

Published in CloudnLoud Tech Community



Manikanta Suru

Mar 24 · 5 min read · Listen

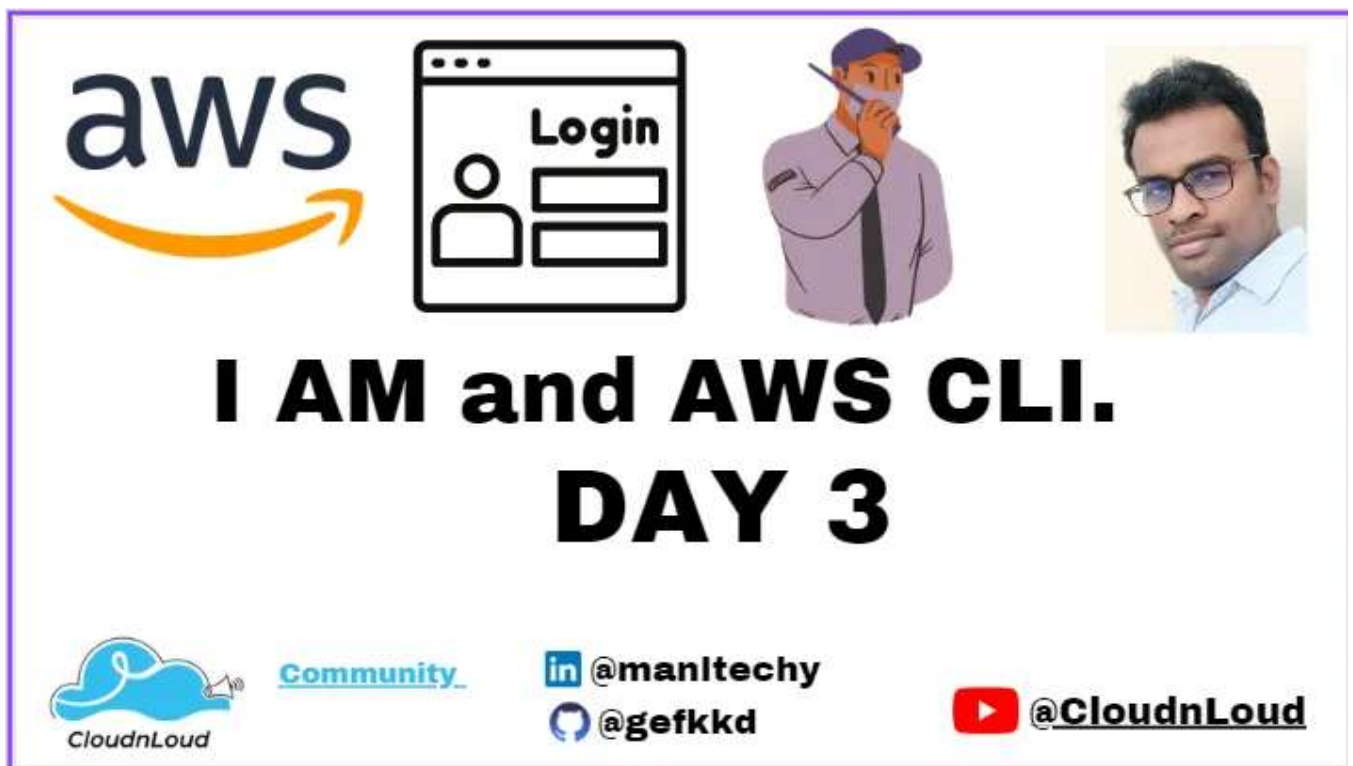


Save



📌 “From Zero to Cloud Hero: Day 1 of the 100-Day Journey to Mastering Cloud Computing”

I AM and AWS CLI.



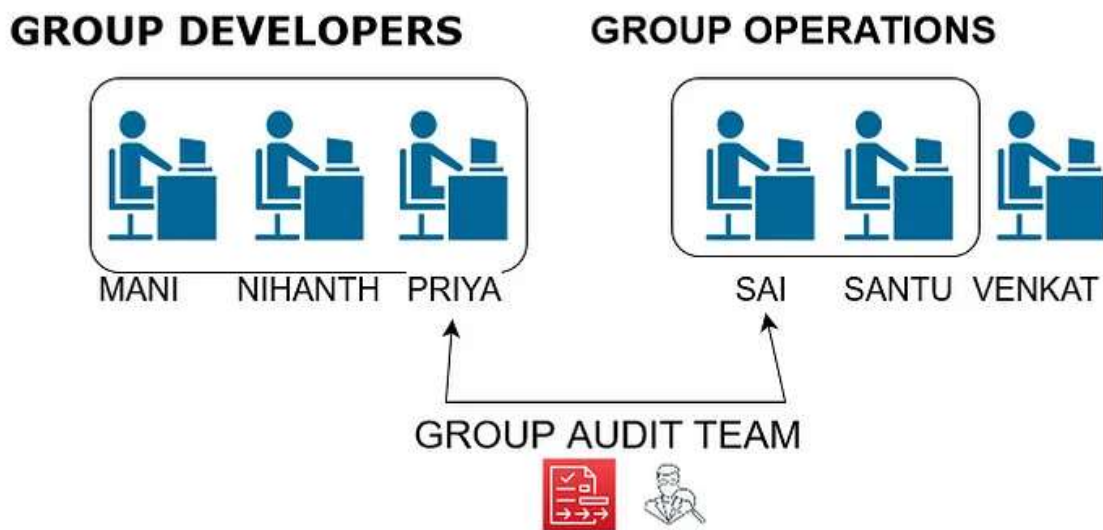
IAM Section :

Users & Groups

- IAM = Identity and Access Management. Global service



- Root account created by default, shouldn't be used or shared
- Users are people within your organization and can be grouped
- Groups only contain users, not other groups
 - Users don't have to belong to a group, and users can belong to multiple groups.



IAM: Permissions :

Users or Groups can be assigned JSON documents called policies

- These policies define the permissions of the users
- In AWS you apply the least privilege principle: don't give more permissions than a user needs

IAM Policies inheritance :

- Users or Groups can be assigned JSON documents called policies
- These policies define the permissions of the users
- In AWS you apply the least privilege principle: don't give more permissions than a user needs

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.

User groups (1/2)

☒ **admin_team**
1
AdministratorAccess
2022-10-11 (5 months ago)

☐ **man/team-users**
1
SupportUser, AmazonEC2FullAccess and 5 more
2022-08-25 (5 months ago)

► **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](#)

Cancel Previous **Next**

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
man

Console password type
Custom password

Show the password now?
No

Permissions summary

Name
admin_team

Type
Group

Used as
Permissions group

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.

Key
Department

Value - optional
Supporting

Remove

Add new tag

You can add up to 40 more tags.

Click 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

IAM > Users > Create 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

Email sign-in instructions

Console sign-in URL
https://man2022.signin.aws.amazon.com/console

User name
man

Console password
Show

Download .csv file

Return to users list

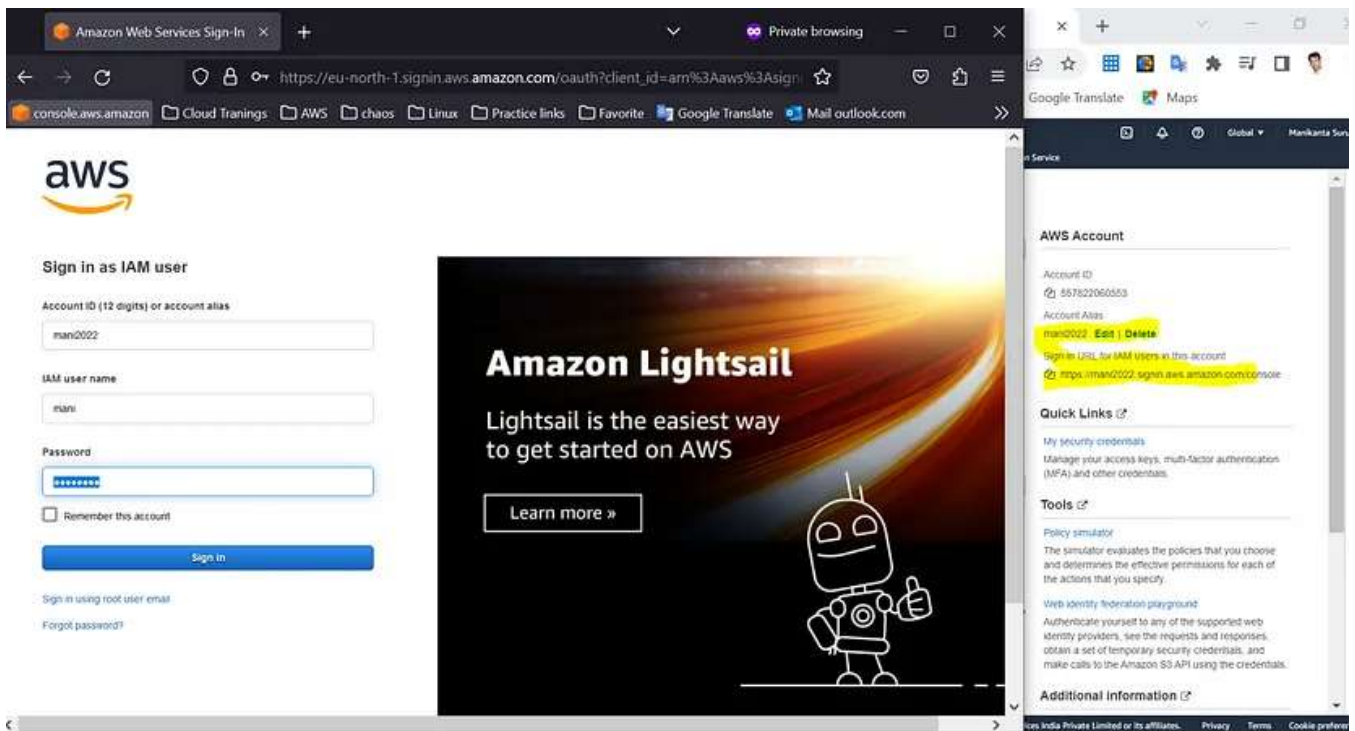
Click the user name will see permission policies added :

Users (1) [info](#)

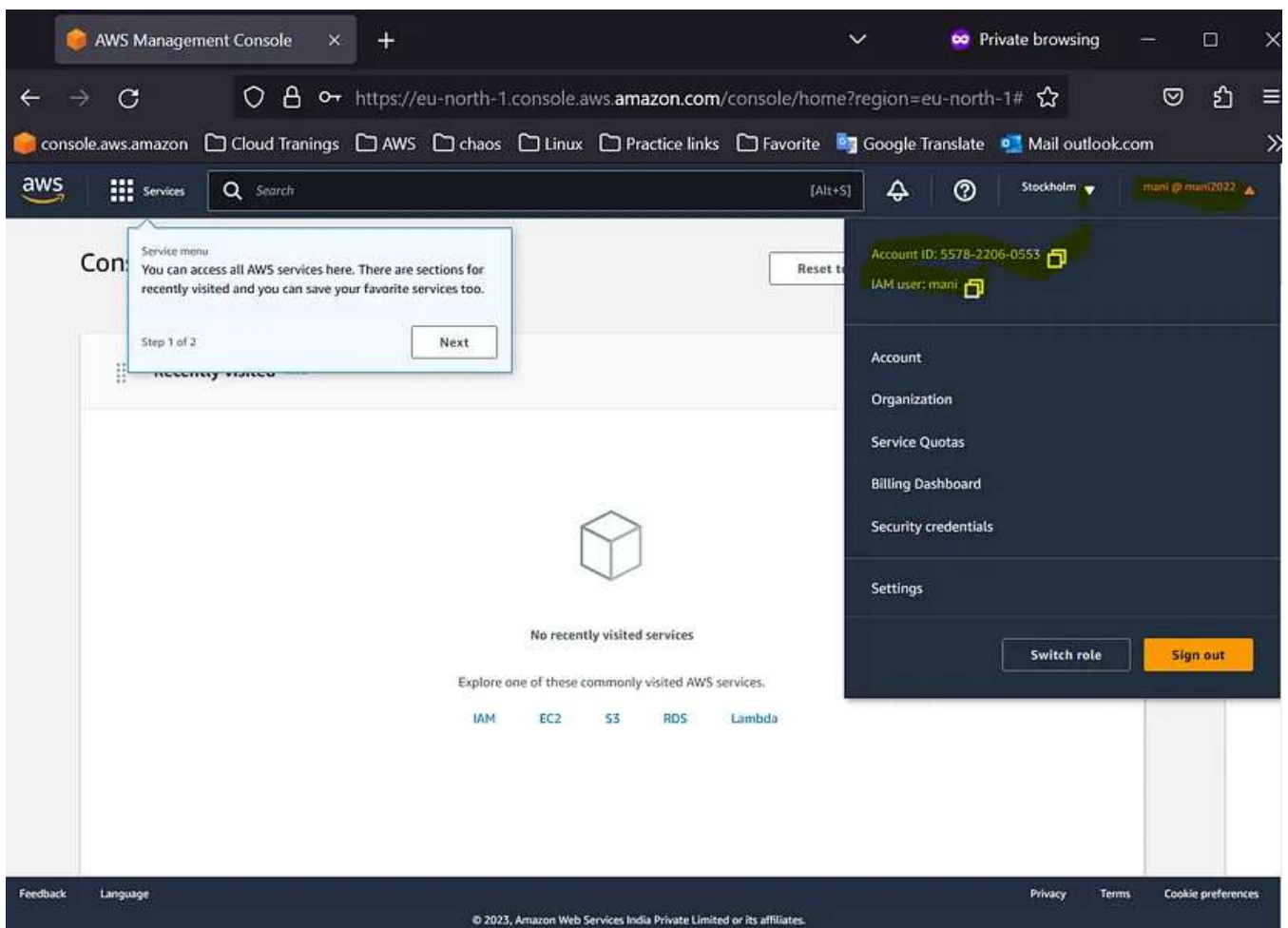
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

<input type="checkbox"/>	User name	Groups	Last activity	MFA	Password age	Active key age
<input type="checkbox"/>	maria	admin_team	Never	None	4 minutes ago	

Now we can log in with the newly created mani user

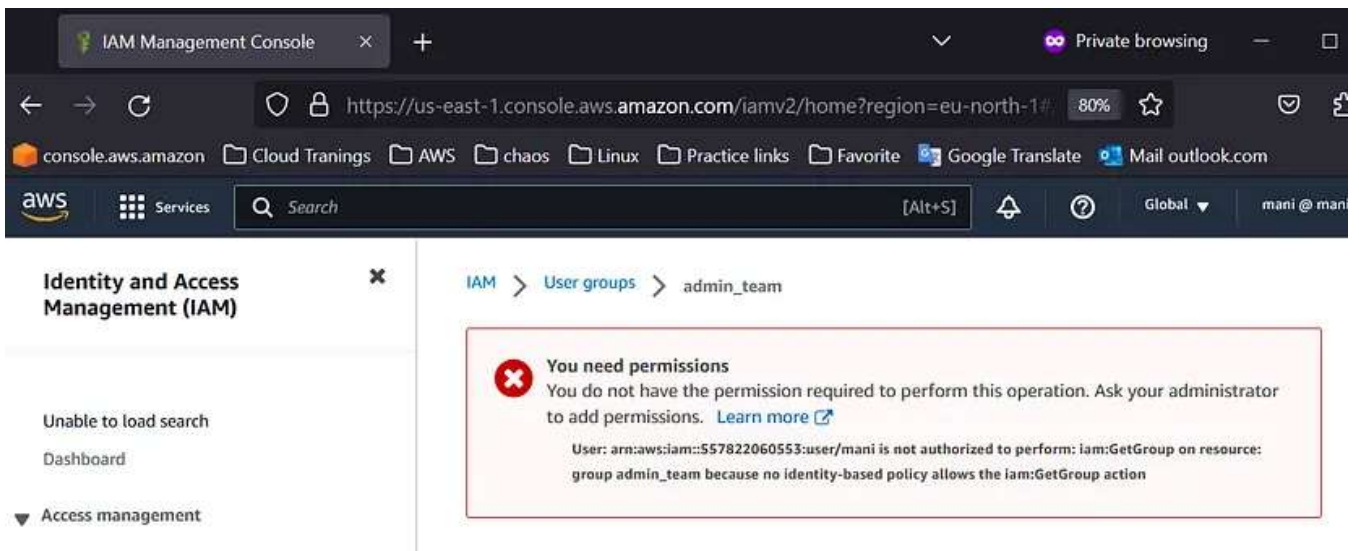


Login successfully with <https://mani2022.signin.aws.amazon.com/console>

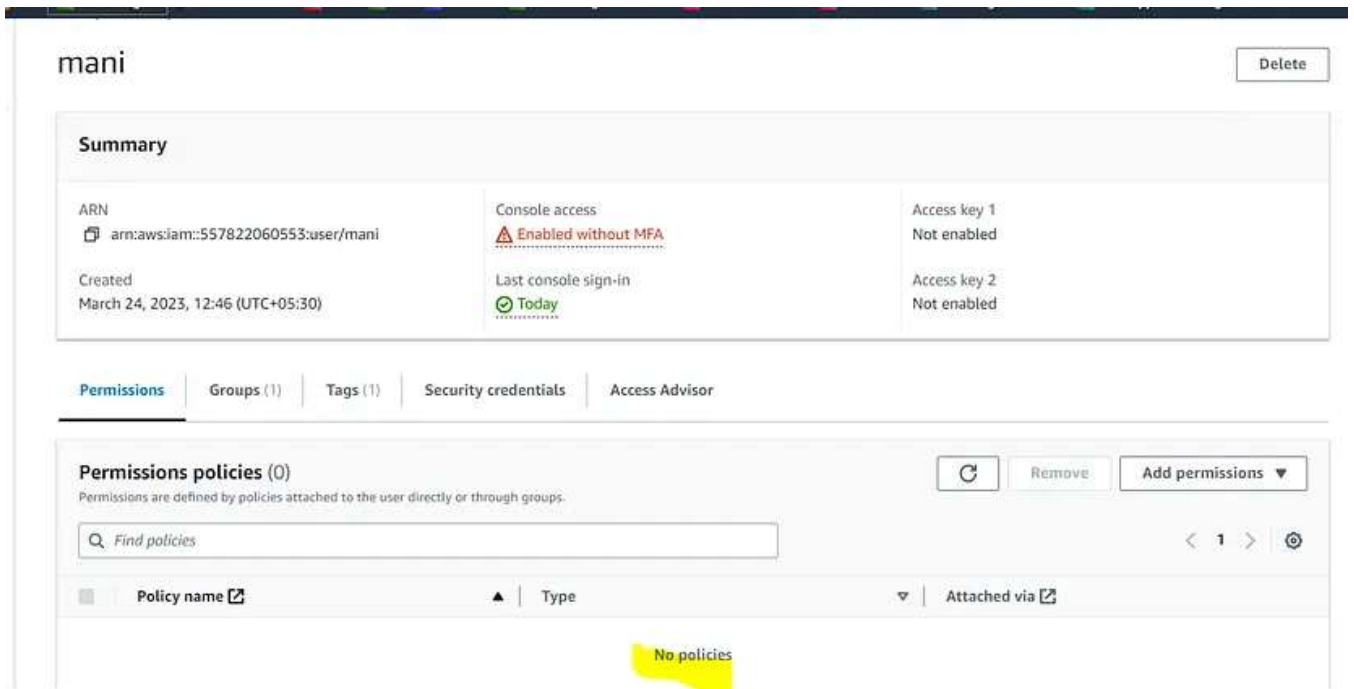


These policies define the permissions of the users :

Example :



users don't have permission and please find below checking with a root user account.



Now we need to attach the policy or create inline policy permissions

admin_team

Delete

Summary

Edit

User group name admin_team	Creation time October 11, 2022, 21:48 (UTC+05:30)	ARN arn:aws:iam::557822060553:group/admin_team
-------------------------------	------------------------------------------------------	-------------------------------------------------------------------

[Users](#) [Permissions](#) [Access Advisor](#)Permissions policies (0) [Info](#)

You can attach up to 10 managed policies.



Simulate

Remove

Add permissions

Attach policies

Create inline policy

<input type="checkbox"/>	Policy name ↗	Type	Description
--------------------------	-------------------------------	------	-------------

No resources to display

one policy was added to the user-added group with AdministratorAccess

1 policy added

mani

Delete

Summary

ARN arn:aws:iam::557822060553:user/mani	Console access Enabled without MFA	Access key 1 Not enabled
Created March 24, 2023, 12:46 (UTC+05:30)	Last console sign-in Today	Access key 2 Not enabled

[Permissions](#) [Groups \(1\)](#) [Tags \(1\)](#) [Security credentials](#) [Access Advisor](#)

Permissions policies (1)

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



Remove

Add permissions

< 1 >

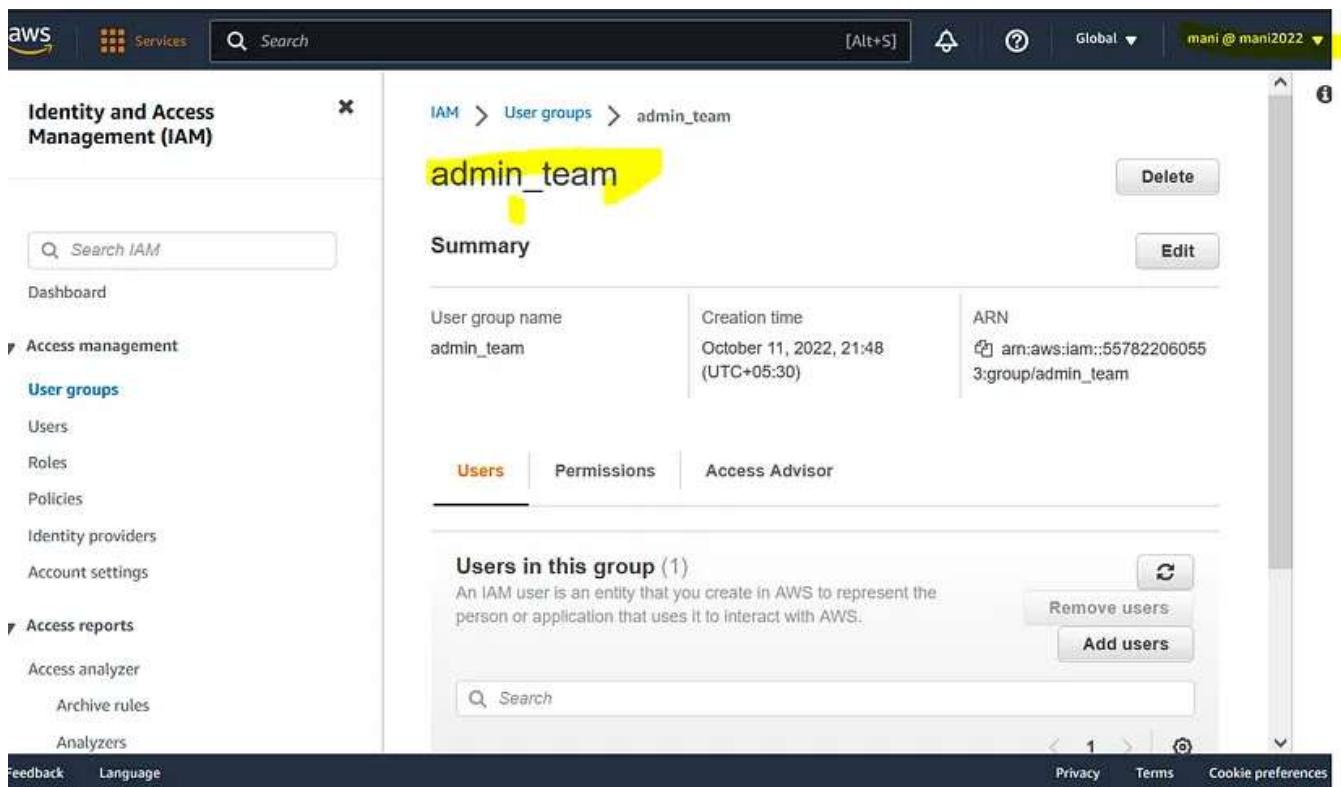
<input type="checkbox"/>	Policy name ↗	Type	Attached via ↗
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function	Directly

AdministratorAccess

Provides full access to AWS services and resources.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "*",  
7       "Resource": "*"  
8     }  
9   ]  
10 }
```


Login with mani I am a user Reference the page able to see the page without permission issue



IAM — Password Policy

- Strong passwords = higher security for your account
- In AWS, you can setup a password policy:
- Set a minimum password length
- Require specific character types:
- including uppercase letters
- lowercase letters
- numbers
- non-alphanumeric characters
- Allow all IAM users to change their own passwords
- Require users to change their password after some time (password expiration)

- Prevent password re-use.

goto [IAM](#) > [Account Settings](#) > [Edit password policy](#) >

[IAM](#) > [Account Settings](#) > [Edit password policy](#)

Edit password policy

☐ IAM default
Default password requirements for IAM users.

☒ Custom
Use a customized password policy.

Password minimum length.
 Enforce a minimum length of characters.
 characters
Needs to be between 6 and 128.

Password strength
☒ Require at least one uppercase letter from the Latin alphabet (A-Z)
☐ Require at least one lowercase letter from the Latin alphabet (a-z)
☐ Require at least one number
☒ Require at least one non-alphanumeric character (!@#\$%^&*()_+-=[]{}|')

Other requirements
☒ Turn on password expiration
 Expire password in day(s)
Needs to be between 1 and 1095 days.
☐ Password expiration requires administrator reset
☐ Allow users to change their own password
☒ Prevent password reuse
 Remember password(s)
Needs to be between 1 and 24.

Cancel
Save changes

Multi-Factor Authentication — MFA

- Users have access to your account and can possibly change configurations or delete resources in your AWS account
- You want to protect your Root Accounts and IAM users
- MFA = password you know + security device you own

- The main benefit of MFA:



Alice

Password



Successful login

if a password is stolen or hacked, the account is not compromised

goto I am **Dashboard** > **Add MFA**>select assign MFA >

The screenshot shows the AWS IAM console interface for assigning MFA. The breadcrumb trail at the top is 'IAM > Security credentials > Assign MFA device'. The left sidebar shows 'Step 1: Select MFA device' and 'Step 2: Set up device'. The main content area is titled 'Set up device' and contains the following steps:

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. Fill in two consecutive codes from your MFA device.

Below step 3, there are two input fields labeled 'MFA code 1' and 'MFA code 2'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Add MFA'.

How can users access AWS?

- To access AWS, you have three options:
- AWS Management Console (protected by password + MFA)
- AWS Command Line Interface (CLI): protected by access keys
- AWS Software Developer Kit (SDK) — for code: protected by access keys
- Access Keys are generated through the AWS Console

- Users manage their own access keys
- Access Keys are secret, just like a password. Don't share them
- Access Key ID ~= username
- Secret Access Key ~= password

What's the AWS CLI?

- A tool that enables you to interact with AWS services using commands in your command-line shell
- Direct access to the public APIs of AWS services
- You can develop scripts to manage your resources
- It's open-source <https://github.com/aws/aws-cli>
 - Alternative to using AWS Management Console

Installing or updating the latest version of the AWS CLI

This topic describes how to install or update the latest release of the AWS Command Line Interface (AWS CLI) on...

docs.aws.amazon.com

AWS CLI Hands-on and please follow the above link to which type of os use like windows or linux.

i am using Linux vm

```

Inflating: aws/dist/docutils/parsers/rst/include/isoGrk4.txt
ubuntu $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu $ aws --version
aws-cli/2.11.5 Python/3.11.2 Linux/5.4.0-131-generic exe/x86_64.ubuntu.20 prompt/off
ubuntu $ █

```

IAM > Users > ani > Access keys > Create access key

IAM > Users > mani > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Access key best practices & alternatives

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

- ☒ **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 EKS, 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 enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
- ☐ **Other**
Your use case is not listed here.

Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

☒ I understand the above recommendation and want to proceed to create an access key.

Cancel **Next**

Access key created and download

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > mani > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAVDYGFJAEV5FJM2OI	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

[Download .csv file](#) **Done**

So now my AWS CLI is configured so we can have a look at how it works.

We can do `aws iam list-users` and press Enter and this will list all the users in my accounts. And as we can see.

```

ubuntu $ aws configure
AWS Access Key ID [*****M2OI]: AKIAYDYGJFAEV5FJM2OI
AWS Secret Access Key [*****2/Jn]: 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
Default region name [us-east-1]:
Default output format [None]:
ubuntu $ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "mani",
      "UserId": "AIDAYDYGJFAEXUOCHWCEQ",
      "Arn": "arn:aws:iam::557822060553:user/mani",
      "CreateDate": "2023-03-24T07:16:41+00:00",
      "PasswordLastUsed": "2023-03-24T07:32:26+00:00"
    },
    {
      "Path": "/",
      "UserName": "mani-V1",
      "UserId": "AIDAYDYGJFAESAWUPN57R",
      "Arn": "arn:aws:iam::557822060553:user/mani-V1",
      "CreateDate": "2023-03-24T09:27:25+00:00"
    }
  ]
}
ubuntu $ █

```

Sources: [AWS](#).

That's it, thank you for reading.

https://github.com/gefkkd/AWS_100-Days_Challenge.git

👉 In case you would like to continue the discussion, you can always reach out to me on [Twitter](#) or on LinkedIn for professional networking, if you feel like following me on [GitHub](#) you can also do that.

👉 Follow [Cloudncloud Tech Community](#) for more insightful knowledge & resources & [CloudnLoud YouTube channel](#).

[AWS](#)

[Cloud Computing](#)

[Learning](#)

[Security](#)

[Careers](#)