This document is published in the IAM User Guide: <https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html>

# Understanding How IAM Works

Before you create users, you should understand how IAM works. IAM provides the infrastructure necessary to control authentication and authorization for your account. The IAM infrastructure includes the following elements:

**Topics**

* [Principal](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-principal)
* [Request](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-request)
* [Authentication](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-authentication)
* [Authorization](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-authorization)
* [Actions or Operations](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-actions)
* [Resources](https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html#intro-structure-resources)



## Principal

A principal is an entity that can make a request for an action or operation on an AWS resource. Users, roles, federated users, and applications are all AWS principals. Your AWS account root user is your first principal. As a best practice, do not use your root user for your daily work. Instead, create IAM users and roles. You can also support federated users or programmatic access to allow an application to access your AWS account.

## Request

When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a request to AWS. The request includes the following information:

* **Actions or operations** – The actions or operations that the principal wants to perform. This can be an action in the AWS Management Console, or an operation in the AWS CLI or AWS API.
* **Resources** – The AWS resource object upon which the actions or operations are performed.
* **Principal** – The user, role, federated user, or application that sent the request. Information about the principal includes the policies that are associated with that principal.
* **Environment data** – Information about the IP address, user agent, SSL enabled status, or the time of day.
* **Resource data** – Data related to the resource that is being requested. This can include information such as a DynamoDB table name or a tag on an Amazon EC2 instance.

AWS gathers the request information into a request context, which is used to evaluate and authorize the request.

## Authentication

As a principal, you must be authenticated (signed in to AWS) to send a request to AWS. Although some services, such as Amazon S3 and AWS STS, allow a few requests from anonymous users, they are the exception to the rule.

To authenticate from the console as a user, you must sign in with your user name and password. To authenticate from the API or AWS CLI, you must provide your access key and secret key. You might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more about the IAM identities that AWS can authenticate, see [Identities (Users, Groups, and Roles)](https://docs.aws.amazon.com/IAM/latest/UserGuide/id.html).

## Authorization

During authorization, AWS uses values from the request context to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as [JSON documents](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#access_policies-json) and specify the permissions that are allowed or denied for principals. There are [several types of policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html) that can affect whether a request is authorized. Those policies types can be categorized as permissions policies or permissions boundaries. Permissions policies define the permissions for the object to which they’re attached. These include identity-based policies (the most common), resource-based policies, and ACLs. A permissions boundary is an advanced feature that allows you to use policies to limit the maximum permissions that a principal can have. These boundaries can be applied to AWS Organizations organizations or to IAM users or roles.

To provide your users with permissions to access the AWS resources in their own account, you need only identity-based policies. Resource-based policies are popular for granting [cross-account access](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_permissions-required.html#UserPermissionsAcrossAccounts). The other policy types are advanced features and should be used carefully.

AWS checks each policy that applies to the context of your request. If a single policy includes a denied action, AWS denies the entire request and stops evaluating. This is called an explicit deny. Because requests are denied by default, AWS authorizes your request only if every part of your request is allowed by the applicable policies. The evaluation logic follows these rules:

* By default, all requests are denied. (In general, requests made using the AWS account root user credentials for resources in the account are always allowed.)
* An explicit allow in a permissions policy overrides this default.
* A permissions boundary (AWS Organizations SCP or user or role boundary) overrides the allow. If there is a permissions boundary that applies, that boundary must allow the request. Otherwise, it is implicitly denied.
* An explicit deny in any policy overrides any allows.

## Actions or Operations

After your request has been authenticated and authorized, AWS approves the actions or operations in your request. Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource. For example, IAM supports about 40 actions for a user resource, including the following actions:

* CreateUser
* DeleteUser
* GetUser
* UpdateUser

To allow a principal to perform an operation, you must include the necessary actions in a policy that applies to the principal or the affected resource. To see a list of actions, resource types, and condition keys supported by each service, see [Actions, Resources, and Condition Keys for AWS Services](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_actions-resources-contextkeys.html).

## Resources

After AWS approves the operations in your request, they can be performed on the related resources within your account. A resource is an object that exists within a service. Examples include an Amazon EC2 instance, an IAM user, and an Amazon S3 bucket. The service defines a set of actions that can be performed on each resource. If you create a request to perform an unrelated action on a resource, that request is denied. For example, if you request to delete an IAM role but provide an IAM group resource, the request fails. To see AWS service tables that identify which resources are affected by an action, see [Actions, Resources, and Condition Keys for AWS Services](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_actions-resources-contextkeys.html).