

CLUSTERPRO® X *for Linux*

PPガイド (Oracle® Database)

2023.09.27
第7.0版

CLUSTERPRO

改版履歴

版	日付	章	変更内容
1	2012/03	—	作成
2	2012/07	付録A, B	付録を追加
		1.2.3	ソフトウェア要件を更新 ・ OSのディストリビューションとkernel要件を更新 ・ ブラウザの要件を更新 ・ Red Hat Enterprise Linux 6 (x86-64)の kernel要件を追加 ・ Red Hat Enterprise Linux 6 (x86-64)の パッケージ要件を追加
		1.3.2	4)シェル制限の設定 Oracleマニュアルの設定項目変更に対応。 ・ /etc/pam.d/loginを/etc/pam.d/system-auth に変更 ・ stack制限を追加
		1.3.5.5	設定項目反映後のクラスタ開始手順を追加
		1.3.10.2	監視(レベル0)の異常判定条件を追加 ②データベースインスタンスが未起動 (SUSPENDED, INSTANCE RECOVERY)の場合
		1.3.10.2	監視(レベル3)を選択した場合の設定を変更
		1.3.2	カーネルパラメーターの設定 に ip_local_port_range の注 意事項を追加
		—	全体的に修正（文言の統一等）
2.1	2012/08	1.2.3	ソフトウェア要件を更新 ・ CLUSTERPROの対応ディストリビューションと kernel要件を更新 ・ Red Hat Enterprise Linux 6 (x86) の kernel要件を追加 ・ Red Hat Enterprise Linux 6 (x86) の パッケージ要件を追加
3	2014/07	—	全面改訂(Oracle Database12c R1対応他)
3.1	2015/04	1.3.2	設定手順を修正
		1.3.6	エラー発生時の対処を追加
		—	その他文章体裁等を修正
4	2017/09	—	全面改訂(Oracle Database12c R2対応他)
4.1	2018/03	4.2.4	ルール・ファイルの誤記を修正
		—	起動、停止スクリプトの誤記を修正
5	2019/04	—	全面改訂(CLUSTERPRO X 4.0 対応) 4.8.1 ノード間でコピーするファイルの誤記を修正 4.10.1 起動、停止スクリプトの誤記を修正
6	2020/09	—	全面改訂(CLUSTERPRO X 4.2 対応、Oracle Database19c 対応)
6.1	2020/11	—	全体的に修正（分かりにくい記載の見直し、文言の統一）
7	2023/09	—	全面改訂(CLUSTERPRO X 5.0 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、一切責任をおいません。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。

Microsoft、Windows、Windows Server、Internet Explorerは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

本書に記載されたその他の製品名および標語は、各社の商標または登録商標です。

その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。

参考URL

- NEC Oracle Response Center -NEORC- (<https://opendb.middle.nec.co.jp/oracle/>)
NECのOracle製品に関するサポートサービス・ポータルサイトです。保守契約締結の上、ユーザー登録をすることにより、24時間365日お客様からのお問い合わせに対応いたします。
- 日本オラクル株式会社 (<https://www.oracle.com/jp/>)
日本オラクル株式会社のホームページです。
- Oracle Technology Network -OTN- (<https://www.oracle.com/technetwork/jp/>)
オラクル社のポータルサイトです。ユーザー登録することでOracle製品のトライアル版やドキュメントをダウンロードすることができます。
- My Oracle Support (<https://support.oracle.com/>)
オラクル社のOracle製品サポート情報サイトです。オラクル社は本サイトにて、製品の技術情報、パッチ情報、不具合情報等のサポート情報を提供しています。保守契約を締結していれば、本サイトによって、Oracle製品に関する技術情報が参照可能です。
- 従来Oracle KROWN拡張検索(KROWN) 、およびKROWNディレクトリ・サービス(KDS)にて提供していた技術ナレッジ文書はMy Oracle Support(MOS)に移行しました。
MOSの検索バーに「XXXXX」(XXXXXはDocID)と入力することで、本手順で引用しているDocIDを使用して、目的の技術情報にアクセス可能です。

目次

1. はじめに.....	5
1.1 対象読者と目的.....	5
1.2 適用範囲	5
1.3 お問い合わせについて.....	5
1.4 障害発生時	5
1.5 CLUSTERPROマニュアル体系.....	6
2. 機能概要.....	7
2.1 構成概要	7
2.2 業務運用中の障害時フェイルオーバー動作	8
2.3 バックアップ・リカバリ中の障害時フェイルオーバー動作.....	8
3. 構成要件.....	9
3.1 ハードウェア・OS要件	10
3.2 ネットワーク・IPアドレス要件.....	13
3.3 ソフトウェア要件	14
3.4 その他考慮事項.....	18
4. 構築手順.....	19
4.1 前提環境	19
4.2 ハードウェア構成後の手順	21
4.3 環境構築の流れ.....	35
4.4 CLUSTERPRO Serverのインストールとライセンス登録.....	36
4.5 CLUSTERPROの設定①.....	38
4.6 Oracle Database ソフトウェアのインストール	55
4.7 データベースの作成	62
4.8 データベース作成後の作業	76
4.9 リスナーの作成.....	78
4.10 CLUSTERPROの設定②.....	80
4.11 クライアントの設定	96
5. 注意事項.....	97
6. 付録.....	98
6.1 双方向スタンバイ構成の場合の設定例	98
6.2 ライセンス定義.....	109

1. はじめに

1.1 対象読者と目的

『CLUSTERPRO® PPガイド』は、クラスタシステムに関して、システムを構築する管理者、およびユーザ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。

1.2 適用範囲

本書は、CLUSTERPRO X 5.0 for Linuxおよび、Oracle® Database 19cの組み合わせで検証しています。

また、12cからの機能であるContainer Database(CDB)のシングルテナント構成と従来の非CDB構成を対象としています。

CDBの詳細は、下記資料をご参照ください。

-
- Oracle® Multitenant 管理者ガイド 19c
(https://docs.oracle.com/cd/F19136_01/multi/index.html)
○第I部 マルチテナント・アーキテクチャ
-

1.3 お問い合わせについて

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致します。

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

本書のOracle製品に関する記載内容のお問い合わせには、原則としてCLUSTERPROの保守契約とOracleの弊社での保守契約が必要です。

1.4 障害発生時

Oracle Databaseの障害発生時には、保守契約に則り、NEC Oracle Response Center (NEORC)に問い合わせてください。

1.5 CLUSTERPROマニュアル体系

CLUSTERPROのマニュアルは、以下の5つに分類されます。各ガイドのタイトルと役割を以下に示します。

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

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

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

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

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

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象とし、CLUSTERPRO の運用手順、各モジュールの機能説明およびトラブルシューティング情報等を記載します。『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。

最新情報の入手先

最新の製品情報は、以下のWebサイトを参照してください。

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

マニュアルは、以下のWebサイトを参照してください。

<https://www.support.nec.co.jp/View.aspx?NoClear=on&id=3140108622>

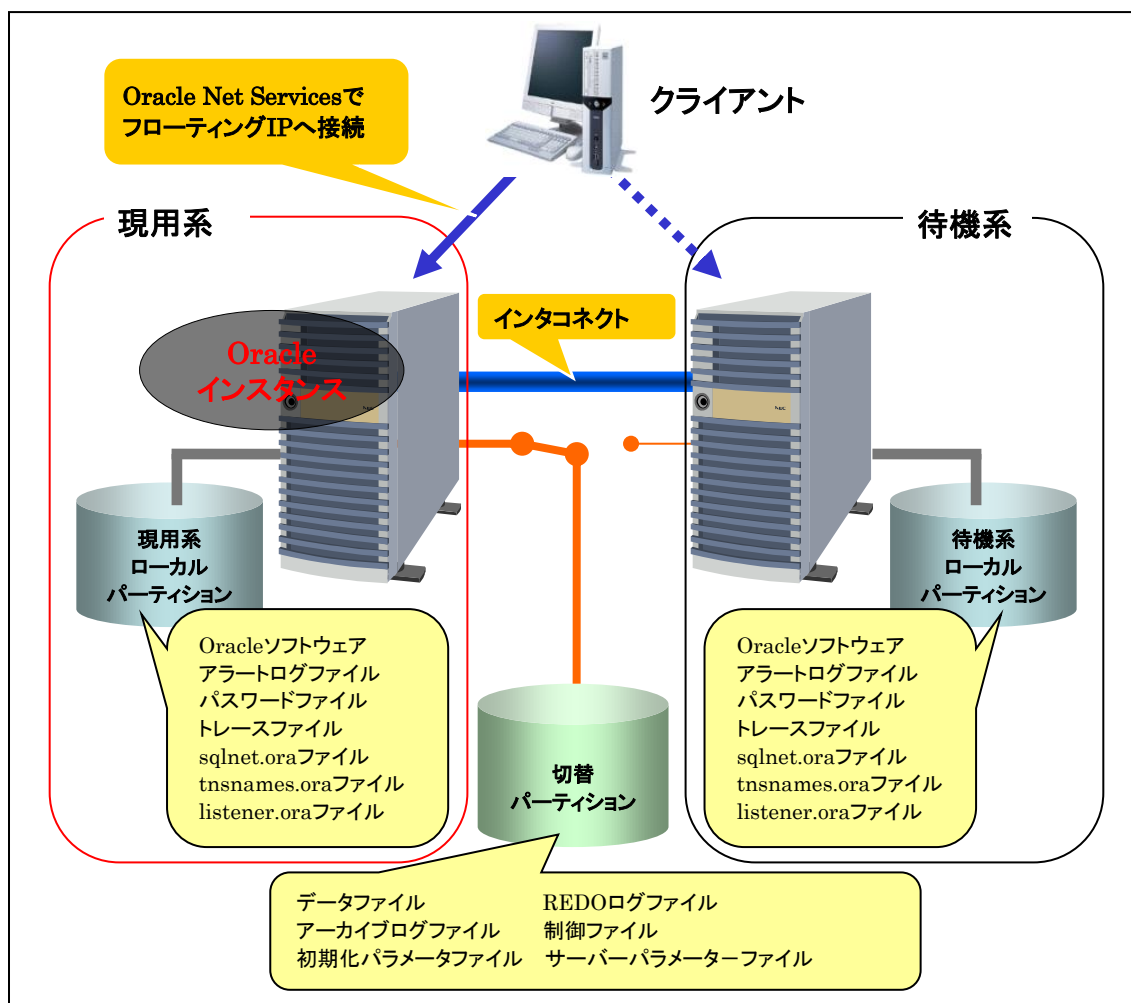
2. 機能概要

本章ではOracle DatabaseをCLUSTERPRO環境下で利用する際の機能概要について記述します。

2.1 構成概要

下図はCLUSTERPRO 環境下でのOracle Databaseの運用のイメージです。クライアントは、通常Oracle Net Servicesで現用系にアクセスします。現用系に障害が発生した場合、待機系でスタートアップのスクリプトに応じてOracle Databaseインスタンスが立ち上がり、クライアントは待機系に接続し、再度運用することが可能です。

片方向スタンバイの共有ディスク構成イメージ



本ドキュメントでは、共有ディスク構成の場合、データベース・ファイルは切替パーティション上に配置し、Oracle Databaseソフトウェアはそれぞれのサーバーのローカルパーティションにインストールすることを前提としています。

またミラーディスク構成の場合、データベース・ファイルはミラーディスク上のデータパーティションに作成し、Oracle Databaseソフトウェアはミラーリングを行わないローカルパーティションにインストールすることを前提としています。なお、非同期ミラー機能は、データベースの一貫性に対し十分に検証されていないため、本手順書では未対象となります。

※ 共有ディスク構成で、サーバー間のデータ引き継ぎに使用するパーティションを「切替パーティション」と呼び、ミラーディスク構成でデータを格納し、ミラーリングの対象となるパーティションを「データパーティション」と呼びます。

2.2 業務運用中の障害時フェイルオーバー動作

現用系の障害によりフェイルオーバーが発生した場合、Oracle Databaseは待機系での立ち上げ時にコミット済のデータをデータベースに反映し、コミットされていないデータをロールバックすることによって、データベースを正常に保ちます。

フローティングIPアドレス(FIP)を使用して接続する場合、フェイルオーバー後も同一のIPアドレスで再接続が可能です。

2.3 バックアップ・リカバリ中の障害時フェイルオーバー動作

コールドバックアップ中にフェイルオーバーが発生した場合には、再度、フェイルオーバー時点のバックアップからやり直す必要があります。ホットバックアップ中にフェイルオーバーが発生した場合は、フェイルオーバー先でリカバリ処理を行った後、バックアップをやり直す必要があります。リカバリ中にフェイルオーバーが発生した場合には、フェイルオーバー先でリカバリを継続して下さい。ただし、フェイルオーバーのタイミングによっては、データベースがリカバリを継続できない状態になる場合があります。この場合、再度バックアップからデータを戻した上で、リカバリ処理をはじめからやり直す必要があります。

3. 構成要件

本章および次章で、本手順書をご利用いただくにあたり、あらかじめご留意いただきたい要件をまとめて抜粋します。ご利用頂く構成が、CLUSTERPROとOracle Database両方の要件を満たしていることを、ご確認ください。

CLUSTERPROの要件詳細は、下記資料をご参照ください。

-
- CLUSTERPRO X 5.0 システム構築ガイド
(<https://www.support.nec.co.jp/View.aspx?NoClear=on&id=3140108622>)
→Linux
○スタートアップガイド
-

Oracle Database 19cの要件詳細は、下記資料をご参照ください。

-
- Oracle Database 19c スタート・ガイド
(https://docs.oracle.com/cd/F19136_01/index.html)
→インストールとアップグレード
→Databaseインストレーション・ガイドfor Linux
○Oracle® Database Databaseインストレーション・ガイド 19c for Linux
-

※ なお、Oracle Database要件の最新情報は、3ページ『参考URL』の「My Oracle Support」からご確認ください。

3.1 ハードウェア・OS要件

3.1.1 メモリ要件

CLUSTERPRO X 5.0と、Oracle Database 19cのメモリの最低要件¹は以下のとおりです。

製品	メモリの最低要件
CLUSTERPRO X 5.0の要件	200(*1) MB
Oracle Database 19cの要件	2 GB
合計	2.2 GB

(*1) オプション類を除く

物理メモリは、以下のコマンドで確認できます。

```
# grep MemTotal /proc/meminfo
MemTotal:      8010812 kB
```

3.1.2 スワップ領域

搭載した物理メモリに応じて、以下のとおりスワップ領域を確保して下さい。

【最小スワップ領域】

使用可能な物理メモリ	必要なスワップ領域
1GB以上2GB未満	物理メモリの1.5倍
2GB以上16GB未満	物理メモリと同等
16GB以上	16GB

スワップ領域は、以下のコマンドで確認できます。

```
# grep SwapTotal /proc/meminfo
SwapTotal:      8314872 kB
```

¹ CLUSTERPRO X 5.0のメモリ要件である200MBは、ユーザーモードで運用した場合の値です。カーネルモードで運用する場合は、CLUSTERPRO X5.0のスタートアップガイド 第4章「CLUSTERPROの動作環境」の「必要メモリ容量とディスクサイズ」に記載された計算式から算出する必要があります。

3.1.3 ディスク要件

パーティション作成要件については以下を想定しています。ここでは作成はせず、実際の共有ディスクおよびミラーディスクのパーティション作成手順は、**章番号『4.2.3 パーティションの作成』**を確認ください。

■共有ディスク構成の場合、ノード間で共有するディスクが必要です。
必要なパーティションやディレクトリは、以下のとおりです。

- ・ディスクハートビート用パーティションを確保します。要件は以下のとおりです。
 - ー容量は10MB以上必要
 - ーパーティションIDは83(Linux)とする
 - ーファイル・システムは構築しない
- ・ディスクリソース用切替パーティションを確保します。要件は以下のとおりです。
 - ー共有ディスクのファイル・システムを、OSの/etc/fstabにエントリしない
 - ーパーティションIDは83(Linux)とする
 - ーファイル・システムは、Linuxでサポートされているものを下記から選択する
 - ext3
 - ext4
 - xfs
 - reiserfs
- ・あらかじめ、ディスクリソース用パーティションをmountするディレクトリを作成します。
ディレクトリは、ディスクリソースを使用する全てのノードで作成します。

■ミラーディスク構成の場合、ミラー用のディスクか、ミラー用の空きパーティションが必要です。
必要なパーティションやディレクトリは、以下のとおりです。

- ・クラスタパーティションを確保します。要件は以下のとおりです。
 - ー容量は1024MB以上必要
 - ーパーティションIDは83(Linux)とする
 - ーファイル・システムの構築は必要ない
- ・データパーティションを確保します。要件は以下のとおりです。
 - ーミラーディスクのファイル・システムを、OSの/etc/fstabにエントリしない
 - ーパーティションIDは83(Linux)とする
 - ーファイル・システムは、Linuxでサポートされているものを下記から選択する
 - ext2
 - ext3
 - ext4
 - xfs
 - reiserfs
- ・あらかじめ、ミラーリソース用パーティションをmountするディレクトリを作成します。
ディレクトリは、ミラーリソースを使用する全てのノードで作成します。

■/tmpディレクトリに1 GB以上のディスク領域が必要です。

/tmp領域は、以下のコマンドで確認できます。

```
# df -h /tmp
```

- CLUSTERPRO X 5.0と、Oracle Database 19c Enterprise Editionをインストールする
ディスクに必要な領域²は、下記のとおりです。

製品	ディスクの最低要件
CLUSTERPRO X 5.0の要件 ²	5.0 GB + 1.0 GB ³
Oracle Database 19cの要件	7.2 GB
合計	12.2 GB + 1.0 GB ³

- CLUSTERPRO X5.0と、Oracle Database 19c Standard Editionをインストールする
ディスクに必要な領域は、下記のとおりです。

製品	ディスクの最低要件
CLUSTERPRO X 5.0の要件 ²	5.0 GB + 1.0 GB ³
Oracle Database 19c R2の要件	7.2 GB
合計	12.2 GB + 1.0 GB ³

インストールする領域は、以下のコマンドで確認できます。

```
# df -h インストールする領域のパス
```

² CLUSTERPRO X 5.0の要件は、運用時の最大値を記載しています。

³ ミラーディスクリソースおよびハイブリッドディスクリソース使用時に必要なディスクサイズです。

3.2 ネットワーク・IPアドレス要件

3.2.1 NIC要件

本手順書では、パブリックLANとインタコネクトLAN用に2つのNICを使用した構成で記載しています。

※NIC優先度の注意点

フローティングIPと同一ネットワークに属するNICが複数存在する場合、フローティングIPを作成するNICの優先度が高くなるように設定してください。

3.2.2 IPアドレス要件

各ノードにパブリックIPアドレスが1つ、インタコネクトIPアドレスが1つ、クラスタ全体でフローティングIPアドレスが1つが必要です。両ノード合わせて以下の静的IPアドレスが必要です。フローティングIPアドレス以外は、hostsファイルで名前解決しておきます。サーバーで設定しているホスト名とCLUSTERPROのサーバー名は一致する必要があります。ホスト名は小文字で設定して下さい。

ーパブリックIPアドレス ×2
ーインタコネクトIPアドレス ×2
ーフローティングIPアドレス ×1

※フローティングIPアドレス

フローティングIPアドレスは、CLUSTERPROによって使用されるIPアドレスです。
クラスタサーバーが所属するLANと、同じネットワークアドレス内で、かつ使用していないアドレスである必要があります。

【片方向スタンバイ環境のネットワーク構成例】

種類	ホスト名	IPアドレス	
○ノード1			
パブリックIP	node1	固定	10.0.0.1
インタコネクトIP	node1-in	固定	192.168.0.1
○ノード2			
パブリックIP	node2	固定	10.0.0.2
インタコネクトIP	node2-in	固定	192.168.0.2
○フローティングIP			
フローティングIP1		固定	10.0.0.11

なお、本構成では通常運用時にデータベースsid1を起動するノードをnode1、待機系となるノードをnode2とします。

※双方向スタンバイの場合のネットワーク構成例は、

『6.1.1 双方向スタンバイ構成の場合のネットワーク・IPアドレス要件例』をご確認ください。

3.3 ソフトウェア要件

3.3.1 OSのディストリビューションとkernel要件

CLUSTERPRO ServerおよびOracle Databaseは、動作可能なOSのディストリビューションとkernelが、厳密に定められています。ご利用頂くOSのディストリビューションとkernelが、CLUSTERPROとOracle Database両方の要件を満たしていることを、ご確認ください。

CLUSTERPROのディストリビューションとkernel要件は、下記資料をご参照ください。

-
- CLUSTERPRO X - 動作環境
(<https://www.support.nec.co.jp/View.aspx?NoClear=on&id=3140108622>)
→動作確認済みkernel一覧
→x86_64
○CLUSTERPRO X 5.0 動作確認済みディストリビューションおよびkernel一覧
-

Oracle Database 19cのディストリビューションとkernel要件は、下記資料をご参照ください。

-
- Oracle Database データベース・インストレージョン・ガイド 19c for Linux
(https://docs.oracle.com/cd/F19136_01/ladbi/index.html)
→4 LinuxでのOracle Databaseのオペレーティング・システムの構成
○x86-64 Linuxプラットフォームのオペレーティング・システム要件
-

※ なお、Oracle Database要件の最新情報は、3ページ『参考URL』の「My Oracle Support」からご確認ください。

ディストリビューションは、以下のコマンドで確認できます。

```
# cat /etc/redhat-release
```

kernelのバージョンは、以下のコマンドで確認できます。

```
# uname -r
```

次にOracle Database 19cに必要なパッケージのインストール作業を行います。
OS上に、以下のパッケージが全てインストールされていることを確認し、不足している
パッケージはインストールして下さい。
本手順では、ノード1、ノード2の両方に対して行います。

インストールされているパッケージは、以下のコマンドで確認できます。

```
# rpm -q パッケージ名
```

パッケージのインストールは、以下のコマンドを使用します。

```
# rpm -ivh インストールするパッケージ名
```

Oracle Database 19cのパッケージ要件については以下をご確認ください。

【Linux x86-64】

OS	必要なパッケージ
Red Hat Enterprise Linux 8 (x86-64)	次のパッケージの最新リリース・バージョンをインストールします。 bc binutils elfutils-libelf elfutils-libelf-devel fontconfig-devel glibc glibc-devel ksh libaio libaio-devel libXrender libX11 libXau libXi libXtst libgcc libnsl librdmacm libstdc++ libstdc++-devel libxcb libibverbs make policycoreutils policycoreutils-python-utils smartmontools

	sysstat 注意:32ビットのクライアント・アプリケーションを使用して64ビットのサーバーにアクセスする場合は、この表にリストされているパッケージの最新の32ビット・バージョンもインストールする必要があります(使用可能な場合)。
--	---

3.3.2 ブラウザの要件

Cluster WebUIを動作させるために必要な環境について記載します。

※2022/4/26 時点の情報です。

最新情報はCLUSTERPRO X 5.0のスタートアップガイドをご参照ください。

動作確認済みOS,ブラウザ

ブラウザ	言語
Firefox	日本語/英語/中国語
Google Chrome	日本語/英語/中国語
Microsoft Edge (Chromium)	日本語/英語/中国語

※Cluster WebUI利用時の注意点

- ・http://<IPアドレス>:29003で接続する際には、該当のIPアドレスを[ローカル イントラネット]の[サイト]に登録する必要があります。
最新情報はCLUSTERPROのWebサイトを参照してください。
- ・タブレットやスマートフォンなどのモバイルデバイスには対応していません。

3.4 その他考慮事項

■Enterprise Manager

軽量のWebベースの管理ツールであるOracle Enterprise Managerを提供しています。

Oracle Database 12cではOracle Database 10g/11gで使用されていたDatabase Controlが廃止され、Oracle Enterprise Manager Database Express (以下、EM Express)が新たに実装されました。

Database ControlのWindowsプラットフォーム上でのアクティブ・スタンバイ構成はオラクル社からサポートされませんでしたが、EM Expressではサポートされるようになりました。

EM Expressの詳細は、該当するドキュメント⁴をご確認ください。

なお、当該ドキュメントにはEM Expressのポート競合の回避方法も提示されています。双方向スタンバイ構成の場合は、ドキュメントの手順が必要です。

■Oracle監視リソースについて

CLUSTERPROのオプション製品である「CLUSTERPRO X Database Agent 5.0 for Linux」により、「Oracle 監視リソース」の機能を使って、Oracle Databaseの監視を行うことが出来ます。

本手順書では、章番号『4.10.2 モニタリソースの作成』で、「Oracle監視リソース」の設定手順を記載しています。「Oracle監視リソース」を使用する場合は、上記のページをご参考下さい。

⁴ DocID 1766802.1 「EM Express の構成方法・注意点」

4. 構築手順

本章以降で、データベース・タイプに依存した手順がある場合、各手順内に以下2つの表記方法で記載します。

- －【CDB(シングルテナント)構成のデータベースを作成する場合】
- －【非CDB構成のデータベースを作成する場合】

4.1 前提環境

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構成を想定し、構築手順の例を紹介します。本手順は共有ディスク構成とミラーディスク構成の両方を対象としています。

クラスタサーバー環境		
	node1	node2
実 IP アドレス	10.0.0.1	10.0.0.2
ローカルデバイス	/dev/sda	/dev/sda

フェイルオーバーグループ情報(グループ 1)		
	node1	node2
フローティング IP アドレス	10.0.0.11	
切替パーティション, データパーティション	/dev/sdb2-> /dev/cp-diska2 (シンボリック・リンク) (マウントポイント : /oradata/sid1)	
クラスタパーティション ※ミラーディスク構成の場合	/dev/sdb1-> /dev/cp-diska1 (シンボリック・リンク) (マウントポイント : /oradata/sid1)	

データベース環境		
	node1	node2
ORACLE_BASE	/u01/app/oracle	
ORACLE_HOME	/u01/app/oracle/product/19.0.0/dbhome_1	
SID 名	sid1	
データベース名	sid1	
PDB 名 (CDB 構成のみ)	sid1pdb	
データファイル格納先	/oradata/sid1	
REDO ログファイル格納先		
制御ファイル格納先		
アーカイブログファイルの出力先	/oradata/sid1/fast_recovery_area/SID1	
初期化パラメーターファイル格納先 (テキスト形式)	/oradata/sid1	
サーバーパラメーターファイル格納先		

リスナー環境		
	node1	node2
リスナー名	LISTENER	
FIP アドレス/ポート番号	10.0.0.11:1521	

共有ディスク構成の場合、トレースファイルおよびログファイルの出力先は、ローカルパーティション／切替パーティションのどちらを指定しても構いません。ただし、以下の点に注意して下さい。

- ・ローカルパーティション上に配置する場合
現用系／待機系で名前が同じファイル（内容は異なる）を二重管理する必要がありますが、切替パーティションでの障害の影響を受けません。

- ・切替パーティション上に配置する場合
ファイルを一元管理することができますが、切替パーティションで障害が発生した場合にトレースならびにログ情報が記録されないなどの様々な影響を受けることがあります。

※双方向スタンバイの場合の前提は、
『6.1.3 双方向スタンバイ構成の場合の前提環境』を確認してください。

上記以外の要件は、日本オラクル株式会社のホームページのシステム要件を参照してください。

4.2 ハードウェア構成後の手順

4.2.1 Oracle用OSグループ、Oracle管理ユーザーの作成⁵

本手順は、ノード1，ノード2の両方に対して行います。
また、本手順はrootユーザーで行ってください。

Oracle用OSグループを作成します。網掛け部分は環境に合わせて適宜読み替えてください。

```
# groupadd -g 1001 oinstall
# groupadd -g 1002 dba
# groupadd -g 1003 oper
# groupadd -g 1004 backupdba
# groupadd -g 1005 dgdba
# groupadd -g 1006 kmdba
# groupadd -g 1007 racdba
```

Oracle管理ユーザーを作成します。網掛け部分は環境に合わせて適宜読み替えてください。

```
# useradd -u 1100 -g oinstall -G dba,oper,backupdba,dgdba,kmdba,racdba oracle
```

各ノード上で、作成したユーザーの情報が一致しているかを確認します。

```
# id oracle
uid=1100(oracle) gid=1001(oinstall)
groups=1001(oinstall),1002(dba),1003(oper),1004(backupdba),1005(dgdba),1006(kmdba),1007(racdba)
```

oracleユーザーのパスワードを設定してください。

```
# passwd oracle
```

4.2.2 Oracle用インストールディレクトリの作成

本手順は、ノード1，ノード2の両方に対して行います。
本構成では、以下のディレクトリにOracle Databaseをインストールします。

製品	用途	パス
Oracle Database	Oracleベース <Oracle_Base>	/u01/app/oracle
	Oracleホーム <Oracle_Home>	/u01/app/oracle/product/19.0.0/dbhome_1

※ <Oracle_Base>,<Oracle_Home>は環境変数ではありません。

Oracle Database をインストールするディレクトリの作成例は、以下のとおりです。

本手順は、root ユーザーで実行します。

```
# mkdir -p /u01/app/oracle
# mkdir -p /u01/app/oraInventory
# chown -R oracle:oinstall /u01/app/oracle
# chown -R oracle:oinstall /u01/app/oraInventory
# chmod -R 775 /u01/app
```

⁵ OSユーザーやOSグループの詳細は、Oracle社のガイドを参照してください。

「Oracle® Database Databaseインストール・ガイド 19c for Linux」

4.2.3 パーティションの作成

本編では、パーティションを以下のとおりに作成します。

共有ディスク構成の場合

パーティション		用途	備考
デバイス名	フォーマット		
/dev/sdb1	しない	ディスクハートビート用	
/dev/sdb2	ext4	sid1用のディスクリソース (データファイル等)	“/oradata/sid1” にマウントする

ミラーディスク構成の場合

パーティション		用途	備考
デバイス名	フォーマット		
/dev/sdb1	しない	クラスタパーティション	
/dev/sdb2	ext4	sid1用のディスクリソース (データファイル等)	“/oradata/sid1” にマウントする

※双方向スタンバイの場合の構成は、『6.1.2 双方向スタンバイ構成の場合のディスクパーティション作成例』を確認してください。

作成手例は以下のとおりです。

本手順は、全てrootユーザーで実行します。

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の作成

※ディスクハートビート用パーティションの場合は片方のノードで実行したら、もう片方のノードから実行する必要はありません。クラスタパーティションの場合は両ノードで実行してください。

(パーティション・テーブルのリストを確認)

```
# fdisk -l
```

(パーティションを作成)

```
# fdisk /dev/sdb
```

コマンド (m でヘルプ): p ← 領域テーブルの表示

ディスク /dev/sdb: 21.5 GB, 21474836480 バイト

ヘッド 64, セクタ 32, シリンダ 20480

Units = シリンダ数 of 2048 * 512 = 1048576 バイト

セクタサイズ (論理 / 物理): 512 バイト / 512 バイト

I/O size (minimum/optimal): 512 bytes / 512 bytes

ディスク識別子: 0xaf8ffc16

デバイス	ブート	始点	終点	ブロック	Id	システム
------	-----	----	----	------	----	------

コマンド (m でヘルプ): n ← パーティションの新規作成

コマンドアクション

e 拡張

p 基本パーティション (1-4)

p ← 基本領域の指定

パーティション番号 (1-4): 1 ← 領域番号 1 の指定

最初 シリンダ (1-20480, 初期値 1): (default のまま Enter)

Last シリンダ, +シリンダ数 or +size [K, M, G] (1-20480, 初期値 20480): +20M ← サイズ 20MB の指定 (クラスタパーティションの場合は 1024MB 以上の値を入力してください)

コマンド (m でヘルプ): t ←パーティションのコードを変更
 16 進数コード (L コマンドでコードリスト表示): 83 ←Linux 領域に設定

コマンド (m でヘルプ): w ← パーティションの確定
 パーティションテーブルは変更されました！

ioctl() を呼び出してパーティションテーブルを再読み込みします。
 ディスクを同期しています。

※クラスタパーティションの場合は以下の通り実施してください。

- ・片方向スタンバイ構成・ミラーディスク構成の場合は上記手順を両ノードで実施してください。
- ・双方向スタンバイ構成・ミラーディスク構成の場合は上記手順と同様に
 両ノードで/dev/sdb1,/dev/sdc1を作成します。

ディスクリソース用パーティションの作成

※片方のノードで実行したら、もう片方のノードから実行する必要はありません。

fdisk /dev/sdb
 コマンド (m でヘルプ): p ← 領域テーブルの表示

ディスク /dev/sdb: 21.5 GB, 21474836480 バイト
 ヘッド 64, セクタ 32, シリンダ 20480
 Units = シリンダ数 of 2048 * 512 = 1048576 バイト
 セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
 I/O size (minimum/optimal): 512 bytes / 512 bytes
 ディスク識別子: 0xaf8ffc16

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sdb1		1	21	21488	83	Linux

コマンド (m でヘルプ): n ← パーティションの新規作成
 コマンドアクション
 e 拡張
 p 基本パーティション (1-4)

p ← 基本領域の指定
 パーティション番号 (1-4): 2 ← 領域番号 2 の指定
 最初 シリンダ (22-20480, 初期値 22): (default のまま Enter)
 初期値 22 を使います
 Last シリンダ, +シリンダ数 or +size {K,M,G} (22-20480, 初期値 20480)
 : (default のまま Enter)
 初期値 20480 を使います

コマンド (m でヘルプ): t ←パーティションのコードを変更
 パーティション番号 (1-4): 2 ←領域番号を選択
 16 進数コード (L コマンドでコードリスト表示): 83 ←Linux 領域に設定

コマンド (m でヘルプ): w ← パーティションの確定
 パーティションテーブルは変更されました！

ioctl() を呼び出してパーティションテーブルを再読み込みします。
 ディスクを同期しています。

※片方向スタンバイ構成・ミラーディスク構成の場合は上記手順を両ノードで実施してください。
※双方向スタンバイ構成・共有ディスク構成の場合は上記手順と同様に/dev/sdb2,/dev/sdc1を作成します。
※双方向スタンバイ構成・ミラーディスク構成の場合は上記手順と同様に両ノードで/dev/sdb2,/dev/sdc2を作成します。

ディスクリソース用パーティションにext4ファイル・システムを作成

※共有ディスク構成では片方のノードで実行したら、もう片方のノードから実行する必要はありません。ミラーディスク構成の場合は両ノードで実行してください。

```
# /sbin/mkfs -t ext4 /dev/sdb2

# /sbin/mkfs -t ext4 /dev/sdc1 ← 双方向スタンバイ構成・共有ディスク構成のみ実施。
# /sbin/mkfs -t ext4 /dev/sdc2 ← 双方向スタンバイ構成・ミラーディスク構成のみ両ノードで実施。
```

(必要なファイル・システムを全て作成した後、各ノードのOSを再起動して設定を反映します。)

```
# systemctl reboot
```

ディスクリソース用パーティションをマウントするディレクトリの作成

※両ノードで実行する必要があります。
※網掛けのパーミッションは、要件に応じて変更して下さい。

```
# mkdir -p /oradata/sid1
# chmod -R 770 /oradata
# chown -R oracle:oinstall /oradata
```

(以下は双方向スタンバイ構成時のみ必要)

```
# mkdir -p /oradata/sid2
# chmod -R 770 /oradata
# chown -R oracle:oinstall /oradata
```

パーティションマウント先の権限を変更

※共有ディスク構成では片方のノードで実行したら、もう片方のノードから実行する必要はありません。ミラーディスク構成の場合は両ノードで実行してください。

```
# mount /dev/sdb2 /oradata/sid1
# chmod -R 770 /oradata
# chown -R oracle:oinstall /oradata
```

(以下は双方向スタンバイ構成・共有ディスク構成時のみ必要)

```
# mount /dev/sdc1 /oradata/sid2
# chmod -R 770 /oradata
# chown -R oracle:oinstall /oradata
```

(以下は双方向スタンバイ構成・ミラーディスク構成時のみ両ノードで必要)

```
# mount /dev/sdc2 /oradata/sid2
# chmod -R 770 /oradata
# chown -R oracle:oinstall /oradata
```

作成したディスクリソース用パーティションのサイズは、パーティションをマウント後に以下のコマンドで確認できます。

ディスクリソース用パーティションのサイズを確認する

```
# df -h /oradata/sid1
# df -h /oradata/sid2 ←双方向スタンバイ構成時のみ
```

4.2.4 ディスクデバイスファイルへのシンボリック・リンク付与

OSがディスクデバイスに対して付与する論理デバイス名(/dev/sdb等)は、ディスクの削除/追加等の要因でずれる可能性があるため、本編ではデバイスに対してシンボリック・リンクを設定します。CLUSTERPROから設定するディスク名としてシンボリック・リンクを指定することで、論理デバイス名がずれた場合でも正しいディスクへアクセスできます。

※ストレージパスの冗長化ソフトウェア(iStorage StoragePathSaviorなど)を使用し、論理ディスクデバイス名が変化しても問題ない構成であれば本手順は不要です。

※ディスクデバイスファイルへの設定方法は、OSバージョンにより手順/設定方法が異なる可能性があります。

以下に設定例を示します。

一意のデバイス識別子(ID_SERIAL)を確認

(ノード1で確認) ※ミラーディスク構成の場合は両ノードで確認してください。

```
[root@node1 ~]# /lib/udev/scsi_id --whitelisted --device=/dev/sdb  
36000c2921090ca3ec5c367f1fdf877b6
```

(双方向スタンバイ時のみ以下も確認)

```
[root@node1 ~]# /lib/udev/scsi_id --whitelisted --device=/dev/sdc  
36000c29fecdc731acaf5e4c839221c84
```

ルール・ファイルを作成

共有ディスク構成の場合：ノード1で作成後、ノード2へコピーしてください。

ミラーディスク構成の場合：各ノードの識別子を元に各ノードでファイルを作成してください。

```
# vi /etc/udev/rules.d/99-clusterpro-devices.rules
```

(ファイル内に以下を記述する)

```
KERNEL=="sd*[^0-9]", ENV{ID_SERIAL}="", IMPORT{program}="/lib/udev/scsi_id --  
whitelisted --device=/dev/%k"  
KERNEL=="sd*[0-9]", ENV{ID_SERIAL}="", IMPORT{parent}="ID_*"  
ENV{ID_SERIAL}=="36000c2921090ca3ec5c367f1fdf877b6", SYMLINK+="cp-diska%n"
```

(以下は双方向スタンバイ構成時のみ必要)

```
ENV{ID_SERIAL}=="36000c29fecdc731acaf5e4c839221c84", SYMLINK+="cp-diskb%n"
```

(以下は共有ディスク構成時のみ必要、scpコマンドでコピーを行う例)

```
# scp /etc/udev/rules.d/99-clusterpro-devices.rules node2:/etc/udev/rules.d
```

※網掛け部分は確認したデバイス識別子(ID_SERIAL)を設定してください。

※scpコマンド初回実行時、以下の表示がされた場合は「yes」と入力してください。

Are you sure you want to continue connecting (yes/no)?

引き続き、パスワードについて表示がされるので相手ノードのパスワードを入力してください。

※scpコマンド実行時、パスワードについて表示がされた場合は相手ノードのパスワードを入力してください。

ファイル作成後に再起動実行 (ノード1、ノード2で実施)

```
# systemctl reboot
```

再起動後にシンボリック・リンク設定の確認（ノード1、ノード2で実施）

※下記例は共有ディスク構成での結果を表示しています。

```
# ls -l /dev/cp-*  
lrwxrwxrwx. 1 root root 3  6月 15 15:34 /dev/cp-diska -> sdb  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diska1 -> sdb1  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diska2 -> sdb2  
(以下は双方向スタンバイ構成時のみ出力)  
lrwxrwxrwx. 1 root root 3  6月 15 15:34 /dev/cp-diskb -> sdc  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diskb1 -> sdc1
```

※下記例はミラーディスク構成での結果を表示しています。

```
# ls -l /dev/cp-*  
lrwxrwxrwx. 1 root root 3  6月 15 15:34 /dev/cp-diska -> sdb  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diska1 -> sdb1  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diska2 -> sdb2  
(以下は双方向スタンバイ構成時のみ出力)  
lrwxrwxrwx. 1 root root 3  6月 15 15:34 /dev/cp-diskb -> sdc  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diskb1 -> sdc1  
lrwxrwxrwx. 1 root root 4  6月 15 15:34 /dev/cp-diskb2 -> sdc2
```

4.2.5 カーネルパラメーターの設定

OSのカーネルパラメーターを両ノードで設定します。
本手順は全てrootユーザーで実行します。

① 設定値の確認

カーネルパラメーターの設定値を確認し、以下に示す推奨値以上の値に設定されていることを確認してください。なお、shmmaxに関しては、注意事項があります。ドキュメント⁶を参照の上、設定値を決定してください。

パラメーター	値	確認方法
semmsl semmns semopm semmni	250 32000 100 128	# /sbin/sysctl -a grep sem
shmmax	・物理メモリ・サイズの半分(バイト)	# /sbin/sysctl -a grep shm
shmmni shmall	4096 ・物理メモリ・サイズの40% (ページ数) (※1)	
file-max	6815744	
ip_local_port_range	最小:9000 最大:65500 (※2)	# /sbin/sysctl -a grep ip_local_port_range
rmem_default	262144	# /sbin/sysctl -a grep net.core.rmem
rmem_max	4194304	
wmem_default	262144	# /sbin/sysctl -a grep net.core.wmem
wmem_max	1048576	
aio-max-nr	1048576 (※3)	# /sbin/sysctl -a grep aio-max-nr
panic_on_oops	1	# /sbin/sysctl -a grep panic_on_oops

※1 1ページあたりのサイズは以下のコマンドで確認できます。

```
# getconf PAGE_SIZE  
4096
```

また、複数のデータベースがサーバーにある場合、または大規模SGAを使用している場合、この値は共有メモリの値と同じにしてください。

※2 この例ではOSが自動的に割り当てるポート番号(エフェメラルポート)を、Oracle Databaseの推奨値である9000～65500に設定しております。

CLUSTERPROがデフォルトで使用するポート番号は、29001以降です。
(クラスタの構成に依存して使用するポート番号が異なりますので詳細な情報はCLUSTERPROのスタートアップガイド「通信ポート番号」を参照ください)
エフェメラルポートの範囲とCLUSTERPROが使用するポート番号が重複しないように以下 (a), (b) 何れかの設定変更を推奨します。

⁶ DocID 1747840.1 「[Linux] Linux x86 及び x86-64 環境の SHMMAX の最大値について」

- (a)下記②の /etc/sysctl.confファイルで net.ipv4.ip_local_port_range の範囲とCLUSTERPROが使用するポート番号が重複しないようにする。
(設定例) net.ipv4.ip_local_port_range = 30000 65500

ただし、(a)の対処を取る場合は、必要に応じて設定したエフェメラル・ポート範囲が、予想されるシステム・ワークロードに十分対応できることを確認して下さい。

その上で、(a)の対処により、Oracle Universal InstallerからOracle Databaseをインストールする際に発生する、エフェメラル・ポート範囲に関する警告は無視できます。

詳しくは、下記マニュアルを確認して下さい。

■ Oracle Technology Network -OTN- (<https://www.oracle.com/technetwork/jp>)
○ Oracle® Database Database インストレーション・ガイド 19c for Linux

- (b)CLUSTERPROが使用するポート番号を9000～65500以外に変更する。

※3 aio-max-nr に設定する値は、環境等に合わせて適宜設定してください。
本構成では 1048576 としています。詳細は該当ドキュメント⁷をご参照ください。

② /etc/sysctl.conf ファイルへの書込み

/etc/sysctl.confファイルを編集して、マシン起動時にカーネルパラメーターを設定するように設定します。sysctl.confファイルを編集後、パラメーター値を即時に反映したい場合は、/sbin/sysctl -p コマンドを実行します。

```
# vi /etc/sysctl.conf
```

(以下、本構成での設定例です)

```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 30000 65500
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 1048576
kernel.panic_on_oops = 1
```

```
# /sbin/sysctl -p
```

(パラメーター値を即時に反映します)

4.2.6 シェル制限の設定

oracleユーザーのシェル制限を、両ノードで設定します。
本手順は、特別な記載がない限りrootユーザーで実行します。

ディストリビューション固有でOSユーザーのリソース制限の設定方法が異なることがあります。
設定方法はOSのシステム管理者や、OSベンダーへ確認してください。

⁷ DocID 1747749.1 「aio-max-nr パラメータについて (Linux)」

① /etc/security/limits.conf ファイルの修正

```
# vi /etc/security/limits.conf
(最終行” # End of file” の上に以下を書込みます。次の例では、対象ユーザーに oracle ユーザーを指定しています)
oracle          soft    nproc    2047
oracle          hard    nproc    16384
oracle          soft    nofile   1024
oracle          hard    nofile   65536
oracle          soft    stack    10240
oracle          hard    stack    32768
```

② /etc/pam.d/system-auth ファイルの修正

```
# vi /etc/pam.d/system-auth
(下記行がない場合は追加します)
session    required    pam_limits.so
```

※上記の行の存在確認は、以下のコマンドで確認ができます。

```
# cat /etc/pam.d/system-auth | grep pam_limits.so
```

③ /etc/profile ファイルの修正

```
# vi /etc/profile
(最終行以降に以下を書込みます。次の例では、対象ユーザーに oracle ユーザーを指定しています)
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -u 16384
        ulimit -n 65536
        ulimit -s 32768
    fi
    umask 022
fi
```

④ ノードの再起動

```
(変更を反映するため、ノードを再起動します)
# systemctl reboot
```

⑤ oracle ユーザーのシェル制限設定の確認

```
(oracle ユーザーでログイン後、以下を確認します)
$ ulimit -Su
(戻り値が 2047 であることを確認します)
$ ulimit -Hu
(戻り値が 16384 であることを確認します)
$ ulimit -Sn
(戻り値が 1024 であることを確認します)
$ ulimit -Hn
(戻り値が 65536 であることを確認します)
$ ulimit -Ss
(戻り値が 10240 であることを確認します)
$ ulimit -Hs
(戻り値が 32768 であることを確認します)
```

4.2.7 OS起動時間を調整する

CLUSTERPROの要件として、クラスタシステムを構成する各ノードの電源を投入してから、OSが起動するまでの時間を、以下の2つより長くなるように調整する必要があります。

- ① 共有ストレージに電源を投入してから使用可能になるまでの時間
(ミラーディスク構成では不要)
- ② ハートビートタイムアウト時間
既定値は90秒です。

この調整は、以下の問題を回避するために必要です。

- 共有ストレージと物理マシンの電源を入れてクラスタシステムを起動した時に、OSが起動するまでの時間が①よりも短い場合、CLUSTERPROによるディスクリソースの活性に失敗する
- ノードの再起動でフェイルオーバーを発生させたい場合に、ハートビートタイムアウト時間内にそのノードが再起動してしまうと、相手側からはハートビートが継続しているとみなされ、フェイルオーバーが発生しない

上記①、②の時間を計測後、以下の手順例に従ってOS起動時間を調整します。

以下、設定例です。ノード1、ノード2の両方に対して行います。

本設定例では、OSローダにGRUBを使用しており、①、②の時間が90秒の前提とします。

- ・ /etc/default/grub を編集します。

GRUB_TIMEOUT <起動時間 (単位は秒)> オプションを指定します。

下記の例の場合には、網掛け部分を90秒に設定して下さい。

```
GRUB_TIMEOUT=90
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb
quiet"
GRUB_DISABLE_RECOVERY="true"
```

(パラメーター値を反映します)

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

4.2.8 ファイアウォールの設定を確認する

CLUSTERPROはいくつかのポート番号を使用します。ファイアウォールの設定を変更して、CLUSTERPROが当該ポート番号を使用できるように設定してください。

ポート番号一覧は、CLUSTERPRO X 5.0のスタートアップガイドをご確認ください。

Oracle Databaseではリスナーのポート番号を使用します。ファイアウォールの設定を変更して、Oracle Clientから当該ポート番号を使用できるように設定してください。

4.2.9 ノードの時刻を同期させる

クラスタシステムでは、クラスタ内の全てのノードの時刻を定期的に同期する運用を推奨します。ntp等を使用して、ノードの時刻を同期させるよう設定して下さい。

ノードの時刻同期を行わない場合、Oracle Databaseで以下の問題が発生する可能性があります。問題の詳細は該当ドキュメント⁸をご確認ください。

※各ノードの時刻が同期されていない場合、障害時に原因の解析に時間がかかることがあります。

4.2.10 ネットワーク設定を確認する

クラスタ内のすべてのノードで、ifconfigコマンドやpingコマンドを使用して、以下のネットワークリソースが正常に動作しているかどうかを確認します。

ここではpingコマンドを用いたノード間の疎通確認を両方のノードで行う正常動作確認の実行例を示します。

```
■パブリックLAN ホスト名
# ping node1
# ping node2

■パブリックLAN IPアドレス
# ping 10.0.0.1
# ping 10.0.0.2

■インタコネクト専用LAN ホスト名
# ping node1-in
# ping node2-in

■インタコネクト専用LAN IPアドレス
# ping 192.168.0.1
# ping 192.168.0.2
```

※それぞれのpingコマンドを実行した際に、一定間隔で以下の出力が得られる事を確認ください。
64 byte from [入力ホスト名 | 入力IPアドレス]: icmp_sec=N ttl=TT time=X.XXX

※フローティングIPアドレスは現時点では設定していないため、接続テストは不要です

⁸ DocID 1704324.1 「システム時刻の変更にともなう注意点」

4.2.11 SELinuxの設定を確認する

環境構築を実施する前にSELinuxを無効または有効にします。

SELinuxが有効な状態でCLUSTERPROを使用する場合には、CLUSTERPROのドライバのロードに失敗する場合がありますので、事前に許可設定をおこないます。この作業はノード1、ノード2の両方で実施してください。

- ① SELinux が有効になっているか無効になっているか、確認します。

```
# getenforce
Enforcing
```

表示される単語の意味は以下のとおりです。

- Enforcing : 有効
- Permissive : SELinuxポリシーが強制されずに、システムは動作し続け、代わりにワーニングがログ出力される
- Disabled : 無効 (SELinuxポリシーはロードされない)

- ② SELinux を無効にする場合は、ファイル `/etc/selinux/config` 内の `SELINUX=enforceing` の行を編集して `enforceing` を `disabled` または `permissive` へ変更します。

SELinux を有効にする場合には、`enforcing` へ変更します。

SELINUX=の後に指定する単語の意味は以下のとおりです。

- enforcing : 有効
- permissive : SELinuxポリシーが強制されずに、システムは動作し続け、代わりにワーニングがログ出力される
- disabled : 無効 (SELinuxポリシーはロードされない)

```
(root ユーザーで実施します)
# vi /etc/selinux/config

(コメント文 This file controls ...)
SELINUX=permissive
```

編集完了後にシステムを再起動し、再度`getenforce`コマンドでSELinuxのステータスを確認します。

```
# systemctl reboot
```

(システム再起動後)

```
# getenforce
Permissive
```

- ③ SELinux を有効 (`enforcing`) にして CLUSTERPRO を使用する場合は、CLUSTERPRO のドライバのロードを事前に許可しておく必要があります。

その場合には、以降の手順をおこなってください。

- ④ 下記のパッケージをインストールします。

- RHEL8 系の場合

```
# dnf -y install selinux-policy-mls
# dnf -y install selinux-policy-devel
```

• RHEL7 系の場合

```
# yum -y install selinux-policy-mls  
# yum -y install selinux-policy-devel
```

- ⑤ 作業用のディレクトリを作成して、そのディレクトリへ移動します。

```
# mkdir -p /tmp/te  
# cd /tmp/te
```

- ⑥ clpka.ko ドライバ用の.te ファイルを作成します。

```
# vi clpka.te
```

clpka.teのファイル内容：

```
# clpka.te  
module clpka 1.0;  
  
require {  
    type unconfined_service_t;  
    type usr_t;  
    class system module_load;  
}  
  
#----- unconfined_service_t -----  
allow unconfined_service_t usr_t:system module_load;
```

- ⑦ clpkhb.ko ドライバ用の.te ファイルを作成します。

```
# vi clpkhb.te
```

clpkhb.teのファイル内容：

```
# clpkhb.te  
module clpkhb 1.0;  
  
require {  
    type unconfined_service_t;  
    type usr_t;  
    class system module_load;  
}  
  
#----- unconfined_service_t -----  
allow unconfined_service_t usr_t:system module_load;
```

- ⑧ liscal.ko ドライバ用の.te ファイルを作成します。

```
# vi liscal.te
```

liscal.teのファイル内容：

```
# liscal.te  
module liscal 1.0;  
  
require {  
    type unconfined_service_t;
```

```
        type usr_t;
        class system module_load;
    }

#----- unconfined_service_t -----
allow unconfined_service_t usr_t:system module_load;
```

- ⑨ 下記コマンドを実行します。

パッケージポリシーファイルが作成されて、インストールされます。

```
# make -f /usr/share/selinux/devel/Makefile
# semodule -i clpka.pp clpkhb.pp liscal.pp
```

- ⑩ パッケージポリシーファイルが3つともインストールされているか、確認します。

```
# semodule -l | grep clp
clpka
clpkhb

# semodule -l | grep liscal
liscal
```

※作成した作業ディレクトリは、削除してもかまいません。

※クラスタ生成後に、必要なドライバがロードされていることを確認してください。

- **clpka** は、ユーザ空間モニタリソースの監視方法として、**keepalive** をしている場合に、クラスタ開始時にロードされます。
また、シャットダウン監視の監視方法として **keepalive** を指定している場合に、シャットダウン監視開始時にロードされます。
- **clpkhb** は、インタコネクにカーネルハートビートを設定している場合や、**clpka** を使用する場合にロードされます。
- **liscal** は、ミラーディスクリソースやハイブリッドディスクリソースが設定されている場合にロードされます。

```
# lsmod | grep clp
clpka
clpkhb

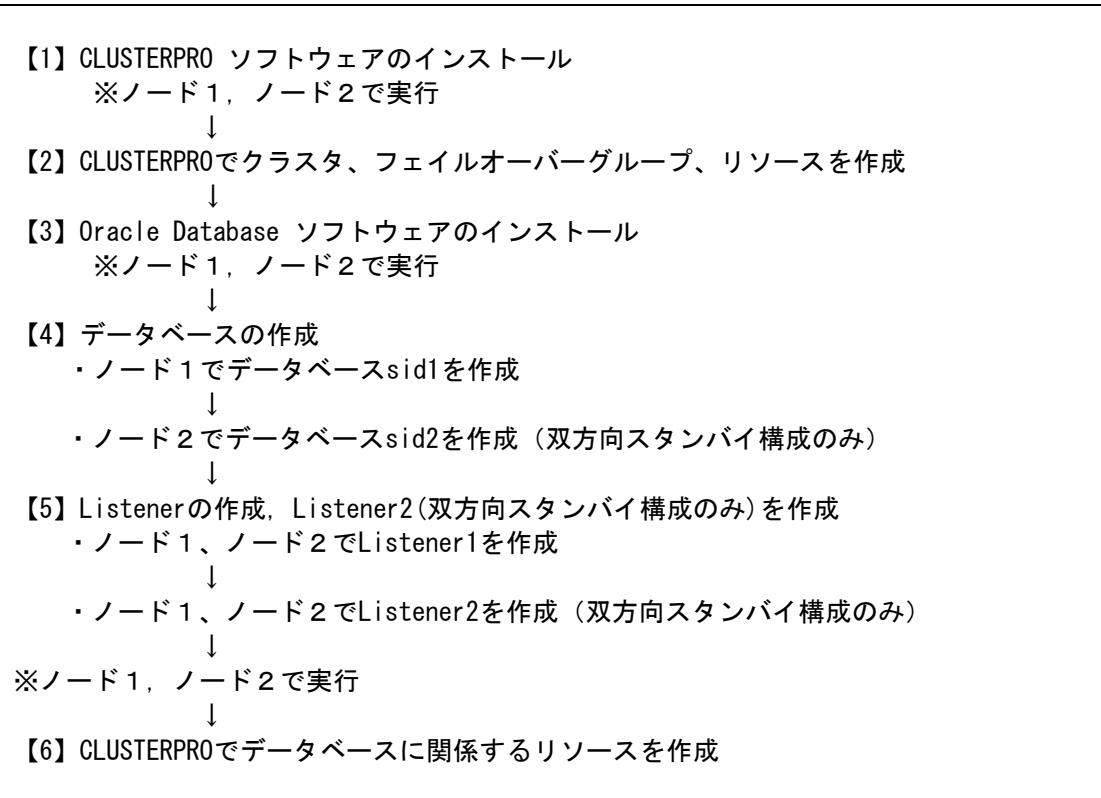
# lsmod | grep liscal
liscal
```

4.2.12 セキュアブートの設定

セキュアブートの設定は無効化してください。

4.3 環境構築の流れ

環境の構築は、以下の手順で行います。



本手順では、CLUSTERPROをインストールし、最初に必要なクラスタ、フェイルオーバーグループ、リソースを作成します。

その後、Oracle Databaseのインストールを行い、データベースとリスナーの作成を行います。最後に、CLUSTERPROでデータベースに関するリソースを作成し、環境構築は完了です。

4.4 CLUSTERPRO Serverのインストールとライセンス登録

4.4.1 CLUSTERPRO Serverのインストール

以下の手順に従って、クラスタを構成する各ノードにCLUSTERPRO ServerのRPMをインストールします。

本手順はrootユーザーで実行して下さい。

※両ノードで実行が必要になります。

1. インストールMediaをmountします。
2. rpmコマンドを実行して、パッケージファイルをインストールします。
Media内の/Linux/5.0/jp/serverに移動した後、下記のコマンドを実行します。

網掛けの部分は、実際に使用するRPMファイルを確認の上、バージョンとアーキテクチャをご入力下さい。

```
# rpm -ivh clusterpro-5.0.0-1.x86_64.rpm
```

コマンドを入力後、インストールが開始されます。

なお、CLUSTERPROは、"/opt/nec/clusterpro"にインストールされますが、このディレクトリを変更するとアンインストールできなくなるため、注意して下さい。

CLUSTERPRO Serverのインストールに関する詳細は、CLUSTERPROの下記マニュアルを確認して下さい。

-
- CLUSTERPRO X 5.0 システム構築ガイド
(<https://www.support.nec.co.jp/View.aspx?NoClear=on&id=3140108622>)
→Linux
○インストール&設定ガイド
-

4.4.2 ライセンス登録

構築するクラスタシステムを実際に動作させるには、ライセンスを登録する必要があります。

ライセンス製品名
CLUSTERPRO X 5.0 for Linux
CLUSTERPRO X Replicator 5.0 for Linux
CLUSTERPRO X Database Agent 5.0 for Linux

Replicatorは、ミラーディスク構成を構築する際、必要なライセンスです。

Database Agentは、データベース監視機能であるOracle監視リソースを使用する際、必要なライセンスです。

以下の手順で、ライセンスを登録します。

なお、共有ディスク構成ではReplicatorは不要のため、下記手順からReplicatorライセンスの登録を実行しない様お願い致します。

下記の各コマンドを実行後に表示されるメッセージに従ってライセンス情報を入力してください。本手順はrootユーザーで実行して下さい。

【ノード1で実行】

```
# clplcncsc -i filepath(CPUライセンス)
# clplcncsc -i filepath(Database Agent)
# clplcncsc -i filepath(Replicator)
```

【ノード2で実行】

```
# clplcncsc -i filepath(Database Agent)
# clplcncsc -i filepath(Replicator)
```

※網掛け部分は、実際のライセンス・ファイル・パスをご入力ください。

※CPUライセンスはマスターサーバーとするサーバーにのみ登録します。

各オプション製品は、各サーバーにライセンスを登録する必要があります。

※ライセンス登録を削除する場合、以下のコマンドを実行します。

```
(製品版ライセンスの場合、削除したいライセンスのシリアルナンバーを指定)
# clplcncsc -d Serial No
(試用版ライセンスの場合、一括削除し再度ライセンスを登録)
# clplcncsc -d -t
```

登録したライセンスを確認するには、以下のコマンドを実行します。

```
# clplcncsc -l -a
```

最後に、ライセンス登録を有効にし、クラスタを稼動するために各ノードを再起動します。

```
# systemctl reboot
```

以上で、CLUSTERPROのインストールとライセンス登録は終了です。

4.5 CLUSTERPROの設定①

Cluster WebUIから、以下の順序でクラスタを作成していきます。

- 1). クラスタ
- 2). フェイルオーバーグループ
- 3). リソース
 - ーフローティングIP(以下fip)リソース
 - ーディスクリソース
 - ースクリプトリソース (データベース構築後に作成)
 - ーモニタリソース (データベース構築後に作成)

フェイルオーバーグループによって、業務を継続するために必要なリソースをまとめ、業務単位でフェイルオーバーを行うことが可能です。

各リソースの詳細は、CLUSTERPROのシステム構築ガイドを確認して下さい。

※以降の画面は、Oracle Database 19cの場合です。

※フェイルオーバーグループはデータベース毎に作成する必要があります。

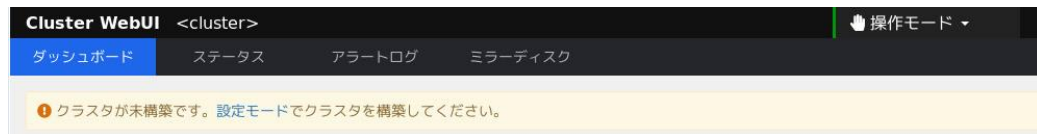
4.5.1 クラスタの作成

本章では、Cluster WebUIからクラスタを作成する手順を紹介します。

- 1) ブラウザに下記のアドレスを入力し、Cluster WebUIを起動します。
※どちらか片方のノードで実行してください。

`http://10.0.0.1:29003/`

- 2) クラスタが未構築の場合、Cluster WebUIを起動させると下記のメッセージが表示されますので「設定モード」を選択してください。
「クラスタが未構築です。設定モードでクラスタを構築してください。」



「設定モード」に切替わったら、「クラスタ生成ウィザード」をクリックします。



※既に作成しているクラスタを再構築する場合

「設定モード」画面にて作成済みのクラスタの「プロパティ」を選択します。



- 3) クラスタ名（ここではclusterとしています）を設定し、「次へ」をクリックします。



クラスタ生成ウィザード

クラスタ → サーバ → インタコネクト → フェンシング → グループ → モニタ

クラスタ名* cluster

コメント

言語* 日本語

管理IPアドレス

① クラスタの生成を開始します。
クラスタの名前を入力して、WebManagerを動作させる環境の言語（ロケール）を選択してください。
統合WebManagerで複数のクラスタを管理する場合、クラスタ名でクラスタを識別するため、重複しない名前を設定してください。
管理IPアドレスはWebManagerの接続に使用するフローティングIPアドレスです。各サーバのIPアドレスを指定して接続する場合は省略可能です。
継続するには[次へ]をクリックしてください。

戻る 次へ キャンセル

- 4) 「追加」をクリックし、node2を追加します。追加したら「OK」でウィンドウを閉じ、「次へ」をクリックします。



クラスタ生成ウィザード

クラスタ → サーバ → インタコネクト → フェンシング → グループ → モニタ

追加 削除

サーバの定義一覧

順位	名前
マスタサーバ	node1

↑ ↓

サーバグループの設定 設定

① 「追加」ボタンを押して、クラスタを構成するサーバを追加します。
サーバの優先順位は「↑」、「↓」ボタンで変更します。
サーバグループを使用する場合は「設定」ボタンでサーバグループを設定します。

OK キャンセル

戻る 次へ キャンセル



クラスタ生成ウィザード

クラスタ → サーバ → インタコネクト → フェンシング → グループ → モニタ

追加 削除

サーバの定義一覧

順位	名前
マスタサーバ	node1
1	node2

↑ ↓

サーバグループの設定 設定

① 「追加」ボタンを押して、クラスタを構成するサーバを追加します。
サーバの優先順位は「↑」、「↓」ボタンで変更します。
サーバグループを使用する場合は「設定」ボタンでサーバグループを設定します。

戻る 次へ キャンセル

- 5) 使用するインタコネクとディスクハートビート用デバイスを入力し、「次へ」をクリックします。

※画像は、パブリック用のIPアドレスを優先度2に登録することで、インタコネクを2つ用意しています。

クラスタ生成ウィザード

クラスタ → 基本設定 → インタコネク → フェンシング → グループ → モニタ

プロパティ 追加 削除

インタコネク一覧

優先度	種別	MDC	node1	node2
1	カーネルモード	使用しない	192.168.0.1	192.168.0.2
2	カーネルモード	使用しない	10.0.0.1	10.0.0.2
3	DISK	使用しない	/dev/cp-diska1	/dev/cp-diska1

↑ ↓

① クラスタを構成するサーバ間のインタコネクを設定します。
「追加」ボタンでインタコネクを追加し、種別を選択します。
「カーネルモード」、「ユーザモード」、「DISK」、「Witness」は、ハートビートに使用する経路を設定します。「ミラー通信専用」はデータミラーリング通信専用使用する経路を設定します。
「カーネルモード」または「ユーザモード」は一つ以上設定する必要があります。二つ以上設定することを推奨します。
「カーネルモード」、「ユーザモード」、「DISK」の場合は各サーバ列のセルをクリックしてIPアドレスまたはデバイスを設定します。
「Witness」の場合は各サーバ列のセルをクリックして「使用する」、「使用しない」を設定し、「プロパティ」ボタンで詳細を設定します。
クラスタサーバ間専用通信のLANを優先的に使用するよう、「↑」、「↓」ボタンで優先度を設定します。
「ミラー通信専用」の場合は各サーバ列のセルをクリックしてIPアドレスを設定します。
データミラーリング通信に使用する通信経路は「MDC」列で通信経路に割り当てるミラーディスクコネク名を選択します。

戻る 次へ キャンセル

※ミラーディスク構成の場合は、「データミラーリング通信に使用する経路(MDC)」の設定を行います。

クラスタ生成ウィザード

クラスタ → 基本設定 → インタコネク → フェンシング → グループ → モニタ

プロパティ 追加 削除

インタコネク一覧

優先度	種別	MDC	node1	node2
1	カーネルモード	mdc1	192.168.0.1	192.168.0.2
2	カーネルモード	使用しない	10.0.0.1	10.0.0.2

↑ ↓

① クラスタを構成するサーバ間のインタコネクを設定します。
「追加」ボタンでインタコネクを追加し、種別を選択します。
「カーネルモード」、「ユーザモード」、「DISK」、「Witness」は、ハートビートに使用する経路を設定します。「ミラー通信専用」はデータミラーリング通信専用使用する経路を設定します。
「カーネルモード」または「ユーザモード」は一つ以上設定する必要があります。二つ以上設定することを推奨します。
「カーネルモード」、「ユーザモード」、「DISK」の場合は各サーバ列のセルをクリックしてIPアドレスまたはデバイスを設定します。
「Witness」の場合は各サーバ列のセルをクリックして「使用する」、「使用しない」を設定し、「プロパティ」ボタンで詳細を設定します。
クラスタサーバ間専用通信のLANを優先的に使用するよう、「↑」、「↓」ボタンで優先度を設定します。
「ミラー通信専用」の場合は各サーバ列のセルをクリックしてIPアドレスを設定します。
データミラーリング通信に使用する通信経路は「MDC」列で通信経路に割り当てるミラーディスクコネク名を選択します。

戻る 次へ キャンセル

- 6) 本手順では、ネットワークパーティション(以下NP)解決を設定せず「次へ」をクリックします。

NP解決をしない場合、クラスターサーバー間の全ネットワーク通信路における障害発生時にサーバーの緊急停止等が発生します。また、共有ディスク構成でNP解決をしない場合、ディスク破壊の可能性があります。詳しくは、**CLUSTERPRO X 5.0 for Linux** インストール&設定ガイドをご確認ください。

PING方式によるNP解決リソースを設定する場合、pingコマンドを受信し、応答を返却可能な常時稼動している装置（ping用装置）がインタコネクトと同一セグメント上に必要です。

ping用装置を用意できる場合は、上記画面にて「追加」をクリックし、設定を行ってください。

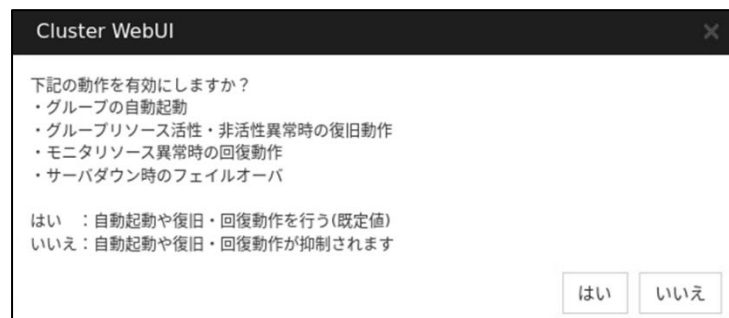
7) フェイルオーバーグループは後で作成するため、ここでは「次へ」をクリックします。



8) モニタリソースは後で作成するため、そのまま「完了」をクリックします。



9) クラスタ動作について尋ねられるので、「はい」をクリックします。



※クラスタ動作の無効化については、下記マニュアルをご参照下さい。

■ CLUSTERPRO X 5.0 システム構築ガイド

(<https://www.support.nec.co.jp/View.aspx?NoClear=on&id=3140108622>)

→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

6.4 2 ノードクラスタ構成情報の作成手順

6.4.4 クラスタ動作を無効化する

以上で、クラスタの作成は終了です。

4.5.2 フェイルオーバーグループの作成

本章では、通常運用時にnode1とデータベースsid1のリソースを持つ、フェイルオーバーグループ「failover1」を作成します。

node1に異常が発生した場合、業務に必要なフェイルオーバーグループに登録されたリソースをnode2に切り替えることにより、フェイルオーバーを実現します。

※双方向スタンバイ環境の場合は、フェイルオーバーグループをsid1とsid2の各インスタンスに作成する必要があります。通常運用時に、node2とsid2のリソースを持つフェイルオーバーグループの作成(failover2)は、本手順を参考の上作成して下さい。

- 1) 「グループ」から、「グループの追加」をクリックします。



- 2) フェイルオーバーグループ名を設定し、「次へ」をクリックします。

A screenshot of a dialog box titled 'グループの定義' (Group Definition) with a close button 'failover X'. The dialog has a breadcrumb trail: '基本設定 → 起動可能サーバ → グループ属性 → グループリソース'. It contains the following fields: 'タイプ*' (Type) with a dropdown menu showing 'フェイルオーバー' (Failover); 'サーバグループ設定を使用する' (Use server group settings) with an unchecked checkbox; '名前*' (Name) with a text input field containing 'failover1'; and 'コメント' (Comment) with an empty text area. A blue information box at the bottom states: '① グループのタイプを選択します。サーバグループを使用する場合、「サーバグループ設定を使用する」チェックボックスをオンにします。' (1. Select the group type. When using server groups, turn on the 'Use server group settings' checkbox). At the bottom right are three buttons: '戻る' (Back), '次へ' (Next), and 'キャンセル' (Cancel).

- 3) 「全てのサーバでフェイルオーバー可能」にチェックが付いていることを確認します。確認したら「次へ」をクリックします。

グループの定義

failover

基本設定 → 起動可能サーバ → グループ属性 → グループリソース

全てのサーバでフェイルオーバー可能 ☒

サーバ

node1

node2

① グループが起動可能なサーバを選択し、サーバの優先順位を設定します。

クラスタに登録されている全てのサーバで起動可能とする場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオンにします。優先順位はクラスタへのサーバ追加時に設定した優先順位となります。

起動するサーバを個別に設定する場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオフにします。右側の「利用可能なサーバ」リストから起動可能なサーバを選択して「追加」ボタンで「起動可能サーバ」リストに追加します。「↑」「↓」ボタンで優先順位を変更します。

戻る

次へ

キャンセル

- ※ 双方向スタンバイ構成でのフェイルオーバーグループ(failover2)を作成する場合は、「全てのサーバでフェイルオーバー可能」のチェックを外し、「起動可能なサーバ」に起動の優先順にnode2、node1を追加、「次へ」をクリックします。

グループの定義

failover

基本設定 → 起動可能サーバ → グループ属性 → グループリソース

全てのサーバでフェイルオーバー可能 ☐

起動可能なサーバ

順位	サーバ
1	node2
2	node1

↑

↓

←
追加

→
削除

利用可能なサーバ

サーバ

利用可能なサーバはありません

① グループが起動可能なサーバを選択し、サーバの優先順位を設定します。

クラスタに登録されている全てのサーバで起動可能とする場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオンにします。優先順位はクラスタへのサーバ追加時に設定した優先順位となります。

起動するサーバを個別に設定する場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオフにします。右側の「利用可能なサーバ」リストから起動可能なサーバを選択して「追加」ボタンで「起動可能サーバ」リストに追加します。「↑」「↓」ボタンで優先順位を変更します。

戻る

次へ

キャンセル

- 4) 要件に合わせて設定し、「次へ」をクリックします。
画像の内容がデフォルトの構成となります。

グループの定義

failover

基本設定 → 起動可能サーバ → **グループ属性** → グループリソース

グループ起動属性

☒ 自動起動 ☐ 手動起動

両系活性チェックを行う

☐

タイムアウト

秒

フェイルオーバー属性

☒ 自動フェイルオーバー

☒ 起動可能なサーバ設定に従う

☐ サーバグループ内のフェイルオーバーポリシーを優先する

☐ スマートフェイルオーバーを行う

☐ ダイナミックフェイルオーバーを行う

☐ サーバグループ内のフェイルオーバーポリシーを優先する

☐ サーバグループ間では手動フェイルオーバーのみを有効とする

☐ 手動フェイルオーバー

フェイルオーバー属性(拡張)

☐ 指定したモニタリソースで異常を検出しているサーバをフェイルオーバー先から除外する☐ 全てのサーバで異常を検出している場合、異常を無視してフェイルオーバーを行う

フェイルバック属性

☐ 自動フェイルバック ☒ 手動フェイルバック

1

フェイルオーバーグループの起動やフェイルオーバーの動作を設定します。
クラスタ起動時にグループを自動起動しない場合は「グループ起動属性」を「手動起動」にしてください。
障害発生時に各サーバのモニタリソースの状態を考慮してフェイルオーバー先を選択する場合は「自動フェイルオーバー」の「ダイナミックフェイルオーバーを行う」を選択してください。サーバグループ設定を使用して、同一サーバグループ内のサーバを優先してフェイルオーバーする場合は、「サーバグループ内のフェイルオーバーポリシーを優先する」を選択してください。

戻る

次へ

キャンセル

フェイルオーバーとフェイルバックの自動/手動の設定は、ここで行って下さい。

- 5) リソースは後で設定するため、そのまま「完了」をクリックします。

グループの定義

failover

基本設定 → 起動可能サーバ → **グループ属性** → **グループリソース**

プロパティ

追加

削除

グループリソース一覧

名前	タイプ
リソースはありません	

1

「追加」ボタンを押して、リソースを追加します。
「プロパティ」ボタンで選択したリソースのプロパティを設定します。

戻る

完了

キャンセル

以上で、フェイルオーバーグループの作成は終了です。

4.5.3 フローティングIPリソースの作成

本章では、Cluster WebUIからfipリソースを作成する手順を紹介します。

クライアントは、fipリソースを使用して、クラスタサーバーに接続することができます。

また、fipリソースを使用することにより、CLUSTERPROがフェイルオーバーまたはフェイルオーバーグループの移動を行っても、クライアントは接続先ノードの切り替えを意識する必要がありません。

fipリソースの作成時には、下記の点に注意して下さい。

- ・fipはクラスタサーバーが所属するLANと、同じネットワークセグメント内で、かつ使用していないアドレスである必要がある
- ・fipリソースの作成は、各フェイルオーバーグループに対して作成する

以下、failover1（データベースsid1）でfipリソースを作成する際の設定例です。

※双方向スタンバイ構成の場合には、同様にfailover2グループのフローティングIPリソース「fip2」を設定してください。

- 1) 対象のフェイルオーバーグループから、「リソースの追加」をクリックします。



- 2) タイプから「フローティングIPリソース」を選択し、fip名リソース名を設定します。
設定したら「次へ」をクリックします。



- 3) デフォルトのまま、「次へ」をクリックします。



- 4) fipリソースの復旧動作を設定し、「次へ」をクリックします。
※ 環境に合わせて復旧動作を設定して下さい。(画像の内容がデフォルトの構成となります)

グループのリソース定義 | failover1 fip ×

情報 ✓ → 依存関係 ✓ → 復旧動作 ✓ → 詳細

活性異常検出時の復旧動作

活性リトライしきい値* 回

フェイルオーバーしきい値* 回

最終動作* 何もしない(次のリソースを活性しない)

☐ 最終動作前にスクリプトを実行する 設定

非活性異常検出時の復旧動作

非活性リトライしきい値* 回

最終動作* クラスタサービス停止とOSシャットダウン

☐ 最終動作前にスクリプトを実行する 設定

戻る 次へ キャンセル

- 5) 使用するfipのアドレスを入力します。入力が完了したら「完了」をクリックし、fipリソースの設定は終了です。

グループのリソース定義 | failover1 fip ×

情報 ✓ → 依存関係 ✓ → 復旧動作 ✓ → 詳細

共通 node1 node2

IPアドレス*

調整

戻る 完了 キャンセル

4.5.4 ディスクリソースの作成

本章では、Cluster WebUIから共有ディスク構成のディスクリソースを作成する手順を紹介합니다。

ミラーディスク構成の場合は、CLUSTERPROのリファレンスガイドを確認の上、ミラーディスクリソースを設定して下さい。

章番号『4.1 前提環境』のとおり、下記的前提でディスクリソースを設定します。

【sid1の前提環境】

切替パーティション	/dev/cp-diska2 (/dev/sdb2 のシンボリックリンク)
マウントポイント	/oradata/sid1

※ミラーディスク構成の場合は、下記的前提でミラーディスクリソースを設定します。

【sid1の前提環境】

データパーティション	/dev/cp-diska2 (/dev/sdb2 のシンボリックリンク)
クラスタパーティション	/dev/cp-diska1 (/dev/sdb1 のシンボリックリンク)
マウントポイント	/oradata/sid1

※双方向ディスク構成・共有ディスク構成の場合は、下記的前提でディスクリソースを設定します。

【sid2の前提環境】

切替パーティション	/dev/cp-diskb1 (/dev/sdc1 のシンボリックリンク)
マウントポイント	/oradata/sid2

※双方向ディスク構成・ミラーディスク構成の場合は、下記的前提でミラーディスクリソースを設定します。

【sid2の前提環境】

データパーティション	/dev/cp-diskb2 (/dev/sdc2 のシンボリックリンク)
クラスタパーティション	/dev/cp-diskb1 (/dev/sdc1 のシンボリックリンク)
マウントポイント	/oradata/sid2

ディスクリソースは、フェイルオーバーグループ毎に、各データベースに対応したパーティションやマウントポイントを設定して作成する必要があります。

切替パーティションとは、クラスタを構成する複数台のノードに接続された、共有ディスク上のパーティションを指します。切替パーティションを使用するノードの切替は、フェイルオーバーグループ毎に、フェイルオーバーポリシーに従って行われます。業務に必要なデータは、切替パーティション上に格納しておくことで、フェイルオーバーまたはフェイルオーバーグループの移動時等に自動的に引き継がれます。

以下、failover1（データベースsid1）でディスクリソースを作成する際の設定例です。

※ 双方向スタンバイ構成の場合には、同様にfailover2グループでディスクリソース「disk2」を設定してください。

- 1) 設定モードでフェイルオーバーグループから、「リソースの追加」をクリックします。



- 2) タイプから「ディスクリソース」を選択し、ディスクリソース名を設定します。
設定が完了したら「次へ」をクリックします。

グループのリソース定義 | failover1 disk ✕

情報 → 依存関係 → 復旧動作 → 詳細

タイプ*

名前*

コメント

! グループリソースの種類を選択して名前を入力してください。

※ミラーディスク構成の場合は、ここで「ミラーディスクリソース」を選択し、同様の設定を行います。設定項目が異なる部分があるため、適宜読み替えて設定して下さい。

グループのリソース定義 | failover1 md ✕

情報 → 依存関係 → 復旧動作 → 詳細

タイプ*

名前*

コメント

! グループリソースの種類を選択して名前を入力してください。

- 3) デフォルトのまま、「次へ」をクリックします。
※ミラーディスク構成の場合は、デフォルトの内容が異なりますが、
そのまま「次へ」をクリックしてください。

グループのリソース定義 | failover1 disk ✕

情報 ✔ → 依存関係 → 復旧動作 → 詳細

既定の依存関係に従う ☒

依存するリソース

- AWS DNSリソース
- AWS Elastic IPリソース
- AWSセカンダリIPリソース
- AWS仮想IPリソース
- Azure DNSリソース
- Azureプローブポートリソース
- ダイナミックDNSリソース
- フローティングIPリソース
- ボリュームマネージャリソース
- 仮想IPリソース

◀ 戻る 次へ ▶ キャンセル

- 4) ディスクリソースの復旧動作を設定し、「次へ」をクリックします。
※ 環境に合わせて復旧動作を設定して下さい。(画像の内容がデフォルトの構成となります)

グループのリソース定義 | failover1 disk ✕

情報 ✔ → 依存関係 ✔ → 復旧動作 → 詳細

活性異常検出時の復旧動作

活性リトライしきい値* 0 回

フェイルオーバーしきい値* 1 回

最終動作* 何もしない(次のリソースを活性しない) ▼

☐ 最終動作前にスクリプトを実行する 設定

非活性異常検出時の復旧動作

非活性リトライしきい値* 0 回

最終動作* クラスタサービス停止とOSシャットダウン ▼

☐ 最終動作前にスクリプトを実行する 設定

◀ 戻る 次へ ▶ キャンセル

- 5) ファイル・システムを設定、デバイス名、マウントポイントを入力し、「完了」をクリックし、ディスクリソースの設定は終了です。

グループのリソース定義 | failover1 disk ✕

情報 ✓ → 依存関係 ✓ → 復旧動作 ✓ → 詳細

共通 node1 node2

ディスクタイプ* disk

ファイルシステム* ext4

デバイス名* /dev/cp-diska2

RAWデバイス名

マウントポイント* /oradata/sid1

調整

◀ 戻る 完了 キャンセル

- ※ ミラーディスク構成の場合は、以下の画面でミラーパーティションデバイス名、マウントポイント、データパーティションデバイス名、クラスタパーティションデバイス名、ファイルシステムを設定します。設定後、ミラーディスクコネクットの「選択」をクリックします。

グループのリソース定義 | failover1 md ✕

情報 ✓ → 依存関係 ✓ → 復旧動作 ✓ → 詳細

共通 node1 node2

ミラーパーティションデバイス名* /dev/NMP1

マウントポイント* /oradata/sid1

データパーティションデバイス名* /dev/cp-diska2

クラスタパーティションデバイス名* /dev/cp-diska1

ファイルシステム* ext4

ミラーディスクコネクット 選択

調整

◀ 戻る 完了 キャンセル

「ミラーディスクコネクットの選択」画面で正しいミラーディスクコネクットが選択されていることを確認します。
 確認後に「OK」をクリックして元の画面に戻ります。
 元の画面で「完了」を選択します。

ミラーディスクコネクットの選択

ミラーディスクコネクット一覧		利用可能なミラーディスクコネクット	
順位	MDC		MDC
1	mdc1		利用可能なミラーディスクコネクットはありません

← 追加
→ 削除

↑ ↓

OK キャンセル 適用

4.5.5 設定項目の反映

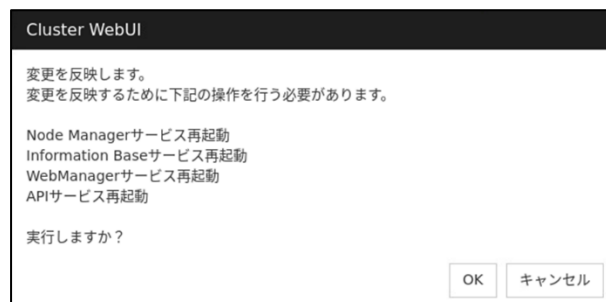
クラスタ、フェイルオーバーグループ、ディスクリソース、フローティングIPリソースの設定が完了したら、設定項目をCLUSTERPROへ反映します。

※設定反映時は、クラスタサービスが停止またはサスペンドされていることを確認してから行うようにしてください。

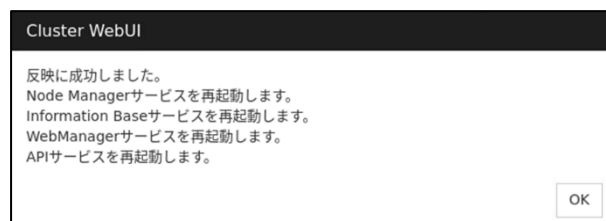
- 1) 「設定モード」画面の上部メニューから「設定の反映」をクリックします。



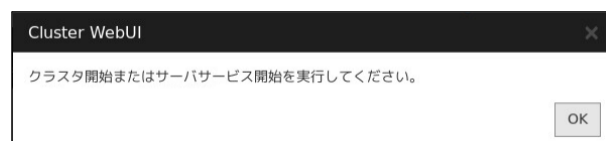
- 2) 「OK」をクリックして、各設定をサーバーに反映します。



- 3) 「OK」をクリックして、設定の反映は完了です。

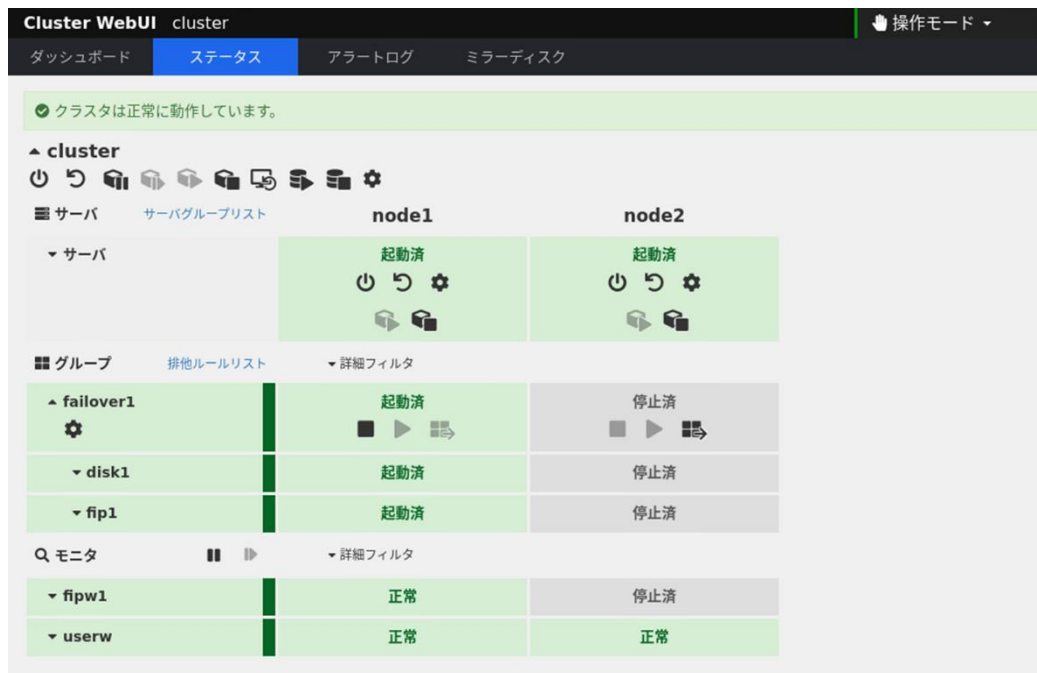


- 4) 以下のダイアログが表示された場合は、「OK」をクリックします。



以上で、failover1の反映は完了です。

設定が正常に反映されていることをCluster WebUI上で確認できます。

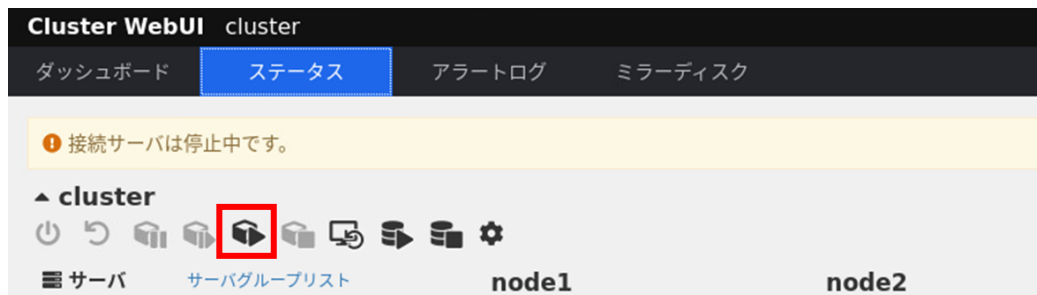


クラスタが起動していない場合は、下記手順で起動することができます。

- 1) 「操作モード」へ切り替えます。



- 2) 「クラスタ開始」を選択します。



- 3) 「開始」をクリックします。クラスタが正常に開始されることを確認してください。



4.6 Oracle Database ソフトウェアのインストール

ノード1とノード2のローカルパーティションにOracle Databaseをインストールします。ここではソフトウェアのみをインストールします。

基本的なインストールの手順は、シングルノードの構築手順と同様です。

なお、RU(Release Update)およびRUR(Release Update Revision)はできる限り最新のものを使用するようにお願いします。

oracleユーザーでログインします。

端末からORACLE_HOMEのディレクトリを作成します。

```
$ mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
```

※Oracle Databaseインストレーション・ガイド等との整合性を保つため、

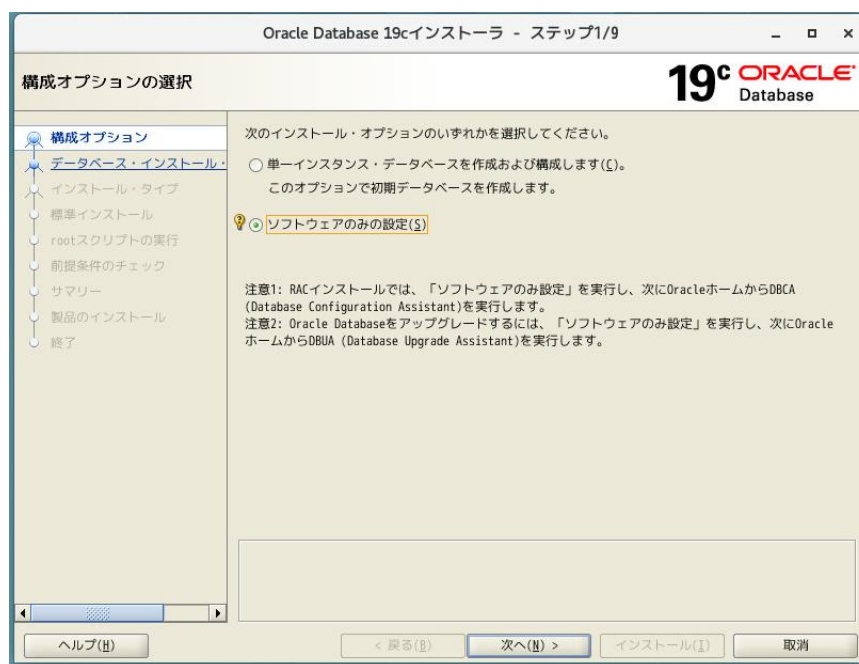
網掛け部分はインストールするOracle Databaseのバージョン名に合わせることを推奨します。

インストール用zipファイルを作成したディレクトリに解凍します。

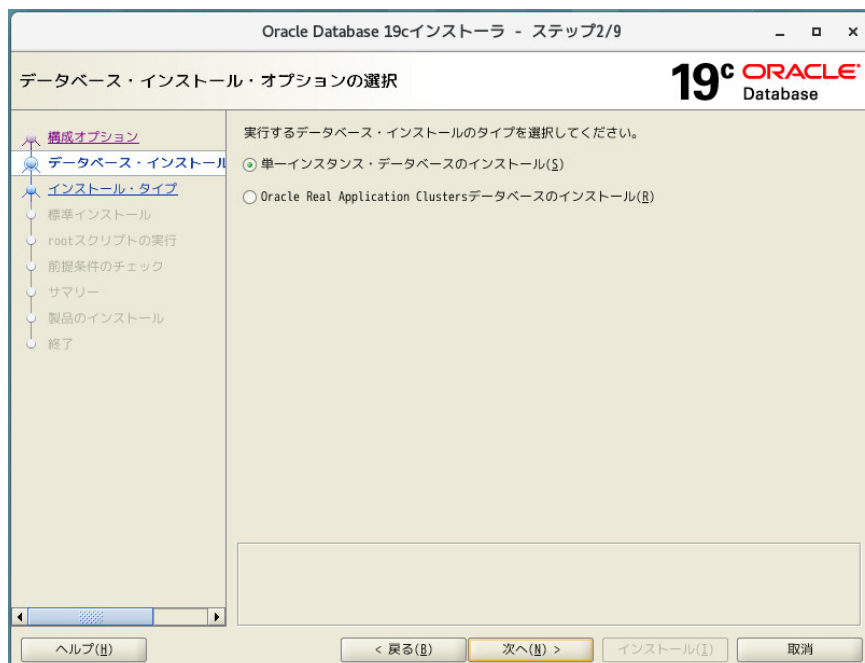
解凍後にディレクトリにある「runInstaller」を実行し、Oracle Universal Installerを起動します。

- 1) 「ソフトウェアのみの設定」を選択し、「次へ」をクリックします。

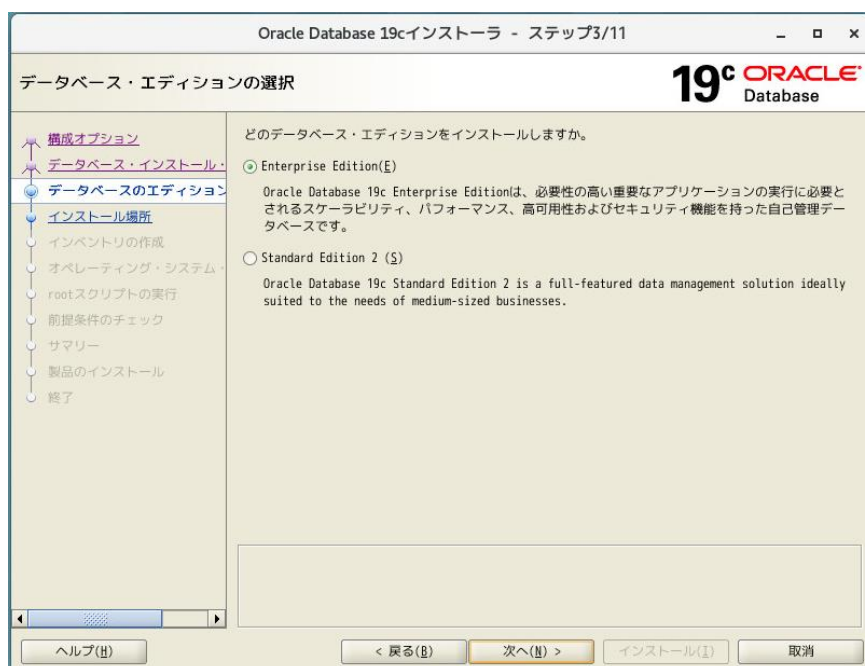
※本編では、データベースの作成はデータベース・ソフトウェアのインストール後に行います。



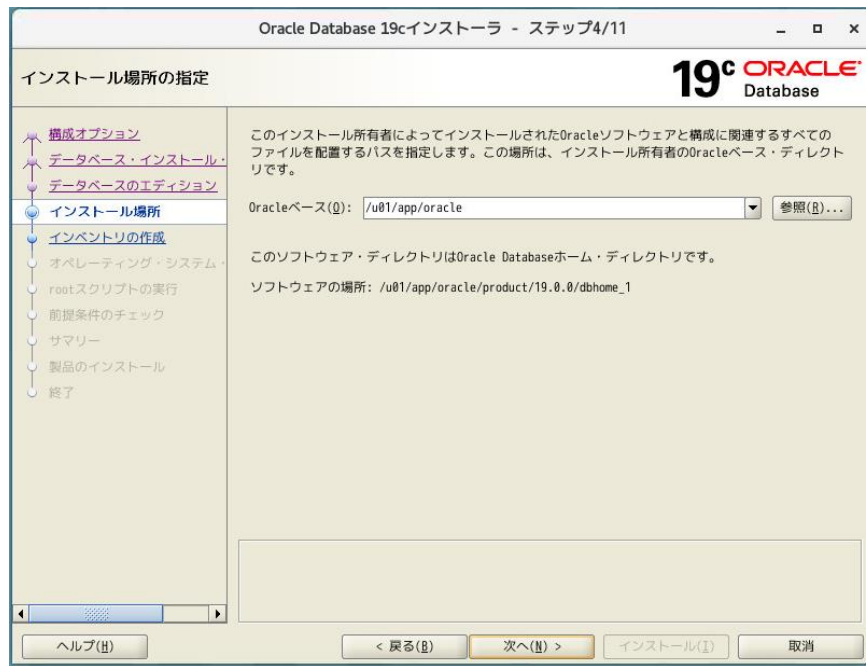
- 2) 「単一インスタンス・データベースのインストール」を選択し、「次へ」をクリックします。



- 3) データベース・エディションを選択します。本手順では、「Enterprise Edition」を対象とします。選択したら「次へ」をクリックします。



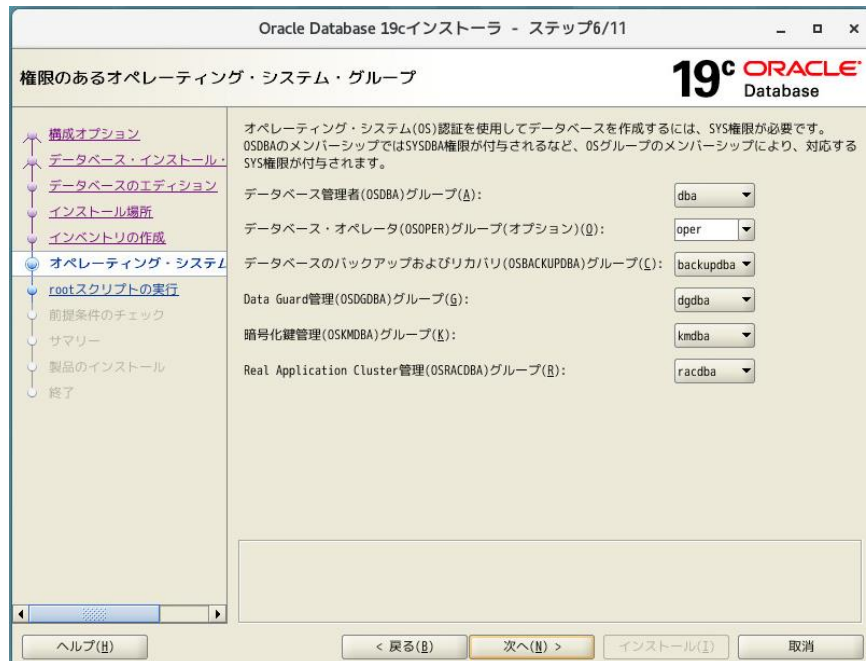
- 4) Oracle Databaseをインストールする場所を確認します。Oracleベースと、ソフトウェアの場所を確認し、「次へ」をクリックします。



- 5) インベントリ・ディレクトリの場所を確認します。また、インベントリ・ディレクトリに書き込み権限を持つグループを選択します。本構成では、oraInventoryグループ名に「oinstall」を選択し、「次へ」をクリックします。



- 6) データベースに対するOS認証に使用するグループを設定します。本構成では、データベース管理者(OSDBA)グループに「dba」、データベース・オペレータ(OSOPER)グループに「oper」、データベースのバックアップおよびリカバリ(OSBACKUPDBA)グループに「backupdba」、DataGuard管理(OSDGDBA)グループに「dgdba」、暗号化鍵管理(OSKMDBA)グループに「kmdba」、Real Application Cluster管理(OSRACDBA)グループに「racdba」を選択します。選択したら「次へ」をクリックします。



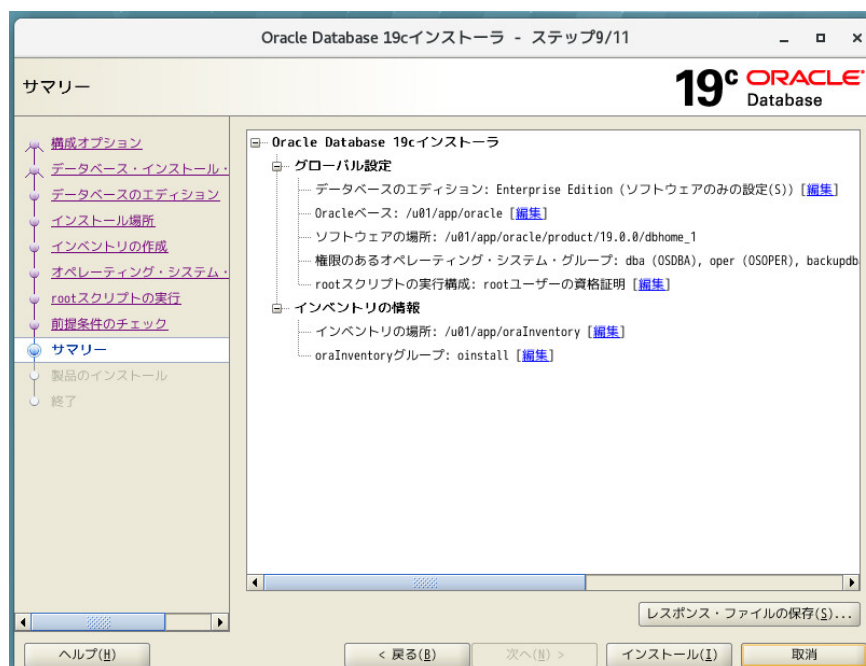
- 7) インストール中にrootユーザーで実行が必要な操作をインストーラで実行できます。「構成スクリプトを自動的に実行」をチェックして、rootユーザーのパスワードを入力します。入力したら「次へ」をクリックします。



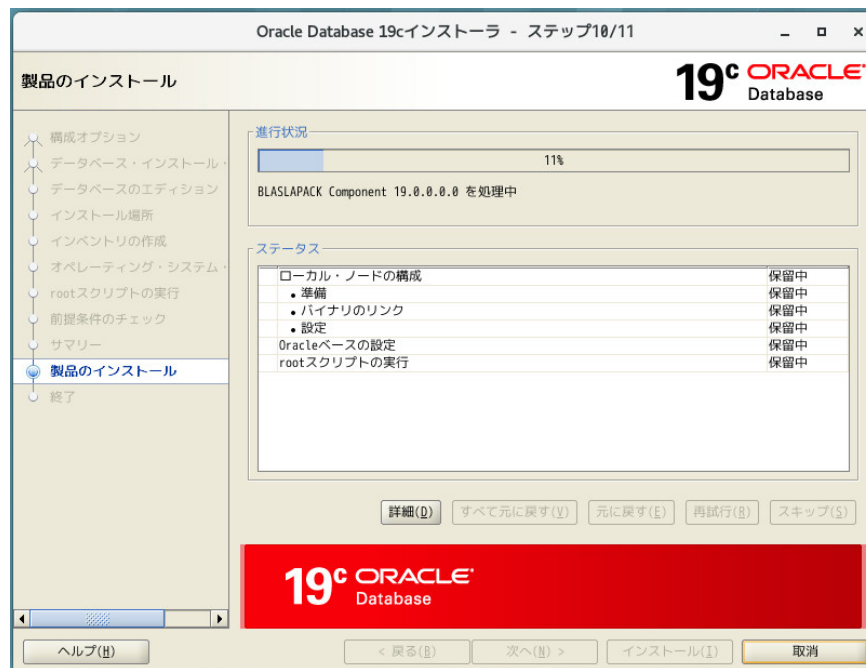
- 8) インストール実行前の前提条件がチェックされます。



- 9) サマリーを確認後、「インストール」をクリックすると、インストールが開始されます。



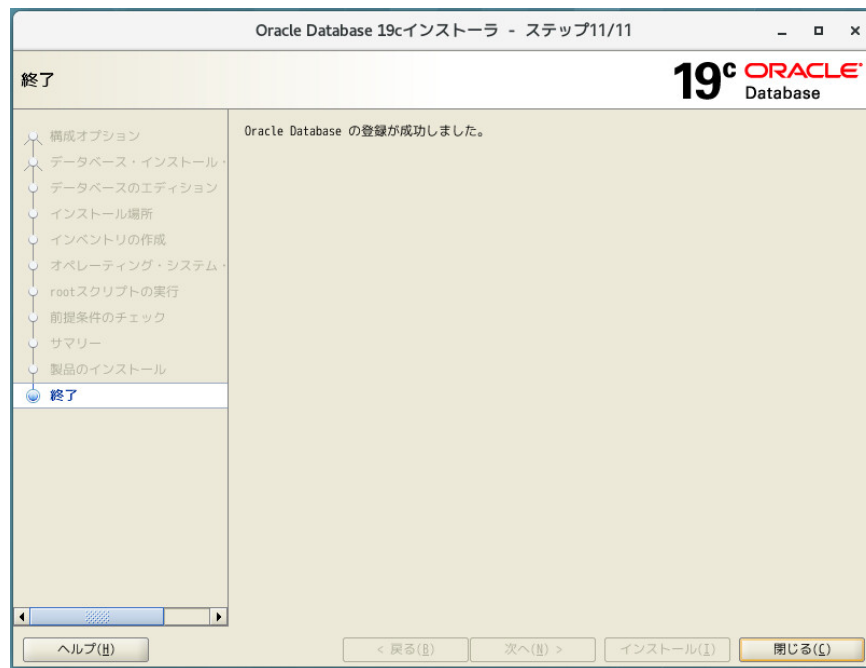
10) インストールが開始されます。



11) インストール中に、下記ダイアログが表示されます。ダイアログの指示に従い、「はい」を選択します。



12) Oracle Databaseのインストールが完了します。



4.7 データベースの作成

4.7.1 環境変数の設定

各ノードのoracleユーザーのホーム・ディレクトリにある「.bash_profile」に、Oracle Databaseに必要な環境変数を追加します。

片方向スタンバイ構成と双方向スタンバイ構成で\$ORACLE_SIDの設定が異なります。以下の設定例を参考にしてください。網掛け部分は、実際のsid名に変更して下さい。

片方向スタンバイの設定例

(ノード1、ノード2に設定)

```
$ vi $HOME/.bash_profile
(下記行がない場合は最終行以降に追加します)
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_LANG=Japanese_Japan.UTF8
export ORACLE_SID=sid1
```

※双方向スタンバイの場合の環境変数の設定例は、
『6.1.5 双方向スタンバイ構成の場合の環境変数設定例』を確認してください。

上記の環境変数を反映するため、以下のコマンドを使用します。

```
$ source ~/.bash_profile
```

4.7.2 DBCAの実行

ノード1でデータベースsid1を作成します。
双方向スタンバイ構成の場合は追加で、ノード2でデータベースsid2を作成します。

本手順では、初期化パラメーターファイル、サーバーパラメーターファイル、制御ファイル、データファイル、REDOログファイル、アーカイブログファイルは切替パーティション、またはデータパーティション上に作成します。

なお、データベースはDatabase Configuration Assistant(以下DBCA)を用いて作成します。

各ノードでdfコマンドを実行し、共有ディスク(/dev/sdb2)のマウント位置が、章番号『4.5.4 ディスクリソースの作成』で設定した位置(本構成では/oradata/sid1)に設定されているか確認してください。

※本手順はrootユーザーで実行します。

df

ファイルシス	1K-ブロック	使用	使用可	使用%	マウント位置
devtmpfs	3969588	0	3969588	0%	/dev
tmpfs	3998636	15968	3982668	1%	/dev/shm
tmpfs	3998636	9836	3988800	1%	/run
tmpfs	3998636	0	3998636	0%	/sys/fs/cgroup
/dev/mapper/rhel-root	52403200	22478752	29924448	43%	/
/dev/sda1	1038336	234064	804272	23%	/boot
/dev/mapper/rhel-home	43060692	7562000	35498692	18%	/home
tmpfs	799724	1180	798544	1%	/run/user/42
tmpfs	799724	8244	791480	2%	/run/user/1100
/dev/sr0	8238560	8238560	0	100%	/run/media/oracle/RHEL----
/dev/sdb2	51322336	53272	48632316	1%	/oradata/sid1
tmpfs	799724	4	799720	1%	/run/user/0

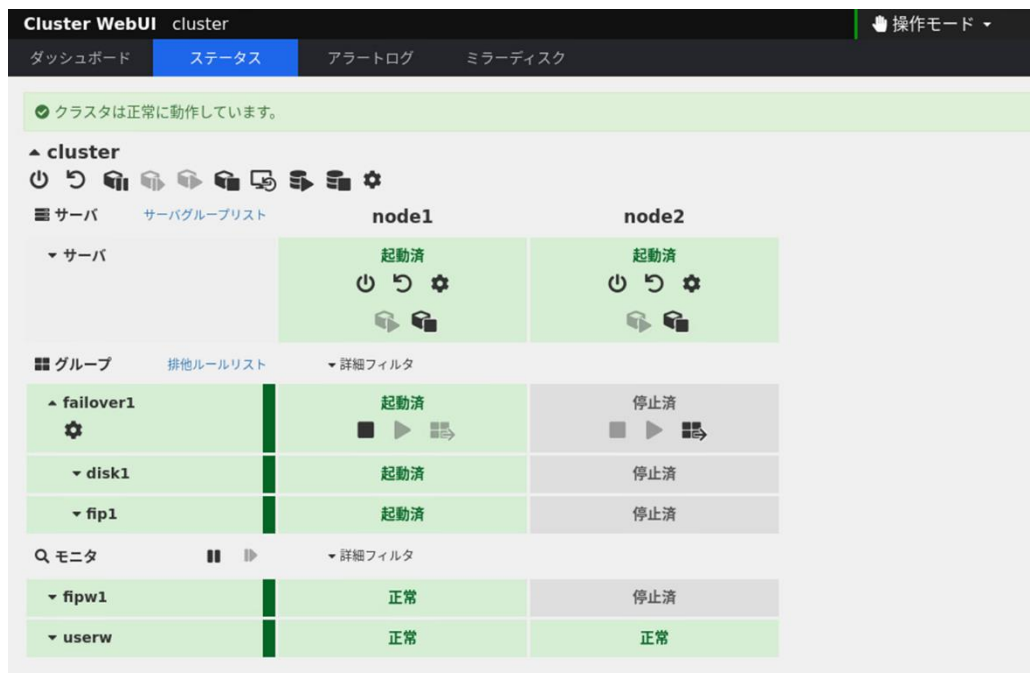
※ノード1から共有ディスクをマウントしている状態で、ノード2からも同じ共有ディスクをマウントしないようご注意ください。

以下は、ノード1からデータベースsid1を作成する際の例です。
双方向スタンバイ構成の場合、ノード1の作成後に同様の手順でノード2からデータベースsid2を作成します。

■ データベースの作成例

実際には、使用するEditionやオプション製品に合わせて、データベースを作成して下さい。
 なお、データベース・ファイル、フラッシュ・リカバリ領域が切替パーティション、またはデータパーティション上に作成されるように設定します。

- 1) Cluster WebUI上にて、フェイルオーバーグループがデータベース作成を行うノードで起動されていることを確認します。



- 2) DBCAを起動します。

※本手順は4. 7. 1の環境変数が適用されたoracleユーザーで実行します。
 \$ dbca

- 3) 「データベースの作成」を選択し、「次へ」をクリックします。



- 4) 作成モードを選択します。本構成では、「拡張構成」を選択したことを前提に行います。選択したら「次へ」をクリックします。「拡張構成」以外を選択した場合、画面の内容等が異なりますのでご注意ください。

- 5) 作成するデータベース・タイプに「Oracle単一インスタンス・データベース」を選択します。データベースのテンプレートを選択し、「次へ」をクリックします。本構成では、「カスタム・データベース」を選択したことを前提に行います。

テンプレート名	データファイル...	詳細
☒ データ・ウェアハウス	はい	詳細の表示
☐ カスタム・データベース	いいえ	詳細の表示
☐ 汎用またはトランザクション処理	はい	詳細の表示

- 6) グローバル・データベース名とSIDを入力し、「次へ」をクリックします。

【CDB(シングルテナント)構成のデータベースを作成する場合】

「コンテナ・データベースとして作成」にチェックを入れ、環境に合わせて設定してください。

The screenshot shows the 'Database Configuration Assistant - データベースの作成(C) - ステップ4/14' window. The 'データベースIDの詳細の指定' (Specify Database ID Details) step is active. The left sidebar shows the navigation tree with 'データベースの識別' (Identify Database) selected. The main area contains the following fields and options:

- 一意のデータベース識別子情報を入力します。Oracleデータベースは、一般的に"name.domain"という形式のグローバル・データベース名で一意に識別されます。
- グローバル・データベース名(G):
- SID(S):
- サービス名(E):
- ☒ コンテナ・データベースとして作成(L)
- 単一のデータベースに複数のデータベースを統合するためにコンテナ・データベースを使用でき、データベースの仮想化を有効にします。コンテナ・データベース(CDB)には、1つ以上のプラガブル・データベース(PDB)を含むことができます。
- ☒ PDB用のローカルUNDO表領域の使用(L)
- ☐ 空のコンテナ・データベースの作成(B)
- ☒ 1つ以上のPDBを含むコンテナ・データベースの作成(A)
- PDBの数(U):
- PDB名(P):

Buttons at the bottom: ヘルプ(H), < 戻る(B), 次へ(N) >, 終了(E), 取消.

【非CDB構成のデータベースを作成する場合】

「コンテナ・データベースとして作成」のチェックを外します。

The screenshot shows the same 'Database Configuration Assistant' window, but with the 'コンテナ・データベースとして作成(L)' checkbox unchecked. The other fields remain the same:

- 一意のデータベース識別子情報を入力します。Oracleデータベースは、一般的に"name.domain"という形式のグローバル・データベース名で一意に識別されます。
- グローバル・データベース名(G):
- SID(S):
- サービス名(E):
- ☐ コンテナ・データベースとして作成(L)
- 単一のデータベースに複数のデータベースを統合するためにコンテナ・データベースを使用でき、データベースの仮想化を有効にします。コンテナ・データベース(CDB)には、1つ以上のプラガブル・データベース(PDB)を含むことができます。
- ☒ PDB用のローカルUNDO表領域の使用(L)
- ☐ 空のコンテナ・データベースの作成(B)
- ☒ 1つ以上のPDBを含むコンテナ・データベースの作成(A)
- PDBの数(U):
- PDB名(P):

Buttons at the bottom: ヘルプ(H), < 戻る(B), 次へ(N) >, 終了(E), 取消.

- 7) 「データベース記憶域属性に次を使用」を選択し、「データベース・ファイルの位置」に切替パーティション、またはデータパーティションの領域を入力します。入力したら「次へ」をクリックします。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ5/14

データベース記憶域オプションの選択

19c ORACLE Database

データベース記憶域属性にテンプレート・ファイルを使用(I)

データベース・ファイルの記憶域のタイプおよび場所が指定されたテンプレート(カスタム・データベース)から選択されます。

データベース記憶域属性に次を使用(E)

すべてのデータベース・ファイルは次の指定された場所にあります。後続の画面で各データファイルの名前および場所をカスタマイズできます。

データベース・ファイルの記憶域タイプ(D): ファイルシステム

データベース・ファイルの位置(L): /oradata/sid1 参照(R)...

Oracle Managed Filesオプションによって、データベース管理を簡略化するためにデータファイルの名前が自動的に生成されます。

☐ Oracle Managed Filesの使用(OMF)(O) REDOログおよび制御ファイルの多重化(M)...

ファイルの位置変数(V)...

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

- 8) 「高速リカバリ領域の指定」と「アーカイブ有効化」をチェックし、「高速リカバリ領域」に切替パーティション、またはデータパーティション上の領域を入力します。入力したら「次へ」をクリックします。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ6/14

高速リカバリ・オプションの選択

19c ORACLE Database

データベースのリカバリ・オプションを選択します。

☒ 高速リカバリ領域の指定(E)

リカバリ・ファイルの記憶域タイプ(S): ファイルシステム

高速リカバリ領域(R): /oradata/sid1/fast_recovery_area 参照(R)...

高速リカバリ領域のサイズ(L): 17271 MB

☒ アーカイブ有効化(E) アーカイブ・モード・パラメータの編集(A)...

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

- 9) リスナーの構成は後で行うため、そのまま「次へ」をクリックします。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ7/14

ネットワーク構成詳細の指定

19c ORACLE Database

リスナーの選択(L)

現在のOracleホームのリスナーを次に示します。現在のOracleホームに新規リスナーを作成するには、リスナーの名前とポートを指定します。

名前	ポート	Oracleホーム	ステータス
----	-----	-----------	-------

☐ 新規リスナーの作成(L)

リスナー名(N):

リスナー・ポート(P): 1521

Oracleホーム: /u01/app/oracle/product/19.0.0/dbhome_1

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

- 10) データベースのコンポーネントを選択します。18c以前のバージョンからのアップグレードやPDBの移行などを考慮して、本構成ではすべてにチェックしますが、必要な機能、ライセンスに応じて適宜選択してください。

【CDB(シングルテナント)構成のデータベースを作成する場合】

本構成では、CDBに対してすべてのコンポーネントがインストールされます。

必要な機能、ライセンスに応じてPDBで利用するコンポーネントに対しても[PDBに含める]にチェックしてください。設定したら「次へ」をクリックします。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ8/15

データベース・オプションの選択

19c ORACLE Database

データベース・コンポーネント(L)

データベースでの使用のために構成する標準データベース・コンポーネントを選択してください。常にこれらのコンポーネントをデータベースにインストールすることをお勧めします。これらのコンポーネントの選択を解除すると、次ページの一部のコンポーネントを選択できなくなる場合があります。

コンポーネントの選択	表領域	PDBに含める
☒ Oracle JVM(J)	SYSTEM	☐
☒ Oracle Text(T)	SYSAUX	☐
☒ Oracle Multimedia(M)	SYSAUX	☐
☒ Oracle OLAP(O)	SYSAUX	☐
☒ Oracle Spatial(S)	SYSAUX	☐
☒ Oracle Label Security(L)	SYSTEM	☐
☒ Oracle Application Express(A)	SYSAUX	☐
☒ Oracle Database Vault(V)	SYSAUX	☐

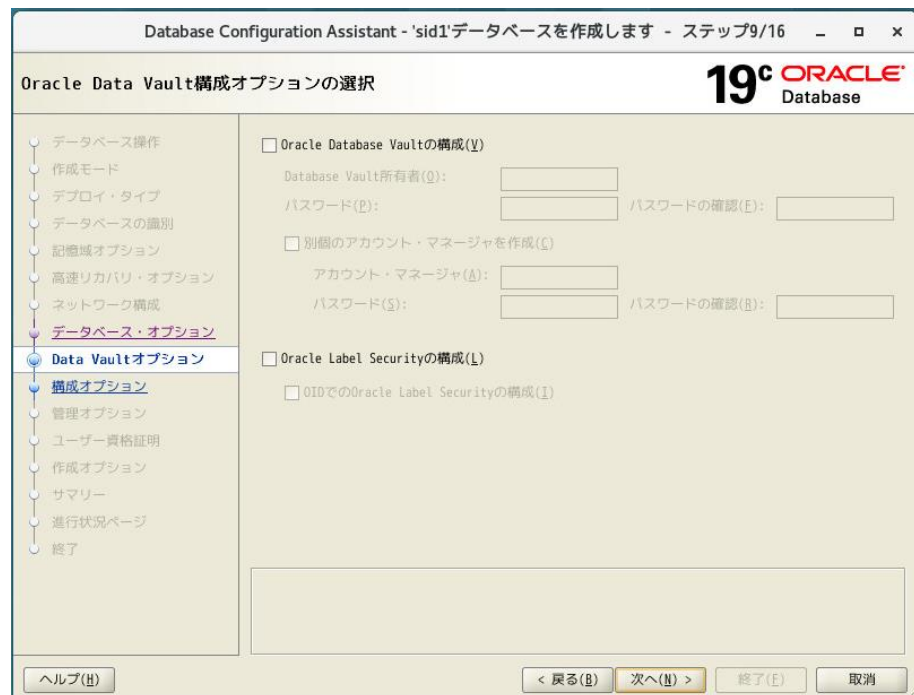
ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

【非CDB構成のデータベースを作成する場合】

本構成では、すべてのコンポーネントがインストールされます。設定したら「次へ」をクリックします。



- 11) Oracle Data Vaultおよび Oracle Label Securityを構成する場合は、アカウント情報などの情報を入力します。本構成では構成しません。設定したら「次へ」をクリックします。



- 12) 構成オプションを指定します。環境や使用目的に応じてそれぞれの項目を適宜設定してください。各項目を設定したら、「次へ」をクリックします。

[メモリー]

メモリーの管理方法と、割り当てるメモリー・サイズを設定します。

Oracle Databaseで使用するメモリーの合計がOSで利用できる物理メモリー内に収まるように設定してください。

[サイズ指定]

基本ブロック・サイズはデータベース作成後には変更できません。変更する場合は、データベースの再作成が必要ですので注意してください。

[キャラクタ・セット]

データベース・キャラクタ・セットは、データベース作成後には変更できません。変更する場合は、データベースの再作成が必要ですので注意してください。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ10/16

構成オプションの指定

19c ORACLE Database

キャラクタ・セット(L) 接続モード(Q) サンプル・スキーマ(E)

メモリー(M) サイズ設定(S)

データベース・キャラクタ・セットにより、データベースでの文字データの格納方法が決まります。

☒ Unicode(AL32UTF8)を使用(U)

文字セットをUnicode(AL32UTF8)に設定すると、複数の言語グループを格納できます。

☐ OSの文字セット(JA16EUC)の使用(L)

文字セットは、このオペレーティング・システムの言語設定に基づいています。

☐ 次の文字セットから選択(B)

データベース文字セット(A): AL32UTF8 - Unicode UTF-8 汎用キャラクタ・セット

☒ 推奨の文字セットのみ表示(B)

各国語文字セット(I): AL16UTF16 - Unicode UTF-16 汎用キャラクタ・セット

デフォルト言語(L): 日本語

デフォルト地域(I): 日本

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(E) 取消

[接続モード]

構成に応じて、「専用サーバー・モード」か「共有サーバー・モード」を選択してください。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ10/16

構成オプションの指定

19c ORACLE Database

キャラクタ・セット(L) 接続モード(Q) サンプル・スキーマ(E)

メモリー(M) サイズ設定(S)

☒ 専用サーバー・モード(B)

データベースは、このモードで各クライアント接続に対して専用のリソースを割り当てます。クライアント接続の合計数が少ない場合、またはクライアントがデータベースに対して長時間に及ぶ要求を行う場合に、このモードを使用します。

☐ 共有サーバー・モード(B)

データベースは、このモードのすべてのクライアント接続に割り当てられたリソースの共有プールを使用します。システム・リソースを効果的に利用し、かつ多くのユーザーがデータベースに同時接続する必要がある場合には、このモードを使用します。

共有サーバーの数を指定してください。この数は、インスタンス起動時に作成されるプロセスの数となります。

共有サーバー(V): 1

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(E) 取消

[サンプル・スキーマ]

必要に応じて、チェックしてください。

各パラメータータブの設定が完了したら、「次へ」をクリックします。

- 13) 「Enterprise Manager (EM) Database Expressの構成」のチェックを外し、「次へ」をクリックします。本手順例では、Enterprise Managerを使用致しません。Enterprise Managerの詳細は章番号『3.4 その他考慮事項』をご確認下さい。

- 14) データベースユーザーのパスワードの設定をし、「次へ」をクリックします。
本構成では、すべてのアカウントに同一のパスワードを設定していますが、セキュリティを考慮した場合、個別のパスワードを設定することを推奨します。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ11/15

データベース・ユーザー資格証明の指定

セキュリティの理由により、新規データベースの次のユーザー・アカウントのパスワードを指定する必要があります。

☐ 別の管理パスワードを使用(U)

パスワード パスワードの確認

SYS(S)

SYSTEM(Y)

PDADMIN

☒ すべてのアカウントに同じ管理パスワードを使用(Y)

パスワード(P): パスワードの確認(C):

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

- 15) 「データベースの作成」にチェックがされていることを確認します。

Database Configuration Assistant - 'sid1'データベースを作成します - ステップ13/16

データベース作成オプションの選択

データベース作成オプションを選択します。

☒ データベースの作成(L)

データベース作成後に実行するSQLスクリプトを指定してください。スクリプトは次にリストされる順序で実行されます。

後処理データベース作成スクリプト(E): 参照(R)...

☐ データベース・テンプレートとして保存(I)

テンプレート名(A): dbca_template_2020-06-09_05-

テンプレートの場所(L): /u01/app/oracle/product/19.0.0/dbhome_1/assistants/d 参照(R)...

説明(S):

☐ データベース作成スクリプトの生成(G)

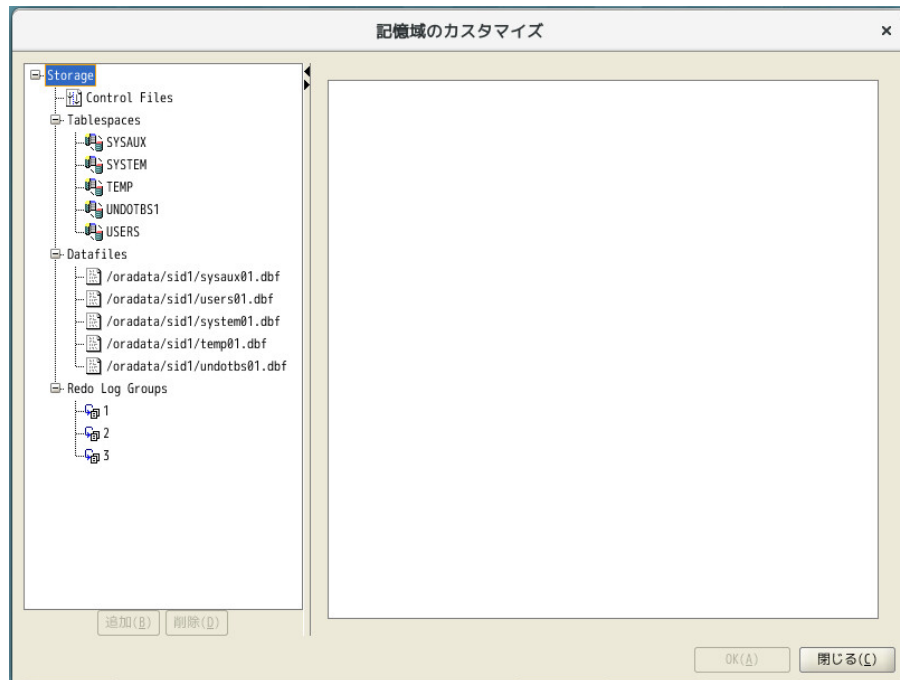
宛先ディレクトリ(D): {ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/scripts 参照(R)...

次の拡張構成オプションを使用して、初期化パラメータを構成し、データベース記憶域の場所をカスタマイズできます。

すべての初期化パラメータ(E)... 記憶域の場所のカスタマイズ(U)...

ヘルプ(H) < 戻る(B) 次へ(N) > 終了(F) 取消

「記憶域の場所のカスタマイズ」をクリックします。



データベース記憶域の構成に問題がなければ、「閉じる」をクリックして前の画面に戻り、「次へ」をクリックします。

16) サマリーの確認を行います。



問題がなければ、「終了」をクリックします。

17) データベースの作成が開始されます。



18) データベースの作成が完了します。



「閉じる」をクリックして、データベースの作成は終了です。

4.8 データベース作成後の作業

4.8.1 データベース作成後の注意点

Database 12cR2以降の環境では、以下の問題が発生する場合があります。

・アラートログにエラー番号「ORA-12012」が出力される

```
ORA-12012: error on auto execute of job "SYS"."ORA$AT_OS_OPT_SY_88"  
ORA-20001: Statistics Advisor: Invalid Task Name For the current user <== Note it  
ORA-06512: at "SYS.DBMS_STATS", line 46861  
ORA-06512: at "SYS.DBMS_STATS_ADVISOR", line 734  
ORA-06512: at "SYS.DBMS_STATS_INTERNAL", line 19930  
ORA-06512: at "SYS.DBMS_STATS_INTERNAL", line 21934  
ORA-06512: at "SYS.DBMS_STATS", line 46851
```

なお、アラートログは以下のPATHに保存されます⁹。

```
$ORACLE_BASE/diag/rdbms/<DB_NAME>/<INSTANCE_NAME>/trace/alert_<SID>.log
```

本現象を確認した場合は、下記対策(タスクの再作成)を実施して下さい。

```
# sqlplus / as sysdba  
SQL> EXEC dbms_stats.init_package();
```

※本現象は、シードデータベースからパッケージdbms_stats.init_packageが正しく実行されずに作成された可能性があります。詳細は該当するドキュメント¹⁰をご確認下さい。

4.8.2 ノード間でディレクトリをコピー

データベースの作成によって作成されたファイルを、データベースを作成していない方のノード(ノード2)にコピーします。

下記コマンドは、ノード1からファイルをコピーした場合の例です。網掛け部分は、実際の環境のsid名やホスト名に変更して下さい。以下の手順は、oracleユーザーで実施します。

【ノード2で実行】

```
$ mkdir $ORACLE_BASE/admin
```

【ノード1で実行】※ノード2で上記のディレクトリを作成してから実行してください。

```
$ scp $ORACLE_HOME/dbs/hc_sid1.dat node2:$ORACLE_HOME/dbs  
$ scp $ORACLE_HOME/dbs/lkSID1 node2:$ORACLE_HOME/dbs  
$ scp $ORACLE_HOME/dbs/orapwsid1 node2:$ORACLE_HOME/dbs  
$ scp -r $ORACLE_BASE/admin/sid1 node2:$ORACLE_BASE/admin/  
$ scp -r $ORACLE_BASE/diag/rdbms/sid1 node2:$ORACLE_BASE/diag/rdbms/
```

※双方向スタンバイ構成の場合、同様にノード2からノード1へコピーする必要があります。

⁹ DocID 1745257.1 「[ADR] 自動診断レポジトリと rdbms, listener, clients 配下のファイルについて (KROWN:133097)」

¹⁰ DocID 2258809.1 「12.2 Database 以降 (例えば18c)で"SYS"."ORA\$AT_OS_OPT_SY_<NN>ジョブを自動実行するとORA-12012が発生する」

4.8.3 /dev/shm のマウント・サイズの変更

Linuxシステムで、オペレーティング・システムの/dev/shmのマウント・サイズが、Oracle Databaseで使用するメモリ領域に対して小さすぎる場合、下記のエラーが発生する可能性があります。

ORA-00845: MEMORY_TARGET not supported on this system.

このエラーは、/dev/shmへの割当てが不足しているために発生します。初期化パラメーターMEMORY_TARGETまたはMEMORY_MAX_TARGETを設定するSGAとPGAの合計メモリー・サイズは、オペレーティング・システムの共有メモリー・ファイル・システム (/dev/shm) のサイズより大きくできません。

本エラーを防ぐために、あらかじめ下記のコマンドによって、/dev/shmのサイズを変更して下さい。

※網掛け部分は、適切な値を入力して下さい。

(以下のコマンドをrootユーザーで実行します。)

```
# mount -t tmpfs shmfs -o size=7g /dev/shm
```

システムを再起動後も変更をした値を保持するために、/etc/fstabに下記のエントリを追加します。

```
shmfs /dev/shm tmpfs size=7g 0
```

本件の詳細は、オラクル社による下記マニュアルを確認して下さい。

-
- Oracle Technology Network -OTN- (<https://www.oracle.com/technetwork/jp>)
 - Oracle® Database データベース管理者リファレンス 19c for Linux and UNIX System-Based Operating Systems
 - ・ A LinuxシステムでのOracle Databaseの管理
 - ・ A.5 共有リソースの割当て
-

4.9 リスナーの作成

フローティングIPアドレスを用いてデータベースにアクセスするためには、各ノードにて以下の作業が必要です。以下の作業は、**oracle**ユーザーで実施します。

- listener.oraの作成
- tnsnames.oraの作成
- パラメーター「LOCAL_LISTENER」の設定

片方向スタンバイ構成の場合、LISTENERがsid1をリスニングします。双方向スタンバイ構成の場合、通常運用時はLISTENERがノード1でデータベースsid1をリスニングし、LISTENER2がノード2でデータベースsid2をリスニングします。

■ listener.oraの作成

ノード1、ノード2のローカルパーティションに、listener.oraファイルを作成します。
listener.oraファイルは、\$ORACLE_HOME/network/adminに配置します。

片方向スタンバイ構成でのlistener.oraファイルの例 (\$ORACLE_HOME/network/admin/listener.ora)

```
LISTENER =  
  (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
```

※ポート番号(PORT=XXXX)はリスナーごとに異なる値を設定してください。

※網掛けの部分は、実際の環境に応じて設定して下さい。

※双方向スタンバイの場合のファイル例は、

『6.1.4 双方向スタンバイ構成の場合のリスナーとリスナーサービスの作成例』を確認してください。

■ tnsnames.oraの作成

listener.oraと同様に、ノード1、ノード2のローカルパーティションに、tnsnames.oraファイルを作成します。

tnsnames.oraファイルも、\$ORACLE_HOME/network/adminに配置します。

片方向スタンバイ構成でのtnsnames.oraファイルの例 (\$ORACLE_HOME/network/admin/tnsnames.ora)

```
listener=
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
    )
  )

SID1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid1)
    )
  )

SID1PDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid1pdb)
    )
  )
```

※ポート番号(PORT=XXXX)はリスナーごとに異なる値を設定してください。

※網掛けの部分は、実際の環境に応じて設定して下さい。

※双方向スタンバイの場合のファイル例は、

『6.1.4 双方向スタンバイ構成の場合のリスナーとリスナーサービスの作成例』を確認してください。

■ パラメーター「LOCAL_LISTENER」の変更

sysユーザーでデータベースsid1に接続し、サーバーパラメーターファイル上の「LOCAL_LISTENER」パラメーターを変更します。

データベースsid1が稼働しているノードに接続し、下記コマンドを実行します。

※両ノードから実行する必要はありません。

```
$ sqlplus / as sysdba
SQL> ALTER SYSTEM SET LOCAL_LISTENER='listener';
```

※双方向スタンバイの場合の変更は、

『6.1.4 双方向スタンバイ構成の場合のリスナーとリスナーサービスの作成例』を確認してください。

4.10 CLUSTERPROの設定②

データベース作成後に、データベースのフェイルオーバーに必要なexecリソースと、データベース監視を行うモニタリソースを作成します。

4.10.1 execリソースの作成

本章では、Cluster WebUIからexecリソースを作成する手順を紹介します。

execリソースでは、フェイルオーバーグループの起動、終了、移動時、またはフェイルオーバーの発生時に、実行するシェルスクリプト等を登録することが出来ます。

本手順では、データベースの起動を行うステートメントが、環境変数 `$CLP_EVENT` の値が”START”または”FAILOVER”の場合に実行されるように開始スクリプトを作成し、execリソースとして作成します。同様に、データベースの停止を行うステートメントが、環境変数 `$CLP_EVENT` の値が”START”または”FAILOVER”の場合に実行されるように、停止スクリプトを作成し、execリソースとして作成します。

スクリプトは、Cluster WebUI からStart script, Stop scriptとして雛形が提供されるため、雛形を用いて作成します。

なお、execリソースは各フェイルオーバーグループに対して作成する必要があります。

スクリプトで記述したエラーを出力する場合は、CLUSTERPROのリファレンスガイドを確認の上、EXECリソースのログ出力先を指定してください。

※ 何も指定しない場合、ログは出力されません。

以下、failover1（データベースsid1）でexecリソースを作成する際の設定例です。

※双方向スタンバイ環境では、本手順を参考にsid2用のスクリプトリソースを設定します。

- 1) 設定モードで、対象のフェイルオーバーグループから、「リソースの追加」をクリックします。



- 2) タイプから「EXECリソース」を選択し、execリソース名を設定します。設定したら「次へ」をクリックします。

- 3) 「規定の依存関係に従う」のチェックを外し、**exec**リソースが先の手順で作成したディスクリソースと**fip**リソースに依存するように設定します。設定したら「次へ」をクリックします。

グループのリソース定義 | failover1 exec ×

情報 ✓ → 依存関係 → 復旧動作 → 詳細

既定の依存関係に従う ☐

依存するリソース

名前	リソースタイプ
disk1	ディスクリソース
fip1	フローティングIPリソース

← 追加

→ 削除

利用可能なリソース

名前

利用可能なリソースはありません

◀ 戻る 次へ ▶ キャンセル

- 4) **exec**リソースの復旧動作を設定し、「次へ」をクリックします。
※環境に合わせて復旧動作を設定して下さい。

グループのリソース定義 | failover1 exec ×

情報 ✓ → 依存関係 ✓ → 復旧動作 → 詳細

活性異常検出時の復旧動作

活性リトライしきい値* 0 回

フェイルオーバーしきい値* 1 回

最終動作* 何もしない(次のリソースを活性しない)

☐ 最終動作前にスクリプトを実行する 設定

非活性異常検出時の復旧動作

非活性リトライしきい値* 0 回

最終動作* クラスタサービス停止とOSシャットダウン

☐ 最終動作前にスクリプトを実行する 設定

◀ 戻る 次へ ▶ キャンセル

- 5) 編集するスクリプトを選択し、「編集」をクリックします。

グループのリソース定義 | failover1

情報 → 依存関係 → 復旧動作 → 詳細

☐ ユーザアプリケーション

☒ この製品で作成したスクリプト

編集 表示 置換

スクリプト一覧

種類	名前
Start Script	start.sh
Stop Script	stop.sh

調整

戻る 完了 キャンセル

※双方向スタンバイの場合にはfailover2でも同様にスクリプトが必要です。スクリプト例は、『6.1.6 双方向スタンバイ構成の場合のスクリプト例』を確認してください。

以下はstart.shの編集の例です。

網掛けの行2箇所を、下記のとおり追記します。

※それぞれの設定値は、環境に合わせて変更してください。また必要に応じて、異常発生時の中断／リカバリ処理をスクリプト内に追記してください

start.shの編集例

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

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l oracle -c 'lsnrctl start listener'
        if [ $? -ne 0 ]
        then
            echo "listener start error"
            exit 1
        fi

        su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog
        @/oradata/sid1/startup.sql'

        if [ $? -ne 0 ]
        then
            echo "db start error"
            su -l oracle -c 'lsnrctl stop listener'
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
```

```

        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l oracle -c 'lsnrctl start listener'
        if [ $? -ne 0 ]
        then
            echo "listener start error"
            exit 1
        fi

        su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog
@/oradata/sid1/startup.sql'

        if [ $? -ne 0 ]
        then
            echo "db start error"
            su -l oracle -c 'lsnrctl stop listener'
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0

```

以下はstop.shの編集の例です。

網掛けの行2箇所を、下記のとおり追記します。

※それぞれの設定値は、環境に合わせて変更してください。また必要に応じて、異常発生時の中断／リカバリ処理をスクリプト内に追記してください

stop.shの編集例

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

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog
@/oradata/sid1/shutdown.sql'

        if [ $? -ne 0 ]
        then
            echo "db shutdown error"
            clplogcmd -m "db shutdown error" -i 1 -l ERR
        fi

        su -l oracle -c 'lsnrctl stop listener'
        if [ $? -ne 0 ]
        then
            echo "listener stop error"
            clplogcmd -m "listener stop error" -i 2 -l ERR
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l oracle -c 'export ORACLE_SID=sid1;sqlplus /nolog
@/oradata/sid1/shutdown.sql'

        if [ $? -ne 0 ]
        then
            echo "db shutdown error"
            clplogcmd -m "db shutdown error" -i 1 -l ERR
        fi
    fi
```

```

su -l oracle -c 'lsnrctl stop listener'
if [ $? -ne 0 ]
then
    echo "listener stop error"
    clplogcmd -m "listener stop error" -i 2 -l ERR
fi

if [ "$CLP_SERVER" = "HOME" ]
then
    echo "FAILOVER2"
else
    echo "ON_OTHER2"
fi
else
    echo "ERROR_DISK from FAILOVER"
    exit 1
fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0

```

変更したら、「OK」をクリックし、保存してください。

- 6) 「スクリプト一覧」画面左下の「調整」をクリックし、「EXECリソース調整プロパティ」画面で開始/終了各々のタイムアウト値を設定してください。「OK」をクリックし、「完了」をクリックします。

※デフォルトのタイムアウト値は1800秒のため、障害発生からフェイルオーバーまで30分を要する可能性があります。要件に応じて適切な値を設定してください。

- 7) 上記の「start.sh」「stop.sh」で使用されるスクリプト「startup.sql」と、「shutdown.sql」を用意します。以下の手順は、ノード1 およびoracleユーザーで実施します。

本編では、「startup.sql」と「shutdown.sql」を、切替パーティション上の”/oradata/sid1”に配置します。網掛けの部分は、実際の環境に応じて設定して下さい。

※双方向スタンバイの場合にはfailover2でも同様にスクリプトが必要です。
スクリプト例は、
『6.1.6 双方向スタンバイ構成の場合のスクリプト例』を確認してください。

startup.sqlの例

```
whenever sqlerror exit 1
connect sys/password as sysdba
startup pfile=/oradata/sid1/init.ora
alter pluggable database all open;
exit;
```

非CDB構成のデータベースを作成した場合、「alter pluggable database all open;」は不要です。

shutdown.sqlの例

```
whenever sqlerror exit 1
connect sys/password as sysdba
shutdown immediate
exit;
```

パスワードはsysユーザーのパスワードを指定します。（Oracle DatabaseのOS認証の機能を使用する場合はsysユーザー名とパスワードを指定する必要はありません）

初期化パラメーターファイルは切替パーティション、またはデータパーティション上に配置し、startup.sqlではフルパスで初期化パラメーターファイルを指定します。

- 8) サーバーパラメーターファイル（SPFILE）を切替パーティション，またはデータパーティション上に配置して使用するには、以下のようなサーバーパラメーターファイルのフルパスのみを記述した初期化パラメーターファイルを作成します。

※双方向スタンバイの場合にはfailover2でも同様にスクリプトが必要です。スクリプト例は、『6.1.6 双方向スタンバイ構成の場合のスクリプト例』を確認してください

初期化パラメーターファイルの例

本環境の場合の設定例は以下のようになります。

サーバーパラメーターファイル（spfilesid1.ora）を/oradata/sid1へ複製してください。

- 以下のコマンドをoracleユーザーで実行してください

```
$ cd $ORACLE_HOME/dbs
$ cp -p spfilesid1.ora /oradata/sid1/
```

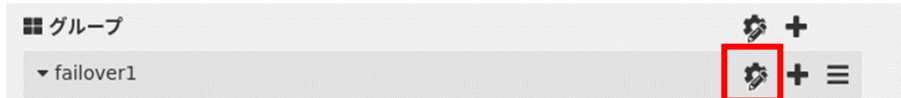
/oradata/sid1/init.oraを作成し、ファイル内に以下を記述してください。

```
$ vi /oradata/sid1/init.ora
（ファイル内に以下を記述する）
spfile=/oradata/sid1/spfilesid1.ora
```

上記で作成した初期化パラメーターをstartupコマンドのpfile句に指定することで、切替パーティション，またはデータパーティション上に配置したサーバーパラメーターファイルを使用してデータベースが起動されます。

9) リソース一覧で作成したリソースと依存関係を確認します。

グループ(failover1)のプロパティをクリックし、「全体の依存関係」タブを選択します。



「全体の依存関係」タブで設定したリソースの依存関係を確認します。
以下の画面では「フローティングIPリソース」を”fip1”、「ディスクリソース」を”disk1”、「EXECリソース」を”exec1”として設定した場合の表示例を示しています。

※ミラーディスク構成の場合は、「ディスクリソース」は「ミラーディスクリソース」となります。

依存関係は下図のとおり、ディスクリソースとfipリソースの活性化を完了してから、execリソースの活性化が開始されるよう設定します。

グループのプロパティ | failover1

リソース一覧 情報 起動サーバ 属性 起動待ち合わせ 停止待ち合わせ 全体の依存関係

活性時 非活性時

図を表示

深度	名前	依存リソース名	タイプ
0	fip1	none	
1	disk1	--	フローティングIPリソース
		--	仮想IPリソース
		--	ダイナミックDNSリソース
		--	ボリュームマネージャリソース
		--	AWS Elastic IPリソース
		--	AWS仮想IPリソース
		--	Azureプロポーポートリソース
		--	AWS DNSリソース
		--	Azure DNSリソース
		--	AWSセカンダリIPリソース
2	exec1	disk1	ディスクリソース
		fip1	フローティングIPリソース

OK キャンセル 適用

確認後、問題なければ「OK」をクリックします。

4.10.2 モニタリソースの作成

モニタリソースであるOracle監視リソースを作成することで、データベース上に監視用テーブルを作成し、SQL文を用いたデータの書き込みと読み込みによる監視が可能です。

ただし、Oracle 監視リソースはCLUSTERPROのオプション機能になるため、別途「Database Agent」ライセンスが必要です。ご使用になる前に製品ライセンスを入手して、章番号『4.4 CLUSTERPRO Serverのインストールとライセンス登録』を参考に、ライセンスを登録して下さい。

Oracle 監視リソースによる監視の結果、以下の場合にCLUSTERPROは異常と判断します。

(レベル0)データベースステータス

Oracle Databaseの管理テーブル(V\$INSTANCE表)を参照し、DBの状態(インスタンスの状態)を確認します。

異常判定：①管理テーブルのステータス(status)が未起動

(MOUNTED, STARTED)の場合

②管理テーブルのデータベースステータス(database_status)が未起動

(SUSPENDED, INSTANCE RECOVERY)の場合

(レベル1)selectでの監視

事前作成した監視テーブルに対して参照のみを行う監視です。監視テーブルに対して実行するSQL文は (select) です。

異常判定：①データベースへの接続に失敗した場合

②SQL文の発行に対する応答で異常が通知された場合

(レベル2)update/selectでの監視

事前作成した監視テーブルに対して更新も行う監視です。SQL文の発行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行するSQL文は (update / select) です。

監視開始時に監視テーブルを自動で作成した場合、監視テーブルに対して実行するSQL 文は(create / insert)です。

異常判定：①データベースの接続に失敗した場合

②SQL文の発行に対する応答で異常が通知された場合

③書き込んだデータと読んだデータが一致していない場合

(レベル3)create/dropでの監視

監視テーブルに対しての更新に加え監視テーブルの作成・削除も毎回行います。SQL文の発行により最大11桁の数値データの書き込みと読み込みを実行します。監視テーブルに対して実行するSQL文は(create / insert / select / drop) です。

異常判定：①データベースの接続に失敗した場合

②SQL文の発行に対する応答で異常が通知された場合

③書き込んだデータと読んだデータが一致していない場合

※デフォルトの監視レベルは「レベル2」です。

なお、レベル1の監視を選択した場合、CLUSTERPROは監視中にテーブルの作成を行いません。事前に、手動にて監視テーブルを作成しておく必要があります。

監視テーブル作成例 (監視テーブル名をorawatchとする場合)

```
SQL> create table orawatch (num number(11,0) primary key);
SQL> insert into orawatch values (0);
SQL> commit;
```

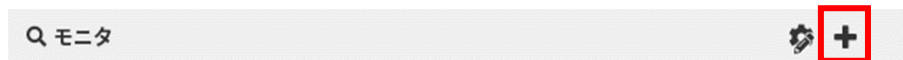
※網掛けの部分は、監視するテーブルを指定してください。

※CLUSTERPRO Oracle監視リソースのユーザー名(後述)に指定するユーザーのスキーマに作成してください。

本機能を利用する場合の手順は以下のとおりです。
なお、本手順は要件に応じて各グループに対して行って下さい。

※以下、failover1（データベースsid1）での設定例です。
※双方向スタンバイ環境では、本手順を参考にsid2用のOracle監視リソースを設定します。

- 1) 設定モードで、「モニタ」から、「モニタリソースの追加」をクリックします。



- 2) タイプとして「Oracleモニタ」を選択し、名前を入力します。
入力したら「次へ」をクリックします。

- 3) 対象リソースとして、先に作成したexecリソースを選択した後に、
「監視を行うサーバを選択する」の右の「サーバ」をクリックします。
「異常検出サーバ」画面では「全てのサーバ」にチェックが入っている事を
確認して、「キャンセル」で画面を閉じます。
「タイムアウト発生時に監視プロセスのダンプを採取する」の項目は、要件に
応じてチェックします。
「次へ」をクリックします。

The screenshot shows the 'Monitoring Resource Definition' (モニタリソースの定義) dialog box. The breadcrumb navigation is: 情報 (Information) → 監視(共通) (Monitoring (Common)) → 監視(固有) (Monitoring (Specific)) → 回復動作 (Recovery Action). The settings are as follows:

- インターバル (Interval): 60 秒 (seconds)
- タイムアウト (Timeout): 120 秒 (seconds)
- タイムアウト発生時に監視プロセスのダンプを採取する (Dump monitoring process on timeout): ☐
- タイムアウト発生時にリトライしない (Do not retry on timeout): ☐
- タイムアウト発生時動作 (Action on timeout): 回復動作を実行する (Execute recovery action)
- リトライ回数 (Retry count): 2 回 (times)
- 監視開始待ち時間 (Monitoring start wait time): 0 秒 (seconds)
- 監視タイミング (Monitoring timing): ☒ 常時 (Always), ☐ 活性時 (Active time)
- 対象リソース (Target resource): exec1 (with a '参照' (Reference) button)
- nice値 (Nice value): 0 (with a slider)
- 監視を行うサーバを選択する (Select server to monitor): サーバ (Server) button
- 監視処理時間メトリクスを送信する (Send monitoring process time metrics): ☐

Buttons at the bottom: ◀ 戻る (Back), 次へ ▶ (Next), キャンセル (Cancel).

The screenshot shows the 'Exception Detection Server' (異常検出サーバ) dialog box. The options are:

- ☒ 全てのサーバ (All servers)
- ☐ 独自に設定する (Set independently)

Below the radio buttons, there are two input fields for server names:

- 名前 (Name): node1
- node2

Buttons at the bottom: OK, キャンセル (Cancel), 適用 (Apply).

※「監視開始待ち時間」について

Oracle監視リソースによる監視を開始するまでの待ち時間は、監視開始待ち時間に設定された値と、タイムアウトに設定された時間のうち大きいほうの値を使用します。

- 4) 「監視レベル」, 「接続文字列」, 「ユーザー名」, 「パスワード」, 「認証方式」, 「監視テーブル名」, 「ORACLE_HOME」, 「文字コード」, 「ライブラリパス」を設定します。

「監視方式」の項目は、デフォルトの「リスナーとインスタンスを監視」を使用します。
「監視レベル」の項目は、本手順書ではデフォルトの「レベル2」とします。

「接続文字列」の項目は、本手順書ではsid1を入力します。

「ユーザー名」の項目で指定するユーザについて、デフォルトでsysとなっていますが、別途監視用ユーザを作成して設定する場合、その監視用ユーザには(CREATE TABLE, DROP ANY TABLE, SELECT, INSERT, UPDATE)といったアクセス権の付与が必要です。本手順書ではsystemとしています。

「パスワード」の項目は、DBCA実行時に設定したパスワードを入力します。

「認証方式」の項目はデフォルトの「DEFAULT」を選択します。

「監視テーブル名」の項目では事前に作成した表名、「orawatch」を入力します。

「ORACLE_HOME」の項目は環境変数として設定したORACLE_HOMEのpathを入力します。バージョンを示すディレクトリ名(本手順書では19.0.0)は適宜変更してください。

「文字コード」の項目は、本手順書ではJAPANESE_JAPAN.JA16EUC(デフォルト値)とします。

「ライブラリパス」の項目ではプルダウンより、末尾に「libclntsh.so.19.1」があるパスを選択してください。その際、バージョンを示すディレクトリ名がデフォルトで19.0.0になっているため、プルダウンから選択した後に適宜ディレクトリ名を手入力に変更してください。

「障害発生時にアプリケーションの詳細情報を採取する」の項目は、要件に応じてチェックしてください。

「Oracleの初期化中またはシャットダウン中をエラーにする」の項目は、CDB環境の場合には必ずチェックを入れてください。非CDB環境の場合には要件に応じてチェックを入れてください。

※監視レベル1を選択した場合には、事前作成した監視用テーブルを「監視テーブル名」に設定します。

※監視レベル2で開始時に監視テーブルがない場合は、監視テーブルが自動で作成されます。

設定が終了したら、「次へ」をクリックします。

- 5) 回復対象にフェイルオーバーグループを選択します。
また、最終動作を環境に合わせて選択します。

「完了」をクリックして、Oracle監視リソースの作成は終了です。

4.10.3 設定項目の反映

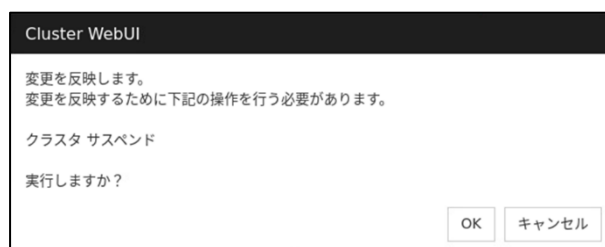
execリソース、モニタリソースの設定が完了したら、設定項目をCLUSTERPROへ反映します。

※双方向スタンバイ環境では、本手順を参考に、sid2用のフェイルオーバーグループの「設定の反映」をします。

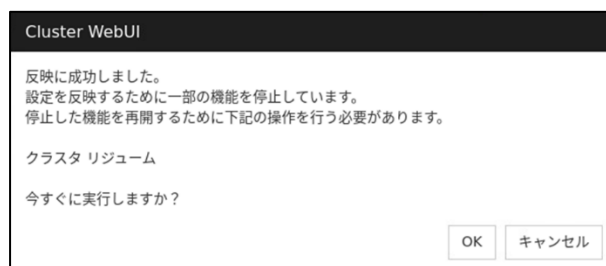
- 1) 「設定モード」画面の上部メニューから「設定の反映」をクリックします。



- 2) 「OK」をクリックして、各設定をサーバーに反映します。



- 3) 「OK」をクリックして、設定の反映は完了です。



- 4) 「操作モード」へ切り替えます。



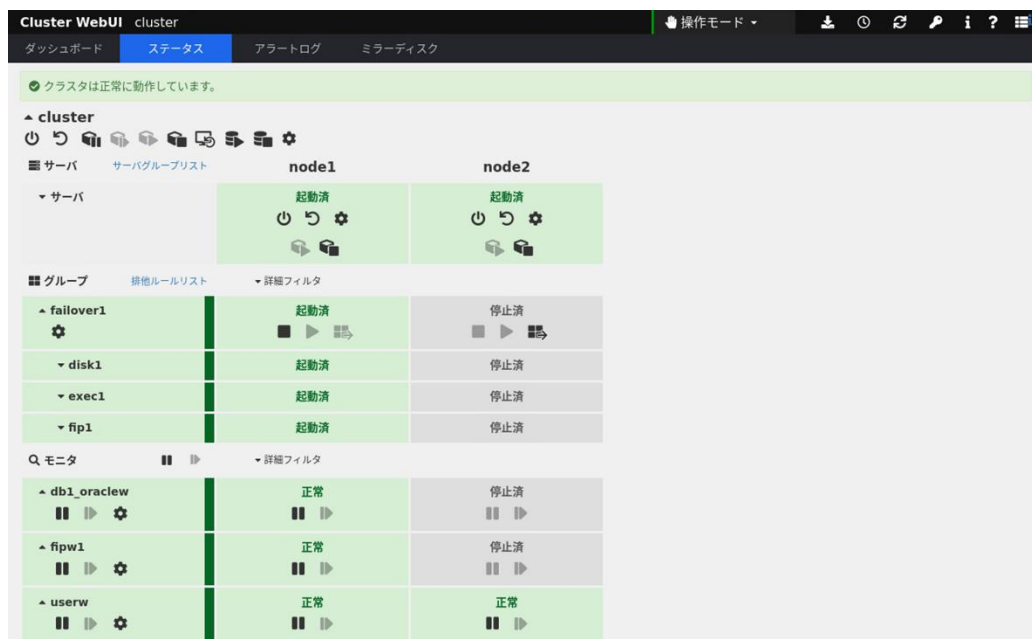
- 5) フェイルオーバーグループの停止を行います。
グループのfailover1から「グループの停止」をクリックします。
その後、「停止」をクリックします。



- 6) フェイルオーバーグループの開始を行います。
グループのfailover1から起動させたいサーバー上の「グループ起動」をクリックします。
その後、「起動」をクリックします。



グループが正常に起動することを確認してください。



以上で、CLUSTERPROの設定は完了です。

4.11 クライアントの設定

- 接続にフローティングIPアドレスを使用することで、フェイルオーバーが発生した後の再接続に、クライアントの設定を変更する必要がなくなります。設定例は以下をご参照下さい。

網掛けの部分は、実際の環境に応じて設定して下さい。

片方向スタンバイ構成でのtnsnames.oraファイルの例(\$ORACLE_HOME/network/admin/tnsnames.ora)

```
SID1 =                                #データベースsid1への接続文字列
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ENABLE = BROKEN)
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid1)
    )
  )

SID1PDB =                             #データベースsid1pdbへの接続文字列
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ENABLE = BROKEN)
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid1pdb)
    )
  )
```

※双方向スタンバイの場合のクライアントの設定例は、『6.1.7 双方向スタンバイ構成の場合のクライアントの設定例』を確認してください。

5. 注意事項

- 終了スクリプトがストールなどにより終了しない場合に備え、スクリプトのタイムアウト時間を設定してください。
- Oracle Databaseのインスタンス起動後に、実行／起動が必要なアプリケーションがある環境では、インスタンスの起動時間を考慮した設定を、スクリプトリソース上で行って下さい。適切な設定がされていないと、アプリケーションが正常に実行／起動されない可能性があります。
- Oracle Database 11g R1から、ユーザー認証パスワードの期限のデフォルトが180日となっています。この影響により、CLUSTERPROからOracle Databaseの監視を行うと、導入後180日で監視が失敗し、サーバーがCLUSETERPROによりシャットダウンされてしまう可能性があります。そのため、CLUSTERPROのOracle監視リソースで使用しているOracleのユーザーパスワードを定期的に変更するか、Oracleユーザーの認証パスワードの期限を無期限に変更するか、セキュリティ要件により実施してください。
※認証パスワードの期限変更は、ドキュメント等を参照してください¹¹。

¹¹ DocID 1704748.1 「ユーザ認証パスワードの有効期限を設定する方法について」

6. 付録

6.1 双方向スタンバイ構成の場合の設定例

双方向スタンバイ構成を採用する場合の設定例を紹介します。

6.1.1 双方向スタンバイ構成の場合のネットワーク・IPアドレス要件例

【双方向スタンバイ環境のネットワーク構成例】

種類	ホスト名	IPアドレス	
○ノード1			
パブリックIP	node1	固定	10.0.0.1
インタコネクトIP	node1-in	固定	192.168.0.1
○ノード2			
パブリックIP	node2	固定	10.0.0.2
インタコネクトIP	node2-in	固定	192.168.0.2
○フローティングIP			
フローティングIP1		固定	10.0.0.11
フローティングIP2		固定	10.0.0.12

6.1.2 双方向スタンバイ構成の場合のディスクパーティション作成例

共有ディスク構成の場合

パーティション		用途	備考
デバイス名	フォーマット		
/dev/sdb1	しない	ディスクハートビート用	
/dev/sdb2	ext4	sid1用のディスクリソース (データファイル等)	“/oradata/sid1” にマウントする
/dev/sdc1	ext4	sid2用のディスクリソース (データファイル等)	“/oradata/sid2” にマウントする

ミラーディスク構成の場合

パーティション		用途	備考
デバイス名	フォーマット		
/dev/sdb1	しない	クラスタパーティション	
/dev/sdb2	ext4	sid1用のディスクリソース (データファイル等)	“/oradata/sid1” にマウントする
/dev/sdc1	しない	クラスタパーティション	
/dev/sdc2	ext4	sid2用のディスクリソース (データファイル等)	“/oradata/sid2” にマウントする

6.1.3 双方向スタンバイ構成の場合の前提環境

双方向スタンバイ構成の構成例を紹介します。本手順は共有ディスク構成とミラーディスク構成の両方に対象としています。

sid1は通常運用時にnode1で起動し、待機系はnode2になります。同様に、sid2は通常運用時にnode2で起動し、待機系はnode1になります。

双方向スタンバイ環境のクラスタ環境構成例

クラスタサーバー環境		
	node1	node2
実 IP アドレス	10.0.0.1	10.0.0.2
ローカルデバイス	/dev/sda	/dev/sda

フェイルオーバーグループ情報(グループ 1)		
	node1	node2
フローティング IP アドレス	10.0.0.11	
切替パーティション, データパーティション	/dev/sdb2-> /dev/cp-diska2 (シンボリック・リンク) (マウントポイント : /oradata/sid1)	
クラスタパーティション ※ミラーディスク構成の場合	/dev/sdb1-> /dev/cp-diska1 (シンボリック・リンク) (マウントポイント : /oradata/sid1)	

フェイルオーバーグループ情報(グループ 2)		
	node1	node2
フローティング IP アドレス	10.0.0.12	
切替パーティション, データパーティション	共有ディスク構成の場合 : /dev/sdc1-> /dev/cp-diskb1 (シンボリック・リンク) ミラーディスク構成の場合 : /dev/sdc2-> /dev/cp-diskb2 (シンボリック・リンク) (マウントポイント : /oradata/sid2)	
クラスタパーティション ※ミラーディスク構成の場合	/dev/sdc1-> /dev/cp-diskb1 (シンボリック・リンク) (マウントポイント : /oradata/sid2)	

データベース環境		
	node1	node2
ORACLE_BASE	/u01/app/oracle	
ORACLE_HOME	/u01/app/oracle/product/19.0.0/dbhome_1	
(グループ 1)		
SID 名	sid1	
データベース名	sid1	
PDB 名 (CDB 構成のみ)	sid1pdb	

データファイル格納先	/oradata/sid1
REDO ログファイル格納先	
制御ファイル格納先	
アーカイブログファイルの出力先	/oradata/sid1/fast_recovery_area/SID1
初期化パラメーターファイル格納先 (テキスト形式)	/oradata/sid1
サーバーパラメーターファイル格納先	
(グループ 2)	
SID 名	sid2
データベース名	sid2
PDB 名 (CDB 構成のみ)	sid2pdb
データファイル格納先	/oradata/sid2
REDO ログファイル格納先	
制御ファイル格納先	
アーカイブログファイルの出力先	/oradata/sid1/fast_recovery_area/SID2
初期化パラメーターファイル格納先 (テキスト形式)	/oradata/sid2
サーバーパラメーターファイル格納先	

リスナー環境(グループ 1)		
	node1	node2
リスナー名	LISTENER	
FIP アドレス/ポート番号	10.0.0.11:1521	

リスナー環境(グループ 2)		
	node1	node2
リスナー名	LISTENER2	
FIP アドレス/ポート番号	10.0.0.12:1526	

6.1.4 双方向スタンバイ構成の場合のリスナーとリスナーサービスの作成例

双方向スタンバイ構成でのlistener.oraファイルの例
(\$ORACLE_HOME/network/admin/listener.ora)

```
LISTENER =  
  (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))  
  
LISTENER2 =  
  (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))
```

※ポート番号(PORT=XXXX)はリスナーごとに異なる値を設定してください。

※網掛けの部分は、実際の環境に応じて設定して下さい。

双方向スタンバイ構成でのtnsnames.oraファイルの例
(\$ORACLE_HOME/network/admin/tnsnames.ora)

```
listener =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))  
    )  
  )  
  
listener2 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))  
    )  
  )  
  
SID1 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = sid1)  
    )  
  )  
  
SID2 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = sid2)  
    )  
  )  
  
SID1PDB =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = sid1pdb)
```

```

    )
  )

SID2PDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid2pdb)
    )
  )
)

```

※ポート番号(PORT=XXXX)はリスナーごとに異なる値を設定してください。

※網掛けの部分は、実際の環境に応じて設定して下さい。

■ パラメーター「LOCAL_LISTENER」の変更

sysユーザーでデータベースsid1に接続し、サーバーパラメーターファイル上の「LOCAL_LISTENER」パラメーターを変更します。

データベースsid1が稼働しているノードに接続し、下記コマンドを実行します。

※両ノードから実行する必要はありません。

```

$ sqlplus / as sysdba
SQL> ALTER SYSTEM SET LOCAL_LISTENER='listener';

```

同様にデータベースsid2が稼働しているノードに接続し、下記コマンドを実行します。

※両ノードから実行する必要はありません。

```

$ sqlplus / as sysdba
SQL> ALTER SYSTEM SET LOCAL_LISTENER='listener2';

```

6.1.5 双方向スタンバイ構成の場合の環境変数設定例

双方向スタンバイの設定例

(ノード1の設定)

```
$ vi $HOME/.bash_profile
(下記行がない場合は最終行以降に追加します)
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_LANG=Japanese_Japan.UTF8
export ORACLE_SID=sid1
```

(ノード2の設定)

```
$ vi $HOME/.bash_profile
(下記行がない場合は最終行以降に追加します)
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export NLS_LANG=Japanese_Japan.UTF8
export ORACLE_SID=sid2
```

※網掛けの部分は、実際の環境に応じて設定して下さい。

6.1.6 双方向スタンバイ構成の場合のスクリプト例

以下はstart.shの編集の例です。

網掛けの行2箇所を、下記のとおり追記します。

※それぞれの設定値は、環境に合わせて変更してください。また必要に応じて、異常発生時の中断／リカバリ処理をスクリプト内に追記してください

start.shの編集例 (failover2用)

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

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l oracle -c 'lsnrctl start listener2'
        if [ $? -ne 0 ]
        then
            echo "listener start error"
            exit 1
        fi

        su -l oracle -c 'export ORACLE_SID=sid2;sqlplus /nolog
@/oradata/sid2/startup.sql'

        if [ $? -ne 0 ]
        then
            echo "db start error"
            su -l oracle -c 'lsnrctl stop listener2'
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l oracle -c 'lsnrctl start listener2'
        if [ $? -ne 0 ]
        then
            echo "listener start error"
            exit 1
        fi
```

```

su -l oracle -c 'export ORACLE_SID=sid2;sqlplus /nolog
@/oradata/sid2/startup.sql'

if [ $? -ne 0 ]
then
    echo "db start error"
    su -l oracle -c 'lsnrctl stop listener2'
    exit 1
fi

if [ "$CLP_SERVER" = "HOME" ]
then
    echo "FAILOVER2"
else
    echo "ON_OTHER2"
fi
else
    echo "ERROR_DISK from FAILOVER"
    exit 1
fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0

```

以下はstop.shの編集の例です。

網掛けの行2箇所を、下記のとおり追記します。

※それぞれの設定値は、環境に合わせて変更してください。また必要に応じて、異常発生時の中断／リカバリ処理をスクリプト内に追記してください

stop.shの編集例

```

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

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        su -l oracle -c 'export ORACLE_SID=sid2;sqlplus /nolog
@/oradata/sid2/shutdown.sql'

        if [ $? -ne 0 ]
        then
            echo "db shutdown error"
            clplogcmd -m "db shutdown error" -i 1 -l ERR
        fi

        su -l oracle -c 'lsnrctl stop listener2'
        if [ $? -ne 0 ]
        then

```

```

        echo "listener stop error"
        clplogcmd -m "listener stop error" -i 2 -l ERR
    fi

    if [ "$CLP_SERVER" = "HOME" ]
    then
        echo "NORMAL2"
    else
        echo "ON_OTHER1"
    fi
else
    echo "ERROR_DISK from START"
    exit 1
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        su -l oracle -c 'export ORACLE_SID=sid2;sqlplus /nolog
@/oradata/sid2/shutdown.sql'

        if [ $? -ne 0 ]
        then
            echo "db shutdown error"
            clplogcmd -m "db shutdown error" -i 1 -l ERR
        fi

        su -l oracle -c 'lsnrctl stop listener2'
        if [ $? -ne 0 ]
        then
            echo "listener stop error"
            clplogcmd -m "listener stop error" -i 2 -l ERR
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0

```

設定を終えたら、「OK」をクリックし、「完了」をクリックする。

上記の「start.sh」「stop.sh」で使用されるスクリプト「startup.sql」と、「shutdown.sql」を用意します。以下の手順は、oracleユーザーで実施します。

本編では、「startup.sql」と「shutdown.sql」を、切替パーティション上の”/oradata/sid2”に配置します。網掛けの部分は、実際の環境に応じて設定して下さい。

startup.sqlの例

```
whenever sqlerror exit 1
connect sys/password as sysdba
startup pfile=/oradata/sid2/init.ora
alter pluggable database all open;
exit;
```

非CDB構成のデータベースを作成した場合、「alter pluggable database all open;」は不要です。

shutdown.sqlの例

```
whenever sqlerror exit 1
connect sys/password as sysdba
shutdown immediate
exit;
```

パスワードはsysユーザーのパスワードを指定します。（Oracle DatabaseのOS認証の機能を使用する場合はsysユーザー名とパスワードを指定する必要はありません）

初期化パラメーターファイルは切替パーティション、またはデータパーティション上に配置し、startup.sqlではフルパスで初期化パラメーターファイルを指定します。

初期化パラメーターファイルの例

本環境の場合の設定例は以下のようになります。

サーバーパラメーターファイル（spfilesid2.ora）を/oradata/sid2へ複製してください。

- ・以下のコマンドをoracleユーザーで実行してください。

```
$ cd $ORACLE_HOME/dbs
$ cp -p spfilesid2.ora /oradata/sid2/
```

/oradata/sid2/init.oraを作成し、ファイル内に以下を記述してください。

```
$ vi /oradata/sid2/init.ora
（ファイル内に以下を記述する）
spfile=/oradata/sid2/spfilesid2.ora
```

上記で作成した初期化パラメーターをstartupコマンドのpfile句に指定することで、切替パーティション、またはデータパーティション上に配置したサーバーパラメーターファイルを使用してデータベースが起動されます。

6.1.7 双方向スタンバイ構成の場合のクライアントの設定例

双方向スタンバイ構成でのtnsnames.oraファイルの例(\$ORACLE_HOME/network/admin/tnsnames.ora)

```
SID1 =                                #データベースsid1への接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ENABLE = BROKEN)
    (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = sid1)
  )
)

SID2 =                                #データベースsid2への接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ENABLE = BROKEN)
    (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = sid2)
  )
)

SID1PDB =                             #データベースsid1pdbへの接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ENABLE = BROKEN)
    (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.11) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = sid1pdb)
  )
)

SID2PDB =                             #データベースsid2pdbへの接続文字列
(DESCRIPTION =
  (ADDRESS_LIST =
    (ENABLE = BROKEN)
    (ADDRESS = (PROTOCOL = TCP) (HOST = 10.0.0.12) (PORT = 1526))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = sid2pdb)
  )
)
```

※網掛けの部分は、実際の環境に応じて設定して下さい。

6.2 ライセンス定義

Oracle Databaseのライセンス、保守につきましては、弊社担当営業にご相談ください。