Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ Altaf Alam

of I7 Department, Semester V with

Roll No. 02 has completed a course of the necessary

experiments in the subject Advance Devops Lab under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 20/10/23

Principal

THADOMAL SHAHANI TSEC ENGINEERING SOLLEGE

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 and Deploy on S3.		24/7/23	
3.	Creating on S3 bucket using AWS Service and uploading a file into it.		31/7/23	
4.	To understand the installation process of Terraform		7/8/23	
5.	To understand the concept of terraform and use it to create and instance using limit, plan, apply & destroy command		21/8/23	
6.	To understand the concept of Sonar cube & server scanner		28/8/23	
7.	To learn how to use lambda to run a simple program from S3 bucket		4/09/23	
8.	To learn how to use lambda to find the content type of object uploaded in S3 bucket.		11/09/23	
9.	To understand and install Nagios monitor host using nagios		18/09/23	
10.	To complete study of Kubernates		25/10/23	
11.	Theory Assignment 1			
12.	Theory Assignment 2			

Name : Altaf Alam

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

Assignment No 1

AIM - To make a EC2 Machine in AWS.

THEORY -

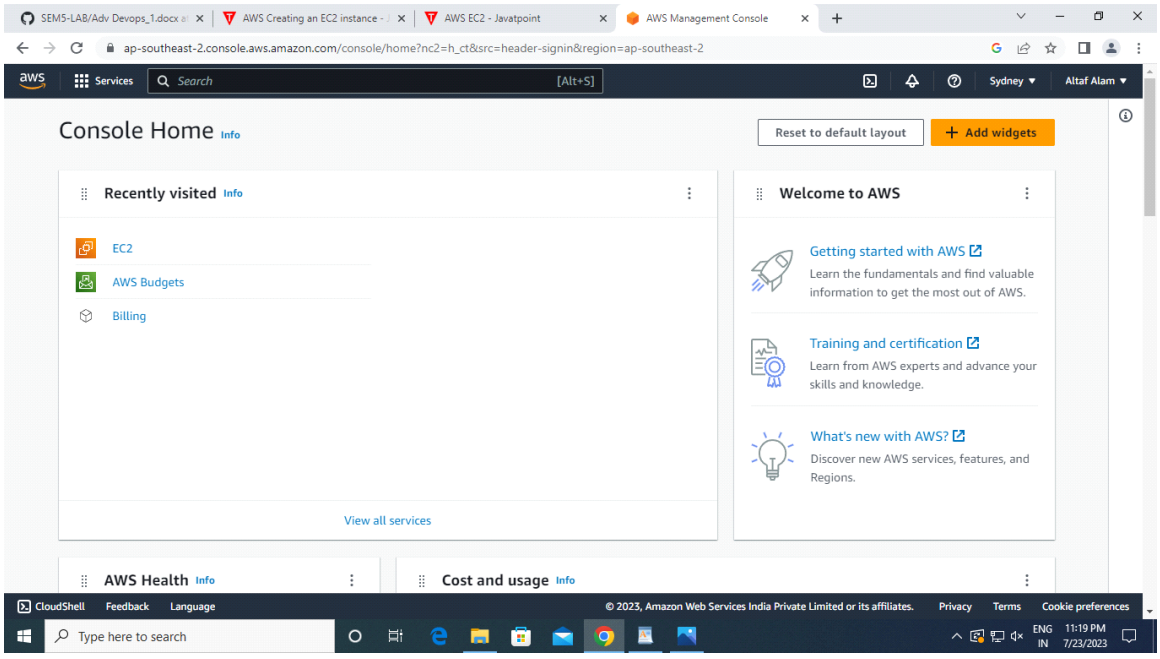
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time required to obtain and boot new user instances to minutes rather than in older days, if you need a server then you had to put a purchase order, and cabling is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

STEPS -

LOGIN TO AWS

ACCOUNT, THEN SEACH

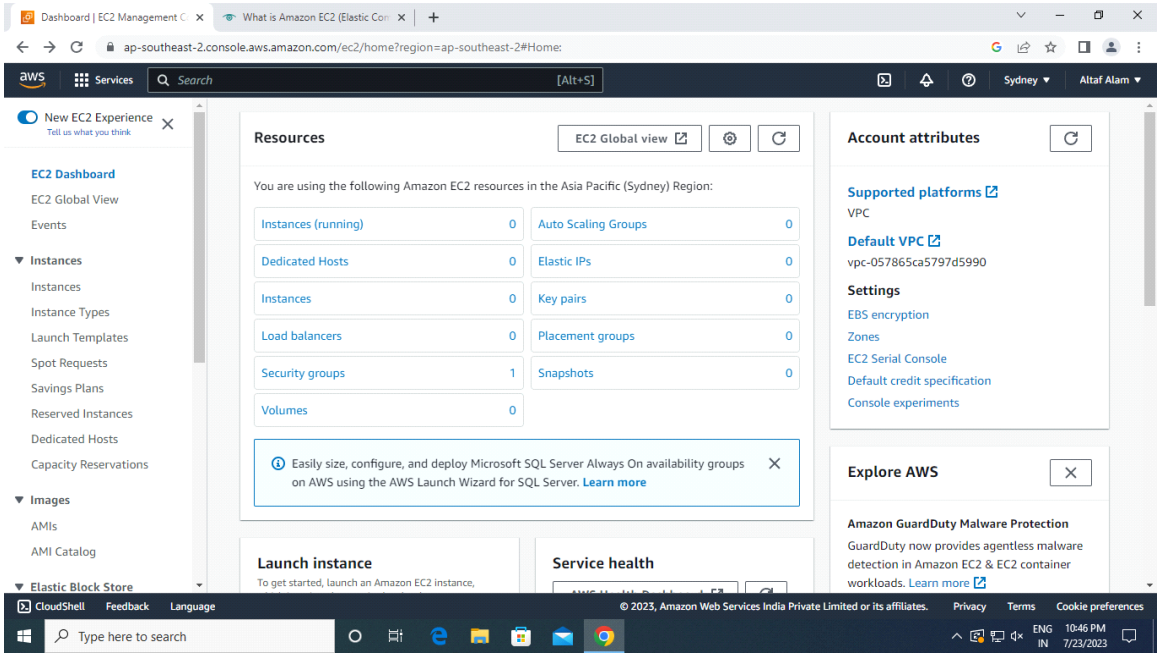
EC2.



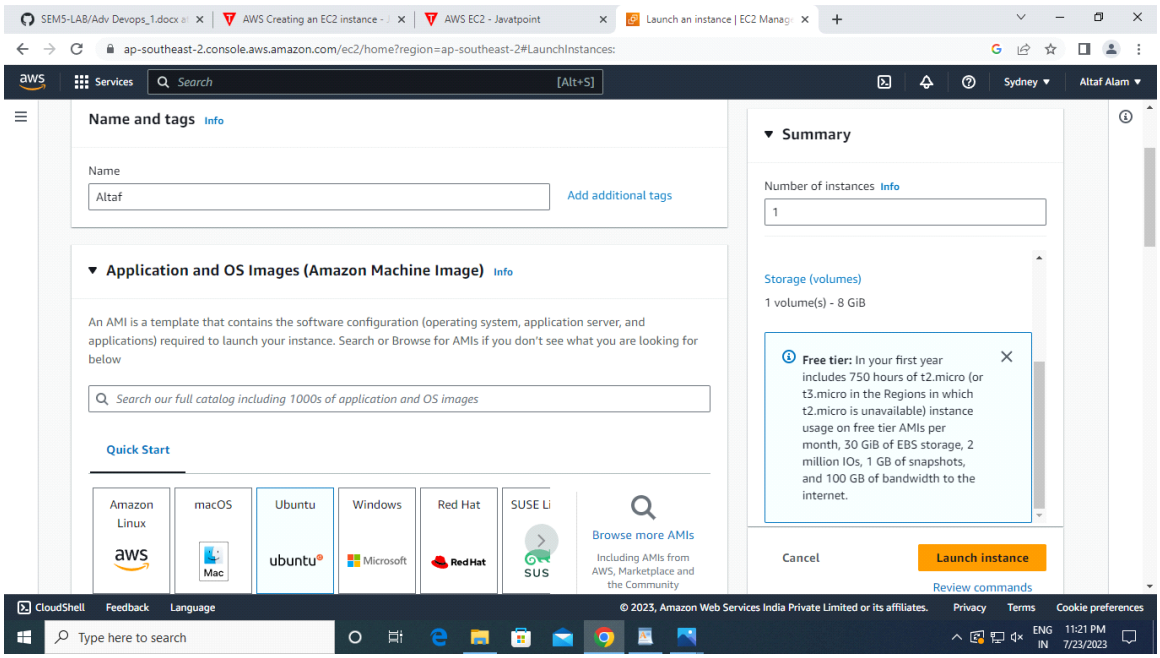
Name : Altaf Alam

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

NOW CLICK ON LAUNCH / CREATE NEW INSTANCES.



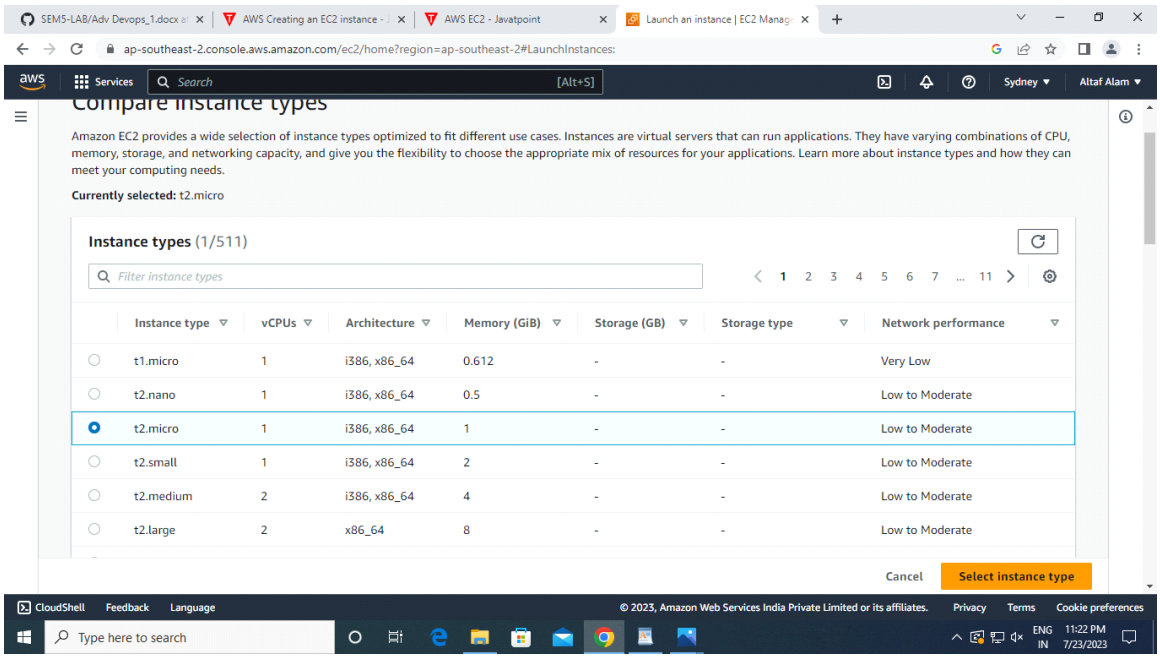
Choose any machine you want to create here I am creating UBUNTU(free tier).



Name : Altaf Alam

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

Click on T2 micro (free tier one)



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

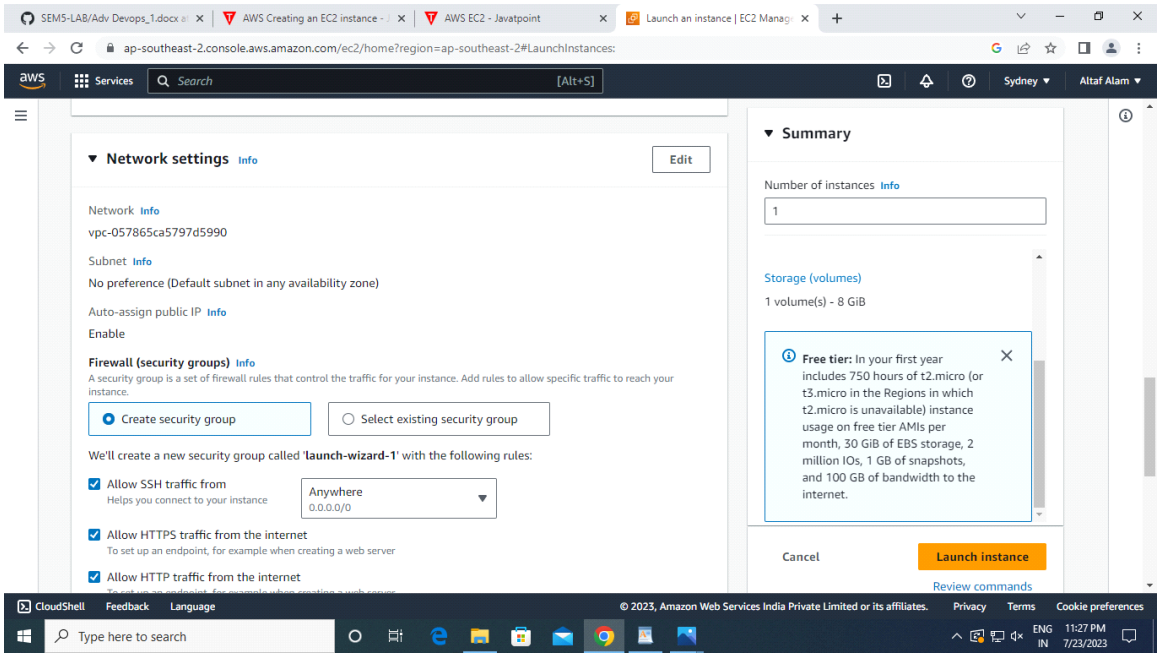
Now add security group ALL TRAFFIC ,

PROTOCOL – ALL,

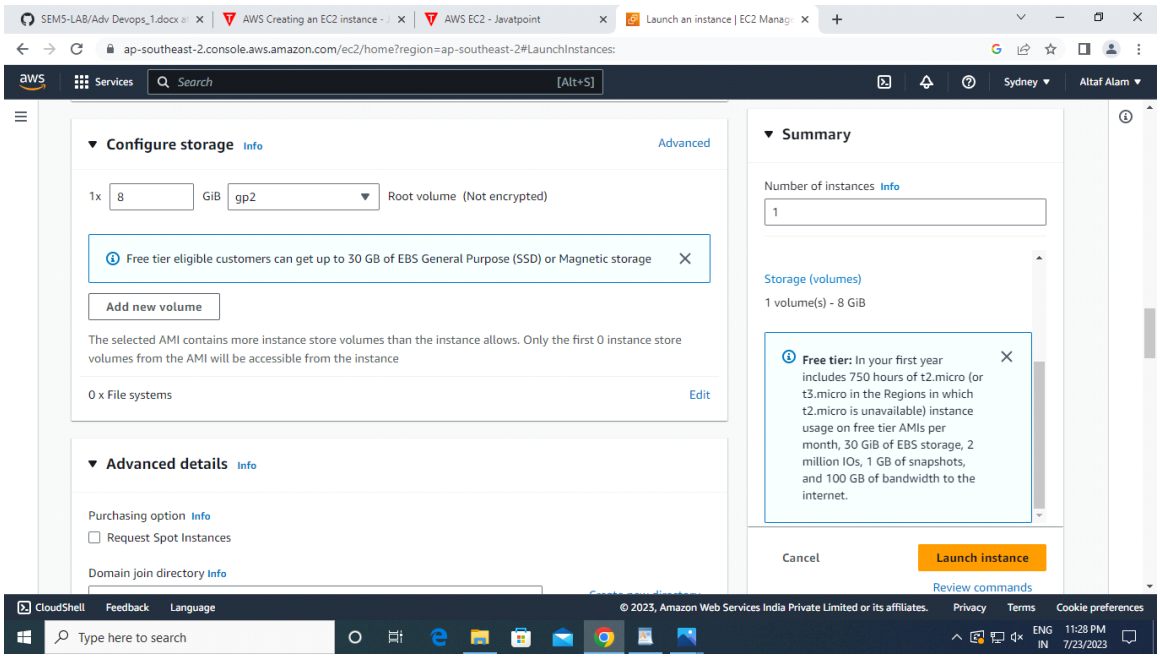
SOURCE- ANYWHERE.

Name : Altaf Alam

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



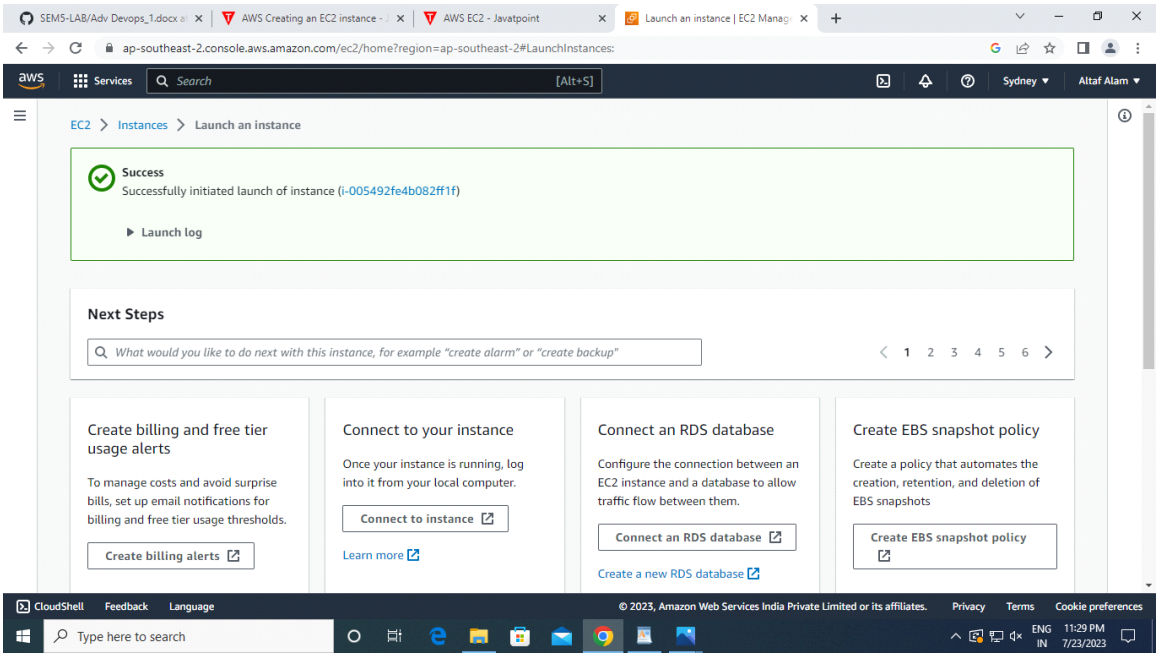
Configure Storage



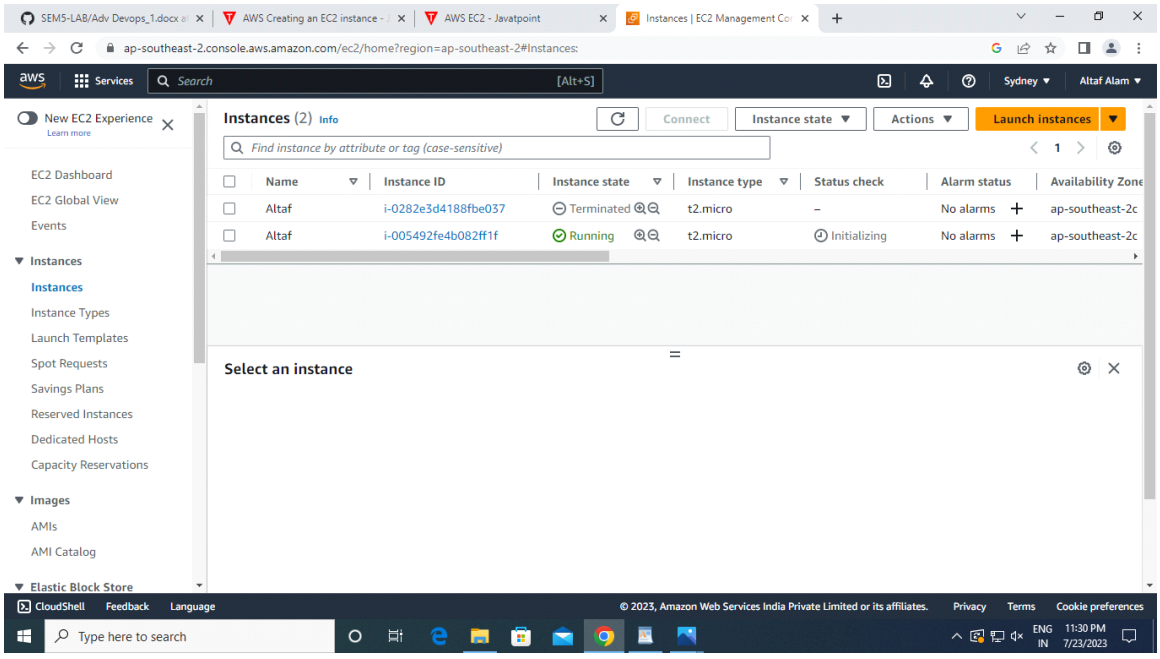
NOW WAIT TILL THE STATUS CHECK IS 2/2 and Instance is running.

Name : Altaf Alam

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



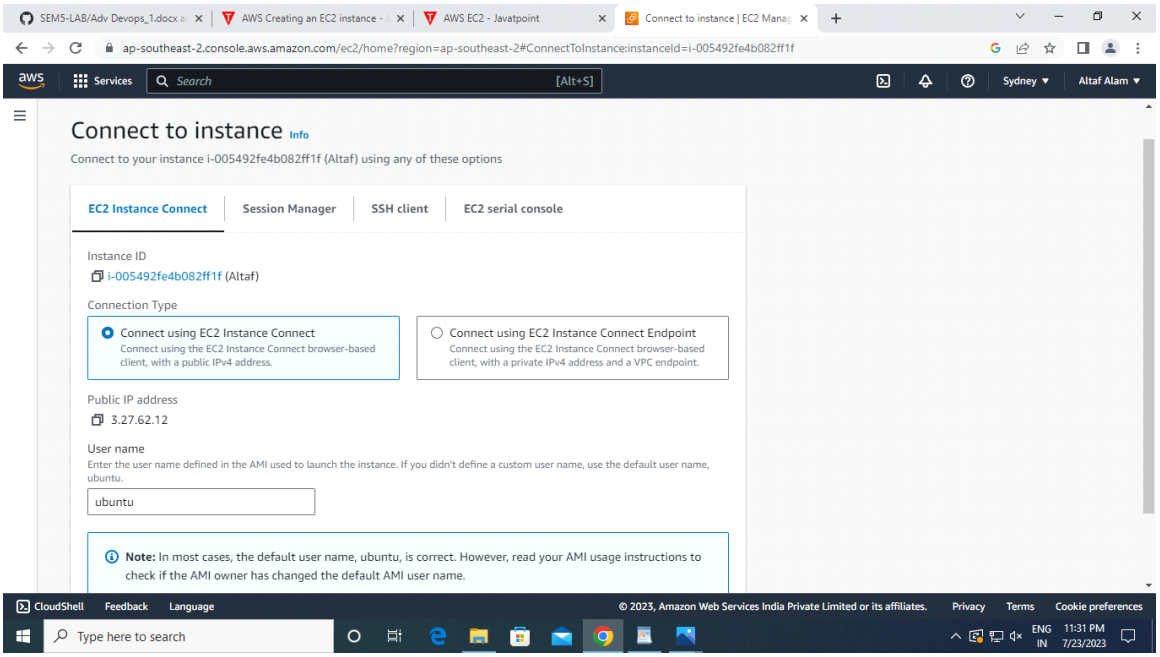
Once Check is complete click on launch instances.



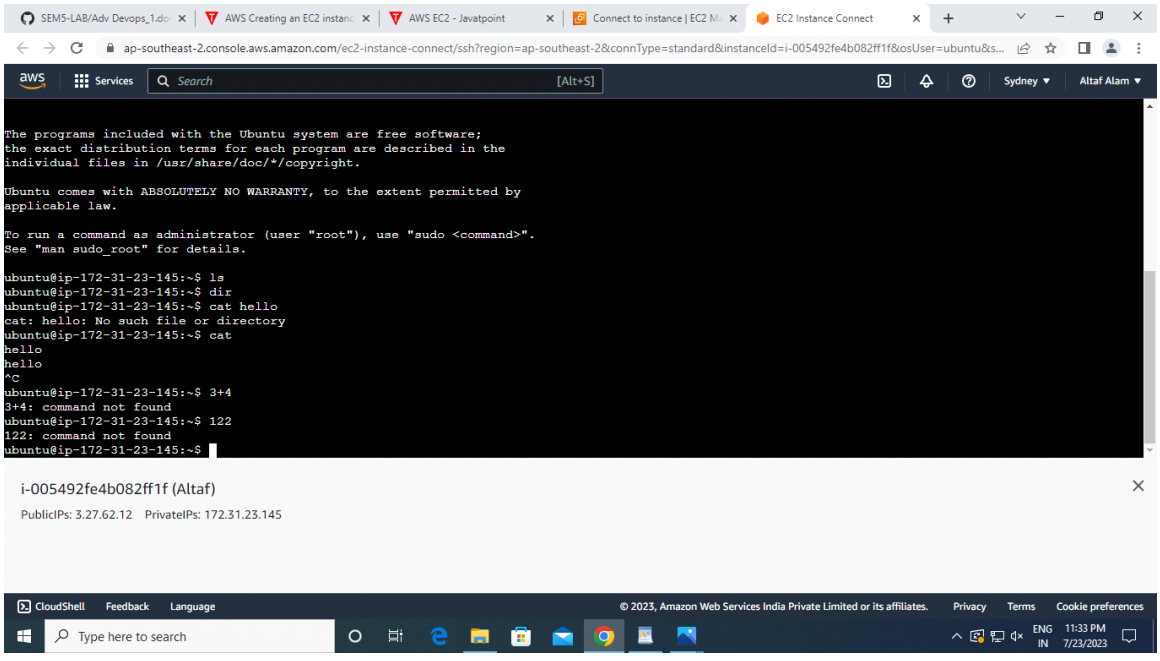
Connecting to Instance :

Name : Altaf Alam

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



FOLLOW SOME BASIC LINUX COMMANDS AS SHOWN BELOW-



CONCLUSION :

Altaf Alam , 02 , T11

Name : Altaf Alam

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

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

Name : Altaf Alam

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

Experiment No 2

Aim : To create a Cloud9 Environment.

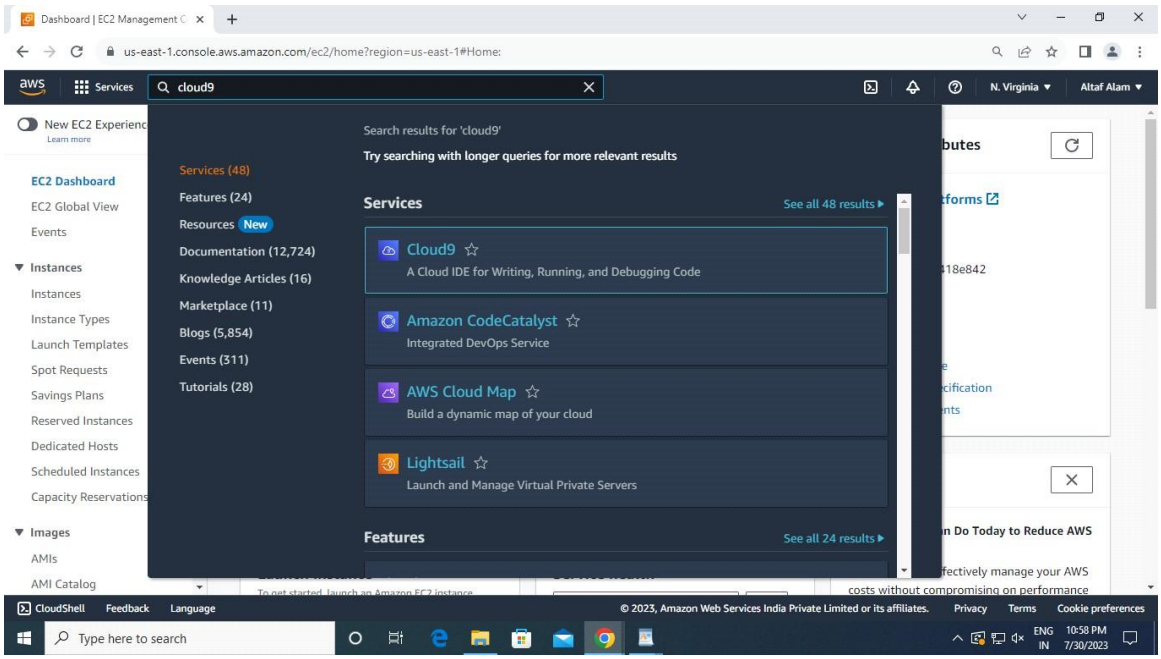
Theory :

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

STEPS-

LOG IN TO YOUR AWS ACCOUNT,

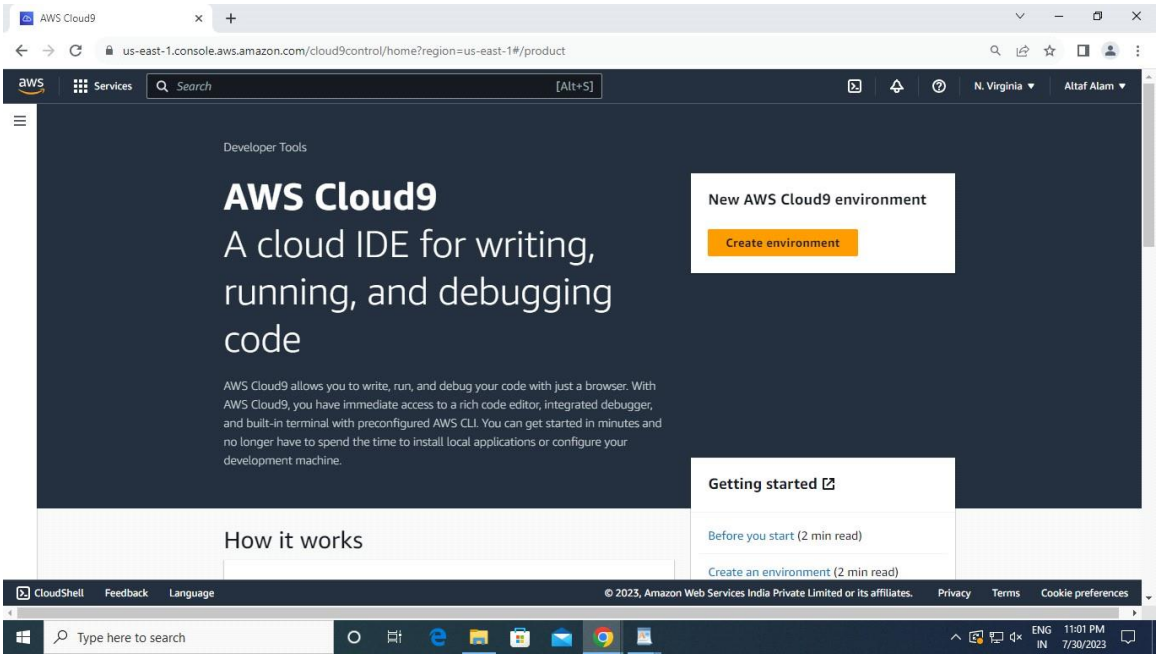
SEARCH FOR CLOUD 9 IN THE SEARCH BAR



CLICK ON CLOUD 9.

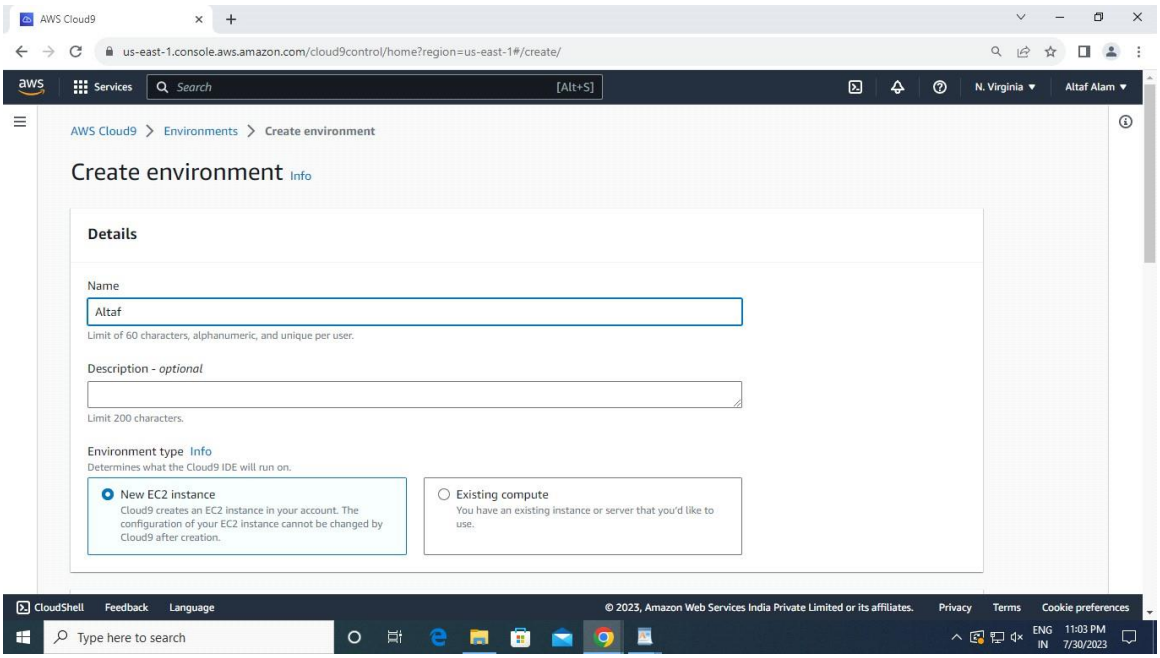
Name : Altaf Alam

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



CLICK ON CREATE ENVIRONMNET,

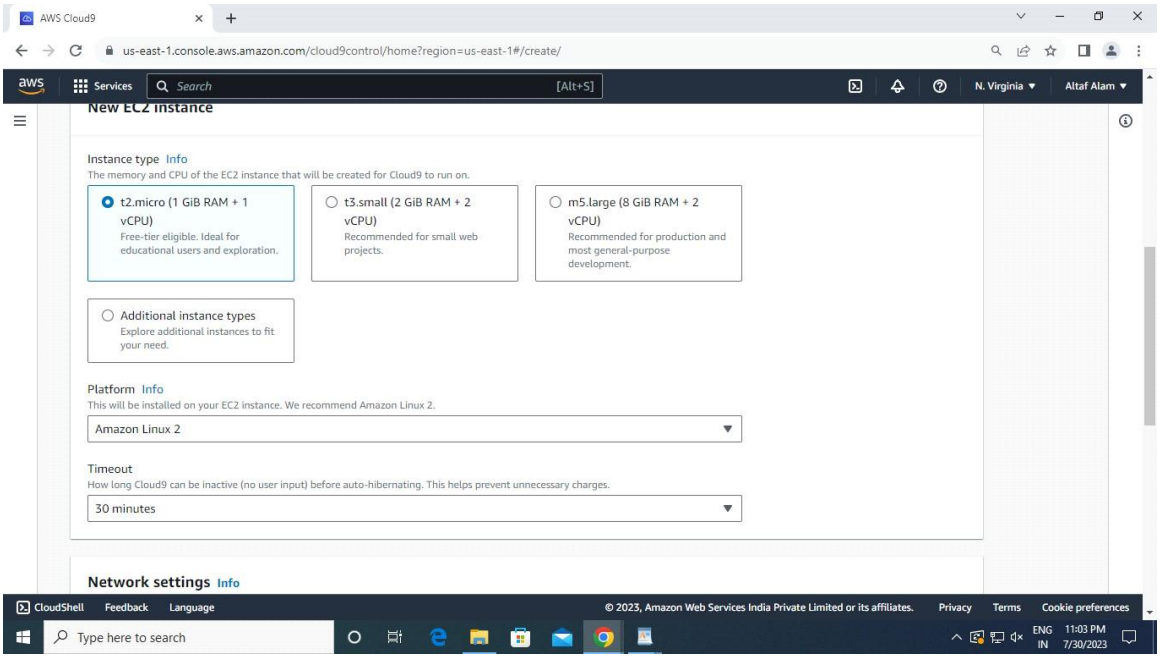
NAME THE ENVIRONMENT



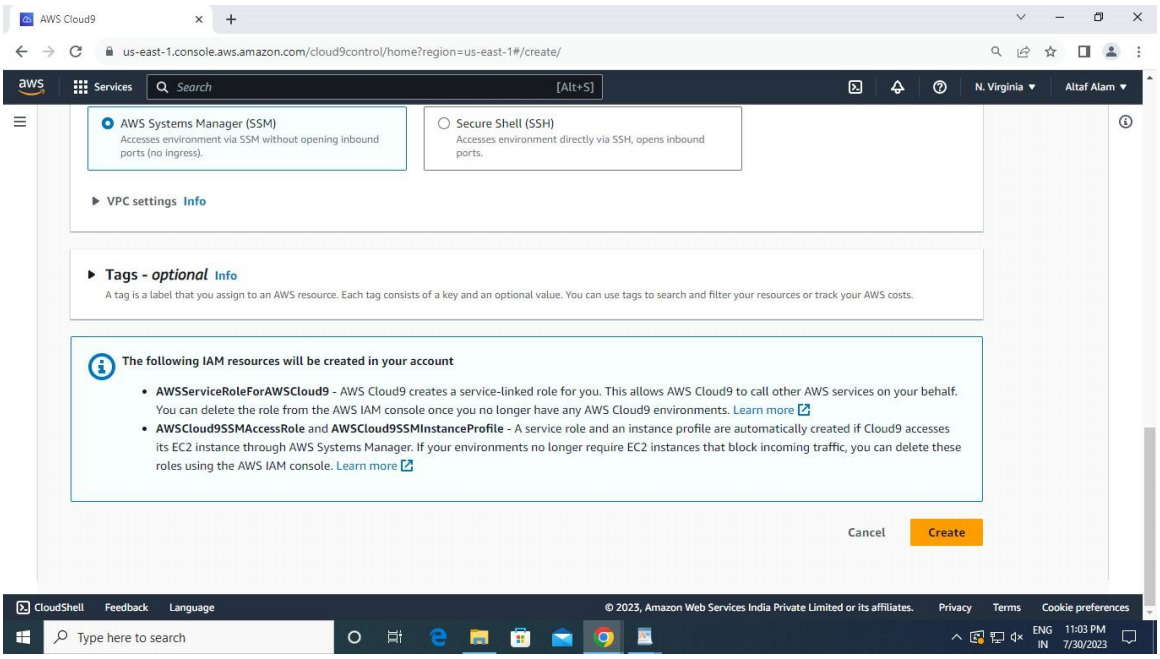
CHOOSE FREE TIER INSTANCE.

Name : Altaf Alam

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



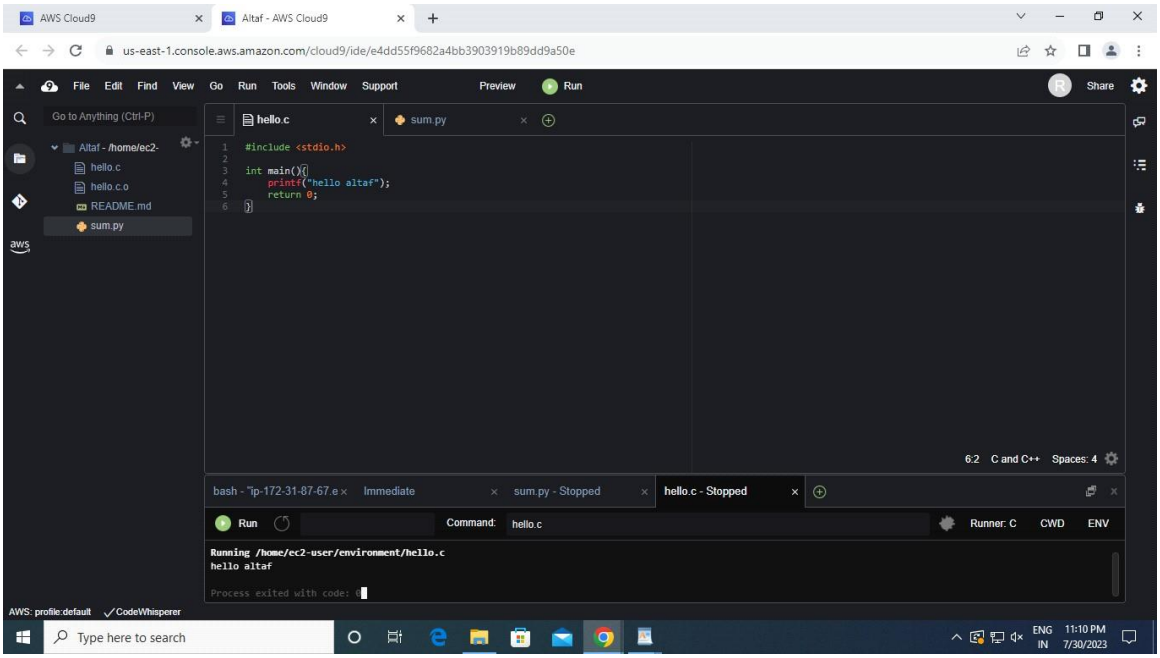
CLICK ON CREATE BUTTON .



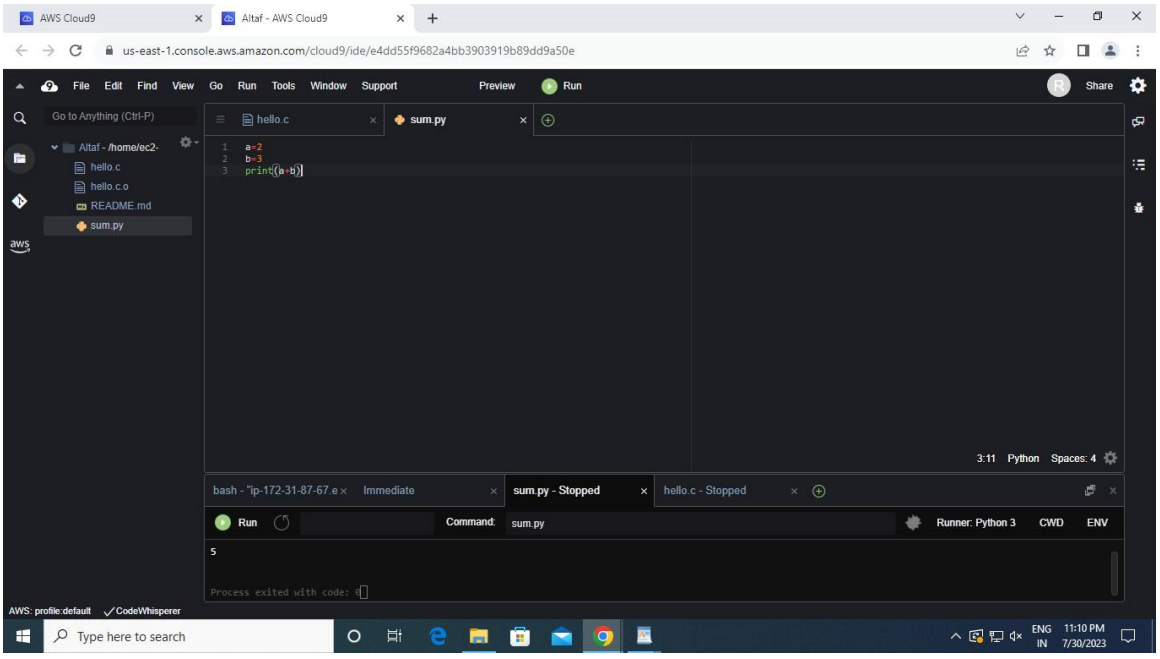
CLOUD9 CREATED. CLICK ON OPEN OPTION.

Name : Altaf Alam

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



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



Conclusion :

Hence learned and implemented steps to Create an Cloud9 environment.

Name : Altaf Alam

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

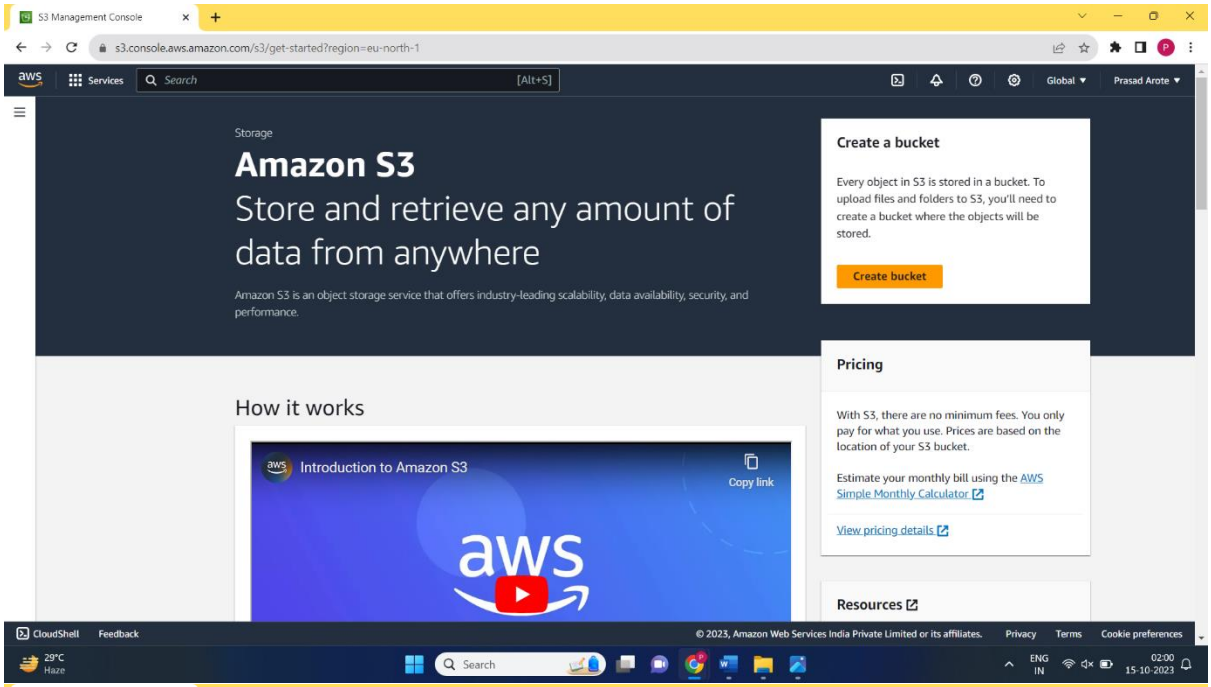
LAB ASSIGNMENT 3

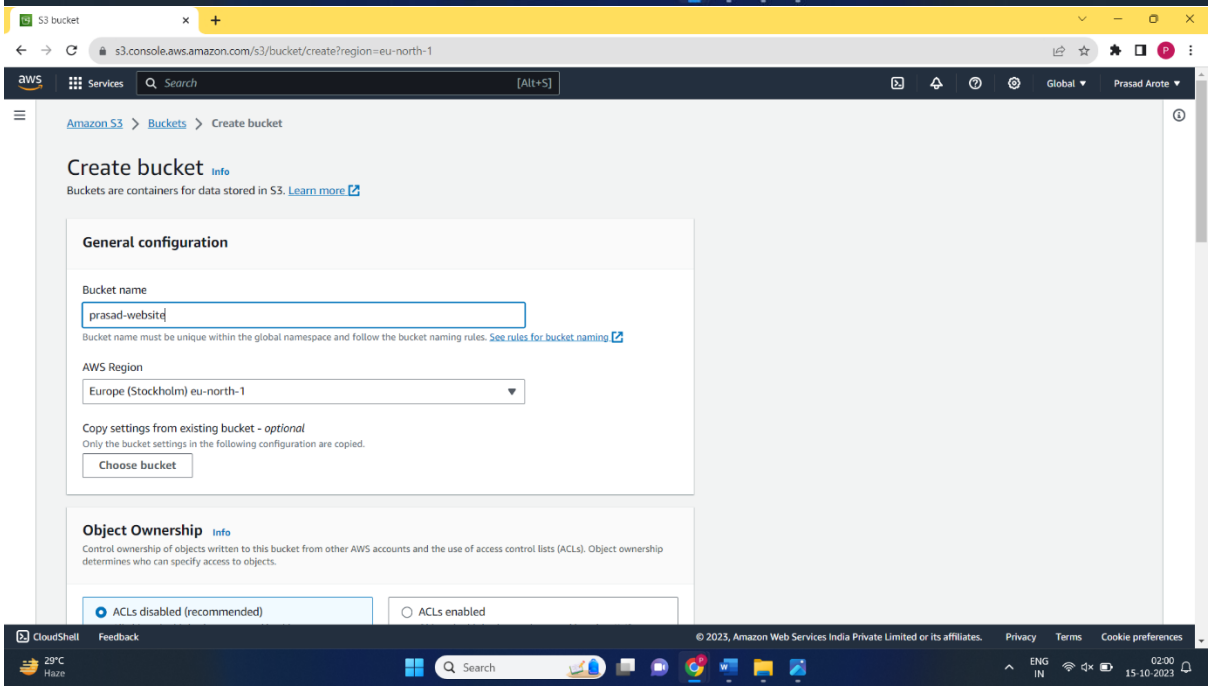
AIM: To study AWS S3 service and create a bucket for hosting static web application.

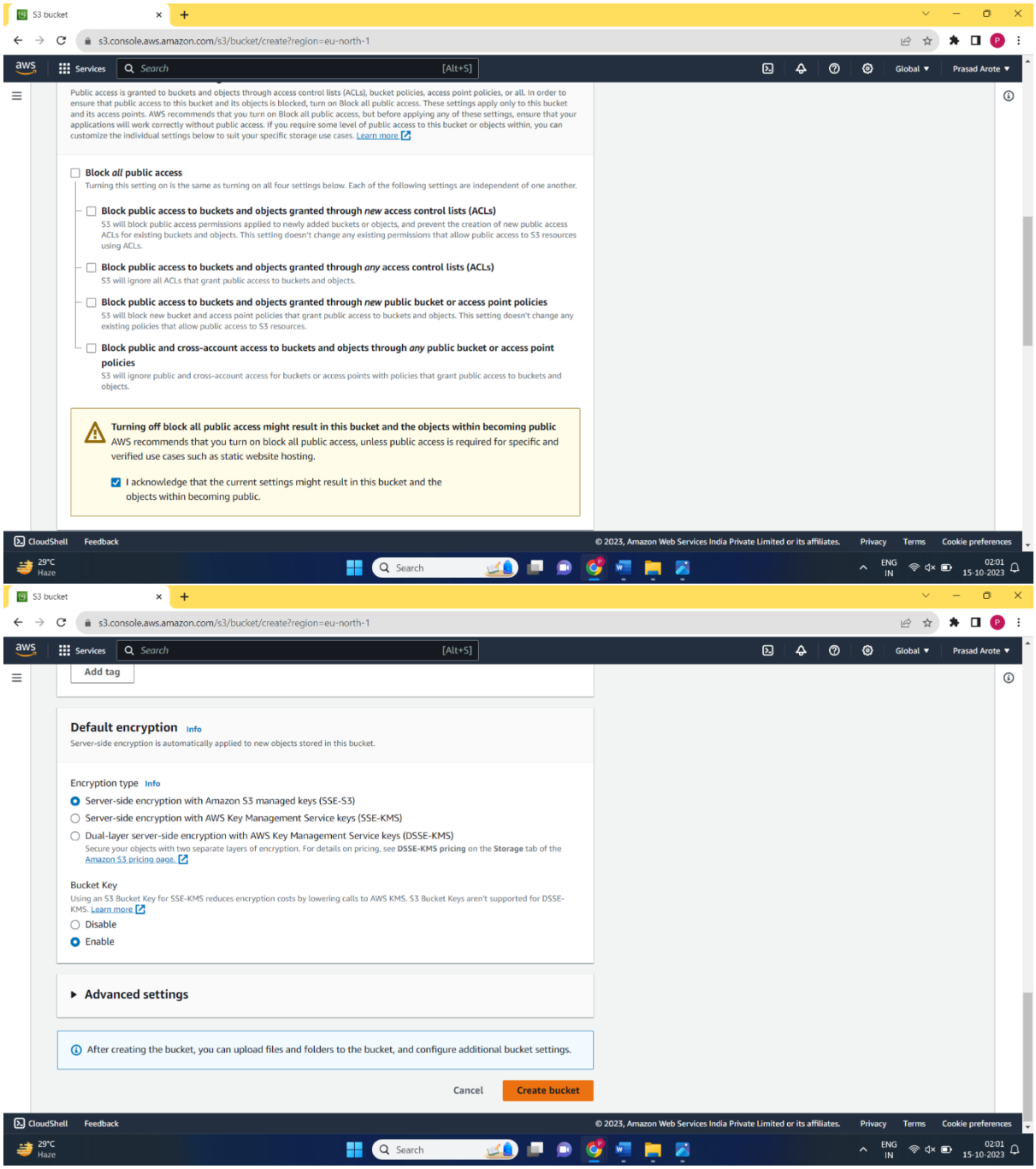
LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.

THEORY:

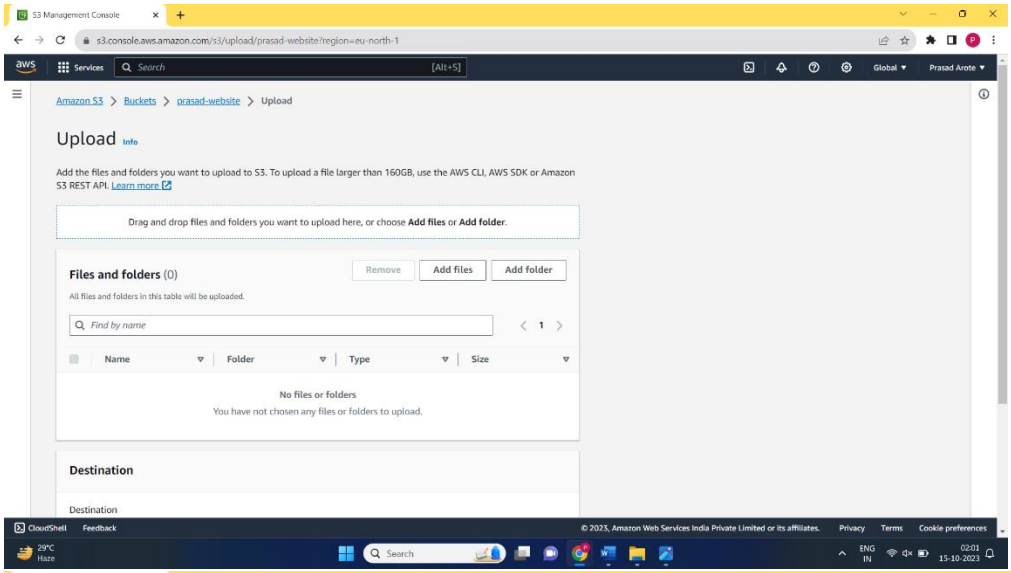
1. Create a S3 bucket.

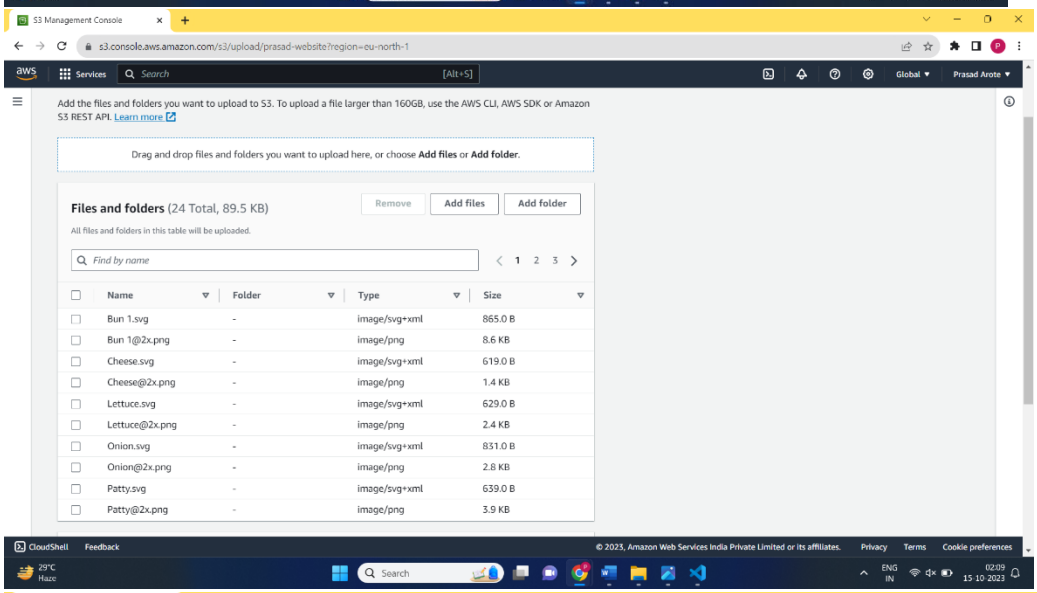


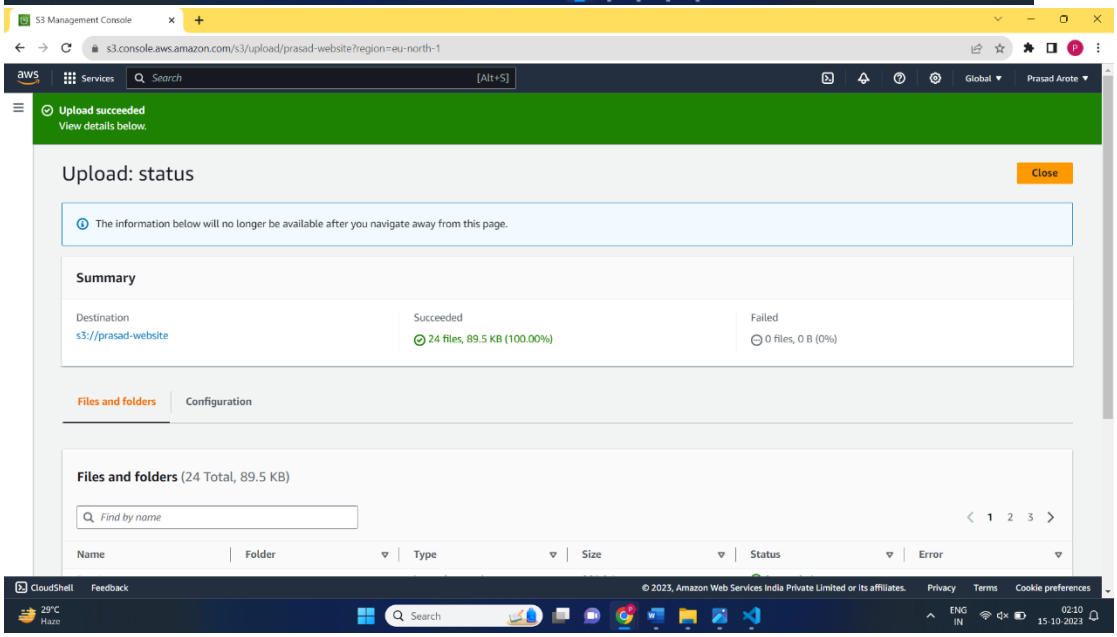




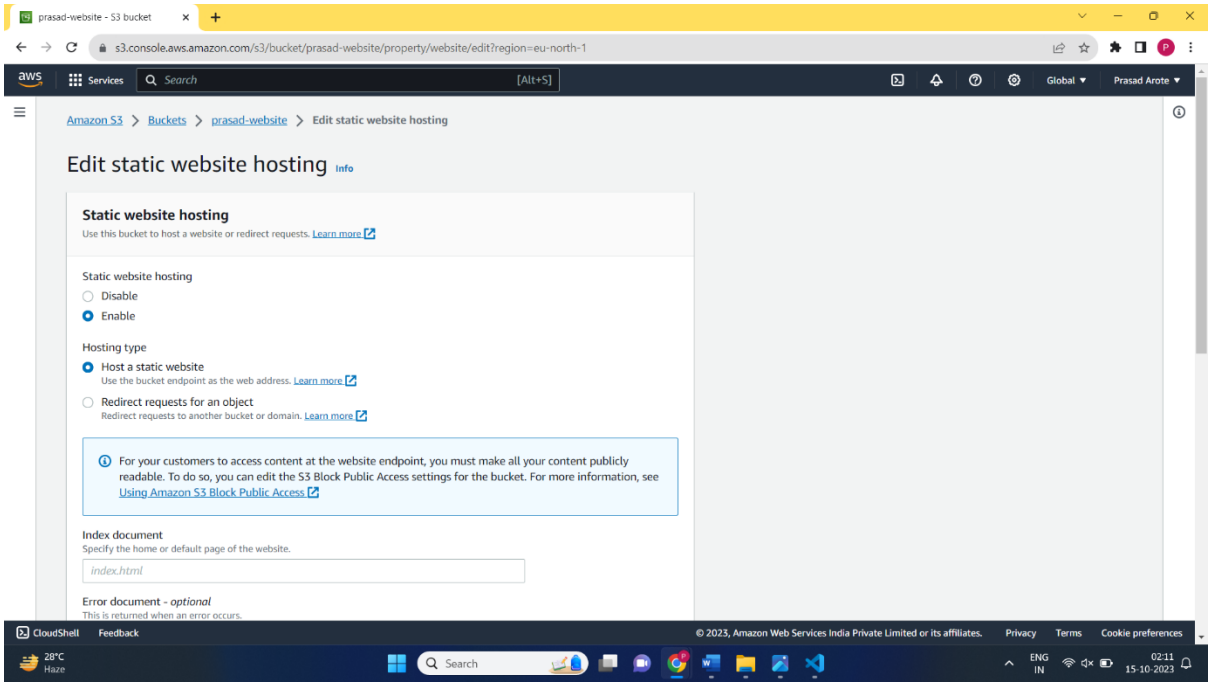
2. Upload the files of web application.

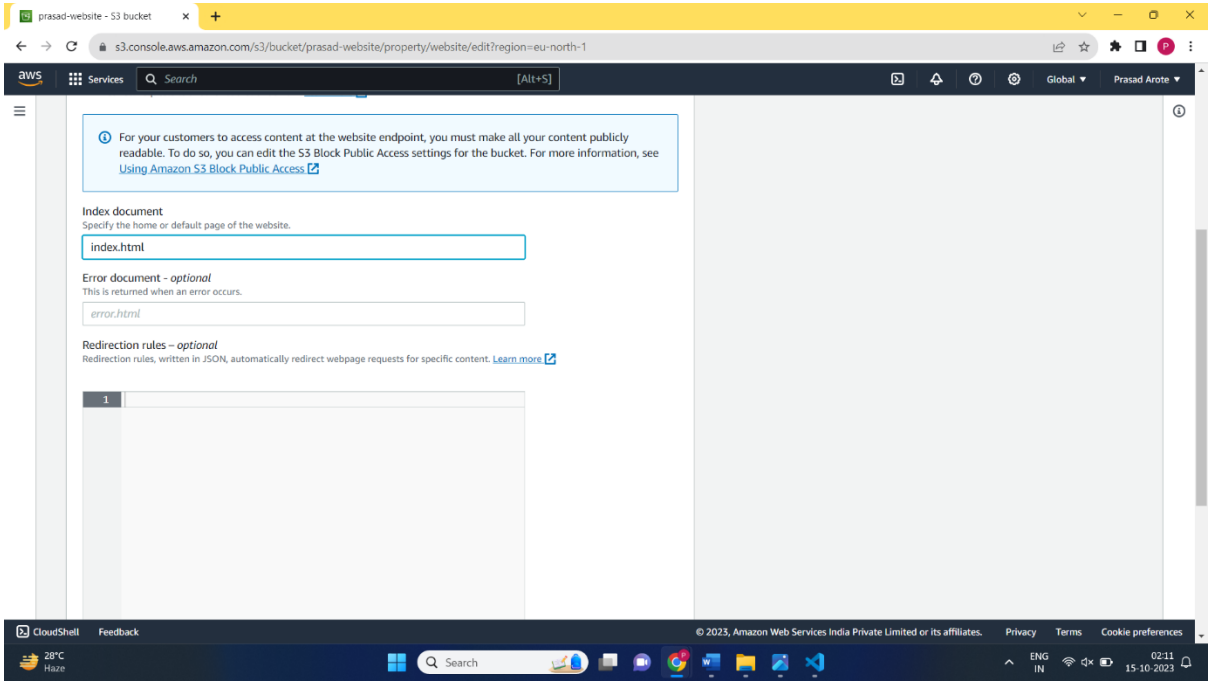

S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
Services
Search
[Alt+S]
Global
Prasad Arote
Amazon S3 > Buckets > prasad-website > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS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

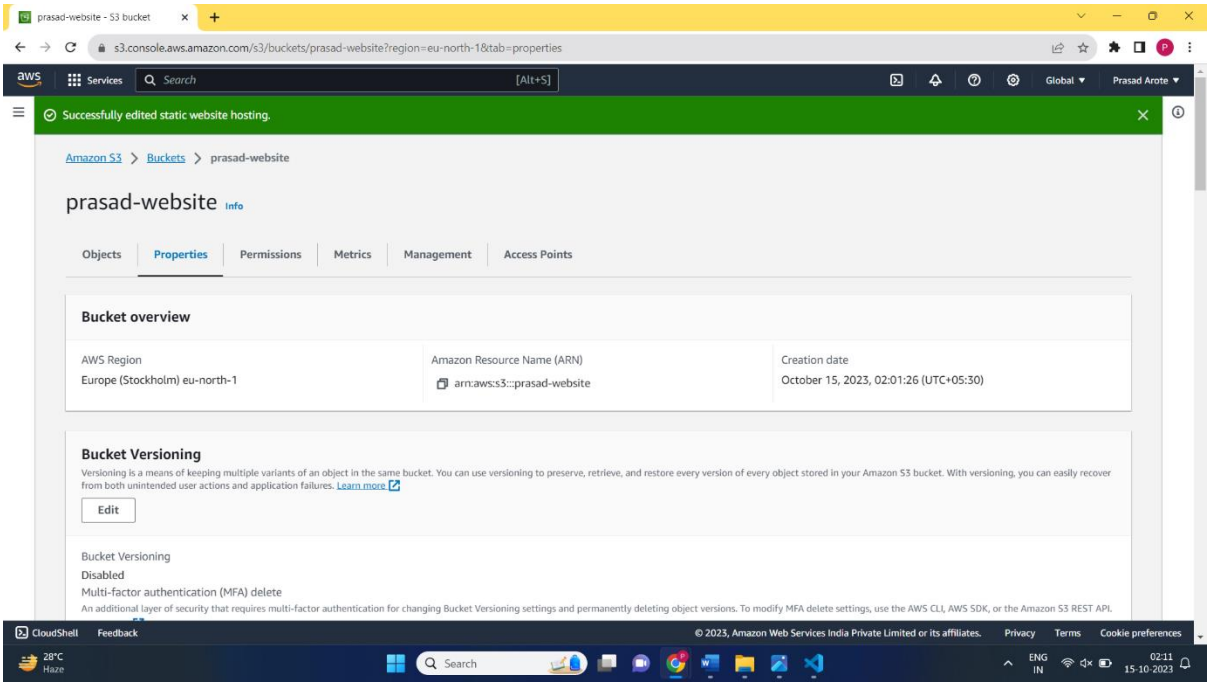

S3 Management Console
s3.console.aws.amazon.com/s3/upload/prasad-website?region=eu-north-1
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose 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)
Name
Folder
Type
Size
Status
Error

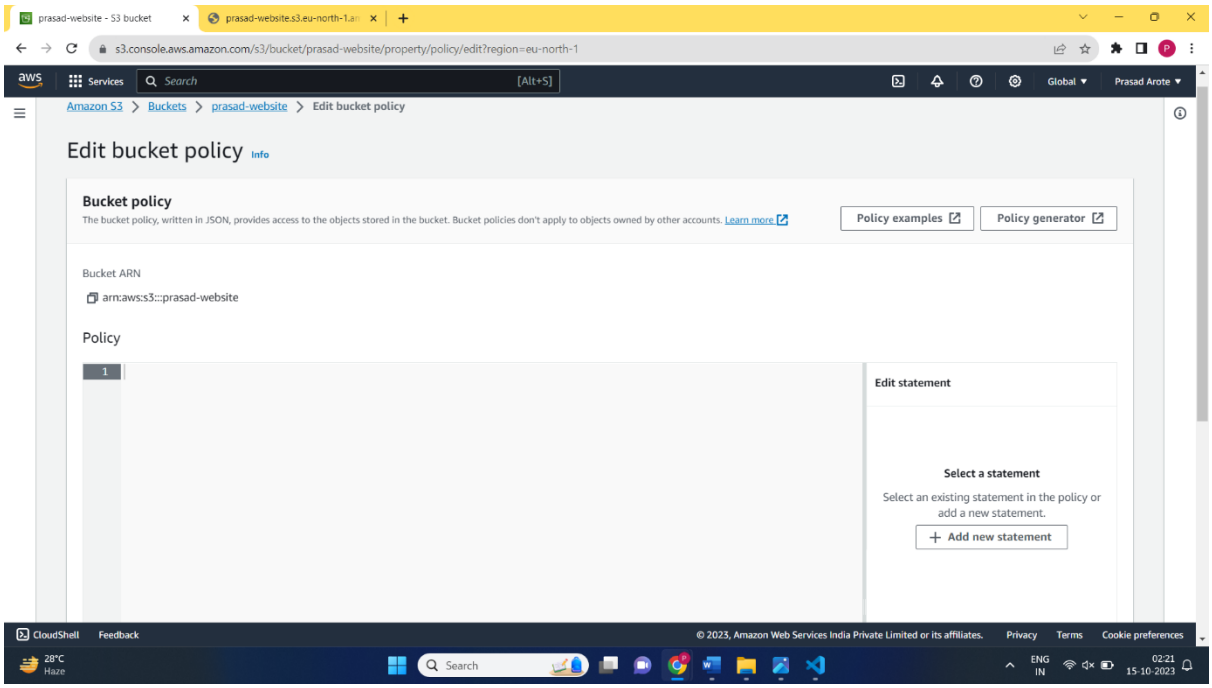
3. Enable Static website hosting

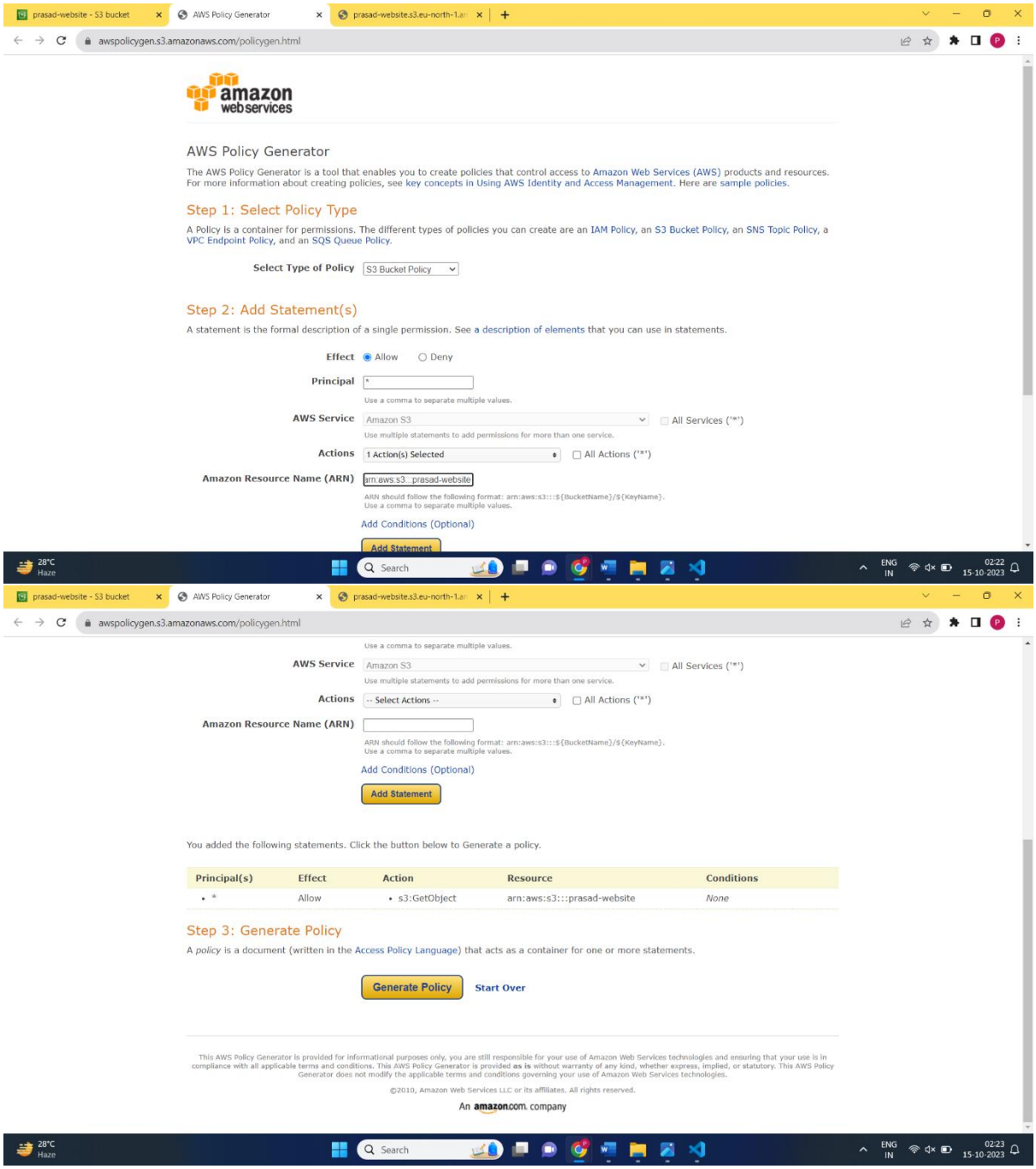




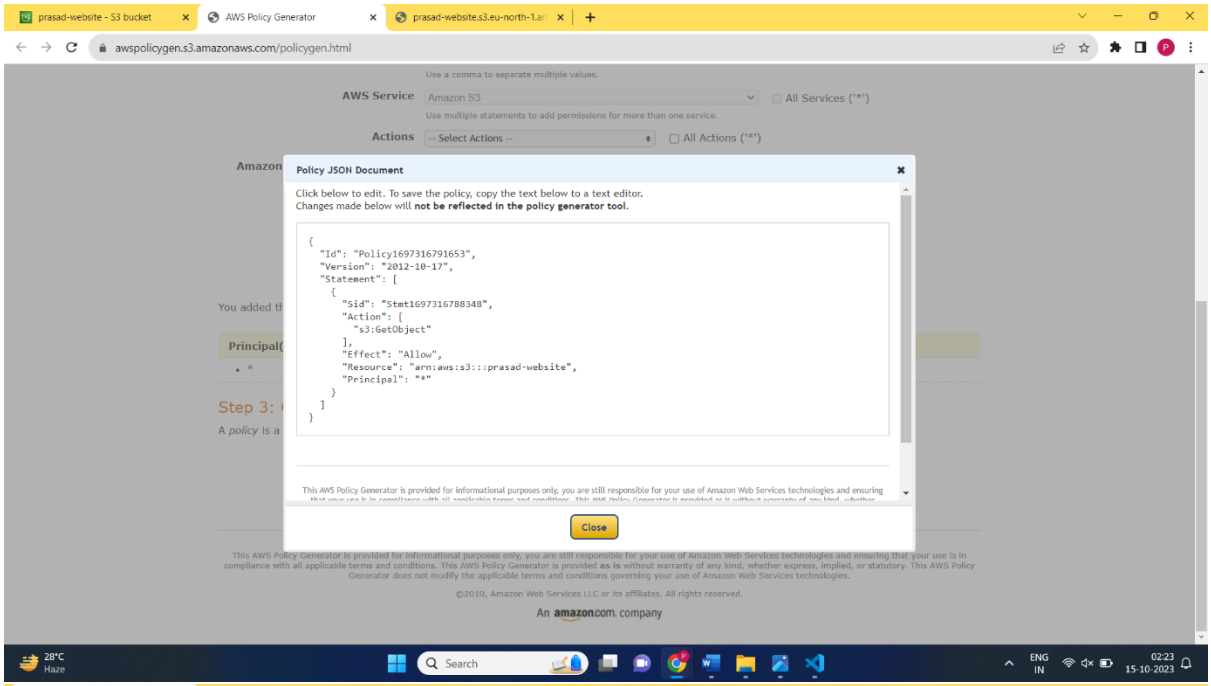


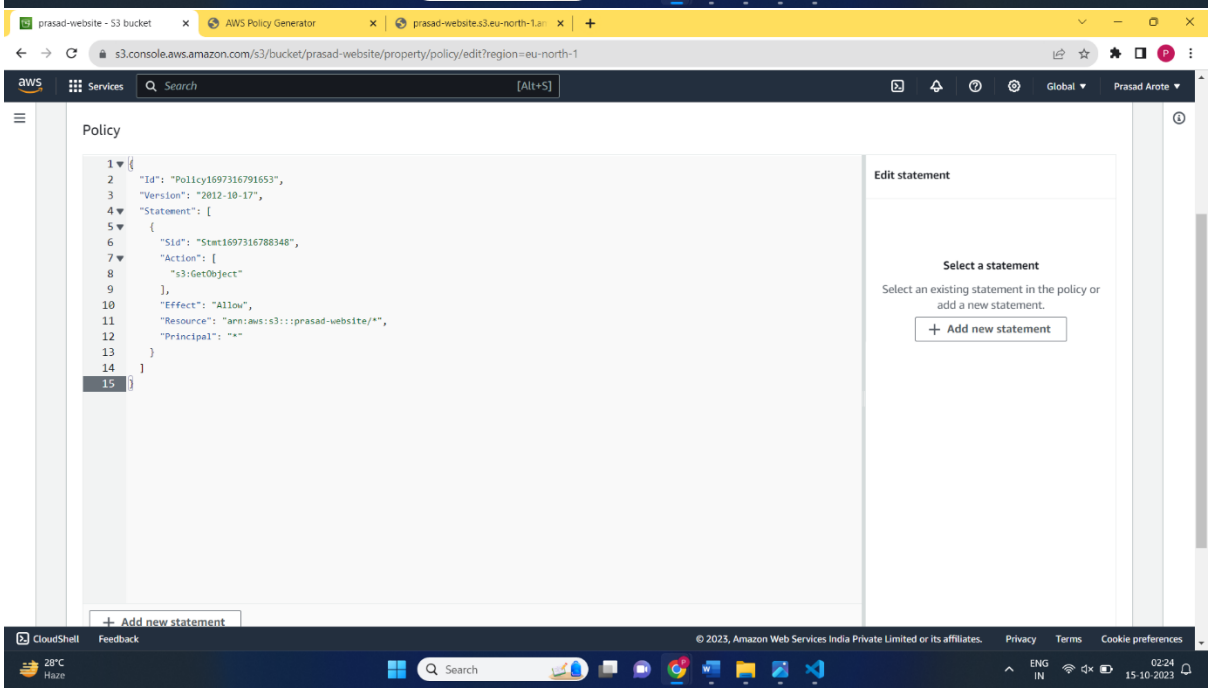
5. Change the Bucket Policy

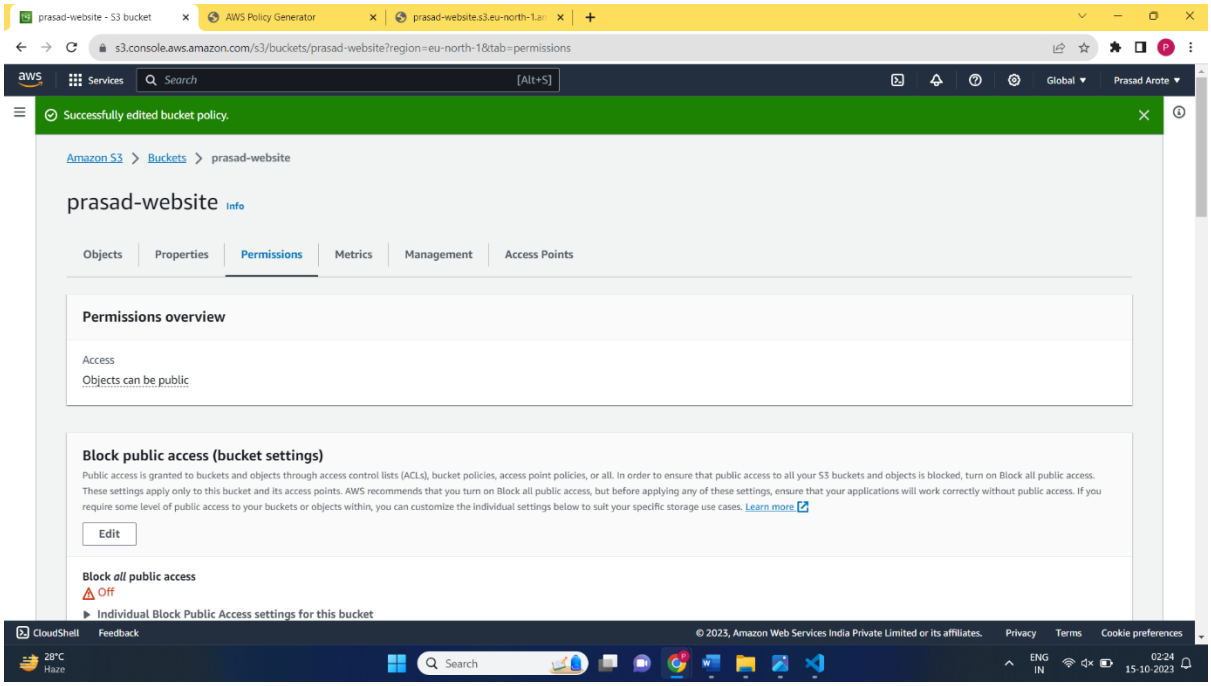



prasad-website - S3 bucket
AWS Policy Generator
prasad-website.s3.eu-north-1.am
awspolicygen.s3.amazonaws.com/policygen.html
amazon web services
AWS Policy Generator
The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.
Step 1: Select Policy Type
A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.
Select Type of Policy
S3 Bucket Policy
Step 2: Add Statement(s)
A statement is the formal description of a single permission. See a description of elements that you can use in statements.
Effect
Allow
Deny
Principal
*
Use a comma to separate multiple values.
AWS Service
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
1 Action(s) Selected
All Actions ('*')
Amazon Resource Name (ARN)
arn:aws:s3:::prasad-website
ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.
Add Conditions (Optional)
Add Statement
28°C
Haze
Search
ENG
IN
02:22
15-10-2023

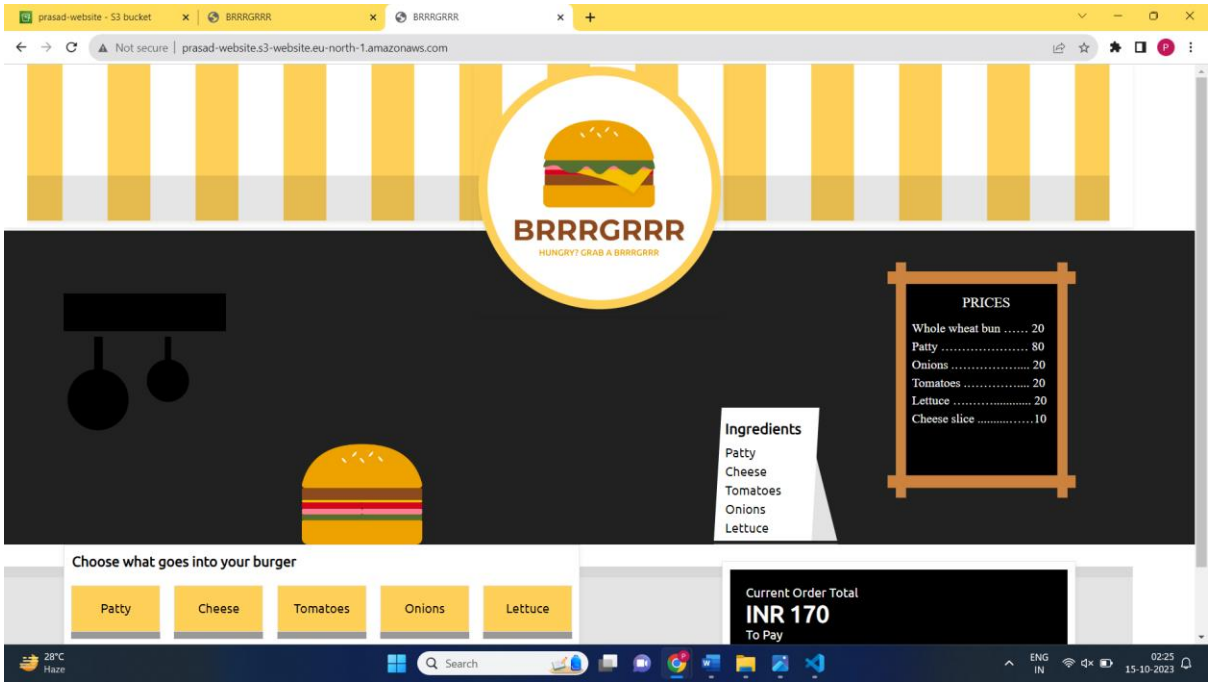
prasad-website - S3 bucket
AWS Policy Generator
prasad-website.s3.eu-north-1.am
awspolicygen.s3.amazonaws.com/policygen.html
Use a comma to separate multiple values.
AWS Service
Amazon S3
All Services ('*')
Use multiple statements to add permissions for more than one service.
Actions
-- Select Actions --
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
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.

Name : Altaf Alam

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

Assignment No 3

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

LAB OUTCOME –

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

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

THEORY -

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

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

A) Installation and Configuration of Terraform in Windows

Step 1: Download terraform

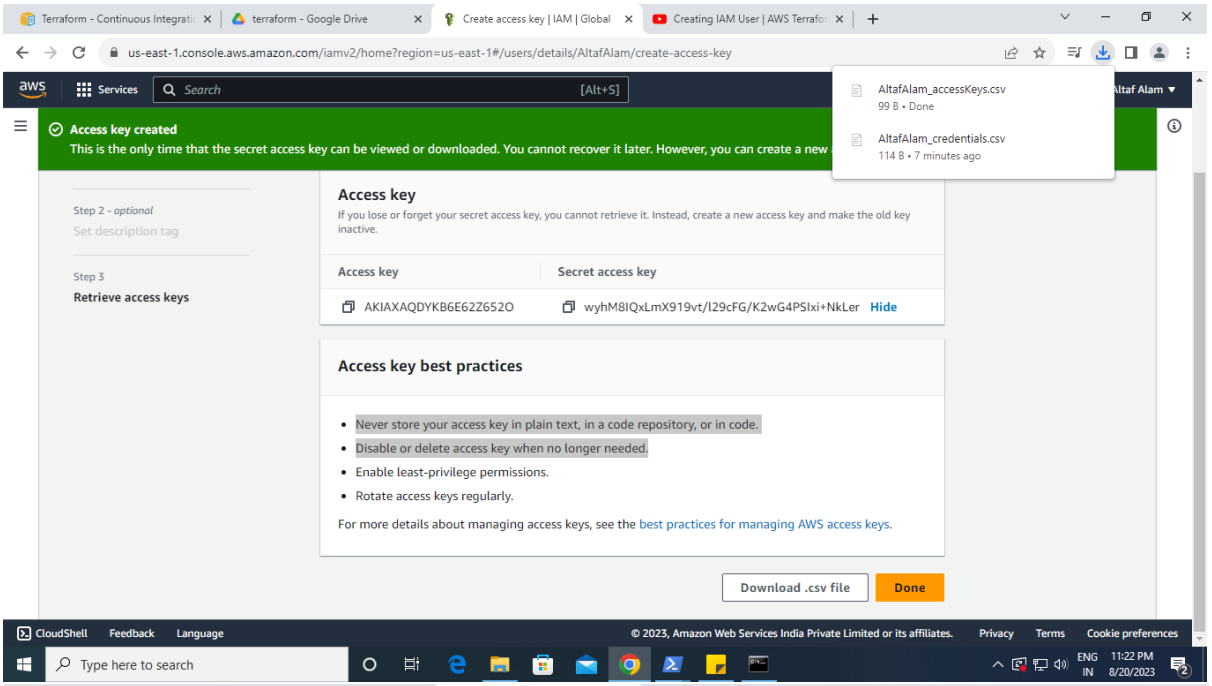
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

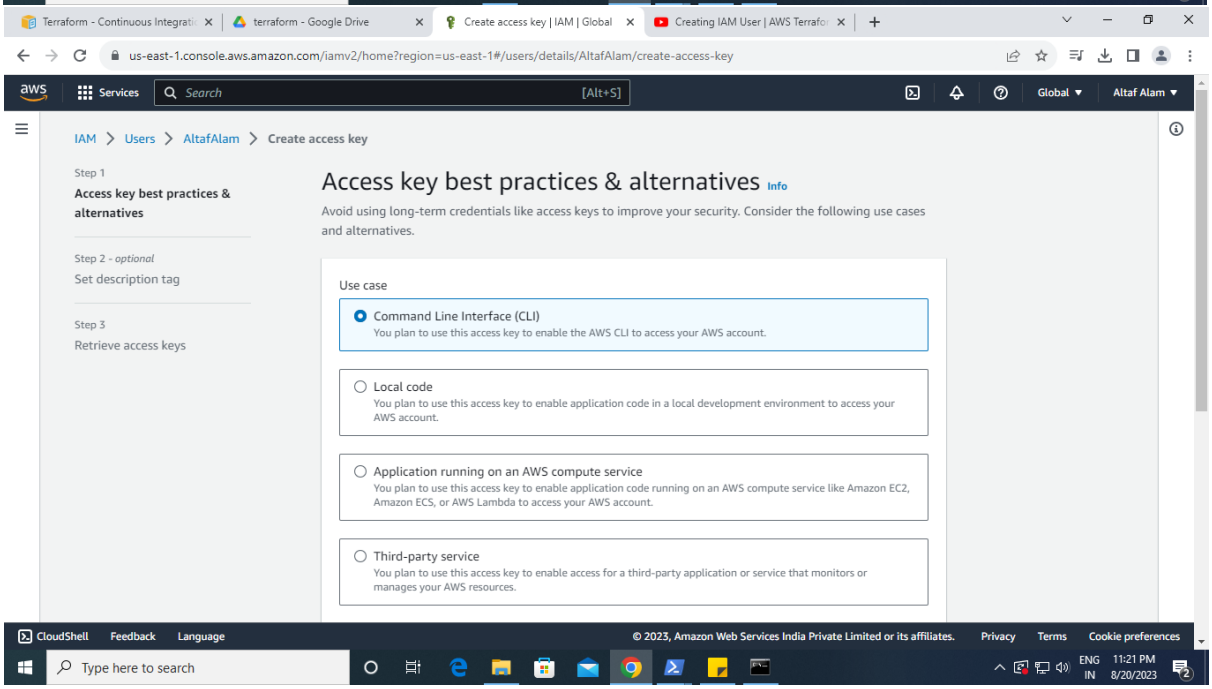
website: https://www.terraform.io/downloads.html

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

Name : Altaf Alam

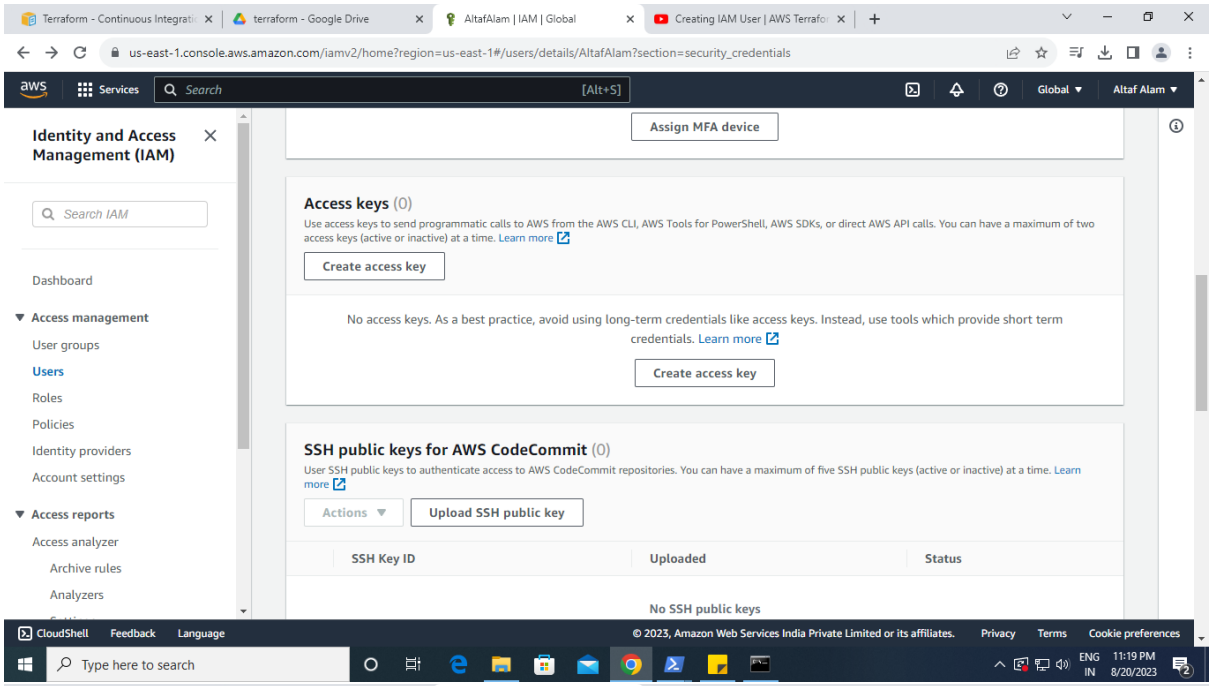
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

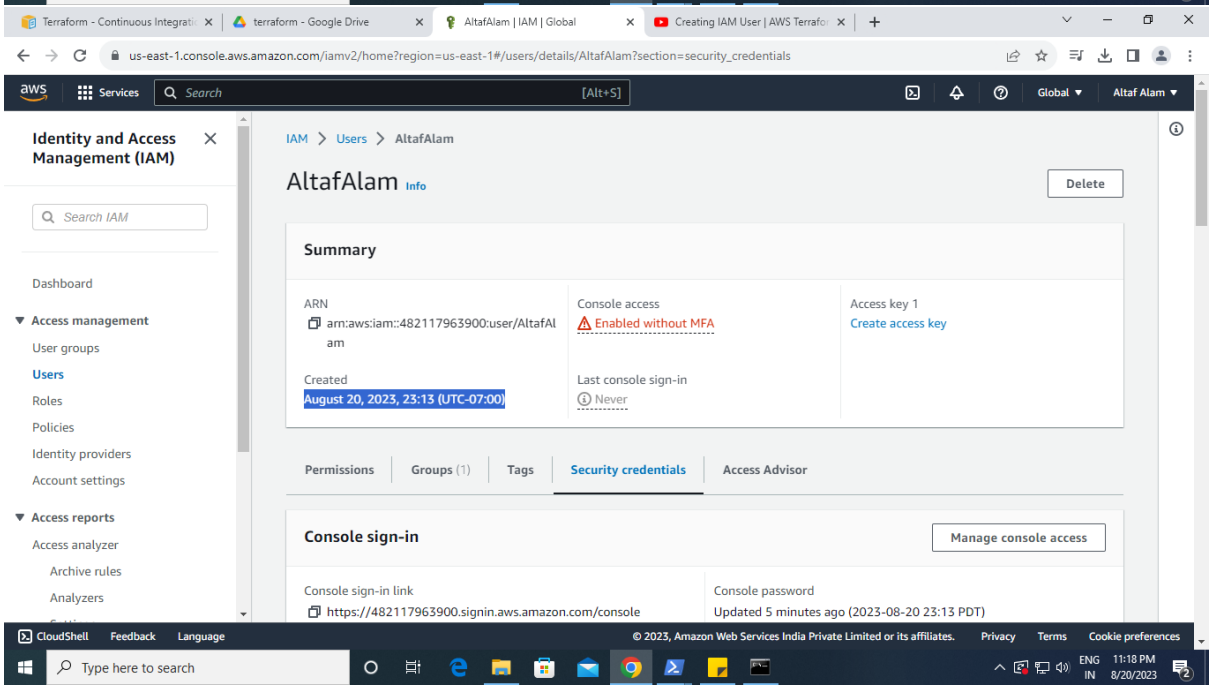




Name : Altaf Alam

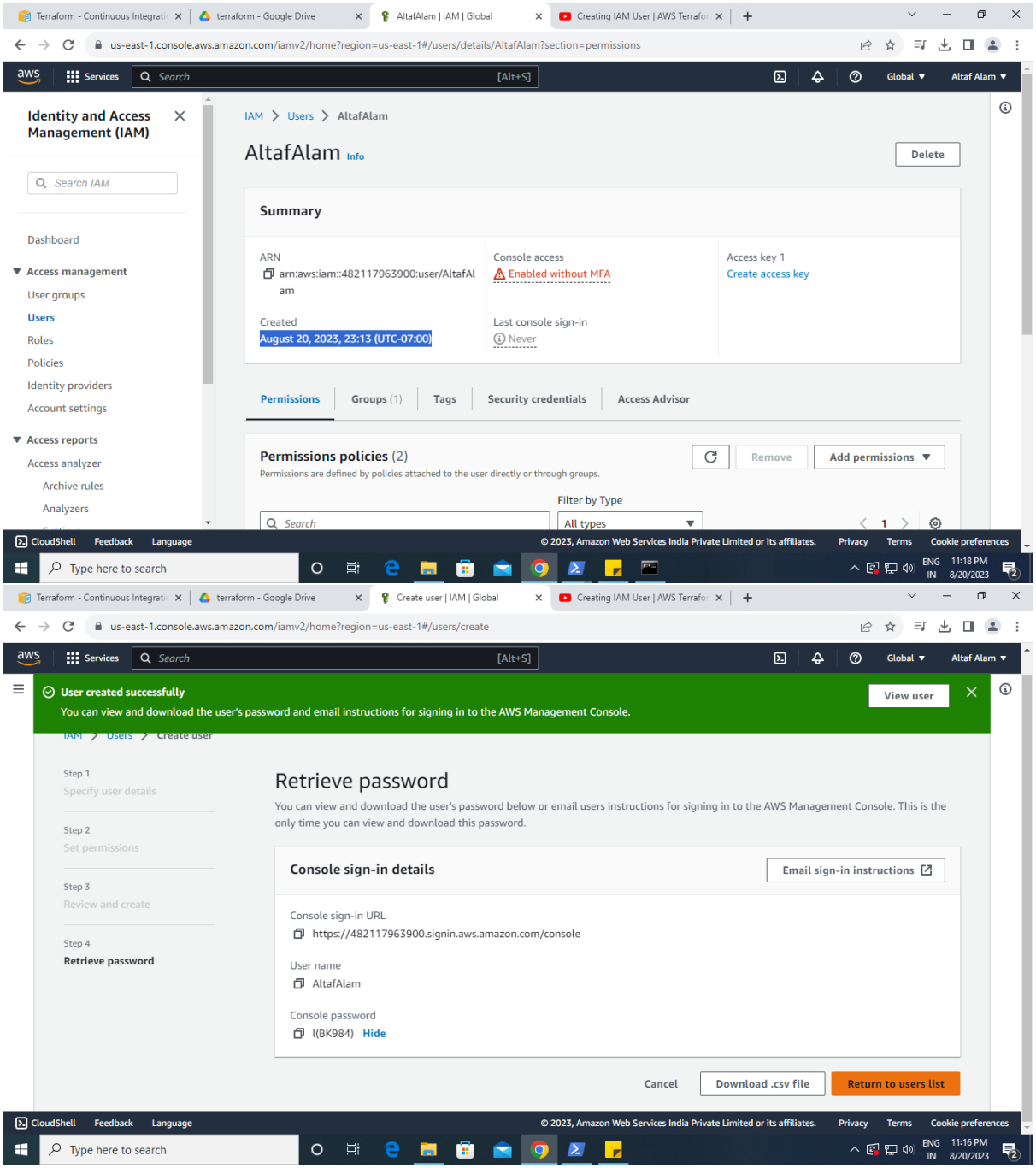
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23





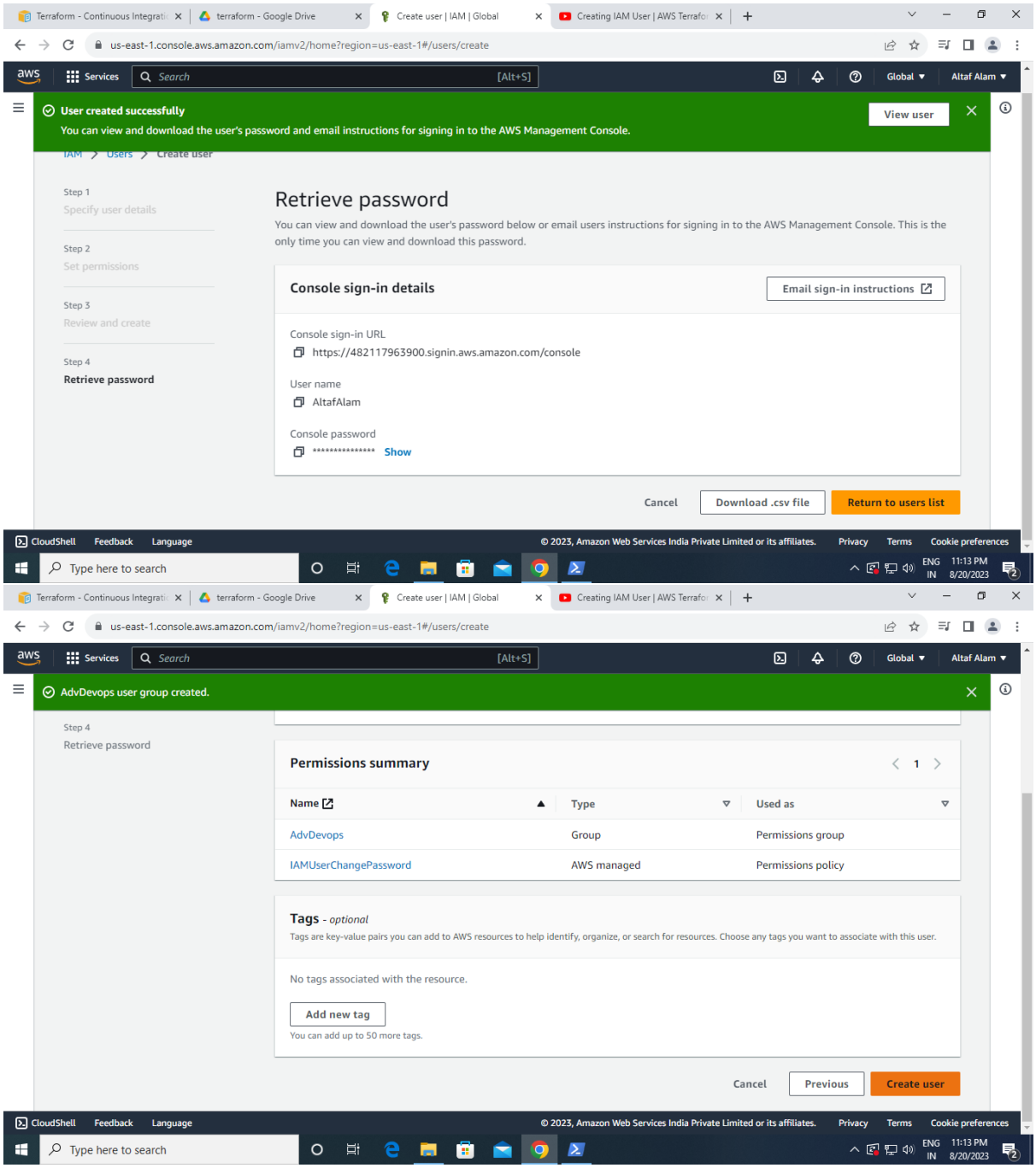
Name : Altaf Alam

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



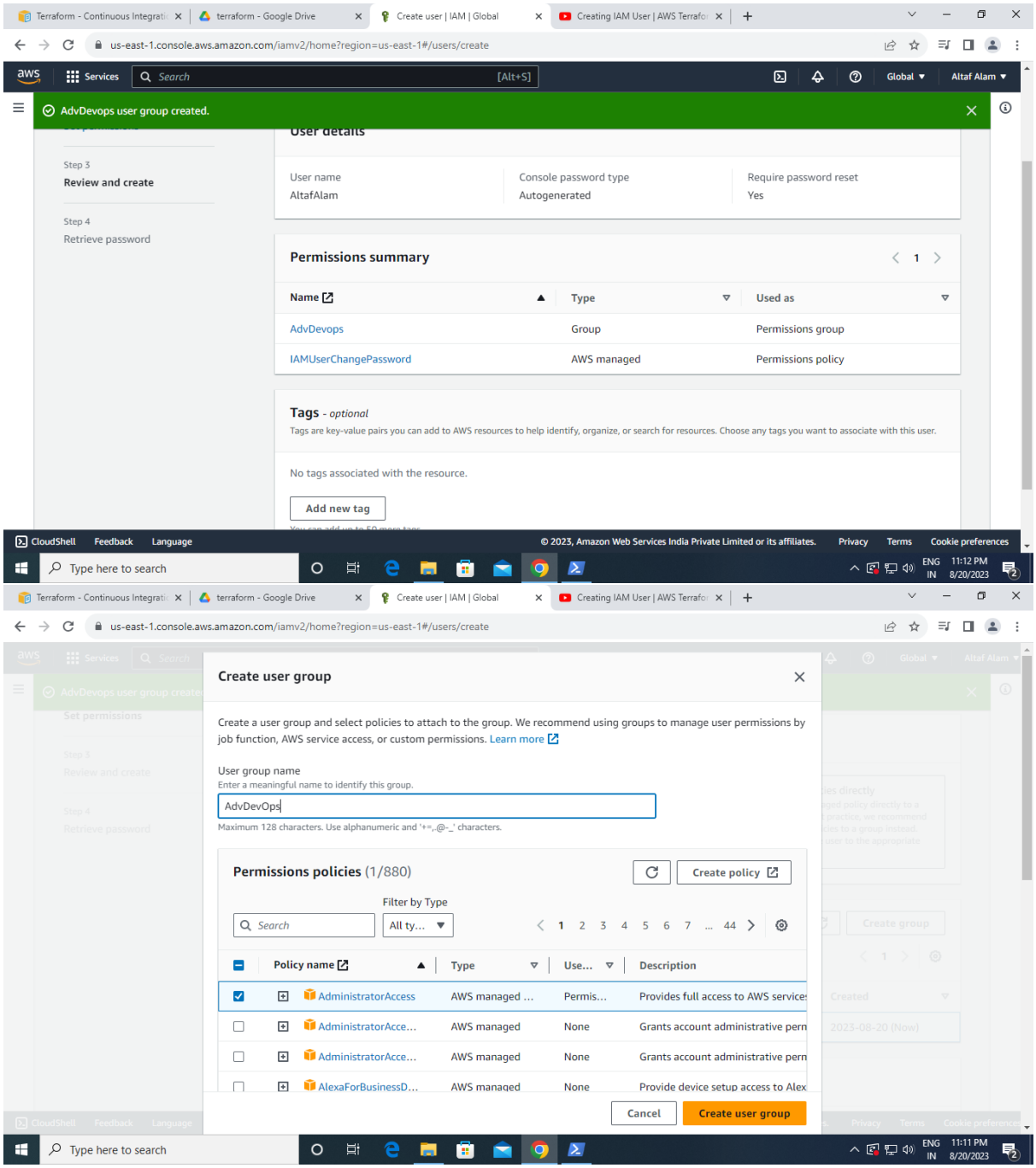
Name : Altaf Alam

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



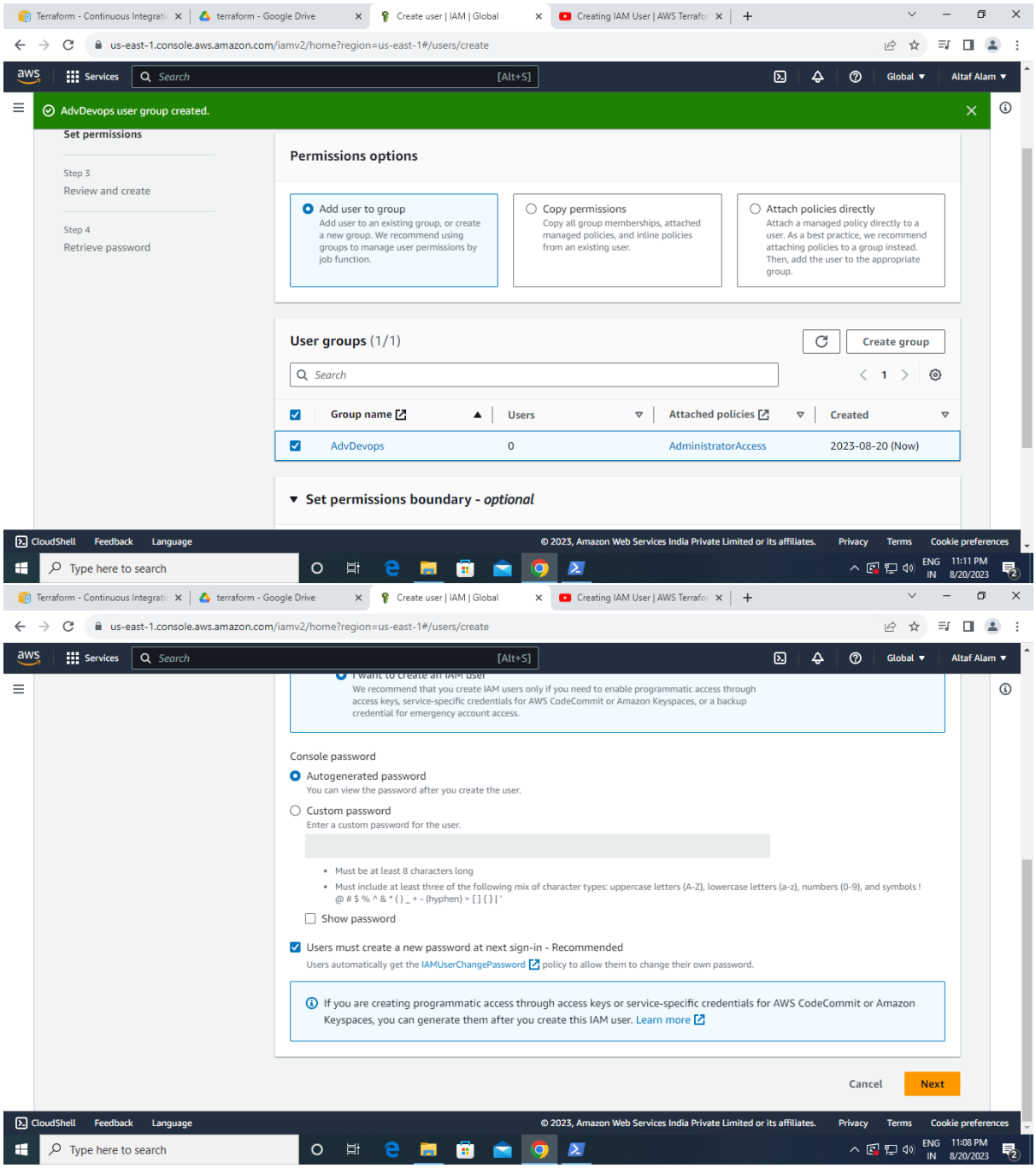
Name : Altaf Alam

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



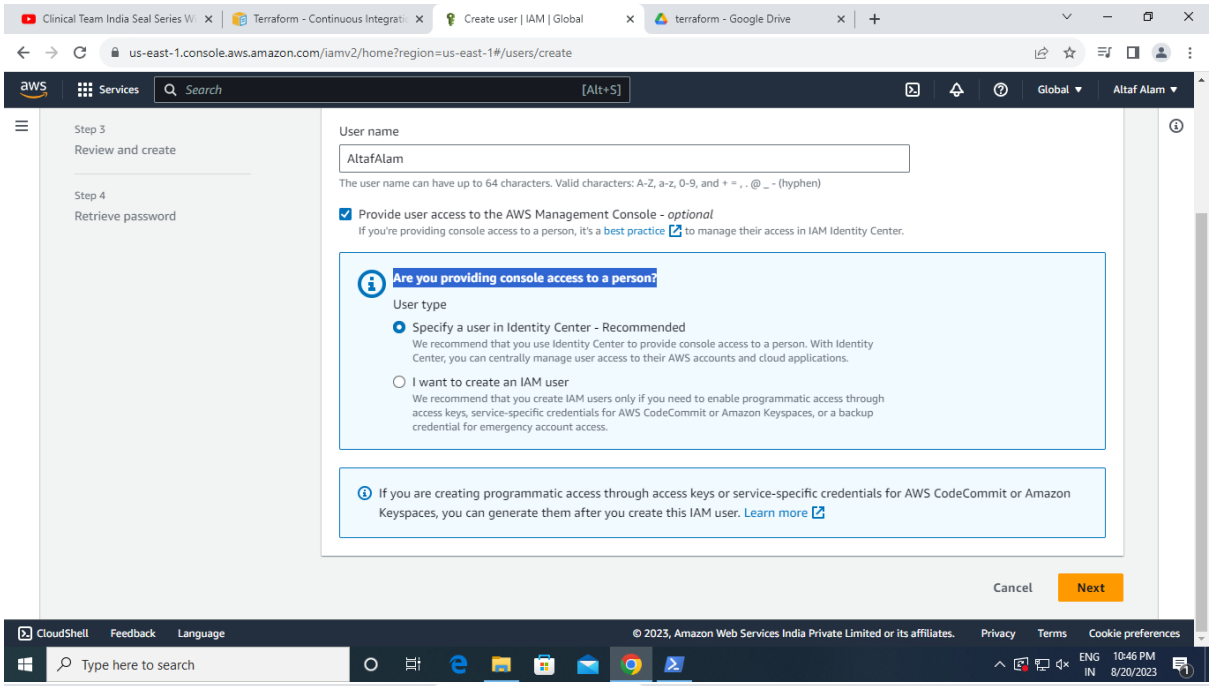
Name : Altaf Alam

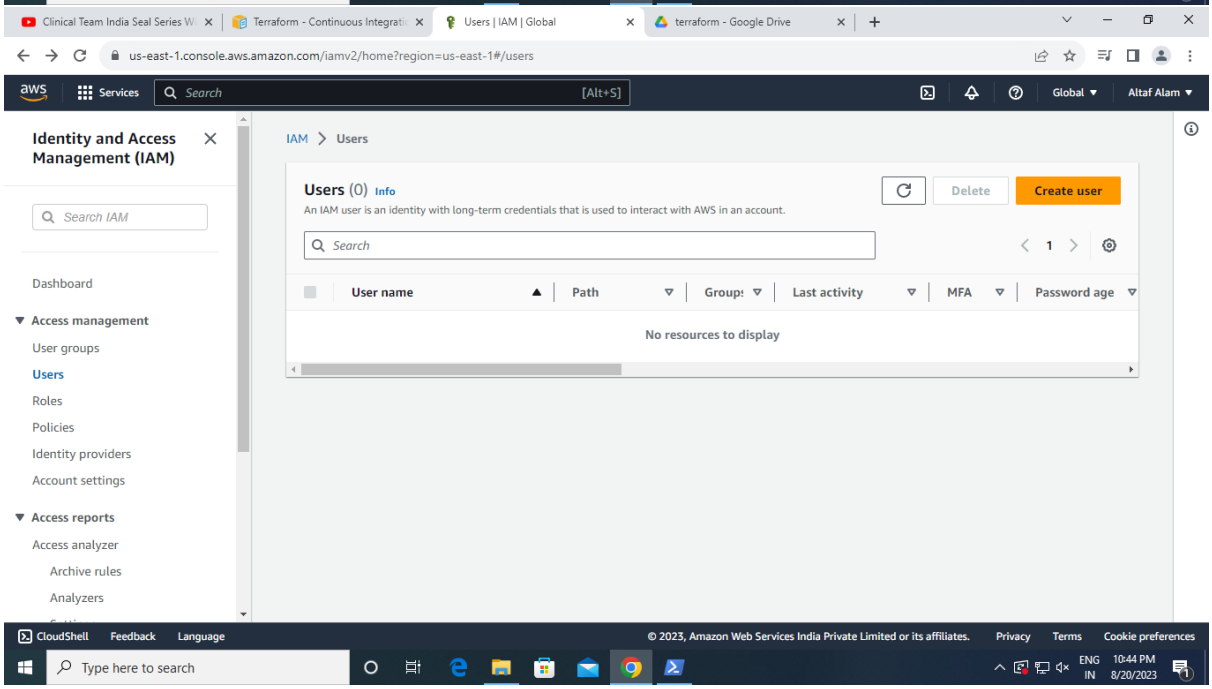
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23



Name : Altaf Alam

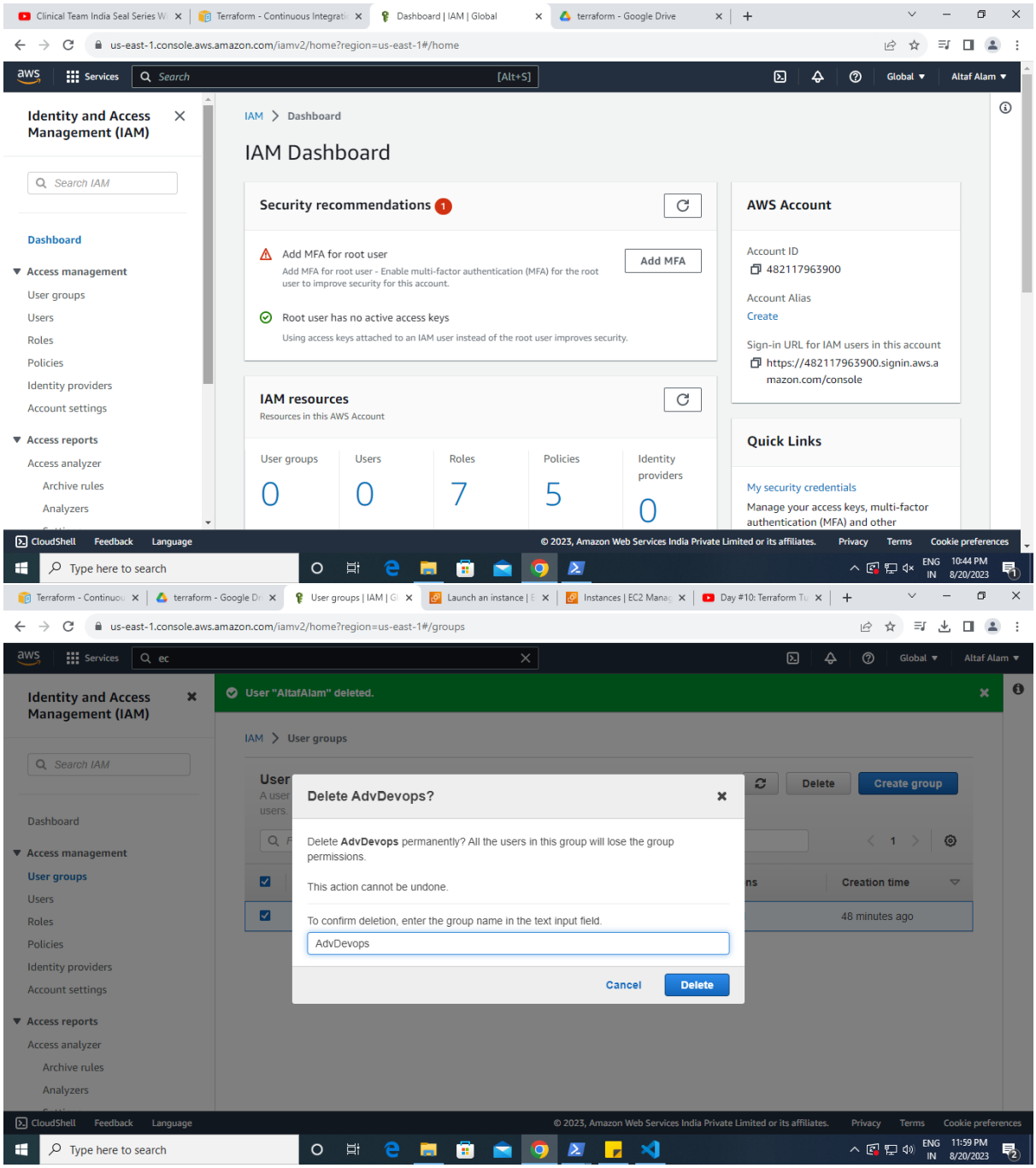
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23





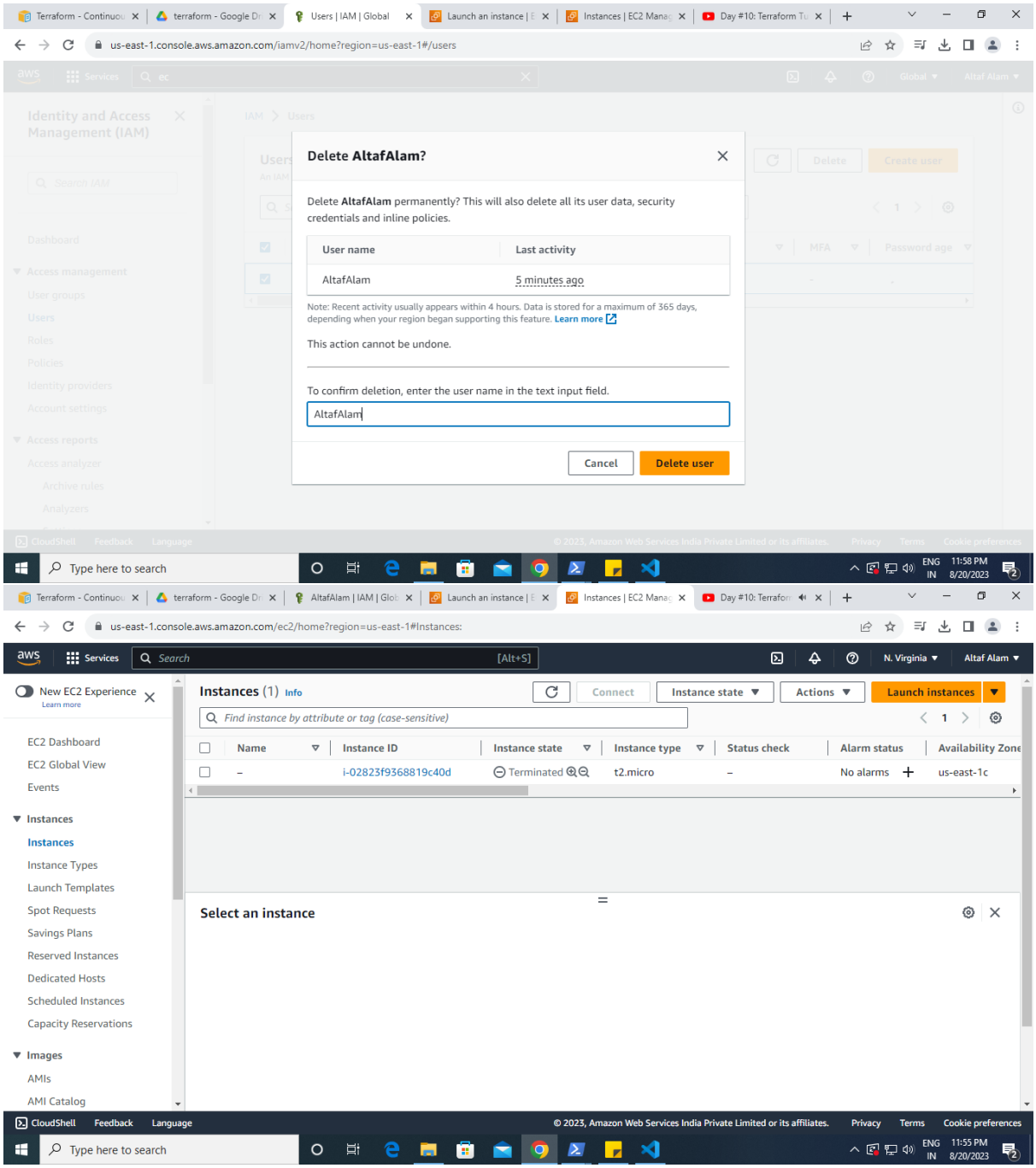
Name : Altaf Alam

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



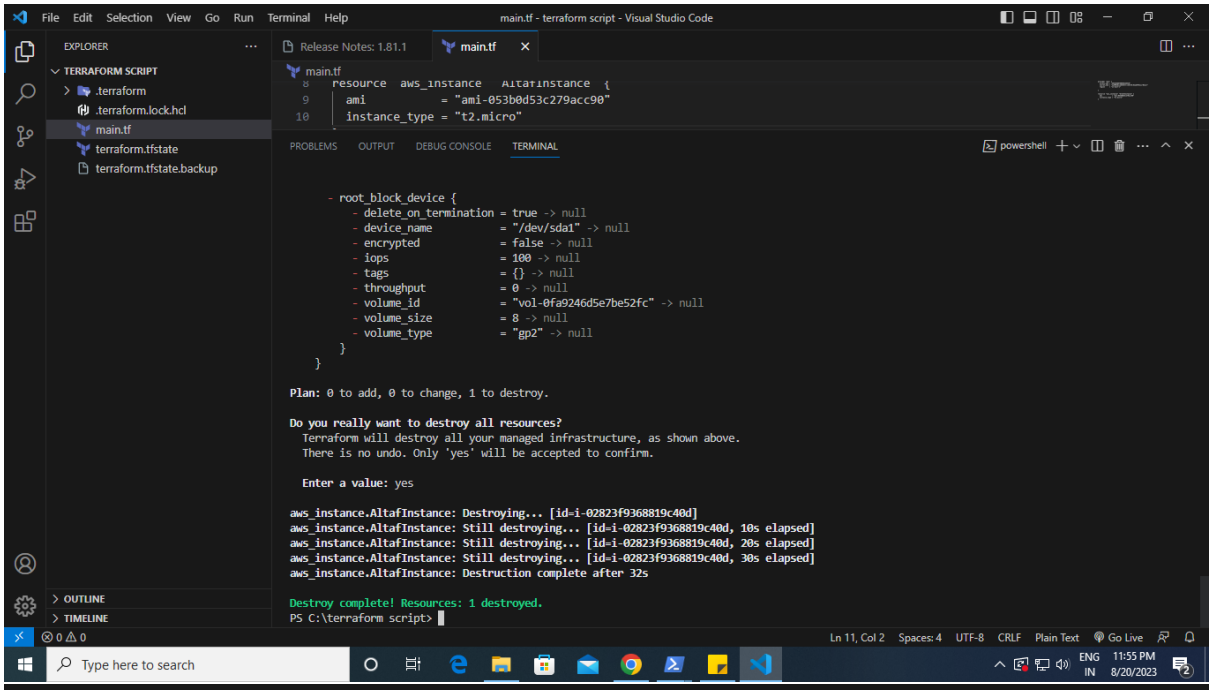
Name : Altaf Alam

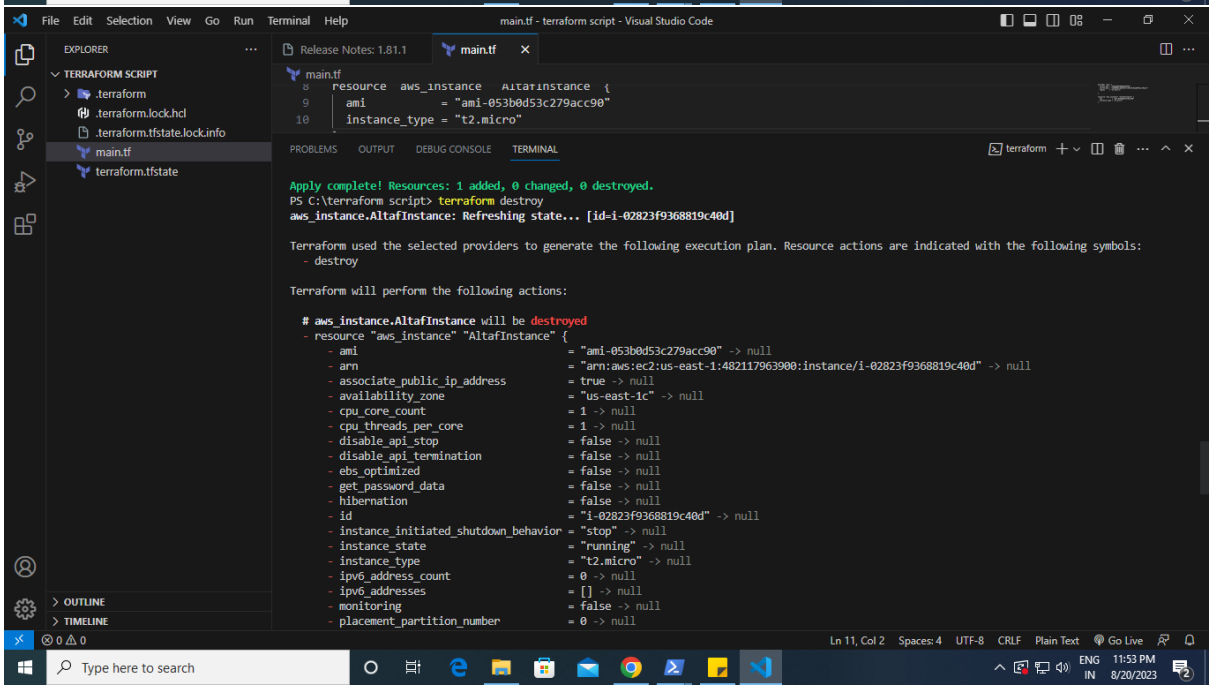
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23



Name : Altaf Alam

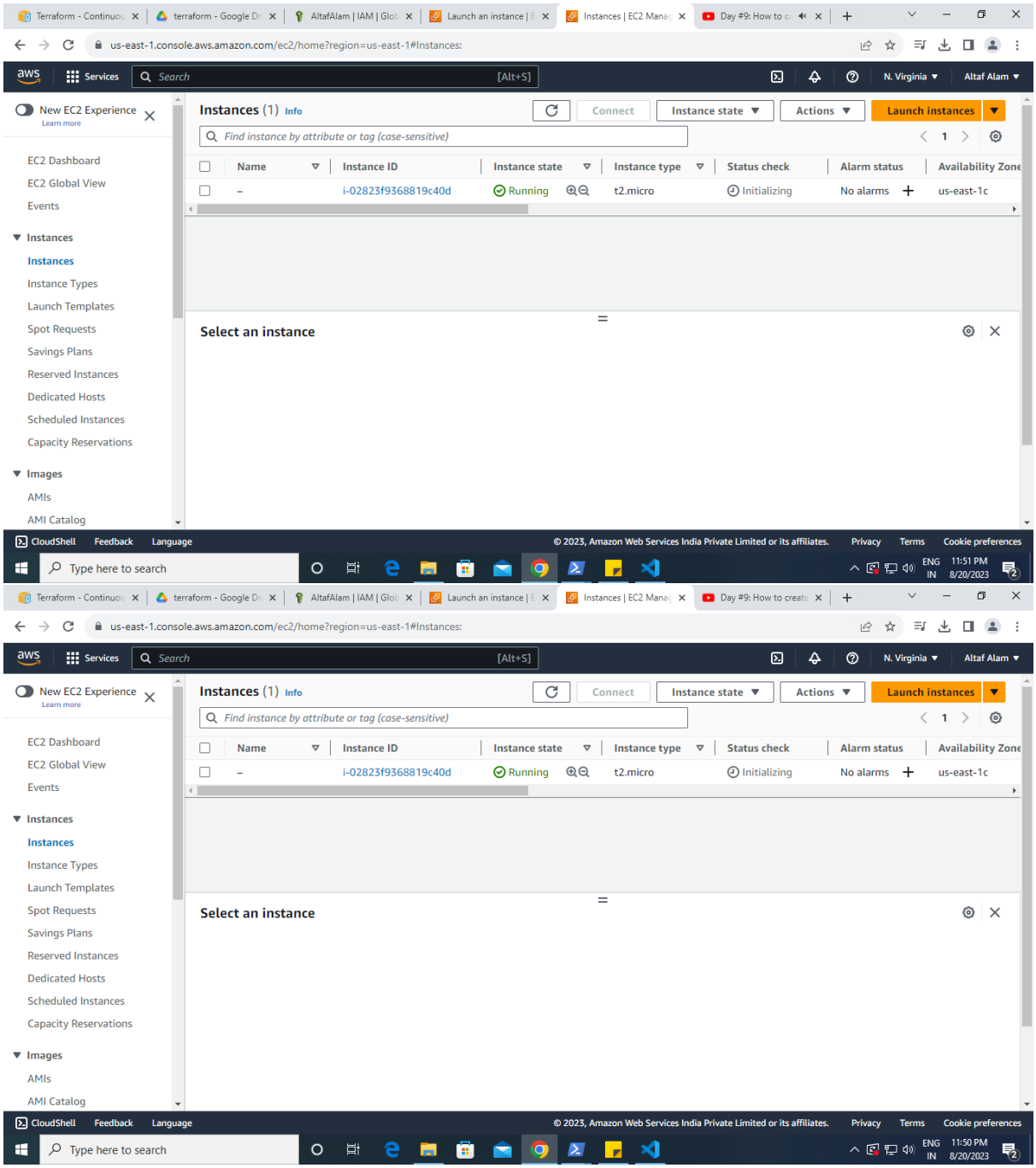
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23





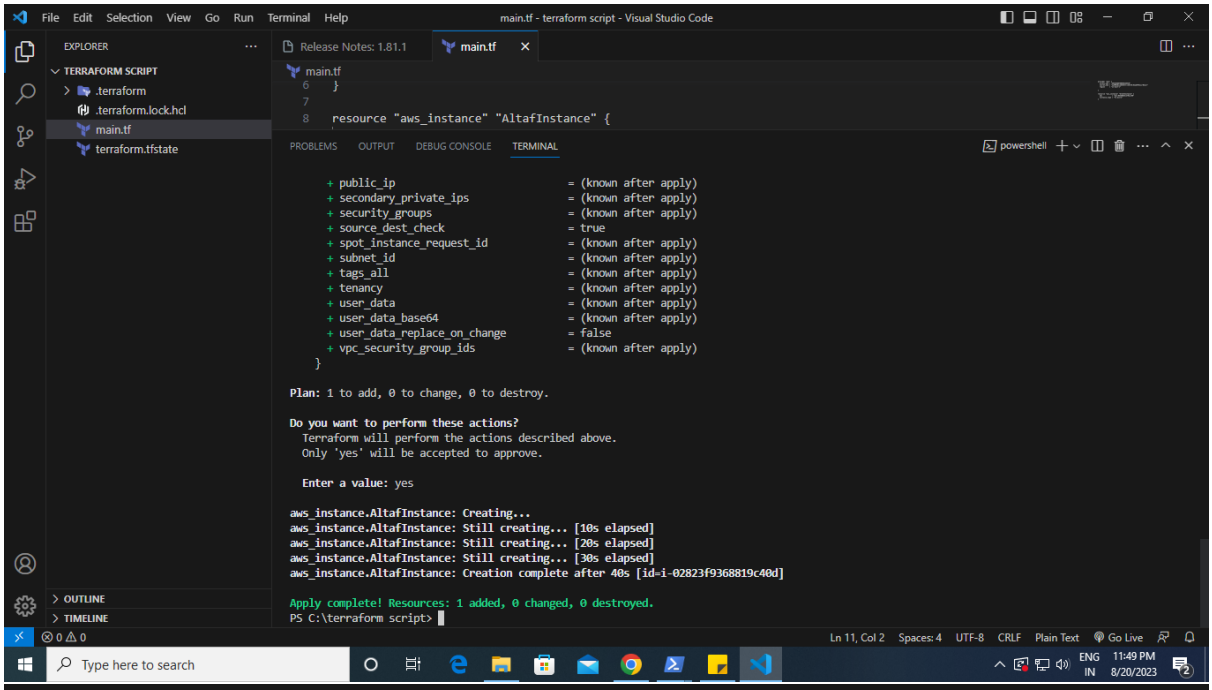
Name : Altaf Alam

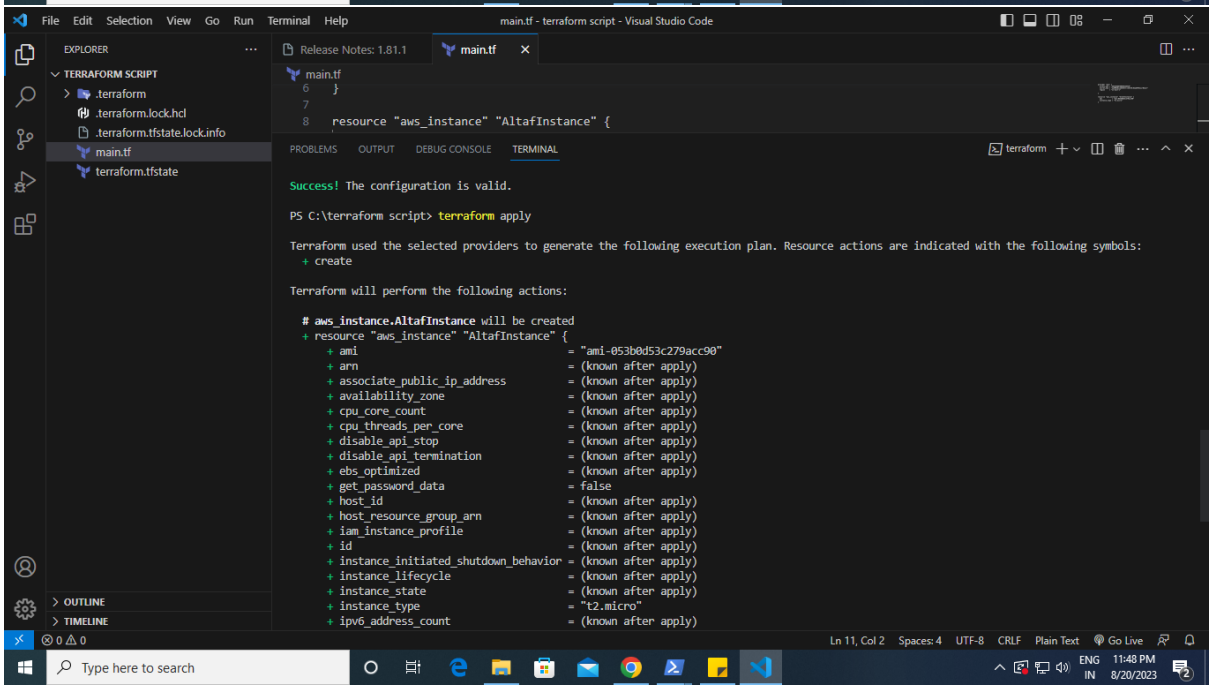
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23



Name : Altaf Alam

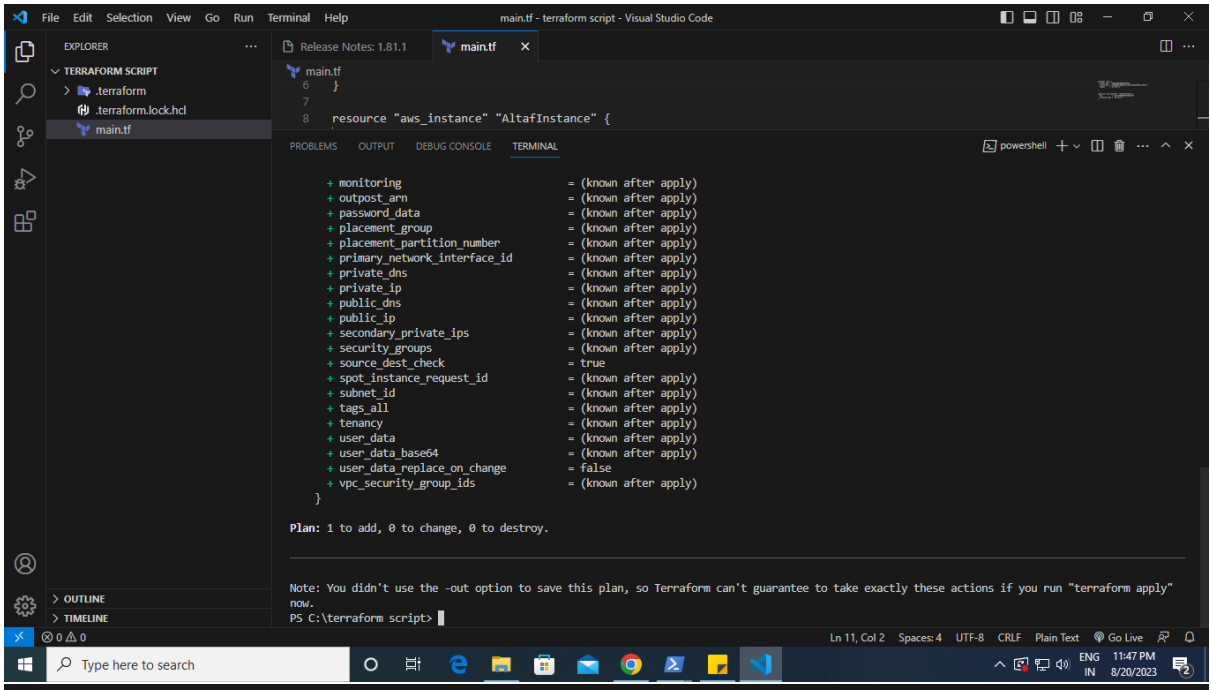
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23

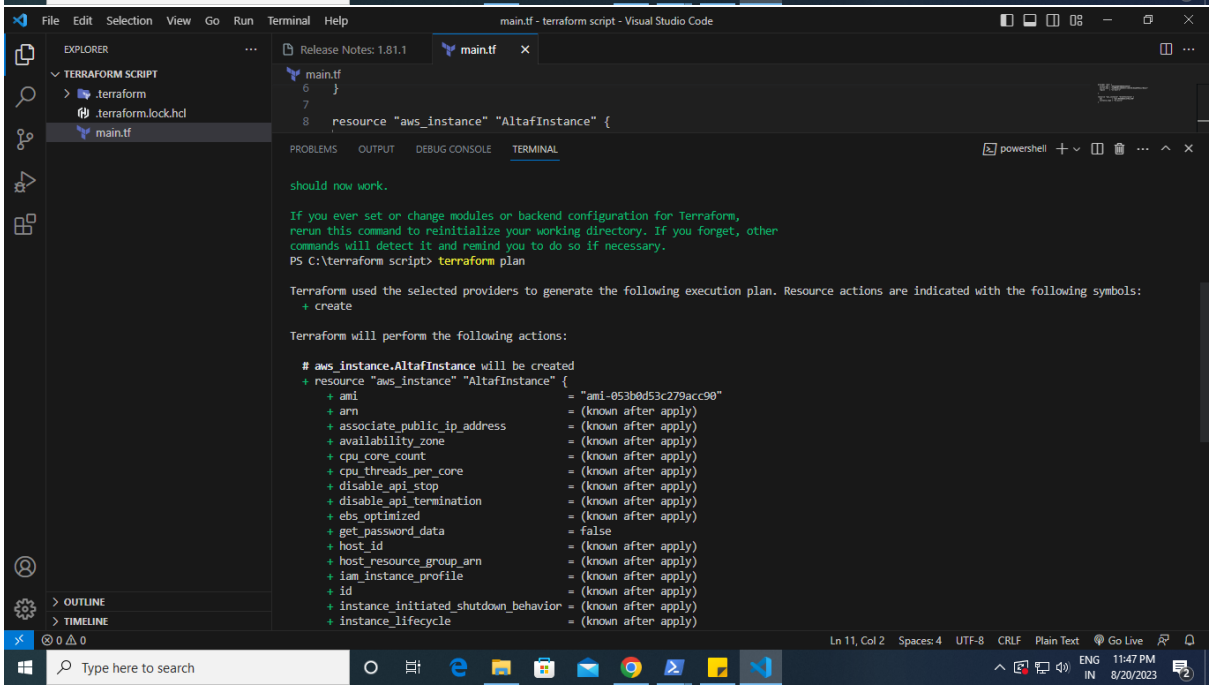




Name : Altaf Alam

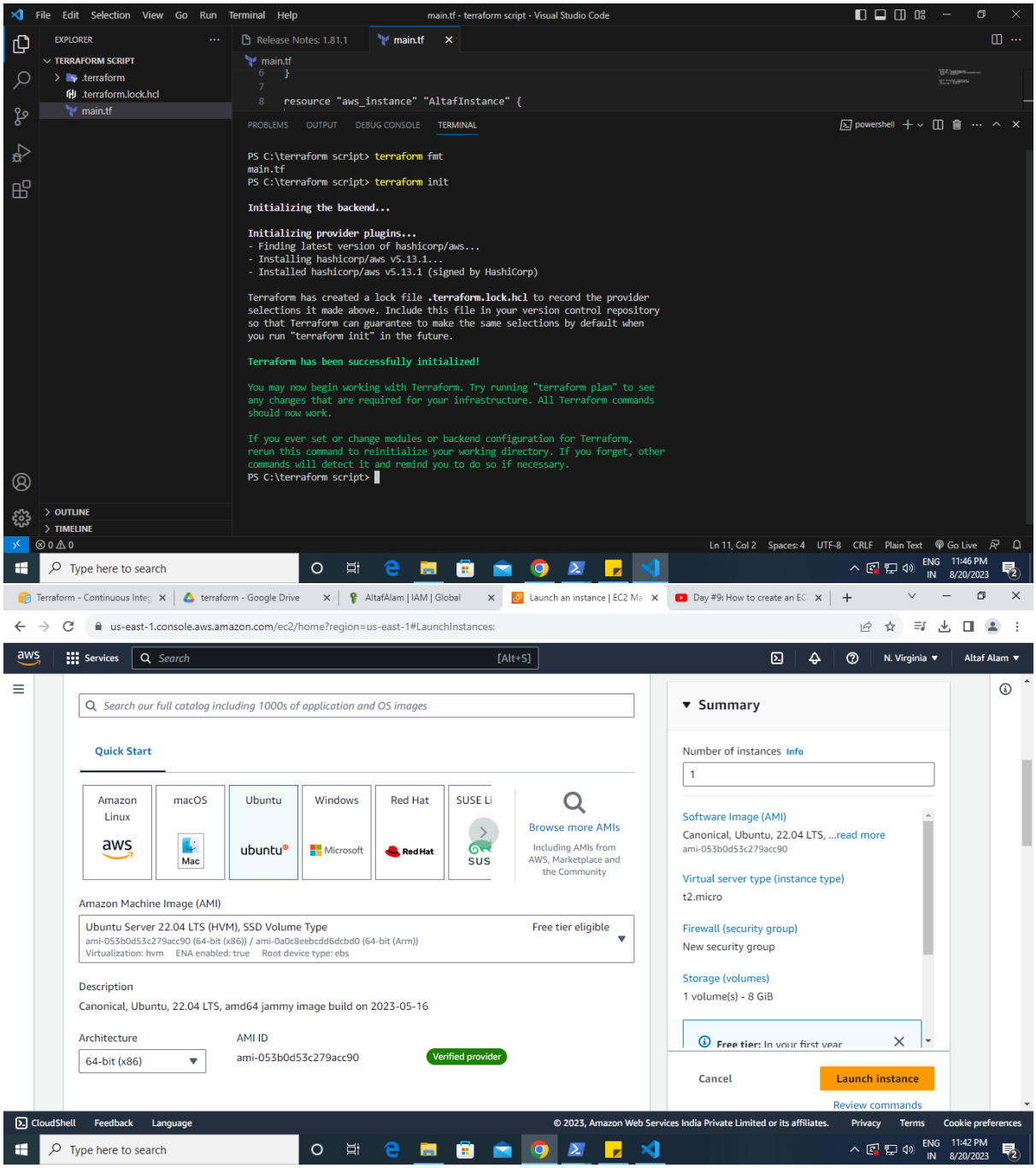
Roll no : 02 , Batch : T11 , Subject : Adv DevOps Lab , Date : 28/08/23





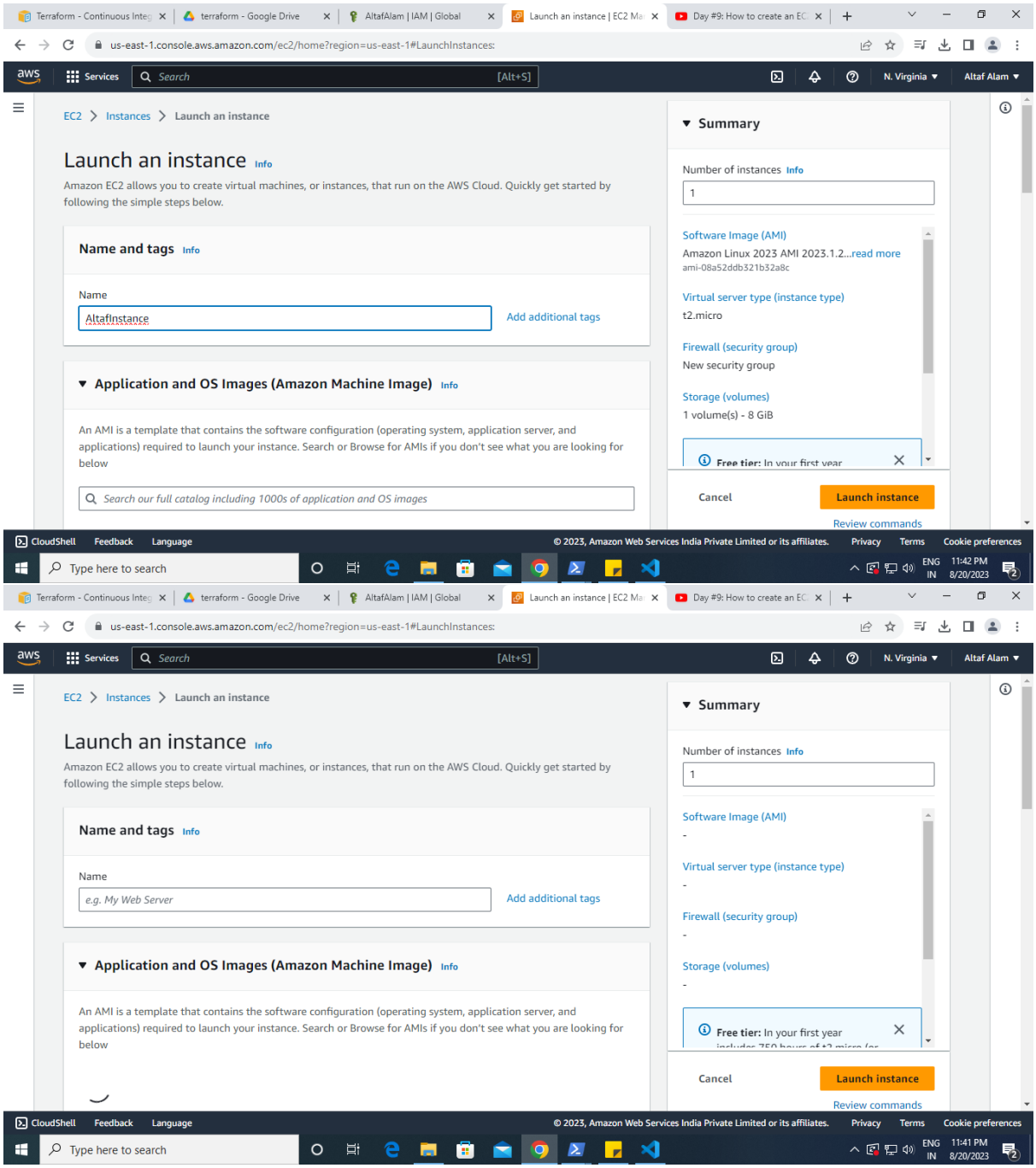
Name : Altaf Alam

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



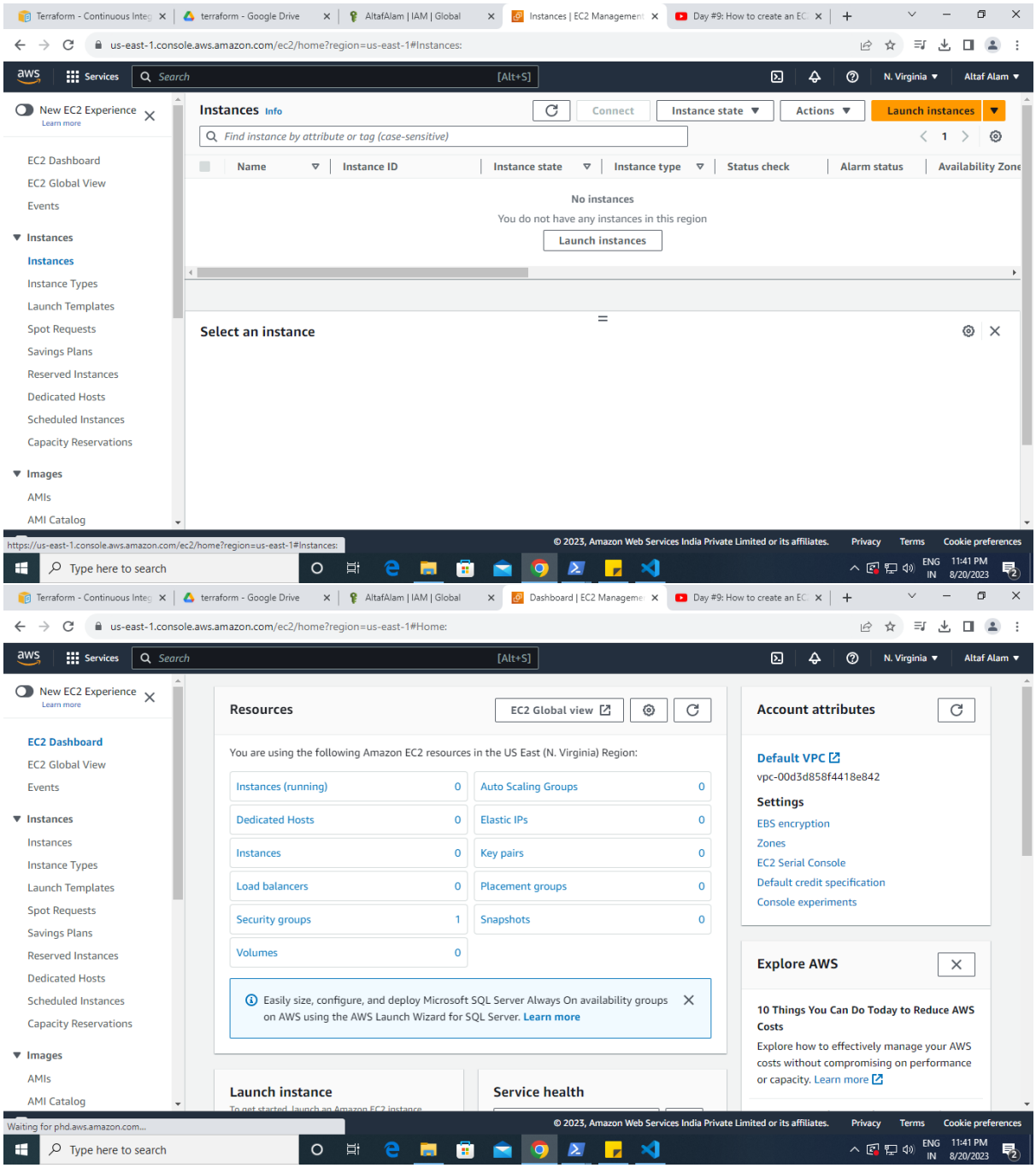
Name : Altaf Alam

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



Name : Altaf Alam

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



CONCLUSION :

Altaf Alam , 02 , T11

Name : Altaf Alam

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

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

LAB Assignment 5

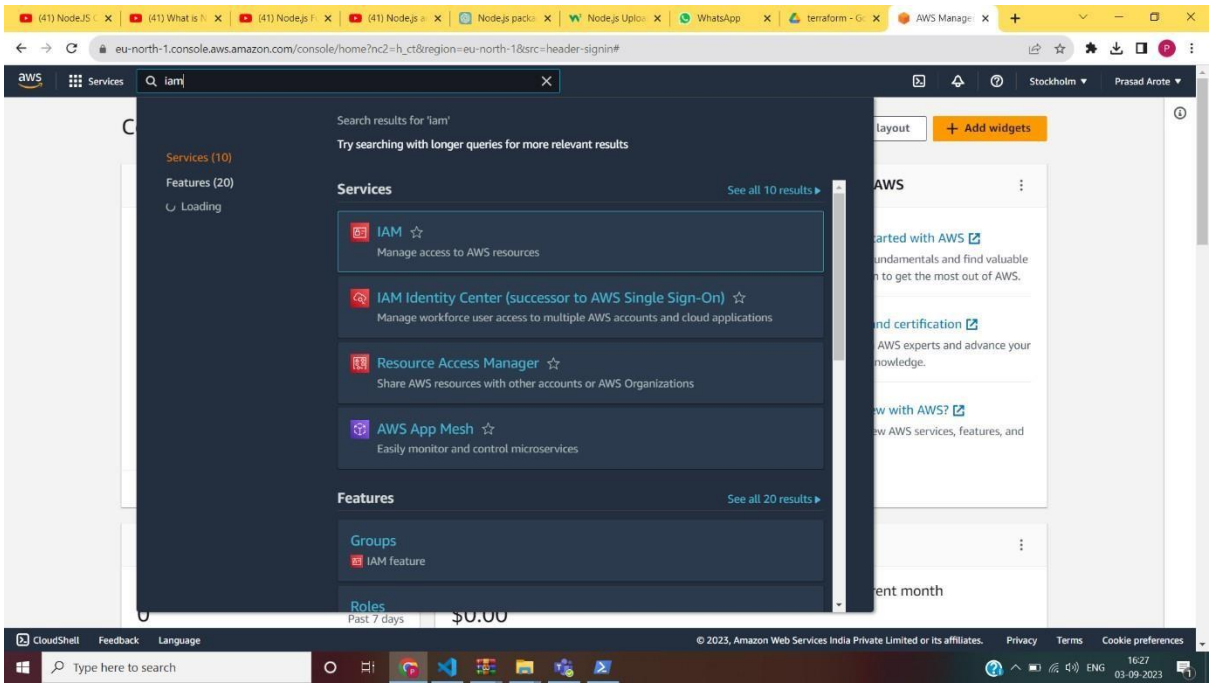
AIM: To Build, change, and destroy AWS infrastructure Using Terraform.

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

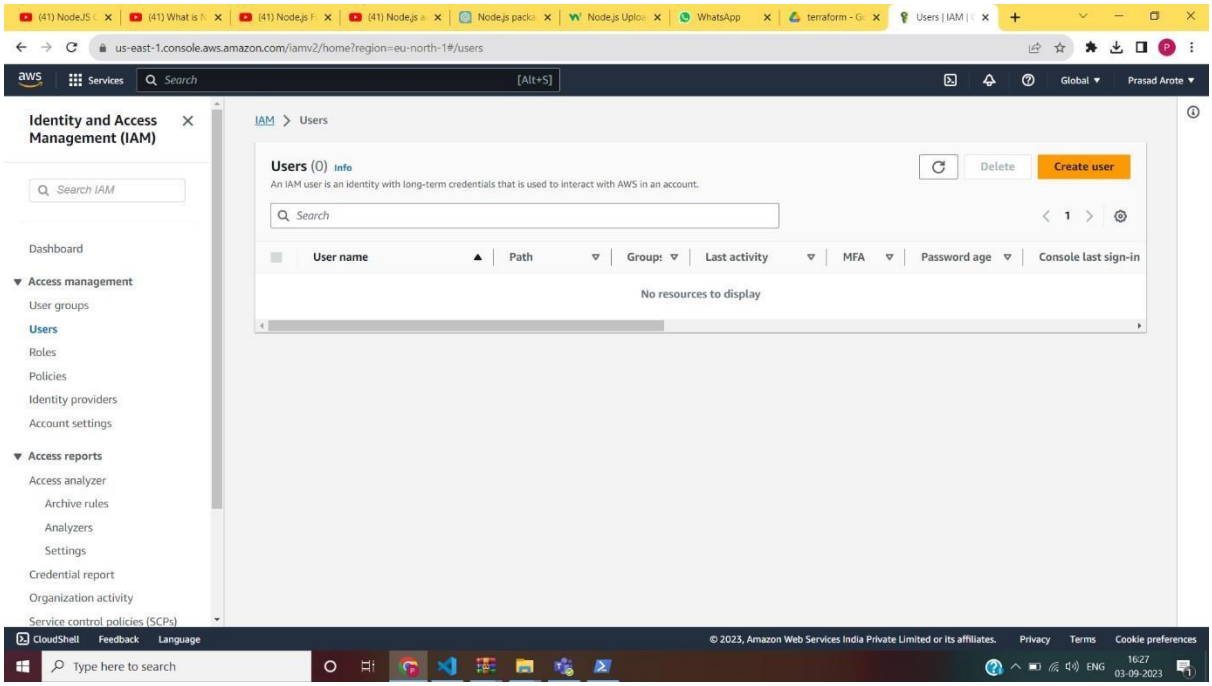
LO5: To use Continuous Monitoring Tools to resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on the business productivity.

Theory:

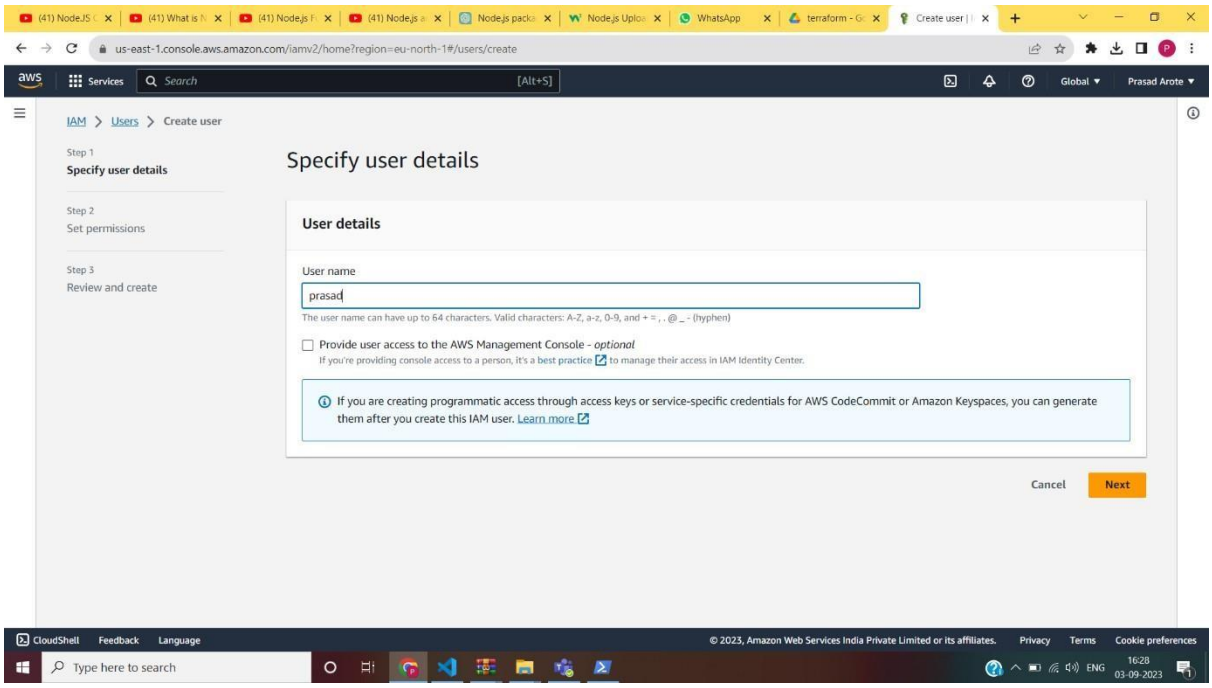
1) Make dir Terraform Scripts Open aws.amazon.com
   Login to your account
   Search IAM



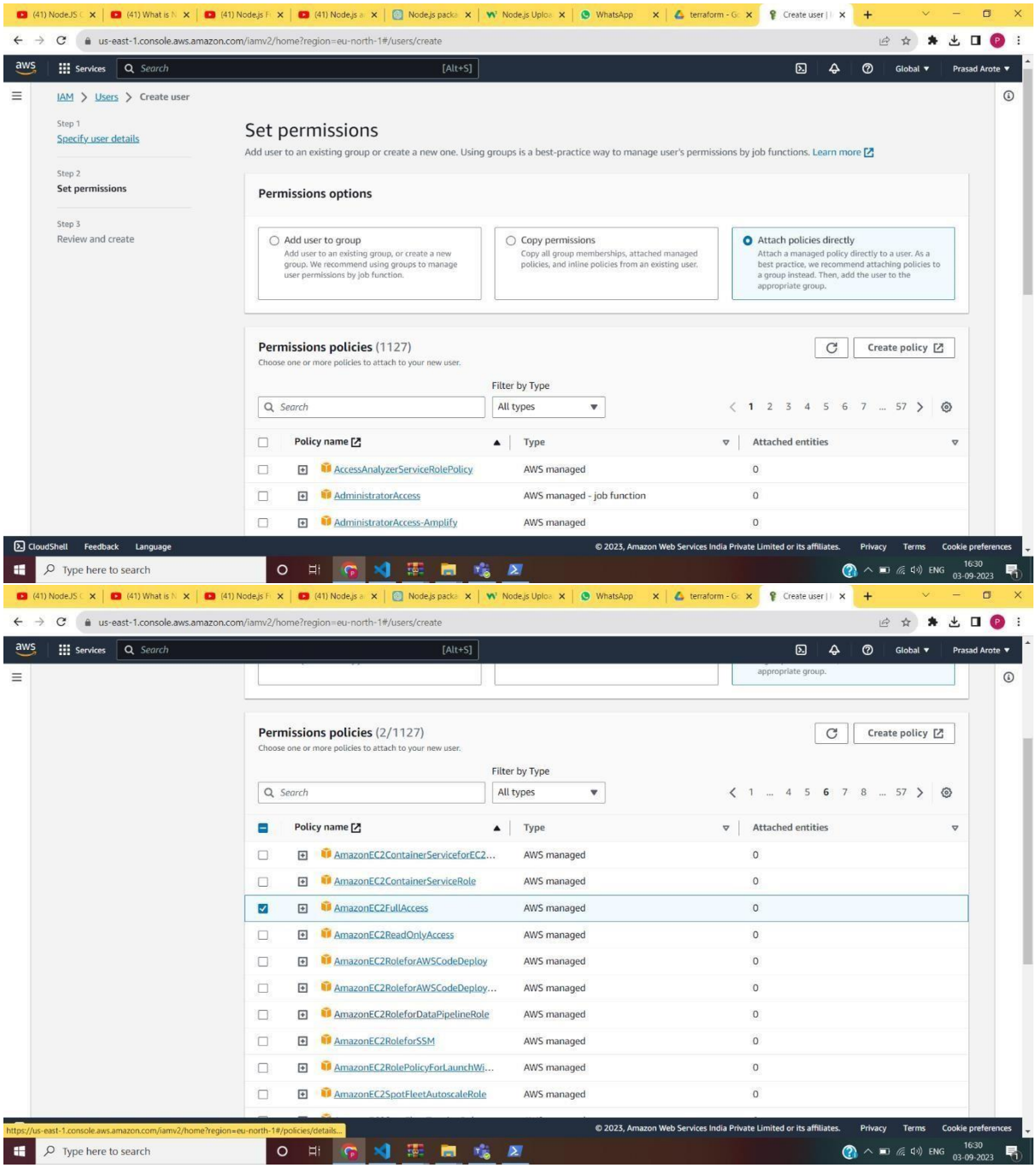
2) Click on Users (on the LHS)



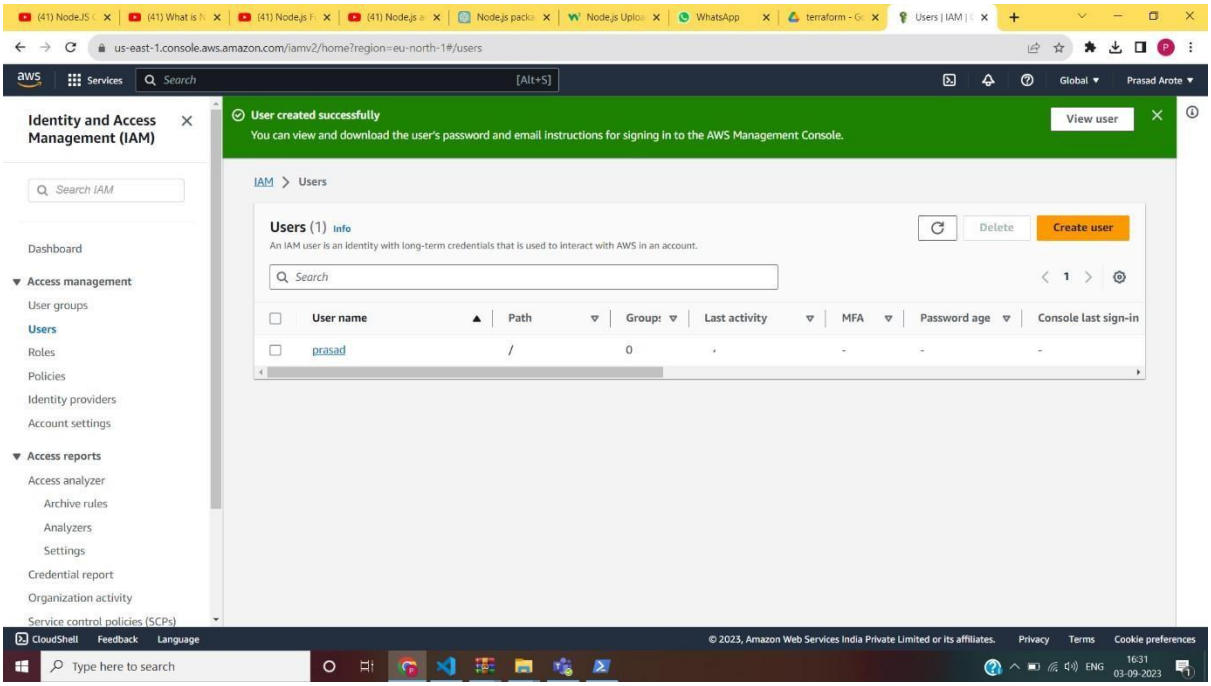
3) Click Add users



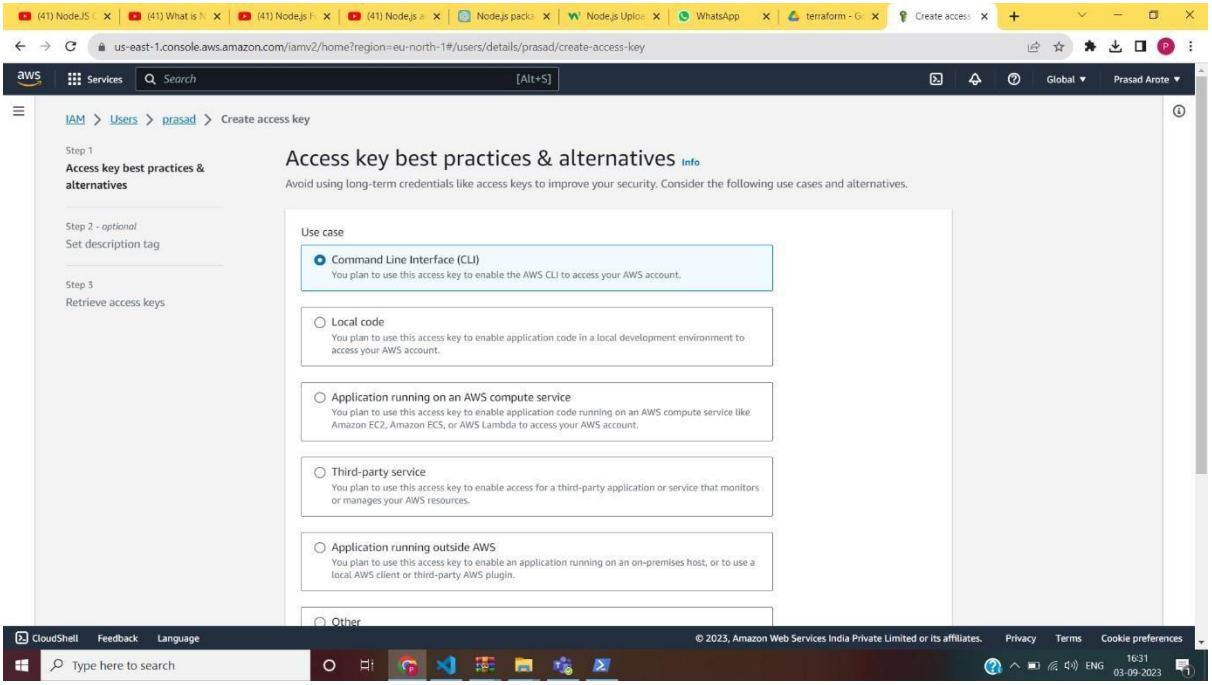
4) Set Permissions -> AmazonEC2FullAccess



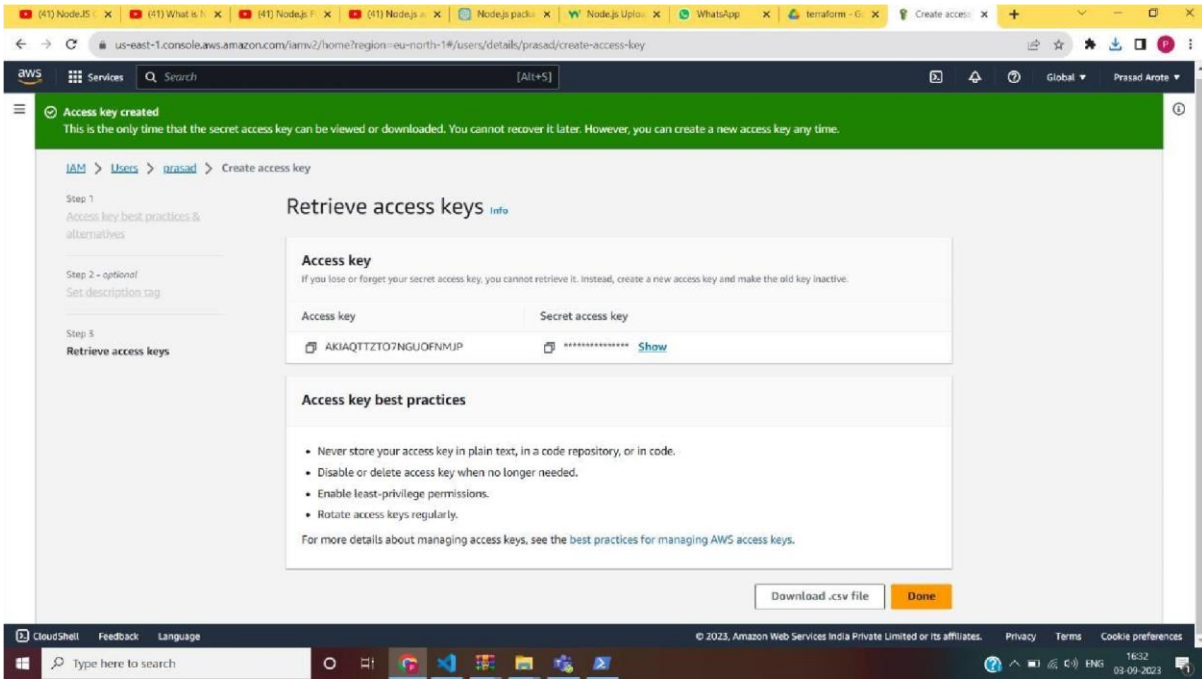
5) Create User



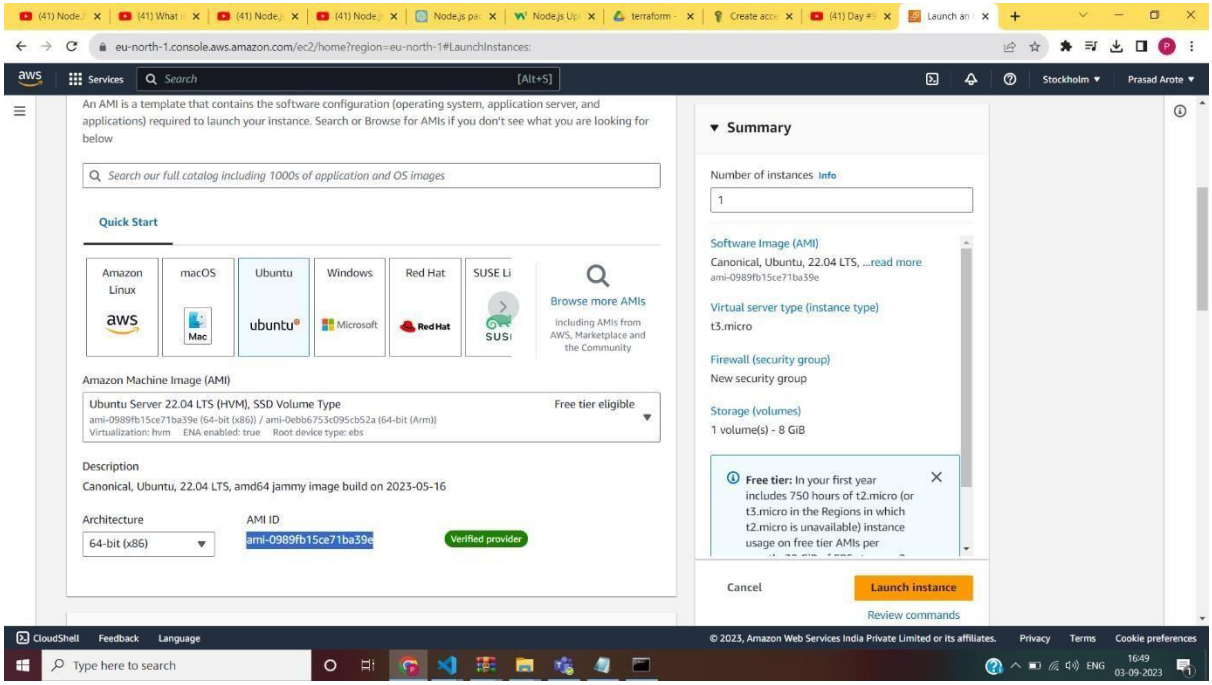
6) Create access key



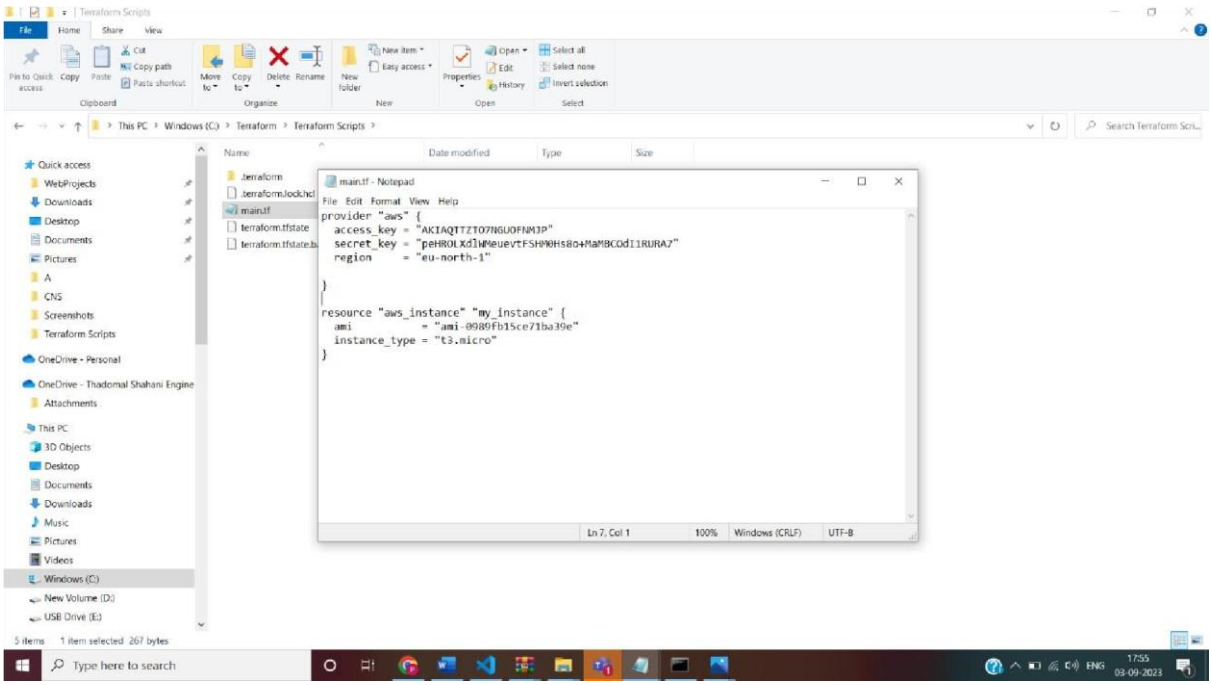
7) Download the .csv file



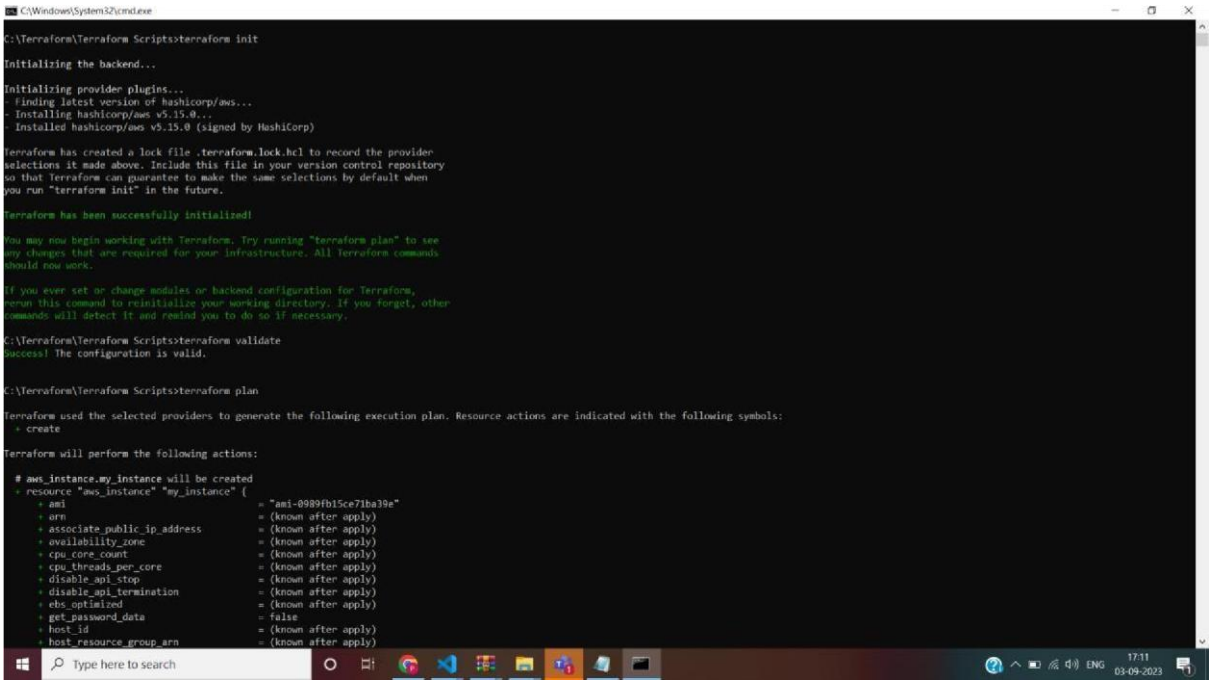
8) Open EC2 Instances and Copy the AMI ID of any one of them.

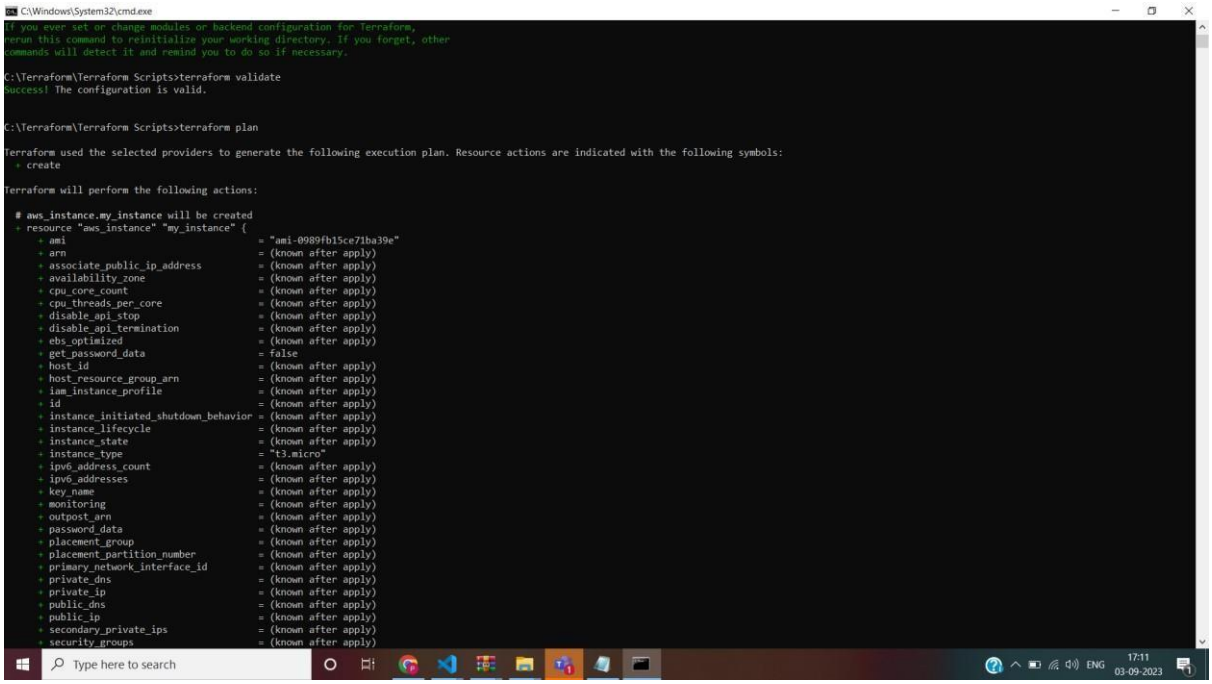


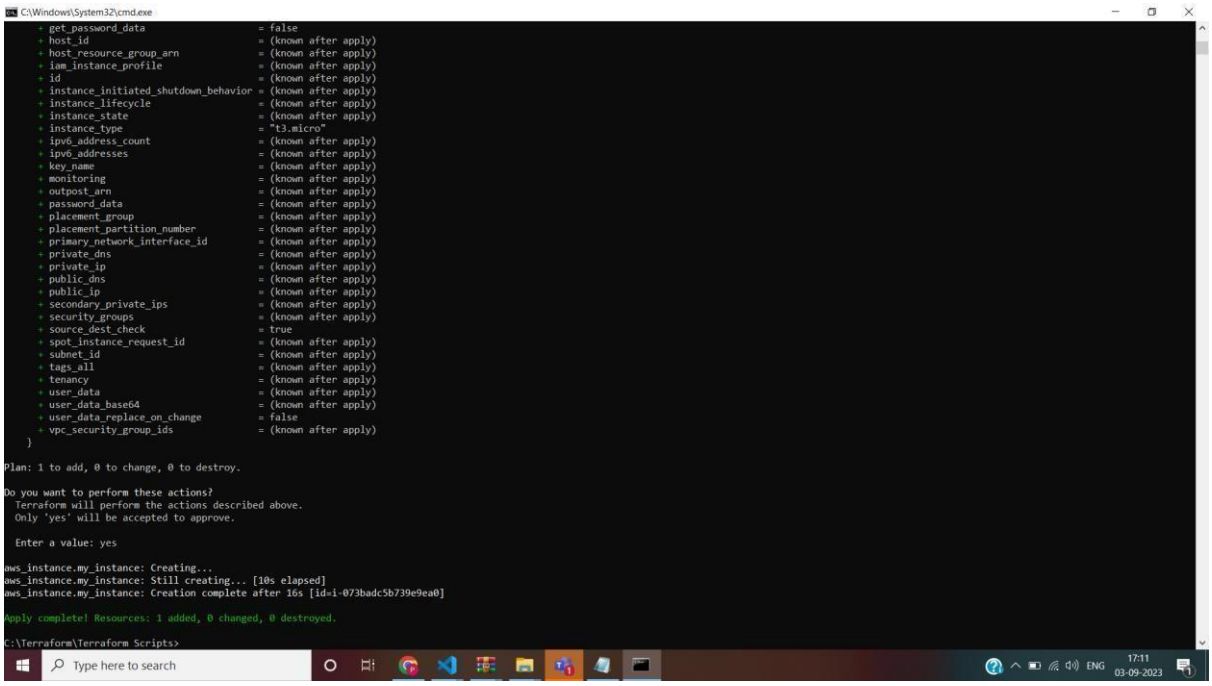
9) Configure the main.tf file

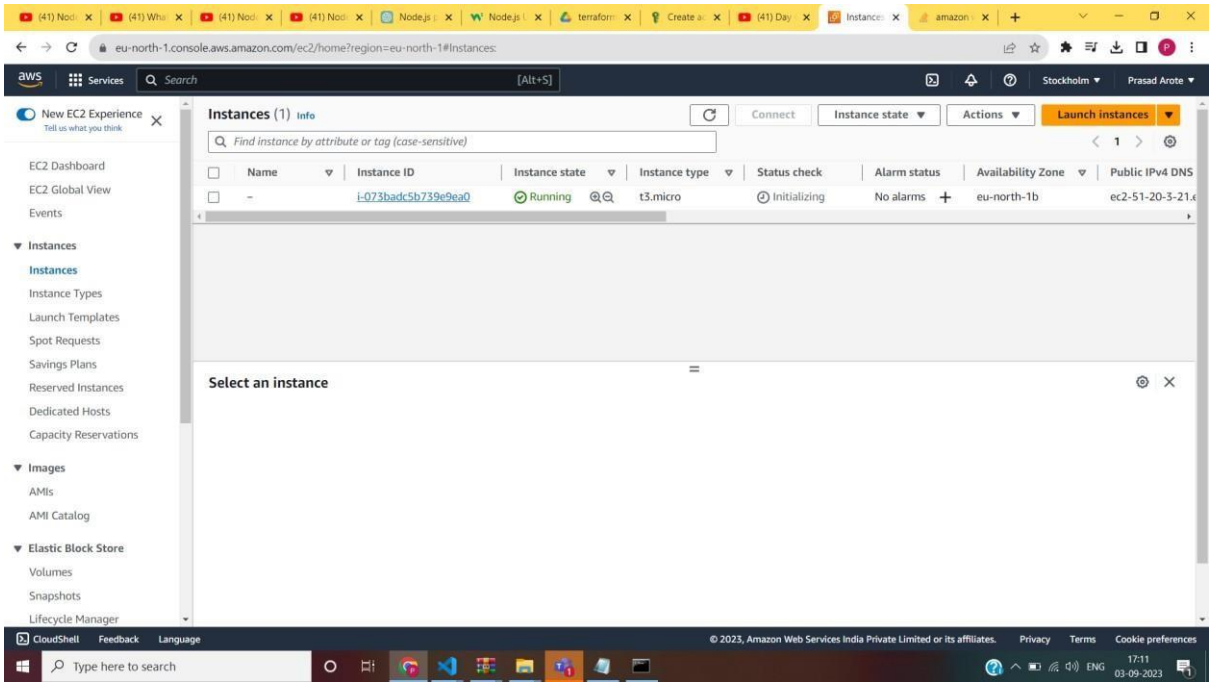


10) Run the commands in cmd -> terraform init , terraform validate , terraform plan, terraform apply to create EC2 instance using terraform.

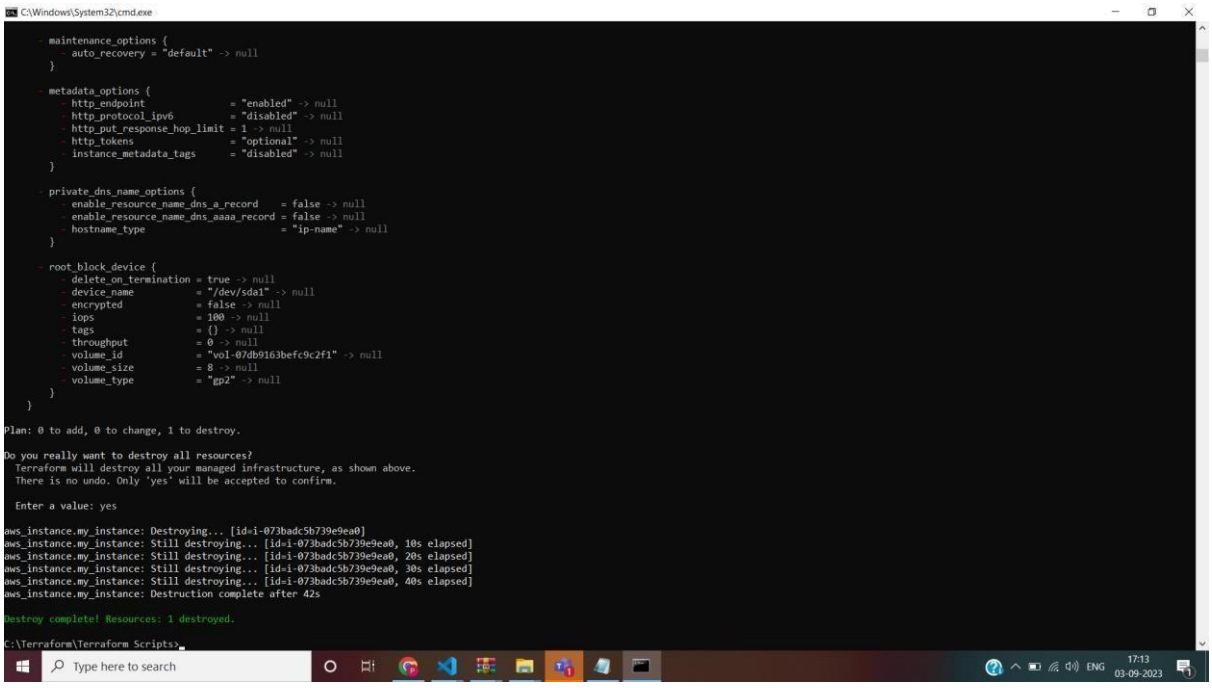



C:\Windows\System32\cmd.exe
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
C:\Terraform\Terraform Scripts>terraform validate
Success!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

Lab Assignment 6

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

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

THEORY:

SonarQube:

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

Features:

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

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

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

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

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

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

SAST (Static Application Security Testing):

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

Features:

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

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

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

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

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

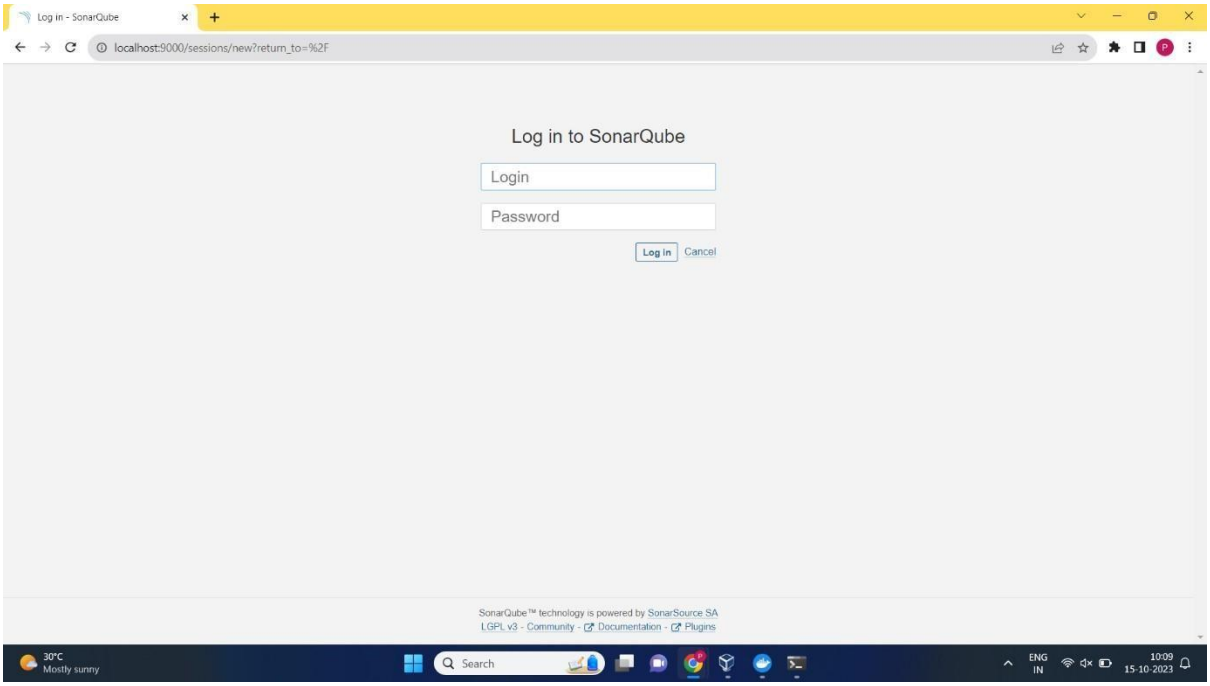
1. INSTALL sonarqube (docker images) and sonarscanner zip file from https://docs.sonarsource.com/sonarqube/latest/analyzingsourcecode/scanners/sonarscanner/ and set up config file as given in docs.

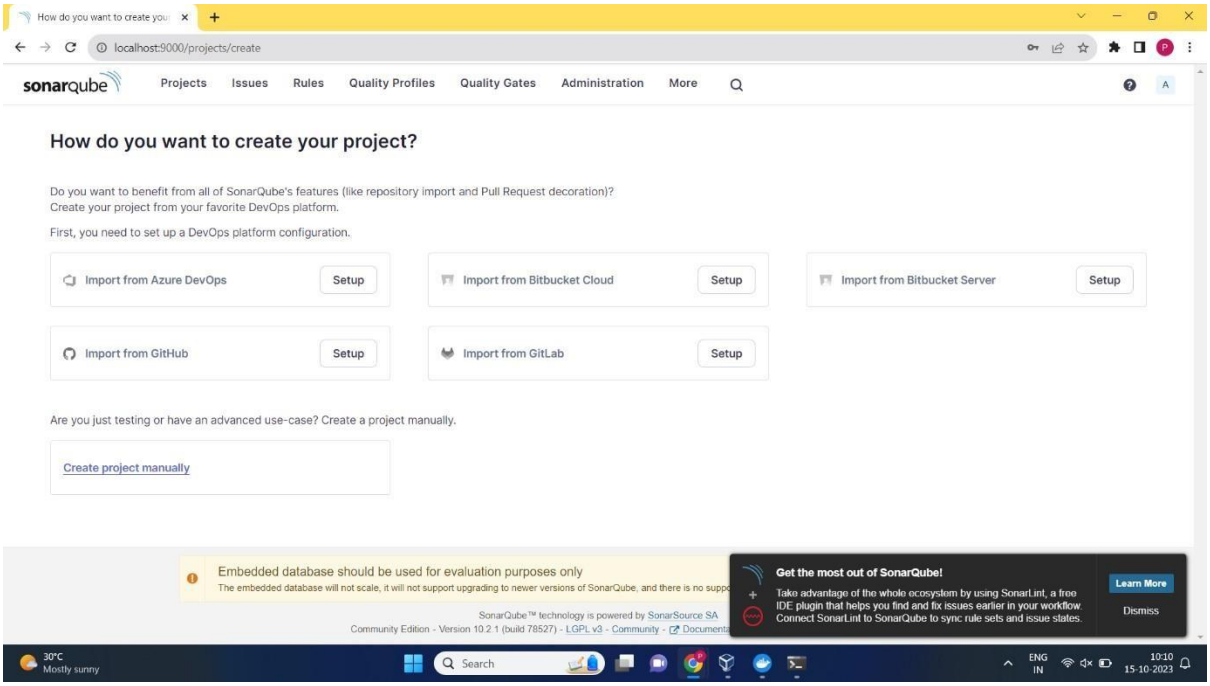
```
Command Prompt
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

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

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

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

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

2. Spin up the container

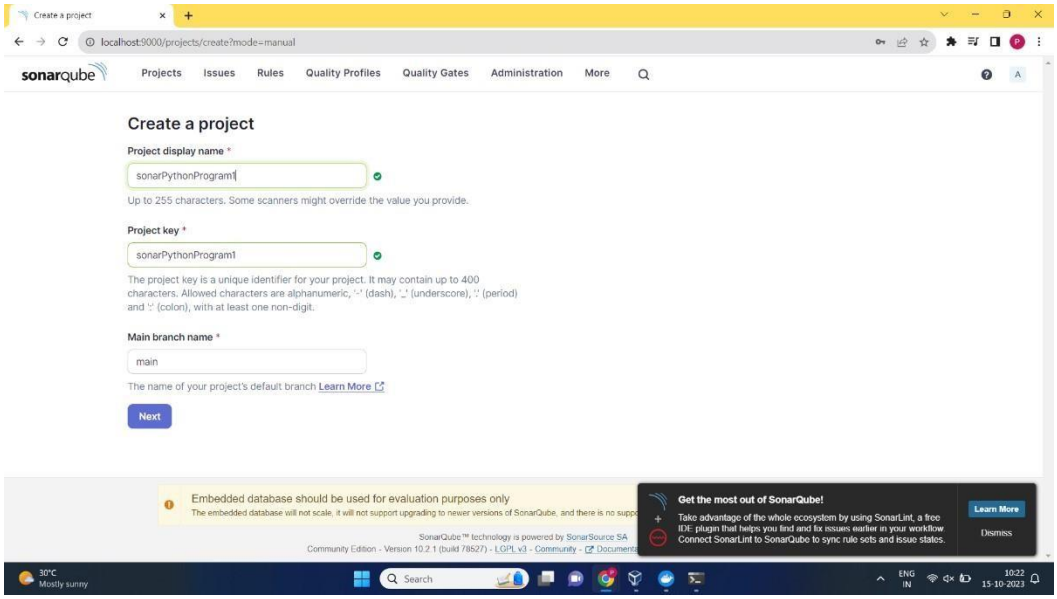
```
C:\Users\Pratik Arote>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
f3630dbc2ffa6e5598ad922085026400a1f9f1564416b0606b5348000f6d1377

C:\Users\Pratik Arote>docker images
REPOSITORY                     TAG       IMAGE ID       CREATED        SIZE
sonarqube                      latest    3183d6818c6e   42 hours ago   716MB
sample-web-app                 latest    713c7cdaaf78   2 weeks ago    42.7MB
myimage                        latest    438bb56a50a3   2 weeks ago    122MB
sonarsource/sonar-scanner-cli  latest    2f384fb1bbd5   5 weeks ago    358MB
ubuntu                         latest    c6b84b685f35   8 weeks ago    77.8MB
hello-world                    latest    9c7a54a9a43c   5 months ago   13.3kB

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

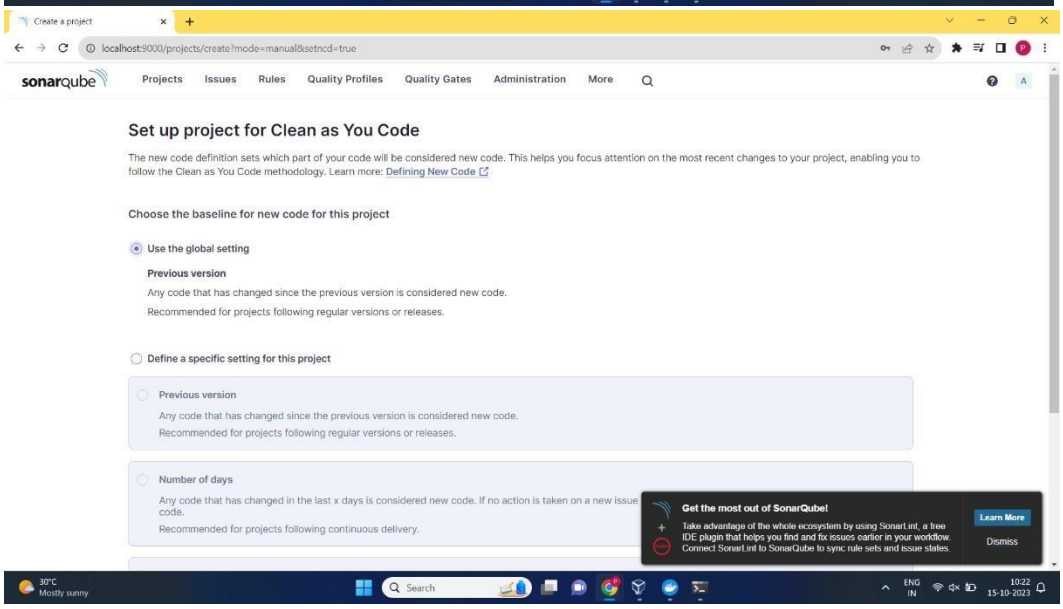
3. Open http://localhost:9000 on the browser. Enter login and password both as “admin” and then set up new password.

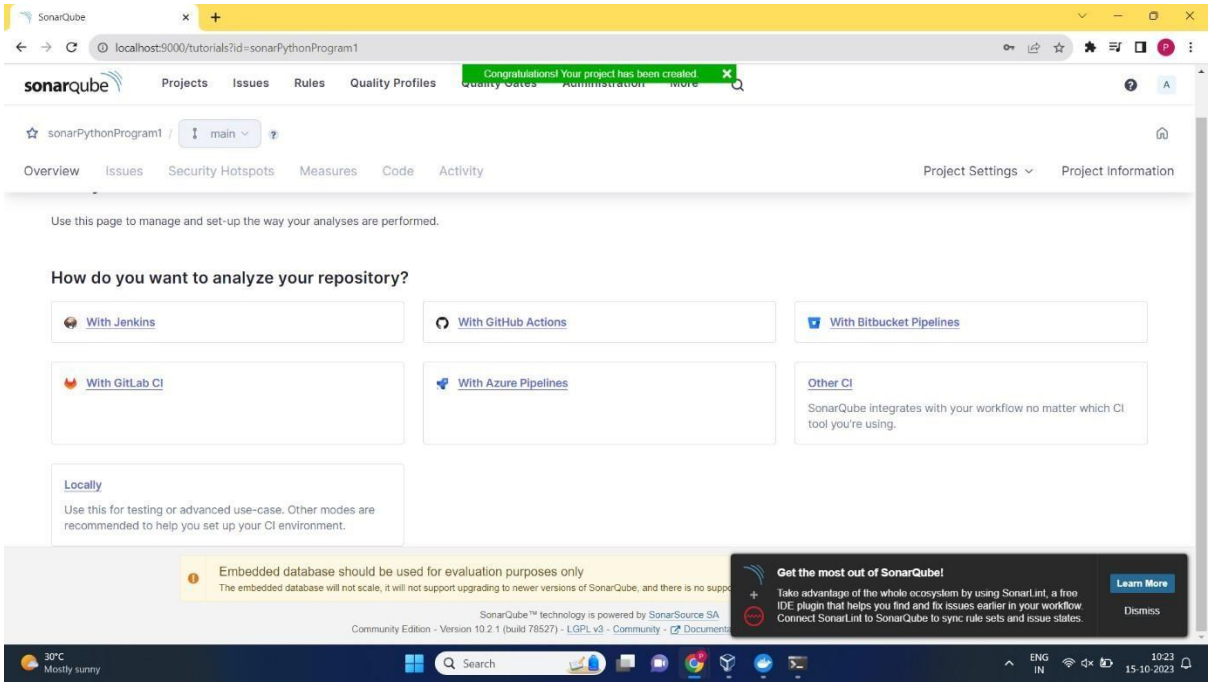




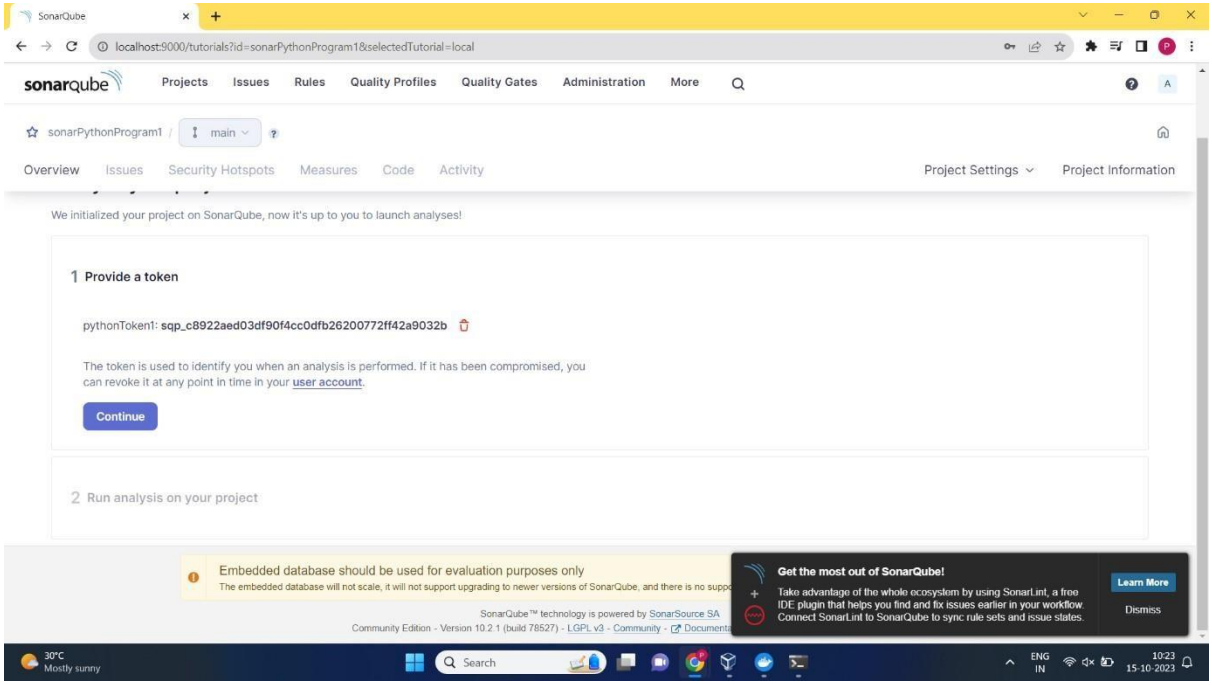
4. Create a project

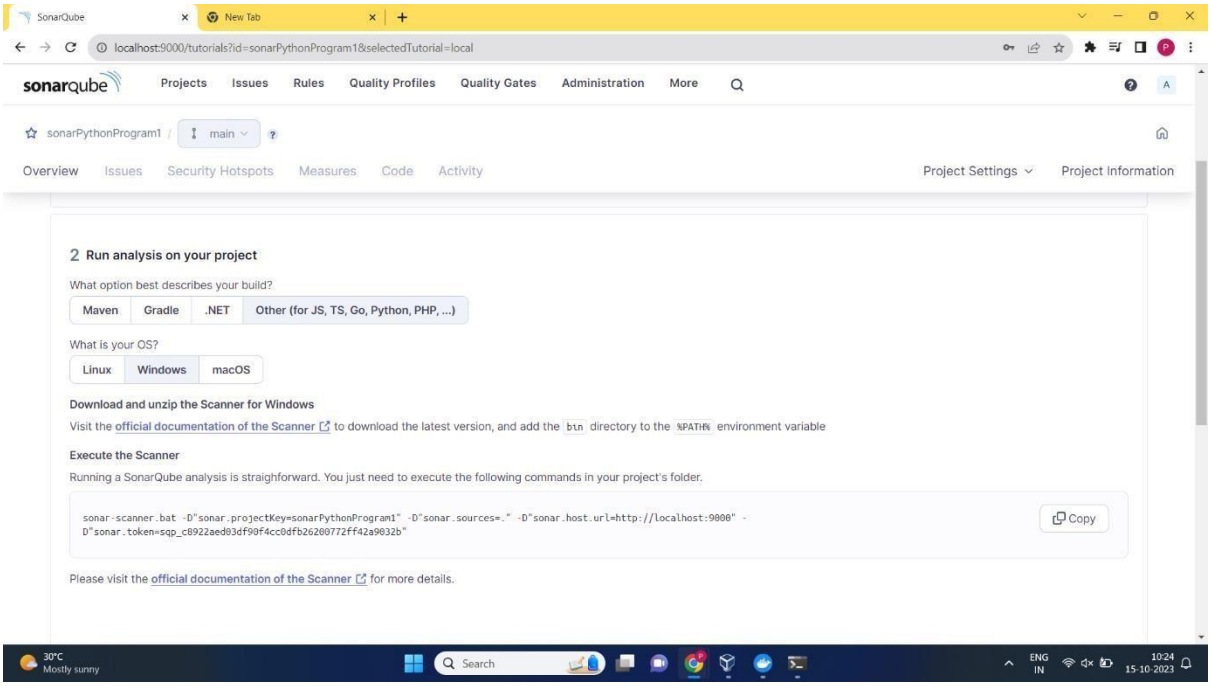

Create a project
localhost:9000/projects/create?mode=manual
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Create a project
Project display name *
sonarPythonProgram1
Up to 255 characters. Some scanners might override the value you provide.
Project key *
sonarPythonProgram1
The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.
Main branch name *
main
The name of your project's default branch Learn More
Next
Embedded database should be used for evaluation purposes only
Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.
Learn More
Dismiss
SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 10.2.1 (build 78527) - LGPL v3 - Community
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

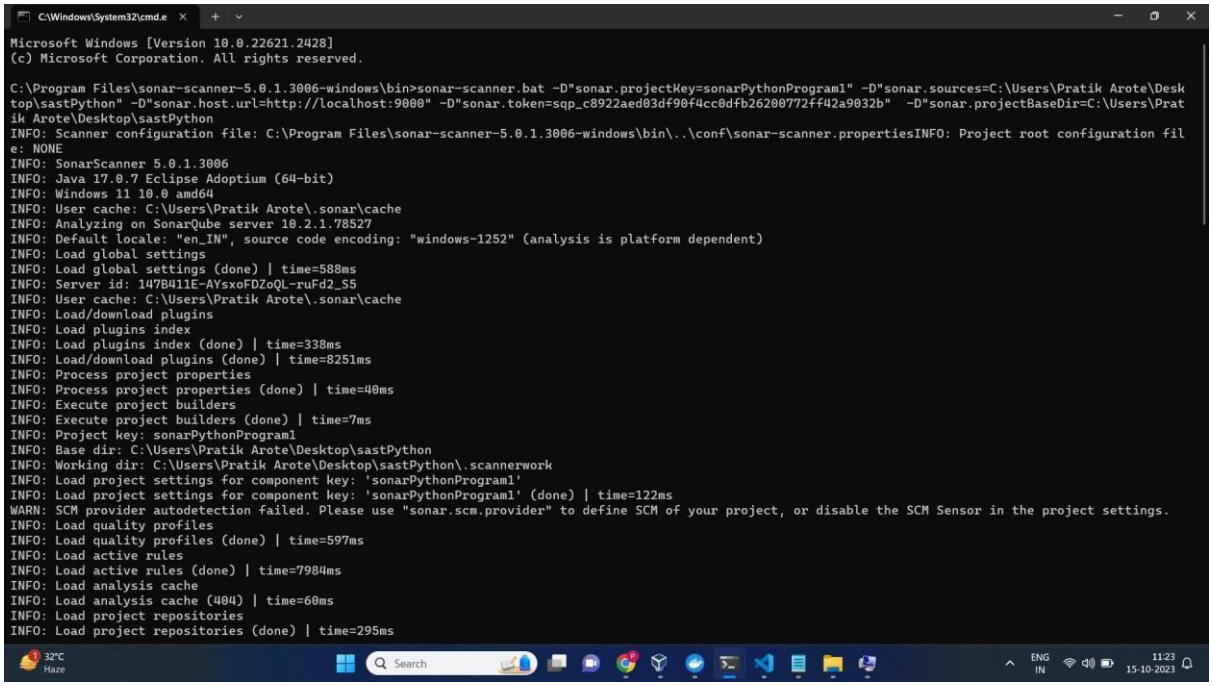


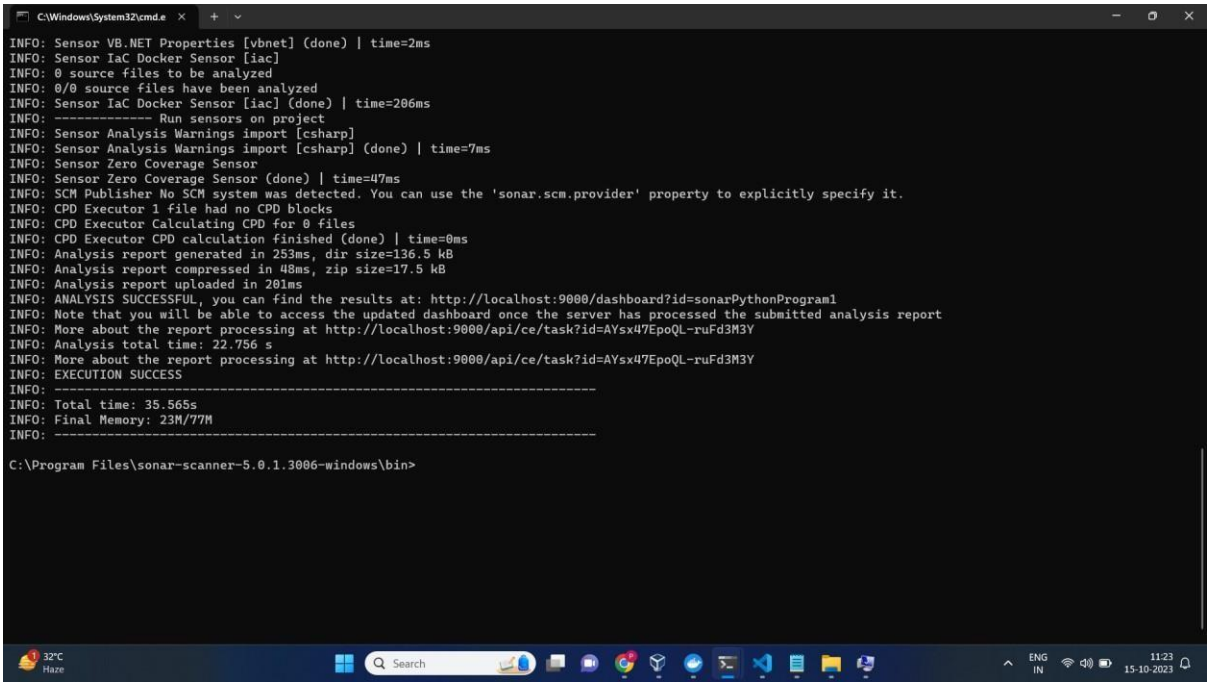
5. Provide token



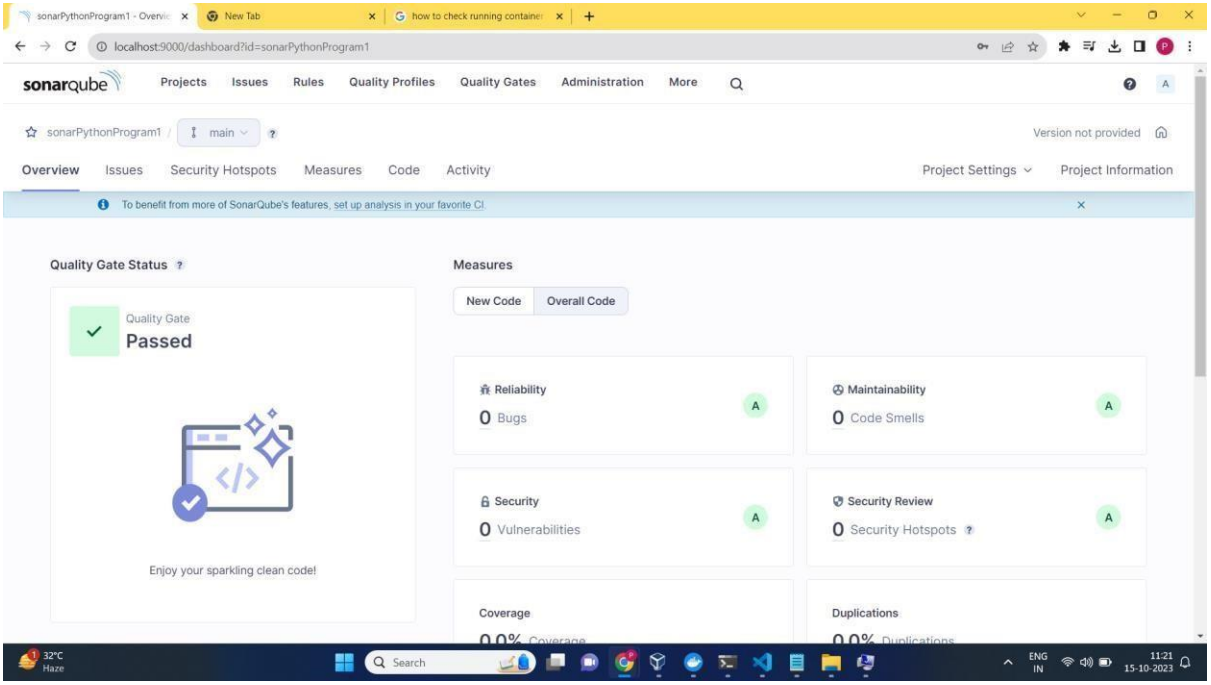


6. Enter the following command





7. See the result of the test



CONCLUSION:

Here we have successfully performed static analysis of python programs.

Lab Assignment 7

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

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

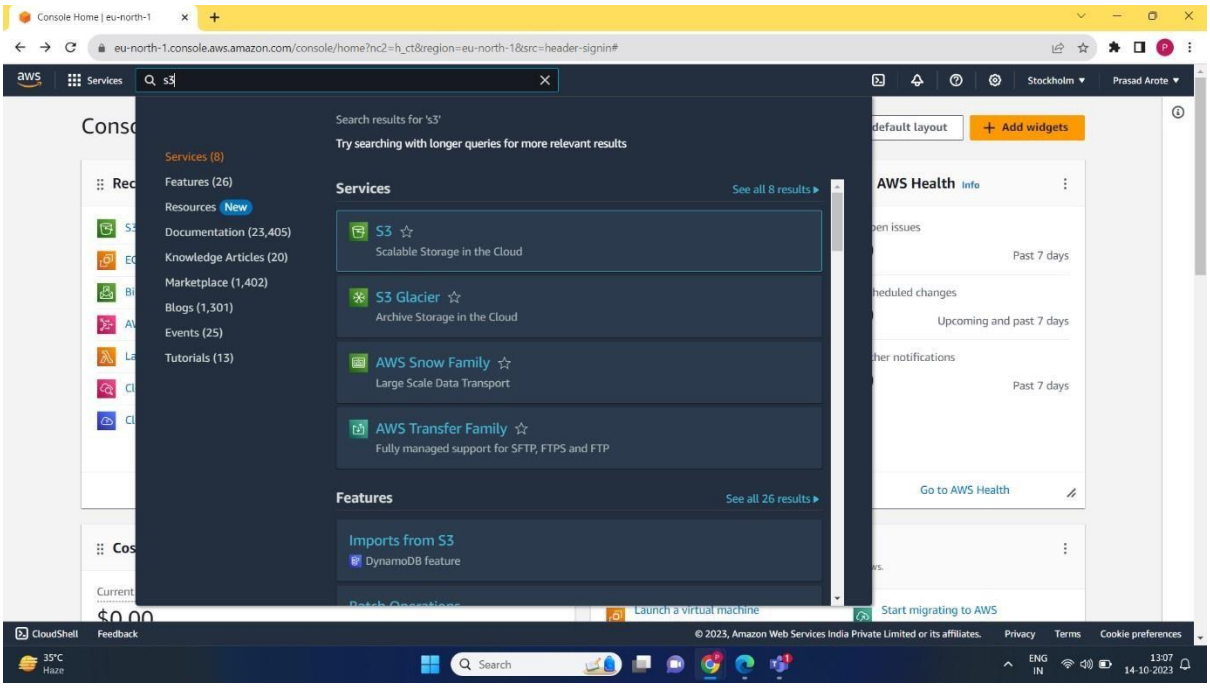
THEORY:

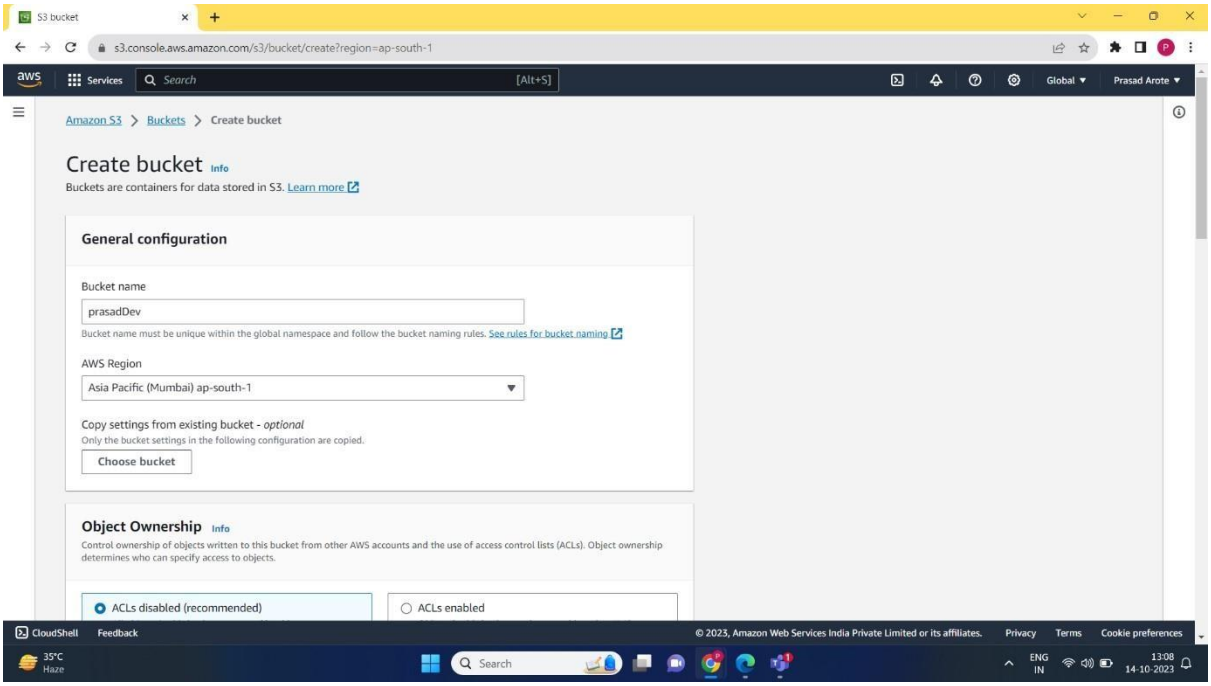
LAMBDA FUNCTION

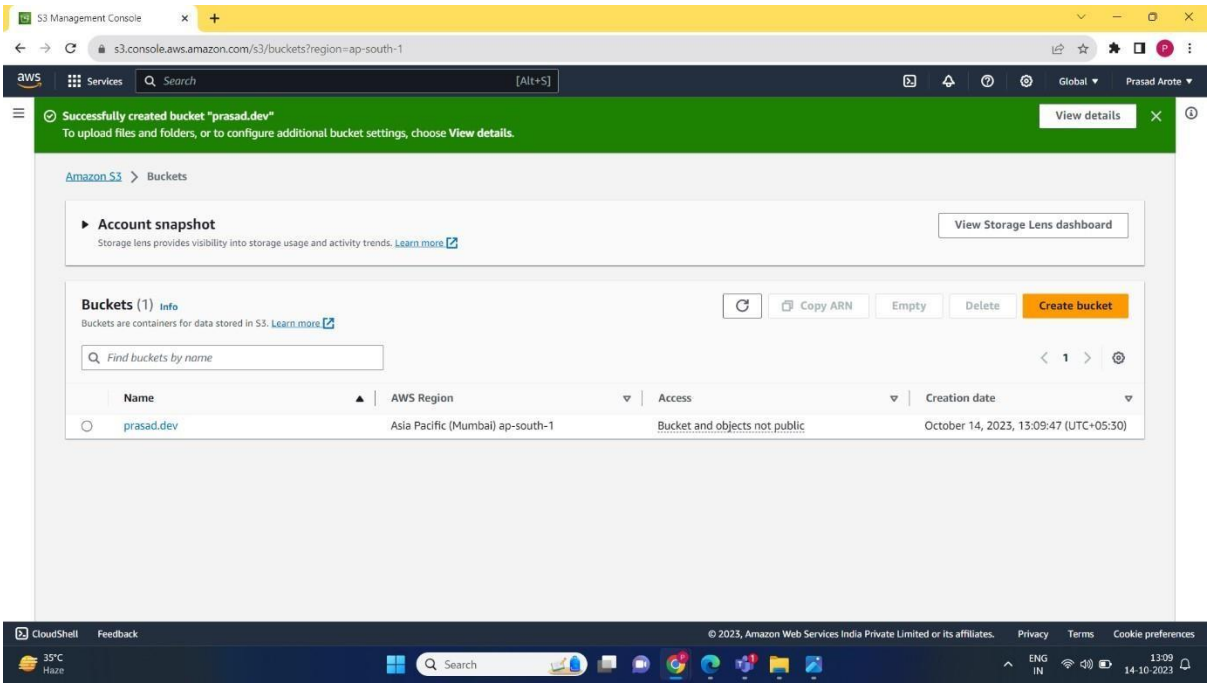
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you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

Installation:

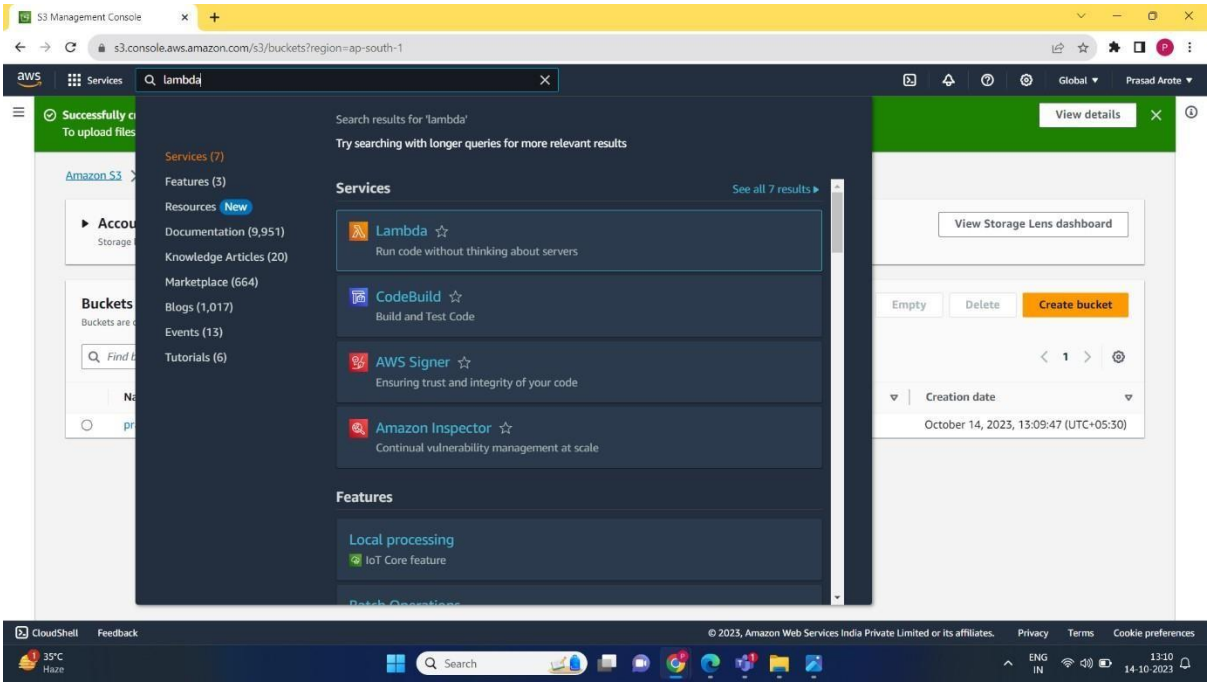
1. Create a S3 bucket

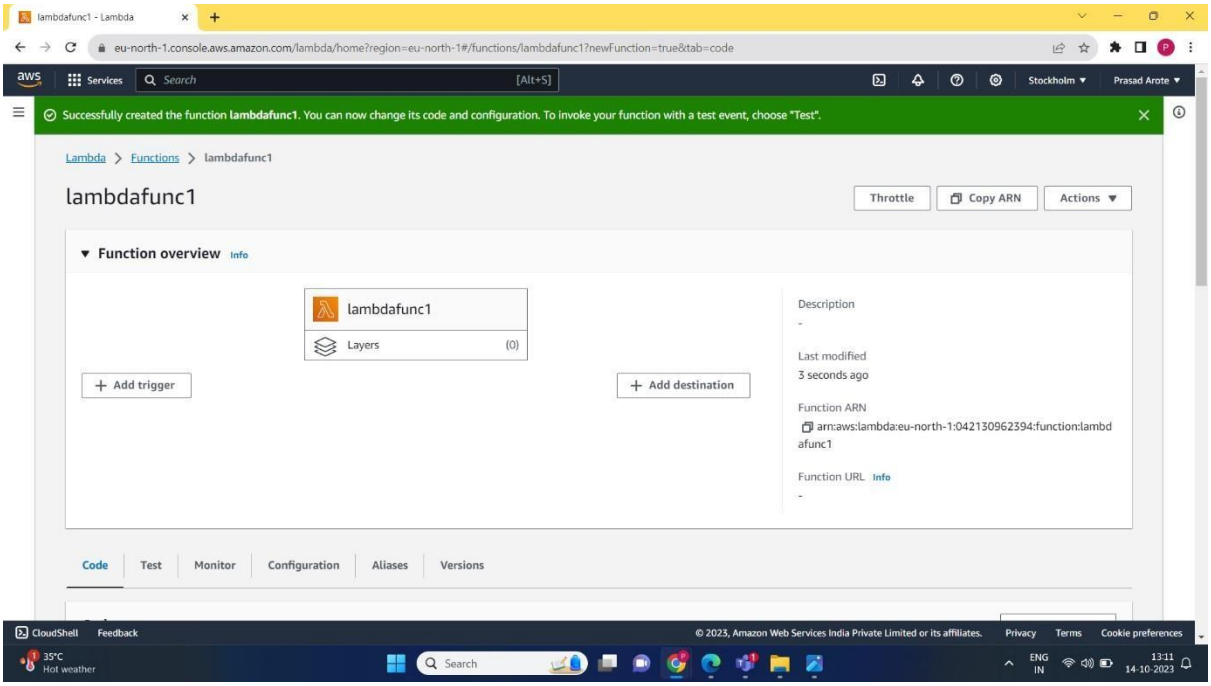


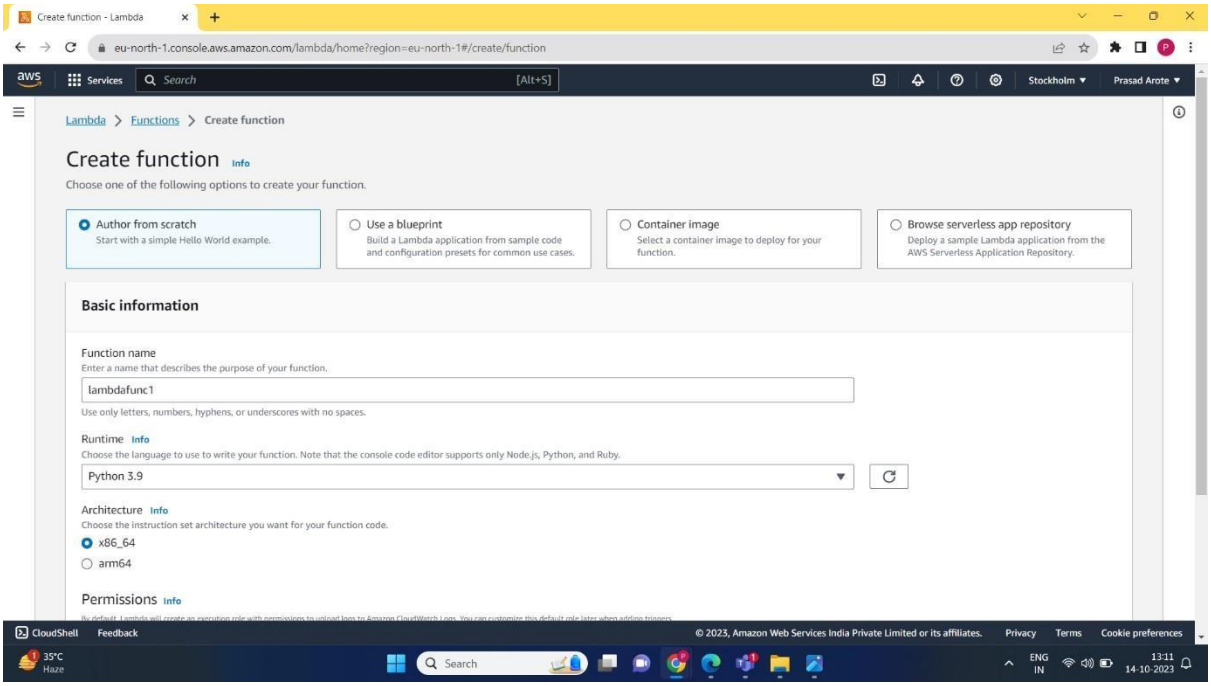




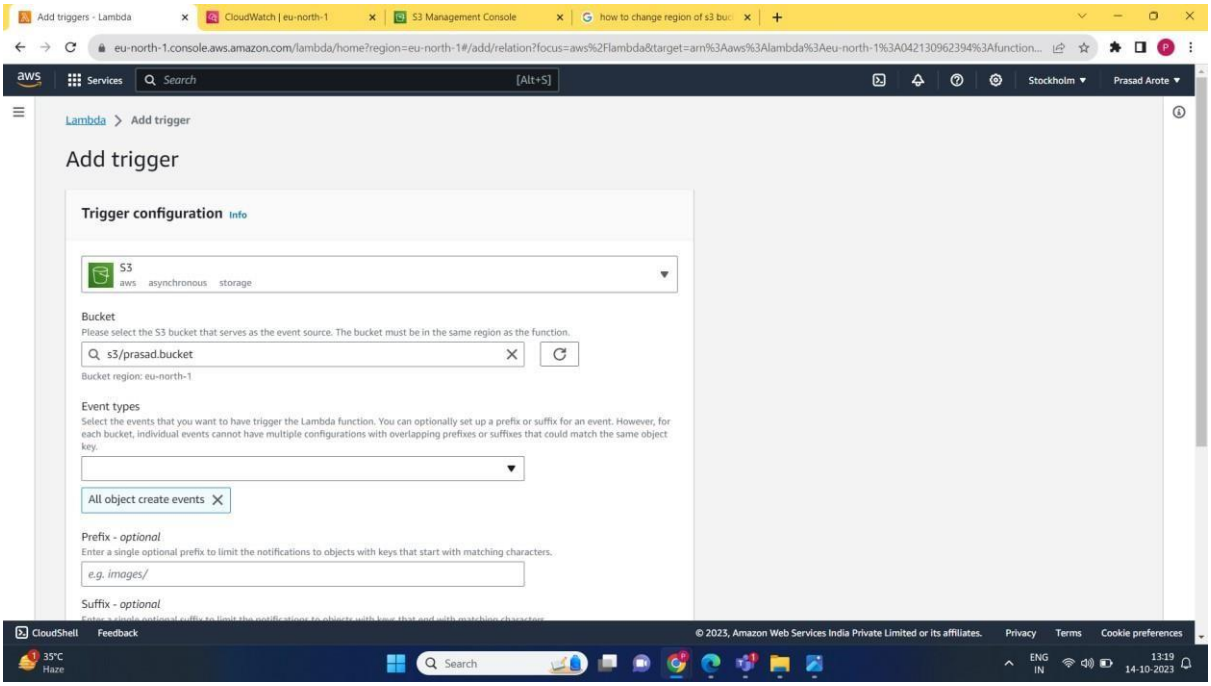
2. Create a Lambda function.

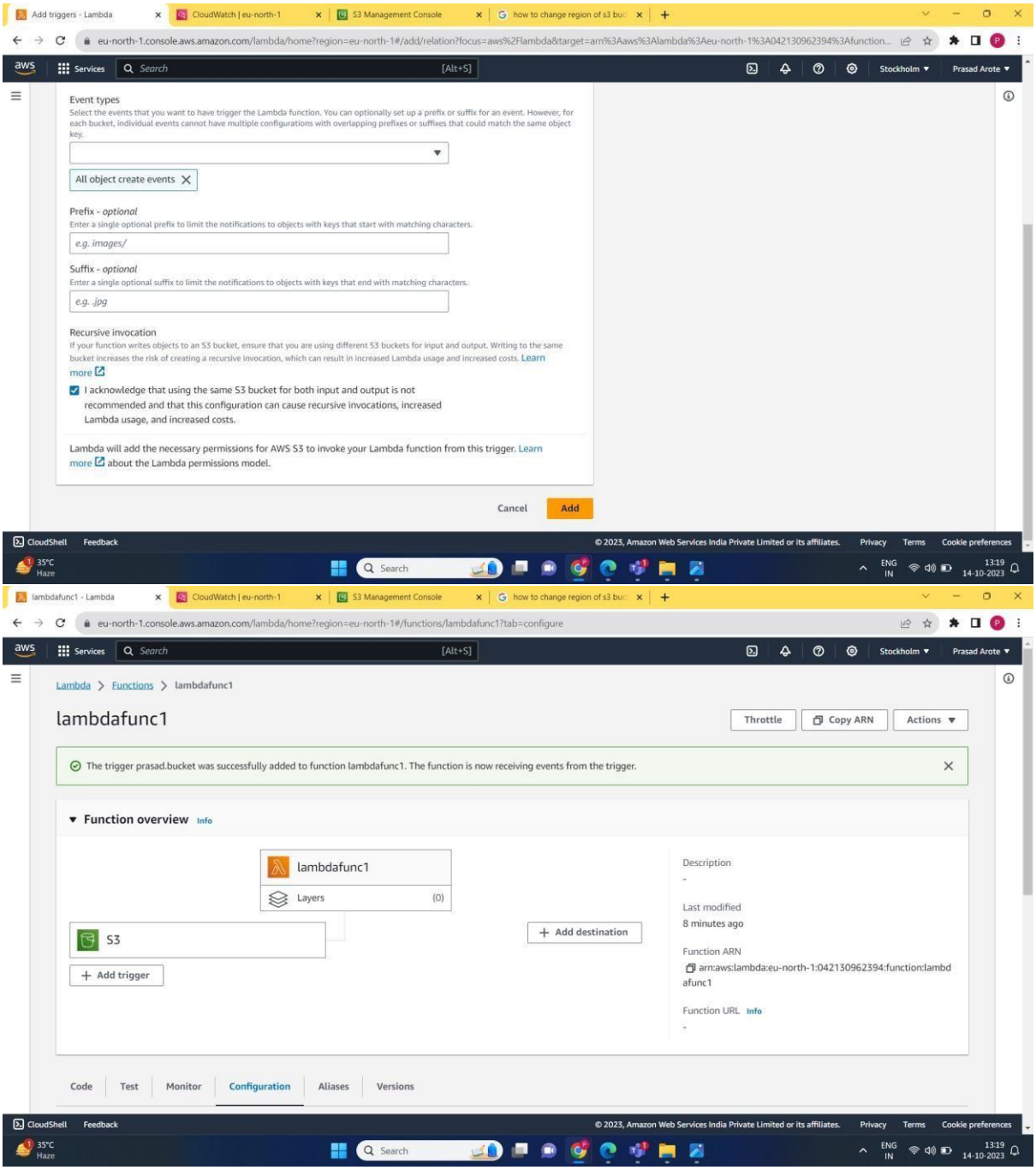

S3 Management Console
s3.console.aws.amazon.com/s3/buckets?region=ap-south-1
lambda
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services (7)
Features (3)
Resources New
Documentation (9,951)
Knowledge Articles (20)
Marketplace (664)
Blogs (1,017)
Events (13)
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
View Storage Lens dashboard
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

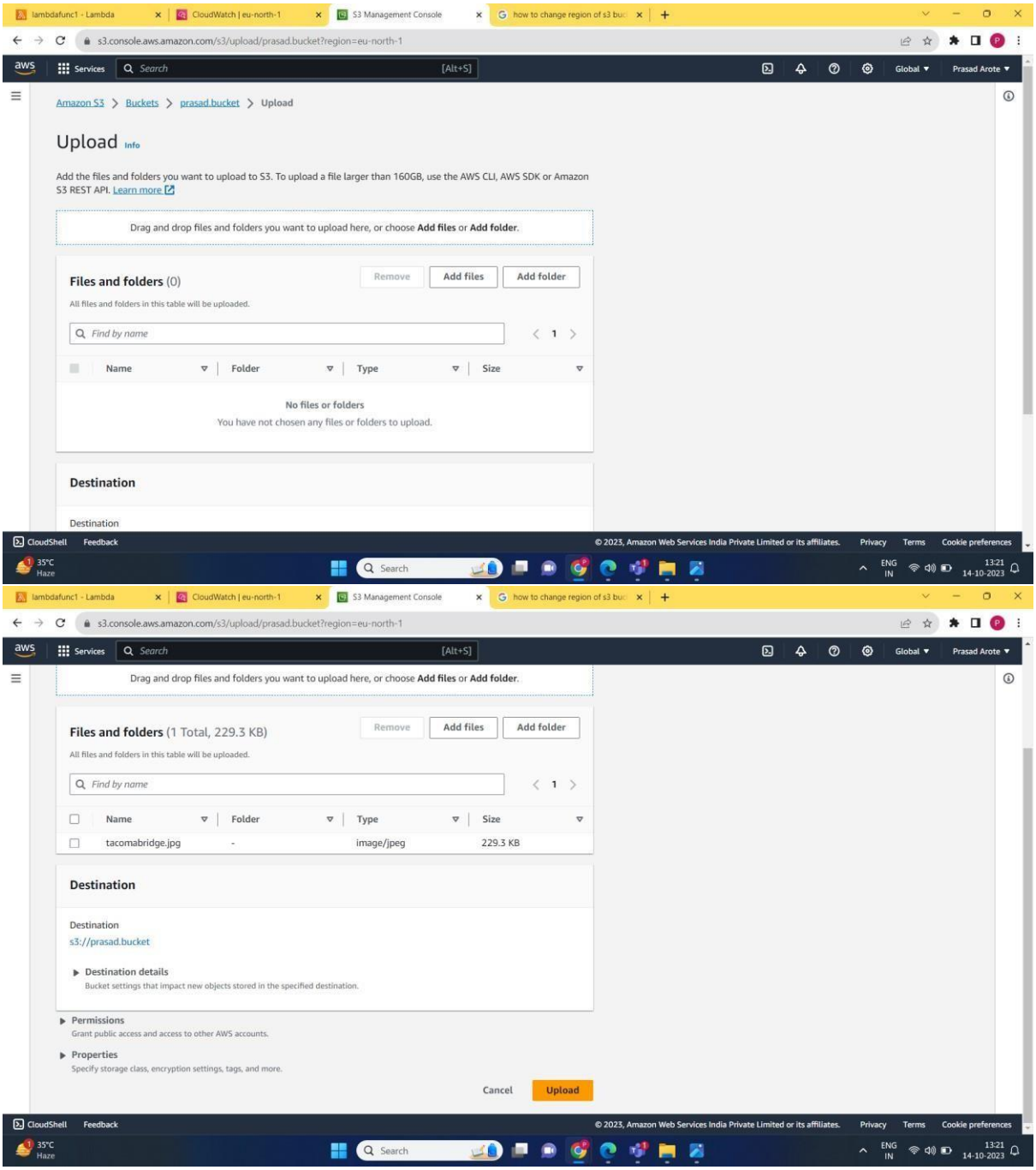

lambdafunc1 - Lambda
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/lambdafunc1?newFunction=true&tab=code
Successfully created the function lambdafunc1. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > 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

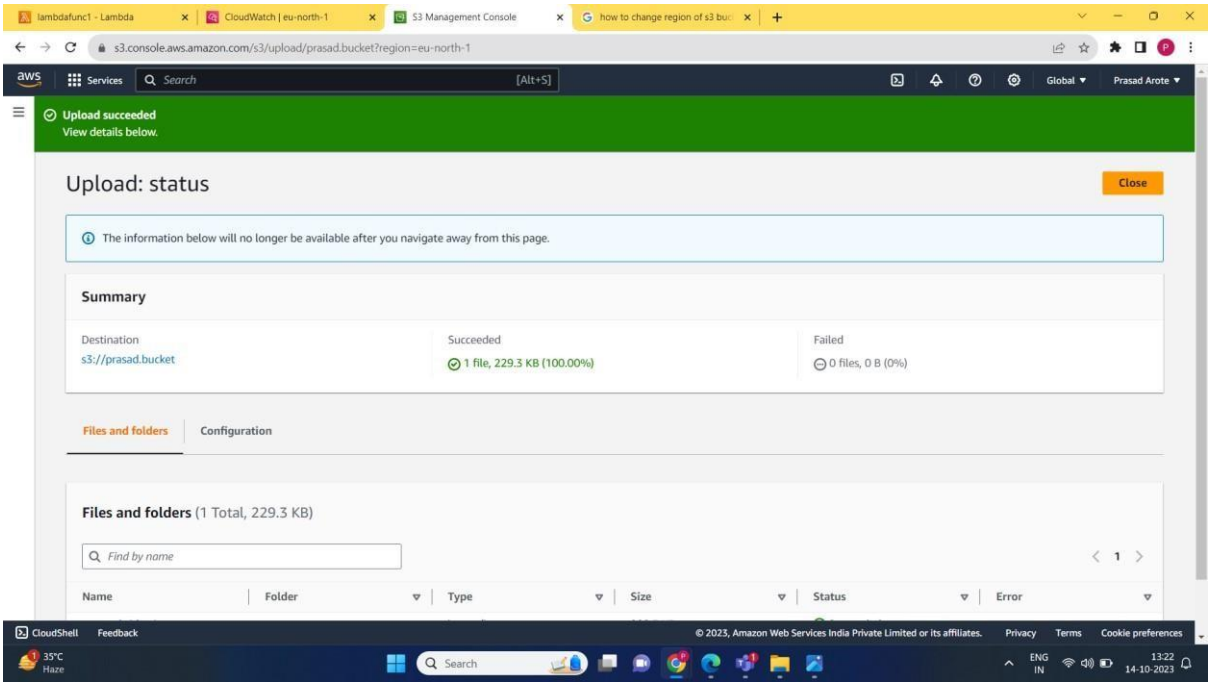


3. Create a trigger

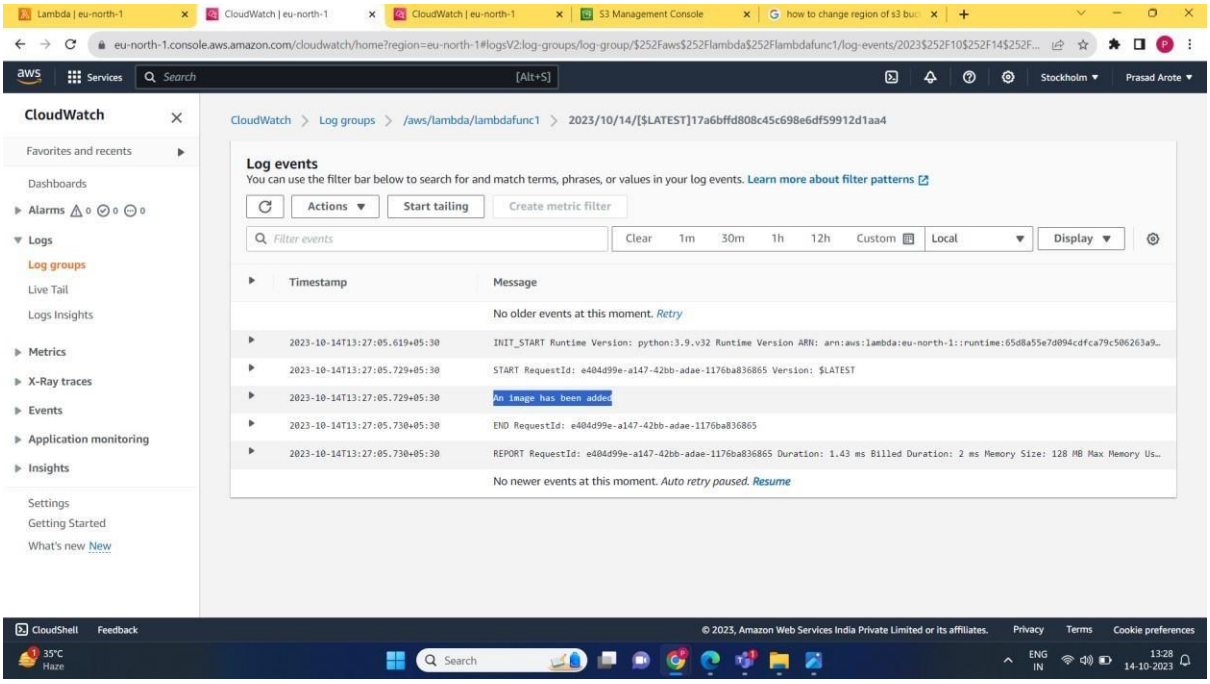



Add triggers - Lambda
CloudWatch | eu-north-1
S3 Management Console
how to change region of s3 buc
eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aeu-north-1%3A042130962394%3Afunction...
Services
Search
[Alt+S]
Stockholm
Prasad 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
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
ENG
IN
13:21
14-10-2023
Files and folders (1 Total, 229.3 KB)
tacomabridge.jpg
-
image/jpeg
229.3 KB
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



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.

Lab Assignment 8

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

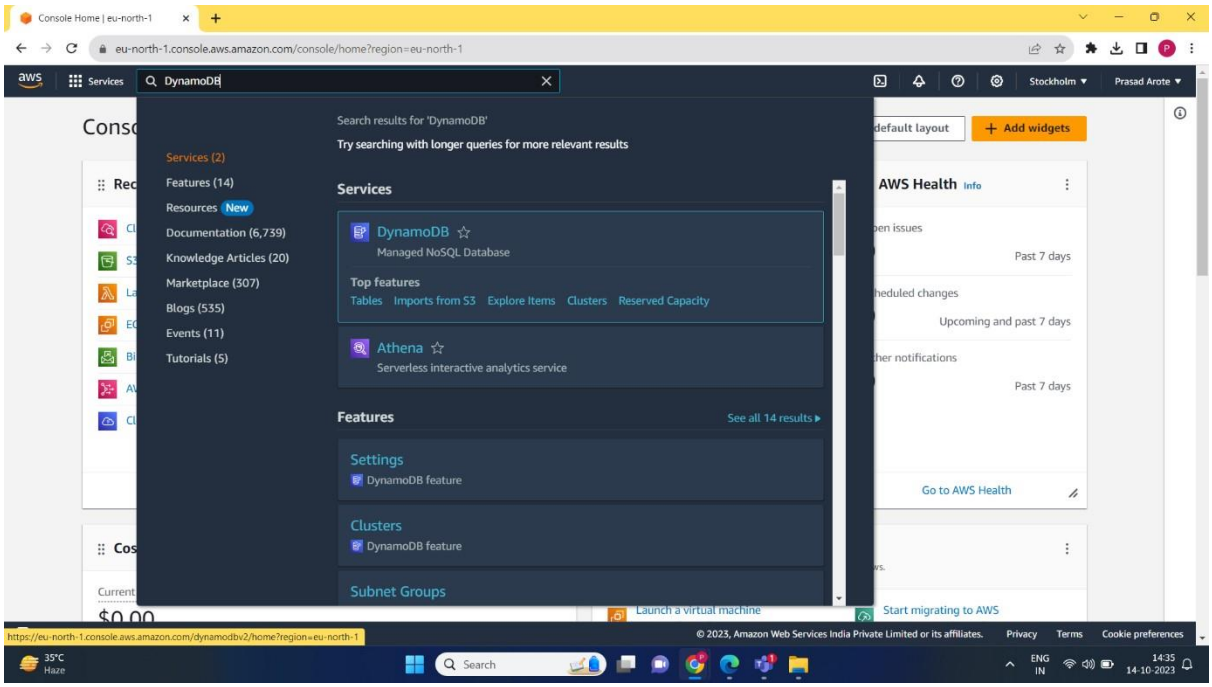
LO6: To engineer a composition of nano services using AWS Lambda and Step Functions with the serverless framework.

THEORY:

DYNAMO DB

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

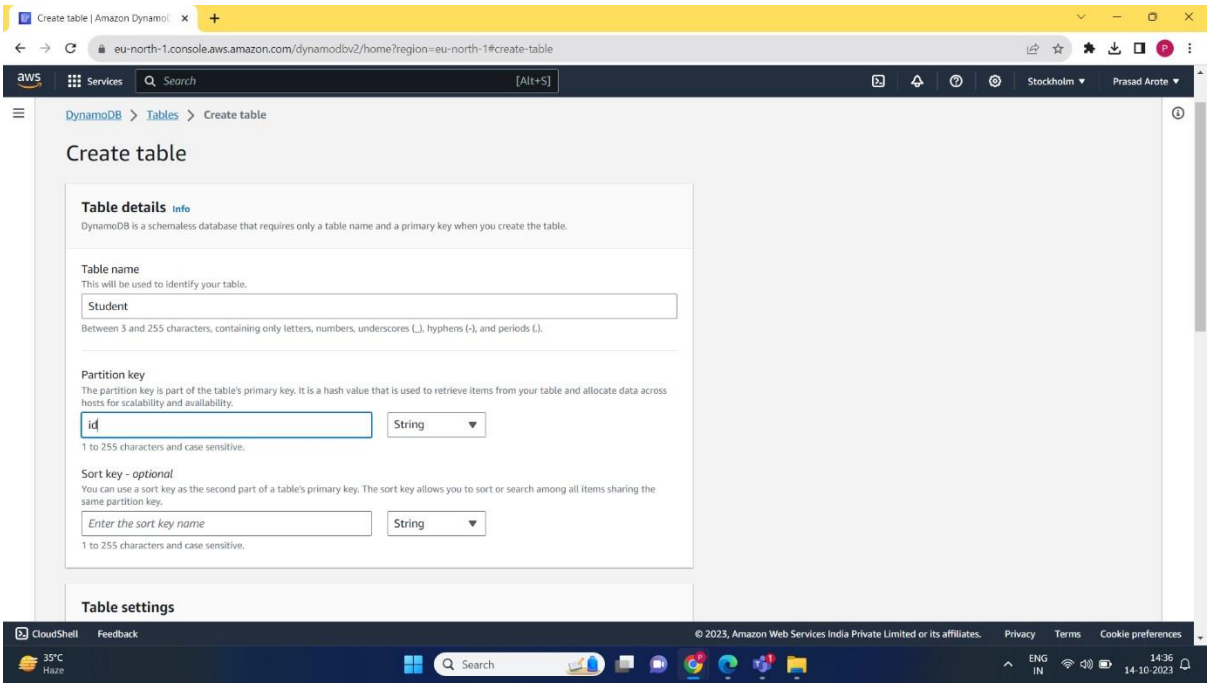
With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource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



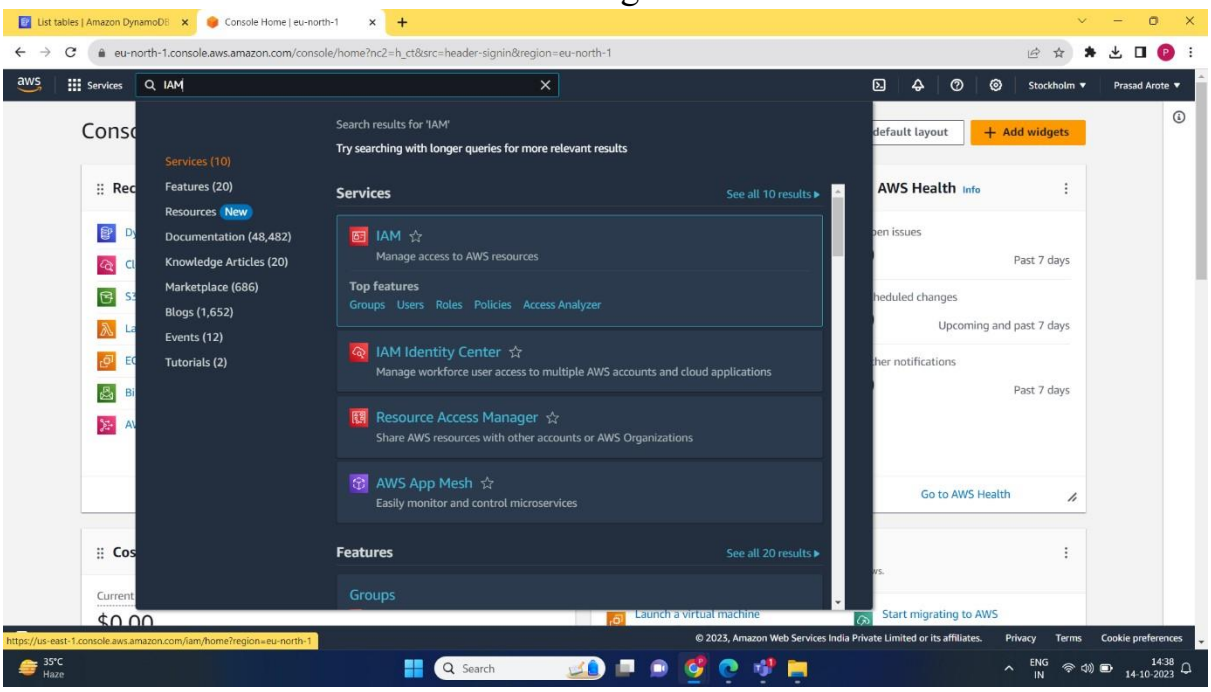
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

STEPS:

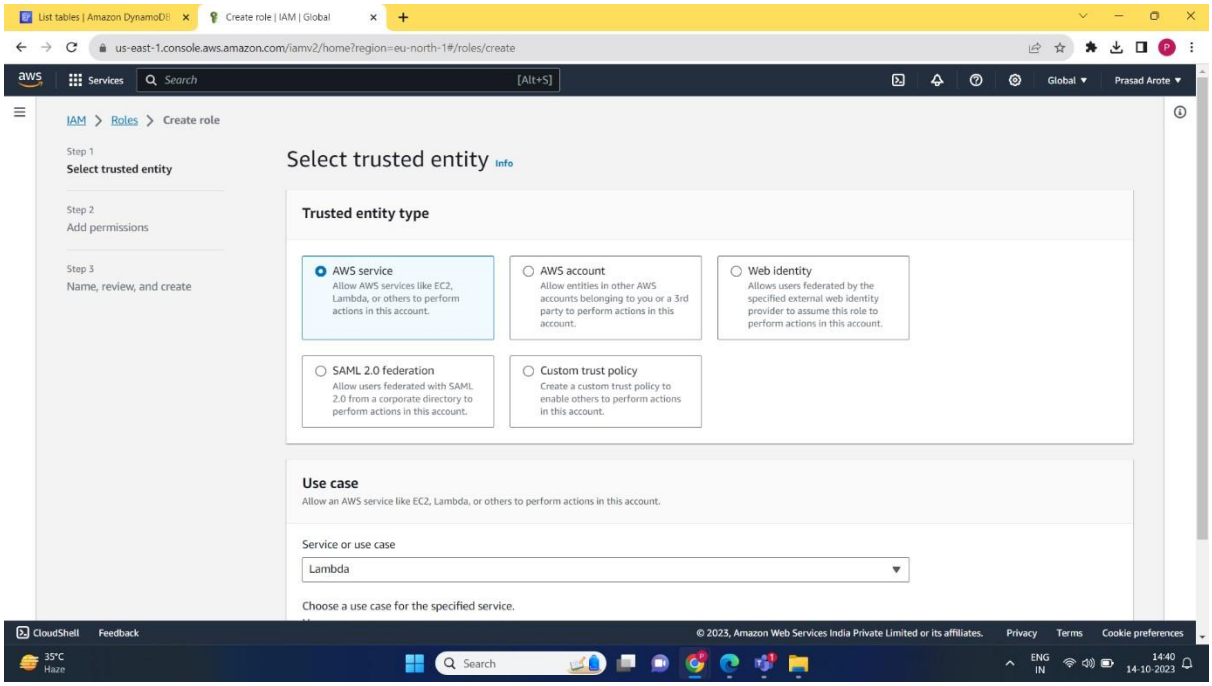
1. Create a table

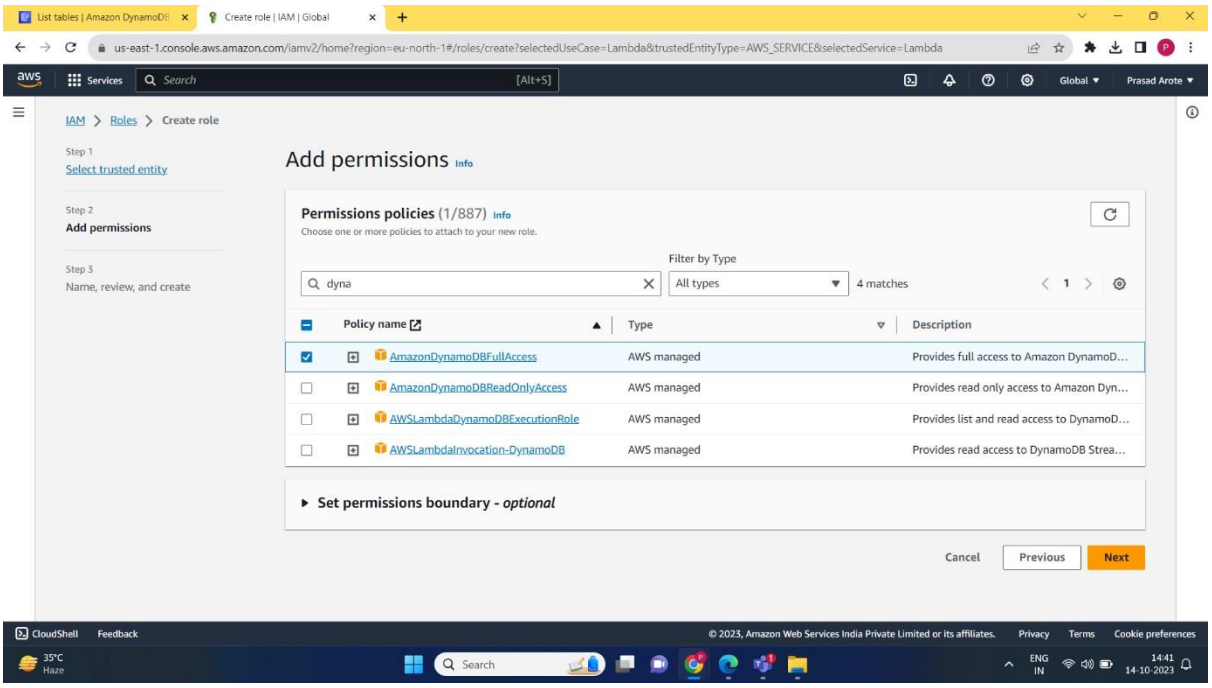


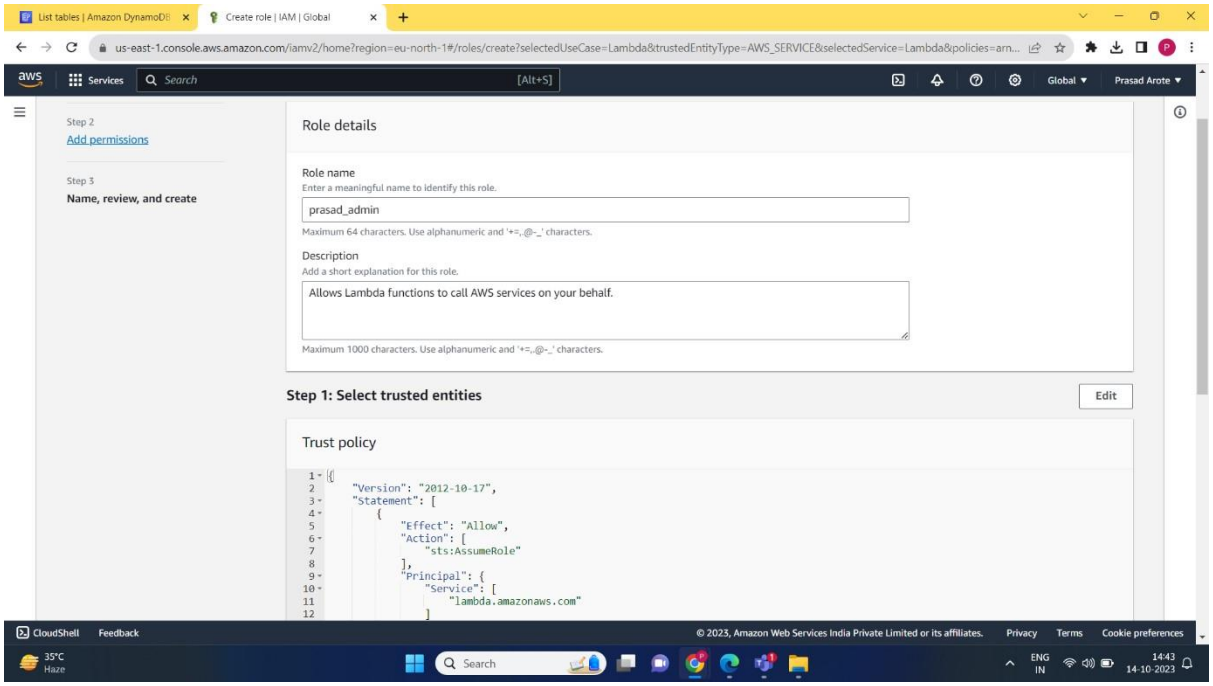
2. Create a role using IAM

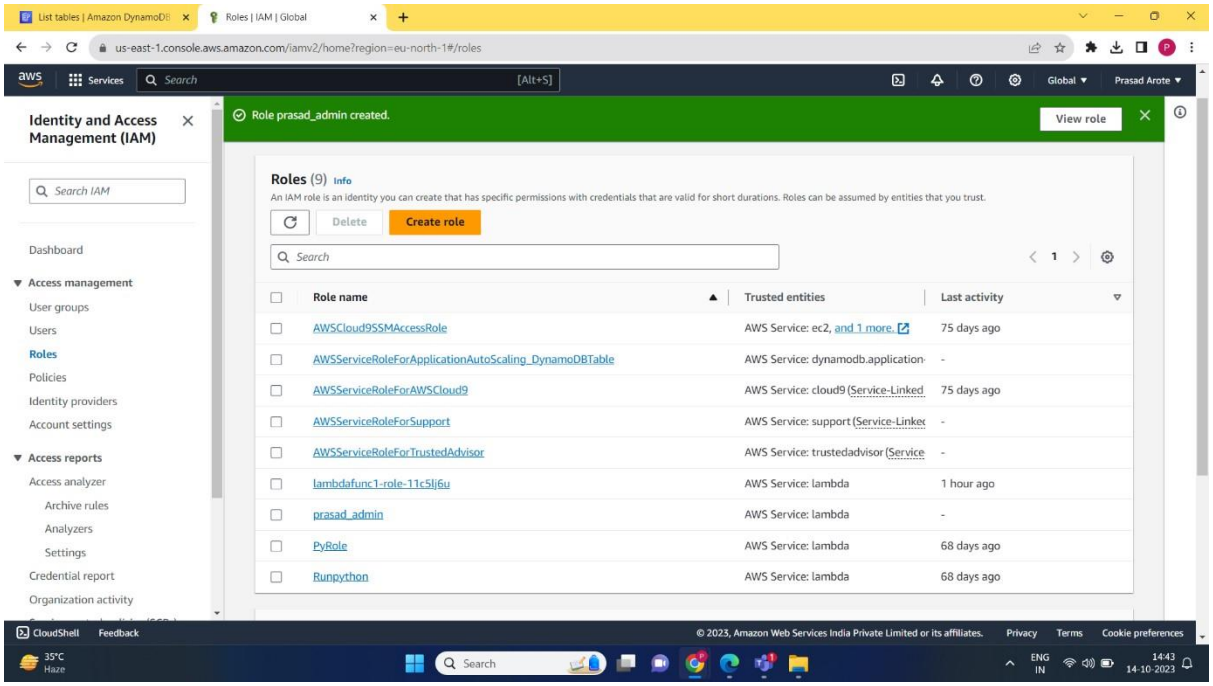


3. Add permissions – AmazonDynamoFullAccess

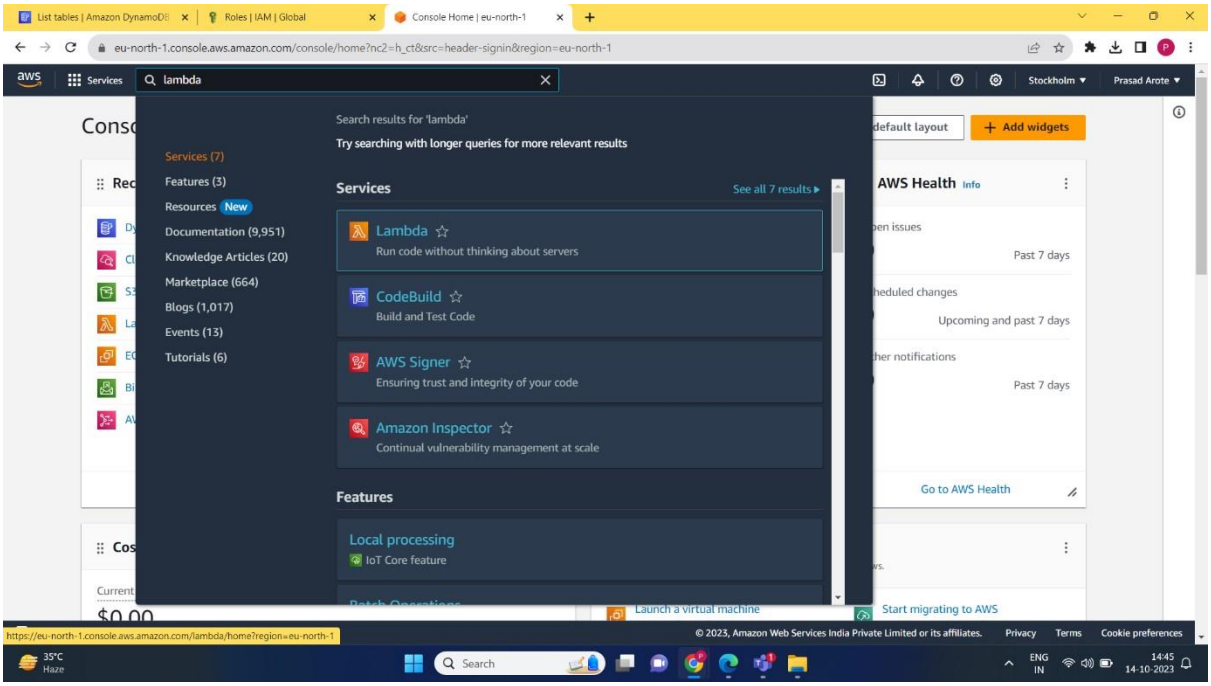

List tables | Amazon DynamoDB
Create role | IAM | Global
us-east-1.console.aws.amazon.com/iamv2/home?region=eu-north-1#/roles/create
Services
Search
[Alt+S]
Global
Prasad Arote
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
Lambda
Choose a use case for the specified service.
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
14:40
14-10-2023

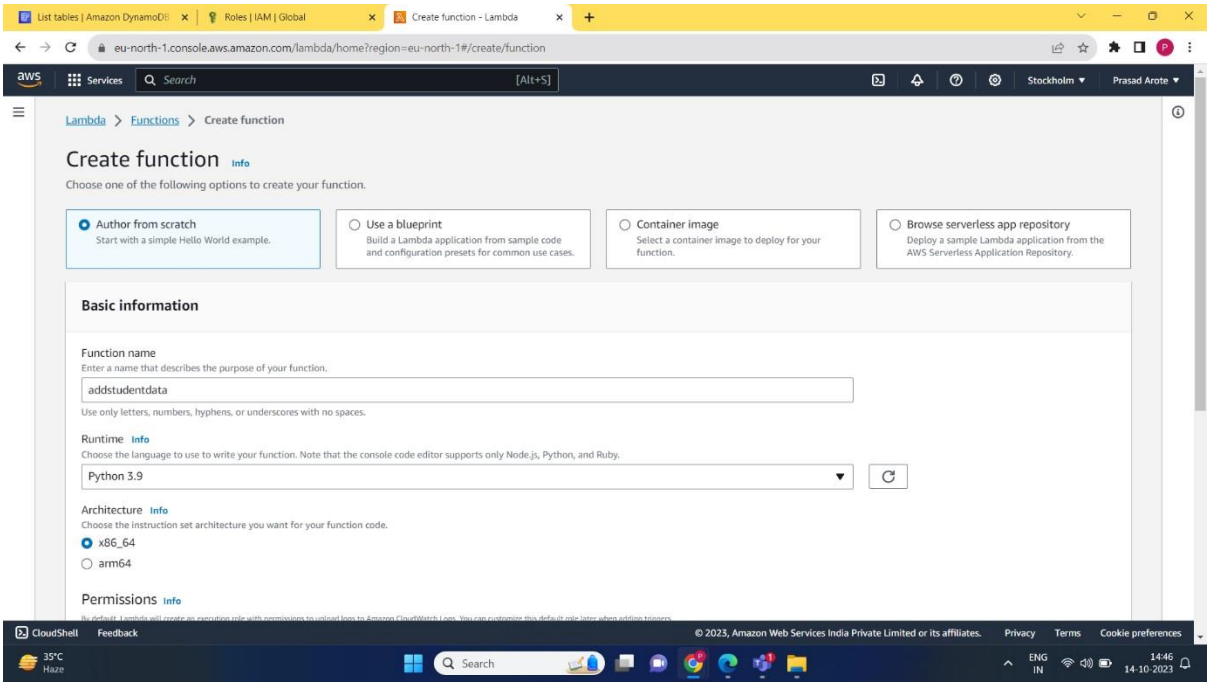

List tables | Amazon DynamoDB
Create role | IAM | Global
us-east-1.console.aws.amazon.com/iamv2/home?region=eu-north-1#/roles/create?selectedUseCase=Lambda&trustedEntityType=AWS_SERVICE&selectedService=Lambda
Services
Search
[Alt+S]
Global
Prasad Arote
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (1/887) Info
Choose one or more policies to attach to your new role.
Filter by Type
dyna
All types
4 matches
1
Policy name
Type
Description
AmazonDynamoDBFullAccess
AWS managed
Provides full access to Amazon DynamoD...
AmazonDynamoDBReadOnlyAccess
AWS managed
Provides read only access to Amazon Dyn...
AWSLambdaDynamoDBExecutionRole
AWS managed
Provides list and read access to DynamoD...
AWSLambdaInvocation-DynamoDB
AWS managed
Provides read access to DynamoDB Strea...
Set permissions boundary - optional
Cancel
Previous
Next
CloudShell
Feedback
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
35°C
Haze
Search
ENG
IN
14:41
14-10-2023

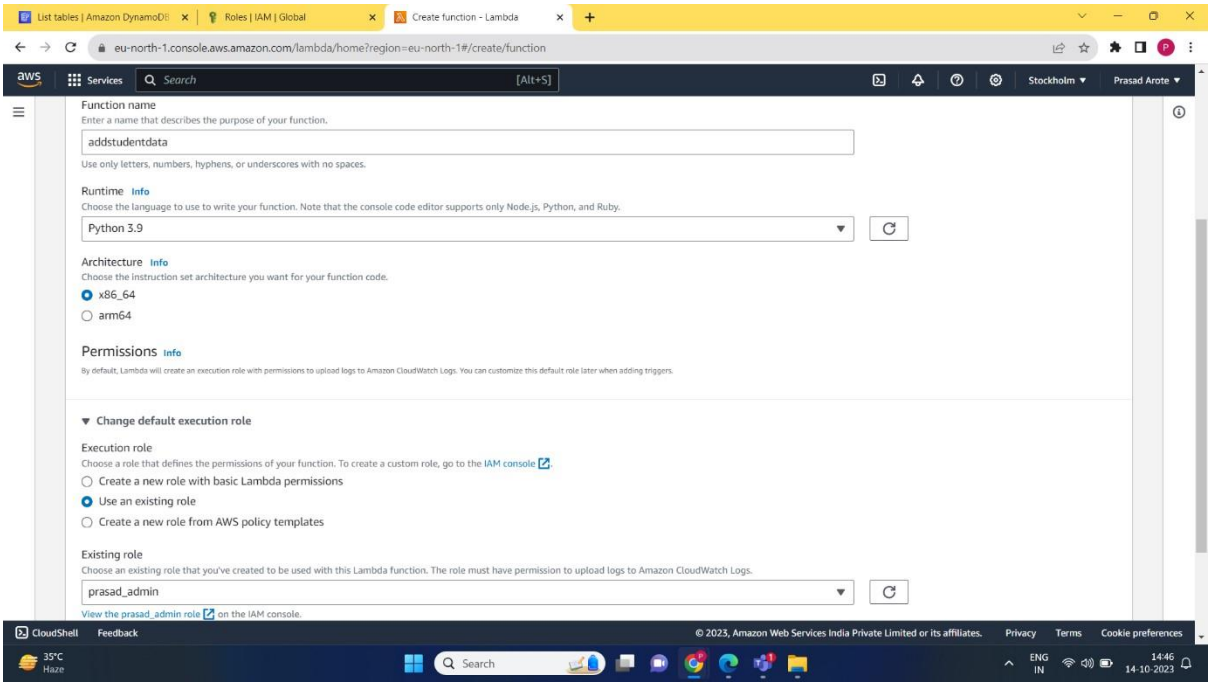


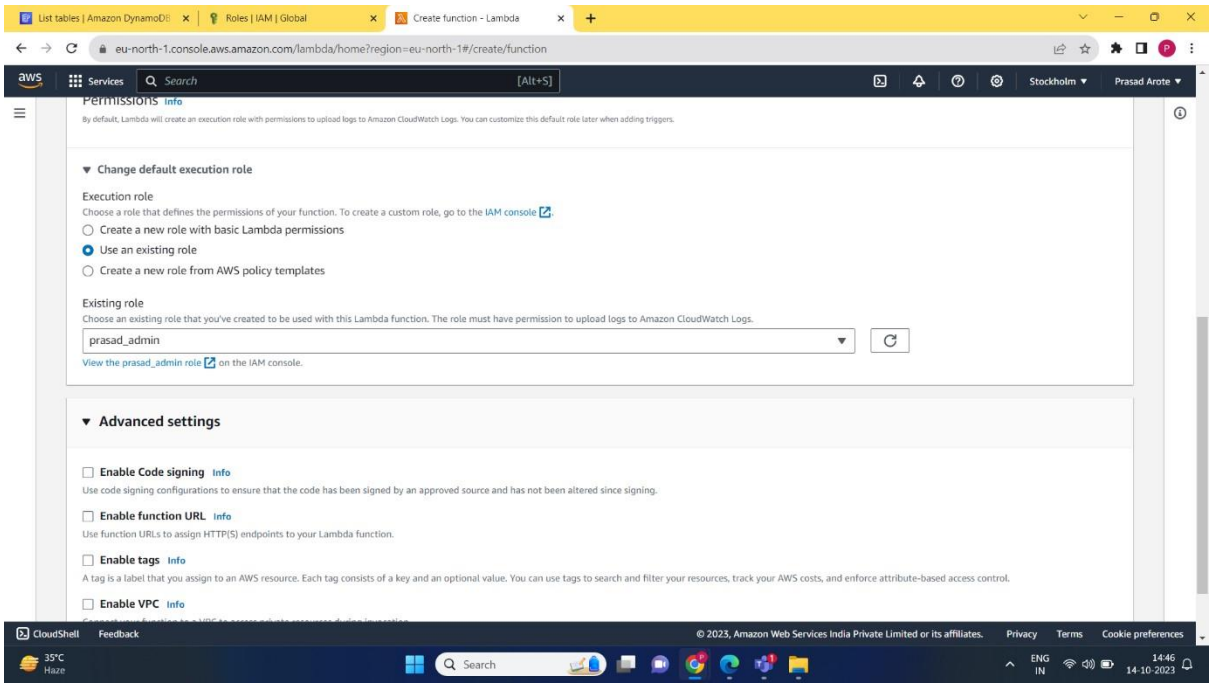


4. Create a Lambda Function

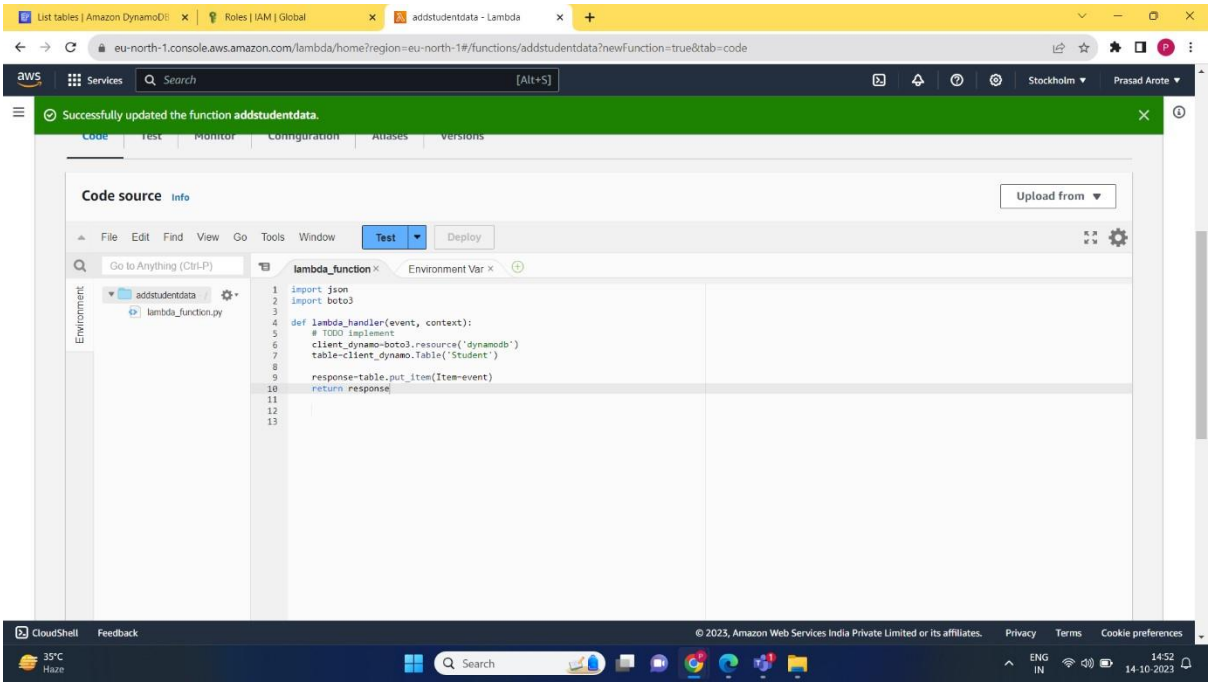

List tables | Amazon DynamoDB
Roles | IAM | Global
Console Home | eu-north-1
eu-north-1.console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin&region=eu-north-1
Services
lambda
Stockholm
Prasad Arote
Search results for 'lambda'
Try searching with longer queries for more relevant results
Services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

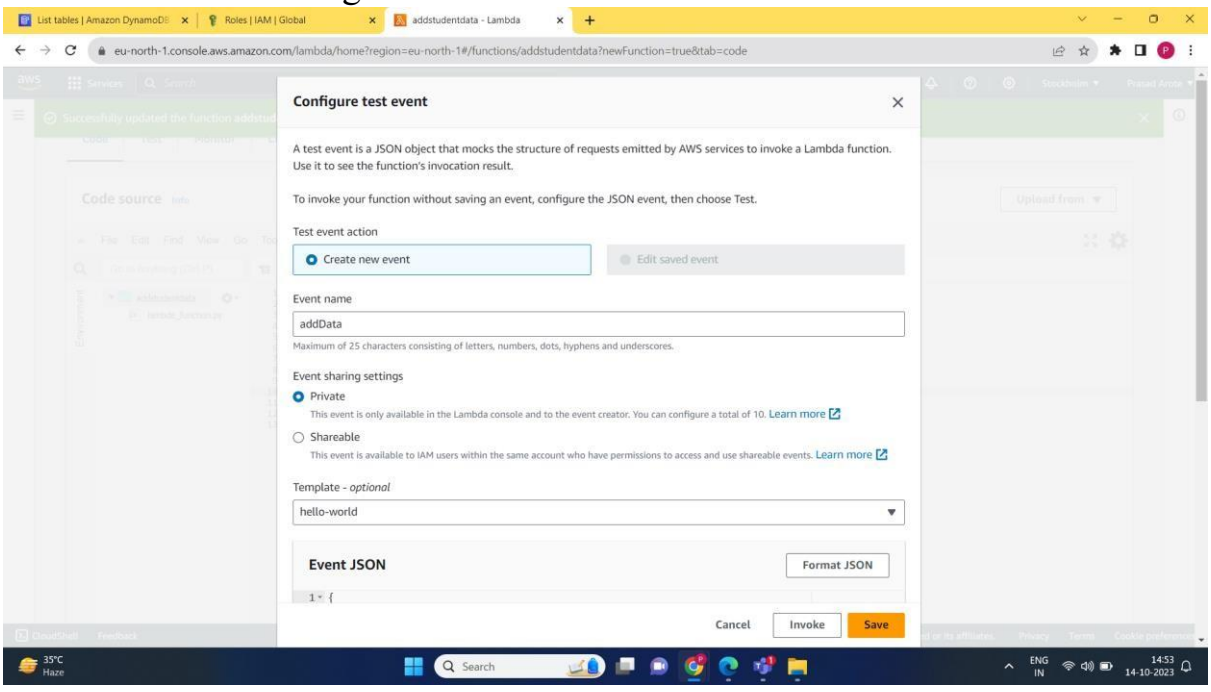


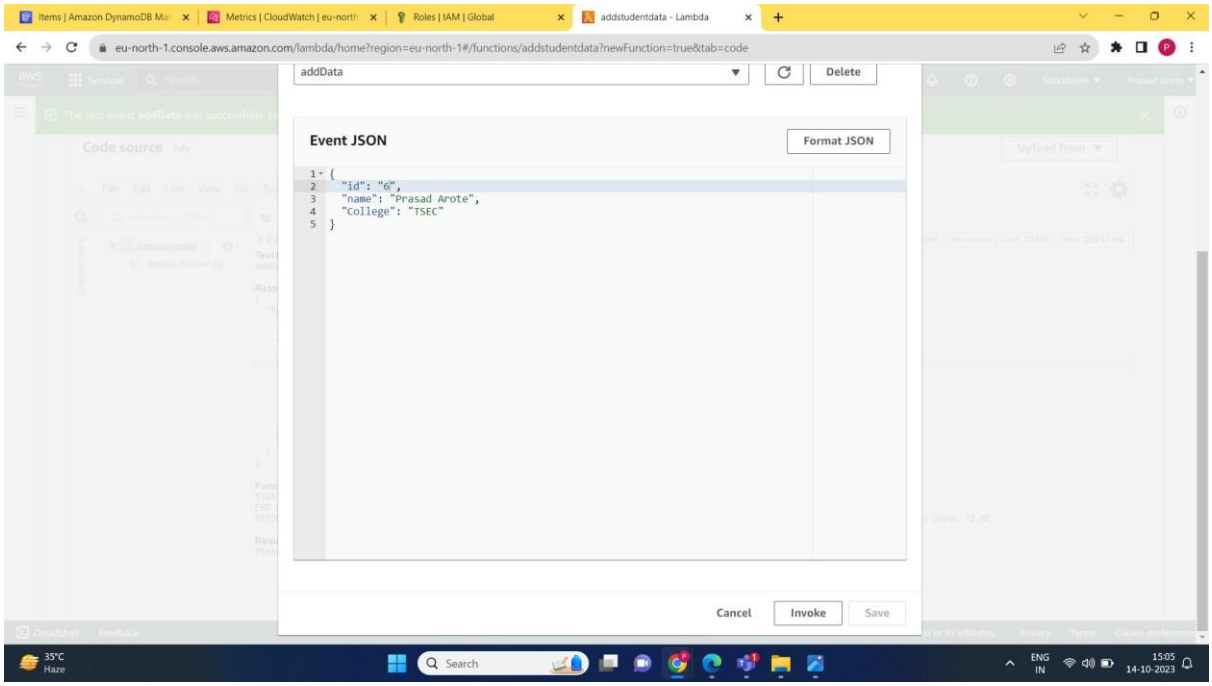


5. Write the following code

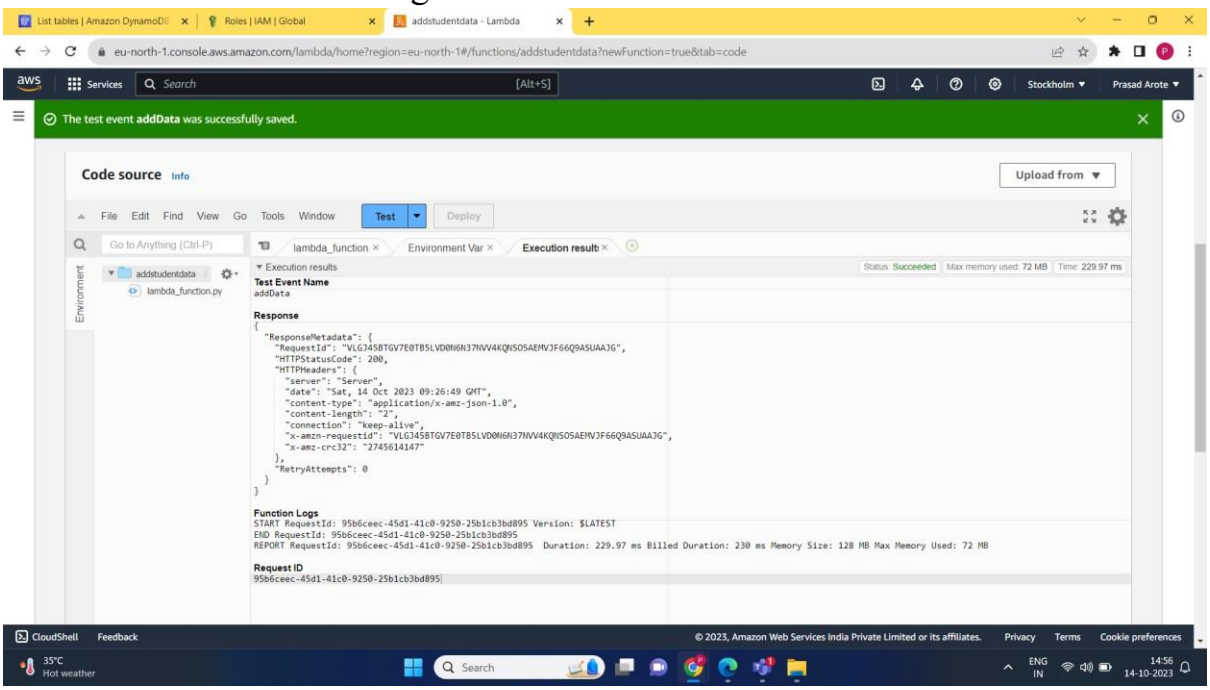


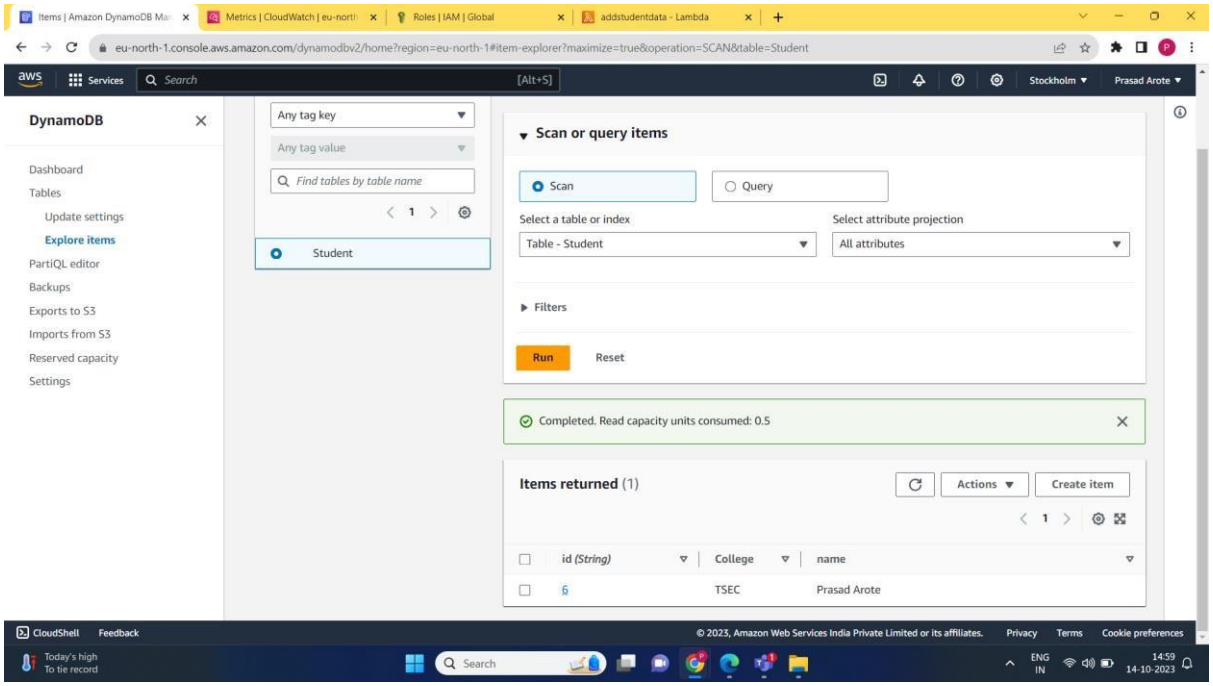
6. Configure test event and Save





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.

Assignment No 9

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

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

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

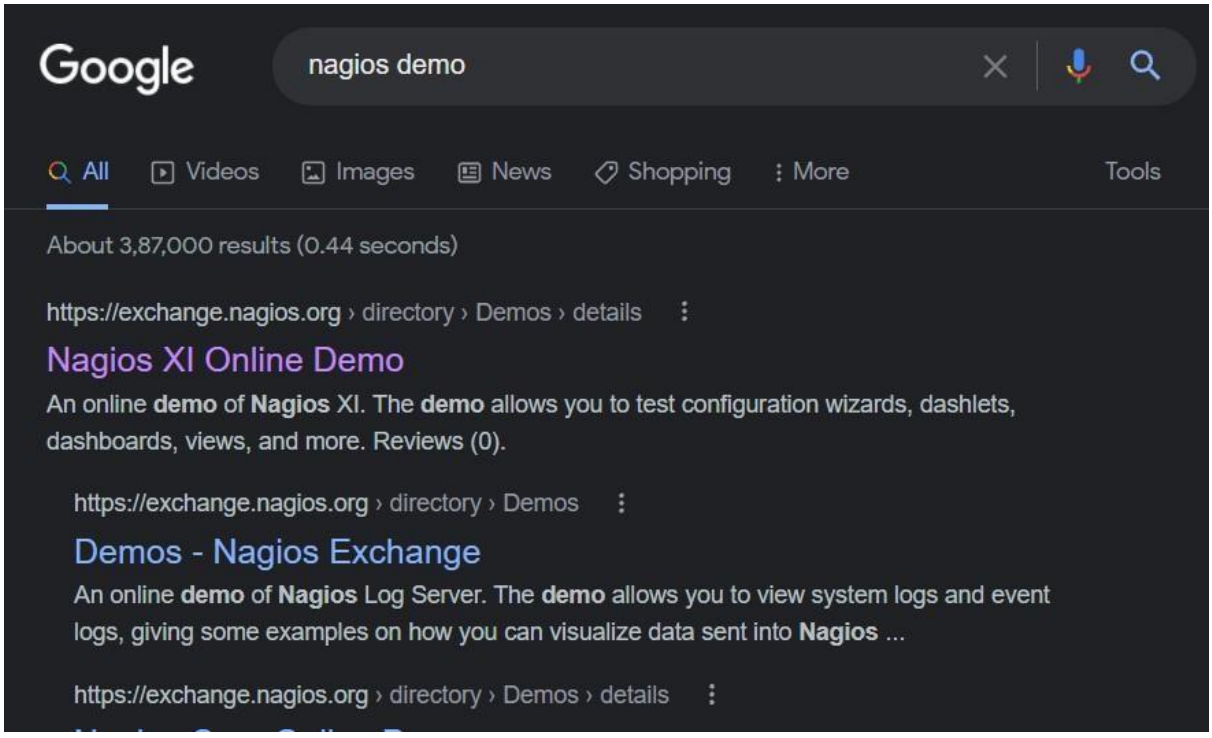
Theory:

What is Nagios and how it works?

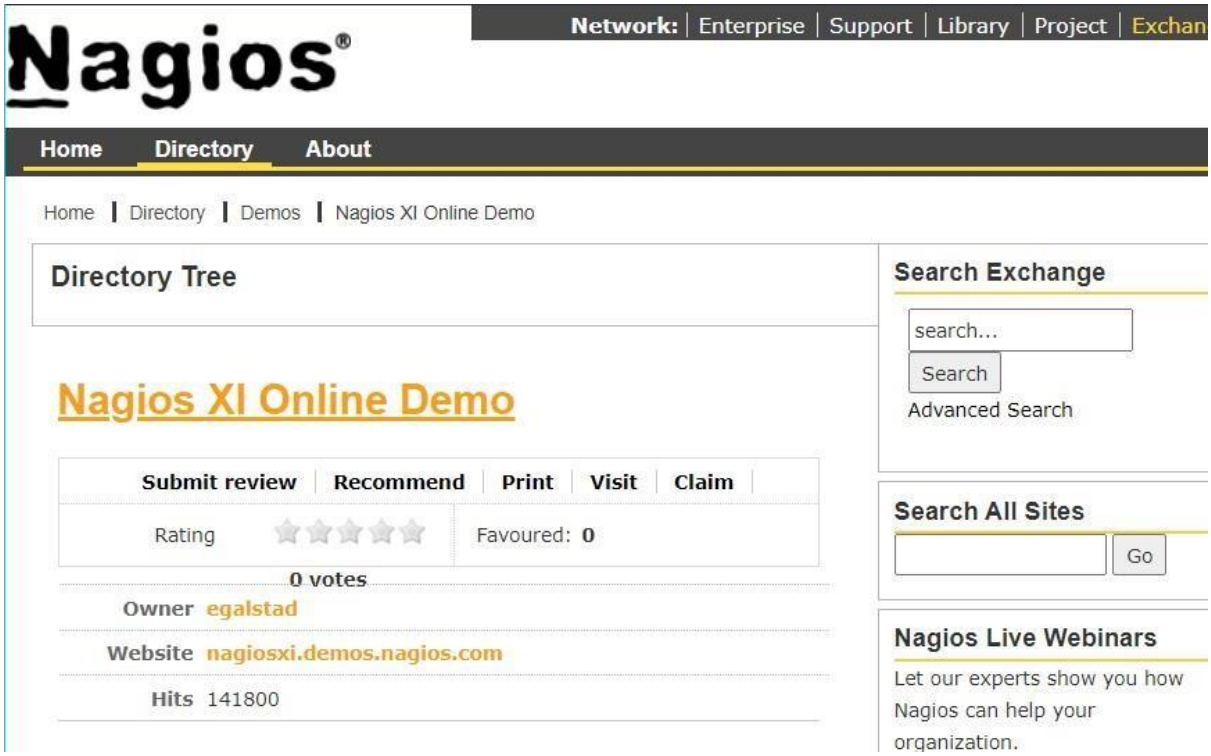
Nagios is an open source monitoring system for computer systems. ... Nagios software runs periodic checks on critical parameters of application, network and server resources. For example, Nagios can monitor memory usage, disk usage, microprocessor load, the number of currently running processes and log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

Steps-

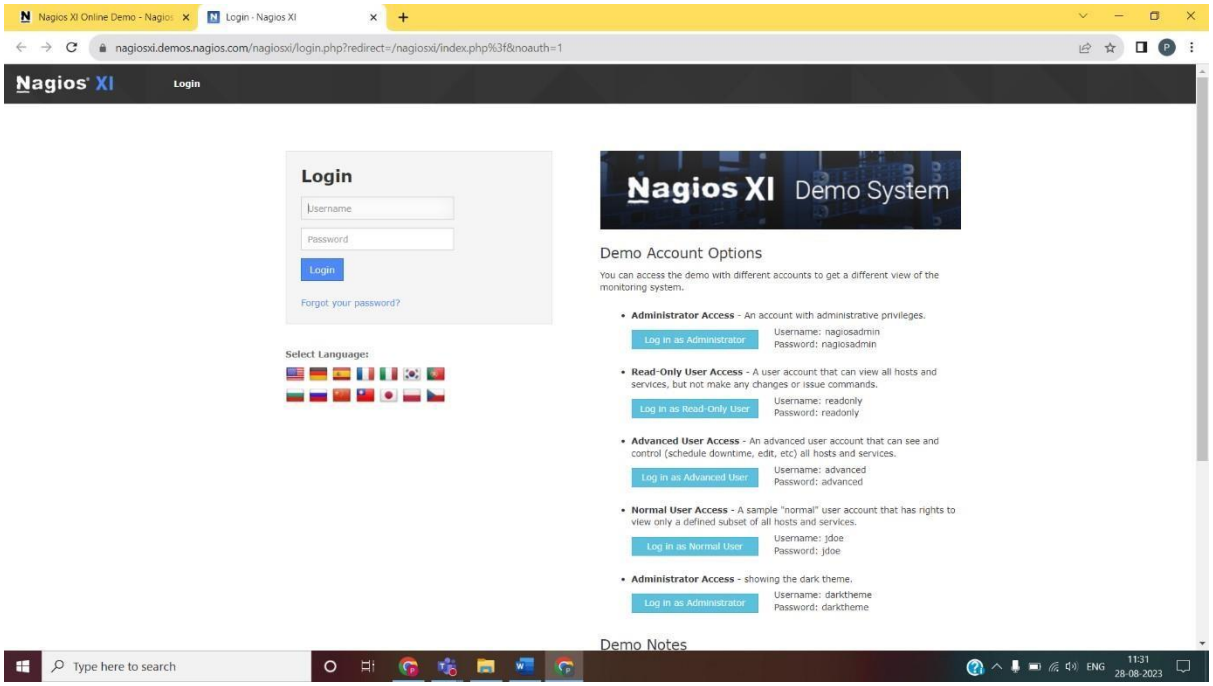
Go to google.com, Search Nagios Demo



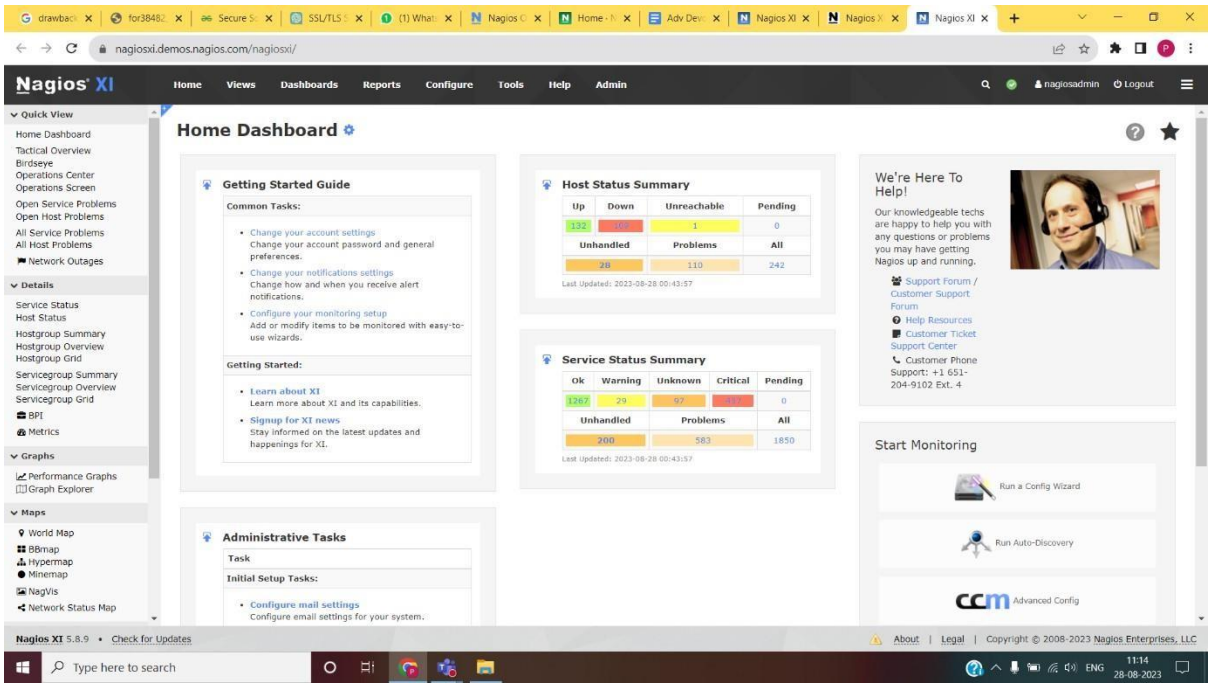
Now click on the website-



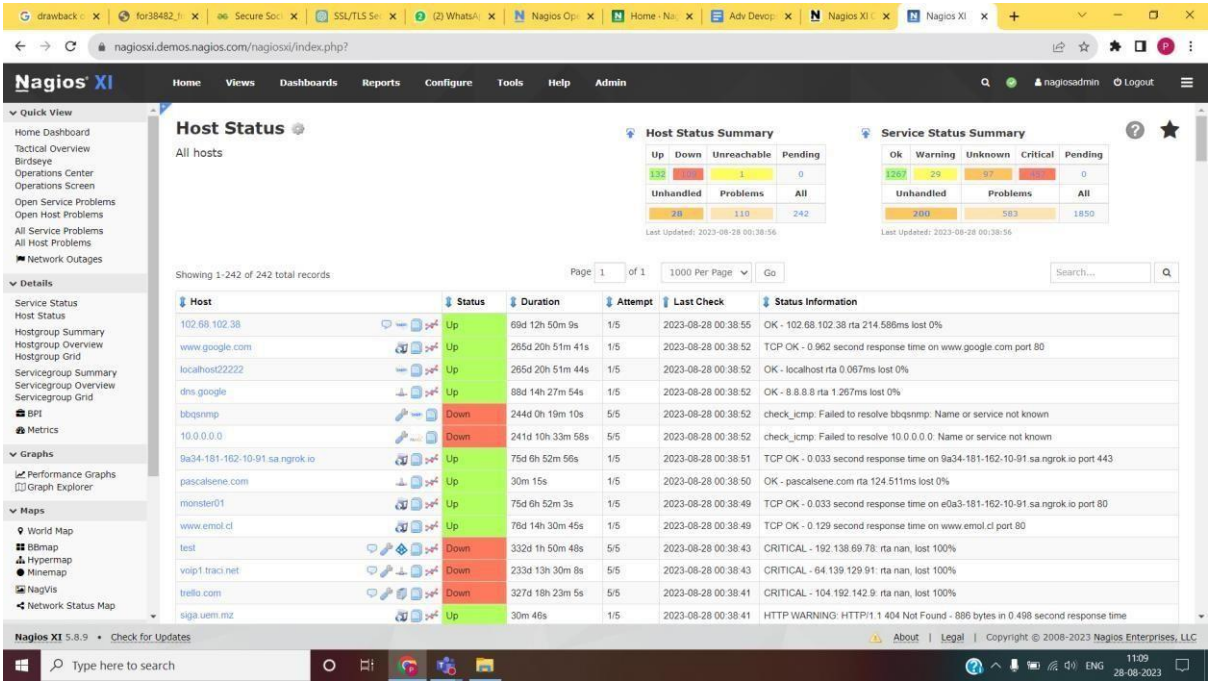
Now click on login as administrator



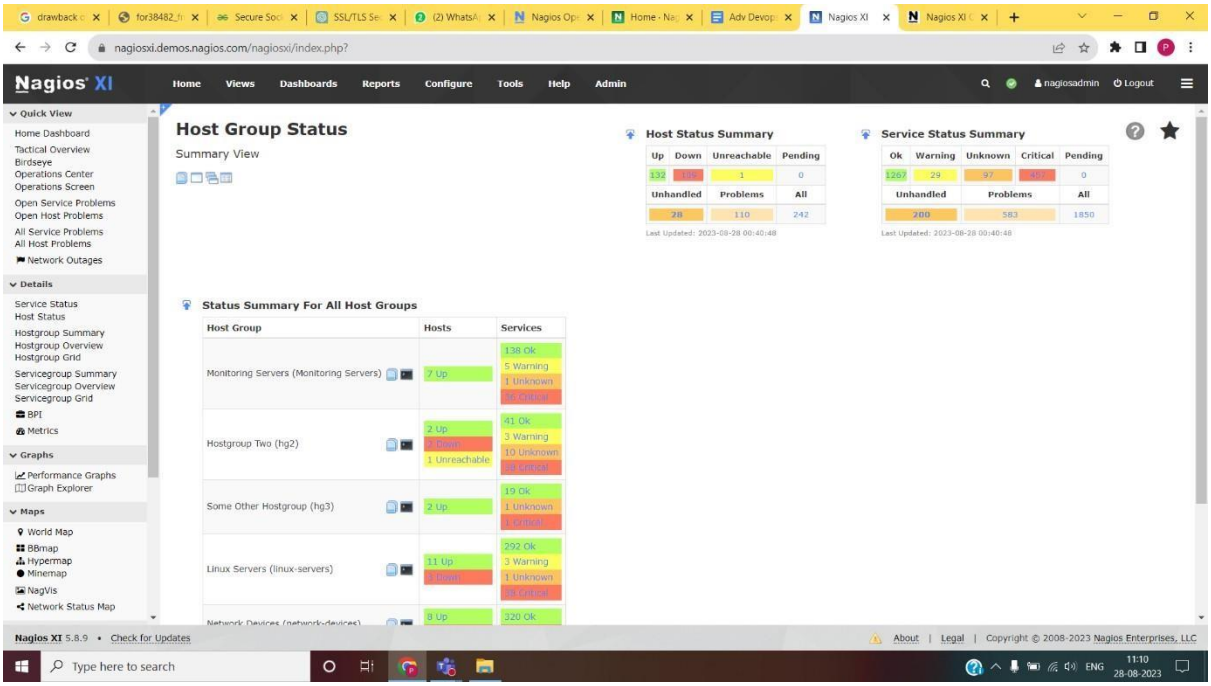
It will have interface like this



Now click on Host status-



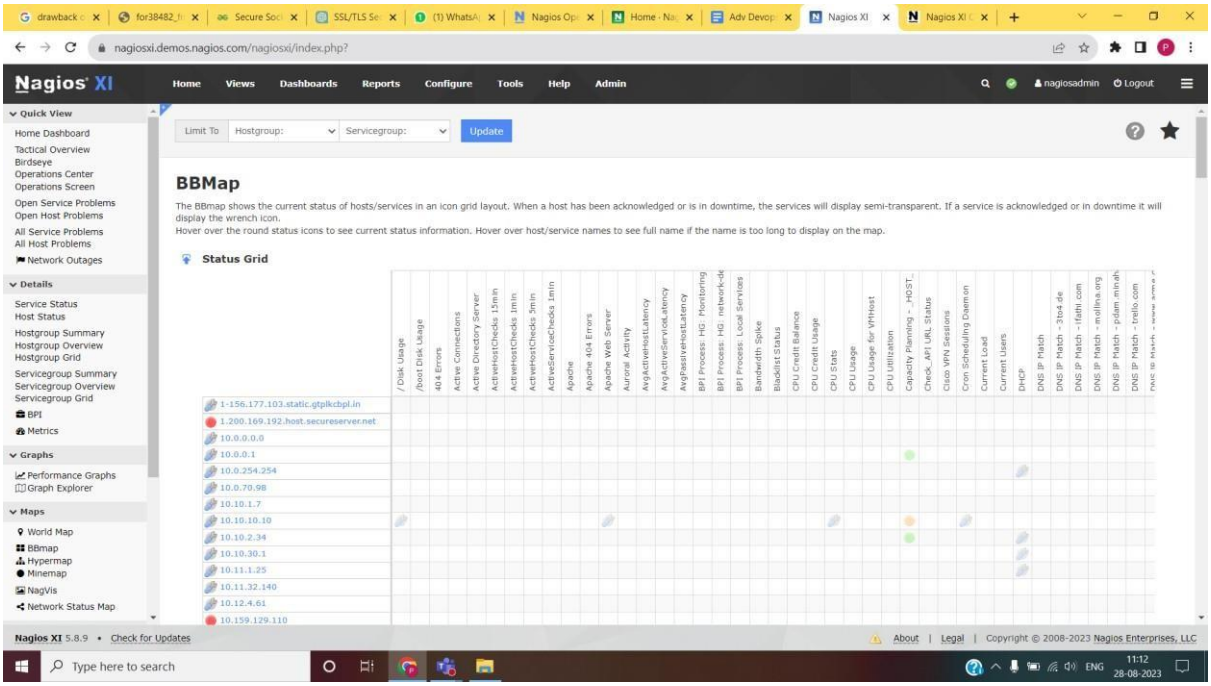
In the above image one can see **Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.**



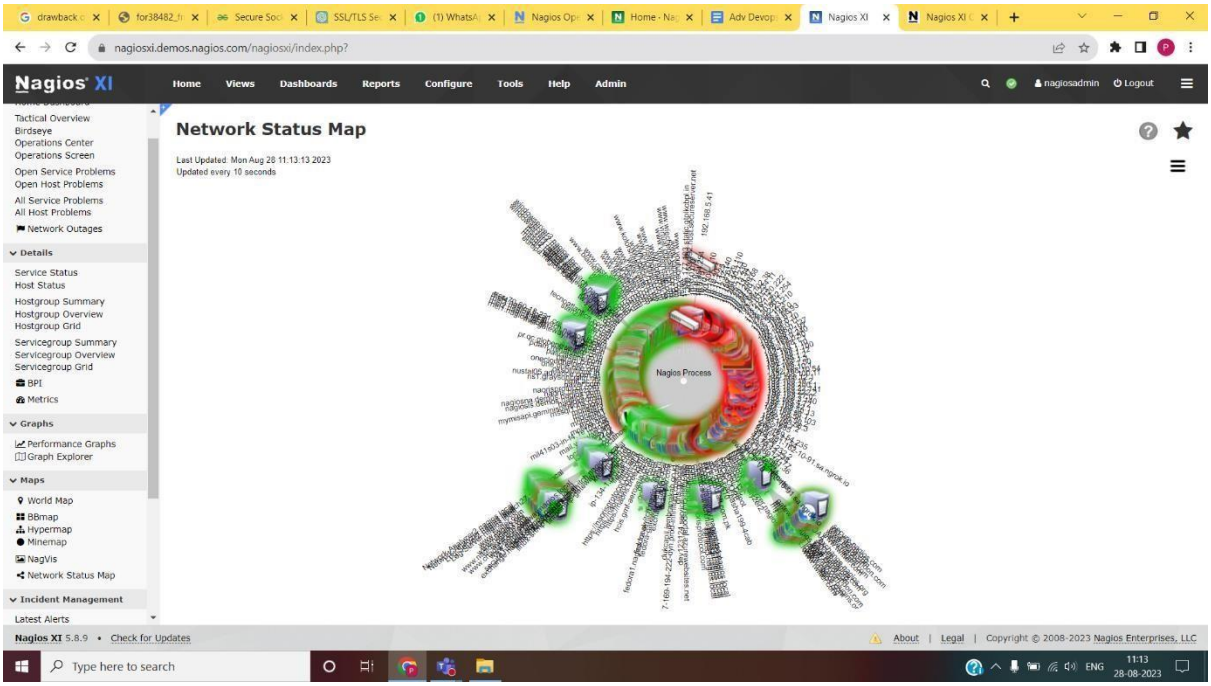
Here we can see Status Summary for All Host Groups

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.

T11 ALTAF ALAM 02

Lab Assignment 10

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

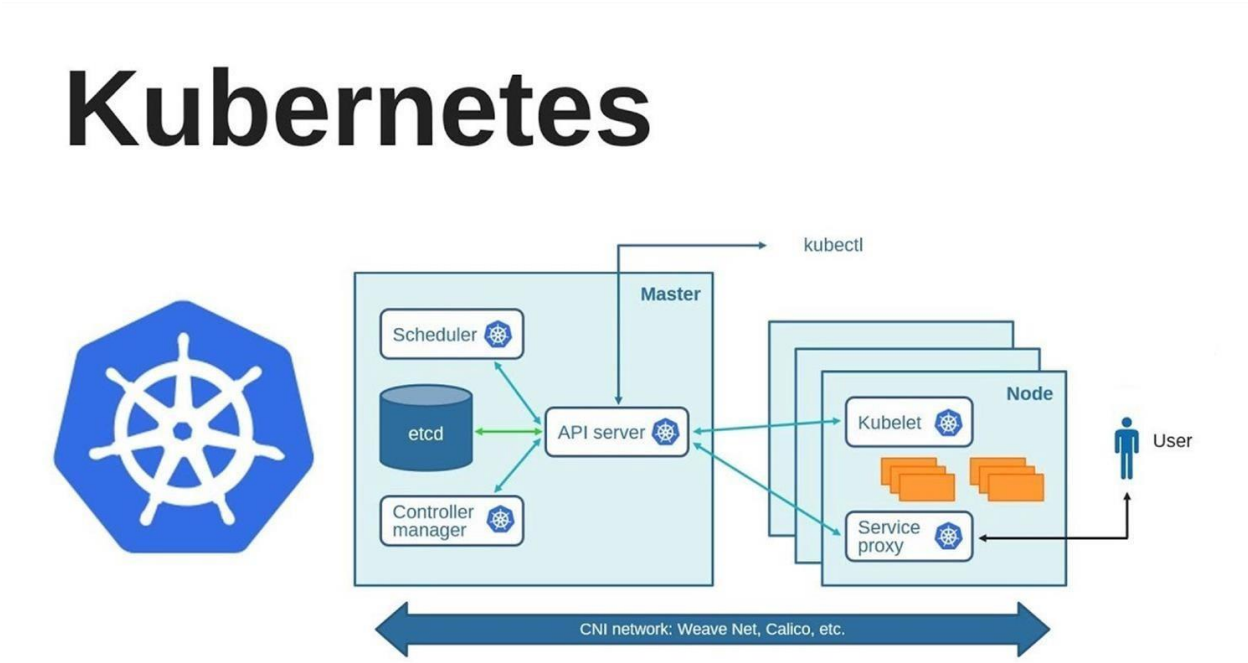
LO2. To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes.

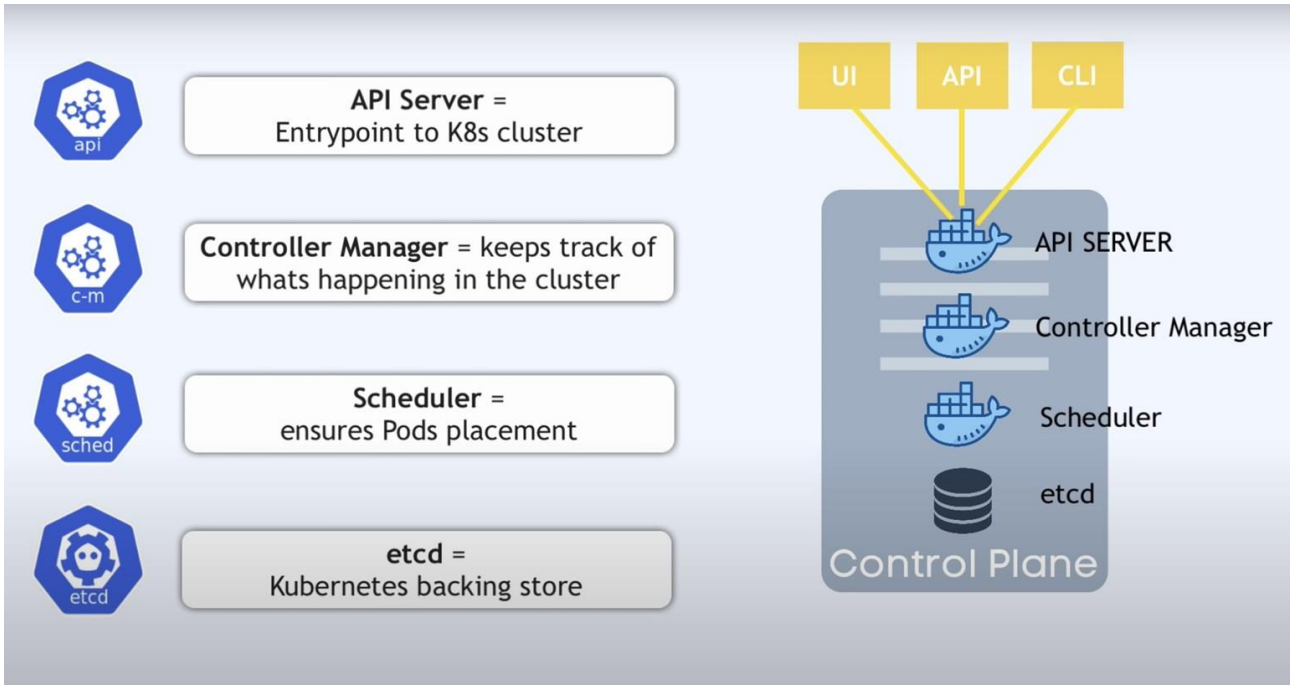
THEORY:

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

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

ARCHITECTURE





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 kubeapi-server)
- This kube-api-server is meant to scale automatically as per load.
- Kube-api-server is the front end of the control plane.

■ etcd

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

- Source of 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

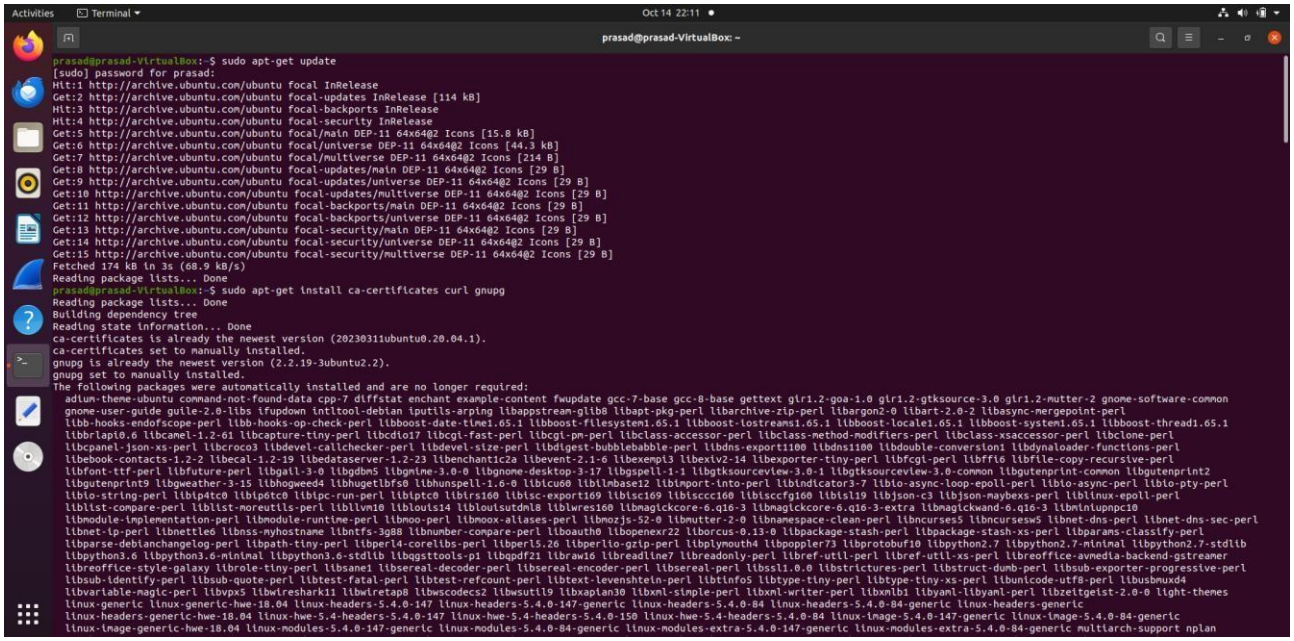
- Works with kubelet
- Pulling images
- Start/Stop Containers
- Exposing containers on ports specified in the manifest.

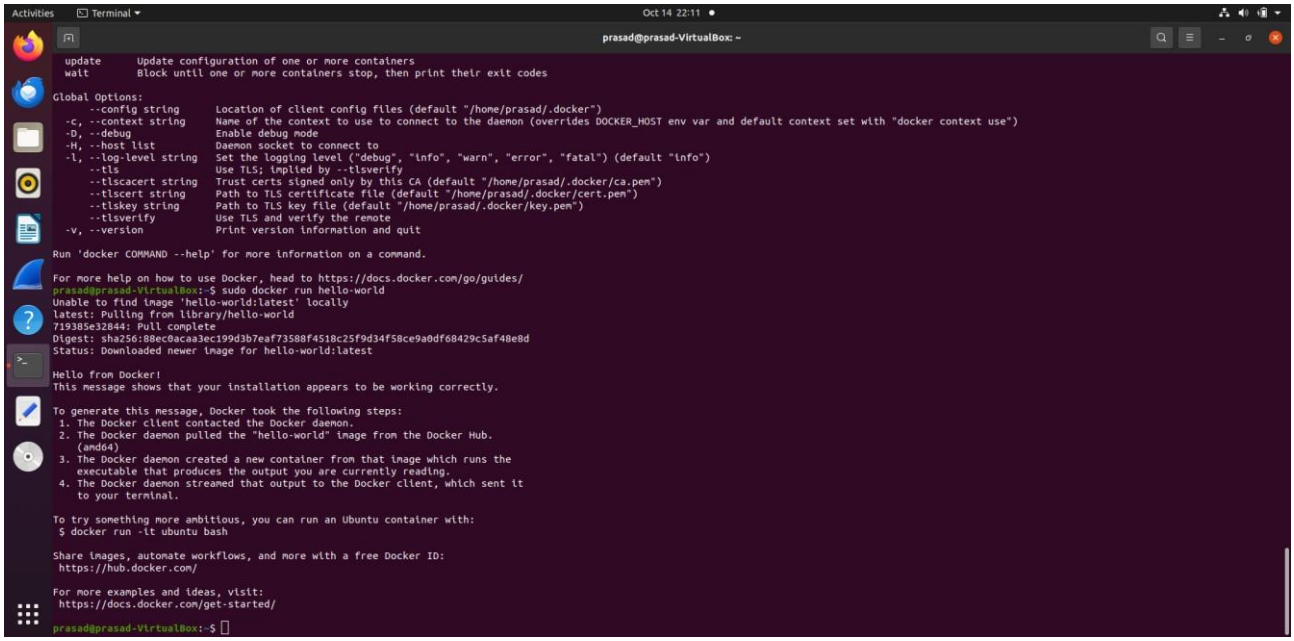
Kube-Proxy

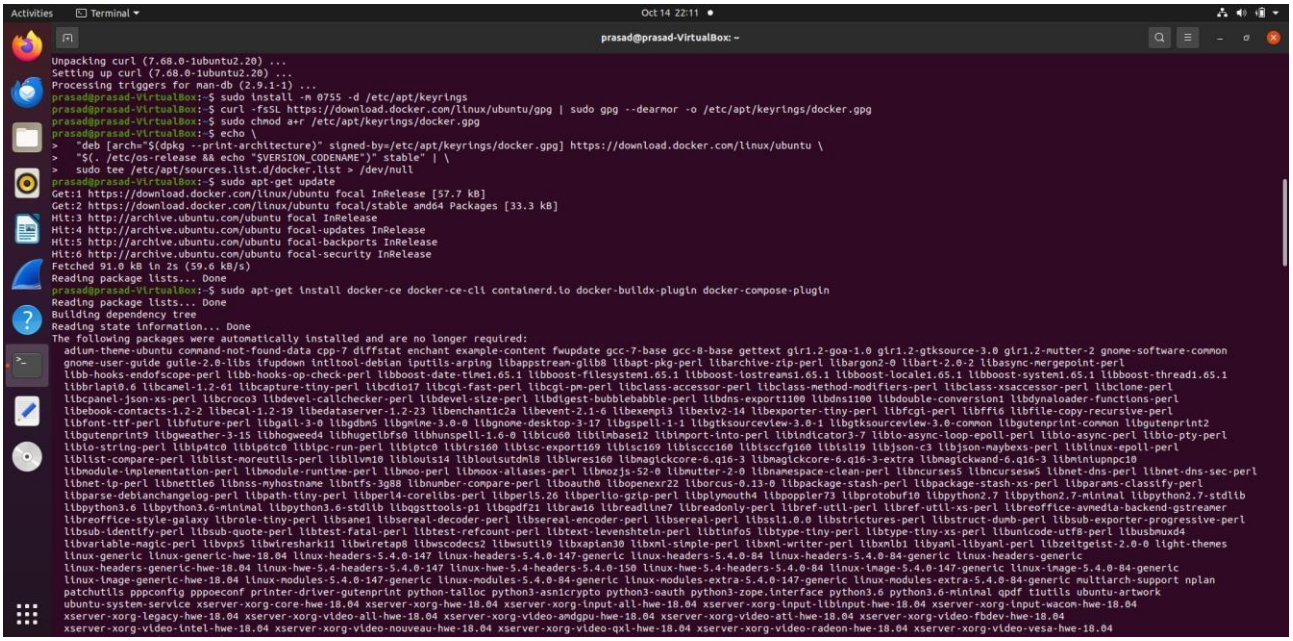
- Assign IP to each pod.
- It is required to assign IP addresses to Pods (dynamic)
- Kube-proxy runs on each node & this makes sure that each pod will get its unique IP Address.
- These 3 components collectively consist of ‘node’.

INSTALLATION:

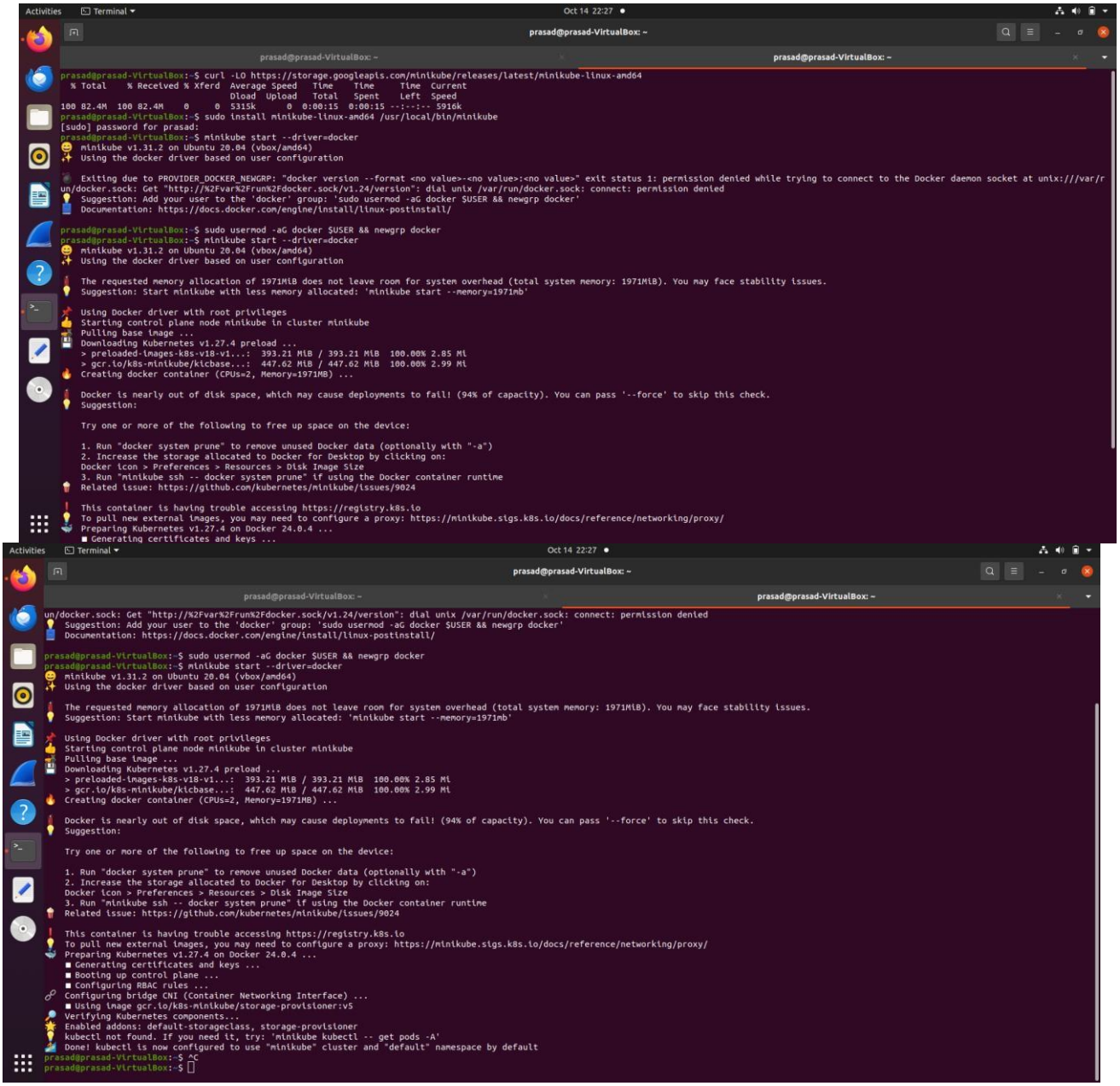
1. Install Docker



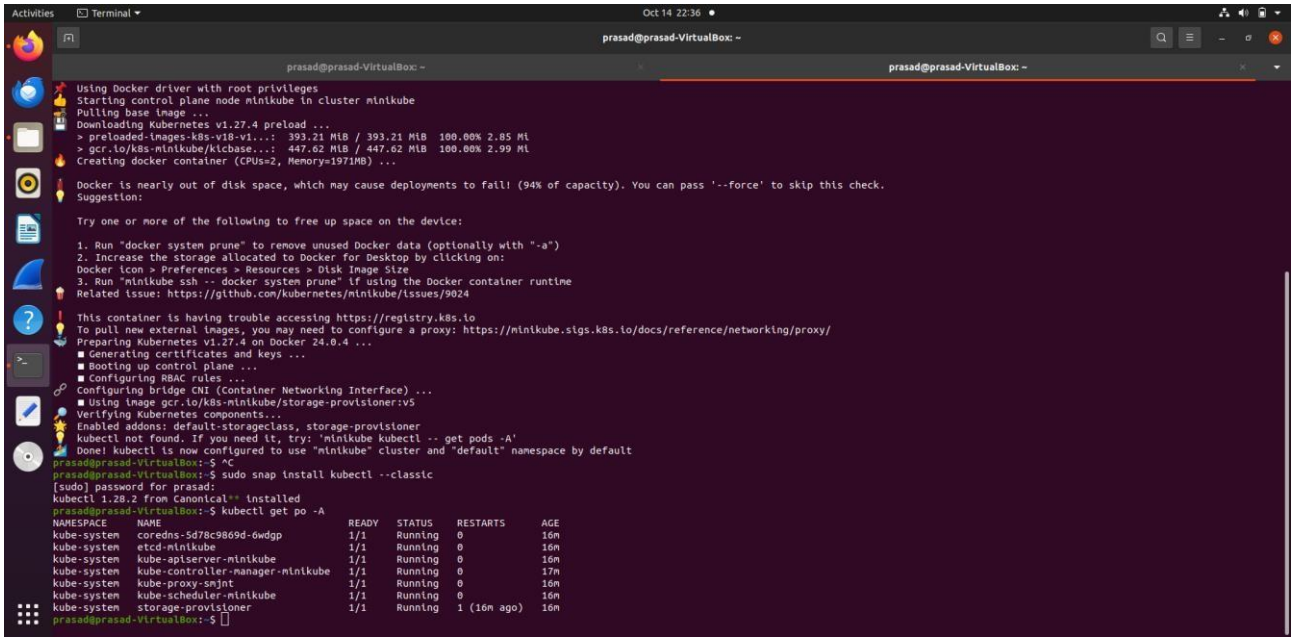




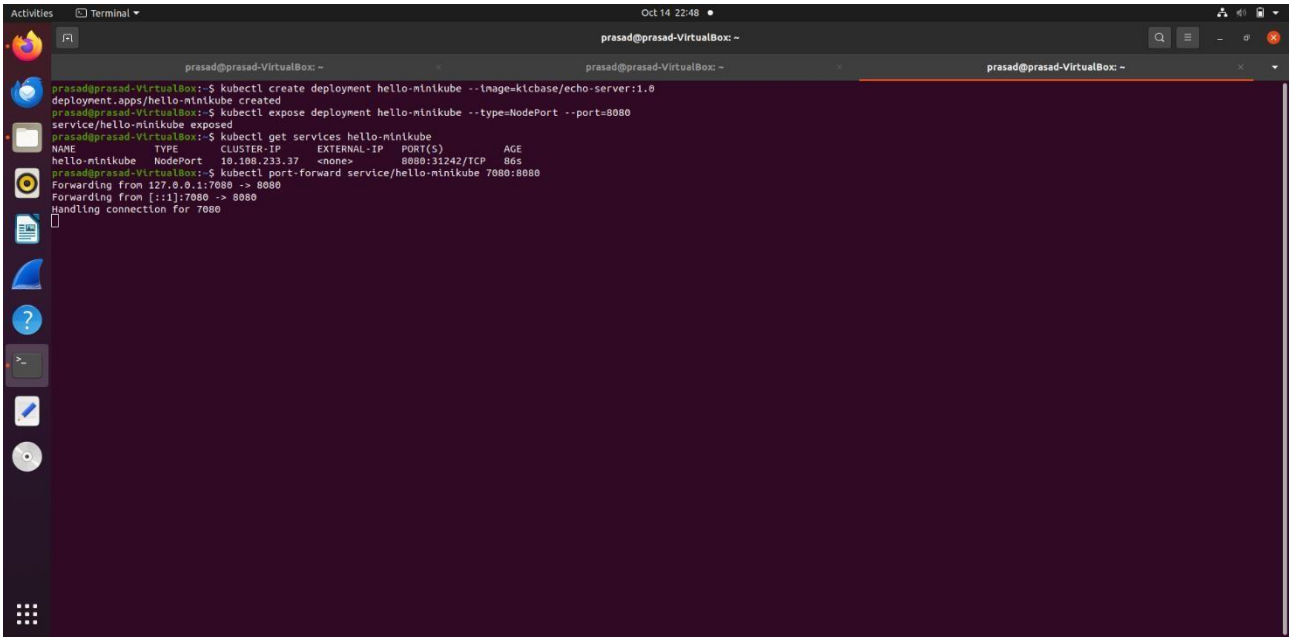
2. Install minikube using following commands

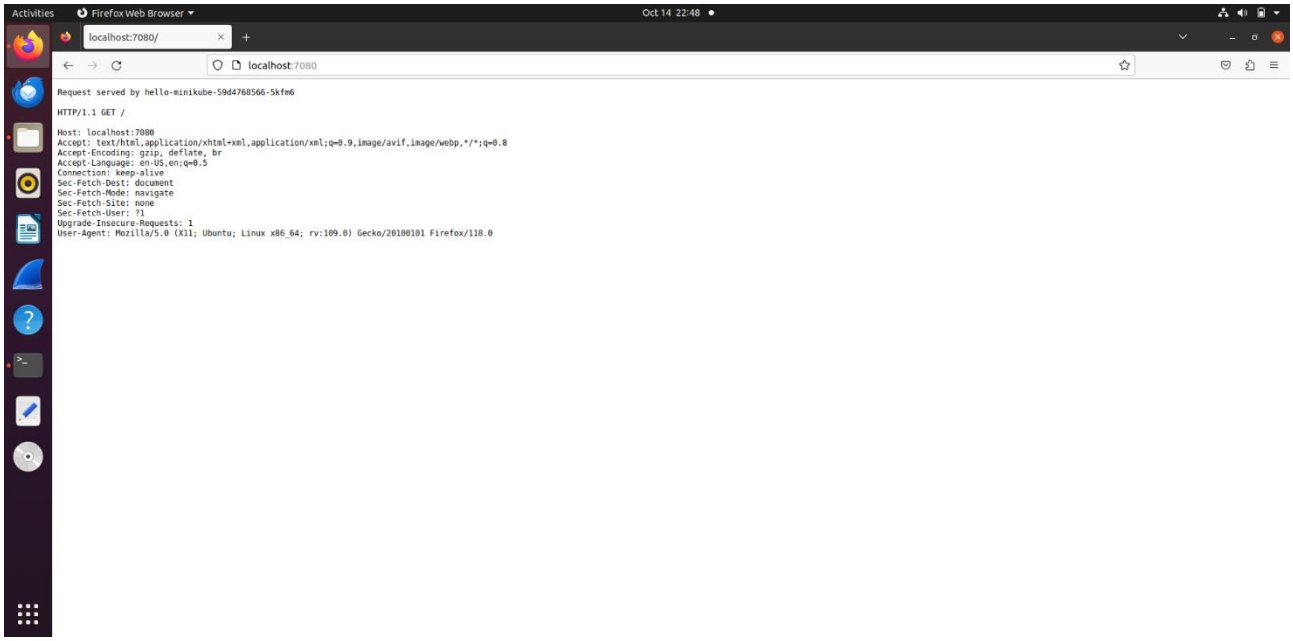


3. Install kubectl



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.

Name : Altaf Alam

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

Written Assignment No 1

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

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

1. Role-Based Access Control (RBAC): Implement RBAC to control who can perform actions within the cluster. Define roles and permissions to restrict access to critical resources, minimizing the risk of unauthorized actions.

2. API Server Access Control: Limit access to the Kubernetes API server by defining network policies and using technologies like firewalls and API gateways. Restricting API server access reduces the attack surface.

3. Pod Security Policies: Enforce security policies at the pod level to control actions like running as a privileged user, accessing the host network, or using a host's PID namespace.

4. Resource Quotas and Limit Ranges: Define resource quotas to limit the number of pods or resource consumption per namespace. Limit ranges can restrict resource requests and limits at a namespace level.

5. Container Security: Employ secure container images, regularly update them, and scan for vulnerabilities. Implement runtime protection mechanisms like AppArmor or SELinux to isolate containers.

6. Network Policies: Use network policies to control network traffic between pods. Limit communication to necessary ports and protocols, minimizing the risk of lateral movement in case of a breach.

7. Secrets Management: Use Kubernetes Secrets for sensitive data like API keys and passwords. Avoid storing secrets in plaintext in YAML files or in environment variables.

8. Pod Identity and Service Accounts: Utilize Service Accounts to manage access to Kubernetes resources. Implement Pod Identity to ensure that pods can access only the resources they need.

9. Audit Logging and Monitoring: Enable Kubernetes audit logging to track all API server requests. Use a robust monitoring and alerting system to detect and respond to security incidents.

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

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

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

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

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

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

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

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

1. Encryption: Encryption is the process of converting data into a code to prevent unauthorized access. There are two main types:
   - Data-at-rest Encryption: This encrypts data stored on disk or in databases, ensuring that even if physical storage devices are compromised, the data remains secure.
   - Data-in-transit Encryption: This secures data as it's transmitted over networks. Techniques like SSL/TLS encrypt data during transmission, preventing eavesdropping.

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

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

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

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

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

1. Install an Ingress Controller: Before using Ingress, you need to have an Ingress controller running in your cluster. Popular choices include Nginx Ingress Controller and Traefik. Install the desired controller based on your requirements.

2. Define an Ingress Resource: Create an Ingress resource, which defines the rules for routing external traffic to your service. Here's an example of an Ingress resource:

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

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

3. DNS Configuration: Ensure that the DNS record for the host specified in the Ingress resource (e.g., "example.com") points to the IP address of the Ingress controller's load balancer.

4. SSL/TLS Configuration (Optional): If you need to secure your Ingress with SSL/TLS, create a Kubernetes Secret containing the SSL certificate and key, and reference it in the Ingress resource.

5. Apply the Ingress Resource: Use the `kubectl apply -f` command to apply the Ingress resource to your cluster:

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

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

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

Q.4 Which service protocols does Kubernetes ingress expose?

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

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

- TCP and UDP Services: To expose services that rely on the TCP or UDP protocol, you can use a Service resource with the `LoadBalancer` type or use a different controller designed for those protocols.

- WebSocket: If you need to support WebSocket connections, some Ingress controllers, like Nginx Ingress, have WebSocket support. You can specify WebSocket routes in your Ingress resource.

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

Name : Altaf Alam

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

Written Assignment No 1

Q.1 How to deploy Lambda function on AWS?

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

1. AWS Account Setup: Before you can deploy a Lambda function, you need an AWS account. If you don't have one, create an AWS account by visiting the AWS Management Console.

2. Create a Lambda Function: Access the AWS Lambda service through the AWS Management Console. Click on "Create Function" and choose one of the following ways to create your function:
   - Author from Scratch: Write your code directly in the Lambda function editor.
   - Use a Blueprint: Start with a predefined blueprint, which can be particularly useful for common use cases like data processing, image resizing, or API endpoints.
   - Upload a .zip File: Package your code and any dependencies into a .zip file and upload it.
   - Upload a .jar File: For Java-based Lambda functions, you can upload a .jar file.

3. Configure the Function: Set the function's name, runtime (e.g., Node.js, Python, Java, etc.), execution role (permissions), and other optional settings like environment variables and VPC configuration.

4. Write the Function Code: If you haven't already written your code, use the Lambda function editor to write your function's code. Ensure that your code follows the AWS Lambda function handler signature for your chosen runtime.

5. Test the Function: You can test the function within the Lambda console. Input test data to see how your function behaves. Make sure it produces the expected results.

6. Configure the Triggers: Lambda functions can be triggered by various AWS services (e.g., S3, API Gateway, SNS, etc.). You can configure triggers within the Lambda console, specifying how your function is invoked.

7. Deploy the Function: Once you are satisfied with your Lambda function's code, configuration, and testing, click the "Create Function" button to deploy it.

8. Monitoring and Logging: Use AWS CloudWatch to set up monitoring and logging for your Lambda function. This helps you track and troubleshoot the function's performance.

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

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

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

1. Console Deployment: You can deploy Lambda functions directly through the AWS Management Console by writing code in the Lambda editor or by uploading deployment packages.

2. AWS Command Line Interface (CLI): The AWS CLI allows you to create, update, and deploy Lambda functions using commands in your terminal. This is particularly useful for automating deployments and integrating Lambda functions into scripts or CI/CD pipelines.

3. AWS SDKs: AWS provides SDKs for various programming languages, such as Python, Node.js, Java, etc. These SDKs allow you to interact with AWS services, including Lambda, programmatically.

4. Serverless Framework: The Serverless Framework is an open-source framework that simplifies the deployment of serverless applications, including Lambda functions. It provides a declarative configuration and supports multiple runtimes.

5. AWS SAM (Serverless Application Model): AWS SAM is an open-source framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way of defining the Amazon API Gateway APIs, AWS Lambda functions, and Amazon DynamoDB tables needed by your serverless application.

6. Third-Party Tools: There are various third-party tools and platforms that offer integrations with AWS Lambda for deployment and management, such as AWS CodePipeline, Jenkins, and Travis CI.

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

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

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

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

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

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

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

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

AWS Lambda functions are composed of several components:

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

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

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

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