

pgBackRest

An ultimate solution for PostgreSQL backups and restores
by [Crunchy Data](#).



Agenda

- Do we need backups?
- Backup types
- Tools to use and the others
- Data files, checkpoints, wal files
-

Who's me?

InterBase / Firebird app developer, DBA (3 years)

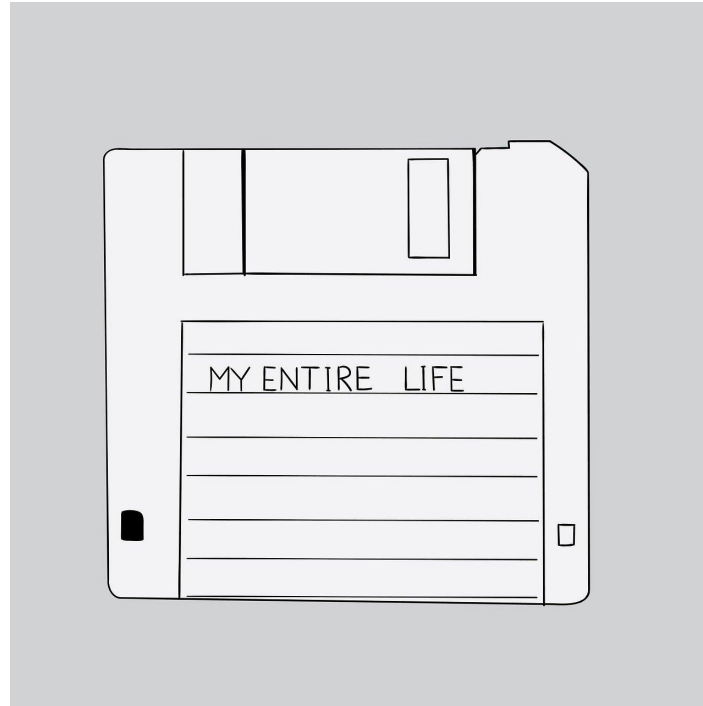
Oracle DBA (17 years)

PostgreSQL DBA (since 2010)

Elephants enthusiast ...



Do we need backups?



Do we need backups?

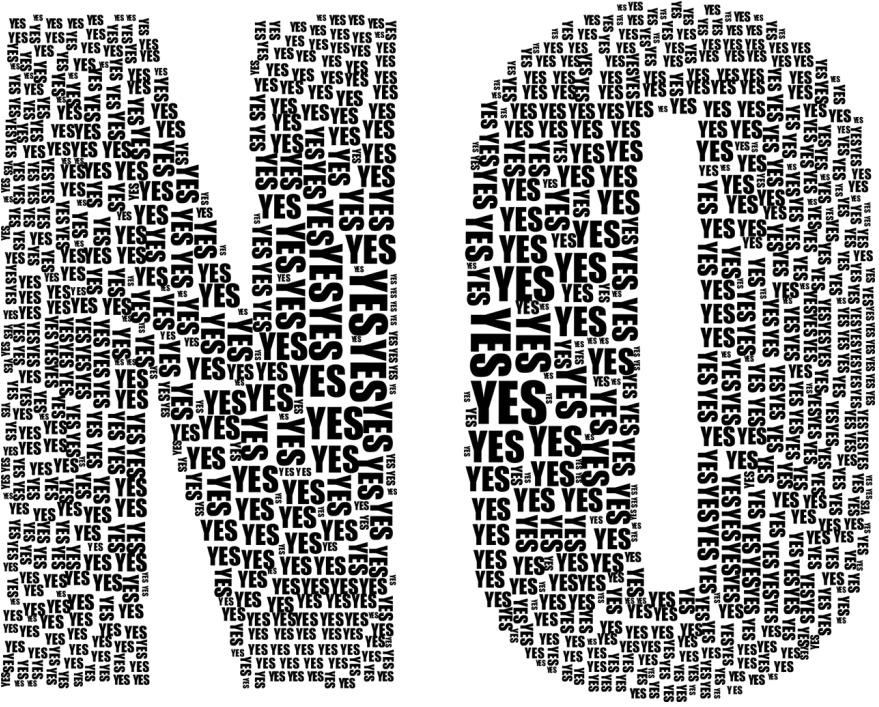
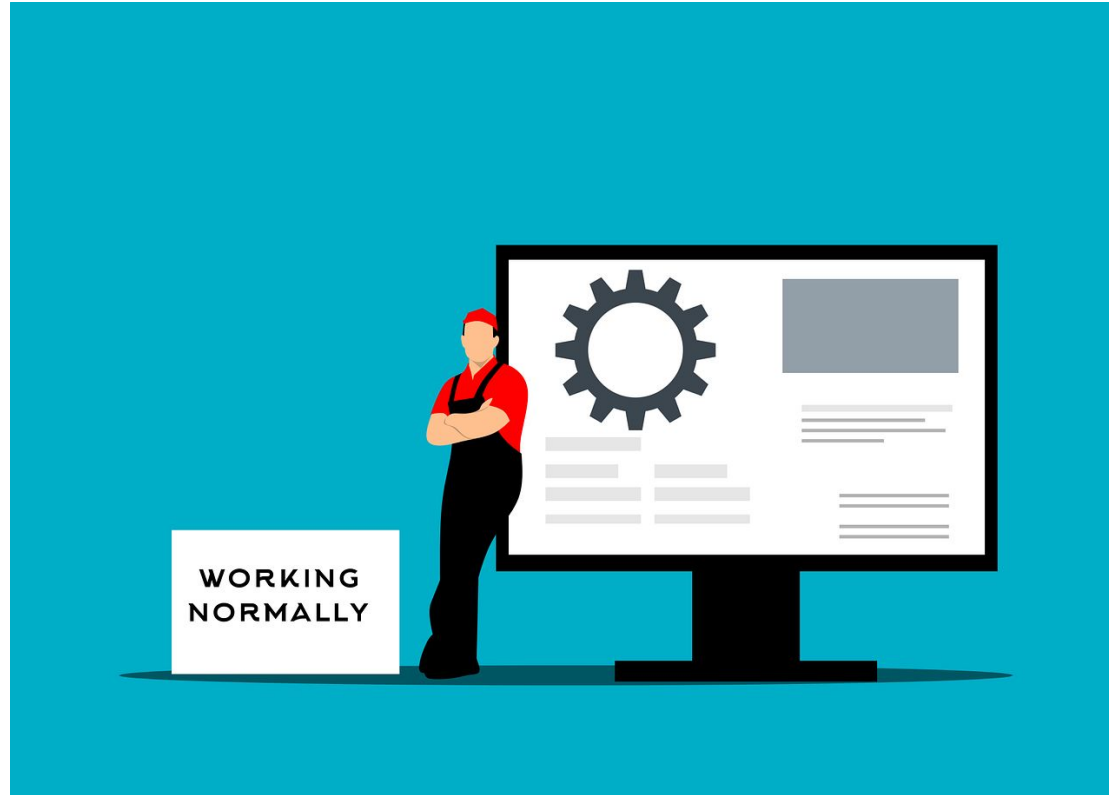


Image by [Gordon Johnson](#) from [Pixabay](#)

Unless you want to use it for restore...



Key backup attribute?

You have to have one before you can use it.

A disaster will happen, the only unknown is **WHEN** it happen.

Image by [Zoltan](#) from [Pixabay](#)



Do we need backups?



Image by [Gordon Johnson](#) from [Pixabay](#)

Backup types

Logical

Snapshot of data in a (potentially) platform independent format, like CSV or TAB separated files.

`man pg_dump`

- plain (SQL)
- custom (compressed)
- tar
- directory

Backup types

Physical

Physical copy of PostgreSQL instance files in a form that can be used for restore.

pgBackRest is one of tools for physical backups.

(physical) Backup tools

- pg_basebackup - part of PostgreSQL core code
 - receives new features, for Pg 17 incremental backups are intended
- Barman
- [Wal-G](#)
- pgBackRest
- ... [Wiki: Ecosystem:Backup](#)

Contents [\[hide\]](#)

- 1 [Amanda](#)
- 2 [Bacula](#)
- 3 [Barman - Backup and Recovery Manager for PostgreSQL](#)
- 4 [Handy Backup](#)
- 5 [Iperius Backup](#)
- 6 [NetVault Backup](#)
- 7 [pg_probackup](#)
- 8 [pgBackRest](#)
- 9 [pgmoneta](#)
- 10 [Simpana](#)
- 11 [Spectrum Protect](#)
- 12 [Veritas NetBackup for PostgreSQL Agent](#)

Don'ts

- Hack “home-made” backup scripts
- Create backups, without regularly tested restores from those backups
- Use proprietary solutions (even some large backup SW vendors) with caution
 - personal experience with one proprietary SW that was unable to resume for an out-of-disk space event in a dedicated working partition used internally by the SW

Offline backups

When a PostgreSQL instance is stopped

- feel free (take care of symbolic links...) to take an offline copy of whole Postgres cluster
 - PGDATA
 - WAL (if outside of PGDATA)
 - TABLESPACES (if used)
- or if you shut down a virtual machine, create a full copy of a VM.

If you can afford outages and Point-In-Time recovery is not requested.

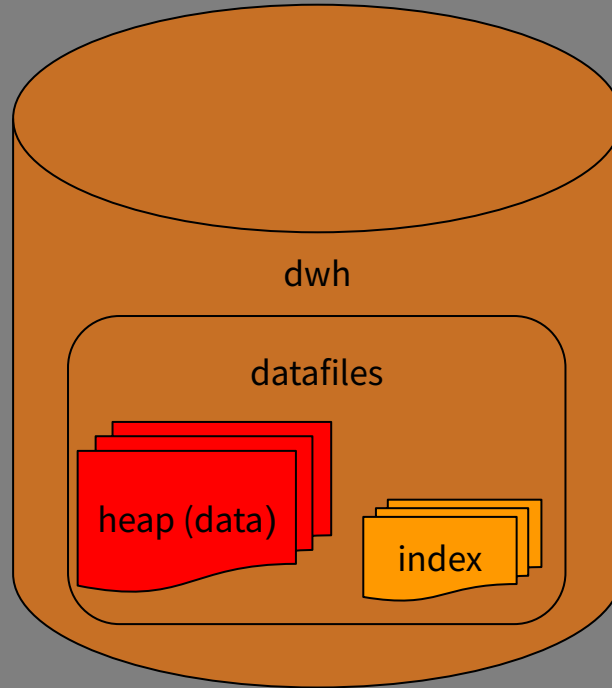
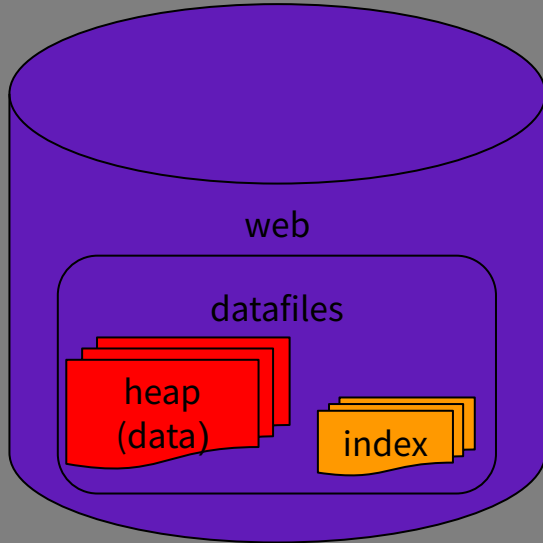
Nowadays - usually not an option :-)

Online physical backups

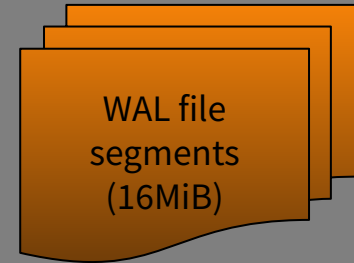
- PG instance is online available to users for Read/Write traffic
 - the data files are being changed during the backup
 - THE COPIED DATA FILES ARE INCONSISTENT
- Write Ahead Log files (WAL - binary transaction log files)
 - allows to recover restored database to a consistent state
 - allows perform point-in-time recovery (PITR)
- `archive_command` or `archive_library` configuration parameters
 - copy WAL file full of data (or after `archive_timeout`) to a SAFE location for recovery purposes
 - SAFE usually mean a remote location
 - could be also local copy but there are some risks...

Simplified database physical structure

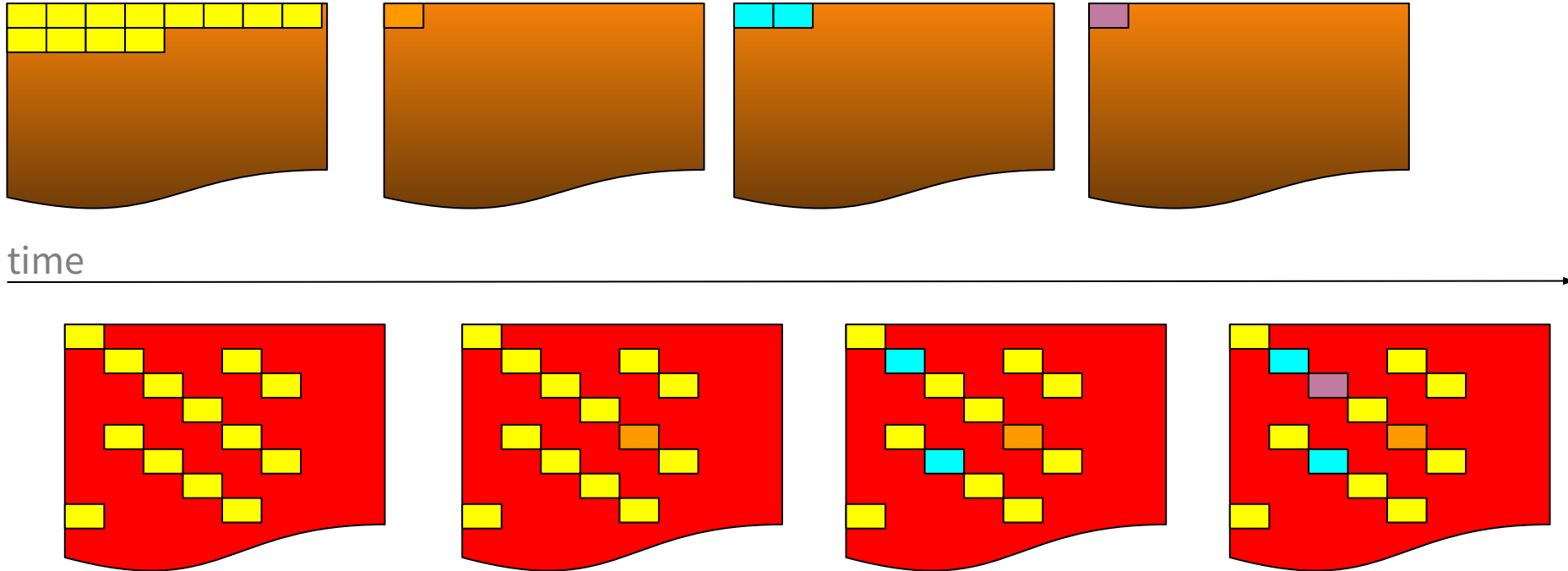
PostgreSQL cluster (instance) files



Write Ahead Log files



Committed changes are in WAL, later in datafiles

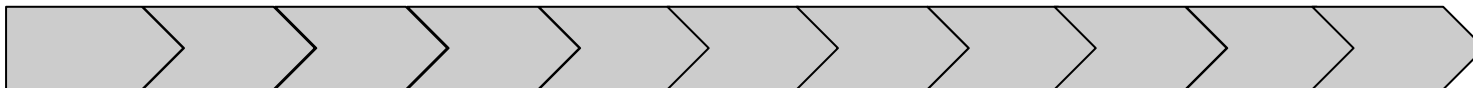


Write Ahead Log, LSN

- Write Ahead Log
 - Logically single journal file
 - holding all databases changes from instance creation
 - of an “infinite” length
 - Physically represented by segments (WAL files), by default 16MB each
- LSN - Log Sequence Number
 - pointer to WAL position
 - `pg_walfile_name()` - get WAL segment name from LSN

```
postgres=# select pg_walfile_name('2/452921A0');
pg_walfile_name
```

```
-----
000000010000000200000045
```

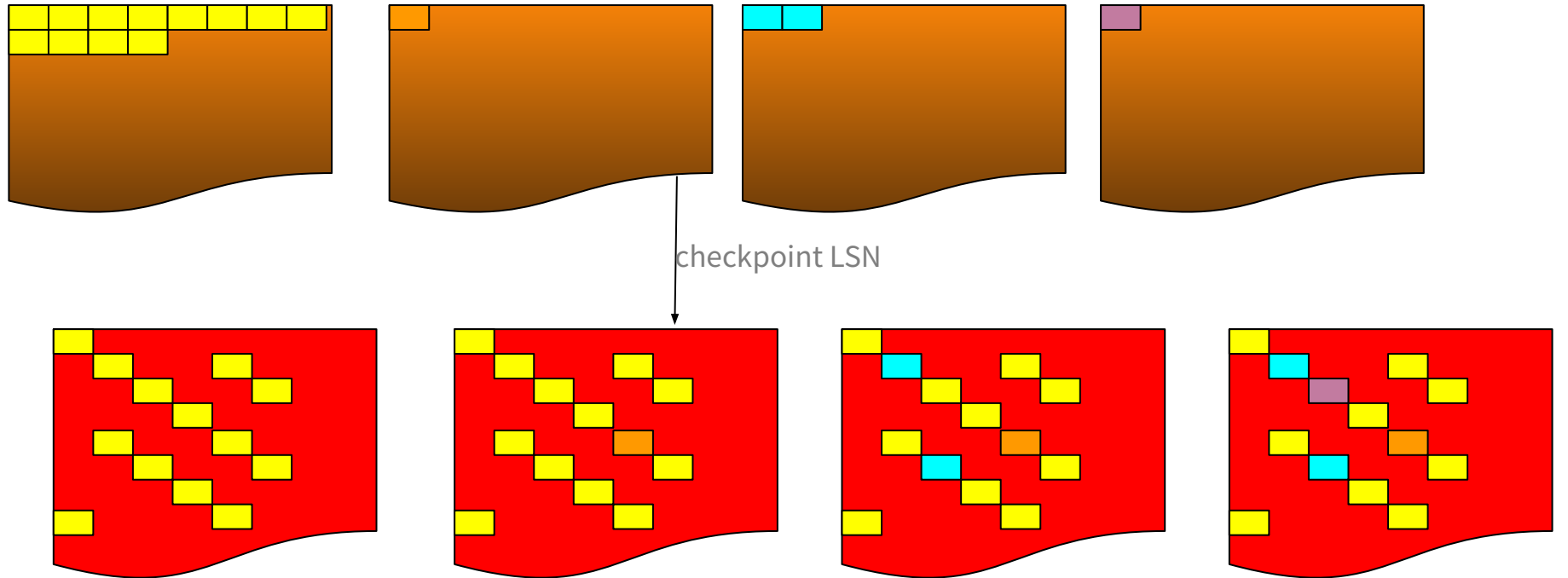


Write Ahead Log and recovery

- Write Ahead Log
 - changes from WAL file can be applied on data files to recover them up to a specific point in time
 - PITR (Recovery Point Objective)
 - recovery from instance creation requires
 - all WAL files from instance creation (space unlimited, reliable storage)
 - a lot of time to perform recovery (months or years of WAL files) (Recovery Time Objective)
 - not a practical approach



Checkpoint



time

Checkpoint

- Ensures that all data up to a certain position in transaction log identified by LSN are synced to data files.
 - WAL files before checkpoint are no longer needed for instance recovery

```
2024-05-14 09:59:49.406 CEST [2994794] LOG:  checkpoint complete: wrote 11 buffers (0.1%);
      0 WAL file(s) added, 0 removed, 20 recycled;
      write=0.002 s, sync=0.003 s, total=0.039 s;
      sync files=5, longest=0.002 s, average=0.001 s;
      distance=326140 kB, estimate=519245 kB;
      lsn=2/452921A0, redo lsn=2/45292168
```

```
pg_waldump /var/lib/postgresql/16/main/pg_wal/000000010000000200000045
```

```
rmgr: Standby ... lsn: 2/45292168, prev 2/452920F0, desc: RUNNING_XACTS ...
rmgr: XLOG ... lsn: 2/452921A0, prev 2/45292168, desc: CHECKPOINT_ONLINE redo 2/45292168; ...
rmgr: Standby ... lsn: 2/45292218, prev 2/452921A0, desc: RUNNING_XACTS ...
```

pg_basebackup

```
postgres@alesnbbs:~$ pg_basebackup -D /tmp/pg16/ -p 5435 -v -l test_backup
pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 2/48000028 on timeline 1
...
pg_basebackup: write-ahead log end point: 2/48000100
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: renaming backup_manifest.tmp to backup_manifest
pg_basebackup: base backup completed
```

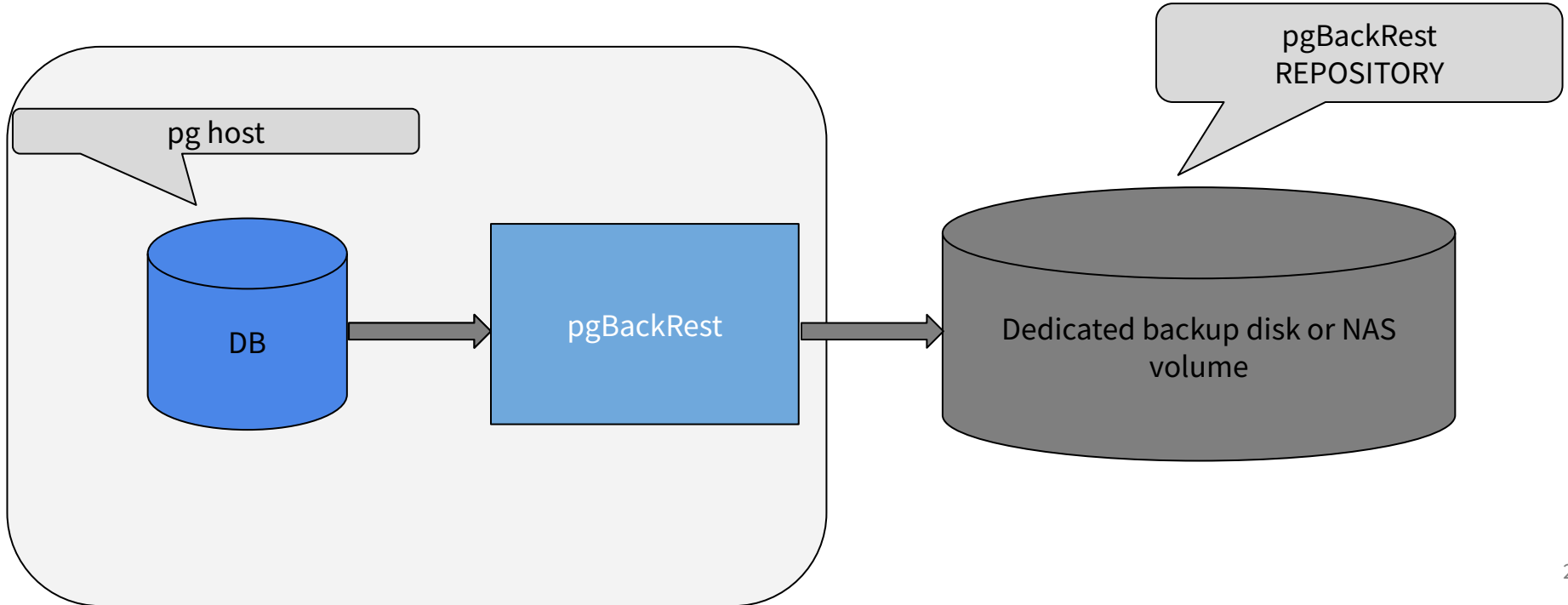
```
...LOG: checkpoint starting: force wait
...LOG: checkpoint complete:
...
lsn=2/48000060, redo lsn=2/48000028
```

```
postgres@alesnbbs:~$ cat /tmp/pg16/backup_label
START WAL LOCATION: 2/48000028 (file 000000010000000200000048)
CHECKPOINT LOCATION: 2/48000060
BACKUP METHOD: streamed
BACKUP FROM: primary
START TIME: 2024-05-15 19:30:14 CEST
LABEL: test_backup
START TIMELINE: 1
```

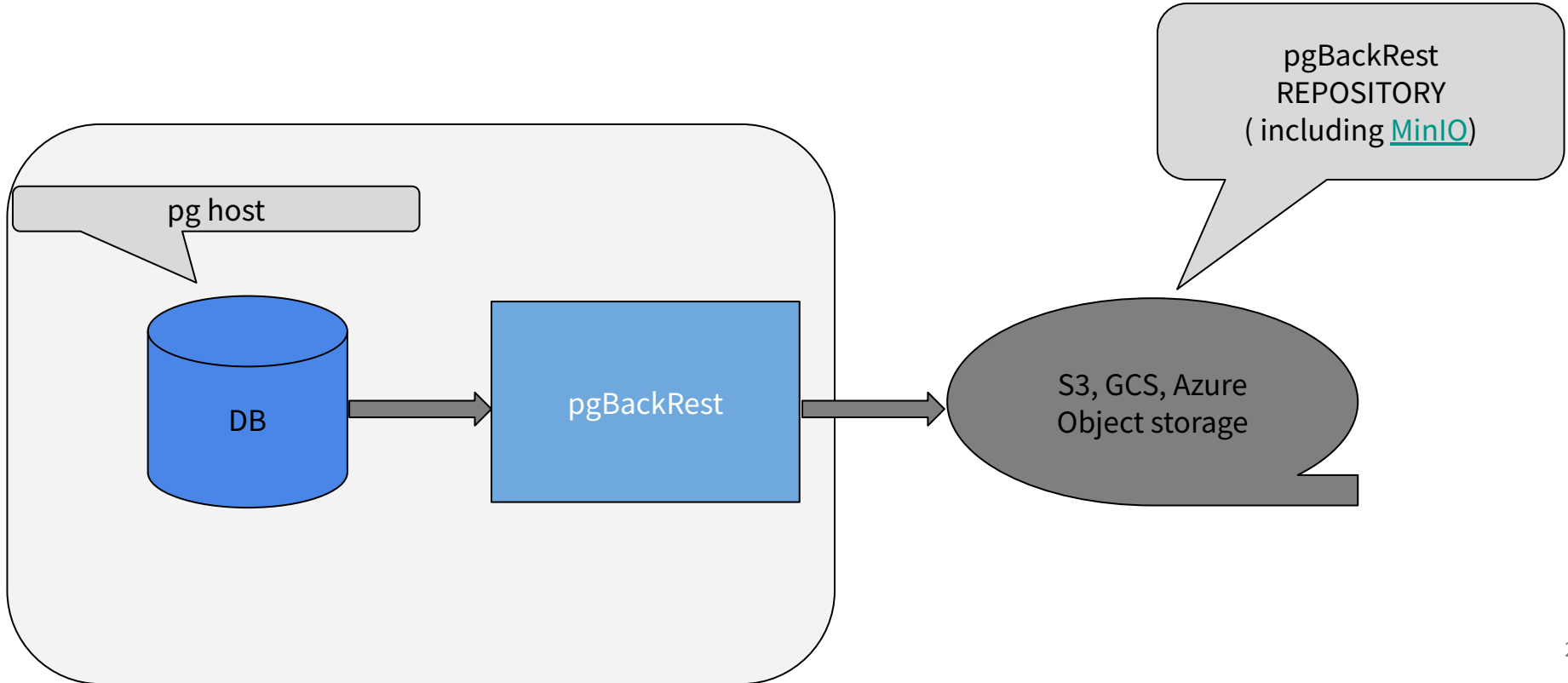
pgBackRest

- Automation
 - Retention policy
 - obsoleted WAL files removal
 - backup repository metadata ⇒ monitoring
- safety
 - validate page checksum (also pg_basebackup)
 - WAL file archivation
 - checksums for backed up files (recent pg_basebackup now create backup manifest with checksums, like pgBackRest)
- Advanced features
 - Full/differential/incremental backups
 - Backups parallelization
 - Resume failed backup
 - Differential restore
 - ...

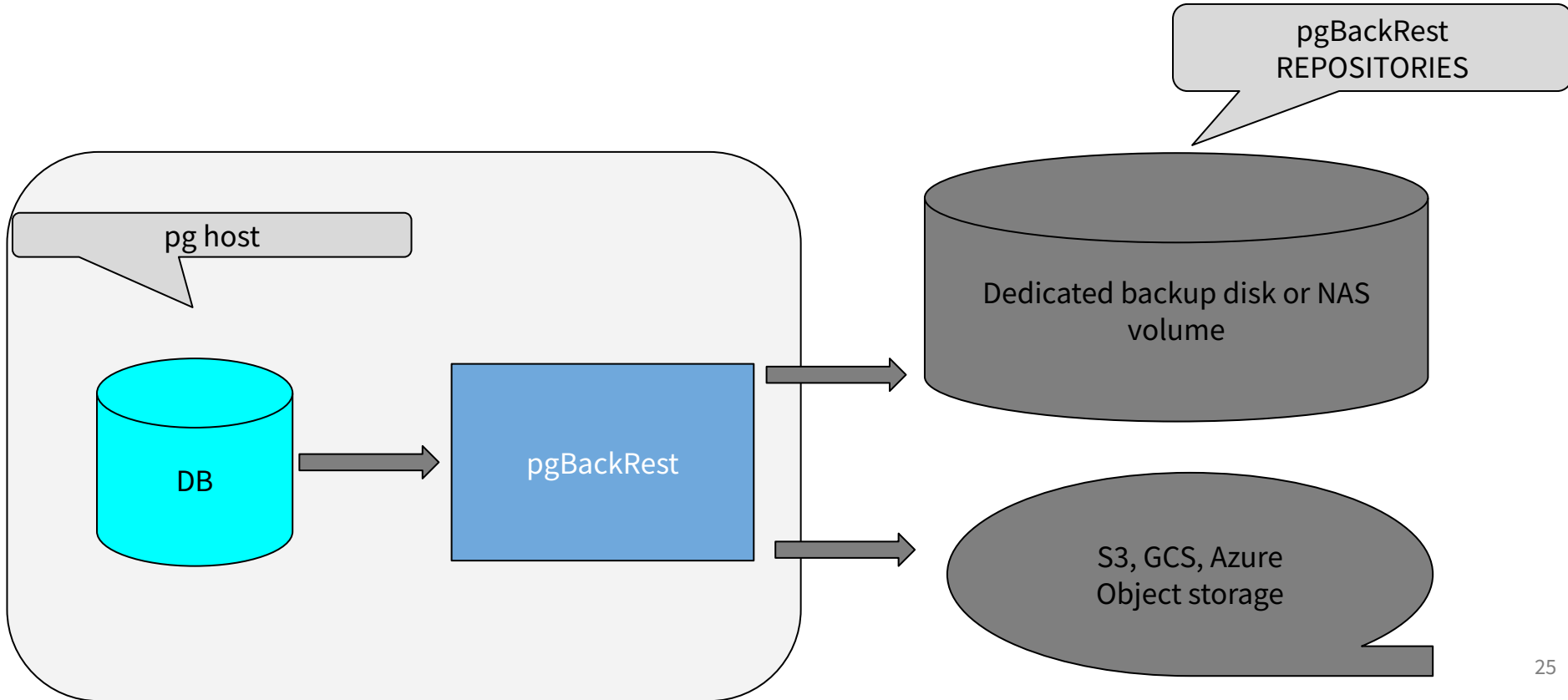
Local backup



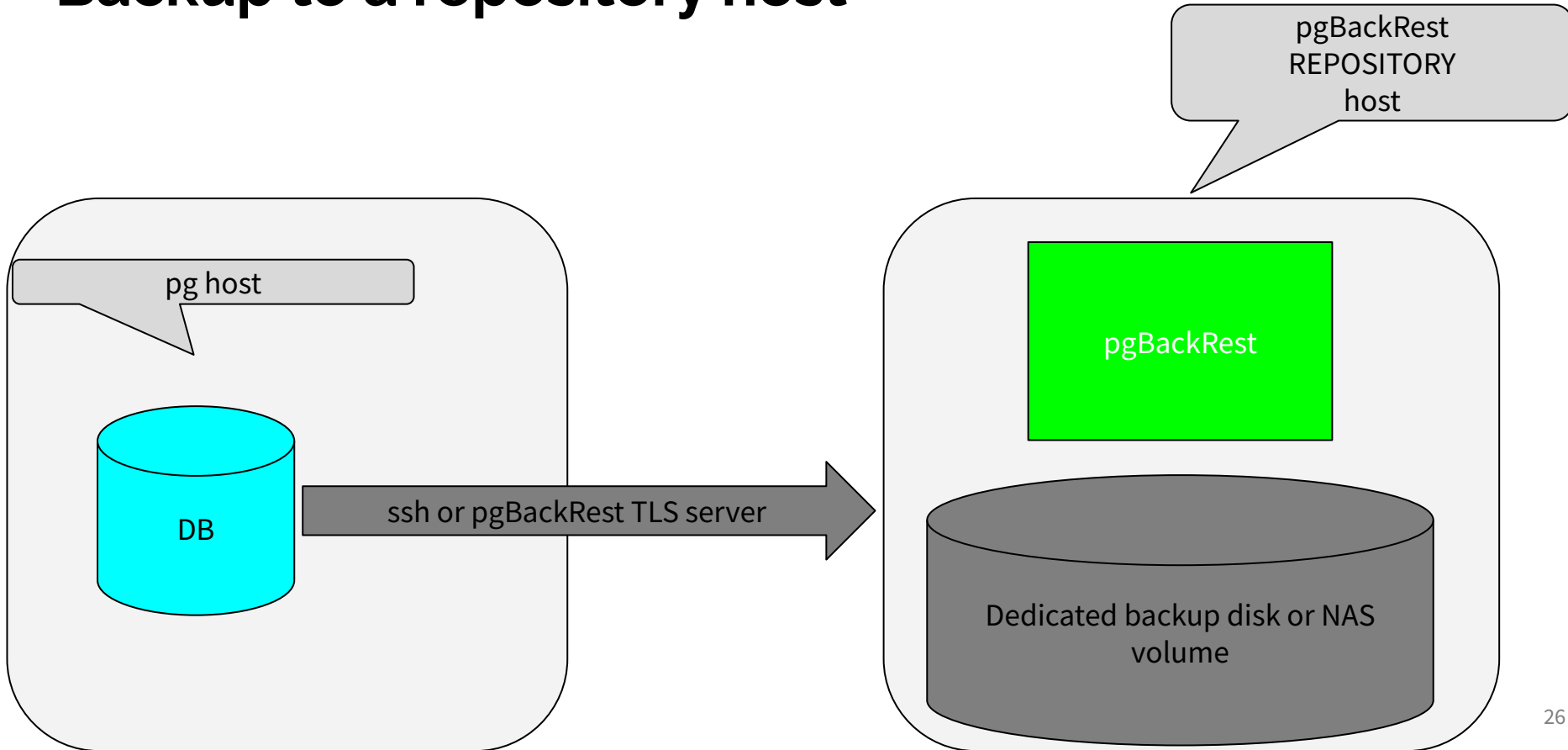
Local backup to Cloud



Local backup to multiple repositories



Backup to a repository host



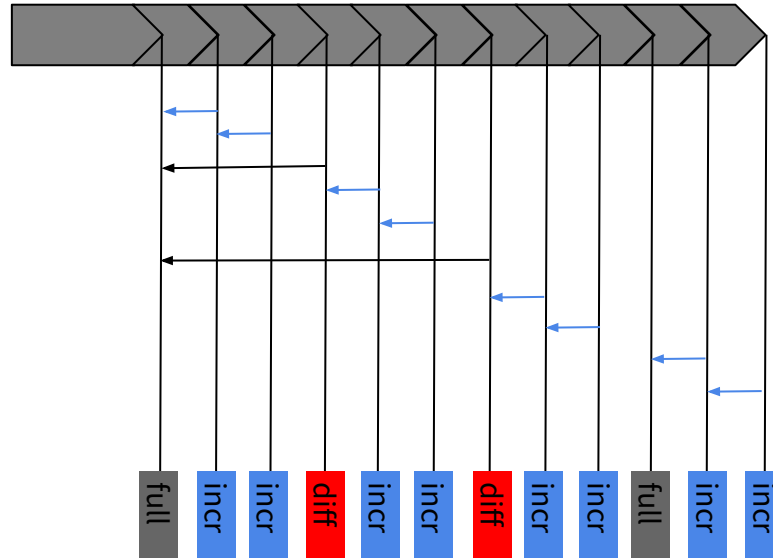
Installation

- OS package or get it from [Crunchy Data](#)
- Building from source is possible, but not recommended by the Crunchy Data as an installation procedure

Packages and configuration file(s)

- Default
 - `/etc/pgbackrest/pgbackrest.conf`
 - directory `/etc/pgbackrest/conf.d` for included files
- YUM, APT...
 - creates configuration file `/etc/pgbackrest.conf` a long time ago deprecated location.
 - the “packages” location still works

Full, incremental and differential backups



Minimal setup – example

- install and start PostgreSQL instance
 - enable `archive_mode` and configure `archive_command` for pgBackRest
- pgBackRest
 - install OS package
 - configure pgBackRest - PostgreSQL instance `DATA_DIR` and `retention`
 - create stanza for PostgreSQL instance (including slave instances if any)
 - run pgBackRest check command (good practice, not mandatory)
 - create a backup :-)
 - enjoy, for a while
 - on day later... restore the PostgreSQL instance
 - hopefully to build a DB slave instance...

install and start PostgreSQL instance

```
sudo dnf install -y  
https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86\_64  
/pgdg-redhat-repo-latest.noarch.rpm
```

```
sudo dnf -qy module disable postgresql  
sudo dnf install -y postgresql15-server postgresql15-contrib  
sudo /usr/pgsql-15/bin/postgresql-15-setup initdb  
sudo systemctl enable postgresql-15  
sudo systemctl start postgresql-15
```

pgBackRest - install OS package

```
sudo dnf install -y epel-release
```

```
sudo dnf install -y pgbackrest-2.51
```

```
# default after package installation
```

```
$ cat /etc/pgbackrest.conf
```

```
[global]
```

```
repo1-path=/var/lib/pgbackrest
```

```
#[main]
```

```
#pg1-path=/var/lib/pgsql/10/data
```

```
$ ls -ld /var/lib/pgbackrest/
```

```
drwx-----. 2 postgres postgres 6 Mar 26 22:52 /var/lib/pgbackrest/
```

configure pgBackRest

```
$ sudo -iu postgres psql -c 'show data_directory';
```

```
data_directory
```

```
-----
```

```
/var/lib/pgsql/15/data
```

```
# create configuration based on PG DATA_DIR
```

```
cat << EOF | sudo bash -c "cat > /etc/pgbackrest.conf"
```

```
[global]
```

```
repo1-path=/var/lib/pgbackrest
```

```
[main]
```

```
pg1-path=/var/lib/pgsql/15/data
```

```
EOF
```


create pgBackrest stanza

```
$ sudo -iu postgres pgbackrest info  
No stanzas exist in the repository.
```

```
$ sudo -iu postgres pgbackrest --stanza=main stanza-create
```

```
$ sudo -iu postgres pgbackrest info  
stanza: main  
  status: error (no valid backups)  
  cipher: none  
  
db (current)  
  wal archive min/max (15): none present
```

pgBackRest Stanza

- configuration section name for pg-host (and it's slave instances)
- directory on repository (S3 bucket path...)
 - data
 - archived logs
 - meta-data for the database instance and stored backups
- `<repo1-path>/backup/<stanza_name>`
 - `/var/lib/pgbackrest/backup/main/`
- `<repo1-path>/archive/<stanza_name>`
 - `/var/lib/pgbackrest/archive/main/`

pgBackrest stanza

```
$ sudo ls -l /var/lib/pgbackrest/backup/main/
```

```
total 8
```

```
-rw-r-----. 1 postgres postgres 370 May 25 20:28 backup.info
```

```
-rw-r-----. 1 postgres postgres 370 May 25 20:28 backup.info.copy
```

```
$ sudo ls -l /var/lib/pgbackrest/archive/main/
```

```
total 8
```

```
-rw-r-----. 1 postgres postgres 253 May 25 20:28 archive.info
```

```
-rw-r-----. 1 postgres postgres 253 May 25 20:28 archive.info.copy
```

pgBackrest stanza - backup info example

```
$ sudo cat /var/lib/pgbackrest/backup/main/backup.info
```

```
[backrest]
```

```
backrest-format=5
```

```
backrest-version="2.51"
```

```
[db]
```

```
db-catalog-version=202209061
```

```
db-control-version=1300
```

```
db-id=1
```

```
db-system-id=7373031384101894753
```

```
db-version="15"
```

```
[db:history]
```

```
1={"db-catalog-version":202209061,"db-control-version":1300,"db-system-id":7373031384101894753,"db-version":"15"}
```

```
[backrest]
```

```
backrest-checksum="1f1287e68228968a8b4c550737244967f1c82c50"
```

PostgreSQL instance side

```
$ sudo -iu postgres /usr/pgsql-15/bin/pg_controldata -D  
/var/lib/pgsql/15/data | head -10  
pg_control version number:          1300  
Catalog version number:            202209061  
Database system identifier:         7373031384101894753  
Database cluster state:             in production  
pg_control last modified:           Sat 25 May 2024 08:29:15 PM UTC  
Latest checkpoint location:         0/25E7FC8  
Latest checkpoint's REDO location:  0/25E7F90  
Latest checkpoint's REDO WAL file:  000000010000000000000000  
Latest checkpoint's TimeLineID:     1  
Latest checkpoint's PrevTimeLineID: 1
```

```
pg_control_system()  
pg_control_checkpoint()
```

run pgBackRest check command

```
$ sudo -iu postgres pgbackrest --stanza=main check  
ERROR: [087]: archive_mode must be enabled
```

```
$ cat <<EOF | sudo -iu postgres psql -v ON_ERROR_STOP=on  
ALTER SYSTEM SET archive_mode = on;  
ALTER SYSTEM SET archive_command = '/bin/true';  
EOF
```

```
$ sudo systemctl restart postgresql-15.service
```

```
$ sudo -iu postgres pgbackrest --stanza=main check  
ERROR: [068]: archive_command '/bin/true' must contain pgbackrest
```

PostgreSQL instance side

```
$ cat <<EOF | sudo -iu postgres psql -v ON_ERROR_STOP=on
ALTER SYSTEM SET archive_command = 'pgbackrest --stanza=main archive-push %p';
SELECT pg_reload_conf();
EOF
ALTER SYSTEM
  pg_reload_conf
-----
 t
(1 row)
```

```
$ sudo -iu postgres pgbackrest --stanza=main check
$ echo $?
0
```

Configure also **restore_command** !

```
restore_command = 'pgbackrest --stanza=main archive-get %f "%p"'
```

pgBackrest log-level-console option

```
$ sudo -iu postgres pgbackrest --log-level-console=detail --stanza=main check
<TS> P00 INFO: check command begin 2.51: --exec-id=15151-0fcc4bc2 --log-level-console=detail
--pg1-path=/var/lib/pgsql/15/data --repo1-path=/var/lib/pgbackrest --stanza=main
<TS> P00 INFO: check repo1 configuration (primary)
<TS> P00 INFO: check repo1 archive for WAL (primary)
<TS> P00 INFO: WAL segment 0000000100000000000000003 successfully archived to
'/var/lib/pgbackrest/archive/main/15-1/0000000100000000/00000001000000000000003-36f1c7cace599b60fc8aee
513adf06e1d815a4bc.gz' on repo1
2024-05-25 21:01:39.099 P00 INFO: check command end: completed successfully (914ms)
```

```
$ sudo -iu postgres pgbackrest --log-level-console=detail --stanza=main info
stanza: main
  status: error (no valid backups)
  cipher: none

  db (current)
  wal archive min/max (15): 0000000100000000000000002/000000010000000000000003
```


Create the first backup

```
$ sudo -iu postgres pgbackrest --log-level-console=detail --stanza=main --type=incr backup
2024-05-25 21:03:35.265 P00 INFO: backup command begin 2.51: --exec-id=15265-654d2979 --log-level-console=detail
--pg1-path=/var/lib/pgsql/15/data --repo1-path=/var/lib/pgbackrest --stanza=main --type=incr
WARN: option 'repo1-retention-full' is not set for 'repo1-retention-full-type=count', the repository may run out of space
HINT: to retain full backups indefinitely (without warning), set option 'repo1-retention-full' to the maximum.
WARN: no prior backup exists, incr backup has been changed to full
2024-05-25 21:03:35.775 P00 INFO: execute non-exclusive backup start: backup begins after the next regular checkpoint completes
2024-05-25 21:03:36.478 P00 INFO: backup start archive = 000000010000000000000005, lsn = 0/5000028
2024-05-25 21:03:36.478 P00 INFO: check archive for prior segment 000000010000000000000004
2024-05-25 21:03:37.094 P01 DETAIL: backup file /var/lib/pgsql/15/data/base/5/1255 (768KB, 2.04%) checksum
01a5a338a7a9fec25b6e916262d90a2b3cd197d
2024-05-25 21:03:37.136 P01 DETAIL: backup file /var/lib/pgsql/15/data/base/4/1255 (768KB, 4.09%) checksum
01a5a338a7a9fec25b6e916262d90a2b3cd197d
...
2024-05-25 21:03:41.770 P01 DETAIL: backup file /var/lib/pgsql/15/data/base/1/13425 (0B, 100.00%)
2024-05-25 21:03:41.871 P00 INFO: execute non-exclusive backup stop and wait for all WAL segments to archive
2024-05-25 21:03:42.072 P00 INFO: backup stop archive = 000000010000000000000005, lsn = 0/5000138
2024-05-25 21:03:42.076 P00 DETAIL: wrote 'backup_label' file returned from backup stop function
2024-05-25 21:03:42.077 P00 INFO: check archive for segment(s) 000000010000000000000005:000000010000000000000005
2024-05-25 21:03:42.093 P00 INFO: new backup label = 20240525-210335F
2024-05-25 21:03:42.138 P00 INFO: full backup size = 36.7MB, file total = 1572
2024-05-25 21:03:42.139 P00 INFO: backup command end: completed successfully (6878ms)
2024-05-25 21:03:42.139 P00 INFO: expire command begin 2.51: --exec-id=15265-654d2979 --log-level-console=detail
--repo1-path=/var/lib/pgbackrest --stanza=main
2024-05-25 21:03:42.145 P00 INFO: option 'repo1-retention-archive' is not set - archive logs will not be expired
2024-05-25 21:03:42.145 P00 INFO: expire command end: completed successfully (6ms)
```

Show backup(s) information

```
$ sudo -iu postgres pgbackrest --log-level-console=detail --stanza=main info
stanza: main
  status: ok
  cipher: none

  db (current)
  wal archive min/max (15):
00000001000000000000000002/00000001000000000000000005

  full backup: 20240525-210335F
    timestamp start/stop: 2024-05-25 21:03:35+00 / 2024-05-25 21:03:41+00
    wal start/stop: 00000001000000000000000005 / 00000001000000000000000005
    database size: 36.7MB, database backup size: 36.7MB
    repo1: backup set size: 4.9MB, backup size: 4.9MB
```

Backup info in a machine readable format

```
$ sudo -iu postgres pgbackrest --output=json info | jq
```

```
[
  {
    "archive": [
      {
        "database": {
          "id": 1,
          "repo-key": 1
        },
        "id": "15-1",
        "max": "000000010000000000000029",
        "min": "000000010000000000000021"
      }
    ],
  },
]
```

```
"backup": [
  {
    "archive": {
      "start": "000000010000000000000021",
      "stop": "000000010000000000000021"
    },
    "backrest": {
      "format": 5,
      "version": "2.51"
    },
    "database": {
      "id": 1,
      "repo-key": 1
    },
    "error": false,
    "info": {
      "delta": 38455698,
      "repository": {
        "delta": 5108888,
        "size": 5108888
      },
      "size": 38455698
    },
    "label": "20240525-212324F",
    "lsn": {
      "start": "0/21000028",
      "stop": "0/21000100"
    },
    "prior": null,
    "reference": null,
    "timestamp": {
      "start": 1716672204,
      "stop": 1716672212
    },
    "type": "full"
  },
]
```

enjoy for a while ... before performing a restore



A gory mistake....

```
[zelenya@pg-sales ~]$ sudo systemctl stop postgresql-15.service  
[zelenya@pg-sales ~]$ sudo rm -rf /var/lib/pgsql/15/data
```

- Do not copy & paste this into your environment
- If you do so, do not blame me, I've warned you :-)

Restore time

```
$ sudo -iu postgres pgbackrest --stanza=main --log-level-console=detail restore
2024-05-27 20:55:24.102 P00 INFO: restore command begin 2.51: --exec-id=2543-28099712
--log-level-console=detail --pg1-path=/var/lib/pgsql/15/data --repo1-path=/var/lib/pgbackrest
--stanza=main
2024-05-27 20:55:24.134 P00 INFO: repo1: restore backup set 20240525-213322F, recovery will start at
2024-05-25 21:33:22
2024-05-27 20:55:24.134 P00 DETAIL: check '/var/lib/pgsql/15/data' exists
2024-05-27 20:55:24.135 P00 DETAIL: create path '/var/lib/pgsql/15/data/base'
2024-05-27 20:55:24.135 P00 DETAIL: create path '/var/lib/pgsql/15/data/base/1'
2024-05-27 20:55:24.135 P00 DETAIL: create path '/var/lib/pgsql/15/data/base/16408'
2024-05-27 20:55:24.135 P00 DETAIL: create path '/var/lib/pgsql/15/data/base/16409'
...
2024-05-27 20:55:28.003 P00 DETAIL: sync path '/var/lib/pgsql/15/data/pg_wal/archive_status'
2024-05-27 20:55:28.003 P00 DETAIL: sync path '/var/lib/pgsql/15/data/pg_xact'
2024-05-27 20:55:28.010 P00 INFO: restore global/pg_control (performed last to ensure aborted restores
cannot be started)
2024-05-27 20:55:28.010 P00 DETAIL: sync path '/var/lib/pgsql/15/data/global'
2024-05-27 20:55:28.011 P00 INFO: restore size = 36.7MB, file total = 1572
2024-05-27 20:55:28.012 P00 INFO: restore command end: completed successfully (3915ms)
```

Recovery

```
$ sudo systemctl start postgresql-15.service
```

```
$ sudo tail -30 /var/lib/pgsql/15/data/log/postgresql-Mon.log
```

```
2024-05-27 20:57:12.900 UTC [2590] LOG: starting PostgreSQL 15.7 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3), 64-bit
2024-05-27 20:57:12.901 UTC [2590] LOG: listening on IPv6 address "::1", port 5432
2024-05-27 20:57:12.901 UTC [2590] LOG: listening on IPv4 address "127.0.0.1", port 5432
2024-05-27 20:57:12.903 UTC [2590] LOG: listening on Unix socket "/run/postgresql/.s.PGSQL.5432"
2024-05-27 20:57:12.908 UTC [2590] LOG: listening on Unix socket "/tmp/.s.PGSQL.5432"
2024-05-27 20:57:12.912 UTC [2594] LOG: database system was interrupted; last known up at 2024-05-25 21:33:22 UTC
2024-05-27 20:57:12.994 UTC [2594] LOG: starting archive recovery
2024-05-27 20:57:12.994 UTC [2594] LOG: starting backup recovery with redo LSN 0/28000028, checkpoint LSN 0/28000060, on timeline ID 1
2024-05-27 20:57:13.087 UTC [2594] LOG: restored log file "000000010000000000000028" from archive
2024-05-27 20:57:13.160 UTC [2594] LOG: redo starts at 0/28000028
2024-05-27 20:57:13.276 UTC [2594] LOG: restored log file "000000010000000000000029" from archive
2024-05-27 20:57:13.426 UTC [2594] LOG: restored log file "00000001000000000000002A" from archive
2024-05-27 20:57:13.497 UTC [2594] LOG: completed backup recovery with redo LSN 0/28000028 and end LSN 0/28000100
2024-05-27 20:57:13.497 UTC [2594] LOG: consistent recovery state reached at 0/28000100
2024-05-27 20:57:13.497 UTC [2590] LOG: database system is ready to accept read-only connections
2024-05-27 20:57:13.527 UTC [2594] LOG: redo done at 0/2A0001C0 system usage: CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.36 s
2024-05-27 20:57:13.622 UTC [2594] LOG: restored log file "00000001000000000000002A" from archive
2024-05-27 20:57:13.709 UTC [2594] LOG: selected new timeline ID: 2
2024-05-27 20:57:13.810 UTC [2594] LOG: archive recovery complete
2024-05-27 20:57:13.813 UTC [2592] LOG: checkpoint starting: end-of-recovery immediate wait
2024-05-27 20:57:13.827 UTC [2592] LOG: checkpoint complete: wrote 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 3 recycled; write=0.001 s, sync=0.004 s, total=0.017 s; sync files=2, longest=0.003 s, average=0.002 s; distance=49152 kB, estimate=49152 kB
2024-05-27 20:57:13.830 UTC [2590] LOG: database system is ready to accept connections
```

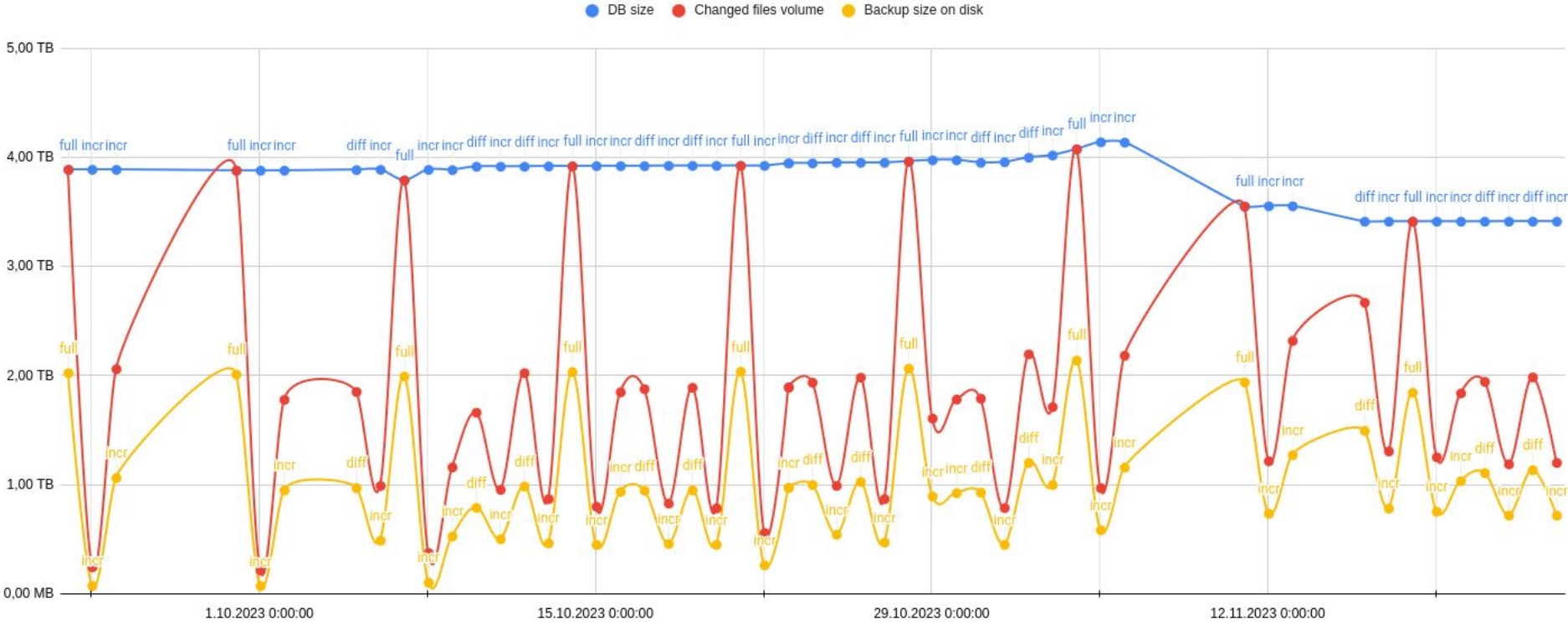
pgBackRest Log Files

```
$ sudo ls -l /var/log/pgbackrest/  
total 40  
-rw-r-----. 1 postgres postgres 23661 May 25 21:33 main-backup.log  
-rw-r-----. 1 postgres postgres  7971 May 25 21:33 main-expire.log  
-rw-r-----. 1 postgres postgres   755 May 27 20:55 main-restore.log  
-rw-r-----. 1 postgres postgres   399 May 25 20:28 main-stanza-create.log
```


Incremental backups

- Basic, based on file size & timestamp (or checksum if `--delta`)
 - easy, fast to decide whether a file needs to be copied into a backup
 - checksum will make it not that easy & fast
 - archives, data warehouse... whenever once DB files are written, there are unlikely file changes (DML)
 - granularity - datafile
- Block Level, based on changes within a file
 - requires IO to read each datafile and detect changed block - additional IO and CPU needed
 - **high efficiency from the resulting backup size** point of view
- Restore will combine full, differential and incremental backups behind the scenes to create a recoverable (by applying WAL files) instance
 - based on recovery target appropriate chain of backups is used
 - it is possible to define which backup use for a restore manually if needed (unlikely unless...)

File level incremental backup



Incremental backup info

```
full backup: 20240525-010508F
timestamp start/stop: 2024-05-25 01:05:08 / 2024-05-25 17:47:14
wal start/stop: 000000020003890800000071 / 000000020003899E000000B1
database size: 4004.6GB, database backup size: 4004.6GB
repo1: backup set size: 2004.8GB, backup size: 2004.8G

incr backup: 20240525-010508F_20240526-010508I
timestamp start/stop: 2024-05-26 01:05:08 / 2024-05-26 05:13:14
wal start/stop: 00000002000389F200000094 / 00000002000389F2000000C4
database size: 4005.3GB, database backup size: 1421.5GB
repo1: backup set size: 2004.6GB, backup size: 567.4GB
backup reference list: 20240525-010508

incr backup: 20240525-010508F_20240527-010503I
timestamp start/stop: 2024-05-27 01:05:03 / 2024-05-27 05:49:11
wal start/stop: 0000000200038A7100000028 / 0000000200038A7700000056
database size: 4007.7GB, database backup size: 1672.2GB
repo1: backup set size: 2005.3GB, backup size: 624.8GB
backup reference list: 20240525-010508F, 20240525-010508F_20240526-010508

diff backup: 20240525-010508F_20240528-010508D
timestamp start/stop: 2024-05-28 01:05:08 / 2024-05-28 05:50:05
wal start/stop: 0000000200038A9C0000008B / 0000000200038A9D00000000
database size: 4008GB, database backup size: 1696.5GB
repo1: backup set size: 2006.7GB, backup size: 631GB
backup reference list: 20240525-010508

incr backup: 20240525-010508F_20240529-010507I
timestamp start/stop: 2024-05-29 01:05:07 / 2024-05-29 05:18:51
wal start/stop: 0000000200038B1E0000000C / 0000000200038B1E00000077
database size: 4008.2GB, database backup size: 1403.5GB
repo1: backup set size: 2007GB, backup size: 560.9GB
backup reference list: 20240525-010508F,
20240525-010508F_20240528-010508

diff backup: 20240525-010508F_20240530-010508D
timestamp start/stop: 2024-05-30 01:05:08 / 2024-05-30 06:20:17
wal start/stop: 0000000200038BC200000043 / 0000000200038BC300000045
database size: 4008.4GB, database backup size: 1780.9GB
repo1: backup set size: 2006.4GB, backup size: 687.9GB
backup reference list: 20240525-010508F
```

~ 50%
compression

~ 35% changed
files (increment)

~ 60%
compression

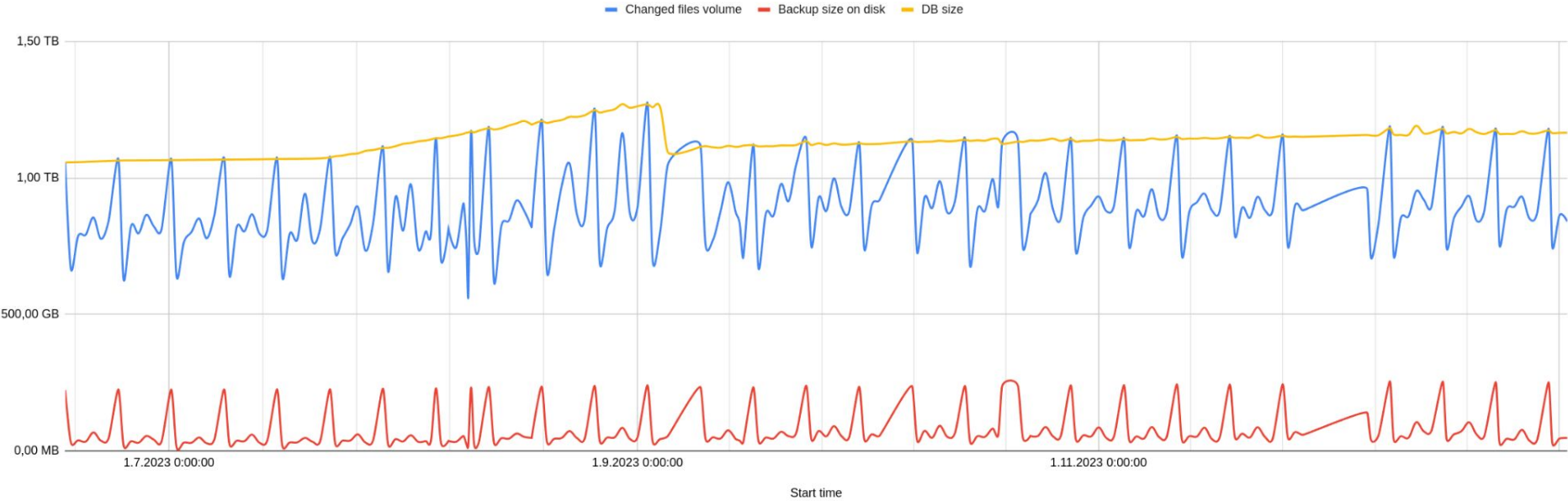
Full PG size
(uncompressed)

Files needed for restore using this
backup
(compressed, incl. referenced backups needed)

Backed up data
(uncompressed)

Only this backup files
(compressed)

Block level incremental backup



Block level incremental backup info

full backup: 20240525-01000F
timestamp start/stop: 2024-05-25 01:00:00+02 / 2024-05-25 14:42:12+02
wal start/stop: 0000000100019D8F0000002A / 0000000100019DA1000000080
database size: 8531.7GB, database backup size: 8531.7GB
repo1: backup size: 2526.4GB

~ 70%
compression

incr backup: 20240525-01000F_20240526-010001I
timestamp start/stop: 2024-05-26 01:00:01+02 / 2024-05-26 01:58:07+02
wal start/stop: 0000000100019DD20000006D / 0000000100019DD300000008F
database size: 8471.7GB, database backup size: 2345.4GB
repo1: backup size: 85.3GB
backup reference list: 20240525-01000F

~ 96%
compression?

~ 28% changed
files (increment)

block-level
increment

incr backup: 20240525-01000F_20240527-010000I
timestamp start/stop: 2024-05-27 01:00:00+02 / 2024-05-27 01:51:11+02
wal start/stop: 0000000100019E1100000080 / 0000000100019E120000004B
database size: 8477.8GB, database backup size: 2098.4GB
repo1: backup size: 74.9GB
backup reference list: 20240525-01000F, 20240525-01000F_20240526-010001I

incr backup: 20240525-01000F_20240528-010001I
timestamp start/stop: 2024-05-28 01:00:01+02 / 2024-05-28 01:46:54+02
wal start/stop: 0000000100019E33000000B9 / 0000000100019E3400000064
database size: 8482.9GB, database backup size: 2178.3GB
repo1: backup size: 58.7GB
backup reference list: 20240525-01000F, 20240525-01000F_20240526-010001I,
20240525-01000F_20240527-010000I

```
$ du -sh 20240525-01000F*
2,5T 20240525-01000F
86G 20240525-01000F_20240526-010001I
75G 20240525-01000F_20240527-010000I
59G 20240525-01000F_20240528-010001I
125G 20240525-01000F_20240529-010000D
86G 20240525-01000F_20240530-010001I
```

diff backup: 20240525-01000F_20240529-010000D
timestamp start/stop: 2024-05-29 01:00:00+02 / 2024-05-29 02:11:19+02
wal start/stop: 0000000100019E5A0000001A / 0000000100019E5B00000059
database size: 8489.7GB, database backup size: 2491.9GB
repo1: backup size: 124.7GB
backup reference list: 20240525-01000F

incr backup: 20240525-01000F_20240530-010001I
timestamp start/stop: 2024-05-30 01:00:01+02 / 2024-05-30 01:57:44+02
wal start/stop: 0000000100019E8A000000C2 / 0000000100019E8C00000010
database size: 8494.8GB, database backup size: 2311.5GB
repo1: backup size: 85.7GB
backup reference list: 20240525-01000F,
20240525-01000F_20240529-010000D

Help the way i like it

```
$ sudo -iu postgres pgbackrest help info
```

```
pgBackRest 2.51 - 'info' command help
```

```
Retrieve information about backups.
```

```
...
```

```
Command Options:
```

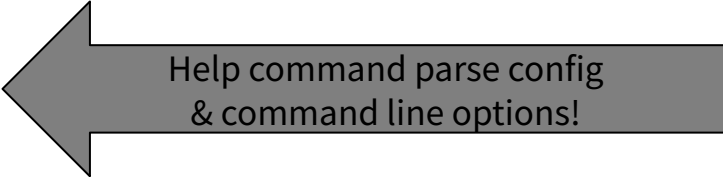
```
--output          output format [default=text]
--set             backup set to detail
--type           filter on backup type
```

```
General Options:
```

```
...
```

```
--config          pgBackRest configuration file
                  [default=/etc/pgbackrest/pgbackrest.conf]
--config-include-path path to additional pgBackRest
                  configuration files
                  [default=/etc/pgbackrest/conf.d]
--config-path     base path of pgBackRest configuration
                  files [default=/etc/pgbackrest]
...
--repo-type       type of storage used for the repository
                  [current=s3, default=posix]
```

- delete configuration file from packages and leverage pgbackrest default locations



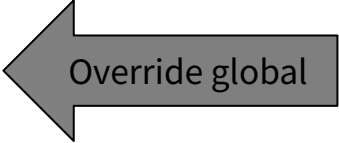
Help command parse config
& command line options!

Settings you may like

```
[global]
repo1-path=/pgbackrest/uat
repo1-type=s3
log-level-console=info
log-level-file=detail
repo1-s3-key-type=auto
repo1-s3-region=eu-central-1
repo1-s3-endpoint=s3.eu-central-1.amazonaws.com
repo1-s3-bucket=pgbackrest-bucket
repo1-retention-full=4
repo1-block=y
repo1-bundle=y
spool-path=/var/spool/pgbackrest
start-fast=y
archive-copy=y
buffer-size=16MiB
compress-type=lz4
compress-level=3
process-max=4
archive-async=y
```

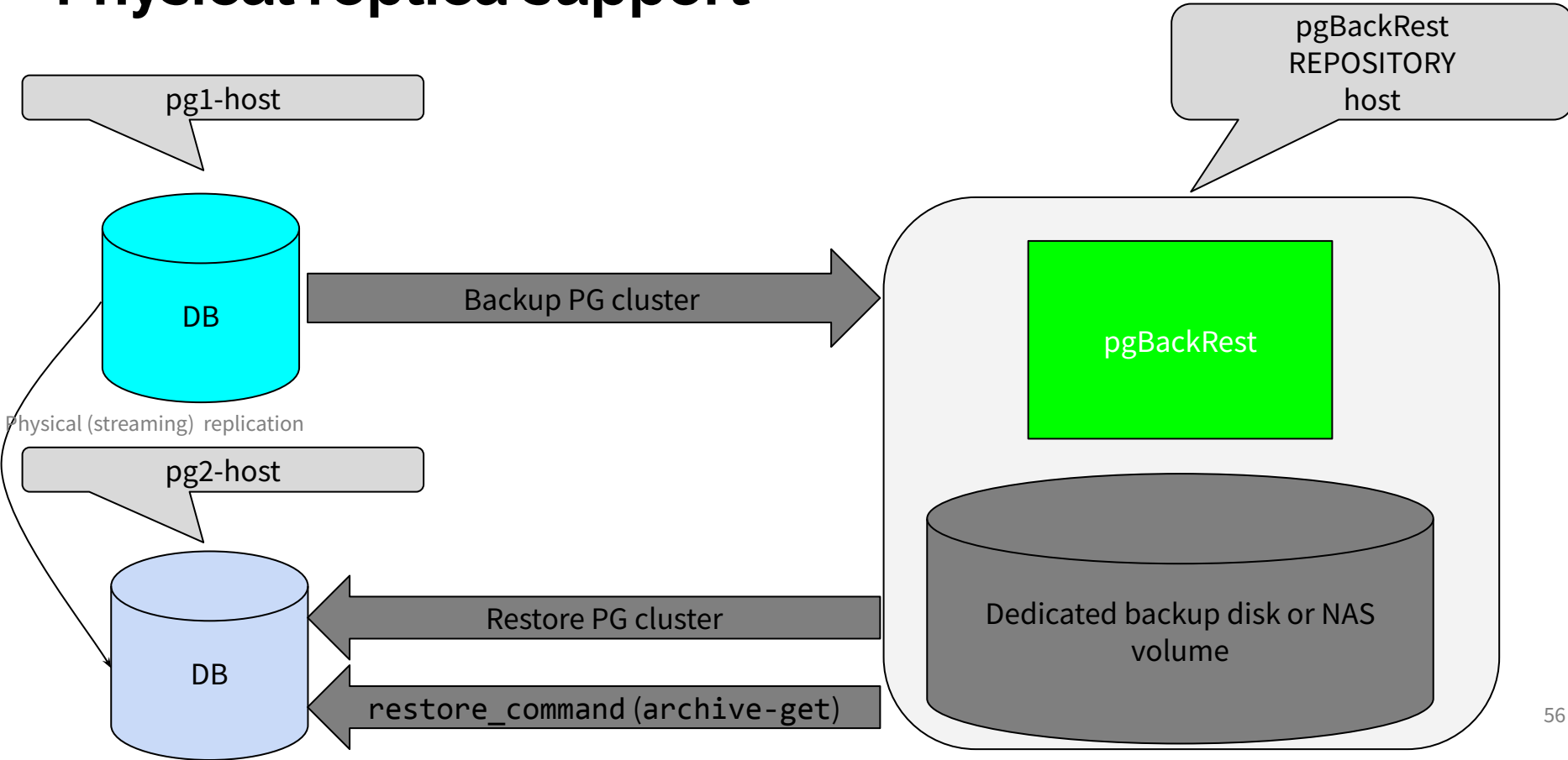


```
[global:archive-push]
log-level-console=warn
process-max=2
compress-type=gz
compress-level=2
```



```
[global:archive-get]
log-level-console=warn
```

Physical replica support



Dedicated repository host

- OS with pgBackRest (usually without PostgreSQL packages)
 - running pgBackRest TLS server or
 - ssh with passwordless communication with PostgreSQL host(s), also pg host(s) → repository host
 - act as shared storage for PG instances backup
 - each PG cluster in separated stanza
 - **BACKUPS** are started on REPOSITORY HOST, not on PG host(s)
 - EXACTLY same pgBackRest version on pgBackRest repository host as on PG host(s) is required
- PostgreSQL instance(s)
 - Configuration defining **repository host** and **stanza**
 - push wal files (archive_command)
 - **RESTORE** is performed on PG host

Help the way I like...

```
[zelenya@pg-web-master ~]$ sudo -u postgres pgbackrest help info
pgBackRest 2.51 - 'info' command help
```

Retrieve information about backups.

...

General Options:

...

<code>--compress-level-network</code>	network compression level [default=3]
<code>--config</code>	pgBackRest configuration file [default=/etc/pgbackrest/pgbackrest.conf]
<code>--config-include-path</code>	path to additional pgBackRest configuration files [default=/etc/pgbackrest/conf.d]
<code>--config-path</code>	base path of pgBackRest configuration files [default=/etc/pgbackrest]

WEB: Configure PostgreSQL master instance

```
ssh -A 192.168.121.128 # 192.168.121.128/24
```

```
sudo dnf install -y  
https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm  
sudo dnf -qy module disable postgresql  
sudo dnf install -y postgresql16-server postgresql16-contrib  
sudo /usr/pgsql-16/bin/postgresql-16-setup initdb  
sudo systemctl enable postgresql-16  
sudo systemctl start postgresql-16
```

```
sudo dnf install -y epel-release  
sudo dnf install -y pgbackrest-2.51
```

WEB: Configure PG instance and DBs

```
cat <<EOF | sudo -iu postgres psql -v ON_ERROR_STOP=on
ALTER SYSTEM SET cluster_name = 'web128';
ALTER SYSTEM SET archive_mode = on;
ALTER SYSTEM SET archive_command = '/bin/true';
CREATE DATABASE web;
CREATE DATABASE web_old;
\c web
CREATE TABLE IF NOT EXISTS web_pages (id serial PRIMARY KEY, content text);
INSERT INTO web_pages(content) VALUES ('The first web page ever.');
```

```
\c web_old
CREATE TABLE IF NOT EXISTS web_pages (id serial PRIMARY KEY, content text);
INSERT INTO web_pages(content) VALUES ('The page before the first one (ever)...');
EOF
```

WEB: Configure pgBackRest on pg host

```
sudo mkdir -p /etc/pgbackrest/conf.d
sudo mv /etc/pgbackrest.conf /etc/pgbackrest/pgbackrest.conf
```

```
sudo -u postgres cat /etc/pgbackrest/pgbackrest.conf
```

```
[global]
```

```
repo1-host=192.168.121.192
```

```
repo1-host-user=pgbackrest
```

```
archive-async=y
```

pgbackrest is default

```
[vgr_web]
```

```
pg1-path=/var/lib/pgsql/16/data
```

PGDATA

```
sudo -u postgres pgbackrest help archive-push archive-async
```

```
...
```

```
current: true
```

```
default: n
```

REPO: Install pgBackRest

```
ssh -A 192.168.121.192
```

```
sudo dnf install -y epel-release
```

```
sudo dnf -qy module disable postgresql
```

```
sudo dnf install -y
```

```
https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86\_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
sudo dnf install -y pgbackrest-2.51
```

```
sudo groupadd pgbackrest
```

```
sudo adduser -gpgbackrest -n pgbackrest
```

```
sudo mkdir -p /etc/pgbackrest/conf.d
```

```
sudo chmod -R 0755 /etc/pgbackrest
```

```
sudo mv /etc/pgbackrest.conf /etc/pgbackrest/pgbackrest.conf
```

```
sudo chmod 640 /etc/pgbackrest/pgbackrest.conf
```

```
sudo chown pgbackrest:pgbackrest /etc/pgbackrest/pgbackrest.conf
```

```
sudo chown pgbackrest:pgbackrest /var/log/pgbackrest
```

```
sudo chown pgbackrest:pgbackrest /var/lib/pgbackrest
```

Package defaults:

```
[zelenya@pgbackrest ~]$ id postgres  
uid=26(postgres) gid=26(postgres) groups=26(postgres)
```

```
[zelenya@pgbackrest ~]$ ls -ld /var/lib/pgbackrest/  
drwx-----. 2 postgres postgres 6 Mar 26 22:52 /var/lib/pgbackrest/
```

```
[zelenya@pgbackrest ~]$ ls -ld /var/log/pgbackrest/  
drwx-----. 2 postgres postgres 6 Mar 26 22:52 /var/log/pgbackrest/
```

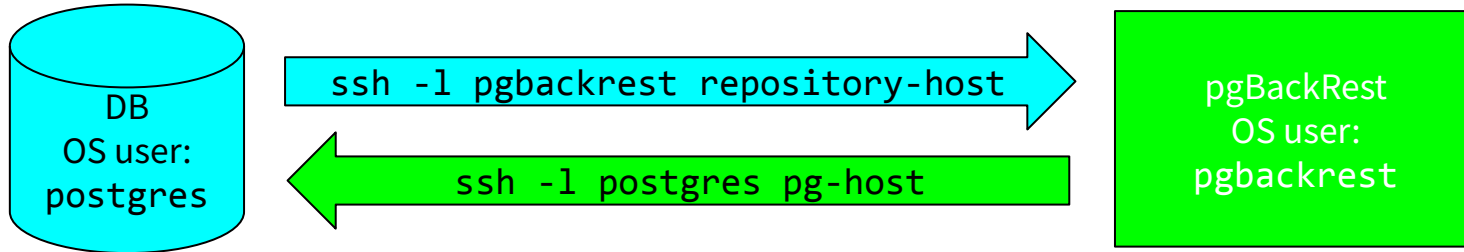
```
[zelenya@pgbackrest ~]$ ls -l /etc/pgbackrest.conf  
-rw-r--r--. 1 root root 82 Mar 26 22:52 /etc/pgbackrest.conf
```

REPO: Configure pgBackRest

```
sudo -u pgbackrest cat /etc/pgbackrest/pgbackrest.conf
[global]
log-level-console=info
log-level-file=detail
repo1-path=/var/lib/pgbackrest
repo1-retention-full=2
start-fast=y

[vgr_web]
pg1-path=/var/lib/pgsql/16/data
pg1-host=192.168.121.128
pg1-host-user=postgres
```

SSH passwordless for pgBackRest



- Ensure host ssh keys are unique
 - otherwise known_hosts updates will confuse you
 - possible reason - VM created by source image clone

SSH passwordless for pgBackRest

```
sudo -u postgres mkdir -m 750 -p /var/lib/pgsql/.ssh  
sudo -u postgres ssh-keygen -f /var/lib/pgsql/.ssh/id_rsa -t rsa -b 4096 -N ""
```

```
sudo -u pgbackrest mkdir -m 750 /home/pgbackrest/.ssh  
sudo -u pgbackrest ssh-keygen -f /home/pgbackrest/.ssh/id_rsa -t rsa -b 4096 -N ""
```

```
sudo -u postgres bash -c "ssh-keyscan -t rsa -H 192.168.121.192 \  
| tee -a /var/lib/pgsql/.ssh/known_hosts"
```

```
sudo -u pgbackrest bash -c "ssh-keyscan -t rsa -H 192.168.121.128 \  
| tee -a /home/pgbackrest/.ssh/known_hosts"
```

SSH passwordless for pgBackRest

```
(echo -n 'no-agent-forwarding,no-X11-forwarding,no-port-forwarding,' && \  
  echo -n 'command="/usr/bin/pgbackrest ${SSH_ORIGINAL_COMMAND#* }" ' && \  
  ssh 192.168.121.192 sudo cat /home/pgbackrest/.ssh/id_rsa.pub) | \  
  sudo -u postgres tee -a /var/lib/pgsql/.ssh/authorized_keys
```

```
(echo -n 'no-agent-forwarding,no-X11-forwarding,no-port-forwarding,' && \  
  echo -n 'command="/usr/bin/pgbackrest ${SSH_ORIGINAL_COMMAND#* }" ' && \  
  ssh 192.168.121.128 sudo cat /var/lib/pgsql/.ssh/id_rsa.pub) | \  
  sudo -u pgbackrest tee -a /home/pgbackrest/.ssh/authorized_keys
```

<https://pgbackrest.org/user-guide.html#repo-host/setup-ssh>

Create backup stanza

```
sudo -u postgres ssh -l pgbackrest 192.168.121.192
```

```
pgBackRest 2.51 - General help
```

```
Usage:
```

```
    pgbackrest [options] [command]
```

```
...
```

```
Use 'pgbackrest help [command]' for more information.
```

```
Connection to 192.168.121.192 closed
```

```
sudo -u pgbackrest ssh 192.168.121.128 -l postgres
postgres@192.168.121.128's password:
```

```
sudo restorecon -FRv /var/lib/pgsql/.ssh/
```

```
Relabeled /var/lib/pgsql/.ssh from unconfined_u:object_r:postgresql_db_t:s0 to system_u:object_r:ssh_home_t:s0
```

```
... # if enabled and configured by hand selinux sometimes needs a fix
```

```
sudo -u pgbackrest ssh 192.168.121.128 -l postgres
```

```
pgBackRest 2.51 - General help
```

```
Usage:
```

```
    pgbackrest [options] [command]
```

```
...
```

```
Use 'pgbackrest help [command]' for more information.
```

```
Connection to 192.168.121.128 closed.
```

Pg host and repository configuration files

```
[global]
repo1-host=192.168.121.192
repo1-host-user=pgbackrest
archive-async=y
```

```
[vgr_web]
pg1-path=/var/lib/pgsql/16/data
```

```
[global]
log-level-console=info
log-level-file=detail
repo1-path=/var/lib/pgbackrest
repo1-retention-full=2
start-fast=y
```

```
[vgr_web]
pg1-path=/var/lib/pgsql/16/data
pg1-host=192.168.121.128
pg1-host-user=postgres
```

Create backup stanza

```
sudo -u postgres pgbackrest info  
No stanzas exist in the repository.
```

```
sudo -u postgres pgbackrest --log-level-console=detail --stanza=vgr_web stanza-create  
2024-06-02 14:30:45.063 P00 INFO: stanza-create command begin 2.51: --exec-id=3648-7b2e4e19 --log-level-console=detail  
--pg1-path=/var/lib/pgsql/16/data --repo1-host=192.168.121.192 --repo1-host-user=pgbackrest --stanza=vgr_web  
2024-06-02 14:30:45.670 P00 INFO: stanza-create for stanza 'vgr_web' on repo1  
2024-06-02 14:30:46.200 P00 INFO: stanza-create command end: completed successfully (1139ms)
```

```
sudo -u pgbackrest pgbackrest --log-level-console=detail --stanza=vgr_web stanza-create  
2024-06-02 20:28:08.216 P00 INFO: stanza-create command begin 2.51: --exec-id=143885-05b67ddb  
...  
2024-06-02 20:28:10.322 P00 INFO: stanza-create for stanza 'vgr_web' on repo1  
2024-06-02 20:28:10.323 P00 INFO: stanza 'vgr_web' already exists on repo1 and is valid  
2024-06-02 20:28:10.524 P00 INFO: stanza-create command end: completed successfully (2312ms)
```

Configure archive_command

```
cat <<EOF | sudo -iu postgres psql -v ON_ERROR_STOP=on
ALTER SYSTEM SET archive_command = 'pgbackrest --stanza=vgr_web archive-push %p';
SELECT pg_reload_conf();
EOF
```

```
sudo -u postgres pgbackrest --log-level-console=detail --stanza=vgr_web check
```

```
2024-06-02 14:37:01.005 P00 INFO: check command begin 2.51: --exec-id=3798-d7283387 --log-level-console=detail
--log-level-file=detail --pg1-path=/var/lib/pgsql/16/data --repo1-host=192.168.121.192 --repo1-host-user=pgbackrest
--stanza=vgr_web
2024-06-02 14:37:01.613 P00 INFO: check repo1 configuration (primary)
2024-06-02 14:37:02.418 P00 INFO: check repo1 archive for WAL (primary)
2024-06-02 14:37:02.724 P00 INFO: WAL segment 000000010000000000000004 successfully archived to
'/var/lib/pgbackrest/archive/vgr_web/16-1/0000000100000000/000000010000000000000004-aa42aa6f2c8534db8eaadd2737ebfe4e64e8a
afc.gz' on repo1
2024-06-02 14:37:02.824 P00 INFO: check command end: completed successfully (1824ms)
```

Backup the WEB pg cluster

```
sudo -u postgres pgbackrest --log-level-console=detail --stanza=vgr_web backup
2024-06-02 14:37:51.358 P00 INFO: backup command begin 2.51: --exec-id=3830-54bc0eec --log-level-console=detail
--log-level-file=detail --pg1-path=/var/lib/pgsql/16/data --repo1-host=192.168.121.192 --stanza=vgr_web
ERROR: [072]: backup command must be run on the repository host
2024-06-02 14:37:51.359 P00 INFO: backup command end: aborted with exception [072]
```

Backup the WEB pg cluster

```
sudo -u pgbackrest pgbackrest --log-level-console=detail --stanza=vgr_web backup
```

```
2024-06-02 14:38:14.949 P00 INFO: backup command begin 2.51: --exec-id=5482-9e2b9b7e --log-level-console=detail --log-level-file=detail --pg1-host=192.168.121.128 --pg1-host-user=postgres --pg1-path=/var/lib/pgsql/16/data --repo1-path=/var/lib/pgbackrest --repo1-retention-full=2 --stanza=vgr_web --start-fast
```

```
WARN: no prior backup exists, incr backup has been changed to full
```

```
2024-06-02 14:38:16.083 P00 INFO: execute non-exclusive backup start: backup begins after the requested immediate checkpoint completes
```

```
2024-06-02 14:38:16.794 P00 INFO: backup start archive = 000000010000000000000007, lsn = 0/7000028
```

```
2024-06-02 14:38:16.794 P00 INFO: check archive for prior segment 000000010000000000000006
```

```
2024-06-02 14:38:18.594 P01 DETAIL: backup file 192.168.121.128:/var/lib/pgsql/16/data/base/5/1255 (784KB, 2.07%)  
checksum 07ccfa98a2e60a0f714d809ba718542602ce2f7b
```

```
2024-06-02 14:38:18.646 P01 DETAIL: backup file 192.168.121.128:/var/lib/pgsql/16/data/base/4/1255 (784KB, 4.15%)  
checksum 07ccfa98a2e60a0f714d809ba718542602ce2f7b
```

```
...
```

```
2024-06-02 14:38:25.891 P00 INFO: execute non-exclusive backup stop and wait for all WAL segments to archive
```

```
2024-06-02 14:38:26.095 P00 INFO: backup stop archive = 000000010000000000000007, lsn = 0/7000138
```

```
2024-06-02 14:38:26.109 P00 DETAIL: wrote 'backup_label' file returned from backup stop function
```

```
2024-06-02 14:38:26.111 P00 INFO: check archive for segment(s) 000000010000000000000007:000000010000000000000007
```

```
2024-06-02 14:38:26.531 P00 INFO: new backup label = 20240602-143816F
```

```
2024-06-02 14:38:26.599 P00 INFO: full backup size = 36.8MB, file total = 1576
```

```
2024-06-02 14:38:26.600 P00 INFO: backup command end: completed successfully (11656ms)
```

```
2024-06-02 14:38:26.601 P00 INFO: expire command begin 2.51: --exec-id=5482-9e2b9b7e --log-level-console=detail  
--log-level-file=detail --repo1-path=/var/lib/pgbackrest --repo1-retention-full=2 --stanza=vgr_web
```

```
2024-06-02 14:38:26.607 P00 INFO: expire command end: completed successfully (7ms)
```

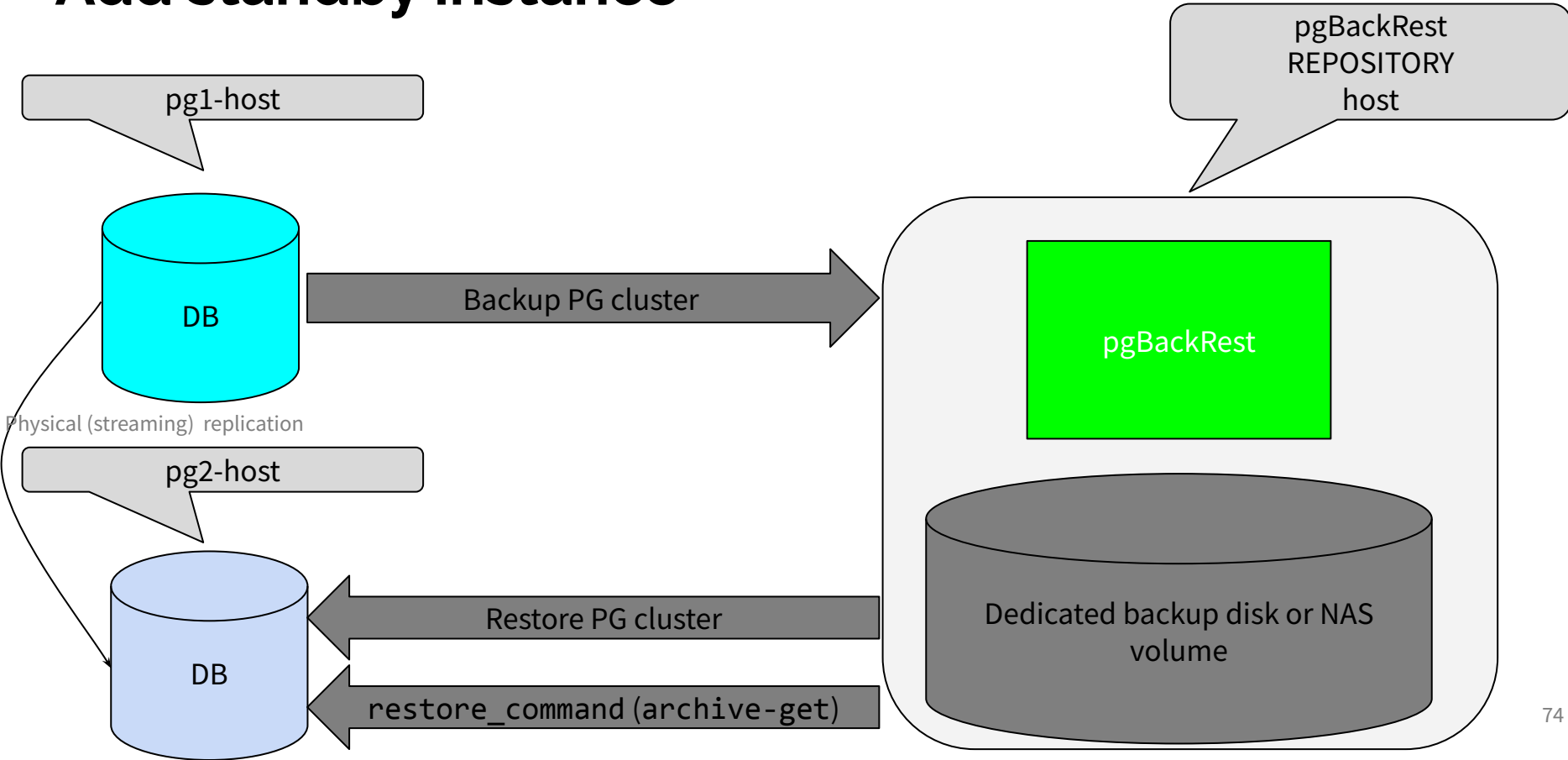

The backup is in place

```
sudo -u pgbackrest pgbackrest --log-level-console=detail --stanza=vgr_web info
stanza: vgr_web
  status: ok
  cipher: none

db (current)
wal archive min/max (16): 00000001000000000000000004/000000010000000000000007

full backup: 20240602-143816F
  timestamp start/stop: 2024-06-02 14:38:16+00 / 2024-06-02 14:38:25+00
  wal start/stop: 00000001000000000000000007 / 000000010000000000000007
  database size: 36.8MB, database backup size: 36.8MB
  repo1: backup set size: 4.9MB, backup size: 4.9MB
```

Add standby instance



Setup standby pg host

```
ssh -A 192.168.121.129
```

```
sudo dnf install -y
```

```
https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86\_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
sudo dnf -qy module disable postgresql
```

```
sudo dnf install -y postgresql16-server postgresql16-contrib
```

```
sudo /usr/pgsql-16/bin/postgresql-16-setup initdb
```

```
sudo systemctl enable postgresql-16
```

```
sudo systemctl start postgresql-16
```

```
sudo dnf install -y epel-release
```

```
sudo dnf install -y pgbackrest-2.51
```

Standby and repository passwordless SSH

```
sudo -u postgres mkdir -m 750 -p /var/lib/pgsql/.ssh
sudo -u postgres ssh-keygen -f /var/lib/pgsql/.ssh/id_rsa -t rsa -b 4096 -N ""

sudo -u postgres bash -c "ssh-keyscan -t rsa -H 192.168.121.192 | tee -a
/var/lib/pgsql/.ssh/known_hosts"

(echo -n 'no-agent-forwarding,no-X11-forwarding,no-port-forwarding,' && \
echo -n 'command="/usr/bin/pgbackrest ${SSH_ORIGINAL_COMMAND#* }" ' && \
ssh 192.168.121.192 sudo cat /home/pgbackrest/.ssh/id_rsa.pub) | \
sudo -u postgres tee -a /var/lib/pgsql/.ssh/authorized_keys
```

```
sudo -u pgbackrest bash -c "ssh-keyscan -t rsa -H 192.168.121.129 | tee -a
/home/pgbackrest/.ssh/known_hosts"

(echo -n 'no-agent-forwarding,no-X11-forwarding,no-port-forwarding,' && \
echo -n 'command="/usr/bin/pgbackrest ${SSH_ORIGINAL_COMMAND#* }" ' && \
ssh 192.168.121.129 sudo cat /var/lib/pgsql/.ssh/id_rsa.pub) | \
sudo -u pgbackrest tee -a /home/pgbackrest/.ssh/authorized_keys
```

Standby passwordless ssh check

```
sudo -u postgres ssh -l pgbackrest 192.168.121.192  
pgBackRest 2.51 - General help
```

Usage:

```
pgbackrest [options] [command]
```

Commands:

annotate	Add or modify backup annotation.
archive-get	Get a WAL segment from the archive.
archive-push	Push a WAL segment to the archive.
backup	Backup a database cluster.
...	
verify	Verify contents of the repository.
version	Get version.

Use 'pgbackrest help [command]' for more information.

Connection to 192.168.121.192 closed.

Standby pgBackRest configuration

```
[global]
log-level-console=info
log-level-file=detail
repo1-host=192.168.121.192
repo1-host-user=pgbackrest

[vgr_web]
pg1-path=/var/lib/pgsql/16/data
```

Restore standby from backup

```
sudo -u postgres pgbackrest --stanza=vgr_web --delta --type=standby restore
```

```
sudo -u postgres pgbackrest --stanza=vgr_web --delta --type=standby restore
2024-06-02 15:01:05.320 P00 INFO: restore command begin 2.51: --delta --exec-id=11381-5bbeb1d6
--log-level-console=info --log-level-file=detail --pg1-path=/var/lib/pgsql/16/data
--repo1-host=192.168.121.192 --repo1-host-user=pgbackrest --stanza=vgr_web --type=standby
2024-06-02 15:01:05.871 P00 INFO: repo1: restore backup set 20240602-143816F, recovery will start
at 2024-06-02 14:38:16
2024-06-02 15:01:05.884 P00 INFO: remove invalid files/links/paths from '/var/lib/pgsql/16/data'
2024-06-02 15:01:09.161 P00 INFO: write updated /var/lib/pgsql/16/data/postgresql.auto.conf
2024-06-02 15:01:09.182 P00 INFO: restore global/pg_control (performed last to ensure aborted
restores cannot be started)
2024-06-02 15:01:09.182 P00 INFO: restore size = 36.8MB, file total = 1576
2024-06-02 15:01:09.183 P00 INFO: restore command end: completed successfully (3867ms)
```

Restore standby from backup

```
sudo -u postgres cat /var/lib/pgsql/16/data/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
cluster_name = 'web128'
archive_mode = 'on'
archive_command = 'pgbackrest --stanza=vgr_web archive-push %p'
```

```
# Recovery settings generated by pgBackRest restore on 2024-06-02 15:01:09
restore_command = 'pgbackrest --stanza=vgr_web archive-get %f "%p"'
```

```
sudo systemctl start postgresql-16.service
```

```
cat <<EOF | sudo -iu postgres psql -v ON_ERROR_STOP=on
ALTER SYSTEM SET cluster_name = 'web129';
EOF
```

```
sudo systemctl restart postgresql-16.service
```

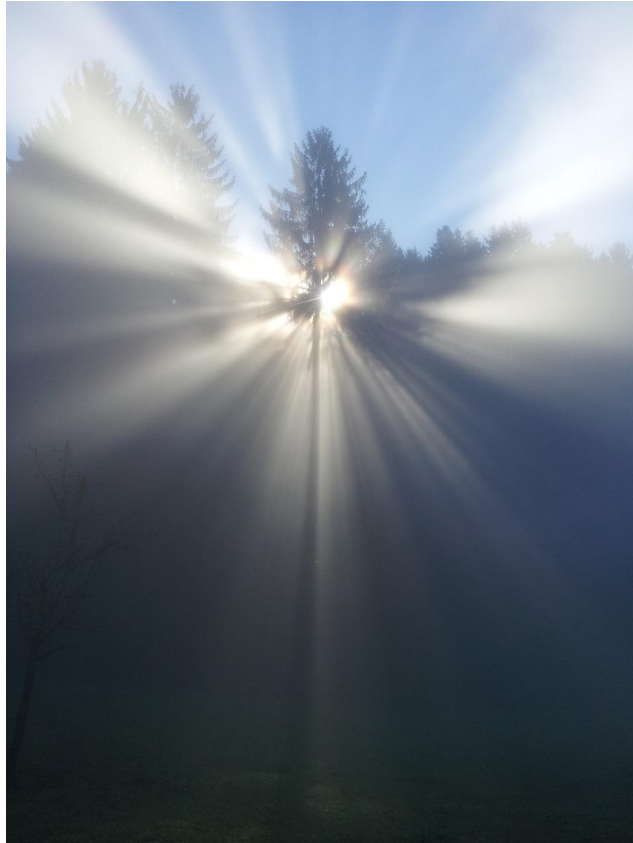

Restore standby from backup

```
[zelenya@pg-web-slave ~]$ sudo -u postgres psql -c "show cluster_name;"
cluster_name
-----
web129
(1 row)
```

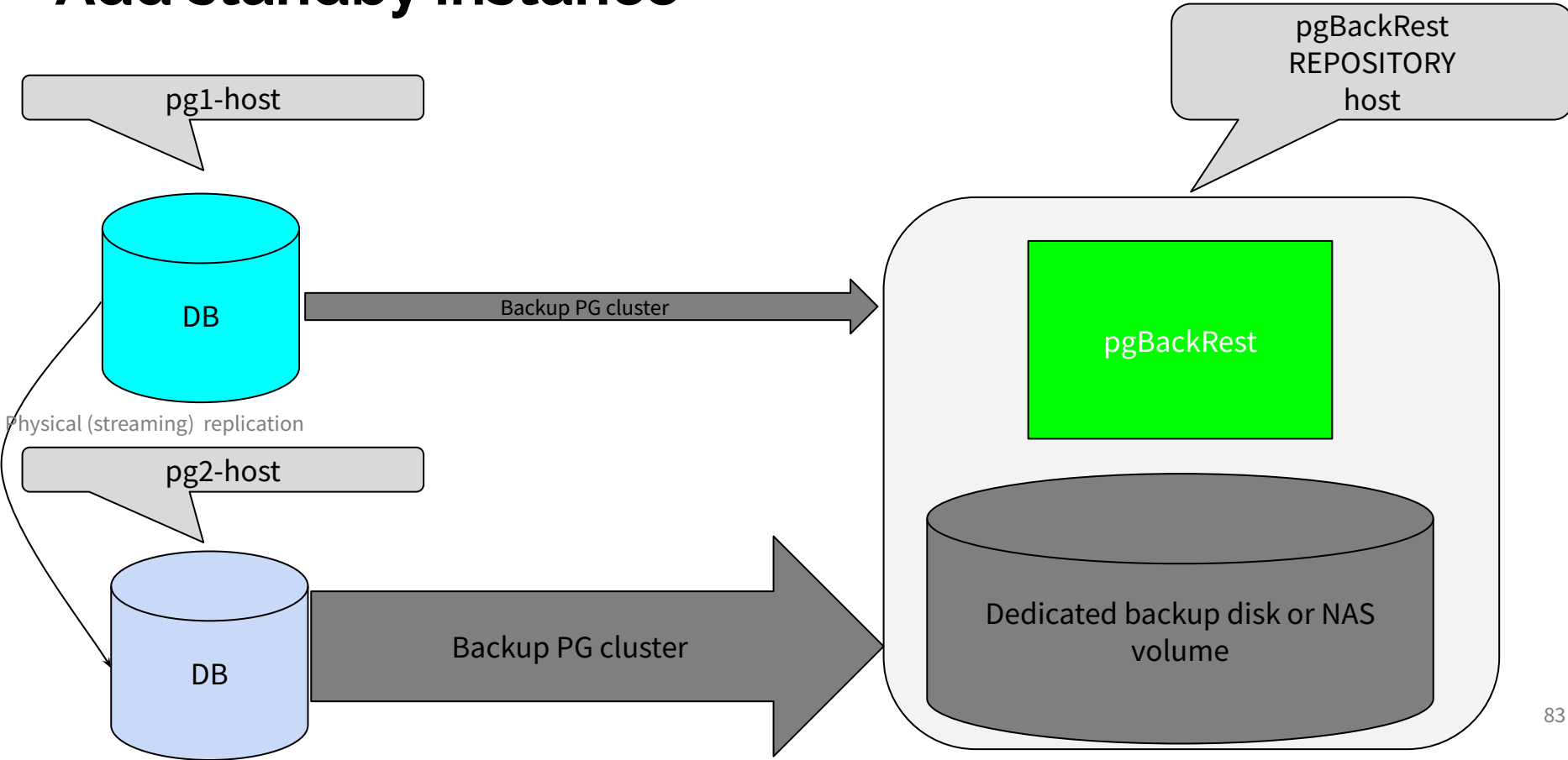
Configure streaming replication as next step, so far after the restore we have log shipping replica (look for the `archive_timeout` parameter)

```
[zelenya@pg-web-slave ~]$ sudo -u postgres psql -c "\l+"
                                List of databases
 Name | Owner  | Encoding | Locale Provider | Collate  | Ctype    | ICU Locale | ICU Rules | Access privileges | Size  | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8    | libc            | en_US.UTF-8 | en_US.UTF-8 |             |           | =c/postgres      | 7500 kB | pg_default | default administrative connection database
template0 | postgres | UTF8    | libc            | en_US.UTF-8 | en_US.UTF-8 |             |           | =c/postgres      | 7345 kB | pg_default | unmodifiable empty database
template1 | postgres | UTF8    | libc            | en_US.UTF-8 | en_US.UTF-8 |             |           | =c/postgres      | 7417 kB | pg_default | default template for new databases
web       | postgres | UTF8    | libc            | en_US.UTF-8 | en_US.UTF-8 |             |           | postgres=CTc/postgres | 7505 kB | pg_default |
web_old   | postgres | UTF8    | libc            | en_US.UTF-8 | en_US.UTF-8 |             |           | postgres=CTc/postgres | 7505 kB | pg_default |
(5 rows)
```

Can pgBackRest do more?

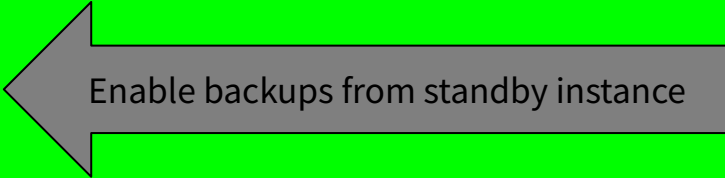


Add standby instance



Backup from standby - save primary IO

```
[global]
log-level-console=info
log-level-file=detail
repo1-path=/var/lib/pgbackrest
repo1-retention-full=2
start-fast=y
backup-standby=y
```



Enable backups from standby instance

```
[vgr_web]
pg1-host=192.168.121.128
pg1-host-user=postgres
pg1-path=/var/lib/pgsql/16/data
pg2-host=192.168.121.129
pg2-host-user=postgres
pg2-path=/var/lib/pgsql/16/data
```



Add pg2-host configuration

Backup from standby

```
sudo -u pgbackrest pgbackrest --log-level-console=detail --stanza=vgr_web backup
```

```
2024-06-02 15:22:08.339 P00 INFO: backup command begin 2.51: --backup-standby --exec-id=14893-6c040934 --log-level-console=detail --log-level-file=detail --pg1-host=192.168.121.128
--pg2-host=192.168.121.129 --pg1-host-user=postgres --pg2-host-user=postgres --pg1-path=/var/lib/pgsql/16/data --pg2-path=/var/lib/pgsql/16/data --repo1-path=/var/lib/pgbackrest
--repo1-retention-full=2 --stanza=vgr_web --start-fast
2024-06-02 15:22:10.285 P00 INFO: last backup label = 20240602-143816F_20240602-151848I, version = 2.51
2024-06-02 15:22:10.285 P00 INFO: execute non-exclusive backup start: backup begins after the requested immediate checkpoint completes
2024-06-02 15:22:10.994 P00 INFO: backup start archive = 000000010000000000000011, lsn = 0/11000028
2024-06-02 15:22:10.994 P00 INFO: wait for replay on the standby to reach 0/11000028
2024-06-02 15:22:41.677 P00 INFO: replay on the standby reached 0/11000028
2024-06-02 15:22:41.678 P00 INFO: check archive for prior segment 000000010000000000000010
2024-06-02 15:22:42.538 P01 DETAIL: backup file 192.168.121.128:/var/lib/pgsql/16/data/log/postgresql-Sun.log (15.9KB, 1.17%) checksum
f4e6b98bfbf4fb802efc2ef922f94f876d4950f7
2024-06-02 15:22:42.545 P02 DETAIL: backup file 192.168.121.129:/var/lib/pgsql/16/data/base/16400/1249 (464KB, 35.28%)
checksum ba9356768970072e125c00c877fe5e84e9506a2f
2024-06-02 15:22:42.547 P01 DETAIL: backup file 192.168.121.128:/var/lib/pgsql/16/data/global/pg_control (8KB, 35.87%)
checksum 9302a209ace4ca5d32a8226d3ec8e1579e758e21
2024-06-02 15:22:42.850 P01 DETAIL: match file from prior backup 192.168.121.128:/var/lib/pgsql/16/data/pg_logical/replorigin_checkpoint (8B, 35.87%) checksum 347fc8f2df71bd4436e38bd1516ccd7ea0d46532
2024-06-02 15:22:42.852 P02 DETAIL: backup file 192.168.121.129:/var/lib/pgsql/16/data/base/16400/2658 (128KB, 45.29%) checksum 875c4e6aac8f03d8f6cf05b9086adcd9f8c5fac3
2024-06-02 15:22:42.874 P02 DETAIL: backup file 192.168.121.129:/var/lib/pgsql/16/data/base/16400/1247 (120KB, 54.11%) checksum 2492c28a9cdd20cb72d6aed04f1ea5cc4c10f969
2024-06-02 15:22:42.886 P02 DETAIL: backup file 192.168.121.129:/var/lib/pgsql/16/data/base/16400/1259 (112KB, 62.35%) checksum fa9d45aac10a0400e4979338e9086a354c6f9cac2
...
2024-06-02 15:22:43.294 P00 DETAIL: reference pg_data/global/6302 to 20240602-143816F
2024-06-02 15:22:43.294 P00 DETAIL: reference pg_data/global/6303 to 20240602-143816F
2024-06-02 15:22:43.294 P00 DETAIL: reference pg_data/postgresql.conf to 20240602-143816F
2024-06-02 15:22:43.601 P00 INFO: execute non-exclusive backup stop and wait for all WAL segments to archive
2024-06-02 15:22:43.804 P00 INFO: backup stop archive = 000000010000000000000012, lsn = 0/12000088
2024-06-02 15:22:43.817 P00 DETAIL: wrote 'backup_label' file returned from backup stop function
2024-06-02 15:22:43.819 P00 INFO: check archive for segment(s) 000000010000000000000011:000000010000000000000012
2024-06-02 15:22:44.246 P00 INFO: new backup label = 20240602-143816F_20240602-152210I
2024-06-02 15:22:44.341 P00 INFO: incr backup size = 1.3MB, file total = 1577
2024-06-02 15:22:44.341 P00 INFO: backup command end: completed successfully (36004ms)
...
```

pgbackrest configuration, the automation way...

```
find /etc/pgbackrest/ -type f
/etc/pgbackrest/conf.d/vgr_web.conf
/etc/pgbackrest/pgbackrest.conf
```

```
sudo cat /etc/pgbackrest/pgbackrest.conf
[global]
log-level-console=info
log-level-file=detail
repo1-path=/var/lib/pgbackrest
repo1-retention-full=2
start-fast=y
backup-standby=y
```

```
sudo cat /etc/pgbackrest/conf.d/vgr_web.conf
[vgr_web]
pg1-host=192.168.121.128
pg1-host-user=postgres
pg1-path=/var/lib/pgsql/16/data
pg2-host=192.168.121.129
pg2-host-user=postgres
pg2-path=/var/lib/pgsql/16/data
```

Unplanned real life example...

- Monitoring system (thanks to json format and *pgbackrest prometheus exporter*) raised an backup alert
- The error:

```
incr backup: 20240525-200353F_20240528-025709I
  timestamp start/stop: 2024-05-28 02:57:09+02 / 2024-05-28 02:58:01+02
  wal start/stop: 0000000100000225000000C7 / 0000000100000225000000CA
  database size: 15GB, database backup size: 9.2GB
  repo1: backup size: 830.5MB
  backup reference list: 20240525-200353F,
20240525-200353F_20240526-025500I, 20240525-200353F_20240527-030728I
  error(s) detected during backup
```

Backup log file

...

2024-04-30 02:59:03.491 P02 DETAIL: backup file 10.27.0.231:/pgsql/cluster/15/data/global/pg_store_plans.stat (bundle 1/983840, 47.3KB, 99.89%) checksum 806b3de22bd3356d5d821edca88522230f1c8e2e

2024-04-30 02:59:03.491 P00 **WARN: page misalignment in file**

10.36.0.239:/pgsql/cluster/15/data/global/pg_store_plans.stat: file size 48473 is not divisible by page size 8192

2024-04-30 02:59:03.491 P02 DETAIL: backup file 10.27.0.231:/pgsql/cluster/15/data/postgresql.conf (bundle 1/993582, 30.3KB, 99.89%) checksum d58a91daab0100db1240ce5efdd35b8773a1be10

2024-04-30 02:59:03.501 P00 DETAIL: reference pg_data/postgresql.conf.218896.2024-03-12@16:28:12~ to 20240427-193931F

2024-04-30 02:59:03.501 P00 DETAIL: reference pg_data/postgresql.conf.685914.2024-01-25@22:59:45~ to 20240427-193931F

2024-04-30 02:59:03.608 P00 INFO: execute non-exclusive backup stop and wait for all WAL segments to archive

2024-04-30 02:59:03.822 P00 INFO: backup stop archive = 00000001000001880000008F, lsn = 188/8F143C78

2024-04-30 02:59:03.823 P00 DETAIL: wrote 'backup_label' file returned from backup stop function

2024-04-30 02:59:03.829 P00 INFO: check archive for segment(s) 00000001000001880000008F:00000001000001880000008F

2024-04-30 02:59:04.580 P00 DETAIL: copy segment 00000001000001880000008F to backup

2024-04-30 02:59:04.688 P00 INFO: new backup label = 20240427-193931F_20240428-023341I

2024-04-30 02:59:04.781 P00 INFO: incr backup size = 14GB, file total = 1989

2024-04-30 02:59:04.781 P00 INFO: **backup command end: completed successfully (86033ms)**

Solution

- the abandoned file was a residuum after an extension testing in the UAT environment
 - file was deleted, since the extension was already uninstalled
- subsequent backup was clean

```
full backup: 20240528-140911F
timestamp start/stop: 2024-05-28 14:09:11+02 / 2024-05-28
14:11:14+02
wal start/stop: 000000010000022800000020 /
000000010000022800000020
database size: 15.0GB, database backup size: 15.0GB
repo1: backup size: 4.0GB
```

- pgBackRest is very good at protecting our data including various validations

This is the end

