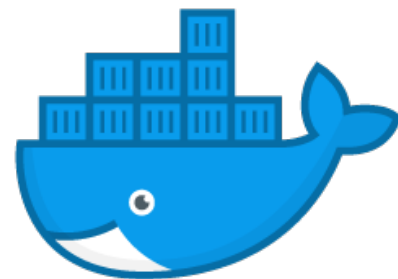


Catching up with Container Storage

Keith Hudgins, Technical Alliances
Docker, Inc



docker.

Introductions

- ❑ I work for Docker
- ❑ I work in business development, but I'm a sysadmin by trade
- ❑ This is not a vendor pitch!
- ❑ But, my experience with non-Docker integrations is not as strong as I'd like it to be yet.
- ❑ Hold questions to the end, but interrupt with corrections please!

What is a container?

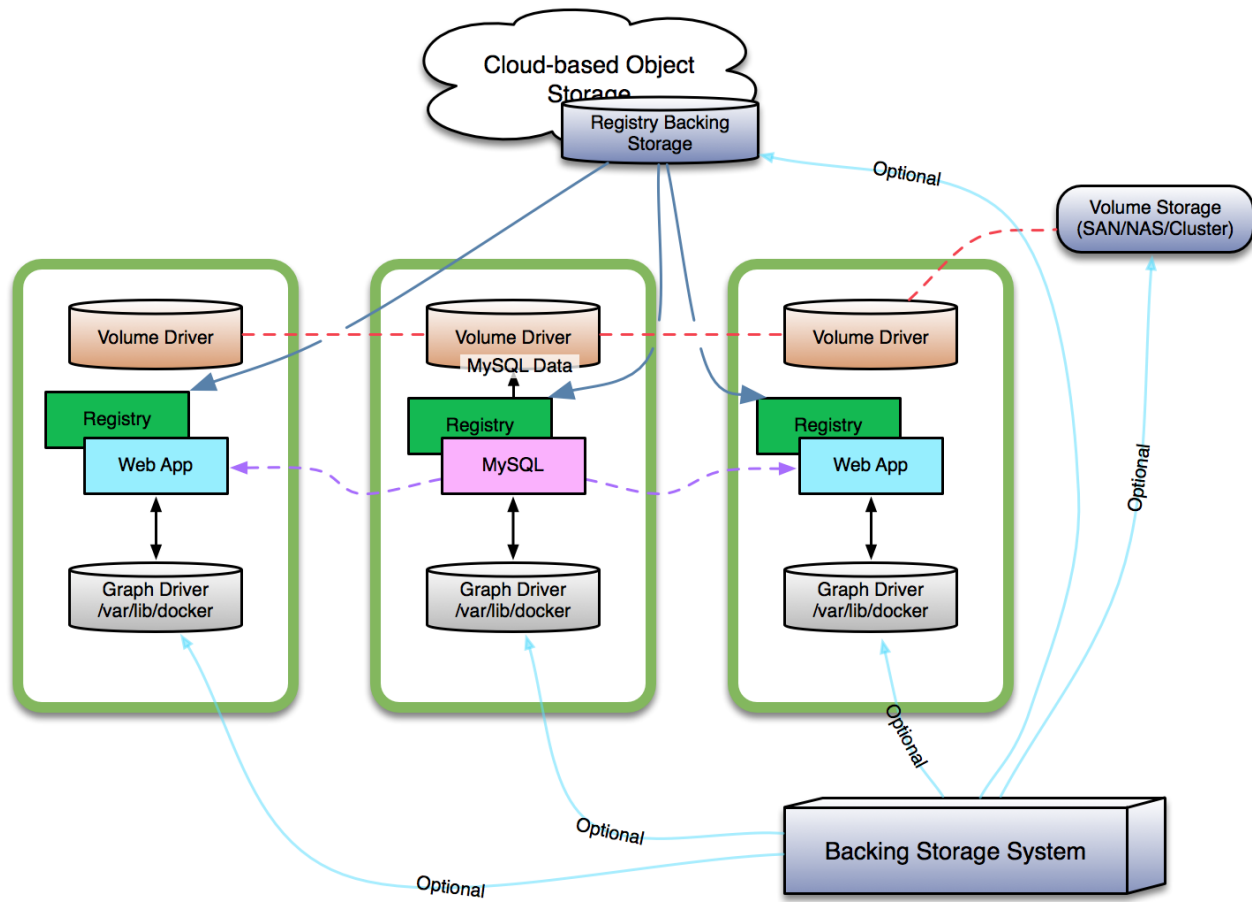
- ❑ Starts with a base OS system image
- ❑ Container init = 1 process only
- ❑ Process runs in its own namespace
- ❑ Shared access to system resources (hardware, network, etc)
- ❑ Containers Copy-on-Write image is NOT saved when process exits!

Container Community Orgs

- ❑ Cloud Native Computing Foundation (CNCF)
- ❑ Open Container Initiative (OCI)
- ❑ Both are part of the Linux Foundation

Container Storage Types

- ❑ Runtime Storage - Graph Driver
- ❑ Cold Storage for Images - Registry Storage
- ❑ Persistent Storage

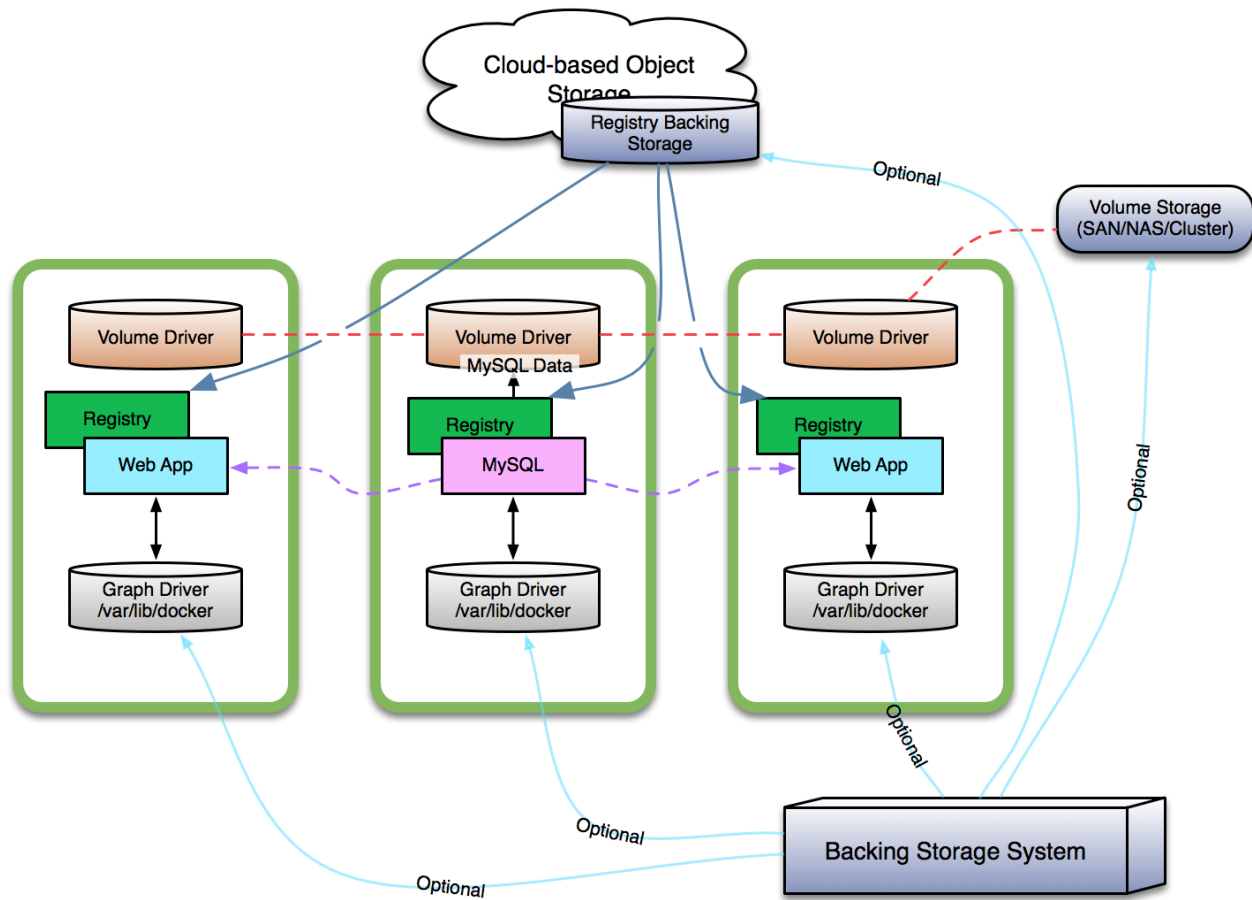


Graph Drivers

- ❑ Integration point determined by container runtime
 - ❑ <https://blog.mobyproject.org/where-are-containerds-graph-drivers-145fc9b7255>
 - ❑ Basically, two driver styles:
 - ❑ Overlay
 - ❑ OverlayFS, AUFS
 - ❑ Snapshot
 - ❑ Btrfs, zfs, devicemapper

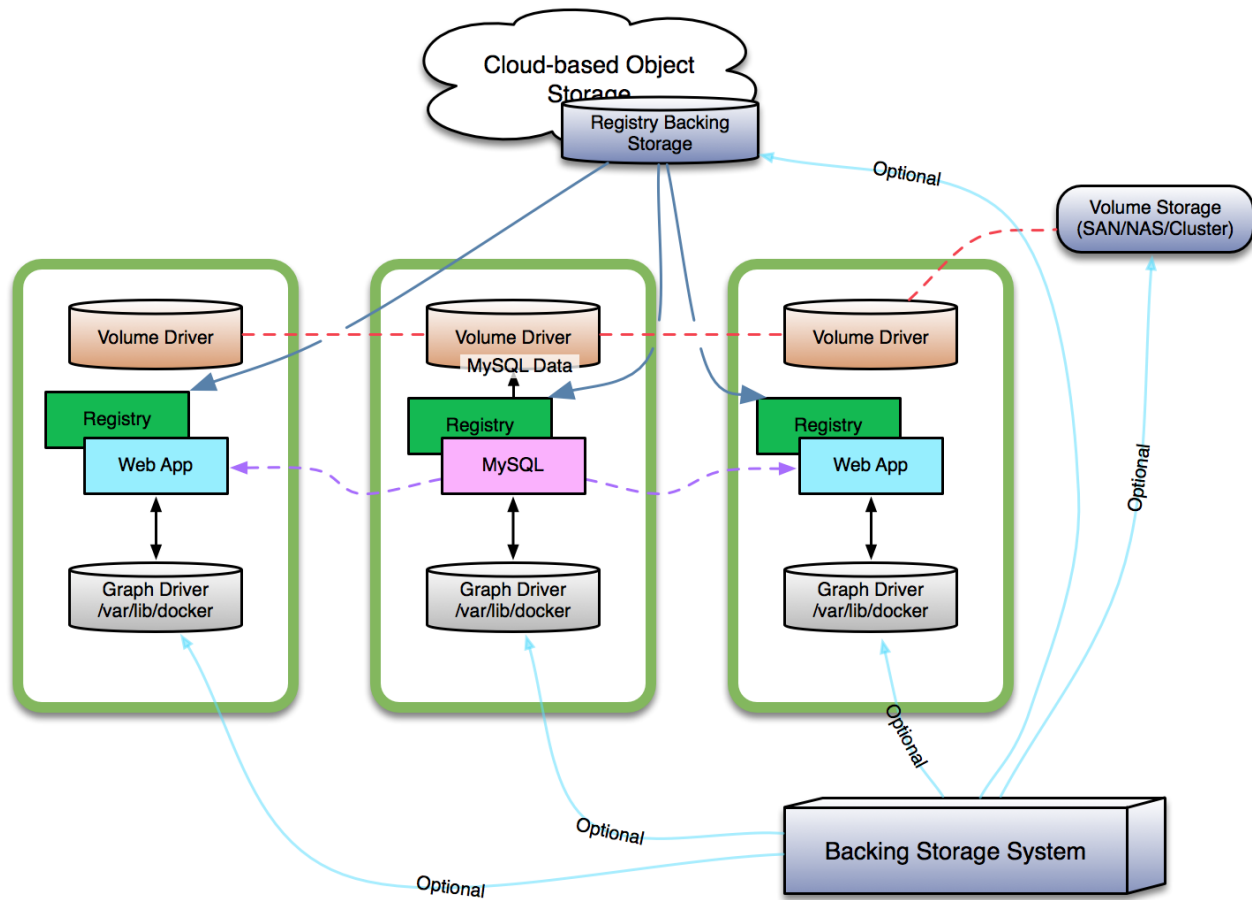
Container Runtimes

- ❑ Docker (containerd)
 - ❑ Default for most container installs, widest user base
- ❑ CoreOS (rkt)
 - ❑ Aims to be ultra-thin, ultra-simple
 - ❑ CoreOS recently acquired by RedHat
- ❑ CRI-O
 - ❑ Intended to be Kubernetes-native runtime
- ❑ Upstream of these projects are all CNCF efforts.



Registry Storage

- ❑ Many private registry options.
- ❑ Pretty much all use either
 - ❑ File-based storage
 - ❑ Object storage
- ❑ Docker Registry can use either
 - ❑ Supports any mounted filesystem for file-based
 - ❑ S3 or OpenStack Swift APIs for object storage



Persistent Storage

This category is the single most important integration point for most storage vendors.

There is little support within the container ecosystem for native persistence, and yet it is a major need for users.

Persistent Storage Details

- ❑ Every implementation works pretty much the same:
 - ❑ Plugin/integration code receives instructions from the framework to manipulate a file or block device
 - ❑ Plugin reaches out to backing store to mount the new device on the host system and prepare it for use (formatting, etc)
 - ❑ Plugin hands mount point back to framework
 - ❑ Framework bind-mounts it into a container prior to container launch
- ❑ APIs are a little different, and some other capabilities (snapshots, RBAC, etc) may or may not be baked into the APIs

Container Frameworks

- ❑ Docker Swarm
- ❑ Kubernetes
- ❑ Mesosphere

Docker Swarm & Mesosphere Integration

- ❑ Docker Volume Plugin supports two versions:
- ❑ V1: Supports Windows and Linux
 - ❑ Installs out-of-band of container solution
 - ❑ Less secure by default (can be manually tightened down)
- ❑ V2:
 - ❑ Install is containerized, so very easy for users
 - ❑ Uses Linux kernel capability controls for security management
 - ❑ Unfortunately, that makes it Linux-only
- ❑ https://docs.docker.com/engine/extend/plugins_volume/
- ❑ Requires Docker Engine, either Community or Enterprise

Kubernetes Integration - Important Points

Kubernetes is the de facto standard for new container projects.

You should be considering Kube Integration.

Kubernetes is a community project. Like all major community projects, there are many cooks. Which means there's several ways to integrate.

Kubernetes Concepts

- ❑ Kube is operator-centric: admins must prepare volume offerings to users
- ❑ Applications are managed in pods
 - ❑ Users don't manage singleton containers in Kube - always app stacks via pods
- ❑ Admins prepare resources for users
- ❑ Storage Class - Admin prepares
 - ❑ Persistent Volume Claim - Declared in pod file
 - ❑ Persistent Volume - Dynamically created at runtime for pod
- ❑ Community links to get started
 - ❑ <https://github.com/kubernetes/community/tree/master/sig-storage>
 - ❑ <https://github.com/kubernetes/community/blob/master/sig-storage/contributing.md>

Kubernetes In-tree Storage Drivers

- ❑ These are baked in to the Kubernetes source
- ❑ Most distros will ship them
- ❑ <https://kubernetes.io/docs/concepts/storage/>
- ❑ Example volume driver:
 - ❑ <https://github.com/kubernetes/kubernetes/tree/master/pkg/volume/rbd>
- ❑ You WILL need to join the Kubernetes community org and sign their Contributor License Agreement - this may not be a good fit for everyone!

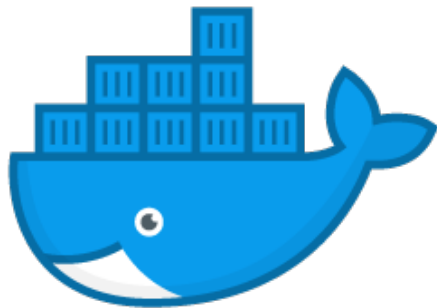
FlexVolume Plugins

- ❑ 1st released persistence API for Kubernetes
- ❑ Fully GA and supported as of Kubernetes 1.8
- ❑ Docs are hard to find - Kube docs team are moving things
- ❑ <https://github.com/kubernetes/community/blob/master/contributors/devel/flxvolume.md>
- ❑ Internal Kubernetes project

Container Storage Interface (CSI)

- ❑ CNCF Incubator Project
 - ❑ This means more overall community support than FlexVolume
 - ❑ However, this is also newer, and not GA yet. Expected GA release for Kubernetes late Q4 2018
 - ❑ Cloud Foundry (not specifically containers) and Mesos have announced support for CSI as well
- ❑ <https://github.com/container-storage-interface>
 - ❑ Several repos there that are useful.
 - ❑ Full specification is in the Spec repository
 - ❑ <https://github.com/container-storage-interface/spec/blob/master/spec.md>

Questions? And thank you!



docker

Keith Hudgins
keith@docker.com