# AWS Organizations



AWS Organizations helps you centrally manage and govern your environment as you grow and scale your AWS resources.

AWS accounts are natural boundaries for permissions, security, costs, and workloads.

Using a multi-account environment is a recommended best-practice when scaling your cloud environment.

AWS Organizations provides many features for managing multi-account environments, including:

- Simplify account creation by programmatically creating new accounts using the AWS Command Line Interface (CLI), SDKs, or APIs.
- Group accounts into organizational units (OUs), or groups of accounts that serve a single application or service.
- Apply tag polices to classify or track resources in your organization and provide attribute-based access control for users or applications.
- Delegate responsibility for supported AWS services to accounts so users can manage them on behalf of your organization.
- Centrally provide tools and access for your security team to manage security needs on behalf of the organization.
- Set up Amazon Single Sign-On (SSO) to provide access to AWS accounts and resources using your active directory, and customize permissions based on separate job roles.
- Apply service control policies (SCPs) to users, accounts, or OUs to control access to AWS resources, services, and Regions within your organization.
- Share AWS resources within your organization using AWS Resource

Allocation Management (RAM).
- Activate AWS CloudTrail across accounts, which creates a log of all activity in your cloud environment that cannot be turned off or modified by member accounts.
- Organizations provides you with a single consolidated bill.
- In addition, you can view usage from resources across accounts and track costs using AWS Cost Explorer and optimize your usage of compute resources using AWS Compute Optimizer.

AWS Organizations is available to all AWS customers at no additional charge.

The **AWS Organizations API** enables automation for account creation and management.

AWS Organizations is available in two feature sets:

- Consolidated billing.
- All features.

By default, organizations support consolidated billing features.

Consolidated billing separates paying accounts and linked accounts.

You can use AWS Organizations to set up a single payment method for all the AWS accounts in your organization through consolidated billing.

With consolidated billing, you can see a combined view of charges incurred by all your accounts.

Can also take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

Limit of 20 linked accounts for consolidated billing (default).

Policies can be assigned at different points in the hierarchy.

Can help with cost control through volume discounts.

Unused reserved EC2 instances are applied across the group.

Paying accounts should be used for billing purposes only.

Billing alerts can be setup at the paying account which shows billing for all linked accounts.

## Key Concepts

Some of the key concepts you need to understand are listed here:

**AWS Organization –** An organization is a collection of AWS accounts that you can organize into a hierarchy and manage centrally.

**AWS Account –** An AWS account is a container for your AWS resources.

**Management Account –** A management account is the AWS account you use to create your organization.

**Member Account –** A member account is an AWS account, other than the management account, that is part of an organization.

**Administrative Root –** An administrative root is the starting point for organizing your AWS accounts. The administrative root is the top-most container in your organization's hierarchy.

**Organizational Unit (OU) –** An organizational unit (OU) is a group of AWS accounts within an organization. An OU can also contain other OUs enabling you to create a hierarchy.

**Policy –** A policy is a "document" with one or more statements that define the controls that you want to apply to a group of AWS accounts. AWS Organizations supports a specific type of policy called a Service Control Policy (SCP). An SCP defines the AWS service actions, such as Amazon EC2

RunInstances, that are available for use in different accounts within an organization.

Best practices for the management account:

- Use the management account only for tasks that require the management account.
- Use a group email address for the management account's root user.
- Use a complex password for the management account's root user.
- Enable MFA for your root user credentials.
- Add a phone number to the account contact information.
- Review and keep track of who has access.
- Document the processes for using the root user credentials.
- Apply controls to monitor access to the root user credentials.

## Migrating accounts between organizations

Accounts can be migrated between organizations.

You must have root or IAM access to both the member and management accounts.

Use the AWS Organizations console for just a few accounts.

Use the AWS Organizations API or AWS Command Line Interface (AWS CLI) if there are many accounts to migrate.

Billing history and billing reports for all accounts stay with the management account in an organization.

Before migration download any billing or report history for any member accounts that you want to keep.

When a member account leaves an organization, all charges incurred by the

account are charged directly to the standalone account.

Even if the account move only takes a minute to process, it is likely that some charges will be incurred by the member account.

## Service Control Policies (SCPs)

Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization.

SCPs offer central control over the maximum available permissions for all accounts in your organization. SCPs help you to ensure your accounts stay within your organization's access control guidelines.

SCPs are available only in an organization that has all features enabled.

SCPs aren't available if your organization has enabled only the consolidated billing features.

SCPs alone are not sufficient to granting permissions to the accounts in your organization.

No permissions are granted by an SCP. An SCP defines a guardrail, or sets limits, on the actions that the account's administrator can delegate to the IAM users and roles in the affected accounts.

The administrator must still attach identity-based or resource-based policies to IAM users or roles, or to the resources in your accounts to grant permissions.

The effective permissions are the logical intersection between what is allowed by the SCP and what is allowed by the IAM and resource-based policies.

SCP Inheritance:

- SCPs ***affect only IAM users and roles*** that are managed by accounts that are part of the organization. SCPs don't affect resource-based policies directly. They also don't affect users or roles from accounts outside the organization.
- An SCP restricts permissions for IAM users and roles in member accounts, including the member account's root user.
- Any account has only those permissions permitted by ***every*** parent above it.
- If a permission is blocked at any level above the account, either implicitly (by not being included in an Allow policy statement) or explicitly (by being included in a Deny policy statement), a user or role in the affected account can't use that permission, even if the account administrator attaches the AdministratorAccess IAM policy with */* permissions to the user.
- SCPs affect only ***member*** accounts in the organization. They have no effect on users or roles in the management account.
- Users and roles must still be granted permissions with appropriate IAM permission policies. A user without any IAM permission policies has no access, even if the applicable SCPs allow all services and all actions.
- If a user or role has an IAM permission policy that grants access to an action that is also allowed by the applicable SCPs, the user or role can perform that action.
- If a user or role has an IAM permission policy that grants access to an action that is either not allowed or explicitly denied by the applicable SCPs, the user or role can't perform that action.
- SCPs affect all users and roles in attached accounts, ***including the root user***. The only exceptions are those described in **Tasks and entities not restricted by SCPs**.
- SCPs ***do not*** affect any service-linked role. Service-linked roles enable other AWS services to integrate with AWS Organizations and can't be restricted by SCPs.
- When you disable the SCP policy type in a root, all SCPs are

automatically detached from all AWS Organizations entities in that root. AWS Organizations entities include organizational units, organizations, and accounts.
- If you reenable SCPs in a root, that root reverts to only the default FullAWSAccess policy automatically attached to all entities in the root.
- Any attachments of SCPs to AWS Organizations entities from before SCPs were disabled are lost and aren't automatically recoverable, although you can manually reattach them.
- If both a permissions boundary (an advanced IAM feature) and an SCP are present, then the boundary, the SCP, and the identity-based policy must all allow the action.

You ***can't*** use SCPs to restrict the following tasks:

- Any action performed by the management account.
- Any action performed using permissions that are attached to a service-linked role.
- Register for the Enterprise support plan as the root user.
- Change the AWS support level as the root user.
- Provide trusted signer functionality for CloudFront private content.
- Configure reverse DNS for an Amazon Lightsail email server as the root user.
- Tasks on some AWS-related services:
    - Alexa Top Sites.
    - Alexa Web Information Service.
    - Amazon Mechanical Turk.
    - Amazon Product Marketing API.

## Resource Groups

You can use resource groups to organize your AWS resources.

In AWS, a resource is an entity that you can work with.

Resource groups make it easier to manage and automate tasks on large numbers of resources at one time.

Resource groups allow you to group resources and then tag them.

The Tag Editor assists with finding resources and adding tags.

**You can access Resource Groups through any of the following entry points:**

- On the navigation bar of the AWS Management Console.
- In the AWS Systems Manager console, from the left navigation pane entry for Resource Groups.
- By using the Resource Groups API, in AWS CLI commands or AWS SDK programming languages.

A resource group is a collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.

In Resource Groups, there are two types of queries on which you can build a group.

Both query types include resources that are specified in the format AWS::service::resource.

- **Tag-based –** Tag-based queries include lists of resources and tags. Tags are keys that help identify and sort your resources within your organization. Optionally, tags include values for keys.
- **AWS CloudFormation stack-based –** In an AWS CloudFormation stack-based query, you choose an AWS CloudFormation stack in your account in the current region, and then choose resource types within the stack that you want to be in the group. You can base your query on only one AWS CloudFormation stack.

Resource groups can be nested; a resource group can contain existing

resource groups in the same region.