

PowerGres on CLUSTERPRO[®] X for Linux

PowerGres Plus on CLUSTERPRO[®] X for Linux

PostgreSQL on CLUSTERPRO[®] X for Linux

HowTo

2007.06.30

第1版

改版履歴

版数	改版日付	内 容
1	2007/06/30	新規作成

目次

対象読者と目的	1
適用範囲	1
本書の構成	1
第 1 章 PowerGres	2
機能概要	2
CLUSTERPRO環境下でのPowerGresの設定	5
構築の流れ	5
片方向スタンバイでの構築手順	6
双方向スタンバイでの構築手順	18
第 2 章 PostgreSQL	31
機能概要	31
CLUSTERPRO環境下でのPostgreSQLの設定	34
構築の流れ	34
片方向スタンバイでの構築手順	35
双方向スタンバイでの構築手順	44
第 3 章 PowerGres Plus	55
機能概要	55
CLUSTERPRO環境下でのPowerGres Plusの設定	58
構築の流れ	58
片方向スタンバイでの構築手順	59
双方向スタンバイでの構築手順	69

はじめに

対象読者と目的

本書は、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。
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。

適用範囲

本書は、CLUSTERPRO X 1.0 for Linux を対象としています。

本書の構成

- 第 1 章 「PowerGres」:PowerGresについて説明します。
- 第 2 章 「PostgreSQL」:PostgreSQLについて説明します。
- 第 3 章 「PowerGres Plus」:PowerGres Plusについて説明します。

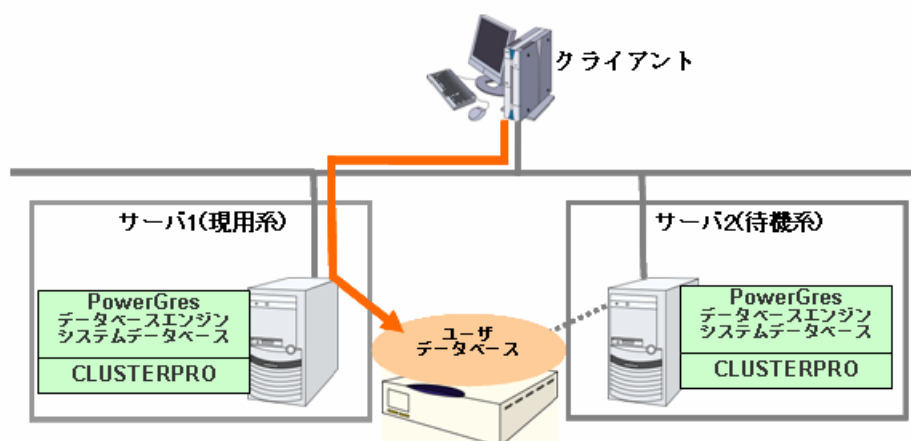
第 1 章 PowerGres

機能概要

PowerGres を、CLUSTERPRO X 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 PowerGres 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。）

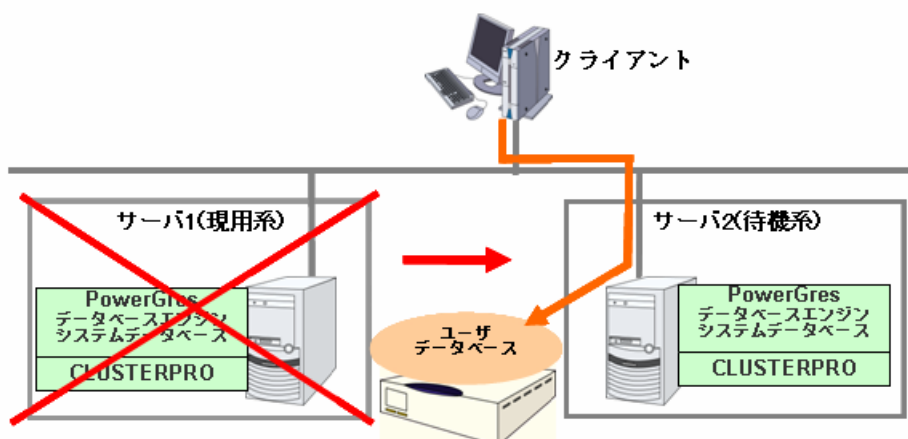
片方向スタンバイ

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。



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 アドレスや仮想コンピュータ名を使用して接続します。

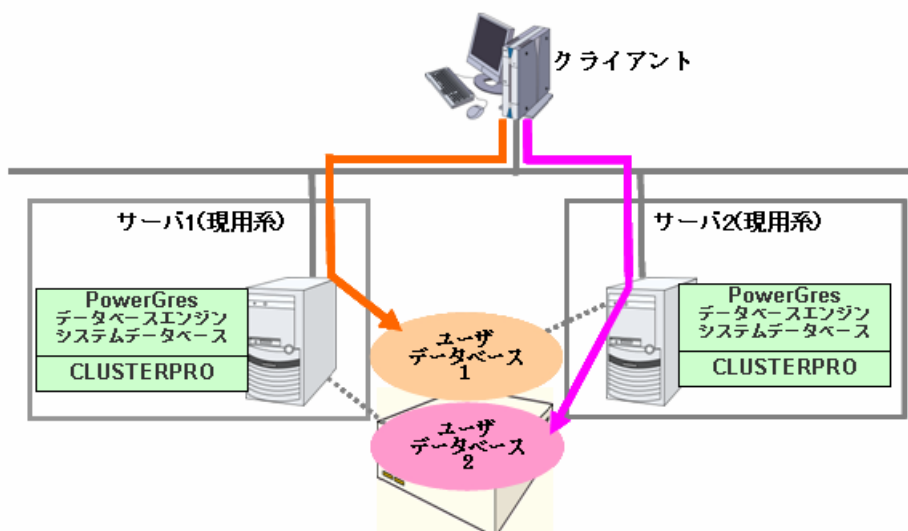
サーバ1に障害が発生すると以下の図のようになります。



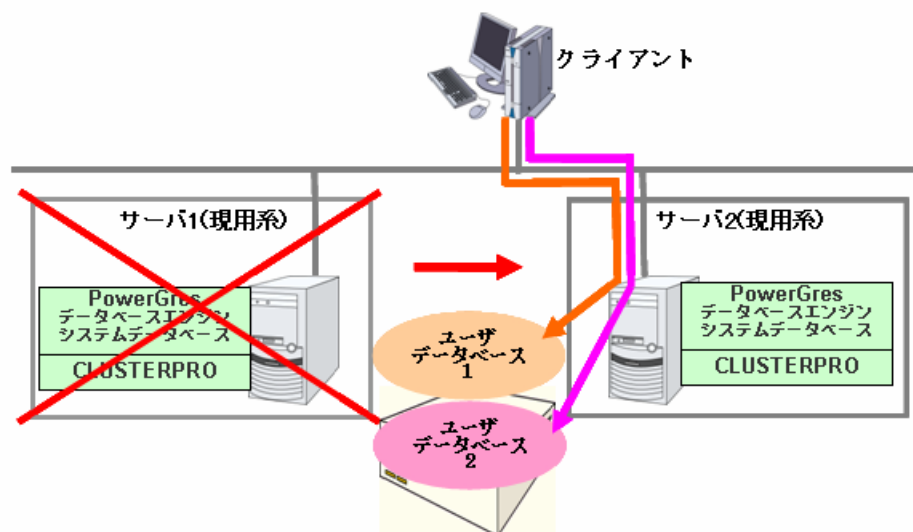
障害によりフェイルオーバーが完了すると、サーバ2でPowerGresのサービスが立ち上がり、ユーザデータベースのデータがサーバ2に移行する為、クライアントはサーバ2へ接続し、運用します。フェイルオーバーにてフローティングIPアドレスがサーバ2へ移行する為、クライアントはサーバが切り替わったことを意識せずに、同一のフローティングIPアドレスで再接続することで運用が可能です。

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。



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。



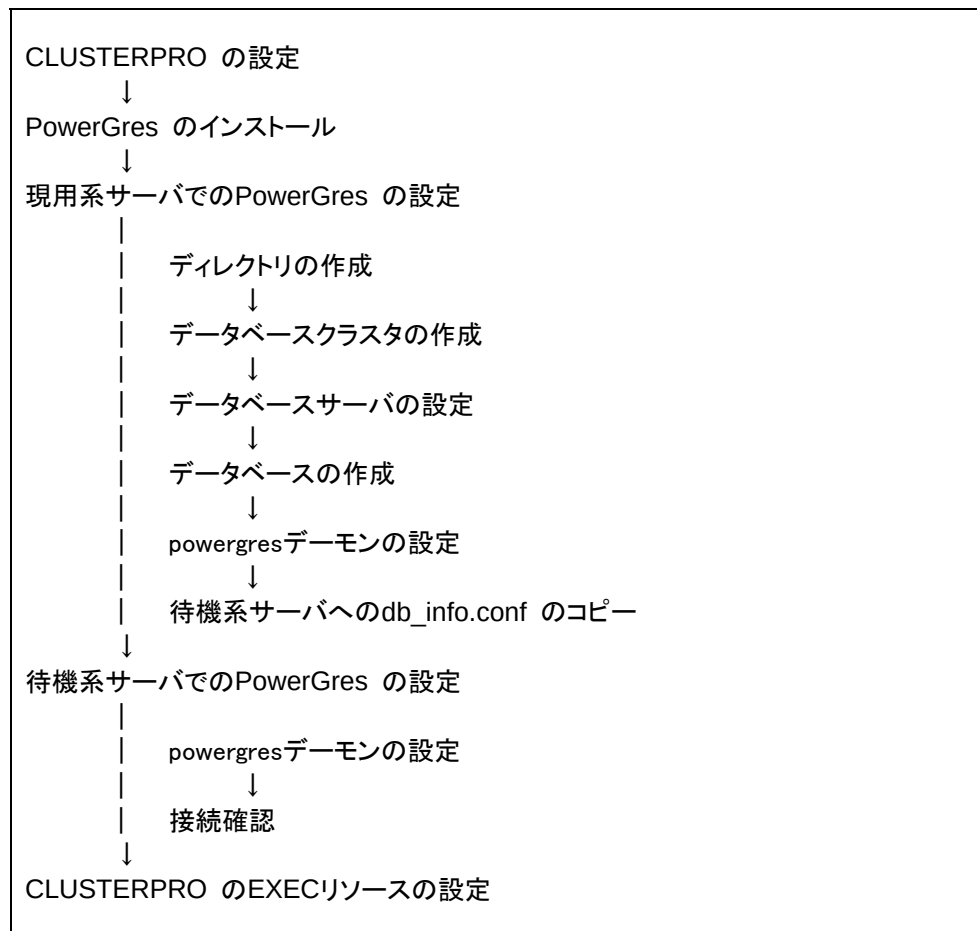
CLUSTERPRO環境下でのPowerGresの設定

CLUSTERPRO 環境下で PowerGres 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。

- ◆ 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初期化やデータベースの作成は、現用系のサーバからのみ実施します。待機系サーバでもデータベースを使用できるようにするために、現用系サーバのdb_info.confファイルを待機系サーバにコピーして、設定を反映します。

構築の流れ

PowerGres 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
クラスタパーティション	/dev/sda1	
データパーティション	/dev/sda2	

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	デバイス名	/dev/sda2
	マウントポイント	/mnt/sda2

PowerGres情報

PowerGresのバージョン	3
インストールディレクトリ	/opt/powergres
スーパーユーザ	postgres
データベースディレクトリ	/mnt/sda2/data
データベース名	db1
使用ポート番号	5432
クライアント認証方式	trust

CLUSTERPRO の設定

CLUSTERPRO の Builder を使用して、PowerGres 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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第 3 章 Builder で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。

PowerGres のインストール

インストールの前に、postgres ユーザがすでに存在する場合は削除します。両サーバにて root ユーザで以下を実行します。

```
# userdel postgres
```

ユーザの削除後、両サーバで PowerGres をインストールします。

スーパーユーザには、「postgres」を指定します。PowerGres Administrator Tool のポート番号は既定値(8080)を指定します。

PowerGres のインストールにより、postgres ユーザが作成されます。作成された postgres ユーザの ID を確認します。

```
# id postgres
```

uid、gid、groups の情報が両サーバで等しく設定されていることを確認してください。

現用系サーバでの PowerGres の設定

ディレクトリの作成

データベースディレクトリを作成します。/mnt/sda2/data ディレクトリを作成し、所有者を postgres ユーザに設定します。

```
# mkdir /mnt/sda2/data  
# chown -R postgres:postgres /mnt/sda2/data  
# chmod
```

データベースクラスタの設定

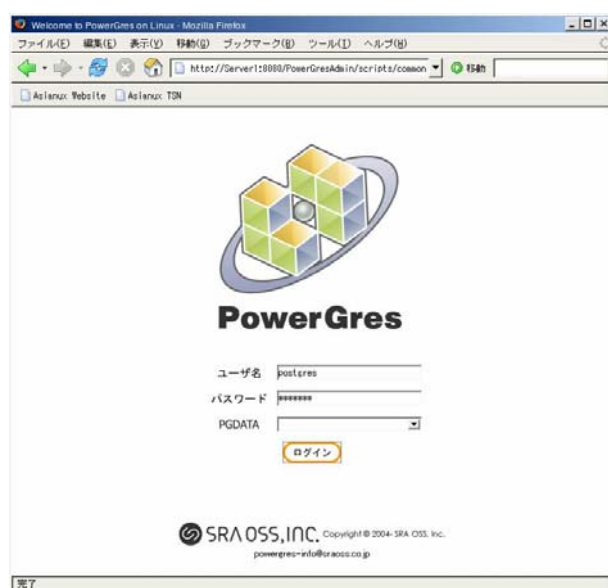
[PowerGres Administrator Tool]を使用して、PowerGres の設定を行います。

Web ブラウザを開き、以下の URL を指定します。

```
http://Server1:8080
```

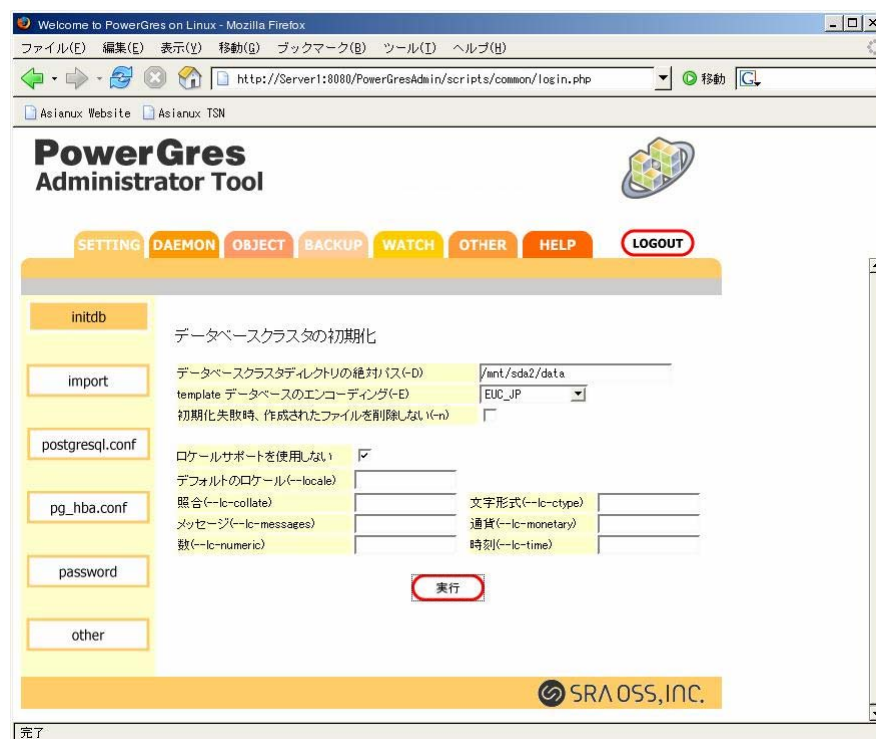
※太斜体の部分には、現用系サーバのサーバ名を指定します。

[ユーザ名][パスワード]には、postgres ユーザとそのパスワードを指定します。[PGDATA]は空白のまま[ログイン]を選択します。



[PowerGres Administrator Tool]画面で[SETTING]タブを選択し、[initdb]を選択します。

[データベースクラスタディレクトリの絶対パス]に/mnt/sda2/data を指定します。データベースのエンコーディング等の設定は、使用する環境に合わせて指定します。[実行]を選択し、以下の画面に「データベースクラスタを作成しました」が表示されることを確認します。

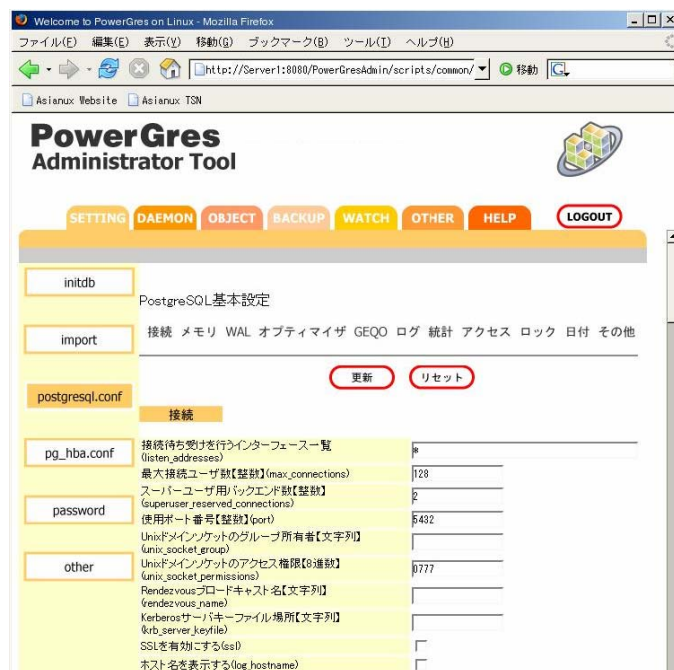


データベースサーバの設定

PowerGres Administrator Tool を使用して、postgresql.conf/pg_hba.conf の設定を行います。

[postgresql.conf]を選択します。[接続待ち受けを行うインターフェース一覧]の値を '*' にしてください。'*' を設定することで、サーバの利用可能な全ての IP インターフェースがクライアントからの接続の待ち受けに使用されます。

また、ポート番号を既定値(5432)から変更する場合は、[使用ポート番号]に適切な値を指定します。本章の例では既定値を使用するため、ポート番号を変更しません。



次に、[pg_hba.conf]を選択します。ここではクライアント接続の認証設定を行います。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。

以下の画面で[追加] を選択します。



ここでは、[接続種類]に host、[データベース][対象ユーザ]には all、[認証方式]には trust を指定します。データベースやユーザに応じて異なる認証設定を実施する場合は、[データベース][対象ユーザ]にデータベース名、ユーザ名を指定します。

[追加]を選択した後、[ネットワーク接続認証]の一覧に情報が追加されたことを確認します。

データベースの作成

データベースを作成するには、データベースのデーモンが起動している必要があります。

[DAEMON]タブを選択します。データベースデーモンの状態が停止状態の場合、[start]を選択してデータベースデーモンを開始します。

次に[OBJECT]タブを選択して[database]を選択します。画面右下の[作成]を選択します。

No	データベース名	所有者	文字エンコード
1	template1	postgres	EUC_JP
17229	template0	postgres	EUC_JP

[データベース名]に db1、[所有者]には postgres を指定し、[文字コード]は環境に合わせて指定してください。[実行]を選択してデータベースを作成します。データベース作成後、[データベース一覧] に db1 が追加されたことを確認してください。



データベースの作成後、データベースデーモンを停止します。[DAEMON]タブを選択して[stop]を選択してください。

これで PowerGres Administrator Tool を使用した PowerGres の設定は完了です。[LOGOUT]を選択して終了します。

powergres デーモンの設定

CLUSTERPRO 環境下においては、powergres の起動／停止は CLUSTERPRO で制御するため、powergres デーモンを手動起動に設定する必要があります。root ユーザで以下を実施してください。

```
# chkconfig --level 2345 powergres off
# chkconfig --list powergres
```

上記の結果、以下のように表示されることを確認してください。

```
powergres 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

待機系サーバへの db_info.conf のコピー

現用系サーバの db_info.conf を待機系サーバにコピーします。ftp コマンド等により、待機系サーバの同ディレクトリにコピーしてください。

```
/opt/powergres/admin/conf/db_info.cnf
```

これで現用系サーバでの PowerGres の設定は完了です。

待機系サーバでの PowerGres 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て、以下を実施してください。

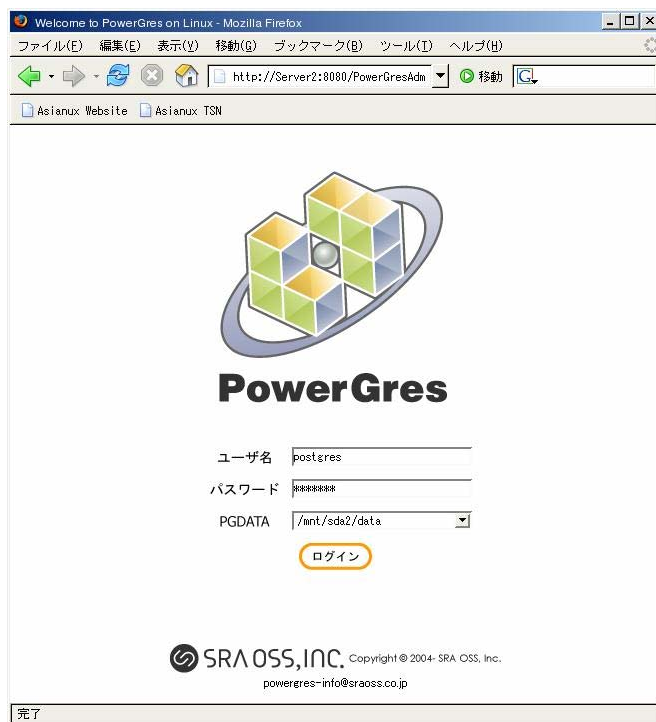
powergres デーモンの設定

[現用系サーバでの PowerGres の設定]の[powergres デーモンの設定]と同様に、powergres デーモンを手動起動に変更してください。

接続確認

現用系サーバで追加したデータベースが、待機系サーバでアクセスできることを確認します。

PowerGres Administrator Tool を起動します。ログイン画面の[PGDATA]には、現用系サーバで作成したデータベースクラスタディレクトリを指定します。



[DAEMON]タブを選択します。[start]を選択してデータベースデーモンを開始します。

psql 等を使用して、データベース:db1 に接続できることを確認してください。

接続確認が完了したら、[DAEMON]タブの[stop]を選択してデータベースデーモンを停止します。

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder を使用して、PowerGres 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) [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、「スクリプト内容例」のように記述します。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。

スクリプト内容例

PowerGres 起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /mnt/sda2/data/postmaster.pid ]
then
    rm -f /mnt/sda2/data/postmaster.pid
fi
if [ -f /tmp/.s.PGSQL.5432 ]
then
    rm -f /tmp/.s.PGSQL.*
fi

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PowerGres start"
    #
    # Modify a PowerGres user name, database domain directory, and port
    # number to appropriate values.
    #
    su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda2/data/
    -l /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
```

----- (1)

----- (2)

----- (3)

```

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 "PostgreSQL start"
    #
    # Modify a PowerGres user name, database domain directory, and port
    # number to appropriate values.
    #
    su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda2/data/
    -l /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

--- (3)

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

データベースディレクトリのパス/postmaster.pid

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

ソケットファイルのパスs.PGSQL.**ポート番号**

(3) PowerGres 起動処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - PowerGres ユーザ -c "/opt/powergres/bin/pg_ctl start -D データベースディレクトリ -l /dev/null -o '-i -p PowerGres ポート番号'"
```

◆ **stop.sh**

```
#!/bin/sh
#*****
#*          stop.sh          *
#*****
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PowerGres stop"
    #
    # Modify a PowerGres user name and database domain directory to
    # appropriate values.
    #
    su - postgres -c '/opt/powergres/bin/pg_ctl stop -D /mnt/sda2/data -m fast'
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$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 "PowerGres stop"
```

----- (1)

```
#
# Modify a PowerGres user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/opt/powergres/bin/pg_ctl stop -D /mnt/sda2/data -m fast' - - - (1)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

(1) PowerGres の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - PowerGres ユーザ -c "/opt/powergres/bin/pg_ctl stop -D データベースディレクトリ -m fast"
```

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
クラスタパーティション	/dev/sda1	
データパーティション	/dev/sda2 , /dev/sda3	

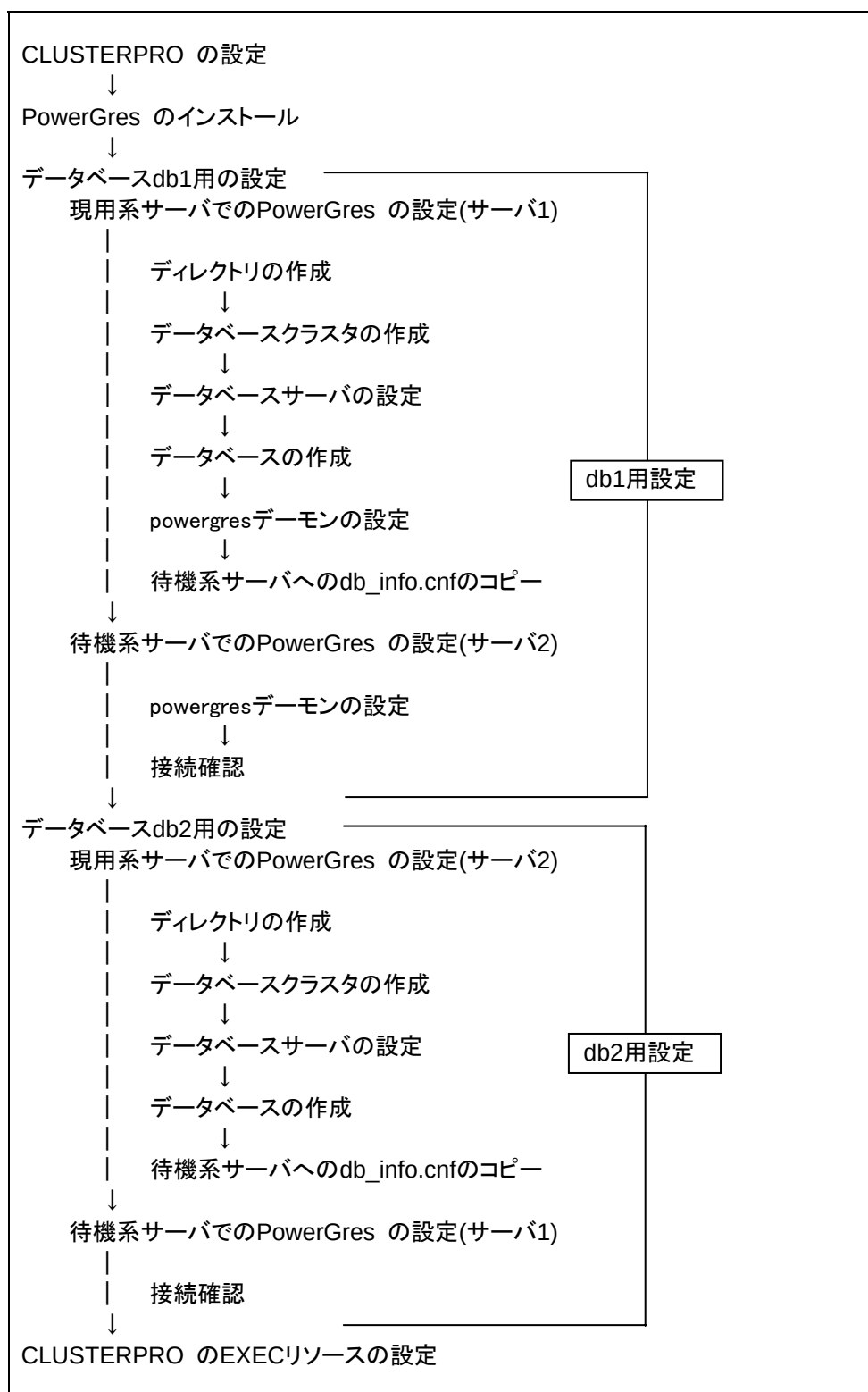
フェイルオーバーグループ情報

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	デバイス名	/dev/sda2
	マウントポイント	/mnt/sda2
グループ名	グループ1	
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	デバイス名	/dev/sda3
	マウントポイント	/mnt/sda3

PowerGres情報

PowerGresのバージョン	3	
インストールディレクトリ	/opt/powergres	
インストールユーザ	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/sda2/data
	データベース名	db1
	使用ポート番号	5432
	クライアント認証方式	trust
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/sda3/data
	データベース名	db2
	使用ポート番号	5433
	クライアント認証方式	trust

双方向の環境を作成する場合は、データベース db1、データベース db2 用の設定を同時に行わないようにします。設定の流れは以下のようになります。



CLUSTERPRO の設定

CLUSTERPRO の Builder を使用して、PowerGres 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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第 3 章 Builder で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。

PowerGres のインストール

インストールの前に、postgres ユーザがすでに存在する場合は削除します。両サーバにて root ユーザで以下を実行します。

```
# userdel postgres
```

ユーザの削除後、両サーバで PowerGres をインストールします。

スーパーユーザには、「postgres」を指定します。PowerGres Administrator Tool のポート番号は既定値(8080)を指定します。

PowerGres のインストールにより、postgres ユーザが作成されます。作成された postgres ユーザの ID を確認します。

```
# id postgres
```

uid、gid、groups の情報が両サーバで等しく設定されていることを確認してください。

現用系サーバでの PowerGres 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。

ディレクトリの作成

データベースディレクトリを作成して所有者の設定を実施します。

- ◆ Server1の場合

```
# mkdir /mnt/sda2/data  
# chown -R postgres:postgres /mnt/sda2/data
```

- ◆ Server2の場合

```
# mkdir /mnt/sda3/data  
# chown -R postgres:postgres /mnt/sda3/data
```

データベースクラスタの設定

[PowerGres Administrator Tool]を使用して、PowerGres の設定を行います。

Web ブラウザを開き、以下の URL を指定します。

http://サーバ名:8080

[ユーザ名][パスワード]には、postgres ユーザとそのパスワードを指定します。

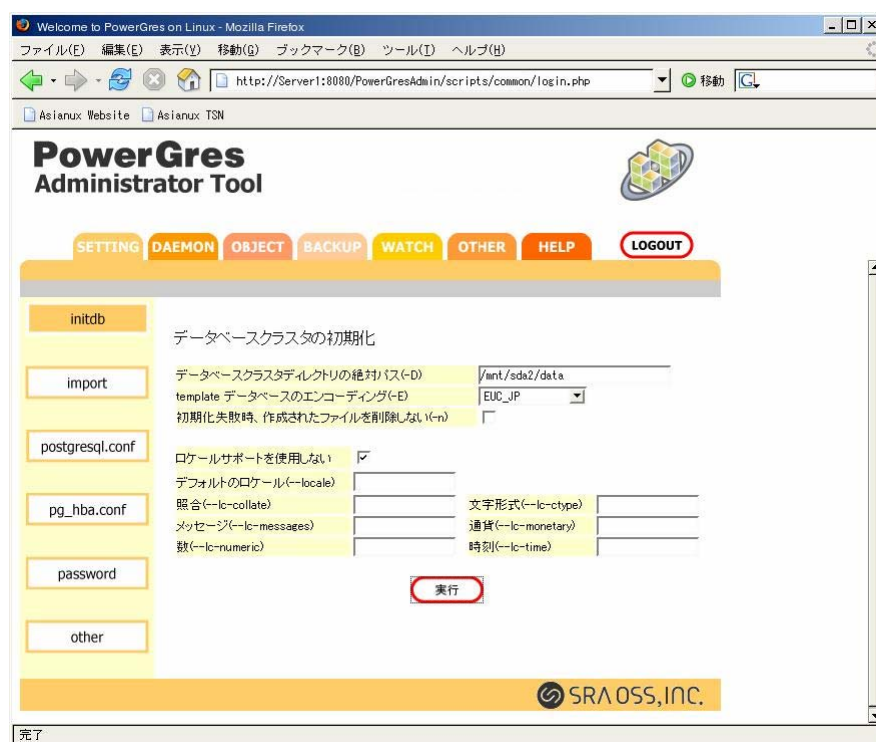


[PowerGres Administrator Tool]画面で[SETTING]タブを選択し、[initdb]を選択します。

[データベースクラスタディレクトリの絶対パス]には以下の値を指定します。

- ◆ Server1の場合 /mnt/sda2/data
- ◆ Server2の場合 /mnt/sda3/data

データベースのエンコーディング等の設定は、使用する環境に合わせて指定します。[実行]を選択し、以下の画面に「データベースクラスタを作成しました」が表示されることを確認します。

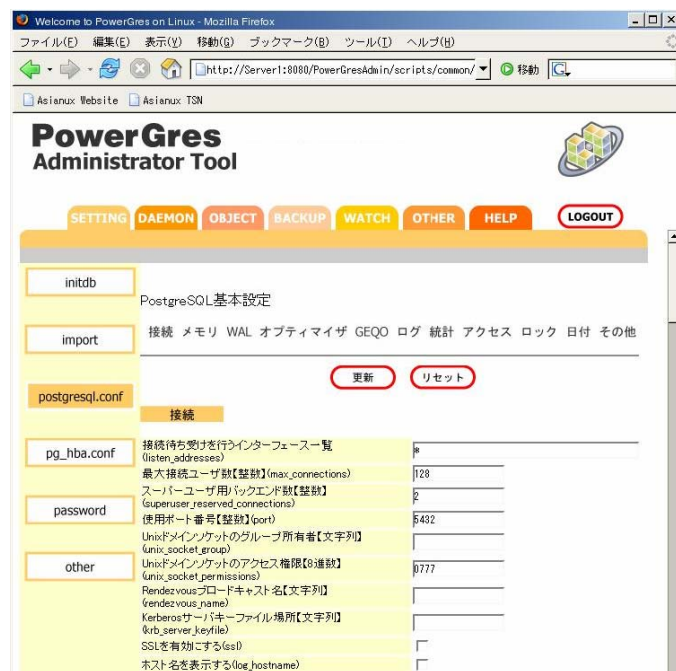


データベースサーバの設定

PowerGres Administrator Tool を使用して、postgresql.conf/pg_hba.conf の設定を行います。

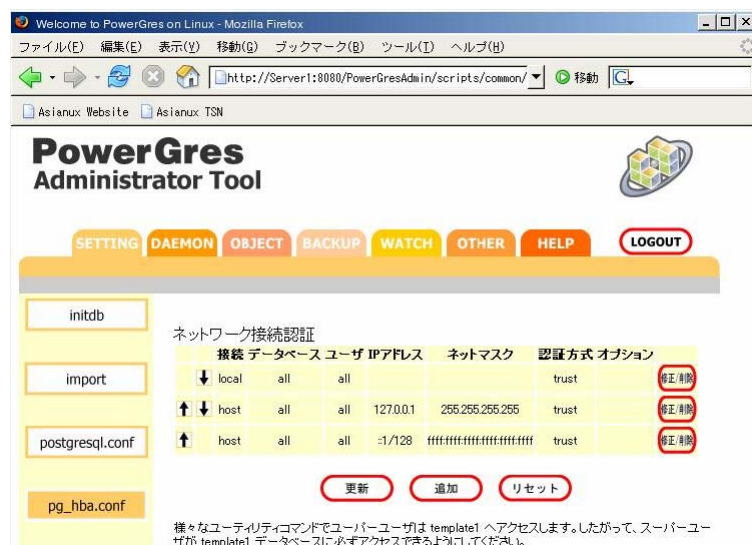
[postgresql.conf]を選択します。[接続待ち受けを行うインターフェース一覧]の値を '*' にしてください。'*' を設定することで、サーバの利用可能な全ての IP インターフェースがクライアントからの接続の待ち受けに使用されます。

Server2 で設定を行う場合は、[使用ポート番号]の値を 5433 に変更します。Server1 の場合は、PowerGres の既定値(5432)を使用するため、[使用ポート番号]の変更は変更しません。



次に、[pg_hba.conf]を選択します。ここではクライアント接続の認証設定を行います。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。

以下の画面で[追加] を選択します。



ここでは、[接続種類]に host、[データベース][対象ユーザ]には all、[認証方式]には trust を指定します。データベースやユーザに応じて異なる認証設定を実施する場合は、[データベース][対象ユーザ]にデータベース名、ユーザ名を指定します。

PowerGres Administrator Tool

SETTING DAEMON OBJECT BACKUP WATCH OTHER HELP LOGOUT

initdb 追加するネットワーク接続認証情報を入力してください

import

接続種類 host

データベース all

対象ユーザ all

IPアドレス 192.168.1.0

ネットマスク 24

認証方式 trust

オプション

追加 戻る

[追加]を選択した後、[ネットワーク接続認証]の一覧に情報が追加されたことを確認します。

データベースの作成

データベースを作成する際には、データベースのデーモンが起動している必要があります。

[DAEMON]タブを選択します。データベースデーモン状態が停止状態の場合、[start]を選択してデータベースデーモンを開始します。

次に[OBJECT]タブを選択して[database]を選択します。画面右下の[作成]を選択します。

PowerGres Administrator Tool

SETTING DAEMON OBJECT BACKUP WATCH OTHER HELP LOGOUT

database データベース一覧

No	データベース名	所有者	文字エンコード
1	template1	postgres	EUC_JP
17229	template0	postgres	EUC_JP

作成

Server1 の場合は[データベース名]に db1、Server2 の場合は[データベース名]に db2 を指定します。

[所有者]には postgres を指定し、[文字コード]は環境に合わせて指定してください。[実行]を選択してデータベースを作成します。[データベース一覧] に db1 および db2 が追加されたことを確認してください。



データベースの作成後、データベースデーモンを停止します。[DAEMON]タブを選択して[stop]を選択してください。

これで PowerGres Administrator Tool を使用した PowerGres の設定は完了です。[LOGOUT]を選択して終了します。

powergres デーモンの設定

CLUSTERPRO 環境下においては、powergres の起動／停止は CLUSTERPRO で制御するため、powergres デーモンを手動起動に設定する必要があります。root ユーザで以下を実施してください。

```
# chkconfig --level 2345 powergres off
# chkconfig --list powergres
```

上記の結果、以下のように表示されることを確認してください。

```
powergres 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

※ 上記の設定をサーバですでに実施済みの場合は、本項を実施する必要はありません。

待機系サーバへの db_info.cnf のコピー

現用系サーバの db_info.cnf を待機系サーバにコピーします。ftp コマンド等により、待機系サーバの同ディレクトリにコピーしてください。

```
/opt/powergres/admin/conf/db_info.cnf
```

待機系サーバでの PowerGres 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て、以下を実施してください。

powergres デーモンの設定

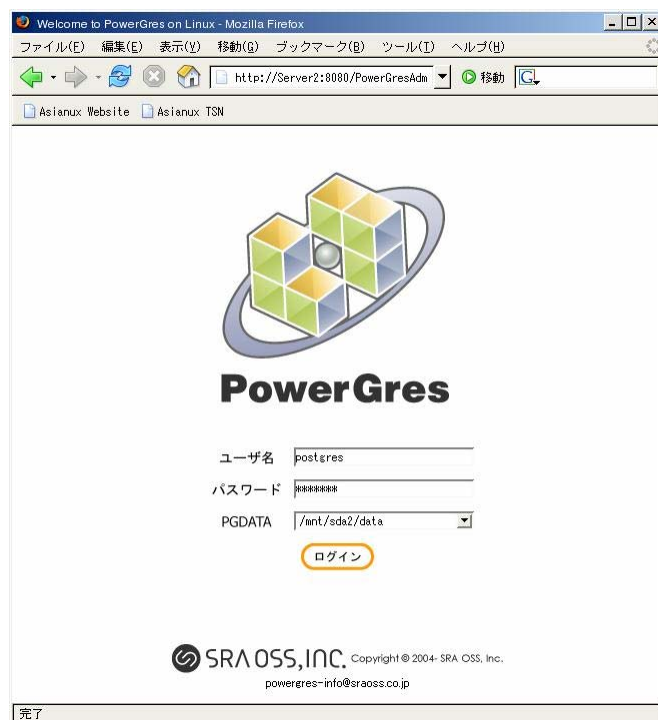
[現用系サーバでの PowerGres の設定]の[powergres デーモンの設定]と同様に、powergres デーモンを手動起動に変更してください。

※上記の設定をサーバですでに実施済みの場合は、本項を実施する必要はありません。

接続確認

現用系サーバで追加したデータベースが、待機系サーバでアクセスできることを確認します。

PowerGres Administrator Tool を起動します。ログイン画面の[PGDATA]には、現用系サーバで作成したデータベースクラスタディレクトリを指定します。



[DAEMON]タブを選択します。[start]を選択してデータベースデーモンを開始します。

psql 等を使用して、データベース:db1 に接続できることを確認してください。

接続確認が完了したら、[DAEMON]タブの[stop]を選択してデータベースデーモンを停止します。

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder を使用して、グループ 1、グループ 2 のそれぞれに対して PowerGres 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) [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、「スクリプト内容例」のように記述します。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。

スクリプト内容例

PowerGres 起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /mnt/sda2/data/postmaster.pid ]
then
    rm -f /mnt/sda2/data/postmaster.pid
fi
if [ -f /tmp/s.PGSQL.5432 ]
then
    rm -f /tmp/s.PGSQL.*
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 "PowerGres start"
    fi

    #
    # Modify a PowerGres user name, database domain directory, and port
    # number to appropriate values.
    #
    su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda2/data/ -l
    /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
```

----- (1)

----- (2)

----- (3)

```

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 "PostgreSQL start"
    #
    # Modify a PowerGres user name, database domain directory, and port
    # number to appropriate values.
    #
    su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda2/data/ -l
    /dev/null -o '-i -p 5432'" ----- (3)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

データベースディレクトリのパス/postmaster.pid

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

ソケットファイルのパス/s.PGSQL.**ポート番号**.pid

(3) PowerGres 起動処理

太斜体の箇所は、環境に合わせて変更してください。

su - **PowerGres ユーザ** -c "/opt/powergres/bin/pg_ctl start -D **データベースディレクトリ** -l /dev/null -o '-i -p **PowerGres のポート番号**' "

フェイルオーバーグループ 1 の場合は、「su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda2/data -l /dev/null -o '-i -p 5432' '"」となります。

フェイルオーバーグループ 2 の場合は、「su - postgres -c "/opt/powergres/bin/pg_ctl start -D /mnt/sda3/data -l /dev/null -o '-i -p 5433' '"」となります。

- ◆ **stop.sh**

```

#!/bin/sh
#####
#*          stop.sh          *
#####
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 "PowerGres stop"

#
# Modify a PowerGres user name and database domain directory to
# appropriate values.
#
su - postgres -c ' /opt/powergres/bin/pg_ctl stop -D /mnt/sda2/data -m fast'
    else
        echo "ERROR_DISK from START"
    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 "PowerGres stop"

```

```

#
# Modify a PowerGres user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/opt/powergres/bin/pg_ctl stop -D /mnt/sda2/data -m fast' ----- (1)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

(1) PowerGres の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```

su - PowerGres ユーザ -c '/opt/powergres/bin/pg_ctl stop -D データベースディレクトリ -m fast'

```

フェイルオーバーグループ 1 の場合は、「su - postgres -c '/opt/powergres/bin/pg_ctl stop -D /mnt/sda2/data -m fast' 」となります。

フェイルオーバーグループ 2 の場合は、「su - postgres -c '/opt/powergres/bin/pg_ctl stop -D /mnt/sda3/data -m fast' 」となります。

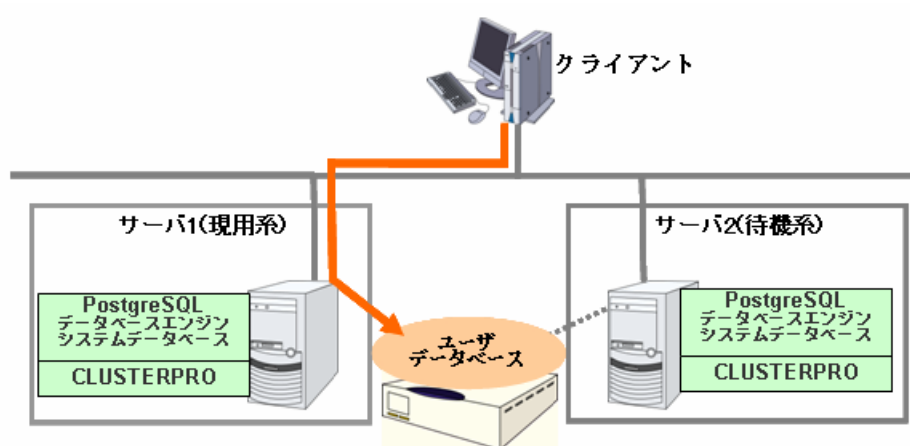
第 2 章 PostgreSQL

機能概要

PostgreSQL を、CLUSTERPRO X 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。）

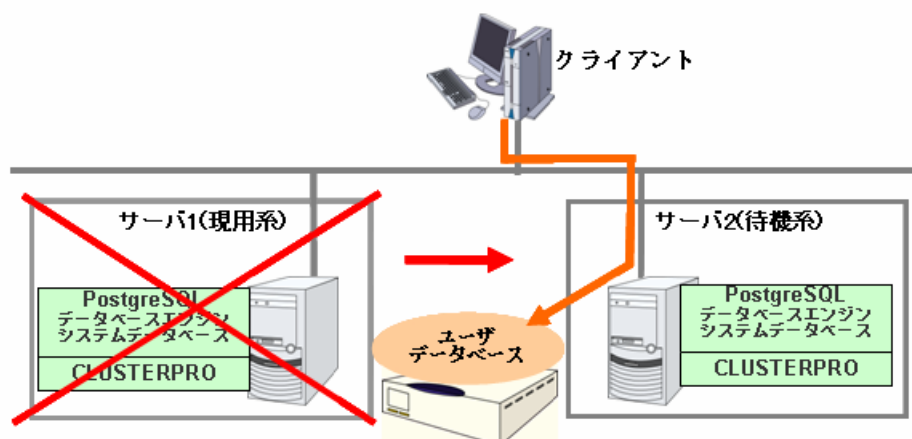
片方向スタンバイ

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。



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 アドレスや仮想コンピュータ名を使用して接続します。

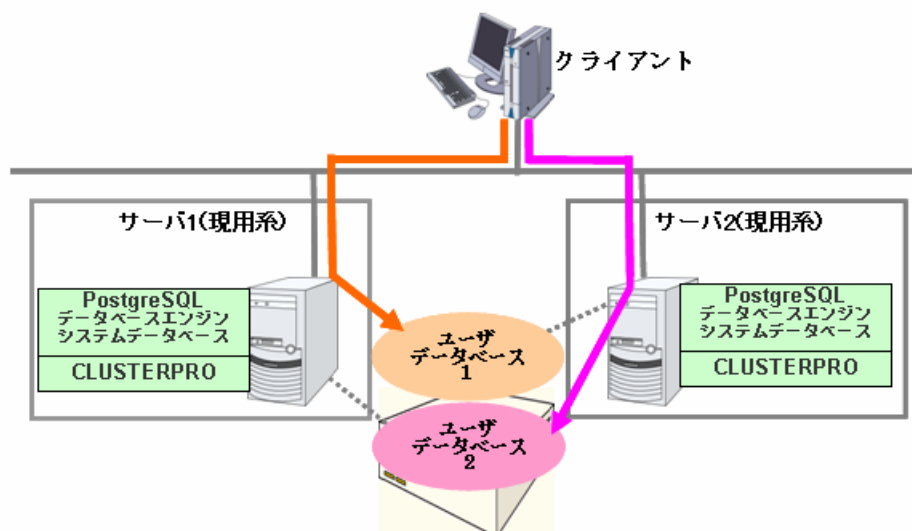
サーバ 1 に障害が発生すると以下の図のようになります。



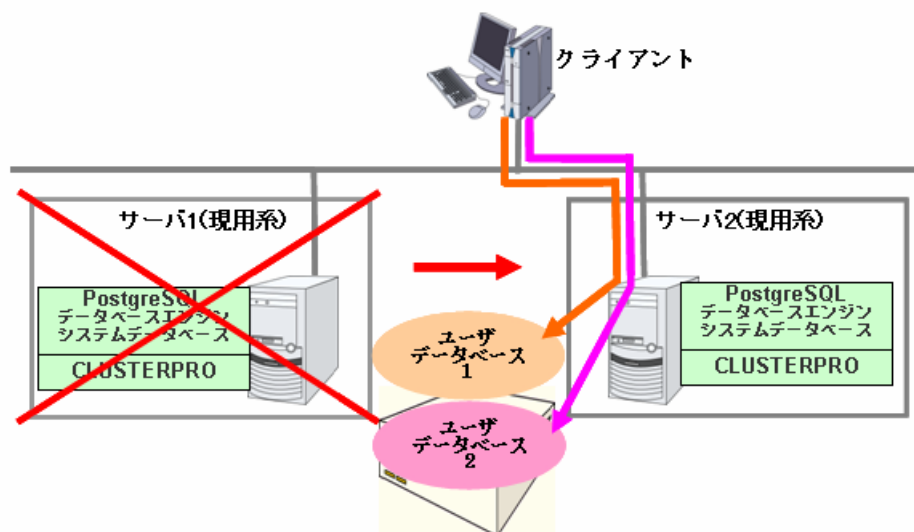
障害によりフェイルオーバーが完了すると、サーバ 2 で PostgreSQL のサービスが立ち上がり、ユーザデータベースのデータがサーバ 2 に移行する為、クライアントはサーバ 2 へ接続し、運用します。フェイルオーバーにてフローティング IP アドレスがサーバ 2 へ移行する為、クライアントはサーバが切り替わったことを意識せずに、同一のフローティング IP アドレスで再接続することで運用が可能です。

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。



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。



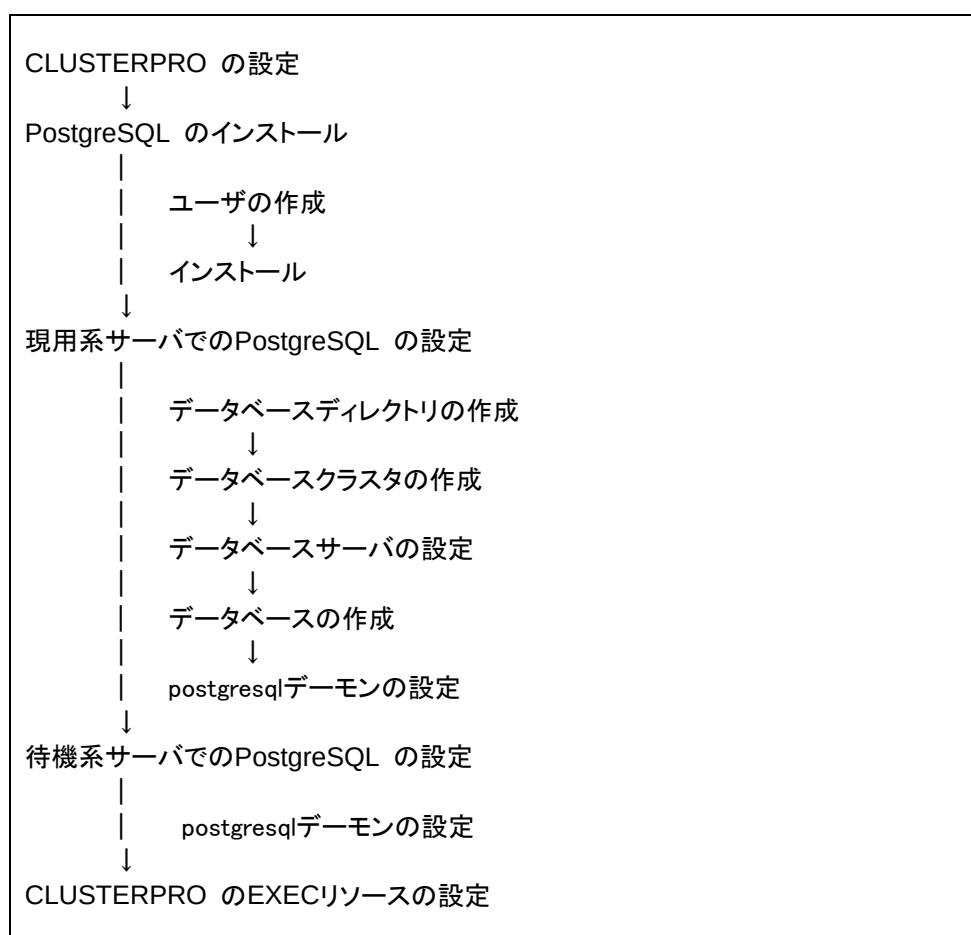
CLUSTERPRO環境下でのPostgreSQLの設定

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。

- ◆ ユーザーデータベースは、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。
- ◆ CLUSTERPRO X Database Agent 1.0 for LinuxのPostgreSQL監視機能をご使用になる場合、使用可能なPostgreSQLのバージョンは、PostgreSQL8.1になります。

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
クラスタパーティション	/dev/sda1	
データパーティション	/dev/sda2	

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	デバイス名	/dev/sda2
	マウントポイント	/mnt/sda2

PostgreSQL情報

PostgreSQLのバージョン	8.1
インストールディレクトリ	/usr/local/pgsql
管理用のユーザ	postgres
データベースディレクトリ	/mnt/sda2/pgsql/data
データベース名	db1
文字コード	EUC_JP
使用ポート番号	5432
クライアント認証方式	trust

CLUSTERPRO の設定

CLUSTERPROのBuilder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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第3章 Builderで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。LinuxOSにバンドルされているPostgreSQLを使用する場合、「インストール」の作業は不要となりますが、PostgreSQL用のユーザの確認は実施してください。

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: xxxxx
Retry new UNIX password: xxxxx
```

※太斜線の部分にパスワードを指定します。

インストール

PostgreSQL のソースをサーバの任意の場所に展開して解凍し、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-8.1.5.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。

```
# cd /tmp
# tar -zxvf postgresql-8.1.5.tar.gz
# cd postgresql-8.1.5
# ./configure --prefix=/usr/local/pgsql
# make all
# make install
# chown -R postgres:postgres /usr/local/pgsql
```

現用系サーバでの PostgreSQL の設定

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。クラスター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。

ここでは、root ユーザで以下のコマンドを実施します。

```
# mkdir -p /mnt/sda2/pgsql/data
# chown -R postgres:postgres /mnt/sda2/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 ユーザにログオンして、以下のコマンドを実施します。

```
# su postgres
$ /usr/local/pgsql/bin/initdb -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
```

下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:

""postmaster -D "/mnt/sda2/pgsql/data"

or

""pg_ctl -D "/mnt/sda2/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/sda2/pgsql/data/postgresql.conf

/mnt/sda2/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 インターフェースの設定や 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_address」の値を以下のように変更します。

```
listen_address = '*'
```

PostgreSQL が使用するポート番号を既定値(5432)から変更する場合は、「port」の値を変更します。

```
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。

◆ 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 アドレスからの接続許可のみがあります。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。以下の一行を追加します。

```
host    all    all    192.168.1.0/24    trust
```

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。

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 ユーザで以下を実行します。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。

```
# su postgres
$ /usr/local/pgsql/bin/pg_ctl start -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/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。

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/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast
```

postgresql デーモンの設定

postgresql デーモンが手動起動になっていない場合、手動起動に設定します。

root ユーザで以下を実行します。

```
# chkconfig --list postgresql
```

上記を実施した結果、起動レベル 3 や 5 で on が設定されている場合は、以下を実施します。

```
# chkconfig --level 35 postgresql off
```

再度"chkconfig --list postgresql" を実施して、起動レベル 3 や 5 で off が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/local/pgsql/bin/pg_ctl start -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/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/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast
```

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder 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) [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、「スクリプト内容例」のように記述します。スクリプトの編集ができたなら、画面に従って EXEC リソースの追加を実施してください。

スクリプト内容例

PostgreSQL の起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /mnt/sda2/pgsql/data/postmaster.pid ]
then
    rm -f /mnt/sda2/pgsql/data/postmaster.pid
fi
if [ -f /tmp/.s.PGSQL.5432 ]
then
    rm -f /tmp/.s.PGSQL.*
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 "PostgreSQL start"

#
# Modify a PostgreSQL user name, database domain directory, and
# port number to appropriate values.
#
su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D
/mnt/sda2/pgsql/data/ -l /dev/null -o '-i -p 5432'" ----- (3)
    else
        echo "ERROR_DISK from START"
    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 " PostgreSQL start"
    fi
#
# Modify a PostgreSQL user name, database domain directory, and
# port number to appropriate values.
#
su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D
/mnt/sda2/pgsql/data/ -l /dev/null -o '-i -p 5432'" ----- (3)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

データベースディレクトリのパス/postmaster.pid

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

ソケットファイルのパス/s.PGSQL.ポート番号

(3) PostgreSQL 起動処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理ユーザ -c "pg_ctl のパス/pg_ctl start -D データベースディレクトリ  
-l /dev/null -o '-i -p PostgreSQL のポート番号'
```

◆ stop.sh

```

#!/bin/sh
#*****
#*          stop.sh          *
#*****
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PostgreSQL stop"
    #
    # Modify a PostgreSQL user name and database domain directory to
    # appropriate values.
    #
    su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data
    -m fast' ----- (1)
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$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 " PostgreSQL stop"

```

```
#
# Modify a PowerGres user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data
-m fast' ----- (1)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

(1) PostgreSQL の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理者ユーザ -c "pg_ctl のパスpg_ctl stop -D データベースディレクトリ -m fast"
```

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
クラスタパーティション	/dev/sda1	
データパーティション	/dev/sda2 , /dev/sda3	

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	デバイス名	/dev/sda2
	マウントポイント	/mnt/sda2
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	デバイス名	/dev/sda3
	マウントポイント	/mnt/sda3

PostgreSQL情報

PostgreSQLのバージョン	8.1	
インストールディレクトリ	/usr/local/pgsql	
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/sda2/pgsql/data
	データベース名	db1
	文字コード	EUC_JP
	使用ポート番号	5432
	クライアント認証方式	trust
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/sda3/pgsql/data
	データベース名	db2
	文字コード	EUC_JP
	使用ポート番号	5433
	クライアント認証方式	trust

CLUSTERPRO の設定

CLUSTERPRO の Builder 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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第3章 Builder で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。LinuxOS にバンドルされている PostgreSQL を使用する場合、「インストール」の作業は不要となりますが、PostgreSQL 用のユーザの確認は実施してください。

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
```

※太斜線の部分にパスワードを指定します。

インストール

PostgreSQL のソースをサーバの任意の場所に展開して解凍し、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-8.1.5.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。

```
# cd /tmp
# tar -zxvf postgresql-8.1.5.tar.gz
# cd postgresql-8.1.5
# ./configure --prefix=/usr/local/pgsql
# make all
# make install
# chown -R postgres:postgres /usr/local/pgsql
```

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/sda2/pgsql/data
# chown -R postgres:postgres /mnt/sda2/pgsql/data
```

◆ Server2の場合

```
# mkdir -p /mnt/sda3/pgsql/data
# chown -R postgres:postgres /mnt/sda3/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 ユーザにログオンして、以下のコマンドを実施します。

◆ Server1の場合

```
# su postgres
$ /usr/local/pgsql/bin/initdb -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
```

◆ Server2の場合

```
# su postgres
$ /usr/local/pgsql/bin/initdb -D /mnt/sda3/pgsql/data -E EUC_JP --no-locale
```

下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:
    ""postmaster -D ""/mnt/sda2/pgsql/data""
or
    ""pg_ctl -D ""/mnt/sda2/pgsql/data "" -l logfile start
```

※ 太斜体の箇所は、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/sda2/pgsql/data/postgresql.conf

```
/mnt/sda2/pgsql/data/pg_hba.conf
```

【Server2】

```
/mnt/sda3/pgsql/data/postgresql.conf
```

```
/mnt/sda3/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 インターフェースの設定や 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_address」の値を以下のように変更します。

```
listen_address = '*'
```

PostgreSQL が使用するポート番号を既定値(5432)から変更する場合は、「port」の値を変更します。

```
port = ポート番号
```

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

◆ 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 アドレスからの接続許可のみがあります。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。以下の一行を追加します。

```
host    all    all    192.168.1.0/24    trust
```

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。

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 ユーザで以下を実行します。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

```
# su postgres
$ /usr/local/pgsql/bin/pg_ctl start -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/bin/createdb -h localhost -U postgres db1
```

◆ Server2の場合

```
# su postgres
$ /usr/local/pgsql/bin/pg_ctl start -D /mnt/sda3/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/bin/createdb -h localhost -U postgres db2
```

※ 下線の部分は、データベース名を指定します。

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/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast
```

◆ Server2の場合

```
$ /usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda3/pgsql/data -m fast
```

postgresql デーモンの設定

postgresql デーモンが手動起動になっていない場合、手動起動に設定します。

root ユーザで以下を実行します。

```
# chkconfig --list postgresql
```

上記を実施した結果、起動レベル 3 や 5 で on が設定されている場合は、以下を実施します。

```
# chkconfig --level 35 postgresql off
```

再度“chkconfig --list postgres” を実施して、起動レベル 3 や 5 で off が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、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。

◆ Server1の場合

```
# su postgres
$ /usr/local/pgsql/bin/pg_ctl start -D /mnt/sda3/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/bin/psql -h localhost -U postgres db2
```

◆ Server2の場合

```
# su postgres
$ /usr/local/pgsql/bin/pg_ctl start -D /mnt/sda2/pgsql/data -E EUC_JP --no-locale
$ /usr/local/pgsql/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。

◆ Server1の場合

```
$ /usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda3/pgsql/data -m fast
```

◆ Server2の場合

```
$ /usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast
```

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder 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) [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、「スクリプト内容例」のように記述します。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。

スクリプト内容例

PostgreSQL の起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /mnt/sda2/pgsql/data/postmaster.pid ]
then
    rm -f /mnt/sda2/pgsql/data/postmaster.pid
fi
if [ -f /tmp/.s.PGSQL.5432 ]
then
    rm -f /tmp/.s.PGSQL.*
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 "PostgreSQL start"
#
# Modify a PostgreSQL user name, database domain directory, and
# port number to appropriate values.
#
su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D
/mnt/sda2/pgsql/data/ -l /dev/null -o '-i -p 5432'" ----- (3)
    else
        echo "ERROR_DISK from START"
    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 " PostgreSQL start"
    #
    # Modify a PostgreSQL user name, database domain directory, and
    # port number to appropriate values.
    #
    su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D
/mnt/sda2/pgsql/data/ -l /dev/null -o '-i -p 5432'" ----- (3)
        else
            echo "ERROR_DISK from FAILOVER"
        fi
    else
        echo "NO_CLP"
    fi
echo "EXIT"
exit 0

```

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

```
データベースディレクトリのパス/postmaster.pid
```

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

```
ソケットファイルのパス/s.PGSQL.ポート番号
```

(3) PostgreSQL 起動処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理ユーザ -c "pg_ctl のパス/pg_ctl start -D データベースディレクトリ -l /dev/null -o '-i -p PostgreSQL のポート番号'
```

グループ 1 の場合は、「su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D /mnt/sda2/pgsql/data/ -l /dev/null -o '-i -p 5432'"」となります。

グループ 2 の場合は、「su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D /mnt/sda3/pgsql/data/ -l /dev/null -o '-i -p 5433'"」となります。

◆ stop.sh

```

#!/bin/sh
#*****
#*          stop.sh          *
#*****
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PostgreSQL stop"
    #
    # Modify a PostgreSQL user name and database domain directory to
    # appropriate values.
    #
    su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data
    -m fast' ----- (1)
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$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 " PostgreSQL stop"
    
```

```
#
# Modify a PowerGres user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast' ----- (1)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

(1) PostgreSQL の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理者ユーザ -c "pg_ctl のパスpg_ctl stop -D データベースディレクトリ -m fast"
```

グループ 1 の場合は、「su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda2/pgsql/data -m fast'」となります。

グループ 2 の場合は、「su - postgres -c '/usr/local/pgsql/bin/pg_ctl stop -D /mnt/sda3/pgsql/data -m fast'」となります。

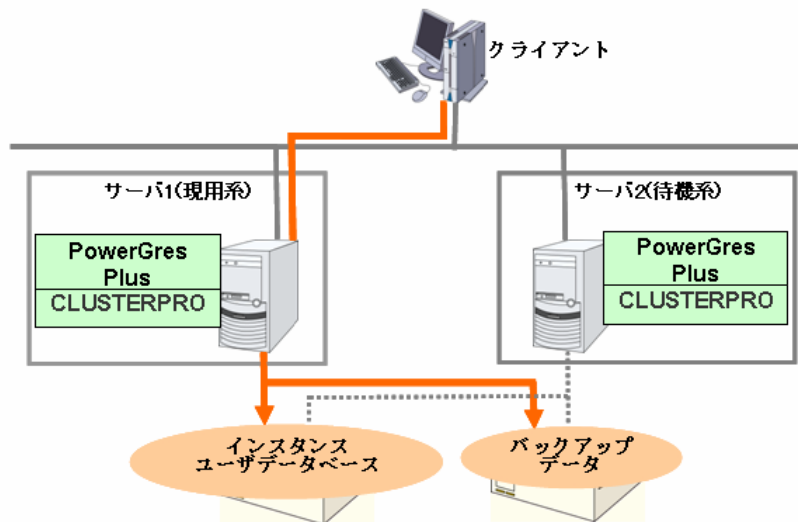
第 3 章 PowerGres Plus

機能概要

PowerGres Plus を、CLUSTERPRO X 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 PowerGres Plus 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。）

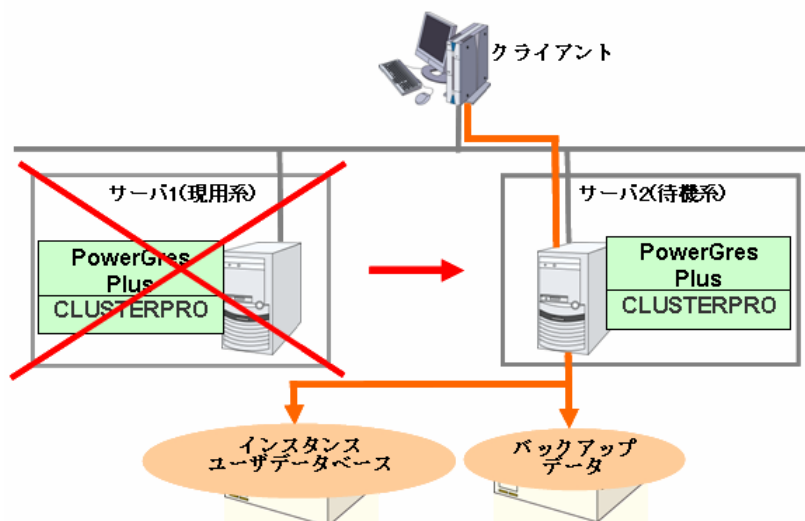
片方向スタンバイ

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。



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 アドレスや仮想コンピュータ名を使用して接続します。

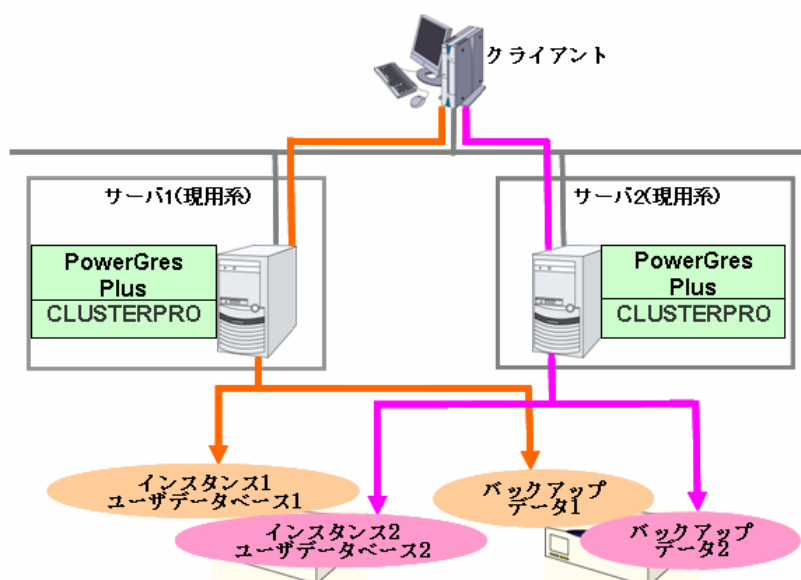
サーバ 1 に障害が発生すると以下の図のようになります。



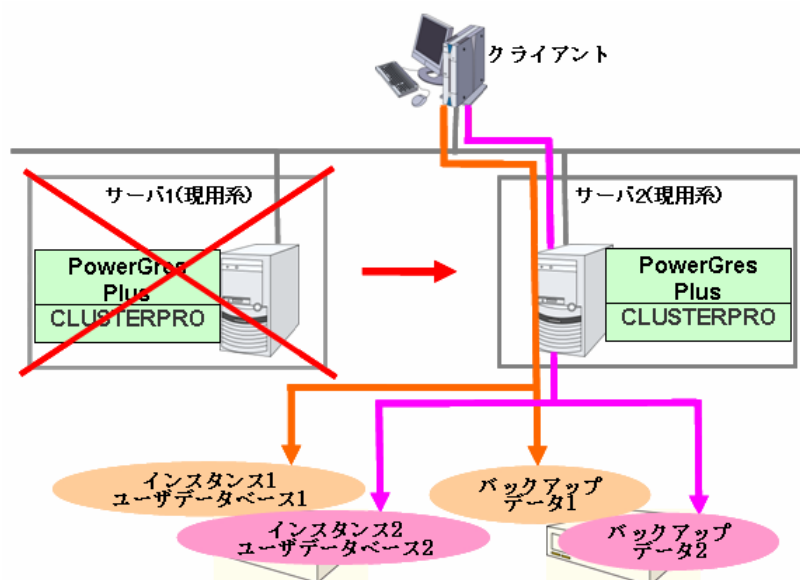
障害によりフェイルオーバーが完了すると、サーバ 2 で PowerGres Plus のサービスが立ち上がり、インスタンス/ユーザデータベース/バックアップデータの資源がサーバ 2 に移行する為、クライアントはサーバ 2 へ接続し、運用します。フェイルオーバーにてフローティング IP アドレスがサーバ 2 へ移行する為、クライアントはサーバが切り替わったことを意識せずに、同一のフローティング IP アドレスで再接続することで運用が可能です。

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。



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、インスタンス 1/ユーザデータベース 1/バックアップデータ 1 の資源がサーバ2に移行します。サーバ1を介して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。



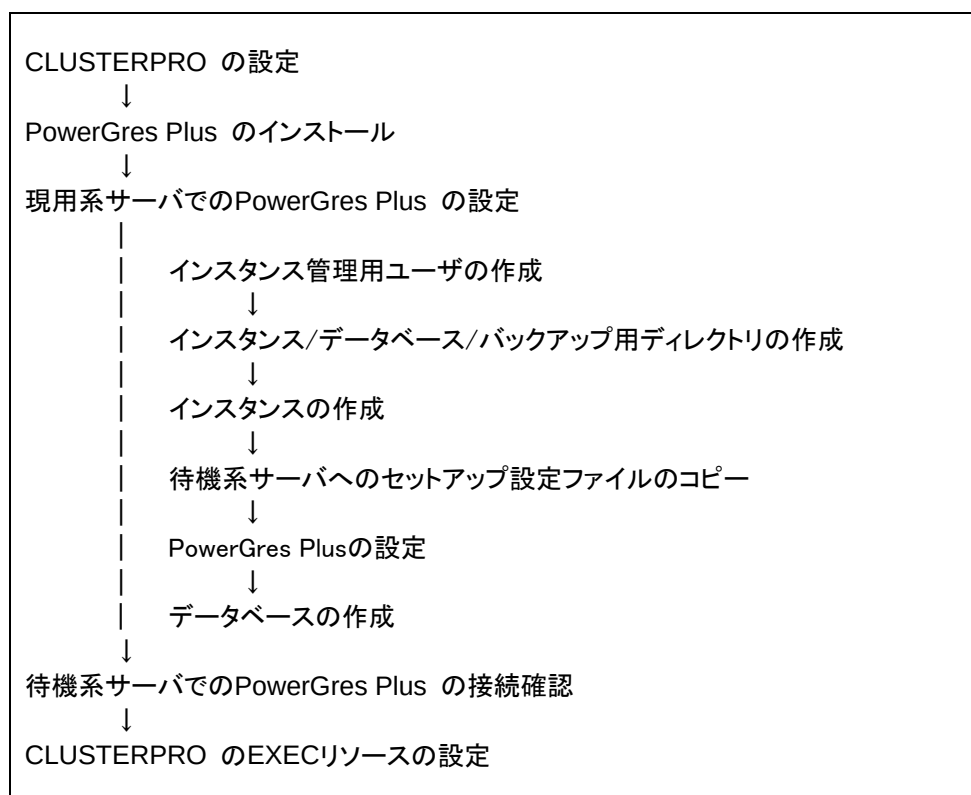
CLUSTERPRO環境下でのPowerGres Plusの設定

CLUSTERPRO 環境下で PowerGres Plus 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。

- ◆ インスタンスデータ／ユーザデータベース／バックアップデータは、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。
- ◆ インスタンスの作成／データベースの作成はクラスタ内の現用系サーバからのみ実施します。現用系サーバで作成されたセットアップ設定ファイル(インスタンス名.html)を待機系サーバにコピーすることにより、待機系サーバでもインスタンス、データベースを認識できるようにします。

構築の流れ

PowerGres Plus 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
クラスタパーティション	/dev/sda1	
データパーティション	/dev/sda2、/dev/sdb1(※)	

※ 各データパーティションはそれぞれ別のディスクに存在する

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1	デバイス名	/dev/sda2
	マウントポイント	/PlusData
ディスクリソース2	デバイス名	/dev/sdb1
	マウントポイント	/PlusBackup

PowerGres Plus情報

PowerGres Plusのバージョン	2.0
インストールディレクトリ	/usr/local/pgsqlplus
インスタンス名	postgres
インスタンス作成ユーザ	postgres
インスタンス情報／ ユーザデータベース格納場所	/PlusData/postgres
バックアップデータ格納場所	/PlusBackup/postgres
データベース名	db1
文字コード	EUC_JP
使用ポート番号	5432
クライアント認証方式	trust

CLUSTERPRO の設定

CLUSTERPRO の Builder を使用して、PowerGres Plus 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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第3章 Builder で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。

PowerGres Plus のインストール

両サーバで PowerGres Plus のインストールを実施します。root ユーザで、以下を実施します。

```
# mount /media/cdrom
# cd /media/cdrom
# sh ./install.sh
```

※太斜体の部分は、CD 媒体のマウントポイントです。お使いの環境に合わせて読み替えてください。

現用系サーバでの PowerGres Plus の設定

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。

インスタンス管理用ユーザの作成

インスタンス管理用のユーザ:postgres を OS に登録します。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する必要があります。両サーバで以下のように実行します。

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

ユーザ追加後、postgres ユーザのパスワードを設定します。

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

※太斜線の部分にパスワードを指定します。

インスタンス/ データベース/ バックアップ用ディレクトリの作成

インスタンス用、データベース用、バックアップ用のディレクトリを作成し、所有者とアクセス権の設定を行います。本章の例では、インスタンス用とデータベース用のディレクトリを同一のディレクトリとします。

インスタンス／データベース用	: /PlusData/postgres
バックアップ用	: /PlusBackup/postgres

root ユーザで以下を実行します。

```
# mkdir -p /PlusData/postgres
# mkdir -p /PlusBackup/postgres
# chown postgres:postgres /PlusData/postgres
# chown postgres:postgres /PlusBackup/postgres
# chmod 700 /PlusData/postgres
# chmod 700 /PlusBackup/postgres
```

インスタンスの作成

インスタンスの作成は、PostgreSQL Plus Setup を使用して行います。

まず root ユーザで以下を実行します。

```
# xhost + 192.168.1.1
```

※太斜体の部分には、現用系サーバの IP アドレスまたはサーバ名を指定します。

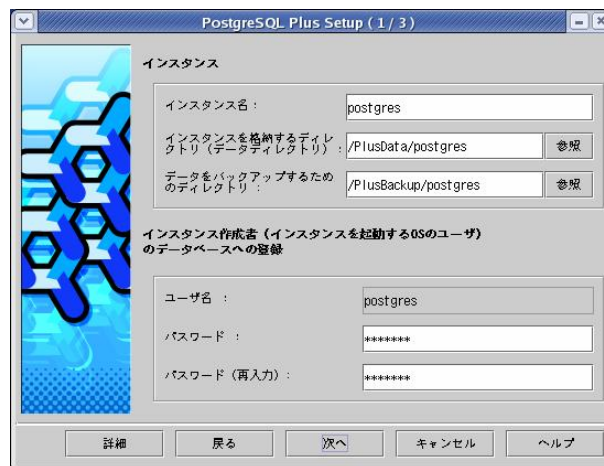
次に postgres ユーザにログオンして、環境変数を設定した後、PostgreSQL Plus Setup を起動します。

```
# su postgres
$ export PATH=/usr/local/pgsqlplus/bin:$PATH
$ /usr/local/pgsqlplus/bin/pgx_setup
```

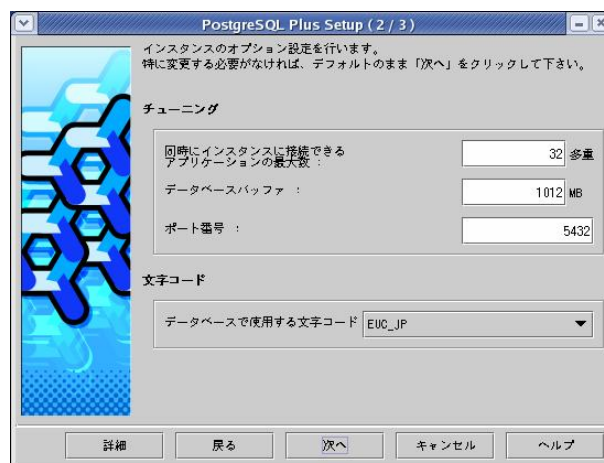
[PowerGres Plus へようこそ]の画面で[次へ]を選択します。

以下の画面では、それぞれ以下を指定します。

インスタンス名	: postgres
インスタンスを格納するディレクトリ(データディレクトリ)	: /PlusData/postgres
データをバックアップするためのディレクトリ	: /PlusBackup/postgres
パスワード/パスワード(再入力)	: postgres ユーザのパスワード



[ポート番号]は既定値(5432)を使用するため、変更しません。[データベースで使用する文字コード]を EUC_JP に変更します。



[PostgreSQL Plus Setup(3/3)]の画面では、[実行]を選択し、セットアップを実行します。

待機系へのセットアップ設定ファイルのコピー

「インスタンスの作成」の手順により作成されたセットアップ設定ファイルを待機系サーバにコピーします。ftp コマンド等により、以下のファイルを待機系サーバの同ディレクトリにコピーしてください。

```
/usr/local/pgsqlplus/package/pgxadmin/webapps/admin/postgres.html
```

※太斜体の部分は、「**インスタンス名.html**」になります。

PowerGres Plus の設定

インスタンス作成時に「インスタンスを格納するディレクトリ(データディレクトリ)」として指定したディレクトリの直下にある、pg_hba.conf ファイルを編集して、PowerGres Plus のクライアント接続認証設定を行います。本章の例では、以下のファイルを編集します。

```
/PlusData/postgres/pg_hba.conf
```

初期状態では、ループバック IP アドレスからの接続許可のみがあります。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。以下の一行を追加してください。

```
host all all 192.168.1.0/24 trust
```

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。

データベースの作成

PostgreSQL Plus Administrator を使用して、インスタンスの起動を実施後にデータベースの作成を行います。

Web ブラウザを開き、URL に以下を指定します。

```
http://localhost:13000/admin/postgres.html
```

※**13000** の部分は、PowerGres Plus のインストール時に設定したポート番号を指定します。

※**postgres** の部分は、「インスタンスの作成」で作成したインスタンス名を指定します。

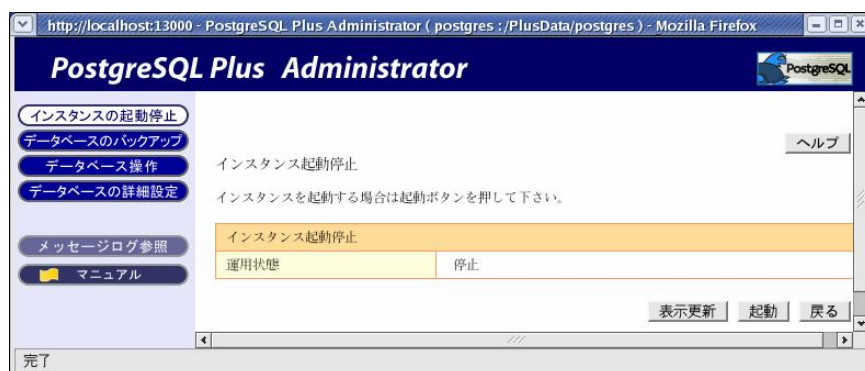
以下の画面で、[PostgreSQL Plus Administrator(Japanese)] を指定します。



[インスタンス作成者]と[パスワード]には、postgres ユーザとそのパスワードを指定します。



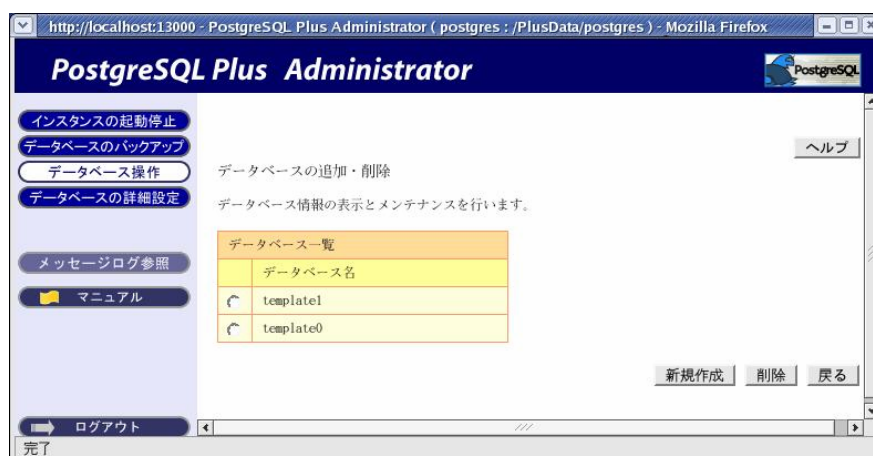
PostgreSQL Plus Administrator の左のメニューから[インスタンスの起動停止]を選択します。インスタンス:postgres が停止状態であることが表示されるので、[起動]を選択してインスタンスを起動します。



インスタンスの起動が正常に終了した旨の画面が表示されたら、[終了]を選択します。

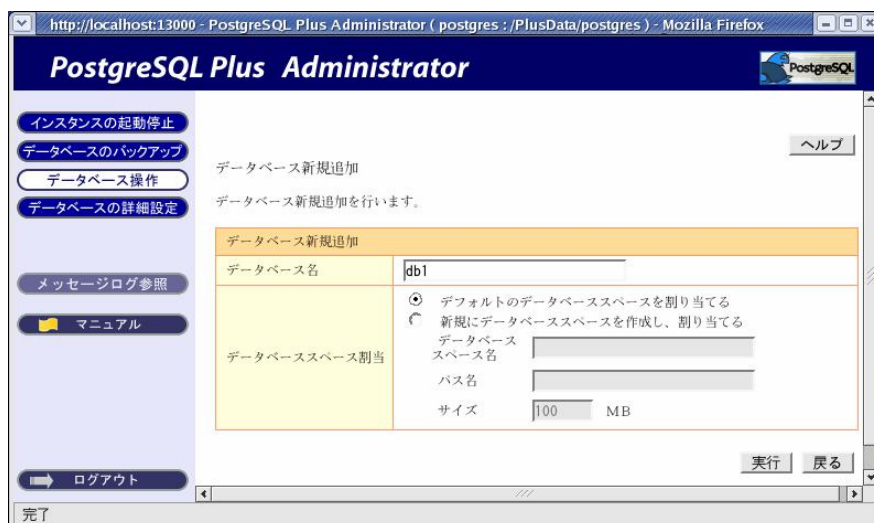
PostgreSQL Plus Administrator の左のメニューから[データベース操作]を選択し、[データベースの追加・削除]を選択します。

以下の画面で[新規作成]を指定します。



[データベース名]に db1 を指定し、[データベース割当]には、[デフォルトのデータベーススペースを割り当てる]にチェックして、[実行]を選択します。

新規作成するデータベース用に新しいデータベーススペースを割り当てる場合は、[新規にデータベーススペースを作成し、割り当てる]を選択して、必要な情報を設定します。



データベースの新規追加が正常に終了した旨の画面が表示されたら、[終了]を選択します。

[データベースの追加・削除]を選択した画面の[データベース一覧]に db1 が追加されていることを確認した後、インスタンスを停止します。

PostgreSQL Plus Administrator の左のメニューから[インスタンスの起動停止]を選択し、[停止]を選択してインスタンスを停止してください。

これで現用系サーバでの設定は終わりです。PostgreSQL Plus Administrator の画面をログアウトします。

待機系サーバでの PowerGres Plus の接続確認

待機系サーバでは、PowerGres Plus の設定を行う必要はありません。

待機系サーバでも現用系サーバで作成したデータベースに接続できることを確認します。

「現用系サーバでの PowerGres Plus の設定」の「データベースの作成」に記述してあるように、PostgreSQL Plus Administrator からインスタンス:postgres を起動します。

インスタンスの起動後、psql 等のクライアントツールを使用して、データベース:db1 に接続できることを確認してください。

接続が問題なく行えることを確認した後、PostgreSQL Plus Administrator からインスタンス:postgres を停止します。

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder を使用して、PowerGres Plus 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。

- (2) 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]を選択します。
- (3) [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。
- (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、「スクリプト内容例」のように記述します。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。

スクリプト内容例

PowerGres Plus の起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /PlusData/postgres/postmaster.pid ]
then
    rm -f /PlusData/postgres /postmaster.pid
fi
if [ -f /tmp/s.PGSQL.5432 ]
then
    rm -f /tmp/s.PGSQL.*
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 "PowerGres Plus start"
    fi

    #
    # Modify a PowerGres Plus user name, database domain directory,
    # and port number to appropriate values.
    #
    su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D
/Plusdata/postgres -l /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
```

----- (1)

----- (2)

----- (3)

```

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 " PowerGres Plus start"
    #
    # Modify a PowerGres Plus user name, database domain directory,
    # and port number to appropriate values.
    #
    su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D
/Plusdata/postgres -l /dev/null -o '-i -p 5432'"
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

- - - - - (3)

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

データベース用のディレクトリパス/postmaster.pid

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

ソケットファイルのパスs.PGSQL.ポート番号

(3) PowerGres Plus 起動処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理ユーザ -c "/usr/local/pgsqlplus/bin/pg_ctl start -D データベース用のディレクトリパス -l /dev/null -o '-i -p PowerGres Plus のポート番号'"
```

◆ stop.sh

```
#!/bin/sh
#*****
#*          stop.sh          *
#*****
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PowerGres Plus stop"
    #
    # Modify a PowerGres Plus user name and database domain directory to
    # appropriate values.
    #
    su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/postgres
    -m fast' ----- (1)
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$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 " PowerGres Plus stop"

#
# Modify a PowerGres Plus user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/postgres
-m fast' ----- (1)
        else
                echo "ERROR_DISK from FAILOVER"
        fi
else
        echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

(4) PowerGres Plus の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```

su - 管理者ユーザ -c "/usr/local/pgsqlplus/bin/pg_ctl stop -D データベース用のディレ
クトリパス -m fast

```

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
クラスタパーティション	D	
データパーティション	/dev/sda2、/dev/sdb1、/dev/sda3、/dev/sdb2 (※)	

※ /dev/sda2 と/dev/sdb1、/dev/sda3 と/dev/sdb2 はそれぞれ別のディスクに存在する。

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1	デバイス名	/dev/sda2
	マウントポイント	/PlusData
ディスクリソース2	デバイス名	/dev/sdb1
	マウントポイント	/PlusBackup
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1	デバイス名	/dev/sda3
	マウントポイント	/PlusData2
ディスクリソース2	デバイス名	/dev/sdb2
	マウントポイント	/PlusBackup2

PowerGres Plus情報

PowerGres Plusのバージョン	2.0	
インストールディレクトリ	/usr/local/pgsqlplus	
ユーザデータベース情報1	インスタンス名	postgres
	インスタンス作成ユーザ	postgres
	インスタンス情報/ユーザデータベース格納場所	/PlusData/postgres
	バックアップデータ格納場所	/PlusBackup/postgres
	データベース名	db1
	文字コード	EUC_JP
	ポート番号	5432
	クライアント認証方式	trust
ユーザデータベース情報2	インスタンス名	postgres2
	インスタンス作成ユーザ	postgres
	インスタンス情報/ユーザデータベース格納場所	/PlusData2/postgres2

	バックアップデータ格納場所	/PlusBackup2/postgres2
	データベース名	db2
	文字コード	EUC_JP
	ポート番号	5433
	クライアント認証方式	trust

CLUSTERPRO の設定

CLUSTERPRO の Builder を使用して、PowerGres Plus 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／仮想コンピュータ名リソース
- ◆ 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 1.0 for Linux インストール&設定ガイド「第 3 章 Builder で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。グループ 1 を Server1 で起動し、グループ 2 を待機系サーバで起動します。

PowerGres Plus のインストール

両サーバで PowerGres Plus のインストールを実施します。root ユーザで、以下を実施します。

```
# mount /media/cdrom
# cd /media/cdrom
# sh ./install.sh
```

※太斜体の部分は、CD 媒体のマウントポイントです。お使いの環境に合わせて読み替えてください。

現用系サーバでの PowerGres Plus の設定

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。以降の手順にある画面は、グループ 1 用のイメージとなります。グループ 2 の設定を行うときは、値を読み替えて設定してください。

インスタンス管理用ユーザの作成

インスタンス管理用のユーザ:postgres を OS に登録します。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する必要があります。両サーバで以下のように実行します。

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

ユーザ追加後、postgres ユーザのパスワードを設定します。

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

```
Retry new UNIX password: xxxxxx
```

※太斜線の部分にパスワードを指定します。

インスタンス/ データベース/ バックアップ用ディレクトリの作成

インスタンス用、データベース用、バックアップ用のディレクトリを作成し、所有者とアクセス権の設定を行います。本章の例では、インスタンス用とデータベース用のディレクトリは同一のディレクトリとします。

◆ Server1の場合

```
インスタンス／データベース用    : /PlusData/postgres
バックアップ用                  : /PlusBackup/postgres
```

◆ Server2の場合

```
インスタンス／データベース用    : /PlusData2/postgres2
バックアップ用                  : /PlusBackup2/postgres2
```

root ユーザで以下を実行します。

◆ Server1の場合

```
# mkdir -p /PlusData/postgres
# mkdir -p /PlusBackup/postgres
# chown postgres:postgres /PlusData/postgres
# chown postgres:postgres /PlusBackup/postgres
# chmod 700 /PlusData/postgres
# chmod 700 /PlusBackup/postgres
```

◆ Server2の場合

```
# mkdir -p /PlusData2/postgres2
# mkdir -p /PlusBackup2/postgres2
# chown postgres:postgres /PlusData2/postgres2
# chown postgres:postgres /PlusBackup2/postgres2
# chmod 700 /PlusData2/postgres2
# chmod 700 /PlusBackup2/postgres2
```

インスタンスの作成

インスタンスの作成は、PostgreSQL Plus Setup を使用して行います。

まず root ユーザで以下を実行します。

```
# xhost + 現用系サーバの IP アドレス
```

次に postgres ユーザにログオンして、環境変数を設定した後、PostgreSQL Plus Setup を起動します。

```
# su postgres
$ export PATH=/usr/local/pgsqlplus/bin:$PATH
```

```
$ /usr/local/pgsqlplus/bin/pgx_setup
```

[PowerGres Plus へようこそ]の画面で[次へ]を選択します。

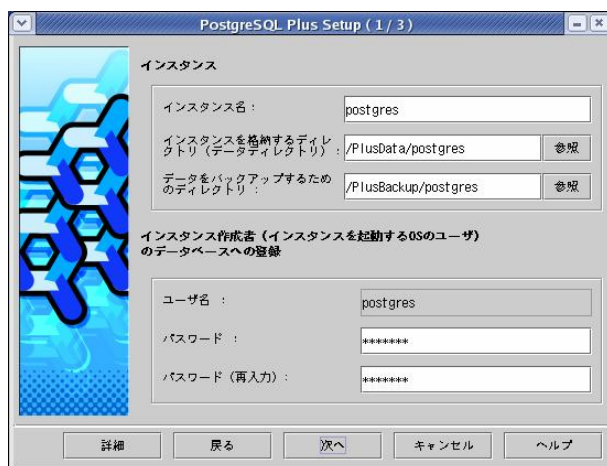
以下の画面では、それぞれ以下を指定します。

◆ Server1の場合

インスタンス名 : postgres
 インスタンスを格納するディレクトリ(データディレクトリ) : /PlusData/postgres
 データをバックアップするためのディレクトリ : /PlusBackup/postgres
 パスワード/パスワード(再入力) : postgres ユーザのパスワード

◆ Server2の場合

インスタンス名 : postgres2
 インスタンスを格納するディレクトリ(データディレクトリ) : /PlusData2/postgres2
 データをバックアップするためのディレクトリ : /PlusBackup2/postgres2
 パスワード/パスワード(再入力) : postgres ユーザのパスワード



Server1 の場合、サーバが使用するポートは既定値(5432)を使用する為、以下の画面の[ポート番号]の値は変更しません。Server2 の場合、[ポート番号]の値を 5433 に変更します。また、[データベースで使用する文字コード]を EUC_JP に変更します。



[PostgreSQL Plus Setup(3/3)]の画面では、[実行]を選択し、セットアップを実行します。

待機系へのセットアップ設定ファイルのコピー

「インスタンスの作成」の手順により作成されたセットアップ設定ファイルを待機系サーバにコピーします。ftp コマンド等により、以下のファイルを待機系サーバの同ディレクトリにコピーしてください。

◆ Server1の場合

```
/usr/local/pgsqlplus/package/pgxadmin/webapps/admin/postgres.html
```

◆ Server2の場合

```
/usr/local/pgsqlplus/package/pgxadmin/webapps/admin/postgres2.html
```

※太斜体の部分は、「**インスタンス名.html**」になります。

PowerGres Plus の設定

インスタンス作成時に「インスタンスを格納するディレクトリ(データディレクトリ)」として指定したディレクトリの直下にある、pg_hba.conf ファイルを編集して、PowerGres Plus のクライアント接続認証設定を行います。本章の例では、以下のファイルを編集します。

◆ Server1の場合

```
/PlusData/postgres/pg_hba.conf
```

◆ Server2の場合

```
/PlusData2/postgres2/pg_hba.conf
```

初期状態では、ループバック IP アドレスからの接続許可のみがあります。ここでは、192.168.1.0/24 からの接続を許可する設定を行います。以下の一行を追加してください。

```
host all all 192.168.1.0/24 trust
```

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。

データベースの作成

PostgreSQL Plus Administrator を使用して、インスタンスの起動を実施後にデータベースの作成を行います。

Web ブラウザを開き、URL に以下を指定します。

◆ Server1の場合

```
http://localhost:13000/admin/postgres.html
```

◆ Server2の場合

```
http://localhost:13000/admin/postgres2.html
```

※**13000** の部分は、PowerGres Plus のインストール時に設定したポート番号を指定します。

※**postgres/postgres2** の部分は、「インスタンスの作成」で作成したインスタンス名を指定します。

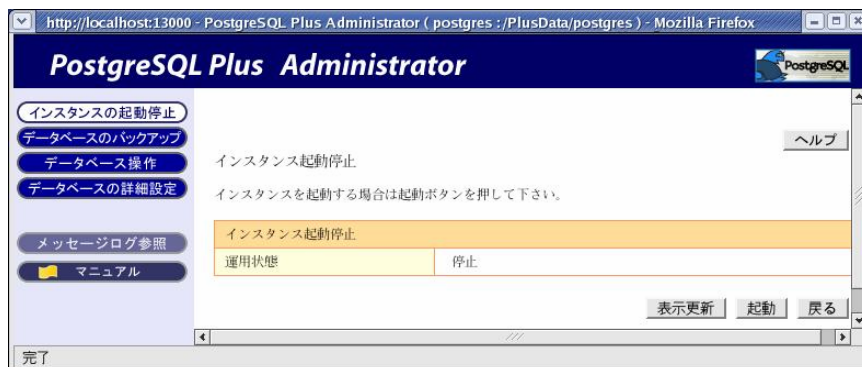
以下の画面で、[PostgreSQL Plus Administrator(Japanese)] を指定します。



[インスタンス作成者]と[パスワード]には、postgres ユーザとそのパスワードを指定します。



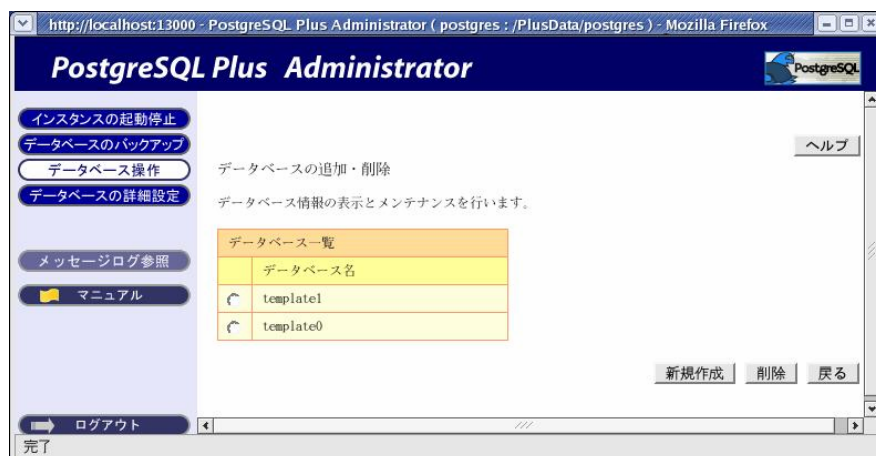
PostgreSQL Plus Administrator の左のメニューから[インスタンスの起動停止]を選択します。インスタンス:postgres(postgres2)が停止状態であることが表示されるので、[起動]を選択してインスタンスを起動します。



インスタンスの起動が正常に終了した旨の画面が表示されたら、[終了]を選択します。

PostgreSQL Plus Administrator の左のメニューから[データベース操作]を選択し、[データベースの追加・削除]を選択します。

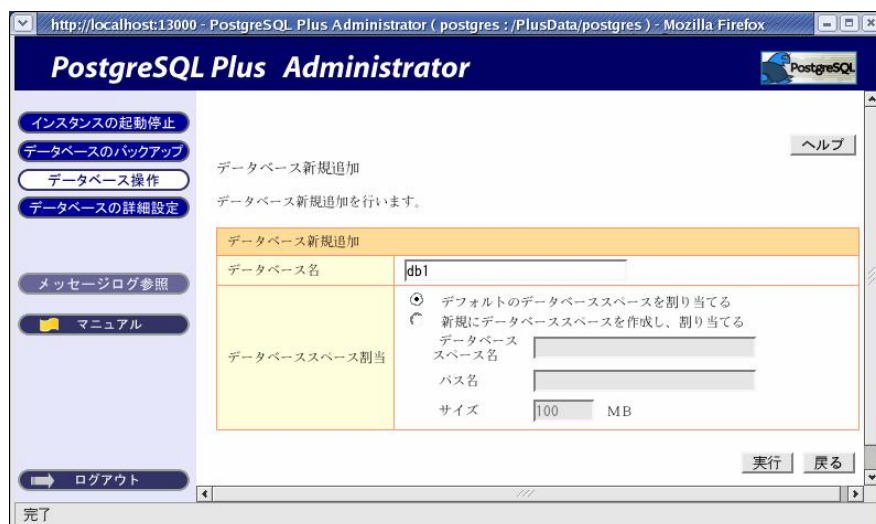
以下の画面で[新規作成]を指定します。



Server1 の場合、[データベース名]に db1 を指定し、[データベース割当]には、[デフォルトのデータベーススペースを割り当てる]にチェックして、[実行]を選択します。

Server2 の場合、[データベース名]に db2 を指定し、[データベース割当]には、[デフォルトのデータベーススペースを割り当てる]にチェックして、[実行]を選択します。

新規作成するデータベース用に新しいデータベーススペースを割り当てる場合は、[新規にデータベーススペースを作成し、割り当てる]を選択して、必要な情報を設定します。



データベースの新規追加が正常に終了した旨の画面が表示されたら、[終了]を選択します。

[データベースの追加・削除]を選択した画面の[データベース一覧]に db1(db2)が追加されていることを確認した後、インスタンスを停止します。

PostgreSQL Plus Administrator の左のメニューから[インスタンスの起動停止]を選択し、[停止]を選択してインスタンスを停止してください。

これで現用系サーバでの設定は終わりです。PostgreSQL Plus Administrator の画面をログアウトします。

待機系サーバでの PowerGres Plus の接続確認

待機系サーバでは、PowerGres Plus の設定を行う必要はありません。

待機系サーバでも現用系サーバで作成したデータベースに接続できることを確認します。

「現用系サーバでの PowerGres Plus の設定」の「データベースの作成」に記述してあるように、PostgreSQL Plus Administrator からインスタンス:postgres(postgres2)を起動します。

インスタンスの起動後、psql 等のクライアントツールを使用して、データベース:db1(db2)に接続できることを確認してください。

接続が問題なく行えることを確認した後、PostgreSQL Plus Administrator からインスタンス:postgres(postgres2)を停止します。

CLUSTERPRO の EXEC リソースの設定

CLUSTERPRO の Builder を使用して、PowerGres Plus 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) [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、「スクリプト内容例」のように記述します。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。

スクリプト内容例

PowerGres Plus の起動・停止用のスクリプト例です。太斜線の部分は環境に合わせて修正します。

◆ start.sh

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

if [ -f /PlusData/postgres/postmaster.pid ]
then
    rm -f /PlusData/postgres /postmaster.pid
fi
if [ -f /tmp/s.PGSQL.5432 ]
then
    rm -f /tmp/s.PGSQL.*
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 "PowerGres Plus start"
#
# Modify a PowerGres Plus user name, database domain directory,
# and port number to appropriate values.
#
su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D
/Plusdata/postgres -l /dev/null -o '-i -p 5432'"
        else
            echo "ERROR_DISK from START"
        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 " PowerGres Plus start"
#
# Modify a PowerGres Plus user name, database domain directory,
# and port number to appropriate values.
#
su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D
/Plusdata/postgres -l /dev/null -o '-i -p 5432'"
        else
            echo "ERROR_DISK from FAILOVER"
        fi
    else
        echo "NO_CLP"
    fi

```

----- (3)

----- (3)

```
echo "EXIT"  
exit 0
```

(1) pid ファイルの削除処理

pid ファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

```
データベース用のディレクトリパス/postmaster.pid
```

グループ 1 の場合は、「/PlusData/postgres/postmaster.pid」となります。

グループ 2 の場合は、「/PlusData2/postgres2/postmaster.pid」となります。

(2) ソケットファイルの削除処理

ソケットファイルのフルパスを指定します。太斜体の箇所は、環境に合わせて変更してください。

```
ソケットファイルのパス/s.PGSQL.ポート番号
```

(3) PowerGres Plus 起動処理

太斜体の箇所は、環境に合わせて変更してください。

```
su - 管理ユーザ -c "/usr/local/pgsqlplus/bin/pg_ctl start -D データベース用のディレクトリパス -l /dev/null -o '-i -p PowerGres Plus のポート番号'"
```

グループ 1 の場合は、「su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D /Plusdata/postgres -l /dev/null -o '-i -p 5432'""」となります。

グループ 2 の場合は、「su - postgres -c "/usr/local/pgsqlplus/bin/pg_ctl start -D /Plusdata2/postgres2 -l /dev/null -o '-i -p 5433'""」となります。

◆ stop.sh

```
#!/bin/sh
#*****
#*          stop.sh          *
#*****
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
        date +"%Y/%m/%d %T"
        echo "PowerGres Plus stop"
    #
    # Modify a PowerGres Plus user name and database domain directory to
    # appropriate values.
    #
    su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/postgres
    -m fast'
    else
        echo "ERROR_DISK from START"
    fi
elif [ "$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 " PowerGres Plus stop"
```

----- (1)

```

#
# Modify a PowerGres Plus user name, database domain directory, and port
# number to appropriate values.
#
su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl stop -D /PlusData/postgres
-m fast' ----- (1)
    else
        echo "ERROR_DISK from FAILOVER"
    fi
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0

```

(1) PowerGres Plus の停止処理

太斜体の箇所は、環境に合わせて変更してください。

```

su - 管理者ユーザ -c "/usr/local/pgsqlplus/bin/pg_ctl stop -D データベース用のディレクトリパス -m fast

```

グループ 1 の場合は、「su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl start -D /Plusdata/postgres -m fast'」となります。

グループ 2 の場合は、「su - postgres -c '/usr/local/pgsqlplus/bin/pg_ctl start -D /Plusdata2/postgres2 -m fast'」となります。