AWS IAM FAQs

General

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 permissionsfor those IAM users to access your resources. You can also grant permissions for users outside of AWS (federated users).

Q: How do I get started with IAM?

To start using IAM, you must subscribe to at least one of the AWS services that is integrated with IAM. You then can create and manage users, groups, and permissions via IAM APIs, the AWS CLI, or the IAM console, which gives you a point-and-click, web-based interface. You can also use the visual editor to create policies.

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 devices), or request temporary security credentials 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: 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: Who can use IAM?

Any AWS customer can use IAM. The service is offered at no additional charge. You will be charged only for the use of other AWS services by your users.

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: How do users call AWS services?

Users can make requests to AWS services using security credentials. Explicit permissions govern a user's ability to call AWS services. By default, users have no ability to call service APIs on behalf of the account.

IAM user management

Q: How are IAM users managed?

IAM supports multiple methods to:

- Create and manage IAM users.
- Create and manage IAM groups.
- Manage users' security credentials.
- Create and manage policies to grant access to AWS services and resources.

You can create and manage users, groups, and policies by using IAM APIs, the AWS CLI, or the IAM console. You also can use the visual editor and the IAM policy simulator to create and test policies.

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.

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 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. For details, see Temporary Security Credentials in the IAM documentation.

Q: Which AWS services support IAM users?
You can find the complete list of AWS services that support IAM users in the AWS Services That Work with IAM section of the IAM documentation. AWS plans to add support for other services over time.

Q: Can I enable and disable user access?
Yes. You can enable and disable an IAM user's access keys via the IAM APIs, AWS CLI, or IAM console. If you disable the access keys, the user cannot programmatically access AWS services.

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, or IAM console.

Q: Can I structure a collection of users in a hierarchical way, such as in LDAP?
Yes. You can organize users and groups under paths, similar to object paths in Amazon S3—for example /mycompany/division/project/joe.

Q: Can I define users regionally?
Not initially. Users are global entities, like an AWS account is today. No region is required to be specified when you define user permissions. Users can use AWS services in any geographic region.

Q: How are MFA devices configured for IAM users?
You (the AWS account holder) can order multiple MFA devices. You can then assign these devices to individual IAM users via the IAM APIs, AWS CLI, or IAM console.

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: Where can I use my SSH keys?

Currently, IAM users can use their SSH keys only with AWS CodeCommit to access their repositories.

Q: Do IAM user names have to be email addresses?

No, but they can be. User names are just ASCII strings that are unique within a given AWS account. You can assign names using any naming convention you choose, including email addresses.

Q: Which character sets can I use for IAM user names?

You can only use ASCII characters for IAM entities.

Q: Are user attributes other than user name supported?

Not at this time.

Q: How are user passwords set?

You can set an initial password for an IAM user via the IAM console, AWS CLI, or IAM APIs. User passwords never appear in clear text after the initial provisioning, and are never displayed or returned via an API call. IAM users can manage their passwords via the My Password page in the IAM console. Users access this page by selecting the Security Credentials option from the drop-down list in the upper right corner of the AWS Management Console.

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. For details, see Setting an Account Policy Password for IAM Users.

Q: Can I set usage quotas on IAM users?

No. All limits are on the AWS account as a whole. For example, if your AWS account has a limit of 20 Amazon EC2 instances, IAM users with EC2 permissions can start instances up to the limit. You cannot limit what an individual user can do.

IAM role management

Q: What is an IAM role?

An IAM role is an IAM entity that defines a set of 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 get started with IAM roles?
You create a role in a way similar to how you create a user—name the role and attach a policy to it. For details, see Creating IAM Roles.

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: How many IAM roles can I assume?
There is no limit to the number of IAM roles you can assume, but you can only act as one IAM role when making requests to AWS services.

Q: Who can use IAM roles?
Any AWS customer can use IAM roles.

Q: How much do IAM roles cost?
IAM roles are free of charge. You will continue to pay for any resources a role in your AWS account consumes.

Q: How are IAM roles managed?
You can create and manage IAM roles via the IAM APIs, AWS CLI, or IAM console, which gives you a point-and-click, web-based interface.

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: Can I add an IAM role to an IAM group?
Not at this time.

Q: How many policies can I attach to an IAM role?
For inline policies: You can add as many inline policies as you want to a user, role, or group, but the total aggregate policy size (the sum size of all inline policies) per entity cannot exceed the following limits:

- User policy size cannot exceed 2,048 characters.
- Role policy size cannot exceed 10,240 characters.
- Group policy size cannot exceed 5,120 characters.

For managed policies: You can add up to 10 managed policies to a user, role, or group. The size of each managed policy cannot exceed 6,144 characters.

Q: How many IAM roles can I create?
You are limited to 1,000 IAM roles under your AWS account. If you need more roles, submit the IAM limit increase request form with your use case, and we will consider your request.

Q: To which services can my application make requests?
Your application can make requests to all AWS services that support role sessions.

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. For details, see IAM Roles for Amazon EC2.

Q: What are the features of IAM roles for EC2 instances?
IAM roles for EC2 instances provides the following features:

- AWS temporary security credentials to use when making requests from running EC2 instances to AWS services.
- Automatic rotation of the AWS temporary security credentials.
- Granular AWS service permissions for applications running on EC2 instances.

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: How do I get started with IAM roles for EC2 instances?
To understand how roles work with EC2 instances, you need to use the IAM console to create a role, launch an EC2 instance that uses that role, and then examine the running instance. You can examine the instance metadata to see how the role credentials are made available to an instance. You can also see how an application that runs on an instance can use the role. For more details, see How Do I Get Started?

Q: Can I use the same IAM role on multiple EC2 instances?
Yes.

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. To learn how to assign a role to a running instance, see IAM Roles for Amazon EC2. You can also change the permissions on the IAM role associated with a running instance, and the updated permissions take effect almost immediately.

Q: Can I associate an IAM role with an already running EC2 instance?
Yes. You can assign a role to an EC2 instance that is already running. To learn how to assign a role to an already running instance, see IAM Roles for Amazon EC2.

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. For more details, see What Is Auto Scaling?in the Auto Scaling Developer Guide.

Q: Can I associate more than one IAM role with an EC2 instance?
No. You can only associate one IAM role with an EC2 instance at this time. This limit of one role per instance cannot be increased.

Q: What happens if I delete an IAM role that is associated with a running EC2 instance?
Any application running on the instance that is using the role will be denied access immediately.

Q: Can I control which IAM roles an IAM user can associate with an EC2 instance?
Yes. For details, see Permissions Required for Using Roles with Amazon EC2.

Q: Which permissions are required to launch EC2 instances with an IAM role?
You must grant an IAM user two distinct permissions to successfully launch EC2 instances with roles:

- Permission to launch EC2 instances.
- Permission to associate an IAM role with EC2 instances.

For details, see Permissions Required for Using Roles with Amazon EC2.

Q: Who can access the access keys on an EC2 instance?
Any local user on the instance can access the access keys associated with the IAM role.

Q: How do I use the IAM role with my application on the EC2 instance?
If you develop your application with the AWS SDK, the AWS SDK automatically uses the AWS access keys that have been made available on the EC2 instance. If you are not using the AWS SDK, you can retrieve the access keys from the EC2 instance metadata service. For details, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances.

Q: How do I rotate the temporary security credentials on the EC2 instance?
The AWS temporary security credentials associated with an IAM role are automatically rotated multiple times a day. New temporary security credentials are made available no later than five minutes before the existing temporary security credentials expire.

Q: Can I use IAM roles for EC2 instances with any instance type or Amazon Machine Image?
Yes. IAM roles for EC2 instances also work in Amazon Virtual Private Cloud (VPC), with spot and reserved instances.

Q: What is a service-linked role?
A 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 assume a service-linked role?
No. A service-linked role can be assumed only by the linked service. This is the reason why the trust policy of a service-linked role cannot be modified.

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 I delete a service-linked role?
You can delete a service-linked role from the IAM console. Choose Roles in the navigation pane, choose the service-linked role that you want to delete, and choose Delete role. (Note: For Amazon Lex, you must use the Amazon Lex console to delete the service-linked role.)

Permissions

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). For more information about managed policies, see Managed Policies and Inline Policies.

Q: How do I create a customer managed policy?

You can use the visual editor or the JSON editor in the IAM console. The visual editor is a point-and-click editor that guides you through the process of granting permissions in a policy without requiring you to write the policy in JSON. You can create policies in JSON by using the CLI and SDK.

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 Policies section 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: How do group-based permissions work?
Use IAM groups to assign the same set of permissions to multiple IAM users. A user can also have individual permissions assigned to them. The two ways to attach permissions to users work together to set overall permissions.

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: How are IAM policies evaluated in conjunction with Amazon S3, Amazon SQS, Amazon SNS, and AWS KMS resource-based policies?
IAM policies are evaluated together with the service's resource-based policies. When a policy of any type grants access (without explicitly denying it), the action is allowed. For more information about the policy evaluation logic, see IAM Policy Evaluation Logic.

Q: Can I use a managed policy as a resource-based policy?
Managed policies can only be attached to IAM users, groups, or roles. You cannot use them as resource-based policies.

Q: How do I set granular permissions using policies?
Using policies, you can specify several layers of permission granularity. First, you can define specific AWS service actions you wish to allow or explicitly deny access to. Second, depending on the action, you can define specific AWS resources the actions can be performed on. Third, you can define conditions to specify when the policy is in effect (for example, if MFA is enabled or not).

Q: How can I easily remove unnecessary permissions?

To help you determine which permissions are needed, the IAM console now displays service last accessed data that shows the hour when an IAM entity (a user, group, or role) last accessed an AWS service. Knowing if and when an IAM entity last exercised a permission can help you remove unnecessary permissions and tighten your IAM policies with less effort.

Q: Can I grant permissions to access or change account-level information (for example, payment instrument, contact email address, and billing history)?

Yes, you can delegate the ability for an IAM user or a federated user to view AWS billing data and modify AWS account information. For more information about controlling access to your billing information, see Controlling Access.

Q: Who can create and manage access keys in an AWS account?

Only the AWS account owner can manage the access keys for the root account. The account owner and IAM users or roles that have been granted the necessary permissions can manage access keys for IAM users.

Q: Can I grant permissions to access AWS resources owned by another AWS account?

Yes. Using IAM roles, IAM users and federated users can access resources in another AWS account via the AWS Management Console, the AWS CLI, or the APIs. See Manage IAM Roles for more information.

Q: What does a policy look like?

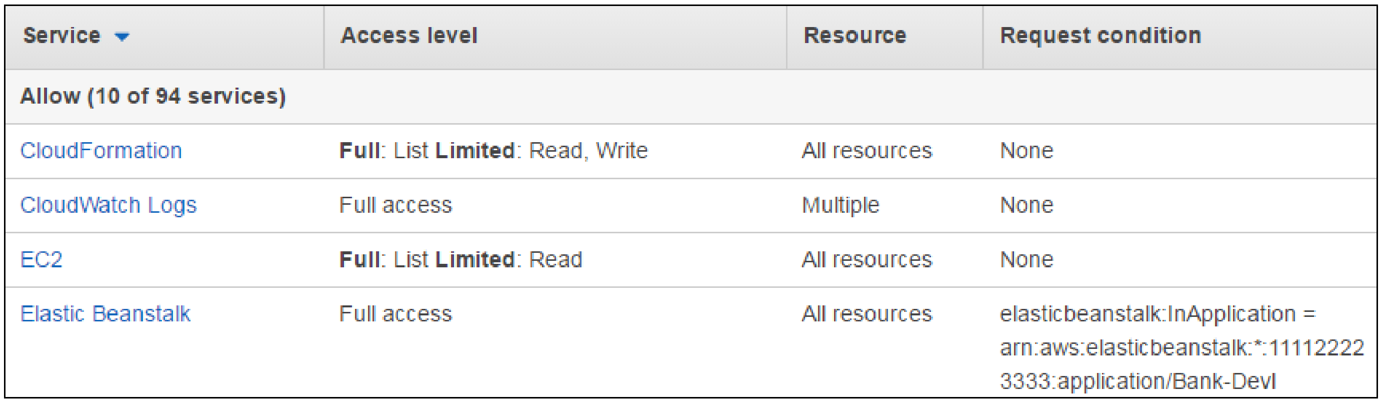
The following policy grants access to add, update, and delete objects from a specific folder, example_folder, in a specific bucket, example_bucket.

```
{
  "Version":"2012-10-17",
  "Statement":[
   {
     "Effect":"Allow",
     "Action":[
       "s3:PutObject",
       "s3:GetObject",
       "s3:GetObjectVersion",
       "s3:DeleteObject",
       "s3:DeleteObjectVersion"
     ],

     "Resource":"arn:aws:s3:::example_bucket/example_folder/*"
    }
  ]
}
```

Q: What is a policy summary?

If you are using the IAM console and choose a policy, you will see a policy summary. A policy summary lists the access level, resources, and conditions for each service defined in a policy (see the following screenshot for an example). The access level (View, Read, Write, or Permissions management) is defined by actions granted for each service in the policy. You can view the policy in JSON by choosing the JSON button.



Service	Access level	Resource	Request condition
Allow (10 of 94 services)			
CloudFormation	**Full**: List **Limited**: Read, Write	All resources	None
CloudWatch Logs	Full access	Multiple	None
EC2	**Full**: List **Limited**: Read	All resources	None
Elastic Beanstalk	Full access	All resources	elasticbeanstalk:InApplication = arn:aws:elasticbeanstalk:*:111122223333:application/Bank-Devl

Policy simulator

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 can the policy simulator be used for?
You can use the policy simulator in several ways. You can test policy changes to ensure they have the desired effect before committing them to production. You can validate existing policies attached to users, groups, and roles to verify and troubleshoot permissions. You can also use the policy simulator to understand how IAM policies and resource-based policies work together to grant or deny access to AWS resources.

Q: Who can use the policy simulator?
The policy simulator is available to all AWS customers.

Q: How much does the policy simulator cost?
The policy simulator is available at no extra cost.

Q: How do I get started?
Go to https://policysim.aws.amazon.com, or click the link on the IAM console under "Additional Information." Specify a new policy or choose an existing set of policies from a user, group, or role that you'd like to evaluate. Then select a set of actions from the list of AWS services, provide any required information to simulate the access request, and run the simulation to determine whether the policy allows or denies permissions to the selected actions and resources. To learn more about the IAM policy simulator, watch our Getting Started video or see the documentation.

Q: What kinds of policies does the IAM policy simulator support?
The policy simulator supports testing of newly entered policies and existing policies attached to users, groups, or roles. In addition, you can simulate whether resource-level policies grant access to a particular resource for Amazon S3 buckets, Amazon Glacier vaults, Amazon SNS topics,

and Amazon SQS queues. These are included in the simulation when an Amazon Resource Name (ARN) is specified in the Resource field in Simulation Settings for a service that supports resource policies.

Q: If I change a policy in the policy simulator, do those changes persist in production?
No. To apply changes to production, copy the policy that you've modified in the policy simulator and attach it to the desired IAM user, group, or role.

Q: Can I use the policy simulator programmatically?
Yes. You can use the policy simulator using the AWS SDKs or AWS CLI in addition to the policy simulator console. Use the iam:SimulatePrincipalPolicy API to programmatically test your existing IAM policies. To test the effects of new or updated policies that are not yet attached to a user, group, or role, call the iam:SimulateCustomPolicy API.

Signing in

Q: How does an IAM user sign in?
To sign in to the AWS Management Console as an IAM user, you must provide your account ID or account alias in addition to your user name and password. When your administrator created your IAM user in the console, they should have provided you with your user name and the URL to your account sign-in page. That URL includes your account ID or account alias.

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

You can also sign in at the following general sign-in endpoint and type your account ID or account alias manually:

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

For convenience, the AWS sign-in page uses a browser cookie to remember the IAM user name and account information. The next time the user goes to any page in the AWS Management Console, the console uses the cookie to redirect the user to the account sign-in page.

Note: IAM users can still use the URL link provided to them by their administrator to sign in to the AWS Management Console.

Q: What is an AWS account alias?
The account alias is a name you define to make it more convenient to identify your account. You can create an alias using the IAM APIs, AWS Command Line Tools, or the IAM console. You can have one alias per AWS account.

Q: Which AWS sites can IAM users access?
IAM users can sign in to the following AWS sites:

- AWS Management Console
- AWS Forums

- AWS Support Center
- AWS Marketplace

Q: Can IAM users sign in to other Amazon.com properties with their credentials?
No. Users created with IAM are recognized only by AWS services and applications.

Q: Is there an authentication API to verify IAM user sign-ins?
No. There is no programmatic way to verify user sign-ins.

Q: Can users SSH to EC2 instances using their AWS user name and password?
No. User security credentials created with IAM are not supported for direct authentication to customer EC2 instances. Managing EC2 SSH credentials is the customer's responsibility within the EC2 console.

Temporary security credentials

Q: What are temporary security credentials?
Temporary security credentials consist of the AWS access key ID, secret access key, and security token. Temporary security credentials are valid for a specified duration and for a specific set of permissions. Temporary security credentials are sometimes simply referred to as tokens. Tokens can be requested for IAM users or for federated users you manage in your own corporate directory. For more information, see Common Scenarios for Temporary Credentials.

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: How can I request temporary security credentials for federated users?
You can call the GetFederationToken, AssumeRole, AssumeRoleWithSAML, or AssumeRoleWithWebIdentity STS APIs.

Q: How can IAM users request temporary security credentials for their own use?
IAM users can request temporary security credentials for their own use by calling the AWS STS GetSessionToken API. The default expiration for these temporary credentials is 12 hours; the minimum is 15 minutes, and the maximum is 36 hours.
You can also use temporary credentials with Multi-Factor Authentication (MFA)-Protected API Access.

Q: How can I use temporary security credentials to call AWS service APIs?
If you're making direct HTTPS API requests to AWS, you can sign those requests with the temporary security credentials that you get from AWS Security Token Service (AWS STS). To do this, do the following:

- Use the access key ID and secret access key that are provided with the temporary security credentials the same way you would use long-term credentials to sign a request. For more information about signing HTTPS API requests, see Signing AWS API Requests in the AWS General Reference.
- Use the session token that is provided with the temporary security credentials. Include the session token in the "x-amz-security-token" header. See the following example request.
  - For Amazon S3, via the "x-amz- security-token" HTTP header.
  - For other AWS services, via the SecurityToken parameter.

Q: Which AWS services accept temporary security credentials?
For a list of supported services, see AWS Services That Work with IAM.

Q: What is the maximum size of the access policy that I can specify when requesting temporary security credentials (either GetFederationToken or AssumeRole)?
The policy plaintext must be 2048 bytes or shorter. However, an internal conversion compresses it into a packed binary format with a separate limit.

Q: Can a temporary security credential be revoked prior to its expiration?
No. When requesting temporary credentials, we recommend the following:

- When creating temporary security credentials, set the expiration to a value that is appropriate for your application.
- Because root account permissions cannot be restricted, use an IAM user and not the root account for creating temporary security credentials. You can revoke permissions of the IAM user that issued the original call to request it. This action almost immediately revokes privileges for all temporary security credentials issued by that IAM user

Q: Can I reactivate or extend the expiration of temporary security credentials?
No. It is a good practice to actively check the expiration and request a new temporary security credential before the old one expires. This rotation process is automatically managed for you when temporary security credentials are used in roles for EC2 instances.

Q: Are temporary security credentials supported in all regions?
Customers can request tokens from AWS STS endpoints in all regions, including AWS GovCloud (US) and China (Beijing) regions. Temporary credentials from AWS GovCloud (US) and China (Beijing) can be used only in the region from which they originated. Temporary credentials requested from any other region such as US East (N. Virginia) or EU (Ireland) can be used in all regions except AWS GovCloud (US) and China (Beijing).

Q: Can I restrict the use of temporary security credentials to a region or a subset of regions?
No. You cannot restrict the temporary security credentials to a particular region or subset of regions, except the temporary security credentials from AWS GovCloud (US) and China (Beijing), which can be used only in the respective regions from which they originated.

Q: What do I need to do before I can start using an AWS STS endpoint?
AWS STS endpoints are active by default in all regions and you can start using them without any further actions.

Q: What happens if I try to use a regional AWS STS endpoint that has been deactivated for my AWS account?
If you attempt to use a regional AWS STS endpoint that has been deactivated for your AWS account, you will see an AccessDenied exception from AWS STS with the following message: "AWS STS is not activated in this region for account: *AccountID*. Your account administrator can activate AWS STS in this region using the IAM console."

Q: What permissions are required to activate or deactivate AWS STS regions from the Account Settings page?
Only users with at least iam:* permissions can activate or deactivate AWS STS regions from the Account Settings page in the IAM console. Note that the AWS STS endpoints in US East (N. Virginia), AWS GovCloud (US), and China (Beijing) regions are always active and cannot be deactivated.

Q: Can I use the API or CLI to activate or deactivate AWS STS regions?
No. There is no API or CLI support at this time to activate or deactivate AWS STS regions. We plan to provide API and CLI support in a future release.

Identity federation

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, Login with Amazon,Facebook, Google, or any OpenID 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: What SAML profiles does AWS support?

The AWS single sign-on (SSO) endpoint supports the IdP-initiated HTTP-POST binding WebSSO SAML Profile. This enables a federated user to sign in to the AWS Management Console using a SAML assertion. A SAML assertion can also be used to request temporary security credentials using the AssumeRoleWithSAML API. For more information, see About SAML 2.0-Based Federation.

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. We have provided asample application that demonstrates how you can enable identity federation, providing users maintained by Microsoft Active Directory access to AWS service APIs. For more information, see Using Temporary Security Credentials to Request Access to AWS Resources.

Q: Can federated users access the AWS Management Console?

Yes. There are a couple ways to achieve this. One way is by programmatically requesting temporary security credentials (such as GetFederationToken or AssumeRole) for your federated users and including those credentials as part of the sign-in request to the AWS Management Console. After you have authenticated a user and granted them temporary security credentials, you generate a sign-in token that is used by the AWS single sign-on (SSO) endpoint. The user's actions in the console are limited to the access control policy associated with the temporary security credentials. For more details, see Creating a URL that Enables Federated Users to Access the AWS Management Console(Custom Federation Broker).
Alternatively, you can post a SAML assertion directly to AWS sign-in (https://signin.aws.amazon.com/saml). The user's actions in the console are limited to the access control policy associated with the IAM role that is assumed using the SAML assertion. For more details, see Enabling SAML 2.0 Federated Users to Access the AWS Management Console.
Using either approach allows a federated user to access the console without having to sign in with a user name and password. We have provided a sample application that demonstrates how you can enable identity federation, providing users maintained by Microsoft Active Directory access to the AWS Management Console.

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 permissions does a federated user need to use the console?

A user requires permissions to the AWS service APIs called by the AWS Management Console. Common permissions required to access AWS services are documented in Using Temporary Security Credentials to Request Access to AWS Resources.

Q: How do I control how long a federated user has access to the AWS Management Console?

Depending on the API used to create the temporary security credentials, you can specify a session limit between 15 minutes and 36 hours (for GetFederationToken and GetSessionToken) and between 15 minutes and 12 hours (for AssumeRole* APIs), during which time the federated user can access the console. When the session expires, the federated user must request a new session by returning to your identity provider, where you can grant them access again. Learn more about setting session duration.

Q: What happens when the identity federation console session times out?

The user is presented with a message stating that the console session has timed out and that they need to request a new session. You can specify a URL to direct users to your local intranet web page where they can request a new session. You add this URL when you specify an Issuer parameter as part of your sign-in request. For more information, see Enabling SAML 2.0 Federated Users to Access the AWS Management Console.

Q: How many federated users can I give access to the AWS Management Console?

There is no limit to the number of federated users who can be given access to the 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, Login with Amazon, Facebook, Google, or any OpenID 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.
For more information about web identity federation and to get started, see About Web Identity Federation.

Q: How do I enable web identity federation with accounts from public IdPs?

For best results, use Amazon Cognito as your identity broker for almost all web identity federation scenarios. Amazon Cognito is easy to use and provides additional capabilities such as anonymous (unauthenticated) access, and synchronizing user data across devices and providers. However, if you have already created an app that uses web identity federation by manually calling the AssumeRoleWithWebIdentity API, you can continue to use it and your apps will still work.
Here are the basic steps to enable identify federation using one of the supported web IdPs:

Sign up as a developer with the IdP and configure your app with the IdP, who gives you a unique ID for your app.
If you use an IdP that is compatible with OIDC, create an identity provider entity for it in IAM.
In AWS, create one or more IAM roles.
In your application, authenticate your users with the public IdP.
In your app, make an unsigned call to the AssumeRoleWithWebidentity API to request temporary security credentials.
Using the temporary security credentials you get in the AssumeRoleWithWebidentity response, your app makes signed requests to AWS APIs.
Your app caches the temporary security credentials so that you do not have to get new ones each time the app needs to make a request to AWS.
For more detailed steps, see Using Web Identity Federation APIs for Mobile Apps.

Q: How does identity federation using AWS Directory Service differ from using a third-party identity management solution?

If you want your federated users to be able to access only the AWS Management Console, using AWS Directory Service provides similar capabilities compared to using a third-party identity management solution. End users are able to sign in using their existing corporate credentials and access the AWS Management Console. Because AWS Directory Service is a managed service, customers do not need to set up or manage federation infrastructure, but rather need to create an AD Connector directory to integrate with their on-premises directory. If you are interested in providing your federated users access to AWS APIs, use a third-party offering, or deploy your own proxy server.

Billing

Q: Does AWS Billing provide aggregated usage and cost breakdowns by user?

No, this is not currently supported.

Q: Does the IAM service cost anything?

No, this is a feature of your AWS account provided at no additional charge.

Q: Who pays for usage incurred by users under an AWS Account?

The AWS account owner controls and is responsible for all usage, data, and resources under the account.

Q: Is billable user activity logged in AWS usage data?

Not currently. This is planned for a future release.

Q: How does IAM compare with Consolidated Billing?

IAM and Consolidated Billing are complementary features. Consolidated Billing enables you to consolidate payment for multiple AWS accounts within your company by designating a single paying account. The scope of IAM is not related to Consolidated Billing. A user exists within the confines of an AWS account and does not have permissions across linked accounts. For more details, see Paying Bills for Multiple Accounts Using Consolidated Billing.

Q: Can a user access the AWS accounts billing information?

Yes, but only if you let them. In order for IAM users to access billing information, you must first grant access to the Account Activity or Usage Reports. See Controlling Access.

Additional questions

Q: What happens if a user tries to access a service that has not yet been integrated with IAM?

The service returns an “Access denied” error.

Q: Are IAM actions logged for auditing purposes?

Yes. You can log IAM actions, STS actions, and AWS Management Console sign-ins by activating AWS CloudTrail. To learn more about AWS logging, see AWS CloudTrail.

Q: Is there any distinction between people and software agents as AWS entities?

No, both of these entities are treated like users with security credentials and permissions. However, people are the only ones to use a password in the AWS Management Console.

Q: Do users work with AWS Support Center and Trusted Advisor?

Yes, IAM users have the ability to create and modify support cases as well as use Trusted Advisor.

Q: Are there any default quota limits associated with IAM?

Yes, by default your AWS account has initial quotas set for all IAM-related entities. For details see Limitations on IAM Entities and Objects.

These quotas are subject to change. If you require an increase, you can access the Service Limit Increase form via the Contact Us page, and choose IAM Groups and Users from the Limit Type drop-down list.

Multi-factor authentication

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.

Q. How does AWS MFA work?

There are two primary ways to authenticate using an AWS MFA device:

- AWS Management Console users: When a user with MFA enabled signs in to an AWS website, they are prompted for their user name and password (the first factor–what they know), and an authentication response from their AWS MFA device (the second factor–what they have). All AWS websites that require sign-in, such as the AWS Management Console, fully support AWS MFA. You can also use AWS MFA together with Amazon S3 secure delete for additional protection of your S3 stored versions.

- AWS API users: You can enforce MFA authentication by adding MFA restrictions to your IAM policies. To access APIs and resources protected in this way, developers can request temporary security credentials and pass optional MFA parameters in their AWS Security Token Service (STS) API requests (the service that issues temporary security credentials). MFA-validated temporary security credentials can be used to call MFA-protected APIs and resources. Note: AWS STS and MFA-protected APIs do not currently support U2F security key as MFA.

Q. How do I help protect my AWS resources with MFA?
Follow two easy steps:

1. Get a MFA device. You have three options:

- Purchase a hardware YubiKey security key from Yubico, a third-party provider.
- Purchase a hardware device from Gemalto, a third-party provider.
- Install a virtual MFA–compatible application on a device such as your smartphone.
Visit the AWS MFA page for details about how to acquire a hardware or virtual MFA device.

2. After you have a MFA device, you must activate it in the IAM console. You can also use the AWS CLI to activate virtual MFA and hardware MFA (Gemalto device) for an IAM user. Note: AWS CLI does not currently support activation of U2F security keys.

Q. Is there a fee associated with using AWS MFA?
AWS does not charge any additional fees for using AWS MFA with your AWS account. However, if you want to use a physical MFA device then you will need to purchase the MFA device that is compatible with AWS MFA either from Gemalto or Yubico, third party providers. For more details, please visit Yubico or Gemalto's website.

Q. Can I have multiple MFA devices active for my AWS account?
Yes. Each IAM user can have its own MFA device. However, each identity (IAM user or root account) can be associated with only one MFA device.

Q. Can I use my U2F security key with multiple AWS accounts?

Yes. AWS allows you to use the same U2F security key with several root and IAM users across multiple accounts.

Q. Can I use virtual, hardware, or SMS MFA with multiple AWS accounts?
No. The MFA device or mobile phone number associated to virtual, hardware, and SMS MFA is bound to an individual AWS identity (IAM user or root account). If you have a TOTP-compatible application installed on your smartphone, you can create multiple virtual MFA devices on the same smartphone. Each one of the virtual MFA devices is bound to a single identity, just like hardware MFA (Gemalto) device. If you dissociate (deactivate) the MFA device, you can then reuse it with a different AWS identity. The MFA device associated to hardware MFA cannot currently be used by more than one identity simultaneously.

Q. I already have a hardware MFA device (Gemalto) from my place of work or from another service I use, can I re-use this device with AWS MFA?

No. AWS MFA relies on knowing a unique secret associated with your hardware MFA (Gemalto) device in order to support its use. Because of security constraints that mandate such secrets never be shared between multiple parties, AWS MFA cannot support the use of your existing Gemalto device. Only a compatible hardware MFA device purchased from Gemalto can be used with AWS MFA. You can re-use an existing U2F security key with AWS MFA, as U2F security keys do not share any secrets between multiple parties.

Purchasing an MFA Device

Q. I'm having a problem with an order for a MFA device using the third-party provider's website. Where can I get help?
Yubico or Gemalto's customer service can assist you.

Q. I received a defective or damaged MFA device from the third party provider. Where can I get help?
Yubico or Gemalto's customer service can assist you.

Q. I just received a MFA device from the third party provider. What should I do?
You simply need to activate the MFA device to enable AWS MFA for your AWS account. See the IAM console to perform this task.

Provisioning a Virtual MFA Device

Q. What is a virtual MFA device?
A virtual MFA device is an entry created in a TOTP compatible software application that can generate six-digit authentication codes. The software application can run on any compatible computing device, such as a smartphone.

Q. What are the differences between a virtual MFA device and physical MFA devices?
Virtual MFA devices use the same protocols as the physical MFA devices. Virtual MFA devices are software based and can run on your existing devices such as smartphones. Most virtual MFA applications also allow you to enable more than one virtual MFA device, which makes them more convenient than physical MFA devices.

Q. Which virtual MFA applications can I use with AWS MFA?
You can use applications that generate TOTP-compliant authentication codes, such as the Google Authenticator application, with AWS MFA. You can provision virtual MFA devices either automatically by scanning a QR code with the device's camera or by manual seed entry in the virtual MFA application.

Visit the MFA page for a list of supported virtual MFA applications.

Q. What is a QR code?
A QR code is a two-dimensional barcode that is readable by dedicated QR barcode readers and most smartphones. The code consists of black squares arranged in larger square patterns on a white background. The QR code contains the required security configuration information to provision a virtual MFA device in your virtual MFA application.

Q. How do I provision a new virtual MFA device?
You can configure a new virtual MFA device in the IAM console for your IAM users as well as for your AWS root account. You can also use the aws iam create-virtual-mfa-device command in the AWS CLI or the CreateVirtualMFADevice API to provision new virtual MFA devices under your account. The aws iam create-virtual-mfa-device and the CreateVirtualMFADevice API return the required configuration information, called a seed, to configure the virtual MFA device in your AWS MFA compatible application. You can either grant your IAM users the permissions to call this API directly or perform the initial provisioning for them.

Q. How should I handle and distribute the seed material for virtual MFA devices?

You should treat seed material like any other secret (for example the AWS secret keys and passwords).

Q. How can I enable an IAM user to manage virtual MFA devices under my account?
Grant the IAM user the permission to call the CreateVirtualMFADevice API. You can use this API to provision new virtual MFA devices.

SMS MFA

Q. Can I still request preview access to the SMS MFA?

We are no longer accepting new participants for the SMS MFA preview. We encourage you to use MFA on your AWS account by using a U2F security key, hardware device, or virtual (software-based) MFA device.

Q. When will the preview for SMS MFA end?

On February 1, 2019, AWS will no longer require IAM users to enter an MFA six-digit code if the IAM user is setup with “An SMS MFA device”. These users will also no longer be provided an SMS code when they sign in. We encourage you to use MFA through a U2F security key, hardware device, or virtual (software-based) MFA device. You can continue using this feature until January 31, 2019.

Enabling AWS MFA Devices

Q. Where do I enable AWS MFA?
You can enable AWS MFA for an AWS account and your IAM users in the IAM console, the AWS CLI, or by calling the AWS API. Note: AWS CLI and AWS API do not currently support enabling U2F security key.

Q. What information do I need to activate a hardware or virtual MFA device?
If you are activating the MFA device with the IAM console then you only need the device. If you are using the AWS CLI or the IAM API then you need the following:

1. The serial number of the MFA device. The format of the serial number depends on whether you are using a hardware device or a virtual device:

- Hardware MFA device: The serial number is on the bar-coded label on the back of the device.

- Virtual MFA device: The serial number is the Amazon Resource Name (ARN) value returned when you run the iam-virtualmfadevicecreate command in the AWS CLI or call the CreateVirtualMFADevice API.

2. Two consecutive MFA codes displayed by the MFA device.

Q. My MFA device seems to be working normally, but I am not able to activate it. What should I do?
Please contact us for help.

Using AWS MFA

Q. If I enable AWS MFA for my AWS root account or my IAM users, do they always have to use MFA to sign in to the AWS Management Console?
Yes. The AWS root credential user and IAM users must have their MFA device with them any time they need to sign in to any AWS website.

If your MFA device is lost, damaged, stolen, or not working, you can sign in using alternative factors of authentication, deactivate the MFA device, and activate a new device. As a security best practice, we recommend that you change your root account's password.

If your IAM users lose or damage their MFA device, or if it is stolen or stops working, you can disable AWS MFA yourself by using the IAM console or the AWS CLI.

Q. If I enable AWS MFA for my AWS root account or IAM users, do they always need to complete the MFA challenge to directly call AWS APIs?
No, it’s optional. However, you must complete the MFA challenge if you plan to call APIs that are secured by MFA-protected API access.

If you are calling AWS APIs using access keys for your AWS root account or IAM user, you do not need to enter an MFA code. For security reasons, we recommend that you remove all access keys from your AWS root account and instead call AWS APIs with the access keys for an IAM user that has the required permissions.

Note: U2F security keys currently do not work with MFA-protected APIs and currently cannot be used as MFA for AWS APIs.

Q. How do I sign in to the AWS Portal and AWS Management Console using my MFA device?
Follow these two steps:

1. If you are signing in as an AWS root account, sign in as usual with your user name and password when prompted. To sign in as an IAM user, use the account-specific URL and provide your user name and password when prompted.
2. If you have enabled virtual, hardware, or SMS MFA, enter the six-digit MFA code that appears on your MFA device. If you have enabled U2F security key, insert the key into the USB port of your computer, wait for the key to blink, and then touch the button or gold disk on your key.

Q. Does AWS MFA affect how I access AWS Service APIs?

AWS MFA changes the way IAM users access AWS Service APIs only if the account administrator(s) choose to enable MFA-protected API access. Administrators may enable this feature to add an extra layer of security over access to sensitive APIs by requiring that callers authenticate with an AWS MFA device. For more information, see the MFA-protected API access documentation in more detail.

Other exceptions include S3 PUT bucket versioning, GET bucket versioning, and DELETE object APIs, which allow you to require MFA authentication to delete or change the versioning state of your bucket. For more information see the S3 documentation discussing Configuring a Bucket with MFA Delete in more detail.

For all other cases, AWS MFA does not currently change the way you access AWS service APIs.

Note: U2F security keys currently do not work with MFA-protected APIs and currently cannot be used as MFA for AWS APIs.

Q. For virtual and hardware MFA, can I use a given MFA code more than once?

No. For security reasons, you can use each MFA code provided by your virtual and hardware MFA device only once.

Q. I was recently asked to resync my MFA device because my MFA codes were being rejected. Should I be concerned?

No, this can happen occasionally. Virtual and hardware MFA relies on the clock in your MFA device being in sync with the clock on our servers. Sometimes, these clocks can drift apart. If this happens, when you use the MFA device to sign in to access secure pages on the AWS website or the AWS Management Console, AWS automatically attempts to resync the MFA device by requesting that you provide two consecutive MFA codes (just as you did during activation).

U2F security keys do not go out of sync and do not need a resync.

Q. My MFA device seems to be working normally, but I am not able to use it to sign in to the AWS Management Console. What should I do?

If you are using virtual or hardware MFA, we suggest you resynchronize MFA devices for your IAM user's credentials. If you already tried to resync and are still having trouble signing in, you can sign in using alternate factors of authentication and reset your MFA device.

If you are using U2F security keys, you can sign in using alternate factors of authentication and reset your MFA device.

If you are still encountering issues, contact us for help.

Q. My MFA device is lost, damaged, stolen, or not working, and now I can’t sign in to the AWS Management Console. What should I do?

If your MFA device is associated with an AWS root account:

You can reset your MFA device on the AWS Management Console by first signing in with your password and then verifying the email address and phone number associated with your root account.

If your MFA device is lost, damaged, stolen or not working, you can sign in using alternative factors of authentication, deactivate the MFA device, and activate a new MFA device. As a security best practice, we recommend that you change your root account's password.

If you need a new MFA device, you can purchase a new MFA device from a third-party provider, Yubico or Gemalto, or provision a new virtual MFA device under your account by using the IAM console.

If you have tried the preceding approaches and are still having trouble signing in, contact AWS Support.

Q. How do I disable AWS MFA?

To disable AWS MFA for your AWS account, you can deactivate your MFA device using the Security Credentials page. To disable AWS MFA for your IAM users, you need to use the IAM console or the AWS CLI.

Q. Can I use AWS MFA in GovCloud?

Yes, you can use AWS virtual MFA and hardware MFA devices in GovCloud.

MFA-protected API access

Q. What is MFA-protected API access?

MFA-protected API access is optional functionality that lets account administrators enforce additional authentication for customer-specified APIs by requiring that users provide a second authentication factor in addition to a password. Specifically, it enables administrators to include conditions in their IAM policies that check for and require MFA authentication for access to selected APIs. Users making calls to those APIs must first get temporary credentials that indicate the user entered a valid MFA code.

Q. Can I use my U2F security key with MFA-protected APIs?

No. MFA-protected APIs currently do not support U2F security keys.

Q. What problem does MFA-protected API access solve?

Previously, customers could require MFA for access to the AWS Management Console, but could not enforce MFA requirements on developers and applications interacting directly with AWS service APIs. MFA-protected API access ensures that IAM policies are universally enforced regardless of access path. As a result, you can now develop your own application that uses AWS and prompts the user for MFA authentication before calling powerful APIs or accessing sensitive resources.

Q. How do I get started with MFA-protected API access?

You can get started in two simple steps:

1. Assign an MFA device to your IAM users. You can purchase a hardware key fob, or download a free TOTP-compatible application for your smartphone, tablet, or computer. See the MFA detail page for more information on AWS MFA devices.
2. Enable MFA-protected API access by creating permission policies for the IAM users and/or IAM groups from which you want to require MFA authentication. To learn more about access policy language syntax, see the access policy language documentation.

Q. How do developers and users access APIs and resources secured with MFA-protected API access?
Developers and users interact with MFA-protected API access both in the AWS Management Console and at the APIs.

In the AWS Management Console, any MFA-enabled IAM user must authenticate with their device to sign in. Users that do not have MFA do not receive access to MFA-protected APIs and resources.

At the API level, developers can integrate AWS MFA into their applications to prompt users to authenticate using their assigned MFA devices before calling powerful APIs or accessing sensitive resources. Developers enable this functionality by adding optional MFA parameters (serial number and MFA code) to requests to obtain temporary security credentials (such requests are also referred to as “session requests”). If the parameters are valid, temporary security credentials that indicate MFA status are returned. See the temporary security credentials documentation for more information.

Q. Who can use MFA-protected API access?
MFA-protected API access is available for free to all AWS customers.

Q. Which services does MFA-protected API access work with?
MFA-protected API access is supported by all AWS services that support temporary security credentials. For a list of supported services, see AWS Services that Work with IAMand review the column labeled Supports temporary security credentials.

Q. What happens if a user provides incorrect MFA device information when requesting temporary security credentials?
The request to issue temporary security credentials fails. Temporary security credential requests that specify MFA parameters must provide the correct serial number of the device linked to the IAM user as well as a valid MFA code.

Q. Does MFA-protected API access control API access for AWS root accounts?
No, MFA-protected API access only controls access for IAM users. Root accounts are not bound by IAM policies, which is why we recommend that you create IAM users to interact with AWS service APIs rather than use AWS root account credentials.

Q. Do users have to have an MFA device assigned to them in order to use MFA-protected API access?
Yes, a user must first be assigned a unique hardware or virtual MFA device.

Q. Is MFA-protected API access compatible with S3 objects, SQS queues, and SNS topics?
Yes.

Q. How does MFA-protected API access interact with existing MFA use cases such as S3 MFA Delete?
MFA-protected API access and S3 MFA Delete do not interact with each other. S3 MFA Delete currently does not support temporary security credentials. Instead, calls to the S3 MFA Delete API must be made using long-term access keys.

Q. Does MFA-protected API access work in the GovCloud (US) region?
Yes.

Q. Does MFA-protected API access work for federated users?
Customers cannot use MFA-protected API access to control access for federated users. The GetFederatedSession API does not accept MFA parameters. Since federated users can't authenticate with AWS MFA devices, they are unable to access resources designated using MFA-protected API access.

Pricing

Q. What will I be charged for using AWS IAM?

IAM is a feature of your AWS account offered at no additional charge. You will be charged only for the use of other AWS services by your users.