How to Create an AWS IAM User: A Step-by-Step Guide

Prerequisites

Before diving into the process of creating an AWS IAM user, you need to have the following:

1. **AWS Account**: You should have access to an AWS account with sufficient permissions.
2. **Basic Understanding of AWS Cloud**: A general knowledge of AWS services and the AWS Management Console.

Introduction to AWS IAM

AWS Identity and Access Management (IAM) is a service that helps you securely manage access to AWS services and resources. IAM allows you to control who can perform actions (like launching an EC2 instance, reading S3 data, etc.), what resources they can access, and under what conditions they can access them.

Key Benefits of AWS IAM

- **Granular Permissions**: You can specify concrete actions for each IAM user, role, or group, ensuring a high level of security.

© 2025 StemUp. All rights reserved.

- **Identity Federation**: IAM allows integration with other identity providers (like Google or Facebook) for Single Sign-On (SSO) capabilities.
- **Multi-Factor Authentication (MFA)**: Enhances security by requiring two forms of identification for access.
- **Fine-Grained Access Control**: Use policies to control access to services like S3, EC2, and others based on various conditions (e.g., time, IP address, etc.).

Why You Should Use IAM

When you create an AWS account, you're initially granted access to all AWS services. However, in a production environment, granting this level of access to every user is not ideal. IAM helps you assign specific roles to users, enabling them to access only their needed services. This not only enhances security but also ensures that services are not misused.

Step-by-Step Guide to Creating an AWS IAM User

Step 1: Log into the AWS Management Console

Head over to the AWS Management Console. You need to sign in as a user with administrative privileges. If this is your first time logging in, you're likely using the "root" user account, which gives full access to all resources.

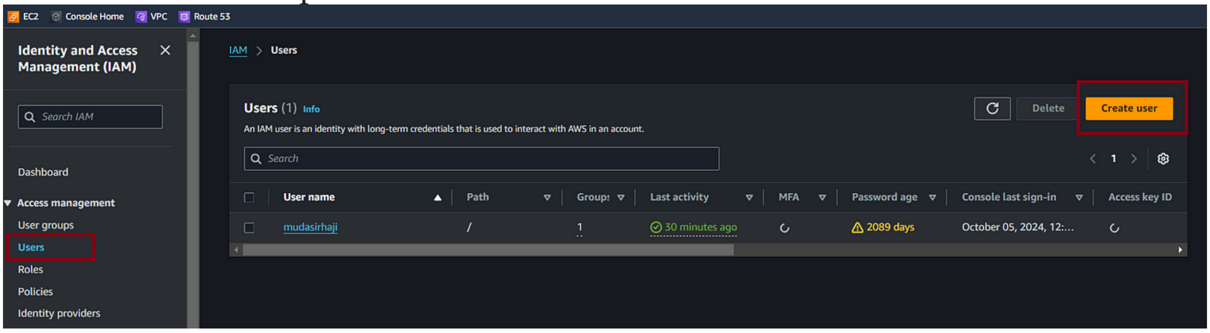
Step 2: Navigate to IAM

Once logged in, search for **IAM** in the AWS services search bar, and click on the **IAM** service to open the IAM dashboard.

Step 3: Add a New User

© 2025 StemUp. All rights reserved.

In the left-hand menu of the IAM dashboard, click **Users** and then click the **Create user** button at the top.



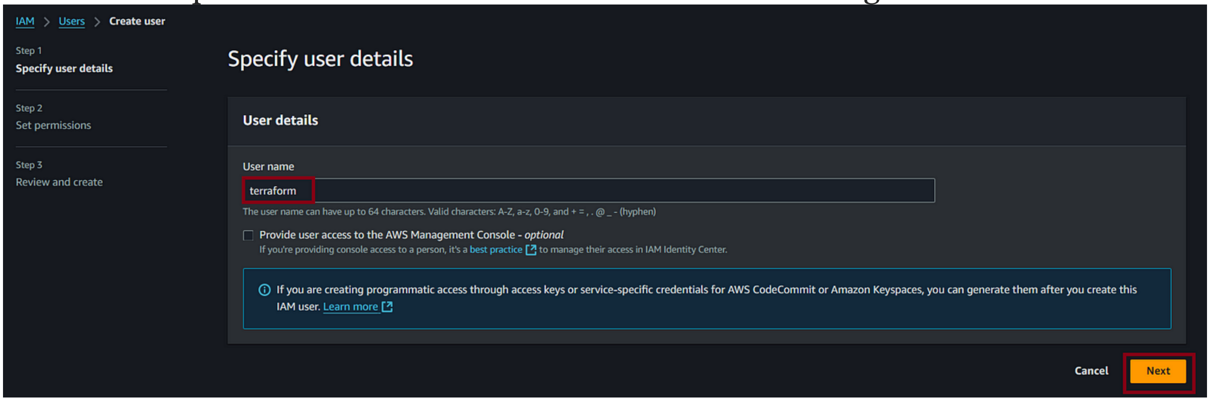
Users (1) Info

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

☐	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
☐	mudasirhaji	/	1	30 minutes ago		2089 days	October 05, 2024, 12:...	

Step 4: Enter User Details

Choose a unique username for the user. This will be used to sign in.



User name: terraform

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

Step 5: Assign Permissions

This step lets you decide the level of access the user will have. You can assign permissions in the following ways:

1. **Add user to group**: If you've already set up IAM groups with specific permissions, simply add the user to an appropriate group.
2. **Copy permissions from existing users**: Copy the permissions of a previously created user.

© 2025 StemUp. All rights reserved.

3. **Attach policies directly**: You can attach specific AWS-managed or custom policies directly to the user.

For beginners, it is often easier to add the user to a predefined AWS group like **AdministratorAccess** or **ReadOnlyAccess**. However, in this guide, we'll focus on **attaching policies directly** and **assigning administrative access** to provide comprehensive permissions for the new user.

Assigning Policies Directly and Assigning Administrative Access

1. **Select "Attach policies directly"**:

- Choose this option to assign specific permissions to the user by attaching policies individually.

2. **Search for the AdministratorAccess Policy**:

- In the policy list, use the search bar to find the **AdministratorAccess** policy. This is an AWS-managed policy that grants full access to all AWS services and resources.

3. **Attach the AdministratorAccess Policy**:

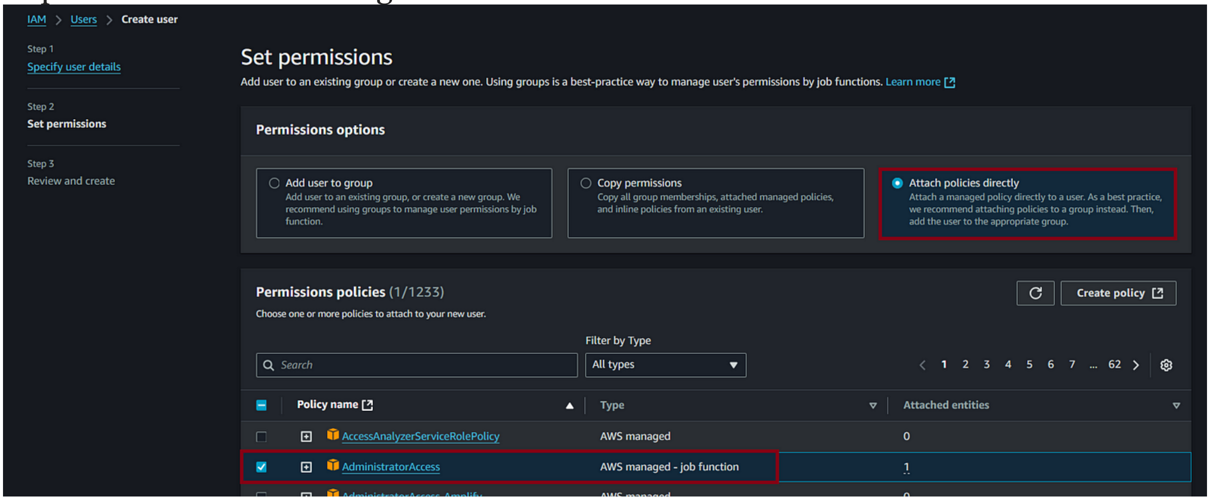
- Check the box next to **AdministratorAccess** to attach it to the user.

4. **Proceed to the Next Step**:

© 2025 StemUp. All rights reserved.

- After selecting the desired policy, click **Next.**

Note: Assigning the **AdministratorAccess** policy grants the user full administrative privileges. Ensure that you assign such permissions only to trusted individuals who require full access to manage AWS resources.



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

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 (1/1233)

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

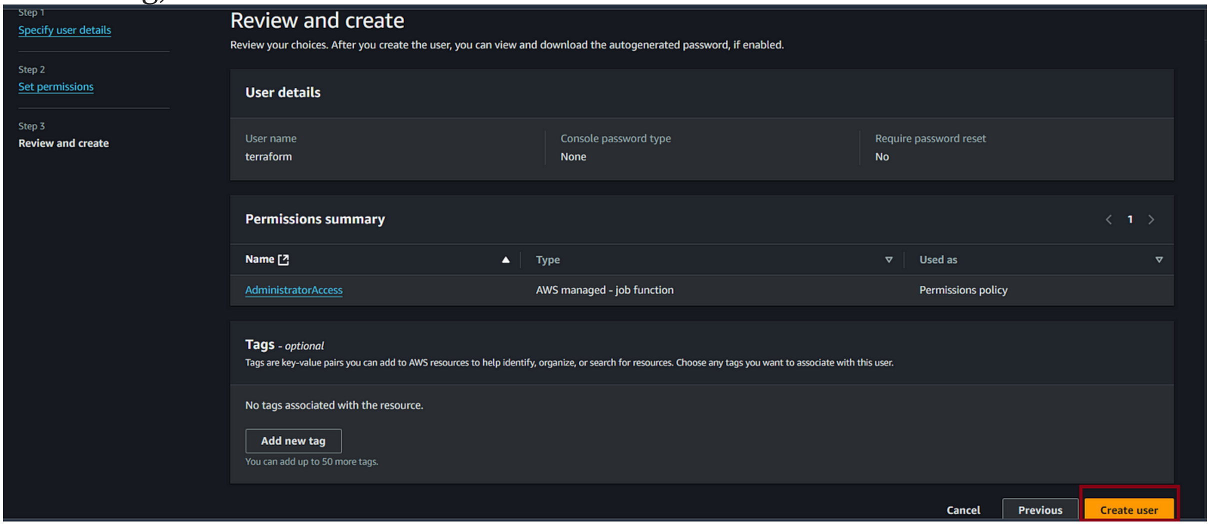
Filter by Type: All types

Search

	Policy name	Type	Attached entities
□	AccessAnalyzerServiceRolePolicy	AWS managed	0
☑	AdministratorAccess	AWS managed - job function	1
□	AdministratorAccess-Amplify	AWS managed	0

Step 6: Review and Create

Review the user details to ensure everything is configured as expected. After confirming, click **Create User**.



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

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
terraform	None	No

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	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.

You can add up to 50 more tags.

Step 7: Create Access Keys for Programmatic Access

© 2025 StemUp. All rights reserved.

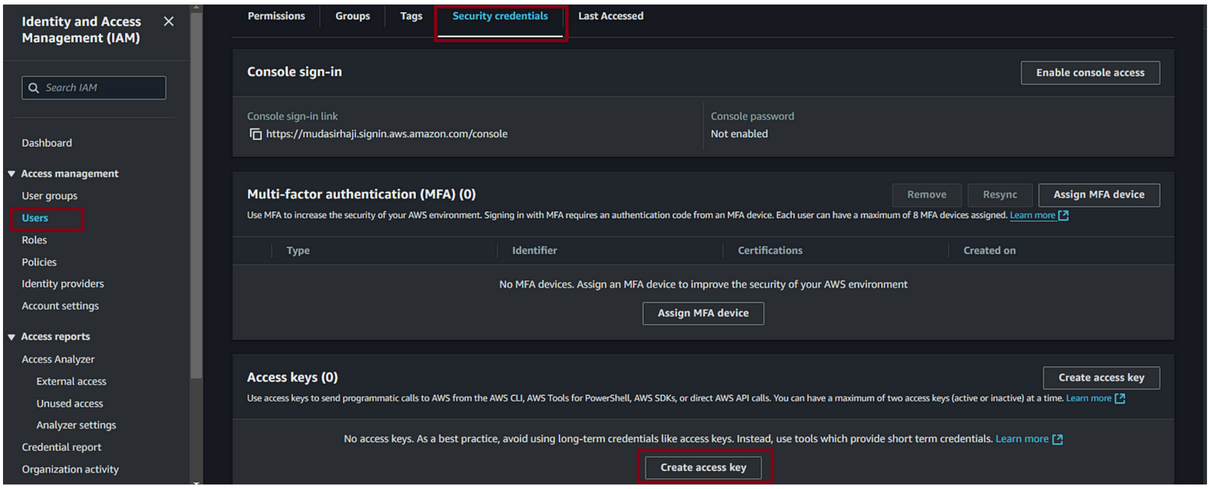
After you've created the user, the next step is to generate access keys if you selected **Programmatic Access** during the user creation process. These keys are crucial if the user will access AWS services via the CLI, SDK, or APIs.

- **Click on the Newly Created User**:

- Once the user is created, return to the **Users** section in the IAM dashboard.
- Find the newly created user in the list and click on the username to open the user details.

2. **Click on the "Security Credentials" Tab**:

- After clicking on the user, you will see various tabs. Select the **Security Credentials** tab.



Console sign-in

Console sign-in link: https://mudasirhaji.signin.aws.amazon.com/console

Console password: Not enabled

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

Type	Identifier	Certifications	Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment

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

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

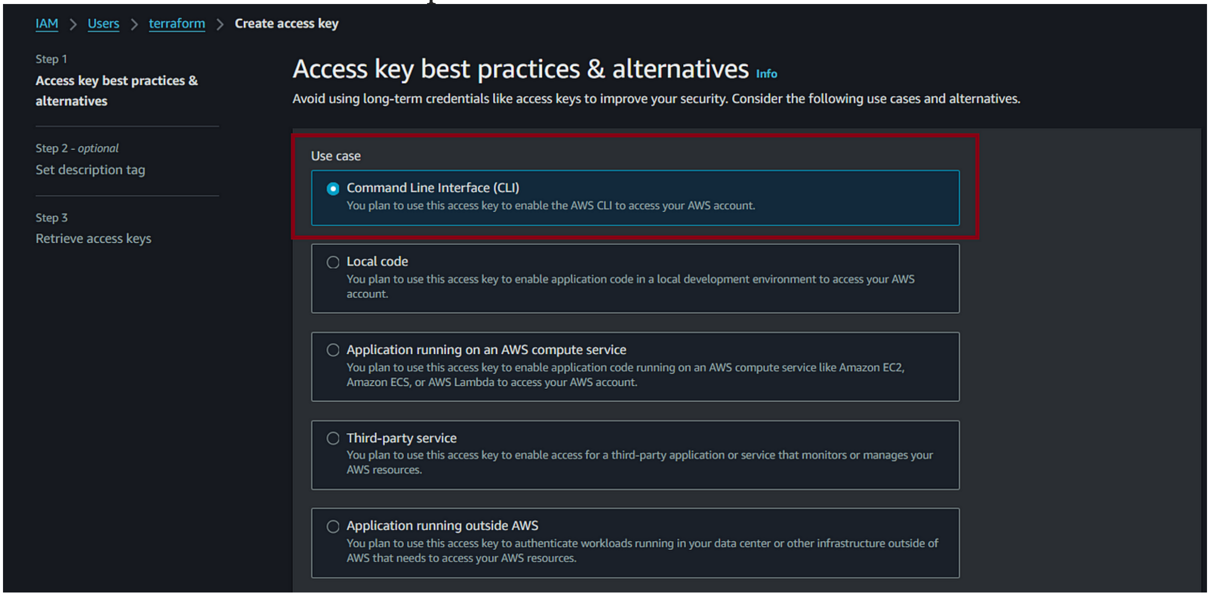
3. **Click on "Create Access Key"**:

  - Scroll down to the **Access Keys** section.

© 2025 StemUp. All rights reserved.

- Click the **Create access key** button.

- Choose the cli option and click next



IAM > Users > terraform > 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 Info

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.

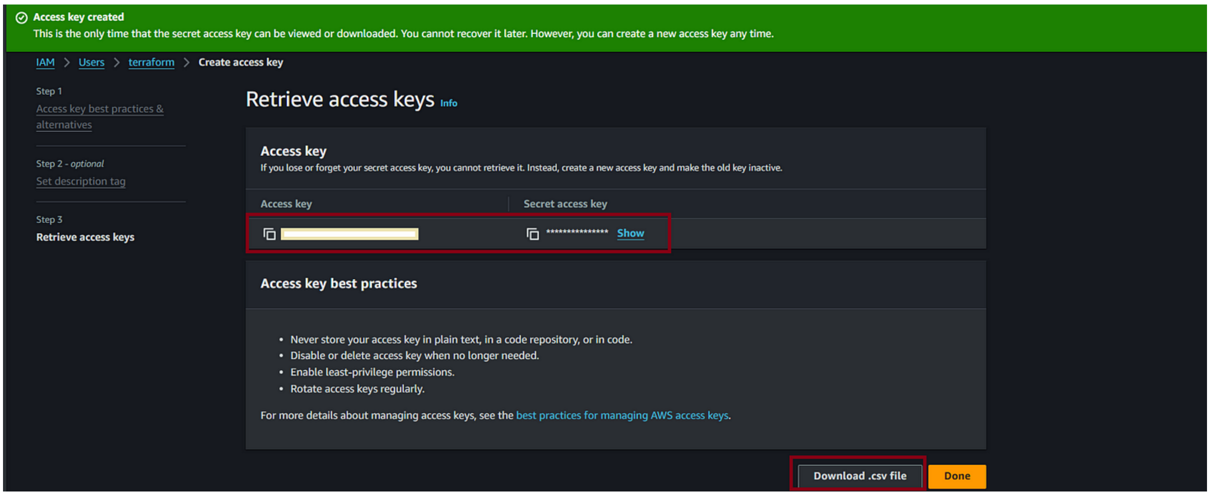
○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

4. **Download or Copy Access Keys**:

- AWS will generate an **Access Key ID** and a **Secret Access Key**.

- Make sure to **download** or **copy** these keys at this point. You won't be able to retrieve the **Secret Access Key** again after this step. Store these keys in a safe location, such as a password manager or an encrypted file.

© 2025 StemUp. All rights reserved.



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 > terraform > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

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
[redacted]	*************** 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.

Note: The access key and secret key are critical for programmatic access to AWS resources. Ensure they are handled securely.

Step 8: Secure the User (Optional but Recommended)

- **Enable MFA (Multi-Factor Authentication)**:

- For enhanced security, it's recommended to enable MFA for the user. MFA provides an extra layer of security by requiring not just a password, but also a temporary code from an authenticator app.

To enable MFA, click on the user's profile in the IAM dashboard, navigate to the **Security Credentials** tab, and click on **Assign MFA device**. Follow the instructions to set it up.

2. Set a Strong Password Policy:

- If you provide console access, ensure that the user follows a strong password policy. IAM allows you to configure password policies under the **Account**

© 2025 StemUp. All rights reserved.

Settings in the IAM dashboard. You can enforce password complexity, expiration, and rotation.

Best Practices for Managing IAM Users

Follow the Principle of Least Privilege:

Always assign only the minimum permissions a user needs to perform their tasks. Over-permissions can expose your AWS environment to unnecessary risks.

2. **Use IAM Groups**:

- Instead of attaching policies directly to individual users, consider creating IAM groups (e.g., `Developers`, `Admins`) and assigning users to these groups. This simplifies permission management and ensures consistency.

3. **Regularly Rotate Credentials**:

- It's a best practice to rotate IAM credentials (passwords, access keys) regularly to mitigate the risk of credential compromise.

4. **Monitor User Activity**:

- Enable logging through AWS CloudTrail to track all activities performed by IAM users. This helps in auditing and investigating potential security incidents.

© 2025 StemUp. All rights reserved.

Conclusion

AWS IAM is a vital tool in managing access control in AWS environments. Creating users, attaching appropriate policies, and managing access keys are crucial steps toward securing your cloud infrastructure. By following best practices like enabling MFA, using IAM groups, and applying the principle of least privilege, you can ensure that your AWS resources remain secure.

You've now successfully created an IAM user, assigned permissions, and generated access keys for programmatic access. These steps help maintain a secure and well-structured AWS environment.

© 2025 StemUp. All rights reserved.