

CLUSTERPRO[®] X *for Linux*

PPガイド(PostgreSQL)

2022/06/01

第3版

CLUSTERPRO

改版履歴

版数	改版日付	内容
1	2020/07/31	新規作成
2	2021/09/24	CLUSTERPRO X4.3、PostgreSQL 13 対応
3	2022/06/01	CLUSTERPRO X5.0、PostgreSQL 14 対応

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、一切責任をおいせん。

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は米国及びその他の国におけるLinus Torvaldsの登録商標です。

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
お問い合わせ先	i
適応範囲	i
関連マニュアル	ii
本書の表記規則	iii
第 1 章 PostgreSQL	1
機能概要	1
CLUSTERPRO 環境下での PostgreSQL の設定	3
動作環境	3
構築の流れ	4
片方向スタンバイでの構築手順	5
双方向スタンバイでの構築手順	15
付録 A サンプルスクリプト	27
start.sh	27
stop.sh	29

はじめに

本書は、PostgreSQLとCLUSTERPROを使用した環境の構築手順や設定例を紹介します。

対象読者と目的

『CLUSTERPRO X 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、およびユーザー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。ここでご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。

お問い合わせ先

本書の記述についてのお問い合わせは、以下窓口までお願いいたします。

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

本書のPostgreSQLに関する記載内容のお問い合わせには、原則としてCLUSTERPROのサポート契約とPostgreSQLのサポート契約が必要です。

https://jpn.nec.com/oss/middle_support/postgresql/maintenance1.html

適応範囲

本書は、以下の製品を対象としています。

CLUSTERPRO X 5.0 for Linux

関連マニュアル

本書の利用にあたっては、必要に応じて以下のマニュアルを参照してください。

1. CLUSTERPROマニュアル

CLUSTERPROのマニュアルは、以下の5つに分類されます。

『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。『CLUSTERPRO X 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。『CLUSTERPRO X インストール&設定ガイド』を補完する役割を持ちます。

CLUSTERPROマニュアルに関しては、以下を参照してください。

『CLUSTERPRO X システム構築ガイド』

<https://jpn.nec.com/clusterpro/clpx/manual.html>

2. PostgreSQLマニュアル

PostgreSQLの詳細については、PostgreSQLコミュニティが公開しているマニュアルを参照してください。

『PostgreSQLマニュアル』(English)

<https://www.postgresql.org/docs/manuals/>

『PostgreSQLマニュアル』(日本語)

<https://www.postgresql.jp/document/>

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
[] 角カッコ	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[] 角カッコ	カッコ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 hostname]</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 太文字(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。 <code>clpcl -s -a</code>
モノスペースフォント 斜体(courier)	ユーザが有効な値に置き換えて入力する項目	<code>clpstat -s [-h hostname]</code>

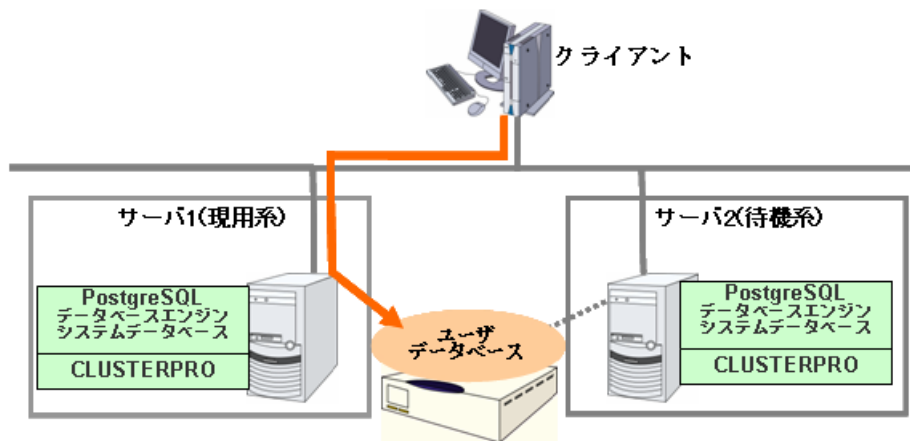
第 1 章 PostgreSQL

機能概要

PostgreSQL を、CLUSTERPRO X 4.0 以降の環境下で利用する際の機能概要について以下に記述します。CLUSTERPRO 環境下での PostgreSQL の運用は、片方向スタンバイ型と双方向スタンバイ型があります。クライアントは、通常、ODBC などを使用して現用系にアクセスします。現用系に障害が発生した場合、クライアントは待機系に接続し、運用することになります。(双方向スタンバイ型ではそれぞれが現用系、待機系となります。)

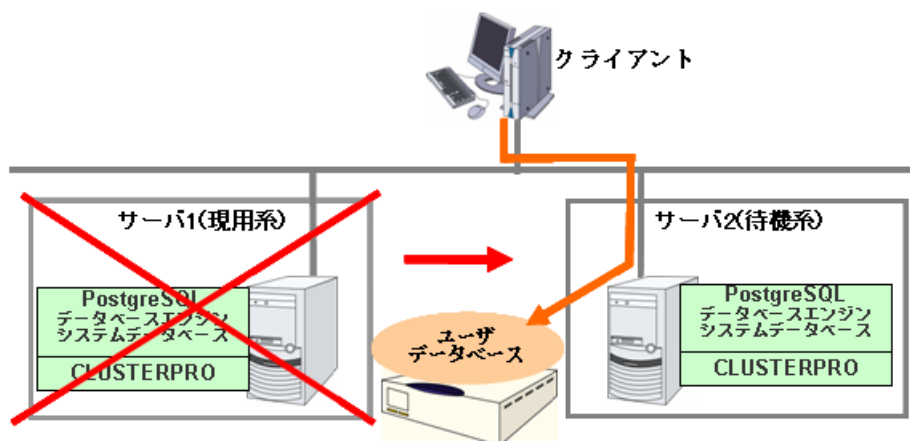
片方向スタンバイ

下図は、片方向スタンバイ型を CLUSTERPRO 環境下でサーバ 1 を現用系、サーバ 2 を待機系として動作させるときのイメージ図です。(イメージ図は共有ディスク型クラスタを想定したのですが、ミラーディスク型クラスタも構成可能です。)



クライアントマシンからは、CLUSTERPRO で設定するフローティング IP アドレスを使用して接続します。

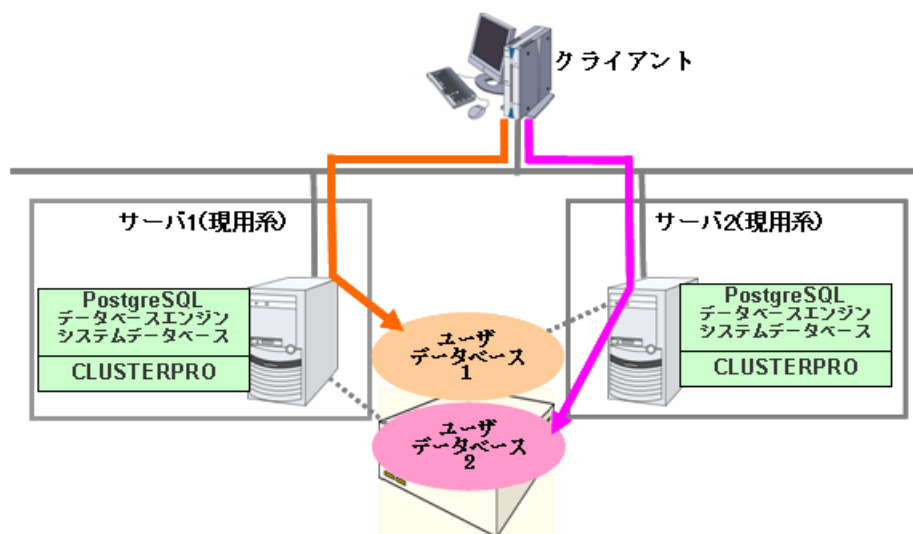
サーバ 1 に障害が発生すると以下の図のようになります。



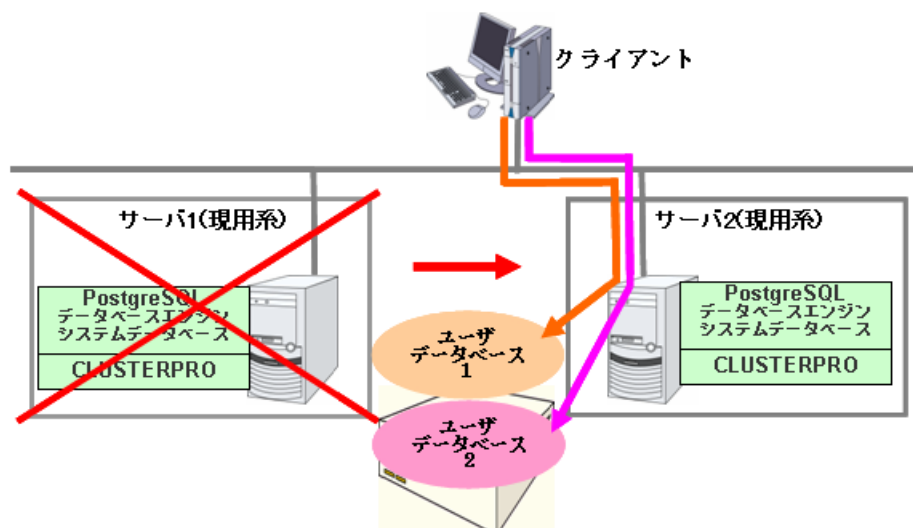
障害によりフェイルオーバーが完了すると、サーバ 2 で PostgreSQL のサービスが立ち上がり、ユーザデータベースのデータがサーバ 2 に移行する為、クライアントはサーバ 2 へ接続し、運用します。フェイルオーバーにてフローティング IP アドレスがサーバ 2 へ移行するため、クライアントはサーバが切り替わったことを意識せずに、同一のフローティング IP アドレスで再接続することで運用が可能です。

双方向スタンバイ

下図は、双方向スタンバイ型を CLUSTERPRO 環境下で動作させるときのイメージです。(イメージ図は共有ディスク型クラスタを想定したものです、ミラーディスク型クラスタも構成可能です。)



サーバ 1 で障害が発生し、フェイルオーバーが発生すると、ユーザデータベース 1 の資源がサーバ 2 に移行します。サーバ 1(ユーザデータベース 1)にアクセスしていたクライアントは、サーバ 2 へ接続し、運用することになります。フェイルオーバーによってフローティング IP アドレスがサーバ 2 へ移行する為、クライアントはサーバが切り替わったことを意識せずに、同一の IP アドレスで再接続することで運用が可能です。



CLUSTERPRO 環境下での PostgreSQL の設定

CLUSTERPRO 環境下で PostgreSQL を設定する場合、非クラスタ環境の場合と以下の点が異なりますのでご注意ください。

- ◆ ユーザデータベースは、必ず CLUSTERPRO で管理する共有ディスクまたはミラー用ディスクに格納する必要があります。データベースクラスタの作成やデータベースの作成は、現用系サーバからのみ行います。

動作環境

下記の PostgreSQL、CLUSTERPRO、OS のバージョンの組み合わせで動作確認を行っております。これ以外の組み合わせについては、お客様にて十分に動作確認を行ってください。

PostgreSQL のバージョン

PostgreSQL 14.3

※PostgreSQL 14.0～14.3 にはインデックス破損の不具合があるため、NEC がサポートする PostgreSQL 14.4 以降をご利用ください。

CLUSTERPRO のバージョン

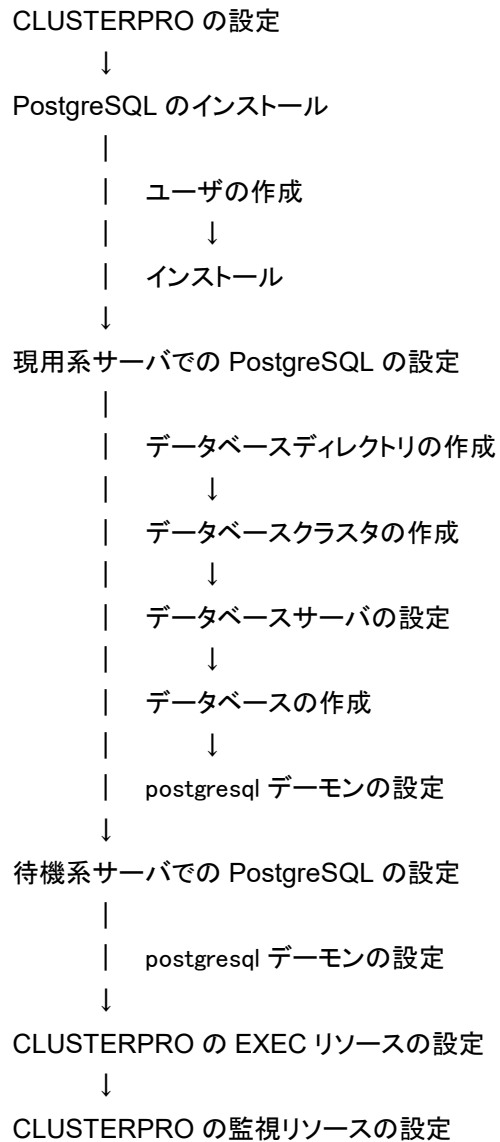
CLUSTERPRO X 5.0 for Linux

OS の種類

RHEL 8.4

構築の流れ

PostgreSQL の CLUSTERPRO 環境での構築は、以下の手順で行います。



片方向スタンバイでの構築手順

本章では、以下のような 2 ノード構成のクラスタでの片方向スタンバイ環境を想定し、説明を行います。

クラスタ環境

	サーバ 1(現用系)	サーバ 2(待機系)
サーバ名	Server1	Server2
実 IP アドレス	192.168.1.1	192.168.1.2
ディスクハートビート用パーティション(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	

フェイルオーバーグループ情報

グループ名	グループ 1	
フェイルオーバーポリシー	Server1 → Server2	
フローティング IP アドレス	192.168.1.10	
ディスクリソース (ミラーディスクリソース)	デバイス名	/dev/sdb2
	マウントポイント	/mnt/sdb2

PostgreSQL 情報

PostgreSQL のバージョン	14.3
インストールディレクトリ	/usr/pgsql-14
管理用のユーザ	postgres
データベースディレクトリ	/mnt/sdb2/pgsql/data
データベース名	db1
文字コード	UTF8
使用ポート番号	5432
クライアント認証方式	md5

CLUSTERPRO の設定

CLUSTERPRO の Builder または Cluster WebUI を使用して、PostgreSQL の運用に使用するフェイルオーバーグループを作成します。フェイルオーバーグループには以下のリソースが必要となります。

- ◆ フローティング IP リソース
- ◆ ディスクリソース または ミラーディスクリソース

上記のリソースを追加する手順については、CLUSTERPRO X 5.0 for Linux インストール&設定ガイド「第 6 章 クラスタ構成情報を作成する」をご参照ください。

フェイルオーバーグループを作成し、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た後、フェイルオーバーグループを現用系サーバで起動します。

PostgreSQL のインストール

両サーバにて、PostgreSQL のインストールを実施します。Linux OS にバンドルされている PostgreSQL をインストールしていた場合、ユーザの作成は不要となりますが、インストールされた PostgreSQL はアンインストールし、PostgreSQL 用のユーザの確認を実施した後に手順を進めてください。

インストールについての詳細は、PostgreSQL のオンラインマニュアルを参照してください。

ユーザの作成

データベース管理用ユーザ(postgres)を、現用系／待機系サーバでユーザ ID が等しくなるように作成します。

```
# useradd -u 26 postgres
```

既に postgres ユーザが存在する場合、以下のコマンドを実施して両サーバの postgres ユーザが同じユーザ ID で設定されているか確認します。

```
# id postgres
```

uid、gid、groups の情報が等しければ、postgres ユーザの再作成は必要ありません。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。

postgres ユーザのパスワードを両サーバで設定します。

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

※太斜字の部分にパスワードを指定します。

インストール

RPM ファイルを使用したインストール方法

下記 URL から RPM ファイルをダウンロードしてください。

URL: <https://yum.postgresql.org/rpmchart.php>

PostgreSQL の RPM ファイルをサーバの任意の場所に格納し、インストールを行います。ここでは入手した PostgreSQL の下記 RPM ファイルをサーバの/tmp に pgsql-14 ディレクトリを作成し、格納しているものとして記述します。

格納する RPM ファイル

- postgresql14-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-contrib-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-libs-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-server-14.3-1PGDG.rhel8.x86_64.rpm

下記コマンドを実行しインストールしてください。

```
# rpm -ivh /tmp/pgsql-14/postgresql14-*
```

ソースを使用したインストール方法

下記 URL からソースをダウンロードしてください。

URL: <https://www.postgresql.org/ftp/source/>

PostgreSQL のソースをサーバの任意の場所に展開し、コンパイルおよびインストールを行います。ここでは入手した PostgreSQL のソース「postgresql-14.3.tar.gz」をサーバの/tmp にコピーしているものとして記述します。

※本手順書は rpm によるインストールを基準に記載しているため、ソースを使用してインストールした場合はインストール先に合わせてパスを置き換えてください。

```
# cd /tmp
# tar -zxvf postgresql-14.3.tar.gz
# cd postgresql14.3
# ./configure --prefix=/usr/local/pgsql-14
# make all
# make install
# chown -R postgres:postgres /usr/local/pgsql-14
```

現用系サーバでの PostgreSQL の設定

ここからはファイルオーバグループが現用系サーバにある状態で行います。

データベースディレクトリの作成

データベースを格納するディレクトリを作成し、所有者を postgres ユーザに設定します。
クラスタで管理するデータパーティション配下に格納する必要があります。

root ユーザで以下のコマンドを実施します。

```
# mkdir -p /mnt/sdb2/pgsql/data
# chown -R postgres:postgres /mnt/sdb2/pgsql/data
# chmod 700 /mnt/sdb2/pgsql/data
```

データベースクラスタの作成

initdb コマンドを使用して、前項で作成したデータベースディレクトリ配下にデータベースクラスタを作成します。

postgres ユーザで以下のコマンドを実施します。文字コード(-E パラメータ)の情報などは、環境に合わせて指定してください。

```
$ /usr/pgsql-14/bin/initdb -D /mnt/sdb2/pgsql/data -E UTF8 --no-locale -W
```

下記のようにデータベーススーパーユーザ postgres のパスワードの入力が求められるため、任意のパスワードを入力します。

```
...(略)...
Enter new superuser password: xxxxxx
Enter it again: xxxxxx
...(略)...
```

※太斜字の部分にパスワードを指定します。

下記のように成功した旨のメッセージが表示されます。

```
Success. You can now start the database server using:

/usr/pgsql-14/bin/pg_ctl -D /mnt/sdb2/pgsql/data -l logfile start
```


データベースサーバの設定

postgresql.conf ファイル/pg_hba.conf ファイルを編集して、データベースサーバの設定を行います。両ファイルは、データベースディレクトリの直下にあります。本章の例の場合、

/mnt/sdb2/pgsql/data/postgresql.conf

/mnt/sdb2/pgsql/data/pg_hba.conf

になります。

◆ postgresql.conf

postgresql.conf ファイルでは、接続の監視に使用する IP インタフェースの設定や PostgreSQL が使用するポート番号の設定を行います。

クライアントからの接続の監視に使用する IP インタフェースの設定を、利用可能なすべての IP インタ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。

```
listen_addresses = '*'
```

PostgreSQL が使用するポート番号を設定します。[port]エントリのコメントアウトを削除し、既定値(5432)から変更する場合は、[port]エントリの設定値を変更します。

```
port = ポート番号
```

本章の例では、既定値を使用するため、port の変更は行いません。

```

postgres@node1:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                                # write an extra PID file
                                                         # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                                # what IP address(es) to listen on;
                                                         # comma-separated list of addresses;
                                                         # defaults to 'localhost'; use '*' for a
ll
                                                         # (change requires restart)
port = 5432                                           # (change requires restart)
max_connections = 100                                # (change requires restart)
#superuser_reserved_connections = 3                  # (change requires restart)
#unix_socket_directories = '/var/run/postgresql, /tmp' # comma-separated list o
f directories
                                                         # (change requires restart)
#unix_socket_group = ''                              # (change requires restart)

```

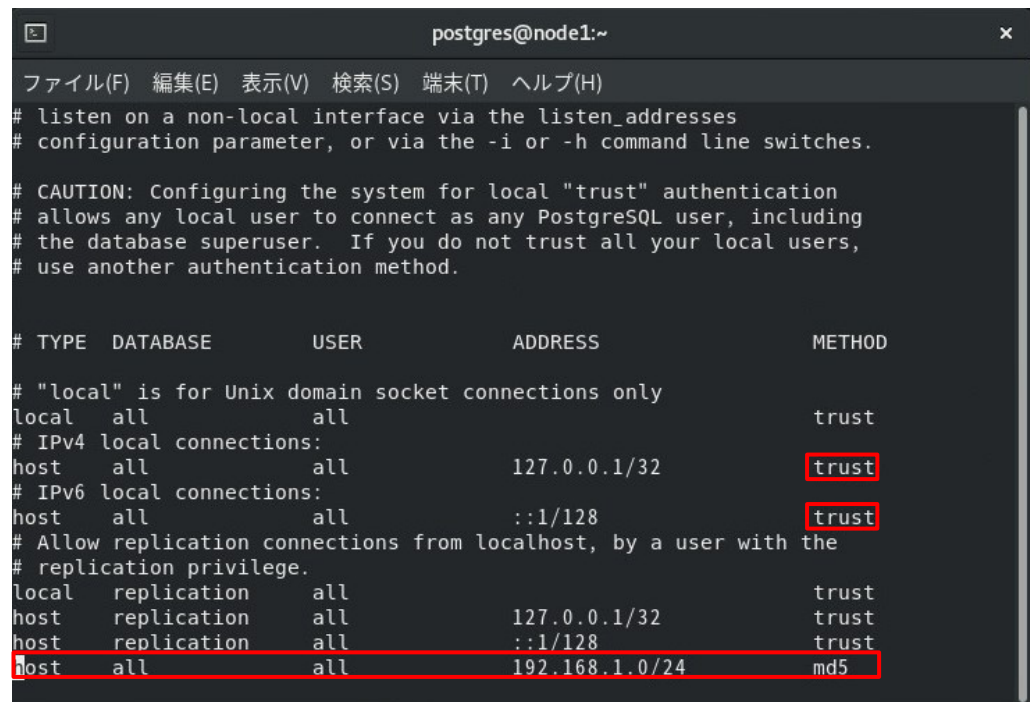
◆ pg_hba.conf

pg_hba.conf ファイルでは、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あります。ここでは、192.168.1.0/24(使用する ip アドレスによるため注意)からの接続を許可する設定を行います。以下の一行を追加します。

host	all	all	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。

また、本書では localhost(127.0.0.1/32)接続が trust に設定されていることを前提に記載されています。peer や ident に設定されている場合は trust に変更してください。



```
postgres@node1:~  
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)  
# listen on a non-local interface via the listen_addresses  
# configuration parameter, or via the -i or -h command line switches.  
  
# CAUTION: Configuring the system for local "trust" authentication  
# allows any local user to connect as any PostgreSQL user, including  
# the database superuser.  If you do not trust all your local users,  
# use another authentication method.  
  
# TYPE  DATABASE        USER            ADDRESS           METHOD  
# "local" is for Unix domain socket connections only  
local   all             all                                     trust  
# IPv4 local connections:  
host    all             all             127.0.0.1/32      trust  
# IPv6 local connections:  
host    all             all             ::1/128           trust  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local   replication    all             trust  
host    replication    all             127.0.0.1/32      trust  
host    replication    all             ::1/128           trust  
host    all             all             192.168.1.0/24    md5
```

データベースの作成

pg_ctl コマンドを使用してデータベースサーバを起動した後、createdb コマンドを使用してデータベースの作成を行います。

postgres ユーザで以下を実施します。

```
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdb2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/createdb -h localhost -U postgres db1
```

※下線の部分は、データベース名を指定します。

(pg_hba.conf の設定によってはユーザ認証のエラーにより、createdb できないことがあります。ユーザ認証のエラーが発生した場合は「[データベースサーバの設定](#)」の pg_hba.conf を参考に再設定後、データベースサーバを再起動する必要があります。)

データベースの作成が完了したら、データベースサーバを停止します。

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdb2/pgsql/data -m fast
```

postgresql デモンの設定

unit ファイルをインストールしていない環境では、本手順は不要です。

postgresql デモンの自動起動設定が有効になっている場合は、無効にしてください。

root ユーザで以下を実行します。

```
# systemctl list-unit-files postgresql-14.service
```

上記を実施した結果、STATE が enabled に設定されている場合は、以下を実行します。

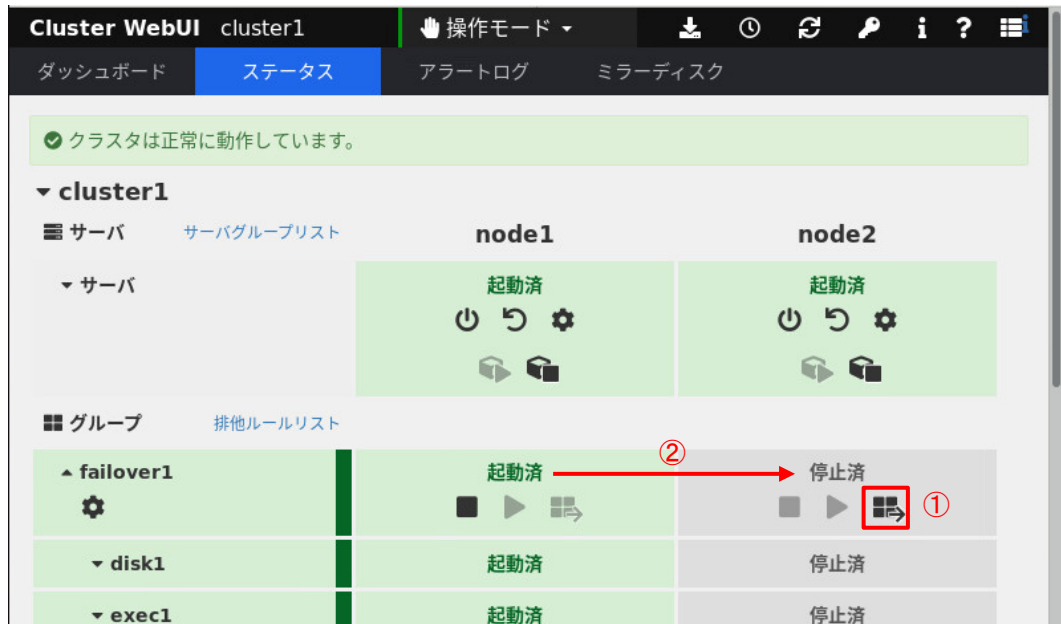
```
# systemctl disable postgresql-14.service
```

再度“systemctl list-unit-files postgresql-14.service”を実施して、STATE が disable に設定されたことを確認してください。

これで現用系サーバでの PostgreSQL 設定は終了です。

待機系サーバでの PostgreSQL の設定

フェイルオーバーグループを待機系サーバに移動します。待機系サーバでの PostgreSQL の設定を行います。



postgresql デーモンの設定

「現用系サーバでの PostgreSQL の設定」の「[postgresql デーモンの設定](#)」と同じ操作を実施してください。

設定が終わったら、以下のような手順でデータベースサーバを起動・データベース接続を実施し、現用系サーバで作成したデータベースへ接続できることを確認してください。

```
# su - postgres
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdb2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/psql -h localhost -U postgres db1
```

※下線の部分は、データベース名を指定します。

確認が完了したら、データベースサーバを停止します。

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdb2/pgsql/data -m fast
```

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder または Cluster WebUI を使用して、PostgreSQL の起動・停止用スクリプトを実行する EXEC リソースを追加します。

- (1) CLUSTERPRO の Builder のツリービューのグループ名のアイコンを右クリックし、[リソースの追加]を選択、または Cluster WebUI の設定モードにてグループ名の横の[+]アイコンを選択します。
- (2) [タイプ]には、[execute resource]を指定し、リソース名を設定して[次へ]を選択します。
- (3) 依存するリソースにデータベースディレクトリを作成したディスクリソースを追加します。
- (4) [この製品で作成したスクリプト]を選択し、start.sh および stop.sh を選択してスクリプトの内容を編集します。スクリプト内容については「[付録 A サンプルスクリプト](#)」のように記述します。サンプルスクリプト冒頭の環境変数の設定値(アンダーラインの箇所)を以下のように編集します。スクリプトの編集ができれば、画面に従って EXEC リソースの追加を実施してください。

start.sh の設定

環境変数名	設定値	説明
SUUSER	"postgres"	PostgreSQL の OS ユーザ名
PGINST	"/usr/pgsql-14"	PostgreSQL のインストールディレクトリ
PGDATA	"/mnt/sdb2/pgsql/data"	PostgreSQL のデータベースディレクトリ
PGPORT	"5432"	PostgreSQL のポート番号

stop.sh の設定

環境変数名	設定値	説明
SUUSER	"postgres"	PostgreSQL の OS ユーザ名
PGINST	"/usr/pgsql-14"	PostgreSQL のインストールディレクトリ
PGDATA	"/mnt/sdb2/pgsql/data"	PostgreSQL のデータベースディレクトリ

CLUSTERPRO の監視リソースの設定

注：本監視リソースを設定するには CLUSTERPRO X Database Agent 5.0 for Linux が必要です。

CLUSTERPRO の Builder または Cluster WebUI を使用して、監視リソースを追加します。

CLUSTERPRO の Builder のツリービューの[Monitors]を右クリックし、[モニタリソースの追加]を選択、または Cluster WebUI の設定モードにて[モニタ]の横の[+]アイコンを選択して[モニタリソースの定義]画面を表示します。[PostgreSQL monitor]を選択します。

PostgreSQL 監視リソースの詳細は、CLUSTERPRO X 5.0 for Linux リファレンスガイド「第 4 章 モニタリソースの詳細」の「PostgreSQL モニタリソースを理解する」をご参照ください。

PostgreSQL 監視リソースの設定

	データベース 1
データベース名	db1
IP アドレス	127.0.0.1
ポート番号	5432
ユーザ名	postgres
パスワード	postgres ユーザのパスワード
監視テーブル名	psqlwatch
パス	/usr/pgsql-14/lib/libpq.so.5.14
対象リソース	PostgreSQL の EXEC リソース
監視レベル	レベル 2(update/select での監視)
回復対象	failover1

双方向スタンバイでの構築手順

本章では、以下のような 2 ノード構成のクラスタでの双方向スタンバイ環境を想定し、説明を行います。

クラスタ環境

	サーバ 1(現用系)	サーバ 2(現用系)
サーバ名	Server1	Server2
実 IP アドレス	192.168.1.1	192.168.1.2
ディスクハートビート用パーティション(クラスタパーティション)	/dev/sdb1, /dev/sdc1	
データパーティション	/dev/sdb2, /dev/sdc2	

フェイルオーバーグループ情報

グループ名	グループ 1	
フェイルオーバーポリシー	Server1 → Server2	
フローティング IP アドレス	192.168.1.10	
ディスクリソース (ミラーディスクリソース)	デバイス名	/dev/sdb2
	マウントポイント	/mnt/sdb2
グループ名	グループ 2	
フェイルオーバーポリシー	Server2 → Server1	
フローティング IP アドレス	192.168.1.11	
ディスクリソース (ミラーディスクリソース)	デバイス名	/dev/sdc2
	マウントポイント	/mnt/sdc2

PostgreSQL 情報

PostgreSQL のバージョン	14.3	
インストールディレクトリ	/usr/pgsql-14	
管理用のユーザ	postgres	
ユーザデータベース情報 1	データベースディレクトリ	/mnt/sdb2/pgsql/data
	データベース名	db1
	文字コード	UTF8
	使用ポート番号	5432
	クライアント認証方式	md5
ユーザデータベース情報 2	データベースディレクトリ	/mnt/sdc2/pgsql/data
	データベース名	db2
	文字コード	UTF8
	使用ポート番号	5433
	クライアント認証方式	md5

CLUSTERPRO の設定

CLUSTERPRO の Builder または Cluster WebUI を使用して、PostgreSQL の運用に使用するフェイルオーバーグループを二つ作成します。フェイルオーバーグループには以下のリソースが必要となります。

- ◆ フローティング IP リソース
- ◆ ディスクリソース または ミラーディスクリソース

上記のリソースを追加する手順については、CLUSTERPRO X 5.0 for Linux インストール&設定ガイド「第 6 章 クラスタ構成情報を作成する」をご参照ください。

フェイルオーバーグループを作成し、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た後、グループ 1 を Server1 で、グループ 2 を Server2 で起動します。

PostgreSQL のインストール

両サーバにて、PostgreSQL のインストールを実施します。Linux OS にバンドルされている PostgreSQL をインストールしていた場合、ユーザの作成は不要となりますが、インストールされた PostgreSQL はアンインストールし、PostgreSQL 用のユーザの確認を実施した後に手順を進めてください。

インストールについての詳細は、PostgreSQL のオンラインマニュアルを参照してください。

ユーザの作成

データベース管理用ユーザ(postgres)を、現用系／待機系サーバでユーザ ID が等しくなるように作成します。

```
# useradd -u 26 postgres
```

既に postgres ユーザが存在する場合、以下のコマンドを実施して両サーバの postgres ユーザが同じユーザ ID で設定されているか確認します。

```
# id postgres
```

uid、gid、groups の情報が等しければ、postgres ユーザの再作成は必要ありません。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。

postgres ユーザのパスワードを両サーバで設定します。

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

※太斜字の部分にパスワードを指定します。

インストール

RPM ファイルを使用したインストール方法

下記 URL から RPM ファイルをダウンロードしてください。

URL: <https://yum.postgresql.org/rpmchart.php>

PostgreSQL の RPM ファイルをサーバの任意の場所に格納し、インストールを行います。ここでは入手した PostgreSQL の下記 RPM ファイルをサーバの/tmp に pgsql-14 ディレクトリを作成し、格納しているものとして記述します。

格納する RPM ファイル

- postgresql14-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-contrib-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-libs-14.3-1PGDG.rhel8.x86_64.rpm
- postgresql14-server-14.3-1PGDG.rhel8.x86_64.rpm

下記コマンドを実行しインストールしてください。

```
# rpm -ivh /tmp/pgsql-14/postgresql14-*
```

ソースを使用したインストール方法

下記 URL からソースをダウンロードしてください。

URL: <https://www.postgresql.org/ftp/source/>

PostgreSQL のソースをサーバの任意の場所に展開し、コンパイルおよびインストールを行います。ここでは入手した PostgreSQL のソース「postgresql-14.3.tar.gz」をサーバの/tmp にコピーしているものとして記述します。

※本手順書は rpm によるインストールを基準に記載しているため、ソースを使用してインストールした場合はインストール先に合わせてパスを置き換えてください。

```
# cd /tmp
# tar -zxvf postgresql-14.3.tar.gz
# cd postgresql14.3
# ./configure --prefix=/usr/local/pgsql-14
# make all
# make install
# chown -R postgres:postgres /usr/local/pgsql-14
```

現用系サーバでの PostgreSQL の設定

本章の例の場合、グループ 1 の現用系サーバは Server1、待機系サーバは Server2 となります。また、グループ 2 の現用系サーバは Server2、待機系サーバは Server1 となります。ここでは、Server1 でグループ 1 の設定、Server2 でグループ 2 の設定を行います。

データベースディレクトリの作成

データベースを格納するディレクトリを作成し、所有者を postgres ユーザに設定します。クラスタで管理するデータパーティション配下に格納する必要があります。

root ユーザで以下のコマンドを実施します。

◆ Server1 の場合

```
# mkdir -p /mnt/sdb2/pgsql/data
# chown -R postgres:postgres /mnt/sdb2/pgsql/data
# chmod 700 /mnt/sdb2/pgsql/data
```

◆ Server2 の場合

```
# mkdir -p /mnt/sdc2/pgsql/data
# chown -R postgres:postgres /mnt/sdc2/pgsql/data
# chmod 700 /mnt/sdc2/pgsql/data
```

データベースクラスタの作成

initdb コマンドを使用して、前項で作成したデータベースディレクトリ配下にデータベースクラスタを作成します。

postgres ユーザで以下のコマンドを実施します。文字コード(-E パラメータ)の情報などは、環境に合わせて指定してください。

◆ Server1 の場合

```
$ /usr/pgsql-14/bin/initdb -D /mnt/sd2/pgsql/data -E UTF8 --no-locale -W
```

◆ Server2 の場合

```
$ /usr/pgsql-14/bin/initdb -D /mnt/sdc2/pgsql/data -E UTF8 --no-locale -W
```

下記のようにデータベーススーパーユーザ postgres のパスワードの入力が求められるため、任意のパスワードを入力します。

```
...(略)...
Enter new superuser password: xxxxxx
Enter it again: xxxxxx
...(略)...
```

※太斜字の部分にパスワードを指定します。

下記のように成功した旨のメッセージが表示されます。

```
Success. You can now start the database server using:

/usr/pgsql-14/bin/pg_ctl -D /mnt/sdb2/pgsql/data -l logfile start
```

※太斜字の箇所は、データベースディレクトリのパスにより表示される内容が変わります。

データベースサーバの設定

postgresql.conf ファイル/pg_hba.conf ファイルを編集して、データベースサーバの設定を行います。両ファイルは、データベースディレクトリの直下にあります。本章の例の場合、以下になります。

【Server1】

/mnt/sdb2/pgsql/data/postgresql.conf

/mnt/sdb2/pgsql/data/pg_hba.conf

【Server2】

/mnt/sdc2/pgsql/data/postgresql.conf

/mnt/sdc2/pgsql/data/pg_hba.conf

◆ postgresql.conf

postgresql.conf ファイルでは、接続の監視に使用する IP インタフェースの設定や PostgreSQL が使用するポート番号の設定を行います。

クライアントからの接続の監視に使用する IP インタフェースの設定を、利用可能なすべての IP インタ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。

```
listen_addresses = '*'
```

PostgreSQL が使用するポート番号を設定します。[port]エントリのコメントアウトを削除し、既定値(5432)から変更する場合は、[port]エントリの設定値を変更します。

```
port = ポート番号
```

本章の例では、Server1 では port の変更を行いません。Server2 では port の値を 5433 に変更します。

```
postgres@node1:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                                # write an extra PID file
                                                         # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                                # what IP address(es) to listen on;
                                                         # comma-separated list of addresses;
                                                         # defaults to 'localhost'; use '*' for a
ll
                                                         # (change requires restart)
port = 5432                                           # (change requires restart)
max_connections = 100                                # (change requires restart)
#superuser_reserved_connections = 3                  # (change requires restart)
#unix_socket_directories = '/var/run/postgresql, /tmp' # comma-separated list o
f directories
                                                         # (change requires restart)
#unix_socket_group = ''                              # (change requires restart)
```

◆ pg_hba.conf

pg_hba.conf ファイルでは、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あります。ここでは、192.168.1.0/24(使用する ip アドレスによるため注意)からの接続を許可する設定を行います。以下の一行を追加します。

host	all	all	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。

また、本書では localhost(127.0.0.1/32)接続が trust に設定されていることを前提に記載されています。peer や ident に設定されている場合は trust に変更してください。

```

postgres@node1:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# CAUTION: Configuring the system for local "trust" authentication
# allows any local user to connect as any PostgreSQL user, including
# the database superuser.  If you do not trust all your local users,
# use another authentication method.

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
host all all 192.168.1.0/24 md5

```

データベースの作成

pg_ctl コマンドを使用してデータベースサーバを起動した後、createdb コマンドを使用してデータベースの作成を行います。

postgres ユーザで以下を実施します。

◆ Server1 の場合

```
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdb2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/createdb -h localhost -p 5432 -U postgres db1
```

◆ Server2 の場合

```
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdc2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/createdb -h localhost -p 5433 -U postgres db2
```

※下線の部分は、データベース名を指定します。

(pg_hba.conf の設定によってはユーザ認証のエラーにより、createdb できないことがあります。ユーザ認証のエラーが発生した場合は「[データベースサーバの設定](#)」の pg_hba.conf を参考に再設定後、データベースサーバを再起動する必要があります。)

データベースの作成が完了したら、データベースサーバを停止します。

◆ Server1 の場合

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdb2/pgsql/data -m fast
```

◆ Server2 の場合

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdc2/pgsql/data -m fast
```

postgresql デーモンの設定

unit ファイルをインストールしていない環境では、本手順は不要です。

postgresql デーモンの自動起動設定が有効になっている場合は、無効にしてください。

root ユーザで以下を実行します。

```
# systemctl list-unit-files postgresql-14.service
```

上記を実施した結果、STATE が enabled に設定されている場合は、以下を実行します。

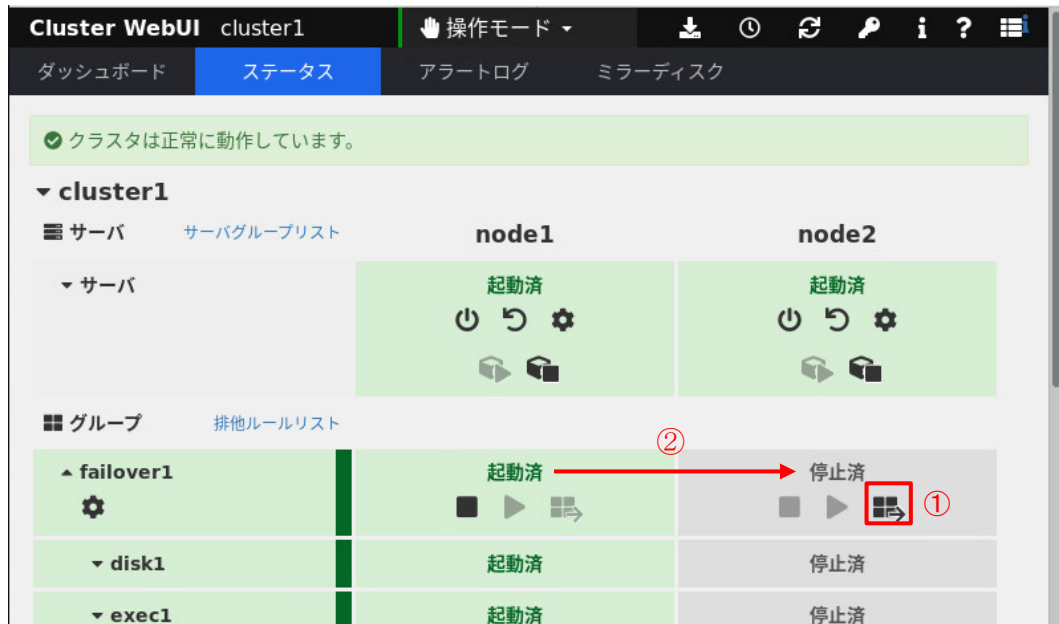
```
# systemctl disable postgresql-14.service
```

再度"**systemctl list-unit-files postgresql-14.service**"を実施して、STATE が disable に設定されたことを確認してください。

これで現用系サーバでの PostgreSQL 設定は終了です。

待機系サーバでの PostgreSQL の設定

グループ 1 を Server1 から Server2、グループ 2 を Server2 から Server1 に移動します。待機系サーバでの PostgreSQL の設定を行います。



postgresql デーモンの設定

「現用系サーバでの PostgreSQL の設定」の「[postgresql デーモンの設定](#)」と同じ操作を実施してください。すでにサーバで postgres デーモンの設定を実施済みの場合は実施する必要はありません。

設定が終わったら、データベースサーバを起動・データベース接続を実施し、現用系サーバで作成したデータベースへ接続できることを確認してください。

◆ Server1 の場合

```
# su - postgres
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdc2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/psql -h localhost -p 5433 -U postgres db2
```

◆ Server2 の場合

```
# su - postgres
$ /usr/pgsql-14/bin/pg_ctl start -D /mnt/sdb2/pgsql/data -l /dev/null
$ /usr/pgsql-14/bin/psql -h localhost -p 5432 -U postgres db1
```

※下線の部分は、データベース名を指定します。

確認が完了したら、データベースサーバを停止します。

◆ Server1 の場合

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdc2/pgsql/data -m fast
```

◆ Server2 の場合

```
$ /usr/pgsql-14/bin/pg_ctl stop -D /mnt/sdb2/pgsql/data -m fast
```


CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder または Cluster WebUI を使用して、PostgreSQL の起動・停止用スクリプトを実行する EXEC リソースを追加します。

- (1) CLUSTERPRO の Builder のツリービューのグループ名のアイコンを右クリックし、[リソースの追加]を選択、または Cluster WebUI の設定モードにてグループ名の横の[+]アイコンを選択します。
- (2) [タイプ]には、[execute resource]を指定し、リソース名を設定して[次へ]を選択します。
- (3) 依存するリソースにデータベースディレクトリを作成したディスクリソースを追加します。
- (4) [この製品で作成したスクリプト]を選択し、start.sh および stop.sh を選択してスクリプトの内容を編集します。スクリプト内容については「[付録 A サンプルスクリプト](#)」のように記述します。サンプルスクリプト冒頭の環境変数の設定値(アンダーラインの箇所)を以下のように編集します。スクリプトの編集ができれば、画面に従って EXEC リソースの追加を実施してください。

start.sh の設定

環境変数名	設定値(グループ 1)	設定値(グループ 2)
SUUSER	“postgres”	“postgres”
PGINST	“/usr/pgsql-14”	“/usr/pgsql-14”
PGDATA	“/mnt/sdb2/pgsql/data”	“/mnt/sdc2/pgsql/data”
PGPORT	“5432”	“5433”

stop.sh の設定

環境変数名	設定値(グループ 1)	設定値(グループ 2)
SUUSER	“postgres”	“postgres”
PGINST	“/usr/pgsql-14”	“/usr/pgsql-14”
PGDATA	“/mnt/sdb2/pgsql/data”	“/mnt/sdc2/pgsql/data”

CLUSTERPRO の監視リソースの設定

注：本監視リソースを設定するには CLUSTERPRO X Database Agent 5.0 for Linux が必要です。

CLUSTERPRO の Builder または Cluster WebUI を使用して、監視リソースを追加します。

CLUSTERPRO の Builder のツリービューの[Monitors]を右クリックし、[モニタリソースの追加]を選択、または Cluster WebUI の設定モードにて[モニタ]の横の[+]アイコンを選択して[モニタリソースの定義]画面を表示します。[PostgreSQL monitor]を選択します。

PostgreSQL 監視リソースの詳細は、CLUSTERPRO X 5.0 for Linux リファレンスガイド「第 4 章 モニタリソースの詳細」の「PostgreSQL モニタリソースを理解する」をご参照ください。

PostgreSQL 監視リソースの設定

	データベース 1	データベース 2
データベース名	db1	db2
IP アドレス	127.0.0.1	127.0.0.1
ポート番号	5432	5433
ユーザ名	postgres	postgres
パスワード	postgres ユーザのパスワード	postgres ユーザのパスワード
監視テーブル名	psqlwatch	psqlwatch
パス	/usr/pgsql-14/lib/libpq.so.5.14	/usr/pgsql-14/lib/libpq.so.5.14
対象リソース	db1 の EXEC リソース	db2 の EXEC リソース
監視レベル	レベル 2(update/select での監視)	レベル 2(update/select での監視)
回復対象	failover1	failover2

付録 A サンプルスクリプト

PostgreSQL のクラスタを構築するために必要なスクリプトのサンプルです。スクリプト内でアンダーラインの箇所は、お使いになられる環境に合わせて適宜編集してください。

start.sh

PostgreSQL を起動するためのスクリプトです。アンダーラインの箇所を適宜編集してお使いください。

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

ulimit -s unlimited

#
# PostgreSQL's Parameters
#
SUUSER="postgres"                # PG user name(OS user)
PGINST="/usr/pgsql-14"          # PG Install directory
PGDATA="/mnt/sdb2/pgsql/data"   # Database directory
PGPORT="5432"                   # Database Port number

if [ -f ${PGDATA}/postmaster.pid ]
then
    rm -f ${PGDATA}/postmaster.pid
fi

if [ -f /tmp/.s.PGSQL. ${PGPORT} ]
then
    rm -f /tmp/.s.PGSQL. ${PGPORT}
fi

if [ -f /var/run/postgresql/.s.PGSQL. ${PGPORT} ]
then
    rm -f /var/run/postgresql/.s.PGSQL. ${PGPORT}
fi

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
```

```

        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PG Database start"

    su - ${SUUSER} -c "${PGINST}/bin/pg_ctl start -D ${PGDATA} -l /dev/null"
    else
        echo "ERROR_DISK from START"
        exit 1

    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
        date +"%Y/%m/%d %T"
        echo "PG Database start"

        su - ${SUUSER} -c "${PGINST}/bin/pg_ctl start -D ${PGDATA} -l /dev/null"
        else
            echo "ERROR_DISK from FAILOVER"
            exit 1

        fi
    else
        echo "NO_CLP"
        exit 1
    fi
echo "EXIT"
exit 0

```

stop.sh

PostgreSQL を停止するためのスクリプトです。アンダーラインの箇所を適宜編集してお使いください。

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

ulimit -s unlimited

#
# PostgreSQL's Parameters
#
SUUSER="postgres"                # PG user name(OS user)
PGINST="/usr/pgsql-14"          # PG Install directory
PGDATA="/mnt/sdb2/pgsql/data"  # Database directory

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +%Y/%m/%d %T
        echo "PG Database stop"

        su - ${SUUSER} -c "${PGINST}/bin/pg_ctl stop -D ${PGDATA} -m fast"
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    fi
fi
```

```
        fi
        date +"%Y/%m/%d %T"
        echo "PG Database stop"

    su - ${SUUSER} -c "${PGINST}/bin/pg_ctl stop -D ${PGDATA} -m fast"
    else
        echo "ERROR_DISK from FAILOVER"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0
```