Guide for Tagging/Labeling Cloud Resources across CSPs for Internal usage

See also: Division/Org Guide - Internal Elastic Cloud Resources

Purpose of this Guide

As we are rolling out an Elastic-wide policy for internal usage environments to tag/label cloud resources for better visibility into internal usage, we are creating this “User Guide” to provide a quick set of instructions on how to tag your cloud resources. As you can create your resources via UI or CLI, we will walk you through examples to tag/label via both ways on the 3 major CSPs (AWS, GCP & Azure). This may not cover all the ways by which you can create resources (such as API, Terraform, Jenkins, etc.).

General Tagging Requirements

For a team/shared owned resource, Keys requirements are:

- <division> = The value will be from an allowed list of values which are mentioned here.
- <org> = The value will be from an allowed list of values which are mentioned here.
- <team> = The team you are on
- <project> = The project you are working on

  Example:
  division: engineering
  org: *engprod*
  team: *ci*
  project: *buildkite*

For individually owned resource, Key requirements are:

- <division> = The value will be from an allowed list of values which are mentioned here.
- <org> = The value will be from an allowed list of values which are mentioned here.
- <team> = The team you are on
- <project> = Your elastic user name -> Your name with no spaces

  Example:
  division: engineering
  org: *engprod*
  team: *ci*
  project: *staceykingpoling*

Identify Non-Compliant Resources

1. You can access this [Tagging] Resources Details Dashboard
2. Use the “Account ID” filter box to restrict the results to the GCP Project ID or AWS Account ID which are relevant for you or your team.
3. In the **“List of Non-Compliant Resources”** table below, you find all resource names which are non-compliant. The column **“Non compliant labels”** lists which labels are missing for the resource.
4. Use the subsequent sections of this guide to add the missing labels to the resource.

⚠️Values are case-sensitive!

AWS

Add tags to an existing resource

UI

1. Open the Amazon EC2 console.
2. Select the region (from the navigation bar) where you need to add tags to an existing resource.
3. Select a resource type you want to add tags for (for example: Instances).
4. Select the resource from the list of resources and choose the “Tags” tab.
5. Click on Manage Tags and add the tags with “key & value” pair. When finished, click on “Save”.

CLI

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=division,Value=engineering Key=org,Value=engprod
Key=team,Value=ci  Key=project,Value=buildkite
```

Add tags to a group of existing resources

UI

1. Open the Amazon EC2 console.
2. Select the region (from the navigation bar) where you need to add tags to already provisioned resources.
3. Choose Tags from the navigation pane.
4. Click on Manage Tags.

5. For Filter: You can select the type of resources want to apply tags for (Example: Instances).
6. Select the check box next to each resource for which you want to add tags
7. Under Add Tag section, add the key & value and click on “Add Tag”.

CLI

```
aws ec2 create-tags \
    --resources i-123456789 i-abcdefgh  \
    --tags Key=division,Value=engineering Key=org,Value=engprod
Key=team,Value=ci Key=project,Value=buildkite
```

Add a tag on resource creation (Example: When you launch an instance)

UI

1. Open the Amazon EC2 console.
2. Select the region (from the navigation bar) where you need to create resources.
3. Under Instances in the navigation pane, click on Instances.
4. Choose Launch Instances to go through the launch wizard to launch an instance.
5. Click on Add additional tags to add key & value to tag the resources.

CLI

The following is an example to show how you can add tags to instances & volumes you may create.

```
aws ec2 run-instances \
    --image-id ami-xxxxx \
    --count 1 \
    --instance-type c6g.medium \
    --tag-specifications
'ResourceType=instance,Tags=[{Key=division,Value=engineering},{Key=or
g,Value=engprod},{Key=team,Value=ci},{Key=project,Value=buildkite}]'
'ResourceType=volume,Tags=[{Key=division,Value=engineering},{Key=org,
Value=engprod},{Key=team,Value=ci},{Key=project,Value=buildkite}]'
```

For more information on using Tags in AWS, please refer to [Tag your Amazon EC2 resources documentation].

GCP

Add labels to an existing resource

UI

1. Open the GCP console.
2. Select the region where you need to add labels to an existing resource.
3. Go to the resource page for the resource you want to add labels for (for example: Instances, Disks).
4. Select the resource for which you want to add label.
5. To expand the labels column, Click “show info panel”
6. In the panel, select “Labels”
7. Click on “Add Label” to add the labels with key & value pair. When finished, click on “Save” button to save the changes.

CLI

```
gcloud compute instances update example-instance \
    --update-labels
division=engineering,org=engprod,team=ci,project=buildkite
```

Add labels to a group of existing resources

UI

1. Open the GCP console
2. Select the region where you need to add labels to an existing created resources.
3. Go to the resource page for the resource you want to add labels for (for example: Instances, Disks)
4. Select the check box next to each resource for which you want to add label
5. To expand the labels column, Click “show info panel”
6. In the panel, select “Labels”
7. Click on “Add Label” to add the labels with key & value pair. When finished, click on “Save” button to save the changes.

CLI

Not Available directly

You can write a shell (bash) script to do:

```
for INSTANCE in "example-instance1" "example-instance2"
```

```
do
    gcloud compute instances update ${INSTANCE} --update-labels
division=engineering,org=engprod,team=ci,project=buildkite
done
```

Add a label on resource creation (Example: When you launch an instance)

UI

1. Open the GCP console
2. Select the region where you need to create a resource.
3. Go to the resource page to create resources (for example: Instances, Disks)
4. Click on “Add Labels” to add the labels with key & value pairs. When finished, click on the “Save” button to save the changes.

CLI

The following is an example to show how you can add tags to instances & volumes you may create.

```
gcloud compute instances create … \
    --labels
division=engineering,org=engprod,team=ci,project=buildkite
```

For more information on using Tags in GCP, please refer to Labeling GCP resources documentation.

Azure

Add tags to an existing resource

UI

1. Open the Azure console.
2. Go to the resource page for the resource you want to add tags for (for example:Virtual Machines).
3. Select the resource for which you want to add tag.
4. Click on “Tags” in the left pane.
5. Add the key (aka name) & value pair you want to add for the tags. When finished, click on “Apply” button to apply the changes.

CLI (To add tags to a resource which doesn’t have any tags)

```
az tag create --resource-id resource-1234 --tags division=engineering org=engprod team=ci project=buildkite
```

CLI (To add tags to a resource which already have any tags)

```
az tag update --resource-id resource-1234 --operation Merge --tags division=engineering org=engprod team=ci project=buildkite
```

Add tags to a group of existing resources

UI

1. Open the Azure console.
2. Go to the resource page for the resource you want to add tags for (for example:Virtual Machines).
3. Select the check box next to each resource for which you want to add tags.
4. Click on “Assign Tags”.
5. Add the key (aka name) & value pair you want to add for the tags. When finished, click on “Save” button to save the changes.

CLI

Not Available directly

Saj Sasi provided a quick shell script to do something similar:

```
for id in `az resource list -g ${AZ_RG} --query "[].id" -o tsv`;
do
```

```
    az tag create --resource-id ${id} --tags division=engineering
org=engprod team=ci project=buildkite
done
```

Add a tag on resource creation (Example: When you launch an instance)

UI

1. Open the Azure console.
2. Go to the resource page to create resources (For example: Virtual Machines).
3. Click on “Tags” to add the tags with key (aka name) & value pairs. When finished, click on the “Review + Create” button to create the resources.

CLI

The following is an example to show how you can add tags to instances you may create.

```
az vm create … \
    --tags division=engineering org=engprod team=ci project=buildkite
```

For more information on using Tags in Azure, please refer to Use Tags to organize Azure resources.

FAQs

1. Which Cloud Service Providers (CSPs) need to conform with the tagging policy?
   All CSPs (AWS, GCP & Azure) which are used to launch internal resources.

2. Which internal environments need to conform with the tagging policy?
   All internal environments (R&D, QA, Staging) which launch any internal resources should conform with the tagging policy.

3. Which resources will be stopped/impacted as part of not following the tagging policy?
   Although, ALL of the internally launched resources should conform with the tagging policy. However, for the initial phase, the most critical resources to be tagged that will be impacted by the cleanup will be VMs, IPs, Storage Volumes (Attached Storage, Root volumes, Local disks etc) and Disk Snapshots.

   You can quickly validate if your VM is affected by checking the following Kibana dashboard: [Tagging] Resources Details. We recommend that you filter by your account ID, i.e. GCP Project ID or AWS Account ID. If you see your resources on this list, it is non-compliant with the tagging policy and

PROPOSAL: Handling Reorgs

Reorgs are inevitable. The goal for tagging resources is to tie all CSP spending to an organization (`org` tag) leader, and give those leaders the information to validate any spending within that org.

Ultimately, we don't *need* to do anything. Each `org` tag has a cognizant leader, and as long as that leader is OK with the spending, there's no requirement that tags get changed. We *should* provide best practices and automation for updating the tags for resources so that this isn't a painful process for anyone, and so that our data continues to be useful.

Any time there is a reorganization that affects division or organization tags, we'll follow this process:

1. **Identify new division or org tags** (as necessary on a case-by-case basis):
    - If there's a new organization, what should the new tag value(s) be? In this case, we can't know which resources with other tags need to be re-tagged, but this isn't an issue because all previous tags will still be valid.
    - If two or more organizations are combining, should we:
        - use one of the existing org's tags (and migrate resources with the other tags to this one)
        - create a new tag (and migrate resources from all of the tags to the new one)
    - If an organization is going away (becoming part of another within the division), should we combine the tags? Should all resources belonging to the old org be changed to the new org, or do something else?
        - A more complicated scenario is if an organization is being split into two or more different organizations. In that case, can we distinguish existing resources based on team or project tag, or some other method to know which resources with the old org tag should be retagged into which other orgs.
    - If an org is moving between divisions, we'll need to update the division tag; if there's another org within the new division with the same name, do we need a separate value to distinguish them, or are they becoming the same org?

    There are probably variations I haven't thought of here. In general, work with the stakeholders to identify how the ownership (and therefore tags) of resources from the old org should be changed to reflect the reorganization.

2. **Update the documentation with the new tag values.** Ideally, tags should be deprecated (and allowed for some period of time) and not reused to refer to a new/different team than was used before. Include the new values and mark the old values as "deprecated". Update the Custodian configuration to allow the new values, but do not remove the old values (yet).

3. **Identify resources that need to be re-tagged.** We should only touch "manually-tagged" resources. For any resources that were tagged by automation, we need to update the automation as well. If the automation can't update values (for instance, it is configured to only work on instance creation), then we can consider re-tagging these. But if tags are managed in Terraform, for instance, we don't want to make any changes outside of Terraform that will introduce state drift.

4. **Identify any automation that's applying the old tags.**

   NOTE: Can we suggest a new (optional) "tagged-by" tag to both identify manual vs. auto-tagged resources, in order to re-tag only "manually-tagged" resources (which don't have the label). This tag should point to a (code repository/script/CI job/contact person/team) that is responsible for updating the automation. Would it help to build a "registry of taggers" so we can more easily notify folks doing automated tagging with a "feed" of new tag values (and requirements)?

5. Communicate with stakeholders how, when, and which resources will be renamed, as well as any target timelines for action on their part.

6. If asked, run a script (To Be Developed! Can Cloud Custodian help with this?) to re-tag any manually-tagged resources.

7. Monitor for compliance by periodically reporting on resources that use old, deprecated tag values.

8. Eventually, update documentation to indicate that tags are no longer "deprecated" but "formerly-used", and update Cloud Custodian to treat the former tags as invalid.