Clusters from Scratch - Apache, DRBD and GFS2

Creating Active/Passive and Active/Active Clusters on Fedora 12

by Andrew Beekhof

pacemaker@clusterlabs.org

Last updated on Thursday, March 11, 2010 for Pacemaker 1.0.5
# Table of Contents

## Read-Me-First

- The Scope of this Document
- Formatting conventions used in this document
- What Is Pacemaker?
- Types of Pacemaker Clusters
- Pacemaker Architecture

## Installation

- OS Installation
- Cluster Software Installation
  - Install the Cluster Software
  - Security Shortcuts

## Setup

- Before You Continue
- Finalize Node Configuration
  - Finalize Networking
  - Configure SSH
  - Short Node Names
  - Configuring OpenAIS
  - Propagate the Configuration
- Verify OpenAIS Installation
Verify Pacemaker Installation 15
Using Pacemaker Tools 16
Create an Active/Passive Cluster 19
Exploring the Existing Configuration 19
Adding a Resource 21
Perform a Failover 22
  Quorum and Two-Node Clusters 23
  Prevent Resources from Moving after Recovery 25
Apache - Adding More Services 27
  Installation 27
  Preparation 29
  Update the Configuration 30
  Ensuring Resources Run on the Same Host 31
  Controlling Resource Start/Stop Ordering 32
  Specifying a Preferred Location 33
  Manually Moving Resources Around the Cluster 34
    Giving Control Back to the Cluster 35
Using DRBD for Shared Storage 36
  Install Pre-requisites 36
  Build DRBD Packages 39
    Install the DRBD Packages 44
  Configure DRBD 46
    Create A Partition for DRBD 46
    Write the DRBD Config 47
    Initialize and Load DRBD 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Populate DRBD with Data</td>
<td>50</td>
</tr>
<tr>
<td>Configure the Cluster for DRBD</td>
<td>51</td>
</tr>
<tr>
<td>Testing Migration</td>
<td>55</td>
</tr>
<tr>
<td><strong>Conversion to Active/Active</strong></td>
<td>57</td>
</tr>
<tr>
<td>Install a Cluster Filesystem - GFS2</td>
<td>57</td>
</tr>
<tr>
<td>Setup Pacemaker-GFS2 Integration</td>
<td>59</td>
</tr>
<tr>
<td>Add the DLM service</td>
<td>59</td>
</tr>
<tr>
<td>Add the GFS2 service</td>
<td>60</td>
</tr>
<tr>
<td>Create a GFS2 Filesystem</td>
<td>63</td>
</tr>
<tr>
<td>Preparation</td>
<td>63</td>
</tr>
<tr>
<td>Create and Populate an GFS2 Partition</td>
<td>64</td>
</tr>
<tr>
<td>Reconfigure the Cluster for GFS2</td>
<td>65</td>
</tr>
<tr>
<td>Reconfigure Pacemaker for Active/Active</td>
<td>67</td>
</tr>
<tr>
<td>Recovery</td>
<td>70</td>
</tr>
<tr>
<td><strong>Configure STONITH</strong></td>
<td>71</td>
</tr>
<tr>
<td>Why You Need STONITH</td>
<td>71</td>
</tr>
<tr>
<td>What STONITH Device Should You Use</td>
<td>71</td>
</tr>
<tr>
<td>Configuring STONITH</td>
<td>71</td>
</tr>
<tr>
<td>Example</td>
<td>72</td>
</tr>
<tr>
<td><strong>Configuration Recap</strong></td>
<td>74</td>
</tr>
<tr>
<td>Final Cluster Configuration</td>
<td>74</td>
</tr>
<tr>
<td>Node List</td>
<td>75</td>
</tr>
<tr>
<td>Cluster Options</td>
<td>75</td>
</tr>
<tr>
<td>Resources</td>
<td>75</td>
</tr>
<tr>
<td>Default Options</td>
<td>75</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Fencing</td>
<td>75</td>
</tr>
<tr>
<td>Service Address</td>
<td>76</td>
</tr>
<tr>
<td>Distributed lock manager</td>
<td>76</td>
</tr>
<tr>
<td>Oracle control daemon</td>
<td>76</td>
</tr>
<tr>
<td>DRBD - Shared Storage</td>
<td>76</td>
</tr>
<tr>
<td>Cluster Filesystem</td>
<td>76</td>
</tr>
<tr>
<td>Apache</td>
<td>77</td>
</tr>
<tr>
<td>Add a Third Node</td>
<td>78</td>
</tr>
<tr>
<td>Appendix: Sample openais.conf</td>
<td>79</td>
</tr>
<tr>
<td>Appendix: Further Reading</td>
<td>81</td>
</tr>
</tbody>
</table>
Read-Me-First

The Scope of this Document

The purpose of this document is to provide a start-to-finish guide to building an example active/passive cluster with Pacemaker and show how it can be converted to an active/active one.

The example cluster will use:

- Fedora 11 as the host operating system
- OpenAIS to provide messaging and membership services,
- Pacemaker to perform resource management,
- DRBD as a cost-effective alternative to shared storage,
- GFS2 as the cluster filesystem (in active/active mode)
- The crm shell for displaying the configuration and making changes

Given the graphical nature of the Fedora install process, a number of screenshots are included. However the guide is primarily composed of commands, the reasons for executing them and their expected outputs.

Formatting conventions used in this document

command to be executed

[root@pcmk-1 ~]# command to be executed
Sample output
Part of the output to look for in particular
More output

If a command is listed and not followed by any output, then it should be assumed that the command does not produce any.
What Is Pacemaker?

Pacemaker is a cluster resource manager. It achieves maximum availability for your cluster services (aka. resources) by detecting and recovering from node and resource-level failures by making use of the messaging and membership capabilities provided by your preferred cluster infrastructure (either OpenAIS or Heartbeat).

Pacemaker’s key features include:

- Detection and recovery of node and service-level failures
- Storage agnostic, no requirement for shared storage
- Resource agnostic, anything that can be scripted can be clustered
- Support for both large and small clusters
- Optionally ensure data integrity with STONITH
- Ability to specify cluster-wide service ordering, colocation and anti-colocation
- Support for services which need to be active on multiple nodes
- Support for services with multiple modes (eg. master/slave, primary/secondary)
- Unified, scriptable, cluster shell

Types of Pacemaker Clusters

Pacemaker makes no assumptions about your environment, this allows it to support practically any redundancy configuration including Active/Active, Active/Passive, N+1, N+M, N-to-1 and N-to-N.

Two-node Active/Passive clusters using Pacemaker and DRBD are a cost-effective solution for many High Availability situations.
By supporting many nodes, Pacemaker can dramatically reduce hardware costs by allowing several active/passive clusters to be combined and share a common backup node.

When shared storage is available, every node can potentially be used for failover. Pacemaker can even run multiple copies of services to spread out the workload.
Pacemaker Architecture

At the highest level, the cluster is made up of three pieces:

- Core cluster infrastructure providing messaging and membership functionality (illustrated in red)
- Non-cluster aware components (illustrated in blue). In a Pacemaker cluster, these pieces include not only the scripts that knows how to start, stop and monitor resources, but also a local daemon that masks the differences between the different standards these scripts implement.
- A brain (illustrated in green) that processes and reacts to events from the cluster (nodes leaving or joining) and resources (eg. monitor failures) as well as configuration changes from the administrator. In response to all of these events, Pacemaker will compute the ideal state of the cluster and plot a path to achieve it. This may include moving resources, stopping nodes and even forcing them offline with remote power switches.
When combined with OpenAIS, Pacemaker also supports popular open source cluster filesystems. Due to recent standardization in the cluster filesystem community, they make use of a common distributed lock manager which makes use of OpenAIS for its messaging capabilities and Pacemaker for its membership (which nodes are up/down) and fencing services.

Even though Pacemaker also supports Heartbeat, the filesystems need to use the stack for messaging and membership and OpenAIS seems to be what they’re standardizing on. Technically it would be possible for them to support Heartbeat as well, however there seems little interest in this.

---

1 Even though Pacemaker also supports Heartbeat, the filesystems need to use the stack for messaging and membership and OpenAIS seems to be what they’re standardizing on. Technically it would be possible for them to support Heartbeat as well, however there seems little interest in this.
Installation

OS Installation

Detailed instructions for installing Fedora are available at http://docs.fedoraproject.org/install-guide/f11/ in a number of languages. The abbreviated version is:

Point your browser to http://fedoraproject.org/en/get-fedora-all, locate the Install Media section and download the install DVD that matches your hardware.

Burn the disk image to a DVD\(^2\) and boot from it. Or use the image to boot a virtual machine as I have done here. After clicking through the welcome screen, select your language and keyboard layout\(^3\)

\(^2\) http://docs.fedoraproject.org/readme-burning-isos/

Assign your machine a host name. I happen to control the clusterlabs.org domain name, so I will use that here.

You will then be prompted to indicate the machine’s physical location and to supply a root password.

Now select where you want Fedora installed. As I don’t care about any existing data, I will accept the default and allow Fedora to use the complete drive.

**IMPORTANT:** If you plan on following the DRBD or GFS2 portions of this guide, you should reserve at least 1Gb of space on each machine from which to create a shared volume.

---


The next step is to configure networking. Do not accept the default. Cluster machines should never obtain an IP address via DHCP. Here I will use the internal addresses for the clusterlab.org network.

Next choose which software should be installed. Deselect the default “Office and Productivity” as its not appropriate for a cluster node. We’ll install any needed software later. After you click next, Fedora will begin installing.
Once the node reboots, follow the on screen instructions\textsuperscript{7} to create a system user and configure the time. It is highly recommended to enable NTP on your cluster nodes. Doing so ensures all nodes agree on the current time and makes reading log files significantly easier.

Click through the next screens until you reach the login window. Click on the user you created and supply the password you indicated earlier.

\textsuperscript{7} \url{http://docs.fedoraproject.org/install-guide/f11/en-US/html/ch-firstboot.html}
**Cluster Software Installation**

**NOTE:** Installing the cluster in future versions of Fedora will be significantly easier now that the entire stack has been accepted into the distribution. However for now there are still some hoops to jump through.

Start a terminal by going to Applications -> System Tools -> Terminal

That was the last screenshot by the way, from here on in we’re going to be working from the terminal.

Switch to the super user account. You will need to supply the password you entered earlier during the installation process.

```
su -
```

Password:

```
[root@pcmk-1 ~]#
```

Note that the username (the text before the @ symbol) now indicates we’re running as the super user “root”.

© Andrew Beekhof 2009
Install the Cluster Software

The tricky part here is to make yum install openais from the server_ha-clustering repository by specifying the exact version to install. This is necessary as the version that comes with Fedora is incompatible with the one upstream currently uses to build Pacemaker packages.

```
yum install -y pacemaker
```

```
[root@pcmk-1 ~]# yum install -y pacemaker
  Setting up Install Process
  Resolving Dependencies
  ---> Running transaction check
  ---> Package pacemaker.x86_64 0:1.1.0-1.fc12 set to be updated
  --> Processing Dependency: heartbeat >= 3.0.0 for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: corosync for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: cluster-glue for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libgnutls.so.26(GNUTLS_1_4)(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libcrclcluster.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: liboib.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libibd1.so.7(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libibclient.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbesntp.so.5(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbxml.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbipis.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbipengine.so.3(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbirm.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libtransieron.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbstonlhd.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libcrncorommon.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbplumb.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbnetsnmp.so.15(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbnetsnmpagent.so.15(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbnetsnmpibs.so.15(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbpe_status.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbsensors.so.4(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbntsnmphelpers.so.15(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbstonlth.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libcrncorolpc.so.4(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libbpe RULES.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: liblrnul.so.2(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  --> Processing Dependency: libhbcibclient.so.1(64bit) for package: pacemaker-1.1.0-1.fc12.x86_64
  ---> Package cluster-glue.x86_64 0:1.0-0.11.b79635605337.fc12 set to be updated
  --> Processing Dependency: perl-TimeDate for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Processing Dependency: libOpenIPMIUtils.so.0(64bit) for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Processing Dependency: libOpenIPMIUtils-posix.so.0(64bit) for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Processing Dependency: libopenhpi.so.2(64bit) for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Processing Dependency: libopenhpi.so.2(64bit) for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Processing Dependency: libhbcibclient.so.1(64bit) for package: cluster-glue-1.0-0.11.b79635605337.fc12.x86_64
  --> Package cluster-glue-libs.x86_64 0:1.0-0.11.b79635605337.fc12 set to be updated
  --> Package corosync.x86_64 0:1.1.2-1.fc12 set to be updated
  --> Package corosync libr.x86_64 0:1.1.2-1.fc12 set to be updated
  --> Processing Dependency: librdmacm.so.1(RDMACM_1.0)(64bit) for package: corosync-1.1.2-1.fc12.x86_64
  --> Processing Dependency: libibverbs.so.1(IBVERBS_1.0)(64bit) for package: corosync-1.1.2-1.fc12.x86_64
```
Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12

--- Processing Dependency: libibverbs.so.1(IBVERBS_1.1)(64bit) for package: corosynclib-1.1.2-1.fc12.x86_64
--- Processing Dependency: libibverbs.so.1()(64bit) for package: corosynclib-1.1.2-1.fc12.x86_64
--- Processing Dependency: librdmacm.so.1()(64bit) for package: corosynclib-1.1.2-1.fc12.x86_64
--- Package gnutls.x86_64 0:2.8.5-1.fc12 set to be updated
--- Processing Dependency: libtasn1.so.3(LIBTASN1_0_3)(64bit) for package: gnutls-2.8.5-1.fc12.x86_64
--- Processing Dependency: libtasn1.so.3()(64bit) for package: gnutls-2.8.5-1.fc12.x86_64
--- Package heartbeat.x86_64 0:3.0.0-0.5.0daab7da36a8.hg.fc12 set to be updated
--- Processing Dependency: PyXML for package: heartbeat-3.0.0-0.5.0daab7da36a8.hg.fc12.x86_64
--- Package libibverbs-1ibs.x86_64 0:3.0.0-0.5.0daab7da36a8.hg.fc12 set to be updated
--- Package libesmtp.x86_64 0:1.0.4-12.fc12 set to be updated
--- Package libtasl.t.x86_64 0:1.1.26-1.fc12 set to be updated
--- Package lm_sensors-1ibs.x86_64 0:3.1.1-4.fc12 set to be updated
--- Package net-snmp-1ibs.x86_64 1:5.4.2-1.18.fc12 set to be updated
--- Package pacemaker-1ibs.x86_64 0:1.1.0-1.fc12 set to be updated
--- Package resource-agents.x86_64 0:3.0.0-1.fc12 set to be updated
--- Running transaction check
--- Package OpenIPMI-1ibs.x86_64 0:2.0.16-4.fc12 set to be updated
--- Package PyXML.x86_64 0:0.8.4-15 set to be updated
--- Package libibverbs.x86_64 0:1.1.3-3.fc12 set to be updated
--- Processing Dependency: libibverbs-driver for package: libibverbs-1.1.3-3.fc12.x86_64
--- Package librdmacm.x86_64 0:1.0.18-1.fc12 set to be updated
--- Package libtasl1.x86_64 0:2.3-3.1.fc12 set to be updated
--- Package openhpi-1ibs.x86_64 0:2.14.0-5.fc12 set to be updated
--- Package perl-TimeDate.noarch 1:1.16-11.fc12 set to be updated
--- Running transaction check
--- Package libmlx4.x86_64 0:1.0.1-3.fc12 set to be updated
--- Finished Dependency Resolution

Dependencies Resolved

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>pacemaker</td>
<td>x86_64</td>
<td>1.1.0-1.fc12</td>
<td>custom</td>
<td>545 k</td>
</tr>
<tr>
<td>Installing for dependencies:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenIPMI-libs</td>
<td>x86_64</td>
<td>2.0.16-4.fc12</td>
<td>fedora</td>
<td>434 k</td>
</tr>
<tr>
<td>PyXML</td>
<td>x86_64</td>
<td>0.8.4-15</td>
<td>fedora</td>
<td>888 k</td>
</tr>
<tr>
<td>cluster-glue</td>
<td>x86_64</td>
<td>1.0-0.11.b79635605337.hg.fc12</td>
<td>fedora</td>
<td>222 k</td>
</tr>
<tr>
<td>cluster-glue-libs</td>
<td>x86_64</td>
<td>1.0-0.11.b79635605337.hg.fc12</td>
<td>fedora</td>
<td>114 k</td>
</tr>
<tr>
<td>corosync</td>
<td>x86_64</td>
<td>1.1.2-1.fc12</td>
<td>fedora</td>
<td>144 k</td>
</tr>
<tr>
<td>corosynclib</td>
<td>x86_64</td>
<td>1.1.2-1.fc12</td>
<td>fedora</td>
<td>138 k</td>
</tr>
<tr>
<td>gnutls</td>
<td>x86_64</td>
<td>2.8.5-1.fc12</td>
<td>fedora</td>
<td>350 k</td>
</tr>
<tr>
<td>heartbeat</td>
<td>x86_64</td>
<td>3.0.0-0.5.0daab7da36a8.hg.fc12</td>
<td>fedora</td>
<td>172 k</td>
</tr>
<tr>
<td>heartbeat-libs</td>
<td>x86_64</td>
<td>3.0.0-0.5.0daab7da36a8.hg.fc12</td>
<td>fedora</td>
<td>265 k</td>
</tr>
<tr>
<td>libesmtp</td>
<td>x86_64</td>
<td>1.0.4-12.fc12</td>
<td>fedora</td>
<td>54 k</td>
</tr>
<tr>
<td>libibverbs</td>
<td>x86_64</td>
<td>1.1.3-3.fc12</td>
<td>updates</td>
<td>42 k</td>
</tr>
<tr>
<td>libmlx4</td>
<td>x86_64</td>
<td>1.0.1-3.fc12</td>
<td>updates</td>
<td>26 k</td>
</tr>
<tr>
<td>librdmacm</td>
<td>x86_64</td>
<td>1.0.10-1.fc12</td>
<td>updates</td>
<td>22 k</td>
</tr>
<tr>
<td>libtasl1</td>
<td>x86_64</td>
<td>2.3-1.fc12</td>
<td>fedora</td>
<td>232 k</td>
</tr>
<tr>
<td>libtool-ltdl</td>
<td>x86_64</td>
<td>2.2.6-15.fc12</td>
<td>fedora</td>
<td>44 k</td>
</tr>
<tr>
<td>libtasl</td>
<td>x86_64</td>
<td>1.1.26-1.fc12</td>
<td>fedora</td>
<td>397 k</td>
</tr>
<tr>
<td>lm_sensors-libs</td>
<td>x86_64</td>
<td>3.1.1-4.fc12</td>
<td>updates</td>
<td>36 k</td>
</tr>
<tr>
<td>net-snmp-libs</td>
<td>x86_64</td>
<td>1:5.4.2-2.1-18.fc12</td>
<td>fedora</td>
<td>1.3 M</td>
</tr>
<tr>
<td>openhpi-libs</td>
<td>x86_64</td>
<td>2.14.0-5.fc12</td>
<td>fedora</td>
<td>128 k</td>
</tr>
<tr>
<td>pacemaker-libs</td>
<td>x86_64</td>
<td>1.1.0-1.fc12</td>
<td>custom</td>
<td>250 k</td>
</tr>
</tbody>
</table>
perl-TimeDate  noarch     1:1.16-11.fc12    fedora     33 k
resource-agents  x86_64     3.0.5-1.fc12    updates    251 k

Transaction Summary
=======================================================================================
Install      23 Package(s)
Upgrade       0 Package(s)

Total download size: 5.9 M

Downloading Packages:
(1/23): OpenIPMI-libs-2.0.16-4.fc12.x86_64.rpm | 434 kB | 00:00
(2/23): PyXML-0.8.4-15.x86_64.rpm               | 808 kB | 00:00
(3/23): cluster-glue-1.0-0.11.b79635605337.hu.fc12.x86_64.rpm | 222 kB | 00:00
(4/23): cluster-glue-libs-1.0-0.11.b79635605337.hu.fc12.x86_64.rpm | 114 kB | 00:00
(5/23): corosync-1.1.2-1.fc12.x86_64.rpm       | 144 kB | 00:00
(6/23): corosync-libs-1.1.2-1.fc12.x86_64.rpm  | 138 kB | 00:00
(7/23): gnutls-2.8.5-1.fc12.x86_64.rpm         | 350 kB | 00:00
(8/23): heartbeat-3.0-0-0.5.0daab7da36a8.hu.fc12.x86_64.rpm | 172 kB | 00:00
(9/23): heartbeat-libs-3.0-0-0.5.0daab7da36a8.hu.fc12.x86_64.rpm | 265 kB | 00:00
(10/23): libesmtp-1.0.4-12.fc12.x86_64.rpm      | 54 kB | 00:00
(11/23): libtool-ltdl-2.2.6-15.fc12.x86_64.rpm  | 42 kB | 00:00
(12/23): libnl-4-1.0-1.fc12.x86_64.rpm          | 26 kB | 00:00
(13/23): librdmacm-1.0-10-1.fc12.x86_64.rpm    | 22 kB | 00:00
(14/23): libtasn1-2.3-1.fc12.x86_64.rpm        | 232 kB | 00:00
(15/23): libtool-ltdl-2.2.6-15.fc12.x86_64.rpm | 44 kB | 00:00
(16/23): libxml2-2.6-1.fc12.x86_64.rpm         | 397 kB | 00:00
(17/23): lm_sensors-libs-3.1.1-4.fc12.x86_64.rpm | 36 kB | 00:00
(18/23): net-snmp-libs-5.4.2.1-8.fc12.x86_64.rpm | 1.3 MB | 00:00
(19/23): openmpi-libs-2.14.8-5.fc12.x86_64.rpm | 128 kB | 00:00
(20/23): pacemaker-1.1.0-1.fc12.x86_64.rpm     | 545 kB | 00:00
(21/23): pacemaker-libs-1.1.0-1.fc12.x86_64.rpm | 250 kB | 00:00
(22/23): perl-TimeDate-1.16-11.fc12.noarch.rpm | 33 kB | 00:00
(23/23): resource-agents-3.0.5-1.fc12.x86_64.rpm | 251 kB | 00:00

Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded

Running Transaction
Installing : libtool-ltdl-2.2.6-15.fc12.x86_64 1/23
Installing : lm_sensors-libs-3.1.1-4.fc12.x86_64 2/23
Installing : 1:net-snmp-libs-5.4.2.1-18.fc12.x86_64 3/23
Installing : libxml2-2.6-1.fc12.x86_64 4/23
Installing : openmpi-libs-2.14.8-5.fc12.x86_64 5/23
Installing : PyXML-0.8.4-15.x86_64 6/23
Installing : libesmtp-1.0.4-12.fc12.x86_64 7/23
Installing : OpenIPMI-libs-2.0.16-4.fc12.x86_64 8/23
Installing : libtasn1-2.3-1.fc12.x86_64 9/23
Installing : gnutls-2.8.5-1.fc12.x86_64 10/23
Installing : 1:perl-TimeDate-1.16-11.fc12.noarch 11/23
Installing : cluster-glue-libs-1.0-0.11.b79635605337.hu.fc12.x86_64 12/23
Installing : cluster-glue-1.0-0.11.b79635605337.hu.fc12.x86_64 13/23
Installing : libtool-ltdl-2.2.6-15.fc12.x86_64 14/23
Installing : resource-agents-3.0.5-1.fc12.x86_64 15/23
Installing : librdmacm-1.0-10-1.fc12.x86_64 16/23
Installing : corosync-1.1.2-1.fc12.x86_64 17/23

© Andrew Beekhof 2009
Security Shortcuts
To simplify this guide and focus on the aspects directly connected to clustering, we will now disable the machine's firewall and SELinux installation. Both of these actions create significant security issues and should not be performed on machines that will be exposed to the outside world.

```
    sed -i.gres "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/selinux/config
    /sbin/chkconfig --del iptables
```

Now reboot all nodes so the new security settings take effect.
Setup

Before You Continue
Repeat the Installation steps so that you have 2 Fedora nodes with the cluster software installed.

For the purposes of this document, the additional node is called `pcmk-2` with address 19.168.9.42.

Finalize Node Configuration

Finalize Networking
Confirm that you can communicate with the two new nodes:

```
ping -c 3 192.168.122.102
```

```
[root@pcmk-1 ~]# ping -c 3 192.168.122.102
PING 192.168.122.102 (192.168.122.102) 56(84) bytes of data.
64 bytes from 192.168.122.102: icmp_seq=1 ttl=64 time=0.343 ms
64 bytes from 192.168.122.102: icmp_seq=2 ttl=64 time=0.402 ms
64 bytes from 192.168.122.102: icmp_seq=3 ttl=64 time=0.558 ms

--- 192.168.122.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.343/0.434/0.558/0.092 ms
```

Now we need to make sure we can communicate with the machines by their name. If you have a DNS server, add additional entries for the three machines. Otherwise, you’ll need to add the machines to `/etc/hosts`. Below are the entries for my cluster nodes:

```
grep test /etc/hosts
```

```
[root@pcmk-1 ~]# grep test /etc/hosts
192.168.122.101 pcmk-1.clusterlabs.org pcmk-1
192.168.122.102 pcmk-2.clusterlabs.org pcmk-2
```

We can now verify the setup by again using ping:

```
ping pcmk-2
```

```
[root@pcmk-1 ~]# ping -c 3 pcmk-2
PING pcmk-2.clusterlabs.org (192.168.122.101) 56(84) bytes of data.
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=1 ttl=64 time=0.164 ms
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from pcmk-1.clusterlabs.org (192.168.122.101): icmp_seq=3 ttl=64 time=0.186 ms

--- pcmk-2.clusterlabs.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.164/0.275/0.475/0.141 ms
```
**Configure SSH**

SSH is a convenient and secure way to copy files and perform commands remotely. For the purposes of this guide, we will create a key without a password (using the \-N \" option) so that we can perform remote actions without being prompted.

**NOTE:** Unprotected SSH keys, those without a password, are not recommended for servers exposed to the outside world.

Create a new key and allow anyone with that key to log in:

```plaintext
ssh-keygen -t dsa -f ~/.ssh/id_dsa -N ""
cp .ssh/id_dsa.pub .ssh/authorized_keys
```

Install the key on the other nodes and test that you can now run commands remotely, without being prompted.

```plaintext
scp -r .ssh pcmk-2:
ssh pcmk-2 -- uname -n
```
Short Node Names

During installation, we filled in the machine’s fully qualifier domain name (FQDN) which can be rather long when it appears in cluster logs and status output. See for yourself how the machine identifies itself:

```
uname -n
dnsdomainname
```

The output from the second command is fine, but we really don’t need the domain name included in the basic host details. To address this, we need to update `/etc/sysconfig/network`. This is what it should look like before we start.

```
cat /etc/sysconfig/network
```

```
NETWORKING=yes
HOSTNAME=pcmk-1.clusterlabs.org
GATEWAY=192.168.9.1
```

All we need to do now is strip off the domain name portion, which is stored elsewhere anyway.

```
sed -i.gres 's/\.[a-z].*//g' /etc/sysconfig/network
```

Now confirm the change was successful. The revised file contents should look something like this.

```
cat /etc/sysconfig/network
```

```
NETWORKING=yes
HOSTNAME=pcmk-1
GATEWAY=192.168.9.1
```

However we’re not finished. The machine wont normally see the shortened host name until about it reboots, but we can force it to update

```
source /etc/sysconfig/network
hostname $HOSTNAME
```

Now check the machine is using the correct names

```
uname -n
dnsdomainname
```

Now repeat on `pcmk-2`. 

© Andrew Beekhof 2009
Configuring OpenAIS

Choose a port number and multi-cast address.

Be sure that the values you chose do not conflict with any existing clusters you might have. For advice on choosing a multi-cast address, see [http://www.29west.com/docs/THPM/multicast-address-assignment.html](http://www.29west.com/docs/THPM/multicast-address-assignment.html)

For this document, I have chosen port 4000 and used 226.94.1.1 as the multi-cast address.

```bash
export ais_port=4000
export ais_mcast=226.94.1.1
```

Next we automatically determine the hosts address. By not using the full address, we make the configuration suitable to be copied to other nodes.

```bash
export ais_addr=`ip addr | grep "inet " | tail -n 1 | awk '{print $4}' | sed s/255/0/`
```

Display and verify the configuration options

```bash
env | grep ais_
```

Once you’re happy with the chosen values, update the OpenAIS configuration

```bash
cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf
sed -i.gres "s/.*mcastaddr:.*/mcastaddr: $ais_mcast/g" /etc/corosync/corosync.conf
sed -i.gres "s/.*mcastport:.*/mcastport: $ais_port/g" /etc/corosync/corosync.conf
sed -i.gres "s/.*bindnetaddr:.*/bindnetaddr: $ais_addr/g" /etc/corosync/corosync.conf
```

Tell Corosync to run as the root user

```bash
cat <<-END > /etc/corosync/corosync.conf
aisexec {
  user: root
  group: root
}
END
```

Finally, tell Corosync to start Pacemaker

```bash
cat <<-END > /etc/corosync/corosync.conf
  service {
```

---


9 [http://en.wikipedia.org/wiki/Multicast_address](http://en.wikipedia.org/wiki/Multicast_address)
The final configuration should look something like the sample in the appendix.

Propagate the Configuration

Now we need to copy the changes so far to the other node:

```
for f in /etc/corosync/corosync.conf /etc/hosts; do scp $f pcmk-2:$f ; done
```

```
[root@pcmk-1 ~]# for f in /etc/ais/openais.conf /etc/hosts; do scp $f pcmk-2:$f ; done
    corosync.conf                                                               100% 1528     1.5KB/s   00:00
    hosts                                                                      100%  281     0.3KB/s   00:00
[root@pcmk-1 ~]#
```

Verify OpenAIS Installation

Start OpenAIS on the first node

```
/etc/init.d/corosync start
```

```
[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync):               [  OK  ]
```

Check the cluster started correctly and that an initial membership was able to form

```
grep -e "Corosync Cluster Engine" -e "configuration file" /var/log/messages
```

```
[root@pcmk-1 ~]# grep TOTEM /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1548]: [TOTEM ] Initializing transport (UDP/IP).
Aug 27 09:05:35 pcmk-1 corosync[1548]: [TOTEM ] A processor joined or left the membership and a new membership was formed.
```

With one node functional, it's now safe to start OpenAIS on the second node as well.

```
ssh pcmk-2 -- /etc/init.d/corosync start
```

```
[root@pcmk-1 ~]# ssh pcmk-2 -- /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync):               [  OK  ]
```
Check the cluster formed correctly

```
grep TOTEM /var/log/messages
```

```
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transport (UDP/IP).
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and a new membership was formed.
Aug 27 09:12:11 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and a new membership was formed.
```

Verify Pacemaker Installation

Now that we have confirmed that OpenAIS is functional we can check the rest of the stack.

```
grep pcmk_startup /var/log/messages
```

```
Aug 27 09:05:35 pcmk-1 corosync[1540]: [pcmk ] info: pcmk_startup: Maximum core file size is: 18446744073709551615
```

Now verify the Pacemaker processes have been started

```
ps axf
```

```
PID TTY STAT TIME COMMAND
2 ? S< 0:00 [kthreadd]
3 ? S< 0:00 __ [migration/0]
... lots of processes ...
2166 pts/0 SLl 0:01 /usr/sbin/corosync
2172 ? Sls 0:00 __/lib64/heartbeat/stonithd
2173 pts/0 S 0:00 __/lib64/heartbeat/cib
2174 pts/0 S 0:00 __/lib64/heartbeat/lrmd
2175 pts/0 S 0:00 __/lib64/heartbeat/attrd
2176 pts/0 S 0:00 __/lib64/heartbeat/pengine
2177 pts/0 S 0:00 __/lib64/heartbeat/crmd
```

And finally, check for any ERRORs during startup, there shouldn’t be any, and display the cluster’s status.

```
grep ERROR: /var/log/messages | grep -v unpack_resources
```

```
crm_mon
```

```
[  root@pcmk-1 ~ ]# grep ERROR: /var/log/messages | grep -v unpack_resources
[  root@pcmk-1 ~ ]# crm_mon
============
Last updated: Thu Aug 27 16:54:55 2009
```
Using Pacemaker Tools

In the dark past, configuring Pacemaker required the administrator to read and write XML. In true UNIX style, there were also a number of different commands that specialized in different aspects of querying and updating the cluster.

Since Pacemaker 1.0, this has all changed and we have an integrated, scriptable, cluster shell that hides all the messy XML scaffolding. It even allows you to queue up several changes at once and commit them atomically.

Take some time to familiarize yourself with what it can do.

```
crm --help
```

```
[root@pcmk-1 ~]# crm --help

usage:
    crm [-D display_type]
    crm [-D display_type] args
    crm [-D display_type] [-f file]

Use crm without arguments for an interactive session.
Supply one or more arguments for a "single-shot" use.
Specify with -f a file which contains a script. Use '-' for
standard input or use pipe/redirection.

crm displays cli format configurations using a color scheme
and/or in uppercase. Pick one of "color" or "uppercase", or
use "-D color,uppercase" if you want colorful uppercase.
Get plain output by "-D plain". The default may be set in
user preferences (options).

Examples:

    # crm -f stopapp2.cli
    # crm < stopapp2.cli
    # crm resource stop global_www
    # crm status
```

The primary tool for monitoring the status of the cluster is crm_mon (also available as crm status). It can be run in a variety of modes and has a number of output options. To find out about any of the tools that come with Pacemaker, simply invoke them with the --help option or consult the included man pages. Both sets of output are created from the tool, and so will always be in sync with each other and the tool itself.

Additionally, the Pacemaker version and supported cluster stack(s) is available via the --version option.

```
crm_mon --version
```
crm_mon --help

[root@pcmk-1 ~]# crm_mon --version
crm_mon 1.0.5 for OpenAIS and Heartbeat (Build: 462f1569a43740667daf7bf0f6b521742e9eb8fa7)

Written by Andrew Beekhof

[root@pcmk-1 ~]# crm_mon --help
crm_mon - Provides a summary of cluster's current state.

Outputs varying levels of detail in a number of different formats.

Usage: crm_mon mode [options]

Options:
-?, --help   This text
-$, --version   Version information
-V, --verbose   Increase debug output

Modes:
-h, --as-html=value   Write cluster status to the named file
-w, --web-cgi   Web mode with output suitable for cgi
-s, --simple-status   Display the cluster status once as a simple one line output (suitable for nagios)
-S, --snmp-traps=value Send SNMP traps to this station
-T, --mail-to=value   Send Mail alerts to this user.  See also --mail-from, --mail-host, --mail-prefix

Display Options:
-n, --group-by-node  Group resources by node
-r, --inactive  Display inactive resources
-f, --failcounts  Display resource fail counts
-o, --operations  Display resource operation history
-t, --timing-details  Display resource operation history with timing details

Additional Options:
-i, --interval=value   Update frequency in seconds
-1, --one-shot   Display the cluster status once on the console and exit
-N, --disable-ncurses   Disable the use of ncurses
-d, --daemonize   Run in the background as a daemon
-p, --pid-file=value   (Advanced) Daemon pid file location
-F, --mail-from=value   Mail alerts should come from the named user
-H, --mail-host=value   Mail alerts should be sent via the named host
-P, --mail-prefix=value   Subjects for mail alerts should start with this string
-E, --external-agent=value   A program to run when resource operations take place.
-e, --external-recipient=value   A recipient for your program (assuming you want the program to send something to someone).

Examples:

Display the cluster's status on the console with updates as they occur:

    # crm_mon

Display the cluster's status on the console just once then exit:

    # crm_mon

Display your cluster's status, group resources by node, and include inactive resources in the list:

    # crm_mon --group-by-node --inactive
Start crm_mon as a background daemon and have it write the cluster's status to an HTML file:

```bash
# crm_mon --daemonize --as-html /path/to/docroot/filename.html
```

Start crm_mon as a background daemon and have it send email alerts:

```bash
# crm_mon --daemonize --mail-to user@example.com --mail-host mail.example.com
```

Start crm_mon as a background daemon and have it send SNMP alerts:

```bash
# crm_mon --daemonize --snmp-traps snmptrapd.example.com
```

Report bugs to pacemaker@oss.clusterlabs.org
Create an Active/Passive Cluster

Exploring the Existing Configuration

When Pacemaker starts up, it automatically records the number and details of the nodes in the cluster as well as which stack is being used and the version of Pacemaker being used.

This is what the base configuration should look like.

```bash
[root@pcmk-2 ~]# crm configure show
node pcmk-1
node pcmk-2
property $id="cib-bootstrap-options"
  dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"
  cluster-infrastructure="openais"
  expected-quorum-votes="2"
```

For those that are not of afraid of XML, you can see the raw configuration by appending “xml” to the previous command.

```bash
[root@pcmk-2 ~]# crm configure show xml
<?xml version="1.0" ?>
<configuration>
  <cluster_property_set id="cib-bootstrap-options">
    <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"/>
    <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure" value="openais"/>
    <nvpair id="cib-bootstrap-options-expected-quorum-votes" name="expected-quorum-votes" value="2"/>
  </cluster_property_set>
</configuration>
```

© Andrew Beekhof 2009
Before we make any changes, its a good idea to check the validity of the configuration.

    crm_verify -L

    [root@pcmk-1 ~]# crm_verify -L
    crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: Resource start-up disabled since no STONITH resources have been defined
    crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: Either configure some or disable STONITH with the stonith-enabled option
    crm_verify[2195]: 2009/08/27_16:57:12 ERROR: unpack_resources: NOTE: Clusters with shared data need STONITH to ensure data integrity
    Errors found during check: config not valid
    -V may provide more details
    [root@pcmk-1 ~]#

As you can see, the tool has found some errors.

In order to guarantee the safety of your data\textsuperscript{10}, Pacemaker ships with STONITH\textsuperscript{11} enabled. However it also knows when no STONITH configuration has been supplied and reports this as a problem (since the cluster would not be able to make progress if a situation requiring node fencing arose).

For now, we will disable this feature and configure it later in the Configuring STONITH section. It is important to note that the use of STONITH is highly encouraged, turning it off tells the cluster to simply pretend that failed nodes are safely powered off. Some vendors will even refuse to support clusters that have it disabled.

To disable STONITH, we set the stonith-enabled cluster option to false.

    crm configure property stonith-enabled=false
    crm_verify -L

    [root@pcmk-1 ~]# crm configure property stonith-enabled=false
    [root@pcmk-1 ~]# crm_verify -L
    [root@pcmk-1 ~]#

With the new cluster option set, the configuration is now valid.

\textsuperscript{10} If the data is corrupt, there is little point in continuing to make it available

\textsuperscript{11} A common node fencing mechanism. Used to ensure data integrity by powering off “bad” nodes.
Adding a Resource

The first thing we should do is configure an IP address. Regardless of where the cluster service(s) are running, we need a consistent address to contact them on. Here I will choose and add 192.168.9.101 as the floating address, give it the imaginative name `ClusterIP` and tell the cluster to check that its running every 30 seconds.

**Important: The chosen address must not be one already associated with a physical node**

```bash
crm configure primitive ClusterIP ocf:heartbeat:IPaddr2 \  
    params ip=192.168.9.101 cidr_netmask=32 \  
    op monitor interval=30s
```

The other important piece of information here is `ocf:heartbeat:IPaddr2`. This tells Pacemaker three things about the resource you want to add. The first field, `ocf`, is the standard to which the resource script conforms to and where to find it. The second field is specific to OCF resources and tells the cluster which namespace to find the resource script in, in this case `heartbeat`. The last field indicates the name of the resource script.

To obtain a list of the available resource classes, run

```bash
crm ra classes
```

```
[root@pcmk-1 ~]# crm ra classes
heartbeat
lsb
ocf / heartbeat pacemaker
stonith
```

To then find all the OCF resource agents provided by Pacemaker and Heartbeat, run

```bash
crm ra list ocf heartbeat
crm ra list ocf pacemaker
```

```
[root@pcmk-1 ~]# crm ra list ocf pacemaker
ClusterMon Dummy Stateful SysInfo SystemHealth controld
ping pingd
[root@pcmk-1 ~]# crm ra list ocf heartbeat
AoeTarget Dummy Stateful SysInfo SystemHealth controld
AudibleAlarm EvmsSCC Eevsms Eevsd Filesystem
Dummy
ICP IPaddr IPaddr2 IPsrcaddr
LVM LinuxSCSI MailTo ManageRAID
ManageVE Pure-FTpd Raid1 Route
SAPDatabase SAPInstance SendArp ServeRAID
SphinxSearchDaemon Squid Stateful SysInfo
VIPArip VirtualDomain WAS WAS6
WinPopup Xen Xinetd anything
apache db2 drbd eDir88
iscSILogicalUnit iSCSI tgt ids iscsi
lDirectord mysql mysql-proxy nfsServer
oracle oralsnr postgresql pingd
portblock rsyncd scsi2reservation sfex
tomcat
```

© Andrew Beekhof 2009
Now verify that the IP resource has been added and display the cluster’s status to see that it is now active.

```bash
crm configure show

crm_mon
```

[output]

Perform a Failover

Being a high-availability cluster, we should test failover of our new resource before moving on.

First, find the node on which the IP address is running.

```bash
crm resource status ClusterIP
```

[output]

Shut down OpenAIS on that machine.

```bash
ssh pcmk-1 -- /etc/init.d/corosync stop
```

[output]
Once OpenAIS is no longer running, go to the other node and check the cluster status with crm_mon.

```
[root@pcmk-2 ~]# crm_mon
==============
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
==============
Online: [ pcmk-2 ]
OFFLINE: [ pcmk-1 ]
```

There are three things to notice about the cluster's current state. The first is that, as expected, pcmk-1 is now offline. However we can also see that *ClusterIP* isn't running anywhere!

**Quorum and Two-Node Clusters**

This is because the cluster no longer has quorum, as can be seen by the text “partition WITHOUT quorum” (highlighted green) in the output above. In order to reduce the possibility of data corruption, Pacemaker’s default behavior is to stop all resources if the cluster does not have quorum.

A cluster is said to have quorum when more than half the known or expected nodes are online, or for the mathematically inclined, whenever the following equation is true:

\[
\text{total_nodes} - 1 < 2 \times \text{active_nodes}
\]

Therefore a two-node cluster only has quorum when both nodes are running, which is no longer the case for our cluster. This would normally make the creation of a two-node cluster pointless\(^2\), however it is possible to control how Pacemaker behaves when quorum is lost. In particular, we can tell the cluster to simply ignore quorum altogether.

```
  crm configure property no-quorum-policy=ignore
  crm configure show
```

\(^2\) Actually some would argue that two-node clusters are always pointless, but that is an argument for another time.
After a few moments, the cluster will start the IP address on the remaining node. Note that the cluster still does not have quorum.

```
[root@pcmk-2 ~]# crm_mon
============
Last updated: Fri Aug 28 15:30:18 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
Online: [ pcmk-2 ]
OFFLINE: [ pcmk-1 ]
ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
```

Now simulate node recovery by restarting the cluster stack on pcmk-1 and check the cluster's status.

```
/etc/init.d/corosync start

[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
```

```
[root@pcmk-1 ~]# crm_mon
============
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
Online: [ pcmk-1 pcmk-2 ]
ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-1
```

Here we see something that some may consider surprising, the IP is back running at its original location!
Prevent Resources from Moving after Recovery

In some circumstances it is highly desirable to prevent healthy resources from being moved around the cluster. Move resources almost always requires a period of downtime and for complex services like Oracle databases, this period can be quite long.

To address this, Pacemaker has the concept of resource stickiness which controls how much a service prefers to stay running where it is. You may like to think of it as the “cost” of any downtime. By default, Pacemaker assumes there is zero cost associated with moving resources and will do so to achieve “optimal” resource placement. We can specify a different stickiness for every resource, but it is often sufficient to change the default.

```
crm configure rsc_defaults resource-stickiness=100
```

If we now retry the failover test, we see that as expected `ClusterIP` still moves to `pcmk-2` when `pcmk-1` is taken offline.

```
ssh pcmk-1 -- /etc/init.d/corosync stop

ssh pcmk-2 -- crm_mon -1
```

It should be noted that Pacemaker’s definition of optimal may not always agree with that of a human’s. The order in which Pacemaker processes lists of resources and nodes create implicit preferences (required in order to create a stable solution) in situations where the administrator had not explicitly specified some.
However when we bring `pcmk-1` back online, `ClusterIP` now remains running on `pcmk-2`. 

```
[root@pcmk-1 ~]# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
[root@pcmk-1 ~]# crm_mon
============
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
1 Resources configured.
============
Online: [ pcmk-1 pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
```
Apache - Adding More Services

Now that we have a basic but functional active/passive two-node cluster, we’re ready to add some real services. We’re going to start with Apache because its a feature of many clusters and relatively simple to configure.

Installation

Before continuing, we need to make sure Apache is installed on both hosts.

```
yum install -y httpd
```

```
[root@ppcmk-1 ~]# yum install -y httpd
Setting up Install Process
Resolving Dependencies
---> Running transaction check
---> Package httpd.x86_64 0:2.2.13-2.fc12 set to be updated
---> Processing Dependency: httpd-tools = 2.2.13-2.fc12 for package: httpd-2.2.13-2.fc12.x86_64
---> Processing Dependency: apr-util-ldap for package: httpd-2.2.13-2.fc12.x86_64
---> Processing Dependency: /etc/mime.types for package: httpd-2.2.13-2.fc12.x86_64
---> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.2.13-2.fc12.x86_64
---> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2.13-2.fc12.x86_64
---> Running transaction check
---> Package apr.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package apr-util.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package apr-util-ldap.x86_64 0:1.3.9-2.fc12 set to be updated
---> Package httpd-tools.x86_64 0:2.2.13-2.fc12 set to be updated
---> Package mailcap.noarch 0:2.1.30-1.fc12 set to be updated
---> Finished Dependency Resolution

Dependencies Resolved

Transaction Summary

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>httpd</td>
<td>x86_64</td>
<td>2.2.13-2.fc12</td>
<td>rawhide</td>
<td>735 k</td>
</tr>
<tr>
<td>apr</td>
<td>x86_64</td>
<td>1.3.9-2.fc12</td>
<td>rawhide</td>
<td>117 k</td>
</tr>
<tr>
<td>apr-util</td>
<td>x86_64</td>
<td>1.3.9-2.fc12</td>
<td>rawhide</td>
<td>84 k</td>
</tr>
<tr>
<td>apr-util-ldap</td>
<td>x86_64</td>
<td>1.3.9-2.fc12</td>
<td>rawhide</td>
<td>15 k</td>
</tr>
<tr>
<td>httpd-tools</td>
<td>x86_64</td>
<td>2.2.13-2.fc12</td>
<td>rawhide</td>
<td>63 k</td>
</tr>
<tr>
<td>mailcap</td>
<td>noarch</td>
<td>2.1.30-1.fc12</td>
<td>rawhide</td>
<td>25 k</td>
</tr>
</tbody>
</table>

Total download size: 1.0 M
Also, we need the wget tool in order for the cluster to be able to check the status of the Apache server.

```
yum install -y wget
```

```
[root@pcmk-1 ~]# yum install -y wget
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.11.4-5.fc12 set to be updated
---> Finished Dependency Resolution

Dependencies Resolved

============================================================================================================
Package              Arch                   Version                          Repository               Size
============================================================================================================
Installing:
    wget                 x86_64                 1.11.4-5.fc12                    rawhide                 393 k

Transaction Summary
============================================================================================================
Install       1 Package(s)
Upgrade       0 Package(s)

Total download size: 393 k
Downloading Packages:
Preparation
First we need to create a page for Apache to serve up. On Fedora the default Apache docroot is `/var/www/html`, so we’ll create an index file there.

```bash
[root@pcmk-1 ~]# cat <<-END >/var/www/html/index.html
<html>
<body>My Test Site - pcmk-1</body>
</html>
END
```

For the moment, we will simplify things by serving up only a static site and manually sync the data between the two nodes. So run the command again on pcmk-2.

```bash
[root@pcmk-2 ~]# cat <<-END >/var/www/html/index.html
<html>
<body>My Test Site - pcmk-2</body>
</html>
END
```

Update the Configuration

At this point, Apache is ready to go, all that needs to be done is to add it to the cluster. Let’s call the resource WebSite. We need to use an OCF script called apache in the heartbeat namespace\textsuperscript{14}, the only required parameter is the path to the main Apache configuration file and we’ll tell the cluster to check once a minute that apache is still running.

\begin{verbatim}
crm configure primitive WebSite ocf:heartbeat:apache params configfile=/etc/httpd/conf/httpd.conf op monitor interval=1min

crm configure show

crm_mon

[...]

crm_mon

[...]

After a short delay, we should see the cluster start apache

\begin{verbatim}
[...]

crm_mon

[...]

Wait a moment, the WebSite resource isn’t running on the same host as our IP address!
\end{verbatim}

\textsuperscript{14} Compare the key used here ocf:heartbeat:apache with the one we used earlier for the IP address: ocf:heartbeat:IPaddr2
Ensuring Resources Run on the Same Host

To reduce the load on any one machine, Pacemaker will generally try to spread the configured resources across the cluster nodes. However, we can tell the cluster that two resources are related and need to run on the same host (or not at all). Here we instruct the cluster that `WebSite` can only run on the host that `ClusterIP` is active on. If `ClusterIP` is not active anywhere, `WebSite` will not be permitted to run anywhere.

```bash
crm configure colocation website-with-ip INFINITY: WebSite ClusterIP

crm configure show

[root@pcmk-1 ~]# crm configure colocation website-with-ip INFINITY: WebSite ClusterIP
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache
    params confiﬁgﬁle="/etc/httpd/conf/httpd.conf"
    op monitor interval="3min"
primitive ClusterIP ocf:heartbeat:IPaddr2
    params ip="192.168.9.101" cidr_netmask="32"
    op monitor interval="30s"
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options"
    dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"
    cluster-infrastructure="openais"
    expected-quorum-votes="2"
    stonith-enabled="false"
    no-quorum-policy="ignore"
    rsc_defaults $id="rsc-options"
        resource-stickiness="100"
[root@pcmk-1 ~]# crm_mon

Last updated: Fri Aug 28 16:14:34 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes conﬁgured, 2 expected votes
2 Resources conﬁgured.

[ Online: [ pcmk-1 pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
WebSite (ocf::heartbeat:apache): Started pcmk-2
```
Controlling Resource Start/Stop Ordering

When Apache starts, it binds to the available IP addresses. It doesn’t know about any addresses we add afterwards, so not only do they need to run on the same node, but we need to make sure ClusterIP is already active before we start WebSite. We do this by adding an ordering constraint. We need to give it a name (chose something descriptive like apache-after-ip), indicate that its mandatory (so that any recovery for ClusterIP will also trigger recovery of WebSite) and list the two resources in the order we need them to start.

```
crm configure order apache-after-ip mandatory: ClusterIP WebSite
```

```
crm configure show
```

```
[root@pcmk-1 ~]# crm configure order apache-after-ip mandatory: ClusterIP WebSite
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \
  params configfile="/etc/httpd/conf/httpd.conf" \
  op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
  params ip="192.168.9.101" cidr_netmask="32" \
  op monitor interval="30s"
colocation website-with-ip inf: WebSite ClusterIP
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
  dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
  cluster-infrastructure="openais" \
  expected-quorum-votes="2" \
  stonith-enabled="false" \
  no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
  resource-stickiness="100"
```
Specifying a Preferred Location

Pacemaker does not rely on any sort of hardware symmetry between nodes, so it may well be that one machine is more powerful than the other. In such cases it makes sense to host the resources there if it is available. To do this we create a location constraint. Again we give it a descriptive name (`prefer-pcmk-1`), specify the resource we want to run there (`WebSite`), how badly we’d like it to run there (we’ll use 50 for now, but in a two-node situation almost any value above 0 will do) and the host’s name.

```
crm configure location prefer-pcmk-1 WebSite 50: pcmk-1
```

```
crm configure show
```

```
crm_mon
```

Wait a minute, the resources are still on pcmk-2!

Even though we now prefer pcmk-1 over pcmk-2, that preference is (intentionally) less than the resource stickiness (how much we preferred not to have unnecessary downtime).

To see the current placement scores, you can use a tool called ptest

```
ptest -sL
```
There is a way to force them to move though...

**Manually Moving Resources Around the Cluster**

There are always times when an administrator needs to override the cluster and force resources to move to a specific location. Underneath we use location constraints like the one we created above, happily you don’t need to care. Just provide the name of the resource and the intended location, we’ll do the rest.

```
crm resource move WebSite pcmk-1
```

```
crm_mon
```

```
[root@pcmk-1 ~]# crm resource move WebSite pcmk-1
[root@pcmk-1 ~]# crm_mon

Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667da7b8f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.

Online: [ pcmk-1 pcmk-2 ]
ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-1
WebSite (ocf::heartbeat:apache): Started pcmk-1
```

Notice how the colocation rule we created has ensured that *ClusterIP* was also moved to *pcmk-1*.

For the curious, we can see the effect of this command by examining the configuration

```
crm configure show
```

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebSite ocf:heartbeat:apache \ 
  params configfile="/etc/httpd/conf/httpd.conf" \ 
  op monitor interval="1m"
primitive ClusterIP ocf:heartbeat:IPaddr2 \ 
  params ip="192.168.9.101" cidr_netmask="32" \ 
  op monitor interval="30s"
location cli-prefer-WebSite WebSite \ 
  rule $id="cli-prefer-rule-WebSite" inf: #uname eq pcmk-1
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \ 
  dc-version="1.0.5-462f1569a43740667da7b8f6b521742e9eb8fa7" \ 
  cluster-infrastructure="openais" \ 
  expected-quorum-votes="2" \ 
  stonith-enabled="false" \ 
  no-quorum-policy="ignore"
  rsc_defaults $id="rsc-options" \ 
  resource-stickiness="100"
```

Highlighted is the automated constraint used to move the resources to *pcmk-1*
Giving Control Back to the Cluster

Once we’ve finished whatever activity that required us to move the resources to pcmk-1, in our case nothing, we can then allow the cluster to resume normal operation with the `unmove` command. Since we previously configured a default stickiness, the resources will remain on `pcmk-1`.

```sh
[scm@pcmk-1 ~]$ crm resource unmove WebSite
[scm@pcmk-1 ~]$ crm configure show

---
	node pcmk-1
	node pcmk-2
	node pcmk-2

primitive WebSite ocf:heartbeat:apache
    params configfile="/etc/httpd/conf/httpd.conf"
    op monitor interval="1min"

primitive ClusterIP ocf:heartbeat:IPaddr2
    params ip="192.168.9.101" cidr_netmask="32"
    op monitor interval="30s"

location prefer-pcmk-1 WebSite 50: pcmk-1

colocation website-with-ip inf: WebSite ClusterIP

---

property $id="cib-bootstrap-options"
    dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"
    cluster-infrastructure="openais"
    expected-quorum-votes="2"
    stonith-enabled="false"
    no-quorum-policy="ignore"

rsc_defaults $id="rsc-options"
    resource-stickiness="100"
```

Note that the automated constraint is now gone. If we check the cluster status, we can also see that as expected the resources are still active on `pcmk-1`.

```sh
[scm@pcmk-1 ~]$ crm_mon

---

Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
2 Resources configured.

---

Online: [ pcmk-1 pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-1
WebSite (ocf::heartbeat:apache): Started pcmk-1
```
Using DRBD for Shared Storage

Even if you’re serving up static websites, having to manually synchronize the contents of that website to all the machines in the cluster is not ideal. For dynamic websites, such as a wiki, it’s not even an option. Not everyone care afford network-attached storage but somehow the data needs to be kept in sync. Enter DRBD which can be thought of as network based RAID-1. See http://www.drbd.org/ for more details.

Install Pre-requisites

DRBD does not currently ship with Fedora 12 and since there is a kernel component, can be sensitive to system updates which may change the kernel’s APIs and ABIs. For this reason we’ll simply build our own DRBD packages - to be sure they are a perfect match for the machine.

First we need to install a few packages that DRBD needs:

```
yum install -y flex gcc glibc-devel kernel-headers kernel-devel rpm-build
```

[output]

Setting up Install Process

Resolving Dependencies

```
---> Running transaction check
 ---> Package flex.x86_64 0:2.5.35-7.fc12 set to be updated
 ---> Package gcc.x86_64 0:4.4.2-7.fc12 set to be updated
 ---> Processing Dependency: libgomp = 4.4.2-7.fc12 for package: gcc-4.4.2-7.fc12.x86_64
 ---> Processing Dependency: cpp = 4.4.2-7.fc12 for package: gcc-4.4.2-7.fc12.x86_64
 ---> Processing Dependency: cloog-ppl >= 0.15 for package: gcc-4.4.2-7.fc12.x86_64
 ---> Processing Dependency: libgomp.so.1()(64bit) for package: gcc-4.4.2-7.fc12.x86_64
 ---> Package glibc-devel.x86_64 0:2.11-2 set to be updated
 ---> Processing Dependency: glibc-headers = 2.11-2 for package: glibc-devel-2.11-2.x86_64
 ---> Processing Dependency: glibc-headers for package: glibc-devel-2.11-2.x86_64
 ---> Package kernel-devel.x86_64 0:2.6.31.6-162.fc12 set to be installed
 ---> Package kernel-headers.x86_64 0:2.6.31.6-162.fc12 set to be updated
 ---> Package rpm-build.x86_64 0:4.7.1-6.fc12 set to be updated
 ---> Processing Dependency: patch >= 2.5 for package: rpm-build-4.7.1-6.fc12.x86_64
 ---> Processing Dependency: elfutils >= 0.128 for package: rpm-build-4.7.1-6.fc12.x86_64
 ---> Processing Dependency: pkgconfig for package: rpm-build-4.7.1-6.fc12.x86_64
 ---> Processing Dependency: unzip for package: rpm-build-4.7.1-6.fc12.x86_64
 ---> Running transaction check
 ---> Package cloog-ppl.x86_64 0:0.15.7-1.fc12 set to be updated
 ---> Processing Dependency: libgmp.so.3()(64bit) for package: cloog-ppl-0.15.7-1.fc12.x86_64
 ---> Processing Dependency: libmmpfr.so.1()(64bit) for package: cloog-ppl-0.15.7-1.fc12.x86_64
 ---> Processing Dependency: libmpfr.so.1()(64bit) for package: cloog-ppl-0.15.7-1.fc12.x86_64
 ---> Package cpp.x86_64 0:4.4.2-7.fc12 set to be updated
 ---> Processing Dependency: libgmp.so.3()(64bit) for package: cpp-4.4.2-7.fc12.x86_64
 ---> Package elfutils.x86_64 0:0.143-1.fc12 set to be updated
```
---> Processing Dependency: elfutils-libs-x86_64 = 0.143-1.fc12 for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libdw.so.1(ELFUTILS_0.127)(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libasm.so.1(ELFUTILS_1.0)(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libdw.so.1(ELFUTILS_0.126)(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libdw.so.1(ELFUTILS_0.138)(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libdw.so.1(ELFUTILS_0.122)(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libdw.so.1()(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Processing Dependency: libasm.so.1()(64bit) for package: elfutils-0.143-1.fc12.x86_64
---> Package glibc-headers.x86_64 0:2.11-2 set to be updated
---> Package libgomp.x86_64 0:4.4.2-7.fc12 set to be updated
---> Package patch.x86_64 0:2.6-1.fc12 set to be updated
---> Package unzip.x86_64 0:5.52-11.fc12 set to be updated
--> Running transaction check
---> Package elfutils-libs.x86_64 0:0.143-1.fc12 set to be updated
---> Package gmp.x86_64 0:4.3.1-5.fc12 set to be updated
---> Package mpfr.x86_64 0:2.4.1-3.fc12 set to be updated
---> Package ppl.x86_64 0:0.10.2-10.fc12 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

========================================================================================================
Package | Arch | Version | Repository | Size
========================================================================================================
Installing:
flex     | x86_64 | 2.5.35-7.fc12 | fedora | 274 k
gcc      | x86_64 | 4.4.2-7.fc12  | fedora | 10 M
glibc-devel | x86_64 | 2.11-2       | fedora | 951 k
kernel-devel | x86_64 | 2.6.31.6-162.fc12 | updates | 6.0 M
kernel-headers | x86_64 | 2.6.31.6-162.fc12 | updates | 743 k
rpm-build | x86_64 | 4.7.1-6.fc12  | fedora | 112 k
Installing for dependencies:
cloog-ppl         | x86_64 | 0.15.7-1.fc12 | fedora | 82 k
cpp                | x86_64 | 4.4.2-7.fc12  | fedora | 3.7 M
elfutils         | x86_64 | 0.143-1.fc12  | fedora | 177 k
elfutils-libs     | x86_64 | 0.143-1.fc12  | fedora | 162 k
glibc-headers    | x86_64 | 2.11-2       | fedora | 580 k
gmp               | x86_64 | 4.3.1-5.fc12  | fedora | 198 k
libgomp          | x86_64 | 4.4.2-7.fc12  | fedora | 98 k
mpfr             | x86_64 | 2.4.1-3.fc12  | fedora | 149 k
patch             | x86_64 | 2.6-1.fc12 | updates | 79 k
pkgconfig         | x86_64 | 1:0.23-9.fc12 | fedora | 68 k
ppl               | x86_64 | 0.10.2-10.fc12 | fedora | 1.1 M
unzip             | x86_64 | 5.52-11.fc12 | fedora | 125 k

Transaction Summary

========================================================================================================
Install  | 18 Package(s)
Upgrade | 0 Package(s)

Total download size: 24 M

Downloading Packages:
(1/18): cloog-ppl-0.15.7-1.fc12.x86_64.rpm | 82 kB 00:00
(2/18): cpp-4.4.2-7.fc12.x86_64.rpm | 3.7 MB 00:02
(3/18): elfutils-0.143-1.fc12.x86_64.rpm | 177 kB 00:00
(4/18): elfutils-libs-0.143-1.fc12.x86_64.rpm | 162 kB 00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test Succeeded
Running Transaction

Installing     : gmp-4.3.1-5.fc12.x86_64
Installing     : mpfr-2.4.1-3.fc12.x86_64
Installing     : cpp-4.4.2-7.fc12.x86_64
Installing     : ppl-0.10.2-10.fc12.x86_64
Installing     : cloog-ppl-0.15.7-1.fc12.x86_64
Installing     : 1:pkgconfig-0.23-9.fc12.x86_64
Installing     : libgomp-4.4.2-7.fc12.x86_64
Installing     : elfutils-libs-0.143-1.fc12.x86_64
Installing     : elfutils-0.143-1.fc12.x86_64
Installing     : patch-2.6-1.fc12.x86_64
Installing     : unzipp-5.52-11.fc12.x86_64
Installing     : kernel-headers-2.6.31.6-162.fc12.x86_64
Installing     : rpm-build-4.7.1-6.fc12.x86_64
Installing     : flex-2.5.35-7.fc12.x86_64
Installing     : glibc-devel-2.11-2.fc12.x86_64
Installing     : glibc-headers-2.11-2.fc12.x86_64
Installing     : gmp-4.3.1-5.fc12.x86_64
Installing     : kernel-devel-2.6.31.6-162.fc12.x86_64
Installing     : kernel-headers-2.6.31.6-162.fc12.x86_64

Installed:
flex.x86_64 0:2.5.35-7.fc12
glibc-devel.x86_64 0:2.11-2
glibc-headers.x86_64 0:2.6.31.6-162.fc12
kernel-devel.x86_64 0:2.6.31.6-162.fc12
rpm-build.x86_64 0:4.7.1-6.fc12

Dependency Installed:
cloog-ppl.x86_64 0:0.15.7-1.fc12
cpp.x86_64 0:4.4.2-7.fc12
elfutils-libs.x86_64 0:0.143-1.fc12
glibc-headers.x86_64 0:2.11-2
gmp.x86_64 0:4.3.1-5.fc12
elfutils.x86_64 0:0.143-1.fc12
libgomp.x86_64 0:4.4.2-7.fc12
mpfr.x86_64 0:2.4.1-3.fc12
patch.x86_64 0:2.6-1.fc12
pkgconfig.x86_64 1:0.23-9.fc12
ppl.x86_64 0:0.10.2-10.fc12
unzip.x86_64 0:5.52-11.fc12

Complete!
[root@pcmk-1 ~]#
Build DRBD Packages

Once the development packages are installed, we can begin building DRBD\textsuperscript{15}.

\begin{verbatim}
tar zxf drbd-8.3.6.tar.gz
cd drbd-8.3.6
./configure --with-km --enable-spec
sed -i.sed s/"devel = %{kernelversion}"/devel/ drbd-km.spec
rpmbuild -ba --define "_sourcedir `pwd`/.." drbd-km.spec
\end{verbatim}

\texttt{[root@pcmk-1 ~]# wget http://oss.linbit.com/drbd/8.3/drbd-8.3.6.tar.gz}
\texttt{--2009-12-08 11:04:36-- http://oss.linbit.com/drbd/8.3/drbd-8.3.6.tar.gz}
\texttt{Resolving oss.linbit.com... 212.69.161.111}
\texttt{Connecting to oss.linbit.com|212.69.161.111|:80... connected.}
\texttt{HTTP request sent, awaiting response... 200 OK}
\texttt{Length: 457464 (447K) [application/x-gzip]}
\texttt{Saving to: "drbd-8.3.6.tar.gz"}
\texttt{100%[==================================================================================] 457,469 1.12M/s in 0.4s}
\texttt{2009-12-08 11:04:37 (1.12 MB/s) - "drbd-8.3.6.tar.gz" saved [457464/457469]}

\texttt{[root@pcmk-1 ~]# tar zxf drbd-8.3.6.tar.gz}
\texttt{[root@pcmk-1 ~]# cd drbd-8.3.6}
\texttt{[root@pcmk-1 drbd-8.3.6]# make rpm}
\texttt{make: *** No rule to make target `rpm'. Stop.}
\texttt{[root@pcmk-1 drbd-8.3.6]# ./configure --with-km --enable-spec}
\texttt{checking for gcc... gcc}
\texttt{checking for C compiler default output file name... a.out}
\texttt{checking whether the C compiler works... yes}
\texttt{checking whether we are cross compiling... no}
\texttt{checking for suffix of executables...}
\texttt{checking for suffix of object files... o}
\texttt{checking whether we are using the GNU C compiler... yes}
\texttt{checking whether gcc accepts -g... yes}
\texttt{checking for gcc option to accept ISO C89... none needed}
\texttt{checking whether ln -s works... yes}
\texttt{checking for sed... /bin/sed}
\texttt{checking for grep... /bin/grep}
\texttt{checking for flex... /usr/bin/flex}
\texttt{checking for rpmbuild... /usr/bin/rpmbuild}
\texttt{checking for xsltproc... /usr/bin/xsltproc}
\texttt{checking for tar... /bin/tar}
\texttt{checking for git... no}
\texttt{checking for dpkg-buildpackage... no}
\texttt{checking for udevadm... /sbin/udevadm}
\texttt{checking for udevinfo... /usr/bin/udevinfo}
\texttt{configure: WARNING: No dpkg-buildpackage found, building Debian packages is disabled.}
\texttt{configure: WARNING: Cannot update buildtag without git. You may safely ignore this warning when building from a tarball.}

\textsuperscript{15}At the time of writing, the latest version was 8.3.6. If a later version is now available it would be advisable to try that first.
checking for /Makefile... no
configure: WARNING: Unable to find a kernel Makefile in . You will have to set KDIR correctly when invoking make.
checking for /etc/gentoo-release... no
checking for /etc/redhat-release... yes
checking for /etc/slackware-version... no
checking for /etc/debian_version... no
checking for /etc/SuSErelease... no
configure: configured for Red Hat (includes Fedora, RHEL, CentOS).
configure: creating ./config.status
config.status: creating drbd.spec
config.status: creating drbd-km.spec

The following build log is quite long and is only included for reference in case it does not work on your machine. Most people can skip to the end.

```
[root@pcmk-1 drbd-8.3.6]# rpmbuild -ba --define "_sourcedir `pwd`/../" drbd-km.spec
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.Q7iVhB
  + umask 022
  + cd /root/rpmbuild/BUILD
  + cd /root/rpmbuild/BUILD
  + rm -rf drbd-8.3.6
  + /usr/bin/gzip -dc /root/drbd-8.3.6.tar.gz
  + /bin/tar -xf -
  + STATUS=0
  + ['0' -ne 0 ']
  + cd drbd-8.3.6
  + /bin/chmod -R -f a+rX,u+w,g-w,o-w .
  + test -d /lib/modules/2.6.31.6-162.fc12.x86_64/build/.
  + KDIR=/lib/modules/2.6.31.6-162.fc12.x86_64/build
  + scripts/get_uts_release.sh
  + test 2.6.31.6-162.fc12.x86_64 = 2.6.31.6-162.fc12.x86_64
  + exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.08bAH0
  + umask 022
  + cd /root/rpmbuild/BUILD
  + cd drbd-8.3.6
  + CFLAGS='-O2 -g'
  + export CFLAGS
  + CXXFLAGS='-O2 -g'
  + export CXXFLAGS
  + FFLAGS='-O2 -g'
  + export FFLAGS
  + ./configure --host=x86_64-unknown-linux-gnu --build=x86_64-unknown-linux-gnu --target=x86_64-redhat-linux
  + --program-prefix= --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --sysconfdir=/
  + etc --datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64 --libexecdir=/usr/libexec --
  + localstatedir=/var --sharedstatedir=/var/lib --mandir=/usr/share/man --infodir=/usr/share/info --
  + without-utils --with-km --without-udev --without-xen --without-pacemaker --without-heartbeat --without-
  + rgmanager --without-bashcompletion
checking for x86_64-unknown-linux-gnu-gcc... no
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
```

checking for gcc option to accept ISO C89... none needed
checking whether ln -s works... yes
checking for sed... /bin/sed
checking for grep... /bin/grep
checking for flex... /usr/bin/flex
checking for rpmbuild... /usr/bin/rpmbuild
checking for xsltproc... /usr/bin/xsltproc
checking for tar... /bin/tar
checking for git... no
checking for dpkg-buildpackage... no
checking for udevadm... /sbin/udevadm
checking for udevinfo... /usr/bin/udevinfo
configure: WARNING: No dpkg-buildpackage found, building Debian packages is disabled.
configure: WARNING: Cannot update buildtag without git. You may safely ignore this warning when building from a tarball.
checking for /Makefile... no
configure: WARNING: Unable to find a kernel Makefile in . You will have to set KDIR correctly when invoking make.
checking for /etc/gentoo-release... no
checking for /etc/redhat-release... yes
checking for /etc/slackware-version... no
checking for /etc/debian_version... no
configure: configured for Red Hat (includes Fedora, RHEL, CentOS).
configure: creating ./config.status
cfg...status: creating Makefile
config.status: creating user/Makefile
config.status: creating scripts/Makefile
config.status: creating documentation/Makefile
+ echo kernelversion=2.6.31.6-162.fc12.x86_64
kernelversion=2.6.31.6-162.fc12.x86_64
+ echo 'kversion=%{kversion}'
kversion=%{kversion}
+ echo 'krelver=2.6.31.6-162.fc12.x86_64'
krelver=2.6.31.6-162.fc12.x86_64
+ make module KDIR=/lib/modules/2.6.31.6-162.fc12.x86_64/build
make[1]: Entering directory `/root/rpmbuild/BUILD/drbd-8.3.6/drbd'
    Calling toplevel makefile of kernel source tree, which I believe is in
    KDIR=/lib/modules/2.6.31.6-162.fc12.x86_64/build
    ~/rpmbuild/BUILD/drbd-8.3.6/drbd
    Using unmodified drbd_config.h
make -C /lib/modules/2.6.31.6-162.fc12.x86_64/build SUBDIRS=/root/rpmbuild/BUILD/drbd-8.3.6/drbd modules
make[2]: Entering directory `/usr/src/kernels/2.6.31.6-162.fc12.x86_64'
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_buildtag.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_bitmap.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_proc.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_receiver.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_worker.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_requester.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_acceptor.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/drbd_actlog.o
 CC [M] /root/rpmbuild/BUILD/drbd-8.3.6/drbd/lru_cache.o
Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12

42
Install the DRBD Packages
The completed build process will store the result in the ~/RPMS/x86_64/ directory and all that is required now is to install the ones we need with YUM.

    yum install -y drbd-utils drbd-pacemaker

Now install the kernel module we built

cd /root/rpmbuild/RPMS/`uname -m`
yum localinstall -y --nogpgcheck drbd-km-*.rpm

[root@pcmk-1 x86_64]# yum localinstall -y --nogpgcheck drbd-utils-8.3.6-*.rpm drbd-pacemaker-8.3.6-*

Setting up Local Package Process
Examining drbd-utils-8.3.6-1.fc12.x86_64.rpm: drbd-utils-8.3.6-1.fc12.x86_64
Marking drbd-utils-8.3.6-1.fc12.x86_64.rpm to be installed

custom | 1.2 kB 00:00
Examining drbd-pacemaker-8.3.6-1.fc12.x86_64.rpm: drbd-pacemaker-8.3.6-1.fc12.x86_64
Marking drbd-pacemaker-8.3.6-1.fc12.x86_64.rpm to be installed
Examining drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64.rpm: drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64
Marking drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64.rpm to be installed

Resolving Dependencies
---> Running transaction check
---> Package drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64 set to be updated
---> Package drbd-pacemaker.x86_64 0:8.3.6-1.fc12 set to be updated
---> Package drbd-utils.x86_64 0:8.3.6-1.fc12 set to be updated
---> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository Size
========================================================================================================
Installing:
drbd-km-2.6.31.6_162.fc12.x86_64 x86_64 8.3.6-12.fc12 /drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64 4.2 M
drbd-pacemaker x86_64 8.3.6-1.fc12 /drbd-pacemaker-8.3.6-1.fc12.x86_64 40 k
drbd-utils x86_64 8.3.6-1.fc12 /drbd-utils-8.3.6-1.fc12.x86_64 580 k

Transaction Summary
========================================================================================================
Install  3 Package(s)
Upgrade  0 Package(s)

Total size: 4.8 M
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing : drbd-utils-8.3.6-1.fc12.x86_64 1/3
Installing : drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64 2/3
Installing : drbd-pacemaker-8.3.6-1.fc12.x86_64 3/3

Installed:
By default DRBD configures itself to start when the machine is powered on, however since we want the cluster to manage it, we will need to disable this behavior:

```
chkconfig --del drbd
```

We could rebuild the drbd package on `pcmk-2`, however if they share the same architecture (x86_64 in this case) we can reuse the ones we built for `pcmk-1`. Assuming this is the case for you, copy them to `pcmk-2` and install:

```
cd /root/rpmbuild/RPMS/
scp drbd-km-*.*rpm pcmk-2:
ssh pcmk-2 -- yum install -y drbd-utils drbd-pacemaker
```

```
scp drbd-*.rpm pcmk-2:
```

```
drbd-8.3.6-1.fc12.x86_64.rpm                                          100%   21KB  20.7KB/s   00:00
drbd-bash-completion-8.3.6-1.fc12.x86_64.rpm                       100%  5260     5.1KB/s   00:00
drbd-heartbeat-8.3.6-1.fc12.x86_64.rpm                           100%  6716     6.6KB/s   00:00
drbd-km-2.6.31.6_162.fc12.x86_64-8.3.6-12.fc12.x86_64.rpm         100% 1271KB   1.2MB/s   00:00
drbd-pacemaker-8.3.6-1.fc12.x86_64.rpm                           100%  18KB  17.7KB/s   00:00
drbd-udev-8.3.6-1.fc12.x86_64.rpm                                 100%  4103     4.0KB/s   00:00
drbd-utils-8.3.6-1.fc12.x86_64.rpm                                100%  265KB 264.8KB/s   00:00
drbd-xen-8.3.6-1.fc12.x86_64.rpm                                   100%  6690     6.5KB/s   00:00
```

```
[...]
```

```
scp drbd-km-*.rpm pcmk-2:
```

```
scp drbd-*.rpm pcmk-2:
```

```
drbd-8.3.6-1.fc12.x86_64.rpm                                          100%   21KB  20.7KB/s   00:00
```

```
[...]
```

```
[...]
```

```
[...]
```

```
clustering from scratch - DRBD, GFS2 and Apache on Fedora 12
```

```
45
```
Configure DRBD

Before we configure DRBD, we need to set aside some disk for it to use.

Create A Partition for DRBD

If you have more than 1Gb free, feel free to use it. For this guide however, 1Gb is plenty of space for a single html file and sufficient for later holding the GFS2 metadata.

```
 lvcreate -n drbd-demo -L 1G VolGroup 
 lv
```

Repeat this on the second node, be sure to use the same size partition.

```
 lv
 lvcreate -n drbd-demo -L 1G VolGroup 
 lv
```

Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12
Write the DRBD Config

There is no series of commands for build a DRBD configuration, so simply copy the configuration below to /etc/drbd.conf

Detailed information on the directives used in this configuration (and other alternatives) is available from http://www.drbd.org/users-guide/ch-configure.html

Be sure to use the names and addresses of your nodes if they differ from the ones used in this guide.

global {
    usage-count yes;
}
common {
    protocol C;
}
resource wwwdata {
    meta-disk internal;
    device /dev/drbd1;
    syncer {
        verify-alg sha1;
    }
    net {
        allow-two-primaries;
    }
    on pcmk-1 {
        disk /dev/mapper/VolGroup-drbd--demo;
        address 192.168.122.101:7789;
    }
    on pcmk-2 {
        disk /dev/mapper/VolGroup-drbd--demo;
        address 192.168.122.102:7789;
    }
}
Initialize and Load DRBD

With the configuration in place, we can now perform the DRBD initialization

```
[drbdadm create-md wwwwdata

[drbdadm create-md wwwwdata

md_offset 12578816
al_offset 12546048
bm_offset 12541952

Found some data
==> This might destroy existing data! <==

Do you want to proceed?
[need to type 'yes' to confirm] yes

Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
```
Repeat on the second node

```bash
drbdadm --force create-md wwwdata
modprobe drbd
drbdadm up wwwdata
cat /proc/drbd
```

```bash
[root@pcmk-2 ~]# drbdadm --force create-md wwwdata
Writing meta data....
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
[root@pcmk-2 ~]# modprobe drbd
WARNING: Deprecated config file /etc/modprobe.conf, all config files belong into /etc/modprobe.d/.
[root@pcmk-2 ~]# drbdadm up wwwdata
[root@pcmk-2 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f0b6e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57

1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:12248
```

Now we need to tell DRBD which set of data to use. Since both sides contain garbage, we can run the following on pcmk-1:

```bash
drbdadm -- --overwrite-data-of-peer primary wwwdata
cat /proc/drbd
```

```bash
[root@pcmk-1 ~]# drbdadm -- --overwrite-data-of-peer primary wwwdata
[root@pcmk-1 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f0b6e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57

1: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r----
   ns:2184 nr:0 dw:0 dr:2472 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:10064
   [========] sync'ed: 33.4% (10064/12248)K
   finish: 0:00:37 speed: 240 (240) K/sec
[root@pcmk-1 ~]# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f0b6e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57

1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
   ns:12248 nr:0 dw:0 dr:12536 al:1 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
```

pcmk-1 is now in the Primary state which allows it to be written to. Which means its a good point at which to create a filesystem and populate it with some data to serve up via our WebSite resource.
Populate DRBD with Data

```
mkfs.ext4 /dev/drbd1
[root@pcmk-1 ~]# mkfs.ext4 /dev/drbd1
mke2fs 1.41.4 (27-Jan-2009)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
3072 inodes, 12248 blocks
612 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=12582912
2 block groups
8192 blocks per group, 8192 fragments per group
1536 inodes per group
Superblock backups stored on blocks:
 8 1 9 3
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Now mount the newly created filesystem so we can create our index file

```
mount /dev/drbd1 /mnt/
cat <<-END >/mnt/index.html
<html>
<body>My Test Site - drbd</body>
</html>
END
```

```
umount /dev/drbd1
[root@pcmk-1 ~]# mount /dev/drbd1 /mnt/
[root@pcmk-1 ~]# cat <<-END >/mnt/index.html
> <html>
> <body>My Test Site - drbd</body>
> </html>
> END
[root@pcmk-1 ~]# umount /dev/drbd1
```

And finally, confirm the data is in sync between the two nodes

```
derbdadm verify wwwdata
echo $?
```

```
[root@pcmk-1 ~]# drbdadm verify wwwdata
[root@pcmk-1 ~]# echo $?
0
```
Configure the Cluster for DRBD

One handy feature of the crm shell is that you can use it in interactive mode to make several changes atomically.

First we launch the shell. The prompt will change to indicate you’re in interactive mode.

```
[ root@pcmk-1 ~ ] # crm
```

```
cib crm(live) #
```

Next we must create a working copy or the current configuration. This is where all our changes will go. The cluster will not see any of them until we say its ok. Notice again how the prompt changes, this time to indicate that we’re no longer looking at the live cluster.

```
cib new drbd
```

```
cib crm(live) # cib new drbd
INFO: drbd shadow CIB created
```

```
crm(drbd) #
```

Now we can create our DRBD clone and display the revised configuration.

```
configure primitive wwwdrbd ocf:linbit:drbd params drbd_resource=wwwdata op monitor interval=60s
configure ms WebData wwwdrbd meta master-max=1 master-node-max=1 
  clone-max=2 clone-node-max=1 notify=true
```

```
configure show
```

```
crm(drbd) # configure primitive ocf:linbit:drbd WebData params drbd_resource=wwwdata op monitor interval=60s
```

```
crm(drbd) # configure ms WebDataClone WebData meta master-max=1 master-node-max=1 
  clone-max=2 clone-node-max=1 notify=true
```

```
crm(drbd) # configure show
```

```
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd 
  params drbd_resource='wwwdata' 
  op monitor interval='60s'
primitive WebSite ocf:heartbeat:apache 
  params configfile='/etc/httpd/conf/httpd.conf' 
  op monitor interval='1min'
primitive ClusterIP ocf:heartbeat:IPaddr2 
  params ip='192.168.9.101' cidr_netmask='32' 
  op monitor interval='30s'
ms WebDataClone WebData 
  meta master-max='1' clone-max='2' clone-node-max='1' notify='true'
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation website-with-ip inf: WebSite ClusterIP
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" 
  dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" 
  cluster-infrastructure="openais" 
  expected-quorum-votes="2" 
  stonith-enabled="false" 
  no-quorum-policy="ignore"
rsc_defaults $id='rsc-options' 
  resource-stickiness="100"
```
Once we’re happy with the changes, we can tell the cluster to start using them and use `crm_mon` to check everything is functioning.

```
cib commit drbd
quit
crm_mon
```

```
crm(drbd)# cib commit drbd
INFO: committed 'drbd' shadow CIB to the cluster
cri(drbd)# quit
bye
[root@pcmk-1 ~]# crm_mon
==========
Last updated: Tue Sep  1 09:37:13 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667da7b8f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
3 Resources configured.
==========

Online: [ pcmk-1 pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr):        Started pcmk-1
WebSite (ocf::heartbeat:apache):        Started pcmk-1
Master/Slave Set: WebDataClone
    Masters: [ pcmk-2 ]
    Slaves: [ pcmk-1 ]
```

Now that DRBD is functioning we can configure a Filesystem resource to use it. In addition to the filesystem’s definition, we also need to tell the cluster where it can be located (only on the DRBD Primary) and when it is allowed to start (after the Primary was promoted).

Once again we’ll use the shell’s interactive mode

```
crm

cib new fs

configure primitive WebFS ocf:heartbeat:Filesystem \ 
    params device="/dev/mapper/VolGroup-drbd--demo" directory="/var/www/html" fstype="ext4"

configure colocation fs_on_drbd inf: WebFS WebDataClone:Master

configure order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
```

We also need to tell the cluster that Apache needs to run on the same machine as the filesystem and that it must be active before Apache can start.
configure colocation WebSite-with-WebFS inf: WebSite WebFS
configure order WebSite-after-WebFS inf: WebFS WebSite
configure show

[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \
  params drbd_resource="wwwdata" \
  op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \
  params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache \
  params configfile="/etc/httpd/conf/httpd.conf" \
  op monitor interval="1m"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
  params ip="192.168.9.101" cidr_netmask="32" \
  op monitor interval="30s"
ms WebDataClone WebData \
  meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
location WebSite-with-WebFS inf: WebSite WebFS
location fs_on_drbd inf: WebFS WebDataClone:Master
location website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
  dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7" \
  cluster-infrastructure="openais" \
  expected-quorum-votes="2" \
  stonith-enabled="false" \
  no-quorum-policy="ignore"
rsck_defaults $id="rsck-options" \
  resource-stickiness="100"
After reviewing the new configuration, we again upload it and watch the cluster put it into effect.

```plaintext
cib commit fs
quit
crm_mon
```

```
crm(fs)# cib commit fs
INFO: commited 'fs' shadow CIB to the cluster
```
Testing Migration

We could shut down the active node again, but another way to safely simulate recovery is to put the node into what is called “standby mode”. Nodes in this state tell the cluster that they are not allowed to run resources. Any resources found active there will be moved elsewhere. This feature can be particularly useful when updating the resources’ packages.

Put the local node into standby mode and observe the cluster move all the resources to the other node. Note also that the node’s status will change to indicate that it can no longer host resources.

```
crm node standby

crm_mon
```

```
[root@pcmk-1 ~]# crm node standby
[root@pcmk-1 ~]# crm_mon
============
Last updated: Tue Sep  1 10:09:57 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667da7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
4 Resources configured.
============

Node pcmk-1: standby
Online: [ pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
WebSite (ocf::heartbeat:apache): Started pcmk-2
Master/Slave Set: WebDataClone
  Masters: [ pcmk-2 ]
  Stopped: [ WebData:1 ]
WebFS (ocf::heartbeat:Filesystem): Started pcmk-2
```
Once we’ve done everything we needed to on pcmk-1 (in this case nothing, we just wanted to see the resources move), we can allow the node to be a full cluster member again.

```
crm node online

crm_mon
```

```
[root@pcmk-1 ~]# crm node online
[root@pcmk-1 ~]# crm_mon

==============
Last updated: Tue Sep 1 10:13:25 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
4 Resources configured.
==============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-2
WebSite (ocf::heartbeat:apache): Started pcmk-2
Master/Slave Set: WebDataClone
   Masters: [ pcmk-2 ]
   Slaves: [ pcmk-1 ]
WebFS (ocf::heartbeat:Filesystem): Started pcmk-2
```

Notice that our resource stickiness settings prevent the services from migrating back to pcmk-1.
Conversion to Active/Active

The primary requirement for an Active/Active cluster is that the data required for your services are available, simultaneously, on both machines. Pacemaker makes no requirement on how this is achieved, you could use a SAN if you had one available, however since DRBD supports multiple Primaries, we can also use that.

The only hitch is that we need to use a cluster-aware filesystem (and the one we used earlier with DRBD, ext4, is not one of those). Both OCFS2 and GFS2 are supported, however here we will use GFS2 which comes with Fedora 12.

**Install a Cluster Filesystem - GFS2**

The first thing to do is install gfs2-utils on each machine.

**NOTE:** Unfortunately there are a number of changes needed to gfs-pcmk that aren’t yet in Fedora-12 (ETA is January 2010). In order to use GFS2 before then, you’ll need to recompile the sources for the cluster package with the following patch:  
http://git.fedorahosted.org/git/cluster.git?p=cluster.git;a=commitid;h=f554d932a817401beaf078e6ee4bb2522a3d0cc

```bash
yum install -y gfs2-utils gfs-pcmk
```

```
[root@pcmk-1 ~]# yum install -y gfs2-utils gfs-pcmk
Setting up Install Process
Resolving Dependencies
---> Running transaction check
    --> Package gfs-pcmk.x86_64 0:3.0.5-2.fc12 set to be updated
    --> Processing Dependency: libSaCkpt.so.3(OPENAIS_CKPT_B.01.01)(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Processing Dependency: dlm-pcmk for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Processing Dependency: libbcs.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Processing Dependency: libdlmcontrol.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Processing Dependency: liblogthread.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Processing Dependency: libSaCkpt.so.3()(64bit) for package: gfs-pcmk-3.0.5-2.fc12.x86_64
    --> Package gfs2-utils.x86_64 0:3.0.5-2.fc12 set to be updated
    --> Running transaction check
    --> Package clusterlib.x86_64 0:3.0.5-2.fc12 set to be updated
    --> Package dlm-pcmk.x86_64 0:3.0.5-2.fc12 set to be updated
    --> Package openaislib.x86_64 0:1.1.8-1.fc12 set to be updated
    --> Finished Dependency Resolution

Dependencies Resolved

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
<th>Repository</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>gfs-pcmk</td>
<td>x86_64</td>
<td>3.0.5-2.fc12</td>
<td>custom</td>
<td>181 k</td>
</tr>
</tbody>
</table>
```

© Andrew Beekhof 2009
**Transaction Summary**

<table>
<thead>
<tr>
<th>Install</th>
<th>5 Package(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade</td>
<td>0 Package(s)</td>
</tr>
</tbody>
</table>

Total download size: 541 k

**Download Packages:**

(1/5): clusterlib-3.0.5-2.fc12.x86_64.rpm | 65 kB 00:00
(2/5): dlm-pcmk-3.0.5-2.fc12.x86_64.rpm | 93 kB 00:00
(3/5): gfs-pcmk-3.0.5-2.fc12.x86_64.rpm | 101 kB 00:00
(4/5): gfs2-utils-3.0.5-2.fc12.x86_64.rpm | 208 kB 00:00
(5/5): openaislib-1.1.0-1.fc12.x86_64.rpm | 76 kB 00:00

Total | 992 kB/s 541 kB 00:00

Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded

Running Transaction

<table>
<thead>
<tr>
<th>Installing</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing</td>
<td>clusterlib-3.0.5-2.fc12.x86_64</td>
<td>1/5</td>
</tr>
<tr>
<td>Installing</td>
<td>openaislib-1.1.0-1.fc12.x86_64</td>
<td>2/5</td>
</tr>
<tr>
<td>Installing</td>
<td>dlm-pcmk-3.0.5-2.fc12.x86_64</td>
<td>3/5</td>
</tr>
<tr>
<td>Installing</td>
<td>gfs-pcmk-3.0.5-2.fc12.x86_64</td>
<td>4/5</td>
</tr>
<tr>
<td>Installing</td>
<td>gfs2-utils-3.0.5-2.fc12.x86_64</td>
<td>5/5</td>
</tr>
</tbody>
</table>

Installed:

- gfs-pcmk.x86_64 0:3.0.5-2.fc12
- gfs2-utils.x86_64 0:3.0.5-2.fc12

Dependency Installed:

- clusterlib.x86_64 0:3.0.5-2.fc12
- dlm-pcmk.x86_64 0:3.0.5-2.fc12
- openaislib.x86_64 0:1.1.0-1.fc12

Complete!

[root@pcmk-1 x86_64]#
Setup Pacemaker-GFS2 Integration

GFS2 needs two services to be running, the first is the user-space interface to the kernel’s distributed lock manager (DLM). The DLM is used to co-ordinate which node(s) can access a given file (and when) and integrates with Pacemaker to obtain node membership\(^\text{16}\) information and fencing capabilities.

The second service is GFS2’s own control daemon which also integrates with Pacemaker to obtain node membership data.

Add the DLM service

The DLM control daemon needs to run on all active cluster nodes, so we will use the shell's interactive mode to create a cloned resource.

```
crm
  cib new stack-glue
  configure primitive dlm ocf:pacemaker:controld op monitor interval=120s
  configure clone dlm-clone dlm meta interleave=true
  configure show
```

```
[root@pcmk-1 ~]# crm
crm(live)# cib new stack-glue
INFO: stack-glue shadow CIB created
crm(stack-glue)# configure primitive dlm ocf:pacemaker:controld op monitor interval=120s
crm(stack-glue)# configure clone dlm-clone dlm meta interleave=true
crm(stack-glue)# configure show xml
crm(stack-glue)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd 
  params drbd_resource="wwwdata" 
  op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem 
  params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache 
  params configfile="/etc/httpd/conf/httpd.conf" 
  op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 
  params ip="192.168.9.101" cidr_netmask="32" 
  op monitor interval="30s"
primitive dlm ocf:pacemaker:controld 
  op monitor interval="120s"
ms WebDataClone WebData
  meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone dlm-clone dlm 
  meta interleave="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options"
  dc-version="1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7"
```

\(^{16}\) The list of nodes the cluster considers to be available
Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response:

```
cib commit stack-glue
quit
```

```
crm_mon
```

Add the GFS2 service

Once the DLM is active, we can add the GFS2 control daemon. Use the `crm` shell to create the gfs-control cluster resource:

```
crm
```

```
cib new gfs-glue
```

```
configure primitive gfs-control ocf:pacemaker:controld
   params daemon=gfs_controld.pcmk args="-g 0"
   op monitor interval=120s
```

```
configure clone gfs-clone gfs-control meta interleave=true
```

```
[root@pcmk-1 ~]# crm
```

```
crm(live)# cib new gfs-glue --force
INFO: gfs-glue shadow CIB created
```

```
crm(gfs-glue)# configure primitive gfs-control ocf:pacemaker:controld params daemon=gfs_controld.pcmk args="-g 0" op monitor interval=120s
```

```
crm(gfs-glue)# configure clone gfs-clone gfs-control meta interleave=true
```

Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12

© Andrew Beekhof 2009
Now ensure Pacemaker only starts the gfs-control service on nodes that also have a copy of the dlm service (created above) already running

```bash
crm(gfs-glue)# configure colocation gfs-with-dlm INFINITY: gfs-clone dlm-clone
crm(gfs-glue)# configure order start-gfs-after-dlm mandatory: dlm-clone gfs-clone
```

Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response

```bash
configure show
cib commit gfs-glue
quit
```

```bash
crm_mon
```
```bash
crm(gfs-glue)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd 
  params drbd_resource="wwndata" 
  op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem 
  params device="/dev/drbd/by-res/wwndata" directory="/var/www/html" fstype="ext4"
primitive WebSite ocf:heartbeat:apache 
  params configfile="/etc/httpd/conf/httpd.conf" 
  op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 
  params ip="192.168.9.101" cidr_netmask="32" 
  op monitor interval="30s"
primitive dlm ocf:pacemaker:controld 
  op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld 
  params daemon="gfs_controld.pcmk" args="-g 0" 
  op monitor interval="120s"
ms WebDataClone WebData 
  meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone dlm-clone dlm 
  meta interleaves="true"
clone gfs-clone gfs-control 
  meta interleaves="true"
location prefer-pcmk-1 WebSite 50: pcmk-1
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSite ClusterIP
order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSite
order apache-after-ip inf: ClusterIP WebSite
order start-gfs-after-dlm inf: dlm-clone gfs-clone
property $id="cib-bootstrap-options" 
  dc-version="1.0.5-462f1569a43740667dad7b0f6521742e9eb8fa7" 
  cluster-infrastructure="openais" 
  expected-quorum-votes="2" 
  stonith-enabled="false" 
```
no-quorum-policy="ignore"

rsc_defaults $id="rsc-options" \
  resource-stickiness="100"
crm(gfs-glue)# cib commit gfs-glue
INFO: commited 'gfs-glue' shadow CIB to the cluster
crm(gfs-glue)# quit
bye
[root@pcmk-1 ~]# crm_mon
==============
Last updated: Thu Sep 3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b8f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
==============

Online: [ pcmk-1 pcmk-2 ]

WebSite (ocf::heartbeat:apache):        Started pcmk-2
 Master/Slave Set: WebDataClone
    Masters: [ pcmk-1 ]
    Slaves: [ pcmk-2 ]
ClusterIP (ocf::heartbeat:IPaddr):        Started pcmk-2
 Clone Set: dlm-clone
    Started: [ pcmk-2 pcmk-1 ]
 Clone Set: gfs-clone
    Started: [ pcmk-2 pcmk-1 ]
WebFS  (ocf::heartbeat:Filesystem):    Started pcmk-1
Create a GFS2 Filesystem

Preparation

Before we do anything to the existing partition, we need to make sure it is unmounted. We do this by tell the cluster to stop the WebFS resource. This will ensure that other resources (in our case, Apache) using WebFS are not only stopped, but stopped in the correct order.

```
crm_resource --resource WebFS --set-parameter target-role --meta --parameter-value Stopped
```

```
crm_mon
```

Note that Apache and WebFS have both been stopped.

© Andrew Beekhof 2009
Create and Populate an GFS2 Partition

Now that the cluster stack and integration pieces are running smoothly, we can create an GFS2 partition.

Important: This will erase all previous content stored on the DRBD device. Ensure you have a copy of any important data.

We need to specify a number of additional parameters when creating a GFS2 partition.

First we must use the -p option to specify that we want to use the Kernel's DLM. Next we use -j to indicate that it should reserve enough space for two journals (one per node accessing the filesystem).

Lastly, we use -t to specify the lock table name. The format for this field is clustername:fsname. For the fsname, we just need to pick something unique and descriptive and since we haven’t specified a clustername yet, we will use the default (pcmk).

To specify an alternate name for the cluster, locate the service section containing “name: pacemaker” in corosync.conf and insert the following line anywhere inside the block:

```
clustername: myname
```

Do this on each node in the cluster and be sure to restart them before continuing.

```
mkfs.gfs2 -p lock_dlm -j 2 -t pcmk:web /dev/drbd1
```

```
[root@pcmk-1 ~]# mkfs.gfs2 -t pcmk:web -p lock_dlm -j 2 /dev/vdb
This will destroy any data on /dev/vdb.
It appears to contain: data
Are you sure you want to proceed? [y/n] y
```

```
Device:                    /dev/vdb
Blocksize:                 4096
Device Size                1.00 GB (131072 blocks)
Filesystem Size:           1.00 GB (131070 blocks)
Journals:                  2
Resource Groups:           2
Locking Protocol:          "lock_dlm"
Lock Table:                "pcmk:web"
UUID:                      6B776F46-177B-BAF8-2C2B-292C0E078613
```

Then (re)populate the new filesystem with data (web pages). For now we'll create another variation on our home page.

```
mount /dev/drbd1 /mnt/
cat <<-END >/mnt/index.html
<html>
<body>My Test Site - GFS2</body>
</html>
END
```

```
umount /dev/drbd1
drbdadm verify wwwdata
```

© Andrew Beekhof 2009

Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12
Reconfigure the Cluster for GFS2

crm

cib new GFS2

configure delete WebFS

configure primitive WebFS ocf:heartbeat:Filesystem \ 
  params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"

Now that we’ve recreated the resource, we also need to recreate all the constraints that used it. This is because the shell will automatically remove any constraints that referenced WebFS.

configure colocation WebSite-with-WebFS inf: WebSite WebFS

configure colocation fs_on_drbd inf: WebFS WebDataClone:Master

configure order WebFS-after-WebData inf: WebDataClone:promote WebFS:start

configure order WebSite-after-WebFS inf: WebFS WebSite

configure colocation WebFS-with-gfs-control INFINITY: WebFS gfs-clone

configure order start-WebFS-after-gfs-control mandatory: gfs-clone WebFS

configure show

© Andrew Beekhof 2009
Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response

cib commit GFS2

quit

crm_mon

[root@pcmk-1 ~]# crm_mon

Last updated: Thu Sep 3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b0f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.

Online: [ pcmk-1 pcmk-2 ]

WebSite (ocf::heartbeat:apache): Started pcmk-2
Master/Slave Set: WebDataClone
   Masters: [ pcmk-1 ]
   Slaves: [ pcmk-2 ]
ClusterIP (ocf::heartbeat:IPaddr):        Started pcmk-2
Clone Set: dlm-clone
    Started: [ pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
    Started: [ pcmk-2 pcmk-1 ]
WebFS   (ocf::heartbeat:Filesystem):    Started pcmk-1

Reconfigure Pacemaker for Active/Active

Almost everything is in place. Recent versions of DRBD are capable of operating in Primary/Primary mode and the filesystem we’re using is cluster aware. All we need to do now is reconfigure the cluster to take advantage of this.

This will involve a number of changes, so we’ll again use interactive mode.

    crm
    cib new active

There’s no point making the services active on both locations if we can’t reach them, so let’s first clone the IP address. Cloned IPaddr2 resources use an `iptables` rule to ensure that each request only processed by one of the two clone instances. The additional `meta` options tell the cluster how many instances of the clone we want (one “request bucket” for each node) and that if all other nodes fail, then the remaining node should hold all of them. Otherwise the requests would be simply discarded.

    configure clone WebIP ClusterIP  \
        meta globally-unique="true" clone-max="2" clone-node-max="2"

Now we must tell the ClusterIP how to decide which requests are processed by which hosts. To do this we must specify the `clusterip_hash` parameter.

Open the ClusterIP resource

    configure edit  ClusterIP

And add the following to the params line

    clusterip_hash="sourceip"

So that the complete definition looks like:

    primitive ClusterIP ocf:heartbeat:IPaddr2  \
        params ip="192.168.9.101" cidr_netmask="32" clusterip_hash="sourceip"  \
        op monitor interval="30s"
Notice how any constraints that referenced `ClusterIP` have been updated to use `WebIP` instead. This is an additional benefit of using the `crm` shell.

Next we need to convert the filesystem and Apache resources into clones. Again, the shell will automatically update any relevant constraints.
configure clone WebFSClone WebFS
configure clone WebSiteClone WebSite

The last step is to tell the cluster that it is now allowed to promote both instances to be Primary (aka. Master).

configure edit WebDataClone

Change master-max to 2

configure show

```
crm(active)# configure clone WebFSClone WebFS
crm(active)# configure clone WebSiteClone WebSite
crm(active)# configure edit WebDataClone
crm(active)# configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \ 
   params drbd_resource="wwndata" \ 
   op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \ 
   params device="/dev/drbd/by-res/wwndata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache \ 
   params configfile="/etc/httpd/conf/httpd.conf" \ 
   op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \ 
   params ip="192.168.9.101" cidr_netmask="32" clusterip_hash="sourceip" \ 
   op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \ 
   op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \ 
   params daemon="gfs_controld.pcmk" args="-g 0" \ 
   op monitor interval="120s"
ms WebDataClone WebData \ 
   meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone WebFSClone WebFS
clone WebIP ClusterIP \ 
   meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
clone dlm-dm dlm \ 
   meta interleave="true"
clone gfs-clone gfs-control \ 
   meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-dm
ocation website-with-ip inf: WebSiteClone WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
order apache-after-ip inf: WebIP WebSiteClone
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
order start-gfs-after-dm inf: dlm-dm gfs-clone
property $id="cib-bootstrap-options" \ 
   dc-version="1.0.5-462f1569ad3740667daf7b0f6b521742e9eb8fa7" \ 
   cluster-infrastructure="openais" \ 
   expected-quorum-votes="2" \ 
   stonith-enabled="false"
```
Review the configuration before uploading it to the cluster, quitting the shell and watching the cluster's response.

```
cib commit active
quit
```

```
crm_mon
```

```
crm(active)# cib commit active
INFO: committed 'active' shadow CIB to the cluster
crm(active)# quit
bye
```

```
[root@pcmk-1 ~]# crm_mon
```

```
=======================
Last updated: Thu Sep 3 21:37:27 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.0.5-462f1569a43740667daf7b8f6b521742e9eb8fa7
2 Nodes configured, 2 expected votes
6 Resources configured.
=======================

Online: [pcmk-1 pcmk-2 ]

Master/Slave Set: WebDataClone
  Masters: [pcmk-1 pcmk-2 ]
Clone Set: dlm-clone
  Started: [pcmk-2 pcmk-1 ]
Clone Set: gfs-clone
  Started: [pcmk-2 pcmk-1 ]
Clone Set: WebIP
  Started: [pcmk-1 pcmk-2 ]
Clone Set: WebFSClone
  Started: [pcmk-1 pcmk-2 ]
Clone Set: WebSiteClone
  Started: [pcmk-1 pcmk-2 ]

```

Recovery
Configure STONITH

Why You Need STONITH

STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and it protects your data from being corrupted by rouge nodes or concurrent access.

Just because a node is unresponsive, this doesn’t mean it isn’t accessing your data. The only way to be 100% sure that your data is safe, is to use STONITH so we can be certain that the node is truly offline, before allowing the data to be accessed from another node.

STONITH also has a role to play in the event that a clustered service cannot be stopped. In this case, the cluster uses STONITH to force the whole node offline, thereby making it safe to start the service elsewhere.

What STONITH Device Should You Use

It is crucial that the STONITH device can allow the cluster to differentiate between a node failure and a network one.

The biggest mistake people make in choosing a STONITH device is to use remote power switch (such as many onboard IMPI controllers) that shares power with the node it controls. In such cases, the cluster cannot be sure if the node is really offline, or active and suffering from a network fault.

 Likewise, any device that relies on the machine being active (such as SSH-based “devices” used during testing) are inappropriate.

Configuring STONITH

1. Find the correct driver: stonith -L

2. Since every device is different, the parameters needed to configure it will vary. To find out the parameters required by the device: stonith -t {type} -n

   Hopefully the developers chose names that make sense, if not you can query for some additional information by finding an active cluster node and running:

   lrmadmin -M stonith {type} pacemaker

   The output should be XML formatted text containing additional parameter descriptions

3. Create a file called stonith.xml containing a primitive resource with a class of stonith, a type of {type} and a parameter for each of the values returned in step 2

4. Create a clone from the primitive resource if the device can shoot more than one node and supports multiple simultaneous connections.

5. Upload it into the CIB using cibadmin: cibadmin -C -o resources --xml-file stonith.xml
Example

Assuming we have an IBM BladeCenter containing our two nodes and the management interface is active on 192.168.9.31, then we would chose the `external/ibmrsa` driver in step 2 and obtain the following list of parameters:

```
stonith -t external/ibmrsa -n
```

Assuming we know the username and password for the management interface, we would create a STONITH resource with the shell command:

```
crm

cib new active

configure primitive rsa-fencing stonith::external/ibmrsa \\
    params hostname="pcmk-1 pcmk-2" ipaddr=192.168.9.31 userid=mgmt passwd=abc123
    type=ibm \\
    op monitor interval="60s"

configure clone Fencing rsa-fencing
```

And finally, since we disabled it earlier, we need to re-enable STONITH

```
crm(stonith)# configure property stonith-enabled="true"
```

```
crm(stonith)# configure show
```

```
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd \\
    params drbd_resource="wwwdata" \\
    op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem \\
    params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache \\
    params configfile="/etc/httpd/conf/httpd.conf" \\
    op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \\
    params ip="192.168.9.101" cidr_netmask="32" clusterip_hash="sourceip" \\
    op monitor interval="30s"
primitive dlm ocf:pacemaker:controld \\
    op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld \\
    params daemon="gfs_control.pcmk" args="-g 0" \\
    op monitor interval="120s"
primitive rsa-fencing stonith::external/ibmrsa \\
```
params hostname="pcmk-1 pcmk-2" ipaddr=192.168.9.31 user id=mgmt passwd=abc123 type=ibm \
  op monitor interval="60s"
ms WebDataClone WebData \
  meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone Fencing rsa-fencing
clone WebFSClone WebFS
clone WebIP ClusterIP \
  meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
clone dim-clone diml \
  meta interleaved="true"
clone gfs-clone gfs-control \
  meta interleaved="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dim-clone
colocation website-with-ip inf: WebSiteClone WebIP
 order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
 order WebSite-after-WebFS inf: WebFSClone WebSiteClone
 order apache-after-ip inf: WebIP WebSiteClone
 order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
 order start-gfs-after-dlm inf: gfs-clone dim-clone gfs-clone
property $id="cib-bootstrap-options" \
  dc-version="1.8.5-462f1569a43740667da7beb8fa7" \
  cluster-infrastructure="openais" \
  expected-quorum-votes="2" \
  stonith-enabled="true" \
  no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
  resource-stickiness="100"
Configuration Recap

Final Cluster Configuration

crm configure show

```
[root@pcmk-1 ~]# crm configure show
node pcmk-1
node pcmk-2
primitive WebData ocf:linbit:drbd 
  params drbd_resource="wwndata" 
  op monitor interval="60s"
primitive WebFS ocf:heartbeat:Filesystem 
  params device="/dev/drbd/by-res/wwndata" directory="/var/www/html" fstype="gfs2"
primitive WebSite ocf:heartbeat:apache 
  params configfile="/etc/httpd/conf/httpd.conf" 
  op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 
  params ip="192.168.9.101" cidr_netmask="32" clusterip_hash="sourceip" 
  op monitor interval="30s"
primitive dlm ocf:pacemaker:controld 
  op monitor interval="120s"
primitive gfs-control ocf:pacemaker:controld 
  params daemon="gfs_controld.pcmk" args="-g 0" 
  op monitor interval="120s"
primitive rsa-fencing stonith::external/ibmrsa 
  params hostname="pcmk-1 pcmk-2" ipaddr=192.168.9.31 userid=mgmt passwd=abc123 type=ibm 
  op monitor interval="60s"
ms WebDataClone WebData 
  meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone Fencing rsa-fencing
clone WebFSClone WebFS
clone WebIP ClusterIP 
  meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
clone dlm-clone dlm 
  meta interleave="true"
clone gfs-clone gfs-control 
  meta interleave="true"
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
colocation gfs-with-dlm inf: gfs-clone dlm-clone
colocation website-with-ip inf: WebSiteClone WebIP
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
order apache-after-ip inf: WebIP WebSiteClone
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
```
Node List

The list of cluster nodes is automatically populated by the cluster.

```
node pcmk-1
node pcmk-2
```

Cluster Options

This is where the cluster automatically stores some information about the cluster:

- **dc-version** - the version (including upstream source-code hash) of Pacemaker used on the DC
- **cluster-infrastructure** - the cluster infrastructure being used (heartbeat or openais)
- **expected-quorum-votes** - the maximum number of nodes expected to be part of the cluster and where the admin can set options that control the way the cluster operates

- **stonith-enabled=true** - Make use of STONITH
- **no-quorum-policy=ignore** - Ignore loss of quorum and continue to host resources.

```
property $id="cib-bootstrap-options" 
  dc-version="1.0.5-462f1569a43740667daf7b0f65b21742e9eb8fa7" 
  cluster-infrastructure="openais" 
  expected-quorum-votes="2" 
  stonith-enabled="true" 
  no-quorum-policy="ignore"
```

Resources

Default Options

Here we configure cluster options that apply to every resource.

- **resource-stickiness** - Specify the aversion to moving resources to other machines

```
property $id="cib-bootstrap-options" 
  dc-version="1.0.5-462f1569a43740667daf7b0f65b21742e9eb8fa7" 
  cluster-infrastructure="openais" 
  expected-quorum-votes="2" 
  stonith-enabled="true" 
  no-quorum-policy="ignore"
```

```
rsc_defaults $id="rsc-options" 
  resource-stickiness="100"
```

Fencing

```
primitive rsa-fencing stonith::external/ibmrsa 
  params hostname="pcmk-1 pcmk-2" ipaddr=192.168.9.31 userid=mgmt passwd=abc123 type=ibm 
  op monitor interval="60s"
clone Fencing rsa-fencing
```
Service Address

Users of the services provided by the cluster require an unchanging address with which to access it. Additionally, we cloned the address so it will be active on both nodes. An `iptables` rule (created as part of the resource agent) is used to ensure that each request only processed by one of the two clone instances. The additional `meta` options tell the cluster that we want two instances of the clone (one “request bucket” for each node) and that if one node fails, then the remaining node should hold both.

```
primitive ClusterIP ocf:heartbeat:IPaddr2 \
  params ip="192.168.9.101" cidr_netmask="32" clusterip_hash="sourceip" \
  op monitor interval="30s"
clone WebIP ClusterIP \
  meta globally-unique=true clone-max=2 clone-node-max=2
```

Distributed lock manager

Cluster filesystems like GFS2 require a lock manager. This service starts the daemon that provides user-space applications (such as the GFS2 daemon) with access to the in-kernel lock manager. Since we need it to be available on all nodes in the cluster, we have it cloned.

```
primitive dlm ocf:pacemaker:controld \
  op monitor interval="120s"
clone dlm-clone dlm \
  meta interleave=true
```

Oracle control daemon

GFS2 also needs a user-space/kernel bridge that runs on every node. So here we have another clone, however this time we must also specify that it can only run on machines that are also running the DLM (colocation constraint) and that it can only be started after the DLM is running (order constraint). Additionally, the gfs-control clone should only care about the DLM instances it is paired with, so we need to set the interleave option.

```
primitive gfs-control ocf:pacemaker:controld \
  params daemon="gfs_controld.pcmk" args="-g 0" \
  op monitor interval="120s"
clone gfs-clone gfs-control \
  meta interleave=true
colocation gfs-with-dlm inf: gfs-clone dlm-clone
order start-gfs-after-dlm inf: dlm-clone gfs-clone
```

DRBD - Shared Storage

Here we define the DRBD service and specify which DRBD resource (from drbd.conf) it should manage. We make it a master/slave resource and, in order to have an active/active setup, allow both instances to be promoted by specifying master-max=2. We also set the notify option so that the cluster will tell DRBD agent when it’s peer changes state.

```
primitive WebData ocf:linbit:drbd \
  params drbd_resource="wwwdata" \
  op monitor interval="60s"
ms WebDataClone WebData \
  meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify=true
```

Cluster Filesystem

The cluster filesystem ensures that files are read and written correctly. We need to specify the block device (provided by DRBD), where we want it mounted and that we are using GFS2. Again it is a clone because it is intended to be active on both nodes. The additional constraints ensure that it can only be started on nodes with active gfs-control and drbd instances.
Apache
Lastly we have the actual service, Apache. We need only tell the cluster where to find it's main configuration file and restrict it to running on nodes that have the required filesystem mounted and the IP address active.
Add a Third Node
Appendix: Sample openais.conf

# Please read the openais.conf.5 manual page
compatibility: whitetank

aisexec {
    # Run as root - this is necessary to be able to manage resources with Pacemaker
    user: root
    group: root
}

service {
    # Load the Pacemaker Cluster Resource Manager
    ver: 0
    name: pacemaker
    use_mgmd: yes
    use_logd: yes
}

totem {
    version: 2

    # How long before declaring a token lost (ms)
    token: 5000

    # How many token retransmits before forming a new configuration
    token_retransmits_before_loss_const: 10

    # How long to wait for join messages in the membership protocol (ms)
    join: 1000

    # How long to wait for consensus to be achieved before starting a new round of membership configuration
    # (ms)
    consensus: 2500

    # Turn off the virtual synchrony filter
    vsftype: none

    # Number of messages that may be sent by one processor on receipt of the token
    max_messages: 20

    # Stagger sending the node join messages by 1..send_join ms
    send_join: 45

    # Limit generated nodeids to 31-bits (positive signed integers)
    clear_node_high_bit: yes
 Cluster from Scratch - DRBD, GFS2 and Apache on Fedora 12
Appendix: Further Reading

Project Website

http://www.clusterlabs.org

Cluster Commands

A comprehensive guide to cluster commands has been written by Novell and can be found at:


OpenAIS

http://www.openais.org