

Fedora Documentation

Guide de démarrage de la virtualisation

Documentation de la virtualisation



Fedora Documentation Guide de démarrage de la virtualisation

Documentation de la virtualisation

Édition 19.0.1

Auteur

Copyright © 2012-2013 Red Hat, Inc. and others..

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/>. The original authors of this document, and Red Hat, designate the Fedora Project as the "Attribution Party" for purposes of CC-BY-SA. 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, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to https://fedoraproject.org/wiki/Legal:Trademark_guidelines.

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.

All other trademarks are the property of their respective owners.

Le guide de démarrage de la virtualisation décrit les principes de base de la virtualisation et des produits et technologies de virtualisation disponibles dans Fedora.

Préface	v
1. Conventions d'écriture	v
1.1. Conventions typographiques	v
1.2. Conventions pour citations mises en avant	vii
1.3. Notes et avertissements	vii
2. Vos commentaires sont importants !	viii
1. Introduction	1
1.1. À qui s'adresse la lecture de ce guide ?	1
1.2. Virtualization in Fedora Linux	1
1.3. Ressources en vue d'une virtualisation	1
2. Qu'est-ce que la virtualisation et la migration ?	3
2.1. Qu'est-ce que la virtualisation ?	3
2.2. What is migration?	3
2.2.1. Bénéfices de la migration de machines virtuelles	4
3. Avantages de la virtualisation et idées fausses à son propos	5
3.1. Les coûts de la virtualisation	5
3.2. Courbe d'apprentissage de la virtualisation	5
3.3. Performances	5
3.4. Souplesse	6
3.5. Reprise sur sinistre	6
3.6. Sécurité	6
3.6.1. Fonctionnalités de sécurité de la virtualisation	6
3.7. Virtualisation pour serveurs ou machines isolées	7
4. Présentation des produits Fedora pour la virtualisation	9
4.1. KVM et virtualisation dans Fedora	9
4.2. libvirt et ses outils	10
4.3. Périphériques matériels virtualisés	11
4.3.1. Périphériques émulés et virtualisés	11
4.3.2. Périphériques para-virtualisés	13
4.3.3. Périphériques d'hôtes physiques	14
4.3.4. CPU models	15
4.4. Stockage	16
4.4.1. Pools de stockage	16
4.4.2. Storage volumes	16
5. Outils pour la virtualisation	17
5.1. virsh	17
5.2. virt-manager	17
5.3. virt-install	17
5.4. guestfish	18
5.5. GNOME Boxes	18
5.6. Autre outils utiles	18
A. Historique de modifications	23

Préface

1. Conventions d'écriture

Ce manuel utilise plusieurs conventions pour souligner l'importance de certains mots ou expressions, mais aussi en vue d'attirer l'attention sur certains passages d'informations précis.

Pour les éditions sur support papier et numérique (PDF), ce manuel utilise des caractères issus de *Liberation Fonts*¹. La police de caractères Liberation Fonts est également utilisée pour les éditions HTML si elle est installée sur votre système. Sinon, des polices de caractères alternatives équivalentes sont utilisées. Notez que Red Hat Enterprise Linux 5 et versions supérieures contiennent la police Liberation Fonts par défaut.

1.1. Conventions typographiques

Quatre conventions typographiques sont utilisées pour attirer l'attention sur certains mots et expressions. Ces conventions et les circonstances auxquelles elles s'appliquent sont les suivantes.

Caractères gras à espacement fixe

Utilisée pour surligner certaines entrées du système, comme les commandes de console, les noms de fichiers et les chemins d'accès. Également utilisé pour surligner les touches et les combinaisons de touches. Par exemple :

Pour consulter le contenu du fichier **mon_nouvel_ouvrage_littéraire** qui se situe dans votre dossier courant, saisissez la commande **cat mon_nouvel_ouvrage_littéraire** à la demande du terminal et appuyez sur **Entrée** pour exécuter la commande.

L'exemple ci-dessus contient un nom de fichier, une commande-console et un nom de touche, tous présentés sous forme de caractères gras à espacement fixe et tous bien distincts grâce au contexte.

Les combinaisons de touches sont différenciées des noms de touches par le caractère « plus » (« + ») qui fait partie de chaque combinaison de touches. Ainsi :

Appuyez sur **Entrée** pour exécuter la commande.

Appuyez sur **Ctrl+Alt+F2** pour passer au premier terminal virtuel. Appuyez sur **Ctrl+Alt+F1** pour retourner à votre session X-Windows.

Le premier paragraphe surligne la touche précise sur laquelle il faut appuyer. Le second surligne deux combinaisons de touches (chacun étant un ensemble de trois touches à presser simultanément).

Si le code source est mentionné, les noms de classes, les méthodes, les fonctions, les noms de variables et les valeurs de retour citées dans un paragraphe seront présentées comme ci-dessus, en **caractères gras à espacement fixe**. Par exemple :

Les classes de fichiers comprennent le nom de classe **filesystem** pour les noms de fichier, **file** pour les fichiers et **dir** pour les dossiers. Chaque classe correspond à un ensemble de permissions associées.

¹ <https://fedorahosted.org/liberation-fonts/>

Caractères gras proportionnels

Cette convention marque le surlignage des mots ou phrases que l'on rencontre sur un système, comprenant des noms d'application, des boîtes de dialogue textuelles, des boutons étiquetés, des cases à cocher et des boutons d'options mais aussi des intitulés de menus et de sous-menus. Par exemple :

Sélectionnez **Système** → **Préférences** → **Souris** à partir de la barre du menu principal pour lancer les **Préférences de la souris**. À partir de l'onglet **Boutons**, cliquez sur la case à cocher **Pour gaucher** puis cliquez sur **Fermer** pour faire passer le bouton principal de la souris de la gauche vers la droite (ce qui permet l'utilisation de la souris par la main gauche).

Pour insérer un caractère spécial dans un fichier **gedit**, choisissez **Applications** → **Accessoires** → **Table de caractères** à partir de la barre du menu principal.

Ensuite, sélectionnez **Rechercher** → **Rechercher...** à partir de la barre de menu de **Table de caractères**, saisissez le nom du caractère dans le champ **Rechercher** puis cliquez sur **Suivant**. Le caractère que vous recherchez sera surligné dans la **Table de caractères**. Double-cliquez sur le caractère surligné pour l'insérer dans le champ **Texte à copier**, puis cliquez sur le bouton **Copier**. Maintenant, revenez à votre document et sélectionnez **Édition** → **Coller** à partir de la barre de menu de **gedit**.

Le texte ci-dessus contient des noms d'applications, des noms de menus et d'autres éléments s'appliquant à l'ensemble du système, des boutons et textes que l'on trouve dans une interface graphique. Ils sont tous présentés sous la forme gras proportionnel et identifiables en fonction du contexte.

Italique gras à espacement fixe ou *Italique gras proportionnel*

Qu'ils soient en caractères gras à espacement fixe ou à caractères gras proportionnels, l'ajout de l'italique indique la présence de texte remplaçable ou variable. Les caractères en italique indiquent la présence de texte que vous ne saisissez pas littéralement ou de texte affiché qui change en fonction des circonstances. Par exemple :

Pour se connecter à une machine distante en utilisant ssh, saisissez **ssh nom d'utilisateur@domain.name (nom.domaine)** après l'invite de commande de la console. Si la machine distante est **exemple.com** et que votre nom d'utilisateur pour cette machine est john, saisissez **ssh john@example.com**.

La commande **mount -o remount système de fichiers** monte le système de fichiers nommé. Ainsi, pour monter **/home** dans le système de fichiers, la commande est **mount -o remount /home**.

Pour connaître la version d'un paquet actuellement installé, utilisez la commande **rpm -q paquet**. Elle vous permettra de retourner le résultat suivant : **version-de-paquet**.

Notez les mots en caractères italiques et gras au dessus de — nom d'utilisateur, domain.name, système fichier, paquet, version et mise à jour. Chaque mot est un paramètre substituable de la ligne de commande, soit pour le texte que vous saisissez suite à l'activation d'une commande, soit pour le texte affiché par le système.

Mis à part l'utilisation habituelle de présentation du titre d'un ouvrage, les caractères italiques indiquent l'utilisation initiale d'un terme nouveau et important. Ainsi :

Publican est un système de publication *DocBook*.

1.2. Conventions pour citations mises en avant

Les sorties de terminaux et les citations de code source sont mis en avant par rapport au texte avoisinant.

Les sorties envoyées vers un terminal sont en caractères **Romains à espacement fixe** et présentées ainsi :

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads     images  notes  scripts svgs
```

Les citations de code source sont également présentées en **romains à espacement fixe** mais sont présentés et surlignés comme suit :

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;


public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo    = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. Notes et avertissements

Enfin, nous utilisons trois styles visuels pour attirer l'attention sur des informations qui auraient pu être normalement négligées :


Remarque

Une remarque est une forme de conseil, un raccourci ou une approche alternative par rapport à une tâche à entreprendre. L'ignorer ne devrait pas provoquer de conséquences négatives, mais vous pourriez passer à côté d'une astuce qui vous aurait simplifiée la vie.



Important

Les blocs d'informations importantes détaillent des éléments qui pourraient être facilement négligés : des modifications de configurations qui s'appliquent uniquement à la session actuelle ou des services qui ont besoin d'être redémarrés avant toute mise à jour. Si vous ignorez une case étiquetée « Important », vous ne perdrez aucune donnée mais cela pourrait être source de frustration et d'irritation.



Avertissement

Un avertissement ne devrait pas être ignoré. Ignorer des avertissements risque fortement d'entraîner des pertes de données.

2. Vos commentaires sont importants !

Si vous trouvez des fautes de frappe ou si vous avez des suggestions pour améliorer ce manuel, n'hésitez surtout pas à nous en faire part ! Veuillez envoyer vos remarques par l'entremise de Bugzilla (<http://bugzilla.redhat.com/bugzilla/>) pour le produit **Fedora 19**.

Lorsque vous soumettez un rapport d'erreurs, veuillez indiquer clairement les références du manuel : *virtualization-getting-started-guide*

Si vous avez des suggestions pour améliorer la documentation, essayez de les décrire le plus précisément possible. Si vous avez trouvé une erreur, veuillez non seulement indiquer le numéro de section où elle se trouve mais également ajouter un extrait du texte qui l'entoure, afin que nous puissions la retrouver facilement.

Introduction

The *Fedora Virtualization Getting Started Guide* introduces the basics of virtualization and assists with the navigation of other virtualization documentation and products that Fedora provides.

Ce guide explique aussi les avantages de la virtualisation, tout en essayant de dissiper certains mythes répandus à son sujet.

1.1. À qui s'adresse la lecture de ce guide ?

Ce guide est conçu pour toute personne désireuse de comprendre les bases de la virtualisation, mais il s'adresse plus particulièrement aux personnes qui :

- découvrent la virtualisation et recherchent des informations sur les avantages offerts,
- envisagent le déploiement de machines virtualisées dans leur environnement,
- recherchent un aperçu des techniques de virtualisation fournies et prises en charge par Fedora.

1.2. Virtualization in Fedora Linux

Fedora contains packages and tools to support a variety of virtualized environments.

Virtualization in Fedora is carried out by KVM (Kernel-based Virtual Machine). KVM is a full virtualization solution built into Fedora.

Refer to [Chapitre 4, Présentation des produits Fedora pour la virtualisation](#) for more about the virtualization products available in Fedora 19.

1.3. Ressources en vue d'une virtualisation

Fedora contient des paquets et des outils pour prendre en charge une grande variété d'environnements virtualisés. La virtualisation dans Fedora met à votre disposition les développements amont lors de la mise au point de la virtualisation pour Red Hat Enterprise Linux. Se reporter au [Chapitre 4, Présentation des produits Fedora pour la virtualisation](#) pour plus d'informations sur les produits de virtualisation disponibles dans Fedora.

Outre ce guide, les ouvrages suivants portent sur la virtualisation avec Fedora :

- *Fedora Virtualization Deployment and Administration Guide* covers the installation, configuration, and maintenance of virtualization hosts and virtual machines.
- *Fedora Virtualization Security Guide*: This guide provides information on virtualization security technologies including sVirt, configuration and recommendations for host and guest security, and network security.

Red Hat Enterprise Virtualization (RHEV) est une autre option de virtualisation pour les serveurs et les postes de travail. Red Hat Enterprise Virtualization fournit une solution de virtualisation bout à bout pour les centres de données, développée à partir du projet amont oVirt. Visitez <https://fedorahosted.org/ovirt/> pour plus d'informations sur oVirt.

La documentation de Red Hat Enterprise Virtualization et la virtualisation avec Red Hat Enterprise Linux est disponible à l'adresse <http://docs.redhat.com/>.

Qu'est-ce que la virtualisation et la migration ?

Ce chapitre présente les termes courants relatifs à la virtualisation et à la migration.

2.1. Qu'est-ce que la virtualisation ?

Virtualization is a broad computing term used for running software, usually multiple operating systems, concurrently and in isolation from other programs on a single system. Most existing implementations of virtualization use a *hypervisor*, a software layer or subsystem that controls hardware and provides *guest operating systems* with access to underlying hardware. The hypervisor allows multiple operating systems, called *guests*, to run on the same physical system by offering virtualized hardware to the guest operating system. There are several methods for virtualizing operating systems.

Virtualization methods

Virtualisation complète

Full virtualization uses the hardware features of the processor to provide guests with total abstraction of the underlying physical system. This creates a new virtual system, called a *virtual machine*, that allows guest operating systems to run without modifications. The guest operating system and any applications on the guest virtual machine are unaware of their virtualized environment and run normally. Hardware-assisted virtualization is the technique used for full virtualization with KVM (Kernel-based Virtual Machine) in Fedora.

Para-virtualisation

La para-virtualisation utilise un ensemble de logiciels et de structures de données qui sont présentés à l'invité virtualisé, ce qui nécessite des modifications logicielles dans l'invité afin d'utiliser l'environnement para-virtualisé. La para-virtualisation peut englober la totalité du noyau, comme c'est le cas pour les invités Xen para-virtualisés ou les pilotes qui virtualisent les périphériques d'E/S.

Virtualisation logicielle (ou émulation)

La virtualisation logicielle utilise une traduction binaire et d'autres techniques d'émulation, toutes plus lentes, afin d'exécuter les systèmes d'exploitations non-modifiés.



Note

For more information and detailed instructions on guest installation, refer to the *Fedora Virtualization Deployment and Administration Guide*.

2.2. What is migration?

Migration describes the process of moving a guest virtual machine from one host to another. This is possible because guests are running in a virtualized environment instead of directly on the hardware. There are two ways to migrate a virtual machine: live and offline.

Migration types

Migration à froid

An offline migration suspends the guest virtual machine, and then moves an image of the virtual machine's memory to the destination host. The virtual machine is then resumed on the destination host and the memory used by the virtual machine on the source host is freed.

Migration à chaud

Live migration is the process of migrating an active virtual machine from one physical host to another.

It is important to understand that the migration process moves the virtual machine's memory, and the disk volume associated with the virtual machine is also migrated. This process is done using live block migration.

In Fedora 19, shared storage is not necessary for storing guest images to be migrated. With live storage migration, a running virtual machine can be migrated from one host to another with no downtime. This capability can be used to optimize performance of virtual machines.

2.2.1. Bénéfices de la migration de machines virtuelles

La migration permet de :

Répartir la charge

When a host machine is overloaded, one or many of its virtual machines could be migrated to other hosts using live migration.

Mettre à jour l'hôte ou y opérer des changements

Quand il est nécessaire de mettre à jour, d'ajouter ou de retirer du matériel sur un hôte, les machines virtuelles peuvent être déplacées de façon sûre. En d'autres termes, les invités virtuels ne subissent aucun arrêt de service du fait des changements qui sont réalisés sur l'un ou l'autre des hôtes.

Économiser l'énergie

Les machines virtuelles peuvent être redistribuées à d'autres hôtes, et ces derniers, ainsi libérés, peuvent être éteints afin d'économiser de l'énergie, ce qui diminue la dépense en période de faible utilisation.

Migrer géographiquement

Les machines virtuelles peuvent être déplacées en un autre lieu physique afin de réduire la latence ou profiter d'autres circonstances particulières.

Un stockage réseau partagé doit être utilisé pour l'entreposage des images des invités virtuels à faire migrer. Sans stockage partagé, la migration n'est pas possible. Il est recommandé d'utiliser **libvirt** pour gérer les pools de stockage partagés.



Note

For more information on migration, refer to the *Fedora Virtualization Deployment and Administration Guide*.

Avantages de la virtualisation et idées fausses à son propos

La virtualisation apporte beaucoup d'avantages, et peut-être autant d'idées fausses à son encontre. Ce chapitre explore ces sujets.

3.1. Les coûts de la virtualisation

La première idée fausse est que la virtualisation serait trop chère pour justifier un changement. La virtualisation peut être coûteuse à mettre en place, mais elle permet souvent d'économiser de l'argent sur le long terme. Il est important d'effectuer une analyse de retour sur investissement (ROI) pour décider du meilleur usage de la virtualisation dans un environnement donné. Voyez les avantages suivants :

Moins d'énergie

L'utilisation de la virtualisation réduit le nombre de plates-formes physiques nécessaires. Cela induit une moindre consommation d'énergie pour le fonctionnement et le refroidissement des machines exploitées, ayant pour résultat une réduction des coûts énergétiques. Le coût initial d'acquisition de plates-formes physiques multiples, combiné au fait que ces machines ont des consommations électriques et frigorifiques réduites, est radicalement réduit grâce à la virtualisation.

Moins de maintenance

Si une planification adéquate est effectuée avant la migration des systèmes physiques vers des environnements virtualisés, moins de temps est consacré à leur maintenance. Cela signifie moins d'argent dépensé pour des pièces et de la main-d'œuvre.

Durée de vie étendue des logiciels installés

Des logiciels, dans leur version la plus ancienne, peuvent ne pas pouvoir s'exécuter directement sur des machines plus récentes, de manière native. Néanmoins, en lançant ces anciens logiciels dans un environnement virtualisé sur un matériel plus puissant, la vie de ces logiciels peut être prolongée tout en profitant des avantages en termes de performances d'un système plus récent.

Empreinte plus légère

Le groupement des serveurs sur moins de machines permet de réduire l'espace physique nécessaire. Cela signifie que l'espace normalement occupé par le matériel serveur peut être utilisé pour autre chose.

3.2. Courbe d'apprentissage de la virtualisation

Une autre idée fausse au sujet de la virtualisation est qu'elle serait difficile à appréhender. En réalité, la virtualisation n'est ni plus ni moins difficile à aborder que tout nouveau processus. Les compétences requises pour administrer et prendre en charge un environnement physique sont facilement transposables à un environnement virtuel. Ils fonctionnent de façon similaire à leurs homologues physiques, ce qui permet d'affirmer que le rapport formation sur amélioration de la productivité est parmi les plus faibles.

3.3. Performances

Avec les plus anciennes versions de virtualisation qui ne prenaient en charge qu'une seule CPU, les machines virtuelles connaissaient des limitations de performances notables. Cela a été à l'origine d'une idée fausse, qui a la vie dure, prétendant que les solutions de virtualisation sont lentes. Ce n'est

plus le cas ; les progrès de la technique permettent à des machines virtuelles de tourner avec des niveaux de performance beaucoup plus importants qu'auparavant.

3.4. Souplesse

La virtualisation apporte une grande souplesse pour l'administration de systèmes. Les machines virtuelles peuvent être copiées ou déplacées afin de tester les mises à jour logicielles et valider les changements de configuration, sans impacter les autres systèmes. Du fait que les systèmes virtualisés sont cloisonnés entre eux, l'arrêt de service d'un système n'en affecte aucun autre.

3.5. Reprise sur sinistre

Une reprise sur sinistre est plus rapide et plus facile lorsque les systèmes sont virtualisés. Sur les systèmes physiques, si un problème sérieux survient, une réinstallation complète du système d'exploitation est souvent requise, nécessitant de nombreuses heures de travail de restauration. Si les prérequis d'une migration à chaud sont respectés, les machines virtuelles peuvent être redémarrées sur un autre hôte, et le délai potentiel le plus important sera celui de la restauration des données d'un invité virtuel.

3.6. Sécurité

Une machine virtuelle utilise SELinux et sVirt afin d'améliorer la sécurité de la virtualisation. Cette section donne un aperçu des options de sécurité disponibles.

3.6.1. Fonctionnalités de sécurité de la virtualisation

SELinux

SELinux a été développé par l'Agence Nationale de Sécurité (NSA) des États-Unis et d'autres afin de fournir une fonctionnalité de contrôle d'accès obligatoire (Mandatory Access Control, MAC) pour Linux. Sous le contrôle de SELinux, tous les processus et tous les fichiers se voient attribuer ce qu'on appelle un *type*, et l'accès est limité par une fine granulométrie de contrôles. SELinux limite les possibilités des personnes malveillantes, et permet de prévenir un grand nombre d'exploitations de failles de sécurité comme les débordements de tampons et les escalades de privilèges.

SELinux renforce le modèle de sécurité des hôtes et des invités virtuels Fedora. SELinux est configuré et testé pour être activé, par défaut, avec tous les outils de virtualisation embarqués dans Fedora.

sVirt

sVirt est une technique incorporée dans Fedora qui marie SELinux et virtualisation. Elle applique les contrôles d'accès obligatoires (MAC) afin d'améliorer la sécurité lors de l'utilisation de machines virtuelles ; elle améliore la sécurité et durcit le système contre des anomalies de l'hyperviseur qui pourraient être utilisées comme vecteur d'attaque sur l'hôte ou une autre machine virtuelle.



Note

For more information on security in Fedora, refer to the *Fedora Security Guide*. To find more information on security for virtualization, refer to the *Fedora Virtualization Security Guide*.

3.7. Virtualisation pour serveurs ou machines isolées

La virtualisation n'est pas réservée qu'aux serveurs ; elle peut aussi être utile pour des ordinateurs individuels. La virtualisation des ordinateurs de bureau permet une gestion centralisée, l'amélioration des solutions pour postes de travail, ainsi qu'une meilleure reprise en cas de sinistre. Il est possible de se connecter à un poste de travail distant en utilisant un logiciel de connexion.

Pour les serveurs, la virtualisation ne s'adresse pas qu'aux réseaux de grande taille ; elle s'applique à toute situation où sont utilisés deux serveurs ou plus. Elle fournit la migration à chaud, la haute disponibilité, la tolérance de faute et une simplification des sauvegardes.

Présentation des produits Fedora pour la virtualisation

Ce chapitre fait découvrir les divers produits de virtualisation disponibles dans Fedora.

4.1. KVM et virtualisation dans Fedora

Qu'est-ce que KVM ?

KVM (Kernel-based Virtual Machine — machine virtuelle au niveau noyau) est une solution de virtualisation complète pour Linux sur des matériels AMD64 et Intel 64 ; il est incorporé dans le noyau Fedora standard. Il peut exécuter plusieurs systèmes d'exploitation invités Linux ou Windows sans modification. L'hyperviseur KVM de Fedora est géré avec l'API **libvirt** et des outils dédiés à cette application (comme **virt-manager** et **virsh**). Les machines virtuelles sont lancées et tournent en tant que processus Linux à unités d'exécution multiples contrôlées par ces outils.

Surexploitation

L'hyperviseur KVM prend en charge une *surexploitation* des ressources système. Surexploiter signifie allouer plus de CPU ou de mémoire virtualisée que n'en dispose le système. La surexploitation de la mémoire permet aux hôtes d'utiliser la mémoire réelle et virtuelle pour augmenter le nombre d'invités.



Important

Overcommitting involves possible risks to system stability. For more information on overcommitting with KVM, and the precautions that should be taken, refer to the *Fedora Virtualization Deployment and Administration Guide*.

Allocation fine

L'allocation fine permet d'allouer des stockage de manière souple ; elle optimise l'espace disponible pour chaque invité. Elle donne l'illusion qu'il y a plus de stockage physique pour l'invité que celui réellement disponible. Ce n'est pas la même chose que la surexploitation étant donné qu'elle ne concerne que le stockage et non pas les allocations mémoire ou CPU. Mais, comme pour la surexploitation, les mêmes précautions s'appliquent.



Important

Thin provisioning involves possible risks to system stability.

KSM

Kernel SamePage Merging (KSM), utilisé par l'hyperviseur KVM, permet aux invités KVM de partager les mêmes pages mémoire. Ces pages partagées sont habituellement des bibliothèques communes ou autres données identiques d'utilisation fréquente. KSM permet d'éviter les doublons en mémoire lorsqu'il y a un grand nombre d'invités avec des systèmes d'exploitation identiques ou semblables.



Note

For more information on KSM, refer to the *Fedora Virtualization Deployment and Administration Guide*.

QEMU guest agent

The *QEMU guest agent* runs on the guest operating system and allows the host machine to issue commands to the guest operating system.



Note

For more information on the QEMU guest agent, refer to the *Fedora Virtualization Deployment and Administration Guide*.

KVM guest virtual machine compatibility

To verify whether your processor supports the virtualization extensions and for information on enabling the virtualization extensions if they are disabled, refer to the *Fedora Virtualization Deployment and Administration Guide*.

4.2. libvirt et ses outils

Le paquet *libvirt* contient une API de virtualisation, indépendante de l'hyperviseur, susceptible d'interagir avec les aptitudes à la virtualisation de toute une gamme de systèmes d'exploitation.

Le paquet *libvirt* met à disposition :

- une couche simple, générique et stable pour gérer en sécurité des machines virtuelles sur un hôte,
- une interface simple pour manier des systèmes locaux et des hôtes en réseau,
- toutes les API requises pour mettre à disposition, créer, modifier, contrôler, migrer et stopper des machines virtuelles, mais uniquement si l'hyperviseur prend en charge ces opérations. Même s'il est possible d'avoir accès à plusieurs hôtes simultanément avec **libvirt**, les API sont limitées à des opérations sur un seul nœud.

Le paquet *libvirt* a été conçu comme un bloc de construction pour outils et applications de gestion pour un niveau plus élevé, par exemple, les outils de gestion en ligne de commande **virt-manager** et **virsh**. Exception faite des possibilités de migration, **libvirt** est focalisé sur une simple gestion d'hôtes ; il fournit des API pour décompter, contrôler et utiliser les ressources disponibles sur le nœud géré, y compris les CPU, la mémoire, le stockage, le travail en réseau et les partitions « Non-Uniform Memory Access (NUMA) » (architecture mémoire non uniforme). Les outils de gestion peuvent être situés sur des machines physiquement séparées de l'hôte en utilisant des protocoles sécurisés.

Fedora prend en charge **libvirt**, y compris ses outils associés, comme méthode par défaut pour la gestion de la virtualisation.

Le paquet *libvirt* est disponible en tant que logiciel libre sous licence « GNU Lesser General Public License ». Le projet *libvirt* vise à fournir, sur le long terme, une API C stable pour des outils de gestion de la virtualisation, en s'exécutant au-dessus de divers types d'hyperviseur.

virsh

L'outil en ligne de commande **virsh** est construit sur l'API de gestion **libvirt** ; il est une alternative à l'application graphique **virt-manager**. La commande **virsh** peut être utilisé en mode lecture seule par les utilisateurs non-privilégiés ; en mode administrateur, il permet toutes les fonctionnalités d'administration. La commande **virsh** est idéale pour la gestion des scripts de virtualisation.

virt-manager

virt-manager est un outil graphique d'ordinateur de bureau pour gérer des machines virtuelles. Il autorise l'accès aux consoles graphiques des invités ; il peut s'utiliser pour administrer la virtualisation, créer des machines virtuelles, migrer et exécuter les tâches de configuration. Il offre aussi la possibilité de voir des machines virtuelles, des statistiques d'hôtes, des informations sur des périphériques et des graphiques de performances. Les hyperviseurs tant locaux que distants peuvent être gérés par l'intermédiaire d'une interface simple.

Note

For more information on **virsh** and **virt-manager**, refer to the *Fedora Virtualization Deployment and Administration Guide*.

4.3. Périphériques matériels virtualisés

La virtualisation sur Fedora offre trois types distincts de périphériques système pour les machines virtuelles. Ce sont les :

- Périphériques émulés et virtualisés
- Périphériques para-virtualisés
- Périphériques physiques partagés

Ces périphériques matériels sont tous vus comme physiquement rattachés à la machine virtuelle, mais les pilotes de périphérique fonctionnent de manière différente.

4.3.1. Périphériques émulés et virtualisés

KVM implémente dans le noyau de nombreux périphériques pour machines virtuelles de manière logicielle. Ces périphériques matériels émulés sont d'une importance majeure pour les systèmes d'exploitation virtualisés.

Les périphériques émulés sont des périphériques virtuels, n'existant que de manière logique.

Les pilotes émulés peuvent utiliser, soit un périphérique physique, soit un périphérique logique virtuel. Les pilotes émulés sont une couche de transition entre la machine virtuelle et le noyau Linux (qui gère le périphérique source). Les instructions au niveau périphérique sont totalement traduites par l'hyperviseur KVM. Tout périphérique, du même type (stockage, réseau, clavier ou souris) reconnu par le noyau Linux, peut être utilisé comme périphérique source d'appui pour les pilotes émulés.

CPU virtuelles (vCPU)

Un système hôte peut avoir jusqu'à 160 CPU virtuelles (vCPU) pouvant être présentées pour utilisation aux invités, quel que soit le nombre de CPU de l'hôte.

Périphériques graphiques émulés

Deux périphériques graphique émulés sont à disposition. Ces périphériques peuvent être connectés avec le protocole SPICE (Simple Protocol for Independent Computing Environments) (Simple protocole pour environnements de calcul indépendants) ou bien avec VNC :

- une carte VGA PCI Cirrus CLGD 5446 (utilisant le périphérique *cirrus*),
- une carte graphique standard VGA avec extensions VESA Bochs (niveau matériel, y compris tous les modes non normalisés).

Composants système émulés

Les composants système du noyau énumérés ci-après sont émulés pour fournir les fonctions système de base :

- pont PCI hôte Intel i440FX,
- pont PCI PIIX3 vers ISA,
- souris et clavier PS/2,
- EvTouch USB graphics tablet
- contrôleur USB UHCI PCI et un hub USB virtualisé,
- ports série émulés,
- contrôleur EHCI, stockage USB virtualisé et souris USB,
- USB 3.0 xHCI host controller

périphériques son émulés.

Fedora provides an emulated (Intel) HDA sound device, `intel-hda`.

Les deux périphériques son émulés suivants sont également disponibles, mais leur utilisation n'est pas recommandée en raison de problèmes de compatibilité avec certains systèmes d'exploitation invités :

- `ac97`, une carte son émulée compatible Audio AC97 Intel 82801AA
- `es1370`, une carte son émulée AudioPCI ENSONIQ ES1370.

Périphériques de surveillance émulés

Fedora provides two emulated watchdog devices. A watchdog can be used to automatically reboot a virtual machine when it becomes overloaded or unresponsive.

Le paquet *watchdog* doit être installé sur l'invité.

Les deux périphériques disponibles sont :

- `i6300esb`, un périphérique de surveillance PCI Intel 6300 ESB émulé.
- `ib700`, un périphérique de surveillance ISA iBase 700 émulé.

Périphériques réseau émulés

Il y a deux périphériques réseau émulés disponibles :

- le périphérique `e1000` mime un adaptateur réseau Intel E1000 (Intel 82540EM, 82573L, 82544GC),
- le périphérique `rtl8139` mime un adaptateur réseau Realtek 8139.

Pilotes de stockage émulés

Les périphériques de stockage ou les pools de stockage peuvent se servir de ces pilotes pour attacher des périphériques de stockage aux machines virtuelles. L'invité utilise un pilote de stockage émulé pour avoir accès au stockage.

Notez que, comme pour tous périphériques virtuels, les pilotes de stockage ne sont pas des périphériques. Ils sont utilisés pour attacher un périphérique de stockage d'appui, un fichier ou un volume du pool de stockage à une machine virtuelle. Le périphérique de stockage d'appui peut être n'importe quel type de périphérique de stockage pris en charge, fichier ou volume du pool de stockage.

Le pilote IDE émulé

KVM met à disposition deux interfaces PCI IDE émulées. Un pilote IDE émulé peut être utilisé pour attacher n'importe quelle combinaison, jusqu'à quatre unités, de disques durs IDE ou lecteurs virtualisés de CD-ROM IDE à chaque machine virtuelle. Le pilote IDE émulé est également utilisé pour les lecteurs CD-ROM et DVD-ROM virtualisés.

Le pilote de lecteur de disquettes émulé


Le pilote de lecteur de disquettes émulé s'utilise pour créer des lecteurs de disquettes virtuels.

4.3.2. Périphériques para-virtualisés

Les périphériques para-virtualisés sont des pilotes pour des périphériques virtuels qui améliorent les performances en E/S des machines virtuelles.

Para-virtualized devices decrease I/O latency and increase I/O throughput to near bare-metal levels. It is recommended to use the para-virtualized device drivers for virtual machines running I/O intensive applications.

The para-virtualized devices must be installed on the guest operating system. The para-virtualized device drivers must be manually installed on Windows guests.


Note

For more information on using the para-virtualized drivers, refer to the *Fedora Virtualization Deployment and Administration Guide*.

Le pilote réseau para-virtualisé (virtio-net)

Ce pilote réseau peut s'utiliser en tant que pilote pour les périphériques réseau existants ou pour les nouveaux périphériques réseau pour machines virtuelles.

Le pilote mode bloc para-virtualisé (virtio-blk)

Ce pilote para-virtualisé mode bloc est un pilote pour tous les périphériques de stockage ; il est pris en charge par l'hyperviseur et est rattaché à la machine virtuelle (excepté pour les lecteurs de disquettes, qui doivent être émulés).

Le contrôleur de périphérique para-virtualisé (virtio-scsi)

Le pilote de contrôleur SCSI para-virtualisé est une alternative plus souple et plus extensible à virtio-blk. Un invité virtio-scsi est susceptible d'hériter l'ensemble des fonctionnalités du périphérique cible ; il peut gérer des centaines de périphériques comparativement à virtio-blk qui ne peut en piloter que 28.

L'horloge para-virtualisée

Les invités utilisant le chronographe « Time Stamp Counter (TSC) » en tant qu'horloge source peuvent être confrontés à des problèmes de mesure du temps. KVM contourne les problèmes des hôtes qui ne disposent pas d'un chronographe fiable en mettant à leur disposition une horloge para-virtualisée.

Le pilote série para-virtualisé (virtio-serial)

Le pilote série para-virtualisé est un pilote orienté flux en mode caractère ; il fournit une interface de communication simple entre l'espace utilisateur de l'hôte et celui de l'invité.

Le pilote bulle (virtio-balloon)

Le pilote bulle peut marquer des parties de RAM d'une machine virtuelle comme n'étant pas utilisées (un processus nommé *gonflement* de la bulle) ; ainsi, cette mémoire peut être libérée pour être utilisée par l'hôte (ou pour d'autres machines virtuelles sur cet hôte). Quand la machine virtuelle a, à nouveau, besoin de mémoire, la bulle peut être *dégonflée* et l'hôte peut rendre la RAM à la machine virtuelle.

The para-virtualized random number generator (virtio-rng)

The para-virtualized random number generator enables virtual machines to collect entropy, or randomness, directly from the host to use for encrypted data and security. Virtual machines can often be starved of entropy because typical inputs (such as hardware usage) are unavailable. Sourcing entropy can be time-consuming; virtio-rng makes this process faster by injecting entropy directly into guest virtual machines from the host.

4.3.3. Périphériques d'hôtes physiques

Certaines plates-formes autorisent l'accès direct des machines virtuelles aux divers périphériques et composants matériels. Ce processus, en virtualisation, est connu sous le terme d'*assignation de périphérique*. Il est aussi désigné sous le nom de *traverse*.

Assignation d'un périphérique PCI

L'hyperviseur KVM accepte l'attachement de périphériques PCI du système hôte aux machines virtuelles. L'assignation des périphériques PCI permet à l'invité d'avoir un accès exclusif aux périphériques pour un ensemble de tâches. Les périphériques PCI apparaissent et se comportent comme s'ils étaient physiquement attachés à la machine virtuelle invitée.

L'assignation de périphérique est acceptée sur les périphériques PCI Express, à l'exception des cartes graphiques. Les périphériques PCI parallèles peuvent être pris en charge en tant que périphériques assignés, mais il y a de sévères limitations pour des raisons de sécurité et de conflits de configuration du système.

Note

For more information on device assignment, refer to the *Fedora Virtualization Deployment and Administration Guide*.

Traverse USB

L'hyperviseur KVM prend en charge l'attachement de périphériques USB sur le système hôte aux machines virtuelles. L'assignation de périphérique USB permet aux invités d'avoir un accès exclusif aux périphériques USB pour un ensemble de tâches. Les périphériques USB apparaissent et se comportent comme s'ils étaient physiquement attachés à la machine virtuelle invitée.

SR-IOV

SR-IOV (Single Root I/O Virtualization) est une norme PCI Express visant à l'extension d'une fonction physique PCI unique pour qu'elle partage ses ressources PCI en fonctions virtuelles distinctes (VFs). Chaque fonction est susceptible d'être utilisée par une machine virtuelle différente par l'intermédiaire d'une assignation de périphérique PCI.

Un périphérique PCI-e, compatible SR-IOV, met à disposition une fonction racine unique (par exemple, un port Ethernet unique) et présente plusieurs périphériques virtuels séparés comme des fonctions de périphériques PCI individuels. Chaque périphérique virtuel doit avoir son propre et unique espace de configuration PCI, des registres mappés en mémoire et des interruptions type MSI propres.



Note

For more information on SR-IOV, refer to the *Fedora Virtualization Deployment and Administration Guide*.

NPV

N_Port ID Virtualization (NPV) est une fonctionnalité disponible sur certains périphériques « Fibre Channel ». NPV partage un unique N_Port (node port) physique sous forme de plusieurs identifiants de N_Port. NPV met à disposition pour les « Fibre Channel Host Bus Adapters (HBA) » (adaptateurs de bus hôte Fibre Channel) une fonctionnalité similaire à celle que SR-IOV fournit pour les interfaces PCIe. Avec NPV, les machines virtuelles ont à leur disposition un point d'entrée de canal « Fibre Channel » virtuel vers les « Storage Area Networks (SAN) » (réseaux de stockage).

NPV permet des environnements virtualisés de haute densité avec des solutions de stockage au niveau entreprise.

4.3.4. CPU models

CPU models define which host CPU features are available to the guest operating system. **qemu-kvm** and **libvirt** contain definitions for a number of current processor models, allowing users to enable features that are available only in newer CPU models. The CPU feature set available to guests depends on support in the host CPU kernel. The **qemu-kvm** code must also allow the feature to be enabled.

To safely migrate virtual machines between hosts with different CPU feature sets, **qemu-kvm** does not expose all CPU features from the host CPU to guest operating systems by default. Instead, CPU features are exposed to virtual machines based on the chosen CPU model.

It is also possible to enable or disable specific CPU features in a virtual machine's XML configuration. However, it is safer to use predefined CPU models, as incorrect configuration can cause compatibility issues with the guest operating system.



Note

For more information on CPU model availability and configuration, refer to the *Fedora Virtualization Deployment and Administration Guide*.

4.4. Stockage

Le stockage des machines virtuelles est disjoint du stockage physique utilisé par la machine virtuelle. Il est rattaché à la machine virtuelle à l'aide de pilotes de périphériques en mode bloc émulés ou para-virtualisés.

4.4.1. Pools de stockage

Un *pool de stockage* est un fichier, un répertoire ou un périphérique de stockage géré par **libvirt** dans le but de mettre à disposition un stockage pour les machines virtuelles. Les pools de stockage sont divisés en *volumes* de stockage qui emmagasinent des images de la machine virtuelle ou qui sont rattachés aux machines virtuelles en tant que stockage d'appoint. Plusieurs invités peuvent partager le même pool de stockage, ce qui permet une meilleure allocation des ressources de stockage.

Pools de stockage locaux

Les pools de stockage locaux sont directement attachés au serveur hôte. Cela comprend les répertoires locaux, les disques directement attachés, les partitions physiques et les groupes de volumes LVM sur des périphériques locaux. Les pools de stockage locaux sont utiles pour le développement, les petits déploiements ou ceux de test qui ne nécessitent pas de migration ou bien pour des grands nombres de machines virtuelles. Les pools de stockage locaux ne conviennent pas aux environnements de production étant donné qu'ils ne prennent pas en charge la migration à chaud.

Pools de stockage sur réseau (partagés)

Networked storage pools include storage devices shared over a network using standard protocols. Networked storage is required when migrating virtual machines between hosts with **virt-manager**, but is optional when migrating with **virsh**. Networked storage pools are managed by **libvirt**.

4.4.2. Storage volumes

Les pools de stockage sont divisés en volumes de stockage. Les volumes de stockage sont une abstraction de partitions physiques, de volumes logiques LVM, d'images disque constituées de fichiers et des autres types de stockages gérés par **libvirt**. Les volumes de stockage se présentent, pour les machines virtuelles, comme des périphériques de stockage locaux quel que soit le matériel sous-jacent.



Note

For more information on storage and virtualization, refer to the *Fedora Virtualization Deployment and Administration Guide*.

Outils pour la virtualisation

Ce chapitre présente sommairement les différents outils d'aide disponibles pour la virtualisation.

5.1. virsh

virsh is a command line interface (CLI) tool for managing the hypervisor and guest virtual machines. The **virsh** command line tool is built on the **libvirt** management API and operates as an alternative to the **qemu-kvm** command and the graphical **virt-manager** application. The **virsh** command can be used in read-only mode by unprivileged users or, with root access, full administrative functionality. The **virsh** command is ideal for scripting virtualization administration. In addition the **virsh** tool is a main management interface for **virsh** guest domains and can be used to create, pause, and shut down domains, as well as list current domains. This tool is installed as part of the *libvirt-client* package.

Note

Refer to the *Fedora Virtualization Deployment and Administration Guide* for more information about managing virtual machines with **virsh**.

5.2. virt-manager

virt-manager est un outil graphique léger pour gérer les machines virtuelles. Il offre la possibilité de contrôler les machines existantes pendant leur cycle de vie, mettre en place de nouvelles machines, gérer des réseaux virtuels, avoir accès à la console graphique des machines virtuelles et voir les statistiques de performances. Cet outil est inclus dans un paquet qui lui est propre, nommé *virt-manager*.

Note

Refer to the *Fedora Virtualization Deployment and Administration Guide* for more information about managing virtual machines with **virt-manager**.

5.3. virt-install

virt-install est un outil en ligne de commandé pour mettre en place de nouvelles machines virtuelles. Il prend en charge à la fois les installations graphiques ou en mode texte ; il utilise des graphiques en paire client/serveur avec consoles série, SDL, SPICE ou VNC. Le média d'installation peut être local ou distant sur un serveur NFS, HTTP ou FTP. L'outil peut également être configuré pour tourner de manière autonome et démarrer l'invité quand son installation est terminée ; cela permet une automatisation facile de l'installation. Cet outil fait partie du paquet *python-virtinst*.

Note

Refer to the *Fedora Virtualization Deployment and Administration Guide* for more information about **virt-install**.

5.4. guestfish

guestfish est un outil en ligne de commande pour examiner et modifier les systèmes de fichiers de l'hôte. Cet outil se sert du paquet *libguestfs* ; il met à disposition toutes les fonctionnalités de l'API **guestfs**. Cet outil est inclus dans un paquet qui lui est propre, nommé *guestfish*.



Avertissement

L'utilisation de **guestfish** sur des machines virtuelles en marche peut entraîner des détériorations de l'image du disque. Utilisez la commande **guestfish** avec l'option **--ro** (lecture seule) si l'image disque est en cours d'utilisation par une machine virtuelle en fonction.

Note

Refer to the *Fedora Virtualization Deployment and Administration Guide* for more information about **guestfish**.

5.5. GNOME Boxes

GNOME Boxes is a lightweight graphical desktop virtualization tool used to view and access virtual machines and remote systems. It provides a way to test different operating systems and applications from the desktop with minimal configuration. **GNOME Boxes** is based on QEMU and is built into the GNOME 3 desktop in Fedora.

5.6. Autre outils utiles

Les outils suivants s'utilisent pour avoir accès au disque d'une machine virtuelle par l'intermédiaire de l'hôte. Le disque de l'invité est généralement atteint directement via le fichier **disk-image** situé sur l'hôte. Néanmoins, il est quelquefois possible d'y accéder par le domaine de **libvirt**. Les commandes suivantes font partie du domaine de **libvirt** et s'utilisent pour avoir accès à l'image disque de l'invité.

guestmount

Un outil en ligne de commande utilisé pour monter des systèmes de fichiers de machine virtuelle et des images disque sur la machine hôte. Cet outil fait partie du paquet *libguestfs-mount*.



Avertissement

L'utilisation de **guestmount** avec l'option **--r/w** (lecture/écriture) pour avoir accès à un disque en cours d'utilisation par un invité peut entraîner une détérioration du disque. N'utilisez pas **guestmount** avec l'option **--r/w** (lecture/écriture) sur des machines virtuelles à chaud. Utilisez **guestmount** avec l'option **--ro** (lecture seule) si l'image disque est en cours d'utilisation.

virt-cat

Un outil en ligne de commande utilisable pour voir rapidement le contenu d'un ou plusieurs fichiers sur le disque ou l'image disque d'une machine virtuelle donnée. Cet outil fait partie du paquet *libguestfs-tools*.

virt-df

Un outil en ligne de commande utilisé pour voir l'utilisation réelle du disque physique par les machines virtuelles. Semblable à l'outil en ligne de commande **df**. Notez que cet outil ne peut pas fonctionner avec des connexions distantes. Il fait partie du paquet *libguestfs-tools*.

virt-edit

Un outil en ligne de commande utilisé pour modifier des fichiers existants sur une machine virtuelle donnée. Cet outil fait partie du paquet *libguestfs-tools*.



Avertissement

L'utilisation à chaud de **virt-edit** sur des machines virtuelles peut entraîner des détériorations du disque dans la machine virtuelle. Même si la commande **virt-edit** essaie d'empêcher les utilisateurs de modifier à chaud des fichiers de machines virtuelles, il n'est pas garanti qu'elle trappe toutes les tentatives. N'utilisez pas **virt-edit** à chaud sur une machine virtuelle.

virt-filesystems

Un outil en ligne de commande pour découvrir les systèmes de fichiers, les partitions, les volumes logiques et leurs tailles dans une image disque ou une machine virtuelle. Il est couramment utilisé dans les scripts d'interpréteur pour itérer sur tous les fichiers dans une image disque. Cet outil fait partie du paquet *libguestfs-tools*.

Cet outil remplace **virt-list-filesystems** et **virt-list-partitions**.

virt-inspector

Un outil en ligne de commande pouvant examiner une machine virtuelle ou une image disque pour déterminer la version de son système d'exploitation et d'autres informations. Il peut également produire une sortie au format XML, qu'il sera possible de canaliser dans d'autres programmes. Notez que **virt-inspector** ne peut vérifier qu'un seul domaine à la fois. C'est un outil qui fait partie du paquet *libguestfs-tools*.

virt-ls

Un outil en ligne de commande qui liste les fichiers et les répertoires dans une machine virtuelle. Cet outil fait partie du paquet *libguestfs-tools*.

virt-make-fs

Un outil en ligne de commande pour créer un système de fichier à partir d'une archive tar ou de fichiers dans un répertoire. Il est semblable aux outils **mkisofs** et **mksquashfs**, mais il peut créer des types de systèmes de fichiers tel que ext2, ext3 et NTFS, et la taille du système de fichier créé peut être égale ou supérieure à la taille des fichiers à classer. Cet outil fait partie du paquet *libguestfs-tools*.

virt-p2v

A graphical tool to convert physical machines into virtual machines. This tool is installed as part of the *virt-v2v* package.

virt-rescue

Un outil en ligne de commande qui met à disposition un terminal de sauvegarde et quelques simples outils de récupération pour des machines virtuelles ou des images disque impossibles à démarrer. Il peut être lancé sur n'importe quelle machine virtuelle connue de **libvirt** ou directement sur les images disque. Cet outil fait partie du paquet *libguestfs-tools*.



Avertissement

Utiliser **virt-rescue** sur des machines virtuelles en cours de fonctionnement peut entraîner des détériorations des disques de la machine virtuelle. **virt-rescue** essaie de bloquer lui-même son fonctionnement sur les machines virtuelles en fonction, mais il n'est pas possible d'intercepter tous les cas.

L'utilisation de la commande avec l'option **--ro** (lecture seule) n'entraînera pas de détérioration de disque, mais peut donner des résultats, soit étranges, soit incohérents. Il est préférable d'éviter l'utilisation de **virt-rescue** sur une machine virtuelle en fonctionnement.

virt-resize

Un outil en ligne de commande pour redimensionner les disques de machine virtuelle et pour retailler ou détruire toute partition sur un disque de machine virtuelle. Il fonctionne en copiant l'image de l'invité tout en laissant intacte l'image disque originale. Cet outil fait partie du paquet *libguestfs-tools*.



Important

Utiliser **virt-resize** sur des machines virtuelles en fonctionnement peut donner des résultats incongrus. Il est préférable d'arrêter les machines virtuelles avant de tenter un redimensionnement.

virt-top

A command line utility similar to **top**, which shows statistics related to virtualized domains. This tool ships in its own package: *virt-top*.

virt-v2v

Un outil graphique pour faire tourner sous KVM les machines virtuelles initialement sous hyperviseurs Xen ou VMware. Cet outil est incorporé dans un paquet qui lui est propre : *virt-v2v*.

virt-viewer

Un outil minimal pour afficher la console graphique sur une machine virtuelle par l'intermédiaire des protocoles VNC et SPICE. Cet outil est incorporé dans un paquet qui lui est propre : *virt-viewer*.

virt-what

Un script d'interpréteur qui détecte si un programme est en train de s'exécuter sur une machine virtuelle. Cet outil est incorporé dans un paquet qui lui est propre : *virt-what*.

virt-who

Le paquet *virt-who* est un agent hôte de Fedora qui questionne **libvirt** à propos des UUID des invités. Il passe ensuite ces données au serveur local de titrage dans l'optique d'émettre des certificats. Cet outil est incorporé dans un paquet qui lui est propre : *virt-who*.

virt-win-reg

Un outil en ligne de commande pour exporter et collationner les entrées du registre Windows pour un invité Windows et réaliser des opérations simples sur le registre. Cet outils fait partie du paquet *libguestfs-tools*.

**Avertissement**

L'utilisation de **virt-win-reg** sur une machine virtuelle en fonctionnement peut causer une détérioration irréversible du disque de la machine virtuelle. **virt-win-reg** essaie lui-même d'empêcher cette manœuvre sur les machines virtuelles en fonction, mais il ne peut pas détecter tous les cas.

**Avertissement**

Modifier le registre Windows est une opération risquée par nature, étant donné que son format est volontairement obscur et non documenté. Des modifications du registre peuvent rendre le système impossible à démarrer, assurez-vous donc d'avoir une sauvegarde fiable avant d'utiliser l'option **--merge**.

virt-xml-validate

Un outil en ligne de commande pour valider la compatibilité des fichiers XML de **libvirt** avec le schéma publié. Cet outil fait partie du paquet *libvirt-client*.

Annexe A. Historique de modifications

Version 1.0-12 Wednesday June 12, 2013 Dayle Parker daylepark@redhat.com
Publish draft to Fedora docs site.

Version 1.0-11 Monday June 10, 2013 Dayle Parker daylepark@redhat.com
Revised Para-virtualized Devices section based on SME feedback.
Verified references to other Fedora virtualization guides.
Added GNOME Boxes description to Tools.

Version 1.0-10 Thursday May 30, 2013 Dayle Parker daylepark@redhat.com
Added virtio-rng description to Para-virtualized Devices section.

Version 1.0-09 Monday May 27, 2013 Dayle Parker daylepark@redhat.com
Updated CPU Models section based on SME feedback.

Version 1.0-08 Thursday May 9, 2013 Dayle Parker daylepark@redhat.com
Rearranged Migration section and included live storage migration feature description.

Version 1.0-07 Monday May 6, 2013 Dayle Parker daylepark@redhat.com
Added xHCI host controller to Emulated system components list.

Version 1.0-06 Friday May 3, 2013 Dayle Parker daylepark@redhat.com
Made initial general updates for Fedora 19.

Version 1.0-05 Monday October 22, 2012 Dayle Parker daylepark@redhat.com
Branche pour Fedora 18 Beta.

Version 1.0-04 Monday October 22, 2012 Dayle Parker daylepark@redhat.com
Ajout de la description des fonctionnalités de virtio-scsi 4.3.2. Périphériques para-virtualisés.

Version 1.0-03 Thursday September 6, 2012 Dayle Parker daylepark@redhat.com
Dans le chapitre 3 : Avantages, ajout du point sur la flexibilité ([BZ#853826](https://bugzilla.redhat.com/show_bug.cgi?id=853826)¹).

Version 1.0-02 Thursday August 23 2012 Dayle Parker daylepark@redhat.com

¹ https://bugzilla.redhat.com/show_bug.cgi?id=853826

Annexe A. Historique de modifications

Dans Outils : retrait de la mention de virt-inspector2, avertissement sur virt-cat, clarification de l'avertissement --r/w suite à remontée d'information.

Version 1.0-01 Tuesday August 14 2012

Dayle Parker daylepark@redhat.com

Création initiale du livre pour Fedora.