https://doc.opensuse.org/documentation/leap/archive/15.0/reference/html/book.opensuse.reference/cha.snapper.html#proc.snapper.restore.cmdl

System Recovery and Snapshot Management with Snapper | Reference

Applies to openSUSE Leap 15.0

AbstractReport Bug

Being able to do file system snapshots providing the ability to do rollbacks on Linux is a feature that was often requested in the past. Snapper, with the `Btrfs` file system or thin-provisioned LVM volumes now fills that gap.

`Btrfs`, a new copy-on-write file system for Linux, supports file system snapshots (a copy of the state of a subvolume at a certain point of time) of subvolumes (one or more separately mountable file systems within each physical partition). Snapshots are also supported on thin-provisioned LVM volumes formatted with XFS, Ext4 or Ext3. Snapper lets you create and manage these snapshots. It comes with a command line and a YaST interface. Starting with openSUSE Leap it is also possible to boot from `Btrfs` snapshots—see Section 3.3, “System Rollback by Booting from Snapshots” for more information.

Using Snapper you can perform the following tasks:

- Undo system changes made by `zypper` and YaST. See Section 3.2, “Using Snapper to Undo Changes” for details.
- Restore files from previous snapshots. See Section 3.2.2, “Using Snapper to Restore Files” for details.
- Do a system rollback by booting from a snapshot. See Section 3.3, “System Rollback by Booting from Snapshots” for details.
- Manually create snapshots on the fly and manage existing snapshots. See Section 3.5, “Manually Creating and Managing Snapshots” for details.

Snapper on openSUSE Leap is set up to serve as an “undo and recovery tool” for system changes. By default, the root partition (`/`) of openSUSE Leap is formatted with `Btrfs`. Taking snapshots is automatically enabled if the root partition (`/`) is big enough (approximately more than 16 GB). Taking snapshots on partitions other than `/` is not enabled by default.

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 3.1.3.1, “Disabling/Enabling Snapshots”.

Keep in mind that snapshots require a Btrfs root file system with subvolumes 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 “normal” partitions. The exact amount strongly depends on the number of snapshots you keep and the amount of data modifications. As a rule of thumb you should consider using twice the size than you normally would. To prevent disks from running out of space, old snapshots are automatically cleaned up. Refer to Section 3.1.3.4, “Controlling Snapshot Archiving” for details.

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. Timeline snapshots are disabled by default.

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.

3.1.2 Directories That Are Excluded from Snapshots Report Bug

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 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.

openSUSE Leap 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.

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"`

Timeline snapshots are enabled by default, except for the root partition.

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.

3.1.3.2 Controlling Installation Snapshots Report Bug

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>
```

3.1.3.3 Creating and Mounting New Subvolumes Report Bug

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.

openSUSE Leap 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`*.

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 3.4.1, “Managing Existing Configurations” for instructions on how to change these values.

3.1.3.5 Using Snapper on Thin-Provisioned LVM Volumes Report Bug

Apart from snapshots on `Btrfs` file systems, Snapper also supports taking snapshots on thin-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 5.2, “LVM Configuration”.

To use Snapper on a thin-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 3.4.1, "Managing Existing Configurations".

Snapper on openSUSE Leap 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 3.1.3.1, "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 3.3, "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 3.3, "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.

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.

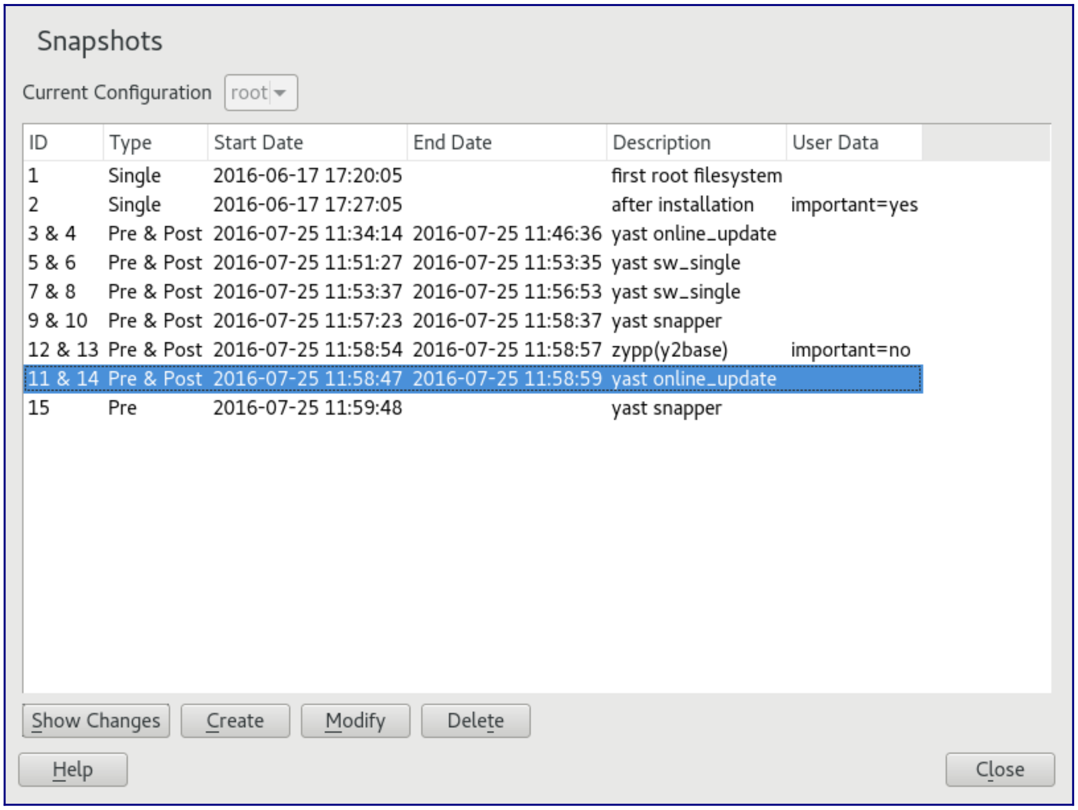
Using the YaST Snapper module or the `snapper` command line tool, you can undo the changes made by YaST/Zypper by restoring files from the “pre-snapshot”. Comparing two snapshots the tools also allow you to see which files have been changed. You can also display the differences between two versions of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

Procedure 3.1: Undoing Changes Using the YaST Module Report Bug #

1. Start the module from the section in YaST or by entering `yast2 snapper`.

2. Make sure is set to . 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 . YaST snapshots are labeled as `zypp(y2base)` in the ; Zypper snapshots are labeled `zypp(zypper)`.

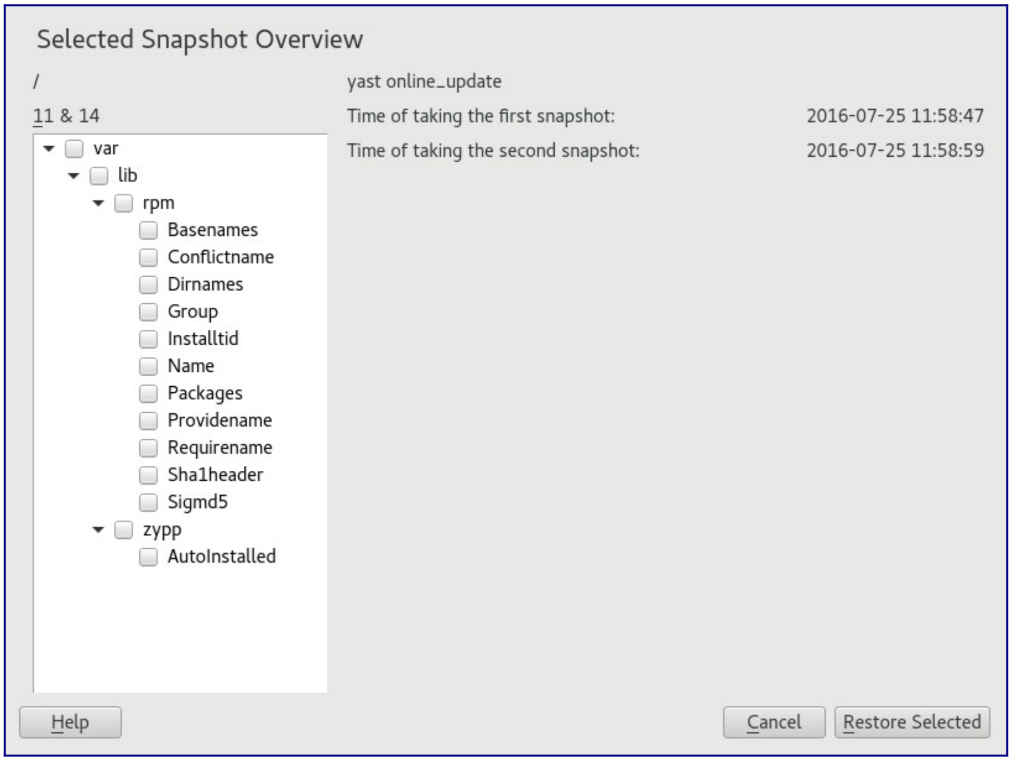


Snapshots

Current Configuration root

ID	Type	Start Date	End Date	Description	User Data
1	Single	2016-06-17 17:20:05		first root filesystem	
2	Single	2016-06-17 17:27:05		after installation	important=yes
3 & 4	Pre & Post	2016-07-25 11:34:14	2016-07-25 11:46:36	yast online_update	
5 & 6	Pre & Post	2016-07-25 11:51:27	2016-07-25 11:53:35	yast sw_single	
7 & 8	Pre & Post	2016-07-25 11:53:37	2016-07-25 11:56:53	yast sw_single	
9 & 10	Pre & Post	2016-07-25 11:57:23	2016-07-25 11:58:37	yast snapper	
12 & 13	Pre & Post	2016-07-25 11:58:54	2016-07-25 11:58:57	zypp(y2base)	important=no
11 & 14	Pre & Post	2016-07-25 11:58:47	2016-07-25 11:58:59	yast online_update	
15	Pre	2016-07-25 11:59:48		yast snapper	

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



Selected Snapshot Overview

/

11 & 14

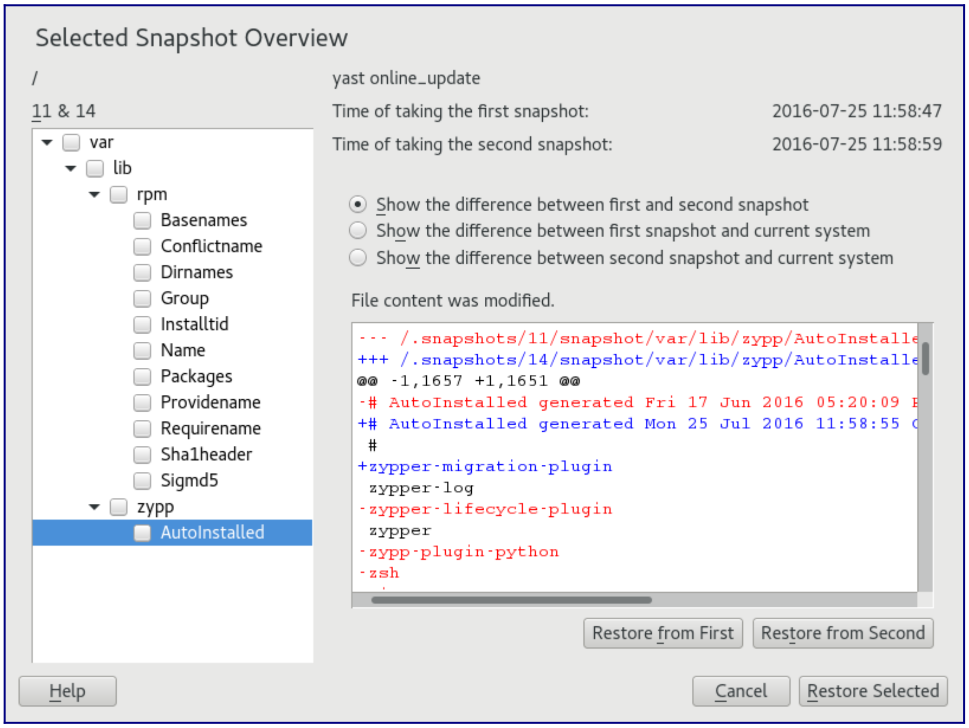
yast online_update

Time of taking the first snapshot: 2016-07-25 11:58:47

Time of taking the second snapshot: 2016-07-25 11:58:59

- □ var
  - □ lib
    - □ rpm
      - □ Basenames
      - □ Conflictname
      - □ Dirnames
      - □ Group
      - □ Installtid
      - □ Name
      - □ Packages
      - □ Providename
      - □ Requirename
      - □ Sha1header
      - □ Sigmd5
    - □ zypp
      - □ AutoInstalled

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



Selected Snapshot Overview

/

11 & 14

yast online_update

Time of taking the first snapshot: 2016-07-25 11:58:47

Time of taking the second snapshot: 2016-07-25 11:58:59

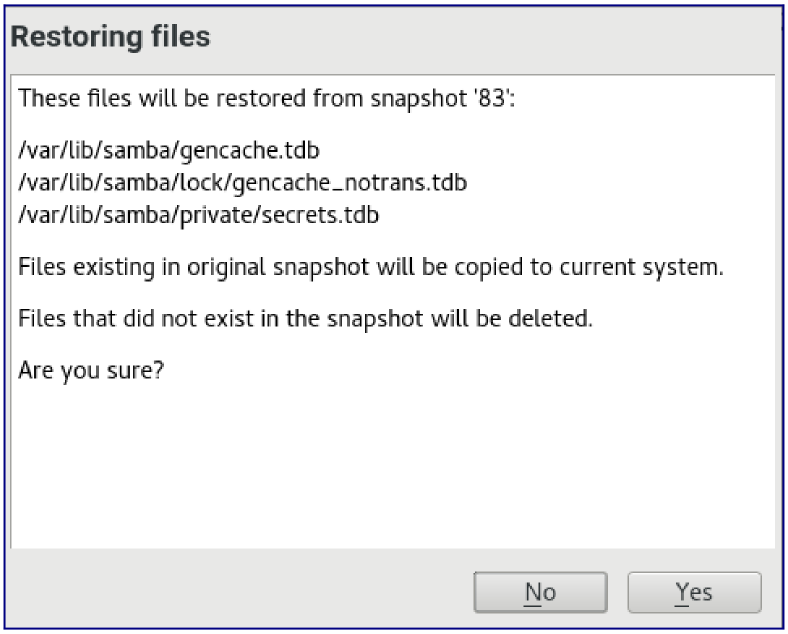
- □ var
  - □ lib
    - □ rpm
      - □ Basenames
      - □ Conflictname
      - □ Dirnames
      - □ Group
      - □ Installtid
      - □ Name
      - □ Packages
      - □ Providename
      - □ Requirename
      - □ Sha1header
      - □ Sigmd5
    - □ zypp
      - □ AutoInstalled

● Show the difference between first and second snapshot
○ Show the difference between first snapshot and current system
○ Show the difference between second snapshot and current system

File content was modified.

```
--- /.snapshots/11/snapshot/var/lib/zypp/AutoInstalle
+++ /.snapshots/14/snapshot/var/lib/zypp/AutoInstalle
@@ -1,1657 +1,1651 @@
-# AutoInstalled generated Fri 17 Jun 2016 05:20:09 
+# AutoInstalled generated Mon 25 Jul 2016 11:58:55 
 #
+zypper-migration-plugin
 zypper-log
-zypper-lifecycle-plugin
 zypper
-zypp-plugin-python
-zsh
```

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



Restoring files

These files will be restored from snapshot '83':

/var/lib/samba/gencache.tdb
/var/lib/samba/lock/gencache_notrans.tdb
/var/lib/samba/private/secrets.tdb

Files existing in original snapshot will be copied to current system.

Files that did not exist in the snapshot will be deleted.

Are you sure?

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

Procedure 3.2: Undoing Changes Using the `snapper` Command Report Bug

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 ; Zypper snapshots are labeled `zypp(zypper)`.

```
tux > sudo snapper list -t pre-post
Pre # | Post # | Pre Date                        | Post Date                       | Description
------+--------+---------------------------------+---------------------------------+--------------
311   | 312    | Tue 06 May 2018 14:05:46 CEST | Tue 06 May 2018 14:05:52 CEST | zypp(y2base)
340   | 341    | Wed 07 May 2018 16:15:10 CEST | Wed 07 May 2018 16:15:16 CEST | zypp(zypper)
342   | 343    | Wed 07 May 2018 16:20:38 CEST | Wed 07 May 2018 16:20:42 CEST | zypp(y2base)
344   | 345    | Wed 07 May 2018 16:21:23 CEST | Wed 07 May 2018 16:21:24 CEST | zypp(zypper)
346   | 347    | Wed 07 May 2018 16:41:06 CEST | Wed 07 May 2018 16:41:10 CEST | zypp(y2base)
348   | 349    | Wed 07 May 2018 16:44:50 CEST | Wed 07 May 2018 16:44:53 CEST | zypp(y2base)
350   | 351    | Wed 07 May 2018 16:46:27 CEST | Wed 07 May 2018 16:46:38 CEST | zypp(y2base)
```

2. Get a list of changed files for a snapshot pair with `snapper status` *PRE..POST*. Files with content changes are marked with , 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/     -p.ttf
+..... /usr/share/fonts/truetype/     -pb.ttf
```

```
+..... /usr/share/fonts/truetype/     -ps.ttf
+..... /usr/share/fonts/truetype/     .ttf
c..... /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache-4
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       2014-
04-23 15:58:57.000000000 +0200
+++ /.snapshots/351/snapshot/usr/share/fonts/truetype/fonts.scale       2014-
05-07 16:46:31.000000000 +0200
@@ -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-iso10646-1
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso8859-1
[...]
```

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/     -p.ttf
     deleting /usr/share/fonts/truetype/     -pb.ttf
     deleting /usr/share/fonts/truetype/     -ps.ttf
     deleting /usr/share/fonts/truetype/     .ttf
     modifying /usr/share/fonts/truetype/fonts.dir
     modifying /usr/share/fonts/truetype/fonts.scale
     modifying /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache-4
     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 tool to remove users.

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 3.4, “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 3.3, “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 3.3: Restoring Files Using the YaST Module Report Bug #

1. Start the module from the section in YaST or by entering `yast2 snapper`.
2. Choose the from which to choose a snapshot.
3. Select a timeline snapshot from which to restore a file and choose . Timeline snapshots are of the type with a description value of .
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 and confirm the action by clicking .

Procedure 3.4: Restoring Files Using the `snapper` Command Report Bug

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
   FILENAME2
   ```

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

3.3 System Rollback by Booting from Snapshots Report Bug

The GRUB 2 version included on openSUSE Leap 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 openSUSE Leap 15.0 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 3.2, “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 Report Bug

- 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, a single partition and a single subvolume. Directories that are excluded from snapshots such as `/srv` (see Section 3.1.2, “Directories That Are Excluded from Snapshots” for a full list) may reside on separate partitions.
- The system needs to be bootable via the installed boot loader.

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

1. Boot the system. In the boot menu choose 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 3.3.1, “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).

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 3.6, “Automatic Snapshot Clean-Up” for details. If the snapshot contains important data, extract the data from the snapshot before it is removed.

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 |     |         |                       |
```

3.3.2 Accessing and Identifying Snapshot Boot Entries Report Bug

To boot from a snapshot, reboot your machine and choose . 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.

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

```
[*]1OS2 (KERNEL3,DATE4TTIME5,DESCRIPTION6)
```

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.

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

3.3.3.1 Directories Excluded from Snapshots Report Bug

Root file system snapshots do not contain all directories. See Section 3.1.2, “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.

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`.

3.4 Creating and Modifying Snapper Configurations Report Bug

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 3.4.1.1, “Configuration Data” for a list of the default values.

Note: Minimum Root File System Size for Enabling Snapshots

As explained in Section 3.1, “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 to automatically enable snapshots during the installation. Currently 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 module to restore files from these snapshots. In YaST you need to select your , 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-data1 create-config /srv/www2
```

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 3.4.1, “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
```

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

List Configurations

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

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

Show a Configuration

Use the subcommand `snapper -c CONFIG get-config` to display the specified configuration. *Config* needs to be replaced by a configuration name shown by `snapper list-configs`. See Section 3.4.1.1, “Configuration Data” for more information on the configuration options.

To display the default configuration run

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

Modify a Configuration

Use the subcommand `snapper -c CONFIG set-config OPTION=VALUE` to modify an option in the specified configuration. *Config* needs to be replaced by a configuration name shown by `snapper list-configs`. Possible values for *OPTION* and *VALUE* are listed in Section 3.4.1.1, “Configuration Data”.

Delete a Configuration

Use the subcommand `snapper -c CONFIG delete-config` to delete a configuration. *Config* needs to be replaced by a configuration name shown by `snapper list-configs`.

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 3.4.1.2, “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 3.6.3, “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 3.6.1, “Cleaning Up Numbered Snapshots” for details.

`QGROUP` / `SPACE_LIMIT`

Adds quota support to the clean-up algorithms. See Section 3.6.5, “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 3.4.1.2, “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 3.6.2, “Cleaning Up Timeline Snapshots” for details.

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 Snapper configurations that grant permissions to users or/and groups can be created. 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 3.5: Enabling Regular Users to Use Snapper Report Bug #

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

1. If not existing, create a Snapper configuration for the partition or subvolume on which the user should be able to use Snapper. Refer to Section 3.4, “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; see Section 3.4.1, “Managing Existing Configurations” for details.

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="yes"
```

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
```

3.5 Manually Creating and Managing Snapshots Report Bug

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 3.4, “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. To create or manage snapshots for your own configuration you need to explicitly choose it. Use the drop-down box in YaST or specify the `-c` on the command line (`snapper -c` *`MYCONFIG COMMAND`*).

Creating a snapshot is done by running `snapper create` or by clicking in the YaST module . The following examples explain how to create snapshots from the command line. It should be easy to adopt them when using the YaST interface.

Tip: Snapshot Description

You should always specify a meaningful description to later be able to identify its purpose. Even more information can be specified via the user data option.

```
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 file 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.

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 module already lists all snapshots. Choose one from the list and click .

```
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.

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

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 3.5.1.2, “Cleanup-algorithms” for details.

Snapshots occupy disk space and over time the amount of disk space occupied by the snapshots may become large. 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 *How to make a snapshot permanent?* .

The default setup (`root`) is configured to do clean-up for numbered snapshots and empty pre and post snapshot pairs. Quota support is enabled—snapshots may not occupy more than 50% of the available disk space of the root partition. Timeline snapshots are disabled by default, therefore the timeline clean-up algorithm is also disabled.

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 3.6.5, “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
```

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 3.1: Example timeline configuration Report Bug

```
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 Report Bug

- 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.

3.6.3 Cleaning Up Snapshot Pairs That Do Not Differ Report Bug

As explained in Section 3.1.1, “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"`.

3.6.4 Cleaning Up Manually Created Snapshots Report Bug

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`.

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 3.4, “Creating and Modifying Snapper Configurations” for more information).

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 timeline 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.

Why does Snapper Never Show Changes in `/var/log`, `/tmp` and Other Directories?

> For some directories we decided to exclude them from snapshots. See Section 3.1.2, “Directories That Are Excluded from Snapshots” for a list and reasons. To exclude a path from snapshots we create a subvolume for that path.

How much disk space is used by snapshots? How to free disk space?

> Displaying the amount of disk space a snapshot allocates is currently not supported by the `Btrfs` tools. However, if you have quota enabled, it is possible to determine how much space would be freed if *all* snapshots would be deleted:

1. Get the quota group ID (`1/0` in the following example):

   ```
   tux > sudo snapper -c root get-config | grep QGROUP
   QGROUP                 | 1/0
   ```

2. Rescan the subvolume quotas:

   ```
   tux > sudo btrfs quota rescan -w /
   ```

3. Show the data of the quota group (`1/0` in the following example):

   ```
   tux > sudo btrfs qgroup show / | grep "1/0"
   1/0           4.80GiB    108.82MiB
   ```

   The third column shows the amount of space that would be freed when deleting all snapshots (`108.82MiB`).

To free space on a `Btrfs` partition containing snapshots you need to delete unneeded snapshots rather than files. Older snapshots occupy more space than recent ones. See Section 3.1.3.4, “Controlling Snapshot Archiving” for details.

Doing an upgrade from one service pack to another results in snapshots occupying a lot of disk space on the system subvolumes, because a lot of data gets changed (package updates). Manually deleting these snapshots after they are no longer needed is recommended. See Section 3.5.4, “Deleting Snapshots” for details.

Can I Boot a Snapshot from the Boot Loader?

Yes—refer to Section 3.3, “System Rollback by Booting from Snapshots” for details.

How to make a snapshot permanent?

Currently Snapper does not offer means to prevent a snapshot from being deleted manually. However, you can prevent snapshots from being automatically deleted by clean-up algorithms. Manually created snapshots (see Section 3.5.2, “Creating Snapshots”) have no clean-up algorithm assigned unless you specify one with `--cleanup-algorithm`. Automatically created snapshots always either have the `number` or `timeline` algorithm assigned. To remove such an assignment from one or more snapshots, proceed as follows:

1. List all available snapshots:

   ```
   tux > sudo snapper list -a
   ```

2. Memorize the number of the snapshot(s) you want to prevent from being deleted.

3. Run the following command and replace the number placeholders with the number(s) you memorized:

```
tux > sudo snapper modify --cleanup-algorithm "" #1 #2 #n
```

4. Check the result by running `snapper list -a` again. The entry in the column `Cleanup` should now be empty for the snapshots you modifie

5. Where can I get more information on Snapper?

See the Snapper home page at http://snapper.io/.