

Issue

Universal Control Plane (UCP) provides a clustered interface to container volume management. The naming conventions used for volumes in UCP are different from a standalone node due to the clustered nature of the interface. They also vary according to the volume storage driver in use.

Prerequisites

- Universal Control Plane version 2.0, 2.1, 2.2, 3.0
- Universal Control Plane [client certificate bundle](https://docs.docker.com/ee/ucp/user-access/cli/) (<https://docs.docker.com/ee/ucp/user-access/cli/>)
- (optional) Container volume storage plugin, such as [Cloudstor](https://docs.docker.com/docker-for-aws/persistent-data-volumes/) (<https://docs.docker.com/docker-for-aws/persistent-data-volumes/>) or [NetApp](https://store.docker.com/plugins/trident) (<https://store.docker.com/plugins/trident>)

Resolution

The following two sections outline volume management in UCP using a `docker` command line client to manipulate volumes created first by the default `local` storage driver and then the [Cloudstor](https://docs.docker.com/docker-for-aws/persistent-data-volumes/) (<https://docs.docker.com/docker-for-aws/persistent-data-volumes/>) volume storage plugin. All steps are performed using a UCP client certificate bundle.

Managing Volumes with the `local` Storage Driver

This section provides examples of managing volumes created by the default `local` storage driver. All commands are issued through a UCP client certificate bundle.

Creating Driver: `local` Volumes

To create a local volume named `<^>mylocalvol<^^>` on every node (`volume ls` output in next section):

```
docker volume create <^>mylocalvol<^^>
```

Volumes created this way are backed by node-local storage. The content of each volume is not automatically synchronized between nodes.

Local volumes can also be created on a per-node basis by providing a node name prefix. To create a local volume named `<^>mylocalvol<^^>` *only* on node `<^>worker-1<^^>`:

```
docker volume create <^>worker-1<^^>/<^>mylocalvol<^^>
```

Listing Driver: `local` Volumes

To list all volumes with the name `<^>mylocalvol<^^>`:

```
docker volume ls -f name=<^>mylocalvol<^^>
```

Output:

DRIVER	VOLUME NAME
local	<^>manager-1<^^>/<^>mylocalvol<^^>
local	<^>manager-2<^^>/<^>mylocalvol<^^>
local	<^>manager-3<^^>/<^>mylocalvol<^^>
local	<^>dtr-1<^^>/<^>mylocalvol<^^>
local	<^>dtr-2<^^>/<^>mylocalvol<^^>
local	<^>dtr-3<^^>/<^>mylocalvol<^^>
local	<^>worker-1<^^>/<^>mylocalvol<^^>
local	<^>worker-2<^^>/<^>mylocalvol<^^>
[...]	

Each volume name is prefixed with the node name since the volumes are separate and local to each node. Implementation of a feature in UCP to enable filtering volume ls output by volume driver is in progress at github.com/docker/swarm/pull/2831 (<https://github.com/docker/swarm/pull/2831>).

Inspecting Driver: local Volumes

To inspect a local volume named <^>mylocalvol<^^> on node <^>worker-1<^^>:

```
docker volume inspect <^>worker-1<^^>/<^>mylocalvol<^^>
```

Output:

```
[
  {
    "Driver": "local",
    "Labels": {
      "com.docker.swarm.whitelists": "[\\"node==<^>manager-1<^^>|<^>manager-2<^^>|...\\"]",
      "com.docker.ucp.access.label": "/",
      "com.docker.ucp.collection": "swarm",
      "com.docker.ucp.collection.root": "true",
      "com.docker.ucp.collection.swarm": "true"
    },
    "Mountpoint": "/var/lib/docker/volumes/<^>mylocalvol<^^>/_data",
    "Name": "<^>mylocalvol<^^>",
    "Options": {},
    "Scope": "local"
  }
]
```

You must provide a node name prefix when inspecting local volumes, otherwise UCP returns a `No such volume` error:

```
$ docker volume inspect <^>mylocalvol<^^>
[]
Error: No such volume: <^>mylocalvol<^^>
```

Removing Driver: local Volumes

To remove the local volume <^>mylocalvol<^^> from all nodes in the cluster:

```
docker volume rm <^>mylocalvol<^^>
```

You can also remove local volumes on a per-node basis by specifying the node name as a prefix. If there are other volumes with the same name on other nodes, they will not be removed:

```
docker volume rm <^>worker-1<^^>/<^>mylocalvol<^^>
```

Managing Volumes Created by a Storage Plugin

This section provides examples of managing volumes created by the [Cloudstor](https://docs.docker.com/docker-for-aws/persistent-data-volumes/) volume storage plugin. All commands are issued through a UCP client certificate bundle.

Creating Storage Plugin Volumes

To create a Cloudstor volume named `<^>mypluginvol<^^>` that can be accessed from multiple nodes at once:

```
docker volume create \  
-d cloudstor \  
--opt backing=shared \  
<^>mypluginvol<^^>
```

You can also provide a node name prefix when creating plugin storage volumes (`<^>worker-1<^^>/<^>mypluginvol<^^>`). This has the same effect as creating the volume without a node name prefix.

Listing Storage Plugin Volumes

To list volumes with the name `<^>mypluginvol<^^>`:

```
docker volume ls -f name=<^>mypluginvol<^^>
```

Output:

```
DRIVER          VOLUME NAME  
cloudstor:latest <^>mypluginvol<^^>  
cloudstor:latest <^>mypluginvol<^^>  
cloudstor:latest <^>mypluginvol<^^>  
cloudstor:latest <^>mypluginvol<^^>  
[...]
```

Notes:

- Volume names are not prefixed with node names since they are not separate volumes.
- One volume is listed for each node that can access the volume.
- Implementation of a feature in UCP to enable filtering `volume ls` output by volume driver is in progress at github.com/docker/swarm/pull/2831 (<https://github.com/docker/swarm/pull/2831>).

Inspecting Storage Plugin Volumes

You can inspect plugin storage volumes by volume name:

```
docker volume inspect <^>mypluginvol<^^>
```

Output:

```
[
  {
    "Driver": "cloudstor:latest",
    "Labels": {
      "com.docker.swarm.whitelists": "[\"node==<^>manager-1<^>|<^>[...]<^>\"]",
      "com.docker.ucp.access.label": "/Shared/Private/<^>myuser<^>",
      <^>[...]<^>,
      "com.docker.ucp.collection.swarm": "true"
    },
    "Mountpoint": "/mnt/efs/reg/mypluginvol",
    "Name": "mypluginvol",
    "Options": {
      "backing": "shared"
    },
    "Scope": "local"
  }
]
```

Inspecting plugin storage volumes does not require node name prefix. Specifying a node name prefix provides the same result.

"Scope": "local" is a driver implementation detail distinct in meaning from `Driver`.

Removing Storage Plugin Volumes

To remove the plugin storage volume `<^>mypluginvol<^>` from the cluster:

```
docker volume rm <^>mypluginvol<^>
```

The volume is removed after a plugin-dependent amount of time, and will be visible via `docker volume ls` in the meantime.

Volumes can also be removed using a node name prefix. Since the Cloudstor plugin makes volumes available to the entire cluster, removing the volume from one node will also remove the volume from all nodes:

```
docker volume rm <^>worker-1<^>/<^>mypluginvol<^>
```

What's Next

- Use Volumes at [docs.docker.com \(https://docs.docker.com/storage/volumes/\)](https://docs.docker.com/storage/volumes/)
- Cloudstor at [docs.docker.com \(https://docs.docker.com/docker-for-aws/persistent-data-volumes/\)](https://docs.docker.com/docker-for-aws/persistent-data-volumes/)