

# CLUSTERPRO<sup>®</sup> X *for Linux*

PPガイド

(PostgreSQL レプリケーション)

2020.4.17  
第01版

**CLUSTERPRO**

改版履歴

版数	改版日付	内容
1	2020/4/17	新規作成

© Copyright NEC Corporation 2020. All rights reserved.

## 免責事項

本書の内容は、予告なしに変更されることがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任をおいませぬ。

また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、お客様の責任とさせていただきます。

本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

## 商標情報

CLUSTERPRO® X は日本電気株式会社の登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における、登録商標または商標です。

本書に記載されたその他の製品名および標語は、各社の商標または登録商標です。

その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。



# 目次

はじめに .....	i
対象読者と目的 .....	i
適用範囲 .....	i
CLUSTERPRO マニュアル体系 .....	ii
本書の表記規則 .....	iii
最新情報の入手先 .....	iv
<b>第 1 章 機能概要</b> .....	<b>1</b>
<b>第 2 章 機能範囲</b> .....	<b>1</b>
<b>第 3 章 動作環境</b> .....	<b>1</b>
<b>第 4 章 構成例</b> .....	<b>2</b>
4.1. システム構成図 .....	3
4.2. サーバの構成 .....	4
4.3. クラスタの構成 .....	5
4.4. PostgreSQLの構成 .....	6
<b>第 5 章 構築手順</b> .....	<b>7</b>
5.1. 構築手順の概要 .....	7
5.2. ミラーディスク構成のプライマリサーバクラスタ作成 .....	7
5.3. スタンバイサーバのPostgreSQLインストール .....	8
5.3.1. PostgreSQLのインストール .....	8
5.3.2. OSユーザの確認 .....	8
5.3.3. OSユーザのパスワード設定 .....	8
5.4. PostgreSQLのレプリケーション設定 .....	9
5.4.1. フェイルオーバーグループの設定変更 .....	9
5.4.2. プライマリサーバのレプリケーション設定 .....	9
5.4.2.1. クライアント認証の設定 .....	9
5.4.2.2. 接続可能なIPアドレスの設定 .....	9
5.4.2.3. ストリーミングレプリケーションの設定 .....	9
5.4.2.4. プライマリデータベースの再起動 .....	9
5.4.3. スタンバイサーバのレプリケーション設定 .....	10
5.4.3.1. データベースユーザのレプリケーション接続パスワード設定 .....	10
5.4.3.2. プライマリ側WALアーカイブディレクトリの作成 .....	10
5.4.3.3. スタンバイ側WALアーカイブディレクトリの作成 .....	10
5.4.3.4. データベースの複製 .....	10
5.4.3.5. スタンバイデータベースの設定 .....	10
5.4.4. SSH公開鍵認証の設定 .....	11
5.4.4.1. SSHの有効化 .....	11
5.4.4.2. SSHの公開鍵作成 .....	12
5.4.5. プライマリサーバのWALアーカイビング設定 .....	12
5.4.5.1. WALアーカイビングの設定 .....	12
5.4.6. レプリケーション動作の確認 .....	12
5.4.6.1. プライマリデータベースの再起動 .....	12
5.4.6.2. スタンバイデータベースの起動 .....	13
5.4.6.3. プライマリサーバにおけるストリーミングレプリケーションの確認 .....	13
5.4.6.4. スタンバイサーバにおけるストリーミングレプリケーションの確認 .....	13
5.5. スタンバイサーバの監視設定 .....	14
5.5.1. スタンバイデータベースの接続パスワード設定 .....	14
5.5.2. スタンバイサーバ監視用カスタムモニタリソースの作成 .....	14

5.5.2.1. カスタムモニタリソースの作成.....	14
5.5.2.2. クラスタ構成情報の反映.....	16
<b>第 6 章 通常時の運用手順.....</b>	<b>17</b>
6.1. クラスタ上のプライマリサーバ操作(通常時).....	17
6.1.1. 稼働系ノードの切り替え.....	17
6.1.2. クラスタ操作とレプリケーションの中断.....	17
6.2. スタンバイサーバの操作(通常時).....	18
6.2.1. PostgreSQLの停止とレプリケーションの中断.....	18
6.2.2. スタンバイサーバの停止とレプリケーションの中断.....	18
<b>第 7 章 障害発生時の運用手順.....</b>	<b>19</b>
7.1. クラスタ上のプライマリサーバ(片系)の障害.....	19
7.2. クラスタ上のプライマリサーバ(両系)の障害.....	20
7.2.1. プライマリサーバ(両系)障害発生時の対応手順.....	20
7.2.1.1. スタンバイサーバの昇格による縮退状態への移行.....	20
7.2.2. プライマリサーバ復旧前の手順.....	20
7.2.2.1. フェイルオーバーグループの停止.....	20
7.2.3. クラスタ稼働状態への移行手順.....	20
7.2.3.1. スタンバイサーバのプライマリDBアクセス用IPアドレス削除.....	20
7.2.3.2. プライマリデータベースの最新化(複製).....	21
7.2.3.3. スタンバイデータベースの再設定.....	21
7.2.3.4. プライマリデータベースの再設定.....	21
7.2.3.5. フェイルオーバーグループの起動.....	21
7.2.3.6. スタンバイデータベースの起動.....	21
7.3. スタンバイサーバの障害.....	23
7.3.1. スタンバイサーバ障害発生時の対応手順.....	23
7.3.1.1. プライマリサーバのWALアーカイビング停止.....	23
7.3.2. スタンバイサーバ復旧後の手順.....	23
7.3.2.1. プライマリサーバのWALアーカイビング再設定.....	23
7.3.2.2. スタンバイデータベースの最新化(複製).....	24

# はじめに

## 対象読者と目的

『CLUSTERPRO® PPガイド』は、クラスタシステムに関して、システムを構築する管理者、およびユーザサポートを行うシステムエンジニア、保守員を対象にしています。

本書では、CLUSTERPRO環境下での動作確認が取れたソフトウェアをご紹介します。ここで紹介するソフトウェアや設定例は、あくまで参考情報としてご提供するものであり、各ソフトウェアの動作保証をするものではありません。

## 適用範囲

本書は、以下の製品を対象としています。

CLUSTERPRO X 4.1 for Linux

CLUSTERPRO X 4.2 for Linux

## CLUSTERPRO マニュアル体系

CLUSTERPRO のマニュアルは、以下の 6 つに分類されます。各ガイドのタイトルと役割を以下に示します。

### 『CLUSTERPRO X スタートアップガイド』 (Getting Started Guide)

すべてのユーザを対象読者とし、製品概要、動作環境、アップデート情報、既知の問題などについて記載します。

### 『CLUSTERPRO X インストール&設定ガイド』 (Install and Configuration Guide)

CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアと、クラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO を使用したクラスタシステム導入から運用開始前までに必須の事項について説明します。実際にクラスタシステムを導入する際の順番に則して、CLUSTERPRO を使用したクラスタシステムの設計方法、CLUSTERPRO のインストールと設定手順、設定後の確認、運用開始前の評価方法について説明します。

### 『CLUSTERPRO X リファレンスガイド』 (Reference Guide)

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象とし、CLUSTERPRO の運用手順、各モジュールの機能説明およびトラブルシューティング情報等を記載します。『インストール&設定ガイド』を補完する役割を持ちます。

### 『CLUSTERPRO X メンテナンスガイド』 (Maintenance Guide)

管理者、および CLUSTERPRO を使用したクラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO のメンテナンス関連情報を記載します。

### 『CLUSTERPRO X ハードウェア連携ガイド』 (Hardware Feature Guide)

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象読者とし、特定ハードウェアと連携する機能について記載します。『インストール&設定ガイド』を補完する役割を持ちます。

### 『CLUSTERPRO X 互換機能ガイド』 (Legacy Feature Guide)

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象読者とし、CLUSTERPRO X 4.0 WebManager および Builder に関する情報について記載します。



## 本書の表記規則

本書では、「注」および「重要」を以下のように表記します。

---

**注:** は、重要ではあるがデータ損失やシステムおよび機器の損傷には関連しない情報を表します。

---

**重要:** は、データ損失やシステムおよび機器の損傷を回避するために必要な情報を表します。

---

**関連情報:** は、参照先の情報の場所を表します。

---

また、本書では以下の表記法を使用します。

表記	使用方法	例
[ ] 角かっこ	コマンド名の前後 画面に表示される語 (ダイアログ ボックス、メニューなど) の前後	[スタート] をクリックします。 [プロパティ] ダイアログ ボックス
コマンドライン中の [ ] 角かっこ	かっこ内の値の指定が省略可能であることを示します。	<code>clpstat -s[-h host_name]</code>
モノスペース フォント (courier)	コマンド ライン、関数、パラメータ	<code>clpstat -s</code>
モノスペース フォント <b>太字</b> (courier)	ユーザが実際にコマンドプロンプトから入力する値を示します。	以下を入力します。 <b><code>clpcl -s -a</code></b>
モノスペース フォント (courier) <i>斜体</i>	ユーザが有効な値に置き換えて入力する項目	<code>clpstat -s [-h host_name]</code>

## 最新情報の入手先

最新の製品情報については、以下のWebサイトを参照してください。

<https://jpn.nec.com/clusterpro/>

# 第 1 章 機能概要

CLUSTERPRO とPostgreSQLのレプリケーション機能を組み合わせたクラスタ環境の構築手順や設定例を紹介します。

# 第 2 章 機能範囲

PostgreSQLについては以下のWebサイトを参照してください。

<https://www.postgresql.org/docs/11/index.html>

# 第 3 章 動作環境

本書の構築例で使用するOS、PostgreSQL及びCLUSTERPROのバージョンは下記の通りです。

<b>OS</b>	Red Hat Enterprise Linux 7.6
<b>PostgreSQL</b>	PostgreSQL 11
<b>CLUSTERPRO</b>	CLUSTERPRO X 4.1 for Linux (ミラーディスク) CLUSTERPRO X 4.2 for Linux (ミラーディスク)

## 第 4 章 構成例

CLUSTERPROのミラーディスク構成クラスタとPostgreSQLのレプリケーション機能が連携する構成例を説明します。本構成例では、ミラーディスクを使用してクラスタ化されたPostgreSQLのプライマリサーバと、クラスタ外のスタンバイサーバとの間でレプリケーションを行います。また、クラスタ側にスタンバイサーバの監視機能を構築し、スタンバイサーバの状態を監視します。

本構成例でのサーバの構成、クラスタの構成、PostgreSQLの構成を以下に記載します。

## 4.1. システム構成図

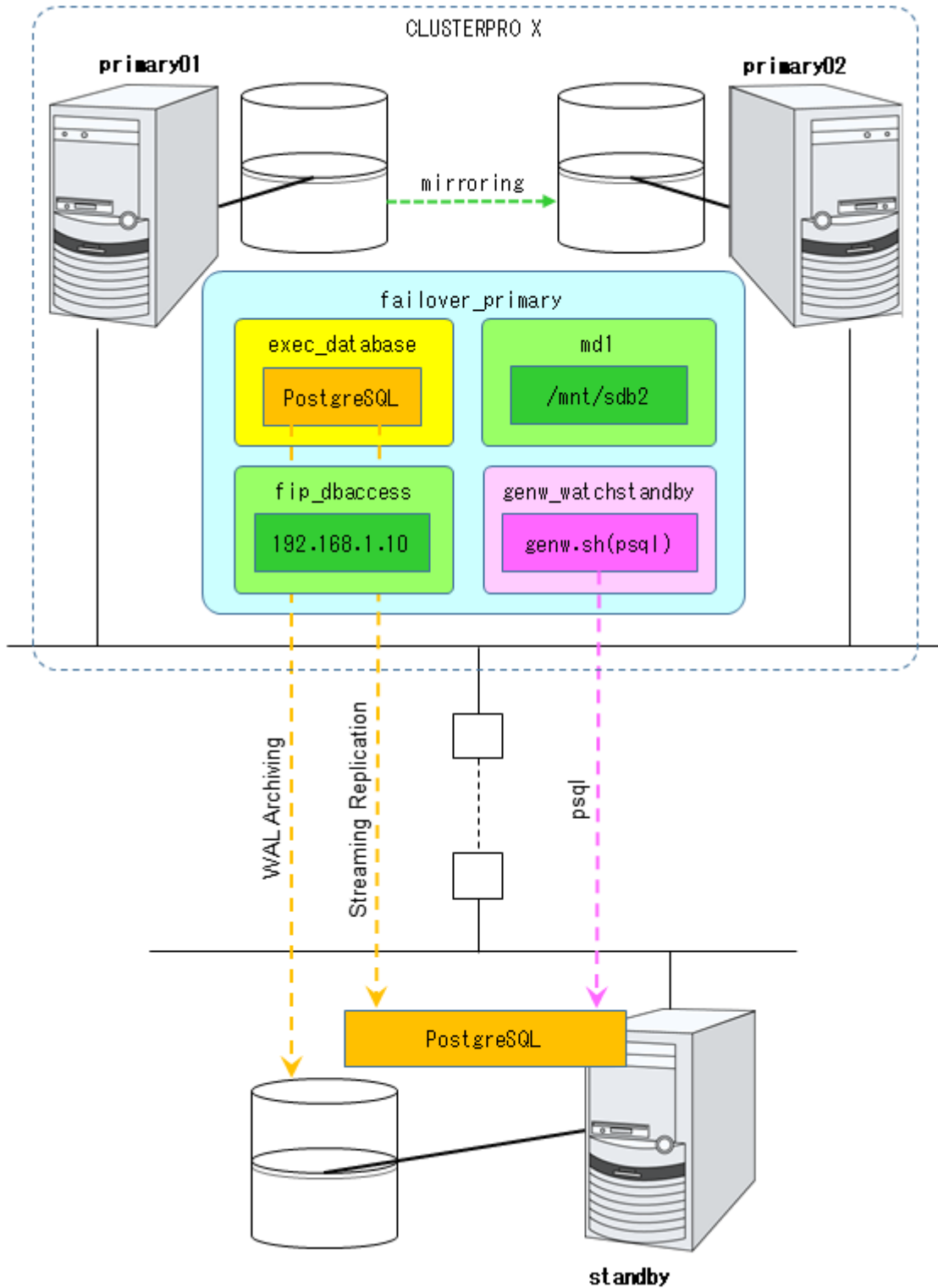


図4-1 ミラーディスク構成クラスターとPostgreSQLレプリケーションの連携例

## 4.2. サーバの構成

本手順で使用する各サーバの構成は以下の通りです。

対象	パラメータ	設定値
1台目のサーバ ・現用系クラスタノード ・ミラーディスク構成 ・レプリケーションでの役割: プライマリ	サーバ名	primary01
	IPアドレス	192.168.1.1
	クラスタパーティション	/dev/sdb1
	データパーティション	/dev/sdb2
2台目のサーバ ・待機系クラスタノード ・ミラーディスク構成 ・レプリケーションでの役割: プライマリ	サーバ名	primary02
	IPアドレス	192.168.1.2
	クラスタパーティション	/dev/sdb1
	データパーティション	/dev/sdb2
3台目のサーバ ・クラスタ外のサーバ ・レプリケーションでの役割: スタンバイ	サーバ名	standby
	IPアドレス	192.168.1.3

## 4.3. クラスタの構成

ミラーディスク型クラスタの構成は以下の通りです。

設定対象	設定パラメータ	設定値
クラスタ	クラスタ名	cluster
	言語	日本語
サーバ	サーバ1	primary01
	サーバ2	primary02
インタコネク	種別	カーネルモード
	MDC	mdc1
	(サーバ1のIPアドレス)	192.168.1.1
	(サーバ2のIPアドレス)	192.168.1.2
<b>フェイルオーバーグループ</b>		
グループ1 ・プライマリDB用グループ	タイプ	フェイルオーバー
	グループ名	failover_primary
	起動サーバ	全てのサーバ
<b>リソース (グループ1)</b>		
リソース1 ・プライマリDBアクセス用IPアドレス	タイプ	フローティングIPリソース
	リソース名	fip_dbaccess
	依存するリソース	なし
	IPアドレス	192.168.1.10
リソース2 ・プライマリDBファイル用ミラーディスク	タイプ	ミラーディスクリソース
	リソース名	md1
	依存するリソース	既定の依存関係
	マウントポイント	/mnt/sdb2
リソース3 ・プライマリDB起動・停止用スクリプト	タイプ	EXECリソース
	リソース名	exec_database
	依存するリソース	既定の依存関係
	データパーティションデバイス	/dev/sdb2
<b>モニタリソース</b>		
モニタリソース1 ・デフォルトで自動作成	タイプ	ユーザ空間モニタ
	モニタリソース名	userw
モニタリソース2 ・フローティングIPリソース作成時に自動作成	タイプ	フローティングIPモニタ
	モニタリソース名	fipw1
	対象リソース	fip_dbaccess
	回復対象	fip_dbaccess
モニタリソース3 ・ミラーディスクリソース作成時に自動作成	タイプ	ミラーディスクコネク
	モニタリソース名	mdnw1
	対象リソース	md1
モニタリソース4 ・ミラーディスクリソース作成時に自動作成	タイプ	ミラーディスクモニタ
	モニタリソース名	mdw1
	対象リソース	md1
モニタリソース5 ・プライマリDB監視用PostgreSQLモニタ	タイプ	PostgreSQLモニタ
	モニタリソース名	psqlw
	対象リソース	exec_database
モニタリソース6 ・スタンバイ監視用カスタムモニタ	タイプ	カスタムモニタ
	モニタリソース名	genw_watchstandby
	対象リソース	exec_database

## 4.4. PostgreSQL の構成

PostgreSQLの構成は以下の通りです。

以降は、下記設定での構築手順を記載します。各設定値は実際の環境に適宜置き換えてお読みください。

インストールの設定	
PostgreSQLのバージョン	11.4
インストール方法	rpm
インストールディレクトリ	/usr/pgsql-11
管理用OSユーザ	postgres
データベースユーザ	postgres
使用ポート番号	5432
クライアント認証方式	md5
データベースの設定	
プライマリ側データベースディレクトリ	/mnt/sdb2/pgsql/data
スタンバイ側データベースディレクトリ	/var/lib/pgsql/11/data
データベース名	db1
文字コード	UTF8
レプリケーションの設定	
レプリケーション方法	ストリーミング
同期方式	非同期
WALアーカイビング	on
プライマリ側WALアーカイブディレクトリ	192.168.1.3:/var/lib/pgsql/11/from_primary
スタンバイ側WALアーカイブディレクトリ	/var/lib/pgsql/11/backpus
レプリケーションスロット	なし



## 第5章 構築手順

CLUSTERPROのミラーディスク構成とPostgreSQLのレプリケーション機能が連携するクラスタの構築手順を構成例に基づいて説明します。

### 5.1. 構築手順の概要

以下の手順でクラスタ化を行います。

- 5.2. ミラーディスク構成のプライマリサーバクラスタ作成
- 5.3. スタンバイサーバのPostgreSQLインストール
- 5.4. PostgreSQLのレプリケーション設定
- 5.5. スタンバイサーバの監視設定

### 5.2. ミラーディスク構成のプライマリサーバクラスタ作成

CLUSTERPROを使用して、ミラーディスク構成のPostgreSQLプライマリサーバクラスタを作成します。対象サーバは、プライマリサーバprimary01とprimary02です。

「CLUSTERPRO X for Linux PowerGres V9.0 / V9.1 / V9.4 / Plus 66 V9.1 / PostgreSQL 9.0 HowTo」(Linux\_PG9\_HowTo\_04.pdf) に従い、primary01 と primary02 両サーバに CLUSTERPRO 及び PostgreSQLをインストールし、ミラーディスク構成のクラスタを作成します。

作成されるフェイルオーバーグループの構成	
グループ	リソース
failover_primaryグループ	fip_dbaccessリソース
	md1リソース
	exec_databaseリソース

インストールされるPostgreSQLの設定	
データベースディレクトリ	/mnt/sdb2/pgsql/data
管理用OSユーザ	postgres
データベースユーザ	postgres

## 5.3. スタンバイサーバの PostgreSQL インストール

スタンバイサーバstandbyにPostgreSQLをインストールします。

### 5.3.1. PostgreSQL のインストール

PostgreSQL のrpmをサーバの任意の場所に展開してインストールを行います。

```
# su - root
(以下、rootユーザで実行)
# rpm -i postgresql11-libs-11.4-1PGDG.rhel7.x86_64.rpm
# rpm -i postgresql11-11.4-1PGDG.rhel7.x86_64.rpm
# rpm -i postgresql11-contrib-11.4-1PGDG.rhel7.x86_64.rpm
# rpm -i postgresql11-server-11.4-1PGDG.rhel7.x86_64.rpm
```

### 5.3.2. OS ユーザの確認

スタンバイサーバstandbyで作成されたOSユーザ postgres が、プライマリサーバprimary01に存在するOSユーザ postgres と同一ユーザIDで設定されているか確認します。

```
# id postgres
```

uid、gid、groupsの情報が両サーバで等しければ、postgresユーザの再設定は必要ありません。両サーバでID が異なる場合は、ID が等しくなるようにpostgresユーザの設定を変更してください。またスタンバイサーバにpostgresユーザが存在しない場合は、プライマリサーバと同一IDになるようにpostgresユーザを作成してください。

### 5.3.3. OS ユーザのパスワード設定

OSユーザ postgres のパスワードを設定します。

```
# passwd postgres
Changing password for user postgres.
New password: xxxxx
Retry new password: xxxxx
```

※太字斜体の部分にパスワードを指定します。

## 5.4. PostgreSQL のレプリケーション設定

プライマリサーバとスタンバイサーバでPostgreSQLのレプリケーション設定を行います。

### 5.4.1. フェイルオーバーグループの設定変更

プライマリサーバでの設定作業中の誤動作を防ぐため、フェイルオーバーグループfailover\_primaryを自動起動から手動起動に変更してください。

### 5.4.2. プライマリサーバのレプリケーション設定

以下の設定をプライマリサーバprimary01で行います。

postgresユーザでプライマリデータベースのデータベースディレクトリに移動します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd /mnt/sdb2/pgsql/data
```

#### 5.4.2.1. クライアント認証の設定

pg\_hba.conf ファイルでクライアント認証の設定を行います。データベースディレクトリが作成された初期状態では、ループバックIP アドレスおよび、UNIX ドメインソケットからの接続許可のみが設定されています。データベースユーザ postgres にreplication仮想データベースへの接続権限を与えます。また、192.168.1.0/24 からの接続を許可する設定を行います。以下の二行を追加します。

```
pg_hba.conf ファイル
host all          all          192.168.1.0/24 md5
host replication postgres 192.168.1.0/24 md5
```

環境に応じて接続を許可・禁止するクライアントの設定を行ってください。

#### 5.4.2.2. 接続可能なIPアドレスの設定

postgresql.conf ファイルで、クライアントからの接続の監視に使用するIPインターフェースの設定を、利用可能な全てのIPインターフェースに対応するように設定します。[listen\_addresses]エントリのコメントアウトを削除し、以下のように変更します。

```
postgresql.conf ファイル
listen_addresses = '*'
```

#### 5.4.2.3. ストリーミングレプリケーションの設定

postgresql.conf ファイルで、ストリーミングレプリケーションのために以下を設定します。

[wal\_level]エントリのコメントアウトを削除し、設定値を replica に変更します

[max\_wal\_senders]エントリのコメントアウトを削除し、設定値を スタンバイデータベースの数+1 に変更します。

```
postgresql.conf ファイル
wal_level = replica
max_wal_senders = 2      # スタンバイデータベースの数 + 1
```

#### 5.4.2.4. プライマリデータベースの再起動

上記の設定をすべて有効にするため、プライマリデータベースを再起動してください。CLUSTERPROからexec\_databaseリソースを停止、起動してください。

### 5.4.3. スタンバイサーバのレプリケーション設定

以下の設定をスタンバイサーバstandbyで行います。

#### 5.4.3.1. データベースユーザのレプリケーション接続パスワード設定

OSユーザ postgresのホームディレクトリに、データベースユーザ postgres がレプリケーション接続する際に使用するパスワードを設定した.pgpassファイルを作成します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd ~
$ touch .pgpass
$ chown postgres:postgres .pgpass
$ chmod 600 .pgpass
```

.pgpassファイルに、データベースユーザ postgres がレプリケーション仮想データベースにパスワードなしで接続できるように設定します。ファイル内の書式は以下の通りです。

hostname:port:database:username:password

hostnameには、fip\_dbaccessに設定したIPアドレス、portにはPostgreSQLの接続に使用するポート番号を指定してください。databaseには"replication"を指定します。

#### .pgpass ファイル

```
192.168.1.10:5432:replication:postgres:XXXXX
```

※太字斜体の部分にパスワードを指定します。パスワードにはOSユーザのパスワードではなく、データベースユーザのパスワードを記載します。

#### 5.4.3.2. プライマリ側WALアーカイブディレクトリの作成

以下の例では、プライマリサーバのWALアーカイブをスタンバイサーバ上に保存するため、アーカイブ用ディレクトリをスタンバイサーバ上に作成します。

```
(以下、postgresユーザで実行)
$ mkdir /var/lib/pgsql/11/from_primary
$ chmod 700 /var/lib/pgsql/11/from_primary
```

#### 5.4.3.3. スタンバイ側WALアーカイブディレクトリの作成

以下の例では、スタンバイサーバがプライマリサーバに昇格した場合にWALアーカイビングを行うため、アーカイブ用ディレクトリをスタンバイサーバ上に作成します。

```
(以下、postgresユーザで実行)
$ mkdir /var/lib/pgsql/11/backpus
$ chmod 700 /var/lib/pgsql/11/backpus
```

#### 5.4.3.4. データベースの複製

pg\_basebackup コマンドを使用してプライマリサーバのデータベースをスタンバイサーバに複製します。

postgres ユーザで、以下のコマンドを実行します。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_basebackup -h 192.168.1.10 -U postgres -D /var/lib/pgsql/11/data -c fast -X
stream -P
Password:XXXXX
```

※太字斜体の部分にパスワードを指定します。

複製終了後はスタンバイサーバ上のデータベースを最新状態で保つため、速やかにプライマリサーバ上のデータベースを停止します。CLUSTERPROからexec\_databaseリソースを停止してください。

#### 5.4.3.5. スタンバイデータベースの設定

複製したデータベースディレクトリ直下に移動します。

```
(以下、postgresユーザで実行)
$ cd /var/lib/pgsql/11/data
```

データベースディレクトリの直下にrecovery.conf ファイルを作成します。

```
(以下、postgresユーザで実行)
$ touch recovery.conf
$ chmod 600 recovery.conf
```

recovery.conf ファイルで、スタンバイモードとストリーミングレプリケーションの設定をします。

[standby\_mode]エントリの設定値を'on'に設定します。

[recovery\_target\_timeline]エントリの設定値を'latest' に設定します。

[restore\_command]エントリに、プライマリ側WALアーカイブからWALファイルを取得するコマンドを設定します。以下の例ではcpコマンドを使用します。

[archive\_cleanup\_command]エントリに、使用済みWALファイルの削除コマンドを設定します。以下の例ではpg\_archivecleanupコマンドを使用します。

[primary\_conninfo]エントリの設定値に、プライマリデータベースへの接続文字列とapplication\_name='任意の接続識別名'を記述します。(この接続識別名は、同期レプリケーションの場合にプライマリデータベースの同期設定で参照します。本構成例は非同期なので参照しませんが、同期方式を変更する場合に備えて適切な値を設定してください)

#### recovery.conf ファイル

```
standby_mode = 'on'
recovery_target_timeline = 'latest'
restore_command = 'cp /var/lib/pgsql/11/from_primary/%f %p'
archive_cleanup_command = '/usr/pgsql-11/bin/pg_archivecleanup /var/lib/pgsql/11/from_primary %r'
primary_conninfo = 'host=192.168.1.10 port=5432 user=postgres application_name=slave'
```

postgresql.conf ファイルで、ホットスタンバイとWALアーカイビングを設定します。WALアーカイビングの設定はスタンバイサーバがプライマリサーバに昇格した場合に使用されます。

[hot\_standby]エントリの設定値を on に設定します。

[archive\_mode]エントリの設定値を on に設定します。

[archive\_command]エントリにWALファイルをWALアーカイブディレクトリにコピーするコマンドを設定します。以下の例ではcpコマンドを使用します。

#### postgresql.conf ファイル

```
hot_standby = on
archive_mode = on
archive_command = 'cp %p /var/lib/pgsql/11/backpus/%f'
```

## 5.4.4. SSH 公開鍵認証の設定

WALアーカイビングでscpコマンドを使う場合、コマンド実行時のパスワード入力を不要にするため、OSユーザ postgres のSSHにおける認証方式をパスワード認証から公開鍵認証に変更します。

以下の設定をプライマリサーバのprimary01とprimary02の両方で行います。

(スタンバイサーバstandbyが昇格後にWALアーカイビングでscpコマンドを使う場合は、スタンバイサーバでも必要。)

### 5.4.4.1. SSHの有効化

OSユーザ postgres でscpコマンドやsshコマンドを使用して、SSH接続が可能かどうか確認します。

SSH接続できない場合は、rootユーザでsshdサービスの状態や/etc/ssh/sshd\_configの内容を確認し、必要な修正を行ってください。

### 5.4.4.2. SSHの公開鍵作成

scpコマンドを使用するサーバ上で、OSユーザ postgres 用に公開鍵を作成し、接続先へ配布します。  
ssh-keygenコマンドを使用して公開鍵を作成します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd ~
$ ssh-keygen                # デフォルト設定で公開鍵作成
Enter file in which to save the key (~/ssh/id_rsa): # 入力せずリターン
Enter passphrase (empty for no passphrase): # 入力せずリターン
Enter same passphrase again: # 入力せずリターン
```

ssh-copy-idコマンドを使用して、接続先サーバに対して公開鍵をコピーします。  
複数の接続先がある場合はすべてのサーバに対してssh-copy-idコマンドを実行します。

```
(以下、postgresユーザで実行)
$ ssh-copy-id postgres@192.168.1.3 # スタンバイサーバに公開鍵をコピー
```

すべての接続先サーバにSSH接続して、パスワードが要求されないことを確認します。

```
(以下、postgresユーザで実行)
$ ssh 192.168.1.3 # スタンバイサーバにsshでログイン
```

接続エラーになる場合は'ssh -v'で詳細エラーを確認し、必要な修正を行ってください。

### 5.4.5. プライマリサーバの WAL アーカイビング設定

以下の設定をプライマリサーバprimary01で行います。

CLUSTERPROからexec\_databaseリソースを起動後、postgresユーザでプライマリデータベースのデータベースディレクトリに移動します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd /mnt/sdb2/pgsql/data
```

#### 5.4.5.1. WALアーカイビングの設定

postgresql.conf ファイルで、WALアーカイビングの設定を行います。

WALアーカイブを有効にすることで、スタンバイデータベースが長期間停止後に再開した場合などに、アーカイブされたWALファイルを使ってプライマリデータベースとスタンバイデータベースの整合をとることが可能です。

[archive\_mode]エントリの設定値を on に設定します。

[archive\_command]エントリの設定値に アーカイブ保存ディレクトリにファイルをコピーするコマンドを記述します。

以下の例では、scpコマンドを使用してプライマリサーバのWALファイルをスタンバイサーバの/var/lib/pgsql/11/from\_primary 配下にコピーします。

```
postgresql.conf ファイル
archive_mode = on
archive_command = 'scp %p 192.168.1.3:/var/lib/pgsql/11/from_primary/%f'
```

### 5.4.6. レプリケーション動作の確認

#### 5.4.6.1. プライマリデータベースの再起動

ここまでの設定を有効にするため、プライマリサーバでプライマリデータベースを起動してください。  
CLUSTERPROからexec\_databaseリソースを起動してください。  
また、フェイルオーバーグループfailover\_primaryを手動起動から自動起動に戻してください。

#### 5.4.6.2. スタンバイデータベースの起動

スタンバイサーバで、スタンバイデータベースを開始します。

```
# su - postgres
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_ctl start -D /var/lib/pgsql/11/data
```

#### 5.4.6.3. プライマリサーバにおけるストリーミングレプリケーションの確認

プライマリサーバでデータベースに接続し、ストリーミングレプリケーションが動作していることを確認します。  
以下を実行し、出力結果と同様な値が返却されることを確認してください。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/psql postgres
Password for user postgres:XXXX

postgres=# select username,application_name,state,sync_state from pg_stat_replication;
(出力結果)
 username | application_name | state | sync_state
-----+-----+-----+-----
 postgres | slave           | streaming | async
(1 row)
```

出力結果が「(0 row)」の場合、レプリケーションが開始されていません。

また、各カラムの値が設定値と異なる場合はレプリケーションの設定に修正が必要です。

いずれの場合も、レプリケーションの設定(プライマリサーバのpostgresql.conf、スタンバイサーバのpostgresql.conf、recovery.conf)を見直して、postgresqlを再起動した後、再度確認してください。

#### 5.4.6.4. スタンバイサーバにおけるストリーミングレプリケーションの確認

スタンバイサーバでデータベースに接続し、ストリーミングレプリケーションが動作していることを確認します。  
以下を実行し、出力結果と同様な値が返却されることを確認してください。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/psql postgres
Password for user postgres:XXXX

postgres=# select status from pg_stat_wal_receiver;
(出力結果)
 status
-----
 streaming
(1 row)
```

出力結果が「(0 row)」の場合、レプリケーションが開始されていません。

また、各カラムの値が設定値と異なる場合はレプリケーションの設定に修正が必要です。

いずれの場合も、レプリケーションの設定(プライマリサーバのpostgresql.conf、スタンバイサーバのpostgresql.conf、recovery.conf)を見直して、postgresqlを再起動した後、再度確認してください。

## 5.5. スタンバイサーバの監視設定

プライマリサーバにスタンバイサーバ監視のための設定を行います。  
スタンバイサーバのレプリケーションを監視することで、レプリケーションの停止やスタンバイサーバへの接続不可などの異常発生時に、CLUSTERPROのCluster WebUI上でアラート表示可能になります。  
具体的にはスタンバイデータベースへの接続パスワード設定と、スタンバイサーバ監視用カスタムモニタリソースの作成を行います。

### 5.5.1. スタンバイデータベースの接続パスワード設定

以下の設定をプライマリサーバのprimary01とprimary02の両方で行います。  
OSユーザ postgresのホームディレクトリに、データベースユーザ postgres がレプリケーション接続する際に使用するパスワードを設定した.pgpassファイルを作成します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd ~
$ touch .pgpass
$ chown postgres:postgres .pgpass
$ chmod 600 .pgpass
```

.pgpassファイルに、データベースユーザ postgres がスタンバイデータベースにパスワードなしで接続できるように設定します。ファイル内の書式は以下の通りです。  
hostname:port:database:username:password  
hostnameには、スタンバイサーバのIPアドレス、portにはPostgreSQLの接続に使用するポート番号を指定してください。databaseには"postgres"を指定します。

```
.pgpass ファイル
192.168.1.3:5432:postgres:postgres:XXXXX
```

※太字斜体の部分にパスワードを指定します。パスワードにはOSユーザのパスワードではなく、データベースユーザのパスワードを記載します。

### 5.5.2. スタンバイサーバ監視用カスタムモニタリソースの作成

以下の設定をプライマリサーバprimary01で行います。

#### 5.5.2.1. カスタムモニタリソースの作成

カスタムモニタリソースgenw\_watchstandbyをクラスタに追加します。  
監視(共有)タブで、監視対象リソースとしてEXECリソースexec\_databaseを設定します。

監視(固有)タブで、「この製品で作成したスクリプト」ファイルgenw.shを編集し、監視スクリプトを作成します。  
以下は監視スクリプトの例です。このスクリプト例では監視タイプは「同期」に設定します。また、ログ出力先を指定することで、ログが出力されます。

#### **genw.sh ファイル**

```
#!/bin/sh
#*****
#*                               genw.sh                               *
#*****

ulimit -s unlimited

psql_osuser="postgres"
```



```

psql_host="192.168.1.10"
psql_stdby="192.168.1.3"
psql_port=5432
psql_dbuser="postgres"

repl_status="streaming"
repl_missing=" [psql-standby: replication streaming is not found.]"

psql_cmd="sudo -u $psql_osuser -i ¥
LANG=C psql -h $psql_stdby -p $psql_port -U $psql_dbuser -w ¥
-c 'select status,sender_host from pg_stat_wal_receiver;'"

# functions
function outputError () {
  # To log file
  echo " [connect:osuser=$psql_osuser, standby=$psql_stdby, port=$psql_port,
dbuser=$psql_dbuser]"
  echo " [request:status=$repl_status, host=$psql_host]"
  echo
  echo "$rslt_msg"
  echo " [exit=$rslt_val]"
  echo

  # To alert log
  # make escaped string
  escaped_msg=`echo $rslt_msg | sed 's/¥([¥¥!"¥ ]¥)/¥¥1/g`
  clogcmd -m "$escaped_msg" --alert -i $rslt_val -l ERR > /dev/null 2>&1
}

#
# main process
#
date -lseconds

# connect and get replication streaming lines
rslt_msg=` ( eval ${psql_cmd} 2>&1 ) `
rslt_val=$?

if [ $rslt_val -ne 0 ]; then
  outputError
  exit $rslt_val
fi

# search replication streaming by status and host
( echo ${rslt_msg} | grep $repl_status | grep $psql_host ) > /dev/null 2>&1
rslt_val=$?

if [ $rslt_val -ne 0 ]; then
  rslt_msg=${repl_missing}
  outputError
  exit $rslt_val
fi

echo " [exit=$rslt_val]"
echo
exit $rslt_val

```

次に回復動作タブを設定します。

## 第 5 章 構築手順

カスタムモニタリソースgenw\_watchstandbyが異常検出した場合、クラスタ側でフェイルオーバーを発生させる必要はないため、以下の設定をします。

回復動作は「最終動作のみ実行」、回復対象はexec\_databaseを指定します。

最終動作は「何もしない」を指定します。

### 5.5.2.2. クラスタ構成情報の反映

モニタリソースを作成した後、クラスタ構成情報をアップロードしてサーバに反映させます。

設定反映後、フェイルオーバーグループを起動することでスタンバイサーバの監視が開始されます。

## 第6章 通常時の運用手順

CLUSTERPROのミラーディスク構成とPostgreSQLのレプリケーション機能が連携するクラスタにおける通常時の運用手順の例を説明します。

### 6.1. クラスタ上のプライマリサーバ操作(通常時)

プライマリサーバprimary01とprimary02について、CLUSTERPROからフェイルオーバーグループfailover\_primary を操作することで、クラスタ上の稼働系ノードの切り替えやPostgreSQLの起動／停止が実行できます。

#### 6.1.1. 稼働系ノードの切り替え

稼働系ノードを切り替える場合は、フェイルオーバーグループfailover\_primaryの移動を実行してください。新たに稼働系ノードとなったプライマリサーバは、スタンバイサーバstandbyとのレプリケーションを継続します。

#### 6.1.2. クラスタ操作とレプリケーションの中断

CLUSTERPROから以下の操作を行った場合、スタンバイサーバstandbyとのレプリケーションは中断されます。停止したものを再び起動すれば、スタンバイサーバstandbyとのレプリケーションは再開されます。

- フェイルオーバーグループfailover\_primaryもしくは同グループ内のリソースを停止した場合
- 全てのプライマリサーバを停止した場合
- クラスタを停止した場合

## 6.2. スタンバイサーバの操作(通常時)

本構成例ではスタンバイサーバstandbyはクラスタ外の構成のため、各種OS操作やPostgreSQL操作はスタンバイサーバ上で適切なOSユーザから行ってください。

### 6.2.1. PostgreSQL の停止とレプリケーションの中断

スタンバイサーバのPostgreSQLを停止した場合、プライマリサーバとのストリーミングレプリケーションは中断されますが、本構成例ではプライマリサーバからスタンバイサーバにネットワーク接続が可能な状態であればWALアーカイビングは継続されます。

ただし、長期間に渡ってPostgreSQLを停止すると、蓄積されたWALアーカイブによってスタンバイサーバのディスクが空き容量不足になる可能性があります。

PostgreSQLを再び起動すれば、蓄積されたWALアーカイブは適切に処理され、プライマリサーバとのストリーミングレプリケーションは再開されます。

### 6.2.2. スタンバイサーバの停止とレプリケーションの中断

スタンバイサーバ自体を停止した場合、プライマリサーバとのストリーミングレプリケーションは中断されます。また本構成例ではスタンバイサーバ上にWALアーカイビングを行うため、WALアーカイビングも停止します。

WALアーカイブされなかったWALファイルはプライマリサーバ上に残り続けるため、長期間に渡ってスタンバイサーバを停止すると、WALファイルによってプライマリサーバのディスクが空き容量不足になる可能性があります。

スタンバイサーバを再び起動すれば、蓄積されたWALファイルは適切に処理され、プライマリサーバとのストリーミングレプリケーションは再開されます。

## 第7章 障害発生時の運用手順

CLUSTERPROのミラーディスク構成とPostgreSQLのレプリケーション機能が連携するクラスタにおける障害発生時の運用手順の例を説明します。

### 7.1. クラスタ上のプライマリサーバ(片系)の障害

クラスタの稼働系ノードにおいて、CLUSTERPROのPostgreSQLモニタリソースによりPostgreSQLの障害が検出された場合、フェイルオーバーグループfailover\_primaryは自動的に待機系ノードにフェイルオーバーされます。

フェイルオーバーした後、障害が発生したノードの復旧作業を行ってください。

障害ノードが復旧した後は、フェイルオーバーグループを手動で移動することで元の稼働系ノードに戻すことができます。

(フェイルオーバー先でも障害が発生した場合は後述の「クラスタ上のプライマリサーバ(両系)の障害」を参照)

## 7.2. クラスタ上のプライマリサーバ(両系)の障害

クラスタの稼働系と待機系の両方のノードでDB監視(PostgreSQLモニタ)によってPostgreSQLの障害が検出された場合、フェイルオーバーグループfailover\_primaryはDB監視(PostgreSQLモニタ)の最終動作に従った動作を行います。

本構成例では、フェイルオーバーグループfailover\_primaryが障害により停止した場合、スタンバイサーバの昇格により、縮退状態で業務継続が可能です。

また、クラスタ上のプライマリサーバが復旧した後は、再度クラスタ稼働状態に移行することができます。

### 7.2.1. プライマリサーバ(両系)障害発生時の対応手順

プライマリサーバ(両系)で障害が発生した場合、以下の手順を実行してください。

#### 7.2.1.1. スタンバイサーバの昇格による縮退状態への移行

スタンバイサーバのPostgreSQLをpg\_ctl promoteコマンドを使用してプライマリデータベースに昇格させます。スタンバイサーバstandbyで以下を実行します。

```
(以下、postgresユーザで実行)  
$ /usr/pgsql-10/bin/pg_ctl promote -D /var/lib/pgsql/11/data
```

データベースが昇格した際に、recovery.confファイルはrecovery.doneに名称変更されます。

スタンバイサーバにプライマリDBアクセス用IPアドレスを設定します。

```
(以下、rootユーザで実行)  
# ip addr add 192.168.1.10 dev ens192 # 一時的なアドレスを設定する場合
```

本構成例の場合、この手順後はクラスタ外のスタンバイサーバ1台による縮退状態での運用となります。

### 7.2.2. プライマリサーバ復旧前の手順

縮退状態からクラスタ上のプライマリサーバの復旧作業をする場合、作業前に以下の手順を実行してください。

#### 7.2.2.1. フェイルオーバーグループの停止

復旧作業の簡易化のため、プライマリサーバ上でCLUSTERPROからフェイルオーバーグループfailover\_primaryを停止してください。また、フェイルオーバーグループfailover\_primaryを自動起動から手動起動に変更してください。

### 7.2.3. クラスタ稼働状態への移行手順

プライマリサーバの復旧作業をした後は、縮退状態からクラスタ稼働状態に移行するため、以下の手順を実行してください。

#### 7.2.3.1. スタンバイサーバのプライマリDBアクセス用IPアドレス削除

スタンバイサーバのプライマリDBアクセス用IPアドレスを削除します。

```
(以下、rootユーザで実行)  
# ip addr del 192.168.1.10 dev ens192 # 一時的なアドレスを削除する場合
```

### 7.2.3.2. プライマリデータベースの最新化(複製)

プライマリサーバ上のデータベースが停止していることを確認してください。プライマリサーバ上のデータベースが起動している場合は、以下の手順で停止してください。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_ctl stop -D /mnt/sdb2/pgsql/data
```

pg\_basebackup コマンドを使用してスタンバイサーバのデータベースをプライマリサーバに複製することで、プライマリサーバ上のデータベースを最新化します。

プライマリサーバのpostgres ユーザで、以下のコマンドを実行します。

```
(以下、postgresユーザで実行)
$ mv /mnt/sdb2/pgsql/data /mnt/sdb2/pgsql/data.old          # 既存データをリネームして退避
$ /usr/pgsql-11/bin/pg_basebackup -h 192.168.1.3 -U postgres -D /mnt/sdb2/pgsql/data -c fast -X
stream -P
Password:XXXXXX
```

※太字斜体の部分にパスワードを指定します。

複製終了後はプライマリサーバ上のデータベースを最新状態で保つため、速やかにスタンバイサーバ上のデータベースを停止します。

スタンバイサーバのpostgres ユーザで、以下のコマンドを実行します。

```
# su - postgres
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_ctl stop -D /var/lib/pgsql/11/data/
```

### 7.2.3.3. スタンバイデータベースの再設定

スタンバイサーバ上のデータベースを、再度スタンバイデータベースとして設定します。

recovery.doneファイルをrecovery.confファイルに戻します。

```
# su - postgres
(以下、postgresユーザで実行)
$ mv /var/lib/pgsql/11/data/recovery.done /var/lib/pgsql/11/data/recovery.conf
```

### 7.2.3.4. プライマリデータベースの再設定

プライマリサーバ上のデータベースをプライマリデータベースとして再設定します。

postgresql.conf ファイルで、WALアーカイビングの再設定を行います。

[archive\_mode]エントリの設定値を on に設定します。

[archive\_command]エントリの設定値に アーカイブ保存ディレクトリにファイルをコピーするコマンドを記述します。

以下の例では、scpコマンドを使用してプライマリサーバのWALファイルをスタンバイサーバの/var/lib/pgsql/11/from\_primary 配下にコピーします。

```
postgresql.conf ファイル
archive_mode = on
archive_command = 'scp %p 192.168.1.3:/var/lib/pgsql/11/from_primary/%f'
```

### 7.2.3.5. フェイルオーバーグループの起動

プライマリサーバでプライマリデータベースを起動するため、CLUSTERPROからexec\_databaseリソースを起動してください。また、フェイルオーバーグループfailover\_primaryを手動起動から自動起動に戻してください。

### 7.2.3.6. スタンバイデータベースの起動

スタンバイサーバで、スタンバイデータベースを開始します。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_ctl start -D /var/lib/pgsql/11/data
```

## 第 7 章 障害発生時の運用手順



## 7.3. スタンバイサーバの障害

スタンバイサーバにおいてPostgreSQLが障害で停止した場合、プライマリサーバとのストリーミングレプリケーションは中断されますが、本構成例ではプライマリサーバのWALアーカイビングが有効なため、スタンバイサーバまたはプライマリサーバにWALファイルが保存され続けます。

そこで、長期間の停止によるディスクの空き容量不足を回避するため、プライマリサーバのWALアーカイビングを停止します。

WALアーカイビングを停止すると、スタンバイデータベースの同期に必要な情報が失われるため、スタンバイサーバ復帰時にはスタンバイデータベースをプライマリデータベースから複製する必要があります。

### 7.3.1. スタンバイサーバ障害発生時の対応手順

スタンバイサーバで障害が発生した場合、以下の手順を実行してください。

#### 7.3.1.1. プライマリサーバのWALアーカイビング停止

WALアーカイビングを停止するため、以下の設定をプライマリサーバprimary01で行います。

postgresユーザでプライマリデータベースのデータベースディレクトリに移動します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd /mnt/sdb2/pgsql/data
```

postgresql.conf ファイルで、WALアーカイビングの設定変更を行います。

[archive\_command]エントリの設定値を変更して、アーカイビングを無効にします。

以下の例では、trueコマンドを使用してWALファイルのコピーを無効化しています。

```
postgresql.conf ファイル
archive_command = 'true'
```

設定を有効にするため、プライマリデータベースの設定を再読み込みします。

```
(以下、postgresユーザで実行)
$ /usr/pgsql-11/bin/pg_ctl reload -D /mnt/sdb2/pgsql/data
```

### 7.3.2. スタンバイサーバ復旧後の手順

スタンバイが復旧した場合、以下の手順を実行してください

#### 7.3.2.1. プライマリサーバのWALアーカイビング再設定

postgresユーザでプライマリデータベースのデータベースディレクトリに移動します。

```
# su - postgres
(以下、postgresユーザで実行)
$ cd /mnt/sdb2/pgsql/data
```

postgresql.conf ファイルで、WALアーカイビングの再設定を行います。

[archive\_command]エントリの設定値に アーカイブ保存ディレクトリにファイルをコピーするコマンドを記述します。

以下の例では、scpコマンドを使用してプライマリサーバのWALファイルをスタンバイサーバの/var/lib/pgsql/11/from\_primary 配下にコピーします。

```
postgresql.conf ファイル
```

```
archive_command = 'scp %p 192.168.1.3:/var/lib/pgsql/11/from_primary/%f'
```

設定を有効にするため、プライマリデータベースの設定を再読み込みします。

```
(以下、postgresユーザで実行)  
$ /usr/pgsql-11/bin/pg_ctl reload -D /mnt/sdb2/pgsql/data
```

### 7.3.2.2. スタンバイデータベースの最新化(複製)

スタンバイサーバ上のデータベースが停止していることを確認してください。スタンバイサーバ上のデータベースが起動している場合は、以下の手順で停止してください。

```
(以下、postgresユーザで実行)  
$ /usr/pgsql-11/bin/pg_ctl stop -D /var/lib/pgsql/11/data
```

pg\_basebackup コマンドを使用してプライマリサーバのデータベースをスタンバイサーバに複製することで、スタンバイサーバ上のデータベースを最新化します。

スタンバイサーバのpostgres ユーザで、以下のコマンドを実行します。

```
(以下、postgresユーザで実行)  
$ mv /var/lib/pgsql/11/data /var/lib/pgsql/11/data.old # 既存データをリネームして退避  
$ /usr/pgsql-11/bin/pg_basebackup -h 192.168.1.10 -U postgres -D /var/lib/pgsql/11/data -c fast -X  
stream -P  
Password: XXXXXX
```

※太字斜体の部分にパスワードを指定します。

複製終了後、スタンバイサーバで、スタンバイデータベースを開始します。

```
(以下、postgresユーザで実行)  
$ /usr/pgsql-11/bin/pg_ctl start -D /var/lib/pgsql/11/data
```