DATAVISION
by soda

Longhorn and the Emergence of Next - Gen Software Defined Storage Technologies

Juan Herrera Utande - SUSE

DATAVISION
by soda

BIO

Juan Herrera Utande

- Head of Technical Marketing team at SUSE
- Background in Consulting and Solutions Architecture at SUSE and Red Hat, extensive experience in Kubernetes, OpenStack, and Software Defined Storage.

Me at the beautiful port of Bermeo near Bilbao ☺

DATAVISION by SODA

Agenda

- Longhorn: Why another SDS?
- Built on Kubernetes, Run for Kubernetes
- Architecture
- Agility, Run Anywhere
- Comprehensive capabilities
- Roadmap

https://metrics.longhorn.io/d/tsrvoC-mz/longhorn?orgId=2&from=now-2y&to=now

Longhorn: Why another SDS?

Guiding principles for Longhorn creation and evolution

- 100% open source, enterprise-grade cloud-native storage
- True community driven project with clear steering model (CNCF hosted)
- Built from scratch on Kubernetes for Kubernetes
- Easy to adopt by Dev and ITOps (GitOps ready)
- Easy to deploy and operate
- Support for block, file and object storage.
- Both community and enterprise support
- Built in security and encryption
- Built in resilience through replication, snapshots and backups
- Fast recovery through definable RPO and RTO
- Go beyond existing solutions (e.g., Multi-AZ block in Cloud)

RANCHER BY SUSE

LONGHORN

DATAVISION by soda

Built on Kubernetes, Run for Kubernetes

What does Kubernetes provide?

- Longhorn relies on all key Kubernetes features: HA, desired state management, seamless updates, ...
- Longhorn is K8s API first: managed through manipulation of CRDs, configurations and status
- CLI and UI always go through API, no sideways nor backdoors to access Longhorn features
- All existing K8s knowledge and tooling can be reused: K8s API, kubectl CLI, GitOps, ...
- No extra Database required to manage Longhorn state
- Kubernetes existing security layers and RBAC access is applied to all Longhorn CRDs

How does it work behind the scenes?

- Longhorn implements distributed storage using containers and microservices, following K8s resource-controller pattern.
- It creates a dedicated storage controller for each block device volume and synchronously replicates the volume across multiple replicas stored on multiple nodes.
- The storage controller and replicas controllers are themselves orchestrated using K8s.
- Native K8s Longhorn controllers will: reconcile resource configuration and status changes, manage and monitor Longhorn components dynamically, respect resource status change for resilience

RANCHER BY SUSE

LONGHORN

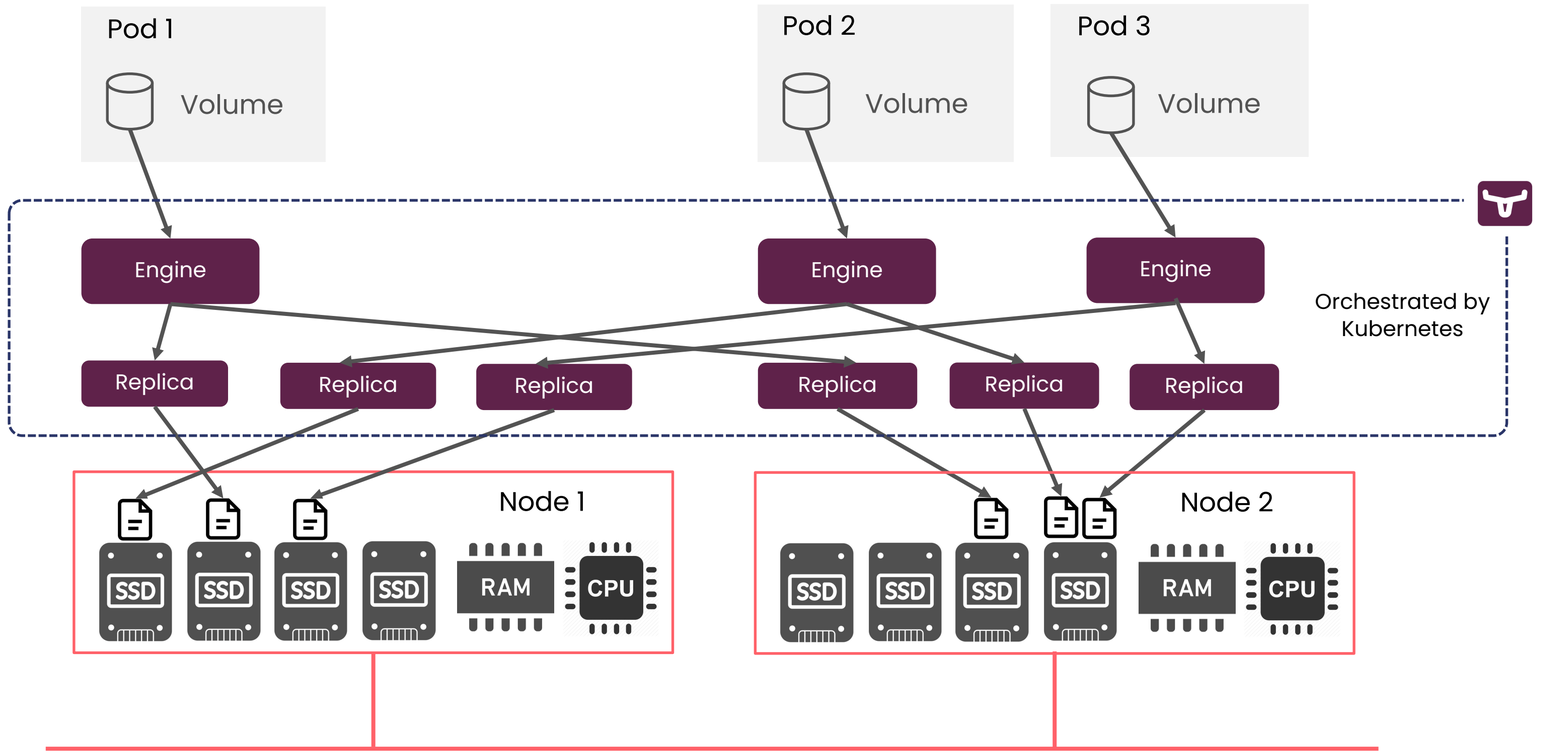
Architecture

Control Plane
- K8s Controller + CR

Data Plane
- Volume Frontend (iSCSI)
- Volume (Engine)
- Volume replica (Replica)

Volume Lifecycle
- CSI
- PVC/PV

Data Placement
- Longhorn disk (FS on host)



Agility, Run Anywhere

On-Prem, Cloud and Edge

Distro
- Any certified Kubernetes

Architectures
- AMD64
- ARM64
- s390x

Install and upgrade
- Helm chart, Rancher catalog
- Kubernetes upgrade awareness

On-Prem
- Physical nodes
- Virtual nodes
- Harvester HCI

Cloud
- AKS, EKS, GKE,...
- Cluster Autoscaler
- Node Pool Upgrade

Edge
- Resource constrained
- SLE Micro
- Low resource footprint

Comprehensive Capabilities

Persistent volume lifecycle & data management in and outside cluster

Kubernetes PV Support

- Block, FS, Object Storage Volume
- RWO, RWX

Kubernetes CSI Protocol Support

- Volume Provision, Attachment, Snapshot, Clone, Expansion

Volume Capabilities

- Thin provisioning
- Snapshot, CoW
- Trim
- Live upgrade

IO Performance

- Data Locality, Local Volume
- SPDK Data Engine

Storage Topology

- Node & Zone replica scheduling anti-affinity
- Storage tag

Data Protection

- Data replication
- Data encryption in transit and at rest
- Bit-rot protection

Data Services

- In-cluster/out-of-cluster backup & restore
- Disaster recovery

Data Reduction

- Space efficiency for Snapshot
- Backup compression

Longhorn Roadmap – Latest release 1.5.0

What's new in 1.5.0*

- v2 Data Engine based on SPDK – Tech Preview
- New Longhorn VolumeAttachment CR
- Cluster Autoscaler - GA
- Volume Backup Compression Methods
- RWX Volume Trim & Trim recurring jobs
- Upgrade Path Enforcement & Downgrade Prevention
- Additional backup storage protocols, including CIFS and Azure
- Snapshot Cleanup & Delete Recurring Job
- Backing Image Management via CSI VolumeSnapshot

* https://github.com/longhorn/longhorn/releases/tag/v1.5.0

Longhorn Roadmap – Latest release 1.5.0

Two new key features

Longhorn Engine v2 – based on SPDK

- Based on Storage Performance Development Kit
- Next-gen Data Engine
- Utilizing NVMe-oF/NVMe-over-TCP protocol
- Use entire disk/partition instead of working on top of filesystem
- Near native performance with NVMe disks in Local Volume Passthrough mode
- Average latency reduced by half
- 3 times more IOPS in all scenarios

Object Store support via s3gw

- Based on Ceph's RADOSGW (RGW)
- Highly available
- Management UI
- Helm based install
- Integrated in Rancher Apps Catalog
- Works with any PVC backend

Longhorn Roadmap - Future

1.6.0 & 1.7.0

Longhorn v1.6.0 (Q4 2023)

- Experimental: Longhorn Engine v2 with feature parity to Engine v1
- Experimental: Object store support
- Local Volume Passthrough
- Volume Group

Longhorn v1.7.0 (Q2 2024)

- GA: SPDK Data Engine
- GA: Object store support.
  - Zero downtime upgrade
  - HA deployment
- Experimental: Replica Sharding

THANK YOU!

LONGHORN

Resources

http://bit.ly/longhorn-soda-data-vision-2023