SDC

STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2014

OpenStack Manila – File Storage

Bob Callaway, PhD

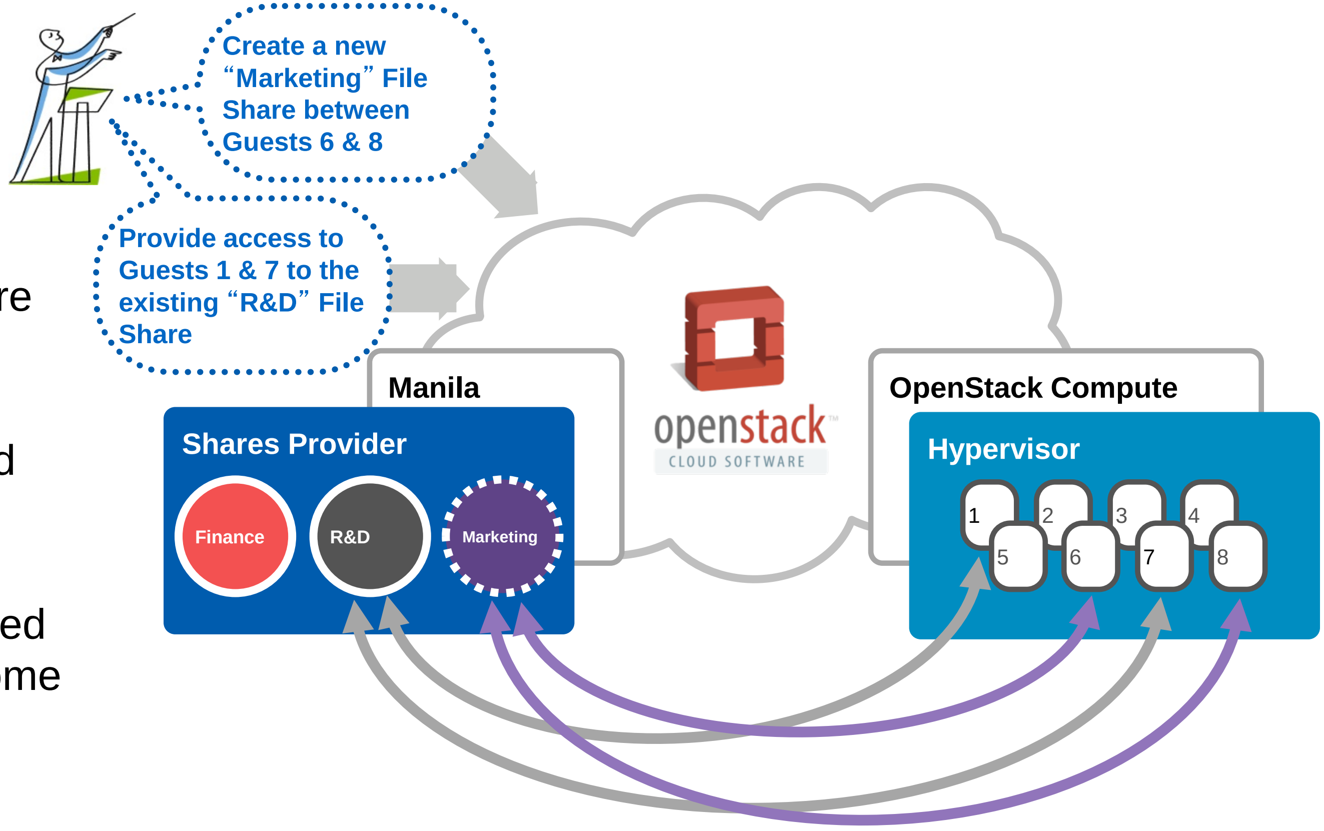
Cloud Solutions Group, NetApp

Agenda

- ❒ Project Overview
- ❒ API Overview
- ❒ Architecture Discussion
- ❒ Driver Details
- ❒ Project Status & Upcoming Features
- ❒ Q & A

Manila: Project Overview

- What is Manila?
  - Multitenant, secure file share as a service
  - “Cinder for shared file systems”
  - NFS & CIFS protocols supported today; more to come

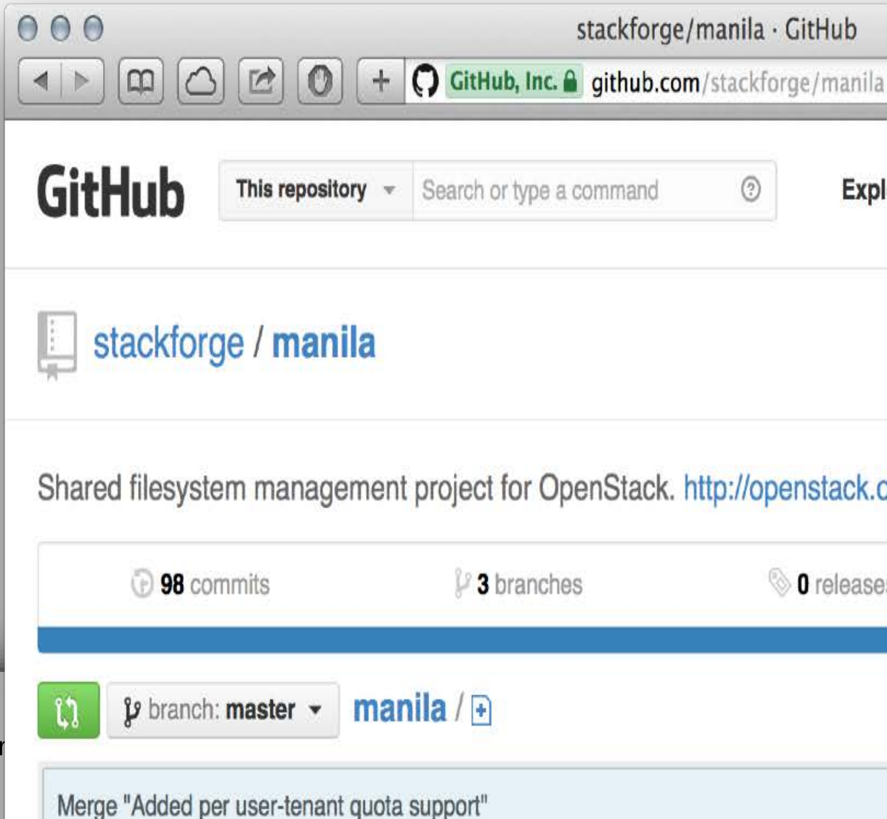


Create a new “Marketing” File Share between Guests 6 & 8

Provide access to Guests 1 & 7 to the existing “R&D” File Share

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?
  - NetApp, Mirantis, Red Hat, EMC, IBM
- Where can I get Manila (and when is it ready)?
  - Ready to download today!
  - https://github.com/stackforge/manila
  - https://wiki.openstack.org/wiki/Manila
  - #openstack-manila on IRC (Freenode)



Shared filesystem management project for OpenStack. http://openstack.c

Merge "Added per user-tenant quota support"

Manila: Overview of Key Concepts

- *Share (an instance of a shared filesystem)*
  - 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 a single 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

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

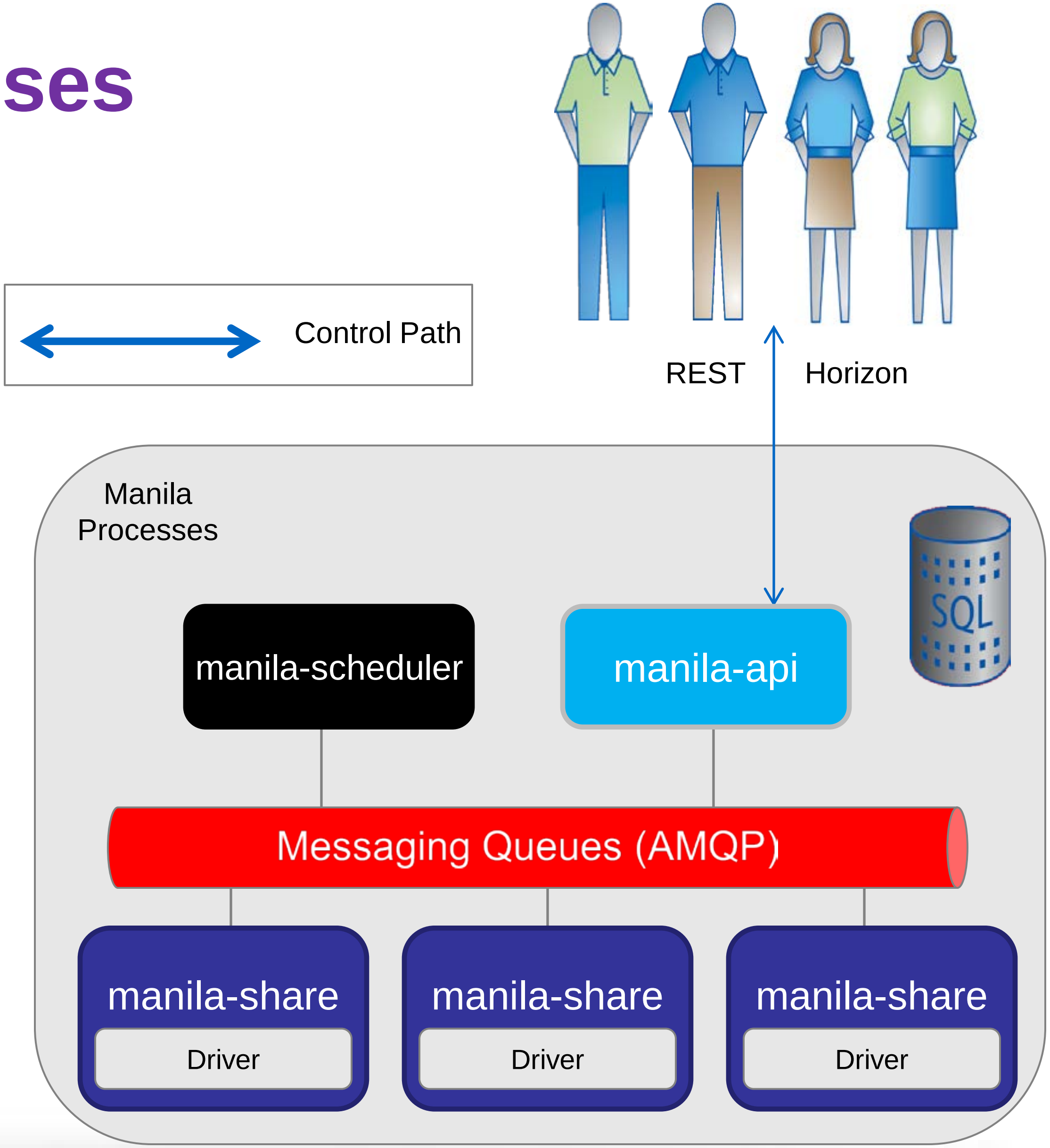
Manila: API Overview – Share Access & Networks

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: Processes

- *Not in the Data Path!*
- manila-api
  - Exposes REST APIs 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: Multitenancy Approaches

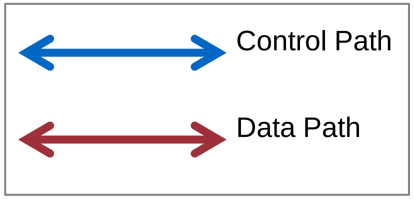
- Manila supports a variety of multitenancy approaches

Multitenancy approach	Storage Network Segmentation Support	Manila Enablement
Flat network	None	Direct
Flat network	VLAN, GRE, VXLAN, etc	Single VLAN for all tenants
Network segmentation	None	Manila creates gateway
Network segmentation	VLAN, GRE, VXLAN, etc	Manila configures storage to join tenant network

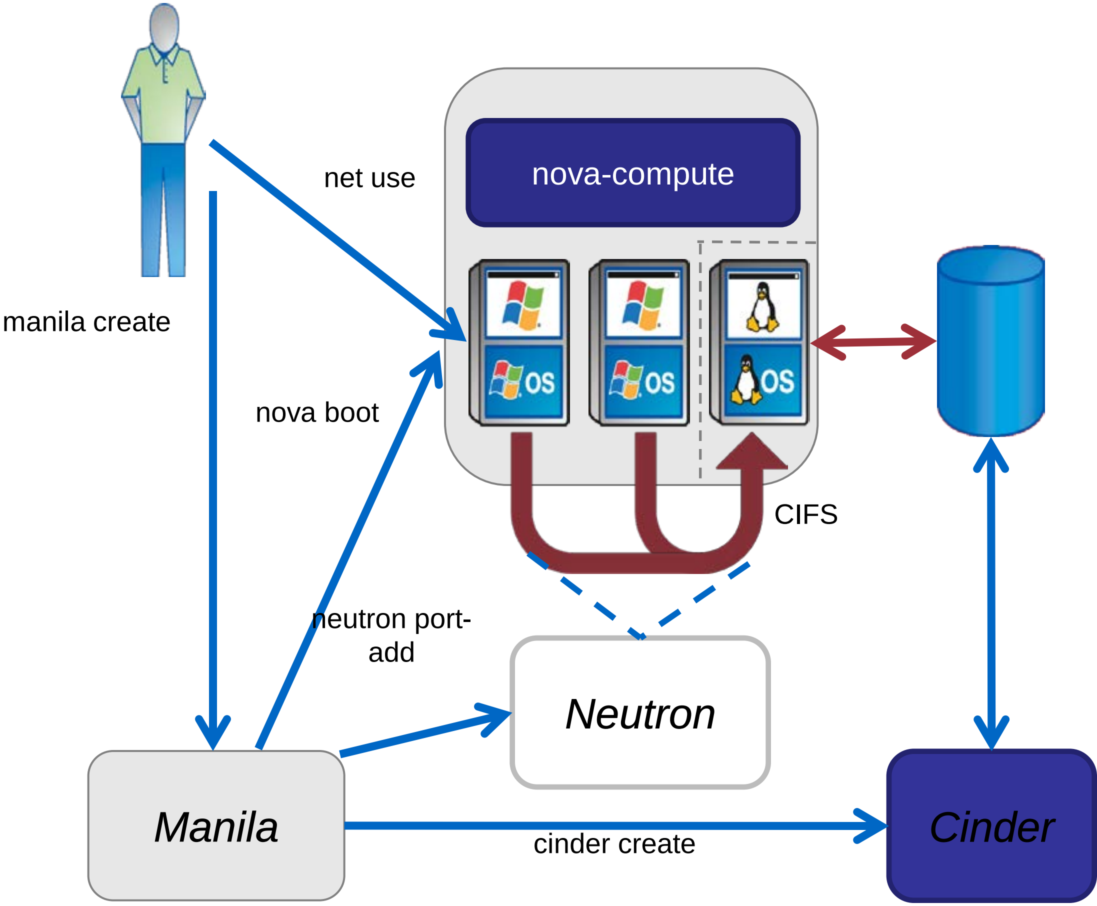
Manila: Mounting from the Guest

- Shares must be manually mounted within the guest today
- Community ideas for mount automation under consideration
  - *Not an exclusive list*
  - cloud-init
  - Userspace daemon
    - Notification pushed from Manila to daemon
  - Manila remotely issues mount command over SSH

Manila: “Generic” Share Driver



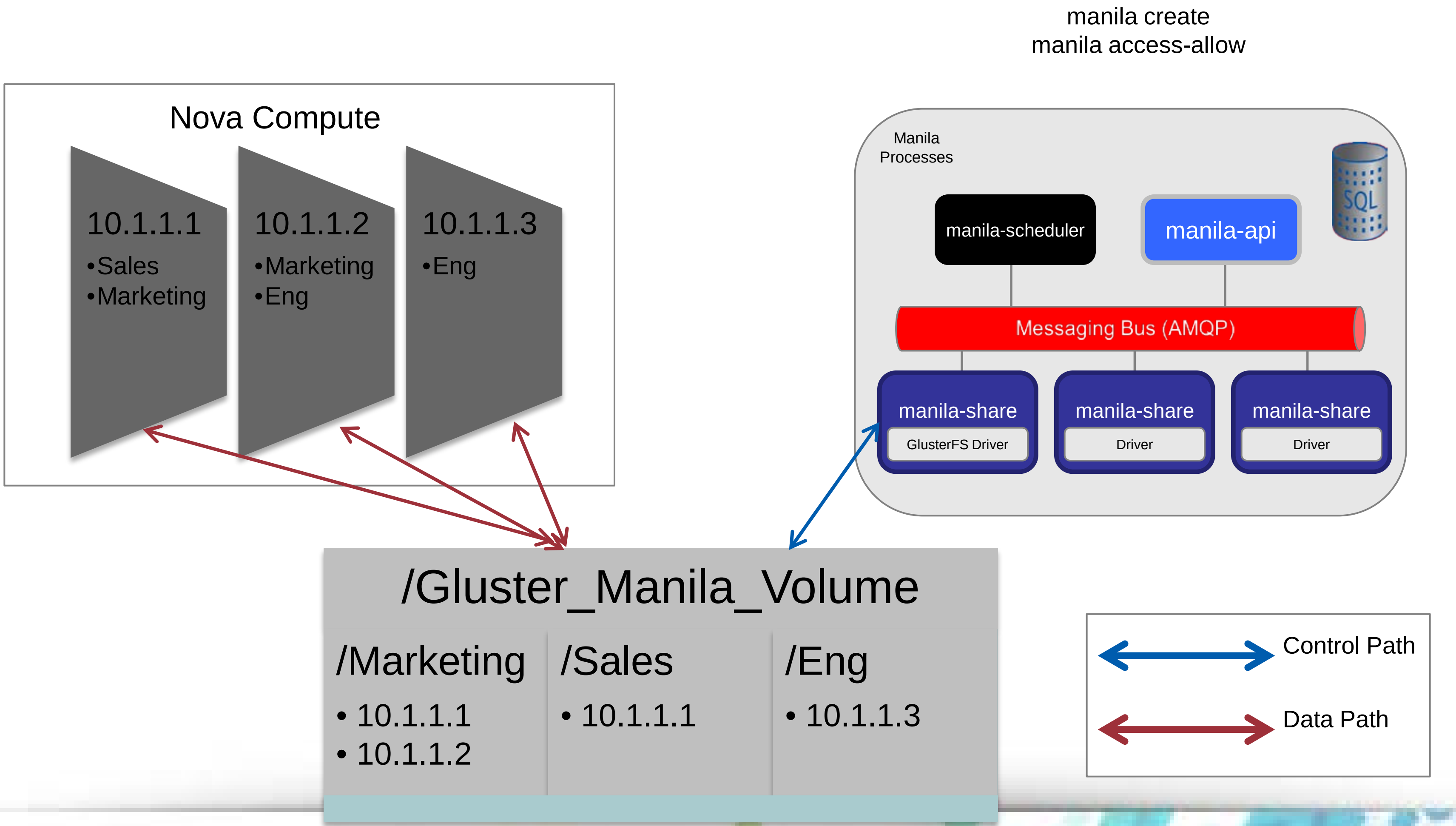
- Creates a Nova instance (not owned by requesting tenant) to offer NFS/CIFS shares backed by Cinder volumes
  - New instance is created for each “share network”
  - Connected into existing Neutron network & subnet
  - Instance flavor, source Glance image, & SSH keypair are configurable in manila.conf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



Red Hat and Manila: GlusterFS Driver

- Use Gluster to provide a Distributed Scale Out File System backend for Manila File Shares
- Current: Single Tenant GlusterFS driver using Gluster-NFS
- Future: Multi-Tenant driver that uses NFS Ganesha

GlusterFS Driver for Manila



manila create
manila access-allow

Nova Compute

10.1.1.1
- Sales
- Marketing

10.1.1.2
- Marketing
- Eng

10.1.1.3
- Eng

Manila Processes

manila-scheduler

manila-api

SQL

Messaging Bus (AMQP)

manila-share
GlusterFS Driver

manila-share
Driver

manila-share
Driver

/Gluster_Manila_Volume

/Marketing
- 10.1.1.1
- 10.1.1.2

/Sales
- 10.1.1.1

/Eng
- 10.1.1.3

Control Path

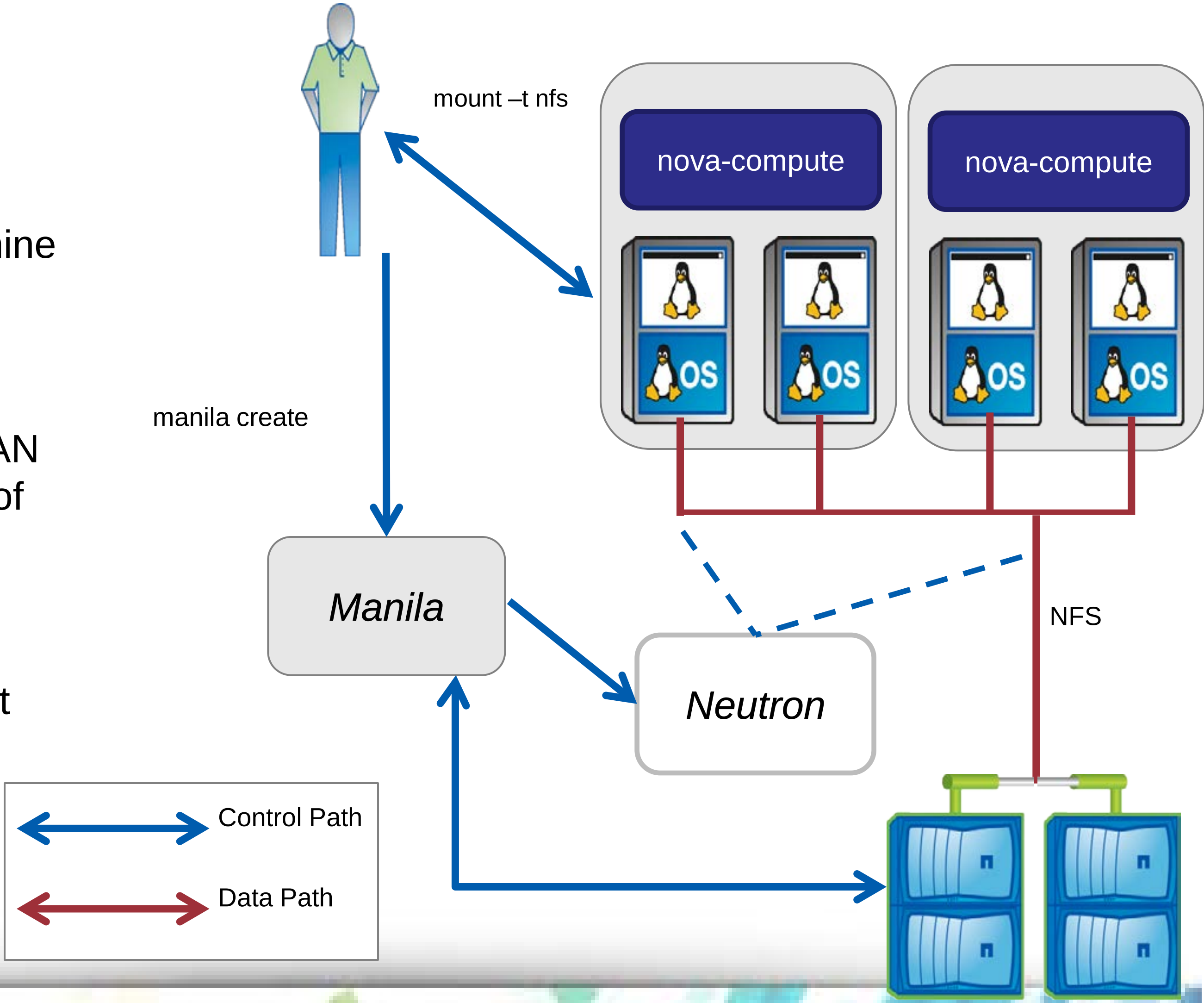
Data Path

Futures: Manila GlusterFS Driver

- Multi-Tenant GlusterFS Driver
  - Address Authentication and Authorization
  - Address Networking
- Migrate to NFS Ganesha
  - Userland NFS V4 server

Manila: NetApp CDOT Share Driver

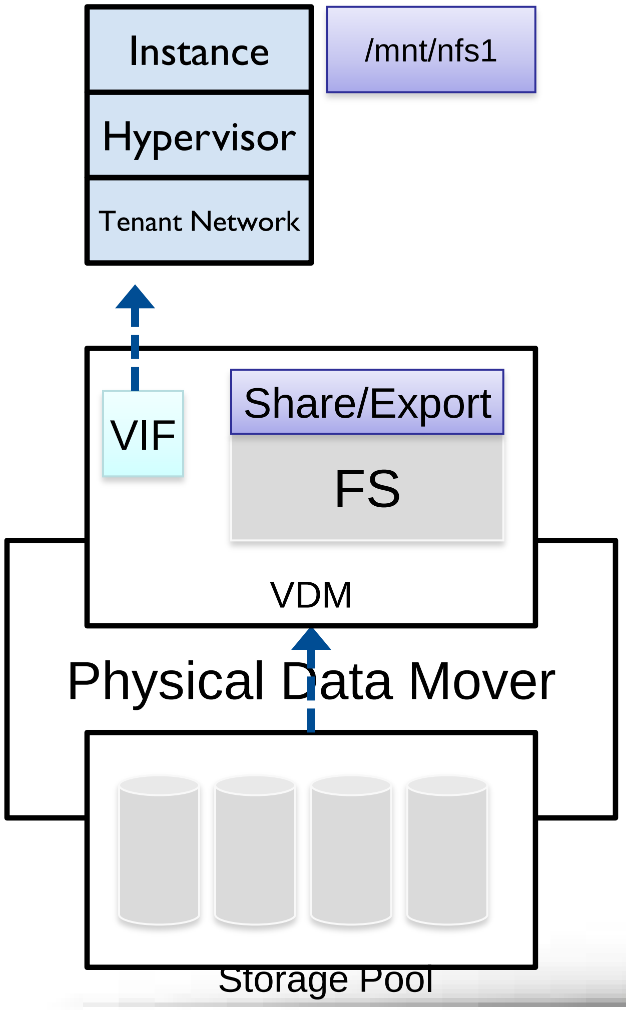
- Driver points at an existing Clustered Data ONTAP deployment
- A new storage virtual machine (SVM) is created for each share network
- Create a new Data Logical Interface (LIF) with the VLAN tag set to same VLAN tag of the Neutron network associated with the share network
- Access permissions are set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



EMC: Manila Integration Overview

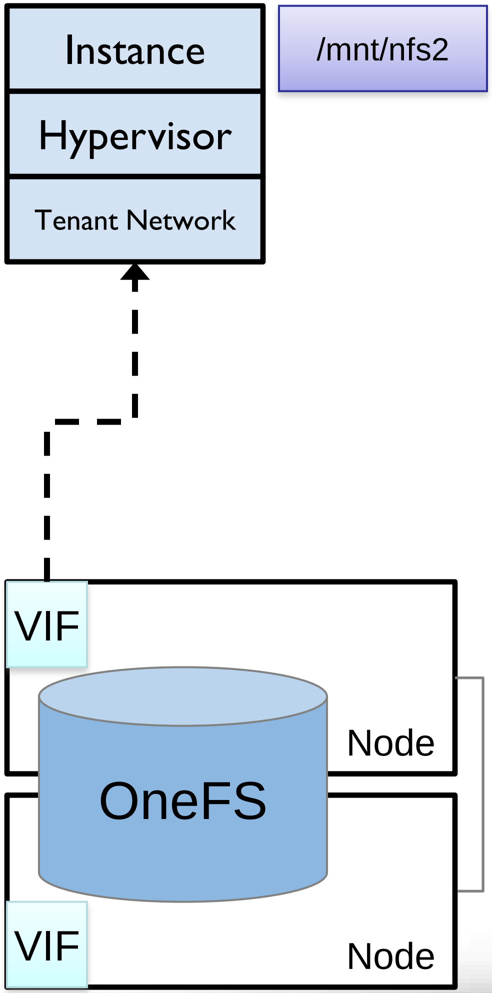
- Common driver w/ product plugins
  - Configuration parameters
  - Manila API
    - Product specific implementation in plugins
- Current “WIP” driver supports two storage platforms
  - More to come!

EMC VNX - Scale Up



- Today:
  - Prototype supports majority of Manila API
  - Currently requires that interface already exist on tenant network
  - The desired data mover or VDM must be specified in configuration file
- 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

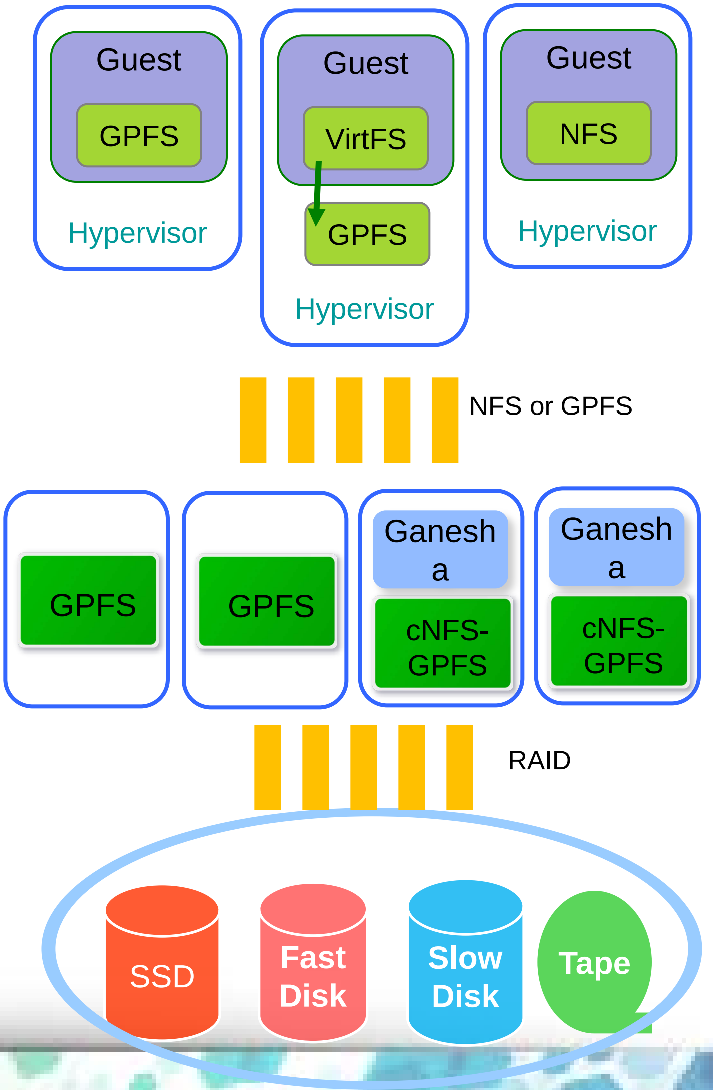
EMC Isilon - Scale Out



- 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"
  - Investigate QoS

IBM: Manila GPFS and NFS Driver

- Goal is for GPFS to provide a single data plane for all data center applications, , including Nova, Glance and Cinder (and even Swift)
  - Manila extends the data plane into the VM
- Support both cNFS and GPFS
  - cNFS adds monitoring and automatic failover to Ganesha NFS server and kNFS server
  - pNFS in roadmap
- Leverage Standard GPFS Enterprise features
  - Scale Out, High Performance, Highly Available, Encryption, Backup, DR, Declustered RAID
  - Tiered Policy-based Storage Pools
- Multi-Tenancy (WIP)
  - Service VM running GPFS / Ganesha for each tenant
  - GPFS Fileset for isolation and quota enforcement for each tenant
  - GPFS encryption provides further benefits such as secure delete
- Current status of NFS prototype
  - Have working single tenant manila driver with Ganesha NFS server
  - Plan to submit as WIP in Juno time frame



Manila: Project Status & Futures

- **Current Project Status**
  - Formally denoted as OpenStack Incubated Project on August 26th, 2014!
  - Customer PoCs have started
- **Current Development Focus**
  - Gateway-based multi-tenancy
  - Share type support (e.g. service levels)
  - Make Neutron dependency optional with pluggable network framework
  - ~~Finish multi-backend support~~
  - Automate mounting shares within instances
- **Expanded integrations**
  - More vendors & backends
  - More protocols (CephFS, GPFS, etc)

Get involved with Manila!

- https://github.com/stackforge/manila
- https://wiki.openstack.org/wiki/Manila
- #openstack-manila on IRC (Freenode)

Questions?