Rook EdgeFS

Kubernetes Native Decentralized Data Fabric

Ilya Grafutko, Director Eng, DDN

https://rook.io/
https://github.com/Nexenta/edgefs

Rook EdgeFS

- **Deployed as Kubernetes Operator**
  - Full service life-cycle management:
    - install, update, rolling-upgrade, uninstall, re-install
  - Monitoring with Prometheus and Grafana
  - Easy of use built-in GUI with CRD Wizard!
  - Runs in embedded environments: < 1GB DRAM, 2 CPU cores!
  - Runs in the clouds, on-prem and at edge frontiers
- **Consumes locally connected raw disks, directories, kv stores, cloud resources**
- **Provides globally available data protocols**
  - Low latency S3 Object and S3X NoSQL Database
  - Scale-Out High-Performance NFS
  - Scale-Out iSCSI Block Devices

EdgeFS just like Git but for Data Fabric!

- **Git like architecture**
  - Reference from Git documentation:
    “””All object primitives are referenced by a SHA, a 40-digit object identity, which has the following properties:
    - **If two objects are identical they will have the same SHA.**
    - **if two objects are different they will have different SHAs.**
    - If an object was only copied partially or another form of data corruption occurred, **recalculating the SHA of the current object will identify such corruption.** ”””
  - EdgeFS objects (any) always cryptographically self-validated, and therefore globally unique
  - Same as in Git, any modification (or stream of modifications) are fully versioned
- **It is Decentralized Data Fabric for Edge/IoT Computing and Multi-Cloud**

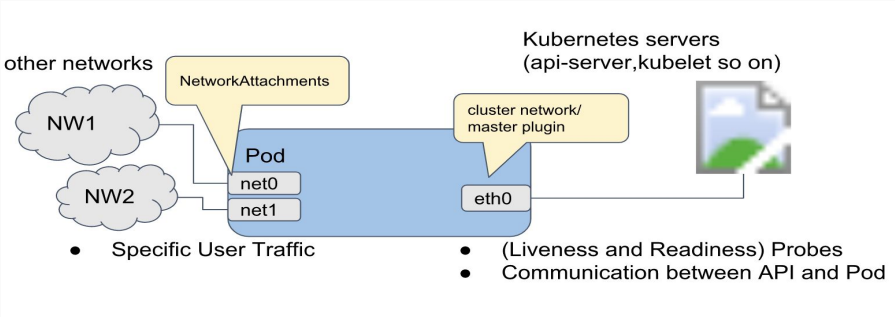
New Features in 1.1

- **Rook EdgeFS CRDs graduated to stable!**
  - install, update, rolling-upgrade, uninstall, re-install
- **Support for Multi-Homed network isolation**
  - enables frontend vs. backend networking isolation
  - improved performance and security
- **Support for device full name path spec**
  - i.e. /dev/disk/by-id/NAME instead of /dev/sdc
  - consistent device naming across reboots

EdgeFS Multi-Homed Network

Benefits of using isolated Multi-Homed Network for Rook EdgeFS:

- Improved Performance characteristics
- Improved Data Security
- Improved QoS and SLA



Rook EdgeFS Cluster CRD:

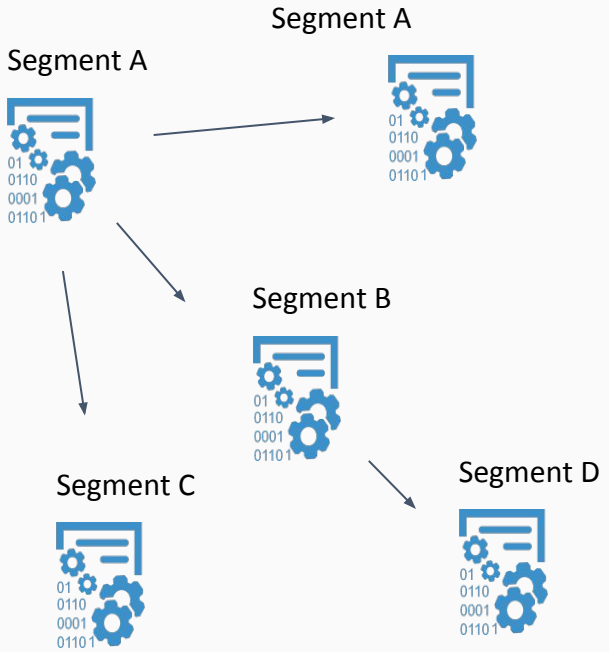
```yaml
apiVersion: edgefs.rook.io/v1beta1
kind: Cluster
metadata:
  name: rook-edgefs
  namespace: rook-edgefs
spec:
  edgefsImageName: edgefs/edgefs:1.2.0
  serviceAccount: rook-edgefs-cluster
  dataDirHostPath: /var/lib/edgefs
  network:
    provider: "multus"
    serverIfName: "flannel2@rep0"
  storage:
    useAllNodes: true
    useAllDevices: true
    config:
      hddReadAhead: "1024"
      lmdbPageSize: "32768"
      useMetadataOffload: "true"
      useMetadataMask: "0xff"
  resources:
    limits:
      memory: "16Gi"
    requests:
      memory: "16Gi"
```

Upcoming features in 1.2

- **Bucket snapshots**
  - eventual snapshot will replicate over connected segments
  - capable of snapshotting buckets with billions of objects
  - clone bucket at any time, at any connected segment
- **Support for KV-SSD backend**
  - works with Samsung KV SSDs
  - support for any KV capable backend
- **Support for hybrid raw disk (e.g. EBS) + S3 (data chunks)**
  - AWS S3 as a storage for chunks > 128KB (configurable)

Bucket snapshots

- **Instantaneous**
  - same segment, immediate
  - other segments, eventual
- **Scales to billions of objects**
  - not dependented on # of objects
- **Integrated with Data Flow Topologies**
  - ISGW links will replicate info
  - supports star and chaining data flow topologies
- **Operations**
  - Snapshot and Clone at any geo-segment
  - any object types: S3, S3X DB, NFS, iSCSI

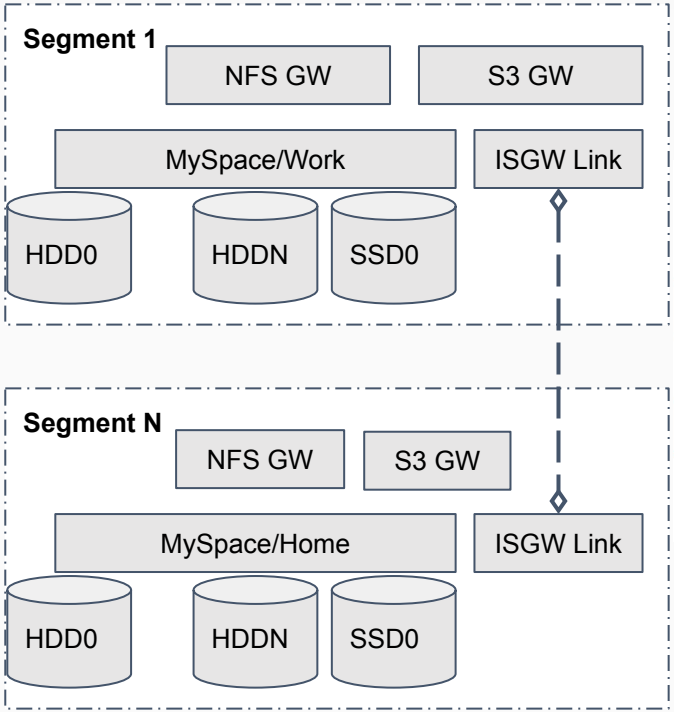


New deployment options

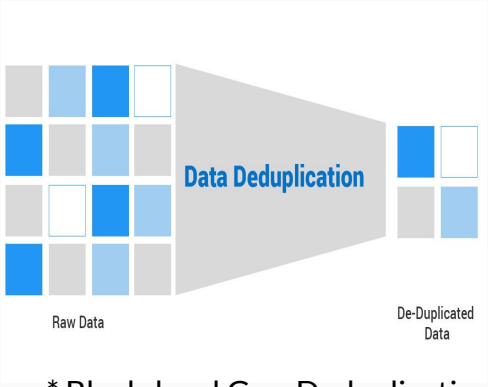
- **How EdgeFS can be deployed as of 1.1?**
  - any Kubernetes PV/PVC
  - any raw block device
  - any hybrid mix of raw HDD and SSD/NVMe
- **We added new ways to deploy in 1.2**
  - emulated Key-Value backends, e.g. RocksDB
  - hardware offloaded Key-Value backends, e.g. Samsung KV-SSD
  - tiered AWS EBS for metadata and AWS S3 for data chunks

EdgeFS Connects Data Segments!

- Global Deduplication
- Clouds Connectivity
- Geo-Transparency
- Global Namespace
- Protocol Transparency
- S3, S3X, NFS, iSCSI



Up to 50x bandwidth savings!



* Block-level Geo-Deduplication
* Metadata-Only Transfers
* Local Caching
* Intelligent Prefetching

EdgeFS Connects Clouds!

- Presents S3 everywhere, transparently syncing regions!
- Metadata-Only synchronization with local caching and prefetching
- Globally de-duplicates cross-cloud or cross-region calls.
- Operates with unmodified, native format objects
- Supported object storage connectors
  - AWS S3
  - GCP GS
  - Azure Blob
  - Alibaba OSS (Coming soon)

EdgeFS Multi-Cloud Layer

- **Geo-Scalability**
  - Spans a network of geographically distributed sites, connected as one global namespace
  - Git-like architecture with fault-tolerance and immutable, versioned metadata design
  - Scales equally well for Object, File, Built-In NoSQL or Block devices
- **Geo-Transparency**
  - Always ON, bi-directional access to same S3 bucket, NFS export
  - Automatic “Last-Writer-Wins” update strategy for S3, Snapview Groups for NFS/iSCSI
- **Geo-Consistency**
  - Snapview groups “floating” within connected geo-namespace
  - Any granularity of protection: Files, Directories, Buckets, LUNs, NoSQL Databases
- **Geo-Locality and Active Caching**
  - Metadata is always replicated, data is prefetched on-demand and cached locally
  - Modifications synchronized asynchronously, thus geographically eventual

EdgeFS Use Cases

- **Multi-Cloud CDN workflow**
  - Efficient distribution of content with advanced local caching features
  - Avoid full replication, with optional pin/unpin/clone of locally cached content
  - Synchronization of primary source content on AWS, Azure, Alibaba, GCP and others
- **Cloud High-Availability**
  - Automatic failover of failed cloud links to redundant dataset in a different region
  - Operate in offline mode for up to 7 days, synchronizing eventually
- **Edge/IoT and Cloud**
  - Capture edge data in local cache and private clouds for AI/ML processing
  - Improve link utilization by sending de-duplicated data asynchronously
  - Access global namespace transparently, while avoiding the need to do full replication
- **Kubernetes Persistent Volumes across clouds**
  - Bi-directional PVs with geo-transparent synchronization across regions
  - CSI managed File or Block PVs, Consistency groups
  - Data segmentation and region awareness

Thank you!

https://github.com/Nexenta/edgefs

https://rook.io/