Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ SAIKARTHIK IYER

of IT Department, Semester V with

Roll No. 41 has completed a course of the necessary

experiments in the subject AdvDevOps under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2024 - 2025

Teacher In-Charge

Head of the Department

Date 18/10/24

Principal

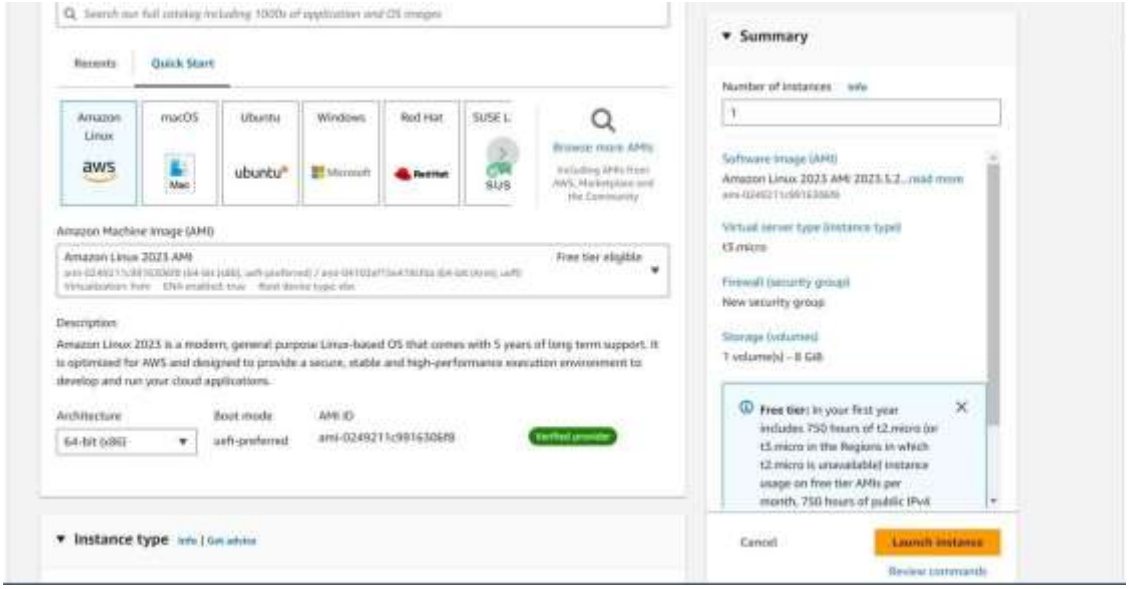
THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

CONTENTS

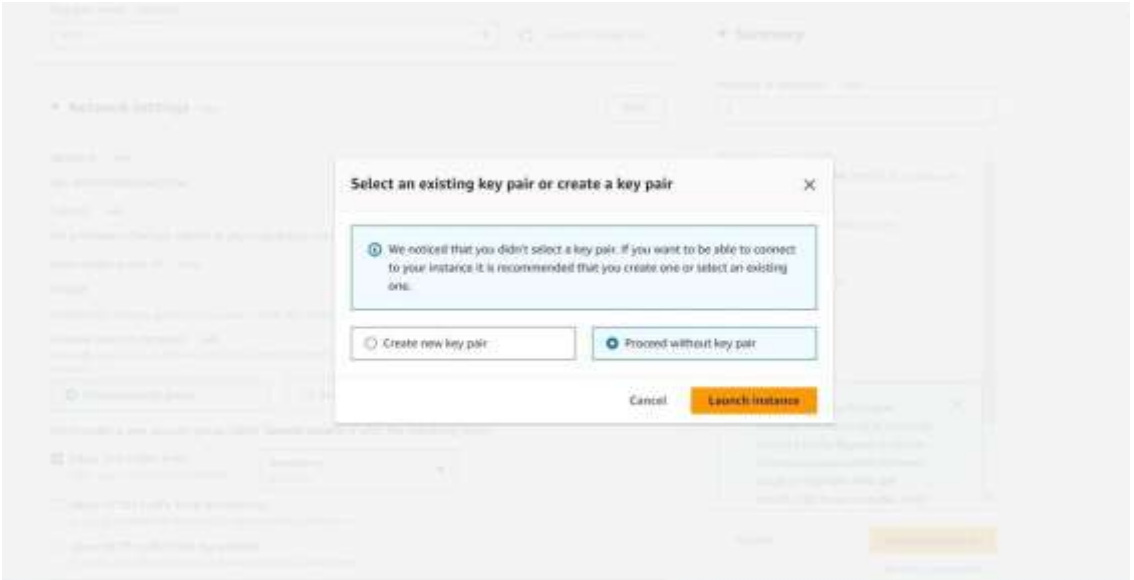
Name : Saikarthik Iyer
Roll No : T1341
Subject : Advance Devops Lab 1
Aim : To implement Ec2 and Cloud9 Service on AWS servers.

Creating an Instance using Ec2 :

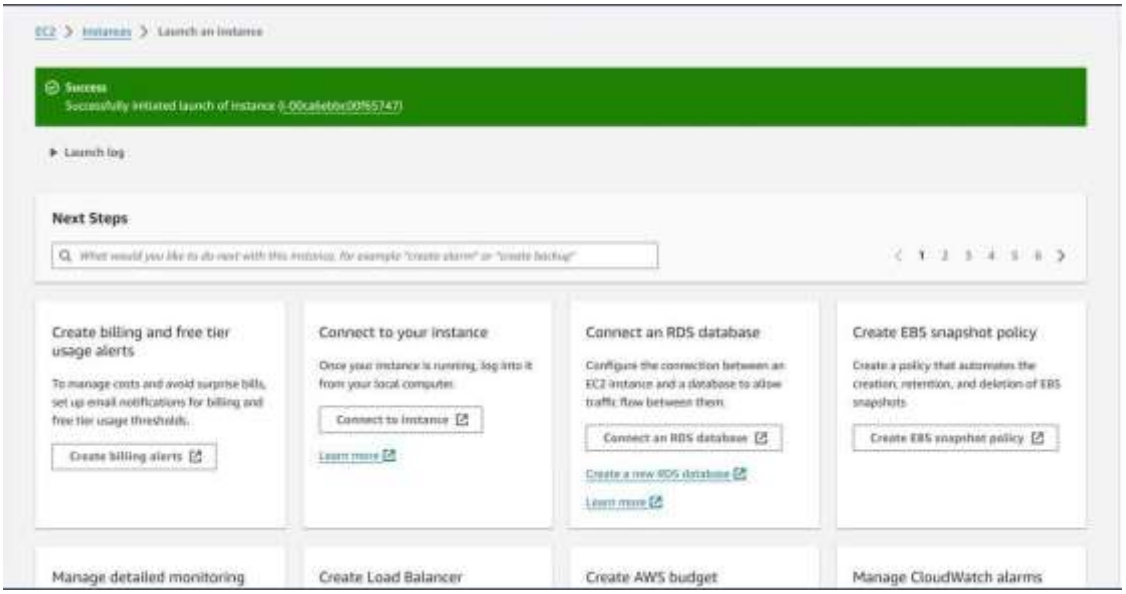
Step – 1 : Launch an Instance.



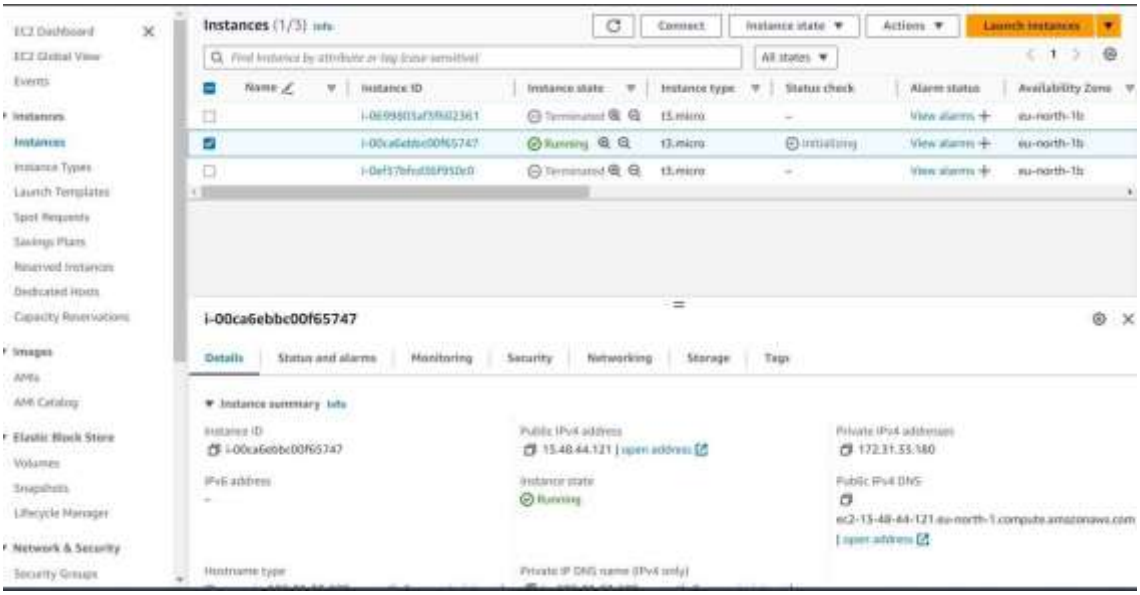
Step – 2 : Without key pair.



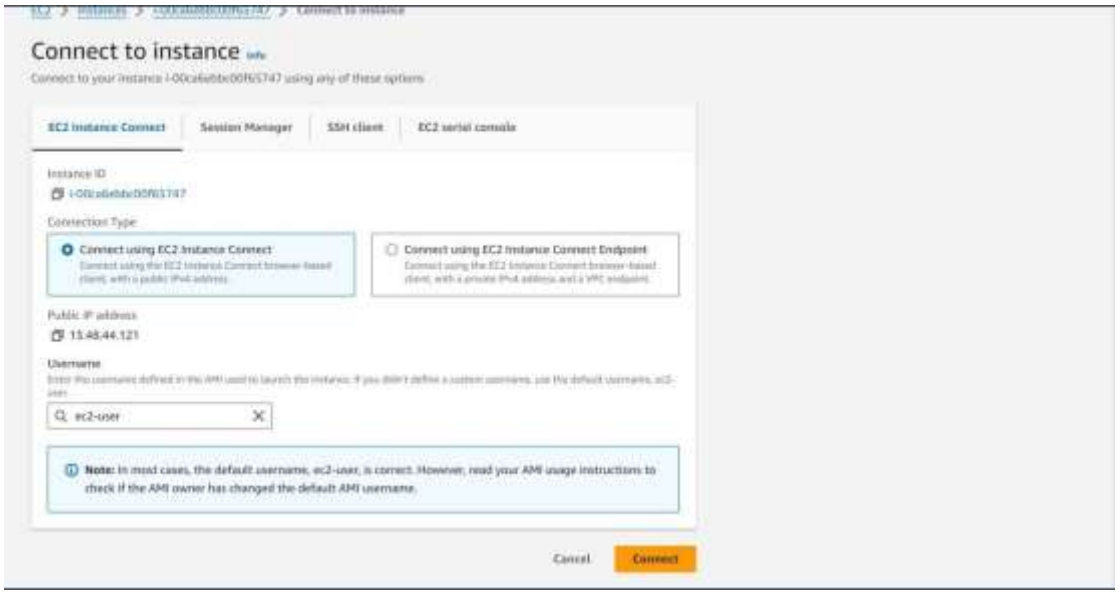
Step – 3 : Instance creation successful.



Step – 4 : Check the created instance.

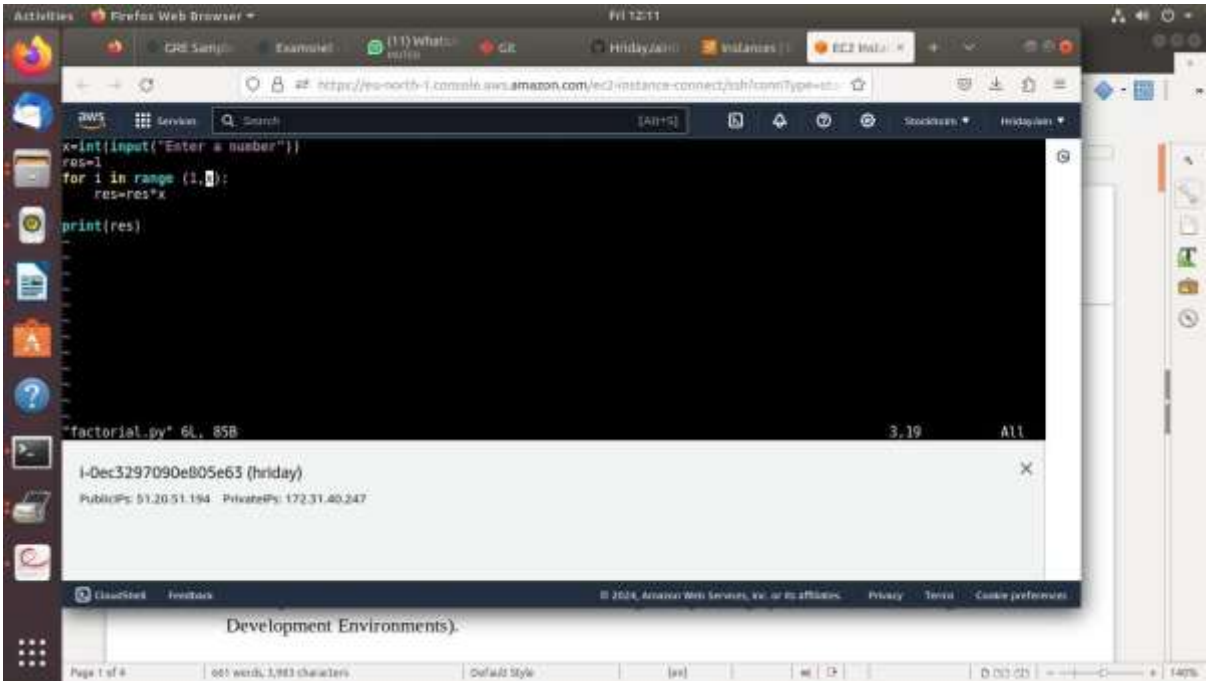


Step – 5 : Connect the created instance.

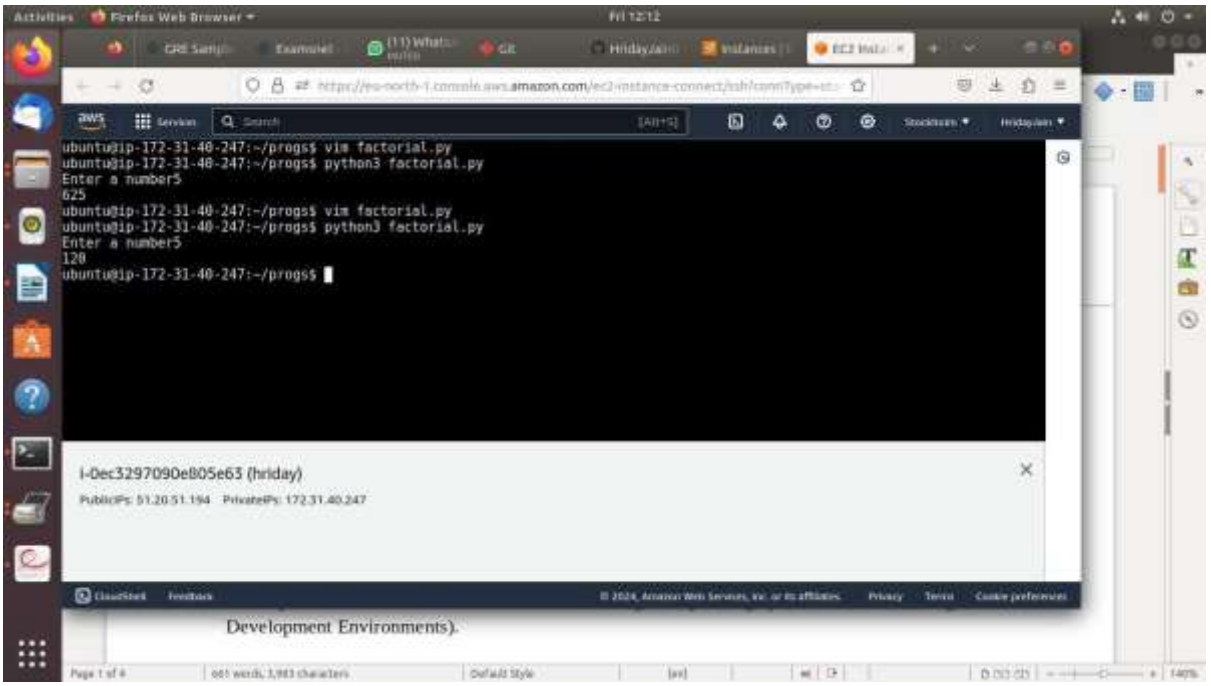


Step – 6 : Linux Instance created successfully.

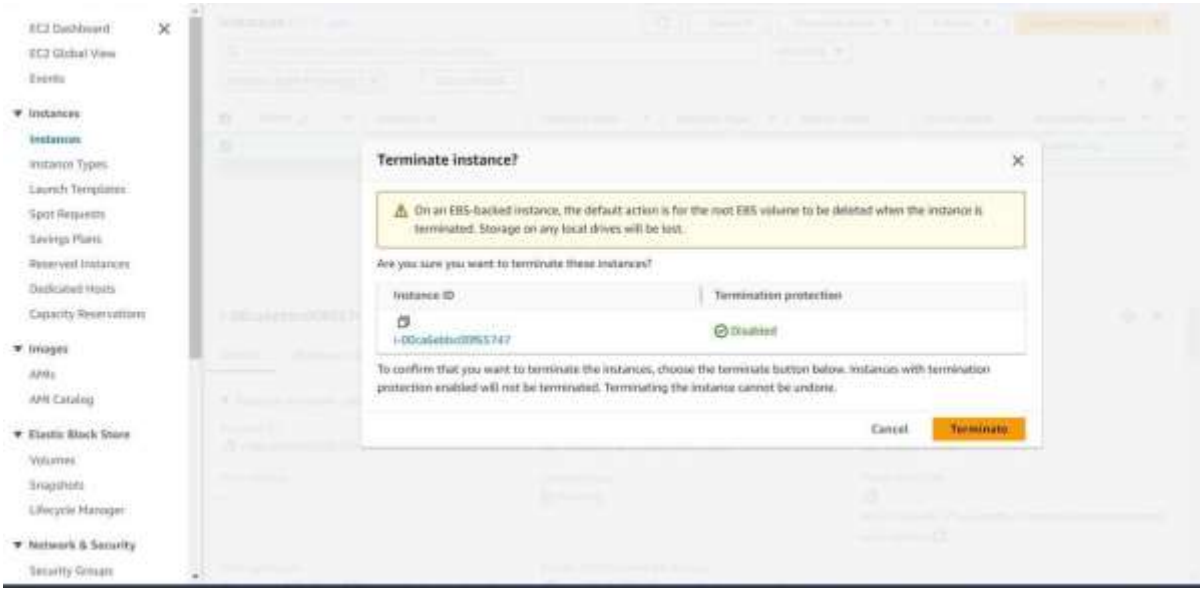
Step 7 – Write a program using the Aws command line



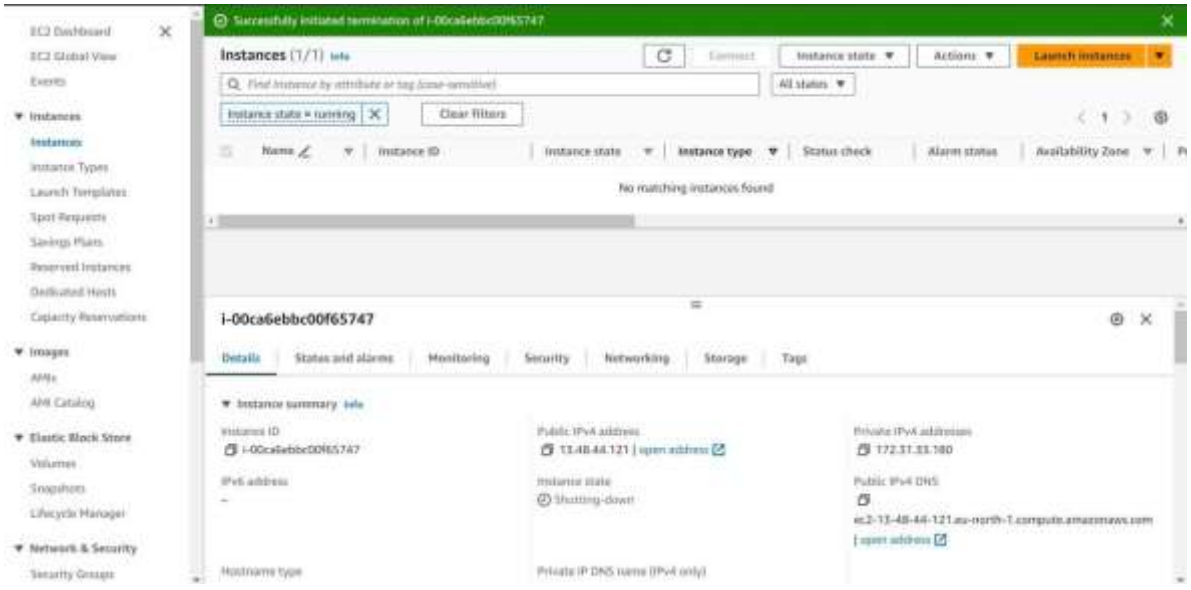
Step 8 – Execute the program using the Aws command line



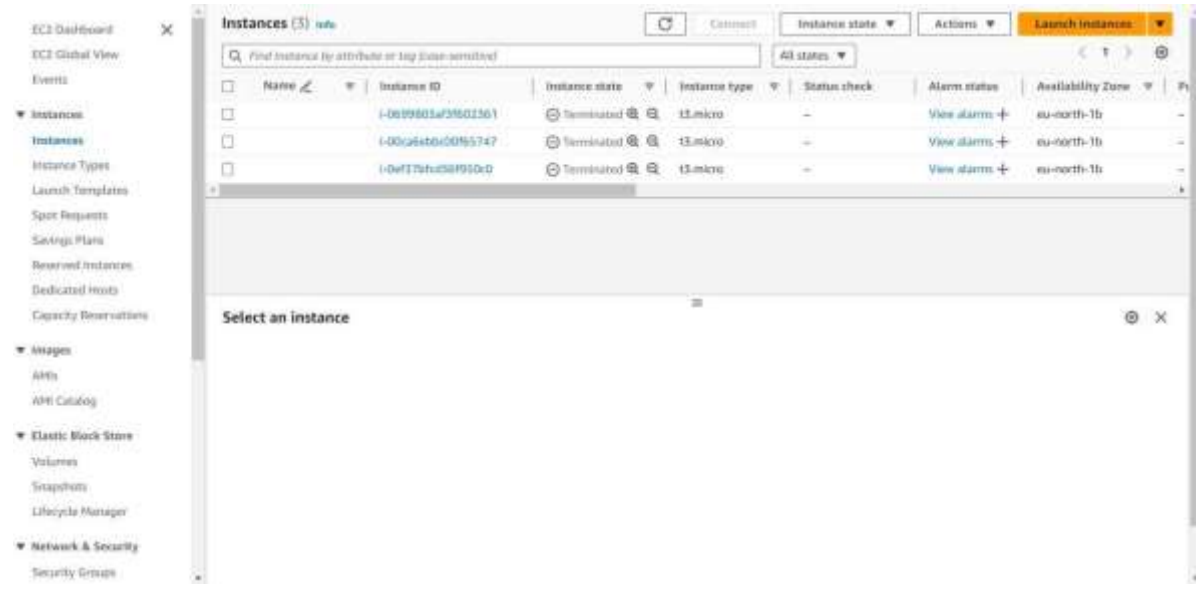
Step – 9 : Terminate Instance.



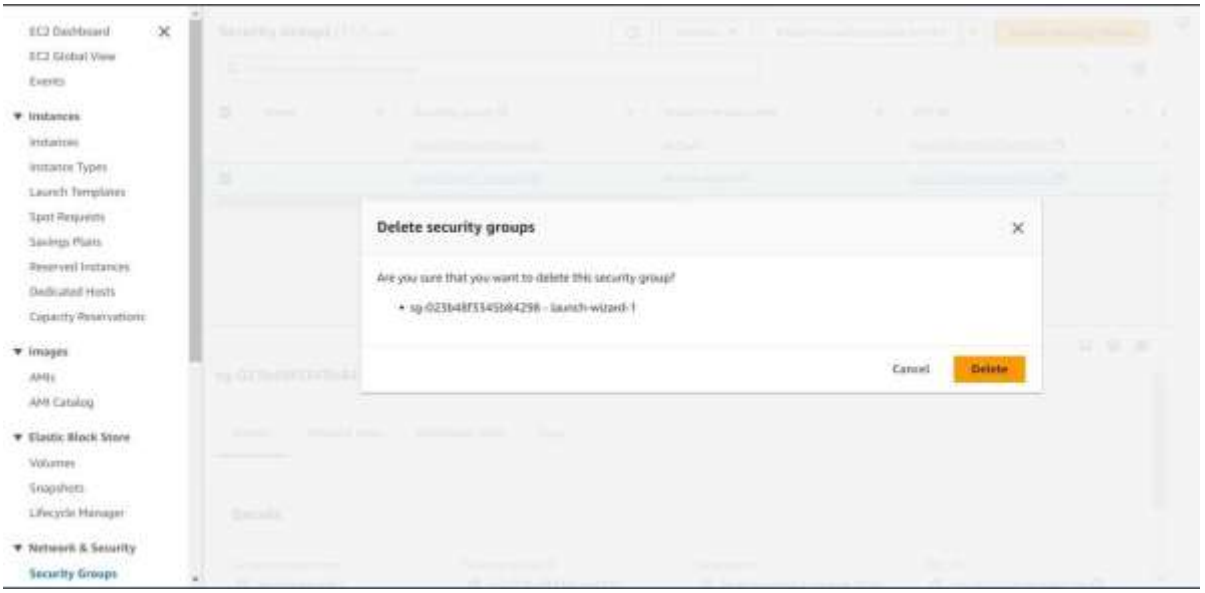
Step – 10 : Instance Terminated Successfully.



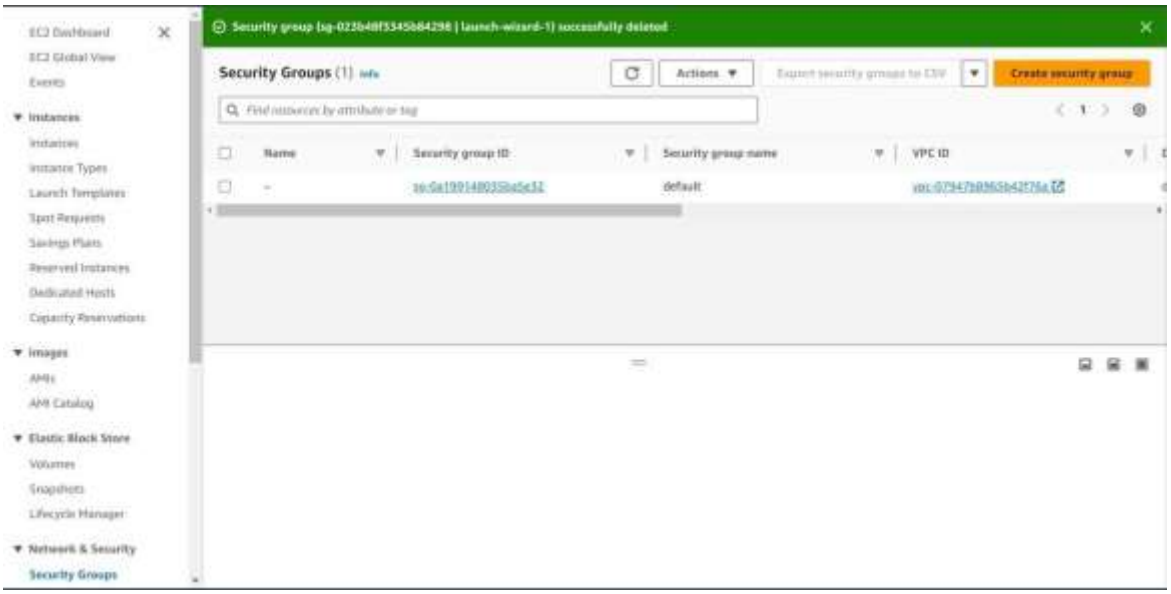
Step – 11 : Check for running instance.



Step – 12 : Delete the security group.

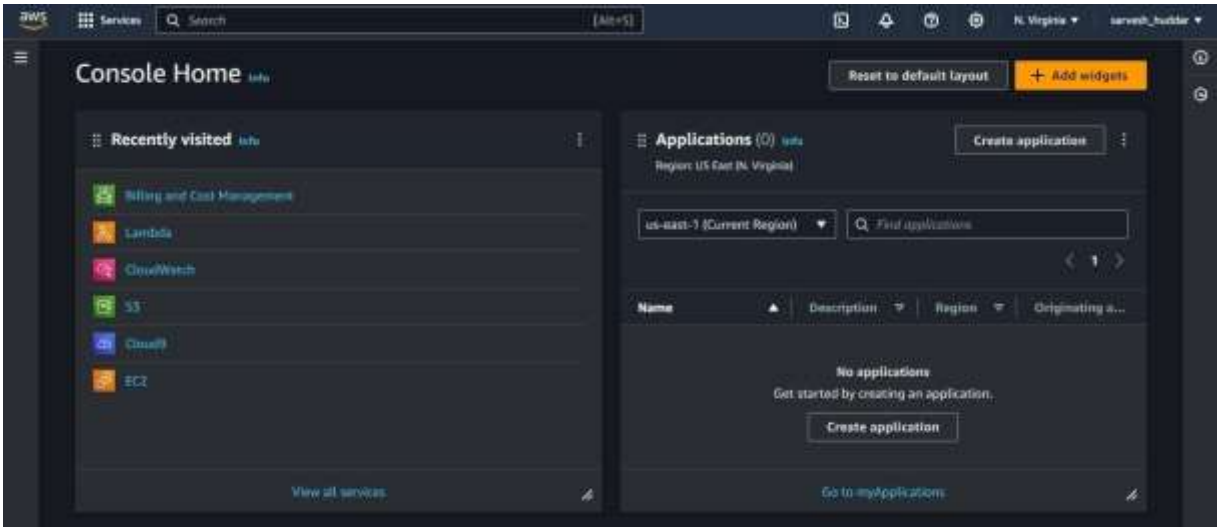


Step – 13 : Security group deleted successfully.

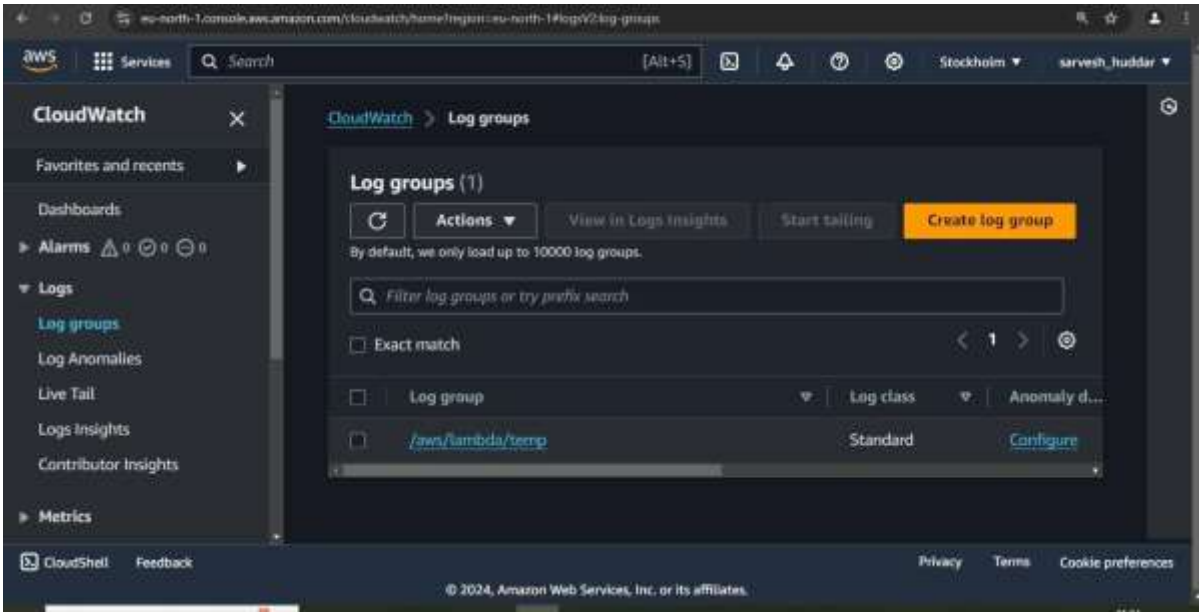


Creating an Instance using Cloud9 :

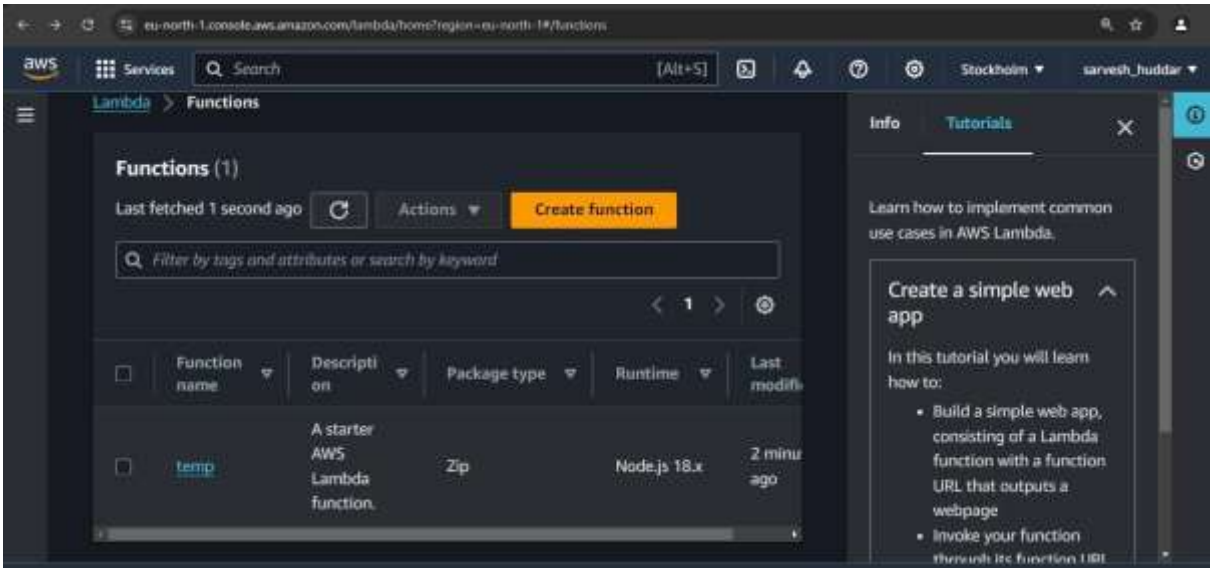
Step 1 - Create and configure the environment in Cloud9



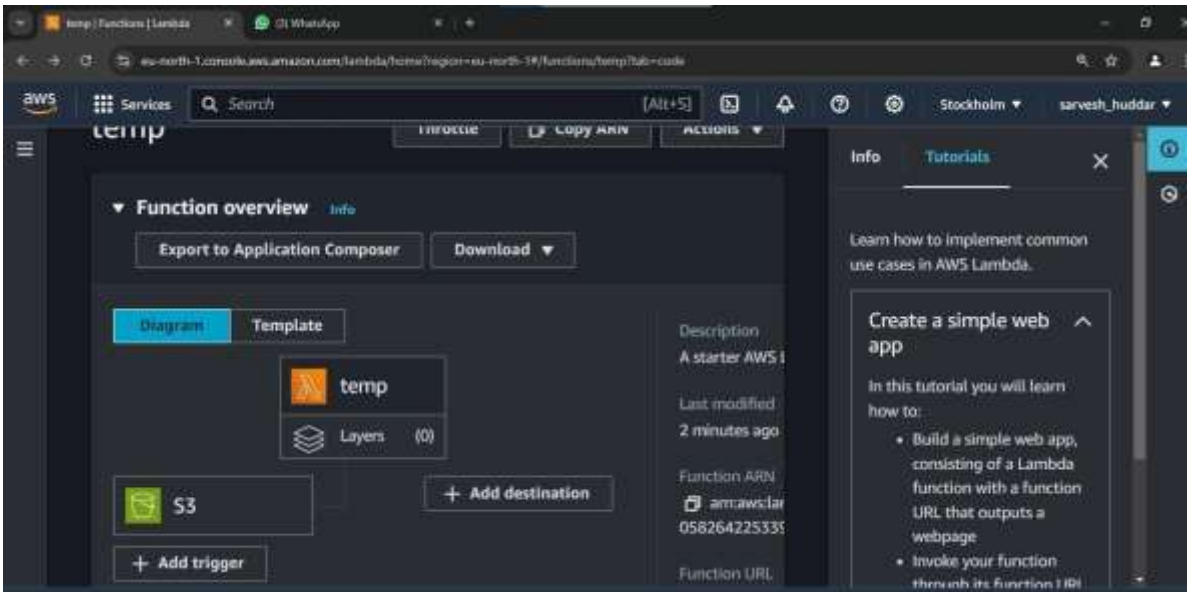
Step 2 – Create a new bucket in S3 and make a log group in Cloudwatch



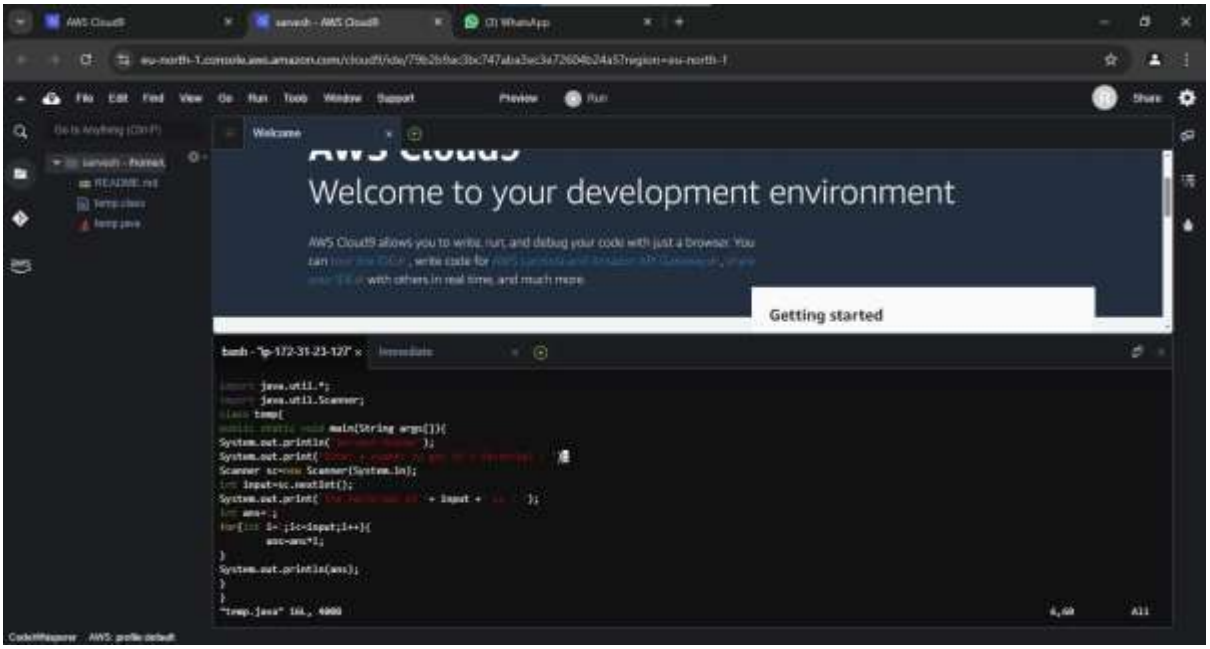
Step 3 – Open the Lambda Console and create a new function



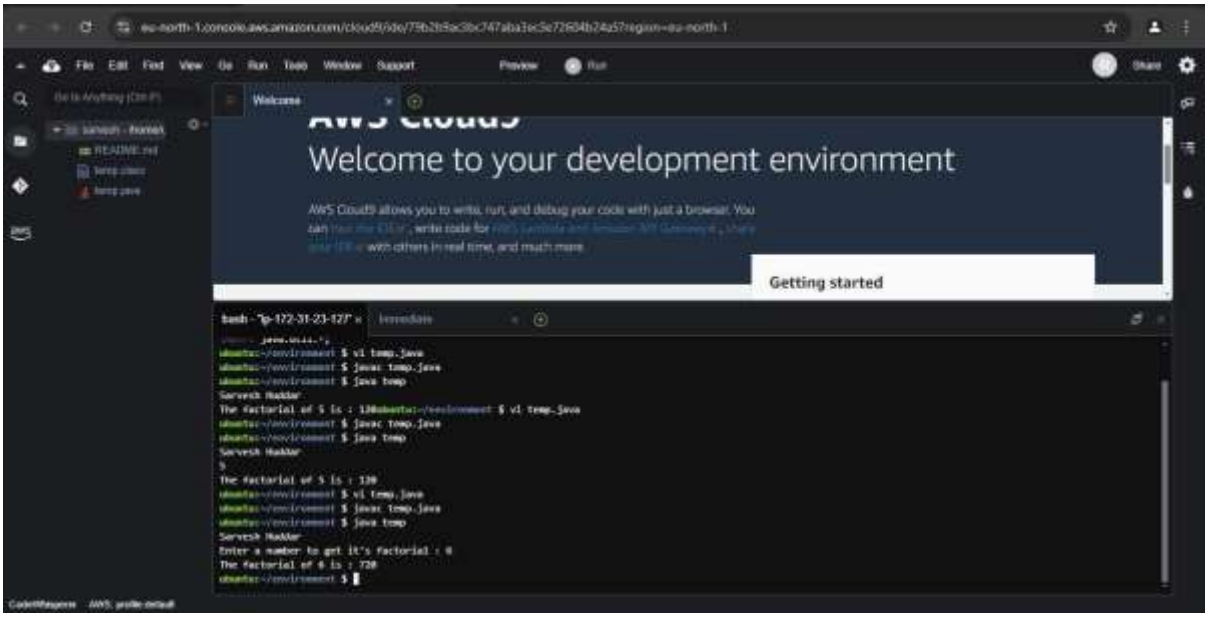
Step 4 - Add S3 Trigger and configure it with Cloudwatch logs for Lambda



Step 5 – Deploy the code from Cloud9



Step 6 – Use the command line to execute the program



Conclusion : We have learnt the creation of instances in AWS EC2 and Cloud9. This knowledge enhances the ability to efficiently develop, test, and deploy applications using AWS's scalable infrastructure.

Name : Saikarthik Iyer

Roll no : 41

batch T13

Assignment 03

• Key Features of Amazon S3:

1. Scalability: S3 can handle massive amounts of data, scaling automatically as the volume of stored objects increases.

2. Durability and Availability: S3 is designed for 99.999999999% durability and 99.99% availability of objects over a given year. Data is redundantly stored across multiple devices in multiple

facilities.

3. Security: S3 provides several security features including encryption of data at rest and in transit, access control mechanisms (such as IAM policies, bucket policies, and ACLs), and logging and

monitoring capabilities.

4. Cost-effectiveness: S3 provides different storage classes, allowing users to optimize costs by choosing the appropriate class based on access frequency and performance needs.

5. Data Management: S3 offers lifecycle management policies that can automatically transition objects between different

storage

classes or delete them after a specified period.

Aim: To implement lambda function Theory:

Amazon S3 (Simple Storage Service): Amazon S3 is a scalable object storage service provided by Amazon Web Services (AWS). It is designed to store and retrieve any amount of data, at any time, from anywhere on the web. S3 is primarily used for

storing files, such as images, videos, backups, and log files, but it can also store data in other formats like JSON, XML, or plain text.

AWS Lambda: AWS Lambda is a serverless compute service that automatically manages the underlying compute resources. It allows you to run code without

provisioning or managing servers. You only pay for the compute time consumed by your code, which makes it cost-effective for many applications.

• Key Features of AWS Lambda:

1. Serverless Computing: AWS Lambda abstracts the infrastructure,

2. Event-driven Execution: Lambda functions can be triggered by various AWS services, such as S3, DynamoDB, SNS, and CloudWatch. They can also be invoked directly via HTTP

requests

uhsaifnngyAdomluazeor nfuAncPtIioGnaitnewreasyp.onse to incoming requests. It can o

4. Pthayo-upsearn-udseoPf rciocinncgu:rYreonut eaxrecbuitliloednso.nly for the number of requests served and the compute time required to run your code. This makes it very cost-efficient for infrequent tasks.

5. Support for Multiple Languages: AWS Lambda supports several

Integration of S3 with Lambda: One of the most common use cases of AWS Lambda is to process files stored in Amazon S3. When a new object is uploaded to an S3 bucket, it can trigger a Lambda function that processes the object.

This

integration allows for automated workflows, such as data processing, file validation, image resizing, and real-time notifications.

• How S3 Triggers Lambda:

1. S3 Event Notifications: An S3 bucket can be configured to send event notifications to AWS Lambda when certain actions occur

(e.g., object creation, deletion, or modification). This notification

contains metadata about the event, such as the object key,

2. Lnbautimficc,kabeted

noam Faatniodun envc tretiniotngg et yErspxe eth. cution: When the event

e specified Lambda function. The function

receives the event metadata as input, and it can then perform dopatear,a tions such as reading the object from S3, processing the

and storing the results in another S3 bucket or a different AWS service.

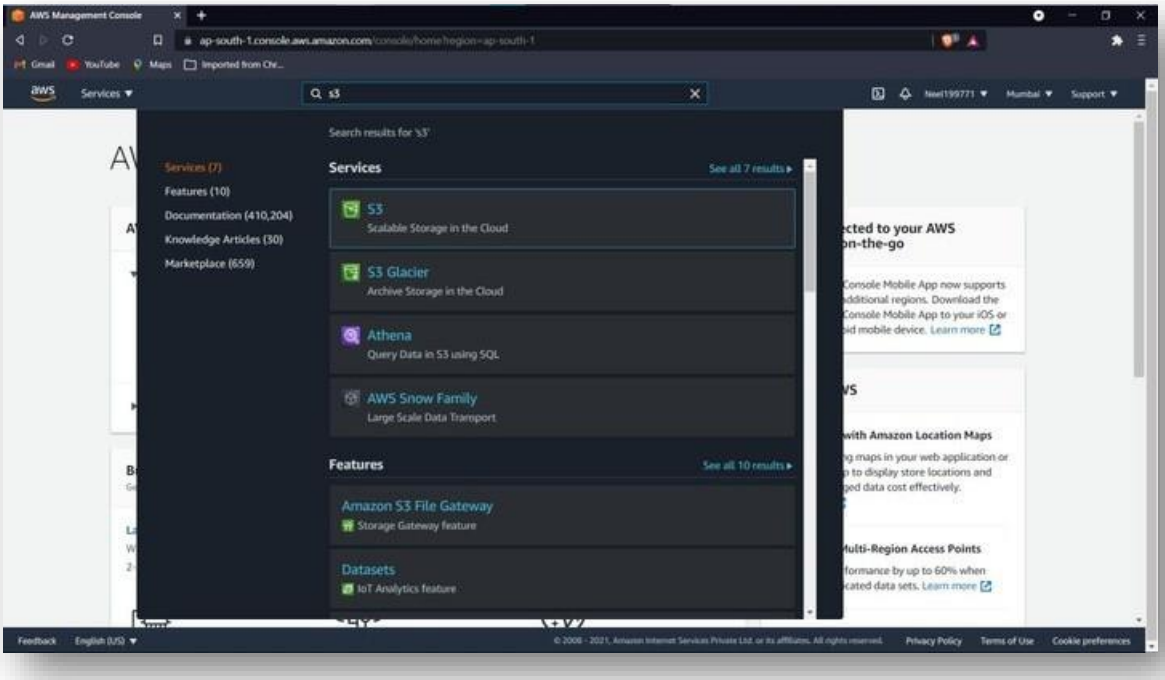
- Example Use Case: Imagine an application where users upload images to an S3 bucket. An S3 event can trigger a Lambda function that automatically resizes the image to different dimensions, compresses it, and saves the processed images back to the S3 bucket. The entire workflow is automated, and no servers are involved in the process.

Advantages of Using S3 with Lambda:

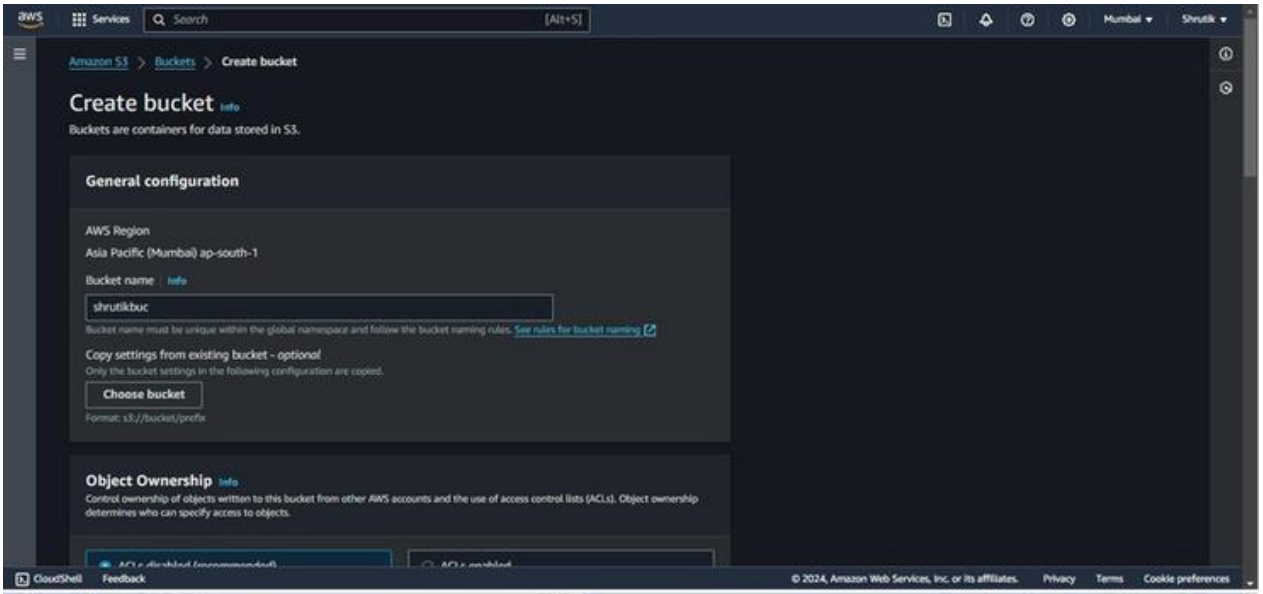
1. Automation: The integration enables seamless automation of workflows iwnittehrovuent ttihoen n. eed for human
2. Scalability: Both S3 and Lambda scale automatically with the amount of rdeaqtua easntsd. number of
3. Cost Efficiency: By using Lambda, you only pay for the compute time used, and by using S3, you only pay for the storage you consume.
4. Ease of Use: The combination of S3 and Lambda simplifies complex tasks like data processing, logging, and notifications.

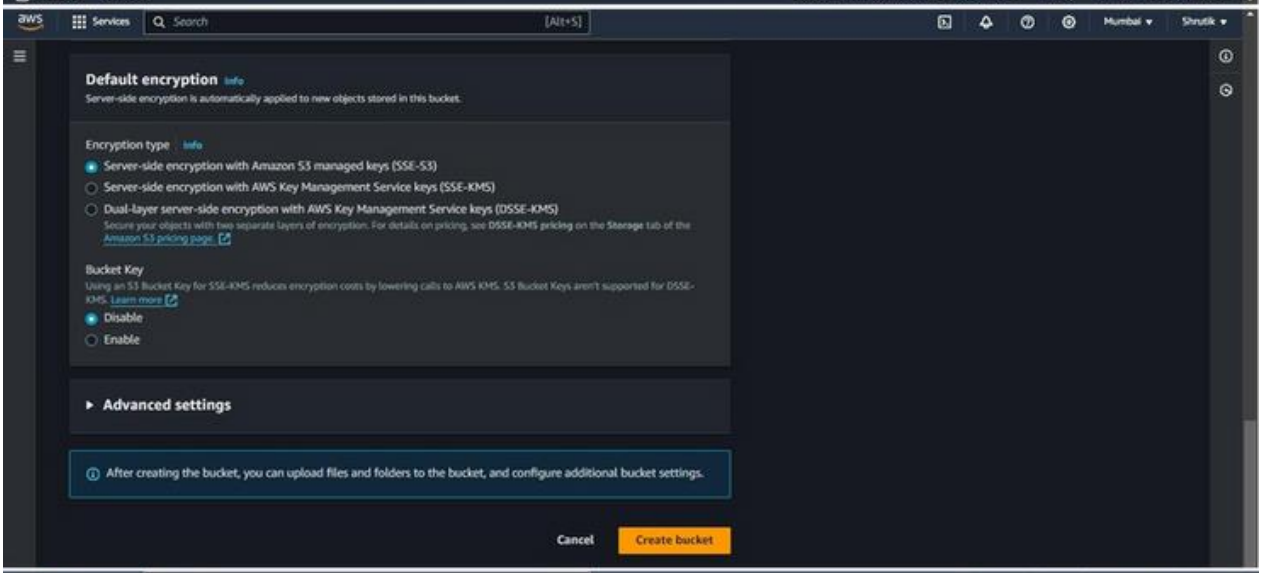
Steps:

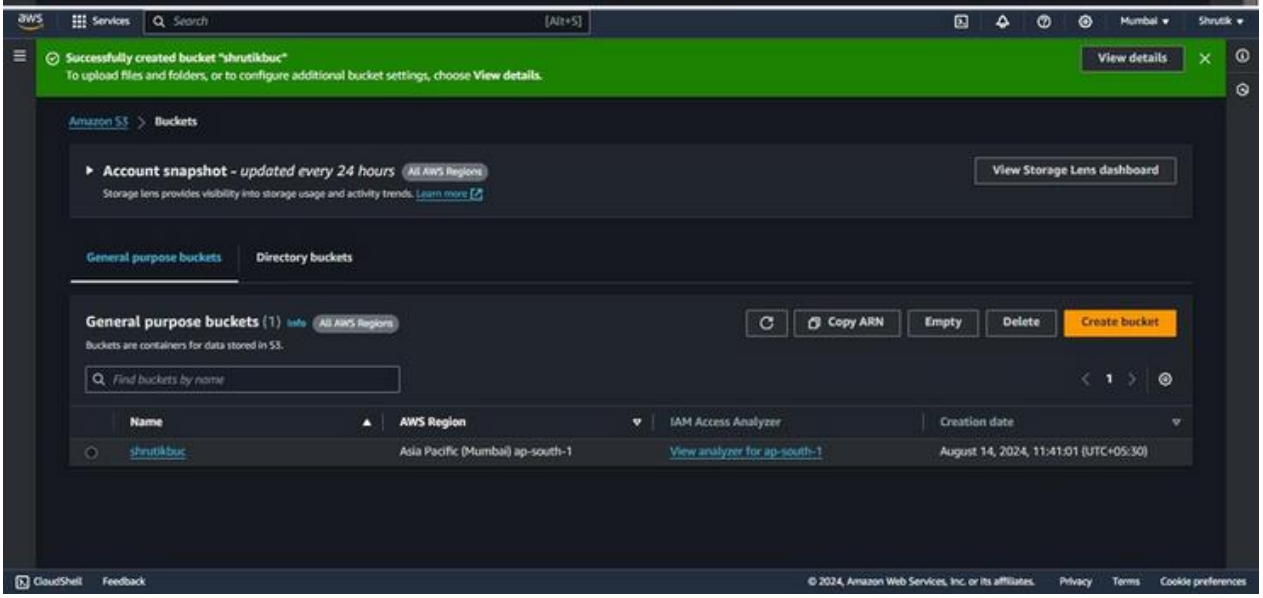
1. Login to AWS account and search S3



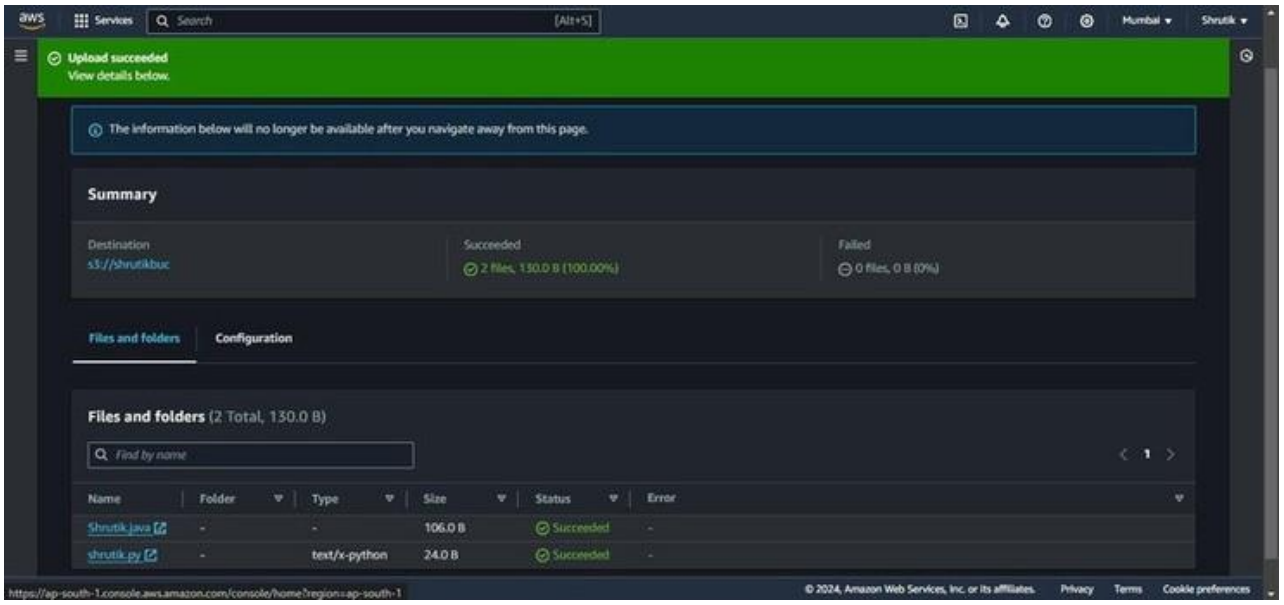
2. Create an S3 bucket by giving it a name



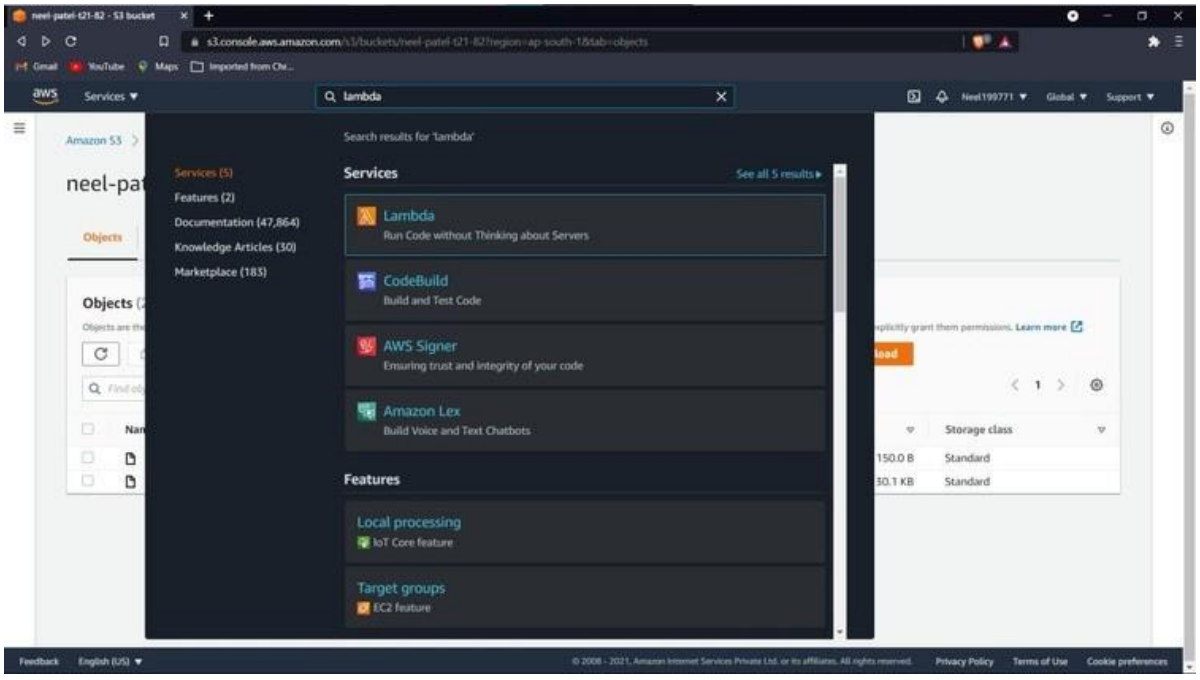




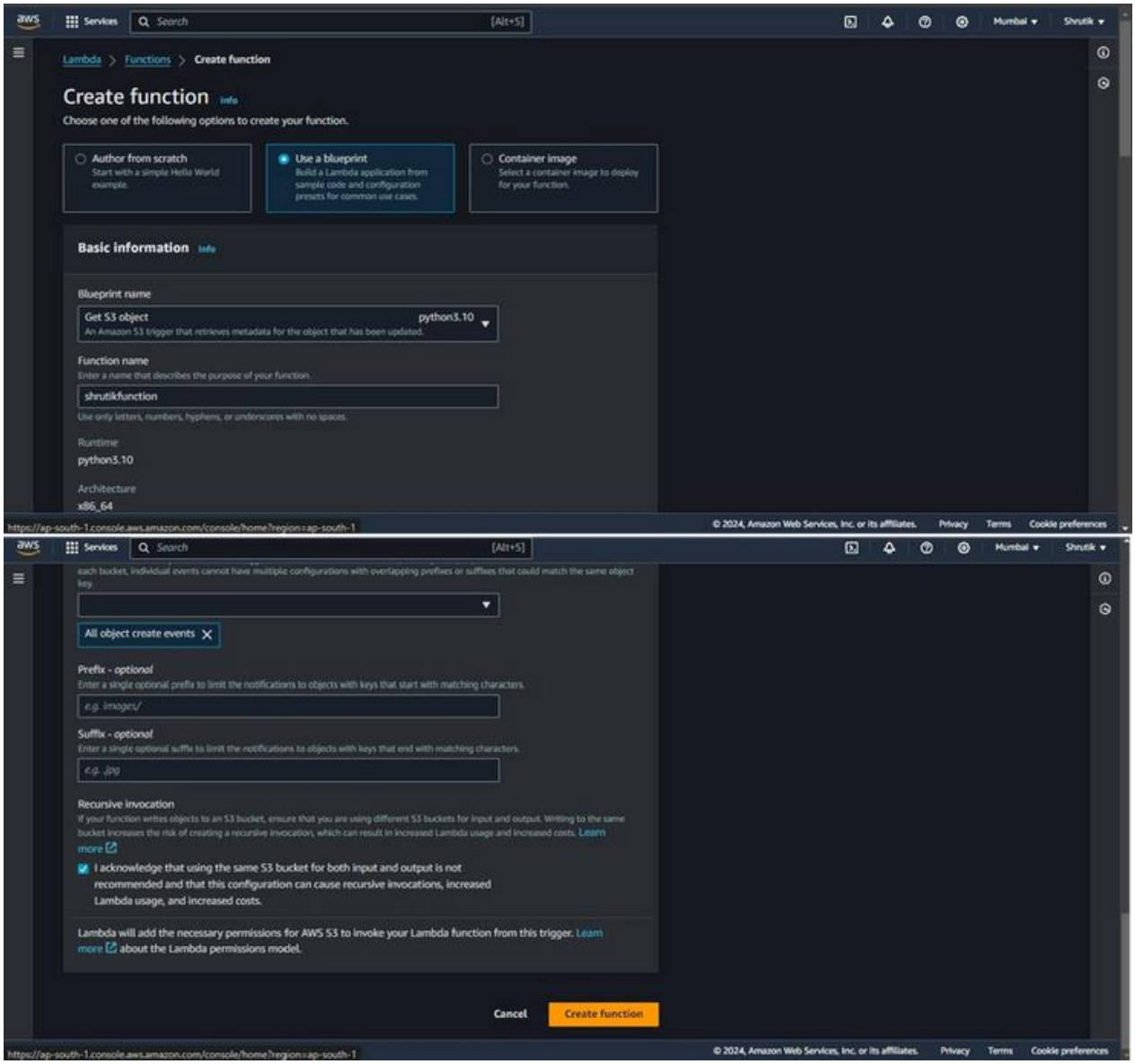
3. Click on upload and add any .py or .java files after S3 bucket is created in the object section



4. Search lambda

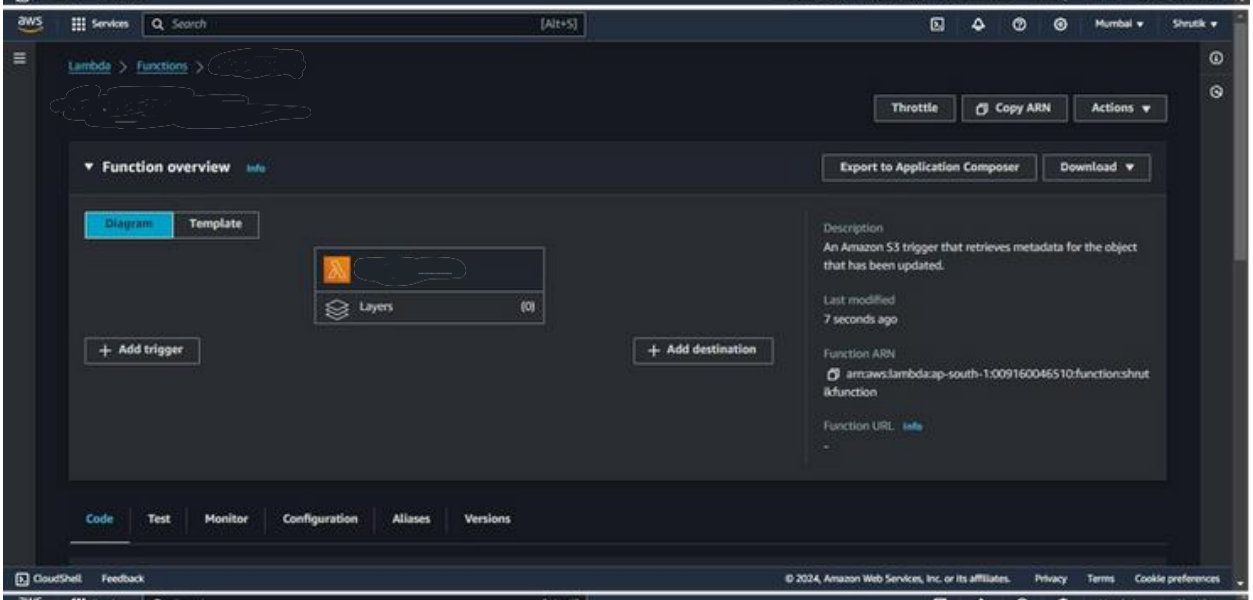


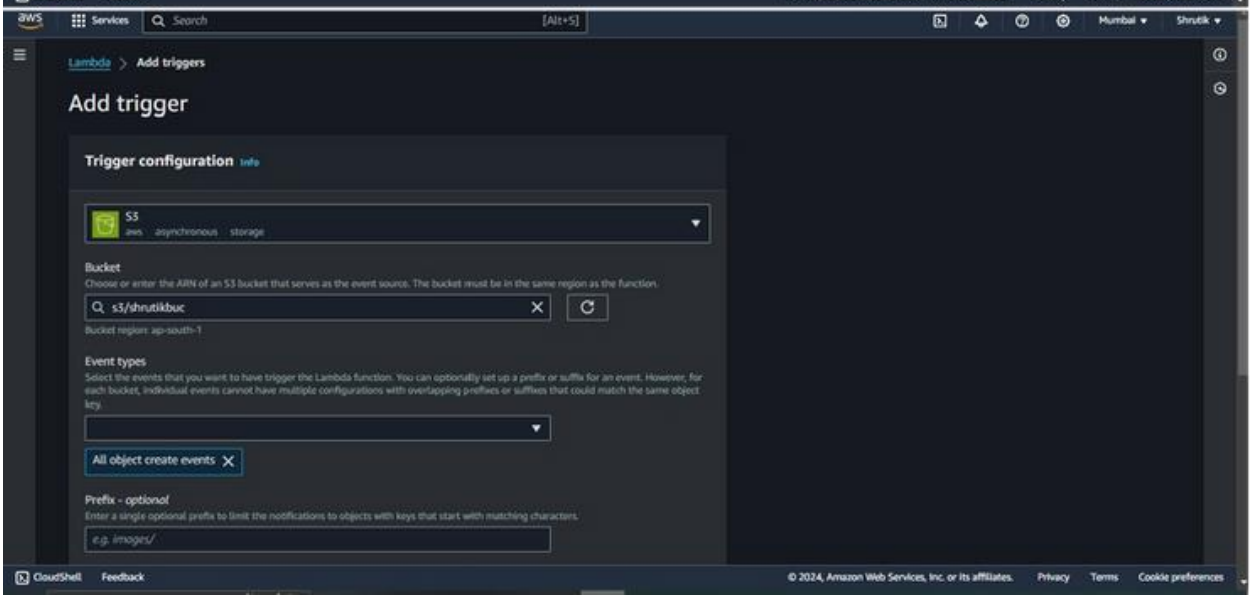
5. Create a function and configure as shown



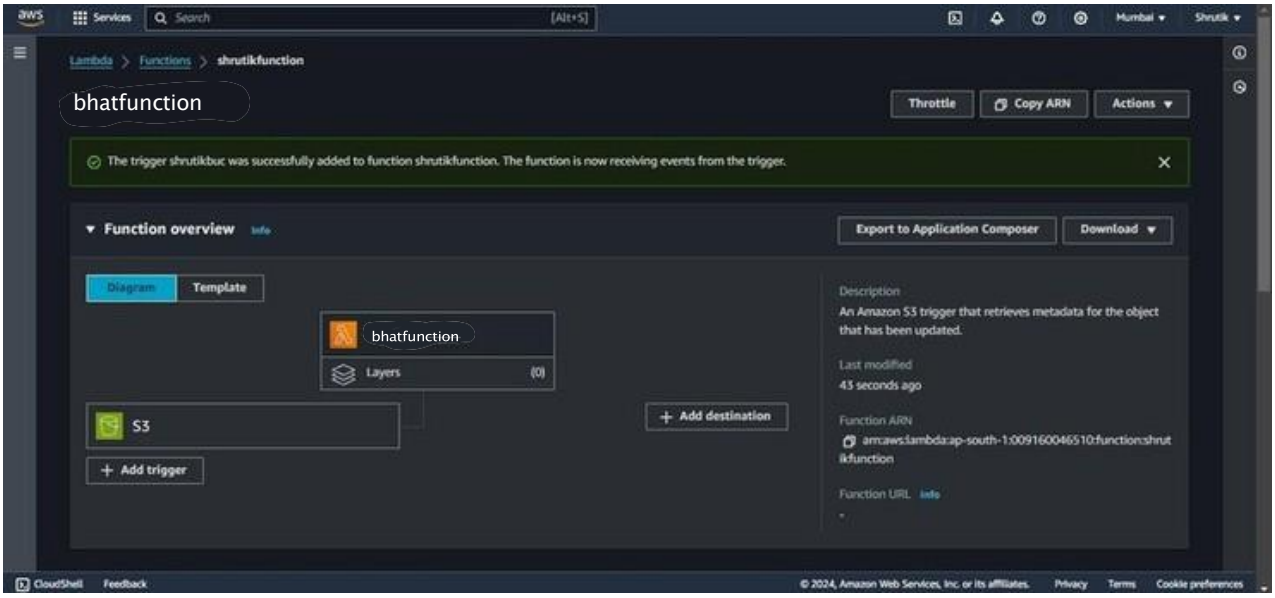
6. Select the bucket created & create trigger, click on create function


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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

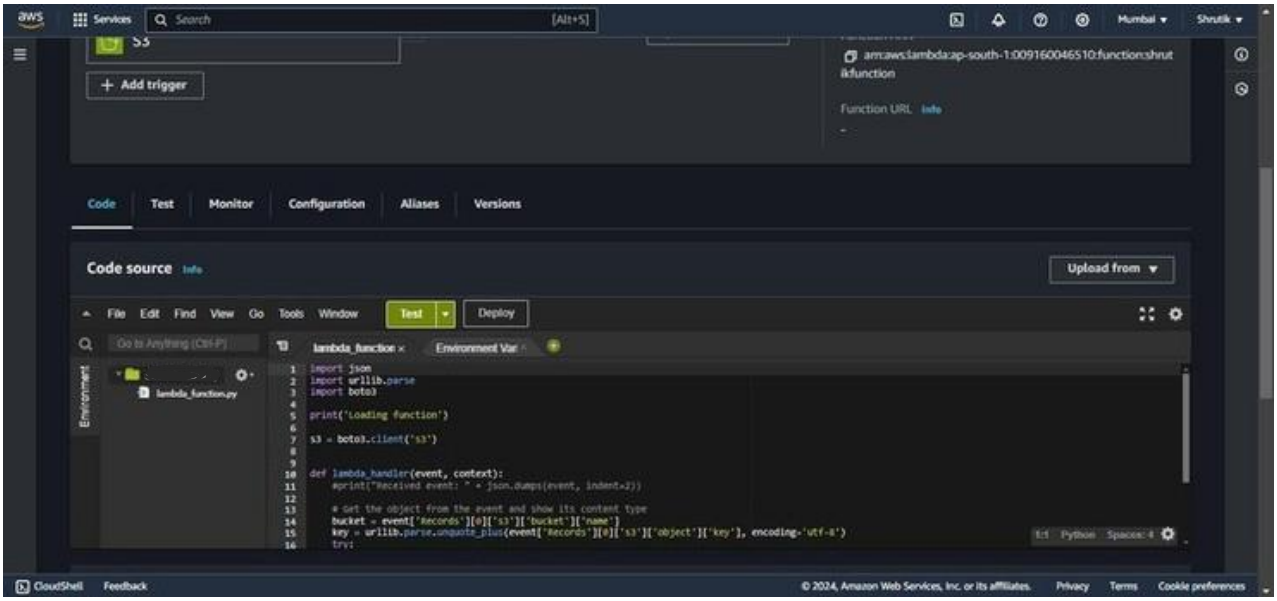

Services
Search
[Alt+S]
Mumbai
Shrutik
Lambda
Functions
Throttle
Copy ARN
Actions
Function overview
Info
Export to Application Composer
Download
Diagram
Template
Layers
(0)
Add trigger
Add destination
Description
An Amazon S3 trigger that retrieves metadata for the object that has been updated.
Last modified
7 seconds ago
Function ARN
arn:aws:lambda:ap-south-1:009160046510:function:shrutikfunction
Function URL
Info
Code
Test
Monitor
Configuration
Aliases
Versions
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Mumbai
Shrutik
Lambda
Add triggers
Add trigger
Trigger configuration
Info
S3
aws
asynchronous
storage
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/shrutikbuc
Bucket region: ap-south-1
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/
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

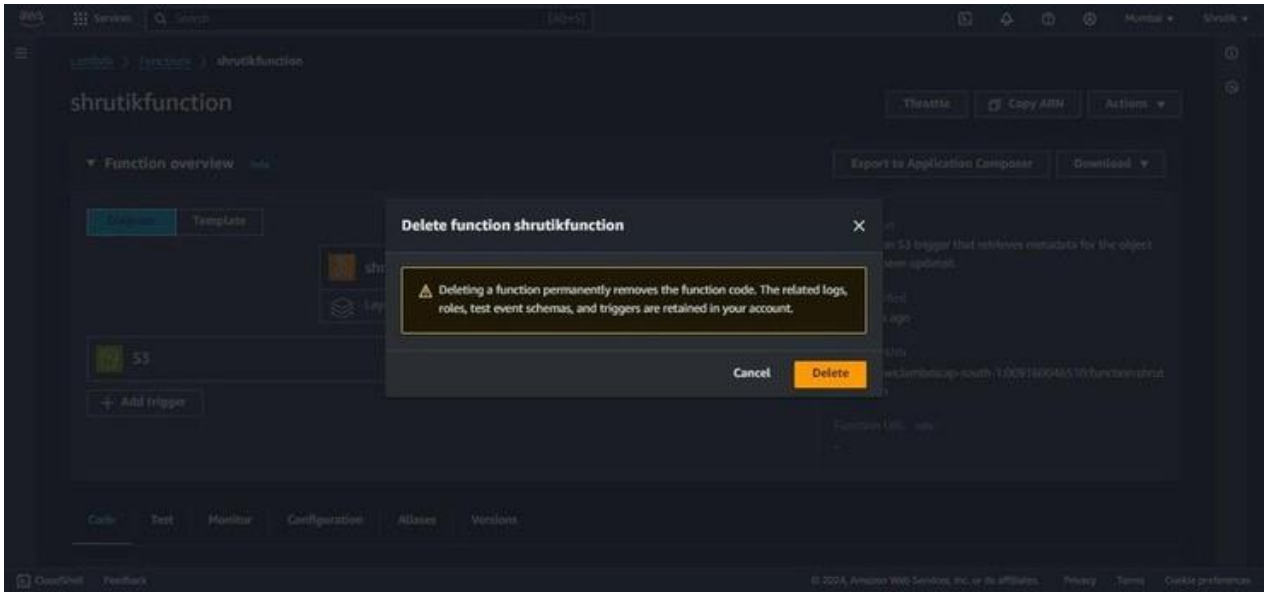
7. Check if the given trigger is created



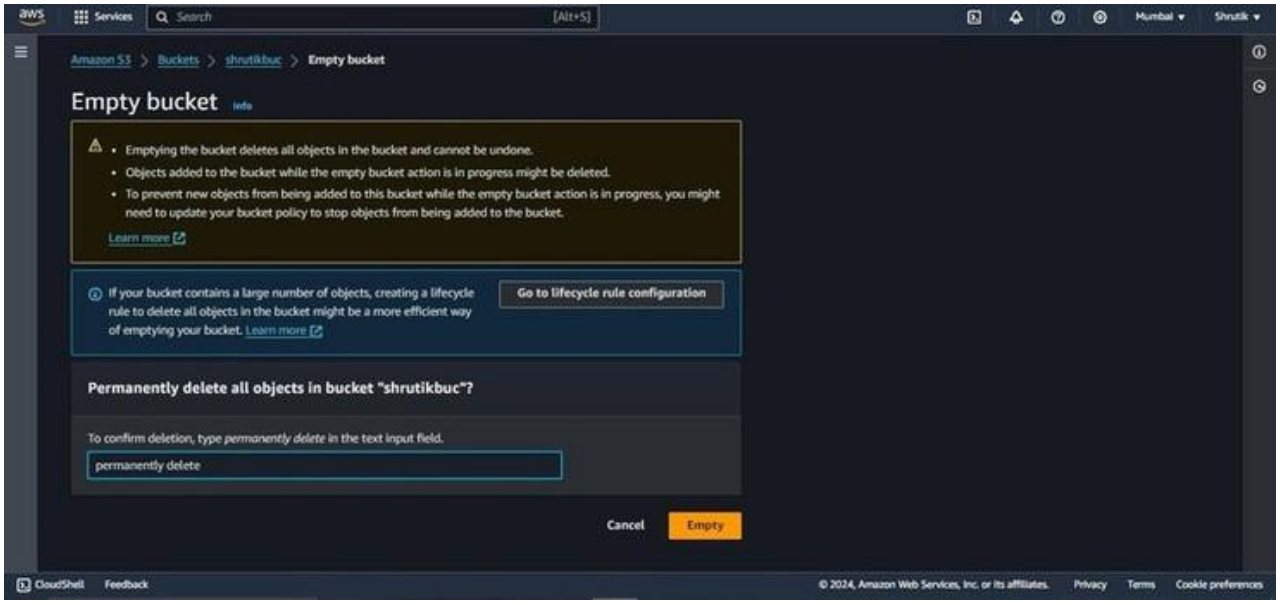
8. Click on test

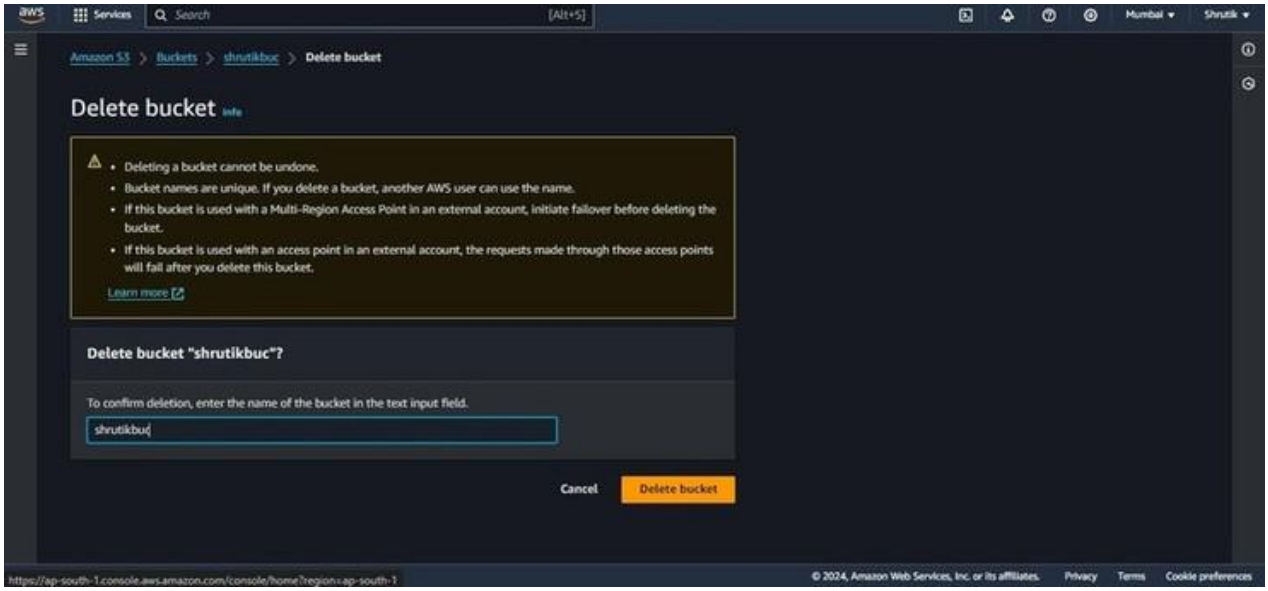


9. Delete the function



10. Empty and delete the bucket





Conclusion: Hence learned & implemented S3 buckets and lambda function

Name : Saikarthik Iyer

Batch : T13

Roll no : 41

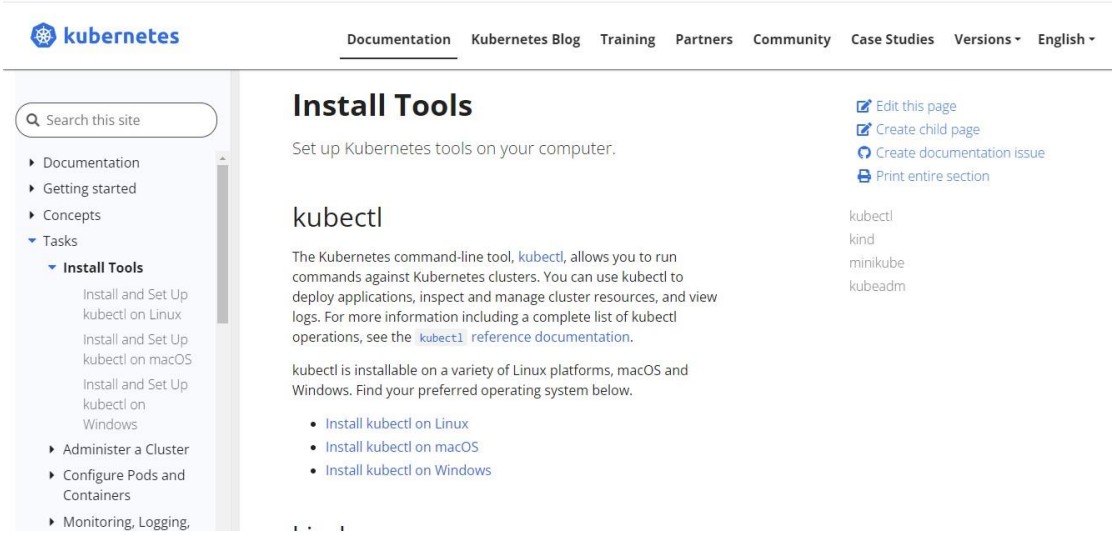
AdvDevops Assignment 3

Aim : To install and understand Kubernetes

Theory and Outcome:

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available. The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s". Google opensourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community.

Go to kubernetes website for installation.

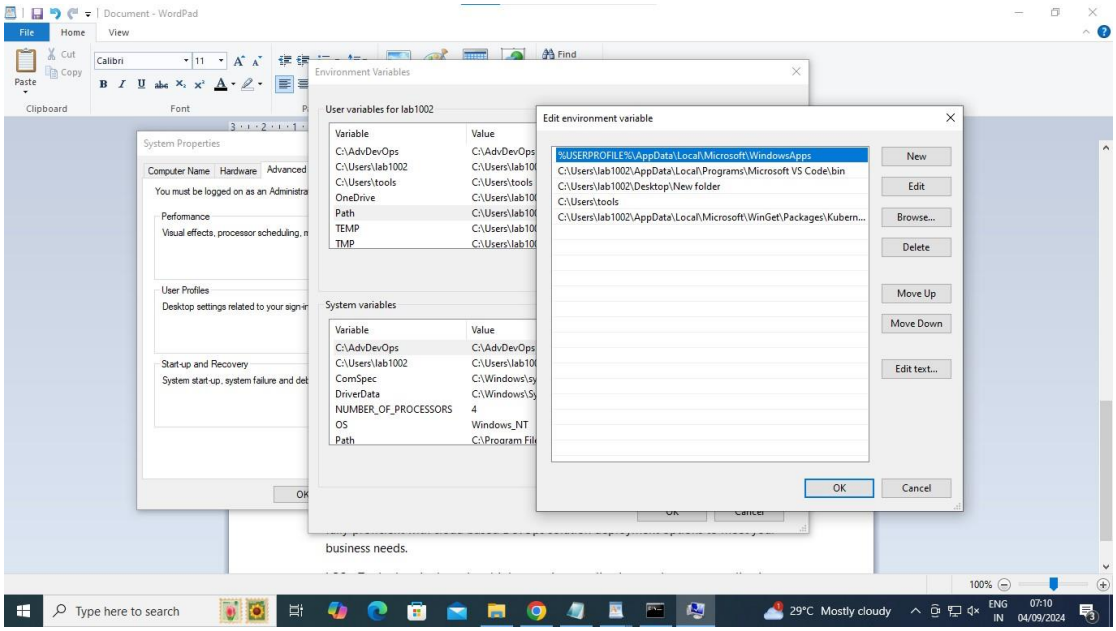


Select your appropriate OS.

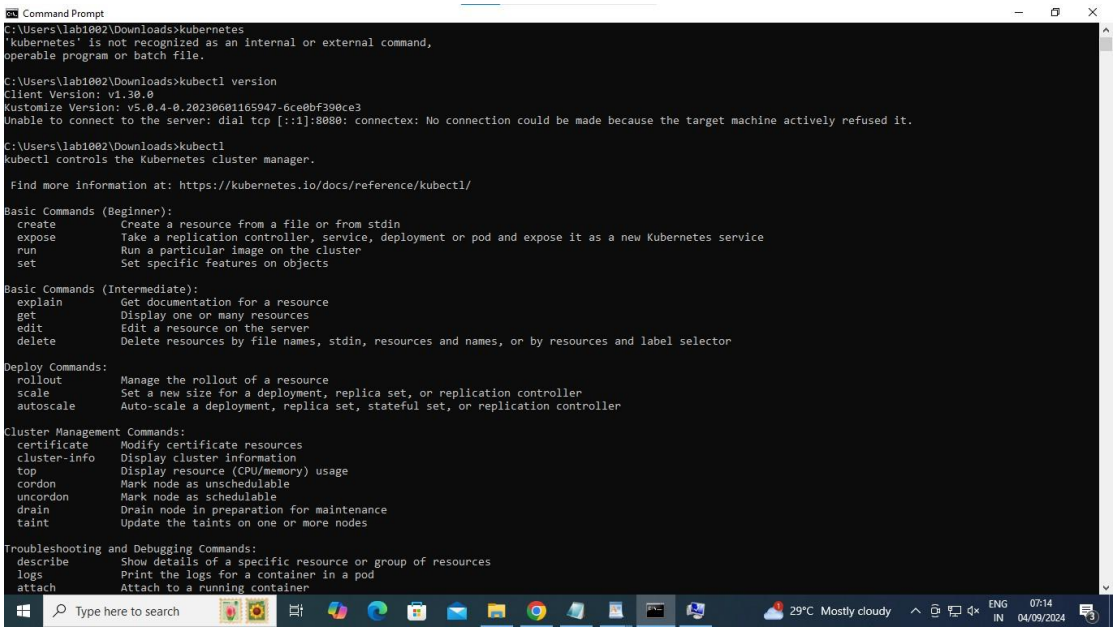
v1.31.0	windows	386	kubectl-convert.exe	dl.k8s.io/v1.31.0/bin/v convert.exe (checksum |
v1.31.0	windows	386	kubectl.exe	dl.k8s.io/v1.31.0/bin/v (checksum | signature |
v1.31.0	windows	amd64	kube-log-runner.exe	dl.k8s.io/v1.31.0/bin/v runner.exe (checksum |
v1.31.0	windows	amd64	kube-proxy.exe	dl.k8s.io/v1.31.0/bin/v proxy.exe (checksum | s
v1.31.0	windows	amd64	kubeadm.exe	dl.k8s.io/v1.31.0/bin/win (checksum | signature |

Install the Kubernetes .

Edit the path in systems environment variables



Open cmd and run and kubernetes command to check for successful installation



Kubernetes is successfully installed.

Conclusion : LO1 - To understand the fundamentals of cloud computing and be fully proficient with cloud based DevOps solution deployment options to meet your business needs. LO2 - To deploy single and multiple container applications and manage application

Saikarthik Iyer

T13

41

AIM: To understand the Kubernetes Cluster Architecture.

THEORY:

What are the various Kubernetes services running on nodes? Describe the role of each service. Kubernetes relies on various services running on cluster nodes to ensure smooth and efficient orchestration of containers. Here are some key services running on Kubernetes nodes and their roles:

1. Kubelet:

- Role: Kubelet is an essential component that runs on each node and communicates with the Kubernetes Control Plane.
- Responsibilities: It ensures that containers are running in a Pod as expected. Kubelet takes PodSpecs (defined by the higher-level orchestration system) and ensures that the containers described in those PodSpecs are running and healthy.

2. Kube Proxy (kube-proxy):

- Role: Kube Proxy is responsible for network communication within the cluster.
- Responsibilities: It maintains network rules on nodes and enables network communication to and from the Pods. Kube Proxy handles tasks such as load balancing, routing, and network security.

3. Container Runtime (e.g., Docker, container):

- Role: The container runtime is responsible for running and managing containers.
- Responsibilities: It interacts with the underlying Linux kernel to create, run, and manage containers. Common container runtimes used with Kubernetes include Docker and containers.

4. Node OS:

- Role: The host operating system on which Kubernetes nodes run.
- Responsibilities: The node OS provides the foundational environment for containers and system-level resources required by Kubernetes services.

5. CNI (Container Network Interface):

- Role: CNI is a set of specifications and libraries for network plugins in Kubernetes.
- Responsibilities: It allows different networking plugins to work with Kubernetes nodes. CNI plugins enable the

creation and management of network interfaces for Pods, facilitating network connectivity.

6. Cloud Provider Components (if applicable):

- Role: Some Kubernetes clusters run on cloud providers like AWS, GCP, or Azure.
- Responsibilities: Cloud provider components, such as AWS cloud controller manager or Azure cloud controller manager, manage interactions between the cluster and the underlying cloud infrastructure. They enable features like load balancing, storage provisioning, and auto-scaling in a cloud-specific manner.

7. Monitoring and Logging Agents (optional):

- Role: These agents collect and send monitoring and logging data to external systems.

- Responsibilities: Agents like Prometheus, Fluentd, and others gather metrics and logs from containers and Pods, aiding in monitoring, troubleshooting, and performance analysis.

8. Kubelet Garbage Collection (optional):

- Role: The kubelet performs garbage collection to remove old resources.
- Responsibilities: It helps maintain cluster cleanliness by removing unused images, volumes, and Pods to free up resources.

These services work together to ensure the proper functioning of Kubernetes nodes and the containers running on them. Each service plays a specific role in managing containers, networking, and

node-level operations within the cluster.

What is the Pod Disruption Budget (PDB)?

A **Pod Disruption Budget (PDB)** is a policy in Kubernetes that helps define the maximum allowable disruption to a set of Pods during voluntary disruptions, such as scaling down or draining nodes. It provides a way to control the impact of actions like node maintenance or scaling events on the availability of workloads running in a Kubernetes cluster.

Key characteristics and concepts of Pod Disruption Budgets include:

1. Selector: A PDB specifies a set of Pods using label selectors. These are the Pods to which the budget applies.

2. Min Available: The PDB defines a `minAvailable` parameter, indicating the minimum number of Pods that must remain available. This helps ensure a level of availability for the application during disruptions.

3. Max Unavailable: Another parameter, `maxUnavailable`, sets the maximum number of Pods that can be unavailable at any given time.

4. Scope: PDBs can be defined at various levels, such as at the Namespace level or for specific Deployments, StatefulSets, or ReplicaSets.

5. Enforcement: Kubernetes controllers, such as the Deployment controller, ensure that the specified

`minAvailable` and `maxUnavailable` values are adhered to during actions like scaling down or draining nodes. The controller won't proceed with actions that would violate the PDB.

Pod Disruption Budgets are particularly useful when you need to ensure that critical applications maintain a certain level of availability during planned maintenance activities or when scaling down resources. They help prevent the excessive disruption of Pods and maintain the reliability of the applications running in a Kubernetes cluster.

What is the role of Load Balance in Kubernetes?

In Kubernetes, a **Load Balancer** plays a crucial role in distributing network traffic across a set of Pods or Services, ensuring high availability, reliability, and efficient resource utilisation. The primary role of a Load Balancer in Kubernetes is as follows:

1. Traffic Distribution: Load Balancers evenly distribute incoming network traffic across a group of Pods or Services. This ensures that no single Pod or Service is overwhelmed with traffic, leading to better performance and fault tolerance.

2. High Availability: Load Balancers enhance the availability of applications by ensuring that traffic is directed to healthy Pods or Services. If one Pod becomes unhealthy or is taken down for maintenance, the Load Balancer automatically redirects traffic to healthy instances, minimising downtime.

3. Scalability: As the demand for a service or application grows, Load Balancers can distribute traffic to multiple Pods or replicas of a Service. This horizontal scaling ensures that the system can handle increased load effectively.

4. Service Discovery: Load Balancers often act as a single entry point for external clients to access a Kubernetes Service. This abstraction simplifies the process of discovering and accessing Services running in the cluster.

5. Network Path Optimization: Load Balancers can optimise the network path for incoming traffic. They can route

traffic to the nearest or least congested Pods or data centres, improving network performance.

6. Security: Load Balancers can be configured to provide security features, such as SSL termination and DDoS protection, to safeguard the applications running in the cluster.

7. Public-Facing Services: For applications or Services that need to be publicly accessible, Load Balancers provide a secure and controlled way to expose the Services to the internet.

Kubernetes supports different types of Load Balancers, including cloud provider-specific Load Balancers (e.g., AWS ELB, GCP Load Balancers) and Kubernetes-native Load Balancers like kube-proxy and kube-router. The choice of Load Balancer type depends on your cluster setup, requirements, and infrastructure.

In summary, Load Balancers in Kubernetes are instrumental in ensuring that network traffic is distributed efficiently and reliably across Pods and Services, enhancing application availability and performance while simplifying the process of accessing Services from external clients.

CONCLUSION:

Hence, I have understood the Kubernetes Cluster Architecture.

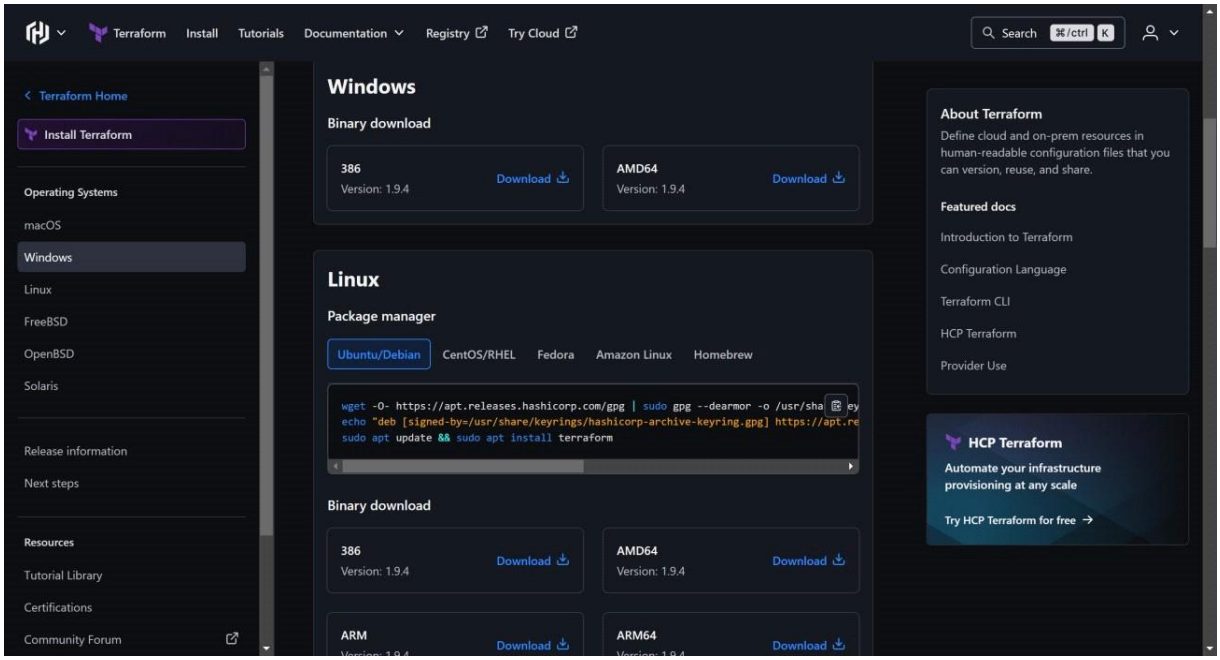
Name : Saikarthik Iyer

Roll No.: T1341

Subject : Advanced DevOps Lab

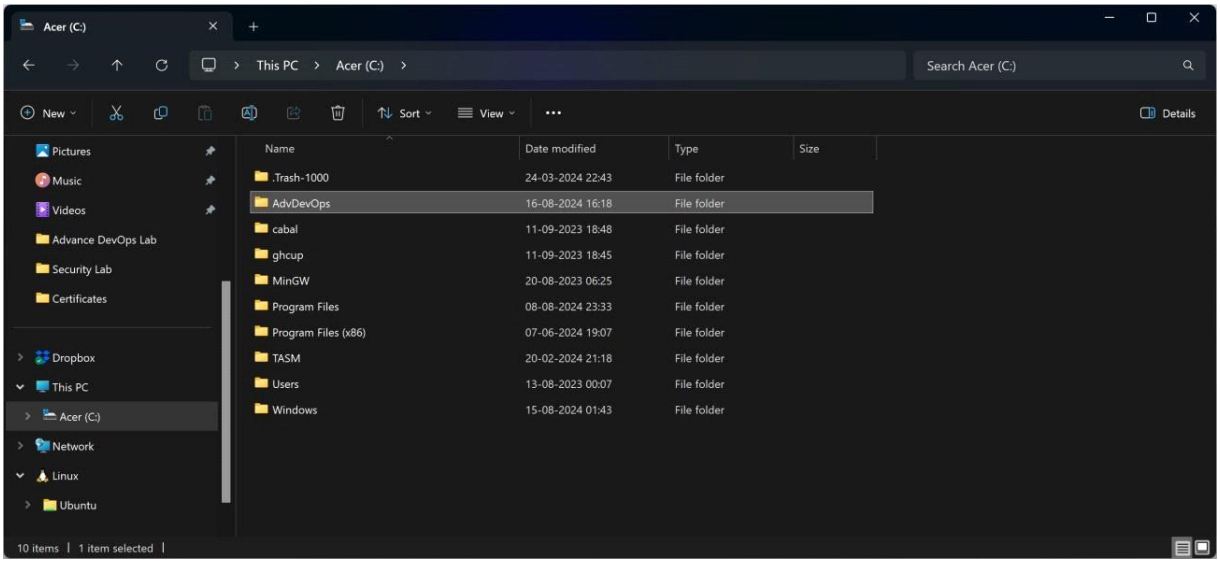
Assignment – 5

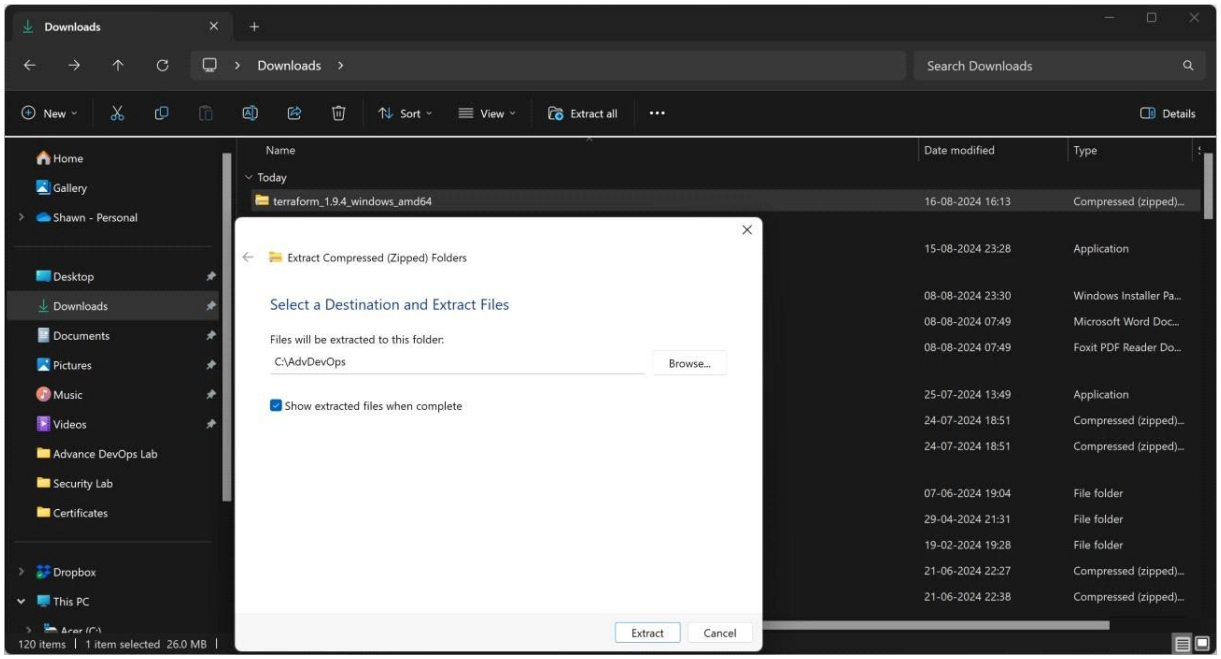
Aim : To understand the terraform lifecycle, core concepts/terminologies and install it.



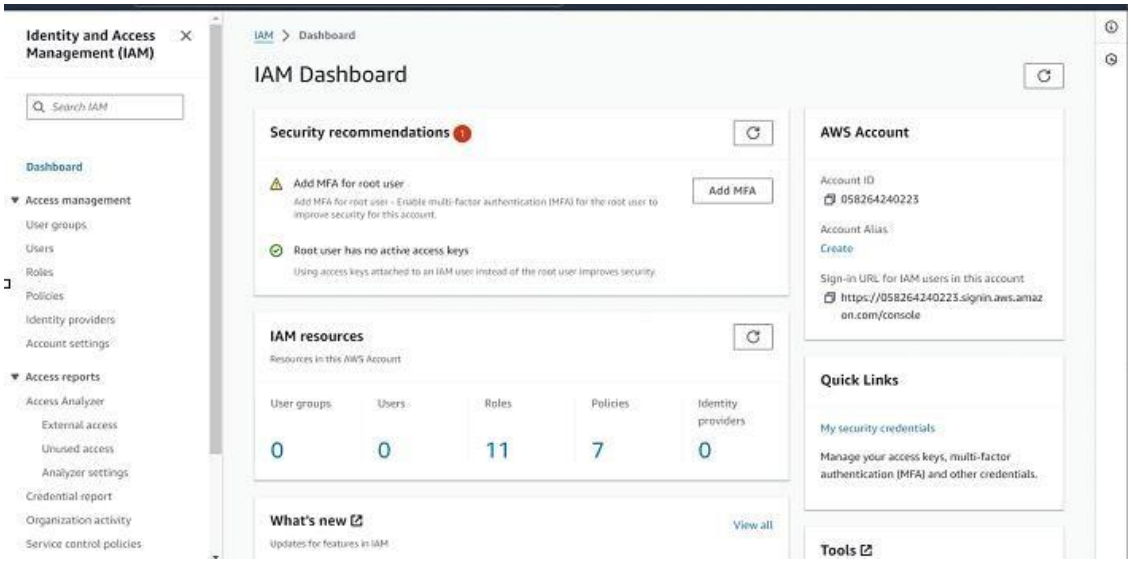
Step – 1 : Browse for installing Terraform on Windows.

Step – 2 : Create a folder name AdvDevOps in C-Drive and extract the files of the downloaded zip in that folder.

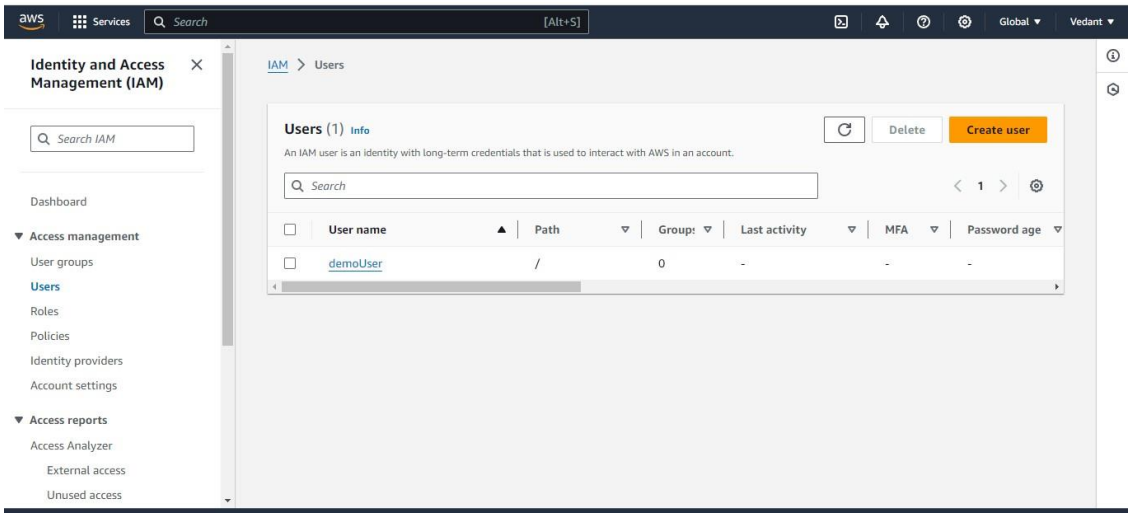


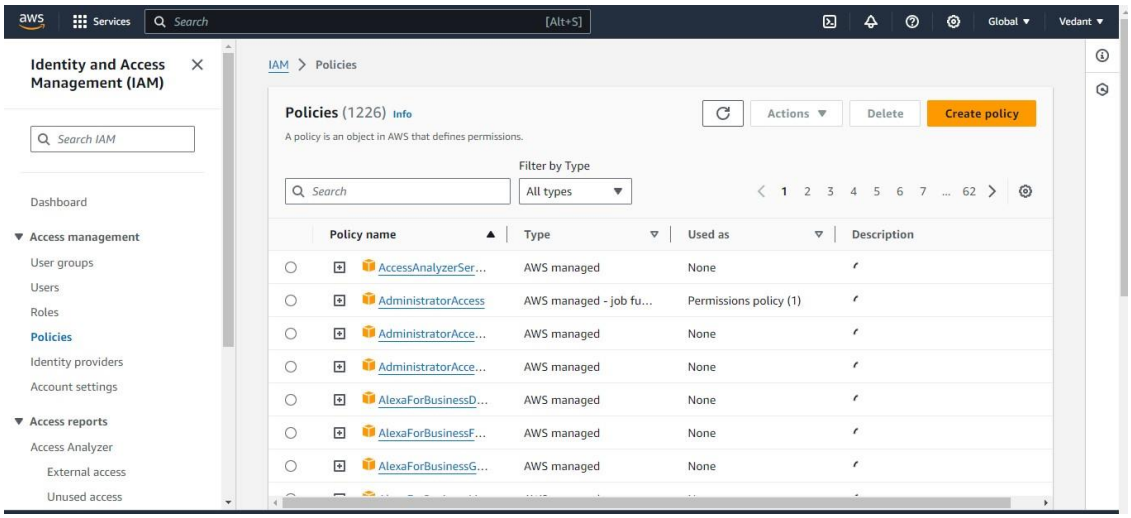


Step – 3 : Log in to AWS and search for IAM.

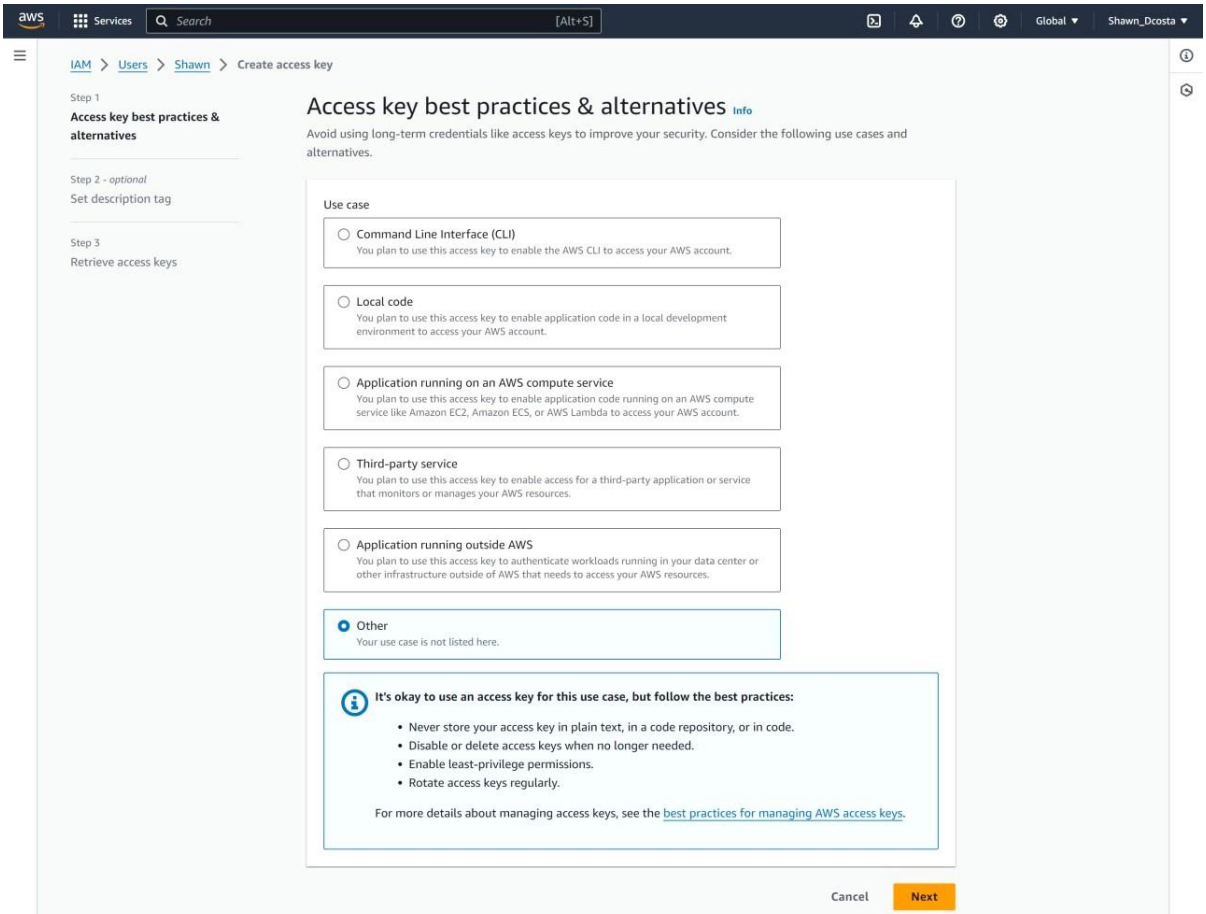


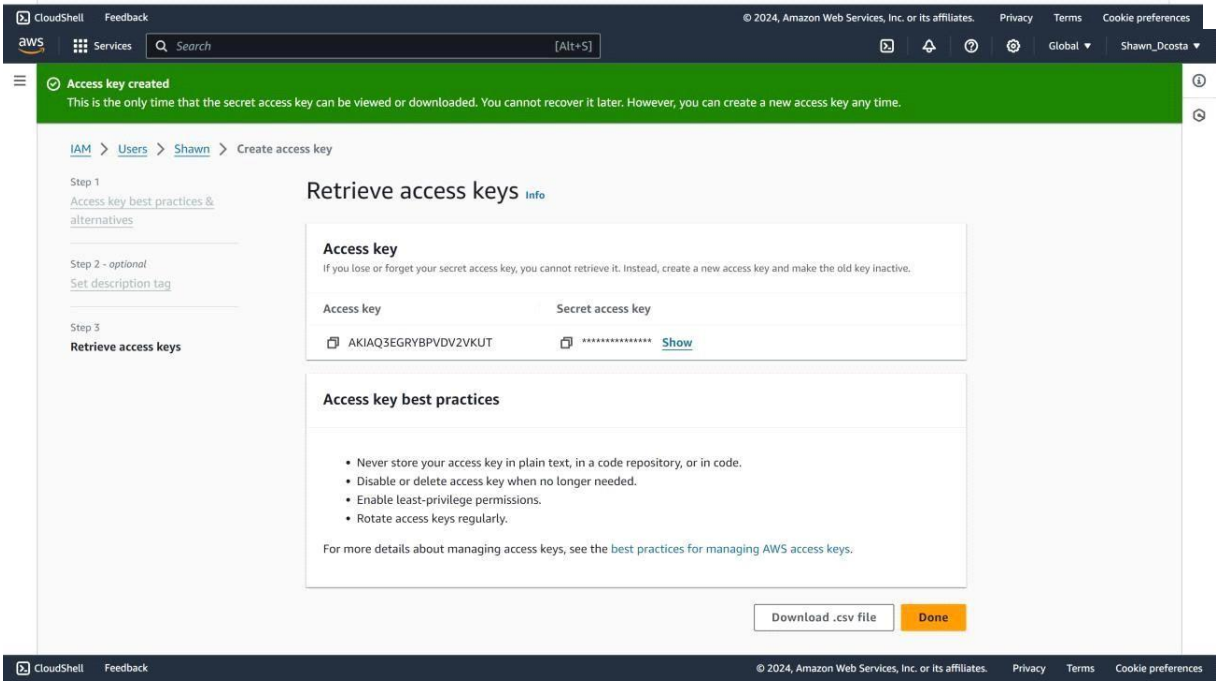
Step – 4 : Create a user and user group.


aws
Services
Search
[Alt+S]
Global
Vedant
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
IAM > Users
Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user
Search
1
User name
Path
Group:
Last activity
MFA
Password age
demoUser
/
0
-
-
-

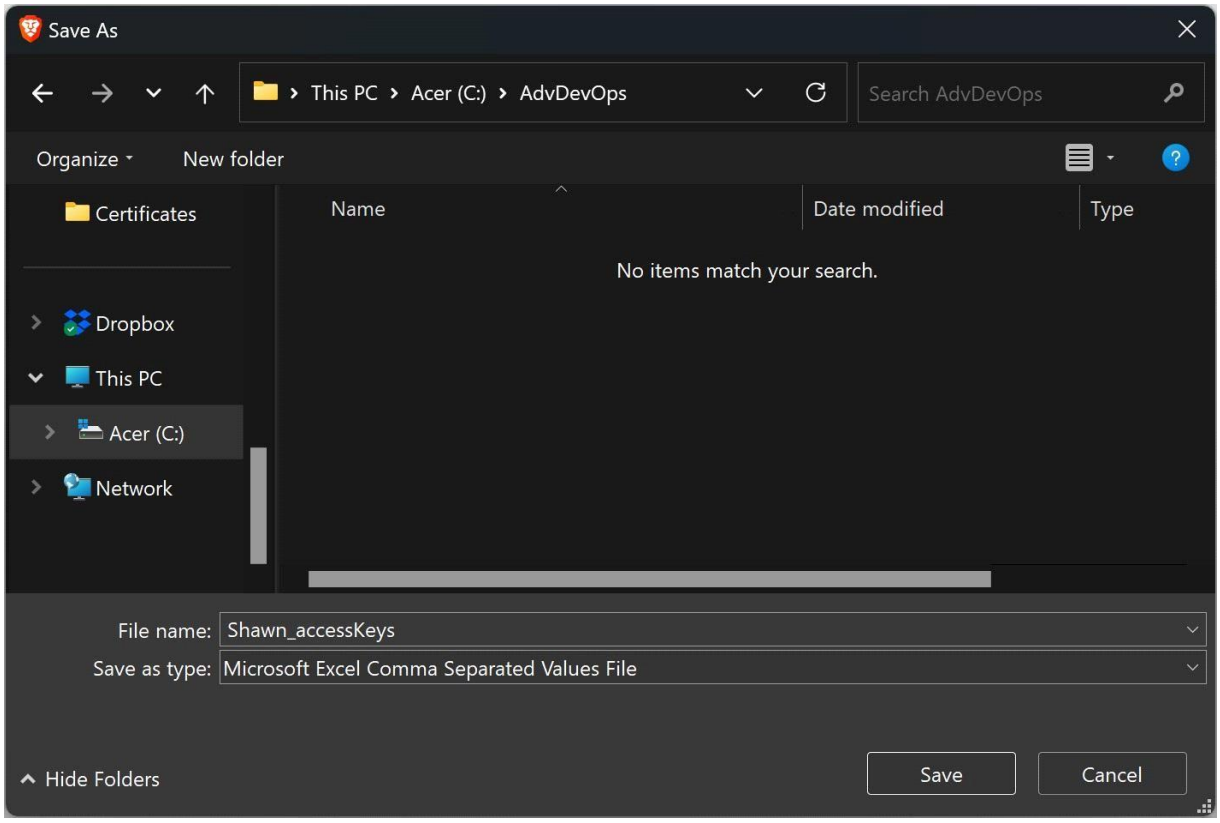

aws
Services
Search
[Alt+S]
Global
Vedant
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
IAM > Policies
Policies (1226) Info
A policy is an object in AWS that defines permissions.
Actions
Delete
Create policy
Filter by Type
Search
All types
1 2 3 4 5 6 7 ... 62
Policy name
Type
Used as
Description
AccessAnalyzerSer...
AWS managed
None
AdministratorAccess
AWS managed - job fu...
Permissions policy (1)
AdministratorAcce...
AWS managed
None
AdministratorAcce...
AWS managed
None
AlexaForBusinessD...
AWS managed
None
AlexaForBusinessF...
AWS managed
None
AlexaForBusinessG...
AWS managed
None

Step – 5 : User created successfully.

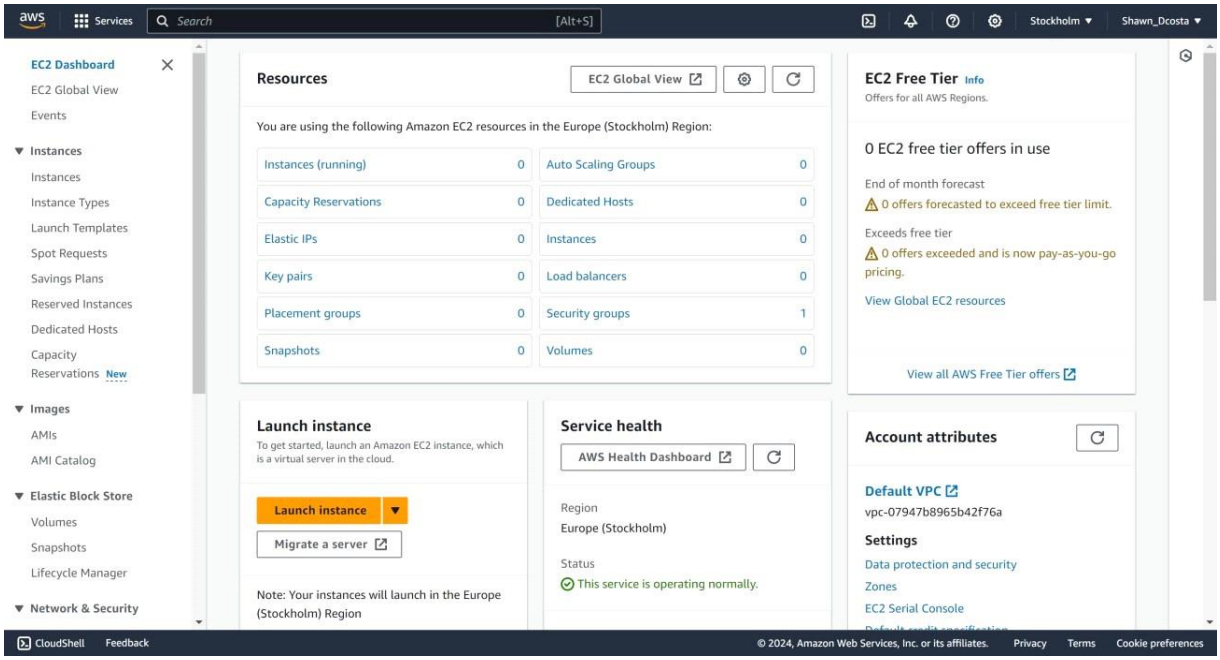




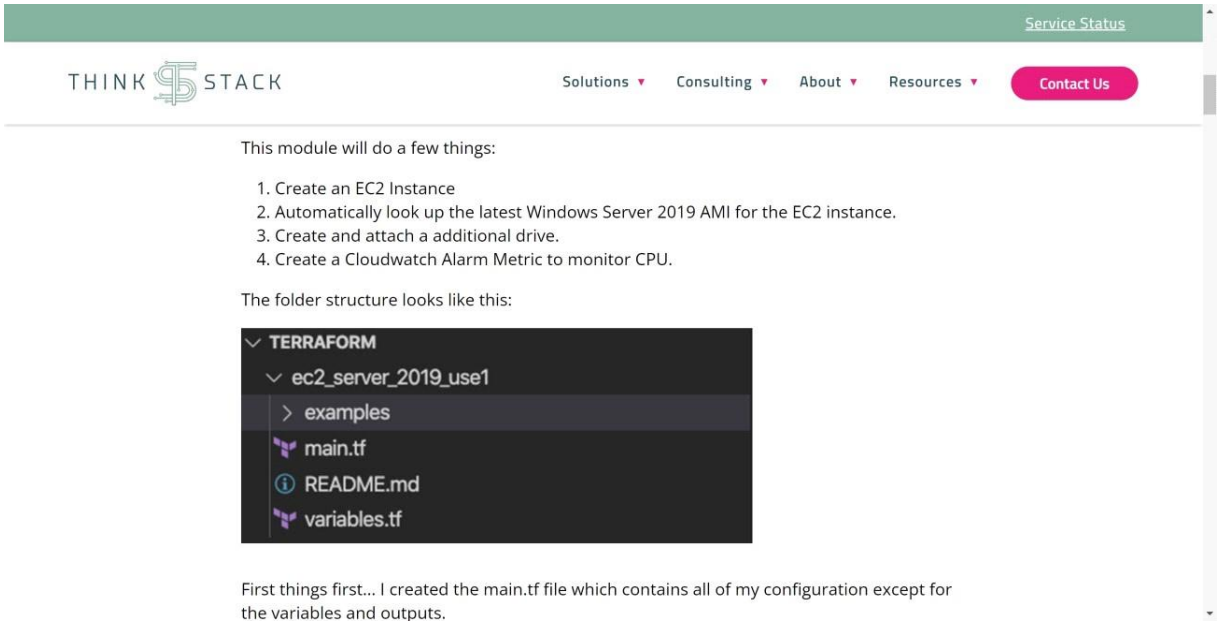
Step – 6 : Create access key.



Step – 7 : Download CSV File.

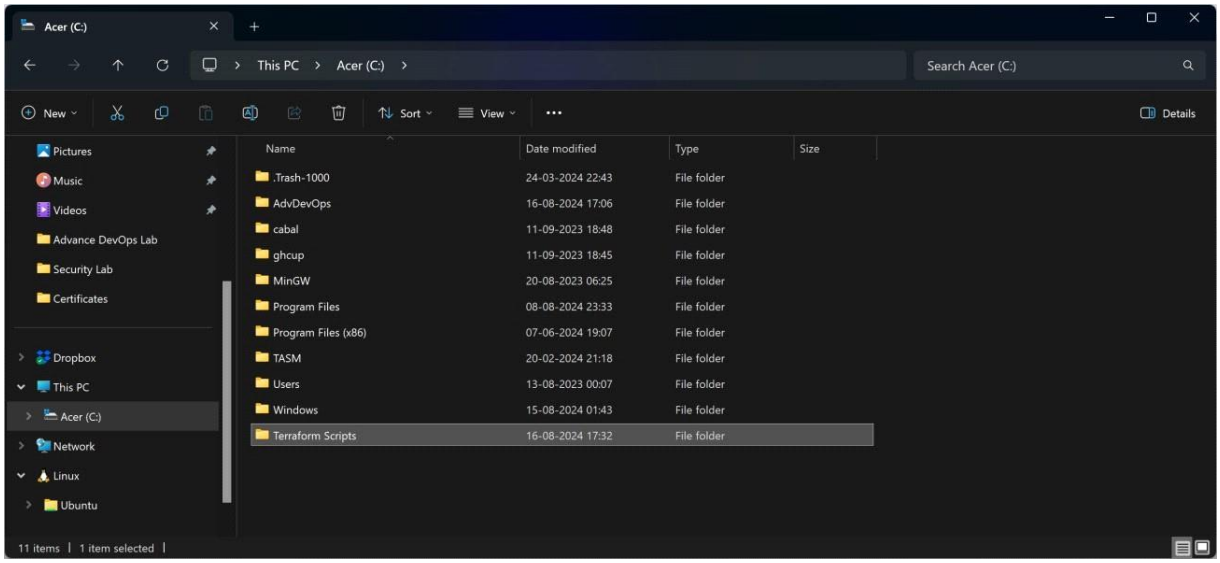


Step – 8 : Go to EC2 in AWS.



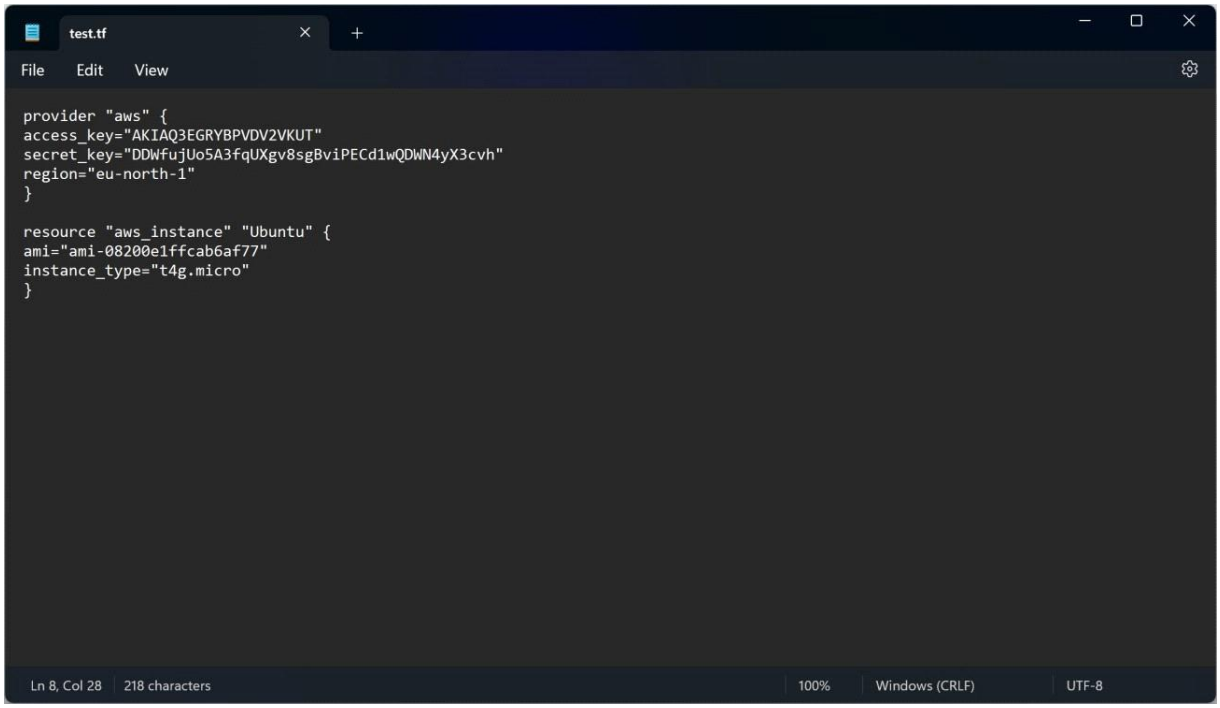
Step – 9 : Browse for Creating an EC2 instance using AWS.

Step – 10 : Create a folder named ‘Terraform Scripts’ in the C-Drive where the ‘AdvDevOps’



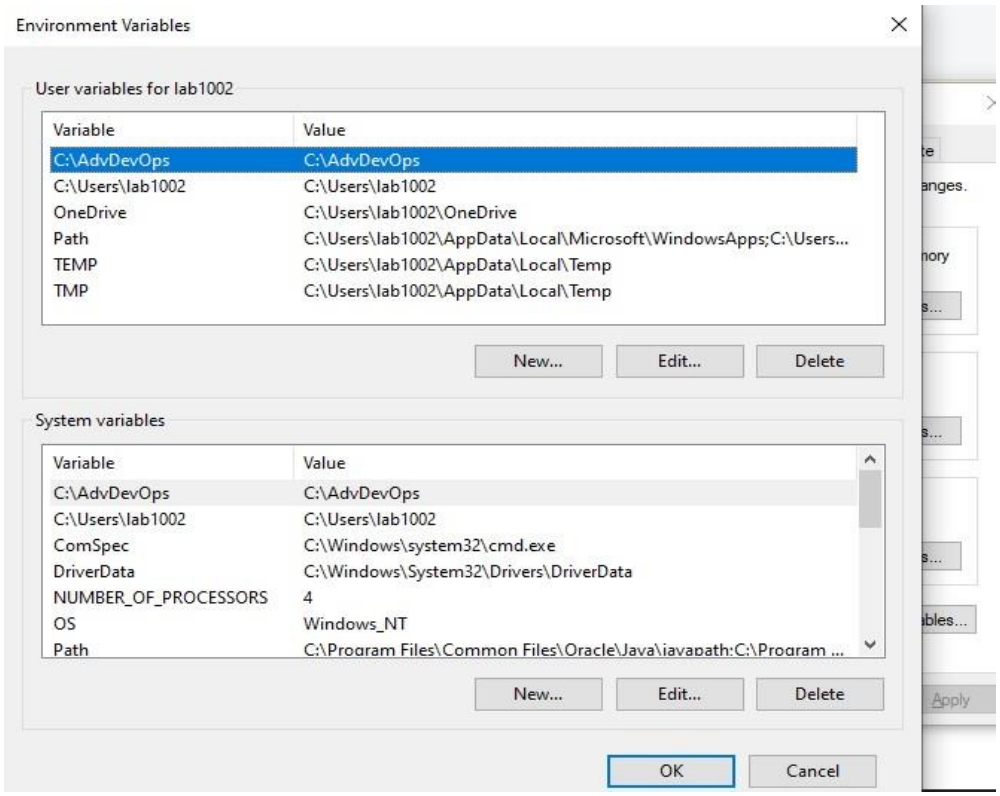
folder is located.

Step – 11 : Write the below script in Notepad and save it the recently created folder in the above step.

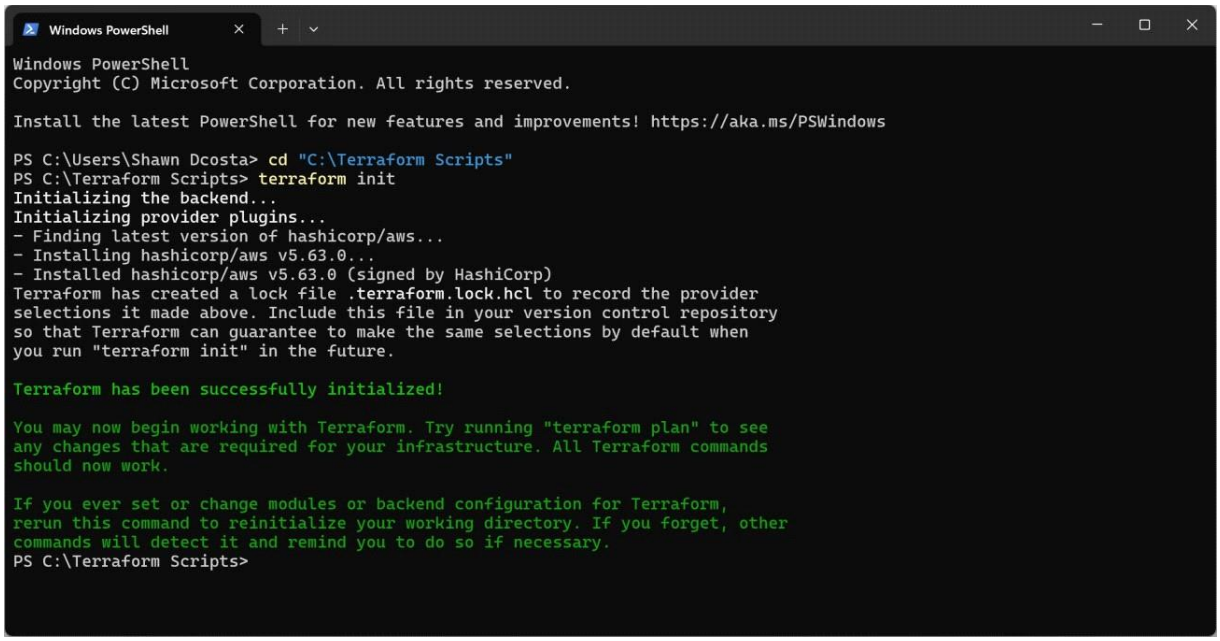


Note : a) Change Access Key and Secret Key to the one in the downloaded CSV file.

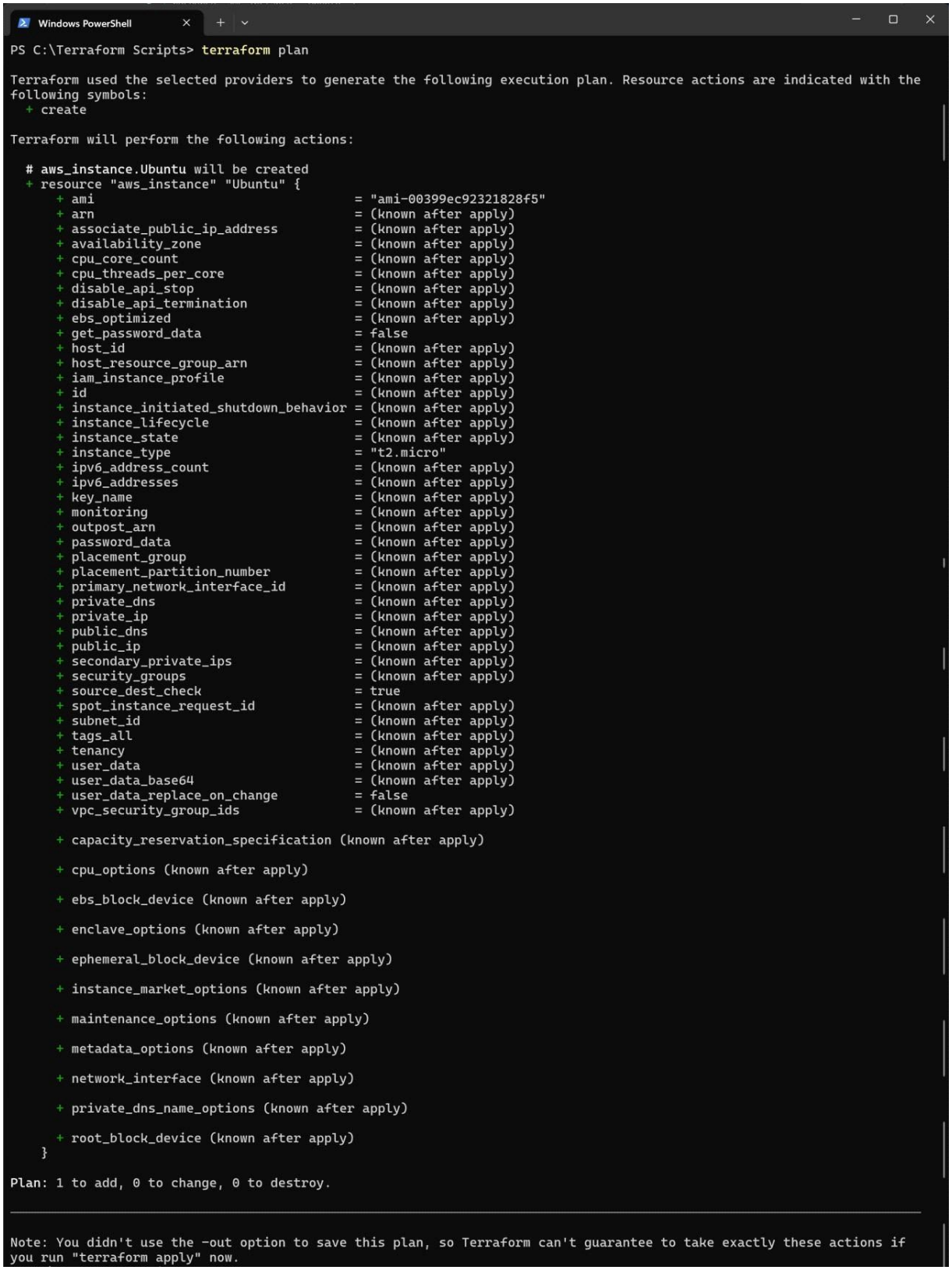
Step – 12 : Add the path of 'AdvDevOps' folder to the PATH in system variables and user variables.

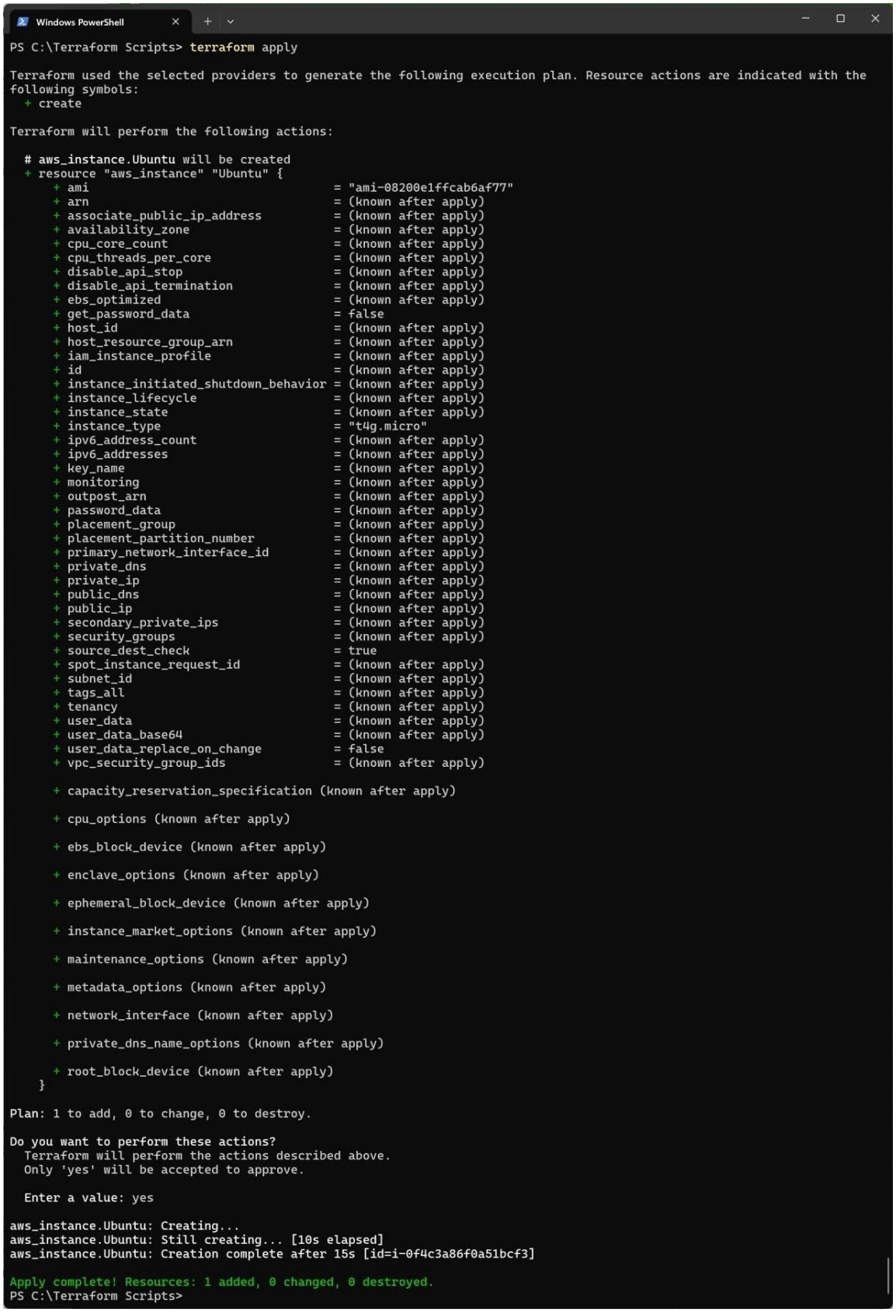


Step – 13 : Run the following command on Command Prompt/PowerShell.

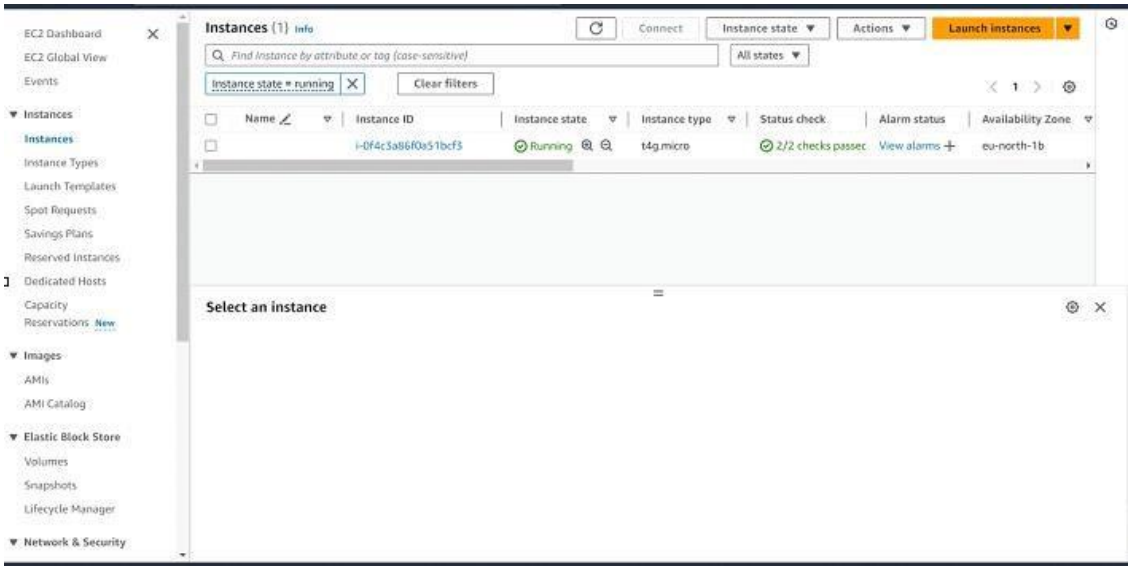


Note: Make sure you enter the right name of the folder while specifying the path.

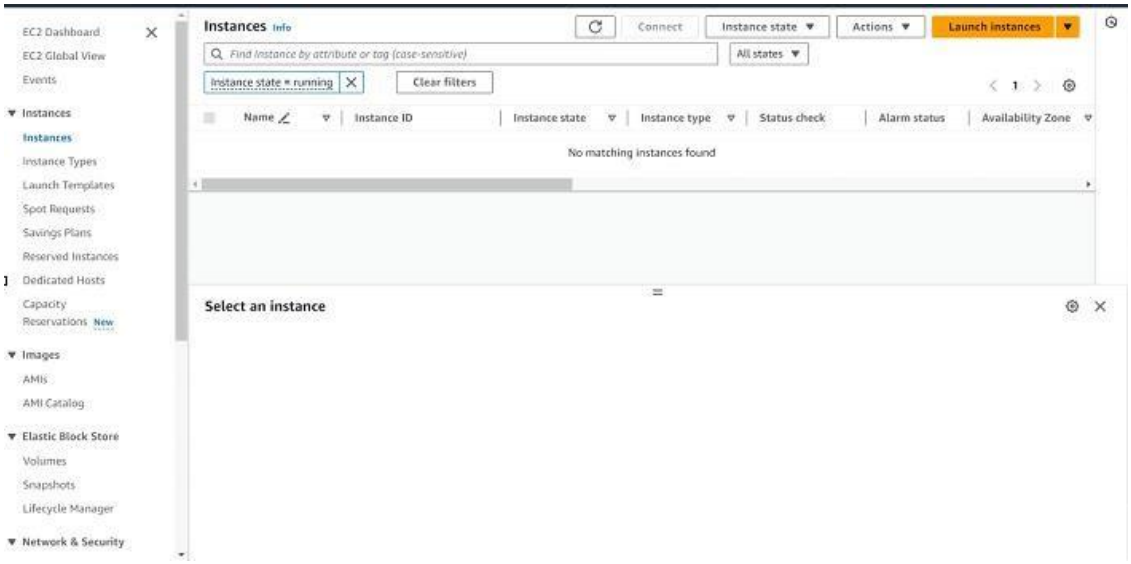

Windows PowerShell
PS C:\Terraform Scripts> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.Ubuntu will be created
+ resource "aws_instance" "Ubuntu" {
+ ami = "ami-00399ec92321828f5"
+ 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 = "t2.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)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
Plan: 1 to add, 0 to change, 0 to destroy.
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.



Step – 14 : Check if the instance has been created and is in the 'Running' state in the EC2 Dashboard.


EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
i-0f4c3a86f0a51bcf3
Running
t4g.micro
2/2 checks passed
View alarms
eu-north-1b
Select an instance

Step – 15 : Terminate the instance by using the following command and check it in EC2 Dashboard.



Conclusion : Successfully installed Terraform and ran an EC2 instance using it on Windows.

Name: Saikarthik Iyer

Roll no : 41

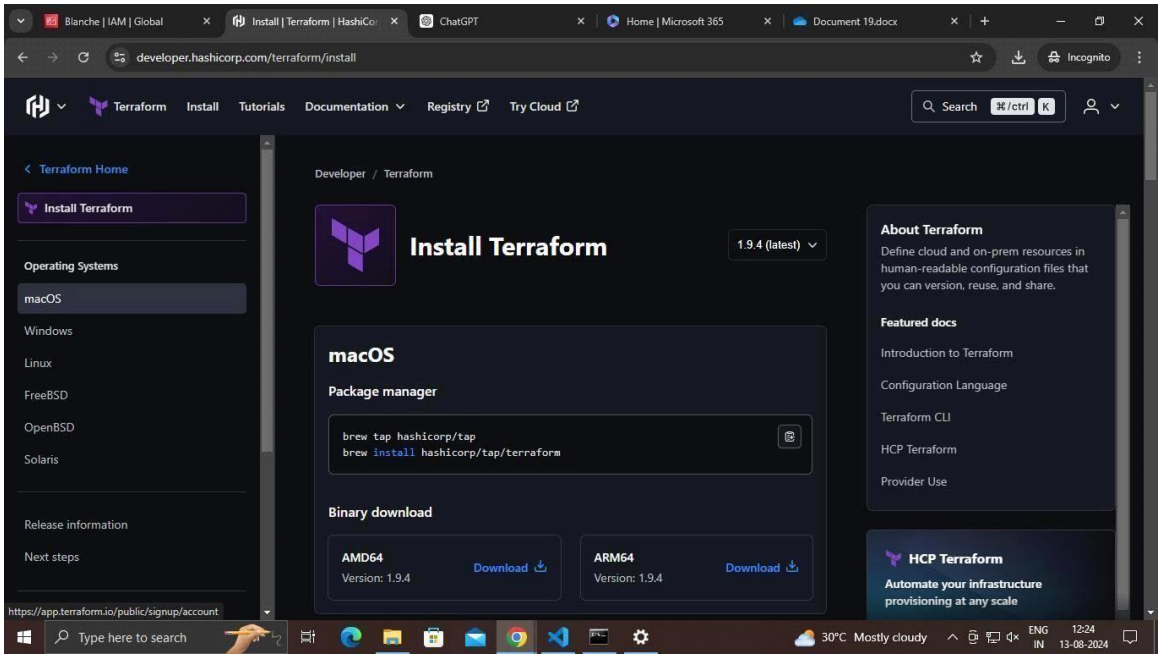
Batch : T13

Assignment 5 & 6: Terraform

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, andversion infrastructure safely and efficiently.

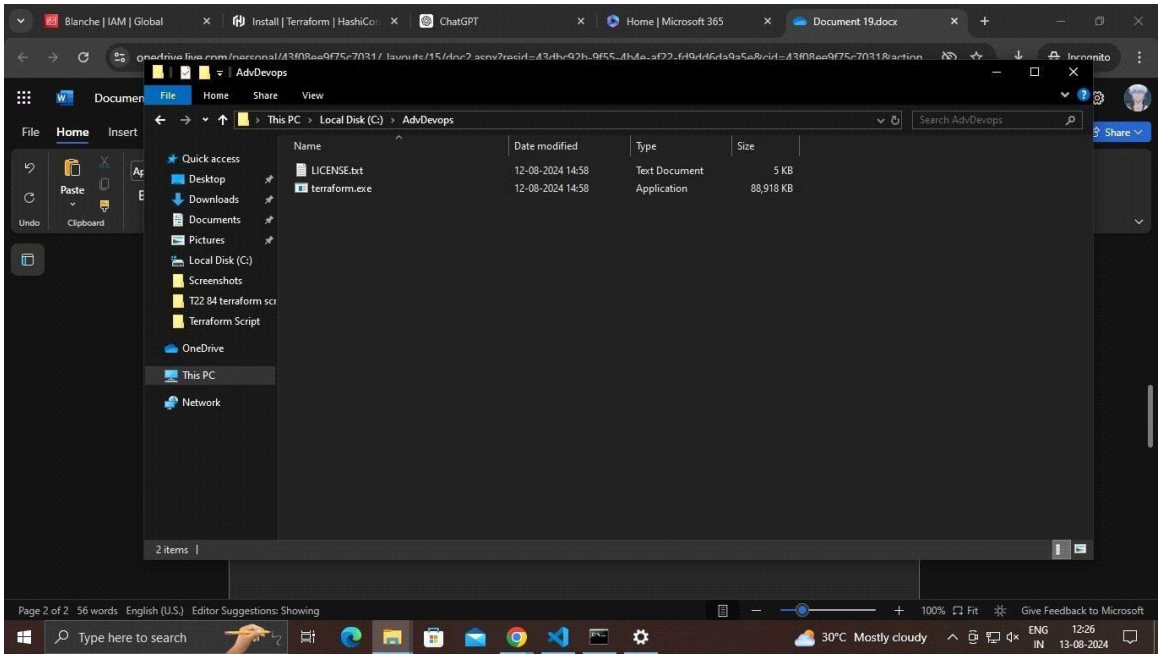
STEPS For Installation-

Google Search – Terraform DOWNLOAD



Create a folder in your c drive named advdevops

Then extract the downloaded file in the c drive flder named advdevops

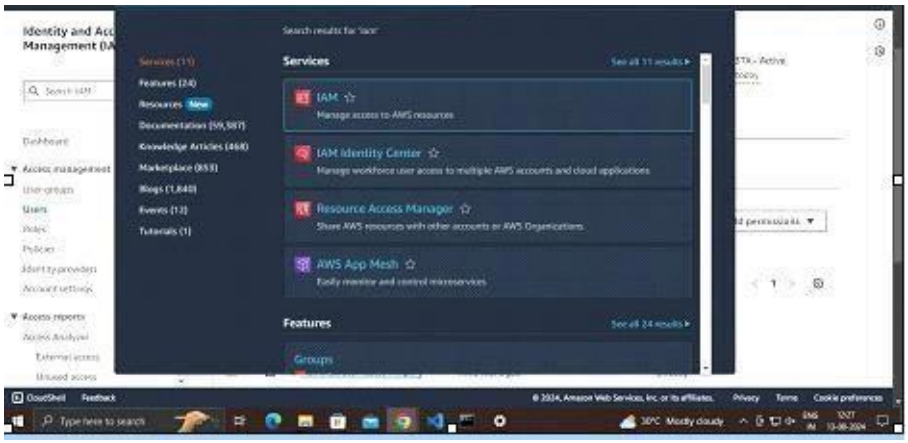


Till that time open

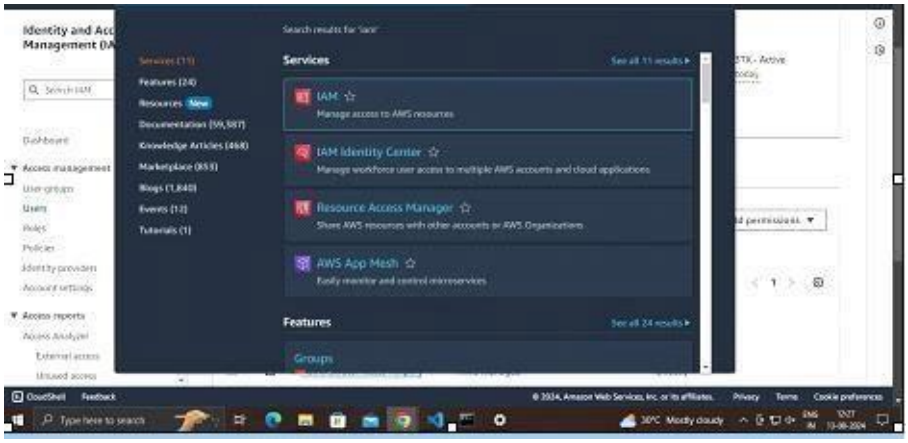
and login aws

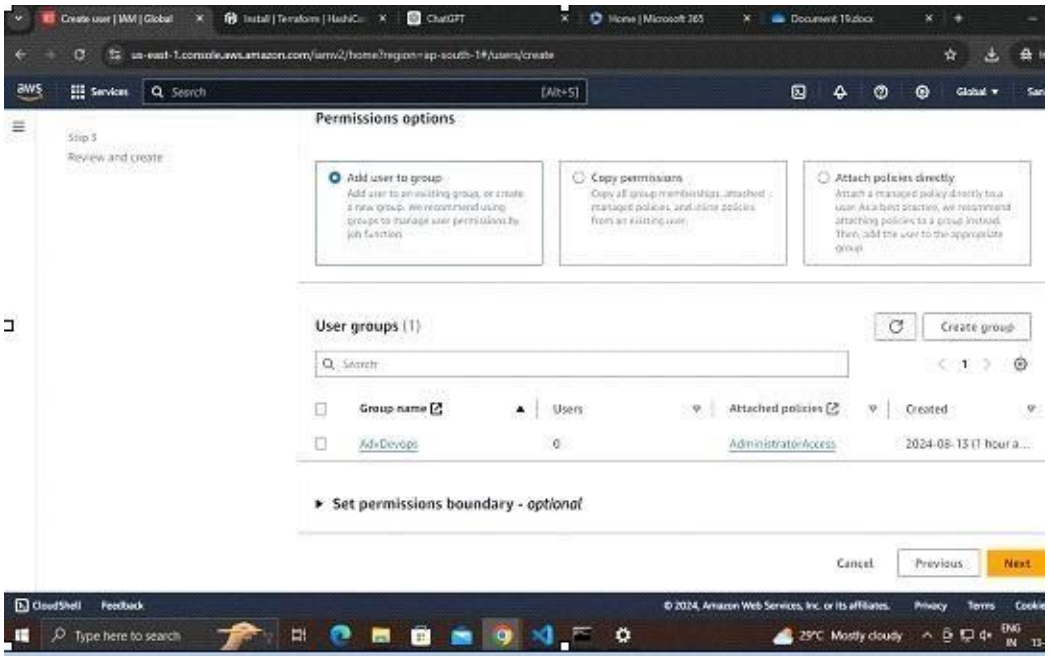
consoleSearch IAM

and click on it



Click on add users in the user section





Give

user

nam

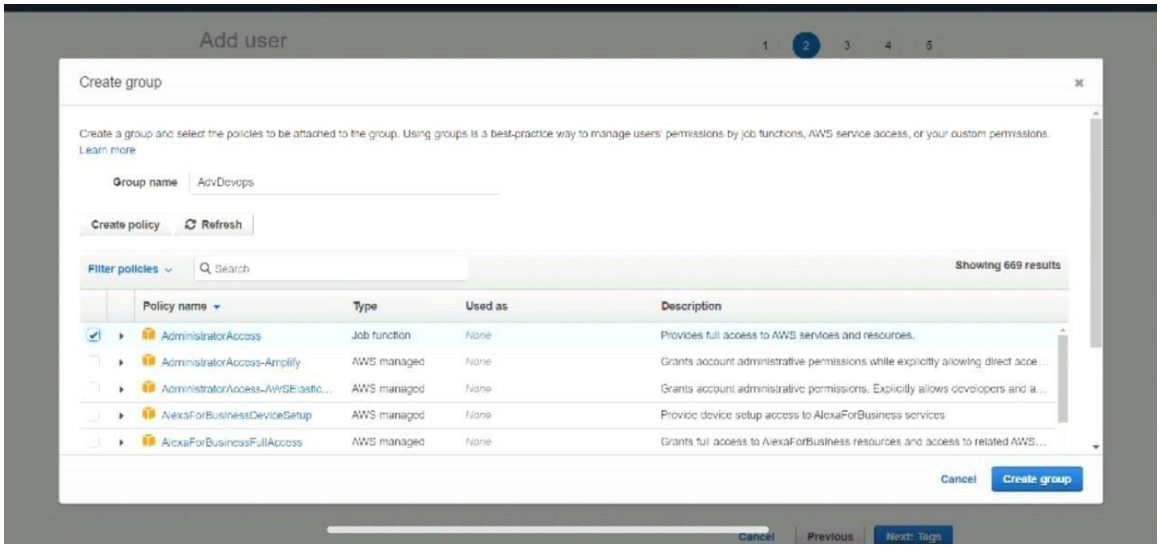
e

Cre
ate

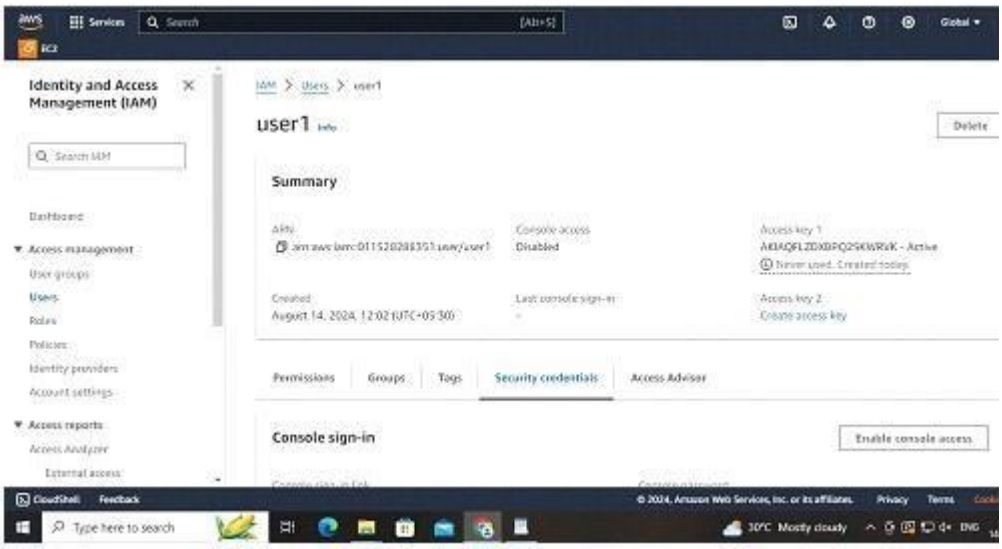
grou

p

Select policies

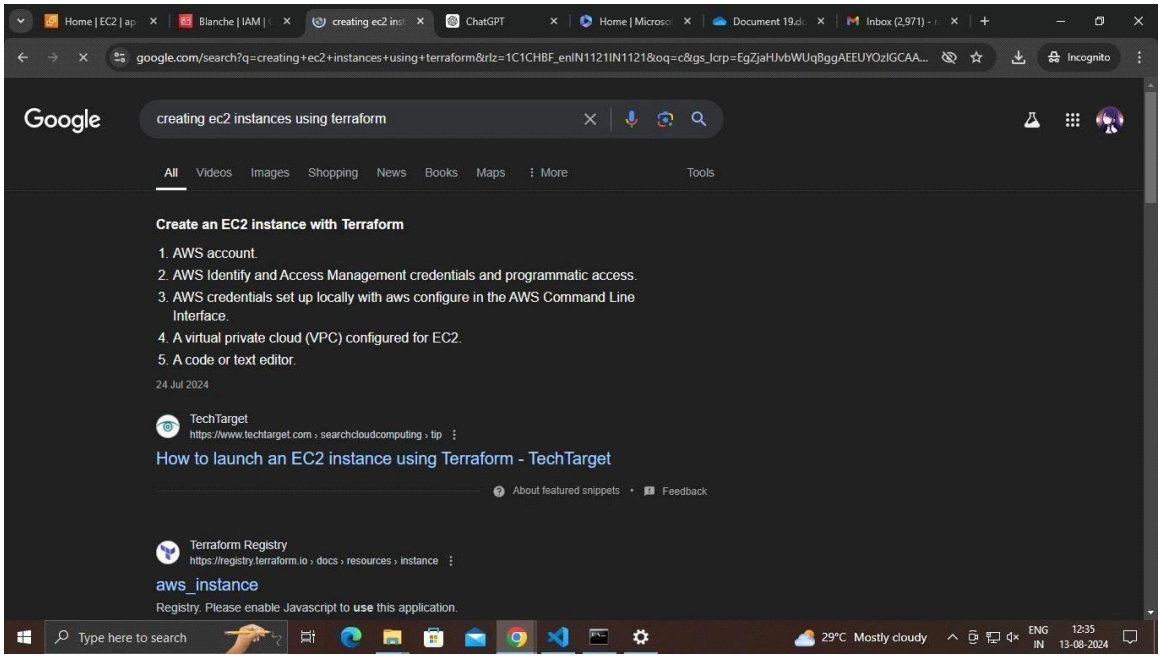


Download .csv file and get acess key



Go to ec2

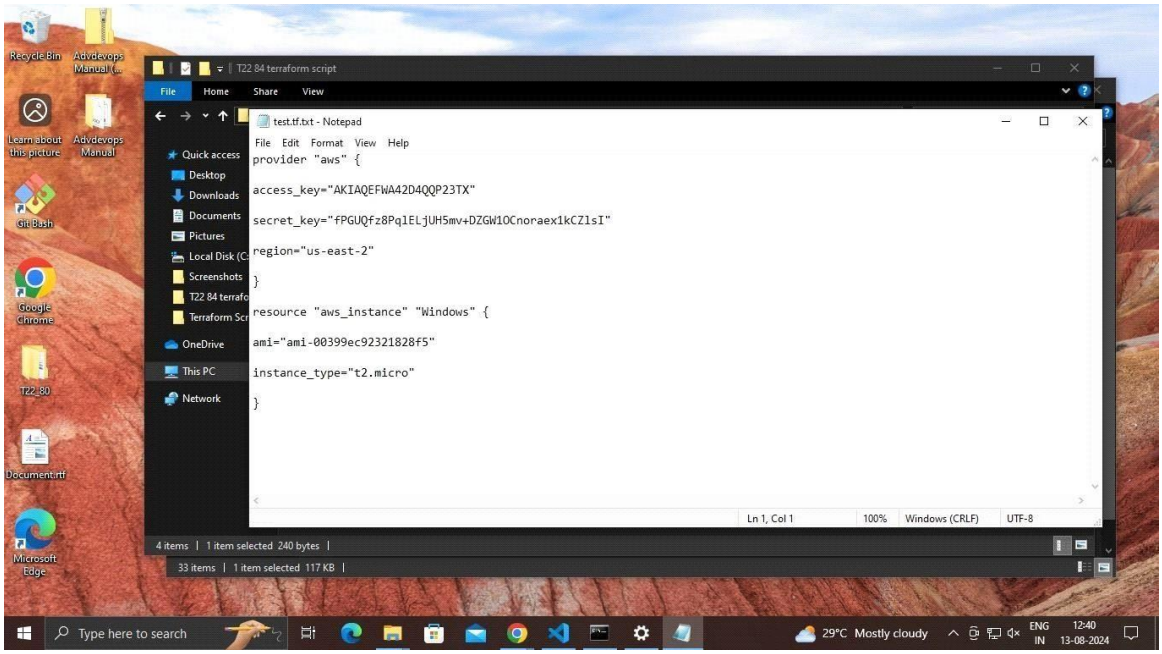
Search the following on google



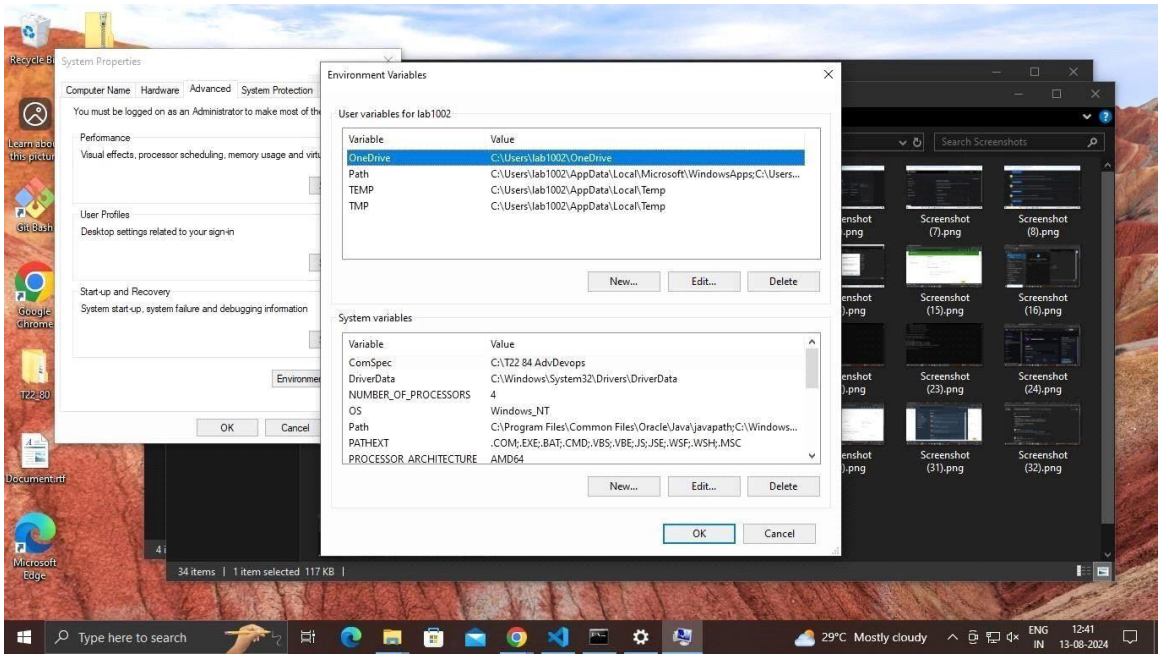
Create a folder names terraform script

go to notepad and type the details properly change acess key and secret key in .csv file

Set region same as below



Now search for system environment variables and and select the path of both as the folderof advdevops



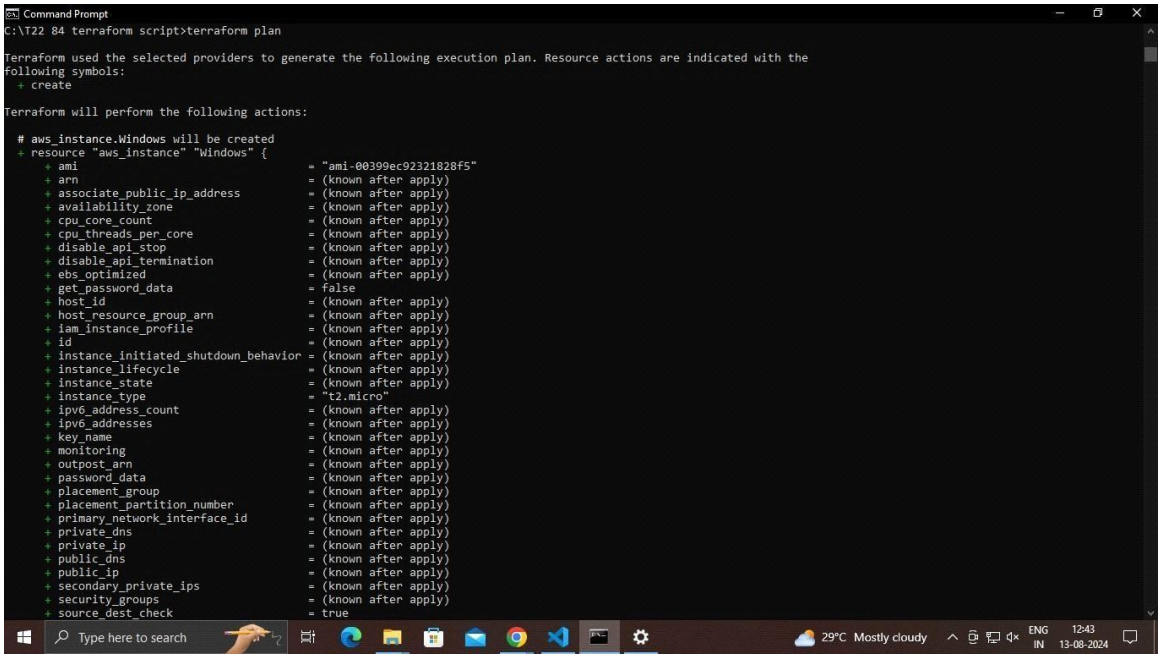
Open command prompt and paste the path of terraform script

Eg cd Terraform Script as shown below

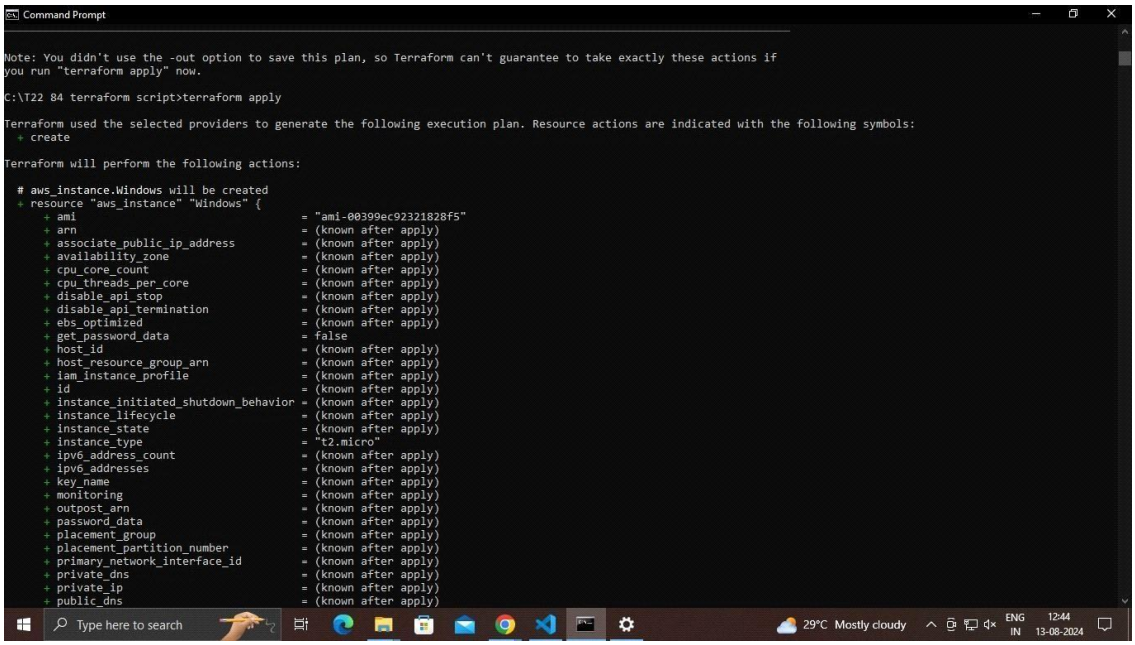
Now type terraform init command

```
C:\T22 84 terraform script>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.62.0...
- Installed hashicorp/aws v5.62.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

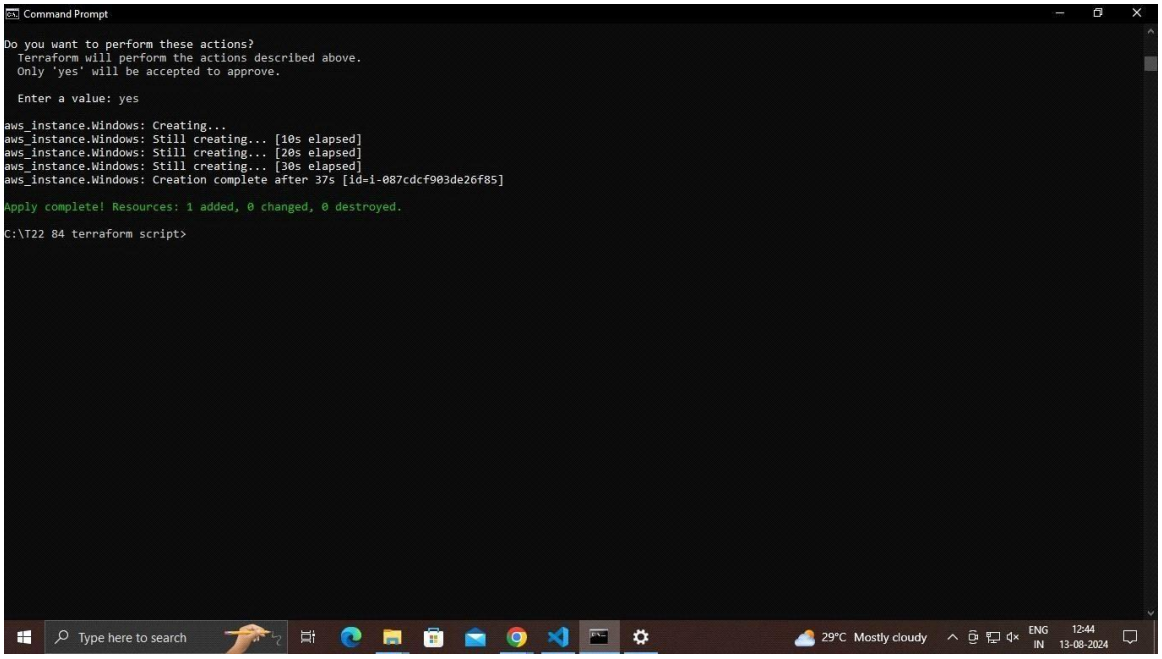
Now terraform plan



Now terraform apply

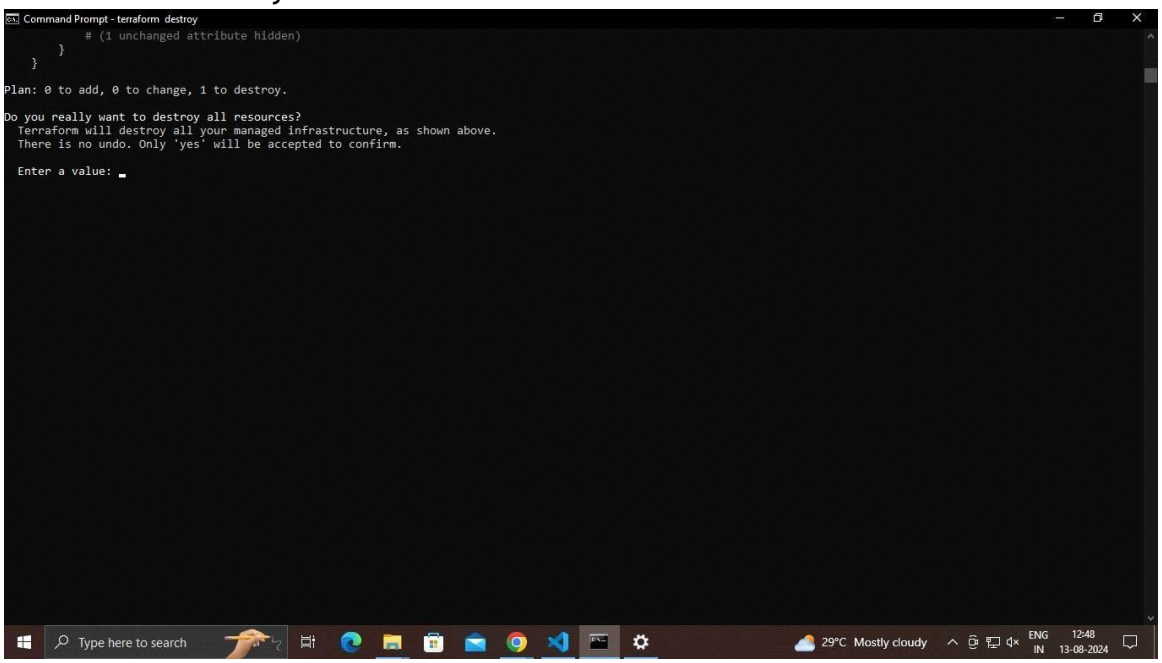


Type yes



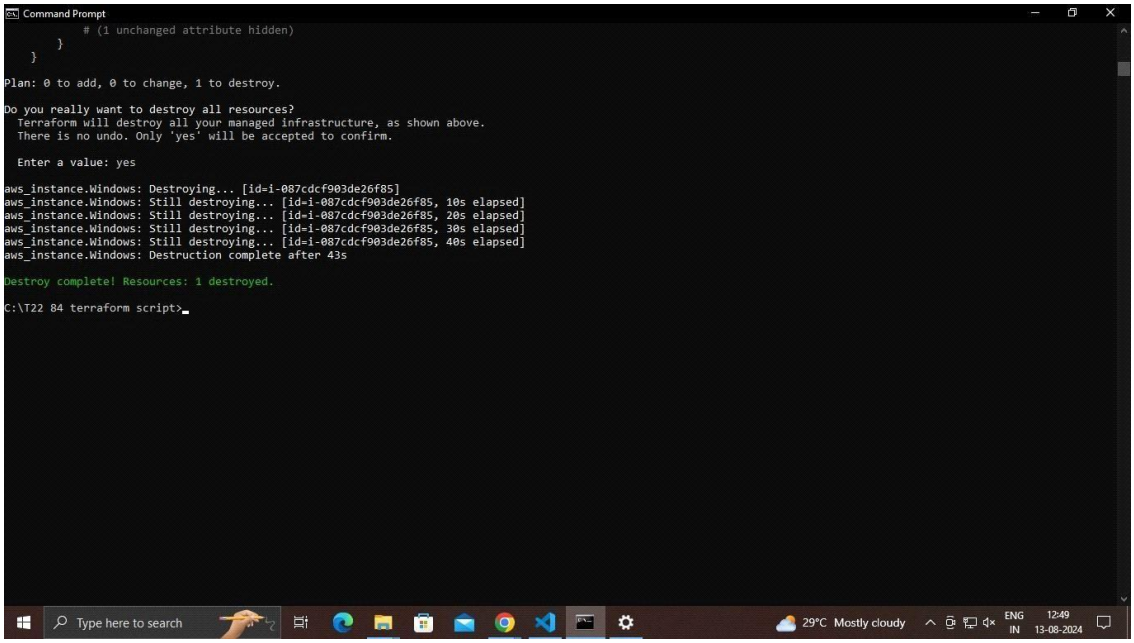
Now go to ec2 and check thaT THERE IS AN INSTANCE created by the name of WINDOWSand is in running status and then come back to the command prompt and terminate the instance by typing

Terraform destroy



Type yes

The resources will be destroyed



Now go back to EC2 check if the instance is terminated , if yes then logout of the awsconsole

And close the command prompt

Conclusion: Terraform performed and LO3 achieved

Saikarthik Iyer

T13

41

ASSIGNMENT 8

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 an S3 Bucket.

THEORY:

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It allows you to run code in response to events without having to provision or manage servers. AWS Lambda functions are small, self-contained units of code that can be executed in the cloud, triggered by various AWS services or external events.

Key features of AWS Lambda functions include:

1. Event-Driven: Lambda functions are designed to be event-driven. They can be triggered by various events such as changes to data in Amazon S3, updates to a DynamoDB table, HTTP requests through Amazon API Gateway, and more.

2. Scaling: AWS Lambda automatically scales your functions in response to incoming traffic. You don't need to worry about server provisioning or capacity planning.

3. Pay-as-You-Go: With Lambda, you pay only for the compute time consumed during function execution. There are no upfront costs or infrastructure management fees.

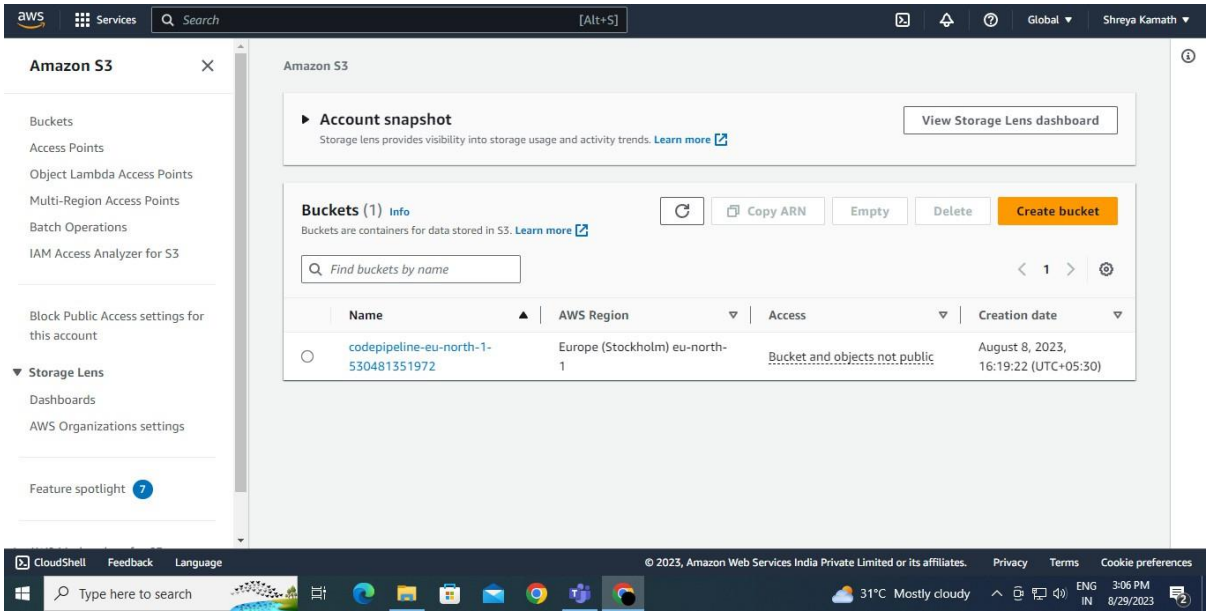
4. Support for Multiple Languages: You can write Lambda functions in various programming languages, including Node.js, Python, Java, C#, Ruby, Go, and custom runtimes.

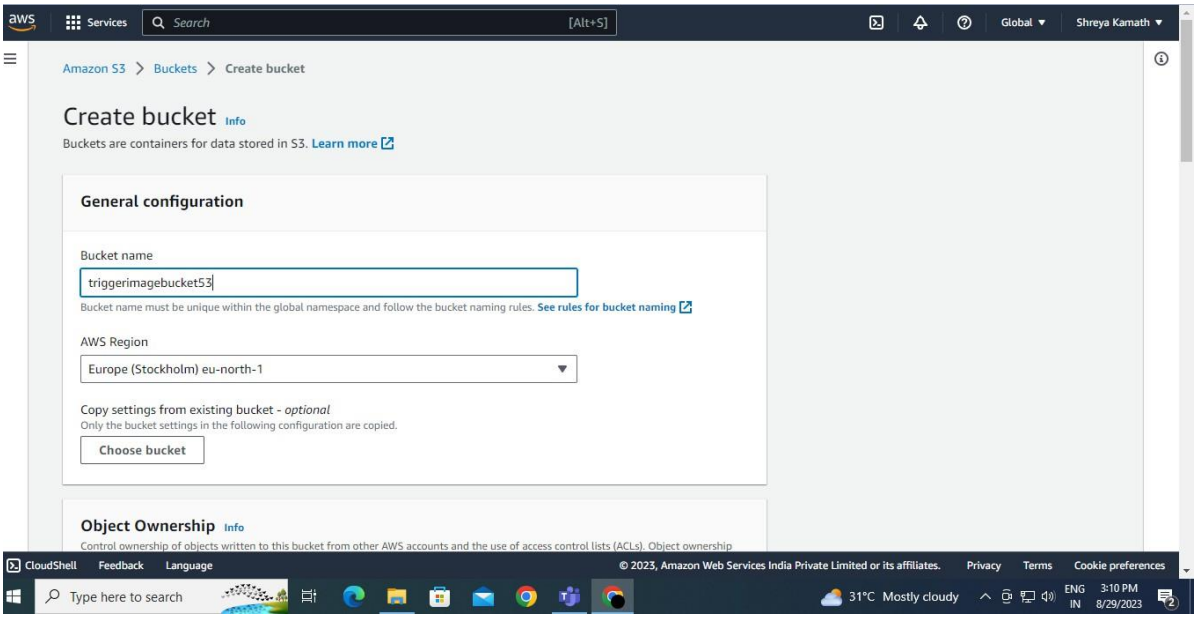
5. Stateless and Isolated: Each Lambda function is stateless and runs in isolation, ensuring that one function's execution doesn't affect another. Data can be stored and retrieved using other AWS services like S3 or DynamoDB.

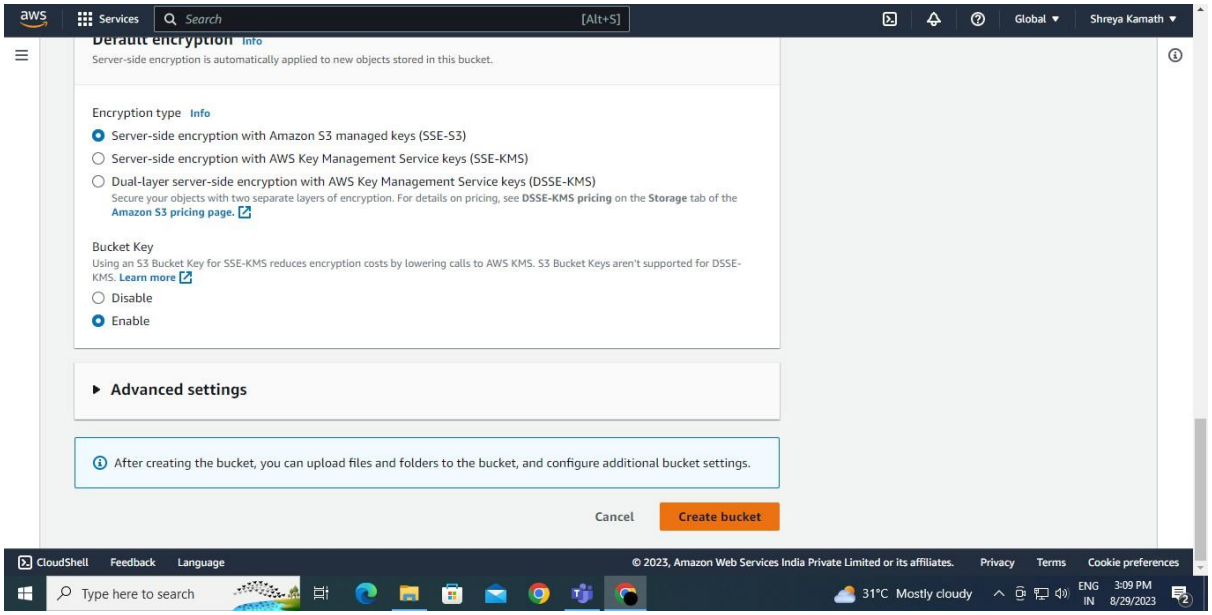
6. Easy Integration: Lambda functions can be easily integrated with other AWS services, creating serverless architectures and event-driven workflows.

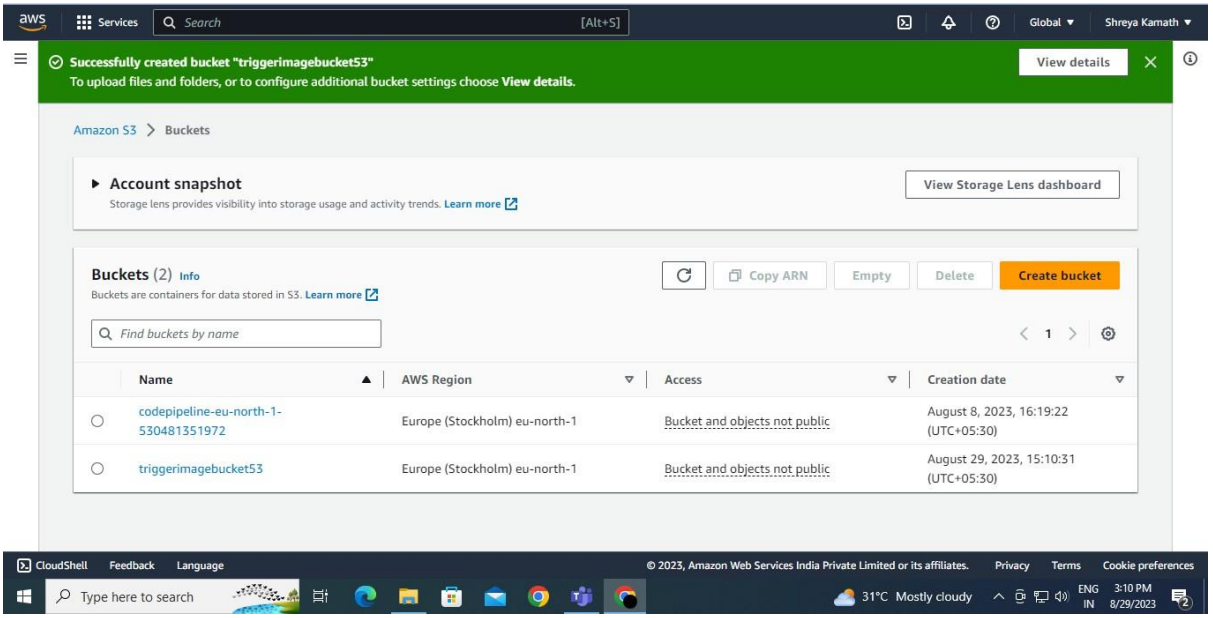
7. Custom Runtimes: You can use custom runtimes to execute functions written in languages not officially supported by AWS.

AWS Lambda is a powerful tool for building serverless applications, automating tasks, and responding to events in a highly scalable and cost-effective manner.

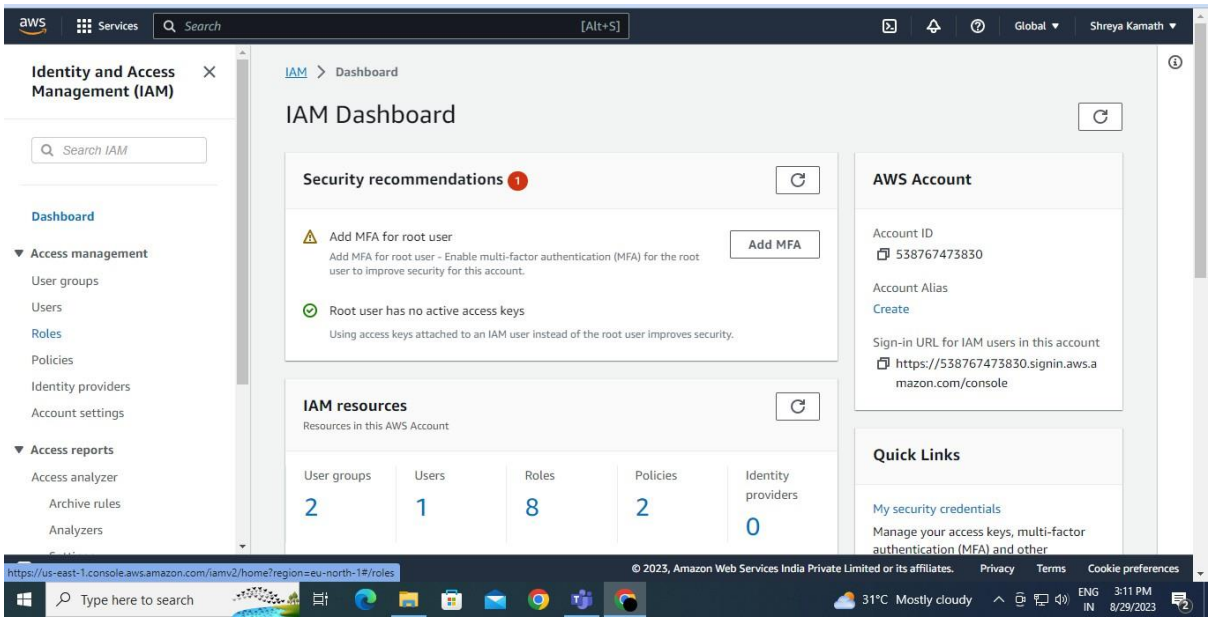

Services
Search
[Alt+S]
Global
Shreya Kamath
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 7
Amazon S3
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
codepipeline-eu-north-1-530481351972
Europe (Stockholm) eu-north-1
Bucket and objects not public
August 8, 2023, 16:19:22 (UTC+05:30)
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:06 PM 8/29/2023

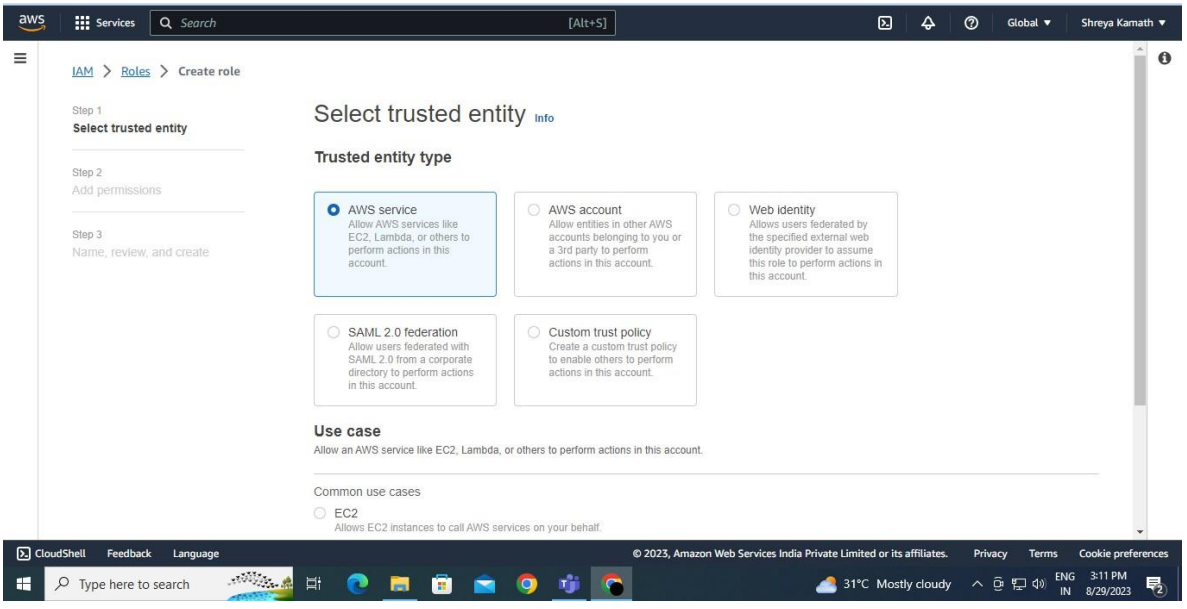

Services
Search
[Alt+S]
Global
Shreya Kamath
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
Bucket name
triggerimagebucket53
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
AWS Region
Europe (Stockholm) eu-north-1
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:10 PM 8/29/2023

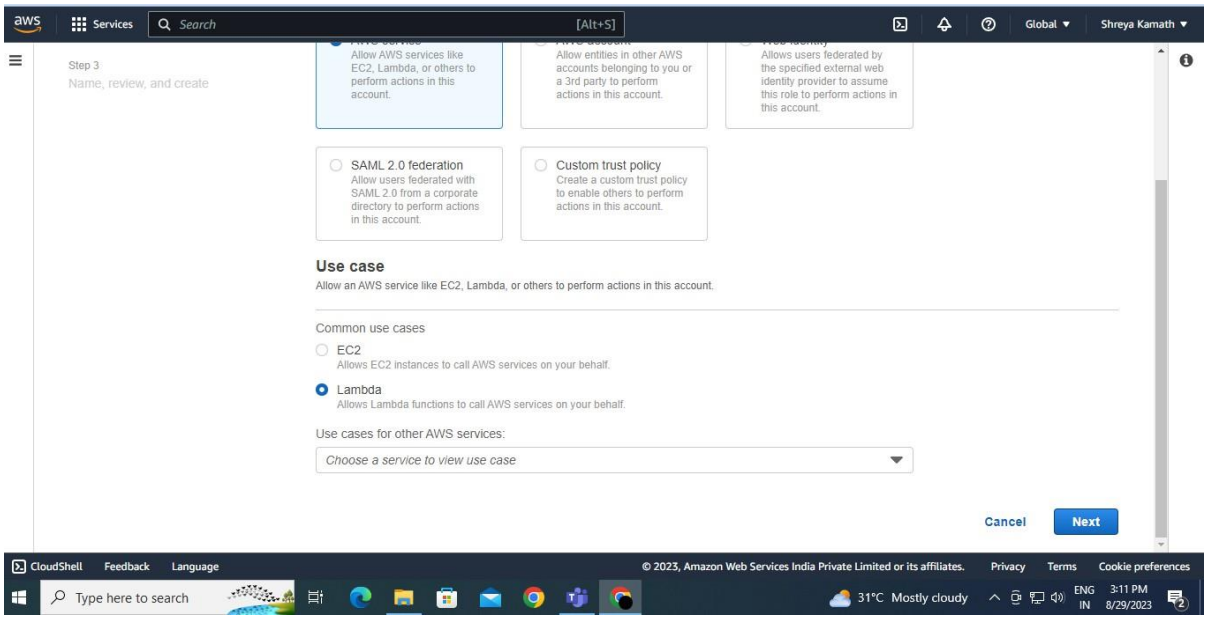

Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

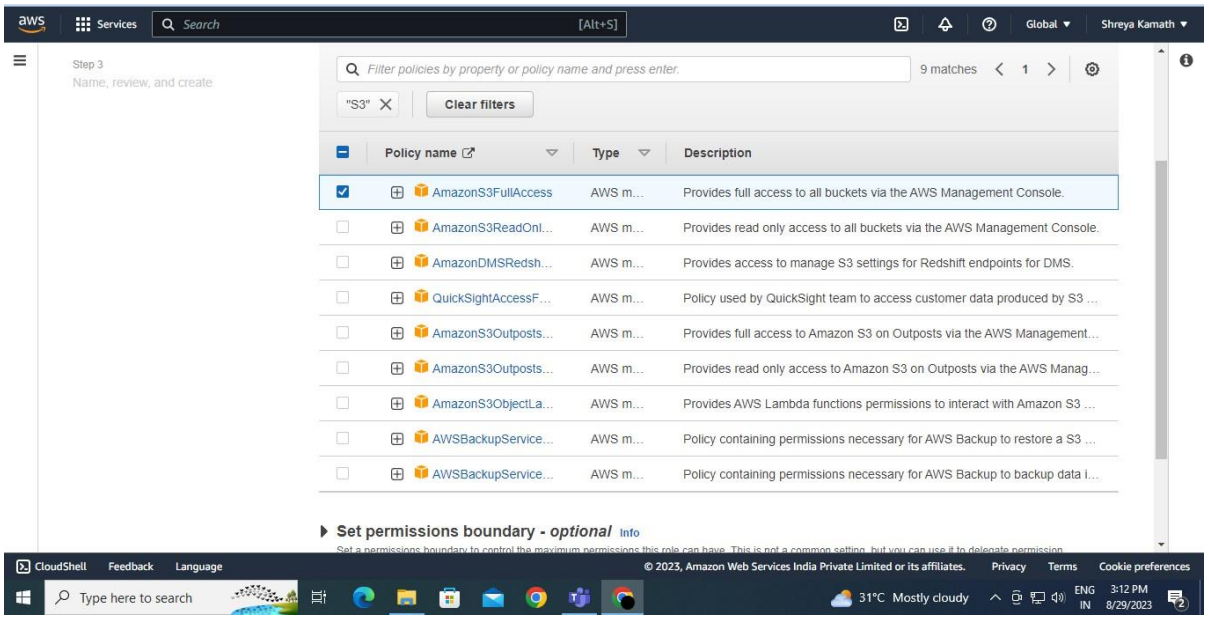

Successfully created bucket "triggerimagebucket53"
To upload files and folders, or to configure additional bucket settings choose View details.
View details
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (2) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
codepipeline-eu-north-1-530481351972
Europe (Stockholm) eu-north-1
Bucket and objects not public
August 8, 2023, 16:19:22 (UTC+05:30)
triggerimagebucket53
Europe (Stockholm) eu-north-1
Bucket and objects not public
August 29, 2023, 15:10:31 (UTC+05:30)

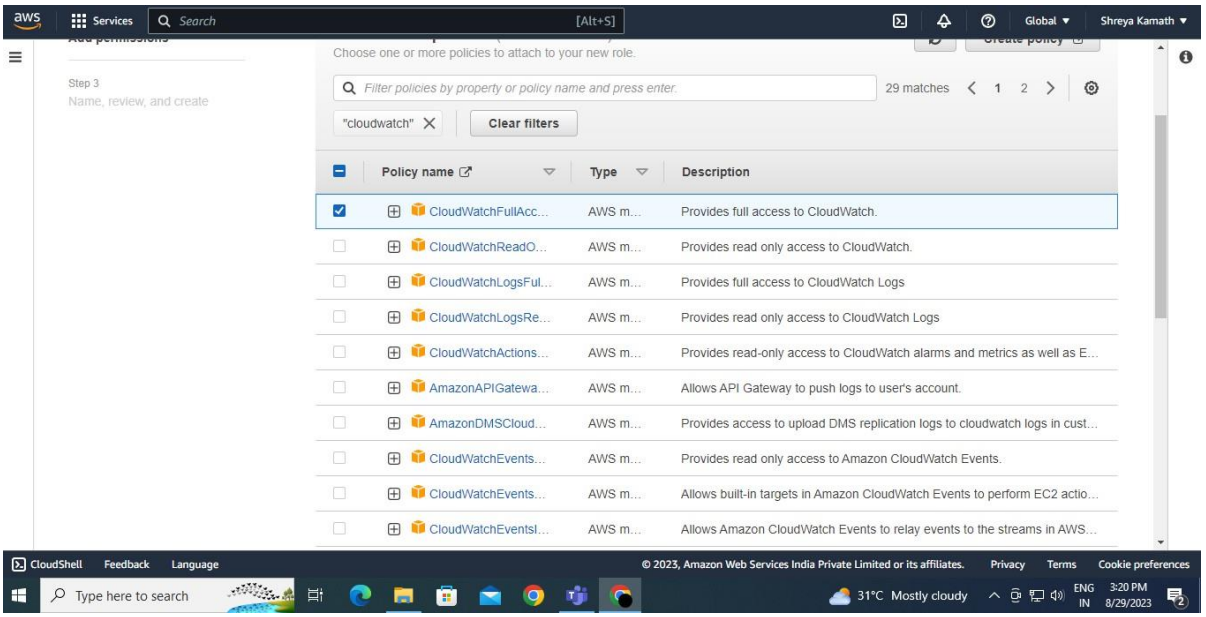
Goint to IAM

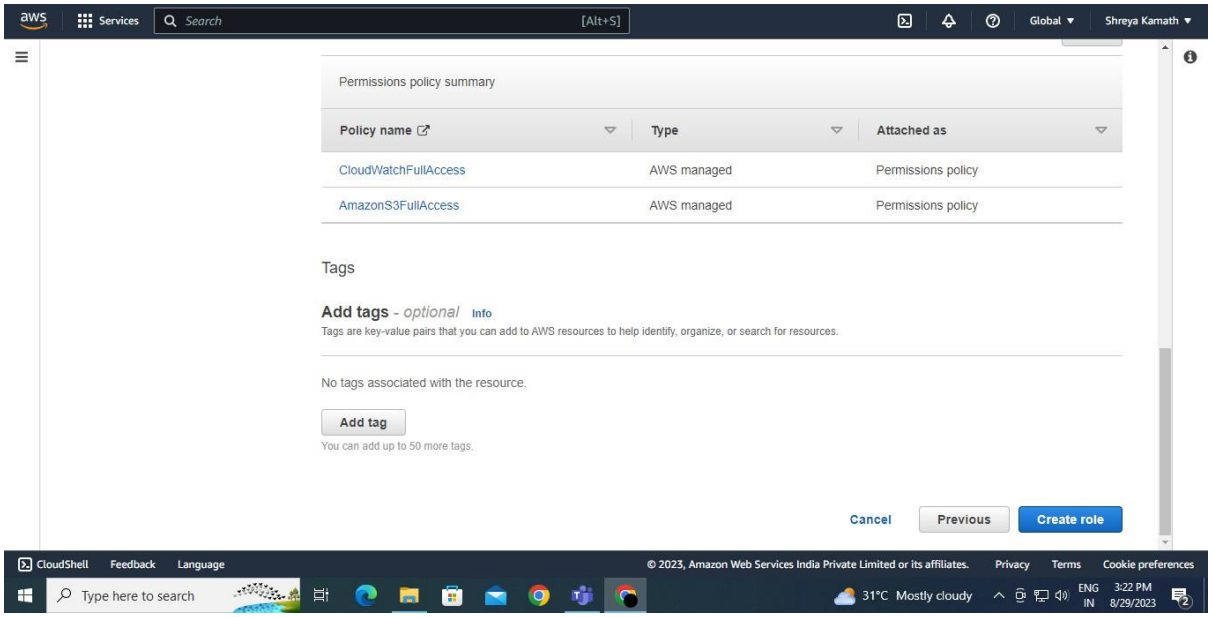

aws
Services
Search
[Alt+S]
Global
Shreya Kamath
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
IAM > Dashboard
IAM Dashboard
Security recommendations 1
Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA
Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.
IAM resources
Resources in this AWS Account
User groups 2
Users 1
Roles 8
Policies 2
Identity providers 0
AWS Account
Account ID
538767473830
Account Alias
Create
Sign-in URL for IAM users in this account
https://538767473830.signin.aws.amazon.com/console
Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other
https://us-east-1.console.aws.amazon.com/iamv2/home?region=eu-north-1#/roles
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:11 PM 8/29/2023

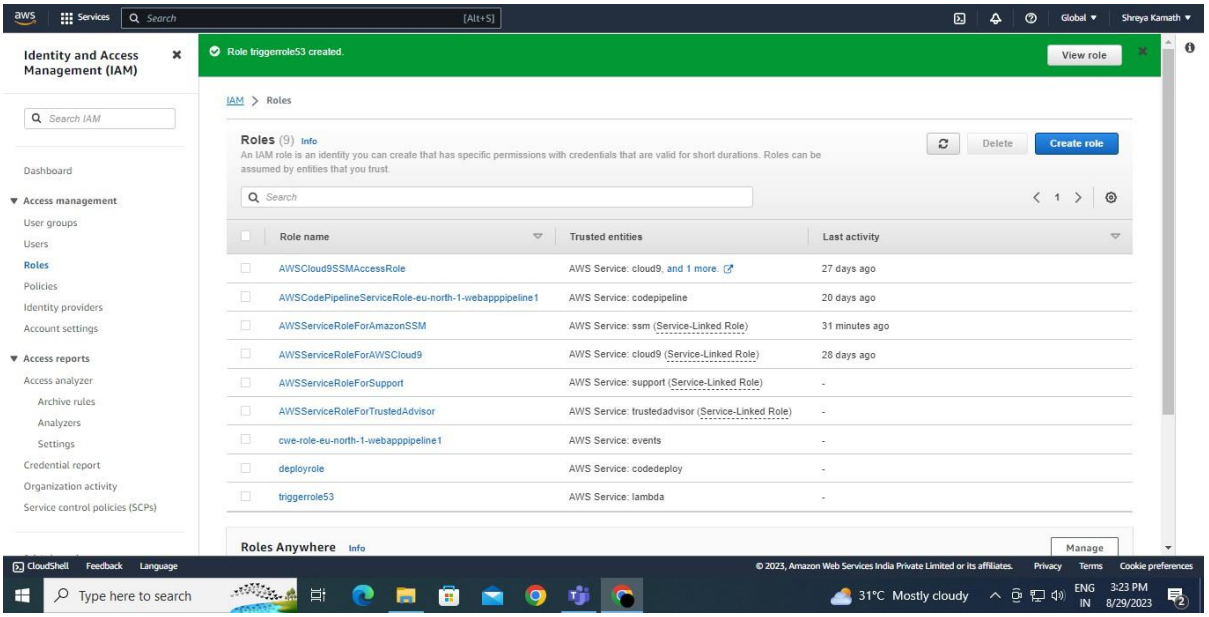

aws
Services
Search
[Alt+S]
Global
Shreya Kamath
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
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.
Common use cases
EC2
Allows EC2 instances to call AWS services on your behalf.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:11 PM 8/29/2023

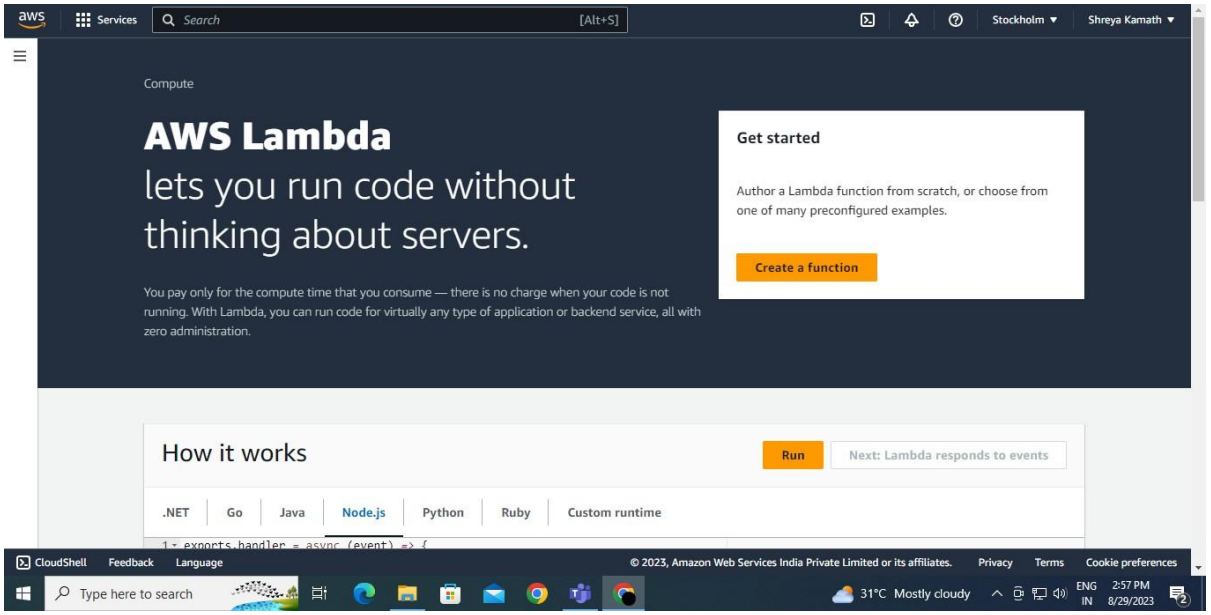

aws
Services
Search
[Alt+S]
Global
Shreya Kamath
Step 3
Name, review, and create
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Allows users federated by the specified 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.
Common use cases
EC2
Allows EC2 instances to call AWS services on your behalf.
Lambda
Allows Lambda functions to call AWS services on your behalf.
Use cases for other AWS services:
Choose a service to view use case
Cancel
Next
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:11 PM
8/29/2023

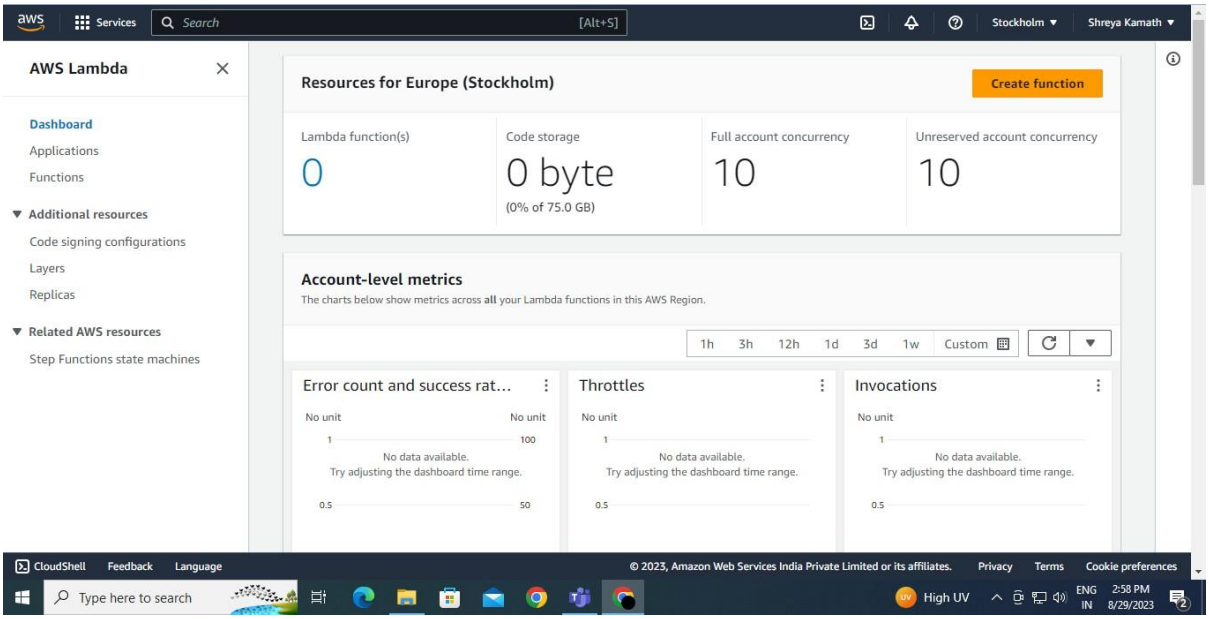

aws
Services
Search
[Alt+S]
Global
Shreya Kamath
Step 3
Name, review, and create
Filter policies by property or policy name and press enter.
9 matches
1
"S3"
Clear filters
Policy name
Type
Description
AmazonS3FullAccess
AWS m...
Provides full access to all buckets via the AWS Management Console.
AmazonS3ReadOnl...
AWS m...
Provides read only access to all buckets via the AWS Management Console.
AmazonDMSRedsh...
AWS m...
Provides access to manage S3 settings for Redshift endpoints for DMS.
QuickSightAccessF...
AWS m...
Policy used by QuickSight team to access customer data produced by S3 ...
AmazonS3Outposts...
AWS m...
Provides full access to Amazon S3 on Outposts via the AWS Management...
AmazonS3Outposts...
AWS m...
Provides read only access to Amazon S3 on Outposts via the AWS Manag...
AmazonS3ObjectLa...
AWS m...
Provides AWS Lambda functions permissions to interact with Amazon S3 ...
AWSBackupService...
AWS m...
Policy containing permissions necessary for AWS Backup to restore a S3 ...
AWSBackupService...
AWS m...
Policy containing permissions necessary for AWS Backup to backup data i...
Set permissions boundary - optional
Info
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:12 PM
8/29/2023

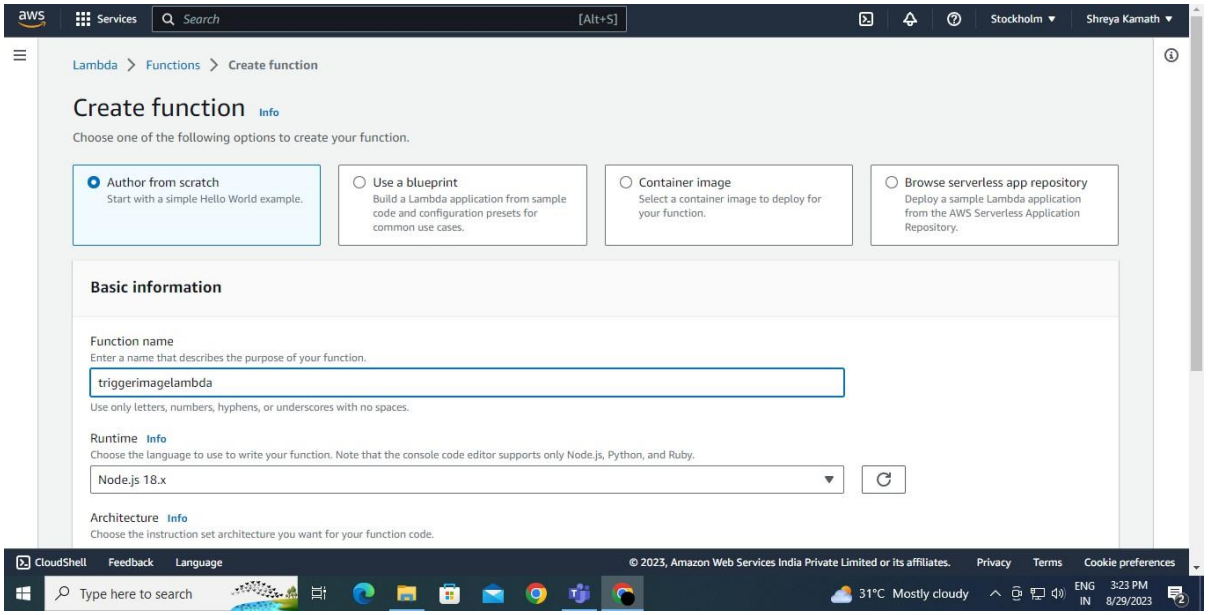

aws
Services
Search
[Alt+S]
Global
Shreya Kamath
Choose one or more policies to attach to your new role.
Step 3
Name, review, and create
Filter policies by property or policy name and press enter.
29 matches
1
2
"cloudwatch"
Clear filters
Policy name
Type
Description
CloudWatchFullAcc...
AWS m...
Provides full access to CloudWatch.
CloudWatchReadO...
AWS m...
Provides read only access to CloudWatch.
CloudWatchLogsFul...
AWS m...
Provides full access to CloudWatch Logs
CloudWatchLogsRe...
AWS m...
Provides read only access to CloudWatch Logs
CloudWatchActions...
AWS m...
Provides read-only access to CloudWatch alarms and metrics as well as E...
AmazonAPIGatewa...
AWS m...
Allows API Gateway to push logs to user's account.
AmazonDMSCloud...
AWS m...
Provides access to upload DMS replication logs to cloudwatch logs in cust...
CloudWatchEvents...
AWS m...
Provides read only access to Amazon CloudWatch Events.
CloudWatchEvents...
AWS m...
Allows built-in targets in Amazon CloudWatch Events to perform EC2 actio...
CloudWatchEventsI...
AWS m...
Allows Amazon CloudWatch Events to relay events to the streams in AWS...
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:20 PM
8/29/2023

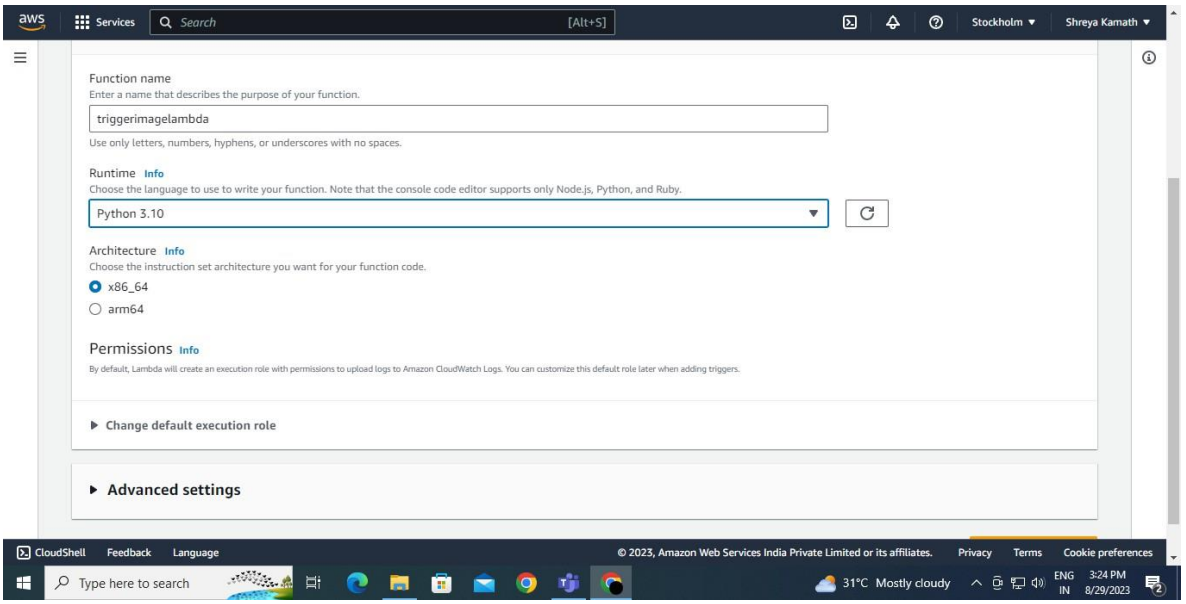

Permissions policy summary
Policy name
Type
Attached as
CloudWatchFullAccess
AWS managed
Permissions policy
AmazonS3FullAccess
AWS managed
Permissions policy
Tags
Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.
Add tag
You can add up to 50 more tags.
Cancel
Previous
Create role
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:22 PM 8/29/2023

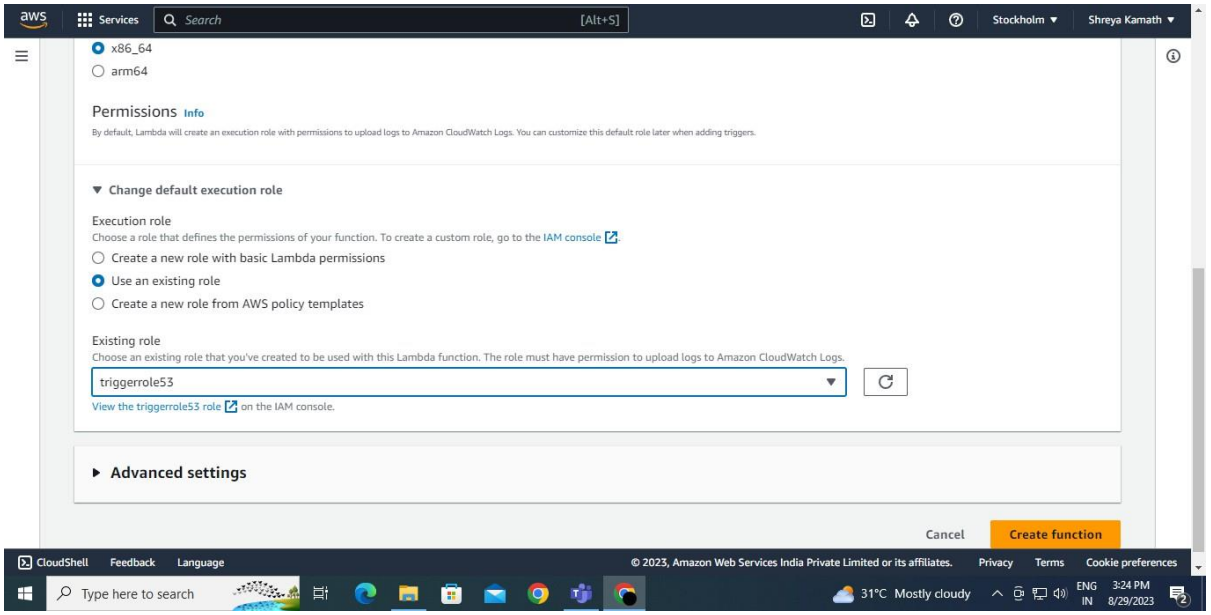

Identity and Access Management (IAM)
Role triggerrole53 created.
View role
IAM > Roles
Roles (9) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Role name
Trusted entities
Last activity
AWSCloud9SSMAccessRole
AWS Service: cloud9, and 1 more.
27 days ago
AWSCodePipelineServiceRole-eu-north-1-webapppipeline1
AWS Service: codepipeline
20 days ago
AWSServiceRoleForAmazonSSM
AWS Service: ssm (Service-Linked Role)
31 minutes ago
AWSServiceRoleForAWSCloud9
AWS Service: cloud9 (Service-Linked Role)
28 days ago
AWSServiceRoleForSupport
AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service-Linked Role)
cwe-role-eu-north-1-webapppipeline1
AWS Service: events
deployrole
AWS Service: codedeploy
triggerrole53
AWS Service: lambda
Roles Anywhere Info
Manage
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:23 PM 8/29/2023

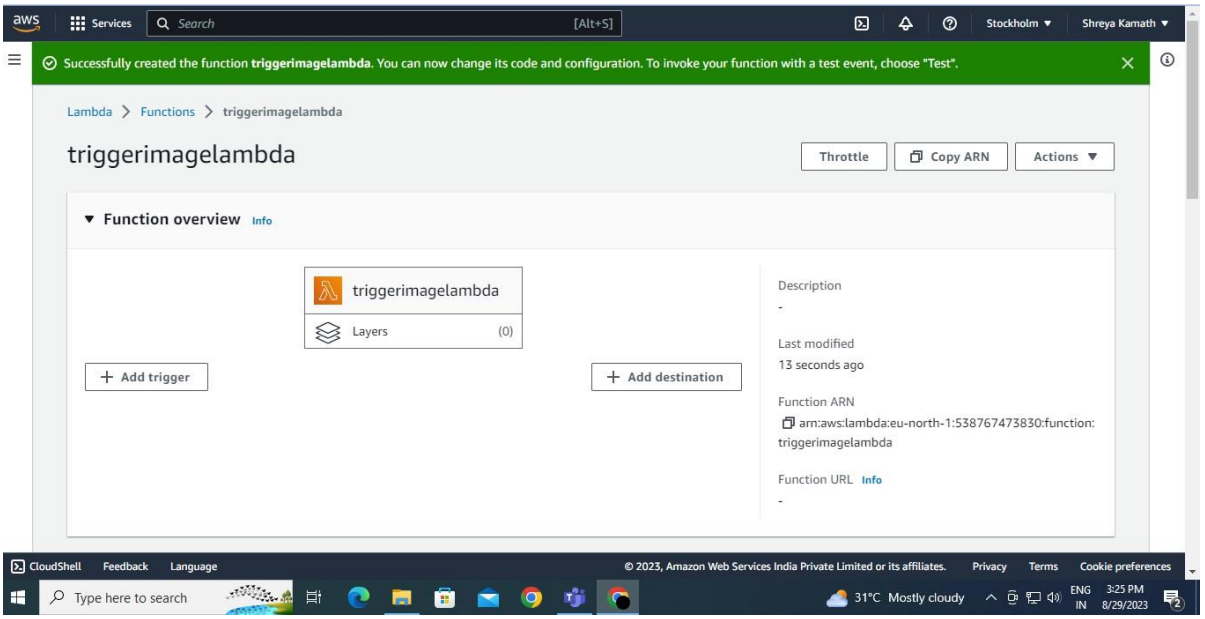

Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Compute
AWS Lambda
lets you run code without thinking about servers.
You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.
Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
Create a function
How it works
Run
Next: Lambda responds to events
.NET
Go
Java
Node.js
Python
Ruby
Custom runtime
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
2:57 PM
8/29/2023

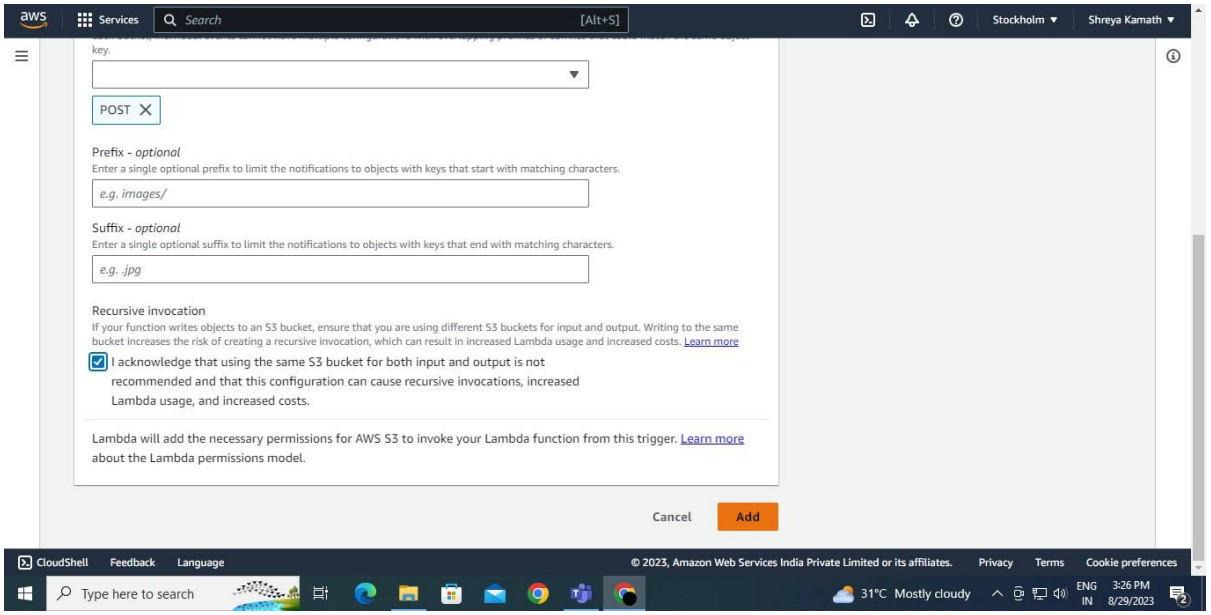

Services
Search
[Alt+S]
Stockholm
Shreya Kamath
AWS Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Layers
Replicas
Related AWS resources
Step Functions state machines
Resources for Europe (Stockholm)
Create function
Lambda function(s)
0
Code storage
0 byte
(0% of 75.0 GB)
Full account concurrency
10
Unreserved account concurrency
10
Account-level metrics
The charts below show metrics across all your Lambda functions in this AWS Region.
1h
3h
12h
1d
3d
1w
Custom
Error count and success rat...
Throttles
Invocations
No unit
No data available.
Try adjusting the dashboard time range.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
High UV
ENG
IN
2:58 PM
8/29/2023

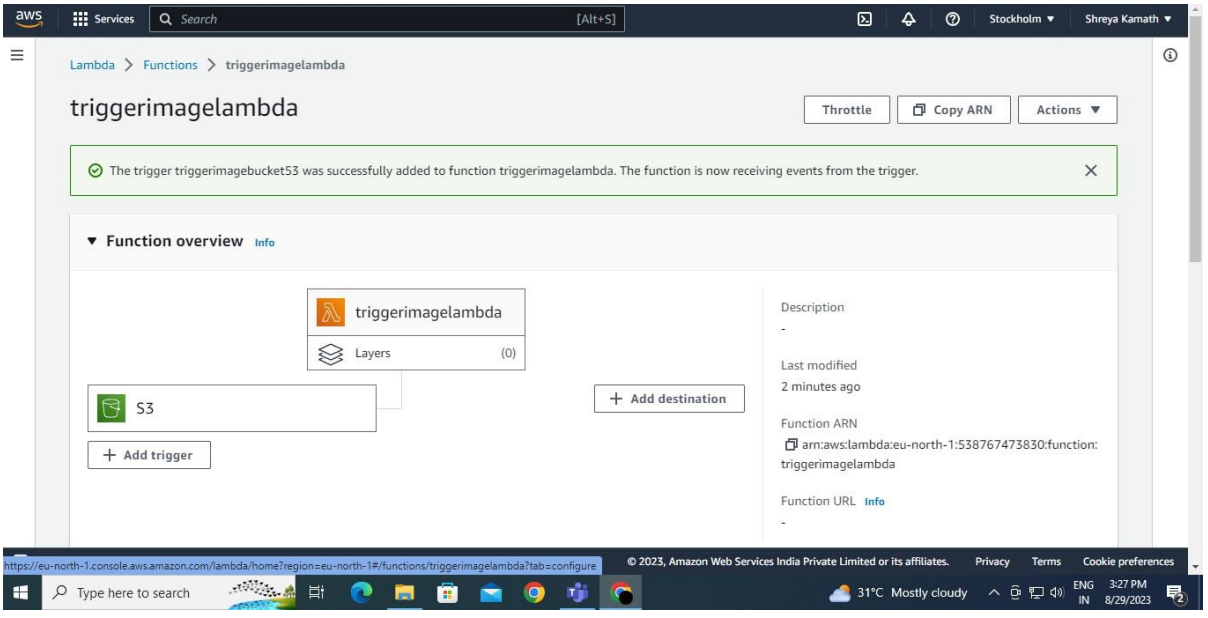

Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Lambda > Functions > Create function
Create function Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration 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.
triggerimagelambda
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.
Node.js 18.x
Architecture Info
Choose the instruction set architecture you want for your function code.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:23 PM
8/29/2023

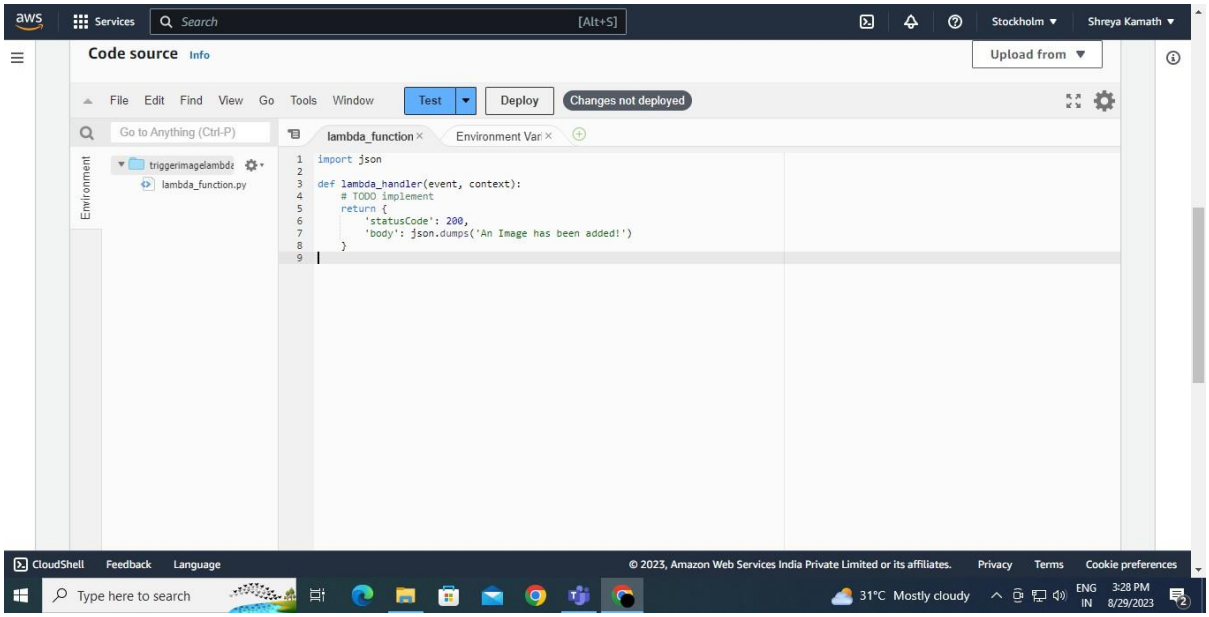

Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Function name
Enter a name that describes the purpose of your function.
triggerimagelambda
Use only letters, numbers, hyphens, or underscores with no spaces.
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.10
Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Advanced settings
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:24 PM
8/29/2023

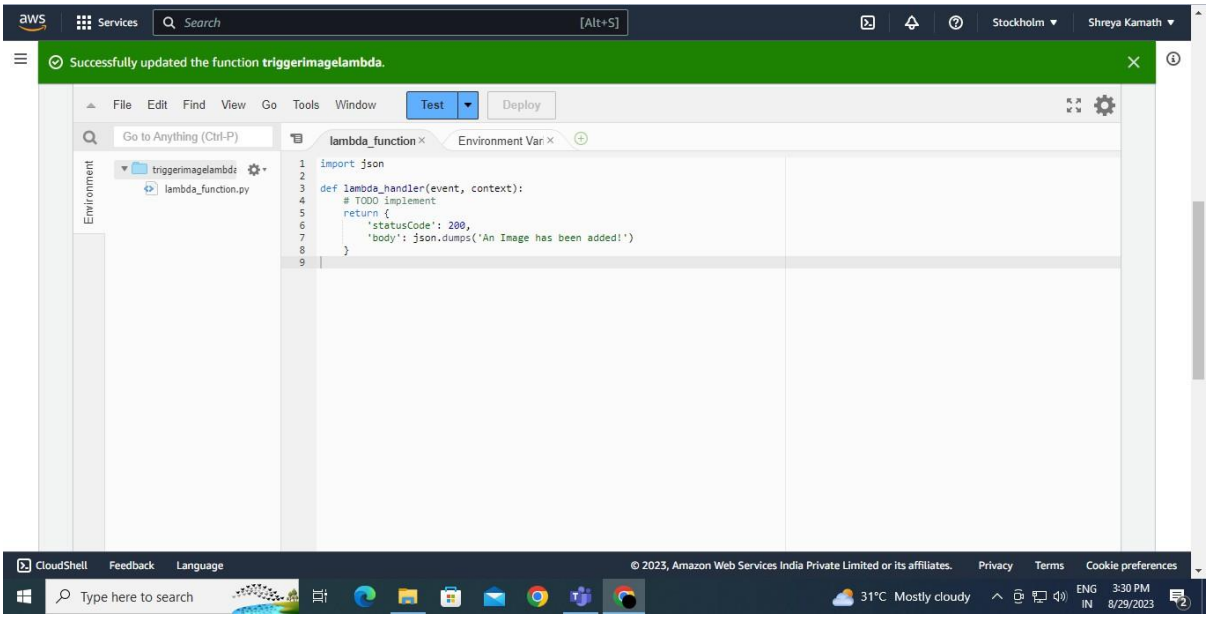

aws
Services
Search
[Alt+S]
Stockholm
Shreya Kamath
x86_64
arm64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
Use an existing role
Create a new role from AWS policy templates
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
triggerrole53
View the triggerrole53 role on the IAM console.
Advanced settings
Cancel
Create function
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:24 PM
8/29/2023


aws
Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Successfully created the function triggerimagelambda. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > triggerimagelambda
triggerimagelambda
Throttle
Copy ARN
Actions
Function overview Info
triggerimagelambda
Layers (0)
Add trigger
Add destination
Description
-
Last modified
13 seconds ago
Function ARN
arn:aws:lambda:eu-north-1:538767473830:function:triggerimagelambda
Function URL Info
-
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:25 PM
8/29/2023

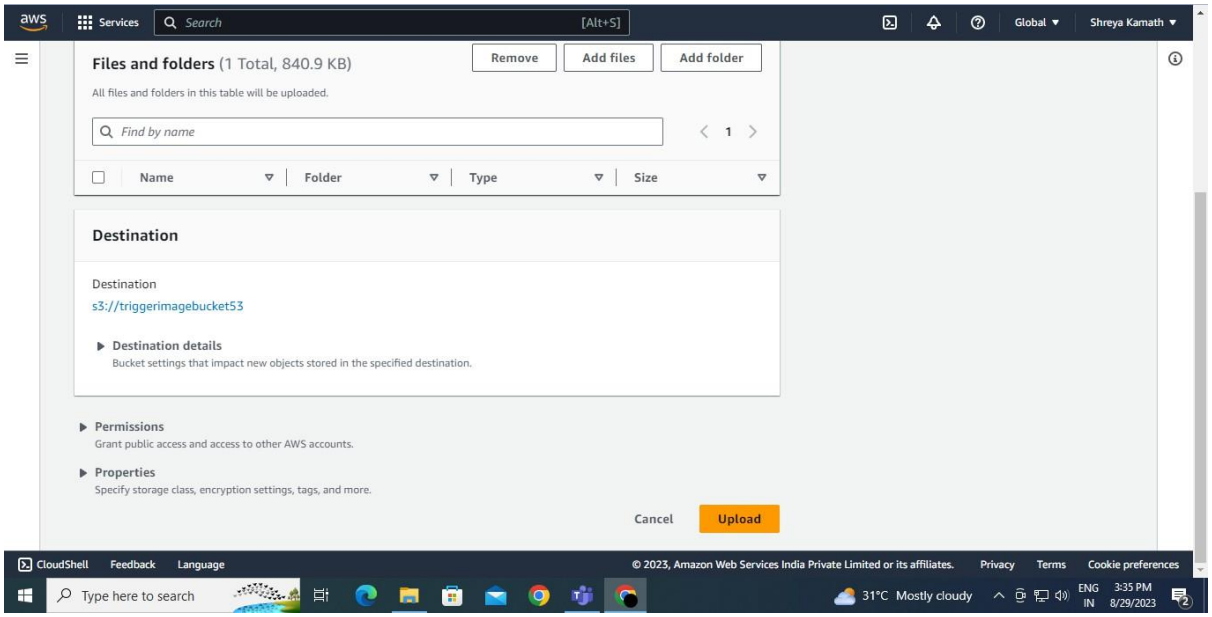

aws
Services
Search
[Alt+S]
Stockholm
Shreya Kamath
POST
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/
Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys 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
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:26 PM
8/29/2023

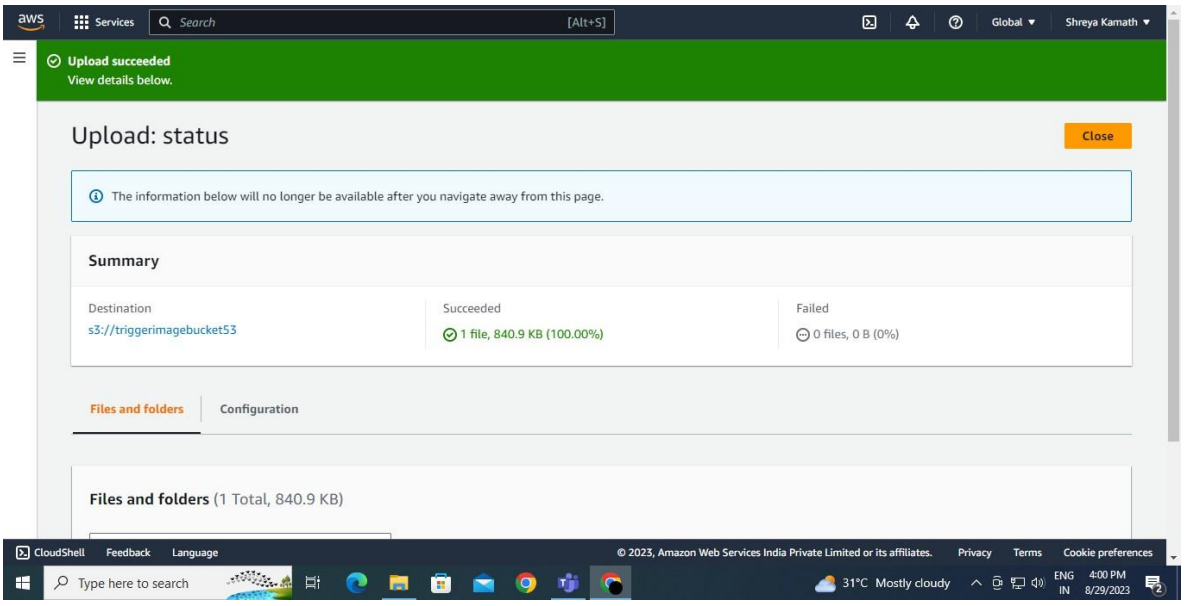

aws
Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Lambda > Functions > triggerimagelambda
triggerimagelambda
Throttle
Copy ARN
Actions
The trigger triggerimagebucket53 was successfully added to function triggerimagelambda. The function is now receiving events from the trigger.
Function overview Info
triggerimagelambda
Layers (0)
S3
Add trigger
Add destination
Description
-
Last modified
2 minutes ago
Function ARN
arn:aws:lambda:eu-north-1:538767473830:function:triggerimagelambda
Function URL Info
-
https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions/triggerimagelambda?tab=configure
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
3:27 PM
8/29/2023

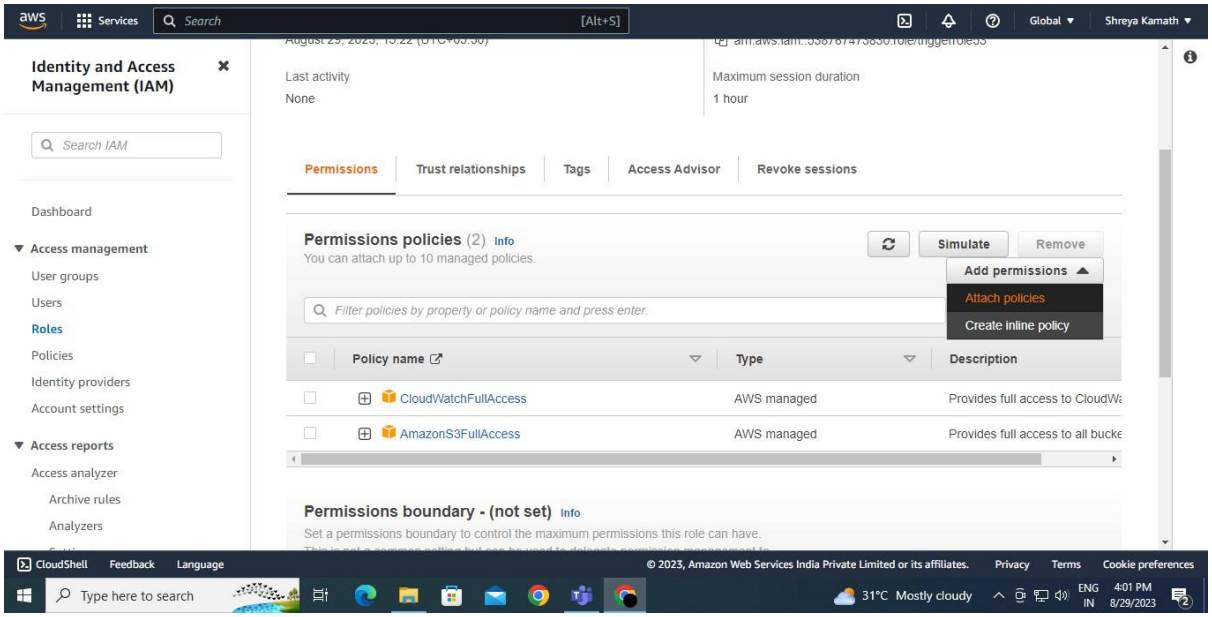

Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Changes not deployed
lambda_function
Environment Var
triggerimagelambda
lambda_function.py
import json
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('An Image has been added!')
}
CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN 3:28 PM 8/29/2023

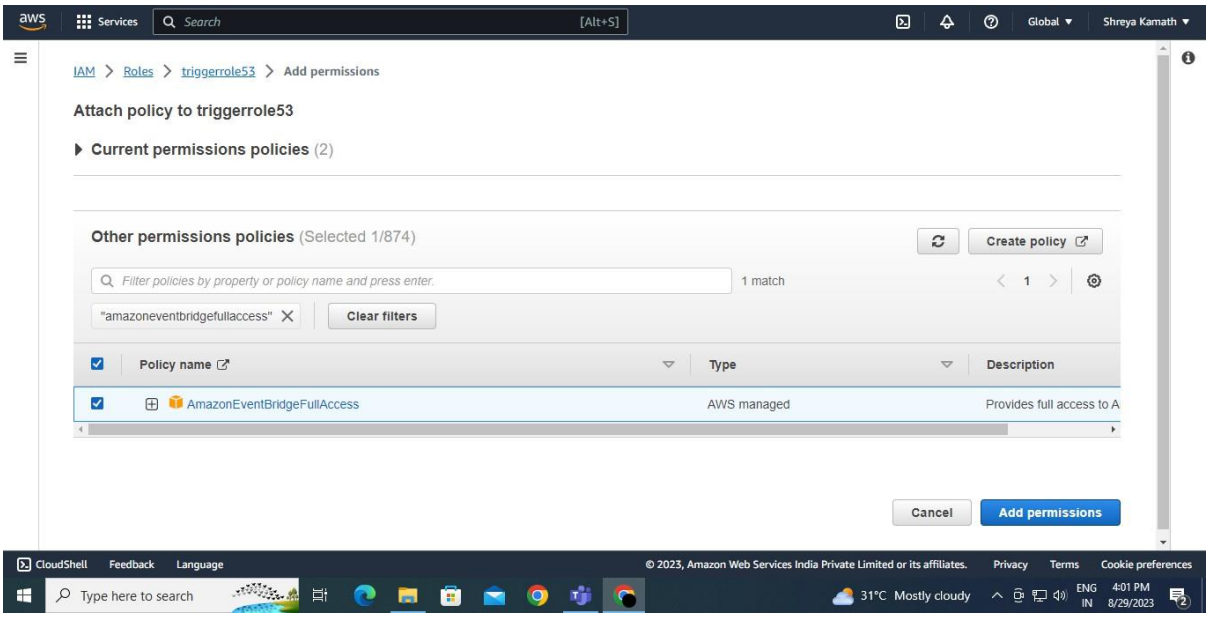

Successfully updated the function triggerimagelambda.
File Edit Find View Go Tools Window
Test
Deploy
lambda_function
Environment Var
triggerimagelambda
lambda_function.py
import json
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('An Image has been added!')
}
CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN 3:30 PM 8/29/2023

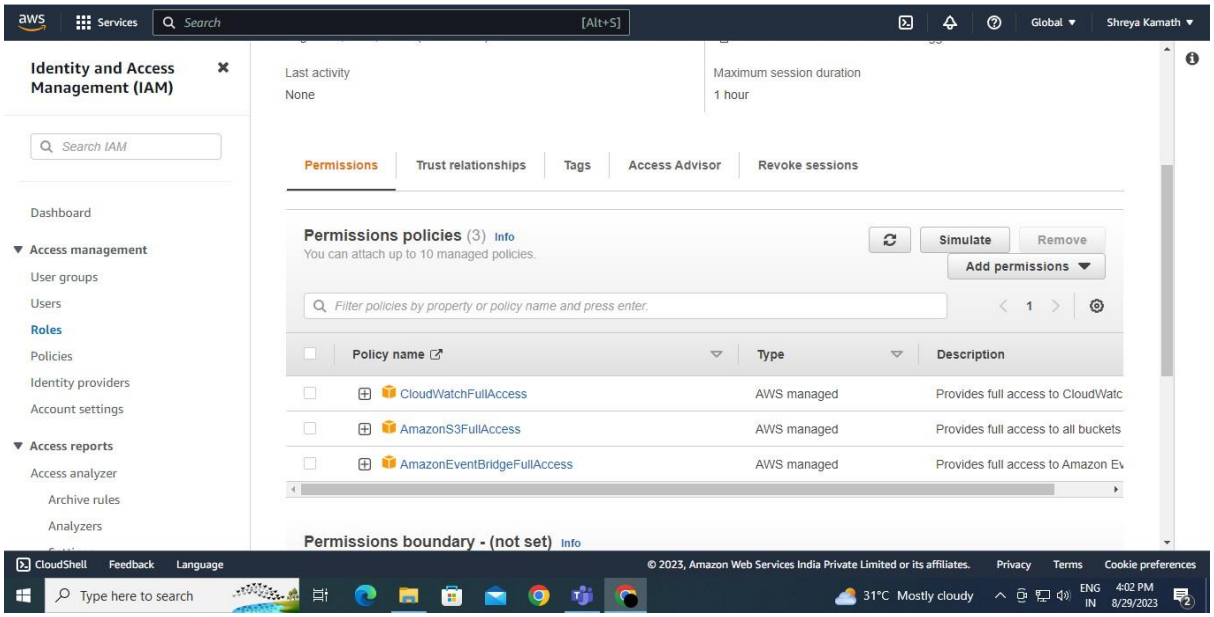
Go back to s3

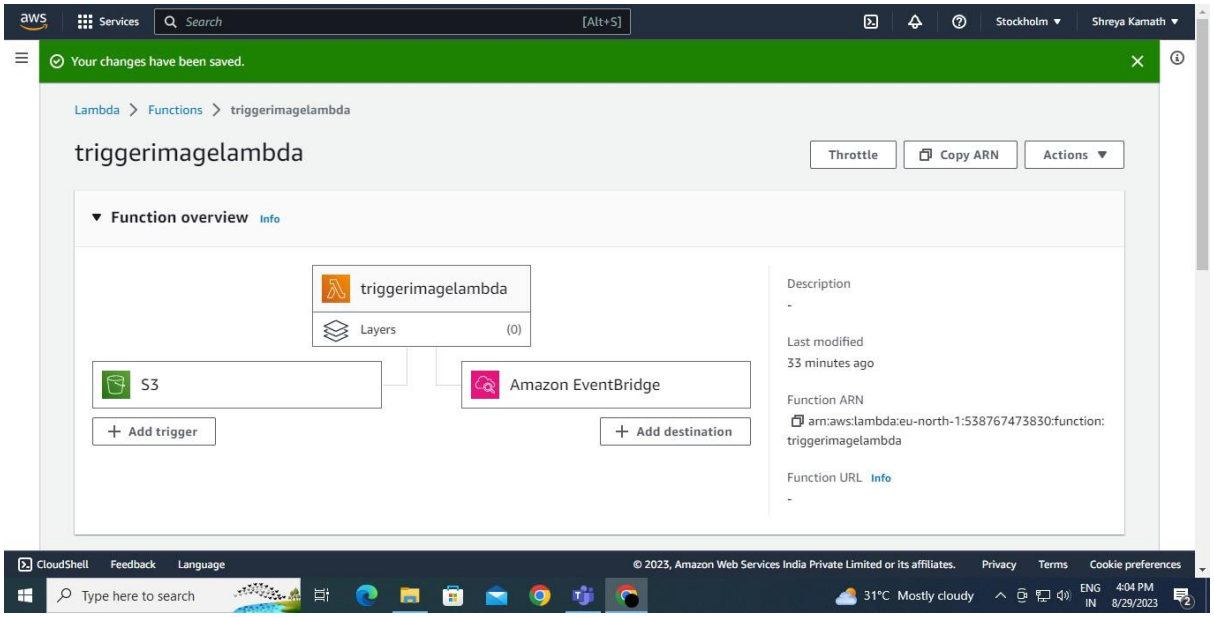

Services
Search
[Alt+S]
Global
Shreya Kamath
Files and folders (1 Total, 840.9 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Type
Size
Destination
Destination
s3://triggerimagebucket53
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.
Properties
Specify storage class, encryption settings, tags, and more.
Cancel
Upload
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
3:35 PM
8/29/2023

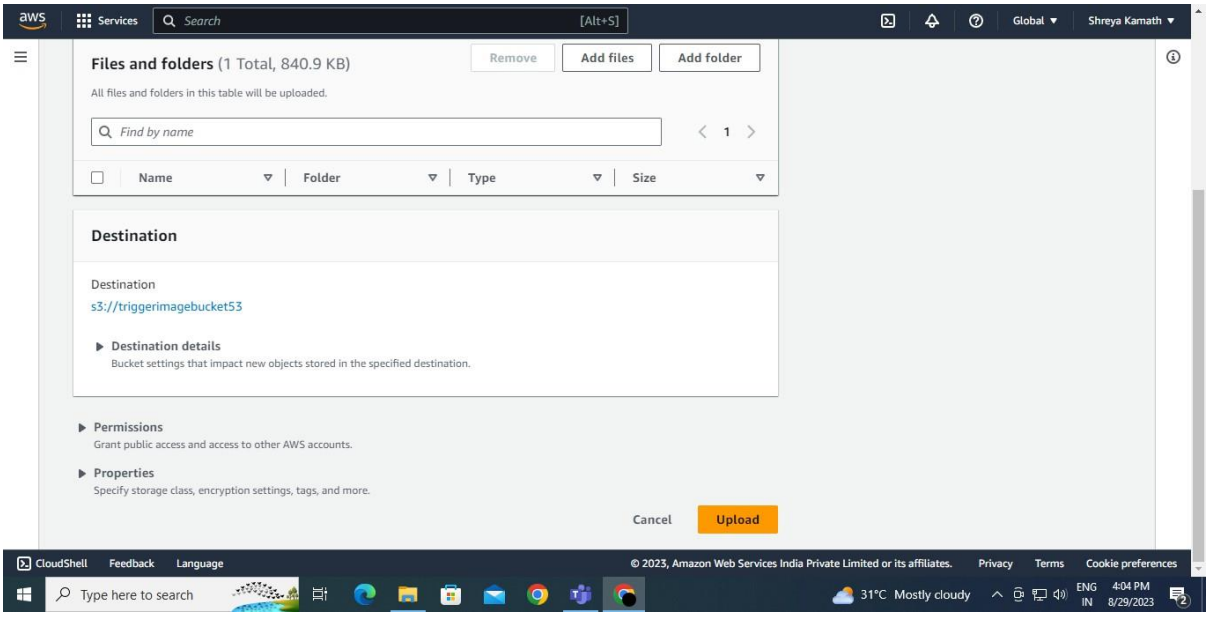

Services
Search
[Alt+S]
Global
Shreya Kamath
Upload succeeded
View details below.
Upload: status
Close
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://triggerimagebucket53
Succeeded
1 file, 840.9 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 840.9 KB)
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
4:00 PM
8/29/2023

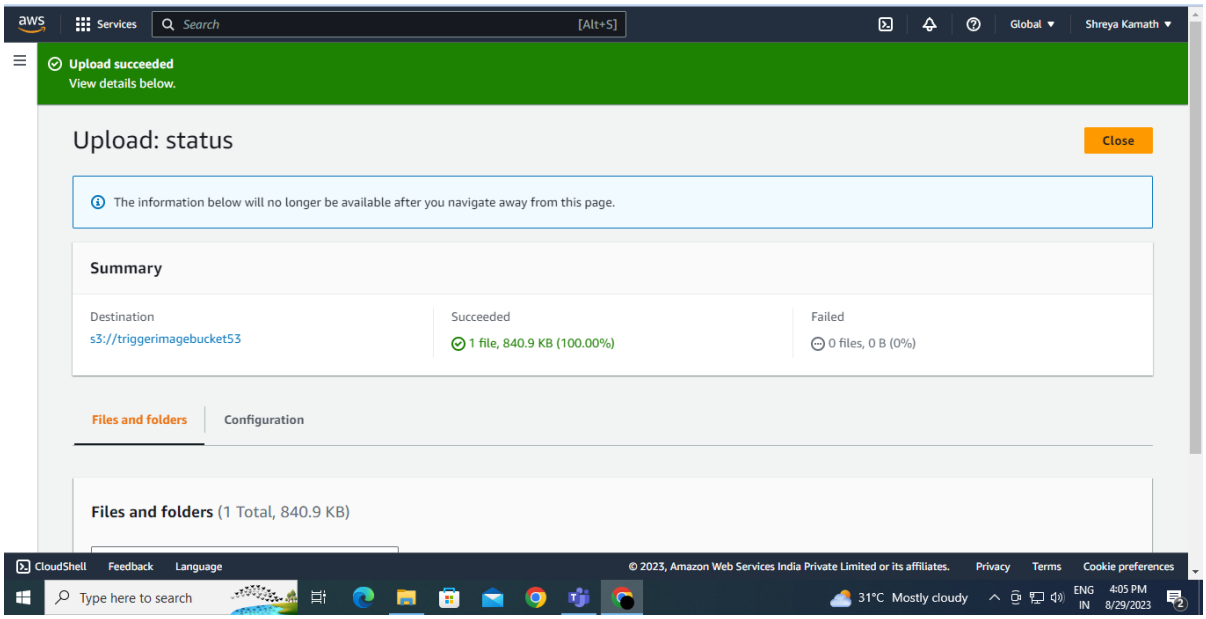

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Last activity
None
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (2) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Attach policies
Create inline policy
Filter policies by property or policy name and press enter.
Policy name
Type
Description
CloudWatchFullAccess
AWS managed
Provides full access to CloudWa
AmazonS3FullAccess
AWS managed
Provides full access to all bucke
Permissions boundary - (not set) Info
Set a permissions boundary to control the maximum permissions this role can have.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
4:01 PM 8/29/2023

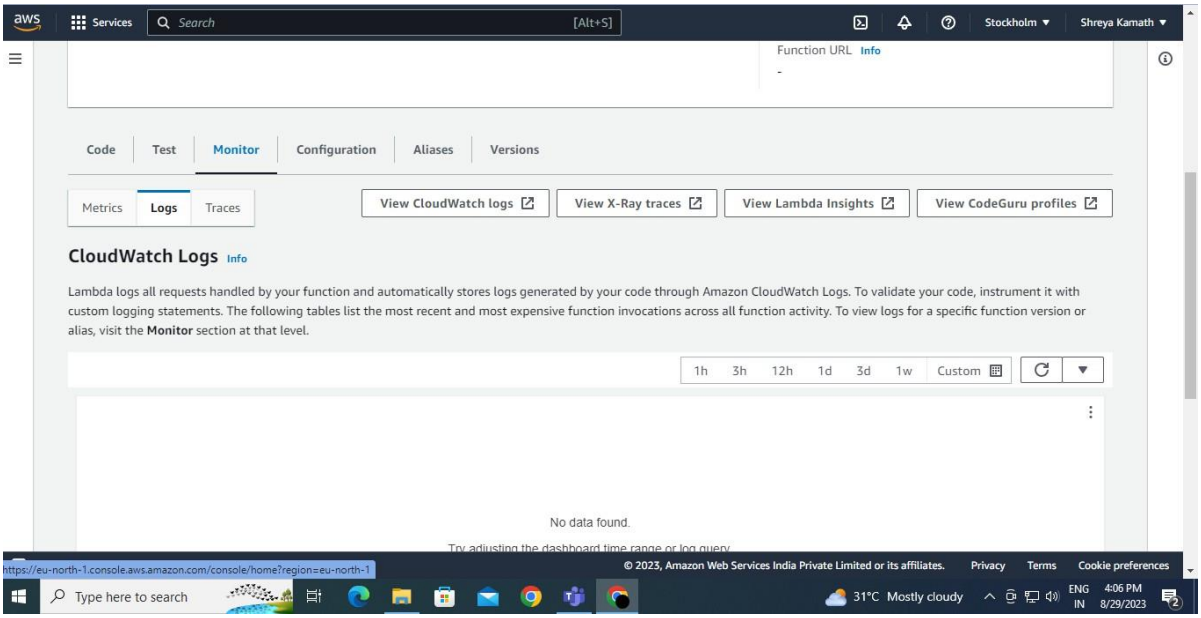

IAM > Roles > triggerrole53 > Add permissions
Attach policy to triggerrole53
Current permissions policies (2)
Other permissions policies (Selected 1/874)
Create policy
Filter policies by property or policy name and press enter.
1 match
"amazoneventbridgefullaccess"
Clear filters
Policy name
Type
Description
AmazonEventBridgeFullAccess
AWS managed
Provides full access to A
Cancel
Add permissions
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
4:01 PM 8/29/2023


aws
Services
Search
[Alt+S]
Global
Shreya Kamath
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Last activity
None
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (3) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter.
1
Policy name
Type
Description
CloudWatchFullAccess
AWS managed
Provides full access to CloudWatc
AmazonS3FullAccess
AWS managed
Provides full access to all buckets
AmazonEventBridgeFullAccess
AWS managed
Provides full access to Amazon Ev
Permissions boundary - (not set) Info
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG
IN
4:02 PM
8/29/2023


Services
Search
[Alt+S]
Stockholm
Shreya Kamath
Your changes have been saved.
Lambda > Functions > triggerimagelambda
triggerimagelambda
Throttle
Copy ARN
Actions
Function overview Info
triggerimagelambda
Layers (0)
S3
Amazon EventBridge
Add trigger
Add destination
Description
-
Last modified
33 minutes ago
Function ARN
arn:aws:lambda:eu-north-1:538767473830:function:triggerimagelambda
Function URL Info
-
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
4:04 PM
8/29/2023


Services
Search
[Alt+S]
Global
Shreya Kamath
Files and folders (1 Total, 840.9 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Type
Size
Destination
Destination
s3://triggerimagebucket53
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.
Properties
Specify storage class, encryption settings, tags, and more.
Cancel
Upload
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly cloudy
ENG IN
4:04 PM
8/29/2023





CONCLUSION:

Hence, I have understood AWS Lambda Functions and created a Lambda function using Python to log “An

Image has been added” message, once a file is added to an S3 Bucket.

Name : Saikarthik Iyer

Batch : T13

Roll no : 41

Assignment 9

Aim : Nagios - Features and Applications / USE

Theory :

Login to Aws

Create Ec2 instances on Aws account of any linux os

Then Run the following command in SS

```
ubuntu@ip-172-31-13-219:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [909 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
```

cd /usr/src/

sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz

```
ubuntu@ip-172-31-13-219:/usr/src$ sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
--2021-10-05 10:24:05--  https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2 [following]
--2021-10-05 10:24:05--  https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2
Resolving codeload.github.com (codeload.github.com)... 13.233.43.20
Connecting to codeload.github.com (codeload.github.com)|13.233.43.20|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'nagios-4.4.2.tar.gz'

nagios-4.4.2.tar.gz                     [    <=>                                                 ]  10.78M  16.9MB/s    in 0.6s

2021-10-05 10:24:06 (16.9 MB/s) - 'nagios-4.4.2.tar.gz' saved [11301457]
```

sudo tar zxf nagios-*.tar.gz cd nagioscore-nagios-*/

```
ubuntu@ip-172-31-13-219:/usr/src$ cd nagioscore-nagios-*/
ubuntu@ip-172-31-13-219:/usr/src/nagioscore-nagios-4.4.2$
```

Now finally run the following command

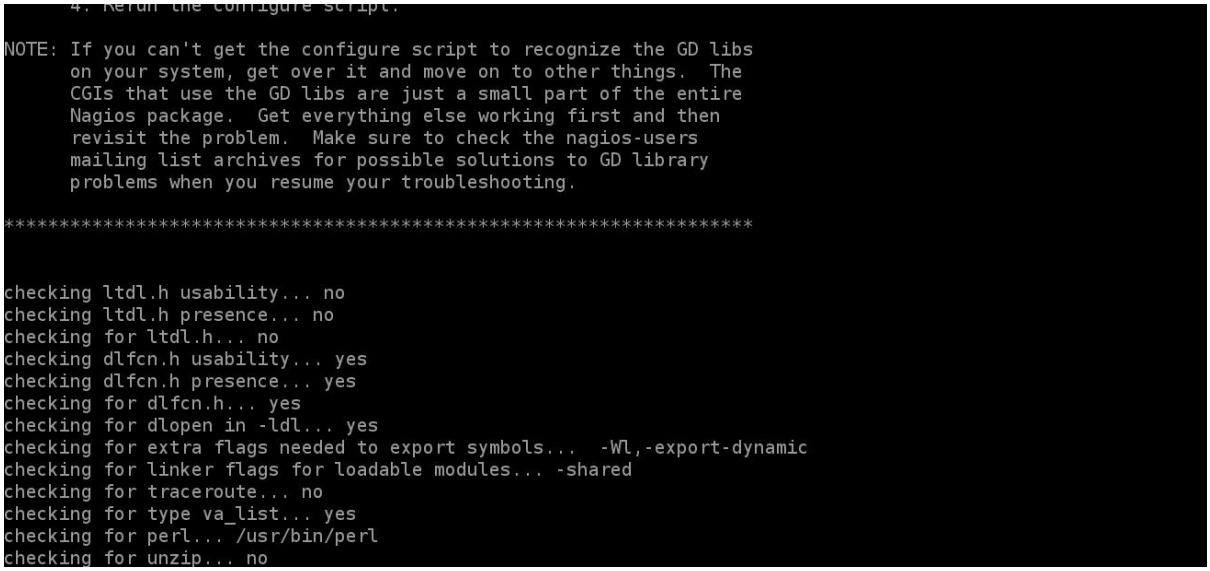
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled

if error comes install c compiler on the linux

by following this link

https://linuxize.com/post/how-to-install-gcc-compiler-on-ubuntu-18-04/

Finally-



Now let us install plugins by

sudo wget -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.g

then

sudo tar zxf nagios-plugins.tar.gz

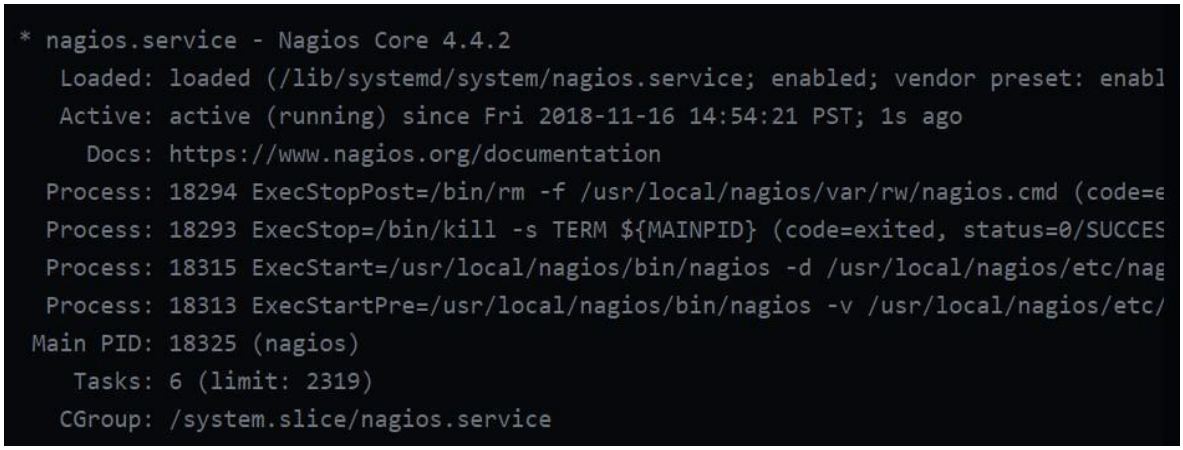
then

cd nagios-plugins-release-2.2.1 finally

start and then check status of nagi os

sudo systemctl start nagios sudo

systemctl status nagios



What is Nagios? State three features with brief explaination and the applications of Nagios?

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.

The three features of NagiOs are as follows:

1. Host and Service Monitoring: Nagios allows you to monitor both hosts (e.g., servers, network devices) and services (e.g., HTTP, FTP, disk usage). It provides real-time visibility into the health and performance of your IT infrastructure.

2. Flexible Configuration: Nagios features a highly configurable setup that allows you to define monitoring checks, schedules, and thresholds through text-based configuration files. This flexibility enables you to tailor monitoring to your specific needs.

3. Distributed Monitoring: Nagios supports distributed monitoring setups, allowing you to monitor multiple sites or networks from a central Nagios server. This is useful for large or geographically dispersed environments.

The applications of NagiOs are:

1. Network Monitoring: Nagios can monitor network devices such as routers, switches, and firewalls. It checks for network connectivity, bandwidth usage, and the status of various network services.

2. Server Monitoring: It is widely used to monitor servers, including their hardware health, operating system performance, and running services. This includes checking CPU load, memory usage, disk space, and more.

3. Application Monitoring: Nagios can be configured to monitor the performance and availability

of applications and services, such as web servers (Apache, Nginx), databases (MySQL, PostgreSQL), and email servers.

4. Performance Monitoring: By integrating with performance graphing tools, Nagios can track and visualize key performance metrics over time. This helps in capacity planning and identifying performance trends.

5. Security Monitoring: Nagios can monitor security-related aspects of your infrastructure, such as unauthorized access attempts or changes in critical system files, helping to ensure the integrity and security of your systems.

Why are we using NagiOs?

Nagios is used primarily for monitoring IT infrastructure to ensure that systems and services are operating correctly. Here are several reasons why organizations choose Nagios:

1. Comprehensive Monitoring: Nagios provides detailed monitoring of a wide range of IT components, including servers, network devices, applications, and services. This comprehensive approach helps ensure that all critical aspects of your infrastructure are under watch.

2. Real-Time Alerts and Notifications: It offers real-time alerts and notifications for issues like downtime, performance degradation, or configuration changes. This allows IT teams to quickly respond to and address problems before they impact users.

3. Customizability: Nagios is highly configurable, allowing users to tailor the monitoring setup to their specific needs. You can define custom checks, thresholds, and notification rules, and create a monitoring solution that fits your environment precisely.

4. Extensive Plugin Ecosystem: With a large repository of plugins and add-ons, Nagios can be extended to monitor a wide variety of applications, services, and hardware. This flexibility helps in adapting the tool to diverse and evolving monitoring needs.

5. Scalability: Nagios can scale from monitoring a single server to a large, distributed network. Its architecture supports distributed and redundant monitoring setups, making it suitable for both small and large environments.

Nagios is used to ensure the reliability, performance, and security of IT systems, providing peace of mind and operational efficiency for organizations.

Conclusion : LO1 - To understand the fundamentals of cloud computing and be fully proficient with cloud based DevOps solution deployment options to meet your business needs.

Name : Saikarthik Iyer

Batch : T13

Roll no : 41

ASSIGNMENT 10

Aim : To implement lambda function

Theory :

AWS Lambda is a serverless compute service that enables you to execute code in response to events without the need to manage servers. This service is designed to be event-driven, with functions triggered by various events and services such as changes in Amazon S3, incoming HTTP requests via Amazon API Gateway, or events from AWS services like SNS or CloudWatch. Lambda functions are small, focused units of code that execute quickly and can be written in various programming languages. With a pay-as-you-go pricing model, you only pay for the compute time your functions consume. It integrates seamlessly with other AWS services, making it a versatile solution for building serverless applications and automating workflows.

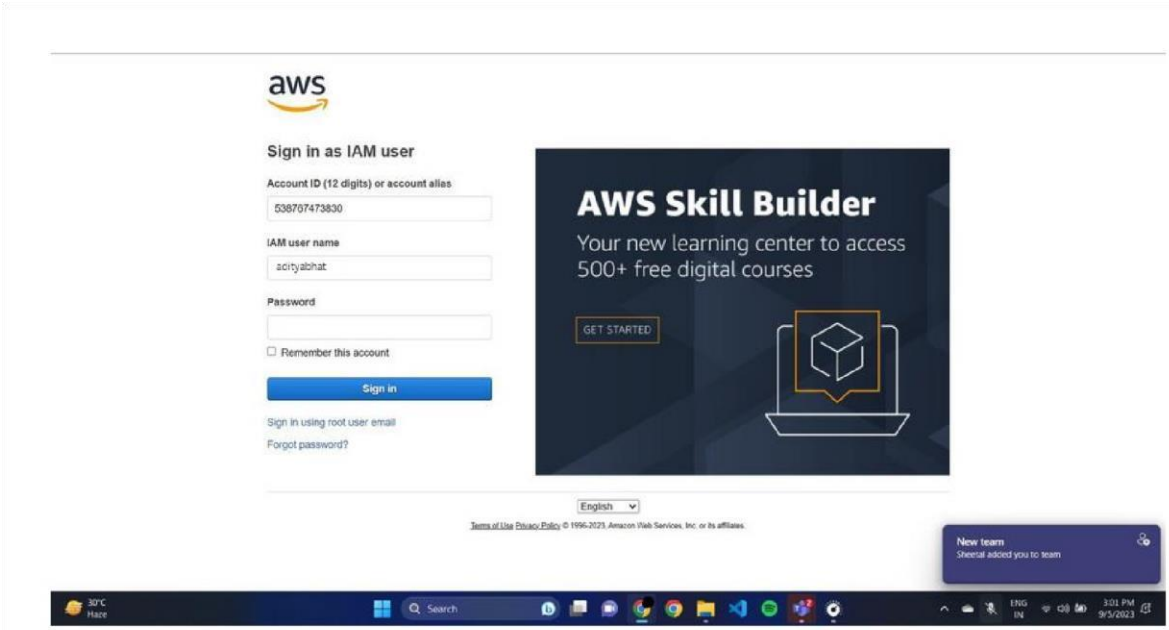
Advantages of Using S3 with Lambda:

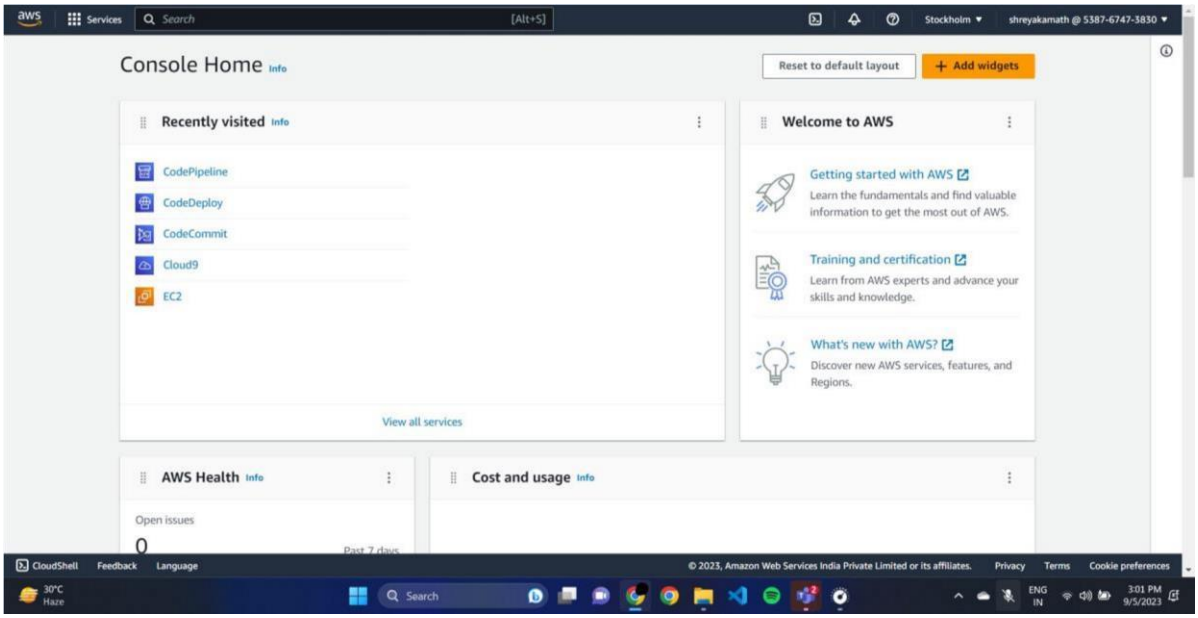
Automation: The integration enables seamless automation of workflows without the need for human intervention.

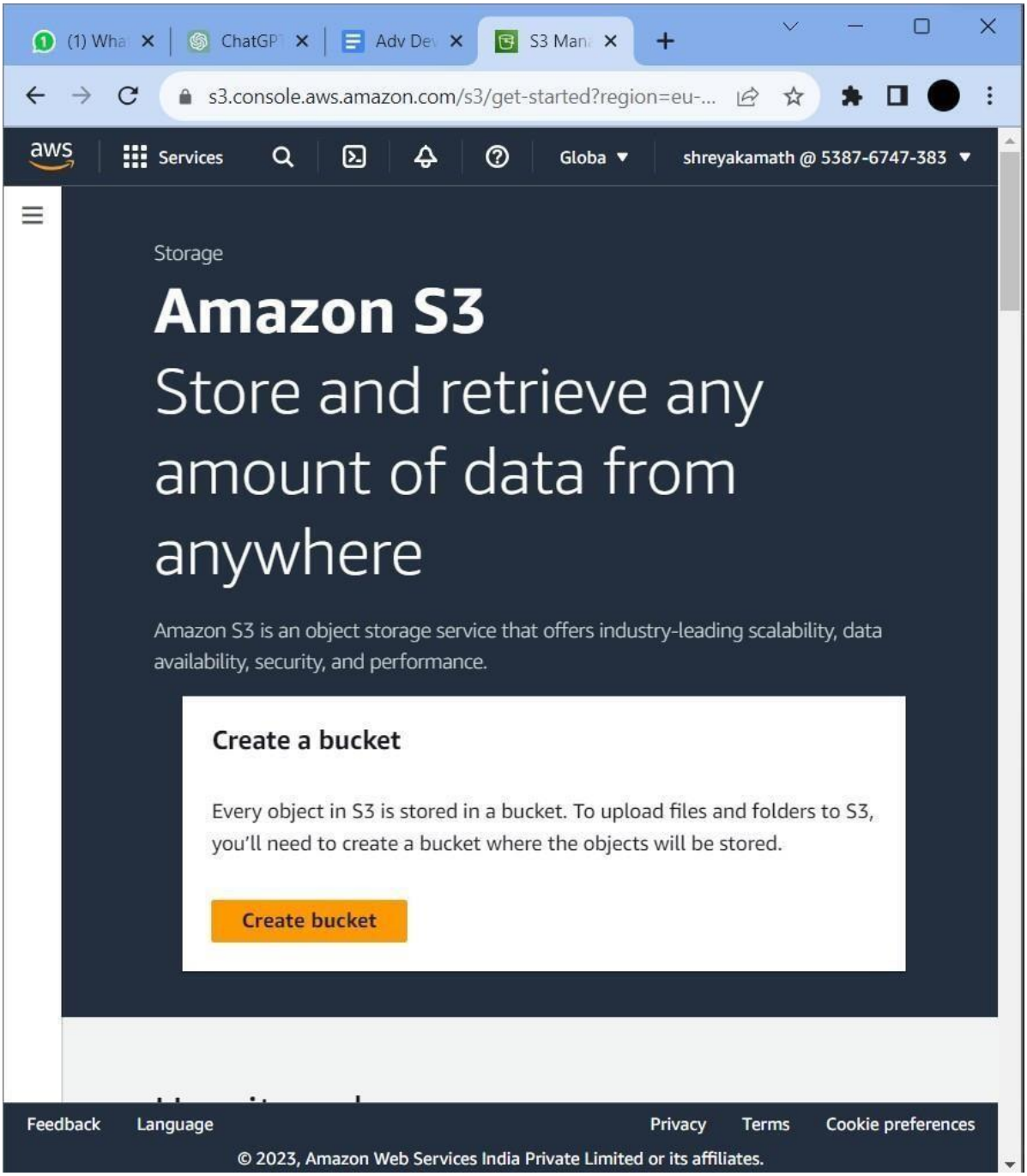
Scalability: Both S3 and Lambda scale automatically with the amount of data and number of requests.

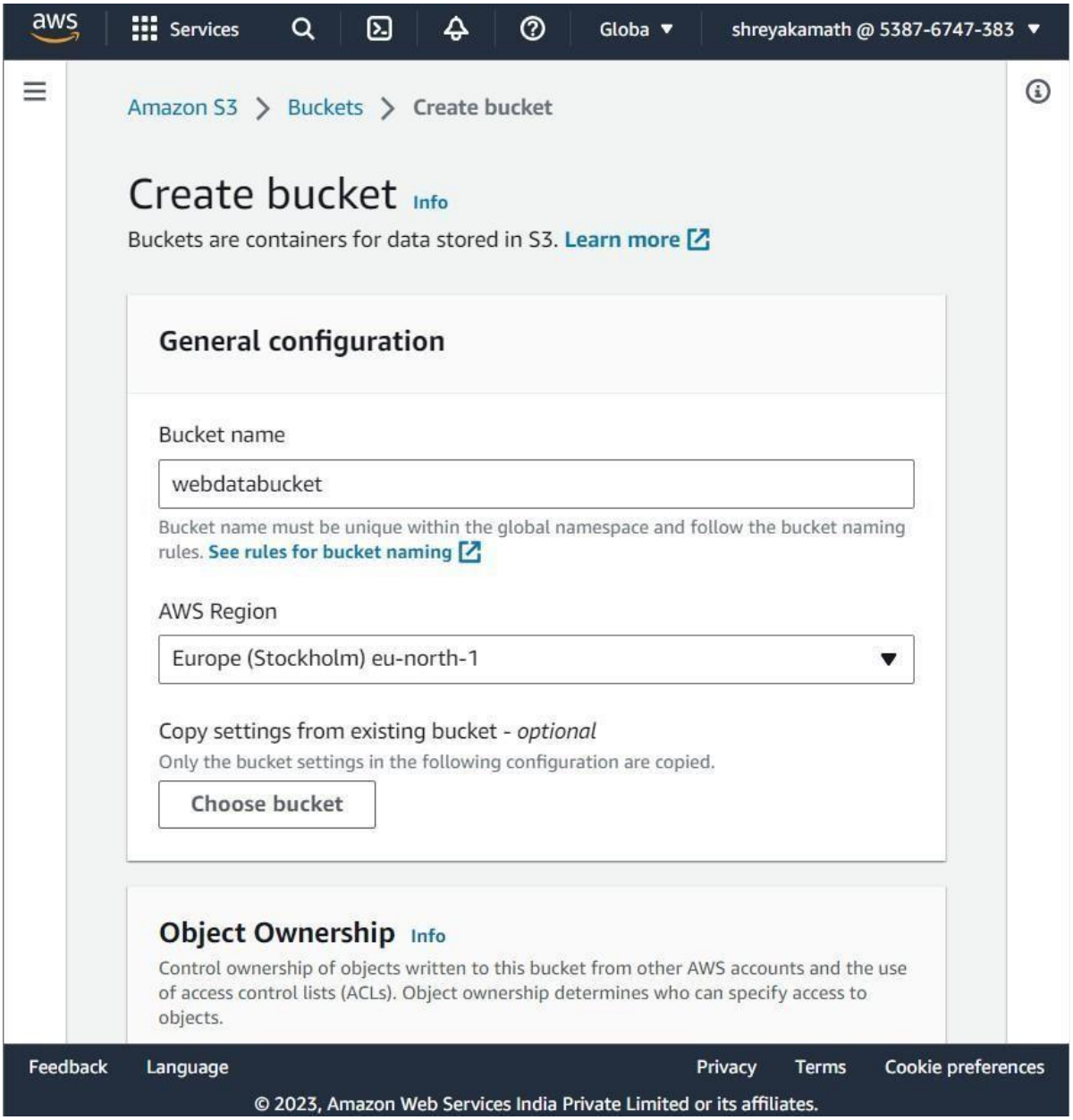
Cost Efficiency: By using Lambda, you only pay for the compute time used, and by using S3, you only pay for the storage you consume.

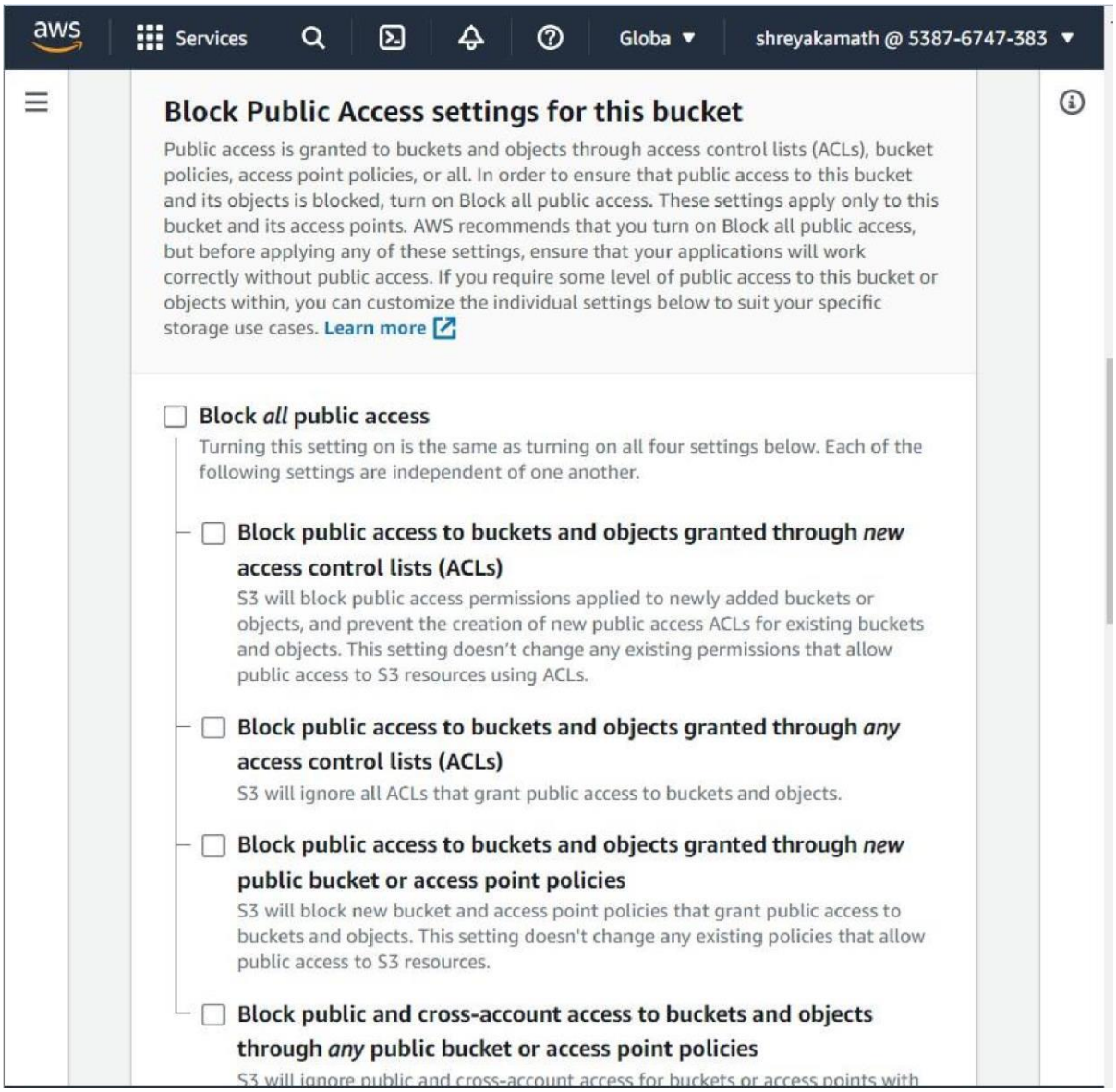
Ease of Use: The combination of S3 and Lambda simplifies complex tasks like data processing, logging, and notifications.

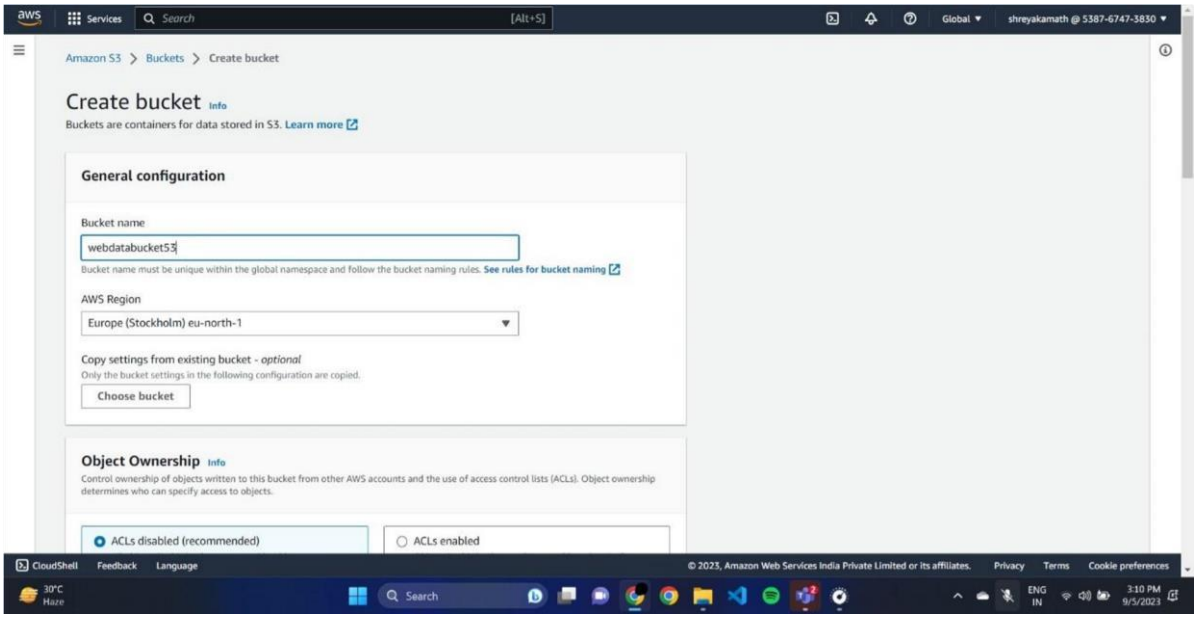

aws
Sign in as IAM user
Account ID (12 digits) or account alias
538707473830
IAM user name
acityabhat
Password
Remember this account
Sign in
Sign in using root user email
Forgot password?
AWS Skill Builder
Your new learning center to access 500+ free digital courses
GET STARTED
English
Terms of Use Privacy Policy © 1996-2023, Amazon Web Services, Inc. or its affiliates.
New team
Sheetal added you to team
30°C Haze
Search
ENG IN
3:01 PM 9/5/2023

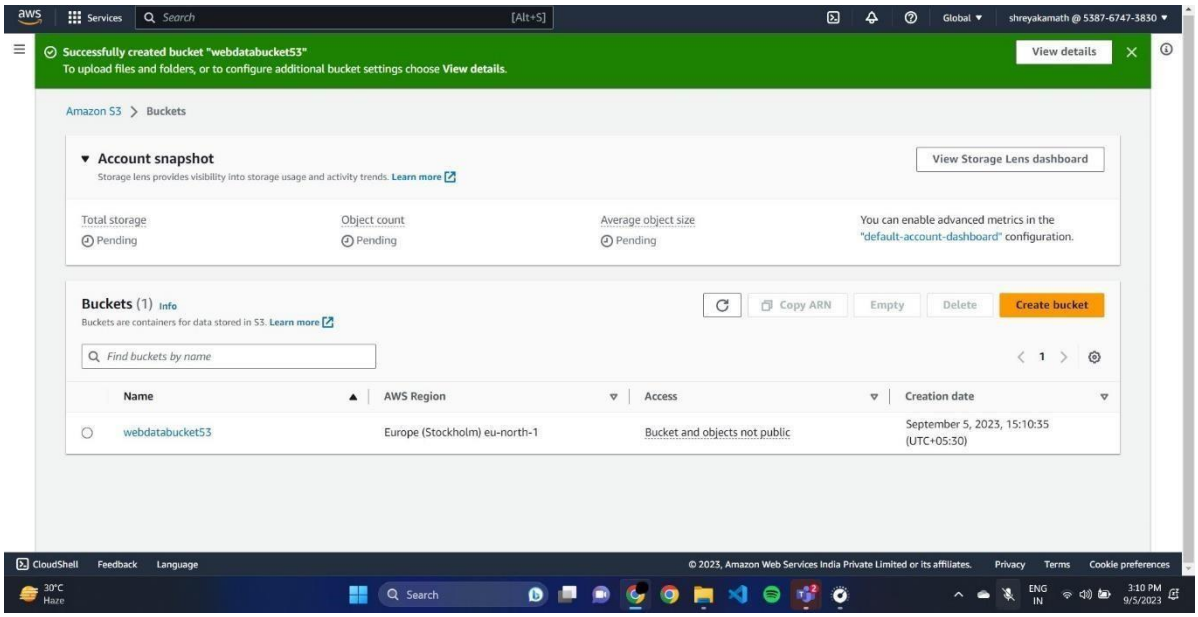

aws
Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
CodePipeline
CodeDeploy
CodeCommit
Cloud9
EC2
View all services
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.
AWS Health Info
Open issues
0
Past 7 days
Cost and usage Info
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
30°C Haze
Search
ENG IN
3:01 PM 9/5/2023

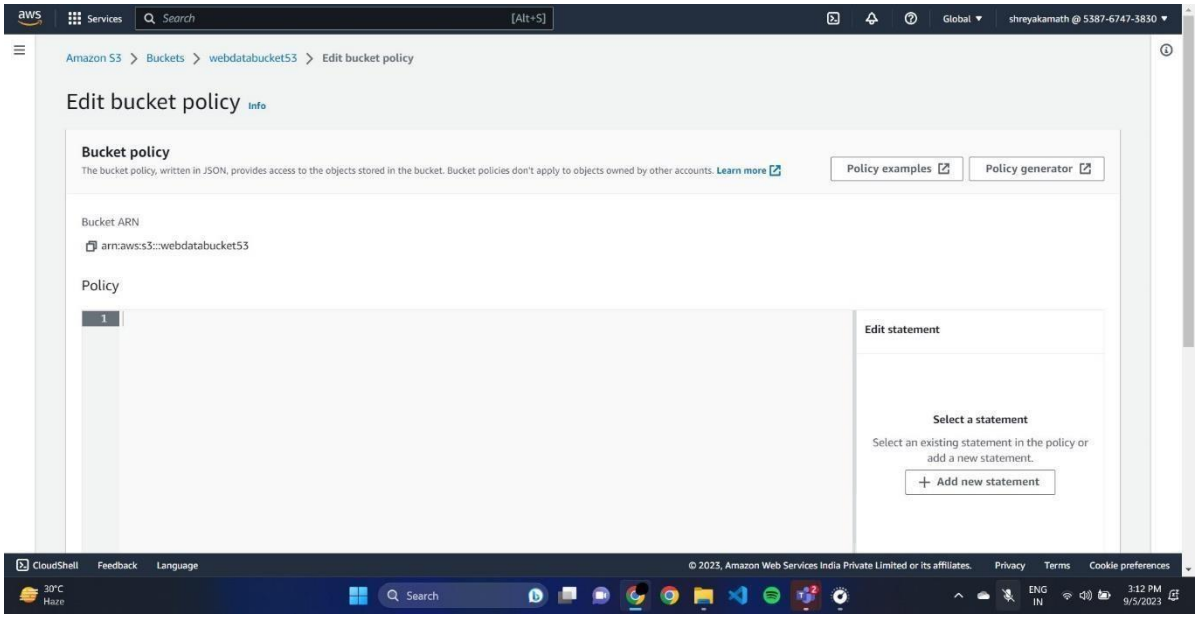

(1) Wha
ChatGP
Adv Dev
S3 Mana
s3.console.aws.amazon.com/s3/get-started?region=eu-...
aws
Services
Globa
shreyakamath @ 5387-6747-383
Storage
Amazon S3
Store and retrieve any amount of data from anywhere
Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.
Create a bucket
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.
Create bucket
Feedback
Language
Privacy
Terms
Cookie preferences
© 2023, Amazon Web Services India Private Limited or its affiliates.

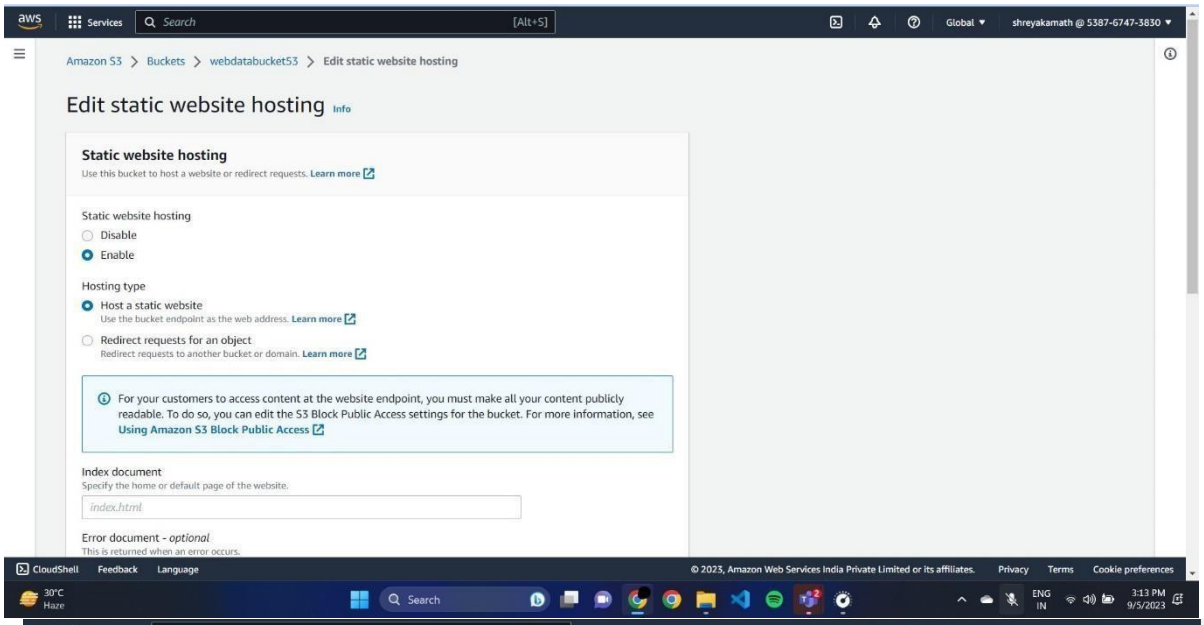

aws
Services
Globa
shreyakamath @ 5387-6747-383
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
Bucket name
webdatabucket
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
AWS Region
Europe (Stockholm) eu-north-1
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
Feedback
Language
Privacy
Terms
Cookie preferences
© 2023, Amazon Web Services India Private Limited or its affiliates.

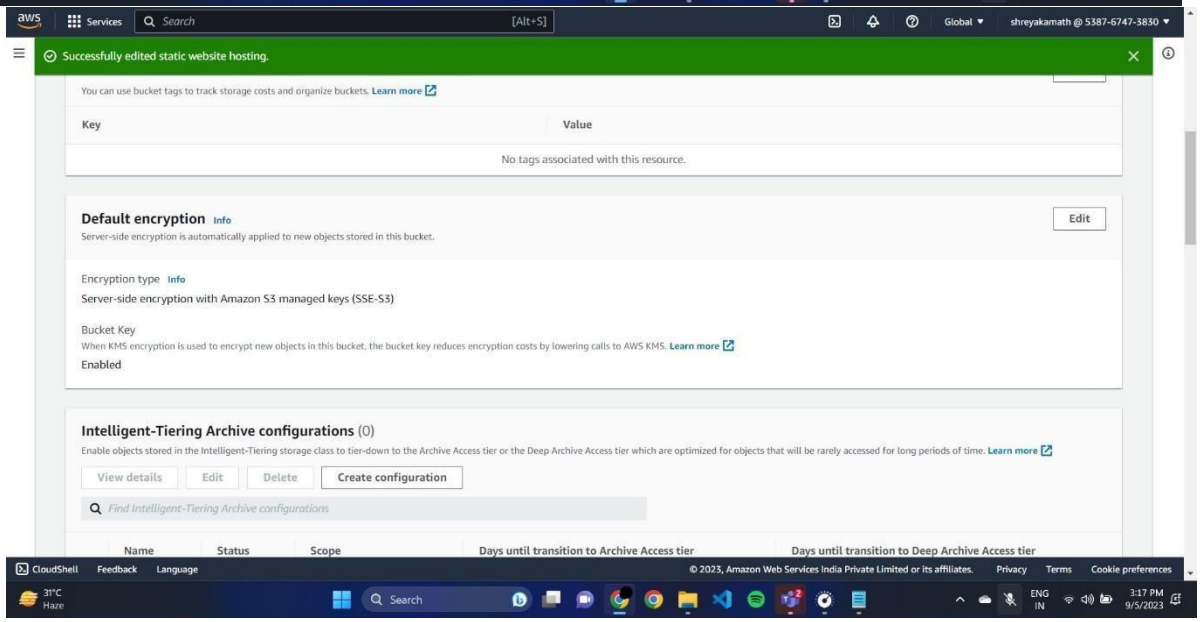

aws
Services
Globa
shreyakamath @ 5387-6747-383
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies

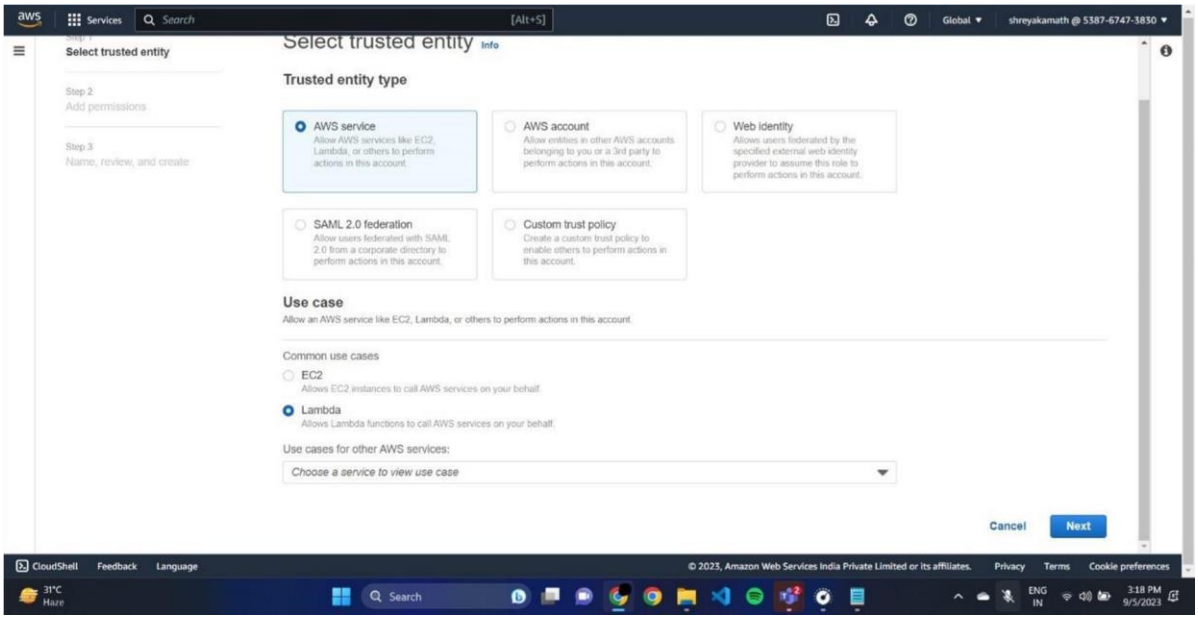

aws
Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
Bucket name
webdatabucket53
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
AWS Region
Europe (Stockholm) eu-north-1
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
ACLs enabled
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
30°C
Haze
Search
ENG
IN
3:10 PM
9/5/2023

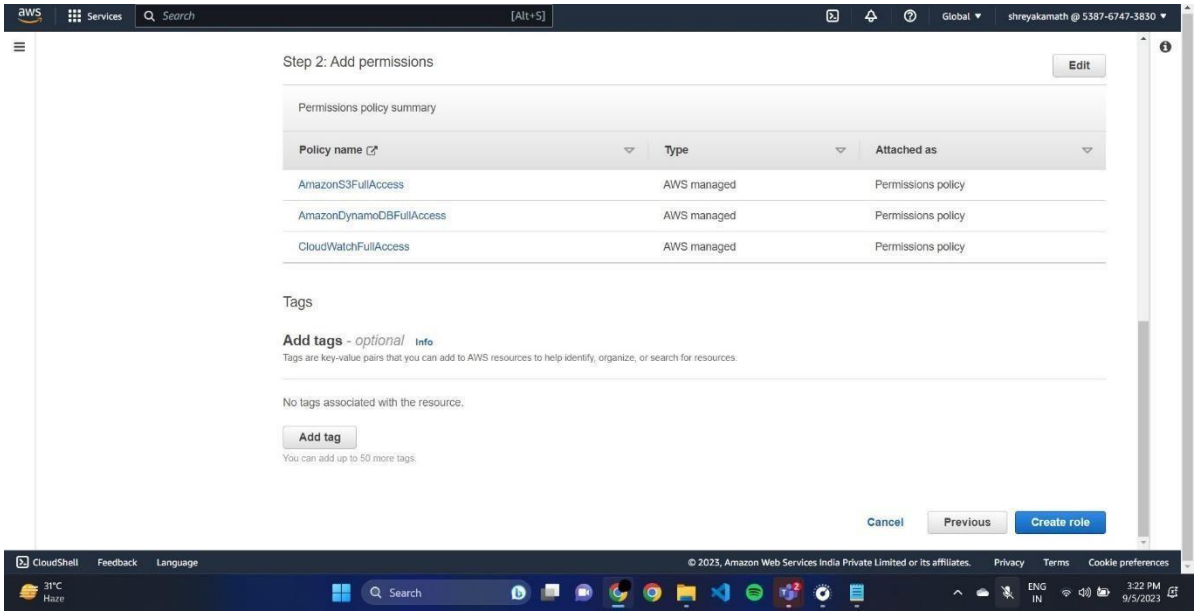

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Successfully created bucket "webdatabucket53"
To upload files and folders, or to configure additional bucket settings choose View details.
View details
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Total storage
Pending
Object count
Pending
Average object size
Pending
You can enable advanced metrics in the "default-account-dashboard" configuration.
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
webdatabucket53
Europe (Stockholm) eu-north-1
Bucket and objects not public
September 5, 2023, 15:10:35 (UTC+05:30)
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
30°C Haze
Search
ENG IN
3:10 PM 9/5/2023

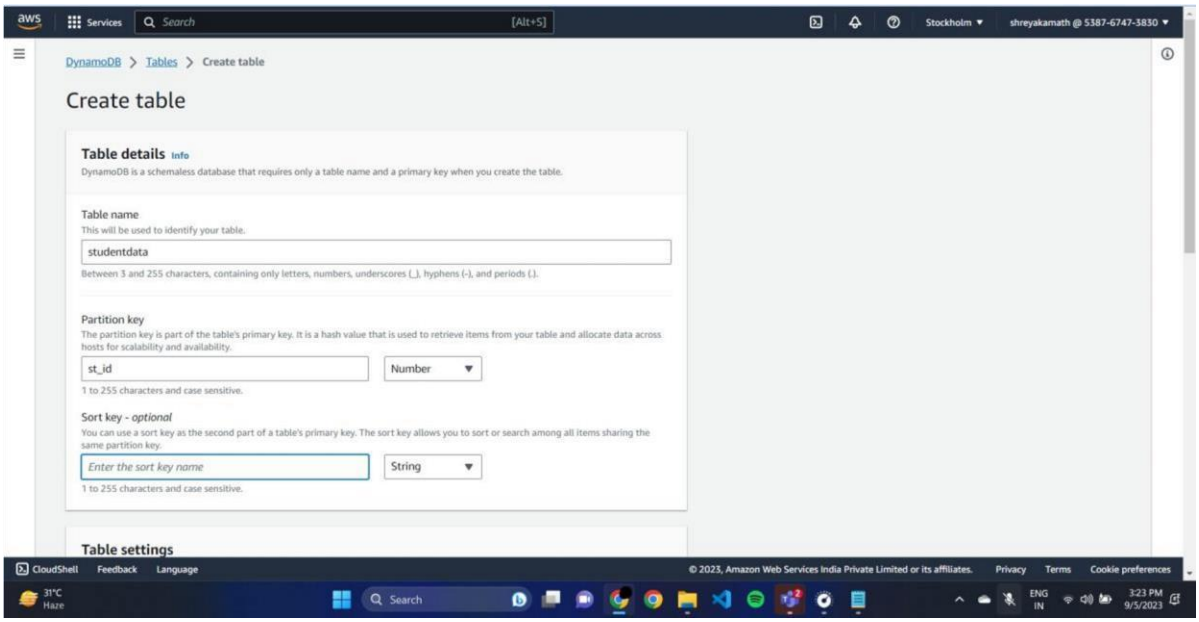

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Amazon S3 > Buckets > webdatabucket53 > Edit bucket policy
Edit bucket policy Info
Bucket policy
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
Policy examples
Policy generator
Bucket ARN
arn:aws:s3:::webdatabucket53
Policy
1
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
Add new statement
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
30°C Haze
Search
ENG IN
3:12 PM 9/5/2023

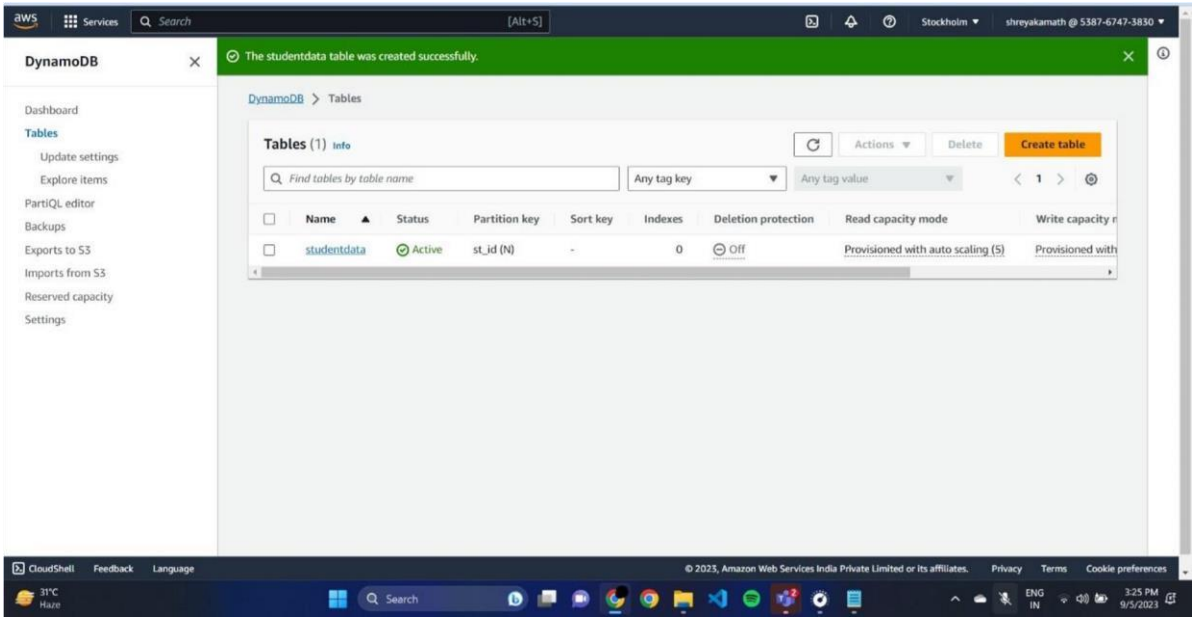

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Amazon S3 > Buckets > webdatabucket53 > Edit static website hosting
Edit static website hosting Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
Index document
Specify the home or default page of the website.
index.html
Error document - optional
This is returned when an error occurs.
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
30°C
Haze
Search
ENG
IN
3:13 PM
9/5/2023

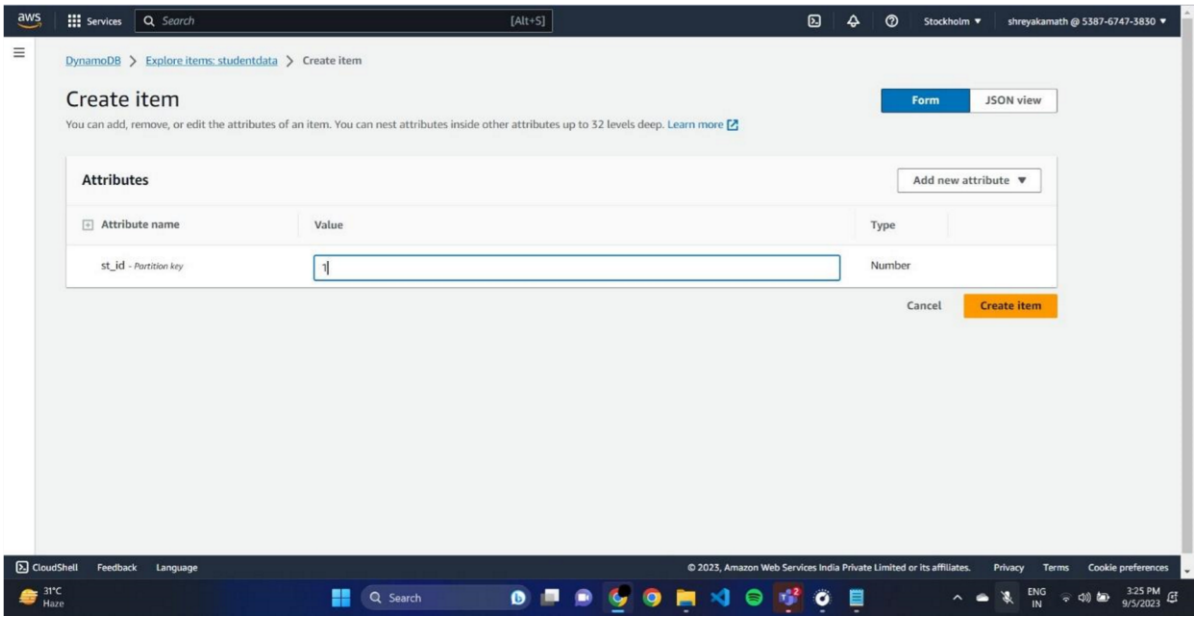

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Successfully edited static website hosting.
You can use bucket tags to track storage costs and organize buckets. Learn more
Key
Value
No tags associated with this resource.
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Edit
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled
Intelligent-Tiering Archive configurations (0)
Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. Learn more
View details
Edit
Delete
Create configuration
Find Intelligent-Tiering Archive configurations
Name
Status
Scope
Days until transition to Archive Access tier
Days until transition to Deep Archive Access tier
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:17 PM
9/5/2023

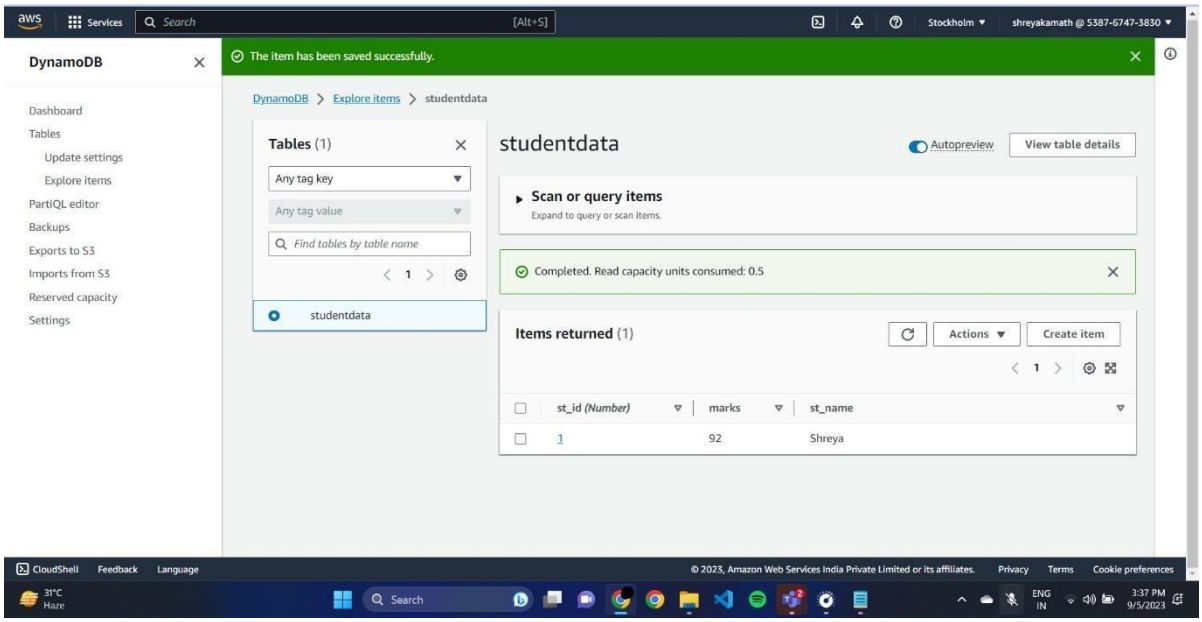

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities 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.
Common use cases
EC2
Allows EC2 instances to call AWS services on your behalf.
Lambda
Allows Lambda functions to call AWS services on your behalf.
Use cases for other AWS services:
Choose a service to view use case
Cancel
Next
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:18 PM
9/5/2023

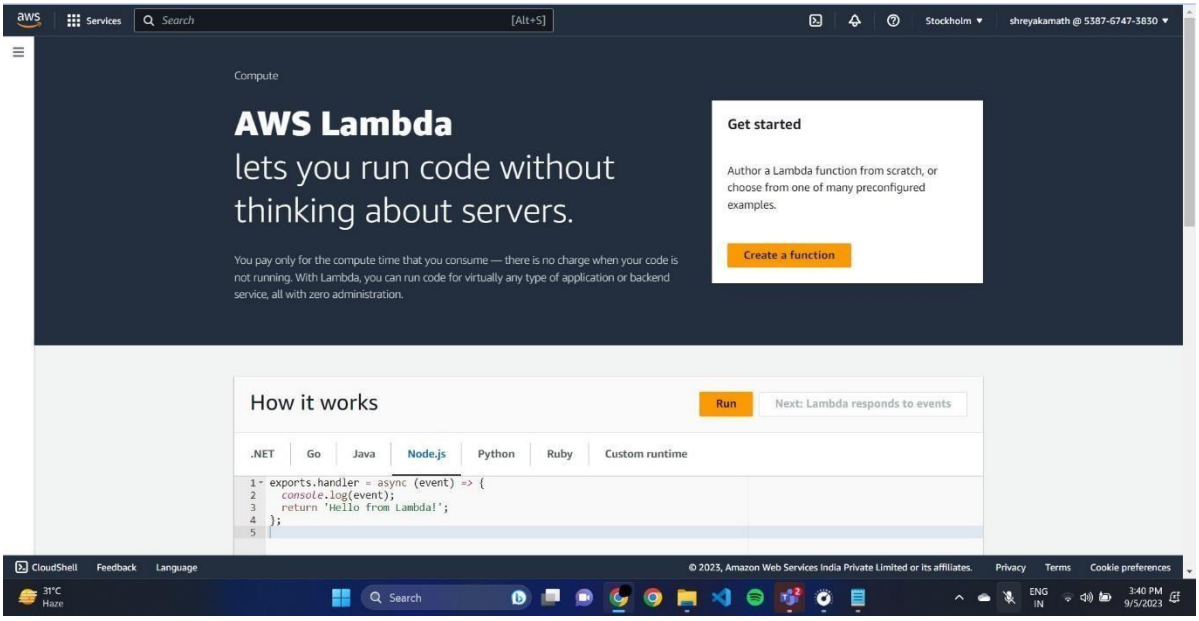

Services
Search
[Alt+S]
Global
shreyakamath @ 5387-6747-3830
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonS3FullAccess
AWS managed
Permissions policy
AmazonDynamoDBFullAccess
AWS managed
Permissions policy
CloudWatchFullAccess
AWS managed
Permissions policy
Tags
Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.
Add tag
You can add up to 50 more tags.
Cancel
Previous
Create role
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:22 PM
9/5/2023

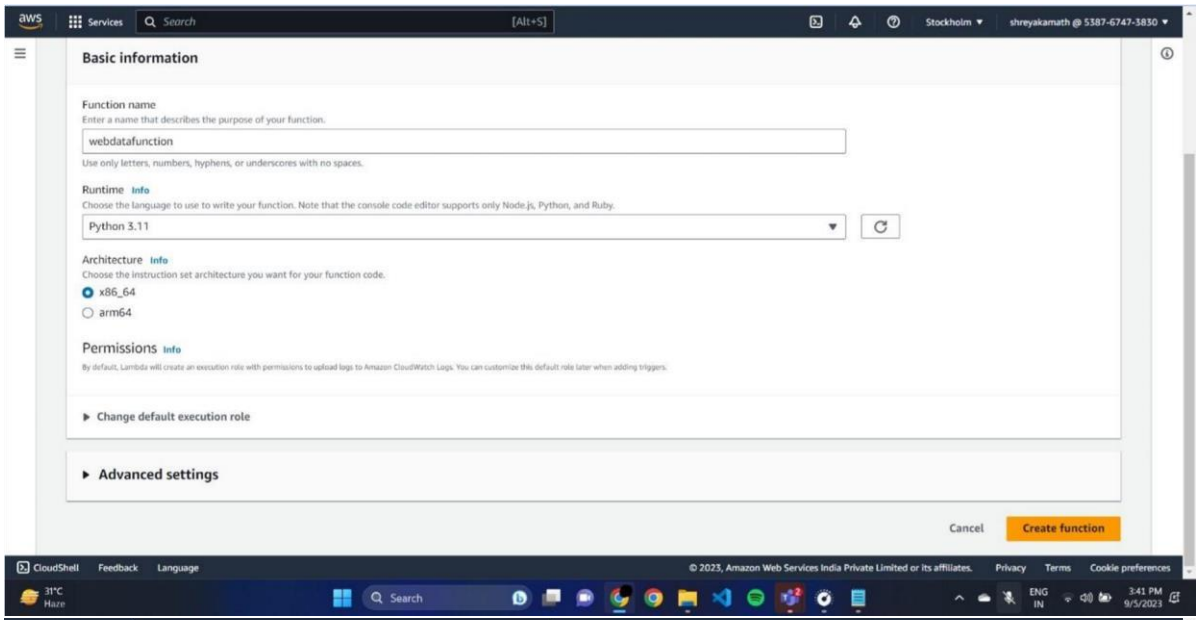

aws
Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
DynamoDB > Tables > Create table
Create table
Table details Info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.
Table name
This will be used to identify your table.
studentdata
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).
Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
st_id
Number
1 to 255 characters and case sensitive.
Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name
String
1 to 255 characters and case sensitive.
Table settings
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:23 PM
9/5/2023

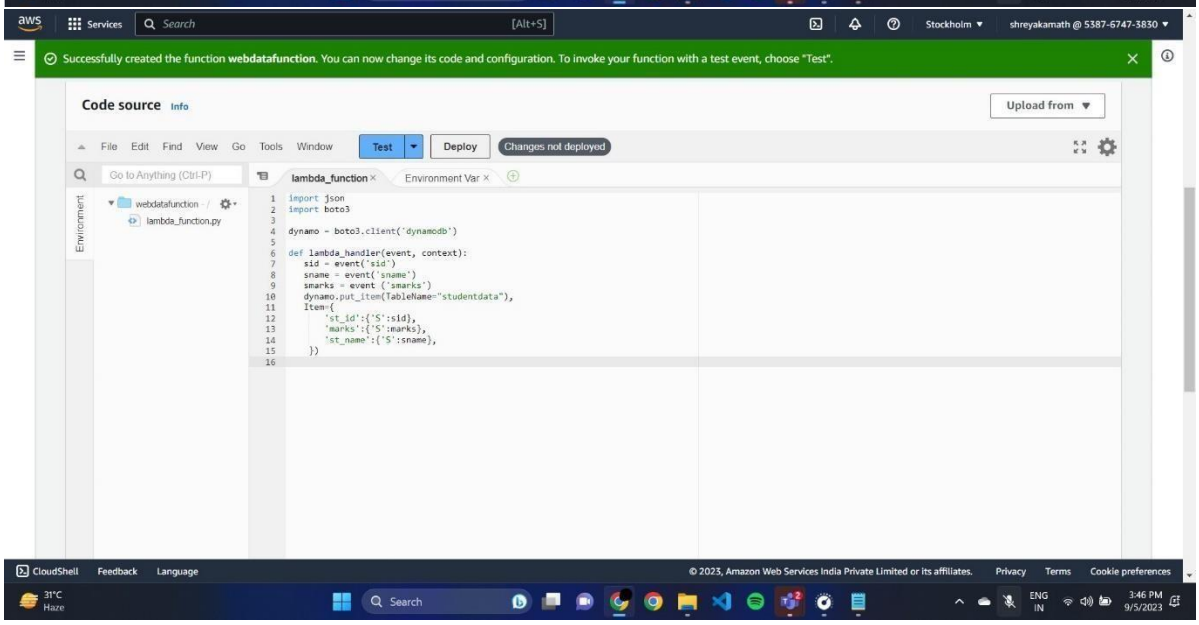

aws
Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
DynamoDB
Dashboard
Tables
Update settings
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Reserved capacity
Settings
The studentdata table was created successfully.
DynamoDB > Tables
Tables (1) Info
Actions
Delete
Create table
Find tables by table name
Any tag key
Any tag value
1
Name
Status
Partition key
Sort key
Indexes
Deletion protection
Read capacity mode
Write capacity
studentdata
Active
st_id (N)
-
0
Off
Provisioned with auto scaling (5)
Provisioned with
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:25 PM
9/5/2023

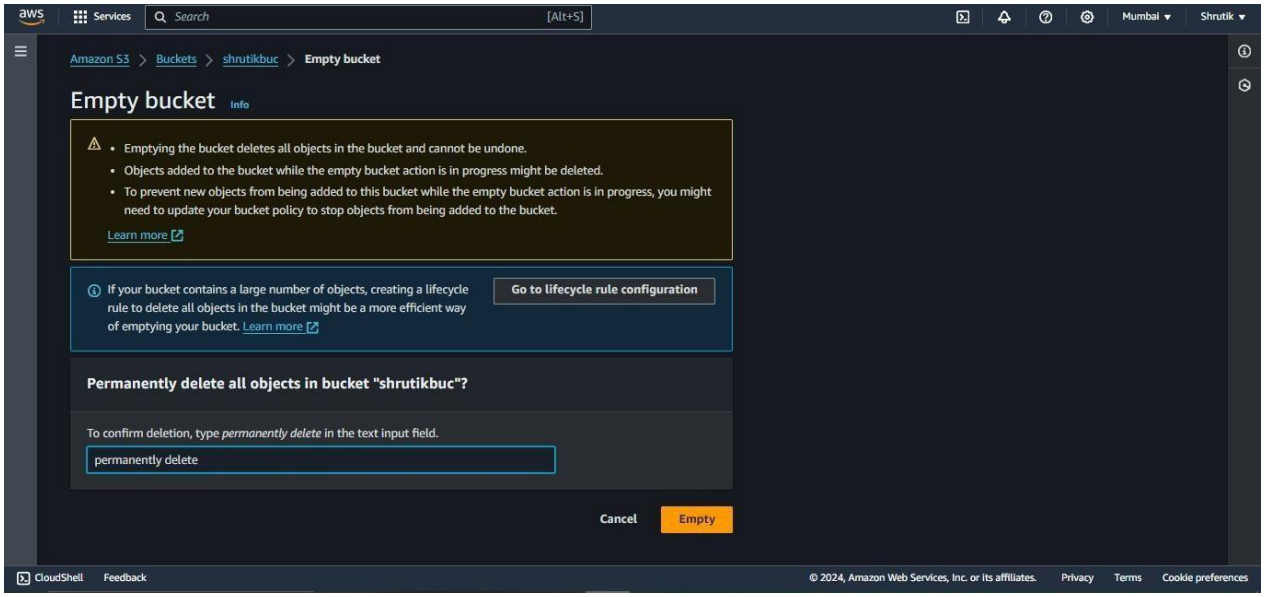

Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
DynamoDB
Explore items: studentdata
Create item
Create item
Form
JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
st_id - Partition key
1
Number
Cancel
Create item
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:25 PM
9/5/2023

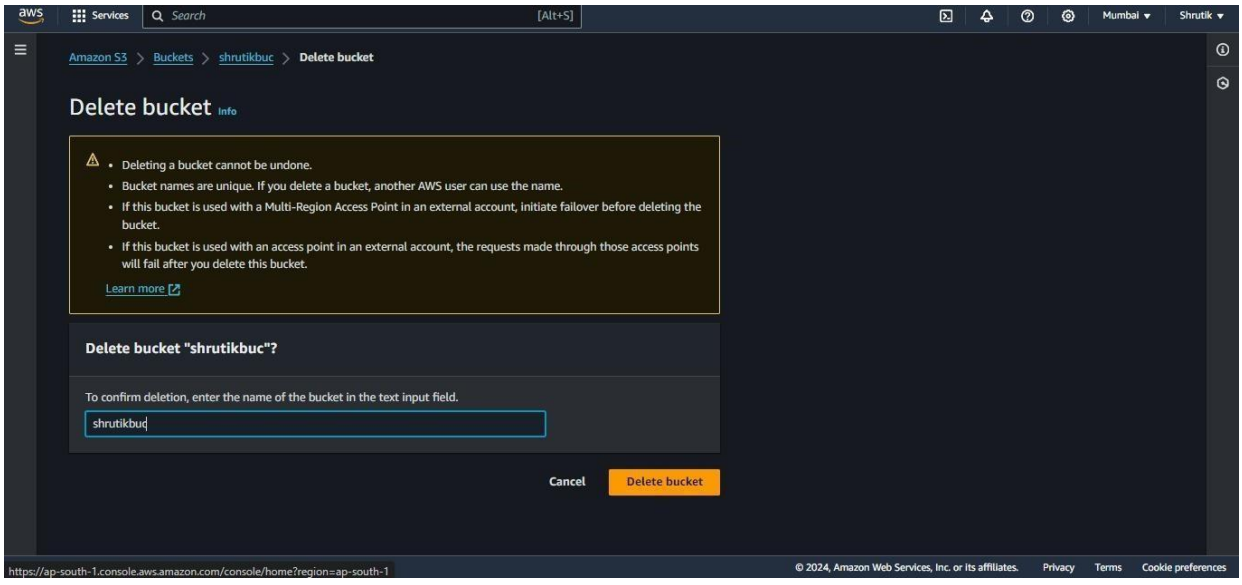

aws
Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
DynamoDB
The item has been saved successfully.
Dashboard
Tables
Update settings
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Reserved capacity
Settings
DynamoDB > Explore items > studentdata
Tables (1)
Any tag key
Any tag value
Find tables by table name
studentdata
studentdata
Autopreview
View table details
Scan or query items
Expand to query or scan items.
Completed. Read capacity units consumed: 0.5
Items returned (1)
Actions
Create item
st_id (Number)
marks
st_name
1
92
Shreya
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:37 PM
9/5/2023


aws
Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
Compute
AWS Lambda
lets you run code without thinking about servers.
You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.
Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
Create a function
How it works
Run
Next: Lambda responds to events
.NET
Go
Java
Node.js
Python
Ruby
Custom runtime
exports.handler = async (event) => {
console.log(event);
return 'Hello from Lambda!';
};
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:40 PM
9/5/2023


Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
Basic information
Function name
Enter a name that describes the purpose of your function.
webdatafunction
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 3.11
Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Advanced settings
Cancel
Create function
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:41 PM
9/5/2023


Services
Search
[Alt+S]
Stockholm
shreyakamath @ 5387-6747-3830
Successfully created the function webdatafunction. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Code source Info
Upload from
File
Edit
Find
View
Go
Tools
Window
Test
Deploy
Changes not deployed
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Environment
webdatafunction
lambda_function.py
import json
import boto3
dynamo = boto3.client('dynamodb')
def lambda_handler(event, context):
sid = event['sid']
sname = event['sname']
smarks = event ['smarks']
dynamo.put_item(TableName="studentdata"),
Item={
'st_id':{'S':sid},
'marks':{'S':marks},
'st_name':{'S':sname},
})
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Haze
Search
ENG
IN
3:46 PM
9/5/2023





Conclusion : Hence, learned & implemented S3 buckets and lambda function

Name : Saikarthik Iyer
Roll no : 41
Batch : T13

Written Assignment No. 1

Q1) What are the best security measures that you can take while using Kubernetes? Ans)
A product follows industry-specific regulations by adhering to the laws, standards, and guidelines set by the relevant regulatory authorities. Here are the general steps a company might take:

- Research: Identify the specific regulations that apply to your industry and product. This may involve local, state, national, or international laws.
- Compliance Assessment: Evaluate your product to determine if it meets the regulatory requirements. This may involve testing, documentation, and risk assessments.
- Design and Development: Integrate compliance considerations into the product design and development process. This may include using materials and manufacturing processes that meet regulatory standards.
- Documentation: Create detailed records and documentation to prove compliance. This may include test results, safety assessments, and labeling requirements.
- Testing and Certification: If necessary, send your product to accredited testing laboratories for certification. This is often required for products in highly regulated industries like healthcare or aerospace.
- Quality Control: Implement quality control measures to ensure that products consistently meet regulatory standards during manufacturing.
- Labeling and Documentation: Ensure that the product is labeled appropriately with required information, such as safety warnings, ingredients, and certifications.
- Registration and Reporting: Register the product with relevant authorities and regularly report data if required. This is common in industries like pharmaceuticals.
- Stay Informed: Keep up-to-date with changes in regulations, as they can evolve over time. Non-compliance can lead to fines, recalls, or legal issues.

- Consult Experts: Sometimes, it's beneficial to seek legal or regulatory affairs experts to ensure full compliance with complex regulations.

Remember, the specific steps and requirements can vary significantly depending on the industry and location. It's crucial to work closely with experts and regulatory bodies to navigate the complex regulatory landscape effectively.

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

Ans)

1. Encryption:
   - Data in Transit*: Use encryption protocols like TLS/SSL to protect data as it moves between different components of the DevOps pipeline, such as from development to testing or production.
   - Data at Rest: Encrypt data stored in databases, configuration files, and other repositories to safeguard it from unauthorized access. Use tools like database encryption or file-level encryption.

2. Access Control and Identity Management:
   - Implement robust access controls to limit who can access and modify data throughout the DevOps process. This involves using tools like role-based access control (RBAC) and ensuring that only authorized personnel have access to critical data.
   - Integrate identity and access management (IAM) solutions to manage user identities, permissions, and authentication, ensuring that only authorized individuals can access the DevOps tools and data.

3. Vulnerability Scanning and Automated Testing:
   - Integrate security into the DevOps pipeline by employing automated vulnerability scanning and testing tools. These tools can identify security issues in code, configurations, and dependencies.
   - Implement Continuous Integration and Continuous Deployment (CI/CD) security checks to automatically scan and assess code for vulnerabilities at various stages of the development pipeline. This helps catch security issues early in the development process.

By incorporating these security techniques into the DevOps workflow, you can better protect sensitive data and ensure that security is an integral part of the software development and delivery process.

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

Ans) Exposing a service using Ingress in Kubernetes involves several steps. Ingress is a Kubernetes resource that manages external access to services within a cluster. Here's a high-level overview of the process:

1. Set Up Kubernetes Ingress Controller: You first need to have an Ingress controller running in your Kubernetes cluster. Common controllers include Nginx Ingress, Traefik, and HAProxy Ingress. You can deploy one of these controllers based on your requirements. For example, you can deploy the Nginx Ingress controller using a command like: shell

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.46.0/deploy
/static/provider/cloud/deploy.yaml
```

2. Define Ingress Resource: Create an Ingress resource that defines how you want to expose your service. This resource specifies the rules for routing external traffic to internal services. Here's an example Ingress resource definition:

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, incoming traffic to example.com/app will be forwarded to the my-service service on port 80.

3. Deploy and Expose Your Service: Ensure that your service (e.g., my-service in the example above) is deployed and running in your cluster. You should expose your service as a ClusterIP or NodePort service, depending on your requirements.

4. DNS Configuration: Ensure that the DNS records for the host specified in the Ingress resource (e.g., example.com) point to the IP address of your Kubernetes cluster or the Load Balancer, if you are using one.

5. Test and Access: After the Ingress resource is created and DNS is configured, you should be able to access your service externally by visiting the specified host and path (e.g., http://example.com/app).

6. TLS/SSL Configuration (Optional): If you want to secure your Ingress using TLS/SSL, you can configure a TLS secret and add it to your Ingress resource.

Here's an example TLS configuration in your Ingress resource:

yaml spec: tls:
- hosts:
- example.com secretName: my-tls-secret

The my-tls-secret should be a Kubernetes Secret containing your TLS certificate and private key.

These steps outline the process of exposing a service using Ingress in Kubernetes. Keep in mind that specifics may vary depending on the Ingress controller and cloud provider you are using, so refer to their documentation for detailed configuration options.

Q.4 Which service protocols does Kubernetes ingress expose?

Ans) Kubernetes Ingress exposes services using HTTP and HTTPS protocols. Ingress is primarily designed for routing external HTTP and HTTPS traffic to services within a Kubernetes cluster based on rules defined in the Ingress resource.

The key features of Ingress include host and path-based routing, SSL termination, and other capabilities related to HTTP and HTTPS traffic. Ingress controllers like Nginx Ingress, Traefik, and others handle the configuration and management of these rules, making it easier to manage and secure external access to services in a Kubernetes cluster.

While Ingress primarily focuses on HTTP and HTTPS, if you need to expose services using other protocols, you might consider different solutions, such as NodePort or LoadBalancer services for protocols like TCP or UDP. These services can provide low-level network access to your pods without the advanced HTTP routing features provided by Ingress.

Name : Saikarthik Iyer
Roll no : 41
Batch : T13

Written Assignment No. 2

Q1) How to deploy Lambda function on AWS?

Ans) Deploying a Lambda function on AWS involves several steps. Lambda is a serverless computing service that allows you to run code in response to events, and it can be triggered by various AWS services or HTTP requests. Here's a general guide on how to deploy a Lambda function:

Create a Lambda Function:

1. Sign in to the AWS Management Console: Go to the AWS Management Console andlog in to your AWS account.
2. Open Lambda Service: Navigate to the Lambda service by searching for "Lambda"in the AWS Console.
3. Create Function: Click on the "Create function" button.
4. Select Author from Scratch: Choose to create a new function from scratch.
5. Basic Information: Provide a name for your function, choose the runtime (e.g.,Python, Node.js, Java), and select an execution role with necessary permissions.
6. Function Code: You can either upload a .zip file with your code or edit it in theLambda editor.
7. Handler: Specify the handler for your function. It's in the format filename.handler, where filename is the name of your script and handler is the name of the function that will be called when your Lambda is triggered.
8. Basic Settings: Configure memory, timeout, and VPC settings for your function.
9. Environment Variables (Optional): Set environment variables if your Lambdafunction relies on them.
10. Tags (Optional): Add tags to your Lambda function for better organization.
11. Execution role: Ensure the execution role has permissions to access any AWSservices or resources your Lambda function needs.

12. Advanced settings (Optional): Configure settings like concurrency, reserved concurrency, and error handling.
13. Triggers: Add triggers to your Lambda function if you want it to be invoked byspecific events (e.g., S3 uploads, API Gateway requests).
14. Review: Review your function configuration and click "Create function." Test Your Function:

After creating the function, you can test it within the Lambda Console by configuring test events. This is useful for verifying that your function works as expected.
Set Up API Gateway (Optional):
If you want to expose your Lambda function as an HTTP API, you can set up AmazonAPI Gateway to create RESTful APIs or HTTP endpoints.

Deploy Your Function:
To deploy your Lambda function, you don't need to perform a separate deploymentstep like traditional application deployments. AWS Lambda automatically managesthe deployment and scaling of your function.

Monitoring and Logging:
Configure CloudWatch Logs to monitor and log the execution of your Lambdafunction. You can set up custom CloudWatch Alarms and metrics to track performance and troubleshoot issues.

Versioning and Aliases (Optional)
You can create versions and aliases for your Lambda functions to maintain differentversions and promote them to production as needed.

Security and Access Control:
Use AWS Identity and Access Management (IAM) to manage permissions and securitysettings for your Lambda function. This ensures that your function has the right level of access to AWS resources.

Scaling and Optimization:
AWS Lambda automatically scales your function based on the number of incomingrequests. You can fine-tune your function's performance and optimize costs usingsettings like memory allocation and concurrency.

Monitoring and Error Handling:
Regularly monitor your Lambda function's performance, set up alarms, and implement error handling to deal with exceptions and issues that may arise duringexecution.

Integration with Other AWS Services:

Integrate your Lambda function with other AWS services to build powerful serverless applications. For example, you can connect it to an S3 bucket, an SNS topic, or a DynamoDB table to perform specific tasks.

Once your Lambda function is deployed and properly configured, it's ready to be triggered by events or HTTP requests as needed. AWS Lambda takes care of the underlying infrastructure, ensuring that your code is executed reliably and efficiently.

Q2) What are the deployment options for AWS Lambda?
Ans) AWS Lambda offers several deployment options to manage the deployment ofyour serverless functions and applications. These options are designed to accommodate different development and deployment scenarios. Here are the main deployment options for AWS Lambda:

Direct Deployment from AWS Management Console:
You can create, configure, and deploy Lambda functions directly from the AWS Management Console using the Lambda service. This is suitable for simple use casesand quick prototypes.
AWS Command Line Interface (CLI):

The AWS CLI allows you to create, package, and deploy Lambda functions from your local development environment. You can use the aws lambda create-function and awslambda update-function-code commands to deploy your functions.
AWS Serverless Application Model (AWS SAM):
AWS SAM is an open-source framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way to define the Amazon APIGateway APIs, AWS Lambda functions, and Amazon DynamoDB tables needed by your serverless application. You can package and deploy your serverless application using the sam deploycommand.
AWS CloudFormation:

AWS CloudFormation is a service that allows you to define your infrastructure as code. You can use CloudFormation templates to specify your Lambda functions andtheir associated resources. This is particularly useful for managing more complex deployments and infrastructure.

Serverless Framework:
The Serverless Framework is an open-source framework for building serverless applications. It simplifies the deployment and management of Lambda functions, APIs, and other AWS resources. You define your serverless application in a serverless.yml file and use the Serverless Framework CLI to deploy your application.

Continuous Integration/Continuous Deployment (CI/CD):

Many organizations integrate AWS Lambda deployments into their CI/CD pipelines.This involves automatically building and deploying Lambda functions using CI/CD tools like Jenkins, Travis CI, CircleCI, and AWS CodePipeline.

Amazon CodeDeploy:
Amazon CodeDeploy is a deployment service that automates application deploymentsto Amazon EC2 instances, on-premises instances, and Lambda functions. You can use CodeDeploy to manage Lambda function deployments and coordinate updates across multiple functions in a release.
Third-Party Deployment Tools:
There are third-party tools and services designed to simplify Lambda functiondeployment and management. Some popular options include the Serverless Framework, Zappa (for Python), and Claudia.js (for Node.js).
The choice of deployment option depends on your specific use case, preferred development workflow, and whether you need to integrate Lambda function deployments into a broader infrastructure-as-code strategy. For simple tasks, direct deployment from the AWS Management Console or the AWS CLI may be sufficient, but for more complex applications, using AWS SAM, CloudFormation, or a serverlessframework can streamline the deployment process and provide better management and automation capabilities.

Q3) What are the 3 full deployment modes that can be used for AWS?
Ans) In AWS, there are three primary full deployment modes or strategies used fordeploying and managing applications and infrastructure:

Blue-Green Deployment:
Blue-Green deployment is a strategy where you maintain two separate environments, often referred to as the "blue" and "green" environments. At any given time, only one of these environments is in production, while the other is a clone of the production environment, ready for updates and testing.
To deploy updates or changes, you switch traffic from the current "blue" environmentto the "green" environment. This minimizes downtime and allows for easy rollback in case of issues.
AWS Elastic Beanstalk, AWS CodeDeploy, and AWS CodePipeline support blue-green deployments.

Canary Deployment:
Canary deployment is a deployment strategy that involves rolling out changes or updates to a small subset of users or instances before making them available to the entire user base. This approach allows you to monitor the effects of the changes on a limited scale before committing to a full rollout.
AWS Elastic Beanstalk, AWS CodeDeploy, and AWS CodePipeline can be configuredto perform canary deployments.
Rolling Deployment:

Rolling deployment is a strategy where updates are gradually applied to a subset ofinstances or resources while others continue to run the previous version. The process is typically automated, and it progresses iteratively until all instances or resources have been updated.
AWS Elastic Beanstalk, AWS CodeDeploy, and AWS CodePipeline support rollingdeployments. These deployment modes provide flexibility and control over how updates and changes are applied to applications and infrastructure in AWS. The choice of deployment mode depends on your specific requirements, such as the level of risk tolerance, the need for rapid updates, and the ability to monitor changes in a controlled manner. Each of these modes can be configured and managed using various AWS services, making it possible to align your deployment strategy with the needs of your application and organization.

Q4) What are the 3 components of AWS Lambda?
Ans) AWS Lambda is a serverless compute service that allows you to run code in response to events without the need to manage servers. Lambda functions consist ofthree main components:

1. Event Source:

- An event source is the trigger that invokes your Lambda function. It could be various AWS services, custom applications, or external systems that send events toLambda.
- Common event sources include Amazon S3 (e.g., object uploads), Amazon SNS(e.g., notifications), Amazon DynamoDB (e.g., database changes), Amazon API Gateway (e.g., HTTP requests), and more.
- You can configure multiple event sources for a single Lambda function, allowing itto respond to different types of events.

2. Lambda Function Code:

- This is the actual code or script that you want to execute in response to the eventstriggered by the event source.
- Lambda functions can be written in several programming languages, includingNode.js, Python, Java, C#, Ruby, and more.
- You package your code along with any required dependencies and libraries and upload it to AWS Lambda. You also specify the handler function that Lambda shouldinvoke when the event occurs.

3. Execution Role:

- The execution role is an AWS Identity and Access Management (IAM) role that defines what AWS resources your Lambda function can access and what it can do.

- You attach a role to your Lambda function that provides permissions to access other AWS services, such as reading from an S3 bucket, writing to a DynamoDB table,or publishing to an SNS topic.
- The role's permissions should be defined based on the principle of least privilege, ensuring that your Lambda function only has the permissions necessary to perform its specific tasks.

These three components work together to create a serverless execution environment where your code responds to events from various sources without you needing to manage server provisioning or scaling. AWS Lambda automatically handles the scaling, availability, and execution of your code in response to the events generated by the event source.