Click to prove you're human

User-defined tags are tags that you define, create, and apply to resources. After you have created and applied the user-defined tags, you can activate by using the Billing and Cost Management console for cost allocation tracking. Cost allocation tags appear on the console after you've enabled Cost Explorer, Budgets, AWS Cost and Usage Reports, or legacy reports. After you activate the AWS services, they appear on your cost allocation report. You can then use the tags on your cost allocation report to track your AWS costs. Tags are not applied to resources that were created before the tags were created. As a best practice, reactivate your cost allocation tags when moving organizations. When an account moves to another organization as a member, previously activated cost allocation tags for that account lose their "active" status and need to be activated again by the new management account.As a best practice, do not include sensitive information in tags.Only a management account in an organization and single accounts that aren't members of an organization have access to the cost allocation tags manager in the Billing and Cost Management console. Applying user-defined cost allocation tags For ease of use and best results, use the AWS Tag Editor to create and apply user-defined tags. The Tag Editor provides a central, unified way to create and manage your user-defined tags. For more information, see the Tagging AWS Resources and Tag Editor User Guide. For supported services, you can also apply tags to resources using the API or the AWS Management Console. Each AWS service has its own implementation of tags. You can work with these implementations individually or use Tag Editor to simplify the process. For a full list of services that support tags, see Supported Resources for Tag-based Groups and Resource Groups Tagging API Reference. The behavior of cost allocation tags varies across AWS services. To learn more about the cost allocation tag behavior for a supported service, refer to the service's documentation. For example, to learn more about using cost allocation tags with Amazon ECS, see Tagging your Amazon ECS resources in the Amazon Elastic Container Service Developer Guide. After you create and apply user-defined tags, you can activate them for cost allocation. If you activate your tags for cost allocation, it's a good idea to devise a set of tag keys that represent how you want to organize your costs. Your cost allocation report displays the tag keys as additional columns with the applicable values for each row, so it's easier to track your costs if you use a consistent set of tag keys. Some services launch other AWS resources that the service uses, such as Amazon EMR launching an EC2 instance. If the supporting service (EC2) supports tagging, you can tag the supporting resources (such as the associated Amazon EC2 instance) for your report. For a full list of resources that can be tagged, use the Tag Editor to search. For more information about how to search for resources using Tag Editor, see Searching for Resources to Tag. AWS Marketplace line items are tagged with the associated Amazon EC2 instance tag. The awsApplication tag will be automatically added to all resources that are associated with applications that are set up in AWS Service Catalog AppRegistry. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions. User-defined tag restrictions For basic tag restrictions, see Tag Restrictions in the Amazon EC2 User Guide. The following restrictions apply to user-defined tags for Cost Allocation: The reserved prefix is aws:. AWS-generated tag names and values are automatically assigned the aws: prefix, which you can't assign. User-defined tag names have the prefix user: in the cost allocation report. Use each key only once for each resource. If you attempt to use the same key twice on the same resource, your request will be rejected. In some services, you can tag a resource when you create it. For more information, see the documentation for the service where you want to tag resources. If you need characters outside of those listed in Tag Restrictions, you can apply standard base-64 encoding to your tag. Billing and Cost Management does not encode or decode your tag for you. User-defined tags on non-metered services can be activated (for example, Account Tagging). However, these tags will not populate in the Cost Management suite because these services are not metered. Voiced by Amazon Polly Organized, meaningful cost and usage data helps make informed decisions for your cloud investment. AWS provides various resources and tools to help you organize resources and accounts, such as AWS Cost Categories, AWS Control Tower, and AWS Organizations. AWS Organizations is a great service to centrally manage and govern your AWS environment. You can programmatically create AWS accounts, group them, and apply policies to them for governance purposes. If you are looking for a way to allocate AWS costs using your existing account structure in your AWS Organizations, then this blog is for you. Today we will take you through how you can use AWS organizations for cost tagging and reporting in your organization, a widely used strategy for many of our customers. This process is based on all costs associated with your AWS account, meaning there is little to no dependency on resource level tags. Many of you use resource level tags for an extra layer of detail. Learn about how you can use the AWS-Generated and User-Defined Cost Allocation Tags, and how you can enforce and validate resource tags. After you enable these resource level tags as AWS Cost Allocation Tags in your Billing and Cost Management Console, the metadata will appear in your cost management tooling such as the Cost and Usage Report (CUR) and Cost Explorer. However, costs such as data transfer and support fees are trickier to tag, causing additional metering and calculations to happen in order to correctly allocate. To get around some of these complexities, we've seen an increasing number of customers track their cost at the account level using AWS Organizations. With the AWS Organizations Cost Tagging method, you can follow the model that, everything in one account is owned by one business unit, one type of environment, and charged to one cost center. To associate some of the metadata to your accounts, which will be useful in cost tagging, we recommend using account tags. To learn about account level tagging, please refer to our user guide Tagging AWS Organizations resources. What to tag where When you start using the AWS Organizations tagging strategy, it is key to work out which tag belongs to which account and which resource. To do this, make sure you align with your stakeholders about why you are tagging. As we are tagging for the use of cost in this blog, we have provided some examples. This blog gives you a list of common tags we have seen customers use and we have sorted them by Account and Resource. This is not necessarily a 'one size fits all' solution, but if you are new to the AWS Organizations cost tagging, it is a good starting point. Currently, your AWS Organizations account level tags can't be activated as Cost Allocation resource tags and therefore the tag data is not in your cost management tooling, such as Cost Explorer or CUR. To solve this problem, we have created a couple of resources for you to utilize. The first is an AWS Well-Architected Lab: Organization Data CUR connection that will show you how you can pull data from your AWS Organizations and associate them with your CUR report in AWS Athena. This way you can see your usage with account identifiers when you run Athena queries. We will go into more depth about this below. If you prefer reporting through AWS Cost Explorer, you can refer to the blog post Cost Reporting Based on AWS Organizations Account ID Tags as an alternative. This explains how you can pull data from AWS Organizations Tags and connect it with summarized data from AWS Cost Explorer and deliver them as a csv file to your designated Amazon S3 Bucket. You can choose an option that better suits your needs. Getting Started with the Well-Architected Lab The Lab guides you through setting up an AWS Lambda function to automatically query and extract data such as account ID, account name and, those all-important account tags from your AWS Organizations. This data will be refreshed once a week but you can adjust the frequency per your preference. This data is then stored in Amazon S3 and then, in Amazon Athena and we can join the Organizations data with the CUR report. Below is an example of how the account names are shown with CUR data. Now you have this account metadata connected to your CUR, reviewing costs and combining them under an account name or cost center becomes a lot easier. However, sometimes you may need instructions to get started. You can refer to this Cost and Usage Reports User Guide for an overview. Getting the Costs to Allocate The CUR, many of you will concur, offers the most comprehensive sets of cost and usage data an AWS customer can access. This report can yet be millions of lines long and have hundreds of columns and contains data showcasing usage across your entire AWS Organizations. From data transfer costs to per second lambda usage, down to each GB off storage you utilize, the CUR has it all. This is invaluable when it comes to Cloud Financial Management (CFM) and getting the most out of the AWS 'pay for what you use' model. Using SQL you can query this data to allocate costs. We have released a new AWS CUR Queries Library in the AWS Well-Architected Labs which contains a collection of curated SQL queries. This will help you to dive deep into your data and extract meaningful results. And, now you have enriched your CUR, using the lab above, you can use this in your reports. Using your AWS Organizations tags in these queries will allow you to break down cost as you need to. Visualizing these Allocated Costs Following this, if you would like to get a more visual experience to see how these costs are broken down and how distribution of spend is across your organization, we have the Enterprise Dashboard. This Dashboard is a great way to remove the complexities of cost & usage analysis and gives you a great insight into how you are using and spending on AWS services. It's super quick to set up and gives you the flexibility to make it your own. Hopefully, this blog has given you some useful insights into how you can use your AWS Organizations account tags to allocate costs and, using the Well-Architected Labs, you can better leverage your Cost & Usage Report and explore how you can get more out of your data. A tag is a label that you or AWS assigns to an AWS resource. Each tag consists of a key and a value. For each resource, each tag key must be unique, and each tag key can have only one value. You can use tags to organize your resources, and cost allocation tags to track your AWS costs on a detailed level. After you activate cost allocation tags, AWS uses the cost allocation tags to organize your resource costs on your cost allocation report, to make it easier for you to categorize and track your AWS costs. AWS provides two types of cost allocation tags, an AWS-generated tags and user-defined tags. AWS, or AWS Marketplace ISV defines, creates, and applies the AWS-generated tags for you, and you define, create, and apply user-defined tags. You must activate both types of tags separately before they can appear in Cost Explorer or on a cost allocation report.The following diagram illustrates the concept. In the example, you've assigned and activated tags on two Amazon EC2 instances, one tag called Cost Center and another tag called Stack. Each of the tags has an associated value. You also activated the AWS-generated tags, createdBy before creating these resources. The createdBy tag tracks who created the resource. The user-defined tags use the user prefix, and the AWS-generated tag uses the aws: prefix.After you or AWS applies tags to your AWS resources (such as Amazon EC2 instances or Amazon S3 buckets) and you activate the tags in the Billing and Cost Management console, AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and costs grouped by your active tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.The cost allocation report includes all of your AWS costs for each billing period. The report includes both tagged and untagged resources, so that you can clearly organize the charges for resources. For example, if you tag resources with an application name, you can track the total cost of a single application that runs on those resources. The following screenshot shows a partial report with columns for each tag.At the end of the billing cycle, the total charges (tagged and untagged) on the billing report with cost allocation tags reconciles with the total charges on your Bills page total and other billing reports for the same period. You can also use tags to filter views in Cost Explorer. For more information about Cost Explorer, see Analyzing your costs with AWS Cost Explorer. For more information about activating the AWS-generated tags, see Activating AWS-generated tags cost allocation tags. For more information about applying and activating user-defined tags, see Using user-defined cost allocation tags. All tags can take up to 24 hours to appear in the Billing and Cost Management console. As a best practice, don't include sensitive information in tags. Only the management account in an organization and single accounts that aren't members of an organization have access to the cost allocation tags manager in the Billing console. To create and update tags, use AWS Tag Editor. For more information about Tag Editor, see Using Tag Editor in the Tagging AWS Resources User Guide. Welcome to another installment of Q-Bits, our regular series showcasing cool ways Amazon employees are leveraging Amazon Q Developer. Cost allocation tags are essential for understanding and managing AWS spending across complex organizational structures.Today, we're diving into Cost allocation Tag and strategies with Amazon Q DeveloperManaging costs in AWS environments becomes increasingly complex as organizations scale. Cost allocation tags provide the granular visibility needed to track and allocate expenses effectively. With Amazon Q Developer's assistance and AWS Organizations' governance capabilities, you can implement comprehensive tagging strategies that drive financial accountability and optimization.AWS supports two types of cost allocation tags:AWS-generated tags: Automatically created by AWS (like aws:createdBy)User-defined tags: Custom tags created by your organization (like environment, project, cost-center) You an also ask Amazon Q developer for more details and examples. Amazon Q Developer will provide tailored tag recommendations, including keys, values, and hierarchies, aligning with your cost allocation and resource management needs for an optimized tag strategy. Modify the prompt based on specific use cases for more targeted recommendations.Fig 2: Interactive Tag Strategy Amazon Q Developer will recommend AWS Organizations configurations, service control policies, and tag enforcement strategies to ensure consistent tag compliance across your AWS resources and accountsFig 3: Step by Step instruction on Enforcement and ComplianceYou can input your code and ask for validation and updates, and Amazon Q developer will analyze your code, identify areas for improvement in cost allocation tagging, and provide recommendations to update your code with proper cost allocation tags. Amazon Q developer leverages its knowledge of AWS best practices, to ensure that your code adheres to optimal tagging conventions for accurate cost allocation and resource management.Fig 3: Step by Step instruction on Enforcement and ComplianceEffective monitoring and reporting are essential components of a robust tagging strategy in AWS. They provide visibility into tag compliance, enable proactive identification of issues, and support data-driven decision-making for cost optimization and resource management.With Amazon Q Developer, you can streamline the following aspects of your tagging strategy:1. Basic Tag Compliance Check Script: Develop a foundational script to scan your AWS resources and identify instances of non-compliance with your organization's tagging policies.2. Organization-wide Compliance Check: Implement a comprehensive solution to monitor tag compliance across multiple AWS accounts and regions, ensuring consistent adherence to your tagging standards.3. Service-Specific Compliance Checks: Leverage Amazon Q Developer expertise to create customized compliance checks tailored to the unique tagging requirements of specific AWS services you utilize.4. Tag Policy Compliance Check: Automate the verification process to ensure that your tagging policies, including naming conventions, allowed values, and mandatory tag requirements, are consistently enforced.5. Automated Remediation Script with Best Practices: Benefit from Amazon Q Developer recommendations to develop an automated remediation script that addresses tag compliance violations, following AWS best practices for scalability and efficiency.By leveraging these Amazon Q Developer features, organizations can maintain consistent, compliant, and effective tagging strategies that support cost allocation and resource management objectives.It helps you,Understanding Cost Allocation TagsProvides tailored tag recommendationsEnsures consistent tag compliance across resources and accountsProvides recommendations to update code with proper tagsProvides visibility into tag compliance Any opinions in this post are those of the individual author and may not reflect the opinions of AWS. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Create account and cost structure for the resources being deployed in AWS. Establish the relationship between costs from AWS spend, how those costs were incurred, and who or what incurred those costs. Common cost structures are based on AWS Organizations, AWS accounts, environments, and entities within your organizations, such as a line of business or workload. Cost structures can be based on multiple attributes to permit the examination of costs in different ways or at different levels of granularity such as rolling up the costs of individual workloads to the line of business they serve. When choosing a cost structure that aligns with the desired outcomes, evaluate the cost allocation mechanisms on the ease of implementation versus desired accuracy. This might include considerations in regards to accountability, tooling availability, and cultural changes. Three popular cost allocation models that AWS customers usually start from are: Account-based — This model requires the least amount of effort and provides high accuracy for showbacks and chargebacks, and is suitable for organizations that have a defined account structure (and is consistent with the recommendations of the Organizing Your AWS Environment Using Multiple Accounts whitepaper). This provides clear cost visibility on a per-account basis. For cost visibility and allocation, you can use AWS Cost Explorer, Cost and Usage Reports, as well as AWS Budgets for cost monitoring and tracking. These tools provide filtering and grouping options by AWS accounts. From a cost allocation perspective, this model doesn't have to rely on accurate tagging of individual resources. Business Unit or Team-based — Cost allocatable to teams, business units, or organizations within an enterprise. This model requires a moderate amount of effort, provides high accuracy for showbacks and chargebacks, and is suitable for organizations that have a defined account structure (typically using AWS Organizations), with separation between various teams, applications, and workload types. This provides clear cost visibility across teams and applications, and as additional benefit reduces the risk of hitting AWS service quotas within a single AWS account. For example, each team may have five accounts (prod, staging, test, dev, sandbox), and no two teams and applications will share the same account. With such structure AWS Cost Categories will then provide the functionality to group accounts or other tags ("meta-tagging") into categories, which can be tracked in the tools mentioned in the previous example. It's important to note that AWS Organizations allows tagging of accounts and organizational units (OUs), however these tags will not be applicable for cost allocation and billing reporting (that is, you cannot group or filter your cost in AWS Cost Explorer by OU). AWS Cost Categories should be used for this purpose. Tag-based — This model requires more effort compared to the previous two and will provide high accuracy for showbacks and chargebacks depending on the requirements and end goal. While we strongly recommend that you adopt the practices outlined in Organizing Your AWS Environment Using Multiple Accounts whitepaper, realistically customers often find themselves with mixed and complex account structures that take time to migrate away from. Implementing a rigorous and effective tagging strategy is the key in this scenario, followed by activating relevant tags for cost allocation in the Billing and Cost Management console (in AWS Organizations, tags can be activated for cost allocation only from the Management Payer account). After tags are activated for cost allocation, then tools for cost visibility and allocation that were mentioned in the previous methods can be used for showbacks and chargebacks. Note that cost allocation tags are not retrospective, and will only appear in billing reporting and cost tracking tools after they were activated for cost allocation. To summarize, if you need to track costs by business unit, you can use AWS Cost Categories to group linked accounts within AWS Organization accordingly and view this grouping in billing reports. When you create separate accounts for production and non-production environments, you can also filter the costs related to environments in tools such as AWS Cost Explorer, or track those costs using AWS Budgets. Finally, if your use case requires more granular cost tracking, for example, by individual workloads or applications, you can tag resources within those accounts accordingly, activate those tag keys for cost allocation on the management account, and then filter that cost by tag keys in the billing reporting tools. Establishing cost reporting and monitoring processes Start with identifying the types of costs that are important for internal stakeholders (for example, daily spend, cost by account, cost by X, amortized costs). By doing so, you can mitigate budgetary risks associated with unexpected or anomalous spend faster than waiting for the finalized AWS invoice. Tags provide the attribution that enables these reporting scenarios. Insights gained from reporting can inform your actions to mitigate the impact from anomalous and unexpected spend on financial budgets. When there is an unexpected surge in costs, it's important to evaluate if there has been an unexpected surge in the value delivered so that you can determine if and what action is required. When developing a tagging strategy to support cost allocation, keep in mind the following elements: AWS Organizations - Cost allocation within multiple accounts can be performed by account, groups of accounts, or group of tags created for resources on those accounts. Tags created for resources residing in individual accounts in AWS Organizations can be used for cost allocation only from the management account. AWS Account - Cost allocation within one AWS account can be performed by additional dimensions such as services or regions. It's possible to further tag resources within an account and work with the groups of such resource tags. Cost Allocation Tags - Both user-created tags and AWS generated tags can be activated for cost allocation, if necessary. Enabling tags for cost allocation in the billing console (of the management account in AWS Organizations) helps with showbacks and chargebacks. Cost Categories - AWS Cost Categories allow grouping accounts and grouping tags ("meta-tagging") within an AWS Organization, which further provides capability to analyze the cost related to these categories through tools such as AWS Cost Explorer, AWS Budgets and AWS Cost and Usage Report. Performing showback and chargeback for business units, teams, or organizations within the enterprise Attribute costs using your cost allocation process supported by your cost structure and cost allocation tags. Tags can be used to provide showback to teams that are not directly responsible to pay for costs but are responsible for having incurred those costs. This approach provides awareness of their contribution to spend and how those costs are incurred. Perform chargeback to the teams that are directly responsible for costs to recover the expense of the resources they have consumed, and to provide them with awareness of those costs and how they were incurred. Measuring and circulating efficiency or value KPIs Agree on a set of unit cost or KPI metrics to measure the impact of your cloud financial management investments. This exercise creates a common language across technology and business stakeholders, and tells an efficiency-based story, rather than a story focused solely on absolute, aggregate spend. For additional information check this blog that talks how unit metrics can help create alignment between business functions. Allocating unallocatable spend Depending on the organization's accounting practices, different charge types might require different treatment. Identify the resources or cost categories that cannot be tagged. Depending on the services used and those planned to be used, agree on the mechanisms on how to treat and measure such unallocatable spend. For example, check the list of resources that are supported by AWS Resource Groups and Tag Editor in the AWS Resource Groups and Tags User Guide. A common example of cost category that cannot be tagged is some fees for commitment-based discounts such as Reserved Instances (RI) and Savings Plans (SP). While subscription fees and unused SP and RI fees cannot be tagged in advance of appearing in billing reporting tools, you can track how RI and SP discounts apply to accounts, resources and their tags in AWS Organizations after the fact. For example, in AWS Cost Explorer it's possible to look at the amortized cost, group that spend by the relevant tag keys and apply filters relevant to your use case. In AWS Cost and Usage Report (CUR), you can filter out lines that correspond to usage covered by RI and SP discounts (read more in the use cases section of the CUR documentation) and select the columns that are only relevant to you. Each tag key activated for cost allocation will be presented in its own separate column at the end of the CUR report, similarly to how it's presented in other legacy billing reports, such as monthly cost allocation report. For additional reference, check the AWS Well-Architected Labs for examples of gaining cost and usage insights from CUR data. Reporting In addition to AWS tools available to assist with showbacks and chargebacks, there is a range of other AWS created and third-party solutions that can help monitor the cost of tagged resources, and measure the effectiveness of the tagging strategy. Depending on both the requirements and the end objective of the organization, one could either invest time and resources into building customized solutions or purchase tools provided by one of the AWS Cloud Management Tools Competency Partners. If you decide to create your own single source of truth cost allocation tool with controlled parameters relevant for the business, AWS Cost and Usage Report (CUR) provides most detailed cost and usage data and enables creation of customized optimization dashboards, allowing filtering and grouping by accounts, services, cost categories, cost allocation tags, and multiple other dimensions. Among CUR-based solutions developed by AWS that can be used as one of these tools, check Cloud Intelligence Dashboards on the AWS Well-Architected Labs website. AWS cost allocation tags can be a powerful tool for organizations looking to better manage and understand their AWS costs. Users can assign metadata to resources, making it easier to track costs. Whether you are just getting started with AWS or are looking to optimize your AWS cost management practices, understanding cost allocation tags can be an important step toward achieving your goals.Purpose of tagging in AWSApplying tags is crucial for almost every AWS user because it helps to organize and manage resources in a more efficient and effective way. Without tags, it can be difficult to keep track of resources, especially in larger and more complex environments. Tagging resources provides a way to add metadata to resources, making it easier to identify, categorize, and manage them.Imagine a scenario where an organization has hundreds or thousands of EC2 instances running across multiple regions and accounts. Without tags, it would be challenging to keep track of which instances are used for development, testing, or production environments. With tags, it becomes easier to filter, search, and organize instances based on their purpose, making it simpler to manage resources and ensure that they are used appropriately.In addition to making it easier to manage resources, tags can also be used for aws cost allocation and chargeback. By tagging resources with specific values, such as department or project, organizations can allocate costs to the appropriate teams or individuals. This can help to improve accountability and reduce waste, as teams are more likely to be mindful of their resource usage when they are aware of the associated costs.Also, tags can be used to improve security and compliance. By tagging resources with specific values, such as classification level or compliance requirements, organizations can ensure that sensitive or regulated resources are properly secured and monitored. This can help to reduce the risk of security breaches and ensure that organizations are meeting their compliance obligations.Avoiding cost allocation tags issuesThere are many guides on applying tags to your resources, yet there are several common mistakes that users can make when doing it in AWS. Here are a few examples and some tips on how to avoid them:One of the most common mistakes is using inconsistent or incomplete tags across different resources. This can make it difficult to search and organize resources and can lead to confusion when trying to determine the purpose of a resource. To avoid this mistake, establish a clear set of tagging conventions and ensure that all resources are tagged consistently. Consider using automation tools like AWS Config or AWS Resource Groups to enforce tagging standards across your environment.Another mistake is creating overly complex tags that are difficult to understand or manage. For example, using tags with multiple values or long descriptions can make it challenging to search for resources and allocate costs accurately. To avoid this mistake, keep tags simple and easy to understand. Use descriptive values that are easy to search for, and avoid creating tags with multiple values unless necessary.Using non-standard or ambiguous tags Non-standard or ambiguous tags can make it difficult to manage and monitor resources. For example, using abbreviations or acronyms that are not commonly understood can lead to confusion, while using vague tag values like "miscellaneous" or "other" can make it hard to understand the purpose of a resource. To avoid this mistake, use standard tag values and conventions that are commonly understood across your organization.Another common mistake is failing to update tags as resources change or evolve over time. For example, if a resource is repurposed or decommissioned, it is important to update the associated tags to reflect the new state of the resource. To avoid this mistake, establish a process for reviewing and updating tags on a regular basis. Consider using automation tools to alert you when resources are no longer being used or are no longer compliant with tagging standards.By avoiding these common mistakes and following best practices for tagging, users can ensure that their resources are properly categorized, making it easier to manage and monitor them over time.SummaryApplying tags is crucial for almost every AWS user because it helps to organize and manage resources more efficiently, allocate costs more accurately, and improve security and compliance. By following best practices for tagging, organizations can ensure that their resources are properly categorized, making it easier to manage and monitor them.Our cost optimization expert can help you to set up or improve AWS cost allocation tags. Just use this link to book a meeting if you have any questions on this matter. AWS cost allocation tags are metadata labels assigned to AWS resources that help in categorizing and tracking costs. These tags consist of key-value pairs that can be used to identify specific resources and attribute costs accordingly. Cost allocation tags are critical for cloud management, allowing organizations to gain detailed insights into their spending, enabling more precise cost management and budgeting. Benefits of Using AWS Cost Allocation Tags for Cost Management Using AWS cost allocation tags offers several benefits for effective cost management: Improved visibility: Cost allocation tags provide granular visibility into AWS expenditures. This helps in identifying cost drivers and understanding which resources contribute to overall expenses. Improved budgeting and forecasting: By categorizing costs based on tags, organizations can create more accurate budgets and forecasts. This helps in planning for future expenses and avoiding unexpected charges. Cost optimization: Tags enable organizations to pinpoint underutilized or idle resources. This information can be used to optimize resource usage, reduce waste, and lower costs. Chargeback and showback: Tags facilitate chargeback and showback processes by attributing costs to specific departments, projects, or teams. This promotes accountability and encourages cost-saving behaviors. Simplified reporting: Tagged resources can be easily filtered and analyzed in cost reports. This simplifies the creation of detailed cost reports and dashboards, aiding in better financial analysis. Types of Cost Allocation Tags AWS cost allocation tags can be user-defined or AWS-generated. User-Defined Tags These custom tags are created by users to categorize and manage their AWS resources according to their requirements. They consist of key-value pairs defined by the user, allowing for flexible and tailored cost tracking. Consider a company that has multiple projects and departments, each requiring its own cost tracking. The company can create user-defined tags to manage these requirements. Here are a few examples: Project tags: Project: Alpha, Project: Beta Department tags: Department: Marketing, Department: Finance Environment tags: Environment: Production, Environment: Development In practice, if an EC2 instance is used for the Alpha project in the Marketing department, it could be tagged as follows: Project: Alpha Department: Marketing Environment: Development By applying these tags, the company can easily filter and analyze costs related to specific projects, departments, or environments. AWS-Generated Tags These tags are automatically created by AWS services to help users track and allocate costs without the need for manual tagging. They provide standard metadata about AWS resources, ensuring consistency and ease of use. AWS provides several predefined tags that are applied automatically to resources. Some common AWS-generated tags include: The Future of Project Management Welcome to our interview with Srikrishnan Ganesan, Co-founder of Rocketlane. Rocketlane is revolutionizing service delivery… Unlocking AI's Full Potential In this email interview, we have connected with Karthi Gopalaswamy, who brings over 20 years of… Digital Identity and Accountable Anonymity As explored in a recent paper titled Personhood Credentials: Artificial Intelligence and the Value of… Created by: aws:createdBy Service: aws:service Region: aws:region For example, if an S3 bucket is created by a user named John in the US East (N. Virginia) region, the AWS-generated tags might look like this: aws:createdBy: John aws:service: S3 aws:region: us-east-1 These tags help users track who created a resource, what service it belongs to, and where it is located, facilitating more detailed and automated cost allocation. Best Practices for Using AWS Cost Allocation Tags Organizations can implement the following practices to ensure the most effective use of tags to manage costs in AWS. Align Tags with Business Objectives When setting up cost allocation tags, it is essential to ensure they align with the organization's business objectives. This alignment helps in creating a tagging strategy that supports the company's financial and operational goals. Start by identifying key business drivers such as departments, projects, and environments that impact costs. Create tags that reflect these drivers to support accurate cost attribution and analysis. For example, tags like Department: Sales, Project: WebsiteRedesign, and Environment: Staging can provide insights into spending patterns and resource usage across different segments of the organization. This also enables more granular financial reporting, helping stakeholders make informed decisions based on precise cost data. Start with Core Tags To avoid overwhelming complexity, begin with a set of core tags that cover the most critical aspects of the organization's cost management needs. Core tags typically include identifiers for project, department, environment, and cost center. Establishing these foundational tags helps capture the most important cost-related data from the outset. Over time, the organization can expand the tagging strategy by adding more specific tags as necessary. This incremental approach helps maintain tag management simplicity and ensures that all stakeholders understand and consistently use the core tags. Use AWS Tagging Service and AWS Config AWS Tagging Service provides tools to manage tags across multiple resources, making it easier to maintain consistency. AWS Config can be used to create rules that check for required tags and notify teams of non-compliant resources. By using these AWS services, organizations can ensure that all resources are tagged correctly and consistently, improving the accuracy of cost tracking and reporting. AWS Tagging Service simplifies bulk tagging operations, allowing admins to apply, edit, or delete tags across numerous resources simultaneously. AWS Config's continuous monitoring capabilities help in identifying and correcting tagging issues promptly, ensuring ongoing compliance with tagging policies. Use AWS Lambda for Custom Tagging Workflows AWS Lambda can be used to automate and customize tagging workflows. For example, admins can create Lambda functions that automatically apply tags based on predefined criteria or events. This automation reduces the manual effort required to manage tags and helps ensure that all resources are tagged in accordance with the organization's policies. Lambda functions can also be used to update or modify tags dynamically as resource usage patterns change, ensuring that the tagging strategy remains relevant. Custom Lambda functions can be tailored to the organization's needs, such as tagging resources based on user roles or time-based events, providing flexibility and precision. Schedule Periodic Reviews Regularly review the tagging strategy and practices to ensure they remain aligned with business objectives and cost management needs. Schedule periodic audits to check for tag accuracy, consistency, and relevance. During these reviews, identify any outdated or unused tags and remove them to maintain a clean and efficient tagging structure. These periodic reviews help in adapting to changes in business requirements and maintaining the effectiveness of cost management efforts. Regular audits also provide an opportunity to refine the tagging strategy, incorporate new best practices, and address any gaps or inefficiencies in the tagging process. Integrate Tagging into CI/CD Pipelines Integrating tags into Continuous Integration/Continuous Deployment (CI/CD) pipelines ensures that all resources created during the deployment process are appropriately tagged. By embedding tagging policies into CI/CD workflows, teams can automate the application of tags to new resources, reducing the risk of untagged or incorrectly tagged resources. This practice ensures that all new deployments adhere to the organization's tagging standards, providing consistent and comprehensive cost tracking. Integrating tagging into CI/CD pipelines also helps in maintaining compliance and reduces the manual workload on development teams, allowing them to focus on delivering features while ensuring proper cost allocation. AWS cost allocation tags are useful for managing and optimizing cloud expenses. By applying both user-defined and AWS-generated tags, organizations can gain detailed insights into their spending, enabling better budgeting, cost optimization, and financial accountability. Following best practices such as aligning tags with business objectives, starting with core tags, using AWS tools for automation, scheduling periodic reviews, and integrating tagging into CI/CD pipelines ensures an effective and scalable tagging strategy. By Gilad David Maayan Principle in cost accounting This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.Find sources: "Cost allocation" – news · newspapers · books · scholar · JSTOR (September 2016) (Learn how and when to remove this message) Cost allocation is a process of providing relief to shared service organization's cost centers that provide a product or service. In turn, the associated expense is assigned to internal clients' cost centers that consume the products and services. For example, the CIO may provide all IT services within the company and assign the costs back to the business units that consume each offering. The core components of a cost allocation system consist of a way to track which organizations provides a product and/or service, the organizations that consume the products and/or services, and a list of portfolio offerings (e.g. service catalog). Depending on the operating structure within a company, the cost allocation data may generate an internal invoice or feed an ERP system's chargeback module. Accessing the data via an invoice or chargeback module are the typical methods that drive personnel behavior. In return, the consumption data becomes a great source of quantitative information to make better business decisions. Today's organizations face growing pressure to control costs and enable responsible financial management of resources. In this environment, an organization is expected to provide services cost-effectively and deliver business value while operating under tight budgetary constraints. One way to contain costs is to implement a cost allocation methodology, where your business units become directly accountable for the services they consume. An effective cost allocation methodology enables an organization to identify what services are being provided and what they cost, to allocate costs to business units, and to manage cost recovery. Under this model, both the service provider and its respective consumers become aware of their service requirements and usage and how they directly influence the costs incurred. This information, in turn, improves discipline within the business units and financial discipline across the entire organization. With the organization articulating the costs of services provided, the business units become empowered – and encouraged – to make informed decisions about the services and availability levels they request. They can make trade-offs between service levels and costs, and they can benchmark internal costs against outsourced providers. Every organization must define their costs, like how funding runs through the organization, who touches it, what they do and how they do it serves as a foundation for this understanding.[1] According to the Office of Management and Budget's (OMB) Uniform Guidance, there are only three types of costs – Indirect, Indirect-Admin (Overhead) and Direct. By correctly defining and allocating costs, true cost of service can be fully captured. Direct costs - are those that can be associated specifically to a final cost objective.[2] You know what you spend on programs. You know what expenses go directly to your mission. Most importantly, you know what expenses go directly to those benefitting from your service. Indirect costs - are things that you need to do for the sake of the organizational health & operations, but don't tie back to a specific program. For instance, the actual time (resources) that need to be dedicated when working with your Internet provider to fix the organization's connection is important, but it 's unlikely that the time spent can be tied to one particular function in a program. Indirect-Admin (Overhead) costs - are defined as those that are not Direct but are necessary for the immediate support of the program. Indirect-Admin costs or Overhead are very important to nonprofits since unlike government, they do not necessarily have separate buckets of funding. If Direct costs are the foundation, the Overhead costs are one touch away. So, when you are working on managing a program, leading those working for a program in their work or simply accounting for program expenses and revenues, you are only one touch away and are, in effect, performing direct activities. These activities can be attributed to your Direct Costs for purposes of expense reimbursement. Efficiency-based absorption costing ^ "Cracking the Overhead Egg | CostTree". Archived from the original on 2016-09-15. Retrieved 2016-09-02. ^ Page 5, Capture Magazine [1] - Uniform Administrative Requirements, Cost Principles, and Audit Requirements for Federal Awards. Retrieved from " Cost allocation tags are a powerful tool that can help organizations track and allocate costs associated with their AWS resources. These tags can identify specific resources, such as development or production environments, and assign costs to the appropriate department or project. This article will cover best practices and how to avoid pitfalls when using cost allocation tags.Design your tagging strategyThe first step when designing your tagging strategy is to identify the areas you want to be able to filter on. For example, it could be departments or teams, applications, or you may want to split the application into storage and compute. It can also be regions, environments (prod dev stage), or even specific people who own the resources.When designing your tagging strategy, it is crucial to establish a consistent and easy-to-understand naming convention for your tags, e.g., environment = prod, application = webserver, team = frontend, owner = john_doe. Refrain from mixing and matching upper and lowercase letters. Please keep it simple, lowercase everywhere. Also, try to avoid having whitespace in tags; this can cause trouble in some AWS services console experiences. It is also preferred to programmatically tag resources without human interaction to ensure they are consistently named.The "billing" tagLet's say your application consists of multiple resources, and you install this application numerous times for different customers. The tag application would then provide the combined cost for all the application installations. Of course, you could combine multiple tags, but we prefer to have a separate tag we call billing that takes the values of other tags and combines them into one.For example, suppose we want to use the tags application = frontend, customer = evolate, and environment = dev to give us the total cost for this particular installation in our dev environment for the customer evolate. In that case, our billing tag could look like this billing = evolate_frontend_dev. Having the billing tag will enable us to group on it in Cost Explorer and give us a nice separation of cost for our different applications as a whole.Implementation of tagsOnce your tags have been designed, the next step is to implement them. This can be done by creating tags at the time of resource creation or by applying them to existing resources. If you use Terraform, providing the tags at the provider level is a great option. Set tags at the resource level where extra detail is needed. For example, application_type could be a resource tag not provided at the provider level but rather at the resource level.It is also highly recommended that you include steps in your pipelines to verify that tags have been provided. If not, they should fail. In the case of Terraform, you can use a tool such as tfsec to enforce tags in your templates.You may already have resources in AWS that either need more tags or maybe have no tags at all. This is because they might have been set up manually through the console. Then you can use the "Tag Editor" found under AWS Resource Explorer to set tags on multiple resources faster than going over them manually.AWS Organizations also allows you to set up tag policies. A tag policy specifies a set of tags and the allowed values for these tags, which I recommend you set up after implementing your tags.Enable cost allocation tagsWhen you feel comfortable and know what tags you need for your strategy, it is time to enable them as cost allocation tags. You do this under your billing dashboard, and on the left menu, find "Cost allocation tags." Locate the tags you want to enable, and be aware that the tags must exist already before you can find them. Enable the checkbox for the tags and click enable. It can take up to 24 hours before the tags have been activated, and you can start to use them.Monitoring costs and maintenance of tagsOnce your tags have been implemented, the next step is to monitor and analyze your costs. This can be done using the AWS Cost Explorer, which allows you to view and analyze your costs by tag. You can also use the AWS Budgets service to set budget alerts for specific tags to notify you if costs exceed a certain threshold.Finally, it is essential to review and update your tags regularly. This can be done by reviewing the tag policies you created. It will tell you what resources that comply with your tag policies. It is also essential to check your tagging strategy and make any necessary changes to ensure that your tags are accurate and effective.But wait, Cost Categories?AWS released Cost Categories in November 2019, and it can help you to map your cost and usage information in a more advanced structure than Cost Explorer can do on its own.For example, you can set up rules for dimensions, such as tags and accounts, and group cost on identifiers in account names, charge types, services, regions, etc. We will follow up on Cost Categories in its own post, so make sure to follow us, so you don't miss out!TL;DRCost allocation tags in AWS can help track and allocate resource costs. When designing your tagging strategy, establish a consistent naming convention and tag resources programmatically. Implement tags through resource creation