



# Prevention: Identity Management

## Security Fundamentals

Welcome to Security Lifecycle – Prevention: Identity Management.

# What you will learn

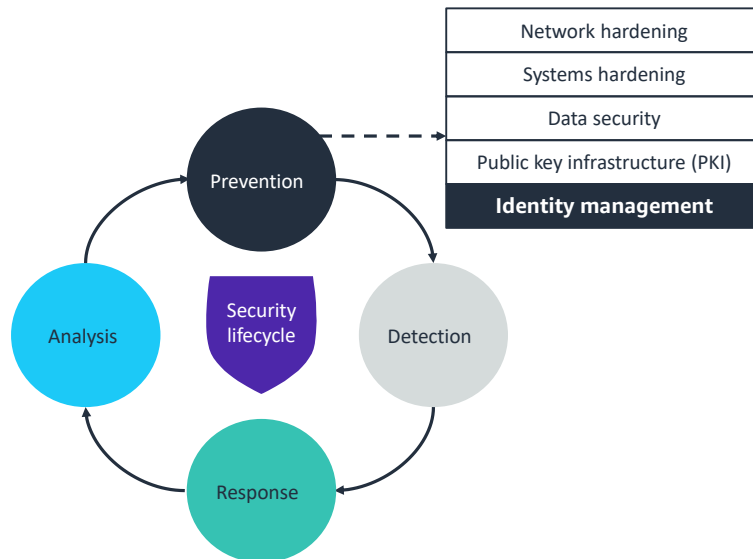
## At the core of the lesson

You will learn how to:

- Describe what identity management is and its different parts
- Explain how authentication works
- Describe different types of authentication factors
- Identify tools and services used to support identity management

In this lesson, you will learn how reliable authentication is a core component of modern IT security solutions. You will also explore how it works.

## Security lifecycle: Prevention – identity management



As a review, the security lifecycle consists of the following phases:

- **Prevention:** Serves as the first line of defense
- **Detection:** Occurs when prevention fails
- **Response:** Describes what you do when you detect a security threat
- **Analysis:** Completes the cycle as you identify lessons learned and implement new measures to prevent the issue from occurring again in the future

In this lesson, you will learn how you can use *identity management* concepts and methods in the prevention phase.

## Introduction to identity management

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In this section, you will learn about identity management and recall the confidentiality, integrity, and availability (CIA) triad.

## What is identity management?

- It is the **active administration** of subjects, objects, and their relationships **regarding access permissions**.
- It **ensures that identities** receive **appropriate access** to resources.
- It ensures that **systems remain scalable** in granting access to resources.

The following table provides an example of identity management within a retail store:

Role	Task	Access
Cashier	Receives and exchanges payment for a retail store	Access to cash register by using badge
Security	Guards the retail store against criminal activity	Access to front and back buildings by using badge
Manager	Supervises store operations and personnel and ensures quality control	Access to everything (cashier, buildings, and office) by using badge

Identity management is the active administration of subjects, objects, and their relationships regarding access permissions.

With identity management, users (or any type of identity) receive appropriate access to the resources that they need at the correct level and appropriate time. By implementing identity management, systems remain scalable as they identify and grant access to resources.

In the example on the slide, a cashier, security guard, and manager are working in a retail store. Their tasks are specific to their roles, and they each have specific access to certain areas or items depending on their roles.

# Identity management principles

**Authentication, authorization, and accounting (AAA) are the primary principles of identity management.**

- **Authentication** is concerned with proving and validating a user's or application's identity.
- **Authorization** is the process of determining what permissions the user and applications have.
- **Accounting** establishes auditing measures by logging access, commands, and changes that users and applications perform.



Authentication, authorization, and accounting are the core tasks that are performed in identity management.

As an example, consider the case of a visitor who tries to gain physical access to a company's facilities:

- **Identification:** The visitor must first prove that they are who they say they are by showing a picture identification to the receptionist.
- **Authentication:** The receptionist authenticates the visitor's identity by comparing the picture to the person who is standing in front of them.
- **Authorization:** To specify that the person is expected and should be allowed into the facilities, the receptionist calls the contact person to grant access to the visitor. The receptionist might also do the following actions:
  - Issue a visitor's badge
  - Require that the company contact escort the visitor into the facilities as proof to other employees of the visitor's authorization.
- **Accounting:** The visitor is required to sign a visitors' log. The visitor provides the following information in the log:
  - Name
  - Date and time they arrived and departed
  - Number of their visitor's badge
  - Name of their contact

- Reason for their visit

## AAA example login process

Identification, authentication, authorization, and accounting (IAAA) are common security steps that are enforced when a user logs into a system.

The following information is an example of a login principle:

A cloud engineer uses an identification card (ID) that their company issued to log in to the company computer. This card is used to **identify** the cloud engineer, and the engineer must be **authenticated** by inputting a password. After identification and authorization are complete, the engineer is granted access to the **authorized** services and folders that the engineer is allowed to work on. Every time the engineer logs in and out of the computer, **accounting** is taking place.

As an analogy, consider the case of a visitor who tries to gain physical access to a company's facilities. Recall the following:

- **Identification:** Something that the visitor must prove
- **Authentication:** Something that can be verified or validated
- **Authorization:** Specific access according to a role or level
- **Accounting:** A way to track the user or visitor with a log



## Authentication

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next, you discuss how authentication happens.

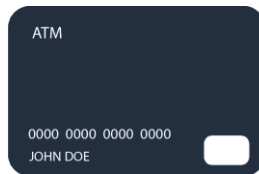
## Authentication factors

You can further control access by using multiple factors to authenticate or verify the identity of a user, process, or device.

Something you know

**Password**

Something you have



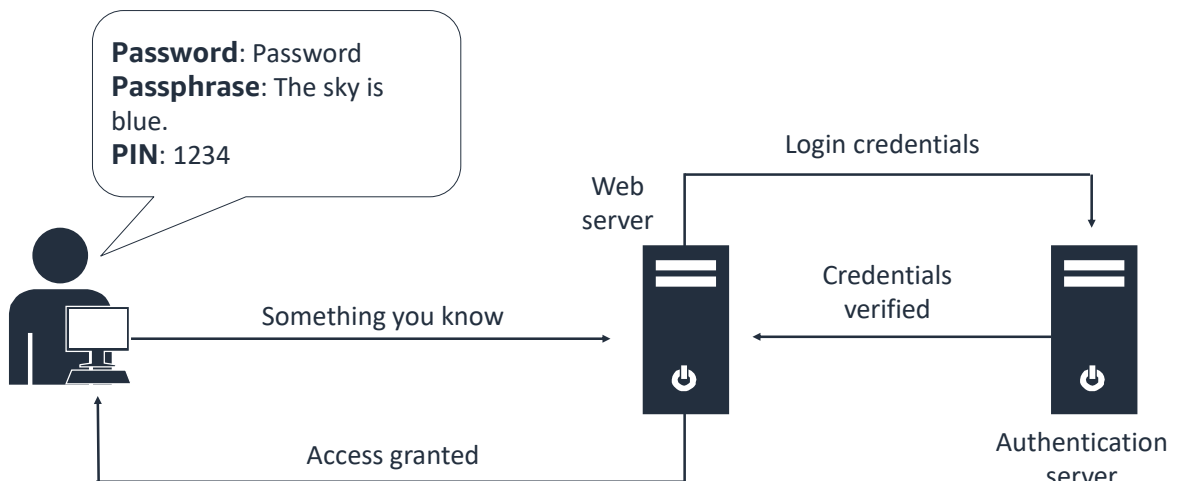
Something you are



The three types of authentication factors are something you know, something you have, and something you are. Every authenticator has one or more authentication factors.

**Multi-factor authentication (MFA)** is an authentication method that requires multiple methods or ways of authentication. For example, you log into a bank website with your password, and MFA is enabled. It might ask you for something that you have, such as your phone number. MFA is asking you for something that you know and something that you have.

## Authentication factors: Something you know



Passwords, passphrases, and personal identification numbers (PINs) are examples of authentication factors. These factors are simpler to implement, but they are also the least secure.

In the example on the slide, a diagram shows the authentication factor of something that you know. A user is inputting the following information:

- A password: Password
- A passphrase: The sky is blue
- PIN: 1234

By using this process, the web server can send the login credentials to the authentication server for verification. Then, it sends the information back to the web server, and access is granted.

## Authentication factors: Something you have

- Authenticate by using something that you physically possess.
- The following are some examples:
  - Smart card
  - Certificate
  - Token
  - USB key
  - Key
  - Virtual cards
  - Transaction Authentication Number (TAN)
  - Smart phone (push notification or authentication apps)
  - Two-factor authentication



RSASecureID



USB key



Physical key



Smart phone



Smart card

Using something that you have is a more secure way to authenticate. This method is often implemented as a second-factor authentication system after you have provided something that you know.

## Authentication factors: Something you are

- Biometric devices are used to validate something that you are.
- Biometric devices authenticate based on a human property.
- The following are some examples:
  - Fingerprint reader
  - Hand geometry
  - Retina scanner
  - Facial recognition
  - Iris recognition
  - Signature analysis

Fingerprint reader



Retinal scanner



Facial recognition

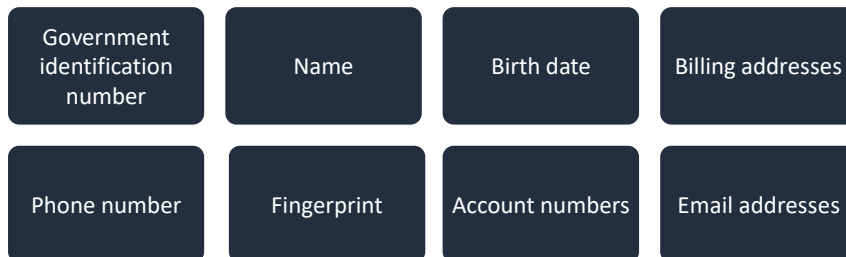


Using an authentication mechanism that validates a unique human property, such as a fingerprint or a retinal pattern, is the most complex and expensive solution. However, the authentication can be highly reliable when it is configured well.

Accuracy in biometrics is the most important factor in this type of authentication.

## Personally identifiable information (PII)

- Any data that can be used to uniquely identify an individual.
- The following are some examples:



*Personally identifiable information (PII)* is a type of data that, when used alone or with other relevant data, identifies an individual. PII might contain direct identifiers, such as passport information, race, ethnicity, or date of birth.

Even if individuals have the same name and possibly the same birth date, they would not have the same government identification number. In this way, PII can identify something unique about an individual among a group of people.

Why is PII important?

PII needs to be protected because it is something that only the individual can uniquely identify. Passwords can be cracked and systems can be hacked, but PII is something that cannot be cracked. Therefore, it is important to keep it in a safe place.

PII can fall under all three authentication factors:

- Something you know: Government identification number
- Something you have: Bank card
- Something you are: Fingerprint

## Authentication: Password policies

- Define rules that control how password are created and maintained.
- Why is a password policy important?
  - A weak policy (small number of characters, no complexity, and no password age) can be cracked in a matter of seconds.
  - Passwords are often the only thing that stands between the data and who is trying to access the data unless MFA is installed.
  - A strong policy ensures that passwords are complex and are changed within a certain time frame.
- A strong password is difficult for the system to compute but easy for humans to remember.

When you use password authentication, controlled password management is critical. The most basic way to manage password authentication is to define a policy with password parameters or rules.

A basic or common policy might have the following parameters:

- Minimum number of characters: for example, eight
- Password complexity: none or at least one capital letter
- Maximum password age: none

An example password for this policy is Password. This password and policy make this password extremely vulnerable to attacks such as dictionary and rainbow table attacks.

A good policy might have the following parameters:

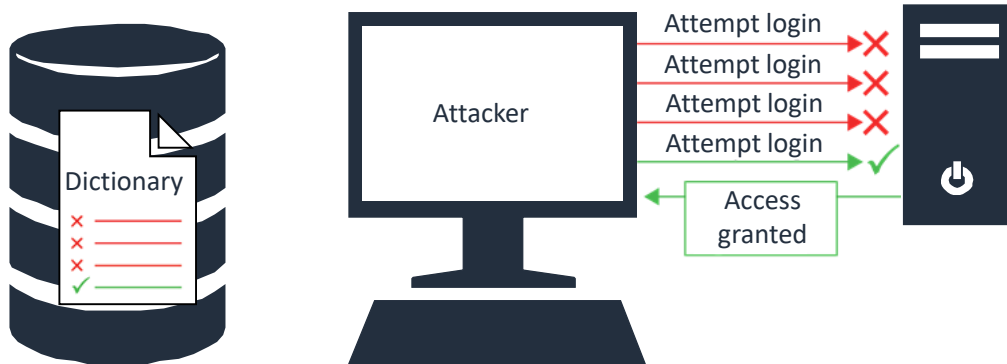
- Minimum number of character: Passwords must be at least 12 characters.
- Password complexity: Passwords must contain at least one capital letter, one number, and a symbol.
- Maximum password age: After 45 days, passwords expire, and the last 10 passwords cannot be used again.

An example password for this policy is Pa\$\$w0rd123!



## Dictionary attacks


A list of predefined words as passwords is used to attempt to log in to a system.



When you define password rules, you must understand the types of attacks that password authentication can be subject to. One type of attack is a *dictionary attack*. A dictionary attack attempts to systematically enter each word in the dictionary as a password until it finds a match. Countermeasures for dictionary attacks include enforcing a strong password policy and locking out access after a fixed number of unsuccessful attempts.

Another type of password authentication attack is a *rainbow table attack*, which uses precomputed hashes of text passwords. These precomputed hashes are compared against stolen hashes to find their corresponding password.

A *password hash* is a unique encrypted value. This password hash is produced by taking the value of the text password and transforming it by using an algorithm. The algorithm always produces the same hashed value for a given input value. The rainbow table lists the plain-text value of encrypted passwords specific to a given hash algorithm. Thus, the text password value of a password can easily be determined when a match on the hash value is found.



## Tools and services

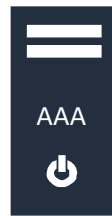
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next, you'll discuss tools and services.

## Password managers

- Operate over a centralized authentication system
- Improve security by requiring extra login steps
- Allow password resets
- Manage services that are used with specific credentials
- Store personal passwords on a local system

- Password consolidation
- Security questions
- Password reset
- Permitted services



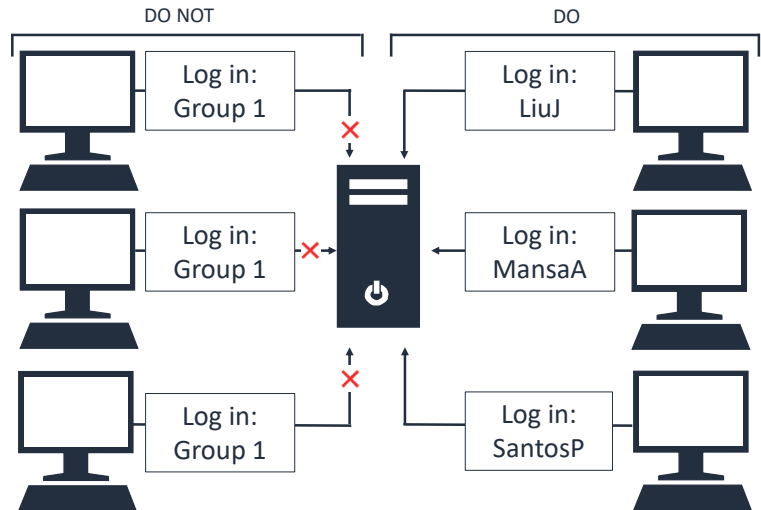
Web  
connection



One of the benefits of a password management system is that it gives users more control over managing their credentials.

## Group accounts

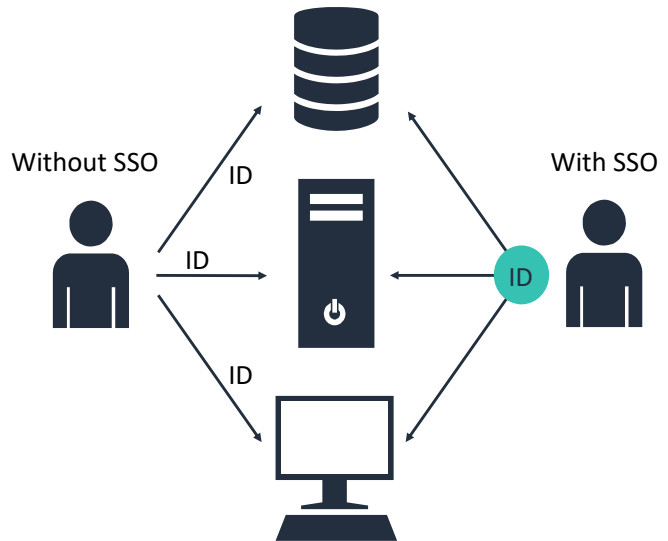
- Do *not* use group accounts.



To harden authentication, as a best practice, avoid the use of group accounts because they provide no accountability. With group accounts, multiple groups can authenticate.

## Single sign-on (SSO)

- Passwords are synchronized between two independent systems.
- Common, trusted login credentials are used across systems.
- Systems remain independent.
- Systems are not in a trust relationship and do not belong to the same directory structure.



With single sign-on (SSO), users log in once and gain access to different applications without the need to re-enter login credentials for each application.

# AWS Single Sign-On

## What is AWS SSO?

- A cloud-based service that you can use to centrally manage SSO access to all Amazon Web Services (AWS) accounts, including user permissions and AWS Organizations

## AWS SSO includes the following common features:

- One-click access to AWS accounts and cloud applications
- Ability to create and manage users and groups
- Compatibility with common cloud applications
- Compatibility with existing AWS Identity and Access Management (IAM) roles, users, and policies

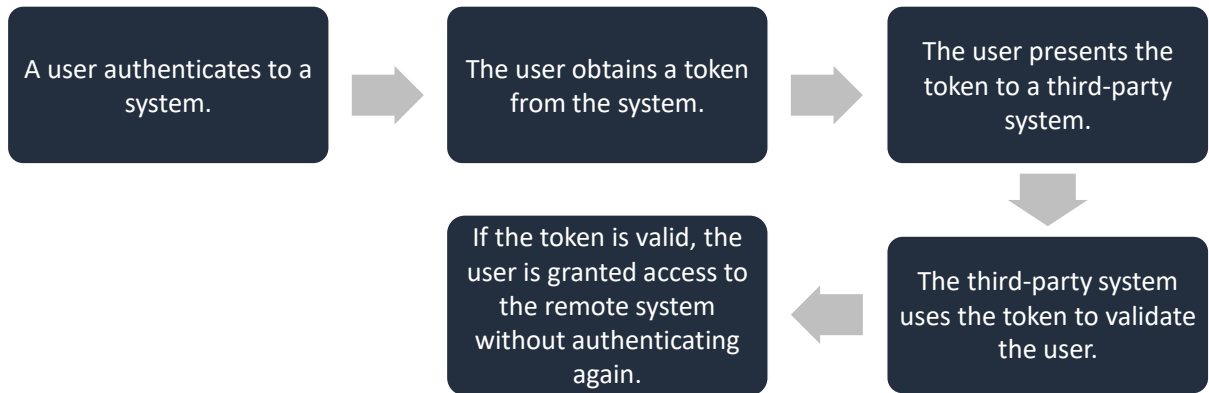
## AWS SSO includes the following features:

- One-click SSO access to AWS accounts: AWS SSO has integrated applications and custom SAML 2.0 based applications, which gives the AWS SSO console quick one-click access to authorized users.
- Ability to create and manage users and groups with AWS SSO: You can create users and groups within AWS SSO in the console for easy access to granting permissions. AWS SSO is also compatible with the AWS Directory Service for Microsoft Active Directory.
- Compatibility with common cloud applications: Because AWS SSO is compatible with commonly used cloud applications, cloud administrators don't need to learn how to administer SSO services to different applications. AWS SSO can be integrated with applications that are noted in the AWS documentation.
- Compatibility with existing IAM roles, users, and policies: Using AWS SSO will have no impact on existing IAM roles, users, or policies.

For more information, see *What is AWS Single Sign-On?* at <https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html>.

## Federated users

- *Federated users* is a form of single sign-on.
- One account is used for multiple services.



*Federated users* is a type of SSO implementation that is used between web identities. It uses a token to verify user identity between distinct systems.

With SSO, individuals can sign into different networks or services by using the same group or personal credentials. For example, by using SSO, you can use your Google account credentials to sign into Facebook.

# Amazon Cognito

What is Amazon Cognito?

- It is an Amazon service that provides user management, authentication, and authorization for your web and mobile apps.
- You can use Amazon Cognito by signing in with a user name and password through a third-party website.

Amazon Cognito includes two main components:

- User pools provide sign-up and sign-in options for app users.
- Identity pools grant users access to AWS services.

Here is how Amazon Cognito works to authenticate and grant access to an AWS service:

1. First, the user signs in through what is called a user pool and receives a user token after they get authenticated.
2. The application will then exchange the user pool token for an AWS credential through the identity pool.
3. For the final step, the user can now use the AWS credentials to access AWS services.

User pools have the following characteristics:

- Can federate through a third-party identity provider
- Function as a user directory for Amazon Cognito
- Give users the ability to sign up or sign in through a web or mobile app by using Amazon Cognito
- Provide SSO, customizable user interface (UI), social sign-in options, user directory management, and MFA

Identity pools have the following characteristics:

- Support anonymous guest users
- Give users the ability to obtain temporary AWS credentials to access AWS services
- Provides SSO, social sign-in, OpenID Connect (OIDC) providers, SAML identity providers, and developer-authenticated identities

For more information, see *What is Amazon Cognito?* at

<https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon->

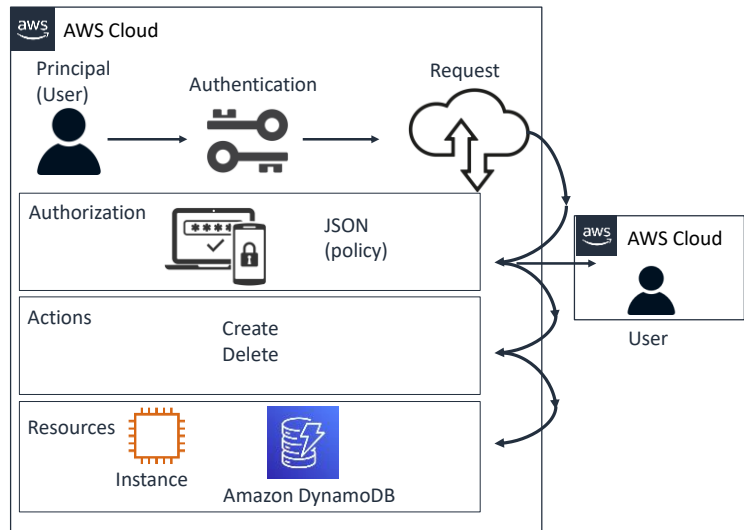


[cognito.html](#).

## How IAM works

AWS Identity and Access Management (IAM) is a service that helps you securely control access to AWS resources by using authentication and authorization.

- Authentication: Use IAM to control who can sign in
- Authorization: Use IAM to control who has access to resources



The diagram on the slide illustrates how IAM works. When the user signs in to the AWS console, the principal is authenticated as the AWS account root user or an IAM entity (user or role). The principal then sends a request to AWS, which contains information about the principal itself, the action it wants to perform, environment data, and resources. After the request is sent, AWS uses policies stored as JavaScript Object Notation (JSON) documents to check whether the user is authorized for the request. As soon as the user is authenticated and authorized and AWS approves the request, the user is allowed to create, delete, and edit resources. The policy defines the actions that the principal is allowed to take on the resources.

A principal is a person that is either the AWS account root user or IAM user or role that makes requests to AWS.

Authentication happens when the principal signs in to AWS by using their AWS credentials.

A request contains the following log of information:

Resources: The AWS resource object being accessed

Principal: The (user or role) sending the request

Environment data: IP address, user agent, enabled status, time of day, and user agent

Resource data: Data that pertains to the resource (for example, a tag on an Amazon Elastic Compute Cloud [Amazon EC2] instance)

Authorization is stored in AWS as JSON documents, which AWS uses as values to

check for policies that apply to these requests. These requests are either allowed or denied.

Action is when the user is authenticated, authorized, and request approved, and the user will then be allowed to create, delete, and edit resources. A policy holds what the principal or user is allowed to create, delete, or edit.

## Checkpoint questions

What are the three primary tasks of identity management?

What are the three types of authentication factors?

Federated users are a form of single sign-on. True or False?

1. The following are the three primary tasks of identity management:

- Authentication
- Authorization
- Accounting

2. The following are the three types of authentication factors:

- Something you know
- Something you have
- Something you are

3. Federated users are a form of single sign-on. True

## Key takeaways



- Identity management ensures that users receive **the appropriate access** to the resources they need, at **the right level**, and at **the appropriate time**.
- **Authentication factors** can be categorized as the following:
  - **Something you know**: For example, a password
  - **Something you have**: For example, a smart card
  - **Something you are**: For example, your fingerprint
- A good **identity management** solution includes creating **password policies**, using **password managers**, and using **single sign-on** and **federated identity management**.
- AWS Identity and Access Management (IAM) is a service that helps you control access to AWS resources in a secure way by using authentication and authorization.

The following are some key takeaways from this lesson:

- Identity management ensures that users receive the appropriate access to the resources that they need, at the right level, and at the appropriate time.
- Authentication factors can be categorized as the following:
  - Something you know: For example, a password
  - Something you have: For example, a smart card
  - Something you are: For example, your fingerprint
- A good identity management solution includes creating password policies, using password managers, and appropriately using single sign-on and federated identity management.
- AWS Identity and Access Management (IAM) is a service that helps you control access to AWS resources in a secure way by using authentication and authorization.



# Thank you



© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.

Thank you for completing this module.