Module 7. AWS Access Management Capabilities

This module covers the following subjects:

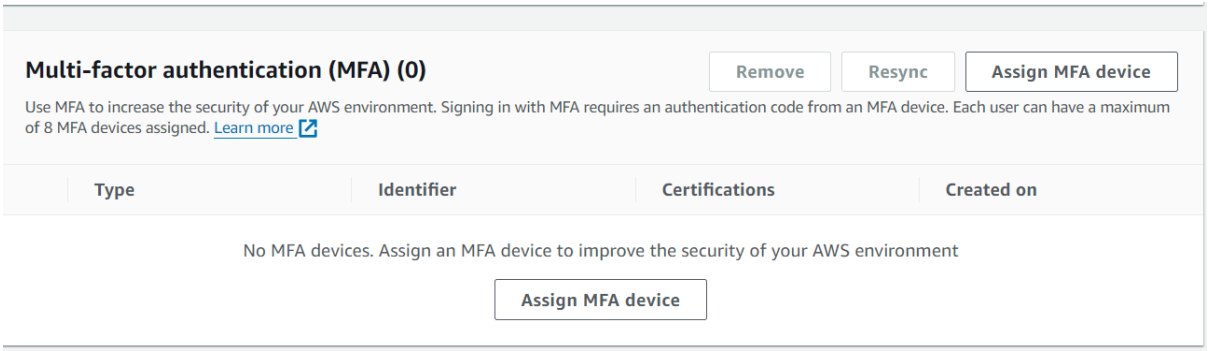
- **Identity and Access Management**: Where would your AWS architecture be without the ability to secure it? It would be in a very, very bad place. IAM is a crucial ingredient for AWS security, and this section of the module ensures you understand the components of IAM in AWS and how the parts work together to help secure your environment.
- **Best Practices with IAM**: While AWS makes IAM pretty simple, you should always follow the generally accepted best practices. This part of the module provides these best practices to you.

You need your users and your fellow engineers to be able to authenticate against AWS and then have their access strictly defined. AWS Identity and Access Management (IAM) is the primary tool for these responsibilities. In this module, get ready for a deep dive into IAM.

IDENTITY AND ACCESS MANAGEMENT

When it comes to accessing your account (the root account) and working inside of it, you need AWS's Identity and Access Management (IAM) services. IAM allows you to grant access to other individuals for team management of the services. IAM permits extremely granular permissions. For example, you might grant someone read access to only a single bucket of objects in S3. Other features of IAM include the following:

- **Access from service to service in AWS**: For example, an application running on an EC2 instance can access an S3 bucket. As you will learn later in this module, we often use roles for such access.
- **Multi-factor authentication (MFA)**: Access is permitted using a password and a code from an approved device, thus greatly strengthening security. Figure 7-1 shows the MFA configuration area in the IAM Management Console.



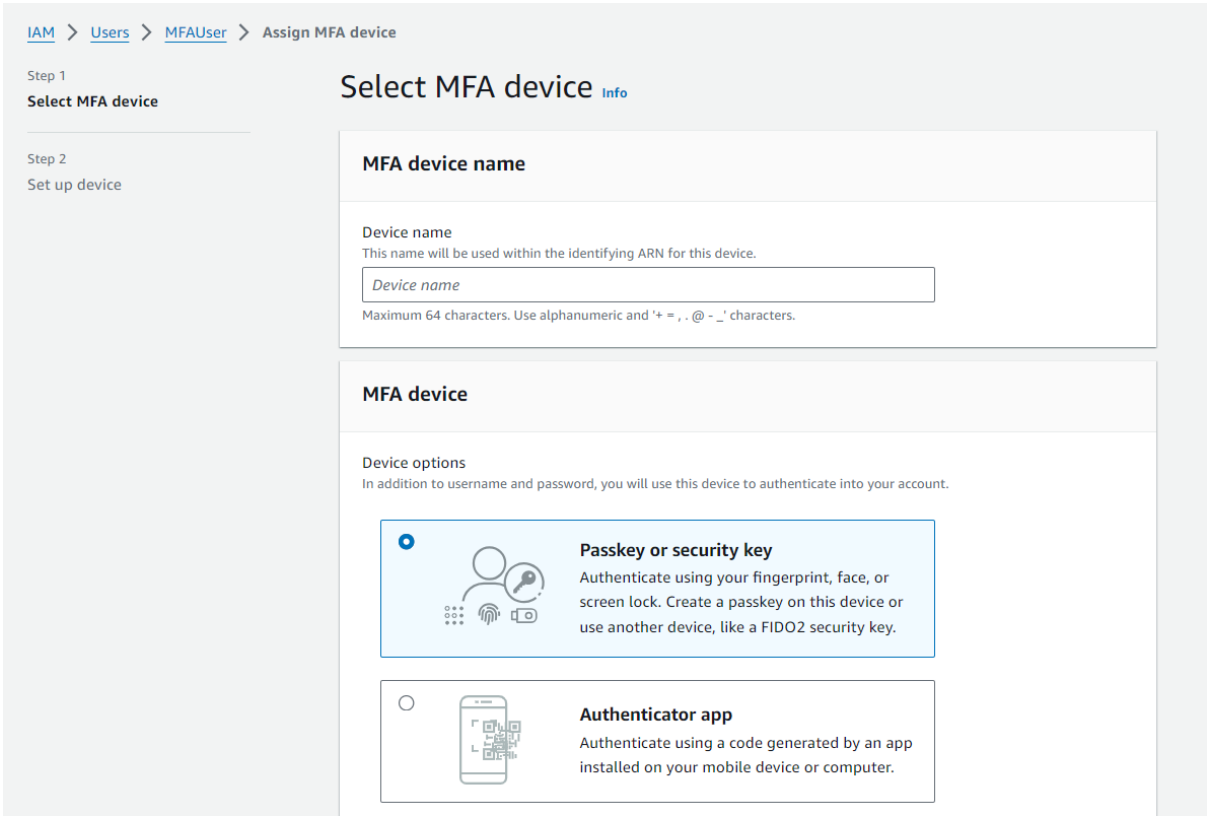


Figure 7-1 Configuring MFA for an Account

- **Identity federation**: Users who have already authenticated with another service can gain temporary access to your account's resources and services.
- **Identity information for assurance**: CloudTrail can trace and log all API activity against every service and resource in your account. Figure 7-2 shows the CloudTrail Dashboard in AWS.

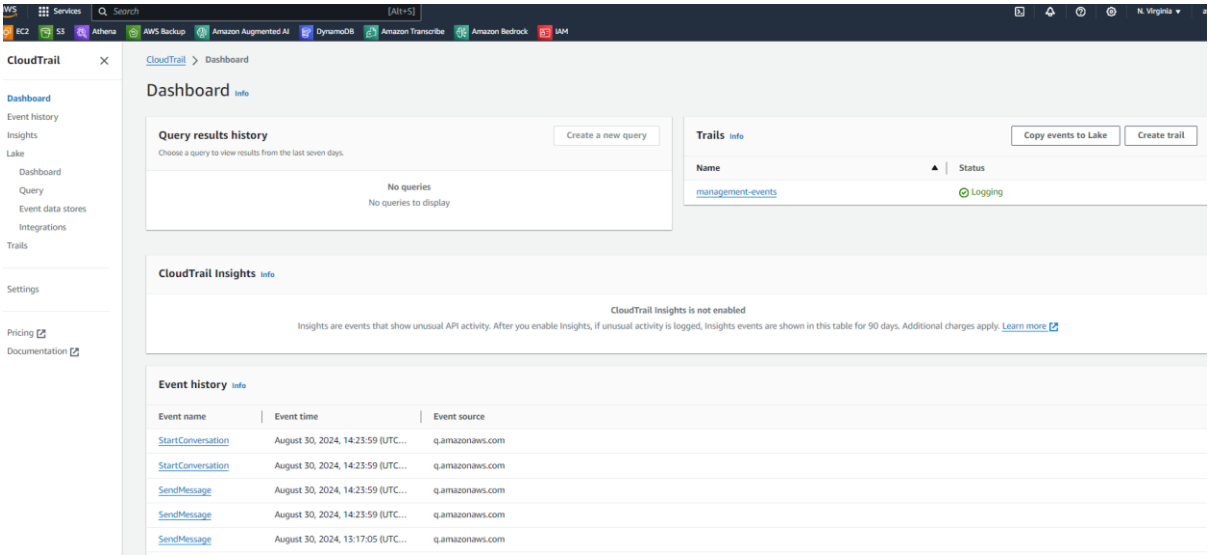


Figure 7-2 The CloudTrail Dashboard

- **PCI DSS compliance**: IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider and has been validated as compliant with the Payment Card Industry (PCI) Data Security Standard (DSS).

https://www.pcisecuritystandards.org/

- **Integration**: To be successful, IAM integrates with every primary service of AWS.
- **Eventually consistent**: Amazon replicates important data around the world with its Global Infrastructure to help ensure high availability (HA). As a result, data in some locations might lag others. Therefore, with IAM, consider implementing your changes for IAM first, then verifying full replication before working with dependent service deployments.
- **Always free**: While some AWS services can be used for one year (using the Free Tier account), IAM services remain free for the life of your account.
- **Accessibility options**: You can access the components of IAM in various ways, including the AWS Management Console, AWS command-line tools, AWS SDKs, and the IAM HTTPS API.

You must understand the primary identities you'll use in IAM. There is more to IAM than these identities, but we are covering the main foundational components at this point in your AWS education.

Identities consist of the following:

- **AWS account root user**: This is the account you established when you signed up for AWS; note that the user name for this account is the email address used for signup.
- **Users**: These entities you create in AWS represent the people or services that use the IAM user to interact with AWS. When you create an IAM user, you grant it permissions by making it a member of a group that has appropriate permission policies attached (recommended) or by directly attaching policies to the user. You can also clone the permissions of an existing IAM user, which automatically makes the new user a member of the same groups and attaches all the same policies. Figure 7-3 shows a user in AWS.

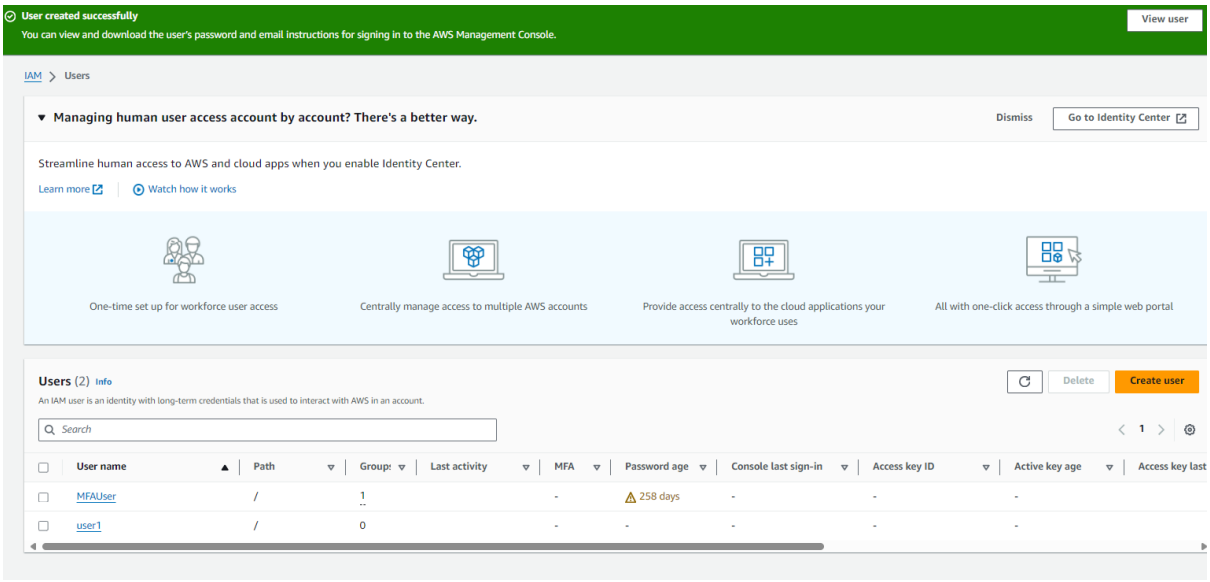


Figure 7-3 A User in AWS IAM

- **Groups**: A collection of IAM users. You can use groups to specify permissions for a collection of users, which can make those permissions easier to manage for those users.
- **Roles**: These are similar to user accounts, but they do not have any credentials (password or access keys) associated with them.

BEST PRACTICES WITH IAM

While IAM in AWS provides many exciting capabilities, its complexity can cause organisations to make fatal flaws when working with the service. This is why following best practices is critical.

You should consider most (if not all) of these recommendations.

- **Store root user access keys securely**: The root user account for your AWS implementation should be used infrequently. You must protect this account's access key ID and secret access key. You must ensure that you have these credentials protected in your infrastructure and treat them with the utmost care. In high-security environments, consider not defining access keys for the root account. Instead, use the email address, a complex password, and physical multi-factor authentication for the rare times this account must be used.

- **Create individual IAM users**: You must create additional user accounts because you do not want to use the root account for your AWS implementation. This would include for yourself so that you are not required to use the root account. In larger organisations, you will have a large team working on AWS. You must create multiple accounts for your staff to ensure that everyone authenticates and is authorised for only those resources and permissions required for each member to do their jobs. You will most likely have at least one account in IAM for every person who requires administrative access.
- **Use groups to assign permissions to IAM users**: If you are the sole administrator of your AWS implementation, you will want to create a group and assign permissions to this group. Why? If you need to grow and hire another administrator, add that user account to the group you created. We always want our AWS implementations to scale, and using groups helps ensure this. It should also be noted that applying permissions to groups instead of individual user accounts will also help eliminate assignment errors, as we minimise the amount of permissions we must grant.

-
- **Use AWS-defined policies for permissions**: Amazon was very kind to us. They defined many policies we can easily leverage when working with IAM. Moreover, AWS maintains and updates these policies as it introduces new services and API operations. The policies AWS created for us are defined around the most common tasks we need to perform. These make up an excellent starting place for your policies. You can copy a given policy and customise it to make it even more secure. Frequently, you will find the default-defined policies are too broad with access.
- **Grant least privilege**: Why might you end up with many different accounts for yourself in AWS? Well, you always want to sign in with the account that provides the least amount of privileges for what you are trying to accomplish. That way, if an attacker does manage to capture your security credentials and begins acting as you in the AWS architecture, they can do a limited amount of damage. For example, if you need to monitor the files in AWS S3 buckets, you can use an account with only read permissions on these buckets. This would certainly limit the damage an attacker can carry out.
- **Review IAM permissions**: You should not use a "set and forget" policy regarding your IAM permissions. You should consistently review the permission level assigned to ensure that you are following least privilege concepts and that you are still granting those permissions to the groups that require them. There is even a policy summary option within IAM to facilitate this.
- **Always configure a strong password policy for your users**: It is a sad fact of human nature. Your users will tend to be lazy about setting (and changing) their passwords. They will tend to use simple passwords that are easy to remember. Unfortunately, these simple passwords are also easy to crack. Help your security by setting a firm password policy that your users must adhere to. Figure 7-4 shows the configuration of a password policy for user accounts in the IAM Management Console.

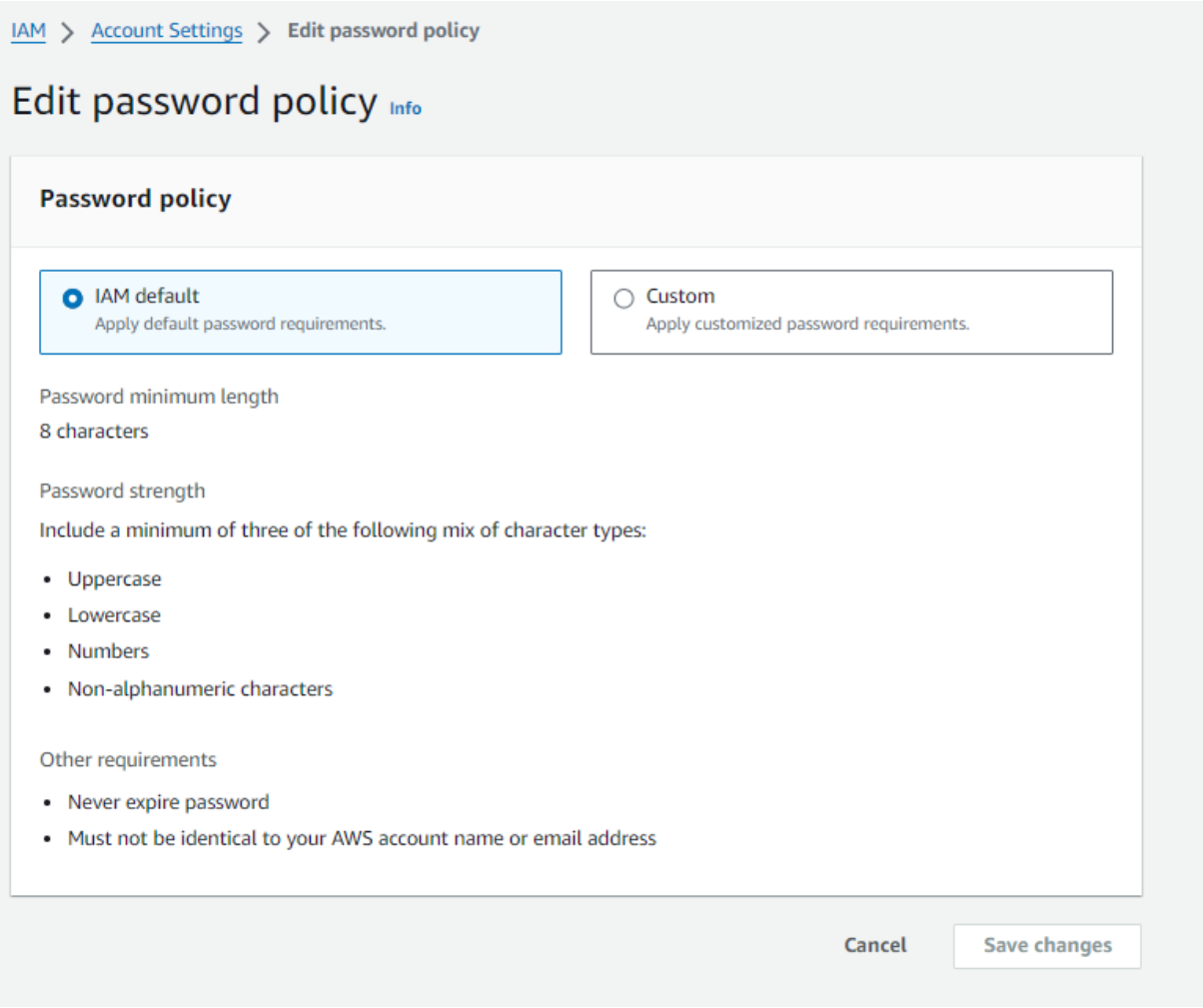


Figure 7-4 Configuring a Password Policy

- **Enable multi-factor authentication for privileged user accounts**: Of course, you do this for the seldom-used AWS root account, but you should also protect key admin accounts you have created in AWS. Using multi-factor authentication (MFA) ensures the user knows something (like a password) and also possesses something (like a smartphone). With most AWS environments today, MFA is considered mandatory.
- **Use roles**: You should consider using roles in AWS when your applications or services run on EC2 instances and need to access other services or resources.
- **Use roles to delegate permissions**: Roles can also prove very valuable when you need to permit one AWS account to access resources in another AWS account. This is a much more secure option to provide the other AWS account with username and password information for your account.

- **Do not share access keys**: It might be tempting to take the access keys that permit programmatic access to a service or resource and share those with another account that needs the same access. Resist this temptation. Remember, you can always create a role that encompasses the required access.
- **Rotate credentials**: Be sure to regularly change passwords and access keys in AWS. If these credentials are compromised, you will have minimised the damage that can be done when the stolen credentials no longer function!
- **Remove unnecessary credentials**: Because it is so easy to learn and test new features in AWS, it can get messy as far as IAM components you leave in place that are no longer needed. Be sure to routinely audit your resources for any "droppings" that are no longer required. AWS even assists in structuring reports around credentials that have not been recently used.
- **Use policy conditions**: Always consider building conditions in your security policies. For example, access might have to come from a select range of IP addresses, MFA might be required, or there can be time-of-day or day-of-week conditions.
- **Monitor, monitor, monitor**: AWS services provide the option for an intense amount of logging. Here are just some of the services where careful logging and analysis can dramatically improve security:
    - CloudFront
    - CloudTrail
    - CloudWatch
    - AWS Config
    - S3

EXAM PREPARATION TASKS

As mentioned in the section "How to Use This Book" in the Introduction, you have a few choices for exam preparation: the exercises here, Module 16, "Final Preparation," and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

- **REVIEW ALL TOPICS**
- **DEFINE ALL KEY TERMS AND CHECK ANSWERS IN THE GLOSSARY.**
- **DO THE QUIZ – REPEAT UNTIL YOU PASS IT (100% PASSMARK).**

DEFINE KEY TERMS

Define the following key terms from this module and check your answers in the Glossary:

IAM

MFA

federation

users

groups

roles

Q&A

1. What is often used when you need to provide access from an application running on an EC2 instance to other resources within AWS?

2. What account is created when you sign up for AWS but then should be used very sparingly after that point?