



## **Integrating xiRAID Classic 4.1.0 Into a Pacemaker Cluster**

# Contents

<b>Integrating xiRAID Classic 4.1.0 Into a Pacemaker Cluster.....</b>	<b>3</b>
Introduction.....	3
Prerequisites.....	3
xiRAID Installation.....	4
Csync <sup>2</sup> Installation & Configuration.....	5
Installing Csync <sup>2</sup> .....	5
Csync <sup>2</sup> Configuration.....	7
Configuring Scheduled Synchronization.....	8
Managing RAIDs in a cluster.....	9
RAID and Cluster Resource Creation.....	9
Changing RAID parameters.....	11
Deleting RAIDs.....	12

# Integrating xiRAID Classic 4.1.0 Into a Pacemaker Cluster

Instructions on how to integrate xiRAID Classic into an existing Pacemaker cluster.

## Introduction

xiRAID Classic can operate within a High Availability (HA) cluster, ensuring data integrity and availability for users by distributing data and system components across two nodes. To set up a High Availability cluster for xiRAID Classic, configure and connect two cluster nodes to a shared set of drives. In case of a node failure, the xiRAID Classic components can migrate to other operational nodes within the cluster, maintaining continuous operations without disruption.

The management of xiRAID Classic components within an HA cluster is conducted by the Pacemaker cluster manager, which coordinates failover mechanisms and ensures transitions of RAIDs between nodes. File synchronization among clustered nodes can be facilitated by the Csync<sup>2</sup> utility, which includes conflict detection, file deletion detection, and update handling. However, users have the option to choose any other utility for this purpose.

This guide provides instructions on integrating xiRAID into an existing Pacemaker cluster.

## Prerequisites

The following prerequisites must be fulfilled to successfully integrate xiRAID Classic 4.1.0 into an existing two-node Pacemaker cluster:

1. The Pacemaker cluster software is installed and configured to run on two nodes.
2. Fencing (STONITH) is configured in the cluster.
3. Nodes in the cluster meet the requirements outlined in the xiRAID Classic Software Requirements document.
4. The Pacemaker cluster manager version is 2.1.6 or later.
5. The firewall on both nodes is configured in a way that allows nodes to communicate with each other.
6. The time is synchronized across all the cluster nodes.
7. Disks that will be used to create xiRAID RAID(s) are accessible to both nodes in the cluster.
8. The jq package (command-line JSON processor) version 1.6 or later is installed on both nodes. Execute the following command to install the jq package:

- a. RHEL and RHEL-based systems:

```
# yum install jq
```

- b. Oracle Linux:

```
# dnf install jq
```



It is crucial that all of the prerequisites above are fulfilled before you proceed to the next section.

## xiRAID Installation

1. Follow the instructions in the xiRAID Classic Installation Guide to install xiRAID Classic 4.1.0 on both nodes in the cluster.
2. Follow the instructions in the xiRAID Classic 4.1.0 Administrator's Guide to apply a xiRAID license to both nodes in the cluster.



Each node in the cluster requires a separate license.

3. On both nodes, install the xiRAID resource agent script for Pacemaker.

During the xiRAID installation, this script was added to the `/etc/xraid/agents` directory. For this agent to work with Pacemaker, you need to create a symbolic link from `/etc/xraid/agents` to the directory with Pacemaker resource agent scripts `/usr/lib/ocf/resource.d`:


```
# mkdir -p /usr/lib/ocf/resource.d/xraid
# ln -s /etc/xraid/agents/raid /usr/lib/ocf/resource.d/xraid/raid
```

## Csync<sup>2</sup> Installation & Configuration

Csync<sup>2</sup> is a utility for asynchronous file synchronization. Csync<sup>2</sup> is used to synchronize RAID config files between nodes in a cluster.

### Installing Csync<sup>2</sup>

This section describes how to install Csync<sup>2</sup>.

 Csync<sup>2</sup> must be installed on both nodes in the cluster.

### RHEL-Based Systems

On RHEL-based systems, Csync<sup>2</sup> is not available from the official repositories, but it is included in the xiRAID repository. The xiRAID Classic repository is installed during the xiRAID Classic installation, so no additional steps are required to install Csync<sup>2</sup>. Simply run the following command:

```
# yum install csync2
```

### Oracle Linux

On Oracle Linux, Csync<sup>2</sup> is not available from the official repositories, but it is included in the xiRAID repository. The xiRAID Classic repository is installed during the xiRAID Classic installation, so no additional steps are required to install Csync<sup>2</sup>. Simply run the following command:

```
# dnf install csync2
```

## Ubuntu

On Ubuntu, Csync<sup>2</sup> can be installed from the official repository:

```
# apt install csync2
```



Instructions in the following sections are written with the assumption that you've installed Csync<sup>2</sup> from the source code or the xiRAID repository. If you've installed Csync<sup>2</sup> from the official repository, make sure to change the path to the Csync<sup>2</sup> binary in configuration files and commands where the path to Csync<sup>2</sup> is specified.

## Installation From Source

1. Install the required dependencies:

a. RHEL and RHEL-based systems:

```
# yum install automake byacc flex git gcc gnutls-devel  
librsync librsync-devel libsqlite3x-devel libpq-devel make  
mysql-devel
```

b. Oracle Linux:

```
# dnf install automake byacc flex git gcc gnutls-devel  
librsync librsync-devel libsqlite3x-devel libpq-devel make  
mysql-devel
```

2. Clone the Csync<sup>2</sup> repository:

```
# git clone https://github.com/LINBIT/csync2.git
```

3. Compile and install Csync<sup>2</sup>:

```
# cd csync2
# ./autogen.sh
# ./configure
# make
# make install
```

4. Create the systemd service unit configuration file:

```
# vi /etc/systemd/system/csync2.service
```

5. Paste the following content into the file:

```
[Unit]
Description=csync2 file synchronization tool using librsync and
current state databases
After=network.target

[Service]
Type=idle
ExecStart=/usr/local/sbin/csync2 -ii -l
Restart=on-failure

[Install]
WantedBy=pcsd.service
```

6. Start the service:

```
# systemctl start csync2.service
```

7. Configure the service to be started automatically at boot:

```
# systemctl enable csync2.service
```

## Csync<sup>2</sup> Configuration



Steps 1-4 should be performed on the first node.

### 1. Create the configuration file:

```
# vi /usr/local/etc/csync2.cfg
```

### 2. Paste the following content into the file:

```
nossl * *;  
group csxiha {  
    host <node1_hostname>;  
    host <node2_hostname>;  
    key /usr/local/etc/csync2.key_ha;  
    include /etc/xiraid/raids;  
}
```

### 3. Generate the encryption key:

```
# csync2 -k /usr/local/etc/csync2.key_ha
```

### 4. Copy the configuration file and the generated key to the second node:

```
# scp /usr/local/etc/csync2.cfg /usr/local/etc/csync2.key_ha  
    <node2_hostname>:/usr/local/etc/
```

### 5. Configure the firewall on both nodes to allow incoming TCP traffic on port 30865.

Firewall-cmd example:

```
# firewall-cmd --add-port=30865/tcp  
# firewall-cmd --permanent --add-port=30865/tcp
```

## Configuring Scheduled Synchronization

**Configure the synchronization task to run every minute using cron:**

### 1. Open the crontab file:

```
# crontab -e
```

### 2. Add the following line to the file to schedule the synchronization task to run every minute:



```
* * * * * /usr/local/sbin/csync2 -x
```

3. Save the file.

### Create a script to immediately sync the configuration files when any configuration file is updated:

1. Create the script file:

```
# vi /etc/xiraid/config_update_handler.sh
```

2. Paste the following content into the file:

```
#!/usr/bin/bash
/usr/local/sbin/csync2 -xv
```



If necessary, update the path to csync2 in the script.

3. Make the script executable:

```
# chmod +x /etc/xiraid/config_update_handler.sh
```

## Managing RAIDs in a cluster

In this chapter you will learn how to create, manage and delete xiRAID Classic RAID objects in HA cluster configuration.

### RAID and Cluster Resource Creation



Spare pools are not supported in an HA cluster configuration.

Follow the steps below to create a xiRAID RAID and add a Pacemaker resource to manage it:

1. Disable RAID autostart:

```
# xicli settings cluster modify --raid_autostart 0
```



It is important to disable xiRAID autostart before creating any RAIDs to prevent data corruption due to accidental active-active mode.

2. Create a xiRAID RAID by following the instructions in the xiRAID Classic 4.1.0 Administrator's Guide.

3. Get the RAID UUID:

```
# xicli raid show -n <raid_name> -f json
```

4. Unload the RAID:

```
# xicli raid unload -n <raid_name>
```

5. Add the Pacemaker resource to manage the RAID using the RAID name (<raid\_name>) and UUID (<raid\_uuid>):

```
# pcs resource create <raid_name> ocf:xraid:raid \  
    name=<raid_name> uuid=<raid_uuid> \  
    op monitor interval=5s \  
    meta migration-threshold=3
```



You can set the *migration-threshold* value for a resource, allowing it to move to a new node after a specified number of failures. When working with xiRAID RAIDs, it is recommended to set this parameter to 3. To learn more, see [Moving Resources Due to Failure](#).

## Configuring Multiple RAID Pacemaker Resources

If you have configured multiple Pacemaker resources to manage xiRAID RAIDs in a single cluster, it is important to add Pacemaker constraints to prevent these RAID resources from starting simultaneously. If you have 2 RAIDs, you only need to add a single constraint:

```
# pcs constraint order start <RAID resource 1> then start <RAID  
resource 2> kind=Serialize
```

, where <RAID resource 1> is the name of one of the RAID resources and <RAID resource 2> is the name of the other resource.

Here's an example with 5 RAID resources:

```
rr_mdt0  
rr_ost0  
rr_ost1  
rr_ost2  
rr_ost3
```

To ensure that these resources do not start in parallel, multiple constraints must be added:

```
# pcs constraint order start rr_mdt0 then start rr_ost0  
kind=Serialize  
# pcs constraint order start rr_mdt0 then start rr_ost1  
kind=Serialize  
# pcs constraint order start rr_mdt0 then start rr_ost2  
kind=Serialize  
# pcs constraint order start rr_mdt0 then start rr_ost3  
kind=Serialize  
  
# pcs constraint order start rr_ost0 then start rr_ost1  
kind=Serialize  
# pcs constraint order start rr_ost0 then start rr_ost2  
kind=Serialize  
# pcs constraint order start rr_ost0 then start rr_ost3  
kind=Serialize  
  
# pcs constraint order start rr_ost1 then start rr_ost2  
kind=Serialize  
# pcs constraint order start rr_ost1 then start rr_ost3  
kind=Serialize  
  
# pcs constraint order start rr_ost2 then start rr_ost3  
kind=Serialize
```

## Changing RAID parameters



Please ensure that you change the RAID parameters on the cluster node where it is currently active.

Change the existing RAID's parameters using the instructions provided in the xiRAID Classic 4.1.0 Administrator's Guide.

Currently, changing the RAID name is not supported. To change the RAID name, you need to delete the Pacemaker resource that manages the RAID, update the RAID name, and then create a new Pacemaker resource with the new name.

## Deleting RAIDs

To delete a RAID in HA cluster:

1. Delete the Pacemaker resource that manages the RAID:

```
# pcs resource delete <raid_name>
```

2. Delete the RAID using the instructions provided in the xiRAID Classic 4.1.0 Administrator's Guide. Example:

```
# xicli raid destroy -n <raid_name>
```