SLE Micro Administration Guide 5.4

SUSE Linux Enterprise Micro is a lightweight and secure operating system for the edge. This guide focuses on administration of this operating system.

WHAT?
: Describes the SLE Micro administration.

WHY?
: You want to learn how to administer SLE Micro.

GOAL
: You'll be able to handle basic management of your system.

Publication Date: 28 Aug 2025

Contents

1 /etc on a read-only file system

SLE Micro was designed to use a read-only root file system. This means that after the deployment is complete, you are not able to perform direct modifications to the root file system. Instead, SLE Micro introduces the concept of transactional updates which enables you to modify your system and keep it up to date.

Even though `/etc` is part of the read-only file system, using an `OverlayFS` layer on this directory enables you to write to this directory. All modifications that you performed on the content of `/etc` are written to the `/var/lib/overlay/SNAPSHOT_NUMBER/etc`. Each snapshot has one associated `OverlayFS` directory.

Whenever a new snapshot is created (for example, as a result of a system update), the content of `/etc` is synchronized and used as a base in the new snapshot. In the `OverlayFS` terminology, the current snapshot's `/etc` is mounted as `lowerdir`. The new snapshot's `/etc` is mounted as `upperdir`. If there were no changes in the `upperdir` `/etc`, any changes performed to the `lowerdir` are visible to the `upperdir`. Therefore, the new snapshot also contains the changes from the current snapshot's `/etc`.

Important: Concurrent modification of `lowerdir` and `upperdir`

If `/etc` in both snapshots is modified, only the changes in the new snapshot (`upperdir`) persist. Changes made to the current snapshot (`lowerdir`) are not synchronized to the new snapshot. Therefore, we do not recommend changing `/etc` after a new snapshot has been created and the system has not been rebooted. However, you can still find the changes in the `/var/lib/overlay/` directory for the snapshot in which the changes were performed.

Note: Using the `--continue` option of the **`transactional-update`** command

When using the `--continue` option and the new snapshot is a descendant of the current snapshot, then the `/etc` overlays of all the snapshots in between will be added as additional directories to the `lowerdir` (the `lowerdir` can have several mount points).

2 Snapshots

Warning: Snapshots are mandatory

As snapshots are crucial for the correct functioning of SLE Micro, do not disable the feature, and ensure that the root partition is big enough to store the snapshots.

When a snapshot is created, both the snapshot and the original subvolume point to the same blocks in the file system. So, initially, a snapshot does not occupy additional disk space. If data in the original file system is modified, changed data blocks are copied while the old data blocks are kept for the snapshot.

Snapshots always reside on the same partition or subvolume on which the snapshot has been taken. It is not possible to store snapshots on a different partition or subvolume. As a result, partitions containing snapshots need to be larger than partitions which do not contain snapshots. The exact amount depends strongly on the number of snapshots you keep and the amount of data modifications. As a rule of thumb, give partitions twice as much space as you normally would. To prevent disks from running out of space, old snapshots are automatically cleaned up.

Snapshots that are known to be working properly are marked as *important*.

2.1 Directories excluded from snapshots

`/home`

Contains users' data. Excluded so that the data is not included in snapshots and thus potentially overwritten by a rollback operation.

`/root`

Contains `root` data. Excluded so that the data is not included in snapshots and thus potentially overwritten by a rollback operation.

`/opt`

Third-party products are usually installed to `/opt`. Excluded so that these applications are not uninstalled during rollbacks.

`/srv`

Contains data for Web and FTP servers. Excluded to avoid data loss on rollbacks.

`/usr/local`

This directory is used when manually installing software. It is excluded to avoid uninstalling these installations on rollbacks.

`/var`

This directory contains many variable files, including logs, temporary caches, third-party products in `/var/opt`, and is the default location for virtual machine images and databases. Therefore, a separate subvolume is created with Copy-On-Write disabled to exclude all such variable data from snapshots.

`/tmp`

The directory contains temporary data.

the architecture-specific `/boot/grub2` directory

Rollback of the boot loader binaries is not supported.

2.2 Showing exclusive disk space used by snapshots

Snapshots share data for efficient use of storage space, so using ordinary commands like **`du`** and **`df`** does not determine the used disk space accurately. When you want to free up disk space on Btrfs with quotas enabled, you need to know how much exclusive disk space is used by each snapshot, rather than shared space. The **`btrfs`** command provides a view of the space used by snapshots:

```
# btrfs qgroup show -p /
qgroupid         rfer         excl parent
--------         ----         ---- ------
0/5          16.00KiB     16.00KiB ---
```

```
[...]
0/272         3.09GiB     14.23MiB 1/0
0/273         3.11GiB    144.00KiB 1/0
0/274         3.11GiB    112.00KiB 1/0
0/275         3.11GiB    128.00KiB 1/0
0/276         3.11GiB     80.00KiB 1/0
0/277         3.11GiB    256.00KiB 1/0
0/278         3.11GiB    112.00KiB 1/0
0/279         3.12GiB     64.00KiB 1/0
0/280         3.12GiB     16.00KiB 1/0
1/0           3.33GiB    222.95MiB ---
```

The `qgroupid` column displays the identification number for each subvolume, assigning a qgroup level/ID combination.

The `rfer` column displays the total amount of data referred to in the subvolume.

The `excl` column displays the exclusive data in each subvolume.

The `parent` column shows the parent qgroup of the subvolumes.

The final item, `1/0`, shows the totals for the parent qgroup. In the above example, 222.95 MiB will be freed after all subvolumes are removed. Run the following command to see which snapshots are associated with each subvolume:

```
# btrfs subvolume list -st /
```

3 Transactional updates

3.1 What are transactional updates?

To keep the base operating system stable and consistent, the SUSE Linux Enterprise Micro uses a read-only root file system. Therefore, you cannot perform direct changes to the root file system, for example, by using the **`zypper`** command. Instead, SLE Micro introduces *transactional updates* that allow you to apply one or more changes to the root file system.

The default **`transactional-update`** behavior is to create a new snapshot from the current root file system after each change. To apply the changes, you need to reboot the host. You cannot run the **`transactional-update`** command multiple times without rebooting to add more changes to the snapshot. This action creates separate independent snapshots that do not include changes from the previous snapshots.

3.2 How do transactional updates work?

Each time you call the **`transactional-update`** command to change your system—either to install a package, perform an update, or apply a patch—the following actions take place:

PROCEDURE 1: MODIFYING THE ROOT FILE SYSTEM

1. A new read-write snapshot is created from your current root file system, or from a snapshot that you specified.
2. All changes are applied (updates, patches or package installation).
3. The snapshot is switched back to read-only mode.
4. If the changes were applied successfully, the new root file system snapshot is set as default.
5. After rebooting, the system boots into the new snapshot.

3.3 Benefits of transactional updates

- They are atomic—the update is applied only if it completes successfully.
- Changes are applied in a separate snapshot and so do not influence the running system.
- Changes can easily be rolled back.

3.4 Environment within the **`transactional-update`** command

Each time you run the **`transactional-update`** command, the changes are performed in a new snapshot. The environment in the snapshot may differ from the one in the shell you run the **`transactional-update`** command from. For example, the current working directory (`$PWD`) is not set to the directory from which you run the **`transactional-update`**, but is set to `/`.

From within the snapshot, you cannot access the `/var` directory. This directory is also not included in the snapshot. However, some directories are not included in the snapshot but are accessible inside the **`transactional-update`** environment, for example, the `/root` directory.

3.5 Applying multiple changes using **transactional-update** without rebooting

To make multiple changes to the root file system without rebooting, you have several options, which are described in the following sections:

3.5.1 The **transactional-update** `--continue` option

Use the **transactional-update** command together with the `--continue` option to make multiple changes without rebooting. A separate snapshot is created on each run that contains all changes from the previous snapshot, plus your new changes. The final snapshot includes all changes. To apply them, reboot the system and your final snapshot becomes the new root file system.

3.5.2 The **transactional-update run** command

The **transactional-update run** command normally runs only a single command. However, you can use it to run multiple commands in one transactional session by concatenating them within a command shell such as **bash**, for example:

```
> sudo transactional-update run bash -c 'ls && date; if [ true ]; then echo -n "Hello ";
 echo '\''world'\''; fi'
```

> **Note**
>
> The **transactional-update run** command has the same limitations as the **transactional-update shell** command described in *Section 3.5.3, "The **transactional-update** shell"* except that the entered commands are logged in the `/var/log/transactional-update.log` file.

3.5.3 The **transactional-update** shell

The **transactional-update shell** command opens a shell in the transactional-update environment. In the shell, you can enter almost any Linux command to make changes to the file system, for example, install multiple packages with the **zypper** command or perform changes to files that are part of the read-only file system. You can also verify that the changes you previously made with the **transactional-update** command are correct.

> **Important**
>
> The transactional shell has several limitations. For example, you cannot operate start or stop services using `systemd` commands, or modify the `/var` partition because it is not mounted. Also, commands entered during a shell session are not logged in the `/transactional-update.log` file.

All changes that you make to the file system are part of a single snapshot. After you finish making changes to the file system and leave the shell with the **exit** command, you need to reboot the host to apply the changes.

3.6 Usage of the **transactional-update** command

The **transactional-update** command syntax is as follows:

```
transactional-update [option] [general_command] [package_command] standalone_command
```

> **Note: Running transactional-update without arguments**
>
> If you do not specify any command or option while running the **transactional-update** command, the system updates itself.

Possible command parameters are described further.

transactional-update OPTIONS

`--interactive, -i`

Can be used along with a package command to turn on interactive mode.

`--non-interactive, -n`

Can be used along with a package command to turn on non-interactive mode.

`--continue [number], -c`

The `--continue` option is for making multiple changes to the root file system without rebooting. Refer to *Section 3.5, "Applying multiple changes using* **`transactional-update`** *without rebooting"* for more details.

Another useful feature of the `--continue` option is that you may select any existing snapshot as the base for your new snapshot. The following example demonstrates running **`transactional-update`** to install a new package in a snapshot based on snapshot 13, and then running it again to install another package:

```
> sudo transactional-update pkg install package_1
```

```
> sudo transactional-update --continue 13 pkg install package_2
```

`--no-selfupdate`

Disables self-updating of **`transactional-update`**.

`--drop-if-no-change, -d`

Discards the snapshot created by **`transactional-update`** if there were no changes to the root file system. If there are changes to the `/etc` directory, those changes merged back to the current file system.

`--quiet`

The **`transactional-update`** command does not output to `stdout` .

`--help, -h`

Prints help for the **`transactional-update`** command.

`--version`

Displays the version of the **`transactional-update`** command.

3.6.1 General commands

This section lists general purpose commands of **`transactional-update`**.

`grub.cfg`

Use this command to rebuild the GRUB boot loader configuration file.

`bootloader`

The command reinstalls the boot loader.

initrd

Use the command to rebuild `initrd`.

kdump

In case you perform changes to your hardware or storage, you may need to rebuild the Kdump initrd.

reboot

The system reboots after the **transactional-update** command is complete.

run *<command>*

Runs the provided command in a new snapshot.

shell

Opens a read-write shell in the new snapshot before exiting. The command is typically used for debugging purposes.

setup-selinux

Installs and enables targeted SELinux policy.

3.7 Performing snapshots cleanup using transactional-update

transactional-update recognizes the following cleanup commands:

cleanup-snapshots

The command marks all unused snapshots for removal by Snapper.

cleanup-overlays

The command removes all unused overlay layers of `/etc` in the `/var/lib/overlay` directory.

cleanup

The command combines the **cleanup-snapshots** and **cleanup-overlays** commands.

3.7.1 How the cleanup works

If you run the command **transactional-update cleanup**, all old snapshots without a cleanup algorithm will have one set. All important snapshots are also marked. The command also removes all unreferenced (and thus unused) `/etc` overlay directories in `/var/lib/overlay`.

The snapshots with the set `number` cleanup algorithm will be deleted according to the rules configured in `/etc/snapper/configs/root` by the following parameters:

NUMBER_MIN_AGE
: Defines the minimum age of a snapshot (in seconds) that can be automatically removed.

NUMBER_LIMIT/NUMBER_LIMIT_IMPORTANT
: Defines the maximum count of stored snapshots. The cleaning algorithms delete snapshots above the specified maximum value, without taking into account the snapshot and file system space. The algorithms also delete snapshots above the minimum value until the limits for the snapshot and file system are reached.

The snapshot cleanup is also regularly performed by `systemd`.

3.8 Performing system rollback

GRUB 2 enables booting from btrfs snapshots and thus allows you to use any older functional snapshot in case the new snapshot does not work correctly.

When booting a snapshot, the parts of the file system included in the snapshot are mounted read-only; all other file systems and parts that are excluded from snapshots are mounted read-write and can be modified.

Tip: Rolling back to a specific installation state

An initial bootable snapshot is created at the end of the initial system installation. You can go back to that state at any time by booting this snapshot. The snapshot can be identified by the description `first root file system`.

There are two methods to perform a system rollback.

- From a running system, you can set the default snapshot, see more in *Procedure 2, “Rollback from a running system”*.
- Especially in cases where the current snapshot is broken, you can boot into the new snapshot and set it to default. For details, refer to *Procedure 3, “Rollback to a working snapshot”*.

If your current snapshot is functional, you can use the following procedure for a system rollback.

PROCEDURE 2: ROLLBACK FROM A RUNNING SYSTEM

1. Identify the snapshot that should be set as the default one and note its number.

```
> sudo snapper list
```

2. Set the snapshot as default.

```
> sudo transactional-update rollback snapshot_number
```

If you omit the *snapshot number*, the current snapshot will be set as default.

Tip: Setting the last working snapshot

To set the last working snapshot as the default one, run **rollback last**.

3. Reboot your system to boot into the new default snapshot.

The following procedure is used in case the current snapshot is broken and you cannot boot into it.

PROCEDURE 3: ROLLBACK TO A WORKING SNAPSHOT

1. Reboot your system and select `Start bootloader from a read-only snapshot`.
2. Choose a snapshot to boot. The snapshots are sorted according to the date of creation, with the latest one at the top.
3. Log in to your system and check whether everything works as expected. The data written to directories excluded from the snapshots will stay untouched.
4. If the snapshot you booted into is not suitable for the rollback, reboot your system and choose another one.
   If the snapshot works as expected, you can perform the rollback by running the following command:

```
> sudo transactional-update rollback
```

And reboot afterwards.

4 User space live patching

Important: Technical preview

On SLE Micro, ULP is a technical preview only.

Note: Live patching on SLE Micro

Only the currently running processes are affected by the live patches. As the libraries are changed in the new snapshot and **not** in the current one, new processes started in the current snapshot still use the non-patched libraries until you reboot. After the reboot, the system switches to the new snapshot, and all started processes will use the patched libraries.

User space live patching (ULP) refers to the process of applying patches to the libraries used by a running process without interrupting them. Every time a security fix is available as a live patch, customer services are secured after applying the live patch without restarting the processes.

Live patching operations are performed using the `ulp` tool that is part of `libpulp`. `libpulp` is a framework that consists of the `libpulp.so` library and the **`ulp`** binary that makes libraries live patchable and applies live patches.

4.1 Preparing the user space live patching

For ULP to work, proceed as follows:

- The ULP must be installed on your system. To do so, run:

  ```
  # transactional-update pkg in libpulp0 libpulp-tools
  ```

  After successful installation, reboot your system.

- To enable live patching on an application, you need to preload the `libpulp.so.0` library when starting the application:

  ```
  > LD_PRELOAD=/usr/lib64/libpulp.so.0 APPLICATION_CMD
  ```

4.1.1 Using libpulp

Note: Supported libraries for patching

Currently, only `glibc` and `openssl` (`openssl1_1`) are supported. Additional packages will be available after they are prepared for live patching. To receive `glibc` and `openssl` live patches, install both `glibc-livepatches` and `openssl-livepatches` packages:

```
> transactional-update pkg in glibc-livepatches openssl-livepatches
```

After successful installation, reboot your system.

To check whether a library is live patchable, use the following command:

```
> ulp livepatchable PATH_TO_LIBRARY
```

A shared object (`.so`) is a live patch container if it contains the ULP patch description embedded into it. You can verify it with the following command:

```
> readelf -S SHARED_OBJECT | grep .ulp
```

If the output shows that there are both `.ulp` and `.ulp.rev` sections in the shared object, then it is a live patch container.

4.2 Managing live patches using ULP

Tip

You can run the **`ulp`** command either as a normal user or a privileged user via the **`sudo`** mechanism. The difference is that running **`ulp`** via **`sudo`** lets you view information about processes or patch processes that are running by `root`.

4.2.1 Applying live patches

Live patches are applied using the **`ulp trigger`** command, for example:

```
> ulp trigger -p PID LIVEPATCH.so
```

Replace `PID` with the process ID of the running process that uses the library to be patched and `LIVEPATCH.so` with the actual live patch file. The command returns one of the following status messages:

SUCCESS

The live patching operation was successful.

SKIPPED

The patch was skipped because it was not designed for any library loaded in the process.

ERROR

An error occurred, and you can retrieve more information by inspecting the `libpulp` internal message buffer. See *Section 4.2.4, "View internal message queue"* for more information.

It is also possible to apply multiple live patches by using wildcards, for example:

```
> ulp trigger '*.so'
```

The command tries to apply every patch in the current folder to every process that has the `libpulp` library loaded. If the patch is not suitable for the process, it is automatically skipped. In the end, the tool shows how many patches it successfully applied to how many processes.

4.2.2 Reverting live patches

You can use the **`ulp trigger`** command to revert live patches. There are two ways to revert live patches. You can revert a live patch by using the `--revert` switch and passing the live patch container:

```
> ulp trigger -p PID --revert LIVEPATCH.so
```

Alternatively, it is possible to remove all patches associated with a particular library, for example:

```
> ulp trigger -p PID --revert-all=LIBRARY
```

In the example, *`LIBRARY`* refers to the actual library, such as `libcrypto.so.1.1`.

The latter approach can be useful when the source code of the original live patch is not available. Or you want to remove a specific old patch and apply a new one while the target application is still running a secure code, for example:

```
> ulp trigger -p PID  --revert-all=libcrypto.so.1.1 new_livepatch2.so
```

4.2.3 View applied patches

It is possible to verify which applications have live patches applied by running:

```
> ulp patches
```

The output shows which libraries are live patchable and patches loaded in programs, as well as which bugs the patch addresses:

```
PID: 10636, name: test
  Livepatchable libraries:
    in /lib64/libc.so.6:
      livepatch: libc_livepatch1.so
        bug labels: jsc#SLE-0000
    in /usr/lib64/libpulp.so.0:
```

It is also possible to see which functions are patched by the live patch:

```
> ulp dump LIVEPATCH.so
```

4.2.4 View internal message queue

Log messages from `libpulp.so` are stored in a buffer inside the library and are not displayed unless requested by the user. To show these messages, run:

```
> ulp messages -p PID
```

5 NetworkManager

NetworkManager is a program that manages the primary network connection and other connection interfaces. NetworkManager has been designed to be fully automatic by default. NetworkManager is handled by `systemd` and is shipped with all necessary service unit files.

NetworkManager stores all network configurations as a connection, which is a collection of data that describes how to create or connect to a network. These connections are stored as files in the `/etc/NetworkManager/system-connections/` directory.

A connection is active when a particular device uses the connection. The device may have more than one connection configured, but only one can be active at a given time. The other connections can be used to fast switch from one connection to another. For example, if the active connection is not available, NetworkManager tries to connect the device to another configured connection.

To manage connections, use the **`nmcli`** command, described in the *Section 5.3, "Managing the network using NetworkManager"*.

5.1 NetworkManager vs **wicked**

NetworkManager is a program that manages the primary network connection and other connection interfaces. **`wicked`** is a network management tool that provides network configuration as a service and enables changing the network configuration dynamically.

NetworkManager and **`wicked`** provide similar functionality; however, they differ in the following points:

`root` privileges

If you use NetworkManager for network setup, you can easily switch, stop or start your network connection. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges.

`wicked` also provides certain ways to switch, stop or start the connection with or without user intervention, like user-managed devices. However, this always requires `root` privileges to change or configure a network device.

Types of network connections

Both **`wicked`** and NetworkManager can handle network connections with a wireless network (with WEP, WPA-PSK, and WPA-Enterprise access) and wired networks using DHCP and static configuration. They also support connection through dial-up and VPN. With NetworkManager, you can also connect a mobile broadband (3G) modem or set up a DSL connection, which is not possible with the traditional configuration.

NetworkManager tries to keep your computer connected at all times using the best connection available. If the network cable is accidentally disconnected, it tries to reconnect. NetworkManager can find the network with the best signal strength from the list of your wireless connections and automatically use it to connect. To get the same functionality with **`wicked`**, more configuration effort is required.

Kubernetes integration

Some Kubernetes plug-ins require NetworkManager to run and are not compatible with **wicked**.

5.2 The `NetworkManager.conf` configuration file

The main configuration file for the NetworkManager is `/etc/NetworkManager/NetworkManager.conf`. This file can be used to configure the behavior of NetworkManager.

The file consists of sections of key-value pairs. Each key-value pair must belong to a section. A section starts with a name enclosed in *[]*. Lines beginning with a # are considered comments. The minimal configuration needs to include the `[main]` section with the `plugins` value:

```
[main]
plugins=keyfile
```

The `keyfile` plug-in supports all the connection types and capabilities of NetworkManager.

The default configuration file contains the `connectivity` section that specifies the URI to check the network connection.

On SUSE Linux Enterprise Micro, you can also use other sections. For details, refer to networkmanager.conf(5) (https://linux.die.net/man/5/networkmanager.conf).

5.3 Managing the network using NetworkManager

5.3.1 The **nmcli** command

NetworkManager provides a CLI interface to manage your connections. By using the **nmcli** interface, you can connect to a particular network, edit a connection, edit a device, etc. The generic syntax of the **nmcli** is as follows:

```
# nmcli OPTIONS SUBCOMMAND SUBCOMMAND_ARGUMENTS
```

where *OPTIONS* are described in *Section 5.3.1.1, “The **nmcli** command options”* and *SUBCOMMAND* can be any of the following:

connection

enables you to configure your network connection. For details, refer to *Section 5.3.1.2, “The **connection** subcommand”*.

`device`
: For details, refer to *Section 5.3.1.3, “The **`device`** subcommand”*.

`general`
: shows status and permissions. For details refer to *Section 5.3.1.4, “The **`general`** subcommand”*.

`monitor`
: monitors activity of NetworkManager and watches for changes in the state of connectivity and devices. This subcommand does not take any arguments.

`networking`
: queries the networking status. For details, refer to *Section 5.3.1.5, “The **`networking`** subcommand”*.

5.3.1.1 The `nmcli` command options

Besides the subcommands and their arguments, the **`nmcli`** command can take the following options:

`-a|--ask`
: The command stops its run to ask for any missing arguments, for example, for a password to connect to a network.

`-c|--color {yes|no|auto}`
: controls the color output: `yes` to enable the colors, `no` to disable them, and `auto` creates color output only when the standard output is directed to a terminal.

`-m|--mode {tabular|multiline}`
: switches between `tabular` (each line describes a single entry, columns define particular properties of the entry) and `multiline` (each entry comprises more lines, each property is on its own line). `tabular` is the default value.

`-h|--help`
: prints help.

`-w|--wait seconds`
: sets a time-out period for which to wait for NetworkManager to finish operations. Using this option is recommended for commands that might take longer to complete, for example, connection activation.

5.3.1.2 The `connection` subcommand

The **`connection`** command enables you to manage connections or view any information about particular connections. The **`nmcli connection`** provides the following commands to manage your network connections:

`show`

to list connections:

```
# nmcli connection show
```

You can also use this command to show details about a specified connection:

```
# nmcli connection show CONNECTION_ID
```

where *`CONNECTION_ID`* is any of the identifiers: *a connection name, UUID or a path*

`up`

to activate the provided connection. Use the command to reload a connection. Also run this command after you perform any change to the connection.

```
# nmcli connection up [--active] [CONNECTION_ID]
```

When `--active` is specified, only the active profiles are displayed. The default is to display both active connections and static configuration.

`down`

to deactivate a connection.

```
# nmcli connection down CONNECTION_ID
```

where: *`CONNECTION_ID`* is any of the identifiers: *a connection name, UUID or a path*
If you deactivate the connection, it will not reconnect later even if it has the `autoconnect` flag.

`modify`

to change or delete a property of a connection.

```
# nmcli connection modify CONNECTION_ID SETTING.PROPERTY PROPERTY_VALUE
```

where:

- *`CONNECTION_ID`* is any of the identifiers: *a connection name, UUID, or a path*
- *`SETTING.PROPERTY`* is the name of the property, for example, `ipv4.addresses`
- *`PROPERTY_VALUE`* is the desired value of *`SETTING.PROPERTY`*

The following example deactivates the `autoconnect` option on the `ethernet1` connection:

```
# nmcli connection modify ethernet1 connection.autoconnect no
```

`add`

to add a connection with the provided details. The command syntax is similar to the **`modify`** command:

```
# nmcli connection add CONNECTION_ID save YES|NO SETTING.PROPERTY PROPERTY_VALUE
```

You should at least specify a `connection.type` or use `type`. The following example adds an Ethernet connection tied to the `eth0` interface with DHCP, and disables the connection's `autoconnect` flag:

```
# nmcli connection add type ethernet autoconnect no ifname eth0
```

`edit`

to edit an existing connection using an interactive editor.

```
# nmcli connection edit CONNECTION_ID
```

`clone`

to clone an existing connection. The minimal syntax follows:

```
# nmcli connection clone CONNECTION_ID NEW_NAME
```

where *`CONNECTION_ID`* is the connection to be cloned.

`delete`

to delete an existing connection:

```
# nmcli connection delete CONNECTION_ID
```

`monitor`

to monitor the provided connection. Each time the connection changes, NetworkManager prints a line.

```
# nmcli connection monitor CONNECTION_ID
```

`reload`

to reload all connection files from the disk. As NetworkManager does not monitor changes performed to the connection files, you need to use this command whenever you make changes to the files. This command does not take any further subcommands.

load

to load/reload a particular connection file, run:

```
# nmcli connection load CONNECTION_FILE
```

For details about the above-mentioned commands, refer to the **nmcli** documentation (https://networkmanager.dev/docs/api/latest/nmcli.html).

5.3.1.3 The **device** subcommand

The **device** subcommand enables you to show and manage network interfaces. The **nmcli device** command recognizes the following commands:

status

to print the status of all devices.

```
# nmcli device status
```

show

shows detailed information about a device. If no device is specified, all devices are displayed.

```
# mcli device show [DEVICE_NAME]
```

connect

to connect a device. NetworkManager tries to find a suitable connection that will be activated. If there is no compatible connection, a new profile is created.

```
# nmcli device connect DEVICE_NAME
```

modify

performs temporary changes to the configuration that is active on the particular device. The changes are not stored in the connection profile.

```
# nmcli device modify DEVICE_NAME [+|-] SETTING.PROPERTY VALUE
```

For possible `SETTING.PROPERTY` values, refer to *nm-settings-nmcli(5)*.

The example below starts the IPv4 shared connection sharing on the device `con1`.

```
# nmcli dev modify con1 ipv4.method shared
```

disconnect

disconnects a device and prevents the device from automatically activating further connections without manual intervention.

```
# nmcli device disconnect DEVICE_NAME
```

`delete`

to delete the interface from the system. You can use the command to delete only software devices like bonds and bridges. You cannot delete hardware devices with this command.

```
# nmcli device DEVICE_NAME
```

`wifi`

lists all available access points.

```
# nmcli device wifi
```

`wifi connect`

connects to a Wi-Fi network specified by its SSID or BSSID. The command takes the following options:

- `password` - password for secured networks
- `ifname` - interface used for activation
- `name` - you can give the connection a name

```
# nmcli device wifi connect SSID [password PASSWORD_VALUE] [ifname INTERFACE_NAME]
```

To connect to a Wi-Fi *GUESTWiFi* with a password `pass$word2#@@`, run:

```
# nmcli device wifi connect GUESTWiFi password pass$word2#@@
```

5.3.1.4 The `general` subcommand

You can use this command to view NetworkManager status and permissions, and change the host name and logging level. The **`nmcli general`** recognizes the following commands:

`status`

displays the overall status of NetworkManager. Whenever you do not specify a command to the **`nmcli general`** command, status is used by default.

```
# nmcli general status
```

`hostname`

if you do not provide a new host name as an argument, the current host name is displayed. If you specify a new host name, the value is used to set a new host name.

```
# nmcli general hostname [HOSTNAME]
```

For example, to set `MyHostname`, run:

```
# nmcli general hostname MyHostname
```

`permissions`

shows your permission for NetworkManager operations like enabling or disabling networking, modifying connections, etc.

```
# nmcli general permissions
```

`logging`

shows and changes NetworkManager logging levels and domains. Without any arguments, the command displays current logging levels and domains.

```
# nmcli general logging [level LEVEL domains DOMAIN]
```

`LEVEL` is any of the values: `OFF`, `ERR`, `WARN`, `INFO`, `DEBUG`, or `TRACE`.
`DOMAIN` is a list of values that can be as follows: `PLATFORM, RFKILL, ETHER, WIFI, BT, MB, DHCP4, DHCP6, PPP, WIFI_SCAN, IP4, IP6, AUTOIP4, DNS, VPN, SHARING, SUPPLICANT, AGENTS, SETTINGS, SUSPEND, CORE, DEVICE, OLPC, WIMAX, INFINIBAND, FIREWALL, ADSL, BOND, VLAN, BRIDGE, DBUS_PROPS, TEAM, CONCHECK, DCB, DISPATCH, AUDIT, SYSTEMD, VPN_PLUGIN, PROXY`.

5.3.1.5 The `networking` subcommand

The subcommand enables you to query the status of the network. Also, by using this command, you can enable or disable networking. The **`nmcli networking`** command takes the following commands:

`on/off`

enables or disables networking. The **`off`** command deactivates all interfaces managed by NetworkManager.

```
# nmcli networking on
```

connectivity

displays the network connectivity state. If `check` is used, NetworkManager performs a new check of the state. Otherwise, the last detected state is displayed.

```
# nmcli networking connectivity
```

Possible states are the following:

- *none* - the host is not connected to any network.
- *portal* - the host is behind a captive portal and cannot reach the full Internet.
- *limited* - the host is connected to a network, but it has no access to the Internet.
- *full* - the host is connected to a network and has full access to the Internet.
- *unknown* - NetworkManager could not determine the network state.

6 Health checker

`health-checker` is a program delivered with SLE Micro that checks whether services are running properly during booting of your system.

During the boot process, `systemd` calls `health-checker`, which in turn calls its plug-ins. Each plug-in checks a particular service or condition. If each check passes, a status file (`/var/lib/misc/health-check.state`) is created. The status file marks the current root file system as correct.

If any of the `health-checker` plug-ins reports an error, the action taken depends on a particular condition, as described below:

The snapshot is booted for the first time.

If the current snapshot is different from the last one that worked properly, an automatic rollback to the last working snapshot is performed. This means that the last change made to the file system has broken the snapshot.

The snapshot has already booted correctly in the past.

There could be just a temporary problem, and the system is rebooted automatically.

The reboot of a previously correctly booted snapshot has failed.

If there was already a problem during boot and automatic reboot has been triggered, but the problem persists, then the system is kept running to enable the administrator to fix the problem. The services that are tested by the `health-checker` plug-ins are stopped if possible.

6.1 Adding custom plug-ins

`health-checker` supports the addition of your own plug-ins to check services during the boot process. Each plug-in is a bash script that must fulfill the following requirements:

- Plug-ins are located within a specific directory—`/usr/libexec/health-checker`
- The service to be checked by the particular plug-in must be defined in the `Unit` section of the `/usr/lib/systemd/system/health-checker.service` file. For example, the `etcd` service is defined as follows:

```
[Unit]
...
After=etcd.service
...
```

- Each plug-in must have functions called `run.checks` and `stop_services` defined. The `run.checks` function checks whether a particular service has started properly. Remember, the service that has not been enabled by `systemd` should be ignored. The function `stop_services` is called to stop the particular service in case the service has not been started properly. You can use the plug-in template for your reference.

7 About toolbox

SLE Micro uses the **`transactional-update`** command to apply changes to the system, but the changes are applied only after reboot. That solution has several benefits, but it also has some disadvantages. If you need to debug your system and install a new tool, the tool will be available only after reboot. Therefore, you cannot debug the currently running system. For this reason, a utility called `toolbox` has been developed.

`toolbox` is a small script that pulls a container image and runs a privileged container based on that image. `toolbox` is stateful so if you exit the container and start it later, the environment is exactly the same.

The root file system of the container is mounted on `/media/root`.

7.1 Starting and removing toolbox

To start the toolbox container as a regular user with root rights, run the following command:

```
> toolbox --root
```

As root, you can omit the --root option:

```
# toolbox
```

If the script completes successfully, you can see the toolbox container prompt.

To remove the container, run the following command:

```
> sudo podman rm toolbox-USER
```

For example, for the root user:

```
# podman rm toolbox-root
```

Note: Obtaining the toolbox image

You can also use Podman or Cockpit to pull the toolbox image and start a container based on that image.

8 Performance Co-Pilot analysis toolkit

The topic covers the basics of PCP.

The toolkit comprises tools for gathering and processing performance information collected either in real time or from PCP archive logs.

The performance data is collected by *performance metrics domain agents* and passed to the pmcd daemon. The daemon coordinates the gathering and exporting of performance statistics in response to requests from the PCP monitoring tools. **pmlogger** is then used to log the metrics. For details, refer to the PCP documentation (https://pcp.readthedocs.io/en/latest/UAG/IntroductionToPcp.html#) ↗.

8.1 Getting the PCP container image

The PCP container image is based on the *BCI-Init* container that utilizes systemd used to manage the PCP services.

You can pull the container image using Podman or from the Cockpit Web management console. To pull the image by using Podman, run the following command:

```
# podman pull registry.suse.com/suse/pcp:latest
```

To get the container image using Cockpit, go to *Podman containers*, click *Get new image*, and search for *pcp*. Then select the image from the `registry.suse.com` for SLE 15 SP4 and download it.

8.2 Running the PCP container

The following command shows minimal options that you need to use to run a PCP container:

```
# podman run -d  \
  --systemd always \
  -p HOST_IP:HOST_PORT:CONTAINER_PORT \
  -v HOST_DIR:/var/log/pcp/pmlogger \
  PCP_CONTAINER_IMAGE
```

where the options have the following meaning:

`-d`
: The container runs in a detached mode without tty.

`--systemd always`
: Runs the container in the `systemd` mode. All services needed to run in the PCP container are started automatically by `systemd` in the container.

`--privileged`
: The container runs with extended privileges. Use this option if your system has SELinux enabled, otherwise the collected metrics are incomplete.

`-v` *`HOST_DIR`*`:/var/log/pcp/pmlogger`
: Creates a bind mount so that **`pmlogger`** archives are written to the *`HOST_DIR`* on the host. By default, **`pmlogger`** stores the collected metrics in `/var/log/pcp/pmlogger`.

`PCP_CONTAINER_IMAGE`
: Is the downloaded PCP container image.

Other useful options of the **podman run** command follow:

OTHER OPTIONS

`-p HOST_IP:HOST_PORT:CONTAINER_PORT`

Publishes ports of the container by mapping a container port onto a host port. If you do not specify *HOST_IP*, the ports are mapped on the local host. If you omit the *HOST_PORT* value, a random port number is used. By default, the **pmcd** daemon listens on and exposes the PMAPI to receive metrics on the port *44321*, so we recommend mapping this port on the same port number on the host. The **pmproxy** daemon listens on and exposes the REST PMWEBAPI to access metrics on the *44322* port by default, so it is recommended to map this port on the same host port number.

`--net host`

The container uses the host's network. Use this option to collect metrics from the host's network interfaces.

`-e`

The option enables you to set the following environment variables:

PCP_SERVICES

Is a comma-separated list of services to start by `systemd` in the container.
Default services are: `pmcd`, `pmie`, `pmlogger`, `pmproxy`.
You can use this variable to run a container with a list of services that is different from the default one, for example, only with `pmlogger`:

```
# podman run -d \
  --name pmlogger \
  --systemd always \
  -e PCP_SERVICES=pmlogger  \
  -v pcp-archives:/var/log/pcp/pmlogger  \
  registry.suse.com/suse/pcp:latest
```

HOST_MOUNT

Is a path inside the container to the bind mount of the host's root file system. The default value is not set.

REDIS_SERVERS

Specifies a connection to a Redis server. In a non-clustered setup, provide a comma-separated list of host specs. In a clustered setup, provide any individual cluster host, other hosts in the cluster are discovered automatically. The default value is: `localhost:6379`.

If you need to use a different configuration than the one provided by the environment variables, proceed as described in *Section 8.3, “Configuring PCP services”*.

8.2.1 Starting the PCP container automatically on boot

After you run the PCP container, you can configure `systemd` to start the container on boot. To do so, follow the procedure below:

1. Create a unit file for the container by using the **`podman generate systemd`** command:

   ```
   # podman generate systemd --name CONTAINER_NAME > /etc/systemd/system/
   container-CONTAINER_NAME.service
   ```

   where *`CONTAINER_NAME`* is the name of the PCP container you used when running the container from the container image.

2. Enable the service in `systemd`:

   ```
   # systemctl enable container-CONTAINER_NAME
   ```

8.3 Configuring PCP services

All services that run inside the PCP container have a default configuration that might not suit your needs. If you need a custom configuration that cannot be covered by the environment variables, create configuration files for the PCP services and pass them to the PCP using a bind mount as follows:

```
# podman run -d \
  --name CONTAINER_NAME \
  --systemd always \
  -v $HOST_CONFIG:CONTAINER_CONFIG_PATH:z \
  -v HOST_LOGS_PATH:/var/log/pcp/pmlogger  \
  registry.suse.com/suse/pcp:latest
```

Where:

`CONTAINER_NAME`

Is an optional container name.

HOST_CONFIG

Is an absolute path to the config you created on the host machine. You can choose any file name you want.

CONTAINER_CONFIG_PATH

Is an absolute path to a particular configuration file inside the container. Each available configuration file is described in the corresponding sections further.

HOST_LOGS_PATH

Is a directory that should be a bind mount to the container logs.

For example, a container called `pcp`, with the configuration file `pmcd` on the host machine and the `pcp-archives` directory for logs on the host machine, is run by the following command:

```
# podman run -d \
  --name pcp  \
  --systemd always \
  -v $(pwd)/pcp-archives:/var/log/pcp/pmlogger \
  -v $(pwd)/pmcd:/etc/sysconfig/pmcd \
registry.suse.com/suse/pcp:latest
```

8.3.1 Custom **pmcd** daemon configuration

The **pmcd** daemon configuration is stored in the `/etc/sysconfig/pmcd` file. The file stores environment variables that modify the behavior of the **pmcd** daemon.

You can add the following variables to the `/etc/sysconfig/pmcd` file to configure the **pmcd** daemon:

PMCD_LOCAL

Defines whether the remote host can connect to the **pmcd** daemon. If set to *0*, remote connections to the daemon are allowed. If set to *1*, the daemon listens only on the local host. The default value is *0*.

PMCD_MAXPENDING

Defines the maximum count of pending connections to the agent. The default value is *5*.

PMCD_ROOT_AGENT

If the `pmdaroot` is enabled (the value is set to *1*), adding a new PDMA does not trigger restarting of other PMDAs. If `pmdaroot` is not enabled, **pmcd** will require restarting all PMDAs when a new PMDA is added. The default value is *1*.

PMCD_RESTART_AGENTS

If set to *1*, the **pmcd** daemon tries to restart any exited PMDA. Enable this option only if you have enabled `pmdaroot`, as **pmcd** itself does not have privileges to restart PMDA.

PMCD_WAIT_TIMEOUT

Defines the maximum time in seconds **pmcd** can wait to accept a connection. After this time, the connection is reported as failed. The default value is *60*.

PCP_NSS_INIT_MODE

Defines the mode in which **pmcd** initializes the NSS certificate database when secured connections are used. The default value is `readonly`. You can set the mode to `readwrite`, but if the initialization fails, the default value is used as a fallback.

An example follows:

```
PMCD_LOCAL=0
PMCD_MAXPENDING=5
PMCD_ROOT_AGENT=1
PMCD_RESTART_AGENTS=1
PMCD_WAIT_TIMEOUT=70
PCP_NSS_INIT_MODE=readwrite
```

8.3.2 Custom **pmlogger** configuration

The custom configuration for the **pmlogger** is stored in the following configuration files:

- `/etc/sysconfig/pmlogger`
- `/etc/pcp/pmlogger/control.d/local`

8.3.2.1 The `/etc/sysconfig/pmlogger` file

You can use the following attributes to configure the **pmlogger**:

PMLOGGER_LOCAL

Defines whether **pmlogger** allows connections from remote hosts. If set to *1*, **pmlogger** allows connections from a local host only.

PMLOGGER_MAXPENDING

Defines the maximum count of pending connections. The default value is 5.

PMLOGGER_INTERVAL

Defines the default sampling interval **pmlogger** uses. The default value is *60 s*. Keep in mind that this value can be overridden by the **pmlogger** command line.

PMLOGGER_CHECK_SKIP_LOGCONF

Setting this option to *yes* disables the regeneration and checking of the **pmlogger** configuration if the configuration **pmlogger** comes from **pmlogconf**. The default behavior is to regenerate configuration files and check for changes every time **pmlogger** is started.

An example follows:

```
PMLOGGER_LOCAL=1
PMLOGGER_MAXPENDING=5
PMLOGGER_INTERVAL=10
PMLOGGER_CHECK_SKIP_LOGCONF=yes
```

8.3.2.2 The /etc/pcp/pmlogger/control.d/local file

The file /etc/pcp/pmlogger/control.d/local stores specifications of the host, which metrics should be logged, the logging frequency (default is 24 hours), and **pmlogger** options. For example:

```
# === VARIABLE ASSIGNMENTS ===
#
# DO NOT REMOVE OR EDIT THE FOLLOWING LINE
$version=1.1

# Uncomment one of the lines below to enable/disable compression behaviour
# that is different to the pmlogger_daily default.
# Value is days before compressing archives, 0 is immediate compression,
# "never" or "forever" suppresses compression.
#
#$PCP_COMPRESSAFTER=0
#$PCP_COMPRESSAFTER=3
#$PCP_COMPRESSAFTER=never

# === LOGGER CONTROL SPECIFICATIONS ===
#
#Host           P?  S?  directory                       args

# local primary logger
LOCALHOSTNAME   y   n   PCP_ARCHIVE_DIR/LOCALHOSTNAME   -r -T24h10m -c config.default -v
 100Mb
```

Note: Defaults point to local host

If you run the **pmlogger** in a container on a different machine than the one that runs the **pmcd** (a client), change the following line to point to the client:

```
# local primary logger
CLIENT_HOSTNAME    y    n    PCP_ARCHIVE_DIR/CLIENT_HOSTNAME   -r -T24h10m -c
 config.default -v 100Mb
```

For example, for the `slemicro_1` host name, the line should look as follows:

```
# local primary logger
slemicro_1    y    n    PCP_ARCHIVE_DIR/slemicro_1   -r -T24h10m -c config.default -v
 100Mb
```

8.4 Managing PCP metrics

8.4.1 Listing PCP metrics

From within the container, you can use the command **pminfo** to list metrics. For example, to list all available performance metrics, run:

```
# pminfo
```

You can list a group of related metrics by specifying the metrics prefix:

```
# pminfo METRIC_PREFIX
```

For example, to list all metrics related to kernel, use:

```
# pminfo disk

disk.dev.r_await
disk.dm.await
disk.dm.r_await
disk.md.await
disk.md.r_await
...
```

You can also specify additional strings to narrow down the list of metrics, for example:

```
# piminfo disk.dev
```

```
disk.dev.read
disk.dev.write
disk.dev.total
disk.dev.blkread
disk.dev.blkwrite
disk.dev.blktotal
...
```

To get online help text of a particular metric, use the `-t` option followed by the metric, for example:

```
# pminfo -t kernel.cpu.util.user

kernel.cpu.util.user [percentage of user time across all CPUs, including guest CPU time]
```

To display a description text of a particular metric, use the `-T` option followed by the metric, for example:

```
# pminfo -T kernel.cpu.util.user

Help:
percentage of user time across all CPUs, including guest CPU time
```

8.4.2 Checking local metrics

After you start the PCP container, you can verify that metrics are being recorded properly by running the following command inside the container:

```
# pcp

Performance Co-Pilot configuration on localhost:

 platform: Linux localhost 5.3.18-150300.59.68-default #1 SMP Wed May 4 11:29:09 UTC 2022
 (ea30951) x86_64
 hardware: 1 cpu, 1 disk, 1 node, 1726MB RAM
 timezone: UTC
 services: pmcd pmproxy
     pmcd: Version 5.2.2-1, 9 agents, 4 clients
     pmda: root pmcd proc pmproxy xfs linux mmv kvm jbd2
 pmlogger: primary logger: /var/log/pcp/pmlogger/localhost/20220607.09.24
     pmie: primary engine: /var/log/pcp/pmie/localhost/pmie.log
```

Now check if the logs are written to a proper destination:

```
# ls PATH_TO_PMLOGGER_LOGS
```

where *PATH_TO_PMLOGGER_LOGS* should be `/var/log/pcp/pmlogger/localhost/` in this case.

8.4.3 Recording metrics from remote systems

You can deploy collector containers that collect metrics from different remote systems than the ones where the **pmlogger** containers are running. Each remote collector system needs the **pmcd** daemon and a set of *pmda*. To deploy several collectors with a centralized monitoring system, proceed as follows.

1. On each system you want to collect metrics from (clients), run a container with the **pmcd** daemon:

   ```
   # podman run -d \
       --name pcp-pmcd \
       --privileged \
       --net host \
       --systemd always \
       -e PCP_SERVICES=pmcd \
       -e HOST_MOUNT=/host \
       -v /:/host:ro,rslave \
       registry.suse.com/suse/pcp:latest
   ```

2. On the monitoring system, create a **pmlogger** configuration file for each client `control.CLIENT` with the following content:

   ```
   $version=1.1

   CLIENT_HOSTNAME n n PCP_ARCHIVE_DIR/CLIENT -N -r -T24h10m -c config.default -v 100Mb
   ```

   Keep in mind that the *CLIENT_HOSTNAME* must be resolvable in DNS. You can use IP addresses or fully qualified domain names (FQDN) instead.

3. On the monitoring system, create a directory for each client to store the recorded logs:

   ```
   # mkdir /root/pcp-archives/CLIENT
   ```

   For example, for `slemicro_1`:

   ```
   # mkdir /root/pcp-archives/slemicro_1
   ```

4. On the monitoring system, run a container with **pmlogger** for each client:

   ```
   # podman run -d \
   ```

```
    --name pcp-pmlogger-CLIENT \
    --systemd always \
    -e PCP_SERVICES=pmlogger \
    -v /root/pcp-archives/CLIENT:/var/log/pcp/pmlogger:z \
    -v $(pwd)/control.CLIENT:/etc/pcp/pmlogger/control.d/local:z \
    registry.suse.com/suse/pcp:latest
```

For example, for a client called `slemicro_1`:

```
# podman run -d \
    --name pcp-pmlogger-slemicro_1 \
    --systemd always \
    -e PCP_SERVICES=pmlogger \
    -v /root/pcp-archives:/var/log/pcp/pmlogger:z \
    -v $(pwd)/control.slemicro_1:/etc/pcp/pmlogger/control.d/local:z \
    registry.suse.com/suse/pcp:latest
```

Note

The second bind mount points to the configuration file created in *Step 2* and replaces the default **pmlogger** configuration. If you do not create this bind mount, **pmlogger** uses the default `/etc/pcp/pmlogger/control.d/local` file and logging from clients fails as the default configuration points to a local host. For details about the configuration file, refer to *Section 8.3.2.2, "The `/etc/pcp/pmlogger/control.d/local` file"*.

5. To check if the log collection is working properly, run:

```
# ls -l pcp-archives/CLIENT/CLIENT
```

For example:

```
# ls -l pcp-archives/slemicro_1/slemicro_1

total 1076
-rw-r--r--. 1 systemd-network systemd-network 876372 Jun  8 11:24 20220608.10.58.0
-rw-r--r--. 1 systemd-network systemd-network    312 Jun  8 11:22
 20220608.10.58.index
-rw-r--r--. 1 systemd-network systemd-network 184486 Jun  8 10:58
 20220608.10.58.meta
-rw-r--r--. 1 systemd-network systemd-network    246 Jun  8 10:58 Latest
-rw-r--r--. 1 systemd-network systemd-network  24595 Jun  8 10:58 pmlogger.log
```

9 Troubleshooting

9.1 The **supportconfig** tool

If problems occur, you can use the **supportconfig** command-line tool to create a detailed system report. The tool collects information about the system, such as the current kernel version, hardware, installed packages, partition setup, and much more.

The command-line tool is provided by the package `supportutils`, which is installed by default. However, **supportconfig** can integrate plug-ins that are used with each running of **supportconfig**. Which plug-ins are available on your system, depends on installed packages. The plug-ins are stored in the `/usr/lib/supportconfig/plugins/` directory.

The **supportconfig** tool creates a TAR archive with detailed system information that you can hand over to Global Technical Support.

9.2 Collecting system information with **supportconfig**

To create a TAR archive with detailed system information that you can hand over to Global Technical Support, follow the procedure:

1. Run **supportconfig** as `root`. Usually, it is enough to run this tool without any options. For common options, refer to *Section 9.2.1, "Common* **supportconfig** *options"*.

```
# supportconfig


           Support Utilities - Supportconfig
                          Script Version: 3.1.11-46.2
                          Library Version: 3.1.11-29.6
                          Script Date: 2022 10 18
[...]
Gathering system information
  Data Directory:    /var/log/scc_d251_180201_1525 1

  Basic Server Health Check...                      Done  2
  RPM Database...                                   Done  2
  Basic Environment...                              Done  2
  System Modules...                                 Done  2
```

```
[...]
  File System List...                              Skipped ③
[...]
  Command History...                               Excluded ④
[...]
  Supportconfig Plugins:                           1 ⑤
    Plugin: pstree...                              Done
[...]
Creating Tar Ball

==[ DONE ]===================================================================
  Log file tar ball: /var/log/scc_d251_180201_1525.txz ⑥
  Log file size:     732K
  Log file md5sum:   bf23e0e15e9382c49f92cbce46000d8b
=============================================================================/
```

The command output is described below this procedure.

2. Wait for the tool to complete the operation.

3. The default archive location is `/var/log`, with the file name format being `scc_HOST_DATE_TIME.txz`. For the archive content description, refer to *Section 9.3, “Overview of the archive content”*.

① The temporary data directory to store the results. This directory is archived as a tar file, see ⑥.

② The feature was enabled (either by default or selected manually) and executed successfully. The result is stored in a file (see *Table 1, “Comparison of features and file names in the TAR archive”*).

③ The feature was skipped because certain files of one or more RPM packages were changed.

④ The feature was excluded because it was deselected via the `-x` option.

⑤ The script found one plug-in and executes the plug-in **`pstree`**. The plug-in was found in the directory `/usr/lib/supportconfig/plugins/`. See the man page for details.

⑥ The tar file name of the archive, compressed with **`xz`** by default.

9.2.1 Common **supportconfig** options

Usually, it is sufficient to run **supportconfig** without any options. However, you may need to use the following options:

`-E` *MAIL*
: To provide the contact e-mail.

`-N` *NAME*
: To provide your name.

`-O` *COMPANY*
: To provide your company name.

`-P` *PHONE*
: To provide your phone number.

`-i` *KEYWORDS*
: To specify keywords that limit the features to check. *KEYWORDS* is a comma-separated list of case-sensitive keywords.
This option is particularly useful if you have already localized a problem that relates to a specific area or feature set only. For example, you have detected problems with LVM and want to test a recent change that you introduced to the LVM configuration. In this case, it makes sense to gather the minimum **supportconfig** information around LVM only:

```
# supportconfig -i LVM
```

`-F`
: To list all keywords that you can use to limit the features to check.

-m
: To reduce the amount of the information being gathered.

-l
: To collect already rotated log files. This is especially useful in high-logging environments or after a kernel crash when syslog rotates the log files after a reboot.

9.3 Overview of the archive content

The TAR archive contains all the results from the features. Depending on what you have selected (all or only a small set), the archive can contain more or fewer files. The set of features can be limited using the `-i` option (see *Section 9.2.1, "Common* **`supportconfig`** *options"*).

To list the contents of the archive, use this **`tar`** command:

```
# tar xf /var/log/scc_&exampleclient;_180131_1545.txz
```

The following file names are always available inside the TAR archive:

MINIMUM FILES IN ARCHIVE

`basic-environment.txt`
: Contains the date when this script was executed and system information like version of the distribution, hypervisor information, and more.

`basic-health-check.txt`
: Contains basic health checks, such as uptime, virtual memory statistics, free memory and hard disk, checks for zombie processes, and more.

`hardware.txt`
: Contains basic hardware checks like information about the CPU architecture, a list of all connected devices, interrupts, I/O ports, kernel boot messages, and more.

`messages.txt`
: Contains log messages from the system journal.

`rpm.txt`
: Contains a list of all installed RPM packages, their names and versions and where they come from.

`summary.xml`
: Contains information in XML format, such as distribution, version and product-specific fragments.

`supportconfig.txt`
: Contains information about the **`supportconfig`** script itself.

`y2log.txt`
: Contains YaST-specific information like specific packages, configuration files and log files.

The following table lists all available features and their file names.

TABLE 1: COMPARISON OF FEATURES AND FILE NAMES IN THE TAR ARCHIVE

Feature	File name
APPARMOR	security-apparmor.txt
AUDIT	security-audit.txt
AUTOFS	fs-autofs.txt
BOOT	boot.txt
BTRFS	fs-btrfs.txt
DAEMONS	systemd.txt
CIMOM	cimom.txt
CRASH	crash.txt
CRON	cron.txt
DHCP	dhcp.txt
DISK	fs-diskio.txt
DNS	dns.txt
DOCKER	docker.txt
DRBD	drbd.txt
ENV	env.txt
ETC	etc.txt
HISTORY	shell_history.txt
ISCSI	fs-iscsi.txt
LDAP	ldap.txt
LIVEPATCH	kernel-livepatch.txt
LVM	lvm.txt
MEM	memory.txt
MOD	modules.txt

Feature	File name
MPIO	mpio.txt
NET	network-*.txt
NFS	nfs.txt
NTP	ntp.txt
NVME	nvme.txt
OCFS2	ocfs2.txt
PAM	pam.txt
PODMAN	podman.txt
PRINT	print.txt
PROC	proc.txt
SAR	sar.txt
SLERT	slert.txt
SLP	slp.txt
SMT	smt.txt
SMART	fs-smartmon.txt
SMB	samba.txt
SRAID	fs-softraid.txt
SSH	ssh.txt
SSSD	sssd.txt
SYSCONFIG	sysconfig.txt
SYSFS	sysfs.txt
TRANSACTIONAL	transactional-update.txt
TUNED	tuned.txt

Feature	File name
`UDEV`	`udev.txt`
`UFILES`	`fs-files-additional.txt`
`UP`	`updates.txt`
`WEB`	`web.txt`

9.4 Submitting information to Global Technical Support

After you have created the archive using the **`supportconfig`** tool, you can submit the archive to SUSE.

9.4.1 Creating a service request number

Before handing over the **`supportconfig`** data to Global Technical Support, you need to generate a service request number first. You will need it to upload the archive to support.

To create a service request, go to https://scc.suse.com/support/requests and follow the instructions on the screen. Write down the service request number.

Note: Privacy statement

SUSE treats system reports as confidential data. For details about our privacy commitment, see https://www.suse.com/company/policies/privacy/ .

9.4.2 Uploading targets

After having created a service request number, you can upload your **`supportconfig`** archives to Global Technical Support. In the examples below, *12345678901* serves as a placeholder for your service request number. Replace the placeholder with the service request number you created in *Section 9.4.1, "Creating a service request number"*.

The following procedures assume that you have already created a **`supportconfig`** archive but have not uploaded it yet.

PROCEDURE 4: SUBMITTING INFORMATION TO SUPPORT ON SERVERS WITH INTERNET CONNECTIVITY

1. Run the **supportconfig** tool as follows:

   a. To use the default upload target `https://support-ftp.us.suse.com/incoming/upload.php?file={tarball}`, run:

   ```
   > sudo supportconfig -ur 12345678901
   ```

   b. For the FTPS upload target `ftps://support-ftp.us.suse.com`, use the following command:

   ```
   > sudo supportconfig -ar 12345678901
   ```

   To use a different upload target, for example, for the EMEA area, use the `-U` followed by the particular URL, either `https://support-ftp.emea.suse.com/incoming/upload.php?file={tarball}"` or `ftps://support-ftp.emea.suse.com/incoming/`:

   ```
   > sudo supportconfig -r 12345678901 -U https://support-ftp.emea.suse.com/
   incoming
   ```

2. After the TAR archive arrives in the incoming directory of our FTP server, it becomes automatically attached to your service request.

If the servers do not provide Internet connectivity, follow the steps below:

PROCEDURE 5: SUBMITTING INFORMATION TO SUPPORT ON SERVERS WITHOUT INTERNET CONNECTIVITY

1. Run the following:

   ```
   > sudo supportconfig -r 12345678901
   ```

2. Manually upload the `/var/log/scc_SR12345678901*txz` archive to one of our servers. The selection of a server depends on your location in the world:

   - North America: HTTPS `https://support-ftp.us.suse.com/incoming/upload.php?file={tarball}`, FTPS `ftps://support-ftp.us.suse.com/incoming/`
   - EMEA, Europe, the Middle East, and Africa: FTP `https://support-ftp.emea.suse.com/incoming/upload.php?file={tarball}`, FTPS `ftps://support-ftp.emea.suse.com/incoming/`

3. After the TAR archive arrives in the incoming directory of our FTP server, it becomes automatically attached to your service request.

10 Legal Notice

A GNU Free Documentation License

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 formats 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 noncommercially, 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.