Upgrading from SLE HA 15 to SLE HA 16.1

Upgrading from SLE HA 15 to SLE HA 16.1 requires a *cluster offline upgrade*. SLE HA 16 includes major changes that are backward incompatible with SLE HA 15, so a cluster *rolling* upgrade isn't possible. In a cluster offline upgrade, you must stop the cluster services on all nodes at once and upgrade the whole cluster before bringing the nodes back online.

Prerequisites

- A system that offers a supported upgraded path to SLE HA 16.1 (see *Section 2, "Supported upgrade paths"*).
- The system must be registered to access the package repositories.

Important: Maintenance window

The time of the upgrade itself depends on the system's performance, the installation size and your network connection speed. The cluster resources will be stopped for the duration of the upgrade.

Thoroughly test this procedure in a test environment before performing it in a production environment.

Publication Date: 06 Nov 2026

Contents

1 Lifecycle 3

2 Supported upgrade paths 3

3 The distribution migration system 4

4 Preparing the upgrade 6

5 Running the High Availability pre-upgrade check 7

6 Stopping the High Availability cluster services 8

7 Customizing the upgrade process 9

8 Performing the upgrade 11

9 Finishing the upgrade 13

10 Starting the High Availability cluster services 15

11 Resolving post-upgrade issues in the High Availability cluster 16

12 For more information 16

13 Legal Notice 17

A GNU Free Documentation License 17

HA glossary 25

1 Lifecycle

This section explains the SUSE Linux Enterprise High Availability lifecycle and when you have to upgrade. Whether it is better to upgrade or perform a fresh installation depends on your specific scenario.

1.1 Product lifecycle

SUSE Linux Enterprise High Availability 16 uses minor versions like 16.0 and 16.1 to add new features or functionality that remain backward-compatible. Minor releases are scheduled annually and always in November. Each minor version has 24 months of general support, giving you 12 months overlap with the next minor version. The last minor version, 16.6, will have 48 months of general support to complete a 10-year lifecycle in November 2035. Additional Long Term Support (LTS) is available for 3 years for all minor versions.

For more information, refer to the product lifecycle support policies page at https://www.suse.com/support/policy-products/. For exact lifecycle dates, refer to https://www.suse.com/lifecycle.

1.2 Upgrading versus fresh installation

Upgrades between two major versions of SUSE Linux Enterprise High Availability are supported by SUSE. Whether it is better to upgrade or perform a fresh installation depends on your specific scenario. While upgrades involve less work, fresh installations ensure you benefit from all the new features of a release, such as disk layout changes, specific file system features, and other improvements. To get the most out of your system, SUSE therefore recommends fresh installations in most scenarios.

In both cases—an upgrade and a fresh installation—customers need to check if system settings and default values still fit their requirements.

2 Supported upgrade paths

This section lists the upgrade paths for SUSE Linux Enterprise High Availability 16.1.

SUSE Linux Enterprise High Availability 16.1 allows upgrades from the following products:

- SLE HA 15 SP5, SP6, and SP7
- SLE HA 16.0

Other products and versions are not supported. Refer to the documentation of your product to upgrade it to a supported version first.

3 The distribution migration system

The distribution migration system provides an upgrade path for an installed SUSE Linux Enterprise High Availability system from one major version to another, for example, from SLE HA 15 SP7 to SLE HA 16.1. No new installation is required. Existing data, such as home and data directories and system configuration, is kept intact.

The distribution migration system uses a specialized upgrade live image to completely isolate the upgrade environment. This significantly reduces the risk of system inconsistencies or unrecoverable failures during major version upgrades.

3.1 Concept

While previous versions of SUSE Linux Enterprise High Availability were upgraded from the Unified Installer, SLE HA 16 uses the distribution migration system. This upgrade is done via the network using the **`zypper migration`** workflow, which sends a request to the repository server, asking for an upgrade path. Supported repository servers are the SUSE Customer Center (SCC) and the Repository Mirroring Tool (RMT). The request response contains the list of repositories required to upgrade the system. This requires the system to be registered before the upgrade.

The repository server providing the updates must have the necessary channels available and those channels must be up to date. This requirement is automatically met when a system is registered to the SUSE Customer Center (SCC) or for "Pay as you go" instances in the public cloud that are automatically registered to the SUSE-operated update infrastructure of the cloud service provider. However, administrative work may be required when the system to be upgraded is connected to an RMT server.

Upgrading to a new major version requires the system to be offline. This downtime is necessary to avoid system inconsistencies that could put the system into a state that does not allow recovery. Therefore, an upgrade live image is used for the upgrade.

The distribution migration system provides the live image and a start-up utility, **`run_migration`**, which reboots the running system into the upgrade live image. Once booted into the upgrade live image, the following chain of services will be executed:

1. Detect the system to be upgraded
2. Mount the necessary file systems
3. Set up the network to match the network configuration of the system to be upgraded
4. Prepare SSH access to the upgrade live image
5. Prepare the package manager for the upgrade task
6. Upgrade the system using **`zypper`**
7. Update the boot loader configuration
8. Unmount all mounted file systems
9. Reboot

If an error occurs prior to the start of the upgrade, the system will be reverted to its original state.

3.2 Upgrade Pre-Checks

The `suse-migration-pre-checks` package contains the script `suse-migration-pre-checks` that checks for possible incompatibilities for performing a migration. For example:

- Encrypted file systems
- Invalid repository types
- Multiple installed kernels with the `multiversion.kernels` option
- Unsupported upgrade paths (see *Section 2, "Supported upgrade paths"*)

This script is run during the installation of the `SLES16-Migration` package on the system to be upgraded. The pre-checks can also be run anytime using the following command:

```
> sudo suse-migration-pre-checks
```

The script must be run as root.

Running the script with the `--fix` option will automatically remediate the following issues:

- Set `multiversion.kernels` to the correct value and remove all old (not currently running) kernels.

4 Preparing the upgrade

Before starting the upgrade procedure, make sure your system is properly prepared. Among other things, preparation involves backing up data and checking the release notes.

Perform the following steps before starting the upgrade:

1. Make sure the latest system updates are installed by running **`zypper patch`**.
2. Read the release notes. Find the list of all changes, new features and known issues in the release notes (https://www.suse.com/releasenotes/).
   Consult the release notes to check whether the following applies:
   - Your hardware needs special considerations
   - Any currently used software packages have changed significantly
   - Your installation requires special precautions
3. Back up your data by copying the existing configuration files to a separate medium (such as a tape device or removable hard disk). This primarily applies to files stored in `/etc` and some directories and files in `/var` and `/opt`.
4. Check the available disk space. If you suspect you are running short of disk space, free up disk space or resize partitions.
5. If your machine serves as a VM Host Server for KVM, properly shut down all running VM Guests prior to the upgrade.
6. Configure SSH key-based login. During the upgrade, it is not possible to login with a password. It is recommended to that at least one of the users on the system has a private key in `~/.ssh/authorized_keys`. To add the key, run:

   ```
   > ssh-copy-id INSTANCE_USER@INSTANCE_IP
   ```

7. Root login by SSH will be disabled by default in SLE 16. You might want to enable it back before starting the migration. To do so, create `/etc/ssh/sshd_config.d/50-permit-root-login.conf` with the following content:

   ```
   PermitRootLogin yes
   ```

   Once the migration is complete, you can install `openssh-server-config-rootlogin` package and remove the previously added file.

5 Running the High Availability pre-upgrade check

Use the cluster health check tool to make sure the High Availability cluster can safely upgrade from SLE HA 15 to SLE HA 16. The tool reports any issues that will prevent the upgrade, as well as issues that can be fixed during or after the upgrade.

You can perform this procedure on any node in the cluster.

1. Log in to the node either as the `root` user or as a user with **`sudo`** privileges.
2. Run the cluster health check:

   ```
   > sudo crm cluster health sles16
   ```

   This command reports issues in two categories:

   - The `WARN` category indicates cluster configuration that isn't recommended, such as deprecated features. Issues in this category can be fixed during or after the upgrade, so the migration check passes. For example:

     ```
     [WARN] Corosync transport "udpu" will be deprecated in favor of "knet" in
      corosync 3.
     ```

   - The `FAIL` category indicates cluster configuration that can't be fixed during or after the upgrade, such as removed resource agents. Issues in this category must be fixed before you can upgrade the cluster, so the migration check fails. The command's output will tell you what to fix. For example:

     ```
     [FAIL] The following resource agents will be removed in SLES 16.
            * ocf:heartbeat:IPaddr: please replace it with ocf:heartbeat:IPaddr2
     ```

3. Fix any `FAIL` issues, then run the cluster health check again to make sure you didn't miss anything.

FOR MORE INFORMATION

- `crm cluster help health`

6 Stopping the High Availability cluster services

Stopping the cluster services means stopping Pacemaker and Corosync. You can stop the whole cluster at once by using the `--all` option. In this mode, the cluster is *no longer running* and all the resources stop.

For a cluster offline upgrade, the resources must *not* be in maintenance mode.



Warning: Graceful shutdown not guaranteed

When you stop the cluster services for the whole cluster without maintenance mode, all the resources stop. However, the resources must finish stopping before the cluster's shutdown timeout. If stopping a resource fails or times out, the resource's node is fenced.

If any resources take a long time to stop, consider stopping them *before* you stop the cluster services.

You can perform this procedure on any node in the cluster.

1. Log in either as the `root` user or as a user with **`sudo`** privileges.
2. Check the status of the cluster:

```
> sudo crm status
```

   - If the resources are in maintenance mode, they will continue to run even while the cluster services are offline.

     

     This is only true if the nodes keep running. If you need to shut down or reboot the nodes, take the cluster out of maintenance mode first.

   - If the resources aren't in maintenance mode, they will all stop when you stop the cluster services.

3. Stop the cluster services on all nodes at once:

```
> sudo crm cluster stop --all
```

4. Check the status of the cluster services:

```
> sudo crm cluster status
```

Both Pacemaker and Corosync should be `inactive`.

7 Customizing the upgrade process

The upgrade live image is preconfigured to run without any further setup. This section describes the custom configuration options.

To configure the distribution migration system, create `/etc/sle-migration-service.yml` on the system to be upgraded:

```
> sudo ssh USER@HOST 'touch /etc/sle-migration-service.yml'
```

7.1 Specify the upgrade target

The default upgrade target is SLES 16.0. This default target can be changed through the `migration_product` setting. The target must be specified with the triplet `name/version/arch` found in `/etc/products.d/baseproduct` of the target product, for example:

```
migration_product: SLES/16.1/x86_64
```

To change the upgrade target to 16.1, run the following command:

```
> sudo ssh USER@HOST 'echo "migration_product: SLES/16.1/x86_64" >> /etc/sle-migration-service.yml'
```

7.2 Optional configuration of the upgrade process

The following configuration options are completely optional. Change only what is necessary and only if you know what you are doing.

Preserve system data

Preserve custom data files, such as udev rules, from the system being migrated and ensure they are applied in the upgraded system.

The `preserve` section has three subsections that govern file preservation and system actions:

- `static`: Files in this subsection are copied into the DMS directly, with no further processing.
- `rules`: If this subsection contains files, they are preserved, and the DMS reloads udev to make these rules effective.
- `sysctl`: Preserving these files triggers sysctl --system to apply the configuration changes.

```
preserve:
    rules:
      - /etc/udev/rules.d/a.rules
      - /etc/udev/rules.d/b.rules
    static:
      - /etc/sysconfig/proxy
      - /path/to/be/preserved/*.suffix
```



Note

udev rules that require custom drivers will not have the desired effect as the upgrade life image does not include these drivers and therefore execution of those rules will fail. Rules with such properties should not be listed.



Note

The DMS provides a set of default preservable files that vary based on the target version and architecture. User-defined values will supplement this default list.

Enable debug mode

If enabled, it prevents the upgrade system from rewinding the setup steps and rebooting due to a failed upgrade, allowing the issue to be debugged.

```
debug: true|false
```

Configure the reboot method

By default, the migration system uses `kexec` to boot back into the host system once migration is complete. If this is in any way problematic, a regular `reboot` can be requested by setting `soft_reboot: false`.

```
soft_reboot: true|false
```

Enable the verbosity of the zypper migration

If enabled, it will run the zypper migration plugin with increased verbosity.

```
verbose_migration: true|false
```

Enable the fix option for pre_checks

If enabled (default), the run_pre_checks systemd process will use the `--fix` option to automatically remediate applicable issues before the migration is started.

```
pre_checks_fix: true|false
```

Configure the initramfs creation method

The live system may not contain all the tools required to create an initramfs that meets the needs of the system being upgraded. Building a host-independent initramfs will create an initramfs that includes the tools and modules available on the system being upgraded. If needed, a host-independent initramfs can be created by setting `build_host_independent_initrd: True`.

```
build_host_independent_initrd: true|false
```

Configure the dependency solver test case generation

It is possible that during the migration packages get installed that were not on the system previously and are pulled in because of dependencies. This setting will set up the migration so that a solver test case is generated. The information from the test case can then be used to understand why a given package was installed.

```
debug_solver: true|false
```

8 Performing the upgrade

This section describes the actual upgrade procedure with the distribution migration system.

8.1 Requirements

- The cluster services must be stopped on all nodes. Run **`crm cluster status`** to make sure both Pacemaker and Corosync are `inactive`.
- The system must be registered with the SUSE Customer Center or an RMT server. "Pay as you go" instances in the public cloud are automatically registered to the SUSE-operated update infrastructure.
- The upgrade path must be supported. Refer to *Section 2, "Supported upgrade paths"* for details.

- The upgrade target is set to 16.1 instead of the default 16.0. Refer to *Section 7, "Customizing the upgrade process"* for details.
- The server that provides the migration target must have the appropriate repositories synced, and they must be up to date. This requirement is automatically met by the SUSE Customer Center, and the SUSE-operated update infrastructure in the public cloud. If you are using an RMT server, make sure to mirror all relevant channels.
- The operating system needs to reside on a single partition. Support for multiple partitions, such as LVM, is limited to configurations where the separated partitions do not contain OS-critical data or processes. For example, a separate `/home` partition will function whereas a separate `/var` will not.

8.2 Upgrading SUSE Linux Enterprise High Availability 15 to 16.1

The distribution migration system can be invoked using two different methods:

- Reboot after installing the `suse-migration-sle16-activation` package.
- Run the **`run_migration`** command included with the `SLES16-Migration` package.

The distribution migration system upgrades both the base operating system and the High Availability cluster. Perform the upgrade on all the cluster nodes. The upgraded nodes won't be able to connect to non-upgraded nodes.

Depending on your architecture, perform the following steps to trigger the upgrade:

Use `reboot` on AMD64/Intel 64, AArch64, and ppc64le:

After installing the `suse-migration-sle16-activation` package, start the migration process by rebooting the system:

```
> sudo zypper in suse-migration-sle16-activation
> sudo reboot
```

The `suse-migration-sle16-activation` package installs `SLES16-Migration` as a dependency and modifies the bootloader configuration to boot into the upgrade image.

Use `run_migration` on s390x:

Modifying the boot loader configuration is not supported on s390x, so you start the migration directly with **`run_migration`**.

```
> sudo zypper in SLES16-Migration
> sudo run_migration
```

The **run_migration** utility uses **Kexec** and does not work in Xen-based environments. If **Kexec** causes a kernel panic, refer to the TID at https://www.suse.com/support/kb/doc/?id=000019733 and set the `soft_reboot` customization option (see *Section 7, "Customizing the upgrade process"*).

After the upgrade, the High Availability cluster services will start automatically during system boot (if the Pacemaker service is enabled). The cluster resources will start again when the upgraded cluster stack is back online and reaches a quorate state.

9 Finishing the upgrade

After the upgrade, you need to perform some additional tasks. The following section guides you through these steps.

1. Rebuild your **rpm** database. During or after the upgrade, you may see a warning similar to this one:

```
warning: Found NDB Packages.db database while attempting bdb backend: using ndb
 backend.
```

   This warning is purely informational and can be ignored. To convert the **rpm** database, run:

```
> sudo rpm --rebuilddb
```

2. Use **zypper packages** to check for orphaned and unneeded packages.

   a. Orphaned packages are no longer available in any of the configured package repositories. They can no longer get updated and become unsupported.
   For a list of orphaned packages, run:

```
> zypper packages --orphaned
```

   b. Unneeded packages are dependencies of packages that have been installed either explicitly by the user or implicitly as part of a pattern or product, and that have been removed in the meantime. They are usually no longer needed and should be removed, too.
   For a list of unneeded packages, run:

```
> zypper packages --unneeded
```



Tip

To avoid unneeded packages, use **`zypper rm`** with the `--clean-deps` option.

c. You can combine both lists into one:

```
> zypper packages --orphaned --unneeded
```

d. Use these lists to determine which packages are still needed and which can be safely removed.



Warning: Do not remove packages you need

If packages are renamed or removed from a pattern or product, **`zypper`** may no longer consider them explicitly installed and mark them as unneeded, even though they are still crucial for your installation.

Carefully review the list of packages you are removing.

To remove all orphaned and unneeded packages with a single command, run:

```
> sudo zypper rm $(zypper --no-refresh packages --orphaned --unneeded | gawk
 '{print $5}' | tail -n +5)
```

Exclude a single package or pattern from being uninstalled:

```
> sudo zypper rm $(zypper --no-refresh packages --orphaned --unneeded | gawk
 '{print $5}' | tail -n +5 | grep -v PACKAGE_TO_EXCLUDE)
```

Exclude multiple packages defined in a text file, separated by a newline:

```
> sudo zypper rm $(zypper --no-refresh packages --orphaned --unneeded | gawk
 '{print $5}' | tail -n +5 | grep -v -f /PACKAGES/TO/KEEP.txt)
```

3. Check for any `*.rpmnew` and `*.rpmsave` files. When an upgrade includes changes to a default configuration file that has been altered after package installation, instead of overwriting the file, one of these file types is created. While `*.rpmnew` contains the new default configuration and leaves your altered file untouched, `*.rpmsave` is a copy of your altered configuration that has been replaced by the new default file.

If you find any of these files, examine their content and merge desirable changes. You do not need to search the whole file system, only the `/etc` directory. Use the following command:

```
> find /etc/ -name "*.rpmnew" -o -name "*.rpmsave"
```

10 Starting the High Availability cluster services

If the Pacemaker service is enabled, the High Availability cluster services will start automatically during system boot. The cluster resources will start again when the upgraded cluster stack is back online and reaches a quorate state.

If the Pacemaker service is disabled, you can start the cluster services for the whole cluster at once by using the `--all` option.

You can perform this procedure on any node in the cluster.

1. Log in either as the `root` user or as a user with **`sudo`** privileges.
2. Start the cluster services on all nodes at once:

   ```
   > sudo crm cluster start --all
   ```

3. Check the status of the cluster services:

   ```
   > sudo crm cluster status
   ```

   Both Corosync and Pacemaker should be `active`.

4. Check the status of the cluster:

   ```
   > sudo crm status
   ```

   The nodes will have the status `UNCLEAN (offline)`, but will soon change to `Online`.

11 Resolving post-upgrade issues in the High Availability cluster

Use the cluster health check tool after the upgrade to list issues that still need to be fixed. You can also use the `--fix` option to automatically fix issues. Issues that can't be fixed automatically are listed along with suggestions for how to fix them manually.

PREREQUISITES

- All cluster nodes are reachable.
- All cluster nodes have been upgraded.

You can perform this procedure on any node in the cluster.

1. Log in to the node either as the `root` user or as a user with **`sudo`** privileges.
2. Run the cluster health check with the `--fix` option to automatically fix issues:

   ```
   > sudo crm cluster health sles16 --fix
   ```

   The output also lists any issues that can't be fixed automatically and tells you how to fix them manually. For example:

   ```
   [WARN] The CIB is not validated with the latest schema version.
          * Latest version: 4.0
          * Current version: 3.10
          Please run "crm configure upgrade force" to upgrade to the latest version.
   ```

3. Fix any issues, then run the cluster health check again to make you didn't miss anything.

FOR MORE INFORMATION

- **`crm cluster help health`**

12 For more information

For more information about the Distribution Migration System, refer to https://documentation.suse.com/suse-distribution-migration-system/15/html/distribution-migration-system/.

13 Legal Notice

Copyright© 2006– 2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see https://www.suse.com/company/legal/. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

A GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image for-

mats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally

prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

HA glossary

ACL (access control list)

Access control lists can be used to limit or fine-tune a user's access to the CIB. An ACL consists of an ordered set of access rules. Each rule allows or denies read or write access to a part of the CIB. Rules can be combined into specific roles, and users can be assigned to a role that matches their tasks.

active/active, active/passive

How resources run on the nodes. Active/passive means that resources only run on the active node, but can move to the passive node if the active node fails. Active/active means that all nodes are active at once, and resources can run on (and move to) any node in the cluster.

arbitrator

An *arbitrator* is a machine running outside the cluster to provide an additional instance for cluster calculations. For example, *QNetd* provides a vote to help *QDevice* participate in *quorum* decisions.

CIB (cluster information base)

An XML representation of the whole cluster configuration and status (cluster options, nodes, resources, constraints, etc.). The CIB manager (`pacemaker-based`) keeps the CIB synchronized across the cluster nodes and handles requests to modify it.

clone

In the context of a cluster *resource*, a clone is a resource that can be active on multiple nodes. Any resource can be cloned if its resource agent supports it.

cluster

A *high-availability* cluster is a group of servers (physical or virtual) designed primarily to secure the highest possible availability of data, applications and services. Not to be confused with a *high-performance* cluster, which shares the application load to achieve faster results.

Cluster LVM (Cluster logical volume manager)

The term *Cluster LVM* indicates that LVM is being used in a cluster environment. This requires configuration adjustments to protect the LVM metadata on shared storage.

cluster offline upgrade

If a new product version includes major changes that are backward incompatible, the cluster requires a cluster offline upgrade. You must stop the cluster services on all nodes and upgrade the whole cluster before bringing the nodes back online.

For minor version upgrades, see *cluster rolling upgrade*.

cluster partition

A cluster partition occurs when communication fails between one or more nodes and the rest of the cluster. The nodes are split into partitions but are still active. They can only communicate with nodes in the same partition and are unaware of the separated nodes. This is known as a *split brain* scenario.

cluster rolling upgrade

In a cluster rolling upgrade, one cluster node at a time is upgraded while the rest of the cluster is still running. Stop the cluster services on a node, upgrade it, then bring it back online to join the cluster. Continue one by one until all cluster nodes are upgraded.

For major version upgrades, see *cluster offline upgrade*.

cluster services

The services that run the cluster. Typically this refers to Pacemaker and Corosync, but might also include SBD and QDevice if those services are used.

cluster stack

The ensemble of software technologies and components that make up a cluster.

colocation constraint

A type of *resource constraint* that specifies which resources can or cannot run together on a node.

concurrency violation

A resource that should be running on only one node in the cluster is running on several nodes.

Corosync

Corosync provides reliable messaging, membership and quorum information about the cluster. This is handled by the Corosync Cluster Engine, a group communication system.

CRM (cluster resource manager)

The management entity responsible for coordinating all non-local interactions in a High Availability cluster. SUSE Linux Enterprise High Availability uses *Pacemaker* as the CRM. It interacts with several components: local executors on its own node and on the other nodes, non-local CRMs, administrative commands, the fencing functionality, and the membership layer.

`crmsh` (CRM Shell)

The command-line utility `crmsh` manages the cluster, nodes and resources.

Csync2

A synchronization tool for replicating configuration files across all nodes in the cluster.

DC (designated coordinator)

The `pacemaker-controld` daemon is the cluster controller, which coordinates all actions. This daemon has an instance on each cluster node, but only one instance is elected to act as the DC. The DC is elected when the cluster services start, or if the current DC fails or leaves the cluster. The DC decides whether a cluster-wide change must be performed, such as fencing a node or moving resources.

disaster

An unexpected interruption of critical infrastructure caused by nature, humans, hardware failure, or software bugs.

disaster recovery

The process by which a function is restored to the normal, steady state after a disaster.

Disaster Recovery Plan

A strategy to recover from a disaster with the minimum impact on IT infrastructure.

DLM (Distributed Lock Manager)

DLM coordinates accesses to shared resources in a cluster, for example, managing file locking in clustered file systems to increase performance and availability.

DRBD

DRBD® is a block device designed for building High Availability clusters. It replicates data on a primary device to secondary devices in a way that ensures all copies of the data remain identical.

existing cluster

The term *existing cluster* is used to refer to any cluster that consists of at least one node. An existing cluster has a basic *Corosync* configuration that defines the communication channels, but does not necessarily have resource configuration yet.

failover

Occurs when a resource or node fails on one machine and the affected resources move to another node.

failover domain

A named subset of cluster nodes that are eligible to run a resource if a node fails.

fencing

Prevents access to a shared resource by isolated or failing cluster members. There are two classes of fencing: *resource-level* fencing and *node-level* fencing. Resource-level fencing ensures exclusive access to a resource. Node-level fencing prevents a failed node from accessing shared resources and

prevents resources from running on a node with an uncertain status. This is usually done by resetting or powering off the node. In a *Pacemaker* cluster, node fencing is managed by the fencing subsystem `pacemaker-fenced`.

GFS2

Global File System 2 (GFS2) is a shared disk file system for Linux computer clusters. GFS2 allows all nodes to have direct concurrent access to the same shared block storage. GFS2 has no disconnected operating mode, and no client or server roles. All nodes in a GFS2 cluster function as peers. GFS2 supports up to 32 cluster nodes. Using GFS2 in a cluster requires hardware to allow access to the shared storage, and a lock manager to control access to the storage.

group

Resource groups contain multiple resources that need to be located together, started sequentially and stopped in the reverse order.

Hawk (High Availability Web Konsole)

A user-friendly Web-based interface for monitoring and administering a High Availability cluster from Linux or non-Linux machines. Hawk can be accessed from any machine that can connect to the cluster nodes, using a graphical Web browser.

heuristics

QDevice supports using a set of commands (*heuristics*) that run locally on start-up of cluster services, cluster membership change, successful connection to the *QNetd* server, or optionally at regular times. The result is used in calculations to determine which partition should have *quorum*.

knet (kronosnet)

A network abstraction layer supporting redundancy, security, fault tolerance, and fast fail-over of network links. In SUSE Linux Enterprise High Availability 16, *knet* is the default transport protocol for the *Corosync* communication channels.

local cluster

A single cluster in one location (for example, all nodes are located in one data center). Network latency is minimal. Storage is typically accessed synchronously by all nodes.

local executor

The local executor is located between *Pacemaker* and the resources on each node. Through the `pacemaker-execd` daemon, Pacemaker can start, stop and monitor resources.

location

In the context of a whole cluster, *location* can refer to the physical location of nodes (for example, all nodes might be located in the same data center). In the context of a *location constraint*, *location* refers to the nodes on which a resource can or cannot run.

location constraint

A type of *resource constraint* that defines the nodes on which a resource can or cannot run.

meta attribute

Parameters that tell the *CRM (cluster resource manager)* how to treat a specific *resource*. For example, you might define a resource's priority or target role.

metro cluster

A single cluster that can stretch over multiple buildings or data centers, with all sites connected by Fibre Channel. Network latency is usually low. Storage is frequently replicated using mirroring or synchronous replication.

network device bonding

Network device bonding combines two or more network interfaces into a single bonded device to increase bandwidth and/or provide redundancy. When using *Corosync*, the bonded device is not managed by the cluster software. Therefore, the bonded device must be configured on every cluster node that might need to access it.

node

Any server (physical or virtual) that is a member of a cluster.

order constraint

A type of *resource constraint* that defines the sequence of actions.

Pacemaker

Pacemaker is the *CRM (cluster resource manager)* in SUSE Linux Enterprise High Availability, or the "brain" that reacts to events occurring in the cluster. Events might be nodes that join or leave the cluster, failure of resources, or scheduled activities such as maintenance, for example. The `pacemakerd` daemon launches and monitors all other related daemons.

parameters (instance attributes)

Parameters determine which instance of a service the *resource* controls.

primitive

A primitive resource is the most basic type of cluster resource.

promotable clone

Promotable clones are a special type of *clone* resource that can be promoted. Active instances of these resources are divided into two states: promoted and unpromoted (also known as “active and passive” or “primary and secondary”).

QDevice

QDevice and *QNetd* participate in *quorum* decisions. The `corosync-qdevice` daemon runs on each cluster node and communicates with QNetd to provide a configurable number of votes, allowing a cluster to sustain more node failures than the standard quorum rules allow.

QNetd

QNetd is an *arbitrator* that runs outside the cluster. The `corosync-qnetd` daemon provides a vote to the `corosync-qdevice` daemon on each node to help it participate in quorum decisions.

quorum

A *cluster partition* is defined to have quorum (be *quorate*) if it has the majority of nodes (or “votes”). Quorum distinguishes exactly one partition. This is part of the algorithm to prevent several disconnected partitions or nodes (“split brain”) from proceeding and causing data and service corruption. Quorum is a prerequisite for fencing, which then ensures that quorum is unique.

RA (resource agent)

A script acting as a proxy to manage a *resource* (for example, to start, stop or monitor a resource). SUSE Linux Enterprise High Availability supports different kinds of resource agents.

ReaR (Relax and Recover)

An administrator tool set for creating *disaster recovery* images.

resource

Any type of service or application that is known to *Pacemaker*, for example, an IP address, a file system, or a database. The term *resource* is also used for *DRBD*, where it names a set of block devices that use a common connection for replication.

resource constraint

Resource constraints specify which cluster nodes a resource can run on, what order the resources should start in, and which other resources a specific resource is dependent on.

See also *colocation constraint*, *location constraint* and *order constraint*.

resource set

As an alternative format for defining location, colocation or order constraints, you can use *resource sets*, where primitives are grouped together in one set. When creating a constraint, you can specify multiple resources for the constraint to apply to.

resource template

To help create many resources with similar configurations, you can define a resource template. After being defined, it can be referenced in primitives or in certain types of constraints. If a template is referenced in a primitive, the primitive inherits all operations, instance attributes (parameters), meta attributes and utilization attributes defined in the template.

SBD (STONITH Block Device)

SBD provides a node *fencing* mechanism through the exchange of messages via shared block storage. Alternatively, it can be used in diskless mode. In either case, it needs a hardware or software *watchdog* on each node to ensure that misbehaving nodes are really stopped.

scheduler

The scheduler is implemented as `pacemaker-schedulerd`. When a cluster transition is needed, `pacemaker-schedulerd` calculates the expected next state of the cluster and determines what actions need to be scheduled to achieve the next state.

split brain

A scenario in which the cluster nodes are divided into two or more groups that do not know about each other (either through a software or hardware failure). Also known as a *partitioned cluster* scenario.

The term *split brain* is also used in *DRBD* but means that the nodes contain different data.

SPOF (single point of failure)

Any component of a cluster that, if it fails, triggers the failure of the entire cluster.

stickiness

Resource stickiness is a *meta attribute* that determines how much a resource prefers to stay on its current node.

switchover

The planned moving of resources to other nodes in a cluster. See also *failover*.

utilization

Tells the CRM what capacity a certain *resource* requires from a node.

watchdog

SBD (STONITH Block Device) needs a watchdog on each node to ensure that misbehaving nodes are really stopped. SBD “feeds” the watchdog by regularly writing a service pulse to it. If SBD stops feeding the watchdog, the hardware enforces a system restart. This protects against failures of the SBD process itself, such as becoming stuck on an I/O error.