# BorgBackup Deduplicating backups

## CONCEPTS

> Set BORG_REPO and BORG_PASSCOMMAND once; then use :: instead of the repo path (borg create ::<archive> <path>).

**Files** your data
↓ create
**Archive** deduplicated, encrypted
↓ prune
**Deleted archive** space still used
↓ compact
**Space freed** repo shrinks

> Every archive looks like a full backup, but identical chunks are stored once, so later runs are small and fast.

> Lose the key or passphrase and the data is gone. Keep a key export and the passphrase outside the repo and the backed-up machine.

## REPOSITORY

**borg init -e** *<mode>* *<repo>*
Create a repository

**borg init -e repokey-blake2** *<repo>*
Encrypted, BLAKE2 auth (no SHA hardware)

**export BORG_REPO=***<repo>*
Default repository

**export BORG_PASSCOMMAND="***<cmd>***"**
Get the passphrase from a command

**export BORG_PASSPHRASE=***<passphrase>*
Passphrase from the environment

**ssh://***<user>***@***<host>***:***<port>***/***<path>*
Remote repository URL

**export BORG_RSH="ssh -i** *<key>***"**
Custom ssh command

**borg info** *<repo>*
Repository size and encryption

**borg key export** *<repo>* *<file>*
Back up the key

## REPOSITORY (cont.)

**borg key change-passphrase** *<repo>*
Change the passphrase

## CREATE

**borg create** *<repo>***::***<archive>* *<path>***...**
Back up paths into a new archive

**borg create --stats --progress** *<repo>***::***<archive>* *<path>*
Show statistics and progress

**borg create --list --dry-run** *<repo>***::***<archive>* *<path>*
Preview what would be stored

**borg create ...** *<repo>***::{hostname}-{now:%Y-%m-%d}** *<path>*
Name archives by host and date

**borg create -e '***<pattern>***' ...**
Exclude matching paths

**borg create --exclude-from** *<file>* **...**
Read excludes from a file

**borg create --exclude-caches ...**
Skip dirs with CACHEDIR.TAG

**borg create --exclude-if-present** *<name>* **...**
Skip dirs holding a marker file

**borg create -x ...**
Stay on one filesystem

**borg create -C zstd,6 ...**
Choose compression

**borg create --patterns-from** *<file>* **...**
Include/exclude rules from a file

### Pattern styles

| Prefix | Meaning |
|---|---|
| `fm:` | shell-like, default for --exclude |
| `sh:` | like fm, ** crosses dirs, * does not |
| `re:` | regular expression, substring match |
| `pp:` | path prefix, dir and contents |
| `pf:` | exact full path |

**-** *<pattern>*
Patterns file: exclude

## BROWSE AND RESTORE

**borg list** *<repo>*
List archives

**borg list** *<repo>***::***<archive>*
List files of an archive

**borg list --short** *<repo>***::***<archive>*
Only paths

**borg list --last** *<n>* *<repo>*
Only the newest archives

**borg extract** *<repo>***::***<archive>*
Restore everything into current dir

**borg extract** *<repo>***::***<archive>* *<path>*
Restore only one path

**borg extract --dry-run --list** *<repo>***::***<archive>*
Check a restore without writing

**borg mount** *<repo>***::***<archive>* *<dir>*
Mount an archive via FUSE

**borg umount** *<dir>*
Unmount

## RETENTION AND UPKEEP

**borg prune -v --list --dry-run** *<keep-options>* *<repo>*
Preview pruning

**borg prune --keep-daily 7 --keep-weekly 4 --keep-monthly 6** *<repo>*
Apply a retention policy

**borg prune -a '{hostname}-*' ...** *<repo>*
Only touch one host's archives

**borg compact** *<repo>*
Free space after prune/delete

**borg delete** *<repo>***::***<archive>*
Delete one archive

**borg check** *<repo>*
Verify repo and archive consistency

## COMMON OPTIONS

**-v / --info**
More output