

# borgbackup

- <https://borgbackup.readthedocs.io/en/stable/index.html>
- <https://www.atarea.es/podcast/hice-un-rm-rf-salvado-por-borg/>

## repositorio

- creación repositorio:

```
borg init --encryption=[none|repokey] [--make-parent-dirs] [--storage-quota QUOTA ] /path/to/repo
```

- backup repokey:

```
borg key export [--paper] [--qr-html] <REPO> /path/to/repokey.export
```

- información del repositorio o archivo:

```
borg info /path/to/repo
```

## archivo (de backup)

- creación archivo de copia:

```
borg create [--stats] /path/to/repo::<archivo> /path/files/tobe/backuped1  
[path/files/tobe/backuped2]
```

- parámetros:

- --list: muestra información de los ficheros que va evaluando. Por defecto no muestra nada
- --filter=<CHARS>: filtra los archivos que se muestran en --list:

```
'A' = regular file, added  
'M' = regular file, modified  
'U' = regular file, unchanged  
'E' = regular file, an error happened while accessing/reading this file
```

A lowercase character means a file type other than a regular file, borg usually just stores their metadata:

```
'd' = directory  
'b' = block device  
'c' = char device  
'h' = regular file, hardlink (to already seen inodes)  
's' = symlink  
'f' = fifo
```

Other flags used include:

```
'i' = backup data was read from standard input (stdin)  
'-' = dry run, item was not backed up  
'x' = excluded, item was not backed up  
'?' = missing status code (if you see this, please file a bug report!)
```

- --exclude-caches: ignora directorios con la firma **CACHEDIR.TAG** (<https://bford.info/cachedir/>)

- `--exclude-from <EXCLUDE_FILE>`: ignora los ficheros/directorios según el patrón
- `--exclude-if-present <NAME>`: nombre del fichero que hace ignorar la copia de directorios y subdirectorios incluidos
  - `--keep-exclude-tags` invierte el uso del `exclude` a `include` (solo se haría copia de los que incluyen el fichero-tag)

## listar & info

- listar archivos en repositorio:

```
borg list /path/to/repo
```

- listar contenido de un archivo:

```
borg list /path/to/repo::
```

- información de repositorio:

```
borg info /path/to/repo
```

- información de un archivo del repositorio:

```
borg info /path/to/repo::
```

## recuperación

- extraer un archivo:

```
borg extract /path/to/repo::
```

## montar

- <https://borgbackup.readthedocs.io/en/stable/usage/mount.html>
- fuse.borgfs (uso en `/etc/fstab`)
- montar el repo:

```
borg mount /path/to/repo /mount/point
```

- hay un parámetro experimental (y versión 1.1.14?) que mezcla todos los archivos del repo en una «vista»: **-o versions**
- `--glob-archives '*-my-home'`: «include» archivos
- `--last 10`
- `--exclude`

- montar un archivo del repo:

```
borg mount:: /path/to/repo /mount/point
```

## desmontar

- `borg umount /mount/point`

## exportar

- `borg export-tar /path/to/repo::<archivo> fichero.tar`
- <https://borgbackup.readthedocs.io/en/stable/usage/tar.html>

## prune

- aka retención
- ejecutar siempre pruebas con `--list --dry-run`
- eliminar copias que no cumplan los criterios de retención establecidos
- parámetros:
  - `--keep-last, --keep-secondly #`: número de segundos archivos a mantener
  - `--keep-minutely #`: number of minutely archives to keep
  - `-H, --keep-hourly #`: number of hourly archives to keep
  - `-d, --keep-daily #`: number of daily archives to keep
  - `-w, --keep-weekly #`: number of weekly archives to keep
  - `-m, --keep-monthly #`: number of monthly archives to keep
  - `-y, --keep-yearly #`: number of yearly archives to keep

- `borg prune \`  
    `--keep-daily 7 \`  
    `--keep-weekly 4 \`  
    `--keep-monthly 12 \`  
    `--keep-yearly 1 \`  
`/path/to/repo`

- mantiene copia de los últimos 7 días
- mantiene copia de uno de los días de las últimas 4 semanas
- mantiene copia de un día de los últimos 12 meses
- mantiene copia de un día del último año

## borrado

- borrado de repositorio y archivos:

```
borg delete /path/to/repo::<archivo>
```

## automatización

- especificar el repositorio:

```
export BORG_REPO=ssh://username@example.com:2022/~//backup/main
```

```
export BORG_REPO=/path/to/repo
```

- especificar la contraseña:

```
export BORG_PASSPHRASE='XYZl0ngandsecurepa_55_phrasea&&123'
```

- si el comando **borg** se lanza con **sudo**, usar **-preserve-env** si se está exportando el repositorio o la contraseña previamente.
- se pueden usar **placeholders**
  - {hostname}, {fqdn}, {reverse-fqdn}
  - {now}, {now:%Y-%m-%d}, {utcnow}
  - {user}, {pid}
  - {borgversion}, {borgmajor}, {borgminor}, {borgpatch}
- variables de entorno:
  - <https://borgbackup.readthedocs.io/en/1.1.11/usage/general.html#environment-variables>

## include, exclude y patterns

- *selectores de estilo* para los patterns:
  - Fmatch → fn:
  - Shell-style patterns → sh:
    - Fmatch + \*\* y \*
  - Regular expressions → re:
  - Path prefix → pp:
  - Path full-match → pf:

### exclude

- selector de estilo por defecto: **fn**:

### patterns

- selector de estilo por defecto: **sh**:
- <https://manpages.debian.org/testing/borgbackup/borg-patterns.1.en.html>
- <https://borgbackup.readthedocs.io/en/stable/usage/help.html#borg-help-patterns>
- permite seleccionar inclusiones (+) y exclusiones (-)
- P indica el estilo de pattern a aplicar según la lista anterior
- R indica el/los *root path* ← permite omitir el path del backup en **borg create**
- + incluye
- - excluye → indicar \*\*/ para archivos y directorios
- ! excluye recursivamente
- [patterns.lst](#)

```
# "sh:" pattern style is the default, so the following line is not
needed:
P sh
R /
# can be rebuild
- /home/*.cache
# they're downloads for a reason
- /home/*/Downloads
# susan is a nice person
# include susans home
+ /home/susan
# don't backup the other home directories
```

```
- /home/*  
# don't even look in /proc  
! /proc
```

## common options

```
--critical: work on log level CRITICAL  
--error: work on log level ERROR  
--warning: work on log level WARNING (default)  
--info, -v, --verbose: work on log level INFO  
--debug: enable debug output, work on log level DEBUG  
--debug-topic TOPIC: enable TOPIC debugging (can be specified multiple times). The  
logger path is borg.debug.<TOPIC> if TOPIC is not fully qualified.  
--progress, -p: show progress information  
--log-json: Output one JSON object per log line instead of formatted text.  
--lock-wait SECONDS: wait at most SECONDS for acquiring a repository/cache lock  
(default: 1).  
--show-version: show/log the borg version  
--show-rc: show/log the return code (rc)  
--umask M: set umask to M (local and remote, default: 0077)  
--remote-path PATH: use PATH as borg executable on the remote (default: "borg")  
--remote-ratelimit RATE: set remote network upload rate limit in kiByte/s (default:  
0=unlimited)  
--consider-part-files: treat part files like normal files (e.g. to list/extract  
them)  
--debug-profile FILE: Write execution profile in Borg format into FILE. For local  
use a Python-compatible file can be generated by suffixing FILE with ".pyprof".  
--rsh RSH: Use this command to connect to the 'borg serve' process (default: 'ssh')
```

## scripts

```
# Backup a remote host locally ("pull" style) using sshfs  
mkdir sshfs-mount  
sshfs root@example.com:/ sshfs-mount  
cd sshfs-mount  
borg create /path/to/repo::example.com-root-{now:%Y-%m-%d} .  
cd ..  
fusermount -u sshfs-mount
```

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/software:backup:borg?rev=1605473965>

Last update: **15/11/2020 12:59**

