Pacemaker 1.1

Clusters from Scratch

Creating Active/Passive and Active/Active Clusters on Fedora

Andrew Beekhof

Pacemaker 1.1 Clusters from Scratch
Creating Active/Passive and Active/Active Clusters on Fedora Edition 5

Author	Andrew Beekhof	andrew@beekhof.net
Translator	Raoul Scarazzini	rasca@miamammausalinux.org
Translator	Dan Frîncu	df.cluster@gmail.com

Copyright © 2009-2012 Andrew Beekhof.

The text of and illustrations in this document are licensed under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA")[1].

In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

In addition to the requirements of this license, the following activities are looked upon favorably:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.

2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.

3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy or CD-ROM expression of the author(s) work.

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:

1. Fedora 13 as the host operating system

2. Corosync to provide messaging and membership services,

3. Pacemaker to perform resource management,

4. DRBD as a cost-effective alternative to shared storage,

5. GFS2 as the cluster filesystem (in active/active mode)

6. 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.

[1] An explanation of CC-BY-SA is available at *http://creativecommons.org/licenses/by-sa/3.0/*

Table of Contents

List of Figures

Preface

Table of Contents

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

[1] https://fedorahosted.org/liberation-fonts/

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla[2] against the product **Pacemaker.**

When submitting a bug report, be sure to mention the manual's identifier: *Clusters_from_Scratch*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

[2] *http://bugs.clusterlabs.org*

Read-Me-First

Table of Contents

1.1. The Scope of this Document

Computer clusters can be used to provide highly available services or resources. The redundancy of multiple machines is used to guard against failures of many types.

This document will walk through the installation and setup of simple clusters using the Fedora distribution, version 14.

The clusters described here will use Pacemaker and Corosync to provide resource management and messaging. Required packages and modifications to their configuration files are described along with the use of the Pacemaker command line tool for generating the XML used for cluster control.

Pacemaker is a central component and provides the resource management required in these systems. This management includes detecting and recovering from the failure of various nodes, resources and services under its control.

When more in depth information is required and for real world usage, please refer to the *Pacemaker Explained*[1] manual.

1.2. 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 Corosync 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
- Supports STONITH for ensuring data integrity
- Supports large and small clusters
- Supports both quorate and resource driven clusters

[1] http://www.clusterlabs.org/doc/

- Supports practically any redundancy configuration
- Automatically replicated configuration that can be updated from any node
- Ability to specify cluster-wide service ordering, colocation and anti-colocation
- Support for advanced service types
  - Clones: for services which need to be active on multiple nodes
  - Multi-state: for services with multiple modes (eg. master/slave, primary/secondary)
- Unified, scriptable, cluster shell

1.3. Pacemaker Architecture

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

- Non-cluster aware components (illustrated in green). These pieces include the resources themselves, scripts that start, stop and monitor them, and also a local daemon that masks the differences between the different standards these scripts implement.
- Resource management Pacemaker provides the brain (illustrated in blue) that processes and reacts to events regarding the cluster. These events include nodes joining or leaving the cluster; resource events caused by failures, maintenance, scheduled activities; and other administrative actions. Pacemaker will compute the ideal state of the cluster and plot a path to achieve it after any of these events. This may include moving resources, stopping nodes and even forcing them offline with remote power switches.
- Low level infrastructure Corosync provides reliable messaging, membership and quorum information about the cluster (illustrated in red).

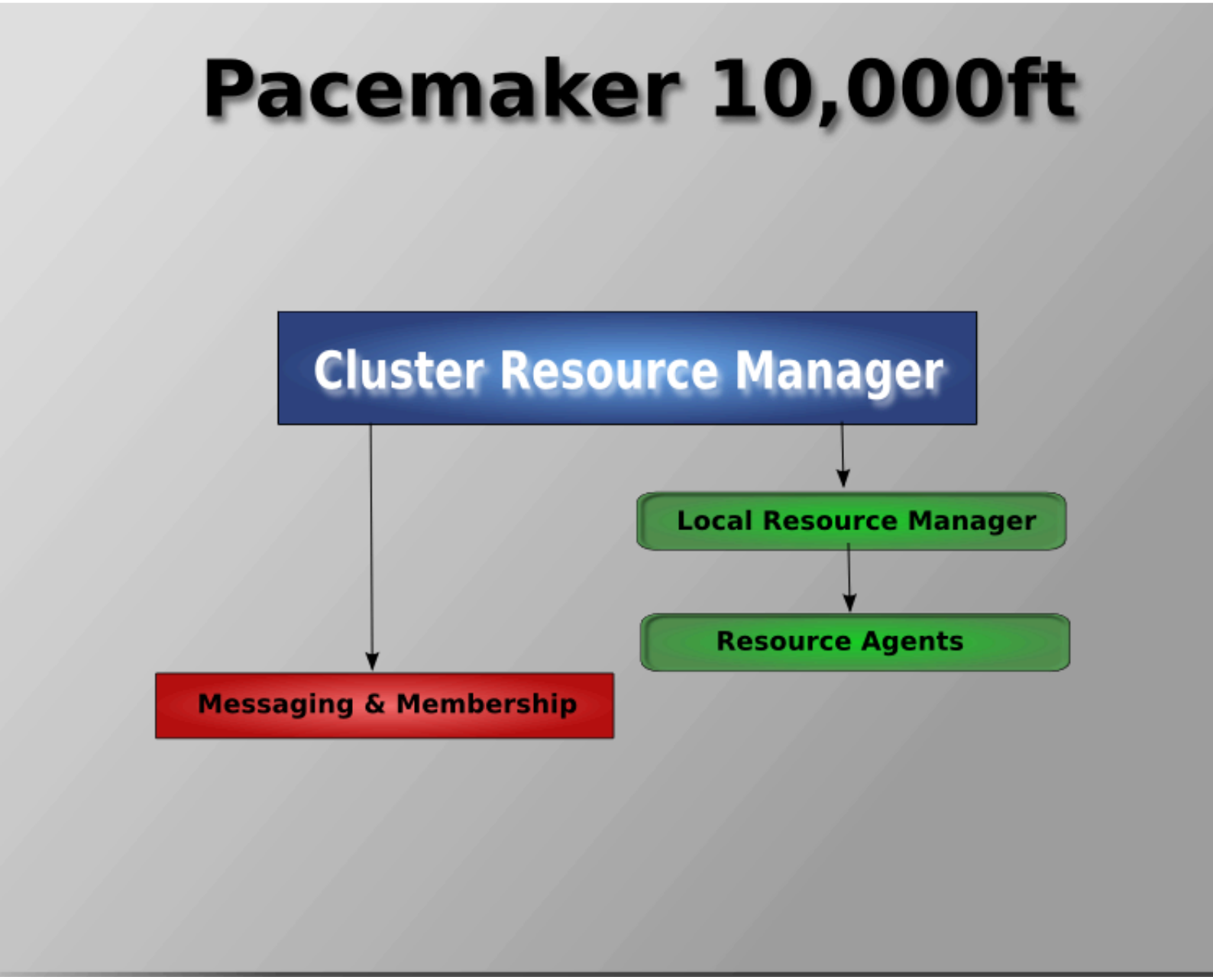


Figure 1.1. Conceptual Stack Overview

When combined with Corosync, Pacemaker also supports popular open source cluster filesystems. [2]

Due to recent standardization within the cluster filesystem community, they make use of a common distributed lock manager which makes use of Corosync for its messaging capabilities and Pacemaker for its membership (which nodes are up/down) and fencing services.

[2] Even though Pacemaker also supports Heartbeat, the filesystems need to use the stack for messaging and membership and Corosync 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.

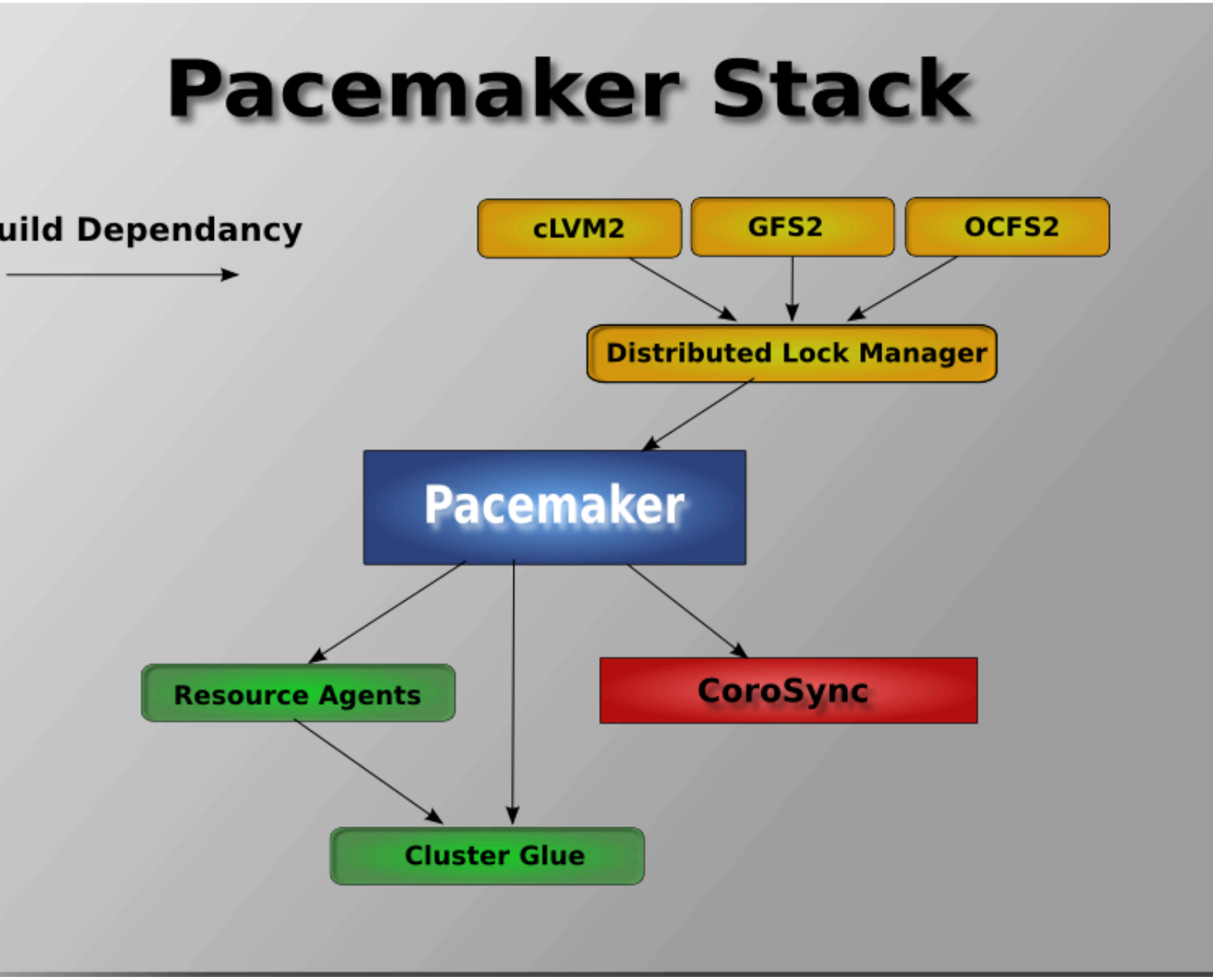


Figure 1.2. The Pacemaker Stack

1.3.1. Internal Components

Pacemaker itself is composed of four key components (illustrated below in the same color scheme as the previous diagram):

- CIB (aka. Cluster Information Base)
- CRMd (aka. Cluster Resource Management daemon)
- PEngine (aka. PE or Policy Engine)
- STONITHd

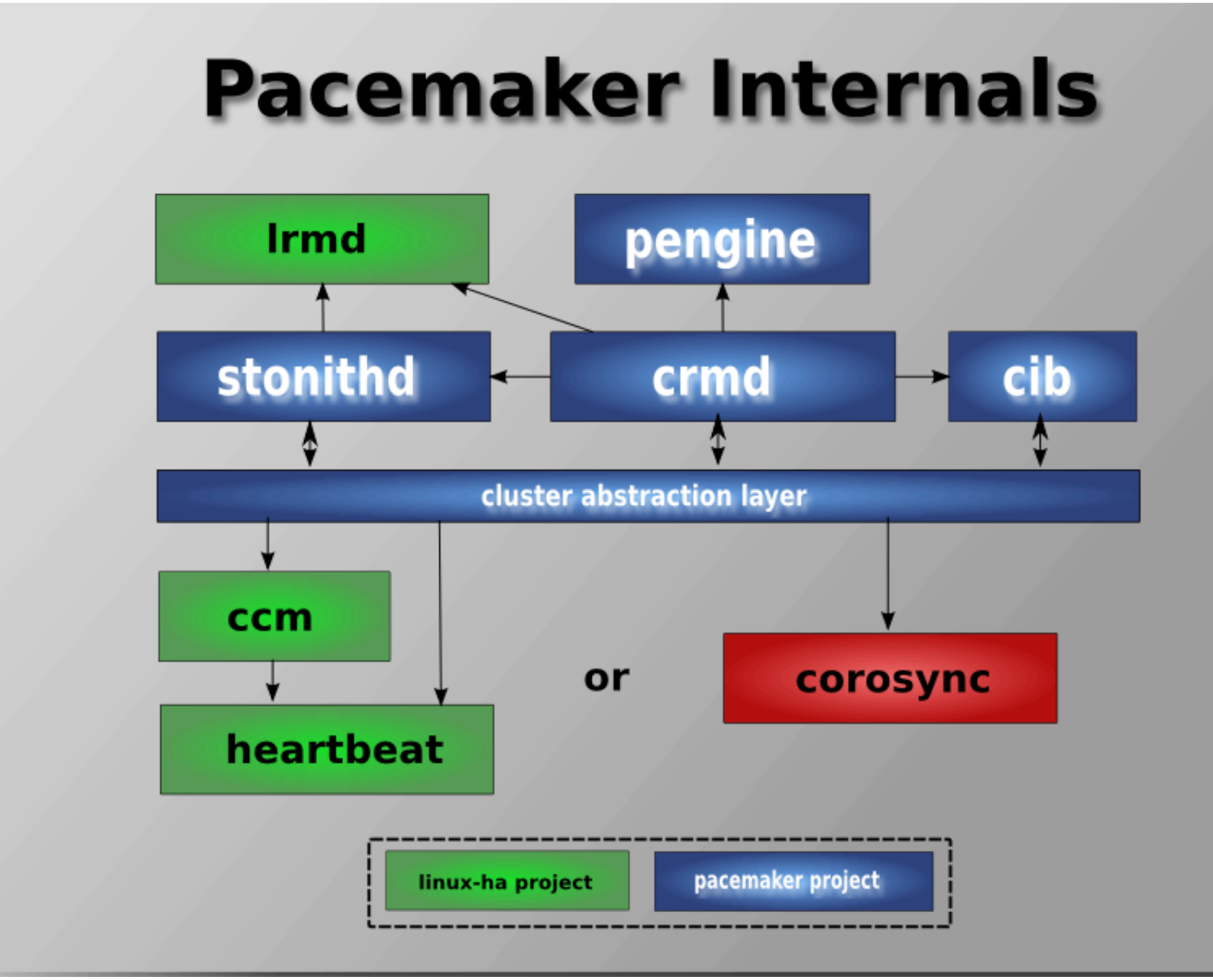


Figure 1.3. Internal Components

The CIB uses XML to represent both the cluster's configuration and current state of all resources in the cluster. The contents of the CIB are automatically kept in sync across the entire cluster and are used by the PEngine to compute the ideal state of the cluster and how it should be achieved.

This list of instructions is then fed to the DC (Designated Co-ordinator). Pacemaker centralizes all cluster decision making by electing one of the CRMd instances to act as a master. Should the elected CRMd process, or the node it is on, fail… a new one is quickly established.

The DC carries out the PEngine's instructions in the required order by passing them to either the LRMd (Local Resource Management daemon) or CRMd peers on other nodes via the cluster messaging infrastructure (which in turn passes them on to their LRMd process).

The peer nodes all report the results of their operations back to the DC and based on the expected and actual results, will either execute any actions that needed to wait for the previous one to complete, or abort processing and ask the PEngine to recalculate the ideal cluster state based on the unexpected results.

In some cases, it may be necessary to power off nodes in order to protect shared data or complete resource recovery. For this Pacemaker comes with STONITHd. STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and is usually implemented with a remote power switch. In Pacemaker, STONITH devices are modeled as resources (and configured in the CIB) to enable them to be easily monitored for failure, however STONITHd takes care of understanding the STONITH topology such that its clients simply request a node be fenced and it does the rest.

1.4. Types of Pacemaker Clusters

Pacemaker makes no assumptions about your environment, this allows it to support practically any *redundancy configuration*[3] including Active/Active, Active/Passive, N+1, N+M, N-to-1 and N-to-N.

In this document we will focus on the setup of a highly available Apache web server with an Active/Passive cluster using DRBD and Ext4 to store data. Then, we will upgrade this cluster to Active/Active using GFS2.

[3] http://en.wikipedia.org/wiki/High-availability_cluster#Node_configurations

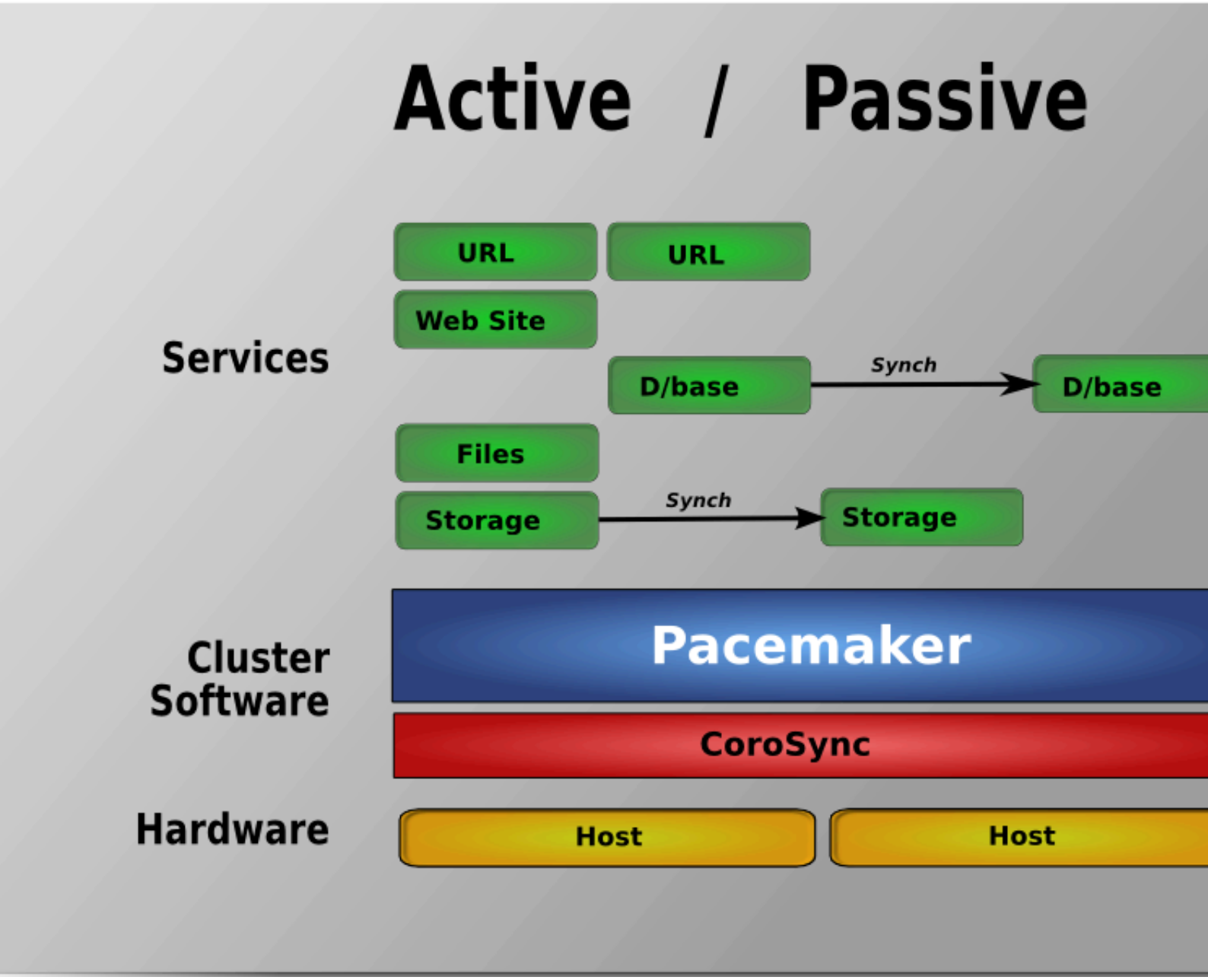


Figure 1.4. Active/Passive Redundancy

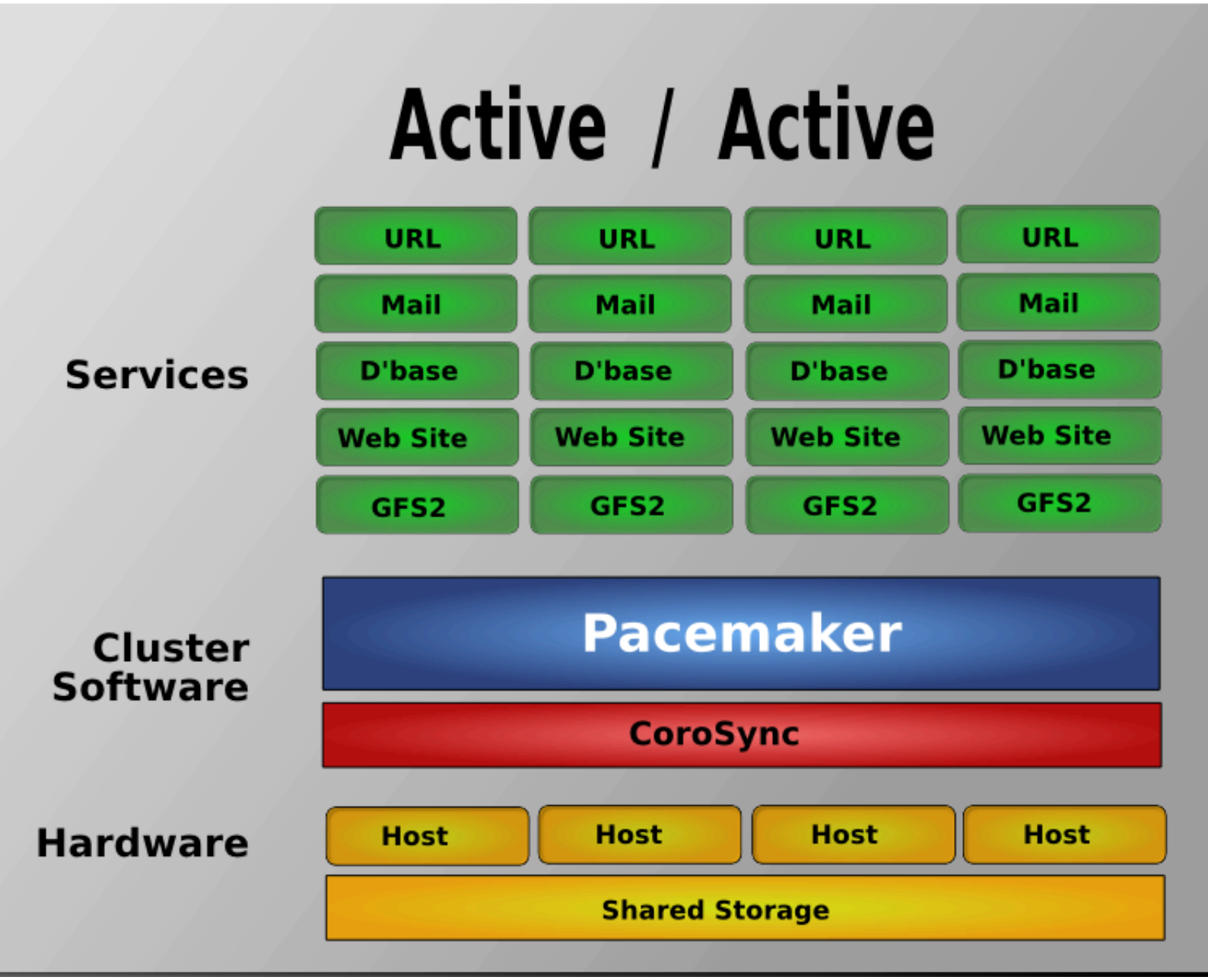


Figure 1.5. N to N Redundancy

Installation

Table of Contents

2.1. OS Installation

Detailed instructions for installing Fedora are available at *http://docs.fedoraproject.org/install-guide/f13/* in a number of languages. The abbreviated version is as follows…

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 [1] 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 [2]

[1] *http://docs.fedoraproject.org/readme-burning-isos/en-US.html*
[2] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/s1-langselection-x86.html*

Figure 2.1. Installation: Good choice

Figure 2.2. Fedora Installation - Storage Devices

Assign your machine a host name. [3] I happen to control the clusterlabs.org domain name, so I will use that here.

[3] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-networkconfig-fedora.html*

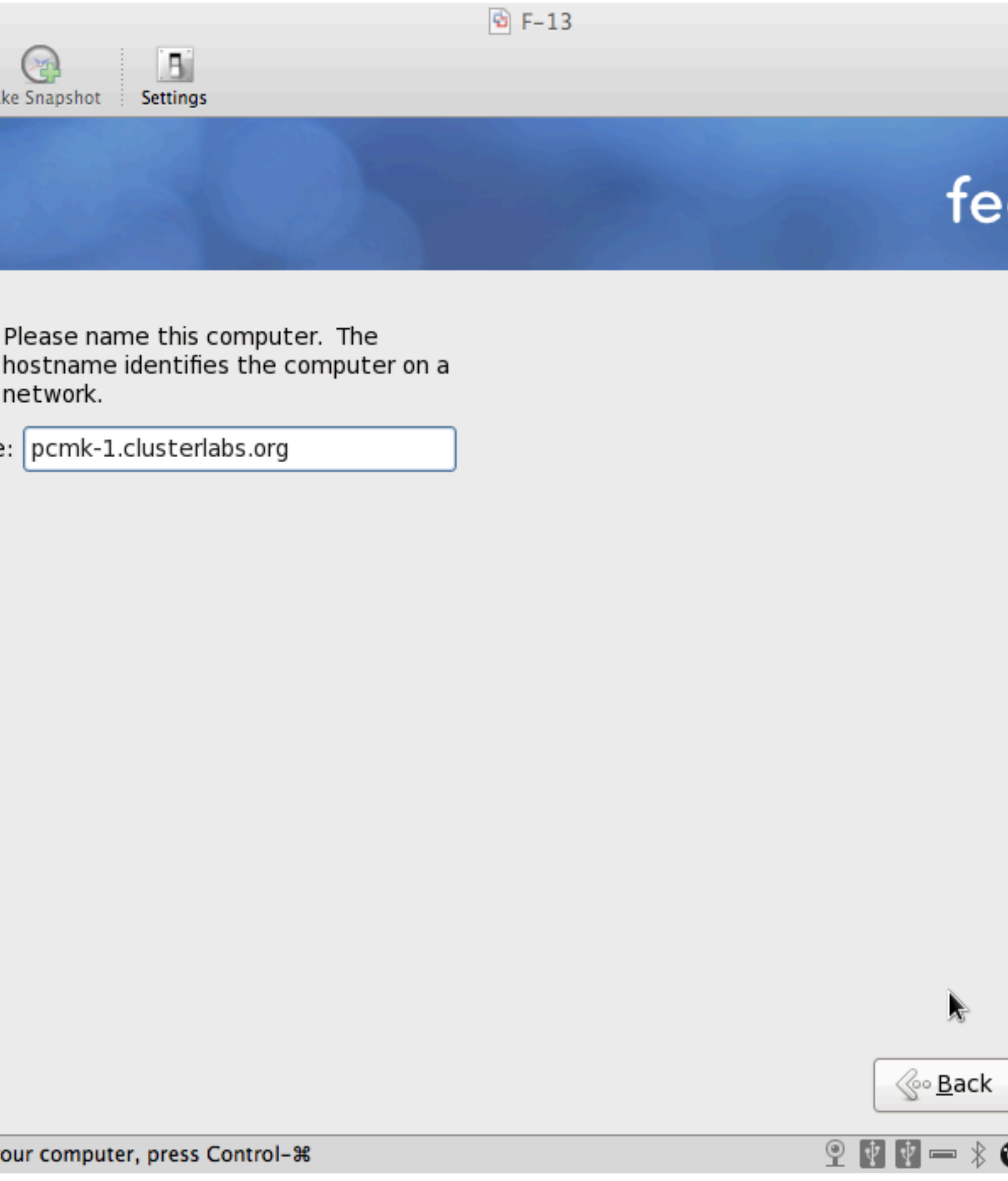


Figure 2.3. Fedora Installation - Hostname

You will then be prompted to indicate the machine's physical location and to supply a root password. [4]

Now select where you want Fedora installed. [5]

As I don't care about any existing data, I will accept the default and allow Fedora to use the complete drive. However I want to reserve some space for DRBD, so I'll check the Review and modify partitioning layout box.

[4] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/sn-account_configuration.html*
[5] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/s1-diskpartsetup-x86.html*

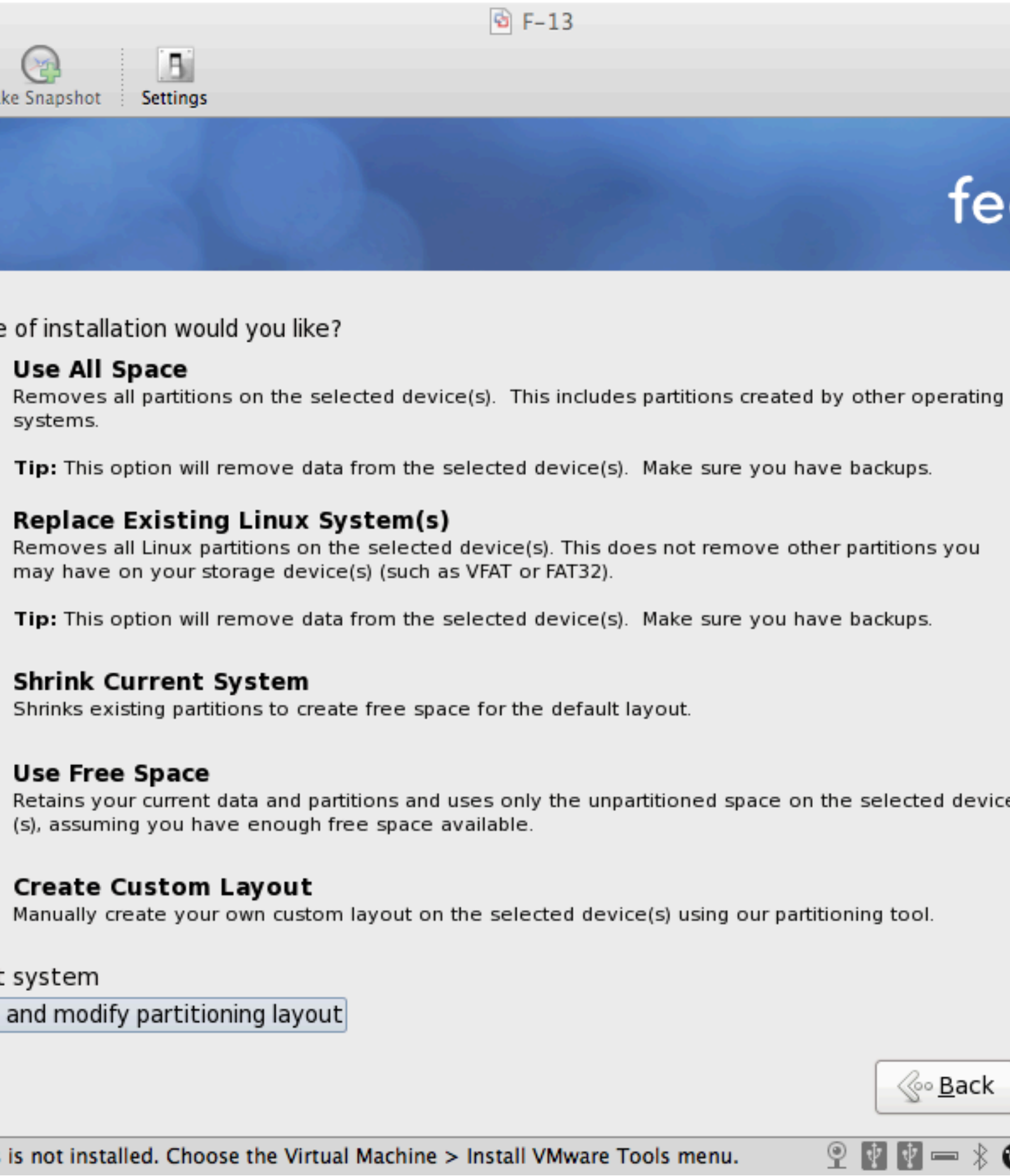


Figure 2.4. Fedora Installation - Installation Type

By default, Fedora will give all the space to the / (aka. root) partition. Wel'll take some back so we can use DRBD.

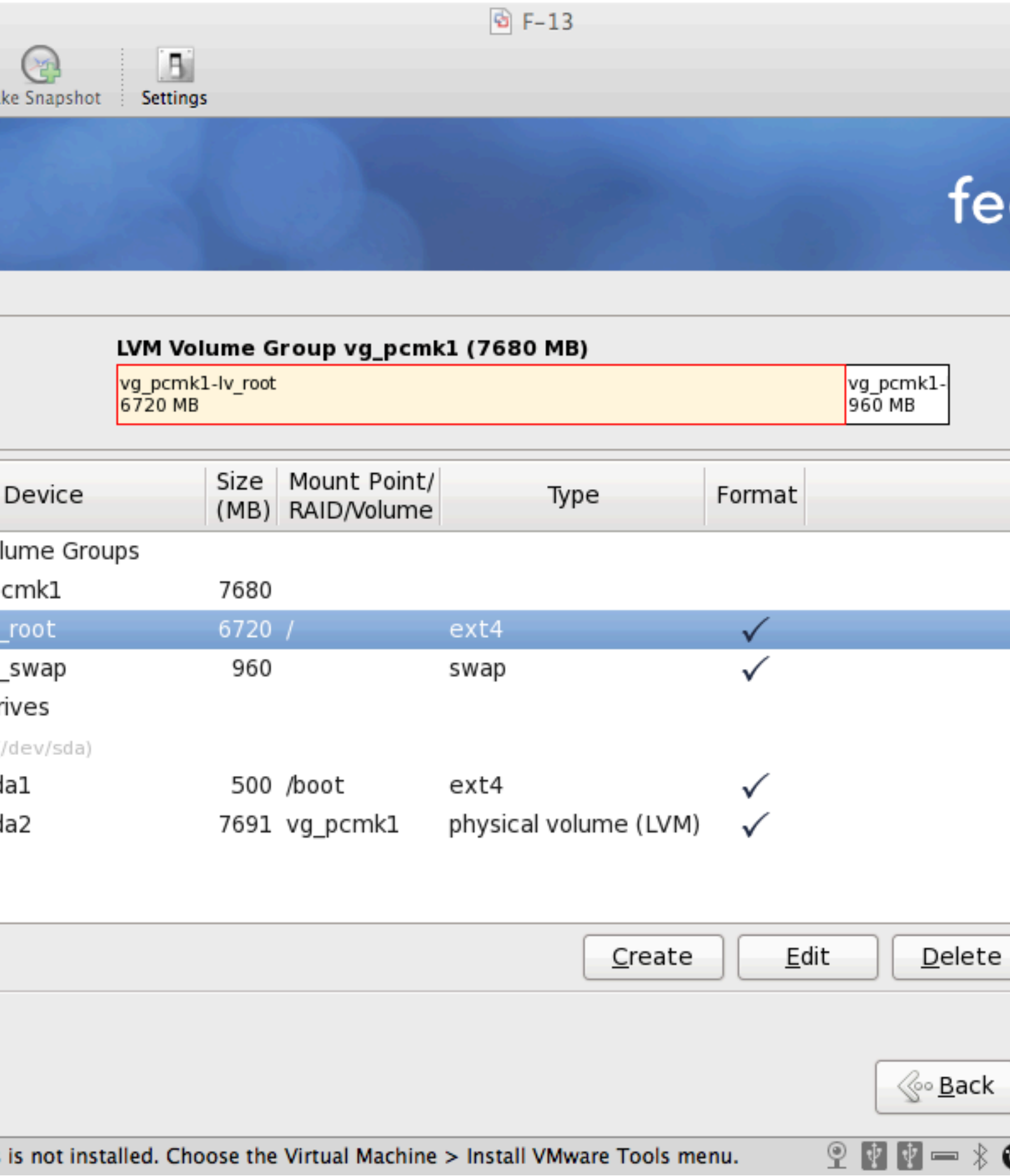


Figure 2.5. Fedora Installation - Default Partitioning

The finalized partition layout should look something like the diagram below.

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. Fedora Installation - Customize PartitioningFedora Installation: Create a partition to use (later) for website data

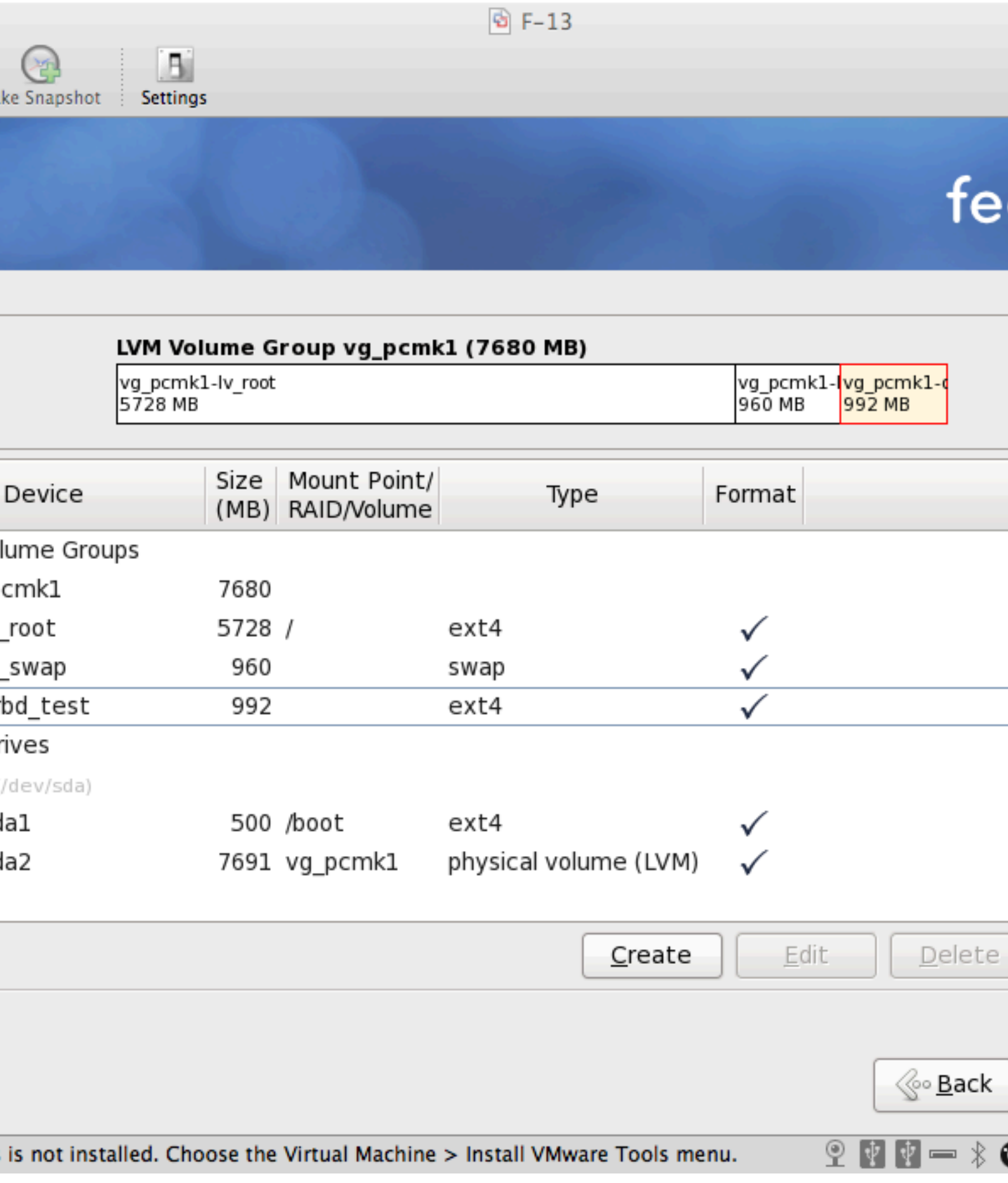


Figure 2.6. Fedora Installation - Customize Partitioning

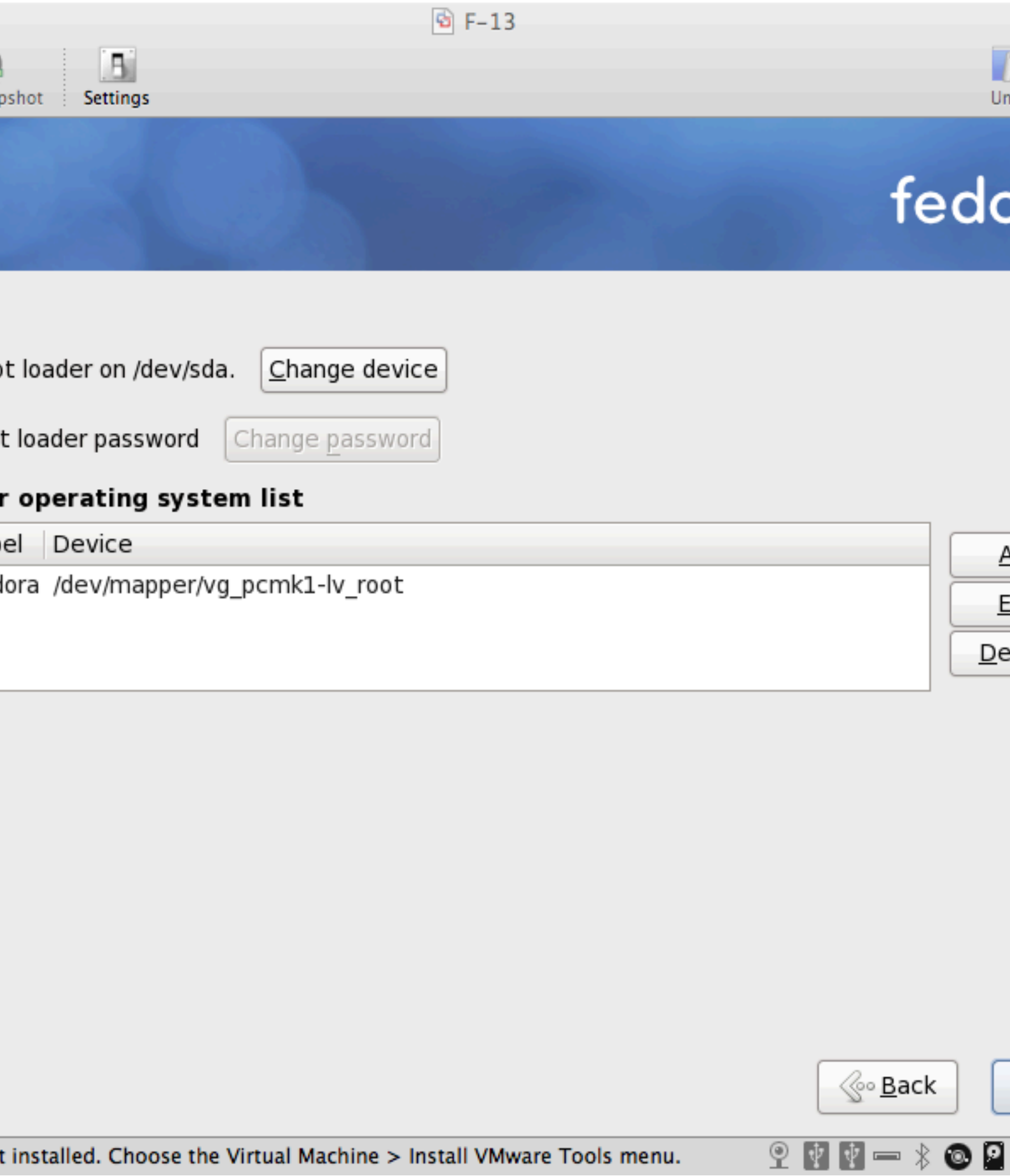


Figure 2.7. Fedora Installation - Bootloader

Next choose which software should be installed. Change the selection to Web Server since we plan on using Apache. Don’t enable updates yet, we’ll do that (and install any extra software we need) later. After you click next, Fedora will begin installing.

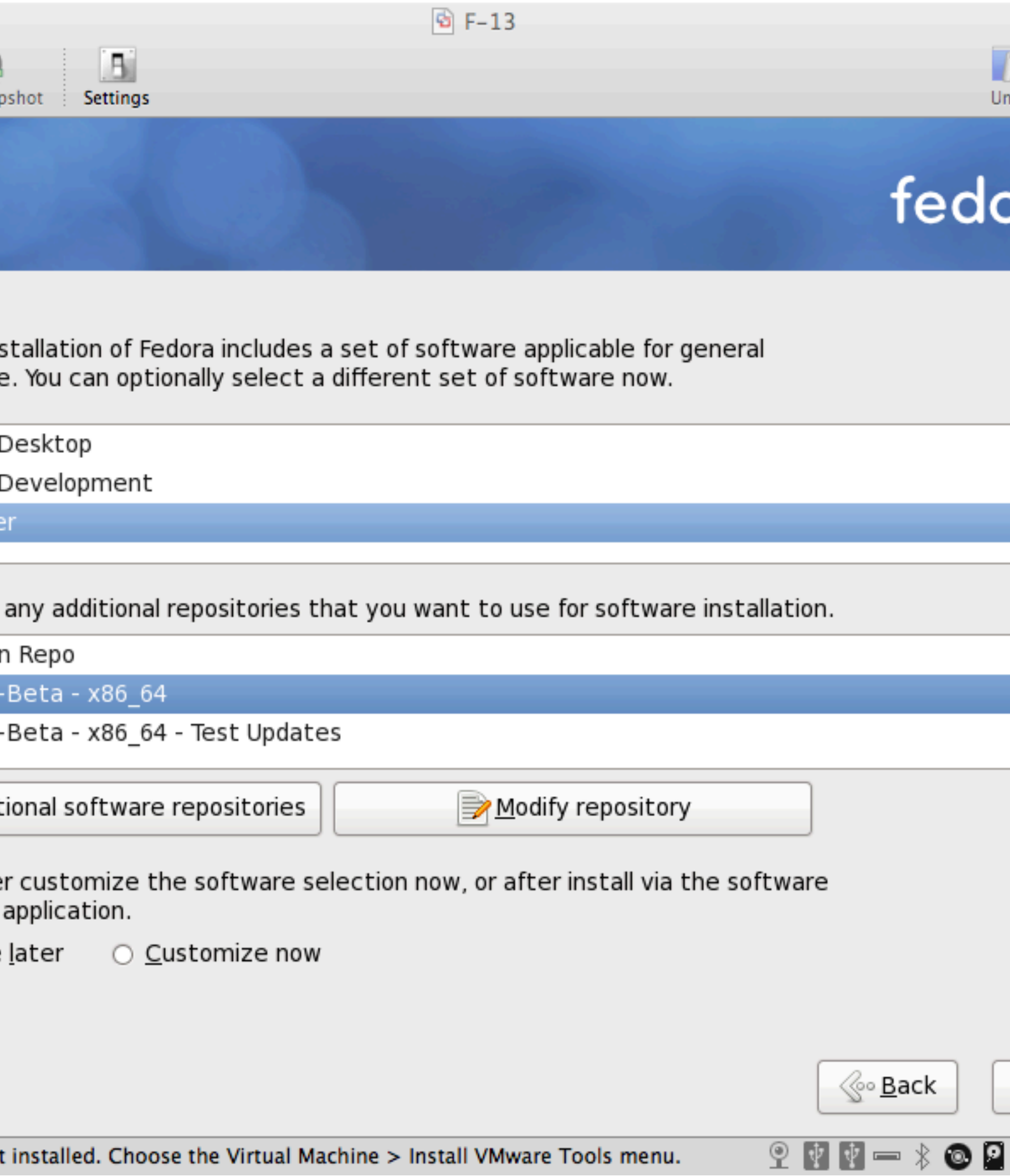


Figure 2.8. Fedora Installation - Software

Go grab something to drink, this may take a while

Figure 2.9. Fedora Installation - Installing

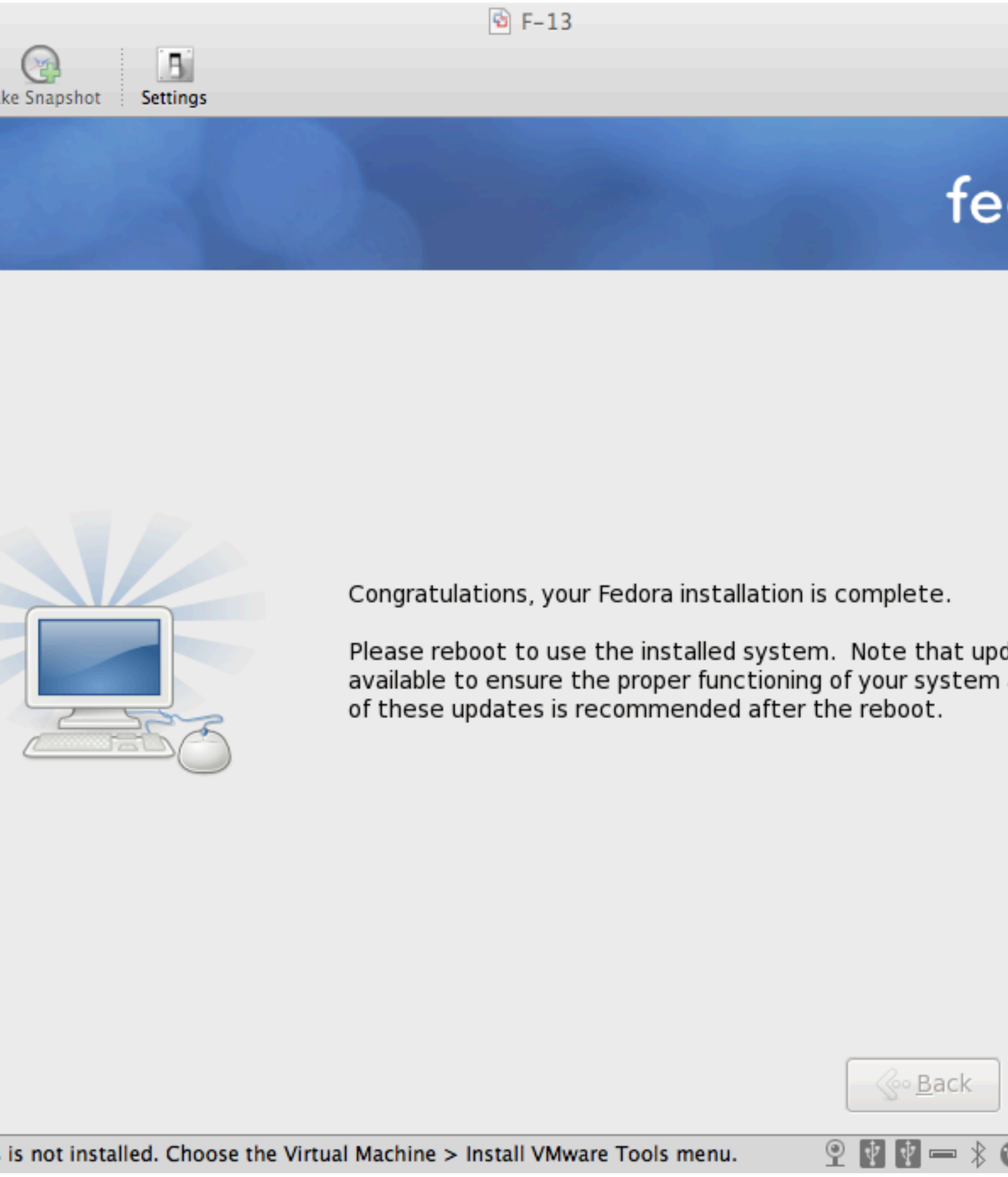


Figure 2.10. Fedora Installation - Installation Complete

Once the node reboots, follow the on screen instructions [6] to create a system user and configure the time.

[6] *http://docs.fedoraproject.org/install-guide/f13/en-US/html/ch-firstboot.html*

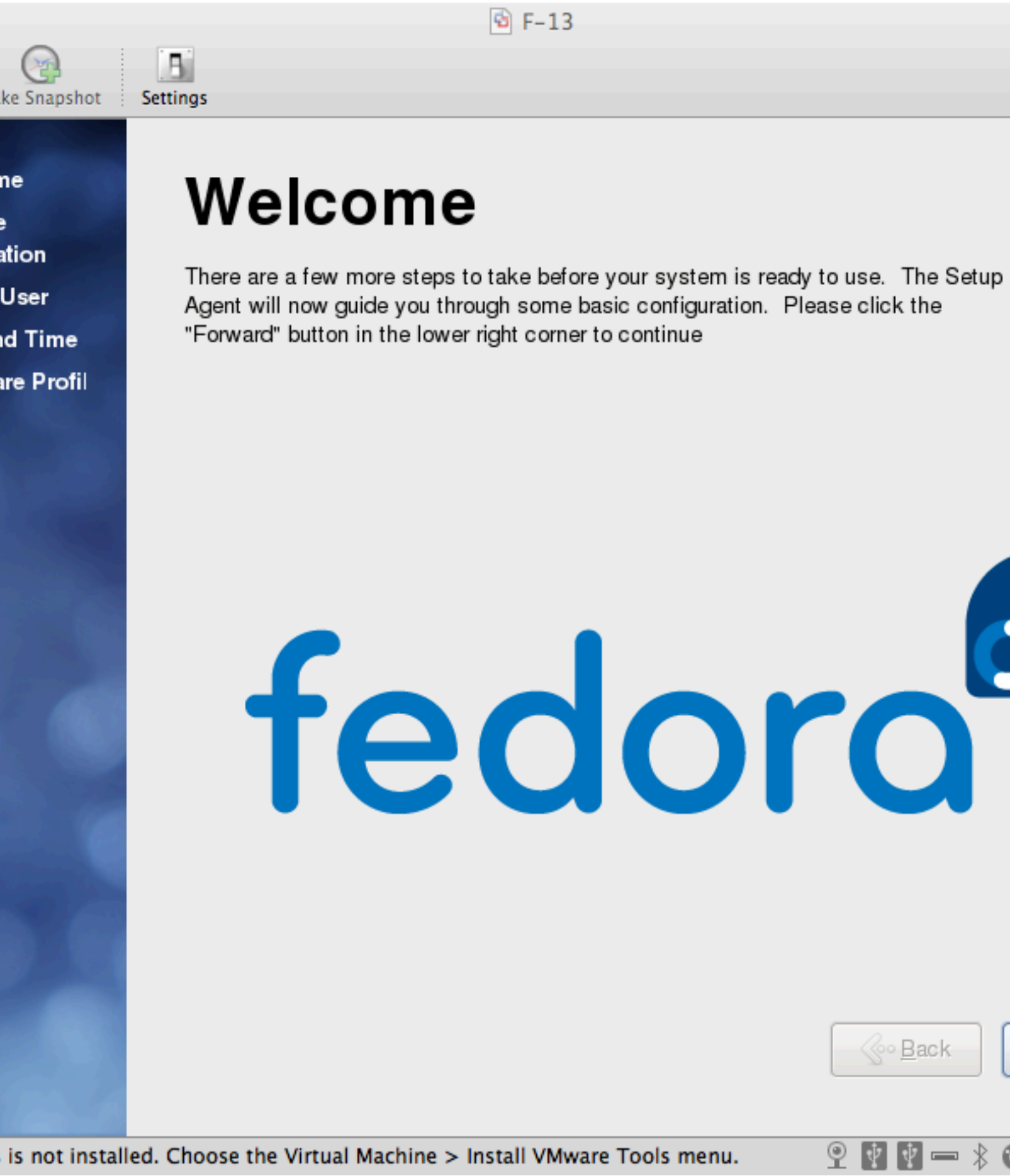


Figure 2.11. Fedora Installation - First Boot

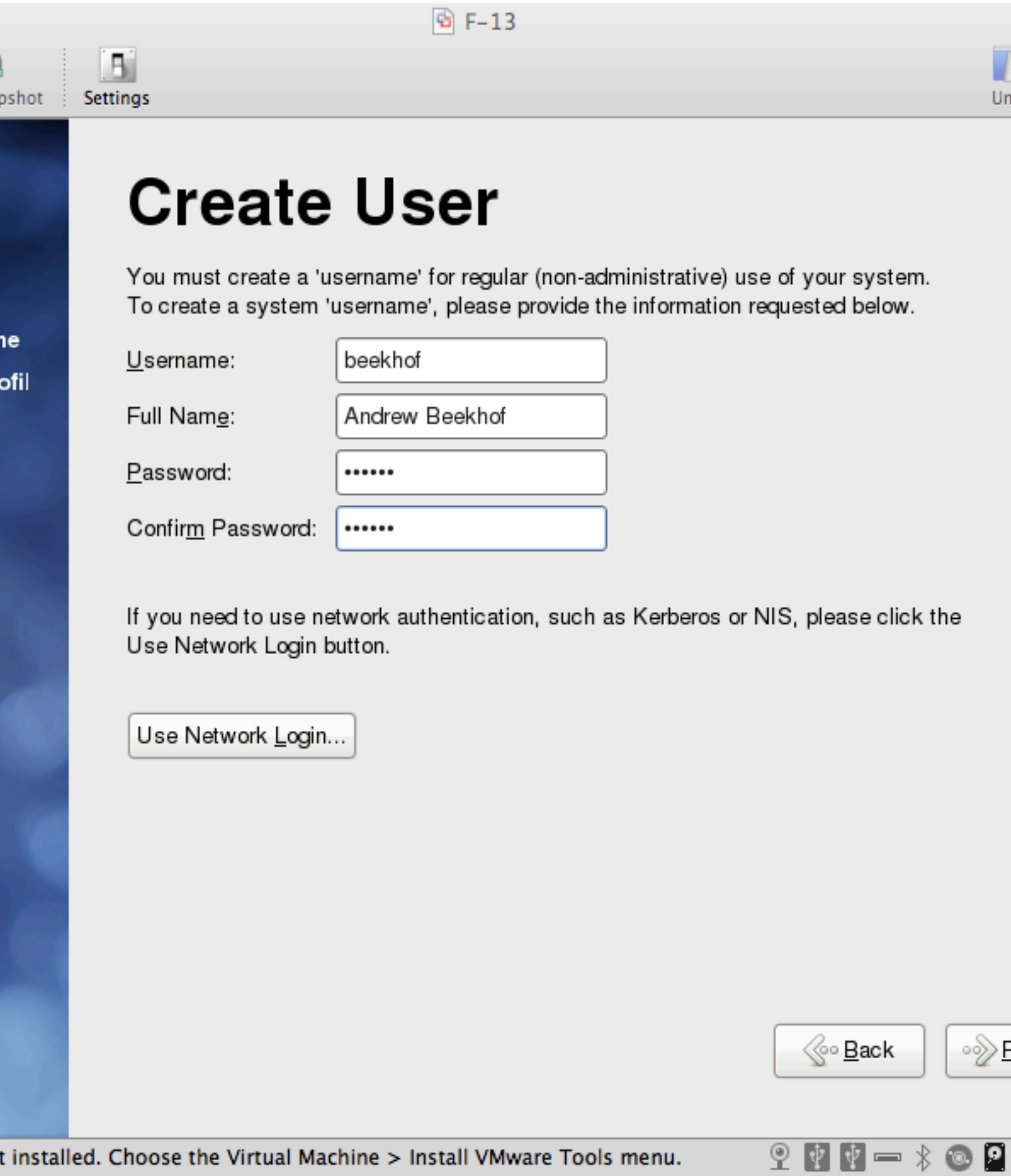


Figure 2.12. Fedora Installation - Create Non-privileged User

Note

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. Fedora Installation - Date and TimeFedora Installation: Enable NTP to keep the times on all your nodes consistent

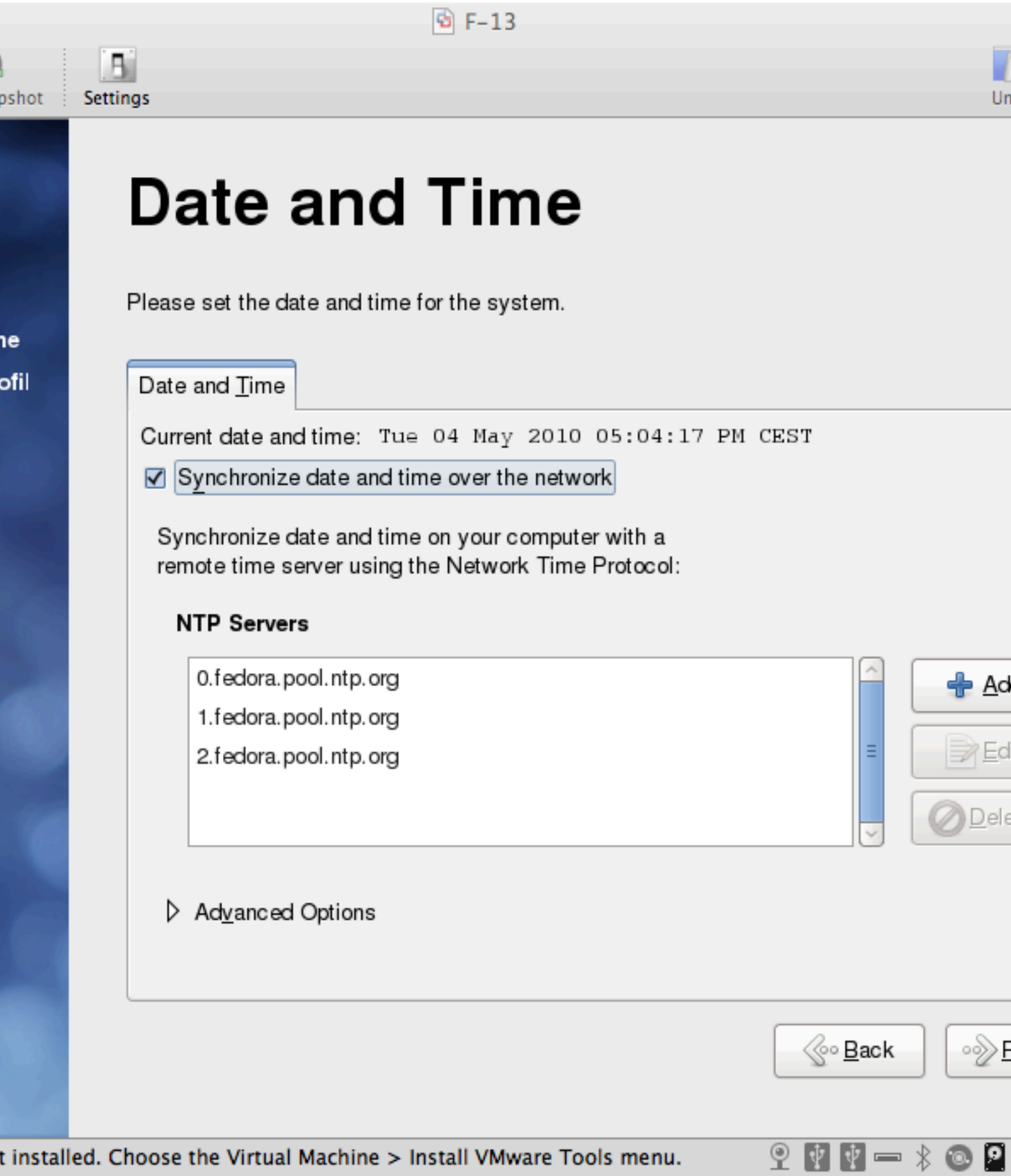


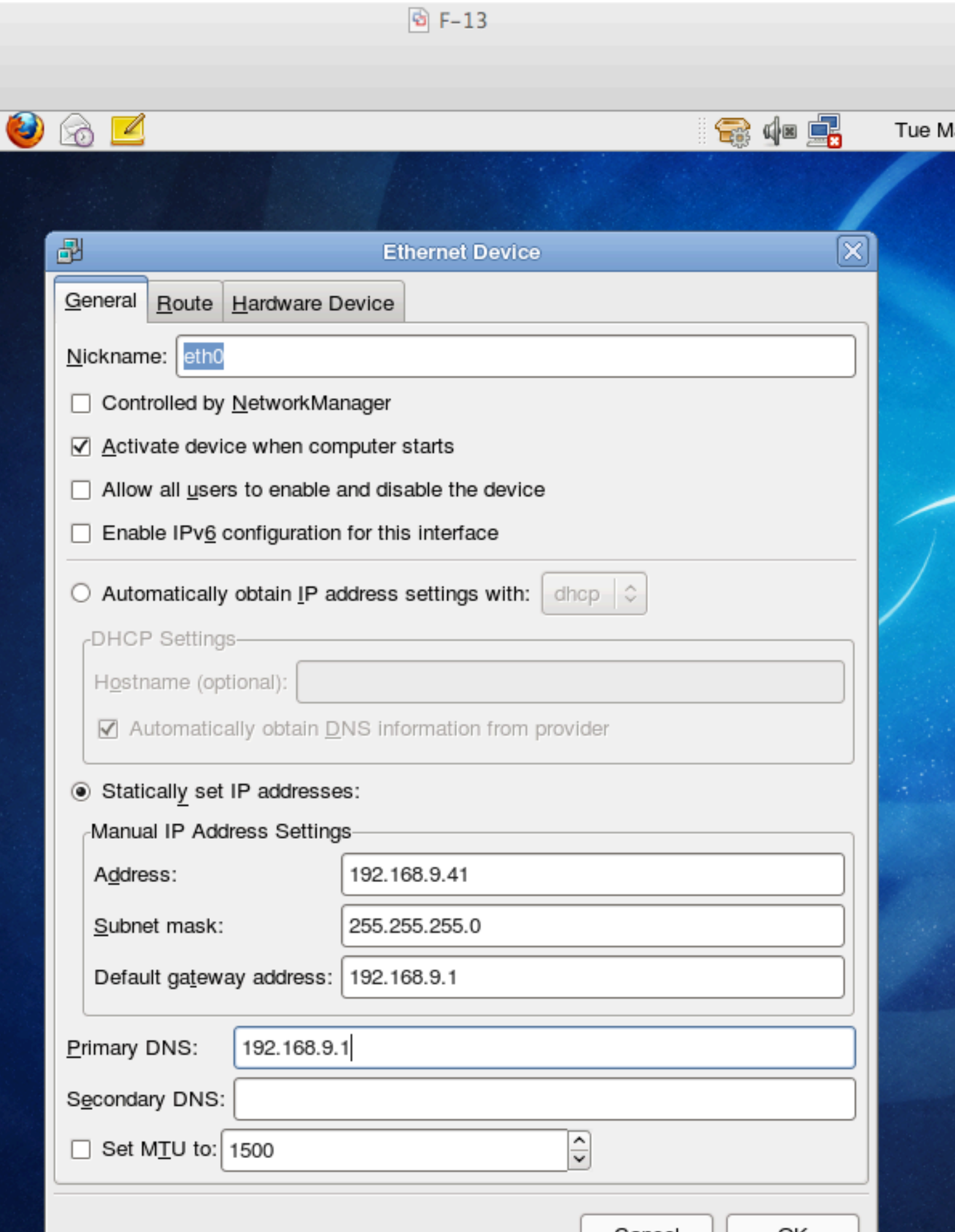
Figure 2.13. Fedora Installation - Date and Time

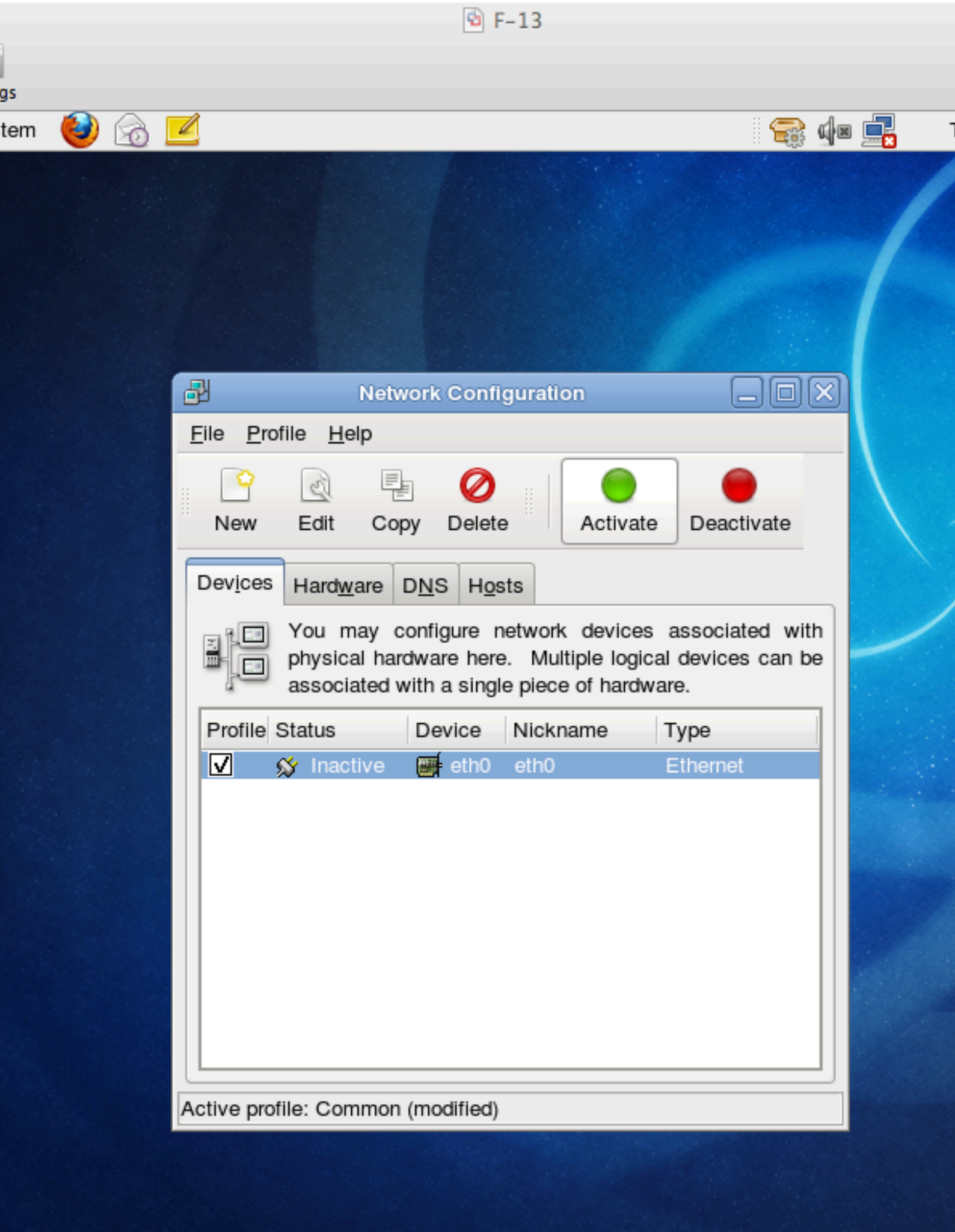
Click through the next screens until you reach the login window. Click on the user you created and supply the password you indicated earlier.

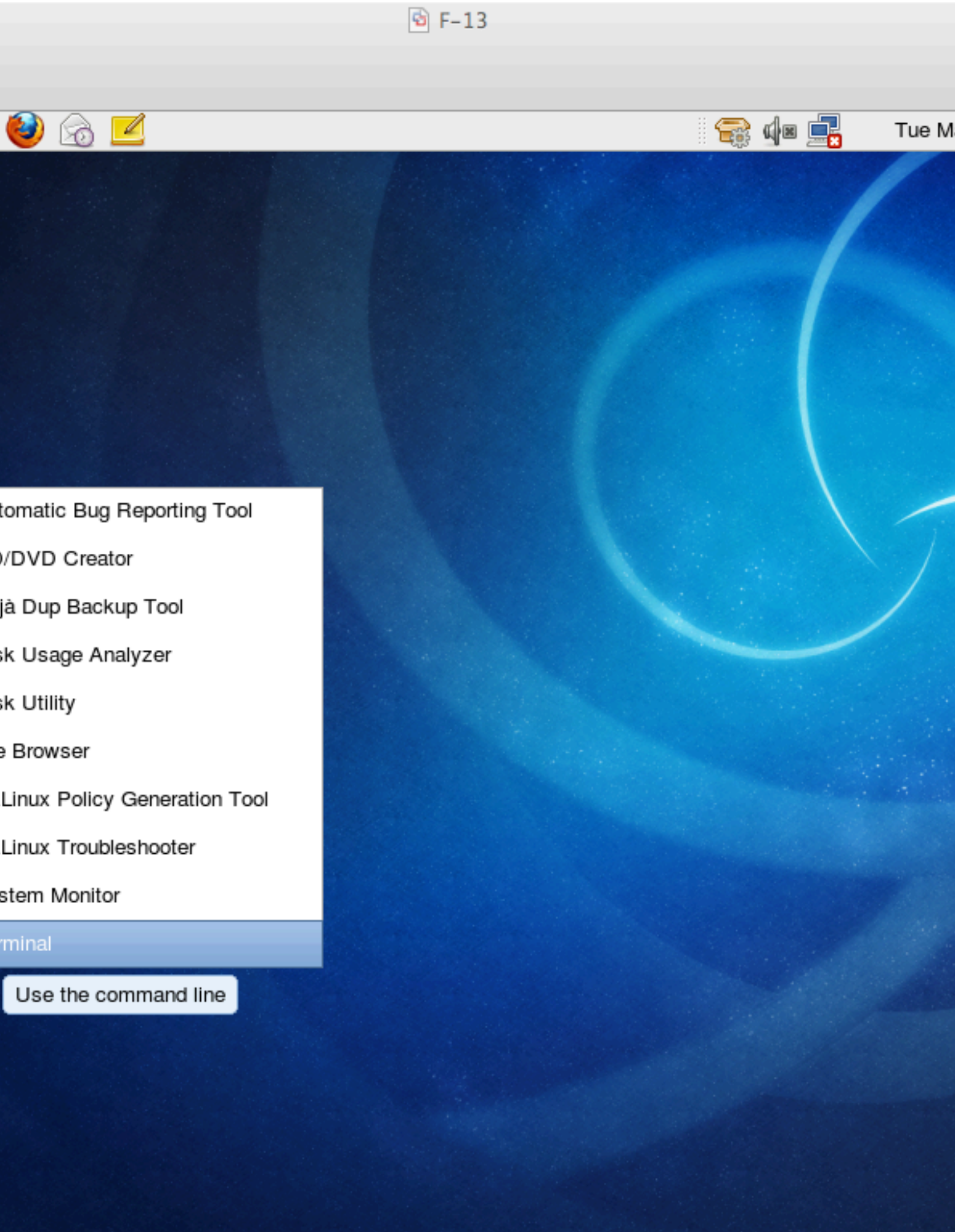

F-13
Tue M
ences
stration
entation
this Computer
creen
t beekhof...
own...
Add/Remove Software
Authentication
Bootloader
Date & Time
Firewall
Language
Network
Configure network devices and connections
Printing
SELinux Management
Services
Software Update
Users and Groups

Important

Do not accept the default network settings. Cluster machines should never obtain an ip address via DHCP. Here I will use the internal addresses for the clusterlab.org network.


F-13
Tue M
Ethernet Device
General
Route
Hardware Device
Nickname:
eth0
Controlled by NetworkManager
Activate device when computer starts
Allow all users to enable and disable the device
Enable IPv6 configuration for this interface
Automatically obtain IP address settings with:
dhcp
DHCP Settings
Hostname (optional):
Automatically obtain DNS information from provider
Statically set IP addresses:
Manual IP Address Settings
Address:
192.168.9.41
Subnet mask:
255.255.255.0
Default gateway address:
192.168.9.1
Primary DNS:
192.168.9.1
Secondary DNS:
Set MTU to:
1500


F-13
Network Configuration
File Profile Help
New Edit Copy Delete Activate Deactivate
Devices Hardware DNS Hosts
You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.
Profile Status Device Nickname Type
Inactive eth0 eth0 Ethernet
Active profile: Common (modified)


F-13
Tue M
tomatic Bug Reporting Tool
/DVD Creator
jà Dup Backup Tool
sk Usage Analyzer
sk Utility
e Browser
Linux Policy Generation Tool
Linux Troubleshooter
stem Monitor
rminal
Use the command line

Note

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

2.2. Cluster Software Installation

Go to the terminal window you just opened and switch to the super user (aka. "root") account with the su command. You will need to supply the password you entered earlier during the installation process.

```
[beekhof@pcmk-1 ~]$ su -
Password:
[root@pcmk-1 ~]#
```

Note

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

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:6f:e1:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.9.41/24 brd 192.168.9.255 scope global eth0
    inet6 ::20c:29ff:fe6f:e158/64 scope global dynamic
       valid_lft 2591667sec preferred_lft 604467sec
    inet6 2002:57ae:43fc:0:20c:29ff:fe6f:e158/64 scope global dynamic
       valid_lft 2591990sec preferred_lft 604790sec
    inet6 fe80::20c:29ff:fe6f:e158/64 scope link
       valid_lft forever preferred_lft forever
# ping -c 1 www.google.com
PING www.l.google.com (74.125.39.99) 56(84) bytes of data.
64 bytes from fx-in-f99.1e100.net (74.125.39.99): icmp_seq=1 ttl=56 time=16.7 ms

--- www.l.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 20ms
rtt min/avg/max/mdev = 16.713/16.713/16.713/0.000 ms
# /sbin/chkconfig network on
#
```

2.2.1. 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.

Important

```
TODO: Create an Appendix that deals with (at least) re-enabling the firewall.
```

```
# sed -i.bak "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/selinux/config
# /sbin/chkconfig --del iptables
# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
```

Note

You will need to reboot for the SELinux changes to take effect. Otherwise you will see something like this when you start corosync:

```
May  4 19:30:54 pcmk-1 setroubleshoot: SELinux is preventing /usr/sbin/
corosync "getattr" access on /. For complete SELinux messages. run sealert -l
 6e0d4384-638e-4d55-9aaf-7dac011f29c1
May  4 19:30:54 pcmk-1 setroubleshoot: SELinux is preventing /usr/sbin/
corosync "getattr" access on /. For complete SELinux messages. run sealert -l
 6e0d4384-638e-4d55-9aaf-7dac011f29c1
```

2.2.2. Install the Cluster Software

Since version 12, Fedora comes with recent versions of everything you need, so simply fire up the shell and run:

```
# sed -i.bak "s/enabled=0/enabled=1/g"
/etc/yum.repos.d/fedora.repo
# sed -i.bak "s/enabled=0/enabled=1/g"
/etc/yum.repos.d/fedora-updates.repo
# yum install -y pacemaker corosync
Loaded plugins: presto, refresh-packagekit
fedora/metalink                                          |  22 kB     00:00
fedora-debuginfo/metalink                                |  16 kB     00:00
fedora-debuginfo                                         | 3.2 kB     00:00
fedora-debuginfo/primary_db                              | 1.4 MB     00:04
fedora-source/metalink                                   |  22 kB     00:00
fedora-source                                            | 3.2 kB     00:00
fedora-source/primary_db                                 | 3.0 MB     00:05
updates/metalink                                         |  26 kB     00:00
updates                                                  | 2.6 kB     00:00
updates/primary_db                                       | 1.1 kB     00:00
updates-debuginfo/metalink                               |  18 kB     00:00
updates-debuginfo                                        | 2.6 kB     00:00
updates-debuginfo/primary_db                             | 1.1 kB     00:00
updates-source/metalink                                  |  25 kB     00:00
updates-source                                           | 2.6 kB     00:00
updates-source/primary_db                                | 1.1 kB     00:00
Setting up Install Process
```

```
Resolving Dependencies
--> Running transaction check
---> Package corosync.x86_64 0:1.2.1-1.fc13 set to be updated
--> Processing Dependency: corosynclib = 1.2.1-1.fc13 for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libquorum.so.4(COROSYNC_QUORUM_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libvotequorum.so.4(COROSYNC_VOTEQUORUM_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcpg.so.4(COROSYNC_CPG_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libconfdb.so.4(COROSYNC_CONFDB_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcfg.so.4(COROSYNC_CFG_0.82)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libpload.so.4(COROSYNC_PLOAD_1.0)(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: liblogsys.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libconfdb.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcoroipcc.so.4()(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcpg.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libquorum.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcoroipcs.so.4()(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libvotequorum.so.4()(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libcfg.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libtotem_pg.so.4()(64bit) for package:
 corosync-1.2.1-1.fc13.x86_64
--> Processing Dependency: libpload.so.4()(64bit) for package: corosync-1.2.1-1.fc13.x86_64
---> Package pacemaker.x86_64 0:1.1.5-1.fc13 set to be updated
--> Processing Dependency: heartbeat >= 3.0.0 for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: net-snmp >= 5.4 for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: resource-agents for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: cluster-glue for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libnetsnmp.so.20()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libcrmcluster.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libpengine.so.3()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libnetsnmpagent.so.20()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libesmtp.so.5()(64bit) for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libstonithd.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libhbclient.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libpils.so.2()(64bit) for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libpe_status.so.2()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libnetsnmpmibs.so.20()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libnetsnmphelpers.so.20()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libcib.so.1()(64bit) for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libccmclient.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libstonith.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: liblrm.so.2()(64bit) for package: pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libtransitioner.so.1()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libpe_rules.so.2()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
```

```
--> Processing Dependency: libcrmcommon.so.2()(64bit) for package:
 pacemaker-1.1.5-1.fc13.x86_64
--> Processing Dependency: libplumb.so.2()(64bit) for package: pacemaker-1.1.5-1.fc13.x86_64
--> Running transaction check
---> Package cluster-glue.x86_64 0:1.0.2-1.fc13 set to be updated
--> Processing Dependency: perl-TimeDate for package: cluster-glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMIutils.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMIposix.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libopenhpi.so.2()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
--> Processing Dependency: libOpenIPMI.so.0()(64bit) for package: cluster-
glue-1.0.2-1.fc13.x86_64
---> Package cluster-glue-libs.x86_64 0:1.0.2-1.fc13 set to be updated
---> Package corosynclib.x86_64 0:1.2.1-1.fc13 set to be updated
--> Processing Dependency: librdmacm.so.1(RDMACM_1.0)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1(IBVERBS_1.0)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1(IBVERBS_1.1)(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: libibverbs.so.1()(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
--> Processing Dependency: librdmacm.so.1()(64bit) for package:
 corosynclib-1.2.1-1.fc13.x86_64
---> Package heartbeat.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13 set to be updated
--> Processing Dependency: PyXML for package: heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64
---> Package heartbeat-libs.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13 set to be updated
---> Package libesmtp.x86_64 0:1.0.4-12.fc12 set to be updated
---> Package net-snmp.x86_64 1:5.5-12.fc13 set to be updated
--> Processing Dependency: libsensors.so.4()(64bit) for package: 1:net-
snmp-5.5-12.fc13.x86_64
---> Package net-snmp-libs.x86_64 1:5.5-12.fc13 set to be updated
---> Package pacemaker-libs.x86_64 0:1.1.5-1.fc13 set to be updated
---> Package resource-agents.x86_64 0:3.0.10-1.fc13 set to be updated
--> Processing Dependency: libnet.so.1()(64bit) for package: resource-
agents-3.0.10-1.fc13.x86_64
--> Running transaction check
---> Package OpenIPMI-libs.x86_64 0:2.0.16-8.fc13 set to be updated
---> Package PyXML.x86_64 0:0.8.4-17.fc13 set to be updated
---> Package libibverbs.x86_64 0:1.1.3-4.fc13 set to be updated
--> Processing Dependency: libibverbs-driver for package: libibverbs-1.1.3-4.fc13.x86_64
---> Package libnet.x86_64 0:1.1.4-3.fc12 set to be updated
---> Package librdmacm.x86_64 0:1.0.10-2.fc13 set to be updated
---> Package lm_sensors-libs.x86_64 0:3.1.2-2.fc13 set to be updated
---> Package openhpi-libs.x86_64 0:2.14.1-3.fc13 set to be updated
---> Package perl-TimeDate.noarch 1:1.20-1.fc13 set to be updated
--> Running transaction check
---> Package libmlx4.x86_64 0:1.0.1-5.fc13 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved


================================================================================
 Package                Arch     Version                          Repository      Size
================================================================================
Installing:
 corosync               x86_64   1.2.1-1.fc13                     fedora         136 k
 pacemaker              x86_64   1.1.5-1.fc13                     fedora         543 k
Installing for dependencies:
 OpenIPMI-libs          x86_64   2.0.16-8.fc13                    fedora         474 k
 PyXML                  x86_64   0.8.4-17.fc13                    fedora         906 k
 cluster-glue           x86_64   1.0.2-1.fc13                     fedora         230 k
 cluster-glue-libs      x86_64   1.0.2-1.fc13                     fedora         116 k
 corosynclib            x86_64   1.2.1-1.fc13                     fedora         145 k
```

```
 heartbeat              x86_64   3.0.0-0.7.0daab7da36a8.hg.fc13      updates        172 k
 heartbeat-libs         x86_64   3.0.0-0.7.0daab7da36a8.hg.fc13      updates        265 k
 libesmtp               x86_64   1.0.4-12.fc12                       fedora          54 k
 libibverbs             x86_64   1.1.3-4.fc13                        fedora          42 k
 libmlx4                x86_64   1.0.1-5.fc13                        fedora          27 k
 libnet                 x86_64   1.1.4-3.fc12                        fedora          49 k
 librdmacm              x86_64   1.0.10-2.fc13                       fedora          22 k
 lm_sensors-libs        x86_64   3.1.2-2.fc13                        fedora          37 k
 net-snmp               x86_64   1:5.5-12.fc13                       fedora         295 k
 net-snmp-libs          x86_64   1:5.5-12.fc13                       fedora         1.5 M
 openhpi-libs           x86_64   2.14.1-3.fc13                       fedora         135 k
 pacemaker-libs         x86_64   1.1.5-1.fc13                        fedora         264 k
 perl-TimeDate          noarch   1:1.20-1.fc13                       fedora          42 k
 resource-agents        x86_64   3.0.10-1.fc13                       fedora         357 k

Transaction Summary
================================================================================
Install      21 Package(s)
Upgrade       0 Package(s)

Total download size: 5.7 M
Installed size: 20 M
Downloading Packages:
Setting up and reading Presto delta metadata
updates-testing/prestodelta                              | 164 kB     00:00
fedora/prestodelta                                       |  150 B     00:00
Processing delta metadata
Package(s) data still to download: 5.7 M
(1/21): OpenIPMI-libs-2.0.16-8.fc13.x86_64.rpm           | 474 kB     00:00
(2/21): PyXML-0.8.4-17.fc13.x86_64.rpm                   | 906 kB     00:01
(3/21): cluster-glue-1.0.2-1.fc13.x86_64.rpm             | 230 kB     00:00
(4/21): cluster-glue-libs-1.0.2-1.fc13.x86_64.rpm        | 116 kB     00:00
(5/21): corosync-1.2.1-1.fc13.x86_64.rpm                 | 136 kB     00:00
(6/21): corosynclib-1.2.1-1.fc13.x86_64.rpm              | 145 kB     00:00
(7/21): heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64.rpm | 172 kB     00:00
(8/21): heartbeat-libs-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64.rpm | 265 kB     00:00
(9/21): libesmtp-1.0.4-12.fc12.x86_64.rpm                |  54 kB     00:00
(10/21): libibverbs-1.1.3-4.fc13.x86_64.rpm              |  42 kB     00:00
(11/21): libmlx4-1.0.1-5.fc13.x86_64.rpm                 |  27 kB     00:00
(12/21): libnet-1.1.4-3.fc12.x86_64.rpm                  |  49 kB     00:00
(13/21): librdmacm-1.0.10-2.fc13.x86_64.rpm              |  22 kB     00:00
(14/21): lm_sensors-libs-3.1.2-2.fc13.x86_64.rpm         |  37 kB     00:00
(15/21): net-snmp-5.5-12.fc13.x86_64.rpm                 | 295 kB     00:00
(16/21): net-snmp-libs-5.5-12.fc13.x86_64.rpm            | 1.5 MB     00:01
(17/21): openhpi-libs-2.14.1-3.fc13.x86_64.rpm           | 135 kB     00:00
(18/21): pacemaker-1.1.5-1.fc13.x86_64.rpm               | 543 kB     00:00
(19/21): pacemaker-libs-1.1.5-1.fc13.x86_64.rpm          | 264 kB     00:00
(20/21): perl-TimeDate-1.20-1.fc13.noarch.rpm            |  42 kB     00:00
(21/21): resource-agents-3.0.10-1.fc13.x86_64.rpm        | 357 kB     00:00

Total                                            539 kB/s | 5.7 MB     00:10
warning: rpmts_HdrFromFdno: Header V3 RSA/SHA256 Signature, key ID e8e40fde: NOKEY
fedora/gpgkey                                            | 3.2 kB     00:00 ...
Importing GPG key 0xE8E40FDE "Fedora (13) <fedora@fedoraproject.org%gt;" from /etc/pki/rpm-
gpg/RPM-GPG-KEY-fedora-x86_64

Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : lm_sensors-libs-3.1.2-2.fc13.x86_64                     1/21
  Installing     : 1:net-snmp-libs-5.5-12.fc13.x86_64                      2/21
  Installing     : 1:net-snmp-5.5-12.fc13.x86_64                           3/21
  Installing     : openhpi-libs-2.14.1-3.fc13.x86_64                       4/21
  Installing     : libibverbs-1.1.3-4.fc13.x86_64                          5/21
  Installing     : libmlx4-1.0.1-5.fc13.x86_64                             6/21
  Installing     : librdmacm-1.0.10-2.fc13.x86_64                          7/21
```

```
  Installing     : corosync-1.2.1-1.fc13.x86_64                                  8/21
  Installing     : corosynclib-1.2.1-1.fc13.x86_64                               9/21
  Installing     : libesmtp-1.0.4-12.fc12.x86_64                                10/21
  Installing     : OpenIPMI-libs-2.0.16-8.fc13.x86_64                           11/21
  Installing     : PyXML-0.8.4-17.fc13.x86_64                                   12/21
  Installing     : libnet-1.1.4-3.fc12.x86_64                                   13/21
  Installing     : 1:perl-TimeDate-1.20-1.fc13.noarch                           14/21
  Installing     : cluster-glue-1.0.2-1.fc13.x86_64                             15/21
  Installing     : cluster-glue-libs-1.0.2-1.fc13.x86_64                        16/21
  Installing     : resource-agents-3.0.10-1.fc13.x86_64                         17/21
  Installing     : heartbeat-libs-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64         18/21
  Installing     : heartbeat-3.0.0-0.7.0daab7da36a8.hg.fc13.x86_64              19/21
  Installing     : pacemaker-1.1.5-1.fc13.x86_64                                20/21
  Installing     : pacemaker-libs-1.1.5-1.fc13.x86_64                           21/21

Installed:
  corosync.x86_64 0:1.2.1-1.fc13                    pacemaker.x86_64 0:1.1.5-1.fc13

Dependency Installed:
  OpenIPMI-libs.x86_64 0:2.0.16-8.fc13
  PyXML.x86_64 0:0.8.4-17.fc13
  cluster-glue.x86_64 0:1.0.2-1.fc13
  cluster-glue-libs.x86_64 0:1.0.2-1.fc13
  corosynclib.x86_64 0:1.2.1-1.fc13
  heartbeat.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13
  heartbeat-libs.x86_64 0:3.0.0-0.7.0daab7da36a8.hg.fc13
  libesmtp.x86_64 0:1.0.4-12.fc12
  libibverbs.x86_64 0:1.1.3-4.fc13
  libmlx4.x86_64 0:1.0.1-5.fc13
  libnet.x86_64 0:1.1.4-3.fc12
  librdmacm.x86_64 0:1.0.10-2.fc13
  lm_sensors-libs.x86_64 0:3.1.2-2.fc13
  net-snmp.x86_64 1:5.5-12.fc13
  net-snmp-libs.x86_64 1:5.5-12.fc13
  openhpi-libs.x86_64 0:2.14.1-3.fc13
  pacemaker-libs.x86_64 0:1.1.5-1.fc13
  perl-TimeDate.noarch 1:1.20-1.fc13
  resource-agents.x86_64 0:3.0.10-1.fc13

Complete!
#
```

2.3. 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 192.168.122.102.

2.4. Setup

2.4.1. Finalize Networking

Confirm that you can communicate with the two new nodes:

```
# 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
```

Figure 2.18. Verify Connectivity by IP address

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 two machines. Otherwise, you'll need to add the machines to /etc/hosts . Below are the entries for my cluster nodes:

```
# grep pcmk /etc/hosts
192.168.122.101 pcmk-1.clusterlabs.org pcmk-1
192.168.122.102 pcmk-2.clusterlabs.org pcmk-2
```

Figure 2.19. Set up /etc/hosts entries

We can now verify the setup by again using ping:

```
# 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
```

Figure 2.20. Verify Connectivity by Hostname

2.4.2. 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.

Warning

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:

Creating and Activating a new SSH Key

```
# ssh-keygen -t dsa -f ~/.ssh/id_dsa -N ""
Generating public/private dsa key pair.
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
91:09:5c:82:5a:6a:50:08:4e:b2:0c:62:de:cc:74:44 root@pcmk-1.clusterlabs.org
```

```
The key's randomart image is:
+--[ DSA 1024]----+
|==.ooEo..        |
|X O + .o o       |
| * A    +        |
|  +      .       |
| .      S        |
|                 |
|                 |
|                 |
|                 |
+-----------------+

# 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

```
# scp -r .ssh pcmk-2:
The authenticity of host 'pcmk-2 (192.168.122.102)' can't be established.
RSA key fingerprint is b1:2b:55:93:f1:d9:52:2b:0f:f2:8a:4e:ae:c6:7c:9a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pcmk-2,192.168.122.102' (RSA) to the list of known
 hosts.root@pcmk-2's password:
id_dsa.pub                                100%  616     0.6KB/s   00:00
id_dsa                                    100%  672     0.7KB/s   00:00
known_hosts                               100%  400     0.4KB/s   00:00
authorized_keys                           100%  616     0.6KB/s   00:00
# ssh pcmk-2 -- uname -npcmk-2
#
```

Figure 2.22. Installing the SSH Key on Another Host

2.4.3. 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
pcmk-1.clusterlabs.org
# dnsdomainname clusterlabs.org
```

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=192.168.122.1
```

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

```
 # sed -i.bak '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=192.168.122.1
```

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 -npcmk-1
# dnsdomainname clusterlabs.org
```

Now repeat on pcmk-2.

2.4.4. Configuring Corosync

Choose a port number and multi-cast [7] address. [8] 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* For this document, I have chosen port 4000 and used 226.94.1.1 as the multi-cast address.

Important

The instructions below only apply for a machine with a single NIC. If you have a more complicated setup, you should edit the configuration manually.

```
# 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.

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

Display and verify the configuration options

```
# env | grep ais_ais_mcast=226.94.1.1
ais_port=4000
ais_addr=192.168.122.0
```

[7] *http://en.wikipedia.org/wiki/Multicast*
[8] *http://en.wikipedia.org/wiki/Multicast_address*

Once you're happy with the chosen values, update the Corosync configuration

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

Finally, tell Corosync to load the Pacemaker plugin.

```
# cat <<-END >>/etc/corosync/service.d/pcmk
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  1
}
END
```

The final configuration should look something like the sample in Appendix B, Sample Corosync Configuration.

Important

When run in version 1 mode, the plugin does not start the Pacemaker daemons. Instead it just sets up the quorum and messaging interfaces needed by the rest of the stack. Starting the dameons occurs when the Pacemaker init script is invoked. This resolves two long standing issues:

a. Forking inside a multi-threaded process like Corosync causes all sorts of pain. This has been problematic for Pacemaker as it needs a number of daemons to be spawned.

b. Corosync was never designed for staggered shutdown - something previously needed in order to prevent the cluster from leaving before Pacemaker could stop all active resources.

2.4.5. Propagate the Configuration

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

```
# for f in /etc/corosync/corosync.conf /etc/corosync/service.d/pcmk /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
#
```

Verify Cluster Installation

Table of Contents

3.1. Verify Corosync Installation

Start Corosync on the first node

```
# /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.*network interface" -e "Corosync Cluster Engine" -e "Successfully read
 main configuration file" /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1540]: [MAIN ] Corosync Cluster Engine ('1.1.0'): started and
 ready to provide service.
Aug 27 09:05:34 pcmk-1 corosync[1540]: [MAIN ] Successfully read main configuration file '/
etc/corosync/corosync.conf'.
# grep TOTEM /var/log/messages
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transport (UDP/IP).
Aug 27 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transmit/receive security:
 libtomcrypt SOBER128/SHA1HMAC (mode 0).
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] The network interface [192.168.122.101] is
 now up.
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] A processor joined or left the membership and
 a new membership was formed.
```

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

```
# 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 09:05:34 pcmk-1 corosync[1540]: [TOTEM ] Initializing transmit/receive security:
 libtomcrypt SOBER128/SHA1HMAC (mode 0).
Aug 27 09:05:35 pcmk-1 corosync[1540]: [TOTEM ] The network interface [192.168.122.101] is
 now up.
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.
```

3.2. Verify Pacemaker Installation

Now that we have confirmed that Corosync 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: CRM: InitializedAug 27
 09:05:35 pcmk-1 corosync[1540]:  [pcmk ] Logging: Initialized pcmk_startup
Aug 27 09:05:35 pcmk-1 corosync[1540]:  [pcmk ] info: pcmk_startup: Maximum core file size
 is: 18446744073709551615
Aug 27 09:05:35 pcmk-1 corosync[1540]:  [pcmk ] info: pcmk_startup: Service: 9Aug 27 09:05:35
 pcmk-1 corosync[1540]:  [pcmk ] info: pcmk_startup: Local hostname: pcmk-1
```

Now try starting Pacemaker and check the necessary processes have been started

```
# /etc/init.d/pacemaker start
Starting Pacemaker Cluster Manager: [ OK ]

# grep -e pacemakerd.*get_config_opt -e pacemakerd.*start_child -e "Starting Pacemaker" /var/
log/messages
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found 'pacemaker' for
 option: name
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found '1' for option: ver
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Defaulting to 'no' for
 option: use_logd
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Defaulting to 'no' for
 option: use_mgmtd
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found 'on' for option:
 debug
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found 'yes' for option:
 to_logfile
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found '/var/log/
corosync.log' for option: logfile
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found 'yes' for option:
 to_syslog
Feb  8 13:31:24 pcmk-1 pacemakerd: [13155]: info: get_config_opt: Found 'daemon' for option:
 syslog_facility
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: main: Starting Pacemaker 1.1.5 (Build:
 31f088949239+):  docbook-manpages publican ncurses trace-logging cman cs-quorum heartbeat
 corosync snmp libesmtp
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14022 for process
 stonith-ng
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14023 for process
 cib
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14024 for process
 lrmd
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14025 for process
 attrd
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14026 for process
 pengine
Feb  8 16:50:38 pcmk-1 pacemakerd: [13990]: info: start_child: Forked child 14027 for process
 crmd

# ps axf PID TTY   STAT  TIME COMMAND
  2 ?    S<   0:00 [kthreadd]
  3 ?    S<   0:00 \_ [migration/0]
... lots of processes ...
13990 ?  S    0:01 pacemakerd
14022 ?  Sa   0:00 \_ /usr/lib64/heartbeat/stonithd
14023 ?  Sa   0:00 \_ /usr/lib64/heartbeat/cib
14024 ?  Sa   0:00 \_ /usr/lib64/heartbeat/lrmd
14025 ?  Sa   0:00 \_ /usr/lib64/heartbeat/attrd
14026 ?  Sa   0:00 \_ /usr/lib64/heartbeat/pengine
14027 ?  Sa   0:00 \_ /usr/lib64/heartbeat/crmd
```

Next, check for any ERRORs during startup - there shouldn't be any.

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

Repeat on the other node and display the cluster's status.

```
# ssh pcmk-2 -- /etc/init.d/pacemaker start
Starting Pacemaker Cluster Manager: [ OK ]
# crm_mon
============
Last updated: Thu Aug 27 16:54:55 2009Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
0 Resources configured.
============
Online: [ pcmk-1 pcmk-2 ]
```

Pacemaker Tools

Table of Contents

4.1. 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

```
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) are available via the --feature option to pacemakerd.

pacemakerd --features

```
Pacemaker 1.1.9-3.fc20.2 (Build: 781a388)
```

```
 Supporting v3.0.7:  generated-manpages agent-manpages ncurses libqb-logging libqb-ipc
 upstart systemd nagios  corosync-native
```

pacemakerd --help

```
pacemakerd - Start/Stop Pacemaker

Usage: pacemakerd mode [options]
Options:
 -?, --help   This text
 -$, --version   Version information
 -V, --verbose   Increase debug output
 -S, --shutdown   Instruct Pacemaker to shutdown on this machine
 -F, --features   Display the full version and list of features Pacemaker was built with

Additional Options:
 -f, --foreground   (Ignored) Pacemaker always runs in the foreground
 -p, --pid-file=value  (Ignored) Daemon pid file location

Report bugs to pacemaker@oss.clusterlabs.org
```

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
 -Q, --quiet   Display only essential output

Modes:
 -h, --as-html=value Write cluster status to the named html file
 -X, --as-xml   Write cluster status as xml to stdout. This will enable one-shot mode.
 -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)

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
 -c, --tickets    Display cluster tickets
 -W, --watch-fencing    Listen for fencing events. For use with --external-agent, --mail-to
 and/or --snmp-traps where supported
 -A, --show-node-attributes  Display node attributes

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
 -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 status on the console with updates as they occur:
```

```
 # crm_mon

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

 # crm_mon -1

Display your cluster 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 status to an HTML file:

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

Start crm_mon and export the current cluster status as xml to stdout, then exit.:

 # crm_mon --as-xml


Report bugs to pacemaker@oss.clusterlabs.org
```

Note

If the SNMP and/or email options are not listed, then Pacemaker was not built to support them. This may be by the choice of your distribution or the required libraries may not have been available. Please contact whoever supplied you with the packages for more details.

Creating an Active/Passive Cluster

Table of Contents

5.1. 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.

```
# crm configure show
node pcmk-1
node pcmk-2
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    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.

The last XML you'll see in this document

```
# crm configure show xml
<?xml version="1.0" ?>
<cib admin_epoch="0" crm_feature_set="3.0.1" dc-uuid="pcmk-1" epoch="13" have-
quorum="1" num_updates="7" validate-with="pacemaker-1.0">
 <configuration>
  <crm_config>
   <cluster_property_set id="cib-bootstrap-options">
    <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.5-
bdd89e69ba545404d02445be1f3d72e6a203ba2f"/>
    <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>
  </crm_config>
  <rsc_defaults/>
  <op_defaults/>
  <nodes>
   <node id="pcmk-1" type="normal" uname="pcmk-1"/>
   <node id="pcmk-2" type="normal" uname="pcmk-2"/>
  </nodes>
  <resources/>
  <constraints/>
 </configuration>
</cib>
```

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

```
# 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
#
```

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

In order to guarantee the safety of your data [1] , Pacemaker ships with STONITH [2] 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
```

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

Warning

The use of stonith-enabled=false is completely inappropriate for a production cluster. We use it here to defer the discussion of its configuration which can differ widely from one installation to the next. See *Section 9.1, "What Is STONITH"* for information on why STONITH is important and details on how to configure it.

5.2. 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 192.168.122.101 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

[1] If the data is corrupt, there is little point in continuing to make it available
[2] A common node fencing mechanism. Used to ensure data integrity by powering off "bad" nodes

```
# crm configure primitive ClusterIP ocf:heartbeat:IPaddr2 \
     params ip=192.168.122.101 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

```
# crm ra classesheartbeat
lsb ocf / heartbeat pacemakerstonith
```

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

```
# crm ra list ocf pacemaker
ClusterMon   Dummy     Stateful    SysInfo    SystemHealth  controld
ping      pingd
# crm ra list ocf heartbeat
AoEtarget       AudibleAlarm      ClusterMon       Delay
Dummy         EvmsSCC        Evmsd         Filesystem
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    iSCSITarget      ids          iscsi
ldirectord       mysql         mysql-proxy      nfsserver
oracle          oralsnr        pgsql         pingd
portblock        rsyncd         scsi2reservation    sfex
tomcat          vmware
#
```

Now verify that the IP resource has been added and display the cluster's status to see that it is now active.

```
# crm configure shownode pcmk-1
node pcmk-2primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
# crm_mon
============
Last updated: Fri Aug 28 15:23:48 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
1 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]
```

```
ClusterIP (ocf::heartbeat:IPaddr): Started pcmk-1
```

5.3. 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.

```
# crm resource status ClusterIP
resource ClusterIP is running on: pcmk-1
#
```

Shut down Pacemaker and Corosync on that machine.

```
# ssh pcmk-1 -- /etc/init.d/pacemaker stop
Signaling Pacemaker Cluster Manager to terminate: [ OK ]
Waiting for cluster services to unload:. [ OK ]
# ssh pcmk-1 -- /etc/init.d/corosync stop
Stopping Corosync Cluster Engine (corosync): [ OK ]
Waiting for services to unload: [ OK ]
#
```

Once Corosync is no longer running, go to the other node and check the cluster status with crm_mon.

```
# crm_mon
============
Last updated: Fri Aug 28 15:27:35 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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!

5.3.1. Quorum and Two-Node Clusters

This is because the cluster no longer has quorum, as can be seen by the text "partition WITHOUT quorum" (emphasised 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:

```
total_nodes < 2 * 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 [3] , 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.

[3] Actually some would argue that two-node clusters are always pointless, but that is an argument for another time

```
# crm configure property no-quorum-policy=ignore
# crm configure show
node pcmk-1
node pcmk-2
primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
```

After a few moments, the cluster will start the IP address on the remaining node. Note that the cluster still does not have quorum.

```
# crm_mon
============
Last updated: Fri Aug 28 15:30:18 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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
Starting Corosync Cluster Engine (corosync): [ OK ]
# /etc/init.d/pacemaker start
Starting Pacemaker Cluster Manager: [ OK ]# crm_mon
============
Last updated: Fri Aug 28 15:32:13 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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!

5.3.2. Prevent Resources from Moving after Recovery

In some circumstances, it is highly desirable to prevent healthy resources from being moved around the cluster. Moving resources almost always requires a period of downtime. 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" [4] 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
# crm configure show
node pcmk-1
node pcmk-2
primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"rsc_defaults $id="rsc-options" \
    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/pacemaker stop
Signaling Pacemaker Cluster Manager to terminate:          [  OK  ]
Waiting for cluster services to unload:.                   [  OK  ]
# ssh pcmk-1 -- /etc/init.d/corosync stop
Stopping Corosync Cluster Engine (corosync):        [ OK ]
Waiting for services to unload:              [ OK ]
# ssh pcmk-2 -- crm_mon -1
============
Last updated: Fri Aug 28 15:39:38 2009
Stack: openais
Current DC: pcmk-2 - partition WITHOUT quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
1 Resources configured.
============

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

However when we bring pcmk-1 back online, ClusterIP now remains running on pcmk-2.

```
# /etc/init.d/corosync start
Starting Corosync Cluster Engine (corosync): [ OK ]
# /etc/init.d/pacemaker start
Starting Pacemaker Cluster Manager: [ OK ]
# crm_mon
============
Last updated: Fri Aug 28 15:41:23 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
1 Resources configured.
============
```

[4] 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 creates implicit preferences in situations where the administrator has not explicitly specified them

```
Online: [ pcmk-1 pcmk-2 ]

ClusterIP    (ocf::heartbeat:IPaddr):    Started pcmk-2
```

Apache - Adding More Services

Table of Contents

6.1. Forward

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.

6.2. Installation

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

```
# yum install -y httpdSetting 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

================================================================================
Package        Arch       Version        Repository     Size
================================================================================
Installing:
httpd        x86_64      2.2.13-2.fc12      rawhide      735 k
Installing for dependencies:
apr          x86_64      1.3.9-2.fc12       rawhide      117 k
apr-util       x86_64      1.3.9-2.fc12       rawhide       84 k
apr-util-ldap    x86_64      1.3.9-2.fc12       rawhide       15 k
httpd-tools      x86_64      2.2.13-2.fc12      rawhide       63 k
mailcap        noarch      2.1.30-1.fc12      rawhide       25 k

Transaction Summary
```

```
================================================================================
Install    6 Package(s)
Upgrade    0 Package(s)

Total download size: 1.0 M
Downloading Packages:
(1/6): apr-1.3.9-2.fc12.x86_64.rpm                 | 117 kB   00:00
(2/6): apr-util-1.3.9-2.fc12.x86_64.rpm               | 84 kB   00:00
(3/6): apr-util-ldap-1.3.9-2.fc12.x86_64.rpm           | 15 kB   00:00
(4/6): httpd-2.2.13-2.fc12.x86_64.rpm                | 735 kB   00:00
(5/6): httpd-tools-2.2.13-2.fc12.x86_64.rpm             | 63 kB   00:00
(6/6): mailcap-2.1.30-1.fc12.noarch.rpm               | 25 kB   00:00
 --------------------------------------------------------------------------------
Total                           875 kB/s | 1.0 MB   00:01
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
 Installing   : apr-1.3.9-2.fc12.x86_64                      1/6
 Installing   : apr-util-1.3.9-2.fc12.x86_64                  2/6
 Installing   : apr-util-ldap-1.3.9-2.fc12.x86_64                3/6
 Installing   : httpd-tools-2.2.13-2.fc12.x86_64               4/6
 Installing   : mailcap-2.1.30-1.fc12.noarch                  5/6
 Installing   : httpd-2.2.13-2.fc12.x86_64                    6/6

Installed:
 httpd.x86_64 0:2.2.13-2.fc12

Dependency Installed:
 apr.x86_64 0:1.3.9-2.fc12      apr-util.x86_64 0:1.3.9-2.fc12
 apr-util-ldap.x86_64 0:1.3.9-2.fc12 httpd-tools.x86_64 0:2.2.13-2.fc12
 mailcap.noarch 0:2.1.30-1.fc12

Complete!
```

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 wgetSetting 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:
wget-1.11.4-5.fc12.x86_64.rpm                      | 393 kB   00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
 Installing   : wget-1.11.4-5.fc12.x86_64                     1/1
```

```
Installed:
 wget.x86_64 0:1.11.4-5.fc12

Complete!
```

6.3. 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.

```
[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.

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

6.4. Enable the Apache status URL

In order to monitor the health of your Apache instance, and recover it if it fails, the resource agent used by Pacemaker assumes the server-status URL is available. Look for the following in /etc/httpd/conf/httpd.conf and make sure it is not disabled or commented out:

```
<Location /server-status>
   SetHandler server-status
   Order deny,allow
   Deny from all
   Allow from 127.0.0.1
</Location>
```

6.5. Update the Configuration

At this point, Apache is ready to go, all that needs to be done is to add it to the cluster. Lets call the resource WebSite. We need to use an OCF script called apache in the heartbeat namespace [1] , 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.

```
# crm configure primitive WebSite ocf:heartbeat:apache params configfile=/etc/httpd/conf/
httpd.conf op monitor interval=1min
# crm configure show
node pcmk-1
node pcmk-2primitive WebSite ocf:heartbeat:apache \ params configfile="/etc/httpd/conf/
httpd.conf" \ op monitor interval="1min"primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
```

[1] Compare the key used here ocf:heartbeat:apache with the one we used earlier for the IP address: ocf:heartbeat:IPaddr2

```
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

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

```
# crm_mon
============
Last updated: Fri Aug 28 16:12:49 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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-1
```

Wait a moment, the WebSite resource isn't running on the same host as our IP address!

6.6. 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.

For the constraint, we need a name (choose something descriptive like website-with-ip), indicate that its mandatory (so that if ClusterIP is not active anywhere, WebSite will not be permitted to run anywhere either) by specifying a score of INFINITY and finally list the two resources.

Note

If ClusterIP is not active anywhere, WebSite will not be permitted to run anywhere.

Important

Colocation constraints are "directional", in that they imply certain things about the order in which the two resources will have a location chosen. In this case we're saying **`WebSite`** needs to be placed on the same machine as **`ClusterIP`**, this implies that we must know the location of **`ClusterIP`** before choosing a location for **`WebSite`**.

```
# crm configure colocation website-with-ip INFINITY: WebSite ClusterIP
# 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="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"colocation website-with-ip inf: WebSite
 ClusterIPproperty $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
# crm_mon
============
Last updated: Fri Aug 28 16:14:34 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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
```

6.7. 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 (choose 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
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="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"
colocation website-with-ip inf: WebSite ClusterIPorder apache-after-ip inf: ClusterIP
 WebSiteproperty $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

6.8. 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
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="192.168.122.101" cidr_netmask="32" \
    op monitor interval="30s"location prefer-pcmk-1 WebSite 50: pcmk-1colocation website-
with-ip inf: WebSite ClusterIP
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
# crm_mon
============
Last updated: Fri Aug 28 16:17:35 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
2 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

ClusterIP    (ocf::heartbeat:IPaddr):    Started pcmk-2WebSite    (ocf::heartbeat:apache):
  Started pcmk-2
```

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
```

Note

Include output There is a way to force them to move though…

6.9. 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
============
Last updated: Fri Aug 28 16:19:24 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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
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="192.168.122.101" 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="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    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

6.9.1. 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.

```
# crm resource unmove WebSite
# 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="192.168.122.101" 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="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    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.

```
# crm_mon
============
Last updated: Fri Aug 28 16:20:53 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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
```

Replicated Storage with DRBD

Table of Contents

7.1. Background

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.

7.2. Install the DRBD Packages

Since its inclusion in the upstream 2.6.33 kernel, everything needed to use DRBD ships with Fedora 13. All you need to do is install it:

```
# yum install -y drbd-pacemaker drbd-udev
Loaded plugins: presto, refresh-packagekit
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package drbd-pacemaker.x86_64 0:8.3.7-2.fc13 set to be updated
--> Processing Dependency: drbd-utils = 8.3.7-2.fc13 for package: drbd-
pacemaker-8.3.7-2.fc13.x86_64
--> Running transaction check
---> Package drbd-utils.x86_64 0:8.3.7-2.fc13 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch           Version              Repository      Size
================================================================================
Installing:
 drbd-pacemaker         x86_64         8.3.7-2.fc13         fedora          19 k
Installing for dependencies:
 drbd-utils             x86_64         8.3.7-2.fc13         fedora         165 k

Transaction Summary
================================================================================
Install       2 Package(s)
Upgrade       0 Package(s)

Total download size: 184 k
Installed size: 427 k
Downloading Packages:
```

```
Setting up and reading Presto delta metadata
fedora/prestodelta                                       | 1.7 kB     00:00
Processing delta metadata
Package(s) data still to download: 184 k
(1/2): drbd-pacemaker-8.3.7-2.fc13.x86_64.rpm            |  19 kB     00:01
(2/2): drbd-utils-8.3.7-2.fc13.x86_64.rpm                | 165 kB     00:02
 ---------------------------------------------------------------------------
Total                                             45 kB/s | 184 kB     00:04
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing     : drbd-utils-8.3.7-2.fc13.x86_64                          1/2
  Installing     : drbd-pacemaker-8.3.7-2.fc13.x86_64                      2/2

Installed:
  drbd-pacemaker.x86_64 0:8.3.7-2.fc13

Dependency Installed:
  drbd-utils.x86_64 0:8.3.7-2.fc13

Complete!
```

7.3. Configure DRBD

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

7.3.1. 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
Logical volume "drbd-demo" created
# lvs
LV    VG    Attr   LSize   Origin Snap% Move Log Copy% Convert
drbd-demo VolGroup -wi-a- 1.00G
lv_root  VolGroup -wi-ao  7.30G
lv_swap  VolGroup -wi-ao 500.00M
```

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

```
# ssh pcmk-2 -- lvs
LV   VG    Attr   LSize   Origin Snap% Move Log Copy% Convert
lv_root VolGroup -wi-ao  7.30G
lv_swap VolGroup -wi-ao 500.00M
# ssh pcmk-2 -- lvcreate -n drbd-demo -L 1G VolGroup
Logical volume "drbd-demo" created
# ssh pcmk-2 -- lvs
LV    VG    Attr   LSize   Origin Snap% Move Log Copy% Convert
drbd-demo VolGroup -wi-a- 1.00G
lv_root  VolGroup -wi-ao  7.30G
lv_swap  VolGroup -wi-ao 500.00M
```

7.3.2. Write the DRBD Config

There is no series of commands for building 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*

Warning

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;
 }
}
```

Note

TODO: Explain the reason for the allow-two-primaries option

7.3.3. Initialize and Load DRBD

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

```
# drbdadm create-md wwwdata
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
```

Now load the DRBD kernel module and confirm that everything is sane

```
# modprobe drbd# drbdadm up wwwdata# cat /proc/drbdversion: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf build by root@pcmk-1, 2009-12-08 11:22:57
 1: cs:WFConnection ro:Secondary/Unknown ds:Inconsistent/DUnknown 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
```

Repeat on the second node

```
# ssh pcmk-2 -- drbdadm --force create-md wwwdata
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
# ssh pcmk-2 -- modprobe drbd
WARNING: Deprecated config file /etc/modprobe.conf, all config files belong into /etc/
modprobe.d/.
# ssh pcmk-2 -- drbdadm up wwwdata
# ssh pcmk-2 -- cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf 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:

```
# drbdadm -- --overwrite-data-of-peer primary wwwdata
# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf 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
    [=====>.............] synced: 33.4% (10064/12248)K
    finish: 0:00:37 speed: 240 (240) K/sec
# cat /proc/drbd
version: 8.3.6 (api:88/proto:86-90)
GIT-hash: f3606c47cc6fcf6b3f086e425cb34af8b7a81bbf 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:0 bm: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 it's a good point at which to create a filesystem and populate it with some data to serve up via our WebSite resource.

7.3.4. Populate DRBD with Data

```
# 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:
```

```
    8193

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
```

7.4. 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.

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

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

```
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.

```
crm(drbd) # configure primitive WebData ocf:linbit:drbd 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=truecrm(drbd) # configure shownode pcmk-1
node pcmk-2primitive 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="192.168.122.101" 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
colocation website-with-ip inf: WebSite ClusterIP
order apache-after-ip inf: ClusterIP WebSite
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    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.

```
crm(drbd) # cib commit drbdINFO: commited 'drbd' shadow CIB to the cluster
crm(drbd) # quitbye
# crm_mon
============
Last updated: Tue Sep 1 09:37:13 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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-1Master/Slave Set: WebDataClone Masters: [
 pcmk-2 ] Slaves: [ pcmk-1 ]
```

Note

Include details on adding a second DRBD resource

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
crm(live) # cib new fs
INFO: fs shadow CIB created
crm(fs) # configure primitive WebFS ocf:heartbeat:Filesystem \
    params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="ext4"
crm(fs) # configure colocation fs_on_drbd inf: WebFS WebDataClone:Master
crm(fs) # 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.

crm(fs) # configure colocation WebSite-with-WebFS inf: WebSite WebFS
crm(fs) # configure order WebSite-after-WebFS inf: WebFS WebSite
```

Time to review the updated configuration:

```
crm(fs) # 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="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" 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
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="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

After reviewing the new configuration, we again upload it and watch the cluster put it into effect.

```
crm(fs) # cib commit fs
INFO: commited 'fs' shadow CIB to the cluster
crm(fs) # quit
bye
# crm_mon
============
Last updated: Tue Sep 1 10:08:44 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
4 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-1 ]
    Slaves: [ pcmk-2 ]
WebFS (ocf::heartbeat:Filesystem): Started pcmk-1
```

7.4.1. 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
============
Last updated: Tue Sep 1 10:09:57 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
4 Resources configured.
============
Node pcmk-1: standbyOnline: [ 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
============
Last updated: Tue Sep 1 10:13:25 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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

Table of Contents

8.1. Requirements

The primary requirement for an Active/Active cluster is that the data required for your services is 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. 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.

We'll also need to use CMAN for Cluster Membership and Quorum instead of our Corosync plugin.

8.2. Adding CMAN Support

CMAN v3[1] is a Corosync plugin that monitors the names and number of active cluster nodes in order to deliver membership and quorum information to clients (such as the Pacemaker daemons).

In a traditional Corosync-Pacemaker cluster, a Pacemaker plugin is loaded to provide membership and quorum information. The motivation for wanting to use CMAN for this instead, is to ensure all elements of the cluster stack are making decisions based on the same membership and quorum data.[2]

In the case of GFS2, the key pieces are the dlm_controld and gfs_controld helpers which act as the glue between the filesystem and the cluster software. Supporting CMAN enables us to use the versions already being shipped by most distributions (since CMAN has been around longer than Pacemaker and is part of the Red Hat cluster stack).

[1] http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html-single/Cluster_Suite_Overview/index.html#s2-clumembership-overview-CSO

[2] A failure to do this can lead to what is called *internal split-brain* - a situation where different parts of the stack disagree about whether some nodes are alive or dead - which quickly leads to unnecessary down-time and/or data corruption.

Warning

Ensure Corosync and Pacemaker are stopped on all nodes before continuing

Warning

Be sure to disable the Pacemaker plugin before continuing with this section. In most cases, this can be achieved by removing /etc/corosync/service.d/pcmk and stopping Corosync.

8.2.1. Installing the required Software

```
# yum install -y cman gfs2-utils gfs2-cluster
Loaded plugins: auto-update-debuginfo
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package cman.x86_64 0:3.1.7-1.fc15 will be installed
--> Processing Dependency: modcluster >= 0.18.1-1 for package: cman-3.1.7-1.fc15.x86_64
--> Processing Dependency: fence-agents >= 3.1.5-1 for package: cman-3.1.7-1.fc15.x86_64
--> Processing Dependency: openais >= 1.1.4-1 for package: cman-3.1.7-1.fc15.x86_64
--> Processing Dependency: ricci >= 0.18.1-1 for package: cman-3.1.7-1.fc15.x86_64
--> Processing Dependency: libSaCkpt.so.3(OPENAIS_CKPT_B.01.01)(64bit) for package:
 cman-3.1.7-1.fc15.x86_64
--> Processing Dependency: libSaCkpt.so.3()(64bit) for package: cman-3.1.7-1.fc15.x86_64
---> Package gfs2-cluster.x86_64 0:3.1.1-2.fc15 will be installed
---> Package gfs2-utils.x86_64 0:3.1.1-2.fc15 will be installed
--> Running transaction check
---> Package fence-agents.x86_64 0:3.1.5-1.fc15 will be installed
--> Processing Dependency: /usr/bin/virsh for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: net-snmp-utils for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: sg3_utils for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: perl(Net::Telnet) for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: /usr/bin/ipmitool for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: perl-Net-Telnet for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: pexpect for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: pyOpenSSL for package: fence-agents-3.1.5-1.fc15.x86_64
--> Processing Dependency: python-suds for package: fence-agents-3.1.5-1.fc15.x86_64
---> Package modcluster.x86_64 0:0.18.7-1.fc15 will be installed
--> Processing Dependency: oddjob for package: modcluster-0.18.7-1.fc15.x86_64
---> Package openais.x86_64 0:1.1.4-2.fc15 will be installed
---> Package openaislib.x86_64 0:1.1.4-2.fc15 will be installed
---> Package ricci.x86_64 0:0.18.7-1.fc15 will be installed
--> Processing Dependency: parted for package: ricci-0.18.7-1.fc15.x86_64
--> Processing Dependency: nss-tools for package: ricci-0.18.7-1.fc15.x86_64
--> Running transaction check
---> Package ipmitool.x86_64 0:1.8.11-6.fc15 will be installed
---> Package libvirt-client.x86_64 0:0.8.8-7.fc15 will be installed
--> Processing Dependency: libnetcf.so.1(NETCF_1.3.0)(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: cyrus-sasl-md5 for package: libvirt-client-0.8.8-7.fc15.x86_64
--> Processing Dependency: gettext for package: libvirt-client-0.8.8-7.fc15.x86_64
--> Processing Dependency: nc for package: libvirt-client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnuma.so.1(libnuma_1.1)(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
```

```
--> Processing Dependency: libnuma.so.1(libnuma_1.2)(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnetcf.so.1(NETCF_1.2.0)(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: gnutls-utils for package: libvirt-client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnetcf.so.1(NETCF_1.0.0)(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libxenstore.so.3.0()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libyajl.so.1()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnl.so.1()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnuma.so.1()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libaugeas.so.0()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
--> Processing Dependency: libnetcf.so.1()(64bit) for package: libvirt-
client-0.8.8-7.fc15.x86_64
---> Package net-snmp-utils.x86_64 1:5.6.1-7.fc15 will be installed
---> Package nss-tools.x86_64 0:3.12.10-6.fc15 will be installed
---> Package oddjob.x86_64 0:0.31-2.fc15 will be installed
---> Package parted.x86_64 0:2.3-10.fc15 will be installed
---> Package perl-Net-Telnet.noarch 0:3.03-12.fc15 will be installed
---> Package pexpect.noarch 0:2.3-6.fc15 will be installed
---> Package pyOpenSSL.x86_64 0:0.10-3.fc15 will be installed
---> Package python-suds.noarch 0:0.3.9-3.fc15 will be installed
---> Package sg3_utils.x86_64 0:1.29-3.fc15 will be installed
--> Processing Dependency: sg3_utils-libs = 1.29-3.fc15 for package:
 sg3_utils-1.29-3.fc15.x86_64
--> Processing Dependency: libsgutils2.so.2()(64bit) for package:
 sg3_utils-1.29-3.fc15.x86_64
--> Running transaction check
---> Package augeas-libs.x86_64 0:0.9.0-1.fc15 will be installed
---> Package cyrus-sasl-md5.x86_64 0:2.1.23-18.fc15 will be installed
---> Package gettext.x86_64 0:0.18.1.1-7.fc15 will be installed
--> Processing Dependency: libgomp.so.1(GOMP_1.0)(64bit) for
 package: gettext-0.18.1.1-7.fc15.x86_64
--> Processing Dependency: libgettextlib-0.18.1.so()(64bit) for
 package: gettext-0.18.1.1-7.fc15.x86_64
--> Processing Dependency: libgettextsrc-0.18.1.so()(64bit) for
 package: gettext-0.18.1.1-7.fc15.x86_64
--> Processing Dependency: libgomp.so.1()(64bit) for package: gettext-0.18.1.1-7.fc15.x86_64
---> Package gnutls-utils.x86_64 0:2.10.5-1.fc15 will be installed
---> Package libnl.x86_64 0:1.1-14.fc15 will be installed
---> Package nc.x86_64 0:1.100-3.fc15 will be installed
--> Processing Dependency: libbsd.so.0(LIBBSD_0.0)(64bit) for package: nc-1.100-3.fc15.x86_64
--> Processing Dependency: libbsd.so.0(LIBBSD_0.2)(64bit) for package: nc-1.100-3.fc15.x86_64
--> Processing Dependency: libbsd.so.0()(64bit) for package: nc-1.100-3.fc15.x86_64
---> Package netcf-libs.x86_64 0:0.1.9-1.fc15 will be installed
---> Package numactl.x86_64 0:2.0.7-1.fc15 will be installed
---> Package sg3_utils-libs.x86_64 0:1.29-3.fc15 will be installed
---> Package xen-libs.x86_64 0:4.1.1-3.fc15 will be installed
--> Processing Dependency: xen-licenses for package: xen-libs-4.1.1-3.fc15.x86_64
---> Package yajl.x86_64 0:1.0.11-1.fc15 will be installed
--> Running transaction check
---> Package gettext-libs.x86_64 0:0.18.1.1-7.fc15 will be installed
---> Package libbsd.x86_64 0:0.2.0-4.fc15 will be installed
---> Package libgomp.x86_64 0:4.6.1-9.fc15 will be installed
---> Package xen-licenses.x86_64 0:4.1.1-3.fc15 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package              Arch        Version                 Repository    Size
================================================================================
```

```
Installing:
 cman                 x86_64      3.1.7-1.fc15            updates      366 k
 gfs2-cluster         x86_64      3.1.1-2.fc15            fedora        69 k
 gfs2-utils           x86_64      3.1.1-2.fc15            fedora       222 k
Installing for dependencies:
 augeas-libs          x86_64      0.9.0-1.fc15            updates      311 k
 cyrus-sasl-md5       x86_64      2.1.23-18.fc15          updates       46 k
 fence-agents         x86_64      3.1.5-1.fc15            updates      186 k
 gettext              x86_64      0.18.1.1-7.fc15         fedora       1.0 M
 gettext-libs         x86_64      0.18.1.1-7.fc15         fedora       610 k
 gnutls-utils         x86_64      2.10.5-1.fc15           fedora       101 k
 ipmitool             x86_64      1.8.11-6.fc15           fedora       273 k
 libbsd               x86_64      0.2.0-4.fc15            fedora        37 k
 libgomp              x86_64      4.6.1-9.fc15            updates       95 k
 libnl                x86_64      1.1-14.fc15             fedora       118 k
 libvirt-client       x86_64      0.8.8-7.fc15            updates      2.4 M
 modcluster           x86_64      0.18.7-1.fc15           fedora       187 k
 nc                   x86_64      1.100-3.fc15            updates       24 k
 net-snmp-utils       x86_64      1:5.6.1-7.fc15          fedora       180 k
 netcf-libs           x86_64      0.1.9-1.fc15            updates       50 k
 nss-tools            x86_64      3.12.10-6.fc15          updates      723 k
 numactl              x86_64      2.0.7-1.fc15            updates       54 k
 oddjob               x86_64      0.31-2.fc15             fedora        61 k
 openais              x86_64      1.1.4-2.fc15            fedora       190 k
 openaislib           x86_64      1.1.4-2.fc15            fedora        88 k
 parted               x86_64      2.3-10.fc15             updates      618 k
 perl-Net-Telnet      noarch      3.03-12.fc15            fedora        55 k
 pexpect              noarch      2.3-6.fc15              fedora       141 k
 pyOpenSSL            x86_64      0.10-3.fc15             fedora       198 k
 python-suds          noarch      0.3.9-3.fc15            fedora       195 k
 ricci                x86_64      0.18.7-1.fc15           fedora       584 k
 sg3_utils            x86_64      1.29-3.fc15             fedora       465 k
 sg3_utils-libs       x86_64      1.29-3.fc15             fedora        54 k
 xen-libs             x86_64      4.1.1-3.fc15            updates      310 k
 xen-licenses         x86_64      4.1.1-3.fc15            updates       64 k
 yajl                 x86_64      1.0.11-1.fc15           fedora        27 k

Transaction Summary
================================================================================
Install      34 Package(s)

Total download size: 10 M
Installed size: 38 M
Downloading Packages:
(1/34): augeas-libs-0.9.0-1.fc15.x86_64.rpm           | 311 kB     00:00
(2/34): cman-3.1.7-1.fc15.x86_64.rpm                  | 366 kB     00:00
(3/34): cyrus-sasl-md5-2.1.23-18.fc15.x86_64.rpm      |  46 kB     00:00
(4/34): fence-agents-3.1.5-1.fc15.x86_64.rpm          | 186 kB     00:00
(5/34): gettext-0.18.1.1-7.fc15.x86_64.rpm            | 1.0 MB     00:01
(6/34): gettext-libs-0.18.1.1-7.fc15.x86_64.rpm       | 610 kB     00:00
(7/34): gfs2-cluster-3.1.1-2.fc15.x86_64.rpm          |  69 kB     00:00
(8/34): gfs2-utils-3.1.1-2.fc15.x86_64.rpm            | 222 kB     00:00
(9/34): gnutls-utils-2.10.5-1.fc15.x86_64.rpm         | 101 kB     00:00
(10/34): ipmitool-1.8.11-6.fc15.x86_64.rpm            | 273 kB     00:00
(11/34): libbsd-0.2.0-4.fc15.x86_64.rpm               |  37 kB     00:00
(12/34): libgomp-4.6.1-9.fc15.x86_64.rpm              |  95 kB     00:00
(13/34): libnl-1.1-14.fc15.x86_64.rpm                 | 118 kB     00:00
(14/34): libvirt-client-0.8.8-7.fc15.x86_64.rpm       | 2.4 MB     00:01
(15/34): modcluster-0.18.7-1.fc15.x86_64.rpm          | 187 kB     00:00
(16/34): nc-1.100-3.fc15.x86_64.rpm                   |  24 kB     00:00
(17/34): net-snmp-utils-5.6.1-7.fc15.x86_64.rpm       | 180 kB     00:00
(18/34): netcf-libs-0.1.9-1.fc15.x86_64.rpm           |  50 kB     00:00
(19/34): nss-tools-3.12.10-6.fc15.x86_64.rpm          | 723 kB     00:00
(20/34): numactl-2.0.7-1.fc15.x86_64.rpm              |  54 kB     00:00
(21/34): oddjob-0.31-2.fc15.x86_64.rpm                |  61 kB     00:00
(22/34): openais-1.1.4-2.fc15.x86_64.rpm              | 190 kB     00:00
(23/34): openaislib-1.1.4-2.fc15.x86_64.rpm           |  88 kB     00:00
```

```
(24/34): parted-2.3-10.fc15.x86_64.rpm                | 618 kB     00:00
(25/34): perl-Net-Telnet-3.03-12.fc15.noarch.rpm      |  55 kB     00:00
(26/34): pexpect-2.3-6.fc15.noarch.rpm                | 141 kB     00:00
(27/34): pyOpenSSL-0.10-3.fc15.x86_64.rpm             | 198 kB     00:00
(28/34): python-suds-0.3.9-3.fc15.noarch.rpm          | 195 kB     00:00
(29/34): ricci-0.18.7-1.fc15.x86_64.rpm               | 584 kB     00:00
(30/34): sg3_utils-1.29-3.fc15.x86_64.rpm             | 465 kB     00:00
(31/34): sg3_utils-libs-1.29-3.fc15.x86_64.rpm        |  54 kB     00:00
(32/34): xen-libs-4.1.1-3.fc15.x86_64.rpm             | 310 kB     00:00
(33/34): xen-licenses-4.1.1-3.fc15.x86_64.rpm         |  64 kB     00:00
(34/34): yajl-1.0.11-1.fc15.x86_64.rpm                |  27 kB     00:00
 ---------------------------------------------------------------------------
Total                                        803 kB/s |  10 MB     00:12
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : openais-1.1.4-2.fc15.x86_64                               1/34
  Installing : openaislib-1.1.4-2.fc15.x86_64                            2/34
  Installing : libnl-1.1-14.fc15.x86_64                                  3/34
  Installing : augeas-libs-0.9.0-1.fc15.x86_64                           4/34
  Installing : oddjob-0.31-2.fc15.x86_64                                 5/34
  Installing : modcluster-0.18.7-1.fc15.x86_64                           6/34
  Installing : netcf-libs-0.1.9-1.fc15.x86_64                            7/34
  Installing : 1:net-snmp-utils-5.6.1-7.fc15.x86_64                      8/34
  Installing : sg3_utils-libs-1.29-3.fc15.x86_64                         9/34
  Installing : sg3_utils-1.29-3.fc15.x86_64                             10/34
  Installing : libgomp-4.6.1-9.fc15.x86_64                              11/34
  Installing : gnutls-utils-2.10.5-1.fc15.x86_64                        12/34
  Installing : pyOpenSSL-0.10-3.fc15.x86_64                             13/34
  Installing : parted-2.3-10.fc15.x86_64                                14/34
  Installing : cyrus-sasl-md5-2.1.23-18.fc15.x86_64                     15/34
  Installing : python-suds-0.3.9-3.fc15.noarch                          16/34
  Installing : ipmitool-1.8.11-6.fc15.x86_64                            17/34
  Installing : perl-Net-Telnet-3.03-12.fc15.noarch                      18/34
  Installing : numactl-2.0.7-1.fc15.x86_64                              19/34
  Installing : yajl-1.0.11-1.fc15.x86_64                                20/34
  Installing : gettext-libs-0.18.1.1-7.fc15.x86_64                      21/34
  Installing : gettext-0.18.1.1-7.fc15.x86_64                           22/34
  Installing : libbsd-0.2.0-4.fc15.x86_64                               23/34
  Installing : nc-1.100-3.fc15.x86_64                                   24/34
  Installing : xen-licenses-4.1.1-3.fc15.x86_64                         25/34
  Installing : xen-libs-4.1.1-3.fc15.x86_64                             26/34
  Installing : libvirt-client-0.8.8-7.fc15.x86_64                       27/34

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

  Installing : nss-tools-3.12.10-6.fc15.x86_64                          28/34
  Installing : ricci-0.18.7-1.fc15.x86_64                               29/34
  Installing : pexpect-2.3-6.fc15.noarch                                30/34
  Installing : fence-agents-3.1.5-1.fc15.x86_64                         31/34
  Installing : cman-3.1.7-1.fc15.x86_64                                 32/34
  Installing : gfs2-cluster-3.1.1-2.fc15.x86_64                         33/34
  Installing : gfs2-utils-3.1.1-2.fc15.x86_64                           34/34

Installed:
  cman.x86_64 0:3.1.7-1.fc15           gfs2-cluster.x86_64 0:3.1.1-2.fc15
  gfs2-utils.x86_64 0:3.1.1-2.fc15

Dependency Installed:
  augeas-libs.x86_64 0:0.9.0-1.fc15
  cyrus-sasl-md5.x86_64 0:2.1.23-18.fc15
  fence-agents.x86_64 0:3.1.5-1.fc15
  gettext.x86_64 0:0.18.1.1-7.fc15
  gettext-libs.x86_64 0:0.18.1.1-7.fc15
```

```
  gnutls-utils.x86_64 0:2.10.5-1.fc15
  ipmitool.x86_64 0:1.8.11-6.fc15
  libbsd.x86_64 0:0.2.0-4.fc15
  libgomp.x86_64 0:4.6.1-9.fc15
  libnl.x86_64 0:1.1-14.fc15
  libvirt-client.x86_64 0:0.8.8-7.fc15
  modcluster.x86_64 0:0.18.7-1.fc15
  nc.x86_64 0:1.100-3.fc15
  net-snmp-utils.x86_64 1:5.6.1-7.fc15
  netcf-libs.x86_64 0:0.1.9-1.fc15
  nss-tools.x86_64 0:3.12.10-6.fc15
  numactl.x86_64 0:2.0.7-1.fc15
  oddjob.x86_64 0:0.31-2.fc15
  openais.x86_64 0:1.1.4-2.fc15
  openaislib.x86_64 0:1.1.4-2.fc15
  parted.x86_64 0:2.3-10.fc15
  perl-Net-Telnet.noarch 0:3.03-12.fc15
  pexpect.noarch 0:2.3-6.fc15
  pyOpenSSL.x86_64 0:0.10-3.fc15
  python-suds.noarch 0:0.3.9-3.fc15
  ricci.x86_64 0:0.18.7-1.fc15
  sg3_utils.x86_64 0:1.29-3.fc15
  sg3_utils-libs.x86_64 0:1.29-3.fc15
  xen-libs.x86_64 0:4.1.1-3.fc15
  xen-licenses.x86_64 0:4.1.1-3.fc15
  yajl.x86_64 0:1.0.11-1.fc15

Complete!
```

8.2.2. Configuring CMAN

Note

The standard Pacemaker config file will continue to be used for resource management even after we start using CMAN. There is no need to recreate all your resources and constraints to the *cluster.conf* syntax, we simply create a minimal version that lists the nodes.

The first thing we need to do, is tell CMAN complete starting up even without quorum. We can do this by changing the quorum timeout setting:

```
# sed -i.sed "s/.*CMAN_QUORUM_TIMEOUT=.*/CMAN_QUORUM_TIMEOUT=0/g" /etc/sysconfig/cman
```

Next we create a basic configuration file and place it in /etc/cluster/cluster.conf. The name used for each clusternode should correspond to that node's uname -n, just as Pacemaker expects. The nodeid can be any positive mumber but must be unique.

Basic cluster.conf for a two-node cluster

```
<?xml version="1.0"?>
<cluster config_version="1" name="my_cluster_name">
  <logging debug="off"/>
  <clusternodes>
    <clusternode name="pcmk-1" nodeid="1"/>
    <clusternode name="pcmk-2" nodeid="2"/>
  </clusternodes>
```

```
</cluster>
```

8.2.3. Redundant Rings

For those wishing to use Corosync's multiple rings feature, simply define an alternate name for each node. For example:

```
    <clusternode name="pcmk-1" nodeid="1"/>
        <altname name="pcmk-1-internal"/>
    </clusternode>
```

8.2.4. Configuring CMAN Fencing

We configure the fence_pcmk agent (supplied with Pacemaker) to redirect any fencing requests from CMAN components (such as dlm_controld) to Pacemaker. Pacemaker's fencing subsystem lets other parts of the stack know that a node has been successfully fenced, thus avoiding the need for it to be fenced again when other subsystems notice the node is gone.

Warning

Configuring real fencing devices in CMAN will result in nodes being fenced multiple times as different parts of the stack notice the node is missing or failed.

The definition should be placed in the fencedevices section and contain:

```
<fencedevice name="pcmk" agent="fence_pcmk"/>
```

Each clusternode must be configured to use this device by adding a fence method block that lists the node's name as the port.

```
<fence>
  <method name="pcmk-redirect">
    <device name="pcmk" port="node_name_here"/>
  </method>
</fence>
```

Putting everything together, we have:

cluster.conf for a two-node cluster with fencing

```
<?xml version="1.0"?>
<cluster config_version="1" name="mycluster">
  <logging debug="off"/>
  <clusternodes>
    <clusternode name="pcmk-1" nodeid="1">
      <fence>
        <method name="pcmk-redirect">
          <device name="pcmk" port="pcmk-1"/>
        </method>
      </fence>
    </clusternode>
```

```
    <clusternode name="pcmk-2" nodeid="2">
      <fence>
        <method name="pcmk-redirect">
          <device name="pcmk" port="pcmk-2"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice name="pcmk" agent="fence_pcmk"/>
  </fencedevices>
</cluster>
```

8.2.5. Bringing the Cluster Online with CMAN

The first thing to do is check that the configuration is valid

```
# ccs_config_validate
Configuration validates
```

Now start CMAN

```
# service cman start
Starting cluster:
   Checking Network Manager...                             [  OK  ]
   Global setup...                                         [  OK  ]
   Loading kernel modules...                               [  OK  ]
   Mounting configfs...                                    [  OK  ]
   Starting cman...                                        [  OK  ]
   Waiting for quorum...                                   [  OK  ]
   Starting fenced...                                      [  OK  ]
   Starting dlm_controld...                                [  OK  ]
   Starting gfs_controld...                                [  OK  ]
   Unfencing self...                                       [  OK  ]
   Joining fence domain...                                 [  OK  ]
```

Once you have confirmed that the first node is happily online, start the second node.

```
[root@pcmk-2 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [  OK  ]
   Global setup...                                         [  OK  ]
   Loading kernel modules...                               [  OK  ]
   Mounting configfs...                                    [  OK  ]
   Starting cman...                                        [  OK  ]
   Waiting for quorum...                                   [  OK  ]
   Starting fenced...                                      [  OK  ]
   Starting dlm_controld...                                [  OK  ]
   Starting gfs_controld...                                [  OK  ]
   Unfencing self...                                       [  OK  ]
   Joining fence domain...                                 [  OK  ]
# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2011-09-28 10:52:21  pcmk-1
   2   M    548   2011-09-28 10:52:21  pcmk-2
```

You should now see both nodes online. To begin managing resources, simply start Pacemaker.

```
# service pacemaker start
 Starting Pacemaker Cluster Manager: [  OK  ]
```

and again on the second node, after which point you can use crm_mon as you normally would.

```
[root@pcmk-2 ~]# service pacemaker start
 Starting Pacemaker Cluster Manager: [  OK  ]
# crm_mon -1
```

8.3. Create a GFS2 Filesystem

8.3.1. Preparation

Before we do anything to the existing partition, we need to make sure it is unmounted. We do this by telling 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
============
Last updated: Thu Sep 3 15:18:06 2009
Stack: openais
Current DC: pcmk-1 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
2 Nodes configured, 2 expected votes
6 Resources configured.
============

Online: [ pcmk-1 pcmk-2 ]

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

Note

Note that both Apache and WebFS have been stopped.

8.3.2. Create and Populate an GFS2 Partition

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

Warning

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 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
This will destroy any data on /dev/drbd1.
It appears to contain: data

Are you sure you want to proceed? [y/n] y

Device:          /dev/drbd1
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#
```

8.4. Reconfigure the Cluster for GFS2

```
# crm
crm(live) # cib new GFS2
INFO: GFS2 shadow CIB created
crm(GFS2) # configure delete WebFS
crm(GFS2) # 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.

```
crm(GFS2) # configure colocation WebSite-with-WebFS inf: WebSite WebFS
crm(GFS2) # configure colocation fs_on_drbd inf: WebFS WebDataClone:Master
crm(GFS2) # configure order WebFS-after-WebData inf: WebDataClone:promote WebFS:start
crm(GFS2) # configure order WebSite-after-WebFS inf: WebFS WebSite
crm(GFS2) # 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="192.168.122.101" 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"
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="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

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

```
crm(GFS2) # cib commit GFS2
INFO: commited 'GFS2' shadow CIB to the cluster
crm(GFS2) # quit
bye
# crm_mon
============
Last updated: Thu Sep 3 20:49:54 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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-2WebFS (ocf::heartbeat:Filesystem):
 Started pcmk-1
```

8.5. 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 lets first clone the IP address. Cloned IPaddr2 resources use an iptables rule to ensure that each request only gets 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="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"
```

Here is the full transcript

```
# crm crm(live)
# cib new active
INFO: active shadow CIB created
crm(active) # configure clone WebIP ClusterIP \
    meta globally-unique="true" clone-max="2" clone-node-max="2"
crm(active) # configure shownode 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="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"
ms WebDataClone WebData \
    meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone WebIP ClusterIP \
    meta globally-unique="true" clone-max="2" clone-node-max="2"
colocation WebSite-with-WebFS inf: WebSite WebFS
colocation fs_on_drbd inf: WebFS WebDataClone:Master
colocation website-with-ip inf: WebSite WebIPorder WebFS-after-WebData inf:
 WebDataClone:promote WebFS:start
order WebSite-after-WebFS inf: WebFS WebSiteorder apache-after-ip inf: WebIP WebSite
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
```

```
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

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.

```
crm(active) # configure clone WebFSClone WebFS
crm(active) # 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).

```
crm(active) # configure edit WebDataClone
```

Change master-max to 2

```
crm(active) # 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="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"
ms WebDataClone WebData \
    meta master-max="2" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
clone WebFSClone WebFSclone WebIP ClusterIP \
    meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSitecolocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
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
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

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

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

```
============
Last updated: Thu Sep 3 21:37:27 2009
Stack: openais
Current DC: pcmk-2 - partition with quorum
Version: 1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f
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: WebIP Started: [ pcmk-1 pcmk-2 ]
Clone Set: WebFSClone Started: [ pcmk-1 pcmk-2 ]
Clone Set: WebSiteClone Started: [ pcmk-1 pcmk-2 ]
```

8.5.1. Testing Recovery

Note

TODO: Put one node into standby to demonstrate failover

Configure STONITH

Table of Contents

9.1. What Is STONITH

STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and it protects your data from being corrupted by rogue 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.

9.2. 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 on-board 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.

9.3. Configuring STONITH

1. Find the correct driver: **`stonith_admin --list-installed`**

2. Since every device is different, the parameters needed to configure it will vary. To find out the parameters associated with the device, run: **`stonith_admin --metadata --agent type`**

   ```
   The output should be XML formatted text containing additional
   parameter descriptions. We will endevor to make the output more
   friendly in a later version.
   ```

3. Enter the shell crm Create an editable copy of the existing configuration cib new stonith Create a fencing resource 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: **`configure primitive …`**

4. If the device does not know how to fence nodes based on their uname, you may also need to set the special **`pcmk_host_map`** parameter. See **`man stonithd`** for details.

5. If the device does not support the list command, you may also need to set the special **pcmk_host_list** and/or **pcmk_host_check** parameters. See **man stonithd** for details.

6. If the device does not expect the victim to be specified with the port parameter, you may also need to set the special **pcmk_host_argument** parameter. See **man stonithd** for details.

7. Upload it into the CIB from the shell: **cib commit stonith**

8. Once the stonith resource is running, you can test it by executing: **stonith_admin --reboot nodename**. Although you might want to stop the cluster on that machine first.

9.4. Example

Assuming we have an chassis containing four nodes and an IPMI device active on 10.0.0.1, then we would chose the fence_ipmilan driver in step 2 and obtain the following list of parameters

Obtaining a list of STONITH Parameters

```
# stonith_admin --metadata -a fence_ipmilan
```

```
<?xml version="1.0" ?>
<resource-agent name="fence_ipmilan" shortdesc="Fence agent for IPMI over LAN">
<longdesc>
fence_ipmilan is an I/O Fencing agent which can be used with machines controlled by IPMI.
 This agent calls support software using ipmitool (http://ipmitool.sf.net/).

To use fence_ipmilan with HP iLO 3 you have to enable lanplus option (lanplus / -P) and
 increase wait after operation to 4 seconds (power_wait=4 / -T 4)</longdesc>
<parameters>
        <parameter name="auth" unique="1">
                <getopt mixed="-A" />
                <content type="string" />
                <shortdesc>IPMI Lan Auth type (md5, password, or none)</shortdesc>
        </parameter>
        <parameter name="ipaddr" unique="1">
                <getopt mixed="-a" />
                <content type="string" />
                <shortdesc>IPMI Lan IP to talk to</shortdesc>
        </parameter>
        <parameter name="passwd" unique="1">
                <getopt mixed="-p" />
                <content type="string" />
                <shortdesc>Password (if required) to control power on IPMI device</shortdesc>
        </parameter>
        <parameter name="passwd_script" unique="1">
                <getopt mixed="-S" />
                <content type="string" />
                <shortdesc>Script to retrieve password (if required)</shortdesc>
        </parameter>
        <parameter name="lanplus" unique="1">
                <getopt mixed="-P" />
                <content type="boolean" />
                <shortdesc>Use Lanplus</shortdesc>
        </parameter>
        <parameter name="login" unique="1">
                <getopt mixed="-l" />
                <content type="string" />
                <shortdesc>Username/Login (if required) to control power on IPMI device</
shortdesc>
        </parameter>
        <parameter name="action" unique="1">
```

```
                <getopt mixed="-o" />
                <content type="string" default="reboot"/>
                <shortdesc>Operation to perform. Valid operations: on, off, reboot, status,
 list, diag, monitor or metadata</shortdesc>
        </parameter>
        <parameter name="timeout" unique="1">
                <getopt mixed="-t" />
                <content type="string" />
                <shortdesc>Timeout (sec) for IPMI operation</shortdesc>
        </parameter>
        <parameter name="cipher" unique="1">
                <getopt mixed="-C" />
                <content type="string" />
                <shortdesc>Ciphersuite to use (same as ipmitool -C parameter)</shortdesc>
        </parameter>
        <parameter name="method" unique="1">
                <getopt mixed="-M" />
                <content type="string" default="onoff"/>
                <shortdesc>Method to fence (onoff or cycle)</shortdesc>
        </parameter>
        <parameter name="power_wait" unique="1">
                <getopt mixed="-T" />
                <content type="string" default="2"/>
                <shortdesc>Wait X seconds after on/off operation</shortdesc>
        </parameter>
        <parameter name="delay" unique="1">
                <getopt mixed="-f" />
                <content type="string" />
                <shortdesc>Wait X seconds before fencing is started</shortdesc>
        </parameter>
        <parameter name="verbose" unique="1">
                <getopt mixed="-v" />
                <content type="boolean" />
                <shortdesc>Verbose mode</shortdesc>
        </parameter>
</parameters>
<actions>
        <action name="on" />
        <action name="off" />
        <action name="reboot" />
        <action name="status" />
        <action name="diag" />
        <action name="list" />
        <action name="monitor" />
        <action name="metadata" />
</actions>
</resource-agent>
```

from which we would create a STONITH resource fragment that might look like this

Sample STONITH Resource

```
# crm crm(live)# cib new stonith
INFO: stonith shadow CIB created
crm(stonith)# configure primitive impi-fencing stonith::fence_ipmilan \
 params pcmk_host_list="pcmk-1 pcmk-2" ipaddr=10.0.0.1 login=testuser passwd=abc123 \
 op monitor interval="60s"
```

And finally, since we disabled it earlier, we need to re-enable STONITH. At this point we should have the following configuration.

```
crm(stonith)# configure property stonith-enabled="true"crm(stonith)# configure shownode
 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="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"primitive ipmi-fencing
 stonith::fence_ipmilan \ params pcmk_host_list="pcmk-1
 pcmk-2" ipaddr=10.0.0.1 login=testuser passwd=abc123 \ 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 WebFSClone WebFS
clone WebIP ClusterIP \
    meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
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
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="true" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
crm(stonith)# cib commit stonithINFO: commited 'stonith' shadow CIB to the cluster
crm(stonith)# quit
bye
```

Appendix A. Configuration Recap

Table of Contents

A.1. Final Cluster Configuration

```
# 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="gfs2"
primitive WebSite ocf:heartbeat:apache \
    params configfile="/etc/httpd/conf/httpd.conf" \
    op monitor interval="1min"
primitive ClusterIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"
primitive ipmi-fencing stonith::fence_ipmilan \
    params pcmk_host_list="pcmk-1 pcmk-2" ipaddr=10.0.0.1 login=testuser passwd=abc123 \
    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 WebFSClone WebFS
clone WebIP ClusterIP \
    meta globally-unique="true" clone-max="2" clone-node-max="2"
clone WebSiteClone WebSite
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
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
property $id="cib-bootstrap-options" \
    dc-version="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="true" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

A.2. Node List

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

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

A.3. 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="1.1.5-bdd89e69ba545404d02445be1f3d72e6a203ba2f" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="true" \
    no-quorum-policy="ignore"
```

A.4. Resources

A.4.1. Default Options

Here we configure cluster options that apply to every resource.

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

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

A.4.2. Fencing

Note

TODO: Add text here

```
primitive ipmi-fencing stonith::fence_ipmilan \
    params pcmk_host_list="pcmk-1 pcmk-2" ipaddr=10.0.0.1 login=testuser passwd=abc123 \
    op monitor interval="60s"
clone Fencing rsa-fencing
```

A.4.3. 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 gets 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="192.168.122.101" cidr_netmask="32" clusterip_hash="sourceip" \
    op monitor interval="30s"
clone WebIP ClusterIP
    meta globally-unique="true" clone-max="2" clone-node-max="2"
```

Note

TODO: The RA should check for globally-unique=true when cloned

A.4.4. 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"
```

A.4.5. 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.

```
primitive WebFS ocf:heartbeat:Filesystem \
    params device="/dev/drbd/by-res/wwwdata" directory="/var/www/html" fstype="gfs2"
clone WebFSClone WebFS
colocation WebFS-with-gfs-control inf: WebFSClone gfs-clone
colocation fs_on_drbd inf: WebFSClone WebDataClone:Master
order WebFS-after-WebData inf: WebDataClone:promote WebFSClone:start
order start-WebFS-after-gfs-control inf: gfs-clone WebFSClone
```

A.4.6. 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.

```
primitive WebSite ocf:heartbeat:apache \
    params configfile="/etc/httpd/conf/httpd.conf" \
    op monitor interval="1min"
clone WebSiteClone WebSite
colocation WebSite-with-WebFS inf: WebSiteClone WebFSClone
colocation website-with-ip inf: WebSiteClone WebIP
order apache-after-ip inf: WebIP WebSiteClone
order WebSite-after-WebFS inf: WebFSClone WebSiteClone
```

Appendix B. Sample Corosync Configuration

Sample Corosync.conf for a two-node cluster

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

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:   6000

    # 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

    # Disable encryption
    secauth:    off

    # How many threads to use for encryption/decryption
    threads:      0

    # Optionally assign a fixed node id (integer)
    # nodeid:     1234

    interface {
        ringnumber: 0

        # The following values need to be set based on your environment
        bindnetaddr: 192.168.122.0
        mcastaddr: 226.94.1.1
        mcastport: 4000
    }
}

logging {
    debug: off
    fileline: off
    to_syslog: yes
    to_stderr: off
    syslog_facility: daemon
    timestamp: on
```

```
}

amf {
    mode: disabled
}
```

Appendix C. 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: *http://www.novell.com/documentation/sles11/book_sleha/index.html?page=/documentation/sles11/book_sleha/data/book_sleha.html*
- Corosync *http://www.corosync.org*

Appendix D. Revision History

Revision 1-1 **Mon May 17 2010** **Andrew Beekhof** *andrew@beekhof.net*

Import from Pages.app

Revision 2-1 **Wed Sep 22 2010** **Raoul Scarazzini** *rasca@miamammausalinux.org*

Italian translation

Revision 3-1 **Wed Feb 9 2011** **Andrew Beekhof** *andrew@beekhof.net*

Updated for Fedora 13

Revision 4-1 **Wed Oct 5 2011** **Andrew Beekhof** *andrew@beekhof.net*

Update the GFS2 section to use CMAN

Revision 5-1 **Fri Feb 10 2012** **Andrew Beekhof** *andrew@beekhof.net*

Generate docbook content from asciidoc sources

Index