The Zenoss Enablement Series:

Zenoss Service Dynamics

Impact and Event Management on Red Hat Cluster Suite

Configuration Guide

Document Version 424-P1

Zenoss, Inc.

www.zenoss.com

Table of Contents

Applies To

The procedure outlined in this document has been tested on the following software versions:

- Zenoss 4.2.3
- Impact version 4.2.4 build 1.2.9.7
- Centos Linux 6.4 Linux nodes
- DRDB version 8.4.3-2
- VMware ESXi 5.0 hypervisor for cluster nodes

Summary

The objective of setting up the Zenoss Service Impact in a 'high availability' cluster is to minimize, to the greatest degree possible, the downtime associated with a hardware or (non Zenoss) software failure of the server hosting Impact. High availability clusters can have various configurations, including, but not limited to:

- Active-Passive, non geo-diverse
- Active-Active non geo-diverse
- Active-Passive, geo-diverse

This document describes an Active – Passive high availability cluster without geo diversity that uses Red Hat Cluster Suite (RCHS) and Distributed Replicated Block Device (DRBD). For our scenario, two identical servers per cluster are deployed. At any given time, one node serves as the 'primary' active server and a second identical server stands by ready to take over provision of the key Zenoss services in the event the first server fails or otherwise becomes unavailable. This solution lacks geo diversity because the two servers are co-located in the same facility. As such, no protection against a scenario that destroys or renders unreachable the facility hosting Impact is provided by this solution.

This manual provides an overview and step-by-step configuration directions to deploy a highly-available Service Impact on a fast local area network with RHEL or CentOS 6.

Overview

The Service Impact service is composed of three main daemons - *zenimpactgraph*, *zenimpactserver* and *zenimpactstate*. By default, these daemons are installed on the Resource Manager master server. For large deployments however, these daemons are deployed on a remote host. This manual assumes that the Service Impact daemons are hosted on a remote host (off-master). The following figure shows a Service Impact implementation.

Understanding Node Roles

Nodes within each cluster assume one of two roles; active and standby. Each cluster consists of two nodes with only one node assuming the active role at a time. The other node serves as the standby node. The active node responds to requests from users until the service is interrupted, then the roles interchange.

Replicating Data with DRDB

Instead of using shared storage, the cluster replicates data to the passive node through DRBD. DRBD, in this case, eliminates shared storage as a single point of failure.

Using Multiple NICs

Although one network interface card (NIC) can suffice, two NICs are recommended for each member node. One NIC is used for the public network external requests and heartbeat. The second NIC is used by DRBD for replication service. This method prevents the public network from becoming saturated by the disk replication or synchronization process.

Understanding the VIP

Each cluster has a floating Virtual IP Address (VIP) on the public network. The VIP is assigned to the active node. In case of interruption, the VIP is re-assigned to the standby node.

Managing the Cluster (Remotely)

The cluster can be managed remotely through luci, a web based GUI for managing RHCS. All cluster-related configurations can be performed through the GUI. Services can be disabled or restarted through the GUI for performing maintenance tasks. The cluster manager is typically installed on its own server, and a single instance can be used to manage multiple clusters. For example, a single luci server could be used to manage separate clusters for Resource Manager, Service Impact, and Analytics.

Naming Conventions

The following naming conventions are used in this guide. Replace the example names with the names or values in your environment.

- **IMPACT** - Cluster name of the Service Impact service.
- **IMPACTVIP** - Virtual IP address of the IMPACT cluster.
- **IMPACT1** - Hostname of the primary Service Impact node.
- **IMPACT2** - Hostname of the secondary Service Impact node.
- **IMPACT{1,2}-PUBLICIP** - Public IP of the Service Impact node. It is not necessarily a public IP address. It can be any address accessible by the service user (for example, the Resource Manager).

- **IMPACT{1,2}-PRIVATEIP** - Private IP of the Service Impact node. This will be used for cluster-related communications only.
- **CLUSTERMGR** - Hostname of the luci server.

Sample commands are prepended with prompts that indicate which user issues the command. These prompts include:

- `#` (pound/hash sign) - execute the command as *root*
- `$` (dollar sign) - execute the command as *zenoss*
- `zends>` - execute the command in the *zends* console

Text in sample commands might be enclosed in less than (<) and greater than (>) symbols. This indicates the text is a placeholder and the placeholder must be replaced with an actual value for your environment. Examples of text that can include placeholders are version numbers and hostnames.

Prerequisites

The following hardware requirements must be satisfied to install and configure Zenoss Service Impact in a high availability environment:

- A working Zenoss Resource Manager server and ZenDS instance.
- One machine for the **luci server**. A single server can be used to manage multiple clusters. Ideally, this machine will have the same architecture as the node systems.
- Two identical RHEL or CentOS machines to function as **Service Impact nodes**. See the *Service Impact and Event Management* installation manual for hardware requirements.
- Two network interface cards per machine (except luci) with IP addresses configured for both public and private networks.
- At least two filesystems – one each for the OS and Zenoss data replication.
- The same architecture for all node systems. The cluster manager node and cluster nodes should have the same processor architecture (x86 or x86_64) and OS version (RHEL or CentOS 6). The cluster manager node configures and manages the clusters and creates DRBD RPM packages. As such, it should share the same architecture as the node systems.

Consider the following prior to implementing Service Impact with Red Hat Cluster Service:

- The host clocks must be synchronized to a time server via Network Time Protocol (NTP).
- SELinux must be disabled on all nodes because it is not supported by Zenoss.
- Nodes should be located in a single LAN with multicast support.
- There must be a resolvable hostname or domain name for both private and public IP addresses. If an authoritative DNS server is not available, you can add hostname entries to the `/etc/hosts` file for each node.

Configuration

The following sections describe the installation and configuration tasks that result in a working Zenoss Service Impact on Red Hat Clustering Service.

Installing luci as the Remote Cluster Manager

The cluster manager node is used to configure and manage the clusters. It is also used to create DRBD RPM packages.

Perform the following procedure to install luci as the cluster manager:

1. Update CLUSTERMGR:

   ```
   # yum update
   ```

2. Enter the following commands to ensure the CLUSTERMGR time is synchronized:

   ```
   # chkconfig ntpd on
   # ntpdate pool.ntp.org
   # /etc/init.d/ntpd start
   ```

3. For setup purposes, on all nodes, enter the following commands to disable the internal software firewall:

   ```
   # chkconfig iptables off
   # service iptables stop
   ```

   Note: After you identify the ports for cluster and Zenoss service communications, the firewall can be re-enabled with the appropriate ports opened.

4. Reboot the machine:

   ```
   # shutdown -r now
   ```

5. On CLUSTERMGR, use yum to install the luci service:

   ```
   # yum install luci
   ```

6. Start luci:

   ```
   # service luci start
   ```

7. Configure luci to start on boot:

   ```
   # chkconfig --level 12345 luci on
   ```

8. Verify that the cluster hostnames resolve from the cluster manager through DNS or the hosts table.

Compiling the DRBD Packages

Perform the following procedure to compile the DRDB packages:

1. On the *CLUSTERMGR* node, install development tools to enable compiling source code into RPM packages:

   ```
   # yum -y install gcc make automake autoconf flex rpm-build kernel-devel
   libxslt
   ```

2. Download the latest version of DRBD from http://oss.linbit.com/drbd/:

   ```
   # cd
   # wget http://oss.linbit.com/drbd/<version>/drbd-<version>.tar.gz
   ```

3. Create target directories for the RPM packages:

   ```
   # mkdir -p rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}
   ```

4. Compile the code with the --*rgmanager* switch:

   ```
   # tar xvfz drbd-<version>.tar.gz
   # cd drbd-<version>
   # ./configure --with-rgmanager
   # make rpm && make km-rpm
   ```

   Note: The rgmanager switch compiles the resource scripts and other utilities to use DRBD on RHCS.

5. Inspect the `~/rpmbuild/RPMS` directory and verify that the following RPMs were created:

   - drbd-pacemaker
   - drbd-utils
   - drbd-km
   - drbd-rgmanager

Because some versions of the source code do not include `drbd-rgmanager` in the 'make rpm' process, it might be necessary to use the `rpmbuild` command to build it manually, for example:

```
# rpmbuild --bb drbd.spec --with rgmanager
```

Preparing the Service Impact Nodes

The section describes the required procedures for preparation of the Service Impact nodes. The information in this section has the following caveats:

- Complete all steps in this section on both nodes unless specifically directed.
- Prepare two identical machines for the Zenoss Service Impact nodes.

Creating the LVM Disks

The following procedure describes the creation of LVM disks for each node. The commands in the procedure assume:

- The disk dedicated to the LVM volume is located at `/dev/sdb`.
- The volume group is `zenoss_data`.

- The logical volume is `lv_impact.`

Perform the following procedure to create LVM disks on two nodes:

1. For setup purposes, enter the following commands on all nodes to disable the internal software firewall:

   ```
   # chkconfig iptables off
   # service iptables stop
   ```

   Note: After you identify the ports for cluster and Zenoss service communications, the firewall can be re-enabled with the appropriate ports opened.

2. Disable SELinux. Because SELinux is not compatible with Zenoss, you must disable it. Enter the following commands on both *Node 1* and *Node 2*:

   ```
   # sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
   ```

3. Update the nodes:

   ```
   # yum update
   ```

4. Enter the following commands on both nodes to ensure that their times are synchronized:

   ```
   # chkconfig ntpd on
   # ntpdate pool.ntp.org
   # /etc/init.d/ntpd start
   ```

5. Reboot each machine:

   ```
   # shutdown -r now
   ```

6. Issue the following command to create a partition on `/dev/sdb` using the `fdisk` utility:

   ```
   # fdisk /dev/sdb
   ```

7. Create the disk partition on */dev/sdb* and tag the disk as **LVM partition (8e)**. Use the following sequence to create the first LVM partition from the first to last block:

   ```
   n,p,1,<enter>,<enter>,t,8e,w
   ```

8. Create the `zenoss_data` volume group and `lv_impact` logical disk:

   ```
   # pvcreate /dev/sdb1
   # vgcreate zenoss_data /dev/sdb1
   # lvcreate -L <size> -n lv_impact zenoss_data
   ```

Creating the DRDB Resource

Perform the following procedure to create the DRDB Resource for the nodes:

1. Copy the following rpm files from the cluster manager to the IM nodes:
   - drbd-pacemaker-< *version* >.rpm
   - drbd-utils-< *version* >.rpm
   - drbd-km-< *version* >.rpm
   - drbd-rgmanager-< *version* >.rpm

2. Install the packages and their dependencies:

   ```
   # rpm -Uhv drbd-utils-<version>.rpm
   # rpm -Uhv drbd-km-<version>.rpm
   # rpm -Uhv drbd-pacemaker-<version>.rpm
   # rpm -Uhv drbd-rgmanager-<version>.rpm
   ```

3. Copy the following DRBD configuration into the */etc/drbd.d/global_common.conf* file:

```
global {
    usage-count no;
}
common {
    handlers {
       pri-on-incon-degr "/usr/lib/drbd/notify-pri-on-incon-degr.sh;
/usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger
; reboot -f";

       pri-lost-after-sb "/usr/lib/drbd/notify-pri-lost-after-sb.sh;
/usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger
; reboot -f";

       local-io-error "/usr/lib/drbd/notify-io-error.sh;
/usr/lib/drbd/notify-emergency-shutdown.sh; echo o > /proc/sysrq-
trigger ; halt -f";

       fence-peer "/usr/lib/drbd/crm-fence-peer.sh"; after-resync-
target "/usr/lib/drbd/crm-unfence-peer.sh";
   }

   disk {
      on-io-error detach;
      fencing resource-only;
      resync-rate 300M;
   }
}
```

4. On both nodes, create the */etc/drbd.d/r0.res* file.
   Note: The hostname must be consistent with the output of `uname -n` on all nodes. If it is not, you will encounter the following error:

   "r0 not defined in your config (for this host)”

Use the following configuration and replace *IMPACT{1,2}* and *IMPACT{1,2}-PRIVATEIP* with your hostnames and ip addresses:

```
resource r0 {
   volume 0 {
        device /dev/drbd0;
        disk /dev/zenoss_data/lv_impact;
        flexible-meta-disk internal;
   }

   net {

       use-rle;
   }

on IMPACT1 {
address IMPACT1-PRIVATEIP:7788;
}

   on IMPACT2 {
      address IMPACT2-PRIVATEIP:7788;
   }
}
```

5. Create the resource r0:

   ```
   # drbdadm create-md r0
   ```

Note: Before starting the DRBD service in the next step, verify that the resource on the other node is configured.

6. Start the DRBD service:

   ```
   # service drbd start
   # chkconfig --level 12345 drbd on
   ```

7. Execute the following command on *IMPACT1* to set is as *primary:*

   ```
   # drbdsetup /dev/drbd0 primary --force
   ```

   Allow the nodes enough time to synchronize their disks. Check the status by running the following command:

   ```
   # drbd-overview
   ```

   Continue to the next step when you see the following output from the 'drbd-overview' command on the primary node:

   Connected Primary/Secondary UpToDate/UpToDate C

8. Initialize the DRDB file system. On *IMPACT1*, run the following command:

   ```
   # mkfs -t ext4 /dev/drbd0
   ```

   Note: It is not necessary to perform this initialization on *IMPACT2* because the action is replicated automatically on *IMPACT2*.

9. On both nodes, create the */opt/zenoss/var/impact* directory:

   ```
   # mkdir -p /opt/zenoss/var/impact
   ```

10. Mount the DRBD disk on *IMPACT1*:

    ```
    # mount /dev/drbd0 /opt/zenoss/var/impact
    ```

    Note: It is not necessary to add an entry for this drive to */etc/fstab* because mounting is managed by the RHCS rgmanager.

Installing the Dependencies on the Service Impact Nodes

Perform the following procedure before configuring the Service Impact nodes:

1. On both nodes, confirm that OpenJDK is not installed. If it is installed, remove it before proceeding.
2. Install Java SE Runtime Environment version 6:

   a. Download the self-installing RPM of Oracle Java SE Runtime Environment 6u31 from the Java SE 6 Downloads page. The file to download is `jre-6u31-linux-x64-rpm.bin`.

   b. Make the RPM installer executable:

   ```
   # chmod +x /path-to-installer/jre-6u31-linux-x64-rpm.bin
   ```

   c. Start the installer:

   ```
   # /path-to-installer/jre-6u31-linux-x64-rpm.bin
   ```

   d. Add the following line to the end of the */etc/profile* file to append the JAVA_HOME environment variable to it:

   ```
   export JAVA_HOME=/usr/java/default
   ```

   e. Verify the installed version is correct (1.6 Update 31):

   ```
   # java -version
   ```

3. Install the Zenoss dependencies RPM:

   ```
   # rpm -ivh http://deps.zenoss.com/yum/zenossdeps-4.2.x-1.el6.noarch.rpm
   ```

4. Install ZenDS and its dependencies:

   ```
   # yum -y install perl-DBI dmidecode
   # yum --nogpgcheck localinstall zends-<version>.rpm
   ```

5. Verify name resolution. If the nodes cannot resolve the fully qualified domain name of the master via DNS, update the */etc/hosts* file on the master and both impact nodes.

   - Append this line to the file on the master:

     ```
     127.0.0.1 [zenoss master FQDN]
     ```

   - Append this line to the files on both impact nodes:

     ```
     [master's IP] [zenoss master FQDN]
     ```

6. Install the Zenoss collector dependencies so the remote collector deployment does not return an error:

   ```
   # yum -y install rrdtool-1.4.7 net-snmp net-snmp-utils libxslt rsync
   ```

7. On the Zenoss Resource Manager server, generate SSH keys that enable the user *zenoss* to login to the

Impact nodes without passwords:

```
# su - zenoss
$ ssh-keygen
$ cat .ssh/id_rsa.pub
```

Note: If you have existing remote hub(s) or collector(s), you might have already done this.

When the keys are created, copy the contents of the public key.

8. On the Impact nodes, create the .ssh directory and within it the authorized_keys file as follows if they do not exist.

   Ensure the *.ssh* directory is mode **700** and *authorized_keys* is mode **640**.

   - Run the following commands as *root* on the IM nodes:

     ```
     # mkdir -p /root/.ssh/
     # touch /root/.ssh/authorized_keys
     # chmod 700 /root/.ssh/
     # chmod 640 /root/.ssh/authorized_keys
     ```

   - Run the following commands as *zenoss* on the IM nodes:

     ```
     # su - zenoss
     $ mkdir -p /home/zenoss/.ssh
     $ chmod 700 /home/zenoss/.ssh
     $ touch /home/zenoss/.ssh/authorized_keys
     $ chmod 640 /home/zenoss/.ssh/authorized_keys
     ```

   When the *authorized_keys* files are created, append the public key into the files.

9. On the Zenoss Resource Manager server, test login access without a password:

   - As the *Zenoss* user, SSH in to the IM nodes as *zenoss.* For example:

     ```
      $ ssh -i [path to key] [node hostname]
     ```

   - SSH into the nodes as the *root* user, using the same key.

   If the test is successful you can login to both nodes as *root* and *zenoss* without entering a password.

   Note: Testing SSH access is important because the remote collector deployment process can fail if it encounters a host key check warning. The host key check occurs when logging in the first time.

10. As *root*, change ownership for the /opt/zenoss/ directory:

    ```
    # chown -R zenoss:zenoss /opt/zenoss/
    ```

Installing the Service Impact ZenPack on the Resource Manager server

Perform the following procedure to install the Service Impact ZenPack:

1. Download the latest Impact ZenPack from the Zenoss support site, http://support.zenoss.com.

2. Install the Service Impact Zenpack as the *zenoss* user:

   ```
   $ zenpack --install ZenPacks.zenoss.Impact-<version>.egg
   ```

3. Restart *zeneventd* and *zenwebserver*:

   ```
   $ zeneventd restart && zenwebserver restart
   ```

4. Stop the Service Impact daemons:

   ```
   $ zenimpactgraph stop
   $ zenimpactserver stop
   $ zenimpactstate stop
   ```

5. Remove the Service Impact daemons from *$ZENHOME/etc/daemons.txt.* If the file is empty, issue the following commands, within the same folder:

   ```
   $ zenoss list > daemons.txt
   $ touch DAEMONS_TXT_ONLY
   ```

   Note: This prevents the Impact daemons from running on the master.

6. In *$ZENHOME/etc/global.conf* file set the **dsahost** value to the IM node's cluster VIP and replace any references to *localhost* or *127.0.0.1* with the domain name of the *master* (or *zenoss-master* if it didn't have one).

7. Restart zenwebserver:

   ```
   $ zenwebserver restart
   ```

8. Perform the following steps to point *dsa.amqp.uri* to the master:

   a. Navigate to the file:
   `$ZENHOME/ZenPacks/ZenPacks.zenoss.Impact-<version>.egg/ZenPacks/zenoss/Impact/etc/zenoss-dsa.properties`

   b. Set the value of **dsa.amqp.uri** to point to the master, or the server running the RabbitMQ, instead of *localhost*.

   Note: Create a soft link to easily locate this settings file.

Configuring the Primary Service Impact Node (IMPACT1)

Perform the following procedure to configure the primary Service Impact node:

1. On the Zends server (or master if Zends is hosted on the master) become the *zenoss* user:

   ```
   # su - zenoss
   ```

2. Login to zends to grant IM nodes access to the database:

   ```
   $ zends -u root
   ```

3. Issue the following zends commands:

   ```
   zends> grant all on *.* to 'root'@'IMPACT1-PUBLICIP' with grant option;
   zends> grant all on *.* to 'root'@'IMPACT2-PUBLICIP' with grant option;
   zends> flush privileges;
   ```

4. In the Zenoss graphical interface, create a remote collector under the *localhost* hub using **IMPACT1** as a

target. Wait for the process to complete.

Note: This step might require some time to complete because it prepares the Zenoss environment on the Service Impact node - it creates a *zenoss* user, $ZENHOME, copies files over, etc.

5. Stop all zenoss daemons on *IMPACT1:*

```
# service zenoss stop
# chkconfig zenoss off
```

6. Edit `$ZENHOME/etc/daemons.txt` to include only the IM daemons, for example:

```
zenimpactgraph
zenimpactserver
zenimpactstate
```

7. Ensure the empty *$ZENHOME/etc/daemons/DAEMONS_TXT_ONLY* file exists. This acts as a trigger for Zenoss to run only the daemons listed in the *$ZENHOME/etc/daemons.txt* file.
8. Copy the *$ZENHOME/etc/zenimpactgraph.conf* file and/or the *$ZENHOME/etc/zenimpactstate.conf* file from the Zenoss Resource Manager server to prevent warnings about missing configuration files.
9. Ensure the DRBD disk is still mounted at *$ZENHOME/var/impact*.
10. Copy *$ZENHOME/var/impact* from the **Zenoss Resource Manager server** to **IMPACT1** using `scp` or `rsync`.
11. Navigate to *$ZENHOME* on the Service Impact node.
12. As *root,* change ownership of */var/impact*. Run the following command:

```
# chown -R zenoss:zenoss var/impact
```

13. On The Zenoss Resource Manager server, remove or rename the *var/impact* directory.
Optionally, you can move the backups file(s) *$ZENHOME/var/impact-*.tar.gz* to **IMPACT1** .

Note: Do not start the daemons (zenoss service) on IMPACT1 yet.

Configuring the Secondary Service Impact Node (IMPACT2)

Perform the following procedure to configure the secondary Service Impact node:

1. In the Zenoss user interface, create a remote collector under the *localhost* hub using **IMPACT2** as a target. Wait for the process to complete.

   Note: This step may take some time because it prepares the Zenoss environment on the IM node. It creates a *zenoss* user, `$ZENHOME`, copies files over, etc.

2. Stop the zenoss daemons on *IMPACT2*:

```
# service zenoss stop
# chkconfig zenoss off
```

3. Edit `$ZENHOME/etc/daemons.txt` to include only the IM daemons, for example:

```
zenimpactgraph
zenimpactserver
zenimpactstate
```

4. Ensure the empty *$ZENHOME/etc/daemons/DAEMONS_TXT_ONLY* file exists. It acts as a trigger for Zenoss to run only the daemons listed in the *$ZENHOME/etc/daemons.txt* file*.*
5. Copy the *$ZENHOME/etc/zenimpactgraph.conf* and/or *$ZENHOME/etc/zenimpactstate.conf* files from the Zenoss Resource Manager server to prevent warnings about missing configuration files.

Preparing the Service Impact Cluster

Perform the following procedure to prepare the cluster:

1. On all IMPACT nodes, install *rgmanager*, *ricci*, and *cman:*

```
# yum -y install rgmanager ricci cman
```

2. Set a password for the user *ricci*:

```
# passwd ricci
```

   Note: This password is used by the cluster manager to access to nodes.

3. Configure *ricci* , *cman*, *rgmanager* and *modclusterd* to start on boot:

```
# chkconfig --level 12345 ricci on
# chkconfig --level 12345 cman on
# chkconfig --level 12345 rgmanager on
# chkconfig --level 12345 modclusterd on
```

4. Start ricci on all IMPACT nodes:

```
# service ricci start
```

5. Browse to **`https://CLUSTERMGR:8084`** and login to luci as *root.*

Configuring the Cluster

Perform the following procedure to configure the cluster:

1. Under *Manage Clusters*, click **Create**.
2. Enter **IMPACT** for the *cluster name*, in this example. Replace *IMPACT* with your cluster name.
3. Enter the *node names*, their *ricci ports*, *private IP address*, *users* and *passwords*.
4. Ensure the node name is a resolvable hostname and resolves to its private IP address.
5. Leave other fields as default.
6. Click **Create Cluster**.
7. If you need to modify a node attribute, click the **Nodes** tab.

Creating a Failover Domain

Perform the following steps to create a failover domain.

1. Under the *IMPACT* cluster, click the **Failover Domains** tab.
2. Click **Add**.
3. In the *name* field, enter **impact_domain**.
4. Check **Prioritized**.
5. Check the member nodes **(IMPACT{1,2})**.
6. Set the priority of *IMPACT1* to **1** and *IMPACT2* to **2**.

   Note: This means *IMPACT1* has the higher priority of the two.
7. Click **Create.**

Creating the Resources

Perform the following steps to create the required resources:

Adding a DRBD Resource

1. Still under the *IMPACT* cluster, click the **Resources** tab.
2. Click **Add** to create the DRBD resource.
3. Select **DRBD Resource** as the *resource* type.
4. Enter **impact_drbd** as the *name* and **r0** as the **DRBD Resource** name.

   Note: *r0* is the resource name you created in the node preparation for this example.
5. Click **Submit**.

Creating the Zenoss File System

1. Click **Add** to create the *filesystem* resource.
2. Select **filesystem** as the *resource* type.
3. Enter **impact_dir** as the *name*.
4. Select **ext4** as the *filesystem* type (or the file system you chose if different).
5. Enter **/opt/zenoss/var/impact** as the *mount point*.
6. Enter **/dev/drbd0** as the *device*.
7. Keep the defaults for mount options.
8. Click **Submit**.

Creating Script Resources

1. Click **Add** to create the *init script resource*.
2. Select **Script** as the *resource* type.
3. Enter **impact_init** as the *name*.

4. Enter **/etc/init.d/zenoss** as the full path to script file.
5. Click **Submit**.

Creating the IP Address Resource

1. Click **Add** to create the *IP Address* resource.
2. Select **IP Address** as the *resource* type.
3. Enter **IMPACTVIP** as the *IP address* for this example. Replace *IMPACTVIP* with your VIP address.
4. Ensure that **monitor link** is checked.
5. Click **Submit**.

Creating the Service Groups

Perform the following steps to create the service groups:

1. Under the *IMPACT* cluster, click the **Service Groups** tab.
2. Click **Add**.
3. Enter **impact** as the *service* name.
4. Check **Automatically Start this Service**.
5. Select **IMPACT_domain** as the *failover* domain.
6. Select **Relocate** as the *recovery policy.*

   Note: This means the service will relocate to the standby server on failure.
7. Click **Add Resource**.
8. Select **impact_drbd**.
9. At the bottom of *impact_drbd*, click **Add Child Resource**.
10. Select **impact_dir**.
11. At the bottom of *impact_dir*, click **Add Child Resource**.
12. Select **impact_init**.
13. At the bottom of *impact_init*, click **Add Child Resource**.
14. Select **IMPACTVIP**.
15. Click **Submit**.

Fencing (optional)

Optionally, configure fencing. See *Appendix B – Fencing in a VMware Environment* for instructions on fencing.

Administration

Clusters can be managed through luci as a cluster manager. The luci server provides a graphical user interface to stop, start, and relocate services. It also enables users to configure the cluster.

Note: Although it is also possible to manage the cluster through the command-line, that discussion is beyond the scope of this document. Consult the RHCS documentation for information about how to manage clusters through the command-line.

Executing Maintenance Tasks on the Cluster

Before performing maintenance tasks on the cluster, ensure the service group is disabled. This is necessary to avoid automatic toggling between nodes. If you must start the service/daemon while performing maintenance, you must start a node manually. Perform the following steps to manually start the daemon:

1. Disable the service group.
2. Choose a node to run the service.
3. Set the *DRBD resource* of that node to **primary**.
4. Mount the disk.
5. Assign the VIP to that node using the *ip* command. For example:

   ```
   ip addr add >IMPACTVIP< /24 dev eth1
   ```
6. Start the service.

Appendix A – Known Errors

Upon an attempt to start, `cman` reports:

```
Can't determine address family of nodename.
```

To troubleshoot, ensure that the node name of the node is resolvable from itself. Add the hostname into the `/etc/hosts` file if you are not using DNS.

Appendix B – Fencing in a VMware Environment

Fencing is an automated means of isolating a node that appears to be malfunctioning to protect the integrity of the DRBD volume(s).

Perform the following steps to configure the ESXi fencing:

1. Under the *IMPACT* cluster, click the **Fence Devices** Tab.
2. Click **Add**.
3. Select the **VMware Fencing (SOAP Interface**).
4. Enter the **ESXi** server name as the *name*.
5. Enter the **hostname** or **IP address**.
6. Enter the **Login** and **Password**.
7. Repeat steps 1-6 for each ESXi host.
8. Navigate to the **Nodes** tab.
9. Select a node.
10. Click the **Add Fence Method** button.
11. Enter **off** as the *method* name.
12. Click **Submit**.

    Note: This turns off the VM, if it was fenced.
13. Click **Add Fence Device**.
14. Select **esxi host** for that device.
15. Enter the **VM name** of the node and its **UUID**.

    Note: These values can be determined from the ESXi management user interface.
16. Check **Use SSL** if the Esxi host requires an SSL connection.
17. Navigate to the **Nodes** tab.
18. Check **all nodes**.
19. Click **Join Cluster**.

If this process is successful, the service group can be started through the graphical user interface and the Impact service can be accessed through its VIP address.
In the Zenoss graphical interface, select the **Services** tab to verify that Service Impact is working.