Metadata Definition Concepts

« (common-image-properties.html) » (glanceapi.html) (https://bugs.launchpad.net/glance/+filebug?field.title=Metadata%20Definition%20Concepts%20in%20glance&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: ______%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A----------------------------------
%0ARelease:%2016.0.1.dev1%20on%20'Thu%20Mar%201%2007:26:57%202018,%20commit%20968f4ae'%0ASHA:%20968f4ae9ce244d9372cb3e8f45acea9d557f317d%0ASour
concepts.rst%0AURL: https://docs.openstack.org/glance/queens/user/metadefs-concepts.html&field.tags=)

UPDATED: 'THU MAR 1 07:26:57 2018, COMMIT 968F4AE'

The metadata definition service was added to Glance in the Juno release of OpenStack.

It provides a common API for vendors, admins, services, and users to meaningfully **define** available key / value pair metadata that can be used on different types of resources (images, artifacts, volumes, flavors, aggregates, and other resources). A definition includes a property's key, its description, its constraints, and the resource types to which it can be associated.

This catalog does not store the values for specific instance properties.

For example, a definition of a virtual CPU topology property for the number of cores will include the base key to use (for example, cpu_cores), a description, and value constraints like requiring it to be an integer. So, a user, potentially through Horizon, would be able to search this catalog to list the available properties they can add to a flavor or image. They will see the virtual CPU topology property in the list and know that it must be an integer.

When the user adds the property its key and value will be stored in the service that owns that resource (for example, Nova for flavors and in Glance for images). The catalog also includes any additional prefix required when the property is applied to different types of resources, such as "hw_" for images and "hw:" for flavors. So, on an image, the user would know to set the property as "hw_cpu_cores=1".

Terminology¶

Background¶

The term *metadata* can become very overloaded and confusing. This catalog is about the additional metadata that is passed as arbitrary key / value pairs or tags across various artifacts and OpenStack services.

Below are a few examples of the various terms used for metadata across OpenStack services today:

Nova	Cinder	Glance
Flavor • *extra specs* **Host Aggregate** • *metadata* **Servers** • *metadata* • *scheduler_hints* • *tags*	**Volume & Snapshot** • *image metadata* • *metadata* **VolumeType** • *extra specs* • *qos specs*	**Image & Snapshot** • *properties* • *tags*

Catalog Concepts¶

The below figure illustrates the concept terminology used in the metadata definitions catalog:

```
A namespace is associated with 0 to many resource types, making it visible to
the API / UI for applying to that type of resource. RBAC Permissions are
managed at a namespace level.

+----------------------------------------------+
|         Namespace                            |
|                                              |
| +-----------------------------------------+  |
| |       Object Definition                 |  |
| |                                         |  |       +--------------------+
| | +-------------------------------------+ |  |  +-->  | Resource Type:     |
| | | Property Definition A (key=integer) | |  |  |     | e.g. Nova Flavor   |
| | +-------------------------------------+ |  |  |     +--------------------+
| |                                         |  |  |
| | +-------------------------------------+ |  |  |
| | | Property Definition B (key=string)  | |  |  |     +--------------------+
| | +-------------------------------------+ |  +--+-->  | Resource Type:     |
| |                                         |  |  |     | e.g. Glance Image  |
| +-----------------------------------------+  |  |     +--------------------+
|                                              |  |
|  +-------------------------------------+     |  |
|  | Property Definition C (key=boolean) |     |  |     +--------------------+
|  +-------------------------------------+     |  +-->  | Resource Type:     |
|                                              |        | e.g. Cinder Volume |
+----------------------------------------------+        +--------------------+

 Properties may be defined standalone or within the context of an object.
```

Catalog Terminology¶

The following terminology is used within the metadata definition catalog.

Namespaces

Metadata definitions are contained in namespaces.

- Specify the access controls (CRUD) for everything defined in it. Allows for admin only, different projects, or the entire cloud to define and use the definitions in the namespace
- Associates the contained definitions to different types of resources

Properties

A property describes a single property and its primitive constraints. Each property can ONLY be a primitive type:

- string, integer, number, boolean, array

Each primitive type is described using simple JSON schema notation. This means NO nested objects and no definition referencing.

Objects

An object describes a group of one to many properties and their primitive constraints. Each property in the group can ONLY be a primitive type:

- string, integer, number, boolean, array

Each primitive type is described using simple JSON schema notation. This means NO nested objects.

The object may optionally define required properties under the semantic understanding that a user who uses the object should provide all required properties.

Resource Type Association

Resource type association specifies the relationship between resource types and the namespaces that are applicable to them. This information can be used to drive UI and CLI views. For example, the same namespace of objects, properties, and tags may be used for images, snapshots, volumes, and flavors. Or a namespace may only apply to images.

Resource types should be aligned with Heat resource types whenever possible. https://docs.openstack.org/heat/latest/template_guide/openstack.html (https://docs.openstack.org/heat/latest/template_guide/openstack.html)

It is important to note that the same base property key can require different prefixes depending on the target resource type. The API provides a way to retrieve the correct property based on the target resource type.

Below are a few examples:

The desired virtual CPU topology can be set on both images and flavors via metadata. The keys have different prefixes on images than on flavors. On flavors keys are prefixed with `hw:`, but on images the keys are prefixed with `hw_`.

For more: https://github.com/openstack/nova-specs/blob/master/specs/juno/implemented/virt-driver-vcpu-topology.rst (https://github.com/openstack/nova-specs/blob/master/specs/juno/implemented/virt-driver-vcpu-topology.rst)

Another example is the AggregateInstanceExtraSpecsFilter and scoped properties (e.g. properties with something:something=value). For scoped / namespaced properties, the AggregateInstanceExtraSpecsFilter requires a prefix of “aggregate_instance_extra_specs:” to be used on flavors but not on the aggregate itself. Otherwise, the filter will not evaluate the property during scheduling.

So, on a host aggregate, you may see:

companyx:fastio=true

But then when used on the flavor, the AggregateInstanceExtraSpecsFilter needs:

aggregate_instance_extra_specs:companyx:fastio=true

In some cases, there may be multiple different filters that may use the same property with different prefixes. In this case, the correct prefix needs to be set based on which filter is enabled.

« (common-image-properties.html) » (glanceapi.html) (https://bugs.launchpad.net/glance/+filebug?field.title=Metadata%20Definition%20Concepts%20in%20glance&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: ______%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A-----------------------------------
%0ARelease:%2016.0.1.dev1%20on%20'Thu%20Mar%201%2007:26:57%202018,%20commit%20968f4ae'%0ASHA:%20968f4ae9ce244d9372cb3e8f45acea9d557f317d%0ASo
concepts.rst%0AURL: https://docs.openstack.org/glance/queens/user/metadefs-concepts.html&field.tags=)

UPDATED: 'THU MAR 1 07:26:57 2018, COMMIT 968F4AE'

FOUND AN ERROR? REPORT A BUG (HTTPS://BUGS.LAUNCHPAD.NET/GLANCE/+FILEBUG?FIELD.TITLE=METADATA%20DEFINITION%20CONCEPTS%20IN%20GLANCE&FIELD.COMMENT=%0A%0A%0ATHIS BUG TRACKER IS FOR ERRORS WITH THE DOCUMENTATION, USE THE FOLLOWING AS A TEMPLATE AND REMOVE OR ADD FIELDS AS YOU SEE FIT. CONVERT [ ] INTO [X] TO CHECK BOXES:%0A%0A- [ ] THIS DOC IS INACCURATE IN THIS WAY: ______%0A- [ ] THIS IS A DOC ADDITION REQUEST.%0A- [ ] I HAVE A FIX TO THE DOCUMENT THAT I CAN PASTE BELOW INCLUDING EXAMPLE: INPUT AND OUTPUT. %0A%0AIF YOU HAVE A TROUBLESHOOTING OR SUPPORT ISSUE, USE THE FOLLOWING RESOURCES:%0A%0A - ASK OPENSTACK: HTTP://ASK.OPENSTACK.ORG%0A - THE MAILING LIST: HTTP://LISTS.OPENSTACK.ORG%0A - IRC: 'OPENSTACK' CHANNEL ON FREENODE%0A%0A-----------------------------------
%0ARELEASE:%2016.0.1.DEV1%20ON%20'THU%20MAR%201%2007:26:57%202018,%20COMMIT%20968F4AE'%0ASHA:%20968F4AE9CE244D9372CB3E8F45ACEA9D557F317D%0ASOURCE:%20HTTPS:/,
CONCEPTS.RST%0AURL: HTTPS://DOCS.OPENSTACK.ORG/GLANCE/QUEENS/USER/METADEFS-CONCEPTS.HTML&FIELD.TAGS=)

QUESTIONS? (HTTP://ASK.OPENSTACK.ORG)

OpenStack Documentation ▾

Page Contents

OpenStack

- Projects (http://openstack.org/projects/)
- OpenStack Security (http://openstack.org/projects/openstack-security/)
- Common Questions (http://openstack.org/projects/openstack-faq/)