

ストリーミングレプリケーション

運用のための機能 技術を知る

- | | | | | |
|--|-----------------------------|--------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> 導入／環境設定 | <input type="checkbox"/> 移行 | <input type="checkbox"/> 性能 | <input type="checkbox"/> チューニング | <input type="checkbox"/> バックアップ／リカバリー |
| <input checked="" type="checkbox"/> 冗長化／負荷分散 | <input type="checkbox"/> 監視 | <input type="checkbox"/> データ連携 | <input type="checkbox"/> 災害対策 | <input type="checkbox"/> 豆知識 |

ストリーミングレプリケーションの仕組みと構成のポイントについては、「ストリーミングレプリケーション～仕組み、構成のポイント～」で説明しました。

ここでは、ストリーミングレプリケーションのフェイルオーバーとリカバリーのポイントについて説明します。なお、本資料では、PostgreSQL 11 で設定可能なパラメーターおよび機能を使用して説明しています。

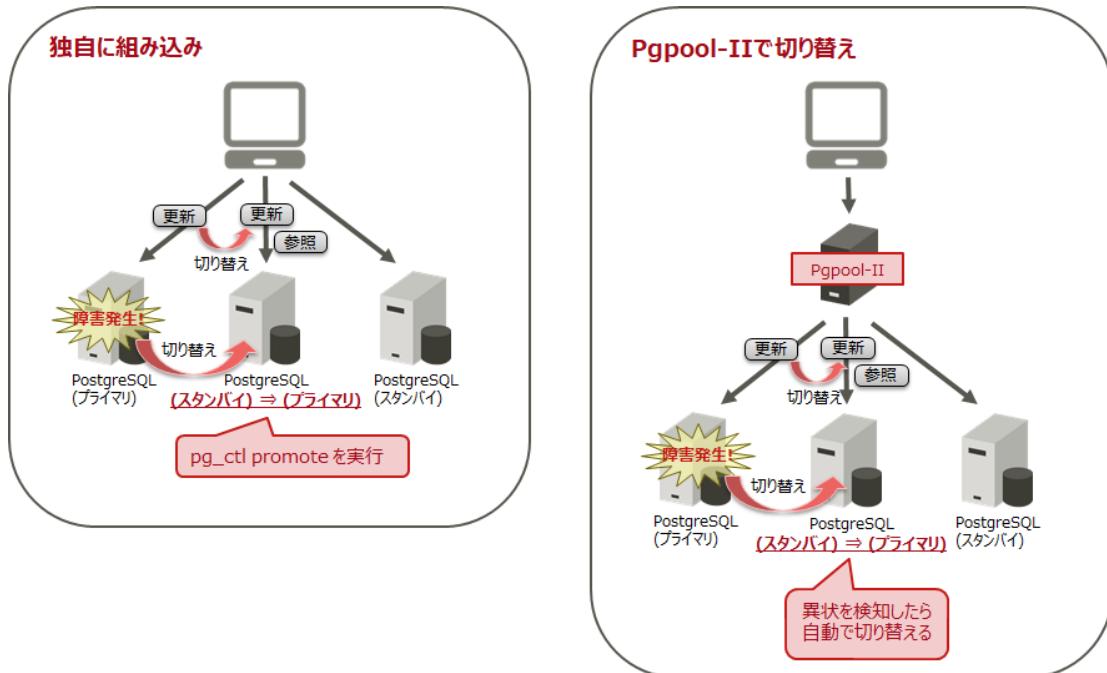
ポイント

PostgreSQL 12 では、設定ファイルの `recovery.conf` に指定するパラメーターが、`postgresql.conf` に統合されました。PostgreSQL 12 を使用する場合、`recovery.conf` が存在すると PostgreSQL サーバーが起動しないなどの仕様変更があります。本資料の `recovery.conf` に関する記事は、別途仕様を確認してください。仕様の詳細については、“PostgreSQL Documentation”の“Release Notes”的“Migration to Version 12”を参照してください。

1. フェイルオーバー

プライマリーサーバーに障害が発生した場合、スタンバイサーバーに切り替えることで、業務を継続することができます。

PostgreSQL には異常を検知する機能や自動的な切り替え機能は存在しないため、異常を検知する仕組みを組み込み、さらに、スタンバイサーバーを昇格 (`pg_ctl promote`) させるスクリプトを組み込んでおく必要があります。または、異常検知や自動切り替え機能を備えた、Pgpool-II などのクラスタソフトウェアを利用してください。



2. リカバリー

サーバーダウン後の対処は、プライマリーサーバーがダウンした場合と、スタンバイサーバーがダウンした場合で異なります。ここでは、同期レプリケーション構成の場合を例に、それぞれの対処方法について説明します。

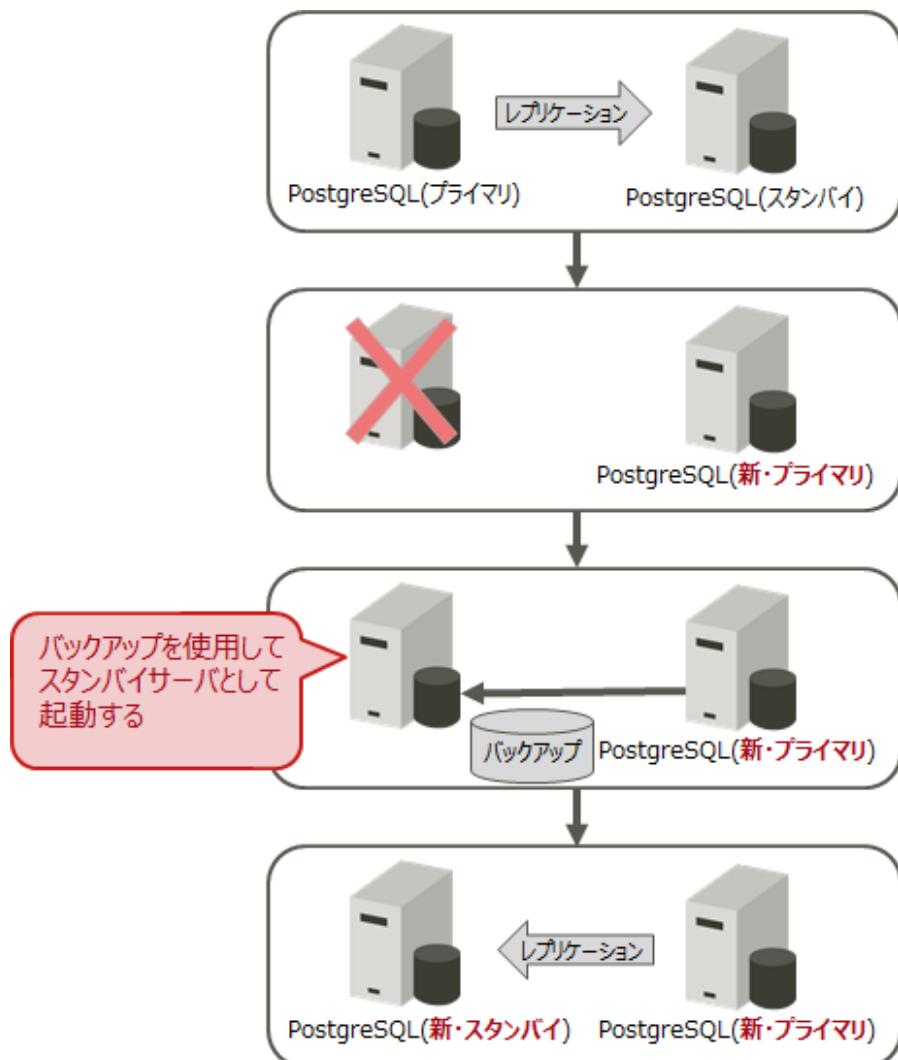
2.1 プライマリーサーバーがダウンした場合

プライマリーサーバーがダウンした後、ダウンしたサーバー（旧プライマリーサーバー）をスタンバイサーバーとして再構築する必要があります。旧プライマリーサーバーを復旧する方法として様々ありますが、以下の代表的な2つの方法を紹介します。

2.1.1 ベースバックアップを利用する方法 (pg_basebackup コマンド)

新プライマリーサーバーから最新のデータベースクラスタをコピー (pg_basebackup コマンド) し、スタンバイサーバーとして必要な設定を行った上で起動することで、ダウンした旧プライマリーサーバーを、新しいスタンバイサーバーとして構築することができます。データベースが大きい場合は、再構築までに時間がかかります。

リカバリーの概要



リカバリーの手順

1. 障害原因の特定と対処を行い、ダウンした旧プライマリーサーバーを復旧します。
2. 旧プライマリーサーバーで、旧プライマリーサーバーのデータベースクラスタを削除します。
3. 旧プライマリーサーバーで、`pg_basebackup` コマンド (-R オプション) を実行し、新プライマリーサーバーから物理バックアップを取得します。
4. 旧プライマリーサーバーで、`recovery.conf` を修正します。
 - `standby_mode=on` を設定
 - `primary_conninfo` に `application_name` を追加
5. 旧プライマリーサーバーで、`postgresql.conf` を修正します。
 - `synchronous_standby_names` の無効化
6. 旧プライマリーサーバーを起動し、新スタンバイサーバーにします。
7. 新プライマリーサーバーで、`postgresql.conf` を修正します。
 - `synchronous_standby_names` に同期レプリケーションを行うスタンバイサーバー名を設定し、リロードします。

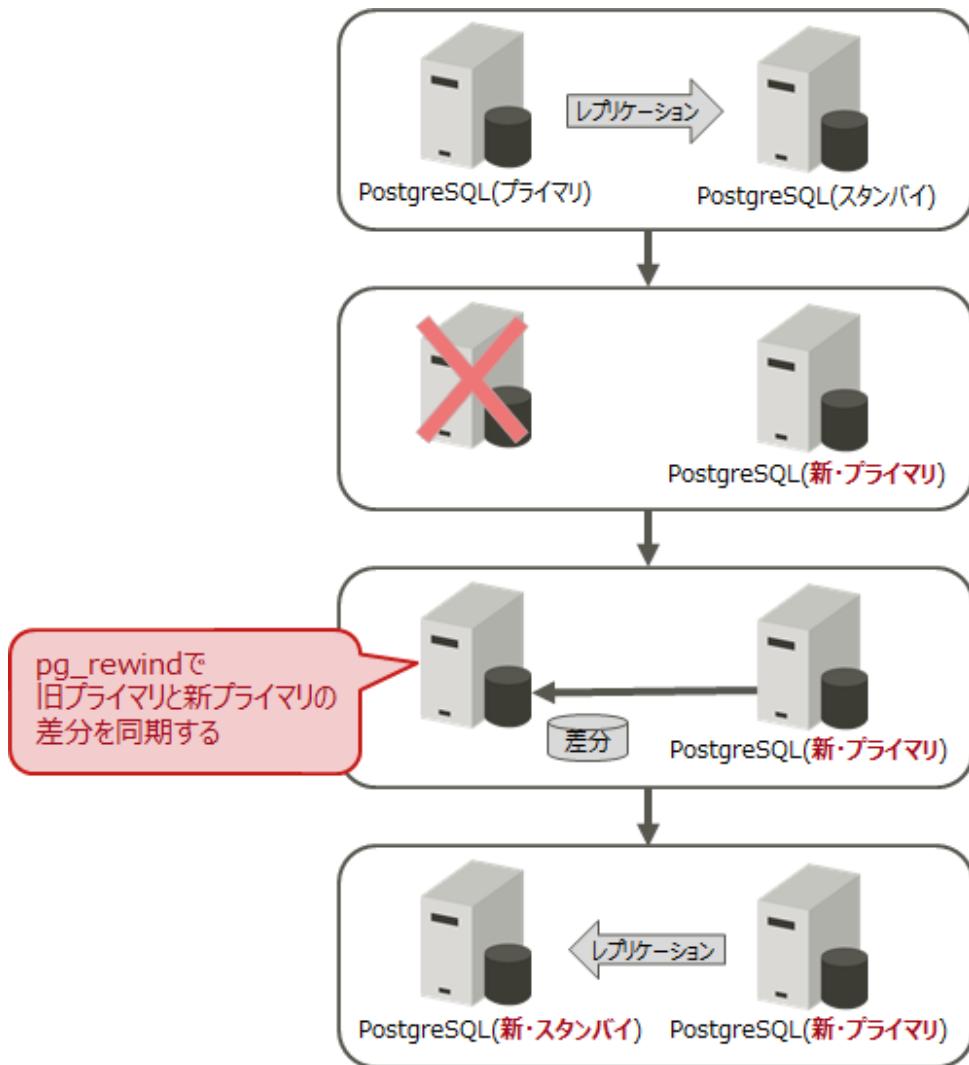
2.1.2 ダウンしたクラスタファイルを利用する方法（`pg_rewind` コマンド）

旧プライマリーサーバーのデータベースにハードウェアを起因とする障害が発生しておらず、データベースを再起動することができる場合は、ダウンしたクラスタファイルを利用して復旧することができます。

旧プライマリーサーバーと新プライマリーサーバーの差分を同期（`pg_rewind` コマンド）させ、スタンバイサーバーとして必要な設定を行った上で起動することで、ダウンした旧プライマリーサーバーを新しいスタンバイサーバーとして構築することができます。`pg_basebackup` コマンドと比べ、`pg_rewind` コマンドは比較的短時間で再構築することができます。ただし、`pg_rewind` コマンドを使用するには、以下の条件があるので注意が必要です。

- 旧プライマリーサーバーが正常停止している
- `postgresql.conf` の `wal_log_hints` パラメーターが有効または `initdb` でデータベースクラスタを初期化する時にデータチェックサムが有効

リカバリーの概要



リカバリーの手順

- 障害原因の特定と対処を行い、ダウンした旧プライマリーサーバーを復旧します。
- 旧プライマリーサーバーは正常停止している必要があるため、停止した旧プライマリーサーバーを一旦起動したあと、正常停止させます。
- 旧プライマリーサーバーで、`pg_rewind` コマンドを実行します。
- 旧プライマリーサーバーで、`recovery.conf` を作成します。
 - `standby_mode=on` を設定
 - `recovery_target_timeline=latest` を指定
 - `primary_conninfo [application_name]` を設定
- 旧プライマリーサーバーで、`postgresql.conf` を修正します。
 - `synchronous_standby_names` の無効化
- 旧プライマリーサーバーを起動し、新スタンバイサーバーにします。
- 新プライマリーサーバーで、`postgresql.conf` を修正します。

- `synchronous_standby_names` に同期レプリケーションを行うスタンバイサーバー名を設定し、リロードします。
8. 新スタンバイサーバーと新プライマリーサーバーのタイムライン ID が揃っていることを確認します。タイムライン ID は、`pg_controldata` コマンドで取得できます。

参考

PostgreSQL 14 以降では、`pg_rewind` コマンド実行時のソースとしてスタンバイサーバーが指定可能になりました。これによりスタンバイサーバーから旧プライマリーサーバーに差分を同期できるようになるため、3 台以上の構成でのリカバリーに役立ちます。

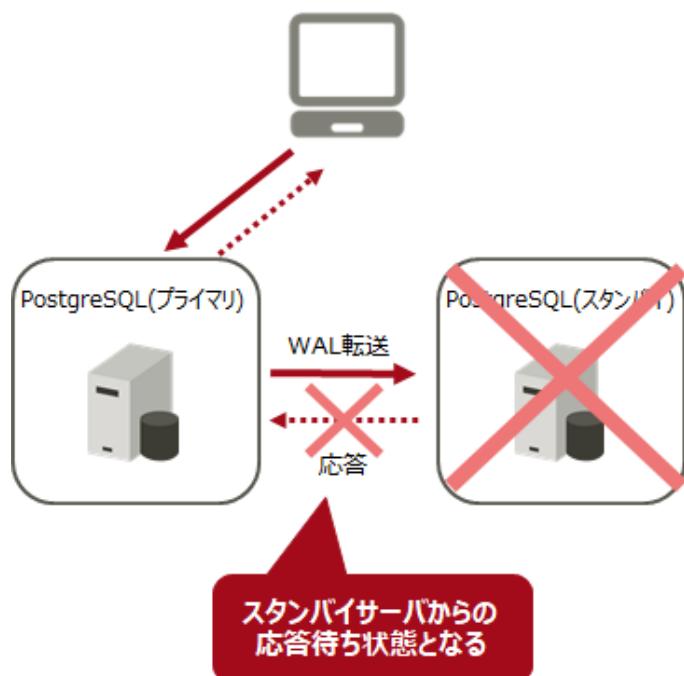
2.2 スタンバイサーバーがダウンした場合

スタンバイサーバーがダウンした場合の対処は、スタンバイサーバーが同期レプリケーションなのか、非同期レプリケーションなのかで異なります。それぞれの対処方法について説明します。

2.2.1 同期レプリケーションの場合

同期レプリケーションとして動作中のスタンバイサーバーがダウンすると、プライマリーサーバーは、スタンバイサーバーからの応答待ちの状態となり、クライアントに COMMIT の応答を返すことができなくなります。そのため、異常があった同期レプリケーションのスタンバイサーバーを切り離す（非同期レプリケーションにする）必要があります。プライマリーサーバーの `postgresql.conf` の `synchronous_standby_names` パラメーターから異常のあったスタンバイサーバー名を削除し、ファイルの再読み込み (`pg_ctl reload`) を行ってください。すべてのスタンバイサーバーが停止した場合は、`synchronous_standby_names` パラメーターを空にしてから再読み込みを行ってください。

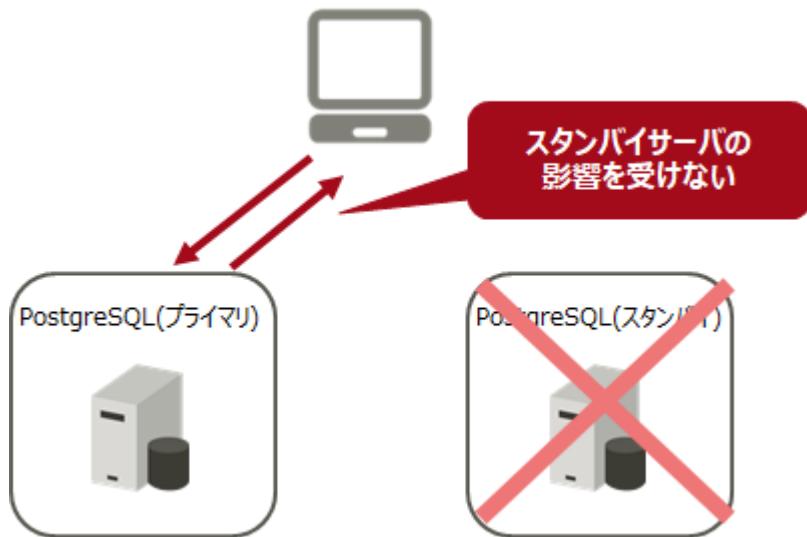
スタンバイサーバーの復旧は、「プライマリーサーバーがダウンした場合」と同様の手順です。



2.2.2 非同期レプリケーションの場合

非同期レプリケーションであるため、スタンバイサーバーがダウンした際に、`postgresql.conf` の編集やファイルの再読み込みを行う必要はありません。

スタンバイサーバーの復旧は、「プライマリーサーバーがダウンした場合」と同様の手順です。



PostgreSQL のストリーミングレプリケーションの運用について解説しました。ストリーミングレプリケーション機能を活用し、データベースの高可用化にお役立てください。

FUJITSU Software Enterprise Postgres では、異常検知や自動フェイルオーバーなど、実際の業務に必要な機能を備えた「データベース多重化機能」があります。

ぜひ、FUJITSU Software Enterprise Postgres のご利用もご検討ください。

2022 年 3 月 25 日