IAM IDENTITY ACCESS MANAGEMENT SERVICE

SECURITY ENGINEERING – USER LEVEL OF SECURITY – GLOBAL SPECIFIC SERVICE

user creation

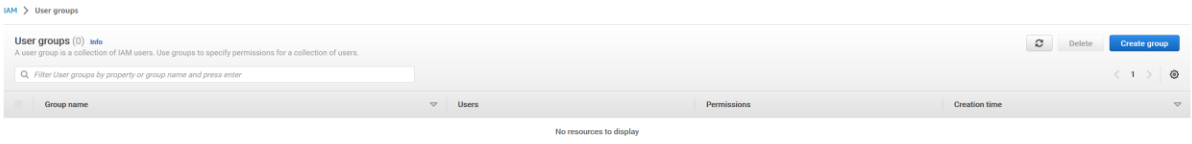
Group creation

Policy

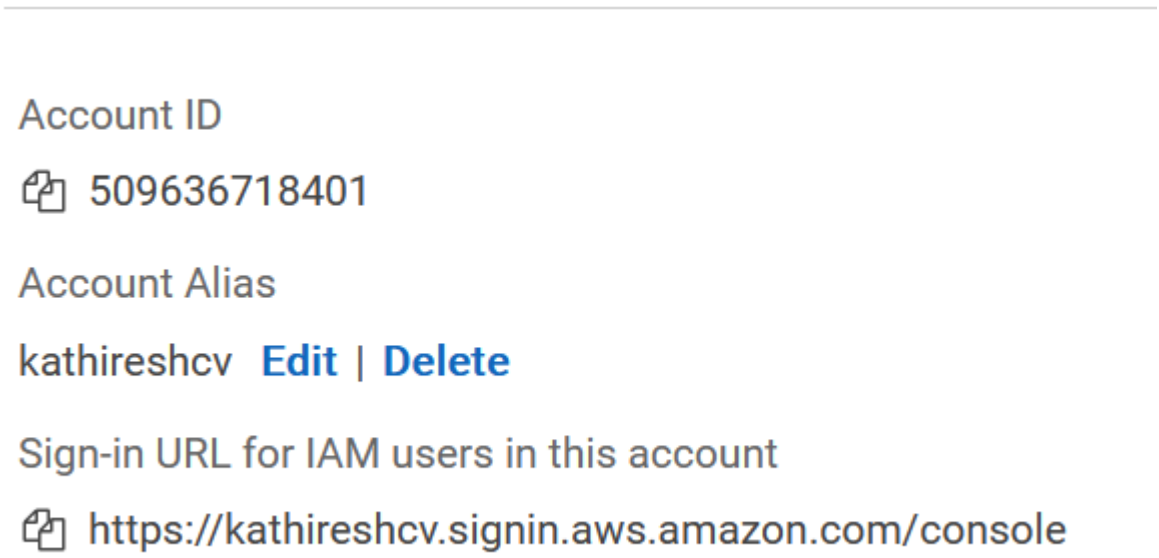
Roles

(*) Search IAM

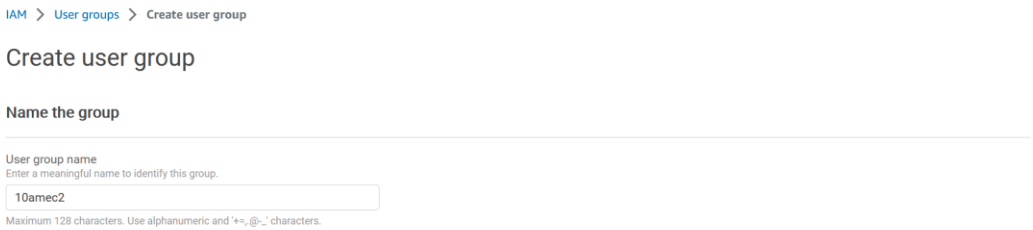
(*) create user group



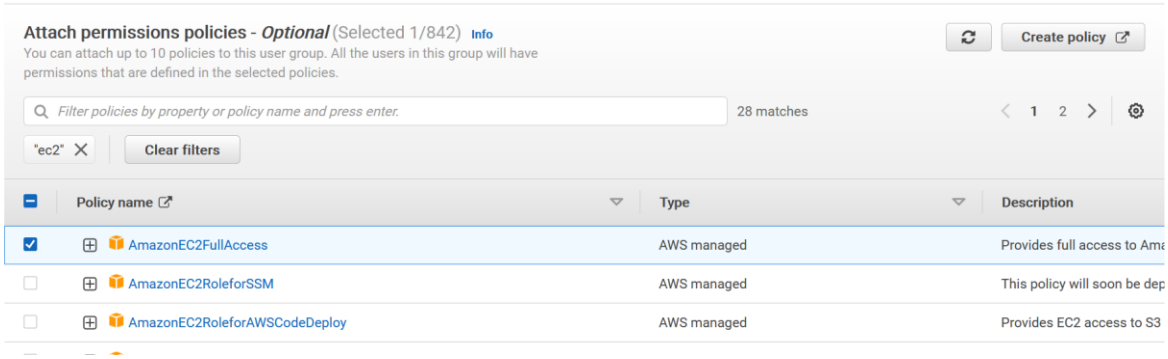
(*) add account alias



(*) create user group



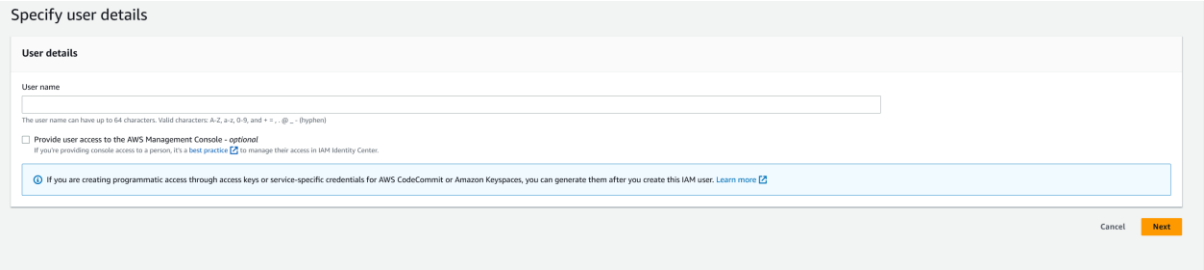
(*) Give permissions as per the client needs client needs full ec2 permission in the search box type ec2 and select amazon ec2 full access permissions.



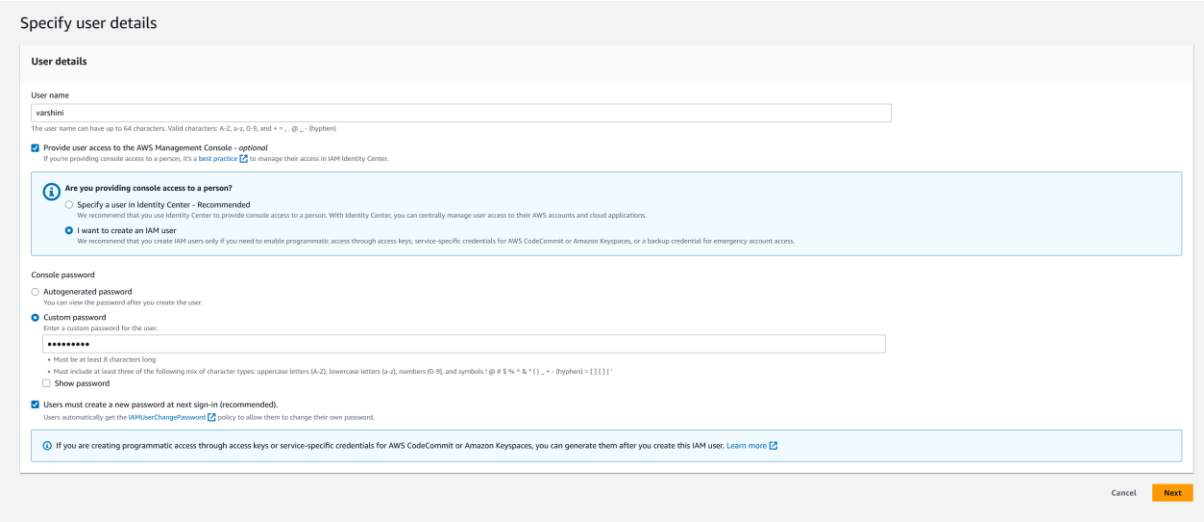
(*) press create group. group created

(*) create users

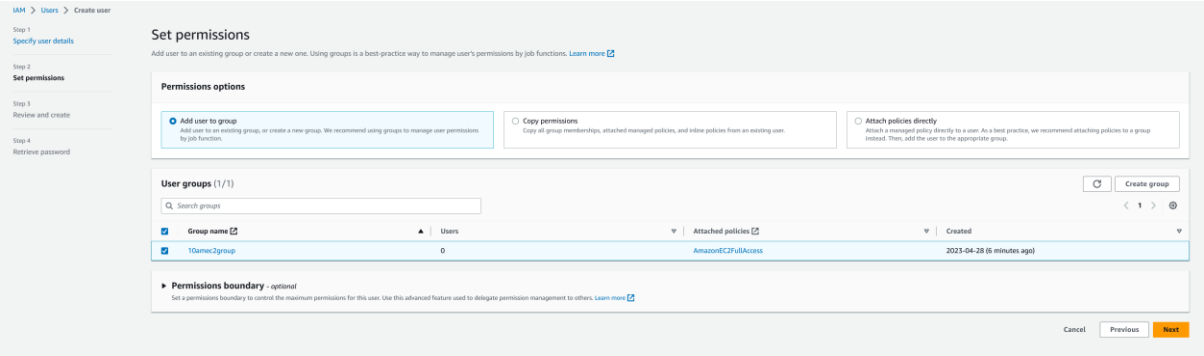
(*) click on add users and enter the name of the user



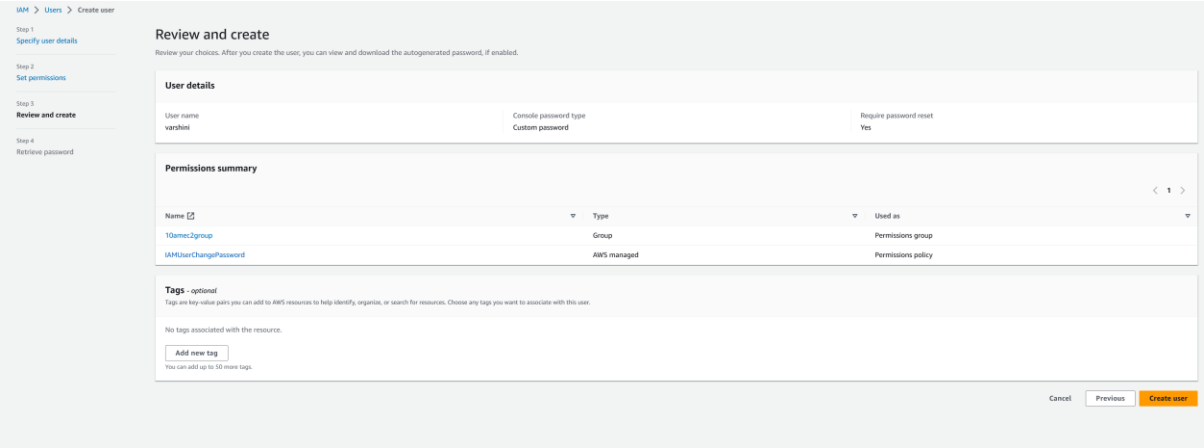
(*) create user name provide user access to AWS management console and password for the user



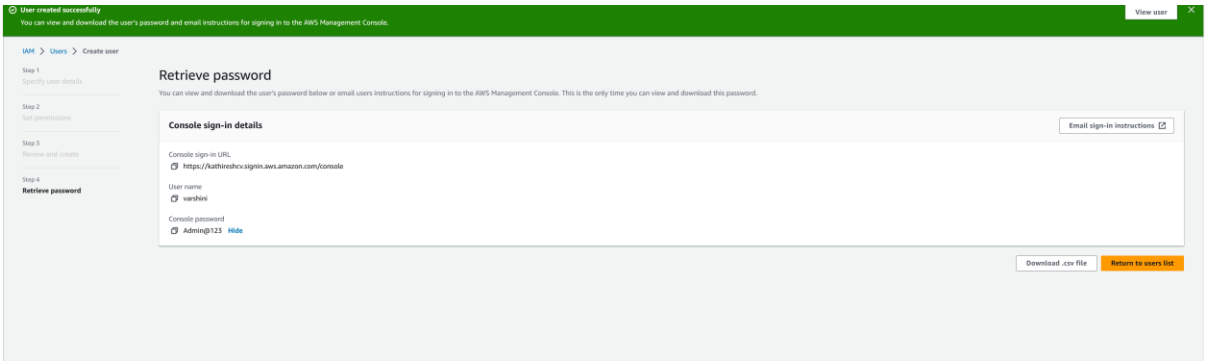
(*) set permissions add user to group and press next



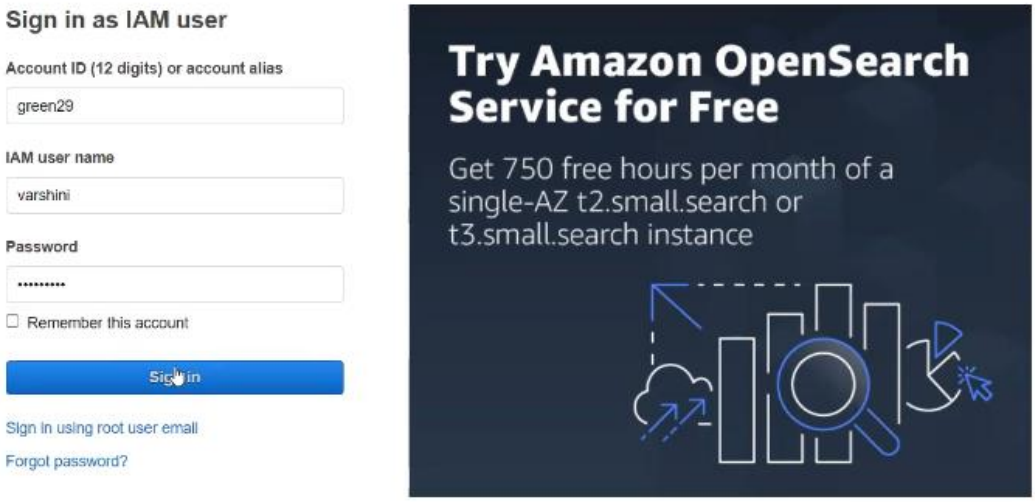
(*) review and create – create user



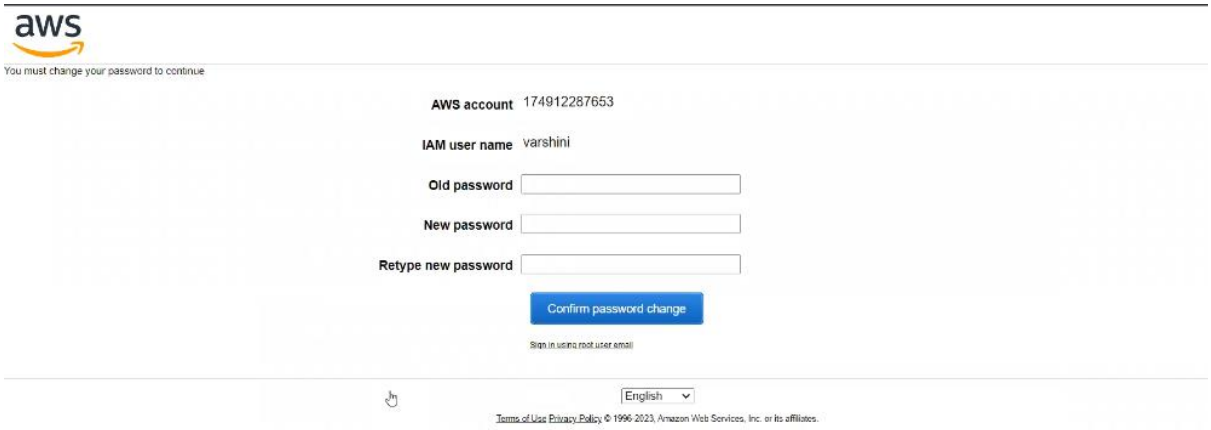
(*) user created successfully



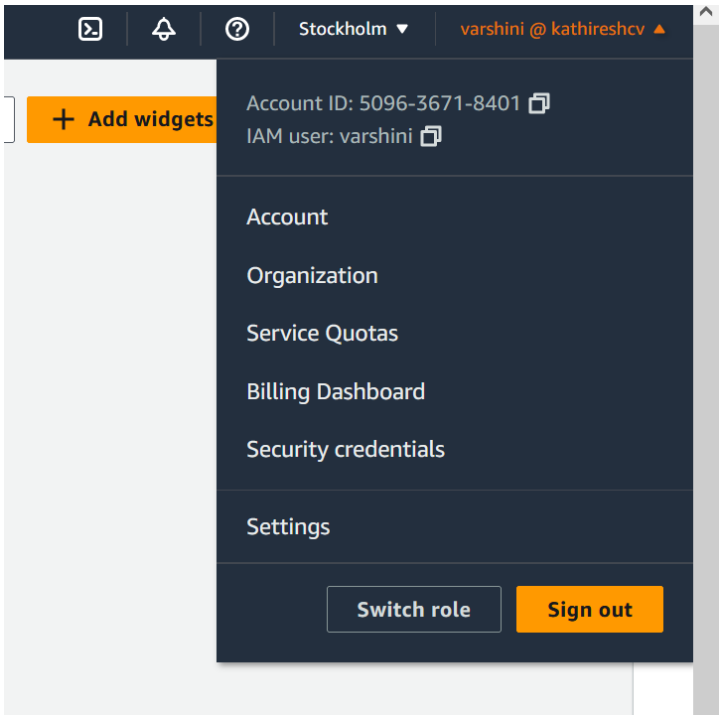
(*) open incognito tab paste the url in console sign in details login to the account



(*) change the password to continue



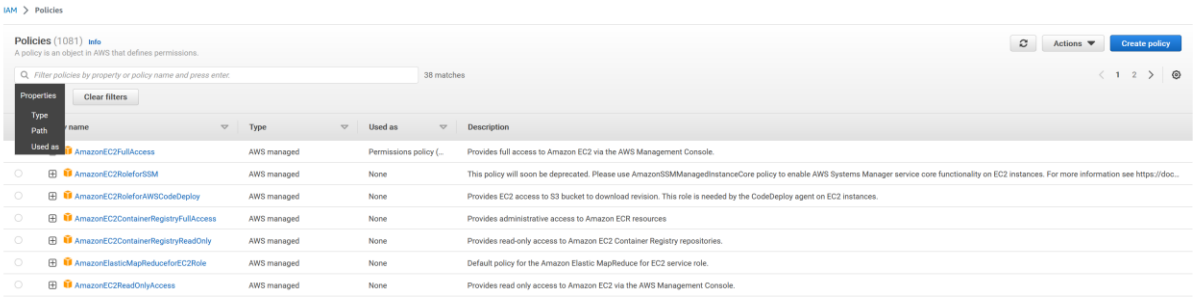
(*) IAM USER



(*) now the user is created with ec2 full access permissions user can access only EC2 and the user cannot access S3 or other components.

<u>POLICIES</u>

800+ POLICIES IN AWS



(*) create policy

(*) choose a service

(*) select customizable permissions

(*) client requested S3 list buckets

Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

Visual editor | JSON

Import managed policy

Expand all | Collapse all

S3 (1 action) Clone | Remove

Service S3

Actions close

Specify the actions allowed in S3

Switch to deny permissions

Filter actions

Manual actions (add actions)

All S3 actions (s3:*)

Access level

Expand all | Collapse all

List (1 selected)

ListAccessPoints
ListBucketMultipartUploads
ListMultiRegionAccessPoints
ListAccessPointsForObjectLambda
ListBucketVersions
ListStorageLensConfigurations
ListAllMyBuckets
ListJobs
ListBucket
ListMultipartUploadParts

Read

Tagging

Write

Permissions management

Resources The actions you chose support all resources.

Request conditions Specify request conditions (optional)

Add additional permissions

Character count: 125 of 6,144.

Cancel Next: Tags

(*) review policy add name for the policy – create policy

Create policy

1 2 3

Review policy

Name* varshinis3policy

Use alphanumeric and '+=,@-_' characters. Maximum 128 characters.

Description varshinis3policy

Maximum 1000 characters. Use alphanumeric and '+=,@-_' characters.

Summary

Filter

Service	Access level	Resource	Request condition
Allow (1 of 374 services) Show remaining 373			
S3	Limited: List	All resources	None

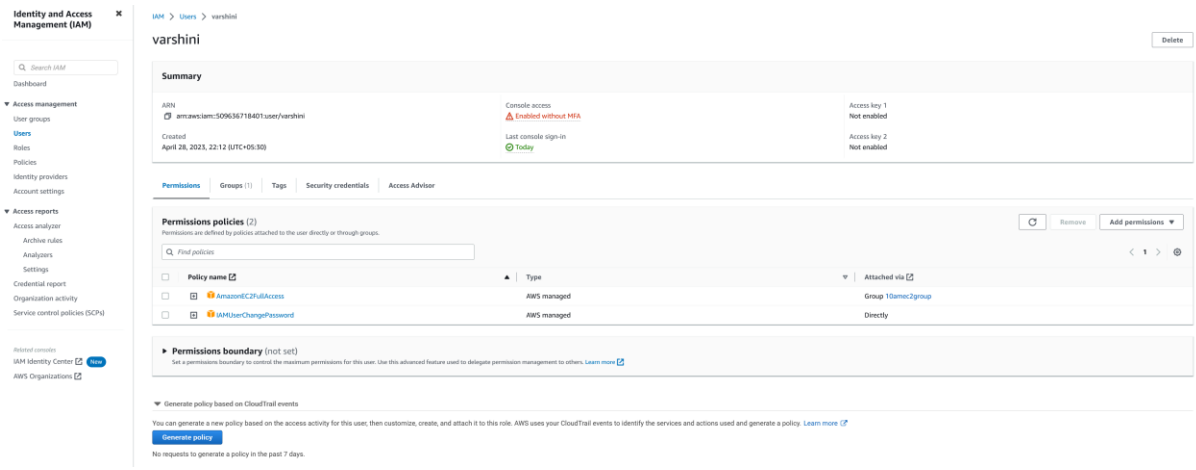
Tags

Key	Value
No tags associated with the resource.	

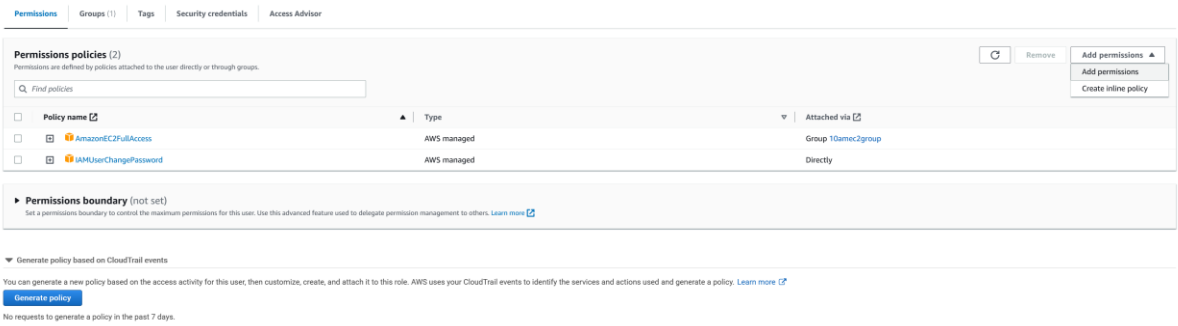
* Required

Cancel Previous Create policy

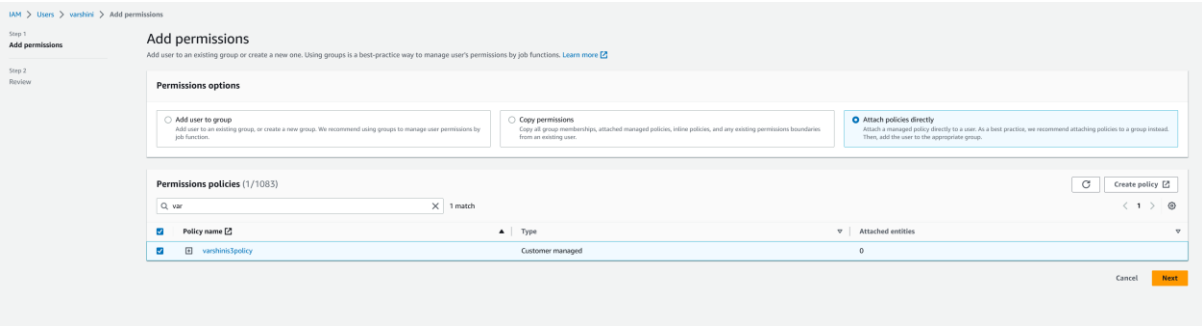
(*) applying permission to the user – go to user -- select user – add permissions



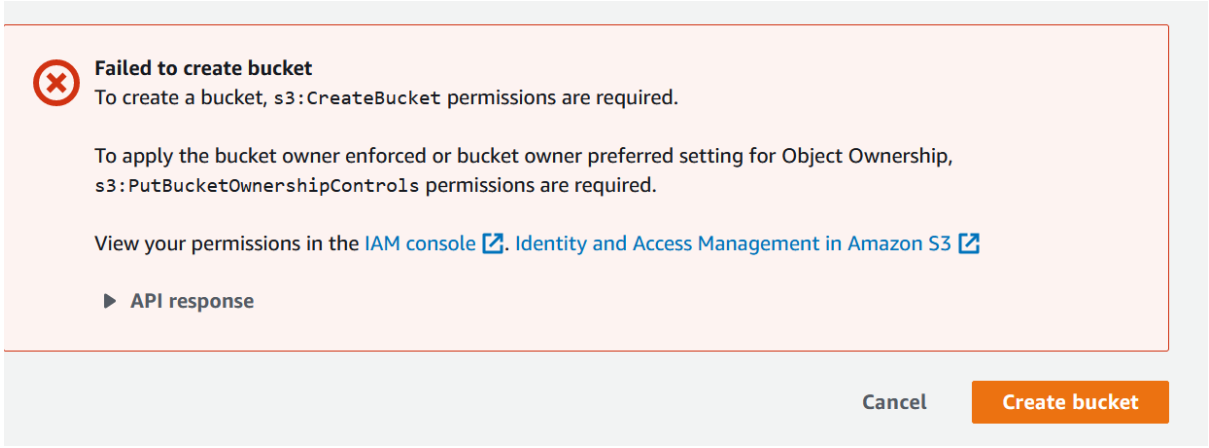
(*) add permission



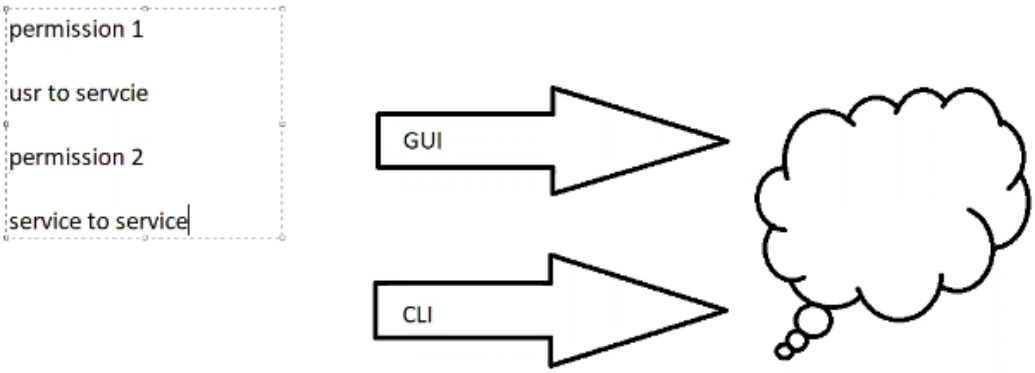
(*) attach policies directly

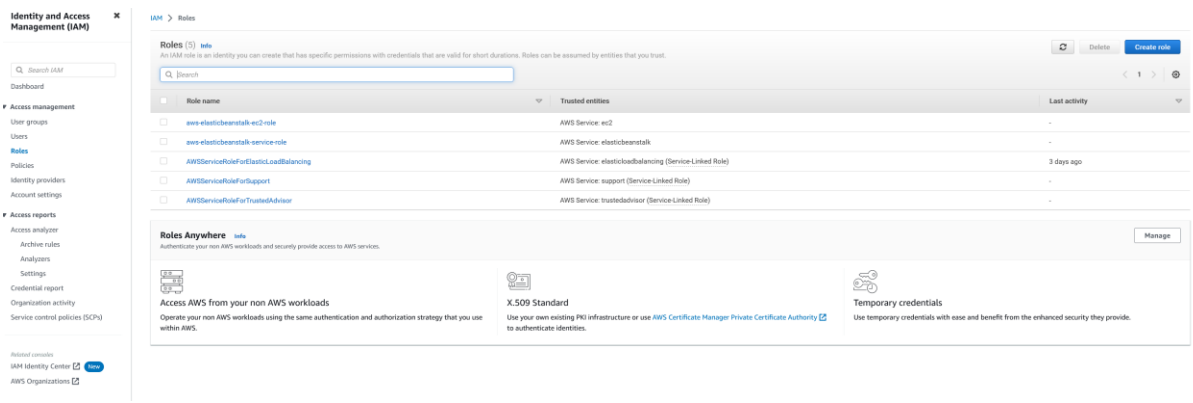


(*) insuffeciant permission error

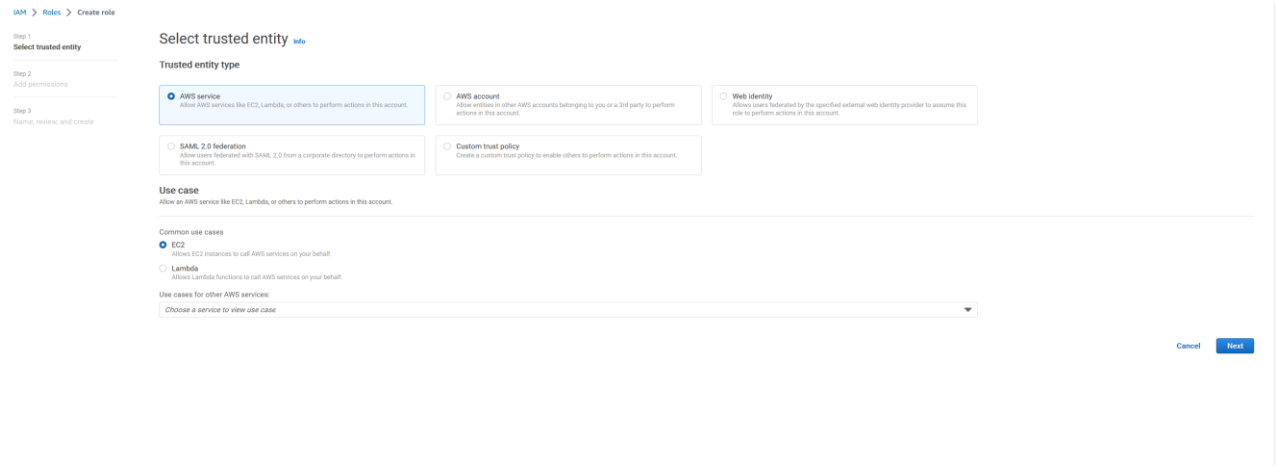


ROLES

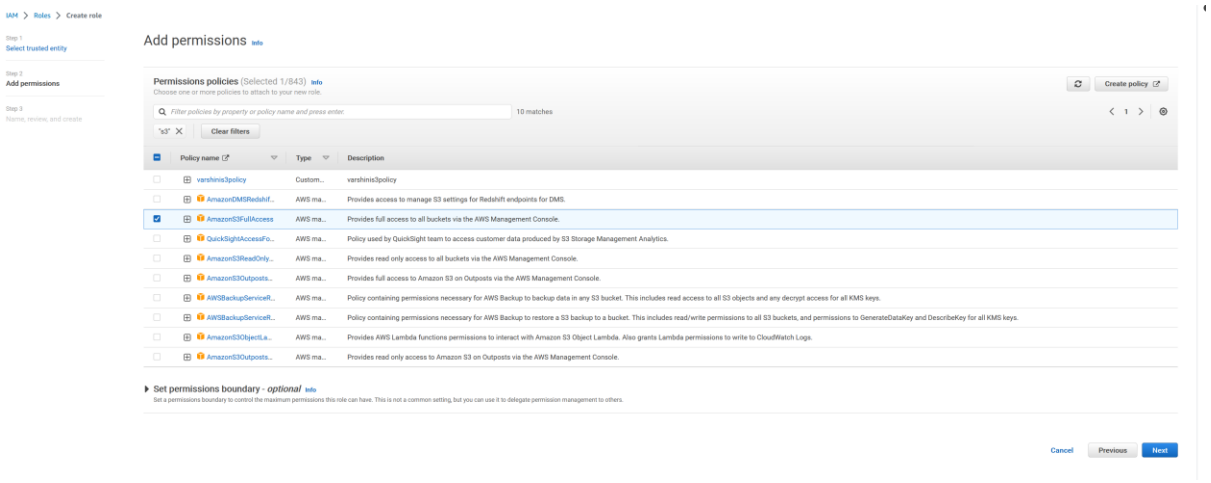




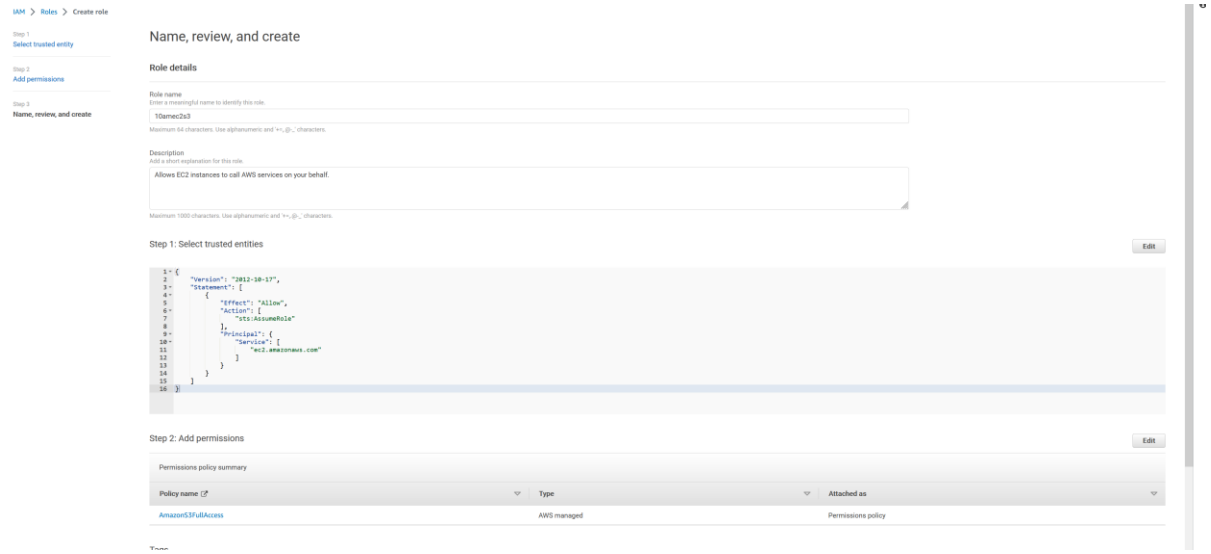
(*) Create a role – select trusted entity – aws service – use case – EC2 --Next



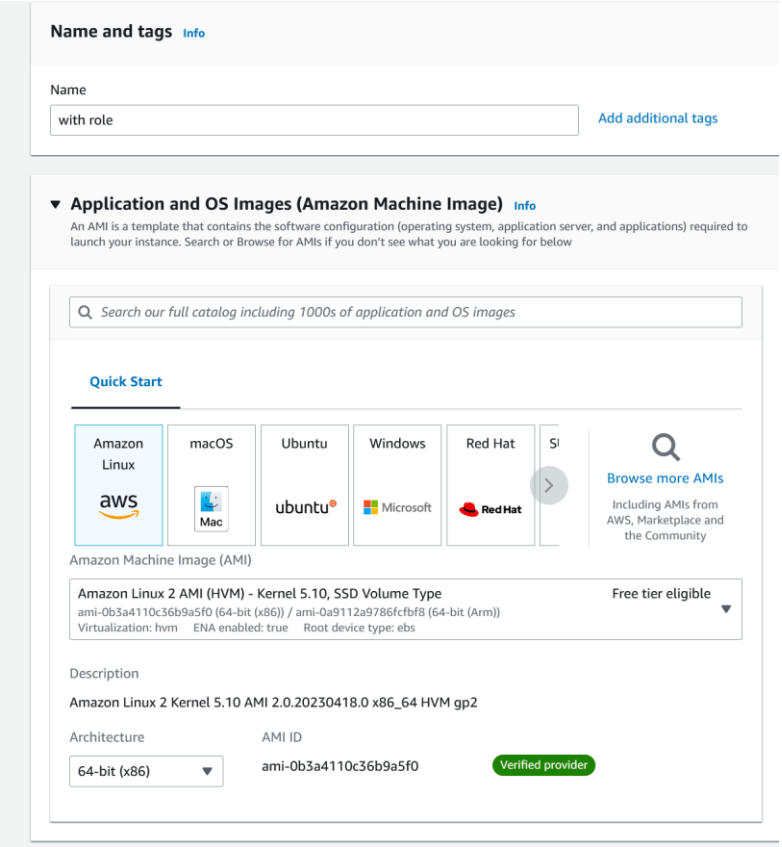
(*) add permissions S3 - next



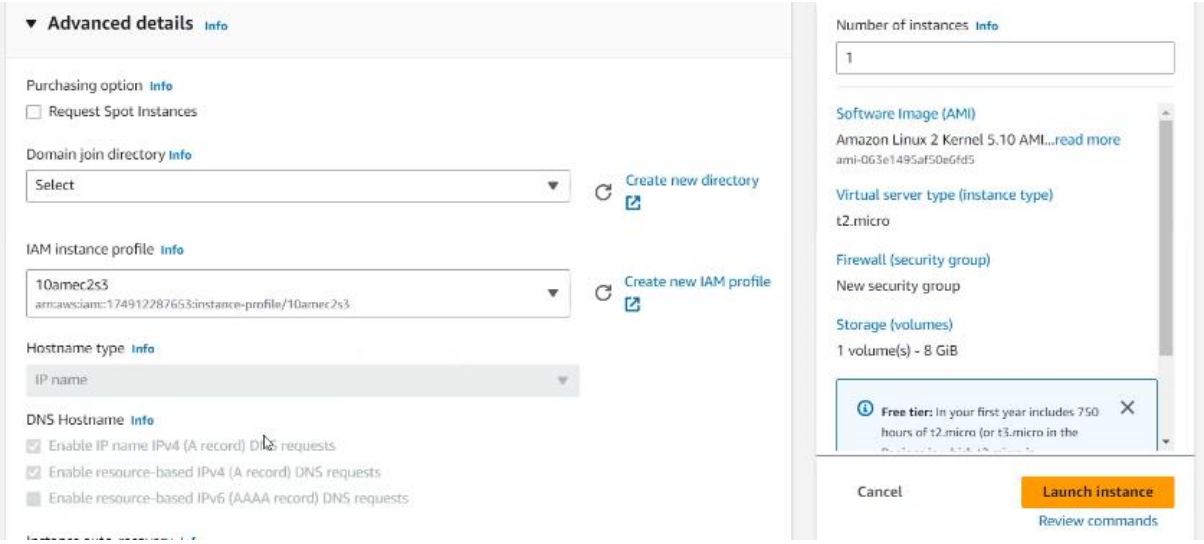
(*) review and create – name – create role



(*) Launch an instance – amazon linux – ppk keypair – advanced settings -- IAM instance profile -- 10amec2s3 – instance created



(*) advance details 10amec2s3



(*) create instance with role and without role

(*) login using PUTTY configurator paste ipv4 address -- ssh- credentials

(*) with role white

(*) without role black

```
ec2-user@ip-172-31-25-50:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-25-50 ~]$ sudo yum update -y
```

(*) Aws s3 ls

```
root@ip-172-31-25-50:~
[ec2-user@ip-172-31-25-50 ~]$ sudo su -
[root@ip-172-31-25-50 ~]# aws s3 ls
2023-04-17 05:12:41 aws1704
2023-04-18 14:02:43 aws1904
2023-03-01 05:10:32 migrationbucket12
[root@ip-172-31-25-50 ~]#
```

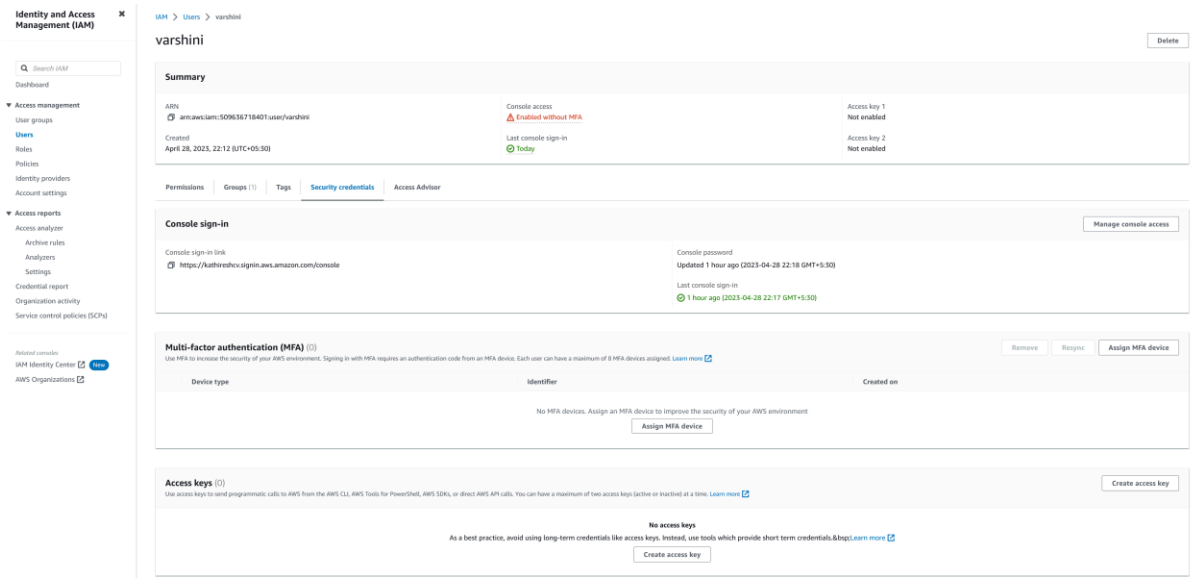
(*) without role

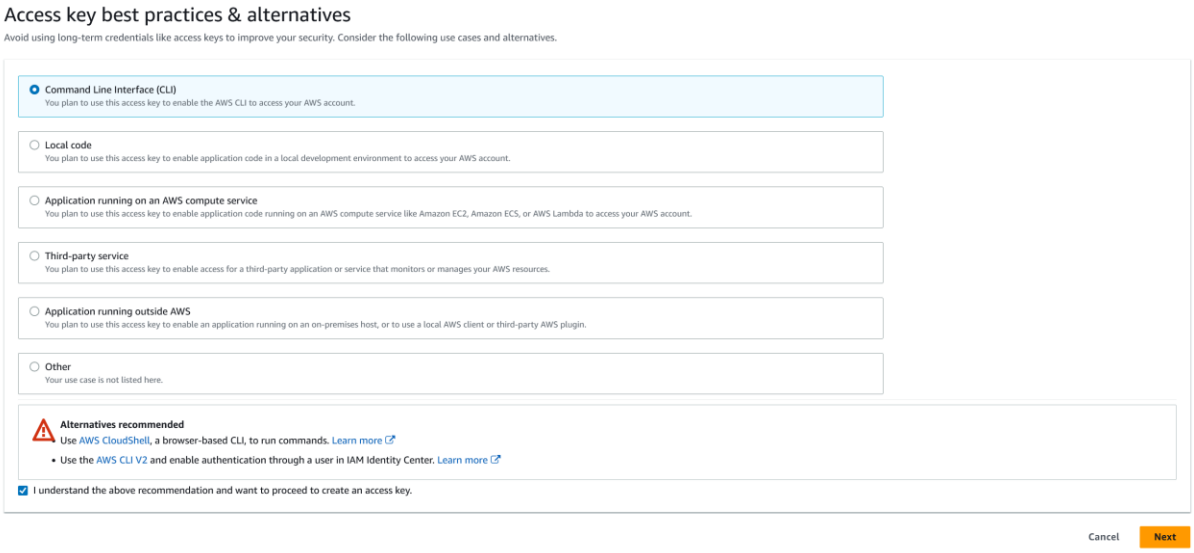
```
ec2-user@ip-172-31-27-102:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-27-102 ~]$ sudo yum update -y
```

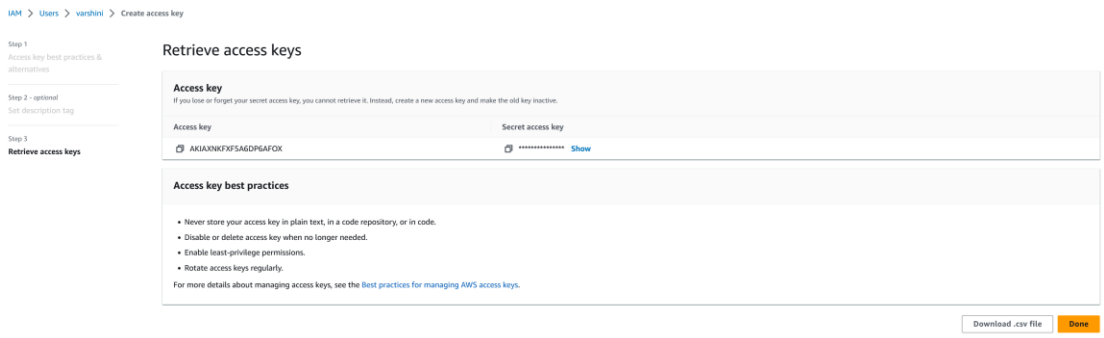
(*) aws s3 ls

```
root@ip-172-31-27-102:~
[ec2-user@ip-172-31-27-102 ~]$ sudo su -
[root@ip-172-31-27-102 ~]# aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
[root@ip-172-31-27-102 ~]#
```

(*) for user login in CLI go to IAM user – select user – security credentials – access keys --


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
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center New
AWS Organizations
IAM > Users > varshini
varshini
Delete
Summary
ARN
arn:aws:iam::509636718401:user/varshini
Console access
Enabled without MFA
Access key 1
Not enabled
Created
April 28, 2023, 22:12 (UTC+05:30)
Last console sign-in
Today
Access key 2
Not enabled
Permissions
Groups (1)
Tags
Security credentials
Access Advisor
Console sign-in
Manage console access
Console sign-in link
https://kathireshcv.signin.aws.amazon.com/console
Console password
Updated 1 hour ago (2023-04-28 22:18 GMT+5:30)
Last console sign-in
1 hour ago (2023-04-28 22:17 GMT+5:30)
Multi-factor authentication (MFA) (0)
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more
Remove
Resync
Assign MFA device
Device type
Identifier
Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment
Assign MFA device
Access keys (0)
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
Create access key
No access keys
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials.&bsp;Learn more
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.
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 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


IAM > Users > varshini > 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
AKIAXNKFXF5A6DP6AFOX
********** 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

root@ip-172-31-27-102:~

```
[ec2-user@ip-172-31-27-102 ~]$ sudo su -
[root@ip-172-31-27-102 ~]# aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
[root@ip-172-31-27-102 ~]# aws configure
AWS Access Key ID [None]: AKIASROMVWOS7BLIPDON
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]: json
[root@ip-172-31-27-102 ~]# aws s3 ls
2023-04-17 05:12:43 aws1704
2023-04-18 14:02:45 aws1904
2023-04-08 08:28:55 migrationbucket12
[root@ip-172-31-27-102 ~]#
```