

# Storage Integration for Docker

Keith Hudgins  
Technical Alliances, Docker Inc.

keith@docker.com

@keithhudgins



## Containers 101:

- A container consists of an image, an execute pipeline, and containment rules
  - Image
    - Typically an OS image, but can be very tight and streamlined
  - Execute pipeline
    - init process (script, executable, etc)
  - Containment rules
    - Control groups & namespaces to restrict containers execute pipeline to minimal resources. (Principle of Least Responsibility)



## Containers 102 (Tips, Tricks, and Gotchas):

- By default, containers write into a running instance of their system image.
  - Running instance is ephemeral, so no persistent data
- Network access must be specified at runtime
- Applications are composed of multiple containers managed by some kind of orchestration layer (Swarm, Kubernetes, Mesos, etc)



## Docker Storage Types

**Registry**

Cold storage of container images

**Graph**

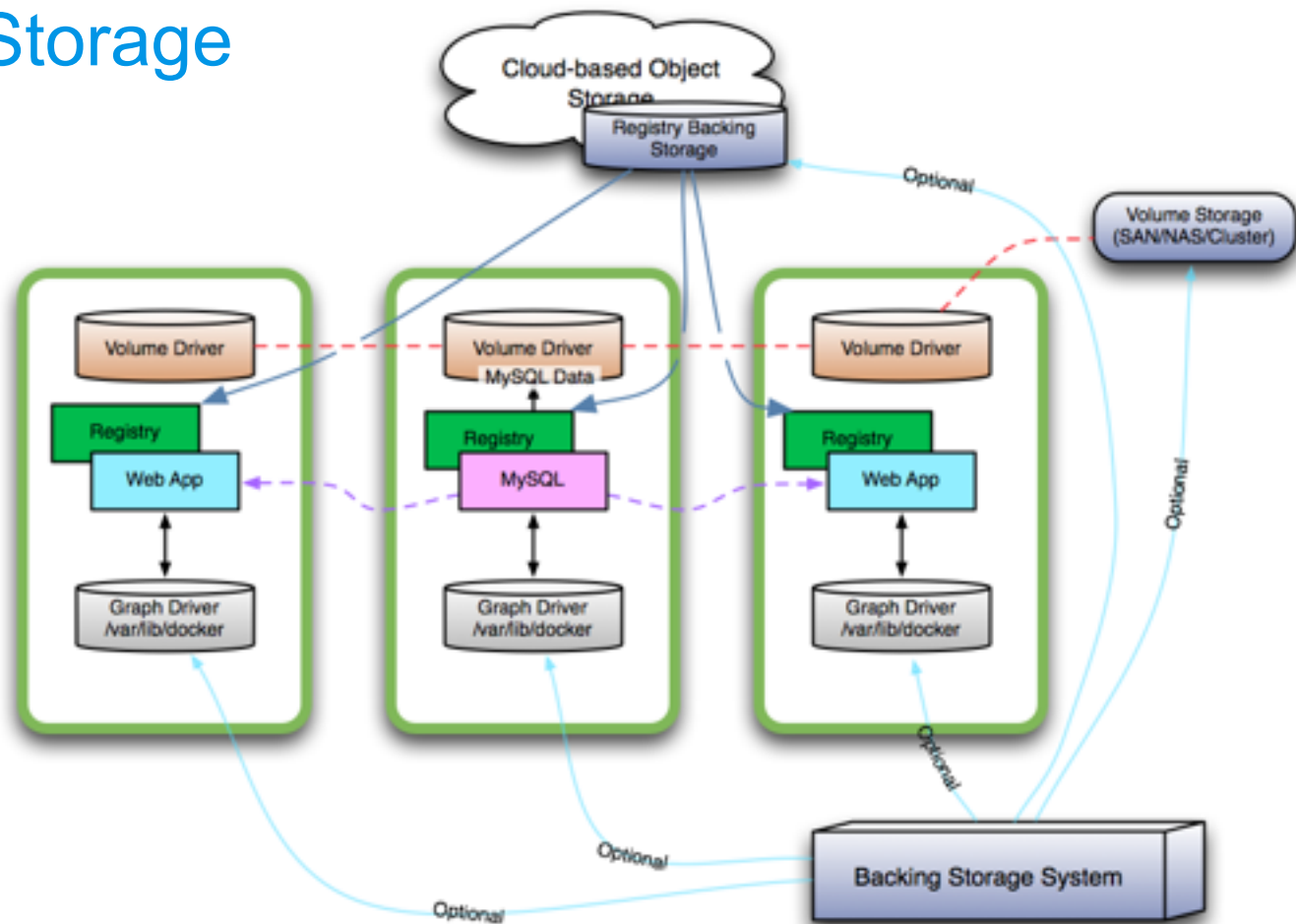
Active storage of running container images

**Volume**

Persistent block storage for data



# Docker Storage

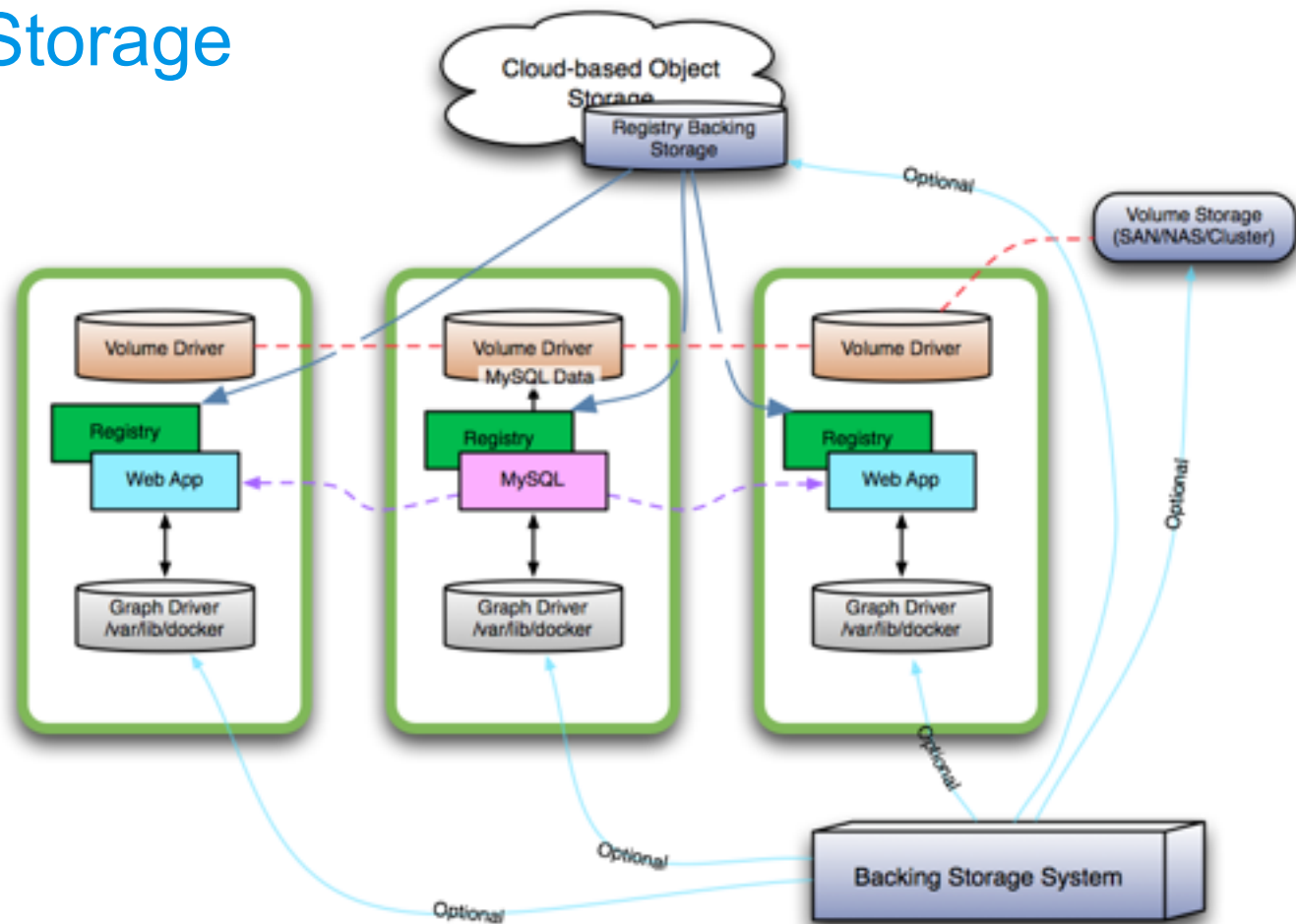


## Docker Registry Storage

- Stores container images when not in use (container at rest)
- Registry service runs as container underneath Docker Engine (For Docker's Opensource and commercial registry products - there are 3rd party available.)
- Config data stored via standard Docker volumes
- Images stored via driver
  - Native filesystem (We don't care what's beneath - attached/remote storage is best!)
  - Drivers available for cloud object storage for images (S3, Swift, GCS)
- No heavy lifting required to integrate



# Docker Storage



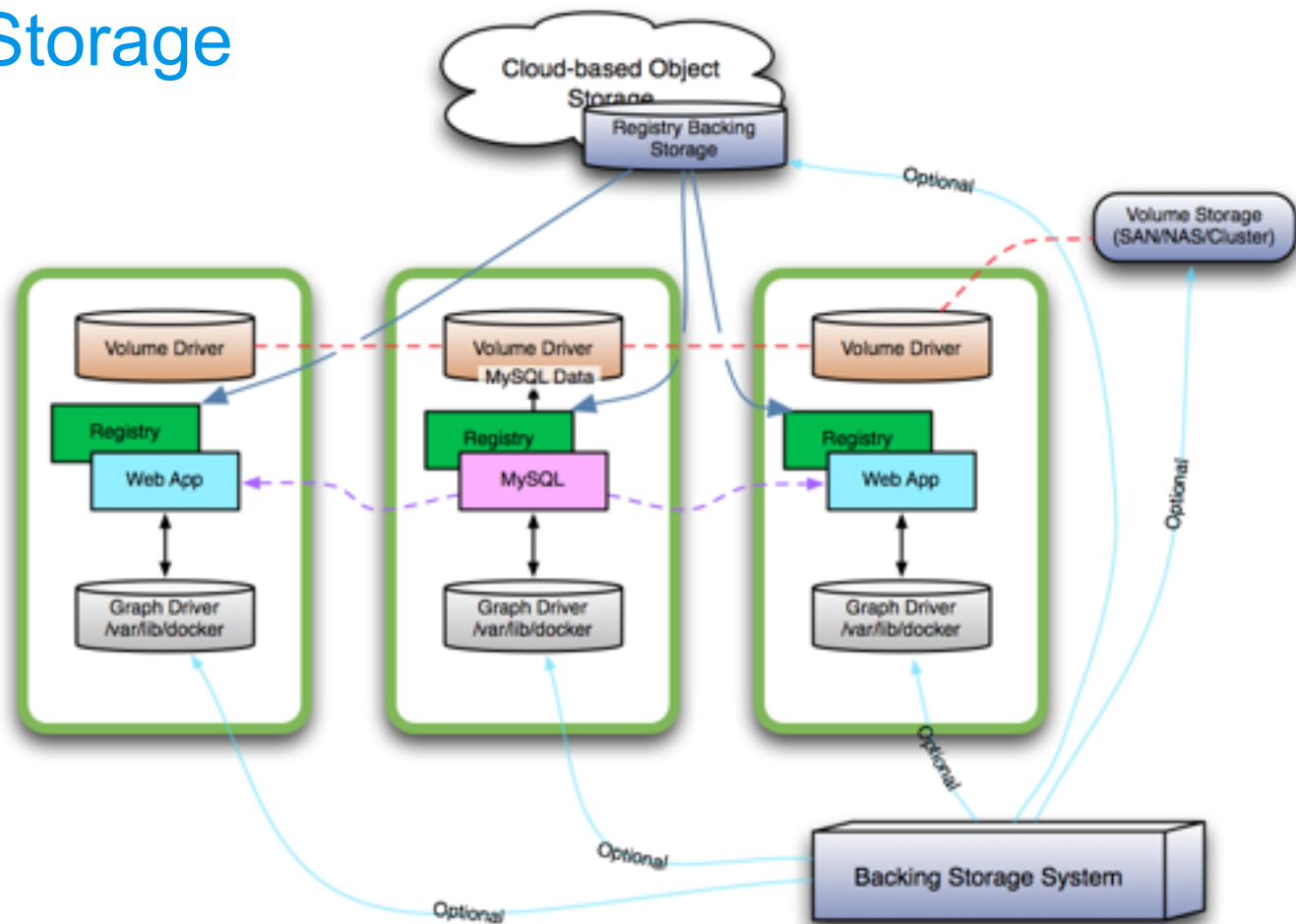
## Docker Graph Driver

- Used for actively running containers on a Docker host. (container in motion)
- Writes to local filesystem (Can be backed by network storage, depending on driver)
- Image is copy-on-write
- Diff layers are removed when container is deleted
- <https://docs.docker.com/engine/userguide/storagedriver/selectadriver/>





# Docker Storage



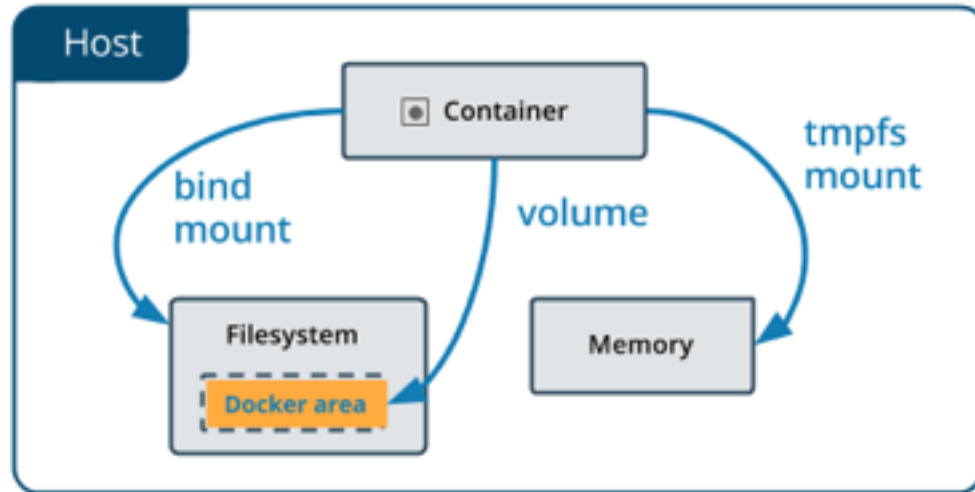
# Docker Volume Storage

- This is where persistent data lives
- Extremely pluggable
- Network attached storage is extremely useful here
  - For storage without a driver, mount NAS to `/var/lib/docker/volumes`
- Driver API is simple, easy to implement
  - [https://docs.docker.com/engine/extend/plugins\\_volume/](https://docs.docker.com/engine/extend/plugins_volume/)
- Supports both software- and hardware-based storage management



# Docker Volume Storage History

- Started with bind mounts
- A bind mount maps a directory on the host system to a mount point in the running container
- When the container shuts down, data in the bind mount directory remains
- `docker run --mount type=bind,source=/tmp/foo,target=/data containername`
- Developed into tmpfs
  - also volumes



## Docker Volume Storage Pt. 2 (Default behavior)

- Persistent data volumes are mapped to directories on Docker host
- `/var/lib/docker/volumes`
- `docker volume create myvol`
  - creates `/var/lib/docker/volumes/myvol`
- mounted into a container at runtime
  - `docker run --mount source=myvol,destination=/data mycontainer`
  - `docker run -v source=myvol,destination=/data mycontainer`
  - Note: `--mount` and `--volume/-v` are not 100% equivalent, but close



## Docker Volume Storage Pt. 3 (Drivers)

- Drivers must be installed before using
- v2 plugins:
  - `docker plugin install [OPTIONS] myplugin key=value`
  - `key=value` is passed to the plugin container for configuration as environment variables
- Plugin implements the Docker storage plugin API (more later)



# Docker Volume Plugin Concepts

- Plugin runs as a container (for v2 plugins)
- Binds to a local socket
  - TCP or local for v1
  - local only for v2
- Process is an http service that receives API requests from Docker Engine
- HTTP service triggers events that interact with storage service
- Container has config file that specifies required privileges & resources
  - <https://docs.docker.com/engine/extend/config/>
- 8 current verbs:
  - Create, Remove, Mount, Path, Unmount, Get, List, Capabilities  
(optional)



## Links

- <https://docs.docker.com/engine/extend/>
  - Plugin subsystem docs
- <https://github.com/docker/go-plugins-helpers>
  - Golang plugin stub
- <https://github.com/vieux/docker-volume-sshfs>
  - Example FUSE-based plugin
- <https://github.com/docker/vol-test>
  - Volume plugin testing suite
- <https://github.com/portworx/lcfs>
  - 3rd party graph driver by Portworx

