Applies to **SUSE Linux Enterprise Server 12 SP5**

7 System Recovery and Snapshot Management with Snapper

Snapper allows creating and managing file system snapshots. File system snapshots allow keeping a copy of the state of a file system at a certain point of time. The standard setup of Snapper is designed to allow rolling back system changes. However, you can also use it to create on-disk backups of user data. As the basis for this functionality, Snapper uses the Btrfs file system or thinly-provisioned LVM volumes with an XFS or Ext4 file system.

Snapper has a command-line interface and a YaST interface. Snapper lets you create and manage file system snapshots on the following types of file systems:

- Btrfs, a copy-on-write file system for Linux that natively supports file system snapshots of subvolumes. (Subvolumes are separately mountable file systems within a physical partition.)

  You can also boot from `Btrfs` snapshots. For more information, see Section 7.3, “System Rollback by Booting from Snapshots”.
- Thinly-provisioned LVM volumes formatted with XFS or Ext4.

Using Snapper, you can perform the following tasks:

- Undo system changes made by **`zypper`** and YaST. See Section 7.2, “Using Snapper to Undo Changes” for details.
- Restore files from previous snapshots. See Section 7.2.2, “Using Snapper to Restore Files” for details.
- Do a system rollback by booting from a snapshot. See Section 7.3, “System Rollback by Booting from Snapshots” for details.

- Manually create and manage snapshots, within the running system. See Section 7.6, “Manually Creating and Managing Snapshots” for details.

7.1 Default Setup

Snapper on SUSE Linux Enterprise Server is set up as an undo and recovery tool for system changes. By default, the root partition (`/`) of SUSE Linux Enterprise Server is formatted with `Btrfs` . Taking snapshots is automatically enabled if the root partition (`/`) is big enough (more than approximately 16 GB). By default, snapshots are disabled on partitions other than `/` .

Tip: Enabling Snapper in the Installed System

If you disabled Snapper during the installation, you can enable it at any time later. To do so, create a default Snapper configuration for the root file system by running:

```
tux > sudo snapper -c root create-config /
```

Afterward enable the different snapshot types as described in Section 7.1.3.1, “Disabling/Enabling Snapshots”.

Note that on a Btrfs root file system, snapshots require a file system with sub-volumes set up as proposed by the installer and a partition size of at least 16 GB.

When a snapshot is created, both the snapshot and the original 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. Therefore, a snapshot occupies the same amount of space as the data modified. So, over time, the amount of space a snapshot allocates, constantly grows. As a consequence, deleting files from a `Btrfs` file system containing snapshots may *not* free disk space!

Note: Snapshot Location

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 not containing 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. Refer to Section 7.1.3.4, “Controlling Snapshot Archiving” for details.

7.1.1 Types of Snapshots

Although snapshots themselves do not differ in a technical sense, we distinguish between three types of snapshots, based on the events that trigger them:

Timeline Snapshots

A single snapshot is created every hour. Old snapshots are automatically deleted. By default, the first snapshot of the last ten days, months, and years are kept. Using the YaST OS installation method (default), timeline snapshots are enabled, except for the root file system.

Installation Snapshots

Whenever one or more packages are installed with YaST or Zypper, a pair of snapshots is created: one before the installation starts (“Pre”) and another one after the installation has finished (“Post”). In case an important system component such as the kernel has been installed, the snapshot pair is marked as important (`important=yes`). Old snapshots are automatically deleted. By default the last ten important snapshots and the last ten “regular” (including administration snapshots) snapshots are kept. Installation snapshots are enabled by default.

Administration Snapshots

Whenever you administrate the system with YaST, a pair of snapshots is created: one when a YaST module is started (“Pre”) and another when the module is closed (“Post”). Old snapshots are automatically deleted. By default the last ten important snapshots and the last ten “regular” snapshots (including installation snapshots) are kept. Administration snapshots are enabled by default.

7.1.2 Directories That Are Excluded from Snapshots

Some directories need to be excluded from snapshots for different reasons. The following list shows all directories that are excluded:

`/boot/grub2/i386-pc`, `/boot/grub2/x86_64-efi`, `/boot/grub2/powerpc-ieee1275`, `/boot/grub2/s390x-emu`
: A rollback of the boot loader configuration is not supported. The directories listed above are architecture-specific. The first two directories are present on AMD64/Intel 64 machines, the latter two on IBM POWER and on IBM IBM Z, respectively.

`/home`
: If `/home` does not reside on a separate partition, it is excluded to avoid data loss on rollbacks.

`/opt`, `/var/opt`
: Third-party products usually get installed to `/opt`. It is excluded to avoid uninstalling these applications on rollbacks.

`/srv`
: Contains data for Web and FTP servers. It is excluded to avoid data loss on rollbacks.

`/tmp`, `/var/tmp`, `/var/cache`, `/var/crash`
: All directories containing temporary files and caches are excluded from snapshots.

`/usr/local`
: This directory is used when manually installing software. It is excluded to avoid uninstalling these installations on rollbacks.

`/var/lib/libvirt/images`

The default location for virtual machine images managed with libvirt. Excluded to ensure virtual machine images are not replaced with older versions during a rollback. By default, this subvolume is created with the option `no copy on write`.

`/var/lib/mailman`, `/var/spool`
: Directories containing mails or mail queues are excluded to avoid a loss of mails after a rollback.

`/var/lib/named`
: Contains zone data for the DNS server. Excluded from snapshots to ensure a name server can operate after a rollback.

`/var/lib/mariadb`, `/var/lib/mysql`, `/var/lib/pgqsl`
: These directories contain database data. By default, these subvolumes are created with the option `no copy on write`.

`/var/log`
: Log file location. Excluded from snapshots to allow log file analysis after the rollback of a broken system. By default, `/var/log` has the NoCOW attribute set, disabling copy-on-write, which improves performance and reduces the number of duplicate blocks. Verify with **`lsattr`**:

```
tux > lsattr -l /var/
/var/log      No_COW
```

7.1.3 Customizing the Setup

SUSE Linux Enterprise Server comes with a reasonable default setup, which should be sufficient for most use cases. However, all aspects of taking automatic snapshots and snapshot keeping can be configured according to your needs.

7.1.3.1 Disabling/Enabling Snapshots

Each of the three snapshot types (timeline, installation, administration) can be enabled or disabled independently.

Disabling/Enabling Timeline Snapshots

Enabling. **`snapper -c root set-config "TIMELINE_CREATE=yes"`**

Disabling. **`snapper -c root set-config "TIMELINE_CREATE=no"`**

Using the YaST OS installation method (default), timeline snapshots are enabled, except for the root file system.

Disabling/Enabling Installation Snapshots

Enabling: Install the package `snapper-zypp-plugin`
Disabling: Uninstall the package `snapper-zypp-plugin`
Installation snapshots are enabled by default.

Disabling/Enabling Administration Snapshots

Enabling: Set `USE_SNAPPER` to `yes` in `/etc/sysconfig/yast2`.
Disabling: Set `USE_SNAPPER` to `no` in `/etc/sysconfig/yast2`.
Administration snapshots are enabled by default.

7.1.3.2 Controlling Installation Snapshots

Taking snapshot pairs upon installing packages with YaST or Zypper is handled by the `snapper-zypp-plugin`. An XML configuration file, `/etc/snapper/zypp-plugin.conf` defines, when to make snapshots. By default, the file looks like the following:

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <snapper-zypp-plugin-conf>
 3  <solvables>
 4   <solvable match="w" ① important="true" ② >kernel-* ③ </solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <solvable match="w">*</solvable> ④
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

① The match attribute defines whether the pattern is a Unix shell-style wild card (`w`) or a Python regular expression (`re`).

② If the given pattern matches and the corresponding package is marked as important (for example kernel packages), the snapshot will also be marked as important.

③ Pattern to match a package name. Based on the setting of the `match` attribute, special characters are either interpreted as shell wild cards or regular expressions. This pattern matches all package names starting with `kernel-`.

④ This line unconditionally matches all packages.

With this configuration snapshot, pairs are made whenever a package is installed (line 9). When the kernel, dracut, glibc, systemd, or udev packages marked as important are installed, the snapshot pair will also be marked as important (lines 4 to 8). All rules are evaluated.

To disable a rule, either delete it or deactivate it using XML comments. To prevent the system from making snapshot pairs for every package installation for example, comment line 9:

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <snapper-zypp-plugin-conf>
 3  <solvables>
 4   <solvable match="w" important="true">kernel-*</solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <!-- <solvable match="w">*</solvable> -->
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

7.1.3.3 Creating and Mounting New Subvolumes

Creating a new subvolume underneath the `/` hierarchy and permanently mounting it is supported. Such a subvolume will be excluded from snapshots. You need to make sure not to create it inside an existing snapshot, since you would not be able to delete snapshots anymore after a rollback.

SUSE Linux Enterprise Server is configured with the `/@/` subvolume which serves as an independent root for permanent subvolumes such as `/opt`, `/srv`, `/home` and others. Any new subvolumes you create and permanently mount need to be created in this initial root file system.

To do so, run the following commands. In this example, a new subvolume `/usr/important` is created from `/dev/sda2`.

```
tux > sudo mount /dev/sda2 -o subvol=@ /mnt
tux > sudo btrfs subvolume create /mnt/usr/important
tux > sudo umount /mnt
```

The corresponding entry in `/etc/fstab` needs to look like the following:

```
/dev/sda2 /usr/important btrfs subvol=@/usr/important 0 0
```

Tip: Disable Copy-On-Write (cow)

A subvolume may contain files that constantly change, such as virtualized disk images, database files, or log files. If so, consider disabling the copy-on-write feature for this volume, to avoid duplication of disk blocks. Use the `nodatacow` mount option in `/etc/fstab` to do so:

```
/dev/sda2 /usr/important btrfs nodatacow,subvol=@/usr/important 0 0
```

To alternatively disable copy-on-write for single files or directories, use the command **`chattr +C`** *`PATH`*.

7.1.3.4 Controlling Snapshot Archiving

Snapshots occupy disk space. To prevent disks from running out of space and thus causing system outages, old snapshots are automatically deleted. By default, up to ten important installation and administration snapshots and up to ten regular installation and administration snapshots are kept. If these snapshots occupy more than 50% of the root file system size, additional snapshots will be deleted. A minimum of four important and two regular snapshots are always kept.

Refer to Section 7.5.1, “Managing Existing Configurations” for instructions on how to change these values.

7.1.3.5 Using Snapper on Thinly-Provisioned LVM Volumes

Apart from snapshots on `Btrfs` file systems, Snapper also supports taking snapshots on thinly-provisioned LVM volumes (snapshots on regular LVM volumes are *not* supported) formatted with XFS, Ext4 or Ext3. For more information and setup instructions on LVM volumes, refer to Section 13.2, “LVM Configuration”.

To use Snapper on a thinly-provisioned LVM volume you need to create a Snapper configuration for it. On LVM it is required to specify the file system with `--fstype=lvm(`*`FILESYSTEM`*`)`. `ext3`, `etx4` or `xfs` are valid values for *`FILESYSTEM`*. Example:

```
tux > sudo snapper -c lvm create-config --fstype="lvm(xfs)" /thin_lvm
```

You can adjust this configuration according to your needs as described in Section 7.5.1, “Managing Existing Configurations”.

7.2 Using Snapper to Undo Changes

Snapper on SUSE Linux Enterprise Server is preconfigured to serve as a tool that lets you undo changes made by `zypper` and YaST. For this purpose, Snapper is configured to create a pair of snapshots before and after each run of `zypper` and YaST. Snapper also lets you restore system files that have been accidentally deleted or modified. Timeline snapshots for the root partition need to be enabled for this purpose—see Section 7.1.3.1, "Disabling/Enabling Snapshots" for details.

By default, automatic snapshots as described above are configured for the root partition and its subvolumes. To make snapshots available for other partitions such as `/home` for example, you can create custom configurations.

Important: Undoing Changes Compared to Rollback

When working with snapshots to restore data, it is important to know that there are two fundamentally different scenarios Snapper can handle:

Undoing Changes

When undoing changes as described in the following, two snapshots are being compared and the changes between these two snapshots are made undone. Using this method also allows to explicitly select the files that should be restored.

Rollback

When doing rollbacks as described in Section 7.3, "System Rollback by Booting from Snapshots", the system is reset to the state at which the snapshot was taken.

When undoing changes, it is also possible to compare a snapshot against the current system. When restoring *all* files from such a comparison, this will have the same result as doing a rollback. However, using the method described in Section 7.3, "System Rollback by Booting from Snapshots" for rollbacks should be preferred, since it is faster and allows you to review the system before doing the rollback.

Warning: Data Consistency

There is no mechanism to ensure data consistency when creating a snapshot. Whenever a file (for example, a database) is written at the same time as the snapshot is being created, it will result in a corrupted or partly written file. Restoring such a file will cause problems. Furthermore, some system files such as `/etc/mtab` must never be restored. Therefore it is strongly recommended to *always* closely review the list of changed files and their diffs. Only restore files that really belong to the action you want to revert.

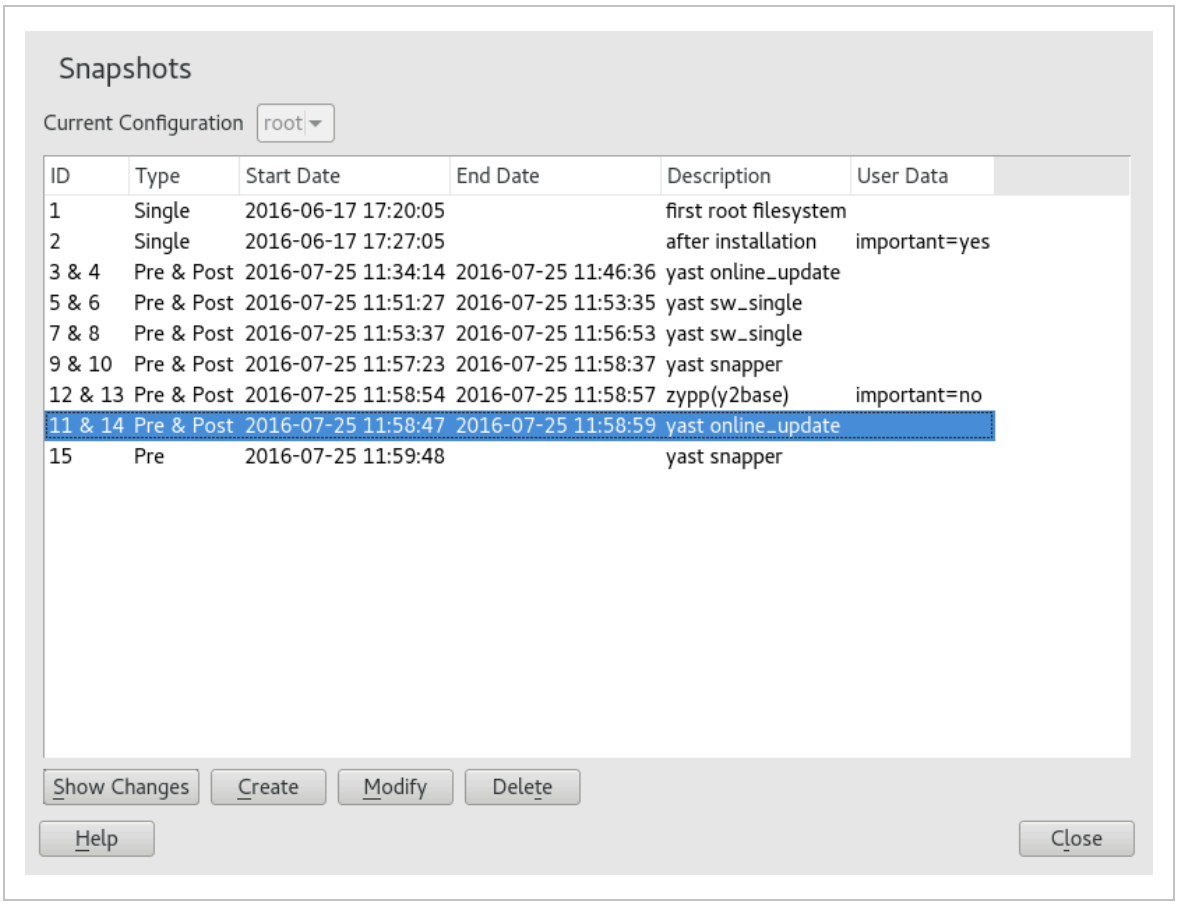
7.2.1 Undoing YaST and Zypper Changes

If you set up the root partition with `Btrfs` during the installation, Snapper—preconfigured for doing rollbacks of YaST or Zypper changes—will automatically be installed. Every time you start a YaST module or a Zypper transaction, two snapshots are created: a "pre-snapshot" capturing the state of the file system before the start of the module and a "post-snapshot" after the module has been finished.

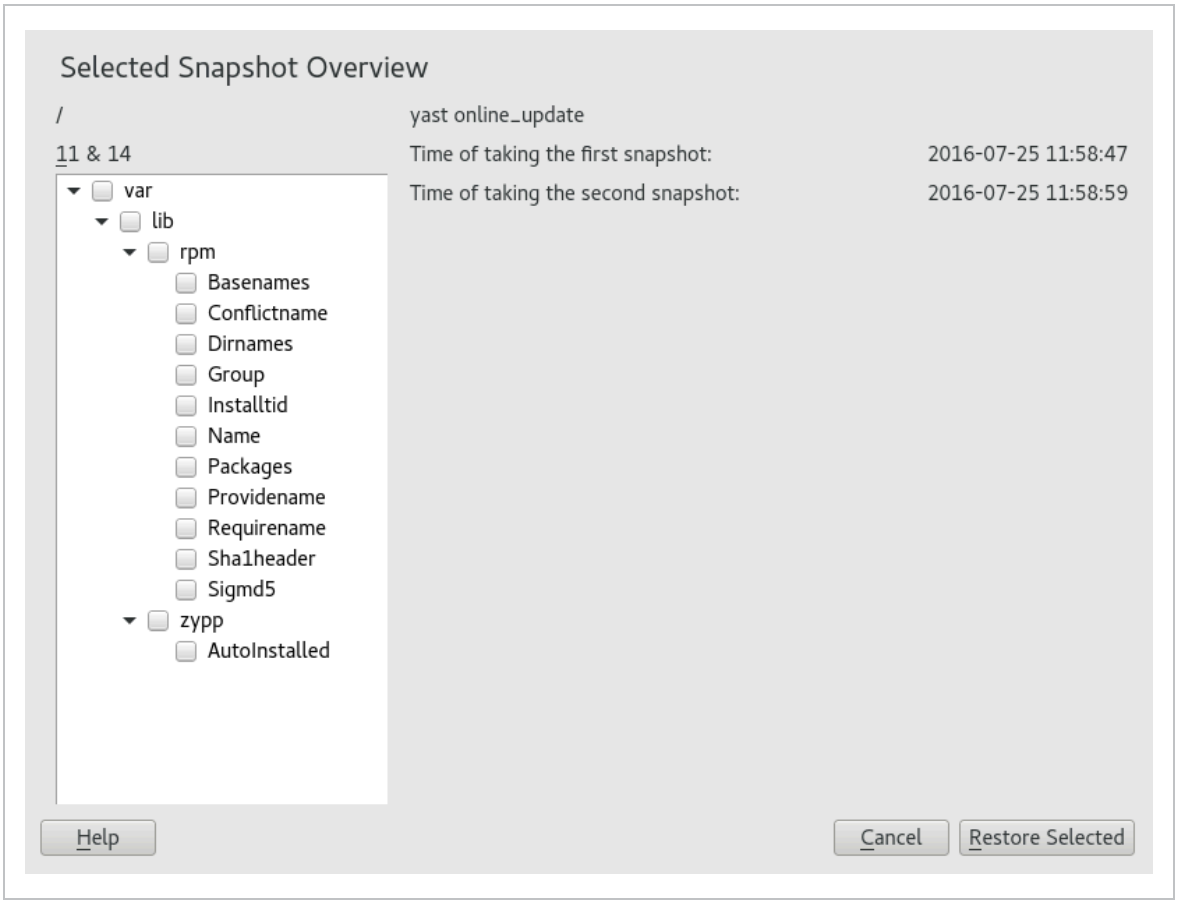
Using the YaST Snapper module or the **`snapper`** command line tool, you can undo the changes made by YaST/Zypper by restoring files from the "pre-snapshot". Comparing two snapshots the tools also allow you to see which files have been changed. You can also display the differences between two versions of a file (diff).

PROCEDURE 7.1: UNDOING CHANGES USING THE YAST *SNAPPER* MODULE

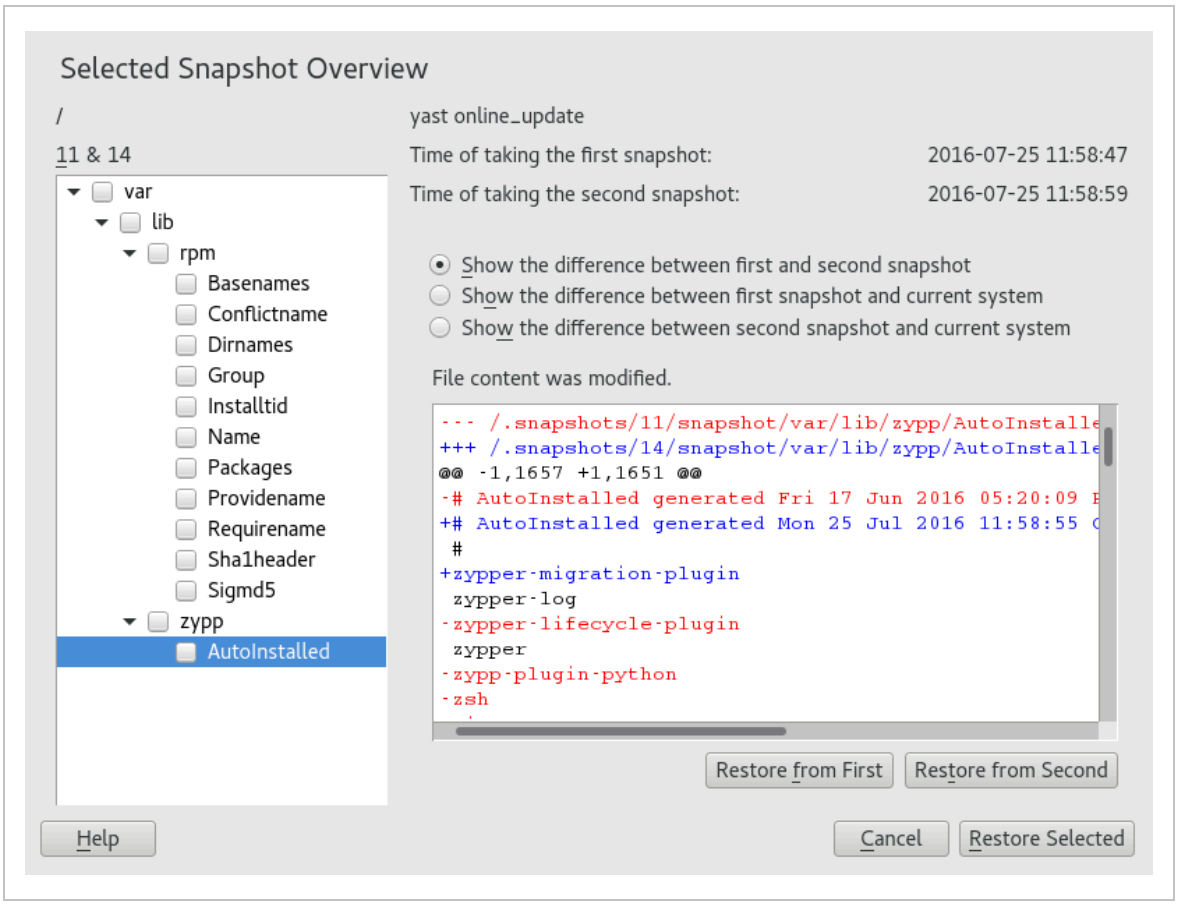
1. Start the *Snapper* module from the *Miscellaneous* section in YaST or by entering **`yast2 snapper`**.
2. Make sure *Current Configuration* is set to *root*. This is always the case unless you have manually added own Snapper configurations.
3. Choose a pair of pre- and post-snapshots from the list. Both, YaST and Zypper snapshot pairs are of the type *Pre & Post*. YaST snapshots are labeled as `zypp(y2base)` in the *Description column*; Zypper snapshots are labeled `zypp(zypper)`.



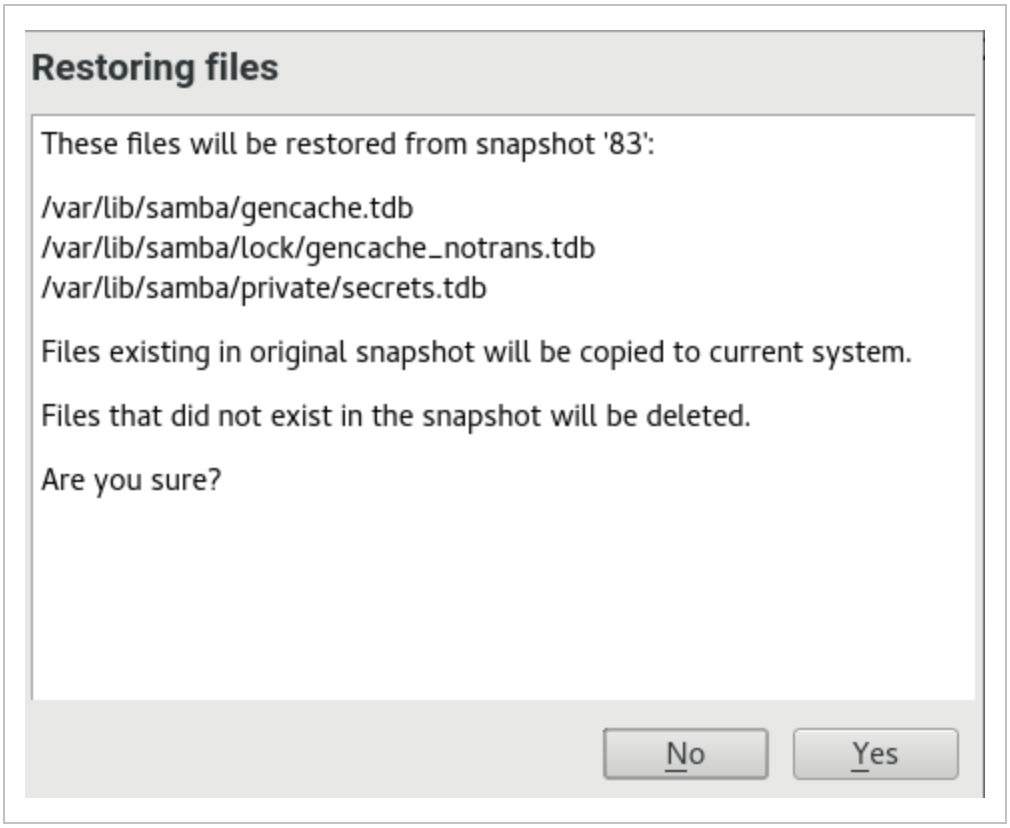
4. Click *Show Changes* to open the list of files that differ between the two snapshots.



5. Review the list of files. To display a “diff” between the pre- and post-version of a file, select it from the list.



6. To restore one or more files, select the relevant files or directories by activating the respective check box. Click *Restore Selected* and confirm the action by clicking *Yes*.



To restore a single file, activate its diff view by clicking its name. Click *Restore From First* and confirm your choice with *Yes*.

PROCEDURE 7.2: UNDOING CHANGES USING THE `snapper` COMMAND

1. Get a list of YaST and Zypper snapshots by running `snapper list -t pre-post`. YaST snapshots are labeled as `yast MODULE_NAME` in the *Description column*; Zypper snapshots are labeled `zypp(zypper)`.

```
tux > sudo snapper list -t pre-post
Pre # | Post # | Pre Date                          | Post Date
------+--------+-----------------------------------+--------------------------
311   | 312    | Tue 06 May 2014 14:05:46 CEST | Tue 06 May 2014 14:05:52
340   | 341    | Wed 07 May 2014 16:15:10 CEST | Wed 07 May 2014 16:15:16
342   | 343    | Wed 07 May 2014 16:20:38 CEST | Wed 07 May 2014 16:20:42
344   | 345    | Wed 07 May 2014 16:21:23 CEST | Wed 07 May 2014 16:21:24
346   | 347    | Wed 07 May 2014 16:41:06 CEST | Wed 07 May 2014 16:41:10
348   | 349    | Wed 07 May 2014 16:44:50 CEST | Wed 07 May 2014 16:44:53
350   | 351    | Wed 07 May 2014 16:46:27 CEST | Wed 07 May 2014 16:46:38
```

2. Get a list of changed files for a snapshot pair with **`snapper status`** *PRE*..*POST*. Files with content changes are marked with *c*, files that have been added are marked with *+* and deleted files are marked with *-*.

```
tux > sudo snapper status 350..351
+..... /usr/share/doc/packages/mikachan-fonts
+..... /usr/share/doc/packages/mikachan-fonts/COPYING
+..... /usr/share/doc/packages/mikachan-fonts/dl.html
c..... /usr/share/fonts/truetype/fonts.dir
c..... /usr/share/fonts/truetype/fonts.scale
+..... /usr/share/fonts/truetype/みかちゃん-p.ttf
+..... /usr/share/fonts/truetype/みかちゃん-pb.ttf
+..... /usr/share/fonts/truetype/みかちゃん-ps.ttf
+..... /usr/share/fonts/truetype/みかちゃん.ttf
c..... /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache
c..... /var/lib/rpm/Basenames
c..... /var/lib/rpm/Dirnames
c..... /var/lib/rpm/Group
c..... /var/lib/rpm/Installtid
c..... /var/lib/rpm/Name
c..... /var/lib/rpm/Packages
c..... /var/lib/rpm/Providename
c..... /var/lib/rpm/Requirename
c..... /var/lib/rpm/Sha1header
c..... /var/lib/rpm/Sigmd5
```

3. To display the diff for a certain file, run **`snapper diff`** *PRE*..*POST* *FILENAME*. If you do not specify *FILENAME*, a diff for all files will be displayed.

```
tux > sudo snapper diff 350..351 /usr/share/fonts/truetype/fonts.scale
--- /.snapshots/350/snapshot/usr/share/fonts/truetype/fonts.scale       20
+++ /.snapshots/351/snapshot/usr/share/fonts/truetype/fonts.scale       20
@@ -1,4 +1,4 @@
-1174
+1486
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso10646
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso8859-
[...]
```

4. To restore one or more files run **`snapper -v undochange`** *PRE*..*POST* *FILENAMES*. If you do not specify a *FILENAMES*, all changed files will be restored.

```
tux > sudo snapper -v undochange 350..351
     create:0 modify:13 delete:7
     undoing change...
     deleting /usr/share/doc/packages/mikachan-fonts
     deleting /usr/share/doc/packages/mikachan-fonts/COPYING
```

```
deleting /usr/share/doc/packages/mikachan-fonts/dl.html
deleting /usr/share/fonts/truetype/みかちゃん-p.ttf
deleting /usr/share/fonts/truetype/みかちゃん-pb.ttf
deleting /usr/share/fonts/truetype/みかちゃん-ps.ttf
deleting /usr/share/fonts/truetype/みかちゃん.ttf
modifying /usr/share/fonts/truetype/fonts.dir
modifying /usr/share/fonts/truetype/fonts.scale
modifying /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_
modifying /var/lib/rpm/Basenames
modifying /var/lib/rpm/Dirnames
modifying /var/lib/rpm/Group
modifying /var/lib/rpm/Installtid
modifying /var/lib/rpm/Name
modifying /var/lib/rpm/Packages
modifying /var/lib/rpm/Providename
modifying /var/lib/rpm/Requirename
modifying /var/lib/rpm/Sha1header
modifying /var/lib/rpm/Sigmd5
undoing change done
```

Warning: Reverting User Additions

Reverting user additions via undoing changes with Snapper is not recommended. Since certain directories are excluded from snapshots, files belonging to these users will remain in the file system. If a user with the same user ID as a deleted user is created, this user will inherit the files. Therefore it is strongly recommended to use the YaST *User and Group Management* tool to remove users.

7.2.2 Using Snapper to Restore Files

Apart from the installation and administration snapshots, Snapper creates timeline snapshots. You can use these backup snapshots to restore files that have accidentally been deleted or to restore a previous version of a file. By using Snapper's diff feature you can also find out which modifications have been made at a certain point of time.

Being able to restore files is especially interesting for data, which may reside on subvolumes or partitions for which snapshots are not taken by default. To be able to restore files from home directories, for example, create a separate Snapper configuration for `/home` doing automatic timeline snapshots. See Section 7.5, “Creating and Modifying Snapper Configurations” for instructions.

Warning: Restoring Files Compared to Rollback

Snapshots taken from the root file system (defined by Snapper's root configuration), can be used to do a system rollback. The recommended way to do such a rollback is to boot from the snapshot and then perform the rollback. See Section 7.3, “System Rollback by Booting from Snapshots” for details.

Performing a rollback would also be possible by restoring all files from a root file system snapshot as described below. However, this is not recommended. You may restore single files, for example a configuration file from the `/etc` directory, but not the complete list of files from the snapshot.

This restriction only affects snapshots taken from the root file system!

PROCEDURE 7.3: RESTORING FILES USING THE YAST *SNAPPER* MODULE

1. Start the *Snapper* module from the *Miscellaneous* section in YaST or by entering **`yast2 snapper`**.
2. Choose the *Current Configuration* from which to choose a snapshot.
3. Select a timeline snapshot from which to restore a file and choose *Show Changes*. Timeline snapshots are of the type *Single* with a description value of *timeline*.
4. Select a file from the text box by clicking the file name. The difference between the snapshot version and the current system is shown. Activate the check box to select the file for restore. Do so for all files you want to restore.
5. Click *Restore Selected* and confirm the action by clicking *Yes*.

PROCEDURE 7.4: RESTORING FILES USING THE **`snapper`** COMMAND

1. Get a list of timeline snapshots for a specific configuration by running the following command:

```
tux > sudo snapper -c CONFIG list -t single | grep timeline
```

`CONFIG` needs to be replaced by an existing Snapper configuration. Use **`snapper list-configs`** to display a list.

2. Get a list of changed files for a given snapshot by running the following command:

```
tux > sudo snapper -c CONFIG status SNAPSHOT_ID..0
```

Replace *SNAPSHOT_ID* by the ID for the snapshot from which you want to restore the file(s).

3. Optionally list the differences between the current file version and the one from the snapshot by running

```
tux > sudo snapper -c CONFIG diff SNAPSHOT_ID..0 FILE NAME
```

If you do not specify *<FILE NAME>*, the difference for all files are shown.

4. To restore one or more files, run

```
tux > sudo snapper -c CONFIG -v undochange SNAPSHOT_ID..0 FILENAME1 FILENA
```

If you do not specify file names, all changed files will be restored.

7.3 System Rollback by Booting from Snapshots

The GRUB 2 version included on SUSE Linux Enterprise Server can boot from Btrfs snapshots. Together with Snapper's rollback feature, this allows to recover a misconfigured system. Only snapshots created for the default Snapper configuration (`root`) are bootable.

Important: Supported Configuration

As of SUSE Linux Enterprise Server 12 SP5 system rollbacks are only supported if the default subvolume configuration of the root partition has not been changed.

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.

Important: Undoing Changes Compared to Rollback

When working with snapshots to restore data, it is important to know that there are two fundamentally different scenarios Snapper can handle:

Undoing Changes

When undoing changes as described in Section 7.2, "Using Snapper to Undo Changes", two snapshots are compared and the changes between these two snapshots are reverted. Using this method also allows to explicitly exclude selected files from being restored.

Rollback

When doing rollbacks as described in the following, the system is reset to the state at which the snapshot was taken.

To do a rollback from a bootable snapshot, the following requirements must be met. When doing a default installation, the system is set up accordingly.

REQUIREMENTS FOR A ROLLBACK FROM A BOOTABLE SNAPSHOT

- The root file system needs to be Btrfs. Booting from LVM volume snapshots is not supported.
- The root file system needs to be on a single device. To check, run `sudo /sbin/btrfs filesystem show`. It needs to report `Total devices 1`. If more than `1` device is listed, your setup is not supported.

  Note: Directories excluded from rollback

  Directories that are excluded from snapshots such as `/srv` (see Section 7.1.2, "Directories That Are Excluded from Snapshots" for a full list) may reside on separate devices.

- The system needs to be bootable via the installed boot loader.

- Only contents of the subvolume `/` will be rolled back. It is not possible to include other subvolumes.

To perform a rollback from a bootable snapshot, do as follows:

1. Boot the system. In the boot menu choose *Bootable snapshots* and select the snapshot you want to boot. The list of snapshots is listed by date —the most recent snapshot is listed first.
2. Log in to the system. Carefully check whether everything works as expected. Note that you cannot write to any directory that is part of the snapshot. Data you write to other directories will *not* get lost, regardless of what you do next.
3. Depending on whether you want to perform the rollback or not, choose your next step:
    a. If the system is in a state where you do not want to do a rollback, reboot to boot into the current system state. You can then choose a different snapshot, or start the rescue system.
    b. To perform the rollback, run

    ```
    tux > sudo snapper rollback
    ```

    and reboot afterward. On the boot screen, choose the default boot entry to reboot into the reinstated system. A snapshot of the file system status before the rollback is created. The default subvolume for root will be replaced with a fresh read-write snapshot. For details, see Section 7.3.1, “Snapshots after Rollback”.

    It is useful to add a description for the snapshot with the `-d` option. For example:

    ```
    New file system root since rollback on DATE TIME
    ```

Tip: Rolling Back to a Specific Installation State

If snapshots are not disabled during installation, 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 `after installation`.

A bootable snapshot is also created when starting a system upgrade to a service pack or a new major release (provided snapshots are not disabled).

7.3.1 Snapshots after Rollback

Before a rollback is performed, a snapshot of the running file system is created. The description references the ID of the snapshot that was restored in the rollback.

Snapshots created by rollbacks receive the value `number` for the `Cleanup` attribute. The rollback snapshots are therefore automatically deleted when the set number of snapshots is reached. Refer to Section 7.7, "Automatic Snapshot Clean-Up" for details. If the snapshot contains important data, extract the data from the snapshot before it is removed.

7.3.1.1 Example of Rollback Snapshot

For example, after a fresh installation the following snapshots are available on the system:

```
root # snapper --iso list
Type   | # |     | Cleanup | Description           | Userdata
-------+---+ ... +---------+-----------------------+--------------
single | 0 |     |         | current               |
single | 1 |     |         | first root filesystem |
single | 2 |     | number  | after installation    | important=yes
```

After running `sudo snapper rollback` snapshot `3` is created and contains the state of the system before the rollback was executed. Snapshot `4` is the new default Btrfs subvolume and thus the system after a reboot.

```
root # snapper --iso list
Type   | # |     | Cleanup | Description           | Userdata
-------+---+ ... +---------+-----------------------+--------------
```

```
single | 0 |      |          | current                |
single | 1 |      | number   | first root filesystem  |
single | 2 |      | number   | after installation     | important=yes
single | 3 |      | number   | rollback backup of #1  | important=yes
single | 4 |      |          |                        |
```

7.3.2 Accessing and Identifying Snapshot Boot Entries

To boot from a snapshot, reboot your machine and choose *Start Bootloader from a read-only snapshot*. A screen listing all bootable snapshots opens. The most recent snapshot is listed first, the oldest last. Use the keys ↓ and ↑ to navigate and press Enter to activate the selected snapshot. Activating a snapshot from the boot menu does not reboot the machine immediately, but rather opens the boot loader of the selected snapshot.

FIGURE 7.1: BOOT LOADER: SNAPSHOTS

Warning: Booting Xen from a Btrfs snapshot using UEFI currently fails

Refer to https://www.suse.com/support/kb/doc/?id=000020602 (https://www.suse.com/support/kb/doc/?id=000020602) ↗ for more details.

Each snapshot entry in the boot loader follows a naming scheme which makes it possible to identify it easily:

```
[*] ① OS ②  (KERNEL ③ ,DATE ④ TTIME ⑤ ,DESCRIPTION ⑥ )
```

① If the snapshot was marked `important`, the entry is marked with a `*`.

② Operating system label.

④ Date in the format `YYYY-MM-DD`.

⑤ Time in the format `HH:MM`.

⑥ This field contains a description of the snapshot. In case of a manually created snapshot this is the string created with the option `--description` or a custom string (see Tip: Setting a Custom Description for Boot Loader Snapshot Entries). In case of an automatically created snapshot, it is the tool that was called, for example `zypp(zypper)` or `yast_sw_single`. Long descriptions may be truncated, depending on the size of the boot screen.

Tip: Setting a Custom Description for Boot Loader Snapshot Entries

It is possible to replace the default string in the description field of a snapshot with a custom string. This is for example useful if an automatically created description is not sufficient, or a user-provided description is too long. To set a custom string *STRING* for snapshot *NUMBER*, use the following command:

```
tux > sudo snapper modify --userdata "bootloader=STRING" NUMBER
```

The description should be no longer than 25 characters—everything that exceeds this size will not be readable on the boot screen.

7.3.3 Limitations

A *complete* system rollback, restoring the complete system to the identical state as it was in when a snapshot was taken, is not possible.

7.3.3.1 Directories Excluded from Snapshots

Root file system snapshots do not contain all directories. See Section 7.1.2, “Directories That Are Excluded from Snapshots” for details and reasons. As a general consequence, data from these directories is not restored, resulting in the following limitations.

Add-ons and Third Party Software may be Unusable after a Rollback
: Applications and add-ons installing data in subvolumes excluded from the snapshot, such as `/opt`, may not work after a rollback, if others parts of the application data are also installed on subvolumes included in the snapshot. Re-install the application or the add-on to solve this problem.

File Access Problems
: If an application had changed file permissions and/or ownership in between snapshot and current system, the application may not be able to access these files. Reset permissions and/or ownership for the affected files after the rollback.

Incompatible Data Formats
: If a service or an application has established a new data format in between snapshot and current system, the application may not be able to read the affected data files after a rollback.

Subvolumes with a Mixture of Code and Data
: Subvolumes like `/srv` may contain a mixture of code and data. A rollback may result in non-functional code. A downgrade of the PHP version, for example, may result in broken PHP scripts for the Web server.

User Data
: If a rollback removes users from the system, data that is owned by these users in directories excluded from the snapshot, is not removed. If a user with the same user ID is created, this user will inherit the files. Use a tool like **`find`** to locate and remove orphaned files.

7.3.3.2 No Rollback of Boot Loader Data

A rollback of the boot loader is not possible, since all “stages” of the boot loader must fit together. This cannot be guaranteed when doing rollbacks of `/boot`.

7.4 Enabling Snapper in User Home Directories

You can enable snapshots for users' `/home` directories, which supports a number of use cases:

- Individual users can manage their own snapshots and rollbacks.
- System users, for example database, system, and network admins, can track copies of configuration files, documentation, and so on.
- Samba shares with home directories and Btrfs back-end.

Each user's directory is a Btrfs subvolume of `/home`. It is possible to set this up manually (see Section 7.4.3, “Manually Enabling Snapshots in Home Directories”). However, a more convenient way is to use `pam_snapper`. The `pam_snapper` package installs the `pam_snapper.so` module and helper scripts, which automate user creation and Snapper configuration.

`pam_snapper` provides integration with the **`useradd`** command, pluggable authentication modules (PAM), and Snapper. By default, it creates snapshots at user login and logout, and also creates time-based snapshots as some users remain logged in for extended periods of time. You can change the defaults using the normal Snapper commands and configuration files.

7.4.1 Installing pam_snapper and Creating Users

The easiest way is to start with a new `/home` directory formatted with Btrfs, and no existing users. Install `pam_snapper`:

```
root # zypper in pam_snapper
```

Add this line to `/etc/pam.d/common-session`:

```
session optional pam_snapper.so
```

Use the **`/usr/lib/pam_snapper/pam_snapper_useradd.sh`** script to create a new user and home directory. By default, the script performs a dry run. Edit the script to change `DRYRUN=1` to `DRYRUN=0`. Now you can create a new user:

```
root # /usr/lib/pam_snapper/pam_snapper_useradd.sh \
username group passwd=password
Create subvolume '/home/username'
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
```

The files from `/etc/skel` will be copied into the user's home directory at their first login. Verify that the user's configuration was created by listing your Snapper configurations:

```
root # snapper list --all
Config: home_username, subvolume: /home/username
Type   | # | Pre # | Date | User | Cleanup | Description | Userdata
-------+---+-------+------+------+---------+-------------+---------
single | 0 |       |      | root |         | current     |
```

Over time, this output will become populated with a list of snapshots, which the user can manage with the standard Snapper commands.

7.4.2 Removing Users

Remove users with the **`/usr/lib/pam_snapper/pam_snapper_userdel.sh`** script. By default, it performs a dry run, so edit it to change `DRYRUN=1` to `DRYRUN=0`. This removes the user, the user's home subvolume, Snapper configuration, and deletes all snapshots.

```
root # /usr/lib/pam_snapper/pam_snapper_userdel.sh username
```

7.4.3 Manually Enabling Snapshots in Home Directories

These are the steps for manually setting up users' home directories with Snapper. `/home` must be formatted with Btrfs, and the users not yet created.

```
root # btrfs subvol create /home/username
root # snapper -c home_username create-config /home/username
root # sed -i -e "s/ALLOW_USERS=\"\"/ALLOW_USERS=\"username\"/g" \
```

```
/etc/snapper/configs/home_username
root # yast users add username=username home=/home/username password=password
root # chown username.group /home/username
root # chmod 755 /home/username/.snapshots
```

7.5 Creating and Modifying Snapper Configurations

The way Snapper behaves is defined in a configuration file that is specific for each partition or `Btrfs` subvolume. These configuration files reside under `/etc/snapper/configs/`.

In case the root file system is big enough (approximately 12 GB), snapshots are automatically enabled for the root file system `/` upon installation. The corresponding default configuration is named `root`. It creates and manages the YaST and Zypper snapshot. See Section 7.5.1.1, “Configuration Data” for a list of the default values.

Note: Minimum Root File System Size for Enabling Snapshots

As explained in Section 7.1, “Default Setup”, enabling snapshots requires additional free space in the root file system. The amount depends on the amount of packages installed and the amount of changes made to the volume that is included in snapshots. The snapshot frequency and the number of snapshots that get archived also matter.

There is a minimum root file system size that is required in order to automatically enable snapshots during the installation. This size is approximately 12 GB. This value may change in the future, depending on architecture and the size of the base system. It depends on the values for the following tags in the file `/control.xml` from the installation media:

```
<root_base_size>
<btrfs_increase_percentage>
```

It is calculated with the following formula: *`ROOT_BASE_SIZE`* * (1 + *`BTRFS_INCREASE_PERCENTAGE`* /100)

Keep in mind that this value is a minimum size. Consider using more space for the root file system. As a rule of thumb, double the size you would use when not having enabled snapshots.

You may create your own configurations for other partitions formatted with `Btrfs` or existing subvolumes on a `Btrfs` partition. In the following example we will set up a Snapper configuration for backing up the Web server data residing on a separate, `Btrfs` -formatted partition mounted at `/srv/www`.

After a configuration has been created, you can either use **`snapper`** itself or the YaST *Snapper* module to restore files from these snapshots. In YaST you need to select your *Current Configuration*, while you need to specify your configuration for **`snapper`** with the global switch `-c` (for example, **`snapper -c myconfig list`**).

To create a new Snapper configuration, run **`snapper create-config`**:

```
tux > sudo snapper -c www-data ① create-config /srv/www ②
```

① Name of configuration file.

② Mount point of the partition or `Btrfs` subvolume on which to take snapshots.

This command will create a new configuration file `/etc/snapper/configs/www-data` with reasonable default values (taken from `/etc/snapper/config-templates/default`). Refer to Section 7.5.1, "Managing Existing Configurations" for instructions on how to adjust these defaults.

Tip: Configuration Defaults

Default values for a new configuration are taken from `/etc/snapper/config-templates/default`. To use your own set of defaults, create a copy of this file in the same directory and adjust it to your needs. To use it, specify the `-t` option with the create-config command:

```
tux > sudo snapper -c www-data create-config -t MY_DEFAULTS /srv/www
```

7.5.1 Managing Existing Configurations

The **`snapper`** command offers several subcommands for managing existing configurations. You can list, show, delete and modify them:

Listing Configurations

Use the subcommand **`snapper list-configs`** to get all existing configurations:

```
tux > sudo snapper list-configs
Config | Subvolume
-------+----------
root   | /
usr    | /usr
local  | /local
```

Showing a Configuration

Use the subcommand **`snapper -c CONFIG get-config`** to display the specified configuration. Replace *`CONFIG`* with one of the configuration names shown by **`snapper list-configs`**. For more information about the configuration options, see Section 7.5.1.1, “Configuration Data”.
To display the default configuration, run:

```
tux > sudo snapper -c root get-config
```

Modifying a Configuration

Use the subcommand **`snapper -c CONFIG set-config OPTION=VALUE`** to modify an option in the specified configuration. Replace *`CONFIG`* with one of the configuration names shown by **`snapper list-configs`**. Possible values for *`OPTION`* and *`VALUE`* are listed in Section 7.5.1.1, “Configuration Data”.

Deleting a Configuration

Use the subcommand **`snapper -c CONFIG delete-config`** to delete a configuration. Replace *`CONFIG`* with one of the configuration names shown by **`snapper list-configs`**.

7.5.1.1 Configuration Data

Each configuration contains a list of options that can be modified from the command line. The following list provides details for each option. To change a value, run **`snapper -c CONFIG set-config "KEY=VALUE"`**.

`ALLOW_GROUPS`, `ALLOW_USERS`

Granting permissions to use snapshots to regular users. See Section 7.5.1.2, “Using Snapper as Regular User” for more information.
The default value is `""`.

`BACKGROUND_COMPARISON`

Defines whether pre and post snapshots should be compared in the background after creation.
The default value is `"yes"`.

`EMPTY_*`

Defines the clean-up algorithm for snapshots pairs with identical pre and post snapshots. See Section 7.7.3, “Cleaning Up Snapshot Pairs That Do Not Differ” for details.

`FSTYPE`
: File system type of the partition. Do not change.
The default value is `"btrfs"`.

`NUMBER_*`
: Defines the clean-up algorithm for installation and admin snapshots. See Section 7.7.1, “Cleaning Up Numbered Snapshots” for details.

`QGROUP` / `SPACE_LIMIT`
: Adds quota support to the clean-up algorithms. See Section 7.7.5, “Adding Disk Quota Support” for details.

`SUBVOLUME`
: Mount point of the partition or subvolume to snapshot. Do not change.
The default value is `"/"`.

`SYNC_ACL`
: If Snapper is used by regular users (see Section 7.5.1.2, “Using Snapper as Regular User”), the users must be able to access the `.snapshot` directories and to read files within them. If SYNC_ACL is set to `yes`, Snapper automatically makes them accessible using ACLs for users and groups from the ALLOW_USERS or ALLOW_GROUPS entries.
The default value is `"no"`.

`TIMELINE_CREATE`
: If set to `yes`, hourly snapshots are created. Valid values: `yes`, `no`.
The default value is `"no"`.

`TIMELINE_CLEANUP` / `TIMELINE_LIMIT_*`
: Defines the clean-up algorithm for timeline snapshots. See Section 7.7.2, “Cleaning Up Timeline Snapshots” for details.

7.5.1.2 Using Snapper as Regular User

By default Snapper can only be used by `root`. However, there are cases in which certain groups or users need to be able to create snapshots or undo changes by reverting to a snapshot:

- Web site administrators who want to take snapshots of `/srv/www`
- Users who want to take a snapshot of their home directory

For these purposes, you can create Snapper configurations that grant permissions to users or/and groups. The corresponding `.snapshots` directory needs to be readable and accessible by the specified users. The easiest way to achieve this is to set the SYNC_ACL option to `yes`.

PROCEDURE 7.5: ENABLING REGULAR USERS TO USE SNAPPER

Note that all steps in this procedure need to be run by `root`.

1. If a Snapper configuration does not exist yet, create one for the partition or subvolume on which the user should be able to use Snapper. Refer to Section 7.5, “Creating and Modifying Snapper Configurations” for instructions. Example:

```
tux > sudo snapper --config web_data create /srv/www
```

2. The configuration file is created under `/etc/snapper/configs/CONFIG`, where CONFIG is the value you specified with `-c/--config` in the previous step (for example `/etc/snapper/configs/web_data`). Adjust it according to your needs. For more information, see Section 7.5.1, “Managing Existing Configurations”.

3. Set values for `ALLOW_USERS` and/or `ALLOW_GROUPS` to grant permissions to users and/or groups, respectively. Multiple entries need to be separated by `Space`. To grant permissions to the user `www_admin` for example, run:

```
tux > sudo snapper -c web_data set-config "ALLOW_USERS=www_admin" SYNC_ACL
```

4. The given Snapper configuration can now be used by the specified user(s) and/or group(s). You can test it with the `list` command, for example:

```
www_admin:~ > snapper -c web_data list
```

7.6 Manually Creating and Managing Snapshots

Snapper is not restricted to creating and managing snapshots automatically by configuration; you can also create snapshot pairs ("before and after") or single snapshots manually using either the command-line tool or the YaST module.

All Snapper operations are carried out for an existing configuration (see Section 7.5, "Creating and Modifying Snapper Configurations" for details). You can only take snapshots of partitions or volumes for which a configuration exists. By default the system configuration (`root`) is used. If you want to create or manage snapshots for your own configuration you need to explicitly choose it. Use the *Current Configuration* drop-down box in YaST or specify the `-c` on the command line (**`snapper -c`** *`MYCONFIG COMMAND`*).

7.6.1 Snapshot Metadata

Each snapshot consists of the snapshot itself and some metadata. When creating a snapshot you also need to specify the metadata. Modifying a snapshot means changing its metadata—you cannot modify its content. Use **`snapper list`** to show existing snapshots and their metadata:

`snapper --config home list`
: Lists snapshots for the configuration `home` . To list snapshots for the default configuration (root), use **`snapper -c root list`** or **`snapper list`** .

`snapper list -a`
: Lists snapshots for all existing configurations.

`snapper list -t pre-post`
: Lists all pre and post snapshot pairs for the default (`root`) configuration.

`snapper list -t single`
: Lists all snapshots of the type `single` for the default (`root`) configuration.

The following metadata is available for each snapshot:

- **Type**: Snapshot type, see Section 7.6.1.1, “Snapshot Types” for details. This data cannot be changed.
- **Number**: Unique number of the snapshot. This data cannot be changed.
- **Pre Number**: Specifies the number of the corresponding pre snapshot. For snapshots of type post only. This data cannot be changed.
- **Description**: A description of the snapshot.
- **Userdata**: An extended description where you can specify custom data in the form of a comma-separated key=value list: `reason=testing, project=foo`. This field is also used to mark a snapshot as important (`important=yes`) and to list the user that created the snapshot (user=tux).
- **Cleanup-Algorithm**: Cleanup-algorithm for the snapshot, see Section 7.7, “Automatic Snapshot Clean-Up” for details.

7.6.1.1 Snapshot Types

Snapper knows three different types of snapshots: pre, post, and single. Physically they do not differ, but Snapper handles them differently.

`pre`
: Snapshot of a file system *before* a modification. Each `pre` snapshot corresponds to a `post` snapshot. For example, this is used for the automatic YaST/Zypper snapshots.

`post`
: Snapshot of a file system *after* a modification. Each `post` snapshot corresponds to a `pre` snapshot. For example, this is used for the automatic YaST/Zypper snapshots.

`single`
: Stand-alone snapshot. For example, this is used for the automatic hourly snapshots. This is the default type when creating snapshots.

7.6.1.2 Cleanup Algorithms

Snapper provides three algorithms to clean up old snapshots. The algorithms are executed in a daily `cron` job. It is possible to define the number of different types of snapshots to keep in the Snapper configuration (see Section 7.5.1, "Managing Existing Configurations" for details).

number
: Deletes old snapshots when a certain snapshot count is reached.

timeline
: Deletes old snapshots having passed a certain age, but keeps several hourly, daily, monthly, and yearly snapshots.

empty-pre-post
: Deletes pre/post snapshot pairs with empty diffs.

7.6.2 Creating Snapshots

To create a snapshot, run `snapper create` or click *Create* in the YaST module *Snapper*. The following examples explain how to create snapshots from the command line. The YaST interface for Snapper is not explicitly described here but provides equivalent functionality.

Tip: Snapshot Description

Always specify a meaningful description to later be able to identify its purpose. You can also specify additional information via the option `--userdata`.

`snapper create --description "Snapshot for week 2 2014"`
: Creates a stand-alone snapshot (type single) for the default (`root`) configuration with a description. Because no cleanup-algorithm is specified, the snapshot will never be deleted automatically.

`snapper --config home create --description "Cleanup in ~tux"`
: Creates a stand-alone snapshot (type single) for a custom configuration named `home` with a description. Because no cleanup-algorithm is specified, the snapshot will never be deleted automatically.

`snapper --config home create --description "Daily data backup" --cleanup-algorithm timeline`

›

Creates a stand-alone snapshot (type single) for a custom configuration named `home` with a description. The snapshot will automatically be deleted when it meets the criteria specified for the timeline cleanup-algorithm in the configuration.

`snapper create --type pre --print-number --description "Before the Apache config cleanup" --userdata "important=yes"`

Creates a snapshot of the type `pre` and prints the snapshot number. First command needed to create a pair of snapshots used to save a “before” and “after” state. The snapshot is marked as important.

`snapper create --type post --pre-number 30 --description "After the Apache config cleanup" --userdata "important=yes"`

Creates a snapshot of the type `post` paired with the `pre` snapshot number `30`. Second command needed to create a pair of snapshots used to save a “before” and “after” state. The snapshot is marked as important.

`snapper create --command` *`COMMAND`* `--description "Before and after COMMAND"`

Automatically creates a snapshot pair before and after running *`COMMAND`*. This option is only available when using snapper on the command line.

7.6.3 Modifying Snapshot Metadata

Snapper allows you to modify the description, the cleanup algorithm, and the user data of a snapshot. All other metadata cannot be changed. The following examples explain how to modify snapshots from the command line. It should be easy to adopt them when using the YaST interface.

To modify a snapshot on the command line, you need to know its number. Use **`snapper list`** to display all snapshots and their numbers.

The YaST *Snapper* module already lists all snapshots. Choose one from the list and click *Modify*.

`snapper modify --cleanup-algorithm "timeline" 10`

Modifies the metadata of snapshot 10 for the default (`root`) configuration. The cleanup algorithm is set to `timeline`.

`snapper --config home modify --description "daily backup" -cleanup-algorithm "timeline" 120`

Modifies the metadata of snapshot 120 for a custom configuration named `home`. A new description is set and the cleanup algorithm is unset.

7.6.4 Deleting Snapshots

To delete a snapshot with the YaST *Snapper* module, choose a snapshot from the list and click *Delete*.

To delete a snapshot with the command-line tool, you need to know its number. Get it by running **`snapper list`**. To delete a snapshot, run **`snapper delete`** *`NUMBER`*.

Deleting the current default subvolume snapshot is not allowed.

When deleting snapshots with Snapper, the freed space will be claimed by a Btrfs process running in the background. Thus the visibility and the availability of free space is delayed. In case you need space freed by deleting a snapshot to be available immediately, use the option `--sync` with the delete command.

Tip: Deleting Snapshot Pairs

When deleting a `pre` snapshot, you should always delete its corresponding `post` snapshot (and vice versa).

`snapper delete 65`

Deletes snapshot 65 for the default (`root`) configuration.

`snapper -c home delete 89 90`

Deletes snapshots 89 and 90 for a custom configuration named `home`.

`snapper delete --sync 23`

Deletes snapshot 23 for the default (`root`) configuration and makes the freed space available immediately.

Tip: Delete Unreferenced Snapshots

Sometimes the Btrfs snapshot is present but the XML file containing the metadata for Snapper is missing. In this case the snapshot is not visible for Snapper and needs to be deleted manually:

```
btrfs subvolume delete /.snapshots/SNAPSHOTNUMBER/snapshot
rm -rf /.snapshots/SNAPSHOTNUMBER
```

Tip: Old Snapshots Occupy More Disk Space

If you delete snapshots to free space on your hard disk, make sure to delete old snapshots first. The older a snapshot is, the more disk space it occupies.

Snapshots are also automatically deleted by a daily cron job. Refer to Section 7.6.1.2, “Cleanup Algorithms” for details.

7.7 Automatic Snapshot Clean-Up

Over time, snapshots grow in size, occupying an ever increasing amount of disk space. To prevent disks from running out of space, Snapper offers algorithms to automatically delete old snapshots. These algorithms differentiate between timeline snapshots and numbered snapshots (administration plus installation snapshot pairs). You can specify the number of snapshots to keep for each type.

In addition to that, you can optionally specify a disk space quota, defining the maximum amount of disk space the snapshots may occupy. It is also possible to automatically delete pre and post snapshots pairs that do not differ.

A clean-up algorithm is always bound to a single Snapper configuration, so you need to configure algorithms for each configuration. To prevent certain snapshots from being automatically deleted, refer to *Can a snapshot be protected from deletion?* .

The default setup (`root`) is configured to do clean-up for numbered snapshots and empty pre and post snapshot pairs. In the default setup, quota support is enabled, and snapshots must leave at least 20% of the available disk space on the root partition free. Timeline snapshots are disabled by default, therefore the timeline clean-up algorithm is also disabled.

Note: Improved Clean-Up Algorithm

Previous implementations of the clean-up algorithm only ensured that snapshots do not use more than the specified amount of disk space (default is 50%). In certain cases, this didn't prevent the system from running of disk space. Starting with SUSE Linux Enterprise Server 12 SP1, Snapper features an improved clean-up algorithm that keeps 20% of the available disk space free at all times.

7.7.1 Cleaning Up Numbered Snapshots

Cleaning up numbered snapshots—administration plus installation snapshot pairs—is controlled by the following parameters of a Snapper configuration.

`NUMBER_CLEANUP`
: Enables or disables clean-up of installation and admin snapshot pairs. If enabled, snapshot pairs are deleted when the total snapshot count exceeds a number specified with `NUMBER_LIMIT` and/or `NUMBER_LIMIT_IMPORTANT` *and* an age specified with `NUMBER_MIN_AGE` . Valid values: `yes` (enable), `no` (disable).
The default value is `"yes"` .
Example command to change or set:

```
tux > sudo snapper -c CONFIG set-config "NUMBER_CLEANUP=no"
```

`NUMBER_LIMIT` / `NUMBER_LIMIT_IMPORTANT`
: Defines how many regular and/or important installation and administration snapshot pairs to keep. Only the youngest snapshots will be kept. Ignored if `NUMBER_CLEANUP` is set to `"no"` .
The default value is `"2-10"` for `NUMBER_LIMIT` and `"4-10"` for `NUMBER_LIMIT_IMPORTANT` .

Example command to change or set:

```
tux > sudo snapper -c CONFIG set-config "NUMBER_LIMIT=10"
```

> **Important: Ranged Compared to Constant Values**
>
> In case quota support is enabled (see Section 7.7.5, “Adding Disk Quota Support”) the limit needs to be specified as a minimum-maximum range, for example `2-10`. If quota support is disabled, a constant value, for example `10`, needs to be provided, otherwise cleaning-up will fail with an error.

`NUMBER_MIN_AGE`

Defines the minimum age in seconds a snapshot must have before it can automatically be deleted. Snapshots younger than the value specified here will not be deleted, regardless of how many exist.
The default value is `"1800"`.
Example command to change or set:

```
tux > sudo snapper -c CONFIG set-config "NUMBER_MIN_AGE=864000"
```

Note: Limit and Age

`NUMBER_LIMIT`, `NUMBER_LIMIT_IMPORTANT` and `NUMBER_MIN_AGE` are always evaluated. Snapshots are only deleted when *all* conditions are met.

If you always want to keep the number of snapshots defined with `NUMBER_LIMIT*` regardless of their age, set `NUMBER_MIN_AGE` to `0`.

The following example shows a configuration to keep the last 10 important and regular snapshots regardless of age:

```
NUMBER_CLEANUP=yes
NUMBER_LIMIT_IMPORTANT=10
NUMBER_LIMIT=10
NUMBER_MIN_AGE=0
```

On the other hand, if you do not want to keep snapshots beyond a certain age, set `NUMBER_LIMIT*` to `0` and provide the age with `NUMBER_MIN_AGE`.

The following example shows a configuration to only keep snapshots younger than ten days:

```
NUMBER_CLEANUP=yes
NUMBER_LIMIT_IMPORTANT=0
NUMBER_LIMIT=0
NUMBER_MIN_AGE=864000
```

7.7.2 Cleaning Up Timeline Snapshots

Cleaning up timeline snapshots is controlled by the following parameters of a Snapper configuration.

`TIMELINE_CLEANUP`
: Enables or disables clean-up of timeline snapshots. If enabled, snapshots are deleted when the total snapshot count exceeds a number specified with `TIMELINE_LIMIT_*` *and* an age specified with `TIMELINE_MIN_AGE`. Valid values: `yes`, `no`.
The default value is `"yes"`.
Example command to change or set:

```
tux > sudo snapper -c CONFIG set-config "TIMELINE_CLEANUP=yes"
```

`TIMELINE_LIMIT_DAILY`, `TIMELINE_LIMIT_HOURLY`, `TIMELINE_LIMIT_MONTHLY`, `TIMELINE_LIMIT_WEEKLY`, `TIMELINE_LIMIT_YEARLY`
: Number of snapshots to keep for hour, day, month, week, and year. The default value for each entry is `"10"`, except for `TIMELINE_LIMIT_WEEKLY`, which is set to `"0"` by default.

`TIMELINE_MIN_AGE`
: Defines the minimum age in seconds a snapshot must have before it can automatically be deleted. The default value is `"1800"`.

EXAMPLE 7.1: EXAMPLE TIMELINE CONFIGURATION

```
TIMELINE_CLEANUP="yes"
TIMELINE_CREATE="yes"
TIMELINE_LIMIT_DAILY="7"
TIMELINE_LIMIT_HOURLY="24"
TIMELINE_LIMIT_MONTHLY="12"
TIMELINE_LIMIT_WEEKLY="4"
TIMELINE_LIMIT_YEARLY="2"
TIMELINE_MIN_AGE="1800"
```

This example configuration enables hourly snapshots which are automatically cleaned up. `TIMELINE_MIN_AGE` and `TIMELINE_LIMIT_*` are always both evaluated. In this example, the minimum age of a snapshot before it can be deleted is set to 30 minutes (1800 seconds). Since we create hourly snapshots, this ensures that only the latest snapshots are kept. If `TIMELINE_LIMIT_DAILY` is set to not zero, this means that the first snapshot of the day is kept, too.

SNAPSHOTS TO BE KEPT

- Hourly: The last 24 snapshots that have been made.
- Daily: The first daily snapshot that has been made is kept from the last seven days.
- Monthly: The first snapshot made on the last day of the month is kept for the last twelve months.

- Weekly: The first snapshot made on the last day of the week is kept from the last four weeks.
- Yearly: The first snapshot made on the last day of the year is kept for the last two years.

7.7.3 Cleaning Up Snapshot Pairs That Do Not Differ

As explained in Section 7.1.1, “Types of Snapshots”, whenever you run a YaST module or execute Zypper, a pre snapshot is created on start-up and a post snapshot is created when exiting. In case you have not made any changes there will be no difference between the pre and post snapshots. Such “empty” snapshot pairs can be automatically be deleted by setting the following parameters in a Snapper configuration:

`EMPTY_PRE_POST_CLEANUP`
: If set to `yes`, pre and post snapshot pairs that do not differ will be deleted. The default value is `"yes"`.

`EMPTY_PRE_POST_MIN_AGE`
: Defines the minimum age in seconds a pre and post snapshot pair that does not differ must have before it can automatically be deleted. The default value is `"1800"`.

7.7.4 Cleaning Up Manually Created Snapshots

Snapper does not offer custom clean-up algorithms for manually created snapshots. However, you can assign the number or timeline clean-up algorithm to a manually created snapshot. If you do so, the snapshot will join the “clean-up queue” for the algorithm you specified. You can specify a clean-up algorithm when creating a snapshot, or by modifying an existing snapshot:

`snapper create --description "Test" --cleanup-algorithm number`
: Creates a stand-alone snapshot (type single) for the default (root) configuration and assigns the `number` clean-up algorithm.

`snapper modify --cleanup-algorithm "timeline" 25`

Modifies the snapshot with the number 25 and assigns the clean-up algorithm `timeline`.

7.7.5 Adding Disk Quota Support

In addition to the number and/or timeline clean-up algorithms described above, Snapper supports quotas. You can define what percentage of the available space snapshots are allowed to occupy. This percentage value always applies to the Btrfs subvolume defined in the respective Snapper configuration.

If Snapper was enabled during the installation, quota support is automatically enabled. In case you manually enable Snapper at a later point in time, you can enable quota support by running **`snapper setup-quota`**. This requires a valid configuration (see Section 7.5, “Creating and Modifying Snapper Configurations” for more information).

Note: Btrfs Quota Groups Can Incur Degraded Performance

On SUSE Linux Enterprise Server 12 SP5, using Btrfs quota groups can degrade file system performance.

Quota support is controlled by the following parameters of a Snapper configuration.

`QGROUP`
: The Btrfs quota group used by Snapper. If not set, run **`snapper setup-quota`**. If already set, only change if you are familiar with **`man 8 btrfs-qgroup`**. This value is set with **`snapper setup-quota`** and should not be changed.

`SPACE_LIMIT`
: Limit of space snapshots are allowed to use in fractions of 1 (100%). Valid values range from 0 to 1 (0.1 = 10%, 0.2 = 20%, ...).

The following limitations and guidelines apply:

- Quotas are only activated in *addition* to an existing number and/or timeline clean-up algorithm. If no clean-up algorithm is active, quota restrictions are not applied.

- With quota support enabled, Snapper will perform two clean-up runs if required. The first run will apply the rules specified for number and time-line snapshots. Only if the quota is exceeded after this run, the quota-specific rules will be applied in a second run.
- Even if quota support is enabled, Snapper will always keep the number of snapshots specified with the `NUMBER_LIMIT*` and `TIMELINE_LIMIT*` values, even if the quota will be exceeded. It is therefore recommended to specify ranged values (*`MIN-MAX`*) for `NUMBER_LIMIT*` and `TIMELINE_LIMIT*` to ensure the quota can be applied.

  If, for example, `NUMBER_LIMIT=5-20` is set, Snapper will perform a first clean-up run and reduce the number of regular numbered snapshots to 20. In case these 20 snapshots exceed the quota, Snapper will delete the oldest ones in a second run until the quota is met. A minimum of five snapshots will always be kept, regardless of the amount of space they occupy.

7.8 Frequently Asked Questions

***Q**: Why does Snapper never show changes in `/var/log`, `/tmp` and other directories?*

***Q**: How much disk space is used by snapshots? How can I free disk space?*

***Q**: Can I boot a snapshot from the boot loader?*

***Q**: Can a snapshot be protected from deletion?*

***Q**: Where can I get more information on Snapper?*