

# What the heck are DHSS Driver Modes in OpenStack Manila?



Victoria Martínez  
de la Cruz

Rodrigo Barbieri

Goutham Pacha Ravi

# Who we are

**Victoria Martínez de la Cruz**

Software Engineer, Red Hat Inc.

OpenStack Manila contributor

**Rodrigo Barbieri**

Software Developer, NetApp Inc.

OpenStack Manila core reviewer

**Goutham Pacha Ravi**

Software Developer, NetApp Inc.

OpenStack Manila core reviewer

# Why we are doing this presentation

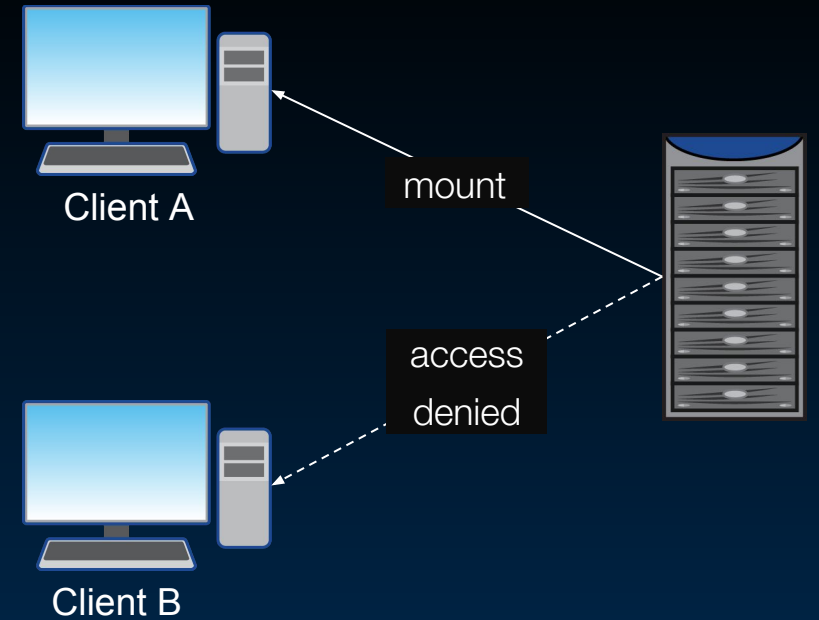
- DHSS (`driver_handles_share_servers`) is THE MOST IMPORTANT config option in Manila
- It is the first manila key concept that deployers stumble when deploying Manila
- It is something deployers should know before deploying Manila, so they can set up the network correctly
- It is documented, but even so, not very easy to understand

# Agenda

- Architecting NAS in a cloud
  - Simplest NAS deployment
  - Introducing multi-tenancy to our simplest NAS deployment
  - Concerns
  - An ideal NAS architecture
- The **OpenStack** Solution
  - Introduction to **Manila**
  - Driver modes
  - DHSS=False deployment
  - DHSS=True deployment
- Things to consider when deploying
- Future enhancements
- Questions

# Simplest NAS deployment

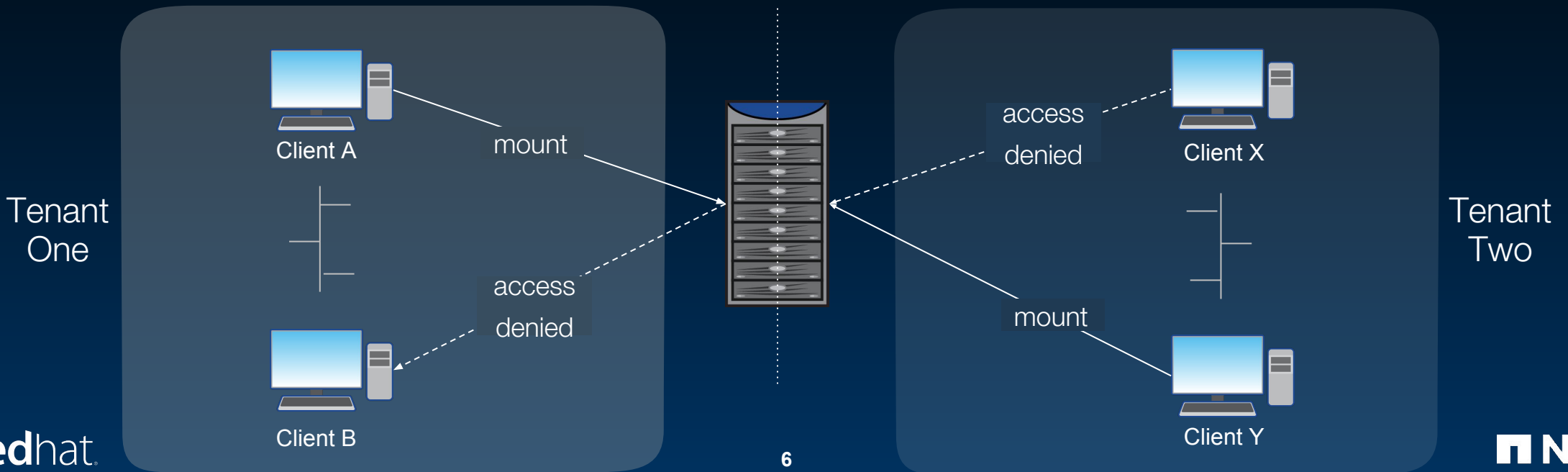
- 1) A server exports shared file systems over a network, we call it a share server
- 2) The share server controls access permissions to different clients
- 3) Client A is authorized and mounts a share provided by the share server
- 4) Client B is not authorized and cannot mount that same share



# Introducing Multi-tenancy

Security concerns arise when providing shares to multiple tenants

- Data isolation - the underlying filesystem should not be shared and exports should not be visible
- Network isolation - there should not be connectivity to prevent spoofing and unintended access
- Filesystem metadata isolation - Filesystems have metadata, the universe of users for tenants is going to be different.

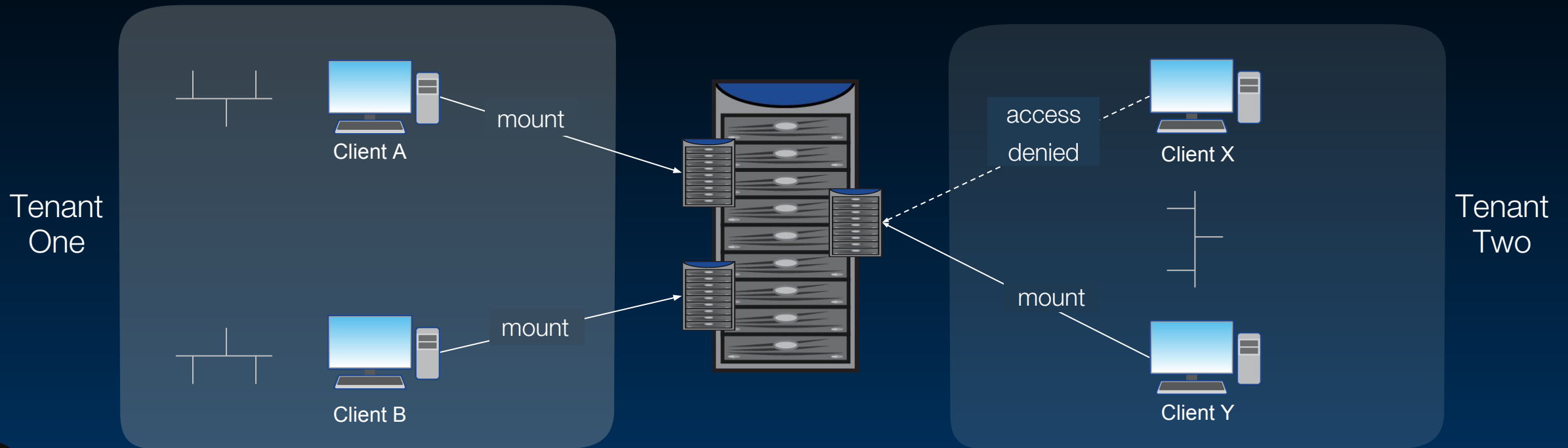


# Scaling through automation

- For single or few tenant clouds (ex: small private clouds), the desired level of isolation can be achieved by
  - Network segmentation outside of OpenStack (Provider Networks)
  - Isolated storage systems or share servers
- For multi-tenant clouds (ex: large private clouds, public clouds), the number of tenants can grow over time. The tasks involved to provision secure shared file systems get harder for a cloud administrator.

# An ideal NAS architecture

- Setting up unique share servers to provide shares to different tenants
- Provides data path and network isolation guarantees between tenants, even while using the same back end storage





# OpenStack Manila

- File share project in OpenStack
  - Provisioning of shared filesystems to VMs
- Manila was conceived with the ideal NAS architecture in mind
- Several supported protocols
  - NFS, CIFS, CephFS, MAPRFS, HDFS, GlusterFS
- Feature-filled
  - Quota Control
  - Share Migration
  - Tenant driven Share Replication
  - Snapshots for Cloning, Recovery and Reverting
  - Storage Service Catalog via Share Types
  - Access Control, Authentication Services
  - Grouping of shares, consistent snapshots



**MANILA**  
*an OpenStack Community Project*

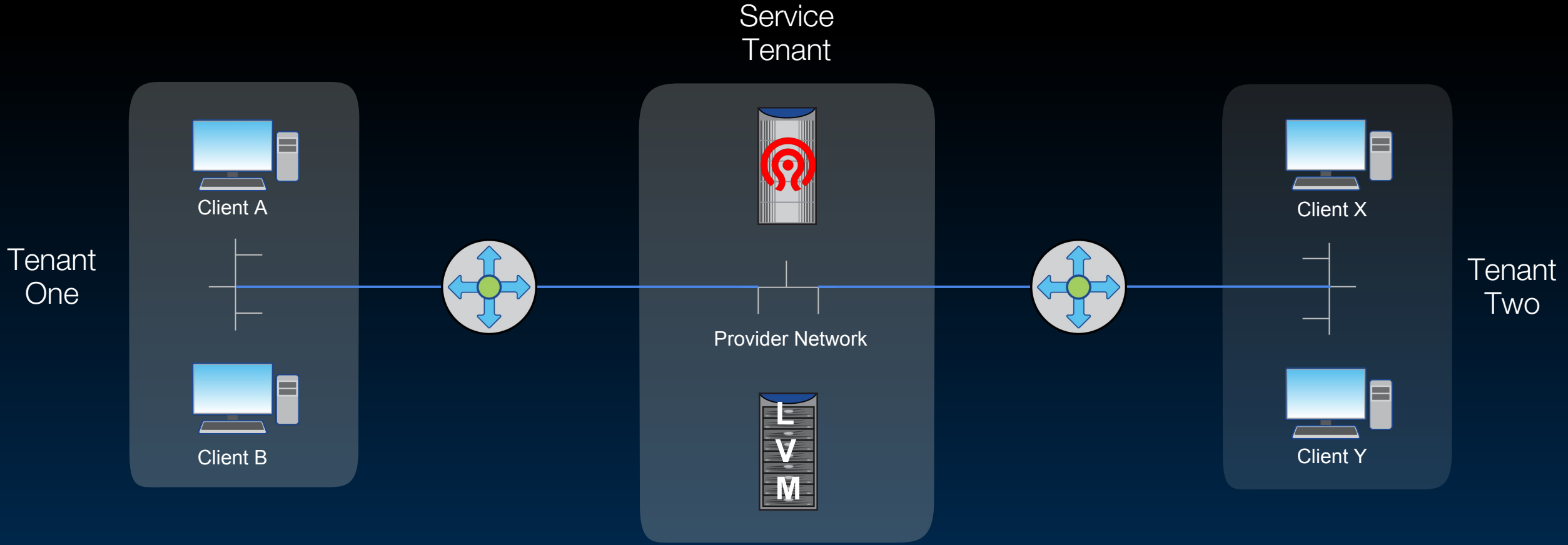
# Driver modes

- Some back ends cannot provide automated ways to scale share servers with isolation
- A flag `driver_handles_share_servers` (DHSS) was created to distinguish that capability
  - **True**: The driver creates multiple share servers to provide multi-tenancy isolation
  - **False**: The driver has a single share server and offer no multi-tenancy or isolation guarantees
- **Share drivers** operate in at least one of the two possible driver modes
  - One instance of the driver can only operate in one driver mode

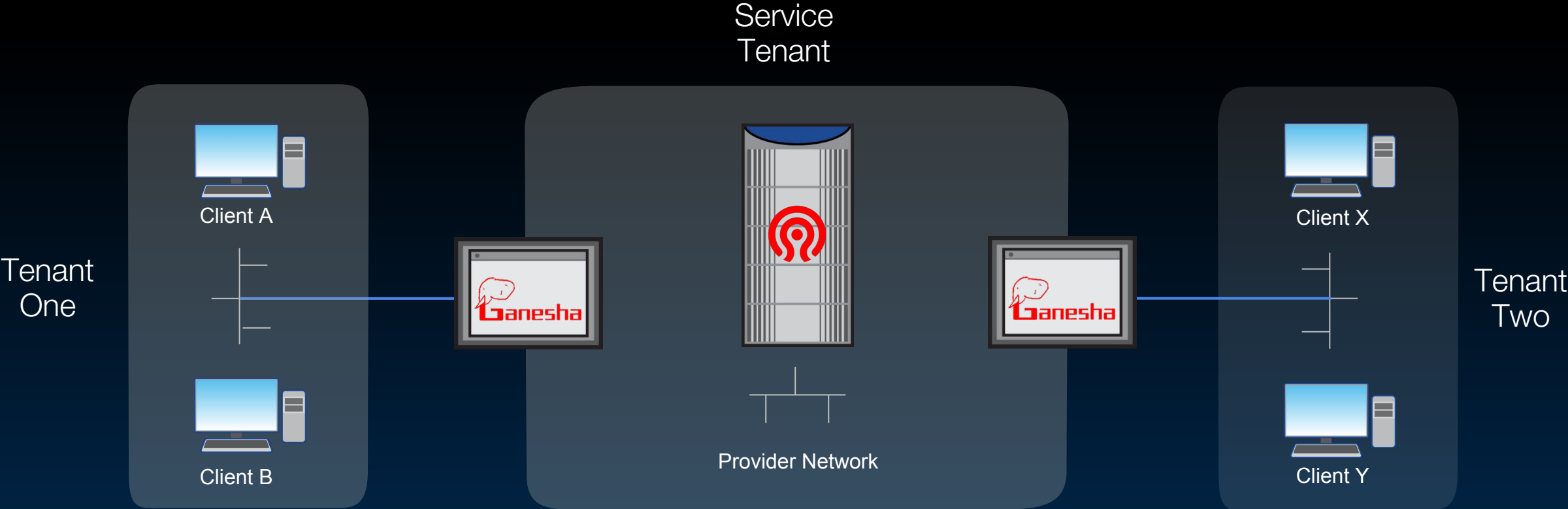
# DHSS = False

- Drivers have a single share server configured for each back end storage system
- All shares are to be provided by this share server, irrespective of the tenant consuming them
- Configuration complexity can be fairly low, especially networking
- Multi-tenancy, data path and network isolation could be achieved outside of Manila, but may not be guaranteed
- Limited by scale, ideal for private clouds with a small number of tenants

# DHSS = False Networking using LVM and CephFS Native drivers



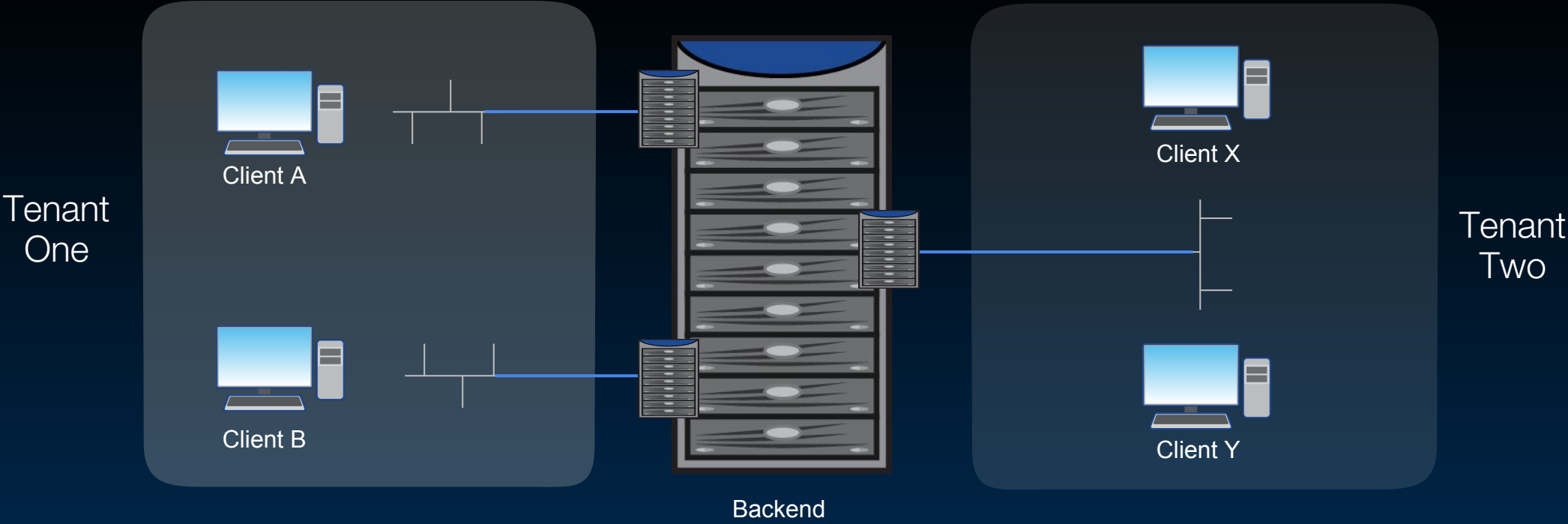
# Let's achieve isolation by playing with the networking



# DHSS = True

- Drivers create share servers per share network
- This multi-tenant focused mode guarantees isolation and provides scalability
- Manila manages the lifecycle of the share server and the associated networking necessary. No administrator intervention is necessary
- Supports tenant defined authentication mechanisms and ACL domains
  - LDAP
  - Active Directory
  - Kerberos

# DHSS = True Networking Highlights



# Demo of a driver configured in DHSS = True mode



**MANILA**



**CINDER**



**NEUTRON**



**NOVA**



# CIFS / Active Directory in a multi-tenant cloud



**MANILA**



**NetApp**

# Things to consider

- The `driver_handles_share_servers` configuration option MUST be specified for each backend stanza in `manila.conf`
- For any driver mode, plan your networking design carefully before deploying
- Share servers are abstracted away from end users, users can request shares to be exported on a “share-network” that they designate. In most cases, this would be the private Neutron network that tenants set up to host their VMs on.
- One or more security services can be associated with a share network.

# Roadmap

- Share Server HA
- Improvements to the Generic Driver
- Support for Dual IPv6, IPv4 networking
- Support for Replication in DHSS=True driver mode
- Integrating Neutron L2GW

# Questions?

Victoria Martínez de la Cruz  
IRC: *vkmc*  
email: *victoria@redhat.com*

Rodrigo Barbieri  
IRC: *ganso*  
email: *rodrigo.barbieri2010@gmail.com*

Goutham Pacha Ravi  
IRC: *gouthamr*  
email: *gouthampravi@gmail.com*

# Thank You

