Chapter 15. Distributed Replicated Block Device (DRBD)

Contents

15.1. Conceptual Overview
15.2. Installing DRBD Services
15.3. Configuring the DRBD Service
15.4. Testing the DRBD Service
15.5. Tuning DRBD
15.6. Troubleshooting DRBD
15.7. For More Information

Abstract

The *distributed replicated block device* (DRBD*) allows you to create a mirror of two block devices that are located at two different sites across an IP network. When used with OpenAIS, DRBD supports distributed high-availability Linux clusters. This chapter shows you how to install and set up DRBD.

15.1. Conceptual Overview

DRBD replicates data on the primary device to the secondary device in a way that ensures that both copies of the data remain identical. Think of it as a networked RAID 1. It mirrors data in real-time, so its replication occurs continuously. Applications do not need to know that in fact their data is stored on different disks.

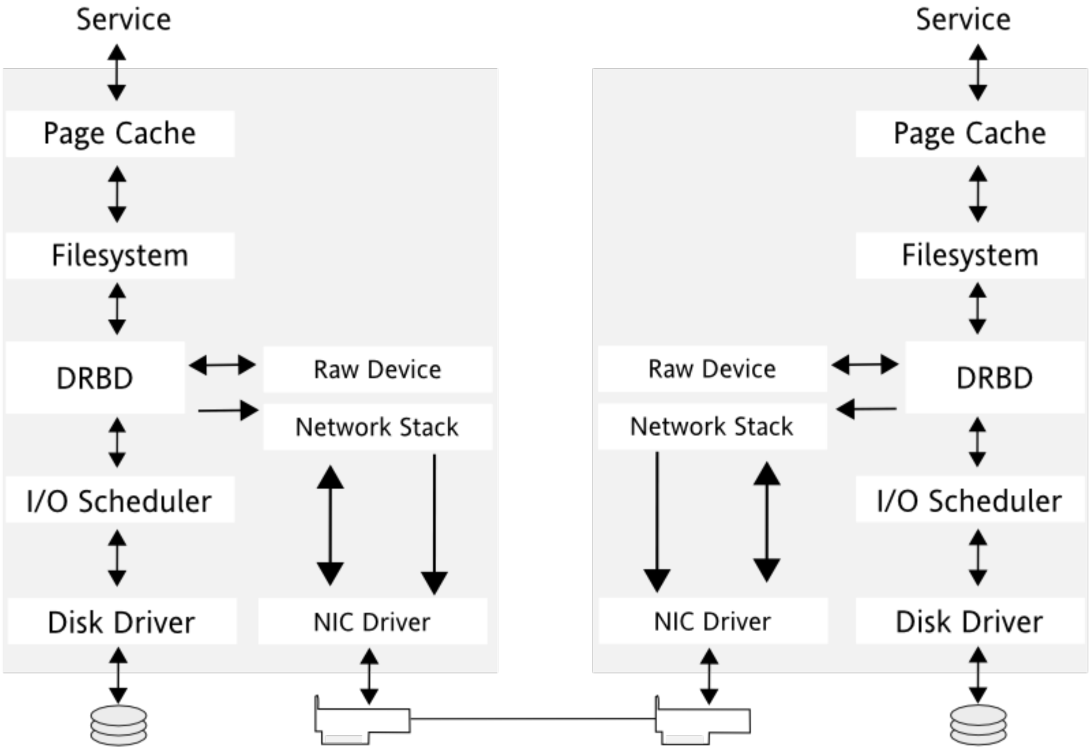
Important: **Unencrypted Data**

The data traffic between mirrors is not encrypted. For secure data

> exchange, you should deploy a Virtual Private Network (VPN) solution for the connection.

DRBD is a Linux kernel module and sits between the I/O scheduler at the lower end and the file system at the upper end, see Figure 15.1, “Position of DRBD within Linux”. To communicate with DRBD, users use the high-level command **drbdadm**. For maximum flexibility DRBD comes with the low-level tool **drbdsetup**.

Figure 15.1. Position of DRBD within Linux



DRBD allows you to use any block device supported by Linux, usually:

- partition or complete hard disk
- software RAID
- Logical Volume Manager (LVM)
- Enterprise Volume Management System (EVMS)

By default, DRBD uses the TCP ports `7788` and higher for communication between DRBD nodes. Make sure that your firewall does not prevent communication on this port.

You must set up the DRBD devices before creating file systems on them. Everything pertaining to user data should be done solely via the `/dev/drbd_R` device and not on the raw device, as DRBD uses the last 128 MB of the raw device for metadata. Make sure to create file systems only on the `/dev/drbd<n>` device and not on the raw device.

For example, if the raw device is 1024 MB in size, the DRBD device has only 896 MB available for data, with 128 MB hidden and reserved for the metadata. Any attempt to access the space between 896 MB and 1024 MB fails because it is not available for user data.

15.2. Installing DRBD Services

To install the needed packages for DRBD, install the High Availability Extension Add-On product on both SUSE Linux Enterprise Server machines in your networked cluster as described in Part I, “Installation and Setup”. Installing High Availability Extension also installs the DRBD program files.

If you do not need the complete cluster stack but just want to use DRBD, refer to Table 15.1, “DRBD RPM Packages”. It contains a list of all RPM packages for DRBD. Recently, the `drbd` package has been split into separate packages.

Table 15.1. DRBD RPM Packages

Filename	Explanation
`drbd`	Convenience package, split into other
`drbd-bash-completion`	Programmable bash completion support for drbdadm
`drbd-heartbeat`	Heartbeat resource agent for DRBD (only needed for Heartbeat)
`drbd-kmp-default`	Kernel module for DRBD (needed)
`drbd-kmp-xen`	Xen kernel module for DRBD
`drbd-udev`	udev integration scripts for DRBD, managing symlinks to DRBD devices in `/dev/drbd/by-res` and `/dev/drbd/by-disk`
`drbd-utils`	Management utilities for DRBD (needed)
`drbd-pacemaker`	Pacemaker resource agent for DRBD
`drbd-xen`	Xen block device management script for DRBD

Filename	Explanation
yast2-drbd	YaST DRBD Configuration (recommended)

To simplify the work with **drbdadm**, use the Bash completion support in the RPM package drbd-bash-completion. If you want to enable it in your current shell session, insert the following command:

```
source /etc/bash_completion.d/drbdadm.sh
```

To use it permanently for root, create a file /root/.bashrc and insert the previous line.

15.3. Configuring the DRBD Service

> Note:
>
> The following procedure uses the server names jupiter and venus, and the cluster resource name r0. It sets up jupiter as the primary node. Make sure to modify the instructions to use your own nodes and filenames.

Before you start configuring DRBD, make sure the block devices in your Linux nodes are ready and partitioned (if needed). The following procedure assumes you have two nodes, jupiter and venus, and they use the TCP port 7788. Make sure this port is open in your firewall.

To set up DRBD manually, proceed as follows:

Procedure 15.1. Manually Configuring DRBD

1. Log in as user root.

2. Change DRBD's configuration files:

   a. Open the file /etc/drbd.conf and insert the following lines, if not available:

      ```
      include "drbd.d/global_common.conf";
      include "drbd.d/*.res";
      ```

      Beginning with DRBD 8.3 the configuration file is split into separate files, located under the directory `/etc/drbd.d/`.

b. Open the file `/etc/drbd.d/global_common.conf`. It contains already some pre-defined values. Go to the `startup` section and insert these lines:

```
startup {
    # wfc-timeout degr-wfc-timeout outdated-wfc-timeout
    # wait-after-sb;
    wfc-timeout 1;
    degr-wfc-timeout 1;
}
```

These options are used to reduce the timeouts when booting, see http://www.drbd.org/users-guide-emb/re-drbdconf.html for more details.

c. Create the file `/etc/drbd.d/r0.res`, change the lines according to your situation, and save it:

```
resource r0 { ❶
  device /dev/drbd_r0 minor 0; ❷
  disk /dev/sda1; ❸
  meta-disk internal; ❹
  on jupiter { ❺
    address  192.168.1.10:7788; ❻
  }
  on venus { ❺
    address 192.168.1.11:7788; ❻
  }
  syncer {
    rate  7M; ❼
  }
}
```

❶ Name of the resource. It is recommended to use resource names like `r0`, `r1`, etc.

❷ The device name for DRBD and its minor number.

In the example above, the device node name, as created with udev, is referenced (`/dev/drbd_r0`, with `r0` representing the resource name). For this usage, you need to have the `drbd-udev` package installed. Alternatively, omit the device node name in the configuration and use the following line instead:

```
device minor 0
```

❸ The device that is replicated between nodes. Note, in this example the devices are the same on both nodes. If you need different devices, move the `disk` parameter into the `on` section.

❹ The meta-disk parameter usually contains the value `internal`, but it is possible to specify an explicit device to hold the meta data. See http://www.drbd.org/users-guide-emb/ch-internals.html#s-metadata for more information.

❺ The `on` section contains the hostname of the node

❻ The IP address and port number of the respective node. Each resource needs an individual port, usually starting with `7788`.

❼ The synchronization rate. Set it to one third of your bandwidth. It only limits the resynchronization, not the mirroring.

3. Check the syntax of your configuration file(s). If the following command returns an error, verify your files:

   ```
   drbdadm dump all
   ```

4. If you have configured Csync2 (which should be the default), the DRBD configuration files are already included in the list of files which need to be synchronized. To syncronize them, use:

   ```
   csync2 -xv
   ```

   If you do not have Csync2 (or do not want to use it), copy the DRBD configuration files manually to the other node:

   ```
   scp /etc/drbd.conf venus:/etc/
   scp /etc/drbd.d/*  venus:/etc/drbd.d/
   ```

5. Initialize the meta data on *both* systems by entering the following on each node:

   ```
   drbdadm -- --ignore-sanity-checks create-md r0
   rcdrbd start
   ```

   If your disk already contains a file system that you do not need anymore, destroy the file system structure with the following command and repeat this step:

```
dd if=/dev/zero of=/dev/sdb1 count=10000
```

6. Watch the DRBD status by entering the following on each node:

   ```
   rcdrbd status
   ```

   You should get something like this:

   ```
   drbd driver loaded OK; device status:
   version: 8.3.7 (api:88/proto:86-91)
   GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by phil@fat-t
   m:res  cs         ro                   ds                         p  m
   0:r0   Connected  Secondary/Secondary  Inconsistent/Inconsistent  C
   ```

7. Start the resync process on your intended primary node (jupiter in this case):

   ```
   drbdadm -- --overwrite-data-of-peer primary r0
   ```

8. Check the status again with **`rcdrbd status`** and you get:

   ```
   ...
   m:res  cs         ro                 ds                 p  mounted  fs
   0:r0   Connected  Primary/Secondary  UpToDate/UpToDate  C
   ```

   The status in the `ds` row (disk status) must be UpToDate on both nodes.

9. Set jupiter as primary node:

   ```
   drbdadm primary r0
   ```

10. Create your file system on top of your DRBD device, for example:

    ```
    mkfs.ext3 /dev/drbd_r0
    ```

11. Mount the file system and use it:

    ```
    mount /dev/drbd_r0 /mnt/
    ```

To use YaST to configure DRBD, proceed as follows:

Procedure 15.2. Using YaST to Configure DRBD

1. Start YaST and select the configuration module *High Availability+DRBD*. If you have already a DRBD configuration, YaST warns you. YaST will change your configuration and will save your old DRBD configuration files as `*.YaSTsave`.

2. In *Start-up Configuration+Booting* select *On* to start DRBD always at boot time.

3. If you need to configure more than one replicated resource, select *Global Configuration*. The input field *Minor Count* selects how many different DRBD resources may be configured without restarting the computer.

4. The actual configuration of the resource is done in *Resource Configuration*. Press *Add* to create a new resource. The following parameters have to be set twice:

Resource Name	The name of the resource, often called `r0`. This parameter is mandatory.
Name	The hostname of the relevant node
Address:Port	The IP address and port number (default 7788) of the respective node
Device	The device that holds the replicated data on the respective node. Use this device to create file systems and mount operations.
Disk	The device that is replicated between both nodes
Meta-disk	The *Meta-disk* is either set to the value `internal` or specifies an explicit device extended by an index to hold the meta data needed by DRBD. When using `internal`, the last 128 MB of the replicated device are used to store the meta data. A real device may also be used for multiple drbd resources. For example, if your *Meta-Disk* is `/dev/sda6[0]` for the first resource, you may use `/dev/sda6[1]` for the second resource. However, there must be at least 128 MB space for each resource available on this disk.

All of these options are explained in the examples in the `/usr/share/doc/packages/drbd-utils/drbd.conf` file and in the man page of

drbd.conf(5).

5. If you have configured Csync2 (which should be the default), the DRBD configuration files are already included in the list of files which need to be synchronized. To syncronize them, use:

   ```
   csync2 -xv
   ```

   If you do not have Csync2 (or do not want to use it), copy the DRBD configuration files manually to the other node (pretending to be another node with the name venus):

   ```
   scp /etc/drbd.conf venus:/etc/
   scp /etc/drbd.d/*  venus:/etc/drbd.d/
   ```

6. Initialize and start the DRBD service on both systems by entering the following on each node:

   ```
   drbdadm create-md r0
   rcdrbd start
   ```

7. Configure node1 as the primary node by entering the following on node1:

   ```
   drbdsetup /dev/drbd0 primary --overwrite-data-of-peer
   ```

8. Check the DRBD service status by entering the following on each node:

   ```
   rcdrbd status
   ```

   Before proceeding, wait until the block devices on both nodes are fully synchronized. Repeat the **rcdrbd status** command to follow the synchronization progress.

9. After the block devices on both nodes are fully synchronized, format the DRBD device on the primary with your preferred file system. Any Linux file system can be used.

   Important:

   Always use the /dev/drbd<n> name in the command, not the actual /dev/disk device name.

15.4. Testing the DRBD Service

If the install and configuration procedures worked as expected, you are ready to run a basic test of the DRBD functionality. This test also helps with understanding how the software works.

1. Test the DRBD service on jupiter.

   a. Open a terminal console, then log in as root.

   b. Create a mount point on jupiter, such as /srv/r0mount:

   ```
   mkdir -p /srv/r0mount
   ```

   c. Mount the **drbd** device:

   ```
   mount -o rw /dev/drbd0 /srv/r0mount
   ```

   d. Create a file from the primary node:

   ```
   touch /srv/r0mount/from_node1
   ```

2. Test the DRBD service on venus.

   a. Open a terminal console, then log in as root.

   b. Unmount the disk on jupiter:

   ```
   umount /srv/r0mount
   ```

   c. Downgrade the DRBD service on jupiter by typing the following command on jupiter:

   ```
   drbdadm secondary r0
   ```

   d. On venus, promote the DRBD service to primary:

   ```
   drbdadm primary r0
   ```

   e. On venus, check to see if venus is primary:

```
rcdrbd status
```

f. On venus, create a mount point such as `/srv/r0mount`:

```
mkdir /srv/r0mount
```

g. On venus, mount the DRBD device:

```
mount -o rw /dev/drbd_r0 /srv/r0mount
```

h. Verify that the file you created on jupiter is viewable.

```
ls /srv/r0mount
```

The `/srv/r0mount/from_node1` file should be listed.

3. If the service is working on both nodes, the DRBD setup is complete.

4. Set up jupiter as the primary again.

a. Dismount the disk on venus by typing the following command on venus:

```
umount /srv/r0mount
```

b. Downgrade the DRBD service on venus by typing the following command on venus:

```
drbdadm secondary r0
```

c. On jupiter, promote the DRBD service to primary:

```
drbdadm primary r0
```

d. On jupiter, check to see if jupiter is primary:

```
rcdrbd status
```

5. To get the service to automatically start and fail over if the server has a problem, you can set up DRBD as a high availability service with OpenAIS. For information about installing and configuring OpenAIS for SUSE Linux Enterprise 11 see Part II, “Configuration and Administration”.

15.5. Tuning DRBD

There are several ways to tune DRBD:

1. Use an external disk for your metadata. This speeds up your connection.

2. Create a udev rule to change the read-ahead of the DRBD device. Save the following line in the file `/etc/udev/rules.d/82-dm-ra.rules` and change the read_ahead_kb value to your workload:

   ```
   ACTION=="add", KERNEL=="dm-*", ATTR{bdi/read_ahead_kb}="4100"
   ```

   This line only works if you use LVM.

3. Activate bmbv on Linux software RAID systems. The `use-bmbv` keyword enables DRBD to process IO requests in units not lager than 4kByte. However, this option should be adapted carefully. Use the following line in the common disk section of your DRBD configuration, usually in `/etc/drbd.d/global_common.conf`:

   ```
   disk {
     use-bmbv;
   }
   ```

15.6. Troubleshooting DRBD

The drbd setup involves many different components and problems may arise from different sources. The following sections cover several common scenarios and recommend various solutions.

15.6.1. Configuration

If the initial `drbd` setup does not work as expected, there is probably something wrong with your configuration.

To get information about the configuration:

1. Open a terminal console, then log in as `root`.

2. Test the configuration file by running **`drbdadm`** with the **`-d`** option. Enter the following command:

   ```
   drbdadm -d adjust r0
   ```

   In a dry run of the **`adjust`** option, **`drbdadm`** compares the actual configuration of the DRBD resource with your DRBD configuration file, but it does not execute the calls. Review the output to make sure you know the source and cause of any errors.

3. If there are errors in the `/etc/drbd.d/*` and `drbd.conf` files, correct them before continuing.

4. If the partitions and settings are correct, run **`drbdadm`** again without the **`-d`** option.

   ```
   drbdadm adjust r0
   ```

   This applies the configuration file to the DRBD resource.

15.6.2. Hostnames

For DRBD, hostnames are case sensitive (`Node0` would be a different host than `node0`).

If you have several network devices and want to use a dedicated network device, the hostname will likely not resolve to the used IP address. In this case, use the parameter `disable-ip-verification`.

15.6.3. TCP Port 7788

If your system is unable to connect to the peer, this might be a problem with your local firewall. By default, DRBD uses the TCP port `7788` to access the other node. Make sure that this port is accessible on both nodes.

15.6.4. DRBD Devices Broken after Reboot

In cases when DRBD does not know which of the real devices holds the latest data, it changes to a split brain condition. In this case, the respective DRBD subsystems come up as secondary and do not connect to each other. In this case, the following message is written to `/var/log/messages`:

```
Split-Brain detected, dropping connection!
```

To resolve this situation, enter the following on the node which has data to be discarded:

```
drbdadm secondary r0
drbdadm -- --discard-my-data connect r0
```

On the node which has the latest data enter the following:

```
drbdadm connect r0
```

15.7. For More Information

The following open source resources are available for DRBD:

- The project home page http://www.drbd.org.
- See Highly Available NFS Storage with DRBD and Pacemaker (↑Highly Available NFS Storage with DRBD and Pacemaker).
- http://clusterlabs.org/wiki/DRBD_HowTo_1.0 by the Linux Pacemaker Cluster Stack Project.
- The following man pages for DRBD are available in the distribution: **drbd(8)**, **drbddisk(8)**, **drbdsetup(8)**, **drbdsetup(8)**, **drbdadm(8)**, **drbd.conf(5)**.
- Find a commented example configuration for DRBD at `/usr/share/doc/packages/drbd/drbd.conf`