Virtuozzo Storage 2.3

Docker Integration Guide

December 12, 2017

Virtuozzo International GmbH

Vordergasse 59

8200 Schaffhausen

Switzerland

Tel: + 41 52 632 0411

Fax: + 41 52 672 2010

https://virtuozzo.com

Contents

1. About This Guide **1**

2. Introduction **2**

2.1 Solution Overview 2

3. Prerequisites **3**

3.1 Hardware Requirements 3

3.2 Setting Up Virtuozzo Storage 3

3.3 Setting Up Docker 5

3.3.1 Configuring Firewall for Docker UCP 6

4. Setting Up Docker Volume Plug-in **8**

4.1 Verifying Virtuozzo Storage and Docker Installations 8

4.2 Installing Docker Volume Plug-in 9

4.3 Configuring Docker Volume Plug-in 10

5. Using Docker Volume Plug-in **11**

5.1 Docker Volume Creation Options 11

5.2 Creating Docker Volumes from Command Line 12

5.3 Using Docker Swarm Command Line 12

5.4 Using Docker UCP 13

5.5 Using Docker Compose 14

6. Addenda **15**

6.1 Miscellaneous Ploop Operations 15

6.1.1 Managing Docker Volume Snapshots 15

6.1.2 Resizing Ploops 16

6.1.3 Checking Ploop Status 16

CHAPTER 1

About This Guide

This guide describes how to use Virtuozzo Storage as persistent Docker volumes. It is intended for system administrators familiar with Linux and its command line.

CHAPTER 2

Introduction

Virtuozzo Storage is a software-defined storage solution that leverages commodity server hardware for building redundant, fault-tolerant, self-healing, and high-performance storage platforms. It is optimized for storing large amounts of data and serving multiple concurrent users.

Virtuozzo Storage is designed to consume a minimal amount of system resources to operate, effectively deploying storage service components to the servers which will have most of their compute power available to the applications. This allows you to place Docker compute and storage components on the same physical servers, making Virtuozzo Storage a perfect Docker storage solution that will enable you to transparently scale out your storage platform as your Docker platform grows and never require separate hardware for your data.

2.1 Solution Overview

The Docker plug-in described in this guide allows using Virtuozzo Storage as persistent storage for dockerized applications. The solution works as follows:

1. Virtuozzo Storage is installed and mounted on multiple nodes to create a redundant, highly available storage pool from local disks.
2. Docker is installed and a Swarm cluster is created on the same nodes.
3. The plug-in is installed to make Virtuozzo Storage volumes usable as Docker persistent volumes.
4. Using command line or Docker UCP, the user creates persistent volumes of needed size and provides them to dockerized applications.

CHAPTER 3

Prerequisites

3.1 Hardware Requirements

Virtuozzo Storage is a software-defined storage solution designed for multi-server cluster environments. The minimum number of servers required for Virtuozzo Storage is:

- three servers are a minimum for test and PoC deployments that do not involve performance benchmarking,
- five or more servers are recommended for performance testing and production.

> **Note:** While it is technically possible to deploy and use Virtuozzo Storage and Docker on just one server, such a setup will not let you fully explore the features of Virtuozzo Storage or use Docker Swarm.

Virtuozzo Storage can effectively scale up to hundreds of servers and terabytes of data. Several dozens of servers in a single storage cluster is a common production configuration.

Virtuozzo Storage performs best when SSD storage is used for journaling and data caching, in addition to rotational disks for cold data. For more details on the recommended hardware configuration, refer to the Virtuozzo Storage 2 Installation Guide or contact your sales representative.

3.2 Setting Up Virtuozzo Storage

Before proceeding with Docker installation and configuration, you need to have Virtuozzo Storage 2 installed and configured.

Virtuozzo Storage requires three physical nodes minimum, each with at least a quad-core CPU, 4GB RAM, 3 x 100GB HDD (one system, one MDS, one storage), 1 Gbps network interface. For better performance, it is recommended to install both system and MDS on an SSD and use HDDs only for storage. It is also recommended to have two network interfaces, as Virtuozzo Storage uses two networks: one private for storage and management traffic and one public for exporting storage data to clients.

To deploy Virtuozzo Storage, you will need to install the management panel and storage components on the first node and just the storage component on the second and other nodes.

1. Boot the first node from the installation media and make the required choices installation sections: set up date and time, choose the destination partition, check and/or configure network settings. In the **Virtuozzo Storage** section, choose **Management Panel and Storage**, select network interfaces for the private and public networks, and create a password.
2. After deploying the first node, log in to the management panel at that node's IP address and port 8888. The first node will be listed in the **UNASSIGNED NODES** section.
3. Click **ADD NODE** and obtain a token. You will need it to authorize and add the second and other nodes.
4. Boot the second node from the installation media and make the required choices installation sections: set up date and time, choose the destination partition, check and/or configure network settings. In the **Virtuozzo Storage** section, choose **Storage**, enter the IP address of the management panel, and the token.

   After deployment, the second node will appear in the **UNASSIGNED NODES** section.
5. Repeat the previous step on remaining nodes. After deployment, they will appear in the **UNASSIGNED NODES** section.
6. Having deployed all nodes, open the management panel and configure network interfaces of all nodes. Virtuozzo Storage uses two networks To do this, click node name > **NETWORK** > **Configure**. Check and/or configure network settings and select the internal **Storage** role (in addition to already selected, if any).
7. After roles are assigned, create a cluster. To do this, click a node in the **UNASSIGNED LIST** section, click **Create cluster**, enter cluster name, make sure that a network interface with the storage role is chosen, and click **New cluster**.

   You can monitor cluster creation progress in the **HEALTHY** list of the **NODES** screen.
8. Add other nodes to the cluster. To do this, click a node in the **UNASSIGNED LIST** section, click **Join cluster**, make sure that a network interface with the storage role is chosen, and click **Join cluster**.

As a result of installation, you should have a Virtuozzo Storage cluster deployed and configured across multiple nodes. You will get 1TB of storage space for free. To have more, you will need to obtain a license from the sales team and install it on the **LICENSES** > **Register key** screen. You can track the current space usage on the cluster **OVERVIEW** screen.

> **Important:** Do not install a trial license unless your cluster is intended for production. Once a trial license expires, the free 1TB will no longer be available.

For the complete installation instructions, see the Virtuozzo Storage 2 Installation Guide.

> **Note:** Docker containers may be shown in the Virtuozzo Storage web interface as regular disks. However, they are not intended to (and cannot) be managed via Virtuozzo Storage.

3.3 Setting Up Docker

Once Virtuozzo Storage is installed and configured, you need to install Docker on every cluster node. Depending on your needs, you may want to install a specific Docker edition:

- If you want to use Docker Swarm, do the following:
  1. Set up Swarm alongside Docker Community Edition (CE). Make sure to open the specified ports on each cluster node.
  2. Add all the servers in your Virtuozzo Storage cluster to your Docker Swarm cluster as explained here.
- If you need the commercial Universal Control Plane (UCP) in addition to Swarm, do the following:
  1. Open the required ports on each cluster node. See *Configuring Firewall for Docker UCP* on page 6.
  2. Install Docker Datacenter (DDC) based on Docker Enterprise Edition (EE).
  3. Once Docker UCP is deployed, make sure you add all the servers in your Virtuozzo Storage cluster to your Docker Swarm cluster via the **Add Node** screen in the **Nodes** menu in the UCP web interface.

  As a result of following these steps, you should have:

- A Docker Swarm cluster sharing the nodes with Virtuozzo Storage,
- Docker UCP (in case you have followed the Docker UCP installation instructions).

Important: Do not install Docker with `yum` as the version available in the default repositories is obsolete and does not support the required functionality (e.g., the `docker volume` command).

3.3.1 Configuring Firewall for Docker UCP

Before installing Docker UCP on a Virtuozzo Storage node, open the required ports in the firewall via the direct interface by means of the `/etc/firewalld/direct.xml` file (create it if needed).

On the controller node, make sure `/etc/firewalld/direct.xml` contains these lines:

```
<?xml version="1.0" encoding="utf-8"?>
<direct>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 443</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2375</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2376 </rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 12376:12390</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 443</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2375</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2376 </rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 7946</rule>
```

```
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 12376:12390</rule>
</direct>
```

On worker nodes, make sure `/etc/firewalld/direct.xml` contains these lines:

```
<?xml version="1.0" encoding="utf-8"?>
<direct>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 443</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2375</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2376 </rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv4" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 12376</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 443</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2375</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 2376 </rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 4789</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m udp -p udp -j ACCEPT
  --dport 7946</rule>
  <rule priority="0" table="filter" ipv="ipv6" chain="INPUT">-m tcp -p tcp -j ACCEPT
  --dport 12376</rule>
</direct>
```

Having created or edited the files, reload the firewall service on each node to apply changes:

```
# firewall-cmd --reload
```

After configuring the firewall, proceed to install Docker UCP.

CHAPTER 4

Setting Up Docker Volume Plug-in

It is assumed that you have Virtuozzo Storage cluster and Docker cluster up and running at this point.

4.1 Verifying Virtuozzo Storage and Docker Installations

To check that Virtuozzo Storage is configured, log in to its management panel and make sure the cluster status is **HEALTHY** on the **OVERVIEW** screen.

To check that Docker is running, run

```
# docker ps
```

You should see a list of running containers (Swarm or UCP instances that you have created). The example below shows output for a UCP master node.

```
# docker ps
CONTAINER ID  IMAGE                          COMMAND                  CREATED      STATUS      <...>
da9dc2b0abab  docker/ucp-controller:1.1.0    "/bin/controller serv"   3 weeks ago  Up 3 hours  <...>
9f678bae435f  docker/ucp-auth:1.1.0          "/usr/local/bin/enzi "   3 weeks ago  Up 3 hours  <...>
16c5f27599aa  docker/ucp-auth:1.1.0          "/usr/local/bin/enzi "   3 weeks ago  Up 3 hours  <...>
81df50bfd328  docker/ucp-auth-store:1.1.0    "/usr/local/bin/rethi"   3 weeks ago  Up 3 hours  <...>
4ae2a1516364  docker/ucp-cfssl:1.1.0         "/bin/cfssl serve -ad"   3 weeks ago  Up 3 hours  <...>
91a237c1009d  docker/ucp-cfssl:1.1.0         "/bin/cfssl serve -ad"   3 weeks ago  Up 3 hours  <...>
31d70d35335e  docker/ucp-swarm:1.1.0         "/swarm manage --tlsv"   3 weeks ago  Up 3 hours  <...>
f1e2680e2443  docker/ucp-swarm:1.1.0         "/swarm join --discov"   3 weeks ago  Up 3 hours  <...>
```

```
edb35d151e98  docker/ucp-proxy:1.1.0        "/bin/run"              3 weeks ago  Up 3 hours   <...>
9bf1ccad64ca  docker/ucp-etcd:1.1.0         "/bin/etcd --data-dir"  3 weeks ago  Up 3 hours   <...>
```

To check that UCP is running, visit its URL in a web browser (typically https://x.x.x.x:443/, where x.x.x.x is the IP address or hostname of the UCP master node). You should see the UCP login screen.

4.2 Installing Docker Volume Plug-in

To enable a Docker volume to be hosted on Virtuozzo Storage, you need to install the plug-in `docker-volume-ploop` on each node in the Docker cluster (as Virtuozzo Storage uses ploop to host filesystem images).

You can install and enable the plug-in with Docker:

```
# docker plugin install virtuozzo/ploop:1.0
```

> **Note:** If you need to set a custom home path instead of the default `/mnt/vstorage`, add the parameter `vstorage.source=<path>` to the command above.

Alternatively, you can build it from source hosted at GitHub:

1. Install the `ploop-devel` package:

   ```
   # yum install ploop-devel
   ```

2. Install Git and Go and configure the `GOPATH` environment variable:

   ```
   # yum -y install golang git
   # echo 'export GOPATH=$HOME/go' >> ~/.bash_profile
   # echo 'PATH=$GOPATH/bin:$PATH' >> ~/.bash_profile
   # . ~/.bash_profile
   ```

3. Install the plug-in:

   ```
   # go get github.com/virtuozzo/docker-volume-ploop
   ```

4. Install the configuration files:

   ```
   # cd $GOPATH/src/github.com/*/docker-volume-ploop && make install
   ```

After installing the plug-in either way, set redundancy for the Docker volume subdirectory. On any Virtuozzo Storage node, run the following command (given that you use the default Virtuozzo Storage mount path):

```
# vstorage set-attr -R /mnt/vstorage/dkv replicas=3
```

This command will set the normal number of replicas for the data in the subdirectory to 3 and the minimum number of replicas to 2. (For more details on data redundancy and high availability features of Virtuozzo Storage, see the Virtuozzo Storage 2 Administrator's Guide.)

4.3 Configuring Docker Volume Plug-in

The Docker volume plug-in is configured automatically during installation. If you need to change its parameters after installation, you can use the `docker plugin set` command.

For example, to change the home path to `/mnt/vstorage/docker`, run

```
# docker plugin set virtuozzo/ploop vstorage.source=/mnt/vstorage/docker
```

To set a default ploop size to 16GB, run

```
# docker plugin set virtuozzo/ploop args="-size 16GB"
```

CHAPTER 5

Using Docker Volume Plug-in

5.1 Docker Volume Creation Options

When creating a Docker volume in CLI, GUI, or docker compose file, you need to specify ploop as the volume driver. This is the only requirement. In addition, you can specify various optional volume parameters. The following parameters are currently supported:

Parameter	Description
size	Volume size. Example: `size=10G` for a volume on which one can store up to 10GB of data.
tier	Virtuozzo Storage tier to use for the volume. Tier 0 is the fastest, and tier 3 is the slowest. Do not use this option if you have not configured tiers in your Virtuozzo Storage installation. Example: `tier=2`.
mode	For advanced users only. The default value should work fine in most cases. Ploop image mode (format) to use for the volume. Possible values are `expanded`, `preallocated`, and `raw`. • `expanded` (default), the image will grow according to the needs of the underlying file system (thin provisioning). • `preallocated`, same as `expanded`, except all the file blocks are allocated during creation. • `raw`, 1:1 mapping between the image file and the device it represents. Example: `mode=preallocated`.

Parameter	Description
clog	For advanced users only. The default value should work fine in most cases. Log2 of ploop cluster block size, in 512-byte sectors. Default is 1M cluster block size, which corresponds to `clog=11` (i.e. 2^11 = 2048 sectors, 2048*512 = 1M). Sensible values are in the range from 7 to 13. Example: `clog=12`.

5.2 Creating Docker Volumes from Command Line

To create a new Docker volume from the command line, run

```
# docker volume create -d virtuozzo/ploop -o size=512G --name <volume>
```

To run a Docker container with the volume from the command line, run

```
# docker run -it -v <volume>:/<mount> alpine /bin/ash
```

Here `<volume>` is the volume name, e.g., `MyVol`, and `<mount>` is the path under which the volume will be available inside a container, e.g., `/media/vol`.

Note: For other operations on volumes, see `man docker volume`.

5.3 Using Docker Swarm Command Line

Make sure you can connect to the Swarm manager. In this case, the `docker info` command should show all your nodes.

Below is an example for Swarm installed along with UCP (if you have installed Swarm manually, make sure to appropriately set environment variables, e.g., `DOCKER_HOST`):

```
# export DOCKER_HOST=tcp://127.0.0.1:2376
# export DOCKER_TLS_VERIFY=true
# export DOCKER_CERT_PATH=/var/lib/docker/volumes/ucp-auth-api-certs/_data
# docker info | grep Nodes
```

```
Nodes: 3
```

Now you can use all the Docker commands as usual, with the difference that they will see all the containers and volumes from all the nodes.

One peculiarity is that every Virtuozzo Storage volume will be listed as many times as the number of nodes you have, for example:

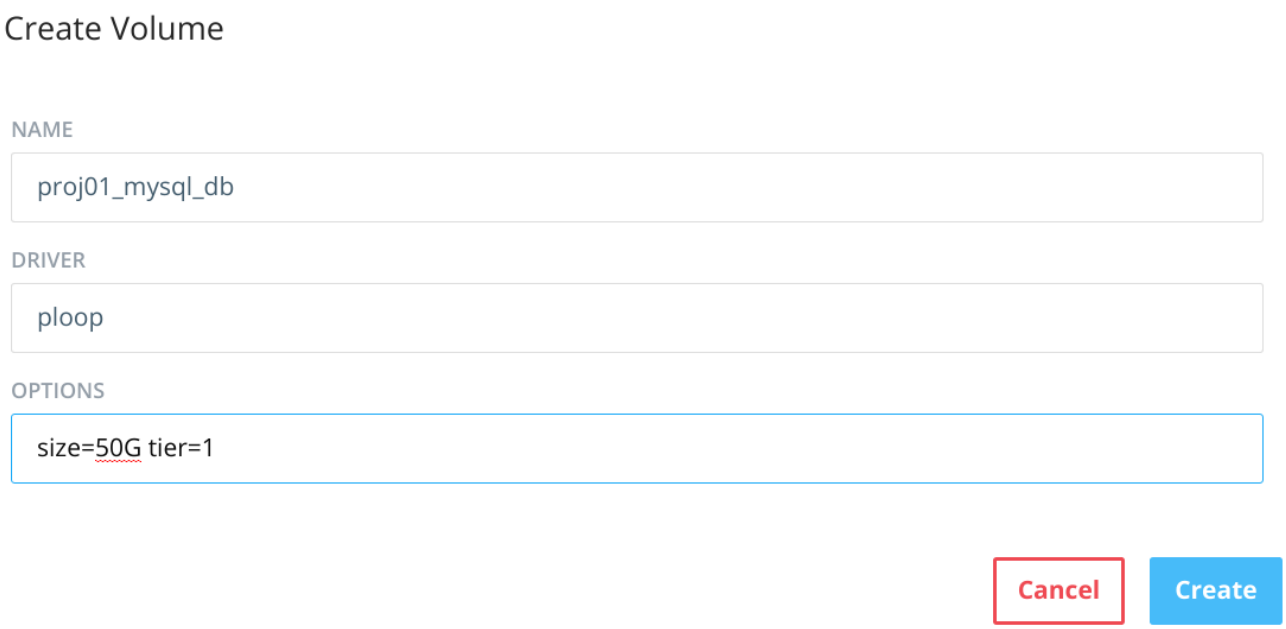
```
# docker volume ls

    ...
    ploop               wp01_mysql
    ploop               wp01_mysql
    ploop               wp01_mysql
    ...
```

This is a known issue with Swarm, see https://github.com/docker/swarm/issues/1970 for details.

5.4 Using Docker UCP

In Docker UCP, you can create, list, and use Virtuozzo Storage volumes like any other volumes. Here is an example of creating a volume:



Note: Docker UCP may show multiple entries for a volume: as many as there are nodes where the volume's driver is installed. These multiple entries are still treated as one and the same volume.

5.5 Using Docker Compose

To use Virtuozzo Storage volumes, you need to have the `driver: ploop` line in the `docker-compose.yml` file. Here is an example:

```
version: '2'
volumes:
    wordpress:
        driver: ploop
        driver_opts:
            size: 50G
            tier: 1
...
services:
    wp:
        image: wordpress
        working_dir: /var/www/html
        volumes:
         - wordpress:/var/www/html/wp-content
...
```

CHAPTER 6

Addenda

6.1 Miscellaneous Ploop Operations

The following is the quick introduction of what operations can be performed with ploop images. For more details about ploop, see https://openvz.org/Ploop.

When using the `ploop` command-line tool, you need to refer to ploop images by path to its DiskDescriptor.xml file. The `docker-volume-ploop` driver creates images under `img` subdirectory of its home (`$DKV_PLOOP_HOME`). So, to use the following example commands, you need to `cd` to the image directory, for example:

```
# cd /pcs/img/MyFirstVol/
```

6.1.1 Managing Docker Volume Snapshots

To create a snapshot, run

```
# ploop snapshot DiskDescriptor.xml
```

To list snapshots, run

```
# ploop snapshot-list DiskDescriptor.xml
```

To delete a snapshot, run

```
# ploop snapshot-delete -u <UUID> DiskDescriptor.xml
```

To mount a snapshot (read-only), run

```
# ploop mount -r -u <UUID> -m <mount_point> DiskDescriptor.xml
```

6.1.2 Resizing Ploops

To resize a running or stopped volume, run

```
# ploop resize -s <size> DiskDescriptor.xml
```

6.1.3 Checking Ploop Status

In case of issues with a ploop image (e.g., it cannot be mounted, may be corrupted, etc.), you can check its status as follows:

```
# ploop check DiskDescriptor.xml
```

If you want to run fsck on ploop's inner filesystem, run

```
# ploop mount -F DiskDescriptor.xml
```

Then make sure to unmount it with

```
# ploop umount DiskDescriptor.xml
```