

```
postgres@  
psql (9.3  
Type "he
```

```
postgres=  
UPDATE 48  
postgres=
```

```
^C
```

```
^C
```

```
^Z
```



```
'black' ;
```

ere!

```
postgres@pg $> psql
```

```
psql (9.3.1)
```

```
Type "help" for help.
```

```
postgres=# DROP DATABASE scott;
```

```
DROP DATABASE
```

```
postgres=# \q
```

```
postgres@pg $> date
```

```
Fri 8 Nov 2013 18:43:15 CET
```

04:00:01

05:12:44

t_{start}

t_{end}



14 hours 43 minutes 14 seconds
of potential data loss (RPO)

18:43:15

transactions

CRASH!!!



04:00:01

04:41:52

t_{start}

t_{end}



1 minute 1 second
of potential data loss (RPO)

18:43:15

transactions

CRASH!!!



18:42:14

2ndQuadrant



Professional PostgreSQL

www.2ndQuadrant.com



Barmman

Backup and recovery
manager for PostgreSQL

GABRIELE BARTOLINI

PGConf.DE 2013 - Oberhausen, 8 November 2013

GABRIELE BARTOLINI

- Co-Founder and Manager of 2ndQuadrant Italia
 - Data Architect, Business critical environments
 - Data warehousing
- Co-Founder Italian PostgreSQL Users Group
- Co-Founder PostgreSQL Europe
- PostgreSQL Contributor and Advocate

DISCLAIMER

This talk assumes you are familiar with disaster recovery concepts and PostgreSQL implementation of Point In Time Recovery

BE AWARE

In 2ndQuadrant, all these concepts usually fit in a 2 day workshop on Disaster Recovery and a 1 day workshop on Barman alone

OUTLINE

- Business continuity / Disaster recovery for databases
- Disaster recovery with Barman for PostgreSQL

PART I

Business continuity / Disaster recovery for databases



BUSINESS CONTINUITY

*activity performed by an organisation to ensure that **critical business** functions will be **available** to customers, suppliers, regulators, and other entities that must have access to those functions - Wikipedia*

INFORMATION TECHNOLOGY

- Business continuity
 - High availability
 - **Disaster recovery**

LAW REQUIREMENTS

In Italy, the “Codice dell’Amministrazione Digitale” defines **business continuity requirements** for public administrations

DISASTER

(touch wood now)

system/hardware failures
unintentional errors
natural disaster

REACT TO A DISASTER

Recover systems, data and infrastructures

TOO LATE!

Do not wait for a disaster to happen

PLAN FOR DISASTERS

“Disasters” will happen. Be prepared.

*“Plans are worthless, but **planning is everything**. There is a very great distinction because when you are planning for an emergency you must start with this one thing: the very definition of “**emergency**” is that it is **unexpected**, therefore it is not going to happen the way you are planning.”*

- Dwight D. Eisenhower

REGULAR CRASH TESTS

DATABASE DISASTER RECOVERY

Let's just focus on **databases!**

REQUIREMENTS

- Automated backups
- Notifications (anomalies)
- Frequency of backups
- Retention policies
- Data protection
- Availability for recovery

TYPES OF BACKUP

- Full backup
- Incremental backup
- Differential backup
- Hot backup
- Logical backup
- Physical backup

POSTGRES BACKUP

- Hot backup
 - MVCC (core)
- Logical Backup
 - pg_dump
- Physical Backup
 - Full backup (base backup)
 - Differential backup (WAL)
 - Incremental backup
 - N/A

TRADITIONAL DR WITH POSTGRESQL

- PostgreSQL primitives for DR are robust and reliable
- High level skills
 - DBA
 - Sysadmins
- Custom scripts
- Hard to integrate in:
 - Backup solutions
 - Disaster Recovery plans
- **Hard to test!**

EXISTING TOOLS

- Omni-PITR
 - WAL centric
- WALmgr
 - WAL centric
- pg-rman
 - Server centric
- WAL-E
 - EC2 centric, but ...
 - good
 - came later

NONE FOR DR

None of them was a pure **disaster recovery** solution.
We wanted something similar to Oracle's RMAN.

FILLING A HOLE

The lack of a DR solution is a **barrier** towards the **adoption** of PostgreSQL from Oracle users' point of view.

OUR GOALS

- Hot, Full, Differential and Incremental backups
- Multiple servers
- Remote backup & recovery
- Backup catalogues
- Retention policies
- Archival and compression
 - WAL segments
 - Periodical backups
- **Automation**
- **Integration**
- **Usability**

WWW.PGBARMAN.ORG



Barmman

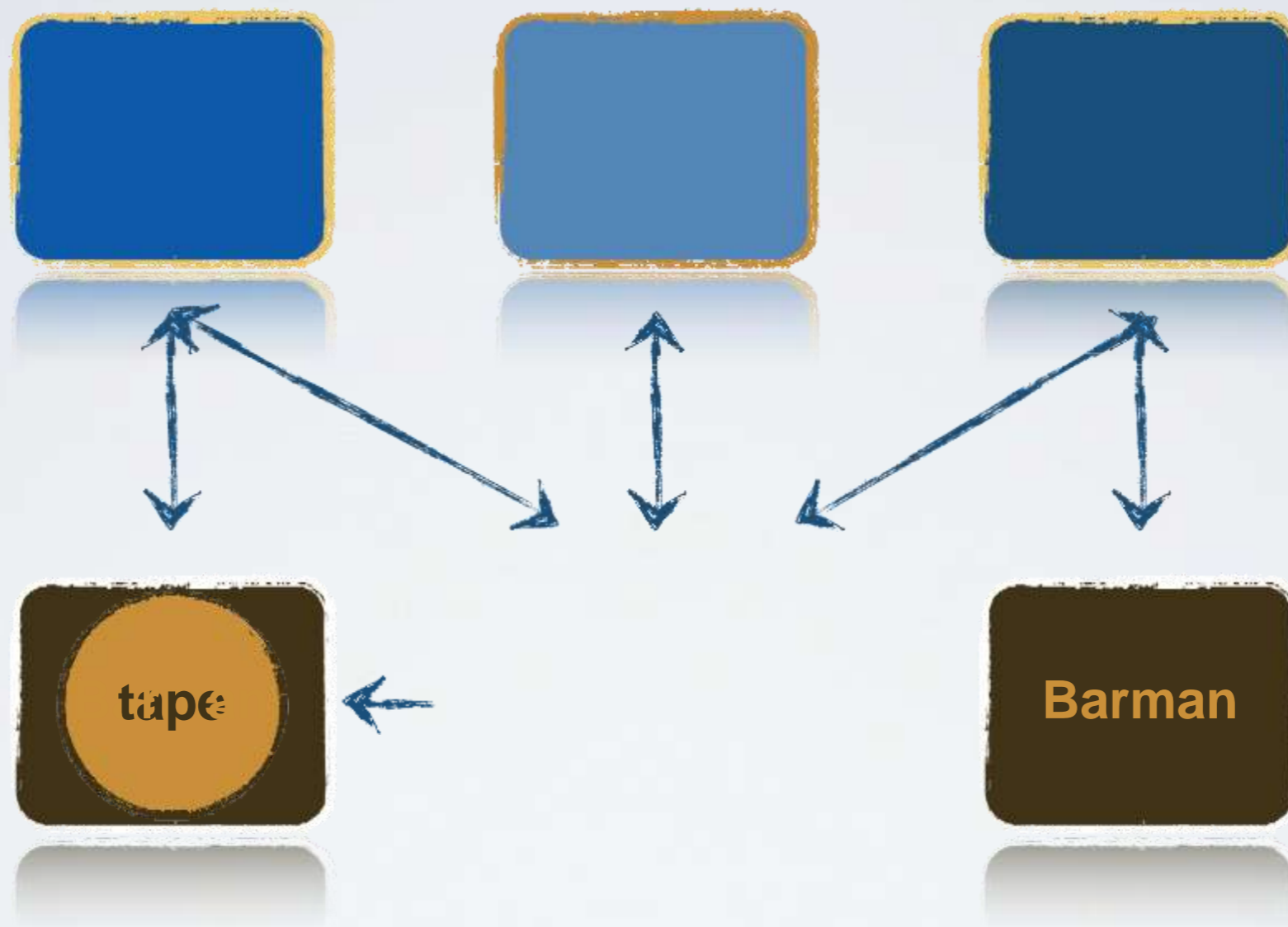
Backup and recovery
manager for PostgreSQL

PART II

Disaster recovery with Barman for PostgreSQL

BARMAN

- GNU GPL 3
- Hosted on Sourceforge.net
- Linux
- Python 2.6/2.7 (3.0 exp.)
- PostgreSQL 8.4 to 9.3
- PyPI package
- RPM package
- Debian/Ubuntu package
- Designed, developed, maintained by 2ndQuadrant



LAN, centralized architecture

data centre 1

data centre 2

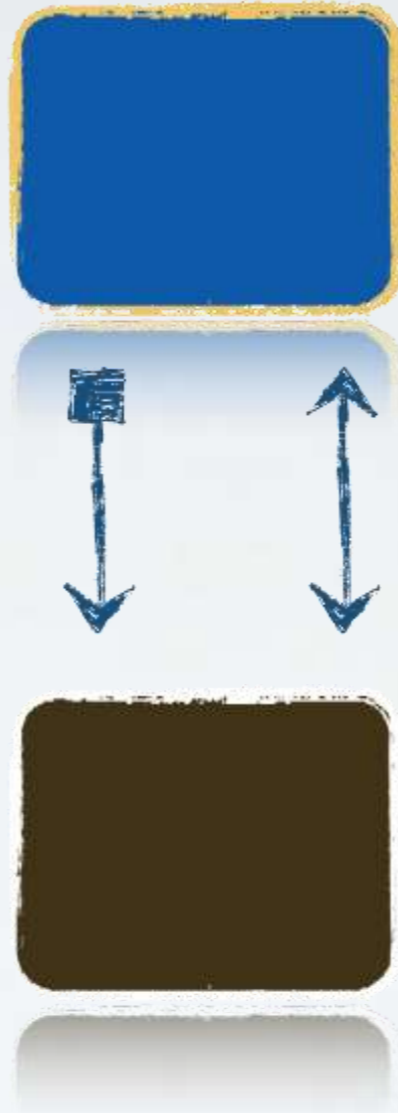


r sync



Geographic redundancy

Continuous archiving
(*WAL shipping via SSH*)



SSH commands
SQL commands



secure channel **cron**

*Barman's
WAL archive*

Postgres

Periodical backup (weekly)
Differential backup
Backup catalogue

Barman

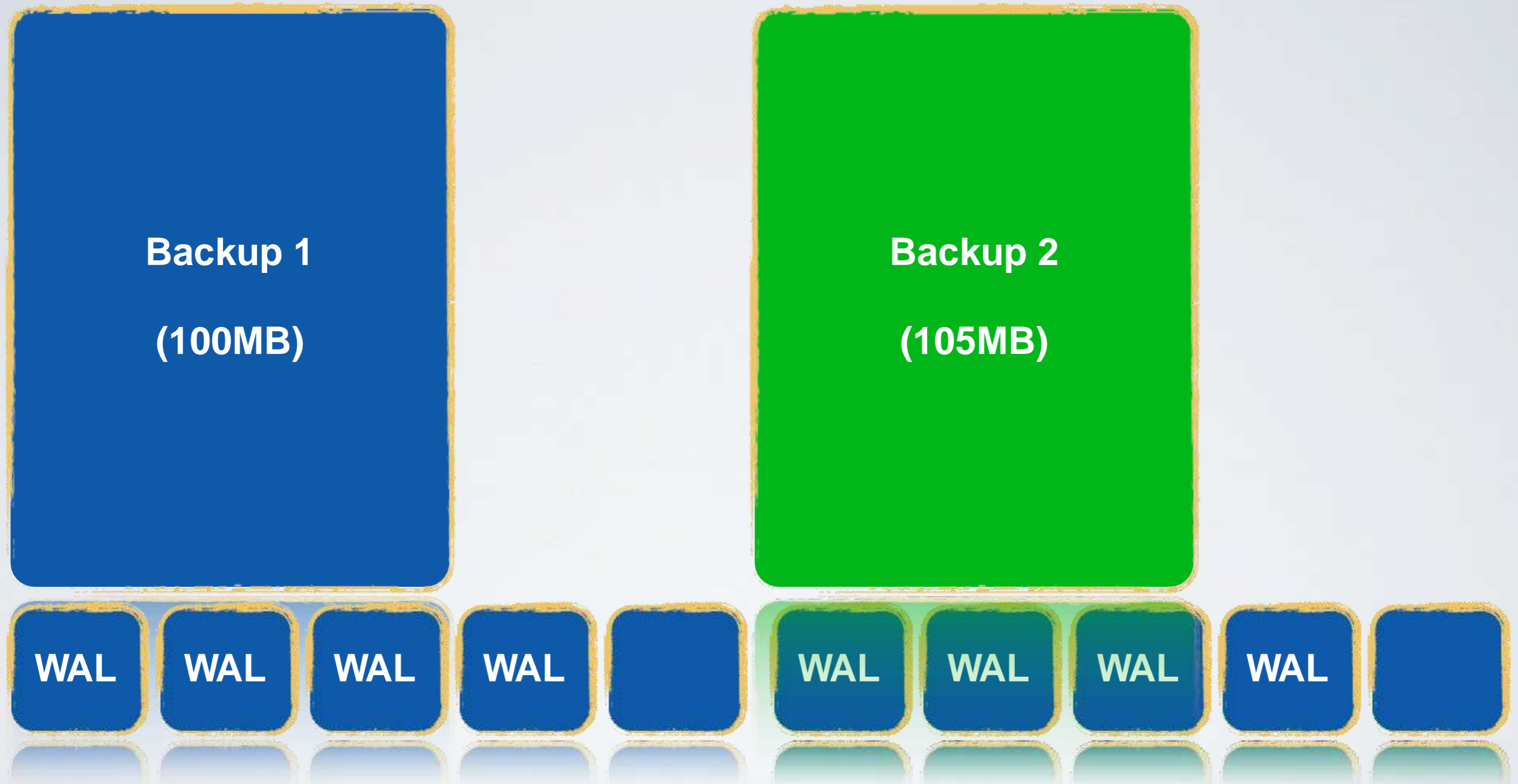
Full backup - Sat 1, 4AM

Full backup - Sat 8, 4AM

Full backup - Sat 15, 4AM

Full backup - Sat 22, 4AM

WAL archive



Size: 100MB + 86MB = 186MB

Size: 105MB + 80MB = 185MB

CONFIGURATION FILE

[barman]

```
barman_home = /srv/barman
barman_user = barman
log_file = /var/log/barman/barman.log
log_level = NOTICE
compression = gzip
```

[production]

```
description = Production PostgreSQL
ssh_command = ssh postgres@pg.2ndQuadrant.it
conninfo = host=pg.2ndQuadrant.it user=postgres
compression = bzip2
```

MULTI-SERVER CONFIGURATION

```
[barman]
; General configuration
; ...

[server_one]
; Configuration for Server 1
; ...

[server_two]
; Configuration for Server 2
; ...

[server_x]
; ...
```


MULTIPLE FILES INCLUSION

```
[barman]
; General configuration
; ...
configuration_files_directory = /etc/barman.d
```

CONVENTION OVER CONFIGURATION

global/per server options
default directory layout

CONVENTIONAL DIRECTORIES FOR BARMAN

- `barman_home (/srv/barman)`
 - *server directory* (`/srv/barman/production`)
 - *base directory* (`/srv/barman/production/base`)
 - *WAL directory* (`/srv/barman/production/wals`)
 - *incoming directory* (`/srv/barman/production/incoming`)

GLOBAL COMMANDS

- **List of managed servers**
 - `barman list-server`
- **Maintenance operations**
 - `barman cron`

SERVER COMMANDS

- Information and diagnostics
 - `barman status`
 - `barman check`
 - `barman show-server`
 - `barman list-backup`
- Backup control
- Recovery control

```
barma1 backupserverbarman list-backup test-9.1
test-9.1 20120821T164531 - Tue Aug 21 16:45:38 2012 - Size 1090 MiB - WAL Size 3040 MiB
test-9.1 20120821T163806 - Tue Aug 21 16:40:37 2012 - Size 68.0 MiB - WAL Size 208.0 MiB
test-9.1 20120821T163146 - Tue Aug 21 16:32:10 2012 - Size 45.0 MiB - WAL Size 432.0
MiB test-9.1 20120821T162630 - Tue Aug 21 16:26:32 2012 - Size 53.0 MiB - WAL Size:
400.0 MiB test-9.1 20120821T161651 - Tue Aug 21 16:16:53 2012 - Size 49.0 MiB
- WAL Size: 48.0 MiB
barma backupserver 1
```

BACKUP CONTROL

- `barman backup`
- `barman show-backup`
- `barman list-files`
 - standalone, data, wal, full
- `barman delete`

```
barman backupserver barman backup test-9.1
Starting backup for server test-9 1 in /tmp/barman-tests/test-9 1/base/20120821T163235
Backup start at xlog location: 0/29000020 (00000001000000000000000029, 00000020)
Copying files.
Copy done
Asking PostgreSQL server to finalize the backup.
Backup end at xlog location: 0/290000A0 (00000001000000000000000029, 000000A0)
Backup completed

barman backupserver | 1
```


SHOW BACKUP

- **General**

- Server name, Postgres version, status, ...

- **Base backup**

- Start/End time, first/last WAL, disk usage, ...

- **WAL**

- Number of associated files
- disk usage

- **Context**

- Previous/Next backup

```
$ barman show-backup angus latest
```

```
Backup 20131102T130001:
```

```
Server Name      : angus
Status           : DONE
PostgreSQL Version: 80417
PGDATA directory : /home/raid/pgdata
Tablespaces:
  tb_angus: /home/raid/postgres (oid: 16385)
  tb_temp: /home/tables (oid: 19812)
```

```
Base backup information:
```

```
Disk usage      : 10.0 TiB
Timeline        : 1
Begin WAL       : 0000000100006C570000000B5
End WAL         : 0000000100006C7E0000000D4
WAL number      : 9977
Begin time      : 2013-11-02 13:00:01.617907
End time        : 2013-11-04 14:07:20.981717
Begin Offset    : 32
End Offset      : 7185968
Begin XLOG      : 6C57/B5000020
End XLOG        : 6C7E/D46DA630
```

```
WAL information:
```

```
No of files     : 56655
Disk usage      : 232.0 GiB
Last available  : 0000000100006D5D000000002
```

RECOVERY CONTROL

- **Recovery target** (full / point in time)
- **Local recovery**
 - `barman recover`
- **Remote recovery**
 - `barman recover --remote-ssh-command`

ADVANCED RECOVERY

- Point In Time Recovery

- `--target-time = TIME`
- `--target-xid = XID`
- `--target-name = NAME`
(for 9.1+)

- Relocation of tablespaces

- `--tablespace
NAME:LOCATION [...]`

COMMON USE CASES

- Unintentional errors recovery
- Disaster recovery
- Sandbox server (BI, staging, ...)

```
barman recover test-9.1 /srv/postgres/recover/
Starting local restore for server test-9.1 using backup 20120821T161651
Destination directory: /srv/postgres/recover/
Copying the base backup
Copying required wal segments.
The archive command was set to 'false' to prevent data losses. Restore done!

Please review the network and archiving related settings in the postgres configuration file before start the just recovered instance.

barman backupserver 1
```

RETENTION POLICIES

- **User-defined policy**
- How long backups are retained for recovery
- Point of recoverability
 - REDUNDANCY
 - RECOVERY WINDOW

RETENTION POLICY CONFIGURATION

```
; Base backup retention policy  
retention_policy = 'redundancy 3'  
retention_policy = 'recovery window of 3 months'
```


BANDWIDTH CONTROL

- You can limit I/O bandwidth usage
 - `bandwidth_limit` global/server option
 - `tablespace_bandwidth_limit` on a per tablespace basis
- Unit of measure: kilobytes (default 0, no limits)

BANDWIDTH LIMIT CONFIGURATION

```
[barman]           = 4096  
bandwidth_limit
```

```
[server_one]       = 1024  
; ...  
bandwidth_limit
```

```
[server_two]  
; ...  
tablespace_bandwidth_limit = tbs01:4096,tbs02:2048
```

BACKLOG

- Incremental backup
- Better recovery support
- Sandbox recovery
- Logical backup integration
 - `pg_dump` on sandbox instances
- Backup from standby
- More hook scripts
- TAR format for backups
- Export/Import of backups
- External backups
- SSH only connections
- Replication protocol support
 - libpq only connections
 - `pg_basebackup`
 - WAL streaming (0 Data Loss)
 - Backup server
- Windows support
- JSON output for full automation
- Backup validation
- ...

OUR COMMITMENT

- Keep it open source
- Reinvest money from sale of DR turnkey solutions in R&D
- Support and maintain RPM/Debian packages
- Accept sponsorships for new features development

CSI PIEMONTE

(One of the top 10 ICT companies in Italy for revenue)

*“We found in Barman the optimal solution for physical backup and disaster recovery of PostgreSQL databases. Barman is **robust and easy to use**. Its command interface allows an easy **integration** with the existing management tools in our environment.”*

Sponsors of RPM package and WAL compression

CONCLUSIONS

- Hides complexity of PITR / Keeps unaltered PITR strenghts
- Not invasive
- Fosters migrations from Oracle
- “*Standard de facto*” for PostgreSQL Disaster Recovery
- Advice: plan for DR (if you have not done it yet)

TIMETO ...

```
yum install barman  
apt-get install barman
```



Barman
Backup and recovery
manager for PostgreSQL

QUESTIONS?

Gabriele.Bartolini@2ndQuadrant.it

Twitter: [_GBartolini_](#)

www.pgbarman.org

www.2ndQuadrant.com

THANK YOU!

LICENSE



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Copyright (c) 2012, 2013 - 2ndQuadrant.it