Persistent Storage Management for Docker containers on Red Hat Enterprise Linux Atomic Host Platform

Mukesh Bafna
Sr. Principal Software Engineer, Veritas Technologies
May 2, 2017
Containers 101

Overhead

Diagram showing the comparison between Hypervisor-based Virtualization and Container virtualization.
Application Deployment

Monolithic Apps on Physical Servers

VM’s Abstraction

Stateless & Horizontal Scalable Apps

Micro-services & Containers

#redhat #rhsummit
RED HAT ENTERPRISE LINUX ATOMIC HOST

**IT IS RED HAT ENTERPRISE LINUX**

- Inherits the complete hardware ecosystem, military-grade security, stability and reliability for which Red Hat Enterprise Linux is known.

**OPTIMIZED FOR CONTAINERS**

- **MINIMIZED FOOTPRINT**
  - Minimized host environment tuned for running Linux containers while maintaining compatibility with Red Hat Enterprise Linux.

- **SIMPLIFIED MAINTENANCE**
  - Atomic updating and rollback means it's easy to deploy, update, and rollback using imaged-based technology.

- **ORCHESTRATION AT SCALE**
  - Build composite applications by orchestrating multiple containers as microservices across multiple hosts.
Persistent storage: a key challenge for containers

**Container Journal**

"Stateful container apps represent the next big IT challenge"

**Gartner**

"Stateful Database applications such as Redis, MySQL, MongoDB among most pulled images on Docker Hub"

**NewStack research.**

"Persistent storage among top issues for container enterprise-readiness in production"
Persistent Storage Types in Docker
Default

No data persistence
Docker Volume

HOST

- /app:/app

Container Persistence
Host Persistence
Docker Volume Plugin Architecture

Docker Client • Basic volume create and delete operations.

Docker Daemon • Vendor-specific volume snapshot and copy operations

Plugin Client • Vendor-supported backend drivers

Plugin Daemon

Storage Backend 1

Storage Backend 2

Storage Backend 3

Storage Backend...
Kubernetes Persistent Volumes

- Similar to Docker volume plugins

- Types of persistent volumes –
  - GCEPersistentDisk
  - AWSElasticBlockStore
  - AzureFile
  - AzureDisk
  - FC (Fibre Channel)
  - Flocker
  - NFS
  - iSCSI
  - RBD (Ceph Block Device)
  - CephFS
  - Cinder (OpenStack block storage)
  - Glusterfs
  - VsphereVolume
  - Quobyte Volumes
  - HostPath (single node testing only)
  - VMware Photon
  - Portworx Volumes
  - ScaleIO Volumes
Enterprise Storage
Key Capabilities
Enterprise Storage Key Capabilities

- Persistent Storage Management
- Scale-Out
- Quality of Service
- Snapshots
Enterprise Storage Key Capabilities

- Policy Driven
- I/O Acceleration
- Zero Impact Backup
- Cloud Mobility
Enterprise Storage Key Capabilities

- Deduplication
- Encryption
- Ecosystem Support
- GUI
Docker Volume Plugin Storage Vendors

- Blockbridge
- NetApp
- Nutanix
- ClusterHQ
- VERITAS
- portworx
- HEDVIG
- ceph
- nimble storage
- ClusterFS
- Azure
About Veritas

![Diagram showing Insight, Protection, Availability, and 360 Data Management]

- Insight
- Protection
- Availability
- 360 Data Management
About Veritas

- Data
- Unstructured Data Visibility
- Information Map
- Enterprise Vault & EV.cloud
- Archiving & Retention
- NetBackup & Backup Exec
- NetBackup Appliance
- Velocity
- Integrated PBBA
- Copy Data Management
- Distributed File System
- InfoScale Availability
- InfoScale Storage
- HyperScale for OpenStack and HyperScale for Containers
- Resiliency Platform
- SDS for OpenStack and containers
- SDS for Unstructured Data (Big Data, IOT, Video)
- SDS for High Performance Workloads
- Data Insight
- Veritas Access
- eDiscovery Platform
- Legal Disposition
- File Classification
- Backup & Recovery
- Business Resiliency/DR
- 360 Data Management
- About Veritas
HyperScale for Containers

Container Orchestration Framework

- Docker
- HyperScale
- Compute Nodes (primary)
- Network switch
- Periodic sync
- Full Copy
- Data Management Nodes

Zero Impact Backup

Ship to Cloud

#redhat #rhsummit
HyperScale for Containers Deployment

Lower CAPEX and OPEX
Files ➤ Red Hat Summit... ➤ HyperScale

<table>
<thead>
<tr>
<th>Name</th>
<th>Modified</th>
<th>Modified By</th>
<th>File Size</th>
<th>Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFC - Docker Service Create with SQL IO Load.mp4</td>
<td>4 minutes ago</td>
<td>Mukesh Bafna</td>
<td>5.58 MB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - Docker Service Create with SQL IO Load.srt</td>
<td></td>
<td>Mukesh Bafna</td>
<td>1.64 KB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - Installation.mp4</td>
<td></td>
<td>Mukesh Bafna</td>
<td>4.08 MB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - Installation.srt</td>
<td></td>
<td>Mukesh Bafna</td>
<td>1013 bytes</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - Introduction.mp4</td>
<td></td>
<td>Mukesh Bafna</td>
<td>5.40 MB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - QOS.mp4</td>
<td></td>
<td>Mukesh Bafna</td>
<td>2.89 MB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - QOS.srt</td>
<td></td>
<td>Mukesh Bafna</td>
<td>1.02 KB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - Volume-From-Snapshot.mp4</td>
<td>3 minutes ago</td>
<td>Mukesh Bafna</td>
<td>12.0 KB</td>
<td>Shared</td>
</tr>
<tr>
<td>HFC - volume-From-Snapshot.srt</td>
<td>4 minutes ago</td>
<td>Mukesh Bafna</td>
<td>520 bytes</td>
<td>Shared</td>
</tr>
<tr>
<td>rhsummit 2017 Veritas v1.pptx</td>
<td>A few seconds ago</td>
<td>Mukesh Bafna</td>
<td>14.6 MB</td>
<td>Shared</td>
</tr>
<tr>
<td>rhsummit Booth Deck v1.pptx</td>
<td>About a minute ago</td>
<td>Mukesh Bafna</td>
<td>982 KB</td>
<td>Shared</td>
</tr>
</tbody>
</table>

Visit us at Veritas Booth
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos

facebook.com/redhatinc
twitter.com/RedHatNews
Backup Slides
Simplified Management & API Driven
## Data Protection and Resiliency

A storage policy guarantees specific levels of performance, availability, and protection. When a policy is assigned to a volume, the volume is provisioned from a specific data store that can satisfy these storage requirements.

### Storage Policies (3)

<table>
<thead>
<tr>
<th>Policy</th>
<th>Performance</th>
<th>Availability</th>
<th>Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GOLD</strong></td>
<td>IOPS Minimum 300, Maximum 30000</td>
<td>Resiliency Factor 2</td>
<td>No. of Snapshots 3, Snapshot Frequency 5 mins</td>
</tr>
<tr>
<td></td>
<td>Latency Minimum 1, Maximum 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SILVER</strong></td>
<td>Performance</td>
<td>Availability</td>
<td>Protection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BRONZE</strong></td>
<td>Performance</td>
<td>Availability</td>
<td>Protection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

#redhat #rhsummit
Predictable Performance via Policies

A storage policy guarantees specific levels of performance, availability, and protection. When a policy is assigned to a volume, the volume is provisioned from a specific data store that can satisfy these storage requirements.

**BRONZE**
- Performance
  - IOPS Minimum 100
  - Maximum 10000
- Availability
  - Resilience Factor 0
- Protection
  - No. of Snapshots 3
  - Snapshot Frequency 5 mins

**GOLD**
- Performance
- Availability
- Protection

**SILVER**
- Performance
- Availability
- Protection

**TEST_POLICY**
- Performance
- Availability
- Protection
LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.