https://docs.aws.amazon.com/accounts/latest/reference/AWSAccountManagementReferenceGuide.pdf#welcome-multiple-accounts

**What is an AWS account?**

This guide contains information about AWS accounts. How to create them, how to manage them, and how to use them.

An account in AWS is a fundamental part of accessing AWS services. It serves these two basic functions:

**• Container** – An AWS account is the basic container for all the AWS resources you can create as an AWS customer. When you create an Amazon Simple Storage Service (Amazon S3) bucket or Amazon Relational Database Service (Amazon RDS) database to store your data, or an Amazon Elastic Compute Cloud (Amazon EC2) instance to process your data, you are creating a resource in your account. Every resource is uniquely identified by an Amazon Resource Name (ARN) that includes the account ID of the account that contains, or owns, the resource.

**• Security boundary** – An AWS account is also the basic security boundary for your AWS resources. Resources that you create in your account are available only to users who have credentials for that same account.

**Do I need multiple AWS accounts?**

AWS accounts serve as the fundamental security boundary in AWS. They serve as a resource container that provides a useful level of isolation. The ability to isolate resources and users is a key requirement to establishing a secure, well governed environment.

Separating your resources into separate AWS accounts helps you to support the following principles in your cloud environment:

**• Security control** – Different applications can have different security profiles, requiring different control policies and mechanisms around them. For example, it’s far easier to talk to an auditor and be able to point to a single AWS account that hosts all elements of your workload that are subject to Payment Card Industry (PCI) Security Standards.

**• Isolation** – An AWS account is a unit of security protection. Potential risks and security threats should be contained within an AWS account without affecting others. There could be different security needs due to different teams or different security profiles.

**• Many teams** – Different teams have their different responsibilities and resource needs. You can prevent teams from interfering with each other by moving them to separate AWS accounts.

**• Data isolation** – In addition to isolating the teams, it's important to isolate the data stores to an account. This can help limit the number of people that can access and manage that data store. This helps contain exposure to highly private data and therefore can help in compliance with the European Union's General Data Protection Regulation (GDPR).

**• Business process** – Different business units or products may have completely different purposes and processes. With multiple AWS accounts, you can support a business unit's specific needs.

**• Billing** – An account is the only true way to separate items at a billing level. Multiple accounts help separate items at a billing level across business units, functional teams, or individual users. You can still get all of your bills consolidated to a single payer (using AWS Organizations and consolidated billing) while having line items separated by AWS account.

**• Quota allocation** – AWS service quotas are enforced separately for each AWS account. Separating workloads into different AWS accounts prevents them from consuming quotas for each other

**Managing multiple AWS accounts**

Before you start adding multiple accounts, you'll want to develop a plan to manage them. For that, we recommend that you use AWS Organizations, which is a free AWS service to manage all of the AWS accounts in your organization.

https://docs.aws.amazon.com/whitepapers/latest/organizing-your-aws-environment/organizing-your-aws-environment.pdf#benefits-of-using-multiple-aws-accounts

**Multi-Account Strategy Best Practices &**

Recommendations

Businesses can benefit from considering the latest guidance for organizing their AWS environments. A multi-account strategy is key to succeed when customers are starting to adopt AWS, expanding their footprint on AWS, or planning to enhance an established AWS environment. Customers might have multiple teams with different security and compliance controls that need to be isolated from one another. Some might have different business processes entirely or be part of different business lines that need clarity around costs incurred.

Customers need explicit security boundaries, a mechanism to have direct control and visibility of their limits and any throttling, and a billing separation to directly map costs to underlying projects. The isolation designed into an AWS account can help you meet these needs. Using multiple AWS accounts to help isolate and manage your business applications and data can help you optimize across most of the AWS Well-Architected Framework pillars including operational excellence, security, reliability, and cost optimization.

AWS accounts

Your cloud resources and data are contained in an AWS account. An account acts as an identity and access management isolation boundary. When you need to share resources and data between two accounts, you must explicitly allow this access. By default, no access is allowed between accounts. For example, if you designate different accounts to contain your production and non-production resources and data, no access is allowed between those environments by default.

Benefits of using multiple AWS accounts

As you adopt AWS, we recommend that you determine how your business, governance, security, and operational requirements can be met in AWS. Use of multiple AWS accounts plays an important role in how you meet those requirements. The use of multiple accounts enables you to realize the benefits in the following sections.

Topics

• Group workloads based on business purpose and ownership (p. 6)

• Apply distinct security controls by environment (p. 6)

• Constrain access to sensitive data (p. 7)

• Promote innovation and agility (p. 7)

• Limit scope of impact from adverse events (p. 7)

• Support multiple IT operating models (p. 8)

• Manage costs (p. 9)

• Distribute AWS Service Quotas and API request rate limits (p. 9)

**Group workloads based on business purpose and ownership**

You can group workloads with a common business purpose in distinct accounts. As a result, you can align the ownership and decision making with those accounts and avoid dependencies and conflicts with how workloads in other accounts are secured and managed. Different business units or product teams might have different processes. Depending on your overall business model, you might choose to isolate distinct business units or subsidiaries in different accounts. Isolation of business units can help them operate with greater decentralized control, but still provides the ability for you to provide overarching guardrails. This approach might also ease divestment of those units over time.

Guardrails are governance rules for security, operations, and compliance that you can define and apply eithdon;t align with your overall requirements. If you acquire a business that is already operating in AWS, you can move the associated accounts intact into your existing organization. This movement of accounts can be an initial step toward integrating acquired services into your standard account structure.

**Apply distinct security controls by environment**

Workloads often have distinct security profiles that require separate control policies and mechanisms to support them. For example, it’s common to apply different security and operational policies for the nonproduction and production environments of a given workload. By using separate accounts for the production and production environments, by default, the resources and data that make up a workload environment are separated from other environments and workloads.

**Constrain access to sensitive data**

When you limit sensitive data stores to an account that is built to manage it, you can more easily constrain the number of people and processes that can access and manage the data store. This approach simplifies the process of achieving least privilege access. Limiting access at the coarse-grained level of an account helps contain exposure to highly sensitive data.

For example, designating a set of accounts to house publicly accessible Amazon S3 buckets enables you to implement policies for all your other accounts to expressly forbid making S3 buckets publicly available.

**Promote innovation and agility**

At AWS, we refer to your technologists as builders because they are all responsible for building value using AWS products and services. Your builders likely represent diverse roles, such as application developers, data engineers, data scientists, data analysts, security engineers, and infrastructure engineers. In the early stages of a workload’s lifecycle, you can help promote innovation by providing your builders with separate accounts in support of experimentation, development, and early testing. These environments often provide greater freedom than more tightly controlled production-like test and production environments by enabling broader access to AWS services while using guardrails to help prohibit access to and use of sensitive and internal data.

• Sandbox accounts are typically disconnected from your enterprise services and do not provide access to your internal data, but offer the greatest freedom for experimentation.

• Development accounts typically provide limited access to your enterprise services and development data, but can more readily support day-to-day experimentation with your enterprise approved AWS services, formal development, and early testing work.

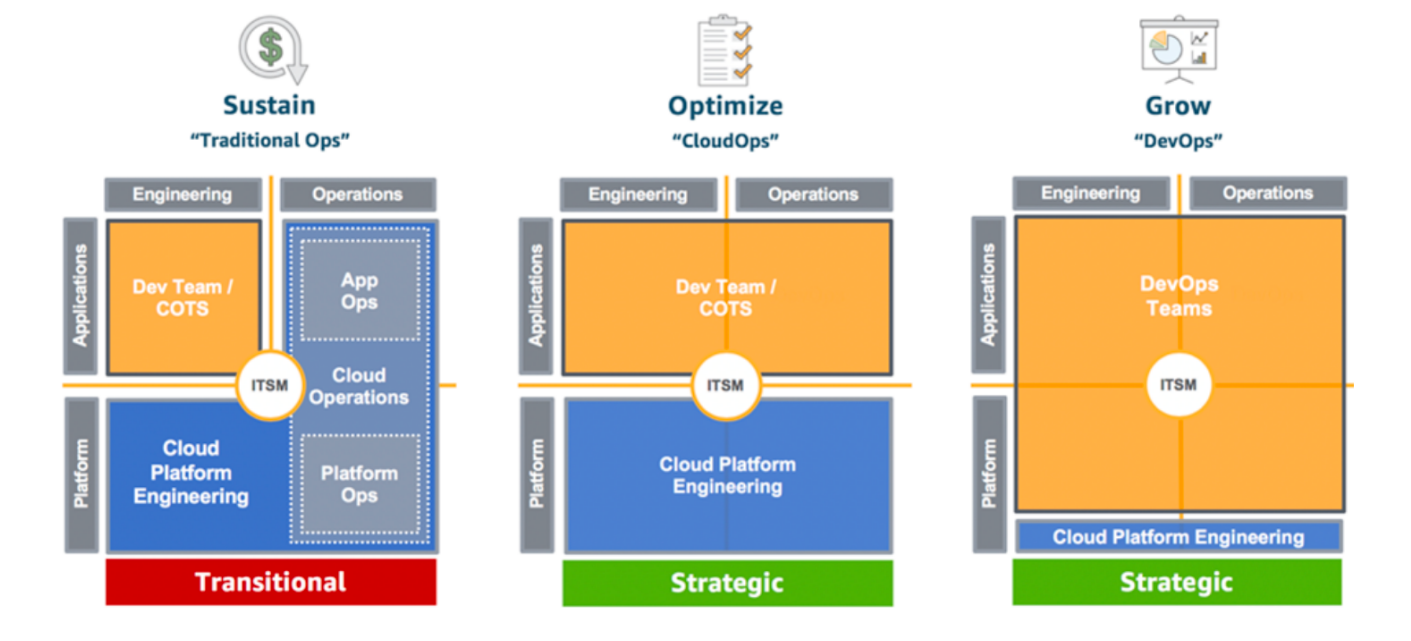
In both cases, we recommend security guardrails and cost budgets so that you limit risks and proactively manage costs. In support of later stages of the workload lifecycle, you can use distinct test and production accounts for workloads or groups of related workloads. Having an environment for each set of workloads can enable owning teams to move faster by reducing dependencies on other teams and workloads and minimizing the impact of changes.

**Limit scope of impact from adverse events**

An AWS account provides security, access, and billing boundaries for your AWS resources that can help you achieve resource independence and isolation. By design, all resources provisioned within an account are logically isolated from resources provisioned in other accounts, even within your own AWS environment.

This isolation boundary provides you with a way to limit the risks of an application-related issue, misconfiguration, or malicious actions. If an issue occurs within one account, impacts to workloads contained in other accounts can be either reduced or eliminated.

Organizations often have multiple IT operating models or ways in which they divide responsibilities among parts of the organization to deliver their application workloads and platform capabilities. The following figure shows three example operating models:



Example operating models

In the Traditional Ops model, teams who own custom and commercial off-the-shelf (COTS) applications are responsible for engineering their applications, but not for their production operations. A cloud platform engineering team is responsible for engineering the underlying platform capabilities. A separate cloud operations team is responsible for the operations of both applications and platform.

In the CloudOps model, application teams are also responsible for production operations of their applications. In this model, a common cloud platform engineering team is responsible for both engineering and operations of the underlying platform capabilities.

In the DevOps model, the application teams take on the additional responsibilities of engineering and operating platform capabilities that are specific to their applications. A cloud platform engineering team is responsible for engineering and operations of shared platform capabilities that are used by multiple applications.

As a practice, IT Service Management (ITSM) is a common element across all of the models. Your overall goals and requirements of ITSM might not change across these models, but the responsible individuals and solutions for meeting those goals and requirements can vary depending on the model. Given the implications of centralized operations versus more distributed operational responsibilities, you will likely benefit from establishing separate groups of accounts in support of different operating models. Use of separate accounts enables you to apply distinct governance and operational controls that are appropriate for each of your operating models.

To learn more about operating models and their implications on your cloud adoption, see the AWS WellArchitected Operational Excellence Pillar Operating Model.

**Manage costs**

An account is the default means by which AWS costs are allocated. Because of this fact, using different accounts for different business units and groups of workloads can help you more easily report, control, forecast, and budget your cloud expenditures. In addition to cost reporting at the account level, AWS has built-in support to consolidate and report costs across your entire set of accounts. When you require fine-grained cost allocation, you can apply cost allocation tags to individual resources in each of your accounts.

For more information about cost optimization, see the AWS Well-Architected Cost Optimization Pillar’s Expenditure and Usage Awareness best practices.

**Distribute AWS Service Quotas and API request rate limits**

AWS Service Quotas, also known as limits, are the maximum number of service resources or operations that apply to an account. For example, the number of Amazon Simple Storage Service (Amazon S3) buckets that you can create for each account. You can use Service Quotas to help protect you from unexpected excessive provisioning of AWS resources and malicious actions that could dramatically impact your AWS costs.

AWS services can also throttle or limit the rate of requests made to their API operations. Because Service Quotas and request rate limits are allocated for each account, use of separate accounts for workloads can help distribute the potential impact of the quotas and limits.

To learn more about managing service quotas, see AWS Well-Architected Reliability Pillar Manage Service Quotas and Constraints.