



Red Hat Enterprise Linux 7

仮想化スタートガイド

仮想化の概念について

Red Hat Enterprise Linux 7 仮想化スタートガイド

仮想化の概念について

Jiri Herrmann
Red Hat Customer Content Services
jherrman@redhat.com

Yehuda Zimmerman
Red Hat Customer Content Services
yzimmerm@redhat.com

Dayle Parker
Red Hat Customer Content Services

Laura Novich
Red Hat Customer Content Services

Jacquelynn East
Red Hat Customer Content Services

Scott Radvan
Red Hat Customer Content Services

法律上の通知

Copyright © 2018 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, 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 Software Collections 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 仮想化スタートガイドは、仮想化の基本事項とRed Hat Enterprise Linuxで利用可能な仮想化製品および技術について説明しています。

目次

第1章 仮想化の概説	3
1.1. 仮想化とは	3
1.2. 仮想化ソリューション	3
第2章 仮想化を使用する理由	5
2.1. 仮想化コスト	5
2.2. パフォーマンス	5
2.3. マイグレーション	6
2.4. セキュリティー	7
2.5. 障害復旧	8
第3章 RED HAT VIRTUALIZATION 製品および機能について	9
3.1. RED HAT ENTERPRISE LINUX での KVM および仮想化	9
3.2. LIBVIRT および LIBVIRT ツール	12
3.3. ドメイン XML 設定	12
3.4. 仮想化ハードウェアデバイス	13
3.5. ストレージ	19
3.6. 仮想ネットワーク	21
第4章 仮想化コマンドラインインターフェースの使用	23
4.1. 仮想化の主なコマンドラインユーティリティー	23
4.2. デモンストレーション: コマンドラインユーティリティーによるゲストの作成および管理	27
第5章 仮想マシンマネージャーの使用	34
5.1. 仮想マシンマネージャーの実行	34
5.2. 仮想化マシンマネージャーのインターフェース	35
付録A 改訂履歴	46

第1章 仮想化の概説

1.1. 仮想化とは

仮想化とは、単一システム上で他のプログラムと切り離して、ソフトウェア (通常、複数のオペレーティングシステム) を同時に実行することを指す広範囲なコンピューター用語です。仮想化は、ハイパーバイザーを使用することで実現されます。ハイパーバイザーとは、単一の (通常は物理) マシン上でハードウェアを制御して、仮想マシン (VM) または ゲスト と呼ばれる複数のオペレーティングシステムを実行可能にするソフトウェア層またはサブシステムのことです。オペレーティングシステムのあるマシンは ホスト と呼ばれます。詳細は [Red Hat カスタマーポータル](#) を参照してください。

仮想化にはいくつかの方法があります。

完全仮想化

完全仮想化では、ゲストオペレーティングシステムに手を加えずにそのまま使用します。ゲストは、ハイパーバイザーが作成したチャンネル経由でホストの CPU のアドレスを指定します。完全仮想化では、ゲストが CPU と直接通信するので、仮想化の手法で最速となっています。

準仮想化

準仮想化では、ゲストオペレーティングシステムを変更して使用します。ゲストは、ハイパーバイザーと通信します。ハイパーバイザーは、変更されていないゲストからの呼び出しを、物理および仮想の CPU や他のインターフェースに渡します。呼び出しがハイパーバイザー経由でルーティングされるので、この手法は完全仮想化よりも遅くなります。

ソフトウェアの仮想化 (またはエミュレーション)

ソフトウェアの仮想化は、バイナリー変換と他のエミュレーション技術を使用して変更されていないオペレーティングシステムを実行します。ハイパーバイザーは、ゲストの呼び出しをホストシステムで使用できる形式に変換します。呼び出しがすべて変換されるので、この手法は仮想化より遅くなります。Red Hat では、Red Hat Enterprise Linux 上のソフトウェアの仮想化はサポートしていません。

コンテナ化

KVM の仮想化は OS カーネルの個別インスタンスを作成しますが、コンテナ化と呼ばれるオペレーティングシステムレベルの仮想化は、既存の OS カーネルの上で動作し、コンテナと呼ばれるホスト OS の分離されたインスタンスを作成します。コンテナに関する詳細は、[Red Hat カスタマーポータル](#) を参照してください。

コンテナには KVM 仮想化の多機能性はありませんが、より軽量で処理の柔軟性に優れています。比較の詳細は、「[Introduction to Linux Containers](#)」を参照してください。

Red Hat Enterprise Linuxでコンテナを使用するには、[Extras チャンネル](#)から [docker パッケージ](#)をインストールします。Red Hat は、[Red Hat Enterprise Linux Atomic Host](#) や [Red Hat OpenShift Container Platform](#) など、コンテナを使用するために最適化されたソリューションを提供します。コンテナサポートの詳細は、[Red Hat ナレッジベース](#)を参照してください。

1.2. 仮想化ソリューション

Red Hat が提供する主な仮想ソリューションは次のとおりです。それぞれ対象ユーザーや機能が異なります。

Red Hat Enterprise Linux

Red Hat Enterprise Linux 7 には、仮想マシンの作成、実行、管理機能や [さまざまな仮想化ツールや機能](#) が含まれます。このソリューションのサポートは、ホストごとに実行されるゲスト数やゲストタイプの種類の面で制限があるので、Red Hat Enterprise Linux 上の仮想化は複数環境でテストする必要のある開発者や、厳密な稼働時間の要件やサービスレベルアグリーメント (SLA) のないサーバーを複数実行する小規模なビジネスなどには有用です。

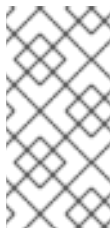


重要

このガイドでは Red Hat Enterprise Linux での仮想化について説明しますが、他の仮想化ソリューションの詳細については触れません。

Red Hat Virtualization

Red Hat Virtualization (RHV) は、Red Hat Enterprise Linux での仮想化のように Kernel-based Virtual Machine ([KVM](#)) 技術をベースにしていますが、数多くの強化機能を提供します。エンタープライズレベルのスケラビリティとパフォーマンスが得られるように設計されており、一元管理されたグラフィカルインターフェースからホスト、仮想マシン、ネットワーク、ストレージ、ユーザーなど仮想インフラストラクチャー全体を管理することができます。



注記

Red Hat Enterprise Linux での仮想化と Red Hat Virtualization の違いの詳細については、Red Hat カスタマーポータルソリューション「[What are the differences between qemu-kvm and qemu-kvm-rhev and all sub-packages?](#)」を参照してください。

Red Hat Virtualization は、大規模なデプロイメントやミッションクリティカルなアプリケーションを実行する企業で 사용할 ことができます。Red Hat Virtualization に適した大規模なデプロイメントには、ダウンタイムなしで継続して稼働する必要のあるデータベース、取引プラットフォーム、メッセージングシステムなどが含まれます。

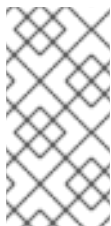


注記

Red Hat Virtualization について詳しく知る、または完全サポート付き 60 日間の評価版をダウンロードするには、<http://www.redhat.com/ja/technologies/virtualization/enterprise-virtualization> をご覧ください。あるいは、「[Product Documentation for Red Hat Virtualization](#)」を参照してください。

Red Hat OpenStack Platform

Red Hat OpenStack Platform は、セキュアで信頼性の高いパブリックまたはプライベートの [OpenStack](#) クラウドを作成、デプロイ、スケーリングするための統合された基盤を提供します。



注記

Red Hat OpenStack Platform について詳しく知る、または 60 日間の評価版をダウンロードするには、<https://www.redhat.com/ja/technologies/linux-platforms/openstack-platform> をご覧ください。あるいは、「[Product Documentation for Red Hat OpenStack Platform](#)」を参照してください。

第2章 仮想化を使用する理由

仮想化は、サーバーのデプロイメントにも個別のデスクトップステーションにも有用です。デスクトップの仮想化は、一元管理のコストを効率化し、障害復旧を改善します。さらに、`ssh` などの接続ツールを使用すると、デスクトップをリモートから接続することができます。

サーバー用に使用する場合には、仮想化は大規模なネットワークだけでなく、複数のサーバーが含まれるデプロイメントにも利益をもたらします。また、仮想化により、ライブマイグレーションや高可用性、耐障害性、バックアップの効率化が可能です。

2.1. 仮想化コスト

仮想化の導入は費用がかかることもありますが、長期的には節約になる場合が多くあります。以下の利点を検討してみてください。

省電力

仮想化により、複数の物理プラットフォームの必要性がほとんどなくなります。これは、マシン稼働や冷却に使われる電力の節約を意味し、エネルギー費用の削減につながります。マシンの消費電力と必要となる冷却を合わせると、複数の物理プラットフォーム購入の初期費用は、仮想化の利用により大幅に削減されます。

メンテナンスの省力化

物理システムから仮想化システムへの移行前に十分なプランニングを行うことで、メンテナンスにかかる時間は少なくなります。つまり、部品にかかる費用や人件費が削減されることになります。

インストール済みソフトウェアの使用期限の延長

以前のバージョンのソフトウェアは、より新しいベアメタルマシン上では直接機能しない可能性があります。以前のソフトウェアをより大規模かつ高速なシステム上で仮想的に実行することで、新しいシステムのパフォーマンスを活用する一方でソフトウェアの使用期限を延長することができます。

予測可能なコスト

Red Hat Enterprise Linux のサブスクリプションは、仮想化サポートが固定料金なので、コストの予測が容易になります。

スペースの節約

マシンの台数を減らすようにサーバーを統合すると、必要な物理的スペースが減ります。

2.2. パフォーマンス

シングル CPU しかサポートしていなかった以前の仮想化バージョンでは、明らかに仮想マシンのパフォーマンスに制約がありました。これにより、仮想化ソリューションは遅い、という長期にわたる誤解が生み出されました。

しかし、最近の状況は変わりました。最新の仮想化技術により、仮想マシンのスピードは格段に速くなりました。仮想マシンはベアメタルシステムとほぼ同程度の効率性で通常のサーバーアプリケーションを実行できると、ベンチマークで示されています。

- Red Hat Enterprise Linux 6.4 および KVM は、IBM DB2 データベースと共に完全に仮想化された x86 環境で実行され、ベアメタルパフォーマンスの 88% を実現し、[業界をリードする TPC-C ベンチマーク](#) で結果を記録しました。データベースは、これまでリソース需要が高いため

に、ベアメタルのデプロイメント向けのみには確保されていました。

- 業界標準である SAP Sales and Distribution (SD) Standard Application Benchmark は、Red Hat Enterprise Linux 6.2 および KVM を同一ハードウェア上で稼働するベアメタルシステムと比較した場合に [85% の仮想化効率](#) を発揮したとしています。
- Red Hat Enterprise Linux 6.1 および KVM は、標準性能評価法人 (Standard Performance Evaluation Corporation) が行った [SPECvirt_sc2010 ベンチマーク](#) で記録的な仮想化パフォーマンスを達成しました。これは公開されている SPECvirt の結果のなかでは最高の仮想パフォーマンス得点となりました。SPECvirt_sc2010 メトリックは、仮想化データセンターサーバー内にあるシステムコンポーネントのエンドツーエンドのパフォーマンスを測定するものです。



注記

仮想化のパフォーマンスチューニングに関する詳細は、[『Red Hat Enterprise Linux 7 仮想化のチューニングと最適化ガイド』](#)を参照してください。

2.3. マイグレーション

マイグレーションとは、ゲスト仮想マシンを任意のホストから別のホストに移行するプロセスです。仮想マシンが直接ハードウェア上で実行するのではなく仮想化環境内で実行するために、このマイグレーションが可能となります。仮想マシンのマイグレーションには、ライブとオフラインの2つの方法があります。

マイグレーションのタイプ

オフラインマイグレーション

オフラインマイグレーションでは、ゲスト仮想マシンを一時停止させて、仮想マシンのメモリのイメージを目的のホストに移動します。次に、目的のホスト上で仮想マシンを再開し、仮想マシンがソースホスト上で使用していたメモリーを解放します。

ライブマイグレーション

ライブマイグレーションは、アクティブな仮想マシンを任意の物理ホストから別の物理ホストに移行するプロセスです。詳しくは、[『仮想化の導入および管理ガイド』](#)を参照してください。

2.3.1. 仮想マシンのマイグレーションの利点

マイグレーションは以下の点で役立ちます。

負荷の分散

ホストマシンがオーバーロードとなった場合には、仮想マシンは単独でも多数でも、ライブマイグレーションで他のホストに移行できます。同様に、実行されていないマシンでオーバーロードの傾向にある場合には、オフラインマイグレーションで移行することができます。

ホストのアップグレードやホストに変更を加える場合

任意のホスト上にあるハードウェアデバイスにアップグレード、追加、削除などの必要性が生じた場合には、仮想マシンを安全に他のホストに移動させることができます。そのため、いずれかのホストへの変更が原因で、ゲスト側でダウンタイムが生じることはありません。

省エネ

仮想マシンを別のホストに再配分することで、アンロードされたホストシステムの電源を電力使用量の少ない時間帯にオフにして節電とコスト削減が可能になります。

地理的なマイグレーション

仮想マシンは、レイテンシーの短縮または他の理由で別の物理的な場所に移動することが可能です。

移行プロセスで仮想マシンのメモリーが移動されると、その仮想マシンに関連付けられたディスクボリュームも移行されます。このプロセスは、ライブブロックマイグレーションを使用して実行されます。



注記

移行に関する詳細は、[『Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド』](#)を参照してください。

2.3.2. V2V (Virtual to Virtual) の移行

Red Hat Enterprise Linux 7 は、仮想マシンを他のタイプのハイパーバイザーから KVM に変換するツールを提供します。**virt-v2v** ツールで、Xen、KVM の別バージョン、VMware ESX からの仮想マシンを変換してインポートします。



注記

V2V に関する詳細は、[V2V のナレッジ記事](#)を参照してください。

さらに、Red Hat Enterprise Linux 7.3 以降では、**virt-p2v** ツールを使用した P2V (Physical to Virtual) の変換をサポートします。詳細は [P2V のナレッジ記事](#)を参照してください。

2.4. セキュリティー

KVM 仮想マシンでは、セキュリティを強化するために以下の機能が使用されています。

SELinux

SELinux (Security-Enhanced Linux) は、Linux システムすべてに対して Mandatory Access Control (MAC) を提供するので、Linux ゲストも同様にこの機能の恩恵を受けることができます。SELinux の制御下では、全プロセスおよびファイルに **タイプ** が指定され、さまざまなタイプできめ細かく管理することでシステム上のアクセスが制限されます。SELinux は攻撃者の行動を制限し、バッファオーバーフロー攻撃や権限の昇格といった多くの一般的なセキュリティの悪用を防ぎます。

sVirt

sVirt とは、Red Hat Enterprise Linux 7 に導入されている、SELinux と仮想化を統合する技術のことです。仮想マシンの使用時には Mandatory Access Control (MAC) を適用してセキュリティを強化し、ホストや別の仮想マシンへの攻撃経路として使用される可能性のあるハイパーバイザー内のバグに対してシステムを堅牢にします。



注記

仮想化のセキュリティに関する詳細は、[『Red Hat Enterprise Linux 7 仮想化セキュリティガイド』](#)を参照してください。

2.5. 障害復旧

システムが仮想化されていると、障害からの回復がより迅速で容易になります。物理システムでは、何らかの重大な障害が発生すると、通常オペレーティングシステムを完全に再インストールする必要が生じ、回復時間は何時間にも及びます。しかしシステムが仮想化されていれば、マイグレーション機能のおかげで回復ははるかに速くなります。ライブマイグレーションの要件に従っている場合には、仮想マシンは別のホスト上で再起動が可能になり、最大の遅延の理由として考えられるのは、ゲストデータの復元などでしょう。また、仮想化システムはそれぞれ完全に独立しているので、1つのシステムのダウンタイムが他のシステムに影響することはありません。

第3章 RED HAT VIRTUALIZATION 製品および機能について

本章では、Red Hat Enterprise Linux 7で利用可能な主要仮想化製品をご紹介します。

3.1. RED HAT ENTERPRISE LINUX での KVM および仮想化

KVM (Kernel-based Virtual Machine) は、さまざまなアーキテクチャー上の Linux 向けの完全仮想化ソリューションのことです。KVM は標準の Red Hat Enterprise Linux 7 カーネルに組み込まれ、Quick Emulator (QEMU) に統合されており、ゲストオペレーティングシステムを複数実行することができます。Red Hat Enterprise Linux の KVM ハイパーバイザーは、**libvirt** API および (**virt-manager** や **virsh** などの) **libvirt** 向けにビルドされたツールで管理されます。仮想マシンは、マルチスレッドの Linux プロセスとして実行/稼働され、これらのツールによって管理されます。



警告

QEMU および **libvirt** は、QEMU Tiny Code Generator (TCG) を使用する動的変換モードをサポートします。このモードでは、ハードウェアの仮想化サポートは不要です。ただし、この構成は Red Hat ではサポートされていません。

この制約に関する詳細な情報については、[『Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド』](#)を参照してください。

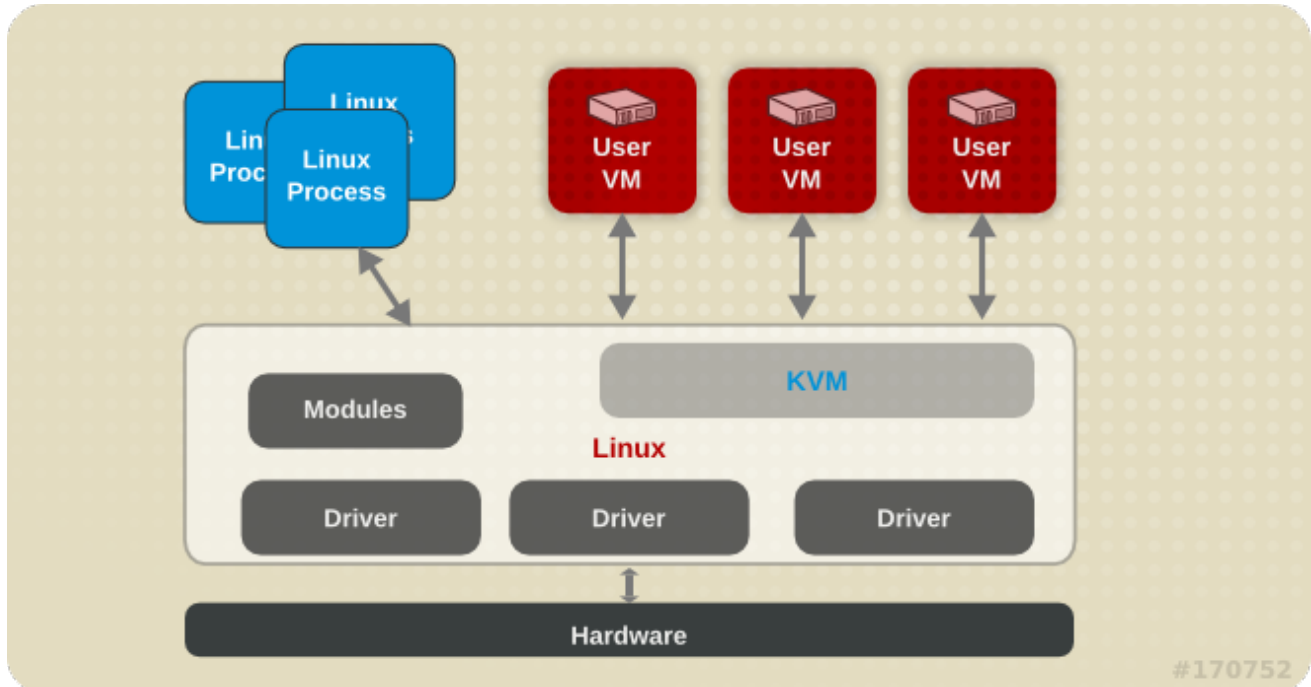


図3.1 KVM アーキテクチャー

Red Hat Enterprise 7 上の KVM でサポートされる仮想化機能には以下が含まれます。

オーバーコミット

KVM ハイパーバイザーはシステムリソースの **オーバーコミット** をサポートします。オーバーコミットとは、システムで利用可能なリソースよりも多い仮想化 CPU またはメモリーを割り当てるこ

とで、あるゲストで必要とされるリソースが別のゲストでは使用されていない場合に、ダイナミックにスワップできます。これにより、ゲストによるホストのリソースの使用の効率性を改善し、必要なホスト数を削減することができます。



重要

オーバーコミットは、システムの安定性へのリスクとなる可能性があります。KVM でのオーバーコミットと必要な予防措置については『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

KSM

KVM ハイパーバイザーが使用する *Kernel Same-page Merging (KSM)* により、KVM ゲストは同一メモリーページを共有することが可能になります。このような共有ページは通常、共通のライブラリーか、使用頻度の高い同一データです。KSM はメモリー重複を防ぐことで同一または類似のゲストオペレーティングシステムのゲスト密度を高めることができます。



注記

KSM に関する詳細は、『[Red Hat Enterprise Linux 7 仮想化のチューニングと最適化ガイド](#)』を参照してください。

QEMU ゲストエージェント

QEMU ゲストエージェントはゲストオペレーティングシステム上で稼働し、ホストマシンによるゲストオペレーティングシステムへのコマンド発行を可能にします。



注記

QEMU ゲストエージェントに関する詳しい情報は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

ディスク I/O スロットリング

複数の仮想マシンが同時に実行される場合、それらは過剰なディスク I/O を使用することでシステム全体のパフォーマンスに影響を与える可能性があります。KVM の *ディスク I/O スロットリング* は、個別の仮想マシンからホストマシンに送られるディスク I/O 要求に制限を設定する機能を提供します。これにより、1 台の仮想マシンが共有リソースを過剰に使用し、他の仮想マシンのパフォーマンスに影響を与えることを防ぐことができます。



注記

ディスク I/O スロットリングを使用する方法については、『[Red Hat Enterprise Linux 7 仮想化のチューニングと最適化ガイド](#)』を参照してください。

NUMA の自動負荷分散

Non-Uniform Memory Access (NUMA) 自動負荷分散 は、スレッドやプロセスなどのタスクをアクセスするメモリーの近くに移動します。これにより、Red Hat Enterprise Linux 7 ゲストに必要な手動のチューニングなしに、Non-Uniform Memory (NUMA) ハードウェアシステム上で実行中のアプリケーションのパフォーマンスが向上されます。



注記

NUMA の自動負荷分散に関する詳細は、『[Red Hat Enterprise Linux 7 仮想化のチューニングと最適化](#)』を参照してください。

仮想 CPU ホットアド

仮想 CPU (vCPU) ホットアド機能があれば、ゲストをシャットダウンせずに、必要に応じて仮想マシンの実行の際に処理能力を向上できます。必要なワークロードに対応したり、ワークロード関連のサービスレベルアグリーメント (SLA) を確保したりするために、仮想マシンに割り当てられた vCPU を実行中のゲストに追加することができます。



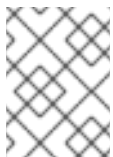
注記

仮想 CPU に関する詳しい情報は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

仮想化のネスト

Red Hat Enterprise Linux 7.2 以降では、ネスト化したハードウェア仮想化支援機能をテクノロジープレビューとして提供しています。この機能では、KVM ゲストがハイパーバイザーとして動作して、それ自体のゲストを作成できるようになります。

たとえば、仮想マシン上のハイパーバイザーをデバッグしたり、数に制限のある物理マシン上の規模の大きい仮想デプロイメントをテストしたりするために使用できます。



注記

ネスト化された仮想化の設定および使用に関する詳細は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

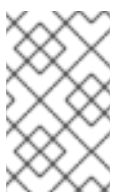
KVM ゲスト仮想マシンの互換性

Red Hat Enterprise Linux 7 サーバーには、サポートに関する制限がいくつかあります。

Red Hat Enterprise Linux のプロセッサおよびメモリー容量の制限については、以下のサイトで説明されています。

- ホストシステムについては、<https://access.redhat.com/ja/articles/1271503> を参照してください。
- KVM ハイパーバイザーについては、<https://access.redhat.com/site/articles/rhel-kvm-limits> を参照してください。

サポートされるオペレーティングシステムとホストおよびゲストの組み合わせの詳細な対応表については、[Red Hat カスタマーポータル](#)を参照してください。



注記

ご使用のプロセッサが仮想化拡張に対応しているかどうかの確認方法および無効になっている仮想化拡張の有効化に関する情報については、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

3.2. LIBVIRT および LIBVIRT ツール

libvirt パッケージはハイパーバイザーに依存しない仮想化 API で、幅広いオペレーティングシステムの仮想化機能との対話が可能です。

- ホスト上の仮想マシンをセキュアに管理する仮想化層
- ローカルおよびネットワーク接続されたホストを管理するインターフェース
- 仮想マシンのプロビジョニング、作成、修正、監視、制御、移行、停止に必要な API。**libvirt** で複数のホストに同時アクセスはできますが、API はシングルノード操作に限られています。



注記

libvirt を使用する場合には、ハイパーバイザーがサポートする操作のみを実行できます。

libvirt は単一ホストの管理にフォーカスし、CPU やメモリー、ストレージ、ネットワーキング、NUMA (Non-Uniform Memory Access) パーティションを含む管理ノード上で利用可能なリソースのエミュレート、監視、使用を行う API を提供します。この管理ツールは、ホストが実行されているマシンと同じ物理マシンに配置する必要はありません。このようなシナリオでは、管理ツールを実行するマシンは、セキュアなプロトコルを使用して、ホストを実行するマシンと通信します。

Red Hat Enterprise Linux 7 は **libvirt** に対応しており、(Red Hat Virtualization Management 同様に) デフォルトの仮想化管理手段として **libvirt** ベースのツールが含まれています。

libvirt パッケージは、GNU 一般公衆利用許諾契約書 (GPL) の下でフリーソフトウェアとして利用可能です。libvirt プロジェクトは、さまざまなハイパーバイザー技術上で稼働する仮想化管理ツールに長期的な安定性のある C API を提供することを目的としています。libvirt パッケージは、Red Hat Enterprise Linux 5 上の Xen と、Red Hat Enterprise Linux 5、6、7 上の KVM に対応しています。

特に、libvirt は Red Hat Enterprise Linux 7 上の仮想化を制御する主要なツール (**virsh** および **virt-manager**) も提供します。

3.3. ドメイン XML 設定

KVM ゲスト仮想マシン (ドメインとも呼ばれます) のすべてのホストベースの設定は、ドメイン XML 設定ファイル (または *ドメイン XML*) に格納されます。これには、仮想ハードウェア、ブートオプション、リソース割り当て、ネットワークインターフェイスなどの設定が含まれます。**virsh setmem** など、ゲストプロパティを永続的に変更するアプリケーションまたはユーティリティを使用すると、この設定がゲストの設定ファイルに書き込まれます。その後、設定ファイルはゲストの起動時にハイパーバイザーによって読み取られます。

特定のゲストの XML 設定を表示するには、**virsh dumpxml *guestname*** コマンドを使用します。

ゲストの XML 設定を編集するには、次のいずれかを使用します。

- **virsh** コマンド: **virsh** コマンドを使用してゲスト仮想マシンに追加された永続的な変更は、ドメイン XML に記録されます。
- **virt-xml** コマンド: このコマンドは提供されたオプションに応じて、指定のゲストのドメイン XML ファイルを設定します。
- **仮想マシンマネージャー**: **仮想マシンマネージャー** でゲスト仮想マシンに追加された変更は、ドメイン XML に記録されます。



注記

ドメイン XML ファイルを直接編集する必要があることがまれにありますが、この場合は **virsh edit guestname** コマンドを使用してください。このコマンドは、root の bash 設定によって決定されるテキストエディター (デフォルトは **vi**) に指定ゲストのドメイン XML 設定を開きます。

virsh edit のエディターを変更するには、対象のユーザーの **.bashrc** ファイルの **EDITOR** 変数を設定または変更します。たとえば、root ユーザーの場合、このファイルは **/root/**ディレクトリーにあります。

virsh edit を使用して行った変更を反映するには、変更した XML 設定を保存し、ゲストを再起動します。



警告

gedit /etc/libvirt/qemu/guestname.xml を使用するなど、テキストエディターでゲストの XML 設定をファイルとして開いて、ゲストの XML 設定を編集しないでください。このように加えられた変更は反映されず、自動的に上書きされます。

ドメイン XML 設定ファイルの詳細な情報については、『[仮想化の導入および管理ガイド](#)』を参照してください。

3.4. 仮想化ハードウェアデバイス

Red Hat Enterprise Linux 7 上の仮想化により、仮想マシンは以下の 3 種の異なるデバイスとして、ホストの物理ハードウェアを使用できるようになります。

- 仮想化デバイスおよびエミュレートされたデバイス
- 準仮想化デバイス
- 物理共有デバイス

これらのハードウェアデバイスはすべて、仮想化マシンに物理的にアタッチされたように表示されますが、デバイスドライバーの機能の仕方が異なります。

3.4.1. 仮想化デバイスおよびエミュレートされたデバイス

KVM は、ソフトウェアに多くの仮想マシン向けコアデバイスを実装します。これらのエミュレートされたハードウェアデバイスは、オペレーティングシステムの仮想化に必須のものです。エミュレートされたデバイスは、完全にソフトウェア内に存在する仮想デバイスのことです。

さらに、エミュレートされたドライバーは、物理デバイスまたは仮想ソフトウェアデバイスを使用することができます。エミュレートされたドライバーは、仮想マシンと (ソースデバイスを管理する) Linux カーネルの間の変換層です。デバイスレベルの指示は KVM ハイパーバイザーによって完全に変換されます。Linux カーネルで認識される、(ストレージやネットワーク、キーボード、マウスなど) 同じタイプのデバイスはいずれも、エミュレートされたドライバーのバックエンドデバイスとして使用可能です。

仮想 CPU (vCPU)

Red Hat Enterprise Linux 7.2 以降では、ホストの CPU の数にかかわらず、ホストシステムに最大 240 個の仮想 CPU (vCPU) を設定して、ゲストで使用できるように公開できます。この数は、Red Hat Enterprise Linux 7.0 の 160 個から増加しています。

エミュレートされたシステムコンポーネント

以下のコアシステムコンポーネントは基本的なシステム機能を提供するためにエミュレートされます。

- Intel i440FX ホスト PCI ブリッジ
- PIIX3 PCI to ISA ブリッジ
- PS/2 マウスおよびキーボード
- EvTouch USB グラフィックタブレット
- PCI UHCI USB コントローラーおよび仮想化 USB ハブ
- エミュレートされたシリアルポート
- EHCI コントローラー、仮想化 USB ストレージ、USB マウス
- USB 3.0 xHCI ホストコントローラー (Red Hat Enterprise Linux 7.3 のホストコントローラー)

エミュレートされたストレージドライバー

ストレージデバイスとストレージプールは、これらのエミュレートされたドライバーを使用してストレージデバイスを仮想マシンにアタッチすることができます。ゲストは、エミュレートされたストレージドライバーを使用してストレージプールにアクセスします。

すべての仮想デバイスと同様、ストレージドライバーはストレージデバイスではない点に注意してください。バックングストレージデバイスやファイル、ストレージプールボリュームなどを仮想ゲストにアタッチするために使用するのがドライバーです。対応する任意のタイプのストレージデバイス、ファイル、ストレージプールボリュームをバックングストレージデバイスにすることができます。

エミュレートされたIDE ドライバー

KVM は 2 種類のエミュレートされた PCI IDE インターフェースを提供します。エミュレートされた IDE ドライバーを使用して、最大 4 つの仮想 IDE ハードディスクまたは仮想 IDE CD-ROM ドライブの組み合わせを仮想マシンにアタッチすることができます。エミュレートされた IDE ドライバーは、仮想CD-ROM および DVD-R ドライブにも使用できます。

エミュレートされたフロッピーディスクドライブドライバー

エミュレートされたフロッピーディスクドライブドライバーは、仮想フロッピードライブの作成に使用されます。

エミュレートされたサウンドデバイス

エミュレートされた (Intel) HDA サウンドデバイスである **intel-hda** は以下のゲストオペレーティングシステムでサポートされます。

- Red Hat Enterprise Linux 7 (AMD64 および Intel 64 アーキテクチャー)

- Red Hat Enterprise Linux 4、5、6 (32-bit AMD および Intel アーキテクチャーと AMD64 および Intel 64 アーキテクチャー)



注記

以下のエミュレートされたサウンドドライバーも利用可能ですが、特定のゲストオペレーティングシステムとの互換性の問題で推奨されません。

- **ac97** エミュレートされた Intel 82801AA AC97 Audio 互換サウンドカード

エミュレートされたグラフィックカード

以下のエミュレートされたグラフィックカードが提供されています。

- Cirrus CLGD 5446 PCI VGA カード
- Bochs VESA 拡張を搭載している標準的な VGA グラフィックカード (すべての非標準モードを含むハードウェアレベル)

ゲストは、Simple Protocol for Independent Computing Environments (SPICE) プロトコルまたは Virtual Network Computing (VNC) システムを使用してこれらのデバイスに接続できます。

エミュレートされたネットワークデバイス

以下のように、エミュレートされたネットワークデバイスが 2 種類提供されます。

- **e1000** デバイスは、Intel E1000 ネットワークアダプター (Intel 82540EM、82573L、82544GC) をエミュレートします。
- **rt18139** デバイスは、Realtek 8139 ネットワークアダプターをエミュレートします。

エミュレートされたウォッチドッグデバイス

ウォッチドッグは、仮想マシンがオーバーロードするか、または応答しない場合に仮想マシンを自動的に再起動するために使用できます。

Red Hat Enterprise Linux 7 には、以下のエミュレートされたウォッチドッグデバイスが含まれます。

- **i6300esb** エミュレートされた Intel 6300 ESB PCI ウォッチドッグデバイス。Red Hat Enterprise Linux versions 6.0 以降のゲストオペレーティングシステムに対応しており、使用が推奨されているデバイスです。
- **ib700** エミュレートされた iBase 700 ISA ウォッチドッグデバイス。**ib700** ウォッチドッグデバイスは、Red Hat Enterprise Linux 6.2 以降を使用するゲストにのみ対応しています。

両方のウォッチドッグデバイスは、Red Hat Enterprise Linux 6.2 以降のゲストオペレーティングシステム向けの 32 ビットおよび 64 ビット AMD および Intel アーキテクチャーでサポートされます。

3.4.2. 準仮想化デバイス

準仮想化は、ゲストがホストマシン上のデバイスを使用するための高速かつ効率的な通信手段を提供します。KVM は、ハイパーバイザーとゲスト間のレイヤーとして Virtio API を使用し、準仮想化されたデバイスを仮想マシンに提供します。

準仮想化デバイスは、I/O のレイテンシーを短縮し、I/O スループットをベアメタルレベル近くにまで高めるものもあれば、その他の方法では利用できない機能を仮想マシンに追加するものもあります。I/O 集約型アプリケーションを稼働している仮想マシンには、エミュレートされたデバイスの代わりに準仮想化ドライバの使用が推奨されます。

すべての virtio デバイスは、ホストデバイスとゲストドライバの 2 つの部分から構成されます。準仮想化デバイスドライバにより、ゲストのオペレーティングシステムはホストシステム上の物理デバイスにアクセスできます。

このデバイスを使用するには、準仮想化デバイスドライバをゲストオペレーティングシステムにインストールする必要があります。デフォルトでは、Red Hat Enterprise Linux 4.7 以降、Red Hat Enterprise Linux 5.4 以降、および Red Hat Enterprise Linux 6.0 以降に準仮想化デバイスドライバが含まれています。



注記

準仮想化デバイスおよびドライバの使用に関する詳しい情報は、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

準仮想化ネットワークデバイス (virtio-net)

準仮想化ネットワークデバイスは、仮想マシンにネットワークアクセスを提供する仮想ネットワークデバイスであり、入出力パフォーマンスを強化し、レイテンシーを短縮します。

準仮想化ブロックデバイス (virtio-blk)

準仮想化ブロックデバイスは、高パフォーマンスの仮想ストレージデバイスであり、これにより、I/O パフォーマンスが強化され、レイテンシーが短縮されます。準仮想化ブロックデバイスはハイパーバイザーによってサポートされ、仮想マシンにアタッチされます (フロッピーディスクドライブは例外で、これにはエミュレートが必要になります)。

準仮想化コントローラーデバイス (virtio-scsi)

準仮想化 SCSI コントローラーデバイスは、virtio-blk に代わる、より柔軟でスケーラブルな選択肢を提供します。virtio-scsi ゲストはターゲットデバイスの機能一式を継承でき、わずか 28 デバイスしか処理できない virtio-blk に比べて数百ものデバイス処理が可能です。

virtio-scsi は以下のゲストオペレーティングシステムに完全対応しています。

- Red Hat Enterprise Linux 7
- Red Hat Enterprise Linux 6.4 以降

準仮想化クロック

タイムスタンプカウンター (TSC) をクロックソースとして使用するゲストは、時間管理の問題に直面することがあります。一定したタイムスタンプカウンターを持たないホストに関しては、KVM はゲストに準仮想化クロックを提供することで対処します。さらに、準仮想化クロックは、ゲストによるスリープ (S3) の実行または RAM へのサスペンド操作の後に必要な時間の調整を支援します。

準仮想化シリアルデバイス (virtio-serial)

準仮想化シリアルデバイスは、バイトストリーム指向の文字ストリームデバイスで、ホストのユーザー領域とゲストのユーザー領域をつなぐシンプルな通信インターフェースを提供します。

バルーンデバイス (virtio-balloon)

バルーンデバイスは仮想マシンの RAM の一部を未使用として指定することが可能で (バルーン 膨

張)、これによりメモリーが解放されて、ホスト (またはホスト上の他の仮想マシン) が使用できるようになります。仮想マシンが再度メモリーを必要とした際には、バルーンを **収縮** させて、ホストが RAM を仮想マシンに分配し直すことができます。

準仮想化乱数ジェネレーター (virtio-rng)

準仮想化乱数ジェネレーターにより、仮想マシンがホストから直接、エントロピーまたは乱数を収集して暗号化データやセキュリティに使用することができます。仮想マシンは (ハードウェアの使用状況など) 一般的なインプットを入手できないため、エントロピーが不足することが頻繁にあります。エントロピーの調達には多大な時間を要することがあるため、**virtio-rng** はホストからゲスト仮想マシンに直接エントロピーを注入することで、このプロセスを高速化します。

準仮想化グラフィックカード (QXL)

準仮想化グラフィックカードは、QXL ドライバーと連携して機能し、リモートホストから仮想マシンのグラフィックスを表示する効率的な方法を提供します。SPICE を使用するには QXL ドライバーが必要です。

3.4.3. 物理ホストデバイス

一部のハードウェアプラットフォームでは、仮想マシンが直接ハードウェアデバイスやコンポーネントにアクセスすることができます。仮想化におけるこのプロセスは、**デバイス割り当て** または **パススルー** として知られています。

VFIO デバイス割り当て

Virtual Function I/O (VFIO) は、Red Hat Enterprise Linux 7 の新しいカーネルドライバであり、これにより、仮想マシンが高パフォーマンスで物理ハードウェアにアクセスできるようになります。

VFIO は、ホストシステム上の PCI デバイスを直接仮想マシンにアタッチするため、ゲストはさまざまなタスクで PCI デバイスへの排他的アクセスが可能になります。これにより、PCI デバイスはゲスト仮想マシンに物理的にアタッチされているかのように表示され、動作します。

VFIO は、KVM ハイパーバイザーからデバイス割り当てを移動して、カーネルレベルでデバイスの分離を強制的に行うことで、これまでの PCI デバイス割り当てのアーキテクチャーを向上します。VFIO は、セキュリティが強化されており、セキュアブートとの互換性があります。VFIO は Red Hat Enterprise Linux 7 におけるデフォルトのデバイス割り当てメカニズムです。

VFIO は、Red Hat Enterprise Linux 6 の最大 8 デバイスから、Red Hat Enterprise Linux 7 の割り当てデバイス数を 32 に増やしました。VFIO は、NVIDIA GPU の割り当てもサポートします。

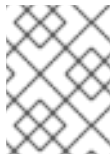


注記

VFIO デバイス割り当てに関する詳しい情報は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

USB、PCI、SCSI パススルー

KVM ハイパーバイザーは、ホストシステム上にある USB、PCI、SCSI デバイスの仮想マシンへのアタッチをサポートします。USB、PCI、SCSI デバイスを割り当てることで、これらのデバイスが仮想マシンに物理的にアタッチされているかのように表示され、動作します。このように、ゲストがさまざまなタスクで、これらのデバイスに排他的にアクセスできるようになります。

**注記**

USB、PCI、SCSI パススルーに関する詳しい情報は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

SR-IOV

SR-IOV (シングルルート I/O 仮想化) は、単一の物理 PCI 機能を拡張して PCI リソースを個別の *仮想機能* (VF) として共有する PCI Express (PCI-e) 規格です。PCI デバイスを割り当てることで、別の仮想マシンは各機能を使用できます。

SR-IOV 対応 PCI-e デバイスは、シングルルート機能 (例えば、単一イーサネットポート) を提供し、複数の個別の仮想デバイスを固有の PCI デバイス機能として提示します。仮想デバイスはそれぞれ、固有の PCI 設定領域、メモリーマップドレジスター、個別の MSI ベースの割り込みを備えている場合があります。

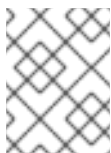
**注記**

SR-IOV に関する詳細については、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

NPIV

N_Port ID Virtualization (NPIV) とは、ファイバーチャネルデバイスの一部で利用可能な機能です。NPIV では単一の物理的な N_Port を複数の N_Port ID として共有します。SR-IOV が PCIe インターフェースに提供するファイバーチャネル Host Bus Adaptor (HBA) に対して同様の機能を提供します。NPIV を使用すると、ストレージエリアネットワーク (SAN) への仮想ファイバーチャネルイニシエーターを仮想マシンに提供できます。

NPIV は、エンタープライズレベルのストレージソリューションで高密度の仮想化環境を提供できます。

**注記**

NPIV に関する詳細は、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

3.4.4. ゲスト CPU モデル

CPU モデル は、ゲストオペレーティングシステムに公開するホスト CPU 機能を定義します。**KVM** と **libvirt** には、複数の現行プロセッサモデルの定義が含まれ、これによりユーザーは新しい CPU モデルでのみ利用できる CPU 機能を有効にすることができます。ゲストに公開できる CPU 機能のセットは、ホスト CPU、カーネル、および **qemu-kvm** コードのサポートによって異なります。

異なる CPU 機能のセットを持つ複数のホスト間で仮想マシンを安全にマイグレーションできるように、デフォルトでは **KVM** により、ホスト CPU の全機能がゲストオペレーティングシステムに公開されず、その代わりに、CPU 機能は選択された CPU モデルに基づいて公開されます。仮想マシンで所定の CPU 機能が有効にされている場合には、この機能をゲストへ公開するようにサポートしていないホストに移行することはできません。



注記

ゲスト CPU モデルに関する詳しい情報は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

3.5. ストレージ

仮想マシン向けのストレージは、仮想マシンに割り当てられた物理ストレージから抽出されます。これは、準仮想化またはエミュレートされたブロックデバイスドライバーを使用して仮想マシンにアタッチされます。

3.5.1. ストレージプール

ストレージプールとは、仮想マシンにストレージを提供するために **libvirt** が管理するファイルやディレクトリー、ストレージデバイスのことです。ストレージプールは、仮想マシンイメージを保存するストレージ ボリューム または仮想マシンに新たなストレージとしてアタッチされるストレージボリュームに分割されます。複数のゲストが同一のストレージプールを共有でき、ストレージリソースをより適切に割り当てることができます。詳細は『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

ローカルストレージプール

ローカルストレージプールは、ホストサーバーに直接アタッチされています。これに含まれるのは、ローカルディレクトリー、直接アタッチされたディスク、物理パーティション、ローカルデバイス上の論理ボリューム管理 (LVM) のボリュームグループです。ローカルストレージプールは開発、テスト、移行や多数の仮想マシンを必要としない小規模開発などに役立ちます。ローカルストレージプールは、ライブマイグレーションに対応していないため、多くの本番環境に適切でない可能性があります。

ネットワーク (共有) ストレージプール

ネットワークストレージプールには、標準プロトコル使用のネットワークにまたがって共有されるストレージデバイスが含まれます。ネットワークストレージは、**virt-manager** を使用したホスト間の仮想マシン移行が必要となりますが、**virsh** を使用した移行の場合はオプションとなります。ネットワークストレージプールは、**libvirt** で管理します。

3.5.2. ストレージボリューム

ストレージプールは、さらにストレージボリュームに分割されます。ストレージボリュームとは、物理パーティション、LVM 論理ボリューム、ファイルベースのディスクイメージ、**libvirt** で処理される他のストレージタイプなどを抽象化したものです。ストレージボリュームは、基盤となるハードウェアに関係なく、ローカルストレージデバイスとして仮想マシンに提示されます。

3.5.3. エミュレートされたストレージデバイス

仮想マシンには、ホストによりエミュレートされるさまざまなストレージデバイスが提供されます。ストレージデバイスのタイプはそれぞれ、特定のユースケースに適しています。これにより、柔軟性やゲストのオペレーティングシステムとの互換性を最大化することができます。

virtio-scsi

virtio-scsi は数多くのディスクや TRIM などの高度なストレージ機能を使用するゲスト用の準仮想化デバイスとして推奨されます。Red Hat Enterprise Linux 7 以外のオペレーティングシステムを使用しているゲストでは、ゲストドライバーのインストールが必要になる場合があります。

virtio-blk

virtio-blk はイメージファイルをゲストに公開するのに適した準仮想化ストレージデバイスです。**virtio-blk** は仮想マシンに対して最高レベルのディスクの I/O パフォーマンスを提供しますが、**virtio-scsi** と比較すると機能の数が少なくなります。

IDE

IDE は virtio ドライバーに対応しないレガシーゲスト用に推奨されます。IDE のパフォーマンスは **virtio-scsi** または **virtio-blk** よりも低いですが、複数の異なるシステムに対する互換性の面では優れています。

CD-ROM

ATAPI CD-ROM および **virtio-scsi** CD-ROM が提供されており、ゲストが ISO ファイルまたはホスト CD-ROM ドライブを使用できるようにします。**virtio-scsi** CD-ROM は、**virtio-scsi** ドライバーがインストールされているゲストと共に使用することができます。ATAPI CD-ROM の方が互換性の幅が広いですが、パフォーマンスはより低くなります。

USB 大容量ストレージデバイスおよびフロッピーディスク

エミュレートされた USB 大容量ストレージデバイスおよびフロッピーディスクは、リムーバブルメディアが必要な場合に利用することができます。フロッピーディスクよりも容量がより大きい USB 大容量ストレージデバイスの使用をお勧めします。

3.5.4. ホストストレージ

ディスクイメージは、ホストに接続される様々なローカルおよびリモートのストレージテクノロジーの上に格納できます。

画像ファイル

イメージファイルはホストファイルシステム上にしか格納できません。イメージファイルは、**ext4** または **xfs** などのローカルファイルシステム上や、**NFS** などのネットワークファイルシステム上に格納することができます。

libguestfs などのツールを使用して、ファイルの管理、バックアップおよび監視を行うことができます。KVM 上で使用されるディスクイメージの形式には以下が含まれます。

raw

Raw イメージファイルには、ディスクのコンテンツが含まれますが、メタデータは一切追加されません。

ホストファイルシステムが許可する場合、Raw ファイルのタイプは **Preallocated** (事前割り当て) ファイルまたは **Sparse** (スパース) ファイルのいずれかになります。**Sparse** ファイルはホストディスク容量をオンデマンドで割り当てるため、一種のシンプロビジョニングとすることができます。**Preallocated** ファイルは完全にプロビジョニングされますが、**Sparse** ファイルよりもパフォーマンスが高くなります。

Raw ファイルは、ディスクの I/O パフォーマンスが重要となる場合や、イメージファイルをネットワーク上で転送する必要がほとんどない場合に適しています。

qcow2

qcow2 イメージファイルは、バックアップファイルや、スナップショット、圧縮および暗号化などの数多くの高度なディスクイメージ機能を提供します。これらのイメージファイルは、テンプレートイメージから仮想マシンのインスタンスを作成するために使用することができます。

qcow2 ファイルは、通常ネットワーク上でより効率的に転送できるファイルです。仮想マシンによって書き込まれるセクターのみがイメージに割り当てられるためです。

Red Hat Enterprise Linux 7.0 は、qcow2 バージョン 3 のイメージファイル形式をサポートします。

LVM ボリューム

論理ボリューム (LV) はディスクイメージに使用でき、システムの LVM ツールを使用して管理できます。LVM のブロックストレージのモデルは他のファイルシステムよりもシンプルであるため、LVM はより高いパフォーマンスを提供します。

LVM シンプロビジョニングはスナップショットを提供し、LVM ボリュームの領域のより効率的な使用を可能にします。さらに、qcow2 に対する移行の代替手段として使用することができます。

ホストデバイス

物理 CD-ROM および Raw ディスク、または論理ユニット番号 (LUN) などのホストデバイスをゲストに提示することができます。これにより、ゲストはパフォーマンスを維持した状態で、SAN や iSCSI LUN、およびローカル CD-ROM メディアを使用できます。

ホストデバイスは、ストレージ管理がホスト上ではなく SAN 上で行なわれている場合に使用することができます。

分散ストレージシステム

Gluster ボリュームはディスクイメージとして使用できます。これにより、ネットワーク上で高パフォーマンスのクラスター化ストレージを使用することができます。

Red Hat Enterprise Linux 7 には、GlusterFS 上のディスクに対するネイティブサポートが含まれます。これにより、KVM ホストは GlusterFS ボリュームから仮想マシンイメージを起動し、仮想マシンのデータディスクとして GlusterFS ボリュームのイメージを使用することができます。GlusterFS FUSE と比較すると、KVM のネイティブサポートのほうが高いパフォーマンスを実現します。



注記

ストレージおよび仮想化に関する詳細は、[『Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド』](#)を参照してください。

3.6. 仮想ネットワーク

仮想ゲストをネットワークに接続する場合、物理ホストのソフトウェアネットワークコンポーネントが使用されます。これらのソフトウェアコンポーネントは、**libvirt** の仮想ネットワーク設定を使用して再配置および再設定することができます。したがって、ホストは [仮想ネットワークのスイッチ](#) として機能し、ゲストのネットワーク設定ニーズに応じてさまざまな設定を行うことができます。

デフォルトでは、1 つのホスト上のゲストは、すべて **default** という名前の同じ libvirt 仮想ネットワークに接続されます。このネットワーク上のゲストは、以下の接続が可能です。

相互の接続および仮想化ホストとの接続

受信および送信トラフィックの両方が可能です。ただし、ゲストオペレーティングシステムのネットワークスタックのファイアウォール、およびゲストインターフェースに適用される [libvirt ネットワークフィルター](#) のルールの影響を受けます。

仮想化ホストの外部にあるネットワーク上の他のホストとの接続

送信トラフィックのみが可能です。ホストシステムのファイアウォールと共に、[Network Address Translation \(NAT\)](#) のルールの影響を受けます。

ただし、必要に応じて、ゲストインターフェースを以下のモードのいずれかに設定することができます。

隔離モード

ゲストは、仮想化ホストの外部にあるホストとのあらゆる通信を許可しないネットワークに接続されます。

ルーティングモード

ゲストは、NAT を実施せずにトラフィックをゲストと外部ホスト間でルーティングするネットワークに接続されます。このモードでは受信接続は可能ですが、外部ネットワーク上のシステムに対して追加のルーティングテーブルエントリが必要になります。

ブリッジモード

ゲストはブリッジデバイスに接続されます。このブリッジデバイスも、ローカルイーサネットに接続された物理イーサネットデバイスに直接接続されています。このモードではゲストは物理ネットワーク上に直接表示されます。したがって、受信接続は可能ですが追加のルーティングテーブルエントリは必要ありません。

仮想マシンからの標準的な送信のみのネットワークアクセスでは、追加のネットワーク設定は必要ありません。libvirt パッケージと共に **デフォルトの** ネットワークがインストールされ、**libvirtd** サービスが起動されると自動的に開始されます。より高度な機能が必要な場合は、[virsh](#) または [virt-manager](#) のどちらかを使用して追加のネットワークを作成して設定することができます。そして、[ゲスト XML 設定ファイル](#) を編集してこれらの新たなネットワークの 1 つを使用することができます。



注記

仮想ネットワークの高度な設定については、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

ゲストオペレーティングシステムの観点からは、仮想ネットワーク接続と通常の物理ネットワーク接続の違いはありません。Red Hat Enterprise Linux 7 ゲストでのネットワーク設定の詳細については、『[Red Hat Enterprise Linux 7 ネットワークガイド](#)』を参照してください。

第4章 仮想化コマンドラインインターフェースの使用

Red Hat Enterprise Linux 7 では仮想化を使用する場合、コマンドラインユーザインターフェース (CLI) を使用するのが標準の方法です。CLI コマンドを入力すると、ホストシステム上で仮想マシンを作成したり、仮想マシンとの対話を行うシステムユーティリティーがアクティブになります。この方法では、**virt-manager** などのグラフィカルなアプリケーションを使用する場合よりもより詳細に制御を行うことができ、スクリプト化や自動化の可能性が広がります。

4.1. 仮想化の主なコマンドラインユーティリティー

以下のサブセクションでは、Red Hat Enterprise Linux 7 で仮想化を設定および管理するために使用できる主なコマンドラインユーティリティーを取り上げます。これらのコマンドとその他の仮想化ユーティリティーの多くは、Red Hat Enterprise Linux リポジトリによって提供されるパッケージに含まれており、[Yum パッケージマネージャー](#)を使用してインストールできます。

仮想化パッケージのインストールに関する詳細は、『[仮想化の導入および管理ガイド](#)』を参照してください。

4.1.1. virsh

virsh は、ハイパーバイザとゲスト仮想マシンを管理するための CLI ユーティリティーです。主にこのユーティリティーを使用して、Red Hat Enterprise Linux 7 上で仮想化を制御します。これには以下の機能が含まれています。

- 仮想マシンを作成、設定、一時停止、表示、およびシャットダウンします。
- 仮想ネットワークの管理
- 仮想マシンの [ディスクイメージ](#)をロードします。

virsh ユーティリティーは仮想化管理スクリプトの作成に適しています。root 権限を持たないユーザーは読み取り専用で **virsh** を使用できます。

virsh の使用

virsh ユーティリティーは標準のコマンドライン入力で使用できますが、インタラクティブシェルとして使用することもできます。シェルモードでは、**virsh** コマンドの接頭辞は必要なく、ユーザーは常に root として登録されます。以下の例は **virsh hostname** コマンドを使用してハイパーバイザーのホスト名を表示します。最初の例は標準モードで、2 つ目は対話モードの例になります。

```
$ virsh hostname
localhost.localdomain

$ virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh # hostname
localhost.localdomain
```



重要

root 権限を持たないユーザーとして **virsh** を使用する場合は、特権のない **libvirt セッション** を使用します。

要素への読み取りアクセスを取得するには **virsh** コマンドで **-c qemu:///system** オプションを使用します。

virsh でのヘルプ

Linux のすべての bash コマンドと同様、**virsh** のヘルプを表示するには、**man virsh** コマンドまたは **--help** オプションを使用します。また、**virsh help** コマンドを使用すると、**virsh** コマンドに固有するヘルプテキストを表示でき、キーワードを使用すると特定のグループに属する **virsh** コマンドをすべてリストできます。

virsh コマンドグループとそれらのキーワードは次のとおりです。

- ゲストの管理: キーワード **domain**
- ゲストの監視: キーワード **monitor**
- ホストおよびハイパーバイザーの監視および管理: キーワード **host**
- ホストシステムのネットワークインターフェースの管理: キーワード **interface**
- 仮想ネットワークの管理: キーワード **network**
- ネットワークフィルターの管理: キーワード **filter**
- ノードデバイスの管理: キーワード **nodedev**
- パスフレーズや暗号化キーなどの秘密の管理: キーワード **secret**
- スナップショットの管理: キーワード **snapshot**
- ストレージプールの管理: キーワード **pool**
- ストレージボリュームの管理: キーワード **volume**
- 一般的な **virsh** の使用方法: キーワード **virsh**

以下の例は、ゲスト仮想マシンの名前を変更する方法を示しています。**virsh help** を使用して最初に適切なコマンドを探し、その構文を調べます。そして、そのコマンドを使用して *Fontaine* というゲストの名前を *Atlas* に変更します。

例4.1 キーワードを使用してすべてのコマンドの help を一覧表示する方法

```
# virsh help domain
Domain Management (help keyword 'domain'):
  attach-device          attach device from an XML file
  attach-disk            attach disk device
  [...]
  domname                convert a domain id or UUID to
domain name
  domrename              rename a domain
  [...]
```

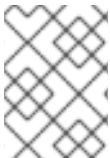
```
# virsh help domrename
NAME
  domrename - rename a domain

SYNOPSIS
  domrename <domain> <new-name>

DESCRIPTION
  Rename an inactive domain.

OPTIONS
  [--domain] <string>  domain name, id or uuid
  [--new-name] <string> new domain name

# virsh domrename --domain Fontaine --new-name Atlas
Domain successfully renamed
```



注記

virsh を使用した仮想マシンの管理に関する詳細は、『[『Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド』](#)』を参照してください。

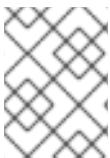
4.1.2. virt-install

virt-install は、新規仮想マシンを作成する CLI ユーティリティです。このユーティリティは、シリアルコンソール、SPICE、または VNC クライアント/サーバーペアグラフィックを使用して、テキストベースのインストールとグラフィカルインストールの両方をサポートします。インストールメディアはローカルに配置でき、NFS、HTTP、または FTP サーバー上でリモートに配置することもできます。このツールを使用すると、無人で稼働し、キックスタートを使用してゲストを準備するよう設定できるため、インストールの自動化が容易になります。このツールは **virt-install** パッケージに含まれています。



重要

root 権限のないユーザーとして **virt-install** を使用する場合は、特権のない **libvirt セッション** を使用します。この場合、作成したゲストは作成者のみに表示され、root 権限を持つユーザーが作成したゲストが持つ機能の一部を使用できません。



注記

virt-install の使用に関する詳細は、『[『Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド』](#)』を参照してください。

4.1.3. virt-xml

virt-xml は、ドメイン XML ファイルを編集するためのコマンドラインユーティリティです。XML の設定を正常に変更するには、ゲストの名前、XML のアクション、および変更内容がコマンドに含まれている必要があります。

たとえば、以下はゲスト起動設定に関連するサブオプションを表示し、**example_domain** ゲストの起動メニューをオンにします。

■

```
# virt-xml boot=?
--boot options:
  arch
  cdrom
  [...]
  menu
  network
  nvram
  nvram_template
  os_type
  smbios_mode
  uefi
  useserial
# virt-xml example_domain --edit --boot menu=on
Domain 'example_domain' defined successfully.
```

コマンドの各呼び出しは、ドメイン XML ファイルごとに 1 つのアクションを実行できることに注意してください。



注記

このツールは `virt-install` パッケージに含まれています。**virt-xml** の使用に関する詳細は **virt-xml** の man ページを参照してください。

4.1.4. guestfish

guestfish は仮想マシンのディスクイメージを検証および編集するためのコマンドラインユーティリティです。libguestfs ライブラリーを使用し、**libguestfs** API によって提供される機能をすべて公開します。

guestfish の使用

guestfish ユーティリティは標準のコマンドライン入力で使用できますが、インタラクティブシェルとして使用することもできます。シェルモードでは、**guestfish** コマンドの接頭辞は必要なく、ユーザーは常に root として登録されます。以下の例は **guestfish** コマンドを使用して *testguest* 仮想マシン上のファイルシステムを表示します。最初の例は標準モードで、2 つ目は対話モードの例になります。

```
# guestfish domain testguest : run : list-fileSYSTEMS
/dev/sda1: xfs
/dev/rhel/root: xfs
/dev/rhel/swap: swap
# guestfish
```

```
Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.
```

```
Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell
```

```
><fs> domain testguest
><fs> run
><fs> list-fileSYSTEMS
```

```
/dev/sda1: xfs
/dev/rhel/root: xfs
/dev/rhel/swap: swap
```

また、自動化の目的で **guestfish** を **bash スクリプト** で使用することもできます。

重要

guestfish を root 権限を持たないユーザーとして使用する場合、特権のない **libvirt セッション** を使用します。この場合、root 権限を持つユーザーによって作成されたゲストのディスクイメージを見ることはできず、対話することもできません。

これらのディスクイメージに読み取り専用アクセスできるようにするには、**guestfish** に **-ro -c qemu:///system** オプションを使用します。また、ディスクイメージファイルの読み取り権限が必要です。

guestfish のヘルプ

すべての bash コマンドと同様、**guestfish** のヘルプを表示するには、**man guestfish** コマンドまたは **--help** オプションを使用します。また、**guestfish help** コマンドを使用すると、**guestfish** コマンドに固有する詳細情報を表示できます。以下の例は、**guestfish add** コマンドに関する情報を表示します。

```
$ guestfish help add
NAME
    add-drive - add an image to examine or modify

SYNOPSIS
    add-drive filename [readonly:true|false] [format:...] [iface:...]
    [name:...] [label:...] [protocol:...] [server:...] [username:...] [secret:...]
    [cachemode:...] [discard:...] [copyonread:true|false]

DESCRIPTION
    This function adds a disk image called filename to the handle.
    filename
    may be a regular host file or a host device.
    [...]
```

注記

guestfish の使用に関する詳細は、『[Red Hat Enterprise Linux 7 仮想化の導入および管理ガイド](#)』を参照してください。

4.2. デモンストラーション: コマンドラインユーティリティーによるゲストの作成および管理

ここでは、CLI で仮想化タスクを実行する方法を実証するために、新しいゲスト仮想マシンを作成し、そのマシンに OS をインストールした後にゲストと通信し、CLI コマンドを使用して仮想マシンを管理する例を説明します。

4.2.1. インストール

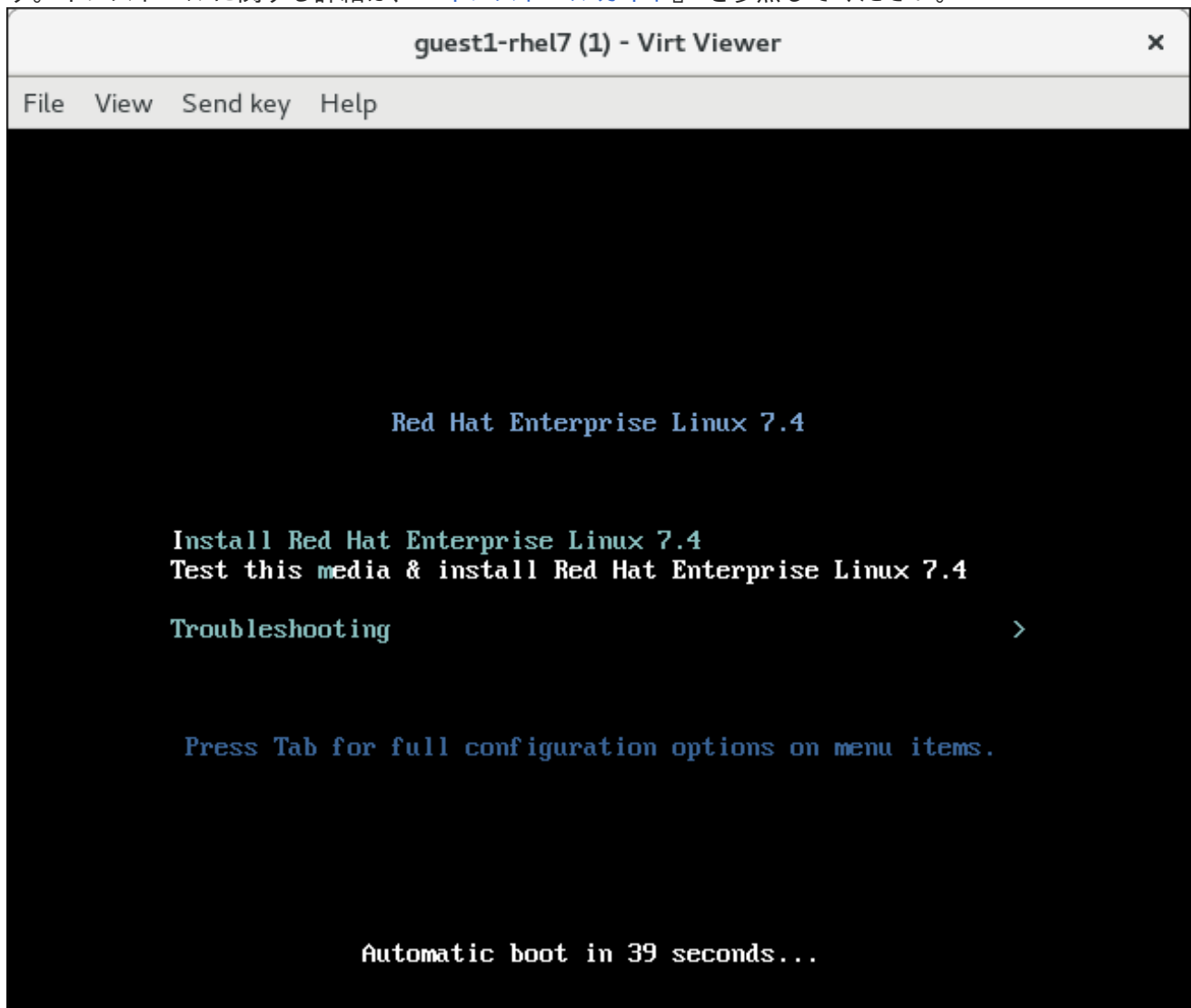
以下の例では、**guest1-rhel7** という名前の新しいゲストを作成し、Red Hat Enterprise Linux 7 ワー

クステーションの ISO イメージから OS のインストールを開始します。このイメージは、[カスタマーポータル](#) から取得でき、この例では `~/Downloads/` フォルダーにあります。ゲストには 2 つの仮想 CPU、2048 MB の RAM、および 8 GB のディスク領域が割り当てられています。

```
# virt-install --name guest1-rhel7 --memory 2048 --vcpus 2 --disk size=8 -  
-cdrom /home/username/Downloads/rhel-workstation-7.4-x86_64-dvd.iso --os-  
variant rhel7
```

```
Starting install...  
Allocating 'guest1-rhel7.qcow2'  
| 8.0 GB  00:00:00
```

これにより、**virt-viewer** アプリケーションのグラフィカル Anaconda インストーラーが起動されます。インストールに関する詳細は、『[インストールガイド](#)』を参照してください。





注記

グラフィカルインターフェースにアクセスできないホストシステムでは、以下のような **virt-install** コマンドを実行し、[テキストベースの Anaconda](#) を使用してゲスト OS をインストールすることができます。

```
# virt-install -name rhel7anaconda-guest -r 1024 --
location=/home/jherrman/Downloads/rhel-workstation-7.4-x86_64-
dvd.iso --disk size=8 --nographics --extra-args="console=tty0
console=ttyS0,115200n8"
```

インストールが正常に終了すると、コマンドラインに以下が表示されます。

```
Domain creation completed.
Restarting guest.
```

これで、ゲストの設定が可能になりましたが、ゲストの設定を安全に管理するため、最初にゲストをシャットダウンすることが推奨されます。

```
# virsh shutdown guest1-rhel7
Domain guest1-rhel7 is being shutdown
```

4.2.2. デバイスの接続

ゲストがホストに接続する USB デバイスを検出し、使用するようするには (この例では Samsung の携帯電話)、最初にホストで **lsusb** コマンドを実行し、デバイスの ID を取得します。

```
# lsusb

[...]
Bus 003 Device 007: ID 04e8:6860 Samsung Electronics Co., Ltd Galaxy (MTP)
```

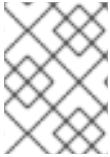
その後、好きなテキストエディターをホストで使用して、デバイス (この例では **samsung_USB_device.xml**) の XML ファイルを作成します。このファイルにベンダーおよび製品 ID を入力します。

```
# vim samsung_USB_device.xml

<hostdev mode='subsystem' type='usb' managed='yes'>
  <source>
    <vendor id='0x04e8' />
    <product id='0x6860' />
  </source>
</hostdev>
```

最後に **virsh attach-device** コマンドを使用してデバイスをゲストに接続します。

```
# virsh attach-device guest1-rhel7 --file samsung_USB_device.xml --config
Device attached successfully
```



注記

稼働中のゲストにデバイスを接続することも可能です。それには、**--live** オプションを使用します。

4.2.3. ゲストとの対話

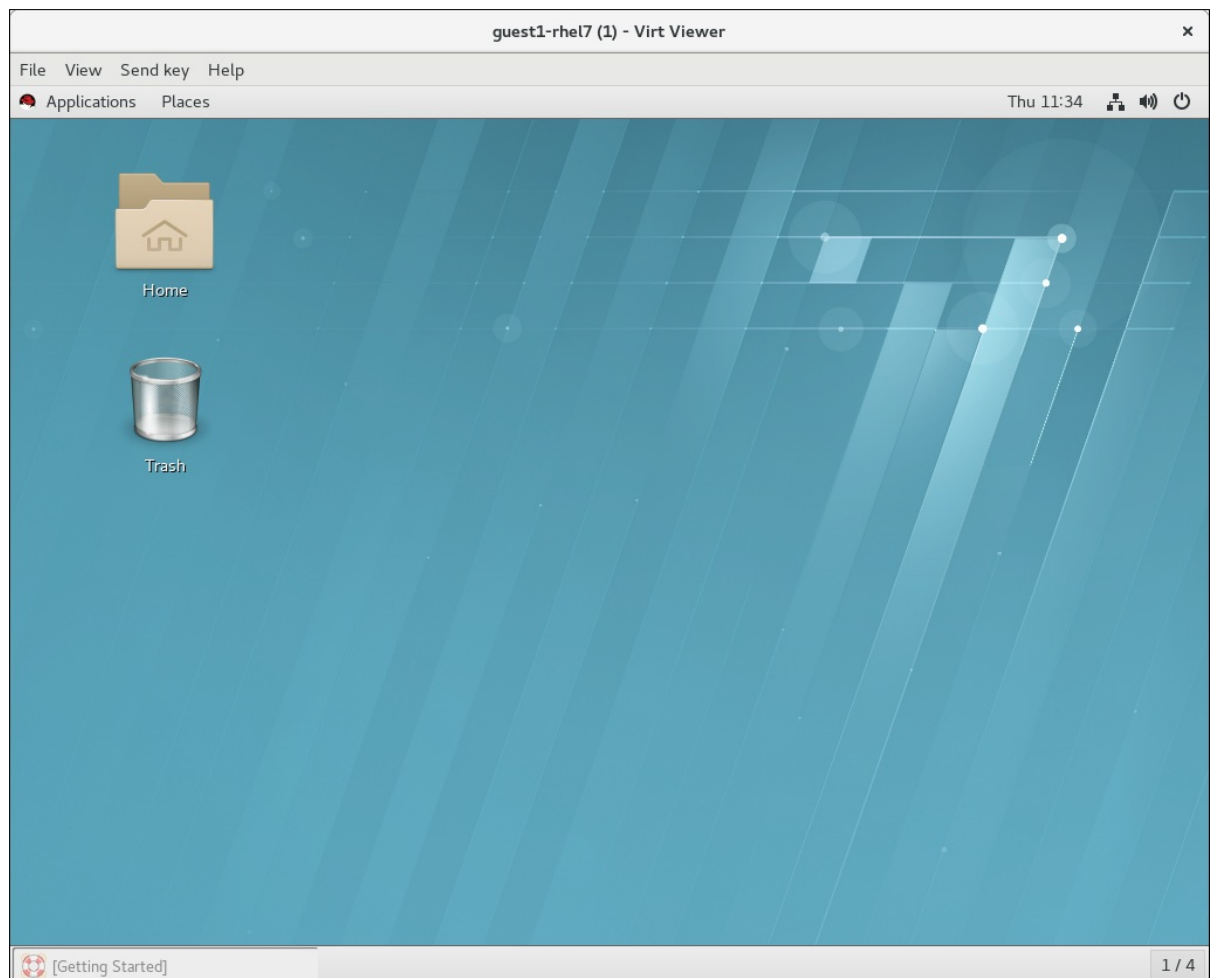
guest1-rhel7 ゲストを起動してから使用します。

```
# virsh start guest1-rhel7
Domain guest1-rhel7 started
```

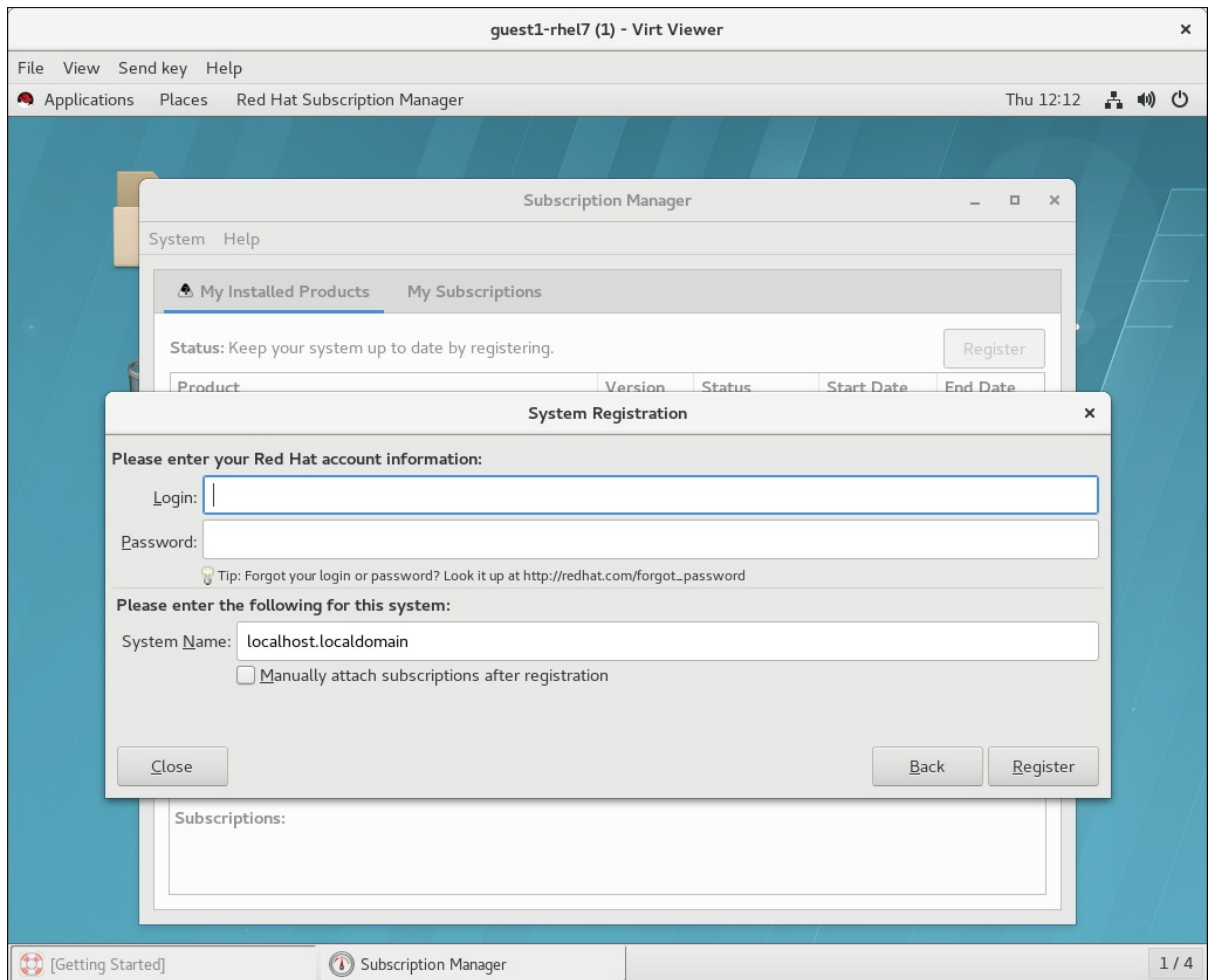
ホストシステムでグラフィック表示を使用できるかどうかに応じて、**virt-viewer** アプリケーションまたは SSH シェルのいずれかを使用してゲストと対話できます。

- グラフィック表示を使用できるシステムでは、**virt-viewer** を使用します。

```
# virt-viewer guest1-rhel7
```



その後、物理マシン上で OS の GUI を使用するように、**virt-viewer** ウィンドウの画面出力と対話できます。たとえば、[サブスクリプションマネージャー](#) アプリケーションを使用して Red Hat Enterprise Linux のゲスト OS を登録できます。



- ホストまたはゲストでテキスト専用インターフェースを使用できる場合は、SSH を使用します。これには、ゲストの IP アドレスを知っている必要があります。IP アドレスが分からない場合は、**virsh domifaddr** コマンドを使用して取得することができます。

```
# virsh domifaddr guest1-rhel7
Name          MAC address          Protocol    Address
-----
vnet0         52:54:00:65:29:21         ipv4        10.34.3.125/24
# ssh root@10.34.3.125
root@10.34.3.125's password:
Last login: Wed Jul 19 18:27:10 2017 from 192.168.122.1
[root@localhost ~]#
```



注記

virsh domifaddr が動作するには、ゲストが動作し、ネットワーク上でアクセスする必要があります。また、アクティブ化に [QEMU ゲストエージェント](#) が必要になることがあります。

その後、ゲストマシンでターミナルを使用するのと同様に、ホストターミナルと対話することができます。たとえば、[subscription-manager](#) ユーティリティを使用して、Red Hat Enterprise Linux のゲスト OS を使用することができます。

```
[root@localhost ~]# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
```

```

Username: username@sample.com
Password:
The system has been registered with ID: 30b5e666-67f9-53bb-4b90-
c2a88e5be789

```

4.2.4. 診断

ゲストの状態に関する一般情報を表示します。

```

# virsh dominfo guest1-rhel7
Id: 1
Name: guest1-rhel7
UUID: ec0c0122-fb63-4a54-b602-5cf84f5e2dfd
OS Type: hvm
State: running
CPU(s): 2
CPU time: 33.4s
Max memory: 2097152 KiB
Used memory: 2097152 KiB
Persistent: yes
Autostart: disable
Managed save: no
Security model: selinux
Security DOI: 0
Security label: unconfined_u:unconfined_r:svirt_t:s0:c102,c792 (enforcing)

```

4.2.5. スナップショットの作成

ゲストの状態をバックアップするには、**virsh snapshot-create** コマンドを使用することができます。

```

# virsh snapshot-create guest1-rhel7
Domain snapshot 1500563241 created

```

現在のスナップショットとそれぞれの XML 設定を表示することができます。

```

# virsh snapshot-list guest1-rhel7
Name                               Creation Time                       State
-----
1500563241                         2017-07-20 17:07:21 +0200         shutoff

# virsh snapshot-dumpxml guest1-rhel7 1500563241
<domainsnapshot>
  <name>1500563241</name>
  <state>shutoff</state>
  <creationTime>1500563241</creationTime>
  <memory snapshot='no' />
  <disks>
    <disk name='vda' snapshot='internal' />
  </disks>
</domainsnapshot>
[...]
```

このスナップショットを後でロードすると、ゲストをこのスナップショットが保存された状態に戻すことができます。

■

```
# virsh snapshot-revert guest1-rhel7 --snapshotname 150056324
```

第5章 仮想マシンマネージャーの使用

仮想マシンマネージャーはゲスト仮想マシンを作成および管理するグラフィカルツールで、**virt-manager** と呼ばれます。本章では、**仮想マシンマネージャー** の概要と実行方法について説明します。



注記

グラフィカルインターフェースを備えたシステム上でのみ仮想マシンマネージャーを実行できます。

仮想マシンマネージャーの使用に関する詳細は、その他の [Red Hat Enterprise Linux 仮想化ガイド](#) を参照してください。

5.1. 仮想マシンマネージャーの実行

仮想マシンマネージャー実行するには、アプリケーションのリストで仮想マシンマネージャーを選択するか、次のコマンドを実行します。

```
# virt-manager
```

仮想マシンマネージャーのメインウィンドウを開きます。

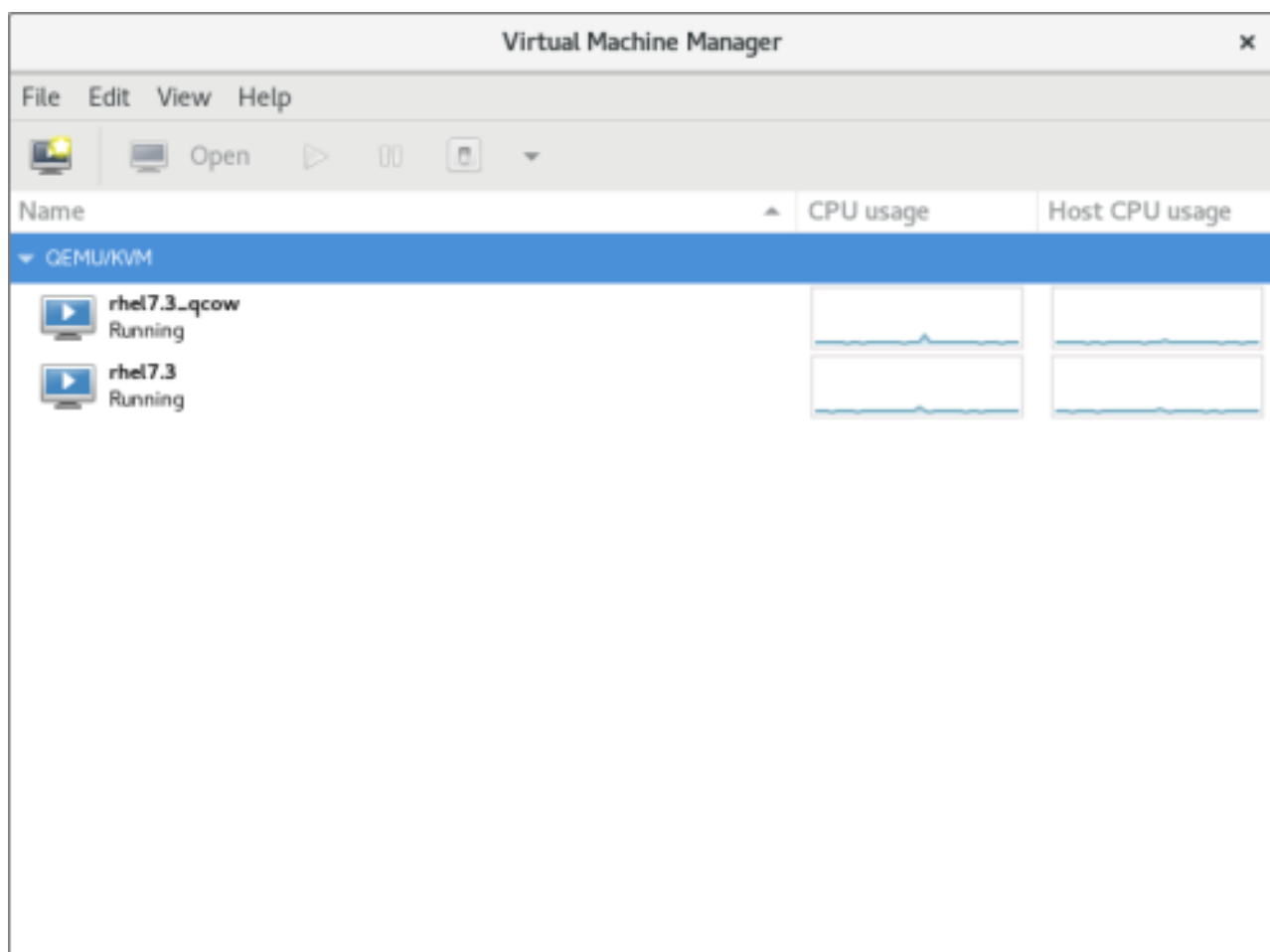


図5.1 仮想マシンマネージャー



注記

virt-manager の実行に失敗した場合は、virt-manager パッケージがインストールされていることを確認してください。virt-manager パッケージのインストールに関する詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「[仮想化パッケージのインストール](#)」を参照してください。

5.2. 仮想化マシンマネージャーのインターフェース

ここでは、仮想マシンマネージャーのユーザーインターフェースについて説明します。ユーザーインターフェースには [仮想マシンマネージャーのメインウィンドウ](#) や [仮想マシンウィンドウ](#) が含まれます。

5.2.1. 仮想マシンマネージャーのメインウィンドウ

次の図は、仮想マシンマネージャーのメインウィンドウインターフェースを示しています。

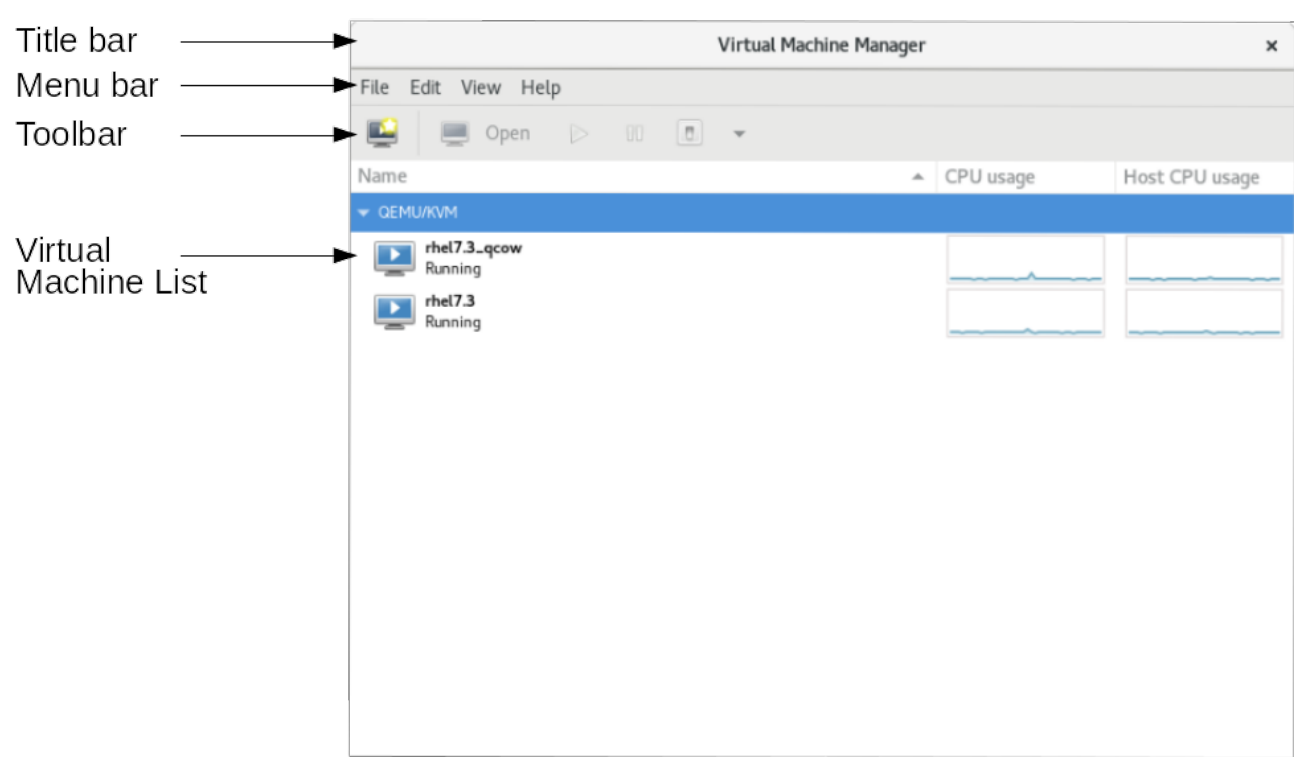


図5.2 仮想マシンマネージャーのウィンドウ

仮想マシンマネージャーのメインウィンドウのタイトルバーには **仮想マシンマネージャー** が表示されます。

5.2.1.1. メインウィンドウのメニューバー

以下の表に、仮想マシンマネージャーのメインウィンドウメニューのエントリーを示します。

表5.1 仮想マシンマネージャーのメインウィンドウメニュー

メニュー名	メニュー項目	説明
-------	--------	----

メニュー名	メニュー項目	説明
ファイル	接続を追加	「接続を追加」ダイアログを開き、ローカルまたはリモートハイパーバイザーに接続します。詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「 リモート接続の追加 」を参照してください。
	新しい仮想マシン	「新しい仮想マシン」ウィザードを開き、新しいゲスト仮想マシンを作成します。詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「 virt-manager を使用したゲストの作成 」を参照してください。
	閉じる	仮想マシンウィンドウを閉じずに仮想マシンマネージャーを閉じます。稼働中の仮想マシンは停止されません。
	終了	仮想マシンマネージャーとすべての仮想マシンウィンドウを閉じます。稼働中の仮想マシンは停止されません。
編集	接続の詳細	選択した接続の「接続の詳細」ウィンドウを開きます。
	仮想マシンの詳細	選択した仮想マシンの「仮想マシン」ウィンドウを開きます。詳細は「 仮想マシンペイン 」を参照してください。
	削除	選択した接続または仮想マシンを削除します。
	設定	仮想マシンマネージャーのオプションを設定する「設定」ダイアログボックスを開きます。
表示	グラフ <ul style="list-style-type: none"> 仮想マシン CPU 使用率 ホスト CPU 使用率 メモリーの使用率 ディスク I/O ネットワーク I/O 	仮想マシンマネージャーのメインウィンドウで仮想マシンのパラメーターの表示を調整します。
ヘルプ	このアプリケーションについて	仮想マシンマネージャーに関する情報をウィンドウに表示します。

5.2.1.2. メインウィンドウのツールバー

以下の表は、仮想マシンマネージャーのメインウィンドウにあるアイコンを示しています。

表5.2 仮想マシンマネージャーのメインウィンドウのツールバー

アイコン	説明
	「新しい仮想マシン」ウィザードを開いて、新しいゲスト仮想マシンを作成します。
	選択した仮想マシンの「仮想マシン」ウィンドウを開きます。
	選択した仮想マシンを起動します。
	選択した仮想マシンを一時停止します。
	選択した仮想マシンを停止します。
	<p>メニューを開き、選択した仮想マシンで実行する以下のアクションの1つを選択します。</p> <ul style="list-style-type: none"> ● 再起動 - 選択した仮想マシンを再起動します。 ● シャットダウン - 選択した仮想マシンをシャットダウンします。 ● 強制的にリセット - 選択した仮想マシンを強制的にシャットダウンおよび再起動します。 ● 強制的に電源OFF - 選択した仮想マシンを強制的にシャットダウンします。 ● 保存 - 選択した仮想ファイルの状態をファイルに保存します。詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「ゲスト仮想マシンの設定の保存」を参照してください。

5.2.1.3. 仮想マシンリスト

仮想マシンリストには、仮想マシンマネージャーが接続されている仮想マシンのリストが表示されます。リスト内の仮想マシンは、接続ごとにグループ化されます。テーブル列のヘッダーをクリックすると、リストを並べ替えることができます。







Name	CPU usage	Host CPU usage
▼ QEMU/KVM		
 rhel7.3 Running		
 rhel7.3_qcow Running		

図5.3 仮想マシンリスト

仮想マシンリストには、各仮想マシンで使用されているリソースに関する情報を示すグラフが表示されます。**編集** メニューにある **設定** ダイアログの **ポーリング** タブからリソースを表示できるようにします。仮想マシンリストに表示できるリソースは次のとおりです。

- CPU の使用率
- ホスト CPU の使用率
- メモリーの使用率
- ディスク I/O
- ネットワーク I/O

表示 メニューの **グラフ** メニュー項目を使用すると、表示するリソースを選択できます。

5.2.2. 仮想マシンウインドウ

ここでは、仮想マシンウインドウのインターフェースについて説明します。

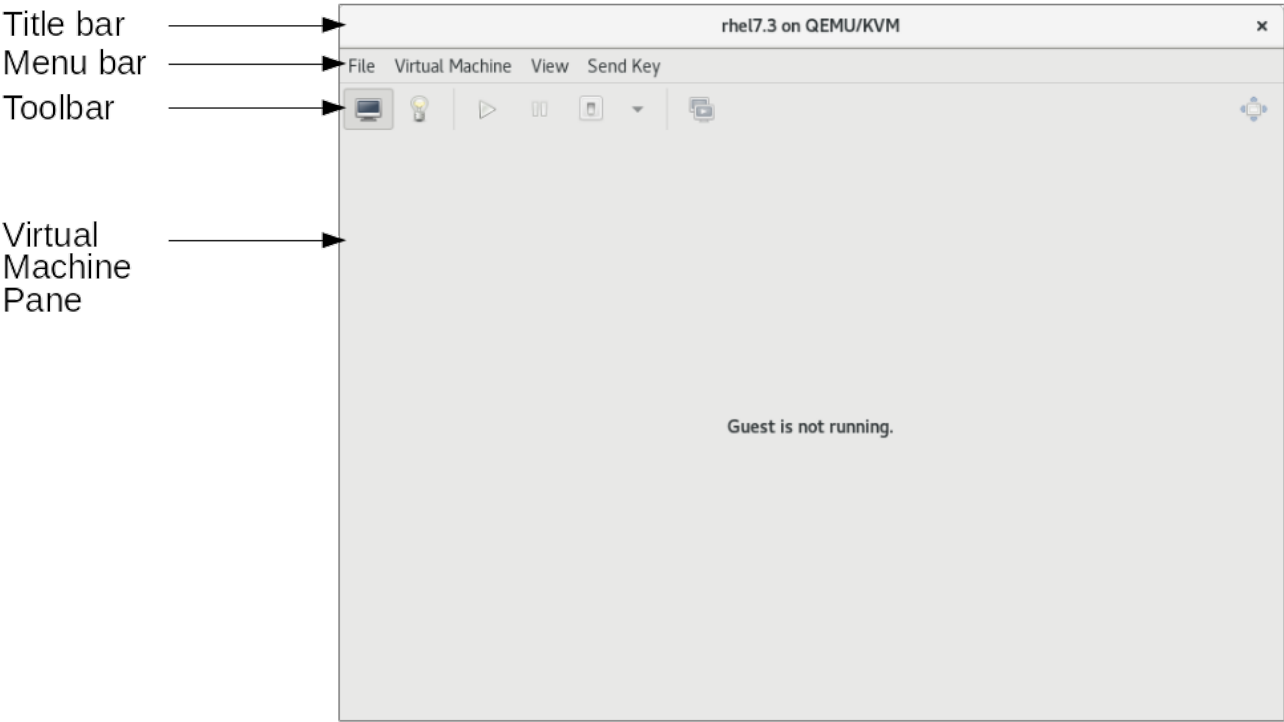


図5.4 仮想マシンウインドウ

タイトルバーには、仮想マシンの名前と仮想マシンが使用する接続が表示されます。

5.2.2.1. 仮想マシンウインドウのメニューバー

仮想マシンウインドウメニューのエントリーは次のとおりです。

表5.3 仮想マシンウインドウのメニュー

メニュー名	メニュー項目	説明
-------	--------	----

メニュー名	メニュー項目	説明
ファイル	仮想マシンマネージャーを表示	メインの仮想マシンマネージャーのウィンドウを開きます。
	閉じる	仮想マシンを停止せずに仮想マシンウィンドウのみを閉じます。
	終了	すべての仮想マシンマネージャーウィンドウを閉じます。実行中の仮想マシンは停止されません。
仮想マシン	実行	仮想マシンを実行します。このオプションは、仮想マシンが実行されていない場合にのみ使用できます。
	一時停止	仮想マシンを一時停止します。このオプションは、仮想マシンがすでに実行されている場合にのみ使用できます。
	シャットダウン	<p>メニューを開いて、仮想マシンで実行する以下のアクションの1つを選択します。</p> <ul style="list-style-type: none"> ● 再起動 - 仮想マシンを再起動します。 ● シャットダウン - 仮想マシンをシャットダウンします。 ● 強制的にリセット - 仮想マシンを強制的にシャットダウンおよび再起動します。 ● 強制的に電源 OFF - 仮想マシンを強制的にシャットダウンします。 ● 保存 - 仮想マシンの状態をファイルに保存します。
	クローン	仮想マシンのクローンを作成します。詳細は『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「 virt-manager を使用したゲストのクローン作成 」を参照してください。
	移行	「仮想マシンの移行」ダイアログを開き、仮想マシンを他のホストに移行します。詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「 virt-manager を使用した移行 」を参照してください。
	削除	仮想マシンを削除します。
	スクリーンショットの取得	仮想マシンコンソールのスクリーンショットを取得します。
	USB デバイスのリダイレクト	「USB リダイレクト対象のデバイスを選択してください」ダイアログを開き、リダイレクトする USB デバイスを選択します。詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「 USB リダイレクト 」を参照してください。

メニュー名	メニュー項目	説明
表示	コンソール	仮想マシンペインにコンソール表示を開きます。
	詳細	仮想マシンペインに詳細表示を開きます。詳細については、「 仮想マシンの詳細ウィンドウ 」を参照してください。
	スナップショット	仮想マシンペインにスナップショット表示を開きます。詳細については、「 スナップショットウィンドウ 」を参照してください。
	フルスクリーン	仮想マシンコンソールをフルスクリーンモードで表示します。
	仮想マシンの画面サイズに合わせる	フルスクリーンのディスプレイのサイズを、仮想マシンに設定されたサイズおよび解像度に変更します。
	画面の縮小拡大	<p>以下のサブメニュー項目の選択に応じて、仮想マシンの表示を調整します。</p> <ul style="list-style-type: none"> ● 常に行う - 仮想マシンの表示は常に仮想マシンウィンドウのサイズに調整されます。 ● フルスクリーン時のみ - 仮想マシンウィンドウがフルスクリーンモードであるときのみ仮想マシンの表示が仮想マシンウィンドウのサイズに調整されます。 ● しない - 仮想マシンの表示は仮想マシンウィンドウのサイズに調整されません。 ● 仮想マシンのウィンドウを自動的にリサイズ - 仮想マシンウィンドウのサイズが調整されると、仮想マシンの表示サイズも自動的に調整されます。
	テキストコンソール	リストで選択した仮想マシンの表示を表示します。仮想マシンの表示には、 シリアル 1 や グラフィカルコンソール Spice などがあります。
	ツールバー	仮想マシンウィンドウツールバーの表示を切り替えます。


メニュー名	メニュー項目	説明
キーの送信	Ctrl+Alt+Backspace	選択したキーを仮想マシンに送信します。
	Ctrl+Alt+Delete	
	Ctrl+Alt+F1	
	Ctrl+Alt+F2	
	Ctrl+Alt+F3	
	Ctrl+Alt+F4	
	Ctrl+Alt+F5	
	Ctrl+Alt+F6	
	Ctrl+Alt+F7	
	Ctrl+Alt+F8	
	Ctrl+Alt+F9	
	Ctrl+Alt+F10	
	Ctrl+Alt+F11	
	Ctrl+Alt+F12	
	Ctrl+Alt+Printscreen	

5.2.2.2. 仮想マシンウィンドウツールバー

以下の表は、仮想マシンウィンドウにあるアイコンを示しています。

表5.4 仮想マシンウィンドウツールバー

アイコン	説明
	仮想マシンのグラフィカルコンソールを表示します。
	仮想マシンの詳細ペインを表示します。
	選択した仮想マシンを起動します。
	選択した仮想マシンを一時停止します。

アイコン	説明
	選択した仮想マシンを停止します。
	メニューを開き、選択した仮想マシンで実行する以下のアクションの 1 つを選択します。 <ul style="list-style-type: none">● 再起動 - 選択した仮想マシンを再起動します。● シャットダウン - 選択した仮想マシンをシャットダウンします。● 強制的にリセット - 選択した仮想マシンを強制的にシャットダウンおよび再起動します。● 強制的に電源OFF - 選択した仮想マシンを強制的にシャットダウンします。● 保存 - 選択した仮想マシンをファイルに保存します。
	仮想マシンペインにスクリーンショット表示を開きます。
	仮想マシンコンソールをフルスクリーンモードで表示します。

5.2.2.3. 仮想マシンペイン

仮想マシンペインには以下の 1 つが表示されます。

- [仮想マシンコンソール](#)
- [仮想マシン詳細ウィンドウ](#)
- [スナップショットウィンドウ](#)

仮想マシンコンソール

仮想マシンコンソールには、仮想マシンのグラフィック出力が表示されます。

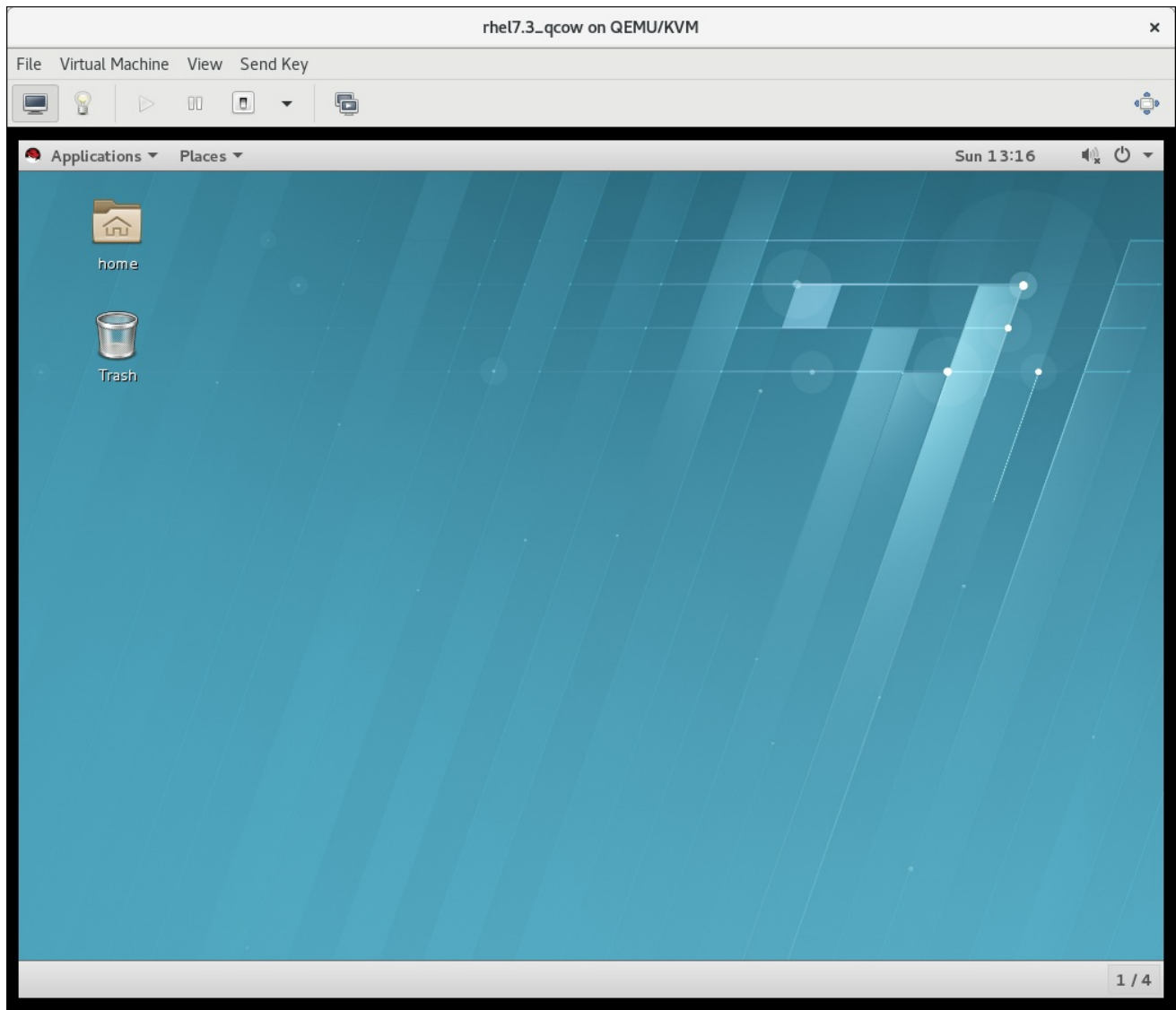


図5.5 仮想マシンコンソール

実際のマシンと同じように、マウスとキーボードを使用して仮想マシンコンソールと対話することができます。仮想マシンコンソールには、仮想マシンで実行されているアクティビティが表示されます。

仮想マシンの詳細ウィンドウ

仮想マシンの詳細ウィンドウには、仮想マシン、そのハードウェア、および設定に関する詳細情報が表示されます。

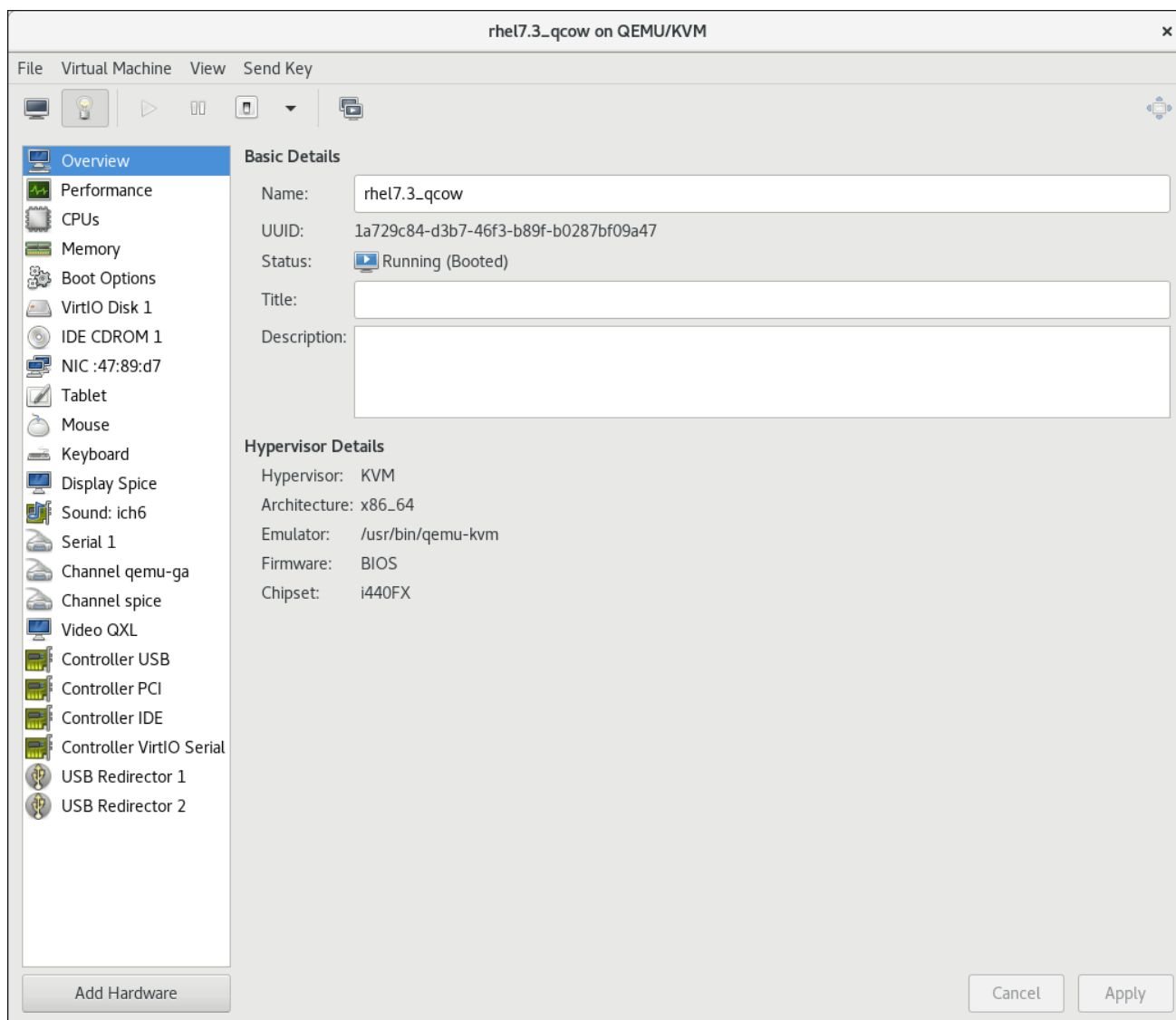


図5.6 仮想マシンの詳細ウィンドウ

仮想マシンの詳細ウィンドウには、仮想マシンのパラメーターのリストが含まれます。リストのパラメーターが選択されると、選択されたパラメーターに関する情報が仮想マシン詳細ウィンドウの右側に表示されます。また、仮想マシンの詳細ウィンドウを使用してハードウェアを追加および設定することもできます。

仮想マシンの詳細ウィンドウに関する詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「[仮想ハードウェアの詳細ウィンドウ](#)」を参照してください。

スナップショットウィンドウ

仮想マシンのスナップショットウィンドウには、仮想マシンに作成されたスナップショットのリストが表示されます。

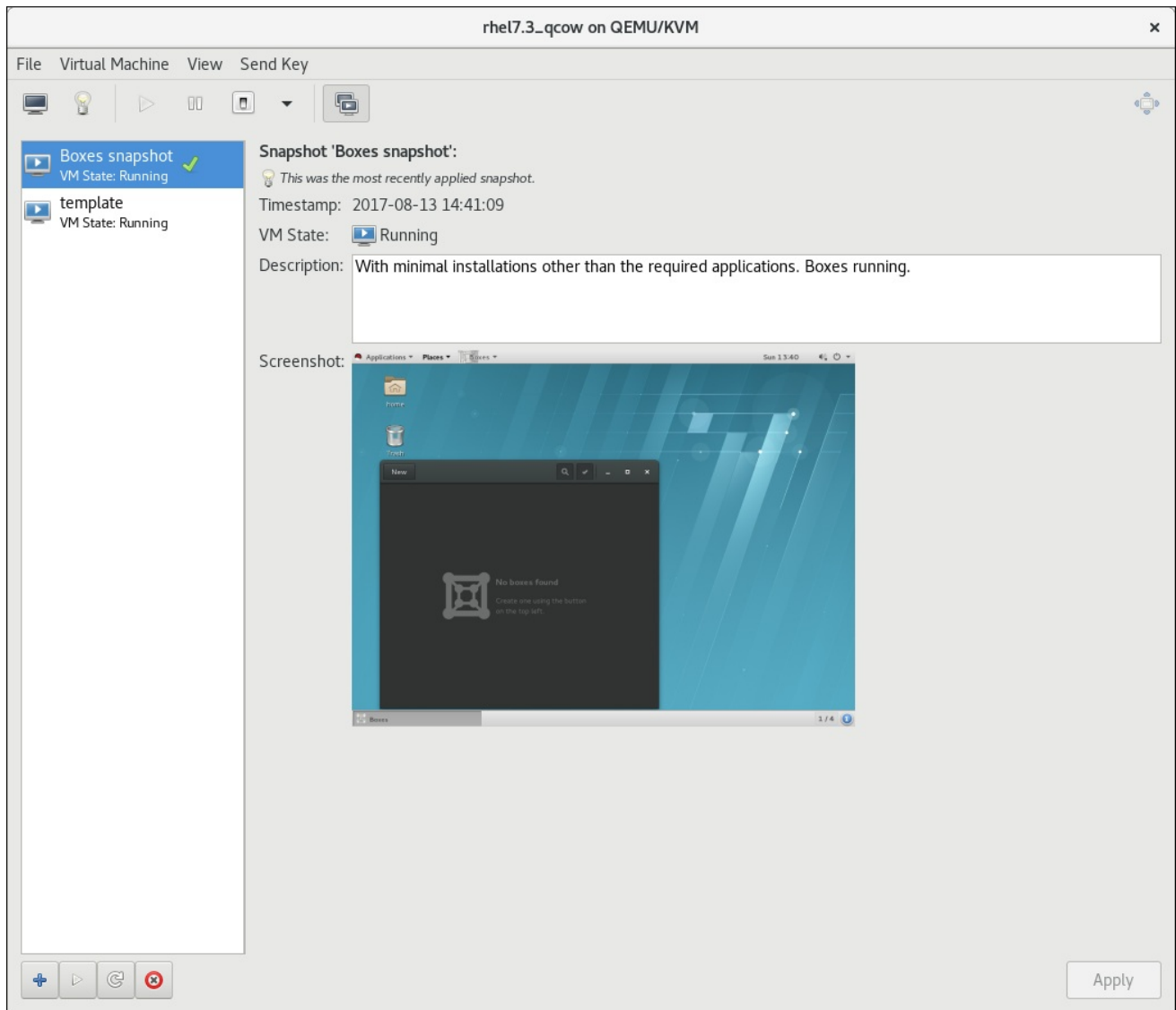


図5.7 仮想マシンのスナップショットウィンドウ

仮想マシンスナップショットウィンドウには、仮想マシン用に保存されたスナップショットのリストが表示されます。リストのスナップショットが選択されると、状態、説明、およびスクリーンショットを含む、選択されたスナップショットの詳細が仮想マシンスナップショットウィンドウの右側に表示されます。仮想マシンのスナップショットウィンドウを使用すると、スナップショットを追加、削除、および実行できます。

スナップショットの管理に関する詳細は、『Red Hat Enterprise Linux 仮想化の導入および管理ガイド』の「[スナップショットの管理](#)」を参照してください。

付録A 改訂履歴

改訂 1.0-53.1 翻訳ファイルを XML ソースバージョン 1.0-53 と同期	Wed Sep 5 2018	Ludek Janda
改訂 1.0-53 7.6 ベータ版公開用バージョン	Thu Aug 5 2018	Jiri Herrmann
改訂 1.0-52 7.5 GA 公開用バージョン	Thu Apr 5 2018	Jiri Herrmann
改訂 1.0-49 7.4 GA 公開用バージョン	Thu Jul 27 2017	Jiri Herrmann
改訂 1.0-46 7.3 GA 公開用バージョン	Mon Oct 17 2016	Jiri Herrmann
改訂 1.0-44 バグ修正後の本ガイドの再発行	Mon Dec 21 2015	Laura Novich
改訂 1.0-43 改訂履歴の整理	Thu Oct 08 2015	Jiri Herrmann
改訂 1.0-42 7.2 ベータリリース用の更新	Sun Jun 28 2015	Jiri Herrmann