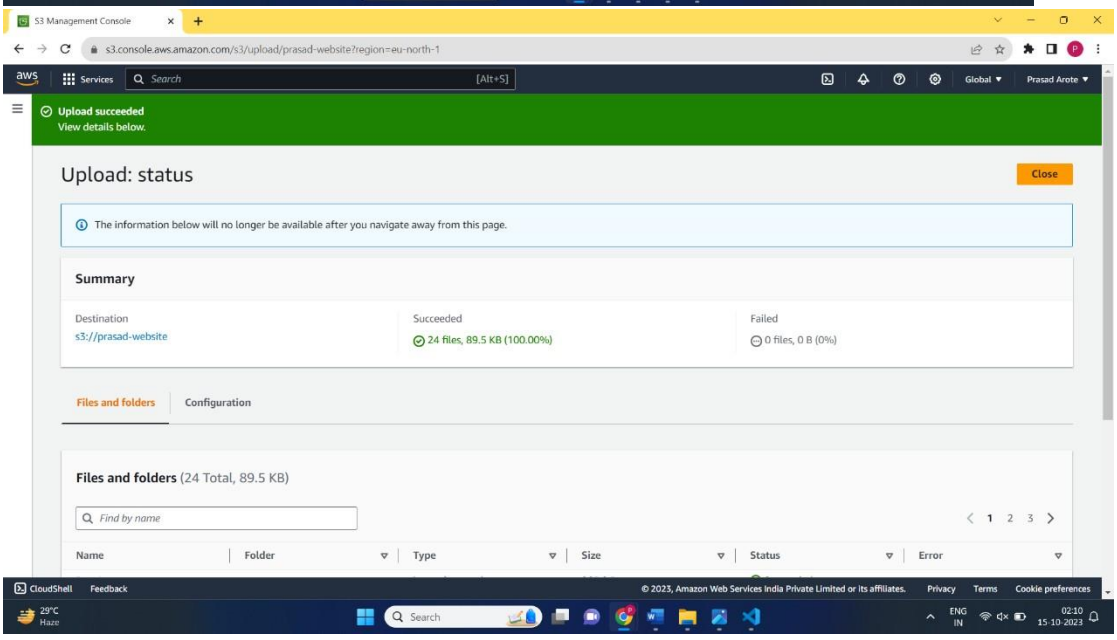




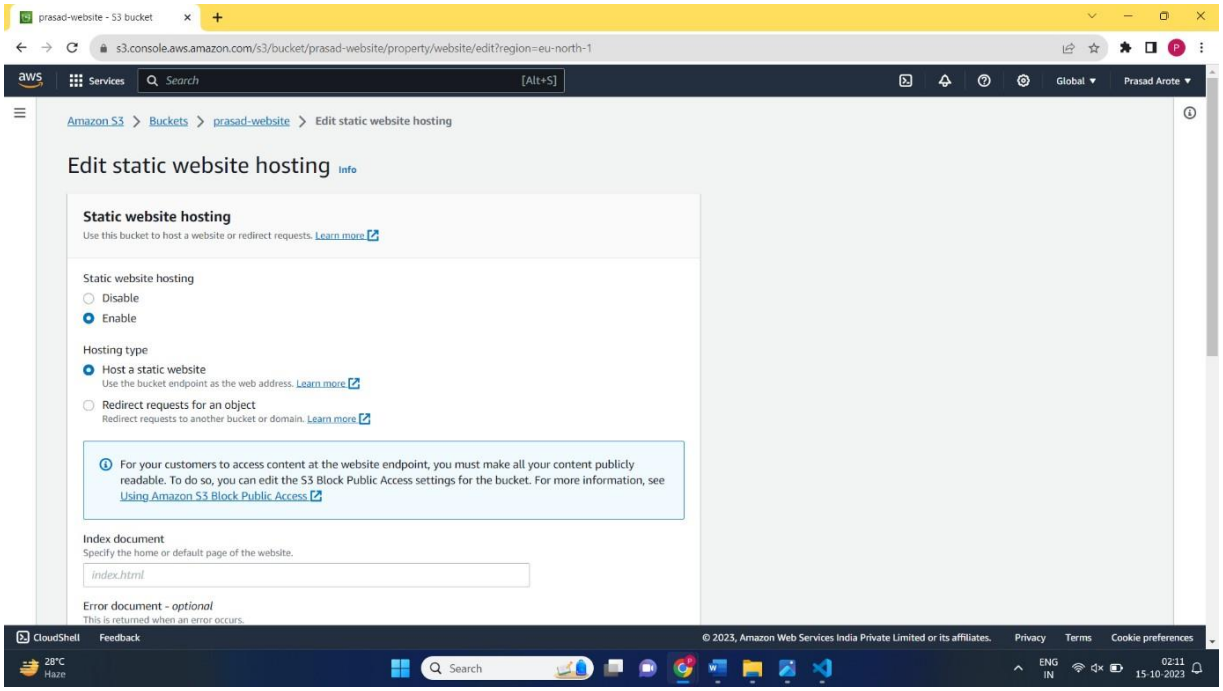
2. Upload the files of web application.

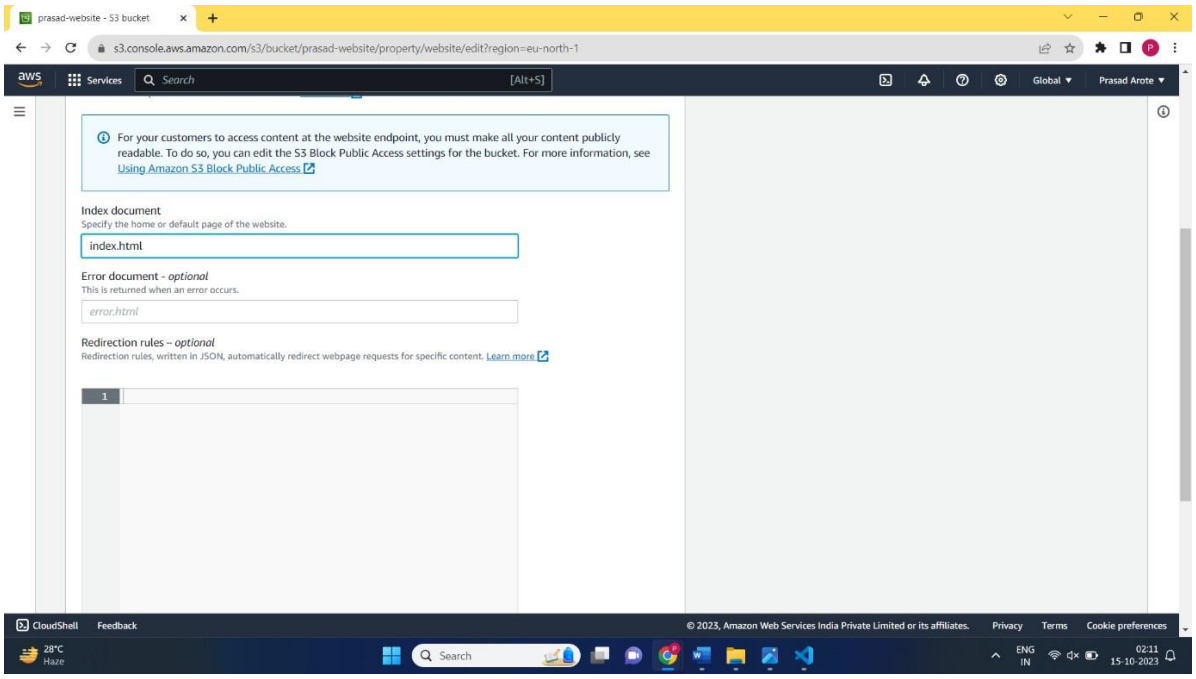

S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
Services
Search
[Alt+S]
Global
Prasad Arote
Amazon S3 > Buckets > prasad-website > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS 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
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

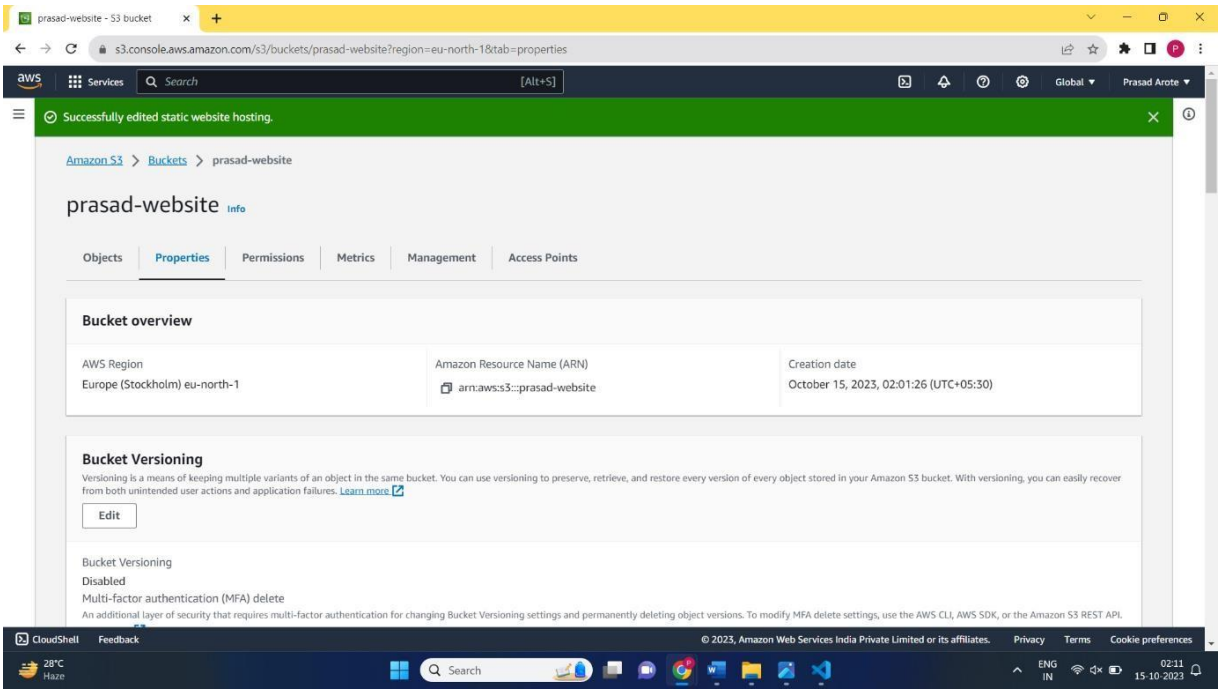

S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (24 Total, 89.5 KB)
All files and folders in this table will be uploaded.
Name
Folder
Type
Size
Bun 1.svg - image/svg+xml 865.0 B
Bun 1@2x.png - image/png 8.6 KB
Cheese.svg - image/svg+xml 619.0 B
Cheese@2x.png - image/png 1.4 KB
Lettuce.svg - image/svg+xml 629.0 B
Lettuce@2x.png - image/png 2.4 KB
Onion.svg - image/svg+xml 831.0 B
Onion@2x.png - image/png 2.8 KB
Patty.svg - image/svg+xml 639.0 B
Patty@2x.png - image/png 3.9 KB


S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
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
Files and folders (24 Total, 89.5 KB)
Find by name
Name
Folder
Type
Size
Status
Error
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

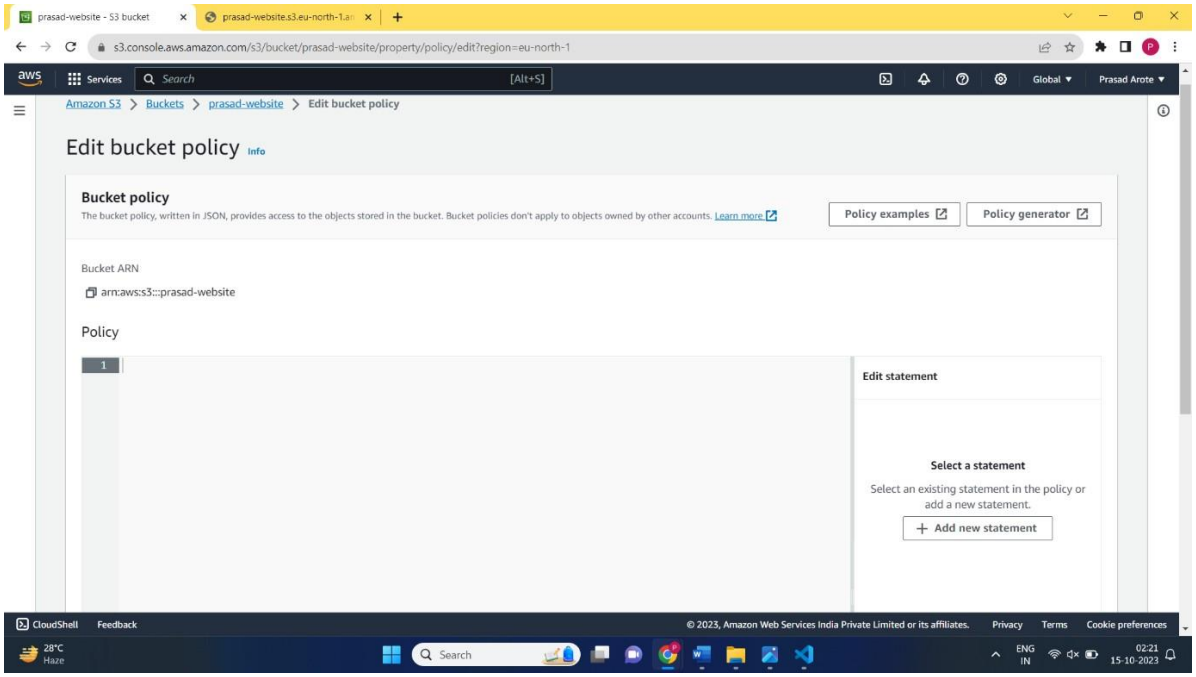
3. Enable Static website hosting

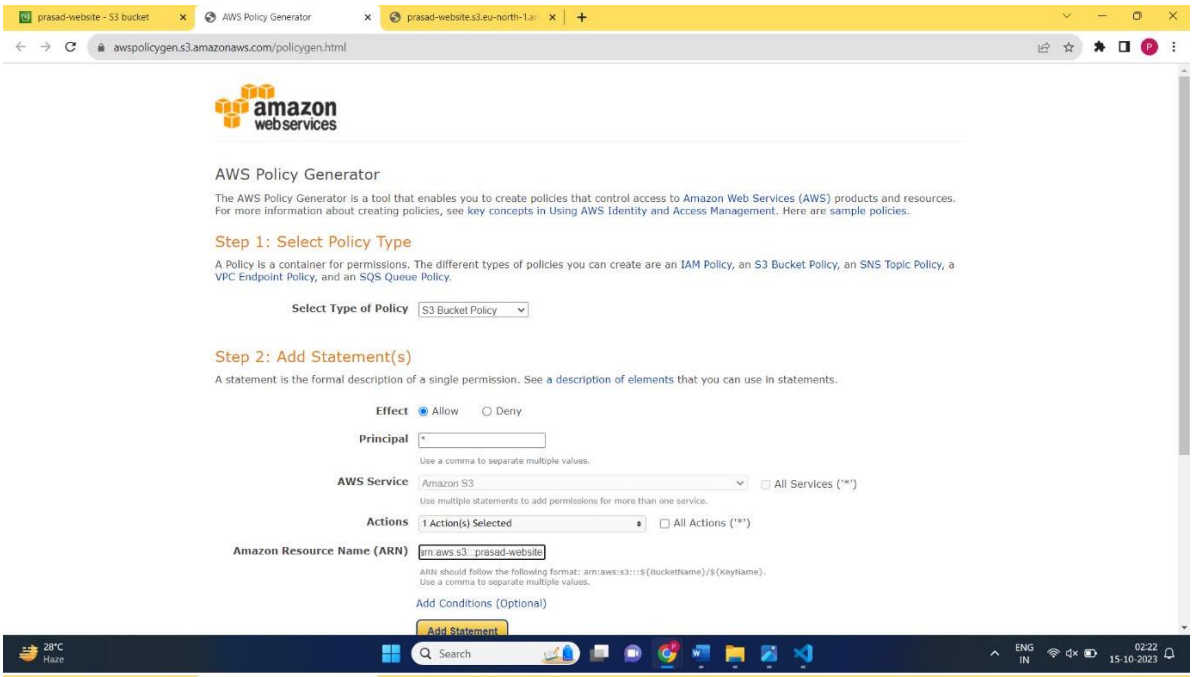


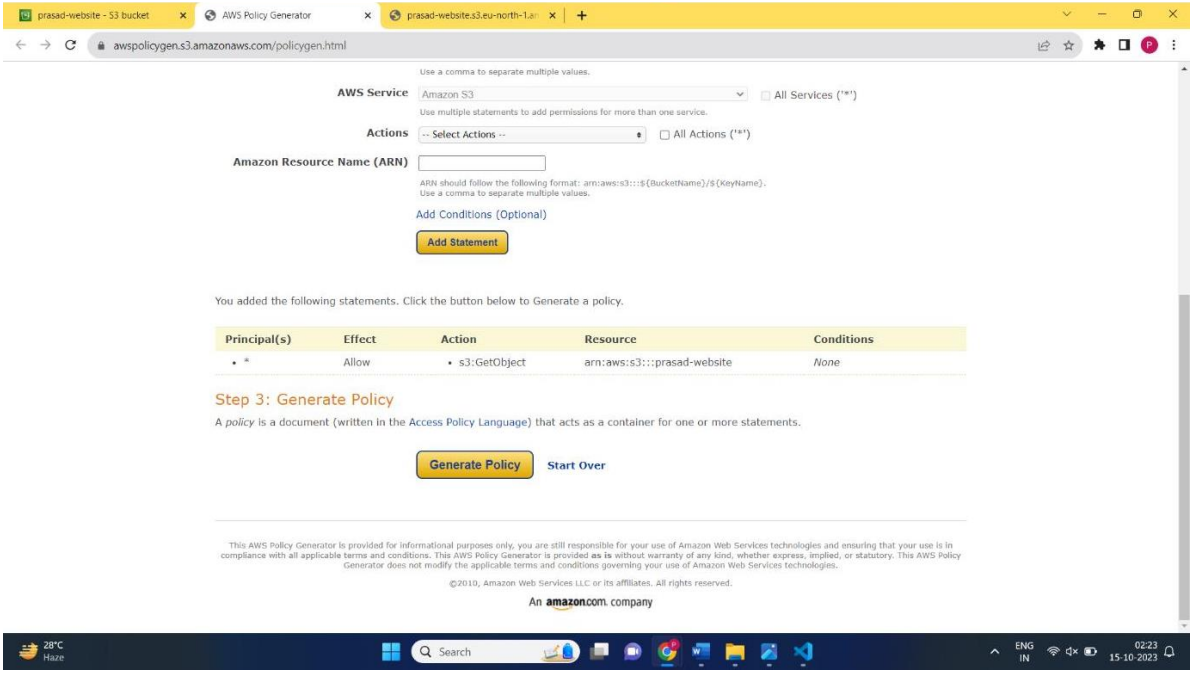


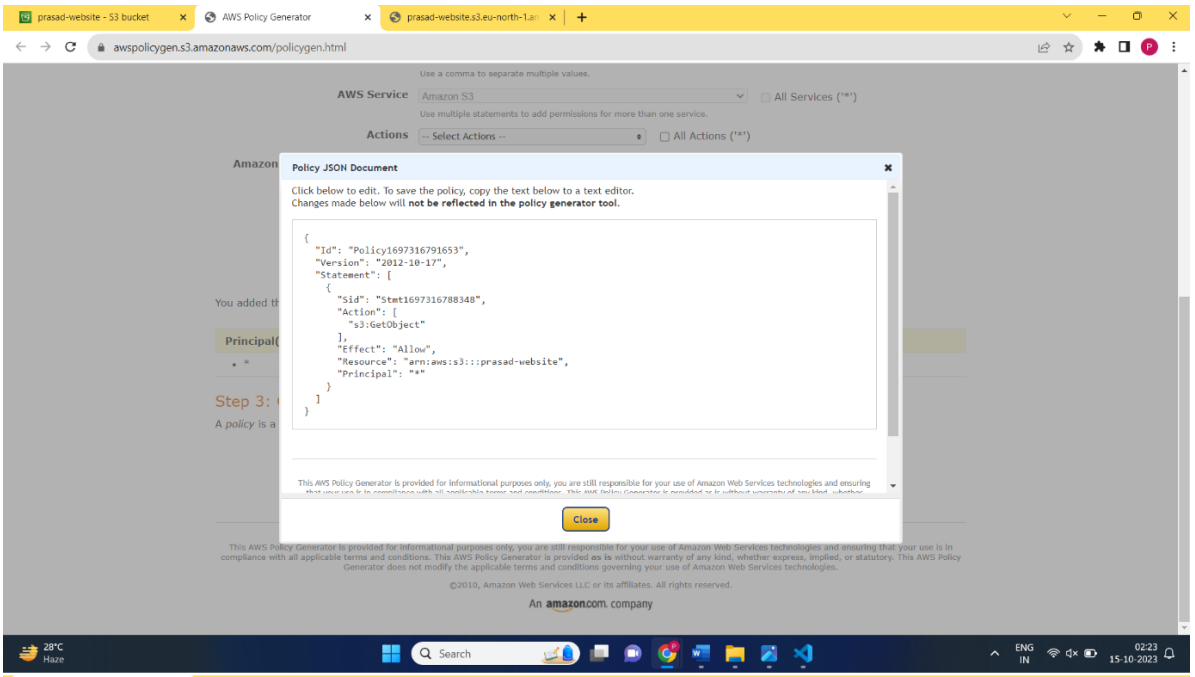


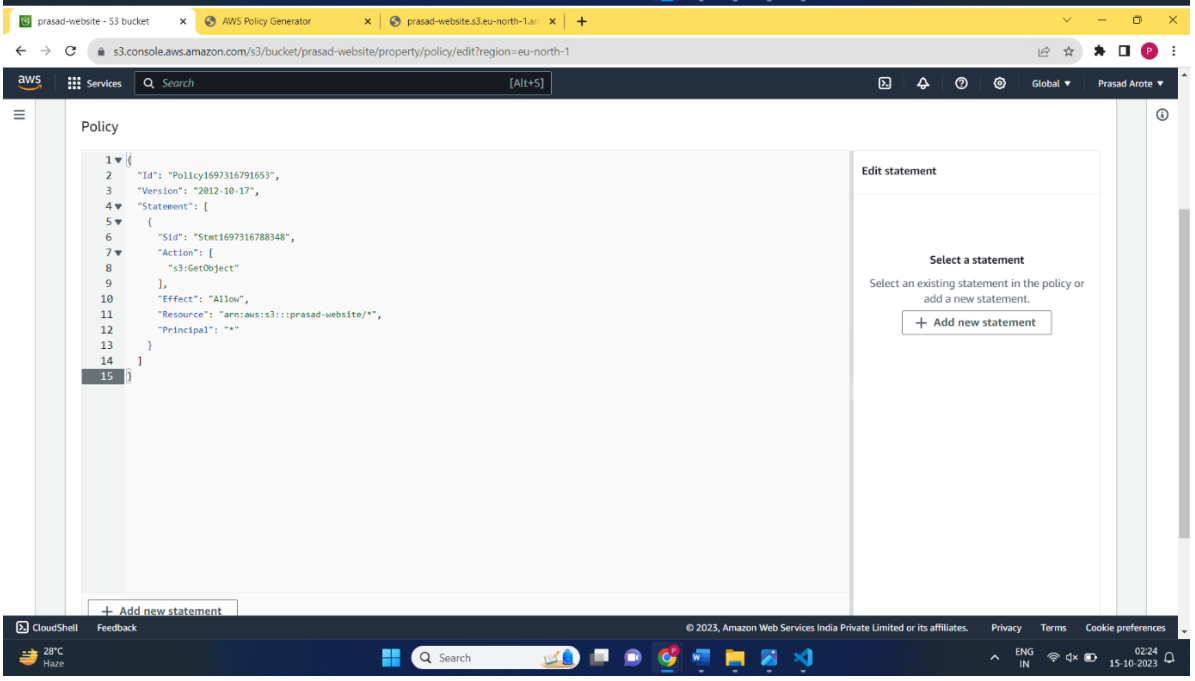
5. Change the Bucket Policy

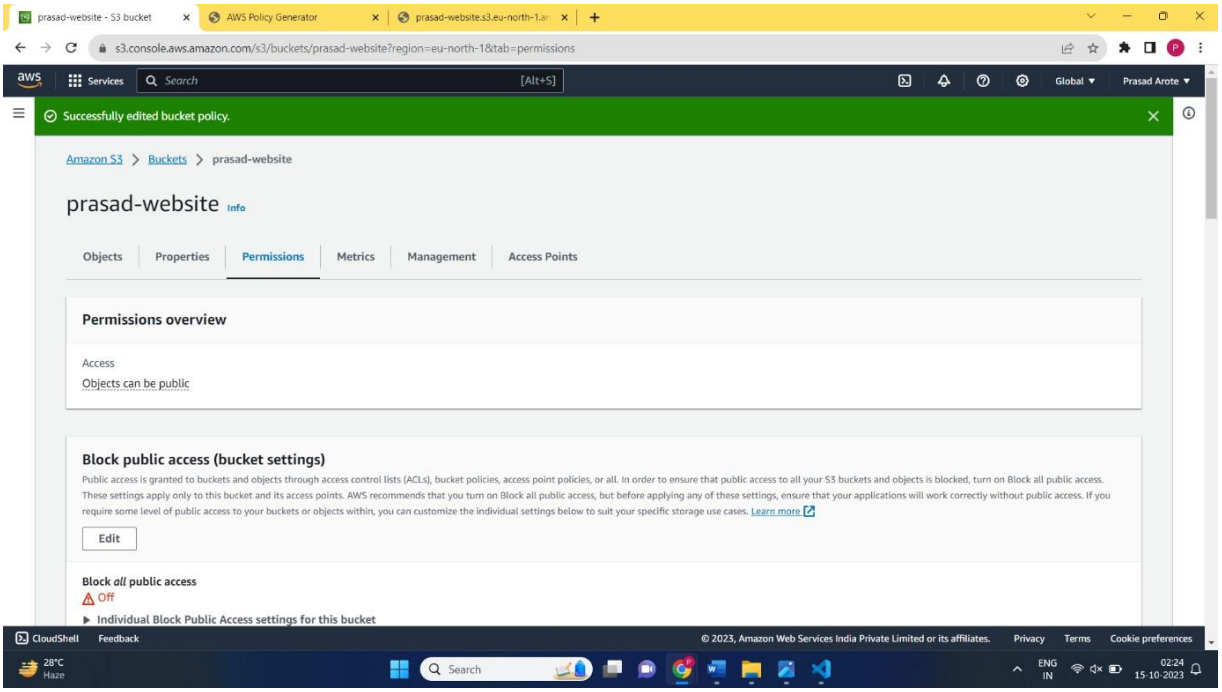



prasad-website - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
AWS Policy Generator
The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, 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
28°C
Haze
Search
ENG
IN
02:22
15-10-2023

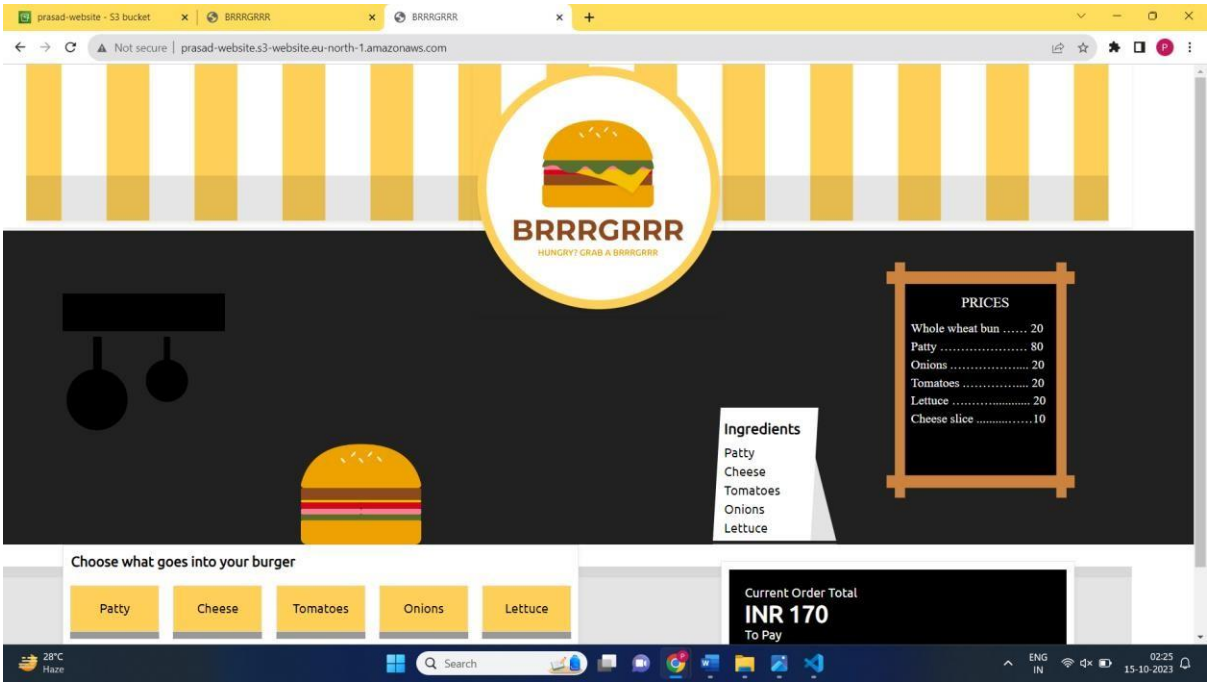

prasad-website - S3 bucket
AWS Policy Generator
awspolicygen.s3.amazonaws.com/policygen.html
Use a comma to separate multiple values.
AWS Service
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
-- Select Actions --
All Actions ('*')
Amazon Resource 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
28°C
Haze
Search
ENG
IN
02:23
15-10-2023


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 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
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.

Roll No:10

Saish Bavalekar

ASSIGNMENT 4

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:

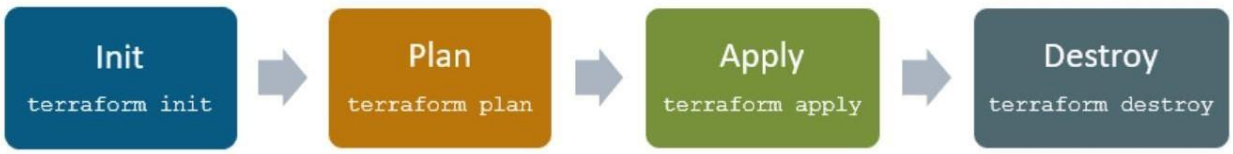
Terraform is one of the most popular Infrastructure-as-code (IaC) tool, used by DevOps teams to automate infrastructure tasks. It is used to automate the provisioning of your cloud resources. Terraform is an open-source, cloud-agnostic provisioning tool developed by HashiCorp and written in GO language.

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.

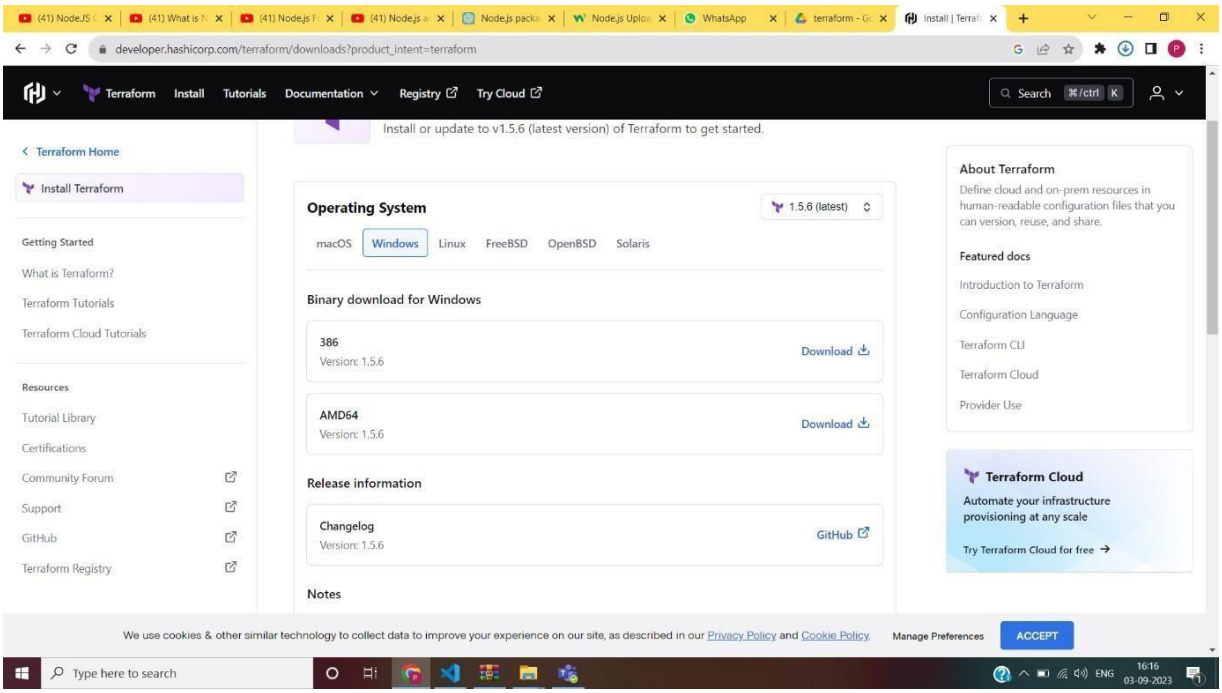
INSTALLATION:

1) Download Terraform

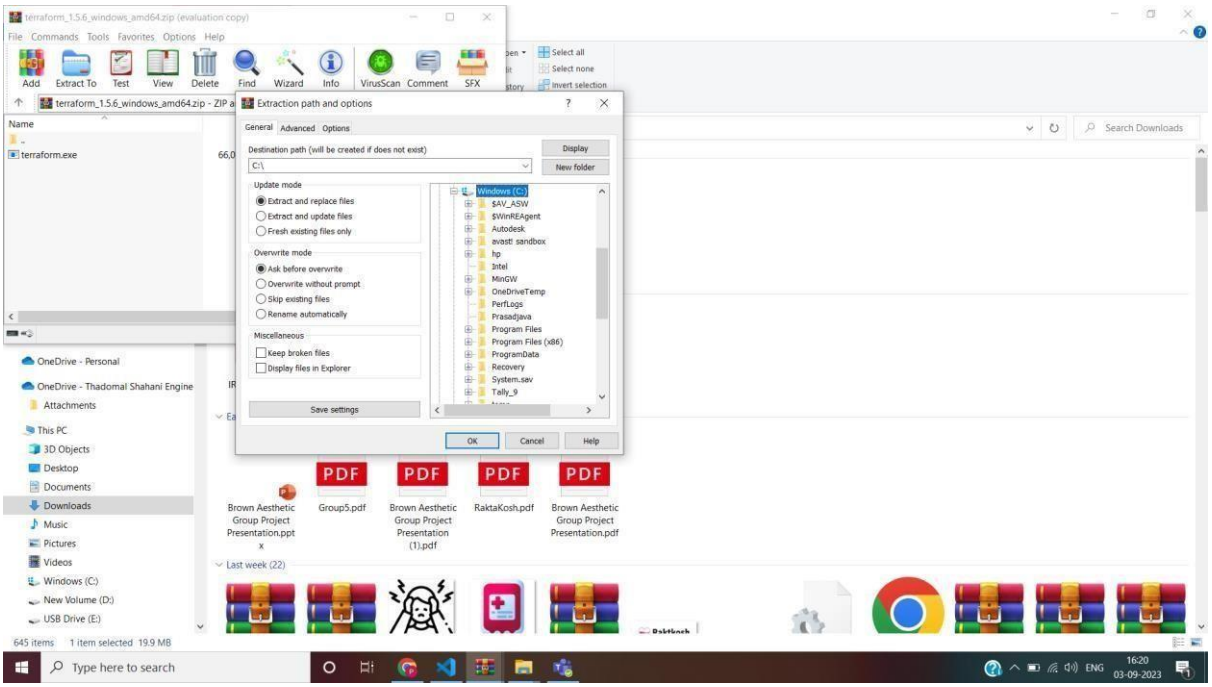
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

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

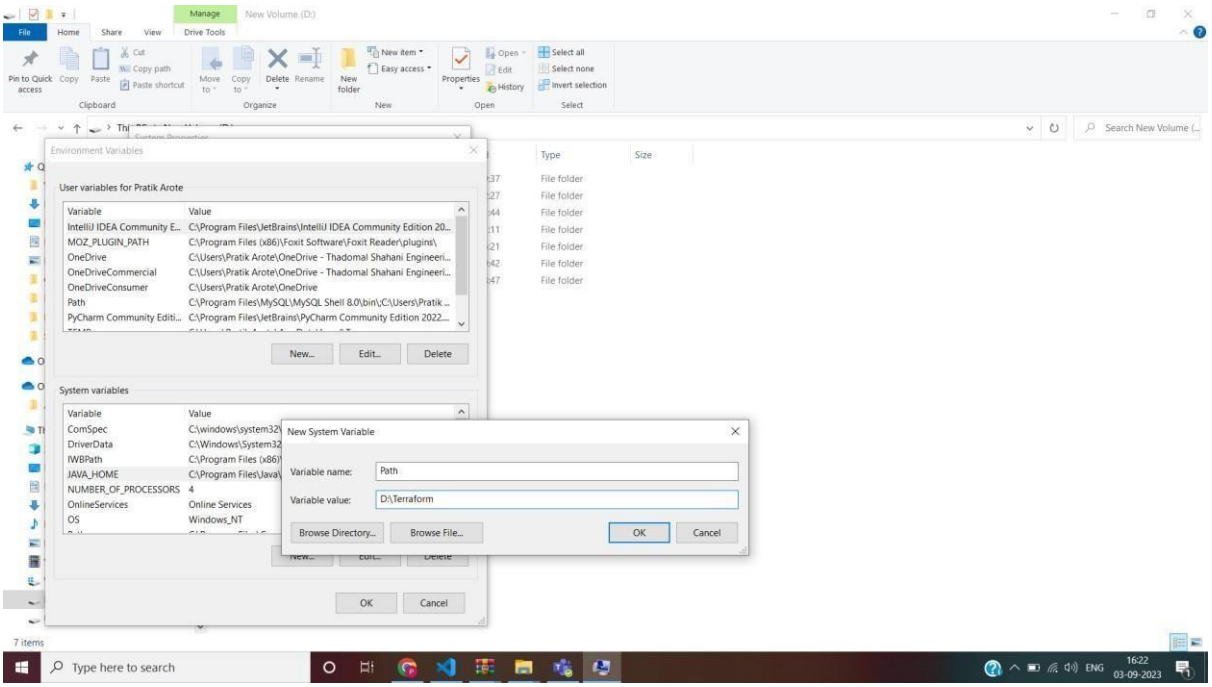
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.



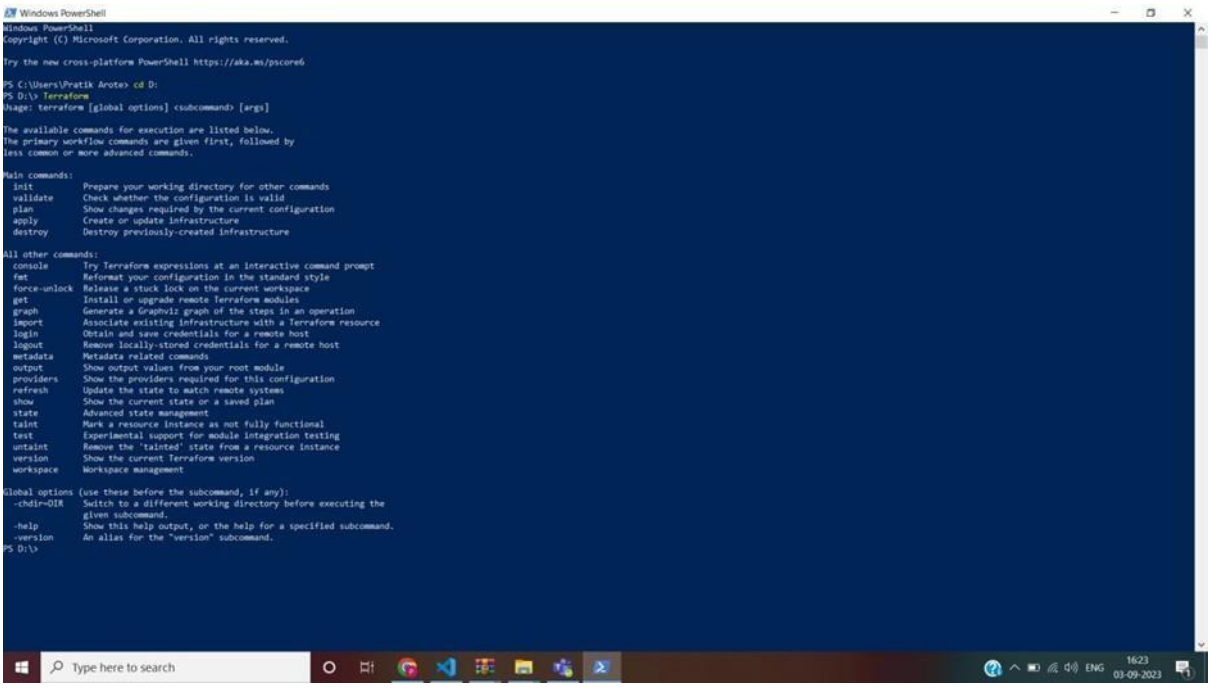
2. Extract the downloaded setup file Terraform.exe in C:\Terraform Directory



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.

Roll No:10

Saish Bavalekar

ASSIGNMENT 5

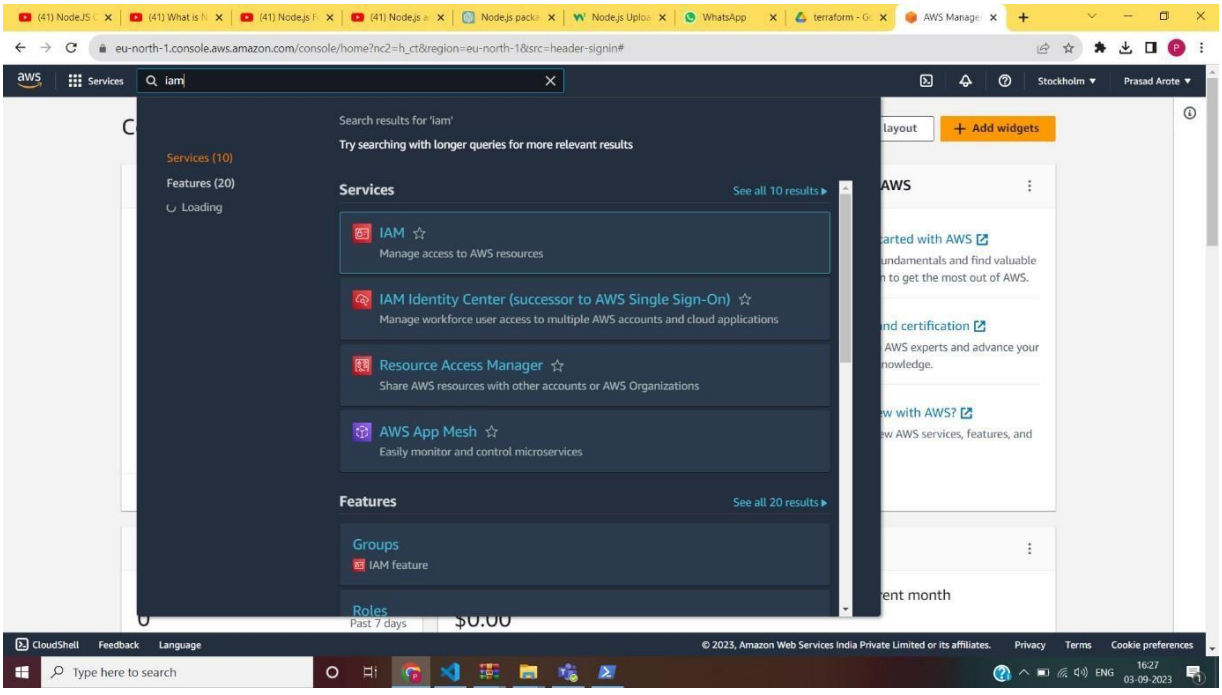
AIM: To Build, change, and destroy AWS infrastructure Using Terraform.

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

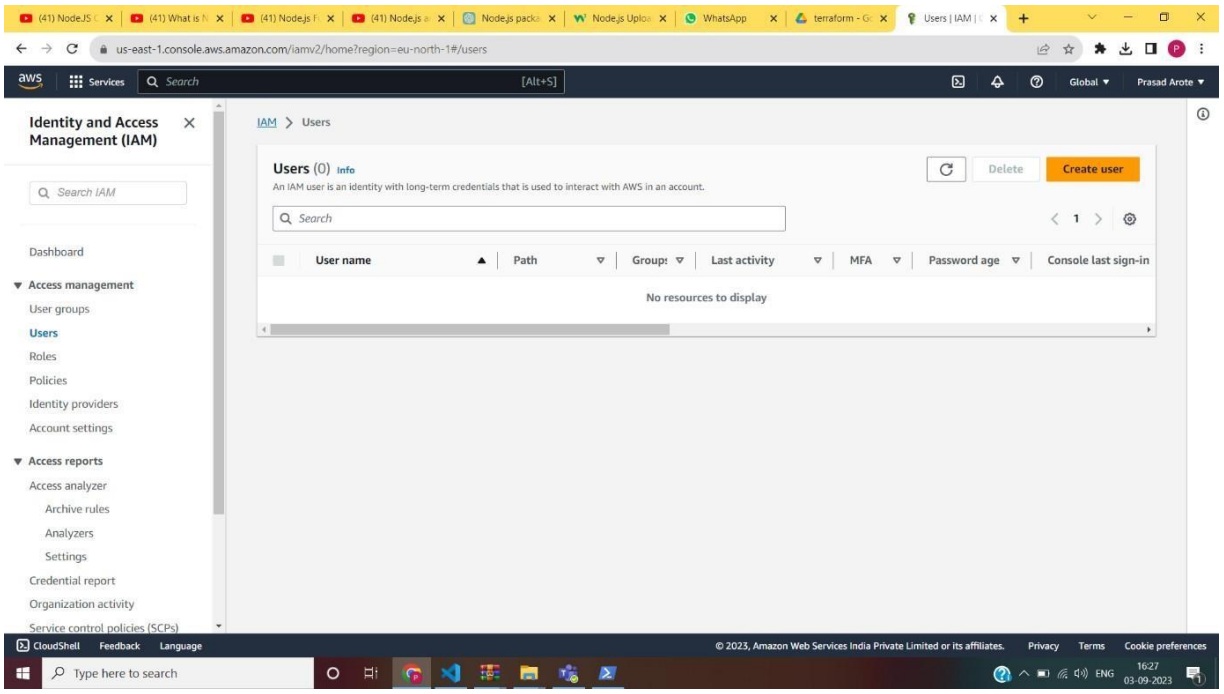
LO5: To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

Theory:

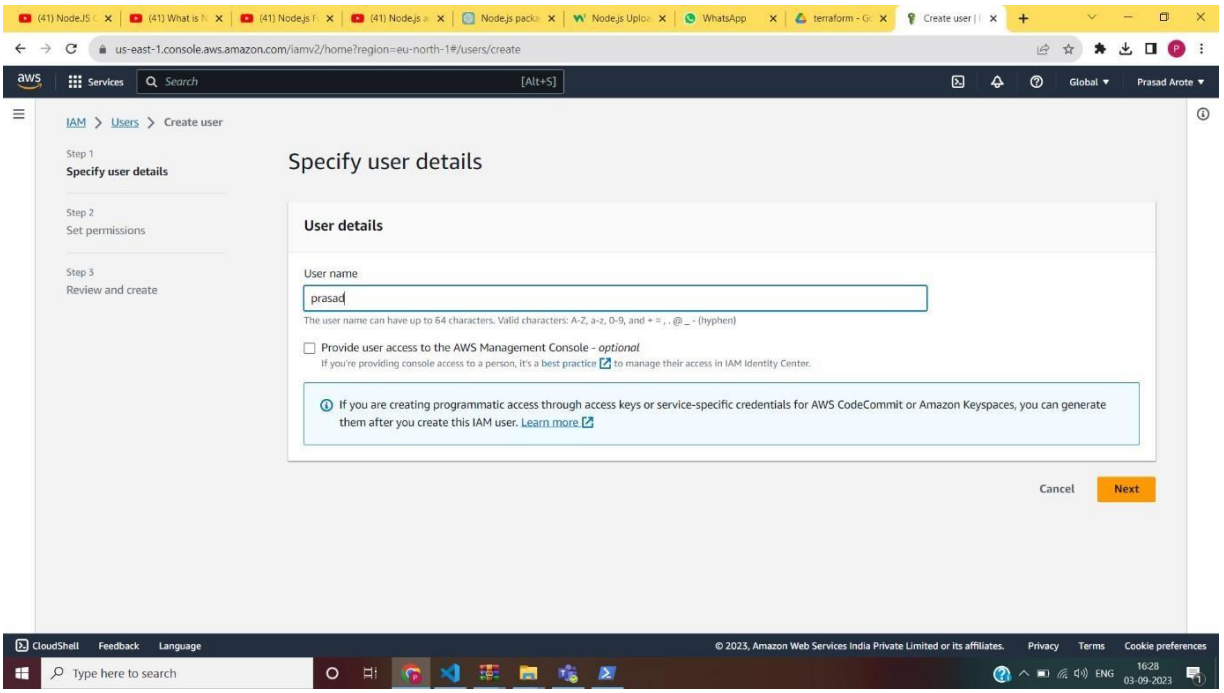
1) Make dir Terraform Scripts Open aws.amazon.com
   Login to your account
   Search IAM



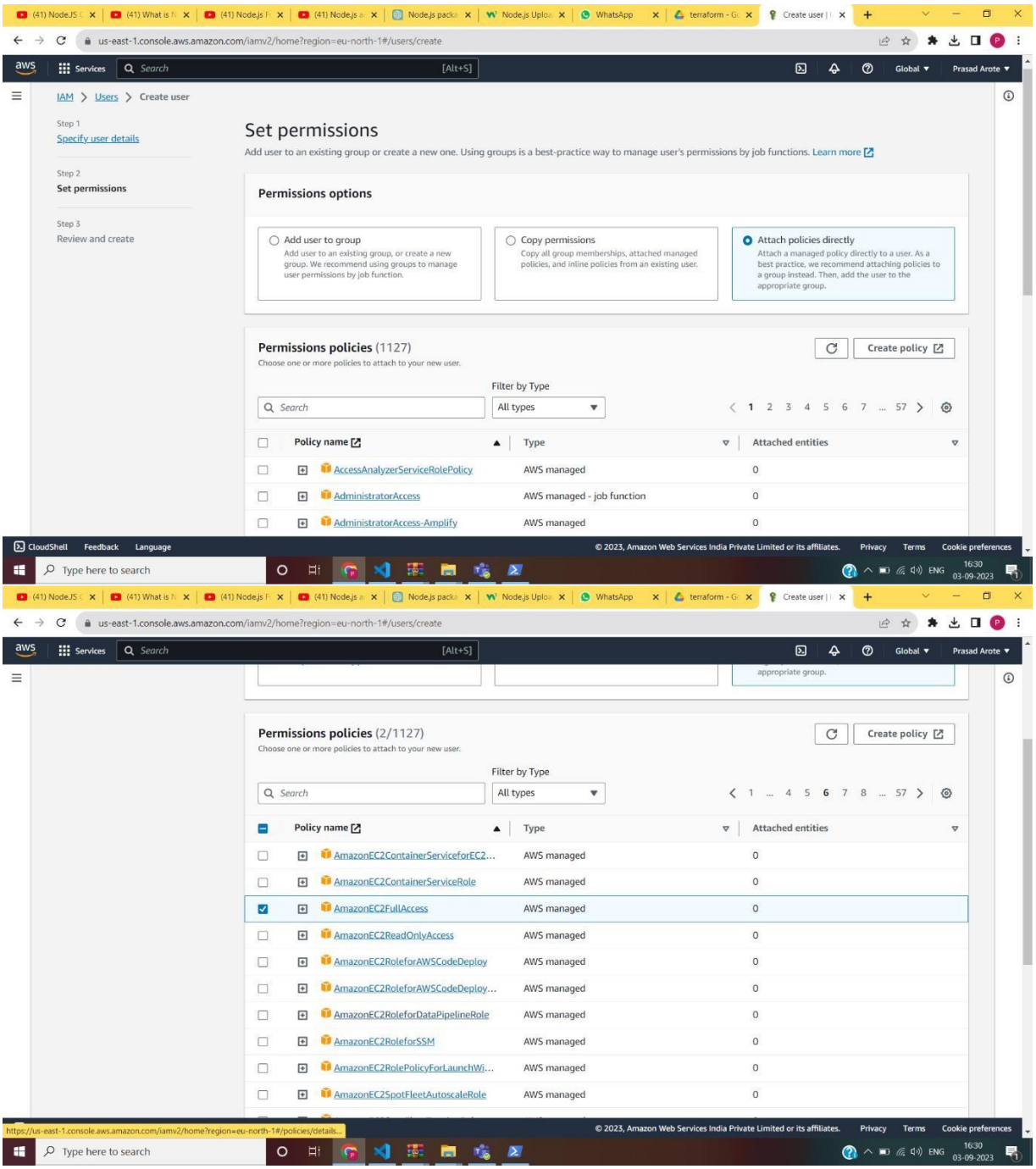
2) Click on Users (on the LHS)



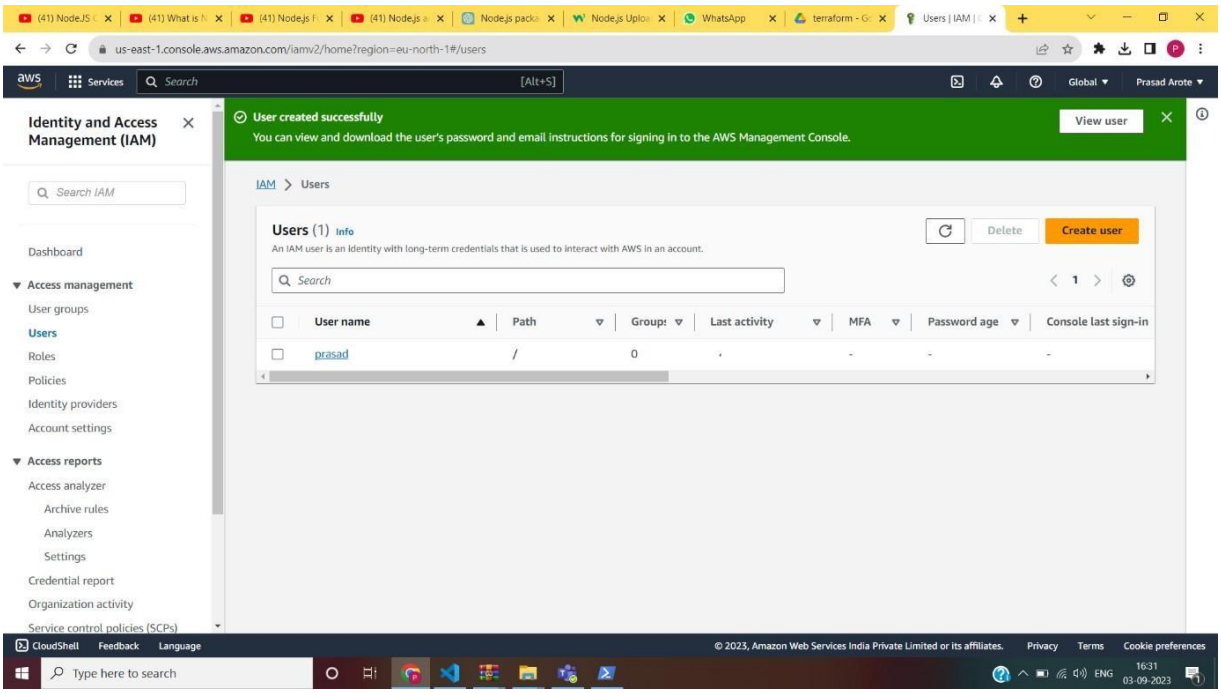
3) Click Add users



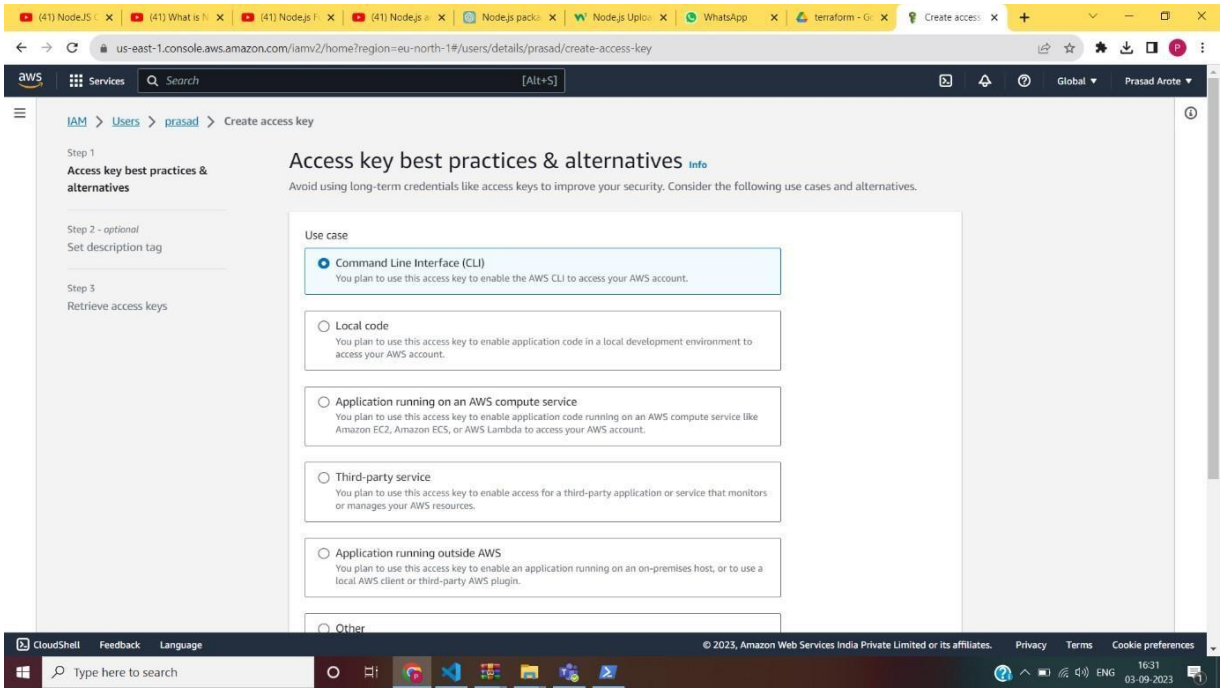
4) Set Permissions -> AmazonEC2FullAccess



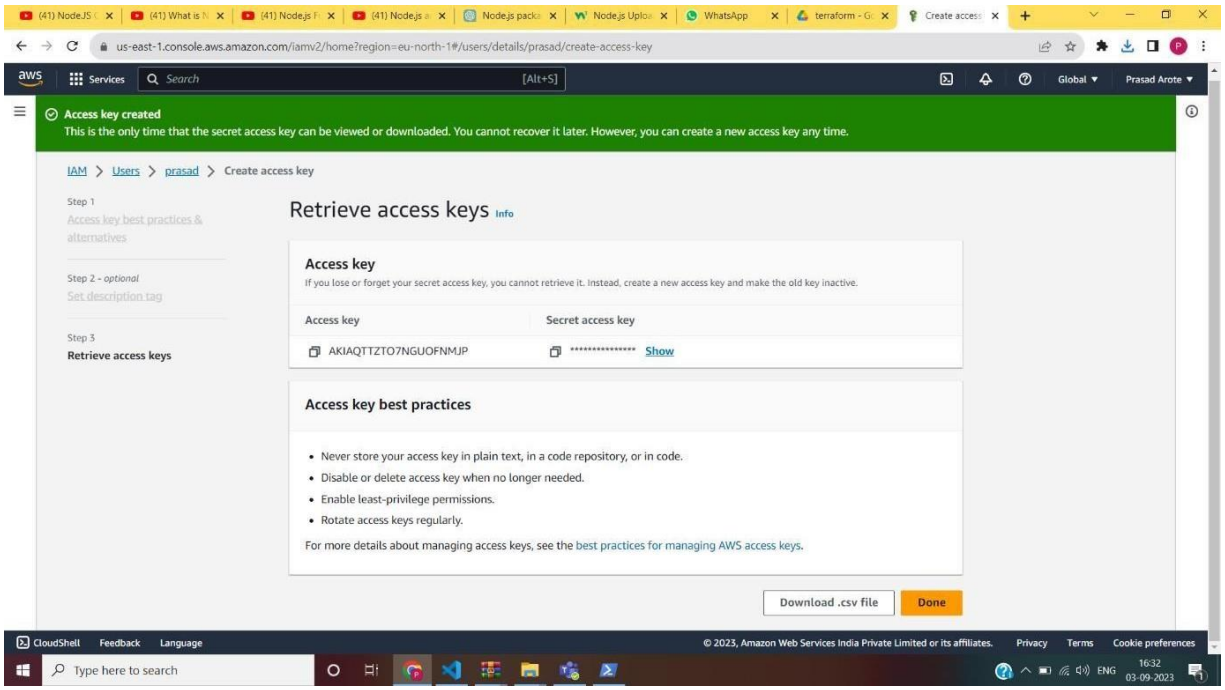
5) Create User



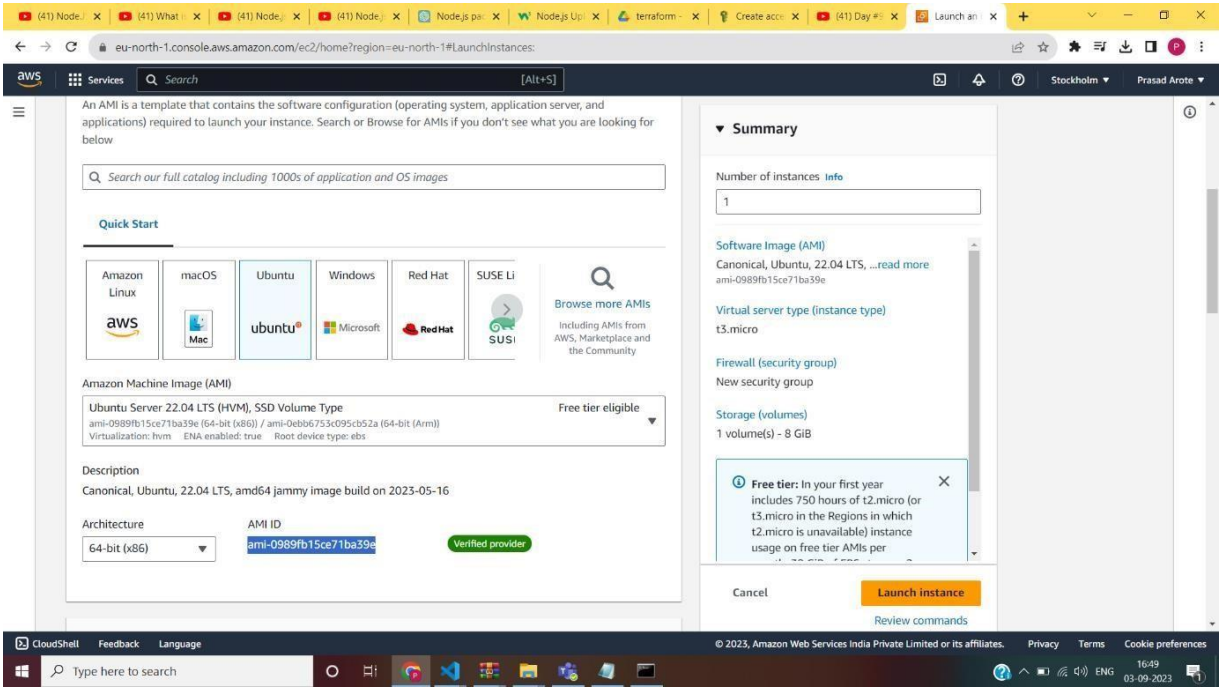
6) Create access key



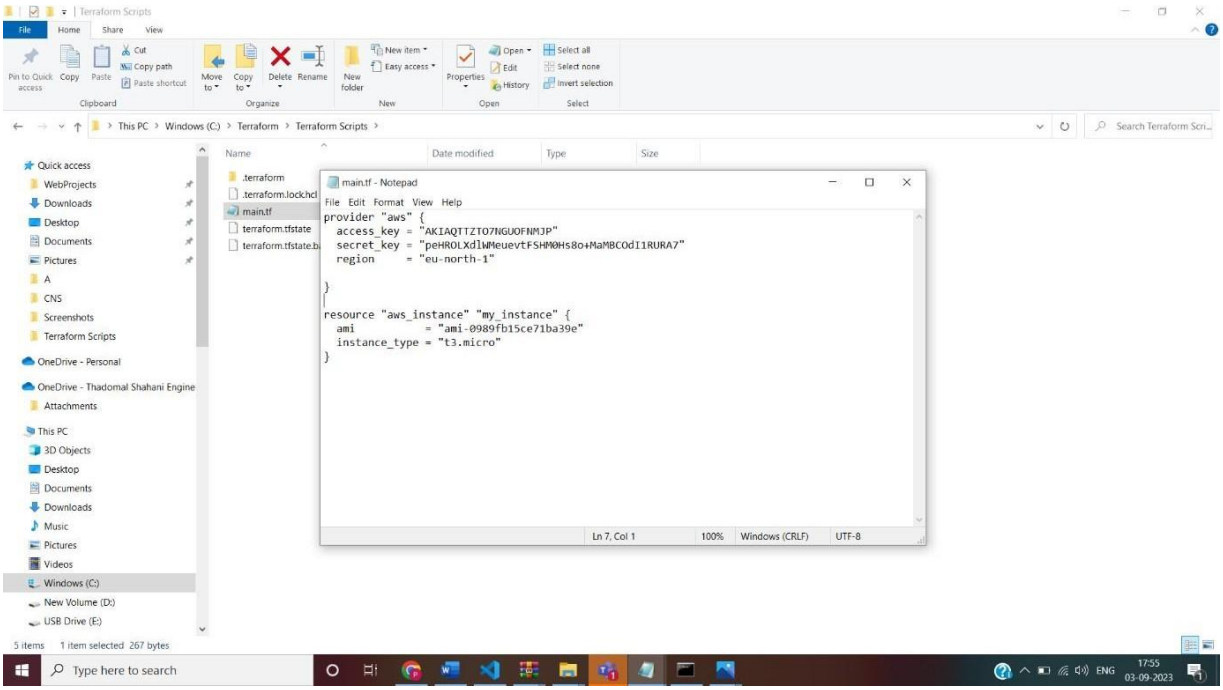
7) Download the .csv file



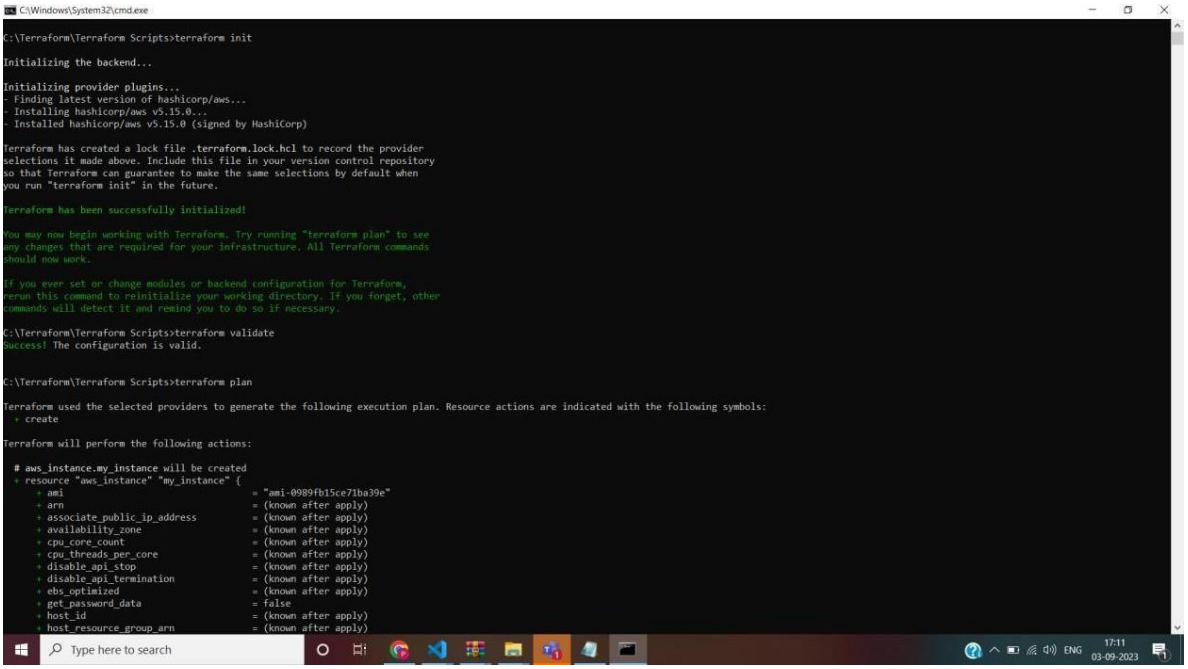
8) Open EC2 Instances and Copy the AMI ID of any one of them.

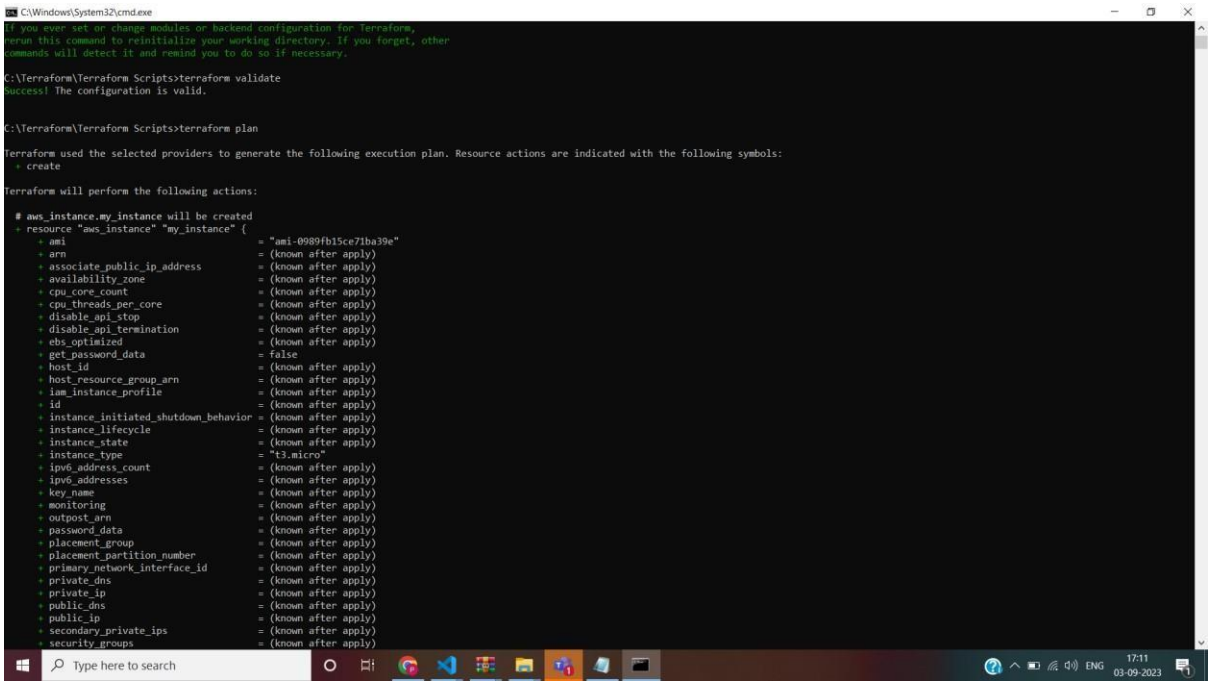


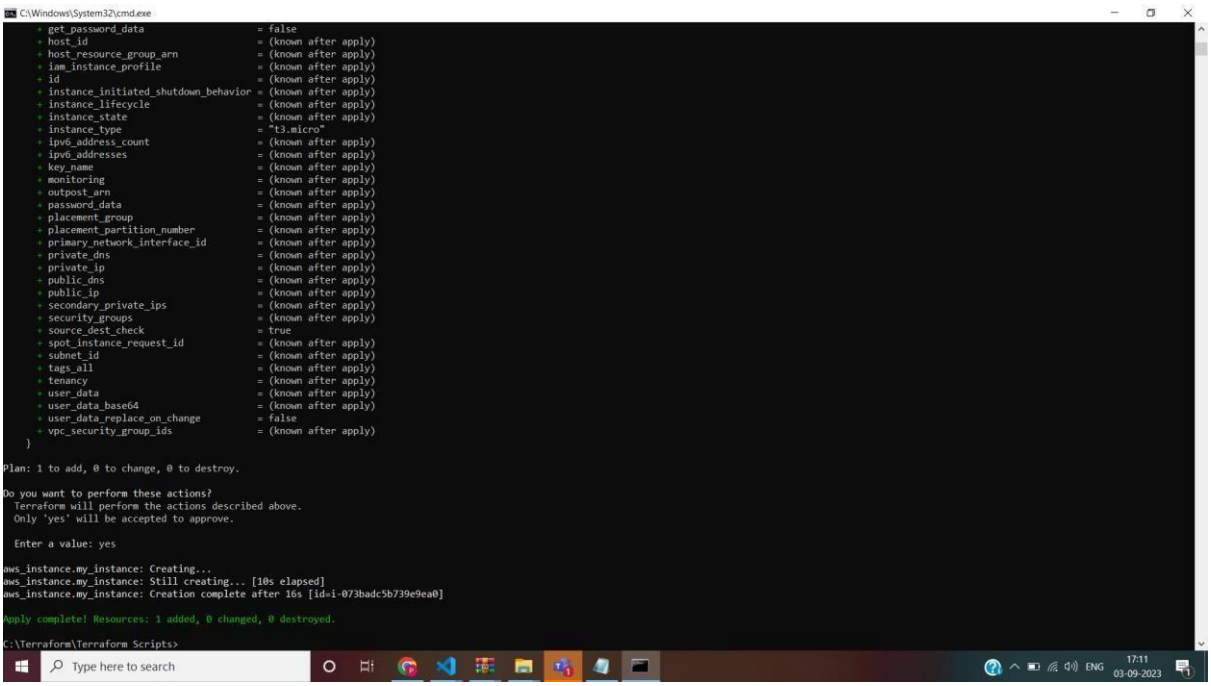
9) Configure the main.tf file

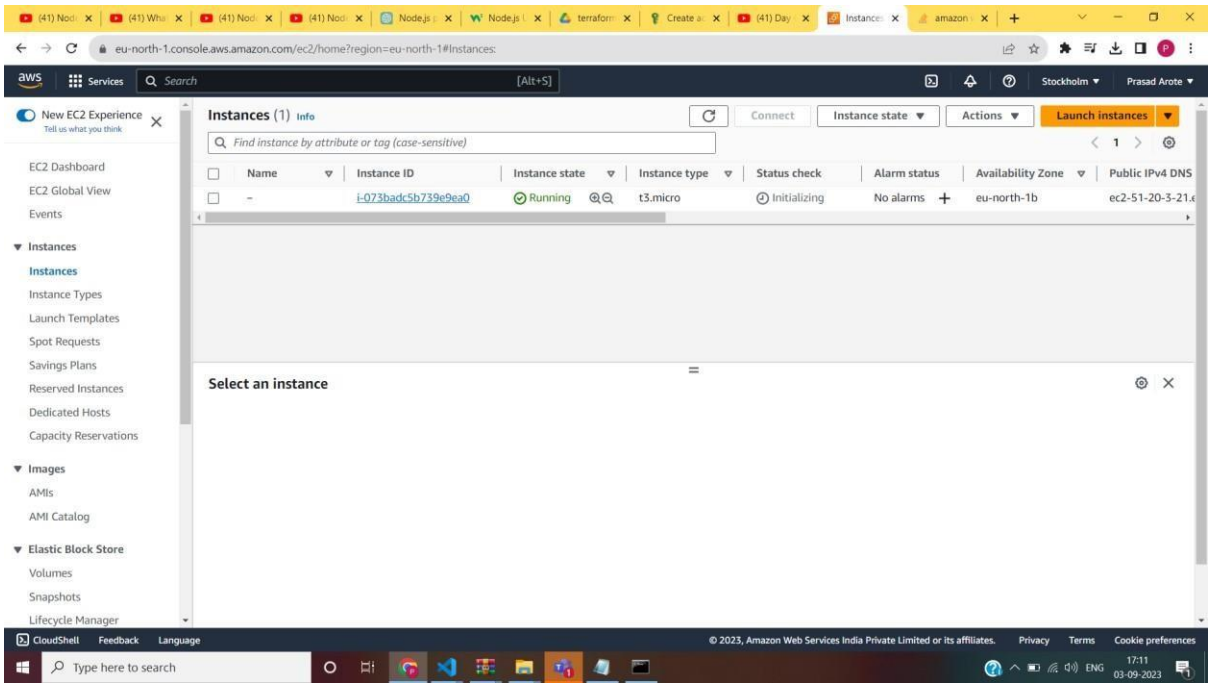


10) Run the commands in cmd -> terraform init , terraform validate , terraform plan, terraform apply to create EC2 instance using terraform.

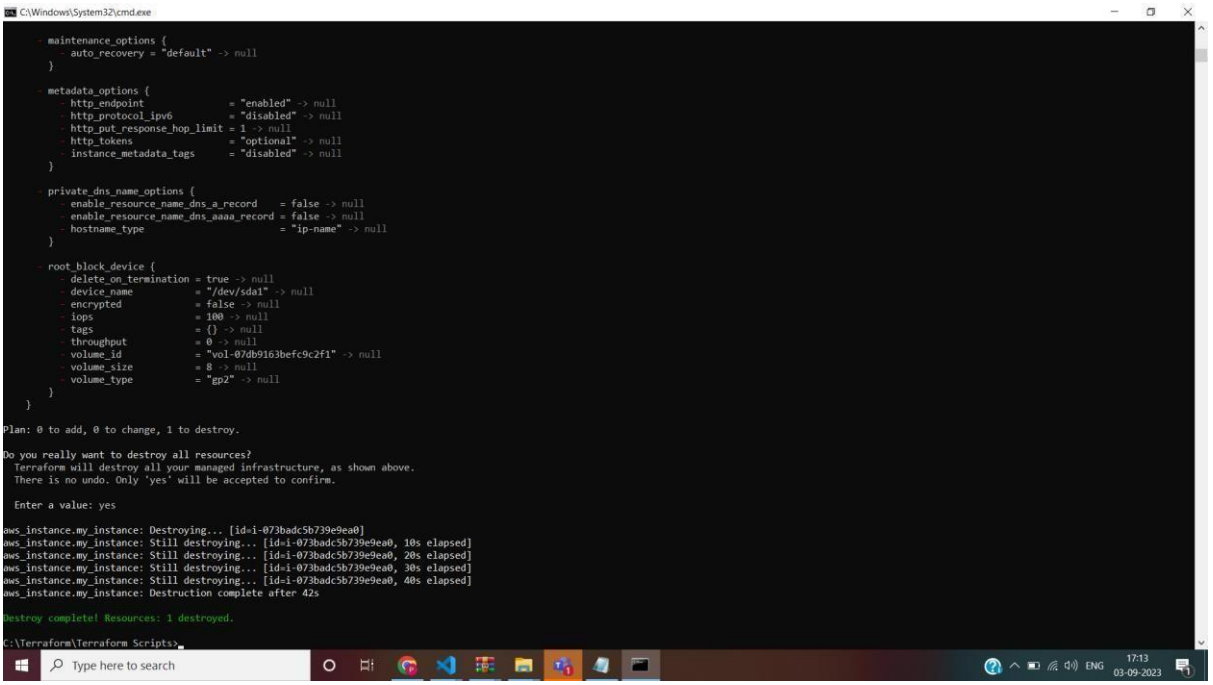



C:\Windows\System32\cmd.exe
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
C:\Terraform\Terraform Scripts>terraform validate
Success! The configuration is valid.
C:\Terraform\Terraform Scripts>terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.my_instance will be created
+ resource "aws_instance" "my_instance" {
+ ami = "ami-0989fb15ce71ba39e"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t3.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)
+ private_ip = (known after apply)
+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
Type here to search
ENG
17:11
03-09-2023


C:\Windows\System32\cmd.exe
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t3.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns = (known after apply)
+ private_ip = (known after apply)
+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags_all = (known after apply)
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
Plan: 1 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
aws_instance.my_instance: Creating...
aws_instance.my_instance: Still creating... [10s elapsed]
aws_instance.my_instance: Creation complete after 16s [id=i-073badc5b739e9ea0]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
C:\Terraform\Terraform Scripts>
Type here to search
ENG
17:11
03-09-2023



11) Destroy the instance using terraform destroy.



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

Roll No:10

Saish Bavalekar

ASSIGNMENT 6

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

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

THEORY:

SonarQube:

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

Features:

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

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

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

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

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

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

SAST (Static Application Security Testing):

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

Features:

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

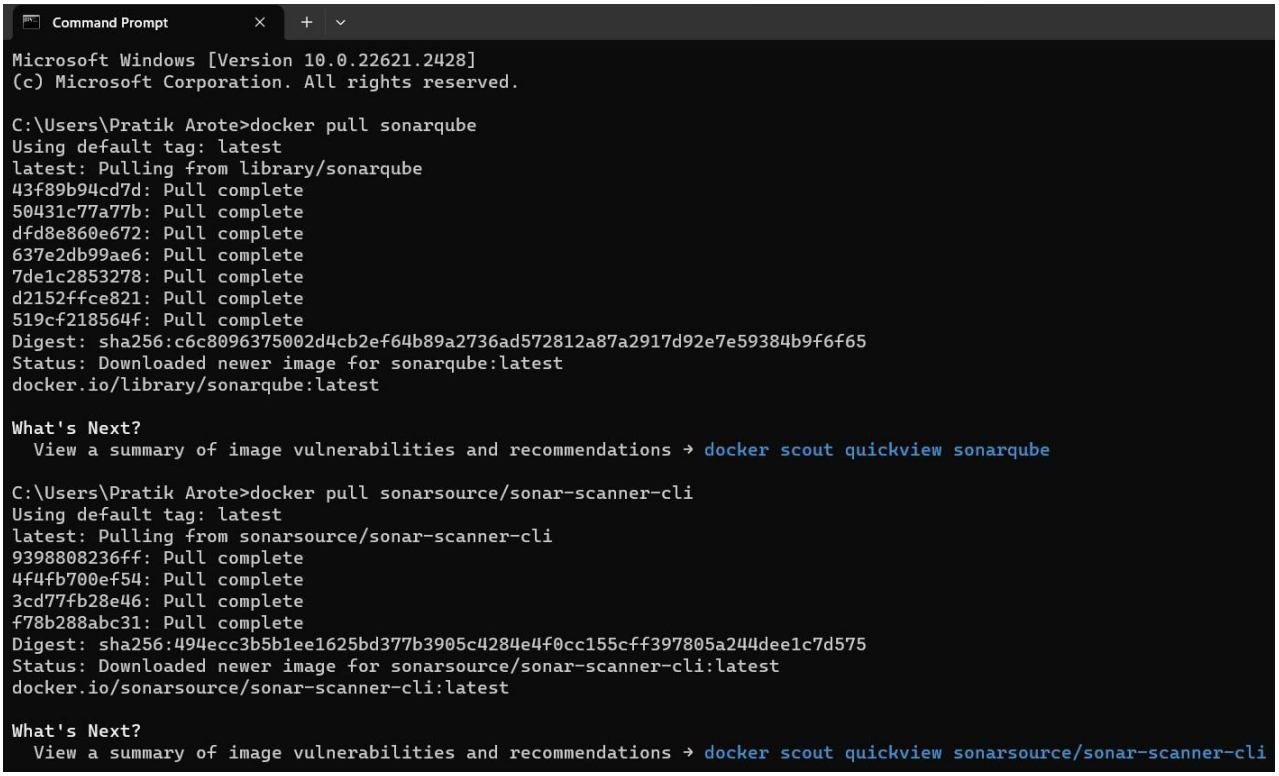
Early Detection: SAST is used early in the development process to find security issues before they can be exploited.

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

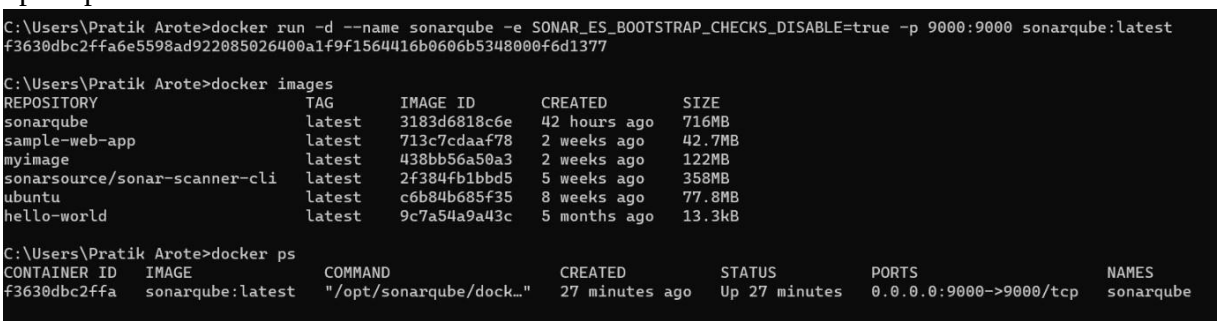
Integration: They can be integrated into CI/CD pipelines to automatically scan code before deployment.

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

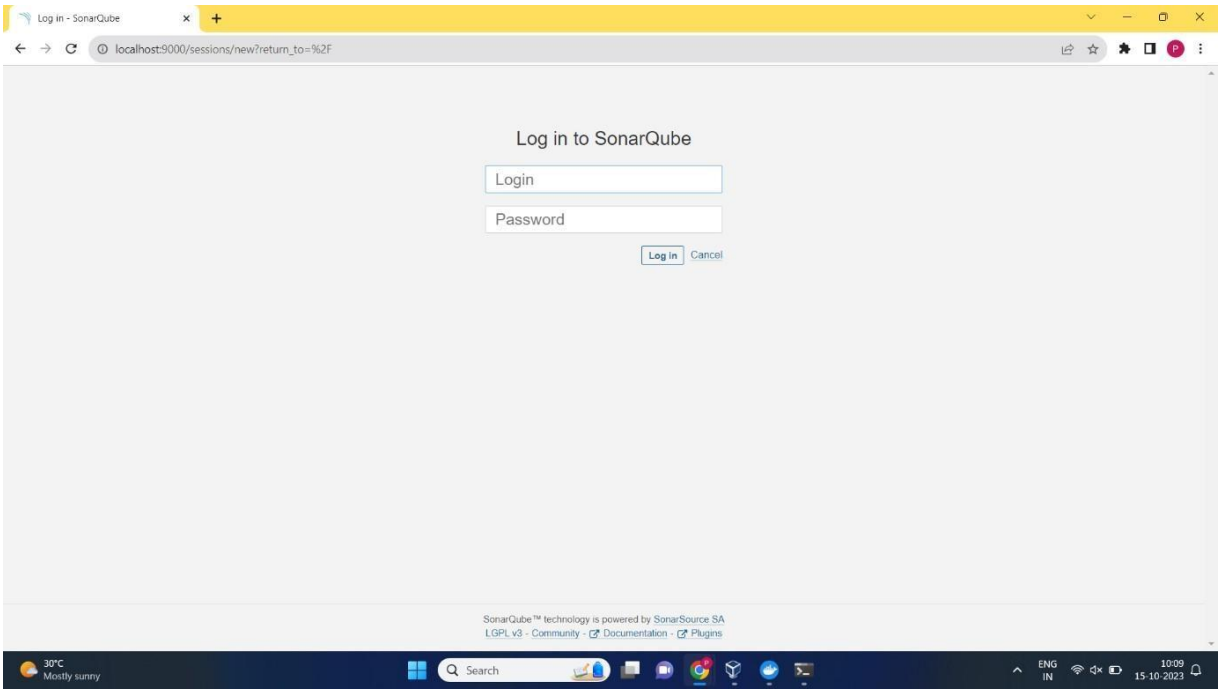
1. INSTALL sonarqube (docker images) and sonarscanner zip file from https://docs.sonarsource.com/sonarqube/latest/analyzing-sourcecode/scanners/sonarscanner/ and set up config file as given in docs.

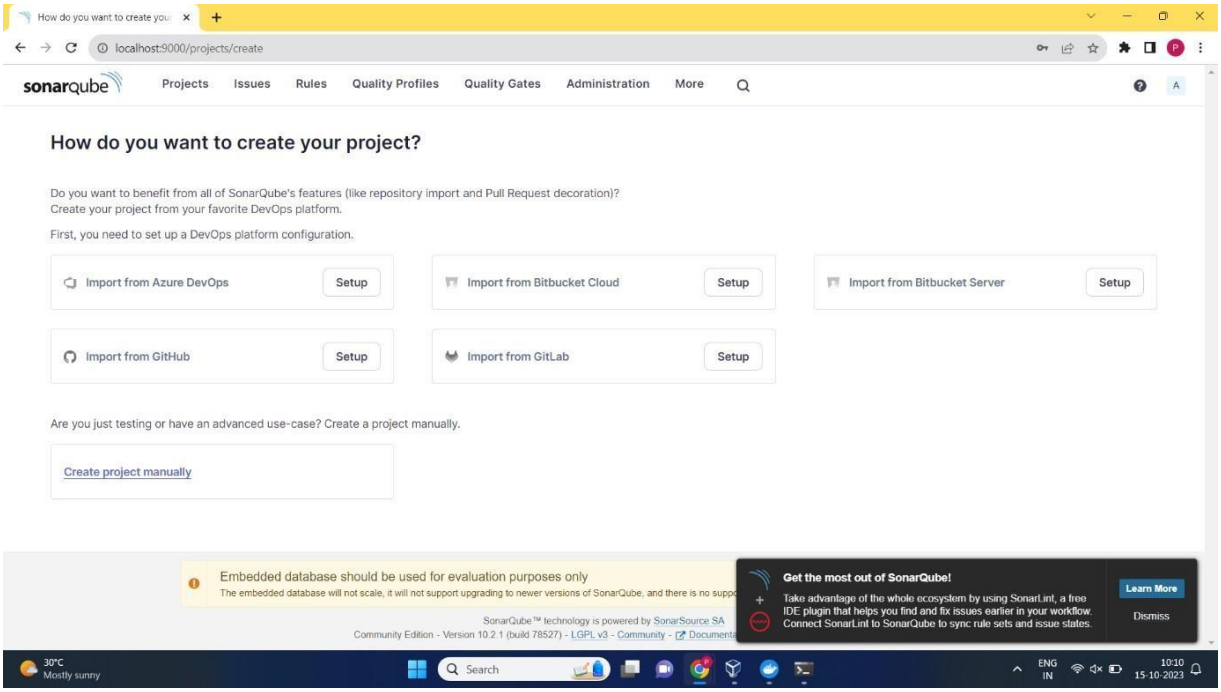


2. Spin up the container

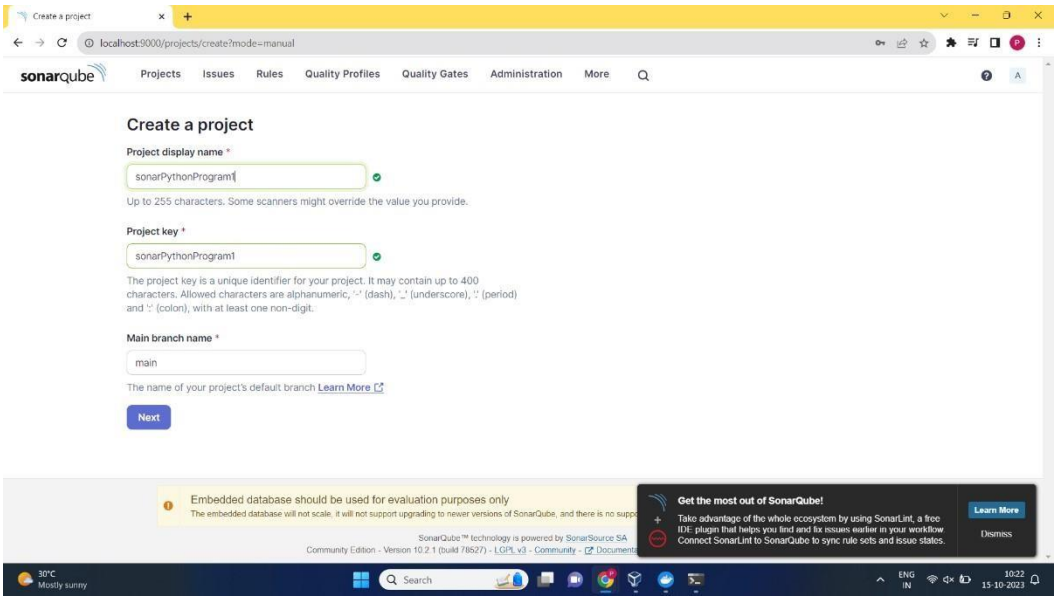


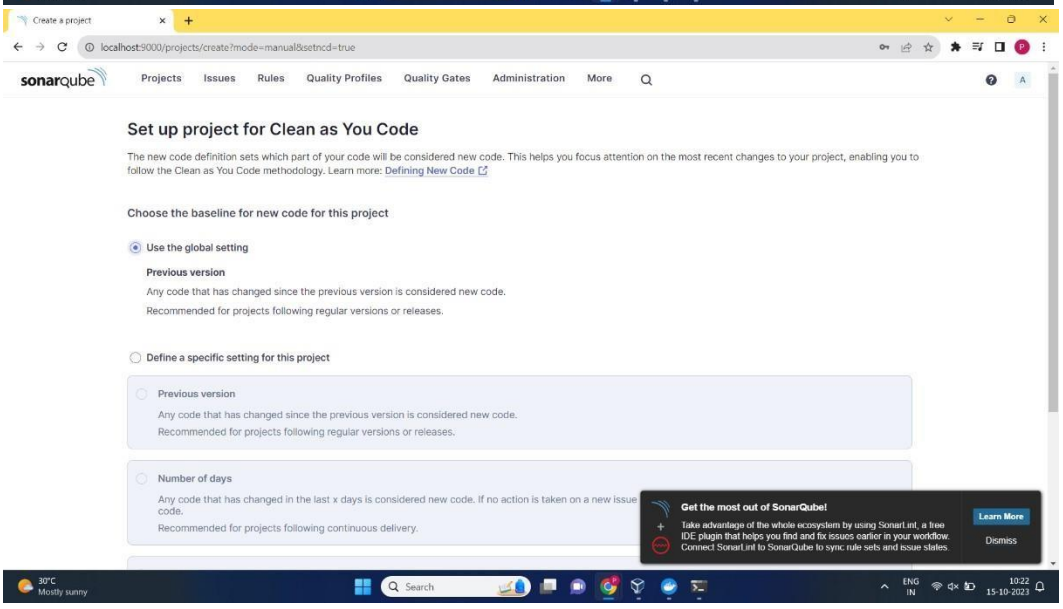
3. Open http://localhost:9000 on the browser. Enter login and password both as “admin” and then set up new password.

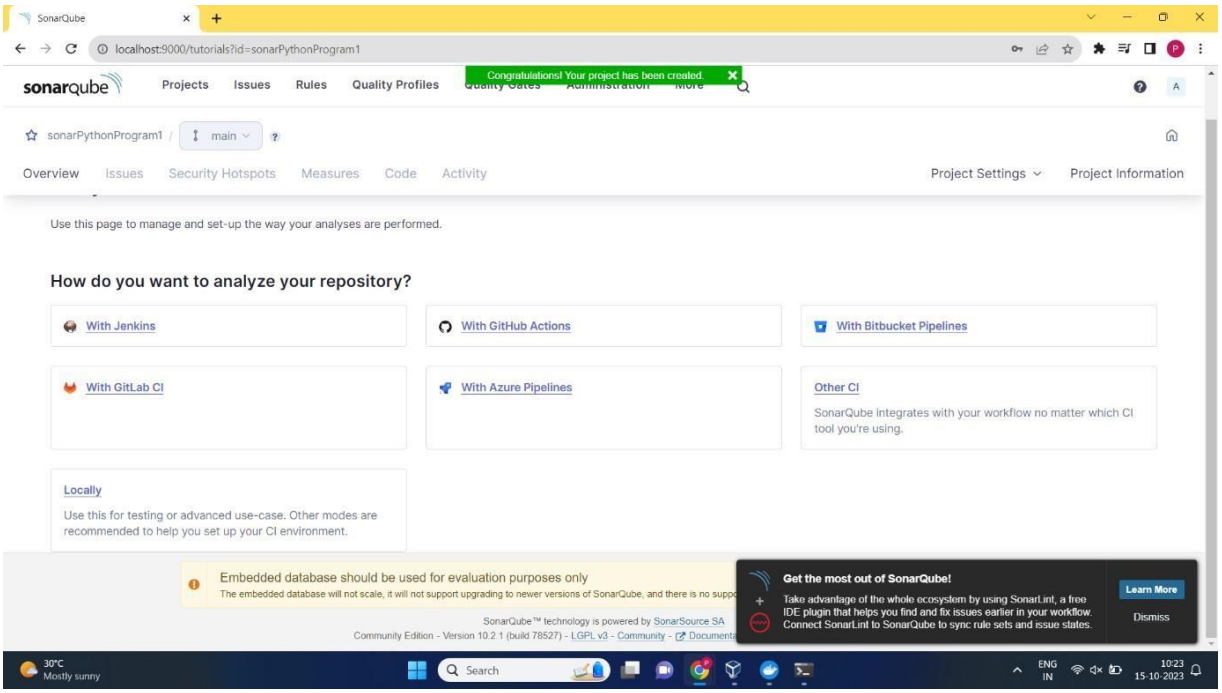




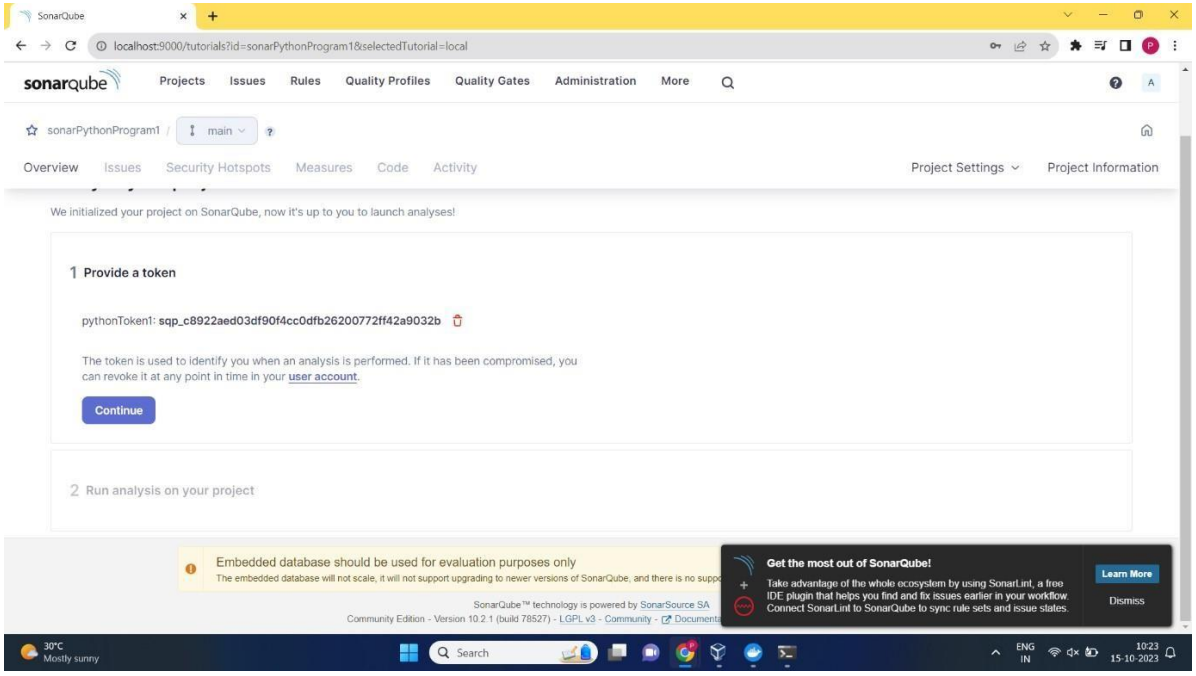
4. Create a project

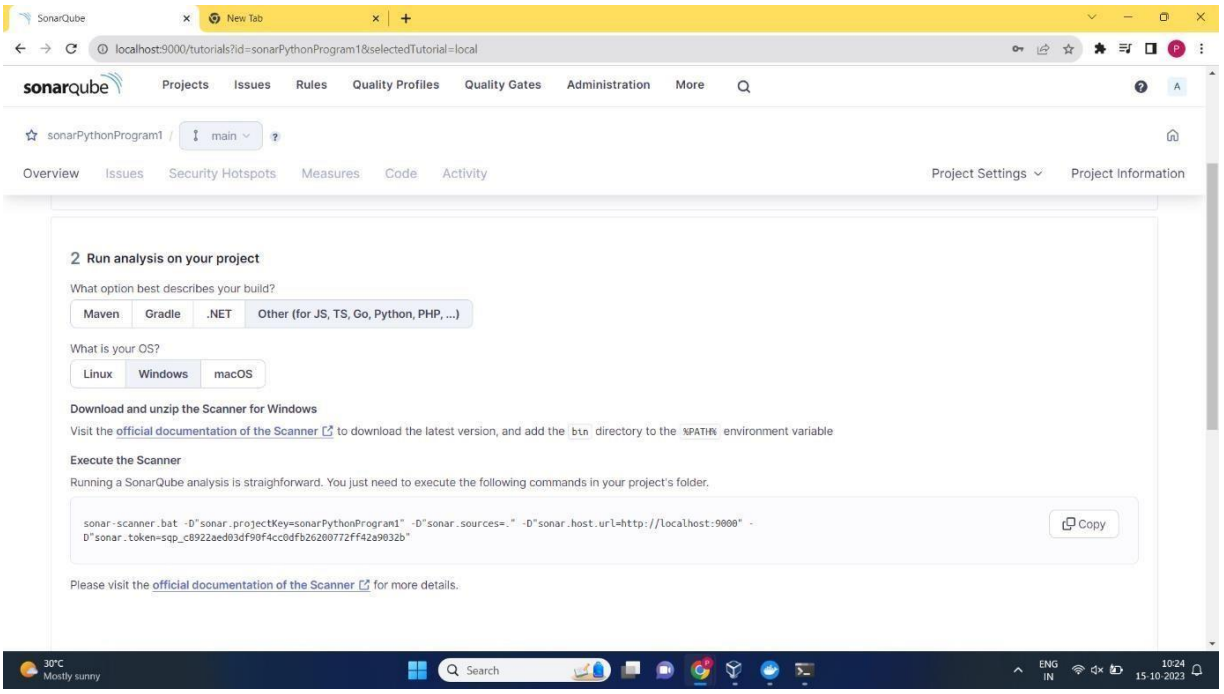

Create a project
localhost:9000/projects/create?mode=manual
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Create a project
Project display name *
sonarPythonProgram1
Up to 255 characters. Some scanners might override the value you provide.
Project key *
sonarPythonProgram1
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
SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 10.2.1 (build 78527) - LGPL v3 - Community - Documentation
30°C
Mostly sunny
Search
ENG
IN
10:22
15-10-2023


Create a project
localhost:9000/projects/create?mode=manual&setncd=true
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Set up project for Clean as You Code
The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: Defining New Code
Choose the baseline for new code for this project
Use the global setting
Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
Define a specific setting for this project
Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.
Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue
code.
Recommended for projects following continuous delivery.
Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
Learn More
Dismiss
30°C
Mostly sunny
Search
ENG
IN
10:22
15-10-2023

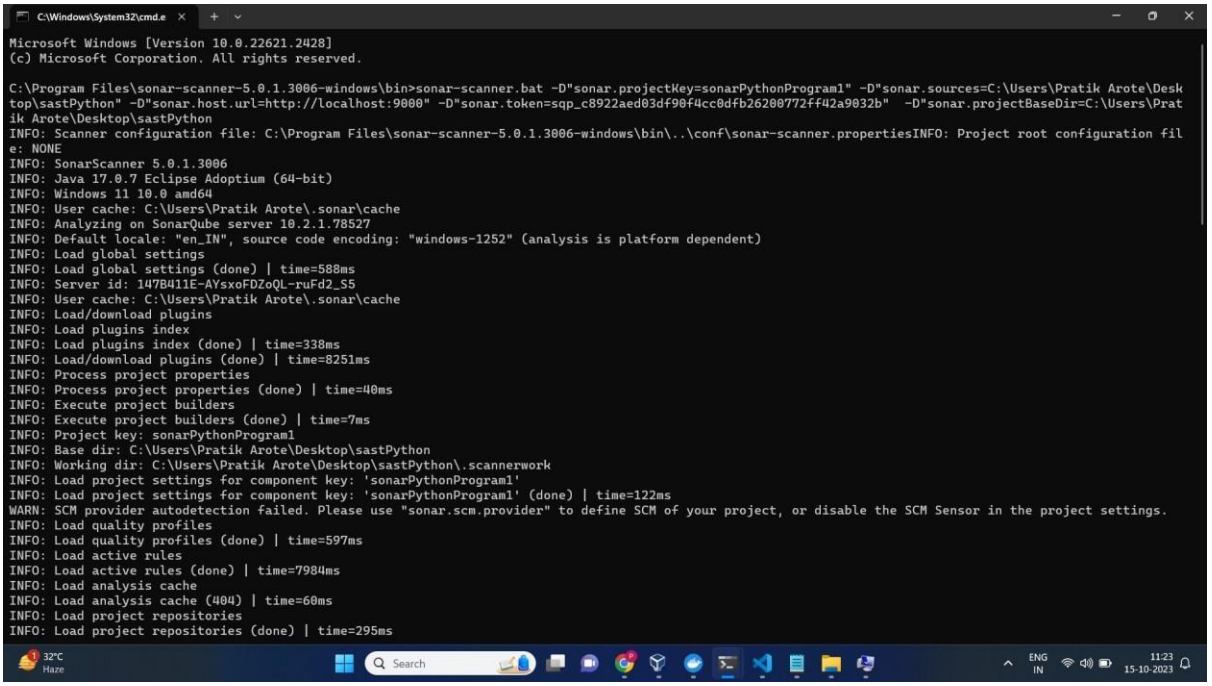


5. Provide token





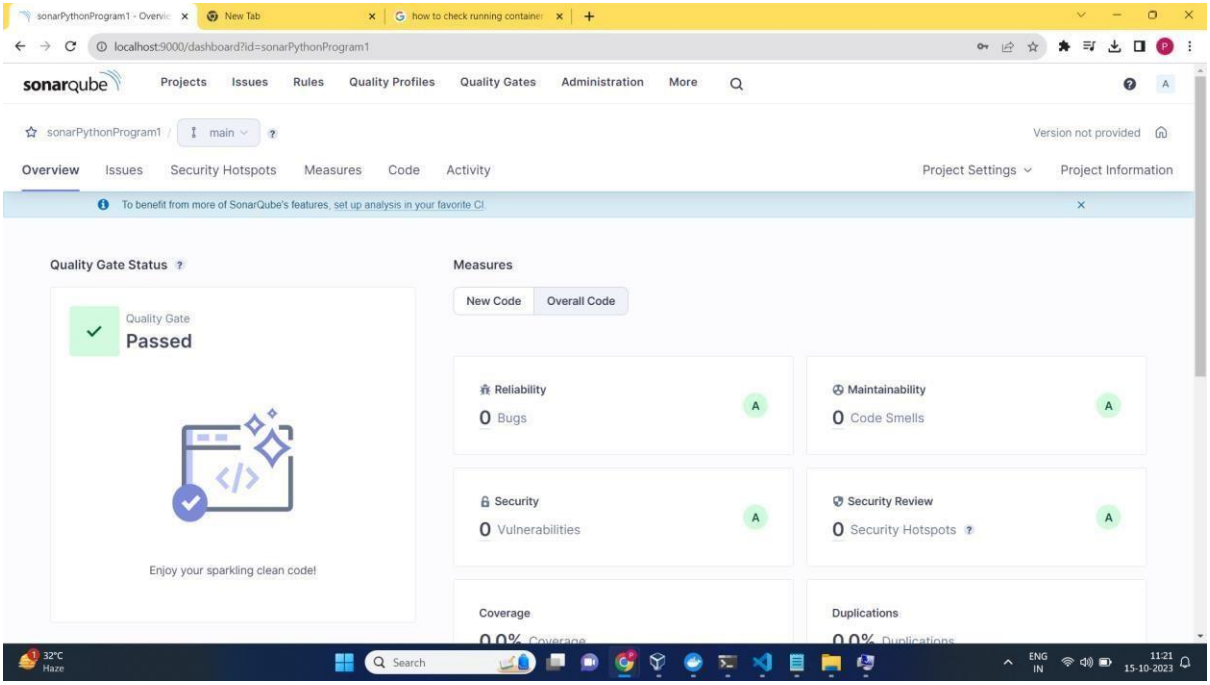
6. Enter the following command



```
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: ------------ Run sensors on project
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=7ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=47ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor 1 file had no CPD blocks
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 253ms, dir size=136.5 kB
INFO: Analysis report compressed in 48ms, zip size=17.5 kB
INFO: Analysis report uploaded in 201ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarPythonProgram1
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AYsx47EpoQL-ruFd3M3Y
INFO: Analysis total time: 22.756 s
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AYsx47EpoQL-ruFd3M3Y
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 35.565s
INFO: Final Memory: 23M/77M
INFO: ------------------------------------------------------------------------

C:\Program Files\sonar-scanner-5.0.1.3006-windows\bin>
```

7. See the result of the test



CONCLUSION:

Here we have successfully performed static analysis of python programs.

Roll No:10

Saish Bavalekar

ASSIGNMENT 7

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

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

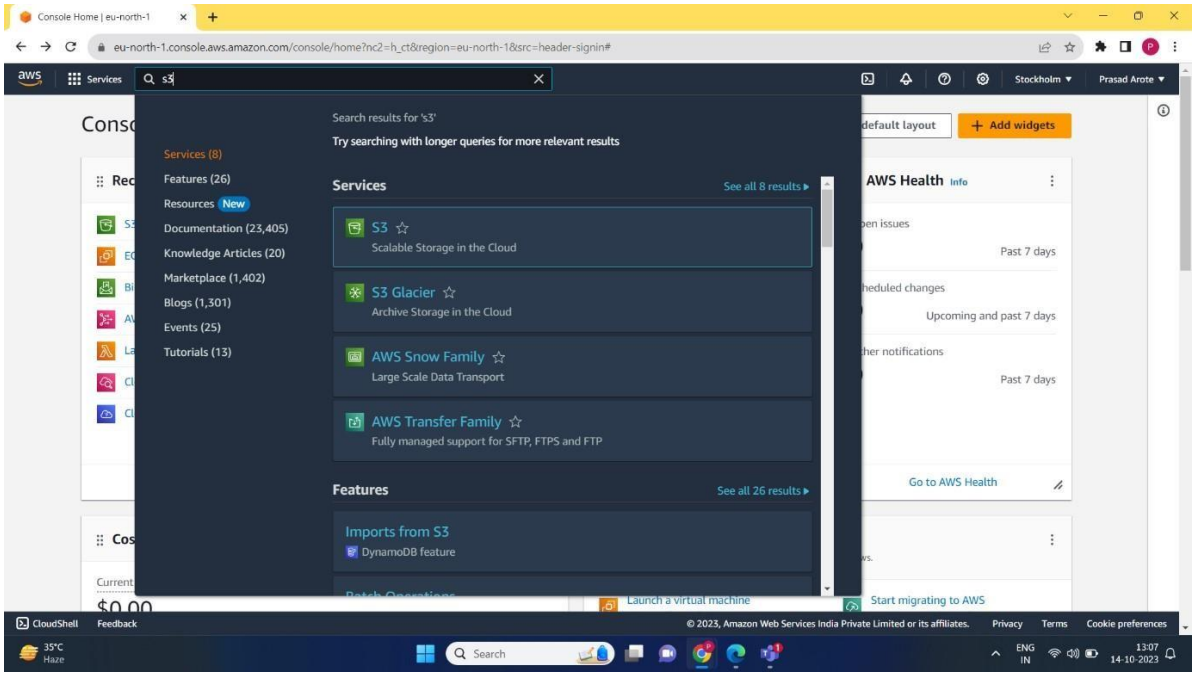
THEORY:

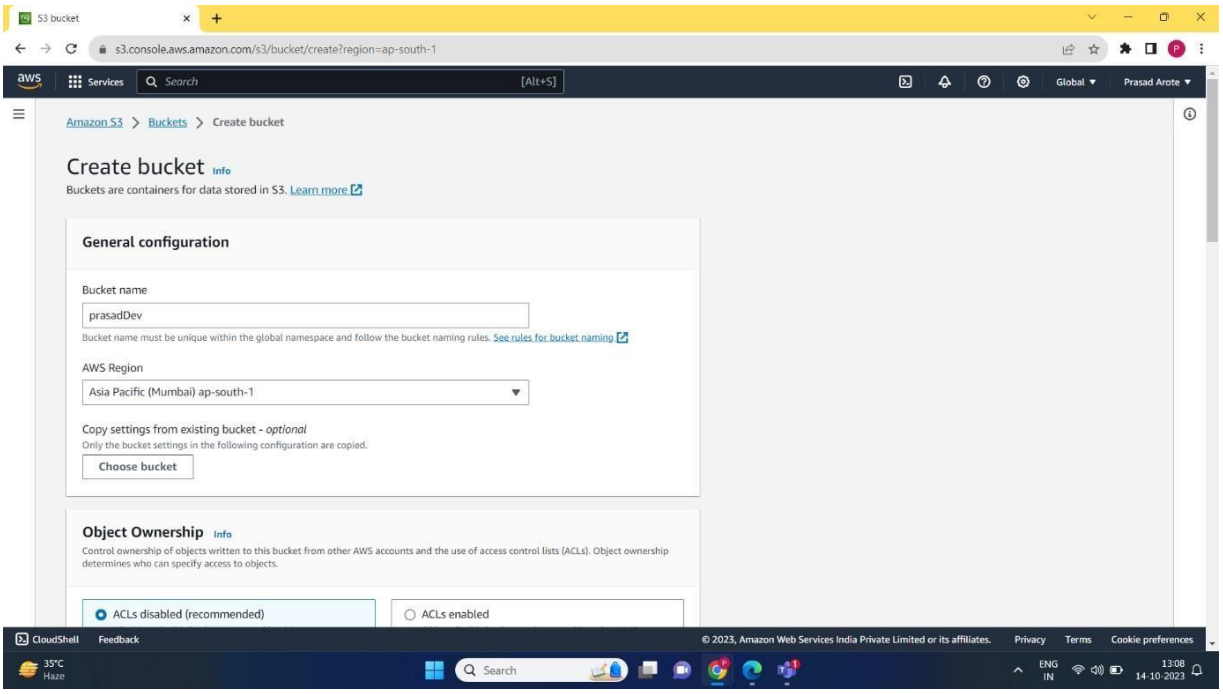
LAMBDA FUNCTION

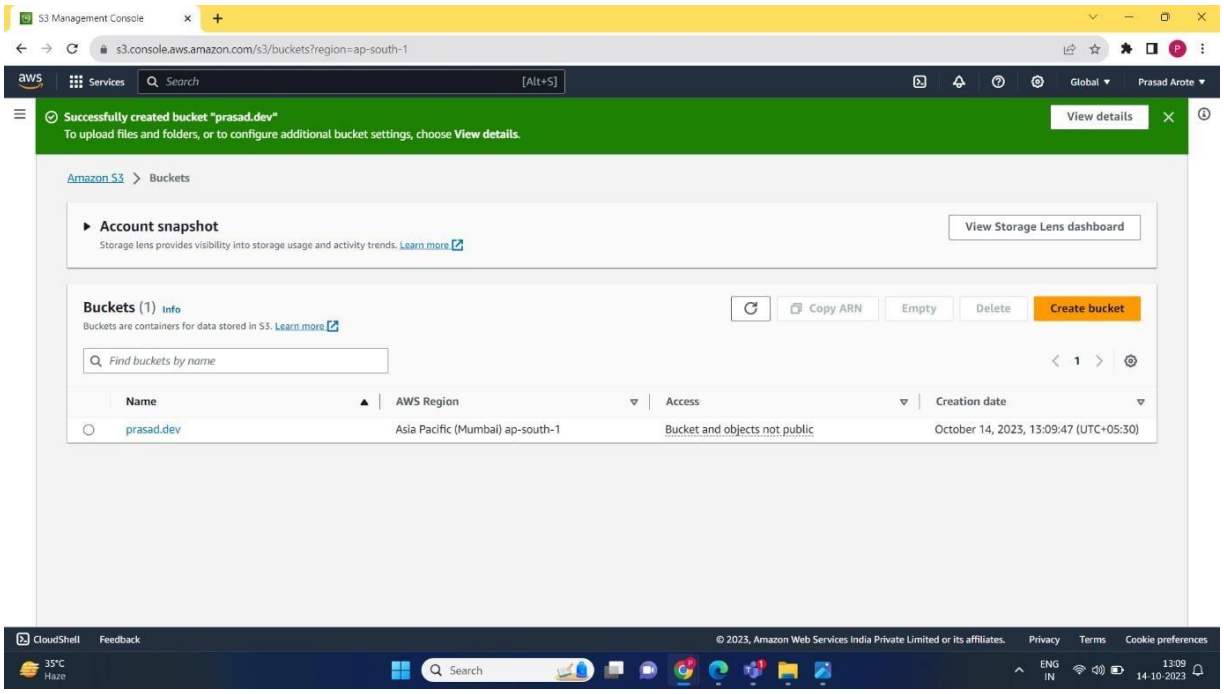
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you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

Installation:

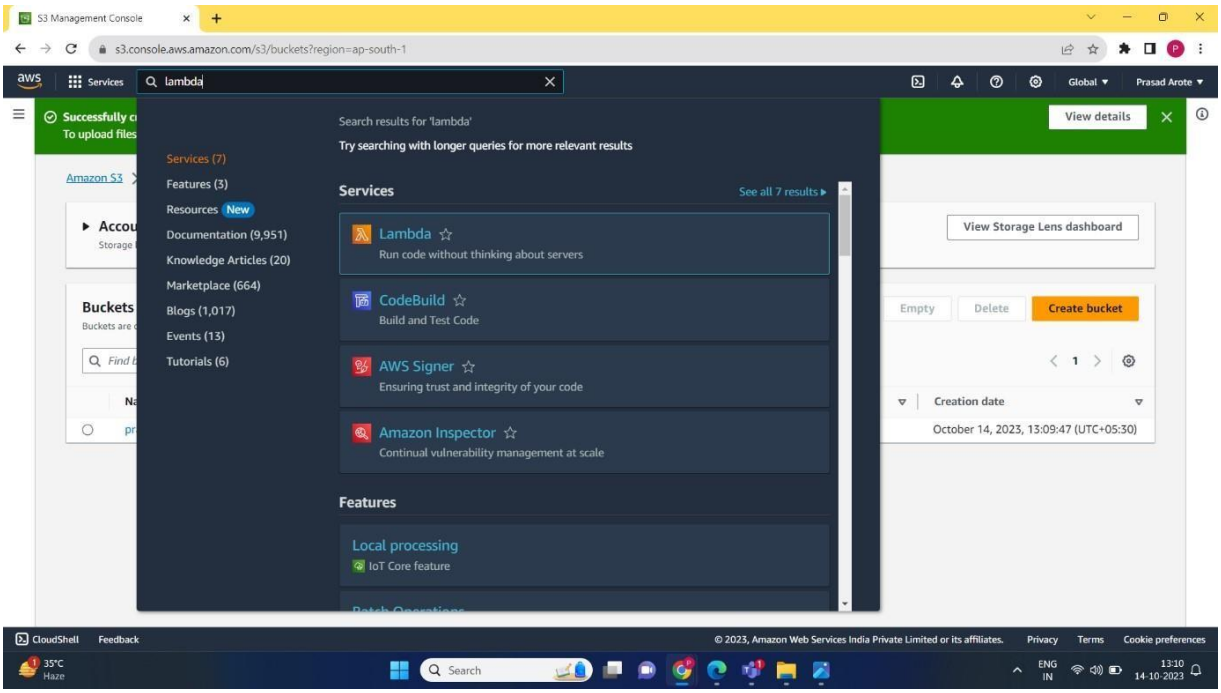
1. Create a S3 bucket

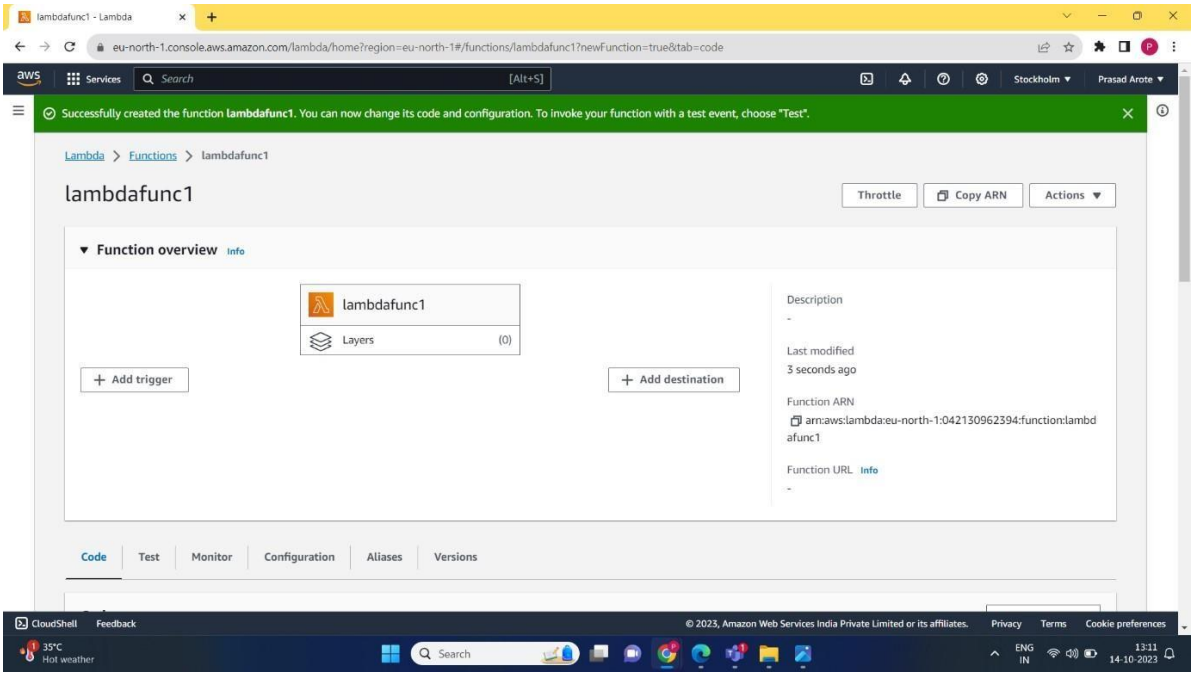


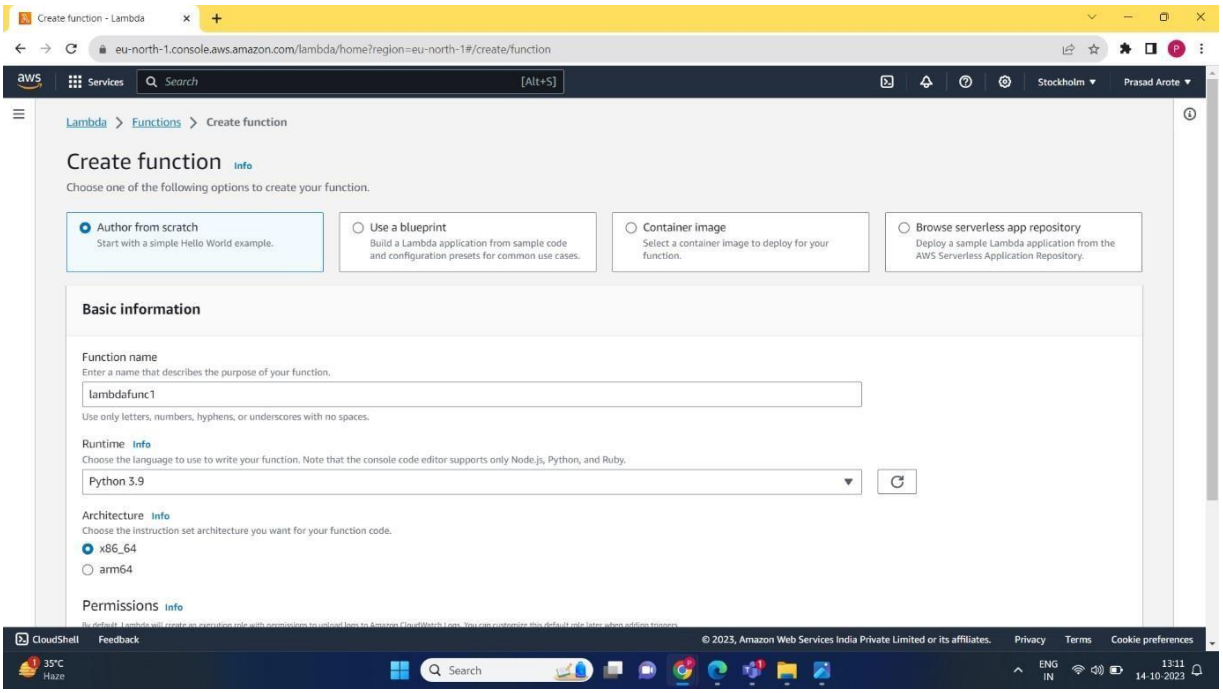




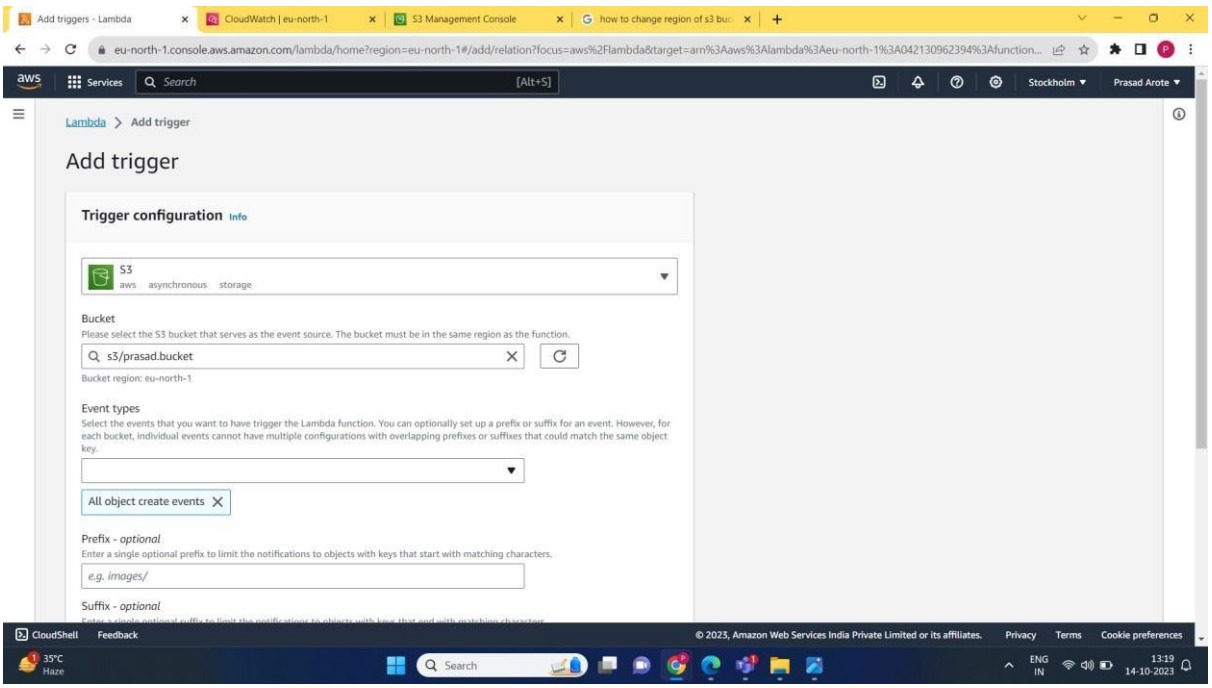
2. Create a Lambda function.

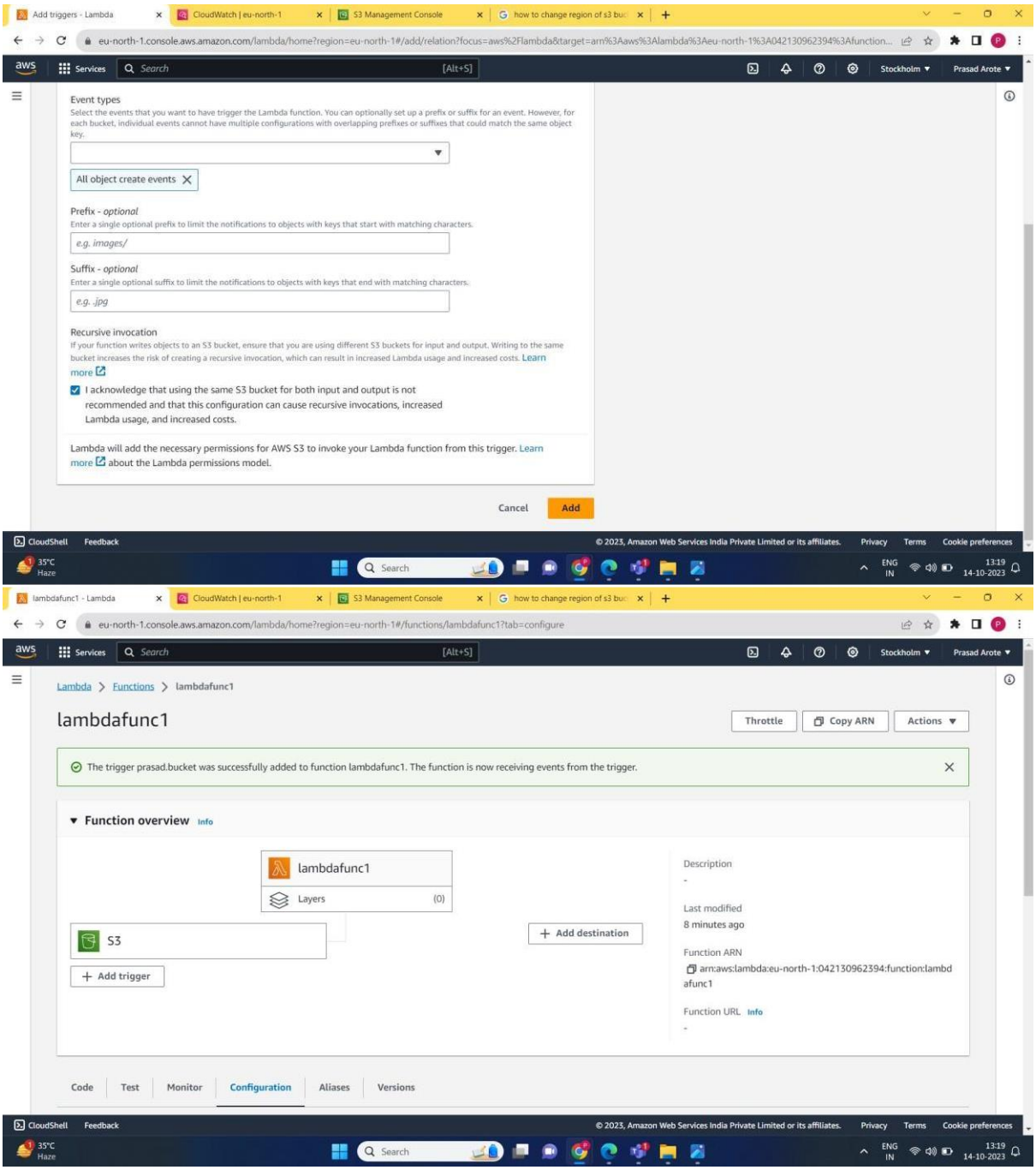

S3 Management Console
s3.console.aws.amazon.com/s3/buckets?region=ap-south-1
Services
lambda
Global
Prasad Arote
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services (7)
Features (3)
Resources New
Documentation (9,951)
Knowledge Articles (20)
Marketplace (664)
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

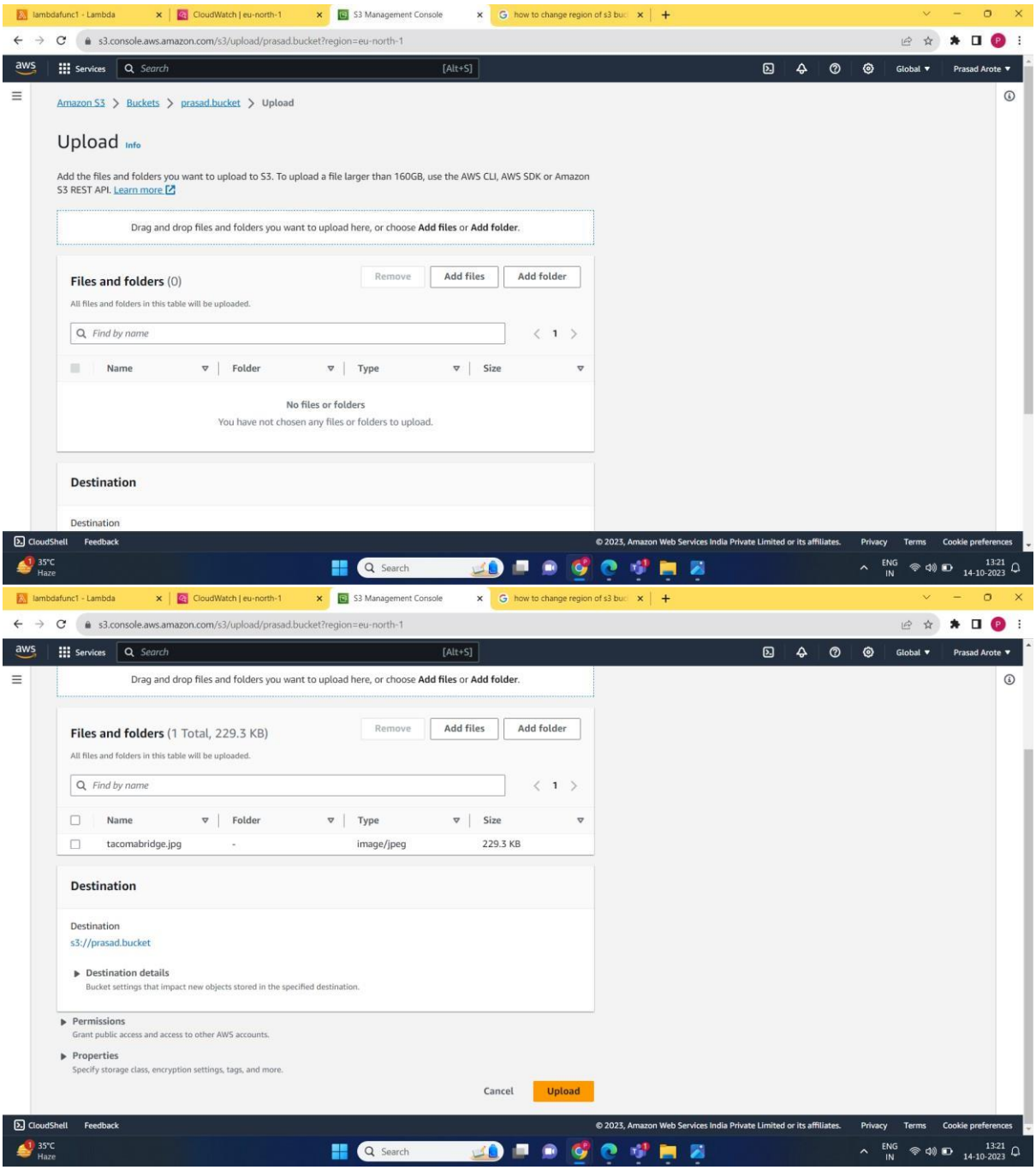

lambdafunc1 - Lambda
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc1?newFunction=true&tab=code
Services
Search
[Alt+S]
Stockholm
Prasad Arote
Successfully created the function lambdafunc1. You can now change its code and configuration. To invoke your function with 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



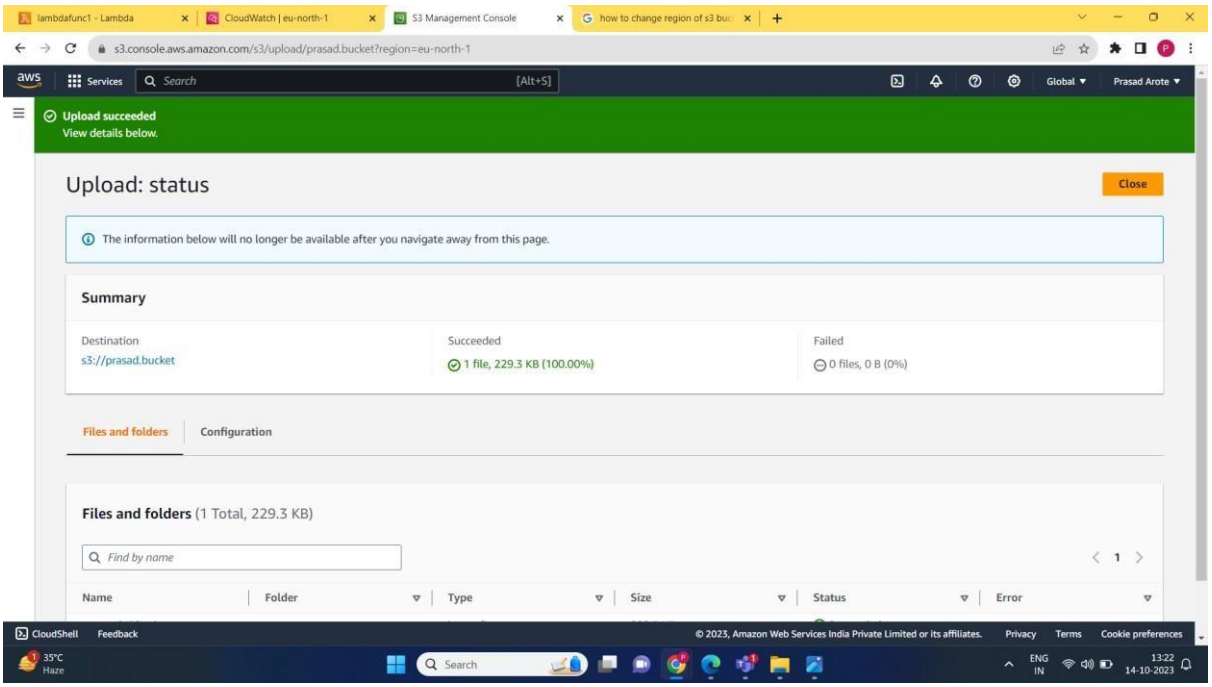
3. Create a trigger



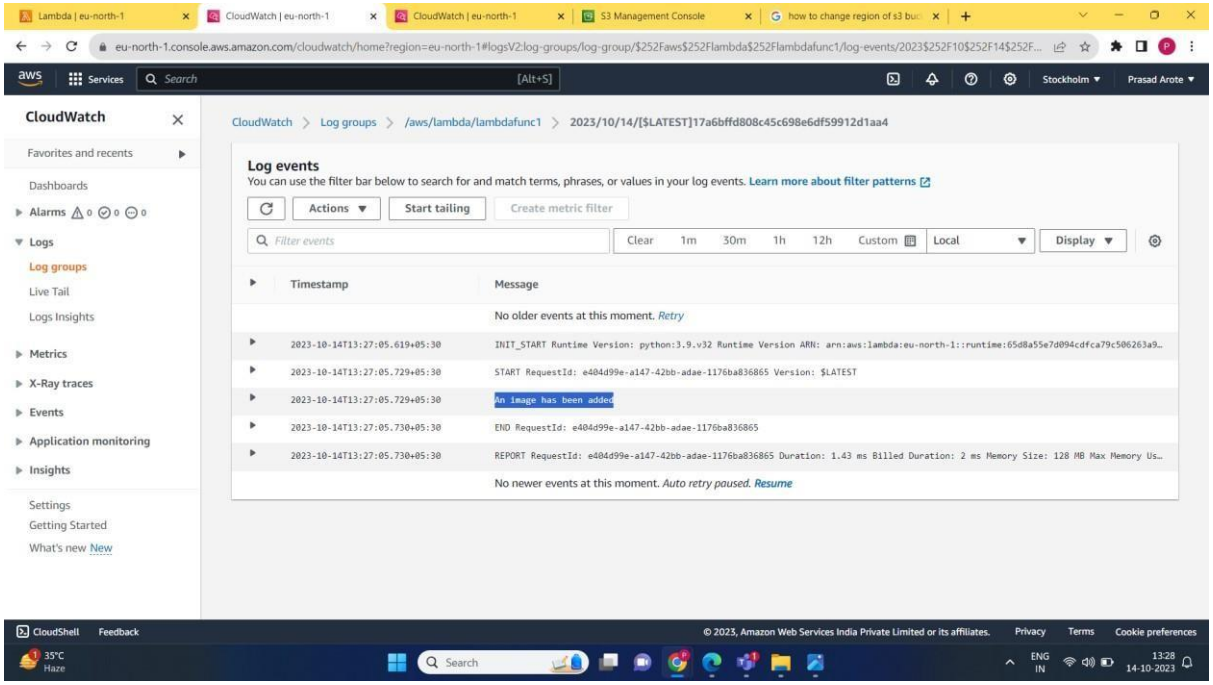

Add triggers - Lambda
CloudWatch | eu-north-1
S3 Management Console
how to change region of s3 buc
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aeu-north-1%3A042130962394%3Afunction...
Services
Search
[Alt+S]
Stockholm
Prasad 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
how to change region of s3 buc
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.

Roll No:10

Saish Bavalekar

ASSIGNMENT 8

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

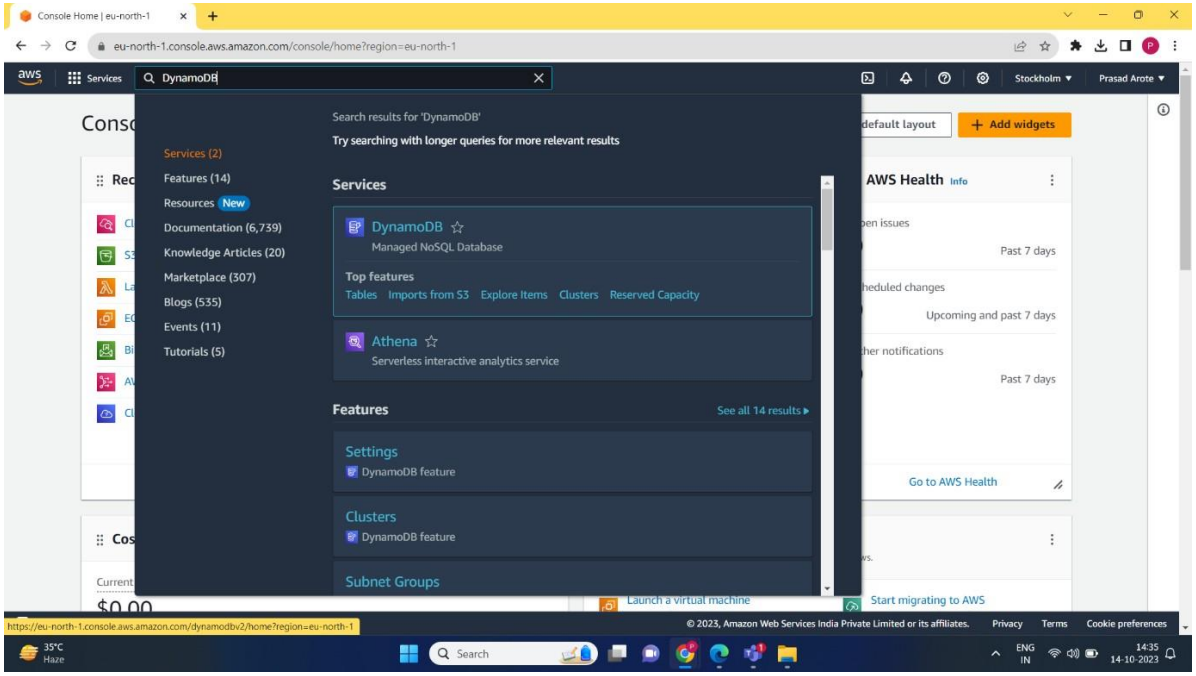
LO6: To engineer a composition of nano services using AWS Lambda and Step Functions with the serverless framework.

THEORY:

DYNAMO DB

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

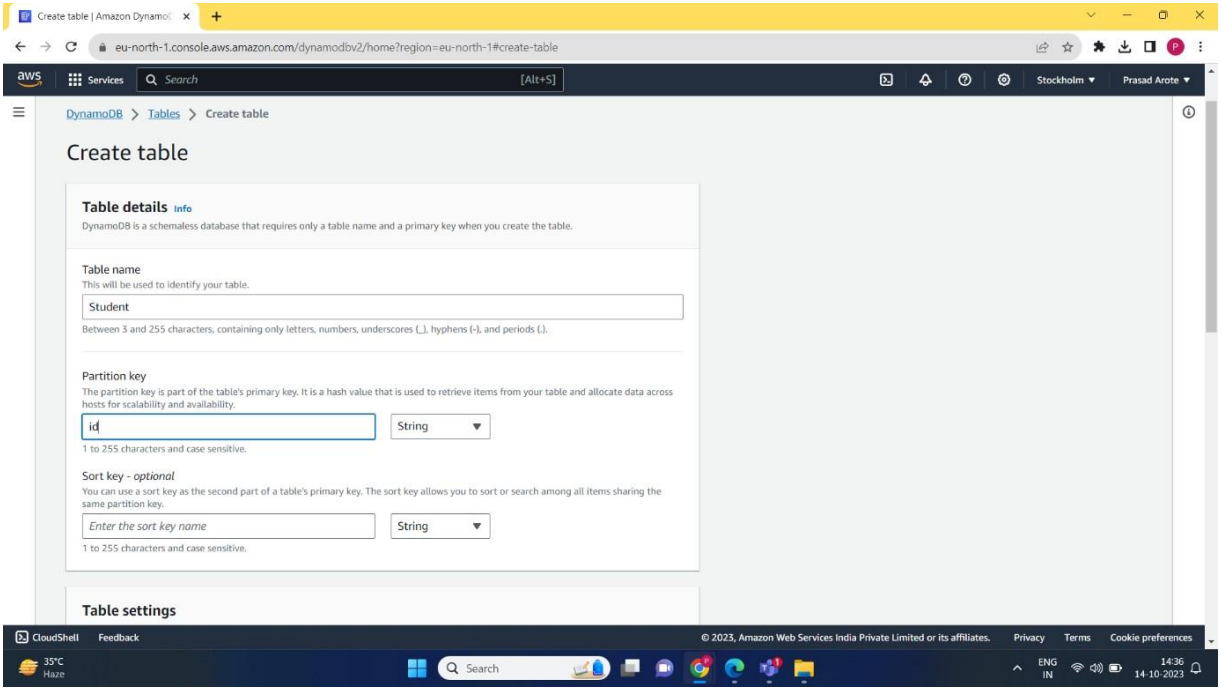
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource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


Console Home | eu-north-1
eu-north-1.console.aws.amazon.com/console/home?region=eu-north-1
aws
Services
DynamoDB
Stockholm
Prasad Arote
Search results for 'DynamoDB'
Try searching with longer queries for more relevant results
Services (2)
Features (14)
Resources New
Documentation (6,739)
Knowledge Articles (20)
Marketplace (307)
Blogs (535)
Events (11)
Tutorials (5)
Services
DynamoDB
Managed NoSQL Database
Top features
Tables Imports from S3 Explore Items Clusters Reserved Capacity
Athena
Serverless interactive analytics service
Features
See all 14 results
Settings
DynamoDB feature
Clusters
DynamoDB feature
Subnet Groups
default layout
Add widgets
AWS Health Info
Past 7 days
Upcoming and past 7 days
Past 7 days
Go to AWS Health
Launch a virtual machine
Start migrating to AWS
https://eu-north-1.console.aws.amazon.com/dynamodbv2/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
14:35
14-10-2023

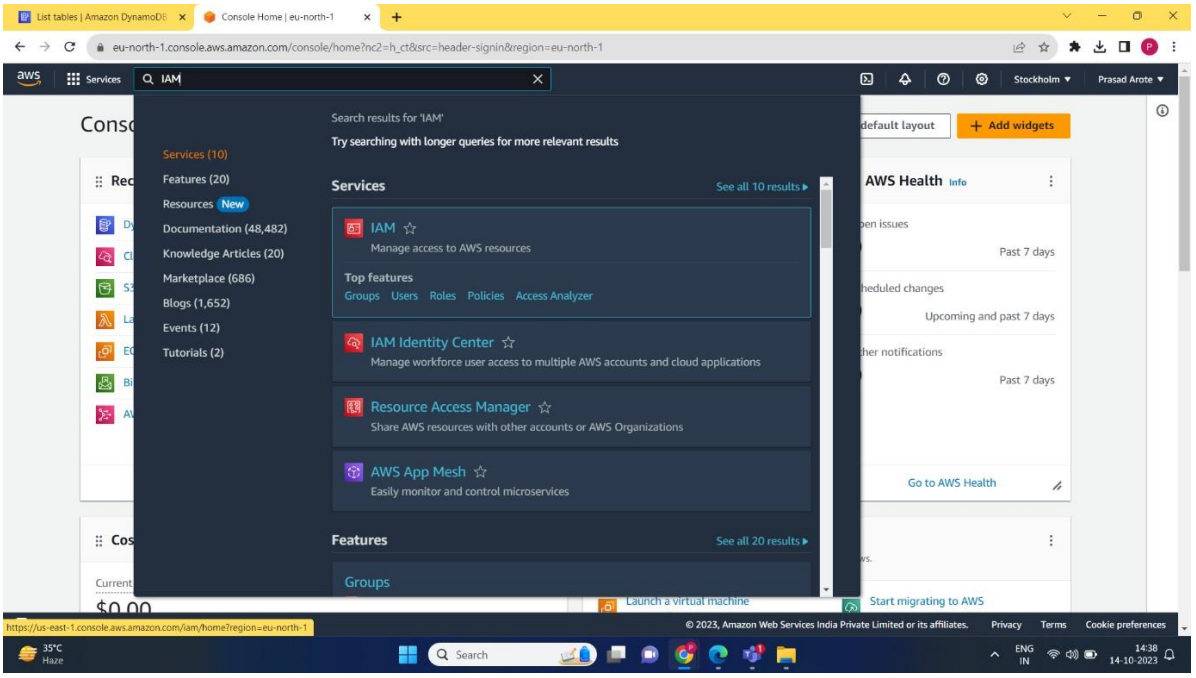
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

STEPS:

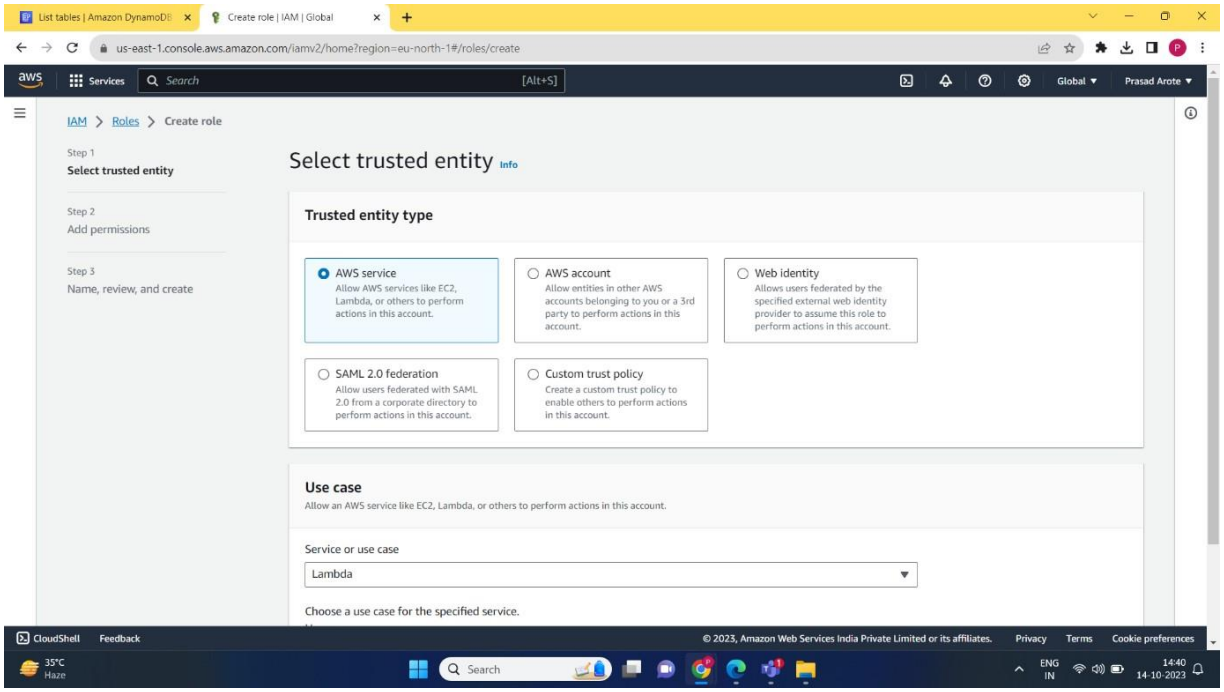
1. Create a table

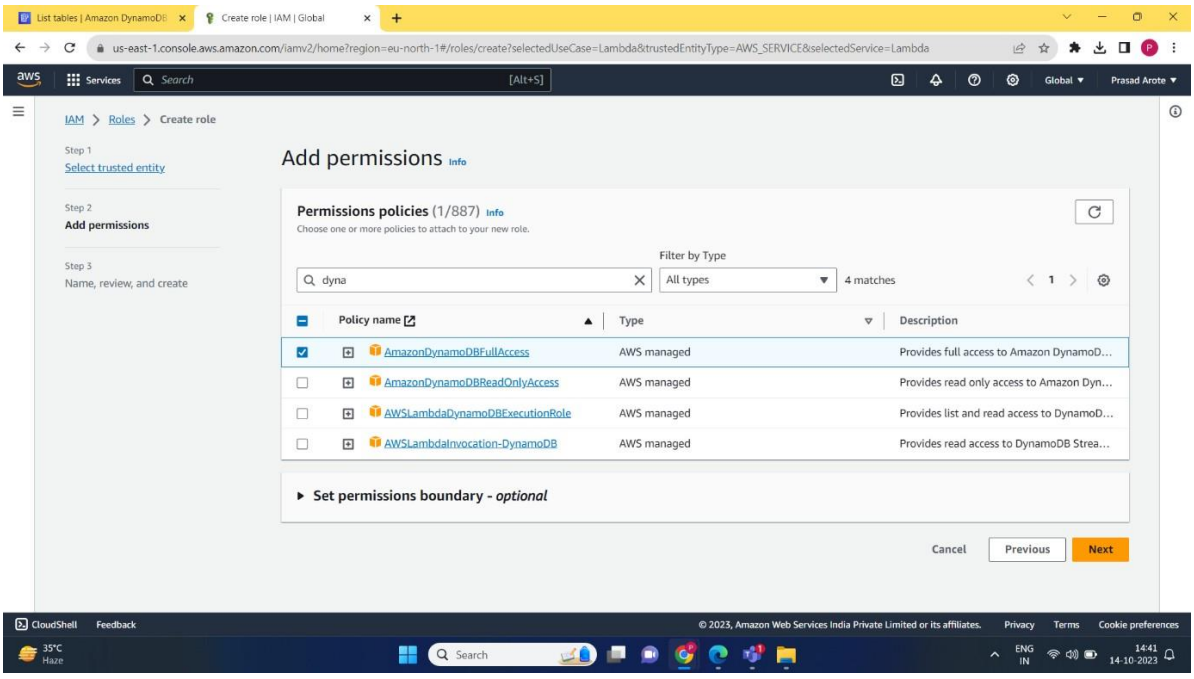


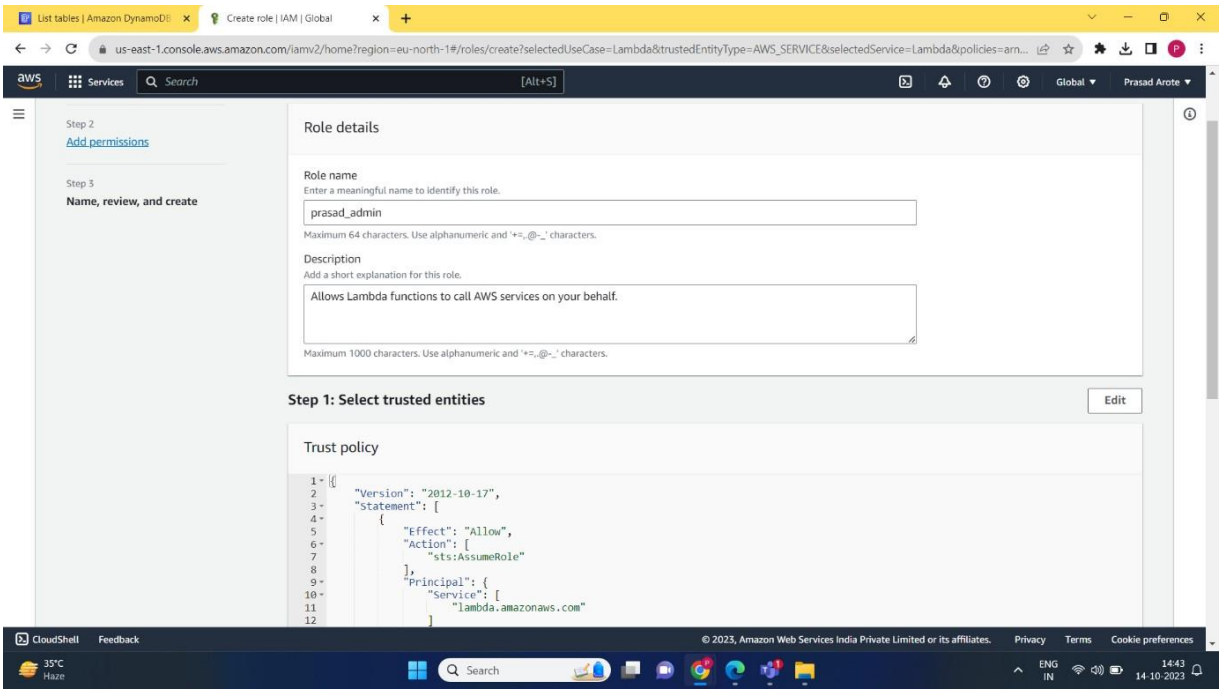
2. Create a role using IAM

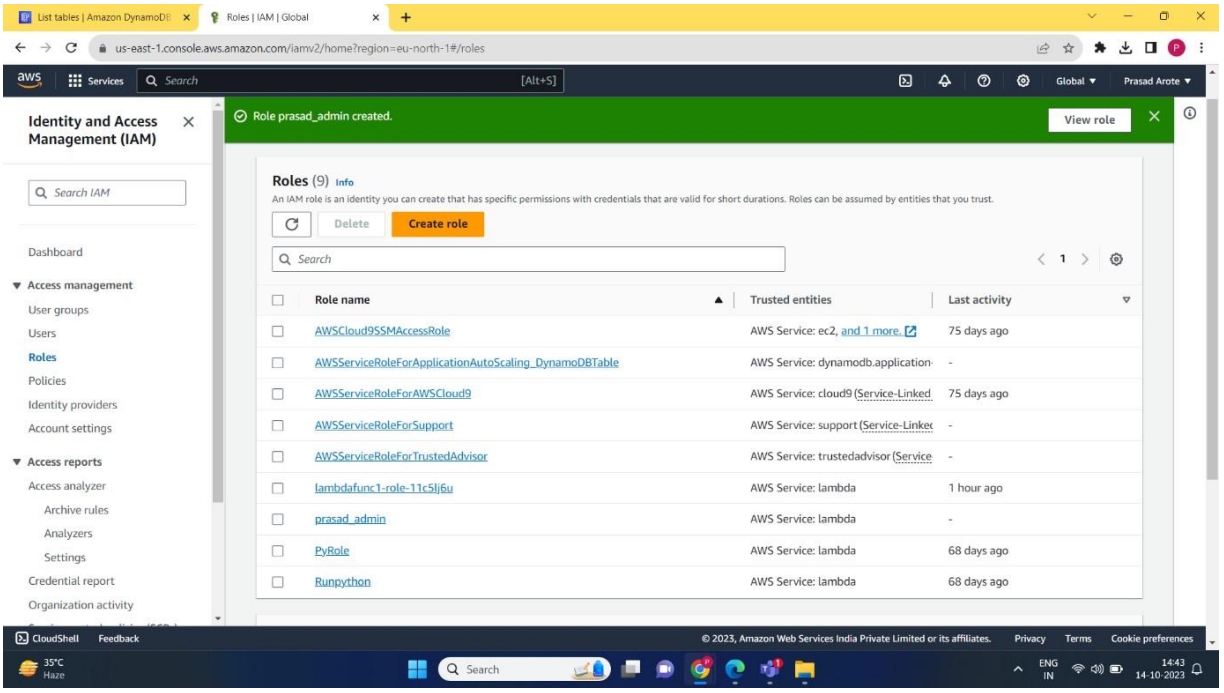


3. Add permissions – AmazonDynamoFullAccess

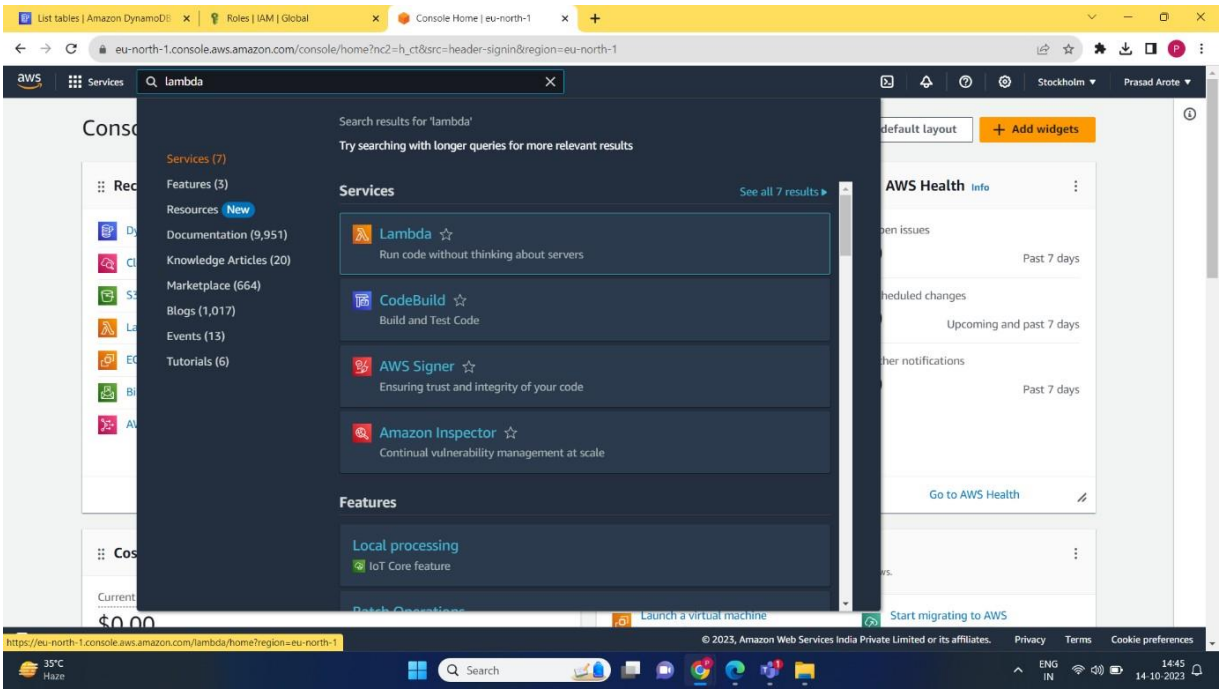


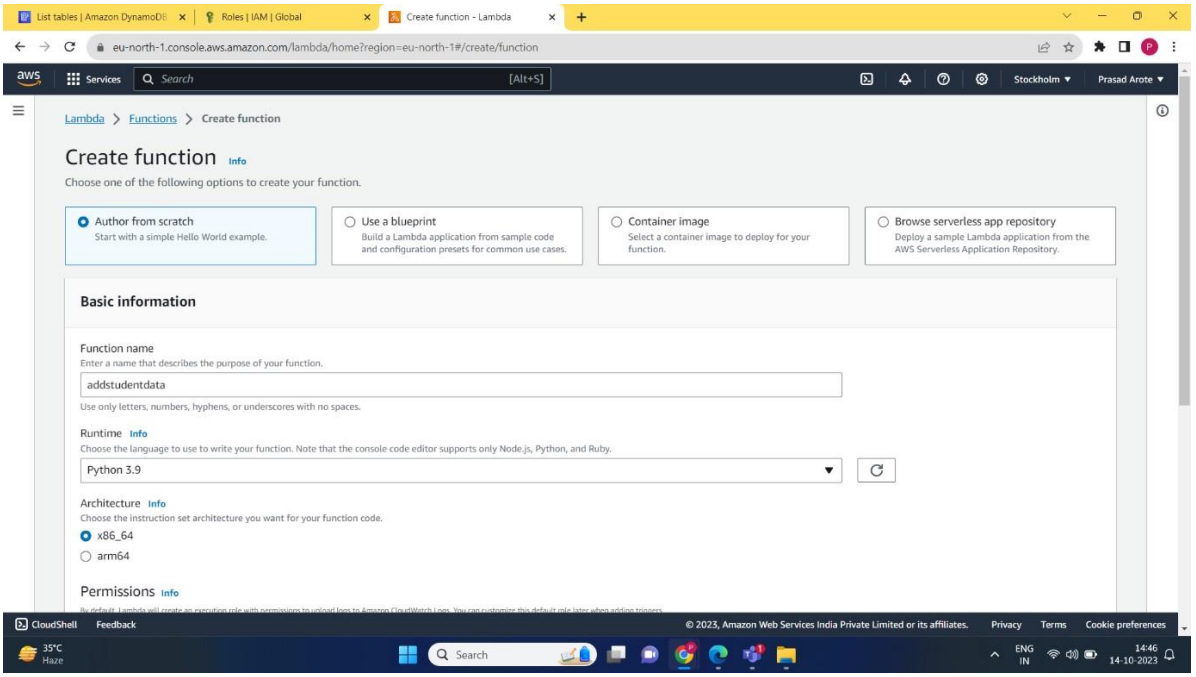


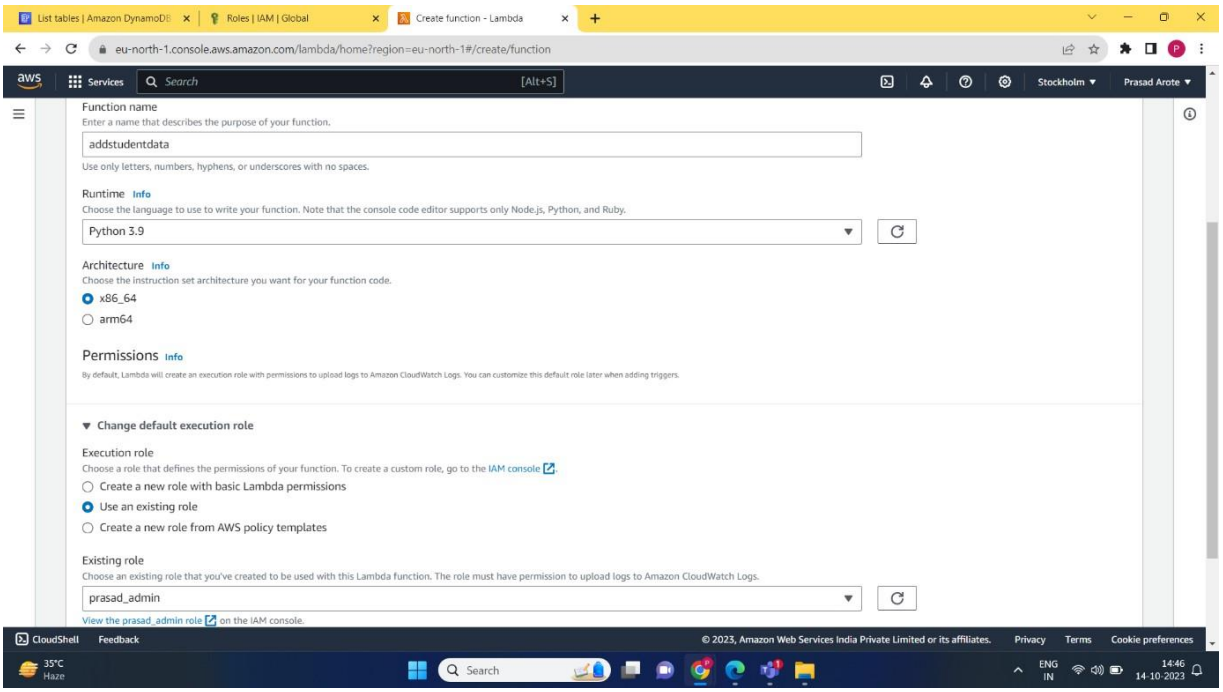


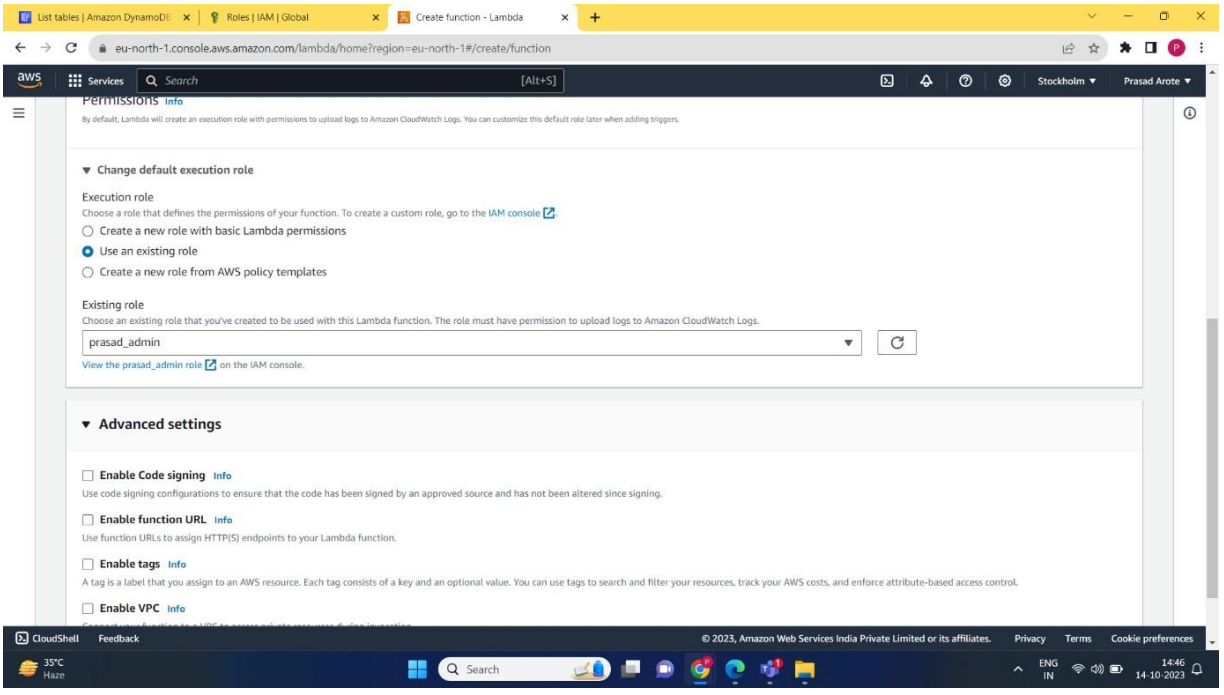


4. Create a Lambda Function

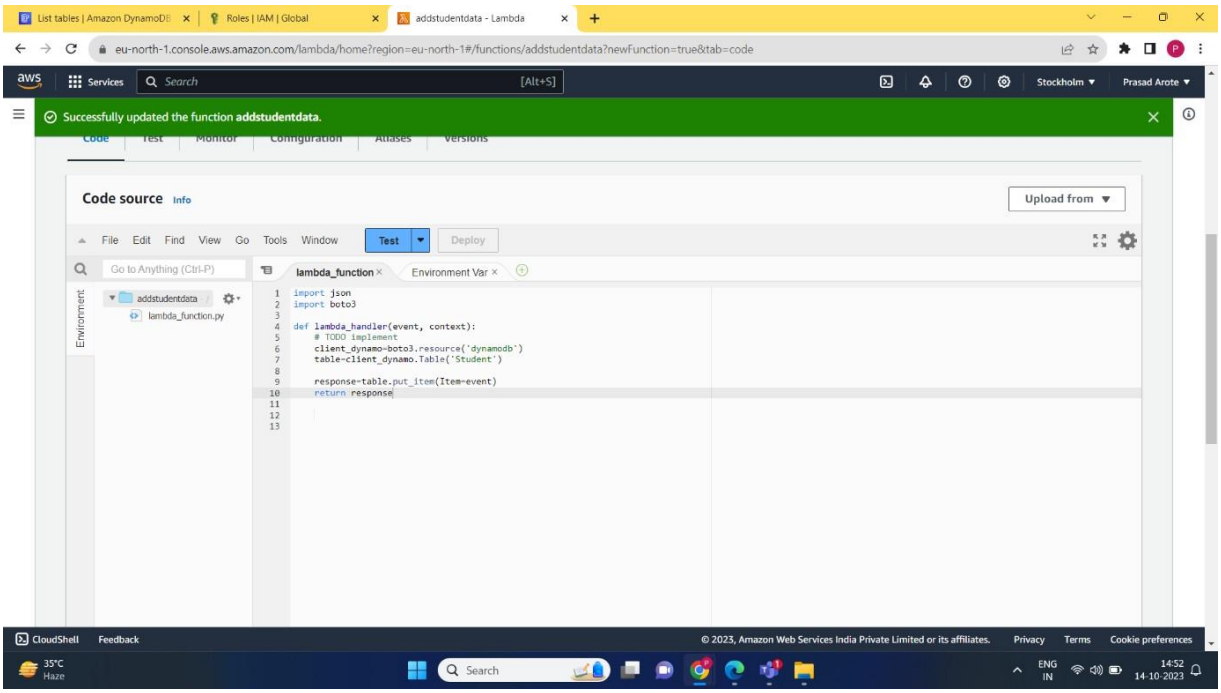

List tables | Amazon DynamoDB
Roles | IAM | Global
Console Home | eu-north-1
eu-north-1.console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin&region=eu-north-1
Services
lambda
Stockholm
Prasad Arote
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services (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

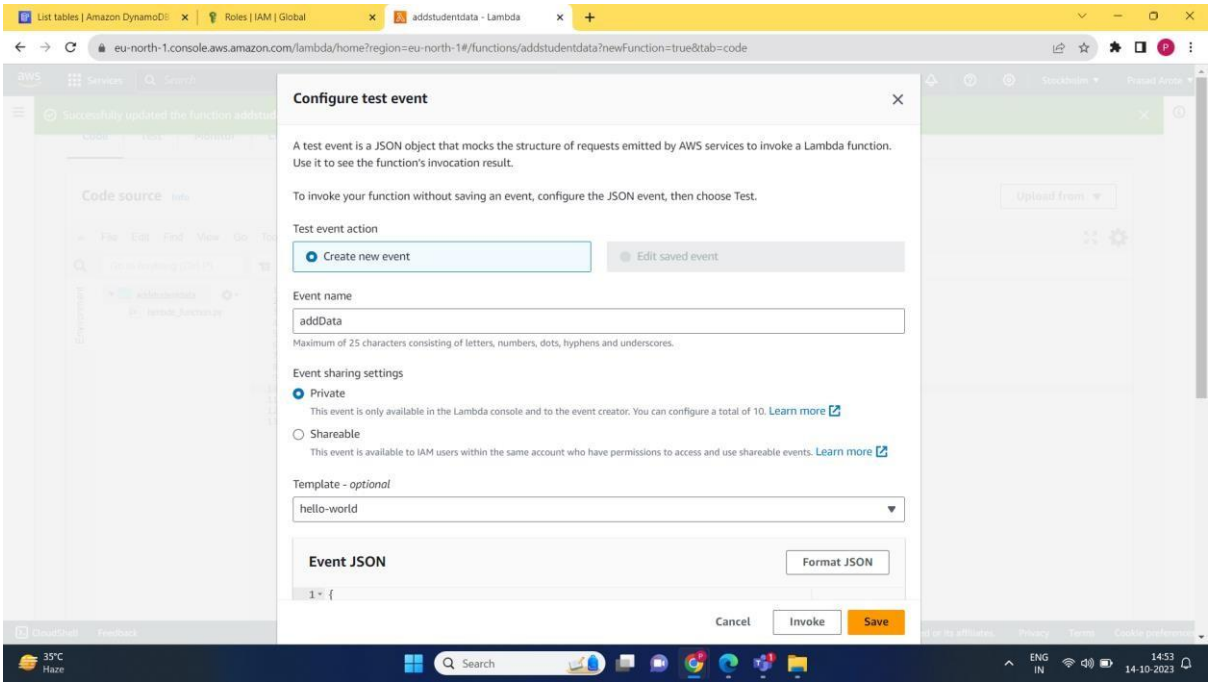


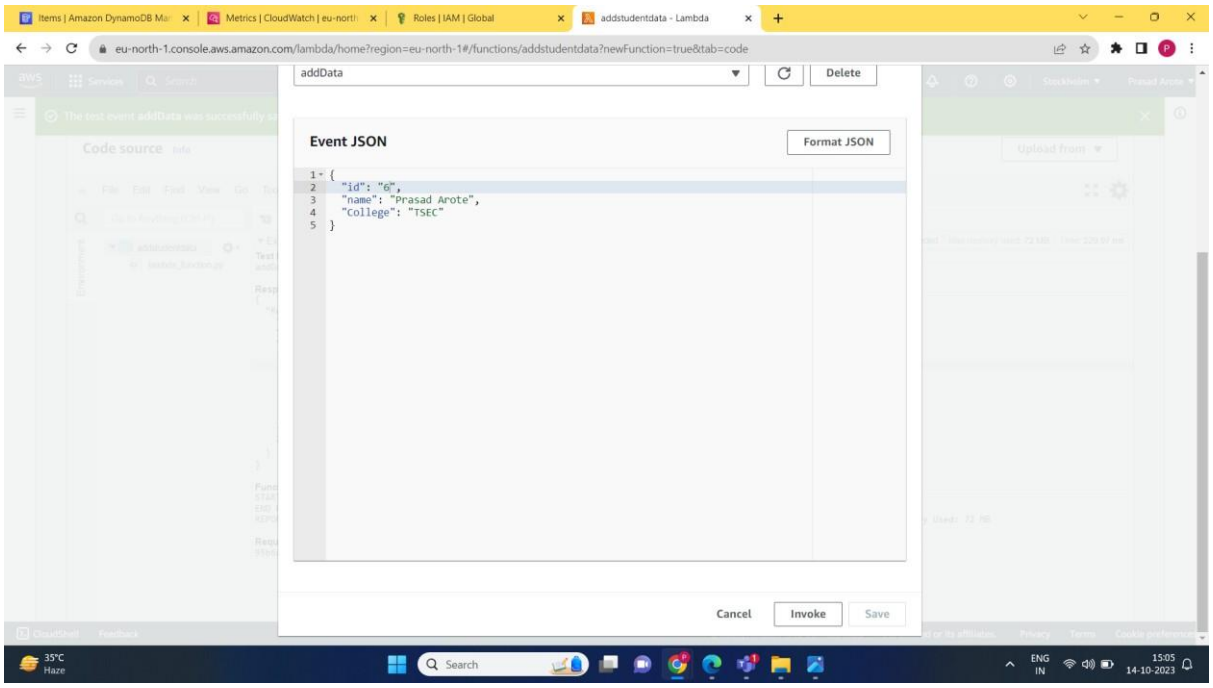


5. Write the following code

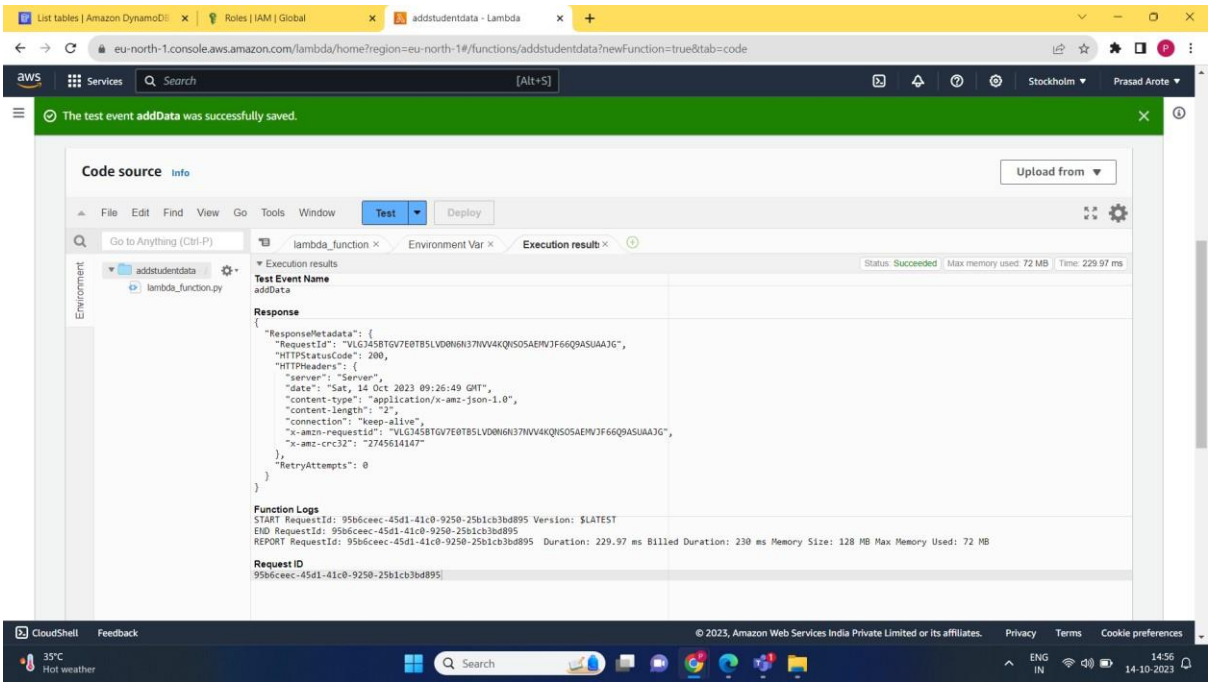


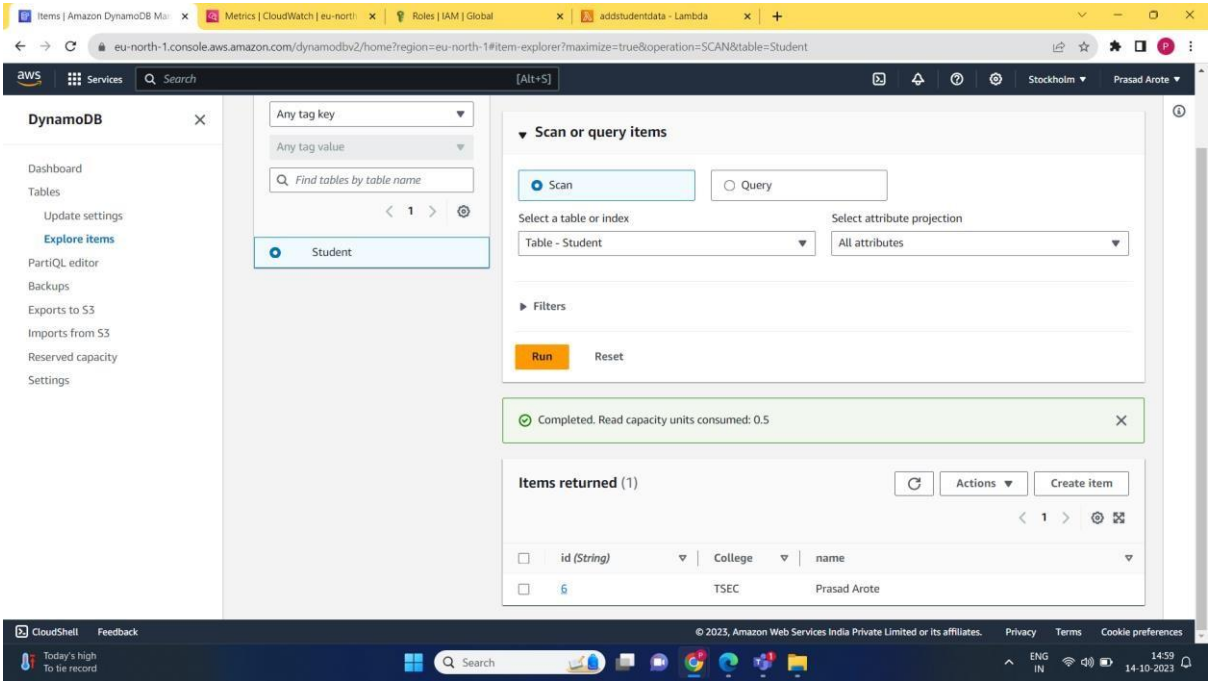
6. Configure test event and Save





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.

Roll No:10

Saish Bavalekar

ASSIGNMENT 9

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

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

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

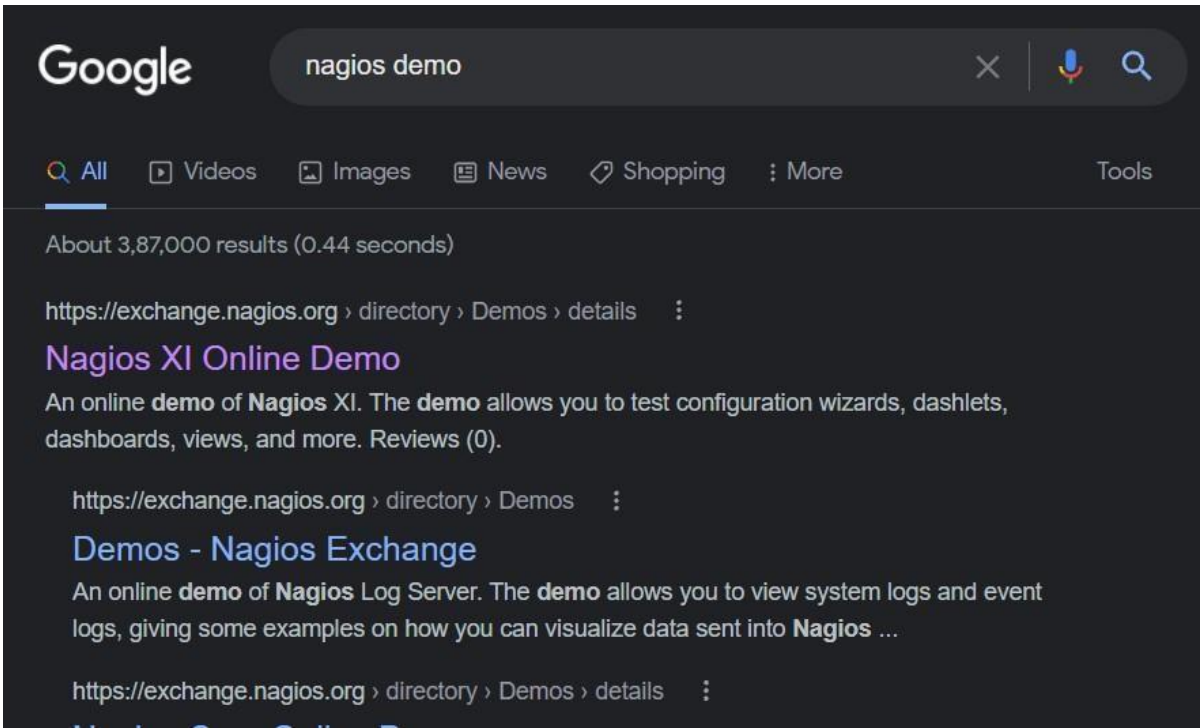
Theory:

What is Nagios and how it works?

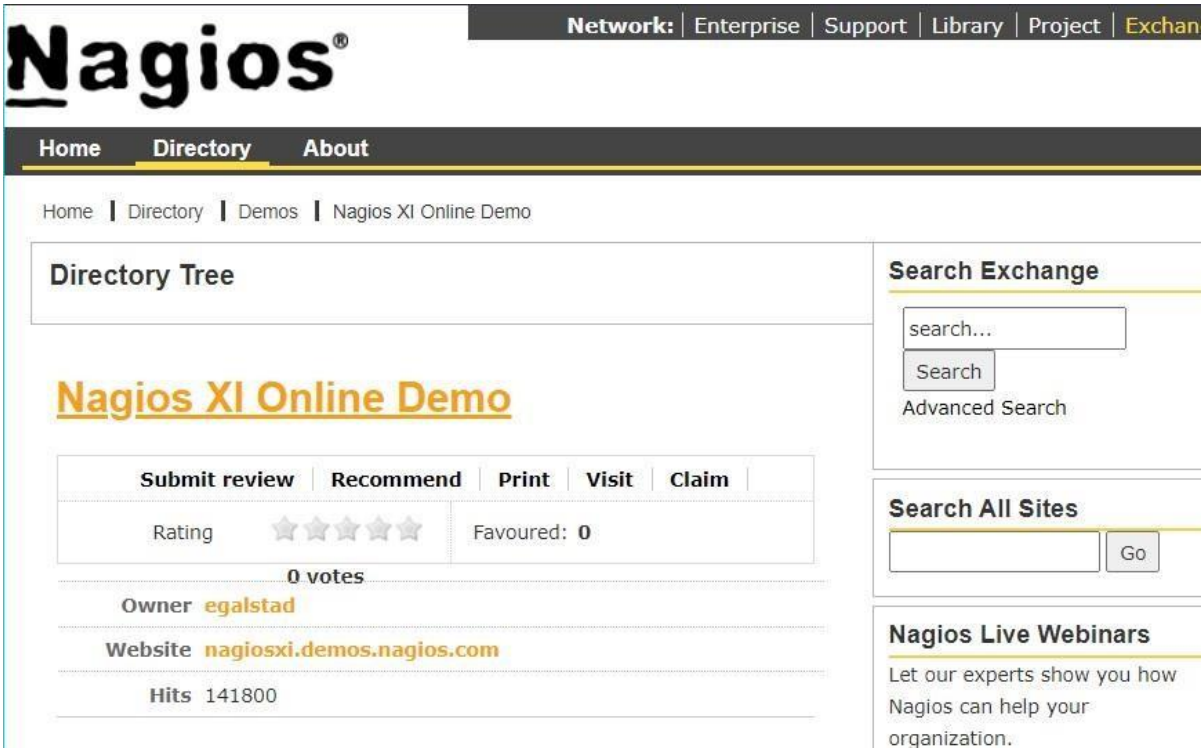
Nagios is an open source monitoring system for computer systems ..... Nagios software runs periodic checks on critical parameters of application, network and server resources. For example, Nagios can monitor memory usage, disk usage, microprocessor load, the number of currently running processes and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

Steps-

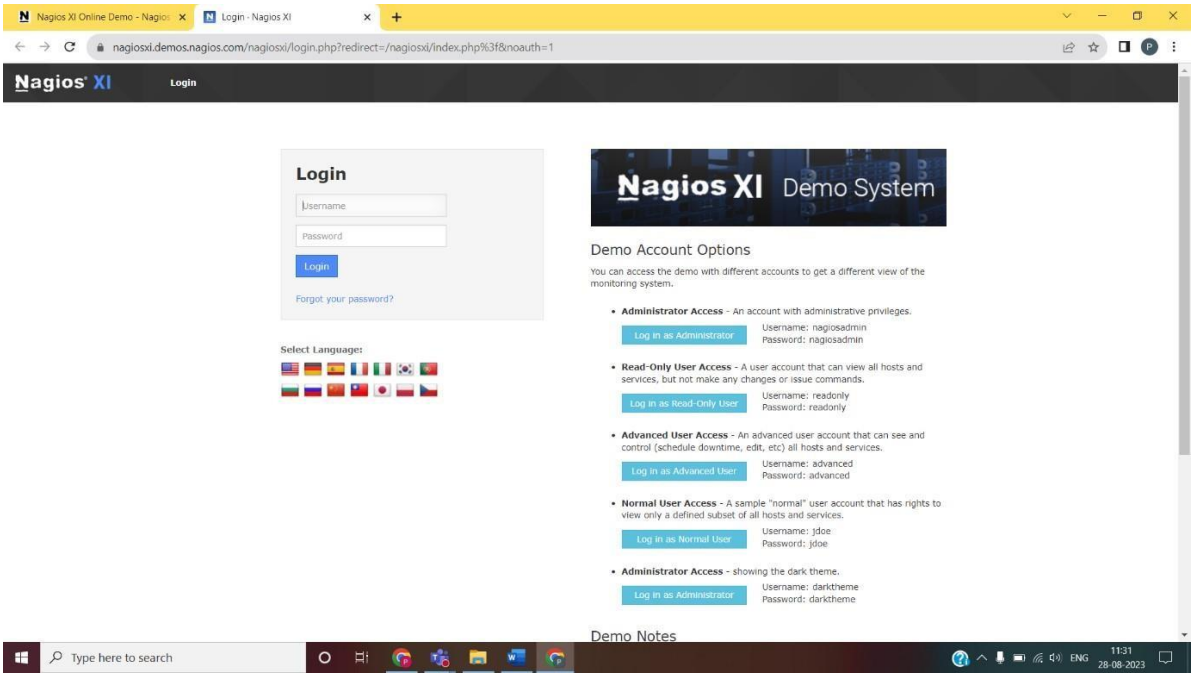
Go to google.com, Search Nagios Demo



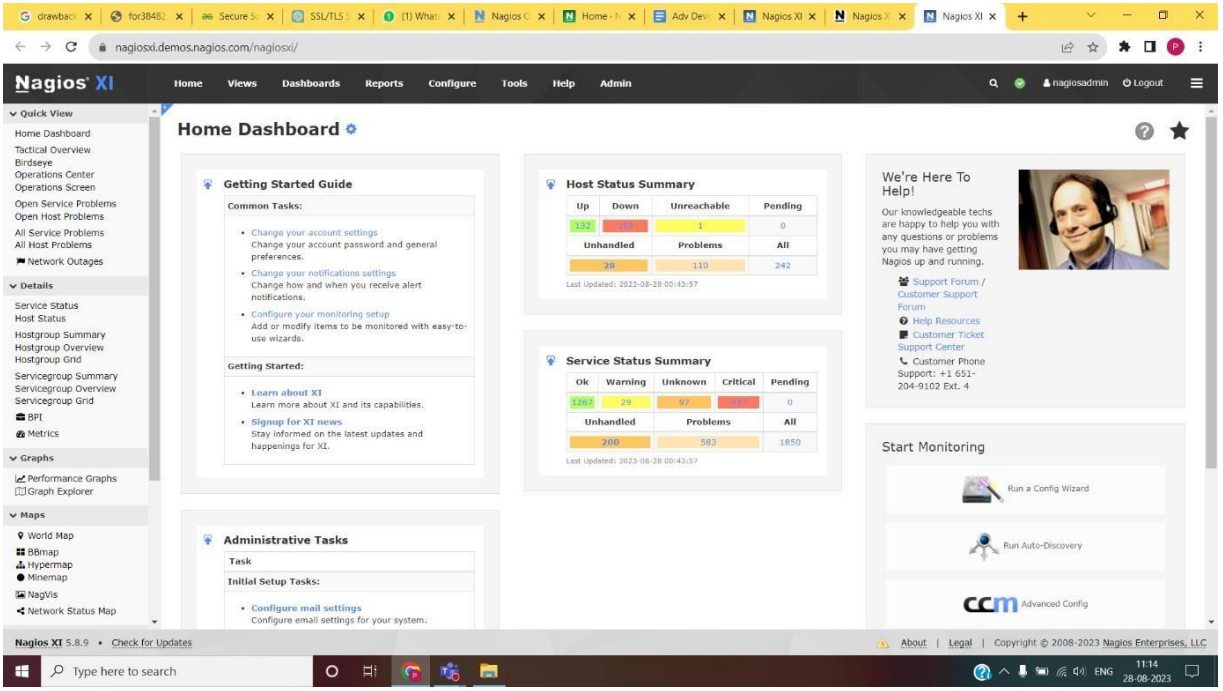
Now click on the website-



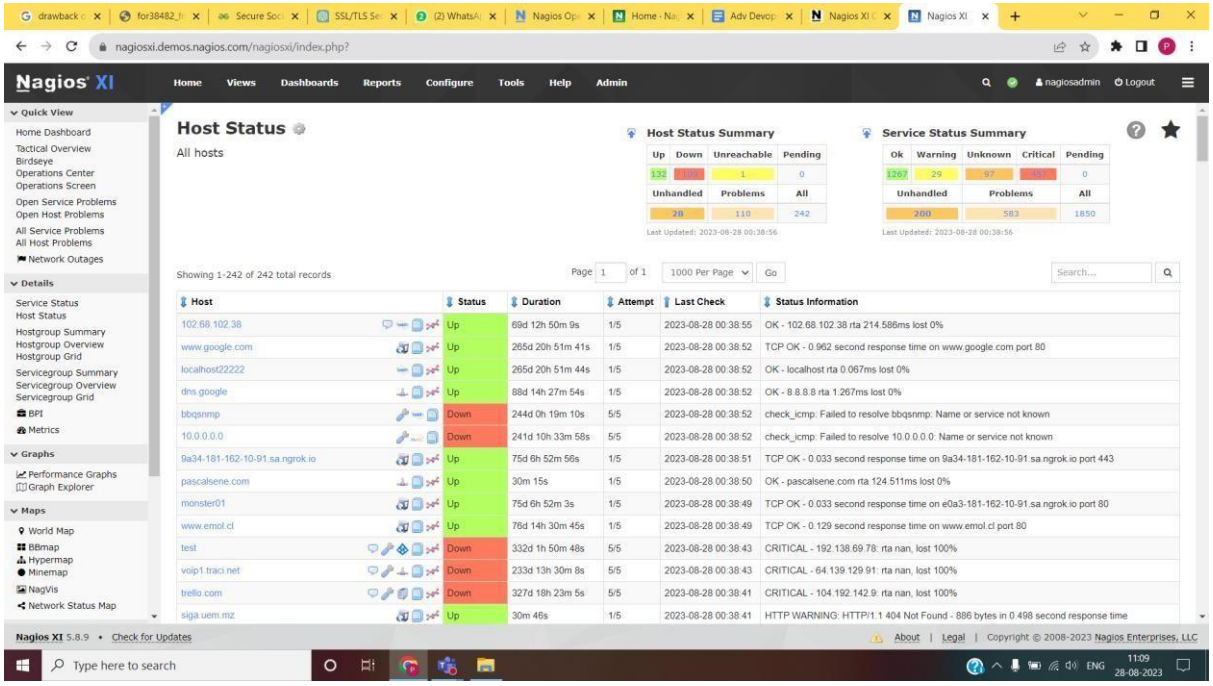
Now click on login as administrator



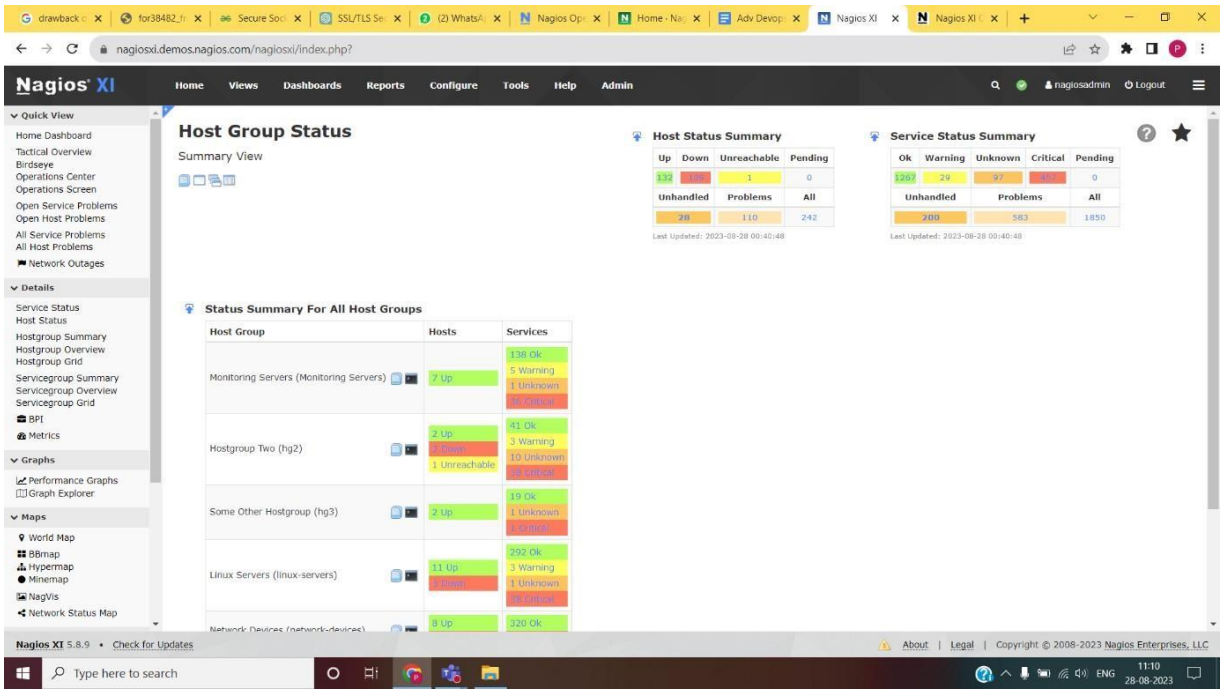
It will have interface like this



Now click on Host status-



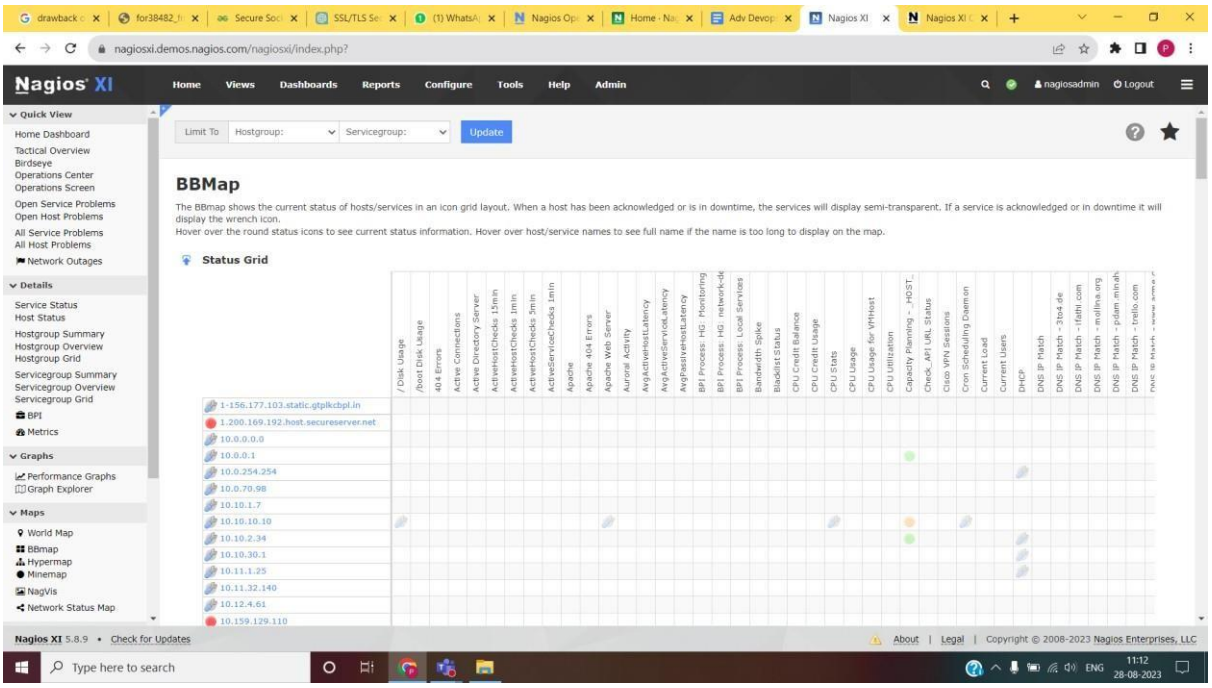
In the above image one can see Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.



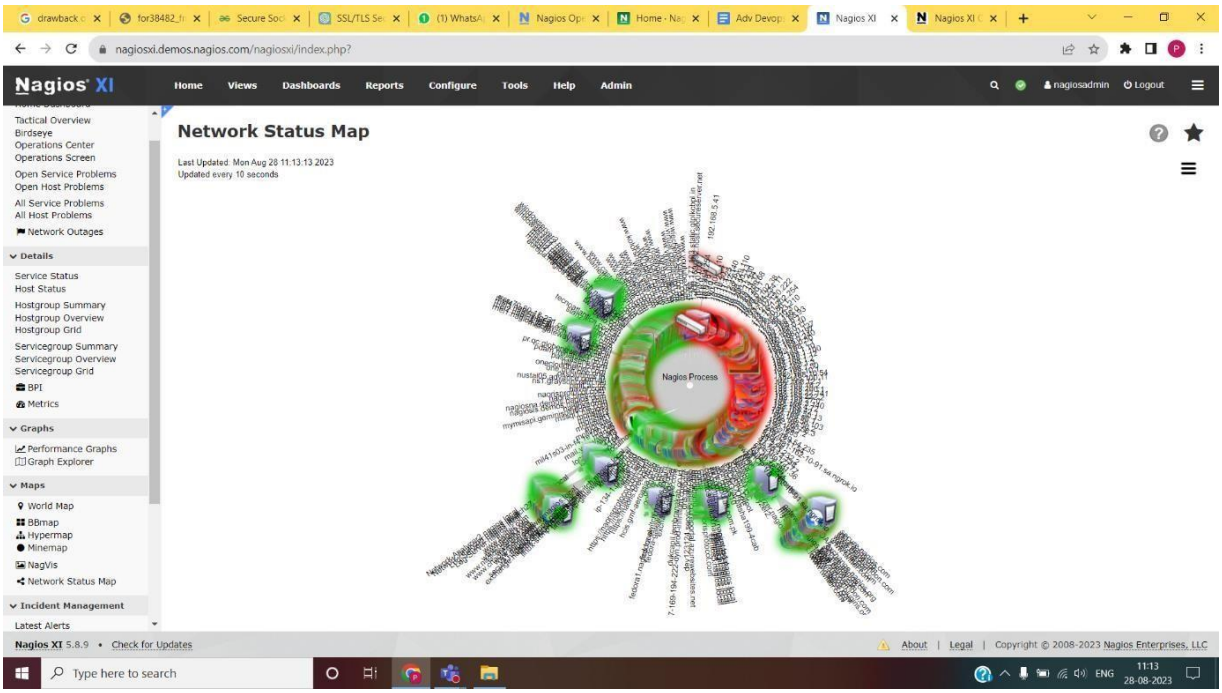
Here we can see Status Summary for All Host Groups

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.

Roll No:10

Saish Bavalekar

ASSIGNMENT 10

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

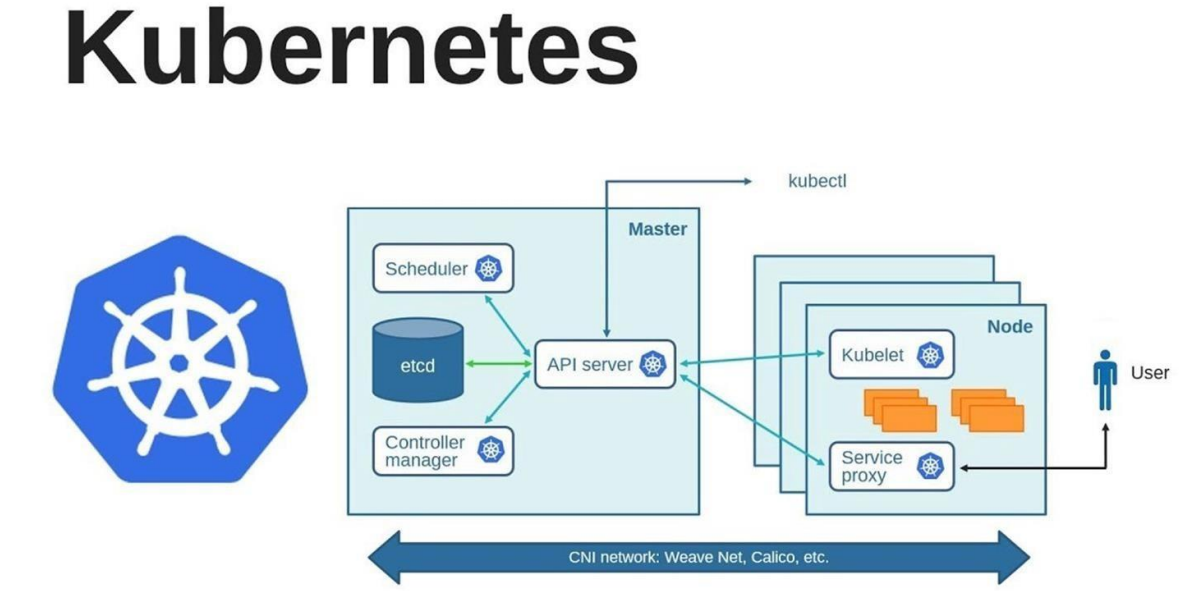
LO2. To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes.

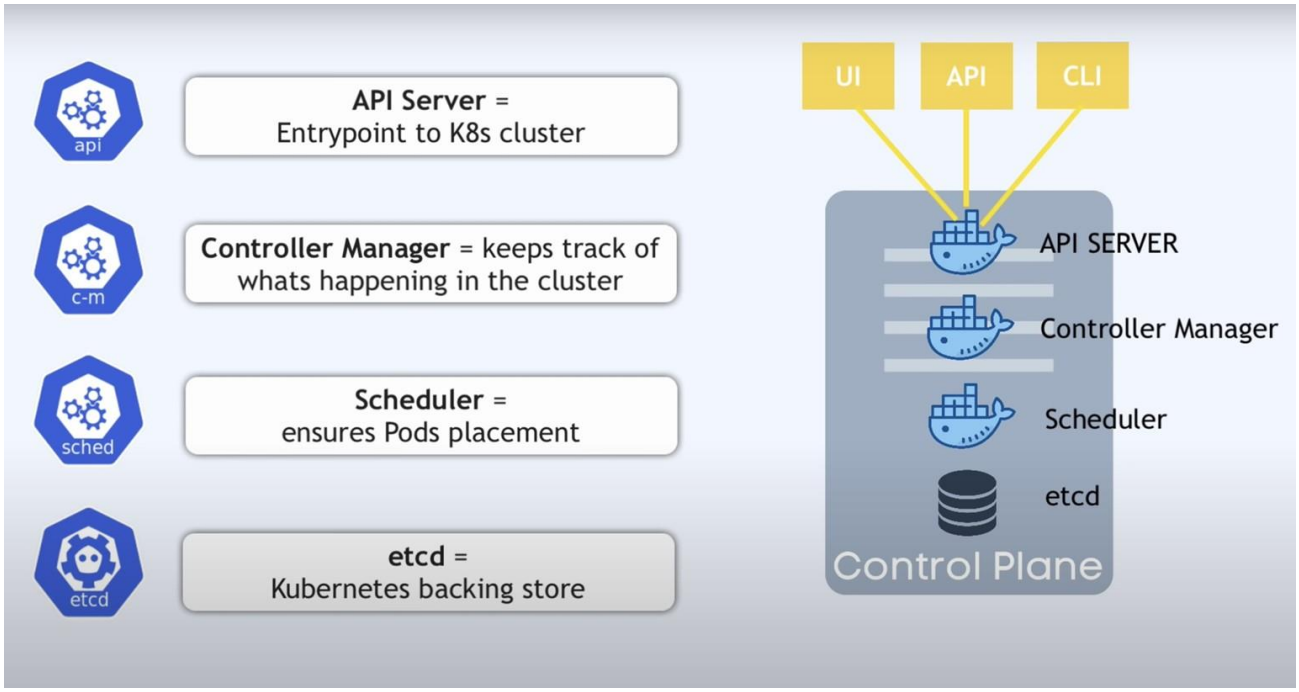
THEORY:

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

It schedules, runs, and manages isolated containers that are running on virtual/physical/cloud machines.
All top cloud providers support Kubernetes.
One popular name for Kubernetes is K8s.

ARCHITECTURE





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

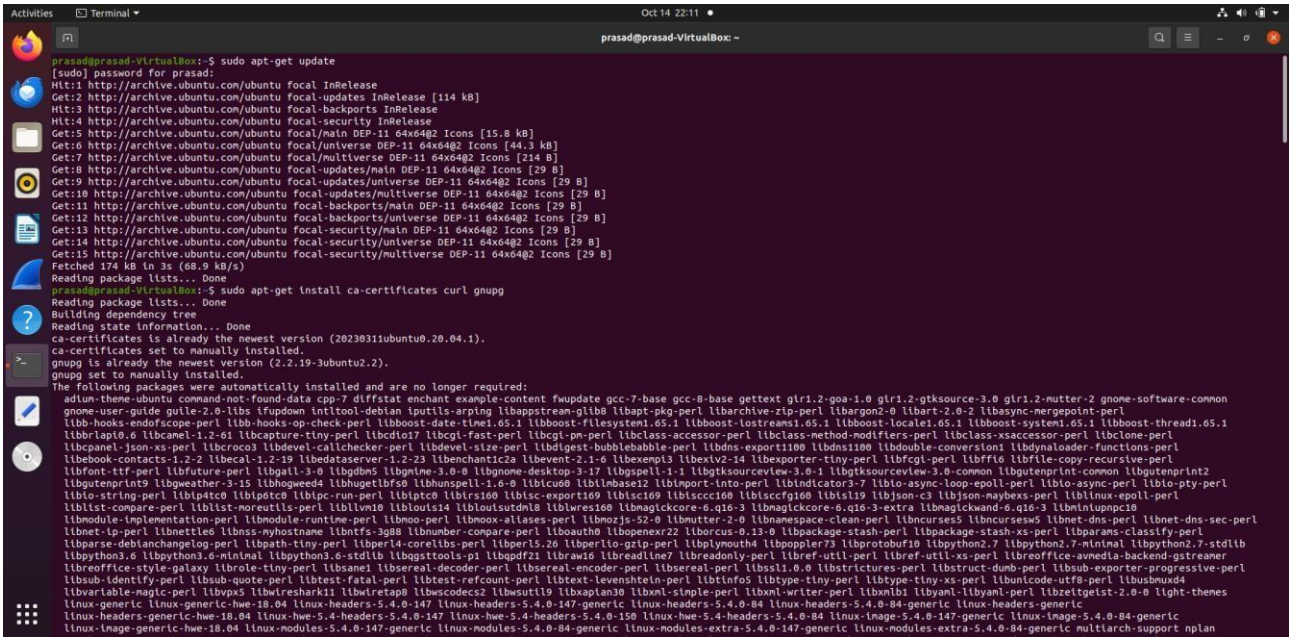
- Works with kubelet
- Pulling images
- Start/Stop Containers
- Exposing containers on ports specified in the manifest.

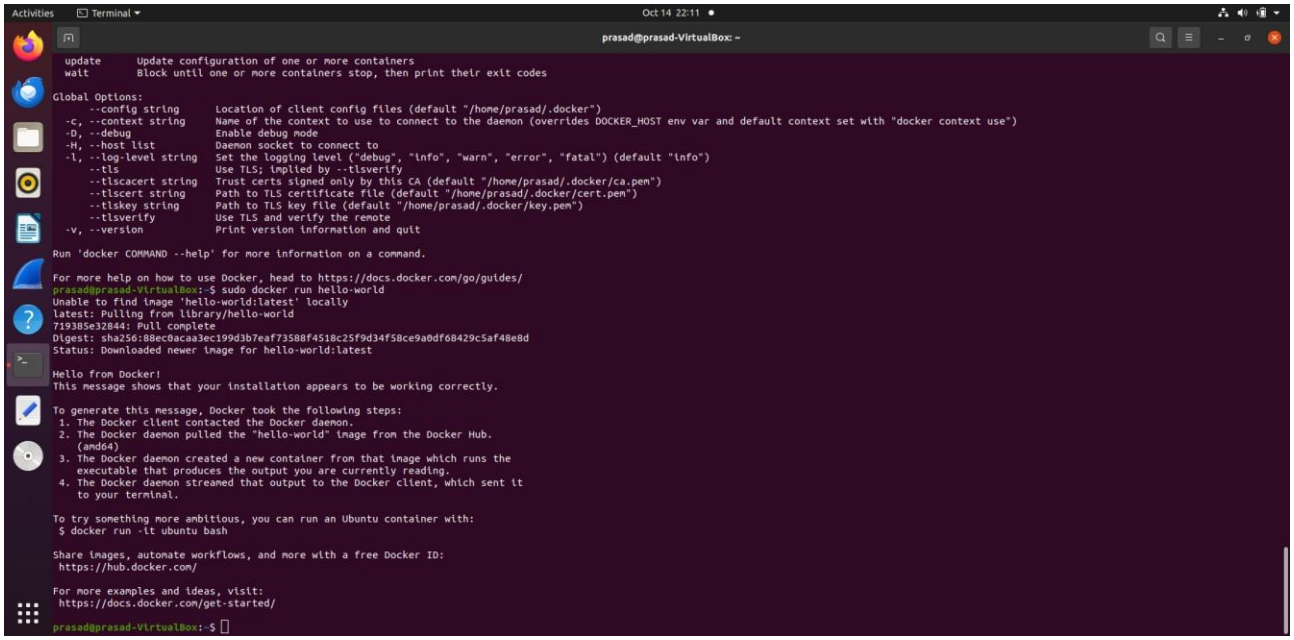
Kube-Proxy

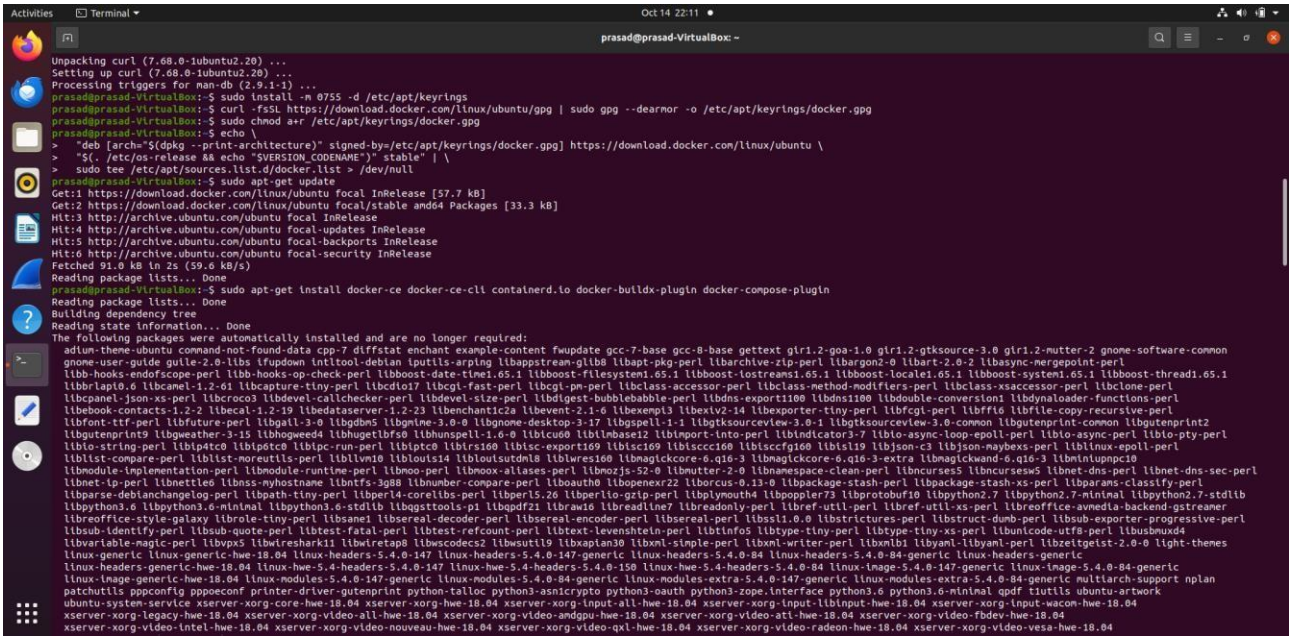
- Assign IP to each pod.
- It is required to assign IP addresses to Pods (dynamic)
- Kube-proxy runs on each node & this makes sure that each pod will get its unique IP Address.
- These 3 components collectively consist of ‘node’.

INSTALLATION:

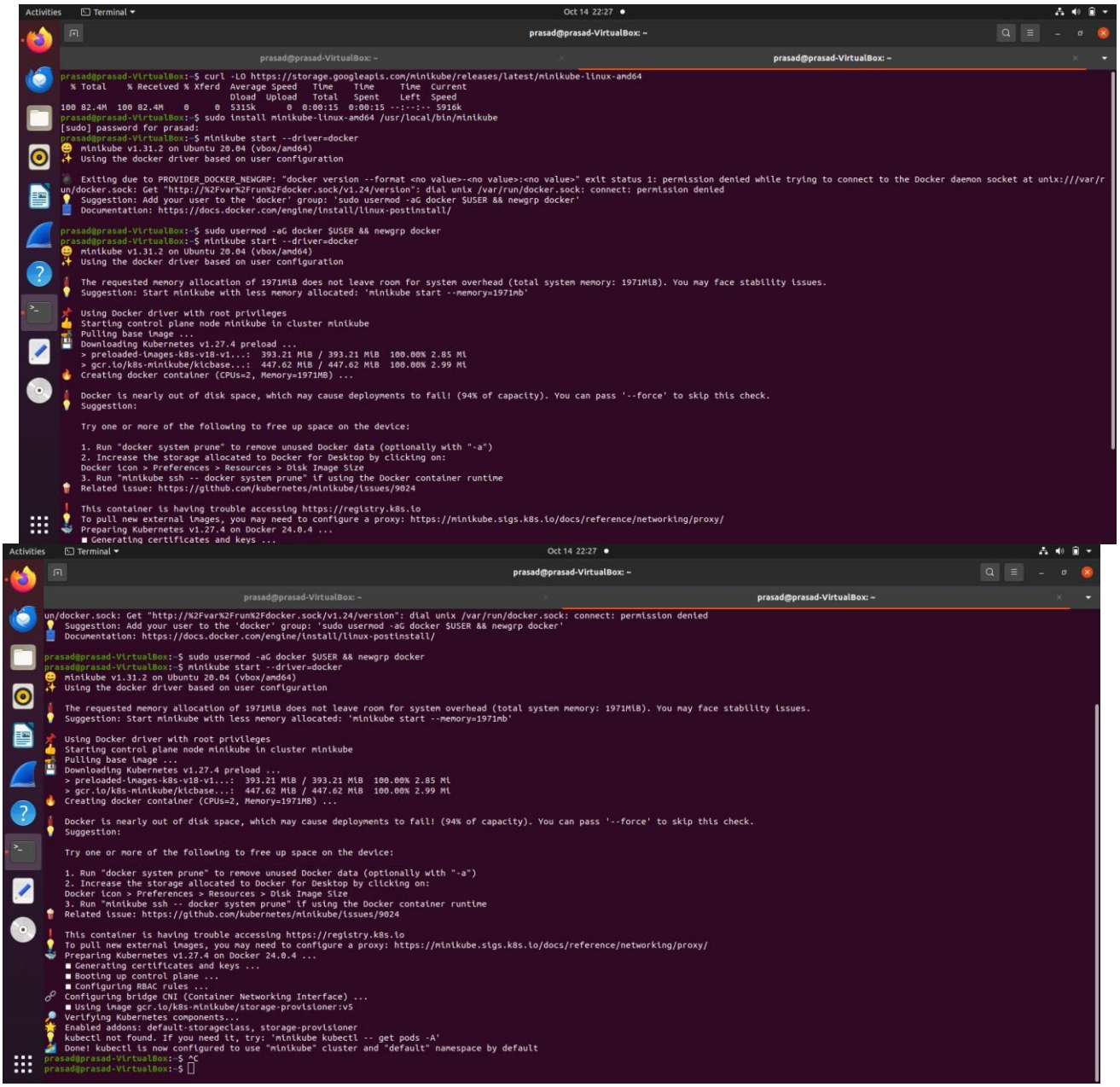
1. Install Docker



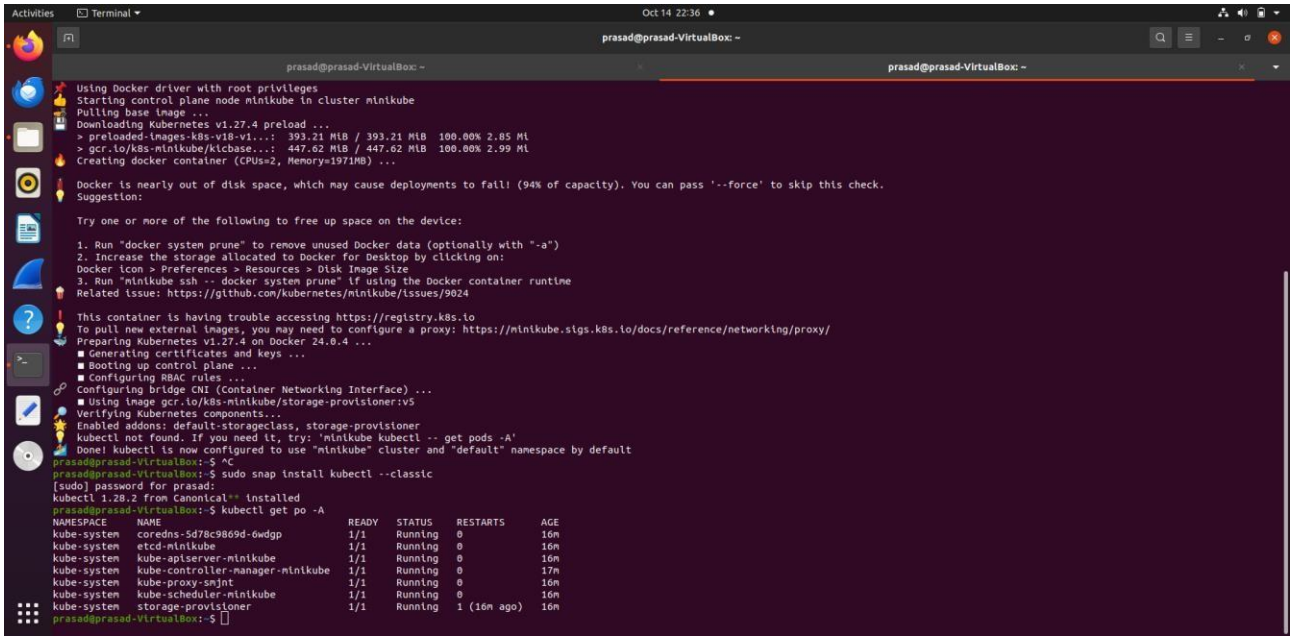




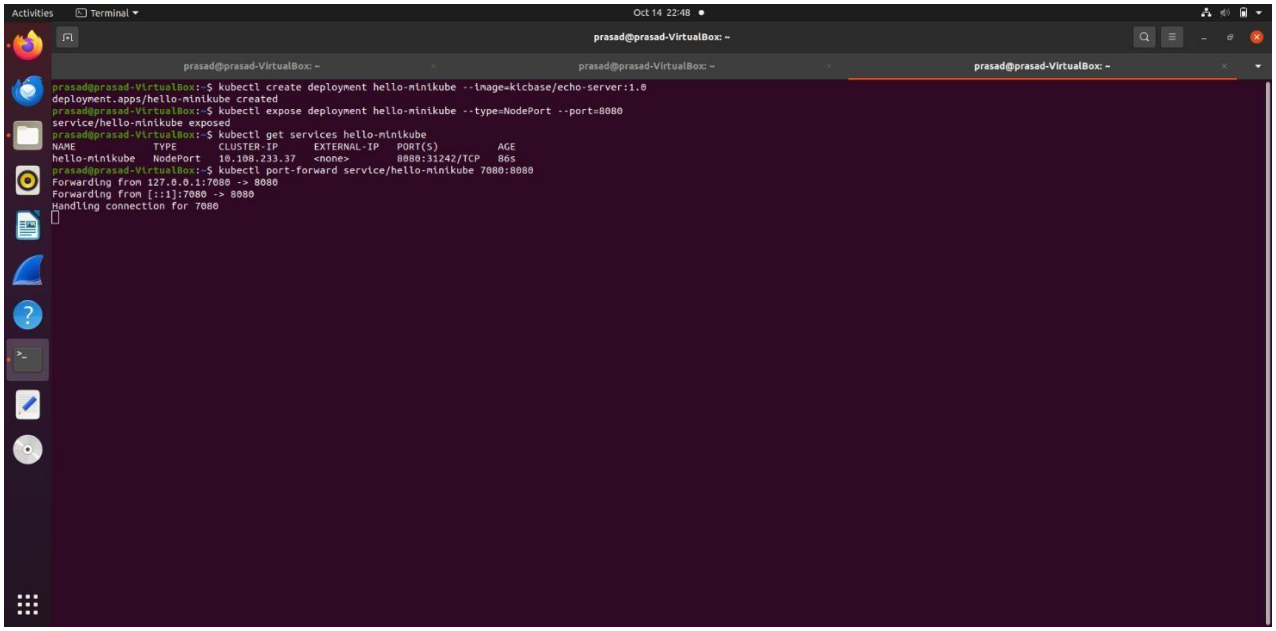
2. Install minikube using following commands

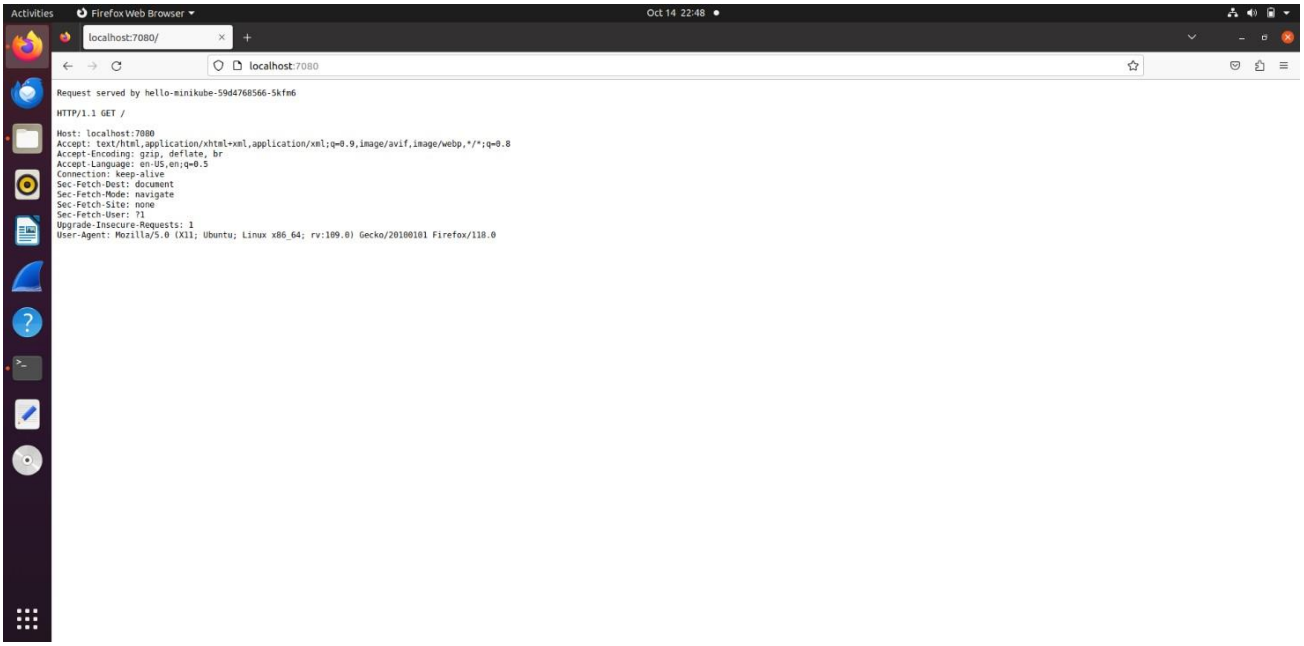


3. Install kubectl



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.

Roll No:10

Saish Bavalekar

Written Assignment 1

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

Ans:-

Securing a Kubernetes cluster is crucial to protect your applications, data, and infrastructure. Here are some best security measures to take while using Kubernetes:

Update Kubernetes Regularly:
Ensure you're using the latest stable version of Kubernetes with security patches and updates. Regular updates help mitigate known vulnerabilities.

Secure Kubernetes API Server:

Use strong authentication mechanisms like certificates or tokens.
Restrict access to the API server to authorized users and IP ranges.
Enable audit logs for monitoring API server access.
Implement RBAC (Role-Based Access Control):

Define specific roles and permissions for users and services to access resources.
Follow the principle of least privilege, granting minimal permissions necessary for tasks.
Secure Network Communication:

Encrypt communication using TLS for all network traffic within the cluster.
Use network policies to control traffic between pods and namespaces.
Protect etcd (Cluster Data Store):

Enable encryption for etcd to secure sensitive data stored in the cluster.
Implement access controls to limit access to etcd.
Secure Container Images:

Use trusted sources for container images.
Scan images for vulnerabilities before deploying them in the cluster.
Isolate Sensitive Workloads:

Isolate sensitive workloads in separate namespaces.
Use network policies to control access to sensitive workloads.

Implement Pod Security Policies (PSP):

Define and enforce policies that dictate the security settings for pods.
Control pod capabilities, privileged access, and host namespaces.
Secure Service Accounts:

Use service accounts judiciously and limit their permissions.
Regularly review and audit service account usage.
Monitor and Audit Cluster Activity:

Use Kubernetes audit logs to monitor and track activities within the cluster.
Utilize external monitoring tools for advanced threat detection.
Implement Network Policies:

Define and enforce network policies to control pod-to-pod communication.
Allow only necessary traffic and deny unnecessary communication.
Regular Security Audits and Penetration Testing:

Conduct security audits and penetration testing to identify vulnerabilities.
Address the identified vulnerabilities promptly.
Backup and Disaster Recovery:

Implement regular backups of critical data and configurations.
Have a well-defined disaster recovery plan and test it periodically.
Educate Users and Administrators:

Educate users and administrators about secure practices and potential security threats.
Promote a security-conscious culture within the organization.Engage
Security Experts:

Consider engaging security experts or consultants to perform security assessments and provide recommendations.
By following these security measures, you can enhance the overall security posture of your Kubernetes environment and protect your applications and data from potential security threats and vulnerabilities.

Q.2 What are three 3 security techniques that can be used to protect data?

Ans:

Protecting data is a critical aspect of ensuring information security. Here are three security techniques that can be used to protect data:

Encryption:
Encryption is a fundamental technique used to protect data by encoding it in a way that only authorized individuals or systems can access and interpret it. The data is converted into ciphertext using cryptographic algorithms and can only be decrypted using the appropriate encryption key. There are two main types of encryption:

Symmetric Encryption: Uses a single secret key for both encryption and decryption.
Asymmetric Encryption (Public-key Encryption): Uses a pair of keys, a public key for encryption and a private key for decryption.
Access Control and Authorization:
Access control and authorization mechanisms ensure that only authorized users have access to specific data and resources. This involves defining user roles, permissions, and privileges based on the principle of least privilege. Key techniques include:

Role-Based Access Control (RBAC): Assigns permissions based on user roles within an organization.
Attribute-Based Access Control (ABAC): Access is granted or denied based on attributes associated with users, the resource, and the environment.
Multi-Factor Authentication (MFA): Requires multiple forms of authentication before granting access, adding an extra layer of security.
Data Masking and Anonymization:
Data masking (also known as data obfuscation) involves hiding original data with modified content to protect sensitive information while maintaining its format. Anonymization goes a step further by irreversibly removing personally identifiable information (PII) from data sets. These techniques are used to share data for testing, analysis, or development purposes without exposing sensitive details. Common methods include:

Tokenization: Replaces sensitive data with unique tokens, retaining the format but making it meaningless.
Pseudonymization: Replaces identifiable information with pseudonyms, allowing for reversibility if needed for specific use cases.
These security techniques play a crucial role in safeguarding data from unauthorized access, breaches, and misuse, ensuring data confidentiality, integrity, and availability. Combining these techniques with a comprehensive data security strategy helps organizations maintain trust and compliance with regulatory requirements.

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

Ans:
In the context of advanced DevOps practices, exposing a service using Ingress in Kubernetes is a critical aspect of managing and deploying applications in a highly efficient, scalable, and secure manner. Here's how using Ingress fits into advanced DevOps practices:

1. **Advanced Traffic Routing and Load Balancing**:
   Ingress in Kubernetes allows for advanced traffic routing and load balancing. Advanced DevOps teams leverage Ingress rules to route traffic based on various parameters like host, URL paths, HTTP methods, headers, etc. This enables sophisticated traffic distribution strategies, essential for high-availability and performance-sensitive applications.

2. **Microservices and Service Mesh Integration**:
   In advanced DevOps setups, applications are often built using microservices architecture. Ingress controllers can be configured to work seamlessly with service meshes like Istio, enabling advanced traffic management, security policies, and canary deployments within the service mesh environment.

3. **Advanced Security and SSL Termination**:
   Ingress in advanced DevOps encompasses robust security practices. SSL termination and encryption at the Ingress level are common patterns to offload SSL processing from backend services, improving performance and simplifying certificate management.

4. **Integration with WAF (Web Application Firewall)**:
   Advanced DevOps teams often integrate Ingress with a Web Application Firewall (WAF) for enhanced security. The WAF can be configured to inspect and filter traffic at the Ingress point, providing an additional layer of protection against malicious attacks.

5. **Global Load Balancing and Multi-Cloud Deployments**:
   In multi-cloud or globally distributed architectures, advanced DevOps utilizes Ingress for global load balancing. Ingress can route traffic across different cloud providers or regions based on latency, proximity, or other defined rules, optimizing user experience.

6. **Automated Ingress Configuration with GitOps**:
   Advanced DevOps embraces GitOps principles to manage infrastructure as code. Ingress configurations can be version-controlled, reviewed, and deployed

automatically through GitOps pipelines, ensuring consistency, traceability, and auditability.

7. **Dynamic Configuration Updates and Autoscaling**:
   Advanced setups often leverage Ingress controllers that support dynamic updates to configuration. Ingress rules can be updated on the fly, allowing for autoscaling and rapid adjustments to handle traffic spikes and changes in application demands.

8. **Advanced Monitoring and Analytics Integration**:
   Ingress data can be integrated into advanced monitoring and analytics platforms. Metrics and logs from the Ingress controllers provide insights into traffic patterns, performance, errors, and user behavior, enabling data-driven optimizations and troubleshooting.

In conclusion, using Ingress in Kubernetes within an advanced DevOps context goes beyond simple traffic routing. It involves advanced traffic management, security enhancements, integration with various tools and services, dynamic configurations, and the seamless orchestration of microservices in a modern, cloud-native, and highly automated DevOps environment.

Q.4 Which service protocols does Kubernetes ingress expose

Ans:

Kubernetes Ingress, by default, is primarily designed to expose HTTP and HTTPS services. It provides a way to route and manage HTTP/HTTPS traffic to services within the Kubernetes cluster based on defined rules. However, Ingress can be extended to support additional protocols beyond HTTP/HTTPS using specialized Ingress controllers and annotations.

The primary protocols exposed by Kubernetes Ingress are:

HTTP (HyperText Transfer Protocol):

Ingress is most commonly used to expose HTTP services. It allows for routing and load balancing of HTTP traffic based on defined rules and paths.
HTTPS (HTTP Secure):

Ingress can also expose HTTPS services, providing a secure way to route and manage HTTPS traffic. SSL/TLS termination can be performed at the Ingress level for HTTPS traffic.
In addition to HTTP and HTTPS, Kubernetes Ingress can be extended to support protocols such as:

TCP (Transmission Control Protocol):

Ingress can be configured to expose TCP-based services. This is useful for non-HTTP applications or services that use TCP for communication.
UDP (User Datagram Protocol):

Similarly, Ingress can be configured to expose UDP-based services. UDP is often used for applications that need a lightweight and faster communication protocol.
gRPC (gRPC Remote Procedure Calls):

Ingress can be configured to handle gRPC traffic, a high-performance, open-source and universal remote procedure call (RPC) framework.
WebSocket:

Ingress extensions can manage WebSocket protocols, which are essential for real-time applications and two-way communication between a client and a server.
To support these additional protocols, specialized Ingress controllers need to be used along with appropriate annotations and configurations. These controllers extend the functionality of Kubernetes Ingress and enable routing and load balancing for a broader range of protocols beyond HTTP/HTTPS. Always refer to the specific Ingress controller's documentation to confirm the supported protocols and configurations.

Roll No:10

Saish Bavalekar

Written Assignment 2

Q.1 How to deploy Lambda function on AWS?

Ans:

Deploying a Lambda function on AWS involves several steps, including creating the function, configuring its settings, and setting up triggers. Here's a step-by-step guide on how to deploy a Lambda function on AWS:

Sign in to AWS Console:
Log in to your AWS account and access the AWS Management Console.

Open Lambda service:
Navigate to the AWS Lambda service from the AWS Management Console.

Create a Lambda function:
Click on the "Create function" button.

Configure the function:

Choose an authoring option: You can either author the code inline or upload a .zip file containing the code.
Function name: Enter a unique name for your Lambda function.
Runtime: Choose the runtime that supports your code (e.g., Node.js, Python, Java, etc.).
Role: Choose an existing role or create a new one with appropriate permissions.
Function code:

If you chose inline code, you can edit the code directly in the AWS Management Console.
If you uploaded a .zip file, upload your deployment package containing the code.
Environment variables (optional):
You can set environment variables for your Lambda function.

Execution role:

Choose an existing role or create a new one with the necessary permissions for your function.

Advanced settings (optional):
Configure additional settings like memory, timeout, networking, and concurrency.

Triggers (optional):
Configure event sources that trigger your Lambda function (e.g., API Gateway, S3, SNS, etc.).

Click "Create function":
Once you've configured all the necessary settings, click the "Create function" button.

Lambda function is now deployed and ready to be triggered based on the configured event sources. We can test our function within the AWS Management Console or trigger it using the configured triggers.

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

Ans:

AWS Lambda offers several deployment options to suit various use cases and preferences. Here are the primary deployment options for AWS Lambda:

Manual Deployment:
Manually deploy a Lambda function by creating it through the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. You provide the function code and configuration settings during this process.

AWS Management Console:
Create, configure, and deploy Lambda functions using the AWS Management Console. This web-based interface allows for a visual and interactive way to manage your functions.

AWS Command Line Interface (CLI):
Use the AWS CLI to package and deploy Lambda functions. The CLI allows for scripting and automation of deployment processes.

AWS CloudFormation:
Define Lambda functions and related AWS resources in AWS CloudFormation templates. CloudFormation automates the deployment and management of these resources, enabling infrastructure as code (IaC) practices.

AWS SAM (Serverless Application Model):

AWS SAM is an open-source framework that extends AWS CloudFormation to simplify the deployment of serverless applications. It provides a streamlined way to define serverless applications, including Lambda functions, APIs, and more.

AWS Serverless Application Repository:
Deploy pre-built serverless applications and components from the AWS Serverless Application Repository. This allows you to use and share existing serverless applications easily.

AWS CodeDeploy:
Integrate AWS Lambda deployments with AWS CodeDeploy, a service that automates application deployments to various compute services, including Lambda. CodeDeploy can help manage updates and rollbacks for Lambda functions.

AWS SAM CLI:
Use the AWS SAM CLI to build, test, and deploy serverless applications defined using AWS SAM. The CLI provides local testing capabilities and simplifies the deployment process.

Third-Party Tools and Frameworks:
Utilize third-party tools and frameworks, such as Serverless Framework, Terraform, and others, to deploy and manage Lambda functions. These tools often provide additional features and flexibility for deployment and management.

Each deployment option has its advantages and is suited for different scenarios based on factors like automation needs, complexity, team collaboration, and integration with existing workflows. Choose the option that aligns best with your specific use case and deployment requirements.

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

Ans:

In the context of AWS (Amazon Web Services), there are three primary deployment modes commonly referred to when deploying applications or services:

Manual Deployment:
Manual deployment involves direct and hands-on configuration and setup of resources using AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. It allows users to define and configure each component individually, making it suitable for small-scale or simple deployments.

Automated Deployment:

Automated deployment involves using automation tools and scripts to manage the deployment process. This can include using AWS services like AWS CloudFormation, AWS SAM (Serverless Application Model), AWS CodeDeploy, or third-party tools like Terraform. Automation streamlines and accelerates the deployment process, making it more efficient and repeatable.

Serverless Deployment:
Serverless deployment, often associated with AWS Lambda, involves deploying applications or functions without managing the underlying infrastructure. AWS Lambda automatically handles scaling, monitoring, and managing the compute resources needed to run the application. Serverless architectures abstract away server management tasks, allowing developers to focus solely on writing code.

These deployment modes offer varying levels of control, automation, and abstraction, catering to different use cases and preferences. Developers and organizations choose the appropriate deployment mode based on their specific requirements, such as application complexity, scalability needs, automation goals, and the level of control they desire over the deployment process.

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

Ans:

AWS Lambda is composed of three primary components that work together to enable serverless compute capabilities:

Function Code:
The function code is the actual program or script that you want AWS Lambda to execute in response to an event. It can be written in supported programming languages, including Node.js, Python, Java, C#, Go, Ruby, and PowerShell Core. You can provide the code directly within the AWS Management Console (inline code) or package it as a .zip file and upload it to AWS Lambda.

Event Sources:
Event sources are the triggers or events that invoke your AWS Lambda function. Lambda functions can be triggered by various AWS services and event sources, including Amazon S3 bucket events, Amazon DynamoDB table updates, Amazon SNS notifications, API Gateway requests, AWS CloudFormation stack updates, and more. Event sources determine when and how your Lambda function is executed.

Execution Role:
An execution role (IAM role) defines the permissions that AWS Lambda functions need to interact with other AWS services. This role grants necessary permissions for

the function to access resources, such as reading from an S3 bucket, writing to a DynamoDB table, or publishing to an SNS topic. AWS Lambda assumes this role when executing the function, allowing it to access the specified resources securely.

These three components form the foundation of AWS Lambda, enabling developers to create, configure, and run serverless functions that respond to events from various sources within the AWS ecosystem.