# Gluster - Future Roadmap

- Atin Mukherjee
  @mukherjee_atin

May 2016

# Credits

Some slides/content borrowed & stolen from:

Jeff Darcy

Luis Pabon

Prasanna Kalever

Vijay Bellur

# Agenda

- What is Gluster.Next?
- How Gluster.Next?
- Why Gluster.Next?
- When Gluster.Next?

# Gluster.Next

# What?

# Gluster.Today

- Scale-out storage system
- Aggregates storage exports over network interconnects to provide an unified namespace
- File, Object, API and Block interfaces
- Layered on disk file systems that support extended attributes

# Gluster.Today - Features

- Scale-out NAS
  - Elasticity, quotas
- Data Protection and Recovery
  - Volume Snapshots
  - Synchronous & Asynchronous Replication
  - Erasure Coding
- Archival
  - Read-only, WORM, bitrot detection
- Native CLI / API for management

# Gluster.Today - Features

- Isolation for multi-tenancy
  - SSL for data/connection, Encryption at rest
- Performance
  - Data, metadata and readdir caching, tiering
- Monitoring
  - Built in io statistics, /proc like interface for introspection
- Provisioning
  - Puppet-gluster, gdeploy
- More..

# What is Gluster.Next?

- Gluster.today
  - Client driven Distribution, Replication and Erasure Coding (with FUSE)
  - Spread across 3 - 100s of nodes
  - Geared towards “Storage as a Platform”

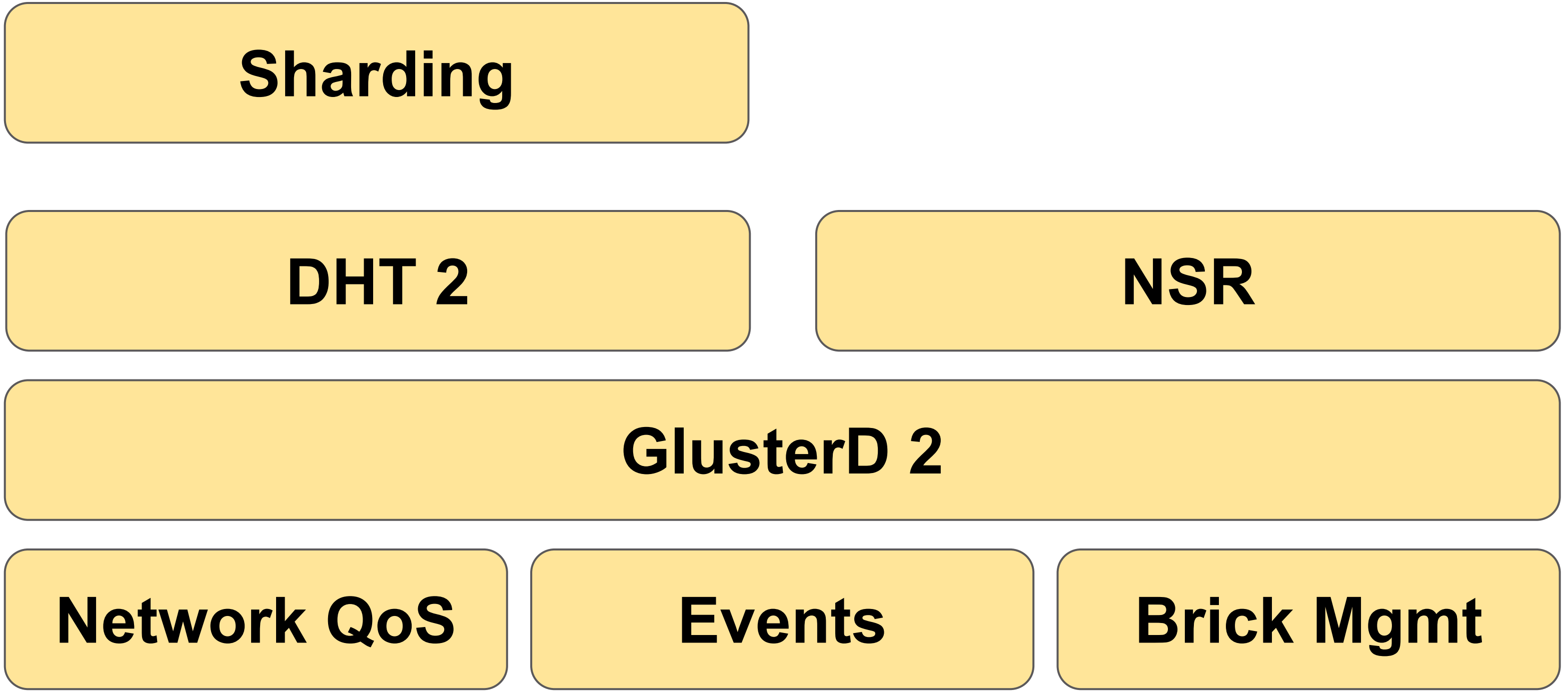
# What is Gluster.Next?

- Gluster.Next
  - Architectural Evolution Spanning over multiple releases (3.8 & 4.0)
  - Scale-out to 1000s of nodes
  - Choice of Distribution, Replication and Erasure Coding on servers or clients
  - Geared towards “Storage as a Platform” and “Storage as a Service”
  - Native REsTful management & eventing for monitoring

# Gluster.Next

# How?

# Gluster.Next - Main Components



Sharding

DHT 2

NSR

GlusterD 2

Network QoS

Events

Brick Mgmt

# DHT 2

- Problem: directories on all subvolumes
  - directory ops can take O(n) messages
- Solution: each directory on one subvolume
  - can still be replicated etc.
  - each brick can hold data, metadata, or both
  - by default, each is both just like current Gluster

# DHT 2 (continued)

- Improved layout handling
  - central (replicated) instead of per-brick
  - less space, instantaneous “fix-layout” step
  - layout generations help with lookup efficiency
- Flatter back-end structure
  - makes GFID-based lookups more efficient
  - good for NFS, SMB

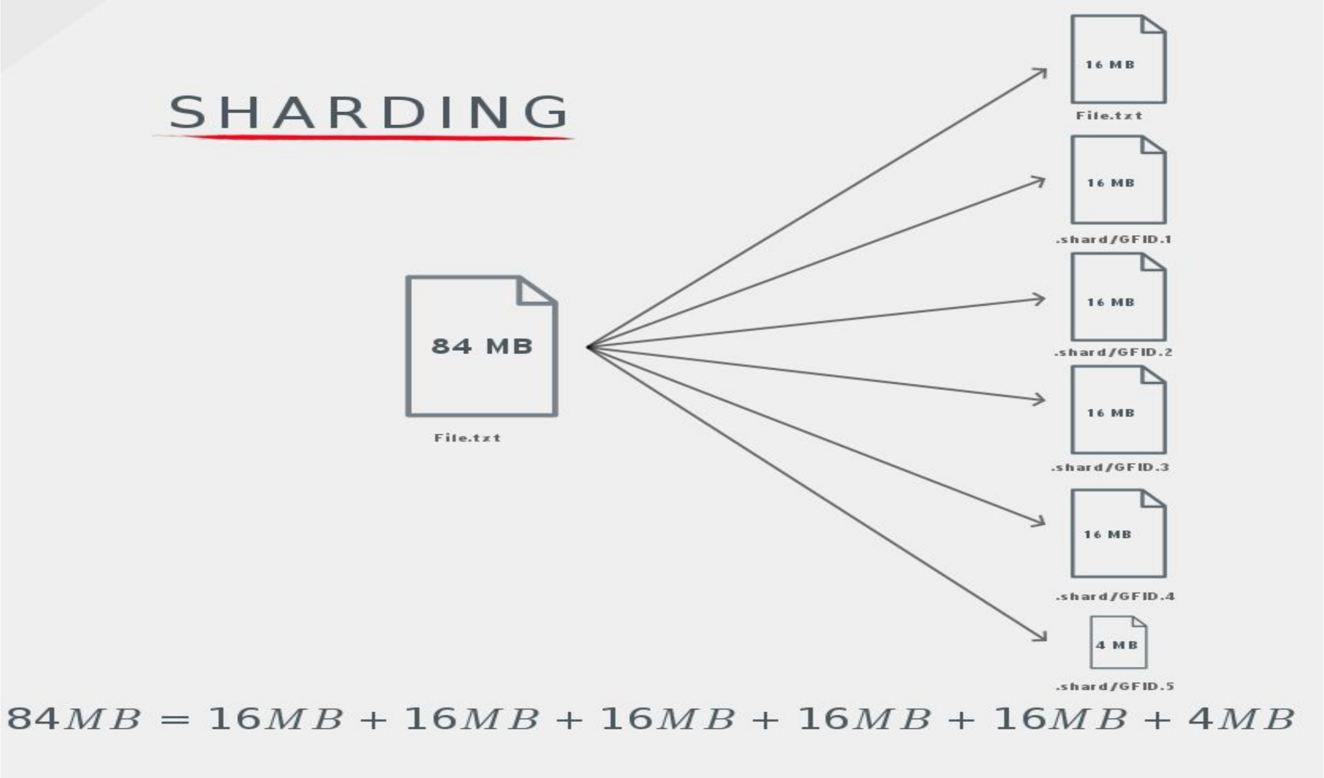
# ~~NSR~~ (JBR)

- Server-side with temporary leader
  - vs. client-side, client-driven
  - can exploit faster/separate server network
- Log/journal based
  - can exploit flash/NVM (“poor man’s tiering”)
- More flexible consistency options
  - fully sync, ordered async, hybrids
  - can replace geo-replication for some use cases

# Sharding

- Spreads data blocks across a gluster volume
- Primarily targeted for VM image storage
- File sizes not bound by brick or disk limits
- More efficient healing, rebalance and geo-replication
- Yet another translator in Gluster

# Sharding Illustrated



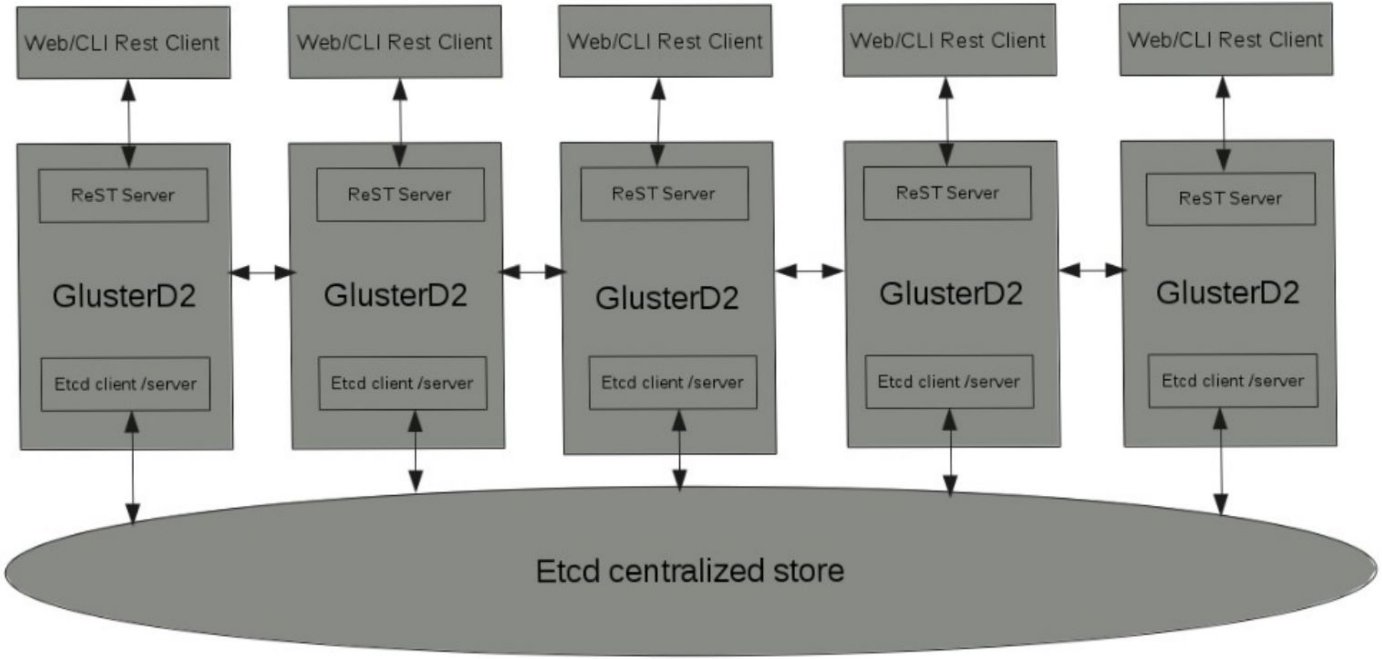
# Network QoS

- Necessary to avoid hot spots at high scale
  - avoid starvation, cascading effects
- A single activity or type of traffic (e.g. self-heal or rebalance) can be:
  - directed toward a separate network
  - throttled on a shared network
- User gets to control front-end impact vs. recovery time

# GlusterD2

- More efficient/stable membership
  - especially at high scale
- Stronger configuration consistency
- Modularity and plugins
- Exposes ReST interfaces for management
- Core implementation in Go

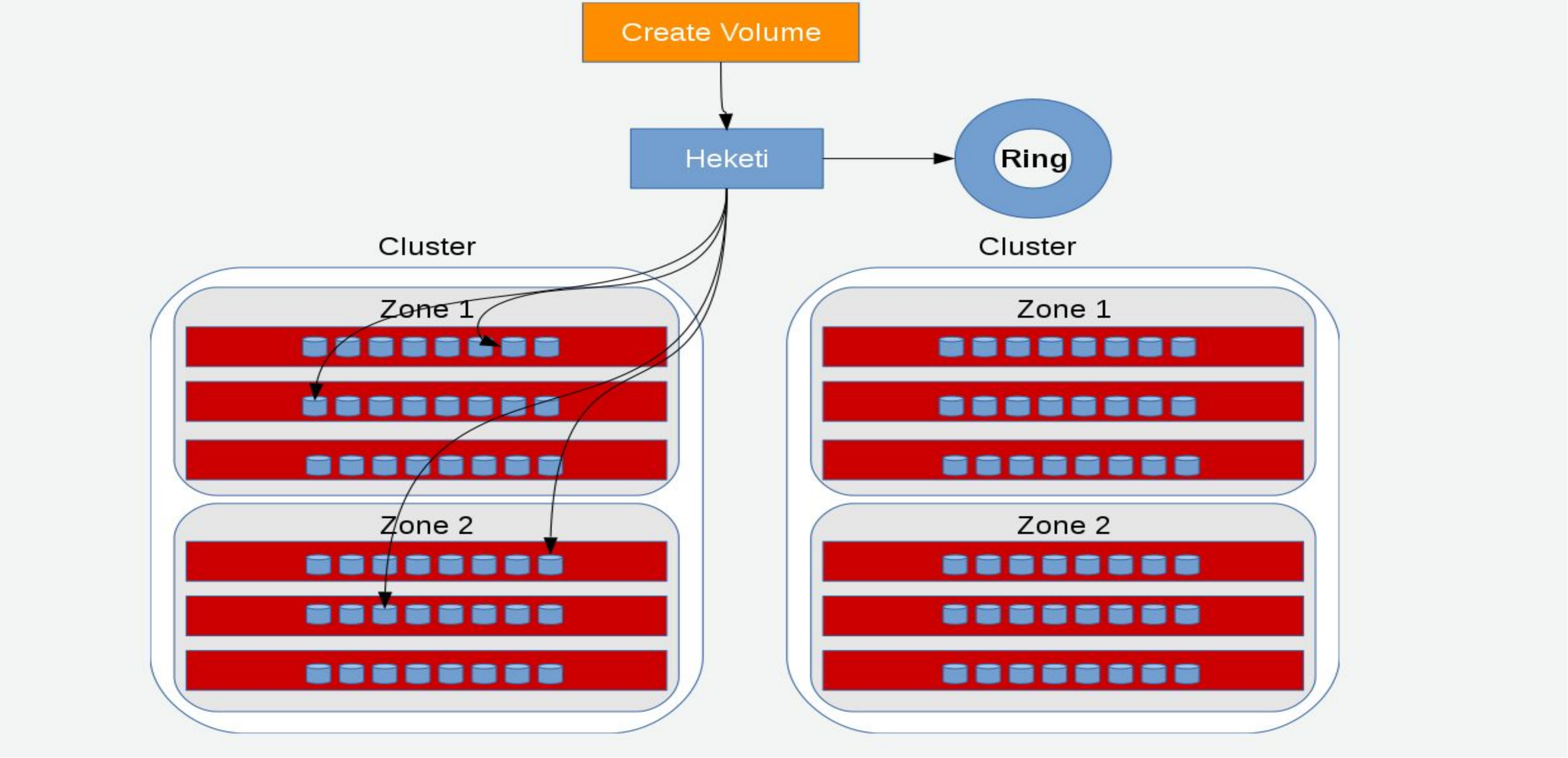
# GlusterD2 - Architecture



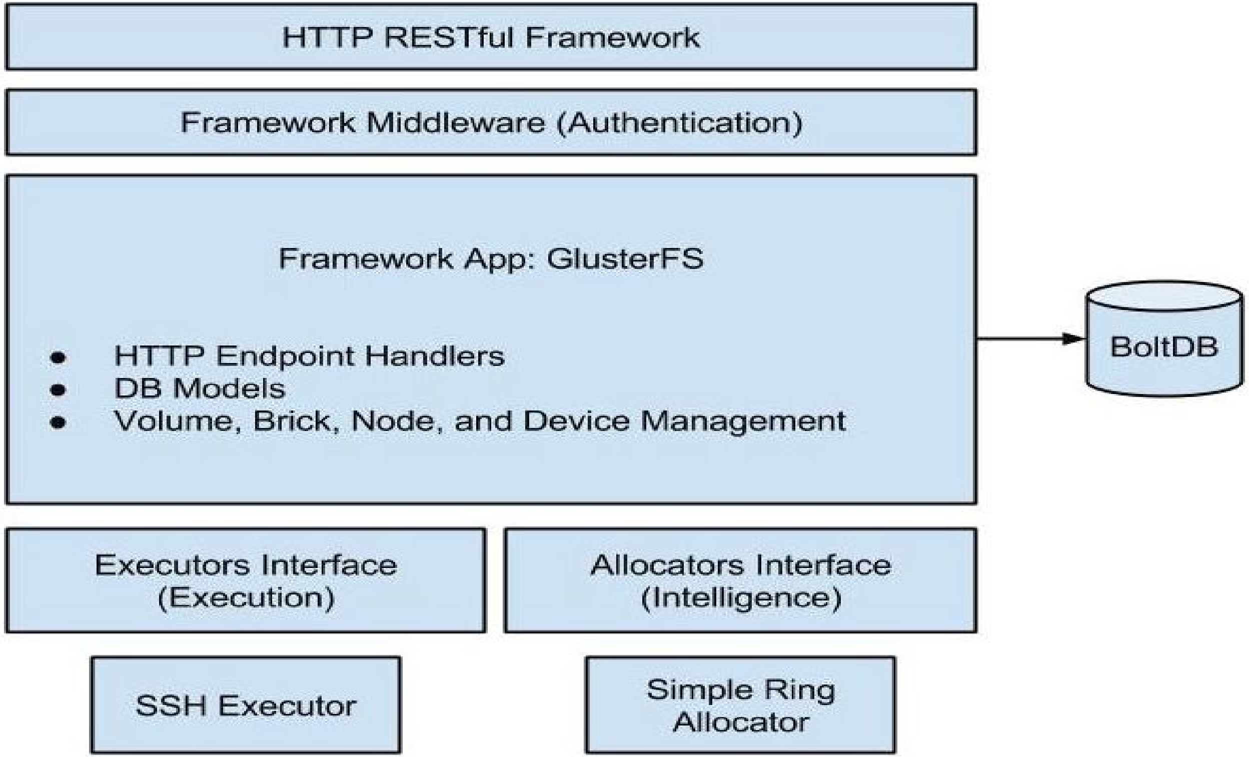
# Heketi

- Dynamic Share Provisioning with Gluster volumes
- Eases brick provisioning - LVs, VGs, filesystem etc.
- Automatically determines brick locations for fault tolerance
- Exposes high level ReST interfaces for management
  - create share, expand share, delete share etc.

# Heketi Illustrated



# Heketi - Architecture



HTTP RESTful Framework

Framework Middleware (Authentication)

Framework App: GlusterFS

- HTTP Endpoint Handlers
- DB Models
- Volume, Brick, Node, and Device Management

BoltDB

Executors Interface (Execution)

Allocators Interface (Intelligence)

SSH Executor

Simple Ring Allocator

# Event Framework

- Export node and volume events in a more consumable way
- Support external monitoring and management

# Brick Management

- Multi-tenancy, snapshots, etc. mean more bricks to manage
  - possibly exhaust cores/memory
- One daemon/process must handle multiple bricks to avoid contention/thrashing
  - core infrastructure change, many moving parts

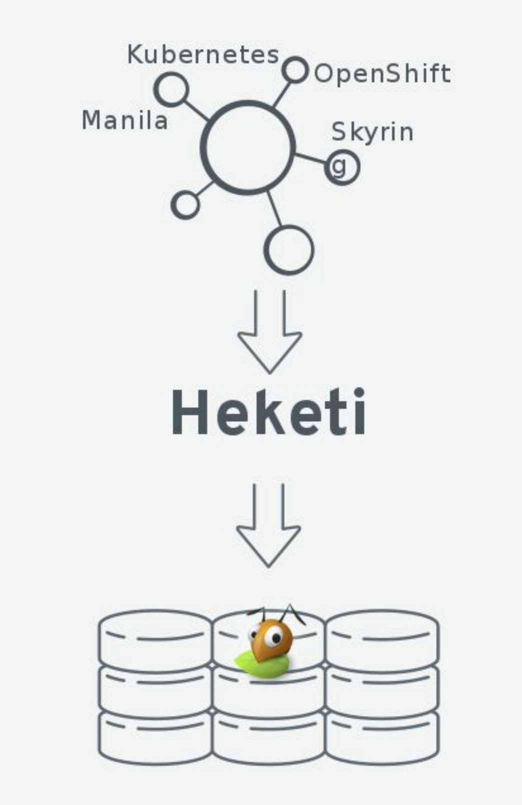
# Gluster.Next

# Why?

# Why Gluster.Next?

- Paradigm Changes in IT consumption
  - Storage as a Service & Storage as a Platform
  - Private, Public & Hybrid clouds
- New Workloads
  - Containers, IoT, <buzz-word> demanding scale
- Economics of Scale

# Why Gluster.Next: StaaS



- User/Tenant driven provisioning of shares.
- Talk to as many Gluster clusters and nodes
- Tagging of nodes for differentiated classes of service
- QoS for preventing noisy neighbors

# Why Gluster.Next: Containers

- Persistent storage for stateless Containers
  - Non-shared/Block : Gluster backed file through iSCSI
  - Shared/File: Multi-tenant Gluster Shares / Volumes

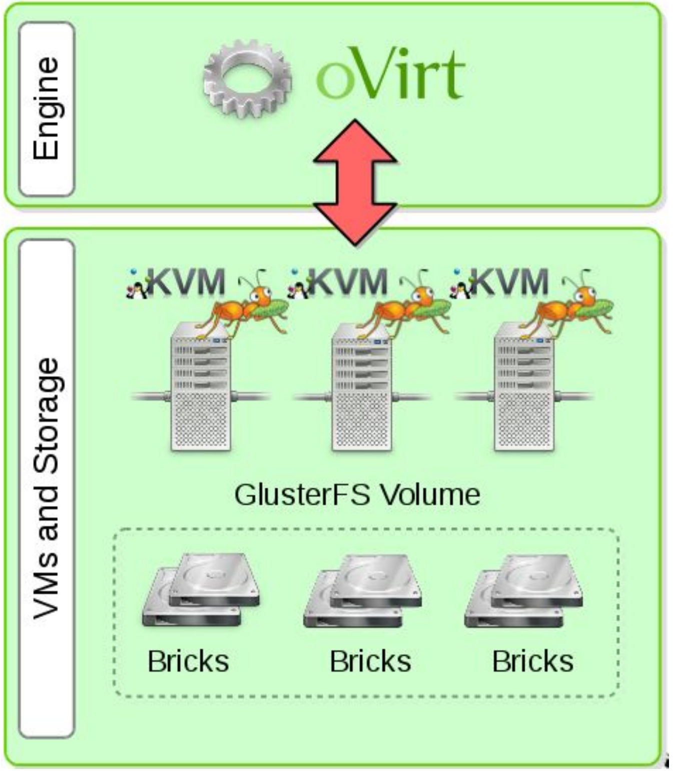
- Shared Storage for container registries
  - Geo-replication for DR

- Heketi to ease provisioning
  - “Give me a non-shared 5 GB share”
  - “Give me a shared 1 TB share”

- Shared Storage use cases being integrated with Docker, Kubernetes & OpenShift

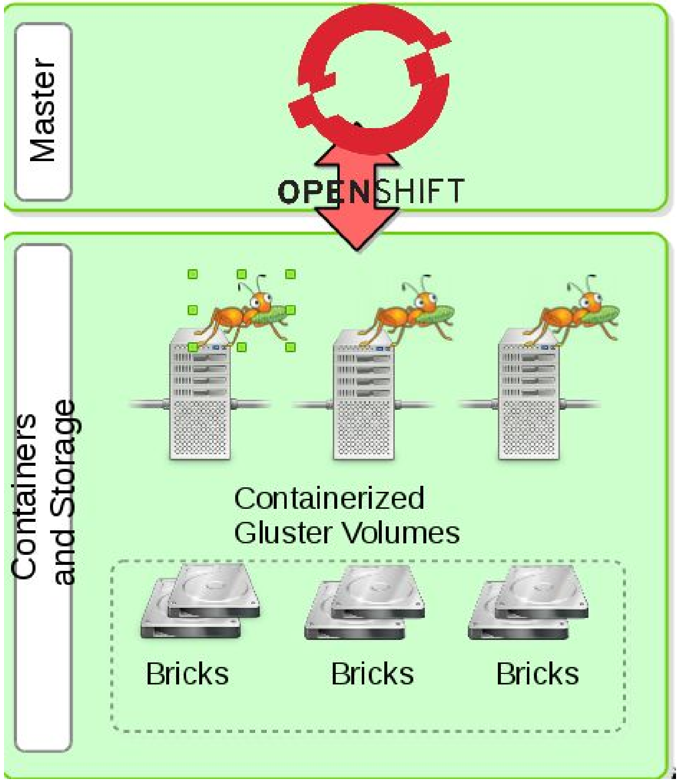
# Why Gluster.Next: Hyperconvergence with VMs

- Gluster Processes are lightweight
- Benign self-healing and rebalance with sharding
- oVirt & Gluster - already integrated management controller
- geo-replication of shards possible!



# Why Gluster.Next: Containers converged with OpenShift

- Server nodes are used both for containers and storage
- Containerized Gluster exports bind mounted directories from hosts
- Tenants consume volumes or sub-directories of volumes exported through FUSE



# Gluster.Next

# When?

# Gluster.Next Phase 1

## Gluster 3.8

- June 2016
- Stabilize Sharding
- Compound FOPs
- Improvements for NFS/SMB accesses
    - Leases, RichACLs, Mandatory Locks etc.
- UNIX-domain sockets for I/O
    - slight boost in hyperconverged setups

# Gluster.Next Phase 2

## Gluster 4.0

- May/June 2017
- Everything that we’ve discussed so far
- And more..

# Other Stuff

- IPv6 support
- “Official” FreeBSD support
- Compression
- Code generation
  - reduce duplication, technical debt
  - ease addition of new functionality
- New tests and test infrastructure

# Thank You!

Resources:
gluster-devel@gluster.org
gluster-users@gluster.org

http://twitter.com/gluster
IRC: #gluster, #gluster-dev on Freenode