

# Dell Wyse Management Suite

バージョン 1.3 高可用性ガイド



## メモ、注意、警告

① | **メモ:** 製品を使いやすくするための重要な情報を説明しています。

△ | **注意:** ハードウェアの損傷やデータの損失の可能性を示し、その問題を回避するための方法を説明しています。

⚠ | **警告:** 物的損害、けが、または死亡の原因となる可能性があることを示しています。

© 2018 年 Dell Inc. またはその子会社。無断転載を禁じます。Dell、EMC、およびその他の商標は、Dell Inc. またはその子会社の商標です。その他の商標は、それぞれの所有者の商標である場合があります。

<b>1 はじめに</b> .....	<b>4</b>
高可用性の概要.....	4
<b>2 高可用性アーキテクチャ</b> .....	<b>5</b>
高可用性のシステム要件.....	5
<b>3 Windows Server 2012 での高可用性</b> .....	<b>7</b>
クラスタ役割の作成.....	7
<b>4 Windows Server 2012 での高可用性の実現</b> .....	<b>10</b>
Windows Server 2012 でのフェールオーバークラスタ機能の追加.....	10
ファイル共有監視の作成.....	16
クラスタクォーラムの設定.....	17
クラスタ役割の作成.....	19
<b>5 MySQL InnoDB での高可用性の実現</b> .....	<b>22</b>
MySQL InnoDB での高可用性.....	22
MySQL InnoDB データベースのインストール.....	22
MySQL InnoDB サーバインスタンスの確認.....	22
MySQL InnoDB 用のクラスタインスタンスの作成.....	23
MySQL InnoDB クラスタへのサーバインスタンスの追加.....	24
MySQL ルーターの設定.....	25
MySQL InnoDB サーバでのデータベースとユーザーの作成.....	26
<b>6 MongoDB での高可用性の実現</b> .....	<b>27</b>
MongoDB のインストール.....	27
MongoDB データベース用のレプリカサーバの作成.....	28
データベースユーザーの作成.....	28
MongoDB 用の DBadmIn ユーザーの作成.....	29
mongod.cfg ファイルの編集.....	29
サーバでのレプリケーションの開始.....	29
<b>7 Teradici デバイスでの高可用性の実現</b> .....	<b>33</b>
HAProxy のインストールと設定.....	33
Wyse Management Suite サーバのインストール.....	35
<b>8 Wyse Management Suite の Windows Server 2012 へのインストール</b> .....	<b>36</b>
<b>9 インストール後のチェック</b> .....	<b>37</b>
<b>10 トラブルシューティング</b> .....	<b>39</b>

# はじめに

Wyse Management Suite バージョン 1.3 は、Dell Wyse Thin Client を設定、監視、および最適化できる次世代の管理ソリューションです。これは、パフォーマンスの向上した高可用性セットアップでのシンクライアントの導入と管理に役立ちます。

クラウドやオンプレミスでの導入、モバイルアプリケーションによる場所を問わない管理、BIOS 設定やポートロックダウンなどのセキュリティの向上など、最新機能オプションを備えています。その他にも、デバイスの検出 / 登録、資産 / インベントリ管理、設定管理、オペレーティングシステム / アプリケーションの導入、リアルタイムコマンド、監視、アラート、レポート、エンドポイントのトラブルシューティングなどの機能があります。

Wyse Management Suite バージョン 1.3 は高可用性をサポートすることでシステムのダウンタイムを最小限に抑えます。またこのソリューションは、システムでの予期しないダウンタイム発生も防止するため、事業目標の達成に必要な可用性の確保にも寄与します。

本ガイドでは、ソリューションのアーキテクチャについて説明し、高可用性クラスタをアプリケーションとデータベースレベルでセットアップ、構成、および維持する方法を解説します。

## 高可用性の概要

Wyse Management Suite バージョン 1.3 での高可用性ソリューションには、次のタスクが付随します。

- 1 高可用性の要件の確認 - 「[高可用性設定のシステム要件](#)」を参照してください。
- 2 Microsoft Windows Server 2012 での高可用性の導入 - 「[Windows Server 2012 での高可用性の導入](#)」を参照してください。
- 3 MySQL InnoDB サーバでの高可用性の導入 - 「[MySQL InnoDB サーバでの高可用性の導入](#)」を参照してください。
- 4 MongoDB での高可用性の導入 - 「[MongoDB での高可用性の導入](#)」を参照してください。
- 5 高可用性プロキシの設定 ( Teradici デバイス用 ) - 「[Teradici サーバ用の高可用性の導入](#)」を参照してください。
- 6 Wyse Management バージョンの Windows Server 2012 へのインストール - 「[Wyse Management Suite の Windows Server 2012 へのインストール](#)」を参照してください。
- 7 インストール後のチェック - 「[インストール後のチェック](#)」を参照してください。
- 8 問題のトラブルシューティングと回避策 - 「[トラブルシューティング](#)」を参照してください。

## 高可用性アーキテクチャ

Wyse Management Suite のアーキテクチャは、フェールオーバークラスタを有効にした Windows Server 2012 で構成されています。Windows クラスタには 1 台のメインコンピュータが含まれており、これは他のアプリケーションをサポートすると同時に、冗長性を確保することでダウンタイムを最小化します。これは、Tomcat、Memcache、MQTT サービスでの、アプリケーションのフェールオーバーに使用されます。MongoDB データベースクラスタは、プライマリデータベースでの障害発生時におけるセカンダリデータベースへの引き継ぎを支援するようにされています。MySQL InnoDB データベースクラスタには、データベースのクラスタリング機構が組み込まれており、プライマリの読み取り / 書き込みデータベースでの処理が失敗した場合は、セカンダリデータベースに引き継がれます。HA プロキシを装備した Linux サーバは一種の負荷バランサであり、EMSDK ( Teradici ) サーバ用の高可用性サーバとして機能します。ローカルリポジトリは、アプリケーション、イメージ、パッケージを含む共有パスの一部として作成され、クラスタセットアップの一部にはなりません。

① | **メモ:** 高可用性システムの要件は、ワークサイトのインフラストラクチャに応じて異なる可能性があります。

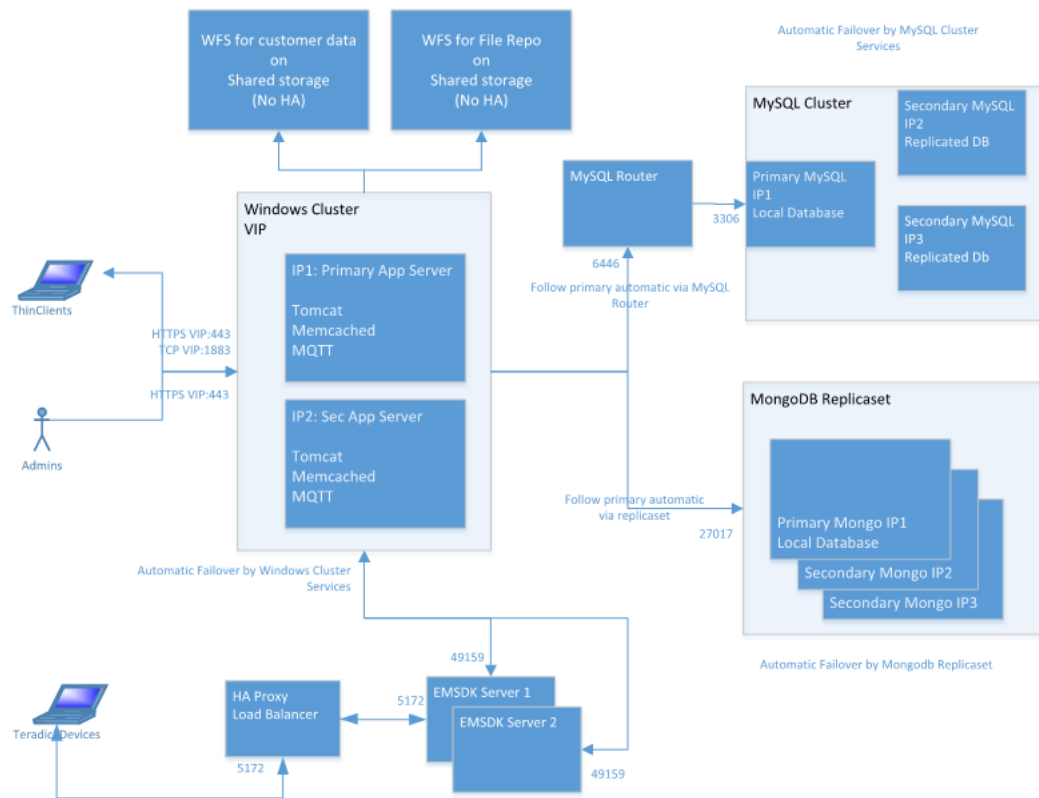


図 1. 高可用性アーキテクチャ

## 高可用性のシステム要件

この表は、ハードウェアとソフトウェアの最小要件をまとめたもので、最大 10,000 台のデバイスをサポートします。EMSDK の各インスタンスは、最大 5000 台のデバイスをサポートできます。導入は、要件に応じて、個々のサーバまたはハイパーバイザ環境で実行できます。

Wyse Management Suite バージョン 1.3 の高可用性を設定するためのハードウェアおよびソフトウェアの要件は、次のとおりです。

表 1. システム要件

製品	ポート	プロトコル	説明
Microsoft Windows Server 2012 R2	<ul style="list-style-type: none"> <li>ネットワーク通信ポート :                             <ul style="list-style-type: none"> <li>UDP : 3343</li> <li>TCP : 3342</li> <li>UDP : 137</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 2</li> <li>最小メモリ ( RAM ) - 8 GB</li> <li>最小 CPU 要件 - 4</li> </ul>	<p>Wyse Management Suite がホストされているサーバ。</p> <p>英語、フランス語、イタリア語、ドイツ語、スペイン語をサポート。</p>
MySQL クラスタ	<ul style="list-style-type: none"> <li>ネットワーク通信ポート - TCP : 3306</li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 3</li> <li>最小メモリ ( RAM ) - 8 GB</li> <li>最小 CPU 要件 - 4</li> </ul>	高可用性セットアップのサーバ。
MySQL ルーター	<ul style="list-style-type: none"> <li>ネットワーク通信ポート :                             <ul style="list-style-type: none"> <li>6446</li> <li>6447</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 2</li> <li>最小メモリ ( RAM ) - 8 GB</li> <li>最小 CPU 要件 - 4</li> </ul>	高可用性セットアップでの通信を確立。
MongoDB	<ul style="list-style-type: none"> <li>ネットワーク通信ポート - TCP : 27017</li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 3</li> <li>最小メモリ ( RAM ) - 8 GB</li> <li>最小 CPU 要件 - 4</li> </ul>	データベース
EMSDK	<ul style="list-style-type: none"> <li>ネットワーク通信ポート - TCP : 5172</li> <li>TCP 49159</li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 2</li> <li>最小メモリ ( RAM ) - 8 GB</li> <li>最小 CPU 要件 - 4</li> </ul>	Enterprise SDK サーバ
HAProxy	<ul style="list-style-type: none"> <li>ネットワーク通信ポート - TCP : 5172</li> </ul>	<ul style="list-style-type: none"> <li>最小ディスク容量 - 40 GB</li> <li>システム最小数 - 1</li> <li>最小メモリ ( RAM ) - 4 GB</li> <li>最小 CPU 要件 - 2</li> </ul>	高可用性セットアップの負荷バランサ。 Ubuntu バージョン 12.04 以降。

① **メモ:**

高可用性のセットアップ中に、ファイアウォール例外リストに TCP ポートの 443、8080、1883 を追加してください。

## Windows Server 2012 での高可用性

フェールオーバークラスタとは、クラスタ化された役割の可用性と拡張性を向上する独立したシステムの一群です。この機能は、ハードウェアまたは仮想マシン上にあるクラスタ群での複数ワークロードの実行をサポートします。

フェールオーバークラスタは、クラスタ化された役割の可用性と拡張性を向上する独立したシステムの一群です。クラスタ化されたサーバ群は、複数のノードを1つのノードに対して相互にネットワーク接続したものです。1つまたは複数のクラスタノードで障害が発生した場合は、他のノードがアクティブになることで、ネットワークにあるシステムのフェールオーバーを防止します。クラスタのセットアップ中にはクラスタ化された役割が作成され、これらがモニタすることで、クラスタ化されたネットワーク中でのシステム群の動作が検証されます。いずれかのシステムが動作を停止した場合、それらの再起動または別ノードへの移動が行われます。

Windows Server 2012 で高可用性を確保するためのフェールオーバークラスタネットワークには、ノード1とノード2の2つのノードが存在し、これらはWindows Server 2012の実行システム上に構成されます。フェールオーバークラスタネットワークにおいて、プライマリノードとして動作中のノード1が障害した場合は、自動的にノード2がプライマリノードとして動作し始めます。ノード1がアクティブになった後は、これが自動的にセカンダリノードになります。システムには、ネットワーク接続された共有ストレージスペースがあります。

① | **メモ:** 1つの例として、イメージにあるシステムの IP アドレスは、各システムが作業する場所に応じて変化します。

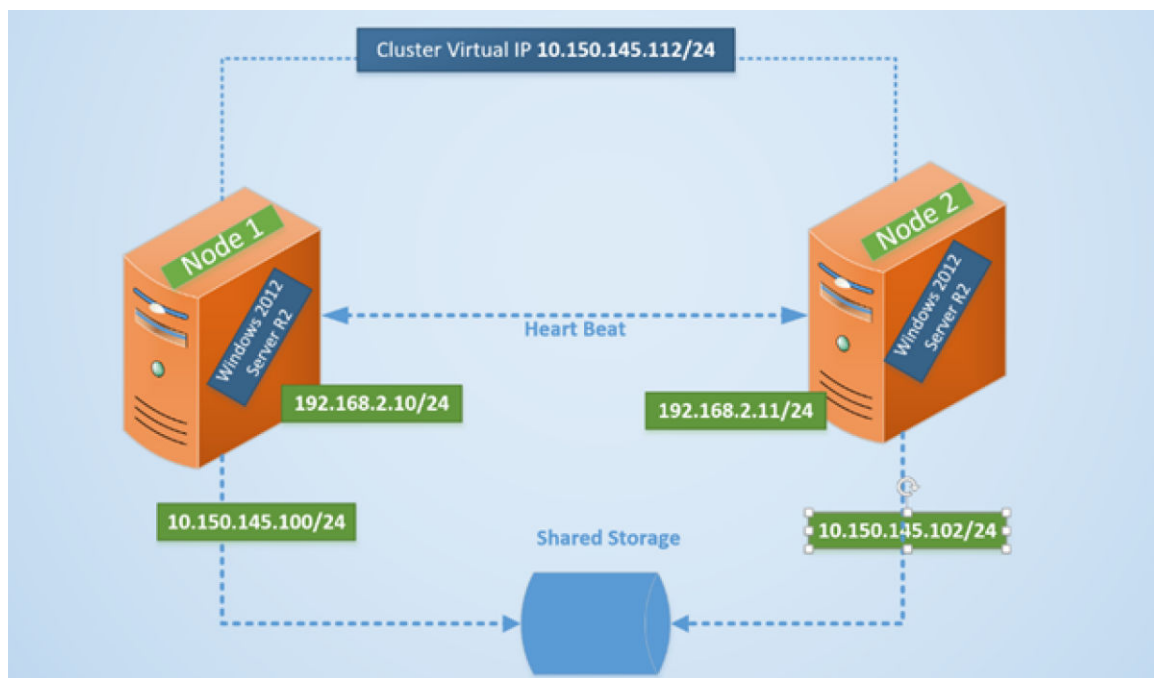


図 2. フェールオーバークラスタのセットアップ

### クラスタ役割の作成

フェールオーバークラスタの作成後、クラスタのワークロードをホストするクラスタ化された役割の作成が可能になります。クラスタ化された役割を作成する前に、Wyse Management Suite がサーバにインストールされており、リモートデータベースをポイントしていることを確認してください。

クラスタ化された役割を作成するには、次の手順を実行します。

- 1 Microsoft Windows Server 2012 で **スタート** メニューを右クリックし、**サーバマネージャ** を選択して、サーバマネージャのダッシュボードを起動します。
- 2 **フェールオーバークラスタマネージャ** をクリックしてクラスタマネージャを起動します。
- 3 **役割** を右クリックして **役割の構成** を選択し、**高可用性ウィザード** 画面を表示します。

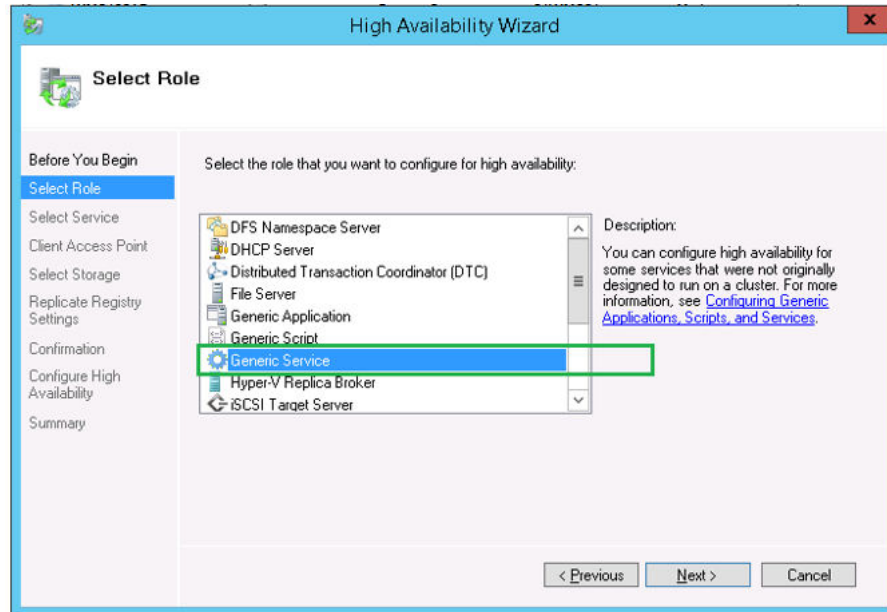


図 3. 高可用性ウィザード

- 4 **汎用サービス** を選択してから **次へ** をクリックし、**サービスの選択** 画面を表示します。

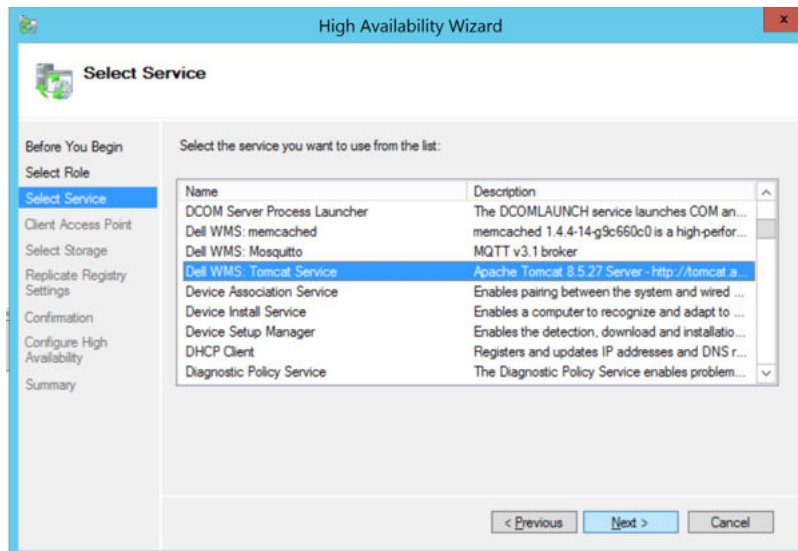


図 4. サービスの選択

- 5 **Dell WMS: Tomcat Service** を選択し、**次へ** をクリックします。

① **メモ:** Wyse Management Suite バージョン 1.3 の関連サービスをクラスタに追加できるのは、Wyse Management Suite バージョン 1.3 をインストールした後だけです。

高可用性ウィザード画面が表示されますが、ここでは、クライアントアクセスポイントの作成および、Windows Server 2012 と Wyse Management Suite 間の接続を確立する必要があります。

- 6 **名前** フィールドにネットワークの名前を入力して、**次へ** をクリックします。**確認** 画面が表示され、サーバのネットワーク名と IP アドレスの詳細が表示されます。

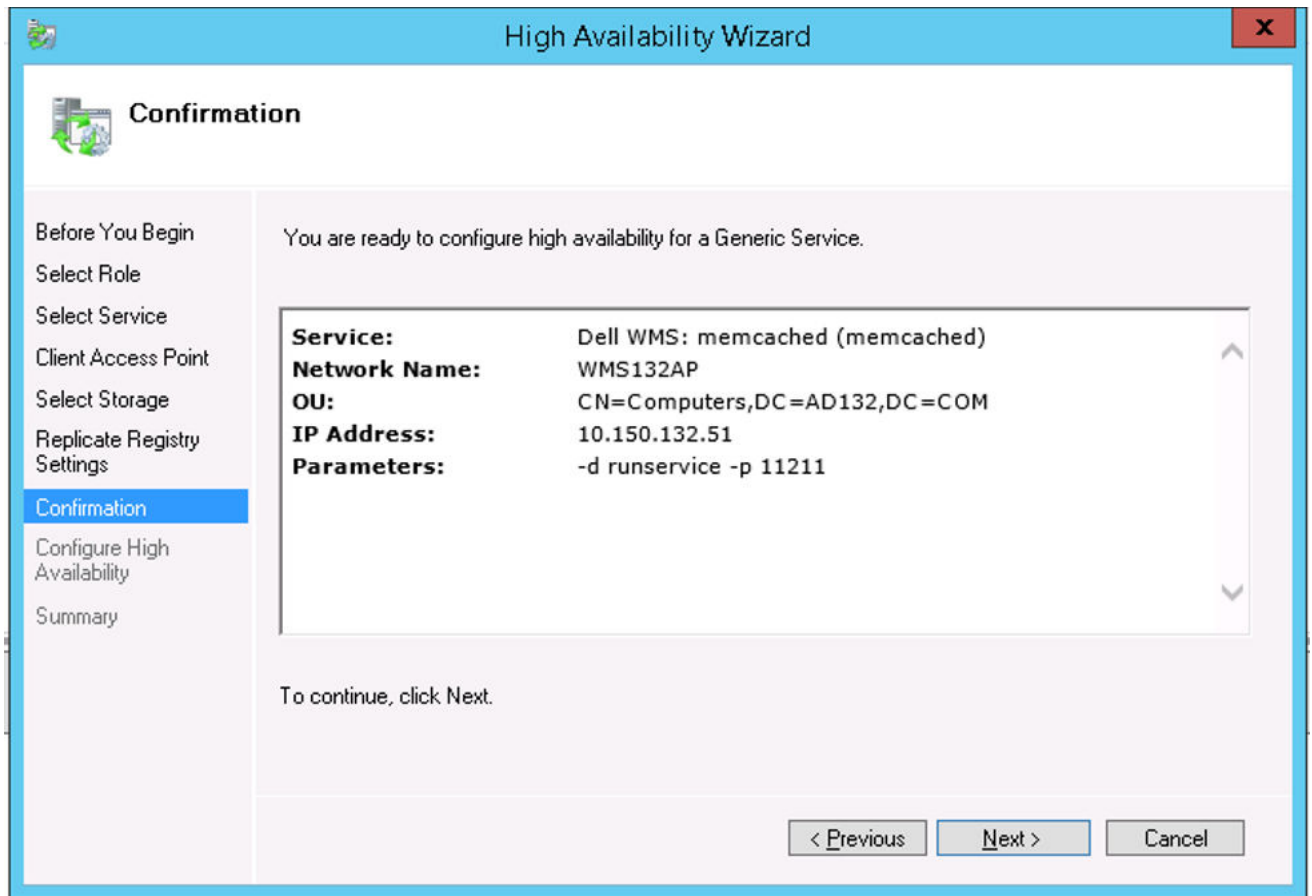


図 5. 確認

- 7 **次へ** をクリックして、プロセスを完了します。
- 8 クラスタの一部として他の Wyse Management Suite サービスを追加するには、**フェールオーバークラスタマネージャー**を起動して **アクション > 役割** に移動し、作成したネットワーク名を表示します。
- 9 ネットワーク名をクリックして、**リソースの追加 > 汎用サービス** に移動します。
- 10 下記のサービスでクラスタの一部として追加する必要があるものを **新しいリソースウィザード** 画面から選択します。
- a Dell WMS: Mosquitto >> MQTT Broker
  - b Dell WMS: memcached
- 11 **次へ** をクリックしてタスクを完了します。
- クラスタの一部として追加された Wyse Management Suite サービスのステータスは **実行中** と表示されます。

# Windows Server 2012 での高可用性の実現

Windows Server 2012 で高可用性を実現する手順は、次のとおりです。

- 1 Windows Server 2012 にフェールオーバークラスタ機能を追加します - 「[Windows Server 2012 でのフェールオーバークラスタ機能の追加](#)」を参照してください。
- 2 ファイル共有を作成します - 「[ファイル共有監視の作成](#)」を参照してください。
- 3 クラスタクォーラムを設定します - 「[クラスタクォーラムの設定](#)」を参照してください。
- 4 クラスタ化された役割を作成します - 「[クラスタ役割の作成](#)」を参照してください。

## Windows Server 2012 でのフェールオーバークラスタ機能の追加

Windows Server 2012 でフェールオーバークラスタリング機能を追加するには、次の手順を実行します。

- 1 Microsoft Windows Server 2012 で **スタート** をクリックして **スタート** 画面を開き、**サーバマネージャー** をクリックして **サーバマネージャー** ダッシュボードを起動します。
  - ① **メモ:** サーバマネージャーは Windows Server 2012 の管理コンソールで、サーバの役割と機能の追加および、サーバの管理と導入をすることができます。
- 2 **役割と機能の追加** をクリックし、**役割と機能の追加ウィザード** 画面で各自の要件に基づいてオプションを選択します。

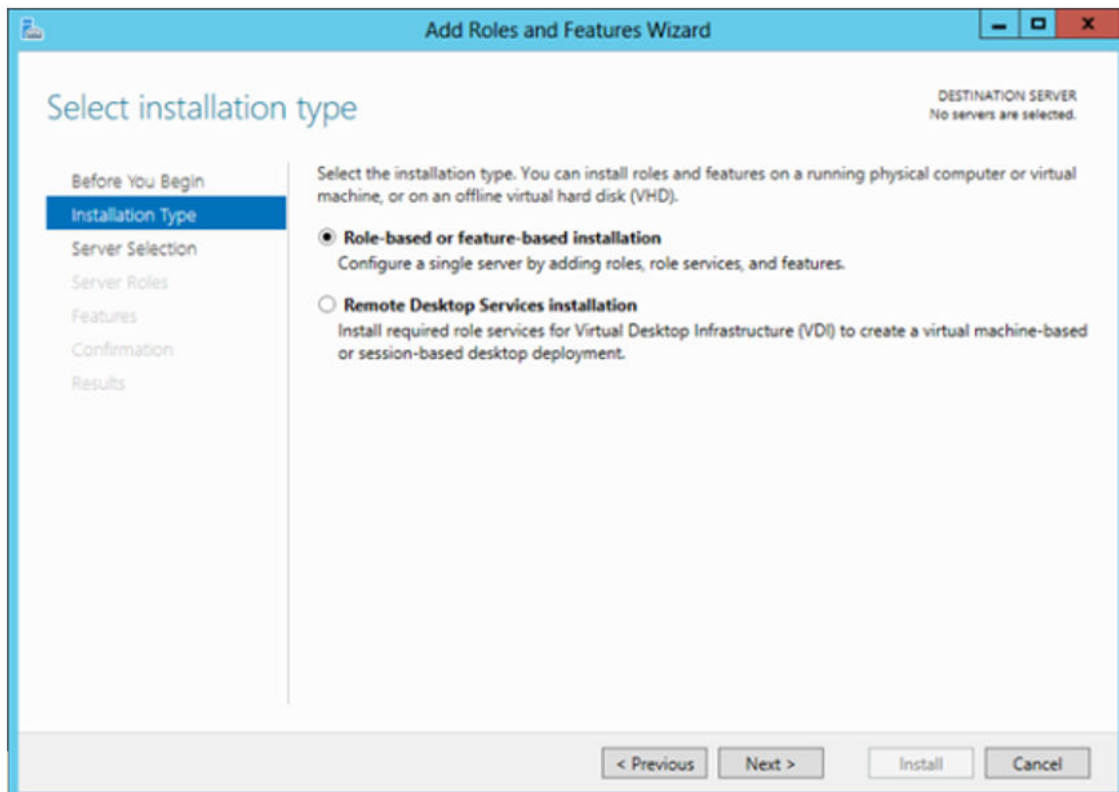


図 6. 役割ベースの選択

- 3 インストールタイプ、役割ベースまたは機能ベースのインストール、次への順にクリックして **対象サーバの選択** 画面を表示させ、サーバのリストを確認します。

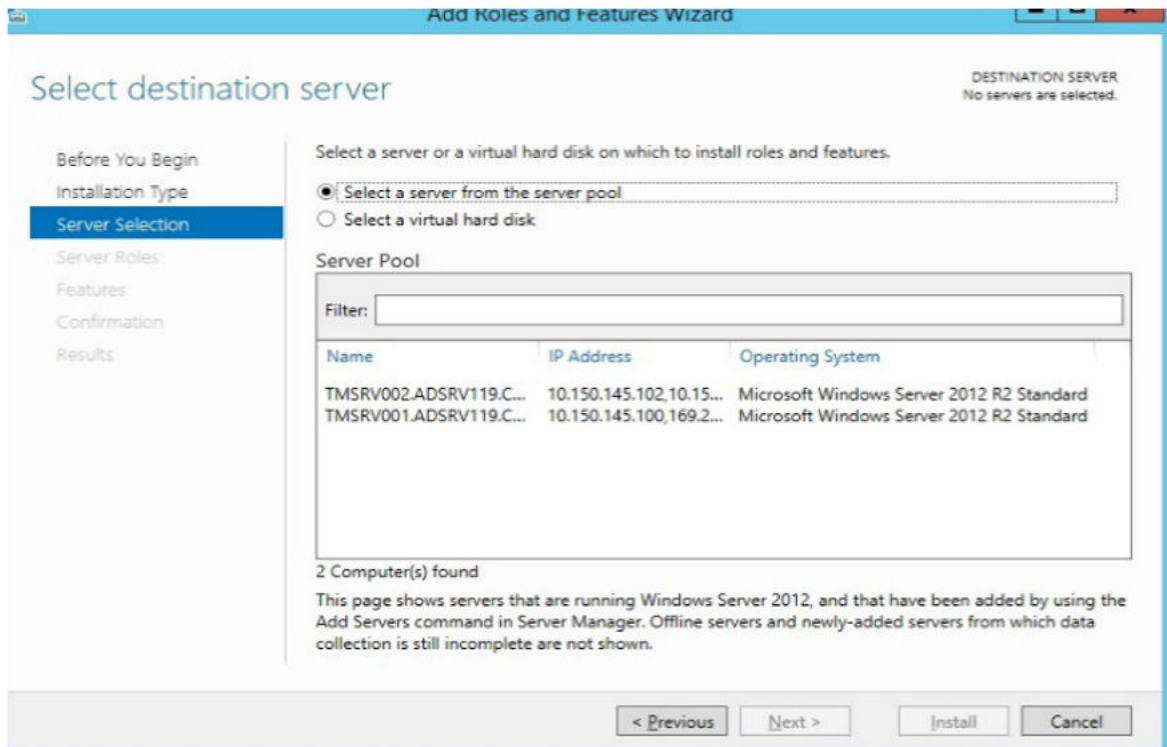


図 7. 対象とするサーバの選択

- 4 フェールオーバークラスタ機能を有効にするサーバを選択し、**次へ** をクリックします。
- 5 **機能** 画面で **フェールオーバークラスタリング** を選択し、**次へ** をクリックします。サーバでフェールオーバークラスタを有効にした後で、ノード 1 のサーバで **フェールオーバークラスタマネージャー** を開きます。
- 6 **はい** をクリックしてインストールを確認し、選択したサーバでのフェールオーバークラスタ機能を有効にします。
- 7 **フェールオーバークラスタマネージャー** 画面で **設定の確認** をクリックして **設定の検証ウィザード** を表示させ、クラスタに必要なサーバまたはノードを追加します。

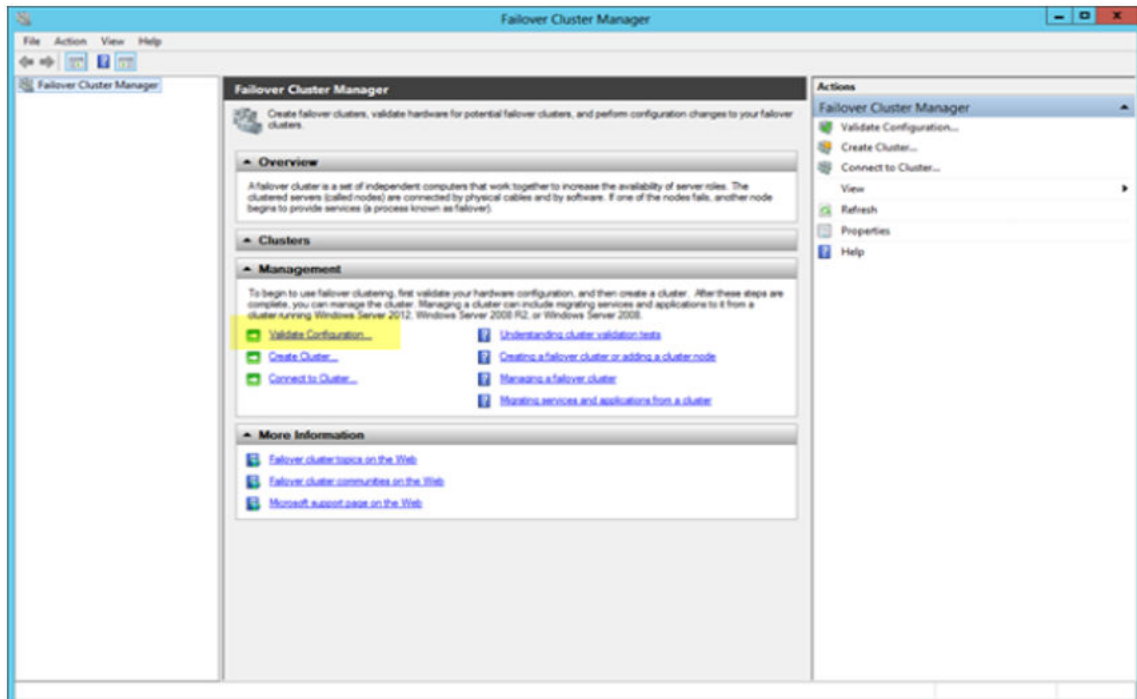


図 8. フェールオーバークラスターマネージャー

- 8 サーバまたはクラスタの選択、参照の順にクリックして、サーバを設定します。
- 9 次へをクリックして、オプションのテスト画面からすべてのテストを実行を選択します。

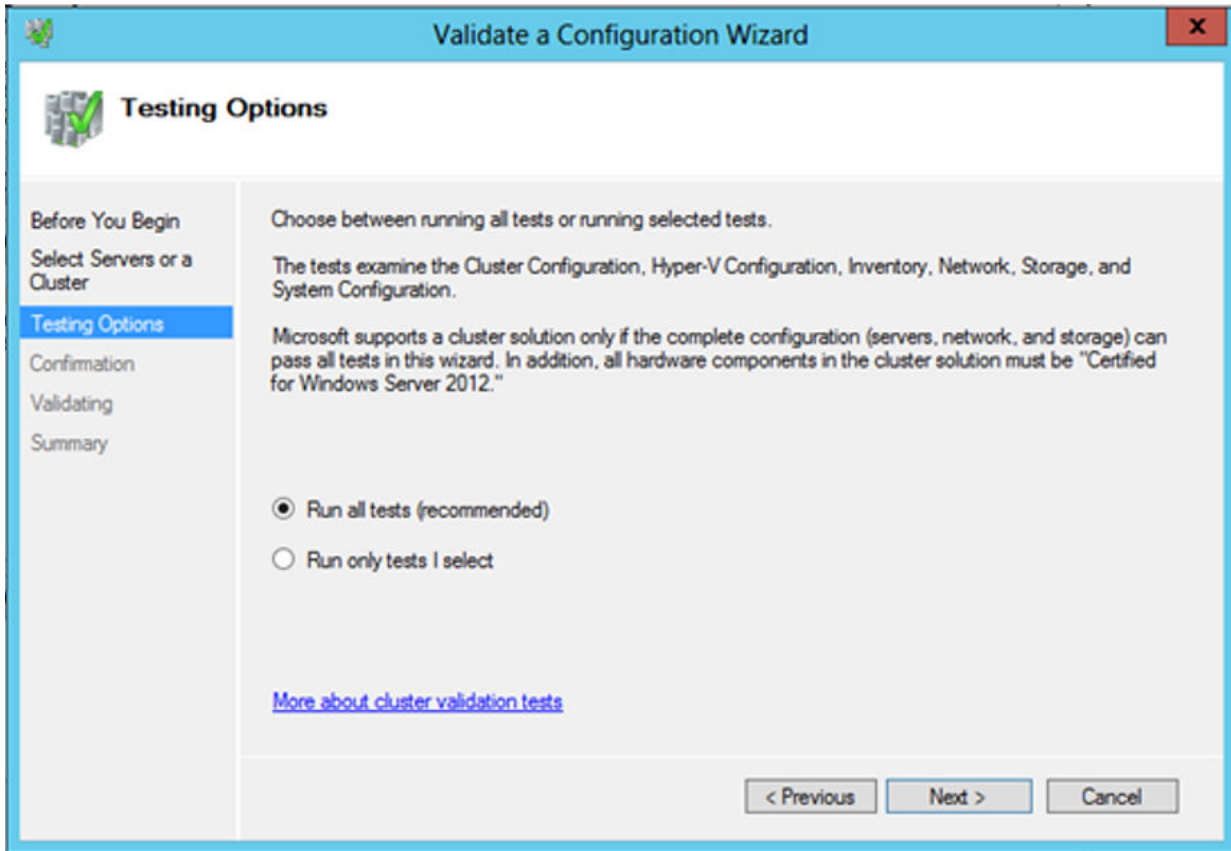


図 9. オプションのテスト

- 10 **次へ** をクリックします。**確認** 画面が表示され、選択したサーバのリストが示されます。

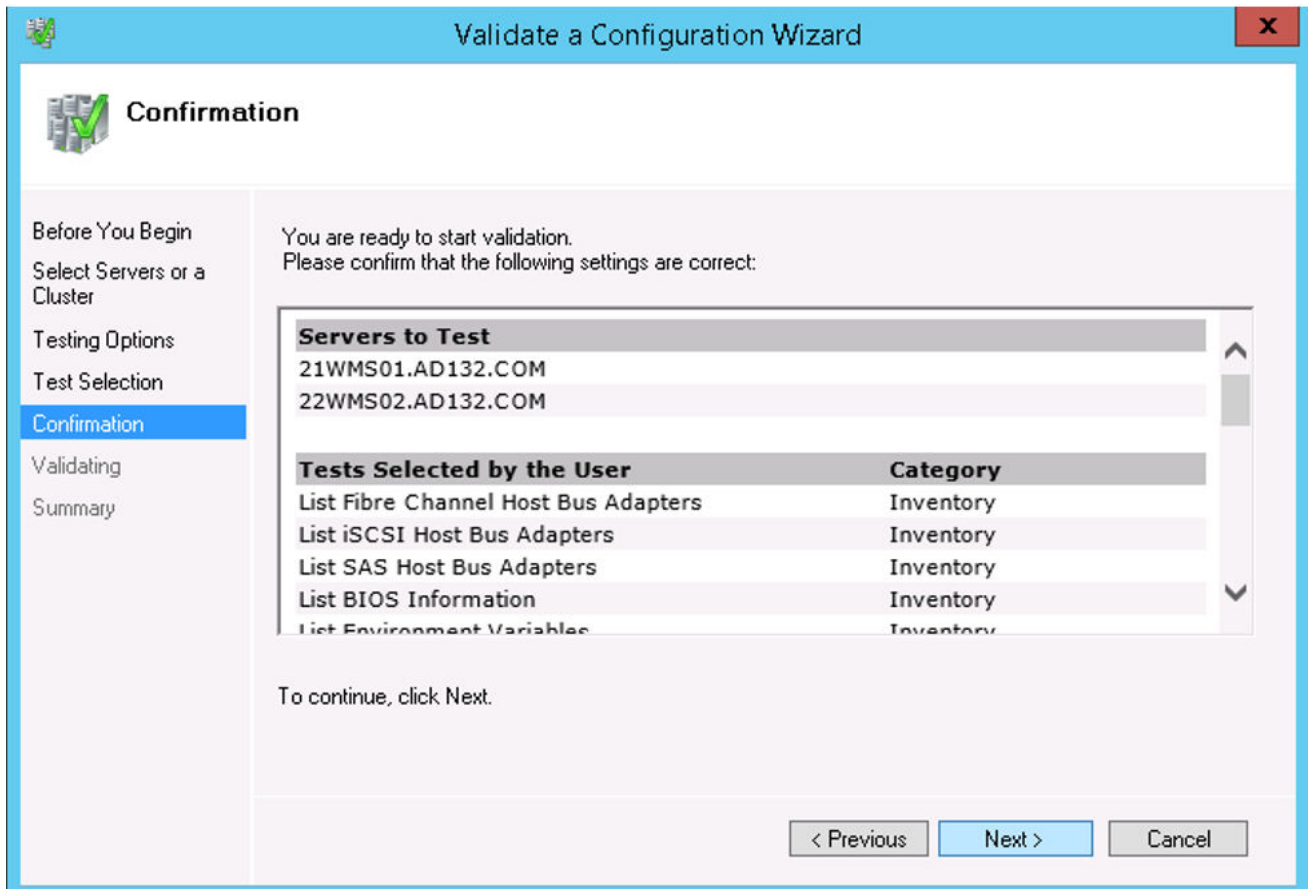


図 10. 確認

- 11 **次へ** をクリックします。**サマリ** 画面に、フェールオーバークラスタの検証レポートが表示されます。

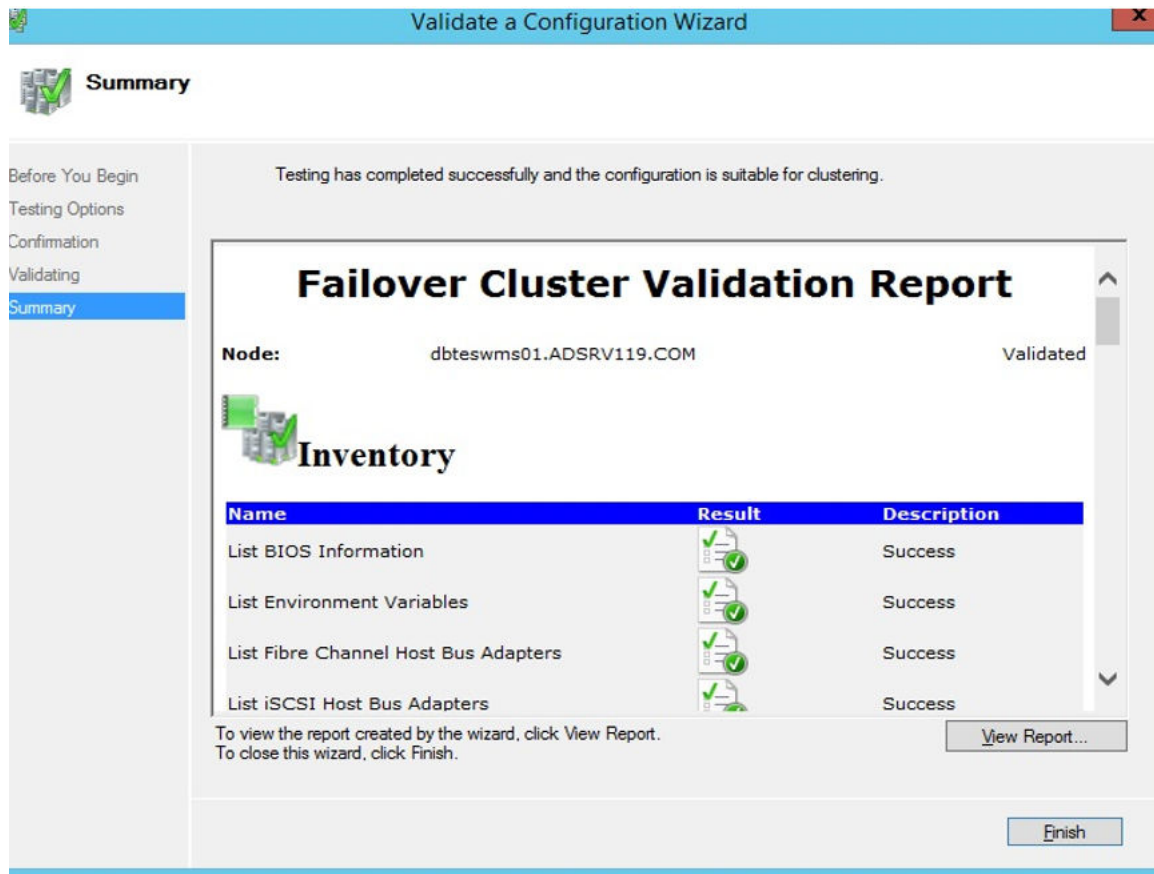


図 11. テストサマリの詳細

- 12 **レポートの表示** をクリックして、レポートを確認します。ステータスが **可** とされていた場合は、次の手順に進むことができます。ステータスが **不可** とされていた場合は、次の手順に進む前にエラーを修正する必要があります。

① **メモ:** 検証エラーがない場合は、クラスタ作成ウィザード画面が表示されます。

- 13 **次へ** をクリックして **クラスタ名** フィールドにクラスタの名前を入力し、システムの IP アドレスを選択します。
- 14 **次へ** をクリックすると **確認** 画面が表示されます。
- 15 **次へ** をクリックすると選択したすべてのクラスタノードでクラスタが作成されるので、その後で **レポートの表示** をクリックして警告メッセージを確認します。
- 16 **終了** をクリックすると、フェールオーバークラスタが作成されます。

## ファイル共有監視の作成

ファイル共有監視は、基本的なファイル共有の1つで、クラスタコンピュータが読み取り / 書き込みアクセス権を持ちます。ファイル共有の位置は、クラスタが存在する同一ドメイン中の異なる Windows Server 2012 上に存在する必要があります。

ファイル共有監視を作成するには、次の手順を実行します。

- 1 Microsoft Windows Server 2012 で **スタート** メニューを右クリックし、**サーバマネージャー** を選択して、サーバマネージャーのダッシュボードを起動します。
- 2 **サーバマネージャー** アイコンをクリックして、サーバマネージャーにアクセスします。
- 3 **ファイルサービスと記憶域サービス > 共有** に移動し、**タスク** をクリックします。
- 4 **新しい共有** をクリックします。**新しい共有ウィザード** が表示されます。
- 5 **プロファイルの選択** をクリックしてファイル共有を作成し、**次へ** をクリックします。
- 6 ファイル共有に関する共有の場所とサーバを **共有の場所** 画面で選択して、**次へ** をクリックします。
- 7 **共有名** 画面で **共有名** フィールドに名前を入力し、**確認** 画面が表示されるまで **次へ** をクリックします。

- 8 **作成** をクリックしてファイル共有を作成すると **結果の表示** 画面が表示されますが、そのステータスが **完了** となっていれば、ファイル共有監視がエラーなしで作成されたことを意味します。
- 9 **閉じる** をクリックして終了します。

## クラスタクォーラムの設定

クラスタ構成データベースは、クォーラムとも呼ばれるもので、ここにはクラスタセットアップの任意の時点でアクティブにされている必要があるサーバの詳細が含まれています。

クラスタクォーラムを設定するには、次の手順を実行します。

- 1 Microsoft Windows Server 2012 で **スタート** をクリックして **スタート** 画面を開き、**サーバマネージャー** をクリックしてサーバマネージャーダッシュボードを起動します。
- 2 **サーバマネージャー** アイコンをクリックしてサーバマネージャーにアクセスしてから、**フェールオーバークラスタマネージャー** をクリックしてクラスタマネージャーを起動します。
- 3 クラスタノードを右クリックして、**その他のアクションクラスタクォーラム設定の構成** に移動して **クラスタクォーラム構成ウィザード** 画面を表示します。
- 4 **次へ** をクリックします。**クォーラム構成オプションの選択** 画面で、**クォーラム監視を選択する** を選択します。

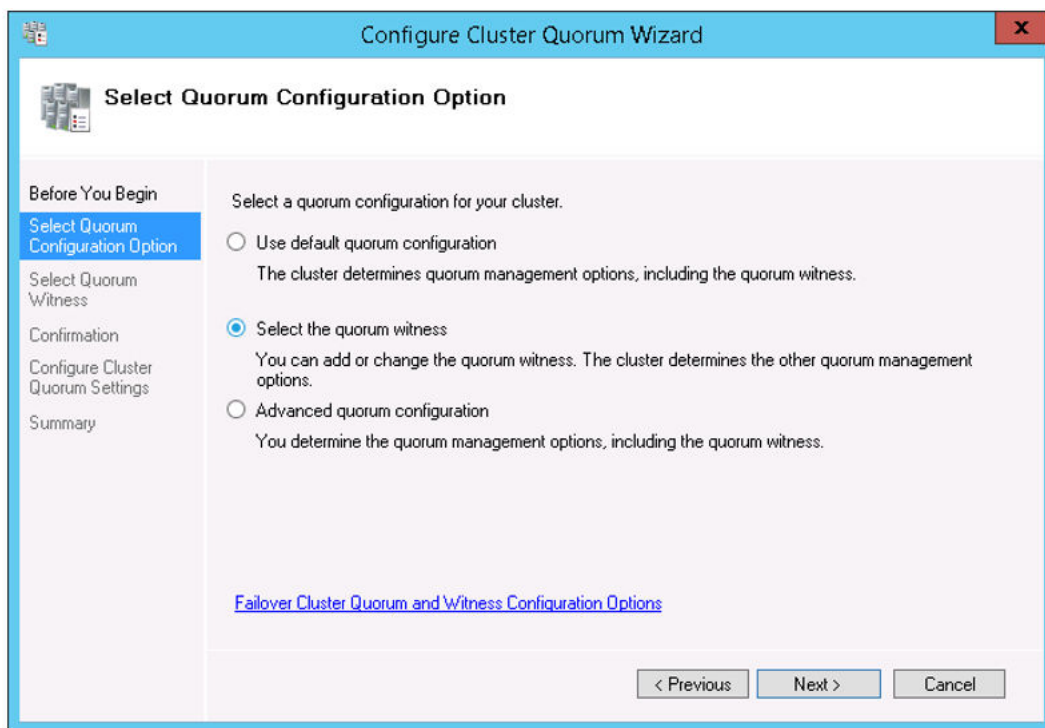


図 12. クラスタクォーラムウィザード

- 5 **次へ** をクリックします。**投票構成の選択** 画面で、**すべてのノード** を選択します。

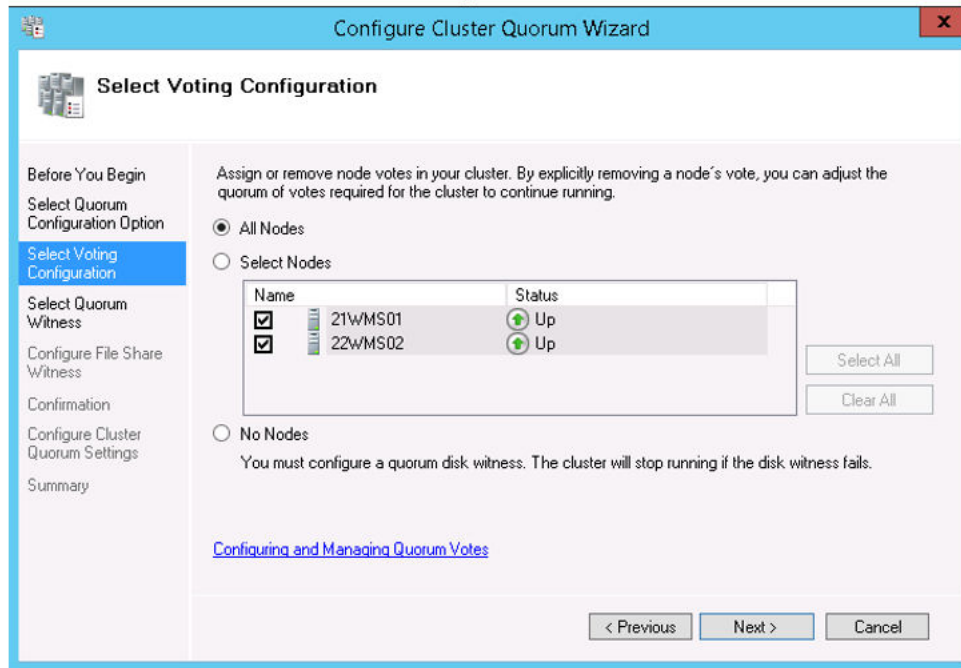


図 13. 投票設定の選択

- 6 次へをクリックします。クォーラム監視の選択 画面で **ファイル共有監視を構成する** を選択します。
- 7 次へをクリックして、**ファイル共有監視を構成する** 画面で **ファイル共有パス** フィールドに共有パスを入力します。

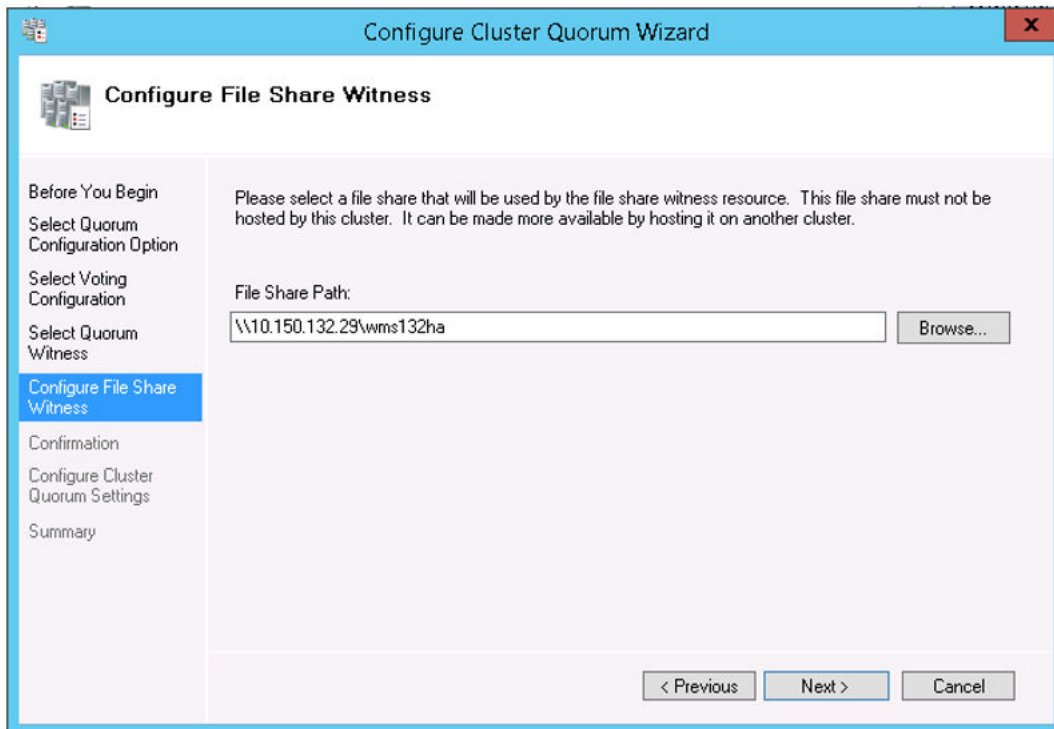


図 14. ファイル共有監視の構成

- 8 次へをクリックします。サマリ 画面に、設定されたクォーラムの内容が表示されます。

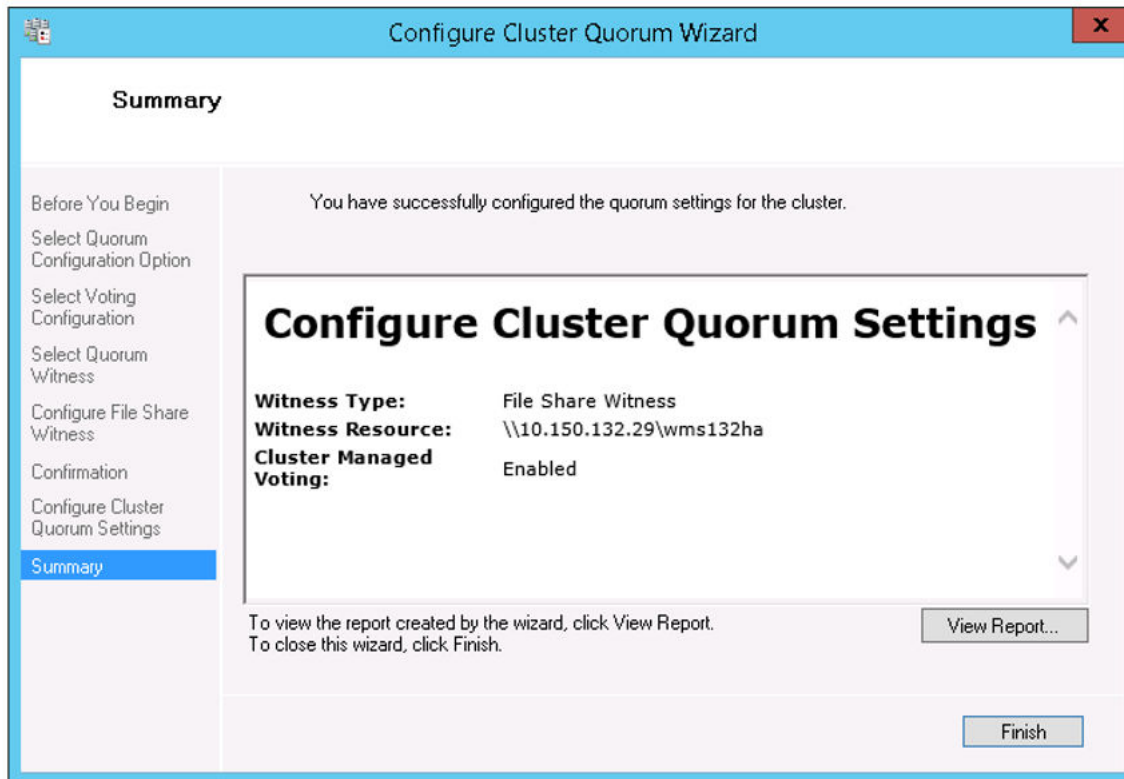


図 15. クォーラム設定のサマリ

- 9 **終了** をクリックして、新しい設定を保存します。

## クラスタ役割の作成

フェールオーバークラスタの作成後、クラスタのワークロードをホストするクラスタ化された役割の作成が可能になります。クラスタ化された役割を作成する前に、Wyse Management Suite がサーバにインストールされており、リモートデータベースをポイントしていることを確認してください。クラスタ化された役割を作成するには、次の手順を実行します。

- 1 Microsoft Windows Server 2012 で **スタート** メニューを右クリックし、**サーバマネージャー** を選択して、サーバマネージャーのダッシュボードを起動します。
- 2 **フェールオーバークラスタマネージャー** をクリックしてクラスタマネージャーを起動します。
- 3 **役割** を右クリックして **役割の構成** を選択し、**高可用性ウィザード** 画面を表示します。

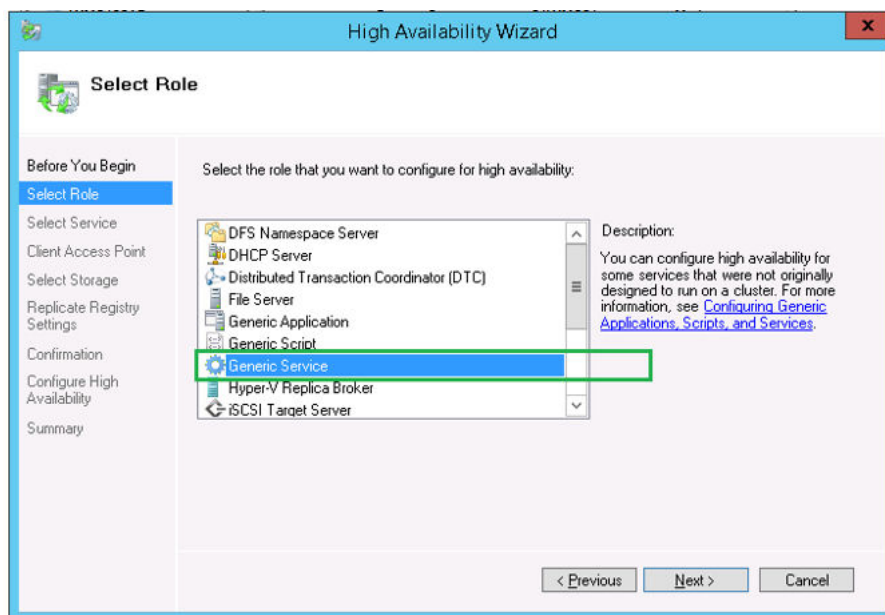


図 16. 高可用性ウィザード

- 4 汎用サービスを選択してから次へをクリックし、サービスの選択画面を表示します。

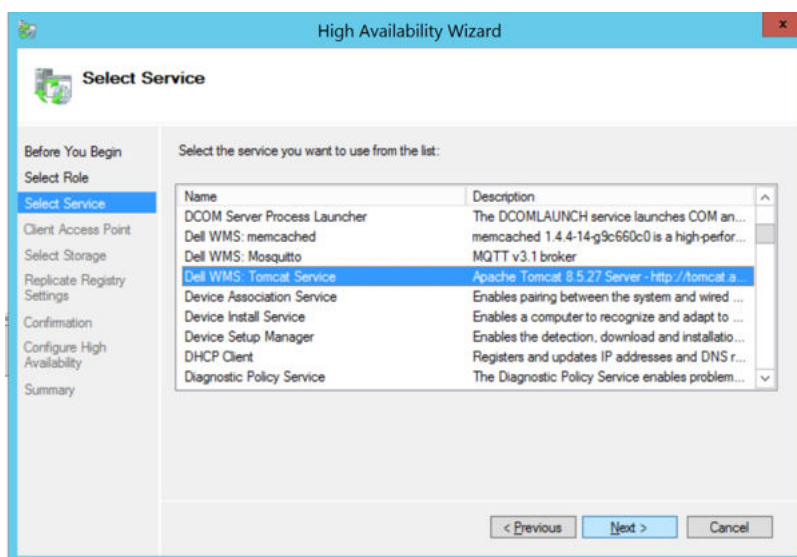


図 17. サービスの選択

- 5 Dell WMS: Tomcat Service を選択し、次へをクリックします。

① **メモ:** Wyse Management Suite バージョン 1.3 の関連サービスをクラスタに追加できるのは、Wyse Management Suite バージョン 1.3 をインストールした後だけです。

高可用性ウィザード画面が表示されますが、ここでは、クライアントアクセスポイントの作成および、Windows Server 2012 と Wyse Management Suite 間の接続を確立する必要があります。

- 6 名前 フィールドにネットワークの名前を入力して、次へをクリックします。確認画面が表示され、サーバのネットワーク名と IP アドレスの詳細が示されます。

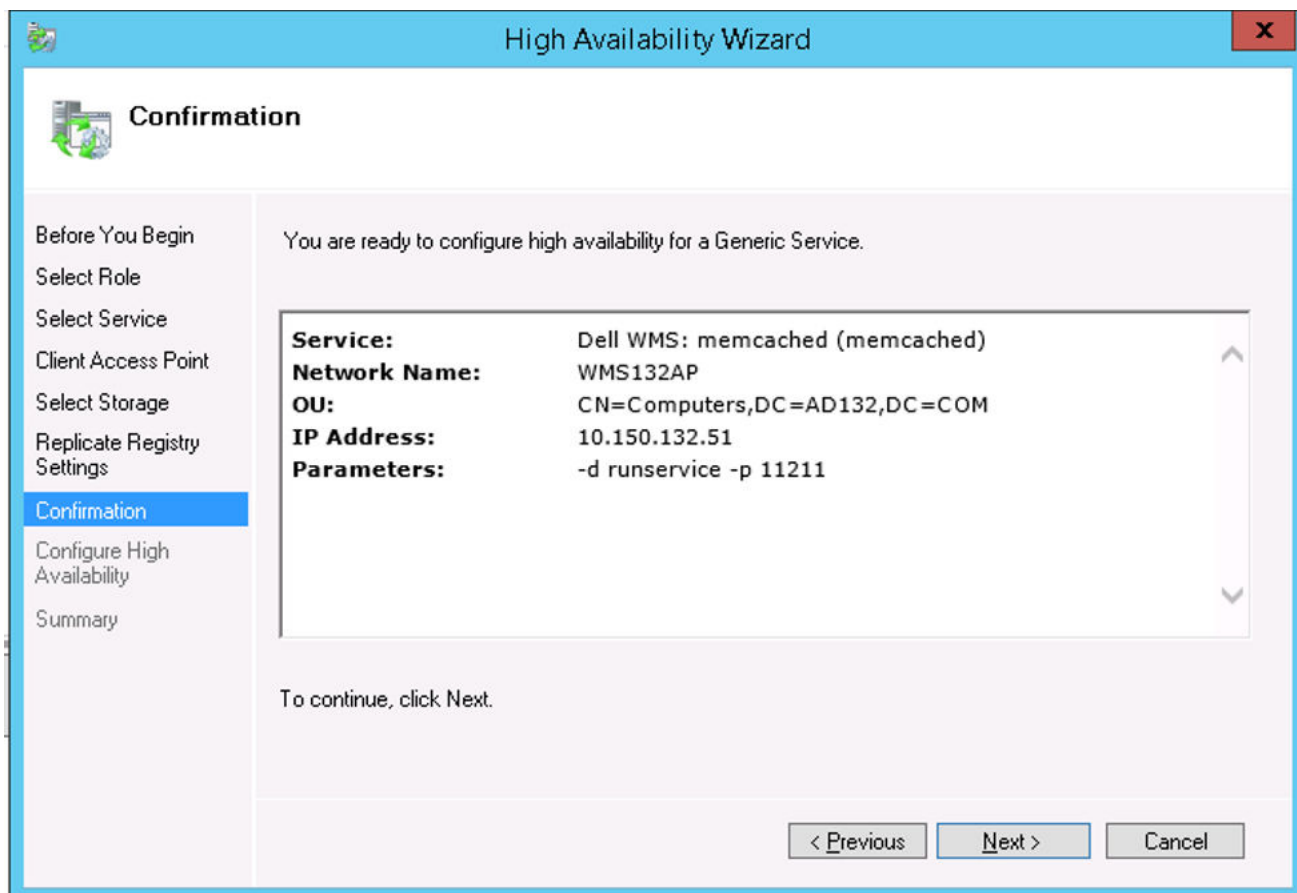


図 18. 確認

- 7 **次へ** をクリックして、プロセスを完了します。
- 8 クラスタの一部として他の Wyse Management Suite サービスを追加するには、**フェールオーバークラスタマネージャー**を起動して **アクション > 役割** に移動し、作成したネットワーク名を表示します。
- 9 ネットワーク名をクリックして、**リソースの追加 > 汎用サービス** に移動します。
- 10 下記のサービスでクラスタの一部として追加する必要があるものを **新しいリソースウィザード** 画面から選択します。
  - a Dell WMS: Mosquitto >> MQTT Broker
  - b Dell WMS: memcached
- 11 **次へ** をクリックしてタスクを完了します。  
 クラスタの一部として追加された Wyse Management Suite サービスのステータスは **実行中** と表示されます。

# MySQL InnoDB での高可用性の実現

次の手順では、MySQL InnoDB での高可用性を実現する方法について説明しています。

- 1 MySQL InnoDB サーバインスタンスを確認します - 「[MySQL InnoDB クラスタの作成](#)」を参照してください。
- 2 MySQL InnoDB にサーバまたはノードを追加します - 「[サーバまたはノードの MySQL InnoDB クラスタへの追加](#)」を参照してください。
- 3 MySQL ルーターを作成します - 「[MySQL ルーターの作成](#)」を参照してください。

## MySQL InnoDB での高可用性

MySQL InnoDB クラスタは、MySQL 用の完全な高可用性ソリューションを提供します。クライアントアプリケーションは、MySQL ルーターを使用して、プライマリノードに接続されます。プライマリノードでの障害発生時には、セカンダリノードが自動的にプライマリノードの役割を担うように昇格し、MySQL ルーターは新しいプライマリノードに向けてリクエストをルーティングします。

MySQL InnoDB クラスタのコンポーネントは、次のとおりです。

- MySQL サーバ
- MySQL ルーター

## MySQL InnoDB データベースのインストール

MySQL InnoDB のインストールについては、[dev.mysql.com](https://dev.mysql.com) を参照してください。

高可用性セットアップに適した環境の設定については、[dev.mysql.com](https://dev.mysql.com) を参照してください。

## MySQL InnoDB サーバインスタンスの確認

MySQL InnoDB をクラスタセットアップに追加する前に、MySQL InnoDB がクラスタ要件に従って作成されていることを確認します。コマンドの実行とシステムの再起動をするには、一連のコマンドを実行するごとに、**root** ユーザーとしてログインする必要があります。

次のコマンドを実行して、設定されたクラスタ要件を MySQL InnoDB サーバインスタンスが満たしていることを確認します。

① | **メモ:** ここでの IP アドレスは作業する場所での使用システムごとに異なるものであり、次に示すコマンドも単なる実行例です。

- To check that the MySQL InnoDB is created as per the requirements, run the following commands at the command prompt:
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address1')`
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address2')`
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address3')`

```

C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
MySQL Shell 8.0.11
Copyright (c) 2016, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type '\help' or '?' for help; '\quit' to exit.

MySQL JS> dba.configureLocalInstance('root@10.150.132.23:3306')
Please provide the password for 'root@10.150.132.23:3306': *****
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as 23MySQL01
Clients and other cluster members will communicate with it through this address by default. If this is not correct, the report_host MySQL system variable should be changed.

Some configuration options need to be fixed:
+-----+-----+-----+-----+
| Variable                | Current Value | Required Value | Note                                     |
+-----+-----+-----+-----+
| binlog_checksun         | CRC32         | NONE           | Update the server variable             |
| enforce_gtid_consistency | OFF          | ON             | Update read-only variable and restart the server |
| gtid_node                | OFF          | ON             | Update read-only variable and restart the server |
| log_bin                 | 0            | 1             | Update read-only variable and restart the server |
| log_slave_updates       | 0            | ON            | Update read-only variable and restart the server |
| master_info_repository  | FILE         | TABLE        | Update read-only variable and restart the server |
| relay_log_info_repository | FILE         | TABLE        | Update read-only variable and restart the server |
| transaction_write_set_extraction | OFF         | XXHASH64      | Update read-only variable and restart the server |
+-----+-----+-----+-----+

The following variable needs to be changed, but cannot be done dynamically: 'log_bin'

Detecting the configuration file...
Found configuration file at standard location: C:\ProgramData\MySQL\MySQL Server 5.7\my.ini
Do you want to modify this file? [y/N]: y
Do you want to perform the required configuration changes? [y/n]: y
Configuring instance...
The instance '10.150.132.23:3306' was configured for cluster usage.
MySQL server needs to be restarted for configuration changes to take effect.

MySQL JS> _

```

図 19. MySQL コマンドプロンプト

3つのクラスタノードでMySQL InnoDBが作成されていることを確認するには、コマンドプロンプトで次のコマンドを実行します。

- mysql-js> dba.checkInstanceConfiguration('root@IPAddress1:3306')
- mysql-js> dba.checkInstanceConfiguration('root@IPAddress2:3306')
- mysql-js> dba.checkInstanceConfiguration('root@IPAddress3:3306')

## MySQL InnoDB 用のクラスタインスタンスの作成

サーバへのMySQL InnoDB インスタンスのインストール後、クラスタインスタンスを作成します。

MySQL InnoDB のクラスタを作成するには、次の手順を実行します。

- 1 コマンドプロンプトから、管理者ユーザーとしてログインします。このユーザーアカウントには、管理権限が必要です。たとえば DBAdmin などです。次の画面は root ユーザーとしてのログイン例を示しています。

```

C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
>
>
MySQL JS> \connect root@10.150.132.23:3306
Creating a session to 'root@10.150.132.23:3306'
Enter password: *****
Fetching schema names for auto completion... Press ^C to stop.
Your MySQL connection id is 7
Server version: 5.7.22-log MySQL Community Server (GPL)
No default schema selected; type \use <schema> to set one.
MySQL [10.150.132.23] JS> _

```

図 20. ログインプロンプト

- 2 次のコマンドを実行して、固有の名前を持つクラスタを作成します。たとえば MySQLCluster などです。

```
MySQL JS> var cluster = dba.createCluster('MySQLCluster')
```

- 3 次のコマンドを実行して、クラスタのステータスを確認します。

```
MySQL JS> cluster.status()
```

作成されたクラスタのステータスが **オンライン** と表示されていれば、クラスタの作成が正常に終了していることを示します。

```
Select C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS> dba.getCluster()
<Cluster:MySQLCluster>
MySQL [10.150.132.23] JS> Cluster.status()
<
  "clusterName": "MySQLCluster",
  "defaultReplicaSet": <
    "name": "default",
    "primary": "10.150.132.23:3306",
    "ssl": "DISABLED",
    "status": "OK_NO_TOLERANCE",
    "statusText": "Cluster is NOT tolerant to any failures.",
    "topology": <
      "10.150.132.23:3306": <
        "address": "10.150.132.23:3306",
        "mode": "R/W",
        "readReplicas": <>,
        "role": "HA",
        "status": "ONLINE"
      >
    >
  >
  "groupInformationSourceMember": "mysql://root@10.150.132.23:3306"
>
MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS>
```

図 21. 確認画面

## MySQL InnoDB クラスタへのサーバインスタンスの追加

MySQL InnoDB クラスタには、プライマリまたはセカンダリとしてサーバインスタンスを追加する必要があります。

MySQL InnoDB クラスタにサーバインスタンスを追加するには、次の手順を実行します。

- 1 コマンドプロンプトから **DB Admin** ユーザーとしてログインします。
- 2 次のコマンドを実行して、サーバインスタンスを MySQL InnoDB クラスタに追加します。

```
cluster.addInstance('root@IPAddress2:3306')
```

```
cluster.addInstance('root@IPAddress3:3306')
```

① **メモ:** ここでの IP アドレスとポート番号は単なる例であり、各自の作業する場所で使われるシステムごとに異なります。

- 3 次のコマンドを実行して、サーバインスタンスのステータスを確認します。

```
cluster.status()
```

① **メモ:** すべてのノードでステータスが **オンライン** と表示されていれば、MySQL InnoDB クラスタセットアップへのノードの追加が正常に終了していることを示します。

```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
MySQL [10.150.132.231 JS] >
MySQL [10.150.132.231 JS] > var cluster = dba.getCluster()
MySQL [10.150.132.231 JS] > dba.getCluster()
<Cluster:MySQLCluster>

MySQL [10.150.132.231 JS] > Cluster.status()
<
  "clusterName": "MySQLCluster",
  "defaultReplicaSet": <
    "name": "default",
    "primary": "10.150.132.23:3306",
    "ssl": "DISABLED",
    "status": "OK",
    "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
    "topology": <
      "10.150.132.23:3306": <
        "address": "10.150.132.23:3306",
        "mode": "R/W",
        "readReplicas": <{}>,
        "role": "HA",
        "status": "ONLINE"
      >,
      "10.150.132.24:3306": <
        "address": "10.150.132.24:3306",
        "mode": "R/O",
        "readReplicas": <{}>,
        "role": "HA",
        "status": "ONLINE"
      >,
      "10.150.132.25:3306": <
        "address": "10.150.132.25:3306",
        "mode": "R/O",
        "readReplicas": <{}>,
        "role": "HA",
        "status": "ONLINE"
      >
    >
  >,
  "groupInformationSourceMember": "mysql://root@10.150.132.23:3306"
>

MySQL [10.150.132.231 JS] >
MySQL [10.150.132.231 JS] >
```

図 22. クラスタのステータス

## MySQL ルーターの設定

Wyse Management Suite と MySQL InnoDB の間の通信ネットワークは MySQL ルーターによって確立されます。MySQL ルーターをインストールするには、次の手順を実行します。

- 1 MySQL ルーターをインストールする Windows Server 2012 にログインします。詳細については、「[MySQL ルーターのインストール](#)」を参照してください。
- 2 **製品と機能の選択** 画面から **MySQL ルーター** を選択し、**インストール完了** 画面が表示されるまで **次へ** をクリックします。
- 3 `\ProgramData\MySQL\MySQL Router` ディレクトリに移動して `mysqlrouter.conf` ファイルを開き、ブートストラップクラスタのプロパティにおいて、設定したすべての MySQL サーバがクラスタセットアップに含まれているかを確認します。

```
mysqlrouter - Notepad
File Edit Format View Help
# File automatically generated during MySQL Router bootstrap
[DEFAULT]
logging_folder=C:/ProgramData/MySQL/MySQL Router/log
runtime_folder=C:/ProgramData/MySQL/MySQL Router/run
data_folder=C:/ProgramData/MySQL/MySQL Router/data
keyring_path=C:/ProgramData/MySQL/MySQL Router/data/keyring
master_key_path=C:/ProgramData/MySQL/MySQL Router/mysqlrouter.key
connect_timeout=30
read_timeout=30

[logger]
level = INFO

[metadata_cache:MySQLCluster]
router_id=2
bootstrap_server_addresses=mysql://10.150.132.23:3306,mysql://10.150.132.24:3306,mysql://10.150.132.25:3306
user=mysql_router2_oqj610zmzwp
metadata_cluster=MySQLCluster
ttl=5

[routing:MySQLCluster_default_rw]
bind_address=0.0.0.0
bind_port=6446
destinations=metadata-cache://MySQLCluster/default?role=PRIMARY
```

図 23. ブートストラップサーバアドレス

## MySQL InnoDB サーバでのデータベースとユーザーの作成

MySQL InnoDB サーバでのデータベースとユーザーアカウントの作成には管理者権限が必要です。

MySQL InnoDB サーバでデータベースを作成するには、次の SQL コマンドを実行します。

```
Create Database stratus DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_unicode_ci;
CREATE USER 'STRATUS'@'LOCALHOST';
CREATE USER 'STRATUS'@'IP ADDRESS';
SET PASSWORD FOR 'STRATUS'@'LOCALHOST' = PASSWORD <db_password>;
SET PASSWORD FOR 'STRATUS'@ <IP Address> = PASSWORD <db_password>;
GRANT ALL PRIVILEGES ON *.* TO 'STRATUS'@<IP Address> IDENTIFIED BY <db_password> WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'STRATUS'@'LOCALHOST' IDENTIFIED BY <db_password> WITH GRANT OPTION;
```

① **メモ:** IP アドレスの代わりに、Wyse Management Suite アプリケーションサーバをインストールする Network /Subnet または Multiple Single ホストエントリの Wildcard を入力することができます。

# MongoDB での高可用性の実現

次の手順では、MongoDB で高可用性を実現する方法について説明しています。

- 1 MongoDB のインストール - 「[MongoDB のインストール](#)」を参照してください。
- 2 レプリカサーバの作成 - 「[レプリカサーバの作成](#)」を参照してください。
- 3 Stratus ユーザーの作成 - 「[Stratus ユーザーアカウントの作成](#)」を参照してください。
- 4 Root ユーザーの作成 - 「[MongoDB 用の root ユーザーの作成](#)」を参照してください。
- 5 MongoDB 設定ファイルの編集 - 「[MongoDB 設定ファイルの編集](#)」を参照してください。

## MongoDB のインストール

すべての 3 つのノードに MongoDB をインストールするには、次の手順を実行します。

① **メモ:** MongoDB のインストールの詳細については、「[MongoDB のインストール](#)」を参照してください。

- 1 MongoDB インストールファイルをシステムにコピーします。
- 2 Drive C 以外のセカンダリドライブに 2 つのフォルダ `Data\log` と `data\db` を作成します。

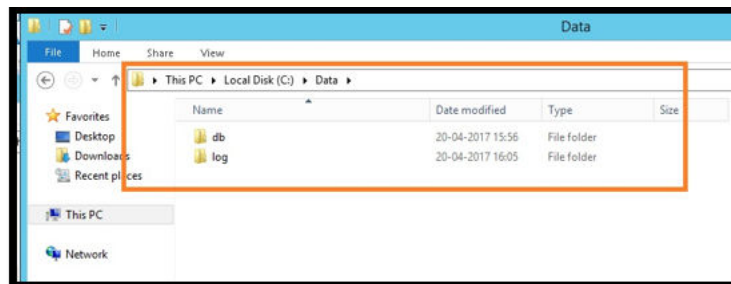


図 24. データファイル

- 3 MongoDB インストールファイルのコピー先フォルダに移動し、コマンドプロンプトから `mongod.cfg` ファイルを作成します。

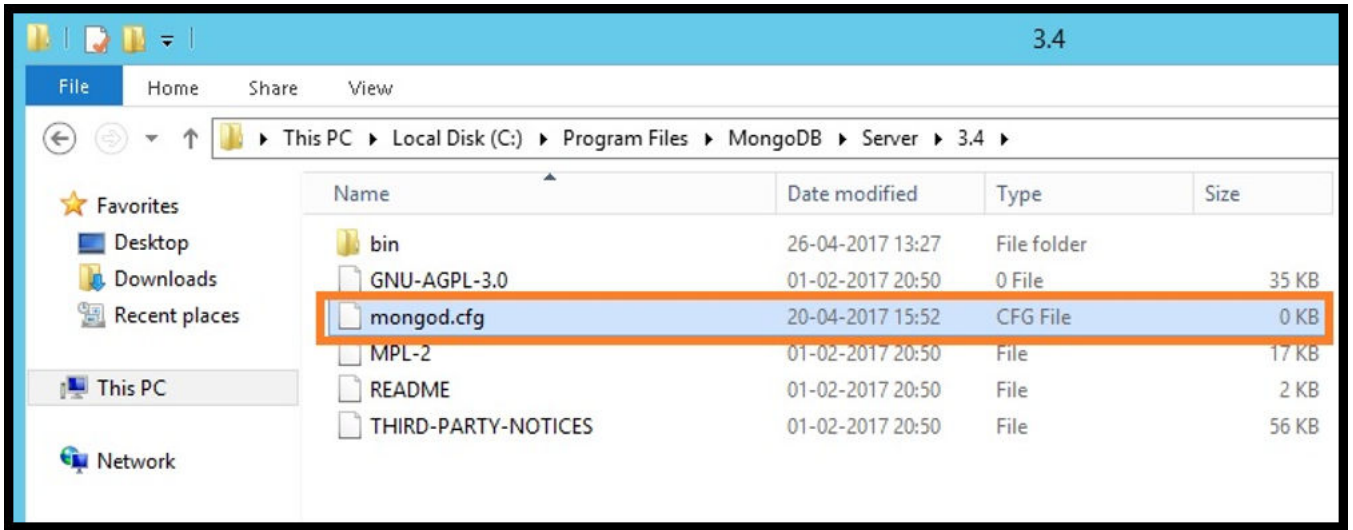


図 25. mongod.cfg ファイル

- 4 テキストエディタで mongod.cfg ファイルを開き、次のエントリを追加します。
  - a SystemLog:destination: file
  - b path: c:\data\log\mongod.log
  - c Storage: dbpath: c:\data\db
- 5 ファイルを保存します。
- 6 MongoDB サーバにログインします。
- 7 次のコマンドを実行して MongoDB サービスを開始します。
  - a C:\MongoDB\bin>.\mongod.exe --config c:\Mongodb\mongod.cfg --install
  - b C:\MongoDB\bin>net start mongod

メッセージ **MongoDB サービスを開始中です** が表示されます。

- 8 作業ディレクトリを \MongoDB\bin に変更します。
- 9 コマンドプロンプトで Mongo.exe を実行し、MongoDB インストールを完了します。

## MongoDB データベース用のレプリカサーバの作成

レプリカサーバの作成は、システム障害を回避するために必要です。レプリカサーバには、複数の分散された読み取り処理を格納できるだけの容量が必要です

レプリカサーバの作成の詳細については、[docs.mongodb.com/manual](https://docs.mongodb.com/manual) にある『レプリカサーバセットの導入』を参照してください

## データベースユーザーの作成

ユーザーを作成します (たとえば、Wyse Management Suite を使用して MongoDB にアクセスする DBUser など)。

① **メモ:** データベースのユーザー名とパスワードなどがその例ですが、ユーザーの作業する場所ごとに異なるユーザー名とパスワードを使用して作成することができます。

DBUser の作成には、次のコマンドを実行します。

```
db.createUser( {
  user: "DBUser",
  pwd: <db_password>,
  roles: [ { role: "userAdminAnyDatabase", db: "admin" },
    { role: "dbAdminAnyDatabase", db: "admin" },
    { role: "readWriteAnyDatabase", db: "admin" },
```

```
{ role: "dbOwner", db: "DBUser" } ]
})
```

## MongoDB 用の DBadmin ユーザーの作成

前のセクションで作成したユーザーアカウントを使用して MongoDB にログインします。DBAdmin ユーザーが、管理権限付きで作成されます。

DBAdmin ユーザーを作成するには、次のコマンドを実行します。

```
mongo -uDBUser -pPassword admin
use admin
db.createUser( {
  user: "DBAdmin",
  pwd: <DBAdmin user password>,
  roles: [ { role: "DBAdmin", db: "admin" } ]
})
```

## mongod.cfg ファイルの編集

MongoDB データベースのセキュリティを有効にするには mongod.cfg ファイルを編集する必要があります。

- 1 すでに作成済みの MongoDB に root ユーザーとしてログインし、次のコマンドを実行します。

```
mongo -uroot -pAdmin#123 admin
```

- 2 \data\bin\mongod.cfg ディレクトリに移動し、テキストエディタで mongod.cfg ファイルを開きます。
- 3 次に示すコマンドを用いて mongod.cfg ファイルを編集します。

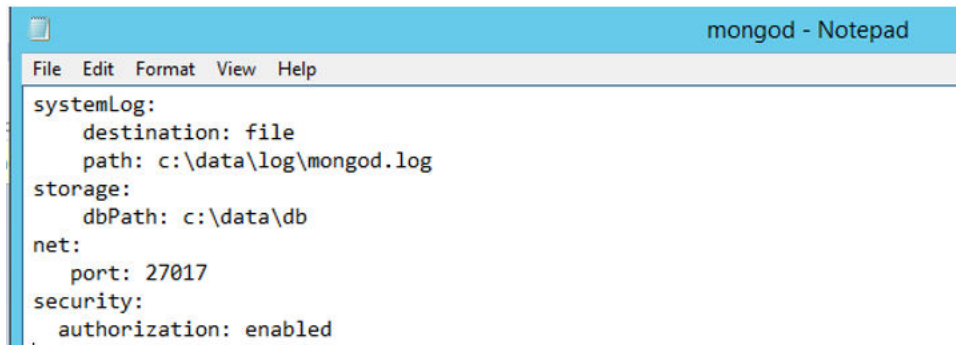


図 26. mongod.cfg の編集

```
systemLog:
destination: file
path: c:\data\log\mongod.log
storage:
dbPath: c:\data\db\Mongo
net:
port: 27017
security:
authorization: enabled
```

① | **メモ:** ポート番号は、作業する場所のシステムに応じて変わります。

- 4 mongod.cfg を保存して終了します。

## サーバでのレプリケーションの開始

Windows でファイアウォールを無効にし、Tomcat サーバを実行している場合は停止します。

- 1 すでに作成済みの MongoDB に root ユーザーとしてログインし、次のコマンドを実行します。

```
mongo -uroot -pAdmin#123 admin
```

- 2 \data\bin\mongod.cfg ディレクトリに移動し、テキストエディタで mongod.cfg ファイルを開きます。

- 3 mongod.cfg ファイルに次の 3 行を追加します。

```
keyFile: c:\data\log\mongod.key.txt
```

```
replication:
```

```
replSetName: wms
```

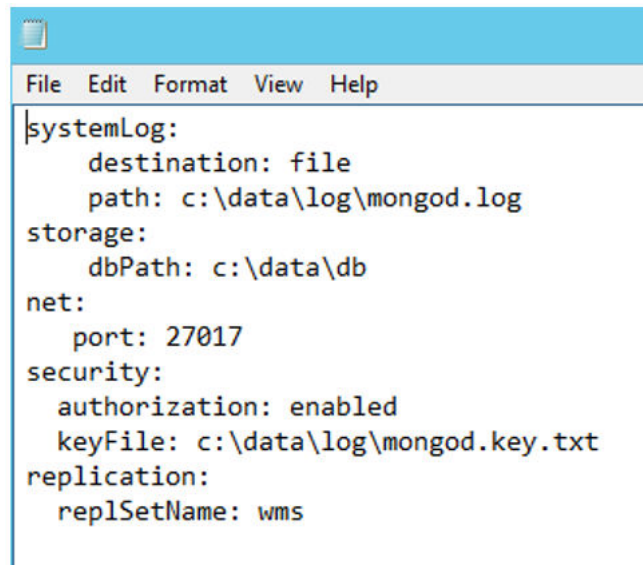


図 27. セキュリティの有効化

- 4 mongod.key.txt ファイルを作成し、3 つのサーバすべてにコピーします。

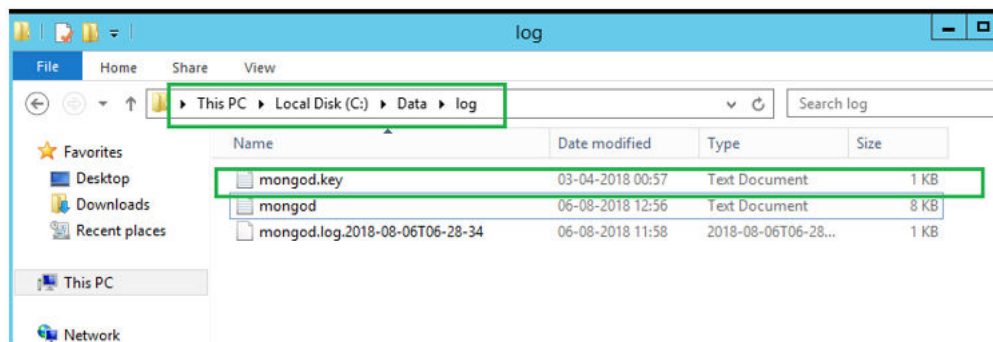


図 28. mongoal キーファイルのコピー

- 5 ファイルのコピー後、次のコマンドを実行して mongod サービスを停止します。

```
net stop mongod
```

- 6 DBadmin ユーザーを使用してログインしている MongoDB クラスタのプライマリノードでレプリケーションを開始してから、次のコマンドを実行します。

```
rs.initiate();
```

- 7 次のコマンドを実行して、レプリケーションのステータスを確認します。

```
rs.status();
```

```
rs:OTHER>
wms:PRIMARY>
wms:PRIMARY> rs.status();
{
  "set" : "wms",
  "date" : ISODate("2018-08-06T09:12:23.235Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "26MONGODB01:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 445,
      "optime" : {
        "ts" : Timestamp(1533546742, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2018-08-06T09:12:22Z"),
      "infoMessage" : "could not find member to sync from",
      "electionTime" : Timestamp(1533546710, 2),
      "electionDate" : ISODate("2018-08-06T09:11:50Z"),
      "configVersion" : 1,
      "self" : true
    }
  ],
  "ok" : 1
}
```

図 29. レプリケーションのステータス

- 8 mongod サービスを開始して、セカンダリノードを MongoDB クラスタ中の 2 番目と 3 番目のノードに追加します。

```
rs.add("IPAddress2:27017")
```

```
rs.add("IPAddress3:27017")
```

① | **メモ:** ポート番号は、ネットワークとシステムに応じて異なります。

- 9 MongoDB クラスタ内のノードを追加後、次のコマンドをプライマリとセカンダリのノードに実行して、レプリケーションのステータスを確認します。

```
rs.status();
```

```

rs:PRIMARY> rs.status();
{
  "set" : "wms",
  "date" : ISODate("2018-08-06T09:20:22.109Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "26MONGODB01:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 924,
      "optime" : {
        "ts" : Timestamp(1533547215, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
      "electionTime" : Timestamp(1533546710, 2),
      "electionDate" : ISODate("2018-08-06T09:11:50Z"),
      "configVersion" : 3,
      "self" : true
    }
  ]
}

```

PRIMARY MONGO DB Server Details

図 30. プライマリサーバのステータス

```

"configVersion" : 3,
"self" : true
}
{
  "_id" : 1,
  "name" : "10.150.132.27:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 14,
  "optime" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
  "optimeDurableDate" : ISODate("2018-08-06T09:20:15Z"),
  "lastHeartbeat" : ISODate("2018-08-06T09:20:22.007Z"),
  "lastHeartbeatRecv" : ISODate("2018-08-06T09:20:21.129Z"),
  "pingMs" : NumberLong(2),
  "syncingTo" : "26MONGODB01:27017",
  "configVersion" : 3
}
{
  "_id" : 2,
  "name" : "10.150.132.28:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 6,
  "optime" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
  "optimeDurableDate" : ISODate("2018-08-06T09:20:15Z"),
  "lastHeartbeat" : ISODate("2018-08-06T09:20:22.013Z"),
  "lastHeartbeatRecv" : ISODate("2018-08-06T09:20:21.914Z"),
  "pingMs" : NumberLong(1),
  "configVersion" : 3
}
}

```

SECONDARY MONGO DB Servers' Details

Activa

図 31. セカンダリサーバのステータス

## Teradici デバイスでの高可用性の実現

Wyse Management Suite は、Ubuntu サーバ 16.04.1 LTS でホストされている HAProxy を使用して、EMSDK サーバ間の負荷バランシングを実行します。HAProxy は、その設定に基づいて高可用性を提供することが可能な負荷バランサプロキシです。これは、TCP/HTTP 負荷バランサとして広く使われているオープンソースソフトウェアであり、Linux オペレーティングシステムで実行可能なプロキシソリューションです。最も一般的な使用目的は、複数のサーバにワークロードを分散することにより、サーバ環境のパフォーマンスと信頼性を向上させることです。

次に説明するのは、Linux オペレーティングシステムで HAProxy を使用して Teradici デバイスでの高可用性を実現する方法です。

- Wyse Management Suite での高可用性の一部として存在する Teradici サーバのインスタンスは 1 つだけです。
- Teradici デバイスのサポートには、EMSDK のインストールが必要です。EMSDK は Teradici によって提供されるソフトウェアコンポーネントで、これは Wyse Management Suite に統合されています。Wyse Management Suite インストーラによる EMSDK のインストール先は、Wyse Management Suite サーバとすることも、別のサーバとすることもできます。5000 以上のデバイスをサポートするには EMSDK のインスタンスが最低 2 つ必要であり、すべての EMSDK サーバはリモートサーバとする必要があります。
- サーバごとにインストールできる EMSDK のインスタンスは 1 つだけです。
- Teradici デバイスのサポートには、PRO ライセンスが必要です。
- Teradici の高可用性は、HAProxy を介して提供されます。
- Teradici サーバがダウンした場合、デバイスは、次の使用可能な EMSDK サーバに自動的に再接続します。

## HAProxy のインストールと設定

ThreadX 5.x デバイスのロードバランサ HAProxy は、Ubuntu Linux バージョン 16.04.1 に HAProxy バージョン 1.6 をインストールして設定します。

Ubuntu Linux システムに HAProxy をインストールして設定するには、次の手順を実行します。

- 1 Ubuntu オペレーティングシステムのインストール中に使用したユーザー認証情報を入力して、Ubuntu システムにログインします。
- 2 次のコマンドを実行して、HAProxy をインストールします。

```
sudo apt-get install software-properties-common
```

```
sudo add-apt-repository ppa:vbernat/haproxy-1.6
```

```
sudo apt-get update
```

```
sudo apt-get install haproxy
```

- 3 次のコマンドを実行して、元の設定をバックアップします。

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/ haproxy.cfg.original
```

- 4 次のコマンドを実行して、HAProxy 設定ファイルを適切なテキストエディタで編集します。

```
sudo nano /etc/haproxy/haproxy.cfg
```

設定ファイルに次のエントリを追加します。

```
Global section: Maxconn <maximum number of connections>
```

```
Frontend tcp-in: bind :5172
```

```
Back end servers: server :5172
```

```
maxconn <maximum number of connections per Teradici device proxy server>
```

① **メモ:** シームレスなフェールオーバーを実行させるには、クライアント容量の合計を上回るバックエンドサーバを管理者が追加する必要があります。

- 5 CTRL+Oを押して、haproxy.cfg ファイルに変更を保存します。

次のテキストは、サンプルの HAProxy 設定ファイルです。

```
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    daemon
    #maxconn is maximum allowed connections
    maxconn 60000
defaults
    log          global
    mode         tcp
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend fe_teradici_5172
    bind :5172
    mode tcp
    backlog 4096
    maxconn 70000
    default_backend be_teradici_5172

backend be_teradici_5172
    mode tcp
    option log-health-checks
    option tcplog
    balance leastconn
    server emsdk1 :5172 check server emsdk2 5172 check : timeout queue 5s timeout server 86400s
    option srvtcpka

#frontend fe_teradici_5172
#replace IP with IP of your Linux proxy machine bind Eg: 10.150.105.119:5172

#default_backend servers

#backend servers
#Add your multiple back end windows machine ip with 5172 as port
# maxconn represents number of connection- replace 10 with limit # (below 20000)
# "server1" "server2" are just names and not keywords

#server server1 10.150.105.121:5172 maxconn 20000 check
#server server2 10.150.105.124:5172 maxconn 20000 check
```

- 6 次のコマンドを実行して、HAProxy 設定を検証します。

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

設定が有効である場合は、**設定は有効です** というメッセージが表示されます。

- 7 次のコマンドを実行して、HAProxy サービスを再起動します。

```
Sudo service haproxy restart
```

- 8 次のコマンドを実行して、HAProxy を停止します。

```
serviceSudo service haproxy stop
```

# Wyse Management Suite サーバのインストール

Wyse Management Suite サーバをインストールする前に、次のコンポーネントが設定されていることを確認します。

- 2つのノード上での Windows フェールオーバークラスタ
- レプリカセットで実行中の MongoDB サーバ
- 実行中の MySQL InnoDB クラスタセットアップ
- 2つのノードにインストールされた MySQL ルーター

Wyse Management Suite サーバをインストールするには、次の手順を実行します。

- 1 Wyse Management Suite v1.3 のインストーラ画面を起動します。
- 2 **カスタムタイプ**と **Teradici EMSDK** を選択し、**次へ** をクリックします。
- 3 **外部 MongoDB** オプションを選択します ( 作成したレプリカセットを含む MongoDB クラスタ )。これに該当するのは、wms などです。リモートのプライマリ Mongo DB サーバ情報、ポート番号、および MongoDB のユーザー名とパスワードを該当するフィールドに入力して、**次へ** をクリックします。
- 4 MySQL 用の **外部 MariaDB** オプションを選択します。MySQL ルーターのアドレスを使用します ( Wyse Management Suite サーバノードにインストールされている場合はローカルホスト )。

① **メモ:** Stratus ユーザーアカウントが MySQL サーバ上に作成されていることを確認します。

- 5 MySQL ルーターの情報を **外部 Maria DB サーバ** のフィールドにポート番号で入力します。最初に作成した MySQL データベースのユーザーアカウント情報を入力します。**ポートの選択** 画面が表示され、ポートの詳細が示されます。このポートは MySQL ルーターによって使用されるものです。デフォルトポートは 6466 です。ただし、このポート番号は変更することもできます。
- 6 管理権限を持つユーザー名と電子メールアドレスを入力しますが、その際には Teradici EMSDK ポート番号と CIFS ユーザーアカウントの情報も必要です。
- 7 ローカルリポジトリのインストール先フォルダのパスおよび共有 UNC パスを入力して、**次へ** をクリックします。**インストールは正常に終了しました** というメッセージが表示されます。

① **メモ:** 共有 UNC パスは、Wyse Management Suite アプリケーションがインストールされている Windows Server の両方から外れている必要があります。

① **メモ:** ノード 2 に Wyse Management Suite アプリケーションをインストールする前に、ノード 1 へのインストール中に作成された Wyse Management のローカルリポジトリから Data フォルダを削除しておいてください。共有された UNC WMS ローカルリポジトリパスからの Data フォルダの削除後に、Windows クラスタのノード 2 への Wyse Management Suite アプリケーションのインストールが可能になります。

# Wyse Management Suite の Windows Server 2012 へのインストール

プライベートクラウドに Wyse Management Suite をインストールするには、次の操作を行います。

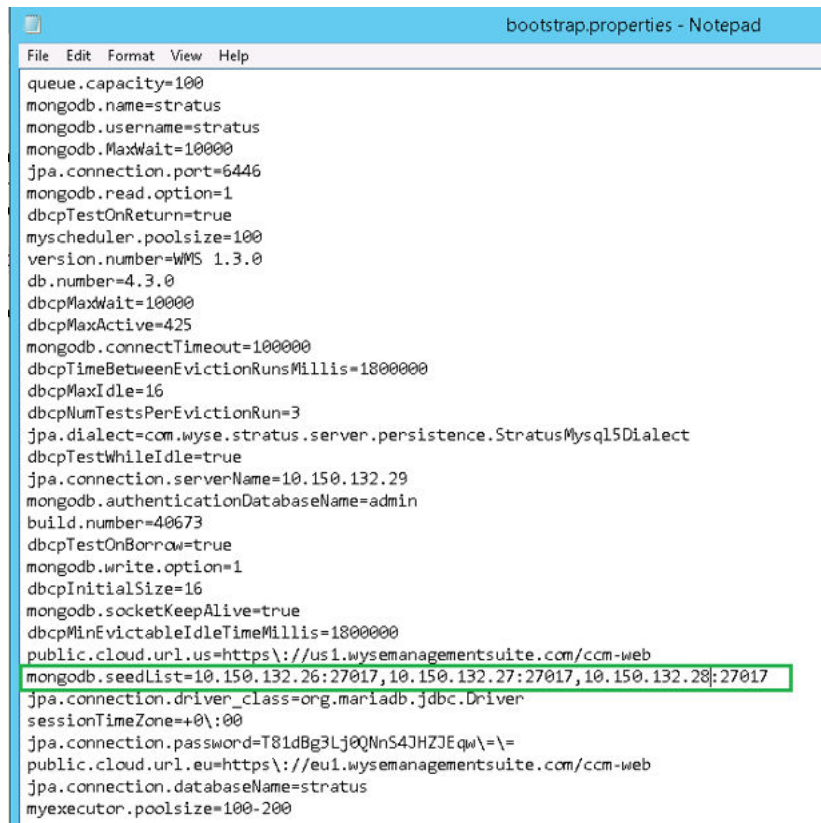
- 1 インストーラパッケージをダブルクリックします。
- 2 **ようこそ** 画面のライセンス契約を読み、**次へ** をクリックします。
- 3 インストールする **セットアップタイプ** を選択して、**次へ** をクリックします。利用できるオプションは次のとおりです。
  - 通常：必要なユーザー操作は最低限であり、組み込みデータベースをインストールします。
  - カスタム：最大限のユーザー操作を必要とする上級ユーザー向けです。
- 4 **セットアップタイプ** で **カスタム** を選択して、**次へ** をクリックします。  
**Mongo データベースサーバ** ページが表示されます。
- 5 **外部 Mongo DB** オプションを選択します。ユーザー名、パスワード、データベースサーバの詳細情報、ポートの詳細情報を入力して、**次へ** をクリックします。  
**① | メモ:** ポートフィールドにはデフォルトのポートが入力されますが、この値は変更できます。
- 6 **インストールは正常に終了しました** というメッセージが表示されるまで **次へ** をクリックします。  
**① | メモ:** サーバまたはノード 2 に Wyse Management Suite をインストールする前に、インストール中にサーバまたはノード 1 に作成されたロケールリポジトリから \Data フォルダを削除しておいてください。

## インストール後のチェック

Wyse Management Suite バージョン 1.3 での高可用性を確認するには、次の手順を実行します。

- Wyse Management Suite 管理ポータルを起動し、ウェブインタフェースを使用してログインができるかを確認します。
- MongoDB 用の `\Dell\WMS\Tomcat-8\webapps\ccm-web\WEB-INF\classes` フォルダ中で、Tomcat サーバにある `bootstrap.properties` ファイルを次のように編集します。

```
mongodb.seedList = MongoDBServer1_IP:27017, MongoDBServer2_IP:27017, MongoDBServer3_IP:27017
```



```
bootstrap.properties - Notepad
File Edit Format View Help
queue.capacity=100
mongodb.name=stratus
mongodb.username=stratus
mongodb.MaxWait=10000
jpa.connection.port=6446
mongodb.read.option=1
dbcpTestOnReturn=true
myscheduler.poolsize=100
version.number=WMS 1.3.0
db.number=4.3.0
dbcpMaxWait=10000
dbcpMaxActive=425
mongodb.connectTimeout=100000
dbcpTimeBetweenEvictionRunsMillis=1800000
dbcpMaxIdle=16
dbcpNumTestsPerEvictionRun=3
jpa.dialect=com.wyse.stratus.server.persistence.StratusMySQL5Dialect
dbcpTestWhileIdle=true
jpa.connection.serverName=10.150.132.29
mongodb.authenticationDatabaseName=admin
build.number=40673
dbcpTestOnBorrow=true
mongodb.write.option=1
dbcpInitialSize=16
mongodb.socketKeepAlive=true
dbcpMinEvictableIdleTimeMillis=1800000
public.cloud.url.us=https://us1.wysemanagementsuite.com/ccm-web
mongodb.seedList=10.150.132.26:27017,10.150.132.27:27017,10.150.132.28:27017
jpa.connection.driver_class=org.mariadb.jdbc.Driver
sessionTimeZone=+0\:00
jpa.connection.password=T81dBg3Lj0QnN54JHZJEqw\=\=
public.cloud.url.eu=https://eu1.wysemanagementsuite.com/ccm-web
jpa.connection.databaseName=stratus
myexecutor.poolsize=100-200
```

図 32. bootstrap.properties ファイルの編集

- MongoDB にログインし、`bootstrapProperties` テーブルで Windows Cluster Virtual IP/Hostname of Access Point とされているものを、次の属性値に基づいて更新します。

```
Stratusapp.server.url
Stratus.external.mqtt.url
Memcached.Servers
Mqtt.server.url
```

MongoDB テーブルを変更するには、次の手順を実行します。

- 1 Stratus データベースで **Collections** にアクセスし、**bootstrapProperties** テーブルを選択します。
- 2 MySQL テーブルを更新し、両方のノードで Tomcat を再起動します。mysql データベーステーブルを手動で更新し、**ServersInCluster** テーブル中の **ServerIp** をアクティブに保持するよう、次のコマンドを実行します。

```
Update serversInCluster set ServerIp = '<VIP address of Windows Cluster>';
```

① | **メモ:** `serversInCluster` テーブル中にレコードが 1 つだけあることを確認し、複数のレコードがある場合は余分なレコードを削除します。

3 `Update queueLock set IpInLock = '<VIP address of Windows Cluster>';`

## トラブルシューティング

このセクションでは、クラスタのセットアップに関する Wyse Management Suite バージョン 1.3 のトラブルシューティング情報について説明します。

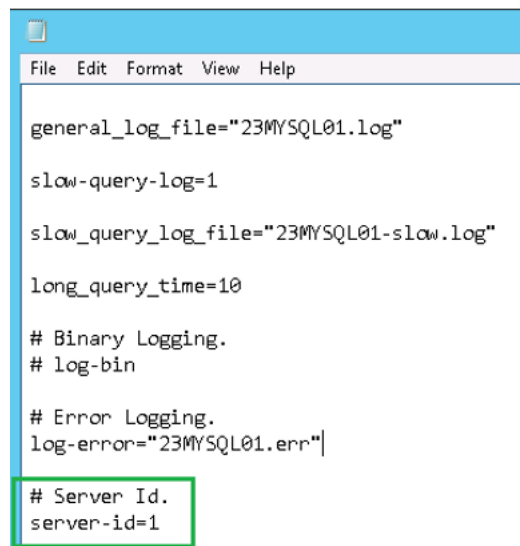
- 問題：サーバインストール関係の問題をチェックするために必要な Wyse Management Suite のログファイルの位置が不明です。  
回避策：ログファイルは %temp% WMSInstall.log フォルダにあります。
- 問題：アプリケーション関係の問題をチェックするために必要な Tomcat サービスに関するログファイルの位置が不明です。  
回避策：ログファイルは \Program Files\DELL\WMS\Tomcat-8\stratus.log フォルダにあります。
- 問題：クラスタ内のいずれかのサーバまたはノードが動作を停止しましたが、それは MySQL InnoDB クラスタの一部ではありません。

回避策：コマンドプロンプトで以下の手順を実行します。

```
var cluster = dba.rebootClusterFromCompleteOutage(); #Reboot the cluster instance
dba.configureLocalInstance('root@Server_IPAddress:3306') #Reconfigure the local instance
cluster.addInstance('root@Server_IPAddress:3306') #Add the cluster instance back to the network
```

- 問題：MySQL InnoDB クラスタにサーバ ID を追加しないと、エラーメッセージが表示されます。

回避策：\ProgramData\MySQL\MySQL Server 5.7 ディレクトリにある my.conf ファイル中のサーバ ID エントリを変更します。



```
File Edit Format View Help

general_log_file="23MySQL01.log"

slow-query-log=1

slow_query_log_file="23MySQL01-slow.log"

long_query_time=10

# Binary Logging.
# log-bin

# Error Logging.
log-error="23MySQL01.err"

# Server Id.
server-id=1
```

図 33. サーバ ID の変更