

Identity and Access Management (IAM)



Welcome to the third lesson of the AWS Solutions Architect Associate level course— Identity and Access Management (IAM).

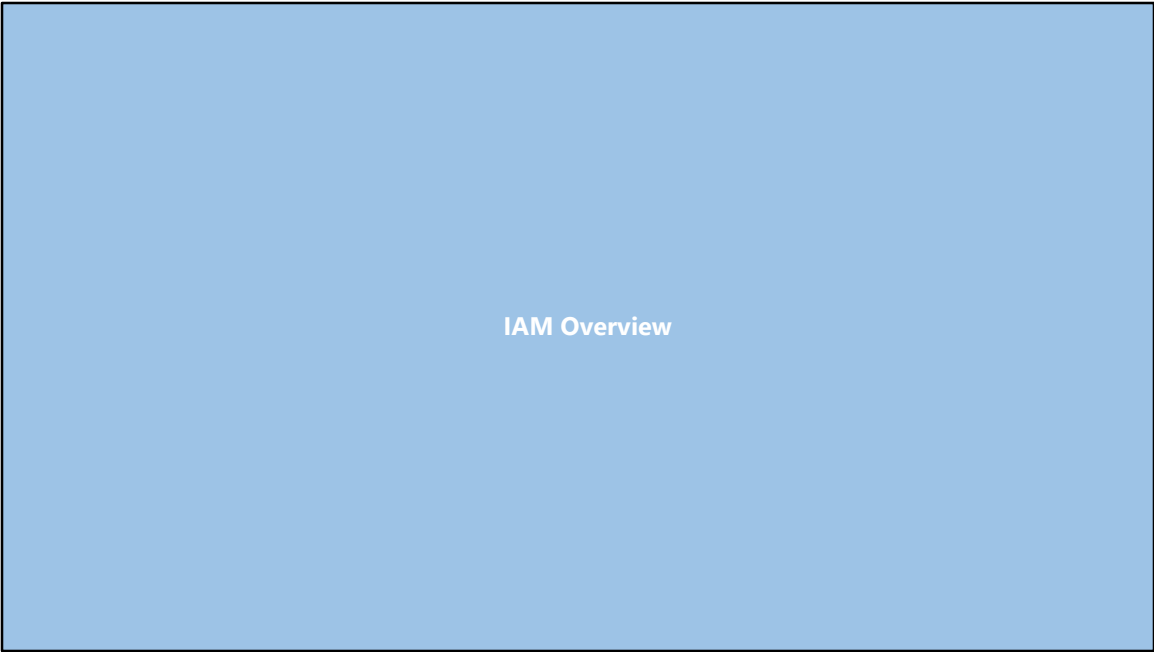
Learning Objectives

By the end of the lesson you will be able to:

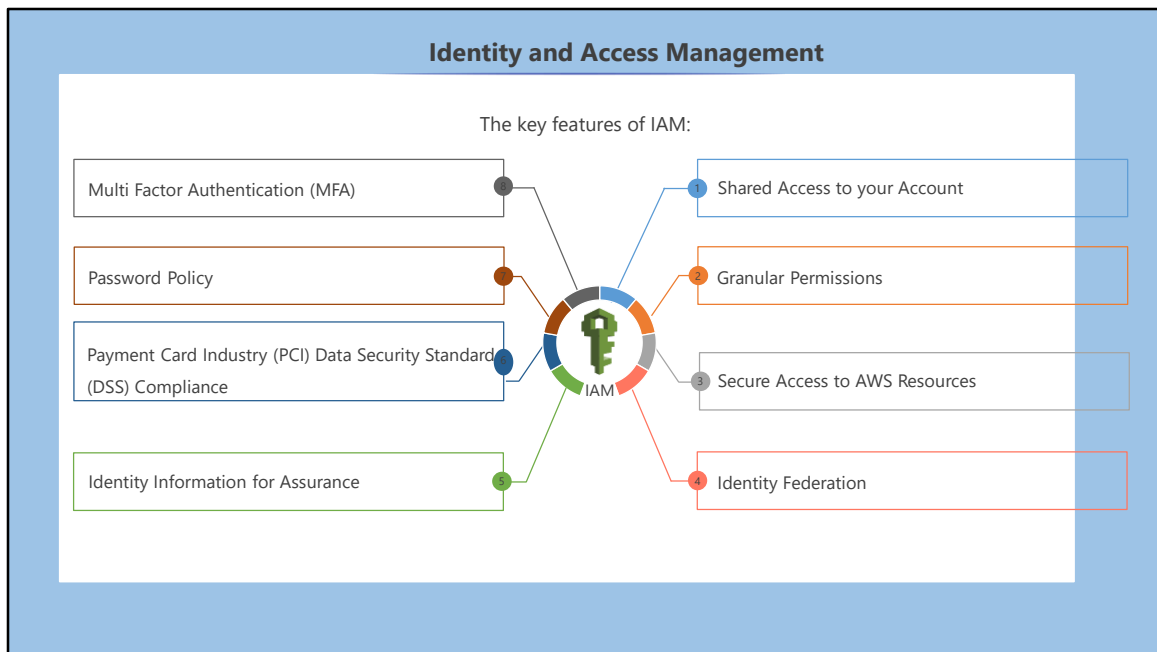
- ☐ Describe the key features of IAM and how they can simplify and secure user access to AWS
- ☐ Define permission to AWS users using AWS policies
- ☐ Explain the steps to create users in AWS
- ☐ Describe how groups simplify IAM management
- ☐ Use roles to delegate access to the AWS resources
- ☐ List the best practices for IAM

By the end of this lesson, you'll be able to:

- Describe the key features of IAM and how they can simplify and secure user access to AWS
- Use the AWS policies to define permission to AWS users
- Explain the steps to create users in AWS
- Describe how groups simplify IAM management
- Use roles to delegate access to the AWS resources
- List the best practices for IAM



Let's start with an overview of AWS IAM.



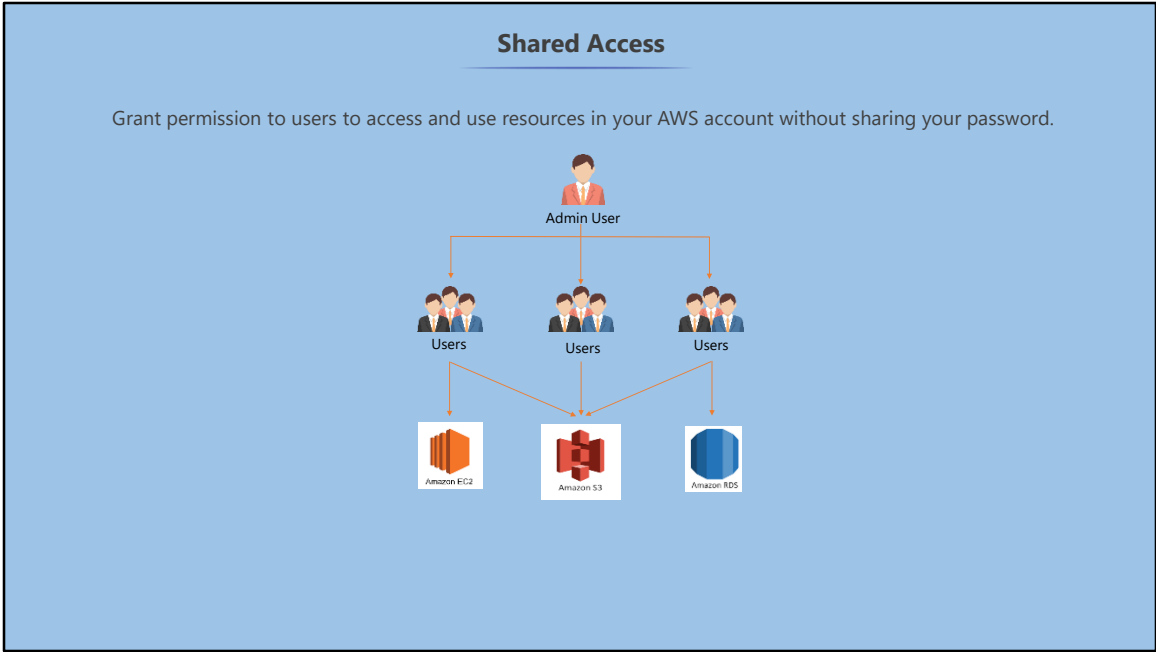
AWS Identity and Access Management or IAM is the service that enables you to securely control user access to all AWS services and resources.

It's based around the concepts of user management, such as users, groups and permissions.

In this session we will go through the key features of IAM and then cover them in detail in later sections.

Following are the key features of IAM:

- It gives shared access to your account.
- It provides granular permissions.
- It gives secure access to AWS resources.
- It provides Identity Federation.
- It provides Identity Information for Assurance.
- It provides PCI DSS Compliance.
- It allows you to set your password policy.
- It provides the option of Multi Factor Authentication, or MFA.

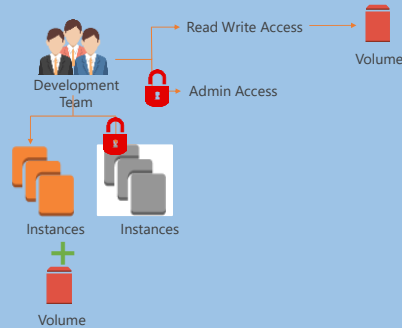


With Shared access, you can grant people in your organization permission to administer and use the resources in your AWS account without the need to share your password or access key.
Example: You can grant permission to users to access specific areas or instances of AWS.

Granular Permissions

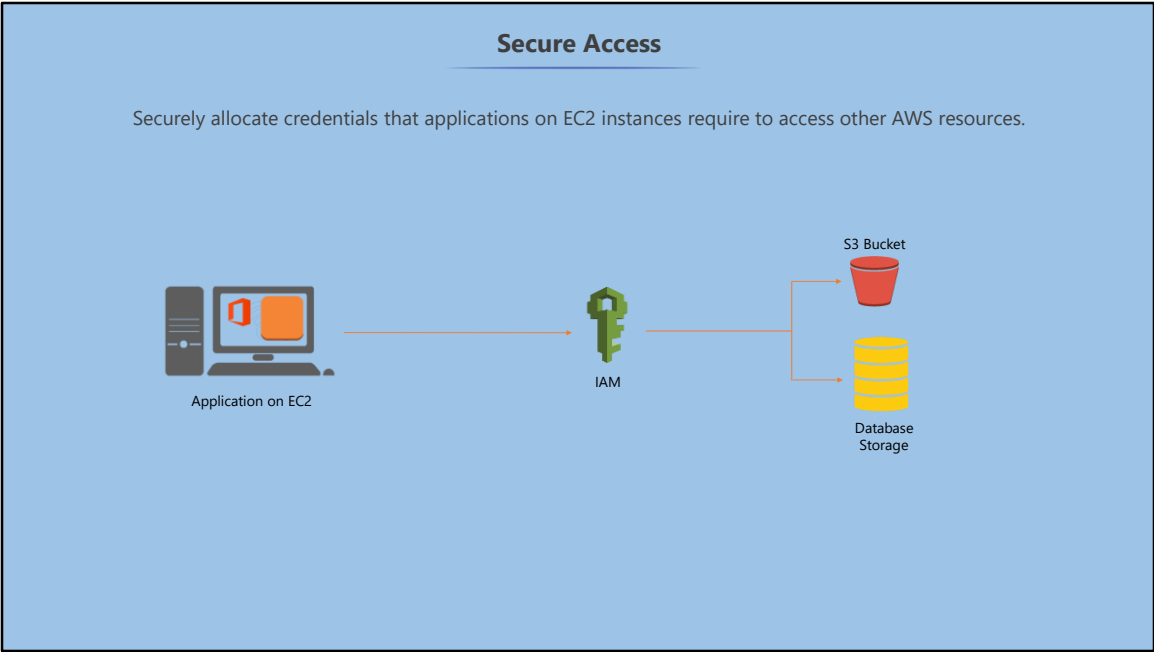
Granular permissions allow different permissions to various users to manage their access to AWS, such as:

- User access to specific services
- Specific permissions for actions
- Specific access to resources



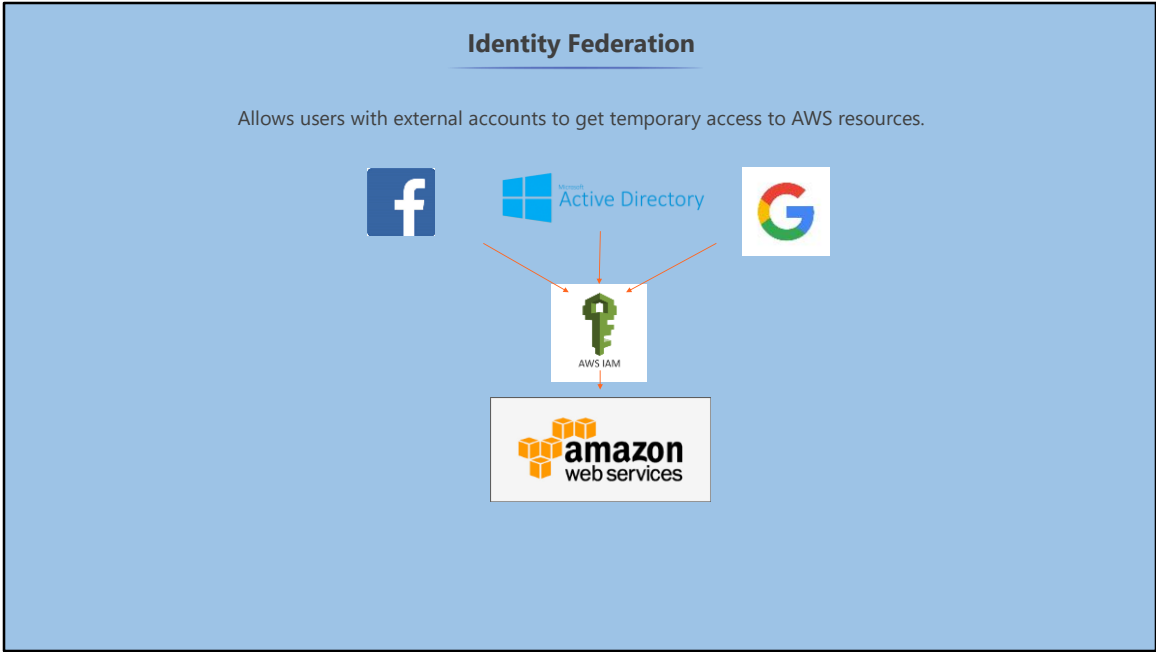
Granular permissions allow you to grant different permissions to different people; this helps you manage their access to AWS. Let's discuss this in some detail.

- You should grant users access to specific services. For example, the development team can access the development EC2 instances but not the production instances.
- You should grant specific permissions for actions. For example, the development team can have read–write access to all volumes but not administrator access.
- You should grant specific access to resources. For example, the development team can add new storage volumes to the development instances, but they aren't allowed to launch new instances.

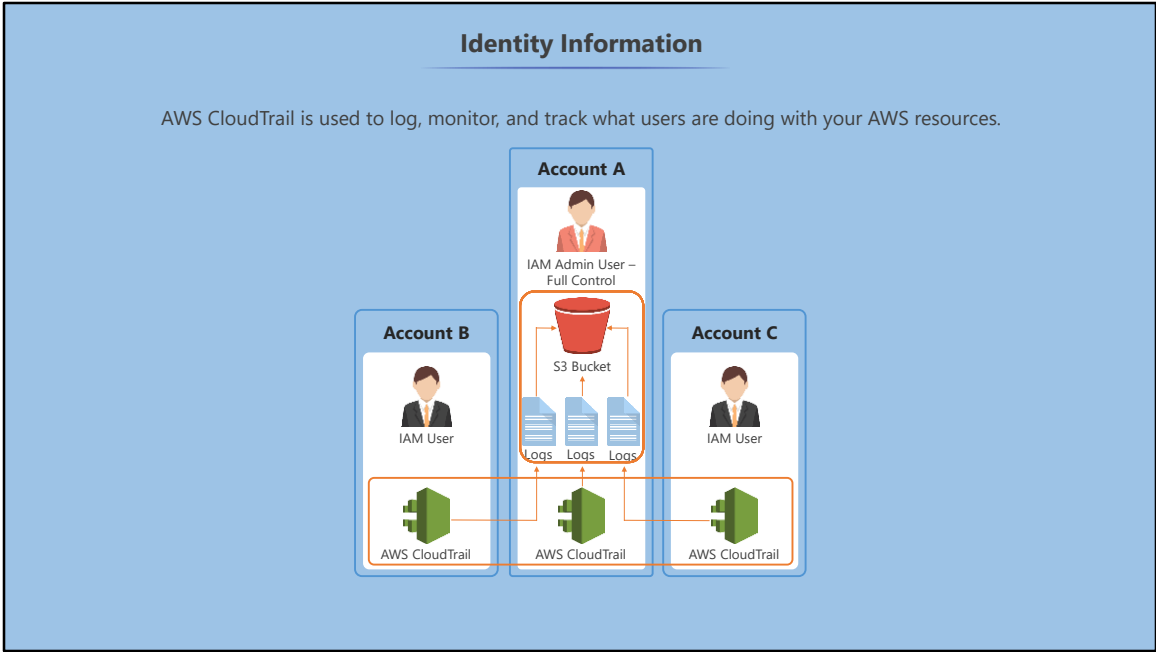


IAM allows you to securely allocate credentials that applications on EC2 instances require to access other AWS resources, for example, S3 buckets, databases or EBS volumes.

With proper configuration, IAM allows applications that access an EC2 instance to automatically inherit the permissions required to access the S3 buckets without the need to store user credentials.



Identity Federation allows users who have passwords elsewhere, such as Microsoft Active Directory or external web identity providers like Google and Facebook, to get temporary access to your AWS account.



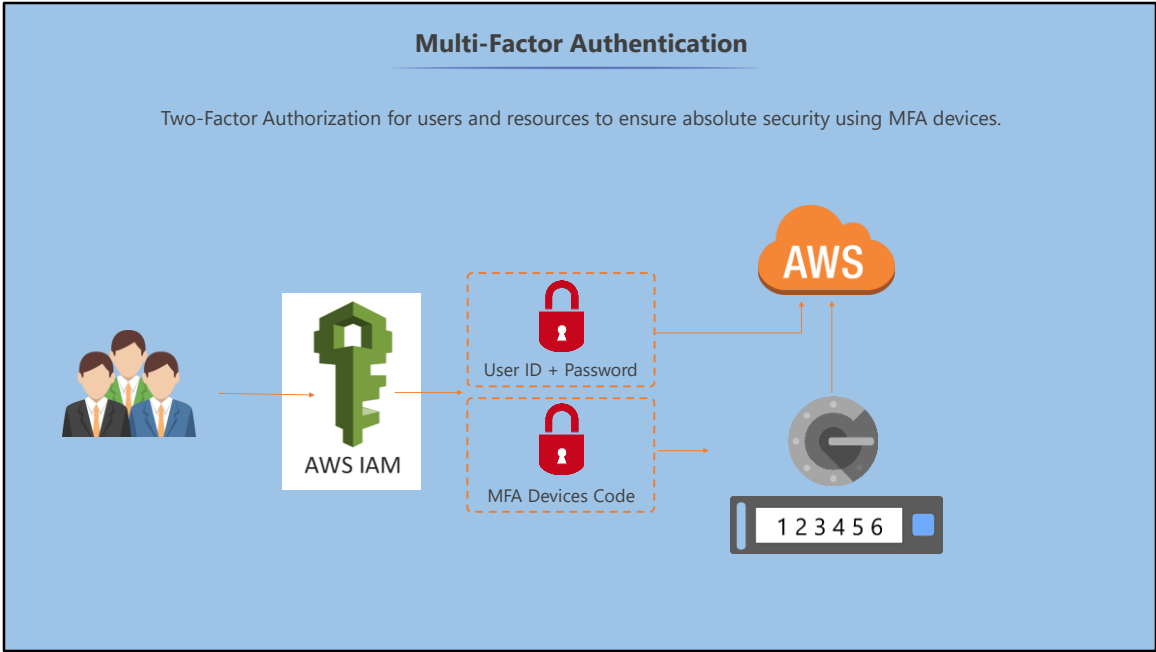
AWS CloudTrail is used to log, monitor, and track what users are doing with your AWS resources. The logs are stored in S3 buckets that allow complete administration control.

PCI DSS Compliance

Payment Card Industry (PCI) and Data Security Standard (DSS) compliant



IAM is Payment Card Industry (PCI) and Data Security Standard (DSS) compliant, so you can process, store, and transmit credit card data from a merchant or service provider.



IAM allows you to configure Two-Factor Authorization for users and resources to ensure absolute security using MFA devices.

You can use a six-digit code from a token to get access or use MFA applications on your mobile phone, such as Google Authenticator.

Password Policy

IAM allows you to define password strength and rotation policies.

Password: [masked]

Password strength: **Weak**

Password: [masked]

Password strength: **Strong**

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 ⓘ

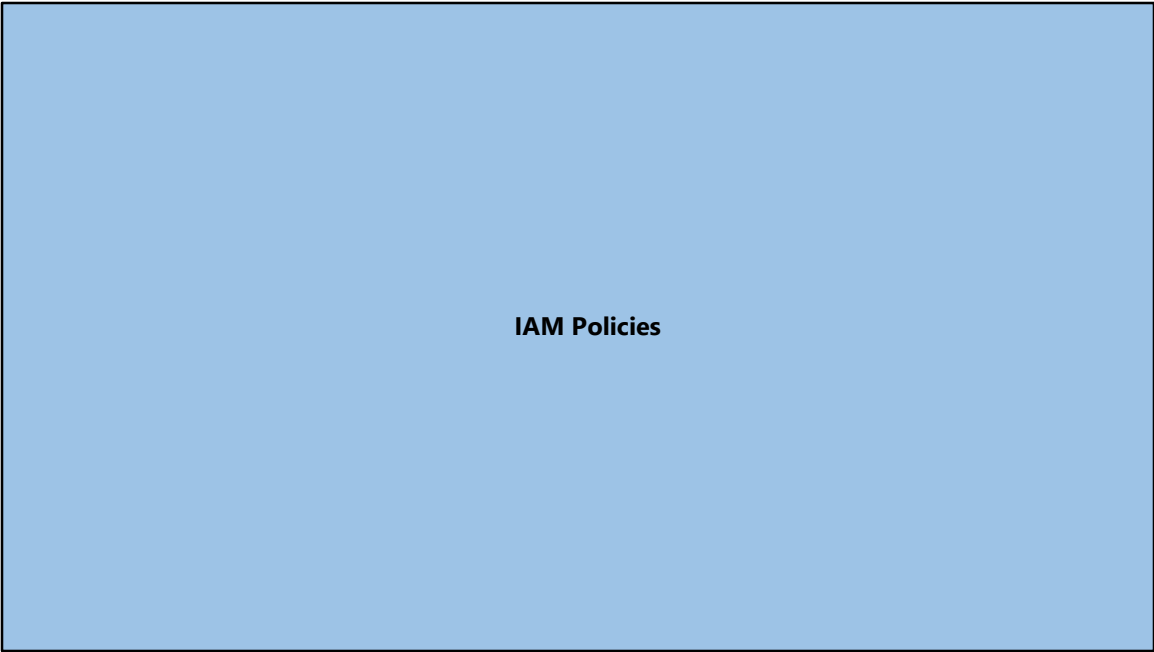
☒ Allow users to change their own password ⓘ

☐ Enable password expiration ⓘ
Password expiration period (in days): [input]

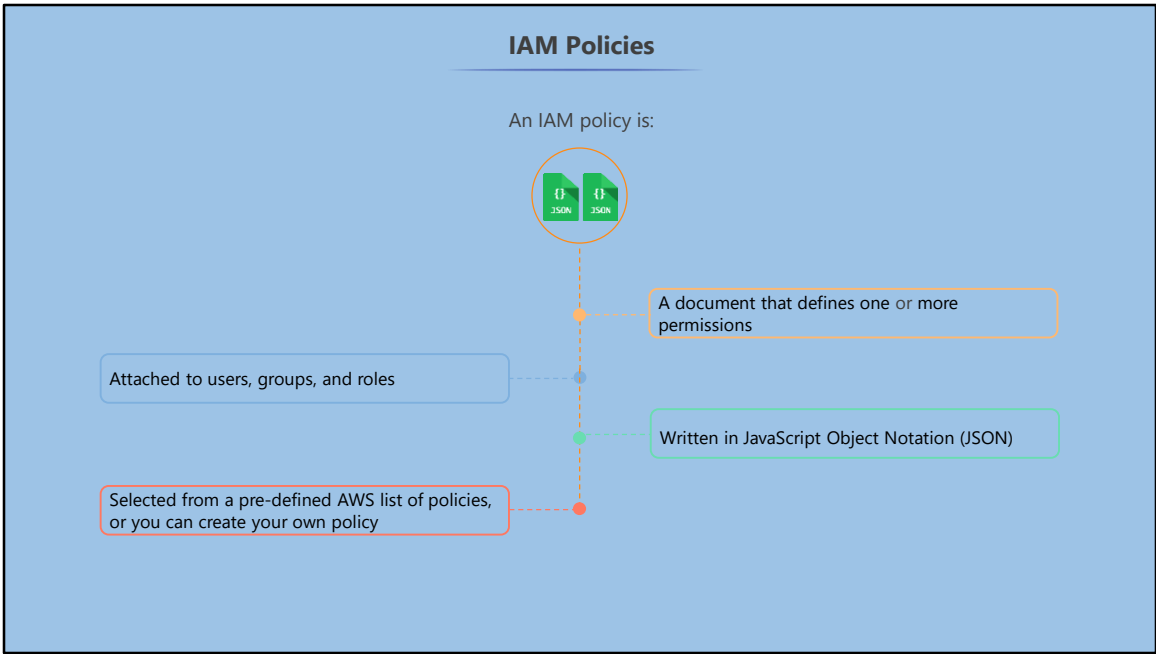
☐ Prevent password reuse ⓘ
Number of passwords to remember: [input]

☐ Password expiration requires administrator reset ⓘ

IAM allows you to define password strength and rotation policies, such as the number of characters, special characters, password expiration, and so on.



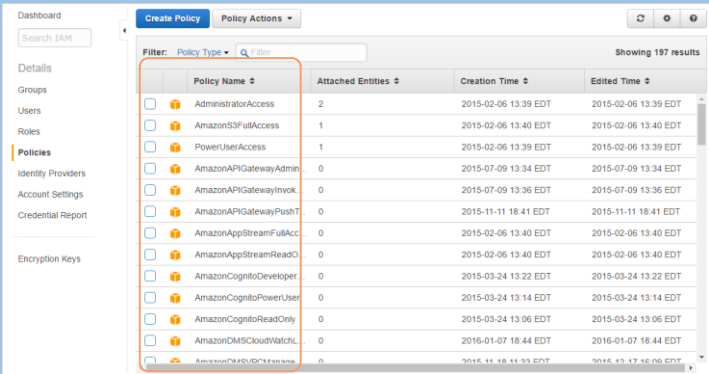
In this section you'll learn about the IAM policies.



An IAM policy is a document that defines one or more permissions. IAM policies can be attached to users, groups, and roles. They are written in JSON and can either be preselected from the AWS list of defined policies or you can edit these policies or even create your own.

AWS Policies

AWS has many predefined policies which allow you to define granular access to AWS resources.



Policy Name	Attached Entities	Creation Time	Edited Time
AdministratorAccess	2	2015-02-06 13:39 EDT	2015-02-06 13:39 EDT
AmazonSSIFullAccess	1	2015-02-06 13:40 EDT	2015-02-06 13:40 EDT
PowerUserAccess	1	2015-02-06 13:39 EDT	2015-02-06 13:39 EDT
AmazonAPIGatewayAdmin	0	2015-07-09 13:34 EDT	2015-07-09 13:34 EDT
AmazonAPIGatewayInvoke	0	2015-07-09 13:36 EDT	2015-07-09 13:36 EDT
AmazonAPIGatewayPushT	0	2015-11-11 18:41 EDT	2015-11-11 18:41 EDT
AmazonAppStreamFullAcc	0	2015-02-06 13:40 EDT	2015-02-06 13:40 EDT
AmazonAppStreamReadO	0	2015-02-06 13:40 EDT	2015-02-06 13:40 EDT
AmazonCognitoDeveloper	0	2015-03-24 13:22 EDT	2015-03-24 13:22 EDT
AmazonCognitoPowerUser	0	2015-03-24 13:14 EDT	2015-03-24 13:14 EDT
AmazonCognitoReadOnly	0	2015-03-24 13:06 EDT	2015-03-24 13:06 EDT
AmazonDMSCloudWatchL	0	2016-01-07 18:44 EDT	2016-01-07 18:44 EDT
AmazonDMSCloudWatchM	0	2016-11-18 11:53 EDT	2016-11-17 16:06 EDT



There are around 200 predefined policies available for you to choose from.

AWS provides many policies to choose from; currently there are around 200 available. In this section, we'll discuss some of the popular one's for you to get an idea of what a policy is.

AdministratorAccess Policy

AdministratorAccess policy provides full access to AWS services and resources.


Admin User



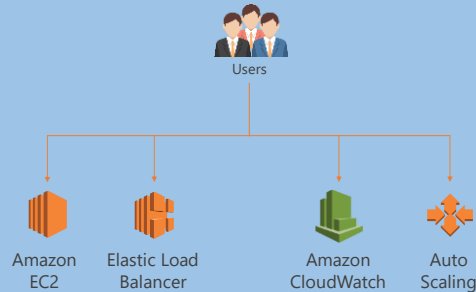


One of the most commonly used policies is the AdministratorAccess policy. This policy allows you to define granular access to AWS resources, for example, administration access allows full access to all the AWS services and resources.

This is usually assigned to a select few people.

AmazonEC2FullAccess Policy

AmazonEC2FullAccess policy provides AWS Directory Service user or groups full access to the Amazon EC2 services and resources.



Another popular policy is the AmazonEC2 Full Access policy; this policy provides an AWS Directory Service user or group full access to the following Amazon EC2 services and resources:

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Amazon CloudWatch
- Auto Scaling

This means that users have full access to EC2 but no access to any other service.

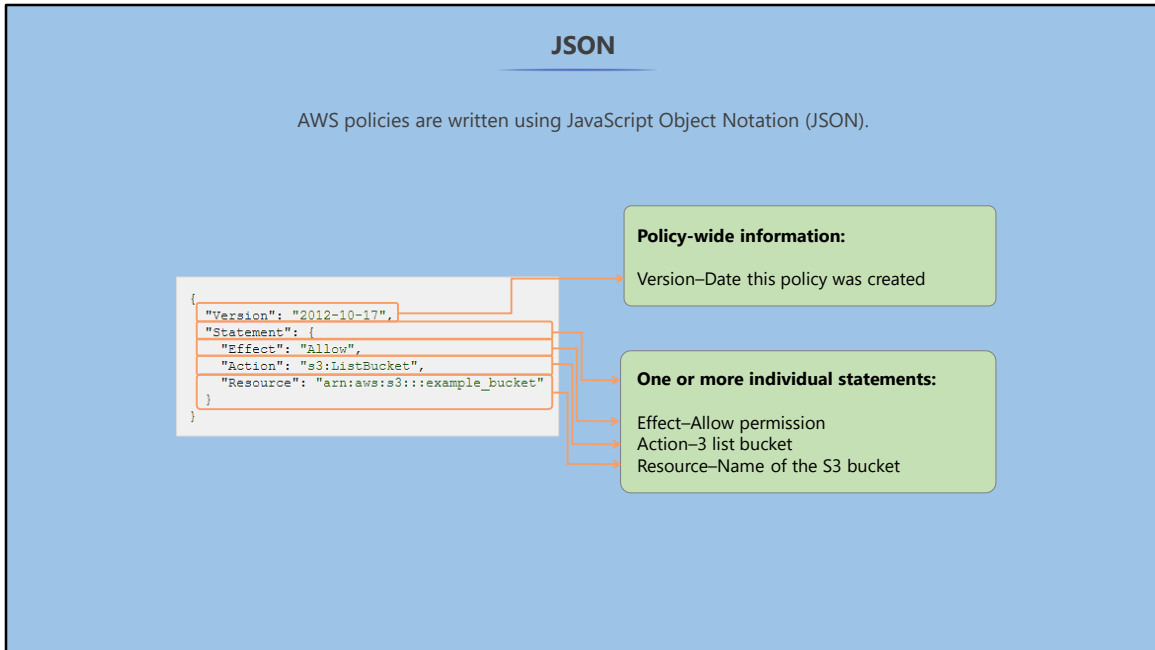
AmazonS3ReadOnlyAccess Policy

AmazonS3ReadOnlyAccess policy provides read-only access to all buckets using the AWS Management Console.



This policy provides read-only access to all buckets through the AWS Management Console.

Users have read-only access to S3 and no access to any other service.



Policies are written in JavaScript Object Notation, or JSON, which stores data in key-value pairs.

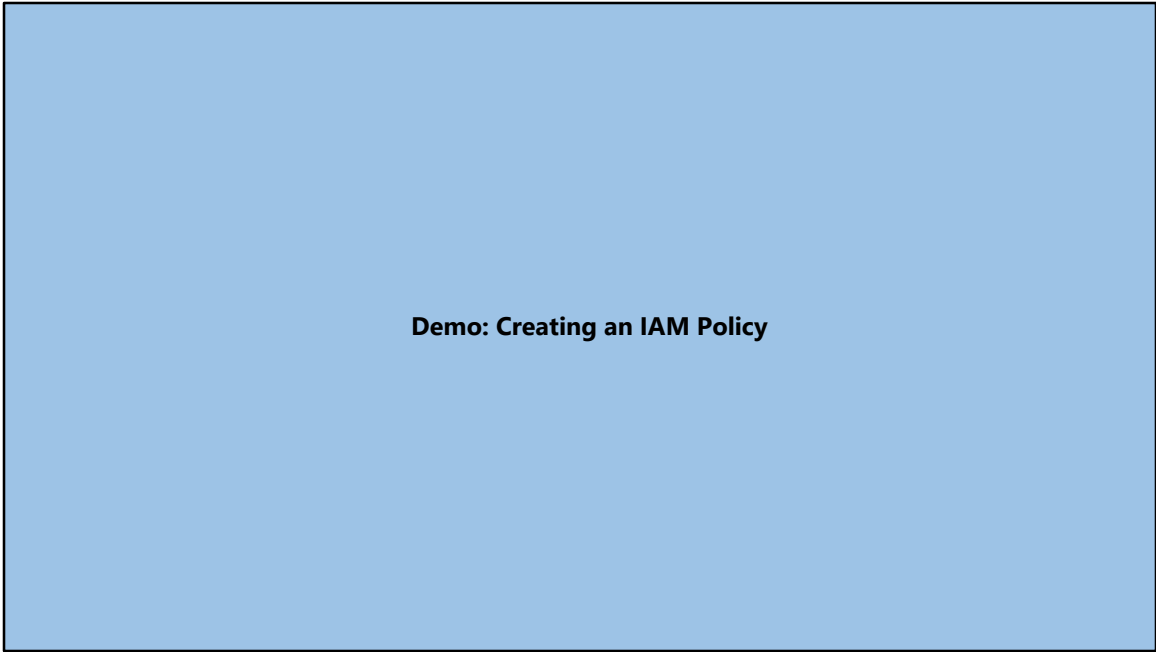
Here is an example of an AWS policy written in JSON.

In this example, you can see that the first line consists of some policy-wide information to help you identify some details about the policy.

In this case, it's the version and the date it was created, but you can update any information you want.

There is a statement section where the policy rules are written. In this example, there are three entries:

- **Effect:** The effect when the user requests access—either allow or deny. The default is that resources are denied to users, so you need to specify that you will allow users access to resources.
- **Actions:** What actions you will allow as each AWS service has its own set of actions. In this example, we allow S3 list bucket access. Any actions that you don't explicitly allow are denied.
- **Resources:** On which resources you should allow the action. In this example, we allow list bucket on the "example_bucket." Users with this policy cannot access any other buckets.



In this demonstration, you'll learn how to create an IAM Policy.

Knowledge Check

What does JSON stand for?

JavaScript Orientated Notation

JavaScript Object Notation

JavaScript Object Notes

JavaScript Open Notation

Knowledge
Check

1

What does JSON stand for?

- a. JavaScript Orientated Notation
- b. JavaScript Object Notation
- c. JavaScript Object Notes
- d. JavaScript Open Notation

The correct answer is **b**

JSON stands for JavaScript Object Notation and is used to write IAM Policies.

Knowledge
Check

2

In a JSON policy, what does the "effect" statement define?

- a. Whether the user is granted or denied permission
- b. The commands a user can perform
- c. The resources a user can run a command against
- d. Whether the user needs to use MFA to authenticate

**Knowledge
Check****2****In a JSON policy, what does the "effect" statement define?**

- a. Whether the user is granted or denied permission
- b. The commands a user can perform
- c. The resources a user can run a command against
- d. Whether the user needs to use MFA to authenticate

The correct answer is **a**

The "effect" statement defines what the effect will be when the user requests access—either allow or deny.

Knowledge
Check

3

What permissions would the AmazonEC2FullAccess policy give a user?

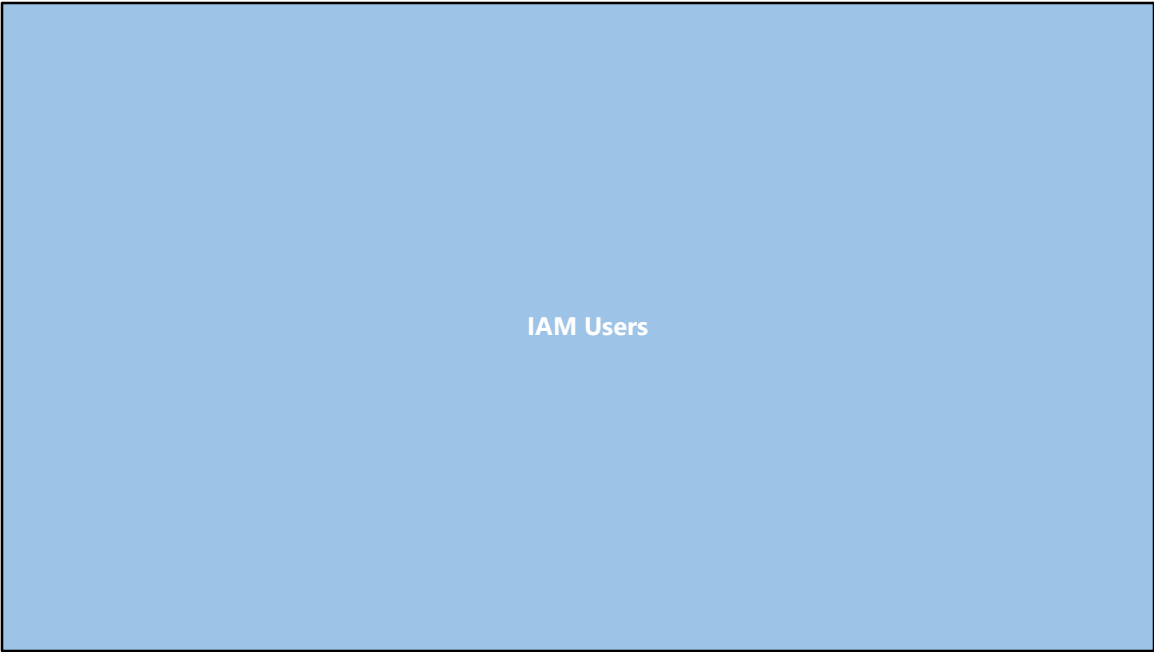
- a. Full Access to permissions to only EC2 instances
- b. Full Access to all AWS resources including EC2
- c. Full Access permissions to Amazon EC2 and only Elastic Load Balancing
- d. Full access to Amazon EC2, Elastic Load Balancer, and Amazon CloudWatch

**Knowledge
Check****3****What permissions would the AmazonEC2FullAccess policy give a user?**

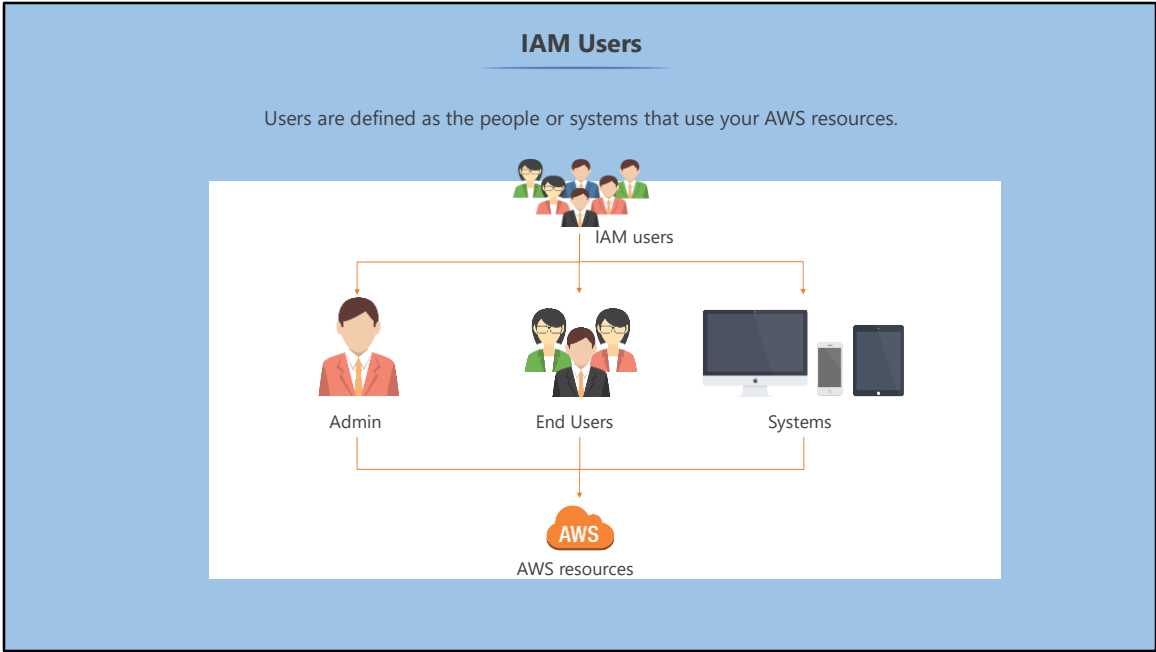
- a. Full Access to permissions to only EC2 instances
- b. Full Access to all AWS resources including EC2
- c. Full Access permissions to Amazon EC2 and only Elastic Load Balancing
- d. Full access to Amazon EC2, Elastic Load Balancer, and Amazon CloudWatch

The correct answer is **d**

This role provides an AWS Directory Service user or group with full access to Amazon EC2 services and the associated services and resources: Amazon Elastic Compute Cloud, Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling.



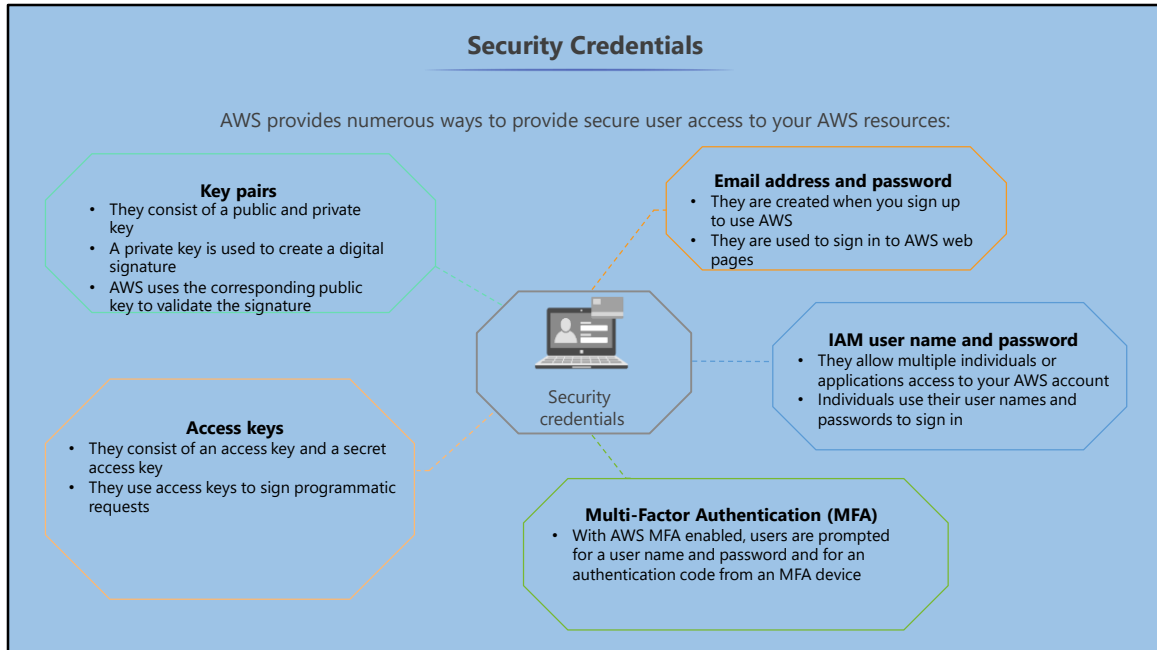
In this section you will learn what IAM Users are.



Users are defined as the people or systems that use your AWS resources.

The IAM users include:

- Administrators who need to access the AWS console and manage your AWS resources
- End users who need to access AWS content
- Systems that need permissions to access your AWS data



AWS provides numerous ways to ensure secure user access to AWS resources. Let's discuss them.

Email address and password: These are created when you sign up to use AWS. These are used to sign in to AWS web pages like the AWS Management Console, AWS Discussion Forums, or AWS Support Center.

IAM user name and password: This allows multiple individuals or applications to access your AWS account.

Users can use their user names and passwords to sign in to the AWS Management Console, AWS Discussion Forums, or AWS Support Center.

Multi-Factor Authentication (MFA): With AWS MFA enabled, when you sign in to an AWS website, you are prompted for your user name and password and for an authentication code from an MFA device.

Together these multiple factors provide increased security for your AWS account settings and resources.

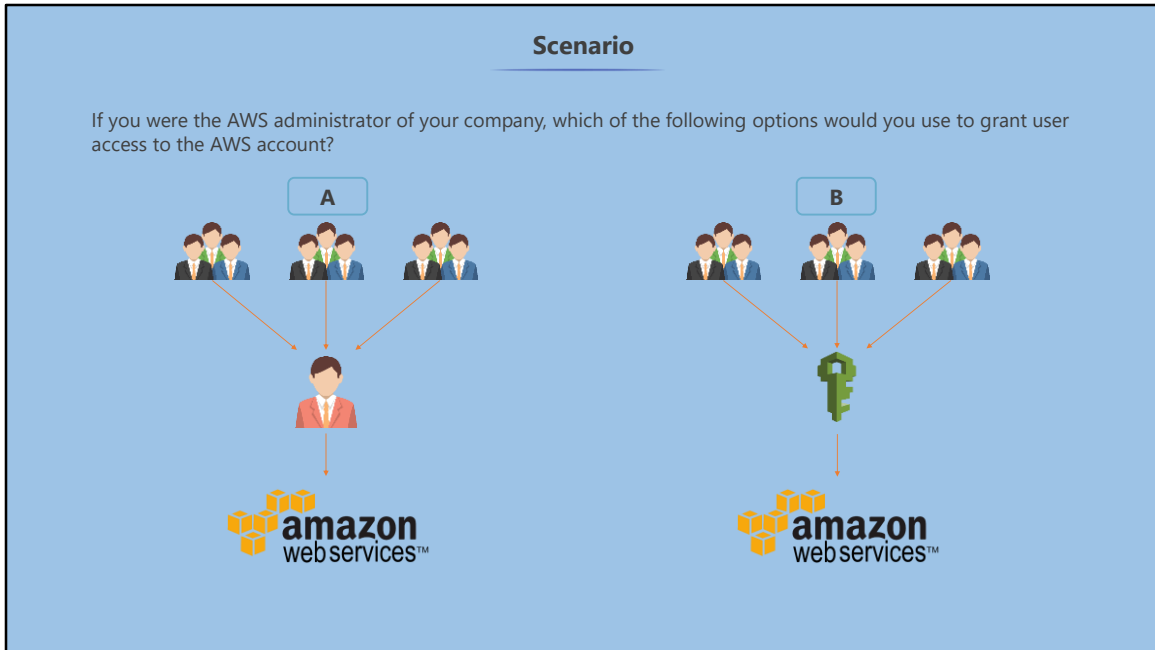
Access keys (access key ID and secret access key): Access keys consist of an access key and a secret access key.

Access keys are used to sign programmatic requests that you make to AWS whether you're using the AWS SDK, REST, or Query APIs.

Key pairs: Key pairs consist of a public and private key.

The private key is used to create a digital signature; AWS then uses the corresponding public key to validate the signature.

Key pairs are only used for Amazon EC2 and Amazon CloudFront.



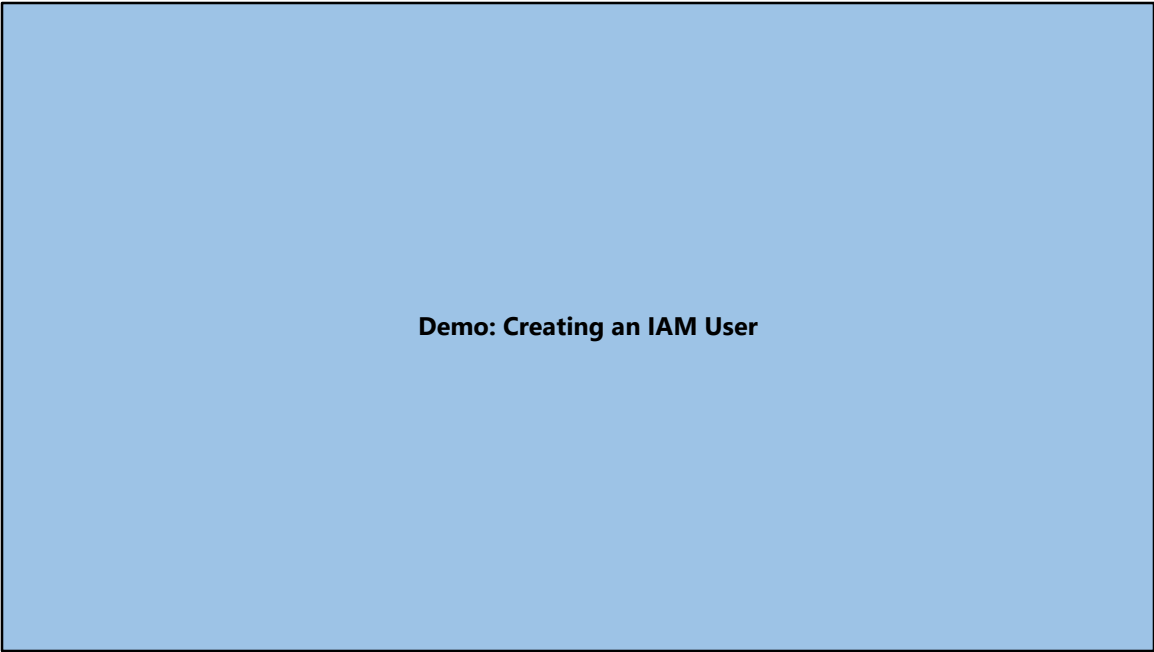
If you were the AWS administrator of your company, which of the following options would you use to grant user access to the AWS account?

Option A – All users accessing AWS using a single admin login, or

Option B – Users accessing AWS with their own user account.

Answer:

Option B – It provides granular access, auditing, security, and is easy to administer when people join or leave the company.



In this demonstration, you'll learn how to create an IAM User.

Knowledge Check

Knowledge
Check

1

What will automatically be generated when you create a new user?

- a. Access Key ID and Secret Access Key
- b. MFA token and password
- c. Secret Key and Encrypted Key
- d. Access Token and Access Key

**Knowledge
Check****1****What will automatically be generated when you create a new user?**

- a. Access Key ID and Secret Access Key
- b. MFA token and password
- c. Secret Key and Encrypted Key
- d. Access Token and Access Key

The correct answer is **a**

New users have an Access Key ID and Secret Access Key ID generated, which are viewable only at the time the IDs are created.

Knowledge
Check

2

What is the first step when you set up an AWS account?

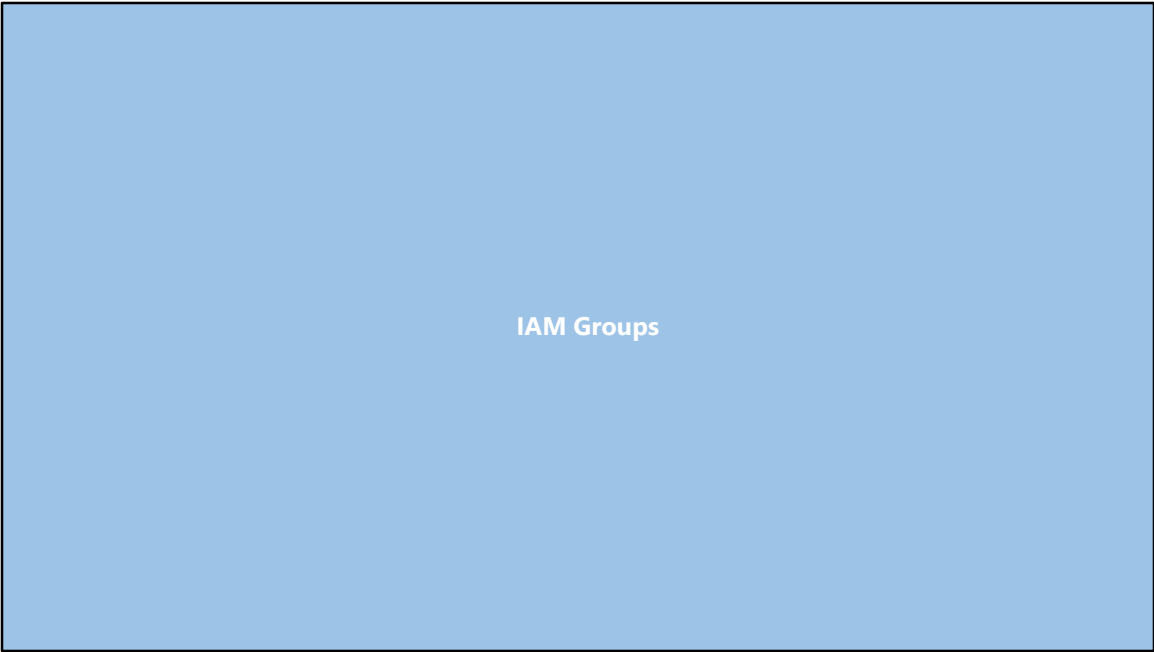
- a. Use CloudTrail to configure your account
- b. Setup a role that has the same name as your company
- c. Setup an account with your company email address
- d. Create a JSON policy to define who in your company can log in

**Knowledge
Check****2****What is the first step when you set up an AWS account?**

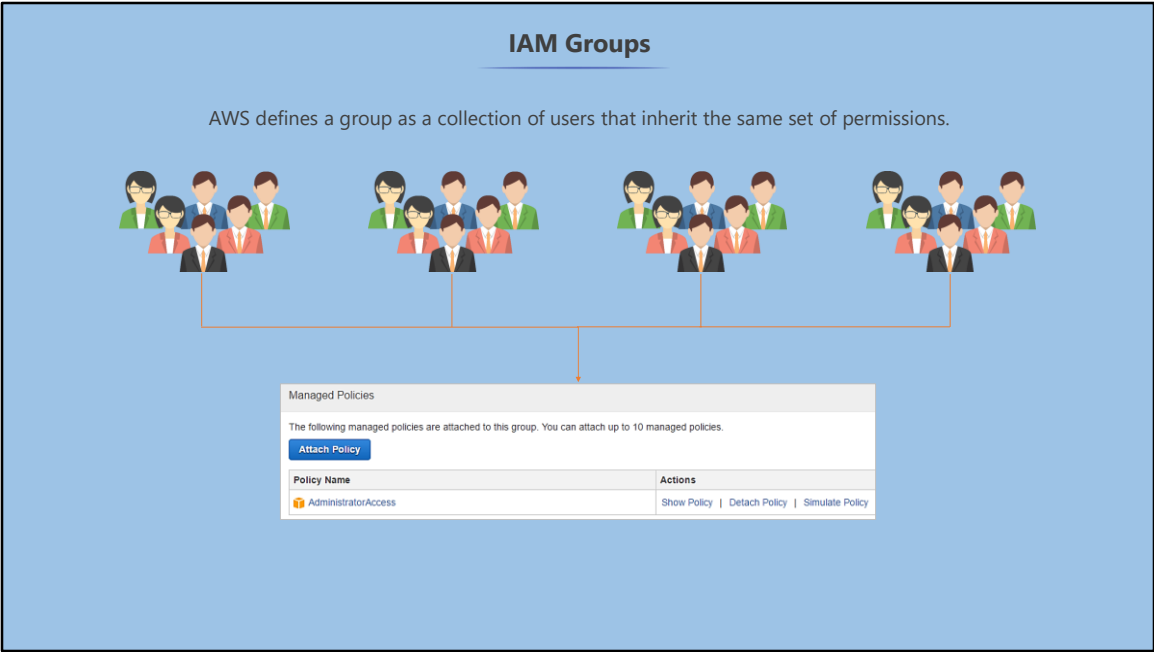
- a. Use CloudTrail to configure your account
- b. Setup a role that has the same name as your company
- c. Setup an account with your company email address
- d. Create a JSON policy to define who in your company can log in

The correct answer is **c**

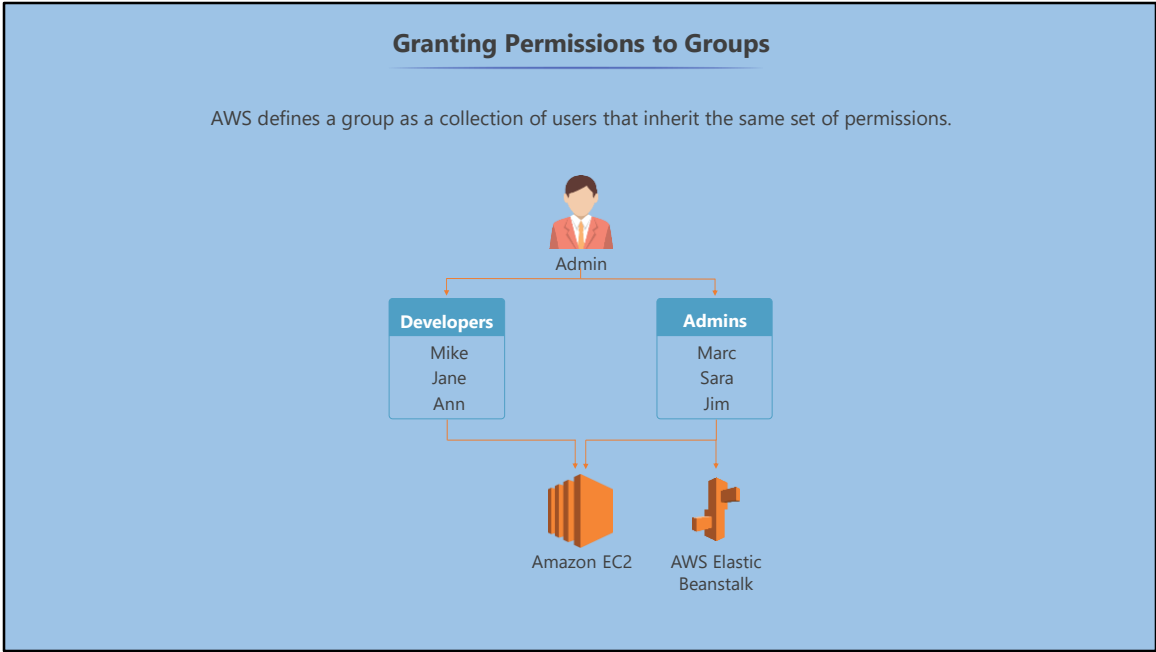
The first step is to create an account using your company email address. This account will be the root account.



In this section you'll learn about IAM groups.



AWS defines a group as a collection of users that inherit the same set of permissions.



Each time new users join your company, you need to grant them the permissions they require. By creating a group, you can reduce user management overhead. You need to attach policies to it just once rather than for doing for each user.

Demo: Creating an IAM Group

In this demonstration, you'll learn how to create an IAM Group.

Knowledge Check

Knowledge
Check

1

How does AWS define a group?

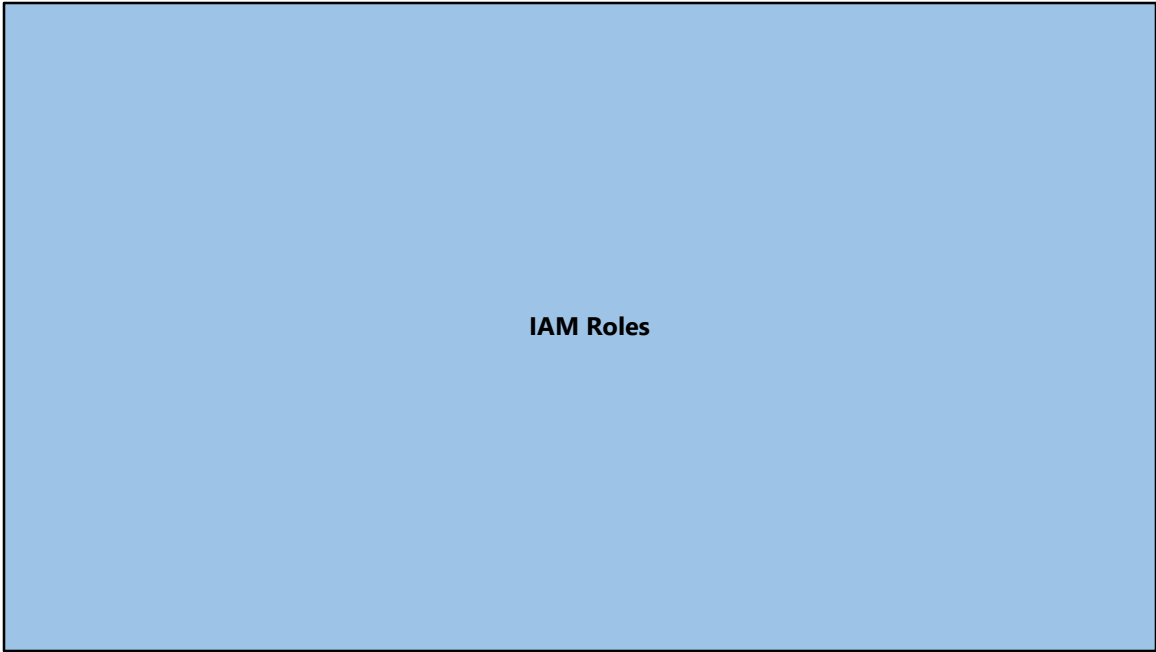
- a. A collection of roles that share similar policy documents
- b. A collection of users that all inherit the same set of permissions
- c. An entity that controls secure access to EC2 resources
- d. A resource to use when setting up MFA

**Knowledge
Check****1****How does AWS define a group?**

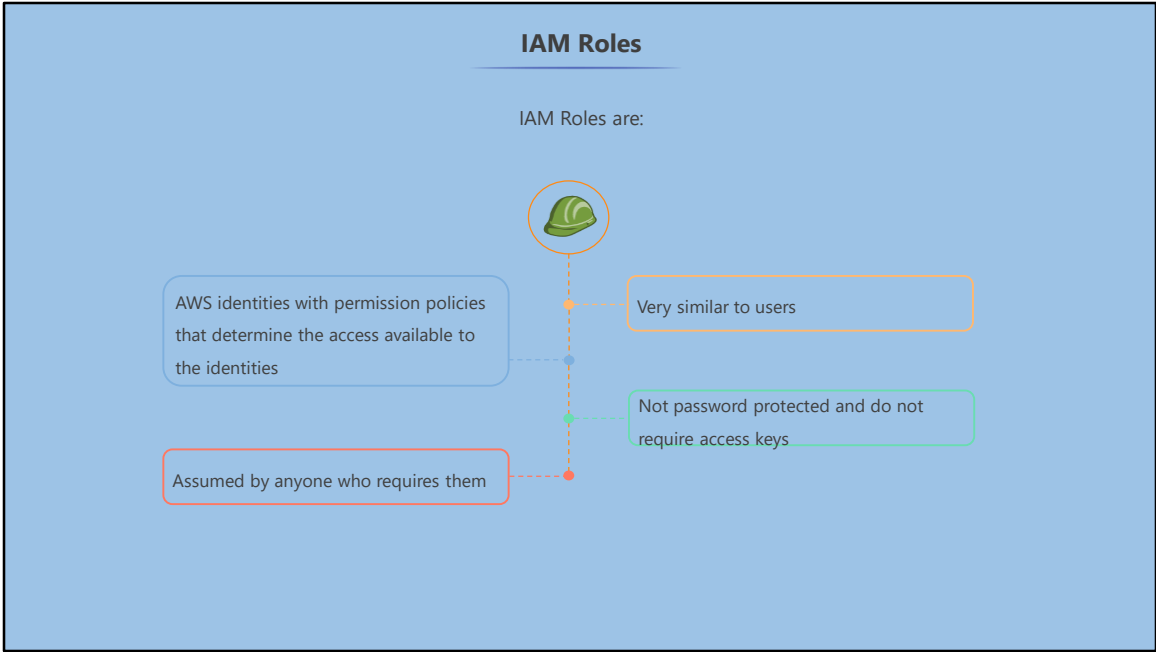
- a. A collection of roles that share similar policy documents
- b. A collection of users that all inherit the same set of permissions
- c. An entity that controls secure access to EC2 resources
- d. A resource to use when setting up MFA

The correct answer is **b**

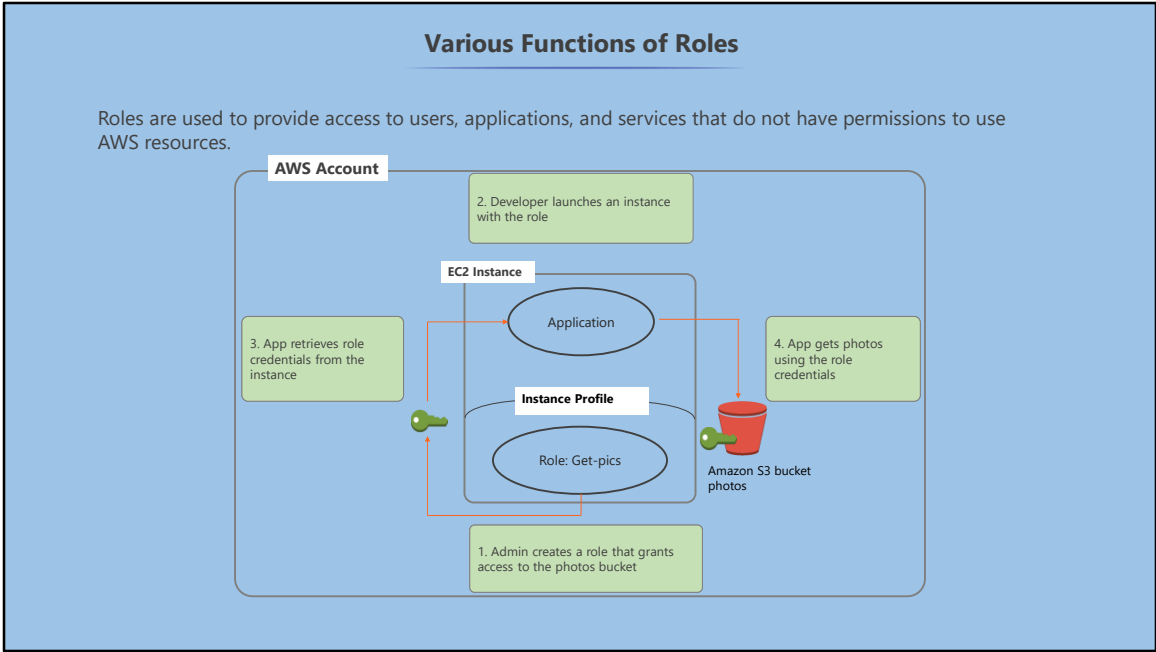
An IAM group is a collection of IAM users. You can use groups to specify permissions for a collection of users, which can make those permissions easier to manage for those users.



In this section you will learn what IAM roles are.



An IAM role is similar to a user.
It’s an AWS identity with permission policies that determine what the identity can and cannot do. However; a role can be assumed by anyone who needs it, and a role does not have any password or access keys associated with it.
Roles can be used to delegate access to users, applications, or services that are not typically able to access your AWS resources.
An example of this would be to allow a mobile app permission to use AWS without storing the AWS keys in the app.



Roles can be used to delegate access to users, applications, or services that are typically unable to access your AWS resources.

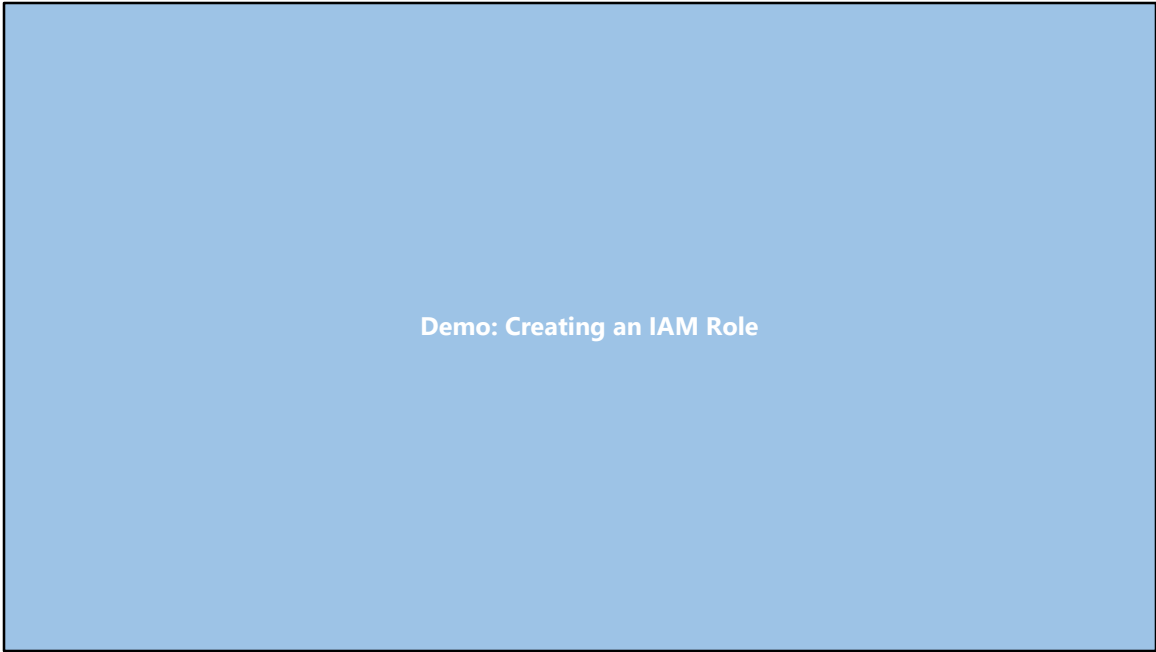
If you look at the diagram, it's an example of a mobile app that wants permission to use AWS but doesn't actually store AWS access keys in the app. We'll cover this in detail in the EC2 section, but here's a brief overview:

Step 1: The administrator creates a role that gives read access to the photos bucket.

Step 2: Then, you launch an instance with this role, which means that any application or user that accesses or uses or runs on this instance will automatically inherit the permissions of the role.

Step 3: When the application runs it, it retrieves the role from the instance, which means it can access the photos bucket.

Step 4: The application performs the action it requires, which is to get photos from the bucket.



In this demonstration you'll learn how to create an IAM Role.

Knowledge Check

Knowledge
Check

1

How do you assign permissions to an IAM user, group, or role?

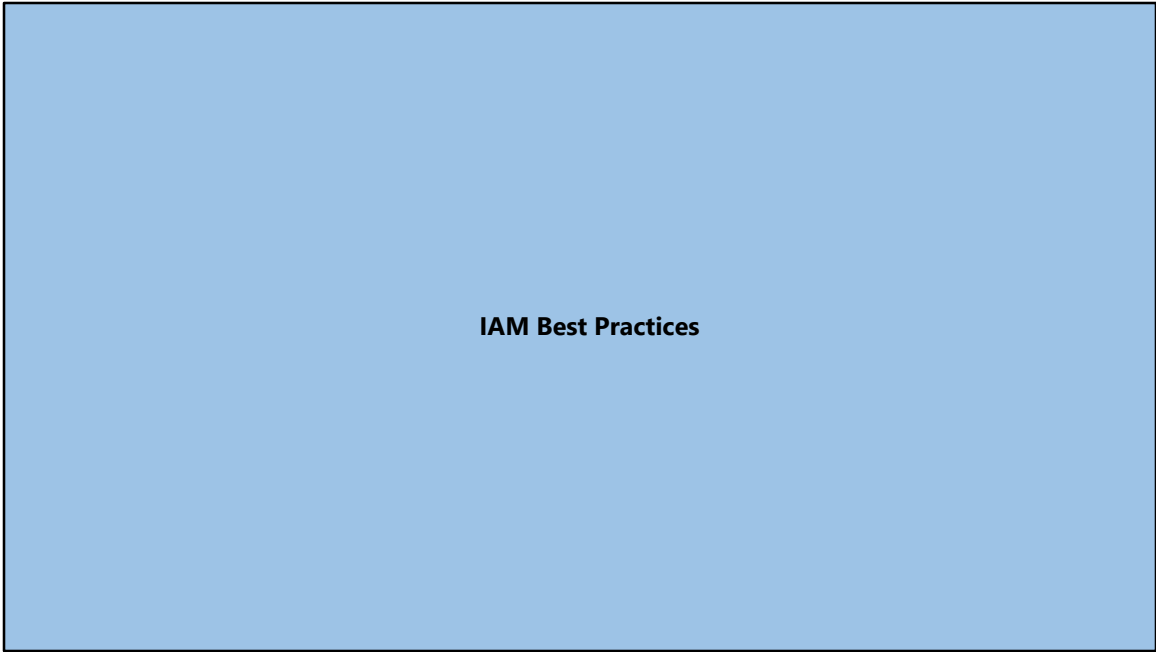
- a. Using a security group
- b. Using a permissions document
- c. Using a policy document
- d. Using Identity Federation

**Knowledge
Check****1****How do you assign permissions to an IAM user, group, or role?**

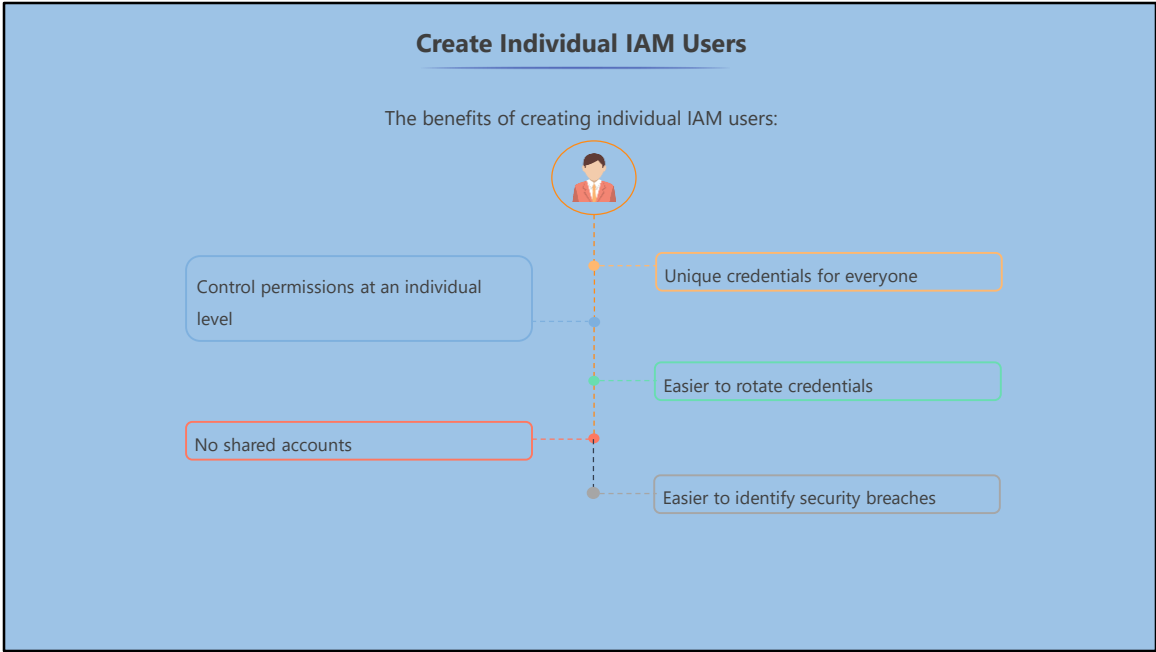
- a. Using a security group
- b. Using a permissions document
- c. Using a policy document
- d. Using Identity Federation

The correct answer is **c**

A policy document written in JSON is used to assign permissions.



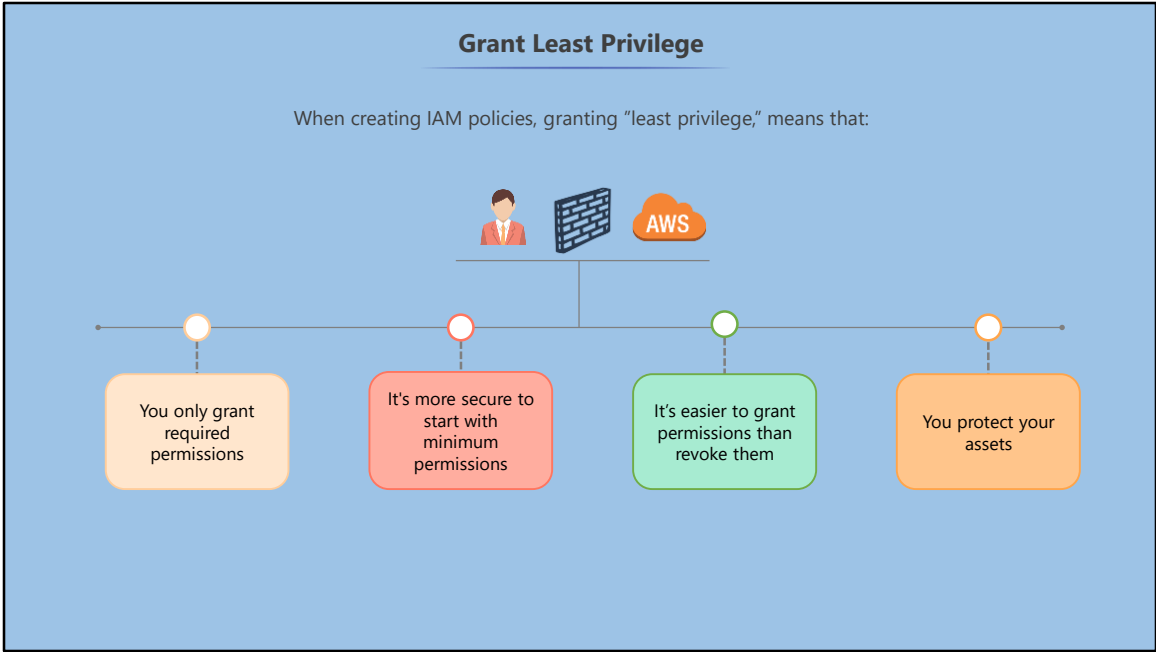
AWS has a list of IAM best practices to ensure that your environment is secure and safe. In this section you'll learn some of the IAM Best Practices.



Do not use your AWS root account credentials to access AWS. Create individual users for each person who needs access to your AWS resources.

This ensures that everyone has unique credentials, and you can control permissions at an individual level rather than people sharing user accounts that have more access rights than they actually require.

Additionally, individual credentials mean you can ensure that users rotate their credentials more often. It is often difficult to change passwords on shared accounts. If there is a security breach or an outage caused by human error, individual credentials make the forensic investigation that much easier.



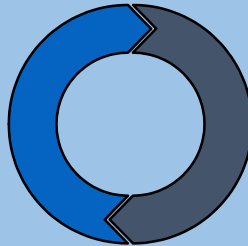
When you create IAM policies, make sure you grant the least privilege, in other words, only grant the permissions that users require to perform their task. Although it's easier to grant elevated credentials from the start, it's not good practice and is more secure to start with the minimum permissions required and grant additional access as required. It's easier to grant permissions to a user than trying to revoke them.

Manage Permissions with Groups

Use permissions with groups to minimize the workload

Easy to assign new permissions

It is easier to assign a new permission to a group than to assign it to many individual users.



Simple to reassign permissions

It is simpler to reassign permissions if a user has a change in responsibilities.

Using groups minimizes your workload.

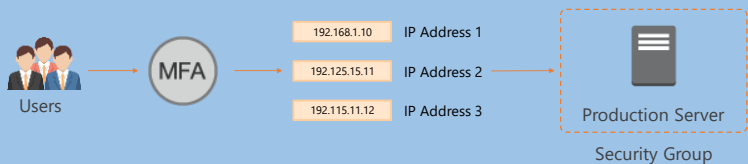
It is easier to assign a new permission to a group than to assign it to many individual users as one change can update the permissions for multiple users.

It's also simpler to reassign permissions if a user has a change in responsibilities.

For example, if a developer moves to the admin team, you just have to move the user account from the developer group to the admin group and your work is complete.

Restrict Access with Further Conditions

Use additional conditions such as MFA and Security Groups to ensure only the intended users get access.



To ensure that your resources are protected, it’s always a good idea to add additional access controls such as:

Requiring the use of MFA to login or

Specifying that access to certain resources can only come from a particular IP address.

A good example of this would be allowing remote desktop protocol (RDP) access to a production server.

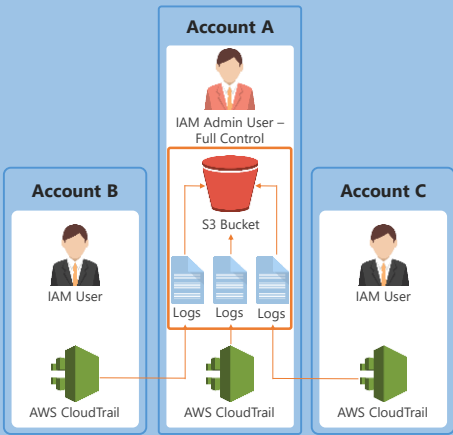
You could change the configuration so that to login with an account that has this permission, users need to connect using MFA.

Then, once logged in, you can specify that the user can only access RDP to the server from a particular range of IP addresses.

Monitor Activity in your AWS Account

AWS has several features to log user actions.

- Logs
- AWS Cloudtrail



AWS offers several features you can use to log user actions. Log files show the time and date of actions, the source IP for an action, which actions failed due to inadequate permissions, and more. The best monitoring tool for IAM is CloudTrail, which logs AWS API calls and related events made by or on behalf of an AWS account.

Create a Strong Password Policy

Ensure that all your users have strong passwords and they rotate their passwords regularly.

Minimum password length:

☐

Require at least one uppercase letter ⓘ

☐

Require at least one lowercase letter ⓘ

☐

Require at least one number ⓘ

☐

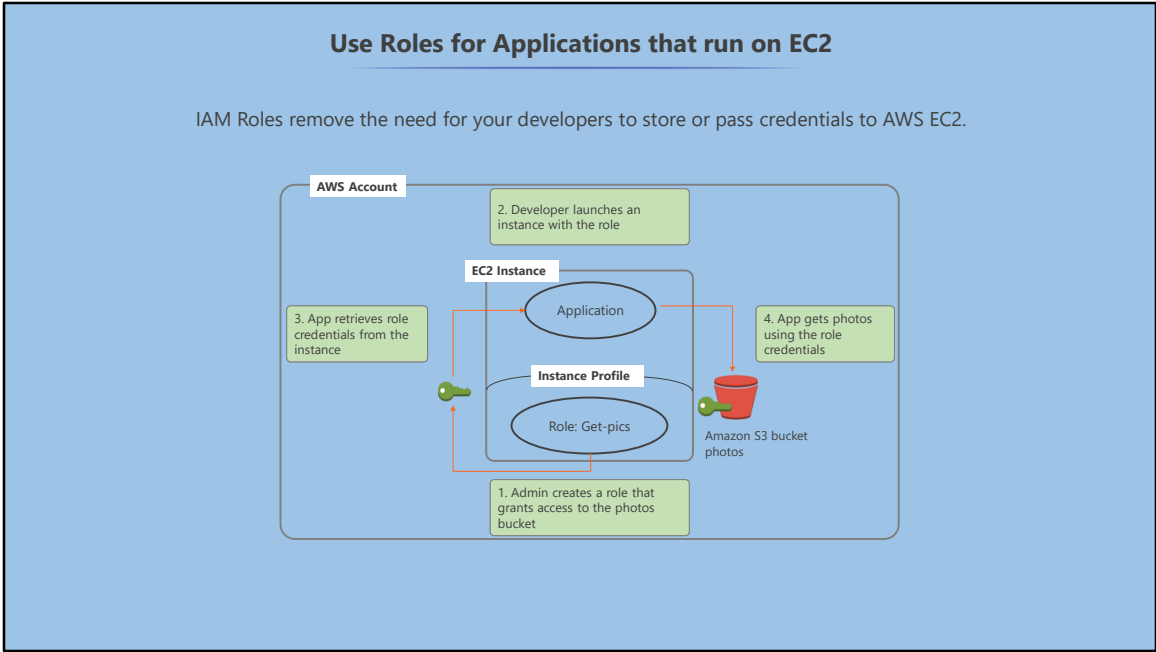
Require at least one non-alphanumeric character ⓘ☒☐Password expiration period (in days): ☐Number of passwords to remember: ☐

Ensure that users require strong passwords and they rotate their passwords periodically.

Define a suitable password policy that sets requirements such as minimum length, whether it requires non-alphanumeric characters, how frequently it must be rotated, and so on.

Regularly enforcing your users to change their passwords and access keys secures your AWS resources.

If a password or access key is compromised without your knowledge, a password rotation policy will limit how long the credentials can be used to access your resources.



Applications that run on Amazon EC2 instances need credentials to interact with other AWS products and services.

Instead of developers passing credentials to AWS EC2, use IAM roles so that temporary credentials are dynamically passed to EC2.

This means that there is no need to share security credentials and no need to store long-term credentials. IAM automatically rotates these credentials for you.

When you launch new EC2 instances, specify an IAM role for the instance, then any applications that run on the EC2 instance will inherit the role's credentials when they access AWS resources.

Reduce or Remove Unnecessary Credentials

To reduce the potential for misuse, run a credential report to identify users that are no longer in use and can be removed.



To reduce the potential for misuse:

- Avoid using the root account unless absolutely necessary; instead, create accounts that have only the required access.
- Remove IAM user credentials that are not needed.
- For example, if you create an account that doesn't need to log in to the AWS console, such as an IAM user that is used for an application, then remove and delete their password and access key.
- Run a credential report to show how recently the credentials have been used.
- Passwords and access keys that have not been used recently should be removed.

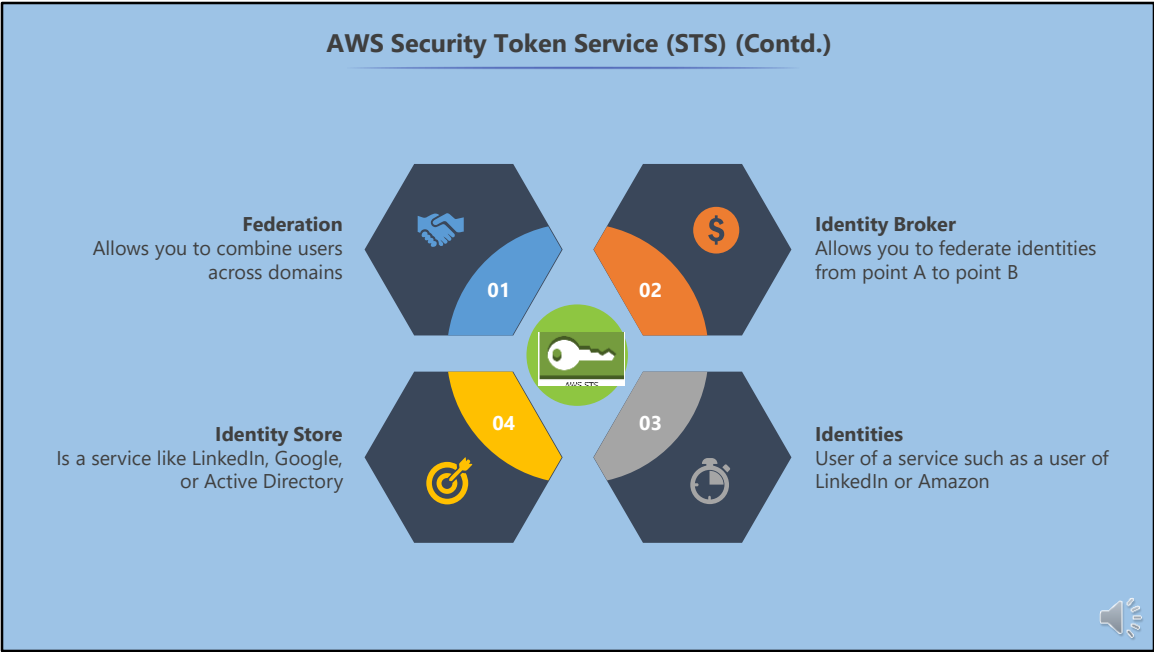
AWS Security Token Service (STS)

It is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management users that you authenticate.



The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users).

Basically it's a way to allow users in one domain, for example IAM, connect to users in another, for example Facebook or Active Directory.

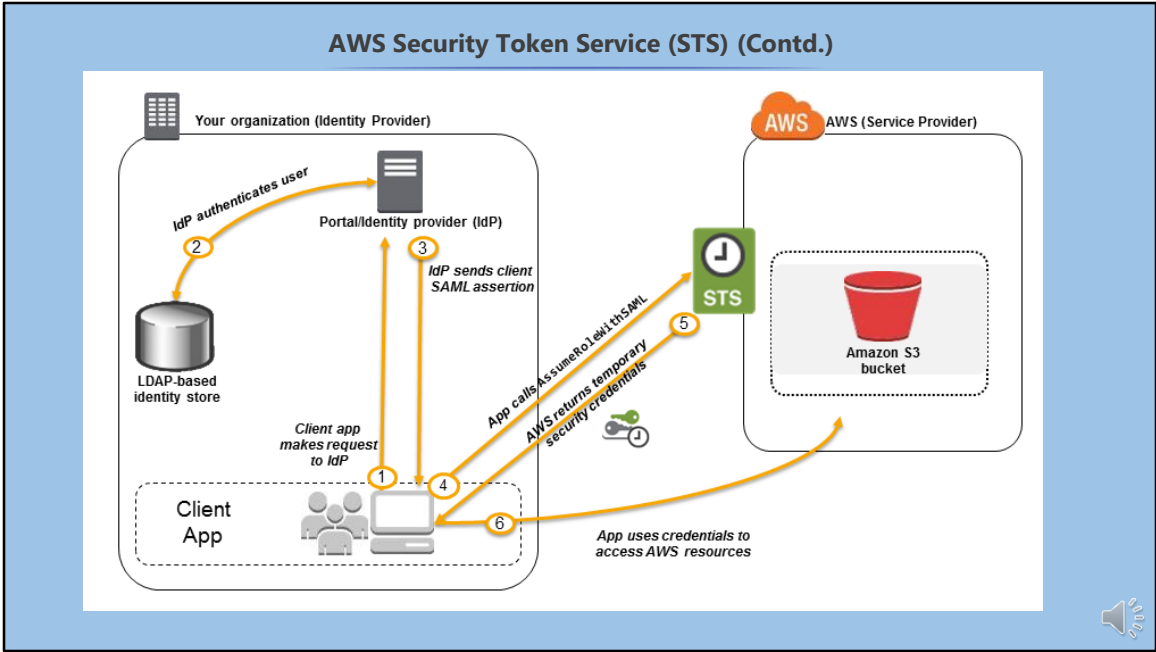


Federation: combining users across domains. For example, join IAM users with LinkedIn users.

Identity Broker: Allows you to federate (join) identities from point A to point B

Identity Store: A service like LinkedIn, Google or Active Directory.

Identities: The user of a service like LinkedIn or Amazon.



Imagine that in your organization, you want to provide a way for users to copy data from their computers to a backup folder. You build an application that users can run on their computers. On the back end, the application reads and writes objects in an S3 bucket.

Users don't have direct access to AWS.

Instead, the following process is used:

A user in your organization uses a client app to request authentication from your organization's IdP.

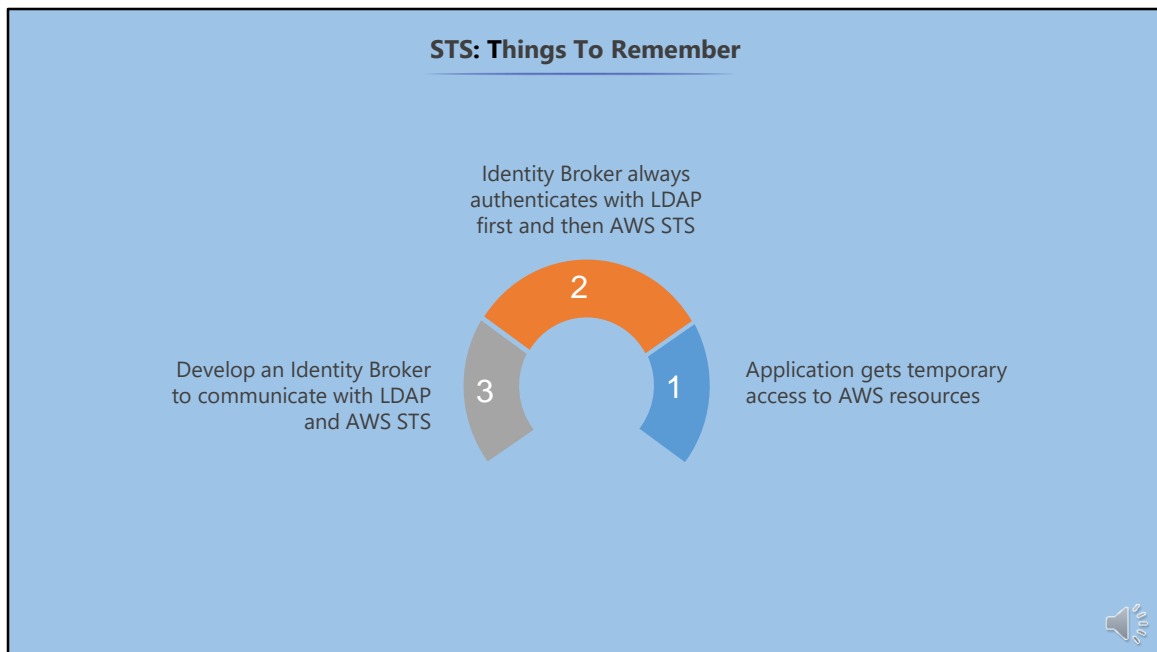
The IdP authenticates the user against your organization's identity store.

The IdP constructs a SAML assertion with information about the user and sends the assertion to the client app.

The client app calls the AWS STS AssumeRoleWithSAML API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.

The API response to the client app includes temporary security credentials.

The client app uses the temporary security credentials to call Amazon S3 APIs.



Imagine that in your organization, you want to provide a way for users to copy data from their computers to a backup folder. You build an application that users can run on their computers. On the back end, the application reads and writes objects in an S3 bucket.

Users don't have direct access to AWS.

Instead, the following process is used:

A user in your organization uses a client app to request authentication from your organization's IdP.

The IdP authenticates the user against your organization's identity store.

The IdP constructs a SAML assertion with information about the user and sends the assertion to the client app.

The client app calls the AWS STS AssumeRoleWithSAML API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.

The API response to the client app includes temporary security credentials.

The client app uses the temporary security credentials to call Amazon S3 APIs.

Knowledge Check

Knowledge
Check

What does MFA stand for?

1

- a. Multi-Faced Access
- b. Multi-Factor Administration
- c. Mission Factored Authentication
- d. Multi-Factor Authentication

**Knowledge
Check****1****What does MFA stand for?**

- a. Multi-Faced Access
- b. Multi-Factor Administration
- c. Mission Factored Authentication
- d. Multi-Factor Authentication

The correct answer is **d**

For increased security, AWS recommends that you configure multi-factor authentication (MFA) to help protect your AWS resources. MFA adds extra security because it requires users to enter a unique authentication code from an approved authentication device or SMS text message when they access AWS websites or services.

Knowledge
Check

2

What AWS tool is used to track, monitor, and log IAM user activity?

- a. CloudFormation
- b. Inspector
- c. CloudWatch
- d. CloudTrail

Knowledge
Check

2

What AWS tool is used to track, monitor, and log IAM user activity?

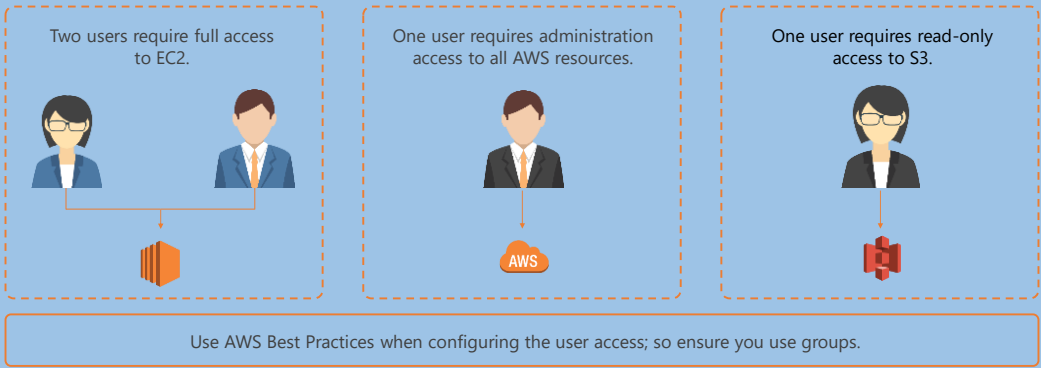
- a. CloudFormation
- b. Inspector
- c. CloudWatch
- d. CloudTrail

The correct answer is **d**

CloudTrail is used to track user activity. CloudFormation allows you to manage resources with templates, CloudWatch monitors application activity, and Inspector analyzes application security.

Practice Assignment: Configuring IAM Access

As the admin for your company's AWS account, you need to assign permissions to four new users:



- AWS Identity and Access Management (IAM) allows you to securely control access to AWS services and resources for your users.
- Policies are written in JSON and allow you to define granular access to AWS resources.
- Users are the people or systems that use your AWS resources
- Groups are a collection of users that inherit the same set of permissions and can be used to reduce your user management overhead.
- IAM roles can be assumed by anyone who needs them, and they do not have an access keys or passwords associated with them.
- AWS has a list of IAM best practices to ensure your environment is secure and safe.

Knowledge Check

Knowledge
Check

What are IAM entities?

1

- a. User, Teams, Roles
- b. User, Group, Companies
- c. Sessions, Group, Organizations
- d. User, Group, Roles

**Knowledge
Check****1****What are IAM entities?**

- a. User, Teams, Roles
- b. User, Group, Companies
- c. Sessions, Group, Organizations
- d. User, Group, Roles

The correct answer is **d**

User, Groups, and Roles are the entities used in IAM.

Knowledge
Check
2

Which AWS compliance allows you to safely and securely manage and store credit card data?

- a. HIPAA
- b. PCI DSS
- c. JSON
- d. EC2

Knowledge
Check

2

Which AWS compliance allows you to safely and securely manage and store credit card data?

- a. HIPAA
- b. PCI DSS
- c. JSON
- d. EC2

The correct answer is **b**

IAM is Payment Card Industry (PCI) Data Security Standard (DSS) compliant so you can process, store, and transmit credit card data from a merchant or service provider.

Knowledge
Check

3

What language is used to authenticate IAM with Federated Access?

- a. JSON
- b. ODBC
- c. SSL
- d. SAML 2.0

**Knowledge
Check****3****What language is used to authenticate IAM with Federated Access?**

- a. JSON
- b. ODBC
- c. SSL
- d. SAML 2.0

The correct answer is **d**

This feature enables federated single sign-on (SSO), so users can log in to the AWS Management Console or call the AWS APIs without you having to create an IAM user for everyone in your organization.

Knowledge
Check

4

What does a user need to login to the AWS console?

- a. Username, Access Key ID, and Secret Access Key ID
- b. MFA token
- c. Username and password
- d. Username and policy document

**Knowledge
Check****4****What does a user need to login to the AWS console?**

- a. Username, Access Key ID, and Secret Access Key ID
- b. MFA token
- c. Username and password
- d. Username and policy document

The correct answer is **c**

The Access Key ID and Secret Access Key ID are generated when you create a user, but to log in to the AWS console you need to generate a password for the user.

Knowledge
Check

5

What is a good way to restrict AWS user access using further conditions?

- a. Inform users they can only login at certain times
- b. Make users commit their Access Key ID to memory
- c. Use Multi-Factor Authentication
- d. Only use policies for administration users

Knowledge
Check

5

What is a good way to restrict AWS user access using further conditions?

- a. Inform users they can only login at certain times
- b. Make users commit their Access Key ID to memory
- c. Use Multi-Factor Authentication
- d. Only use policies for administration users

The correct answer is **c**

MFA request users to pass an additional authentication check to be able to login. Other examples of further conditions are specifying that access to certain resources can only come from a particular IP address.