

# How to setup rsnapshot on Ubuntu 16.04

```
sudo apt-get install rsnapshot
```

```
sudo ssh-keygen -t rsa
sudo ssh-copy-id -i /root/.ssh/id_rsa.pub root@example.com
sudo ssh root@example.com
```

## Install Rsnapshot

```
sudo apt-get install rsnapshot rsync
```

## Create a backup directory

```
sudo mkdir -p /raidbackup/rsnapshot
```

## Configuring Rsnapshot

```
sudo cp /etc/rsnapshot.conf /etc/rsnapshot.conf.bak
sudo vi /etc/rsnapshot.conf
```

```
#####
# rsnapshot.conf - rsnapshot configuration file #
#####
#
# PLEASE BE AWARE OF THE FOLLOWING RULES:
#
# This file requires tabs between elements
#
# Directories require a trailing slash:
#   right: /home/
#   wrong: /home
#
#####

#####
# CONFIG FILE VERSION #
#####

config_version 1.2

#####
# SNAPSHOT ROOT DIRECTORY #
#####

# All snapshots will be stored under this root directory.
#
# The first decision you will need to make is where you would like to store
your backups. We will use the directory "/backup" as our backup location.
```

```
Search for and edit the following variable to set the backup location.
#snapshot_root /var/cache/rsnapshot/
snapshot_root /raidbackup/rsnapshot/

# If no_create_root is enabled, rsnapshot will not automatically create the
# snapshot_root directory. This is particularly useful if you are backing
# up to removable media, such as a FireWire or USB drive.
#
#no_create_root 1

#####
# EXTERNAL PROGRAM DEPENDENCIES #
#####

# LINUX USERS: Be sure to uncomment "cmd_cp". This gives you extra features.
# EVERYONE ELSE: Leave "cmd_cp" commented out for compatibility.
#
# See the README file or the man page for more details.
#
cmd_cp /bin/cp

# uncomment this to use the rm program instead of the built-in perl routine.
#
cmd_rm /bin/rm

# rsync must be enabled for anything to work. This is the only command that
# must be enabled.
#
cmd_rsync /usr/bin/rsync

# Uncomment this to enable remote ssh backups over rsync.
#
cmd_ssh /usr/bin/ssh

# Comment this out to disable syslog support.
#
cmd_logger /usr/bin/logger

# Uncomment this to specify the path to "du" for disk usage checks.
# If you have an older version of "du", you may also want to check the
# "du_args" parameter below.
#
cmd_du /usr/bin/du

# Uncomment this to specify the path to rsnapshot-diff.
#
#cmd_rsnapshot_diff /usr/bin/rsnapshot-diff

# Specify the path to a script (and any optional arguments) to run right
# before rsnapshot syncs files
#
#cmd_preexec /path/to/preexec/script
```

```

# Specify the path to a script (and any optional arguments) to run right
# after rsnapshot syncs files
#
#cmd_postexec    /path/to/postexec/script

# Paths to lvcreate, lvremove, mount and umount commands, for use with
# Linux LVMs.
#
#linux_lvm_cmd_lvcreate /sbin/lvcreate
#linux_lvm_cmd_lvremove /sbin/lvremove
#linux_lvm_cmd_mount    /bin/mount
#linux_lvm_cmd_umount   /bin/umount

#####
#             BACKUP INTERVALS             #
# Must be unique and in ascending order #
# i.e. hourly, daily, weekly, etc.      #
#####

#retain    hourly  6
#retain    daily   7
retain     weekly  4
retain     monthly 3

#####
#             GLOBAL OPTIONS                #
# All are optional, with sensible defaults #
#####

# Verbose level, 1 through 5.
# 1      Quiet          Print fatal errors only
# 2      Default        Print errors and warnings only
# 3      Verbose         Show equivalent shell commands being executed
# 4      Extra Verbose   Show extra verbose information
# 5      Debug mode      Everything
#
verbose    2

# Same as "verbose" above, but controls the amount of data sent to the
# logfile, if one is being used. The default is 3.
#
loglevel    3

# If you enable this, data will be written to the file you specify. The
# amount of data written is controlled by the "loglevel" parameter.
#
#logfile     /var/log/rsnapshot.log

# If enabled, rsnapshot will write a lockfile to prevent two instances
# from running simultaneously (and messing up the snapshot_root).
# If you enable this, make sure the lockfile directory is not world

```

```
# writable. Otherwise anyone can prevent the program from running.
#
lockfile    /var/run/rsnapshot.pid

# By default, rsnapshot check lockfile, check if PID is running
# and if not, consider lockfile as stale, then start
# Enabling this stop rsnapshot if PID in lockfile is not running
#
#stop_on_stale_lockfile    0

# Default rsync args. All rsync commands have at least these options set.
#
#rsync_short_args    -a
#rsync_long_args     --delete --numeric-ids --relative --delete-excluded
rsync_long_args     --delete --numeric-ids --relative --delete-excluded --
bwlimit=5000

# ssh has no args passed by default, but you can specify some here.
#
#ssh_args    -p 22

# Default arguments for the "du" program (for disk space reporting).
# The GNU version of "du" is preferred. See the man page for more details.
# If your version of "du" doesn't support the -h flag, try -k flag instead.
#
#du_args      -csh

# If this is enabled, rsync won't span filesystem partitions within a
# backup point. This essentially passes the -x option to rsync.
# The default is 0 (off).
#
#one_fs      0

# The include and exclude parameters, if enabled, simply get passed directly
# to rsync. If you have multiple include/exclude patterns, put each one on a
# separate line. Please look up the --include and --exclude options in the
# rsync man page for more details on how to specify file name patterns.
#
#include      ???
#include      ???
#exclude     ???
#exclude     ???

# The include_file and exclude_file parameters, if enabled, simply get
# passed directly to rsync. Please look up the --include-from and
# --exclude-from options in the rsync man page for more details.
#
#include_file  /path/to/include/file
#exclude_file /path/to/exclude/file

# If your version of rsync supports --link-dest, consider enable this.
# This is the best way to support special files (FIFOs, etc) cross-platform.
```

```
# The default is 0 (off).
#
#link_dest 0

# When sync_first is enabled, it changes the default behaviour of rsnapshot.
# Normally, when rsnapshot is called with its lowest interval
# (i.e.: "rsnapshot hourly"), it will sync files AND rotate the lowest
# intervals. With sync_first enabled, "rsnapshot sync" handles the file sync,
# and all interval calls simply rotate files. See the man page for more
# details. The default is 0 (off).
#
#sync_first 0

# If enabled, rsnapshot will move the oldest directory for each interval
# to [interval_name].delete, then it will remove the lockfile and delete
# that directory just before it exits. The default is 0 (off).
#
#use_lazy_deletes 0

# Number of rsync re-tries. If you experience any network problems or
# network card issues that tend to cause ssh to crap-out with
# "Corrupted MAC on input" errors, for example, set this to a non-zero
# value to have the rsync operation re-tried
#
#rsync_numtries 0

# LVM parameters. Used to backup with creating lvm snapshot before backup
# and removing it after. This should ensure consistency of data in some special
# cases
#
# LVM snapshot(s) size (lvcreate --size option).
#
#linux_lvm_snapshotsizesize 100M

# Name to be used when creating the LVM logical volume snapshot(s).
#
#linux_lvm_snapshotname rsnapshot

# Path to the LVM Volume Groups.
#
#linux_lvm_vgpath /dev

# Mount point to use to temporarily mount the snapshot(s).
#
#linux_lvm_mountpath /path/to/mount/lvm/snapshot/during/backup

#####
### BACKUP POINTS / SCRIPTS ###
#####

backup root@server.edu:/etc/ backupetc/
backup root@server.edu:/export/ backupexport/
```

---

Once you have made all your changes, run the following command to verify that the config file is syntactically valid.

```
rsnapshot configtest
```

If all is well, you will see the following output.

```
Syntax OK
```

## Testing backups

Run the following command to test backups.

```
rsnapshot weekly  
#rsnapshot monthly
```

## Verifying backups

Check the whether the backups are really stored in the Root backup directory in the Backup server.

```
ls /rsnapbackup/
```

You will see the following **output**:

```
weekly.0
```

Check the **weekly.0 directory**:

```
ls /rsnapbackup/weekly.0/
```

## Add crontab job

```
$ crontab -e
```

```
# m h dom mon dow    command
```

```
0 21 * * 5 [ $(date +%d) -le 07 ] && /usr/bin/rsnapshot weekly
```

```
0 21 * * 5 [ $(date +%d) -le 07 ] && /usr/bin/rsnapshot monthly
```

<https://www.ostechnix.com/setup-backup-server-using-rsnapshot-linux/>