

pg_basebackup 実行手順例

※前提として PostgreSQL はインストール済みとします

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|--|---|
| 1 | WAL のアーカイブを保存するディレクトリを作成する | # mkdir /usr/local/pgsql/archive | |
| 2 | 1 で作成したディレクトリのオーナーを postgres に変更する | # chown postgres:postgres /usr/local/pgsql/archive | |
| 3 | バックアップ対象サーバの postgresql.conf を設定する | listen_addresses = '*' wal_level = hot_standby archive_mode = on archive_command = 'cp -i %p /usr/local/pgsql/archive' max_wal_senders = 3 | |
| 4 | バックアップ対象サーバの pg_hba.conf を設定する | # TYPE DATABASE USER ADDRESS METHOD host replication test 10.1.12.12/24 password | |
| 5 | バックアップ対象サーバの PostgreSQL を再起動する | \$ pg_ctl restart | |
| 6 | pg_basebackup コマンドを介して DB に接続するユーザに replication 権限を付与する | postgres=# alter role test with replication ; | psql 接続してメタコマンドで確認する postgres=# \du -[RECORD3]----- Role name test Attributes Replication Member of {} |
| 7 | バックアップサーバで pg_basebackup を実行し、バックアップを取得する | \$ pg_basebackup -h 10.1.12.11 -p 5432 -U test -D /tmp/backup -x -v (※) | 以下のメッセージが出力される pg_basebackup: base backup completed |
| 8 | バックアップが取得できているか確認する | -D で指定したディレクトリ内にデータベースクラスタ配下のファイルが存在するか確認する \$ ls -l /tmp/backup | |

※手順 7 で使用したオプション

- -h/--host : バックアップ対象サーバのホスト名
- -p/--port : バックアップ対象の PostgreSQL が稼働しているポート
- -U/--username : 接続ユーザ名
- -D/--pgdata : データベースクラスタのコピーの保存先
- -x/--xlog : WAL ファイルをバックアップに含める
- -v/--verbose : 詳細を出力する

pg_rman 導入手順例

※前提として PostgreSQL はインストール済みとします

※pg_rman は rpm からインストールするものとします

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|--|---|
| 1 | rpm パッケージを入手する | 公開先: https://github.com/osscc-db/pg_rman | |
| 2 | rpm パッケージをサーバにインストールする | # rpm -ivh pg_rman-1.3.0-1.pg94.rhel6.x86_64.rpm | 以下のように確認できる # rpm -qa grep pg_rman pg_rman-1.3.0-1.pg94.rhel6.x86_64 |
| 3 | postgres ユーザになり pg_rman のバックアップカタログを初期化する | \$ pg_rman init -B /usr/local/pgsql/pg_rman -D /usr/local/pgsql/data/ -A /usr/local/pgsql/archive/ (※) | -B オプションで指定したディレクトリ内に次のファイルが作成されているか確認する • backup • pg_rman.ini • timeline_history |
| 4 | pg_rman.ini にリテンションポリシーを記述する | ARCLOG_PATH=' /usr/local/pgsql/archive/' SRVLOG_PATH=' /usr/local/pgsql/data/pg_l | |

| | | |
|--|--|--|
| | <pre> og' BACKUP_MODE = full COMPRESS_DATA = YES KEEP_ARCLOG_FILES = 10 KEEP_ARCLOG_DAYS = 10 KEEP_DATA_GENERATIONS = 3 KEEP_DATA_DAYS = 120 KEEP_SRVLOG_FILES = 10 KEEP_SRVLOG_DAYS = 10 </pre> | |
|--|--|--|

※手順 3 で使用したオプション

- B/--backup-path : バックアップカATALOGの保存先
- D/--pgdata : データベースクラスタ
- A/--arclog-path : アーカイブ WAL 保存先のディレクトリ

pg_rman 実行手順例

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|-----------------------------|--------------------------------------|---|
| 1 | pg_rman コマンドを実行しバックアップを取得する | \$ pg_rman backup --backup-mode=full | 以下のメッセージが出力される INFO: backup complete |
| 2 | バックアップが取得されたか確認する | \$ pg_rman show | |
| 3 | バックアップを validate して有効にする | \$ pg_rman validate | 以下のメッセージが出力される INFO: validate: "2016-02-25 11:17:35" backup and archive log files by CRC INFO: backup "2016-02-25 11:17:35" is valid |

Barman 導入手順例

※前提として PostgreSQL はインストール済みとします

※Barman は rpm からインストールするものとします

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|--|---|---|
| 1 | Barman の rpm パッケージを取得するため、サーバに PGDG(PostgreSQL Global Development Group)リポジトリが登録されているか確認する | # ls /etc/yum.repos.d/ | 以下のように確認できる # ls /etc/yum.repos.d/ pgdg-94-centos.repo |
| 2 | 依存関係のある Python のモジュールがインストールされているか確認する | # rpm -qa grep python | 以下のパッケージがインストールされているか確認する •python-psycopg2-2.6.1-1.rhel6.x86_64 •python-argcomplete-0.3.7-1.rhel6.noarch •python-2.6.6-64.el6.x86_64 •python-pycurl-7.19.0-8.el6.x86_64 •python-iniparse-0.3.1-2.1.el6.noarch •python-argparse-1.2.1-2.1.el6.noarch •python-dateutil-1.4.1-6.el6.noarch •python-argh-0.23.0-1.rhel6.noarch •python-libs-2.6.6-64.el6.x86_64 •python-urlgrabber-3.9.1-9.el6.noarch •rpm-python-4.8.0-47.el6.x86_64 •newt-python-0.52.11-3.el6.x86_64 |
| 3 | Barman をインストールする | # yum install barman | |
| 4 | Barman が正常にインストールされたか確認する | # rpm -qa grep barman | 以下のように確認できる # rpm -qa grep barman barman-1.5.1-1.rhel6.noarch |
| 5 | barman.conf の Barman に関する項目を設定する | <pre> barman_home = /var/lib/barman barman_lock_directory = /var/run/barman barman_user = barman </pre> | |

| | | | |
|----|--|---|--|
| | | <pre>log_file = /var/log/barman/barman.log compression = gzip reuse_backup = copy configuration_files_directory = /etc/barman.d</pre> | |
| 6 | barman.conf のバックアップ対象サーバに関する項目を設定する (barman.conf 内に DB 接続のパスワードを記述したくない場合パスワードファイルを使用することも可能) | <pre>[postgres01] description = "PostgreSQL01 Database" ssh_command = ssh postgres@10.1.12.11 conninfo = host=10.1.12.11 user=barman password=barman minimum_redundancy = 1 retention_policy = REDUNDANCY 2 retention_policy = RECOVERY WINDOW OF 4 WEEKS</pre> | |
| 7 | 手順 6 で barman.conf に設定したサーバが Barman のバックアップ対象として正しく認識されているか確認する | \$ barman list-server | 下記のように対象サーバが出力される <pre>\$ barman list-server postgres01 - PostgreSQL01 Database postgres02 - PostgreSQL02 Database postgres03 - PostgreSQL03 slave Database</pre> |
| 8 | 手順 6 で登録した ssh 接続ユーザを、バックアップ対象の PostgreSQL の pg_hba.conf に登録する | <pre># TYPE DATABASE USER ADDRESS METHOD host all barman 10.1.12.20/24 password</pre> | |
| 9 | バックアップ対象 PostgreSQL を再起動する | \$ pg_ctl restart | |
| 10 | Barman 用のエクステンションツールの pgespresso を取得する | 公開先: https://github.com/2ndquadrant-it/pgespresso | |
| 11 | バックアップ対象のサーバにインストールする | <pre>\$ unzip pgespresso-master.zip \$ make && make install</pre> | make、make install が正常に終了するか |
| 12 | バックアップ対象 PostgreSQL にエクステンションとして取り込む | <pre>\$ psql postgres=# create extension pgespresso; CREATE EXTENSION</pre> | 正常に登録されているかメタコマンドから確認する <pre>postgres=# \dx List of installed extensions -[RECORD1]----- Name pgespresso Version 1.0 Schema public Description pgespresso extension for Barman backups</pre> |
| 13 | Barman からバックアップを取得できる状態であるか、バックアップ対象サーバのステータスを確認する | \$ barman check postgres01 | 以下のように全項目 OK ならバックアップが実行できる <pre>Server postgres01: PostgreSQL: OK archive_mode: OK wal_level: OK archive_command: OK continuous archiving: OK directories: OK retention policy settings: OK backup maximum age: OK (no last_backup_maximum_age provided) compression settings: OK minimum redundancy requirements: OK (have 2 backups, expected at least 1) ssh: OK (PostgreSQL server) pgespresso extension: OK</pre> |

Barman 実行手順例

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|---|--|
| 1 | Barman をインストールしたサーバで barman コマンドを実行しバックアップを取得する | \$ barman backup postgres01 | 以下のようにメッセージが出力される Starting backup for server postgres01 in /var/lib/barman/postgres01/base/20160308T185634 Backup start at xlog location: 0/12000028 (0000000100000000000000012, 00000028) Copying files. Copy done. Asking PostgreSQL server to finalize the backup. Backup size: 26.9 MiB Backup end at xlog location: 0/13000000 (0000000100000000000000012, 00000000) Backup completed |
| 2 | バックアップファイルが存在するか確認する | \$ ls -l /var/lib/barman/postgres01/base/ | 1 のメッセージに出力されたファイルが存在する \$ ls /var/lib/barman/postgres01/base/20160308T185634 |

OmniPITR 導入手順例

※前提として PostgreSQL はインストール済みとします

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|--|--|
| 1 | 依存関係のある perl のモジュールがインストールされているか確認する | # rpm -qa grep perl | 以下のパッケージがインストールされているか確認する ・perl-Module-Pluggable-3.90-141.el6_7.1.x86_64 ・perl-libs-5.10.1-141.el6_7.1.x86_64 ・perl-5.10.1-141.el6_7.1.x86_64 ・perl-Pod-Escapes-1.04-141.el6_7.1.x86_64 ・perl-version-0.77-141.el6_7.1.x86_64 ・perl-Pod-Simple-3.13-141.el6_7.1.x86_64 ・perl-Time-HiRes-1.9721-141.el6_7.1.x86_64 |
| 2 | OmniPITR を取得する(zip 形式) | 公開先: https://github.com/omniti-labs/omnipitr | |
| 3 | 取得した OmniPITR ファイルをサーバ上で解凍する | # unzip omnipitr-master.zip | 以下のファイルが確認できる ・META.json ・README.pod ・bin ・doc ・lib ・t ・test |
| 4 | 解凍したファイルのオーナーを postgres に変更する | # chown postgres:postgres -R omnipitr-master | 下記コマンドで確認可能 # ls -l |
| 5 | OmniPITR のスクリプトが正常に実行できるか sanity-check.sh を実行して確認する | \$./sanity-check.sh | 以下のメッセージが出力されれば正常に実行できる Checking: - /home/postgres/omnipitr-master/bin - /home/postgres/omnipitr-master/lib 9 programs, 31 libraries. Tar version All checked, and looks ok. |

OmniPITR 実行手順例

全スクリプト共通の事前準備

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|------------------------------------|--|------|
| 1 | omnipitr-*スクリプト実行ログの格納先ディレクトリを作成する | # mkdir /var/log/omnipitr/ | |
| 2 | 1で作成したディレクトリのオーナーをpostgresに変更する | # chown postgres:postgres /var/log/omnipitr/ | |

omnipitr-archive の実行手順

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|---|---|
| 1 | アーカイブ WAL の保存先ディレクトリを作成する | # mkdir /usr/local/pgsql/omni-wal-archive | |
| 2 | 1で作成したディレクトリのオーナーをpostgresに変更する | # chown postgres:postgres /usr/local/pgsql/omni-wal-archive | |
| 3 | postgresql.conf の archive_command にスクリプトを設定する | archive_command = '/home/postgres/omnipitr-master/bin/omnipitr-archive -D /usr/local/pgsql/data -l "/var/log/omnipitr/omnipitr-^Y-^m-^d.log" -dl /usr/local/pgsql/omni-wal-archive -db /usr/local/pgsql/omni-dst-backup "%p"' (※) | |
| 4 | PostgreSQL を再起動する | \$ pg_ctl restart | |
| 5 | WAL のアーカイブが正常に行われているか確認する | 3 の -dl オプションで指定したディレクトリ内に WAL ファイルが存在するか確認する | \$ ls -l /usr/local/pgsql/omni-wal-archive/ |

※手順 1 で使用したオプション

- -D/--data-dir : PostgreSQL のデータディレクトリ
- -l/--log : omnipitr-archive の実行ログの出力先
- -dl/--dst-local : ローカル内での WAL のアーカイブ先(リモートを指定する場合は -dr/--dst-remote を使用する)
- -db/--dst-backup : バックアップ取得中の WAL の保存先(omnipitr-master-backup を使用しない場合は不要)

omnipitr-restore の実行手順

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|--|--|---|
| 1 | バックアップサーバの recovery.conf の restore_command にスクリプトを設定する | restore_command = '/home/postgres/omnipitr-master/bin/omnipitr-restore -l "/var/log/omnipitr/restore-^Y-^m-^d.log" -s /usr/local/pgsql/omni-wal-archive/ -v %f %p' (※) | |
| 2 | PostgreSQL を起動する | \$ pg_ctl start | |
| 3 | WAL が正常にリストアされているか確認する | -l オプションで指定したログの出力内容を確認する。 | 以下のメッセージが出力されていたら正常に WAL がリストアされている LOG : Called with parameters: -l /var/log/omnipitr/restore-^Y-^m-^d.log -s /usr/local/pgsql/omni-wal-archive/ -v 000000010000000000000002B pg_xlog/RECOVERYXLOG LOG : Timer [Copying segment 000000010000000000000002B to ./pg_xlog/RECOVERYXLOG] took: 0.050s LOG : Segment 000000010000000000000002B restored |

※手順 1 で使用したオプション

- -l/--log : omnipitr-restore の実行ログの出力先
- -s/--source : WAL セグメントをロードしてくる場所(※omnipitr-archive の-dr オプションで指した場所と同じ場所)
- -v/--verbose : 詳細を出力する

omnipitr-backup-master の実行手順

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|---|--|
| 1 | バックアップファイルの保存先ディレクトリを作成する | # mkdir /usr/local/pgsql/omni-backup-master/ | |
| 2 | 1 で作成したディレクトリのオーナーを postgres に変更する | # chown postgres:postgres /usr/local/pgsql/omni-backup-master/ | |
| 3 | omnipitr-master-backup スクリプトを実行しバックアップを取得する | \$ /home/postgres/omnipitr-master/bin/omnipitr-backup-master -D /usr/local/pgsql/data/ -l /var/log/omnipitr/backup.log -dl /usr/local/pgsql/omni-backup-master/ -x /usr/local/pgsql/omni-backup-master/omni-dst-backup.xlogs -v (※) | スクリプトが正常終了すると-l で指定したログに以下のメッセージが出力される LOG : Timer [Compressing xlogs] took: 0.281s LOG : Timer [Making xlog archive] took: 0.499s LOG : Timer [Delivering to all remote destinations] took: 0.000s LOG : Timer [Delivering meta files] took: 0.000s LOG : Timer [Whole backup procedure] took: 109.919s LOG : All done. |
| 4 | バックアップが取得できているか確認する | -dl で指定したディレクトリ内に tar ファイルが存在するか確認する \$ ls -l /usr/local/pgsql/omni-backup-master/ | 以下のファイルが確認できる ・[ホスト名]-data-2016-02-05.tar ・[ホスト名]-meta-2016-02-05.tar ・[ホスト名]-xlog-2016-02-05.tar |

※手順 3 で使用したオプション

- -D/--data-dir : PostgreSQL のデータディレクトリ
- -l/--log : omnipitr-backup-master の実行ログの出力先
- -dl/--dst-local : ローカル内でのバックアップの保存先(リモートを指定する場合は-dr/--remoteを使用する)
- -x/--xlogs : バックアップ取得中の WAL の保存先(※バックアップ取得前に存在していないこと)
- -v/--verbose : 詳細を出力する

omnipitr-backup-slave の実行手順

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|--|--|
| 1 | バックアップファイルの保存先ディレクトリを作成する | # mkdir /usr/local/pgsql/omni-backup-slave/ | |
| 2 | 1 で作成したディレクトリのオーナーを postgres に変更する | # chown postgres:postgres /usr/local/pgsql/omni-backup-slave/ | |
| 3 | omnipitr-slave-backup スクリプトを実行し、ストリーミングレプリケーションのスタンバイサーバからバックアップを取得する | \$ /home/postgres/omnipitr-master/bin/omnipitr-backup-slave -D /usr/local/pgsql/data/ -l /var/log/omnipitr/slavebackup.log -dl /usr/local/pgsql/omnipitr-backup-slave/ -s /usr/local/pgsql/data/pg_xlog -v (※) | スクリプトが正常終了すると-l で指定したログに以下のメッセージが出力される LOG : Timer [Compressing xlogs] took: 1.452s LOG : Timer [Making xlog archive] took: 66.569s LOG : Timer [Delivering to all remote destinations] took: 0.000s LOG : Timer [Delivering meta files] took: 0.000s LOG : Timer [Whole backup procedure] took: 68.257s LOG : All done. |
| 4 | バックアップが取得できているか確認する | -dl で指定したディレクトリ内に tar ファイルが存在するか確認する \$ ls -l /usr/local/pgsql/omnipitr-backup-slave/ | 以下のファイルが確認できる ・[ホスト名]-data-2016-02-05.tar ・[ホスト名]-meta-2016-02-05.tar ・[ホスト名]-xlog-2016-02-05.tar |

※手順 3 で使用したオプション

- D/--data-dir : PostgreSQL のデータディレクトリ
- l/--log : omnipitr-backup-slave の実行ログの出力先
- dl/--dst-local : ローカル内でのバックアップの保存先(リモートを指定する場合は-dr/--remoteを使用する)
- s/--source : 参照する WAL アーカイブディレクトリ(omnipitr-restore で指定したディレクトリと同じ)
- v/--verbose : 詳細を出力する

omnipitr-synch の実行手順

| No | 手順内容 | 作業内容・実行コマンド・設定パラメータ | 確認方法 |
|----|---|---|---|
| 1 | omnipitr-synch スクリプトを実行し、データベースクラスタのバックアップを取得する | \$ /home/postgres/omnipitr-master/bin/omnipitr-synch -o postgres@10.1.12.20:/tmp/backup (※) | <p>以下のようなメッセージが出力される</p> <p>Source directory: /usr/local/pgsql/data</p> <p>Outputs:</p> <ul style="list-style-type: none"> - directory /tmp/backup on host 10.1.12.20, logged as postgres <p>All data in output directories will be overwritten</p> <p>Are you sure you want to continue? (enter: YES): YES</p> <p>LOG : pg_start_backup('omnipitr') returned 1/B000028.</p> <p>LOG : Started [Sending /usr/local/pgsql/data] with pid: 14209</p> <p>postgres@10.1.12.20's password:</p> <p>LOG : Job 14209 finished with status 0.</p> <p>LOG : pg_stop_backup() returned 1/B0000B8.</p> <p>LOG : All done.</p> |
| 2 | バックアップが取得できているか確認する | <p>-o で指定したディレクトリ内にデータベースクラスタ配下のファイルが存在するか確認する</p> <p>\$ ls -l /tmp/backup</p> | |

※手順 1 で使用したオプション

- o/--output : データベースクラスタのコピーの保存先