BLACKBUCKS INTERNSHIP REPORT

STATIC WEBSITE HOSTING FOR BOOKS

SUBMITTED BY

CHITTURI SIVA SATISH KUMAR
(21B91A6111)
BACHALA VENKATA SAI SURESH
(21B91A6103)

UNDER THE GUIDANCE OF MR. AASHU DEV.

Blackbuck Engineers Pvt. Ltd
Road No 36, Jubilee Hills,
Hyderabad

BLACKBUCK INTERNSHIP WORK

Team Members:

- CHITTURI SIVA SATISH KUMAR (21B91A6111)
- BACHALA VENKATA SAI SURESH (21B91A6103)

Title:

STATIC WEBSITE HOSTING FOR BOOKS.

Abstract:

In this project, website is hosted for Bookshop using AWS services such as IAM, EC2, S3. The goal is to create an IAM user with specific permissions, launch an EC2 instance, configure an S3 bucket for static web hosting, and then integrate the S3 bucket's website endpoint with for enhanced performance and scalability. Additionally, an IAM role will be created to grant EC2 instances full access to AWS services, which mirrors the permissions of the IAM user. The final step involves uploading files to the S3 bucket and pushing the S3 bucket's website endpoint for efficient content delivery.

Table Of Contents

Services used:

- IAM (Identity and Access Management)
- EC2 (Elastic Compute Cloud)
- S3 BUCKET

Cloud computing:

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage Compared to traditional on-premises IT, and depending on the cloud servicesyou select, cloud computing helps do the following:

- **Lower IT costs:**

  Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.

- **Improve agility and time-to-value:**

  With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users—specifically developers and data scientists.

- **Scale more easily and cost-effectively:**

  Cloud provides elasticity—instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider's global network to spread your applications closer to users around the world.

The term 'cloud computing' also refers to the technology that makes cloud work. This includes some form of virtualized IT infrastructure—servers, operating system software, networking, and other infrastructure that's abstracted, using special software, so that it can be pooled and divided irrespective of physical hardware boundaries.

For example, a single hardware server can be divided into multiple virtual servers.

Cloud Computing Services:

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-service) are the three most common models of cloud services, and it's not uncommon for an organization to use all three.

IaaS (Infrastructure-as-a-Service):

IaaS provides on-demand access to fundamental computing resources—physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or 'owned' infrastructure and for overbuying resources to accommodate periodic spikes in usage.

In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud.

IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.

PaaS (Platform-as-a-service):

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their data center. Developers simply pick from a menu to 'spin up' servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around containers, a virtualized compute model one step removed from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

SaaS (Software-as-a-Service):

SaaS—also known as cloud-based software or cloud applications—is application software that's hosted in the cloud, and that user's access via a web browser, a dedicated desktop client, or an API that integrates with a desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer 'pay-as-you-go' pricing based on your actual usage. In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:**

  With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.

- **Protection from data loss:**

  Because SaaS stores application data in the cloud with the application, users don't lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today- there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications to powerful enterprise software database and AI (artificial intelligence) software.

Cloud Service Providers:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Oracle
- IBM cloud
- Salesforce

Amazon Web Services:

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay- as-you-go basis. Oftentimes, clients will use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic).

These cloud computing web services provide various services related to networking, computing, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware, and operatingsystems.

One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard disk /SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

Amazon provides select portions of security for subscribers (e.g., physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g., account management, vulnerability scanning, patching). AWS operates for many global geographical regions including seven in North America.

Amazon markets AWS to subscribers as a way of obtaining large-scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. of 2021 Q4, AWS has 33% market share for cloud infrastructure while the next competitors Microsoft Azure and Google Cloud have 21%, and 10% respectively, according to Synergy Group.

Why AWS?

- **Easy to use:**

  AWS is designed to allow application providers, ISVs, and vendors to host your applications quickly and securely – whether an existing application or a new SaaS- based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform**.**

- **Flexible:**

  AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.

- **Cost-effective:**

  You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

- **Reliable:**

  With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com 's multi-billion-dollar online business that has been honed for over a decade**.**

- **Scalable and High performance:**

  Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

- **Secure:**

  Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

List of AWS Services:

Amazon, the preeminent cloud vendor, broke new ground by establishing the first cloud computing service, Amazon EC2, in 2008. AWS offers more solutions and features than any other provider and has free tiers with access to the AWS Console, where users can centrally control their ministrations.

Designed around ease-of-use for various skill sets, AWS is tailored for those unaccustomed to software development utilities. Web applications can be deployed in minutes with AWS facilities, without provisioning servers or writing additional code.

- Amazon EC2 (Elastic Compute Cloud)
- Amazon RDS (Relational Database Services)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon Cognito
- Amazon Glacier
- Amazon SNS (Simple Notification Service)
- Amazon VPC (Virtual Private Cloud)
- Amazon LightSail
- Amazon CloudWatch
- Amazon Cloud9
- Amazon Elastic Beanstalk
- Amazon Code Commit
- Amazon IAM (Identity and Access Management)
- Amazon Inspector
- Amazon Kinesis

- Amazon Dynamo DB
- Amazon Code catalyst
- Amazon Kinesis
- AWS Athena
- AWS Amplify
- AWS Quick sight
- AWS Cloud formation

Amazon EC2:

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com 's cloud- computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providinga web service through which a user can boot an Amazon Machine Image (AMI)to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server- instances as needed, paying by the second for active servers – hence the term "elastic".

EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy. In November 2010, Amazon switched its own retail website platform to EC2 and AWS.

Amazon announced a limited public beta test of EC2 on August 25, 2006, offering access on a first-come, first-served basis. Amazon added two new instance types (Large and Extra-Large) on October 16, 2007. On May 29, 2008, two more types were added, High-CPU Medium and High-CPU Extra Large. There were twelve types of instances available.

Amazon added three new features on March 27, 2008, static IP addresses, availability zones, and user selectable kernels. On August20,2008, Amazon added Elastic Block Store (EBS) This provides persistent storage, a feature that had been lacking since the service was introduced.

Instance types:

Initially, EC2 used Xen virtualization exclusively. However, on November 6, 2017, Amazon announced the new C5 family of instances that were based on a custom architecture around the KVM hypervisor, called Nitro. Each virtual machine, called an "instance", functions as a virtual private server. Amazon sizes instances based on "Elastic Compute Units". The performance of otherwise identical virtual machines may vary. On November 28, 2017, AWS announced a bare-metal instance type offering marking a remarkable departure from exclusively offering virtualized instance types.

As of January 2019, the following instance types were offered:

- General Purpose: A1, T3, T2, M5, M5a, M4, T3a
- Compute Optimized: C5, C5n, C4
- Memory Optimized: R5, R5a, R4, X1e, X1, High Memory, z1d
- Accelerated Computing: P3, P2, G3, F1
- Storage Optimized: H1, I3, D2As of April 2018, the following payment methods by instance were offered:
- On-demand: pay by the hour without commitment.
- Reserved: rent instances with one-time payment receiving discounts on the hourly charge.
- Spot: bid-based service runs the jobs only if the spot price is below the bid specified by bidder. The spot price is claimed to be supply-demand based, however a 2011 study concluded that the price was generally not set to clear the market but was dominated by an undisclosed reserve price.

Amazon RDS:

Amazon Relational Database Service (or) **Amazon RDS** is a distributed relational database service by Amazon Web Services (AWS). It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications. Administration processes like patching the database software, backing up databases and enabling point-in-time recovery are managed automatically. Scaling storage and compute resources can be performed by a single API call to the AWS control plane on-demand. AWS does not offer an SSH connection to the underlying virtual machine as part of the managed service.

Multiple Availability Zone (AZ) Deployment:

In May 2010 Amazon announced Multi-Availability Zone deployment support. Amazon RDS Multi-Availability Zone (AZ) allows users to automatically provision and maintain a synchronous physical or logical "standby" replica, depending on database engine, in a different Availability Zone (independent infrastructure in a physically separate location). Multi-AZ database instance can be developed at creation time or modified to run as a multi-AZ deployment later. Multi-AZ deployments aim to provide enhanced availability and data durability for MySQL, MariaDB, Oracle, PostgreSQL and SQL Server instances and are targeted for production environments. In the event of planned database maintenance or unplanned service disruption, Amazon RDS automatically fails over to the up-to-date standby, allowing database operations to resume without administrative intervention.

Multi-AZ RDS instances are optional and have a cost associated with them. When creating a RDS instance, the user is asked if they would like to use a multi-AZ RDS instance. In Multi- AZ RDS deployments backups are done in the standby instance so I/O activity is not suspended any time, but users may experience elevated latencies for a few minutes during backups.

Read replicas:

Read replicas allow different use cases such as scale in for read-heavy database workloads. There are up to five replicas available for MySQL, MariaDB, and PostgreSQL. Instances use the native, asynchronous replication functionality of their respective database engines. They have no backups configured by default and are accessible and can be used for read scaling.

MySQL and MariaDB read replicas and can be made writeable again since October 2012; PostgreSQL read replicas do not support it. Replicas are done at database instance level and do not support replication at database or table level.

Performance metrics and monitoring:

Performance metrics for Amazon RDS are available from the AWS Management Console or the Amazon CloudWatch API. In December 2015, Amazon announced an optional enhanced monitoring feature that provides an expanded set of metrics for the MySQL, MariaDB, and Aurora database engines.

Amazon VPC:

Amazon Virtual Private Cloud (VPC) is a commercial cloud computing service that provides a virtual private cloud, by provisioning a logically isolated section of Amazon Web Services (AWS) Cloud. Enterprise customers are able to access the Amazon Elastic Compute Cloud (EC2) over an IPsec based virtualprivate network.

Unlike traditional EC2 instances which are allocated internal and external IP numbers by Amazon, the customer can assign IP numbers of their choosing from one or more subnets.

Amazon Web Services launched Amazon Virtual Private Cloud on 26August 2009, which allows the Amazon Elastic Compute Cloud service to be connected to legacy infrastructure over an IPsec VPN. In AWS, the basic VPC isfree to use, with users being charged by usage for additional features.EC2 andRDS instances running in a VPC can also be purchased using Reserved Instances, however will have a limitation on resources being guaranteed[citation needed].

IBM Cloud launched IBM Cloud VPC on 4 June 2019, provides an ability to manage virtual machine-based compute, storage, and networking resources. Pricing for IBM Cloud Virtual Private Cloud is applied separately for internet data transfer, virtual server instances, and block storage used within IBM Cloud VPC.

Google Cloud Platform resources can be provisioned, connected, and isolated in a virtual private cloud (VPC) across all GCP regions. With GCP, VPCs are global resources and subnets within that VPC are regional resources. This allows user to connect zones and regions without the use of additional networking complexity as all data travels, encrypted in transit and at rest, on Google's own global, private network. Identy

resources. This allows users to connect zones and regions without the use of additional networking complexity as all data travels, encrypted in transit and at rest, on Google's own global, private network. Identity management policies and security rules allow for private access to Google's storage, big data, and analytics managed services. VPCs on Google Cloud Platform leveragethe security of Google's data centers.

Amazon S3:

Amazon S3 manages data with an object storage architecture which aims to provide scalability, high availability, and low latency with high durability. The basic storage units of Amazon S3 are objects which are organized into buckets. Each object is identified by a unique, user- assigned key. Buckets can be managed using the console provided by Amazon S3, programmatically with the AWS SDK, or the REST application programming interface.

Objects can be up to five terabytes in size. Requests are authorized using an access control list associated with each object bucket and support versioning which is disabled by default. Since buckets are typically the size of an entire file system mount in other systems, this access control scheme is very coarse-grained. In other words, unique access controls cannot be associated with individual files. [citation needed] Amazon S3 can be used to

replace static web-hosting infrastructure with HTTP client-accessible objects, index document support and error document support.

The Amazon AWS authentication mechanism allows the creation of authenticated URLs, valid for a specified amount of time. Every item in a bucket can also be served as a BitTorrent feed. The Amazon S3 store can act as a seed host for a torrent and any BitTorrent client can retrieve the file. This can drastically reduce the bandwidth cost for the download of popular objects. A bucket can be configured to save HTTP log information to a sibling bucket; this can be used in data mining operations.

There are various User Mode File System (FUSE)–based file systems for Unix-like operating systems (for example, Linux) that can be used to mount an S3 bucket as a file system. The semantics of the Amazon S3 file system is not that of a POSIX file system, so the file system may not behave entirely as expected.

Amazon IAM:

IAM provides the infrastructure necessary to control authentication and authorization for your AWS account. The IAM infrastructure is illustrated by the following diagram.

First, a human user or an application uses their sign-in credentials to authenticate with AWS. Authentication is provided by matching the sign-in credentials to a principal (an IAM user, federated user, IAM role, or application) trusted by the AWS account.

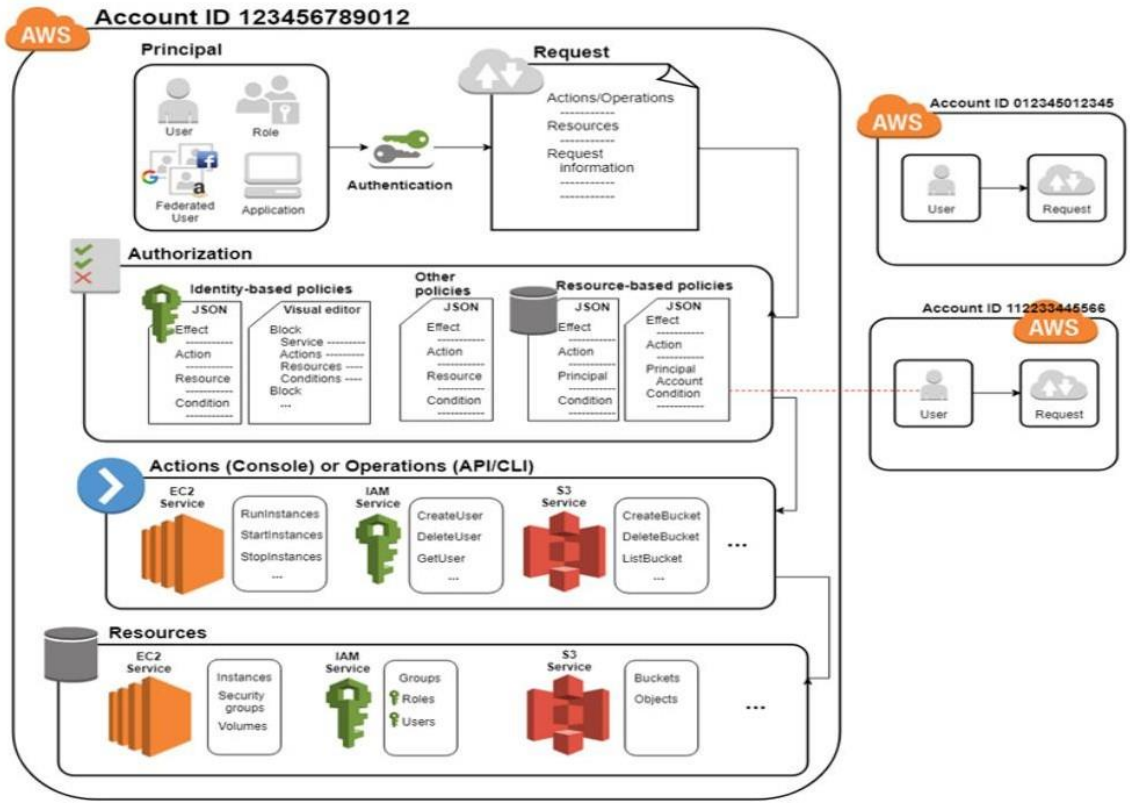
Next, a request is made to grant the principal access to resources. Access is granted in response to an authorization request.

For example, when you first sign into the console and are on the console home page, you are not accessing a specific service. When you select a service, the request for authorization is sent to that service and it looks to see if your identity is on the list of authorized users, what policies are being enforced to

control the level of access granted, and any other policies that might be in effect. Authorization requests can be made by principals within your AWS account or from another AWS account that you trust.

Once authorized, the principal can take action or perform operations on resources in your AWS account. For example, the principal could launch a new Amazon Elastic Compute Cloud instance, modify IAM group membership, or delete Amazon Simple Storage Service buckets.

The previous illustration we used specific terminology to describe how to obtain access to resources. These IAM terms are commonly used when working with AWS:



IAM Resources:

The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

IAM Identities:

The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

IAM Entities:

The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

Principals:

A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. Principals include federated users and assumed roles.

Human users:

Also known as human identities; the people, administrators, developers, operators, and consumers of your applications.

Workload:

A collection of resources and code that delivers business value, such as an application or backend process. Can include applications, operational tools, and components.

Cloud Front:

Amazon CloudFront is a content delivery network (CDN) service provided by Amazon Web Services (AWS). It is designed to deliver content, such as web pages, images, videos, and other assets, to end-users with low latency and high data transfer speeds. CloudFront helps improve the performance, reliability, and scalability of websites and applications by caching content closer to the users, reducing the load on the origin server and improving the overall user experience.

Key features of Amazon CloudFront:

Global Edge Network: CloudFront operates from a vast network of edge locations worldwide, strategically positioned to ensure that content is delivered to users from the nearest edge location, reducing the distance data must travel and minimizing latency.

Content Caching: CloudFront caches content at its edge locations, storing copies of static and dynamic content. This helps reduce the load on the origin server, as the edge locations can directly serve the cached content to users, improving response times.

Dynamic Content Support:

CloudFront can also cache and accelerate dynamic content by using custom caching configurations and server-side techniques, making it suitable for a wide range of applications, including e-commerce and personalized content delivery.

Security and Access Control:

Cloud Front offers various security features, such as SSL/TLS encryption for data in transit, integration with AWS Web Application Firewall (WAF) to protect against common web application attacks, and integration with AWS Identity and Access Management (IAM) for access control.

Origin Shield:

This feature acts as a shield for the origin server by reducing the number of requests directly hitting it, particularly during traffic spikes. The shield (a middle layer) can help to offload the origin, increase cache efficiency, and protect the backend infrastructure.

Real-Time Logs and Analytics:

CloudFront provides access logs and real-time analytics that enable website owners to monitor usage patterns, identify potential issues, and optimize their content delivery strategy.

Integration with Other AWS Services:

CloudFront seamlessly integrates with other AWS services, such as Amazon S3 for storing

Environments and computing resources:

Behind the scenes, there are a couple of ways you can connect your environments to computing resources:

- You can instruct AWS Cloud9 to create an Amazon EC2 instance, and then connect the environment to that newly created EC2 instance. This type of setup is called an EC2 environment.

- You can instruct AWS Cloud9 to connect an environment to an existing cloud compute instance or to your own server. This type of setup is called an SSH environment.

EC2 environments and SSH environments have some similarities and some differences. If you're new to AWS Cloud9, we recommend that you use an EC2 environment because AWS Cloud9 takes care of much of the configuration for you. As you learn more about AWS Cloud9, and want to understand these similarities and differences better, see EC2 environments compared with SSH environments in AWS Cloud9.

Amazon Aurora:

Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora. With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL without requiring changes to most of your existing applications.

Aurora includes a high-performance storage subsystem. Its MySQL- and PostgreSQL- compatible database engines are customized to take advantage of that fast distributed storage. The underlying storage grows automatically as needed. An Aurora cluster volume can grow to a maximum size of 128 tebibytes (TiB). Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration.

Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS). Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. If you are not already familiar with Amazon RDS, see the **Amazon Relational Database Service** User Guide.

The following points illustrate how Amazon Aurora relates to the standard MySQL and PostgreSQL engines available in Amazon RDS:

- You choose Aurora MySQL or Aurora PostgreSQL as the DB engine option when setting up new database servers through Amazon RDS.

- Aurora takes advantage of the familiar Amazon Relational Database Service (Amazon RDS) features for management and administration. Aurora uses the Amazon RDS AWS Management Console interface, AWS CLI commands, and API operations to handle routine database

tasks such as provisioning, patching, backup, recovery, failure detection, and repair.

- Aurora management operations typically involve entire clusters of database servers that are synchronized through replication, instead of individual database instances. The automatic clustering, replication, and storage allocation make it simple and cost- effective to set up, operate, and scale your largest MySQL and PostgreSQL deployments.

- You can bring data from Amazon RDS for MySQL and Amazon RDS for PostgreSQL into Aurora by creating and restoring snapshots, or by setting up one-way replication. You can use push-button migration tools to convert your existing RDS for MySQL and RDS for PostgreSQL applications to Aurora.

AWS Autoscaling:

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in

each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases. instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.

Auto scaling benefits:

Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS Cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

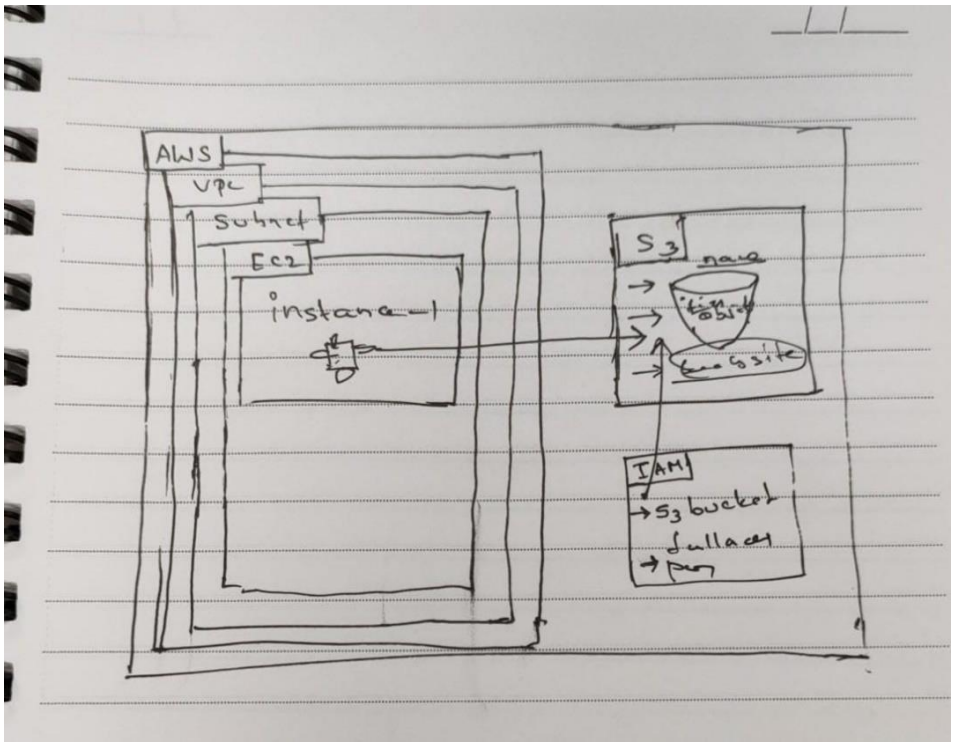
- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.

- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.

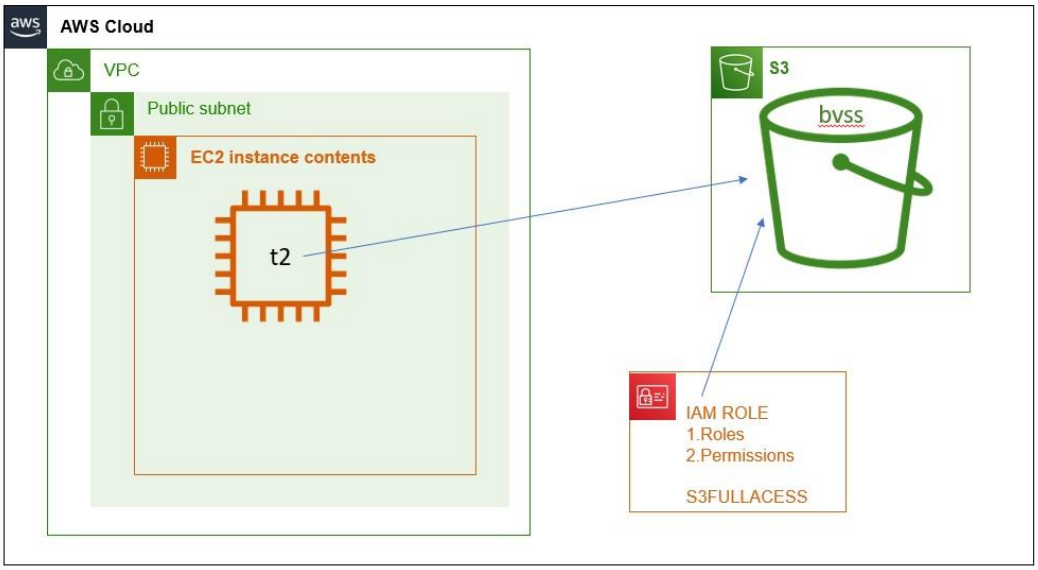
- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

You can use environment variables to adjust your function's behavior without updating code. An environment variable is a pair of strings that is stored in a function's version-specific configuration. The Lambda runtime makes environment variables available to your code and sets additional environment variables that contain information about the function and invocation request.

ROUGH ARCHITECTURE:



FINIAL ARCHITECTURE:



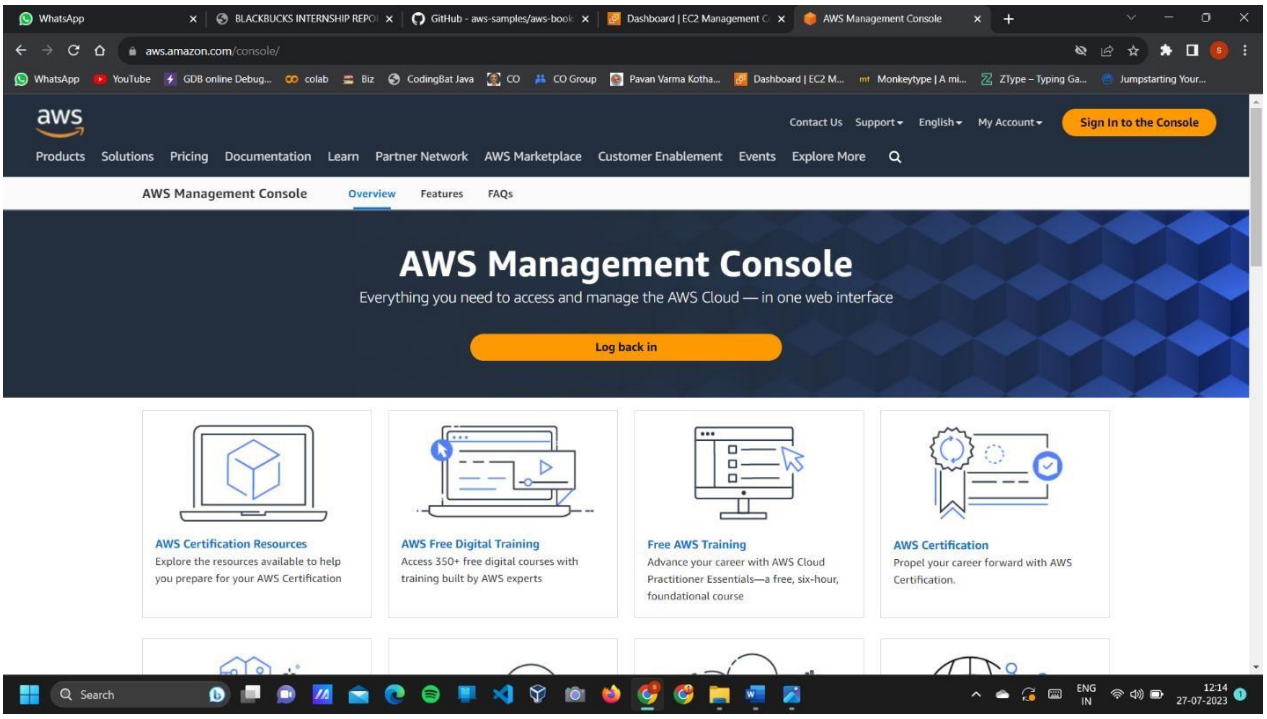
https://bvssbucket.s3.ap-southeast-1.amazonaws.com/index+(1).html

IMPLEMENTATION

Steps to perform:

Step-1: -

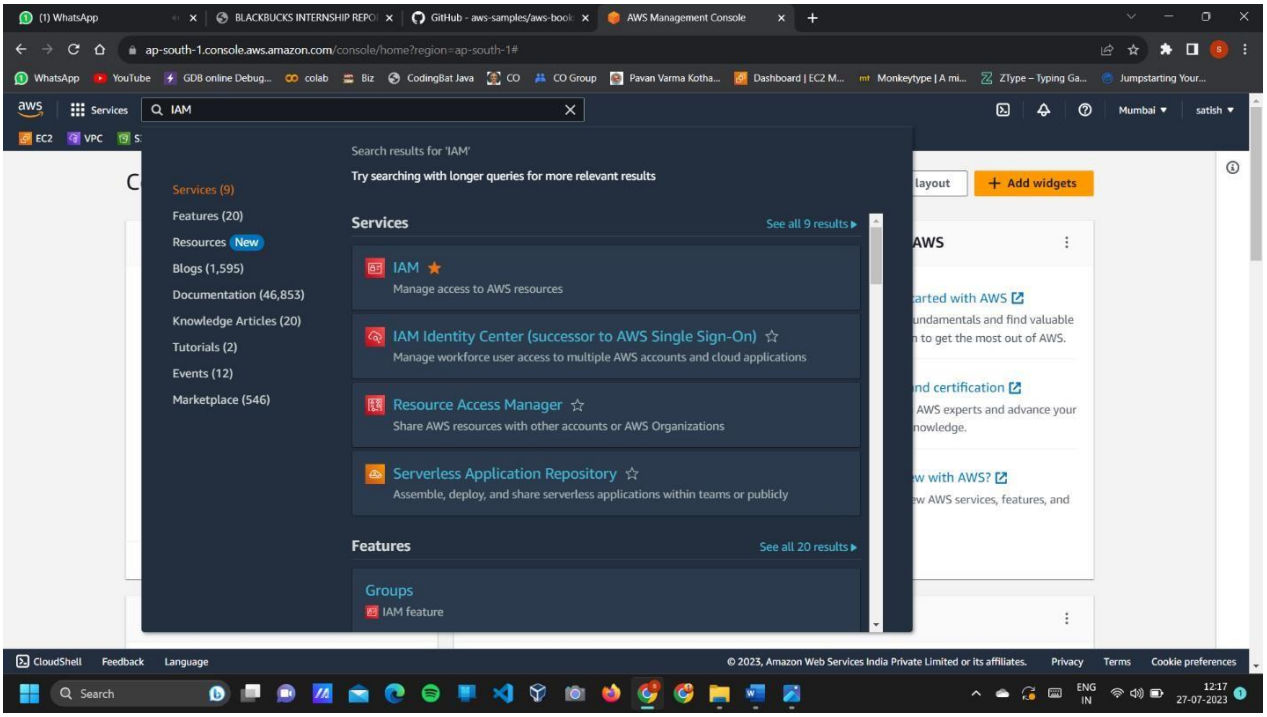
Open your web browser and go to the AWS Management Console.

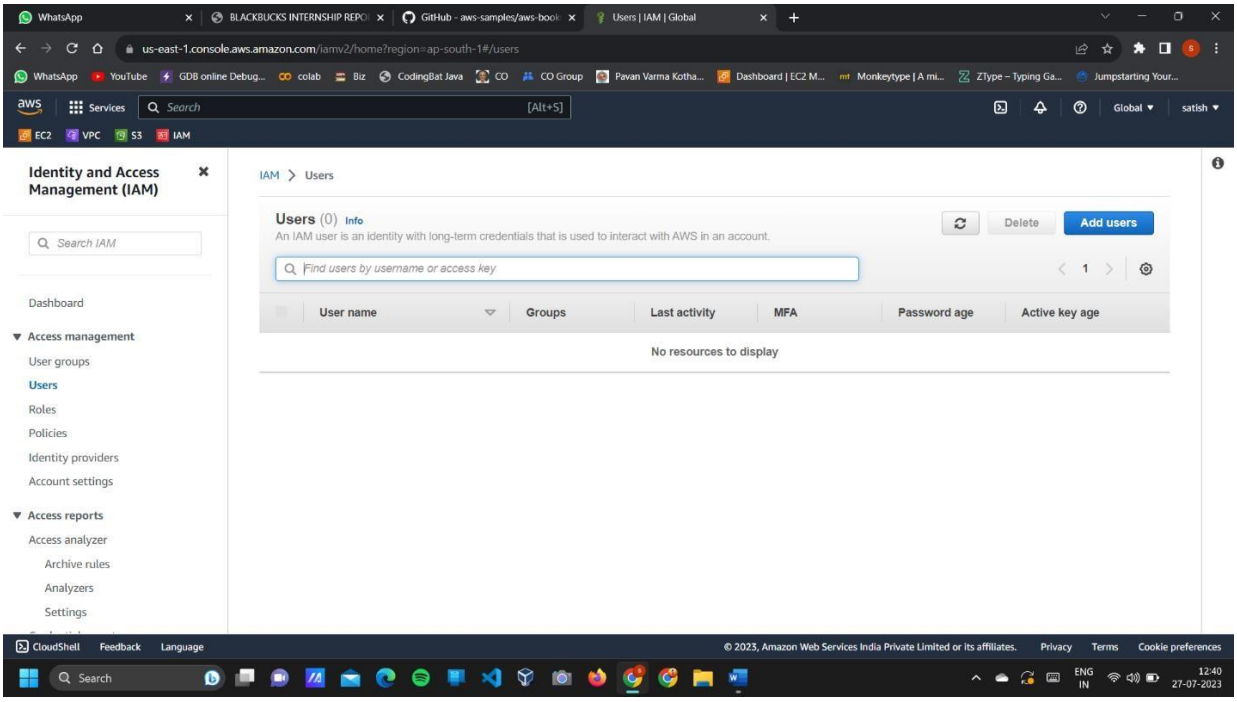


Step-2: -

Access the IAM Dashboard:

Once you're logged in, search for "IAM" in the AWS Management Console search bar or find the "IAM" service under the "Security, Identity & Compliance" section. Click on it to access the IAM dashboard.

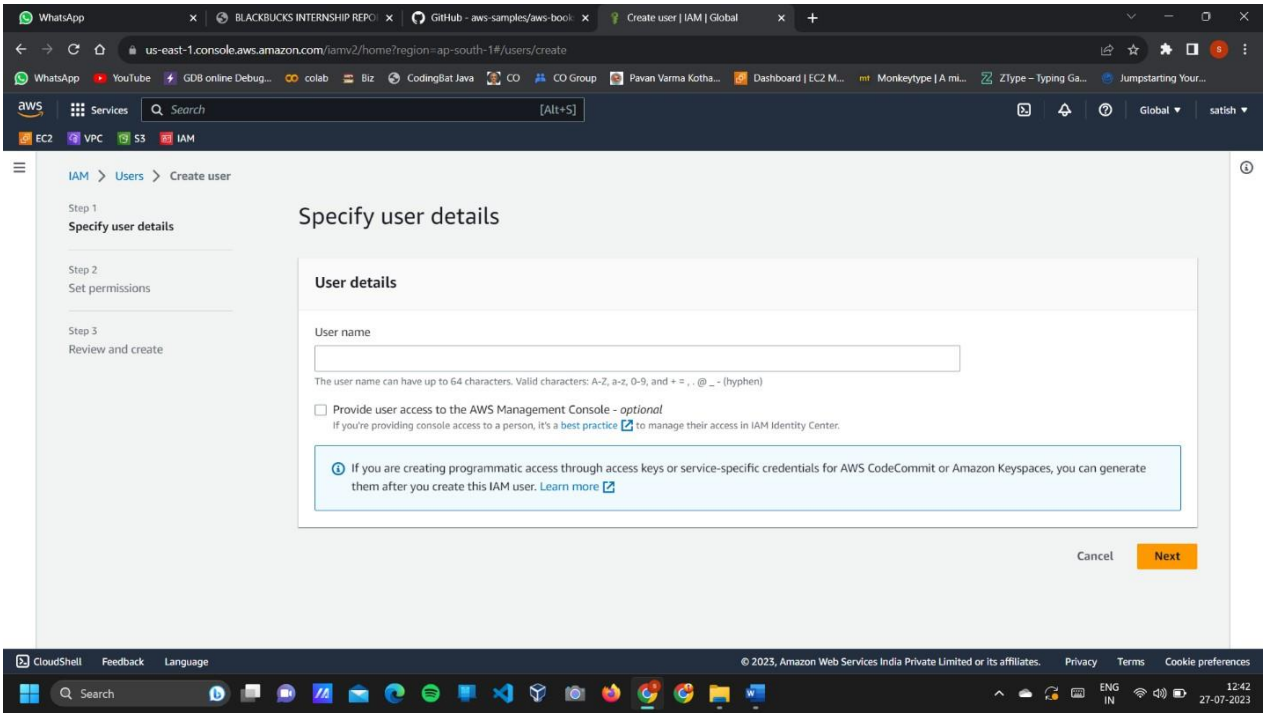

AWS Management Console
ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1#
IAM
Search results for 'IAM'
Try searching with longer queries for more relevant results
Services (9)
Features (20)
Resources New
Blogs (1,595)
Documentation (46,853)
Knowledge Articles (20)
Tutorials (2)
Events (12)
Marketplace (546)
Services
See all 9 results
IAM
Manage access to AWS resources
IAM Identity Center (successor to AWS Single Sign-On)
Manage workforce user access to multiple AWS accounts and cloud applications
Resource Access Manager
Share AWS resources with other accounts or AWS Organizations
Serverless Application Repository
Assemble, deploy, and share serverless applications within teams or publicly
Features
See all 20 results
Groups
IAM feature
Add widgets
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Users | IAM | Global
us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users
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
Settings
IAM > Users
Users (0) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Add users
Find users by username or access key
User name
Groups
Last activity
MFA
Password age
Active key age
No resources to display
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Step-3: -

Navigate to "Users":

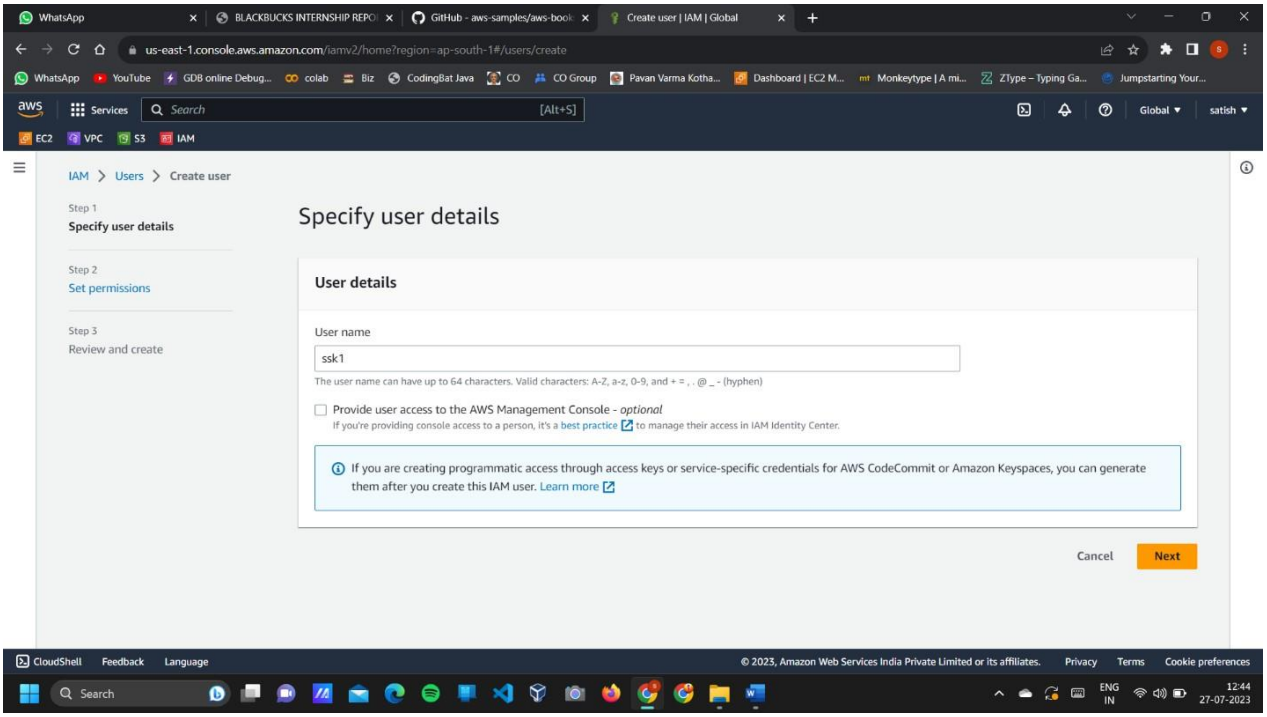
In the IAM dashboard, you'll find the navigation pane on the left side. Click on "Users" under the "Access management" section to go to the IAM users' page. On the IAM users' page, click the "Add user" button to start the user creation process.



Step-4: -

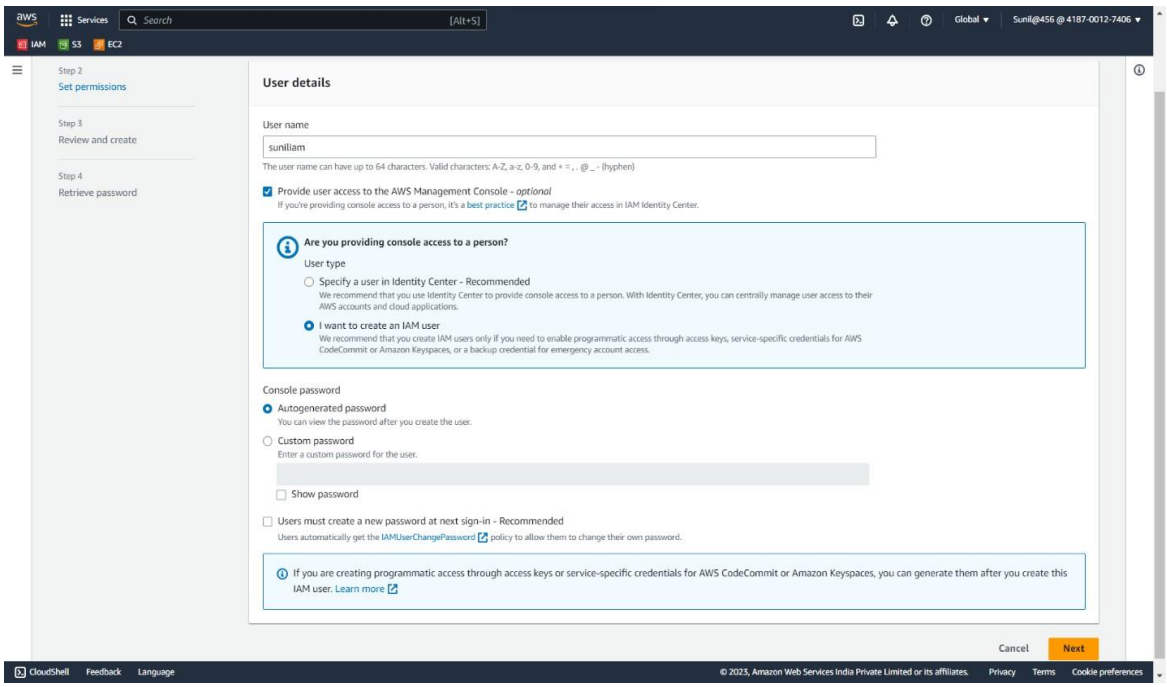
Enter the user details:

In the “Add user” form you’ll need to provide the following information username. you need to choose a unique IAM user name.



Step-5:-Provide user access to the aws management console:

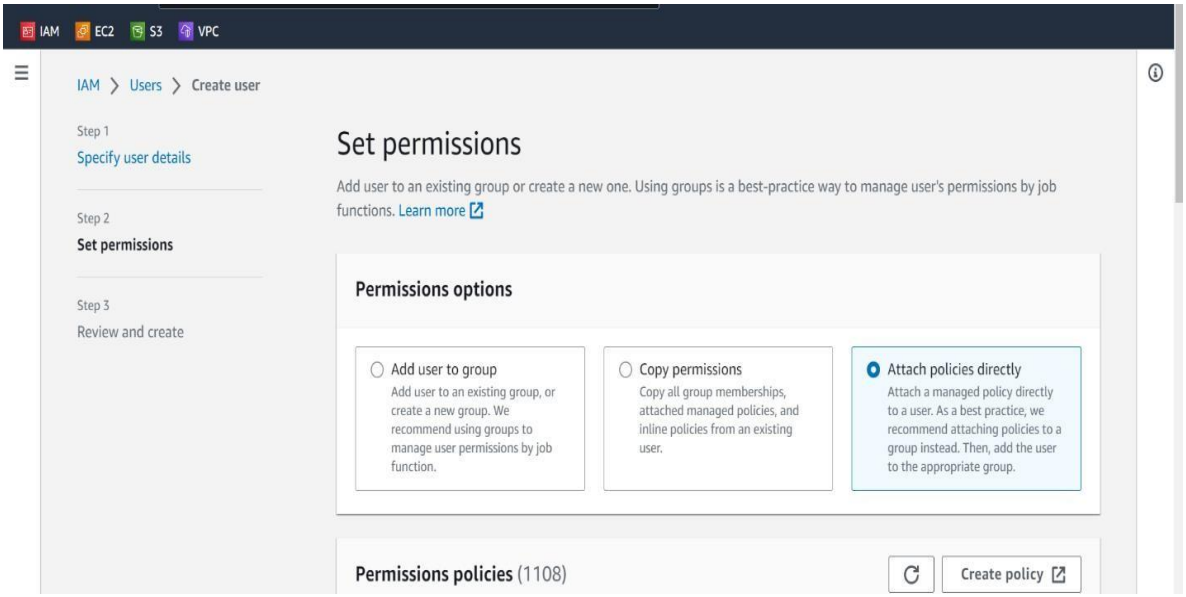
Click on “I want to create an IAM user”. Set passwords,click on “Next” button.



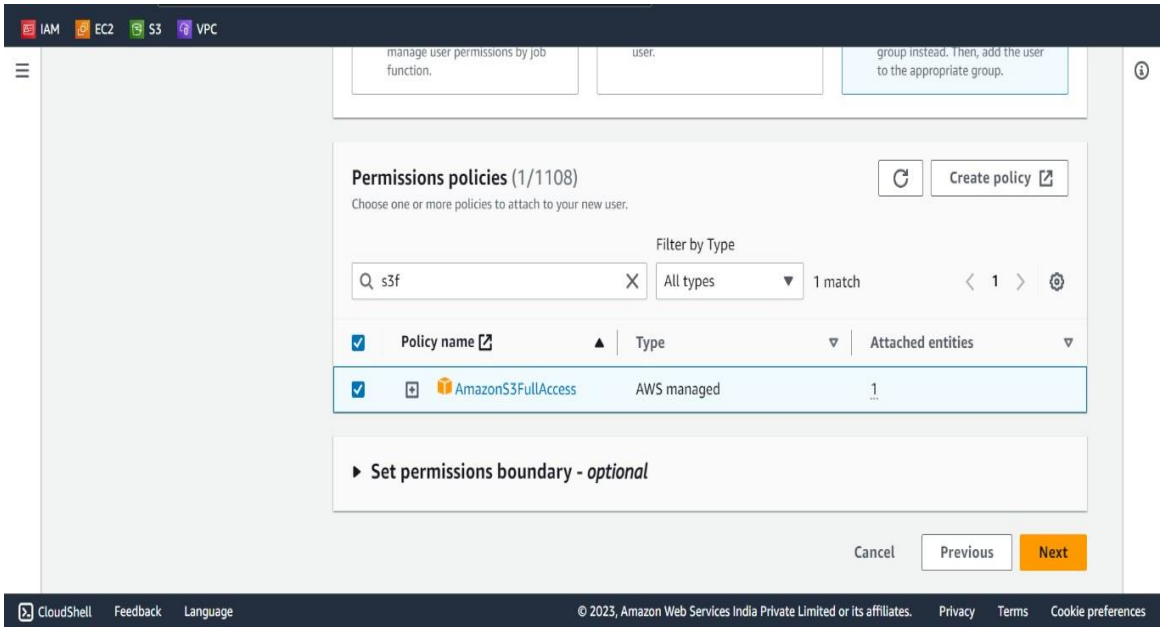
Step-6: -

Set permissions:

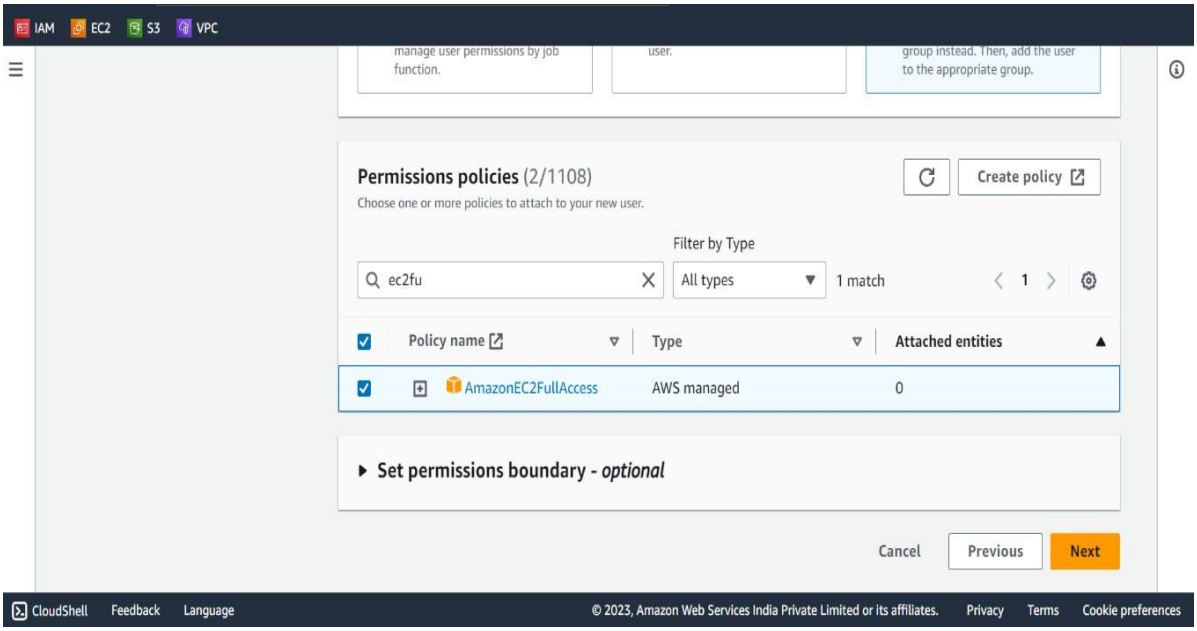
On the next screen, you can attach existing IAM policies to the user to grant specific permissions. AWS provides some predefined policies, or you can create custom policies that suit your requirements. You can also skip this step and attach policies later. in this we are giving four types of policies.



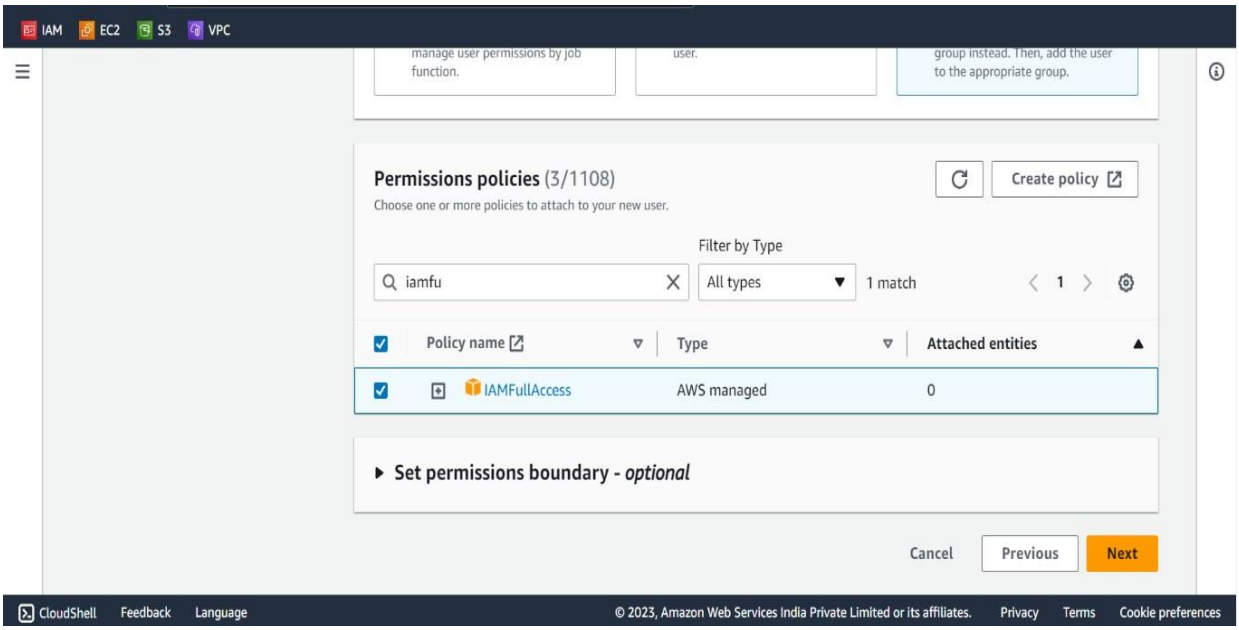
Step-7: - Add AmazonS3FullAccess:



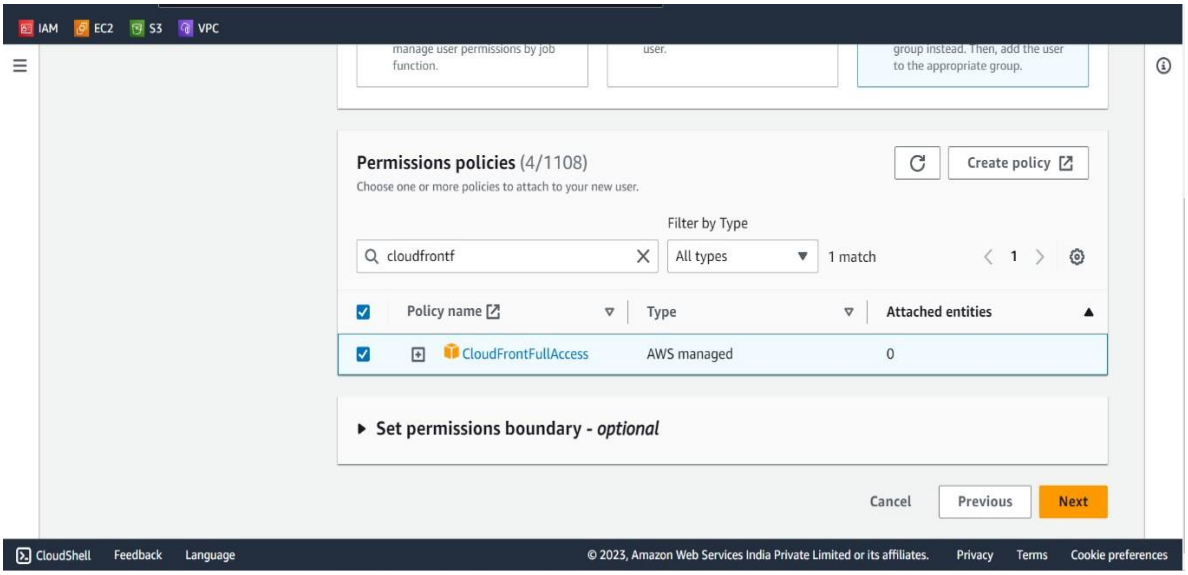
Step-8: - Add AmazonEC2FullAccess:



Step-9: - Add IAM Full Access:

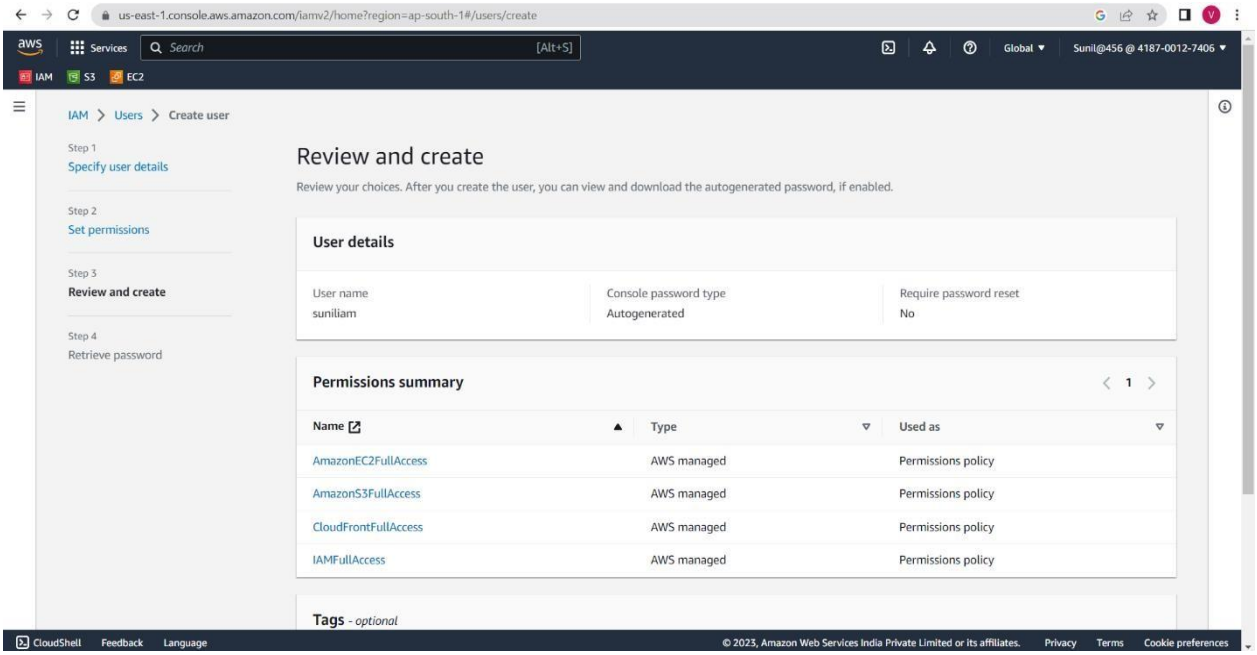


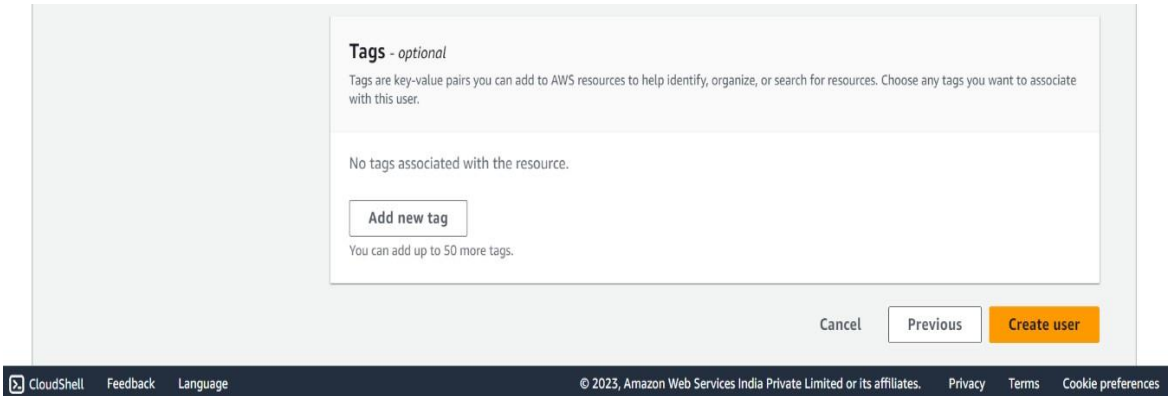
Step-9: - Add Cloud Front Full Access:



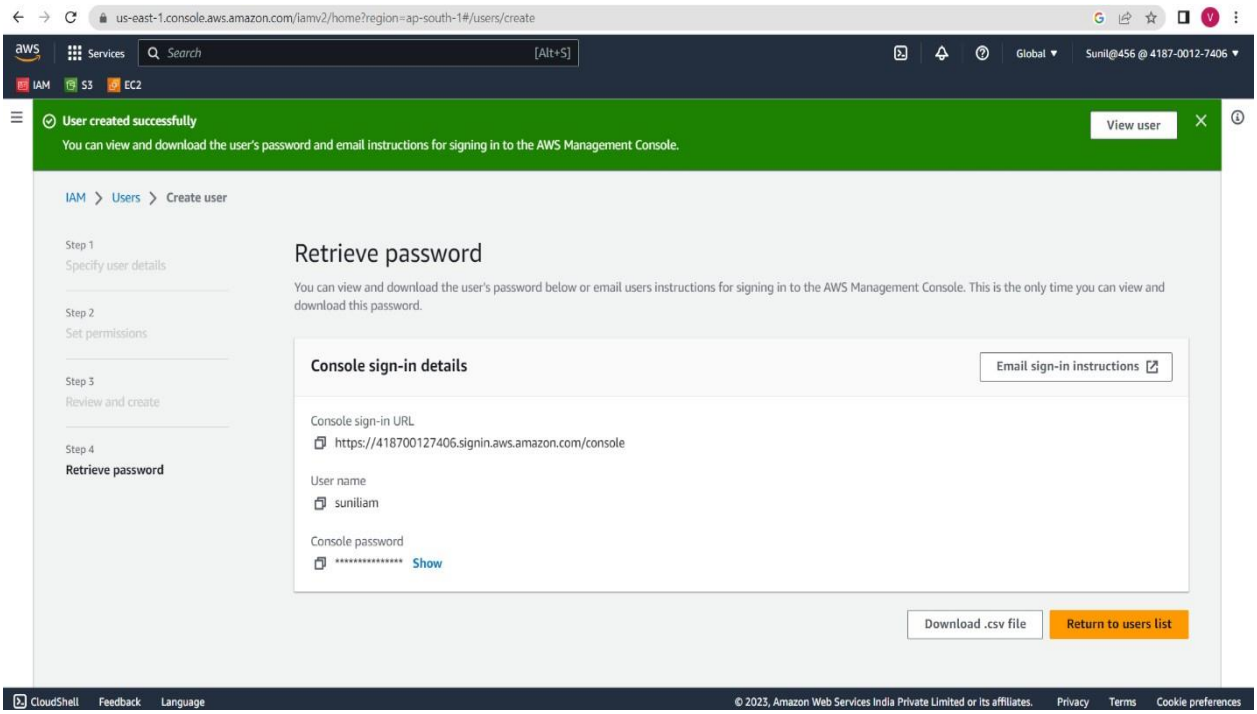
Step-10: - Review and create:

Review the user details, permissions, and other settings to ensure they are correct. If everything looks good, click the "Create user" button.





Step-8. IAM user is created:

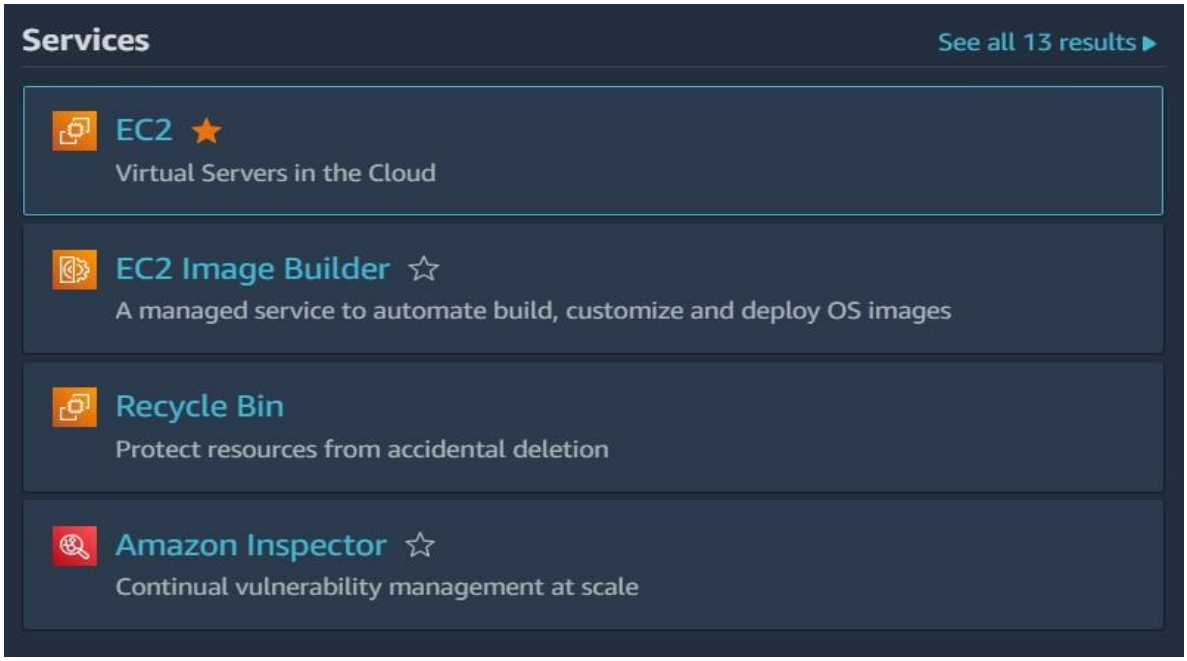


Now create EC2 Instance.

Creating Ec2 Instance

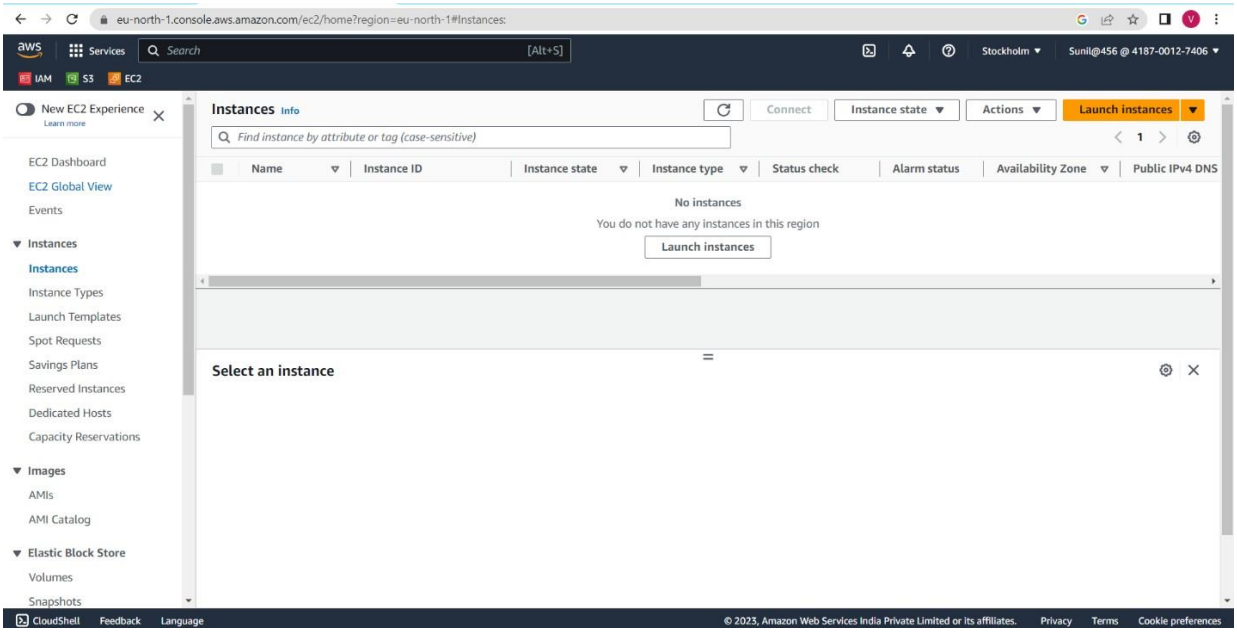
Step-1: -

Initially Login to The Aws Account And Go To The Ec2 Service.



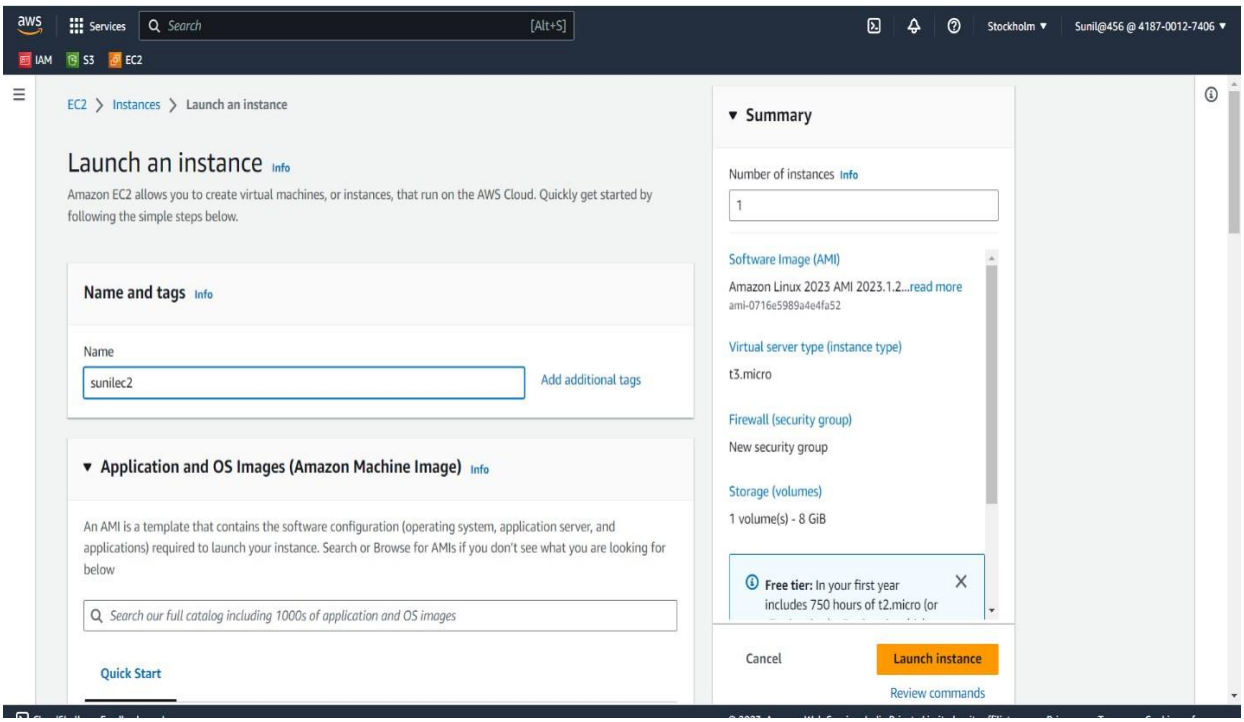
Step-2: -

Go to Instances and Click on Launch Instance.



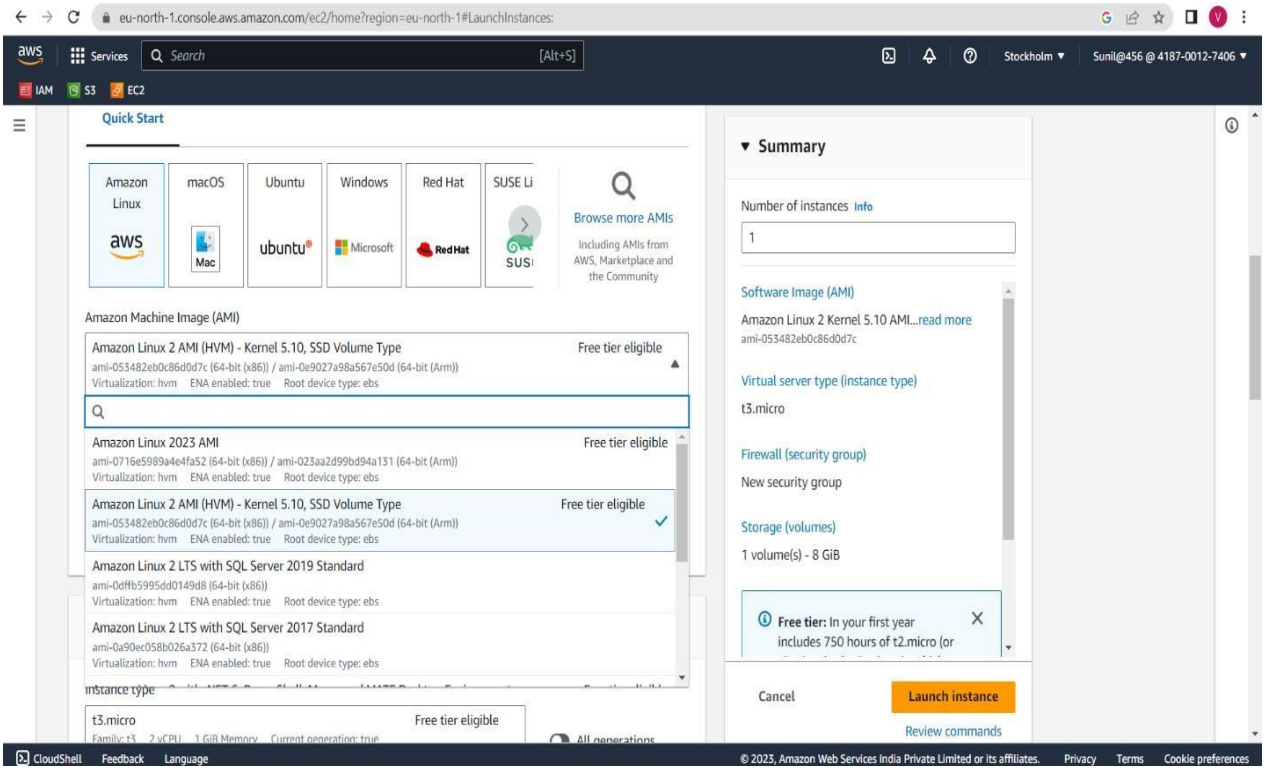
Step-3: -

Name the instance.



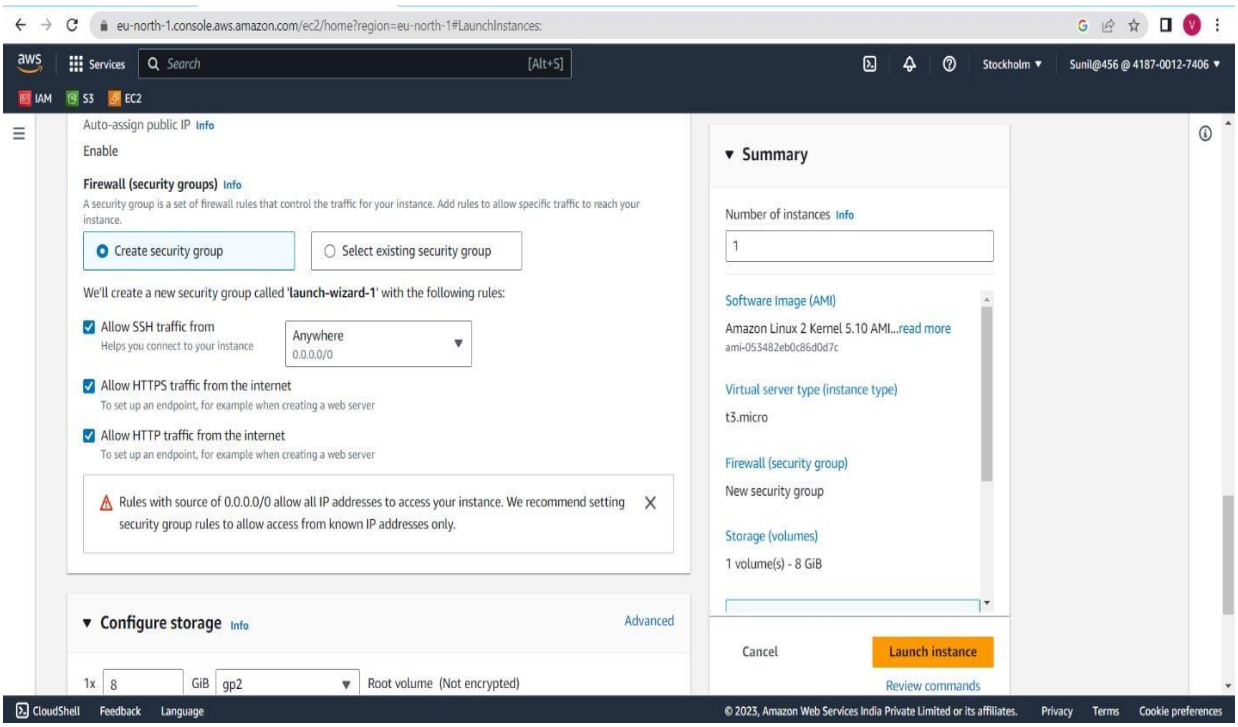
Step-4: -

Select Software Image (AMI): Amazon Linux 2 AMI(HVM) and also Virtual server type (instance type): t2.micro and select key pair (login) and select key.

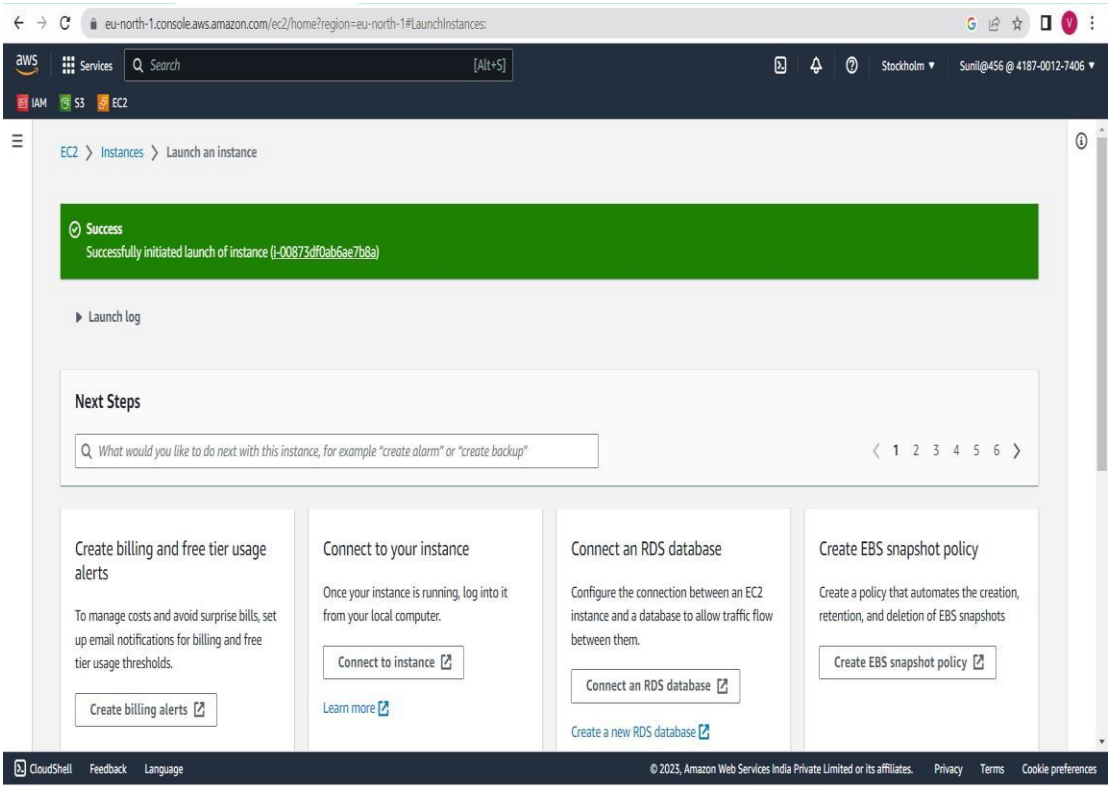


Step-5: -

Allow security permissions and click on launch instance.



Step-6: -Instance launched successfully.

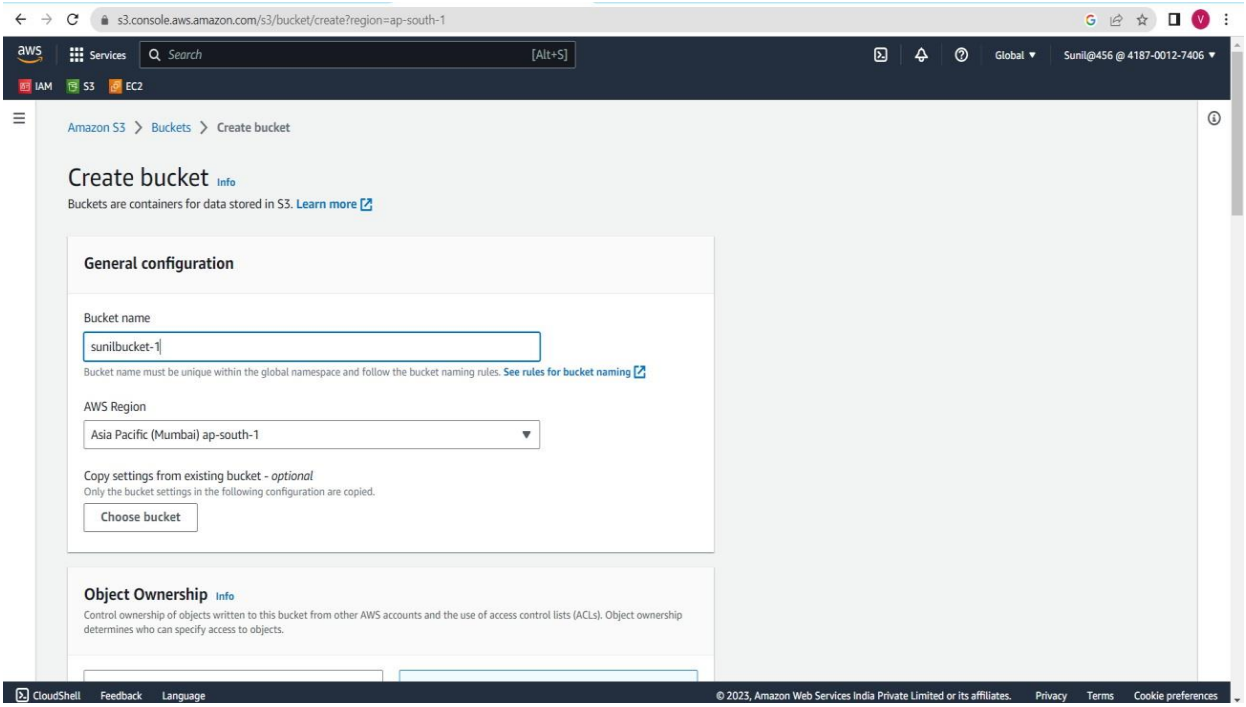


Static website hosting with S3

To deploy a static website on AWS S3, follow the steps below:

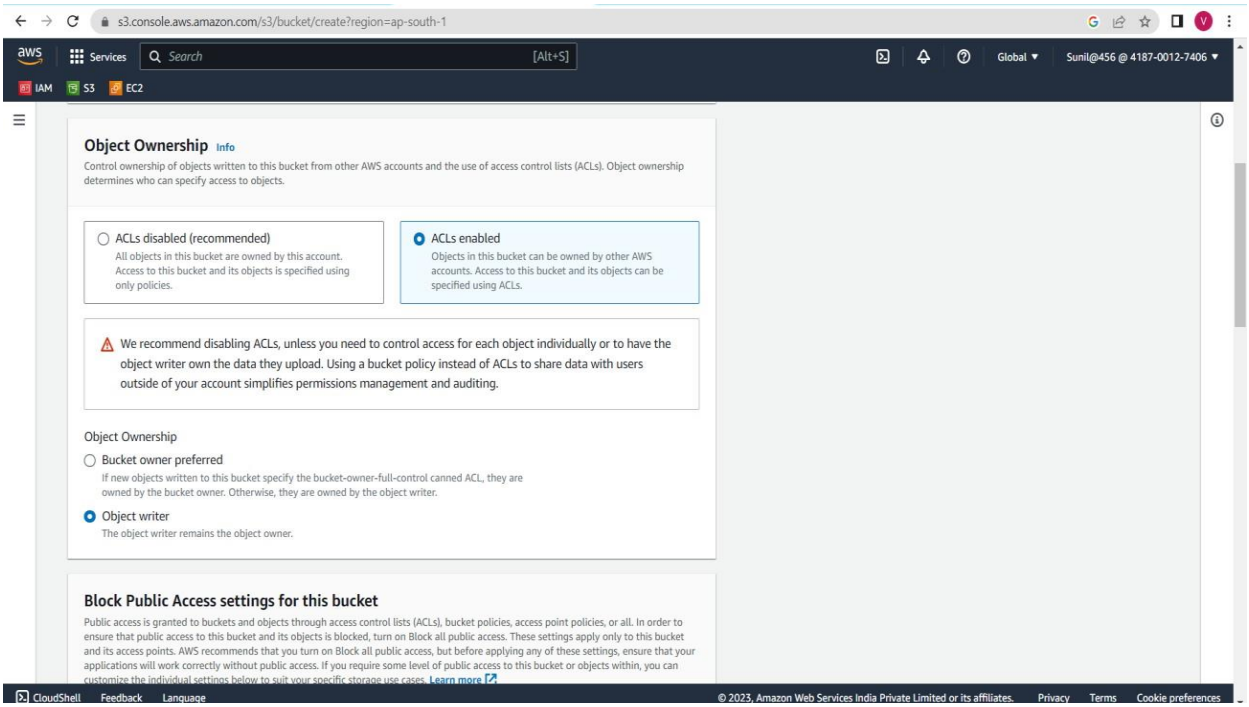
Step-1: -Creating an S3 bucket:

Login to AWS account, go to the S3 service. Click on “Create bucket”. Provide a unique name and choose the closest region.



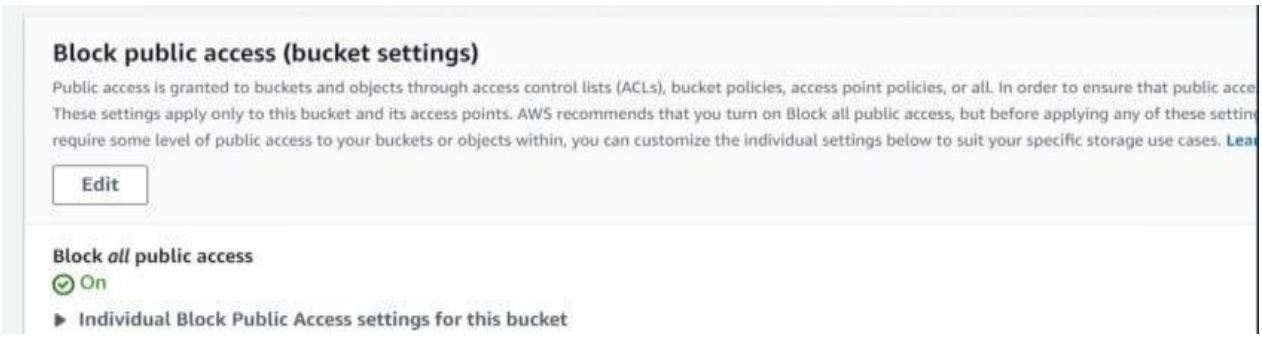
Step-2: -

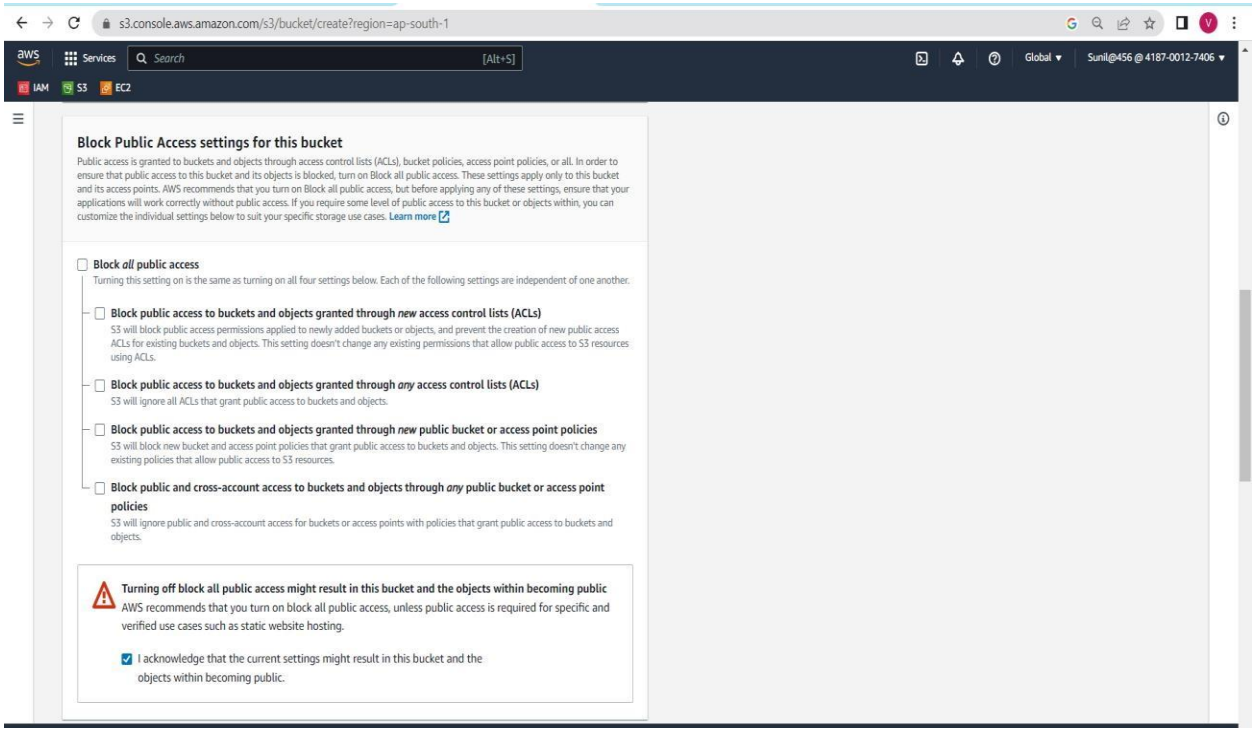
Select ACLs enabled in the object ownership.



Step-3: -Setting permissions:

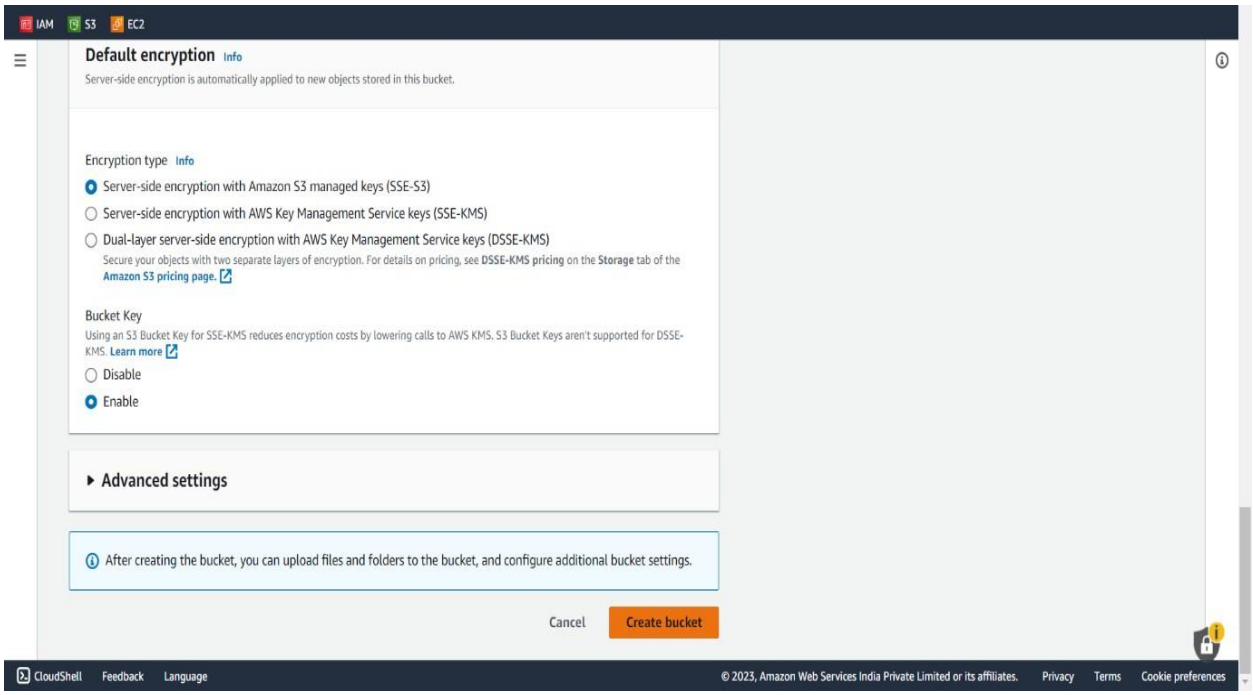
Click on permissions and edit the block public access, uncheck the checkbox and save changes.





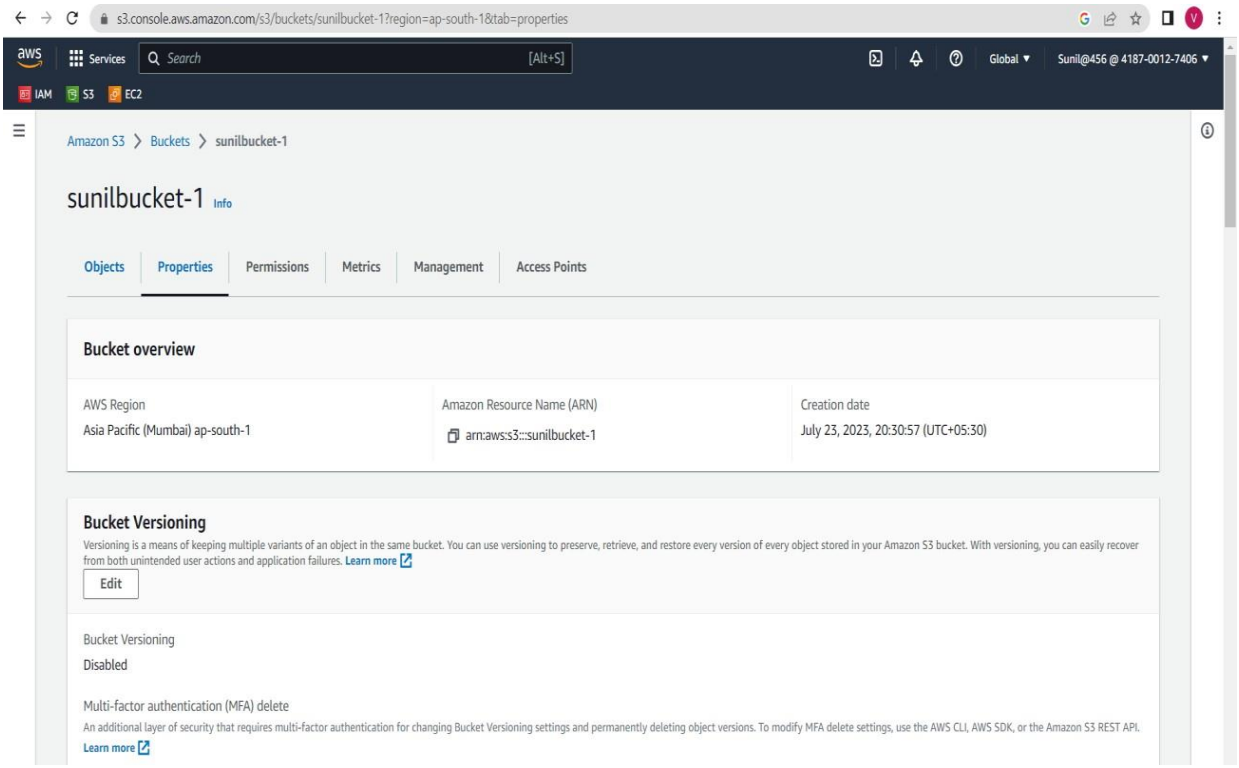
Step-4: -

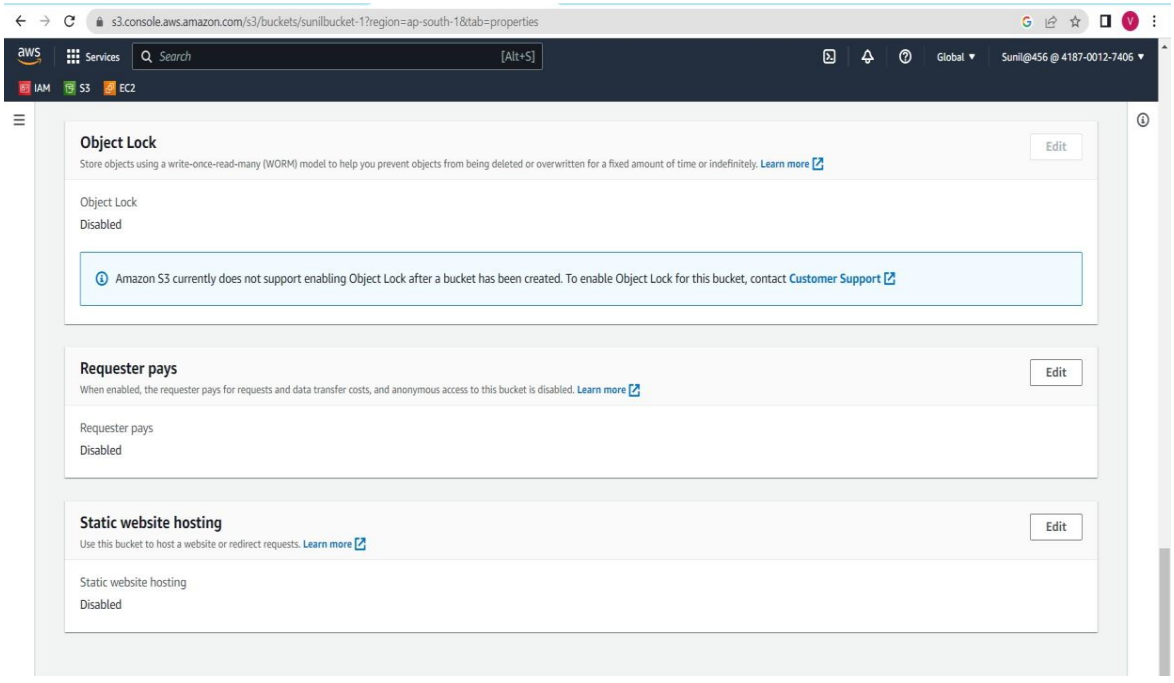
Click on create bucket.



Step-5: -

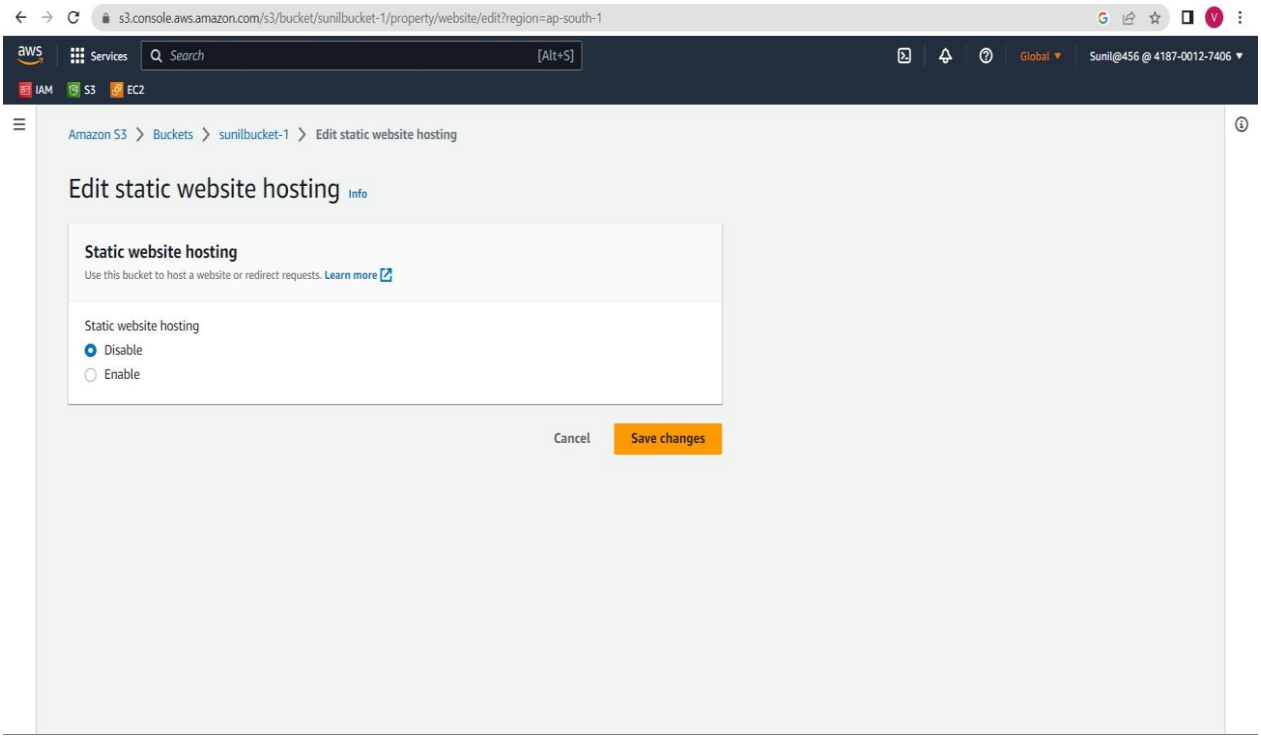
On properties, click on edit in the Static web hosting.





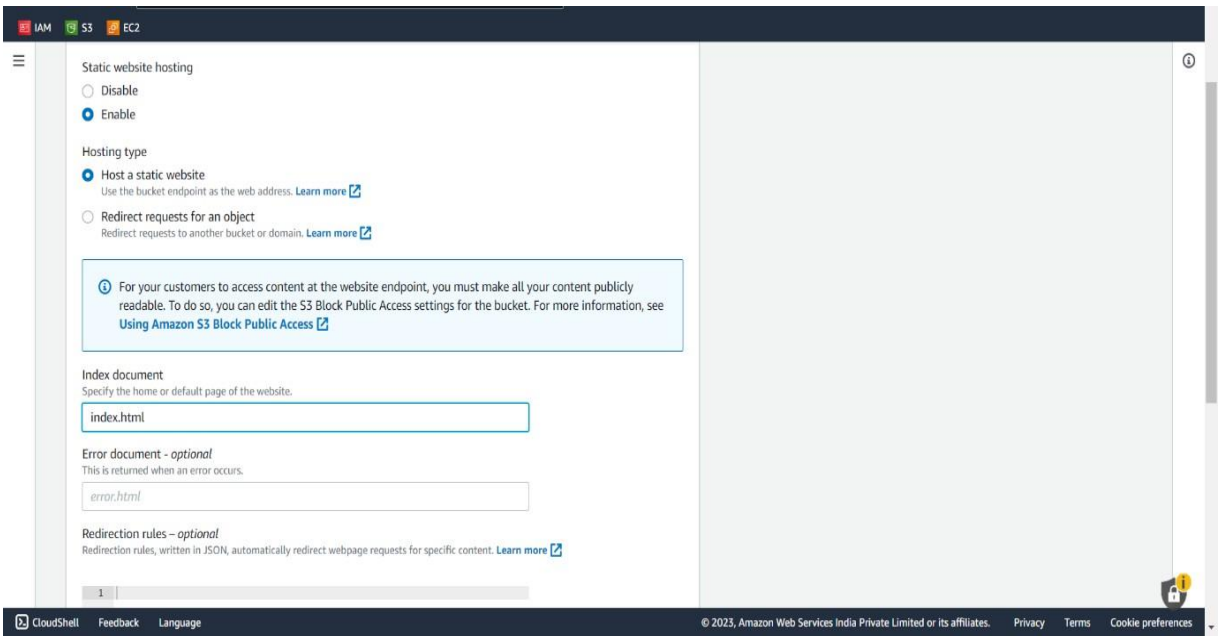
Step-6: -

Enable Static website hosting and click on save changes button.



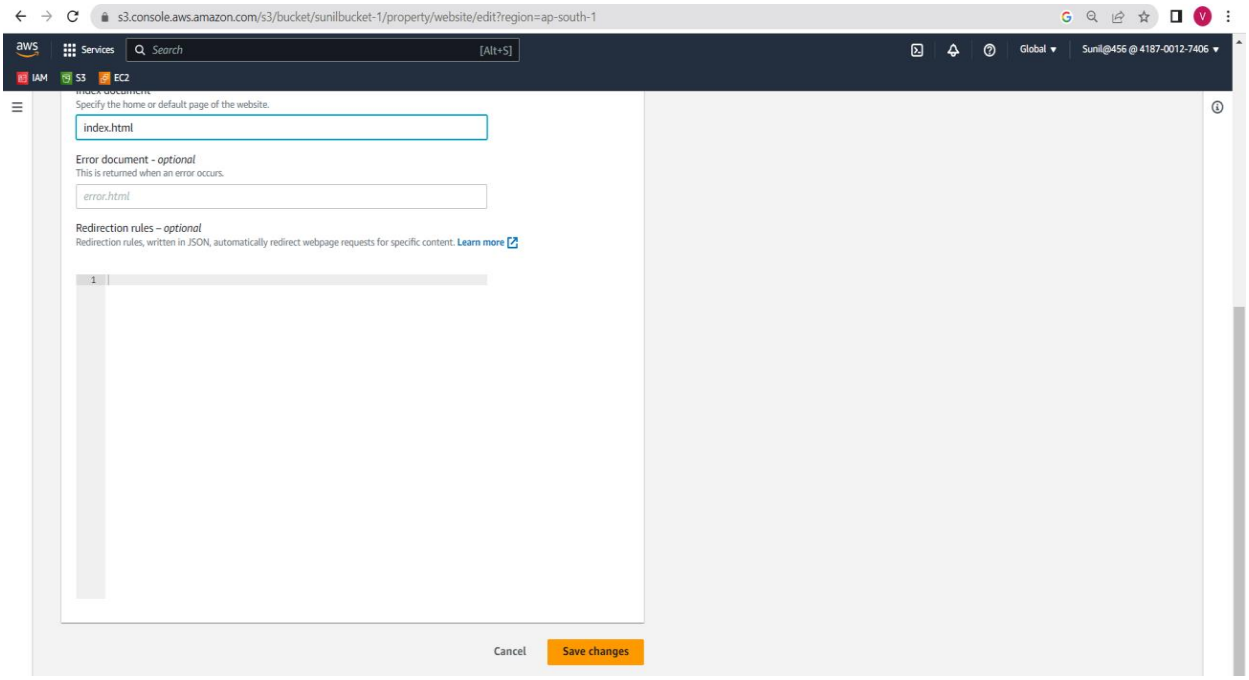
Step-7: -

Give index document as “index.html”.



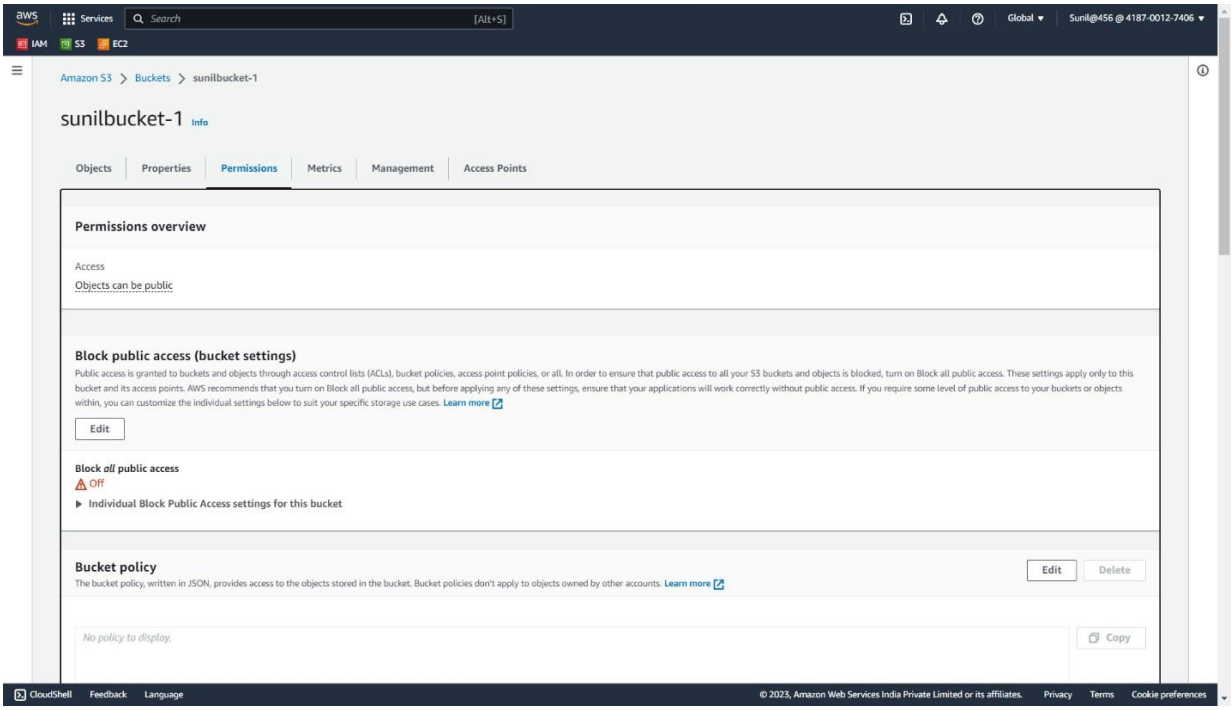
Step-8: -

Click on “Save changes”.

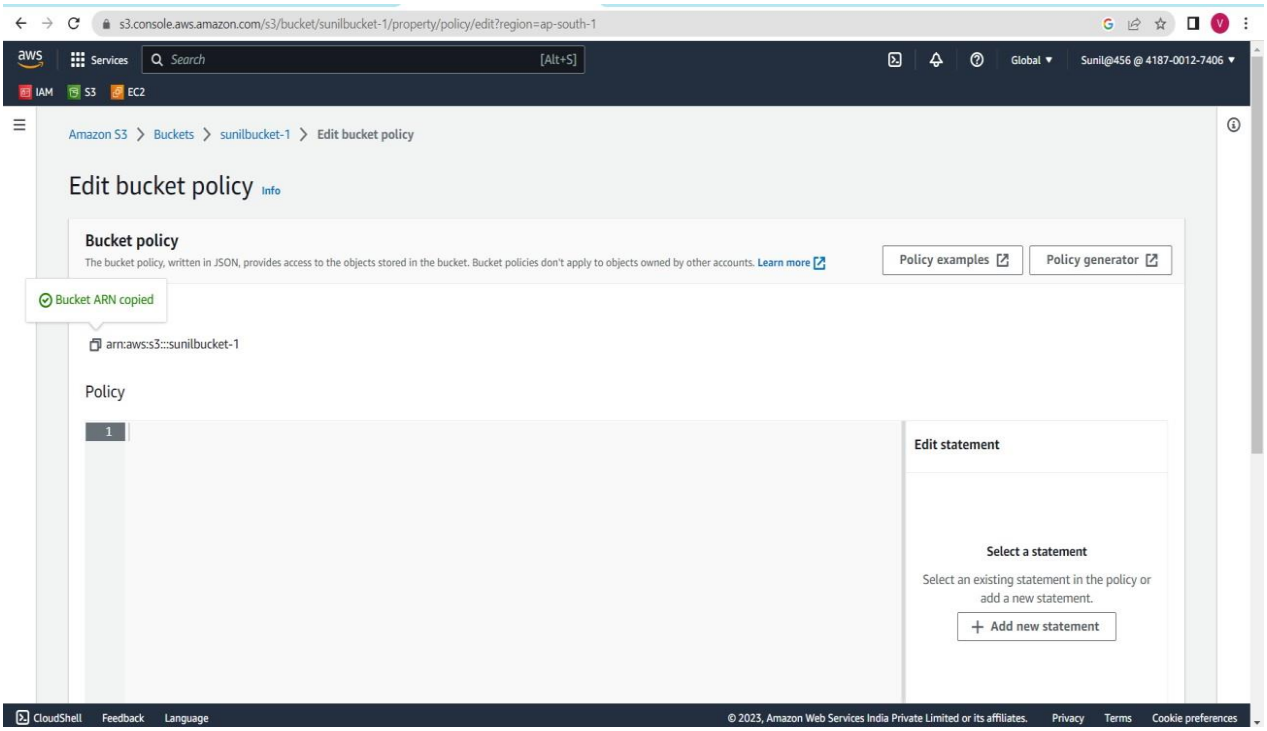


Step-9: -

On permissions, click on edit in the bucket policy.

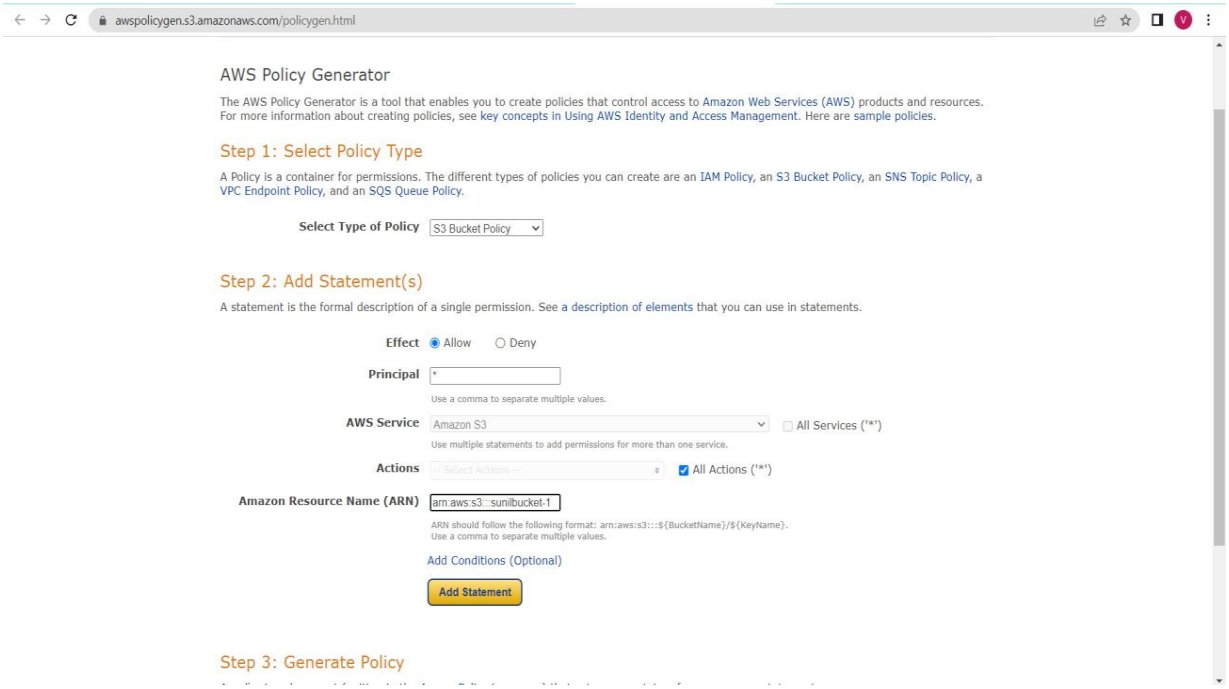


Step-10: -Copy the bucket ARN and click on policy generator.



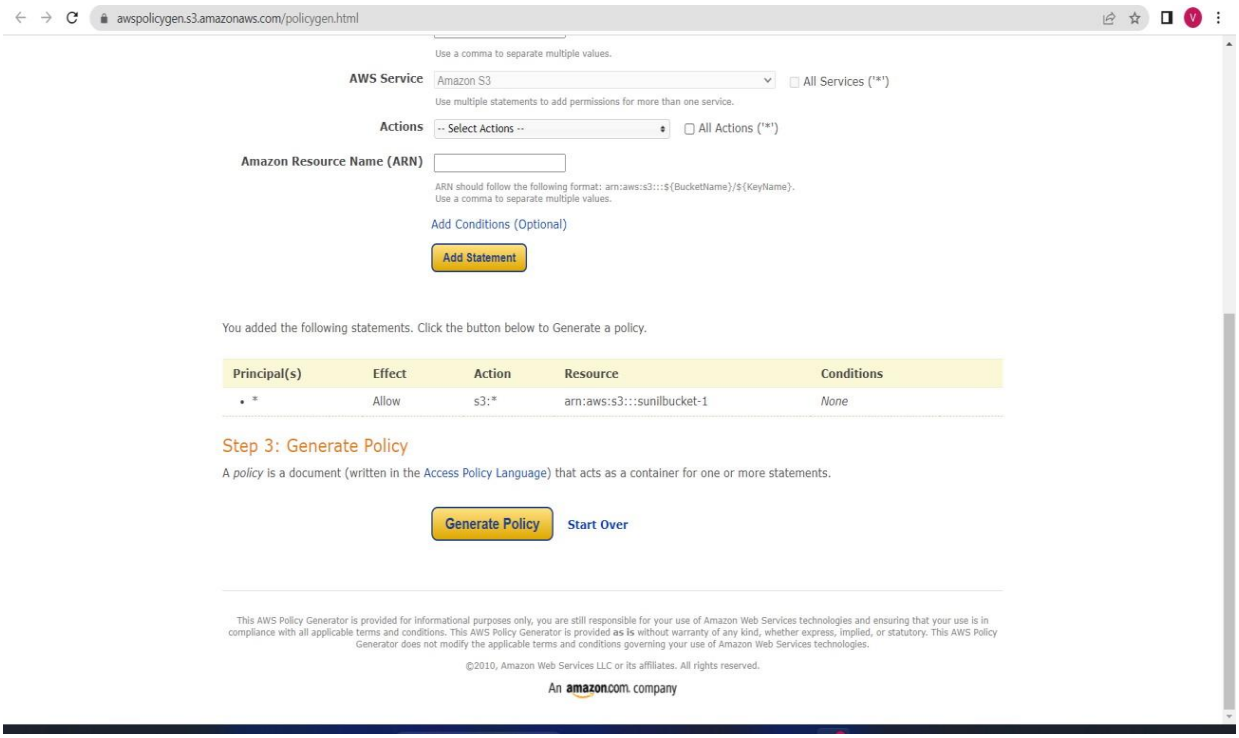
Step-12: -

Select s3 bucket policy In principle type * (everyone) Select All actions.Paste ARN and Click on Add statement.



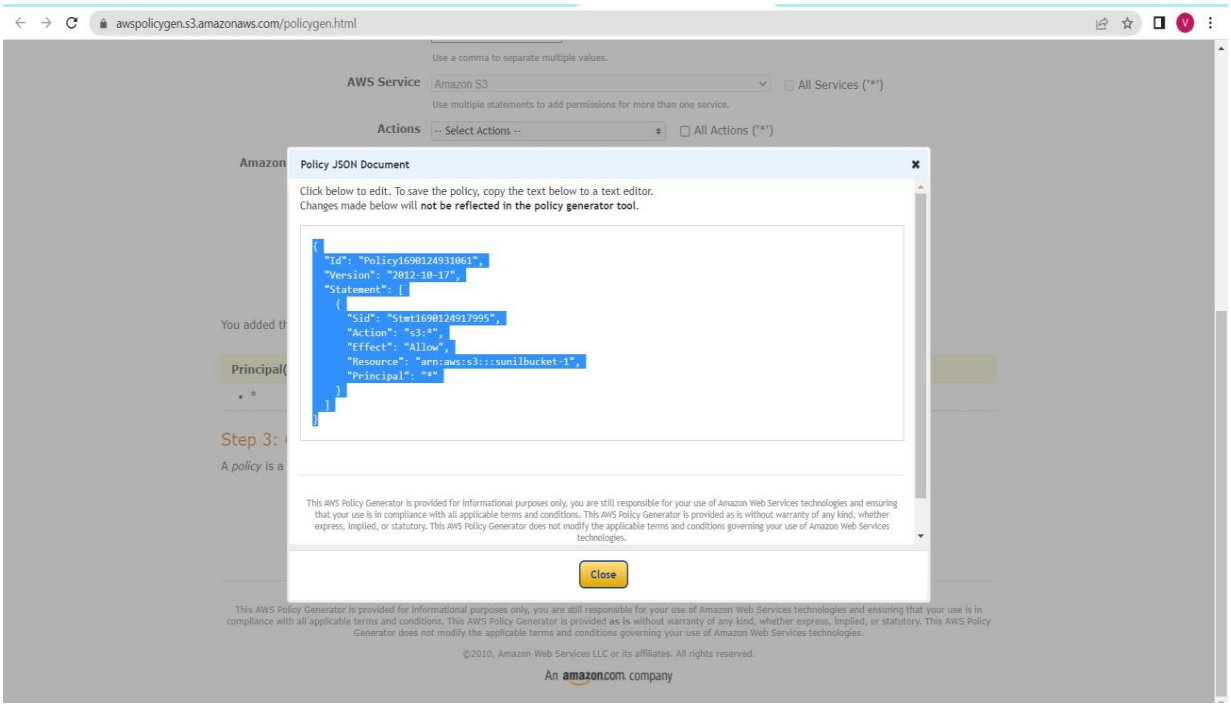
Step-13: -

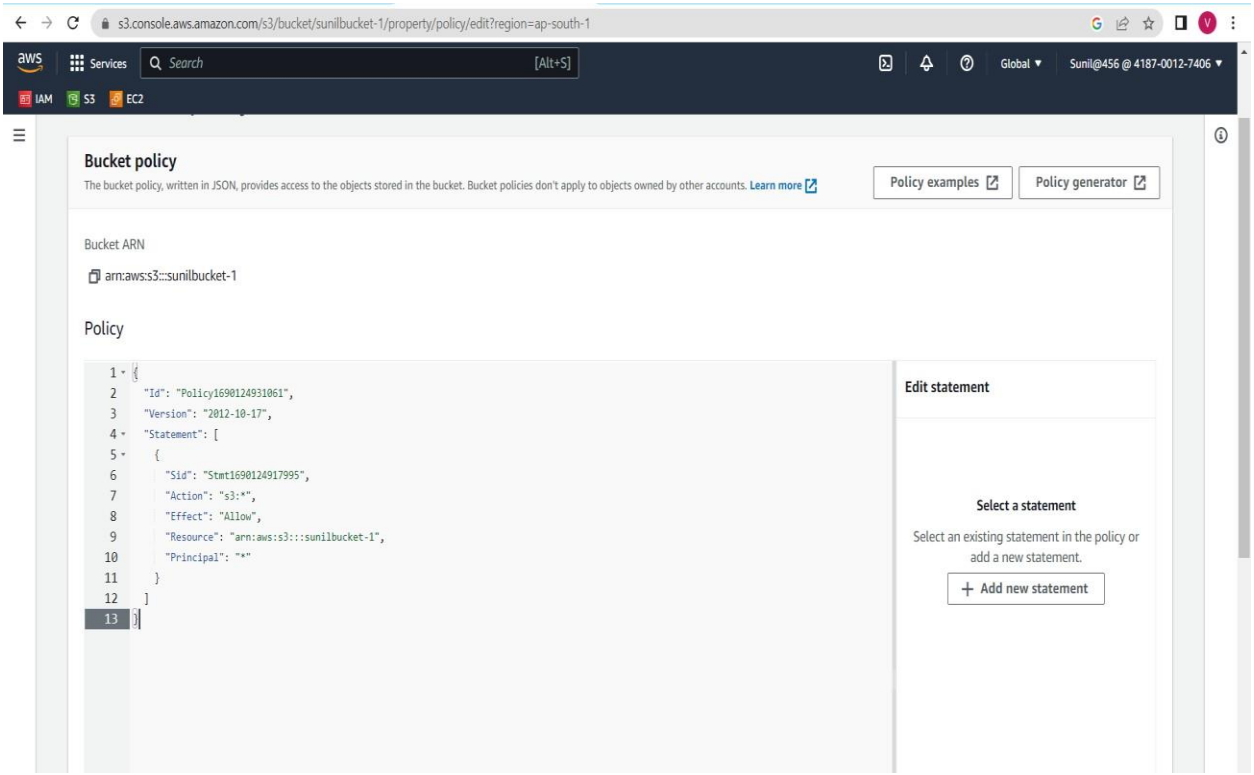
Click on Generate Policy.



Step-14: -

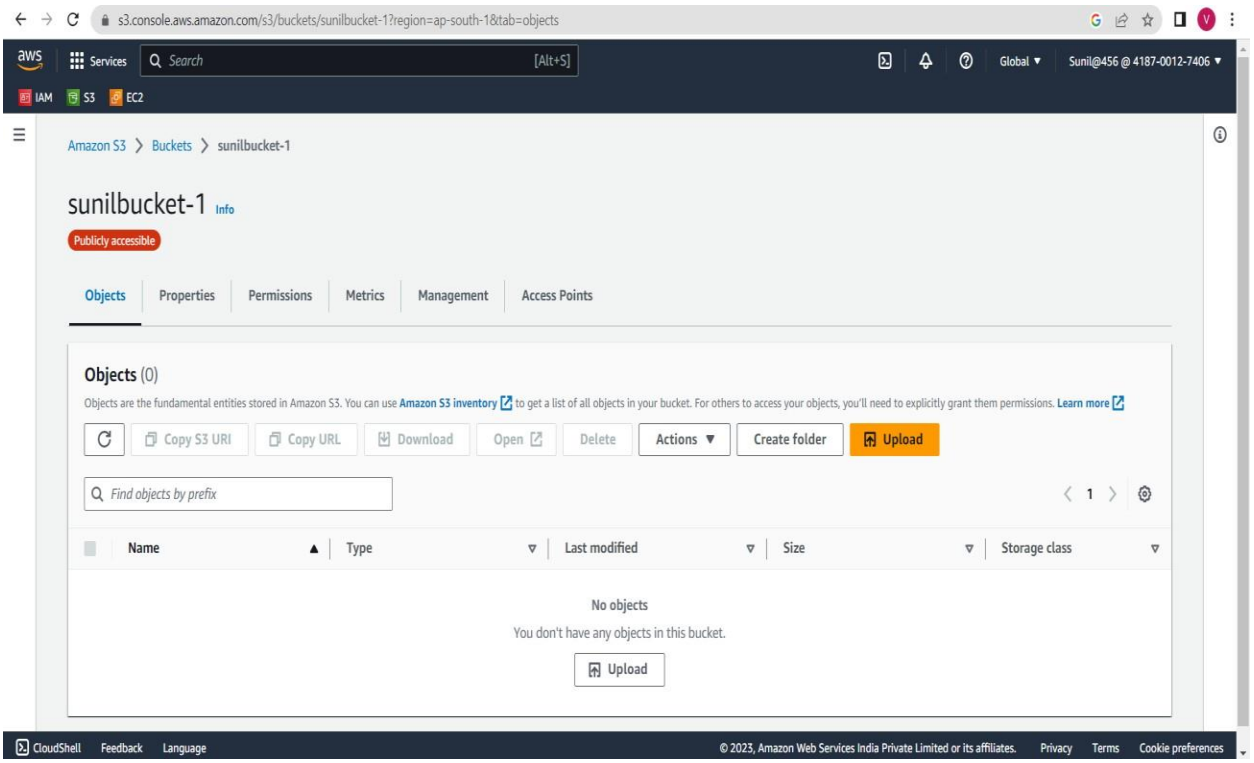
Copy the policy and paste it in the bucket policy and click on save changes.





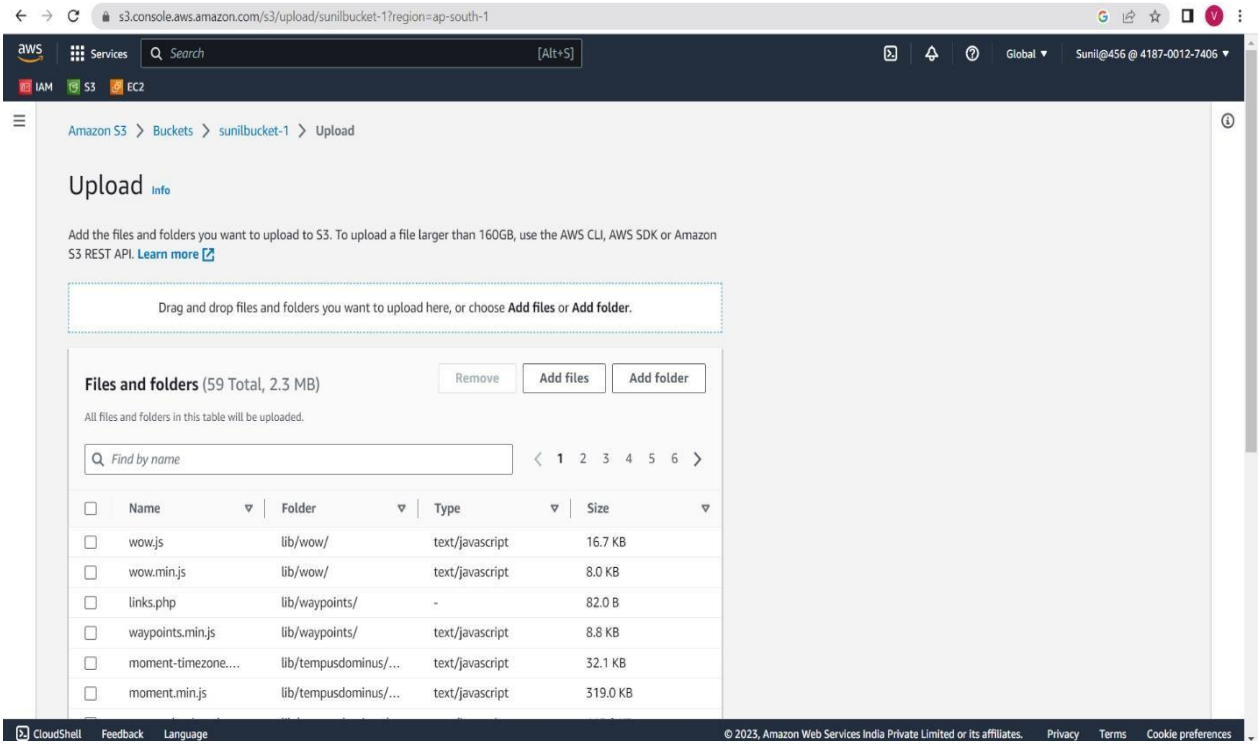
Step-15: -

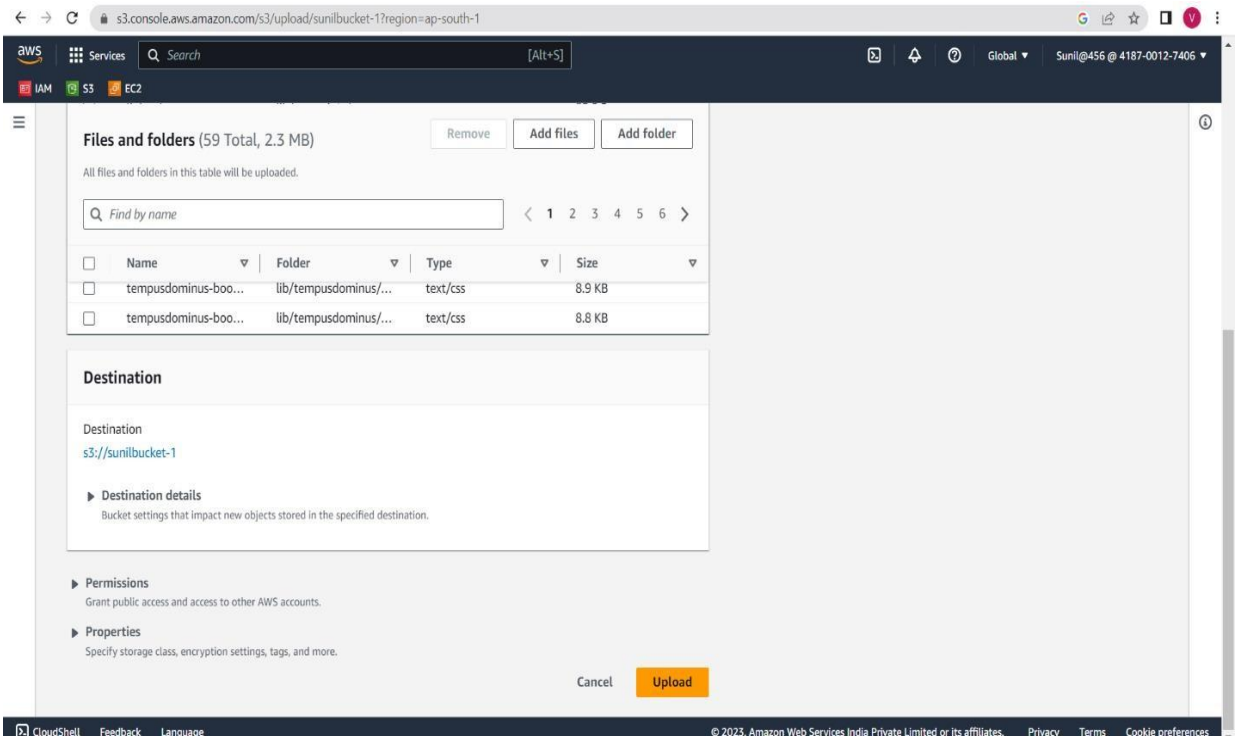
On objects tab, click on upload.

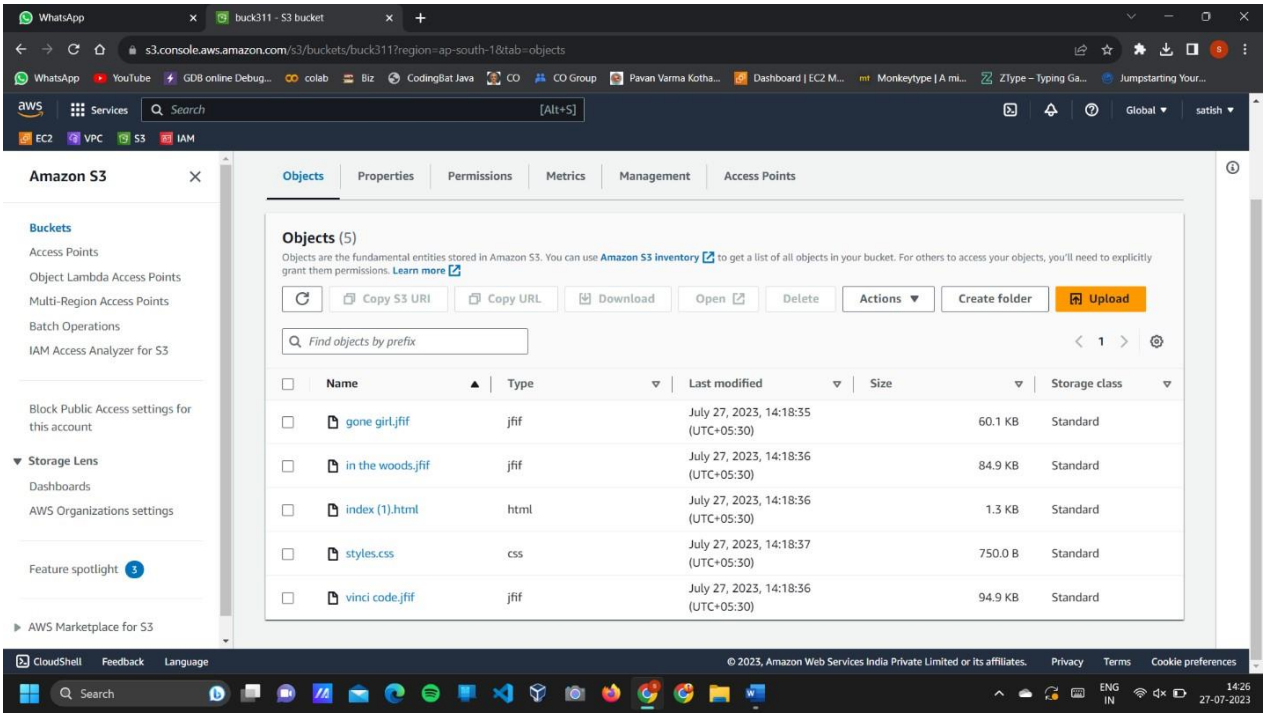


Step-16: -

Add files and folders that are related to tourist website then click on upload.



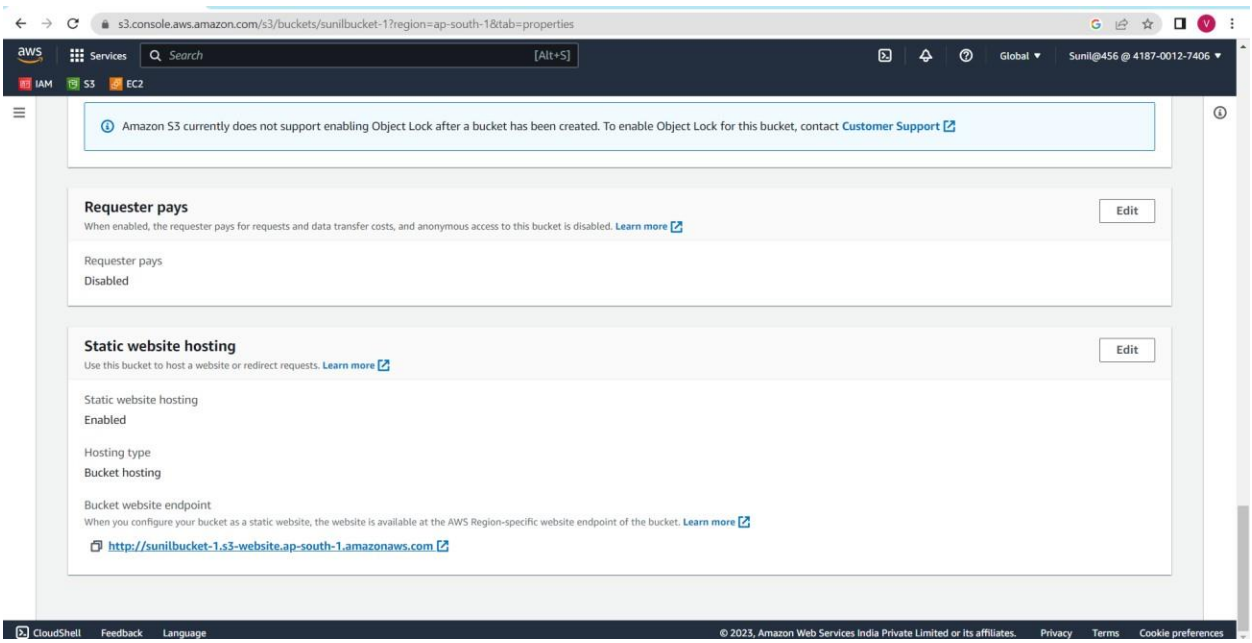




Successfully uploaded the files and folders in object section.

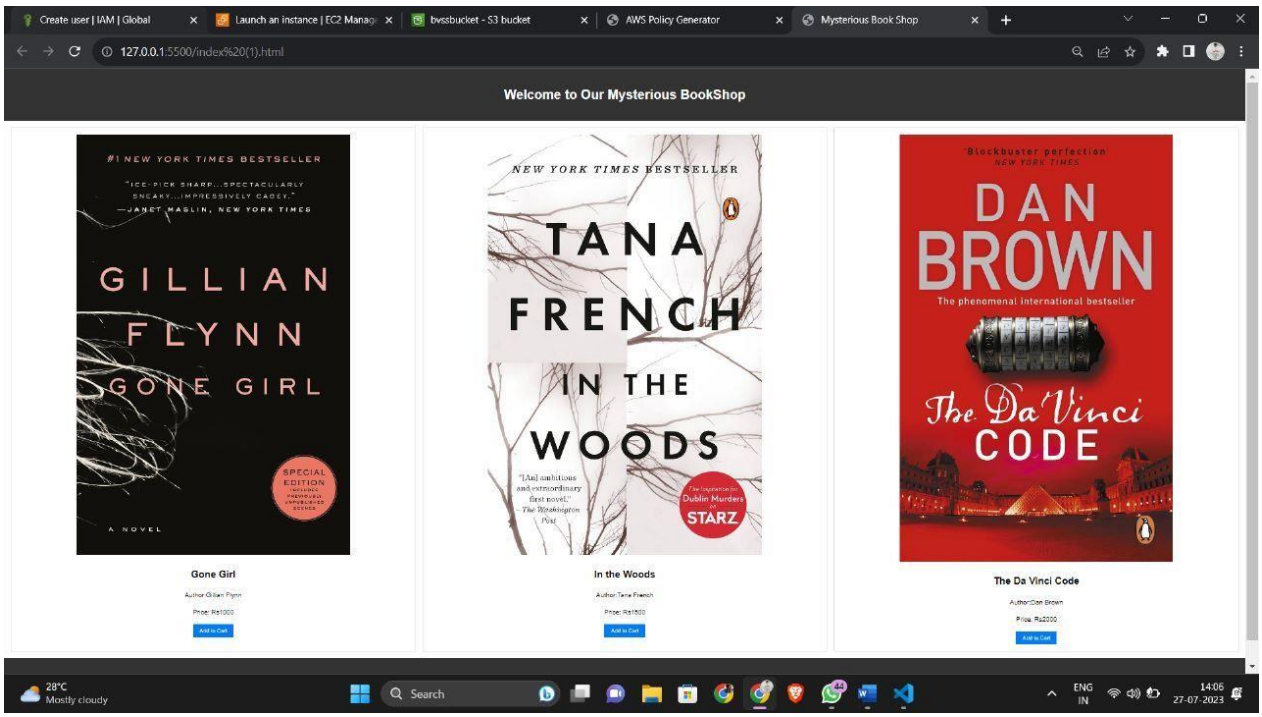
Step-17: -

On properties section, go to Static website hosting and copy the endpoint URL of the tourist website using s3 bucket.



Step-18: -

Verify the website by browsing the endpoint URL of the bucket.



DECLARATION

I hereby declare that "AWS architecture to send alert message in case of greater CPU utilization" is the result of the project work carried out by me under the guidance of Mr. Aashu Dev.

DATE: **SIGNATURE**: