

# Snapper

## Introduction

This article assumes you will be using Snapper with the Btrfs filesystem.

During a fresh install of openSUSE, the default is to use Btrfs on root, meaning / and the filesystem XFS on /home. If you select the option to not separate /home during openSUSE install, the default is for everything, including /home, to be under Btrfs.

If you decide to use Btrfs & Snapper (default) during a fresh openSUSE install, it will automatically configure Snapper for you. This automatic configuration of Snapper means that snapshots will be made automatically when you use YaST2 and/or zypper. Please read section *Software Requirements* below for caveat.

You may convert EXT3 & EXT4 filesystems to Btrfs to take advantage of Snapper. Using Snapper on EXT4 filesystem is *highly* discouraged.

## Important Details

The default behaviour when Snapper is configured to run on root, meaning /, is to **exclude** every Btrfs subvolume. This the same behaviour as on SUSE Enterprise. This means that if you have the subvolume /srv Snapper will not snapshot anything in /srv unless told specifically by you to do so. This also means that if you add a subvolume to / at any point in time, it will be excluded from Snapper's default snapshot behaviour.

Always be root when using Snapper or YaST2, unless otherwise specified.

YaST2 comes with a Snapper module called yast2-snapper and may be used to check and manipulate snapshots. It does not, however, have the ability to change Snapper configurations. That must be done in the terminal.

Taking snapshots is automatically enabled if the root partition (/) is big enough (approximately more than 8GB). Please remember: taking snapshots on partitions other than / is not enabled by default.

To see what subvolumes are created under / and therefore see which directories (subvolumes) are excluded from the default snapshots behaviour:

```
# btrfs subvolume list /
```

To have Snapper create a snapshot of a subvolume, you can either do it manually one-time via the snapper program, or create a new snapper config file. More information in the tutorial.

# Software Requirements

You always need this package:

```
snapper
```

If you use YaST2 to install, update or remove packages and want to have snapper automatically create snapshots when you use this tool:

```
yast2-snapper
```

If you use zypper to install, update or remove packages and want to have snapper automatically create snapshots when you use this tool:

```
snapper-zypp-plugin
```

If you want to have the ability for advanced btfs snapshot boot menu management:

```
grub2-snapper-plugin
```

**WARNING:** the above packages are not always installed by default. What gets installed depends on what choices you make during the openSUSE installation process. Make sure you have the appropriate (usually that means all) packages listed above installed before making changes to your system, otherwise you may be in for a surprise when snapper does not automatically create snapshots you thought it would.

# Snapper Basic Usage

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. 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`).

# List all snapshots for the default configuration (root)

```
snapper list
```

or

```
snapper ls
```

# Show which files and directories have been changed between snapshots

For files/directories changed between snapshot 21 and 22:

```
snapper status 21..22
```

You can also show what files and directories have been changed from snapshot 41 going back to 39:

```
snapper status 41..39
```

The output consists of a string encoding the status followed by the filename. The characters of the status string are:

1. "+" means the file was created
2. "-" means the file was deleted
3. "c" means the content of the file has changed
4. "t" means the type of the file has changed (e.g. from regular file to directory)
5. "p" means the permissions are have changed
6. "u" means the user ownership has changed
7. "g" means the group ownership has changed
8. "x" means the extended attribute information has changed
9. "a" means the ACL information has changed
10. "." means no change

# Show the diff (difference in actual files) between snapshots

For diff between snapshot 71 and 72:

```
snapper diff 71..72
```

You can also use this command to show the difference between a specific file. For example to show the diff from snapshot 71 to 72 only for file `/etc/zypp/zypp.conf` you would do the following:

```
snapper diff 71..72 /etc/zypp/zypp.conf
```

## Cleanup Algorithms

Unless you have a good reason to do otherwise, you should always specify the cleanup algorithm when creating a snapshot, otherwise the snapshot will never be deleted unless you do it manually. You do this by adding the following to your snapper commands:

```
--cleanup-algorithm <number|timeline|empty-pre-post>
```

The following commands assume you will be creating snapshots on the default system configuration (root). As mentioned above, if you want to use the non-default configuration, add the following to the snapshot commands:

```
-c MYCONFIG
```

Create 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.

```
snapper create --type pre --print-number --description "Before the Apache config cleanup" --cleanup-algorithm number
```

Create 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.

```
snapper create --type post--pre-number 30 --description "After the Apache config cleanup" --cleanup-algorithm number
```

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 create --description "Snapshot for week 2 2014"
```

## Deleting Snapshots

Delete snapshot 65 for the default (root) configuration:

```
snapper delete 65
```

## Snapper Tutorial

First let us check how YaST has configured snapper:

```
# snapper list-configs
Config | Subvolume
-----+-----
root   | /
```

As you can see YaST has created a snapper config called "root" for your root file system. You can see what snapshots exist:

```
# snapper list
Type   | # | Pre # | Date                               | Cleanup | Description | Userdata
-----+---+-----+-----+-----+-----+-----
single | 0 |      |                                     |         | current     |
single | 1 |      | Tue 22 Nov 2011 10:30:02 CET       | timeline | timeline    |
```

Snapshot #0 always refers to the current system. There might already be several other snapshots depending on the uptime of your system and on whether you already used YaST or zypper.

Now we want to try snapper as a undo tool for YaST. First we run YaST:

```
# yast2 system_settings
```

Within YaST enable the SysReq. After you have finished YaST you see two new snapshots:

```
# snapper list
Type   | # | Pre # | Date                               | Cleanup | Description | Userdata
-----+---+-----+-----+-----+-----+-----
single | 0 |      |                                     |         | current     |
single | 1 |      | Tue 22 Nov 2011 10:30:02 CET       | timeline | timeline    |
pre    | 2 |      | Tue 22 Nov 2011 10:41:28 CET       | number  | yast system_settings |
post   | 3 | 2     | Tue 22 Nov 2011 10:41:49 CET       | number  |              |
```

## Types of Snapshots

It's time to explain the type of snapshots. Snapper creates a snapshot before and after YaST runs, these snapshots are called pre and post respectively. The post snapshots knows which pre snapshots belongs to it. By having a pre and post snapshot we can see what changes happened to the file system while YaST was running. Single snapshots have no special relationship to other snapshots.

## Seeing What Has Changed (YaST)

Using snapper, it is easy to see what has changed while YaST was running. To do so you have to pass the number of the pre and post snapshot:

```
# snapper status 2..3
c... /etc/sysctl.conf
```

The 'c' means that the content of the file has changed.

We can also see the diff of the file:

```
# snapper diff 2..3
--- /.snapshots/2/snapshot/etc/sysctl.conf      2011-11-22 10:35:43.355493753 +0100
+++ /.snapshots/3/snapshot/etc/sysctl.conf      2011-11-22 10:41:47.019512185 +0100
@@ -17,7 +17,7 @@
# See sysctl.conf(5) and sysctl(8) for more information
#
####
-kernel.sysrq = 0
+kernel.sysrq = 1
net.ipv4.ip_forward = 0
net.ipv4.tcp_syncookies = 1
net.ipv6.conf.all.forwarding = 0
```

So, if you don't like the change done by YaST and want to revert it call:

```
# snapper undochange 2..3
create:0 modify:1 delete:0
undoing change...
undoing change done
```

But note that snapper does not tell the kernel about the change like YaST did so you must either do so yourself or reboot.

## Adding /home To Snapper

During installation YaST does not setup a snapper config for /home. We can do so manually:

```
# snapper -c home create-config /home
```

Whenever you want to use snapper for /home you must provide the option -c home.

Now you can see what files have changed since the last hourly snapshot:

```
# snapper -c home list
Type   | # | Pre # | Date                               | Cleanup | Description | Userdata
-----+-----+-----+-----+-----+-----+-----
single | 0 |      |                                     |          | current     |
single | 1 |      | Tue 22 Nov 2011 11:30:01 CET       | timeline | timeline    |

# snapper -c home status 1..0
comparing snapshots... done
+... /home/tux/just-married.jpg
```

The '+' sign means that the file is new.

# Manually Creating A Snapshot

If you manually want to create a snapshot use:

```
# snapper -c home create --description "before the big cleanup"
```

# YaST Snapper GUI

Finally yast2-snapper provides a YaST UI for snapper.

# Troubleshooting

## Logs

Snapper logs can be found in `/var/log/snapper.log`, check this file if you are having issues as it may contain useful troubleshooting information.

# Snapper Videos

[Snapper Quick Overview \(4min\)](#)

[Greg Kroah-Hartman & SUSE Linux Enterprise Product Manager Demo Snapper \(~7min\)](#)

[Snapper, Btrfs & LVM Presentation \(22min\)](#)

---

Revision #3

Created 2026-04-01 17:13:54 CEST by Philip

Updated 2026-04-13 19:25:47 CEST by Philip