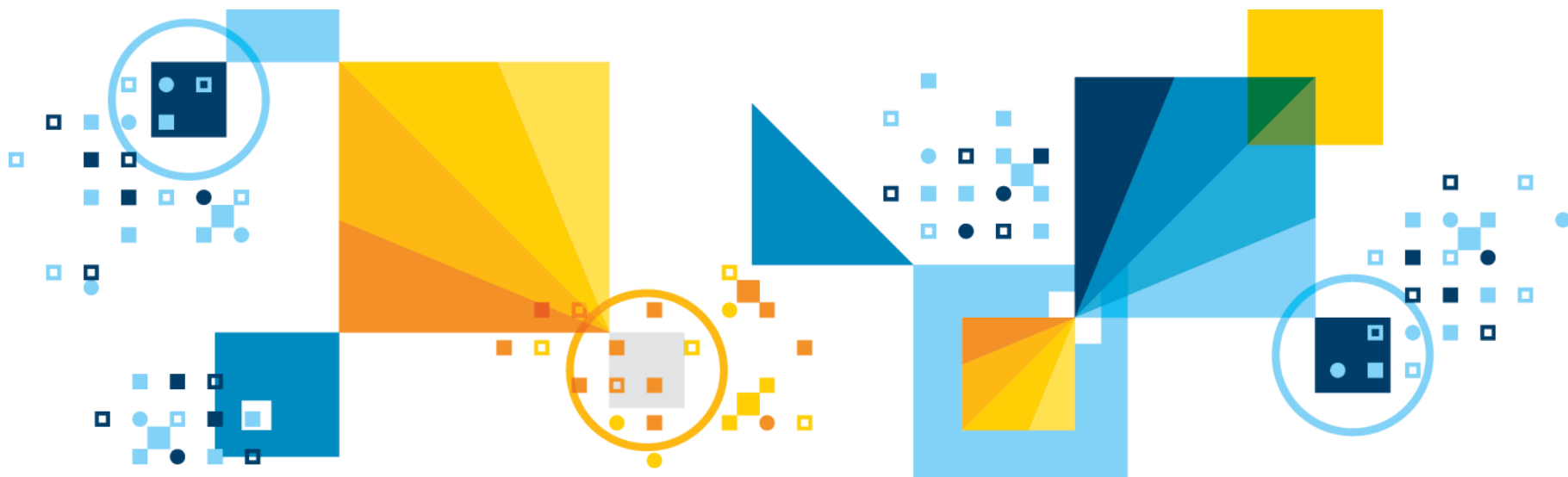


HADR+Pacemaker連携 構築運用ガイド

(db2cmによるDb2 HADR制御用Pacemaker構成/運用)

2021/04 初版(Db2 v11.5.5対応)
2021/10 Db2 v11.5.6対応



Disclaimer

- 本ガイドは、「db2cmによる高可用性構成ガイド」のDb2 v11.5.6対応版となります。
- 本ガイドはDb2 LUWのHADR環境の構築の知識を持つ読者を対象にしております。そのため、OSおよびDb2 LUW(およびHADR)の導入構築に関する詳細説明は省略しております。
- 本ガイド作成に際しては細心の注意を払っておりますが、正式なレビューを受けたものではなく、必ずしも記述内容を保証するものではありません。ご使用になられる場合には、必要に応じて正規かつ最新のマニュアルによる確認、修正情報の調査、および事前のテストなど十分に行われるようお願いいたします。
- また、構成例、構築手順、および操作手順については、全ての環境に対して適用できることを保障するものではありません。実環境での検証を十分に行ってください。
- 本ガイドでは以下の構成を前提として記載しております
 - OS: Red Hat Enterprise Linux Server release 8.2(Ootpa)
 - Cluster: Pacemaker 2.0.4 + Corosync 3.0.4(Db2 v11.5.6 導入イメージに含まれる)
 - DB: Db2 LUW 11.5.6
- 本ガイドに示す、コマンド出力結果などの情報は、ご利用のバージョンによって、表記内容が異なる場合があります。

目次(1/2)

1. Pacemaker/db2cm概要
 - 1.1 Pacemakerとは
 - 1.2 Pacemakerの提供形態
 - 1.3 IBMからDb2制御用にPacemakerを提供開始
 - 1.4 db2cm概要
 - 1.5 db2cm - Db2クラスター・マネージャー・ユーティリティー
2. 本ガイドの前提
3. Pacemaker導入・構成
 - 3.1 Db2導入時の注意点
 - 3.2 Pacemaker導入手順
 - 3.3 QDeviceクォーラムのインストール
4. db2cmコマンドによるPacemaker構成
 - 4.1 HADR構成の環境構成図
 - 4.2 Db2/Pacemakerパラメータ抜粋
 - 4.3 Pacemakerのリソース定義概要
 - 4.4 db2cmコマンドによるPacemaker構成
5. PacemakerによるHADR運用
 - 5.1 正常運用
 - 5.2 障害対応

目次(2/2)

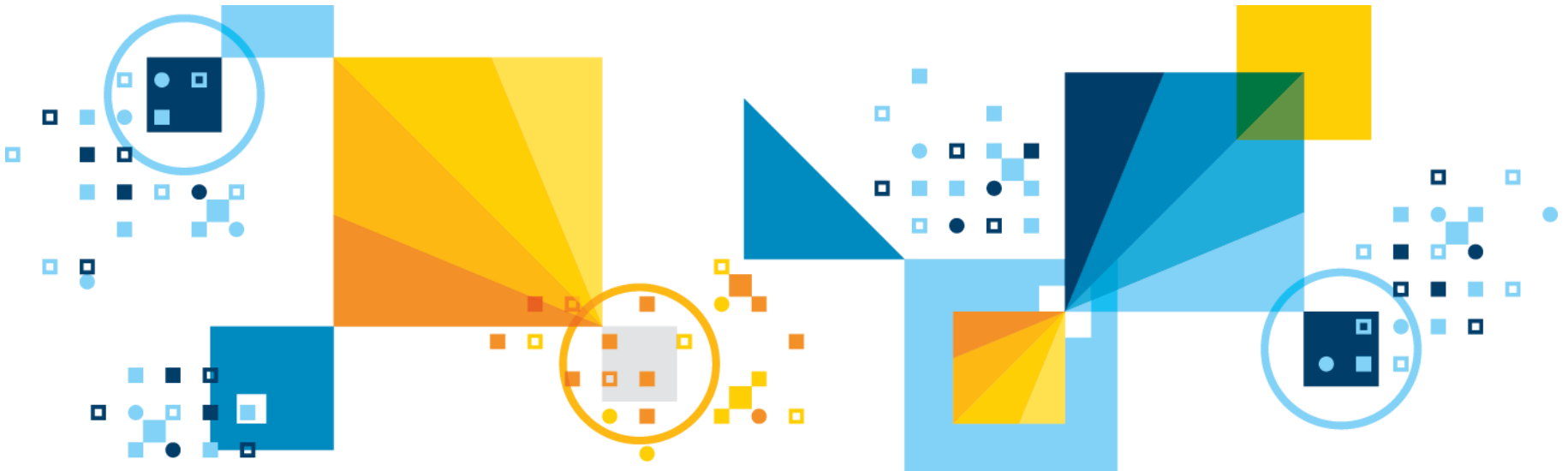
6. その他運用

- 6.1 db2cm(HADR)構成の削除手順
- 6.2 計画停止時のローリングメンテナンス
- 6.3 HADRデータベースのリソース追加登録
- 6.4 インスタンスとHADRデータベースのリソース追加登録
- 6.5 既存のTSAクラスターからPacemakerクラスターへの変換
- 6.6 クラスター構成情報のバックアップ取得・リストア

7. 参考:Pacemaker導入(Db2 V11.5.5以前)

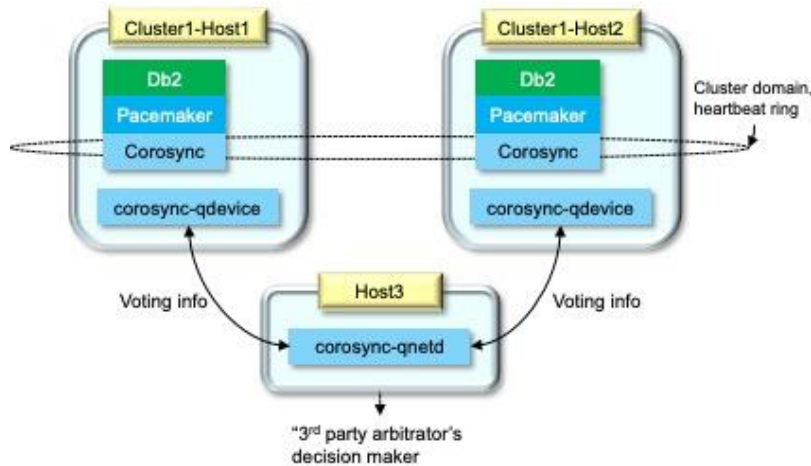
- 7.1 Pacemaker導入手順(Db2 V11.5.5以前)
- 7.2 QDevice クォーラムのインストール(Db2 V11.5.5以前)

1. Pacemaker/db2cm概要



1.1 Pacemakerとは

- コンピュータ・クラスターで使われる高可用性リソース・マネージャ・ソフトウェア
- オープンソースとして2004年に登場
- 2007年頃まではLinux-HAプロジェクトの一部であり、そののち独立したプロジェクトとしてスピンアウト



- 公式サイト: <http://clusterlabs.org/>
- Linux-HA: http://www.linux-ha.org/w/index.php?title=Main_Page&setlang=ja
- Linux-HA Japanプロジェクト: <http://linux-ha.osdn.jp/>
- Red Hat HIGH AVAILABILITY ADD-ON リファレンス: https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/index

1.2 Pacemakerの提供形態

- 従来、提供されるバイナリーは大きく2系統存在した
- Linux-HAコミュニティ版
 - pacemaker 1.1 + corosync 2 (pacemaker 2系以降のイメージは、各プロバイダーによる提供の方向)
 - ・ Pacemakerのインターフェースコマンドは、以前から使用されているcrmコマンドと、新しく提供されたpcsコマンドが存在する
 - ・ crmコマンドを推奨しており、コミュニティによるインターネット上の情報は殆どがcrmコマンドを使用
- 各プロバイダーによる提供版
 - RHELでは、RHEL7以降HA add-onとしてPacemakerを正式採用
 - Red Hat社からの入手、サポートには有償のHA add-onサブスクリプションが必要
 - 対応OS
 - ・ RHEL7: pacemaker 1.1 + corosync 2、RHEL8: pacemaker 2 + corosync 3
 - ・ SLES, Ubuntuなどでも採用済み
 - 各社、pcsコマンドを使用。Red Hat社のドキュメント・ナレッジなどはpcsコマンドによる手順のみ記載
- これらは、(Db2を含む)多くのサービスに対する制御ロジック(リソース・エージェント)を提供しており、db2リソース・エージェントにはHADR制御ロジックも含む
 - HADR制御 db2 (Master/Slave)
https://www.systutorials.com/docs/linux/man/7-ocf_heartbeat_db2/

1.3 IBMからDb2制御用にPacemakerを提供開始(1/2)

- 前ページ記載の従来の2系統とは別の提供形態
- HADRの自動フェイルオーバーのためにPacemakerをクラスター・マネージャーとして使用する機能が登場
 - v11.5.4ではテクニカル・プレビュー版として提供
 - v11.5.5から正式にGAされ、サポートを開始
 - v11.5.6からPacemakerクラスター・マネージャーがパッケージ化され、Db2と共にインストール可能
- Pacemaker/corosyncのイメージはIBMから提供
- 対応OS(2021/10 現在)
 - Intel LinuxおよびLinux on IBM Z(AIXやLinux on POWERは当ドキュメント時点で未対応)
 - RHEL 8.1
 - RHEL 8.2(Db2 v11.5.6以降のみ)
 - SUSE 15 SP1
 - SUSE 15 SP2(Db2 v11.5.6以降のみ)
- ユーザーインターフェース
 - IBM提供のdb2cmコマンドまたはcrmコマンド

1.3 IBMからDb2制御用にPacemakerを提供開始(2/2)

■ IBMによるPacemakerのサポート

- IBMサポートを受けるには、v11.5.5以降のIBM提供のPacemaker/corosyncイメージ、db2cmユーティリティを使用して構成されている場合に限られる
- v11.5.4版(テクニカル・プレビュー版)のPacemakerイメージを使用している場合はサポート外
- Pacemaker に関する制約事項
<https://www.ibm.com/docs/ja/db2/11.5?topic=utility-restrictions-pacemaker>
- Pacemaker を使用した統合ソリューションの前提
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-prerequisites-integrated-solution-using>

1.4 db2cm概要

db2cm (Db2 クラスタ・マネージャユーティリティ)とは、

- Pacemaker で管理されるクラスタ環境のDb2データベースを構成および管理するために使用できるコマンドラインベースのユーティリティ
- Db2と共にインストール可能 (Db2 v11.5.6以降)
 - Db2 v11.5.5では、Db2のインストーラーとは別にインストールが必要
- サポートされるクラスタ構成
 - HADR構成 (マルチスタンバイ構成、Cloud上での構成も含む)
 - ・ マルチスタンバイ構成についての考慮点は以下の通り
 - ・ Pacemakerを使用した複数スタンバイのサポートは、v11.5.5以降
 - ・ 自動フェイルオーバーはプリンシパル・スタンバイでのみサポート
 - ・ 補助スタンバイの1つにテークオーバーする場合は、手動でテークオーバーを実行し、そのスタンバイをプライマリーにする必要がある
 - ・ HADR 複数スタンバイ・データベース
<https://www.ibm.com/docs/ja/db2/11.5?topic=solution-hadr-multiple-standby-databases>
- 未サポートの構成 (Db2 v11.5.6 現在)
 - DPF (Db2 Database Partitioning Feature)
 - pureScale構成
 - 共有ディスク構成

1.5 db2cm – Db2クラスター・マネージャ・ユーティリティ

db2cm – Db2 クラスター・マネージャ・ユーティリティ

<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-db2cm-db2-cluster-manager-utility>

db2cm

```
[ -create ] [ -delete ] [ -disable ] [ -enable ]  
[ -list ] [ -export ] [ -import ] [ -copy_resources ] [ -dump ] [ -help ]
```

– create/delete

Pacemaker ドメインおよびドメイン内のリソースを作成・削除する

– enable/disable

Pacemaker ドメイン内のすべてのPacemakerリソースの自動化を使用可能・または使用不可能にする

– list

クラスター構成とステータス情報を表示

– export/import

ファイルにクラスター構成のバックアップ取得または取得したバックアップからリストアする

– dump

ローカル・ホストに関連したクラスター情報を、ローカル・ディレクトリーの圧縮ファイルにエクスポートする

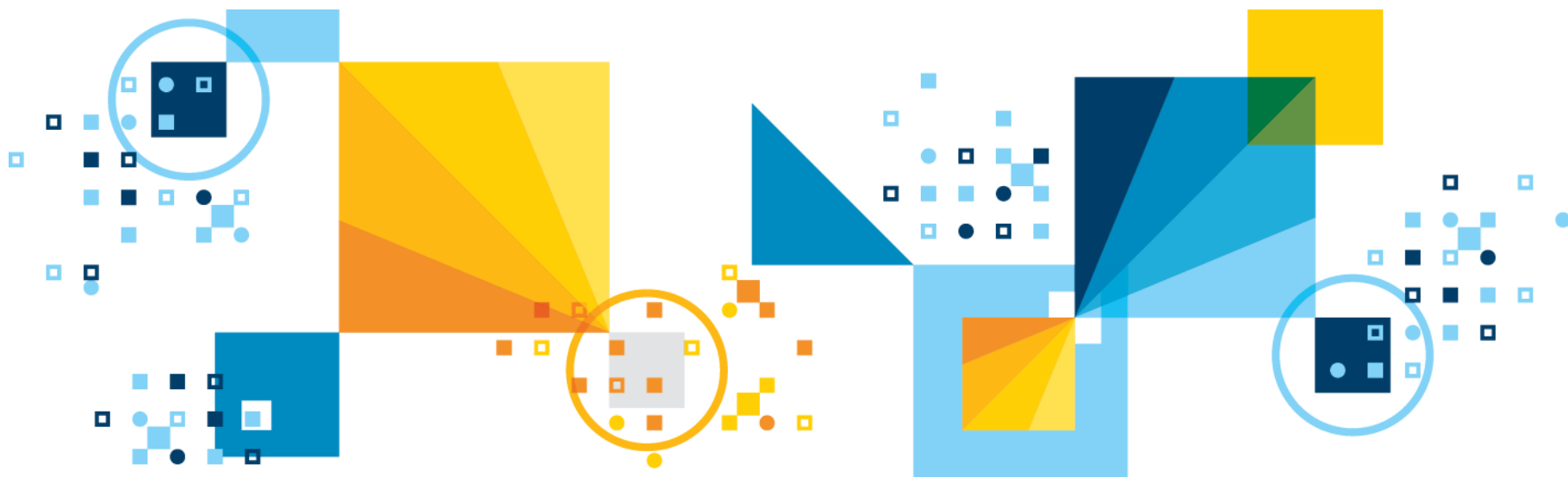
– copy_resources

指定されたパスにあるリソース・エージェント・スクリプトを、指定されたホストのリソース・パスにコピーする

– help

・ db2cmの使用方法についての情報出力する

2. 本ガイドの前提



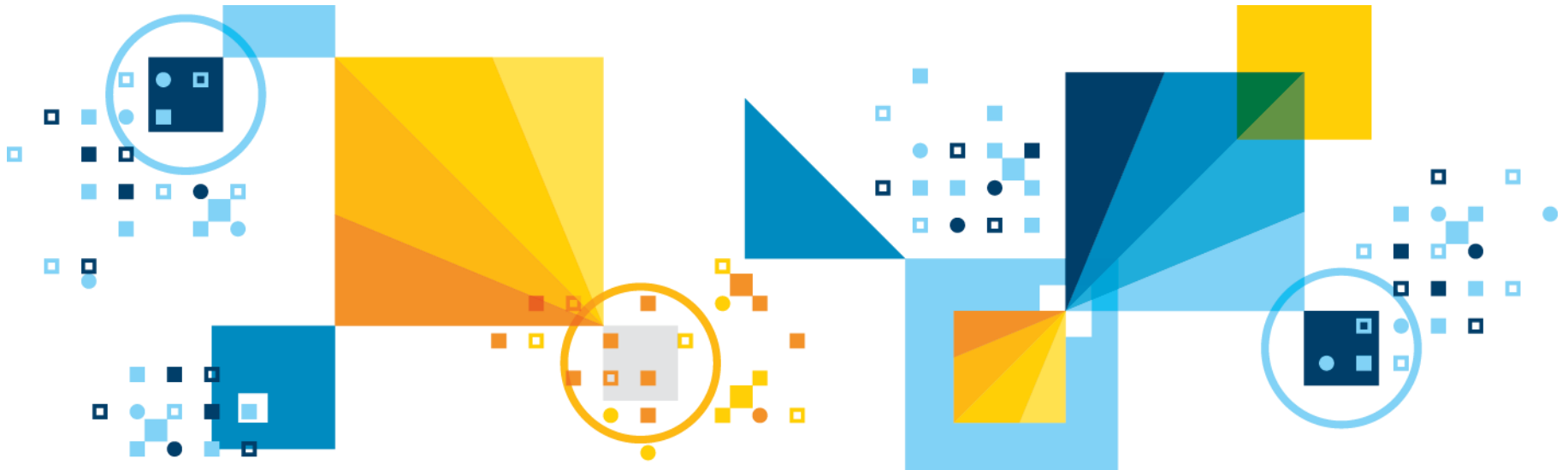
2. 本ガイドの前提

- 本ガイドでは、以下環境での構築手順を記述しています。

<環境情報>

- OS: Red Hat Enterprise Linux Server release 8.2 (Ootpa)
- Cluster: Pacemaker 2.0.4 + Corosync 3.0.4 (Db2 v11.5.6 導入イメージに含まれる)
- Db2バージョン: Db2 LUW 11.5.6
- Db2インスタンスユーザー: db2inst1
- Db2インスタンスディレクトリー: /home/db2inst1
- 各ノードホスト名
 - Primaryノードホスト名: cm1-rhel8
 - Standbyノードホスト名: cm2-rhel8
 - クォーラムデバイスノードホスト名: cm3-rhel8
- DB名: MYDB

3. Pacemaker導入・構成



3.1 Db2導入時の注意点

- Db2 v11.5.6以降、Pacemakerクラスター・マネージャーがパッケージ化され、Db2と共にインストール可能です。
 - Db2 v11.5.5以前は、Db2モジュール自体にはPacemakerクラスター・ソフトウェア・パッケージは含まれていないため、Pacemakerクラスター・ソフトウェア・パッケージを別途ダウンロード・インストールする必要があります。Db2 v11.5.5以前のPacemakerのインストール手順については、後述の「7. 参考:Pacemaker導入(Db2 V11.5.5以前)」をご参照下さい。
- OSと製品バージョンの組み合わせがサポートされていることをリンク先よりご確認ください。
 - Software Product Compatibility Reports
<http://www-969.ibm.com/software/reports/compatibility/clarity/index.html>
 - System requirements for IBM Db2 for Linux, UNIX, and Window
<https://www.ibm.com/support/pages/system-requirements-ibm-db2-linux-unix-and-windows>
- Db2のインストール手順詳細は以下オンラインマニュアルの関連リンクを参照ください。
 - Db2 LUW 11.5 データベース製品のインストール要件(v11.5版)
<https://www.ibm.com/docs/ja/db2/11.5?topic=installing-requirements-db2-products>
 - Db2 データベース製品のインストール要件(v11.5版)
<https://www.ibm.com/docs/ja/db2/11.5?topic=installing-requirements-db2-products>
 - Db2 データベース・サーバーのインストール(v11.5版)
<https://www.ibm.com/docs/ja/db2/11.5?topic=installing-db2-database-servers>

3.2 Pacemaker導入手順



3.2.1 Pacemakerを使用した統合ソリューションの前提条件・前提作業(1/2)

■ 前提条件

- 各ノードにてDb2インスタンスのユーザーIDとグループIDがセットアップされていることを確認
- 各ノードからのTCP/IP接続確認
 - ・ 各HADRノード間でTCP/IP接続が可能なことを確認
 - ・ クォーラムデバイスノードと各HADRノード間においてTCP/IP接続が可能なことを確認
- 全てのHADRペアがプライマリ/スタンバイのロールで起動し、ピア状態であることを確認
- 必要なストレージ容量が満たされていること
 - ・ 容量に関しては以下を参照
 - ・ Pacemaker を使用した統合ソリューションの前提条件
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-prerequisites-integrated-solution-using>
- “db2prereqcheck -l” にて出力されるパッケージに加えて、前提パッケージとして、ksh、およびRHELの場合にはpython3-dnf-plugin-versionlockが導入されていること
- VIPを使用する場合は、プライマリ・スタンバイのIP、およびVIPのサブネットが同一であることを確認
 - ・ Pacemaker (Linux)
仮想 IP アドレスに関する DB2 データベースの要件
<https://www.ibm.com/docs/ja/db2/11.5?topic=software-pacemaker-linux>

3.2.1 Pacemakerを使用した統合ソリューションの前提条件・前提作業(2/2)

■ 前提作業

- /etc/hostsのセットアップ
 - ・ 各ノードから、ローカルおよびリモートのノードに、そのノードの正式ホスト名および短縮ホスト名を設定
- HADRノード間でのSSH接続確認
 - ・ rootユーザーとインスタンス・ユーザーの両方が、HADRノード間で長いホスト名と短縮ホスト名の両方を使用してSSHを使用可能にし、事前に接続できることを確認
 - ・ rootユーザーとインスタンス・ユーザーの両方が、HADRノードの間でパスワードレスSSHを使用可能にし、事前に接続できることを確認
- PacemakerおよびCorosync用のポートの登録と開放
 - ・ ファイアウォールの設定で、以下リンク記載のポートを開ける
 - ・ Pacemaker を使用した統合ソリューションの前提条件
Pacemaker および Corosync のポートの使用情報
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-prerequisites-integrated-solution-using>
 - ・ High Availability Add-Onのポートの有効化を行うことで、Pacemakerに必要なポートをまとめて解放することが可能
 - ・ 4.7. High Availability Add-On のポートの有効化
https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/8/html/configuring_and_managing_high_availability_clusters/proc_enabling_ports_for_high_availability-creating_high_availability-cluster
 - ・ なお、上記内容に含まれるpcsdが使用するTCP 2224は、db2cmユーティリティーを使用したPacemaker構成・運用では使用しない

3.2.2 Pacemakerを使用した統合ソリューションの制約事項

■ 制約事項

- Db2障害モニターは使用不可
- db2cm構成では、SYNC同期モードおよびNEARSYNC同期モードのみサポート
- DB構成パラメーターHADR_PEER_WINDOWを120秒以上に設定する必要がある
 - ・ HADR_PEER_WINDOWは、プライマリー・データベースにおいてスタンバイ・データベースとの接続が失われた場合に、HADRのプライマリーとスタンバイのデータベース・ペアをピア状態とみなして動作を継続する期間を表します。
 - ・ HADR_PEER_WINDOWは、同期モードSYNC及びNEARSYNC時のみ設定可能
 - ・ スタンバイ・データベース障害やネットワーク障害によりスタンバイ・データベースへの接続が失われると、アプリケーションの実行は、例えばHADR_PEER_WINDOWを120秒に設定した場合には、最大HADR_TIMEOUTの秒数+120秒ウェイトします。
 - ・ hadr_peer_window – HADR ピア・ウィンドウ構成パラメーター
<https://www.ibm.com/docs/ja/db2/11.5?topic=parameters-hadr-peer-window-hadr-peer-window>
- 2021/10現在、複数HADRペアが存在する場合、それらを一緒に引き継ぐ設定はありません。もしプライマリーをどちらか一方に集める要件があるのであれば、単一HADRペア構成とするなどの検討を行ってください。
- 最新情報については、db2cmユーティリティを実行するバージョンのDb2 LUWのオンラインマニュアルをご確認ください。
 - ・ Pacemaker を使用した統合ソリューションの前提条件
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-prerequisites-integrated-solution-using>

3.2.3 Pacemaker導入手順(1/2)

■ 導入手順

- Db2 インストーラーを使用した Pacemaker のインストール
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-installing-using-db2-installer>
- 2021/10現在、db2setup(応答ファイル)とGUIによるインストールは、まだサポートされていません。

※ 以下はDb2 v11.5.6での導入手順になります。Db2 v11.5.5以前のインストール手順については、後述の「7. 参考: Pacemaker導入(Db2 V11.5.5以前)」をご参照下さい。

1. 製品モジュールの解凍 * 以下1~2の手順は、HADR両ノードで実行する
 - Db2モジュールを任意のディレクトリーに配置し解凍(以下の例では/workに配置)
 - 解凍後、Db2イメージ・ディレクトリー(以下の例ではserver_decディレクトリー)が作成される

```
# cd /work
# tar -xvf v11.5.6_linuxx64_server_dec.tar.gz
# ls -ld server_dec
drwxr-xr-x 3 root root 4096 Jun 12 02:50 server_dec
```

3.2.3 Pacemaker導入手順(2/2)

■ 導入手順(続き)

2. 製品インストール

- Db2イメージ・ディレクトリーに移動
- db2_installコマンドを使用してDb2およびPacemakerをインストール
- db2_installコマンドによるインストール時には、デフォルトでPacemakerがインストールされる
 - ・ ただし、RHELの場合には、前提としてpython3-dnf-plugin-versionlockが必要
 - ・ Pacemaker がインストールされないようにするには、`-f NOPCMK` オプションを実行

```
# cd /work/server_dec/  
# ./db2_install -y -f NOTSAMP -b <install_path> -p SERVER
```

3. 製品インストールの確認

- 以下コマンドで、Pacemakerがインストールされたことを確認可能

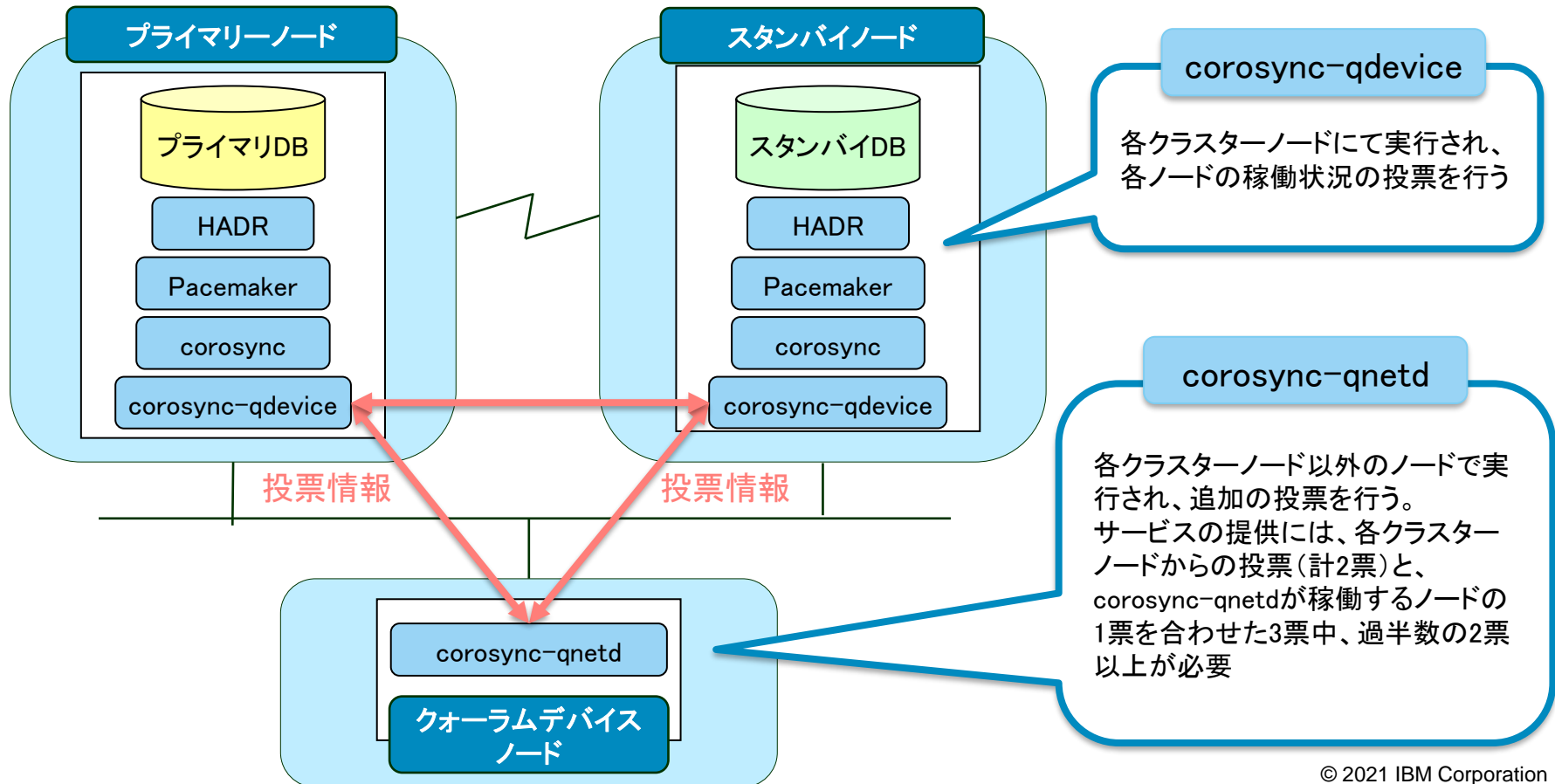
```
# rpm -q pacemaker  
pacemaker-2.0.4-2.db2pcmk.e18.x86_64
```

3.3 QDeviceクォーラムのインストール



3.3.1 QDeviceクォーラムのインストール(概要)(1/2)

- Pacemakerを使用したHADR構成では、スプリットブレインを防止するために、プライマリノード、スタンバイノードの他に第3のノードとしてクォーラムデバイスノードを用意する構成とすることが推奨される
- ただしAWS上で構成を行う場合には2ノード構成をとることも可能(詳細は後述)



3.3.1 QDeviceクォーラムのインストール(概要)(2/2)

■ クォーラム概要

- クラスターノードの過半数がオンラインである場合、クラスター内でクォーラムが確立されません。一方で過半数がオフラインとなり、クラスター内でクォーラムが確立されない場合は、Pacemakerはすべてのリソースを停止します
- クォーラムは投票システムを使用して確立されます。クラスターノードが正常に機能しない場合や他のノードとの通信が切断された場合は、正常に稼働している過半数のノードから分離されます
- このクォーラム機能により、スプリットブレイン(クラスター間通信が遮断され、各ノードがプライマリーDBとして動作してしまうこと)を回避可能

■ クォーラムデバイスノードの構成に関して

- クォーラムデバイスノードにはDb2およびPacemakerのインストールは不要
- corosync-qnetdパッケージのみインストールを行う
- クォーラムデバイスノードは低スペックなサーバーで問題ない
- プライマリー・スタンバイ両ノードからTCP/IP通信が可能な状態にする
- クォーラムデバイスノードは、複数クラスターで共有可能
- 詳細は以下のリンクを参照

- ・ Pacemakerのクォーラムデバイスのサポート

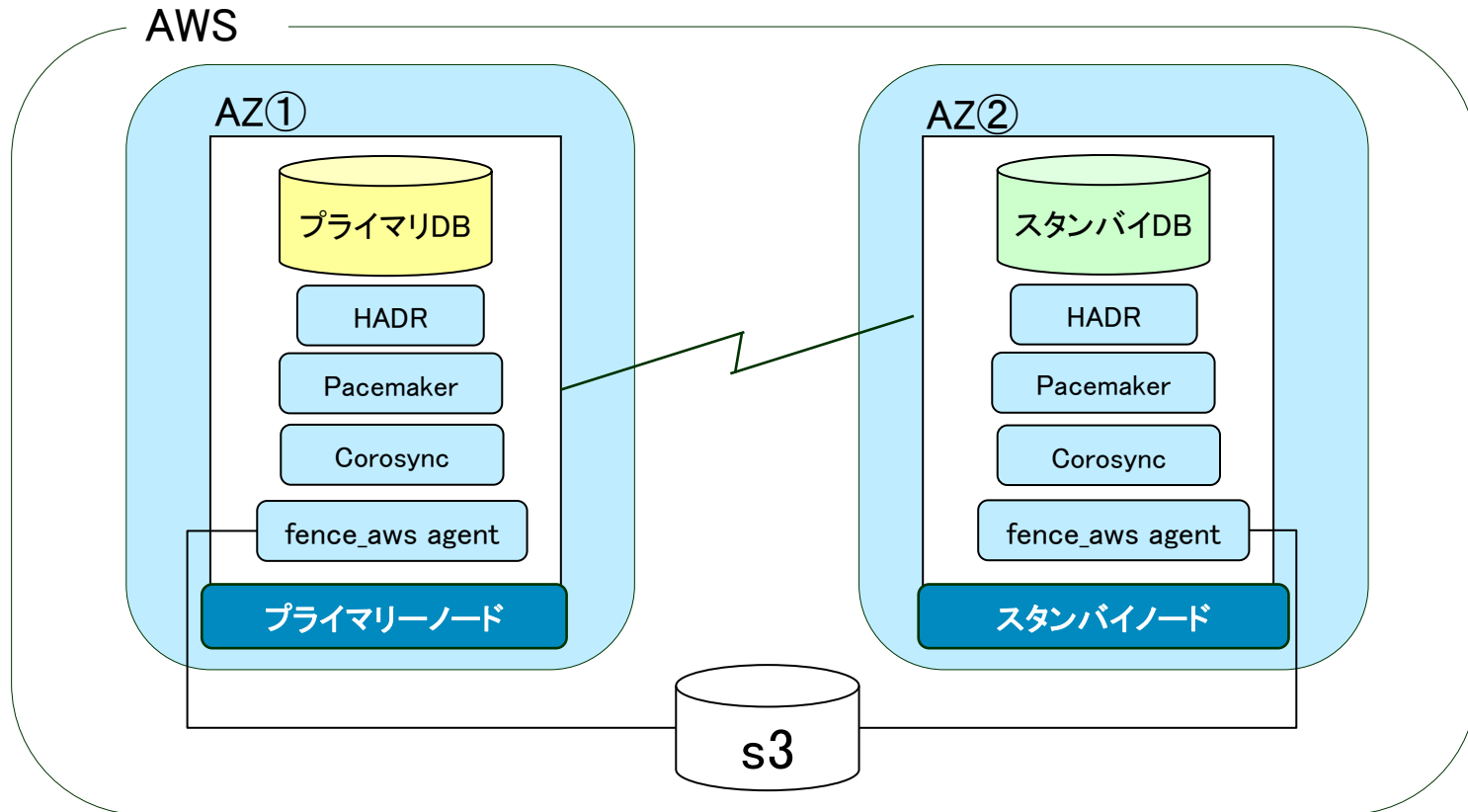
<https://www.ibm.com/docs/ja/db2/11.5?topic=component-quorum-devices-support-pacemaker>

- ・ QDevice クォーラムのインストールと構成

<https://www.ibm.com/docs/ja/db2/11.5?topic=utility-install-configure-qdevice-quorum>

参考情報) AWSサービスS3を使用した構成(1/2)

- AWS上にPacemakerを使用してHADR構成する場合には、クォーラムデバイスノードを構成する
代わりにS3ストレージへの接続によるfencing機能を使用した構成を選択することも可能



参考情報) AWSサービスS3を使用した構成(2/2)

■ メリット

- 必要ノード数が2ノードとなるため、クォーラムデバイスノードを構成する方法と比較してコストが抑えられる

■ デメリット

- プライマリーノード障害時の引き継ぎ時間は、クォーラムデバイスノードを構成する方法と比較して非常に遅い
- 引き継ぎ時のデータ整合性を保証する為のHADR_PEER_WINDOWを少なくとも300秒以上に設定する必要がある。(環境によってはより長くする必要が生じる可能性あり)
 - スタンバイノード障害などの際に300秒(以上) + HADR_TIMEOUTの時間分、プライマリーDBへの更新が抑止される
- 2020/11に提供された11.5.5の新機能であり、設定方法等の情報は以下のみ(2021/10現在)

■ 参考リンク

- Alternate or additional configurations available on Amazon Web Services (AWS)
<https://www.ibm.com/support/pages/node/6327425>
- Setting up two-node Db2 HADR Pacemaker cluster with fencing on AWS
<https://www.ibm.com/support/pages/node/6359159>
- How to set up a Db2 HADR Pacemaker cluster with Overlay IP as a Virtual IP on AWS
<https://www.ibm.com/support/pages/node/6359155>

3.3.2 QDeviceクォーラムのインストール

■ 導入手順

– QDevice クォーラムのインストールと構成

<https://www.ibm.com/docs/ja/db2/11.5?topic=utility-install-configure-qdevice-quorum>

※ 以下はDb2 v11.5.6での導入手順になります。Db2 v11.5.5以前のインストール手順については、後述の「7. 参考:Pacemaker 導入(Db2 V11.5.5以前)」をご参照下さい。

1. クォーラムデバイスノードにて製品モジュールの解凍

- Db2モジュールを任意のディレクトリーに配置し解凍(以下の例では/workに配置)
- 解凍後、Db2イメージ・ディレクトリー(以下の例ではserver_decディレクトリー)が作成される

```
# cd /work
# tar -xvf v11.5.6_linuxx64_server_dec.tar.gz
# ls -ld server_dec
drwxr-xr-x 3 root root 4096 Jun 12 02:50 server_dec
```

2. クォーラムデバイスノードにてcorosync-qnetdパッケージをインストール(rootで実行)

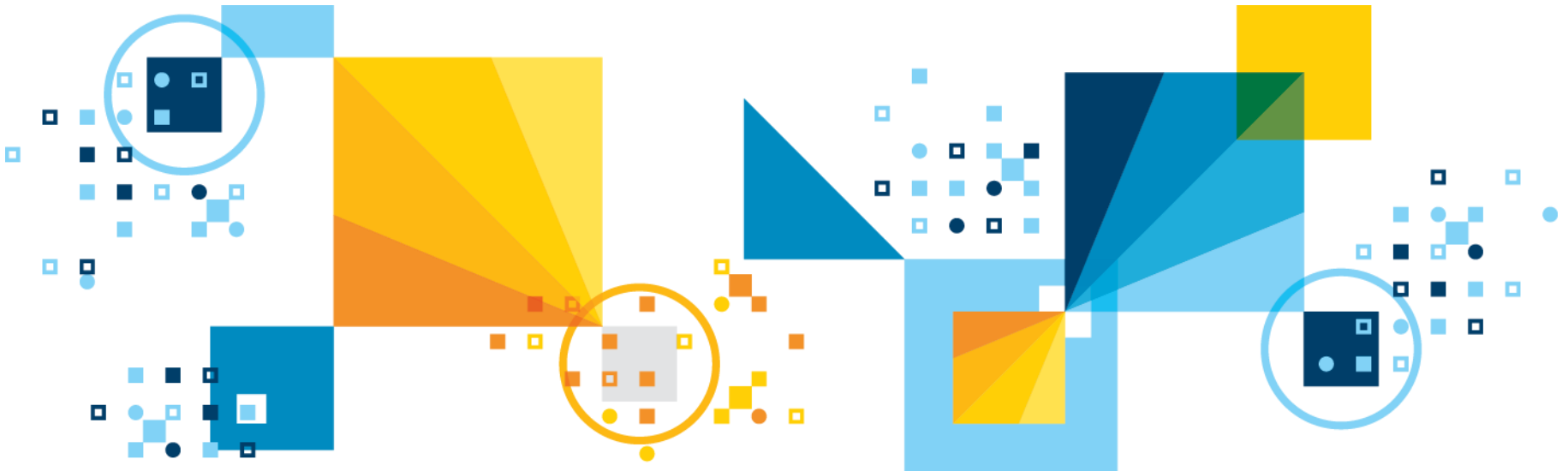
```
# dnf install ¥
/work/server_dec/db2/linuxamd64/pcmk/Linux/rhel/x86_64/corosync-qnetd*
```

3. クォーラムデバイスノードにて製品インストールの確認

- 以下コマンドで、corosync-qnetdがインストールされたことを確認可能

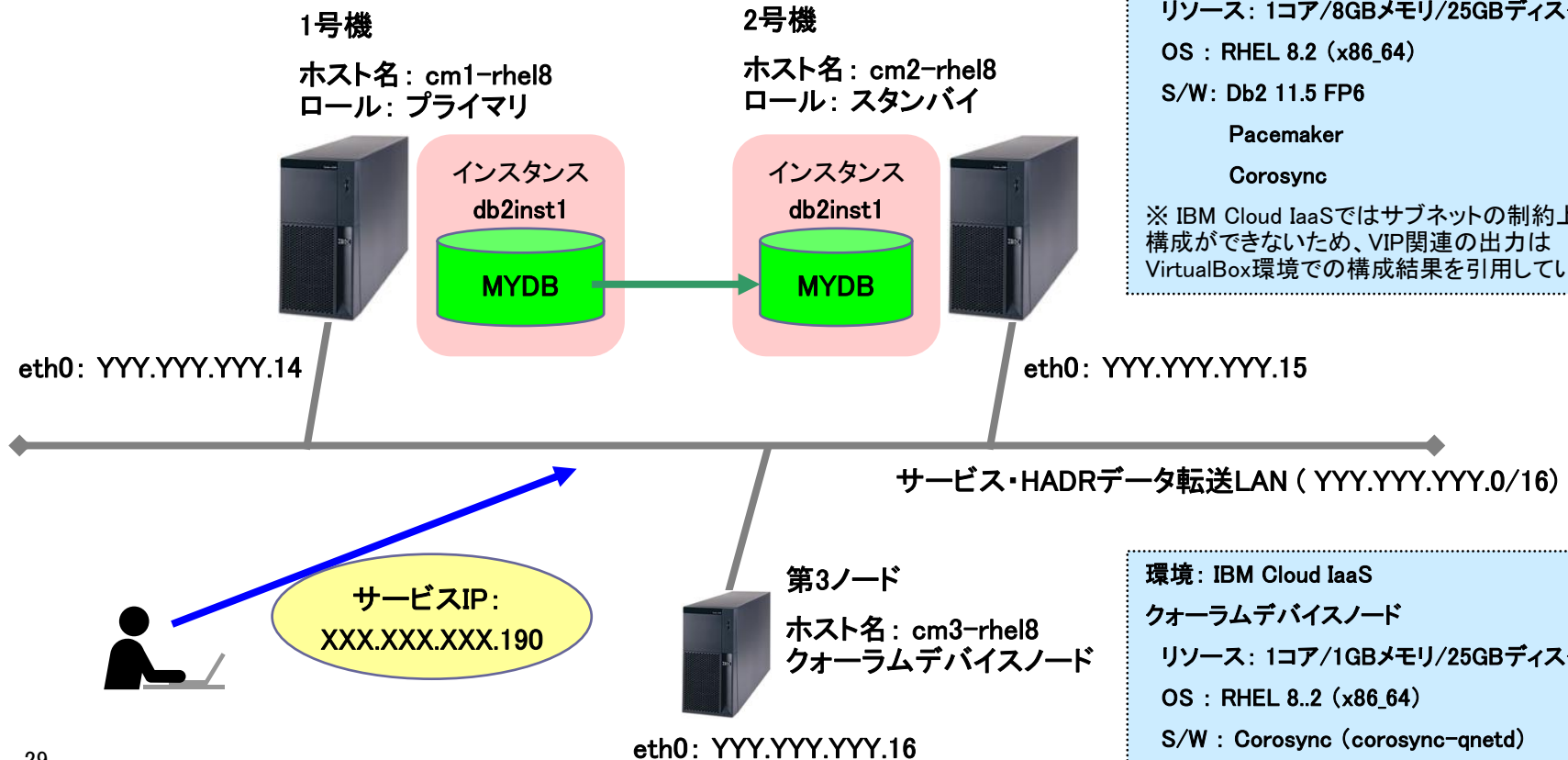
```
# rpm -q corosync-qnetd
corosync-qnetd-3.0.0-4.db2pcmk.el8.x86_64
```

4. db2cmコマンドによるPacemaker構成



4.1 HADR構成の環境構成図

- 本ガイドは、以下のテスト環境にて検証を実施しています。
実際に適用する環境に合わせてカスタマイズを実施してください。
- Cloud環境に構築する場合は、サービスIPを使用した構成ではなくクライアントリルートを使用した構成にすることをご検討ください。



環境: IBM Cloud IaaS

プライマリ/スタンバイノード

リソース: 1コア/8GBメモリ/25GBディスク

OS : RHEL 8.2 (x86_64)

S/W: Db2 11.5 FP6

Pacemaker

Corosync

※ IBM Cloud IaaSではサブネットの制約上VIPの構成ができないため、VIP関連の出力はVirtualBox環境での構成結果を引用しています。

環境: IBM Cloud IaaS

クォーラムデバイスノード

リソース: 1コア/1GBメモリ/25GBディスク

OS : RHEL 8.2 (x86_64)

S/W : Corosync (corosync-qnetd)

4.2 Db2/Pacemakerパラメータ抜粋

カテゴリ	パラメータ項目	設定値	備考
OS	Db2オーナー	db2inst1	
	Db2グループ	db2iadm1	
Db2	インスタンス名	db2inst1	
	Db2インスタンスホームディレクトリ	/db2/db2inst1	
	アーカイブログディレクトリ	/db2/arclog	
	データベース名	MYDB	
	HADRポート(プライマリ)	50010/tcp	※1
	HADRポート(スタンバイ)	50020/tcp	※1
	HADR_SYNCMODE	nearsync	同期モードはSYNCまたはNEARSYNCから選択 HADR_PEER_WINDOWは120秒以上の設定が必須
	HADR_TIMEOUT	30	
	HADR_PEER_WINDOW	150	
	クラスタノード(アクティブ)	cm1-rhel8/cm1-rhel8.localdomain	
	クラスタノード(スタンバイ)	cm2-rhel8/cm2-rhel8.localdomain	
	クォーラムデバイスノード	cm3-rhel8/cm3-rhel8.localdomain	

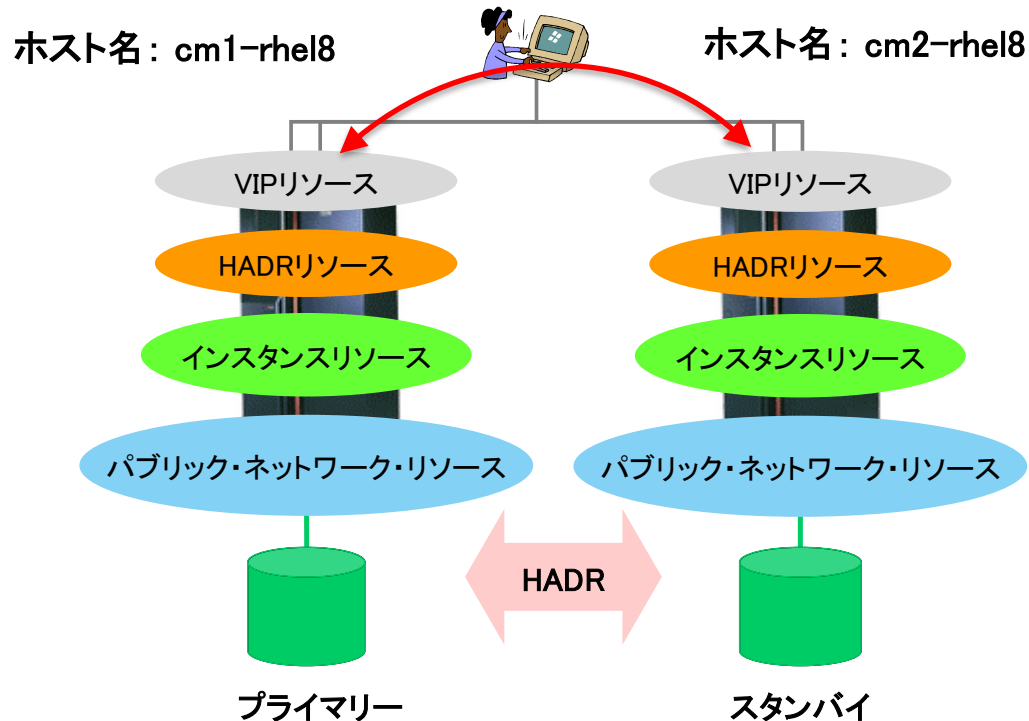
※1 HADR用通信ポートの登録と開放

/etc/servicesにHADR用通信ポートを定義し、ファイアウォールの設定でポートを開ける

DB2_HADRP	50010/tcp
DB2_HADRS	50020/tcp

4.3 Pacemakerのリソース定義概要

- HADRを構成した2ノードで、db2cmコマンドを使用してクラスター・ドメインを作成
- クラスタードメイン内に作成するリソース・グループ
 - パブリック・ネットワーク・リソース: ネットワーク・インターフェース監視 db2ethmon
 - インスタンスリソース: DBインスタンス制御 db2inst
 - HADRリソース: HADR制御 db2hadr (Master/Slave)
 - (オプション) サービスアドレス制御 IPAddr2



4.4 db2cmコマンドによるPacemaker構成(1/12)

- 以降の手順は、プライマリーノード、またはスタンバイノードいずれかのノードにてrootにて実行する
(以下の例ではプライマリーノードにて実行)

1. Pacemakerクラスターとパブリック・ネットワーク・リソースの作成

- 以下の例ではドメイン名として「db2domain」指定
- hostオプションにてプライマリーノード、スタンバイノードのホスト名を指定
- publicEthernetオプションにてネットワークデバイス名「eth0」を指定

```
[root@cm1-rhel8 ~]# cd /opt/ibm/db2/V11.5/bin

[root@cm1-rhel8 bin]# ./db2cm -cluster -list
Cluster Status

There is no cluster on this host.
[root@cm1-rhel8 bin]# ./db2cm -cluster -create -domain db2domain -host cm1-rhel8 -
publicEthernet eth0 -host cm2-rhel8 -publicEthernet eth0
Created db2_cm1-rhel8_eth0 resource.
Created db2_cm2-rhel8_eth0 resource.
Cluster created successfully.
```


4.4 db2cmコマンドによるPacemaker構成(2/12)

1. Pacemakerクラスターとパブリック・ネットワーク・リソースの作成(続き)

- Pacemakerクラスターとパブリック・ネットワーク・リソースが作成されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
Cluster Status

Domain information:
Domain name           = db2domain
Pacemaker version    = 2.0.4-2.db2pcmk.e18
Corosync version      = 3.0.4-dirty
Current domain leader = cm1-rhel8
Number of nodes       = 2
Number of resources   = 2

Node information:
Name name             State
-----
cm1-rhel8             Online
cm2-rhel8             Online
```

指定したドメイン名で
ドメインが作成され、
両ノードのStateが「Online」に
なることを確認

(続き)
Resource Information:

```
Resource Name           = db2_cm1-rhel8_eth0
State                   = Online
Managed                = true
Resource Type           = Network Interface
Node                    = cm1-rhel8
Interface Name          = eth0

Resource Name           = db2_cm2-rhel8_eth0
State                   = Online
Managed                = true
Resource Type           = Network Interface
Node                    = cm2-rhel8
Interface Name          = eth0
```

```
Fencing Information:
Not configured
Quorum Information:
Two-node quorum
```

両ノードのリソースStateが
「Online」になることを確認

4.4 db2cmコマンドによるPacemaker構成(3/12)

2. インスタンスリソースの作成

- Instanceオプションにインスタンス名を指定(以下の例ではdb2inst1)
- hostオプションにてホスト名を指定する
- インスタンスリソース作成コマンドはプライマリー・スタンバイノード用それぞれ実行(ただし作成コマンドは同一ノードから行う)

```
[root@cm1-rhel8 bin]# ./db2cm -create -instance db2inst1 -host cm1-rhel8  
Created db2_cm1-rhel8_db2inst1_0 resource.  
Instance resource for db2inst1 on cm1-rhel8 created successfully.
```

```
[root@cm1-rhel8 bin]# ./db2cm -create -instance db2inst1 -host cm2-rhel8  
Created db2_cm2-rhel8_db2inst1_0 resource.  
Instance resource for db2inst1 on cm2-rhel8 created successfully.
```

4.4 db2cmコマンドによるPacemaker構成(4/12)

2. インスタンスリソースの作成(続き)

– インスタンスリソースが作成されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
Cluster Status

Domain information:
Domain name           = db2domain
Pacemaker version    = 2.0.4-2.db2pcmk.e18
Corosync version     = 3.0.4-dirty
Current domain leader = cm1-rhel8
Number of nodes      = 2
Number of resources  = 4

Node information:
Name name            State
-----
cm1-rhel8            Online
cm2-rhel8            Online

Resource Information:

Resource Name        = db2_cm1-rhel8_db2inst1_0
State                = Online
Managed             = true
Resource Type        = Instance
Node                 = cm1-rhel8
Instance Name        = db2inst1
```

(続き)

```
Resource Name        = db2_cm1-rhel8_eth0
State                = Online
Managed             = true
Resource Type        = Network Interface
Node                 = cm1-rhel8
Interface Name       = eth0

Resource Name        = db2_cm2-rhel8_db2inst1_0
State                = Online
Managed             = true
Resource Type        = Instance
Node                 = cm2-rhel8
Instance Name        = db2inst1

Resource Name        = db2_cm2-rhel8_eth0
State                = Online
Managed             = true
Resource Type        = Network Interface
Node                 = cm2-rhel8
Interface Name       = eth0

Fencing Information:
Not configured
Quorum Information:
Two-node quorum
```

両ノードのインスタンスリソースの
Stateが「Online」になることを確認

4.4 db2cmコマンドによるPacemaker構成(5/12)

3. HADRリソースの作成

- dbオプションにデータベース名を指定(以下の例ではMYDB)
- Instanceオプションにてインスタンス名を指定(以下の例ではdb2inst1)

```
[root@cm1-rhel8 bin]# ./db2cm -create -db MYDB -instance db2inst1
Database resource for MYDB created successfully.
```

- 作成されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
```

(中略)

```
Resource Name           = db2_db2inst1_db2inst1_MYDB
Resource Type           = HADR
DB Name                  = MYDB
Managed                 = true
HADR Primary Instance   = db2inst1
HADR Primary Node       = cm1-rhel8
HADR Primary State      = Online
HADR Standby Instance   = db2inst1
HADR Standby Node       = cm2-rhel8
HADR Standby State      = Online
```

```
Fencing Information:
  Not configured
Quorum Information:
  Two-node quorum
```

HADRリソースが作成され
Stateが「Online」になることを確認

4.4 db2cmコマンドによるPacemaker構成(6/12)

3. HADRリソースの作成(続き)

- crm statusコマンドにてクラスタの確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.e18-2deceaa3ae) - partition with quorum
* Last updated: Fri Oct 22 15:21:55 2021
* Last change: Fri Oct 22 15:19:33 2021 by root via cibadmin on cm1-rhel8
* 2 nodes configured
* 6 resource instances configured

Node List:
* Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
* Masters: [ cm1-rhel8 ]
* Slaves: [ cm2-rhel8 ]
```

作成したパブリック・ネットワーク・リソース、
インスタンスリソース、HADRリソースが表示され、
「Started」となることを確認

Masters(プライマリロール)はcm1-rhel8
Slaves(スタンバイロール)はcm2-rhel8

4.4 db2cmコマンドによるPacemaker構成(7/12)

4. (オプション)VIP リソースの作成

- primaryVIP/standbyVIPオプションにてサービスIPアドレスを指定

```
[root@cm1-rhel8 bin]# ./db2cm -create -primaryVIP 192.168.10.17 -db MYDB -instance db2inst1
Primary VIP resource created successfully.
```

```
[root@cm1-rhel8 bin]# ./db2cm -create -standbyVIP 192.168.10.18 -db MYDB -instance db2inst1
Standby VIP resource created successfully.
```

- 作成されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
```

(中略)

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-primary-VIP
State              = Online
Managed           = true
Resource Type      = IP
Node               = cm1-rhel8
Ip Address         = 192.168.10.17
```

VIPリソースが作成され、
Stateが「Online」になることを確認

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-standby-VIP
State              = Online
Managed           = true
Resource Type      = IP
Node               = cm2-rhel8
Ip Address         = 192.168.10.18
```

4.4 db2cmコマンドによるPacemaker構成(8/12)

4. (オプション)VIP リソースの作成(続き)

- 再度crm statusコマンドにてクラスターの確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.e18-2deceaa3ae) - partition with quorum
 * Last updated: Fri Oct 22 15:35:37 2021
 * Last change: Fri Oct 22 15:35:06 2021 by root via cibadmin on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

作成したVIPリソースが表示され、「Started」となることを確認

4.4 db2cmコマンドによるPacemaker構成(9/12)

5. クォーラムのセットアップ

- プライマリー・スタンバイノードいずれかのノードにてrootにて実行する
(以下の例ではプライマリーノードにて実行している)
- qdeviceオプションにてクォーラムデバイスノードのホスト名を指定(以下の例ではcm3-rhel8)

```
[root@cm1-rhel8 bin]# ./db2cm -create -qdevice cm3-rhel8
Successfully configured qdevice on nodes cm1-rhel8 and cm2-rhel8
Attempting to start qdevice on cm3-rhel8
Quorum device cm3-rhel8 added successfully.
```


4.4 db2cmコマンドによるPacemaker構成(10/12)

5. クォーラムのセットアップ(続き)

- プライマリー・スタンバイノードにてcorosyncコマンドを実行して、クォーラムが正しくセットアップされたことを確認

(以下の例はプライマリーノードの実行結果)

```
[root@cm1-rhel8 bin]# corosync-qdevice-tool -s
Qdevice information
-----
Model:                               Net
Node ID:                              1
Configured node list:
  0   Node ID = 1
  1   Node ID = 2
Membership node list: 1, 2

Qdevice-net information
-----
Cluster name:          db2domain
QNetd host:           cm3-rhel8:5403
Algorithm:             LMS
Tie-breaker:          Node with lowest node ID
State:                 Connected
```

ドメイン名、クォーラムノード名が
正しく設定され、Stateが「Connected」
であることを確認

4.4 db2cmコマンドによるPacemaker構成(11/12)

5. クォーラムのセットアップ(続き)

- クォーラムデバイスノードにてcorosyncコマンドを実行して、クォーラム・デバイスが正常に稼働していることを確認

```
[root@cm3-rhel8 ~]# corosync-qnetd-tool -l
Cluster "db2domain":
  Algorithm:          LMS
  Tie-breaker:       Node with lowest node ID
  Node ID 1:
    Client address:   ::ffff:192.168.10.14:54492
    Configured node list: 1, 2
    Membership node list: 1, 2
    Vote:             ACK (ACK)
  Node ID 2:
    Client address:   ::ffff:192.168.10.15:45656
    Configured node list: 1, 2
    Membership node list: 1, 2
    Vote:             ACK (ACK)
```

4.4 db2cmコマンドによるPacemaker構成(12/12)

5. クォーラムのセットアップ(続き)

- クォーラムのセットアップ構成後のdb2cm -listコマンド出力結果

```
[root@cm1-rhel8 bin]# ./db2cm -list
```

(中略)

```
Fencing Information:
  Not configured
```

```
Quorum Information:
  Qdevice
```

```
Qdevice information
-----
```

```
Model:           Net
Node ID:         1
```

```
Configured node list:
  0 Node ID = 1
  1 Node ID = 2
```

```
Membership node list: 1, 2
```

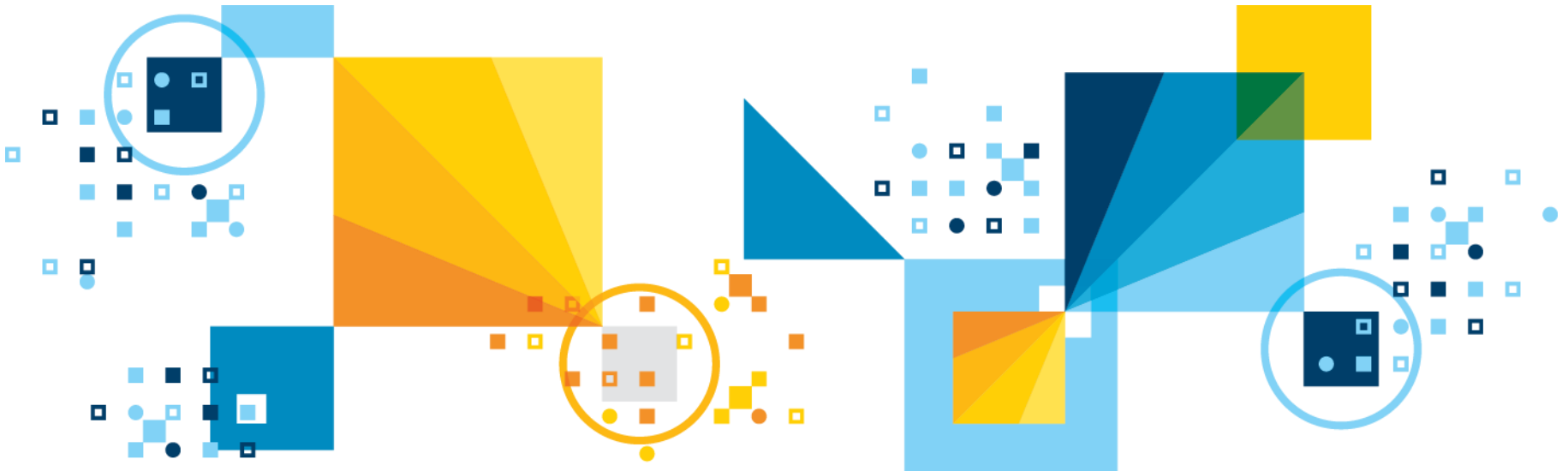
```
Qdevice-net information
-----
```

```
Cluster name:      db2domain
QNetd host:        cm3-rhel8:5403
Algorithm:         LMS
Tie-breaker:       Node with lowest node ID
State:             Connected
```

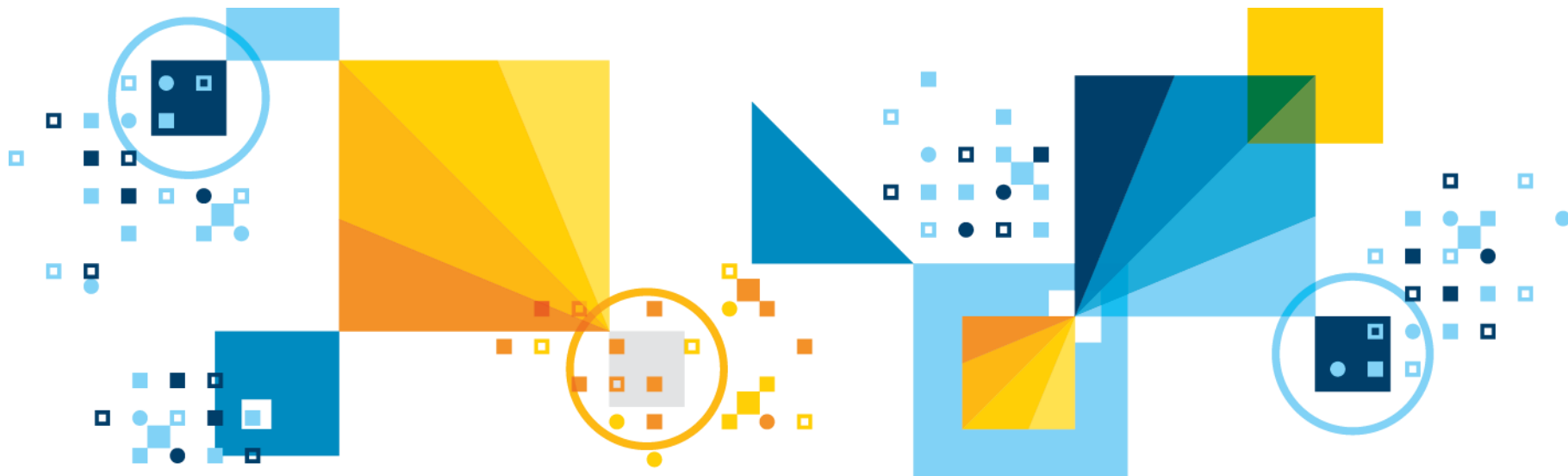
Quorum Informationが
「Two-node quorum」から「Qdevice」
となることを確認

Qnetd hostとしてクォーラムデバイスノードの
ホスト名が指定され、Stateが「Connected」と
なることを確認

5. PacemakerによるHADR運用



5.1 正常運用



5.1.1 正常起動(1/3)

- Pacemakerドメイン内のすべてリソースに対する自動制御を、db2cm -enableコマンドにて有効化

1. 現在のリソースの制御状況を確認

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Resource Information:

```
Resource Name      = db2_cm1-rhel8_db2inst1_0
State              = Online
Managed          = false
Resource Type     = Instance
Node              = cm1-rhel8
Instance Name     = db2inst1
```

```
Resource Name      = db2_cm1-rhel8_eth0
State              = Online
Managed          = false
Resource Type     = Network Interface
Node              = cm1-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_cm2-rhel8_db2inst1_0
State              = Online
Managed          = false
Resource Type     = Instance
Node              = cm2-rhel8
Instance Name     = db2inst1
```

全てのリソースのManagedのステータスが「false」であることを確認

(続き)

```
Resource Name      = db2_cm2-rhel8_eth0
State              = Online
Managed          = false
Resource Type     = Network Interface
Node              = cm2-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB
Resource Type     = HADR
DB Name          = MYDB
Managed          = false
HADR Primary Instance = db2inst1
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst1
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-primary-VIP
State              = Online
Managed          = false
Resource Type     = IP
Node              = cm1-rhel8
Ip Address        = 192.168.10.17
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-standby-VIP
State              = Online
Managed          = false
Resource Type     = IP
Node              = cm2-rhel8
Ip Address        = 192.168.10.18
```

(後略)

5.1.1 正常起動(2/3)

2. プライマリー・スタンバイいずれかのノードでdb2cm -enableコマンドを実行し、Pacemakerによるリソースの自動制御を有効化
 - v11.5.5以前ではdb2cm -enableのみで実行可能でしたが、v11.5.6では、“-all”、または“instance *instance_name* -host *host-name-of-the-instance*”オプションが必要です

```
[root@cm1-rhel8 bin]# ./db2cm -enable -all
Automation for Db2 resources are enabled.
```

- db2cm - Db2 クラスタ・マネージャ・ユーティリティ
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-db2cm-db2-cluster-manager-utility>

5.1.1 正常起動(3/3)

3. Pacemakerによるリソースの自動制御が有効化されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Resource Information:

```
Resource Name      = db2_cm1-rhel8_db2inst1_0
State              = Online
Managed          = true
Resource Type     = Instance
Node              = cm1-rhel8
Instance Name    = db2inst1
```

```
Resource Name      = db2_cm1-rhel8_eth0
State              = Online
Managed          = true
Resource Type     = Network Interface
Node              = cm1-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_cm2-rhel8_db2inst1_0
State              = Online
Managed          = true
Resource Type     = Instance
Node              = cm2-rhel8
Instance Name    = db2inst1
```

(続き)

```
Resource Name      = db2_cm2-rhel8_eth0
State              = Online
Managed          = true
Resource Type     = Network Interface
Node              = cm2-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB
Resource Type     = HADR
DB Name           = MYDB
Managed          = true
HADR Primary Instance = db2inst1
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst1
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-primary-VIP
State              = Online
Managed          = true
Resource Type     = IP
Node              = cm1-rhel8
Ip Address        = 192.168.10.17
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-standby-VIP
State              = Online
Managed          = true
Resource Type     = IP
Node              = cm2-rhel8
Ip Address        = 192.168.10.18
```

(後略)

全てのリソースのManagedのステータスが「true」であることを確認

5.1.2 正常停止 (1/3)

- Pacemakerドメイン内のすべてリソースに対する自動制御を、db2cm -disableコマンドにて停止

1. 現在のリソースの制御状況を確認

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Resource Information:

```
Resource Name      = db2_cm1-rhel8_db2inst1_0
State              = Online
Managed          = true
Resource Type     = Instance
Node              = cm1-rhel8
Instance Name     = db2inst1
```

```
Resource Name      = db2_cm1-rhel8_eth0
State              = Online
Managed          = true
Resource Type     = Network Interface
Node              = cm1-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_cm2-rhel8_db2inst1_0
State              = Online
Managed          = true
Resource Type     = Instance
Node              = cm2-rhel8
Instance Name     = db2inst1
```

全てのリソースのManagedのステータスが「true」であることを確認

(続き)

```
Resource Name      = db2_cm2-rhel8_eth0
State              = Online
Managed          = true
Resource Type     = Network Interface
Node              = cm2-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB
Resource Type     = HADR
DB Name           = MYDB
Managed          = true
HADR Primary Instance = db2inst1
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst1
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-primary-VIP
State              = Online
Managed          = true
Resource Type     = IP
Node              = cm1-rhel8
Ip Address        = 192.168.10.17
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-standby-VIP
State              = Online
Managed          = true
Resource Type     = IP
Node              = cm2-rhel8
Ip Address        = 192.168.10.18
```

(後略)

5.1.2 正常停止 (2/3)

2. プライマリー・スタンバイいずれかのノードでdb2cm -disableコマンドを実行し、Pacemakerによるリソースの自動制御を無効化
 - v11.5.5以前ではdb2cm -disableのみで実行可能でしたが、v11.5.6では、“-all”、または“instance *instance_name* -host *host-name-of-the-instance*”オプションが必要です

```
[root@cm1-rhel8 bin]# ./db2cm -disable -all
Automation for Db2 resources are disabled.
```

5.1.2 正常停止 (3/3)

3. Pacemakerによるリソースの自動制御が無効化されたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Resource Information:

```
Resource Name      = db2_cm1-rhel8_db2inst1_0
State              = Online
Managed           = false
Resource Type     = Instance
Node              = cm1-rhel8
Instance Name     = db2inst1
```

```
Resource Name      = db2_cm1-rhel8_eth0
State              = Online
Managed           = false
Resource Type     = Network Interface
Node              = cm1-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_cm2-rhel8_db2inst1_0
State              = Online
Managed           = false
Resource Type     = Instance
Node              = cm2-rhel8
Instance Name     = db2inst1
```

(続き)

```
Resource Name      = db2_cm2-rhel8_eth0
State              = Online
Managed           = false
Resource Type     = Network Interface
Node              = cm2-rhel8
Interface Name    = eth0
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB
Resource Type     = HADR
DB Name           = MYDB
Managed           = false
HADR Primary Instance = db2inst1
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst1
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-primary-VIP
State              = Online
Managed           = false
Resource Type     = IP
Node              = cm1-rhel8
Ip Address        = 192.168.10.17
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB-standby-VIP
State              = Online
Managed           = false
Resource Type     = IP
Node              = cm2-rhel8
Ip Address        = 192.168.10.18
```

(後略)

全てのリソースのManagedの
ステータスが「false」であることを確認

参考情報: リソースの状態確認方法(1/2)

- crm statusコマンドでも同様にリソースの状態を確認することが可能
 - Pacemakerによるリソースの自動制御が無効化されている場合

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

```
* Stack: corosync
* Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Mon Oct 11 11:53:24 2021
* Last change: Mon Oct 11 11:24:42 2021 by root via crm_resource on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured
```

```
Node List:
```

```
* Online: [ cm1-rhel8 cm2-rhel8 ]
```

```
Full List of Resources:
```

```
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8 (unmanaged)
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8 (unmanaged)
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8 (unmanaged)
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8 (unmanaged)
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable) (unmanaged):
  * db2_db2inst1_db2inst1_MYDB (ocf::heartbeat:db2hadr): Master cm1-rhel8 (unmanaged)
  * db2_db2inst1_db2inst1_MYDB (ocf::heartbeat:db2hadr): Slave cm2-rhel8 (unmanaged)
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8 (unmanaged)
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8 (unmanaged)
```

全てのリソースが「unmanaged」の状態であることを確認

参考情報: リソースの状態確認方法(2/2)

- crm statusコマンドでも同様にリソースの状態を確認することが可能(続き)
 - Pacemakerによるリソースの自動制御が有効化されている場合

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Mon Oct 11 12:04:52 2021
* Last change: Mon Oct 11 12:00:21 2021 by root via crm_resource on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
  * Masters: [ cm1-rhel8 ]
  * Slaves: [ cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

全てのリソースから
「unmanaged」の表記が
なくなることを確認

5.1.3 正常引き継ぎ(1/3)

1. 切り替え前のクラスタの状態を確認

- crmコマンドにてHADRのロールを確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Mon Oct 11 12:10:29 2021
* Last change: Mon Oct 11 12:00:21 2021 by root via crm_resource on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2instmon): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2instmon): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
  * Masters: [ cm1-rhel8 ]
  * Slaves: [ cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

primary-VIPがcm1-rhel8ノード
standby-VIPがcm2-rhel8ノード
に紐づいていることを確認

5.1.3 正常引き継ぎ(2/3)

2. 切り替え前のHADRロールを確認

- db2pdコマンドにてHADRロールを確認

- ・ cm1-rhel8ノード(現在はPrimary)

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr | grep ROLE  
HADR_ROLE = PRIMARY
```

- ・ cm2-rhel8ノード(現在はStandby)

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr | grep ROLE  
HADR_ROLE = STANDBY
```

3. スタンバイノード(cm2-rhel8)にてtakeoverコマンドの実行

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ db2 takeover hadr on db mydb  
DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.
```

4. 切り替え後のHADRロールを確認

- db2pdコマンドにてHADRロールを確認

- ・ cm1-rhel8ノード(現在はStandby)

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr | grep ROLE  
HADR_ROLE = STANDBY
```

- ・ cm2-rhel8ノード(現在はPrimary)

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr | grep ROLE  
HADR_ROLE = PRIMARY
```

5.1.3 正常引き継ぎ (3/3)

5. 切り替え後のクラスタの状態を確認

- crmコマンドにてHADRのロールが切り替わったことを確認

```
[root@cm2-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Oct 11 12:54:12 2021
 * Last change: Mon Oct 11 12:00:18 2021 by root via crm_resource on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

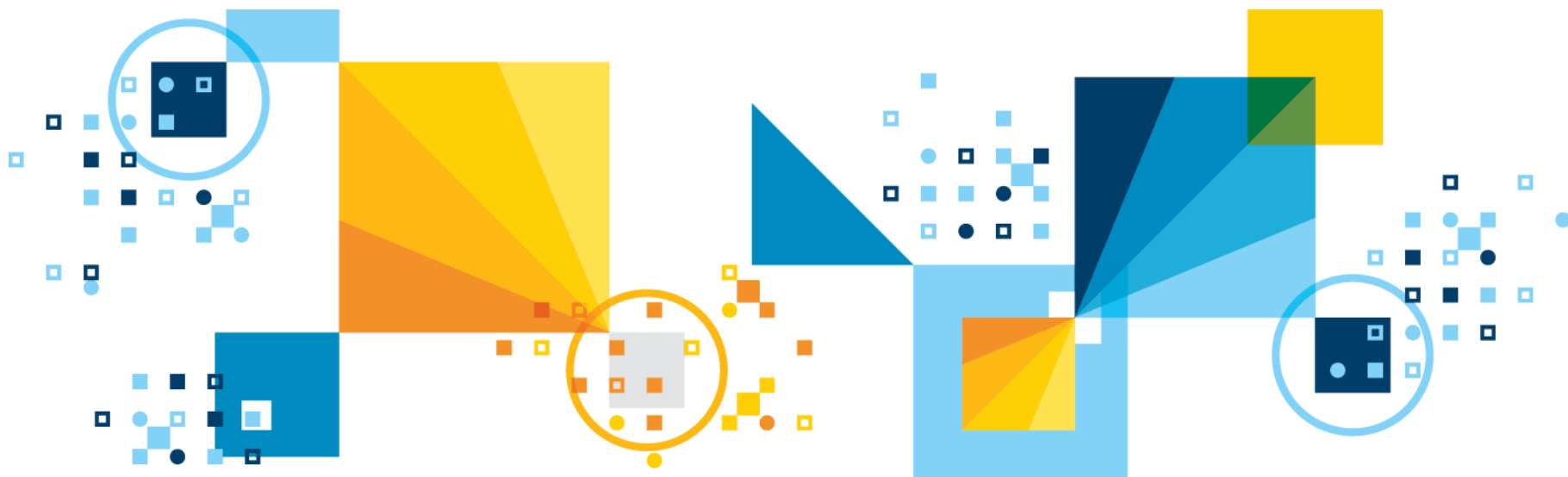
Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm2-rhel8 ]
 * Slaves: [ cm1-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
```

Masters(稼働系)がcm2-rhel8ノード
Slaves(待機系)がcm1-rhel8ノード

primary-VIPがcm2-rhel8ノード
standby-VIPがcm1-rhel8ノード
に紐づいていることを確認

5.2. 障害対応



5.2.1 障害時の対応

■ 基本的な動き

- 障害発生時はPacemakerによって、回復処理が行われる。
- 障害要因が取り除かれた場合、下記条件を満たすことができれば、HADRステータスは自動的にPEER(※1)に戻る。
 - ・ takeover by force時にログギャップが無い(この点に関してはdb2cm構成の場合担保される)
 - ・ テークオーバー後に、旧プライマリDBで更新処理(別のログ・シーケンスが開始されている)が無い
テークオーバー以降のログが新プライマリDBに存在する
 - ・ 旧プライマリDBが破損していない

(※1) PEER状態とは、プライマリ/スタンバイDB間でのデータ整合性が取れる状態のことをさします。

参考情報: Db2 高可用性災害時リカバリー (HADR) データベースの状態

<https://www.ibm.com/docs/ja/db2/11.5?topic=databases-hadr-database-states>

■ 上記条件が満たされない場合は、HADRの初期設定(バックアップのリストア)が必要となります

- [Db2] HADR スタンバイの復旧方法
<https://www.ibm.com/support/pages/node/956395>

5.2.2 障害時サマリー

ケース	障害ケース	挙動
1-1	プライマリーノード停止	<ul style="list-style-type: none"> • TAKEOVERが実行され、ノードが引き継がれる • 回復: 障害ノードが再起動できれば、自動的にインスタンスが起動しロールが切り替わりPEER状態に復帰
1-2	スタンバイノード停止	<ul style="list-style-type: none"> • サービス継続 • 回復: 障害ノードが再起動できれば、自動的にインスタンスが起動しPEER状態に復帰
1-3	クォーラムノード停止	<ul style="list-style-type: none"> • サービスに影響しない • 追加のノード障害発生した場合はサービス停止
2-1	プライマリーノード インスタンス停止	<ul style="list-style-type: none"> • ノードが引き継がれる • 回復: 自動的にインスタンスが再起動しロールが切り替わってPEER状態に復帰
2-2	スタンバイノード インスタンス停止	<ul style="list-style-type: none"> • サービス継続 • 回復: 自動的にインスタンスが再起動しPEER状態に復帰
3-1	プライマリーNIC障害 (NW冗長構成なし)	<ul style="list-style-type: none"> • TAKEOVERが実行され、ノードが引き継がれる • 回復: 障害ノードが再起動できれば、自動的にインスタンスが起動しロールが切り替わりPEER状態に復帰
3-2	スタンバイNIC障害 (NW冗長構成なし)	<ul style="list-style-type: none"> • サービス継続 • 回復: 障害ノードが再起動できれば、自動的にインスタンスが起動しPEER状態に復帰
3-3	クォーラムノードNIC障害 (NW冗長構成なし)	<ul style="list-style-type: none"> • サービスに影響しない • 追加のノード障害発生した場合はサービス停止
4-1	クォーラム動作検証-1	<ul style="list-style-type: none"> • クォーラムノード停止状態でのスタンバイノード停止時は、プライマリーノードのDb2インスタンスが停止
4-2	クォーラム動作検証-2	<ul style="list-style-type: none"> • クォーラムノード停止状態でのプライマリーノード停止時は、スタンバイノードのDb2インスタンスが停止

5.2.3 ノード障害 (ケース1-1 プライマリーノード障害) (1/3)

■ 正常稼働時 (障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Mon Oct 11 13:13:24 2021
* Last change: Mon Oct 11 12:00:21 2021 by root via crm_resource on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
  * Masters: [ cm1-rhel8 ]
  * Slaves: [ cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters (稼働系) が cm1-rhel8 ノード
Slaves (待機系) が cm2-rhel8 ノード

5.2.3 ノード障害 (ケース1-1 プライマリーノード障害) (2/3)

■ 障害発生時

- Pacemakerによりプライマリーノードの障害が検知された場合、自動でTAKEOVERコマンドが実行される
- その結果、cm2-rhel8がプライマリーノードとして稼働を開始する

```
[root@cm2-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.e18-2deceaa3ae) - partition with quorum
- * Last updated: Mon Oct 11 15:00:09 2021
- * Last change: Mon Oct 11 14:59:50 2021 by root via crm_attribute on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm2-rhel8]
- * OFFLINE: [cm1-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): **Stopped**
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): **Stopped**
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters:** [cm2-rhel8]
 - * **Stopped:** [cm1-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): **Stopped**

Masters(稼働系)が
cm2-rhel8ノードに
切り替わる

cm1-rhel8ノードのリソースは
Stoppedとなる

5.2.3 ノード障害 (ケース1-1 プライマリーノード障害) (3/3)

■ 障害回復後

- 障害回復後、障害が発生したノード(cm1-rhel8)では、HADRステータスは自動的にスタンバイDBとしてPEERに戻る
- cm2-rhel8がプライマリーノード、cm1-rhel8がスタンバイノードとして稼働
- (必要に応じて)旧プライマリーノード(cm1-rhel8)にて手動でTAKEOVERコマンドを実行しロールを戻す

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2 takeover hadr on db mydb
DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.
```

```
[root@cm1-rhel8 bin]# crm status
```

Cluster Summary:

- * Stack: corosync
- * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Mon Oct 11 15:57:42 2021
- * Last change: Mon Oct 11 15:32:23 2021 by root via crm_attribute on cm1-rhel8
- * 2 nodes configured
- * 8 resource instances configured

Node List:

- * Online: [cm1-rhel8 cm2-rhel8]

Full List of Resources:

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth0): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth0): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters:** [cm1-rhel8]
 - * **Slaves:** [cm2-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.4 ノード障害 (ケース1-2 スタンバイノード障害) (1/3)

■ 正常稼働時 (障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Tue Oct 12 16:18:24 2021
* Last change: Mon Oct 11 15:32:23 2021 by root via crm_attribute on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2inst1_eth0): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2inst1_eth0): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
  * Masters: [ cm1-rhel8 ]
  * Slaves: [ cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters (稼働系) が cm1-rhel8 ノード
Slaves (待機系) が cm2-rhel8 ノード

5.2.4 ノード障害 (ケース1-2 スタンバイノード障害) (2/3)

■ 障害発生時

- cm1-rhel8ノードにてサービスは継続される(サービスへの影響なし)

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Tue Oct 12 16:40:20 2021
* Last change: Tue Oct 12 16:31:00 2021 by db2inst1 via crm_resource on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 ]
* OFFLINE: [ cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Stopped
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Stopped
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable) (unmanaged):
  * db2_db2inst1_db2inst1_MYDB (ocf::heartbeat:db2hadr): Master cm1-rhel8 (unmanaged)
  * Stopped: [ cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Stopped
```

Masters (稼働系) が cm1-rhel8 ノードのまま
cm2-rhel8 ノードのリソースは「Stopped」となる

5.2.4 ノード障害(ケース1-2 スタンバイノード障害)(3/3)

■ 障害回復後

- 障害回復後、障害が発生したノード(cm2-rhel8)では、HADRステータスは自動的にスタンバイDBとしてPEERに戻る

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Tue Oct 12 16:45:53 2021
- * Last change: Tue Oct 12 16:45:20 2021 by root via crm_attribute on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters: [cm1-rhel8]**
 - * **Slaves: [cm2-rhel8]**
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.5 ノード障害 (ケース1-3 クォーラムノード障害) (1/4)

■ 正常稼働時 (障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働
- Qdevice-net informationのStateが「Connected」となっている

```
[root@cm2-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Fencing Information:

Not configured

Quorum Information:

Qdevice

Qdevice information

Model: Net

Node ID: 2

Configured node list:

0 Node ID = 1

1 Node ID = 2

Membership node list: 1, 2

Qdevice-net information

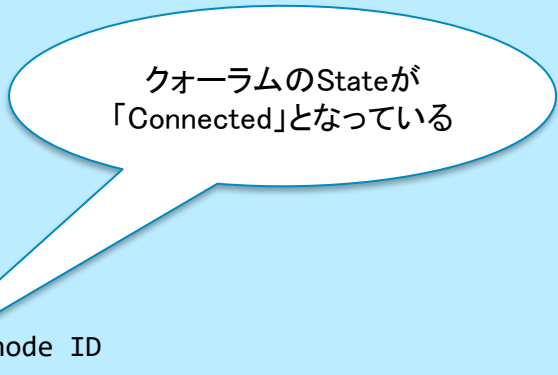
Cluster name: db2domain

QNetd host: cm3-rhel8:5403

Algorithm: LMS

Tie-breaker: Node with lowest node ID

State: **Connected**



クォーラムのStateが
「Connected」となっている

5.2.5 ノード障害 (ケース1-3 クォーラムノード障害) (2/4)

■ 障害発生時

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働 (サービス影響なし)
- ただし、さらなるノード障害が発生した場合にはサービス停止する

```
[root@cm2-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Tue Oct 12 18:14:05 2021
- * Last change: Tue Oct 12 16:45:19 2021 by root via crm_attribute on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters:** [cm1-rhel8]
 - * **Slaves:** [cm2-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8

Masters (稼働系) が cm1-rhel8 ノード
Slaves (待機系) が cm2-rhel8 ノード

5.2.5 ノード障害 (ケース1-3 クォーラムノード障害) (3/4)

■ 障害発生時 (続き)

- Qdevice-net informationのStateが「Connect failed」になる

```
[root@cm2-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Fencing Information:

Not configured

Quorum Information:

Qdevice

Qdevice information

Model: Net

Node ID: 2

Configured node list:

0 Node ID = 1

1 Node ID = 2

Membership node list: 1, 2

Qdevice-net information

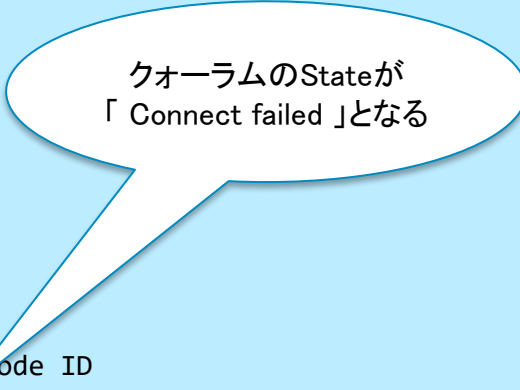
Cluster name: db2domain

QNetd host: cm3-rhel8:5403

Algorithm: LMS

Tie-breaker: Node with lowest Node ID

State: **Connect failed**



クォーラムのStateが
「Connect failed」となる

5.2.5 ノード障害(ケース1-3 クォーラムノード障害)(4/4)

■ 障害回復後

- OS再起動後、クォーラムノードでは自動でcorosync-qnetdプロセスが自動回復する
- Qdevice-net informationのStateも「Connected」に戻る

```
[root@cm3-rhel8 ~]# ps -ef | grep corosync-qnetd | grep -v grep
coroqne+    955      1  0 19:09 ?        00:00:00 /usr/bin/corosync-qnetd -f
```

```
[root@cm2-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Fencing Information:

Not configured

Quorum Information:

Qdevice

Qdevice information

Model: Net

Node ID: 2

Configured node list:

0 Node ID = 1

1 Node ID = 2

Membership node list: 1, 2

Qdevice-net information

Cluster name: db2domain

QNetd host: cm3-rhel8:5403

Algorithm: LMS

Tie-breaker: Node with lowest node ID

State: **Connected**

クォーラムのStateが
「Connected」に戻る

5.2.6 DB障害(ケース2-1 プライマリーDB障害)(1/3)

■ 正常稼働時(障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Wed Oct 13 12:46:16 2021
- * Last change: Tue Oct 12 16:45:20 2021 by root via crm_attribute on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters:** [cm1-rhel8]
 - * **Slaves:** [cm2-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.6 DB障害(ケース2-1 プライマリーDB障害)(2/3)

- 障害発生時(※db2syscプロセスをkill -9コマンドにて停止させた場合)
 - Pacemakerによりプライマリーノードの障害が検知された場合、自動でTAKEOVERコマンドが実行される
 - その結果、cm2-rhel8がプライマリーノードとして稼働を開始する
 - 障害が発生したcm1-rhel8ノードのインスタンスもPacemakerによって自動回復する

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

```
* Stack: corosync
* Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
* Last updated: Wed Oct 13 12:53:08 2021
* Last change: Wed Oct 13 12:52:57 2021 by root via crm_attribute on cm1-rhel8
* 2 nodes configured
* 8 resource instances configured
```

```
Node List:
```

```
* Online: [ cm1-rhel8 cm2-rhel8 ]
```

```
Full List of Resources:
```

```
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2_cm1-rhel8_eth0): Started cm1-rhel8
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2_cm2-rhel8_eth0): Started cm2-rhel8
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
  * Masters: [ cm2-rhel8 ]
  * Slaves: [ cm1-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
```

Masters(稼働系)がcm2-rhel8ノード
Slaves(待機系)がcm1-rhel8ノード

5.2.6 DB障害(ケース2-1 プライマリーDB障害)(3/3)

■ 障害回復後

- 障害回復後、障害が発生したノード(cm1-rhel8)では、HADRステータスは自動的にスタンバイDBとしてPEERに戻る
- cm2-rhel8がプライマリーノード、cm1-rhel8がスタンバイノードとして稼働
- (必要に応じて)旧プライマリーノード(cm1-rhel8)にて手動でTAKEOVERコマンドを実行し、ロールを戻す

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2 takeover hadr on db mydb
DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.
```

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.e18-2deceaa3ae) - partition with quorum
 * Last updated: Wed Oct 13 13:03:20 2021
 * Last change: Wed Oct 13 12:52:57 2021 by root via crm_attribute on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.7 DB障害(ケース2-2 スタンバイDB障害)(1/3)

■ 正常稼働時(障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm2-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Wed Oct 13 13:45:44 2021
 * Last change: Wed Oct 13 12:52:56 2021 by root via crm_attribute on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2inst1_eth0): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2inst1_eth0): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst1): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.7 DB障害(ケース2-2 スタンバイDB障害)(2/3)

- 障害発生時(db2syscプロセスをkill -9コマンドにて停止)
 - cm1-rhel8ノードにてサービス継続(サービスへの影響なし)

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Wed Oct 13 14:07:39 2021
- * Last change: Wed Oct 13 14:07:38 2021 by root via crm_attribute on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth0) Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth0) Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): FAILED cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * db2_db2inst1_db2inst1_MYDB (ocf::heartbeat:db2hdr): FAILED cm2-rhel8
 - * Masters: [cm1-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8

```
(後略)
```

Masters(稼働系)が
cm1-rhel8ノードのまま

5.2.7 DB障害(ケース2-2 スタンバイDB障害)(3/3)

■ 障害回復後

- 障害回復後、障害が発生したインスタンス(cm2-rhel8)では、HADRステータスは自動的にスタンバイDBとしてPEERに戻る
- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Wed Oct 13 14:23:46 2021
 * Last change: Wed Oct 13 14:07:53 2021 by root via crm_attribute on cm2-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2cm1-eth0): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2cm2-eth0): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.8 NIC障害(シングルネットワーク構成時)

- ネットワークが冗長化されていない構成の場合、NIC障害時の挙動はノード障害と同じ挙動となる
 - ケース3-1 プライマリーノードのNIC障害
 - ・ プライマリーノードでは、Db2インスタンスが停止される
 - ・ スタンバイノードでTAKEOVERコマンドが実行され、新プライマリーノードとなる
 - ・ NIC障害回復後、障害が発生した旧プライマリーノードでは、自動的に、Db2インスタンスの起動、HADRロールのスタンバイへの切り替えが行われ、ステータスがPEERに戻る
 - ・ (必要に応じて) 旧プライマリーノード(cm1-rhel8) にて手動でTAKEOVERコマンドを実行し、ロールを戻す(cm1-rhel8がプライマリーノード、 cm2-rhel8がスタンバイノード)
 - ケース3-2 スタンバイノードのNIC障害
 - ・ プライマリーノードでは、HADR_TIMEOUTの間更新がブロックされるが、サービスは継続
 - ・ スタンバイノードでは、Db2インスタンスが停止される
 - ・ NIC障害回復後、障害が発生したノード(cm2-rhel8) では、自動的に、Db2インスタンスが起動され、スタンバイDBとしてPEERに戻る
 - ケース3-3 クォーラムノードNIC障害
 - ・ サービスに影響なし
 - ・ ただし、さらなるノード障害が発生した場合にはサービス停止する

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (1/7)

■ 正常稼働時(障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Wed Oct 13 14:32:15 2021
 * Last change: Wed Oct 13 14:07:53 2021 by root via crm_attribute on cm2-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth) Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (2/7)

■ クォーラムノード障害時

- Qdevice-net informationのStateが「Connect failed」になる
- 引き続きサービスは継続される

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
```

(中略)

Fencing Information:

Not configured

Quorum Information:

Qdevice

Qdevice information

Model: Net

Node ID: 1

Configured node list:

0 Node ID = 1

1 Node ID = 2

Membership node list: 1, 2

Qdevice-net information

Cluster name: db2domain

QNetd host: cm3-rhel8:5403

Algorithm: LMS

Tie-breaker: Node with lowest node ID

State: **Connect failed**

クォーラムのstateが
「Connect failed」となる

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (3/7)

■ 加えてスタンバイノード障害時

- サービスの停止
- プライマリーノード(cm1-rhel8)においてDB非活動化、Db2インスタンスが停止する

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
* Stack: corosync
* Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition WITHOUT quorum
* Last updated: Wed Oct 13 19:10:42 2021
* Last change: Wed Oct 13 14:07:53 2021 by root via crm_attribute on cm2-rhel8
* 2 nodes configured
* 8 resource instances configured

Node List:
* Online: [ cm1-rhel8 ]
* OFFLINE: [ cm2-rhel8 ]

Full List of Resources:
* db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Stopped
* db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Stopped
* db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Stopped
* db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Stopped
* Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
* Stopped: [ cm1-rhel8 cm2-rhel8 ]
* db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2): Stopped
* db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2): Stopped
```

全てのリソースが
「Stopped」となる

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (4/7)

■ クォーラムノード障害回復後

- クォーラムノードのOS再起動後、クォーラムノードでは自動でcorosync-qnetd プロセスが自動回復する
- Qdevice-net informationのStateもConnectedに戻る

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
(中略)
Fencing Information:
  Not configured
Quorum Information:
  Qdevice

Qdevice information
-----
Model:                Net
Node ID:              1
Configured node list:
  0  Node ID = 1
  1  Node ID = 2
Membership node list: 1

Qdevice-net information
-----
Cluster name:         db2domain
QNetd host:          cm3-rhel8:5403
Algorithm:            LMS
Tie-breaker:         Node with lowest node ID
State:                Connected
```

クォーラムのstateが
「Connected」に戻る

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (5/7)

■ 加えてスタンバイノード障害回復後

- スタンバイノードのOS再起動後、スタンバイノードでは自動でDb2インスタンスが再起動され、DBがスタンバイDBとしてHADRが開始される

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ ps -ef | grep db2sysc | grep -v grep
db2inst1      1721      1719    2 19:30 ?                00:00:00 db2sysc 0
```

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ db2pd -hadr -db mydb
```

```
Database Member 0 -- Database MYDB -- Standby -- Up 0 days 00:06:19 -- Date 2021-10-13-19:33:23.18635
```

```

HADR_ROLE = STANDBY
REPLAY_TYPE = PHYSICAL
HADR_SYNCMODE = NEARSYNC
STANDBY_ID = 0
LOG_STREAM_ID = 0
HADR_STATE = REMOTE_CATCHUP_PENDING
HADR_FLAGS =
PRIMARY_MEMBER_HOST = NULL
PRIMARY_INSTANCE = NULL
PRIMARY_MEMBER = NULL
STANDBY_MEMBER_HOST = cm2-rhel8.localdomain
STANDBY_INSTANCE = db2inst1
STANDBY_MEMBER = 0
HADR_CONNECT_STATUS = DISCONNECTED
HADR_CONNECT_STATUS_TIME = 10/13/2021 19:30:47.568047 (1634121047)

```

スタンバイDBとして
自動起動される

(後略)

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (6/7)

- クォーラムノード、スタンバイノード回復後のプライマリーノードの状態
 - 過半数以上のノード回復後、プライマリーノードでは自動でDb2インスタンスは再起動されたが、DBはactivateされなかった。そのため、スタンバイノード回復後に、手動でDBをactivateし、回復を確認した。

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ ps -ef | grep db2sysc | grep -v grep
db2inst1      1876      1866  4 19:32 ?                00:00:02 db2sysc 0
```

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2 activate db mydb
DB20000I The ACTIVATE DATABASE command completed successfully.
```

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr
```

```
Database Member 0 -- Database MYDB -- Active -- Up 0 days 00:02:49 -- Date 2021-10-13-19.40.52.198885
```

```

HADR_ROLE = PRIMARY
REPLAY_TYPE = PHYSICAL
HADR_SYNCMODE = NEARSYNC
STANDBY_ID = 1
LOG_STREAM_ID = 0
HADR_STATE = PEER
HADR_FLAGS = TCP_PROTOCOL
PRIMARY_MEMBER_HOST = cm1-rhel8.localdomain
PRIMARY_INSTANCE = db2inst1
PRIMARY_MEMBER = 0
STANDBY_MEMBER_HOST = cm2-rhel8.localdomain
STANDBY_INSTANCE = db2inst1
STANDBY_MEMBER = 0
HADR_CONNECT_STATUS = CONNECTED

```

プライマリーDBとして
手動で起動

5.2.9 クォーラム動作検証-1

(ケース4-1 クォーラムノード停止状態でのスタンバイノード停止) (7/7)

■ 回復後の最終的な状態

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Wed Oct 13 20:03:53 2021
- * Last change: Wed Oct 13 19:38:07 2021 by db2inst1 via crm_resource on cm1-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth0): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth0): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters: [cm1-rhel8]**
 - * **Slaves: [cm2-rhel8]**
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPAddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPAddr2): Started cm2-rhel8

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止)(1/7)

■ 正常稼働時(障害前)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 bin]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
- * Last updated: Thu Oct 14 14:50:34 2021
- * Last change: Wed Oct 13 19:38:07 2021 by db2inst1 via crm_resource on cm1-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm1-rhel8 cm2-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm1-rhel8
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2eth): Started cm2-rhel8
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 - * **Masters:** [cm1-rhel8]
 - * **Slaves:** [cm2-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPAddr2): Started cm1-rhel8
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPAddr2): Started cm2-rhel8

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止)(2/7)

■ クォーラムノード障害

- Qdevice-net informationのstateが「Connect failed」になる
- 引き続きサービスは継続される

```
[root@cm1-rhel8 bin]# ./db2cm -list -cluster
(中略)

Fencing Information:
  Not configured
Quorum Information:
  Qdevice

Qdevice information
-----
Model:                Net
Node ID:              1
Configured node list:
  0   Node ID = 1
  1   Node ID = 2
Membership node list: 1, 2

Qdevice-net information
-----
Cluster name:         db2domain
QNetd host:           cm3-rhel8:5403
Algorithm:             LMS
Tie-breaker:          Node with lowest node ID
State:                 Connect failed
```

クォーラムのStateが
「Connect failed」となる

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止) (3/7)

■ 加えてプライマリーノード障害

- サービスの停止
- スタンバイノード(cm2-rhel8)におけるのDB非活動化、Db2インスタンス停止

```
[root@cm2-rhel8 ~]# crm status
```

```
Cluster Summary:
```

- * Stack: corosync
- * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition WITHOUT quorum
- * Last updated: Thu Oct 14 15:11:25 2021
- * Last change: Thu Oct 14 15:09:25 2021 by db2inst1 via crm_resource on cm2-rhel8
- * 2 nodes configured
- * 8 resource instances configured

```
Node List:
```

- * Online: [cm2-rhel8]
- * OFFLINE: [cm1-rhel8]

```
Full List of Resources:
```

- * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2ethmon): Stopped
- * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2ethmon): Stopped
- * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Stopped
- * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Stopped
- * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable) (unmanaged):
 - * db2_db2inst1_db2inst1_MYDB (ocf::heartbeat:db2hadr): Slave cm2-rhel8 (unmanaged)
 - * Stopped: [cm1-rhel8]
- * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPAddr2): Stopped
- * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPAddr2): Stopped

全てのリソースが
Stoppedとなる

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止)(4/7)

■ クォーラムノード障害回復後

- クォーラムノードのOS再起動後、クォーラムノードでは自動でcorosync-qnetd プロセスが自動回復する
- Qdevice-net informationのStateもConnectedに戻る

```
[root@cm2-rhel8 bin]# ./db2cm -list -cluster  
(中略)
```

```
Fencing Information:  
  Not configured  
Quorum Information:  
  Qdevice
```

```
Qdevice information  
-----
```

```
Model:                Net  
Node ID:               2  
Configured node list:  
  0   Node ID = 1  
  1   Node ID = 2  
Membership node list:  2
```

```
Qdevice-net information  
-----
```

```
Cluster name:         db2domain  
QNetd host:          cm3-rhel8:5403  
Algorithm:           LMS  
Tie-breaker:         Node with lowest node ID  
State:               Connected
```

クォーラムのStateが
「Connected」に戻る

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止) (5/7)

■ クォーラムノード障害回復後(続き)

- 過半数以上のノード回復後、スタンバイノード(cm2-rhel8)では自動でDb2インスタンスが再起動され、DBがスタンバイDBとしてHADRが開始される

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ ps -ef | grep db2sysc | grep -v grep
db2inst1 1892725 1892723  0 15:26 ?          00:00:33 db2sysc 0
```

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ db2pd -d mydb -hadr
```

```
Database Member 0 -- Database MYDB -- Standby -- Up 0 days 00:06:44 -- Date 2021-10-14-15.33.22.971017
```

```

HADR_ROLE = STANDBY
REPLAY_TYPE = PHYSICAL
HADR_SYNCMODE = NEARSYNC
STANDBY_ID = 0
LOG_STREAM_ID = 0
HADR_STATE = REMOTE_CATCHUP_PENDING
HADR_FLAGS =
PRIMARY_MEMBER_HOST = NULL
PRIMARY_INSTANCE = NULL
PRIMARY_MEMBER = NULL
STANDBY_MEMBER_HOST = cm2-rhel8.localdomain
STANDBY_INSTANCE = db2inst1
STANDBY_MEMBER = 0
HADR_CONNECT_STATUS = DISCONNECTED

```

スタンバイDBとして
自動起動される

(後略)

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止)(6/7)

■ 加えてプライマリーノード障害回復後

- プライマリーノードのOS再起動後、プライマリーノード(cm1-rhel8)では自動でDb2インスタンスが再起動され、DBがプライマリーDBとしてHADRが開始される

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ ps -ef | grep db2sysc | grep -v grep
db2inst1      1876      1866  4 15:47 ?                00:00:02 db2sysc 0
```

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2pd -db mydb -hadr
```

```
Database Member 0 -- Database MYDB -- Active -- Up 0 days 00:14:42 -- Date 2021-10-14-16.02.48.857575
```

```

HADR_ROLE = PRIMARY
REPLAY_TYPE = PHYSICAL
HADR_SYNCMODE = NEARSYNC
STANDBY_ID = 1
LOG_STREAM_ID = 0
HADR_STATE = PEER
HADR_FLAGS = TCP_PROTOCOL
PRIMARY_MEMBER_HOST = cm1-rhel8.localdomain
PRIMARY_INSTANCE = db2inst1
PRIMARY_MEMBER = 0
STANDBY_MEMBER_HOST = cm2-rhel8.localdomain
STANDBY_INSTANCE = db2inst1
STANDBY_MEMBER = 0
HADR_CONNECT_STATUS = CONNECTED

```

プライマリーDBとして
自動起動される

(後略)

5.2.10 クォーラム動作検証-2

(ケース4-2 クォーラムノード停止状態でのプライマリーノード停止)(7/7)

■ 加えてプライマリーノード障害回復後(続き)

- cm1-rhel8がプライマリーノード、cm2-rhel8がスタンバイノードとして稼働

```
[root@cm1-rhel8 ~]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Thu Oct 14 15:49:21 2021
 * Last change: Thu Oct 14 15:48:43 2021 by db2inst1 via crm_resource on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0 (ocf::heartbeat:db2cm): Started cm1-rhel8
 * db2_cm2-rhel8_eth0 (ocf::heartbeat:db2cm): Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst): Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPAddr2): Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPAddr2): Started cm2-rhel8
```

Masters(稼働系)がcm1-rhel8ノード
Slaves(待機系)がcm2-rhel8ノード

5.2.11 HADRデータベース・ハング検出の向上

- Db2 v11.5.6にて、HADRデータベース・ハング検出が改善された
- プライマリーノードでデータベースへの接続が応答しない場合、スタンバイノードへのフェイルオーバーが発生
 - 参照: 高可用性、バックアップ、回復力、およびリカバリーの機能拡張
<https://www.ibm.com/docs/ja/db2/11.5?topic=v1156-high-availability-backup-resiliency-recovery-enhancements>
- データベース接続ハング検出フィーチャーを有効にするためには、DB2_HADR_HANG_DETECTION=CONNECTの設定が必要
 - プライマリー・スタンバイノードで、インスタンス所有者の\$HOME/sqlib/db2profileファイルに次の行を追加

```
export DB2_HADR_HANG_DETECTION=CONNECT
```
 - 参照: 自動 HADR に関するデータベース接続ハング検出フィーチャーの使用可能
<https://www.ibm.com/docs/ja/db2/11.5?topic=solution-enabling-database-connection-hang-detection-automated-hadr>

5.2.12 障害回復後にDBが自動activateされなかった場合の対応

- 2ノード障害からの回復時、自動でDb2インスタンス、DBのactivateが行われないことがある
(常に発生する訳ではない)
 - 例
 - ・ ケース4-1: クォーラムノードとスタンバイノードでの障害
 - ・ ケース4-2: クォーラムノードとプライマリーノードでの障害
 - その場合、インスタンス再起動、DBのactivateを手動で行うことで回復可能

- 参照: データベース・リソースが停止状態のままになる
<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-database-resource-stuck-in-stopped-state>

6. その他運用



6.1 db2cm (HADR) 構成の削除手順

■ db2cm -delete -cluster コマンドの実行

- db2cm -delete コマンドにて現在定義されているクラスター構成を完全に削除する
 - ・ クラスター内に定義されているリソース定義も同時に削除される
 - ・ db2 インスタンスや HADR による同期は停止されることなく、そのまま残される

```
[root@cm1-rhel8 bin]# ./db2cm -delete -cluster
Cluster deleted successfully.
```

■ 削除されたことを確認

- db2cm -list コマンド

```
[root@cm1-rhel8 bin]# ./db2cm -list
Cluster Status
There is no cluster on this host.
```

- crm status コマンド

```
[root@cm1-rhel8 bin]# crm status
Could not connect to the CIB: Transport endpoint is not connected
crm_mon: Error: cluster is not available on this node
ERROR: status: crm_mon (rc=102):
```

■ 参考: Pacemaker を使用する自動化 HADR クラスターの削除

<https://www.ibm.com/docs/ja/db2/11.5?topic=domain-remove-automated-hadr-cluster-pacemaker>

6.2 計画停止時のローリングメンテナンス(1/3)

- Pacemaker で自動化された Db2 高可用性災害時リカバリー (HADR) 環境でのローリング更新の実行

<https://www.ibm.com/docs/ja/db2/11.5?topic=pm-performing-rolling-updates-in-pacemaker-automated-db2-high-availability-disaster-recovery-hadr-environment>

- 常にStandbyロールになっているサーバーのメンテナンスを行うことを前提

1. 1号機がプライマリ、2号機がスタンバイの状態、2号機でのメンテナンスを実施

【2号機(スタンバイノード)】

1. すべてのデータベースのロールがSTANDBYに設定されていることを確認
2. DBのdeactivate
3. Db2インスタンス停止
4. rootユーザーとしてPacemakerおよびCoroysnc全てのプロセスを停止
5. メンテナンス実施

(以下の6~8の手順は、Db2 V11.5.6以降では不要)

6. rootユーザーとしてIBM提供のPacemakerおよびCorosyncの新しいパッケージをインストール
7. root ユーザーとして新規ユーティリティーdb2cmを/<tarFilePath>/Db2/db2cmから
/home/<inst_user>/sqllib/admにコピー
8. rootユーザーとしてリソース・エージェント・スクリプト(db2hadr、db2inst、db2ethmon) を
/<tarFilePath>/Db2agentsから/usr/lib/ocf/resource.d/heartbeat/にコピー
 - 初回の実行時にこのステップを既に完了している場合は、冗長になるため実行しない

6.2 計画停止時のローリングメンテナンス(2/3)

【2号機(スタンバイノード)】(続き)

9. rootユーザーとして、Pacemaker およびCorosync全てのプロセスを開始
10. root ユーザーとして、手動で、または `crm_verify` ツール(使用可能な場合)を使用して構成を確認
11. Db2インスタンス開始
12. DBのactivate
 - ・ スタンバイノードでの計画停止・メンテナンス完了
13. TAKEOVERコマンドを実行し、ロールの切り替えを行う

2. 1号機がスタンバイ、2号機がプライマリの状態で、1号機でのメンテナンスを実施

【1号機(新スタンバイノード)】

14. 元プライマリーノード(新スタンバイノード)にて、1~12の手順を同様に行う
(以下の15~19の手順は、Db2 V11.5.6に更新する場合にのみ必要)
15. 各データベースのmigration-threshold meta属性を更新するために、既存の属性を削除し、新しい値を使用して設定
16. 各データベースのfailure-timeout属性を更新
17. 各データベースについてmigration-threshold属性およびfailure-timeout属性が更新されていることを確認
18. クラスタ構成を更新してsymmetric-clusterをtrueに設定
19. Corosync構成を、ミリ秒のタイム・スタンプを使用するように更新
20. TAKEOVERコマンドを実行し、ロールを元の状態に戻す
21. すべてのデータベースが PEER 状態になっていることを確認

6.2 計画停止時のローリングメンテナンス(3/3)

3. Qdeviceが構成されている場合、Qdeviceホスト上で以下を実施

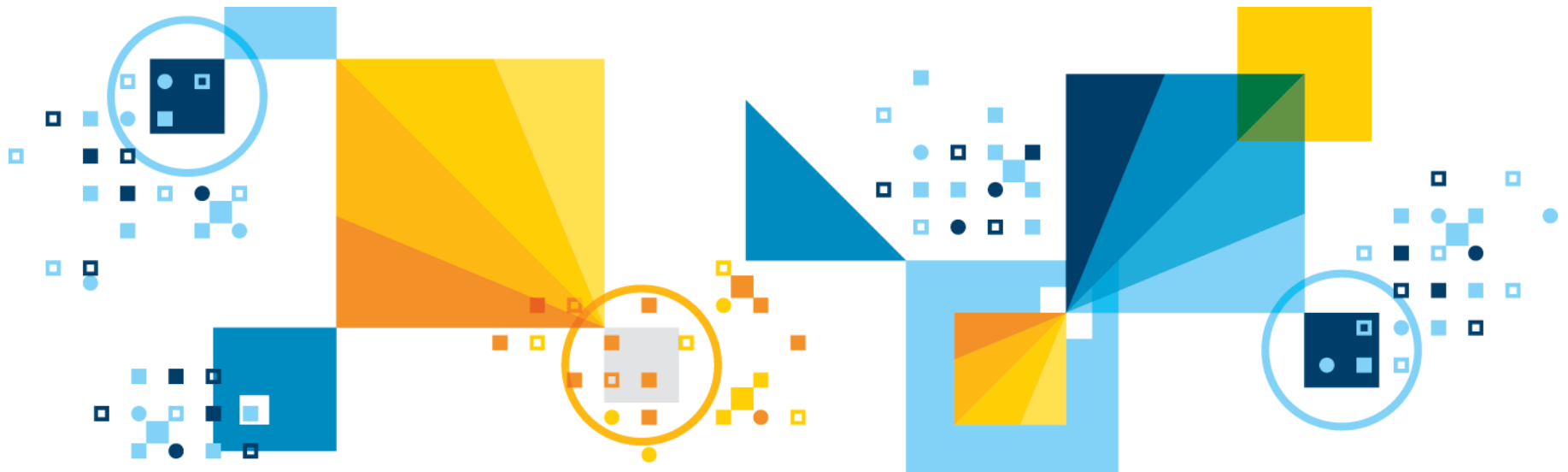
【Qdeviceホスト】

22. rootユーザーcorosync-qnetdプロセスを停止
23. Db2のバージョンに応じて、rootユーザーとしてIBMによって提供されているcorosync-qnetdパッケージを更新
24. rootユーザーとしてqnetdプロセスを開始

4. クラスターが正常な状態であることを確認

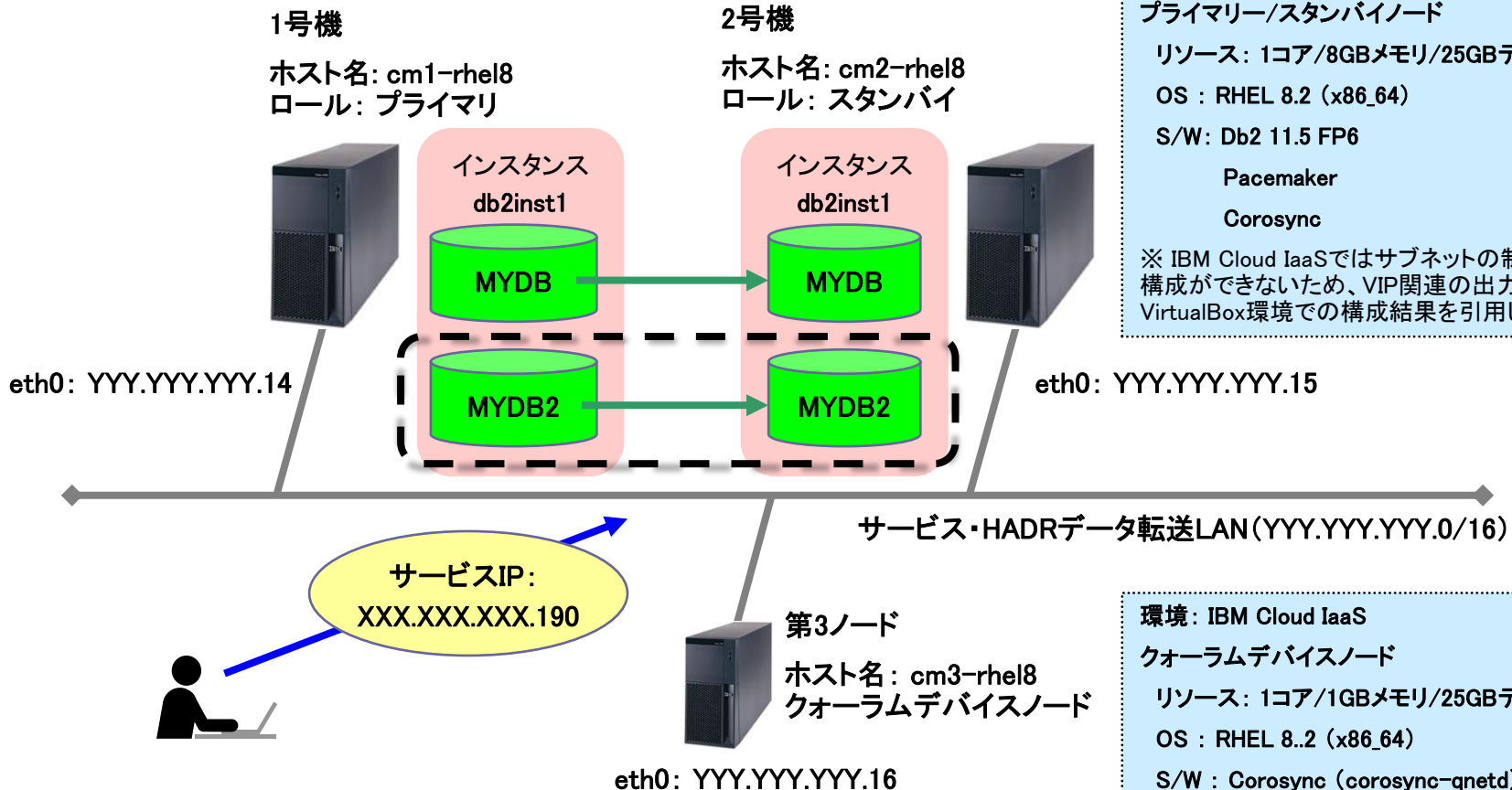
25. DBのactivate
 - ・ 「unmanaged」状態のリソースがなく、すべてのリソースが予期されている役割で開始されていることを確認

6.3 HADRデータベースのリソース追加登録



6.3.1 HADR構成の環境構成図

- 本ガイドは、以下のテスト環境にて検証を実施しています。実際に適用する環境に合わせてカスタマイズを実施してください。
- Cloud環境に構築する場合は、サービスIPを使用した構成ではなくクライアントリルートを使用した構成にすることをご検討ください。



環境: IBM Cloud IaaS

プライマリ/スタンバイノード

リソース: 1コア/8GBメモリ/25GBディスク

OS: RHEL 8.2 (x86_64)

S/W: Db2 11.5 FP6

Pacemaker

Corosync

※ IBM Cloud IaaSではサブネットの制約上VIPの構成ができないため、VIP関連の出力はVirtualBox環境での構成結果を引用しています。

環境: IBM Cloud IaaS

クォーラムデバイスノード

リソース: 1コア/1GBメモリ/25GBディスク

OS: RHEL 8.2 (x86_64)

S/W: Corosync (corosync-qnetd)

6.3.2 Db2/Pacemakerパラメータ抜粋

カテゴリ	パラメータ項目	設定値	備考
OS	Db2オーナー	db2inst1	
	Db2グループ	db2iadm1	
Db2	インスタンス名	db2inst1	
	Db2インスタンスホームディレクトリ	/db2/db2inst1	
	アーカイブログディレクトリ	/db2/arclog	
	データベース名	MYDB2	
	HADRポート(プライマリ)	50011/tcp	※1
	HADRポート(スタンバイ)	50021/tcp	※1
	HADR_SYNCMODE	nearsync	同期モードはSYNCまたはNEARSYNCから選択 HADR_PEER_WINDOWは120秒以上の設定が必須
	HADR_TIMEOUT	30	
	HADR_PEER_WINDOW	120	
	クラスタノード(アクティブ)	cm1-rhel8/cm1-rhel8.localdomain	
	クラスタノード(スタンバイ)	cm2-rhel8/cm2-rhel8.localdomain	
	クォーラムデバイスノード	cm3-rhel8/cm3-rhel8.localdomain	

※1 HADR用通信ポートの登録と開放

/etc/servicesにHADR用通信ポートを定義し、ファイアウォールの設定でポートを開ける

DB2_01_HADRP	50011/tcp
DB2_01_HADRS	50021/tcp

6.3.3 HADRリソースの追加登録(1/4)

- 既存のインスタンスに新規データベース・リソースを作成
 - リソース・モデルへの HADR データベース・リソースの追加
<https://www.ibm.com/docs/ja/db2/11.5?topic=domain-add-hadr-database-resource-resource-model>
 - 追加するDBのペアがプライマリー/スタンバイのロールで起動し、ピア状態であることを確認
 - 現在のクラスター構成を確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Nov  1 13:06:03 2021
 * Last change:  Mon Oct 25 12:54:12 2021 by db2inst1 via crm_resource on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm1-rhel8
 * db2_cm2-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2):        Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2):        Started cm2-rhel8
```

6.3.3 HADRリソースの追加登録(2/4)

- 既存のインスタンスに新規データベース・リソースを作成(続き)
 - db2cmコマンドにてHADRリソースの追加を行う
 - ・ dbオプションにデータベース名を指定(以下の例ではMYDB2)
 - ・ instanceオプションにてインスタンス名を指定(以下の例ではdb2inst1)

```
[root@cm1-rhel8 bin]# ./db2cm -create -db MYDB2 -instance db2inst1
Database resource for MYDB2 created successfully.
```

6.3.3 HADRリソースの追加登録(3/4)

- 現在のクラスター構成を確認し、正常にDBリソースの追加が行われたことを確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Nov  1 13:07:59 2021
 * Last change: Mon Nov  1 13:07:13 2021 by root via cibadmin on cm1-rhel8
 * 2 nodes configured
 * 10 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm1-rhel8
 * db2_cm2-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP      (ocf::heartbeat:IPaddr2):        Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP      (ocf::heartbeat:IPaddr2):        Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB2-clone [db2_db2inst1_db2inst1_MYDB2] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
```

MYDB2リソースが追加されていることを確認
Masters(プライマリロール)はcm1-rhel8
Slaves(スタンバイロール)はcm2-rhel8

6.3.3 HADRリソースの追加登録(4/4)

■ db2cmコマンドでもリソースの状態を確認可能

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
Cluster Status

Domain information:
Domain name           = db2domain
Pacemaker version    = 2.0.4-2.db2pcmk.el8
Corosync version     = 3.0.4-dirty
Current domain leader = cm1-rhel8
Number of nodes      = 2
Number of resources   = 8

Node information:
Name name             State
-----
cm1-rhel8             Online
cm2-rhel8             Online

Resource Information:

Resource Name         = db2_cm1-rhel8_db2inst1_0
State                 = Online
Managed              = true
Resource Type         = Instance
Node                  = cm1-rhel8
Instance Name         = db2inst1

Resource Name         = db2_cm1-rhel8_eth0
State                 = Online
Managed              = true
Resource Type         = Network Interface
Node                  = cm1-rhel8
Interface Name        = eth0
```

(中略)

(続き)

```
Resource Name         = db2_db2inst1_db2inst1_MYDB2
Resource Type         = HADR
DB Name               = MYDB2
Managed              = true
HADR Primary Instance = db2inst1
HADR Primary Node     = cm1-rhel8
HADR Primary State    = Online
HADR Standby Instance = db2inst1
HADR Standby Node     = cm2-rhel8
HADR Standby State    = Online
```

Fencing Information:

Not configured

Quorum Information:

Qdevice

Qdevice information

Model: Net

Node ID: 1

Configured node list:

0 Node ID = 1

1 Node ID = 2

Membership node list: 1, 2

Qdevice-net information

Cluster name: db2domain

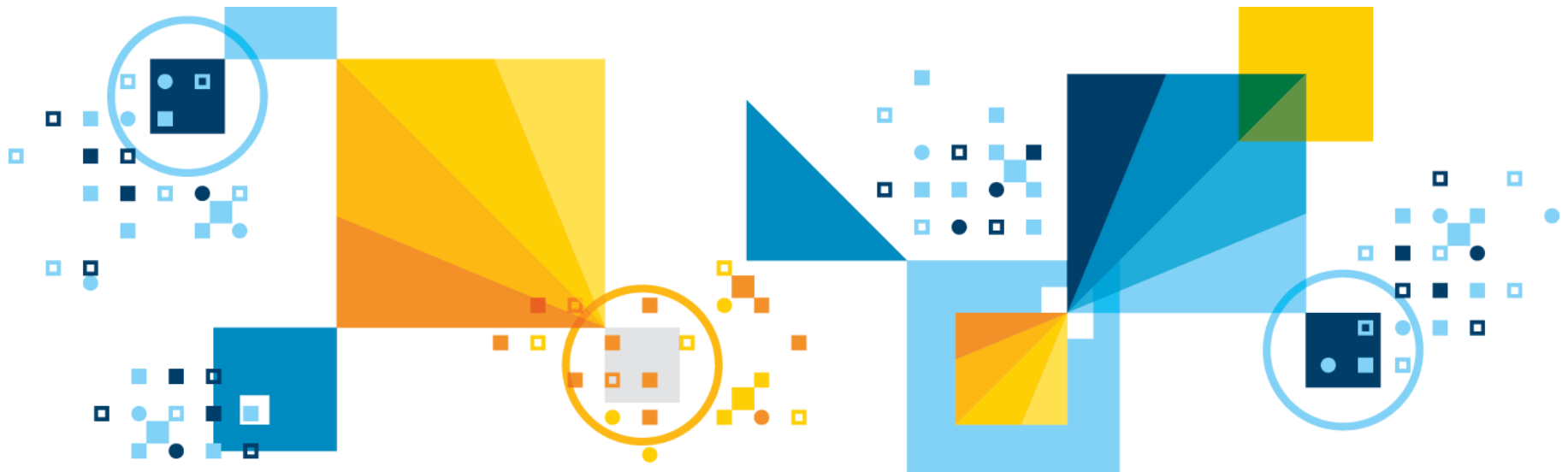
QNetd host: cm3-rhel8:5403

Algorithm: LMS

Tie-breaker: Node with lowest node ID

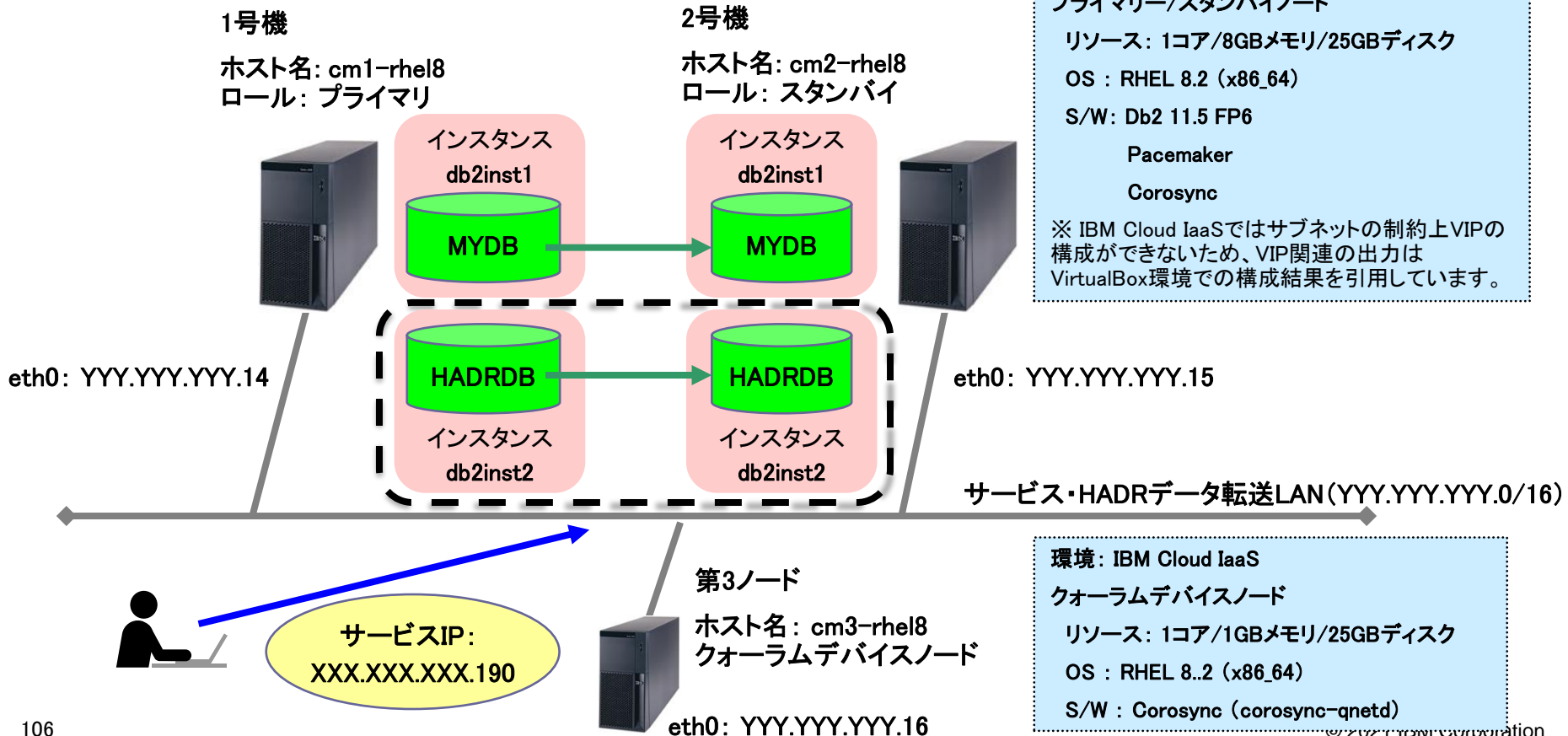
State: Connected

6.4. インスタンスとHADRデータベースの リソース追加登録



6.4.1 HADR構成の環境構成図

- 本ガイドは、以下のテスト環境にて検証を実施しています。実際に適用する環境に合わせてカスタマイズを実施してください。
- Cloud環境に構築する場合は、サービスIPを使用した構成ではなくクライアントリルートを使用した構成にすることをご検討ください。



6.4.2 Db2/Pacemakerパラメータ抜粋

カテゴリ	パラメータ項目	設定値	備考
OS	Db2オーナー	db2inst2	
	Db2グループ	db2iadm2	
Db2	インスタンス名	db2inst2	
	Db2インスタンスホームディレクトリ	/db2/db2inst2	
	アーカイブログディレクトリ	/db2/arclog	
	データベース名	HADRDB	
	HADRポート(プライマリ)	50012/tcp	※1
	HADRポート(スタンバイ)	50022/tcp	※1
	HADR_SYNCMODE	nearsync	同期モードはSYNCまたはNEARSYNCから選択 HADR_PEER_WINDOWは120秒以上の設定が必須
	HADR_TIMEOUT	30	
	HADR_PEER_WINDOW	120	
	クラスタノード(アクティブ)	cm1-rhel8/cm1-rhel8.localdomain	
	クラスタノード(スタンバイ)	cm2-rhel8/cm2-rhel8.localdomain	
	クォーラムデバイスノード	cm3-rhel8/cm3-rhel8.localdomain	

※1 HADR用通信ポートの登録と開放

/etc/servicesにHADR用通信ポートを定義し、ファイアウォールの設定でポートを開ける

DB2_02_HADRP	50012/tcp
DB2_02_HADRS	50022/tcp

6.4.3 インスタンスリソースとHADRリソースの追加登録(1/4)

- 新規のインスタンスリソースおよび新規データベース・リソースを追加
 - リソース・モデルへの HADR データベース・リソースの追加
<https://www.ibm.com/docs/ja/db2/11.5?topic=domain-add-hadr-database-resource-resource-model>
 - 追加するDBのペアがプライマリー/スタンバイのロールで起動し、ピア状態であることを確認
 - 現在のクラスター構成を確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Nov  1 14:35:09 2021
 * Last change:  Mon Nov  1 14:33:21 2021 by root via cibadmin on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm1-rhel8
 * db2_cm2-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP (ocf::heartbeat:IPaddr2):        Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP (ocf::heartbeat:IPaddr2):        Started cm2-rhel8
```

6.4.3 インスタンスリソースとHADRリソースの追加登録(2/4)

■ 新規のインスタンスリソースおよび新規データベース・リソースを追加(続き)

– db2cmコマンドにてインスタンスリソースの登録を行う

- ・ Instanceオプションにインスタンス名を指定(以下の例ではdb2inst2)
- ・ hostオプションにてホスト名を指定する
- ・ インスタンスリソース作成コマンドはプライマリー・スタンバイノード用それぞれ実行(ただし作成コマンドは同一ノードから行う)

```
[root@cm1-rhel8 bin]# ./db2cm -create -instance db2inst2 -host cm1-rhel8
Created db2_cm1-rhel8_db2inst2_0 resource.
Instance resource for db2inst2 on cm1-rhel8 created successfully.
```

```
[root@cm1-rhel8 bin]# ./db2cm -create -instance db2inst2 -host cm2-rhel8
Created db2_cm2-rhel8_db2inst2_0 resource.
Instance resource for db2inst2 on cm2-rhel8 created successfully.
```

– db2cmコマンドにてHADRリソースの追加を行う

- ・ dbオプションにデータベース名を指定(以下の例ではMYDB2)
- ・ Instanceオプションにてインスタンス名を指定(以下の例ではdb2inst2)

```
[root@cm1-rhel8 bin]# ./db2cm -create -db HADRDB -instance db2inst2
Database resource for HADRDB created successfully.
```

6.4.3 インスタンスリソースとHADRリソースの追加登録(3/4)

- 現在のクラスター構成を確認し、正常にDBリソースの追加が行われたことを確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm1-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Nov  1 14:42:15 2021
 * Last change:  Mon Nov  1 14:41:41 2021 by root via cibadmin on cm1-rhel8
 * 2 nodes configured
 * 12 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm1-rhel8
 * db2_cm2-rhel8_eth0      (ocf::heartbeat:db2ethmon):      Started cm2-rhel8
 * db2_cm1-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0 (ocf::heartbeat:db2inst):        Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
 * db2_db2inst1_db2inst1_MYDB-primary-VIP      (ocf::heartbeat:IPAddr2):        Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP      (ocf::heartbeat:IPAddr2):        Started cm2-rhel8
 * db2_cm1-rhel8_db2inst2_0 (ocf::heartbeat:db2inst):        Started cm1-rhel8
 * db2_cm2-rhel8_db2inst2_0 (ocf::heartbeat:db2inst):        Started cm2-rhel8
 * Clone Set: db2_db2inst2_db2inst2_HADRDB-clone [db2_db2inst2_db2inst2_HADRDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
```

インスタンスリソースが
各ノードごとに追加され、
「Started」となることを確認

HADRリソースが追加されていることを確認
Masters(プライマリロール)はcm1-rhel8
Slaves(スタンバイロール)はcm2-rhel8

6.4.3 インスタンスリソースとHADRリソースの追加登録(4/4)

- db2cmコマンドでもリソースの状態がonlineとなることを確認

```
[root@cm1-rhel8 bin]# ./db2cm -cluster -list
(中略)
```

Resource Information:

```
Resource Name      = db2_cm1-rhel8_db2inst1_0
State              = Online
Managed           = true
Resource Type      = Instance
Node               = cm1-rhel8
Instance Name      = db2inst1
```

```
Resource Name      = db2_cm1-rhel8_db2inst2_0
State              = Online
Managed           = true
Resource Type      = Instance
Node               = cm1-rhel8
Instance Name      = db2inst2
```

```
Resource Name      = db2_cm1-rhel8_eth0
State              = Online
Managed           = true
Resource Type      = Network Interface
Node               = cm1-rhel8
Interface Name     = eth0
```

```
Resource Name      = db2_cm2-rhel8_db2inst1_0
State              = Online
Managed           = true
Resource Type      = Instance
Node               = cm2-rhel8
Instance Name      = db2inst1
```

(続き)

```
Resource Name      = db2_cm2-rhel8_db2inst2_0
State              = Online
Managed           = true
Resource Type      = Instance
Node               = cm2-rhel8
Instance Name      = db2inst2
```

```
Resource Name      = db2_cm2-rhel8_eth0
State              = Online
Managed           = true
Resource Type      = Network Interface
Node               = cm2-rhel8
Interface Name     = eth0
```

```
Resource Name      = db2_db2inst1_db2inst1_MYDB
Resource Type      = HADR
DB Name            = MYDB
Managed           = true
HADR Primary Instance = db2inst1
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst1
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

```
Resource Name      = db2_db2inst2_db2inst2_HADRDB
Resource Type      = HADR
DB Name            = HADRDB
Managed           = true
HADR Primary Instance = db2inst2
HADR Primary Node   = cm1-rhel8
HADR Primary State  = Online
HADR Standby Instance = db2inst2
HADR Standby Node   = cm2-rhel8
HADR Standby State  = Online
```

(後略)

6.5 既存のTSAクラスターからPacemakerクラスターへの移行(1/3)

- 以下の手順に従って、既存のTSAクラスターからPacemakerクラスターへ移行を行う

- db2haicuユーティリティーによって構成されたTSAクラスターからの移行を前提とします
- 既存の Tivoli SA MP クラスターの Pacemaker クラスターへの変換
<https://www.ibm.com/docs/ja/db2/11.5?topic=cceudu-converting-existing-tivoli-sa-mp-cluster-pacemaker-cluster>
- 以下ログは、Db2 v11.5.5環境での検証結果

1. 既存のTSAクラスター構成のバックアップを取得(db2インスタンスユーザーで実行)

※ このバックアップイメージはあくまでもdb2haicu構成に切り戻す際に使用するものであり、Pacemakerへの移行手順では使用しません

```
$ db2haicu -o <バックアップファイル名>.xml
```

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2haicu -o db2domain_haicu.xml  
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).
```

```
db2haicu determined the current DB2 database manager instance is 'db2inst1'. The cluster  
configuration that applies to this instance will be exported.  
All cluster configurations have been completed successfully. db2haicu exiting ...
```

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ ls | grep db2domain_haicu.xml  
db2domain_haicu.xml
```


6.5 既存のTSAクラスターからPacemakerクラスターへの移行(2/3)

2. 既存のTSAクラスター構成を削除(db2インスタンスユーザーで実行)

- プライマリー、スタンバイ両ノードでdb2haicu -deleteコマンドを実行

(以下はプライマリー側の実行例)

```
[db2inst1@cm1-rhel8:/home/db2inst1]$ db2haicu -delete
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).
```

You can find detailed diagnostic information in the DB2 server diagnostic log file called db2diag.log. Also, you can use the utility called db2pd to query the status of the cluster domains you create.

For more information about configuring your clustered environment using db2haicu, see the topic called 'DB2 High Availability Instance Configuration Utility (db2haicu)' in the DB2 Information Center.

db2haicu determined the current DB2 database manager instance is 'db2inst1'. The cluster configuration that follows will apply to this instance.

When you use db2haicu to configure your clustered environment, you create cluster domains. For more information, see the topic 'Creating a cluster domain with db2haicu' in the DB2 Information Center. db2haicu is searching the current machine for an existing active cluster domain ...

db2haicu found a cluster domain called 'db2domain' on this machine. The cluster configuration that follows will apply to this domain.

Removing HADR database 'MYDB' from the domain ...

Removing HADR database 'MYDB' from the domain was successful.

Removing DB2 database partition '0' from the cluster ...

Removing DB2 database partition '0' from the cluster was successful.

All cluster configurations have been completed successfully. db2haicu exiting ...

- クラスターが削除されたことを確認

```
[db2inst1@cm2-rhel8:/home/db2inst1]$ lssam
lssam: No online domain found.
```

6.5 既存のTSAクラスターからPacemakerクラスターへの移行(3/3)

3. PacemakerおよびQDevice クォーラムのインストール

- 「3.2. Pacemaker導入手順」、「3.3 QDeviceクォーラムのインストール」に記載された手順にてプライマリー・スタンバイ両ノードへのPacemaker、クォーラムデバイスノードへのQDeviceクォーラムのインストールを行う

4. db2cmコマンドによるPacemakerクラスターの構成

- 「4. db2cmコマンドによるPacemaker構成」に記載された手順にて、Pacemakerクラスターの構成を行う
 - ・ Pacemaker クラスターとパブリック・ネットワーク・リソースの作成
 - ・ インスタンスのリソースモデルを作成
 - ・ HADRリソースの作成
 - ・ (オプション) VIPの作成
 - ・ クォーラムのセットアップ

5. Pacemakerクラスター構成の確認

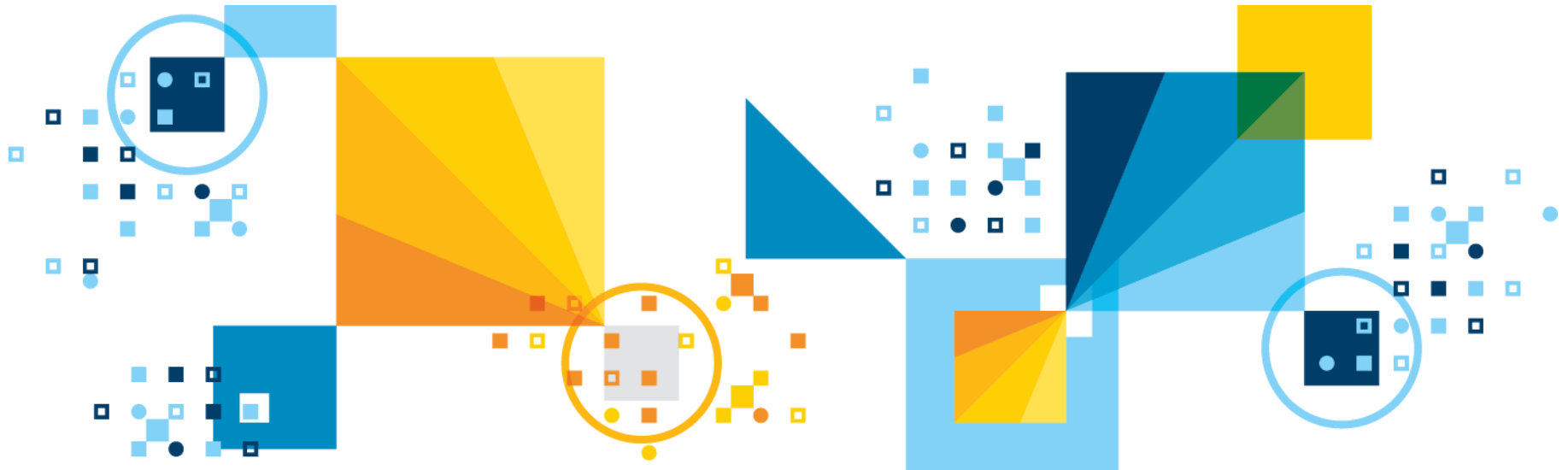
- db2cm -list -clusterコマンドまたはcrm statusコマンドにて構成を確認する

6. TSAのアンインストール

- Db2のインストールイメージまたはTSAのインストールディレクトリーにあるuninstallSAMアンインストール・スクリプトまたはDb2インストーラーを使用してアンインストールを行う
 - ・ IBM Tivoli System Automation for Multiplatforms (SA MP) のアンインストール

<https://www.ibm.com/docs/ja/db2/11.5?topic=uninstalling-tivoli-sa-mp>

6.6. クラスタ構成情報の バックアップ取得・リストア



6.6.1 クラスタ構成情報のバックアップ取得(1/2)

■ db2cm exportコマンド

- 現在のクラスタ構成をテキストファイルに出力させることが可能
- バックアップ・クラスタ構成情報

<https://www.ibm.com/docs/ja/db2/11.5?topic=domain-backup-cluster-configuration-information>

- exportオプションにて出力先ファイル名を指定(以下の例では/tmp/backup.conf)

```
[root@cm1-rhel8 bin]# ./db2cm -export /tmp/backup.conf
Exporting cluster configuration to /tmp/backup.conf...
Export completed successfully.
```

- 出力ファイルを確認

```
[root@cm1-rhel8 bin]# ls -l /tmp/backup.conf
-rw-r--r-- 1 root root 4339 Nov  1 16:02 /tmp/backup.conf
```

6.6.1 クラスター構成情報のバックアップ取得(2/2)

- db2cm exportコマンド(続き)
 - 出力されたテキストファイルの内容例(一部抜粋)

```
[root@cm1-rhel8 bin]# cat /tmp/backup.conf
__BEGIN__CIB__
node 1: cm1-rhel8 ¥
    attributes db2hadr-db2inst1_db2inst1_MYDB_reint=-1
node 2: cm2-rhel8 ¥
    attributes db2hadr-db2inst1_db2inst1_MYDB_reint=-1
primitive db2_cm1-rhel8_db2inst1_0 db2inst ¥
    params instance=db2inst1 hostname=cm1-rhel8 ¥
    op monitor timeout=120s interval=10s on-fail=restart ¥
    op start interval=0s timeout=900s ¥
    op stop interval=0s timeout=900s ¥
    meta migration-threshold=0 is-managed=true
primitive db2_cm1-rhel8_eth0 db2ethmon ¥
    params interface=eth0 hostname=cm1-rhel8 repeat_count=4 repeat_interval=4 ¥
    op monitor timeout=30s interval=4 ¥
    op start timeout=60s interval=0s ¥
    op stop interval=0s timeout=20s ¥
    meta is-managed=true
primitive db2_cm2-rhel8_db2inst1_0 db2inst ¥
    params instance=db2inst1 hostname=cm2-rhel8 ¥
    op monitor timeout=120s interval=10s on-fail=restart ¥
    op start interval=0s timeout=900s ¥
    op stop interval=0s timeout=900s ¥
    meta migration-threshold=0 is-managed=true
primitive db2_cm2-rhel8_eth0 db2ethmon ¥
    params interface=eth0 hostname=cm2-rhel8 repeat_count=4 repeat_interval=4 ¥
    op monitor timeout=30s interval=4 ¥
    op start timeout=60s interval=0s ¥
    op stop interval=0s timeout=20s ¥
    meta is-managed=true
```

(以下省略)

6.6.2 クラスタ構成情報のリストア(1/4)

■ db2cm importコマンド

- 取得したクラスタ構成のバックアップファイルからリストアすることが可能
- 保存した Pacemaker クラスタ構成からリストアする

<https://www.ibm.com/docs/ja/db2/11.5?topic=domain-restore-from-saved-pacemaker-cluster-configuration>

- 現在クラスタ構成が定義されていないことを確認

- ・ 定義が存在する場合には、「6.1 db2cm(HADR) 構成の削除手順」に記載されている、`db2cm -delete -cluster`コマンドを実行する

```
[root@cm1-rhel8 bin]# ./db2cm -list
Cluster Status
```

```
There is no cluster on this host.
```

- バックアップファイルからリストア

- ・ importオプションにてバックアップを指定する(以下の例の場合は/tmp/backup.conf)

```
[root@cm1-rhel8 bin]# ./db2cm -import /tmp/backup.conf
Importing cluster configuration from /tmp/backup.conf...
Import completed successfully.
```

6.6.2 クラスター構成情報のリストア(2/4)

■ db2cm importコマンド(続き)

- db2cm -listコマンドでクラスター情報が正しくリストアされたことを確認

```
[root@cm1-rhel8 bin]# ./db2cm -list
Cluster Status

Domain information:
Domain name           = db2domain
Pacemaker version     = 2.0.4-2.db2pcmk.e18
Corosync version      = 3.0.4-dirty
Current domain leader = cm2-rhel8
Number of nodes       = 2
Number of resources   = 8

Node information:
Name name             State
-----
cm1-rhel8             Online
cm2-rhel8             Online

Resource Information:

Resource Name         = db2_cm1-rhel8_db2inst1_0
State                 = Online
Managed              = true
Resource Type         = Instance
Node                  = cm1-rhel8
Instance Name         = db2inst1
```

(続き)

```
Resource Name         = db2_cm1-rhel8_eth0
State                 = Online
Managed              = true
Resource Type         = Network Interface
Node                  = cm1-rhel8
Interface Name        = eth0

Resource Name         = db2_cm2-rhel8_db2inst1_0
State                 = Online
Managed              = true
Resource Type         = Instance
Node                  = cm2-rhel8
Instance Name         = db2inst1

Resource Name         = db2_cm2-rhel8_eth0
State                 = Online
Managed              = true
Resource Type         = Network Interface
Node                  = cm2-rhel8
Interface Name        = eth0
```

6.6.2 クラスター構成情報のリストア(3/4)

■ db2cm importコマンド(続き)

– db2cm -listコマンドでクラスター情報が正しくリストアされたことを確認(続き)

(続き)

```
Resource Name           = db2_db2inst1_db2inst1_MYDB
Resource Type           = HADR
DB Name                 = MYDB
Managed                = true
HADR Primary Instance  = db2inst1
HADR Primary Node      = cm1-rhel8
HADR Primary State     = Online
HADR Standby Instance  = db2inst1
HADR Standby Node      = cm2-rhel8
HADR Standby State     = Online
```

```
Resource Name           =
db2_db2inst1_db2inst1_MYDB-primary-VIP
State                   = Online
Managed                = true
Resource Type           = IP
Node                   = cm1-rhel8
Ip Address              = 192.168.10.17
```

```
Resource Name           =
db2_db2inst1_db2inst1_MYDB-standby-VIP
State                   = Online
Managed                = true
Resource Type           = IP
Node                   = cm2-rhel8
Ip Address              = 192.168.10.18
```

(続き)

```
Fencing Information:
  Not configured
Quorum Information:
  Qdevice

Qdevice information
-----
Model:                    Net
Node ID:                  1
Configured node list:
  0   Node ID = 1
  1   Node ID = 2
Membership node list:    1, 2

Qdevice-net information
-----
Cluster name:             db2domain
QNetd host:              cm3-rhel8:5403
Algorithm:                LMS
Tie-breaker:             Node with lowest node ID
State:                   Connected
```


6.6.2 クラスター構成情報のリストア(4/4)

■ db2cm importコマンド(続き)

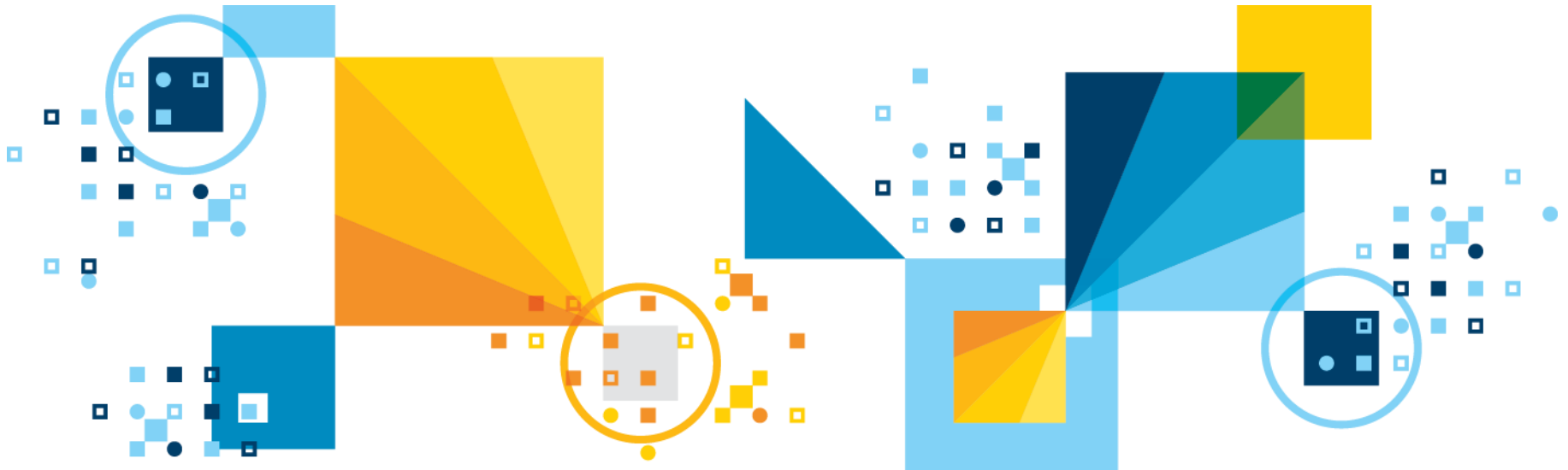
- crm statusコマンドでクラスター情報が正しくリストアされたことを確認

```
[root@cm1-rhel8 bin]# crm status
Cluster Summary:
 * Stack: corosync
 * Current DC: cm2-rhel8 (version 2.0.4-2.db2pcmk.el8-2deceaa3ae) - partition with quorum
 * Last updated: Mon Nov  1 16:26:19 2021
 * Last change:  Mon Nov  1 16:25:01 2021 by root via cibadmin on cm1-rhel8
 * 2 nodes configured
 * 8 resource instances configured

Node List:
 * Online: [ cm1-rhel8 cm2-rhel8 ]

Full List of Resources:
 * db2_cm1-rhel8_db2inst1_0      (ocf::heartbeat:db2inst):      Started cm1-rhel8
 * db2_cm1-rhel8_eth0          (ocf::heartbeat:db2ethmon):    Started cm1-rhel8
 * db2_cm2-rhel8_db2inst1_0      (ocf::heartbeat:db2inst):      Started cm2-rhel8
 * db2_cm2-rhel8_eth0          (ocf::heartbeat:db2ethmon):    Started cm2-rhel8
 * db2_db2inst1_db2inst1_MYDB-primary-VIP      (ocf::heartbeat:IPaddr2):      Started cm1-rhel8
 * db2_db2inst1_db2inst1_MYDB-standby-VIP      (ocf::heartbeat:IPaddr2):      Started cm2-rhel8
 * Clone Set: db2_db2inst1_db2inst1_MYDB-clone [db2_db2inst1_db2inst1_MYDB] (promotable):
 * Masters: [ cm1-rhel8 ]
 * Slaves: [ cm2-rhel8 ]
```

7. 参考：Pacemaker導入 (Db2 V11.5.5以前)



7.1 Pacemaker導入手順 (Db2 V11.5.5以前) (1/5)

■ 製品モジュールの入手

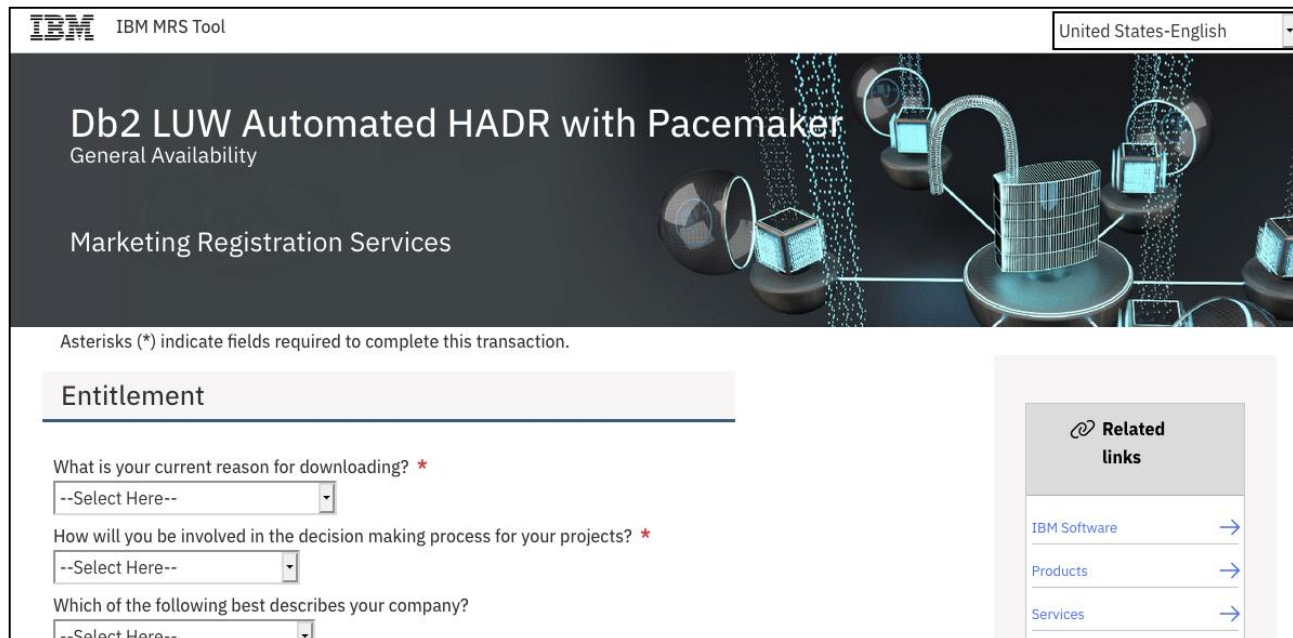
- 以下のリンクからダウンロード可能

<https://www.ibm.com/docs/ja/db2/11.5?topic=utility-installing-pacemaker-cluster-software-stack>

- 開始する前に

- ・ Pacemakerクラスター・ソフトウェア・パッケージを、IBMのWebサイト (Db2® Automated HADR with Pacemaker) からダウンロードします。

<https://www-01.ibm.com/marketing/iwm/platform/mrs/assets?source=mrs-db2pcmk>



IBM MRS Tool United States-English

Db2 LUW Automated HADR with Pacemaker

General Availability

Marketing Registration Services

Asterisks (*) indicate fields required to complete this transaction.

Entitlement

What is your current reason for downloading? *

--Select Here--

How will you be involved in the decision making process for your projects? *

--Select Here--

Which of the following best describes your company?

--Select Here--

Related links

- [IBM Software](#) →
- [Products](#) →
- [Services](#) →

7.1 Pacemaker導入手順 (Db2 V11.5.5以前) (2/5)

■ 導入手順

- Pacemaker クラスター・ソフトウェア・スタックのインストール

<https://www.ibm.com/docs/ja/db2/11.5?topic=pacemaker-installing-cluster-software-stack>

1. 製品モジュールの解凍 * 以下1~6の手順は、HADR両ノードで実行する

- HADR両ノードにPacemaker クラスター・ソフトウェアを任意のディレクトリーに配置し解凍 (以下の例では/workに配置)
- 解凍後、Db2_v11.5.x.x_Pacemaker_202xxxxx_RHEL8.1_x86_64ディレクトリーが作成される (以下の例の場合はDb2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64)

```
# cd /work
# tar -xvf Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64.tar.gz
```

2. 製品インストール

- 解凍先ディレクトリー配下のRPMSディレクトリーに移動し、epelリリースをインストール後 (RHELの場合)、dnfコマンドにてRPM をインストール

```
# cd /work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/RPMS/
# dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
# dnf install */*.rpm
```

7.1 Pacemaker導入手順 (Db2 V11.5.5以前) (3/5)

■ 導入手順(続き)

3. パッケージがインストールされていることを確認

```
# rpm -q corosync
corosync-3.0.4-1.db2pcmk.el8.x86_64

# rpm -q pacemaker
pacemaker-2.0.4-1.db2pcmk.el8.x86_64

# rpm -q crmsh
crmsh-4.2.0-0.db2pcmk.el8.noarch
```

4. db2cm ユーティリティを、クラスター・ソフトウェア・ディレクトリーからインスタンスホームディレクトリー配下のsqlllib/admディレクトリーにコピー

```
# ls /home/db2inst1/sqlllib/adm | grep db2cm

# cp /work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/Db2/db2cm /home/db2inst1/sqlllib/adm/

# ls /home/db2inst1/sqlllib/adm | grep db2cm
db2cm
```

7.1 Pacemaker導入手順 (Db2 V11.5.5以前) (4/5)

- 導入手順(続き)

5. コピーしたdb2cmユーティリティーに権限を付与

```
# ls -al /home/db2inst1/sqllib/adm/db2cm
-rw-r--r-- 1 root db2iadm1 100623 Dec  2 18:50 /home/db2inst1/sqllib/adm/db2cm

# chmod 755 /home/db2inst1/sqllib/adm/db2cm

# ls -al /home/db2inst1/sqllib/adm/db2cm
-rwxr-xr-x 1 root db2iadm1 100623 Dec  2 18:50 /home/db2inst1/sqllib/adm/db2cm
```

7.1 Pacemaker導入手順(Db2 V11.5.5以前) (5/5)

■ 導入手順(続き)

6. 解凍先ディレクトリー配下のDb2agents以下にあるリソース・エージェント・スクリプト (db2hadr、db2inst、db2ethmon)を/usr/lib/ocf/resource.d/heartbeat/にコピー
 - db2cm -copy_resourcesコマンドにて実行
 - Hostオプションにてコピーするホスト名を指定(以下の例ではcm1-rhel8)

```
# ls /usr/lib/ocf/resource.d/heartbeat/ | grep db2
db2

# /home/db2inst1/sqllib/adm/db2cm -copy_resources
/work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/Db2agents/ -host cm1-rhel8
Resources copied from /work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/Db2agents/
to /usr/lib/ocf/resource.d/heartbeat/ successfully on the local host cm1-rhel8

# ls /usr/lib/ocf/resource.d/heartbeat/ | grep db2
db2
db2ethmon
db2hadr
db2inst
```

7. 1~6の手順を、もう片方のノード上でも同様に実施する

7.2 QDevice クォーラムのインストール (Db2 V11.5.5以前) (1/2)

■ 導入手順

– QDevice クォーラムのインストールと構成

<https://www.ibm.com/docs/ja/db2/11.5?topic=utility-install-configure-qdevice-quorum>

1. プライマリー・スタンバイ両ノードにcorosync-qdeviceパッケージがインストールされていることを確認 (rootで実行)

```
# rpm -qa | grep corosync-qdevice
corosync-qdevice-3.0.0-3.db2pcmk.e18.x86_64
corosync-qdevice-debuginfo-3.0.0-3.db2pcmk.e18.x86_64
corosync-qdevice-debugsource-3.0.0-3.db2pcmk.e18.x86_64
```

- インストールされていない場合は、インストールします。(RHELの場合)

```
# dnf install ¥
/work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/RPMS/corosync-qdevice*
```

2. Pacemaker クラスタ・ソフトウェア・パッケージをダウンロードし、クォーラムデバイスノードの任意のディレクトリに配置 (以降の例では/work) (rootで実行)

7.2 QDevice クォーラムのインストール (Db2 V11.5.5以前) (2/2)

■ 導入手順 (続き)

3. クォーラムデバイスノードにてPacemakerクラスター・ソフトウェア・パッケージを解凍 (rootで実行)

```
# cd /work  
# tar -xvf Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64.tar.gz
```

4. クォーラムデバイスノードにてcorosync-qnetdパッケージをインストール (rootで実行)

```
# dnf install ¥  
/work/Db2_v11.5.5.0_Pacemaker_20201118_RHEL8.1_x86_64/RPMS/*/corosync-qnetd*
```