RED HAT GLUSTER STORAGE TECHNICAL PRESENTATION

MARCEL HERGAARDEN
Sr. Solution Architect Benelux

APRIL, 2017

INTRODUCTION

RED HAT GLUSTER STORAGE

Open source, software-defined storage for unstructured file data at petabyte scale

- Media, video
- Machine, Log Data
- GeoSpatial
- Persistent Storage
- Documents

RED HAT GLUSTER STORAGE ADVANTAGES

OPEN Open, software-defined distributed file and object storage system	• Based on GlusterFS open source community project • Uses proven local file system (XFS) • Data is stored in native format
SCALABLE No Metadata Server	• Uses an elastic hashing algorithm for data placement • Uses local filesystem's xattrs to store metadata • Nothing shared scale-out architecture
ACCESSIBLE Multi-Protocol the Same Data	• Global name space • NFS, SMB, Object (SWIFT+S3), Gluster native protocol • Posix compliant
MODULAR No Kernel Dependencies	• GlusterFS is based on filesystem in userspace (FUSE) • Modular stackable arch allows easy addition of features ...without being tied to any kernel version
ALWAYS-ON High-Availability across data, systems and applications	• Synchronous replication with self-healing for server failure • Asynchronous geo-replication for site failure

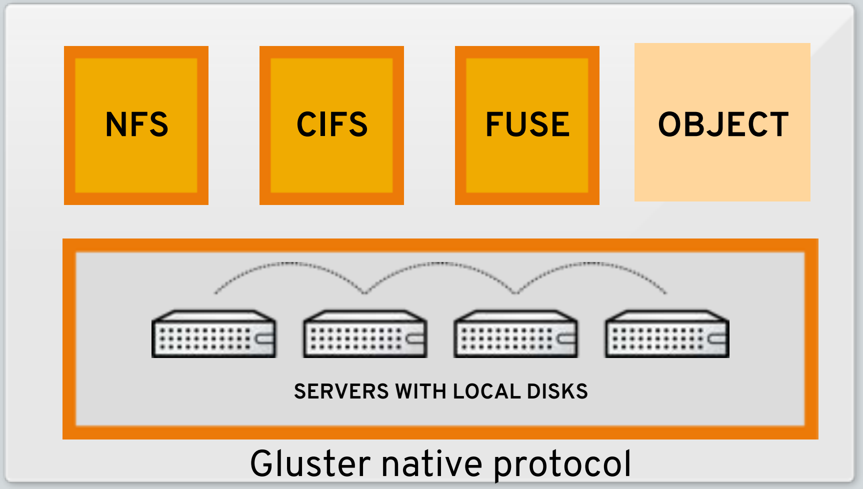
redhat.

TERMINOLOGY

- **Brick:** basic unit of storage, realized as mount point
- **Subvolume:** a brick after being processed by at least one translator
- **Volume:** logical collection of bricks / subvolumes
- **glusterd:** process to manage glusterfsd processes on the server
- **glusterfsd:** process to manage one specific brick
- **GFID:** 128 bit identifier of file in GlusterFS
- **(Trusted) Storage Pool:** Group of GlusterFS servers that know and trust each other

GLUSTER ARCHITECTURE

Distributed scale out storage using industry standard hardware

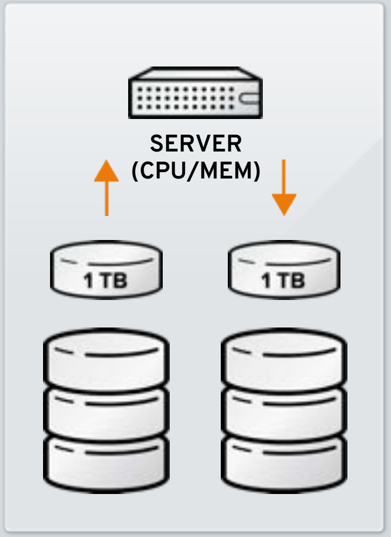


Aggregates systems to one cohesive unit
and presents using common protocols

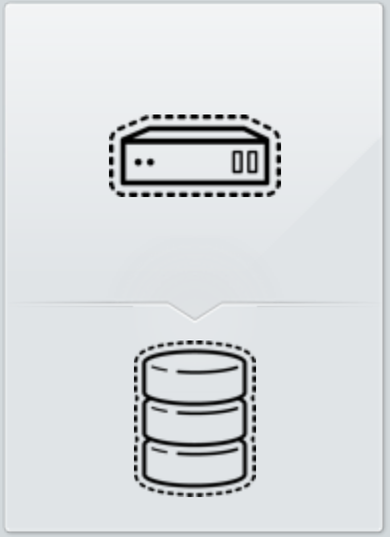
WHAT IS A SYSTEM?

Can be physical, virtual or cloud

PHYSICAL

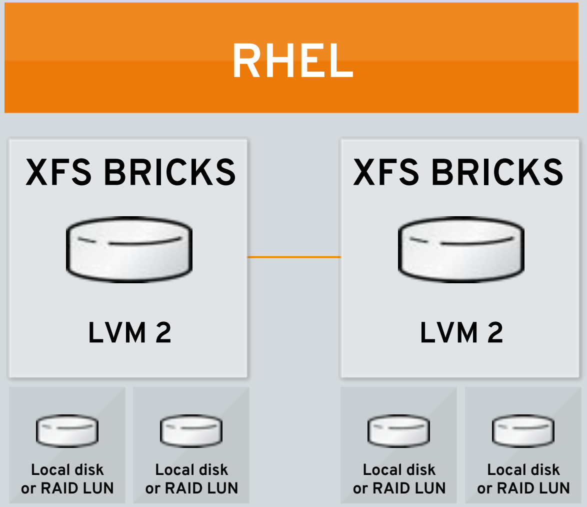


VIRTUAL



CLOUD

ANATOMY OF A SYSTEM

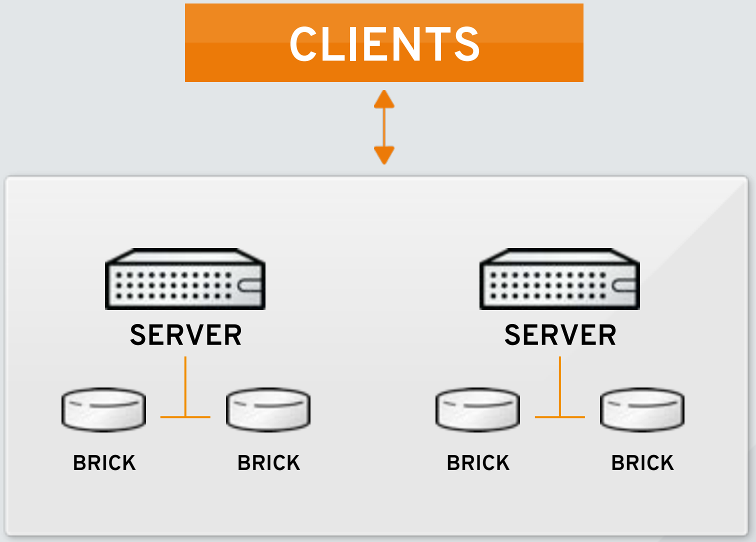


RHEL and Gluster make disk resources clustered and available as bricks using proven technology such as LVM and XFS

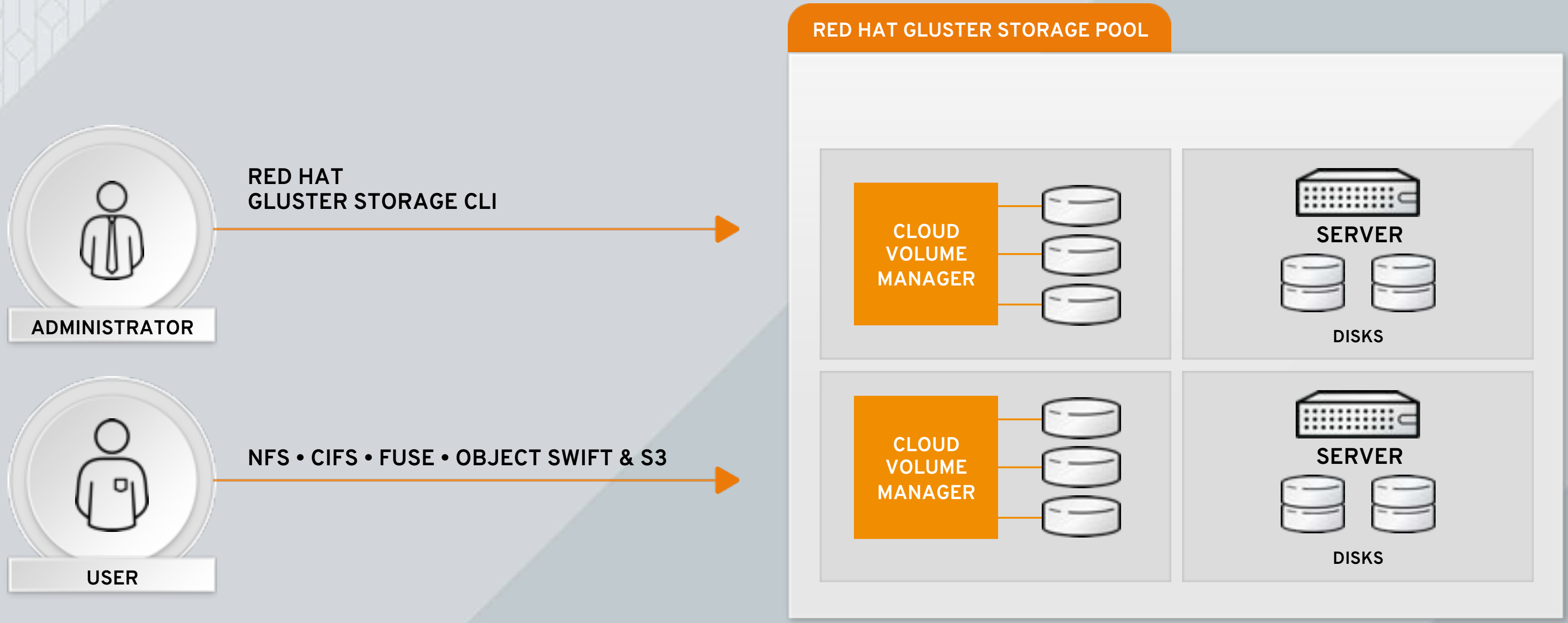
GLUSTER VOLUME

Bricks coming from multiple hosts become one addressable unit

- High availability as needed
- Load balanced data
- Managed by Gluster

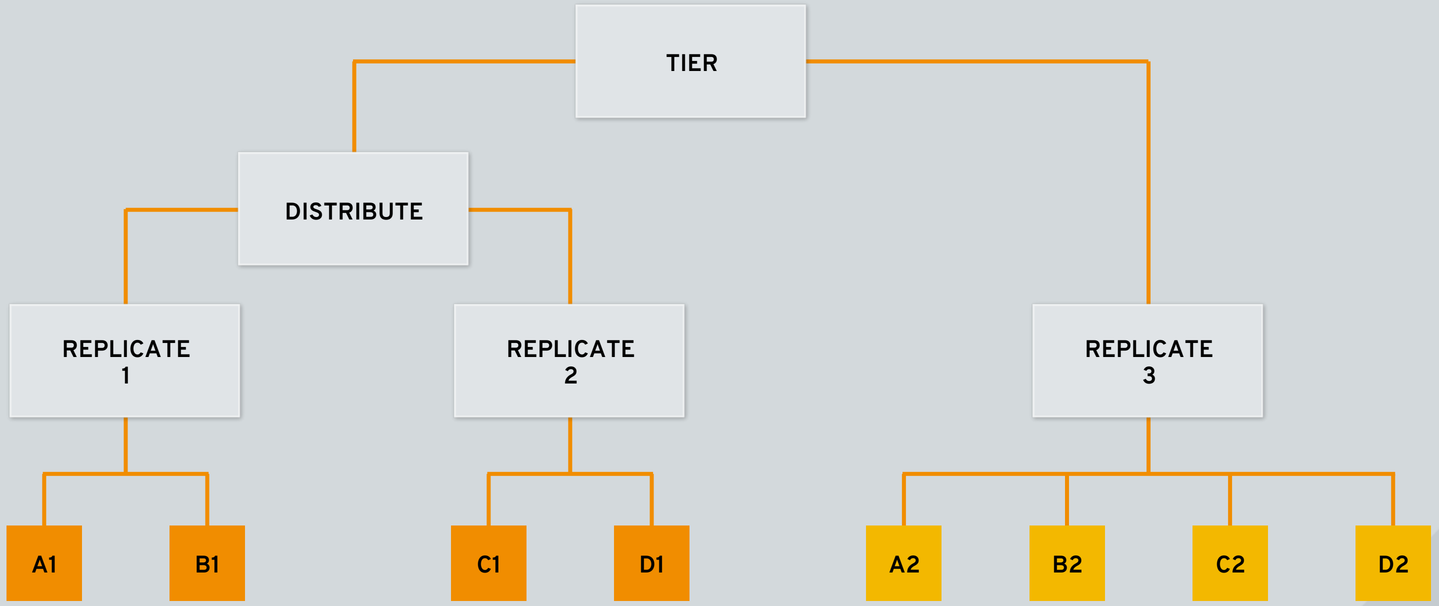


RED HAT GLUSTER STORAGE ARCHITECTURE

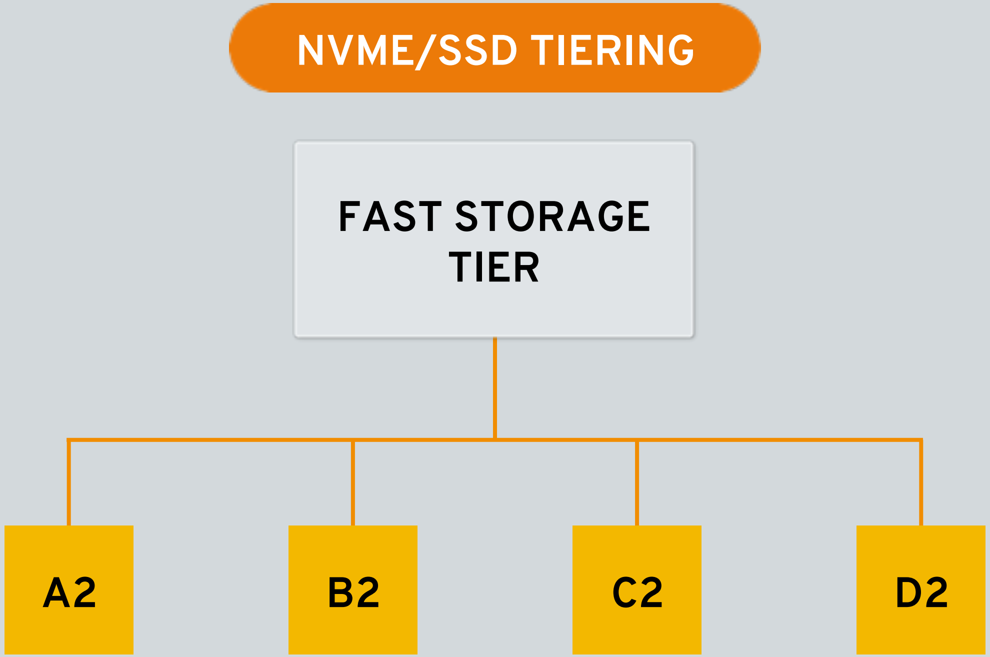


TIERING

MULTIPLE VOLUMES WITH
DIFFERENT STORAGE PROFILES



TIERING



DATA PLACEMENT

BASIC COMPONENT CONCEPTS

SERVER/NODES

Contain
the bricks

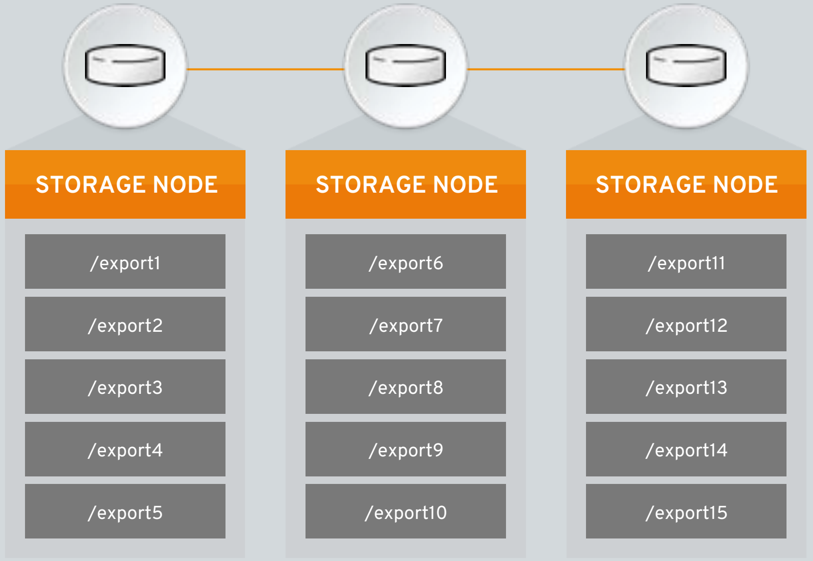
BRICK

The basic unit
of storage

VOLUME

A namespace presented as
a POSIX mount point

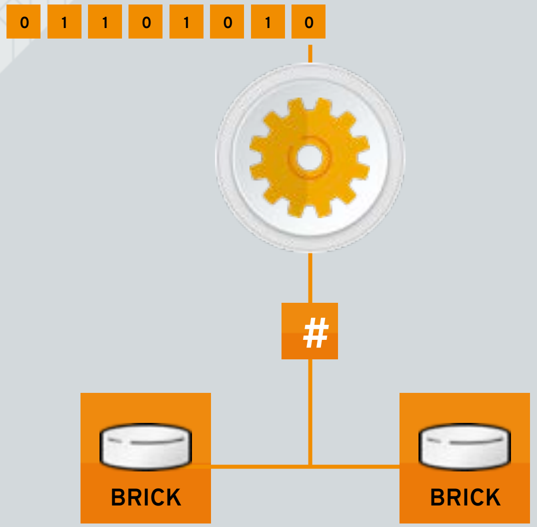
BRICKS



A Brick is the combination of a node and file system (hostname:/dir)

- Each brick inherits limits of underlying file system (XFS)
- Red Hat Gluster Storage operates at the brick level, not the node level
- Ideally, each brick in a volume should be the same size

ELASTIC HASH ALGORITHM



No Central Metadata Server

- Suitable for unstructured data storage
- No single point of failure

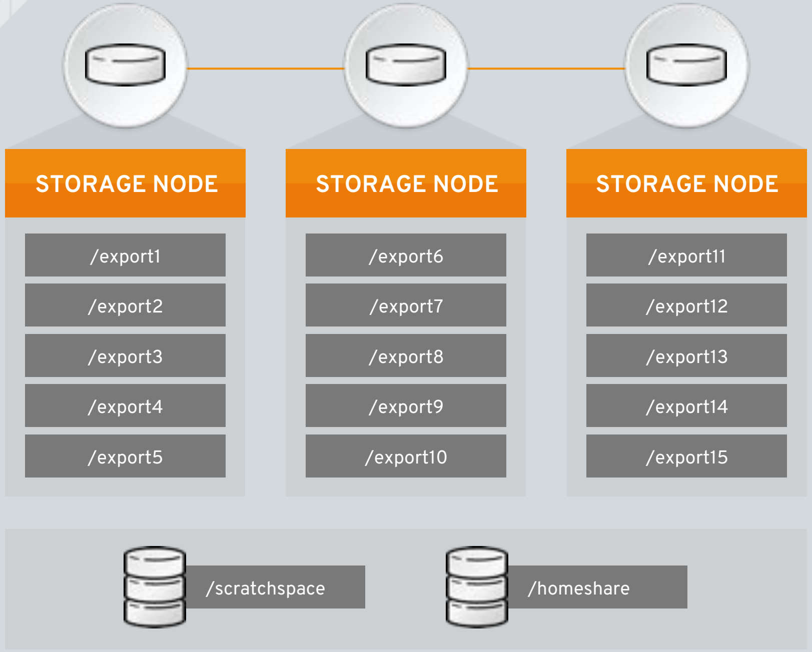
Elastic Hashing

- Files assigned to virtual volumes
- Virtual volumes assigned to multiple bricks
- Volumes are easily reassigned on-the-fly

Location Hashed on Filename

- No performance bottleneck
- Eliminates risk scenarios

GLUSTER VOLUMES



A Volume is a number of bricks >1, exported by Red Hat Gluster Storage

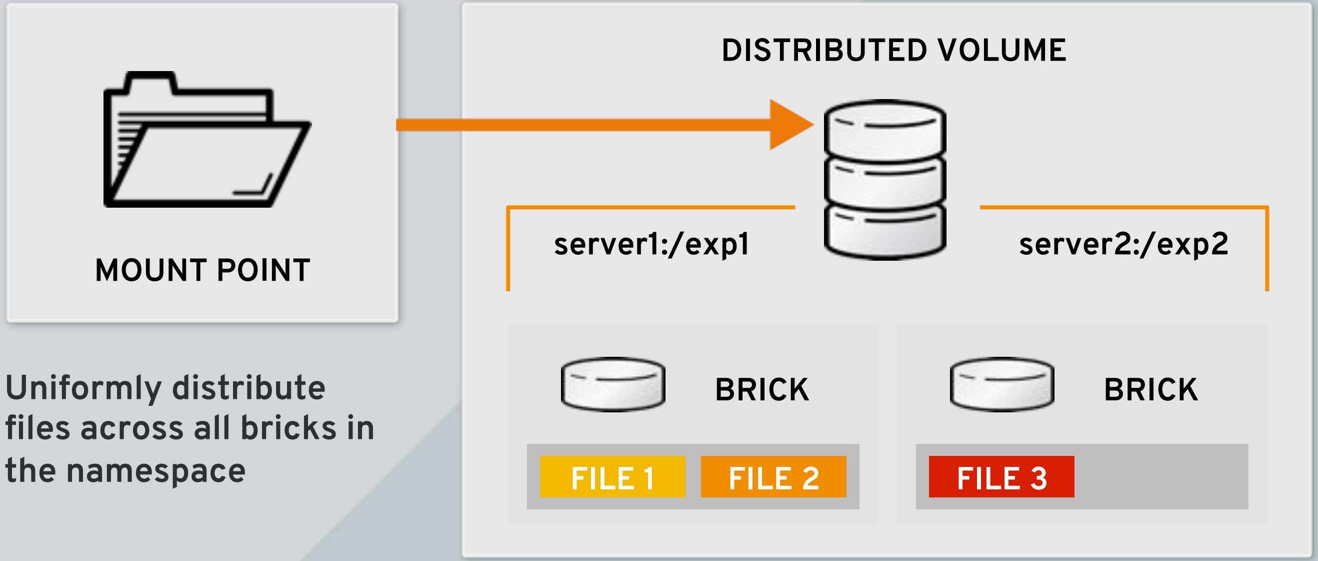
- Volumes have administrators assigned names
- A brick can be a member of one volume
- Data in different volumes physically exists on different bricks
- Volumes can be mounted on clients

DATA PLACEMENT STRATEGIES

VOLUME TYPE	CHARACTERISTICS
Distributed	• Distributes files across bricks in the volume • Used where scaling and redundancy requirements are not important, or provided by other hardware or software layers
Replicated	• Replicates files across bricks in the volume • Used in environments where high availability and high reliability are critical • Protection provided by software
Distributed-Replicated	• Offers improved read performance in most environments • Used in environments where high reliability and scalability are critical
Erasure Coded	• Sharded Volume type • Protection without dual or triple replication • Economical alternative, very suitable for archive like workload types
Tiered	• NVME/SSD Volume Tier • Performance enhancement for workloads with often requested files and small files

DEFAULT DATA PLACEMENT

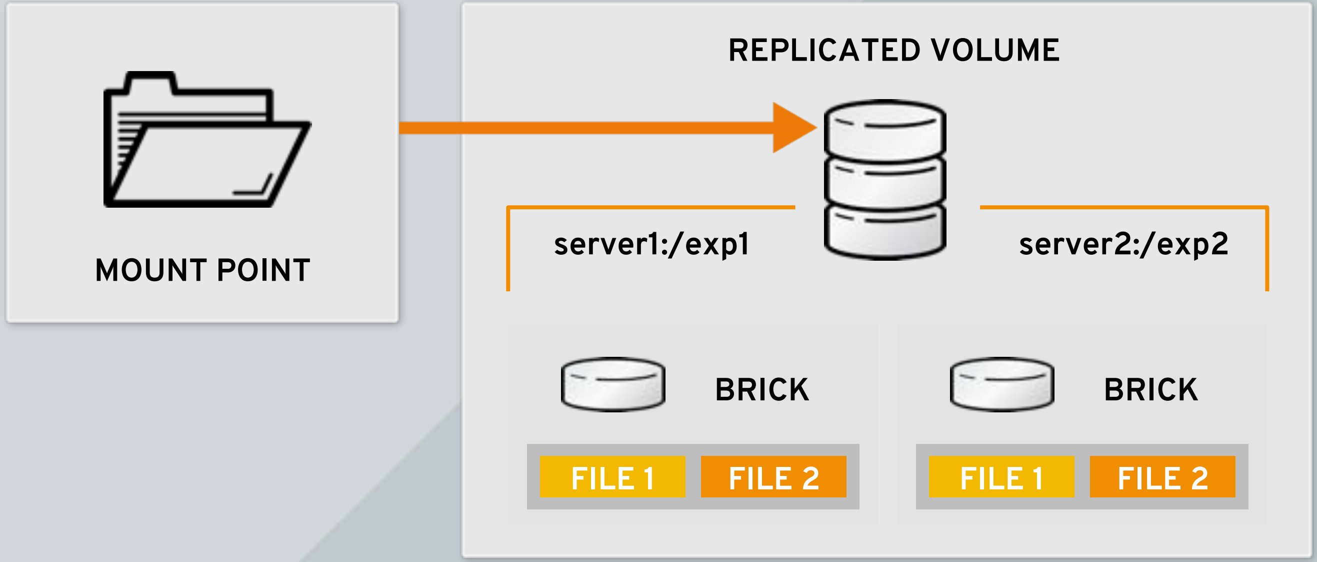
Distributed Volume



Uniformly distribute files across all bricks in the namespace

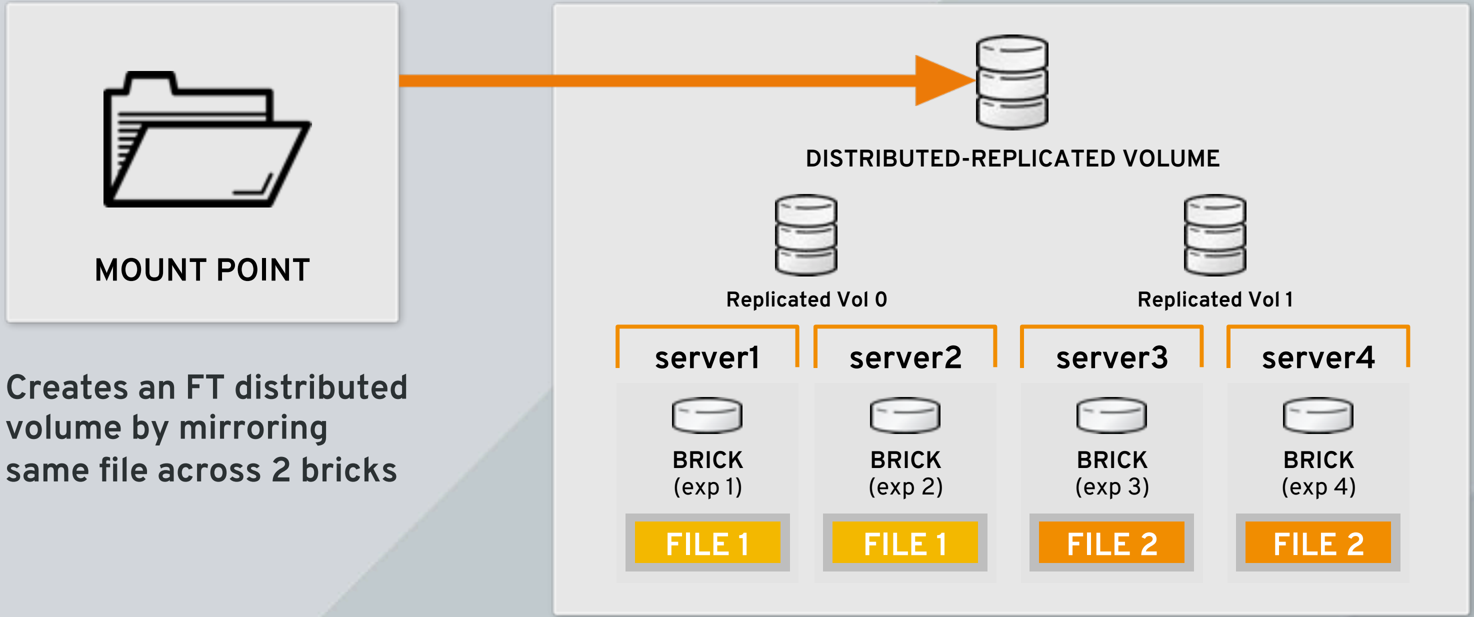
DEFAULT DATA PLACEMENT

Replicated Volume



FAULT-TOLERANT DATA PLACEMENT

Distributed-Replicated Volume

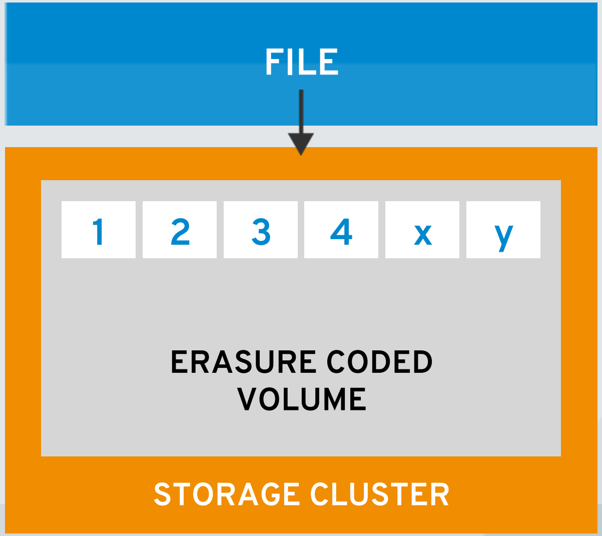


Creates an FT distributed volume by mirroring same file across 2 bricks

ERASURE CODING

Storing more data with less hardware

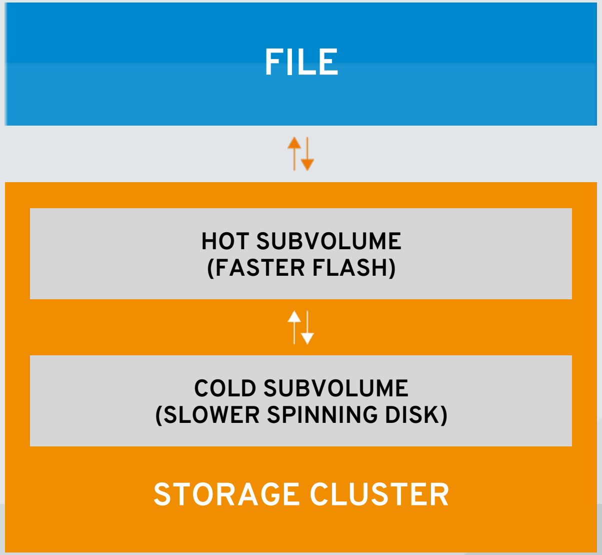
- **RECONSTRUCT** corrupted or lost data
- **ELIMINATES** the need for RAID
- **CONSUMES FAR LESS SPACE** than replication
- **APPROPRIATE** for capacity-optimized use cases.



TIERING

Cost-effective flash acceleration

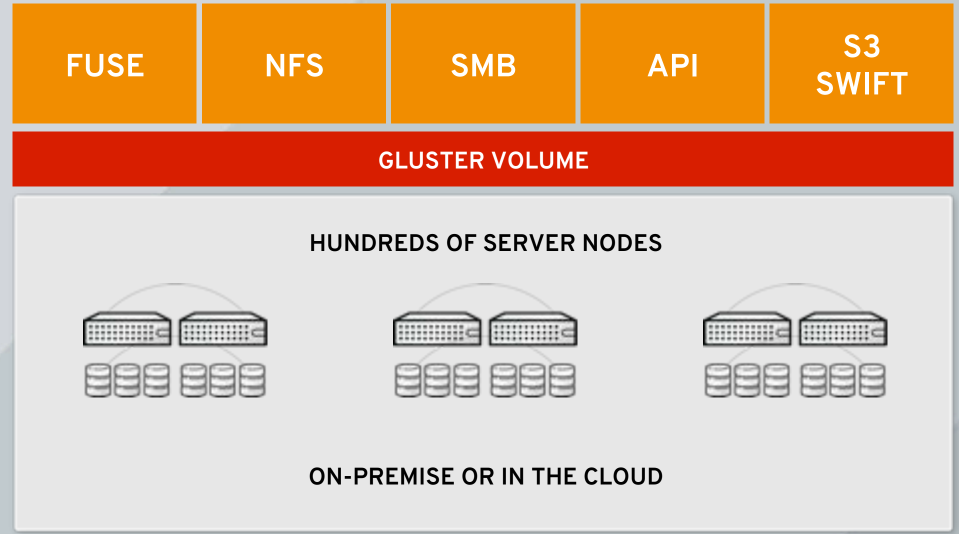
- **AUTOMATED** promotion and demotion of data between “hot” and “cold” sub volumes
- **BASED** on frequency of access.


FILE
HOT SUBVOLUME (FASTER FLASH)
COLD SUBVOLUME (SLOWER SPINNING DISK)
STORAGE CLUSTER

DATA ACCESSIBILITY

MULTI-PROTOCOL ACCESS

Primarily accessed as scale-out file storage with optional APIs, Swift or S3 object



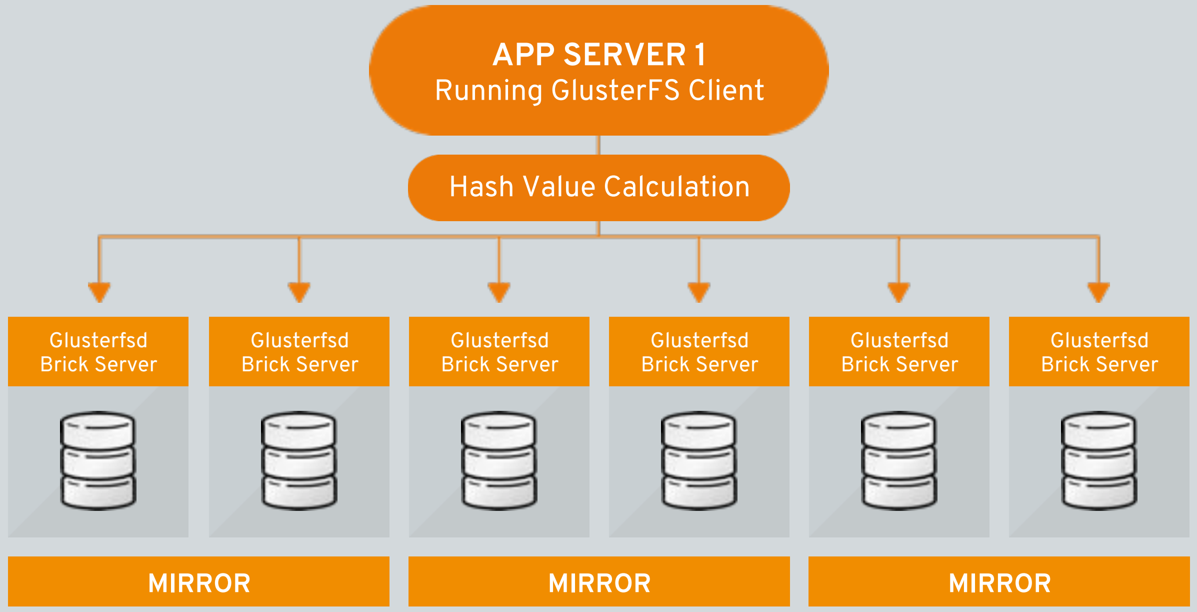
GlusterFS NATIVE CLIENT

- **BASED ON FUSE KERNEL MODULE,** which allows the file system to operate entirely in userspace
- **SPECIFY MOUNT** to any GlusterFS server
- **NATIVE CLIENT** fetches **volfile** from mount server, then communicates directly with all other nodes to access data

- Load inherently balanced across distributed volumes
- Recommended for high concurrency & high write performance

GlusterFS NATIVE CLIENT

Clients talk directly to the data bricks based on elastic hash



NFS

Accessibility from UNIX and Linux systems

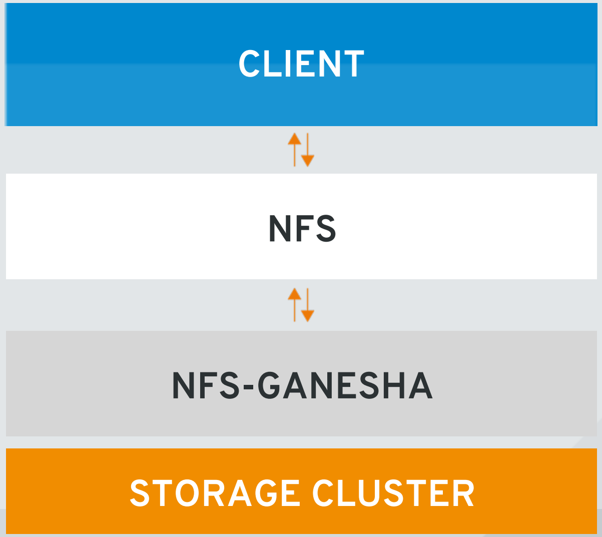
- **STANDARD NFS** connects to NFS Ganesha process on storage node
- **MOUNT GLUSTERFS VOLUME** from any storage node
- **NFS GANESHA** includes network lock manager to synchronize locks
- **LOAD BALANCING** managed externally
- **STANDARD AUTOMOUNTER** is supported.
- **SUPPORTED FEATURES:** ACLs, NFSv4, Kerberos auth

Better performance reading many small files from a single client

Ganesha NFS

Scalable & Secure NFSv4 client support

- **PROVIDES** client access with simplified failover and failback in the case of a node or network failure.
- **INTRODUCES** ACLs for additional security
- **KERBEROS** authentication
- **DYNAMIC** export management.



SMB/CIFS

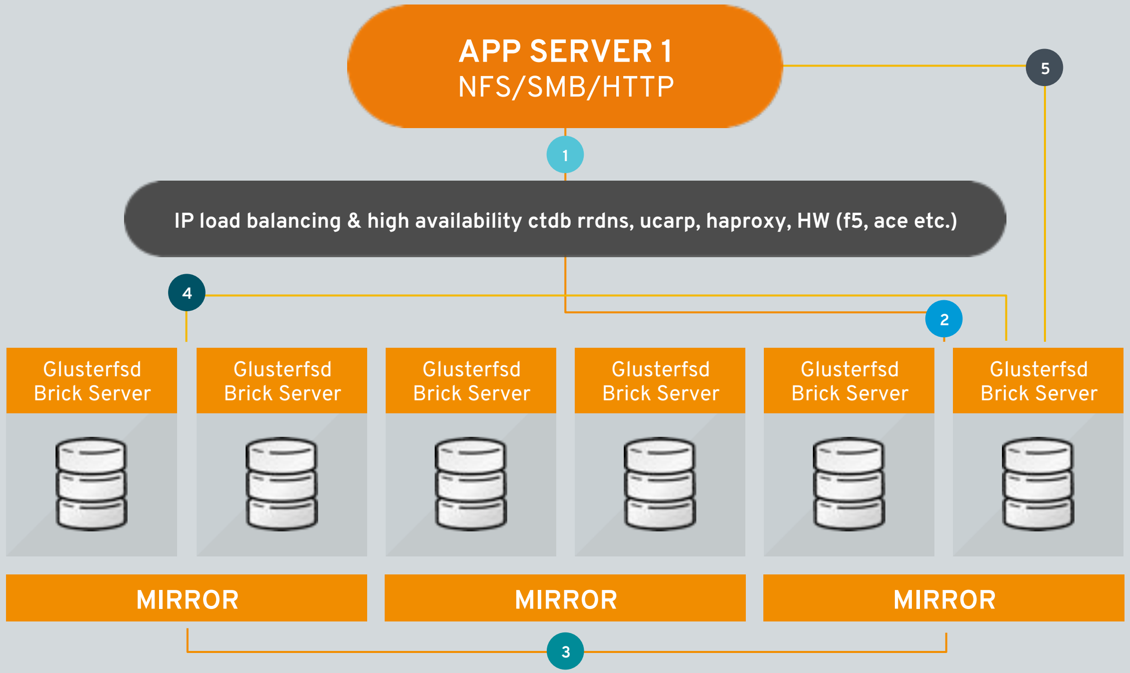
Accessibility from Windows systems

- **STORAGE NODE** uses Samba with winbind to connect with AD
- **SMB CLIENTS** can connect to any storage node running Samba
- **SMB VERSION** 3 supported
- **LOAD BALANCING** managed externally
- **CTDB** is required for Samba clustering

Samba uses RHGS gfapi library to communicate directly with GlusterFS server process without going through FUSE

NFS & CIFS DATA FLOW

Clients talk to the mounted storage node, then directed to the data bricks



OBJECT ACCESS

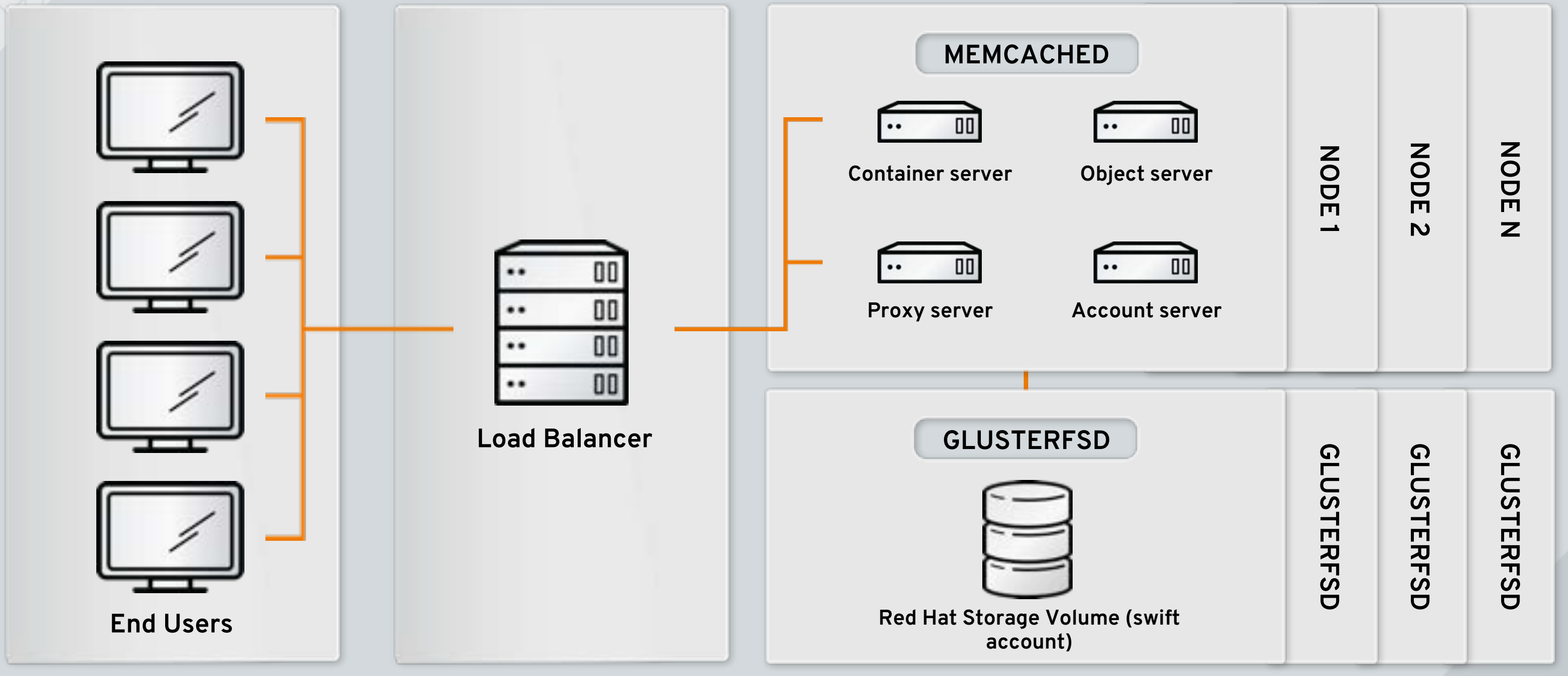
of GlusterFS Volumes

- **BUILT UPON** OpenStack’s Swift object storage system, can also do S3
- **BACK-END FILE SYSTEM** for OpenStack Swift Accounts as GlusterFS volumes
- **STORE AND RETRIEVE** files using the REST interface
- **SUPPORT INTEGRATION** with SWAuth and Keystone authentication service

Implements objects as files and directories
under the container (”Swift/S3 on File”)

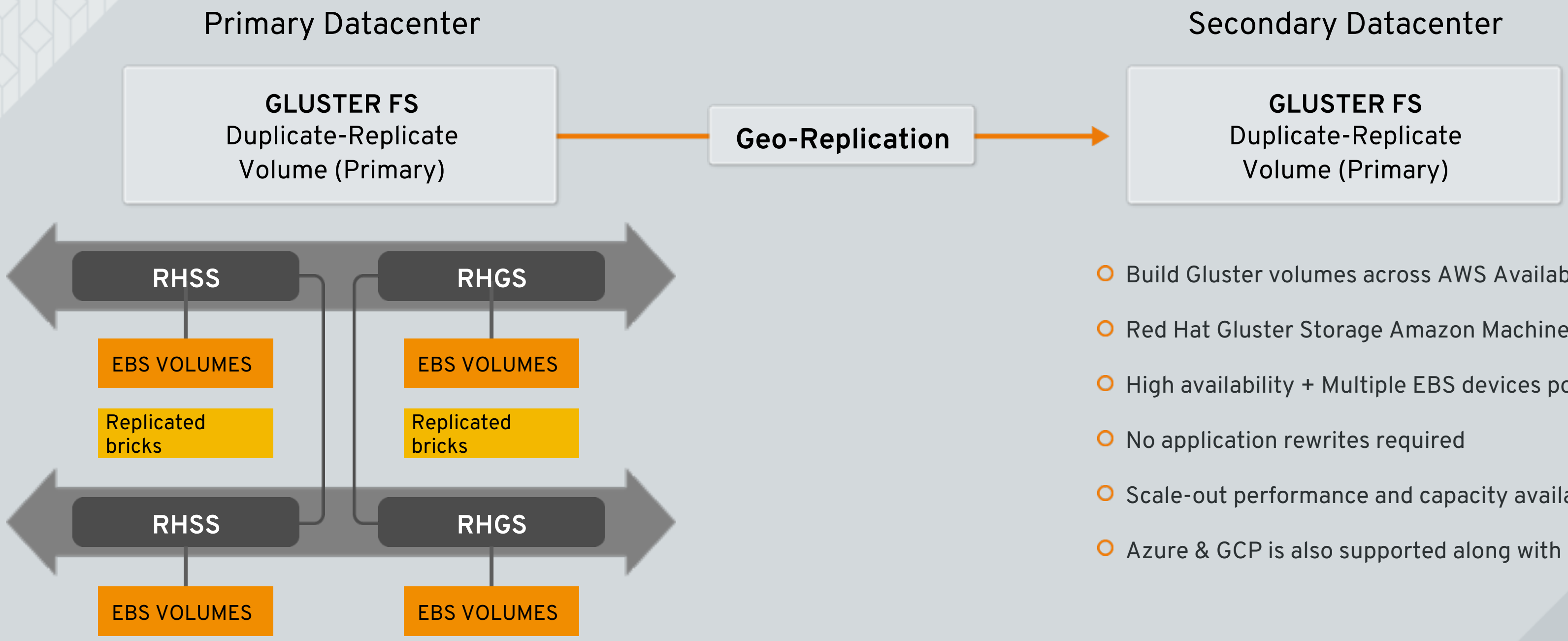
redhat.

OBJECT STORE ARCHITECTURE



DEPLOYMENT

DEPLOYMENT IN PUBLIC CLOUDS

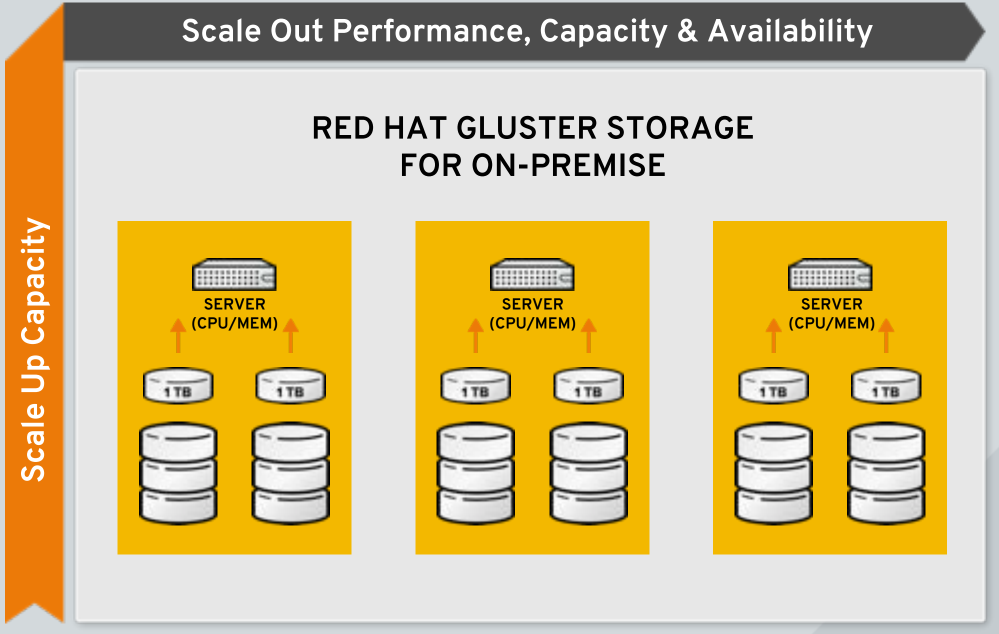


- Build Gluster volumes across AWS Availability Zones
- Red Hat Gluster Storage Amazon Machine Images (AMIs)
- High availability + Multiple EBS devices pooled
- No application rewrites required
- Scale-out performance and capacity availability as needed
- Azure & GCP is also supported along with AWS

CHOICE OF DEPLOYMENT

Single, Global namespace

- Deploys on Red Hat-supported servers and underlying storage: DAS, JBOD
- Scale-out linearly
- Replicates synchronously and asynchronous



Scale Out Performance, Capacity & Availability
Scale Up Capacity
RED HAT GLUSTER STORAGE FOR ON-PREMISE
SERVER (CPU/MEM)
1 TB
SERVER (CPU/MEM)
1 TB
SERVER (CPU/MEM)
1 TB

HOW IS GLUSTER DEPLOYED?

Red Hat Gluster Storage

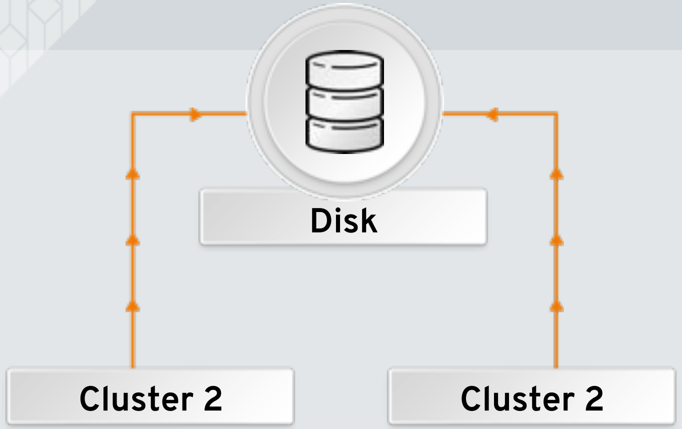
PHYSICAL	VIRTUAL	CONTAINERS	CLOUD
RED HAT® GLUSTER STORAGE	RED HAT® GLUSTER STORAGE	RED HAT® GLUSTER STORAGE	RED HAT® GLUSTER STORAGE
RED HAT® ENTERPRISE LINUX®	RED HAT® ENTERPRISE LINUX® RED HAT® ENTERPRISE VIRTUALIZATION	RED HAT® ENTERPRISE LINUX® ATOMIC HOST OPENSHIFT ENTERPRISE by Red Hat®	RED HAT® ENTERPRISE LINUX® Google Cloud Platform amazon web services™

redhat.

DATA PROTECTION

AVOIDING SPLIT-BRAIN

Server Side Quorum



- **SERVER-SIDE QUORUM** is based on the liveliness of glusterd daemon
- **VOLUME LEVEL** enforcement of quorum
- **NETWORK OUTAGE** breaker switch based on percentage ratio
- **TRIGGERED BY ACTIVE NODES** is more than 50% of the total storage nodes
- **QUORUM ENFORCEMENT** will require an arbitrator in the trusted storage pool

```
# gluster volume set <volname> cluster.server-quorum-type none/server
# gluster volume set all cluster.server-quorum-ratio <percentage%>
```

SERVER-SIDE QUORUM

Scenarios

In a storage pool with 4-nodes (A, B, C and D) in a 2X2 distributed replicated configuration, A and B are replicated and C and D are replicated. The quorum ratio is set to the default value of > 50%

Node A dies, and a write destined for A « » B pair arrives	• Write will happen to B • When A comes back online, self-heal will kick in to fix the discrepancy • No change in this behavior with or without quorum enabled
Node A dies, and a write destined for C « » D pair arrives	• Write will happen to C and D. • No change in this behavior with or without quorum enabled
If both A & B die, a write destined for the A « » B pair arrives	• Quorum is enabled, and the quorum ratio is not met. All the bricks in A, B, C, and D will go down. • Quorum is not enabled. Write will fail, and bricks in C & D will continue to be alive
If both A & B die, a write destined for the C « » D pair arrives	• Quorum is enabled, and the quorum ratio is not met. All the bricks in A, B, C, and D will go down. • Quorum is not enabled. Write to C & D will succeed

CLIENT SIDE QUORUM

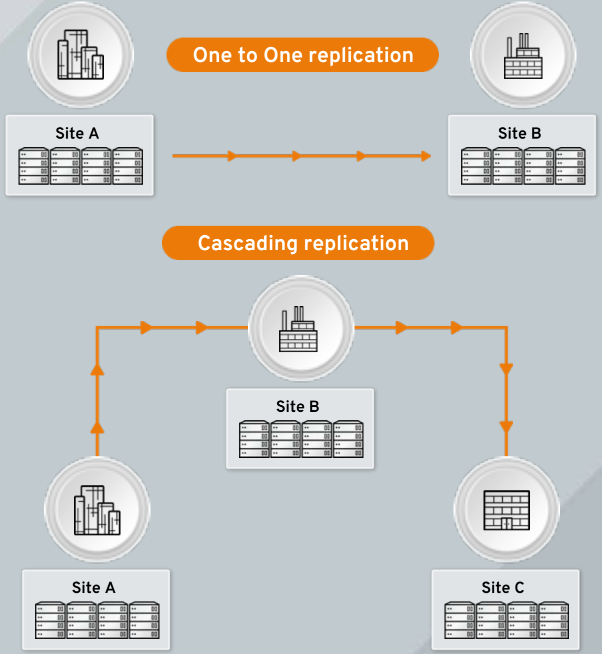
Avoiding Split-Brain

Cluster.quorum-type	Cluster.quorum-type	Behavior
None	Not applicable	Quorum not in effect
Auto	Not applicable	• Allow writes to a file only if more than 50% of the total number of bricks • Exception: For replica count=2, first brick in the pair must be online to allow writes.
Fixed	1 thru replica-count	The minimum number of bricks that must be active in a replica-set to allow writes.

GEO-REPLICATION

Multi-site content distribution

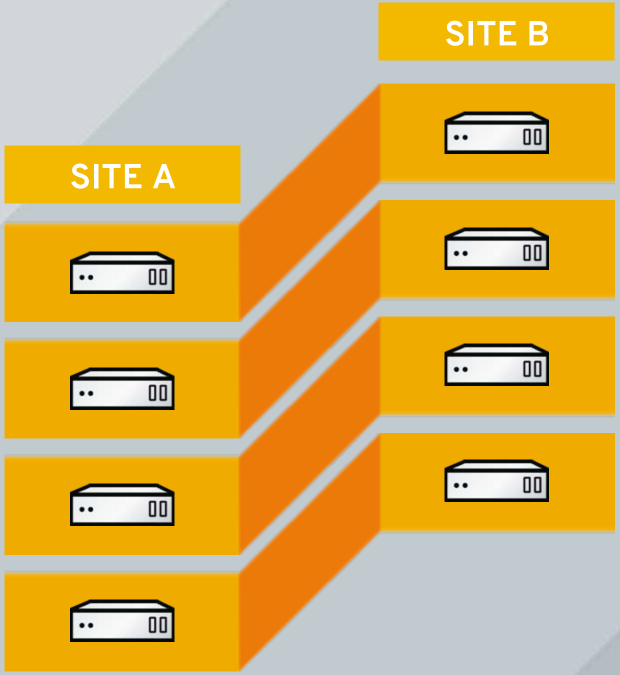
- Asynchronous across LAN, WAN, or Internet
- Master-slave model, cascading possible
- Continuous and incremental
- Multiple configurations
  - One to one
  - One to many
  - Cascading



GEO-REPLICATION

Features

- **PERFORMANCE**
  Parallel transfers
  Efficient source scanning
  Pipelined and batched
  File type/layout agnostic
- **CHECKPOINTS**
- **FAILOVER AND FAILBACK**



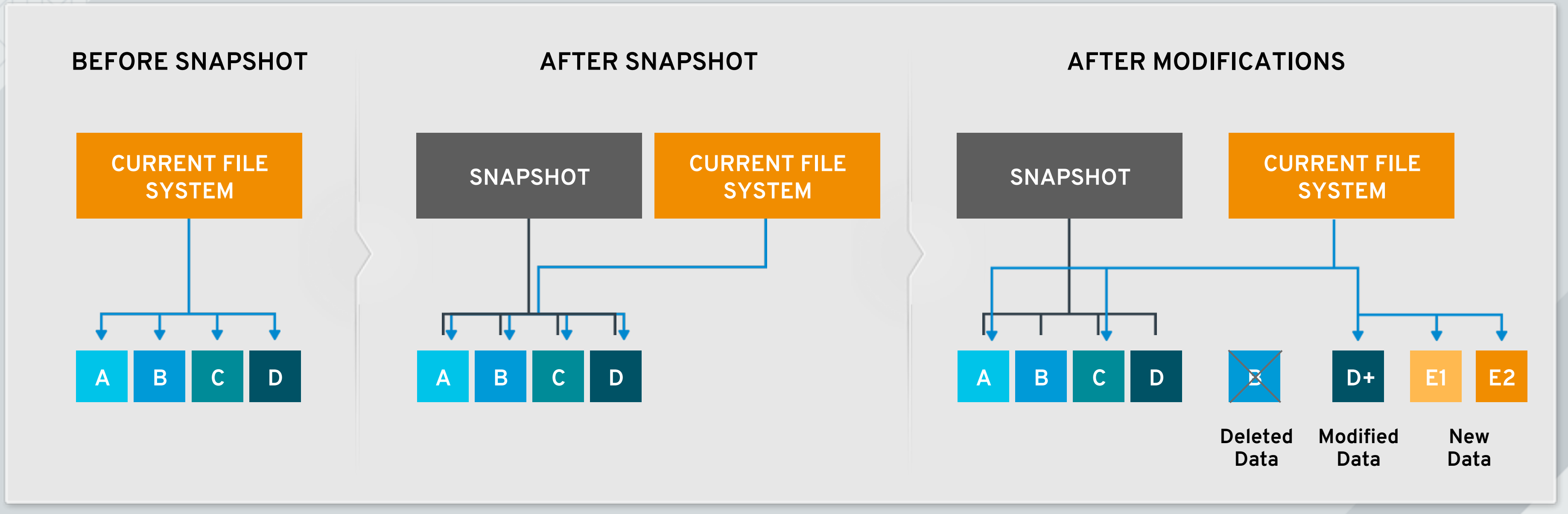
GEO REPLICATION V.S. REPLICATED VOLUMES

Geo-Replication	Replicated Volumes
Mirrors data across geographically distributed trusted storage pools. Provides high-availability.	**Mirrors data** across bricks within one trusted storage pool.
Backups of data for disaster recovery.	**Provides high-availability.**
Asynchronous replication: checks for changes in files. Syncs them on detecting differences.	**Synchronous replication:** each and every file operation is applied to all the bricks
Potential of data loss: minutes/hours	**Potential of data loss:** none

GLUSTER VOLUME SNAPSHOTS

- Point-in-time state of storage system/data
- Volume level, ability to create, list, restore, and delete
- LVM2 based, operates only on thin-provisioned volumes
- Produces Crash Consistent image
- Support a max of 256 snapshots per volume
- Snapshot can be taken on one volume at a time
- Snapshot names need to be cluster-wide unique
- Managed via CLI
- User serviceable snapshots

RED HAT GLUSTER STORAGE



USING SNAPSHOTS

1
```
# gluster snapshot create <snap-name> <volname> [description
 <description>] [force]
```

2
```
# gluster snapshot list [volname]
```

3
```
# gluster  snapshot restore <snapname>
```

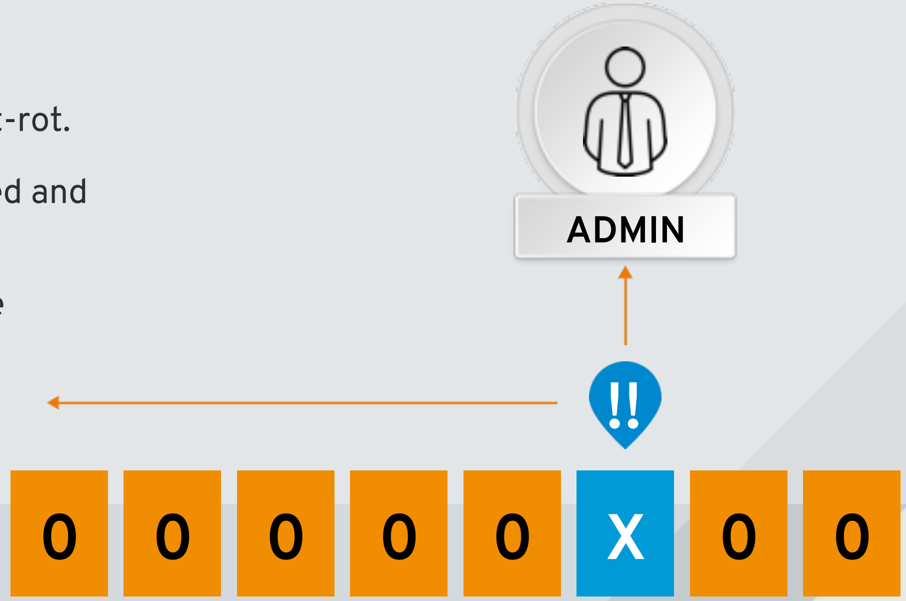
4
```
# gluster  snapshot delete <snapname>
```

5
```
# mount -t glusterfs <hostname>:/snaps/<snapname>/<volname>
 <mountdir>
```

BIT ROT DETECTING

Detection of silent data corruption

- **RED HAT GLUSTER STORAGE 3.1** provides a mechanism to scan data periodically and detect bit-rot.
- **CHECKSUMS** are computed when files are accessed and compared against previously stored values.
- **IF THEY DO NOT MATCH** an error is logged for the storage admin.



MANAGEMENT AND MONITORING

(Erasure Coding, Tiering, Bit-rot Detection and NFS-Ganesha)

DISK UTILIZATION AND CAPACITY MANAGEMENT

Quota

- **CONTROL THE DISK UTILIZATION** at both a directory and volume level
- **TWO LEVELS** of quota limits: Soft and hard
- **WARNING MESSAGES** issued on reaching soft quota limit
- **WRITE FAILURES** with EDQUOTA message after hard limit is reached
- **HARD AND SOFT** quota timeouts
- **THE DEFAULT SOFT LIMIT** is an attribute of the volume that is a percentage

JOBS IN THE QUOTA SYSTEM

Accounting

- **MARKER TRANSLATOR** loaded on each brick of the volume
- **ACCOUNTING** happens in the background
- **UPDATE** is sent upwards up to the root of the volume

JOBS IN THE QUOTA SYSTEM

Enforcement

The enforcer updates its 'view' of directory's disk usage on the incidence of a file operation

Enforcer uses quotad to get the aggregated disk usage of a directory

JOBS IN THE QUOTA SYSTEM

Aggregator (quotad)

- **QUOTAD IS A DAEMON** that serves volume-wide disk usage of a directory
- **QUOTAD IS PRESENT** on all nodes in the cluster
- **ONE QUOTAD** per node
- **QUOTAD MANAGES** all the volumes on which quota is enabled

MONITORING STORAGE USING NAGIOS

- **BASED ON NAGIOS** open IT infrastructure monitoring framework
- **MONITOR LOGICAL ENTITIES:** Cluster, volume, brick, node
- **MONITOR PHYSICAL ENTITIES:** CPU, disk, network
- **ALERTING VIA SNMP** when critical components fails
- **REPORTING:** Historical record of outages, events, notifications
- **INTERFACE WITH NAGIOS** Web Console and/or dashboard view

MONITORING SCENARIOS

SCENARIO 1

User has no existing monitoring infrastructure in place or does not use Nagios

SCENARIO 2

User already has Nagios infrastructure in place, use plugins only

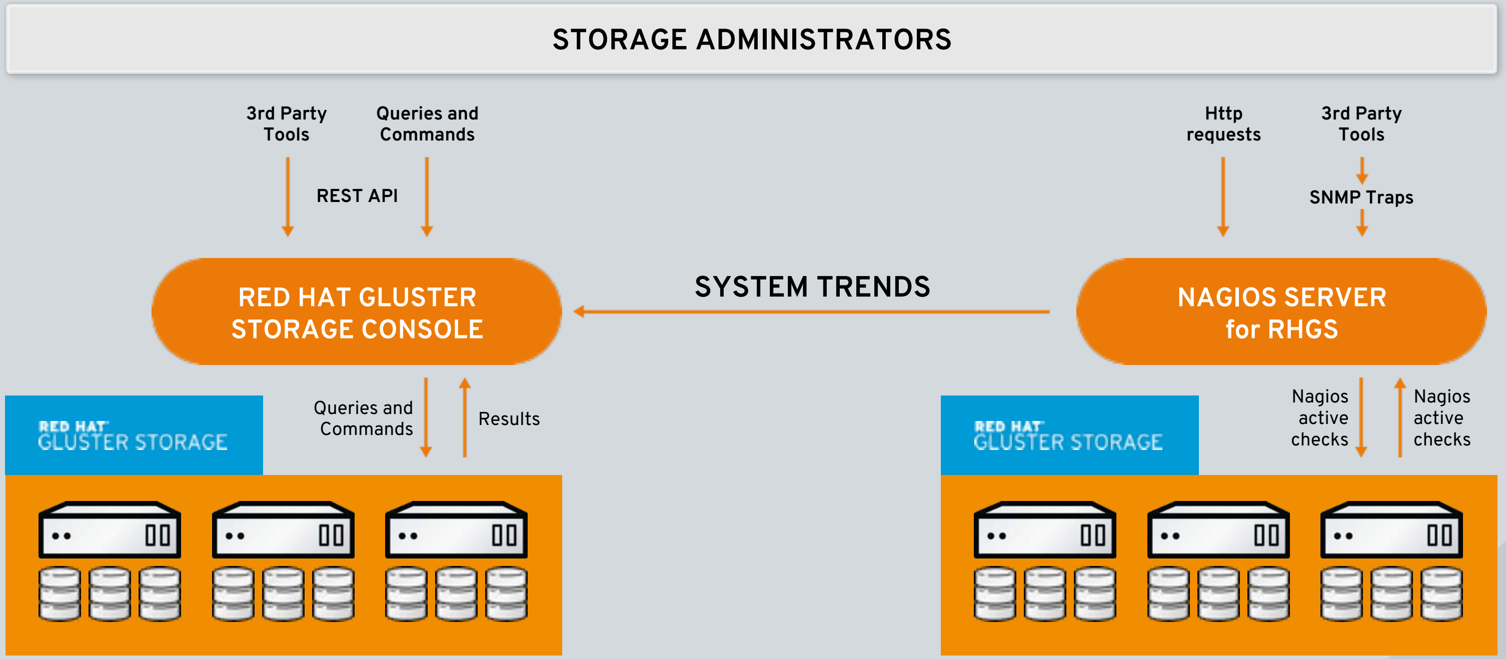
SCENARIO 3

Usage in conjunction with Red Hat Gluster Storage console

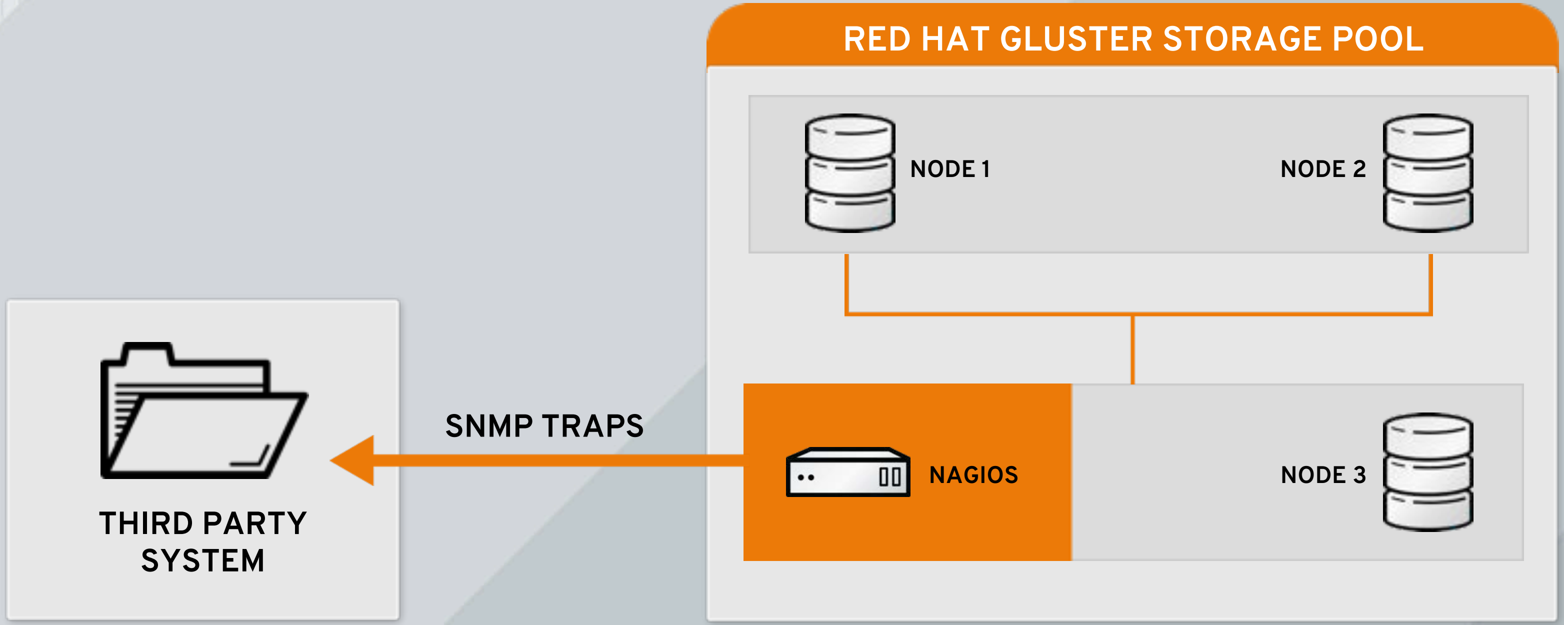
Support SNMP traps for all scenarios

FUNCTIONAL ARCHITECTURE

of Red Hat Gluster Storage monitoring



NAGIOS DEPLOYED ON RED HAT GLUSTER NODE



SIMPLIFIED AND UNIFIED STORAGE MANAGEMENT

Single view for converged storage and compute

Storage operations

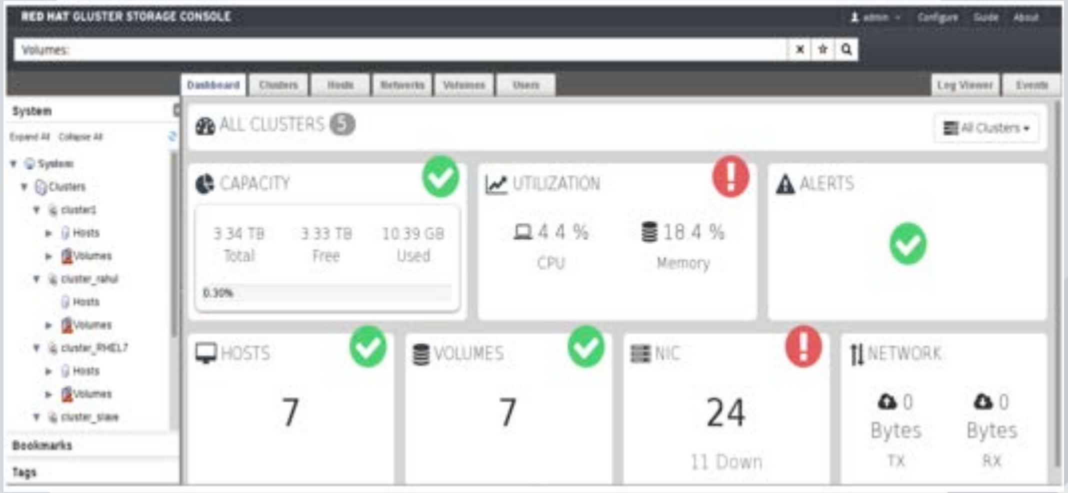
- Intuitive user interface
- Volume management
- On-premise and public cloud

Virtualization and storage

- Shared management with Red Hat Enterprise Virtualization Manager

Provisioning

- Installation and configuration
- Update management
- Lifecycle management
  Familiar Red Hat Enterprise Linux tools



SECURITY

Network Encryption at Rest and In Transit

- **SUPPORTS** network encryption using TLS/SSL for authentication and authorization, in place of the home grown authentication framework used for normal connections
- **SUPPORT** encryption in transit and transparent encryption (at rest)
- **TWO TYPES OF ENCRYPTION:**
  - I/O encryption - encryption of the I/O connections between the Red Hat Gluster Storage clients and servers
  - Management encryption - encryption of the management (glusterd) connections within a trusted storage pool

THANK YOU

redhat.