AWS TAGGING STRATEGIES

“How should I tag my AWS resources?”

Overview

Amazon Web Services (AWS) allows customers to assign metadata to their AWS resources in the form of *tags*. Each tag is a simple label consisting of a customer-defined key and an optional value that can make it easier to manage, search for, and filter resources. Although there are no inherent types of tags, they enable customers to categorize resources by purpose, owner, environment, or other criteria. This document describes commonly used tagging categories and strategies to help AWS customers implement a consistent and effective tagging strategy. The following sections assume basic knowledge of AWS resources, tagging, detailed billing, and AWS Identity and Access Management (IAM).

General Best Practices

When creating a tagging strategy for AWS resources, make sure that it accurately represents organizationally relevant dimensions and adheres to the following tagging best practices:

- Always use a standardized, case-sensitive format for tags, and implement it consistently across all resource types.
- Consider tag dimensions that support the ability to manage resource access control, cost tracking, automation, and organization.
- Implement automated tools to help manage resource tags. The Resource Groups Tagging API enables programmatic control of tags, making it easier to automatically manage, search, and filter tags and resources. It also simplifies backups of tag data across all supported services with a single API call per AWS Region.
- Err on the side of using too many tags rather than too few tags.
- Remember that it is easy to modify tags to accommodate changing business requirements, however consider the ramifications of future changes, especially in relation to tag-based access control, automation, or upstream billing reports.

Tagging Categories

Companies that are most effective in their use of tags typically create business-relevant tag groupings to organize their resources along technical, business, and security dimensions. Companies that use automated processes to manage their infrastructure also include additional, automation-specific tags to aid in their automation efforts.

Technical Tags

Name – Used to identify individual resources

Application ID – Used to identify disparate resources that are related to a specific application

Application Role – Used to describe the function of a particular resource (e.g. web server, message broker, database)

Cluster – Used to identify resource farms that share a common configuration and perform a specific function for an application

Environment – Used to distinguish between development, test, and production infrastructure

Version – Used to help distinguish between different versions of resources or applications

Tags for Automation

Date/Time – Used to identify the date or time a resource should be started, stopped, deleted, or rotated

Opt in/Opt out – Used to indicate whether a resource should be automatically included in an automated activity such as starting, stopping, or resizing instances

Security – Used to determine security requirements, such as encryption, enabling Amazon VPC Flow Logs, and also to identify route tables or security groups that deserve extra scrutiny

Business Tags

Owner – Used to identify who is responsible for the resource

Cost Center/Business Unit – Used to identify the cost center or business unit associated with a resource; typically for cost allocation and tracking

Customer – Used to identify a specific client that a particular group of resources serves

Project – Used to identify the project(s) the resource supports

Security Tags

Confidentiality – An identifier for the specific data-confidentiality level a resource supports

Compliance – An identifier for workloads designed to adhere to specific compliance requirements

Common Tagging Strategies

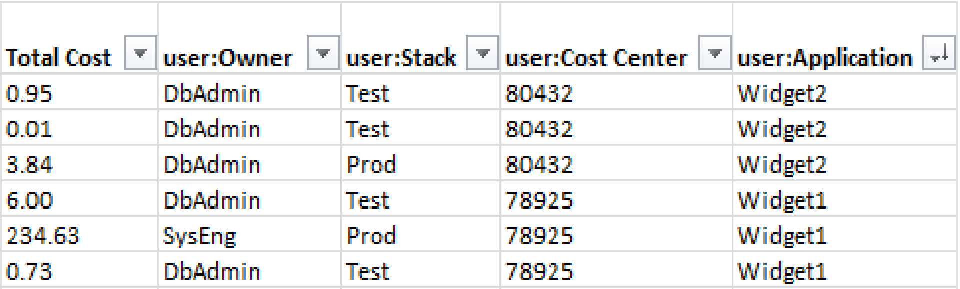
The following sections describe common tagging strategies to help identify and manage AWS resources.

Tags for AWS Console Organization

Tags are a great way to organize AWS resources in the AWS Management Console. You can configure tags to be displayed with resources, and can search and filter by tag. By default, the AWS Management Console is organized by AWS service. However, the Resource Groups tool allows customers to create a custom console that organizes and consolidates AWS resources based on one or more tags or portions of tags. Using this tool, customers can consolidate and view data for applications that consist of multiple services, resources, and regions in one place.

Tags for Cost Allocation

AWS Cost Explorer and detailed billing reports support the ability to break down AWS costs by tag. Typically, customers use business tags such as *cost center/business unit*, *customer*, or *project* to associate AWS costs with traditional cost-allocation dimensions. However, a cost allocation report can include any tag. This allows customers to easily associate costs with technical or security dimensions, such as specific applications, environments, or compliance programs. The table to the right shows a partial cost allocation report.



Total Cost	user:Owner	user:Stack	user:Cost Center	user:Application
0.95	DbAdmin	Test	80432	Widget2
0.01	DbAdmin	Test	80432	Widget2
3.84	DbAdmin	Prod	80432	Widget2
6.00	DbAdmin	Test	78925	Widget1
234.63	SysEng	Prod	78925	Widget1
0.73	DbAdmin	Test	78925	Widget1

Customers can activate an AWS-generated `createdBy` tag that is automatically applied for cost allocation purposes, to help account for resources that might otherwise go uncategorized. The `createdBy` tag is available for supported AWS services and resources only, and its value contains data associated with specific API or console events. For detailed information, see AWS-Generated Cost Allocation Tags in the AWS Billing and Cost Management User Guide.

Tags for Automation

Resource or service-specific tags are often used to filter resources during infrastructure automation activities. Automation tags are used to opt in or opt out of automated tasks or to identify specific versions of resources to archive, update, or delete. For example, many customers run automated start/stop scripts that turn off development environments during non-business hours to reduce costs. In this scenario, Amazon Elastic Compute Cloud (Amazon EC2) instance tags are a simple way to identify the specific development instances to opt out of this action. For scripts used to locate and delete stale, out-of-date, or rolling Amazon EBS snapshots, snapshot tags can add an extra dimension of search criteria.

Tags for Access Control

IAM policies support tag-based conditions, enabling customers to constrain IAM permissions based on specific tags or tag values. For example, IAM user or role permissions can include conditions to limit EC2 API calls to specific environments (e.g. development, test, or production) or Amazon Virtual Private Cloud (Amazon VPC) networks based on their tags. Support for tag-based, resource-level IAM permissions is service specific. When leveraging tag-based conditions for access control, make sure to also define and restrict who can modify those tags. See AWS Services That Work with IAM for detailed information about leveraging tags to control API access to AWS resources.

Tagging Governance

As mentioned in the general best practices, an effective tagging strategy uses standardized tags and implements them consistently and programmatically across AWS resources. Customers use both reactive and proactive approaches for governing the use of tags in their AWS environments. Reactive governance is used to identify improper tags, programmatically using tools such as the Resource Groups Tagging API, AWS Config Rules, and custom scripts, or manually using Tag Editor and detailed billing reports. Proactive governance leverages tools such as AWS CloudFormation, AWS Service Catalog, or IAM resource-level permissions to ensure standardized tags are consistently applied at resource creation. For example, you can use the AWS CloudFormation Resource Tags property to apply tags to certain resource types. In AWS Service Catalog, you can add portfolio and product tags that are combined and applied to a provisioned product automatically when it is launched.

More rigorous forms of proactive governance include automated tasks; for example, using the Resource Groups Tagging API to regularly scan an AWS environment's tags, or running scripts to quarantine or delete improperly tagged resources.

The most suitable governance approach for a company primarily depends on its AWS maturity model. Reactive governance typically works the best for customers who have not fully developed a tagging standard or who are not yet ready to proactively enforce a tagging strategy across their company. Proactive governance works well for more mature companies, especially those that can incorporate a tagging strategy with other standardization efforts, such as standardized environment builds using AWS CloudFormation or AWS Service Catalog.

Resources

Resource Groups Tagging API	http://docs.aws.amazon.com/resourcegroupstagging/latest/APIReference/Welcome.html
Working with Tag Editor	http://docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg/tag-editor.html
Working with Resource Groups	http://docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg/resource-groups.html
Service Catalog Tagging	http://docs.aws.amazon.com/servicecatalog/latest/adminguide/portfoliomgmt-tags.html
AWS Billing and Cost Management	http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html
Using Tags in IAM	https://aws.amazon.com/premiumsupport/knowledge-center/iam-ec2-resource-tags/
AWS CloudFormation Resource Tags Type	http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-resource-tags.html