

AWS MULTIPLE ACCOUNT BILLING STRATEGY *“How do I manage multiple AWS accounts for billing purposes?”*

Overview

Amazon Web Services (AWS) enables customers to achieve significant gains in productivity, innovation, and cost reduction when they move to the AWS Cloud. AWS offers a variety of services and features that allow for flexible control of cloud computing resources and also of the AWS account(s) managing those resources. These options help to ensure proper cost allocation, agility, and security, however customers are sometimes unsure of how to best implement an account strategy—especially when working with multiple AWS accounts. This document provides best practices and high-level strategic guidance to help customers use AWS Organizations to structure and manage multiple AWS accounts for billing purposes.

The following sections assume basic knowledge of AWS Organizations, AWS service limits, resource groups and tagging, AWS Identity and Access Management (IAM), and Reserved Instances (RIs).

General Best Practices

Customers leverage AWS to increase speed and business agility, and so it is common for AWS account structures to change over time in response to a company's use of AWS. Therefore, when considering a billing strategy for multiple accounts, do not over-engineer your initial account structure in an attempt to create a perfect, immutable set of AWS accounts. Instead, take advantage of AWS's flexibility and use an iterative approach to creating and structuring your accounts. With adaptability in mind, consider these best practices for account strategies:

- Use a group alias rather than an individual email address as the account email address to ensure continuity of communication from AWS, regardless of the availability of the individual who creates or manages your account. This will also make it easier to identify and distinguish accounts from one another. Do this for all the notification emails configured on your account.
- Create and implement AWS tagging standards across your accounts to standardize how your company categorizes, controls access to, and reports on AWS resources (See the [AWS Tagging Strategies](#) Solution Brief for additional guidance). Verify compliance regularly, either manually in the console or with an automated script.
- Leverage AWS APIs and scripts to automatically and consistently apply your company's baseline configuration across all AWS accounts. Consider leveraging compliance-monitoring scripts to provide insight into how well your accounts comply with the policies and standards you defined for your company's use of AWS.

Implementation Considerations

The following sections offer advice and guidelines to help identify the appropriate account structure for a company's use case. Keeping an iterative approach in mind, assess your current and future operational and cost models to ensure the structure of your AWS accounts reflects that of your company.

When to Create Multiple Accounts

While there is no one-size-fits-all answer for how many AWS accounts a particular customer should have, most companies will want to create more than one AWS account because multiple accounts provide the highest level of resource and billing isolation. Answering “yes” to any of the following questions is a good indication that you should create additional AWS accounts:

- *Does the business require strong fiscal and budgetary billing isolation between specific workloads, business units, or cost centers?*
Billing isolation by account is required if customers want to use different payment instruments to pay for different AWS resource consumption or if they require the ability to associate 100% of specific AWS costs to a particular application workload, environment, cost center, business unit (BU), or customer.

- *Does the business require administrative isolation between workloads?*
The most straightforward approach for granting independent groups different levels of administrative control over their AWS resources is to isolate accounts based on the workload, development lifecycle, BU, or data sensitivity.
- *Does the business require a particular workload to operate within specific AWS service limits and not impact the limits of another workload?*
Customers can use AWS account service limits to impose restrictions on a particular business unit, development team, or project. For example, if they create an AWS account for a specific project group, they can limit the number of Amazon Elastic Compute Cloud (Amazon EC2) or High Performance Computing (HPC) instances that they are able to launch.
- *Do the business's workloads depend on specific instance reservations to support high availability (HA) or disaster recovery (DR) capacity requirements?*
Reserved Instances (RIs) ensure reserved instances and reserved capacity for services such as Amazon EC2 and Amazon Relational Database Service (Amazon RDS) at the individual account level.

When to Manage Accounts with AWS Organizations

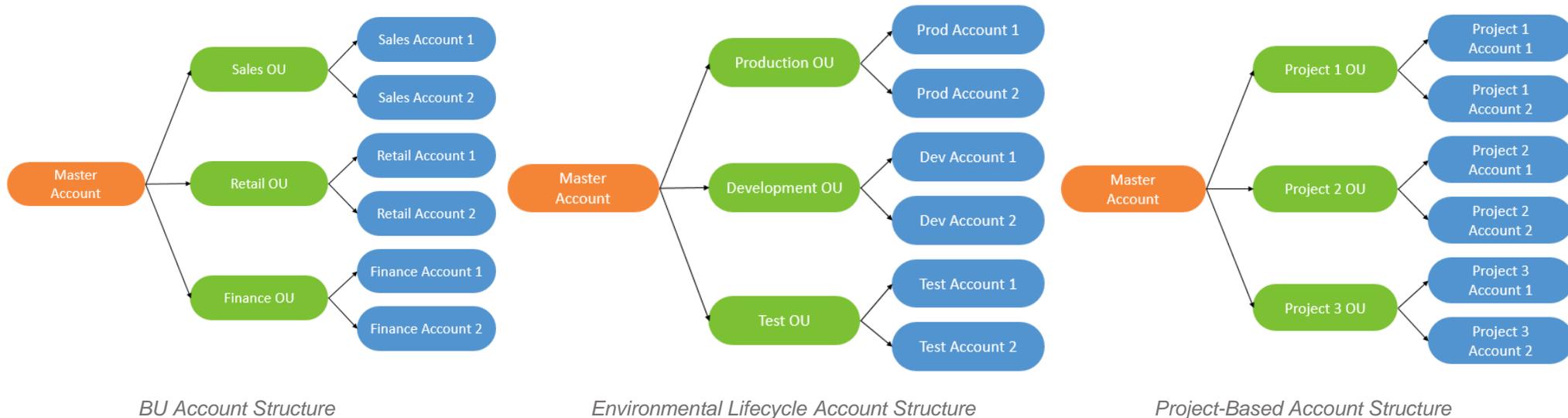
AWS Organizations enables you to create groups of AWS accounts and then centrally manage policies across those accounts. AWS Organizations provides [consolidated billing](#) in both [feature sets](#), which allows you set up a single payment method in the organization's master account and still receive an invoice for individual activity in each member account. If you answer "yes" to any of the following questions, it is a good indication that you might want to structure your AWS accounts using AWS Organizations:

- *Do you want to reuse a billing instrument (e.g. credit card or Purchase Order) to pay for multiple AWS accounts?*
- *Do you want to maximize your potential volume or RI discounts by aggregating AWS resource usage and spreading any applicable discounts across multiple accounts?*

AWS Account Structures

While AWS accounts are not technically hierarchical, you can use [organizational units \(OUs\)](#) with AWS Organizations to create hierarchical and logical groupings to better manage accounts. Note that there is a soft limit of 20 accounts per *organization*, and a hard limit of one level of billing hierarchy; for example, a master (paying) account cannot be in the same organization as another master (paying) account.

The following account structures are based on common AWS customer organizational, operational, and cost models.



Business Unit (BU) Account Structure

This account structure can be beneficial for customers who want to align their AWS operational and billing controls with individual BUs. It offers individual units operational autonomy while providing a company with a consolidated bill and combined view of all AWS charges, separated by group, OU, or cost center. This structure also simplifies the ability to trigger cost alerts based on BU consumption of AWS resources.

This structure typically works well in a decentralized IT environment where each BU is responsible for its own IT administration, operations, and costs.

Environment Lifecycle Account Structure

This account structure can be beneficial for customers who want to align their AWS operational and billing controls with their application deployment lifecycle. It offers development-lifecycle operational autonomy while providing a company with a consolidated bill and combined view of all AWS charges, separated by development environment. This structure also simplifies the ability to trigger cost alerts based on each development environment's consumption of AWS resources.

This structure typically works well in a traditional IT model with strict separation of duties between application environments where IT administration and operations are performed by different teams based on the application's lifecycle phase.

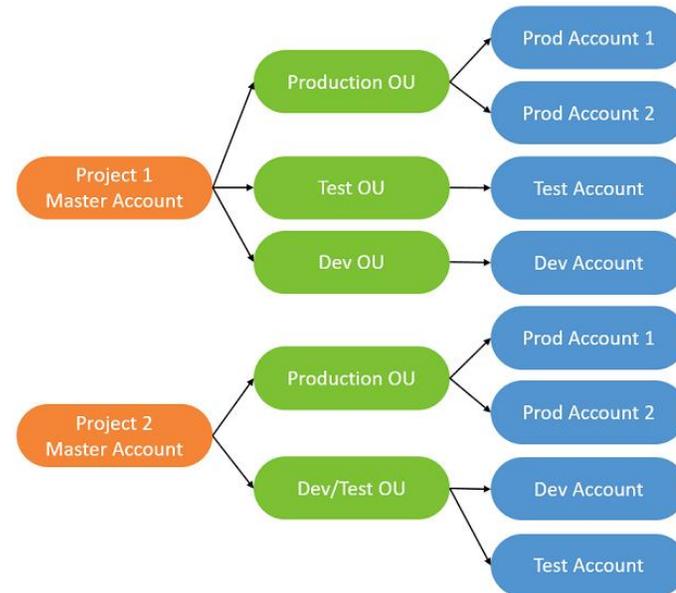
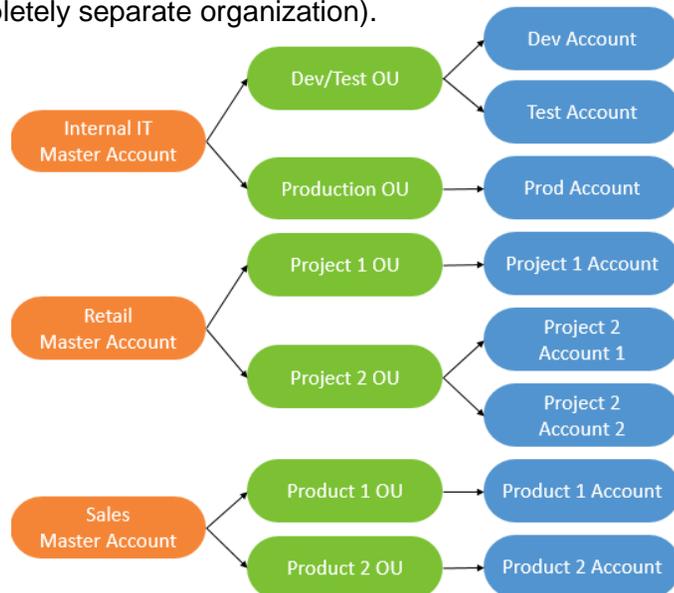
Project-Based Account Structure

This account structure can be beneficial for customers who want to align their AWS operational and billing controls by product, application workload, or program. It offers project or workload operational autonomy while providing the company with a consolidated bill and combined view of all AWS charges, separated by project. This structure also simplifies the ability to trigger cost alerts based on project, application workload, or program consumption of AWS resources.

This structure typically works well in DevOps environments where a single team is responsible for developing and operating a given application workload.

Hybrid AWS Account Structures

The basic account structures described previously work for most small companies, however some larger AWS customers find it advantageous to create hybrid combinations that group accounts by multiple dimensions. For example, DevOps environments may still want to create strong AWS-account administrative isolation between production and non-production systems to limit the potential blast radius of changes. In this case, a company can consolidate accounts by project and application lifecycle. Similarly, a BU-based company might group multiple product or project-based accounts by BU to accommodate AWS policies that vary from BU to BU or program to program. The following diagrams depict these hybrid models (note that each master account is a completely separate organization).



Resources

[AWS Tagging Strategies](https://aws.amazon.com/answers/account-management/aws-tagging-strategies/)

<https://aws.amazon.com/answers/account-management/aws-tagging-strategies/>

[AWS Organizations](https://aws.amazon.com/organizations/)

<https://aws.amazon.com/organizations/>

[Reserved Instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html)

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html>

[AWS Billing and Cost Management](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html)

<http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html>

[AWS Consolidated Billing](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/consolidated-billing.html)

<http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/consolidated-billing.html>

[AWS Service Limits](http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html)

http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html