



Barman, past, present, and future of a pioneer community project

Giulio Calacoci,
Principal Engineer

Martín Marqués,
Director Engineering II

Agenda

- Motivation and first steps
- Evolution and main pivot points
- Interaction with the community
- Barman's kitchen: how releases happen
- Plans for the future

A man with short brown hair and glasses is looking out a window. The window has vertical bars or frames. The view outside is a hazy, overcast landscape with some greenery and buildings in the distance. The lighting is soft and natural, coming from the window.

Why Barman?

Why Barman: Inspiration

- Customers migrating from Oracle were seeking for a tool comparable to RMAN
- PostgreSQL users were using scripts to execute physical backups
 - No standards
 - No recovery procedures
- Over the years lots of projects inspired new Barman features
- We like to think that Barman inspired other projects

Barman Archaeology

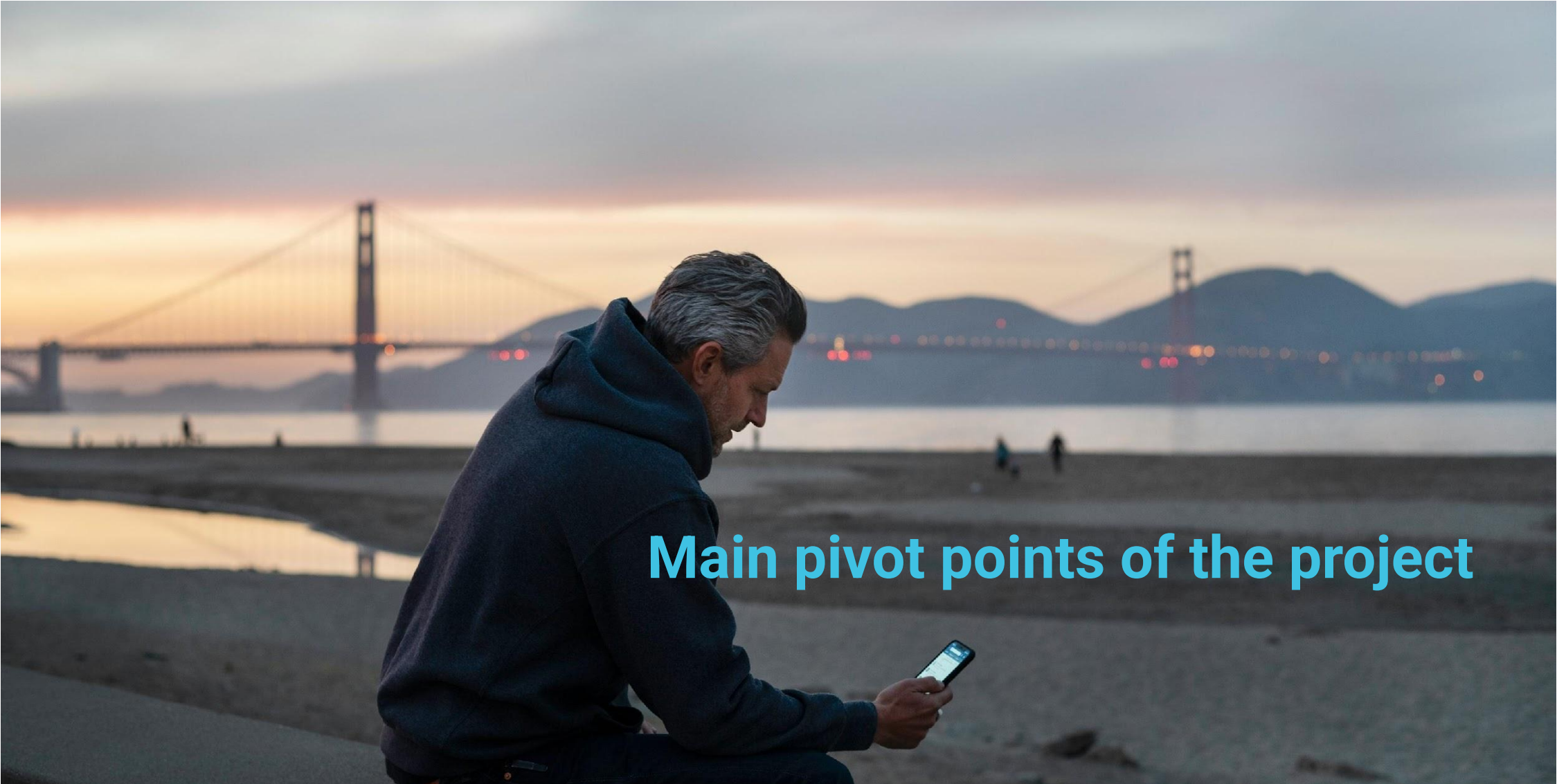


Barman archaeology: Version 1.0.0

- Released july 2012
- Barman was among the first open source solutions for Postgres physical backups
- Barman features:
 - Servers/backups catalog (supporting different versions, 8.4+)
 - Backup management (backup, delete, restore)
 - WAL archiving
 - WAL compression (gzip, bzip2)
 - Backup execution based on rsync

Barman archaeology: sponsors impact

- Barman would not have been born without sponsorship, besides 2ndQuadrant
- Founding sponsors:
 - CSI Piemonte
 - GestionaleAuto.com
 - Navionics
 - 4CaaSt project



Main pivot points of the project

2012-2016

- The development team expanded throughout the years
 - Development,
 - QA, testing,
 - Documentation
- Constant stream of releases and features
- Constant interactions with the community

2017-2018

- Team members moved to other projects
- Remaining members stretched on multiple projects
- Reduced interaction with the community

2019

- The cloud had taken its place
- Barman needed to evolve and try to bridge the gap with the cloud
- barman-cloud-wal-archive and barman-cloud-backup were created
 - Released in december 2019
 - The scripts allowed postgres servers to directly archive backups and wal files into S3
 - No real support for catalogue
 - Detached from the “classic” barman “server”

2020-2023

- EDB acquires 2ndQuadrant
 - Leadership of the project moved to Abhijit Menon-Sen
 - Rebuild of the team
- Add new barman-cloud features:
 - `barman-cloud-restore`, `barman-cloud-wal-restore`
 - `barman-cloud-backup-list`
 - `barman-cloud-backup-delete`
- Extend cloud support for Azure and GCP
 - Cloud snapshots
- Backup compression for backups (lz4 and zstd)
- Barman keep
- Barman models

2024-????

- Building a new team
- Integrate Postgres block-level incremental backup in Barman
- Refactor WAL compression for Barman and barman-cloud
 - Add zstd and lz4 for WAL compression
- Immutable storage support and AWS snapshot lock
- Initial support for encryption of Backups

Community interaction



Community interaction

- GitHub project
 - <https://github.com/EnterpriseDB/barman/>
 - Issues
 - PRs
- User group mailing list
 - Google Group
 - <https://groups.google.com/g/pgbarman>
 - pgbarman@googlegroups.com
- Documentation
 - Renewed look & feel in 2024
 - Coming soon: Contributor docs

Community contributions

Some examples

- AWS snapshot lock protection (Rui Marinho)
- Improve Barman to fully use local loggers (Eric Thiebaut-George)
- Removal of head-bucket bucket checks from barman-cloud (Brought up from CNPG users)
- Add `--if-not-exists` flag to `receive-wal` (@crazybolillo)
- Fix `barman backup --wait` on Postgres 17
- Check for `USAGE` privilege in `pg_has_role` (Daniele Giudice)
- Run `barman switch-wal --force` when `pg_checkpoint` role is granted (Thomas Ziegler)
- Permit `archive_mode=always` for Postgres 9.5 servers (Christoph Moench-Tegeder)
- Add support for the `pigz` compressor (Stefano Zacchioli)

A person wearing a red and blue plaid shirt is gesturing with their hands while sitting at a desk. In the background, a laptop is open, displaying a web application with various charts and data. A smartphone is resting on a green notebook on the desk.

Changes in Pull Request contribution workflow

Contribution Pull Request workflow

1. Submission of a Pull Request
2. Review from a Barman developer
3. Follow up from author on needs from review
4. Barman developers give a “soft” approval
5. Pull Request is moved to our internal CI/CD for thorough testing, addition of release notes, etc.
6. On the next release the commit(s) from the author will be pushed

Our Release process in a nutshell

1. Work is done internally
 - a. We run integration tests through private repositories
 - b. There's an internal branch which mirrors the public branch
 - c. Community contributions are pulled to perform thorough testing
2. Release is scheduled (internally at EDB)
3. Security and compliance tests are run
 - a. Blackduck, Sonarqube, etc.
4. All changes since the previous release are pushed to the public branch
5. Artifacts are push to GitHub and PyPI, release created in GitHub
6. Notification sent to PGDG packagers (rpm and deb) about the new release

Plans for the future



The future of Barman & backups in Postgres

- Import/Export of backups
- Extend ransomware coverage
 - S3 bucket lock protection
 - Azure snapshot lock protection
 - GCP snapshot lock protection
- More encryption
 - Encryption of rsync backups
 - Encryption in pg_basebackup

The future of Barman & backups in Postgres

- Evergreen
- Collaboration in Postgres
 - Barman can take block level incremental backup without the need of pg_basebackup
 - Improvements in pg_combinebackup to reduce cost and time
 - Have list of files that pg_combinebackup will use
- Better integration with cloud providers
 - Cloud support in Barman (non-cloud)
 - Separation of compute and storage

The future of Barman & backups in Postgres

- Add a Barman service
 - Remove the need to run barman cron regularly
 - Adds the possibility for Barman to perform tasks at more appropriate intervals
 - Don't apply retentions every minute
- Integrate a API interface
 - Good for external observability and monitoring
 - Gives flexibility when integrating with HA tools like Patroni or repmgr
- Grandfather - Father - Son support
 - Differentiation of retention policies by tiers
 - Reduce costs for compliance backups (older backups stored in cheaper storage)

Feedback





Question?