**Security, Identity and Compliance**

**AWS Identity and Access Management (IAM)**

Q: What is AWS Identity and Access Management (IAM)?

You can use AWS IAM to securely control individual and group access to your AWS resources. You can create and manage user identities ("IAM users") and grant [permissions](https://aws.amazon.com/iam/details/manage-permissions/) for those IAM users to access your resources. You can also grant permissions for users outside of AWS ( [federated users](https://aws.amazon.com/identity/federation/)).

Q: What problems does IAM solve?

IAM makes it easy to provide multiple users secure access to your AWS resources. IAM enables you to:

* Manage IAM users and their access: You can create users in AWS's identity management system, assign users individual security credentials (such as access keys, passwords, [multi-factor authentication](https://aws.amazon.com/identity/saml/) devices), or request [temporary security credentials](http://docs.aws.amazon.com/STS/latest/UsingSTS/Welcome.html) to provide users access to AWS services and resources. You can specify permissions to control which operations a user can perform.
* Manage access for [federated users](https://aws.amazon.com/identity/federation/): You can request security credentials with configurable expirations for users who you manage in your corporate directory, allowing you to provide your employees and applications secure access to resources in your AWS account without creating an IAM user account for them. You specify the permissions for these security credentials to control which operations a user can perform.

Q: What is a user?

A user is a unique identity recognized by AWS services and applications. Similar to a login user in an operating system like Windows or UNIX, a user has a unique name and can identify itself using familiar security credentials such as a password or access key. A user can be an individual, system, or application requiring access to AWS services. IAM supports users (referred to as "IAM users") managed in AWS's identity management system, and it also enables you to grant access to AWS resources for users managed outside of AWS in your corporate directory (referred to as "federated users").

Q: What can a user do?

A user can place requests to web services such as Amazon S3 and Amazon EC2. A user's ability to access web service APIs is under the control and responsibility of the AWS account under which it is defined. You can permit a user to access any or all of the AWS services that have been integrated with IAM and to which the AWS account has subscribed. If permitted, a user has access to all of the resources under the AWS account. In addition, if the AWS account has access to resources from a different AWS account, its users may be able to access data under those AWS accounts. Any AWS resources created by a user are under control of and paid for by its AWS account. A user cannot independently subscribe to AWS services or control resources.

Q: What is a group?  
A group is a collection of IAM users. Manage group membership as a simple list:

* Add users to or remove them from a group.
* A user can belong to multiple groups.
* Groups cannot belong to other groups.
* Groups can be granted permissions using access control policies. This makes it easier to manage permissions for a collection of users, rather than having to manage permissions for each individual user.
* Groups do not have security credentials, and cannot access web services directly; they exist solely to make it easier to manage user permissions. For details, see [Working with Groups and Users](http://docs.aws.amazon.com/IAM/latest/UserGuide/Using_WorkingWithGroupsAndUsers.html).

Q: What kinds of security credentials can IAM users have?

IAM users can have any combination of credentials that AWS supports, such as an AWS access key, X.509 certificate, SSH key, password for web app logins, or an [MFA](https://aws.amazon.com/identity/saml/) device. This allows users to interact with AWS in any manner that makes sense for them. An employee might have both an AWS access key and a password; a software system might have only an AWS access key to make programmatic calls; IAM users might have a private SSH key to access AWS CodeCommit repositories; and an outside contractor might have only an X.509 certificate to use the EC2 command-line interface.

Q: Who is able to manage users for an AWS account?

The AWS account holder can manage users, groups, security credentials, and permissions. In addition, you may grant permissions to individual users to place calls to IAM APIs in order to manage other users. For example, an administrator user may be created to manage users for a corporation—a recommended practice. When you grant a user permission to manage other users, they can do this via the IAM APIs, [AWS CLI](http://aws.amazon.com/developertools/AWS-Identity-and-Access-Management/4143), or [IAM console](https://console.aws.amazon.com/iam/home).

Q: What kind of key rotation is supported for IAM users?

User access keys and X.509 certificates can be rotated just as they are for an AWS account's root access identifiers. You can manage and rotate programmatically a user's access keys and X.509 certificates via the IAM APIs, AWS CLI, or IAM console.

Q: Can IAM users have individual EC2 SSH keys?

Not in the initial release. IAM does not affect EC2 SSH keys or Windows RDP certificates. This means that although each user has separate credentials for accessing web service APIs, they must share SSH keys that are common across the AWS account under which users have been defined.

Q: Can I define a password policy for my user’s passwords?

Yes, you can enforce strong passwords by requiring minimum length or at least one number. You can also enforce automatic password expiration, prevent re-use of old passwords, and require a password reset upon the next AWS sign-in.

Q: What is an IAM role?

An IAM role is an IAM entity that defines a set of [permissions](https://aws.amazon.com/iam/details/manage-permissions/) for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

Q: What problems do IAM roles solve?

IAM roles allow you to delegate access with defined permissions to trusted entities without having to share long-term access keys. You can use IAM roles to delegate access to IAM users managed within your account, to IAM users under a different AWS account, or to an AWS service such as EC2.

Q: How do I assume an IAM role?

You assume an IAM role by calling the AWS Security Token Service (STS) AssumeRole APIs (in other words, AssumeRole, AssumeRoleWithWebIdentity, and AssumeRoleWithSAML). These APIs return a set of temporary security credentials that applications can then use to sign requests to AWS service APIs.

Q: What is the difference between an IAM role and an IAM user?

An IAM user has permanent long-term credentials and is used to directly interact with AWS services. An IAM role does not have any credentials and cannot make direct requests to AWS services. IAM roles are meant to be assumed by authorized entities, such as IAM users, applications, or an AWS service such as EC2.

Q: When should I use an IAM user, IAM group, or IAM role?

An IAM user has permanent long-term credentials and is used to directly interact with AWS services. An IAM group is primarily a management convenience to manage the same set of permissions for a set of IAM users. An IAM role is an AWS Identity and Access Management (IAM) entity with permissions to make AWS service requests. IAM roles cannot make direct requests to AWS services; they are meant to be assumed by authorized entities, such as IAM users, applications, or AWS services such as EC2. Use IAM roles to delegate access within or between AWS accounts.

Q: What is IAM roles for EC2 instances?

IAM roles for EC2 instances enables your applications running on EC2 to make requests to AWS services such as Amazon S3, Amazon SQS, and Amazon SNS without you having to copy AWS access keys to every instance.

Q: What problem does IAM roles for EC2 instances solve?

IAM roles for EC2 instances simplifies management and deployment of AWS access keys to EC2 instances. Using this feature, you associate an IAM role with an instance. Then your EC2 instance provides the temporary security credentials to applications running on the instance, and the applications can use these credentials to make requests securely to the AWS service resources defined in the role.

Q: Can I change the IAM role on a running EC2 instance?

Yes. Although a role is usually assigned to an EC2 instance when you launch it, a role can also be assigned to an EC2 instance that is already running.

Q: Can I associate an IAM role with an Auto Scaling group?

Yes. You can add an IAM role as an additional parameter in an Auto Scaling launch configuration and create an Auto Scaling group with that launch configuration. All EC2 instances launched in an Auto Scaling group that is associated with an IAM role are launched with the role as an input parameter.

Q: What is a service-linked role?

A [service-linked role](https://docs.aws.amazon.com/console/iam/service-linked-role) is a type of role that links to an AWS service (also known as a linked service) such that only the linked service can assume the role. Using these roles, you can delegate permissions to AWS services to create and manage AWS resources on your behalf.

Q: Can I delete a service-linked role?

Yes. If you no longer want an AWS service to perform actions on your behalf, you can delete its service-linked role. Before you delete the role, you must delete all AWS resources that depend on the role. This step ensures that you do not inadvertently delete a role required for your AWS resources to function properly.

Q: How do permissions work?

Access control policies are attached to users, groups, and roles to assign permissions to AWS resources. By default, IAM users, groups, and roles have no permissions; users with sufficient permissions must use a policy to grant the desired permissions.

Q: How do I assign permissions using a policy?

To set permissions, you can create and attach policies using the AWS Management Console, the IAM API, or the AWS CLI. Users who have been granted the necessary permissions can create policies and assign them to IAM users, groups, and roles.

Q: What are managed policies?

Managed policies are IAM resources that express permissions using the IAM policy language. You can create, edit, and manage separately from the IAM users, groups, and roles to which they are attached. After you attach a managed policy to multiple IAM users, groups, or roles, you can update that policy in one place and the permissions automatically extend to all attached entities. Managed policies are managed either by you (these are called customer managed policies) or by AWS (these are called AWS managed policies).

Q: How do I assign commonly used permissions?

AWS provides a set of commonly used permissions that you can attach to IAM users, groups, and roles in your account. These are called AWS managed policies. One example is read-only access for Amazon S3. When AWS updates these policies, the permissions are applied automatically to the users, groups, and roles to which the policy is attached. AWS managed policies automatically appear in the Policiessection of the IAM console. When you assign permissions, you can use an AWS managed policy or you can create your own customer managed policy. Create a new policy based on an existing AWS managed policy, or define your own.

Q: What is the difference between assigning permissions using IAM groups and assigning permissions using managed policies?

Use IAM groups to collect IAM users and define common permissions for those users. Use managed policies to share permissions across IAM users, groups, and roles. For example, if you want a group of users to be able to launch an Amazon EC2 instance, and you also want the role on that instance to have the same permissions as the users in the group, you can create a managed policy and assign it to the group of users and the role on the Amazon EC2 instance.

Q: What is the IAM policy simulator?

The IAM policy simulator is a tool to help you understand, test, and validate the effects of your access control policies.

Q: What are the benefits of temporary security credentials?

Temporary security credentials allow you to:

* Extend your internal user directories to enable federation to AWS, enabling your employees and applications to securely access AWS service APIs without needing to create an AWS identity for them.
* Request temporary security credentials for an unlimited number of federated users.
* Configure the time period after which temporary security credentials expire, offering improved security when accessing AWS service APIs through mobile devices where there is a risk of losing the device.

Q: What is identity federation?

AWS Identity and Access Management (IAM) supports identity federation for delegated access to the AWS Management Console or AWS APIs. With identity federation, external identities are granted secure access to resources in your AWS account without having to create IAM users. These external identities can come from your corporate identity provider (such as Microsoft Active Directory or from the AWS Directory Service) or from a web identity provider (such as [Amazon Cognito](https://aws.amazon.com/cognito/), [Login with Amazon](http://login.amazon.com/), [Facebook](https://www.facebook.com/about/login), [Google](https://developers.google.com/+/), or any [OpenID Connect](http://openid.net/connect/)-compatible provider).

Q: What are federated users?

Federated users (external identities) are users you manage outside of AWS in your corporate directory, but to whom you grant access to your AWS account using temporary security credentials. They differ from IAM users, which are created and maintained in your AWS account.

Q: Do you support SAML?

Yes, AWS supports the Security Assertion Markup Language (SAML) 2.0.

Q: Can federated users access AWS APIs?

Yes. You can programmatically request temporary security credentials for your federated users to provide them secure and direct access to AWS APIs.

Q: How do I control what a federated user is allowed to do when signed in to the console?

When you request temporary security credentials for your federated user using an AssumeRole API, you can optionally include an access policy with the request. The federated user’s privileges are the intersection of permissions granted by the access policy passed with the request and the access policy attached to the IAM role that was assumed. The access policy passed with the request cannot elevate the privileges associated with the IAM role being assumed. When you request temporary security credentials for your federated user using the GetFederationToken API, you must provide an access control policy with the request. The federated user’s privileges are the intersection of the permissions granted by the access policy passed with the request and the access policy attached to the IAM user that was used to make the request. The access policy passed with the request cannot elevate the privileges associated with the IAM user used to make the request. These federated user permissions apply to both API access and actions taken within the AWS Management Console.

Q: What is web identity federation?

Web identity federation allows you to create AWS-powered mobile apps that use public identity providers (such as [Amazon Cognito](https://aws.amazon.com/cognito/), [Login](http://login.amazon.com/) with Amazon, [Facebook](https://www.facebook.com/about/login), [Google](https://developers.google.com/+/), or any [OpenID Connect](http://openid.net/connect/)-compatible provider) for authentication. With web identity federation, you have an easy way to integrate sign-in from public identity providers (IdPs) into your apps without having to write any server-side code and without distributing long-term AWS security credentials with the app.

Q. What is AWS MFA?

AWS multi-factor authentication (AWS MFA) provides an extra level of security that you can apply to your AWS environment. You can enable AWS MFA for your AWS account and for individual AWS Identity and Access Management (IAM) users you create under your account.

**Amazon Cognito**

Q: What is Amazon Cognito?

Amazon Cognito lets you easily add user sign-up and authentication to your mobile and web apps. Amazon Cognito also enables you to authenticate users through an external identity provider and provides temporary security credentials to access your app’s backend resources in AWS or any service behind Amazon API Gateway. Amazon Cognito works with external identity providers that support SAML or OpenID Connect, social identity providers (such as Facebook, Twitter, Amazon) and you can also integrate your own identity provider.

In addition, Amazon Cognito enables you to synchronize data across a user’s devices so that their app experience remains consistent when they switch between devices or upgrade to a new device. Your app can save data locally on users’ devices allowing your applications to work even when the devices are offline and then automatically synchronize the data when the device is back online.

With Amazon Cognito, you can focus on creating great app experiences instead of worrying about building, securing, and scaling a solution to handle user management, authentication, and sync across platforms and devices.

Q: Who should use Amazon Cognito?

Amazon Cognito is designed for developers who want to add user management and sync functionality to their mobile and web apps. Developers can use Cognito Identity to add sign-up and sign-in to their apps and to enable their users to securely access their app’s resources. Cognito also enables developers to sync data across devices, platforms, and applications.

Q: Can I have my own identity provider to support user sign-up and sign-in?

Yes, you can easily and securely add sign-up and sign-in functionality to your apps with Cognito Identity. Your users can sign-up and sign-in using email, phone number, or user name. You can also implement enhanced security features, such as email verification, phone number verification, and multi-factor authentication. Cognito Identity also enables you to customize workflows by, for example, adding app-specific logic to user registration for fraud detection and user validation through AWS Lambda.

Q: What is a User Pool?

A User Pool is your user directory that you can configure for your web and mobile apps. A User Pool securely stores your users’ profile attributes. You can create and manage a User Pool using the AWS console, AWS CLI, or AWS SDK.

Q: Is it possible to customize user sign-up and sign-in workflows?

Yes, you can customize sign-up and sign-in by adding app-specific logic to the user sign-up and sign-in flows using AWS Lambda. For example, you can create AWS Lambda functions to identify fraud or perform additional validations on user data. You are able to trigger developer-provided Lambda functions at pre-registration, at post-confirmation, at pre-authentication, during authentication to customize the challenges, and at post-authentication. You can also use Lambda functions to customize messages sent as part of email or phone number verification and multi-factor authentication.

Q: Can I use Cognito Identity to federate identities and secure access to AWS resources?

Yes, Cognito Identity enables you to authenticate users through an external identity provider and provides temporary security credentials to access your app’s backend resources in AWS or any service behind Amazon API Gateway. Amazon Cognito works with external identity providers that support SAML or OpenID Connect, social identity providers (such as Facebook, Twitter, Amazon) and you can also integrate your own identity provider.

Q: Which public identity providers can I use with Amazon Cognito Identity?

You can use Amazon, Facebook, Twitter, Digits, Google and any other OpenID Connect compatible identity provider.

Q: Can I register and authenticate my own users?

Cognito Identity can integrate with your existing authentication system. With a simple API call you can retrieve a Cognito ID for your end users based on your own unique identifier for your users. Once you have retrieved the Cognito ID and OpenID Token Cognito Identity provides, you can use the Cognito Identity client SDK to access AWS resources and synchronize user data. Cognito Identity is a fully managed identity provider to make it easier for you to implement user sign-up and sign-in for your mobile and web apps.

Q: How does Cognito Identity help me control permissions and access AWS services securely?

Cognito Identity assigns your users a set of temporary, limited privilege credentials to access your AWS resources so you do not have to use your AWS account credentials. The permissions for each user are controlled through AWS IAM roles that you create. You can define rules to choose the IAM role for each user, or if you are using groups in a Cognito user pool, you can assign IAM roles based on groups. Cognito Identity also allows you to define a separate IAM role with limited permissions for guest users who are not authenticated. In addition, you can use the unique identifier that Cognito generates for your users to control access to specific resources. For example you can create a policy for an S3 bucket that only allows each user access to their own folder within the bucket.

Q: What is the Amazon Cognito sync store?

The Amazon Cognito Sync store is a key/value pair store linked to an Amazon Cognito identity. There is no limit to the number of identities you can create in your identity pools and sync store. Each Amazon Cognito identity within the sync store has its own user information store.

Q: Can I validate data before it is saved?

Amazon Cognito Events allows developers to run an AWS Lambda function in response to important events in Cognito. The Sync Trigger event is an event that occurs when any dataset is synchronized. Developers can write an AWS Lambda function to intercept the synchronization event. The function can evaluate the changes to the underlying Dataset and manipulate the data before it is stored in the cloud and synchronized back to the user's other devices. Alternatively, the AWS Lambda function could fail the sync operation so that the data is not synchronized to the user's other devices.

Q: How is data synchronized with Amazon Cognito?

You can programmatically trigger the sync of data sets between client devices and the Amazon Cognito sync store by using the synchronize() method in the AWS Mobile SDK. The synchronize() method reads the latest version of the data available in the Amazon Cognito sync store and compares it to the local, cached copy. After comparison, the synchronize() method writes the latest updates as necessary to the local data store and the Amazon Cognito sync store. By default Amazon Cognito maintains the last-written version of the data. You can override this behavior and resolve data conflicts programmatically. In addition, push synchronization allows you to use Amazon Cognito to send a silent push notification to all devices associated with an identity to notify them that new data is available.

Basically, a sync operation compares the local data store on a device to the Amazon Cognito sync store in the cloud and synchronizes the two data stores.

**AWS WAF**

Q: What is AWS WAF?

AWS WAF is a web application firewall that helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define. These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection and cross-site scripting.

Q: How does AWS WAF block or allow traffic?

As the underlying service receives requests for your web sites, it forwards those requests to AWS WAF for inspection against your rules. Once a request meets a condition defined in your rules, AWS WAF instructs the underlying service to either block or allow the request based on the action you define.

Q: How does AWS WAF protect my web site or application?

AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end users. This means security doesn’t come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on Application Load Balancer, your rules run in region and can be used to protect internet-facing as well as internal load balancers.

Q: What types of attacks can AWS WAF help me to stop?

AWS WAF helps protects your website from common attack techniques like SQL injection and Cross-Site Scripting (XSS). In addition, you can create rules that can block attacks from specific user-agents, bad bots, or content scrapers.