SDC 19

September 23-26, 2019
Santa Clara, CA

Ensuring Application Data Locality in Kubernetes

David Hay
LINBIT

The Importance of Data Locality

- Application performance
  - Data access throughput and latency
  - Schedulability
- Infrastructure design
  - Data storage capability
  - Composability

2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

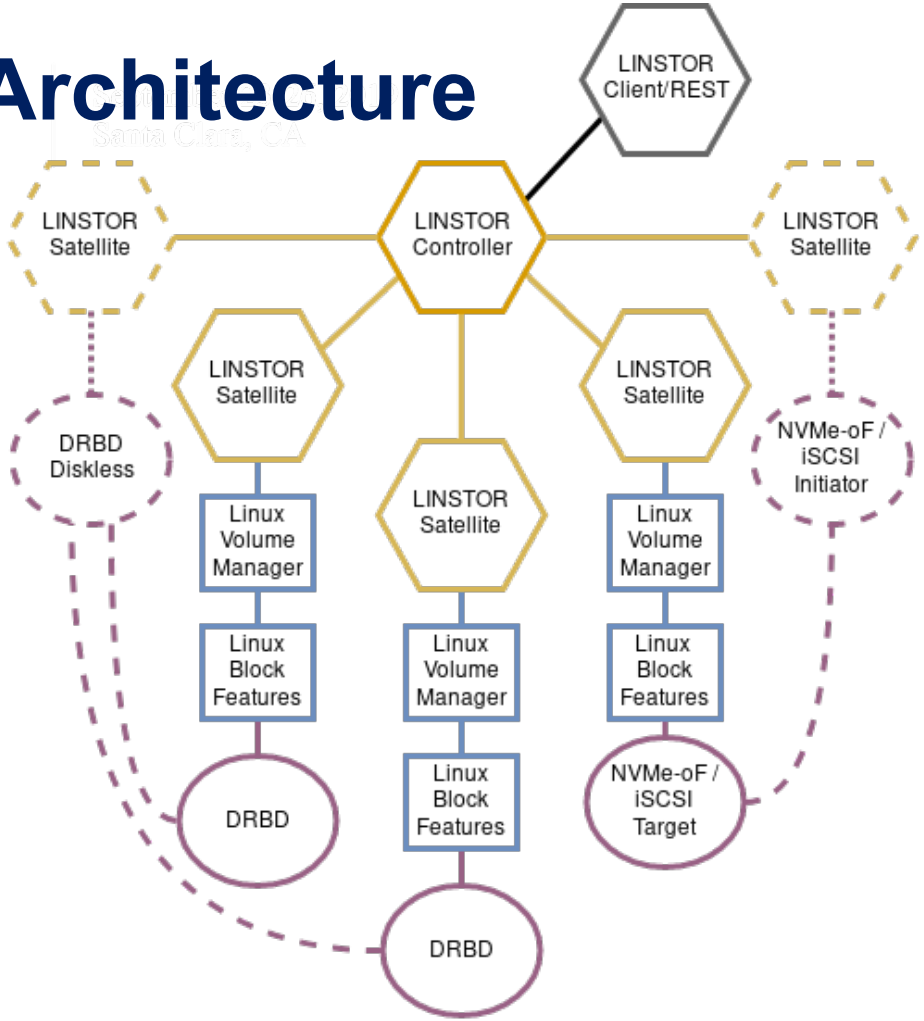
Orchestrating Storage

- LINSTOR is an open source SDS
  - Controls native Linux storage systems
  - Separate control plane accessible via REST
  - Provisions local and/or remote storage
  - Close to zero overhead
- Integration with Kubernetes is simple
  - CSI plugin
  - Typical PVC/PV controls
  - Helm chart and operator

2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

LINSTOR Architecture

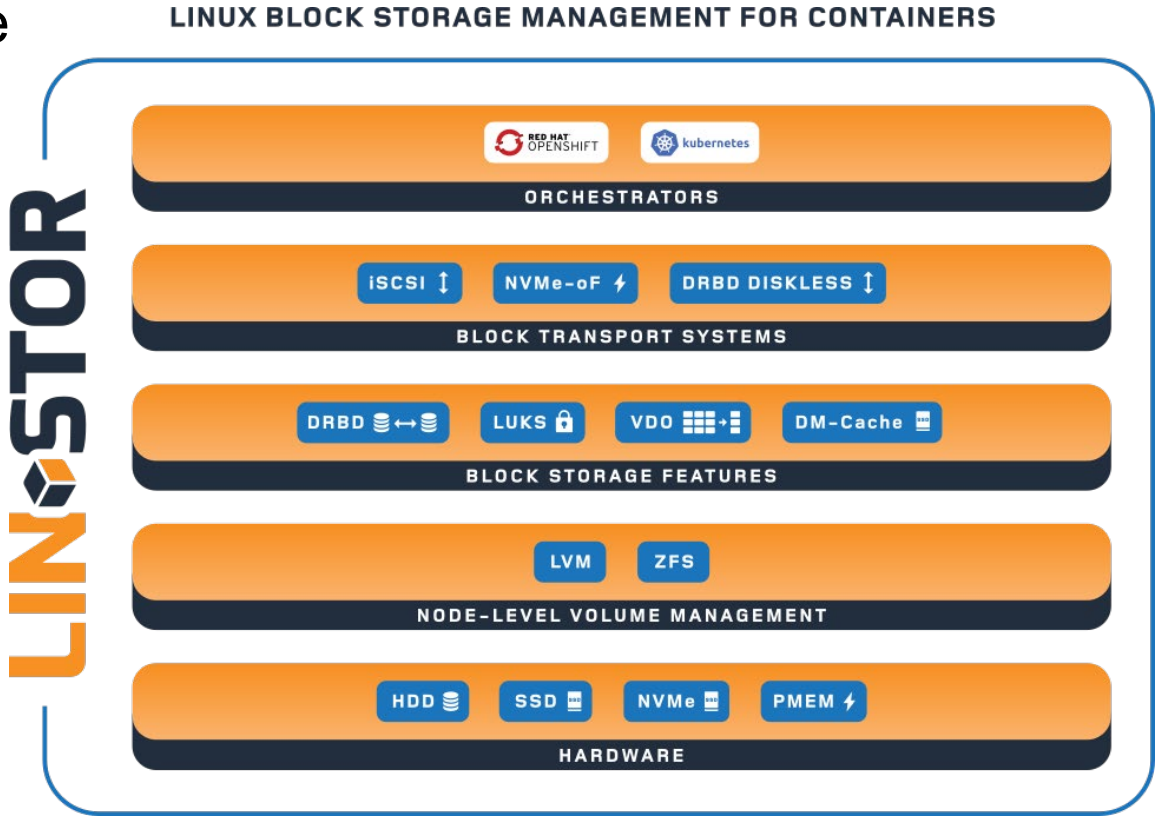
- One “Controller”
  - External to K8S
  - Or inside K8S
- Many “Satellites”
  - Via traditional init
  - Via Daemonset
    - Privileged
- REST API
  - CSI
  - Various clouds
  - CLI client



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

LINSTOR Architecture

- Block Storage
  - Linux
  - Replication
  - Locally
  - Remotely
  - Rich feature set



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Orchestrating Applications

- Kubernetes is an obvious choice
  - Container images package well
  - Pods provide predictable environments
  - Integrations are rich
- But stateful applications are challenging
  - K8S clusters are “flat”, resources anywhere by “default”
  - Stateful applications run best when “close” to their storage and the rest of the stack
  - Yet over-collocating is possible (noisy neighbor)

2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Ensuring Locality

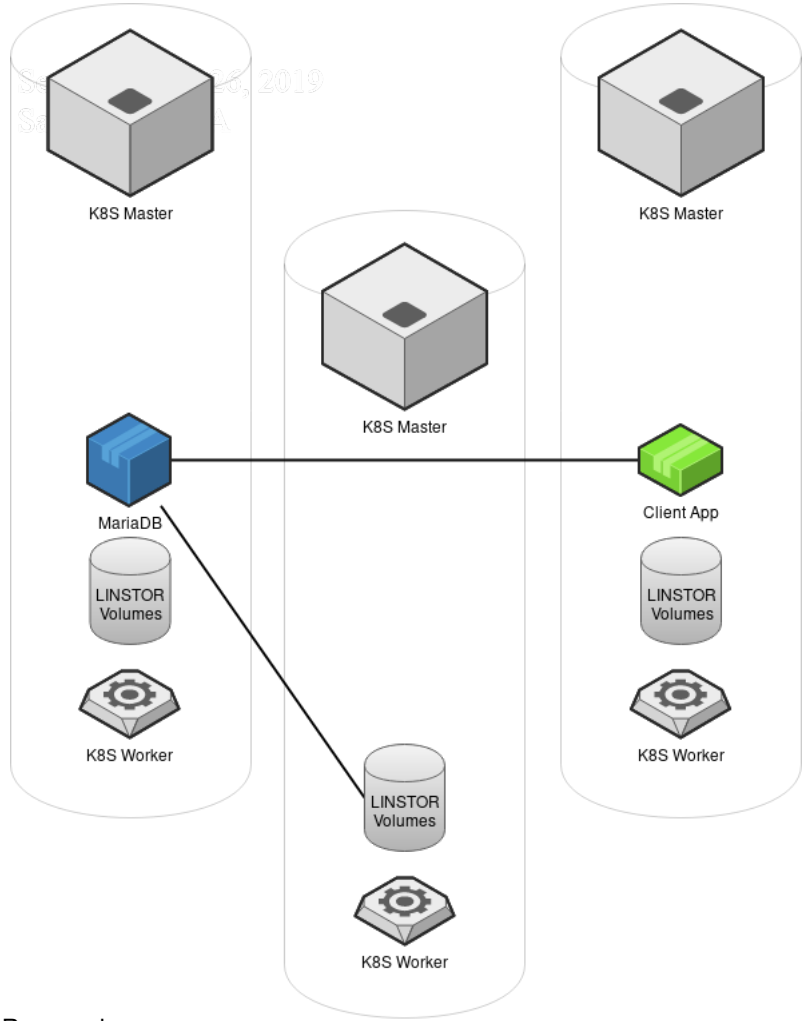
- LINSTOR controls remote and local storage
  - DRBD can be accessed via a local replica
  - LVM/ZFS/loop volumes can be used “bare” locally
  - DRBD, NVMe-oF, and iSCSI target/initiators
- Kubernetes can account for placement
  - CSI “PlacementPolicy” collocates storage + pods (new)
  - CSI “localStoragePolicy” can also accomplish this
  - Pod and node affinity / anti-affinity control placement
  - Taints and tolerances control node aversion

2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Exploring Theory and Practice

Looking at the Infrastructure

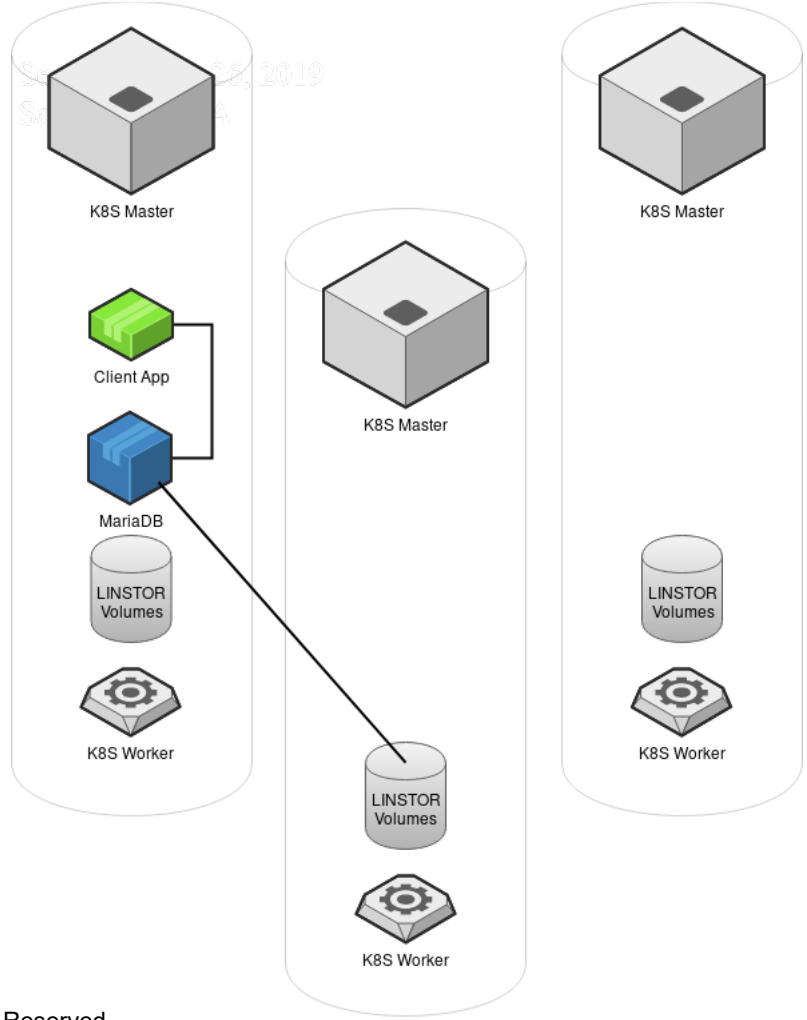
- K8S masters and workers:
  - HA
  - Multi-AZ deployment
  - Single worker node per AZ
- Resources
  - Pods placed anywhere
  - Storage placed anywhere
- What's "wrong" here?
  - App inter-stack latency
  - DB backing data is remote
  - Only one data replica



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Reducing Latency

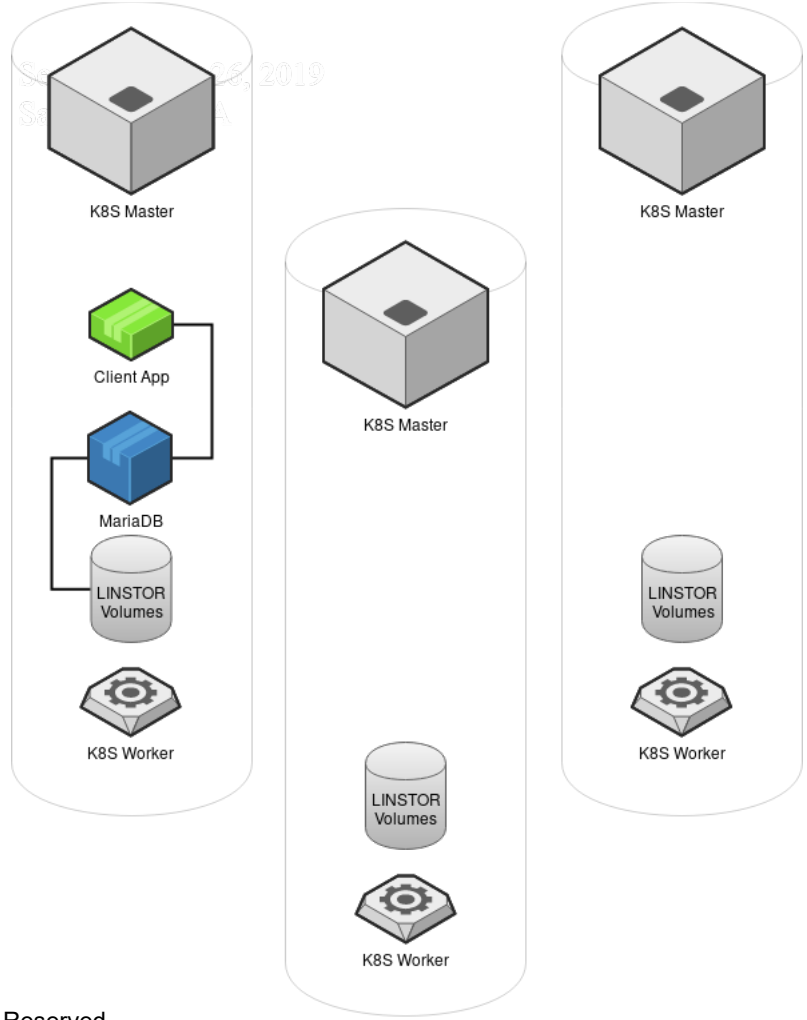
- App-to-app latency
  - Colocating reduces latency
  - Can be accomplished with node affinity or inter-pod affinity
  - Easy to accomplish
- What's "wrong" here?
  - DB backing data is remote
  - Only one data replica



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Reducing Latency

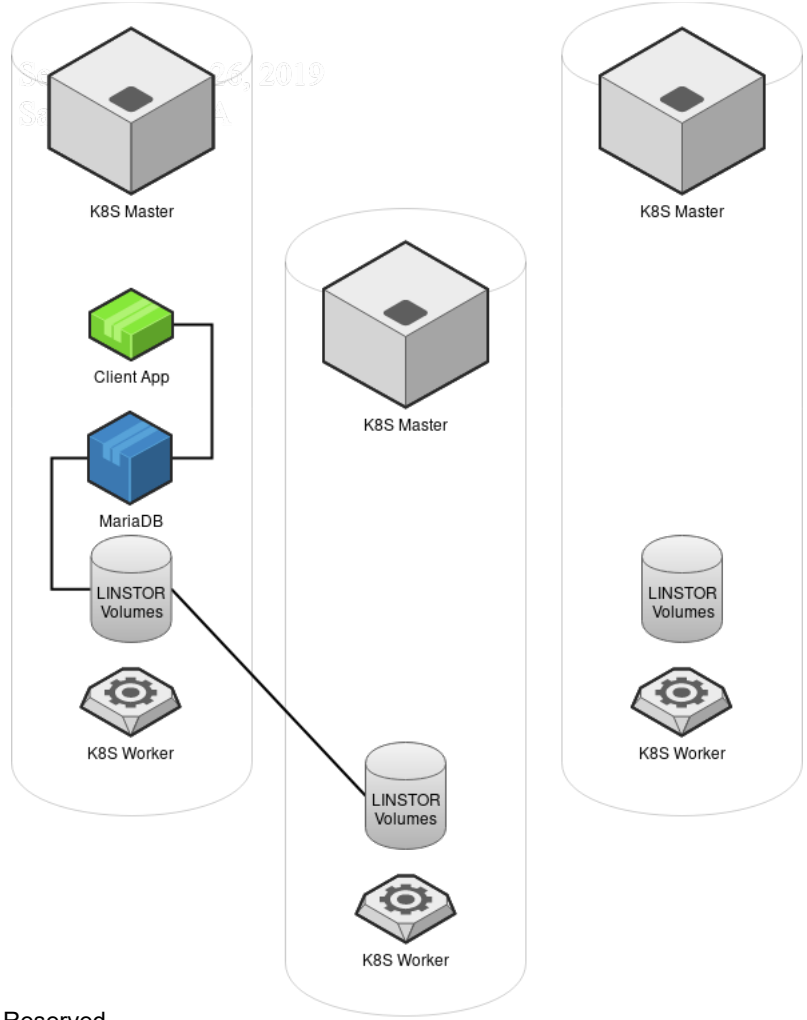
- Storage latency
  - Collocating reduced latency
  - This is looking pretty good
- What's "wrong" here?
  - Only one DB instance
  - Only one data replica



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Increasing Durability

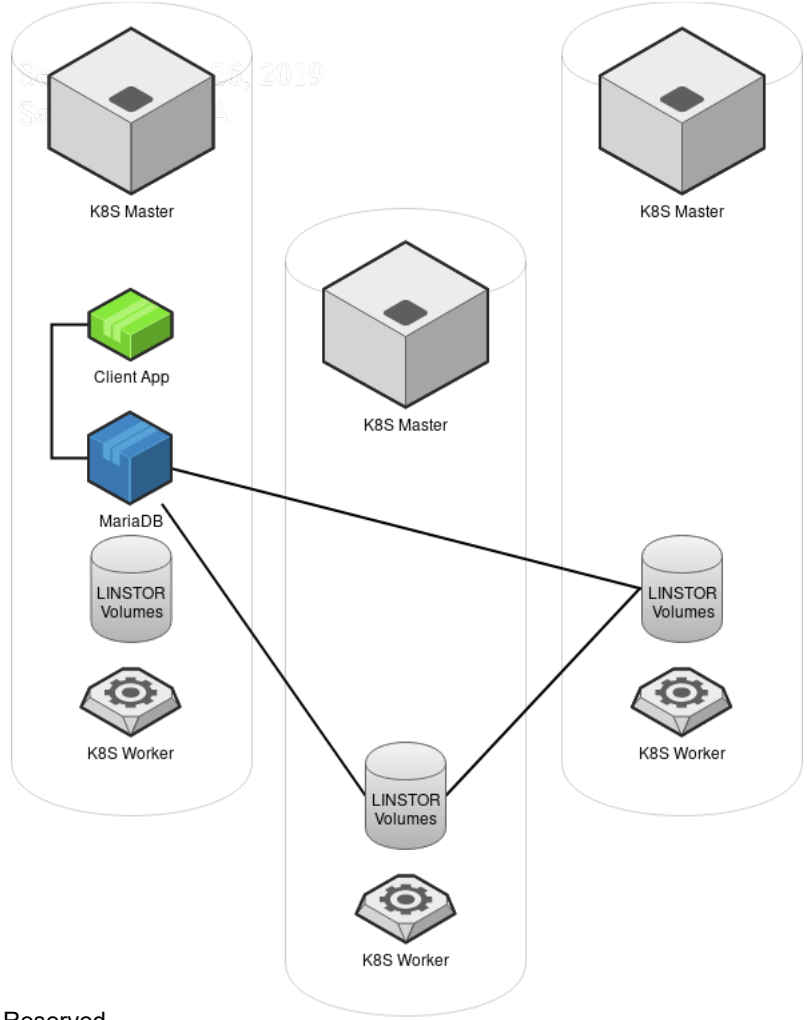
- Storage replication
  - DRBD replicates block storage
  - Data is local in two places, and remote in one
  - Reads can be balanced across replicas
  - Replicas are “HA”
- What’s “wrong” here?
  - Only one DB instance



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Increasing Durability

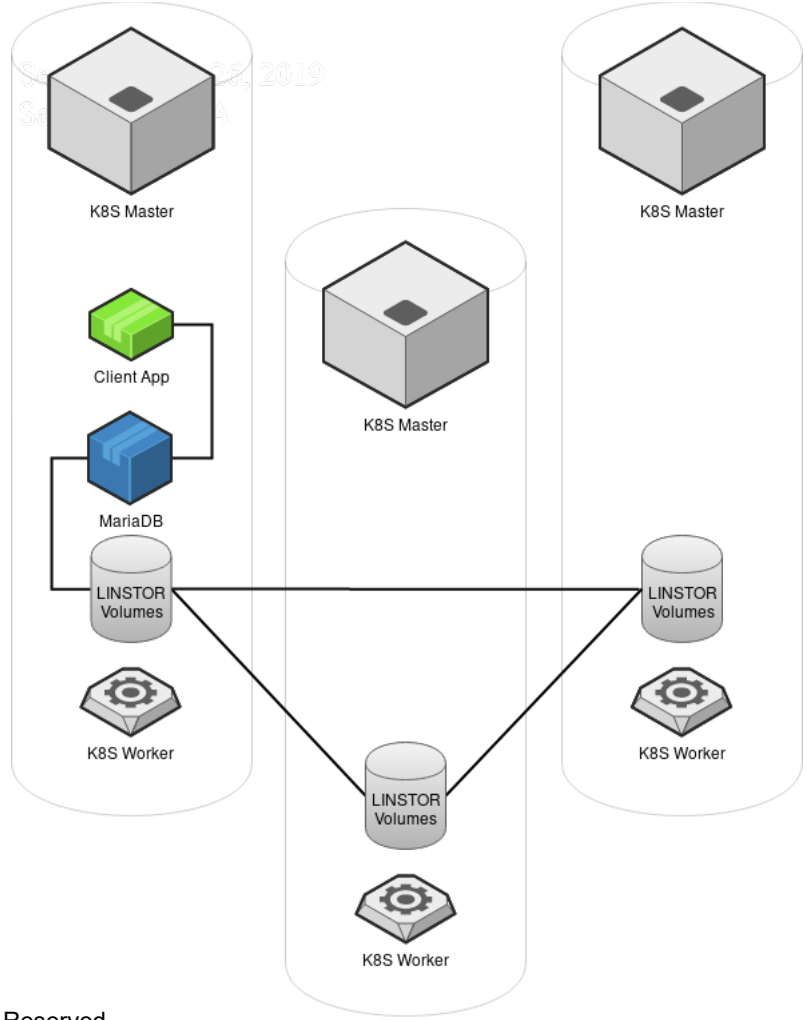
- Storage replication
  - DRBD replicates block storage
  - Data is local in two places, and remote in one
  - Reads can be balanced across replicas
  - Remote replicas are “HA”
- What’s “wrong” here?
  - Only one DB instance
  - DB storage is remote



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Increasing Durability

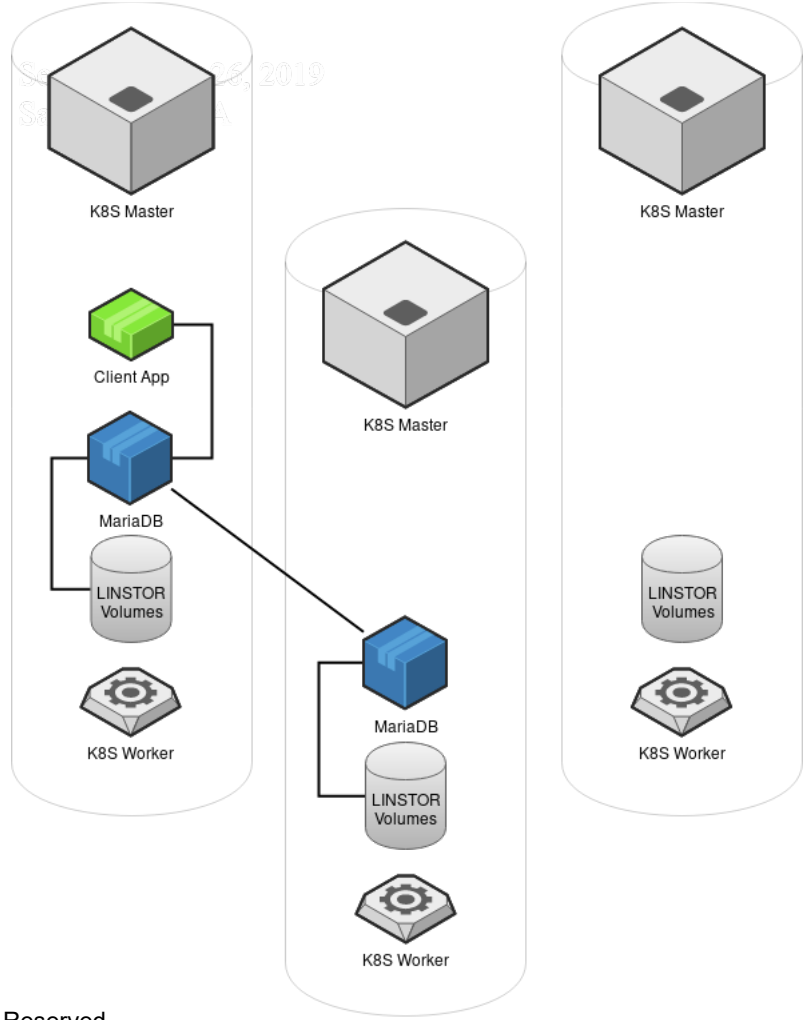
- Storage replication
  - DRBD replicates block storage
  - Data is local in three places
  - Reads can be balanced across replicas
  - More replicas can be lost without impact
  - Quorum possible
- What's "wrong" here?
  - Only one DB instance



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Increasing Durability and Availability

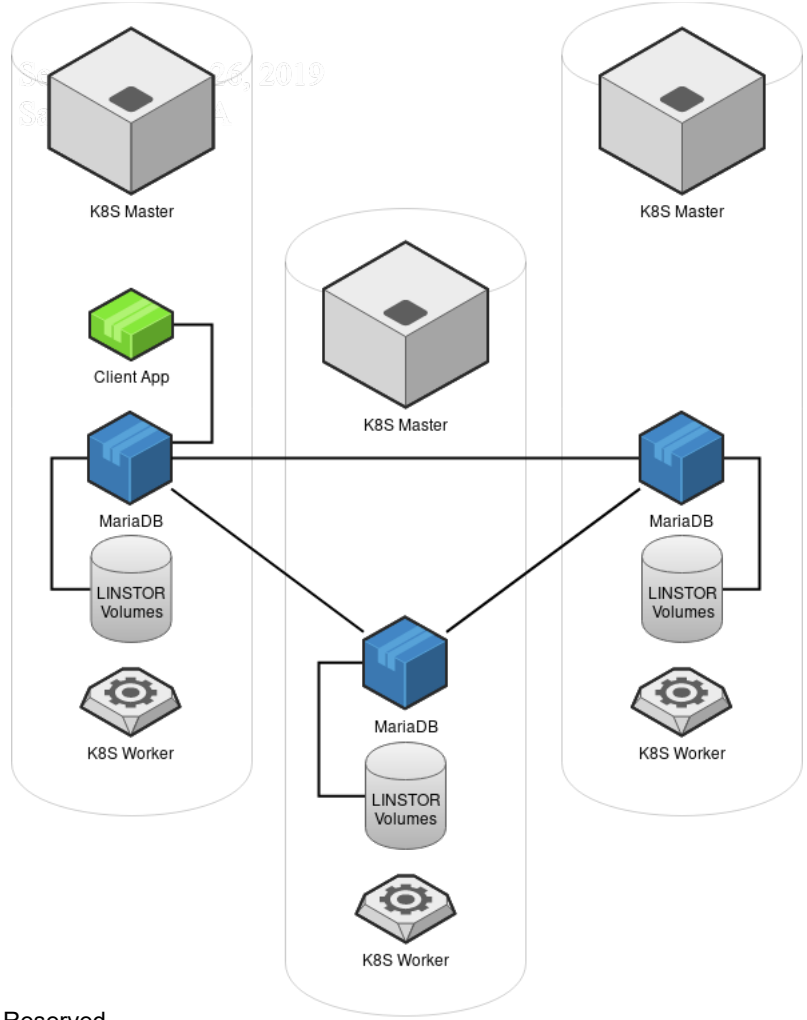
- DB replication
  - DB replicates entries
  - DB is now HA
  - Application is aware of replicated data
  - Atomic unit of transfer is smaller than a single block
  - Very little network overhead
    - Unless multi-master



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Increasing Durability and Availability

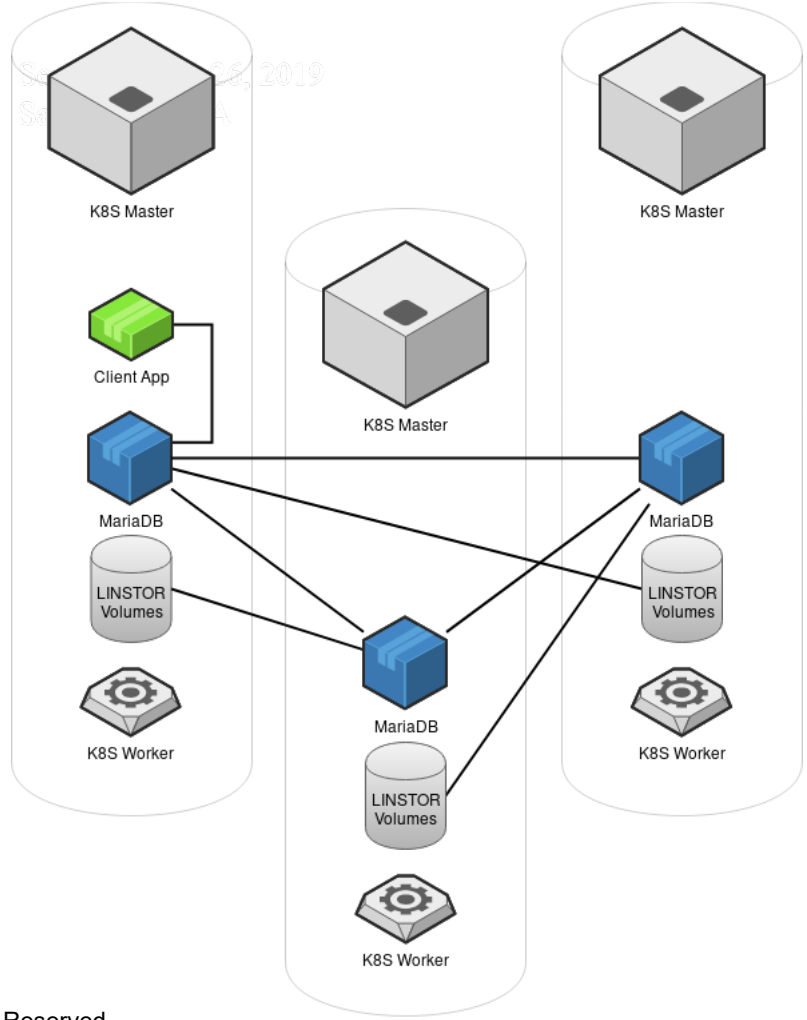
- DB replication
  - DB replicates entries
  - DB is now HA
  - Application is aware of replicated data
  - Atomic unit of transfer is smaller than a single block
  - Very little network overhead
    - Unless multi-master
- What's "wrong" here?
  - It's not always this ideal out of the box



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Considering Locality

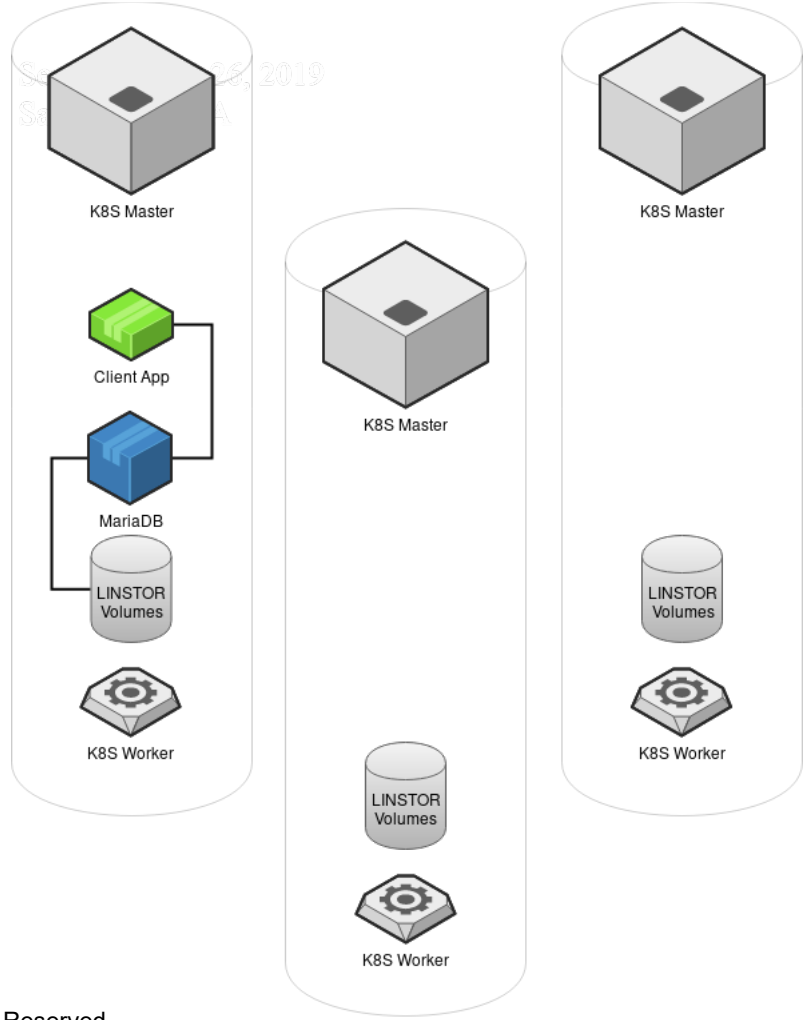
- A bit messy
  - Pods can be placed anywhere
  - Data replicas can be auto-placed
  - It can get weirder than this
    - Like collocated DB
- What's "wrong" here?
  - It was supposed to took like the previous diagram.



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Considering Locality

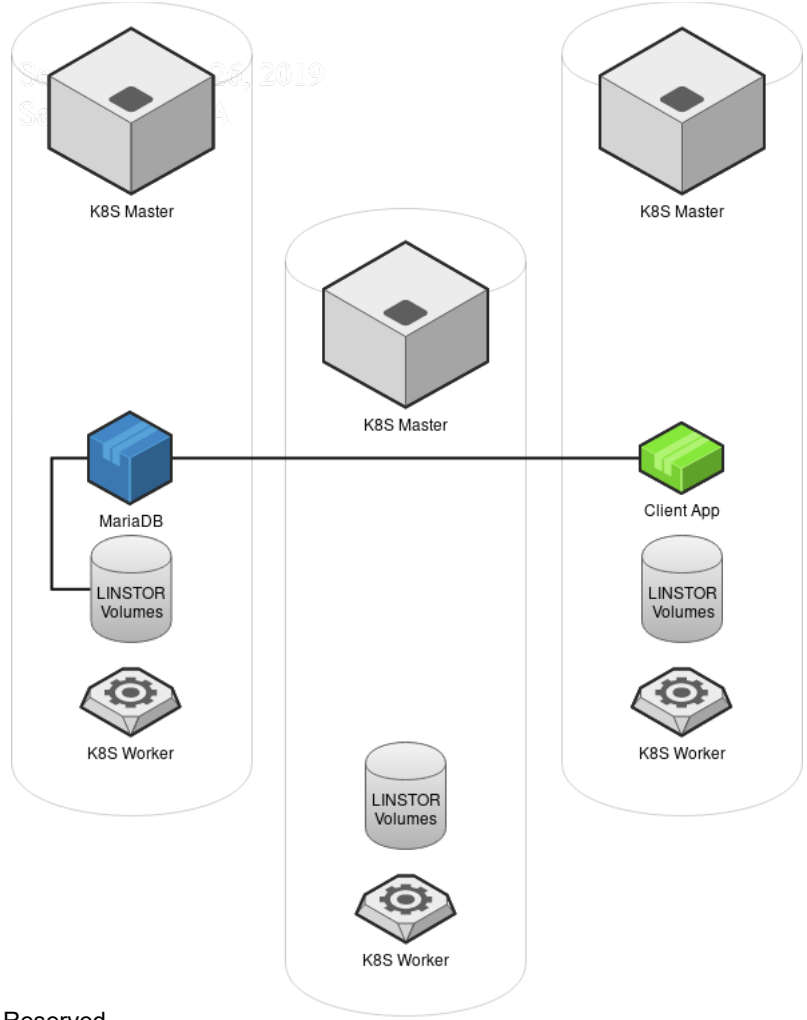
- Back to reality
  - Pod affinity / anti-affinity
    - Can be a suggestion
    - Against nodes
      - Schedules lighter, simple
    - Against other pods
      - More dynamic, complex
  - LINSTOR PlacementPolicy
    - Collocates pods and volumes in failure domains
    - Intelligently allocates
- What's "wrong" here?
  - Potential noisy neighbor (if failure domain is a node)



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Considering Locality

- The case for anti-affinity
  - Resource scheduling contention
  - Security and other motivators for workload isolation
  - Differently-abled nodes
    - Can also be easily accomplished with node affinity
- Remote access
  - While remote, remote DB access is typically leaner over the network than remote block access, so is preferred.



2019 Storage Developer Conference. © LINBIT USA, LLC. All Rights Reserved.

Please take a moment
to rate this session.

Your feedback matters to us.