

LAB - Create IAM Access Keys and Configure AWS CLI:

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), Access Management, and Access reports. The main content area displays the details for a user named 'user1_viaya'. The 'Summary' section shows the user's ARN, creation date, and console access status. The 'Permissions' tab is selected, showing a list of permissions policies attached to the user, including 'IAMUserChangePassword'.

user1_viaya info [Delete](#)

Summary

ARN
arn:aws:iam::686699774218:user/user1_viaya

Created
January 19, 2026, 13:56 (UTC+01:00)

Console access
Enabled without MFA

Last console sign-in
Never

Access key 1
[Create access key](#)

Permissions Groups Tags (1) Security credentials Last Accessed

Permissions policies (1) [Remove](#) [Add permissions](#)

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

Filter by Type
All types

Policy name	Type	Attached via
IAMUserChangePassword	AWS managed	Directly

The screenshot shows the 'Retrieve access keys' page in the AWS IAM console. A green banner at the top states: '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.' The page includes a section for 'Access key' with fields for 'Access key' and 'Secret access key'. The 'Secret access key' field is masked with asterisks and has a 'Show' button. Below this, there is a section for 'Access key best practices' with a list of recommendations. At the bottom, there are buttons for 'Download .csv file' and 'Done'.

Retrieve access keys info

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

[AKIAZ7YT7AEFCP7KMFPI](#) | [***** 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](#)

Install AWS CLI


```
vijayagagla@DESKTOP-FP40P26:~$ sudo apt install awscli
[sudo] password for vijayagagla:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package awscli is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'awscli' has no installation candidate
```

```
vijayagagla@DESKTOP-FP40P26:~$ sudo snap install aws-cli --classic
2026-01-19T14:11:02+01:00 INFO Waiting for automatic snapd restart...
aws-cli (v2/stable) 2.33.2 from Amazon Web Services (aws/) installed
vijayagagla@DESKTOP-FP40P26:~$ aws --version
aws-cli/2.33.2 Python/3.13.11 Linux/5.15.167.4-microsoft-standard-WSL2 exe/x86_64.ubuntu.24
vijayagagla@DESKTOP-FP40P26:~$ |
```

Configure AWS CLI

```
vijayagagla@DESKTOP-FP40P26:~$ aws configure
AWS Access Key ID [None]: AKIAZ7YT7AEFCP7KMFPI
AWS Secret Access Key [None]: 5ypWXPzhRGGUCsI65NUwC4V0Ztcr0Z1MLdQQsln
Default region name [None]: us-east-1
Default output format [None]: json
```

Verify AWS CLI Configuration

```
vijayagagla@DESKTOP-FP40P26:~$ aws s3 ls
2026-01-06 09:00:29 2048-ci-cd-artifacts-nidhal-2026
2025-11-20 11:52:17 ansible-ssm-1763635934-hydra
2025-11-19 23:26:20 ansible-ssm-temp-1763591178
2025-11-21 17:15:45 aws-photo-bucket-yacine
2025-11-22 22:34:35 neeha-meta-args-data-db13867f
2025-11-22 22:34:35 neeha-meta-args-protected-dede4ee2
2025-11-18 19:10:01 project1-max-fotios
vijayagagla@DESKTOP-FP40P26:~$ |
```