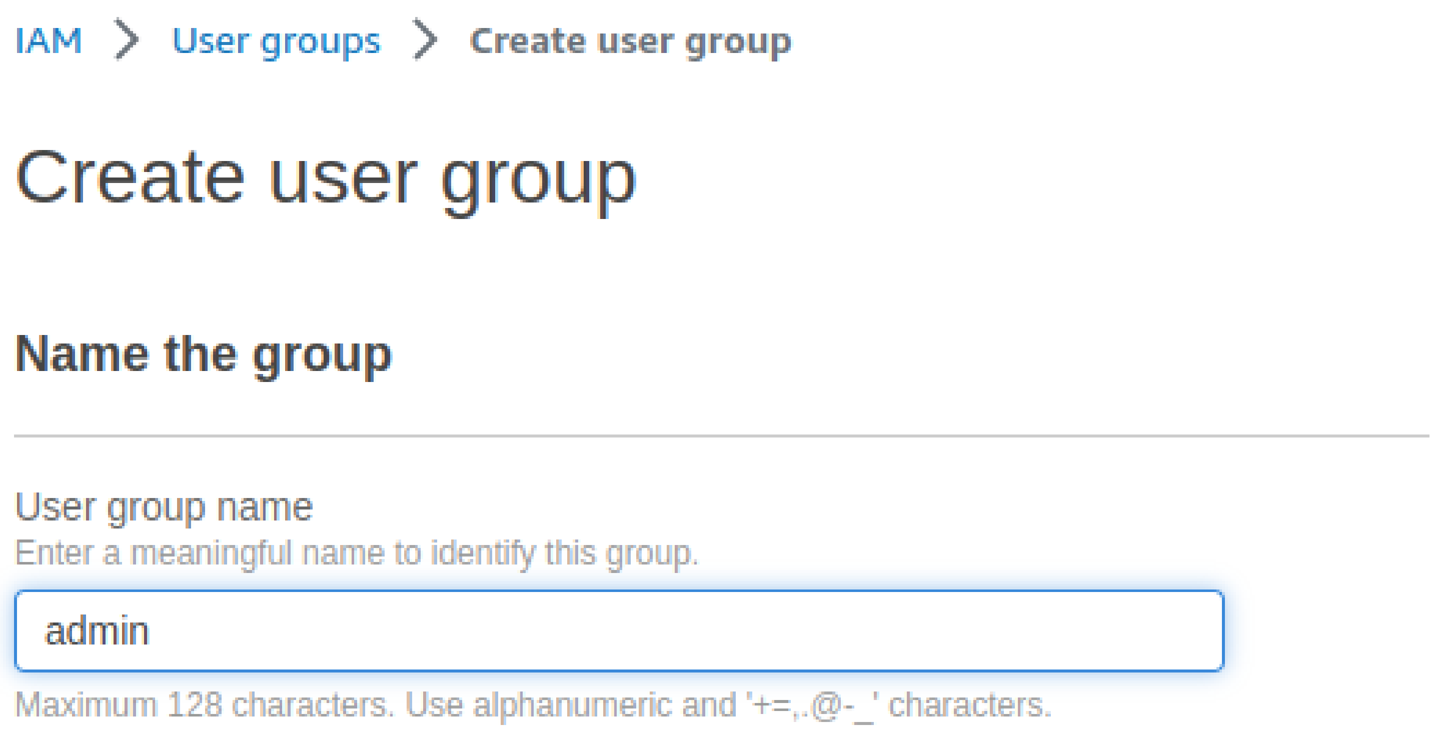
SetUpUser and AwsCli

Create User Groups:


IAM > User groups > Create user group

Create user group

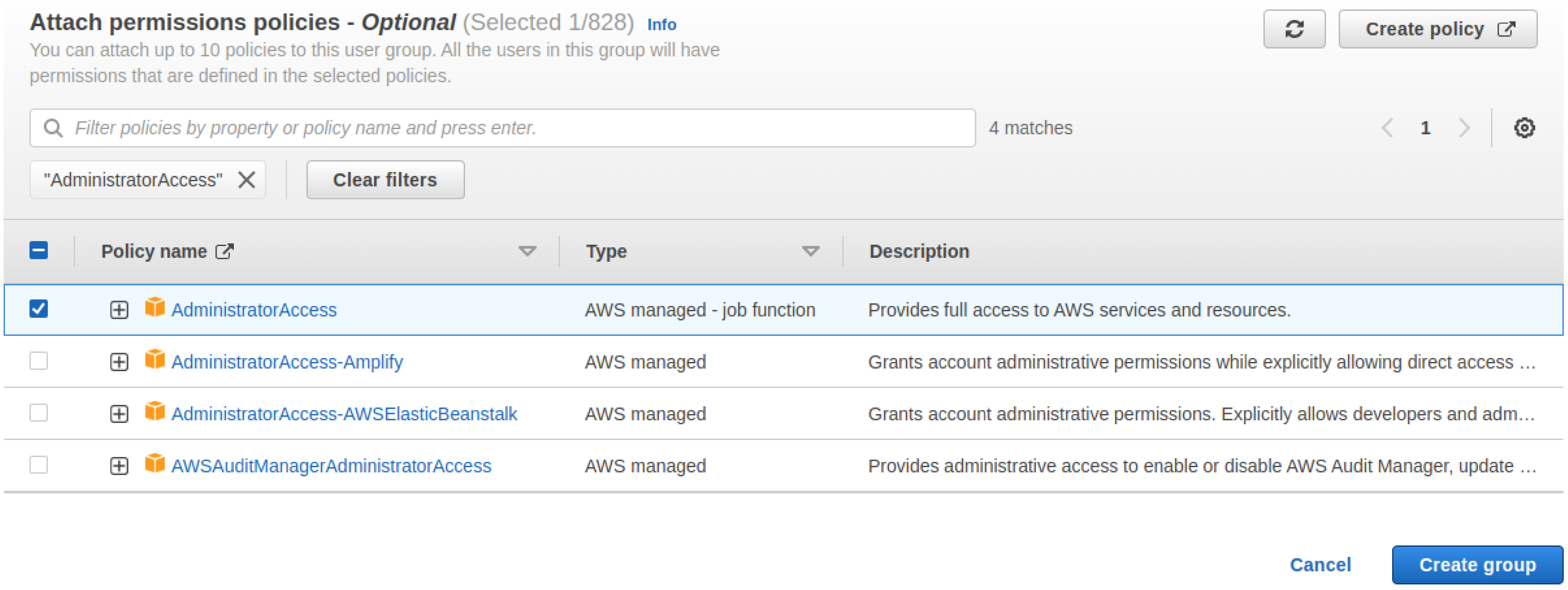
Name the group

User group name
Enter a meaningful name to identify this group.

admin

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

Attach permission policies

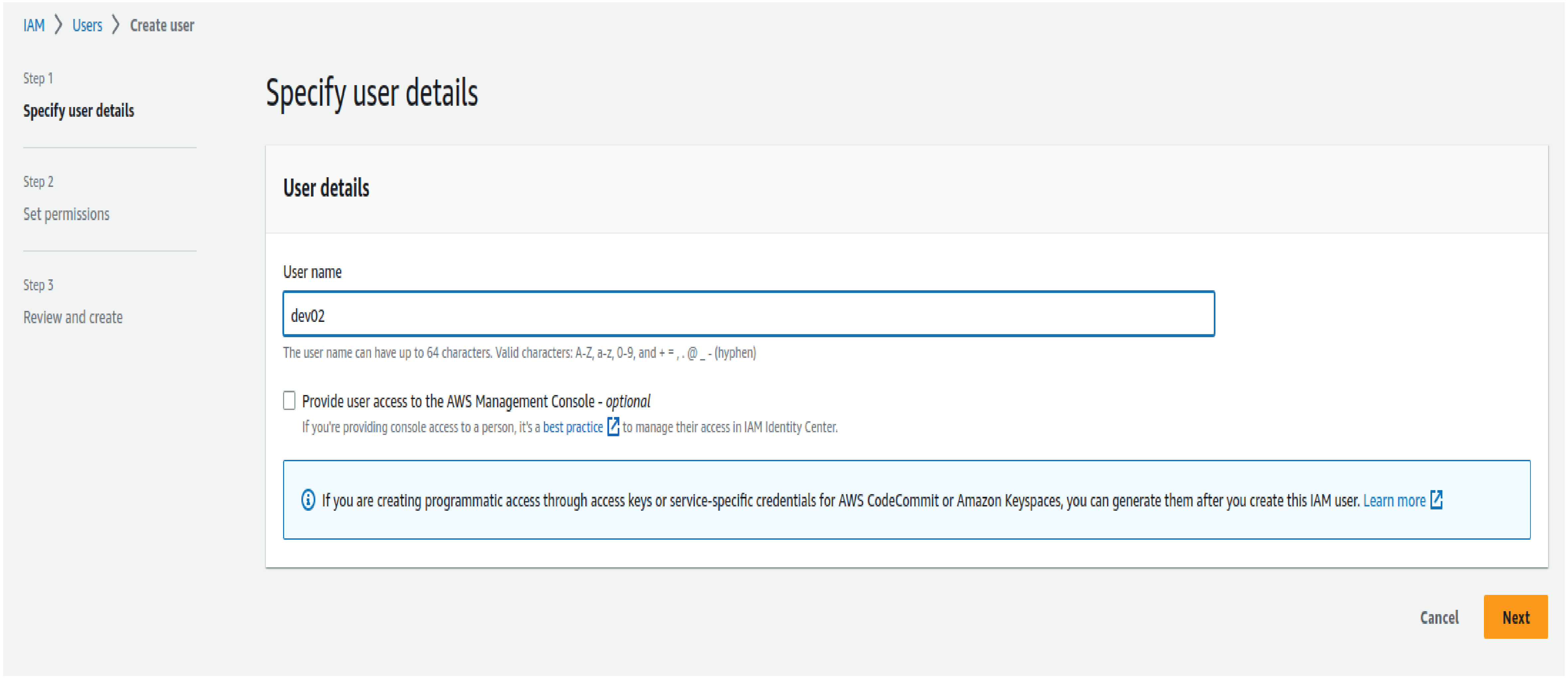


Attach permissions policies - *Optional* (Selected 1/828) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

	Policy name	Type	Description
✓	AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.
	AdministratorAccess-Amplify	AWS managed	Grants account administrative permissions while explicitly allowing direct access …
	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissions. Explicitly allows developers and adm…
	AWSAuditManagerAdministratorAccess	AWS managed	Provides administrative access to enable or disable AWS Audit Manager, update …

Create IAM User



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

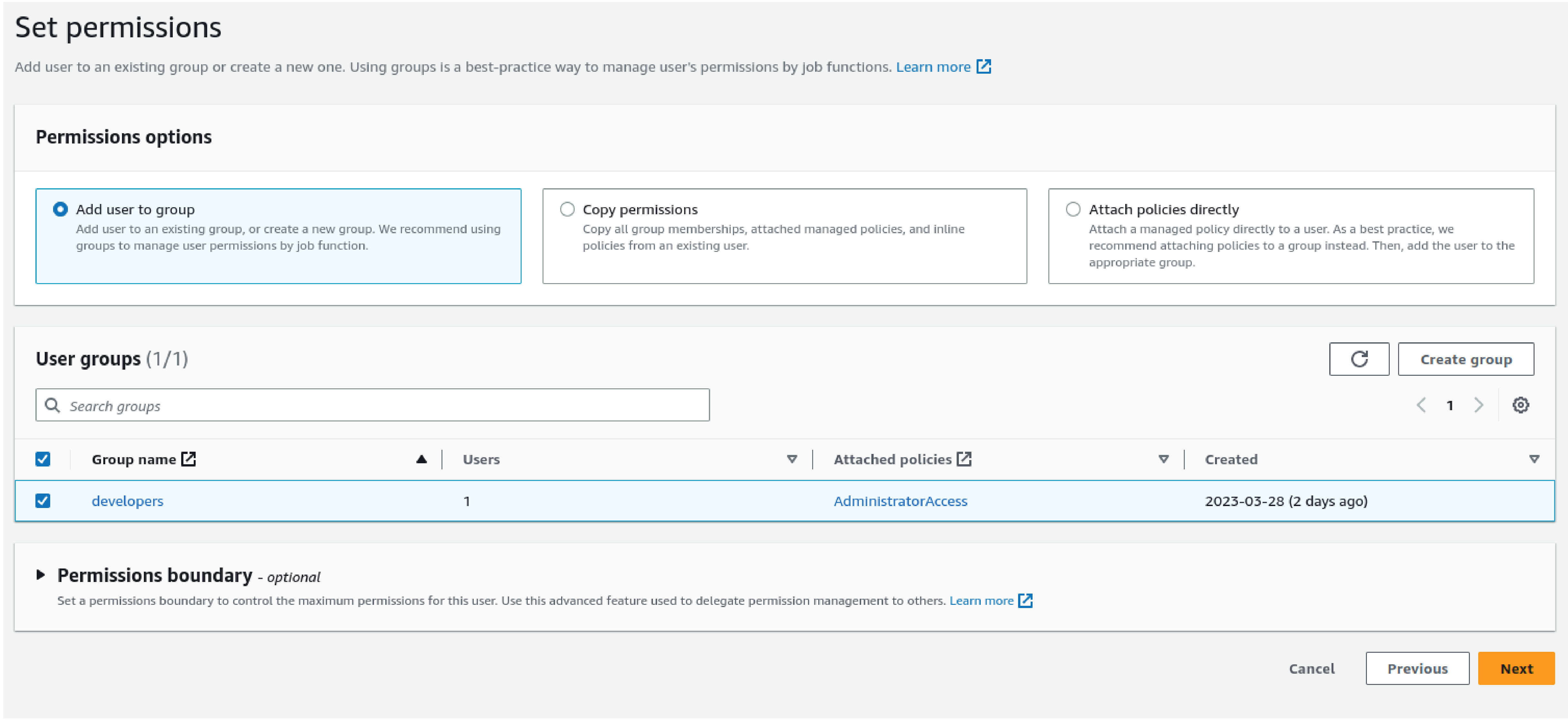
dev02

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.

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



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.

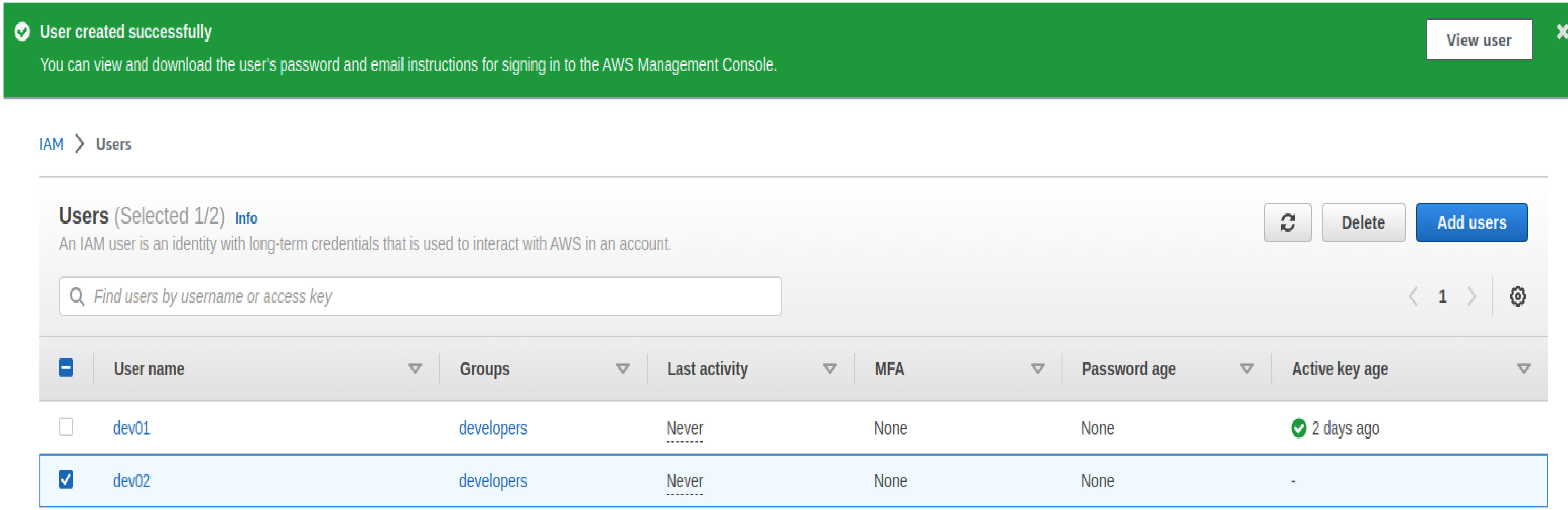
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/1)

	Group name	Users	Attached policies	Created
✓	developers	1	AdministratorAccess	2023-03-28 (2 days 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



User created successfully

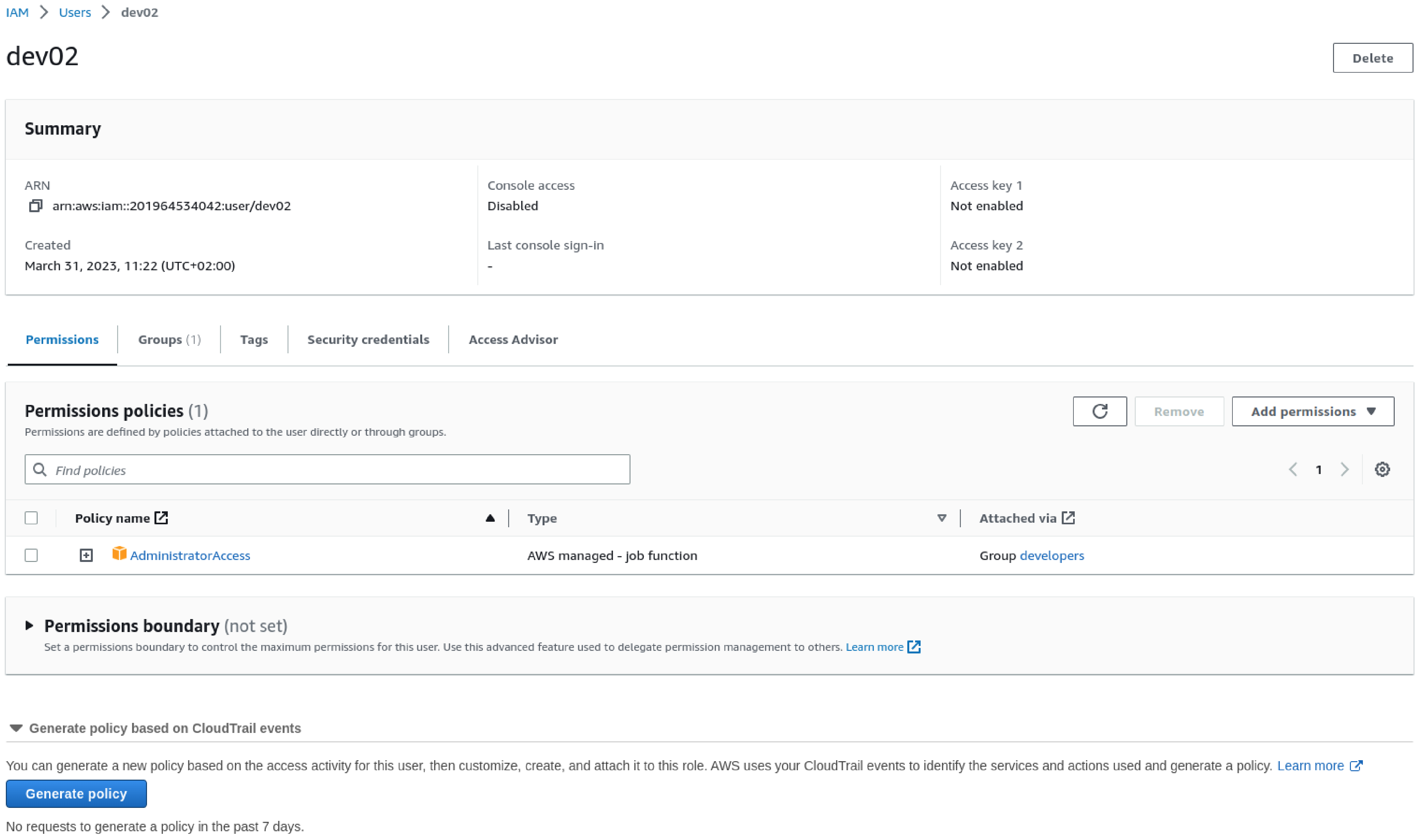
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users

Users (Selected 1/2) Info

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

	User name	Groups	Last activity	MFA	Password age	Active key age
☐	dev01	developers	Never	None	None	✓ 2 days ago
☑	dev02	developers	Never	None	None	-



IAM > Users > dev02

dev02

Delete

Summary

ARN arn:aws:iam::201964534042:user/dev02	Console access Disabled	Access key 1 Not enabled
Created March 31, 2023, 11:22 (UTC+02:00)	Last console sign-in -	Access key 2 Not enabled

Permissions | Groups (1) | Tags | Security credentials | Access Advisor

Permissions policies (1)

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

Remove | Add permissions

Find policies

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Group developers

Permissions boundary (not set)

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

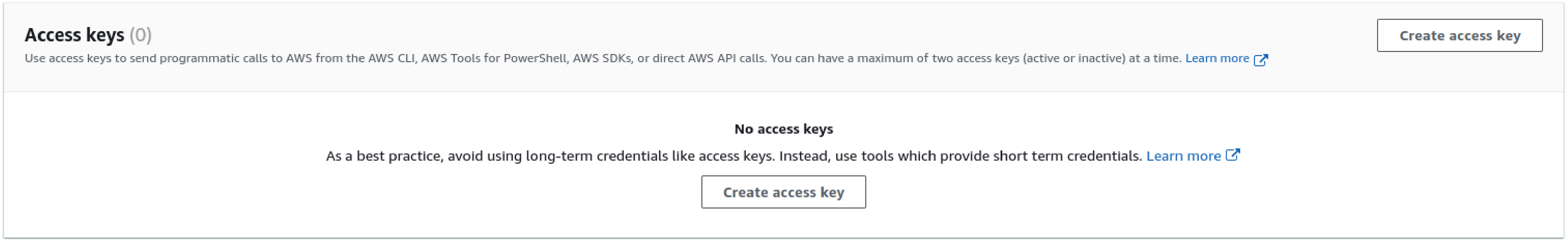
Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

Generate policy

No requests to generate a policy in the past 7 days.

Create Access Key



Access keys (0)

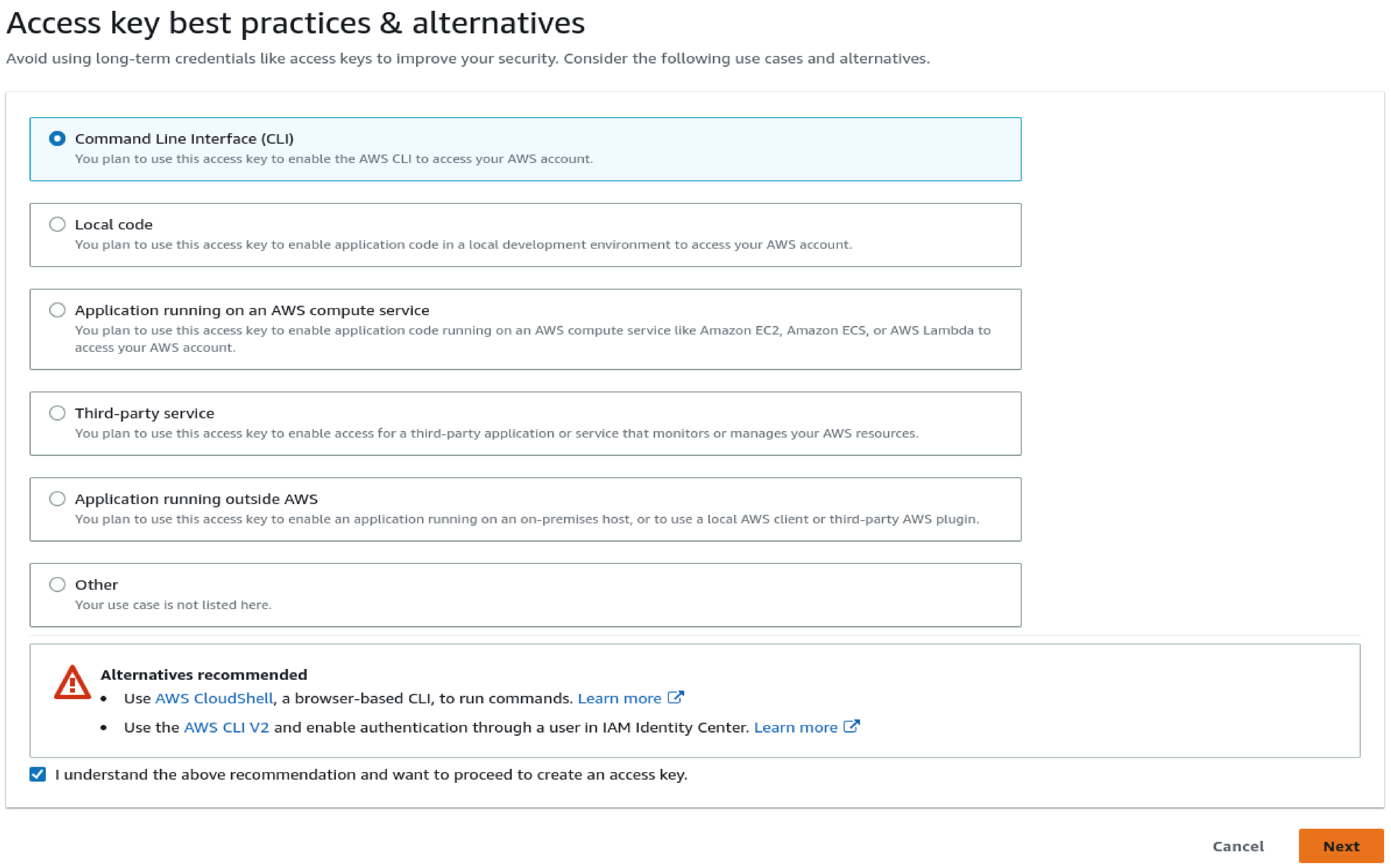
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

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. Learn more

Create access key



Access key best practices & alternatives

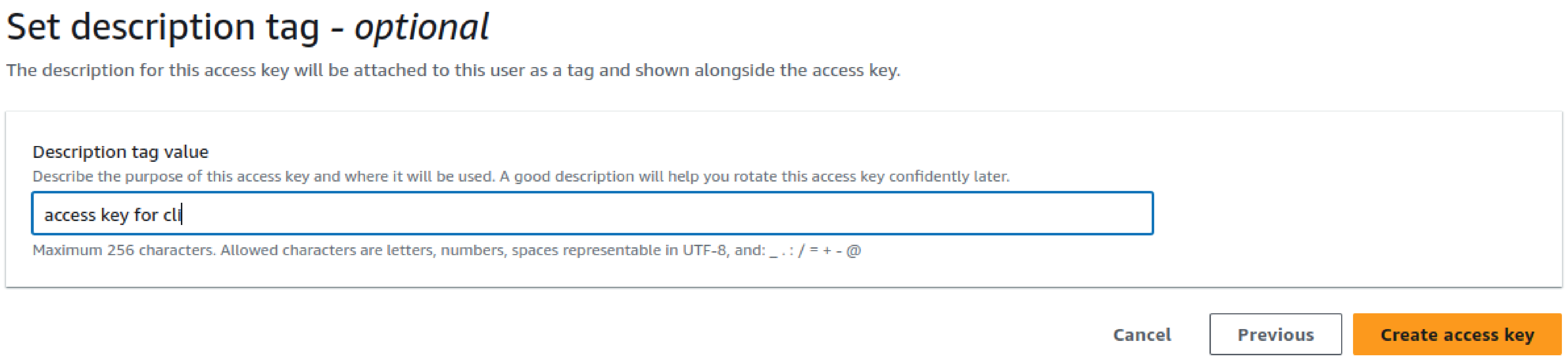
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

- ● **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.


Set description tag - *optional*

The description for this access key will be attached to this user as a tag and shown alongside the access key.

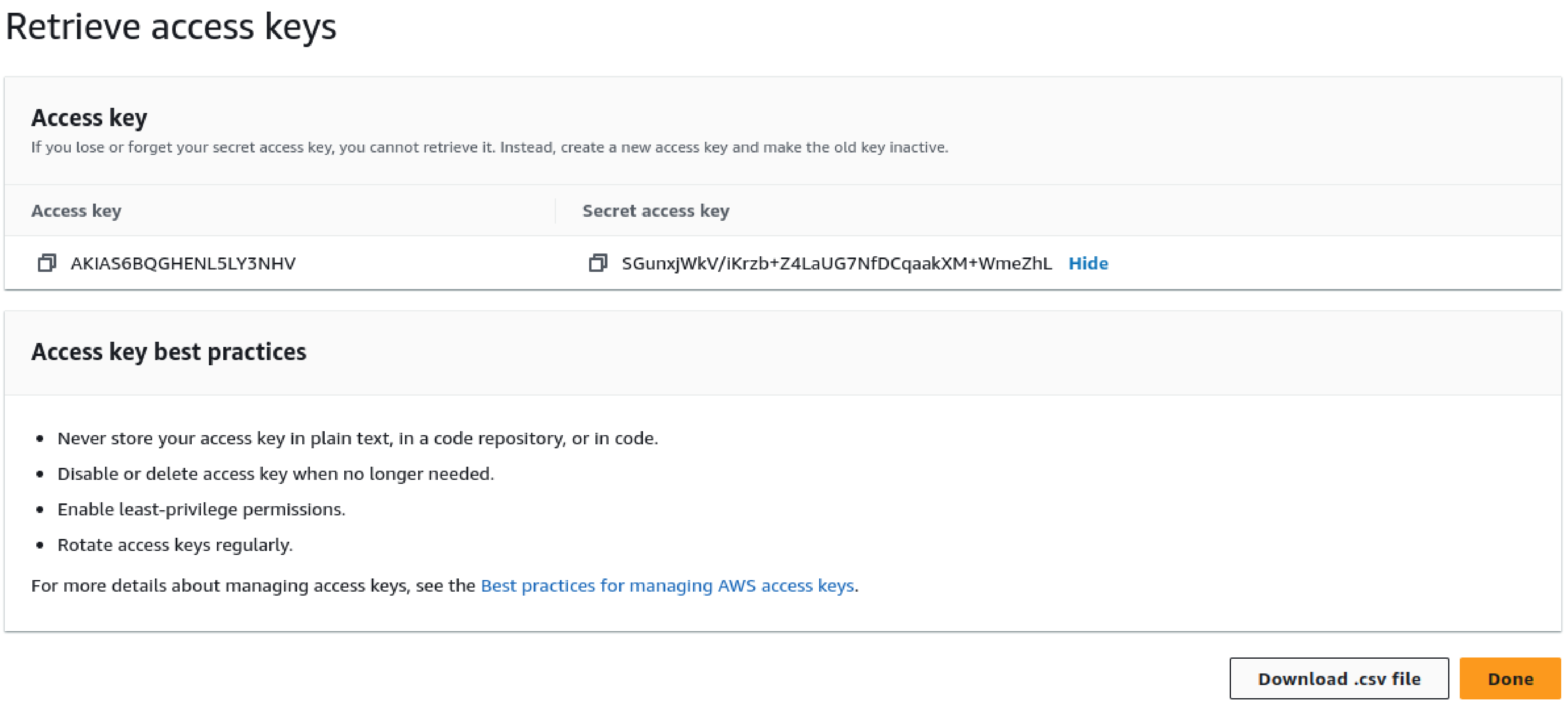
Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

access key for cli

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Download the csk and keep for future reference


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
AKIAS6BQGHENL5LY3NHV	SGunxjWkV/iKrzb+Z4LaUG7NfDCqaakXM+WmeZhL Hide

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.

SetUp AWS cli terminal.

- https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

- curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
- unzip awscliv2.zip
- sudo ./aws/install

```
~$ aws ecr get-login-password --region us-east-1
```

- Unable to locate credentials. You can configure credentials by running "aws configure".
- ~$ aws configure
- AWS Access Key ID [None]: ***************
- AWS Secret Access Key [None]: *******************
- Default region name [eu-west-1]: us-east-1
- Default output format [json]:
- ~$ aws ecr get-login-password --region us-east-1
- eyJwYXlsb2FkIjoiQi9EM0NDL1pMRnpSYmY5QWJXRER2MWx1eGJDMkRnUlVlTnlsa0dORVIweS8vb1ZZcTRyZ2llalMrMjg0d1RudnlJR0syU3JwMUxJcVpyZ***1CRUVER0VqRHRoOVA5cUlNKzhkMVFJQkVJQTcxK3B0ZkJyWE9Lb2g3Ujg4TFZnR2w1cjlIUWVHT3RKbHlKMUQyOUU5enNNTzB3S2lrd2Ftc05rNTlzQnROWk5JNGQrNmFkK0dNVTRSL1Z3PSIsInZlcnNpb24iOiIyIiwidHlwZSI6IkRBVEFfS0VZIiwiZXhwa*************

Validate access

- ~$ aws s3api list-buckets --query "Buckets[].Name"
- [
-     "elasticbeanstalk-us-east-1-201964534042"
-]