AWS IAM User Management and EC2 Permission Restriction

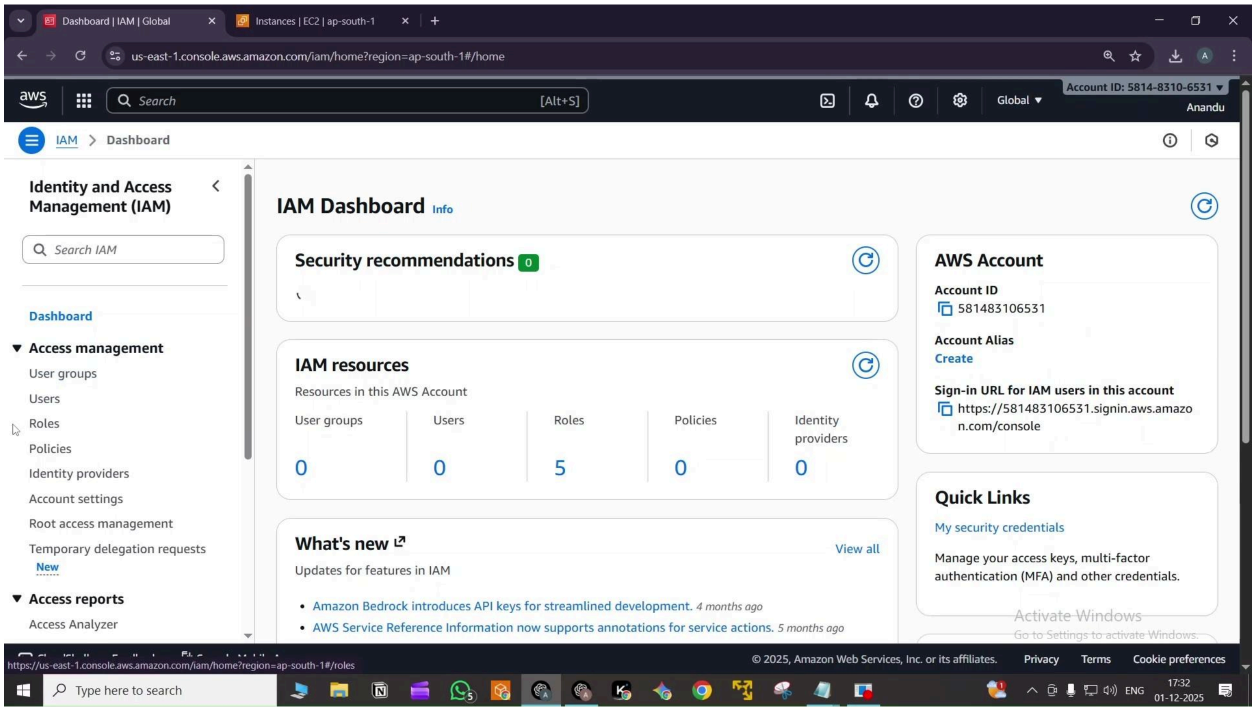
INTRODUCTION
Identity and Access Management (IAM) is one of the most critical security components in Amazon Web Services (AWS). It enables organizations to securely control who can access specific AWS services and resources. In real cloud environments, using the root account for day-to-day tasks is considered unsafe. Instead, IAM users with restricted privileges are created following the “least privilege principle”.

This project focuses on creating and managing IAM users with limited access to Amazon EC2 (Elastic Compute Cloud). The objective was to configure two different types of IAM users — one with AWS Management Console access and another with Programmatic (CLI) access — ensuring that both users are restricted strictly to EC2 operations while being denied access to all other AWS services.

The tasks performed include creating users, assigning custom IAM policies, generating Access Keys, logging in through the AWS console, configuring authentication through AWS CLI, and verifying restricted access. All actions were validated with screenshots, and the report documents each step in detail.

1. IAM Dashboard Overview

The process begins by accessing the **IAM Dashboard** from the AWS root account. The IAM dashboard provides access to user management, policies, roles, groups, and security analytics. From here, new users and access policies can be created securely.


Dashboard | IAM | Global
Instances | EC2 | ap-south-1
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/home
Search
[Alt+S]
Global
Account ID: 5814-8310-6531
Anandu
IAM > Dashboard

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management
- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests New

Access reports
- Access Analyzer

IAM Dashboard Info

Security recommendations 0

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	5	0	0

What's new

View all

Updates for features in IAM

- Amazon Bedrock introduces API keys for streamlined development. *4 months ago*
- AWS Service Reference Information now supports annotations for service actions. *5 months ago*

AWS Account

Account ID
581483106531

Account Alias
Create

Sign-in URL for IAM users in this account
https://581483106531.signin.aws.amazon.com/console

Quick Links

My security credentials

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

Activate Windows
Go to Settings to activate Windows.
https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/roles
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
ENG
17:32
01-12-2025

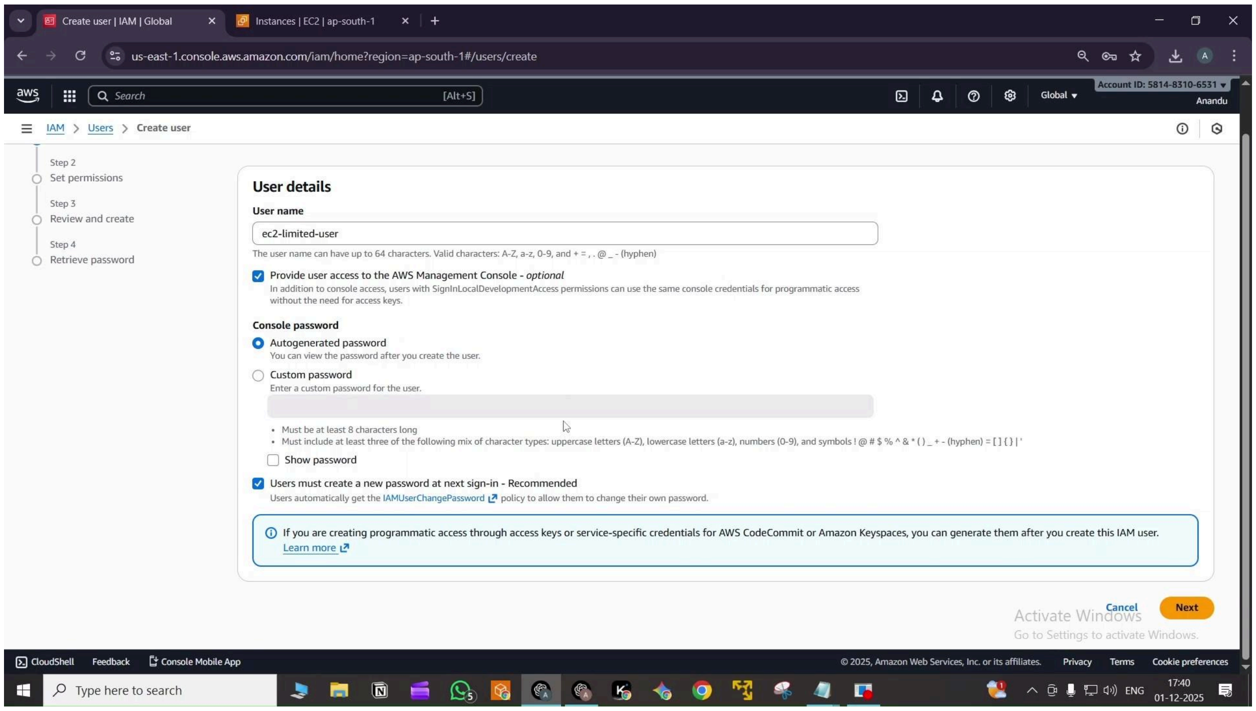
1.IAM dashboard

2. Creating an IAM User (Management Console Access)

A new IAM user was created through the **Users** → **Create User** flow.
During creation:

- A username was assigned.
- **Management Console Access** was enabled.
- A custom password was generated.
- The "Require password reset" option was unchecked, allowing immediate login.

This user is intended to log in through the AWS console using a username and password.


us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create

IAM > Users > Create user

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

User details

User name

ec2-limited-user

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*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

Console password

[✓ Autogenerated password]
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

- 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

Create user | IAM | Global
Instances | EC2 | ap-south-1
Search [Alt+S]
Global
Account ID: 5814-8310-6531
Anandu
Cancel
Next
Activate Windows
Go to Settings to activate Windows.
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
ENG
17:40
01-12-2025

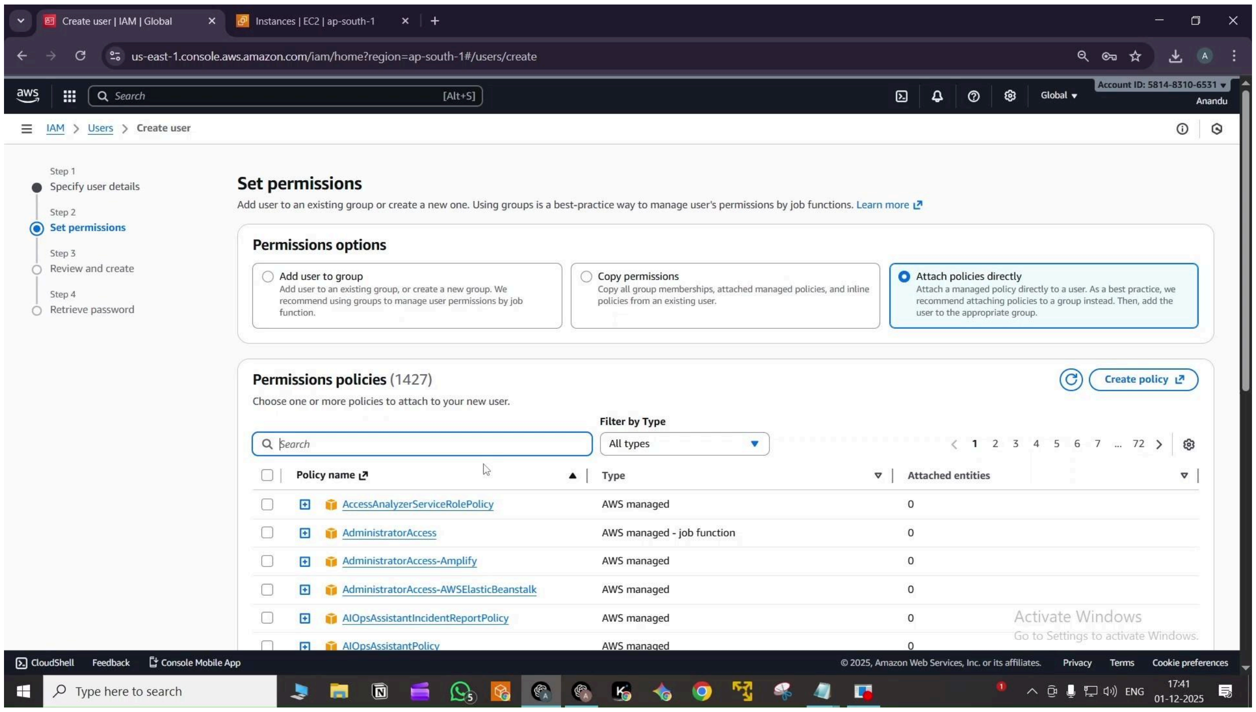
2.IAM user creation

3. Creating a Custom EC2-Only Access Policy

To follow the least-privilege model, a **custom IAM policy** was created manually under **Policies → Create policy**.A JSON policy was added with the following permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    }
  ]
}
```

Meaning:
 This allows the user to perform **all EC2 operations only**, while every other AWS service is denied.



Create user | IAM | Global
Instances | EC2 | ap-south-1
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create
Search [Alt+S]
Global
Account ID: 5814-8310-6531
Anandu
IAM > Users > Create user

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password

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.

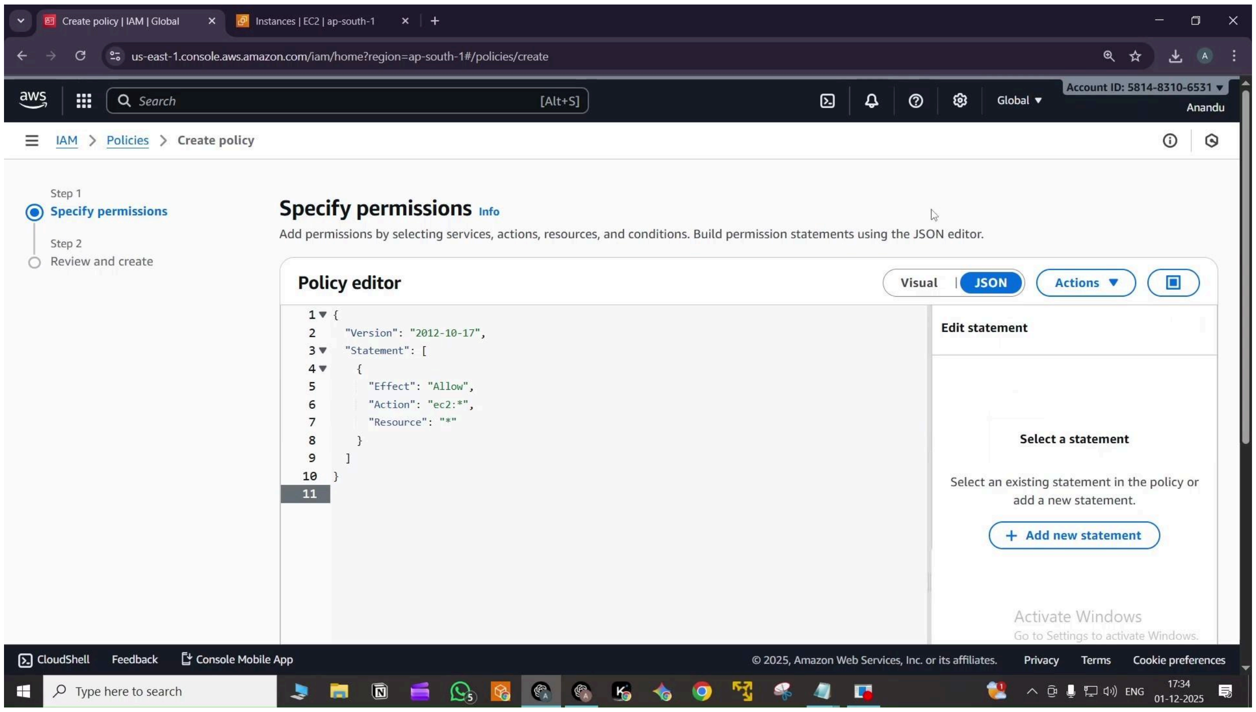
Permissions policies (1427)

Choose one or more policies to attach to your new user.

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
AIOpsAssistantIncidentReportPolicy	AWS managed	0
AIOpsAssistantPolicy	AWS managed	0

3.creating a policy



Create policy | IAM | Global
Instances | EC2 | ap-south-1
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/policies/create
Search [Alt+S]
Global
Account ID: 5814-8310-6531
Anandu
IAM > Policies > Create policy

Step 1
Specify permissions
Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    }
  ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

3.1.json policy

4. Attaching the EC2-Full-Access-Only Policy

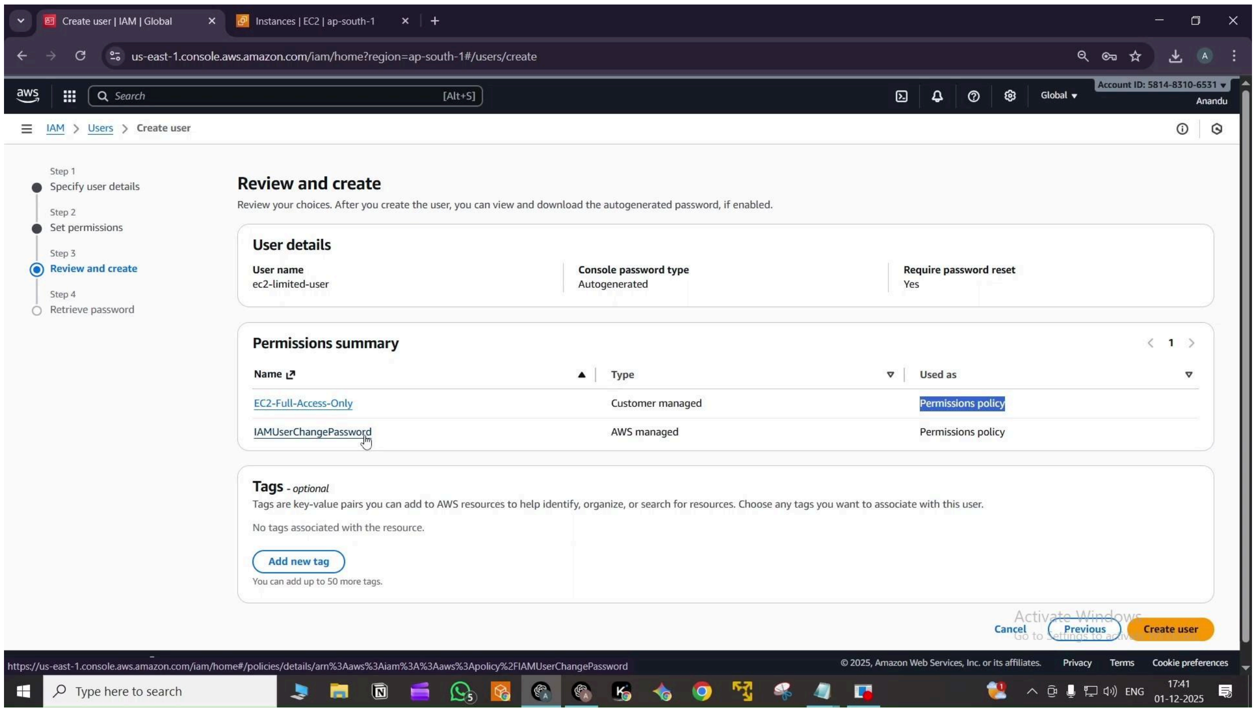
After creating the policy, it was successfully attached to the IAM user.
This ensured that the user can:

✔ Access EC2 Dashboard
✔ Launch, start, stop EC2 instances

But cannot:

✖ Open S3
✖ Access IAM
✖ Use Lambda or other AWS services

This completes the console user restriction setup.



IAM > Users > Create user

Step 1 Specify user details
Step 2 Set permissions
Step 3 Review and create
Step 4 Retrieve password

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
ec2-limited-user	Autogenerated	Yes

Permissions summary

Name	Type	Used as
EC2-Full-Access-Only	Customer managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

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.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

https://us-east-1.console.aws.amazon.com/iam/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FIAMUserChangePassword

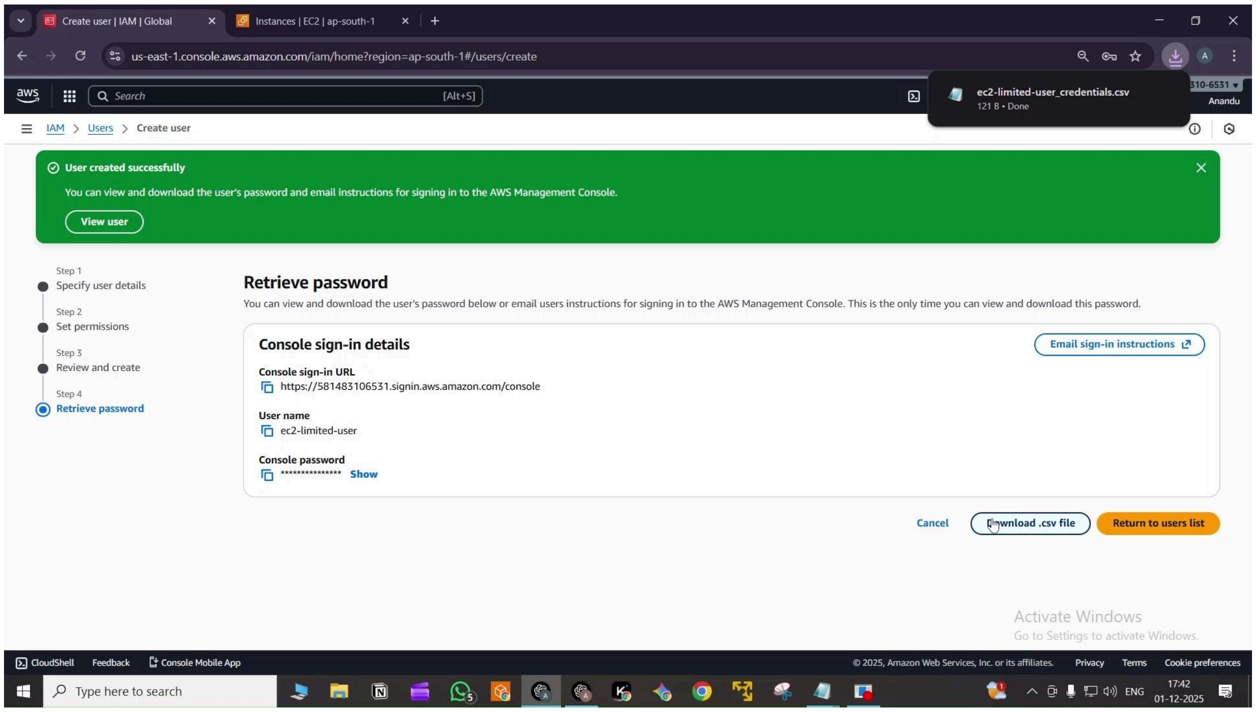
4.policy created and creating the user

5. Downloading the Login Credentials (.csv File)

After user creation, the **.csv file** containing the username and password was downloaded.
 This file contains:

- Username
- Console login URL
- Temporary or generated password

These credentials were later used to log in as the restricted IAM user.


Create user | IAM | Global
Instances | EC2 | ap-south-1
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create
ec2-limited-user_credentials.csv
121 B • Done
Search [Alt+S]
Anandu
IAM > Users > Create user

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password

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://581483106531.signin.aws.amazon.com/console

User name
ec2-limited-user

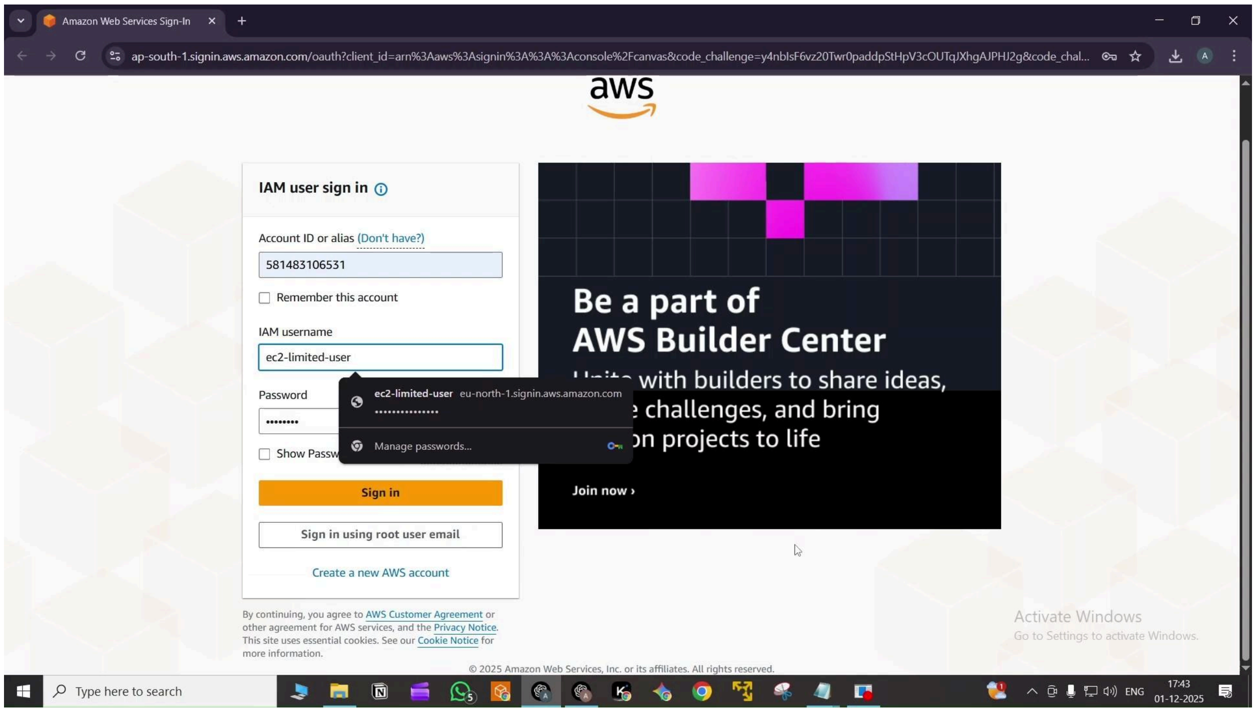
Console password
*************** Show

5.successfully created a user

6. Logging in as the Restricted IAM User

Using the credentials from the `.csv` file, login was performed through the IAM console URL.

The user interface displayed fewer services compared to the root user, confirming that permissions were limited. Only EC2-related features were accessible, and access to all other services was denied.


Amazon Web Services Sign-In

ap-south-1.signin.aws.amazon.com/oauth?client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&code_challenge=y4nblsF6vz20Twr0paddpStHpV3cOUTqJXhgAJPHJ2g&code_chal...

IAM user sign in

Account ID or alias (Don't have?)

581483106531

Remember this account

IAM username

ec2-limited-user

Password

••••••••

Show Password

ec2-limited-user eu-north-1.signin.aws.amazon.com

Manage passwords...

Sign in

Sign in using root user email

Create a new AWS account

By continuing, you agree to AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

Be a part of AWS Builder Center

Unite with builders to share ideas, challenges, and bring on projects to life

Join now ›

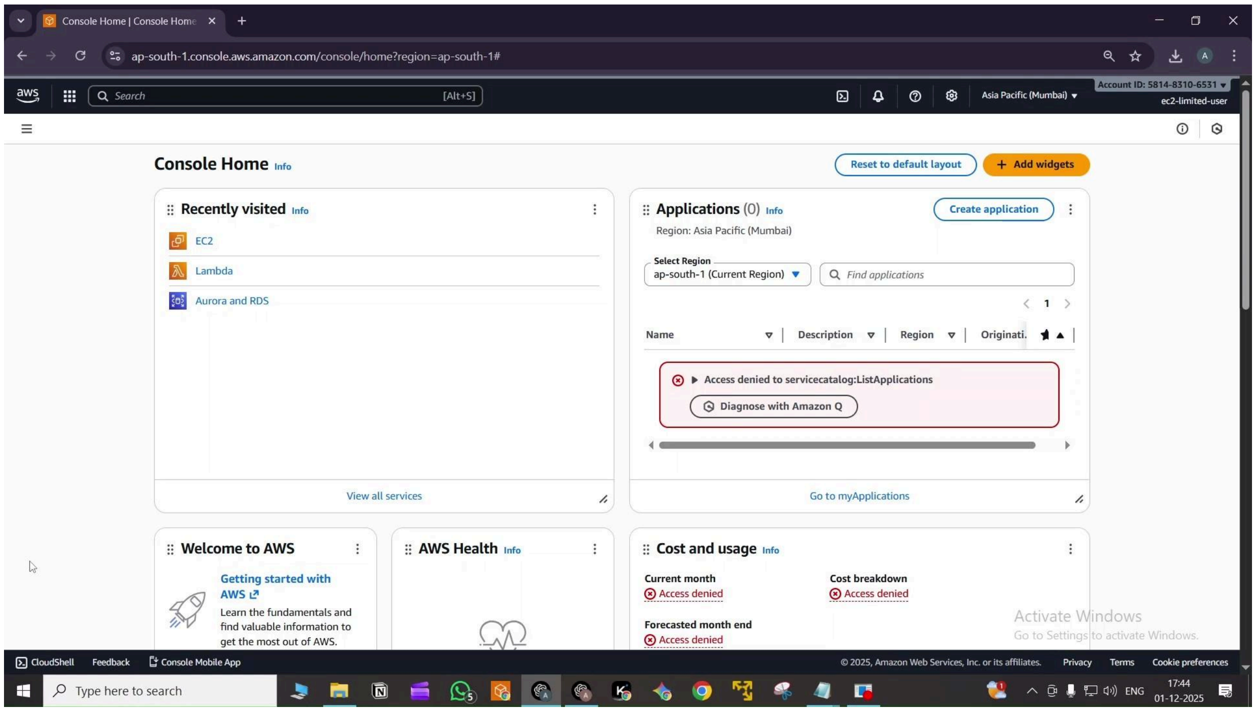
© 2025 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activate Windows
Go to Settings to activate Windows.

Type here to search

17:43 01-12-2025

6.login as IAM user


Console Home | Console Home

ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1#

Search [Alt+S]

Asia Pacific (Mumbai)

Account ID: 5814-8310-6531

ec2-limited-user

Console Home Info

Reset to default layout

Add widgets

Recently visited Info

EC2

Lambda

Aurora and RDS

View all services

Applications (0) Info

Create application

Region: Asia Pacific (Mumbai)

Select Region

ap-south-1 (Current Region)

Find applications

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.

AWS Health Info

Cost and usage Info

Current month
Access denied

Cost breakdown
Access denied

Forecasted month end
Access denied

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

17:44 01-12-2025

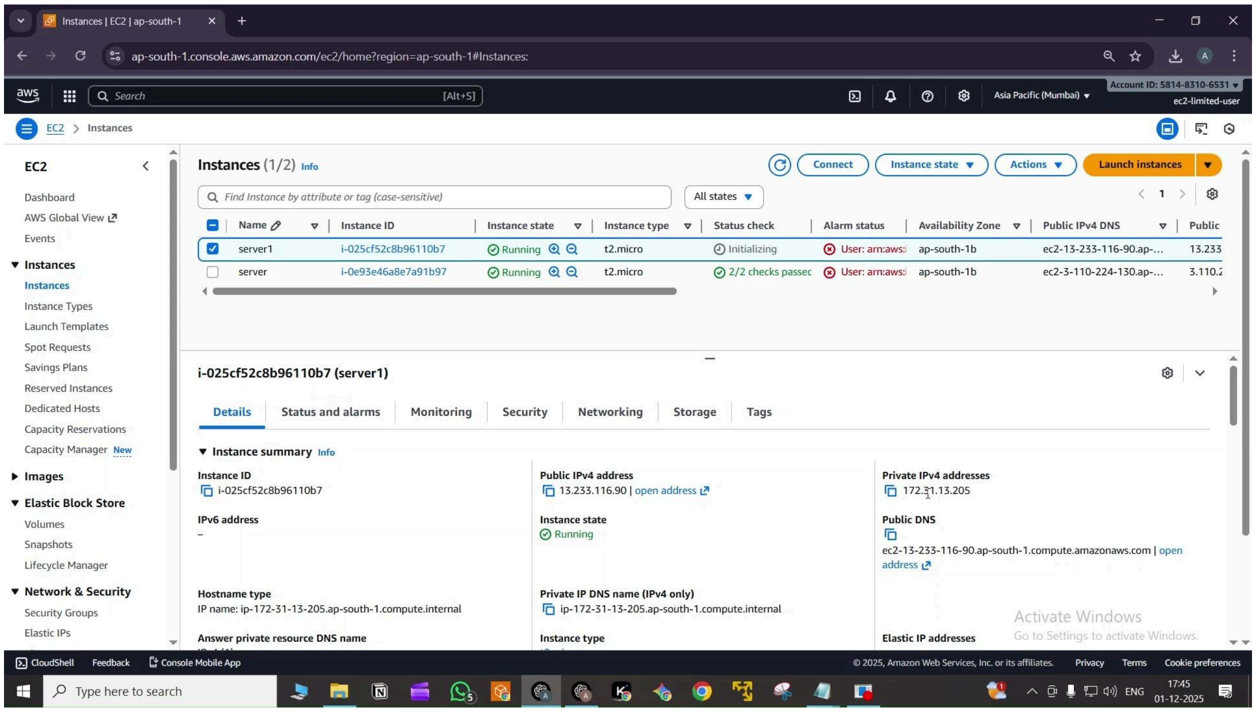
7. Launching an EC2 Instance

From the restricted user console:

- EC2 Dashboard was accessed

- An instance was launched successfully

This verified that the **EC2-only policy** was functioning correctly.



Instances | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

Account ID: 5814-8310-6531 — ec2-limited-user — Asia Pacific (Mumbai)

EC2 > Instances

Instances (1/2)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
✓	server1	i-025cf52c8b96110b7	Running	t2.micro	Initializing	User: arn:aws:	ap-south-1b	ec2-13-233-116-90.ap-...	13.233
	server	i-0e93e46a8e7a91b97	Running	t2.micro	2/2 checks passec	User: arn:aws:	ap-south-1b	ec2-3-110-224-130.ap-...	3.110.2

i-025cf52c8b96110b7 (server1)

Instance summary

Instance ID: i-025cf52c8b96110b7	Public IPv4 address: 13.233.116.90 | open address	Private IPv4 addresses: 172.31.13.205
IPv6 address: –	Instance state: Running	Public DNS: ec2-13-233-116-90.ap-south-1.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-13-205.ap-south-1.compute.internal	Private IP DNS name (IPv4 only): ip-172-31-13-205.ap-south-1.compute.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses

7.launching an instance

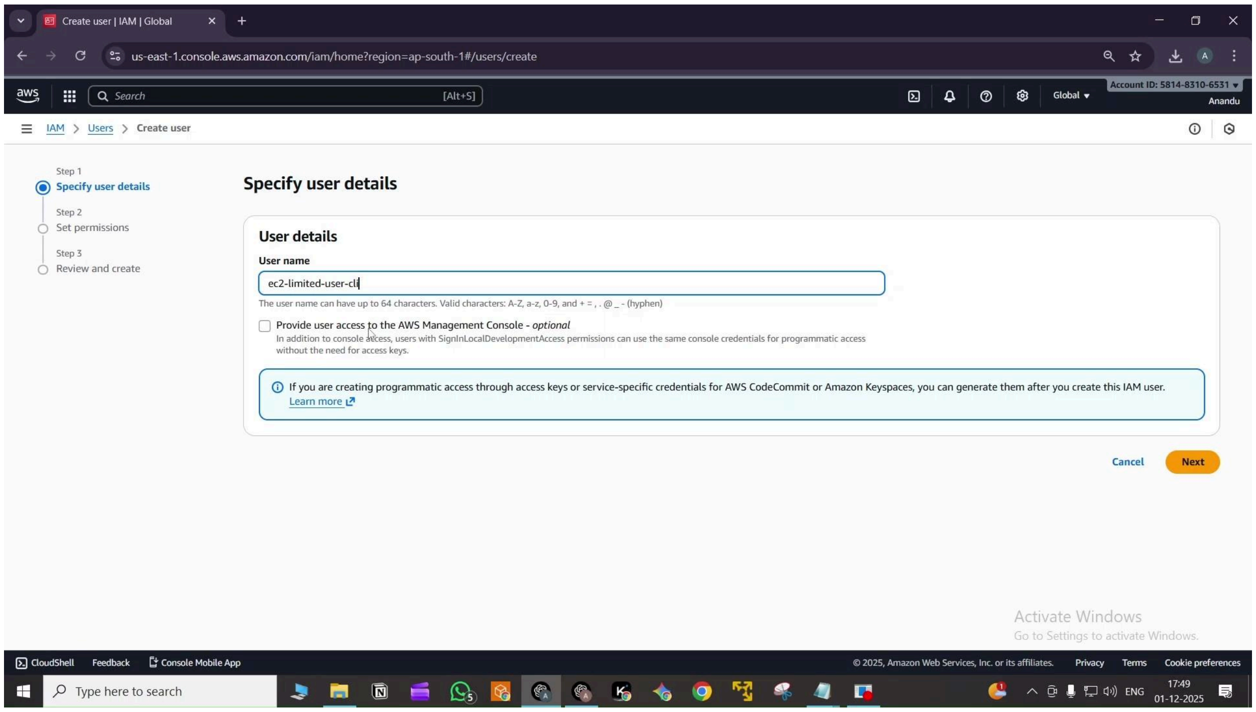
8. Creating a Programmatic IAM User (No Console Access)

A second IAM user was created specifically for CLI usage.

During creation:

- The option **Management Console Access** was left **unchecked**.
- Only **Programmatic Access** was enabled, allowing use via AWS CLI or SDK.

This user cannot log in from the AWS website.


us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create

IAM > Users > Create user

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create

Specify user details

User details

User name

ec2-limited-user-cli

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*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

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

8.Programmatic IAM User

9. Creating an Access Key for CLI

Under **Security Credentials**, a new **Access Key** was generated.

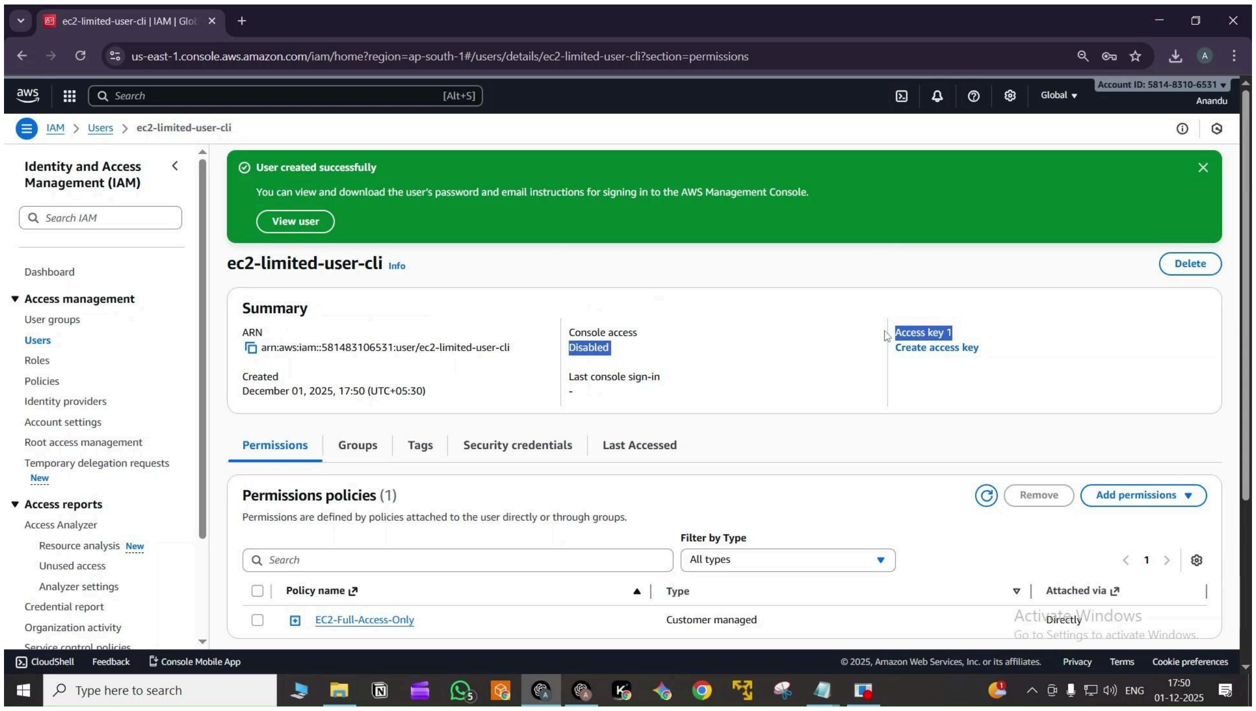
- Login method selected: **CLI (Command Line Interface)**

- A new key pair was created and downloaded as a `.csv` file

This file contains:

- Access Key ID
- Secret Access Key

These keys are required for AWS CLI authentication.


IAM > Users > ec2-limited-user-cli

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

ec2-limited-user-cli

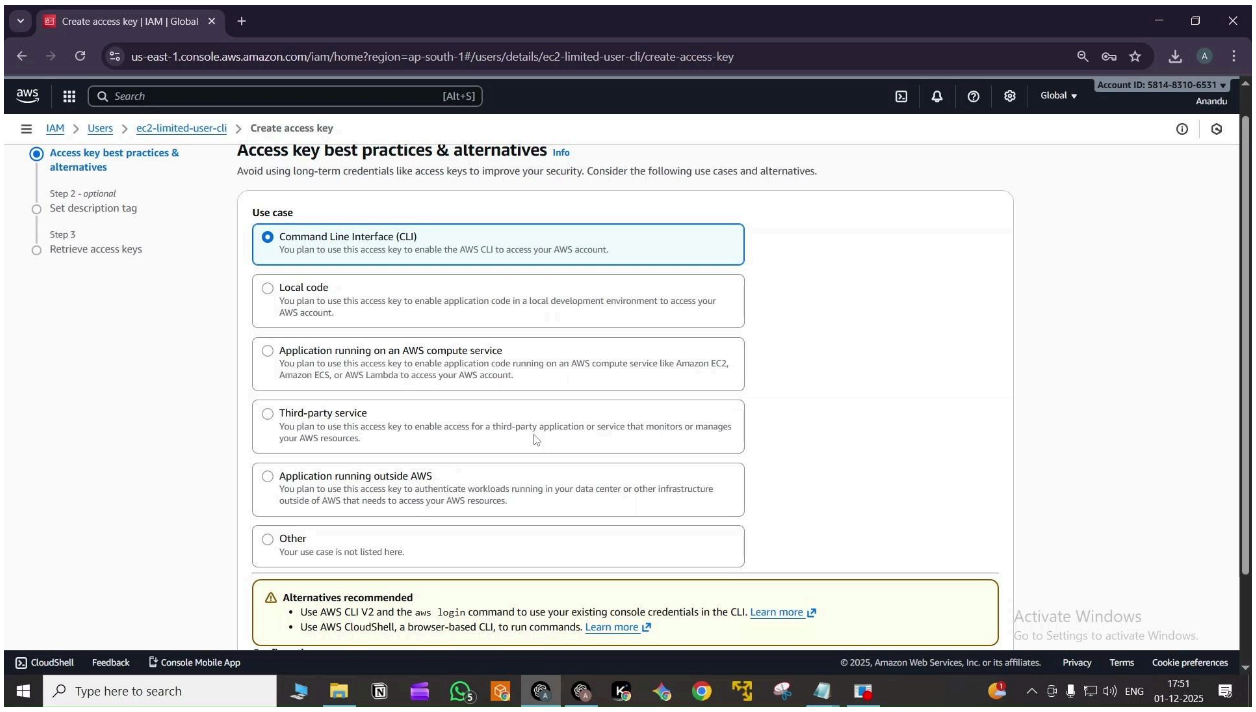
Summary

ARN: arn:aws:iam::581483106531:user/ec2-limited-user-cli
Created: December 01, 2025, 17:50 (UTC+05:30)
Console access: Disabled
Last console sign-in: -
Access key 1: Create access key

Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
EC2-Full-Access-Only	Customer managed	Directly

9.Access Key

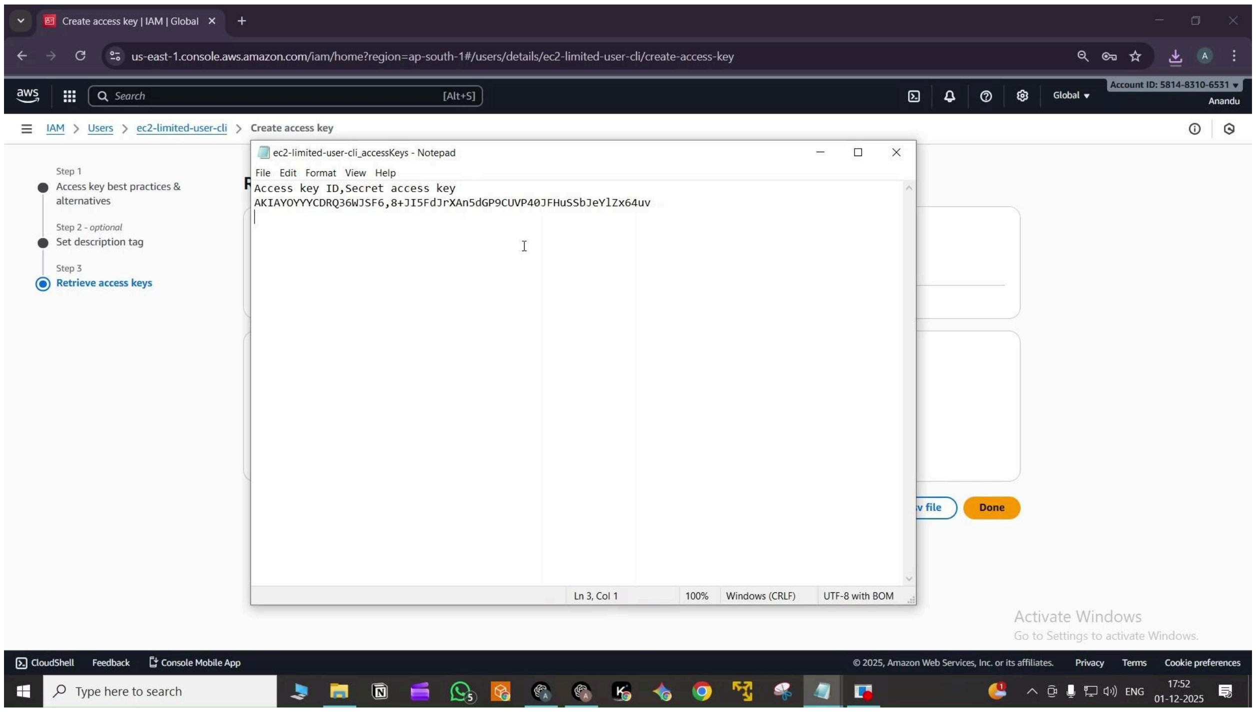

IAM > Users > ec2-limited-user-cli > Create access key

Access key best practices & alternatives
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case
- [✓ Command Line Interface (CLI)] You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code: You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service: You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service: You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS: You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other: Your use case is not listed here.

Alternatives recommended
- Use AWS CLI V2 and the `aws login` command to use your existing console credentials in the CLI. Learn more
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more

9.1.CLI



```
Access key ID,Secret access key
AKIAYOYYYCDRQ36WJSF6,8+JI5FdJrXAn5dGP9CUVP40JFHuSSbJeYlZx64uv
```

Create access key | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/ec2-limited-user-cli/create-access-key
IAM > Users > ec2-limited-user-cli > Create access key
ec2-limited-user-cli_accessKeys - Notepad
File Edit Format View Help
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Done
Ln 3, Col 1
100%
Windows (CRLF)
UTF-8 with BOM

9.2.*(.csv)file

10. AWS CLI Configuration (Using PuTTY → EC2 Instance)

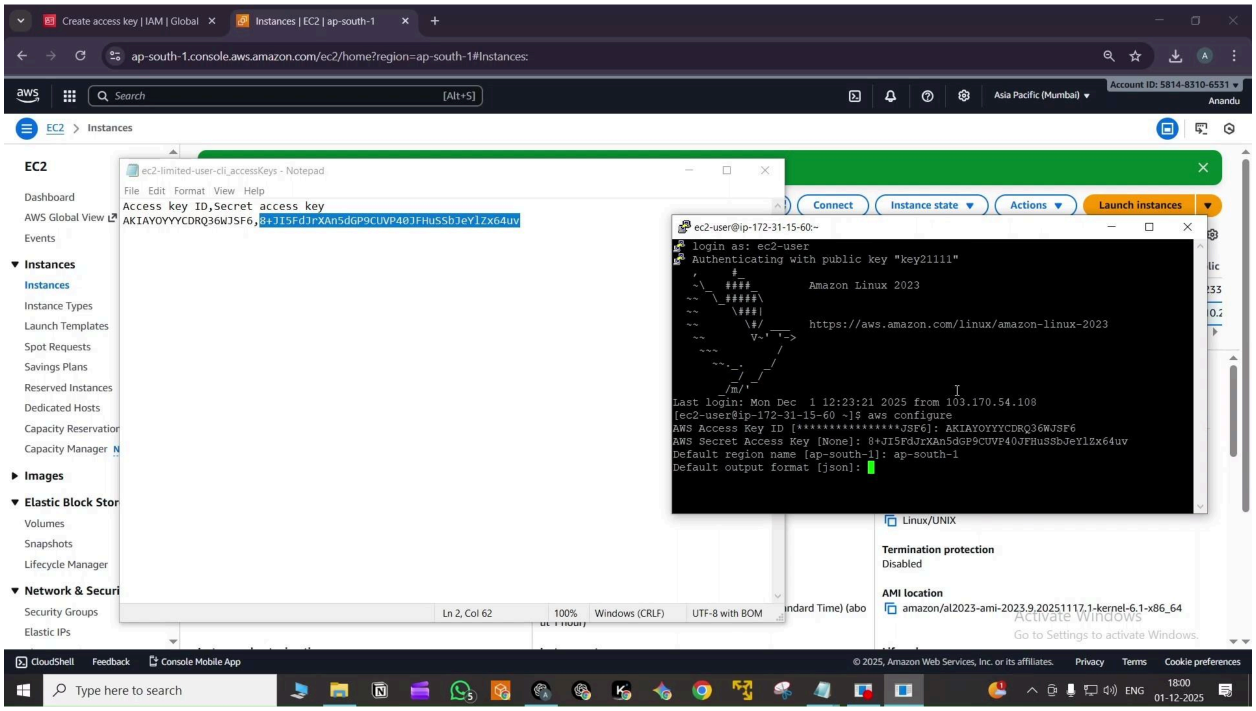
Using a PuTTY connection to an EC2 instance:The following command was executed:

```
aws configure
```

This step required entering:

- Access Key ID
- Secret Access Key
- Default region (ap-south-1)
- Output format (json)

AWS CLI was then authenticated as the programmatic user.



```
Access key ID,Secret access key
AKIAYOYYYCDRQ36WJSF6,8+JI5FdJrXAn5dGP9CUVP40JFHuSSbJeYlZx64uv
```

```
login as: ec2-user
Authenticating with public key "key21111"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Dec  1 12:23:21 2025 from 103.170.54.108
[ec2-user@ip-172-31-15-60 ~]$ aws configure
AWS Access Key ID [***************JSF6]: AKIAYOYYYCDRQ36WJSF6
AWS Secret Access Key [None]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]:
```

10.accesing user via putty

11. Verifying Access Through AWS CLI

The identity of the logged-in CLI user was verified using:

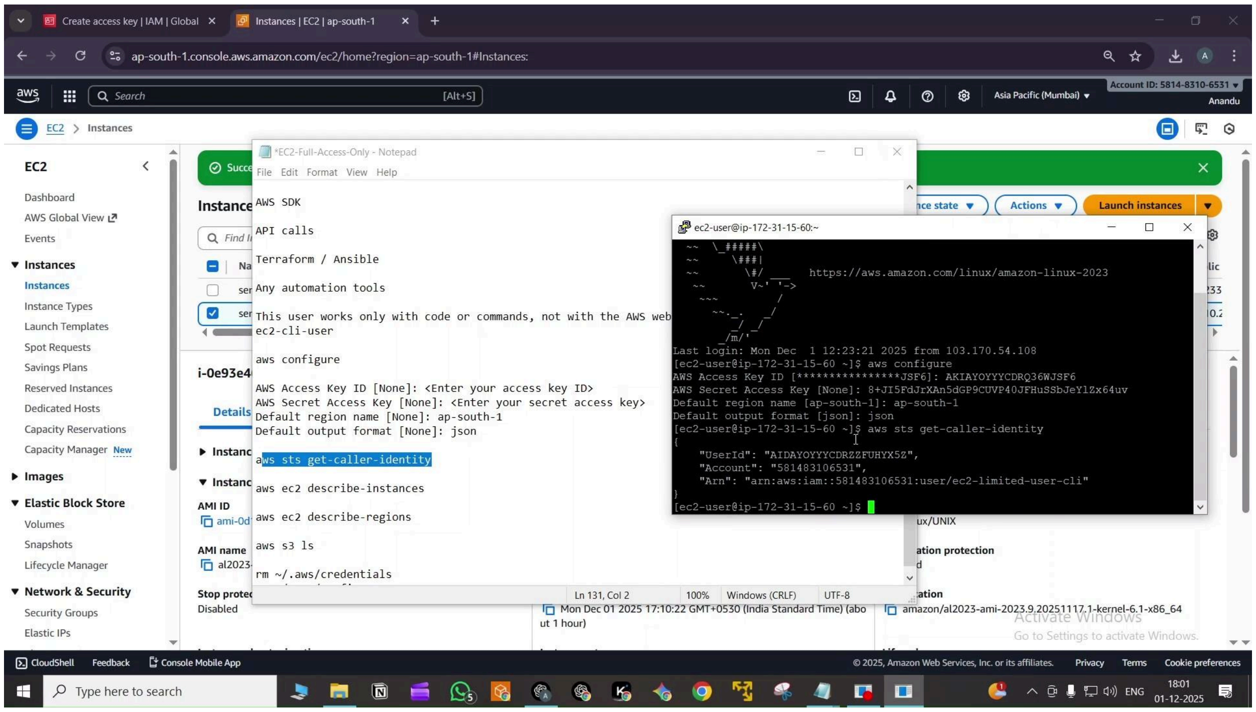
```
aws sts get-caller-identity
```

The output confirmed that the programmatic user was active and authenticated.

Further tests:

✔ `aws ec2 describe-instances` → Worked
❌ `aws s3 ls` → AccessDenied (expected)

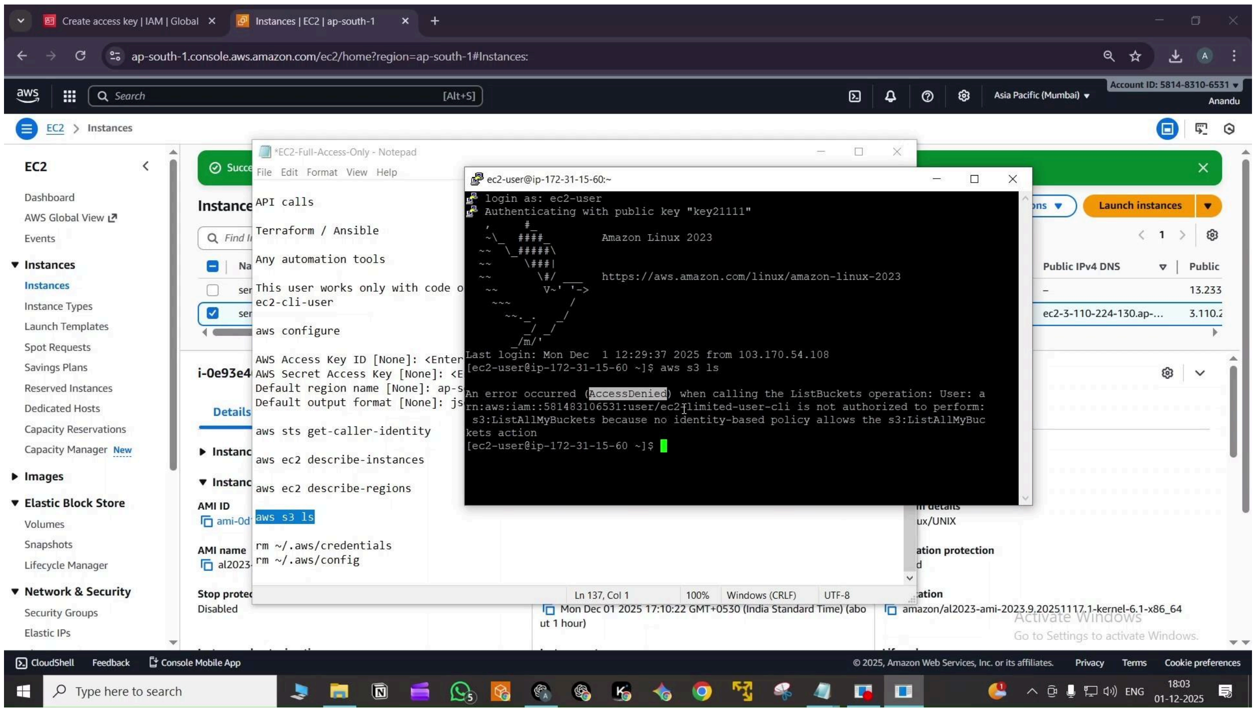
This validated the restricted permissions.



```
AWS SDK

API calls

Terraform / Ansible

Any automation tools

This user works only with code or commands, not with the AWS web
ec2-cli-user

aws configure

AWS Access Key ID [None]: <Enter your access key ID>
AWS Secret Access Key [None]: <Enter your secret access key>
Default region name [None]: ap-south-1
Default output format [None]: json

aws sts get-caller-identity

aws ec2 describe-instances

aws ec2 describe-regions

aws s3 ls

rm ~/.aws/credentials
```

```
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Dec  1 12:23:21 2025 from 103.170.54.108
[ec2-user@ip-172-31-15-60 ~]$ aws configure
AWS Access Key ID [****************JSF6]: AKIAYOYYYCDRQ36WJSF6
AWS Secret Access Key [None]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
[ec2-user@ip-172-31-15-60 ~]$ aws sts get-caller-identity
{
    "UserId": "AIDAYOYYYCDRZZFUHYX5Z",
    "Account": "581483106531",
    "Arn": "arn:aws:iam::581483106531:user/ec2-limited-user-cli"
}
[ec2-user@ip-172-31-15-60 ~]$
```

11.1.command-aws sts get-caller-identity



```
API calls

Terraform / Ansible

Any automation tools

This user works only with code o
ec2-cli-user

aws configure

AWS Access Key ID [None]: <Enter
AWS Secret Access Key [None]: <E
Default region name [None]: ap-s
Default output format [None]: js

aws sts get-caller-identity

aws ec2 describe-instances

aws ec2 describe-regions

aws s3 ls

rm ~/.aws/credentials
rm ~/.aws/config
```

```
login as: ec2-user
Authenticating with public key "key21111"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Dec  1 12:29:37 2025 from 103.170.54.108
[ec2-user@ip-172-31-15-60 ~]$ aws s3 ls

An error occurred (AccessDenied) when calling the ListBuckets operation: User: a
rn:aws:iam::581483106531:user/ec2-limited-user-cli is not authorized to perform:
 s3:ListAllMyBuckets because no identity-based policy allows the s3:ListAllMyBuc
kets action
[ec2-user@ip-172-31-15-60 ~]$
```

11.2.command-❌ aws s3 ls