AWS Introduction

Most Important AWS Topics/Services for Data Science

Here's a curated list of **essential AWS services** specifically for Data Science, with beginner-friendly explanations:

AWS Service	Why It Matters (Simple Explanation)
S3 (Simple Storage Service)	Like a big online USB drive. You use it to store datasets, CSVs, models, logs, etc.
EC2 (Elastic Compute Cloud)	Like renting a remote computer in the cloud where you can run Jupyter, Python code, train models, etc.
SageMaker	AWS's main tool for Data Science. Lets you build, train, deploy ML models without worrying about hardware or backend setup.
IAM (Identity & Access Management)	Like a lock & key system. Controls who can access your AWS stuff (important for security).
Lambda	Run Python code automatically when something happens (like when new data is uploaded). No server needed.
CloudWatch	Like a security camera + diary. It watches your AWS resources, tracks usage, errors, etc.
RDS (Relational Database Service)	Managed SQL databases in the cloud. Think MySQL/PostgreSQL without needing to install anything.
Glue	Helps move and transform data between systems. Useful for cleaning and preparing large datasets.
Athena	Query large CSV/Parquet/JSON files stored in S3 using SQL. No need to move data anywhere.
Step Functions	Like a flowchart engine. Helps automate workflows like "first clean data → then train model → then email results".

AWS Service	Why It Matters (Simple Explanation)
CloudFormation	Like a recipe. You write code to describe your AWS setup, and AWS builds it for you automatically.
ECR + ECS *(Optional later)*	For containerized apps. Helpful if you use Docker for model deployment. We'll revisit this later.

1. **Amazon S3** – Your cloud "hard drive" for storing datasets, models, and results.

   Why? All data science starts with data, and S3 is where you'll store it.

2. **AWS Lambda** – Run code without managing servers (e.g., preprocess data automatically).

   Why? Automate small tasks like cleaning data or triggering ML pipelines.

3. **Amazon SageMaker** – Fully managed service to build, train, and deploy ML models.

   Why? The heart of AWS ML—no need to set up servers or frameworks manually.

4. **AWS Glue** – Prepares and transforms data for analysis (ETL: Extract, Transform, Load).

   Why? Real-world data is messy; Glue cleans it for you.

5. **Amazon EC2** – Virtual computers in the cloud (for heavy computations).

   Why? When you need more power than your laptop for training big models.

6. **AWS IAM** – Controls who can access your AWS resources (security first!).

   Why? Avoid accidental data leaks or unauthorized costs.

7. **Amazon Athena** – Query data in S3 using SQL (no database setup needed).

   Why? Quickly analyze large datasets without moving them.

8. **Amazon QuickSight** – Create dashboards and visualize your data.

   Why? Share insights with non-technical teams.

9. **AWS Step Functions** – Coordinate multiple AWS services into workflows.

   Why? Chain data preprocessing → training → deployment automatically.

10. **Bedrock (for GenAI)** – Access to foundation models like Claude, Llama 2.

    Why? Add generative AI (text, images) to your apps without training from scratch.

Simple Questions

Which topic excites you the most? Or ask:

- "How do I store a CSV file in AWS?" → **S3**
- "How do I train a model without installing anything?" → **SageMaker**
- "How do I automate a data pipeline?" → **Lambda + Glue**

Common Bill Shock Scenarios (And Fixes)

Scenario	Why It Happens	How to Prevent
EC2 left running 24/7	Forgot to turn off a test server	Use **AWS Instance Scheduler** or set auto-termination timers
S3 bucket with public access	Accidentally exposed files	Enable **S3 Block Public Access**
Lambda functions spamming	Infinite loop in code	Set **Concurrency Limits**

Key Tips for Beginners

1. **Always enable MFA (Multi-Factor Auth):** Prevents hackers from mining Bitcoin on your account.
2. **Delete unused resources:** Check **AWS Cost Explorer** monthly for "ghost" services.
3. **Use the Free Tier wisely:** Stick to `t2.micro` EC2 instances and small S3 buckets.
4. **Mock bills with the Pricing Calculator:** AWS Pricing Calculator.

Reality Check

- AWS **won't charge you without warning** (you'll get alerts first).
- Small experiments (e.g., running a server for 1 hour) cost pennies.
- Major leaks usually happen due to **misconfigured auto-scaling** or **public data transfers**.

📦 What Is the AWS Free Tier?

AWS offers a **12-month Free Tier** for most services.

Examples:

- **S3**: 5 GB of storage free
- **EC2**: 750 hours/month of micro instance (t2.micro or t3.micro)
- **SageMaker**: 250 hours of ml.t2.medium
- **Bedrock**: limited free LLM tokens for some models

💡 But: If you go beyond the free limits (like 6 GB S3 instead of 5 GB), then you're charged for the **extra 1 GB only.**

💡 Pro Tips for New Users

Mistake	Prevention
Leaving EC2 running 24/7	Stop or terminate it after use
Choosing large instance types	Stick to `t2.micro`, `t3.micro` for Free Tier
Not deleting SageMaker notebooks	Stop the notebook to avoid surprise charges
Using services not in Free Tier	Double-check AWS Free Tier website

What is an EC2 Instance?

Imagine an **EC2 instance as a "virtual computer"** that you rent from AWS.

- It's like a **remote laptop** with CPU, RAM, storage, and OS (Linux/Windows).
- You control it entirely: install software, run code, or host websites.
- You pay only for the time it's running (**per second/hour**).
- You can **log in**, **run code**, **train ML models**, **host apps**, etc.

💡 Real-Life Example:

🧪 Case: You want to train a model on 1 million rows of data but your laptop is too slow.

- You create an EC2 instance with more memory and CPU (or GPU).
- Upload your data and training script.
- Run the training on EC2.
- Download the model weights.
- **Stop the instance**.

Cost? Just the hours you used it — could be ₹10 or ₹100 depending on size/time.

🚫 Common Beginner Mistakes

Mistake	Fix
Leaving EC2 running	Stop it after use from AWS console
Picking big instance types	Start with Free Tier (`t2.micro`)
Not setting SSH key	You won't be able to login — always download the `.pem` key safely
Using EC2 to store data permanently	EC2 storage gets erased on termination — store important data in S3

🔐 Smart Strategy for Beginners

Use Case	What to Do
Learning EC2, Jupyter, Python setup	✅ Use `t2.micro` or `t3.micro`, stop it when done

Use Case	What to Do
Hosting website/chatbot (low traffic)	✅ Use `t3.micro` , set a **₹500 budget alert**, monitor usage
Hosting only during the day	✅ Use **Lambda or Fargate** (event-based, cheaper)
Need GPU for model training	✅ Use EC2 **just for the few hours you train**, then stop

💰 Cost Insight

Setup	Monthly Cost (est.)
Single `t3.medium` instance	₹1,300–₹1,600
Two `t3.micro` instances	₹1,200–₹1,500 combined
One `g4dn.xlarge` GPU instance (training only)	₹12,000–₹20,000+ 💸

Types of Cloud Infrastructure Services

- **Infrastructure as a Service (IaaS)**
- **Platform as a Service (PaaS)**
- **Software as a Service (SaaS)**

1. IaaS (Infrastructure as a Service)

- **Analogy:** You get the land and foundation.

Your responsibility:

- Operating system (Linux, Windows)
- All your software (Python, libraries, specific versions)
- Applications (Jupyter notebooks, custom data pipelines)
- Network configuration *within* your virtual network
- Security patching *of your OS and software*

AWS's responsibility:

- Physical servers, networking hardware
- Virtualization layer (making one big server act like many small ones)
- Power, cooling, physical security of data centers

Key Characteristics:

- **Maximum control:** You decide everything.
- **Flexibility:** Tailor environments exactly to your needs.
- **Cost-effective for variable loads:** Pay only for what you use.
- **Requires expertise:** You need to know how to manage servers.

AWS Examples for Data Science:

- **Amazon EC2 (Elastic Compute Cloud):** Your virtual machine. Spin up a powerful EC2 instance with many CPUs or GPUs for training complex ML models. Install specific deep learning frameworks (TensorFlow, PyTorch) directly onto it.
- **Amazon S3 (Simple Storage Service):** Object storage for massive datasets. Store raw data lakes, processed data, model artifacts, and project files. Highly durable and scalable, no server to manage for S3 itself.

- **Amazon EBS (Elastic Block Store):** Persistent storage volumes attached to EC2 instances. Like a hard drive for your virtual server, good for databases or when your EC2 needs fast, dedicated storage.
- **Amazon VPC (Virtual Private Cloud):** Your isolated network in the cloud. Define your own IP addresses, subnets, and security rules. Essential for securing your data science environments.

- **Why:** Max control, highly flexible.

2. PaaS (Platform as a Service)

- **Analogy:** You get a partially built house.

Your responsibility:

- Your application code (e.g., Python script for a model API)
- Your data
- Configuration specific to your application

AWS's responsibility:

- All IaaS components (servers, network, OS)
- Runtime environment (e.g., Python interpreter, Node.js)

- Middleware (web servers like Apache/Nginx, database software)
- Scalability and load balancing for the platform
- Security patching for the platform itself

Key Characteristics:

- **Faster development/deployment:** Focus on coding, not infrastructure.
- **Managed environment:** AWS handles most operational tasks.
- **Built-in scalability:** Often scales automatically or with minimal config.
- **Less control/flexibility:** Limited access to the underlying infrastructure.

AWS Examples for Data Science:

- **Amazon RDS (Relational Database Service):** Managed relational databases (PostgreSQL, MySQL, SQL Server, etc.). You don't install or patch the database server; AWS handles it. You just connect and query. Great for structured data.
- **AWS Lambda (Serverless Compute):** Run your code without managing servers. Upload a Python function to process data, trigger it when a new file lands in S3 or on a schedule. Pay only when your code runs. Perfect for event-driven tasks.
- **Amazon SageMaker:** A comprehensive ML platform. Provides managed environments for:
    - **Jupyter Notebooks (SageMaker Studio):** Pre-configured, powerful notebooks for exploration and development.
    - **Training Jobs:** Run large-scale model training without setting up servers.
    - **Model Endpoints:** Deploy models as APIs for real-time predictions; SageMaker manages the hosting.
- **AWS Elastic Beanstalk:** Deploy web applications (like a Flask app serving an ML model) easily. AWS handles the underlying EC2 instances, load balancing, and auto-scaling.

- **Why:** Focus on coding, faster deployment.

3. SaaS (Software as a Service)

- **Analogy:** You rent a furnished apartment.

Your responsibility:

- Using the application.
- Your data *within* the application.

AWS's responsibility:

- Everything: the application itself, its data, runtime, middleware, OS, servers, network, storage.
- All maintenance, updates, security.

Key Characteristics:

- **Easiest to use:** Just log in and go.
- **No maintenance burden:** Provider handles everything.
- **Subscription-based:** Usually paid per user or per usage.
- **Least control/customization:** You use it as it's provided.

AWS Examples (less direct "infrastructure" but good to know):

- **Amazon QuickSight:** Business intelligence service. Connects to your data sources (like S3, RDS) and lets you build interactive dashboards and reports. You don't manage any servers for QuickSight; you just use the web interface.
- **AWS Management Console:** The web interface you use to manage all your AWS services. It's a SaaS application itself!

- **Why:** Easiest to use, no maintenance.

🔐 Bonus: Limit Costs Programmatically

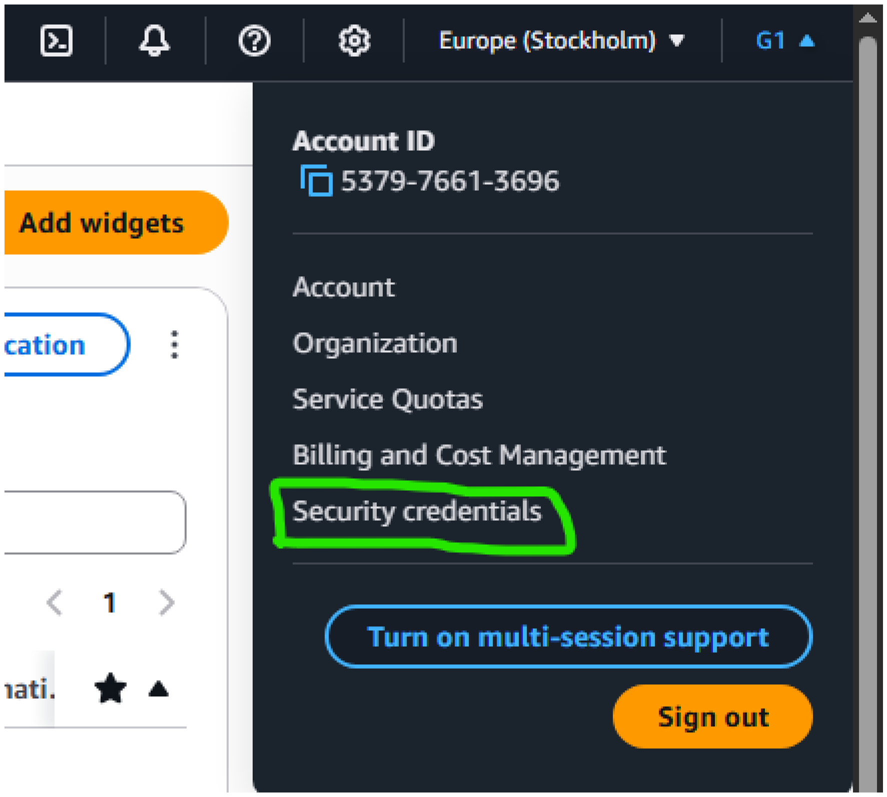
You can even set up a **Lambda function** + **CloudWatch Alarm** to:

- Stop EC2 or SageMaker automatically when you hit budget limits
- Useful if you want to enforce hard spending stops

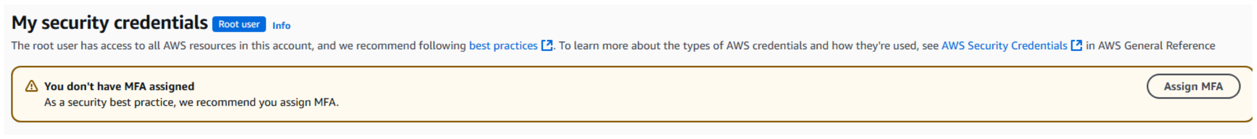
MFA (Multi-Factor Authentication) on your AWS Root Account

1. Login console as root user

2. Click your name in upper right hand corner → Security Credentials



3. Assign MFA


My security credentials Root user Info

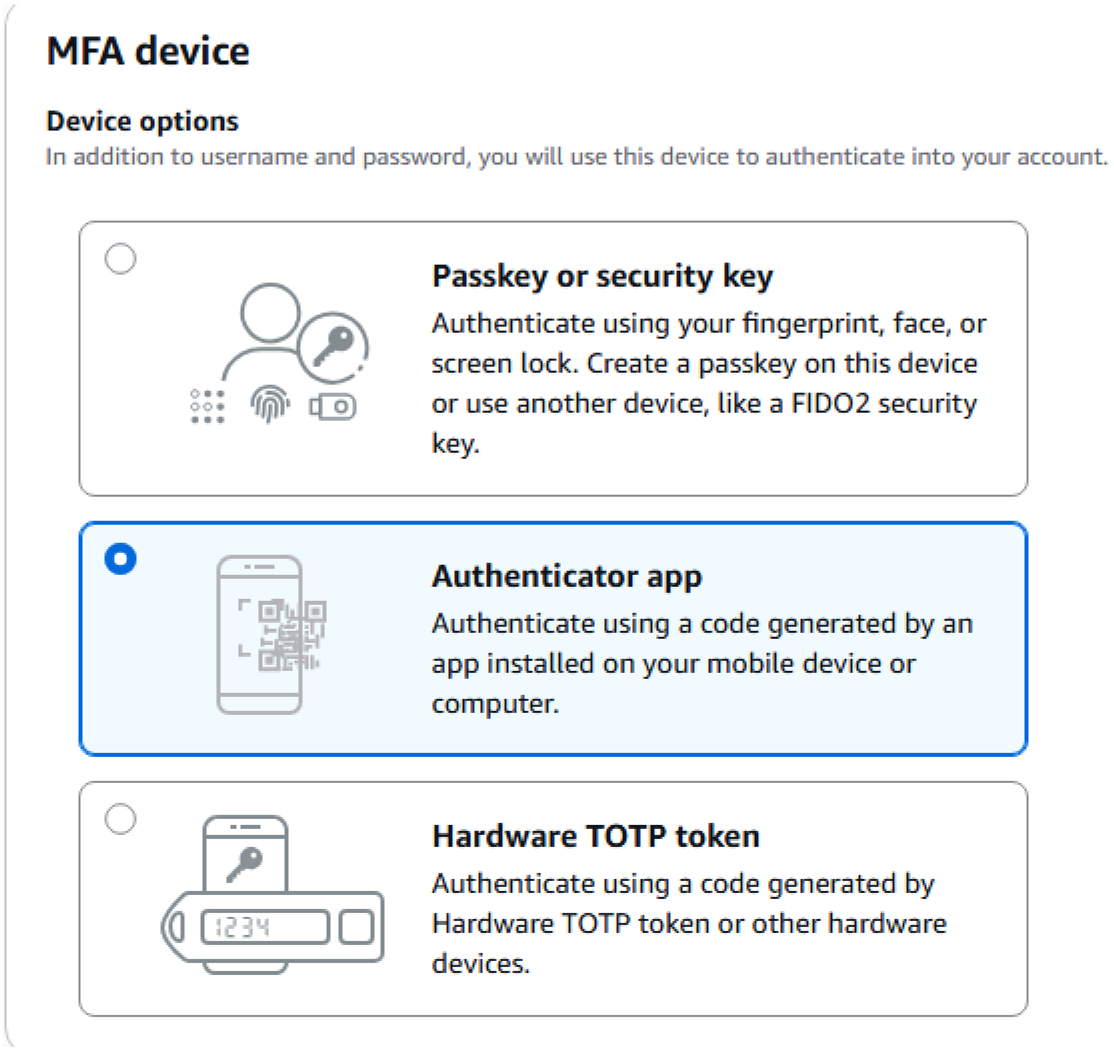
The root user has access to all AWS resources in this account, and we recommend following best practices. To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference

You don't have MFA assigned

As a security best practice, we recommend you assign MFA.

Assign MFA

4. Choose Authenticator App



MFA device

Device options

In addition to username and password, you will use this device to authenticate into your account.

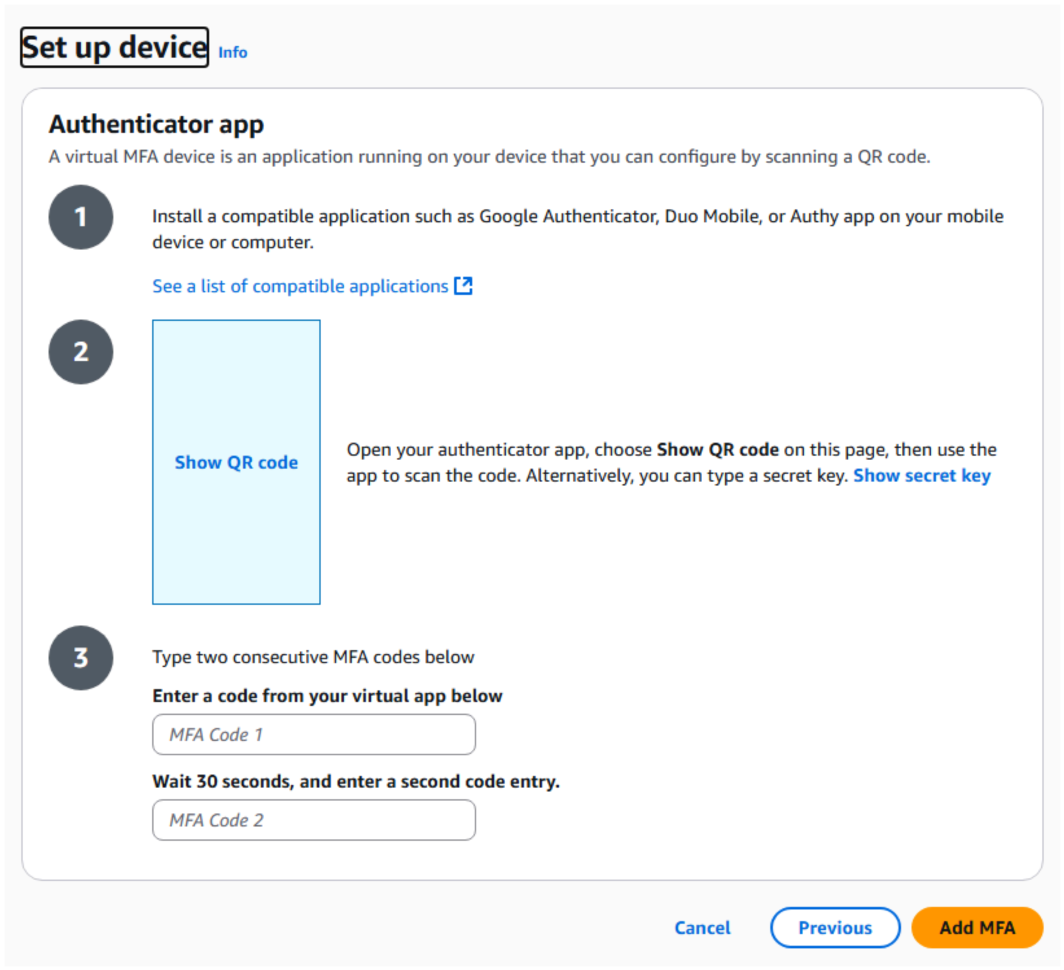
( ) **Passkey or security key**
Authenticate using your fingerprint, face, or screen lock. Create a passkey on this device or use another device, like a FIDO2 security key.

[✓ **Authenticator app**]
Authenticate using a code generated by an app installed on your mobile device or computer.

( ) **Hardware TOTP token**
Authenticate using a code generated by Hardware TOTP token or other hardware devices.

5. Set up device

- Show QR code
- Scan on Google Auth App



Set up device Info

Authenticator app

A virtual MFA device is an application running on your device that you can configure by scanning a QR code.

1. Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer.

   See a list of compatible applications

2. Open your authenticator app, choose **Show QR code** on this page, then use the app to scan the code. Alternatively, you can type a secret key. Show secret key

3. Type two consecutive MFA codes below

   Enter a code from your virtual app below

   MFA Code 1

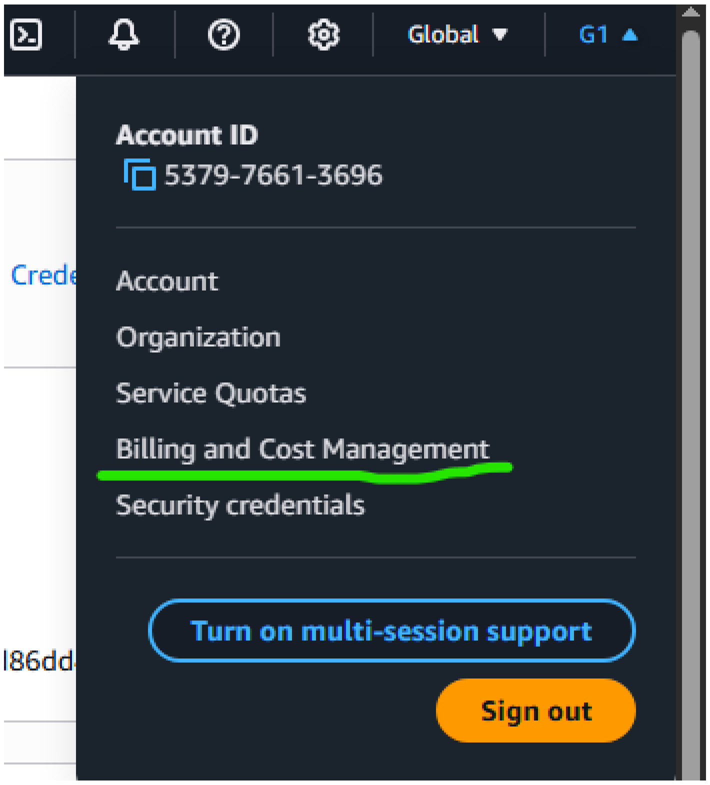
   Wait 30 seconds, and enter a second code entry.

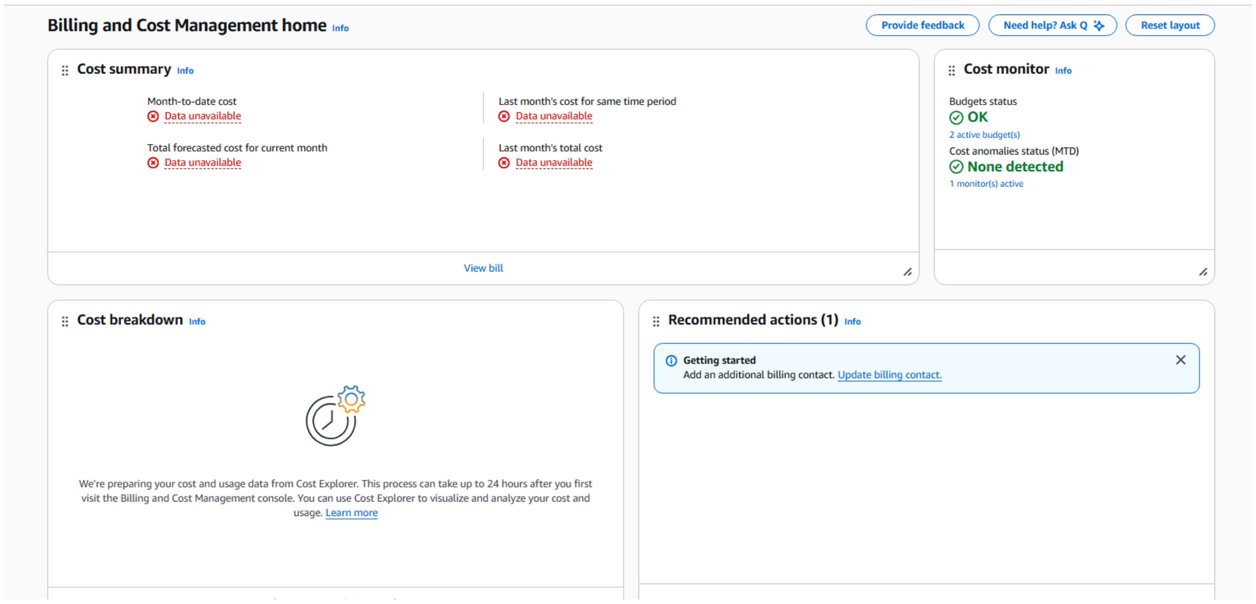
   MFA Code 2

- Enter 2 codes
- Click Add MFA

Check the Bill

- Login to console (Root)




Billing and Cost Management home Info

Provide feedback | Need help? Ask Q | Reset layout

Cost summary Info

Month-to-date cost
Data unavailable

Last month's cost for same time period
Data unavailable

Total forecasted cost for current month
Data unavailable

Last month's total cost
Data unavailable

View bill

Cost monitor Info

Budgets status
OK
2 active budget(s)

Cost anomalies status (MTD)
None detected
1 monitor(s) active

Cost breakdown Info

We're preparing your cost and usage data from Cost Explorer. This process can take up to 24 hours after you first visit the Billing and Cost Management console. You can use Cost Explorer to visualize and analyze your cost and usage. Learn more

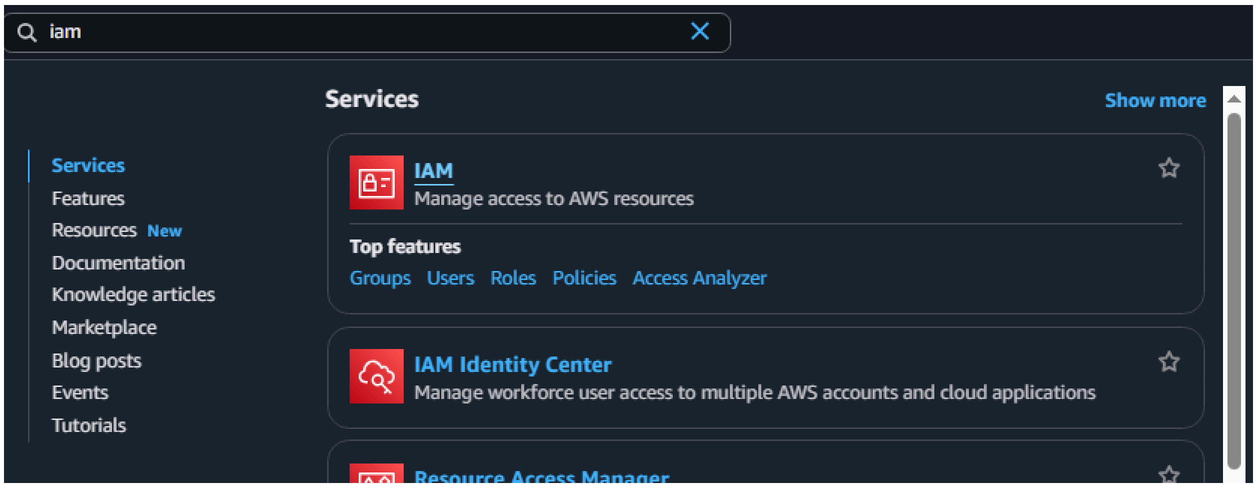
Recommended actions (1) Info

Getting started
Add an additional billing contact. Update billing contact.

🚩Create an IAM User (!!Extremely Fookin' IMP)

- It's Global service

1. Go to IAM



iam

Services

Show more

Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials

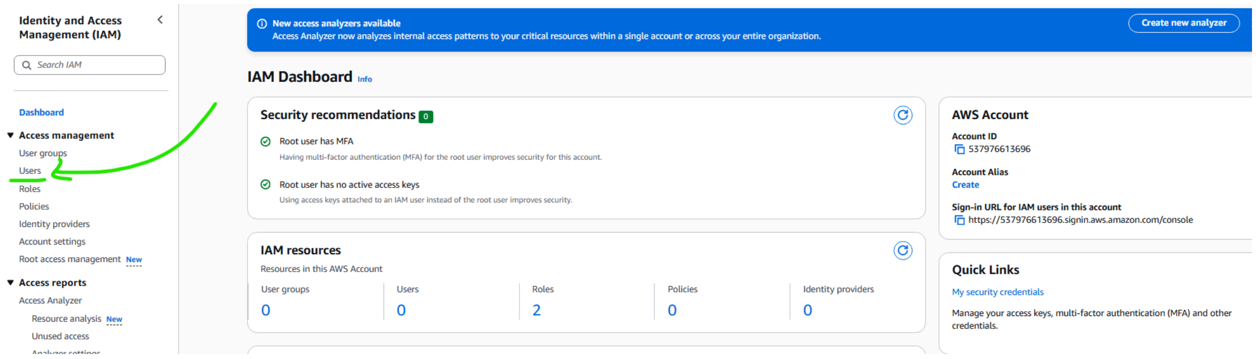
IAM
Manage access to AWS resources

Top features
Groups Users Roles Policies Access Analyzer

IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager

2. Click Users



Identity and Access Management (IAM)

Search IAM

Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access

New access analyzers available
Access Analyzer now analyzes internal access patterns to your critical resources within a single account or across your entire organization.

Create new analyzer

IAM Dashboard Info

Security recommendations 0

Root user has MFA
Having multi-factor authentication (MFA) for the root user improves security for this account.

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

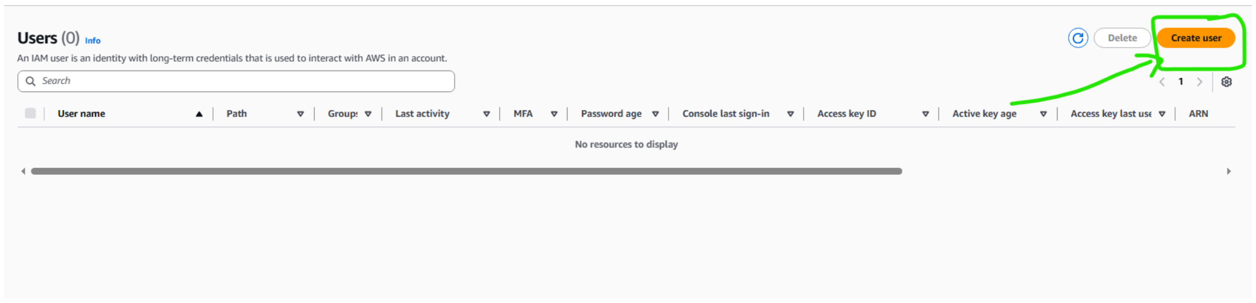
IAM resources
Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	2	0	0

AWS Account
Account ID
537976613696
Account Alias
Create
Sign-in URL for IAM users in this account
https://537976613696.signin.aws.amazon.com/console

Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

3. Create User



Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

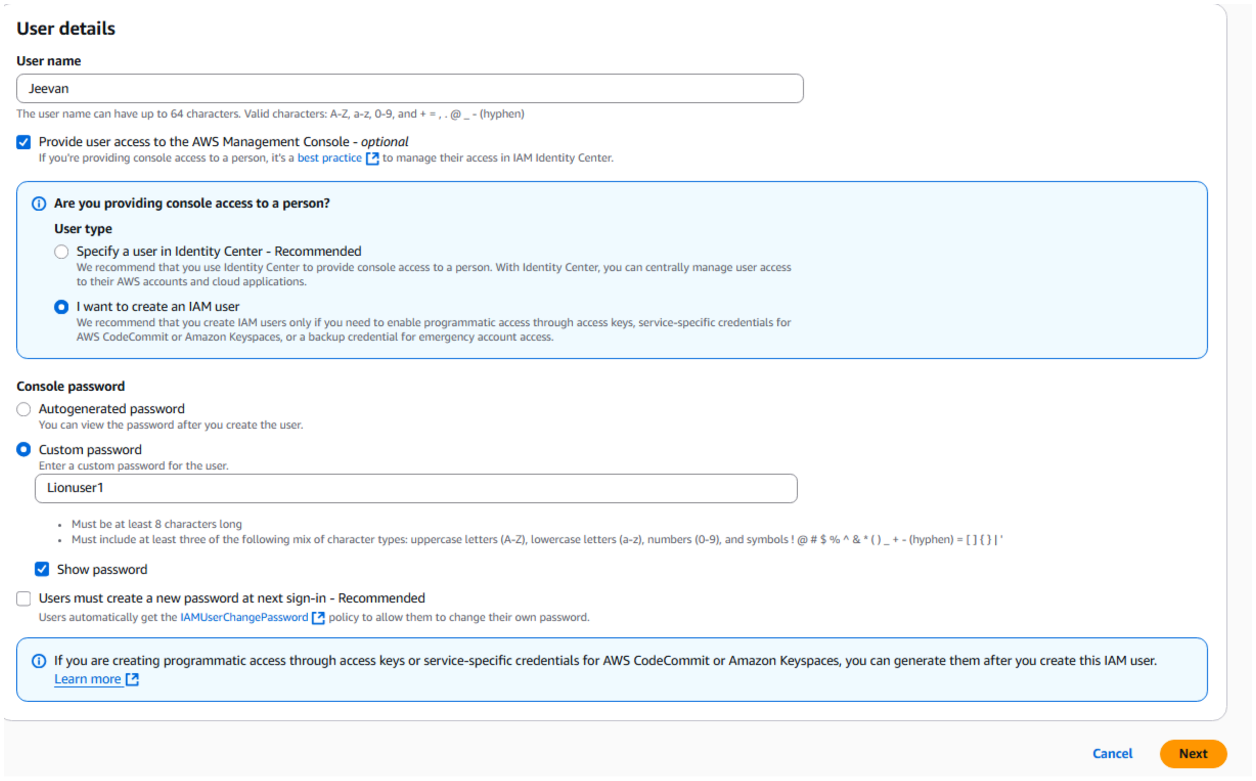
Delete | Create user

Search

User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key last use	ARN

No resources to display

1. Tick: Provide user access to the AWS Management Console
2. I want to create an IAM user
3. Custom password
   - Enter Password
4. Untick: Users must create a new password at next sign-in - Recommended



User details

User name

Jeevan

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

[✓] Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

( ) Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

(●) I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

( ) Autogenerated password
You can view the password after you create the user.

(●) Custom password
Enter a custom password for the user.

Lionuser1

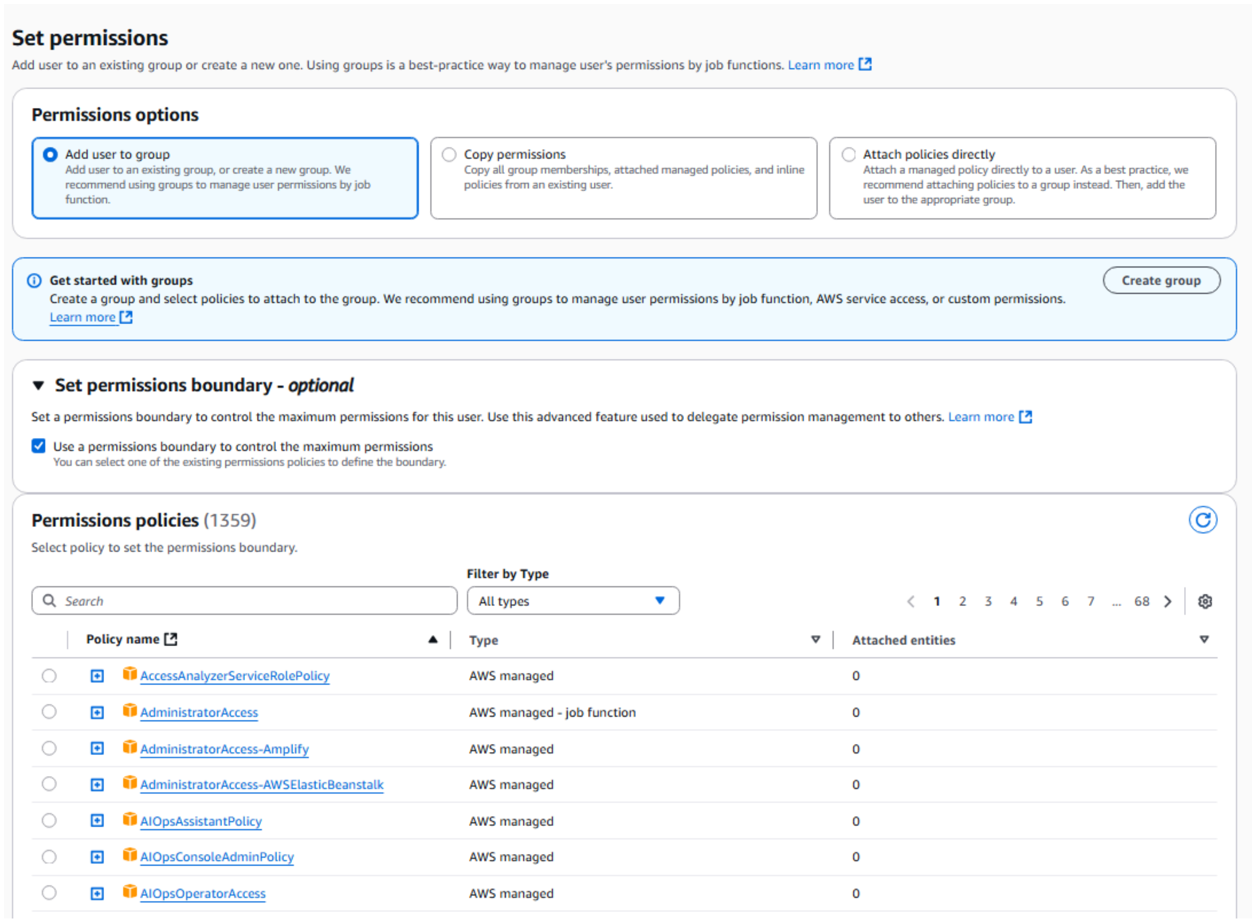
- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '

[✓] Show password

[ ] Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

4. Set Permissions



Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

- [✓ Add user to group] Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions: Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

Set permissions boundary - *optional*

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. Learn more

[✓] Use a permissions boundary to control the maximum permissions
You can select one of the existing permissions policies to define the boundary.

Permissions policies (1359)

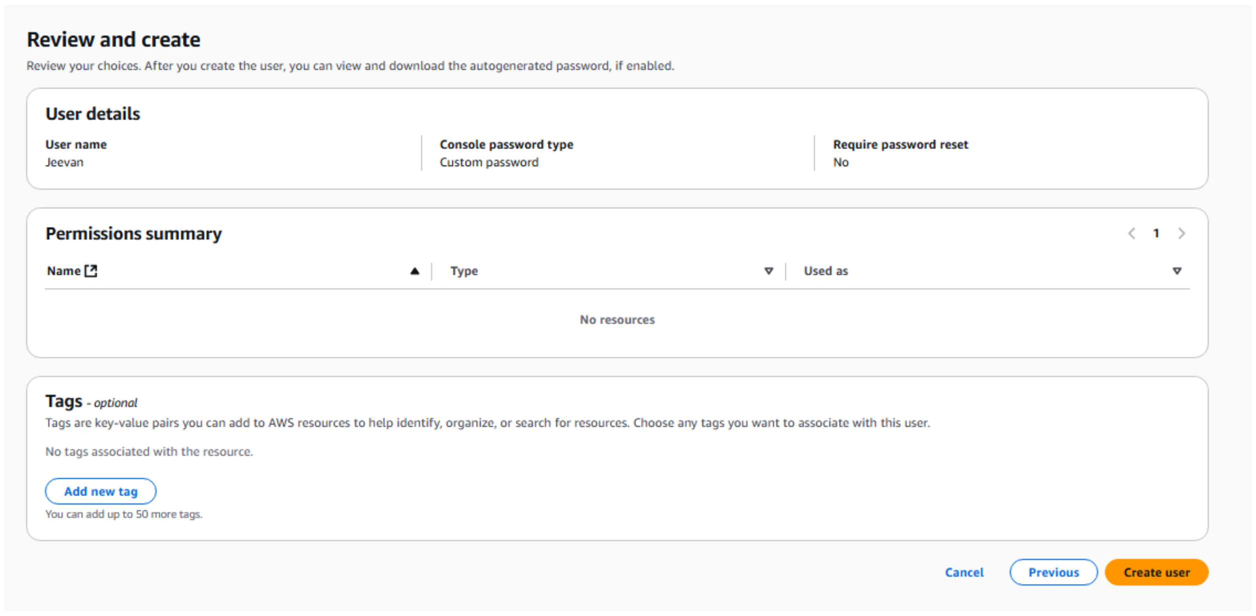
Select policy to set the permissions boundary.

Filter by Type: All types

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess	AWS managed - job function	0
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSElasticBeanstalk	AWS managed	0
AIOpsAssistantPolicy	AWS managed	0
AIOpsConsoleAdminPolicy	AWS managed	0
AIOpsOperatorAccess	AWS managed	0

> For now, we'll proceed without any permissions.

5. Create User


Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
Jeevan	Custom password	No

Permissions summary

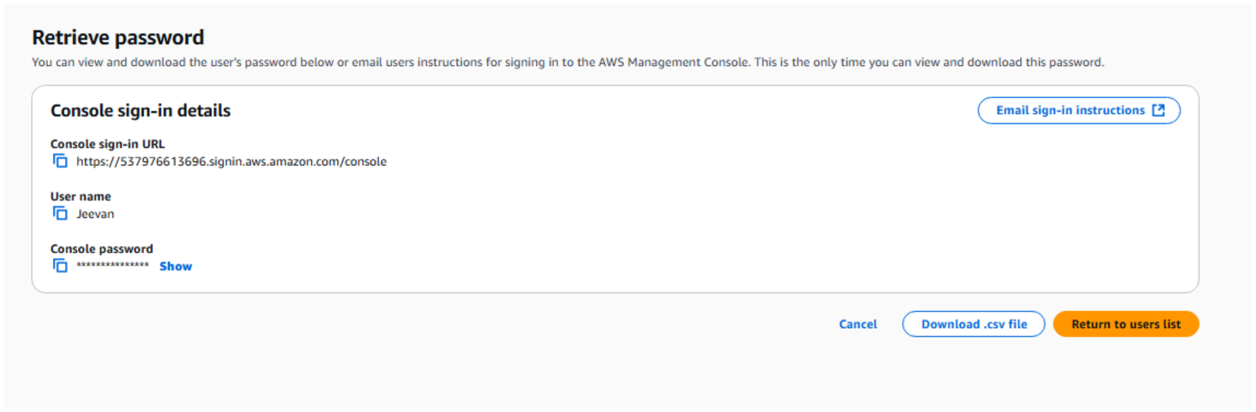
Name	Type	Used as
No resources		

Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.


Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
https://537976613696.signin.aws.amazon.com/console

User name
Jeevan

Console password
*************** Show

Console sign-in URL

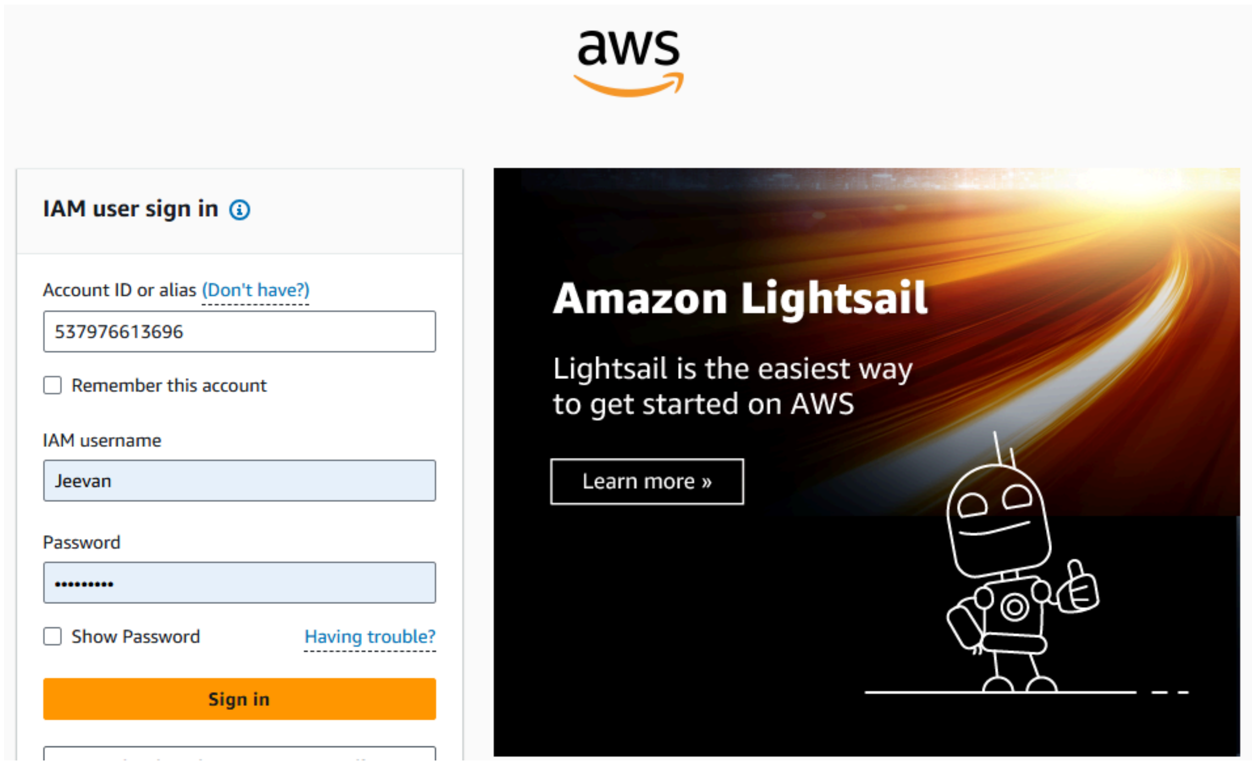
https://537976613696.signin.aws.amazon.com/console

User name

Jeevan

6. Use the link & Sign-in

https://537976613696.signin.aws.amazon.com/console



IAM user sign in

Account ID or alias (Don't have?)

537976613696

Remember this account

IAM username

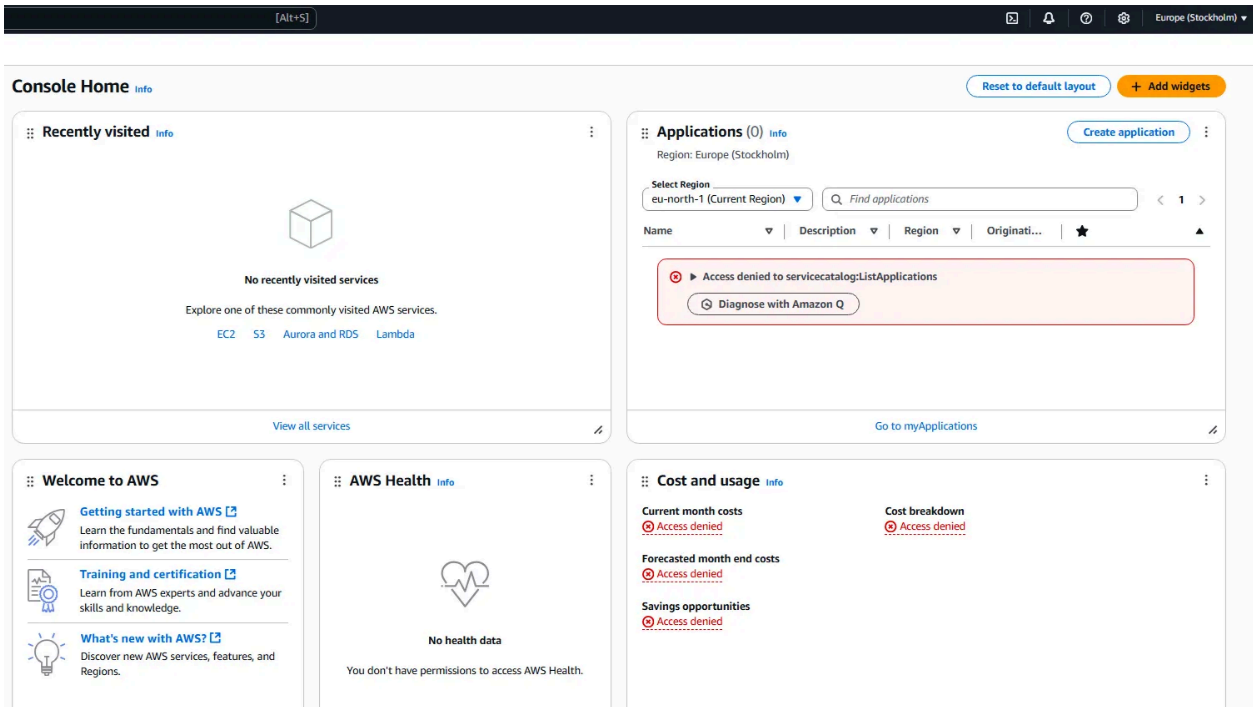
Jeevan

Password

•••••••••

Show Password   Having trouble?

Sign in



Console Home Info

Reset to default layout   + Add widgets

Recently visited Info

No recently visited services

Explore one of these commonly visited AWS services.

EC2   S3   Aurora and RDS   Lambda

View all services

Applications (0) Info   Create application

Region: Europe (Stockholm)

Select Region: eu-north-1 (Current Region)   Find applications   1

Name	Description	Region	Originati...

Access denied to servicecatalog:ListApplications

Diagnose with Amazon Q

Go to myApplications

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

No health data

You don't have permissions to access AWS Health.

Cost and usage Info

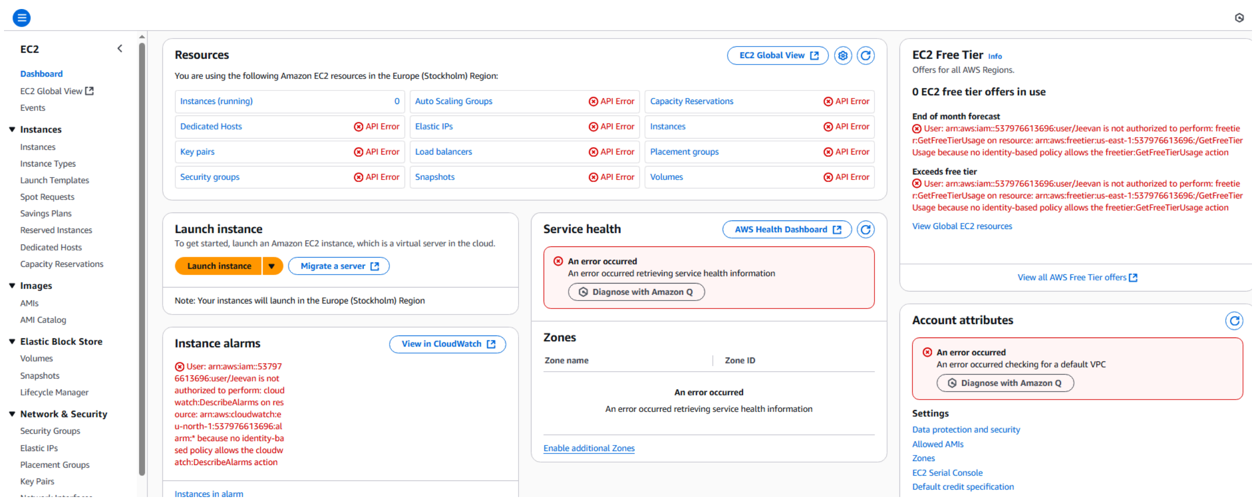
Current month costs
Access denied

Cost breakdown
Access denied

Forecasted month end costs
Access denied

Savings opportunities
Access denied

💡 **You do not have access to anything.**



Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	0	Auto Scaling Groups	API Error	Capacity Reservations	API Error
Dedicated Hosts	API Error	Elastic IPs	API Error	Instances	API Error
Key pairs	API Error	Load balancers	API Error	Placement groups	API Error
Security groups	API Error	Snapshots	API Error	Volumes	API Error

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Europe (Stockholm) Region

Instance alarms

User: arn:aws:iam::537976613696:user/Jeevan is not authorized to perform: cloudwatch:DescribeAlarms on resource: arn:aws:cloudwatch:eu-north-1:537976613696:alarm:* because no identity-based policy allows the cloudwatch:DescribeAlarms action

Instances in alarm

Service health

An error occurred
An error occurred retrieving service health information

Zones

Zone name	Zone ID

An error occurred
An error occurred retrieving service health information

Enable additional Zones

EC2 Free Tier Info
Offers for all AWS Regions.

0 EC2 free tier offers in use

End of month forecast
User: arn:aws:iam::537976613696:user/Jeevan is not authorized to perform: freetier:GetFreeTierUsage on resource: arn:aws:freetier:us-east-1:537976613696:/GetFreeTierUsage because no identity-based policy allows the freetier:GetFreeTierUsage action

Exceeds free tier
User: arn:aws:iam::537976613696:user/Jeevan is not authorized to perform: freetier:GetFreeTierUsage on resource: arn:aws:freetier:us-east-1:537976613696:/GetFreeTierUsage because no identity-based policy allows the freetier:GetFreeTierUsage action

View Global EC2 resources

View all AWS Free Tier offers

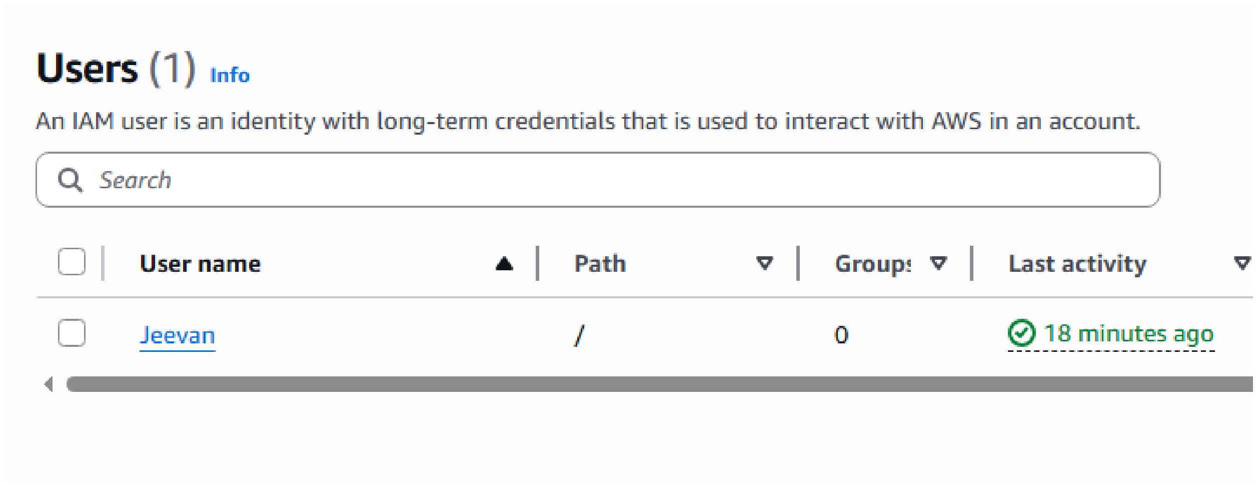
Account attributes

An error occurred
An error occurred checking for a default VPC

Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console
Default credit specification

7. Give permissions

- Login your root account
- IAM → Select user



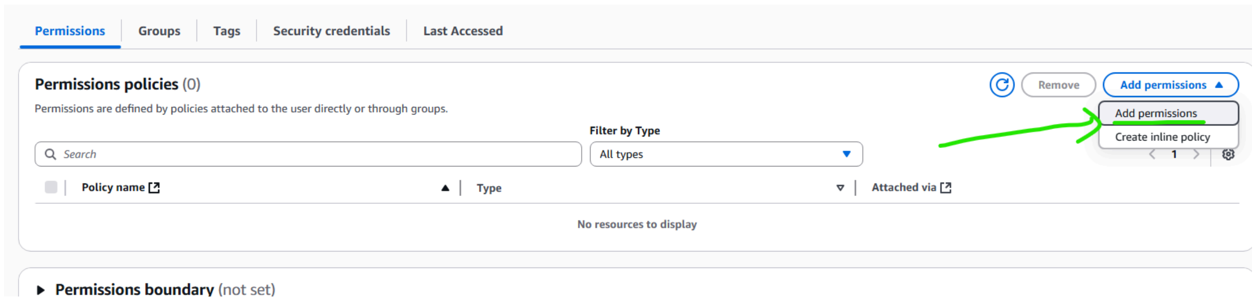
Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

User name	Path	Groups	Last activity
Jeevan	/	0	18 minutes ago

- Add permissions



Permissions policies (0)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
No resources to display		

Permissions boundary (not set)

- Attach policies directly

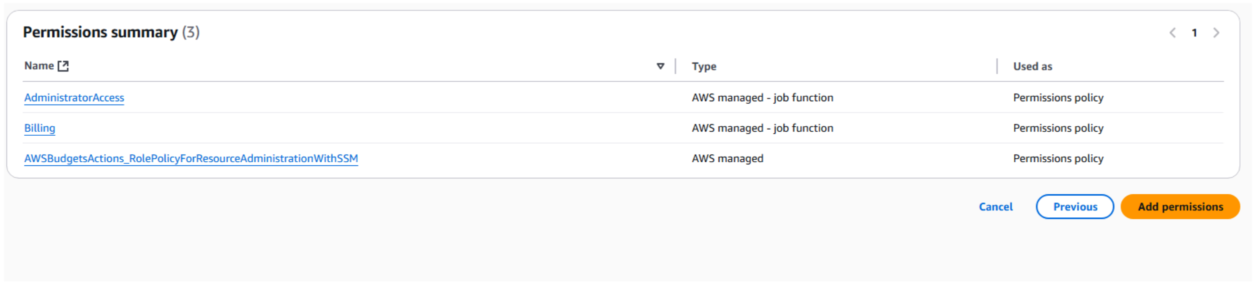
- **Select the permissions:**

  AdministratorAccess,

  Billing,

  AWSBudgetsActions_RolePolicyForResourceAdministrationWithSSM

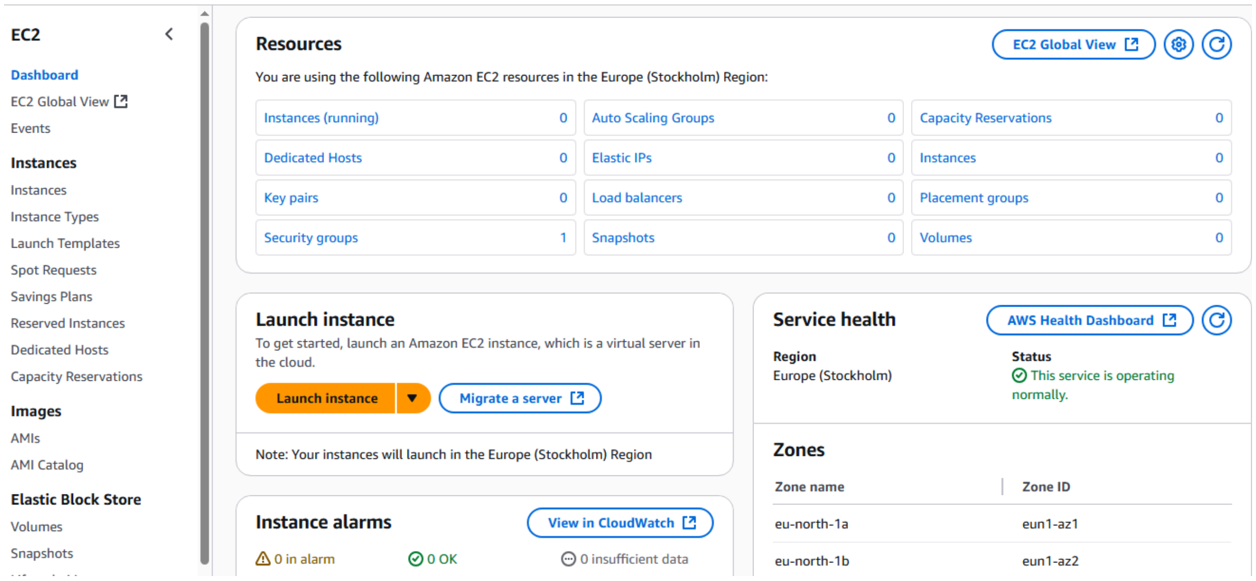
- Add permissions



Permissions summary (3)

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
Billing	AWS managed - job function	Permissions policy
AWSBudgetsActions_RolePolicyForResourceAdministrationWithSSM	AWS managed	Permissions policy

8. Now login to your IAM User



Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	0	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Europe (Stockholm) Region

Service health

Region: Europe (Stockholm)

Status: This service is operating normally.

Zones

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2

Instance alarms

0 in alarm · 0 OK · 0 insufficient data

You'll see the EC2 instance

Even with `AdministratorAccess`, an IAM user does NOT have access to the `aws-portal` (billing) or other root-specific actions by default.