Creating a 2-node replicated block device setup with WinDRBD

Johannes Thoma <johannes@johannesthoma.com>

Table of Contents

Chapter 1. Introduction

1.1. About

This document provides a step-by-step guide of how to create a replicated block device between two Windows Server 2016 hosts. Use cases for such a setup are, amongst others, disaster recovery (standby node can take over immediately, because it has up-to-date data) and software defined storage (for example, migrating virtual machines).

1.2. Overview

Rolling out a 2 node setup using WinDRBD involves following steps:

- Prerequisites describes what you need for a 2-node setup of WinDRBD.
- Obtaining an installable version of WinDRBD describes how to get an installable version of WinDRBD from Linbit.
- Installing WinDRBD describes how to install WinDRBD on your Windows Server 2016 machine.
- Create backing storage devices describes how to create backing storage devices.
- Configure Windows firewall describes how to modify your Windows firewall settings for WinDRBD.
- Configure the WinDRBD resource describes how to configure the WinDRBD resources.
- Start WinDRBD resources describes how to bring the WinDRBD resources up.
- Using WinDRBD devices describes how you can use the WinDRBD devices.
- Testing WinDRBD describes how you can check if WinDRBD is actually doing the right thing.
- Troubleshooting WinDRBD describes what you can do if something goes wrong.

Chapter 2. Installing WinDRBD

2.1. Prerequisites

To deploy a 2-node setup of WinDRBD you need 2 machines connected over an internet protocol (IP) based network. At least one machine should run a Microsoft Windows operating system supported by WinDRBD. The other machine may be a Linux machine or another machine with a supported Windows OS.

Current supported versions of Windows include:

- Windows Server 2016 (64 bit)
- Windows 10 (64 bit)
- Windows 7 (service pack 1) (64 bit)

If you wish to use WinDRBD on a 32 bit Windows OS, please contact Linbit.

For Linux, most modern distributions (RHEL, Debian, Ubuntu, SLES, ...) should work.

The machines may be physical or virtual machines.

The machines should have free storage (unpartitioned storage area) or a partition which is currently unused for use as a WinDRBD backing device. It is possible to convert an existing NTFS partition for use as a WinDRBD backing device. However we recommend not to use partitions that contain valuable data for testing WinDRBD since WinDRBD is still beta (as of 12/2018).

For the network connectivity, the machines don't need to be in the same LAN, they may also be located in different geographical regions. However when using WinDRBD over long distances, using DRBD Proxy is recommended.

As of now (12/2018), network should be IPv4 based, i.e. the nodes should be able to communicate with each other over IPv4.

It is highly recommended that you make a full backup of your Windows installation before installing WinDRBD.

2.2. Obtaining an installable version of WinDRBD

Since 64-bit Windows versions require drivers to be digitally signed, Linbit releases signed versions of WinDRBD with a trusted certificate. These releases contain a self-extracting EXE with an inno-setup based installer and thus should be also be installable on older versions of Microsoft Windows.

To obtain an installable version of WinDRBD, please go to https://www.linbit.com/en/drbd-community/drbd-download/ When you scroll down you will find installable EXE files. They contain both the kernel driver and the utility programs (drbd-utils), as well as this tech guide.

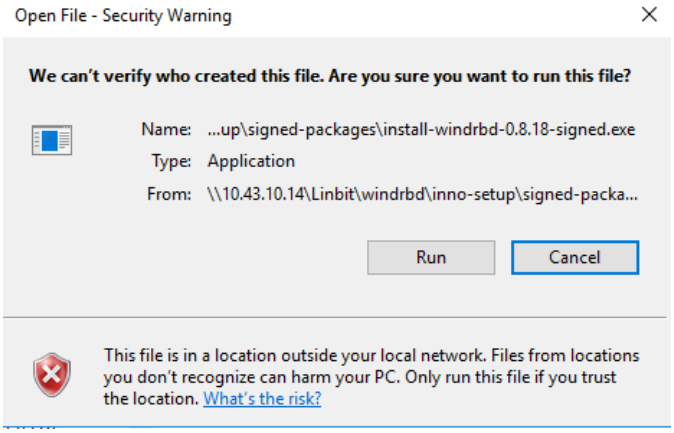
The EXE files follow the naming scheme

```
install-windrbd-x.y.z-signed.exe
```

where x.y.z is the version code of WinDRBD (as shown by `drbdadm --version`).

2.3. Installing WinDRBD

Once you have downloaded WinDRBD simply start the installer by double clicking the EXE file. Since the installer will be run as Administrator user (it has to, since it will install a kernel driver), Windows will ask if you really want to execute the installer. Click Run to run it.



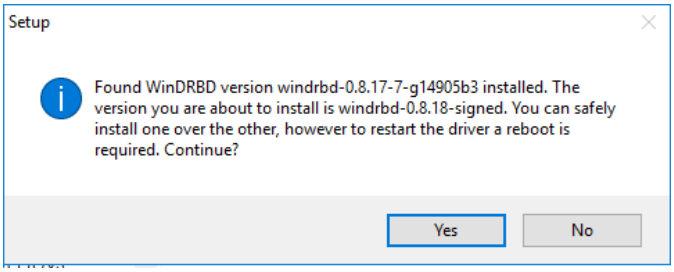
Once started the installer will ask for a language for the installation (this is just for the installation, most texts will still be in English, especially the help texts of the userspace utilities, like `drbdadm`).

2.3.1. Upgrade from earlier versions

Then, the installer checks for previously installed versions of WinDRBD. In the rare case you have beta 4 installed, you must choose to uninstall it first, since it is incompatible with newer versions of WinDRBD. Please do not forget to reboot the machine after installation has finished in case you had beta4 installed before (the installer will not ask for a reboot, you have to do that manually).

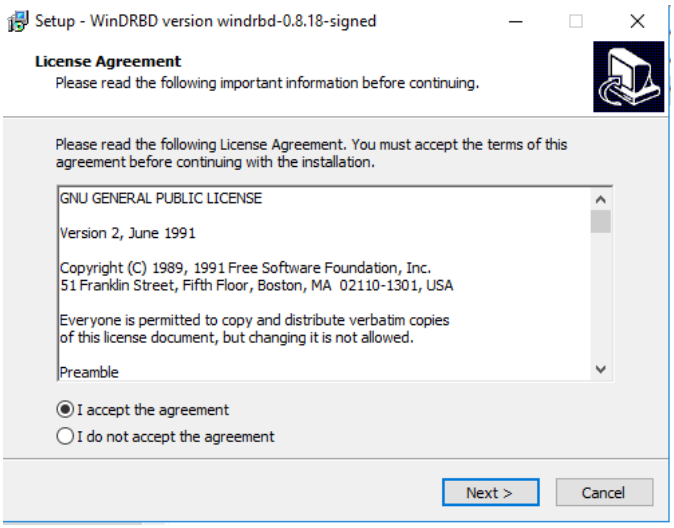
For all other versions, you will get a panel that asks if you wish to upgrade from a previously installed version. If you don't get a panel (and license agreement is shown right away) you don't have WinDRBD already installed and also don't need a reboot after installation.

An upgrade panel might look like this:



2.3.2. License agreement

You will then be asked to accept the WinDRBD license agreement. WinDRBD is licensed under the GNU general public license and therefore is Open Source. Note that there is no kind of warranty for GPL licensed software and so also for WinDRBD. Please read the license carefully, since it tells you what rights you have.



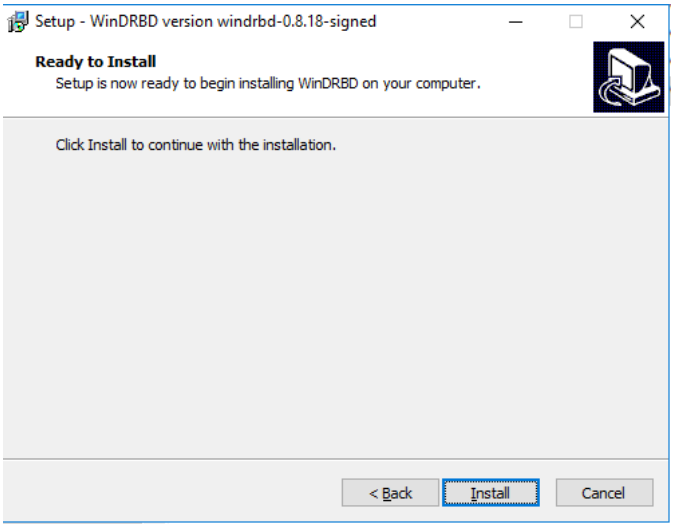
To accept the license, select I accept the agreement and click Next.

2.3.3. Release notes

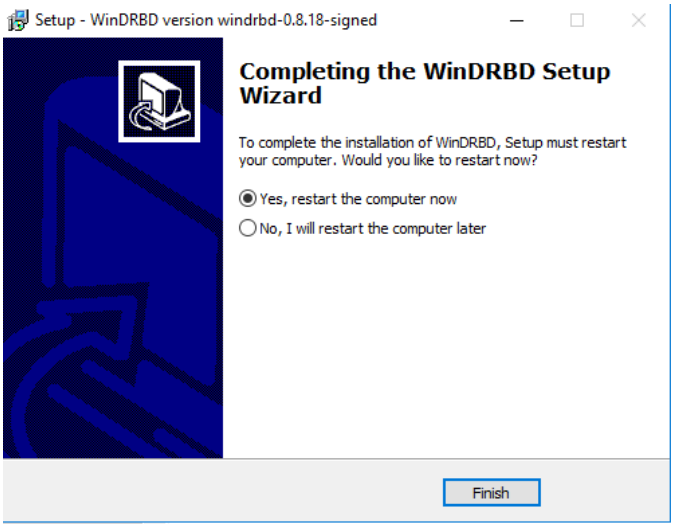
Next screen shows an overview about what has changed recently in WinDRBD. This is mainly of interest if you are upgrading from an earlier release of WinDRBD. Once you are done reading it, click next to continue.

2.3.4. Installation

We are now ready to install WinDRBD on your computer. Click install to start the installation process. The process itself should take no more than 15 seconds.



You then might get prompted to reboot the machine if this is an upgrade. Save all your work and click Finish in order to finish WinDRBD installation.



In case this is a new installation, no reboot is required. Instead, you will be asked if you wish to view the windrbd.pdf readme file (you are currently reading it already).

Chapter 3. Preparing your computer for use with WinDRBD

The next three sections describe what to do in order to connect two nodes with WinDRBD.

3.1. Create backing storage devices

Even though having a drive for backing storage of the WinDRBD device is not strictly necessary (there can be diskless nodes, they even can be Primary if there is a network connection to a diskful node), we will assume in our setup that both nodes have backing storage configured.

The following instructions explain how to create a partition for use as backing storage for WinDRBD.

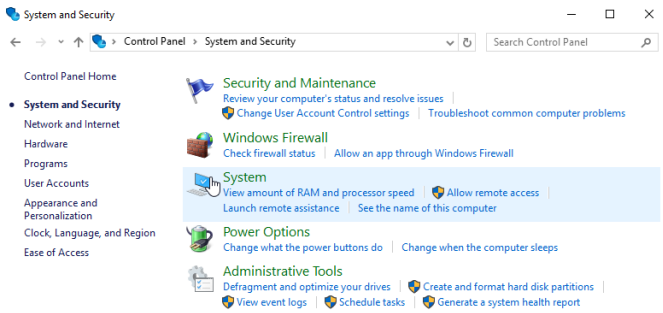
On each node you need one partition as backing storage, provided that you will use internal meta data (which is what we recommend, because it is less work to set up). Please keep in mind that the meta data will use some disk space of the partition, so the actual usable data will be slightly less that the size of the partition. Also, the size of the partition for the backing storage must be the same on both nodes.

If you really do not have free space on your machine, you also may use a USB stick as backing device. You then need to plug the USB stick and use its drive letter for the disk clause of the WinDRBD configuration. If you use an USB stick you can skip to the firewall settings section now.

3.1.1. Start Disk Management

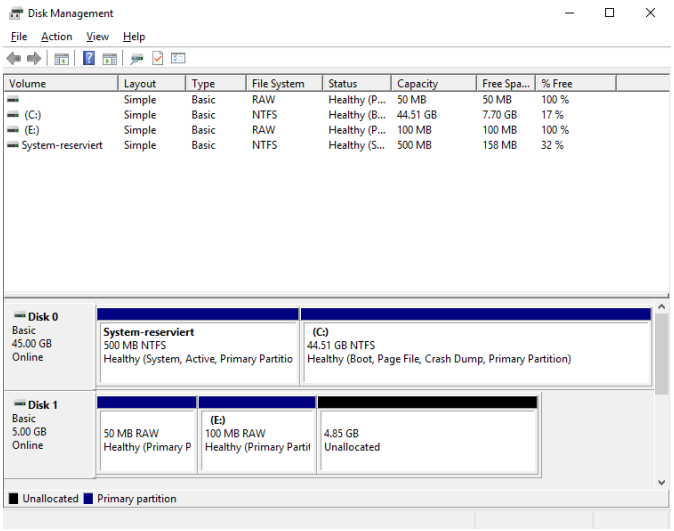
To start Disk Management, open the control panel, which can be found under Windows System in the start menu on Windows Server 2016 (other releases of Windows might have different ways to open control panel). Note that Control Panel is different from System settings.

From Control Panel, select System and Security (click on the green heading). Then, under Administrative Tools (last section) there is a blue link 'Create and format hard disk partitions'. Clicking this link will open the Disk Management facility.



3.1.2. Find empty space

The Disk Management overview will show empty space if there is any. The caption will say "Unallocated".



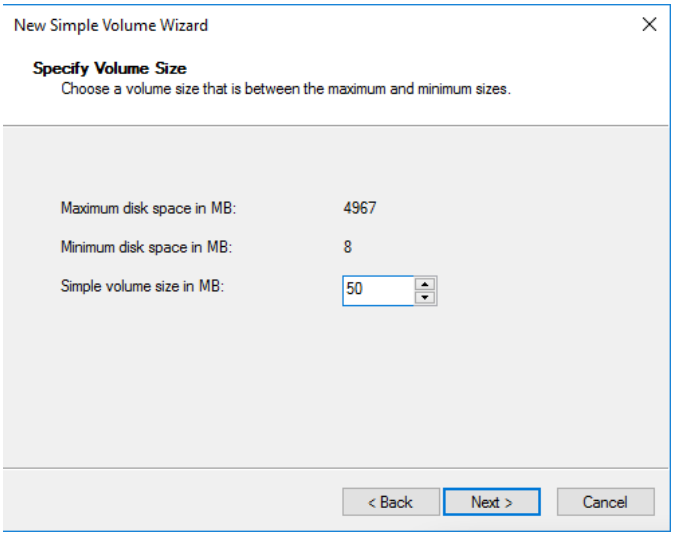
If there is no empty space, either add an extra hard drive to the system (recommended) or shrink an existing partition. We recommend not to shrink the system volume (C:) since that may render the installation unusable.

To shrink a partition, right-click on the partition to shrink and select "Shrink Volume". On the panel that appears enter the amount of space to shrink of the volume (the size needed for the WinDRBD backing storage).

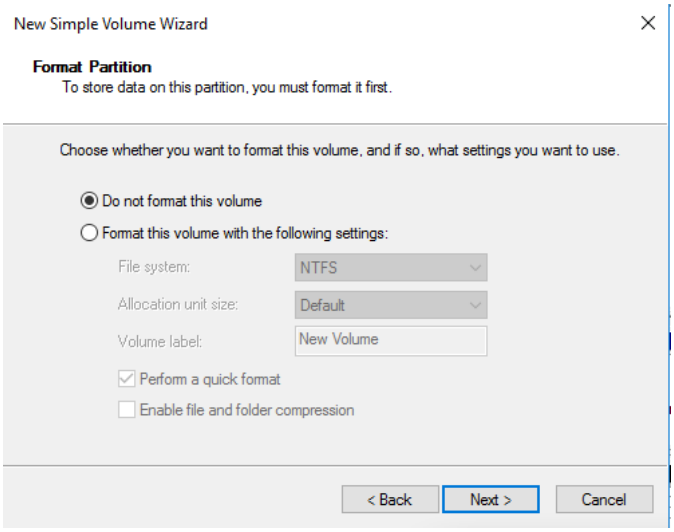
3.1.3. Partition your drive

To create backing storage for the WinDRBD device, right-click an empty section in Disk Management and select New simple Volume. You will be guided through creation of the new partition.

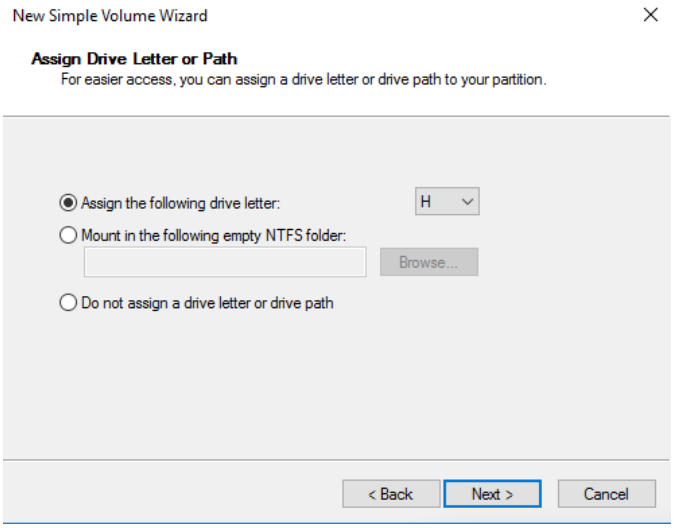
First select the size in megabytes, including the storage needed for internal meta data. Enter the size and click Next.



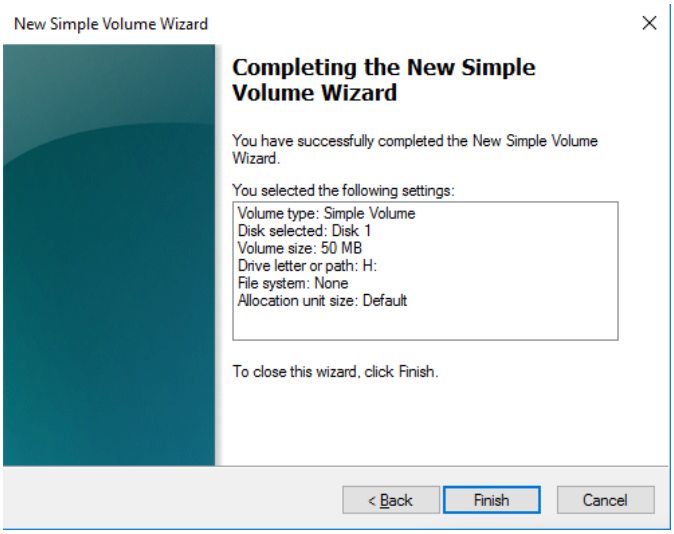
Then, the system asks if you wish to format the partition. Since we will format the WinDRBD device later on, select don't format here. Click Next.



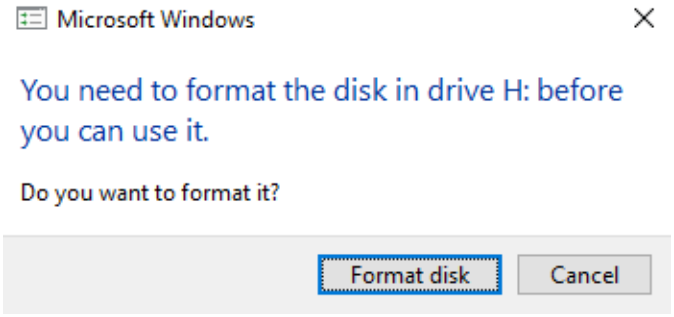
You then will be prompted to assign a drive letter to the new partition. For test setups you can assign a drive letter here (remember this drive letter, you will need it for the disk parameter in the WinDRBD configuration later). For production setups with many devices it is recommended not to assign a drive letter here and use the partition's GUID to address the disk instead. Note that this drive letter is different to the drive letter used by WinDRBD (which we will use to access data). Do not use the backing device's drive letter to modify data, since WinDRBD then would not replicate it to the other node. Click Next when done.



Finally, the wizard shows a summary. If everything is ok here, click Finish to actually create the partition.



If you have assigned a drive letter, Windows will now ask once again if you wish to format the partition. Click Cancel because we do not want to format the backing device.

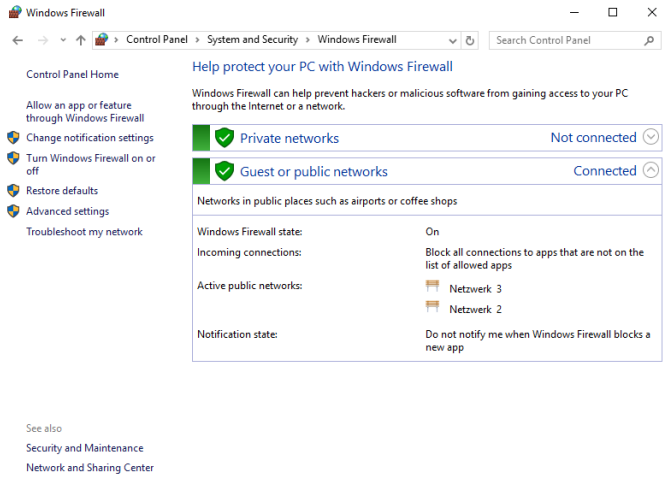


3.2. Configure Windows firewall

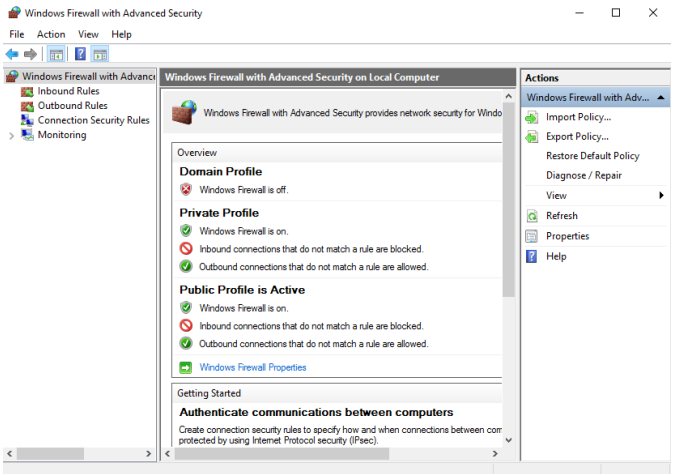
Since WinDRBD communicates via TCP/IP over ports defined in the WinDRBD configuration, the Windows firewall has to be configured not to block connections on that ports. We assume for now that the resource will use port 7500 for communication.

3.2.1. Opening Windows Firewall Advanced settings

To open the firewall settings, open control panel (you will find it under Windows System under Windows Server 2016). Then click on System and Security (green caption), then on Windows Firewall (again, green caption).



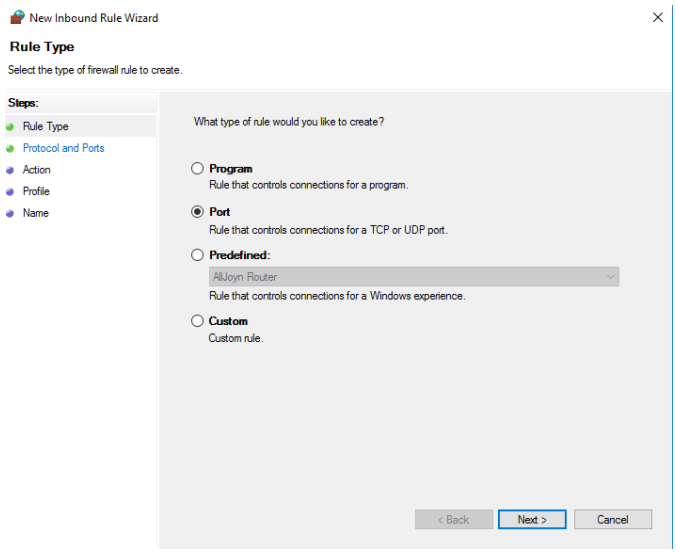
Click on Advanced settings to open the firewall settings where you can define rules for individual ports.



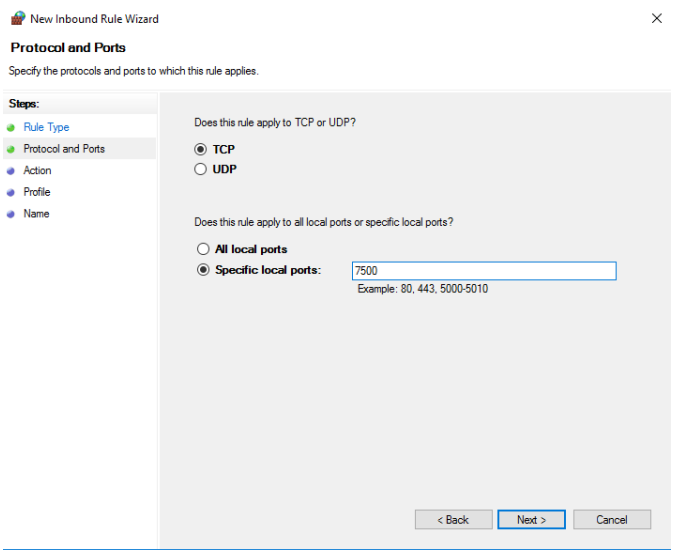
3.2.2. Creating rules for WinDRBD ports

Since WinDRBD ports are not well known and do not belong to a specific application, you'll have to create individual rules for all ports you are using.

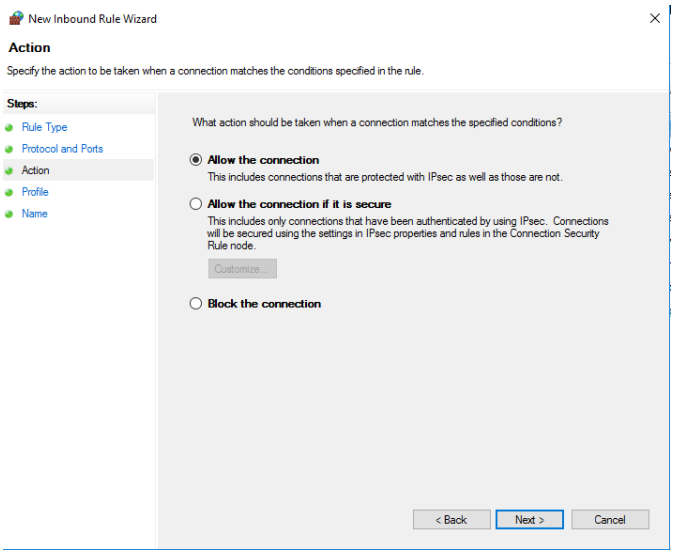
- Click on Inbound Rules in the left section of the Window.
- Click on new rule in the right section of the Window. A wizard Window will pop up.



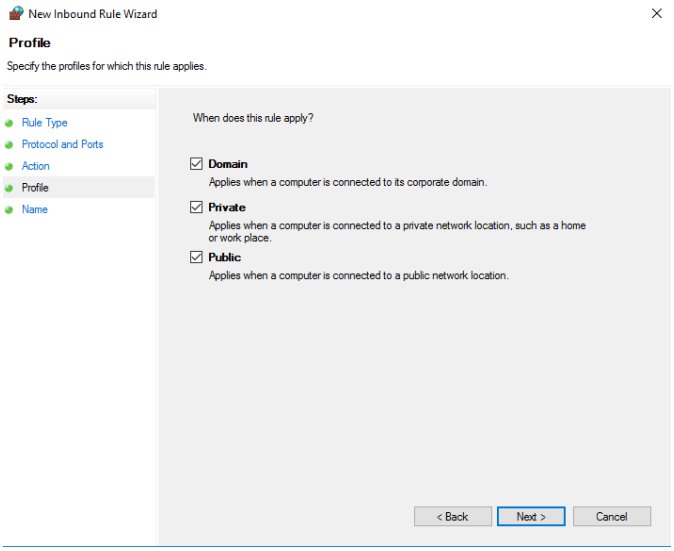
- Select Port as the type of the new rule and click Next.
- On the next Window, select TCP and select specific ports.



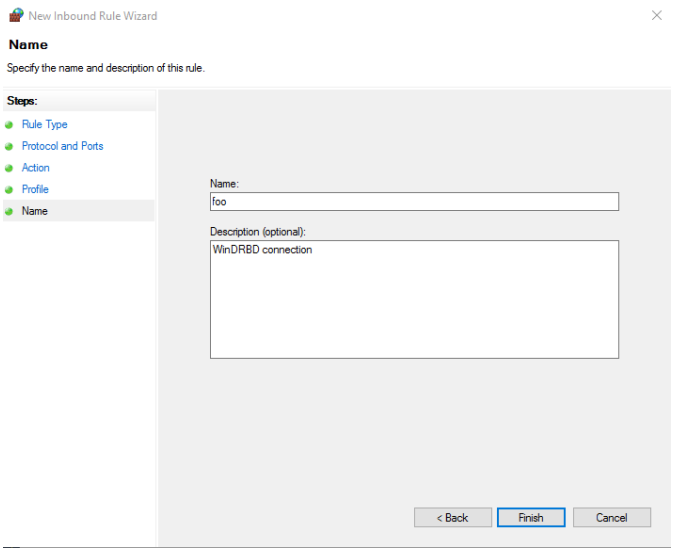
- In the list of ports enter the port number (7500 in our case) and click Next.
- On the next Window select Allow connection and click Next.



- On the next Window select all network profiles. Click Next.



- For the Name on the next Window you can enter the name of the resource ("foo") or some name that helps you to identify the rule later. Click Finish to create the rule.



Create another rule for an outgoing connection with the remote port (we recommend to use the same port number for

local and remote ports).

- Click outgoing connections.
- Repeat the above steps for creating an outgoing rule that allows connections to port 7500.

Your Windows firewall should now be prepared to allow WinDRBD connections for the specific port. You will learn how to configure this port in the next chapter.

3.3. Configure the WinDRBD resource

To configure WinDRBD, as with the Linux version, text files are used. You will find (after installation of WinDRBD) the folder C:\Windrbd which contains a folder etc (which is where UNIX configuration files are located). This folder, in turn, contains a folder named drbd.d, which contains files with the extension .res . These files contain the configuration for the WinDRBD devices (we call them resources).

3.3.1. Locating the WinDRBD configuration folder

First of all, open the WinDRBD configuration directory, it is:

```
C:\Windrbd\etc\drbd.d
```

You can do so with explorer, but there is also (under the WinDRBD menu entry) a start menu shortcut to open this folder.

3.3.2. Creating a new resource file

Create a copy of the windrbd-sample.res file with explorer. The new file should be named after the resource it contains. So if the resource should be named "foo" the file name should be foo.res. When creating new files be sure to keep the extension .res else the file will not be included by the drbd.conf in the etc directory.

Note that .res is also the extension of Windows resources, which typically open with Visual Studio. In order to open it with a text editor, right click the file and chose Open With to select a text editor. Since the configuration files understand Windows line endings (CR/LF) you also may edit them with older versions of Notepad. If that does not work, you may also use WordPad to edit the file if you save it as a Text file (without formatting).

3.3.3. Adapting windrbd-sample.res

You have to change following entries in the sample configuration:

- The resource name: It should be named after the file (else you get confused finding the resources later). So if the filename is foo.res the resource should be named "foo". So it should look like:

  ```
  resource "foo" {
      ...
  }
  ```

- The host names: There are two "on" sections in the file. They have to be named after the host names of the two machines. So, if one machine is "alice" and the other machine is "bob" it should look like:

```
resource "foo" {
    ...
    on alice {
        ...
    }
    on bob {
        ...
    }
}
```

- The IP address and port numbers: the address clause in the on section define the IP address of alice and bob. Currently only IP v4 is supported by WinDRBD, so the format is:

```
on alice {
    address 10.0.0.5:7500
    ...
}
on bob {
    address 10.0.0.6:7500
    ...
}
```

where 10.0.0.5 is the IP v4 address of alice, 10.0.0.6 is the IP address of bob and 7500 is the port number. The port number can be any unused port number (less than 65536). We recommend not using so-called well-known ports (less than 1024). We also recommend using the same port number on all hosts (which isn't enforced by WinDRBD but is good practice).

The port number must be configured as Allow the connection in the Windows firewall settings (see previous chapter), otherwise WinDRBD will fail to establish the connection.

- The backing storage: On each host there are volume sections. A resource can contain more than one volume. To keep things simple we have only one volume for our foo resource. The disk clause defines the backing storage of the volume. Put the drive letter you have assigned in the last chapter as parameter to the disk parameter (you may also put the GUID of the drive, which can be found with the mountvol utility).

The volume section should look like:

```
volume 1 {
    disk "E:";
    ....
}
```

provided that E: is the drive letter of the backing device.

- Finally, the WinDRBD device itself has to be configured. Here we must specify a mount point (a drive letter) and a DRBD minor number. The minor number (and the mount point) must be unique on that host.

Let's say you want to access your data as drive F: the volume section now looks like:

```
volume 1 {
    disk "E:";
    device "F:" minor 1;
    ....
}
```

In theory, once you have configured IP address, backing storage and WinDRBD device, you are done configuring your WinDRBD device.

Copy the file to your other machine. If the other machine is also a Windows machine, you can use the same file without modifications. If the other machine is a Linux host, and the drbd-utils version is less than 9.7.0 you have to change the disk and device clauses to the /dev notation. So for Linux this may look like (note that this are sample values):

```
volume 1 {
    disk /dev/sdb1;
    device /dev/drbd1;
    ....
}
```

Chapter 4. Using WinDRBD

In this chapter we will discuss various ways of using WinDRBD. Most of these will involve the drbdadm command line command, which is the primary way of issuing DRBD commands.

4.1. Opening a command line shell

Although WinDRBD uses the CygWin library for its userland utilities (drbd-utils) you don't have to install CygWin. Instead you can use a regular cmd shell (PowerShell probably also works but hasn't been tested) to interact with WinDRBD.

To start a command line cmd shell, navigate to the C:\Windows\System32 directory and locate the cmd.exe program. Right click it and select Run As Administrator.

If you had a shell open before installing WinDRBD, you need to restart this shell. This is because the installer modifies the PATH variable, which is used to find programs. It adds the directory the WinDRBD utilities can be found to the PATH variable.

4.2. Installing CygWin

Note that since WinDRBD comes with a bundled cygwin1.dll, this step is optional.

If you are familiar with the UNIX/Linux command line, it will probably helpful to install CygWin. Please go to http://www.cygwin.com/install.html and follow the instructions there. Please note that there might be a version conflict between the CygWin DLL that comes with WinDRBD (you will find it under C:\Program Files\WinDRBD\cygwin1.dll) and the cygwin DLL that comes with your new CygWin installation. If commands fail, please delete the cygwin DLL that came with WinDRBD.

4.3. Checking syntax

You can check the syntax of the WinDRBD resource file (without modifying the system) using the `drbdadm dump <res>` command. If your resource is called foo, the command will be:

```
drbdadm dump foo
```

If it reports any errors (it will report the line number of the clause that caused the error) you have to fix the errors first.

4.4. Creating meta data

DRBD (and thus WinDRBD) maintains device metadata to keep track of updated data on the devices. This metadata contains, amongst other things, a bitmap that tracks modified blocks on the device.

You normally only create metadata on first usage. The command to create meta data is `drbdadm create-md <res>`. So if the resource is called foo, you can create meta data by running (as Administrator user)

```
drbdadm create-md foo
```

Be careful not to overwrite existing metadata unless you really mean it (`drbdadm create-md` will ask).

4.5. Start WinDRBD resources

Bringing a WinDRBD resource up (and in Secondary mode) is very easy: the `drbdadm up <res>` command does

this. Again, for resource foo, the command is (run as Administrator):

```
drbdadm up foo
```

Now, if the driver hasn't been loaded yet, you will notice that WinDRBD starts the driver (alongside with two userland services, one for logging and one for user mode helpers). Once the driver is started it remains in memory until the machine is rebooted (there are ways to manually stop the driver but this is out of scope of this document). This is also the reason why there is a reboot required when WinDRBD is updated to a new version or when it is uninstalled.

To test if your resource is really up you can use the `drbdadm status` command (without specifying a resource it will display all resources). Do

```
drbdadm status
```

to see if your resource is up. `drbdadm status` will also display the network connection status which should be Connecting as long as you haven't brought up the resource on the other node.

The disk state for this new disk is Inconsistent. You will learn later how to change this to UpToDate.

4.6. Connecting the nodes

In order to connect the nodes, simply bring up the resource on the other node by running

```
drbdadm up foo
```

After a while (it may take up to 30-60 seconds), executing `drbdadm status` should show the nodes as connected with the peer disk state Inconsistent for now.

If the nodes do not connect, there are several possible reasons. The exact reason can (currently) only be found by looking at the kernel log file, which is located in

```
C:\windrbd\windrbd-kernel.log
```

Please refer to section Troubleshooting WinDRBD for an overview of possible reasons why connections can fail.

Note that the `drbdadm up` command will fail if the resource is already up or half up. If you change anything in the configuration you have two options:

- Doing a `drbdadm down <res>` followed by a `drbdadm up <res>`
- Or you can do a `drbdadm adjust <res>`. This works for most, but not for all changes to the configuration.

4.7. Using WinDRBD devices

4.7.1. WinDRBD roles: Primary and Secondary

Most file systems and applications are not cluster aware. That means that they don't have means of synchronization when running concurrently on two (or more) different nodes, accessing (and possibly modifying) the same data.

Therefore, in a WinDRBD cluster, even though there might be up to 32 nodes participating in the cluster, only one node is allowed to read from and write to the device. This node is called the Primary node. WinDRBD can be configured to allow more than one node to be primary, but if you access them for example with an NTFS file system, your data will get corrupted almost instantaneously.

Having explained that, select one node you want to use as the primary node and execute (foo being the name of the resource):

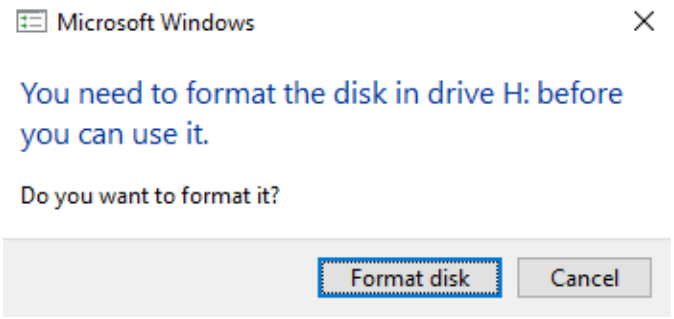
```
drbdadm primary foo --force
```

The force parameter makes WinDRBD treat data on the device as UpToDate. The force parameter is only needed the first time to tell WinDRBD to make the disk UpToDate. Do not use the force parameter later, since it may result in data loss.

We are now ready to actually write data to the WinDRBD device, for example by formatting it.

Note that in contrast to Linux DRBD, WinDRBD devices exist only on nodes that are Primary. The reason is that if we should create Windows devices when the node is Secondary, the Windows Kernel (most notably the NTFS driver) will start to cache data above the WinDRBD driver, thereby causing data corruption (since WinDRBD doesn't know about the modified data).

So, you probably will see a panel popping up, offering to format the new device (F: in our sample). This is now the WinDRBD device which we will use for I/O, in contrast to the backing storage (E: in our sample), which should never be touched (in fact, you will get an error message if you try to access it).



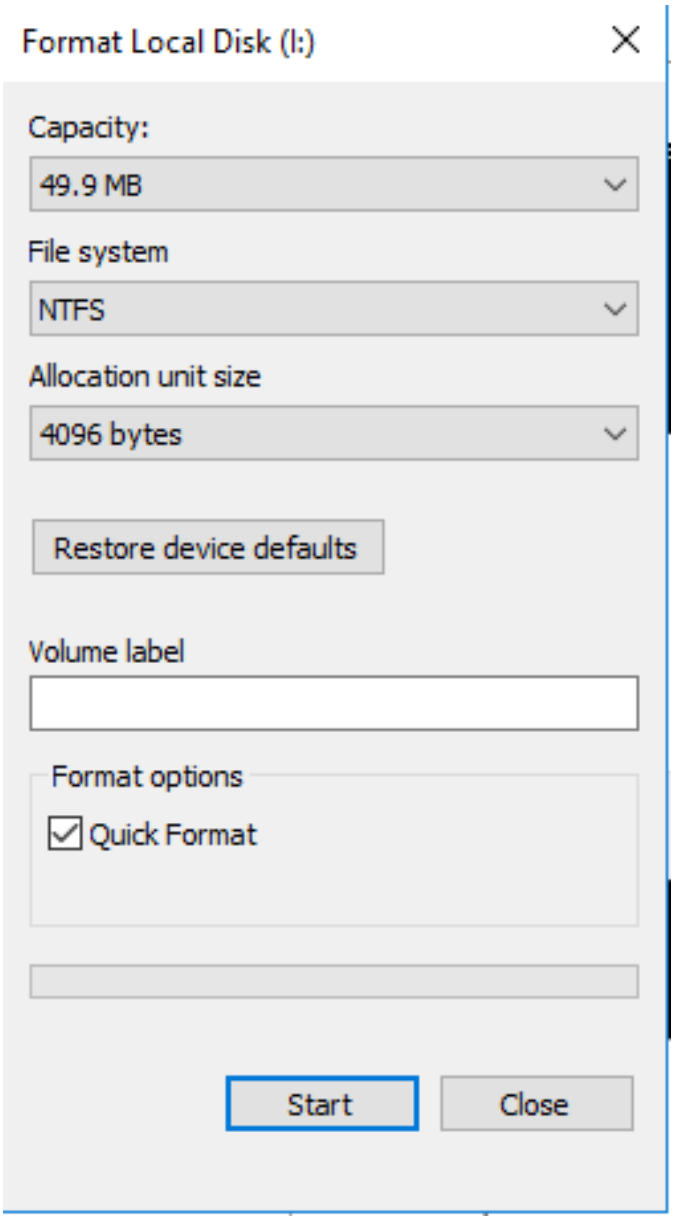
4.7.2. Formatting the device

Currently, WinDRBD allows only NTFS and RAW (for example, virtual machine images) as filesystems on top of the WinDRBD device. Since Windows doesn't support a wide range of file systems and NTFS is the most common this is not a major drawback at the moment.

Having said that, you can either format the WinDRBD device via the GUI or via the command line: the command on the command line is (run this as Administrator):

```
format f: /fs:ntfs
```

where F: is the drive letter of the WinDRBD device (not the backing device). Be sure to specify the correct drive letter to the format command, else you may destroy existing data.



4.7.3. Reading and Writing the device

Since WinDRBD is a block device which sits under the file system driver (more close to the physical drive), the WinDRBD driver works transparently to the application. That means that any application that can read or write files on an NTFS file system can do so also on an NTFS file system sitting on top of a WinDRBD device, without being modified for doing so.

For example you can open any text editor (NotePad, WordPad, Microsoft Office) to create and edit text documents.

For now, create a text file or some other data which we use later for a failover test.

4.7.4. Shutting down the WinDRBD device

To shut down the WinDRBD device, use the `drbdadm down` command:

```
drbdadm down foo
```

will bring the device down. Usually there is no need to do that manually (unless you want to manually upgrade WinDRBD without a reboot). In particular, there is no need to bring the resource down when you reboot the machine.

Chapter 5. Testing and troubleshooting WinDRBD

Congratulations, you have successfully set up a 2-node WinDRBD setup. In that chapter you will learn what to do if the backing device should fail (which is the main cause why one would use DRBD and WinDRBD).

5.1. Testing WinDRBD

DRBD and WinDRBD mirror data on-line and in real time. This means that write requests are propagated to all connected nodes as the application is running. Depending on the protocol (A, B or C) used, applications are notified immediately when the data is about to be sent (protocol A, which is fastest) or once the data has been written by the peer node (protocol C which is safest).

5.1.1. Doing a manual failover

To test if replication is working we do a manual failover test. First, on the primary node, make the node secondary (be sure to save any data on the node before doing so): on alice, do

```
drbdadm secondary foo
```

This needs to be done before another node is made primary. Then, on host bob, do

```
drbdadm primary foo
```

This time no panel offering to format the device pops up since the device is already formatted (remember that WinDRBD replicates data on-line and in real time, this also applies to data written by the format utility).

Instead you will have a new drive (F: or what ever you specified in the device clause on that node) appearing in the File Explorer.

Opening that drive will show the file you created on the other node. It should have the content you created when alice was primary.

5.1.2. I/O errors

Let us assume that the backing device fails. We can emulate this by putting the backing device on a USB stick and unplugging the USB stick while an I/O intensive application is running.

Let's say on node alice, we've configured a USB stick as backing device. Let's assume you have put an NTFS file system on the WinDRBD device.

Now, start an application that does intensive I/O on the device (if you have CygWin installed, you could do so by running a loop like:

```
while true ; do cp /dev/zero /cygdrive/f ; done
```

but there are other possibilities to create I/O.

Then unplug the USB stick.

What should happen is that WinDRBD (via the logic contained in the DRBD engine) detects the backing device failure and from there on, routes I/O requests transparently to the application to the remote node (which has a working backing device). So the application shouldn't even recognize that there was an I/O error, the only thing that may happen is the I/O gets a little bit slower.

5.2. Troubleshooting WinDRBD

This section contains some hints where to look if something goes wrong.

5.2.1. Log file

The WinDRBD kernel driver logs by sending UDP packets to a local or remote computer. By default, this is configured to log to the local host. There is a service configured that writes the kernel log messages received from the kernel driver to the log file

```
C:\windrbd\windrbd-kernel.log
```

In that file you will find log messages produced by the kernel. There is another log file, called

```
C:\windrbd\windrbd-umhelper.log
```

which contains log messages of user mode helper programs called by the kernel. You normally don't need that unless you wish to configure user mode helper scripts (which is out of the scope of this document).

5.2.2. Rotating the log file

Right now, the log file can become very big (it might grow to over several gigabytes). Currently, there is no log rotation mechanism implemented, so you have to do that manually. You first have to stop the log server, then delete the log file and then start the log server again. To do so, execute (as administrator):

```
sc stop windrbdlog
del c:\windrbd\windrbd-kernel.log
sc start windrbdlog
```

5.2.3. Connection establishment fails

The most hairy thing in configuring a WinDRBD device is to have two nodes connecting. If you experience one (or both) nodes going into standalone mode, there might be following reasons:

- A split brain occurred: This happens if data diverged on the two nodes. This can happen, for example, if both nodes are Primary without being connected. In that case you have to select one node containing valid data. On that node, execute:

  ```
  drbdadm primary --force
  ```

This tells WinDRBD to consider data on this node as valid and will discard the data on the other node.

Another possibility to recover from a split brain is to execute:

```
drbdadm connect --discard-my-data
```

This will render data on this node where it is executed as invalid.

The log file will contain a line saying Split brain detected but unresolved.

- Size mismatch (the backing devices have different sizes). In that case you will find the sizes of both nodes in the log file. Keep in mind that the meta data doesn't count for usable data, that is if you have two partitions of the same size and one has internal and the other external meta data, there will be a size mismatch.

5.2.4. Where did my data go

Finally, we will explain what happens to the backing devices in case it contained a file system before WinDRBD was configured.

In order to prevent the Windows operating system from touching the backing device, we trick Windows into thinking that it contains no file system. We do so by modifying the NTFS filesystem signature in the boot sector. It usually contains NTFS, we replace that string by DRBD.

Windows then thinks that the device is a RAW device (containing no filesystem at all). However, the data on the file system is preserved (and will be accessible via the WinDRBD device). If you wish to access the data without WinDRBD you will have to change the file system signature back to NTFS again.

However this will only work when the WinDRBD resource is down.

The command to do so is (provided that E: is the drive letter of the backing device):

```
windrbd show-filesystem E:
```

You then will find the E: drive in explorer which will contain the data you have written while the WinDRBD was active.

If you use WinDRDB again, it will automatically hide the backing device again (it won't work on an NTFS backing device).

5.2.5. Uninstall WinDRBD

To uninstall WinDRBD, either use Uninstall WinDRBD in the start menu or the entry in the Windows Add/Remove Programs facility.

Currently, to complete uninstall (to remove the driver) a reboot is required (you will get prompted).

Unless you want to keep it, delete the C:\windrbd folder also. The Uninstall routine will not delete this folder automatically, since it typically will contain files created and modified by you.

5.2.6. Getting help on the internet

To get help using WinDRBD, you have several possibilities:

- Subscribe and write to drbd-user@lists.linbit.com mailing list. This list is mainly read by people using the Linux variant of WinDRBD, however since the problems are very similar, you most likely will find somebody to help you.
- Visit the github site:

  ```
  http://www.github.com/LINBIT/windrbd
  ```

Especially if you think you have found a bug or have a feature request. We also would like to hear from you if you have successfully set up a 2-node setup with WinDRBD.

- IRC

You can find some of us on Freenode.net in #drbd.

- Linbit support:

Last but not least, in the near future (when there is a 1.0 release) there will be the possibility to get commercial 24/7 support for WinDRBD from the software's sponsor Linbit. To learn more about Linbit and their services, go to www.linbit.com or write an email to sales@linbit.com

Chapter 6. Congratulations!

You have successfully set up a 2-node WinDRBD cluster. Stay tuned for upcoming releases which will add DRBD Proxy support (for long distance disaster recovery support), LINSTOR support (for managing WinDRBD in a set of cluster nodes) and cluster manager support (for creating high availability clusters using WinDRBD).

Thank you for helping WinDRBD development by using it.

- Johannes Thoma