openstack.

May 2018

OpenStack Manila

Project Update, OpenStack Summit Vancouver

Tom Barron
IRC: tbarron
EMAIL: tbarron@redhat.com, tpb@dyncloud.net

Agenda

What’s Manila

Problem Space, Problem Background

Queens Release Update

API changes, Features, Driver improvements

Rocky and Beyond

Priorities, Themes

Manila?

Manila is the control plane to provision and manage shared filesystems across storage systems.

One popular oversimplification: *Manila is Cinder for file shares.*

Fork of OpenStack Cinder, built by a shared pool of developers, shares much of Cinder’s architecture.

But it solves a new class of problems that Cinder does not naturally address.

Think: manila *file folders*

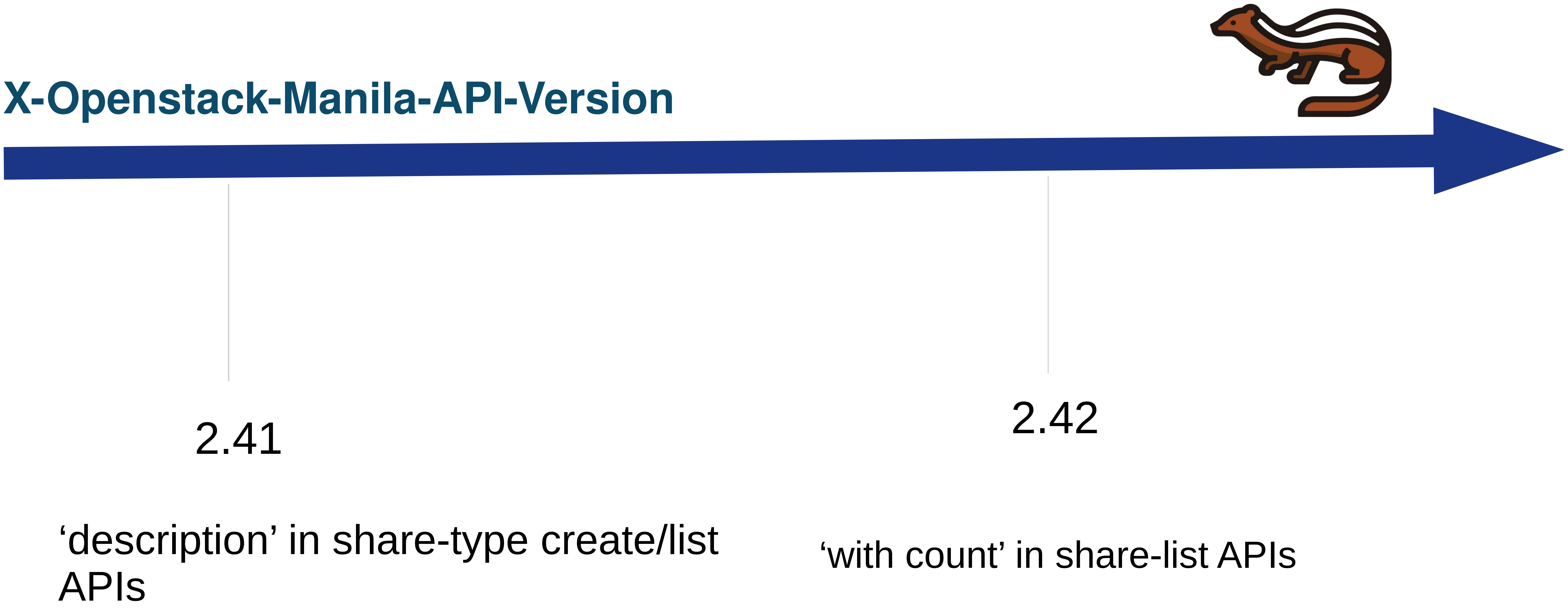
Manila?

- Consistent, simple REST API
- Open Source drivers & drivers for proprietary vendor back ends (close to 30 now)
- Multiple NAS protocols
- Queens release had 41 unique reviewers and 44 unique contributors representing 21 companies
- 7th production quality release
- Growing adoption (Ocata user survey: 14%)
- Increasing development activity and use outside manila proper in TripleO, Kolla, Charms, gophercloud, gophercloud, etc. to integrate and deploy manila both inside and outside of OpenStack proper

Queens

17th release of OpenStack / 7th official release of Manila

API Changes



X-Openstack-Manila-API-Version

2.41

‘description’ in share-type create/list APIs

2.42

‘with count’ in share-list APIs

New in Queens – IPv6 gets real

Pike added support for IPv6 exports and access rules

But lacked driver support except for the LVM back end.

* NetApp

* DellEMC Unity

* DellEMC VNX

* fixed some nasty bugs

* IPv6 scenario tests in upstream CI

New in Queens – get total count info in share list APIs

- Now we don’t have to list all the instances of the shares for a user in order to show how much they are using.
- Real world use case (Huawei public cloud)
- API change, backwards compatible

New in Queens – smart ensure-share

- When the manila-share service starts up an internal method called ‘ensure-share’ is run to reconcile manila’s view of the back end state as maintained in its database with the back end’s actual state.
- Export locations could have changed – e.g. back end became IPv6 capable.
- Back end software could have been upgraded
- Driver can ensure that back end state is correct or update manila’s state as appropriate.
- Most of the time nothing needs to be done but if the back end has many shares and each one needs to be checked startup can be delayed.
- So we introduced a mechanism where the manager calls get_backend_info() and the back end returns a dictionary of key/values for potential update.
- Manager computes a hash of these and compares it with its last hash. If these match, nothing needs to be done.
- If they don’t match the manager runs ensure_shares() which returns a list of model updates for each share on the back end.

New in Queens – Driver additions and enhancements

- New Infinidat Infinibox driver
- New Veritas Access driver
- Fixed new share size when creating from NetApp snapshot
- Store ganesha exports and export counter in Ceph RADOS
- Huawei driver supports revert-to-snapshot
- QNAP driver added support for thin-provisioning, de-duplication, compression, SSD cache
- NetApp “NVE” (volume encryption at rest) support
- Fixes to make the generic driver more robust

New in Queens – OpenStack community goals

- Policy in code
- Completed migration of doc in-tree (and lots of doc fixes)
- Enabled mutable configuration via oslo (Rocky goal)
- Manila tempest test plugin moved to its own repository
- Zuul v3 migration

OpenStack Rocky – python 3 readiness

- Distributions have started to drop support for python 2
  - Debian has already dropped support, others expected to follow in the next couple years since upstream support for python 2 is going away by 2020
- Manila-ui was not python 3 ready
- Gate jobs need to convert to python 3
  - Replace CentOS jobs with Fedora jobs

OpenStack Rocky – add json schema validation

- Manila currently has no consistent request validation layer
  - Some resources validate input at resource controller
  - Some just fail in the back end
- So add a jsonschema for each resource and use a jsonschema validator object to check each resource against its schema
- If the validation passes, allow the request to follow the existing flow of conrol through the resource manager to the back end.
- If the validation fails, return HTTPBadRequest

OpenStack Rocky – metadata for access rules

- Experience with Huawei public cloud indicated it would be useful to be able to add descriptive metadata to access rules
- Goal: improve user experience when managing access rules
- Users can add key-value pairs to describe access rules
- Users can filter access rules with specific metadata when listing access rules

OpenStack Rocky – access rules priority

- Manila access rules currently support two access levels, read-write and read-only
- IP based access rules can overlap
  - Allow 192.168.1.0/24 r/o
  - Allow 192.168.1.1 rw
  - This is ambiguous: which rule wins for address 192.168.1.1?
  - Manila just left it up to the back end to decide
- Instead, rules have priorities in a range 1-200
- Default is 200 and lower numbers are higher
- User can explicitly set a priority to dis-ambiguate

Rocky and Beyond – some goals

- Need manila in OpenStack SDK, OpenStack client, shade and OpenStack ansible modules!
- Serve manila shares to bare-metal and container-based workloads, in and out of OpenStack
- More production-quality open source software defined back ends for manila
- Finish python 3 work
- More IPv6 back end support (e.g. CephFS)

Cross-project, cross-community

- Better deployment tool integration: tripleO, kolla, charms, etc.
- Manila “standalone”
  - Several senses of standalone, several goals here
- Back up and DR solutions
- Storage “intelligence” and management

How to give feedback

Manila Ops Feedback Forum Session today at 1:50 in room 221-222.
Put your ideas in the etherpad

Also:

- Tag [manila] on the OpenStack developers and operators mail lists
- Find us in #openstack-manila on freenode
- Feel free to reach out to me personally as PTL

How to contribute

Project OnBoarding Session

Today at 10am in room 223

openstack

Q&A

Thank you!

openstack

@OpenStack

openstack

OpenStackFoundation