SDC 19

September 23-26, 2019
Santa Clara, CA

Integrating Kubernetes Persistent Volumes into a Composable Infrastructure Platform

Brian Pawlowski
Jean-François Remy
DriveScale Inc.

The DriveScale Composable Infrastructure platform works seamlessly with Kubernetes to provide performant dynamic volumes allowing you to bring data intensive scale-out applications under this emerging data center orchestration standard without compromise.

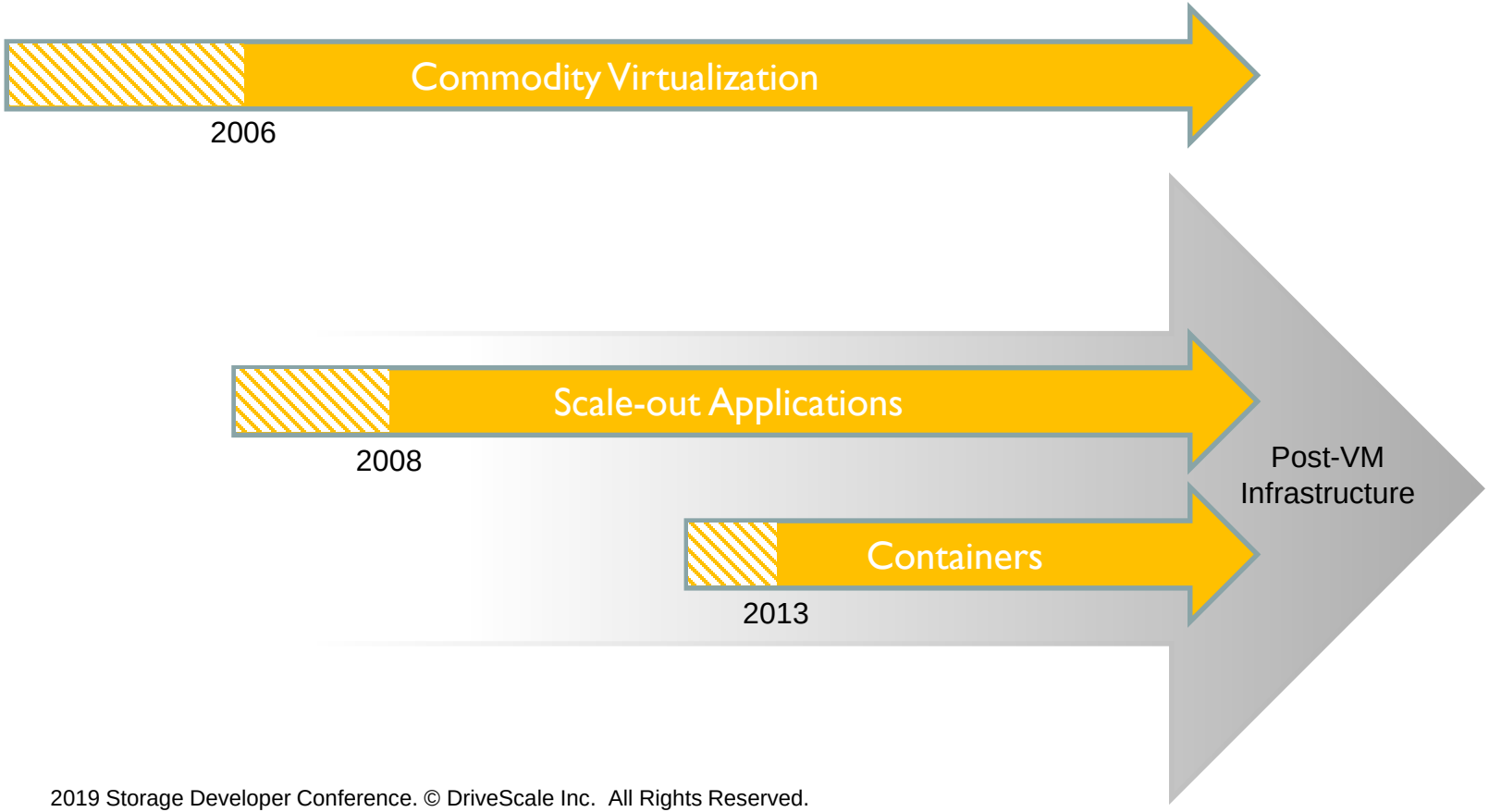
Agenda

- Kubernetes
- Container Storage Interface (CSI)
- Scale-out Applications
- DriveScale Composable Infrastructure
- DriveScale CSI Plug-in for Kubernetes
- Future
- Questions

- [Terminology] and [References]

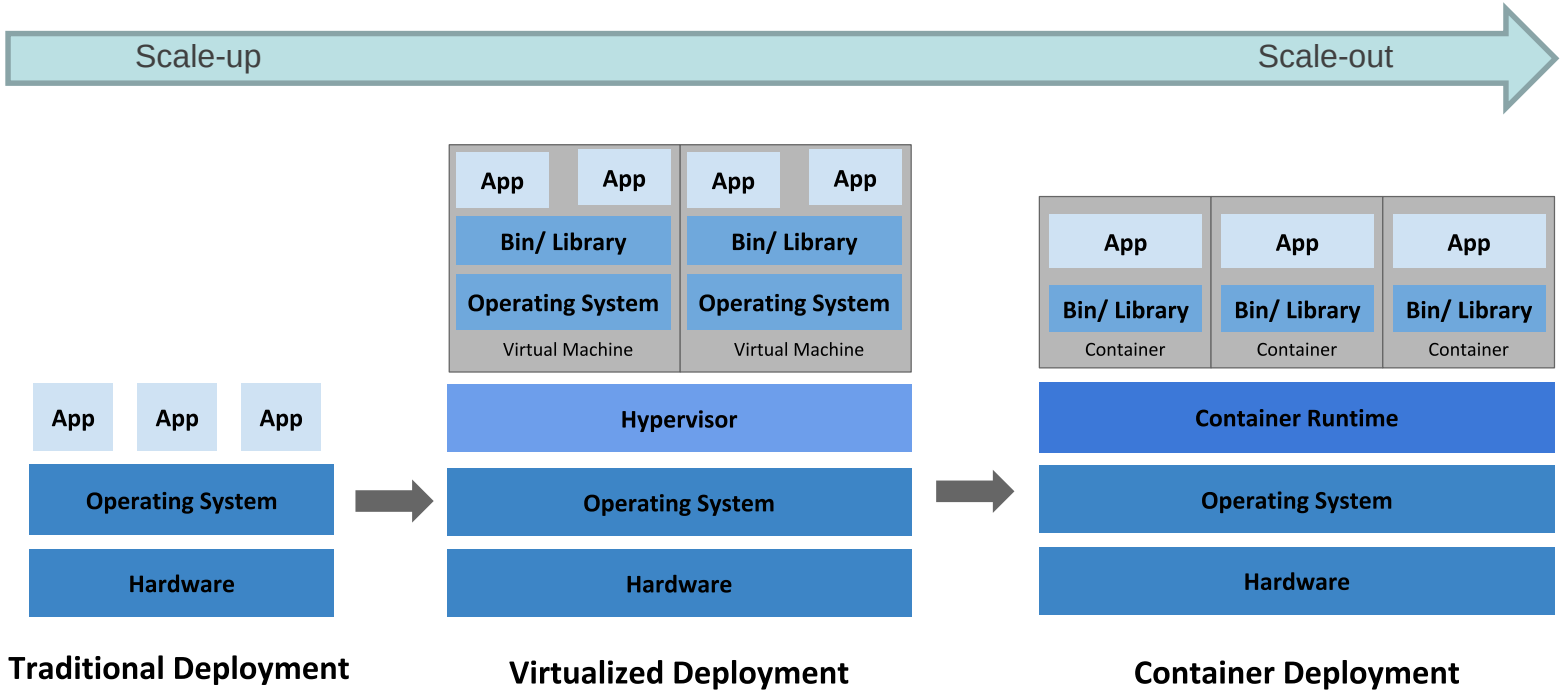
Technology Trends

Three Technology Trends



Kubernetes

Journey to Containers



From https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Containers

- An operating-system level virtualization method to run multiple isolated lightweight (Linux) systems (containers) on a single physical host
- Containers are lightweight
  - MBs for a containers vs. GBs for VMs
  - Only one copy of the host operating system
  - Common binaries/libraries can be shared by multiple applications
- Containers are highly scalable
- Containers are both hardware-agnostic and platform-agnostic.
  - They can run on your laptop or on a bare metal platform or an EC2 instance in exactly the same way
- *Containers can simplify scale-out application deployment – but don't provide the means to manage at scale*

Kubernetes

- An open-source container-orchestration platform for automating deployment, scaling and management of containerized applications.
  - Started at Google, now maintained by Cloud Native Computing Foundation
- The name Kubernetes originates from Greek, meaning helmsman or pilot. (It is also the root of *cybernetics*)
- Manage clusters of hosts (reminds one of scale-out apps)
  - Deploy, maintain, and scale applications based on CPU, memory
  - Provides grouping, load balancing, auto-healing & scaling features
  - The basic scheduling unit in Kubernetes is a pod – which co-locates containers on a host machine to share resources

CSI

Container Storage Interface (CSI)

- Consistent, orchestrator-independent volume management API
  - CSI is the preferred volume storage provider API in Kubernetes
  - Enables a wide variety of storage plug-ins
- Enables Kubernetes to flexibly support apps requiring persistent storage.
- Supports dynamic provisioning and deprovisioning of a volume.
- Exists outside of containers (Docker)
  - Attaching/detaching a volume from a node.
  - Mounting/unmounting a volume from a node.

CSI

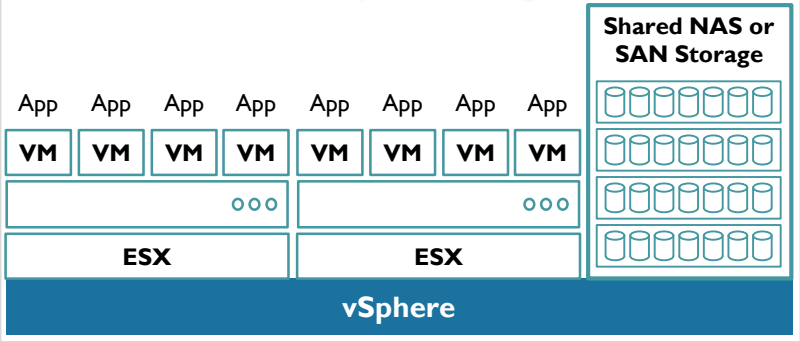
- Kubernetes is evolving as the de facto orchestration for all applications
  - Containers originally only supported stateless applications, in part because of lack of *volume management.*
  - CSI allows enterprises to use a single framework to manage stateless and stateful applications

Scale-Out Applications

Evolution of Workloads

Virtualized Workloads

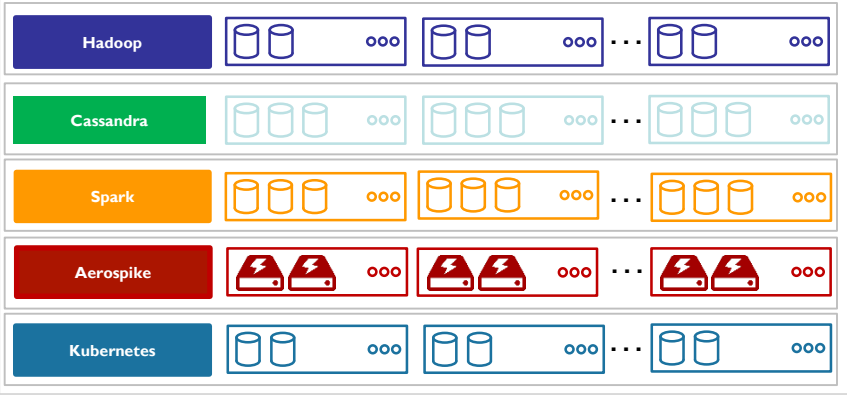
Many apps managed on each server with shared enterprise storage



- Solves the unused CPU problem by providing secure sandboxes for each application
- Each VM contains an entire copy of the OS – as if an application library
- First real standard software orchestration to application deployment

Horizontal Scale-out Data Intensive Workloads

Apps running on clusters of commodity servers with local storage



- Commodity platforms lowers cost
- Local storage configuration and management (compression/replication) done by application
- VMs and NAS introduce I/O performance bottlenecks and cost
- Traditional SAN does not scale, adds cost

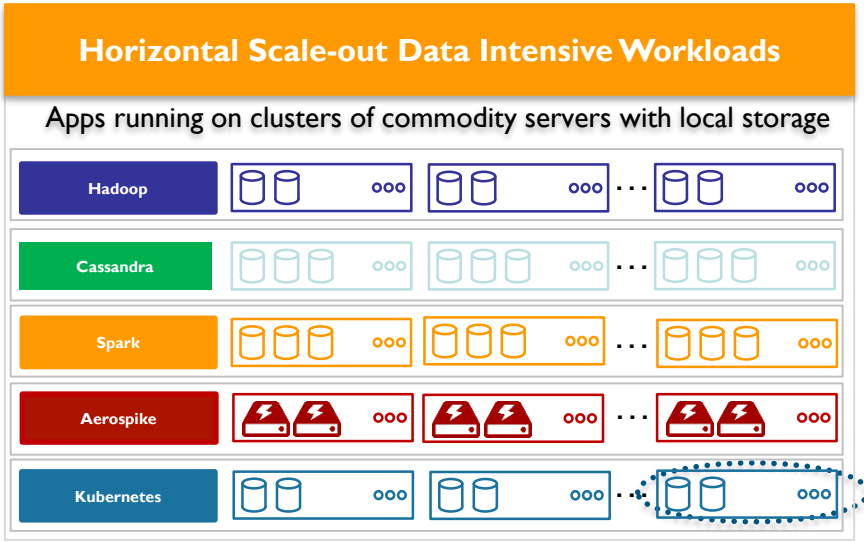
Scale-out Applications

- Applications embed high availability and resiliency
  - Usually triple replication
  - Recovery - rebuild
- Performance achieved with local Direct-Attached Storage (DAS)
  - Bring compute to storage
  - Scale compute/storage together

- ***Kubernetes does not change the basic requirements and challenges to scale-out applications***
  - But it does promise better management at scale

DriveScale in a Few Slides

Static data center architecture

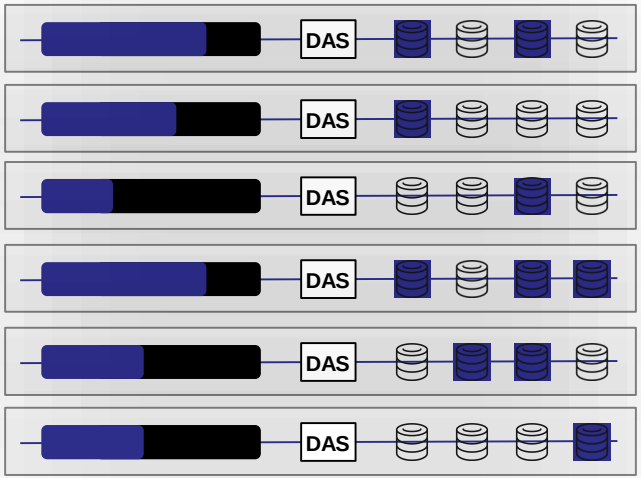


Server with local storage

- Tightly coupled, fixed resources
- Commonly overprovisioned
- Stranded compute and storage resources
- Server SKU sprawl
- Lifecycles tied together

Composable Infrastructure

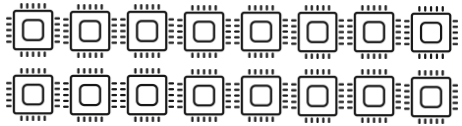
Captive, fixed DAS



Purchase Time
Defined Infrastructure

Disaggregate

Composable



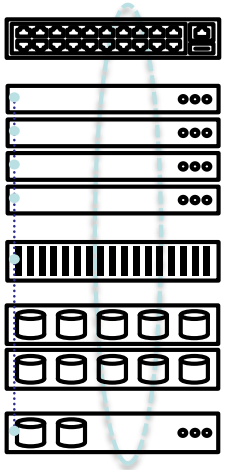
Right sized

Software Defined Infrastructure

The DriveScale Composable Platform

DriveScale Composer

- Policy
- Orchestration
- Monitoring



ToR Ethernet switches
10G, 25G, 100G, 400G

Diskless servers (boot drive)
DriveScale Server Agent

eBODs, JBODs (Flash, HDD)*
DriveScale Adapter Software

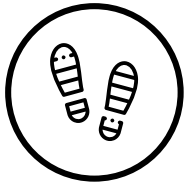
Each resource treated as independent from its enclosure

- Automated creation of dynamic servers
- Automated end-to-end set up for NVMe/TCP, NVMe/RDMA or iSCSI
- Patented load balancing
- Vendor mix-and-match

* eBOD – ethernet-attached Bunch Of Drives

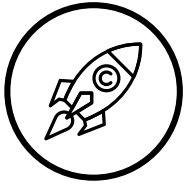
Transformative economics of DriveScale Composable Infrastructure

50% lower cost than legacy and cloud



69% less data center footprint

44% savings for upgrades

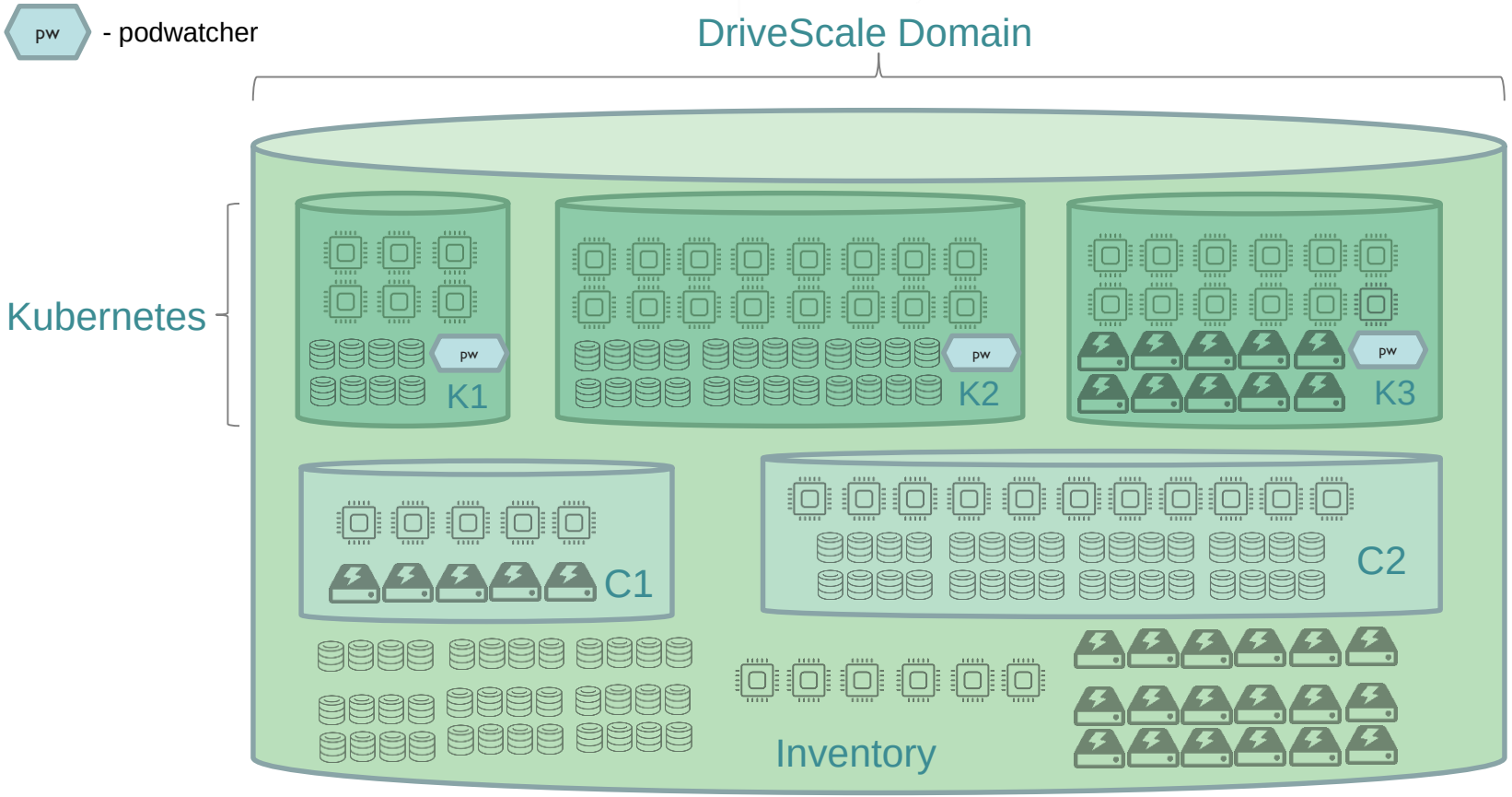


2 minutes to deploy infrastructure

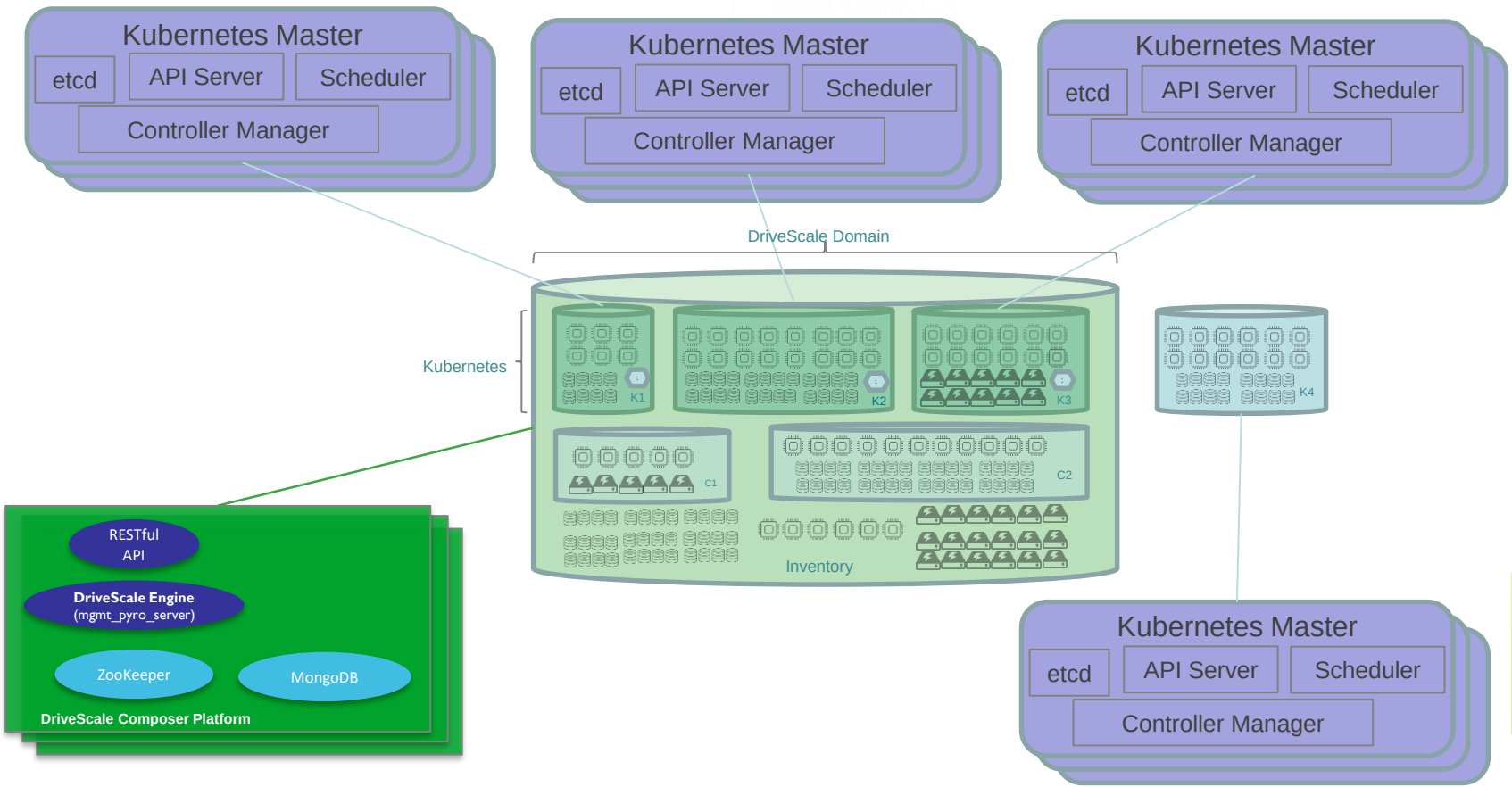
No more overprovisioning

DriveScale CSI Driver

Kubernetes on DriveScale



Kubernetes on DriveScale



Kubernetes on DriveScale

- A Kubernetes cluster maps to a single DriveScale cluster
  - Kubernetes is the software/container orchestrator
  - DriveScale is the hardware/physical orchestrator
  - DriveScale server agent runs on all nodes
- Multiple Kubernetes clusters can run in a DriveScale domain (data center)
  - DriveScale clusters for Kubernetes are created dynamically
  - Kubernetes clusters can exist outside of DriveScale
- Kubernetes and DriveScale both approach configuration management from a *desired state* approach.
  - etcd and zookeeper (respectively) are the distributed key-value stores.
  - Highly available, durable and consistent in face of controller node failures and network partitions

Configuration Link to Kubernetes

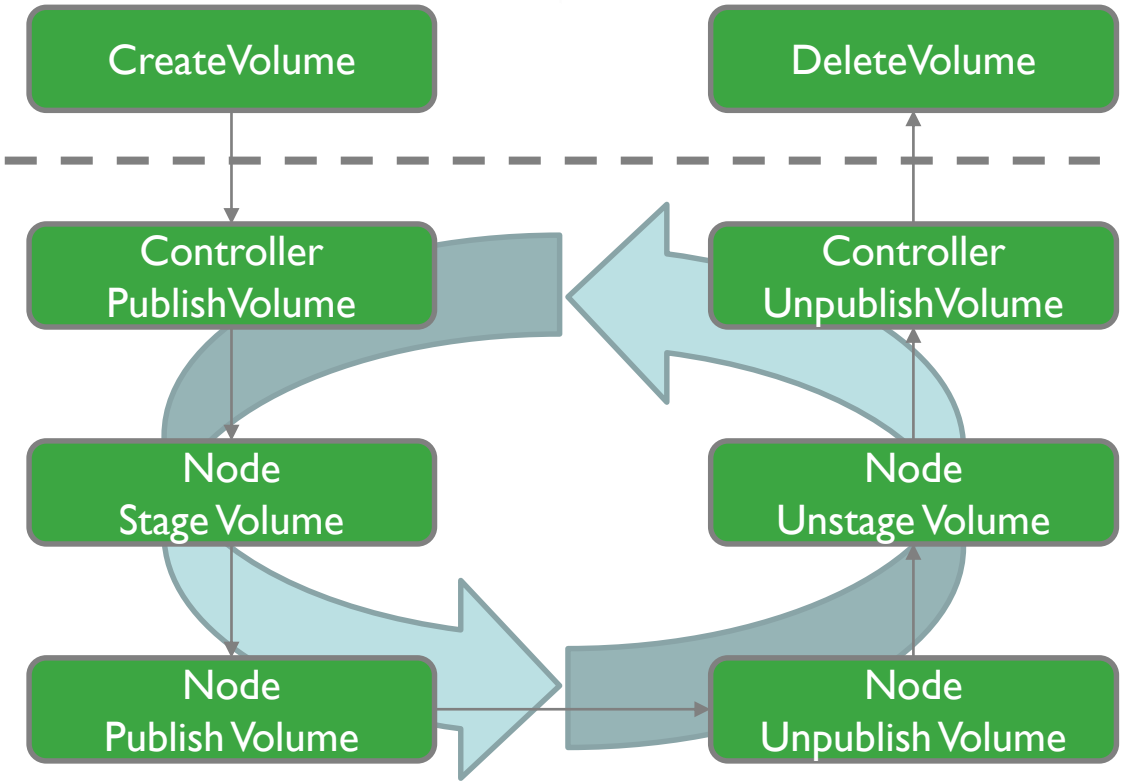
drivescale-secret.yaml:

```
apiVersion: v1
kind: Secret
metadata:
 name: drivescale-secret
 namespace: dscsi
type: Opaque
data:
```

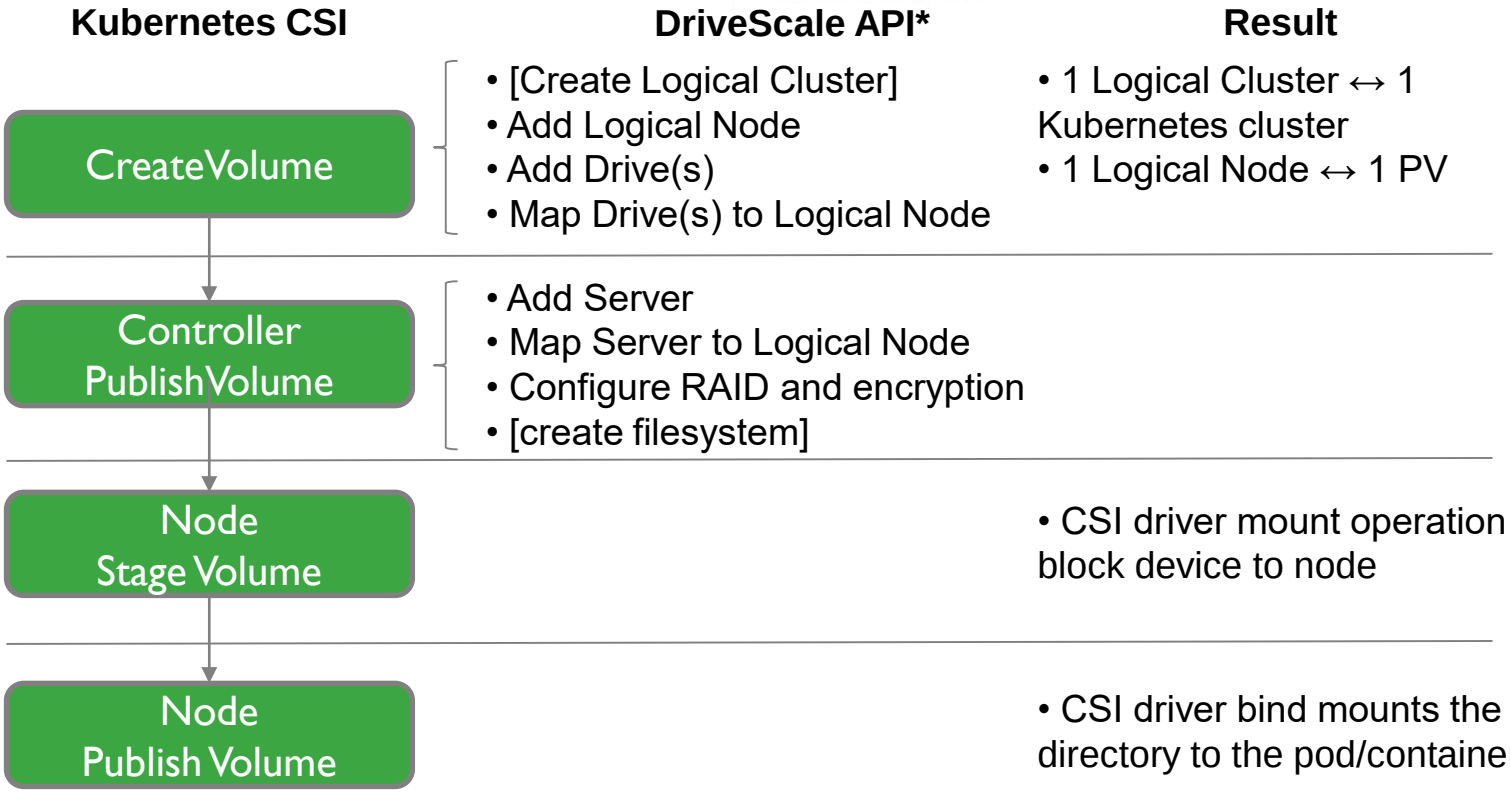
dmsUser: <base64 encoded DriveScale admin username>
dmsPassword: <base64 encoded DriveScale admin password>
dmsServer: <base64 encoded DriveScale Management server>
dmsClusterName: <base64 encoded cluster name to create - if not set, "CSI_Cluster" will be used>

- Plus the (user provided) storage class referencing provisioner csi.drivescale.com

Dynamic PV Lifecycle

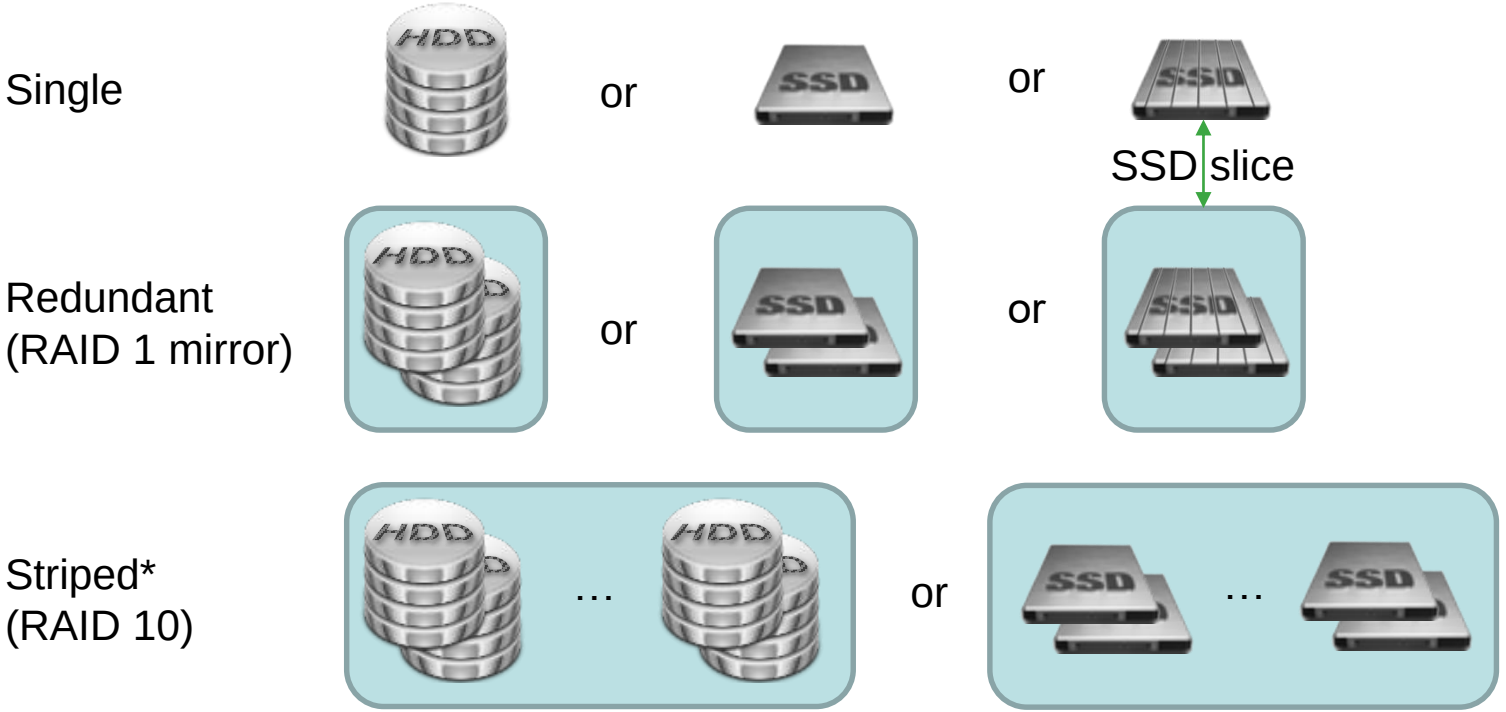


Creating a DriveScale Volume



* Items in brackets '[ ]' done on first PV in cluster or first reference of a PV for existing cluster.

DriveScale Drive Types (PVs)

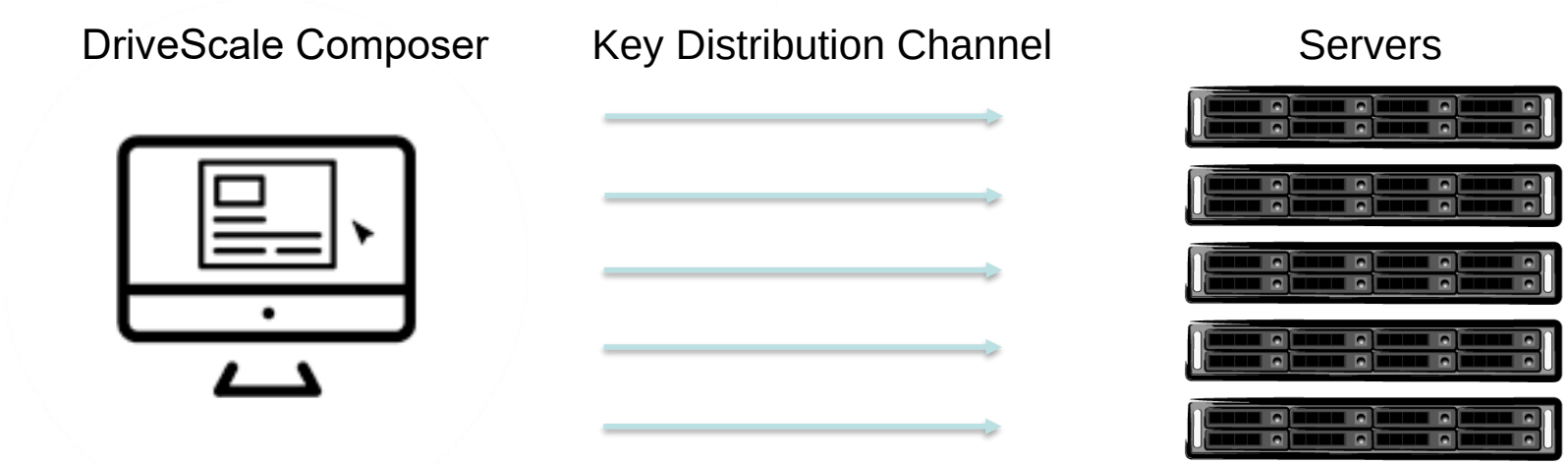


* If a request us greater than the size of an SSD or HDD, whole drives are allocated in a RAID 10 stripe. SSD slices are dynamically allocated as a right-sized portion of an SSD.

SSD Slicing

- To support Kubernetes, the DriveScale Composer was extended to automatically create SSD Slices on the fly.
- SSD can be carved into 1 GB-aligned chunks to serve multiple clients with right-sized allocations
  - The high performance (both IOPS and throughput) of SSDs make this feasible
  - Cost effective SSD sizes are increasingly too large for particular applications
- Slicing is never enabled for HDDs (sequential performance collapse)

Persistent Data Secure by design



- ✓ All filesystem and data in-flight from the server, and at-rest on the drives are encrypted
- ✓ Fully automated encryption: scales seamlessly, keys follow drive within security domain
- ✓ DriveScale Composer and Server Agents automatically create a Drive Encryption Key derived from a customer-supplied secret
- ✓ Key exchange between DriveScale Composer and Server Agents performed over a secure Key Distribution Channel
- ✓ DriveScale Server Agent plumbs Linux dm-crypt to deliver data encryption

DriveScale CSI key options

type (string)	**hdd / ssd / slice - default hdd:** type of volume (hard drive, ssd drive, slice of ssd drive)
rpm (int)	minimum RPM value of drives used in the cluster (ignored for ssd / slice)
fsType (string)	**ext4 / xfs - default ext4:** the filesystem type to use on the volume
striping (bool)	**default true:** if the plugin cannot fit the volume size on one drive, it will try create a **RAID10** array that can accommodate the total size requested. This is not supported for slices.
redundancy (bool)	**default false:** the plugin will create the volume as a RAID1 (mirror) array to ensure the volume can survive a drive failure
raidMaxDrives (int)	**default 16:** how many drives to use in a RAID array at most
sgResiliency (bool)	**default true:** should the plugin ensure that RAID volumes will survive a storage group failure (volume creation will fail if not possible). If false, the plugin will still try to provide storage group failure resiliency if it can
jbodResiliency (bool)	**default true:** should the plugin ensure that RAID volumes will survive a JBOD failure (volume creation will fail if not possible). If false, the plugin will still try to provide JBOD failure resiliency if it can

DriveScale CSI add'l options

encrypt (bool)	**default false:** should the volume be encrypted
maxVolumeSize (int)	the maximum size allowed for a volume
minVolumeSize (int)	the minimum size allowed for a volume
softDomains (bool)	**default false:** should drives meet the bandwidth domains requirements
requiredTags (string)	comma separated list of all the tags that the drives / slices used in the volume must have set
excludedTags (string):	comma separated list of any tag that would exclude a drive from being used in the volume
storageGroup (string)	comma separated list of storage groups to choose the drives from
networkTransports Allowed (string)	comma separated list of network transports allowed for the volume (allowed values are **iscsi, nvmetcp, roce**). If left empty, the system will use the most performant transport available.

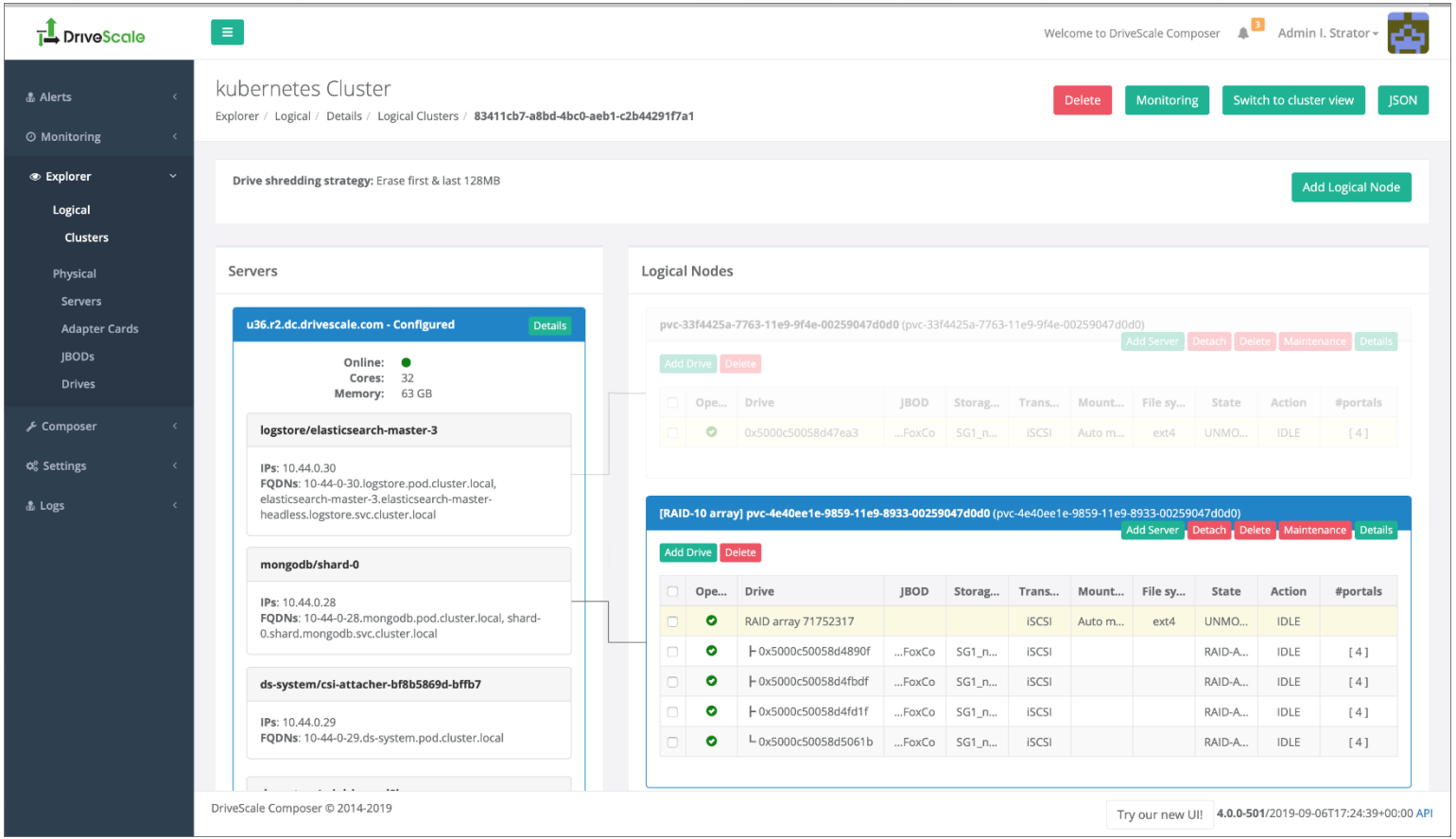
Generic CSI Reclaim Policy

- Default CSI behavior is to destroy the storage on delete of the PVC
  - Can be overridden to retain on PVC delete

Podwatcher (pw)

- Single podwatcher instance in Kubernetes cluster
  - podwatcher uses Kubernetes API to listens for pod changes
  - Annotates the DriveScale Composer with
    - pod → server/node mappings
    - PVC/Logical Node → pod bindings
- Else Logical Nodes are attached to a server with no idea why!

GUI display of podwatcher info



DriveScale CSI Driver Recap

- DriveScale CSI driver available (see resources at end of talk). It is GA.
- Scalable Persistent Storage for Containers
  - Up to 10,000 compute and 100,000 drives currently
- Shared Nothing – focus on scale-out apps
  - Deliver (local) native performance of the drives to containers
  - Equivalent to applications running in Bare Metal servers
- Data locality for Containers
  - I/O scalability/performance
  - Failure domain optimization
- Logical connection between drives and containers
  - Container mobility critical to deliver efficiency gains – avoid copies or rebuilds
  - Transparent to applications running in containers

Future Work

Future Work

- Automatic move of pod/PVC after node HW failure
- podwatcher enhancements to provide more useful descriptions of PVCs (besides their UID)
- Additional RAID types (besides 1 and 10) as needed
- Volume expansion for SSD slices possible
- Block volumes will be supported in the future.

Support Failure/Performance Domains

- Kubernetes weakness: requires manual specification of failure and performance domains
  - DriveScale currently automatically spreads drive allocations across failure domains
  - DriveScale also automatically determines bandwidth domains
  - Want to extend these capabilities to Kubernetes in future

Questions?

Thank you

Feel free to email me!
beepy@drivescale.com

Backup

History of Kubernetes

Old Days	Google Borg, etc.
2008	cgroups introduced to mainstream Linux
2013	Docker first released
2014	Google open-sources Kubernetes, world rejoices
2015	• Kubernetes v1.0 released July 21, 2015 • CNCF launches, Google's managed Kubernetes GKE opn K8s to a wider community • Red Hat's OpenShift launches
2016	• Focus moves to other clouds or on prem to adopt Kubernetes
2017	• FlexVolumes released • Managed Kubernetes begins to appear on AWS and Azure • Docker and Mesosphere announce support for Kubernetes • Kubernetes becomes the de facto standard • Production deployments at scale begin
2018	• CSI GA • Additional managed services (DigitalOcean, Oracle) • Investment shifts to lifecycle management in Kubernetes
2019	Kubernetes V1.16 released

Terminology

Term	Definition
Container	Similar to a VM, a container has its own filesystem, CPU, memory, process space, but shares the operating system on a node and therefore is considered much lighter weight.
Container Orchestrator (CO)	Automate the provisioning of containerized infrastructure and provide load balancing for the services that containers are used to create.
Container Storage Interface (CSI)	The Container Storage Interface provides a standard interface for any Container Orchestration systems (like Kubernetes) to expose arbitrary storage systems to their container workloads via Out-of-tree plug-ins.
Volume	A unit of storage made available inside of a CO-managed container, via the CSI. "Volume" or Persistent.
Persistent Volume (PV)	Storage that persists beyond the lifetime of a container.
Persistent Volume Claim (PVC)	A PVC abstracts the storage request from a pod to allow dynamic volumes.
Block Volume	A volume that will appear as a block device inside the container.
Mount Volume	A volume that will be mounted as a file system and appears as a directory inside the container.
FlexVolume	Earlier Kubernetes-specific volume API preceding CSI. (Out-of-tree) <well known location, etc.>
Out-of-tree	A plug-in, such as a CSI driver, that ships separately from the Kubernetes distribution. Versus built in storage drivers (in-tree)
SP	Storage Provider, the vendor of a CSI plugin implementation.
Drive	Any of HDD, SSD, or SSD slice
HDD	Hard Disk Drive (spinning disk)
SSD	Solid State Drive/Device
SSD slice	A right-sized virtual slice of an SSD
RAID	Redundant Array of Independent Drives (RAID 0 – striped, RAID 1 – mirror, RAID 5 and 6 – parity)
Pod	Smallest deployable unit of computing created and managed by Kubernetes. One or more containers (such as Docker containers) exist in a pod, share an IP address, have a Kubelet agent.
Kubelet	An agent that runs on each node in the cluster. It makes sure that containers are running in a pod
gRPC	Google Remote Procedure Call.
Node	A host where the user workload will be running, uniquely identifiable from the perspective of a Plugin by a node ID.
Plugin	Aka "plugin implementation", a gRPC endpoint that implements the CSI Services.
Plugin Supervisor	Process that governs the lifecycle of a Plugin, MAY be the CO.
Workload	The atomic unit of "work" scheduled by a CO. A container or a collection of containers.

References

- What is Kubernetes?
- CSI Specification
- Kubernetes CSI Introduction
- DriveScale CSI plug-in
- Volumes, plus see also CSI
- Persistent Volumes
- Kubernetes Pods
- gRPC Principles
- Introduction to the DriveScale Architecture and API (PDF link at bottom of page)
- Children's Guide to Kubernetes
- Omega: flexible, scalable schedulers for large computer clusters
- Kubernetes (Wikipedia)
- Persistent volumes by example
- DriveScale is certified OpenShift and now published in the Red Hat Container Catalog: https://access.redhat.com/containers/
- DriveScale Kubernetes Solution Brief