



OpenStack Manila

An Overview of Manila Liberty & Mitaka

October 7, 2015



SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced in their entirety without modification
 - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Presenters

Ben Swartzlander
NetApp
OpenStack Manila Project Team Lead



Alex McDonald,
NetApp
Vice Chair, SNIA Ethernet Storage Forum (ESF)
Chair, SNIA Cloud Storage Initiative (CSI)

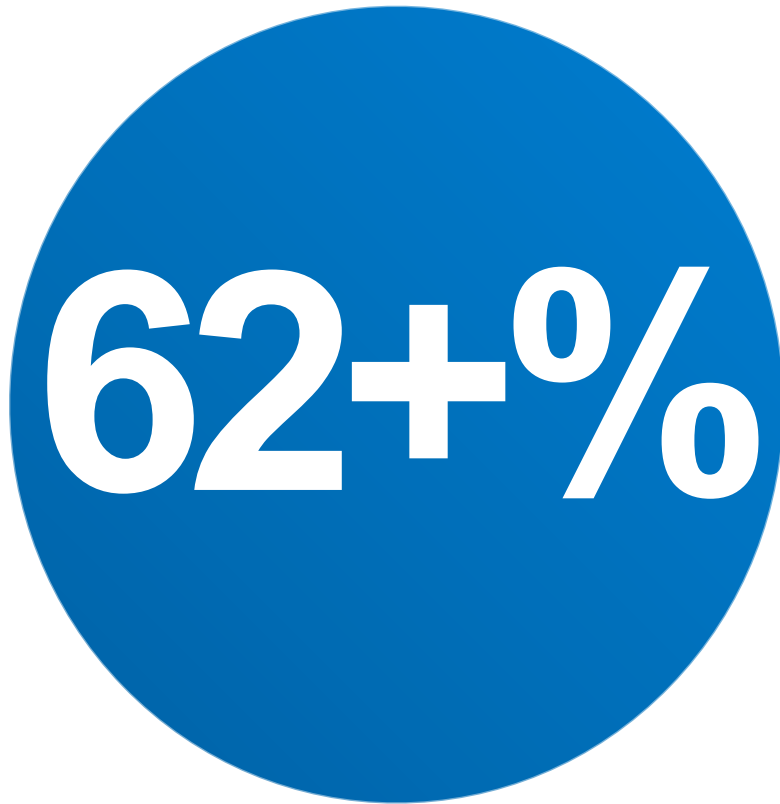


- Manila is the OpenStack file share service that provides the management of file shares (for example, NFS and CIFS) as a core service to OpenStack. Intended to be provide an open standards, highly available and fault tolerant component of OpenStack, Manila also aims to provide API-compatibility with popular systems like Amazon EC2.
- In this session, we'll give you a brief overview of Manila, talk about new features that are being delivered for OpenStack Liberty (due October 2015) and Mitaka beyond that.

Agenda

- Project Overview
- Key Concepts
 - ◆ API Overview
 - ◆ Architecture Overview
 - ◆ Drivers and Integration
 - ◆ Automation
- What was in Kilo?
- What is in Liberty?
- What is Proposed for Mitaka?

Shared File Services

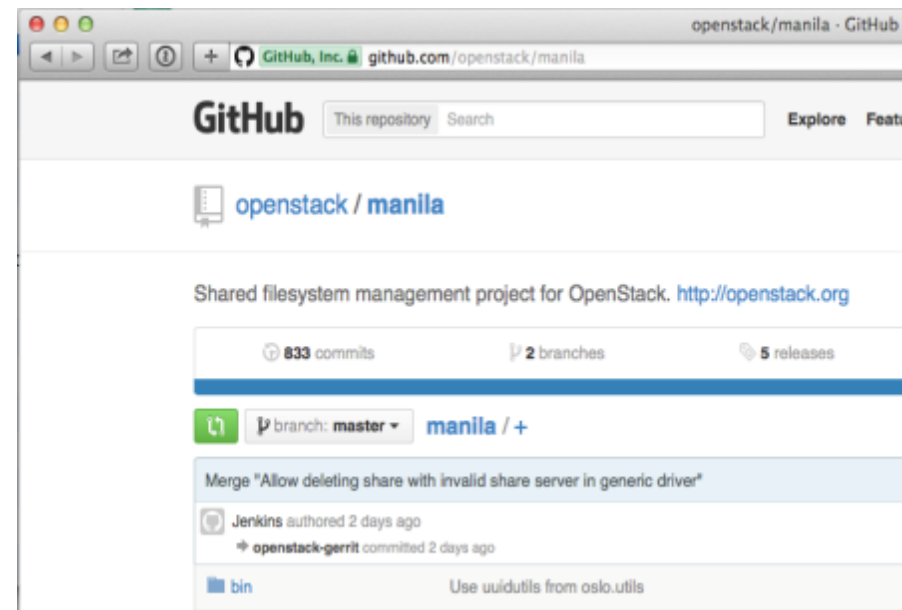


of all storage sold is
for file-based
use cases

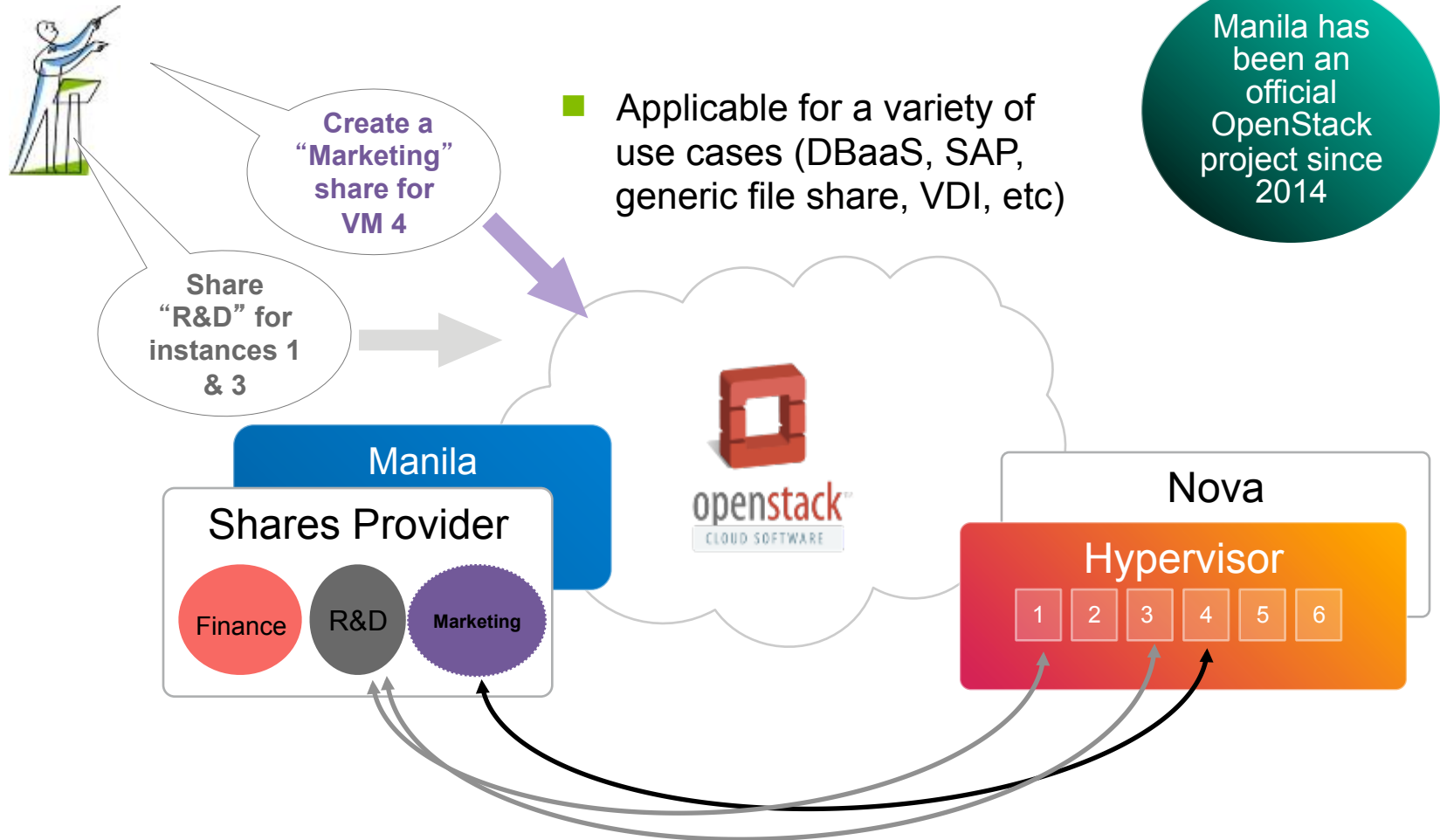
Source: IDC 2014

Manila: Project Overview

- ▶ What's the customer use case for Manila?
 - ◆ 62+% of disk capacity slated for file storage
 - ◆ Self-service management & provisioning of shared file systems is hard
 - ◆ Customers invent this themselves via scripting, automation, etc.
- ▶ Who is contributing to Manila?
 - ◆ NetApp, Mirantis, RedHat, EMC, HP, SUSE, more...
- ▶ Where can I get Manila?
 - ◆ <https://github.com/openstack/manila>
 - ◆ <https://wiki.openstack.org/wiki/Manila>
 - ◆ #openstack-manila on IRC (Freenode)

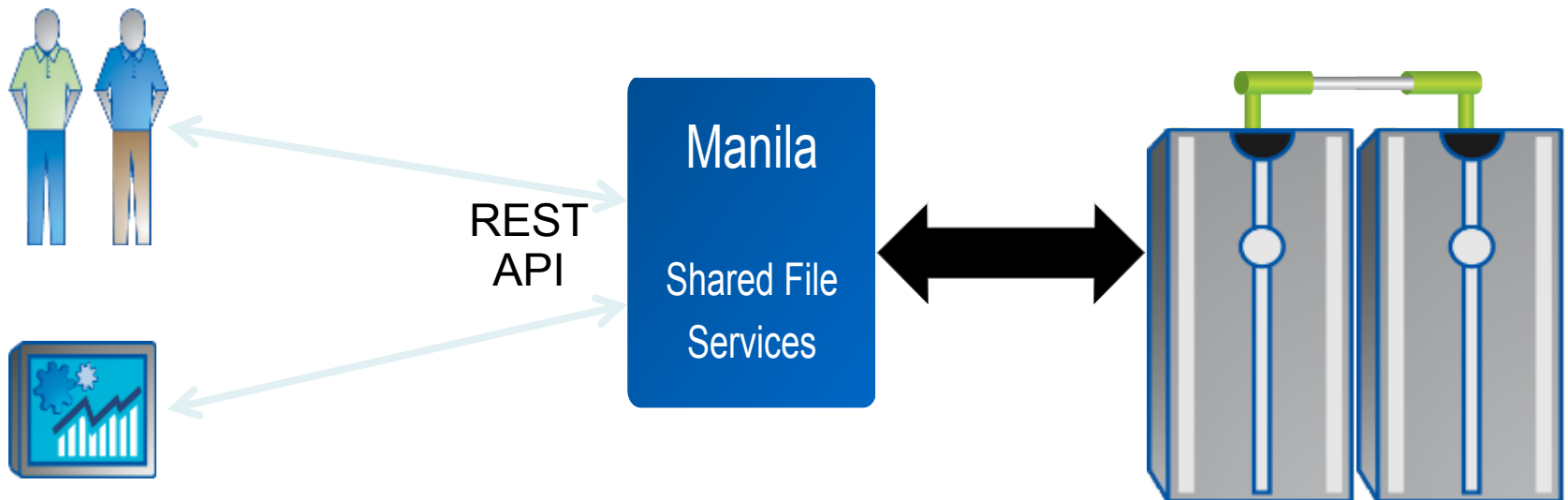


Manila: Project Overview



Manila: Overview of Key Concepts

- Replace homegrown or legacy systems
- Improve IT responsiveness by providing self-service share management
- Integrate Manila with existing automation frameworks through REST API or CLI



Manila: Overview of Key Concepts



- **Share (an instance of a shared file system)**
 - ◆ User specifies size, access protocol, “share type”
 - ◆ Can be accessed concurrently by multiple instances



- **Share access rules (ACL)**
 - ◆ Defines which clients can access the share
 - ◆ Specified by IP in CIDR notation



- **Share network**
 - ◆ Defines the Neutron network & subnet through which instances access the share
 - ◆ A share can be associated with only one share network

Manila: Overview of Key Concepts



➤ Security service

- ◆ Finer-grained client access rules for Authn/z (e.g. LDAP, Active Directory, Kerberos)
- ◆ Security service can be associated with one or more share networks



➤ Snapshots

- ◆ Point-in-time copy of share contents
- ◆ New share can be created from a snapshot



➤ Backend

- ◆ Provider of shares; a share resides on a single backend



➤ Driver

- ◆ Vendor or technology-specific implementation of backend API

Manila: API Overview - Shares

Operation	CLI command	REST API
Create share	manila create	POST /shares
Delete share	manila delete <id>	DELETE /shares/{id}
List shares	manila list	GET /shares
Show share details	manila show <id>	GET /shares/{id}
Rename share	manila rename	PUT /shares/{id}
Edit share metadata	manila metadata	PUT /shares/{id}/metadata
Show share metadata	manila metadata-show	POST /shares/{id}/metadata

Manila: API Overview – Share Access & Networks

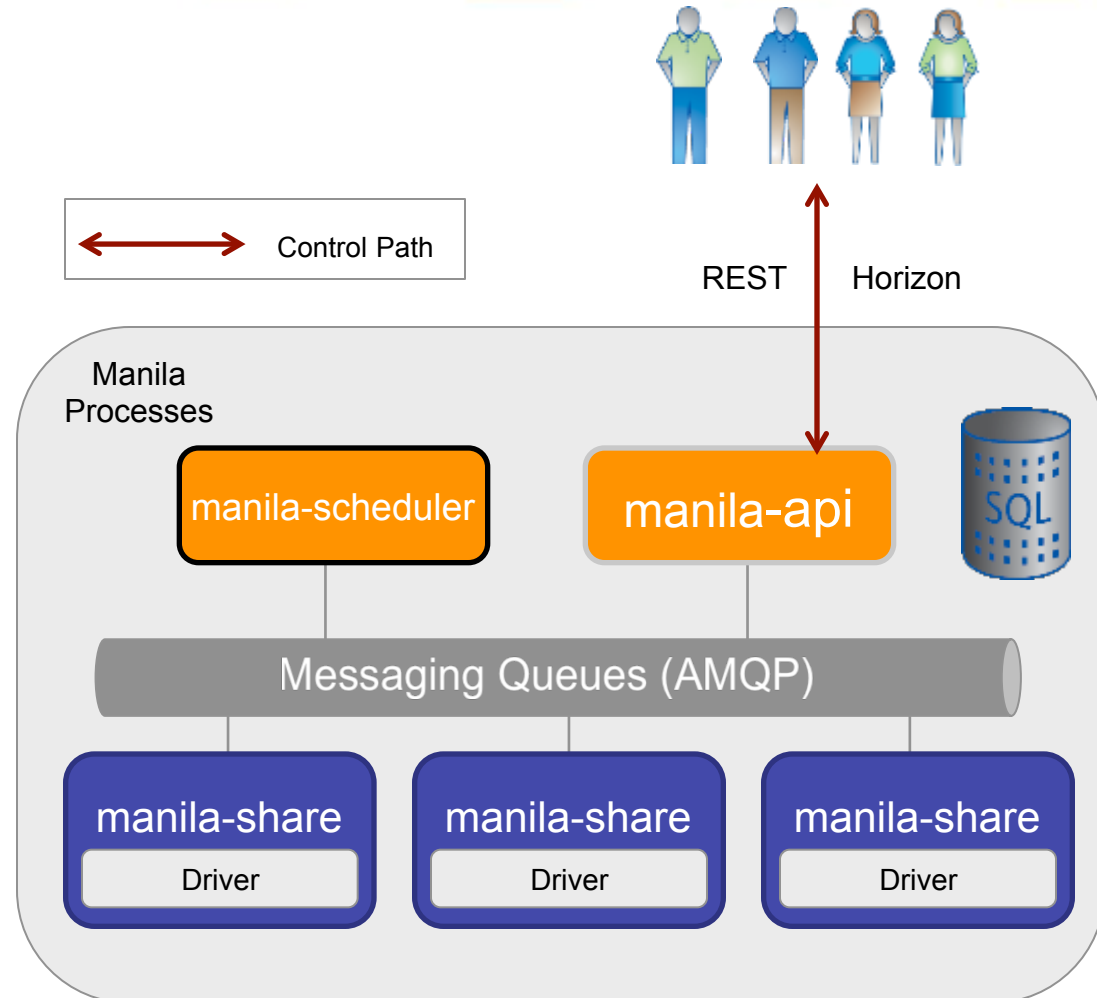


Operation	CLI command	REST API
Allow share access	manila access-allow	POST /shares/{id}/action
Deny share access	manila access-deny	POST /shares/{id}/action
List share access	manila access-list	POST /shares/{id}/action

Operation	CLI command	REST API
Create share network	manila share-network-create	POST /share-networks
Delete share network	manila share-network-delete	DELETE /share-networks/{id}
List share networks	manila share-network-list	GET /share-networks
Modify share network	manila share-network-update	PUT /share-networks/{id}

Manila: Overview of Architecture

- Manila is not in the data path!
- `manila-api`
 - ◆ Exposes REST API through WSGI application
- `manila-scheduler`
 - ◆ Makes provisioning decisions for share requests
- `manila-share`
 - ◆ Manager process + one process per backend
 - ◆ Responsible for communicating with storage subsystems



➤ Reference Implementation (Generic Driver)

- ◆ Nova VM (from Glance image) hosts NFS, CIFS servers
- ◆ Cinder volume (of requested size) per share provides storage capacity

➤ NetApp cDOT

➤ Red Hat GlusterFS

➤ EMC VNX and Isilon

➤ IBM GPFS (Kilo)

➤ Oracle ZFS (Kilo)

➤ Huawei (Kilo)

➤ HP 3PAR

➤ HDFS

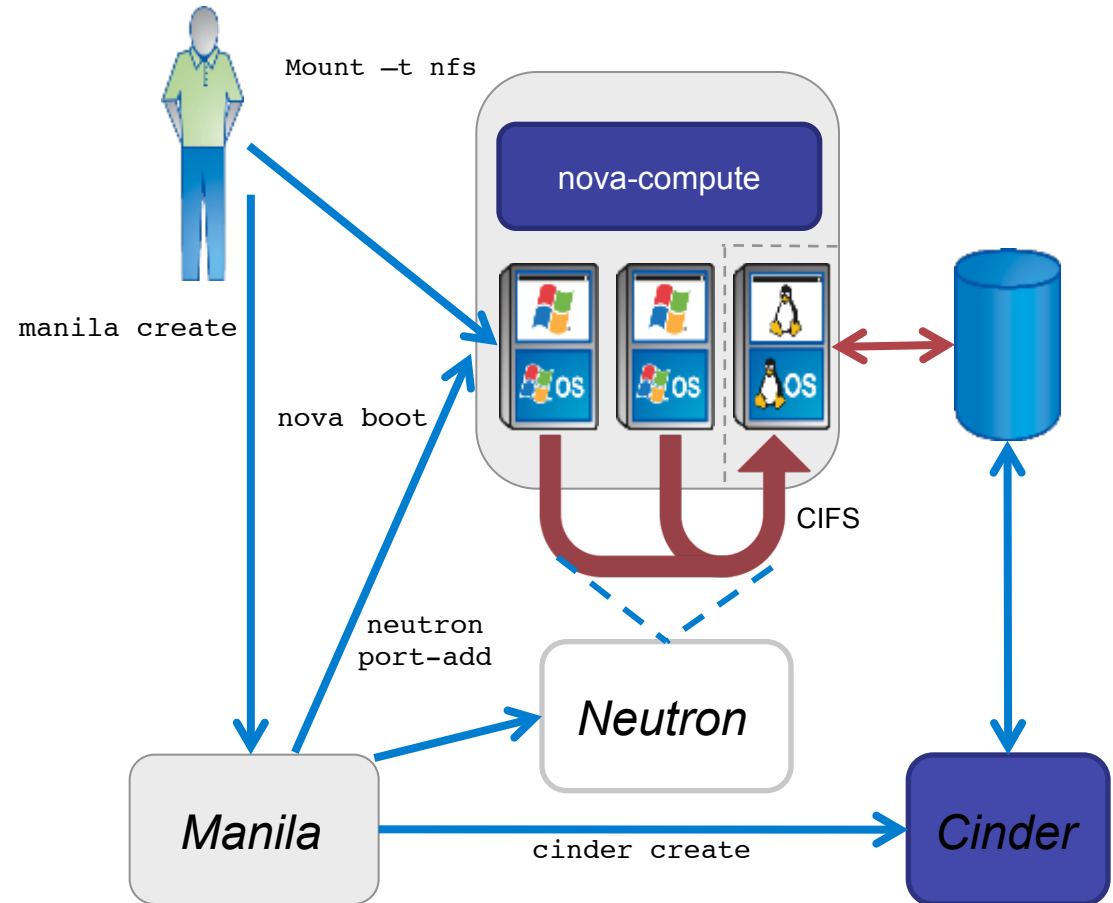
➤ Hitachi HNAS (Liberty)

➤ Quobyte (Liberty)

➤ Windows SMB (Liberty)

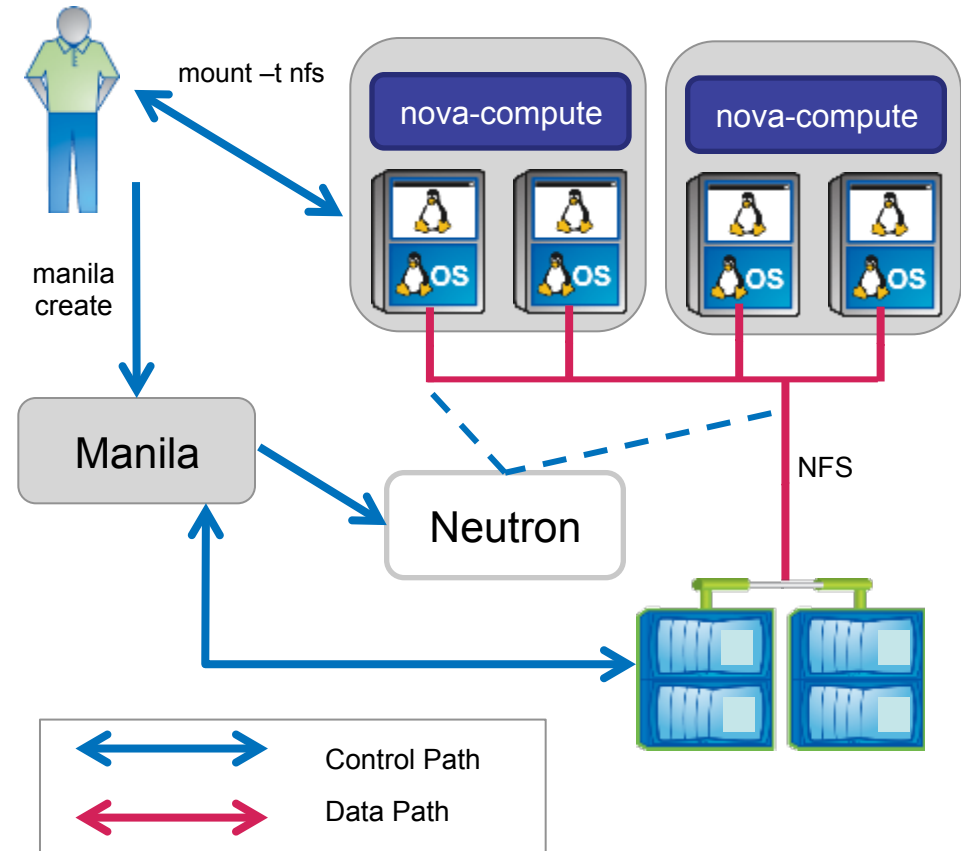
Manila: Generic Share Driver

- Creates a Nova instance (not owned by requesting tenant) to offer NFS/CIFS shares backed by Cinder volumes
- New instance is created for each “share network”
- Connected into existing Neutron network & subnet
- Instance flavor, source Glance image, & SSH keys are configurable in manila.conf
- Manila creates shares in instance using Linux commands over SSH



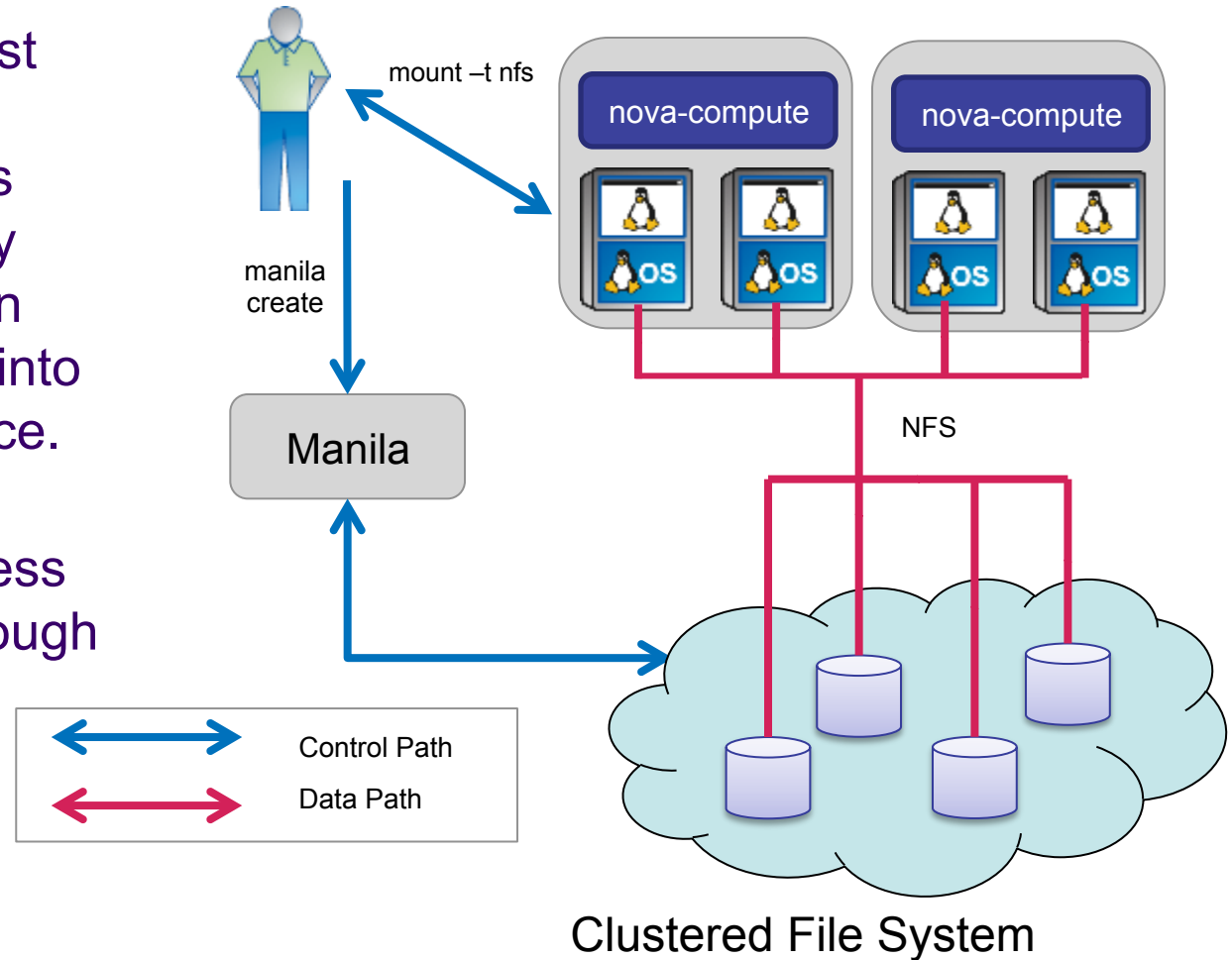
Manila: Commercial Drivers

- Multiple drivers exist for traditional “storage controllers”
- Manila is the control path
- Data path is direct from VMs/hypervisors to storage



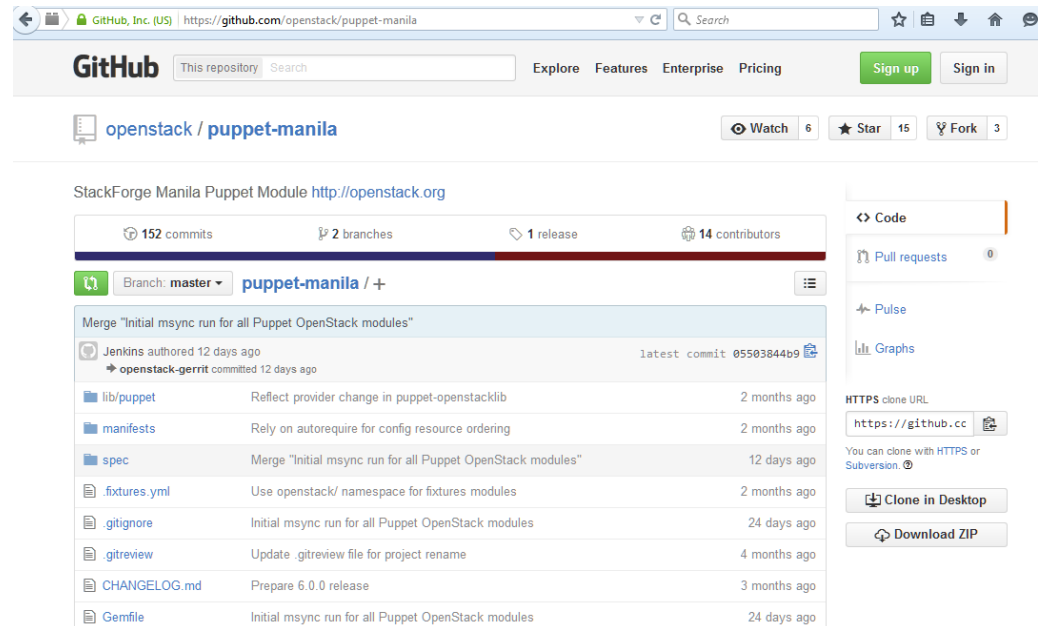
Manila: Open Source / Software Drivers

- Multiple drivers exist for pure software storage systems as well, which typically combine storage on multiple machines into a unified namespace.
- These drivers can provide native access or NFS access through a translation layer.



Manila: Automation with Puppet

- Puppet is a configuration management and deployment automation system.
- The Manila Puppet module enables Puppet to manage the entirety of Manila.
- Combination of Puppet manifests and ruby code to delivery configuration and extra functionality through types and providers.



The screenshot shows the GitHub repository page for `openstack/puppet-manila`. The repository is located at `https://github.com/openstack/puppet-manila`. It has 152 commits, 2 branches, 1 release, and 14 contributors. The current branch is `master`. The repository is part of the StackForge Manila Puppet Module, available at `http://openstack.org`.

The file list includes:

File	Description	Last Commit
<code>lib/puppet</code>	Reflect provider change in puppet-openstacklib	2 months ago
<code>manifests</code>	Rely on autorequire for config resource ordering	2 months ago
<code>spec</code>	Merge "Initial msync run for all Puppet OpenStack modules"	12 days ago
<code>fixtures.yml</code>	Use openstack/ namespace for fixtures modules	2 months ago
<code>.gitignore</code>	Initial msync run for all Puppet OpenStack modules	24 days ago
<code>.gitreview</code>	Update .gitreview file for project rename	4 months ago
<code>CHANGELOG.md</code>	Prepare 6.0.0 release	3 months ago
<code>Gemfile</code>	Initial msync run for all Puppet OpenStack modules	24 days ago

Manila: Status

- **Juno and Kilo deployed in production**
- **Liberty release imminent (Oct 15)**
- 3rd party vendor CI implemented
- Community diverse and growing
- 14 drivers from 10 storage vendors

JUNO
THE TENTH OPENSTACK RELEASE

KILO
OPENSTACK GOES TO ELEVEN

LIBERTY
THE TWELFTH RELEASE OF OPENSTACK 12

Manila: What was new in Kilo

- Greatly improved test coverage and stability
- Integration with devstack, tempest, horizon
- Manage and Un-manage of Shares
- Share types
- Storage pools
- Major features...

Manila: Kilo Driver Changes

- Kilo made support of “share servers” optional
 - ◆ Share servers are storage VMs that provide secure isolation between tenants
- Drivers can implement support for share servers and/or no share servers
 - ◆ Allows Manila to support more storage controllers
- Admin decides which type(s) to use

Manila: Kilo Network Plugins

- Network plugins give administrator more control over networking
- Segmented (VLAN, VXLAN) vs Flat
 - ◆ Goal is to simplify deployment
- Neutron, nova-network, Standalone
 - ◆ Support wider variety of use cases

Manila: What is new in Liberty

- Complete documentation
- Open sourcing of generic server image
- Oversubscription
- Expand/shrink shares
- Consistency Groups
- Micro-versions
- Experimental features
- Major features...

Manila: Liberty Mount Automation

- Major difference between block storage and shared file systems is the attachment
- Mount automation provides a mechanism for VMs to mount new shares without user intervention
- Liberty adds support for some of the most common use cases
- More work left to be done...

Manila: Liberty Share Migration

- Share migration allows moving a share from one storage controller to another
- Administrator-controlled
- Support for optimized migration and universal fallback
- Admin use cases
 - ◆ Evacuation for maintenance
 - ◆ Load balancing
- Will allow future features
 - ◆ Share re-type
 - ◆ Change availability zone
 - ◆ Share security domain

Manila: Mitaka and beyond

- Migration 2.0
- More first party drivers
 - ◆ Open source drivers supported by the community
 - ◆ Basis for functional testing of Manila
- Better support for rolling upgrades and HA
 - ◆ Eliminate downtime in large-scale deployments of Manila
- Share replication
 - ◆ Copies of your shares in different failure domains

Manila: How to get Started

➤ Manila Wiki

- ◆ <https://wiki.openstack.org/wiki/Manila>

➤ Distributions

- ◆ RedHat RDO
 - › RPMs and Packstack support are available now
- ◆ OpenSUSE
 - › RPMs are available now; installer integration coming soon

➤ IRC on Freenode

- ◆ #openstack-manila
 - › any time
- ◆ #openstack-meeting-alt
 - › 15:00 UTC on Thursdays



After This Webcast

- This webcast will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand
 - ◆ <http://www.snia.org/forums/esf/knowledge/webcasts>
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog
 - ◆ <http://sniaesfblog.org/>
- Follow us on Twitter @ [SNIAESF](https://twitter.com/SNIAESF)

Conclusion

Thank You