

Advanced PostgreSQL backup & recovery methods

Anastasia Lubennikova
pgconf.eu 2018

Agenda

- Basic info about backup tools
- Features of advanced backup tools
- Overview of backup tools:
 - Barman
 - pgBackRest
 - pg_probackup
 - WAL-G



About me

- PostgreSQL developer since 2014
- pg_probackup co-maintainer
- Development manager
at Postgres Professional

Why do you need a backup?

- Restore the database after an accident
- Set up a new replica
- Create a test environment
- Inspect data from the past



What is not a database backup tool?

- Storage snapshot
- Replica
- Set of custom scripts



pg_dump & pg_restore

- dump - “logical backup”
- only provides a “snapshot” of a database state
- recovery takes a long time
 - data loading
 - index creation
- no statistics



Why pg_basebackup is not enough?

1. Take a backup
2. ???
3. Restore the backup
4. PROFIT!



What is a good database backup tool?

- **Usable**
 - documentation & support
 - out-of-box automatization of various routines
- **Scalable**
 - parallel execution
 - compression
 - incremental & differential backups
- **Reliable**
 - archive & streaming backups
 - backup validation

What backup tools exist?

- Barman
- pgBackRest
- pg_probackup
- WAL-G

- BART
 - part of the “EDB Advanced Server”
 - requires pg_basebackup
- WAL-E

Who is who?

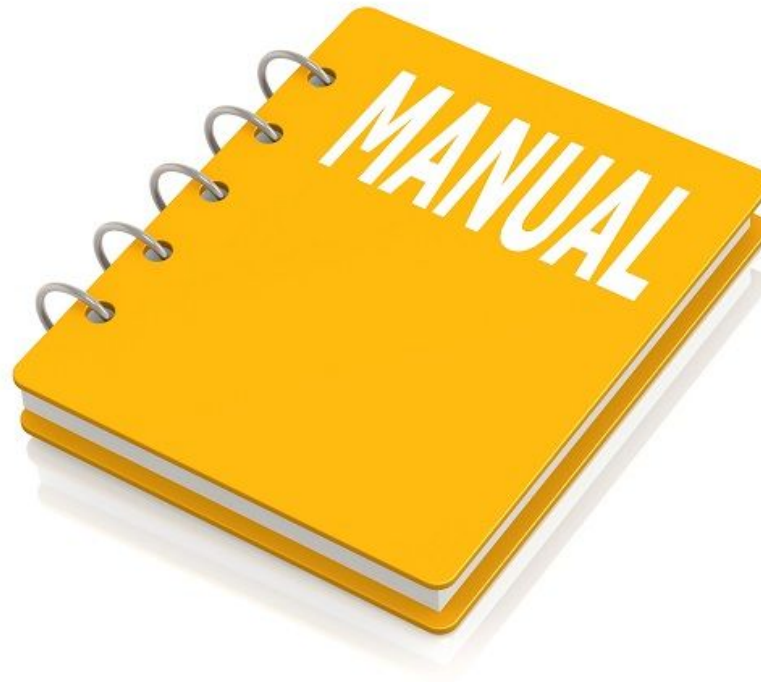
- Barman
 - 2ndQuadrant, GPL v 3.0, python
 - first release: 2011
 - basebackup & rsync modes
- pgBackRest
 - Crunchy Data, MIT License, perl & C
 - first release: 2014



Who is who? (2)

- [pg_probackup](#)
 - Postgres Professional, PostgreSQL License, C
 - first release: 2017 (based on pg_arman)
- [WAL-G](#)
 - introduced by Citus Data, Apache License, Version 2.0, Go now maintained by Andrey Borodin (Yandex)
 - first release: 2017 ("based on" WAL-E)

Documentation & Support





Backup many PostgreSQL servers

- Barman
 - SERVER_NAME
- pgBackRest
 - --stanza
- pg_probackup
 - --instance
- WAL-G
 - DIY

Logging

- **Barman**
 - global logfile
 - DEBUG, INFO, WARNING, ERROR, CRITICAL
- **pgBackRest**
 - --log-level-console, --log-level-file, --log-level-stderr, --log-path
 - OFF, ERROR, WARN, INFO, DETAIL, DEBUG, TRACE
- **pg_probackup**
 - --log-level-console, --log-level-file, --log-filename, --error-log-filename
 - --log-directory, --log-rotation-size
 - VERBOSE, LOG, INFO, NOTICE, WARNING, ERROR, OFF
- **WAL-G**
 - No

Archive management (backup)

- Barman
 - `archive_command = 'rsync ...`
- pgBackRest
 - `archive_command = 'pgBackRest archive-push ...`
`archive-async`
- pg_probackup
 - `archive_command = 'pg_probackup archive-push ...`
- WAL-G
 - `archive_command = 'wal-g wal-push ...`
 - `wal prefetch`

Archive management (restore)

- Barman
 - restore_command = 'barman get-wal
- pgBackRest
 - restore_command = 'pgBackRest archive-get ...
 - archive-async
- pg_probackup
 - restore_command = 'pg_probackup archive-get ...
- WAL-G
 - restore_command = 'wal-g wal-fetch ...
 - wal prefetch

Retention policies

- Barman

- retention_policy = {REDUNDANCY value
RECOVERY WINDOW OF value {DAYS | WEEKS | MONTHS}}

- pgBackRest

- Full & Differential Backup Retention - number of backups to retain
 - Archive Retention

Retention policies

- `pg_probackup`
 - `--retention-redundancy`
 - `--retention-window`
 - `delete --expired --wal`
- **WAL-G**
 - `retain N`
 - `delete before`

Remote backup

- Barman
 - SSH
- pgBackRest
 - SSH
- pg_probackup
 - NFS only
- WAL-G
 - Yes

Backup to a cloud

- Barman
 - DIY
- pgBackRest
 - S3
- pg_probackup
 - DIY
- WAL-G
 - S3

Parallel backup & restore

- Barman
 - parallel_jobs = n (rsync-mode only)
- pgBackRest
 - --process-max
- pg_probackup
 - -j num_threads
- WAL-G
 - WALG_UPLOAD_CONCURRENCY
 - WALG_DOWNLOAD_CONCURRENCY

Compression

- Barman
 - `compression = gzip` (basebackup-mode only)
 - `network_compression` (rsync-mode only)
- pgBackRest
 - `--compress` (gzip)
 - `--compress-level`
 - `--compress-level-network`



Compression (2)

- **pg_probackup**
 - --compress-algorithm (zlib, pglz)
 - --compress-level
- **WAL-G**
 - WALG_COMPRESSION_METHOD (lz4, lzma, zstd)

Incremental backups

- Barman
 - file-level incremental (rsync-mode only)
- pgBackRest
 - file-level incremental (compare file timestamps)
 - file-level differential

1 Gb granularity

Incremental backups (2)

- **pg_probackup**
 - page-level incremental
 - PTRACK (requires patch)
 - PAGE (requires WAL archive)
 - DELTA (compare page LSNs)
- **WAL-G**
 - page-level incremental DELTA backup

8Kb granularity

Backup validation

- Barman
 - DIY with custom hooks
- pgBackRest
 - file-level checksums
 - page checksums on backup
- pg_probackup
 - file-level checksums
 - page-level checksums
 - validate command
- WAL-G
 - No

Streaming backups (zero data loss)

- Barman
 - streaming_archiver
 - slot_name
- pgBackRest
 - No
- pg_probackup
 - --stream
 - --slot
- WAL-G
 - No

Conclusion

	Barman basebackup	Barman rsync	pgBackRest	pg_probackup	WAI-G
Many instances	+	+	+	+	-
Logging	+	+	+	+	-
Archive management	+	+	+	+	+
Retention policies	+	+	+	+	+
Remote backup	+	+	+	-	+
Backup to a cloud	-	-	+	-	+

Conclusion

	Barman basebackup	Barman rsync	pgBackRest	pg_probackup	WAI-G
Compression	+	-	+	+	+
Parallel backup	-	+	+	+	+
Parallel restore	-	+	+	+	+
Incremental (file-level)	+	+	+	+	+
Incremental (page-level)	-	-	-	+	+

Conclusion

	Barman basebackup	Barman rsync	pgBackRest	pg_probackup	WAI-G
Verification of data	-	-	+	+	-
Streaming backup	-	+	-	+	-

and the
winner is...