

BorgBackup

Danny Robson

Poll

- Who is actually running backups?
- Who is actually running backups *regularly*?
- Who is actually running backups regularly and *verifying* them?

My previous system

- python
- btrfs
- optimism

Problems

- I don't have time/interest in maintaining scripts
- I hate dealing with btrfs
- If it's easy I *might actually perform backups*

Requirements

- **Must** be packaged for:
 - Gentoo and Arch
 - ARM and AMD64
- Minimal system configuration
- Space and network efficient
- Locally hosted

Other Candidates

bacula

Eyewateringly complicated.

bup

Personal prejudice against git packs.

rdiff-backup, duplicity, rsnapshot

Personal prejudice against rdiff

Features

- Security: aes, hmac-sha256, ssh.
- Compression: lz4, zstd, zlib, lzma
- Deduplication
- Networked: ssh transport

Simplicity

```
DST=$HOSTNAME@backup:/srv/backup/$HOSTNAME
```

```
borg init -e none $DST
```

```
borg create $DST::{now} $SRC
```

```
borg prune -d 7 -w 4 -m 6 -y 2
```

```
borg check $DST
```


Live Demo



Variables

```
REPO="$HOSTNAME@backup:/srv/backup/$HOSTNAME"  
PREFIX="home"  
SUFFIX='{now:%Y%m%d-%H%Mh%S}'  
ARCH="${REPO}::${PREFIX}::${SUFFIX}"
```

repository

a path to store a collection of related backups

archive

a named set of files from at one location and time

Init

Create a 'repository', a place to store our backups.

```
borg init -e {none,authenticated,repokey,...} /srv/backup/munro
```

```
borg init -e authenticated /srv/backup/munro
```

Create

Create an 'archive', a set of files we can restore later.

```
borg create <dst> <src> [...]
```

```
borg --progress --one-file-system --exclude /home --exclude-cache
```

Prune

```
borg prune ${REPO} [--prefix <name>] --keep-{daily,weekly,monthly
```

```
borg prune ${REPO} --prefix home:: -d 7 -w 4 -m 6 -y 2
```

Info/List

```
borg info <${REPO}|borg info ${REPO}::${ARCHIVE}>
```

```
borg list <${REPO}|${REPO}::${ARCHIVE}>
```

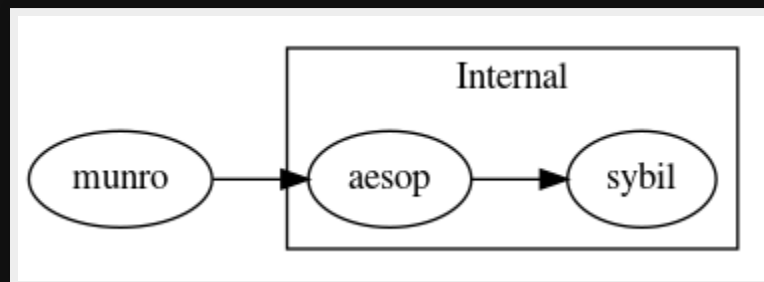
Extract

```
borg extract ${REPO}::${ARCHIVE} <dst>
```

Mount

```
borg mount ${REPO}::${ARCHIVE} <mnt>  
borg umount <mnt>
```


Automation



```
export BORG_PASSPHRASE='hunter2'  
export BORG_PASSCOMMAND='cat /totally_secure'  
  
REPO="backup@sybil:/srv/backup/$HOSTNAME"  
ARCH='${REPO}::root::{now}'  
  
borg create ${ARCH} /  
borg prune ${REPO} --prefix root:: -d 7 -w 4 -m 6 -y 2  
borg check ${REPO} || freakout
```

aesop:.ssh/authorized_keys

```
command="/usr/bin/env nc sybil 22",restrict ssh-ed25519 totallyle
```

sybil:~/.ssh/authorized_keys]

```
command="borg serve --restrict-to-path /srv/backup/munro",restrict
```

Caveats

- One repository per host
- No simultaneous access to a repo
- Data format isn't as simple as other projects
- Not great for Windows

Other neat tricks

- Append only backups
- `borg with-lock <cmd> [...
]`
- Various compression options
- `borgmatic`