HEDVIG

Provisioning Hedvig Virtual Disks with Mesos and Docker Containers

Solution Whitepaper

Version 1.0

April 2017

Table of contents

Information sourced from: http://mesos.apache.org/documentation/latest/docker-volume/#how-does-it-work

Executive overview

Organizations are maturing data centers beyond virtualized infrastructure into true production clouds with distributed computing and dynamic provisioning. New solutions and architectures enable organizations to deliver applications to market faster to quickly capitalize on business opportunities. Distributed orchestration, computing, networking, and storage services built for modern business applications combine to simplify deployment and scaling across a cluster of commodity server-based infrastructure. The integration of Apache Mesos and Mesosphere orchestration platforms, with Docker containerization for applications and microservices, and Hedvig software-defined storage deliver easily composable infrastructure that enable enterprises to evolve and adapt services as requirements emerge.

Introduction

This document outlines how to integrate the Hedvig Distributed Storage Platform and its Virtual Disk abstraction with a Docker container-based environment in combination with Apache Mesos. As organizations move toward running "stateful" applications like MySQL or other databases in containers where maintaining business data for reuse beyond the life of a container is a critical requirement, an effective solution for automation in combination with resilient, dynamic, scalable storage becomes essential.

Docker Engine volume plugins enable Docker deployments to be integrated with external storage systems such as Hedvig, and enable data volumes to persist beyond the lifetime of a single Docker host. Mesos, and by extension Mesosphere DC/OS, take advantage of the Hedvig Docker volume plugin that utilizes the Docker *VolumeDriver* interface to ensure shared Virtual Disks and associated data are available to apps and microservices without user intervention even as containers are restarted and rescheduled on another host in a compute cluster.

With the combined solution enterprises can achieve:

- Faster time-to-market for new services
- Hyperscale efficiency, flexibility, and automation
- Seamless deployment, scaling, and failover of applications
- Enterprise-class storage services for modern applications
- Lower total cost of ownership

Solution components

Apache Mesos

Apache Mesos is a distributed systems kernel that abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems for container and non-container workloads to easily be built and run effectively. It enables the management of multiple machines as if they were a single pool of resources.

Mesosphere DC/OS

Mesosphere DC/OS, is a software solution that integrates Mesos and adds value-added services and functionality to simplify system operation and management. Mesosphere includes a toolset including a simple UI, security, advanced networking, and a set of command line-based tools for advanced configurations.

Docker

Docker is an infrastructure independent platform that automates the deployment of applications inside software containers, providing an abstraction layer and automation of operating-system-level virtualization. Docker containers isolate applications from other applications and the underlying infrastructure, packaging code and dependencies together to guarantee that software runs the same, regardless of where it's deployed.

Hedvig

The Hedvig Distributed Storage Platform is a software-defined storage solution that lets you tailor an elastic storage system built with low-cost commodity server hardware to support any application, hypervisor, container, or cloud. Hedvig is built with cloud-like provisioning and simplicity for infrastructure-as-a-service (IaaS) and plugs seamlessly into cloud orchestration solutions to deliver automated, on-demand storage provisioning for any scale.

# Integrating Hedvig with Mesos and Docker

To enable persistent storage for stateful applications with containers, Mesos 1.0 added Docker volume support to the MesosContainerizer. The Docker *VolumeDriver* API defines an interface between the container runtime and an external storage system such as Hedvig.

Within Mesos, a docker/volume isolator interacts with the Hedvig Docker volume plugin using dvdcli, an open-source command line tool. When a new task with Docker volumes is

launched, the docker/volume isolator invokes dvdcli to create and/or mount the corresponding Docker volume onto the host and then onto the container. When the task finishes, or is stopped, the docker/volume isolator will invoke dvdcli to unmount the corresponding Docker volume.
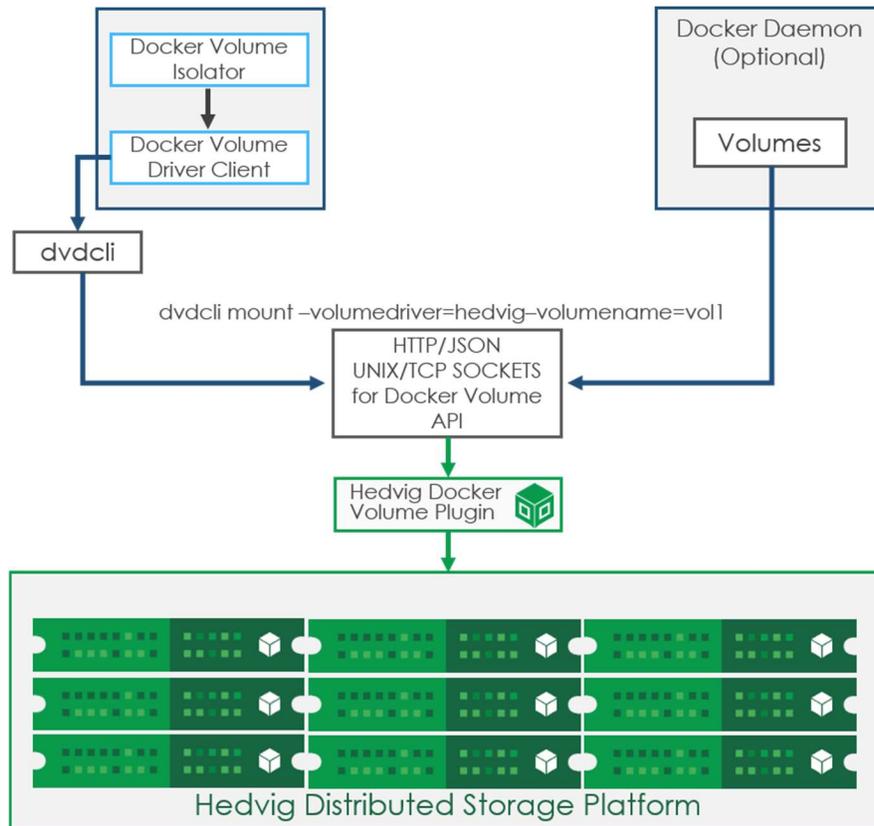
Workflow

Below is a detailed workflow for the volume process:

1. A framework specifies external volumes in ContainerInfo when launching a task.
2. The master sends the launch task message to the agent.
3. The agent receives the message and asks all isolators (including the docker/volume isolator) to prepare for the container with the ContainerInfo.
4. The isolator invokes dvdcli to mount the corresponding external volume to a mount point on the host.
5. The agent launches the container and bind-mounts the volume into the container.
6. The bind-mounted volume inside the container will be unmounted from the container automatically when the container finishes, as the container is in its own mount namespace.

5

7. The agent invokes isolator cleanup which invokes dvdcli to unmount all mount points for the container.

Configuration

To use the docker/volume isolator, the following steps are required to configure docker/volume isolator and specify the Docker volumes.

Pre-conditions

Install dvdcli version 0.1.0 on each agent.

Install the Hedvig Docker volume plugin on each agent.

Note: for more information on pre-requisite Hedvig software setup, please review the Hedvig Technical Overview Whitepaper.

You can either explicitly create the Docker volumes that are going to be accessed by Mesos tasks or volumes will be implicitly created by dvdcli. To ensure the volumes and corresponding Hedvig Virtual Disk fits the resource requirement, additional desired Virtual Disk policies can be indicated, otherwise "default" settings will apply.

Configuring Docker Volume Isolator

In order to configure the docker/volume isolator, configure two flags at agent startup as follows:

```
sudo mesos-agent \
  --master=<master IP> \
  --ip=<agent IP> \
  --work_dir=/var/lib/mesos \
  --isolation=filesystem/linux,docker/volume \
  --docker_volume_checkpoint_dir=<mount info checkpoint path>
```

The docker/volume isolator must be specified in the --isolation flag at agent startup; the docker/volume isolator has a dependency on the filesystem/linux isolator.

The --docker_volume_checkpoint_dir is an optional flag with a default value of /var/run/mesos/isolators/docker/volume. The docker/volume isolator will checkpoint all Docker volume mount point information under --docker_volume_checkpoint_dir for recovery. The checkpoint information under the default --docker_volume_checkpoint_dir will be cleaned up after agent restart. Therefore, it is recommended to set --docker_volume_checkpoint_dir to a directory which will survive agent restart.

How Mesos enables the use of Docker volumes

To enable the support of the Docker volume functionality, Mesos updated the Volume protobuf message as shown below. To use Hedvig with Mesos, it is unnecessary to modify anything with respect to the framework.

```
message Volume {
  ...

  required string container_path = 1;

  message Source {
    enum Type {
      UNKNOWN = 0;
      DOCKER_VOLUME = 1;
    }

    message DockerVolume {
      optional string driver = 1;
      required string name = 2;
      optional Parameters driver_options = 3;
    }

    optional Type type = 1;
    optional DockerVolume docker_volume = 2;
  }

  optional Source source = 5;
}
```

When requesting a Docker volume for a container, the framework developer needs to set Volume for the container, which includes mode, container_path and source.

The source field specifies where the volume comes from. Framework developers need to specify the type, Docker volume driver, name and options.

How to specify container_path:

1. If you are launching a Mesos container without rootfs. If container_path is an absolute path, you need to make sure the absolute path exists on your host root file system as the container shares the host root file system; otherwise, the task will fail.

2. For other cases like launching a Mesos container without rootfs and container_path is a relative path, or launching a task with rootfs and container_path is an absolute path, or launching a task with rootfs and container_path as a relative path, the isolator will help create the container_path as the mount point.

7

Examples of launching containers with Docker volumes

Following are two examples of launching tasks that mount Docker volumes for use by a container:

Launch a task with one Docker volume using the default command executor.

```
TaskInfo {
  ...
  "command" : ...,
  "container" : {
    "volumes" : [
      {
        "container_path" : "/mnt/volume",
        "mode" : "RW",
        "source" : {
          "type" : "DOCKER_VOLUME",
          "docker_volume" : {
            "driver" : "hedvig",
            "name" : "myvolume"
          }
        }
      }
    ]
  }
}
```

Launch a task with two Docker volumes using the default command executor.

```
TaskInfo {
  ...
  "command" : ...,
  "container" : {
   "volumes" : [
     {
       "container_path" : "volume1",
       "mode" : "RW",
       "source" : {
        "type" : "DOCKER_VOLUME",
        "docker_volume" : {
          "driver" : "hedvig",
          "name" : "volume1"
        }
       }
     },
     {
       "container_path" : "volume2",
       "mode" : "RW",
       "source" : {
        "type" : "DOCKER_VOLUME",
        "docker_volume" : {
          "driver" : "hedvig",
          "name" : "volume2",
          "driver_options" : {
           "parameter" : [{
             "key" : <key>,
             "value" : <value>
           }, {
             "key" : <key>,
             "value" : <value>
           }]
          }
        }
       }
     }
    ]
  }
}
```

Note: The task launch will be failed if one container uses multiple Docker volumes with the same driver and name.

■ ◗ ◢ ◈ ◆

Summary and conclusion

Infrastructure automation solutions like the Mesos/Mesosphere, Docker, and Hedvig solution outlined above deliver a new level of flexibility for provisioning applications and microservices along with required persistent storage services, ensuring critical applications and data are always available.

Hedvig's automation and programmability combined with the flexibility of Mesos/Mesosphere and the Docker volume framework ensures storage volumes are created, mounted and made available dynamically in fast-changing cloud environments. The combined solution enables an elastic and simple way for enterprises to realize the promise of digital business, improving business responsiveness and reducing costs through agile infrastructure.

Additional resources

Hedvig and Docker resources: http://www.hedviginc.com/docker-resources

Hedvig product information: http://www.hedviginc.com/product

Hedvig technical overview whitepaper: http://www.hedviginc.com/technical-overview-whitepaper

Hedvig datasheet: http://www.hedviginc.com/hs-fs/hub/448929/file-2663687738-pdf