openstack.

November 2017

Glance Image Import is here … Now it's time to start using it

Erno Kuvaja, irc: jokke_
Brian Rosmaita, irc: rosmaita

redhat.

Erno Kuvaja

Senior Software Engineer

Brian Rosmaita

Distinguished Member of the Technical Staff
Verizon Wireless

What is new Image Import?

- ➡ Replacement for Image Upload
- ➡ Multiple source options
- ➡ Enabling operator tasks
- ➡ Discoverable

- ➡ Needs configuration
- ➡ New workflow

Replacement for Image Upload

- API changes from 1-2 calls to 2-3 calls
  - Image-create works as before, just informs about the new workflow with the OpenStack-image-import-methods header
  - Providing the data is the optional second call (currently required as other options are not yet implemented). ‘glance-direct’ works as Image upload, does not activate the image automatically. PUT the data to staging.
  - ‘import’ the new call to activate the Image if needed data is provided. May trigger async tasks before image goes active. 202 does not mean successfully activated image, needs to be monitored by caller.

Replacement for Image Upload

- ➡ Command Line Interface
  - provides all the API calls as separate commands:
    - glance image-create
    - glance image-stage
    - glance image-import
  - or can combine them into one create call just like with the previous workflow:
    - glance image-create-via-import
      - Accepts a --file argument or will accept the data from stdin
  - plus, the discovery call:
    - glance import-info

Replacement for Image Upload

- Command Line Interface
  - Main difference between import and upload is that the Image does not go active right at successful response.
  - Monitoring image status going ‘active’ will also be needed when working with the CLI.
    - glance image-show

Multiple source options

- ‘glance-direct’
  - Closest to the current image-upload
  - Saves the image data to staging space before processing
  - Currently only implemented option

Multiple source options

- ‘glance-direct’
  - Closest to the current image-upload
  - Saves the image data to staging space before processing
  - Currently only implemented option

- ‘http-download’
  - The long waited ‘copy-from’ from Images API v1
  - Does not require user to upload the data directly to Glance
  - Implementation timeline Queens release

Multiple source options

- ‘swift-local’
  - Gives possibility to provide user uploaded images without letting them upload the data directly to glance
  - Image data is accessed from the swift container provided in the ‘import’ call
  - Design has been agreed, but the implementation timeline is still open

Operator tasks

- Provides operators hook to provide their own tasks into the taskflow before the image goes active
  - Hooks will be implemented in the Queens release
  - These tasks can include actions such as automatic image conversion, metadata injection for certain groups of users, virus checks, etc.
  - Tasks are asynchronous so that the client does not need to keep the connection to the server while they are executed
  - The tasks are not user selectable. They will be executed on each image import. Task will have the info from request context so it can select different behaviour based on conditions (for example, no-op when owner is admin)

Discoverability

- ➡ New discovery API is included with the Image Import
  - ‘/info/import’ endpoint provides details of the capabilities the installation provides.
    - What import methods are available
    - Any image limitations such as container formats, max size, etc.
  - While this information is available across the API the reasoning for the ‘/info/’ endpoint is to provide one-stop shop for all the discovering for the deployment
- ➡ Current implementation is quite limited but will be extended over time
- ➡ Client will be taking advantage of the data provided for decision making and advance notification of likely failures.

Configuration

Taking full advantage of the Image Import will need extensive configuration and understanding of the needs of the deployment

- ➡ If ‘glance-direct’ is enabled, shared filesystem between the nodes will be required for staging (staging utilizes glance_store library and there are plans to support other store types for this such as Ceph)
  - Staging should *never* share path with image-cache or the filesystem image store
  - Sizing of the staging area is very deployment and usage pattern dependant

- ➡ Tasks will likely require workspace in the node. There should be attention paid to decide if this should share the space with staging

Configuration

- The ‘enable_image_import’ configuration option is there only to provide upgrade path and is already deprecated.
  - Pike: False by default, True will enable the EXPERIMENTAL Image Import API
  - Queens: False by default, True will enable the Image Import feature on the CURRENT API
  - Rocky: True by default
  - S-release: option will be removed

Configuration

- ➡ The enabled Import methods are an operator configurable option
  - Either ‘glance-direct’ or ‘swift-local’ will most likely be part of the trademark requirements in the future (deployment needs to expose one of the two)
  - More than one option can be supported at the same time
  - There are no technical reasons to limit the methods to these three

Configuration

- ‘Http-download’ method will have possibilities to filter allowed URIs
  - Whitelisting
  - Blacklisting
  - Limited to certain ports only (this will be defaulting to 80 and 443) to address https://wiki.openstack.org/wiki/OSSN/OSSN-0078

New workflow

- Special attention to tools developers who are consuming the API
  - The methods may vary but GET to '/info/import' will always provide the available methods on the deployment
  - The current Image Upload workflow will likely to be limited for admin/service internal use only
  - On the positive side, one way to create images will be available on the clouds that carry OpenStack Trademark
  - The image data may change between upload and download! For example if conversion is implemented. The immutability of the images still apply for all active images (the content won't change once the image has gone to status 'active').

Current challenges

- Devstack
  - The feature does not currently work on devstack Pike or master
  - Can be worked around by installing Ocata and upgrading the Glance to Pike or master
- Only file store currently supported for staging
  - There must be shared filesystem (such as NFS) available for glance-api nodes
- The tasks hooks does not exists yet

openstack

Q&A

Thank you!

 openstack  @OpenStack  openstack  OpenStackFoundation