When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account.

Multi-factor authentication (MFA)

You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device.

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.

Permissions are categorized as permissions policies and permissions boundaries. Most permission policies are JSON policy documents in AWS that, when attached to an identity or resource, define their permissions. A permissions boundary is an advanced feature that allows you to use policies to limit the maximum permissions that a principal can have.

In Amazon Elastic Compute Cloud you log into an instance with a key pair (for Linux instances) or using a user name and password (for Microsoft Windows instances).

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.

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.

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. 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: 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.

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. You are limited to 1,000 IAM roles under your AWS account.

Grant Least Privilege

When you create IAM policies, follow the standard security advice of granting least privilege—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform only those tasks.

Policy Types

The following policy types, listed in order of frequency, are available for use in AWS. For more details, see the sections below for each policy type.

Identity-based policies – Attach managed and inline policies to IAM identities, such as users, groups to which users belong, and roles.

Resource-based policies – Attach inline policies to resources. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies.

Organizations SCPs – Use an AWS Organizations service control policy (SCP) to apply a permissions boundary to an AWS Organizations organization or organizational unit (OU).

Access control lists (ACLs) – Use ACLs to control what principals can access a resource. ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document structure.

When you create a new user, that user Will be able to interact with AWS using their access key ID and secret access key using the API, CLI, or the AWS SDKs.

Security Assertion Markup Language 2.0 (SAML 2.0) is a version of the SAML standard for exchanging authentication and authorization data between security domains. SAML 2.0 is an XML-based protocol that uses security tokens containing assertions to pass information about a principal (usually an end user) between a SAML authority, named an Identity Provider, and a SAML consumer, named a Service Provider. SAML 2.0 enables web-based, cross-domain single sign-on (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to the user.

Users and Policy Documents only once, as these are applied globally.

IAM account signin URL is different from the Root account signin URL