

# DVX Linux Virtualization Setup Guide

Linux Hyperdriver Setup and Administration



# Table of Contents

<b>1 Introduction</b>	<b>4</b>
<b>1.1 Version History</b>	<b>4</b>
<b>2 DVX Overview</b>	<b>5</b>
<b>2.1 DVX Compute Nodes</b>	<b>5</b>
<b>2.2 Linux Host Hardware Requirements</b>	<b>5</b>
<b>2.3 Linux Host Software Requirements</b>	<b>5</b>
<b>3 Linux Host Preparation</b>	<b>6</b>
<b>3.1 DVX Data Network</b>	<b>7</b>
<b>3.2 Identifying DVX Flash Devices</b>	<b>8</b>
<b>3.3 DVX Software Install</b>	<b>9</b>
<b>3.4 Services and Settings</b>	<b>10</b>
3.4.1 Firewalls	10
3.4.2 SELinux	10
3.4.3 IPTables	11
<b>3.5 DVX Hyperdriver</b>	<b>11</b>
<b>3.6 Automatic Startup and Mounting</b>	<b>12</b>
<b>4 Managing the Datrium Datastore Mount Points</b>	<b>12</b>
<b>4.1 Multiple Datastores</b>	<b>12</b>
<b>4.2 Storage Domains</b>	<b>13</b>
<b>4.3 SetupLinuxHost Utility</b>	<b>14</b>
<b>5 Insane Mode</b>	<b>15</b>
<b>5.1 Enabling Insane Mode</b>	<b>16</b>
<b>5.2 Disabling Insane Mode</b>	<b>16</b>
<b>6 Removing DVX Host Software</b>	<b>16</b>
<b>6.1 Services and Settings</b>	<b>16</b>
<b>6.2 Removing Software</b>	<b>17</b>
<b>6.3 Cleaning Up Data Node</b>	<b>17</b>
<b>7 Known Issues</b>	<b>17</b>
<b>7.1 SSD access after reboot</b>	<b>18</b>
<b>7.2 NFS locking</b>	<b>18</b>
<b>7.3 Working with DVX ISO Storage Domain</b>	<b>19</b>

## *List of Tables and Figures*

<i>Table 1</i>	<i>6</i>
<i>Figure 1</i>	<i>8</i>
<i>Figure 2</i>	<i>14</i>

# 1 Introduction

The Datrium DVX system provides an enterprise level platform for running Linux virtualization. This Setup Guide will cover the basics of configuring Linux virtualization hosts for Datrium DVX.

This Setup Guides is intended for solution designers, system architects, or systems administrators that are looking to deploy Datrium DVX systems with Linux virtualization configurations.

The purpose of this Setup Guide is to cover enough details about the initial setup and configuration of the Datrium DVX to facilitate adding Linux systems for today's virtualized private cloud data centers solutions. Readers should be familiar with the basics of the DVX system, Linux operating system and Linux virtualization components.

The font formats used in this Setup Guide are intended to help with the steps involved. In particular, these fonts are used for different purposes:

- *Show steps that need to be entered at the CLI*
- Show typical output expected from commands
- *NOTE: When other considerations are important*

## 1.1 Version History

Version	Date	Author	Notes
1.0	2/2/2019	mmclaughlin@datrium.com	Initial Release

## 2 DVX Overview

The Datrium DVX system is comprised of one or more Compute Nodes equipped with local flash storage that run the Datrium software. These Compute Node servers are connected to one or more Data Nodes providing the persistent primary and backup copy of the data in the storage system.

The Datrium software running on the Compute Node servers is installed along with a Linux operating system and virtualization configuration. The supported Linux configurations are:

- Red Hat Enterprise Linux (RHEL)
- Red Hat Virtualization (RHVH/RHVM)
- CentOS Linux
- CentOS KVM (oVirt)

This document will cover the setup for the Linux virtualization solutions on both Red Hat and CentOS. The Datrium DVX also supports the VMware ESX hypervisor virtualization platform.

### 2.1 DVX Compute Nodes

The DVX System includes a DVX Hyperdriver software component for the Linux virtualization host for Red Hat and CentOS environments. The following sections describe host setup and DVX Hyperdriver installation and use. This software gets installed and configured onto the Linux host. The Linux host should also be configured with local flash devices (SSD, NVMe, etc.) that are dedicated for use by Datrium DVX system.

### 2.2 Linux Host Hardware Requirements

The current server requirements for running Datrium DVX software are available here: <https://www.datrium.com/resources/dvx-compute-node-specifications/>

In short, the server needs to have CPU resources to run the DVX software, 10G network data connections and at least one SSD device to provide the local host flash storage capabilities. Servers can be acquired from Datrium or you can use 3rd party servers (existing or new) if they meet the hardware requirements define in the link above. This document will cover basics on setting up the network and SSD for use with the DVX software.

### 2.3 Linux Host Software Requirements

A minimal Linux server or hypervisor host setup distribution will typically contain all of the packages that are required to use the Linux-based DVX Hyperdriver. Depending on actual installation distribution used there may be some additional packages that need to be installed. The following table describes the required packages:

Software	Comments
Linux Kernel 3.10 or later	RHEL 7.3+, RHVH 4.1+, CentOS 7.3+, oVirtNode 4.1+ distributions - minimal level
Python 2.7.3 or later	typically included with base installation
sgdisk	from the gdisk package
nfs-utils, rpcbind	The DVX Hyperdriver starts a local NFS server to service I/O. NFS traffic does not cross the storage network, but the shared DVX Datastore must be mounted
libvirt-daemon, libvirt-python	The DVX Hyperdriver registers the default DVX datastore as a libvirt storage pool named <b>“Datastore1”</b>
systemd	The DVX Hyperdriver installs a systemd script at /usr/lib/systemd/system/datrium.service. The service is configured to startup automatically at boot time

Table 1

### 3 Linux Host Preparation

Since there are several ways to get a host configured with the requisite Red Hat or CentOS Linux OS and virtualization capabilities, we will defer the actual host OS installation and preparation details to the site administrator. Follow the recommendations and practices for your organization and subscription method to obtain the basic Linux requirements with a minimal install along with the appropriate virtualization packages identified in the software requirements section above.

Once the system has been installed with all of the components, you may want to run one last update before joining the host to a virtualization management system (e.g., RHVM or oVirt).

*NOTE: When configuring for Red Hat, be sure to have the appropriate Red Hat subscription and repos enabled for the packages needed to complete the setup. It is possible to configure Red Hat virtualization separately (e.g., RHVH) or add it to an already configured base Red Hat (RHEL) host. The choice of approach is left to the administrator.*

Once the base OS and virtualization capabilities have been installed, there are a couple of configuration steps that can be performed before installing and configuring the Datrium DVX software.

Enable and start the libvirt capabilities:

```
systemctl start libvirtd
systemctl enable libvirtd
```

Disable the NFS server. The Datrium DVX software will present itself as an NFS based mount partition but will do this through special software mechanisms using an nfsbind setup discussed later in this guide. The only way to actually access the NFS storage presentation is through the DVX host based software.

```
systemctl stop nfs-server
```

```
systemctl disable nfs-server
```

### 3.1 DVX Data Network

For proper setup of the DVX system, a separate 10G data network is recommended. The data network may be either logically or physically separated. Follow best practices for your particular site requirements to configure connectivity from the Linux host to the DVX Data Node data interfaces. For more detailed information about DVX networking, refer to the System Management Guide. The Networking Best Practices on the Datrium Resources web site (<https://www.datrium.com/resources/>) is also a good resource even though its current version is primarily for VMware configurations.

In some situations, it may be easier to configure the Linux host management (ovirtmgmt) interfaces before joining the host to a common management infrastructure like RHVM or oVirt. In all cases, ensure that the host can route to the Data Node floating and fixed data IP addresses. For example:

```
ping <IP address of Data Node floating or fixed NIC>
```

It is also good practice to validate that the DVX Data Node can reach the data network interfaces of the Linux hosts. This can be tested through the DVX Data Node CLI using the following command:

```
network tools ping <IP address of Linux host data NIC>
```

Once the data network between the Linux host(s) and DVX Data Node has been established, make note of the floating data IP address (or FQDN ?) of the Data Node as this will be used in subsequent steps of the software installation and storage setup.

As an example, the final network setup configured in RHVM Administration Portal with data, management, and VM networks might look something like this:

▲ Name	Description	Role	VLAN tag	Label	MTU
DATA	DVX Data Network		47	DATA	Default (1500)
ovirtmgmt	Management Network		210	MGMT	Default (1500)
VM	VM Network		235	VM	Default (1500)

Figure 1

## 3.2 Identifying DVX Flash Devices

In order to properly configure the DVX Linux host software, at least one local host flash device must be provisioned for dedicated use by the DVX system. If using a 3rd Party Compute Node configuration, refer to the hardware specifications for SSD size and quantity options. It is not recommended to use a single SSD device for multiple purposes such as OS and DVX. Datrium also recommends using at least two SSD devices to improve resiliency of the host configuration.

Through a combination of the following commands you should be able to identify the specific paths to the target SSDs to include into the DVX configuration setup. Determine which devices are visible and how they are being used (e.g., system device vs DVX flash device):

```
lsblk
```

Example host listing for device sda looks like this:

```
# lsblk
NAME                                MAJ:MIN    RM    SIZE    RO  TYPE  MOUNTPOINT
sda                                  8:0        0    894.3G  0   disk
sda1                                  8:1        0    894.3G  0   part
35002538a071d5290                    253:1     0    894.3G  0   mpath
35002538a071d5290p1                  253:5     0    894.3G  0   part
```

Determine the host SCSI path to each device(s) to be used for DVX host data caching:

```
ls -o /dev/disk/by-id/scsi*
```

The path to the SSD device using the SCSI ID is the desired information. Make note of this information for subsequent configuration setup use. For example we see for device **sda** something like the following:

```
# ls -o /dev/disk/by-id/scsi*
lrwxrwxrwx 1 root  9 Dec 27 13:00 /dev/disk/by-id/scsi-35002538a071d5290 -> ../../sda
lrwxrwxrwx 1 root 10 Dec 27 13:00 /dev/disk/by-id/scsi-35002538a071d5290-part1 -> ../../sda1
```

The value “/dev/disk/by-id/scsi-35002538a071d5290” will be used to indicate the device found at **sda**.

*NOTE: if reusing or reconfiguring SSD devices, it may be necessary to delete any existing partitions and data sets on the target SSD devices. You can use fdisk to do this if needed. The DVX setup will create the necessary partitions on the devices.*

### 3.3 DVX Software Install

This section will cover the steps necessary to manually install the DVX software from the shell command line method on the Linux server.

*NOTE: You will need to use the Linux host CLI to add each Linux host to the DVX System.*

For this example, we use the curl utility to download the Datrium setup script (SetupLinuxHost.py) from the Data Node and then run the software installation and initial configuration. The SetupLinuxHost.py script is used to install the Datrium Hyperdriver. This setup utility will be discussed in more detail later in this guide. The following example sets up a host that will use one local SSD for DVX operations.

- Specify the Floating IP Address for the Data Network on the Data Node. The example uses the IP address 192.168.47.2. The Datrium software setup uses the argument “--datanode” to identify the Data Node.
- Specify the path to at least one SSD device. It is strongly recommended that you specify persistent /dev/disk/by-id/ paths to the devices. A simple way to do this is described earlier in this document. The SetupLinuxHost.py utility, discussed later, will also resolve the paths accordingly once the devices are identified.

The following example shows the configuration of one SSD for DVX use with Data Node at 192.168.47.2

```
curl http://192.168.47.2/static/SetupLinuxHost.py | python --datanode 192.168.47.2 \
-device /dev/disk/by-id//dev/disk/by-id/scsi-35002538a071d5290
```

For your particular environment, substitute the Data Node floating data IP for 192.168.47.2 entries and the SSD details (e.g., scsi-35002538a071d5290) with the device identifier for this Linux host. This information should have been captured in earlier steps.

*NOTE: This example uses the “--datanode” argument. For environments before DVX 4.0.1.x please contact Datrium support for alternate syntax.*

## 3.4 Services and Settings

This section covers various services and startup settings needed to enable the DVX software to run on this host.

### 3.4.1 Firewalls

The preferred approach for firewall configuration is to open the ports defined in `/etc/firewalld/services/datrium.xml` that was installed as part of the initial setup step:

```
firewall-cmd --reload  
firewall-cmd --add-service=datrium  
firewall-cmd --permanent --add-service=datrium
```

As an alternative, you can choose to disable any firewalls between the Linux host and the DVX Data Node. The following example uses `firewalld`:

```
systemctl stop firewalld  
systemctl disable firewalld
```

If you are not using `firewalld`, you may need to examine the file `/etc/firewalld/services/datrium.xml` and configure the firewall manually.

### 3.4.2 SELinux

Before starting the Datrium DVX software, you will need to check and disable SELinux if enabled. To check on current configuration of SELinux, run one of the following commands:

```
sestatus  
getenforce
```

The output of these commands when SELinux is disabled will look something like this:

```
SELinux status:                disabled  
Disabled
```

One method of changing the status to disabled is to use the following command and then reboot the host for the change to take effect. After making this change and rebooting the host it is always good practice to check the SELinux status as described above.

```
sed -i -e 's/^SELINUX=.*SELINUX=disabled/' /etc/sysconfig/selinux /etc/selinux/config  
reboot
```

### 3.4.3 IPTables

If your site configuration is still using IPTables instead of firewall configurations, it will be necessary to fix IP tables for DVX access. To do this, run the following commands to configure / reconfigure proper access – particularly after reboots or other system configuration changes:

```
/opt/datrium/bin/SetupIPTables.bash  
iptables-save > /etc/sysconfig/iptables  
systemctl restart iptables
```

## 3.5 DVX Hyperdriver

After the initial configuration with SetupLinuxHost.py and the related services, follow these steps to start the Datrium host service. The setup will also create the default “**Datastore1**” mount point to the DVX storage system.

Start the Hyperdriver service and check the status:

```
systemctl start datrium  
systemctl status datrium
```

The output should have key status desired as **Loaded** and **Active**. It will look something like the following:

```
datrium.service - Datrium Storage Service  
Loaded: loaded (/usr/lib/systemd/system/datrium.service; enabled; vendor preset:  
disabled)  
Active: active (running) since Tue 2017-04-18 14:31:55 PDT; 23s ago  
Process: 6458 ExecStartPre=/opt/datrium/bin/SetupLinuxHost.py --upgrade-if-  
necessary (code=exited, status=0/SUCCESS)  
Main PID: 6461 (procmgr.py)
```

If the hyperdriver starts as expected, then mount the (default) Datrium datastore “Datastore1” manually as follows:

```
systemctl start mnt-datrium-Datastore1.mount
```

Verify that the service is running correctly:

```
systemctl status mnt-datrium-Datastore1.mount
```

The output should have key status desired as **Loaded**. It will look something like the following:

```
mnt-datrium-Datastore1.mount - Datrium Datastore1 Mount Point  
Loaded: loaded (/etc/systemd/system/mnt-datrium-Datastore1.mount; enabled;  
vendor preset: disabled)  
Active: active (mounted) since Sat 2018-12-15 08:57:45 PST; 2 weeks 0 days ago
```

```
Where: /mnt/datrium/Datastore1
What: localhost:/192.168.47.62:2181/Datastore1
Tasks: 0
Memory: 120.0K
```

## 3.6 Automatic Startup and Mounting

It is possible to have the Datrium DVX software and datastore mounting operations enabled to start with the host.

*NOTE: Because of a known Linux OS issue with SSD discovery on boot (see Known Issues section), we are not recommending automatic startup of Datrium services or mount points at this time.*

To disable automatic mounting:

```
systemctl disable mnt-datrium-Datastore1.mount
```

To enable automatic mounting:

```
systemctl enable mnt-datrium-Datastore1.mount
```

# 4 Managing the Datrium Datastore Mount Points

The Datrium DVX storage mount point(s) for this Data Node can be shared with other hosts that have been configured with the Datrium DVX software for the same DVX system. The datastore mount point can also be subdivided to support Linux virtualization Storage Domains. With DVX version 4.1 or later, it is also possible to create multiple datastore mount points to the same DVX.

## 4.1 Multiple Datastores

In order to configure additional datastores to the Linux hosts, the datastore must first be created on the Data Node. Refer to the latest Datrium System Management Guide for more details. Once the additional datastore has been created, it can be added to the Linux host access with `SetupLinuxHost.py` and then configured into the system just like the default datastore “**Datastore1**”. Use this command to configure the host to the additional datastore:

```
SetupLinuxHost.py --datastore <name of added datastore>
```

You should see various output messages about the environment and existing datastores as well as a final message about the new entry such as:

```
SetupLinuxHost.py ['--datastore', '<name of added datastore>'] exited with
ReturnCode: None
```

It is useful to note that the primary (default) datastore “**Datastore1**” will have a slightly different internal host path representation than the added datastores constructs. This is important in the registration of the paths to the datastores when creating Storage Domain entries. To see the actual host path used to map each datastore, you can use the following command:

```
systemctl show mnt-datrium-Datastore1.mount -p What
```

```
systemctl show mnt-datrium-Datastore2.mount -p What
```

The output of this command will look something like this:

```
What=localhost:/192.168.47.62:2181/Datastore1  
What=localhost:/192.168.47.62:2181/3e190c38-dc48-11e8-9e3e-f353238cf3ac
```

## 4.2 Storage Domains

The DVX datastore mount point on each host can be presented as a Storage Domain to a shared multi-host management layer like RHVM or oVirt. The DVX software will be using an nfsbind style presentation method when configuring the Storage Domain(s). In order to facilitate this method, copy the provided file to the appropriate location:

```
cp /opt/datrium/examples/mount.nfsbind /sbin/
```

You can mount sub-directories of "**Datastore1**" as separate storage, for example Data, ISO, or Export storage domains. Determine the actual mounted folder for the target datastore. For example, using the df command you can list the Datrium specific storage targets:

```
df-h | grep -i "/mnt/datrium"
```

Will produce output similar to this:

```
localhost:/192.168.47.62:2181/Datastore1          14T  1.2T  12T  
9% /mnt/datrium/Datastore1  
localhost:/192.168.47.62:2181/3e190c38-dc48-11e8-9e3e-f353238cf3ac  14T  1.2T  12T  
9% /mnt/datrium/Datastore2
```

From this listing we can see two datastores mounted at /mnt/datrium and the actual DVX host path that will be used later when establishing the Storage Domain entities:

Mount Point	Host Path (prefix)
/mnt/datrium/Datastore1 (default)	localhost:/192.168.47.62:2181/Datastore1
/mnt/datrium/Datastore2 (additional)	localhost:/192.168.47.62:2181/3e190c38-dc48-11e8-9e3e-f353238cf3ac

The following example creates a subdirectory for a data storage domain called RHVDATA in datastore "**Datastore1**" and sets the appropriate ownership and permissions (36:36)

```
mkdir /mnt/datrium/Datastore1/RHVDATA
```

```
chown vdsm.kvm /mnt/datrium/Datastore1/RHVDATA
```

```
chmod 755 /mnt/datrium/Datastore1/RHVDATA
```

Once the subdirectory folder has been configured, it is then possible to add this folder as a Storage Domain (e.g., in RHVM or oVirt Administration Portal). When adding the Storage Domain use these selections as then shown in the figure below:

- **Storage Type:** POSIX compliant FS
- **VFS Type:** nfsbind
- **Path:** use the full host path to the mount point determined above when creating the Storage Domain directory path
- The other inputs will be site specific and up to the administrator

The screenshot shows a 'New Domain' configuration window. The fields are as follows:

Data Center	Sandbox (V4)	Name	DATA
Domain Function	Data	Description	DATA
Storage Type	POSIX compliant FS	Comment	
Host to Use	n1516		
Path	localhost/192.168.47.62:2181/Datastore1/RHVDATA E.g.: /path/to/my/data		
VFS Type	nfsbind		
Mount Options			

Figure 2

### 4.3 SetupLinuxHost Utility

The DVX host software includes a configuration utility for managing the DVX host configuration. The syntax for this utility is listed here:

```
# SetupLinuxHost.py --help
usage: SetupLinuxHost.py [-h] [--download DOWNLOAD] [--version VERSION]
                        [--is-version-downloaded IS_VERSION_DOWNLOADED]
                        [--switch-version SWITCH_VERSION] [--force]
                        [--upgrade-if-necessary] [--install] [--show]
                        [--wait-for-fe] [--fe-crash-disabled]
                        [--run-preinstall-check RUN_PREINSTALL_CHECK]
                        [--disruptive-upgrade] [--force-upgrade]
                        [--libvirt LIBVIRT] [--datanode DATANODE]
                        [--port PORT] [--datastore DATASTORE]
                        [--device DEVICE] [--hostMgmtIp HOSTMGMTIP]
                        [--partition PARTITION] [--insane | --no-insane]
                        [--DaLog.enableStderr]
                        [--DaLog.logFileName LOGFILENAME]
                        [--DaLog.logLevel LOGLEVEL]
```

Linux Host bootstrap, upgrade, and configuration script.

optional arguments:

```

-h, --help                show this help message and exit
--download DOWNLOAD      URL of HyperDriver bundle to download.
--version VERSION        Version of HyperDriver bundle being downloaded.
--is-version-downloaded IS_VERSION_DOWNLOADED
                          See if the given version has been downloaded.
--switch-version SWITCH_VERSION
                          Set target version and initiate shutdown.
--force                  Force shutdown.
--upgrade-if-necessary
                          Upgrade the HyperDriver if necessary.
--install                (Re)install latest HyperDriver bundle.
--show                  Show current configuration.
--wait-for-fe           Wait for FE to be ready.
--fe-crash-disabled     Returns 0 if FE is crash disabled.
--run-preinstall-check RUN_PREINSTALL_CHECK
                          URL of host preinstall check script to run.
--disruptive-upgrade    Disruptive upgrade.
--force-upgrade         Force upgrade.
--libvirt LIBVIRT       Libvirt connection URI.
--datanode DATANODE     Master Data Node floating data IP address.
--port PORT             Master Data Node ZooKeeper port.
--datastore DATASTORE
                          Datastore(s) to mount.
--device DEVICE         Path to SSD device(s) (e.g. /dev/sdb).
--hostMgmtIp HOSTMGMTIP
                          Host management IP address.
--partition PARTITION
                          Path to FStore partition(s) (e.g. /dev/sdb1).
--insane                Enable insane mode.
--no-insane             Disable insane mode.
--DaLog.enableStderr   Log to stderr.
--DaLog.logFileName LOGFILENAME
                          Log file.
--DaLog.logLevel LOGLEVEL
                          Log level.

```

There are a couple of key uses for the setup utility after initial installation. This includes:

- Reviewing the current status with the `--show` option
- Reconfiguring or adding more SSD devices with the `--device` option
- Adding additional datastore with the `--datastore` option
- Changing Insane mode setting with `--insane` or `--no-insane` option
- Managing software installation or updates with the `--install` option and others

It is best to work with Datrium Technical Support if you run into any issues or questions about managing the DVX host configuration with `SetupLinuxHost.py` utility.

## 5 Insane Mode

To determine if Insane mode is enabled on the Compute Node, login to the CLI of the Linux host and run this command looking for the Insane Mode status:

```
/opt/datrium/bin/SetupLinuxHost.py --show
```

Example output if enabled:

```
Insane mode: Enabled
```

Example output if disabled:

```
Insane mode: Disabled
```

## 5.1 Enabling Insane Mode

To enable Insane mode on each Linux host, follow these steps from the Compute Node CLI:

```
/opt/datrium/bin/SetupLinuxHost.py --insane  
systemctl restart datrium
```

## 5.2 Disabling Insane Mode

To disable Insane mode on each Linux host, follow these steps from the Compute Node CLI:

```
/opt/datrium/bin/SetupLinuxHost.py --no-insane  
systemctl restart datrium
```

# 6 Removing DVX Host Software

The following steps will remove the Linux host from the DVX system and clean up the DVX software installation. These steps assume you have removed the DVX storage dependencies throughout the Linux / KVM layers that may have been configured to the DVX. This includes shared Storage Domains and any Pool (virsh style) definitions.

## 6.1 Services and Settings

To remove the DVX software and configuration from the Linux host, log in to the Compute Node and performing the following steps.

Stop and Disable any Datrium mount point services for each defined datastore (for example “**Datastore1**”):

```
systemctl stop mnt-datrium-Datastore1.mount  
systemctl disable mnt-datrium-Datastore1.mount
```

Disable and stop any Datrium host specific services:

```
systemctl stop datrium  
systemctl disable datrium
```

NOTE: For DVX 4.1 or later, if multiple datastores have been configured, then disable and stop services for each additional datastore.

## 6.2 Removing Software

Remove the directories and files installed as part of the DVX software setup. This includes:

```
/da  
/etc/firewalld/services/datrium.xml  
/etc/profile.d/datrium.sh  
/mnt/datrium  
/opt/datrium  
/usr/lib/systemd/system/datrium.service  
/usr/lib/systemd/system/mnt-datrium-Datastore1.mount  
/var/log/datrium
```

For multiple mount point support when deployed with adding separate Storage Domains, you should also remove the nfsbind configuration applied. Make sure other resources on this host are not affected by the nfsbind changes:

```
/sbin/mount.nfsbind
```

## 6.3 Cleaning Up Data Node

You will need to manually remove the unconfigured Linux host from the DVX Data Node. Do this by logging in to the Data Node using the floating management IP of the Data Node and the admin account. Then run the following command to identify the Compute Nodes currently attached to the Data Node:

```
hosts show
```

Choose the appropriate host by IP address or name and remove:

```
hosts remove <host>
```

NOTE: this may need the --force option if host is still active

## 7 Known Issues

The following sections identify some known issues and workarounds with the current supported version of Linux and DVX. If you encounter any other issues, please consult our Support Knowledge Base here: <https://support.datrium.com/> or contact Technical Support at <https://www.datrium.com/support/>

## 7.1 SSD access after reboot

Problems have been identified on Linux host reboot with the proper discovery of the SSD partitions that were created during initial DVX software setup. The Linux host OS does not identify the devices that are needed by the DVX configuration. This causes the DVX to attempt to reconfigure the services and get hung in an invalid state.

The work around for this condition is to not have the Datrium services and mount point(s) automatically configured on reboot. If a host needs to be rebooted for other maintenance reasons, then the DVX configuration can be manually started after the host has come back online and the SSD devices are properly discovered by the OS.

After the host is up, it is then possible to login to the host and force discovery of the configured SSD devices with the following command:

```
partprobe
```

Before starting the DVX services manually as described in [Section 3.5 DVX Hyperdriver](#) you should check to make sure the SSD devices are discovered as expected - see [Section 3.2 Identifying DVX Flash Devices](#)

## 7.2 NFS locking

The Datrium DVX system uses an NFS style presentation of the storage on each Linux host. For proper setup and sharing across multiple DVX enabled hosts, the mount needs to be configured without locking. To make sure the nolock has been set appropriately do the following:

```
mount | grep -i nolock
```

You should see something like the following:

```
localhost:/192.168.47.110:2181/Datastore1 on /mnt/datrium/Datastore1 type  
nfs (rw, relatime,vers=3,rsize=65536,wsiz=65536,namlen=255,hard,nolock,  
proto=tcp, timeo=600,retrans=2,sec=sys,mountaddr=127.0.0.1,mountvers=3,  
mountport=2049,mountproto=tcp,local_lock=all,addr=127.0.0.1)
```

If “nolock” is not set then edit the file for each datastore defined /etc/systemd/system/multi-user.target.wants/mnt-datrium-Datastore1.mount and add this line to the end of the [Mount] section (without quotes):

```
“Options=vers=3,nolock”
```

Refresh system configuration file as follows:

```
systemctl daemon-reload
```

## 7.3 Working with DVX ISO Storage Domain

Populating the ISO Storage Domain without native NFS services requires a bit of a work around. In short, you will need to copy files to the proper location on a host with the DVX software installed and configured and then manually update the directory configuration used for the ISO Storage Domain. In brief, copy ISO files to this folder:

```
<ISO Storage Domain top level directory>/images/11111111-1111-1111-1111-111111111111
```

Then change to that location:

```
cd <ISO Storage Domain top level directory>images/11111111-1111-1111-1111-111111111111
```

and update ownership and access control as follows:

```
# Change ownership back to vdsm.kvm and control permissions  
chown vdsm.kvm <files>  
chmod 755 <files>
```

*NOTE: There is a good external reference here for working with populating ISO Storage Domains:*  
<https://linuxcluster.wordpress.com/2014/10/23/create-and-populate-a-new-iso-nfs-storage-domain-for-rhev/>