

**XINNOR**

**xiRAID Classic 4.3.1  
Installation and Update Guides**

# Contents

|   |          |
|---|----------|
| <b>xiRAID Classic 4.3.1 Installation and Update Guides.....</b> | <b>3</b> |
| xiRAID Classic 4.3.1 System Requirements.....                   | 3        |
| Hardware Requirements.....                                      | 3        |
| Software Requirements.....                                      | 3        |
| xiRAID Classic 4.3.1 Installation Guide.....                    | 7        |
| Installing xiRAID Classic on RHEL.....                          | 7        |
| Installing xiRAID Classic on Alma Linux and Rocky Linux.....    | 12       |
| Installing xiRAID Classic on Oracle Linux UEK.....              | 16       |
| Installing xiRAID Classic on Ubuntu.....                        | 19       |
| Installing xiRAID Classic on Proxmox.....                       | 23       |
| Uninstalling xiRAID Classic 4.3.1.....                          | 27       |
| xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 Update Guide.....  | 28       |
| Updating xiRAID Classic on RHEL and RHEL-based Systems.....     | 29       |
| Updating xiRAID Classic on Oracle Linux.....                    | 36       |
| Updating xiRAID Classic on Ubuntu.....                          | 43       |
| Updating xiRAID Classic on Proxmox.....                         | 51       |

# xiRAID Classic 4.3.1 Installation and Update Guides

## xiRAID Classic 4.3.1 System Requirements

### Hardware Requirements

| Component | Requirements  |
|-----------|---|
| Processor | x86_64 instruction set.<br><br>With AVX support.<br><br>With AVX2 support (RAID N+M requires AVX2). |
| RAM       | Minimum 16 GiB.   |

### Software Requirements

The generic version of xiRAID Classic 4.3.1 is compatible with the distributions and cores listed below. Support of other Linux distributions and custom user kernels may be provided on request at [support@xinnor.io](mailto:support@xinnor.io)

For versions of distributions with DKMS, it is possible to use kernels higher than the version specified in the requirements as part of patch versions. For more details, see the chapter “DKMS”.

### Secure Boot

To ensure that xiRAID Classic operates correctly, Secure Boot must be disabled.

## DKMS

### For the xiraid kernel module with DKMS:

- The *xiraid* kernel module uses DKMS (Dynamic Kernel Module Support) technology and is automatically built and is installed for the specified version of the Linux kernel or higher, but only within the patch versions (without kernel API or ABI changes).
- Headers of your current kernel version are required to install the *xiraid* kernel module.
- If you update the kernel, you need headers for the new kernel version.

Do not update the kernel more than a patch update (with kernel API or ABI changes) because the *xiraid* kernel module will not be loaded.

### For the xiraid kernel module without DKMS:

The *xiraid* kernel module is installed only for the specified version of the Linux kernel. Do not update the kernel version on builds with *xiraid* without DKMS: the *xiraid* kernel module will not be loaded after updating the system kernel.

## Supported Operating System Distributions

| Operating System                             |         | Minimum Required Kernel      |
|--|---------|------------------------------|
| Name   | Version |                              |
| RHEL & RHEL-based (Rocky Linux & Alma Linux) | 8       | kernel-4.18.0-305.el7.x86_64 |
|  | 9.0     | kernel-5.14.0-70.el9_0       |
|  | 9.1     | kernel-5.14.0-162.el9_1      |
|  | 9.2     | kernel-5.14.0-284.el9_2      |
|  | 9.3     | kernel-5.14.0-362.el9_3      |
|  | 9.4     | kernel-5.14.0-427.13.1.el9_4 |

| Operating System   |         | Minimum Required Kernel        |
|--------------------|---------|--------------------------------|
| Name               | Version |                                |
|                    | 9.5     | kernel-5.14.0-503.el9_5        |
|                    | 9.6     | kernel-5.14.0-570.5.1.el9_6    |
|                    | 9.7     | kernel-5.14.0-611.5.1.el9_7    |
|                    | 10      | kernel-6.12.0-55.9.1.el10_0    |
|                    | 10.1    | kernel-6.12.0-124.8.1.el10_1   |
| Oracle Linux (UEK) | 8.4     | kernel-uek-5.4.17-2102el8uek   |
|                    | 8.6     | kernel-uek-5.4.17-2102el8uek   |
|                    | 9       | kernel-uek-core-5.15.0-0el9uek |
| Ubuntu             | 20.04   | linux-image-5.4.0-54-generic   |
|                    | 22.04   | linux-image-5.15.0-27-generic  |
|                    | 24.04   | linux-image-6.8.0-36-generic   |
| Proxmox            | 8       | pve-kernel-6.2.16-3-pve        |
|                    | 8.1     | proxmox-kernel-6.5.13-5-pve    |
|                    | 8.2     | proxmox-kernel-6.8.4-2-pve     |
|                    | 8.3     | proxmox-kernel-6.8.12-4-pve    |
|                    | 8.4     | proxmox-kernel-6.8.12-9-pve    |
|                    | 9.0     | proxmox-kernel-6.14.8-2-pve    |
|                    | 9.1     | proxmox-kernel-6.17.2-1-pve    |

## Program Packages provided with xiRAID Classic 4.3.1

| General List of Program Packages | Additional Packages for Specific Distributions               |
|----------------------------------|--|
| coreutils<br>dkms                | <b>For RHEL, Alma Linux, Rocky Linux, Oracle Linux (EL):</b> |

| General List of Program Packages | Additional Packages for Specific Distributions     |
|----------------------------------|--|
| gcc                              | kernel-devel                                       |
| make                             | <b>For Oracle Linux (UEK):</b><br>kernel-uek-devel |
| python3-pyudev                   |  |
| python3-systemd                  | <b>For Ubuntu:</b><br>linux-headers                |
| python3-bpfcc                    |  |
| python-bcc                       | <b>For Proxmox:</b><br>pve-headers                 |
| nvme-cli                         |  |
| sg3-utils                        |  |
| smartmontools                    |  |
| udev                             |  |
| ledmon                           |  |
| versionlock                      |  |
| lsof                             |  |
| jq v. >= 1.6                     |  |
| systemd                          |  |

## Additional Program Packages

These program packages are not provided with the product and need to be manually installed in order to use the xiRAID Classic in cluster mode:

| Program Package    | Version  |
|--------------------|----------|
| Pacemaker          | >= 2.1.6 |
| Csync <sup>2</sup> | >= 2.0   |

(continued)

| Program Package | Version |
|-----------------|---------|
| jq              | >= 1.6  |

## xiRAID Classic 4.3.1 Installation Guide

This document describes instructions on installation of the xiRAID Classic 4.3.1 software.

You can install your xiRAID Classic 4.3.1 using the “xiraid-repo” repository. For details, see the chapter “Installing xiRAID Classic 4.3.1” for your system.

Most of the commands presented in this document are run only with superuser privileges. Please log in as an administrator or root to run these.

## Installing xiRAID Classic on RHEL

### Prerequisites

During the installation process, additional packages required by xiRAID Classic 4.3.1 are automatically installed on your system (see the list of additional packages in the document xiRAID Classic 4.3.1 System Requirements). To install additional packages correctly, make sure that the appropriate repositories are configured for them.



Please ensure that the Protobuf library for Python is not already installed on your system via pip before installing xiRAID Classic 4.3.1



Please ensure that Secure Boot is disabled on your system.

The Version Lock plugin will be installed along with the other xiRAID packages. Once the installation is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (apt/yum/dnf update).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at <https://www.xinnor.io/>. We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module.



However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

## Installation on RHEL

When installing Xinnor xiRAID on RHEL, the latest kernel version will be installed. To revert your system to the kernel version used prior to



installation, you need to change the default kernel that is launched. Specify the kernel version you want to return to {kernel\_version}:

```
# grubby --set-default "/boot/vmlinuz-<kernel_version>"
```

To install xiRAID Classic 4.3.1 on a RHEL system:

1. Install EPEL depending on your OS version:

EPEL (Extra Packages for Enterprise Linux) is a repository containing additional software packages required for the xiRAID installation.

## ◦ RHEL 8:

```
# subscription-manager repos --enable
codeready-builder-for-rhel-8-$(arch)-rpms
# dnf install -y https://dl.fedoraproject.org/pub/epel/
epel-release-latest-8.noarch.rpm
```

## ◦ RHEL 9:

```
# subscription-manager repos --enable
codeready-builder-for-rhel-9-$(arch)-rpms
# dnf install -y https://dl.fedoraproject.org/pub/epel/
epel-release-latest-9.noarch.rpm
```

## ◦ RHEL 10:

```
# subscription-manager repos --enable
codeready-builder-for-rhel-10-$(arch)-rpms
# dnf install -y https://dl.fedoraproject.org/pub/epel/
epel-release-latest-10.noarch.rpm
```

## 2. Install kernel-headers for the currently loaded kernel:

```
# dnf install kernel-devel-$(uname -r)
```

## 3. Install xiraid-repo for your OS:

- RHEL 8:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/8/kver-4.18/xiraid-repo-1.3.0-1588.kver.4.18.noarch.rpm
```

- RHEL 9:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/9/kver-5.14/xiraid-repo-1.3.0-1588.kver.5.14.noarch.rpm
```

- RHEL 10:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/10/kver-6.12/xiraid-repo-1.3.0-1588.kver.6.12.noarch.rpm
```

#### 4. Install xiraid.

This command installs xiRAID Classic 4.3.1.

```
# dnf install xiraid-core
```

## Testing the Installation

To ensure that the installation was successful:

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the installation has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

3. Check the status of your license:

```
xicli license show
```

The first time you call the command after the installation, the system prompts you to accept the EULA conditions. If you accept the conditions, the command outputs your license details.



To request a trial license, please do not hesitate to contact us at [support@xinnor.io](mailto:support@xinnor.io).

#### 4. Check the xiRAID Classic operability:

```
# xicli raid show
```

This command shows a table with RAIDs. If the output is not in the form of a table or if there is no output at all, it indicates that the installation was not successful.

#### 5. Make sure that xiRAID Classic has been installed on the latest kernel version available on your system.

Otherwise, follow these steps to use xiRAID Classic after switching to the latest kernel version:

##### a. Stop the xiraid target service:

```
# systemctl stop xiraid.target
```

##### b. Boot into the desired kernel.

##### c. Install the development kernel-headers package for the current kernel.

##### d. Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

After the installation, you can find the installation logs at `/var/log/xraid/`.

If installation was not successful:

- check the kernel version;
- try to reinstall xiRAID Classic 4.3.1.

If this does not help, please contact Xinnor Support team at [support@xinnor.io](mailto:support@xinnor.io) with attached logs from `/var/log/xraid/`.

## Installing xiRAID Classic on Alma Linux and Rocky Linux

### Prerequisites

During the installation process, additional packages required by xiRAID Classic 4.3.1 are automatically installed on your system (see the list of additional packages in the document xiRAID Classic 4.3.1 System Requirements). To install additional packages correctly, make sure that the appropriate repositories are configured for them.



Please ensure that the Protobuf library for Python is not already installed on your system via pip before installing xiRAID Classic 4.3.1



Please ensure that Secure Boot is disabled on your system.

The Version Lock plugin will be installed along with the other xiRAID packages. Once the installation is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (apt/yum/dnf update).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at <https://www.xinnor.io/>. We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module.



However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

## Installation on Alma Linux and Rocky Linux

When installing Xinnor xiRAID on Alma Linux 9.\* and Rocky Linux 9.\*, the latest kernel version will be installed. To revert your system to the



kernel version used prior to installation, you will need to change the default kernel that is launched. Specify the kernel version you want to return to (`kernel_version`):

```
# grubby --set-default "/boot/vmlinuz-<kernel_version>"
```

To install xiRAID Classic 4.3.1 on Alma Linux or Rocky Linux:

1. Install EPEL depending on your OS version:

EPEL (Extra Packages for Enterprise Linux) is a repository containing additional software packages required for the xiRAID installation.

```
# dnf install -y epel-release
```

2. Install kernel-headers for the currently loaded kernel:

- a. Rocky Linux 8 & Alma Linux 8:

```
# dnf install kernel-headers-$(uname -r)
kernel-devel-$(uname -r)
```

- b. Rocky Linux and Alma Linux versions 9 and 10:

```
# dnf install kernel-{headers,devel-matched}-$(uname -r)
```

3. Install xiraid-repo for your OS:

#### a. Rocky Linux 8 & Alma Linux 8:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/8/kver-4.18/xiraid-repo-1.3.0-1588.kver.4.18.noarch.rpm
```

#### b. Rocky Linux 9 & Alma Linux 9:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/9/kver-5.14/xiraid-repo-1.3.0-1588.kver.5.14.noarch.rpm
```

#### c. Rocky Linux 10 & Alma Linux 10:

```
# dnf install https://pkg.xinnor.io/repository/Repository/xiraid/el/10/kver-6.12/xiraid-repo-1.3.0-1588.kver.6.12.noarch.rpm
```

### 4. Install xiraid.

This command installs xiRAID Classic 4.3.1.

```
# dnf install xiraid-core
```

## Testing the Installation

To ensure that the installation was successful:

#### 1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the installation has not completed successfully.

#### 2. Check the xiRAID Classic version:

```
# xicli -v
```

#### 3. Check the status of your license:

```
xicli license show
```

The first time you call the command after the installation, the system prompts you to accept the EULA conditions. If you accept the conditions, the command outputs your license details.



To request a trial license, please do not hesitate to contact us at [support@xinnor.io](mailto:support@xinnor.io).

#### 4. Check the xiRAID Classic operability:

```
# xicli raid show
```

This command shows a table with RAIDs. If the output is not in the form of a table or if there is no output at all, it indicates that the installation was not successful.

#### 5. Make sure that xiRAID Classic has been installed on the latest kernel version available on your system.

Otherwise, follow these steps to use xiRAID Classic after switching to the latest kernel version:

##### a. Stop the xiraid target service:

```
# systemctl stop xiraid.target
```

##### b. Boot into the desired kernel.

##### c. Install the development kernel-headers package for the current kernel.

##### d. Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

After the installation, you can find the installation logs at `/var/log/xraid/`.

If installation was not successful:

- check the kernel version;
- try to reinstall xiRAID Classic 4.3.1.

If this does not help, please contact Xinnor Support team at [support@xinnor.io](mailto:support@xinnor.io) with attached logs from `/var/log/xraid/`.

## Installing xiRAID Classic on Oracle Linux UEK

### Prerequisites

During the installation process, additional packages required by xiRAID Classic 4.3.1 are automatically installed on your system (see the list of additional packages in the document xiRAID Classic 4.3.1 System Requirements). To install additional packages correctly, make sure that the appropriate repositories are configured for them.



Please ensure that the Protobuf library for Python is not already installed on your system via pip before installing xiRAID Classic 4.3.1



Please ensure that Secure Boot is disabled on your system.

The Version Lock plugin will be installed along with the other xiRAID packages. Once the installation is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (apt/yum/dnf update).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at <https://www.xinnor.io/>. We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module.



However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

## Installation on Oracle Linux UEK

To install xiRAID Classic 4.3.1 on Oracle Linux with UEK:

1. Install kernel-headers for the currently loaded kernel:

```
# dnf install kernel-uek-devel-$(uname -r)
```

2. Install xiraid-repo for your OS:

- Oracle 8.4 and Oracle 8.6:

```
# dnf install https://pkg.xinnor.io/repository/  
Repository/xiraid/oracle/8.4/kver-5.4/  
xiraid-repo-1.3.0-1588.kver.5.4.noarch.rpm
```

- Oracle 9:

```
# dnf install https://pkg.xinnor.io/repository/  
Repository/xiraid/oracle/9/kver-5.15/  
xiraid-repo-1.3.0-1588.kver.5.15.noarch.rpm
```

3. Install xiraid.

This command installs xiRAID Classic 4.3.1.

```
# dnf install xiraid-core
```

## Testing the Installation

To ensure that the installation was successful:

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the installation has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

3. Check the status of your license:

```
xicli license show
```

The first time you call the command after the installation, the system prompts you to accept the EULA conditions. If you accept the conditions, the command outputs your license details.



To request a trial license, please do not hesitate to contact us at [support@xinnor.io](mailto:support@xinnor.io).

4. Check the xiRAID Classic operability:

```
# xicli raid show
```

This command shows a table with RAIDs. If the output is not in the form of a table or if there is no output at all, it indicates that the installation was not successful.

5. Make sure that xiRAID Classic has been installed on the latest kernel version available on your system.

Otherwise, follow these steps to use xiRAID Classic after switching to the latest kernel version:

- a. Stop the xiraid target service:

```
# systemctl stop xiraid.target
```

- b. Boot into the desired kernel.
- c. Install the development kernel-headers package for the current kernel.
- d. Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

After the installation, you can find the installation logs at `/var/log/xraid/`.

If installation was not successful:

- check the kernel version;
- try to reinstall xiRAID Classic 4.3.1.

If this does not help, please contact Xinnor Support team at [support@xinnor.io](mailto:support@xinnor.io) with attached logs from `/var/log/xraid/`.


## Installing xiRAID Classic on Ubuntu

### Prerequisites

During the installation process, additional packages required by xiRAID Classic 4.3.1 are automatically installed on your system (see the list of additional packages in the document xiRAID Classic 4.3.1 System Requirements). To install additional packages correctly, make sure that the appropriate repositories are configured for them.




Please ensure that the Protobuf library for Python is not already installed on your system via pip before installing xiRAID Classic 4.3.1

 Please ensure that Secure Boot is disabled on your system.

The Version Lock plugin will be installed along with the other xiRAID packages. Once the installation is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (apt/yum/dnf update).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at <https://www.xinnor.io/>. We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module.

 However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

## Installation on Ubuntu

Information in this chapter is for the following systems:

- Ubuntu 20.04 LTS with 5.4 kernel;
- Ubuntu 22.04 LTS with 5.15 kernel;
- Ubuntu 24.04 LTS with 6.8 kernel.

To install xiRAID Classic 4.3.1 on Ubuntu:

1. Install linux-headers for the currently loaded kernel:

```
# apt update && apt install linux-headers-$(uname -r)
```

2. Change your current working directory to `/tmp`.

```
# cd /tmp/
```

3. Install the xiraid-repo package:

```
curl -O https://pkg.xinnor.io/repository/Repository/xiraid/  
ubuntu/multi-pack/xiraid-repo_1.3.0-1588.kver.6.8_amd64.deb  
apt install ./xiraid-repo_1.3.0-1588.kver.6.8_amd64.deb
```

4. Install xiraid:

```
# apt update && apt install xiraid-core
```

## Testing the Installation

To ensure that the installation was successful:

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the installation has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

3. Check the status of your license:

```
xicli license show
```

The first time you call the command after the installation, the system prompts you to accept the EULA conditions. If you accept the conditions, the command outputs your license details.



To request a trial license, please do not hesitate to contact us at [support@xinnor.io](mailto:support@xinnor.io).

#### 4. Check the xiRAID Classic operability:

```
# xicli raid show
```

This command shows a table with RAIDs. If the output is not in the form of a table or if there is no output at all, it indicates that the installation was not successful.

#### 5. Make sure that xiRAID Classic has been installed on the latest kernel version available on your system.

Otherwise, follow these steps to use xiRAID Classic after switching to the latest kernel version:

##### a. Stop the xiraid target service:

```
# systemctl stop xiraid.target
```

##### b. Boot into the desired kernel.

##### c. Install the development kernel-headers package for the current kernel.

##### d. Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

After the installation, you can find the installation logs at `/var/log/xraid/`.

If installation was not successful:

- check the kernel version;
- try to reinstall xiRAID Classic 4.3.1.

If this does not help, please contact Xinnor Support team at [support@xinnor.io](mailto:support@xinnor.io) with attached logs from `/var/log/xraid/`.

# Installing xiRAID Classic on Proxmox

## Prerequisites

During the installation process, additional packages required by xiRAID Classic 4.3.1 are automatically installed on your system (see the list of additional packages in the document xiRAID Classic 4.3.1 System Requirements). To install additional packages correctly, make sure that the appropriate repositories are configured for them.



Please ensure that the Protobuf library for Python is not already installed on your system via pip before installing xiRAID Classic 4.3.1



Please ensure that Secure Boot is disabled on your system.

The Version Lock plugin will be installed along with the other xiRAID packages. Once the installation is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (apt/yum/dnf update).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at <https://www.xinnor.io/>. We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module.



However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

## Installation on Proxmox



Before installing xiRAID Classic 4.3.1 on Proxmox, make sure you have a subscription for its package repositories.

To install xiRAID Classic 4.3.1 on Proxmox:

1. Install linux-headers for the currently loaded kernel:

```
# apt update && apt install pve-headers-$(uname -r)
```

2. Install xiraid-repo for your OS:

a. Proxmox 8.0, 8.1, 8.2, 8.3, 8.4:

```
# cd /tmp/  
# curl -O https://pkg.xinnor.io/repository/  
Repository/xiraid/proxmox/8/kver-6.2/  
xiraid-repo_1.3.0-1588.kver.6.2_amd64.deb  
# apt install ./xiraid-repo_1.3.0-1588.kver.6.2_amd64.deb
```

b. Proxmox 9.0:

```
# cd /tmp/  
# curl -O https://pkg.xinnor.io/repository/  
Repository/xiraid/proxmox/9/kver-6.14/  
xiraid-repo_1.3.0-1672.kver.6.14_amd64.deb  
# apt install ./xiraid-repo_1.3.0-1672.kver.6.14_amd64.deb
```

3. Install xiraid:

```
# apt update && apt install xiraid-core
```

## Testing the Installation

To ensure that the installation was successful:

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the installation has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

3. Check the status of your license:

```
xicli license show
```

The first time you call the command after the installation, the system prompts you to accept the EULA conditions. If you accept the conditions, the command outputs your license details.



To request a trial license, please do not hesitate to contact us at [support@xinnor.io](mailto:support@xinnor.io).

#### 4. Check the xiRAID Classic operability:

```
# xicli raid show
```

This command shows a table with RAIDs. If the output is not in the form of a table or if there is no output at all, it indicates that the installation was not successful.

#### 5. Make sure that xiRAID Classic has been installed on the latest kernel version available on your system.

Otherwise, follow these steps to use xiRAID Classic after switching to the latest kernel version:

##### a. Stop the xiraid target service:

```
# systemctl stop xiraid.target
```

##### b. Boot into the desired kernel.

##### c. Install the development kernel-headers package for the current kernel.

##### d. Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

After the installation, you can find the installation logs at `/var/log/xraid/`.

If installation was not successful:

- check the kernel version;
- try to reinstall xiRAID Classic 4.3.1.

If this does not help, please contact Xinnor Support team at [support@xinnor.io](mailto:support@xinnor.io) with attached logs from `/var/log/xraid/`.

## Uninstalling xiRAID Classic 4.3.1

Stop using the xiRAID Classic devices before uninstalling xiRAID Classic 4.3.1:

1. Disable all the applications that are using the xiRAID Classic devices.
2. Unmount all file systems that use the xiRAID Classic devices (using `umount`, `systemd` or other suitable tool) and disable automatic mounting of file systems at system startup.
3. Make sure all file systems that use the xiRAID devices are unmounted:

```
# df -h
```



Ignoring these steps may result in data loss.

These commands uninstall xiRAID Classic 4.3.1:

## 1. On RHEL, RHEL-based systems, and Oracle Linux:

```
# dnf remove xiraid-appimage xiraid-core xiraid-kmod && dnf
autoremove
# dnf remove xiraid-repo
```

## 2. On Ubuntu and Proxmox:

```
# apt remove xiraid-appimage xiraid-core xiraid-kmod
# apt remove xiraid-repo
# apt autoremove
```

# xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 Update Guide

This document provides the instructions on updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1.



If you have integrated xiRAID Classic 4.3.0 into a Pacemaker cluster, first refer to [Updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 \(HA\) \(Integrating xiRAID Classic 4.3.1 Into a Pacemaker Cluster\)](#) for additional details about the update procedure.

xiRAID Classic 4.3.1 currently supports Oracle, Ubuntu, Proxmox, Alma Linux, Rocky Linux and RHEL systems (refer to the [xiRAID Classic 4.3.1 System Requirements](#) for the full list of supported distributions). You can update xiRAID Classic 4.3.0 using the “xiraid-repo” repository (see updating instructions for your OS below).

During the update, additional packages required by xiRAID Classic 4.3.1 will be automatically installed on your system (see [xiRAID Classic 4.3.1 System Requirements](#)).

The Version Lock plugin will be installed along with the xiRAID packages. Once the update is finished, this plugin will lock the current version of the xiRAID packages, preventing it from being automatically updated on general system update commands (`apt/yum/dnf update`).

To update xiRAID Classic to a new available version, you must disable the Version Lock plugin as described in the xiRAID Classic Update Guide. The instructions on how to do that safely will be available at [xinnor.io](http://xinnor.io).

We recommend setting up email notifications to receive information about the latest xiRAID Classic releases. Detailed instructions on how to do this are listed in the xiRAID Classic Administrator's Guide. The notifications will be sent to you once every three days. The corresponding messages will be added to journalctl logs regardless of your notifications settings.

## Updating xiRAID Classic on RHEL and RHEL-based Systems

### Preparing for the Update

This section provides a list of steps that should be taken as a part of the update from xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Make sure your system meets the requirements listed in [xiRAID Classic 4.3.1 System Requirements](#).

2. Update you local package index and cached packages:

```
# dnf clean all
```

3. Make sure the new xiRAID Classic version is available for update:

```
# xicli update check
```

4. Ensure that the output of the command below does not contain errors. If errors are present in the output, contact our support team.

```
xicli settings log show
```

5. Stop using xiRAID devices:



Ignoring these steps may result in filesystem panic and even data loss.

- a. Disable all the applications that are using xiRAID devices.
- b. Unmount all file systems that use xiRAID devices (using `umount`, `systemd` or other suitable tool) and disable automatic mounting of file systems at system startup. For example, if you use the mounting approach described in the 'File System Mounting Aspects' section of the xiRAID Classic Administrator's Guide, do the following:

- if automatic mounting was enabled through `systemd.mount`, stop the service that mounts file systems for all xiRAID devices:

```
# systemctl stop mnt-raid.mount
```

Disable the automatic mounting service for all xiRAID devices at system startup:

```
# systemctl disable mnt-raid.mount
```

- if automatic mounting was enabled through `/etc/fstab`, unmount all file systems that use the xiRAID devices (using `umount` or other suitable tool):

```
# umount /dev/xi_<device_name>
```

Comment out the lines in the `/etc/fstab` file that are responsible for automatic mounting of the xiRAID devices. To do this, insert a comment symbol (`#`) before those lines. For example:

```
# /dev/xi_raidname    /mnt/raid/    xfs    defaults    0
0
```

- make sure all file systems that use the xiRAID devices are unmounted:

```
# df -h
```

6. Make sure that all RAID's are in one of the following states *online*, *initialized* (not applicable for RAID 0) , or *none*:

```
# xicli raid show
```

```

user@testvm:~$ xicli raid show
RAIDS
┌───┬───┬───┬───┬───┐
│ name │ static │ state │ devices │ info │
├───┬───┬───┬───┬───┤
│ example │ size: 19 GiB │ online │ 0 /dev/sdb online │ │
│ │ level: 0 │ │ 1 /dev/sdc online │ │
│ │ strip_size: 16 │ │ │ │ │
│ │ block_size: 4096 │ │ │ │ │
│ │ sparepool: - │ │ │ │ │
│ │ active: True │ │ │ │ │
│ │ config: True │ │ │ │ │
└───┬───┬───┬───┬───┘

```

#### Command output example

- Copy the xiRAID Classic configuration files to the home directory (or any other directory of your choice) as a backup in case the update does not complete successfully:

```
cp -r /etc/xiraid/raids/ ~/
```

xiRAID is installed on the active kernel version of your OS and supports regular kernel updates by automatically rebuilding its kernel module. However, if your system has multiple kernel versions installed and xiRAID is not installed on the latest one, it will not function correctly when switching to the latest kernel version. Make sure you are booted into the correct kernel version if you want to install xiRAID on it.

Otherwise, follow these steps to switch to the desired kernel version:



- Boot into the desired kernel.
- Install the development kernel-headers package for the current kernel.
- Run:

```
# dkms autoinstall
```

```
# systemctl restart xiraid.target
```

Once you have completed all the steps in this section, follow the instructions for your operating system to update xiRAID Classic 4.3.1:

## Performing the Update

Information in this chapter is for the following systems:

- RHEL & RHEL-based 8;
- RHEL & RHEL-based 9.

When updating xiRAID Classic 4.3.0 on RHEL (or a RHEL-based distribution) 9.0 to xiRAID Classic 4.3.1 on RHEL or a RHEL-based distribution version 9.1, 9.2, 9.3 and 9.4, the output may include the following warning: The error message displayed below may be shown during the update process in the following



```
Error! The /var/lib/dkms/xiraid/.../dkms.conf
for module xiraid includes a BUILD_EXCLUSIVE
directdoes not match this kernel/arch.
This indicates that it should not be built.
```

When installing Xinnor xiRAID on RHEL or RHEL-based 9\*, the latest kernel version will be installed. To revert your system to the kernel version used prior to installation, you need to change the default kernel that is launched. Specify the kernel version you want to return to (`kernel_version`):



```
# grubby --set-default "/boot/vmlinuz-<kernel_version>"
```

To update xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 on RHEL or a RHEL-based distribution:

1. Disable the Version Lock plugin using the Update Check Service (once the update is finished, the plugin will be automatically re-enabled to lock the version of the installed xiRAID packages):

```
# xicli update prepare
```

2. Update the xiraid package:

```
# dnf upgrade 'xiraid-*
```

## After updating

This section outlines the necessary steps that need to be taken after updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the update has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

The version number must be 4.3.1. Otherwise, the update has not completed successfully.

3. Check the xiRAID Classic operability:

```
# xicli raid show
```

The command outputs a list of created RAID configurations in the form of a table. All RAID configurations must be in the state *online* (RAID 0) or *online, initialized* (other RAID configurations). Otherwise, the update has not completed successfully.

```
user@testvm:~$ xicli raid show
RAIDs
┌───┬───┬───┬───┬───┐
│ name │ static │ state │ devices │ info │
├───┬───┬───┬───┬───┤
│ example │ size: 19 GiB │ online │ 0 /dev/sdb online │ │
│ │ level: 0 │ │ 1 /dev/sdc online │ │
│ │ strip_size: 16 │ │ │ │ │
│ │ block_size: 4096 │ │ │ │ │
│ │ sparepool: - │ │ │ │ │
│ │ active: True │ │ │ │ │
│ │ config: True │ │ │ │ │
└───┬───┬───┬───┬───┘
```

#### Command output example

4. Mount all file systems that use xiRAID devices and enable automatic mounting of file systems at system startup:

- a. Add the following line to the Install section of your systemd mount unit files:

```
WantedBy=dev-xi_raid.device
```

The `WantedBy` option specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `WantedBy` is `dev-xi_raid.device`.

The complete systemd unit file, including the new line, should look like this:

```
[Unit]
Description=Mount filesystem on xiRAID Classic
DefaultDependencies=no
Before=umount.target
Conflicts=umount.target

[Mount]
What=/dev/xi_raid
Where=/mnt/raid/
Options=defaults
Type=xfs

[Install]
WantedBy=multi-user.target
WantedBy=dev-xi_raid.device
```

- `/dev/xi_raid` is the device to mount.
- `/mnt/raid/` is the mount point.

Enable systemd mount units that were previously disabled:

```
# systemctl enable mnt-raid.mount
```

To manually start the mount service and mount the file system, run the following command:

```
# systemctl start mnt-raid.mount
```

b. In `/etc/fstab`, update your entries to match the following template:

```
/dev/xi_raid    /mnt/raid/    xfs
defaults,x-systemd.wanted-by=dev-xi_raid.device 0 0
```

- `/dev/xi_raid` is the device to mount. Use the block device name (e.g., `/dev/xi_raid`) rather than the UUID or LABEL.
- `/mnt/raid/` is the mount point.
- `x-systemd.wanted-by` specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `x-systemd.wanted-by` is `dev-xi_raid.device`.

Run the following command to mount all devices configured in `/etc/fstab`:

```
# mount -a
```

c. Reload the systemd daemon:

```
# systemctl daemon-reload
```

d. Make sure all file systems that use xiRAID devices are mounted:

```
# df -h
```

5. Enable all the applications that are using the xiRAID devices.

6. If the update was not successful, please contact xiRAID Classic Support Team at [support@xiraid.com](mailto:support@xiraid.com) with attached logs from `/var/log/xraid/setup.log`.

7. Optionally, configure reserved memory for your RAID's:

```
# xicli raid modify -n <raid_name> -mp 2048
```

where `<raid_name>` is the name of the RAID for which to allocate reserved memory and `2048` is the amount of memory to allocate, in MiB. Note that the specified amount must not exceed the available system memory minus 1 GiB, leaving at least 1 GiB free. For more details about changing RAID parameters, see this topic.

## Updating xiRAID Classic on Oracle Linux

### Preparing for the Update

This section provides a list of steps that should be taken as a part of the update from xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Make sure your system meets the requirements listed in [xiRAID Classic 4.3.1 System Requirements](#).

2. Update you local package index and cached packages:

```
# dnf clean all
```

3. Make sure the new xiRAID Classic version is available for update:

```
# xicli update check
```

4. Ensure that the output of the command below does not contain errors. If errors are present in the output, contact our support team.

```
xicli settings log show
```

5. Stop using xiRAID devices:



Ignoring these steps may result in filesystem panic and even data loss.

- a. Disable all the applications that are using xiRAID devices.
- b. Unmount all file systems that use xiRAID devices (using `umount`, `systemd` or other suitable tool) and disable automatic mounting of file systems at system startup. For example, if you use the mounting approach described in the 'File System Mounting Aspects' section of the xiRAID Classic Administrator's Guide, do the following:

- if automatic mounting was enabled through `systemd.mount`, stop the service that mounts file systems for all xiRAID devices:

```
# systemctl stop mnt-raid.mount
```

Disable the automatic mounting service for all xiRAID devices at system startup:

```
# systemctl disable mnt-raid.mount
```

- if automatic mounting was enabled through `/etc/fstab`, unmount all file systems that use the xiRAID devices (using `umount` or other suitable tool):

```
# umount /dev/xi_<device_name>
```

Comment out the lines in the `/etc/fstab` file that are responsible for automatic mounting of the xiRAID devices. To do this, insert a comment symbol (`#`) before those lines. For example:

```
# /dev/xi_raidname    /mnt/raid/    xfs    defaults    0
0
```

- make sure all file systems that use the xiRAID devices are unmounted:

```
# df -h
```

6. Make sure that all RAID's are in one of the following states *online*, *initialized* (not applicable for RAID 0) , or *none*:

```
# xicli raid show
```



## Performing the Update

Information in this chapter is for the following systems:

- Oracle 8.4 and 8.6;
- Oracle 9.

To update xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 on an Oracle system:

1. Disable the Version Lock plugin using the Update Check Service (once the update is finished, the plugin will be automatically re-enabled to lock the version of the installed xiRAID packages):

```
# xicli update prepare
```

2. Update the xiraid package:

```
# dnf upgrade 'xiraid-*'
```

## After updating

This section outlines the necessary steps that need to be taken after updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the update has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

The version number must be 4.3.1. Otherwise, the update has not completed successfully.

3. Check the xiRAID Classic operability:

```
# xicli raid show
```

The command outputs a list of created RAIDs in the form of a table. All RAIDs must be in the state “*online*” (RAID 0) or “*online, initialized*” (other RAID configurations). Otherwise, the update has not completed successfully.

```
user@testvm:~$ xicli raid show
```

| RAIDs   |  |        |  |      |
|---------|--|--------|--|------|
| name    | static   | state  | devices                                | info |
| example | size: 19 GiB<br>level: 0<br>strip_size: 16<br>block_size: 4096<br>sparepool: -<br>active: True<br>config: True | online | 0 /dev/sdb online<br>1 /dev/sdc online |      |

#### Command output example

4. Mount all file systems that use xiRAID devices and enable automatic mounting of file systems at system startup:

- a. Add the following line to the Install section of your systemd mount unit files:

```
WantedBy=dev-xi_raid.device
```

The `WantedBy` option specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `WantedBy` is `dev-xi_raid.device`.

The complete systemd unit file, including the new line, should look like this:

```
[Unit]
Description=Mount filesystem on xiRAID Classic
DefaultDependencies=no
Before=umount.target
Conflicts=umount.target

[Mount]
What=/dev/xi_raid
Where=/mnt/raid/
Options=defaults
Type=xfs

[Install]
WantedBy=multi-user.target
WantedBy=dev-xi_raid.device
```

- `/dev/xi_raid` is the device to mount.
- `/mnt/raid/` is the mount point.

Enable systemd mount units that were previously disabled:

```
# systemctl enable mnt-raid.mount
```

To manually start the mount service and mount the file system, run the following command:

```
# systemctl start mnt-raid.mount
```

b. In `/etc/fstab`, update your entries to match the following template:

```
/dev/xi_raid    /mnt/raid/    xfs
defaults,x-systemd.wanted-by=dev-xi_raid.device 0 0
```

- `/dev/xi_raid` is the device to mount. Use the block device name (e.g., `/dev/xi_raid`) rather than the UUID or LABEL.
- `/mnt/raid/` is the mount point.
- `x-systemd.wanted-by` specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `x-systemd.wanted-by` is `dev-xi_raid.device`.

Run the following command to mount all devices configured in `/etc/fstab`:

```
# mount -a
```

c. Reload the systemd daemon:

```
# systemctl daemon-reload
```

d. Make sure all file systems that use xiRAID devices are mounted:

```
# df -h
```

5. Enable all the applications that are using the xiRAID devices.

6. If the update was not successful, please contact xiRAID Classic Support Team at [support@xiraid.com](mailto:support@xiraid.com) with attached logs from `/var/log/xraid/setup.log`.

7. Optionally, configure reserved memory for your RAID's:

```
# xicli raid modify -n <raid_name> -mp 2048
```

where `<raid_name>` is the name of the RAID for which to allocate reserved memory and `2048` is the amount of memory to allocate, in MiB. Note that the specified amount must not exceed the available system memory minus 1 GiB, leaving at least 1 GiB free. For more details about changing RAID parameters, see this topic.

## Updating xiRAID Classic on Ubuntu

### Preparing for the Update

This section provides a list of steps that should be taken as a part of the update from xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Make sure your system meets the requirements listed in [xiRAID Classic 4.3.1 System Requirements](#).

2. Update you local package index and cached packages:

```
# apt update
```

3. Make sure the new xiRAID Classic version is available for update:

```
# xicli update check
```

4. Ensure that the output of the command below does not contain errors. If errors are present in the output, contact our support team.

```
xicli settings log show
```

5. Stop using xiRAID devices:



Ignoring these steps may result in filesystem panic and even data loss.

- a. Disable all the applications that are using xiRAID devices.
- b. Unmount all file systems that use xiRAID devices (using `umount`, `systemd` or other suitable tool) and disable automatic mounting of file systems at system startup. For example, if you use the mounting approach described in the 'File System Mounting Aspects' section of the xiRAID Classic Administrator's Guide, do the following:

- if automatic mounting was enabled through `systemd.mount`, stop the service that mounts file systems for all xiRAID devices:

```
# systemctl stop mnt-raid.mount
```

Disable the automatic mounting service for all xiRAID devices at system startup:

```
# systemctl disable mnt-raid.mount
```

- if automatic mounting was enabled through `/etc/fstab`, unmount all file systems that use the xiRAID devices (using `umount` or other suitable tool):

```
# umount /dev/xi_<device_name>
```

Comment out the lines in the `/etc/fstab` file that are responsible for automatic mounting of the xiRAID devices. To do this, insert a comment symbol (`#`) before those lines. For example:

```
# /dev/xi_raidname    /mnt/raid/    xfs    defaults    0
0
```

- make sure all file systems that use the xiRAID devices are unmounted:

```
# df -h
```

6. Make sure that all RAID's are in one of the following states *online*, *initialized* (not applicable for RAID 0) , or *none*:

```
# xicli raid show
```



## Performing the Update

Information in this chapter is for the following systems:

- Ubuntu 20.04;
- Ubuntu 22.04;
- Ubuntu 24.04.

During the update process, you may encounter the following warning messages:

```
KeyError: 'strip_size'
dpkg: error processing package xiraid-core (--configure):
 installed xiraid-core package post-installation script
 subprocess returned error
 exit status 1
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Errors were encountered while processing:
 xiraid-core
needrestart is being skipped since dpkg has failed
E: Sub-process /usr/bin/dpkg returned an error code (1)
```



These messages are expected and can be safely ignored. The update will continue, and you can proceed without any concerns.

To update xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 on an Ubuntu system:

1. Disable the Version Lock plugin using the Update Check Service (once the update is finished, the plugin will be automatically re-enabled to lock the version of the installed xiRAID packages):

```
# xicli update prepare
```

2. Update the xiraid package:

```
# apt update  
# apt install --only-upgrade 'xiraid-*
```

## After updating

This section outlines the necessary steps that need to be taken after updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the update has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

The version number must be 4.3.1. Otherwise, the update has not completed successfully.

3. Check the xiRAID Classic operability:

```
# xicli raid show
```

The command outputs a list of created RAID configurations in the form of a table. All RAID configurations must be in the state *online* (RAID 0) or *online, initialized* (other RAID configurations). Otherwise, the update has not completed successfully.

```
user@testvm:~$ xicli raid show
RAIDS
┌───┬───┬───┬───┬───┐
│ name │ static │ state │ devices │ info │
├───┬───┬───┬───┬───┤
│ example │ size: 19 GiB │ online │ 0 /dev/sdb online │ │
│ │ level: 0 │ │ 1 /dev/sdc online │ │
│ │ strip_size: 16 │ │ │ │ │
│ │ block_size: 4096 │ │ │ │ │
│ │ sparepool: - │ │ │ │ │
│ │ active: True │ │ │ │ │
│ │ config: True │ │ │ │ │
└───┬───┬───┬───┬───┘
```

#### Command output example

4. Mount all file systems that use xiRAID devices and enable automatic mounting of file systems at system startup:

- a. Add the following line to the Install section of your systemd mount unit files:

```
WantedBy=dev-xi_raid.device
```

The `WantedBy` option specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `WantedBy` is `dev-xi_raid.device`.

The complete systemd unit file, including the new line, should look like this:

```
[Unit]
Description=Mount filesystem on xiRAID Classic
DefaultDependencies=no
Before=umount.target
Conflicts=umount.target

[Mount]
What=/dev/xi_raid
Where=/mnt/raid/
Options=defaults
Type=xfss

[Install]
WantedBy=multi-user.target
WantedBy=dev-xi_raid.device
```

- `/dev/xi_raid` is the device to mount.
- `/mnt/raid/` is the mount point.

Enable systemd mount units that were previously disabled:

```
# systemctl enable mnt-raid.mount
```

To manually start the mount service and mount the file system, run the following command:

```
# systemctl start mnt-raid.mount
```

b. In `/etc/fstab`, update your entries to match the following template:

```
/dev/xi_raid    /mnt/raid/    xfs
defaults,x-systemd.wanted-by=dev-xi_raid.device 0 0
```

- `/dev/xi_raid` is the device to mount. Use the block device name (e.g., `/dev/xi_raid`) rather than the UUID or LABEL.
- `/mnt/raid/` is the mount point.
- `x-systemd.wanted-by` specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `x-systemd.wanted-by` is `dev-xi_raid.device`.

Run the following command to mount all devices configured in `/etc/fstab`:

```
# mount -a
```

c. Reload the systemd daemon:

```
# systemctl daemon-reload
```

d. Make sure all file systems that use xiRAID devices are mounted:

```
# df -h
```

5. Enable all the applications that are using the xiRAID devices.

6. If the update was not successful, please contact xiRAID Classic Support Team at [support@xiraid.com](mailto:support@xiraid.com) with attached logs from `/var/log/xraid/setup.log`.

7. Optionally, configure reserved memory for your RAID:

```
# xicli raid modify -n <raid_name> -mp 2048
```

where `<raid_name>` is the name of the RAID for which to allocate reserved memory and `2048` is the amount of memory to allocate, in MiB. Note that the specified amount must not exceed the available system memory minus 1 GiB, leaving at least 1 GiB free. For more details about changing RAID parameters, see this topic.

## Updating xiRAID Classic on Proxmox

### Preparing for the Update

This section provides a list of steps that should be taken as a part of the update from xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Make sure your system meets the requirements listed in [xiRAID Classic 4.3.1 System Requirements](#).

2. Update you local package index and cached packages:

```
# apt update
```

3. Make sure the new xiRAID Classic version is available for update:

```
# xicli update check
```

4. Ensure that the output of the command below does not contain errors. If errors are present in the output, contact our support team.

```
xicli settings log show
```

5. Stop using xiRAID devices:



Ignoring these steps may result in filesystem panic and even data loss.

- a. Disable all the applications that are using xiRAID devices.
- b. Unmount all file systems that use xiRAID devices (using `umount`, `systemd` or other suitable tool) and disable automatic mounting of file systems at system startup. For example, if you use the mounting approach described in the 'File System Mounting Aspects' section of the xiRAID Classic Administrator's Guide, do the following:

- if automatic mounting was enabled through `systemd.mount`, stop the service that mounts file systems for all xiRAID devices:

```
# systemctl stop mnt-raid.mount
```

Disable the automatic mounting service for all xiRAID devices at system startup:

```
# systemctl disable mnt-raid.mount
```

- if automatic mounting was enabled through `/etc/fstab`, unmount all file systems that use the xiRAID devices (using `umount` or other suitable tool):

```
# umount /dev/xi_<device_name>
```

Comment out the lines in the `/etc/fstab` file that are responsible for automatic mounting of the xiRAID devices. To do this, insert a comment symbol (`#`) before those lines. For example:

```
# /dev/xi_raidname /mnt/raid/ xfs defaults 0
0
```

- make sure all file systems that use the xiRAID devices are unmounted:

```
# df -h
```

6. Make sure that all RAID's are in one of the following states *online*, *initialized* (not applicable for RAID 0), or *none*:

```
# xicli raid show
```



## Performing the Update

Information in this chapter is for the following systems:

- Proxmox 8.0;
- Proxmox 8.1;
- Proxmox 8.2;
- Proxmox 8.3;
- Proxmox 8.4;
- Proxmox 9.0.

To update xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1 on Proxmox:

1. Disable the Version Lock plugin using the Update Check Service (once the update is finished, the plugin will be automatically re-enabled to lock the version of the installed xiRAID packages):

```
# xicli update prepare
```

2. Update the xiraid package:

```
# apt update  
# apt install --only-upgrade 'xiraid-*
```

## After updating

This section outlines the necessary steps that need to be taken after updating xiRAID Classic 4.3.0 to xiRAID Classic 4.3.1. These steps are applicable to all systems.

1. Check the xiraid module:

```
# lsmod | grep xiraid
```

The command shows if the xiraid module is loaded. Empty output indicates that the update has not completed successfully.

2. Check the xiRAID Classic version:

```
# xicli -v
```

The version number must be 4.3.1. Otherwise, the update has not completed successfully.

### 3. Check the xiRAID Classic operability:

```
# xicli raid show
```

The command outputs a list of created RAIDs in the form of a table. All RAIDs must be in the state *“online”* (RAID 0) or *“online, initialized”* (other RAID configurations). Otherwise, the update has not completed successfully.

```
user@testvm:~$ xicli raid show
RAIDs
┌───┬───┬───┬───┬───┐
│ name │ static │ state │ devices │ info │
├───┬───┬───┬───┬───┤
│ example │ size: 19 GiB  
level: 0  
strip_size: 16  
block_size: 4096  
sparepool: -  
active: True  
config: True │ online │ 0 /dev/sdb online  
1 /dev/sdc online │ │
└───┬───┬───┬───┬───┘
```

Command output example

### 4. Mount all file systems that use xiRAID devices and enable automatic mounting of file systems at system startup:

- a. Add the following line to the Install section of your systemd mount unit files:

```
WantedBy=dev-xi_raid.device
```

The `wantedBy` option specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `wantedBy` is `dev-xi_raid.device`.

The complete systemd unit file, including the new line, should look like this:

```
[Unit]
Description=Mount filesystem on xiRAID Classic
DefaultDependencies=no
Before=umount.target
Conflicts=umount.target

[Mount]
What=/dev/xi_raid
Where=/mnt/raid/
Options=defaults
Type=xf

[Install]
WantedBy=multi-user.target
WantedBy=dev-xi_raid.device
```

- `/dev/xi_raid` is the device to mount.
- `/mnt/raid/` is the mount point.

Enable systemd mount units that were previously disabled:

```
# systemctl enable mnt-raid.mount
```

To manually start the mount service and mount the file system, run the following command:

```
# systemctl start mnt-raid.mount
```

b. In `/etc/fstab`, update your entries to match the following template:

```
/dev/xi_raid    /mnt/raid/    xfs
defaults,x-systemd.wanted-by=dev-xi_raid.device 0 0
```

- `/dev/xi_raid` is the device to mount. Use the block device name (e.g., `/dev/xi_raid`) rather than the UUID or LABEL.
- `/mnt/raid/` is the mount point.
- `x-systemd.wanted-by` specifies when to mount the device. The value should be the device name with `dev-` prepended and `.device` appended. For example, if the device is `/dev/xi_raid`, the corresponding value for `x-systemd.wanted-by` is `dev-xi_raid.device`.

Run the following command to mount all devices configured in `/etc/fstab`:

```
# mount -a
```

c. Reload the systemd daemon:

```
# systemctl daemon-reload
```

d. Make sure all file systems that use xiRAID devices are mounted:

```
# df -h
```

5. Enable all the applications that are using the xiRAID devices.

6. If the update was not successful, please contact xiRAID Classic Support Team at [support@xiraid.com](mailto:support@xiraid.com) with attached logs from `/var/log/xraid/setup.log`.

7. Optionally, configure reserved memory for your RAIDs:

```
# xicli raid modify -n <raid_name> -mp 2048
```

where `<raid_name>` is the name of the RAID for which to allocate reserved memory and `2048` is the amount of memory to allocate, in MiB. Note that the specified amount must not exceed the available system memory minus 1 GiB, leaving at least 1 GiB free. For more details about changing RAID parameters, see this topic.