Borg Backup

Christian Neukirchen, Tristan Helmich

DevOps Meetup Munich · May 16, 2017
Why backups?

• Nobody wants *backup*, everyone wants *restore*.
• Who is running backups of their personal systems?
• Typical pain points:
  • Off-site storage
  • Storage overhead
  • Privacy & Confidentiality
  • Maintenance cost
  • Vendor lock-in
Why use Borg Backup?

Off-site storage

  • SSH transport (needs borg on remote)

Storage overhead

  • Compression
  • Content deduplication
  • Deleting archives is possible

Privacy & Confidentiality

  • Client-side encryption
  • Encrypted transport
Why use Borg Backup?

Maintenance cost

- Configurable pruning
- Non-interactive use possible
- Partial backups can be resumed

Vendor lock-in

- Open Source (3-clause-BSD)
- Python 3 (portable: Linux, *BSD, macOS)
How to use Borg Backup?

```bash
borg init user@hostname:backup

borg create user@hostname:backup::work-{now:%Y-%m-%d} /home/me --exclude '*~' --compression zlib

rm -rf /home/me/work  # oops

borg list user@hostname:backup
cd /home/me
borg extract user@hostname:backup::work-2017-05-16 work
```
More features

• FUSE file system to browse contents of whole repository
• Time-based pruning (how many archives to keep per hour, day, week, month, year)
• Good metadata support: {a,c,m}times with nanosecond precision, xattrs (incl. caps and ACL), device nodes
• Backup/restore of streams via stdin/stdout
• Append-only repos
• Meta-data and data checking is possible (cryptographic signatures)
• Client-side encryption: passphrase (default), keyfile, none
  • also of meta data (file names, sizes)
Content deduplication

• Renaming and copying file is essentially free.
• Not only file-based deduplication, but content-based chunking.
• Detects insertions/deletion of parts of files.
• Great for:
  • append-only files, logs
  • VM images
  • mbox (even with deletions)
• Multiple hosts can backup into same archive
• Chunking is very customizable (if needed)
# Content deduplication results

57 backups of a full notebook since 2014:

<table>
<thead>
<tr>
<th></th>
<th>Original size</th>
<th>Compressed size</th>
<th>Deduplicated size</th>
</tr>
</thead>
<tbody>
<tr>
<td>This archive:</td>
<td>143.27 GB</td>
<td>115.84 GB</td>
<td>1.34 GB</td>
</tr>
<tr>
<td>All archives:</td>
<td>6.87 TB</td>
<td>5.66 TB</td>
<td>169.10 GB</td>
</tr>
</tbody>
</table>

384 backups of a development workstation since Mar 2016:

<table>
<thead>
<tr>
<th></th>
<th>Original size</th>
<th>Compressed size</th>
<th>Deduplicated size</th>
</tr>
</thead>
<tbody>
<tr>
<td>This archive:</td>
<td>162.25 GB</td>
<td>112.37 GB</td>
<td>246.65 MB</td>
</tr>
<tr>
<td>All archives:</td>
<td>45.87 TB</td>
<td>29.16 TB</td>
<td>346.78 GB</td>
</tr>
</tbody>
</table>

Unique chunks | Total chunks
---           |          
Chunk index:  | 9359546   | 903415142
### Downsides

- Needs free local disk space to create a backup (and for all other writing operations)
- No support for dumb remote storage, needs SSH and Borg installed
- Locking: no restore when a backup is running
- Sub-optimal for long-time-archiving (complex data format, needs `borg` to restore)
Alternatives

- Attic (predecessor to Borg, unmaintained, has critical bugs)
- Restic (very similar to Borg, in Go, beta)
- Duplicati (many dumb backends, in C#)
- rdedup (in Rust, public-key, no remote)
Questions?

Borg

https://borgbackup.readthedocs.io/

Thank you.