

# Images

Creates, lists, shows, updates, deletes, and performs other operations on images.

## General information

### Images

An *image* is represented by a JSON Object, that is, as a set of key:value pairs. Some of these keys are *base properties* that are managed by the Image service. The remainder are properties put on the image by the operator or the image owner.

#### Note

Another common term for “image properties” is “image metadata” because what we’re talking about here are properties that *describe* the image data that can be consumed by various OpenStack services (for example, by the Compute service to boot a server, or by the Volume service to create a bootable volume).

Here’s some important information about image properties:

- The base properties are always included in the image representation. A base property that doesn’t have a value is displayed with its value set to `null` (that is, the JSON null data type).
- Additional properties, whose value is always a string data type, are only included in the response if they have a value.
- Since version 2.2, the Images API allows an operator to configure *property protections*, by which the create, read, update, and delete operations on specific image properties may be restricted to particular user roles. Consult the documentation of your cloud operator for details.
- Arguably the most important properties of an image are its *id*, which uniquely identifies the image, its *status*, which indicates the current situation of the image (which, in turn, indicates what you can do with the image), and its *visibility*, which indicates who has access to the image.

#### Note

In addition to image properties, there’s usually a data payload that is accessible via the image. In order to give image consumers some guarantees about the data payload (for example, that the data associated with image `06b73bc7-9d62-4d37-ad95-d4708f37734f` is the same today as it was when you used it to boot a server yesterday) the Image service controls particular image properties (for example, `checksum`) that cannot be modified. A shorthand way to refer to the way the image data payload is related to its representation as an *image* in the Images API is to say that “images are immutable”. (This obviously applies to the image data payload, not its representation in the Image service.) See the [Image Data](#) section of this document for more information.

### Image status

The possible status values for images are presented in the following table.

Status	Description
queued	The Image service reserved an image ID for the image in the catalog but did not yet upload any image data.
saving	The Image service is in the process of saving the raw data for the image into the backing store.
active	The image is active and ready for consumption in the Image service.
killed	An image data upload error occurred.
deleted	The Image service retains information about the image but the image is no longer available for use.
pending_delete	Similar to the <code>deleted</code> status. An image in this state is not recoverable.
deactivated	The image data is not available for use.
uploading	Data has been staged as part of the interoperable image import process. It is not yet available for use. <i>(Since Image API 2.6)</i>
importing	The image data is being processed as part of the interoperable image import process, but is not yet available for use. <i>(Since Image API 2.6)</i>

### Image visibility

The possible values for image visibility are presented in the following table.

Visibility	Description
<code>public</code>	Any user may read the image and its data payload. Additionally, the image appears in the default image list of all users.
<code>community</code>	Any user may read the image and its data payload, but the image does <i>not</i> appear in the default image list of any user other than the owner. <i>(This visibility value was added in the Image API v2.5)</i>
<code>shared</code>	An image must have this visibility in order for <i>image members</i> to be added to it. Only the owner and the specific image members who have been added to the image may read the image or its data payload. The image appears in the default image list of the owner. It also appears in the default image list of members who have <i>accepted</i> the image. See the <a href="#">Image Sharing</a> section of this document for more information. If you do not specify a visibility value when you create an image, it is assigned this visibility by default. Non-owners, however, will not have access to the image until they are added as image members. <i>(This visibility value was added in the Image API v2.5)</i>
<code>private</code>	Only the owner image may read the image or its data payload. Additionally, the image appears in the owner's default image list. <i>Since Image API v2.5, an image with private visibility cannot have members added to it.</i>

Note that the descriptions above discuss *read* access to images. Only the image owner (or an administrator) has write access to image properties and the image data payload. Further, in order to promise image immutability, the Image service will allow even the owner (or an administrator) only write-once permissions to specific image properties and the image data payload.

## Show image GET /v2/images/{image\_id}

Shows details for an image. *(Since Image API v2.0)*

The response body contains a single image entity.

Preconditions

- The image must exist.

Normal response codes: 200

Error response codes: 400, 401, 403, 404

### Request

Name	In	Type	Description
image_id	path	string	The UUID of the image.

### Response Parameters

Name	In	Type	Description
------	----	------	-------------

checksum	body	string	Hash that is used over the image data. The Image service uses this value for verification. The value might be <code>nu11</code> (JSON null data type).
container_format	body	enum	Format of the image container.Values may vary based on the configuration available in a particular OpenStack cloud. See the <a href="#">Image Schema</a> response from the cloud itself for the valid values available.Example formats are: <code>ami</code> , <code>ari</code> , <code>aki</code> , <code>bare</code> , <code>ovf</code> , <code>ova</code> , or <code>docker</code> .The value might be <code>nu11</code> (JSON null data type).
created_at	body	string	The date and time when the resource was created.The date and time stamp format is <a href="#">ISO 8601</a> : <code>CCYY-MM-DDThh:mm:ss±hh:mm</code> For example, <code>2015-08-27T09:49:58-05:00</code> .The <code>±hh:mm</code> value, if included, is the time zone as an offset from UTC.
disk_format	body	enum	The format of the disk.Values may vary based on the configuration available in a particular OpenStack cloud. See the <a href="#">Image Schema</a> response from the cloud itself for the valid values available.Example formats are: <code>ami</code> , <code>ari</code> , <code>aki</code> , <code>vhd</code> , <code>vhdx</code> , <code>vmdk</code> , <code>raw</code> , <code>qcow2</code> , <code>vdi</code> , <code>ploop</code> or <code>iso</code> .The value might be <code>nu11</code> (JSON null data type). <b>Newton changes:</b> The <code>vhdx</code> disk format is a supported value. <b>Ocata changes:</b> The <code>ploop</code> disk format is a supported value.
file	body	string	The URL for the virtual machine image file.
id	body	string	A unique, user-defined image UUID, in the format: <code>nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn</code> Where <b>n</b> is a hexadecimal digit from 0 to f, or F.For example: <code>b2173dd3-7ad6-4362-baa6-a68bce3565cb</code> If you omit this value, the API generates a UUID for the image.
min_disk	body	integer	Amount of disk space in GB that is required to boot the image. The value might be <code>nu11</code> (JSON null data type).
min_ram	body	integer	Amount of RAM in MB that is required to boot the image. The value might be <code>nu11</code> (JSON null data type).
name	body	string	The name of the image. Value might be <code>nu11</code> (JSON null data type).
os_hash_algo	body	string	The algorithm used to compute a secure hash of the image data for this image. The result of the computation is displayed as the value of the <code>os_hash_value</code> property. The value might be <code>nu11</code> (JSON null data type). The algorithm used is chosen by the cloud operator; it may not be configured by end users. <i>(Since Image API v2.7)</i>
os_hash_value	body	string	The hexdigest of the secure hash of the image data computed using the algorithm whose name is the value of the <code>os_hash_algo</code> property. The value might be <code>nu11</code> (JSON null data type) if data has not yet been associated with this image, or if the image was created using a version of the Image Service API prior to version 2.7. <i>(Since Image API v2.7)</i>
			This field controls whether an image is displayed in the default image-list response. A “hidden” image is out of date somehow (for

os_hidden	body	boolean	example, it may not have the latest updates applied) and hence should not be a user's first choice, but it's not deleted because it may be needed for server rebuilds. By hiding it from the default image list, it's easier for end users to find and use a more up-to-date version of this image. <i>(Since Image API v2.7)</i>
owner	body	string	An identifier for the owner of the image, usually the project (also called the "tenant") ID. The value might be <code>null</code> (JSON null data type).
protected	body	boolean	A boolean value that must be <code>false</code> or the image cannot be deleted.
schema	body	string	The URL for the schema describing a virtual machine image.
self	body	string	The URL for the virtual machine image.
size	body	integer	The size of the image data, in bytes. The value might be <code>null</code> (JSON null data type).
status	body	string	The image status.
tags	body	array	List of tags for this image, possibly an empty list.
updated_at	body	string	The date and time when the resource was updated. The date and time stamp format is <a href="#">ISO 8601</a> : CCYY-MM-DDThh:mm:ss±hh:mm For example, <code>2015-08-27T09:49:58-05:00</code> . The <code>±hh:mm</code> value, if included, is the time zone as an offset from UTC. In the previous example, the offset value is <code>-05:00</code> . If the <code>updated_at</code> date and time stamp is not set, its value is <code>null</code> .
virtual_size	body	integer	The virtual size of the image. The value might be <code>null</code> (JSON null data type).
visibility	body	string	Image visibility, that is, the access permission for the image.
direct_url (Optional)	body	string	The URL to access the image file kept in external store. <i>It is present only if the <code>show_image_direct_url</code> option is <code>true</code> in the Image service's configuration file. <b>Because it presents a security risk, this option is disabled by default.</b></i>
locations (Optional)	body	array	A list of objects, each of which describes an image location. Each object contains a <code>url</code> key, whose value is a URL specifying a location, and a <code>metadata</code> key, whose value is a dict of key:value pairs containing information appropriate to the use of whatever external store is indicated by the URL. <i>This list appears only if the <code>show_multiple_locations</code> option is set to <code>true</code> in the Image service's configuration file. <b>Because it presents a security risk, this option is disabled by default.</b></i>

The response may also include additional properties specified as key:value pairs if such properties have been added to the image by the owner or an administrator.

## Response Example

```
{
  "status": "active",
```

```

"name": "cirros-0.3.2-x86_64-disk",
"tags": [],
"container_format": "bare",
"created_at": "2014-05-05T17:15:10Z",
"disk_format": "qcow2",
"updated_at": "2014-05-05T17:15:11Z",
"visibility": "public",
"self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
"min_disk": 0,
"protected": false,
"id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
"file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
"checksum": "64d7c1cd2b6f60c92c14662941cb7913",
"os_hash_algo": "sha512",
"os_hash_value":
"073b4523583784fbe01daff81eba092a262ec37ba6d04dd3f52e4cd5c93eb8258af44881345ec
da0e49f3d8cc6d2df6b050ff3e72681d723234aff9d17d0cf09",
"os_hidden": false,
"owner": "5ef70662f8b34079a6eddb8da9d75fe8",
"size": 13167616,
"min_ram": 0,
"schema": "/v2/schemas/image",
"virtual_size": null
}

```

## List images GET /v2/images

Lists public virtual machine (VM) images. (*Since Image API v2.0*)

### Pagination

Returns a subset of the larger collection of images and a link that you can use to get the next set of images. You should always check for the presence of a `next` link and use it as the URI in a subsequent HTTP GET request. You should follow this pattern until a `next` link is no longer provided.

The `next` link preserves any query parameters that you send in your initial request. You can use the `first` link to jump back to the first page of the collection. If you prefer to paginate through images manually, use the `limit` and `marker` parameters.

### Query Filters

The list operation accepts query parameters to filter the response.

A client can provide direct comparison filters by using most image attributes, such as `name=Ubuntu`, `visibility=public`, and so on.

To filter using image tags, use the filter `tag` (note the singular). To filter on multiple tags, include each tag separately in the query. For example, to find images with the tag **ready**, include `tag=ready` in your query string. To find images tagged with **ready** and **approved**, include `tag=ready&tag=approved` in your query string. (Note that only images containing *both* tags will be included in the response.)

A client cannot use any `link` in the json-schema, such as `self`, `file`, or `schema`, to filter the response.

You can list VM images that have a status of `active`, `queued`, or `saving`.

## The in Operator

As a convenience, you may specify several values for any of the following fields by using the `in` operator:

- `container_format`
- `disk_format`
- `id`
- `name`
- `status`

For most of these, usage is straight forward. For example, to list images in `queued` or `saving` status, use:

**GET /v2/images?status=in:saving,queued**

To find images in a particular list of image IDs, use:

**GET /v2/images?id=in:3afb79c1-131a-4c38-a87c-bc4b801d14e6,2e011209-660f-44b5-baf2-2eb4babae53d**

Using the `in` operator with the `name` property of images can be a bit trickier, depending upon how creatively you have named your images. The general rule is that if an image name contains a comma ( , ), you must enclose the entire name in quotation marks ( " ). As usual, you must URL encode any characters that require it.

For example, to find images named `glass`, `darkly` or `share me`, you would use the following filter specification:

**GET v2/images?name=in:"glass,%20darkly",share%20me**

As with regular filtering by name, you must specify the complete name you are looking for. Thus, for example, the query string `name=in:glass,share` will only match images with the exact name `glass` or the exact name `share`. It will not find an image named `glass, darkly` or an image named `share me`.

## Size Comparison Filters

You can use the `size_min` and `size_max` query parameters to filter images that are greater than or less than the image size. The size, in bytes, is the size of an image on disk.

For example, to filter the container to include only images that are from 1 to 4 MB, set the `size_min` query parameter to `1048576` and the `size_max` query parameter to `4194304`.

## Time Comparison Filters

You can use a *comparison operator* along with the `created_at` or `updated_at` fields to filter your results. Specify the operator first, a colon ( `:` ) as a separator, and then the time in [ISO 8601 Format](#). Available comparison operators are:

Operator	Description
<code>gt</code>	Return results more recent than the specified time.
<code>gte</code>	Return any results matching the specified time and also any more recent results.
<code>eq</code>	Return any results matching the specified time exactly.
<code>neq</code>	Return any results that do not match the specified time.
<code>lt</code>	Return results older than the specified time.
<code>lte</code>	Return any results matching the specified time and also any older results.

For example:

```
GET v2/images?created_at=gt:2016-04-18T21:38:54Z
```

## Sorting

You can use query parameters to sort the results of this operation.

- `sort_key`. Sorts by an image attribute. Sorts in the natural sorting direction of the image attribute.
- `sort_dir`. Sorts in a sort direction.
- `sort`. Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, the default is `desc`.

To sort the response, use the `sort_key` and `sort_dir` query parameters:

```
GET /v2/images?sort_key=name&sort_dir=asc&sort_key=status&sort_dir=desc
```

Alternatively, specify the `sort` query parameter:

```
GET /v2/images?sort=name:asc,status:desc
```

Note



Although this call has been available since version 2.0 of this API, it has been enhanced from release to release. The filtering and sorting functionality and syntax described above apply to the most recent release (Newton). Not everything described above will be available in prior releases.

Normal response codes: 200

Error response codes: 400, 401, 403

## Request

Name	In	Type	Description
limit (Optional)	query	integer	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker (Optional)	query	string	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
name (Optional)	query	string	Filters the response by a name, as a string. A valid value is the name of an image.
owner (Optional)	query	string	Filters the response by a project (also called a “tenant”) ID. Shows only images that are shared with you by the specified owner.
protected (Optional)	query	boolean	Filters the response by the ‘protected’ image property. A valid value is one of ‘true’, ‘false’ (must be all lowercase). Any other value will result in a 400 response.
status (Optional)	query	integer	Filters the response by an image status.
tag (Optional)	query	string	Filters the response by the specified tag value. May be repeated, but keep in mind that you’re making a conjunctive query, so only images containing <i>all</i> the tags specified will appear in the response.
visibility (Optional)	query	string	Filters the response by an image visibility value. A valid value is <code>public</code> , <code>private</code> , <code>community</code> , or <code>shared</code> . (Note that if you filter on <code>shared</code> , the images included in the response will only be those where your member status is <code>accepted</code> unless you explicitly include a <code>member_status</code> filter in the request.) If you omit this parameter, the response shows <code>public</code> , <code>private</code> , and those <code>shared</code> images with a member status of <code>accepted</code> .
os_hidden (Optional)	query	boolean	When <code>true</code> , filters the response to display only “hidden” images. By default, “hidden” images are not included in the image-list response. (Since Image API v2.7)
member_status (Optional)	query	string	Filters the response by a member status. A valid value is <code>accepted</code> , <code>pending</code> , <code>rejected</code> , or <code>all</code> . Default is <code>accepted</code> .
size_max			

(Optional)	query	string	Filters the response by a maximum image size, in bytes.
size_min (Optional)	query	string	Filters the response by a minimum image size, in bytes.
created_at (Optional)	query	string	Specify a <i>comparison filter</i> based on the date and time when the resource was created. (See <a href="#">Time Comparison Filters</a> ).The date and time stamp format is <a href="#">ISO 8601</a> : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC.For example, 2015-08-27T09:49:58-05:00 .If you omit the time zone, the UTC time zone is assumed.
updated_at (Optional)	query	string	Specify a <i>comparison filter</i> based on the date and time when the resource was most recently modified. (See <a href="#">Time Comparison Filters</a> ).The date and time stamp format is <a href="#">ISO 8601</a> : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC.For example, 2015-08-27T09:49:58-05:00 .If you omit the time zone, the UTC time zone is assumed.
sort_dir (Optional)	query	string	Sorts the response by a set of one or more sort direction and attribute ( <code>sort_key</code> ) combinations. A valid value for the sort direction is <code>asc</code> (ascending) or <code>desc</code> (descending). If you omit the sort direction in a set, the default is <code>desc</code> .
sort_key (Optional)	query	string	Sorts the response by an attribute, such as <code>name</code> , <code>id</code> , or <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the <code>sort_key</code> image attribute.
sort (Optional)	query	string	Sorts the response by one or more attribute and sort direction combinations. You can also set multiple sort keys and directions. Default direction is <code>desc</code> .Use the comma ( , ) character to separate multiple values. For example: <code>GET /v2/images?sort=name:asc,status:desc</code>

## Response Parameters

Name	In	Type	Description
images	body	array	A list of <i>image</i> objects, as described by the <a href="#">Images Schema</a> .
first	body	string	The URI for the first page of response.
next	body	string	The URI for the next page of response. Will not be present on the last page of the response.
schema	body	string	The URL for the schema describing a list of images.

## Response Example

```
{
  "images": [
    {
      "status": "active",
```

```

    "name": "cirros-0.3.2-x86_64-disk",
    "tags": [],
    "container_format": "bare",
    "created_at": "2014-11-07T17:07:06Z",
    "disk_format": "qcow2",
    "updated_at": "2014-11-07T17:19:09Z",
    "visibility": "public",
    "self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
    "min_disk": 0,
    "protected": false,
    "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
    "file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
    "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
    "os_hash_algo": "sha512",
    "os_hash_value":
"073b4523583784fbe01daff81eba092a262ec37ba6d04dd3f52e4cd5c93eb8258af44881345ec
da0e49f3d8cc6d2df6b050ff3e72681d723234aff9d17d0cf09",
    "os_hidden": false,
    "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
    "size": 13167616,
    "min_ram": 0,
    "schema": "/v2/schemas/image",
    "virtual_size": null
},
{
    "status": "active",
    "name": "F17-x86_64-cfntools",
    "tags": [],
    "container_format": "bare",
    "created_at": "2014-10-30T08:23:39Z",
    "disk_format": "qcow2",
    "updated_at": "2014-11-03T16:40:10Z",
    "visibility": "public",
    "self": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c",
    "min_disk": 0,
    "protected": false,
    "id": "781b3762-9469-4cec-b58d-3349e5de4e9c",
    "file": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c/file",
    "checksum": "afab0f79bac770d61d24b4d0560b5f70",
    "os_hash_algo": "sha512",
    "os_hash_value":
"ea3e20140df1cc65f53d4c5b9ee3b38d0d6868f61bbe2230417b0f98cef0e0c7c37f0ebc5c645
6fa47f013de48b452617d56c15fdb25e100379bd0e81ee15ec",
    "os_hidden": false,
    "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
    "size": 476704768,
    "min_ram": 0,
    "schema": "/v2/schemas/image",
    "virtual_size": null

```

```
    }
  ],
  "schema": "/v2/schemas/images",
  "first": "/v2/images"
}
```

## Delete image DELETE /v2/images/{image\_id}

(Since Image API v2.0) Deletes an image.

You cannot delete images with the `protected` attribute set to `true` (boolean).

### Preconditions

- You can delete an image in any status except `deleted`.
- The `protected` attribute of the image cannot be `true`.
- You have permission to perform image deletion under the configured image deletion policy.

### Synchronous Postconditions

- The response is empty and returns the HTTP `204` response code.
- The API deletes the image from the images index.
- If the image has associated binary image data in the storage backend, the OpenStack Image service deletes the data.

Normal response codes: 204

Error response codes: 400, 401, 403, 404, 409

## Request

Name	In	Type	Description
image_id	path	string	The UUID of the image.