Using the SUSE Distribution Migration System

Marcus Schäfer, Jesús Velázquez, Keith Berger, and Robert Schweikert

Publication Date: 2025-06-12

Contents

- Contributors:
- Latest Version: 1.2.0
- Code available: suse-migration-services (https://github.com/SUSE/suse-migration-services)

1 Concept

The Distribution Migration System provides an upgrade path for an installed SUSE Linux Enterprise system from one major version to another, for example, from SUSE Linux Enterprise Server 12 SP5 to SUSE Linux Enterprise Server 15 SP3. For a service pack upgrade from one Service Pack (SP) to another within a given major version release series the existing Zypper migration workflow provides the supported upgrade path. The distribution migration system provides the ability to upgrade across major distributions without the need to use the next major version installation media to perform the system upgrade.

The upgrade is done via the network using the Zypper migration workflow which sends a request to the repository server, asking for an upgrade path. SUSE supported repository servers are the SUSE Customer Center (SCC) and the Repository Management Tool (RMT). The request response contains the list of repositories required to upgrade the system. This requires the system to be upgraded to be registered. Additionally the 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). However administrative work may be required when the system to be upgraded is connected to an RMT server. The migration implementation also supports an upgrade mode that works with systems not registered to a repository service. For details, see *Section 5, “Optional Customization of the Upgrade Process”*.

The upgrade to a new major version requires the system to be migrated to be offline during the upgrade to avoid system inconsistencies that may leave the system in a state that does not allow recovery. This behavior is implemented using a Live Migration Image.

The distribution migration system provides the live image and a startup utility named: `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. Setup 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

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

2 Upgrade Prerequisites

Requirement for using the Zypper migration workflow

Systems that are to be upgraded need to be registered. "Pay as you go"-instances in the Public Cloud are automatically registered to the SUSE operated update infrastructure. All other systems must be connected to the SUSE Customer Center (SCC) or a Repository Management tool (RMT) server. For systems managed via SUSE Manager, use the upgrade path provided by SUSE Manager. The server that provides the repositories must have the appropriate repositories synched and they must be up to date. This requirement is automatically met by the SUSE update infrastructure in the Public Cloud and by SCC.

Recommendation for SSH access during upgrade

During the upgrade, it is only possible to log in via SSH key-based login. If your system is not configured for it, it is recommended that at least one of the users on the system has a `~/.ssh/authorized_keys` file with a private key accessible by the person executing the system upgrade.

Requirement for root filesystem

Root "/" and all core OS functional folders such as /var, /etc, /usr must be on a single partition. Multiple partition support, such as LVM, is limited to configurations where the separated partitions do not contain OS critical data or processes. For example, DMS will function if /home is its own partition, but not if /var is on a separate partition from root.

SLES Version Support

> Upgrade is supported from SLES12 SP5 to SLES 15 SP3. It is possible to configure other versions of SLES as the migration target, but doing so is not a tested or supported use case.

Support for SLES12 SP4 will end December 31 2023. From 2024 onward SLES12 SP5 is the only supported migration starting point.

3 Upgrade Pre-Checks

The suse-migration-pre-checks package contains the `suse-migration-pre-checks` script that checks for possible incompatibilities when doing a migration.

These incompatibilities include:

1. Encrypted file systems
2. Invalid repository types
3. multiversion.kernels enabled and multiple kernels installed
4. Unsupported upgrade path with enabled products

This script is run during the install of SLES15-Migration. It can also be run anytime using

```
/usr/bin/suse-migration-pre-checks
```

The script must be run as root.

A `-f/--fix` option that will remediate the following issues:

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

4 Installation

The distribution migration system is available from the Public Cloud module. Therefore this module has to be enabled on the system prior to upgrade. For running on-demand instances this module is already enabled.

Note

For data center customers it is recommended to continue to use the documented offline distribution upgrade using the next distribution version installation media.

The distribution migration process can be invoked using different methods. One method of activating the migration is the `run_migration` included with the SLES15-Migration package. The second method to invoking the migration process is via reboot after installing the suse-migration-sle15-activation package.

Option 1 - Trigger via run_migration

```
tux > sudo zypper in SLES15-Migration
```

The `run_migration` uses `kexec` to boot into the kernel delivered with the upgrade image delivered by the SLES15-Migration package. Once this system is live after the `kexec` the distribution migration process is automatically started. However, `kexec` is not supported and does not function in certain conditions. The `run_migration` utility does not work in Xen based environments.

If kexec causes a kernel panic this can cause the system to hang and the distribution migration to fail. In that case refer to this TID: https://www.suse.com/support/kb/doc/?id=000019733 ↗ And set the "soft_reboot" customization option:

```
echo "soft_reboot: false" >> /etc/sle-migration-service.yml
```

Option 2 - Trigger via reboot

```
tux > sudo zypper in SLES15-Migration suse-migration-sle15-activation
```

Starting the migration via reboot after installing the suse-migration-sle15-activation package covers the Xen use case but does not work in cases where there is no direct access to the root file system from the bootloader or on architectures other than x86_64. During installation of the suse-migration-sle15-activation package the bootloader configuration is modified such that on the next boot the system will boot into the upgrade image. This in turns starts the automated distribution migration process.

5 Optional Customization of the Upgrade Process

The upgrade live image is pre-configured to run without any further setup. The migration system reads a custom configuration file from the system to be upgraded. The content of this file modifies the behavior of the upgrade process. Prior to the start of the upgrade process, create the following file if a change of the default behavior is needed:

```
tux > ssh INSTANCE_USER@IP_OF_INSTANCE 'touch /etc/sle-migration-service.yml'
```

The custom config file supports the following settings:

Control Zypper Installation Mode

If the upgrade process is used on systems that are not registered or for which the repository server has no upgrade path, it's required to switch off the use of the migration workflow.

```
use_zypper_migration: true|false
```

Note

The use of the migration workflow is the default behavior. If the migration workflow is not used, the setup of the repositories must be performed manually. Once done, the upgrade process uses `zypper dup` and expects all required repositories to be setup correctly.

Specify Migration Product

By default the system will be migrated to SLES15 SP3. This default target can be changed via the `migration_product` setting. The product must be specified with the triplet `name/version/arch` found in '/etc/products.d/baseproduct' of the target product, for example:

```
migration_product: SLES/15.3/x86_64
```

Warning

Changing the default product leads to unsupported territory and is not tested nor covered by the SUSE support offering ! The specified product name must be supported by the repository server used for the migration. If the given product does not exist or the repository server cannot calculate an upgrade path, an error message from the repository server will be logged in the migration log file. Also see: Lifecycle and support (https://documentation.suse.com/sles/15-SP6/html/SLES-all/cha-upgrade-background.html)

Preserve System Data

Preserve custom data file(s) e.g. udev rules from the system to be migrated into the upgrade live system and make sure they will become effective.
Under preserve section, there are two subsections: rules and static. The difference between 'rules' and 'static' sections is that files preserved as udev rules will also make the DMS to reload udev and its rules to make the new rule set effective, while the files in the static section are copied with no further action.

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

Note

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

Enable Debug Mode

If enabled, 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 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 verbosity for 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 Make initrd Method

The live system may not contain all necessary tools to create an initrd that meets the need of the system being upgraded. Building a host independent initrd will create an initrd in a way that contains the tools and modules available on the system being upgraded. If this is needed, a host independent initrd can be created by setting `build_host_independent_initrd: True`.

```
build_host_independent_initrd: true|false
```

Configure 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 setup the migration such that a solver test case is generated. The information form the test case can then be used to understand why a given package was installed.

```
debug_solver: true|false
```

6 Run the Migration

Migration can be triggered either via run_migration or via reboot.

Option 1 - Running Migration via run_migration

After the install of the `SLES15-Migration` package, start the migration process by calling the following command:

```
tux > sudo run_migration
```

Option 2 - Running Migration via reboot

> **Note**
>
> If using the `reboot` method to start migration, reboot the system:

```
tux > sudo reboot
```

After Migration has been triggered via either method

> **Note**
>
> After the upgrade has started, the only way to access the system during the upgrade process is via ssh with a user called `migration`:

```
tux > sudo ssh migration@IP_OF_INSTANCE
```

> **Note**
>
> There is no need to provide any other information or key. The known SSH keys on the system to be upgraded have been imported into the upgrade system. Password-based login is not possible.

The system automatically reboots after the update. To disable this (e.g., for debugging), edit the boot command line in the GRUB menu (select the migration entry and press `e`). Add `migration.noreboot` to the end of the `linux` line, then press `Ctrl-X` or `F10` to start the update. Important: You must reboot manually afterward.

7 After the Migration

Whether the upgrade succeeded or not, a log file is available in `/var/log/distro_migration.log` and it will contain information about the upgrade process. If the upgrade failed, the file `/etc/issue` will contain a pointer to the respective log file.

In addition, the distribution migration RPM packages `SLES15-Migration` and `suse-migration-sle15-activation` will be removed. This is to prevent the migration from being run multiple times and causing a failure loop. To start the migration again, the RPMs will need to be reinstalled following the commands from the Installation Section of this document.

8 Caveats and Unsupported Conditions

- Configuration files that have been modified in the original system will not be overwritten by the upgrade process. The new version of the respective configuration file will be copied into the same directory with the file name extension `.rpmnew`. It is recommended to compare the existing and the new configuration files and make manual adjustments when needed.

- Repositories not registered via `SUSEConnect` and added to the system manually will remain untouched.

- Upgrade is only possible for systems that use unencrypted root file systems, at the OS level. Encrypting the root device using a cloud framework encryption mechanism happens at a different level.

- Upgrade has been tested for SLES 12 SP5 to SLES 15 SP3

- The system is primarily intended for Public Cloud instance upgrade use. The system also works for simple setups in a data center setting on physical installations. However, for any more complex configurations the off line upgrade path via install ISO file should be used as documented in the SUSE Linux Enterprise Server documentation.

- In systems that contain multiple root file systems on different mount points only the root file system mounted on `/` (primary system) will be migrated.

- Upgrade is not supported for systems having the SLE 12 HPC module installed. In SLE 15, HPC is no longer a module but rather a product. With this change, there is not a migration path from SLE 12 (with the HPC module) to SLE 15 HPC.

8.1 Public and Private Cloud Specific

- Migration initiation for a cloud instance is only supported via a reboot. The required GRUB changes to make this process are automated and provided with the suse-migration-sle15-activation package. We recommend to use the provided automation.

- Public Cloud instances from SUSE images have a custom `/etc/motd` file that makes a reference to the distribution version. This needs to be updated manually after the upgrade.

- The instance metadata will not change. As far as the cloud framework is concerned, you will still be running an instance of the SLES version you started with. This cannot be changed.
- The default migration path in the Public Cloud is from the final service pack of SLES 12 (SP5) to SLES 15 SP3. The target may be changed by the customer to service packs greater than SP3, but the source may not be earlier than 12 SP5.