Amazon Web Service

AWS-IAM

Prakash Kumar-DevOps Trainer

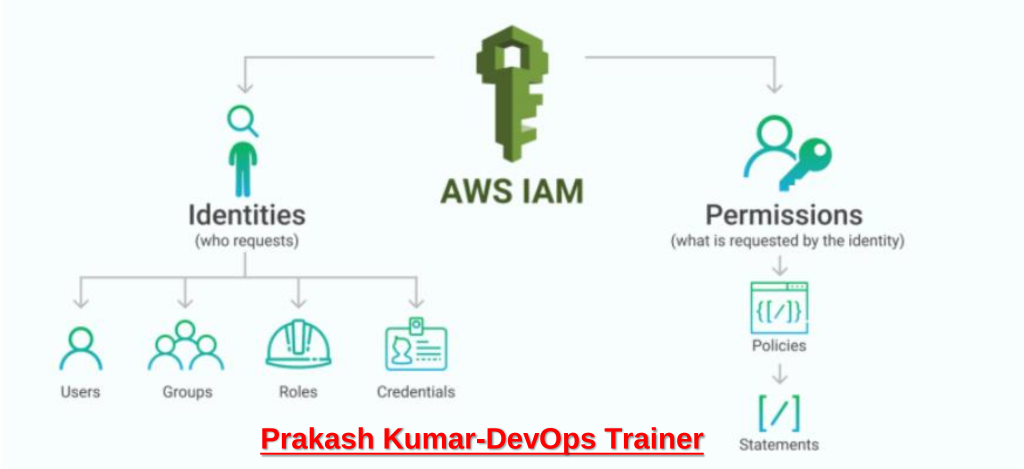
AWS IAM

What Is IAM?

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. You use IAM to control who can use your AWS resources (*authentication*) and what resources they can use and in what ways (*authorization*).

IAM is a Global service (it works in every AWS region)

Essentially, IAM allows you to manage users and their level of access to the AWS console. It is important to understand IAM and how it works, both for the exam and for administrating a organization's AWS account in real life.



AWS IAM- Features

IAM gives you the following features:

Shared access to your AWS account: You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key (by creating IAM User)

Granular permissions: You can grant different permissions to different people for different resources (Customer managed policy)

Secure access to AWS resources for applications that run on Amazon EC2: You can use IAM features to securely give applications that run on EC2 instances the credentials that they need in order to access other AWS resources

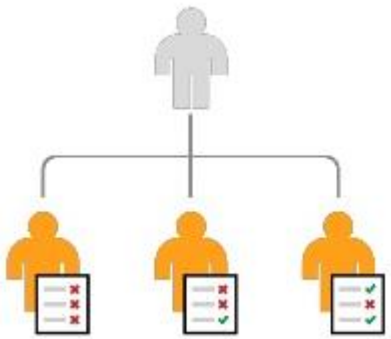
Multi-factor authentication (MFA): MFA

Identity Federation: Federation with Facebook, LinkedIn, Active Directory

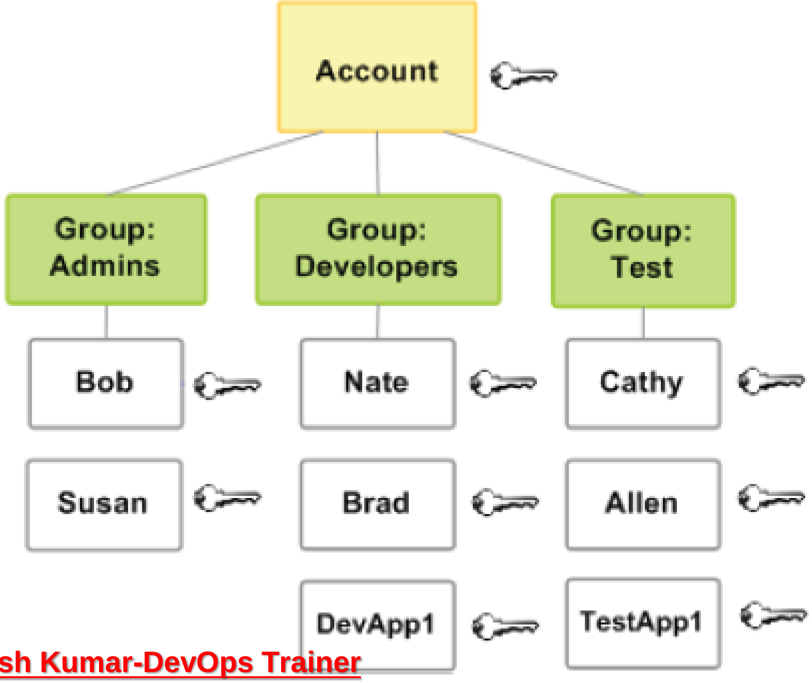
AWS IAM

- IAM User - Identity (think user) in AWS account
- IAM Group - Collection of IAM Users
- IAM Role - An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.
  (think One AWS services can call another AWS services on behalf of user)
- IAM Policy – A document that define one or more permission, Policies are collection of rules

User



Group:



Account

Group: Admins

Group: Developers

Group: Test

Bob

Susan

Nate

Brad

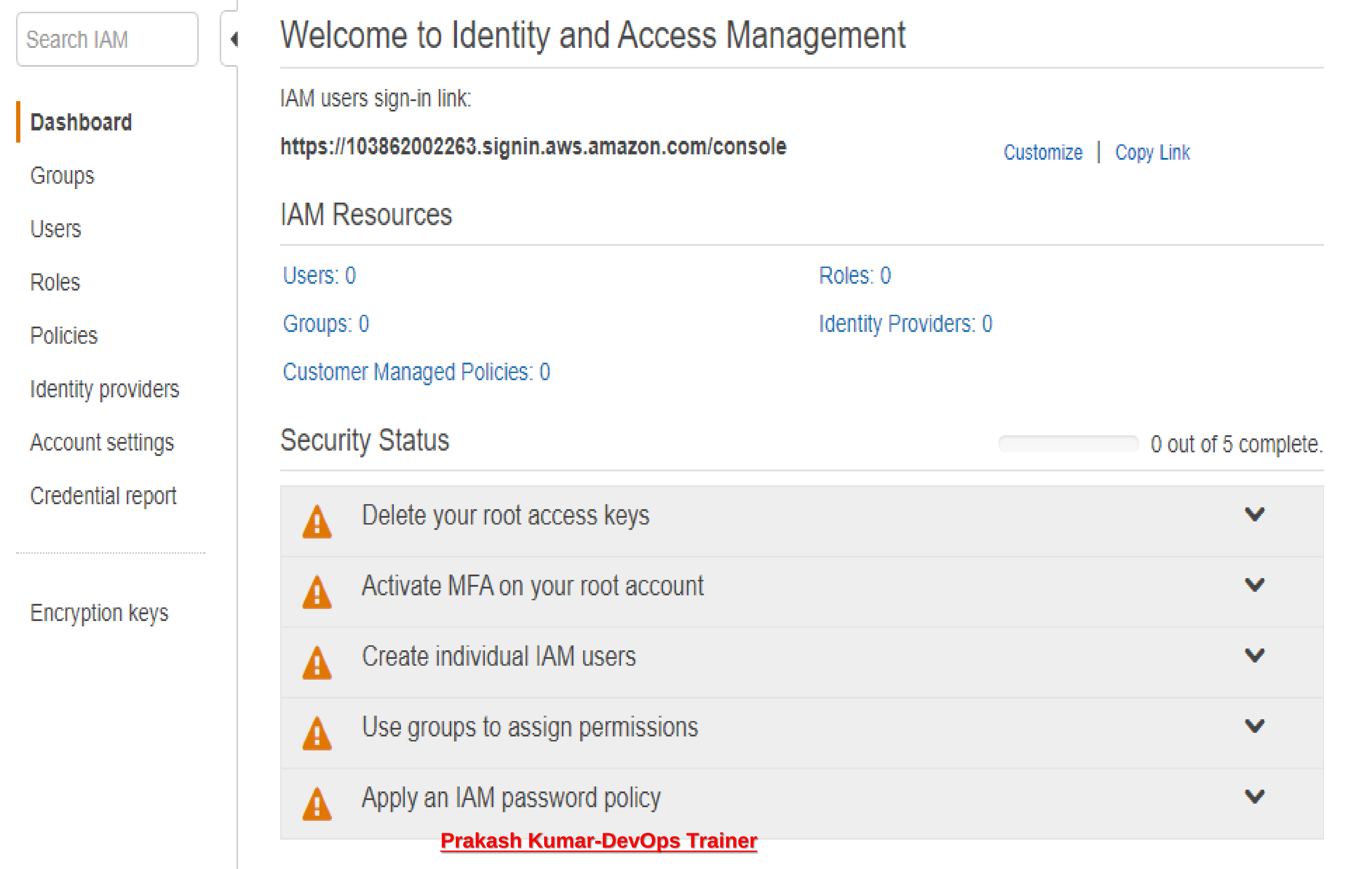
DevApp1

Cathy

Allen

TestApp1

AWS IAM-First Lab



Welcome to Identity and Access Management

IAM users sign-in link:

https://103862002263.signin.aws.amazon.com/console

IAM Resources

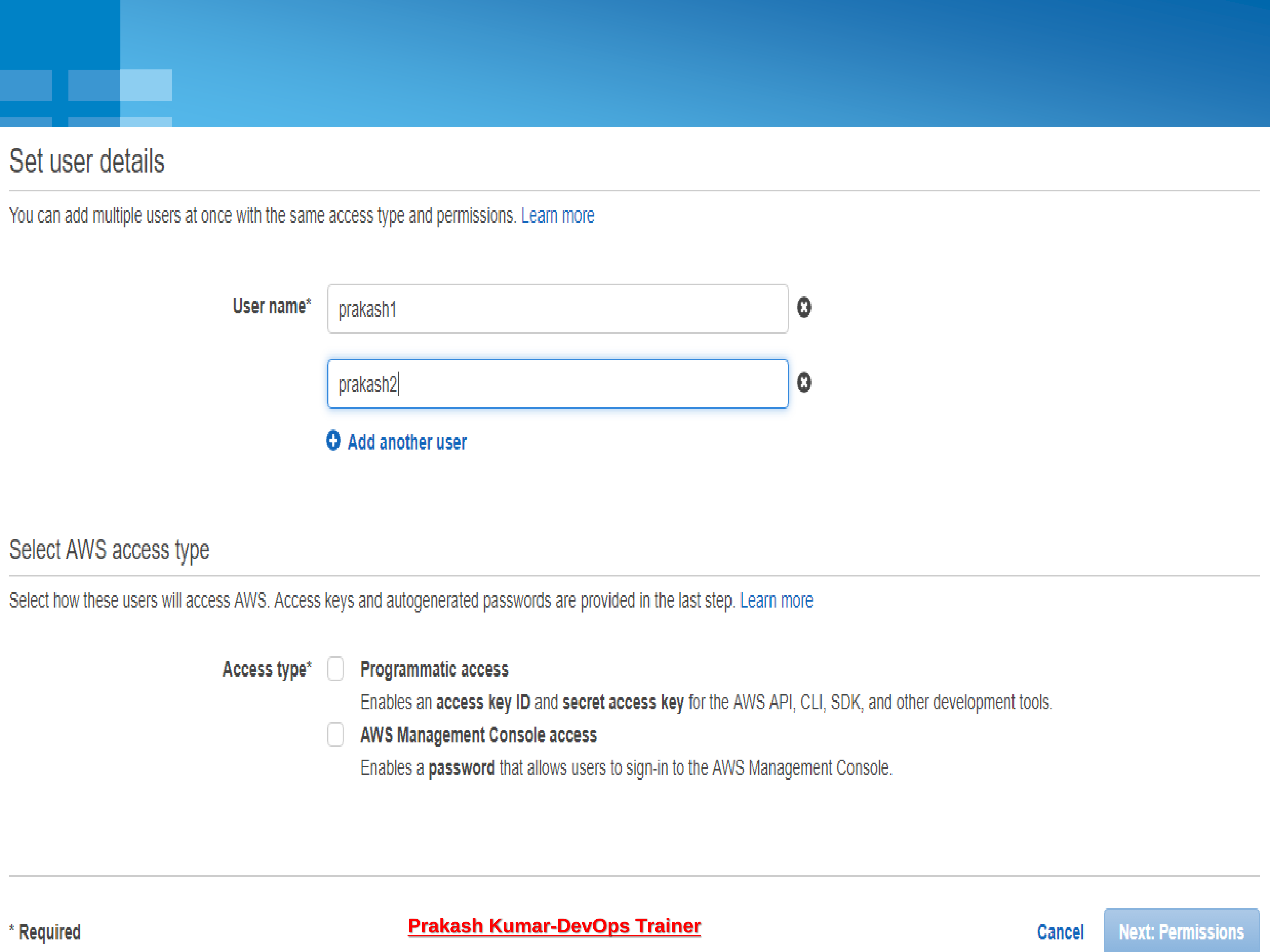
Users: 0	Roles: 0
Groups: 0	Identity Providers: 0
Customer Managed Policies: 0	

Security Status

0 out of 5 complete.

- Delete your root access keys
- Activate MFA on your root account
- Create individual IAM users
- Use groups to assign permissions
- Apply an IAM password policy

Prakash Kumar-DevOps Trainer



Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* prakash1

prakash2

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

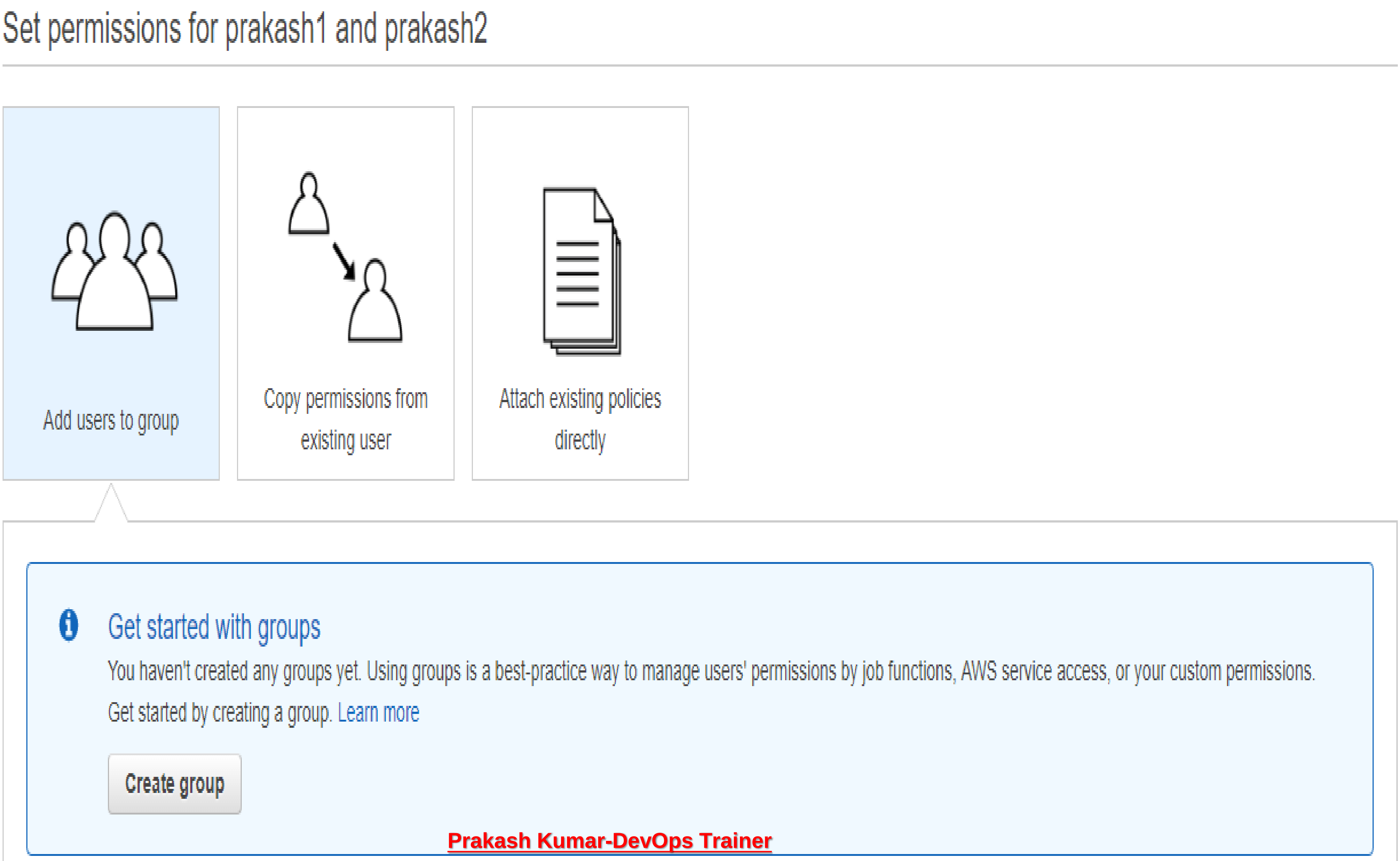
Access type* □ **Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ **AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required



Set permissions for prakash1 and prakash2

Add users to group

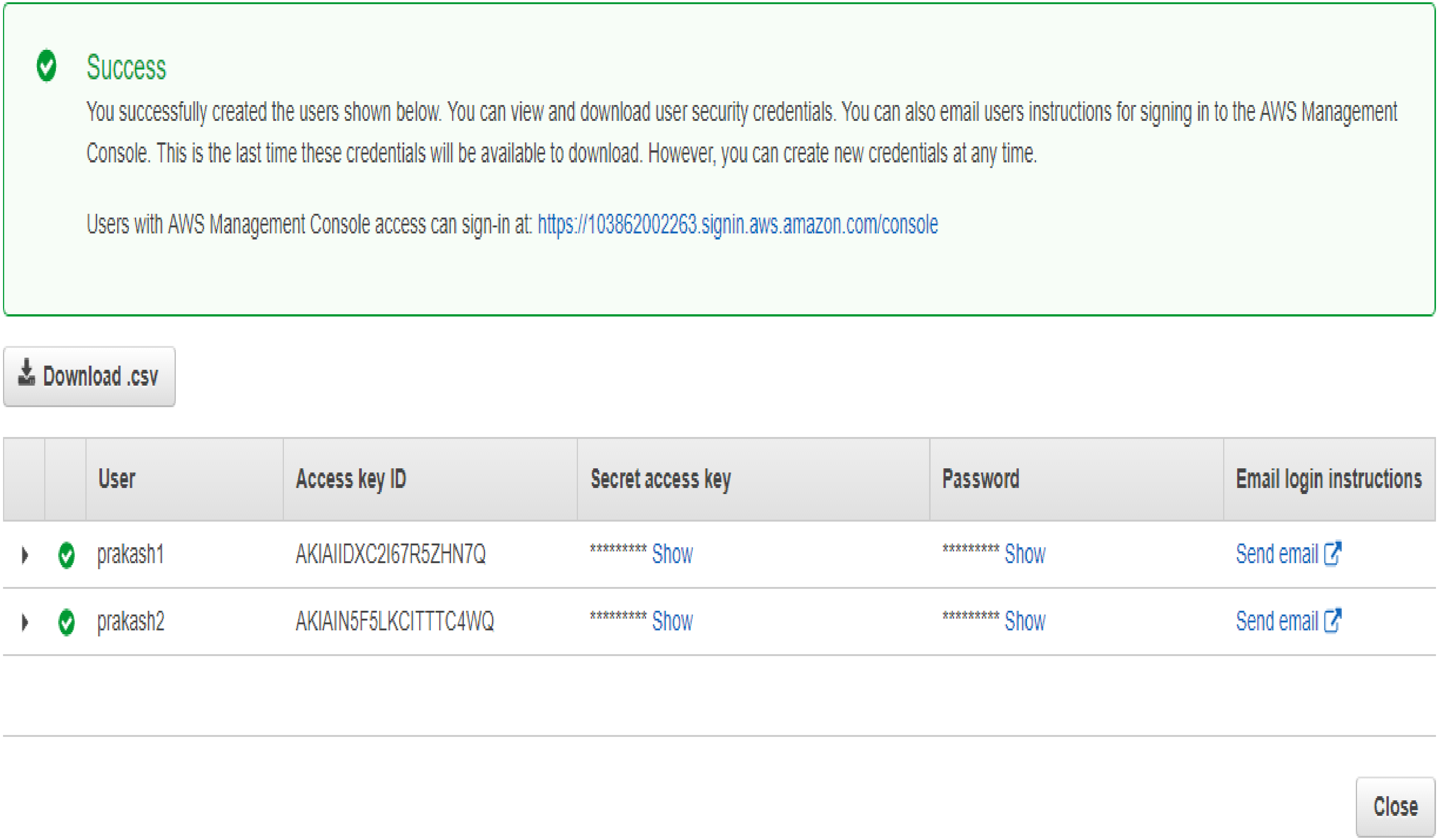
Copy permissions from existing user

Attach existing policies directly

Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. Learn more

Prakash Kumar-DevOps Trainer



Success

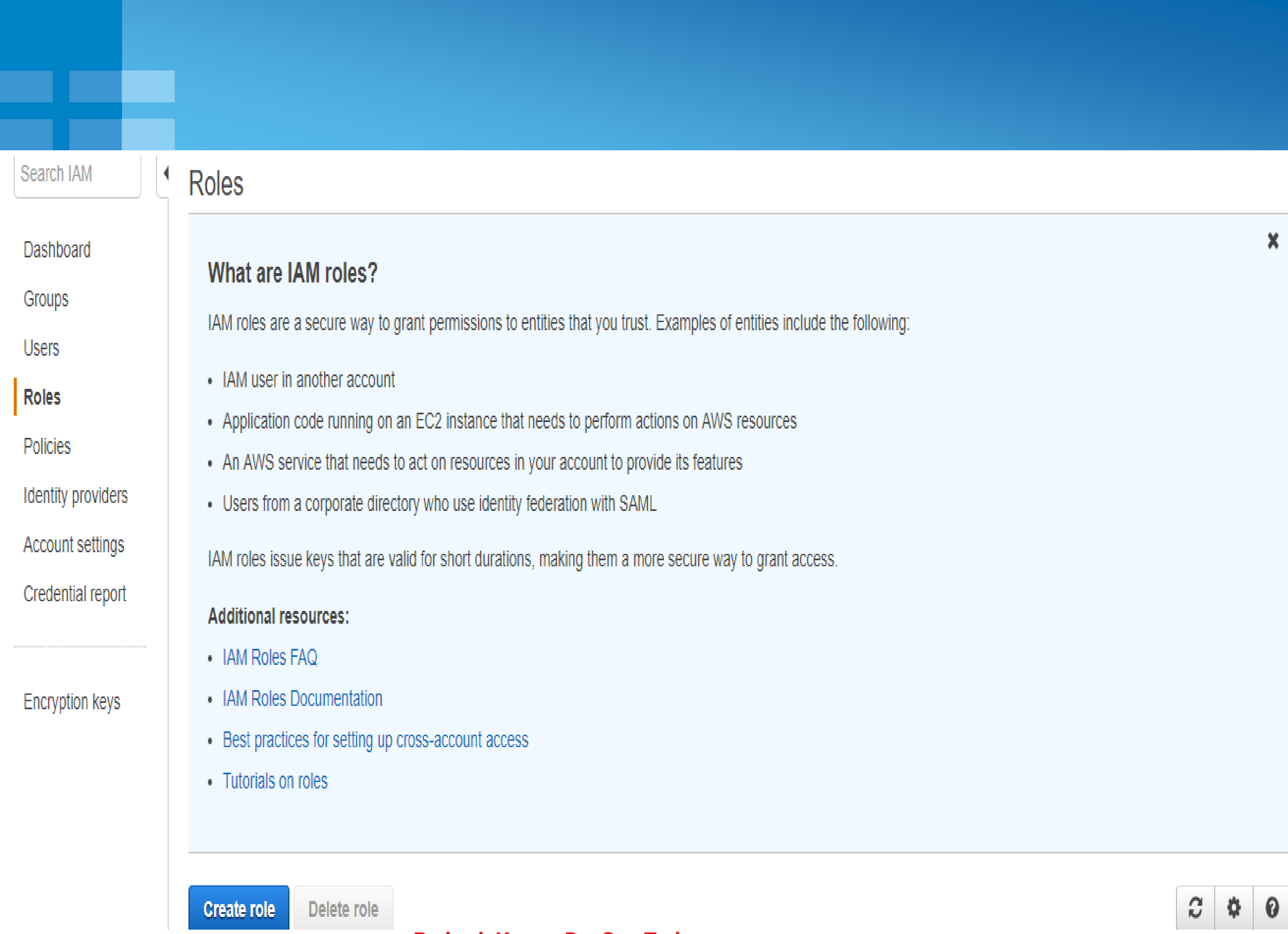
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://103862002263.signin.aws.amazon.com/console

Download .csv

		User	Access key ID	Secret access key	Password	Email login instructions
▸	✓	prakash1	AKIAIIDXC2I67R5ZHN7Q	********* Show	********* Show	Send email
▸	✓	prakash2	AKIAIN5F5LKCITTTC4WQ	********* Show	********* Show	Send email

Close



Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

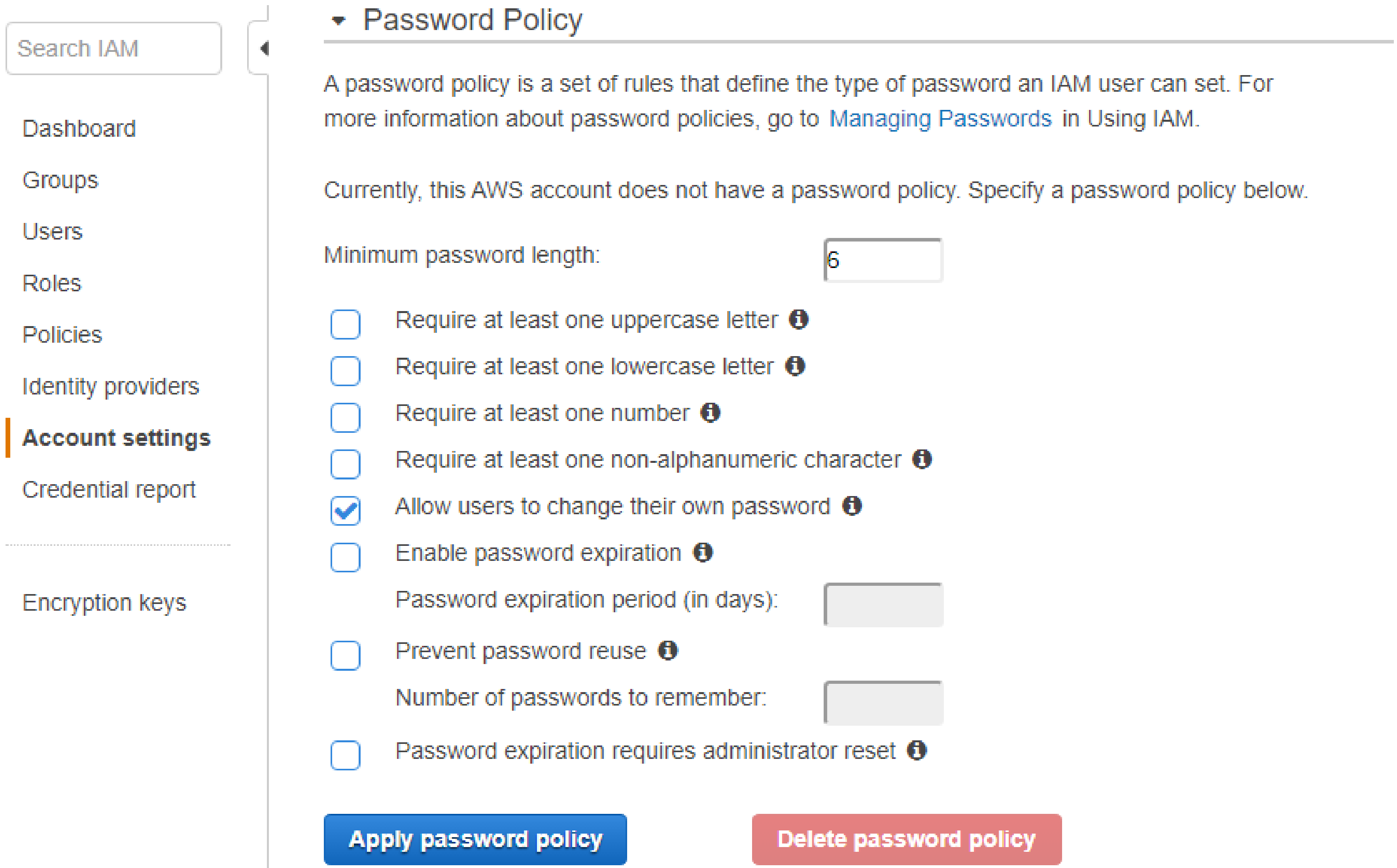
- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Best practices for setting up cross-account access
- Tutorials on roles

Prakash Kumar-DevOps Trainer



Password Policy

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to Managing Passwords in Using IAM.

Currently, this AWS account does not have a password policy. Specify a password policy below.

Minimum password length: 6

□ Require at least one uppercase letter

□ Require at least one lowercase letter

□ Require at least one number

□ Require at least one non-alphanumeric character

☑ Allow users to change their own password

□ Enable password expiration

Password expiration period (in days):

□ Prevent password reuse

Number of passwords to remember:

□ Password expiration requires administrator reset

AWS IAM Key Points

Key Points:

1. IAM is **universal**, (IAM is not specific to only one Region in AWS)

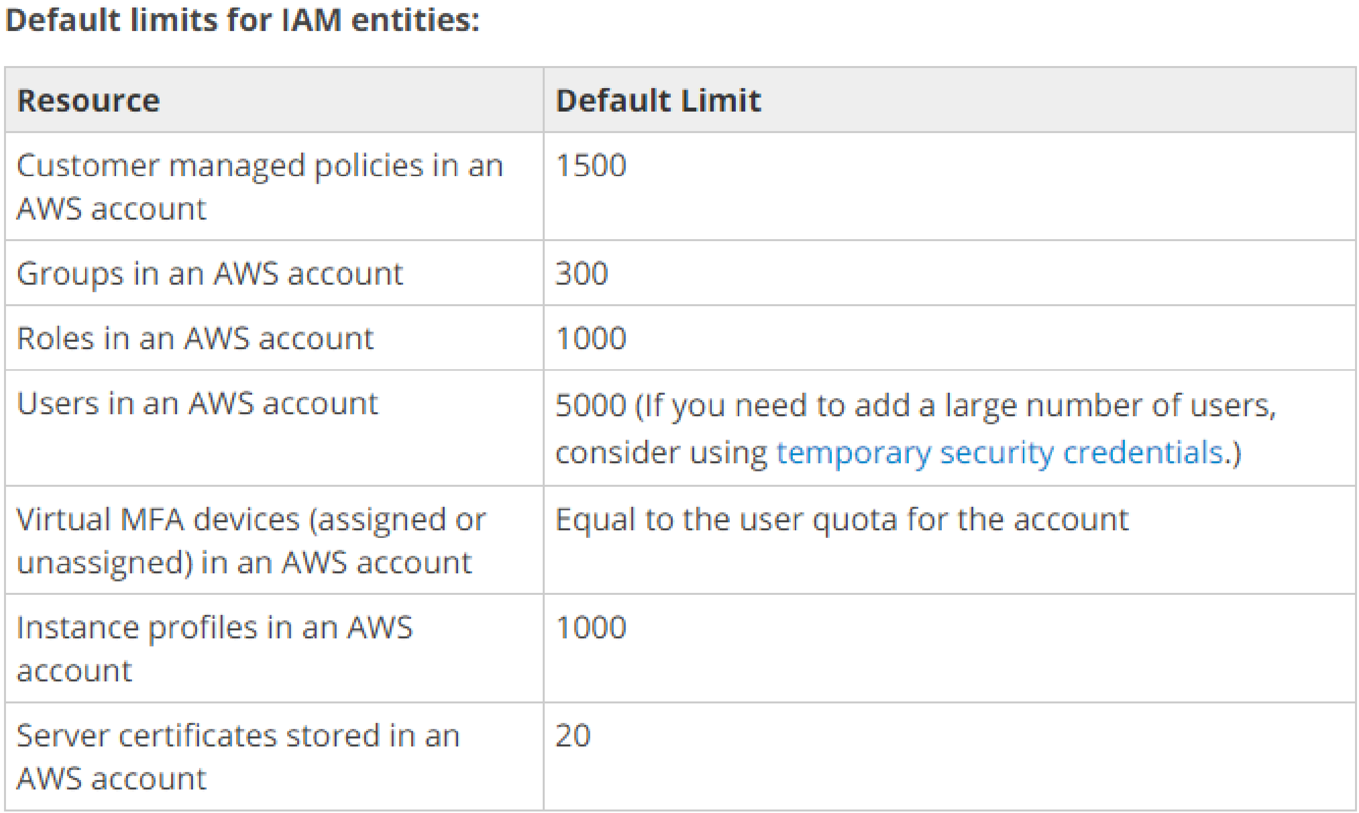
2. Root Account: The **Root** is simply the account who register or signup first time.

3. New users have no permission when first created.

4. New users are assigned **Access Key** & **Secret Key** when first created.

5. **Access Key** & **Secret Key** can be used to connect AWS services via AWS CLI, SDK etc

AWS IAM Limitation

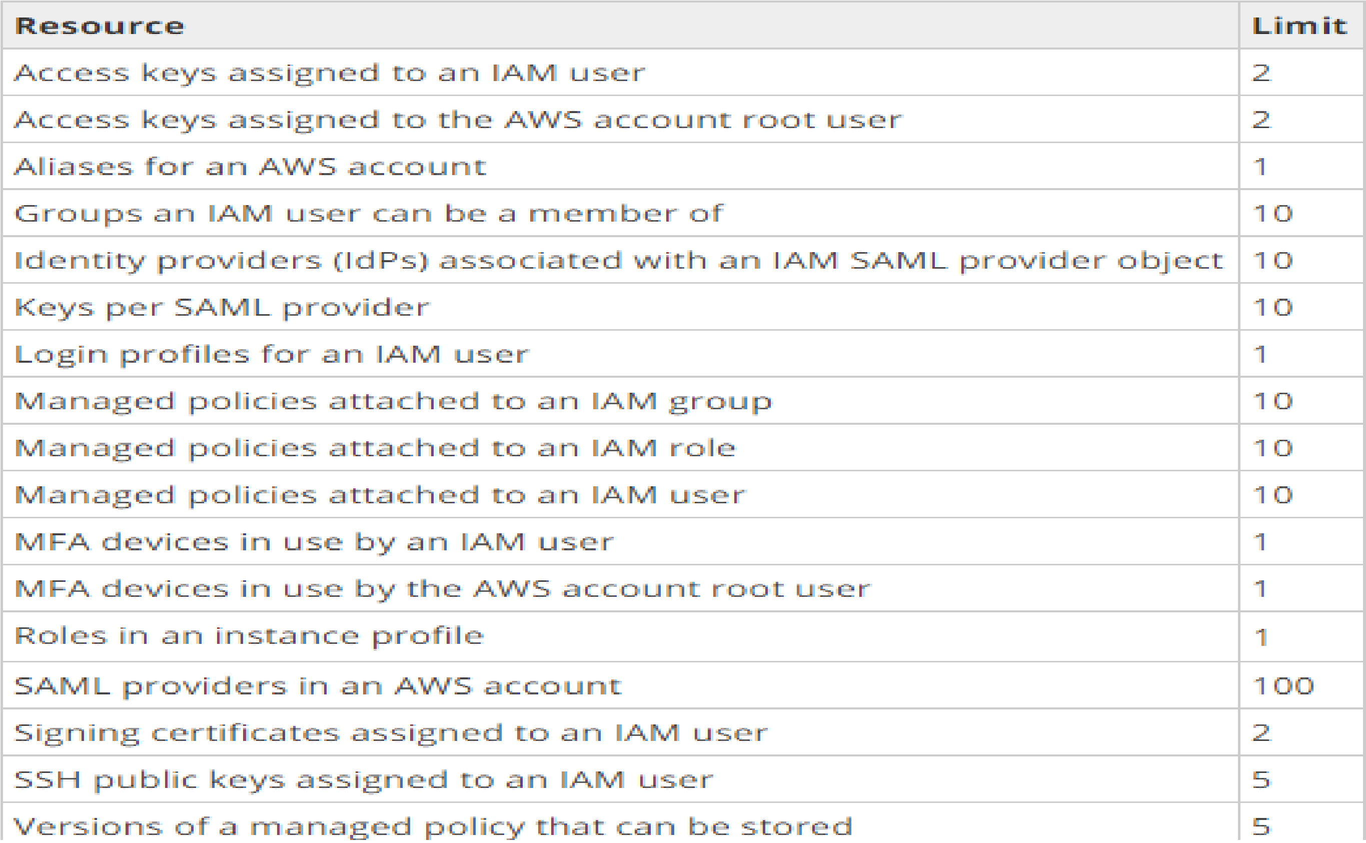


Default limits for IAM entities:

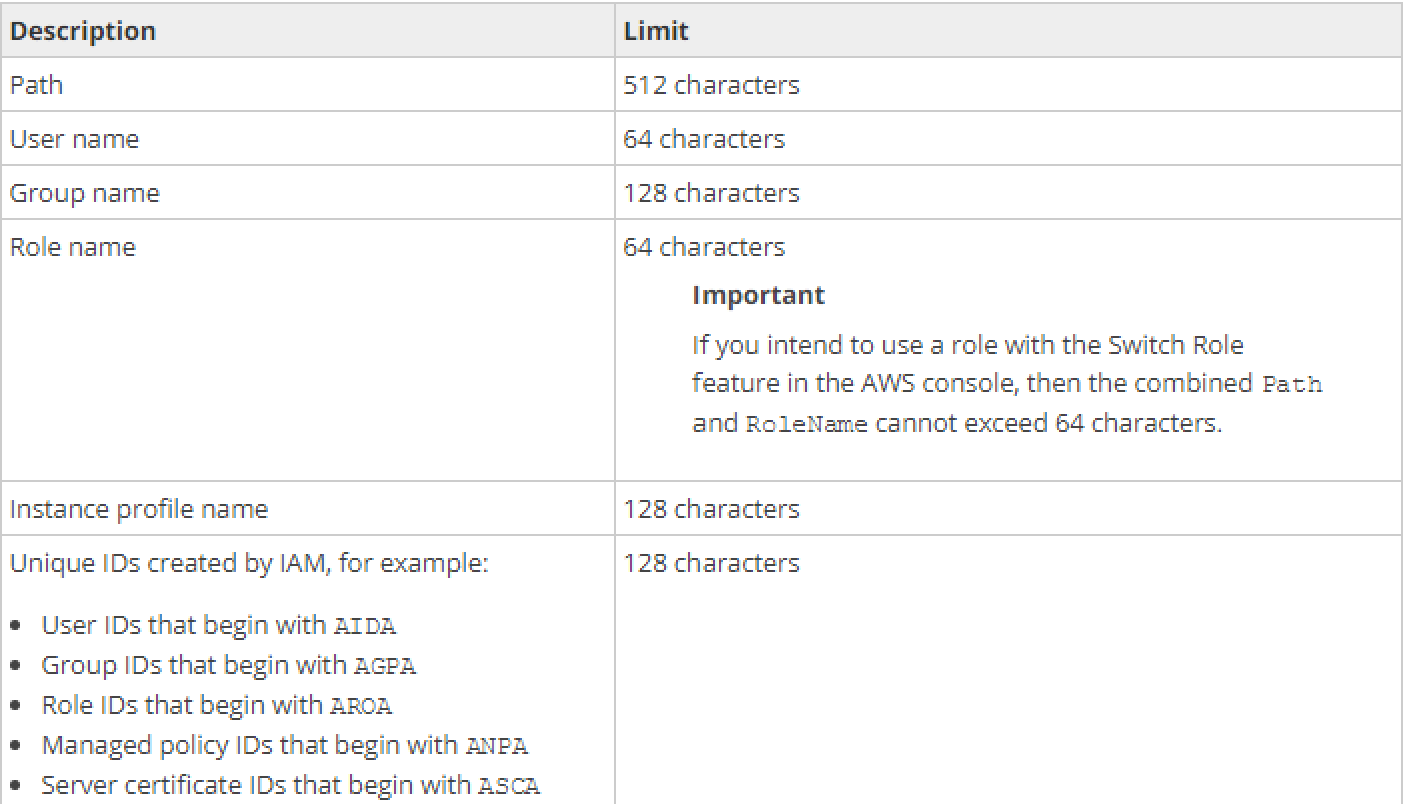
Resource	Default Limit
Customer managed policies in an AWS account	1500
Groups in an AWS account	300
Roles in an AWS account	1000
Users in an AWS account	5000 (If you need to add a large number of users, consider using temporary security credentials.)
Virtual MFA devices (assigned or unassigned) in an AWS account	Equal to the user quota for the account
Instance profiles in an AWS account	1000
Server certificates stored in an AWS account	20

For most IAM entity limits, you cannot request a limit increase:

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-limits.html



Resource	Limit
Access keys assigned to an IAM user	2
Access keys assigned to the AWS account root user	2
Aliases for an AWS account	1
Groups an IAM user can be a member of	10
Identity providers (IdPs) associated with an IAM SAML provider object	10
Keys per SAML provider	10
Login profiles for an IAM user	1
Managed policies attached to an IAM group	10
Managed policies attached to an IAM role	10
Managed policies attached to an IAM user	10
MFA devices in use by an IAM user	1
MFA devices in use by the AWS account root user	1
Roles in an instance profile	1
SAML providers in an AWS account	100
Signing certificates assigned to an IAM user	2
SSH public keys assigned to an IAM user	5
Versions of a managed policy that can be stored	5



Description	Limit
Path	512 characters
User name	64 characters
Group name	128 characters
Role name	64 characters **Important** If you intend to use a role with the Switch Role feature in the AWS console, then the combined `Path` and `RoleName` cannot exceed 64 characters.
Instance profile name	128 characters
Unique IDs created by IAM, for example: • User IDs that begin with `AIDA` • Group IDs that begin with `AGPA` • Role IDs that begin with `AROA` • Managed policy IDs that begin with `ANPA` • Server certificate IDs that begin with `ASCA`	128 characters

Managed Policies and Inline Policies

Using IAM, you apply permissions to IAM users, groups, and by creating policies. You can create two types of IAM, or*identity-based policies*:

1. Managed policies – Standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies apply only to identities (users, groups, and roles) - not resources. You can use two types of managed policies:

A. AWS managed policies – Managed policies that are created and managed by AWS. If you are new to using policies, we recommend that you start by using AWS managed policies.

B. Customer managed policies – Managed policies that you create and manage in your AWS account. Using customer managed policies, you have more precise control over your policies than when using AWS managed policies.

2. Inline policies – Policies that you create and manage, and that are *embedded* directly into a single user, group, or role. Resource-based policies are another form of inline policy.

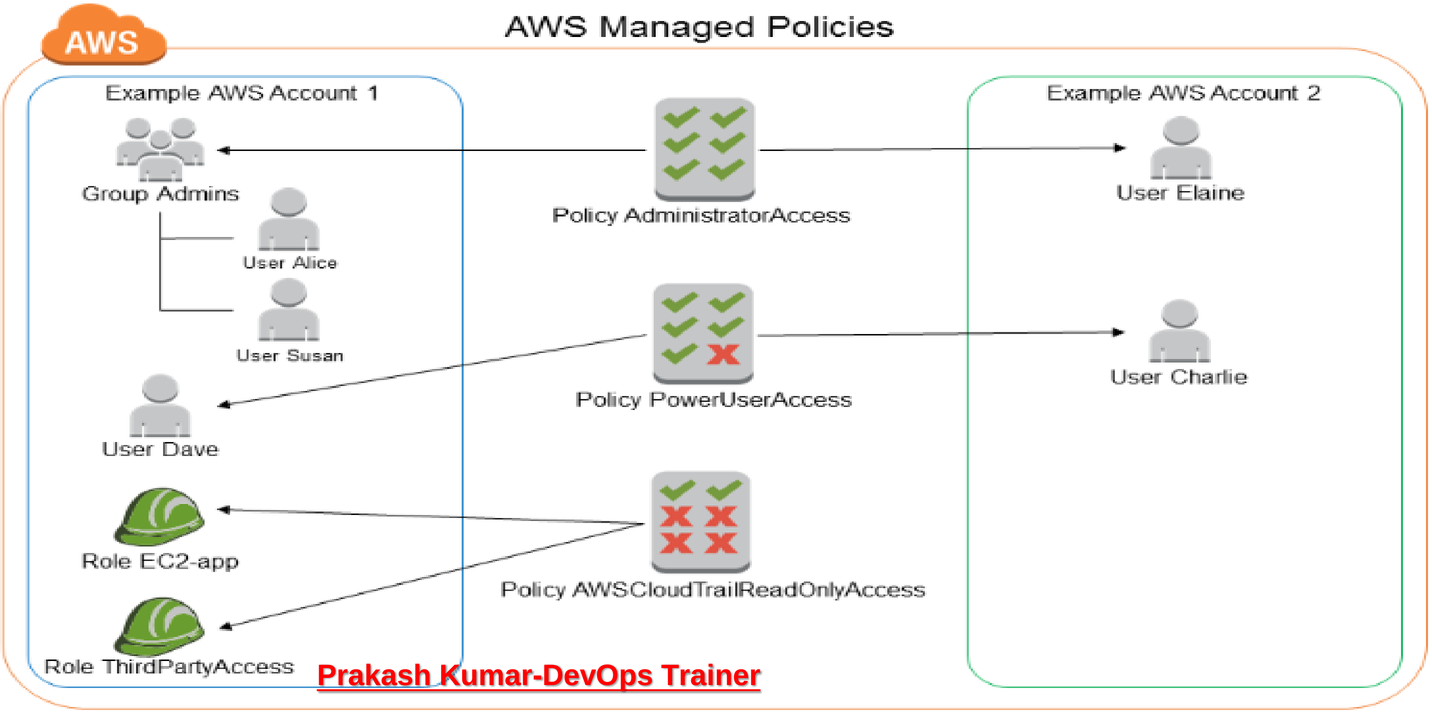
Identity-based (IAM) policies can be either inline or managed. Resource-based policies are attached to the resources (inline) only and are not managed.

AWS Managed Policies

An AWS managed policy is a standalone policy that is created and administered by AWS

When to use: AWS managed policies make it easier for you to assign appropriate permissions to users, groups, and roles than if you had to write the policies yourself.

Management: AWS is most likely to update an AWS managed policy when a new AWS service is launched or new APIs become available for existing services, and the policy needs to include permissions for the new service



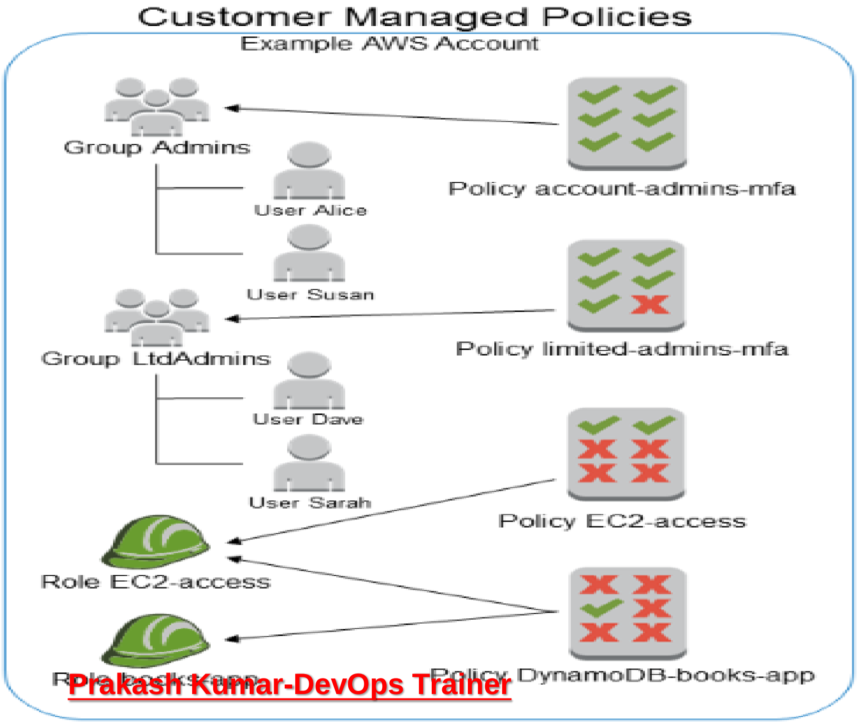
Prakash Kumar-DevOps Trainer

Customer Managed Policies

You can create standalone policies that you administer in your own AWS account, which we refer to as *customer managed policies*

When to use: More control

Management: Customer specific



Self Study

https://aws.amazon.com/iam/faqs/