



whitepaper

AWS TAGGING: **WHY, WHAT, AND IMPLEMENTATION**

The true power of Amazon Web Services resides in its ability to simultaneously lower users' costs and improve users' agility through self-provisioning and scalability. However, both these capabilities also contain obstacles, such as reducing the control IT departments have on asset, security, and compliance functions, and difficulties in identifying resources.

This paper will show you how tagging provides visibility into AWS by restoring control. With proper resource tagging, you can track and manage assets, security, and compliance. Moreover, you can identify resource usage, as well as measure, assign, and allocate costs.

What is List Cost?

CloudCheckr allows you to translate the usage charges within AWS into costs as if each payee account were stand-alone. This allows an AWS partner, such as reseller or managed service provider, to create invoices that correctly represent the cost had the payee account not been in the AWS partner's Consolidated Billing family. This is a more "accurate" cost to report to a payee over blended or unblended costs.

AWS Consolidated Billing families create many discounts through RI and tiering. This is a feature from AWS to ensure the Consolidated Billing family is charged the least amount that it should. CloudCheckr List Cost reverses many of these discounts and creates invoices that an AWS partner can fairly present to an end customer that is a payee within the Consolidated Billing family. The AWS partner still pays AWS the lowest possible cost, but the benefit of consolidating the accounts flows to the AWS partner rather than arbitrarily to the payee accounts.

Why Tag Resources

Amazon Web Services' ability to simultaneously lower users' costs and improve users' agility was groundbreaking. Traditionally, these two goals required a compromise, but AWS bridges this divide and delivers both without sacrificing either. AWS enables this through two critical capabilities: self-provisioning and scalability.

Self-provisioning is the means by which users requisition resources. While traditional IT required weeks and months for a user to obtain resource access, AWS provides users the opportunity to requisition those same resources within minutes. Users can obtain access without involving the IT department. This significantly reduces application testing and development time.

Scalability allows users to reduce costs. When using a data center or co-locating, users need to provision resources for peak capacity plus a margin of error. This is because resources cannot be added quickly and users risk not meeting demand. With AWS, this is not an issue, since AWS allows user to provision only what is needed at any given time, and encourages users to scale resources up and down based upon need. AWS only charges based upon what is being used – so using scaling to effectively provision will result in dramatic cost savings.

However, both these capabilities also contain obstacles. As powerful as self-provisioning is, it also impedes the asset management, security, and compliance control functions

that were provided by the traditional IT department. As great as scalability is, without identifying resources, their usage, and their associated applications, users cannot hope to scale effectively.

Finally, AWS presents one new challenge relative to traditional IT: the pay as you go model. Rather than being able to assign a budget number to a business unit, users receive monthly bills based upon actual usage. AWS delivers these bills on an account basis and users are responsible to properly apportion the usage within their own organization.

Tagging is the means to restore visibility within AWS, allowing asset management, security, and compliance tracking. Resource usage can be identified and scaled effectively, allowing costs to be measured, minimized, and allocated.

Strategies for Effective Implementation

When considering which resources are appropriate for tagging, the answer is easy: all possible resources. There is no cost associated with tagging, and it does not impede performance. Thus, there is no downside associated with tagging.

Users need to be aware of the following basic tagging rules:

- **Maximum number of tags per resource—10**
- **Maximum key length—127 Unicode characters**
- **Maximum value length—255 Unicode characters**
- **Tag keys and values are case sensitive**
- **Do not use the “aws:” prefix in your tag names or values because it is reserved for AWS use. You can’t edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit**

It is also important to remember that tags function as if they are key value pairs. Use the Key portion of the tag as the “how” of what you want report. The Value portion is the “what”. For example, if you are trying to tag a server to determine whether it is in Test or Production, your Key could be “Environment” and your value could be “Production”. Figure 1, taken from the AWS documentation, illustrates this concept.

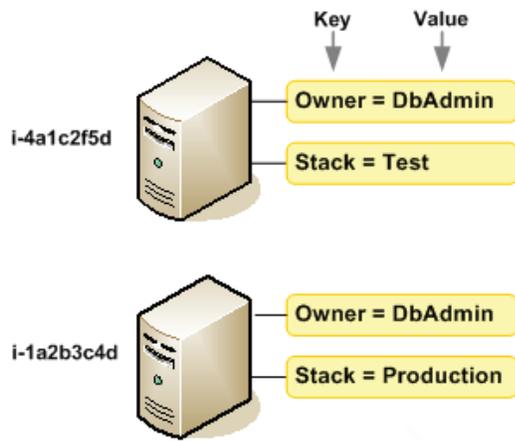


Figure 1: Sample Key-Value Pair

Below are some guidelines to optimize tag implementation:

1. Consider what data is important to the organization. Users typically want to tag resources with multiple Keys – AWS allows 10. A starting list for Keys could be Business Unit, Department, Application/Product Name, Version, Cost Center, Environment, and Owner. This list should be tailored to and reflective of each organization’s needs.

2. After determining the appropriate minimum Keys for the organization, require that users employ the minimum Keys. Users can certainly

employ additional tags, but accurate resource, control, and cost reporting will depend upon universal usage of the organization’s mandatory tags.

3. Remain consistent in your tagging conventions. Create a set schema with a well-defined nomenclature. Ensure that users are aware of the assigned Keys and also set values. For example, make it clear that Env is the Key for Environment and that Prod is the standard value for Production. Remind users that Tags are case sensitive.

4. Automate tagging as much as possible. For users launching resources programmatically, this may be easier to incorporate. However, tagging is equally essential for users who launch resources manually and the greater the “automation”, the more likely manual users will adhere to the organizational tagging policy. Whether you tag programmatically or manually, identify which tag keys are the most important and apply those tags across your infrastructure consistently.

5. Periodically evaluate your infrastructure. Even the best designed tagging schema will fail if users do not consistently employ it. Organizations need to remain vigilant in tag monitoring and enforcement.

Using CloudCheckr to Leverage Tagging

CloudCheckr provides sophisticated reporting across cost, usage, compliance, and security based on AWS tagging.

Tagging Rule Implementation and Enforcement: CloudCheckr provides multiple reports to assist in both implementing and enforcing tagging rules. Tagged and untagged resource reports allow users to instantly view and sort resources. Users can also define

their own tagging rules and CloudCheckr will automatically scan and alert on violations. Users can rely on these and other CloudCheckr features to ensure that their schema is both fits their business needs and is being followed.

Cost Reporting: CloudCheckr's cost reporting allows users to create detailed cost reports tied to individual and groups of both Keys and Values. Budget alerts can be created to ensure that users remain within budget allocations. Cadenced reporting to users, managers, and finance can be automated through pre-set filters. Sophisticated cost tracking and allocation within and across an organization's AWS accounts can be automated by using CloudCheckr.

Usage: CloudCheckr enables users to group resources through tags. Users can employ over 150 prepackaged reports or create their own to track utilization metrics. Alerts and reports can be created to identify areas of over and under provisioned resources. By grouping the resources according the organization's tagging schema, the user can immediately identify whether their infrastructure requires adjustment. This enables user to both improve performance and reduce costs.

Compliance and Security: CloudCheckr performs automated scans of its customer's infrastructure. The results are compared against nearly 300 compliance and security checks. Tagging enables users to receive customized reports highlighting exceptions and changes to their infrastructure. CloudCheckr also allows user to also create CloudTrail alerts based upon resource tagging. This enables organizations to confidently allow self-provisioning while retaining visibility into their security and compliance requirements.

Additional Resources

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html

http://media.amazonwebservices.com/AWS_Operational_Checklists.pdf