

## The Zenoss Enablement Series:

---

Zenoss Service Dynamics Analytics on Red Hat Cluster Suite  
Configuration Guide

Document Version 424-P1

Zenoss, Inc.

[www.zenoss.com](http://www.zenoss.com)

Copyright © 2013 Zenoss, Inc., 275 West St., Suite 204, Annapolis, MD 21401, U.S.A. All rights reserved.

Zenoss and the Zenoss logo are trademarks or registered trademarks of Zenoss, Inc. in the United States and other countries. All other trademarks, logos, and service marks are the property of Zenoss or other third parties. Use of these marks is prohibited without the express written consent of Zenoss, Inc. or the third-party owner.

DRBD® and the DRBD® logo are trademarks or registered trademarks of LINBIT® in Austria, the United States and other countries.

Oracle, the Oracle logo, Java, and MySQL are registered trademarks of the Oracle Corporation and/or its affiliates.

Linux is a registered trademark of Linus Torvalds.

RabbitMQ is a trademark of VMware, Inc.

vSphere is a trademark of VMware, Inc. in the United States and/or other jurisdictions.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries. All other companies and products mentioned are trademarks and property of their respective owners.

# Table of Contents

<b>Applies To</b> .....	<b>1</b>
<b>Summary</b> .....	<b>1</b>
<b>Overview</b> .....	<b>2</b>
<b>Naming Conventions</b> .....	<b>2</b>
<b>Prerequisites</b> .....	<b>3</b>
<b>Configuration</b> .....	<b>5</b>
<b>Installing luci as the Remote Cluster Manager</b> .....	<b>5</b>
<b>Compiling the DRBD Packages</b> .....	<b>6</b>
<b>Preparing the Analytics DB Nodes</b> .....	<b>6</b>
Creating the LVM Disks.....	7
Creating the DRDB Resource.....	7
<b>Installing the Analytics DB</b> .....	<b>9</b>
<b>Setting up the Analytics DB Cluster</b> .....	<b>10</b>
Preparing the Analytics DB Cluster for Configuration.....	10
Creating the Analytics DB Cluster.....	11
Creating a Failover Domain.....	11
Creating Resources.....	11
Creating Service Groups.....	12
<b>Preparing the Analytics Nodes</b> .....	<b>13</b>
Creating the LVM Disks.....	13
Creating the DRDB Resource.....	13
Installing Analytics Dependencies on the Analytics Nodes.....	16
Installing Analytics on ANALYTICS1.....	17
Installing Analytics on ANALYTICS2.....	18
Testing Analytics Manual Failover/Failback.....	18
Preparing the Analytics Cluster for Configuration.....	19
Configuring the Analytics Cluster.....	19
Creating the Analytics Failover Domain.....	20
Creating the DRDB Resources.....	20
Creating the Service Groups.....	21
<b>Failure Modes</b> .....	<b>22</b>
<b>Node Failure</b> .....	<b>22</b>
Node Failure Test Case.....	22
<b>Zenoss Analytics or ZenDS Failure</b> .....	<b>22</b>
Analytics or ZenDS Failure Test Case.....	22
<b>Administration</b> .....	<b>22</b>



## Applies To

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

- Analytics version 4.2.4 build 1897
- Centos Linux 6.4
- 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 Zenoss 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 Zenoss. High availability clusters can have various configurations, including, but not necessarily 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 (RHCS) and a Distributed Replicated Block Device (DRBD). For our scenario four identical servers are deployed.

At any given time, two nodes serve as the 'primary' active servers and two identical servers stand by ready to take over the provision of the key Zenoss services in the event the primary servers fail or otherwise become unavailable. This solution is termed *lacking geo diversity* because both nodes in any given cluster are co-located in the same facility. As such, no protection against a scenario that destroys or renders unreachable the facility hosting Zenoss is provided by this solution.

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

## Overview

The Analytics service is composed of the Analytics application and database servers. For large deployments, these services are usually located on different hosts. To minimize single point of failure, two clusters are recommended for Analytics – *ANALYTICS* and *ANALYTICSDB*.

### 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 a DRBD. The DRBD, in this case, eliminates shared storage as a single point of failure.

### 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.

### Understanding the Deployed Networks

For this scenario, two networks are deployed. The first is a public network used for external requests, while the other is used by the replication, heartbeat and clustering service. This prevents the public network from becoming saturated by replication and heartbeat communications.

### 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 can be installed on a separate machine from the Resource Manager. A single *luci* 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

For the purpose of generalizing the document, the following naming conventions are used. Replace the example names with the names or values in your environment.

- **ANALYTICS** - cluster name of the Analytics application service.
- **ANALYTICSDB** - cluster name of the Analytics DB (zends).
- **ANALYTICSVIP** - virtual IP address of the ANALYTICS cluster.
- **ANALYTICS1** - hostname of the primary Analytics node.
- **ANALYTICS2** - hostname of the secondary Analytics node.
- **ANALYTICSDB1** - hostname of the primary Analytics DB node.
- **ANALYTICSDB2** - hostname of the secondary Analytics DB node.

- **ANALYTICS{1,2}-PUBLICIP**- public IP of the Analytics node. Although not necessarily a public IP address, it must be reachable by users.
- **ANALYTICS{1,2}-PRIVATEIP**- private IP of the Analytics node.
- **ANALYTICSDBVIP** - virtual IP address of the ANALYTICSDB cluster.
- **DB1-PUBLICIP** - public IP of the primary Analytics DB node. Although not necessarily a public IP address, it must be reachable by users, for example, the Analytics application server.
- **DB2-PRIVATEIP** - private IP of the primary Analytics DB node.
- **CLUSTERMGR**- hostname of the luci server.

## Prerequisites

The following hardware requirements must be satisfied to successfully install and configure an Active-Passive HA cluster to host Zenoss Analytics:

- Two pairs of identical, fully dedicated servers for Zenoss Analytics cluster nodes.
- Two network interface cards per machine, (*eth0* and *eth1*). The second NIC card, *eth1*, is dedicated to DRBD and cluster traffic.
- DRDB for replication of key file system directories across the two cluster nodes.
- At least two disks per machine for `/dev/sda` and `/dev/sdb`, with the `sdb` hard drive reserved solely for Zenoss data.
- 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.
- A cluster virtual IP address (VIP). The VIP can be claimed by either of the two nodes when required to move access to Zenoss from one node to the other. The VIP is set up as a second IP address for one of the NIC cards residing on one of the nodes (at any given time).
- Red Hat Cluster Suite for cluster resource manager software to manage file system mounting, VIP assignment, and startup of key services on the backup node should the primary node fail (or a migration of services is required).
- Heartbeat monitoring of critical services running on the primary node, so that a migration from the primary to the backup can be triggered when needed.

Consider the following prior to implementing Zenoss Analytics with Red Hat Cluster Suite:

- The host clocks must be synchronized to a time server via Network Time Protocol (NTP).
- SELinux must be disabled on all hosts because it is not supported by Zenoss.
- The nodes should be located within a single LAN with multicast support.
- The nodes must have hostname(s) that resolve to their private IP address. (for example: `resourcemgr1.private.domain.com`)
- There must be a resolvable hostname or domain name for both private and public IP addresses.

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*.

For the purposes of this document, an example scenario is referred to including sample values for items such as IP addresses, hostnames, and subnet masks. Administrators using this guide to deploy Zenoss in a high availability configuration must substitute actual values where applicable.

These instructions assume that Centos 6 is installed as the operating system and that the 'Basic Server' install option was chosen. The commands listed are run as the root user unless otherwise specified. We also assume that the Linux servers hosting Zenoss have access to the standard Yum repositories -meaning they have internet access - and are fully patched.

# Configuration

The following sections describe the installation and configuration tasks that result in a working Zenoss Service Dynamics Analytics (Analytics) on Red Hat Cluster Suite. Complete the following preparatory steps on all cluster nodes and the cluster management system:

Perform the following procedure to install NTP:

1. Determine whether NTP is installed or not. As root, issue the following command on all nodes:

```
# rpm -qa ntp
```

Consult the result of the command to determine whether ntp is installed or not.

2. Install NTP (if required). As *root*, issue the following command:

```
# yum install ntp
```

3. Start ntp and configure it to start on system boot. As *root*, issue the following commands:

```
# chkconfig ntpd on
```

```
# ntpdate pool.ntp.org
```

```
# /etc/init.d/ntpd start
```

4. For setup purposes, enter the following commands to disable the internal software firewall:

```
# chkconfig iptables off
```

```
# service iptables stop
```

**Note:** After identifying the ports for cluster and Zenoss service communication, the firewall can be re-enabled with the appropriate ports opened.

5. Because SELinux is not compatible with Zenoss, enter the following commands to disable SELinux:

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

6. Fully update all nodes:

```
# yum -y update
```

7. Reboot all nodes.

## 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. On CLUSTERMGR, install luci using yum:

```
# yum install luci
```

2. Start luci:

```
# service luci start
```

3. Configure luci to start on boot:

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

4. Verify that the cluster hostnames can be resolved from the cluster manager through DNS or the hosts table.

## Compiling the DRBD Packages

Perform the following procedure to compile the DRBD 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 the 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 build it manually, for example:

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

## Preparing the Analytics DB Nodes

This section describes the procedure required to prepare the ZenDS nodes. The information in this section has the following caveats:

- Complete all steps in this section and subsections for all nodes unless specifically directed.
- Prepare two identical machines for the database software.
- Each machine must have at least two disks or partitions for the operating system and Analytics files.
- Ensure that public and private hostnames of member nodes are mutually resolvable.

## Creating the LVM Disks

The following procedure describes how to create the 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_zends`.

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

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

```
# fdisk /dev/sdb
```
2. 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
```
3. Create the `zenoss_data` volume group and `lv_zends` logical disk:  

```
# pvcreate /dev/sdb1
# vgcreate zenoss_data /dev/sdb1
# lvcreate -L <size> -n lv_zends 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 database 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:** In the following configuration, replace `ANALYTICSDB{1,2}` and `DB{1,2}-PRIVATEIP` with the names and IP addresses used in your environment.

**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 information for the configuration:

```

resource r0 {
    volume 0 {
        device /dev/drbd0;
        disk /dev/zenoss_data/lv_zends;
        flexible-meta-disk internal;
    }
    net {
        use-rle;
    }
    on ANALYTICSDB1 {
        address DB1-PRIVATEIP:7788;
    }
    on ANALYTICSDB2 {
        address DB2-PRIVATEIP:7788;
    }
}

```

```
}
}
```

5. Create the resource r0:

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

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

6. Start the DRBD service:

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

7. Set ANALYTICSDB1 as primary, execute the following command on ANALYTICSDB1:

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

8. Give the nodes enough time to synchronize their disks. Check the status with 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
```

9. Initialize the DRBD filesystem. On ANALYTICSDB1:

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

**Note:** It is not necessary to perform this initialization on ANALYTICSDB2 because this action is automatically replicated on ANALYTICSDB2.

10. On both nodes, create the /opt/zends/data directory:

```
# mkdir -p /opt/zends/data
```

**Note:** The directory /opt/zends/data is explicitly specified instead of the /opt/zends directory because the RHCS rgmanager requires the /opt/zends directory to be mounted on both nodes to check the status of the zends application.

11. Mount the DRBD disk on ANALYTICSDB1:

```
# mount /dev/drbd0 /opt/zends/data
```

**Note:** It is not necessary to add the disk to the /etc/fstab file because mounting is managed by the RHCS rgmanager.

## Installing the Analytics DB

Perform the following procedure to install the Analytics DB:

1. Download the latest ZenDS RPM package into the nodes.  
Refer to the *Zenoss Service Dynamics Analytics and Optimization Installation and Administration* guide for the ZenDS installation procedure.  
Do not start the zends service yet.
2. Delete the contents of the /opt/zends/data/ directory on ANALYTICSDB2.  
On ANALYTICSDB2, issue the following command:

```
# rm -rf /opt/zends/data/*
```

3. Do **not** start ZenDS on ANALYTICSDB2. Start ZenDS on ANALYTICSDB1:

```
# service zends start
```

**Note:** It is not necessary to start ZenDS on boot because the RHCS rgmanager will start the service.

4. Create the necessary privileges for the Analytics application and give all Analytics nodes in the cluster access. In the example below, *ANALYTICS1-PUBLICIP* and *ANALYTICS2-PUBLICIP* are the public IPs of the Analytics nodes.

- a. Become *zenoss*:

```
# su - zenoss
```

- b. Start *zends*:

```
$ zends -u root
```

- c. Issue the following *zends* commands:

```
zends> grant all on *.* to 'root'@'ANALYTICS1-PUBLICIP' with grant option;
zends> grant all on *.* to 'reporting'@'ANALYTICS1-PUBLICIP' identified by
'reporting';
zends> grant all on *.* to 'root'@'ANALYTICS2-PUBLICIP' with grant option;
zends> grant all on *.* to 'reporting'@'ANALYTICS2-PUBLICIP' identified by
'reporting';
zends> flush privileges;
```

**Note:** Reporting privileges are based on values set in the `/etc/zenoss_analytics/zenoss_analytics.conf` file.

## Setting up the Analytics DB Cluster

The following sections describe creating and configuring the Analytics DB cluster.

### Preparing the Analytics DB Cluster for Configuration

Perform the following procedure to prepare the cluster for configuration:

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

```
# yum install rgmanager ricci cman
```

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

```
# passwd ricci
```

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 ANALYTICSDB nodes:  

```
# service ricci start
```
5. Browse to <https://CLUSTERMGR:8084>.
6. Login to *luci* as **root**.

## Creating the Analytics DB Cluster

Perform the following procedure within the *luci* GUI to create the cluster:

1. Under *Manage Clusters*, click **Create**.
2. Enter **ANALYTICSDB** for the cluster *name*, in this example. Replace **ANALYTICSDB** 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 node attribute, click the **Nodes** tab.

## Creating a Failover Domain

Perform the following procedure to create a failover domain:

1. Under the *ANALYTICS* cluster, click the **Failover Domains** tab.
2. Click **Add**.
3. Enter **ANALYTICS\_domain** in the *name* field, using your cluster name.
4. Check **Prioritized**.
5. Check the member nodes, in this example **ANALYTICS{1,2}**.
6. Set the priority of *ANALYTICS1* to **1** and *ANALYTICS2* to **2**. This means *ANALYTICS1* has higher priority of the two.
7. Click **Create**.

## Creating Resources

Perform the following procedure to create the required resource for the Analytics DB cluster:

1. Under the ANALYTICSDB cluster, the click **Resources** tab.
2. Click **Add** to create a DRBD resource.
3. Select **DRBD Resource** as *resource type*.
4. Enter *zends\_drbd* as the name and *r0* as the DRBD resource name.
5. Click **Apply**.
6. Click **Add** to create filesystem resource.
7. Select **filesystem** as *resource type*.
8. Enter *zends\_dir* as the resource name.

9. Click **Apply**.
10. Select **ext4** as the *filesystem* type.
11. Enter `/opt/zends/data` as the mount point.
12. Enter `/dev/drbd0` as device and defaults as mount options.
13. Click **Submit**.
14. Click **Add** to create an *init script* resource.
15. Select **Script** as *resource* type.
16. Enter `zends_init` as the *resource* name.
17. Enter `/etc/init.d/zends` as the *full path* to the script file.
18. Click **Submit**.
19. Click **Add** to create an IP Address resource.
20. Select **IP Address** as *resource* type.
21. Enter the *IP address*, in this example, `ANALYTICSDBVIP`. Replace `ANALYTICSDBVIP` with your value.
22. Ensure the **monitor** link is checked.
23. Click **Submit**.

## Creating Service Groups

Perform the following procedure to create the service groups for the Analytics DB cluster:

1. Under the ANALYTICSDB cluster, click the **Service Groups** tab.
2. Click **Add**.
3. Enter `analyticsdb` for the *service* name.
4. Check **Automatically Start This Service**.
5. Select **ANALYTICSDB\_domain** for the *failover domain*.
6. Select **Relocate** as the *recovery* policy.
7. Click **Add Resource**.
8. Select `zends_drbd`.
9. At the bottom of `zends_drbd`, click **Add Child Resource**.
10. Select `zends_dir`.
11. At the bottom of `zends_dir`, click **Add Child Resource**.
12. Select `zends_init`.
13. At the bottom of `zends_init`, click **Add Child Resource**.
14. Select **ANALYTICSDBVIP**.
15. Click **Submit**.

When this procedure successfully completes, the cluster can be started and accessed through its VIP address.

## Preparing the Analytics Nodes

The following sections describe setting up and configuring the Analytics nodes.

### Creating the LVM Disks

The following procedure describes how to create the 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_analytics`.
1. Issue the following command to create a partition on `/dev/sdb` using the `fdisk` utility:  

```
# fdisk /dev/sdb
```
  2. 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
```
  3. Create the `zenoss_data` volume group and `lv_analytics` logical disk:  

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

### Creating the DRDB Resource

The following procedure creates the DRDB resources:

1. Copy the following rpm files from the cluster manager to the database nodes:
  - `drbd-pacemaker-<version>.rpm`
  - `drbd-utils-<version>.rpm`
  - `drbd-km-<version>.rpm`
  - `drbd-rgmanager-<version>.rpm`
2. Issue the following commands as root to install the packages:  

```
# 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-incondegr.sh;
            /usr/lib/drbd/notify-emergency-reboot.sh; echo b >
            /proc/sysrqtrigger;

            reboot -f";

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

            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-resynctarget
            "/usr/lib/drbd/crm-unfence-peer.sh";
        }

        disk {

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

```

#### 4. Create the `/etc/drbd.d/r0.res` file.

**Note:** In the following configuration, replace `ANALYTICS{1,2}` and `{1,2}-PRIVATEIP` with the names and IP addresses used in your environment.

**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 to create the file:

```
resource r0 {

    volume 0 {
        device /dev/drbd0;
        disk /dev/zenoss_data/lv_analytics;
            flexible-meta-disk internal;
    }

    net {
        use-rle;
    }

    on ANALYTICS1 {
        address ANALYTICS1-PRIVATEIP:7788;
    }

    on ANALYTICS2 {
        address ANALYTICS2-PRIVATEIP:7788;
    }
}
```

5. Create the resource **r0**:

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

6. Start the DRBD service.

**Note:** Ensure the resource on the other node is already configured before starting the DRBD service.

```
# service drbd start
```

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

7. Set ANALYTICS1 as *primary*, execute the following command on ANALYTICS1:

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

8. Give the nodes enough time to synchronize their disks. Check the status with 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
```

9. On ANALYTICS1, initialize the DRBD disk filesystem:

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

**Note:** It is not necessary to perform this on the second node because this action is replicated

automatically.

10. On both nodes, create the `/opt/zenoss_analytics` directory:

```
# mkdir -p /opt/zenoss_analytics
```

11. Mount the DRBD disk on the active or ANALYTICS1 node:

```
# mount /dev/drbd0 /opt/zenoss_analytics
```

**Note:** It is not necessary to add the disk to the `/etc/fstab` file because mounting is managed by the RHCS `rgmanager`.

## Installing Analytics Dependencies on the Analytics Nodes

Perform the following procedure to install dependencies for Analytics on the nodes:

1. On both nodes, remove conflicting messaging systems:

```
# rpm -qa | egrep -i "matahari|qpuid"
```

2. Download and install the Sun JDK version 6 installer from Oracle website.

**Note:** JRE and JDK are not the same.

- a. Download the self-installing RPM of Oracle JDK from the Java JDK 6 Downloads page.

- b. Make the RPM installer executable:

```
# chmod +x /path-to-installer/jdk_filename-rpm.bin
```

- c. Start the installer:

```
# /path-to-installer/jdk_filename.bin
```

3. Set the `JAVA_HOME` path in `/etc/profile`:

- a. Locate the path to the JDK directory. Typically, this is `/usr/java/jdkVersionNumber`.

- b. Add the following lines to the `/root/.bash_profile` file:

```
export JAVA_HOME=/usr/java/jdkVersionNumber
export PATH=$JAVA_HOME/bin:$PATH
```

4. Reboot the system.

5. Issue the following command to verify the path:

```
#echo $JAVA_HOME
```

6. Start DRBD on both nodes:

```
# service drbd start
```

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

7. Issue the following command on the primary node:

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

8. Re-mount the drbd drive on `/opt/zenoss_analytics`:

```
# mount /dev/drbd0 /opt/zenoss_analytics
```

9. Download and install the latest ZenDS package. Do not start the service.

```
# yum install perl-DBI dmidecode
# rpm -ivh zends-<version>.rpm
```

10. On the Zenoss Resource Manager server, install the *zenetl* zenpack:

**Note:** If there are remote hubs and collectors, update them also.

a. Issue the following command as the zenoss user:

```
# zenpack --install [egg name]
```

b. Update the *daemons.txt* and *collectordaemons.txt* files, if necessary.

c. Restart Zenoss.

11. Restart the zenoss service on the Zenoss Resource Manager server, collectors and hubs.

12. Ensure that hostnames are resolvable from both nodes.

## Installing Analytics on ANALYTICS1

Perform the following procedure to install Analytics on ANALYTICS1:

1. Issue the following command to Install Analytics:

```
# yum --nogpgcheck localinstall zenoss_analytics-<version>.noarch.rpm
```

2. Enter the correct hostname, IP address, and credentials to the ZenDS in the */etc/zenoss\_analytics/zenoss\_analytics.conf* file.

**Note:** Replace ANALYTICSDBVIP with the actual DB cluster VIP in your environment.

```
# Zenoss Analytics Data Warehouse Settings
etl.jdbc.hostname=ANALYTICSDBVIP
etl.jdbc.port=13306
etl.jdbc.dbname=reporting
etl.jdbc.username=reporting
etl.jdbc.password=reporting

# Zenoss Analytics UI Database Settings
analytics.jndiName=jdbc/jasperserver
analytics.jdbc.driver=com.mysql.jdbc.Driver
analytics.jdbc.username=reporting
analytics.jdbc.password=reporting
analytics.jdbc.hostname=ANALYTICSDBVIP
analytics.jdbc.port=13306
analytics.jdbc.dbname=zenoss_analytics
analytics.jdbc.serverType=mysql
analytics.hibernate.dialect=org.hibernate.dialect.MySQLDialect

# Database Superuser credentials
# Used for creating initial databases and users
```

```
# It is safe to comment/remove these credentials after installation
etl.jdbc.adminUser=root
etl.jdbc.adminPassword=
analytics.jdbc.adminUser=root
analytics.jdbc.adminPassword=
```

3. Verify connectivity to the database:

```
$ mysql -P 13306 -uDBVIP -u root
```

4. Install the database schema with the command:

```
# /opt/zenoss_analytics/bin/upgrade_db.py
```

5. Start the analytics service:

```
# service zenoss_analytics start
```

**Note:** It is not necessary to set the service to start on boot because it is handled by the RHCS rgmanager.

6. Login to the Zenoss Resource Manager and create a user with the **ReportingUser** role.

7. Login to <http://ANALYTICS1:7070> as **superuser** (password *superuser*) and configure the database credentials:

- a. On the *Analytics Getting Started* page, click **View Reports**.
- b. In the *Refine* area, change the *Reports* item to **Data Sources**.
- c. In the *JDBC Data Sources* area, right-click the **Zenoss Reporting** data source.
- d. Select **Edit** from the list of options.
- e. Enter the **DB cluster's IP address** in the *URL* field.
- f. Click **Test Connection** to verify the changes are valid.
- g. Click **Save** to save the changes and exit.

## Installing Analytics on ANALYTICS2

Perform the following procedure to install Analytics on ANALYTICS2:

1. On ANALYTICS2, download and run the Analytics installer:

```
# yum --nogpgcheck localinstall zenoss_analytics-<version>. noarch.rpm
```

2. Copy configuration files from ANALYTICS1 to ANALYTICS2. On ANALYTICS1, execute the following command:

```
# scp -r /etc/zenoss_analytics/* root@ANALYTICS2:/etc/zenoss_analytics/
```

3. On ANALYTICS2, delete the contents of the `/opt/zenoss_analytics` directory. The contents will be replaced by the replicated data in the DRBD disk. Issue the following command:

```
# rm -rf /opt/zenoss_analytics/*
```

## Testing Analytics Manual Failover/Failback

Perform the following procedure to test and verify a manual failover of the Analytics DB.

To verify that Analytics failover works before configuring automatic failover, run the following commands to initiate failover:

1. On ANALYTICS1, stop the *zenoss\_analytics* service:  

```
# service zenoss_analytics stop
```
2. Unmount the */dev/drbd0* device:  

```
# umount /dev/drbd0
```
3. On ANALYTICS1, make the *drbd* disk the **secondary**:  

```
# drbdadm secondary r0
```
4. On ANALYTICS2, make the *drbd* disk the **primary**:  

```
# drbdadm primary r0
```
5. Mount */dev/drbd0* to **/opt/zenoss\_analytics**:  

```
# mount /dev/drbd0 /opt/zenoss_analytics
```
6. Start *zenoss\_analytics*:  

```
# service zenoss_analytics start
```

The analytics service can now be accessed through ANALYTICS2's public IP address.

**Note:** When the clustering is configured, the analytics cluster can be accessed through its cluster or VIP address.

7. To failback, repeat steps 1-6 and interchange the names *ANALYTICS1* and *ANALYTICS2*.

## Preparing the Analytics Cluster for Configuration

Perform the following procedure to prepare the cluster for configuration:

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

```
# yum install rgmanager ricci cman
```
2. Set a password for the user *ricci*:  

```
# passwd ricci
```
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 Analytics nodes:  

```
# service ricci start
```
5. Browse to <https://CLUSTERMGR:8084>.
6. Login to *luci* as **root**.

## Configuring the Analytics Cluster

Perform the following procedure to configure the Analytics cluster:

1. Under *Manage Clusters*, click **Create**.
2. Enter **ANALYTICS** as the *cluster* name, in this example. Replace *ANALYTICS* with your cluster name.
3. Enter the *node names* and their *ricci ports, users* and *passwords*.
4. Leave Other Fields as *default*.
5. Click **Create Cluster**.
6. If you need to modify node attributes, click the **Nodes** tab to display the node attribute pane.

## Creating the Analytics Failover Domain

Perform the following procedure to create the Analytics failover domain:

1. Under the *ANALYTICS* cluster, click the **Failover Domains** tab.
2. Click **Add**.
3. Enter the **ANALYTICS\_domain** in the *name* field, using your cluster name.
4. Check **Prioritized**.
5. Check the *member nodes*, in this example **ANALYTICS{1,2}**.
6. Set the priority of *ANALYTICS1* to **1** and *ANALYTICS2* to **2**. This means *ANALYTICS1* has the higher priority of the two nodes.
7. Click **Create**.

## Creating the DRBD Resources

Perform the following procedure to create the necessary DRBD resources:

1. Under the *ANALYTICS* cluster, click the **Resources** tab.
2. Click **Add** to create the *DRBD resource*.
3. Select **DRBD Resource** as the *resource type*.
4. Enter **analytics\_drbd** as the *name* and **r0** as the *DRBD resource name*.
5. Click **Submit**.
6. Click **Add** to create the *filesystem resource*.
7. Select **Filesystem** as the *resource type*.
8. Enter **analytics\_dir** as the *name*.
9. Select *ext4* as the *filesystem type*, or if yours is different, specify it.
10. Enter **/opt/zenoss\_analytics** as the *mount point*.
11. Enter **/dev/drbd0** as the *device*.
12. Enter defaults as mount options.
13. Click **Submit**.
14. Click **Add** to create the *init script resource*.

15. Select **Script** as the *resource type*.
16. Enter **analytics\_init** as the *name*.
17. Enter **/etc/init.d/zenoss\_analytics** as the *full path to script file*.
18. Click **Submit**.
19. Click **Add** to create the *IP Address resource*.
20. Select **IP Address** as the *resource type*.
21. Enter **ANALYTICSVIP** as the *IP address* in this example. Replace the *ANALYTICSVIP* string with your VIP address.
22. Ensure the **Monitor Link** is checked.
23. Click **Submit**.

## Creating the Service Groups

Perform the following procedure to create the required service groups:

1. Under the *ANALYTICS* cluster, click the **Service Groups** tab.
2. Click **Add**.
3. Enter **analytics** for the *service name*.
4. Check **Automatically Start this Service**.
5. Select **ANALYTICS\_domain** as the *failover domain*.
6. Select **Relocate** as the *recovery policy*.
7. Click **Add Resource**.
8. Select **analytics\_drbd**.
9. At the bottom of *analytics\_drbd*, click **Add Child Resource**.
10. Select **analytics\_dir**.
11. At the bottom of *analytics\_dir*, click **Add Child Resource**.
12. Select **analytics\_init**.
13. At the bottom of *analytics\_init*, click **Add Child Resource**.
14. Select **ANALYTICSVIP**.
15. Click **Submit**.

The cluster can now be started and accessed through its VIP address.

16. Ensure the Zenoss Resource Manager is configured to point to the *Analytics Cluster VIP*.

## Failure Modes

This section describes failure modes for the cluster.

### Node Failure

In case one of the cluster nodes suffers an outage, the cluster shifts all resources to the other node. Because DRBD provides a synchronous replica of all zends or analytics data to the other node, the cluster can continue to serve from the secondary nodes.

#### Node Failure Test Case

1. Shut down primary node.
2. Wait.
3. Start the primary node. The following output confirms the failure:

```
Storage Subsystem Failure
Disk unmounted
Disk umount lazy - failed. needs user intervention.
Sep 1 10:32:33 analytics1 rgmanager[2833]: stop on drbd "analytics_drbd"
returned 1 (generic error)
Sep 1 10:32:33 analytics1 rgmanager[2833]: #12: RG service:zenoss_analytics
failed to stop;> intervention required Sep 1 10:32:33 analytics1
rgmanager[2833]: Service service:zenoss_analytics is failed
```

### Zenoss Analytics or ZenDS Failure

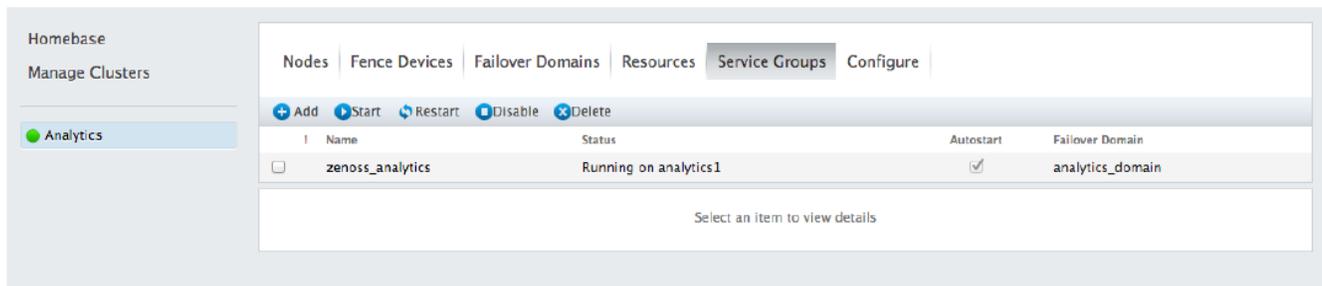
This section describes the failure for either Zenoss Analytics or ZenDS.

#### Analytics or ZenDS Failure Test Case

```
daemon accidentally stopped
Cluster Network Failure
Public Network Failure
Recurring Storage Failure (Toggling)
```

## 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.