OpenStack Manila
Shared File Services for the Cloud

January 29, 2015

SNIA™
Cloud Storage Initiative

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.

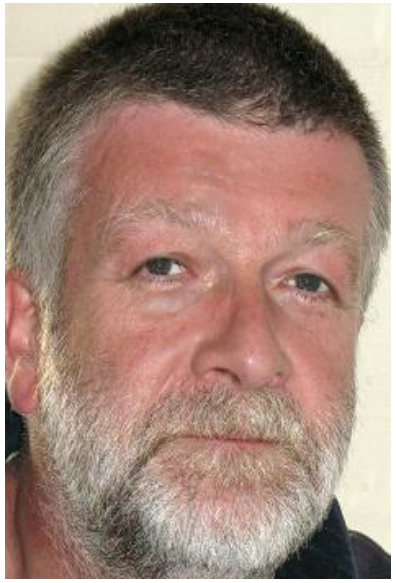
  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Presenters

SNIA™
Cloud Storage Initiative

Greg Loughmiller,
Technical Marketing Engineer,
NetApp
Cloud Solutions Group

Alex McDonald, SNIA – CSI
Cloud Storage Initiative Chair - NetApp



Agenda

SNIA™
Cloud Storage Initiative

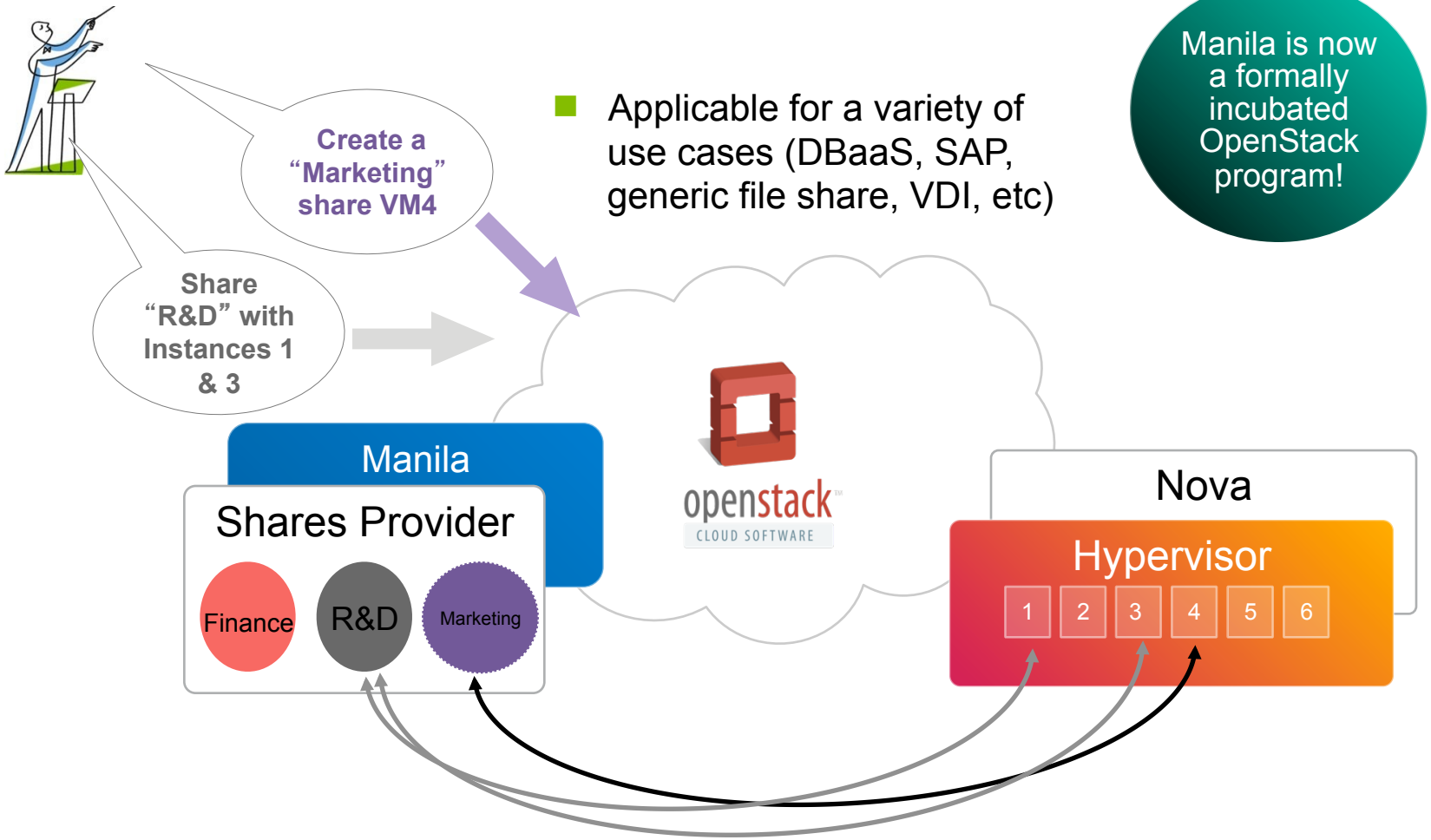
- Project Overview
- Key Concepts
- API Overview
- Architecture Overview
- Drivers and Integration
- Automation
- Roadmap and Futures

Shared File Services

of all storage sold is for file-based use cases

per IDC, 2012

Manila: Project Overview

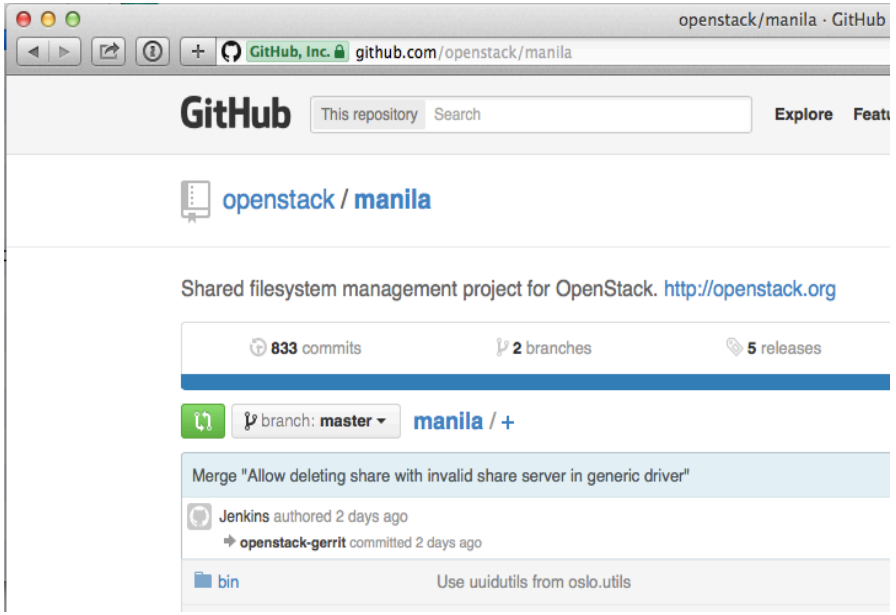


- Applicable for a variety of use cases (DBaaS, SAP, generic file share, VDI, etc)

Manila is now a formally incubated OpenStack program!

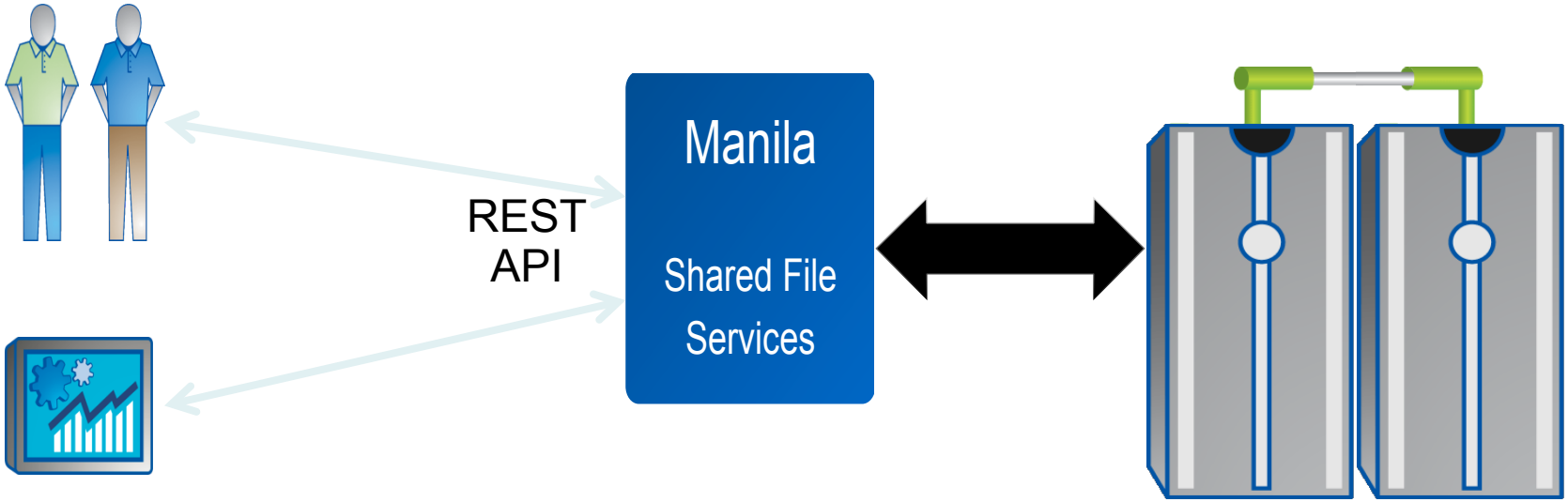
Manila: Project Overview

- What's the customer use case for Manila?
  - 65% of disk capacity slated for file storage (IDC 2012)
  - Self-service management & provisioning of shared file systems is hard
  - Customers invent this themselves via scripting, automation, etc.
- Who is contributing to Manila?
  - EMC, IBM, Mirantis, NetApp, Red Hat, SUSE
- Where can I get Manila?
  - https://github.com/openstack/manila
  - https://wiki.openstack.org/wiki/Manila
  - `http://netapp.github.io/openstack/2014/08/15/manila-devstack/`
  - `#openstack-manila` on IRC (Freenode)



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)
  - Share can be associated to multiple security services

- Snapshots
  - Read-only 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

SNIA Cloud Storage Initiative

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

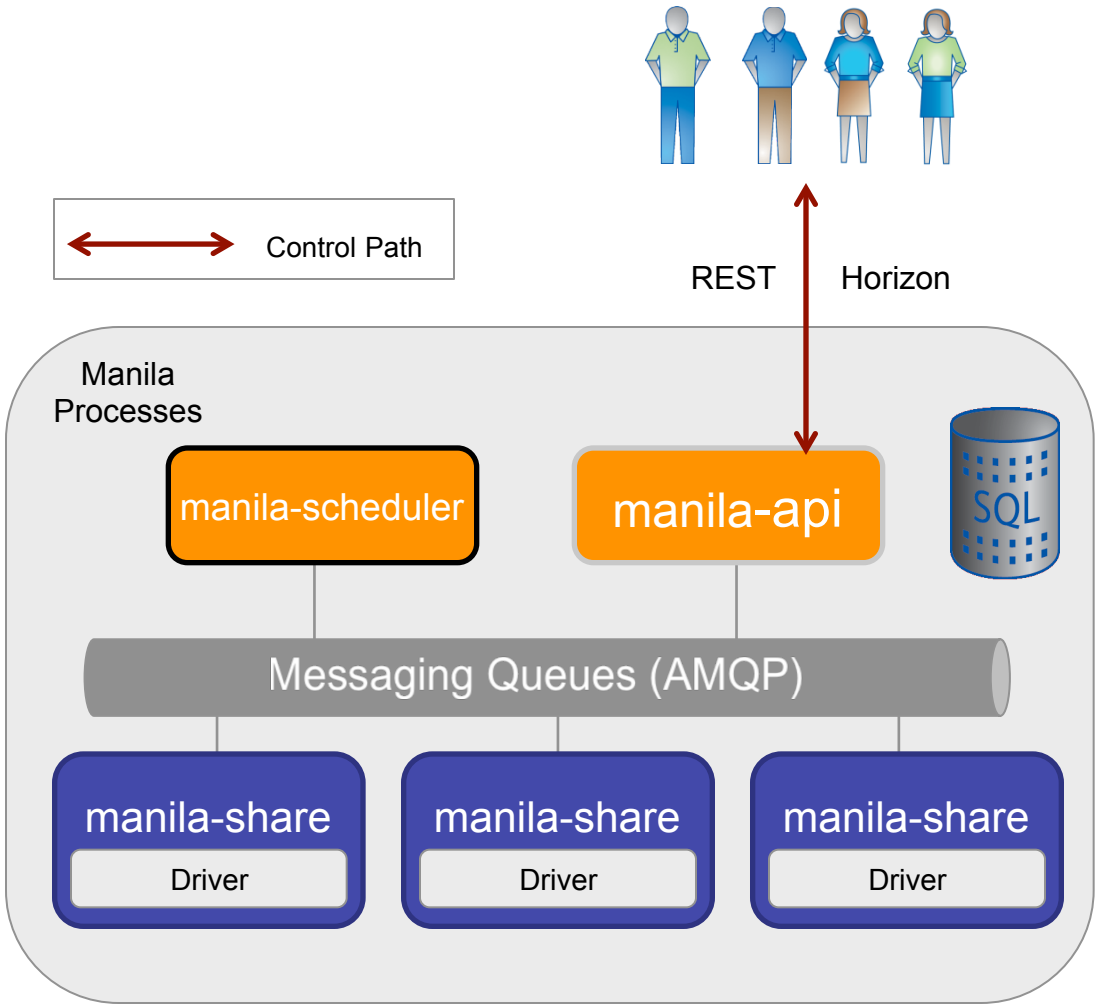
Manila: API Overview – Share Access & Networks

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
Activate share network	manila share-network-activate	POST /share-networks/{id}/action

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

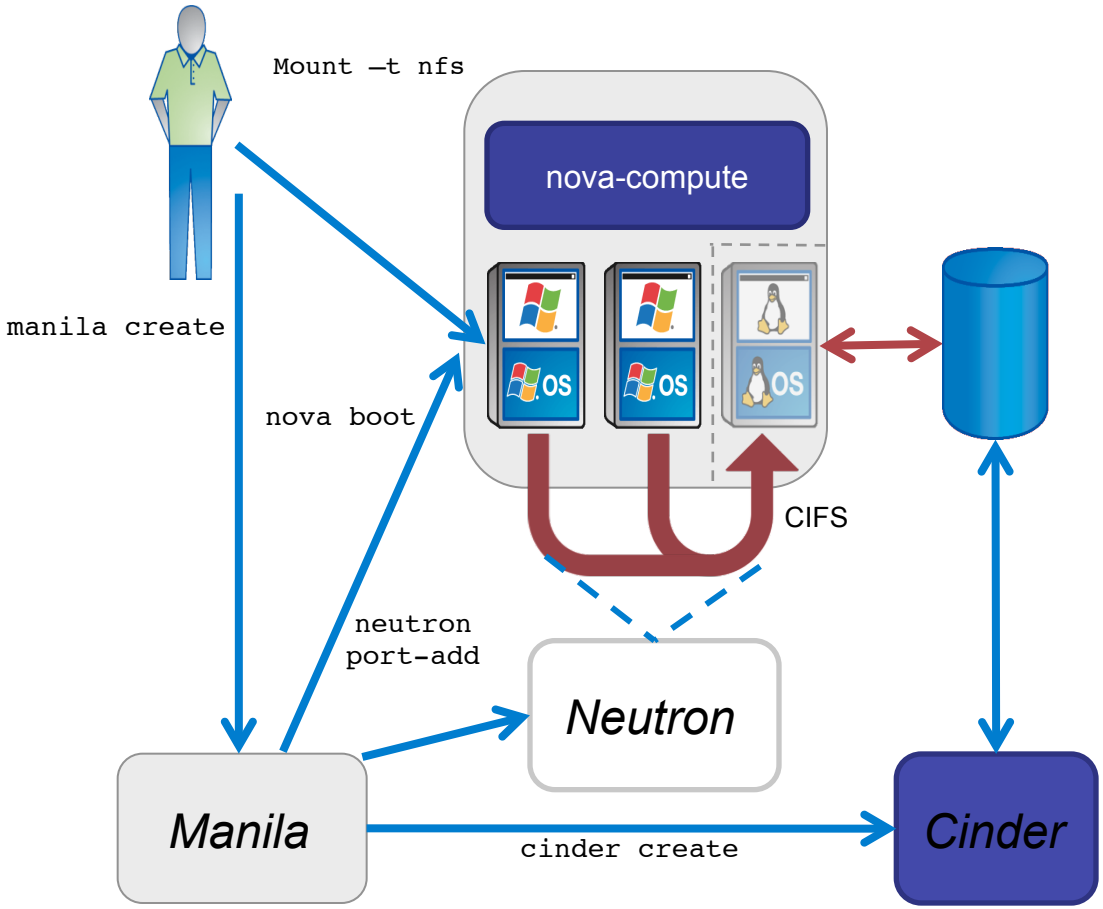


Manila: Vendor Drivers

- Reference Implementation
  - Nova VM (from Glance image) hosts NFS, CIFS servers
  - Cinder volume (of requested size) per share provides storage capacity
- NetApp clustered Data ONTAP
- Red Hat GlusterFS
- EMC VNX and Isilon
- IBM GPFS (Kilo)
- NFS-Ganesha(Kilo)
- *Oracle ZFS(Kilo)*
- Huawei (Kilo)
- *More to come…*

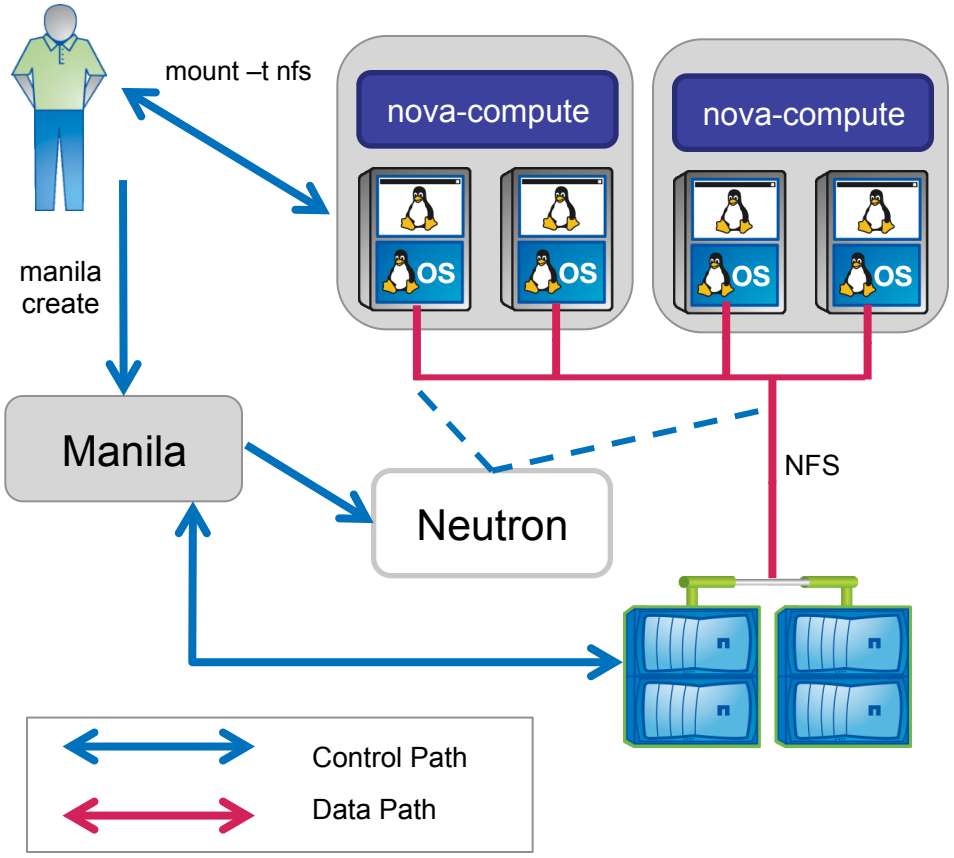
Manila: Generic Share Driver

- 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 keypair are configurable in manila.conf
- Manila creates shares in instance using Linux commands over SSH



Manila: NetApp clustered Data ONTAP

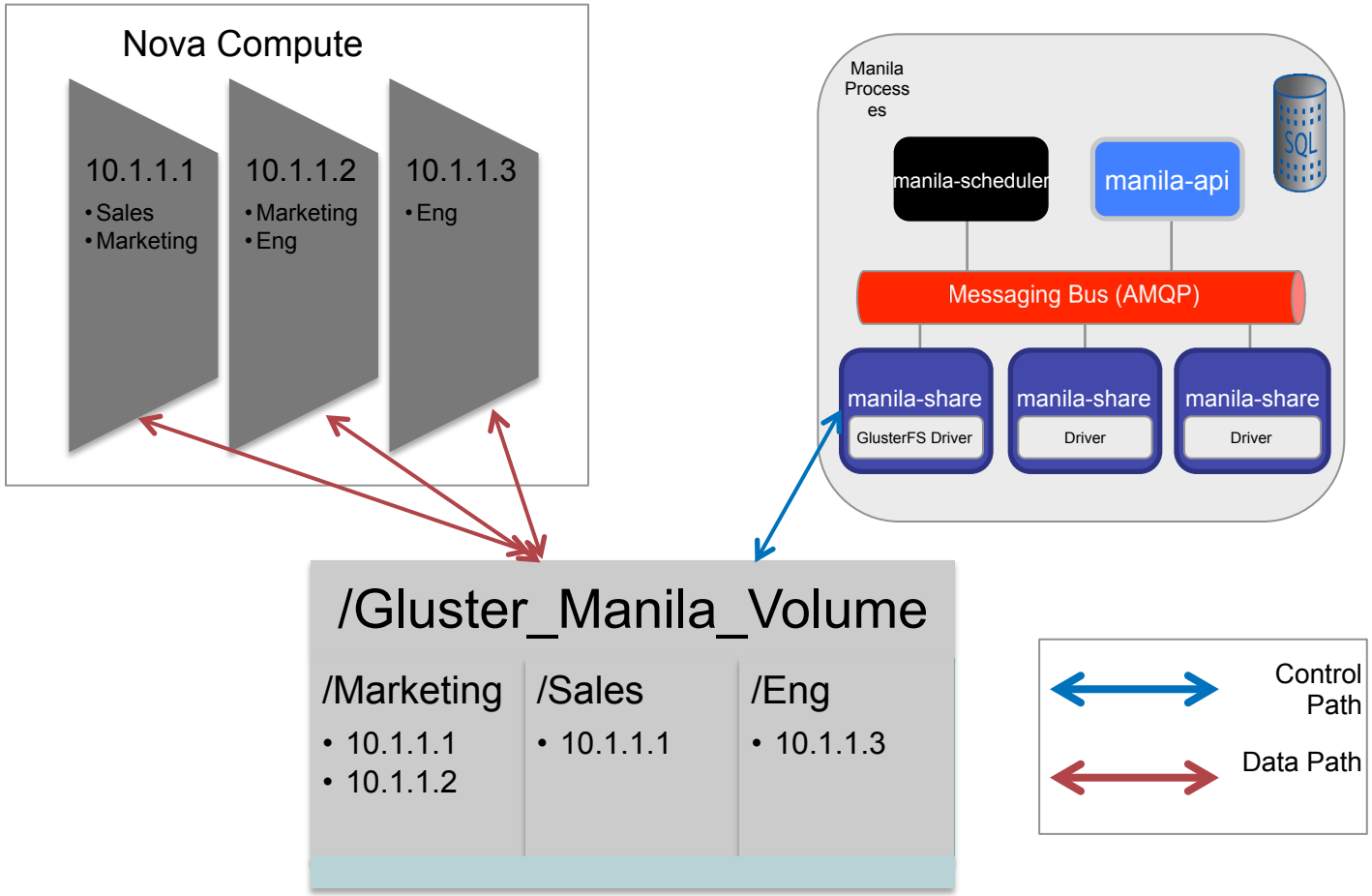
- Driver points at an existing clustered Data ONTAP deployment
- A new storage virtual machine (SVM) is created for each share network
- Create a new Data LIF with the VLAN tag set to same VLAN tag of the Neutron network
- Access permissions are set at the controller for guests



Manila: Red Hat GlusterFS

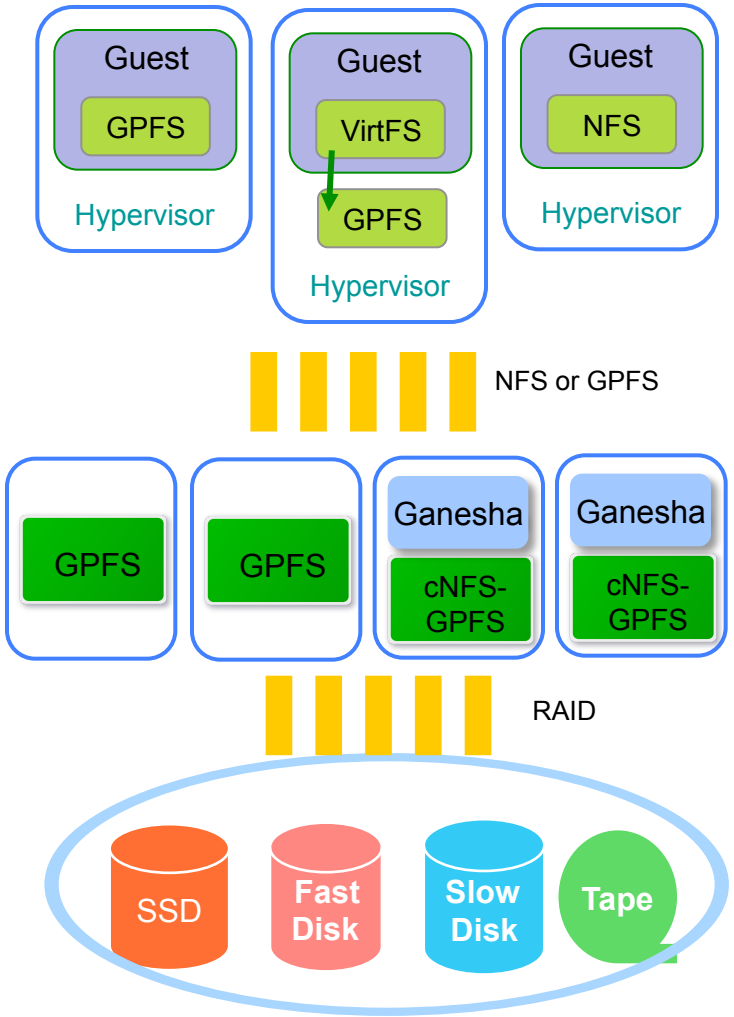
manila create
manila access-allow

- Use Gluster to provide a Distributed Scale Out File System backend for Manila File Shares
- Current: Single Tenant GlusterFS driver using Gluster-NFS
- Future: Be able to use NFS Ganesha(Kilo)



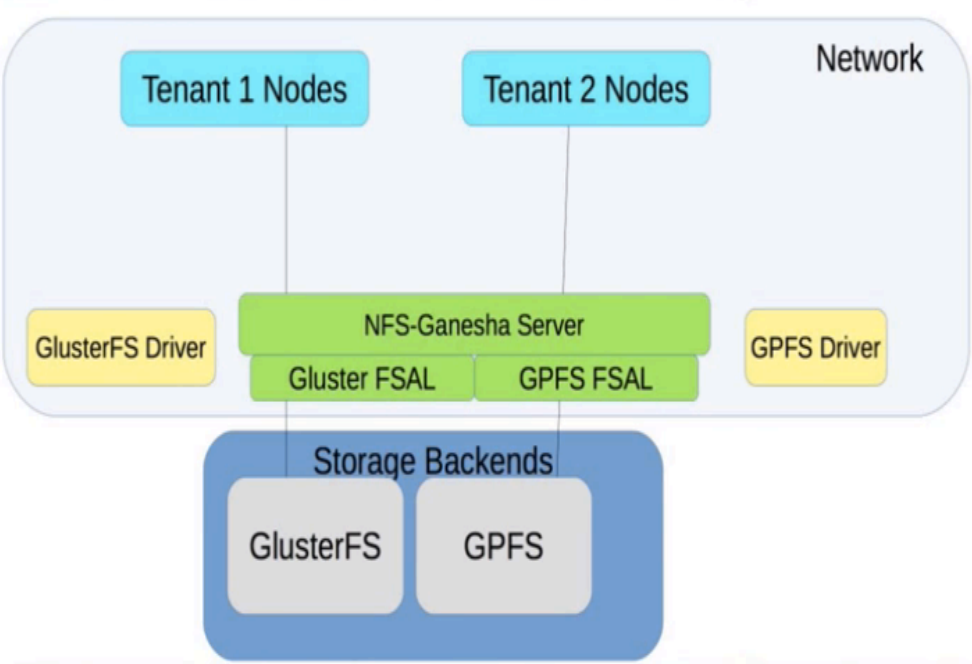
Manila: IBM GPFS

- Goal is for GPFS to provide a single data plane for all data center applications.
  - Manila extends the data plane into the VM
- Support both cNFS and GPFS
  - pNFS in roadmap
- Leverage Standard GPFS Enterprise features
  - Scale Out, High Performance, Highly Available, Encryption, Backup, DR, Declustered RAID
  - Tiered Storage Pools



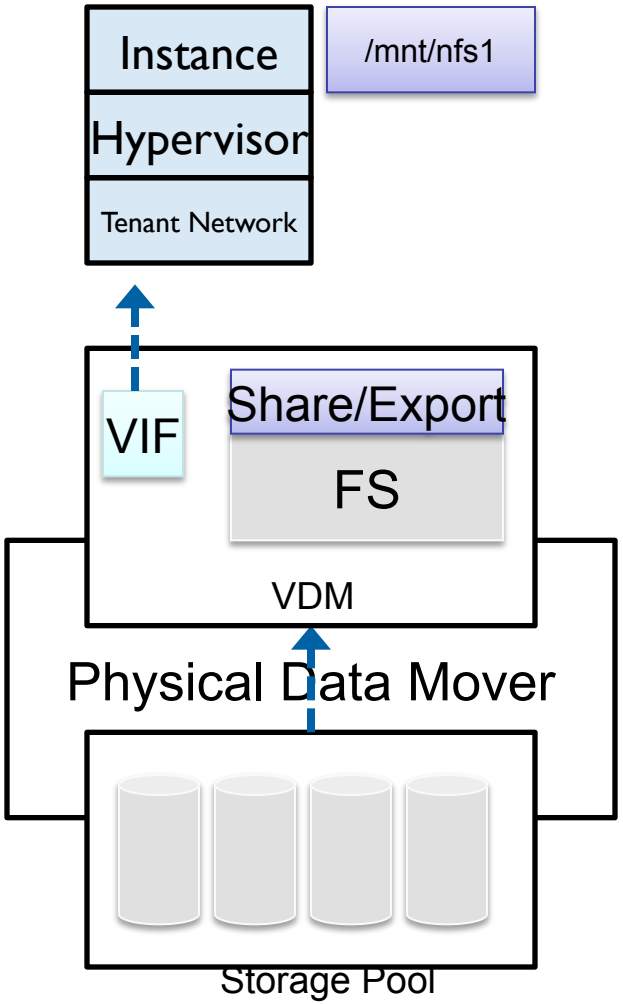
Manila: NFS-Ganesha Driver

- The NFS-Ganesha is a User Space NFS Server
- It is an Open Source project with active community of both Company and Independent sponsored developers
- Provides a modular architecture providing File System Abstraction Layer(FSAL) for Storage Backends
- Ganesha driver will be a gateway mediated model between Manila and Storage Pools
- Current work in progress for Manila
  - GlusterFS
  - GPFS



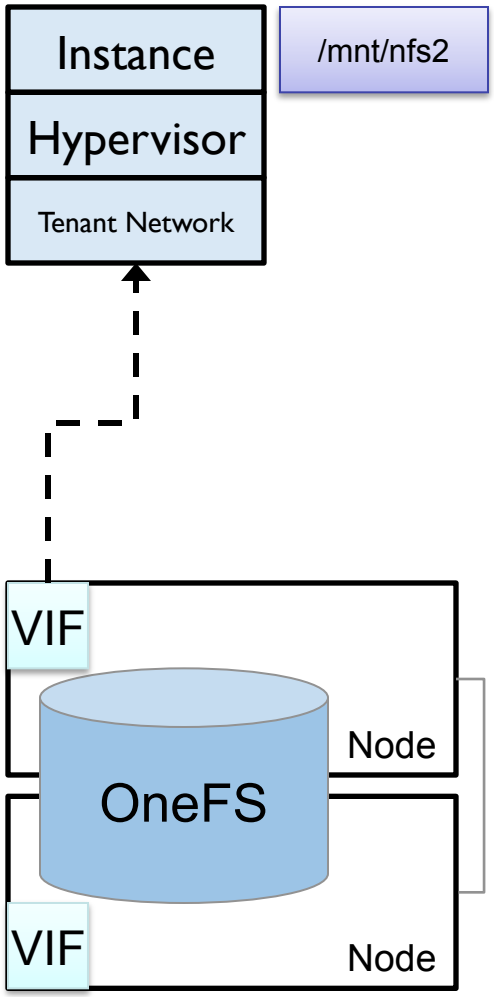
Manila: EMC VNX

- Today:
  - Common driver with Plugin capability
  - Prototype supports majority of Manila API
  - Currently requires that interface already exist on tenant network
- WIP Items:
  - Implement multi-tenancy by automating management of Virtual Data Movers (VDM) and VIF (via network setup/teardown)
  - Add logic for selecting other physical data movers



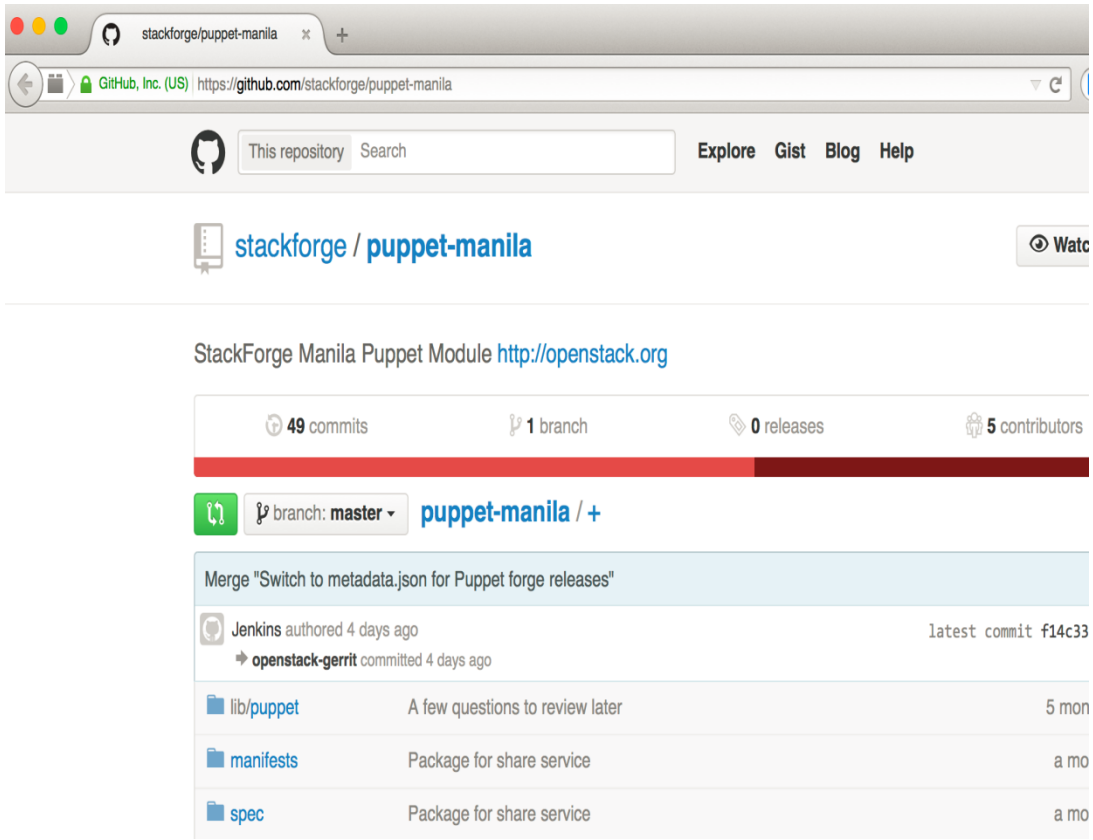
Manila: EMC Isilon

- Today:
  - Creates export/share and allocates quota
  - Supports majority of Manila API
  - Leverages Host ACL for both SMB and NFS
- WIP Items:
  - SmartConnect and Access Zones (for multi-tenancy)
  - Define “update_share_status”



Manila: Automation

- The manila module is a thorough attempt to make Puppet capable of managing the entirety of manila.
- Combination of Puppet manifests and ruby code to delivery configuration and extra functionality through types and providers.



Manila: Status

- **Officially an incubated OpenStack program (August 26th, 2014)!**
- Work towards Project Graduation
- Multiple POC's in progress
- Additional contributors and drivers being developed
- Active Community contributors

Manila: What is new in Juno

- Manila repos are now under “OpenStack” Github organization
- puppet-manila now on stackforge
- Increased developer & admin documentation
- Removal of obsolete/unmaintained drivers
- Added tempest integration and gate tests

Manila: Kilo and beyond

SNIA™ Cloud Storage Initiative

- Automate mounting shares within instances
- Share Type Support
- Replication
- Manage and Un-manage of Shares
- Backup
- Trove Enablement for Manila Awareness
- Additional Vendor Drivers and Backends
- Driver Extensions

Manila: How to get Started

- Manila Wiki
  - https://wiki.openstack.org/wiki/Manila
- Distributions
  - RDO
    - RPMs are available now; Packstack integration coming soon
  - OpenSUSE
    - RPMs are available now; installer integration coming soon
- IRC on Freenode
  - `#openstack-manila`
    - any time
  - `#openstack-meeting-alt`
    - 15:00 UTC on Thursdays

Other SNIA-CSI Webcasts

SNIA Cloud Storage Initiative

- Check out on-demand Webcasts from this Cloud Developers' Tutorial Series:
- Introducing CDMI 1.1
- OpenStack Cloud Storage
  - http://www.snia.org/forum/csi/knowledge/webcasts
- Upcoming CSI Webcasts:
- LTFS Bulk Transfer Standard – Feb. 10th https://www.brighttalk.com/webcast/663/138935
- Hybrid Clouds: Bridging Private & Public Cloud Infrastructures – March 18 https://www.brighttalk.com/webcast/663/143055

After This Webcast

SNIA™ Cloud Storage Initiative

- This webcast and a copy of the slides will be posted to the SNIA Cloud Storage Initiative (CSI) website and available on-demand
  - http://www.snia.org/forum/csi/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-CSI blog
  - http://www.sniacloud.com/
- Follow us on Twitter @SNIACloud

Conclusion

SNIA™
Cloud Storage Initiative

Thank You