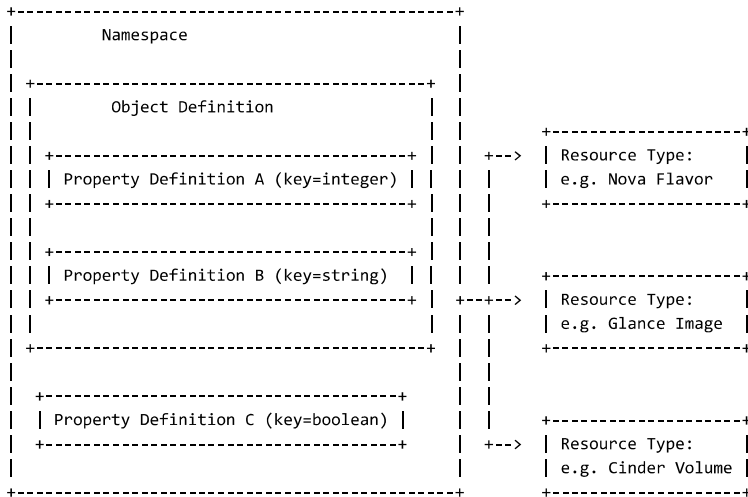




A namespace **is** associated **with** 0 to many resource types, making it visible to the API / UI **for** applying to that type of resource. RBAC Permissions are managed at a namespace level.



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

## Catalog Terminology<sup>¶</sup>

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) ([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
```

[« \(common-image-properties.html\)](#)
[» \(glanceapi.html\)](#)

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



Except where otherwise noted, this document is licensed under [Creative Commons Attribution 3.0 License \(https://creativecommons.org/licenses/by/3.0/\)](https://creativecommons.org/licenses/by/3.0/). See all [OpenStack Legal Documents \(http://www.openstack.org/legal\)](http://www.openstack.org/legal).

**?** QUESTIONS? ([HTTP://ASK.OPENSTACK.ORG](http://ask.openstack.org))



OpenStack Documentation ▼

(../index.html)

User guide (index.html)

- Image Identifiers ([identifiers.html](#))
- Image Statuses ([statuses.html](#))
- Task Statuses ([statuses.html#task-statuses](#))
- Disk and Container Formats ([formats.html](#))
- Common Image Properties ([common-image-properties.html](#))
- Metadata Definition Concepts
- Using Glance's Image Public APIs ([glanceapi.html](#))
- Using Glance's Client Tools ([glanceclient.html](#))
- Using Glance's Metadata Definitions Catalog Public APIs ([glancemetadefcatalogapi.html](#))
- Image Signature Verification ([signature.html](#))

Administration guide (../admin/index.html)

Installation (../install/index.html)

Glance Configuration Options (../configuration/index.html)

Command Line Interface ([../cli/index.html](#))

Glance Contribution Guidelines ([../contributor/index.html](#))

[Glossary \(../glossary.html\)](#)

## Page Contents

## Terminology

- Background
- Catalog Concepts
- Catalog Terminology

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