Red Hat Enterprise Linux 9

Microsoft Azure への RHEL 9 のデプロイ

RHEL システムイメージの取得と Azure 上での RHEL インスタンスの作成

Last Updated: 2024-04-18

Red Hat Enterprise Linux 9 Microsoft Azure への RHEL 9 のデプロイ

RHEL システムイメージの取得と Azure 上での RHEL インスタンスの作成

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Red Hat Enterprise Linux (RHEL) をパブリッククラウド環境で使用するには、RHEL システムイメージを作成し、Microsoft Azure などのさまざまなクラウドプラットフォームにデプロイできます。Azure 上で Red Hat High Availability (HA) クラスターを作成および設定することもできます。次の章では、Azure 上でクラウド RHEL インスタンスと HA クラスターを作成する手順について説明します。これらのプロセスには、必要なパッケージとエージェントのインストール、フェンシングの設定、ネットワークリソースエージェントのインストールが含まれます。

目次

多様性を受け入れるオープンソースの強化 3

RED HAT ドキュメントへのフィードバック (英語のみ) 4

第1章 パブリッククラウドプラットフォームでの RHEL の導入 5
1.1. パブリッククラウドで RHEL を使用する利点 5
1.2. RHEL のパブリッククラウドのユースケース 6
1.3. パブリッククラウドへの移行時によくある懸念事項 6
1.4. パブリッククラウドデプロイメント用の RHEL の入手 7
1.5. RHEL クラウドインスタンスを作成する方法 8

第2章 VHD イメージを MICROSOFT AZURE クラウドにプッシュする 10

第3章 MICROSOFT AZURE での仮想マシンとして RED HAT ENTERPRISE LINUX イメージをデプロイ 13
3.1. AZURE での RED HAT ENTERPRISE LINUX イメージオプション 13
3.2. ベースイメージの理解 14
3.3. MICROSOFT AZURE のカスタムベースイメージの設定 17
3.4. イメージの固定 VHD 形式への変換 20
3.5. AZURE CLI のインストール 22
3.6. AZURE でのリソースの作成 22
3.7. AZURE イメージのアップロードおよび作成 26
3.8. AZURE での仮想マシンの作成および起動 27
3.9. その他認証方法 28
3.10. RED HAT サブスクリプションの割り当て 29
3.11. AZURE GOLD IMAGE の自動登録の設定 29
3.12. 関連情報 31

第4章 MICROSOFT AZURE での RED HAT HIGH AVAILABILITY クラスターの設定 32
4.1. パブリッククラウドプラットフォームで高可用性クラスターを使用する利点 32
4.2. AZURE でのリソースの作成 33
4.3. 高可用性に必要なシステムパッケージ 37
4.4. AZURE VM 設定 37
4.5. HYPER-V デバイスドライバーのインストール 38
4.6. MICROSOFT AZURE のデプロイメントに必要な設定変更を行う 39
4.7. AZURE ACTIVE DIRECTORY アプリケーションの作成 42
4.8. イメージの固定 VHD 形式への変換 43
4.9. AZURE イメージのアップロードおよび作成 45
4.10. RED HAT HA パッケージおよびエージェントのインストール 46
4.11. クラスターの作成 47
4.12. フェンシングの概要 48
4.13. フェンスデバイスの作成 49
4.14. AZURE 内部ロードバランサーの作成 51
4.15. ロードバランサーリソースエージェントの設定 52
4.16. 共有ブロックストレージの設定 53
4.17. 関連情報 57

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. Jira の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 パブリッククラウドプラットフォームでの RHEL の導入

パブリッククラウドプラットフォームは、コンピューティングリソースをサービスとして提供します。オンプレミスのハードウェアを使用する代わりに、Red Hat Enterprise Linux (RHEL) システムなどの IT ワークロードをパブリッククラウドインスタンスとして実行できます。

1.1. パブリッククラウドで RHEL を使用する利点

パブリッククラウドプラットフォーム上に配置されたクラウドインスタンスとしての RHEL には、RHEL オンプレミスの物理システムまたは仮想マシン (VM) に比べて次の利点があります。

- **リソースの柔軟性と詳細な割り当て**
  RHEL のクラウドインスタンスは、クラウドプラットフォーム上の仮想マシンとして実行されます。この仮想マシンは通常、クラウドサービスのプロバイダーによって維持管理されるリモートサーバーのクラスターです。したがって、特定のタイプの CPU やストレージなどのハードウェアリソースのインスタンスへの割り当ては、ソフトウェアレベルで行われ、簡単にカスタマイズできます。

  また、ローカルの RHEL システムと比較すると、物理ホストの機能によって制限されることがありません。むしろ、クラウドプロバイダーが提供する選択肢に基づいて、さまざまな機能から選択できます。

- **領域とコスト効率**
  クラウドワークロードをホストするためにオンプレミスサーバーを所有する必要がありません。これにより、物理ハードウェアに関連するスペース、電力、メンテナンスの要件が回避されます。

  代わりに、パブリッククラウドプラットフォームでは、クラウドインスタンスの使用料をクラウドプロバイダーに直接支払います。通常、コストはインスタンスに割り当てられたハードウェアとその使用時間に基づきます。したがって、要件に基づいてコストを最適化できます。

- **ソフトウェアで制御される設定**
  クラウドインスタンスの設定全体がクラウドプラットフォーム上にデータとして保存され、ソフトウェアによって制御されます。したがって、インスタンスの作成、削除、クローン作成、または移行を簡単に行うことができます。また、クラウドインスタンスは、クラウドプロバイダーのコンソールでリモートで操作され、デフォルトでリモートストレージに接続されます。

  さらに、クラウドインスタンスの現在の状態をいつでもスナップショットとしてバックアップできます。その後、スナップショットをロードしてインスタンスを保存した状態に復元できます。

- **ホストからの分離とソフトウェアの互換性**
  ローカルの仮想マシンと同様に、クラウドインスタンス上の RHEL ゲストオペレーティングシステムは仮想化されたカーネル上で実行されます。このカーネルは、ホストオペレーティングシステムや、インスタンスへの接続に使用する **クライアント** システムとは別のものです。

  したがって、任意のオペレーティングシステムをクラウドインスタンスにインストールできます。つまり、RHEL パブリッククラウドインスタンスでは、ローカルオペレーティングシステムでは使用できない RHEL 固有のアプリケーションを実行できます。

  さらに、インスタンスのオペレーティングシステムが不安定になったり侵害されたりした場合でも、クライアントシステムには一切影響がありません。

関連情報

- パブリッククラウドとは
- ハイパースケーラーとは
- クラウドコンピューティングの種類
- RHEL のパブリッククラウドのユースケース
- パブリッククラウドデプロイメント用の RHEL の入手

1.2. RHEL のパブリッククラウドのユースケース

パブリッククラウドへのデプロイには多くの利点がありますが、すべてのシナリオにおいて最も効率的なソリューションであるとは限りません。RHEL デプロイメントをパブリッククラウドに移行するかどうかを評価している場合は、ユースケースがパブリッククラウドの利点を享受できるかどうかを検討してください。

有益なユースケース

- パブリッククラウドインスタンスのデプロイは、デプロイメントのアクティブなコンピューティング能力を柔軟に増減する (**スケールアップ** および **スケールダウン** とも呼ばれます) 場合に非常に効果的です。したがって、次のシナリオではパブリッククラウドで RHEL を使用することを推奨します。
  - ピーク時のワークロードが高く、一般的なパフォーマンス要件が低いクラスター。要求に応じてスケールアップおよびスケールダウンすることで、リソースコストの面で高い効率が得られる場合があります。
  - クラスターを迅速にセットアップまたは拡張できます。これにより、ローカルサーバーのセットアップにかかる高額な初期費用が回避されます。
- クラウドインスタンスは、ローカル環境で何が起こっても影響を受けません。したがって、バックアップや障害復旧に使用できます。

問題が発生する可能性のあるユースケース

- 調整不可能な既存の環境を運用している場合。既存のデプロイメントの特定のニーズに合わせてクラウドインスタンスをカスタマイズすることは、現在のホストプラットフォームと比較して費用対効果が低い可能性があります。
- 厳しい予算制限の中で運用している場合。通常、ローカルデータセンターでデプロイメントを維持管理すると、パブリッククラウドよりも柔軟性は低くなりますが、最大リソースコストをより細かく制御できます。

次のステップ

- パブリッククラウドデプロイメント用の RHEL の入手

関連情報

- アプリケーションをクラウドに移行すべきか?また、その決定方法

1.3. パブリッククラウドへの移行時によくある懸念事項

RHEL ワークロードをローカル環境からパブリッククラウドプラットフォームに移行すると、それに伴う変更について懸念が生じる可能性があります。よくある質問は次のとおりです。

RHEL は、クラウドインスタンスとして動作する場合、ローカル仮想マシンとして動作する場合とは異なる動作になりますか?

パブリッククラウドプラットフォーム上の RHEL インスタンスは、ほとんどの点で、オンプレミスサーバーなどのローカルホスト上の RHEL 仮想マシンと同じように機能します。注目すべき例外には次のようなものがあります。

- パブリッククラウドインスタンスは、プライベートオーケストレーションインターフェイスの代わりに、プロバイダー固有のコンソールインターフェイスを使用してクラウドリソースを管理します。
- ネストされた仮想化などの特定の機能が正しく動作しない可能性があります。特定の機能がデプロイメントにとって重要な場合は、選択したパブリッククラウドプロバイダーとその機能の互換性を事前に確認してください。

ローカルサーバーと比べて、パブリッククラウドではデータは安全に保たれますか?

RHEL クラウドインスタンス内のデータの所有権はユーザーにあり、パブリッククラウドプロバイダーはデータにアクセスできません。さらに、主要なクラウドプロバイダーは転送中のデータ暗号化をサポートしているため、仮想マシンをパブリッククラウドに移行する際のデータのセキュリティーが向上します。

RHEL パブリッククラウドインスタンスの一般的なセキュリティーは次のように管理されます。

- パブリッククラウドプロバイダーは、クラウドハイパーバイザーのセキュリティーを担当します。
- Red Hat は、RHEL ゲストオペレーティングシステムのセキュリティー機能をインスタンスに提供します。
- ユーザーは、クラウドインフラストラクチャーにおける特定のセキュリティー設定とプラクティスを管理します。

ユーザーの地理的リージョンは、RHEL パブリッククラウドインスタンスの機能にどのように影響しますか?

RHEL インスタンスは、地理的な場所に関係なく、パブリッククラウドプラットフォームで使用できます。したがって、オンプレミスサーバーと同じリージョンでインスタンスを実行できます。

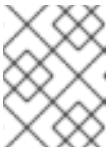
ただし、物理的に離れたリージョンでインスタンスをホストすると、操作時に待ち時間が長くなる可能性があります。さらに、パブリッククラウドプロバイダーによっては、特定のリージョンで、追加機能が提供される場合や、より高いコスト効率が得られる場合があります。RHEL インスタンスを作成する前に、選択したクラウドプロバイダーで利用可能なホスティングリージョンのプロパティーを確認してください。

1.4. パブリッククラウドデプロイメント用の RHEL の入手

RHEL システムをパブリッククラウド環境にデプロイするには、以下を行う必要があります。

1. 要件と市場の現在のオファーに基づいて、ユースケースに最適なクラウドプロバイダーを選択します。
   現在、RHEL インスタンスの実行が認定されているクラウドプロバイダーは次のとおりです。
   - Amazon Web Services (AWS)
   - Google Cloud Platform (GCP)

- Microsoft Azure



注記

このドキュメントでは、Microsoft Azure への RHEL のデプロイについて特に説明します。

2. 選択したクラウドプラットフォーム上に RHEL クラウドインスタンスを作成します。詳細は、RHEL クラウドインスタンスを作成する方法 を参照してください。

3. RHEL デプロイメントを最新の状態に保つには、Red Hat Update Infrastructure (RHUI) を使用します。

関連情報

- RHUI ドキュメント
- Red Hat Open Hybrid Cloud

1.5. RHEL クラウドインスタンスを作成する方法

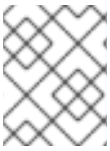
RHEL インスタンスをパブリッククラウドプラットフォームにデプロイするには、次のいずれかの方法を使用できます。

RHEL のシステムイメージを作成し、クラウドプラットフォームにインポートします。

- システムイメージを作成するには、RHEL Image Builder を使用するか、イメージを手動で構築します。
- これは既存の RHEL サブスクリプションを使用する方法で、**bring your own subscription** (BYOS) とも呼ばれます。
- 年間サブスクリプションを前払いすると、Red Hat お客様割引を利用できます。
- カスタマーサービスは Red Hat によって提供されます。
- 複数のイメージを効率的に作成するには、**cloud-init** ツールを使用できます。

RHEL インスタンスをクラウドプロバイダーマーケットプレイスから直接購入します。

- サービスの利用に対して時間料金を後払いで支払います。したがって、この方法は **従量課金制** (PAYG) とも呼ばれます。
- カスタマーサービスはクラウドプラットフォームプロバイダーによって提供されます。



注記

さまざまな方法を使用して Microsoft Azure に RHEL インスタンスをデプロイする詳細な手順については、このドキュメントの次の章を参照してください。

関連情報

- ゴールデンイメージとは

- RHEL 9 の cloud-init の設定および管理

第2章 VHD イメージを MICROSOFT AZURE クラウドにプッシュする

RHEL Image Builder を使用して **.vhd** イメージを作成できます。作成したら、出力された **.vhd** イメージを、Microsoft Azure クラウドサービスプロバイダーの Blob Storage にプッシュできます。

前提条件

- システムへの root アクセス権があります。
- RHEL Web コンソールの RHEL Image Builder インターフェイスにアクセスできる。
- ブループリントを作成している。Web コンソールインターフェイスでの RHEL Image Builder ブループリントの作成 を参照してください。
- Microsoft ストレージアカウント が作成されました。
- 書き込み可能な Blob Storage が準備されました。

手順

1. RHEL Image Builder ダッシュボードで、使用するブループリントを選択します。
2. **Images** タブをクリックします。
3. **Create Image** をクリックして、カスタマイズした **.vhd** イメージを作成します。
   Create image ウィザードが開きます。
   a. **Type** ドロップダウンメニューリストから **Microsoft Azure (.vhd)** を選択します。
   b. イメージを Microsoft Azure クラウドにアップロードするには、**Upload to Azure** チェックボックスをオンします。
   c. **Image Size** を入力し、**Next** をクリックします。
4. **Upload to Azure** ページで、次の情報を入力します。
   a. 認証ページで、次のように入力します。
      i. **Storage account** の名前。これは、Microsoft Azure portal の **Storage account** ページにあります。
      ii. **Storage access key**. これは、**Access Key** ストレージページにあります。
      iii. **Next** をクリックします。
   b. **Authentication** ページで、次のように入力します。
      i. イメージ名
      ii. **Storage container**。これは、イメージのアップロード先の BLOB コンテナーです。Microsoft Azure portal の **Blob service** セクションにあります。
      iii. **Next** をクリックします。

5. **Review** ページで **Create** をクリックします。RHEL Image Builder が起動し、アップロードプロセスが開始します。
   Microsoft Azure Cloud にプッシュしたイメージにアクセスします。

6. Microsoft Azure ポータル にアクセスします。

7. 検索バーに "storage account" と入力し、リストから **Storage accounts** をクリックします。

8. 検索バーに "Images" と入力し、**Services** の下にある最初のエントリーを選択します。**Image Dashboard** にリダイレクトされます。

9. ナビゲーションパネルで、**Containers** をクリックします。

10. 作成したコンテナーを見つけます。コンテナー内には、RHEL Image Builder を使用して作成およびプッシュした **.vhd** ファイルがあります。

検証

1. 仮想マシンイメージを作成して起動できることを確認します。

   a. 検索バーに images account と入力し、リストから **Images** をクリックします。

   b. **+Create** をクリックします。

   c. ドロップダウンリストから、前に使用したリソースグループを選択します。

   d. イメージの名前を入力します。

   e. **OS type** で **Linux** を選択します。

   f. **VM generation** で **Gen 2** を選択します。

   g. **Storage Blob** で **Browse** をクリックし、VHD ファイルに到達するまでストレージアカウントとコンテナーをクリックします。

   h. ページの最後にある **Select** をクリックします。

   i. Account Type を選択します (例: **Standard SSD)**。

   j. **Review + Create** をクリックし、**Create** をクリックします。イメージが作成されるまでしばらく待機します。

2. 仮想マシンを起動するには、次の手順に従います。

   a. **Go to resource** をクリックします。

   b. ヘッダーのメニューバーから **+ Create VM** をクリックします。

   c. 仮想マシンの名前を入力します。

   d. **Size** セクションと **Administrator account** セクションに入力します。

   e. **Review + Create** をクリックし、**Create** をクリックします。デプロイメントの進行状況を確認できます。
      デプロイメントが完了したら、仮想マシン名をクリックしてインスタンスのパブリック IP アドレスを取得し、SSH を使用して接続します。

   f. ターミナルを開いて SSH 接続を作成し、仮想マシンに接続します。

関連情報

- Microsoft Azure ストレージのドキュメント。
- Microsoft Azure ストレージアカウントを作成します。
- Red Hat カスタマーポータルでのケースの作成
- ヘルプとサポート
- Red Hat に問い合わせる

第3章 MICROSOFT AZURE での仮想マシンとして RED HAT ENTERPRISE LINUX イメージをデプロイ

Microsoft Azure に Red Hat Enterprise Linux 9(RHEL 9) イメージをデプロイするには、以下の情報に従ってください。本章の内容は次のとおりです。

- イメージを選ぶ際の選択肢について
- ホストシステムおよび仮想マシン (VM) のシステム要件のリストまたは参照
- ISO イメージからカスタム仮想マシンを作成し、そのイメージを Azure にアップロードして、Azure 仮想インスタンスを起動する手順

重要

ISO イメージからカスタム仮想マシンを作成することは可能ですが、**Red Hat Image Builder** 製品を使用して、特定のクラウドプロバイダーで使用するようにカスタマイズされたイメージを作成することを推奨します。Image Builder を使用すると、Azure Disk Image (VHD 形式) を作成してアップロードできます。詳細は Composing a Customized RHEL System Image を参照してください。

Azure でセキュアに使用できる Red Hat 製品のリストは、Red Hat on Microsoft Azure を参照してください。

前提条件

- Red Hat カスタマーポータル のアカウントに登録している。
- Microsoft Azure アカウントに登録している。

3.1. AZURE での RED HAT ENTERPRISE LINUX イメージオプション

以下の表には、Microsoft Azure 上での RHEL 9 用イメージの選択肢を記載し、イメージオプションの相違点を示しています。

表3.1 イメージオプション

イメージオプション	サブスクリプション	サンプルシナリオ	留意事項
Red Hat Gold Image をデプロイする	既存の Red Hat サブスクリプションを使用する	Azure で Red Hat Gold Image を選択します。Gold Image の詳細と、AWS で Gold Image にアクセスする方法については、Red Hat Cloud Access Reference Guide を参照してください。	このサブスクリプションには、Red Hat のコストが含まれていますが、他のインスタンスのコストは、Microsoft 社に支払うことになります。

イメージオプション	サブスクリプション	サンプルシナリオ	留意事項
Azure に移動するカスタムイメージをデプロイする	既存の Red Hat サブスクリプションを使用する	カスタムイメージをアップロードし、サブスクリプションを割り当てます。	このサブスクリプションには、Red Hat のコストが含まれていますが、他のインスタンスのコストは、Microsoft 社に支払うことになります。
RHEL を含む既存の Azure イメージをデプロイする	Azure イメージには、Red Hat 製品が含まれる	Azure コンソールを使用して仮想マシンを作成する際に RHEL イメージを選択するか、Azure Marketplace から仮想マシンを選択します。	Microsoft 社に、**従量課金** モデルで 1 時間ごとに支払います。このようなイメージはオンデマンドと呼ばれています。Azure は、サポート契約に基づいてオンデマンドイメージのサポートを提供します。 Red Hat は、イメージの更新を提供します。Azure により、Red Hat Update Infrastructure (RHUI) から更新を利用できるようにします。

関連情報

- Using Red Hat Gold Images on Microsoft Azure
- Azure Marketplace
- Billing options in the Azure Marketplace
- Red Hat Enterprise Linux Bring-Your-Own-Subscription Gold Images in Azure
- Red Hat Cloud Access リファレンスガイド

3.2. ベースイメージの理解

このセクションでは、事前設定されたベースイメージおよびその設定を使用する方法を説明します。

3.2.1. カスタムベースイメージの使用

仮想マシン (VM) を手動で設定するには、まずベース (スターター) となる仮想マシンイメージを作成します。続いて、設定を変更して、仮想マシンがクラウドで動作するために必要なパッケージを追加できます。イメージのアップロード後に、特定のアプリケーションに追加の設定変更を行うことができます。

RHEL のクラウドイメージを準備するには、以下のセクションの手順に従います。RHEL の Hyper-V クラウドイメージを準備するには、Hyper-V Manager から Red Hat ベースの仮想マシンの準備 を参照してください。

3.2.2. 必要なシステムパッケージ

RHEL の基本イメージを作成して設定するには、ホストシステムに次のパッケージがインストールされている必要があります。

表3.2 システムパッケージ

パッケージ	リポジトリー	説明
libvirt	rhel-9-for-x86_64-appstream-rpms	プラットフォーム仮想化を管理するためのオープンソース API、デーモン、および管理ツール。
virt-install	rhel-9-for-x86_64-appstream-rpms	仮想マシンを構築するためのコマンドラインユーティリティー
libguestfs	rhel-9-for-x86_64-appstream-rpms	仮想マシンファイルシステムにアクセスして変更するためのライブラリー
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	仮想マシン用のシステム管理ツール。**virt-customize** ユーティリティーが含まれています

3.2.3. Azure VM 設定

Azure 仮想マシンには、以下の設定が必要です。設定の一部は、最初の仮想マシン作成時に有効になります。Azure 用の仮想マシンイメージのプロビジョニング時に、その他の設定が設定されます。この手順を進める際には、この設定に留意してください。必要に応じてこの設定を参照します。

表3.3 仮想マシンの設定

設定	推奨事項
ssh	Azure 仮想マシンへのリモートアクセスを提供するには、ssh を有効にする必要があります。
dhcp	プライマリー仮想アダプターは、dhcp (IPv4 のみ) 用に設定する必要があります。
swap 領域	専用 swap ファイルまたは swap パーティションは作成しないでください。Windows Azure Linux Agent (WALinuxAgent) を使用してスワップ領域を設定できます。
NIC	プライマリー仮想ネットワークアダプター用に **virtio** を選択します。
暗号化	カスタムイメージの場合には、Azure で完全なディスク暗号化に Network Bound Disk Encryption (NBDE) を使用します。

3.2.4. ISO イメージからのベースイメージの作成

以下の手順は、カスタム ISO イメージ作成の手順と初期設定の要件を示しています。イメージを設定したら、このイメージを、追加の仮想マシンインスタンスを作成するためのテンプレートとして使用できます。

前提条件

- 仮想化用のホストマシンを有効にしていることを確認します。設定および手順は、RHEL 9 で仮想化を有効にする を参照してください。

手順

1. Red Hat カスタマーポータル から最新の Red Hat Enterprise Linux 9 DVD ISO イメージをダウンロードします。
2. 基本的な Red Hat Enterprise Linux 仮想マシンを作成し、起動します。手順は、仮想マシンの作成 を参照してください。
   a. コマンドラインを使用して仮想マシンを作成する場合は、デフォルトのメモリーと CPU を仮想マシンの容量に設定するようにしてください。仮想ネットワークインターフェイスを **virtio** に設定します。
   たとえば、次のコマンドは **rhel-9.0-x86_64-kvm.qcow2** イメージを使用して **kvmtest** 仮想マシンを作成します。

   ```
   # virt-install \
     --name kvmtest --memory 2048 --vcpus 2 \
     --disk rhel-9.0-x86_64-kvm.qcow2,bus=virtio \
     --import --os-variant=rhel9.0
   ```

   b. Web コンソールを使用して仮想マシンを作成する場合は、Web コンソールで仮想マシンの作成 の手順を行います。以下の点に注意してください。
      - **仮想マシンをすぐに起動** は選択しないでください。
      - **メモリー** を希望のサイズに変更します。
      - インストールを開始する前に、**仮想ネットワークインターフェイス設定** で **モデル** を **virtio** に変更し、**vCPUs** を仮想マシンの容量設定に変更していることを確認します。
3. 以下の追加インストールの選択と変更を確認します。
   - **標準 RHEL** オプションを使用して **最小インストール** を選択します。
   - **インストール先** で、**カスタムストレージ設定** を選択します。以下の設定情報を使用して選択を行います。
     - **/boot** で、500 MB 以上であることを確認してください。
     - ファイルシステムの場合は、**boot** パーティションおよび **root** パーティションの両方に xfs、ext4、ext3 のいずれかを使用します。
     - swap 領域を削除します。swap 領域は、WALinuxAgent により Azure 内の物理ブレードサーバー上で設定されます。
   - **インストール概要** 画面で、**ネットワークおよびホスト名** を選択します。**イーサネット** を **オン** に切り替えます。

4. インストール開始時に、以下を行います。
   - **root** のパスワードを作成します。
   - 管理者ユーザーアカウントを作成します。
5. インストールが完了したら、仮想マシンを再起動して root アカウントにログインします。
6. **root** でログインしたら、イメージを設定できます。

3.3. MICROSOFT AZURE のカスタムベースイメージの設定

仮想マシン (VM) のカスタムベースイメージを作成し、Azure で特定の設定を使用して RHEL 9 VM をデプロイすることができます。以下のセクションでは、Azure で必要な追加の設定変更について説明します。

3.3.1. Hyper-V デバイスドライバーのインストール

Microsoft は、Linux Integration Services (LIS) for Hyper-V パッケージの一部として、ネットワークおよびストレージデバイスのドライバーを提供しています。Hyper-V デバイスドライバーを Azure 仮想マシン (VM) としてプロビジョニングする前に、仮想マシンイメージへのインストールが必要になる場合があります。**lsinitrd | grep hv** コマンドを使用して、ドライバーがインストールされていることを確認します。

手順

1. 以下の **grep** コマンドを実行して、必要な Hyper-V デバイスドライバーがインストールされているかどうかを確認します。

```
# lsinitrd | grep hv
```

以下の例では、必要なドライバーがすべてインストールされています。

```
# lsinitrd | grep hv
drwxr-xr-x   2 root     root            0 Aug 12 14:21 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv
-rw-r--r--   1 root     root        31272 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv/hv_vmbus.ko.xz
-rw-r--r--   1 root     root        25132 Aug 11 08:46 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/net/hyperv/hv_netvsc.ko.xz
-rw-r--r--   1 root     root         9796 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/scsi/hv_storvsc.ko.xz
```

すべてのドライバーがインストールされていない場合は、残りの手順を完了してください。

注記

hv_vmbus ドライバーは、すでにこの環境に追加されている可能性があります。このドライバーが存在する場合でも、次の手順を実行してください。

2. **/etc/dracut.conf.d** に **hv.conf** という名前のファイルを作成します。
3. 以下のドライバーパラメーターを **dracut.conf** ファイルに追加します。

```
add_drivers+=" hv_vmbus "
add_drivers+=" hv_netvsc "
add_drivers+=" hv_storvsc "
add_drivers+=" nvme "
```

注記

引用符の前後に空白に注意してください (例: **add_drivers+=" hv_VMBus ")**。これにより、環境内にその他の Hyper-V ドライバーが存在している場合に、一意のドライバーが読み込まれます。

4. **initramfs** イメージを再生成します。

```
# dracut -f -v --regenerate-all
```

検証

1. マシンを再起動します。
2. **lsinitrd | grep hv** コマンドを実行して、ドライバーがインストールされていることを確認します。

3.3.2. Microsoft Azure のデプロイメントに必要な設定変更を行う

カスタムベースイメージを Azure にデプロイする前に、追加の設定変更を実行して、仮想マシン (VM) が Azure で適切に動作できるようにする必要があります。

手順

1. VMI にログインします。
2. 仮想マシンを登録し、Red Hat Enterprise Linux 9 リポジトリーを有効にします。

```
# subscription-manager register --auto-attach
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status: Subscribed
```

3. **cloud-init** および **hyperv-daemons** パッケージがインストールされていることを確認します。

```
# dnf install cloud-init hyperv-daemons -y
```

4. Azure サービスとの統合に必要な **cloud-init** 設定ファイルを作成します。
   a. Hyper-V Data Exchange Service (KVP) へのログ記録を有効にするには、**/etc/cloud/cloud.cfg.d/10-azure-kvp.cfg** 設定ファイルを作成し、そのファイルに次の行を追加します。

```
reporting:
    logging:
        type: log
    telemetry:
        type: hyperv
```

–

b. Azure をデータソースとして追加するには、**/etc/cloud/cloud.cfg.d/91-azure_datasource.cfg** 設定ファイルを作成し、そのファイルに次の行を追加します。

```
datasource_list: [ Azure ]
datasource:
    Azure:
        apply_network_config: False
```

5. 特定のカーネルモジュールが自動的にロードされないようにするには、**/etc/modprobe.d/blocklist.conf** ファイルを編集または作成し、そのファイルに次の行を追加します。

```
blacklist nouveau
blacklist lbm-nouveau
blacklist floppy
blacklist amdgpu
blacklist skx_edac
blacklist intel_cstate
```

6. **udev** ネットワークデバイスルールを変更します。

a. 次の永続的なネットワークデバイスルールが存在する場合は削除します。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
# rm -f /etc/udev/rules.d/80-net-name-slot-rules
```

b. Azure で Accelerated Networking が意図したとおりに動作するようにするには、新しいネットワークデバイスルール **/etc/udev/rules.d/68-azure-sriov-nm-unmanaged.rules** を作成し、次の行を追加します。

```
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION=="add",
ENV{NM_UNMANAGED}="1"
```

7. **sshd** サービスが自動的に起動するように設定します。

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

8. カーネルブートパラメーターを変更します。

a. **/etc/default/grub** ファイルを開き、**GRUB_TIMEOUT** 行に次の値があることを確認します。

```
GRUB_TIMEOUT=10
```

b. 次のオプションがある場合は、**GRUB_CMDLINE_LINUX** 行の末尾から削除します。

```
rhgb quiet
```

c. **/etc/default/grub** ファイルに、指定されたすべてのオプションを含む次の行が含まれていることを確認します。

■

```
GRUB_CMDLINE_LINUX="loglevel=3 crashkernel=auto console=tty1 console=ttyS0
earlyprintk=ttyS0 rootdelay=300"
GRUB_TIMEOUT_STYLE=countdown
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

d. **grub.cfg** ファイルを再生成します。

```
# grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

システムが **grub.cfg** にデフォルト以外の場所を使用している場合は、それに応じてコマンドを調整してください。

9. Windows Azure Linux Agent (**WALinuxAgent**) を設定します。

a. **WALinuxAgent** パッケージをインストールして有効にします。

```
# dnf install WALinuxAgent -y
# systemctl enable waagent
```

b. プロビジョニングされた VM でスワップパーティションが使用されないようにするには、**/etc/waagent.conf** ファイルの次の行を編集します。

```
Provisioning.DeleteRootPassword=y
ResourceDisk.Format=n
ResourceDisk.EnableSwap=n
```

10. Azure プロビジョニング用に VM を準備します。

a. Red Hat Subscription Manager から仮想マシンの登録を解除します。

```
# subscription-manager unregister
```

b. 既存のプロビジョニングの詳細をクリーンアップします。

```
# waagent -force -deprovision
```

注記

このコマンドは警告を生成しますが、Azure が VM のプロビジョニングを自動的に処理するため、これは想定されています。

c. シェル履歴をクリーンアップし、仮想マシンをシャットダウンします。

```
# export HISTSIZE=0
# poweroff
```

3.4. イメージの固定 VHD 形式への変換

すべての Microsoft Azure 仮想マシンイメージは、固定 **VHD** 形式である必要があります。イメージは、VHD に変換する前に 1 MB の境界で調整する必要があります。イメージを **qcow2** から固定 **VHD** 形式に

変換し、イメージを整列するには、次の手順を参照してください。イメージを変換したら、Azure にアップロードできます。

手順

1. イメージを **qcow2** 形式から **raw** 形式に変換します。

```
$ qemu-img convert -f qcow2 -O raw <image-name>.qcow2 <image-name>.raw
```

2. 以下の内容でシェルスクリプトを作成します。

```
#!/bin/bash
MB=$((1024 * 1024))
size=$(qemu-img info -f raw --output json "$1" | gawk 'match($0, /"virtual-size": ([0-9]+),/, val)
{print val[1]}')
rounded_size=$((($size/$MB + 1) * $MB))
if [ $(($size % $MB)) -eq  0 ]
then
 echo "Your image is already aligned. You do not need to resize."
 exit 1
fi
echo "rounded size = $rounded_size"
export rounded_size
```

3. スクリプトを実行します。この例では **align.sh** という名前を使用します。

```
$ sh align.sh <image-xxx>.raw
```

   - **Your image is already aligned. You do not need to resize. (イメージはすでに整列しています。サイズを変更する必要はありません。)**と表示されたら、次の手順に進みます。
   - 値が表示されると、イメージは調整されません。

4. 次のコマンドを使用して、ファイルを固定 **VHD** 形式に変換します。
   サンプルでは qemu-img バージョン 2.12.0 を使用します。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

   変換されると、**VHD** ファイルは Azure にアップロードする準備が整います。

5. **raw** イメージが整列していない場合は、以下の手順で整列させてください。

   a. 検証スクリプトの実行時に表示される丸め値を使用して、**raw** ファイルのサイズを変更します。

```
$ qemu-img resize -f raw <image-xxx>.raw <rounded-value>
```

   b. **raw** イメージファイルを **VHD** 形式に変換します。
   サンプルでは qemu-img バージョン 2.12.0 を使用します。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

変換されると、**VHD** ファイルは Azure にアップロードする準備が整います。

3.5. AZURE CLI のインストール

Azure コマンドラインインターフェイス (Azure CLI 2.1) をインストールするには、以下の手順を実行します。Azure CLI 2.1 は、Azure で仮想マシンを作成し、管理する Python ベースのユーティリティーです。

前提条件

- Azure CLI を使用するための Microsoft Azure のアカウントがある。
- Azure CLI をインストールするために Python 3.x がインストールされている。

手順

1. Microsoft リポジトリーキーをインポートします。

```
$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. ローカル Azure CLI リポジトリーエントリーを作成します。

```
$ sudo sh -c 'echo -e "[azure-cli]\nname=Azure
CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" >
/etc/yum.repos.d/azure-cli.repo'
```

3. **dnf** パッケージインデックスを更新します。

```
$ dnf check-update
```

4. Python バージョンを確認 (**python --version**) し、必要に応じて Python 3.x をインストールします。

```
$ sudo dnf install python3
```

5. Azure CLI をインストールします。

```
$ sudo dnf install -y azure-cli
```

6. Azure CLI を実行します。

```
$ az
```

関連情報

- Azure CLI
- Azure CLI コマンドリファレンス

3.6. AZURE でのリソースの作成

VHD ファイルをアップロードして Azure イメージを作成する前に、以下の手順に従って、必要な Azure リソースを作成します。

手順

1. Azure でシステムを認証し、ログインします。

```
$ az login
```

> **注記**
>
> お使いの環境でブラウザーが利用可能な場合、CLI は Azure サインインページでブラウザーを開きます。詳細およびオプションは、Sign in with Azure CLI を参照してください。

2. Azure リージョンにリソースグループを作成します。

```
$ az group create --name <resource-group> --location <azure-region>
```

以下に例を示します。

```
[clouduser@localhost]$ az group create --name azrhelclirsgrp --location southcentralus
{
  "id": "/subscriptions//resourceGroups/azrhelclirsgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirsgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. ストレージアカウントを作成します。有効な SKU 値の詳細は、SKU Types を参照してください。

```
$ az storage account create -l <azure-region> -n <storage-account-name> -g <resource-group> --sku <sku_type>
```

以下に例を示します。

```
[clouduser@localhost]$ az storage account create -l southcentralus -n azrhelclistact -g azrhelclirsgrp --sku Standard_LRS
{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Storage/storageAccounts/azrhelclistact",
  "kind": "StorageV2",
  "lastGeoFailoverTime": null,
```

```
  "location": "southcentralus",
  "name": "azrhelclistact",
  "primaryEndpoints": {
    "blob": "https://azrhelclistact.blob.core.windows.net/",
    "file": "https://azrhelclistact.file.core.windows.net/",
    "queue": "https://azrhelclistact.queue.core.windows.net/",
    "table": "https://azrhelclistact.table.core.windows.net/"
},
"primaryLocation": "southcentralus",
"provisioningState": "Succeeded",
"resourceGroup": "azrhelclirsgrp",
"secondaryEndpoints": null,
"secondaryLocation": null,
"sku": {
  "name": "Standard_LRS",
  "tier": "Standard"
},
"statusOfPrimary": "available",
"statusOfSecondary": null,
"tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

4. ストレージアカウントの接続文字列を取得します。

```
$ az storage account show-connection-string -n <storage-account-name> -g <resource-group>
```

以下に例を示します。

```
[clouduser@localhost]$ az storage account show-connection-string -n azrhelclistact -g azrhelclirsgrp
{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact
AccountKey=NreGk...=="
}
```

5. 接続文字列をコピーし、次のコマンドに貼り付けて、接続文字列をエクスポートします。この文字列は、システムをストレージアカウントに接続します。

```
$ export AZURE_STORAGE_CONNECTION_STRING="<storage-connection-string>"
```

以下に例を示します。

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

6. ストレージコンテナーを作成します。

```
$ az storage container create -n <container-name>
```

以下に例を示します。

```
[clouduser@localhost]$ az storage container create -n azrhelclistcont
{
  "created": true
}
```

7. 仮想ネットワークを作成します。

```
$ az network vnet create -g <resource group> --name <vnet-name> --subnet-name <subnet-name>
```

以下に例を示します。

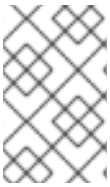
```
[clouduser@localhost]$ az network vnet create --resource-group azrhelclirsgrp --name azrhelclivnet1 --subnet-name azrhelclisubnet1
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
      "10.0.0.0/16"
      ]
  },
  "dhcpOptions": {
    "dnsServers": []
  },
  "etag": "W/\"\"",
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Network/virtualNetworks/azrhelclivnet1",
  "location": "southcentralus",
  "name": "azrhelclivnet1",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelclirsgrp",
  "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
  "subnets": [
    {
      "addressPrefix": "10.0.0.0/24",
      "etag": "W/\"\"",
      "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Network/virtualNetworks/azrhelclivnet1/subnets/azrhelclisubnet1",
      "ipConfigurations": null,
      "name": "azrhelclisubnet1",
      "networkSecurityGroup": null,
      "provisioningState": "Succeeded",
      "resourceGroup": "azrhelclirsgrp",
      "resourceNavigationLinks": null,
      "routeTable": null
    }
  ],
  "tags": {},
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": null
  }
}
```

関連情報

- Azure Managed Disks Overview
- SKU Types

3.7. AZURE イメージのアップロードおよび作成

以下の手順に従って、**VHD** ファイルをコンテナーにアップロードし、Azure カスタムイメージを作成します。



注記

システムを再起動すると、エクスポートしたストレージ接続文字列は維持されません。以下の手順でいずれかのコマンドが失敗した場合は、再び接続文字列をエクスポートしてください。

手順

1. ストレージコンテナーに **VHD** ファイルをアップロードします。これには数分かかる場合があります。ストレージコンテナーのリストを表示するには、 **az storage container list** を実行します。

```
$ az storage blob upload \
    --account-name <storage-account-name> --container-name <container-name> \
    --type page --file <path-to-vhd> --name <image-name>.vhd
```

以下に例を示します。

```
[clouduser@localhost]$ az storage blob upload \
--account-name azrhelclistact --container-name azrhelclistcont \
--type page --file rhel-image-{ProductNumber}.vhd --name rhel-image-{ProductNumber}.vhd

Percent complete: %100.0
```

2. アップロードした **VHD** ファイルの URL を以下の手順で取得します。

```
$ az storage blob url -c <container-name> -n <image-name>.vhd
```

以下に例を示します。

```
$ az storage blob url -c azrhelclistcont -n rhel-image-9.vhd
"https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd"
```

3. Azure カスタムイメージを作成します。

```
$ az image create -n <image-name> -g <resource-group> -l <azure-region> --source <URL>
--os-type linux
```

注記

仮想マシンのハイパーバイザーのデフォルトの生成は V1 です。必要に応じて、**--hyper-v-generation V2** オプションを使用して V2 ハイパーバイザーの世代を指定できます。第 2 世代の仮想マシンは、UEFI ベースのブートアーキテクチャーを使用します。第 2 世代の仮想マシンの詳細は、Support for generation 2 VMs on Azure を参照してください。

このコマンドは、Only blobs formatted as VHDs can be imported(VHD としてフォーマットされたブロブのみがインポート可能) というエラーを返す場合があります。 このエラーは、イメージが **VHD** に変換される前に最も近い 1 MB の境界に合致していないことを意味します。

以下に例を示します。

```
$ az image create -n rhel9 -g azrhelclirsgrp2 -l southcentralus --source
https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd --os-type linux
```

3.8. AZURE での仮想マシンの作成および起動

以下の手順では、イメージから管理ディスクの Azure 仮想マシンを作成するための最小限のコマンドオプションを説明します。追加オプションは、az vm create を参照してください。

手順

1. 次のコマンドを実行して仮想マシンを作成します。

```
$ az vm create \
   -g <resource-group> -l <azure-region> -n <vm-name> \
   --vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
   --os-disk-name <simple-name> --admin-username <administrator-name> \
   --generate-ssh-keys --image <path-to-image>
```

注記

--generate-ssh-keys オプションを指定すると、秘密鍵と公開鍵のペアが作成されます。秘密鍵ファイルと公開鍵ファイルが、システムの ~/**.ssh** に作成されます。公開鍵は、**--admin-username** オプションで指定したユーザーの仮想マシン上の **authorized_keys** ファイルに追加されます。詳細は、その他認証方法 を参照してください。

以下に例を示します。

```
[clouduser@localhost]$ az vm create \
-g azrhelclirsgrp2 -l southcentralus -n rhel-azure-vm-1 \
--vnet-name azrhelclivnet1 --subnet azrhelclisubnet1 --size Standard_A2 \
--os-disk-name vm-1-osdisk --admin-username clouduser \
--generate-ssh-keys --image rhel9

{
  "fqdns": "",
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Compute/virtualMachines/rhe
```

```
l-azure-vm-1",
  "location": "southcentralus",
  "macAddress": "",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "<public-IP-address>",
  "resourceGroup": "azrhelclirsgrp2"
```

publicIpAddress を書き留めます。以下の手順で仮想マシンにログインするには、このアドレスが必要です。

2. SSH セッションを開始し、仮想マシンにログインします。

```
[clouduser@localhost]$ ssh -i /home/clouduser/.ssh/id_rsa clouduser@<public-IP-address>.
The authenticity of host ',<public-IP-address>' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '<public-IP-address>' (ECDSA) to the list of known hosts.

[clouduser@rhel-azure-vm-1 ~]$
```

ユーザープロンプトが表示された場合は、Azure VM が正常にデプロイされます。

これで、Microsoft Azure ポータルにアクセスして、リソースの監査ログとプロパティーを確認できます。このポータルでは、仮想マシンを直接管理できます。複数の仮想マシンを管理している場合は、Azure CLI を使用する必要があります。Azure CLI では、Azure 内のリソースに強力なインターフェイスを利用できます。Microsoft Azure で仮想マシンを管理するために使用するコマンドの詳細は、CLI で **az --help** を実行するか、Azure CLI コマンドリファレンス を参照してください。

3.9. その他認証方法

セキュリティーを強化するために推奨されますが、Azure 生成のキーペアを使用する必要はありません。以下の例は、SSH 認証用の 2 つの方法を示しています。

例 1: 次のコマンドオプションは、公開鍵ファイルを生成せずに新しい仮想マシンをプロビジョニングします。これにより、SSH は、パスワードを使用した SSH 認証を許可できます。

```
$ az vm create \
   -g <resource-group> -l <azure-region> -n <vm-name> \
   --vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
   --os-disk-name <simple-name> --authentication-type password \
   --admin-username <administrator-name> --admin-password <ssh-password> --image <path-to-
image>
```

```
$ ssh <admin-username>@<public-ip-address>
```

例 2: このコマンドオプションにより、新しい Azure 仮想マシンがプロビジョニングされ、既存の公開鍵ファイルを使用した SSH 認証が許可されます。

```
$ az vm create \
   -g <resource-group> -l <azure-region> -n <vm-name> \
   --vnet-name <vnet-name> --subnet <subnet-name> --size Standard_A2 \
   --os-disk-name <simple-name> --admin-username <administrator-name> \
   --ssh-key-value <path-to-existing-ssh-key> --image <path-to-image>
```

```
$ ssh -i <path-to-existing-ssh-key> <admin-username>@<public-ip-address>
```

3.10. RED HAT サブスクリプションの割り当て

subscription-manager コマンドを使用すると、Red Hat サブスクリプションを RHEL インスタンスに登録し、割り当てることができます。

前提条件

- サブスクリプションが有効になっている。

手順

1. システムを登録します。

```
# subscription-manager register --auto-attach
```

2. サブスクリプションを割り当てます。
   - アクティベーションキーを使用して、サブスクリプションを割り当てることができます。詳細は、カスタマーポータルのアクティベーションキーを作成する を参照してください。
   - または、サブスクリプションプール (Pool ID) の ID を使用してサブスクリプションを手動で割り当てることができます。コマンドラインでのサブスクリプションのアタッチと削除 を参照してください。
3. **オプション：** Red Hat Hybrid Cloud Console でインスタンスに関するさまざまなシステムメトリックを収集するには、インスタンスを Red Hat Insights に登録できます。

```
# insights-client register --display-name <display-name-value>
```

   Red Hat Insights の詳細な設定の詳細は、Red Hat Insights のクライアント設定ガイド を参照してください。

関連情報

- カスタマーポータルのアクティベーションキーを作成する
- コマンドラインでのサブスクリプションのアタッチと削除
- Red Hat Subscription Manager の使用および設定
- Red Hat Insights のクライアント設定ガイド

3.11. AZURE GOLD IMAGE の自動登録の設定

Micorsoft Azure 上に RHEL 9 の仮想マシン (VM) をより早く、より快適にデプロイするために、RHEL 9 の Gold Image を Red Hat Subscription Manager(RHSM) に自動的に登録するように設定することができます。

前提条件

- RHEL 9 Gold Image は Microsoft Azure で利用できます。手順については、Azure でのゴールドイメージの使用 を参照してください。

注記

Microsoft Azure アカウントは、一度に 1 つの Red Hat アカウントにしか割り当てできません。そのため、Red Hat アカウントにアタッチする前に、他のユーザーが Azure アカウントへのアクセスを必要としていないことを確認してください。

手順

1. Gold Image を使用して、Azure インスタンスに RHEL 9 仮想マシンを作成します。手順については、Creating and starting the VM in Azure を参照してください。
2. 作成した仮想マシンを起動します。
3. RHEL 9 仮想マシンで、自動登録を有効にします。

```
# subscription-manager config --rhsmcertd.auto_registration=1
```

4. **rhsmcertd** サービスを有効にします。

```
# systemctl enable rhsmcertd.service
```

5. **redhat.repo** リポジトリーを無効にします。

```
# subscription-manager config --rhsm.manage_repos=0
```

6. 仮想マシンの電源を切り、Azure 上で管理イメージとして保存します。手順については、How to create a managed image of a virtual machine or VHD を参照してください。
7. 管理イメージを使用して仮想マシンを作成します。自動的に RHSM に登録されます。

検証

- 上記の手順で作成した RHEL 9 仮想マシンで、**subscription-manager identity** コマンドを実行して、システムが RHSM に登録されていることを確認します。登録に成功したシステムでは、システムの UUID が表示されます。以下に例を示します。

```
# subscription-manager identity
system identity: fdc46662-c536-43fb-a18a-bbcb283102b7
name: 192.168.122.222
org name: 6340056
org ID: 6340056
```

関連情報

- Azure での Red Hat ゴールドイメージ
- Overview of RHEL images in Azure
- Configuring cloud sources for Red Hat services

3.12. 関連情報

- Red Hat in the Public Cloud
- Red Hat Cloud Access リファレンスガイド
- Frequently Asked Questions and Recommended Practices for Microsoft Azure

第4章 MICROSOFT AZURE での RED HAT HIGH AVAILABILITY クラスターの設定

ノード障害が発生した場合に RHEL ノードがワークロードを自動的に再分配するクラスターを作成するには、Red Hat High Availability Add-On を使用します。このような高可用性(HA)クラスターは、Microsoft Azure を含むパブリッククラウドプラットフォームでホストすることもできます。Azure で RHEL HA クラスターの作成は、非クラウド環境で HA クラスターを作成することと類似していますが、特定の詳細があります。

Azure 仮想マシン(VM)インスタンスをクラスターノードとして使用して Azure で Red Hat HA クラスターを設定するには、以下のセクションを参照してください。これらのセクションの手順では、Azure のカスタムイメージを作成していることを前提としています。クラスターに使用する RHEL 9 イメージを取得するオプションは複数あります。Azure のイメージオプションに関する詳細は、Azure における Red Hat Enterprise Linux Image オプションを参照してください。

次のセクションで説明します。

- Azure 用の環境を設定するための前提手順。環境を設定したら、Azure VM インスタンスを作成して設定できます。
- 個々のノードを Azure 上の HA ノードのクラスターに変換する、HA クラスターの作成に固有の手順。これには、各クラスターノードに高可用性パッケージおよびエージェントをインストールし、フェンシングを設定し、Azure ネットワークリソースエージェントをインストールする手順が含まれます。

前提条件

- Red Hat カスタマーポータル のアカウントにサインアップします。
- 管理者権限で Microsoft Azure アカウント にサインアップします。
- Azure コマンドラインインターフェイス (CLI) をインストールする必要があります。詳細は、Azure CLI のインストール を参照してください。

4.1. パブリッククラウドプラットフォームで高可用性クラスターを使用する利点

高可用性 (HA) クラスターは、特定のワークロードを実行するためにリンクされた一連のコンピューター (**ノード** と呼ばれます) です。HA クラスターの目的は、ハードウェアまたはソフトウェア障害が発生した場合に備えて、冗長性を提供することです。HA クラスター内のノードに障害が発生しても、Pacemaker クラスターリソースマネージャーがワークロードを他のノードに分散するため、クラスター上で実行されているサービスに顕著なダウンタイムが発生することはありません。

HA クラスターはパブリッククラウドプラットフォームで実行することもできます。この場合、クラウド内の仮想マシン (VM) インスタンスを個々のクラスターノードとして使用します。パブリッククラウドプラットフォームで HA クラスターを使用すると、次の利点があります。

- 可用性の向上: VM に障害が発生した場合、ワークロードがすぐに他のノードに再分散されるため、実行中のサービスが中断されません。
- スケーラビリティー: 需要が高いときに追加のノードを起動し、需要が低いときにノードを停止することができます。
- 費用対効果: 従量課金制の料金設定では、実行中のノードに対して料金を支払うだけで済みます。

- 管理の簡素化: 一部のパブリッククラウドプラットフォームでは、HA クラスターの設定を容易にする管理インターフェイスが提供されています。

Red Hat Enterprise Linux (RHEL) システムで HA を有効にするために、Red Hat は High Availability Add-On を提供しています。High Availability Add-On は、RHEL システム上で HA クラスターを作成するために必要なすべてのコンポーネントを提供します。コンポーネントには、高可用性サービス管理ツールとクラスター管理ツールが含まれます。

関連情報

- High Availability Add-On の概要

4.2. AZURE でのリソースの作成

リージョン、リソースグループ、ストレージアカウント、仮想ネットワーク、および可用性セットを作成するには、以下の手順を実施します。Microsoft Azure でクラスターをセットアップするには、これらのリソースが必要です。

手順

1. Azure でシステムを認証し、ログインします。

```
$ az login
```

> **注記**
>
> お使いの環境でブラウザーが利用可能な場合、CLI は Azure サインインページでブラウザーを開きます。

以下に例を示します。

```
[clouduser@localhost]$ az login

To sign in, use a web browser to open the page https://aka.ms/devicelogin and enter the code
FDMSCMETZ to authenticate.
 [
  {
    "cloudName": "AzureCloud",
    "id": "Subscription ID",
    "isDefault": true,
    "name": "MySubscriptionName",
    "state": "Enabled",
    "tenantId": "Tenant ID",
    "user": {
      "name": "clouduser@company.com",
      "type": "user"
    }
  }
 ]
```

2. Azure リージョンにリソースグループを作成します。

```
$ az group create --name resource-group --location azure-region
```

以下に例を示します。

```
[clouduser@localhost]$ az group create --name azrhelclirsgrp --location southcentralus

{
  "id": "/subscriptions//resourceGroups/azrhelclirsgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirsgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. ストレージアカウントを作成します。

```
$ az storage account create -l azure-region -n storage-account-name -g resource-group -
-sku sku_type --kind StorageV2
```

以下に例を示します。

```
[clouduser@localhost]$ az storage account create -l southcentralus -n azrhelclistact -g
azrhelclirsgrp --sku Standard_LRS --kind StorageV2

{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Storage/storageAccounts/azr
helclistact",
  "kind": "StorageV2",
  "lastGeoFailoverTime": null,
  "location": "southcentralus",
  "name": "azrhelclistact",
  "primaryEndpoints": {
    "blob": "https://azrhelclistact.blob.core.windows.net/",
    "file": "https://azrhelclistact.file.core.windows.net/",
    "queue": "https://azrhelclistact.queue.core.windows.net/",
    "table": "https://azrhelclistact.table.core.windows.net/"
},
"primaryLocation": "southcentralus",
"provisioningState": "Succeeded",
"resourceGroup": "azrhelclirsgrp",
"secondaryEndpoints": null,
"secondaryLocation": null,
"sku": {
  "name": "Standard_LRS",
  "tier": "Standard"
},
"statusOfPrimary": "available",
"statusOfSecondary": null,
```

```
"tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

4. ストレージアカウントの接続文字列を取得します。

```
$ az storage account show-connection-string -n storage-account-name -g resource-group
```

以下に例を示します。

```
[clouduser@localhost]$ az storage account show-connection-string -n azrhelclistact -g
azrhelclirsgrp
{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact
AccountKey=NreGk...=="
}
```

5. 接続文字列をコピーし、次のコマンドに貼り付けて、接続文字列をエクスポートします。この文字列は、システムをストレージアカウントに接続します。

```
$ export AZURE_STORAGE_CONNECTION_STRING="storage-connection-string"
```

以下に例を示します。

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffi
x=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

6. ストレージコンテナーを作成します。

```
$ az storage container create -n container-name
```

以下に例を示します。

```
[clouduser@localhost]$ az storage container create -n azrhelclistcont

{
  "created": true
}
```

7. 仮想ネットワークを作成します。すべてのクラスターノードが同じ仮想ネットワークにある必要があります。

```
$ az network vnet create -g resource group --name vnet-name --subnet-name subnet-
name
```

以下に例を示します。

```
[clouduser@localhost]$ az network vnet create --resource-group azrhelclirsgrp --name
azrhelclivnet1 --subnet-name azrhelclisubnet1
{
  "newVNet": {
```

```
    "addressSpace": {
      "addressPrefixes": [
      "10.0.0.0/16"
      ]
  },
  "dhcpOptions": {
    "dnsServers": []
  },
  "etag": "W/\"\"",
  "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1",
  "location": "southcentralus",
  "name": "azrhelclivnet1",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelclirsgrp",
  "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
  "subnets": [
    {
      "addressPrefix": "10.0.0.0/24",
      "etag": "W/\"\"",
      "id":
"/subscriptions//resourceGroups/azrhelclirsgrp/providers/Microsoft.Network/virtualNetworks/azr
helclivnet1/subnets/azrhelclisubnet1",
      "ipConfigurations": null,
      "name": "azrhelclisubnet1",
      "networkSecurityGroup": null,
      "provisioningState": "Succeeded",
      "resourceGroup": "azrhelclirsgrp",
      "resourceNavigationLinks": null,
      "routeTable": null
    }
  ],
  "tags": {},
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": null
  }
}
```

8. 可用性セットを作成します。すべてのクラスターノードは同じ可用性セットにある必要があります。

```
$ az vm availability-set create --name MyAvailabilitySet --resource-group
MyResourceGroup
```

以下に例を示します。

```
[clouduser@localhost]$ az vm availability-set create --name rhelha-avset1 --resource-group
azrhelclirsgrp
{
  "additionalProperties": {},
    "id":
"/subscriptions/.../resourceGroups/azrhelclirsgrp/providers/Microsoft.Compute/availabilitySets/rh
elha-avset1",
    "location": "southcentralus",
```

```
    "name": “rhelha-avset1",
    "platformFaultDomainCount": 2,
    "platformUpdateDomainCount": 5,

[omitted]
```

関連情報

- Sign in with Azure CLI
- SKU Types
- Azure Managed Disks Overview

4.3. 高可用性に必要なシステムパッケージ

この手順では、Red Hat Enterprise Linux を使用して Azure HA 用の仮想マシンイメージを作成していることを前提としています。この手順を正常に行うには、以下のパッケージをインストールする必要があります。

表4.1 システムパッケージ

パッケージ	リポジトリー	説明
libvirt	rhel-9-for-x86_64-appstream-rpms	プラットフォーム仮想化を管理するためのオープンソース API、デーモン、および管理ツール。
virt-install	rhel-9-for-x86_64-appstream-rpms	仮想マシンを構築するためのコマンドラインユーティリティー
libguestfs	rhel-9-for-x86_64-appstream-rpms	仮想マシンファイルシステムにアクセスして変更するためのライブラリー
guestfs-tools	rhel-9-for-x86_64-appstream-rpms	仮想マシン用のシステム管理ツール。**virt-customize** ユーティリティーが含まれています

4.4. AZURE VM 設定

Azure 仮想マシンには、以下の設定が必要です。設定の一部は、最初の仮想マシン作成時に有効になります。Azure 用の仮想マシンイメージのプロビジョニング時に、その他の設定が設定されます。この手順を進める際には、この設定に留意してください。必要に応じてこの設定を参照します。

表4.2 仮想マシンの設定

設定	推奨事項
ssh	Azure 仮想マシンへのリモートアクセスを提供するには、ssh を有効にする必要があります。

設定	推奨事項
dhcp	プライマリー仮想アダプターは、dhcp (IPv4 のみ) 用に設定する必要があります。
swap 領域	専用 swap ファイルまたは swap パーティションは作成しないでください。Windows Azure Linux Agent (WALinuxAgent) を使用してスワップ領域を設定できます。
NIC	プライマリー仮想ネットワークアダプター用に **virtio** を選択します。
暗号化	カスタムイメージの場合には、Azure で完全なディスク暗号化に Network Bound Disk Encryption (NBDE) を使用します。

4.5. HYPER-V デバイスドライバーのインストール

Microsoft は、Linux Integration Services (LIS) for Hyper-V パッケージの一部として、ネットワークおよびストレージデバイスのドライバーを提供しています。Hyper-V デバイスドライバーを Azure 仮想マシン (VM) としてプロビジョニングする前に、仮想マシンイメージへのインストールが必要になる場合があります。**lsinitrd | grep hv** コマンドを使用して、ドライバーがインストールされていることを確認します。

手順

1. 以下の **grep** コマンドを実行して、必要な Hyper-V デバイスドライバーがインストールされているかどうかを確認します。

```
# lsinitrd | grep hv
```

以下の例では、必要なドライバーがすべてインストールされています。

```
# lsinitrd | grep hv
drwxr-xr-x   2 root     root            0 Aug 12 14:21 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv
-rw-r--r--   1 root     root        31272 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/hv/hv_vmbus.ko.xz
-rw-r--r--   1 root     root        25132 Aug 11 08:46 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/net/hyperv/hv_netvsc.ko.xz
-rw-r--r--   1 root     root         9796 Aug 11 08:45 usr/lib/modules/3.10.0-
932.el9.x86_64/kernel/drivers/scsi/hv_storvsc.ko.xz
```

すべてのドライバーがインストールされていない場合は、残りの手順を完了してください。

注記

hv_vmbus ドライバーは、すでにこの環境に追加されている可能性があります。このドライバーが存在する場合でも、次の手順を実行してください。

2. **/etc/dracut.conf.d** に **hv.conf** という名前のファイルを作成します。

3. 以下のドライバーパラメーターを **dracut.conf** ファイルに追加します。

```
add_drivers+=" hv_vmbus "
add_drivers+=" hv_netvsc "
add_drivers+=" hv_storvsc "
add_drivers+=" nvme "
```

注記

引用符の前後に空白に注意してください (例: **add_drivers+=" hv_VMBus ")**。これにより、環境内にその他の Hyper-V ドライバーが存在している場合に、一意のドライバーが読み込まれます。

4. **initramfs** イメージを再生成します。

```
# dracut -f -v --regenerate-all
```

検証

1. マシンを再起動します。

2. **lsinitrd | grep hv** コマンドを実行して、ドライバーがインストールされていることを確認します。

4.6. MICROSOFT AZURE のデプロイメントに必要な設定変更を行う

カスタムベースイメージを Azure にデプロイする前に、追加の設定変更を実行して、仮想マシン (VM) が Azure で適切に動作できるようにする必要があります。

手順

1. VMI にログインします。

2. 仮想マシンを登録し、Red Hat Enterprise Linux 9 リポジトリーを有効にします。

```
# subscription-manager register --auto-attach
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status: Subscribed
```

3. **cloud-init** および **hyperv-daemons** パッケージがインストールされていることを確認します。

```
# dnf install cloud-init hyperv-daemons -y
```

4. Azure サービスとの統合に必要な **cloud-init** 設定ファイルを作成します。

   a. Hyper-V Data Exchange Service (KVP) へのログ記録を有効にするには、**/etc/cloud/cloud.cfg.d/10-azure-kvp.cfg** 設定ファイルを作成し、そのファイルに次の行を追加します。

```
reporting:
```

```
    logging:
      type: log
    telemetry:
      type: hyperv
```

b. Azure をデータソースとして追加するには、**/etc/cloud/cloud.cfg.d/91-azure_datasource.cfg** 設定ファイルを作成し、そのファイルに次の行を追加します。

```
datasource_list: [ Azure ]
datasource:
  Azure:
    apply_network_config: False
```

5. 特定のカーネルモジュールが自動的にロードされないようにするには、**/etc/modprobe.d/blocklist.conf** ファイルを編集または作成し、そのファイルに次の行を追加します。

```
blacklist nouveau
blacklist lbm-nouveau
blacklist floppy
blacklist amdgpu
blacklist skx_edac
blacklist intel_cstate
```

6. **udev** ネットワークデバイスルールを変更します。

a. 次の永続的なネットワークデバイスルールが存在する場合は削除します。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# rm -f /etc/udev/rules.d/75-persistent-net-generator.rules
# rm -f /etc/udev/rules.d/80-net-name-slot-rules
```

b. Azure で Accelerated Networking が意図したとおりに動作するようにするには、新しいネットワークデバイスルール **/etc/udev/rules.d/68-azure-sriov-nm-unmanaged.rules** を作成し、次の行を追加します。

```
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION=="add",
ENV{NM_UNMANAGED}="1"
```

7. **sshd** サービスが自動的に起動するように設定します。

```
# systemctl enable sshd
# systemctl is-enabled sshd
```

8. カーネルブートパラメーターを変更します。

a. **/etc/default/grub** ファイルを開き、**GRUB_TIMEOUT** 行に次の値があることを確認します。

```
GRUB_TIMEOUT=10
```

b. 次のオプションがある場合は、**GRUB_CMDLINE_LINUX** 行の末尾から削除します。

```
rhgb quiet
```

c. **/etc/default/grub** ファイルに、指定されたすべてのオプションを含む次の行が含まれていることを確認します。

```
GRUB_CMDLINE_LINUX="loglevel=3 crashkernel=auto console=tty1 console=ttyS0
earlyprintk=ttyS0 rootdelay=300"
GRUB_TIMEOUT_STYLE=countdown
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

d. **grub.cfg** ファイルを再生成します。

```
# grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

システムが **grub.cfg** にデフォルト以外の場所を使用している場合は、それに応じてコマンドを調整してください。

9. Windows Azure Linux Agent (**WALinuxAgent**) を設定します。

a. **WALinuxAgent** パッケージをインストールして有効にします。

```
# dnf install WALinuxAgent -y
# systemctl enable waagent
```

b. プロビジョニングされた VM でスワップパーティションが使用されないようにするには、**/etc/waagent.conf** ファイルの次の行を編集します。

```
Provisioning.DeleteRootPassword=y
ResourceDisk.Format=n
ResourceDisk.EnableSwap=n
```

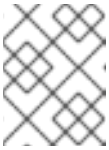
10. Azure プロビジョニング用に VM を準備します。

a. Red Hat Subscription Manager から仮想マシンの登録を解除します。

```
# subscription-manager unregister
```

b. 既存のプロビジョニングの詳細をクリーンアップします。

```
# waagent -force -deprovision
```



注記

このコマンドは警告を生成しますが、Azure が VM のプロビジョニングを自動的に処理するため、これは想定されています。

c. シェル履歴をクリーンアップし、仮想マシンをシャットダウンします。

```
# export HISTSIZE=0
# poweroff
```

4.7. AZURE ACTIVE DIRECTORY アプリケーションの作成

Azure Active Directory (AD) アプリケーションを作成するには、以下の手順を行います。Azure AD アプリケーションは、クラスター内のすべてのノードに対する HA 操作のアクセスを許可し、自動化します。

前提条件

- Azure コマンドラインインターフェイス (CLI) がシステムにインストールされている。
- Microsoft Azure サブスクリプションの管理者または所有者である。Azure AD アプリケーションを作成するには、この承認が必要です。

手順

1. HA クラスター内の任意のノードで、Azure アカウントにログインします。

```
$ az login
```

2. Azure フェンスエージェントのカスタムロールの **json** 設定ファイルを作成します。次の設定を使用します。ただし、**<subscription-id>** はご自身のサブスクリプション ID に置き換えます。

```
{
    "Name": "Linux Fence Agent Role",
    "description": "Allows to power-off and start virtual machines",
    "assignableScopes": [
        "/subscriptions/<subscription-id>"
    ],
    "actions": [
        "Microsoft.Compute/*/read",
        "Microsoft.Compute/virtualMachines/powerOff/action",
        "Microsoft.Compute/virtualMachines/start/action"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
}
```

3. Azure フェンスエージェントのカスタムロールを定義します。前の手順で作成した **json** ファイルを使用します。

```
$ az role definition create --role-definition azure-fence-role.json

{
  "assignableScopes": [
    "/subscriptions/<my-subscription-id>"
  ],
  "description": "Allows to power-off and start virtual machines",
  "id": "/subscriptions/<my-subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-id>",
  "name": "<role-id>",
  "permissions": [
    {
      "actions": [
        "Microsoft.Compute/*/read",
```

```
        "Microsoft.Compute/virtualMachines/powerOff/action",
        "Microsoft.Compute/virtualMachines/start/action"
      ],
      "dataActions": [],
      "notActions": [],
      "notDataActions": []
    }
  ],
  "roleName": "Linux Fence Agent Role",
  "roleType": "CustomRole",
  "type": "Microsoft.Authorization/roleDefinitions"
}
```

4. Azure Web コンソールインターフェイスで、**Virtual Machine** を選択 → 左側のメニューの **Identity** をクリックします。
5. **On** を選択→ **Save** をクリック→ **Yes** をクリックして確認します。
6. **Azure role assignments** → **Add role assignment** をクリックします。
7. ロールに必要な **Scope** (例: **Resource Group**) を選択します。
8. 必要な **Resource Group** を選択します。
9. **オプション:** 必要に応じて **Subscription** を変更します。
10. **Linux Fence Agent Role** ロールを選択します。
11. **Save** をクリックします。

検証

- Azure AD に表示されるノードを表示します。

```
# fence_azure_arm --msi -o list
node1,
node2,
[...]
```

このコマンドでクラスター上のすべてのノードが出力された場合、AD アプリケーションは正常に設定されています。

関連情報

- View the access a user has to Azure resources
- Create a custom role for the fence agent
- Assign Azure roles by using Azure CLI

4.8. イメージの固定 VHD 形式への変換

すべての Microsoft Azure 仮想マシンイメージは、固定 **VHD** 形式である必要があります。イメージは、VHD に変換する前に 1 MB の境界で調整する必要があります。イメージを **qcow2** から固定 **VHD** 形式に変換し、イメージを整列するには、次の手順を参照してください。イメージを変換したら、Azure に

アップロードできます。

手順

1. イメージを **qcow2** 形式から **raw** 形式に変換します。

```
$ qemu-img convert -f qcow2 -O raw <image-name>.qcow2 <image-name>.raw
```

2. 以下の内容でシェルスクリプトを作成します。

```
#!/bin/bash
MB=$((1024 * 1024))
size=$(qemu-img info -f raw --output json "$1" | gawk 'match($0, /"virtual-size": ([0-9]+),/, val)
{print val[1]}')
rounded_size=$((($size/$MB + 1) * $MB))
if [ $(($size % $MB)) -eq  0 ]
then
 echo "Your image is already aligned. You do not need to resize."
 exit 1
fi
echo "rounded size = $rounded_size"
export rounded_size
```

3. スクリプトを実行します。この例では **align.sh** という名前を使用します。

```
$ sh align.sh <image-xxx>.raw
```

- **Your image is already aligned. You do not need to resize. (イメージはすでに整列しています。サイズを変更する必要はありません。)**と表示されたら、次の手順に進みます。
- 値が表示されると、イメージは調整されません。

4. 次のコマンドを使用して、ファイルを固定 **VHD** 形式に変換します。
サンプルでは qemu-img バージョン 2.12.0 を使用します。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

変換されると、**VHD** ファイルは Azure にアップロードする準備が整います。

5. **raw** イメージが整列していない場合は、以下の手順で整列させてください。

   a. 検証スクリプトの実行時に表示される丸め値を使用して、**raw** ファイルのサイズを変更します。

```
$ qemu-img resize -f raw <image-xxx>.raw <rounded-value>
```

   b. **raw** イメージファイルを **VHD** 形式に変換します。
   サンプルでは qemu-img バージョン 2.12.0 を使用します。

```
$ qemu-img convert -f raw -o subformat=fixed,force_size -O vpc <image-xxx>.raw
<image.xxx>.vhd
```

   変換されると、**VHD** ファイルは Azure にアップロードする準備が整います。

4.9. AZURE イメージのアップロードおよび作成

以下の手順に従って、**VHD** ファイルをコンテナーにアップロードし、Azure カスタムイメージを作成します。

注記

システムを再起動すると、エクスポートしたストレージ接続文字列は維持されません。以下の手順でいずれかのコマンドが失敗した場合は、再び接続文字列をエクスポートしてください。

手順

1. ストレージコンテナーに **VHD** ファイルをアップロードします。これには数分かかる場合があります。ストレージコンテナーのリストを表示するには、 **az storage container list** を実行します。

```
$ az storage blob upload \
    --account-name <storage-account-name> --container-name <container-name> \
    --type page --file <path-to-vhd> --name <image-name>.vhd
```

以下に例を示します。

```
[clouduser@localhost]$ az storage blob upload \
--account-name azrhelclistact --container-name azrhelclistcont \
--type page --file rhel-image-{ProductNumber}.vhd --name rhel-image-{ProductNumber}.vhd

Percent complete: %100.0
```

2. アップロードした **VHD** ファイルの URL を以下の手順で取得します。

```
$ az storage blob url -c <container-name> -n <image-name>.vhd
```

以下に例を示します。

```
$ az storage blob url -c azrhelclistcont -n rhel-image-9.vhd
"https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd"
```

3. Azure カスタムイメージを作成します。

```
$ az image create -n <image-name> -g <resource-group> -l <azure-region> --source <URL>
--os-type linux
```

注記

仮想マシンのハイパーバイザーのデフォルトの生成は V1 です。必要に応じて、**--hyper-v-generation V2** オプションを使用して V2 ハイパーバイザーの世代を指定できます。第 2 世代の仮想マシンは、UEFI ベースのブートアーキテクチャーを使用します。第 2 世代の仮想マシンの詳細は、Support for generation 2 VMs on Azure を参照してください。

このコマンドは、Only blobs formatted as VHDs can be imported(VHD としてフォーマットされたブロブのみがインポート可能) というエラーを返す場合があります。 このエラーは、イメージが **VHD** に変換される前に最も近い 1 MB の境界に合致していないことを意味します。

以下に例を示します。

```
$ az image create -n rhel9 -g azrhelclirsgrp2 -l southcentralus --source
https://azrhelclistact.blob.core.windows.net/azrhelclistcont/rhel-image-9.vhd --os-type linux
```

4.10. RED HAT HA パッケージおよびエージェントのインストール

すべてのノードで以下の手順を実行します。

手順

1. SSH ターミナルセッションを起動し、管理者名とパブリック IP アドレスを使用して仮想マシンに接続します。

```
$ ssh administrator@PublicIP
```

Azure 仮想マシンのパブリック IP アドレスを取得するには、Azure ポータルで仮想マシンプロパティーを開くか、以下の Azure CLI コマンドを入力します。

```
$ az vm list -g <resource-group> -d --output table
```

以下に例を示します。

```
[clouduser@localhost ~] $ az vm list -g azrhelclirsgrp -d --output table
Name    ResourceGroup          PowerState      PublicIps       Location
------  ----------------------  --------------  -------------   --------------
node01  azrhelclirsgrp         VM running      192.98.152.251  southcentralus
```

2. 仮想マシンを Red Hat に登録します。

```
$ sudo -i
# subscription-manager register --auto-attach
```

注記

--auto-attach コマンドが失敗する場合は、手動で仮想マシンをサブスクリプション登録します。

3. すべてのリポジトリーを無効にします。

```
# subscription-manager repos --disable=*
```

4. RHEL9 サーバーの HA リポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
```

5. すべてのパッケージを更新します。

```
# dnf update -y
```

6. High Availability チャンネルの Azure フェンシングエージェントとともに、Red Hat High Availability Add-On ソフトウェアパッケージをインストールします。

```
# dnf install pcs pacemaker fence-agents-azure-arm
```

7. **hacluster** ユーザーは、前の手順で pcs および pacemaker のインストール時に作成されました。すべてのクラスターノードに **hacluster** のパスワードを作成します。すべてのノードに同じパスワードを使用します。

```
# passwd hacluster
```

8. **firewalld.service** がインストールされている場合は、RHEL ファイアウォールに **高可用性** サービスを追加します。

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

9. **pcs** サービスを起動し、システムの起動時に開始できるようにします。

```
# systemctl start pcsd.service
# systemctl enable pcsd.service

Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to
/usr/lib/systemd/system/pcsd.service.
```

検証

- **pcs** サービスが実行していることを確認します。

```
# systemctl status pcsd.service
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Fri 2018-02-23 11:00:58 EST; 1min 23s ago
Docs: man:pcsd(8)
        man:pcs(8)
Main PID: 46235 (pcsd)
  CGroup: /system.slice/pcsd.service
        └─46235 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
```

4.11. クラスターの作成

ノードのクラスターを作成するには、以下の手順を実施します。

手順

1. ノードのいずれかで以下のコマンドを実行し、pcs ユーザー **hacluster** を認証します。コマンドで、クラスター内の各ノードの名前を指定します。

```
# pcs host auth <hostname1> <hostname2> <hostname3>
```

以下に例を示します。

```
[root@node01 clouduser]# pcs host auth node01 node02 node03
Username: hacluster
Password:
node01: Authorized
node02: Authorized
node03: Authorized
```

2. クラスターを作成します。

```
# pcs cluster setup <cluster_name> <hostname1> <hostname2> <hostname3>
```

以下に例を示します。

```
[root@node01 clouduser]# pcs cluster setup new_cluster node01 node02 node03

[...]

Synchronizing pcsd certificates on nodes node01, node02, node03...
node02: Success
node03: Success
node01: Success
Restarting pcsd on the nodes in order to reload the certificates...
node02: Success
node03: Success
node01: Success
```

検証

1. クラスターを有効にします。

```
[root@node01 clouduser]# pcs cluster enable --all
node02: Cluster Enabled
node03: Cluster Enabled
node01: Cluster Enabled
```

2. クラスターを起動します。

```
[root@node01 clouduser]# pcs cluster start --all
node02: Starting Cluster...
node03: Starting Cluster...
node01: Starting Cluster...
```

4.12. フェンシングの概要

クラスター内のノードの 1 つと通信が失敗した場合に、障害が発生したクラスターノードがアクセスする可能性があるリソースへのアクセスを、その他のノードが制限したり、解放したりできるようにする必要があります。クラスターノードが応答しない可能性があるため、そのクラスターノードと通信しても成功しません。代わりに、フェンスエージェントを使用した、フェンシングと呼ばれる外部メソッドを指定する必要があります。

応答しないノードがデータへのアクセスを続けている可能性があります。データが安全であることを確認する場合は、STONITH を使用してノードをフェンスすることが唯一の方法になります。STONITH は Shoot The Other Node In The Head の頭字語で、不安定なノードや同時アクセスによるデータの破損を防ぐことができます。STONITH を使用すると、別のノードからデータをアクセスする前に、そのノードが完全にオフラインであることを確認できます。

関連情報

- Fencing in Red Hat High Availability Cluster

4.13. フェンスデバイスの作成

フェンシングを設定するには、以下の手順を実行します。クラスターの任意のノードからこのコマンドを完了します。

前提条件

クラスタープロパティー **stonith-enabled** を **true** に設定する必要があります。

手順

1. 各 RHEL 仮想マシンの Azure ノード名を特定します。Azure ノード名を使用してフェンスデバイスを設定します。

```
# fence_azure_arm \
    -l <AD-Application-ID> -p <AD-Password> \
    --resourceGroup <MyResourceGroup> --tenantId <Tenant-ID> \
    --subscriptionId <Subscription-ID> -o list
```

以下に例を示します。

```
[root@node01 clouduser]# fence_azure_arm \
-l e04a6a49-9f00-xxxx-xxxx-a8bdda4af447 -p
z/a05AwCN0IzAjVwXXXXXXXXEWIoeVp0xg7QT//JE=
--resourceGroup azrhelclirsgrp --tenantId 77ecefb6-cff0-XXXX-XXXX-757XXXX9485
--subscriptionId XXXXXXXX-38b4-4527-XXXX-012d49dfc02c -o list

node01,
node02,
node03,
```

2. Azure ARM STONITH エージェントのオプションを表示します。

```
# pcs stonith describe fence_azure_arm
```

以下に例を示します。

```
# pcs stonith describe fence_apc
```

```
Stonith options:
password: Authentication key
password_script: Script to run to retrieve password
```

警告

method オプションを提供するフェンスエージェントでは、cycle の値を指定しないため、データの破損が発生する可能性があるため、この値は指定しないでください。

1 つのノードのみをフェンスできるフェンスデバイスや、複数のノードをフェンスできるデバイスもあります。フェンスデバイスの作成時に指定するパラメーターは、フェンスデバイスが対応しているか、必要としているかにより異なります。

フェンスデバイスの作成時に **pcmk_host_list** パラメーターを使用すると、フェンスデバイスで制御されるすべてのマシンを指定できます。

フェンスデバイスの作成時に **pcmk_host_map** パラメーターを使用すると、フェンスデバイスに関する仕様にホスト名をマッピングできます。

3. フェンスデバイスを作成します。

```
# pcs stonith create clusterfence fence_azure_arm
```

検証

1. 他のノードのいずれかに対してフェンスエージェントをテストします。

```
# pcs stonith fence azurenodename
```

以下に例を示します。

```
[root@node01 clouduser]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: node01 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Fri Feb 23 11:44:35 2018
Last change: Fri Feb 23 11:21:01 2018 by root via cibadmin on node01

3 nodes configured
1 resource configured

Online: [ node01 node03 ]
OFFLINE: [ node02 ]

Full list of resources:

  clusterfence  (stonith:fence_azure_arm):  Started node01
```

```
Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

2. 前の手順でフェンシングされたノードを起動します。

```
# pcs cluster start <hostname>
```

3. ステータスを確認して、ノードが起動したことを確認します。

```
# pcs status
```

以下に例を示します。

```
[root@node01 clouduser]# pcs status
Cluster name: newcluster
Stack: corosync
Current DC: node01 (version 1.1.18-11.el7-2b07d5c5a9) - partition with quorum
Last updated: Fri Feb 23 11:34:59 2018
Last change: Fri Feb 23 11:21:01 2018 by root via cibadmin on node01

3 nodes configured
1 resource configured

Online: [ node01 node02 node03 ]

Full list of resources:

clusterfence    (stonith:fence_azure_arm):  Started node01

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

関連情報

- Fencing in a Red Hat High Availability Cluster
- フェンシングデバイスの一般的なプロパティー

4.14. AZURE 内部ロードバランサーの作成

Azure 内部ロードバランサーは、ヘルスプローブ要求に応答しないクラスターノードを削除します。

以下の手順を実行し、Azure 内部ロードバランサーを作成します。各ステップは特定の Microsoft 手順を参照し、HA のロードバランサーをカスタマイズするための設定が含まれます。

前提条件

Azure コントロールパネル

手順

1. 基本ロードバランサーを作成 します。IP アドレスの割り当てタイプの場合は、**内部ロードバランサー**、**基本 SKU**、および **動的** を選択します。

2. バックエンドのアドレスプールを作成します。バックエンドプールを HA に Azure リソースを作成した時に作成された可用性セットに関連付けます。ターゲットネットワーク IP 設定は設定しないでください。

3. ヘルスプローブを作成します。ヘルスプローブの場合は **TCP** を選択し、ポート **61000** を入力します。別のサービスに干渉しない TCP ポート番号を使用できます。特定の HA 製品アプリケーション (SAP HANA や SQL Server など) については、Microsoft と連携して使用する正しいポートを指定する必要がある場合があります。

4. ロードバランサールールを作成します。ロードバランシングルールを作成する場合は、デフォルト値が事前に設定されます。**フローティング IP (ダイレクトサーバーを返す)** を **有効** に設定してください。

4.15. ロードバランサーリソースエージェントの設定

ヘルスプローブを作成したら、**ロードバランサー** リソースエージェントを設定する必要があります。このリソースエージェントは、Azure ロードバランサーからヘルスプローブ要求に応答し、要求に応答しないクラスターノードを削除するサービスを実行します。

手順

1. 全ノードに **nmap-ncat** リソースエージェントをインストールします。

```
# dnf install nmap-ncat resource-agents-cloud
```

単一ノードで以下の手順を実行します。

2. **pcs** リソースおよびグループを作成します。IPaddr2 アドレスにロードバランサーの FrontendIP を使用します。

```
# pcs resource create resource-name IPaddr2 ip="10.0.0.7" --group cluster-resources-group
```

3. **ロードバランサー** リソースエージェントを設定します。

```
# pcs resource create resource-loadbalancer-name azure-lb port=port-number --group cluster-resources-group
```

検証

- **pcs status** を実行して結果を表示します。

```
[root@node01 clouduser]# pcs status
```

出力例:

```
Cluster name: clusterfence01
Stack: corosync
Current DC: node02 (version 1.1.16-12.el7_4.7-94ff4df) - partition with quorum
Last updated: Tue Jan 30 12:42:35 2018
Last change: Tue Jan 30 12:26:42 2018 by root via cibadmin on node01
```

```
3 nodes configured
3 resources configured

Online: [ node01 node02 node03 ]

Full list of resources:

clusterfence (stonith:fence_azure_arm):      Started node01
Resource Group: g_azure
    vip_azure  (ocf::heartbeat:IPaddr2):       Started node02
    lb_azure   (ocf::heartbeat:azure-lb):      Started node02

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

4.16. 共有ブロックストレージの設定

Microsoft Azure 共有ディスクを使用して Red Hat High Availability クラスターの共有ブロックストレージを設定するには、次の手順を使用します。この手順はオプションであり、以下の手順では、1 TB の共有ディスクを備えた 3 つの Azure 仮想マシン (3 ノードクラスター) を想定していることに注意してください。

注記

これは、ブロックストレージを設定するスタンドアロンの例です。この手順では、クラスターを作成していないことを前提としています。

前提条件

- ホストシステムに Azure CLI をインストールし、SSH キーを作成している。
- 以下リソースの作成を含むクラスター環境を Azure で作成している。リンクは、Microsoft Azure のドキュメントにあります。
  - リソースグループ
  - 仮想ネットワーク
  - ネットワークセキュリティーグループ
  - ネットワークセキュリティーグループルール
  - サブネット
  - ロードバランサー (オプション)
  - ストレージアカウント
  - 近接配置グループ
  - 可用性セット

手順

1. Azure コマンド **az disk create** を使用して、共有ブロックボリュームを作成します。

```
$ az disk create -g <resource_group> -n <shared_block_volume_name> --size-gb
<disk_size> --max-shares <number_vms> -l <location>
```

たとえば、以下のコマンドは、Azure Availability Zone **westcentralus** 内 のリソースグループ **sharedblock** に **shared-block-volume.vhd** という名前の共有ブロックボリュームを作成します。

```
$ az disk create -g sharedblock-rg -n shared-block-volume.vhd --size-gb 1024 --max-shares
3 -l westcentralus

{
  "creationData": {
    "createOption": "Empty",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": null,
    "sourceUniqueId": null,
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
  "diskAccessId": null,
  "diskIopsReadOnly": null,
  "diskIopsReadWrite": 5000,
  "diskMbpsReadOnly": null,
  "diskMbpsReadWrite": 200,
  "diskSizeBytes": 1099511627776,
  "diskSizeGb": 1024,
  "diskState": "Unattached",
  "encryption": {
    "diskEncryptionSetId": null,
    "type": "EncryptionAtRestWithPlatformKey"
  },
  "encryptionSettingsCollection": null,
  "hyperVgeneration": "V1",
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/sharedblock-
rg/providers/Microsoft.Compute/disks/shared-block-volume.vhd",
  "location": "westcentralus",
  "managedBy": null,
  "managedByExtended": null,
  "maxShares": 3,
  "name": "shared-block-volume.vhd",
  "networkAccessPolicy": "AllowAll",
  "osType": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "sharedblock-rg",
  "shareInfo": null,
  "sku": {
    "name": "Premium_LRS",
    "tier": "Premium"
  },
  "tags": {},
```

```
  "timeCreated": "2020-08-27T15:36:56.263382+00:00",
  "type": "Microsoft.Compute/disks",
  "uniqueId": "cd8b0a25-6fbe-4779-9312-8d9cbb89b6f2",
  "zones": null
}
```

2. Azure コマンド **az disk show** を使用して、共有ブロックボリュームが作成されたことを確認します。

```
$ az disk show -g <resource_group> -n <shared_block_volume_name>
```

たとえば、次のコマンドは、リソースグループ **sharedblock-rg** 内の共有ブロックボリューム **shared-block-volume.vhd** の詳細を表示します。

```
$ az disk show -g sharedblock-rg -n shared-block-volume.vhd

{
  "creationData": {
    "createOption": "Empty",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": null,
    "sourceUniqueId": null,
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
  "diskAccessId": null,
  "diskIopsReadOnly": null,
  "diskIopsReadWrite": 5000,
  "diskMbpsReadOnly": null,
  "diskMbpsReadWrite": 200,
  "diskSizeBytes": 1099511627776,
  "diskSizeGb": 1024,
  "diskState": "Unattached",
  "encryption": {
    "diskEncryptionSetId": null,
    "type": "EncryptionAtRestWithPlatformKey"
  },
  "encryptionSettingsCollection": null,
  "hyperVgeneration": "V1",
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/sharedblock-
rg/providers/Microsoft.Compute/disks/shared-block-volume.vhd",
  "location": "westcentralus",
  "managedBy": null,
  "managedByExtended": null,
  "maxShares": 3,
  "name": "shared-block-volume.vhd",
  "networkAccessPolicy": "AllowAll",
  "osType": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "sharedblock-rg",
  "shareInfo": null,
  "sku": {
    "name": "Premium_LRS",
```

```
    "tier": "Premium"
  },
  "tags": {},
  "timeCreated": "2020-08-27T15:36:56.263382+00:00",
  "type": "Microsoft.Compute/disks",
  "uniqueId": "cd8b0a25-6fbe-4779-9312-8d9cbb89b6f2",
  "zones": null
}
```

3. Azure コマンド **az network nic create** を使用して、3 つのネットワークインターフェイスを作成します。それぞれ異なる **<nic_name>** を使用して、以下のコマンドを 3 回実行します。

```
$ az network nic create \
    -g <resource_group> -n <nic_name> --subnet <subnet_name> \
    --vnet-name <virtual_network> --location <location> \
    --network-security-group <network_security_group> --private-ip-address-version IPv4
```

たとえば、以下のコマンドは、**shareblock-nodea-vm-nic-protected** という名前のネットワークインターフェイスを作成します。

```
$ az network nic create \
    -g sharedblock-rg -n sharedblock-nodea-vm-nic-protected --subnet sharedblock-subnet-protected \
    --vnet-name sharedblock-vn --location westcentralus \
    --network-security-group sharedblock-nsg --private-ip-address-version IPv4
```

4. Azure コマンド **az vm create** を使用して 3 つの仮想マシンを作成し、共有ブロックボリュームを割り当てます。オプションの値は、各仮想マシンに独自の **<vm_name>**、**<new_vm_disk_name>** および **<nic_name>** が指定される点で異なります。

```
$ az vm create \
    -n <vm_name> -g <resource_group> --attach-data-disks <shared_block_volume_name> \
    --data-disk-caching None --os-disk-caching ReadWrite --os-disk-name <new-vm-disk-name> \
    --os-disk-size-gb <disk_size> --location <location> --size <virtual_machine_size> \
    --image <image_name> --admin-username <vm_username> --authentication-type ssh \
    --ssh-key-values <ssh_key> --nics <nic_name> --availability-set <availability_set> --ppg <proximity_placement_group>
```

たとえば、次のコマンドは、**sharedblock-nodea-vm** という名前の仮想マシンを作成します。

```
$ az vm create \
-n sharedblock-nodea-vm -g sharedblock-rg --attach-data-disks shared-block-volume.vhd \
--data-disk-caching None --os-disk-caching ReadWrite --os-disk-name sharedblock-nodea-vm.vhd \
--os-disk-size-gb 64 --location westcentralus --size Standard_D2s_v3 \
--image /subscriptions/12345678910-12345678910/resourceGroups/sample-azureimagesgroupwestcentralus/providers/Microsoft.Compute/images/sample-azure-rhel-9.3.0-20200713.n.0.x86_64 --admin-username sharedblock-user --authentication-type ssh \
--ssh-key-values @sharedblock-key.pub --nics sharedblock-nodea-vm-nic-protected --availability-set sharedblock-as --ppg sharedblock-ppg

{
  "fqdns": "",
```

```
  "id": "/subscriptions/12345678910-12345678910/resourceGroups/sharedblock-
rg/providers/Microsoft.Compute/virtualMachines/sharedblock-nodea-vm",
  "location": "westcentralus",
  "macAddress": "00-22-48-5D-EE-FB",
  "powerState": "VM running",
  "privateIpAddress": "198.51.100.3",
  "publicIpAddress": "",
  "resourceGroup": "sharedblock-rg",
  "zones": ""
}
```

検証

1. クラスター内の VM ごとに、**ssh** コマンドと VM の IP アドレスを使用して、ブロックデバイスが使用可能であることを確認します。

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T '"
```

たとえば、次のコマンドは、仮想マシン IP **198.51.100.3** のホスト名およびブロックデバイスを含む詳細をリスト表示します。

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T '"

nodea
sdb    8:16   0    1T  0 disk
```

2. **SSH** コマンドを使用して、クラスター内の各仮想マシンが同じ共有ディスクを使用していることを確認します。

```
# ssh <ip_address> "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info
--query=all --name=/dev/{} | grep '^E: ID_SERIAL='"
```

たとえば、以下のコマンドは、インスタンスの IP アドレスが **198.51.100.3** のホスト名および共有ディスクボリューム ID が含まれる詳細をリスト表示します。

```
# ssh 198.51.100.3 "hostname ; lsblk -d | grep ' 1T ' | awk '{print \$1}' | xargs -i udevadm info -
-query=all --name=/dev/{} | grep '^E: ID_SERIAL='"

nodea
E: ID_SERIAL=3600224808dd8eb102f6ffc5822c41d89
```

共有ディスクが各仮想マシンに割り当てられていることを確認したら、クラスターの回復性の高いストレージを設定できます。

関連情報

- クラスターに GFS2 ファイルシステムを設定
- GFS2 ファイルシステムの設定

4.17. 関連情報

- Support Policies for RHEL High Availability Clusters - Microsoft Azure Virtual Machines as Cluster Members
- 高可用性クラスターの設定および管理