MPRASHANT

AWS IAM

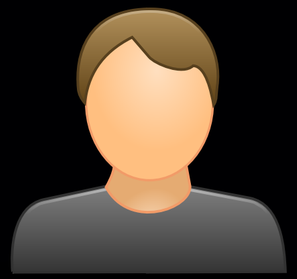
AWS Identity & Access Management

MPRASHANT

IAM is a service that helps you securely control access to AWS resources.

It allows you to manage users, roles, and permissions to define who can access what within your AWS environment.

MPRASHANT



root

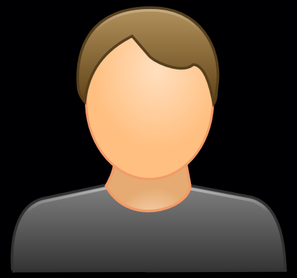
MPRASHANT

- Free Service: IAM is offered at no additional cost
- Global Service
- Root account created by default, shouldn’t be used or shared

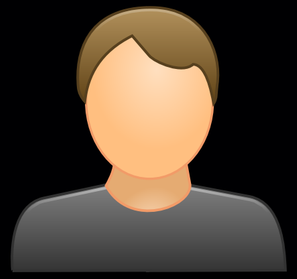
MPRASHANT

- **Create Users**: You can create individual user accounts for people who need access to your AWS resources.
- **Assign Permissions**: You can assign specific permissions to users, groups, or roles to control what actions they can perform on AWS services.
- **Create Groups**: You can group users together and assign permissions to the group, making management easier for multiple users.

Users

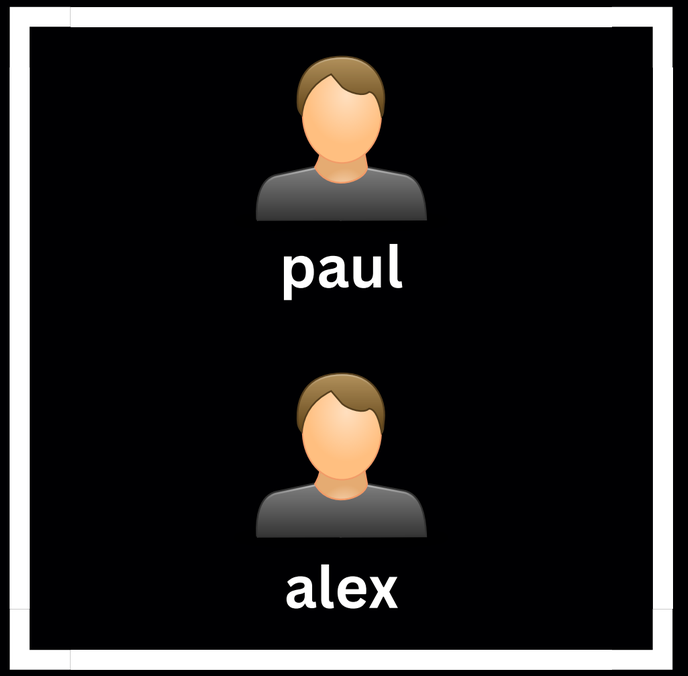


paul



alex

Group



operations

MPRASHANT

- **Create Roles:** You can create roles to assign temporary permissions to AWS services or users, especially useful for securely managing permissions across different AWS resources.
- **Define Policies**: You can create and attach custom policies to define fine-grained permissions for controlling access to AWS resources.
- **Manage Federated Access:** IAM allows integrating with external identity providers (like Active Directory) for centralized management of user access across AWS.

MPRASHANT

AWS IAM

MFA

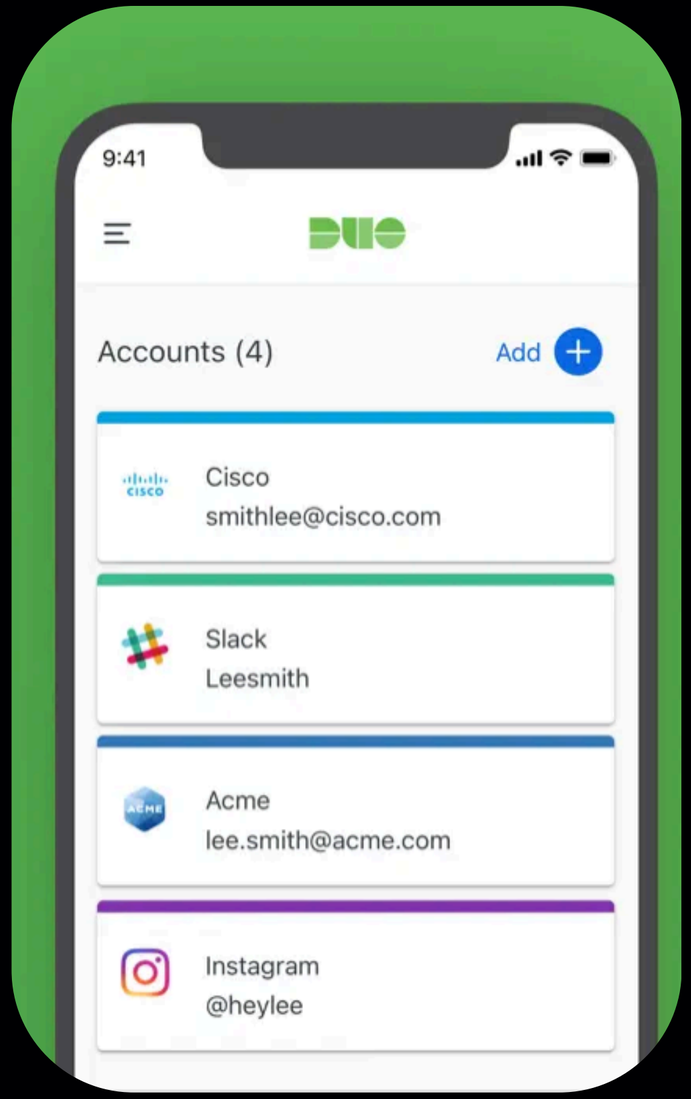
MFA (Multi-Factor Authentication) is an extra layer of security that requires users to provide two or more forms of verification, like a password and a code from their phone, to access their accounts.

MPRASHANT

AWS IAM

MFA

username + password + security code


9:41
Accounts (4)
Add
Cisco
smithlee@cisco.com
Slack
Leesmith
Acme
lee.smith@acme.com
Instagram
@heylee

DUO Mobile

MPRASHANT

AWS IAM

Ways of accessing AWS

MPRASHANT

- The **AWS Management Console** provides a graphical, web-based approach.
- The **AWS CLI** provides a command-line, scripting approach.
- **AWS SDKs and APIs** offer programmatic, code-based access, allowing users to integrate AWS directly into their applications.

MPRASHANT

AWS IAM Best Practices

- Avoid using root account except of account setup.
- Add user to a group and assign permission to group
- Use password policy or MFA
- Use ACCESS KEYS for CLI/SDK
- Never share ACCESS KEYS or Password
- Audit the permission using IAM credential report.