

Fedora 12

Virtualisatie gids

De gezaghebbende gids voor virtualisatie in Fedora



Christoph Curran

Fedora 12 Virtualisatie gids

De gezaghebbende gids voor virtualisatie in Fedora

Uitgave 1

Auteur

Christoph Curran

ccurran@redhat.com

Copyright © 2009 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. 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.

All other trademarks are the property of their respective owners.

De Fedora 12 Virtualisatie gids bevat informatie over installatie, instellen, beheren, tips, trucjes en fout zoeken van de virtualisatie technologie die gebruikt wordt in Fedora 12.

Voorwoord	vii
1. Over dit boek	vii
2. Document Conventie	vii
2.1. Typografische Conventies	vii
2.2. Pull-quote Conventies	ix
2.3. Noten en waarschuwingen	x
3. We hebben terugkoppeling nodig!	x
I. Installation	1
1. Het installeren van de virtualisatie pakketten	3
1.1. KVM installeren tijdens een nieuwe Fedora installatie	3
1.2. KVM pakketten installeren op een bestaand Fedora systeem	5
2. Gevirtualiseerde guest installatie overzicht	7
2.1. Guests aanmaken met virt-install	7
2.2. Guests aanmaken met virt-manager	8
2.3. Guests installeren met PXE	16
3. Guest besturingssysteem installatie procedures	23
3.1. Red Hat Enterprise Linux 5 installeren als een para-gevirtualiseerde guest	23
3.2. Red Hat Enterprise Linux installeren als een volledig gevirtualiseerde guest	65
3.3. Windows XP installeren als een volledig gevirtualiseerde guest	74
3.4. Windows Server 2003 installeren als een volledig gevirtualiseerde guest	92
3.5. Windows Server 2008 installeren als een volledig gevirtualiseerde guest	95
II. Configuration	107
4. Gevirtualiseerde blok apparaten	109
4.1. Een gevirtualiseerde floppy disk controller aanmaken	109
4.2. Opslag apparaten toevoegen aan guests	110
4.3. Blijvende opslag configureren	113
4.4. Voeg een gevirtualiseerde CD-ROM of DVD station toe aan een guest	116
5. Gedeelde opslag en virtualisatie	117
5.1. iSCSI gebruiken voor het opslaan van guests	117
5.2. NFS gebruiken voor het opslaan van guests	117
5.3. GFS2 gebruiken voor het opslaan van guests	117
6. Server beste praktijken	119
7. Beveiliging voor virtualisatie	121
7.1. SELinux en virtualisatie	121
7.2. SELinux overwegingen	122
8. Netwerk configuratie	123
8.1. Netwerk adres vertaling (NAT) met libvirt	123
8.2. Brug netwerken met libvirt	124
9. KVM para-gevirtualiseerde drivers	127
9.1. De KVM Windows para-gevirtualiseerde drivers installeren	127
III. Administration	137
10. Guests beheren met xend	139

11. KVM guest tijds beheer	141
12. KVM live migratie	145
12.1. Live migratie vereisten	145
12.2. Gedeelde opslag voorbeeld: NFS voor een eenvoudige migratie	146
12.3. Live KVM migratie met virsh	147
12.4. Migreren met virt-manager	148
13. Beheer op afstand van gevirtualiseerde guests	159
13.1. Beheer op afstand met SSH	159
13.2. Beheer op afstand met TLS en SSL	160
13.3. Transport modes	161
IV. Virtualisatie referentie gids	167
14. Virtualisatie gereedschappen	169
15. Guests beheren met virsh	173
16. Guests beheren met de Virtual Machine Manager (virt-manager)	183
16.1. Het open connection venster	183
16.2. Het Virtual Machine Manager hoofd scherm	184
16.3. Het Virtual Machine Manager details venster	185
16.4. Virtual Machine grafische console	186
16.5. Starting virt-manager	187
16.6. Een opgeslagen machine herladen	188
16.7. Guest details laten zien	189
16.8. Status volgen	194
16.9. Guest identifiers laten zien	196
16.10. De status van een guest laten zien	197
16.11. Virtuele CPU's laten zien	198
16.12. CPU gebruik laten zien	199
16.13. Geheugen gebruik laten zien	200
16.14. Een virtueel netwerk beheren	201
16.15. Een virtueel netwerk aanmaken	203
V. Tips and Tricks	211
17. Tips en trucjes	213
17.1. Guests automatisch starten	213
17.2. Omschakelen tussen de KVM en Xen hypervisors	213
17.2.1. Xen naar KVM	213
17.2.2. KVM naar Xen	215
17.3. Het gebruik van qemu-img	216
17.4. Overcommit met KVM	218
17.5. Het veranderen van /etc/grub.conf	220
17.6. Virtualisatie uitbreidingen verifiëren	221
17.7. Guest type en implementatie identificeren	222
17.8. Een nieuw uniek MAC adres aanmaken	222
17.9. Heel veilig ftpd	223
17.10. LUN vasthoudendheid instellen	224
17.11. SMART schijf controleren uitzetten	226
17.12. Guest configuratie bestanden klonen	226
17.13. Een bestaande guest dupliceren en zijn configuratie bestand	226

18. Aangepaste libvirt scripts aanmaken	229
18.1. XML configuratie bestanden gebruiken met virsh	229
VI. Troubleshooting	231
19. Troubleshooting	233
19.1. Loop apparaat fouten	233
19.2. Intel VT en AMD-V virtualisatie hardware uitbreidingen aanzetten in de BIOS	233
A. Extra hulpbronnen	235
A.1. On-line hulpbronnen	235
A.2. Geïnstalleerde documentatie	235
B. Herzieningsgeschiedenis	237
C. Colofon	239
Woordenlijst	241

Voorwoord

Dit boek is de Fedora 12 Virtualisatie gids. De gids behandelt alle aspecten van het gebruik en beheer van virtualisatie in Fedora 12.

1. Over dit boek

Dit boek is verdeeld in 7 onderdelen:

- Systeem vereisten
- Installation
- Configuration
- Administration
- Referentie
- Tips and Tricks
- Troubleshooting

2. Document Conventie

Dit handboek hanteert verscheidene conventies om bepaalde woorden of zinsdelen te benadrukken en aandacht te vestigen op specifieke delen van informatie.

In PDF en papieren edities gebruikt dit handboek *Liberation Fonts set*¹ lettertypen. Het Liberation lettertype wordt ook gebruikt in HTML-edities indien dit lettertype op uw computer geïnstalleerd is. Indien dat niet het geval is, worden alternatieve, gelijkwaardige lettertypen gebruikt. Noot: bij Red Hat Enterprise Linux 5 en later wordt de Liberation Font set standaard meegeleverd.

2.1. Typografische Conventies

Vier typografische conventies worden gebruikt om aandacht te vestigen op specifieke woorden en zinsdelen. Deze conventies -en de omstandigheden waaronder zij gebruikt worden- luiden als volgt:

Mono-spaced Bold

Wordt gebruikt om systeem input, waaronder shell commando's, bestandsnamen en paden aan te geven. Wordt ook gebruikt bij toetsaanduiding of toetsencombinaties. Voorbeeld:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

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

Press **Enter** to execute the command.

Press **Ctrl+Alt+F1** to switch to the first virtual terminal. Press **Ctrl+Alt+F7** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **Mono-spaced Bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the `rpm -q package` command. It will return a result as follows: *package-version-release*.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules (MPMs)*. Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

2.2. Pull-quote Conventions

Twee, normaal gesproken uit meerdere regels bestaande, datatypes worden visueel van de omringende tekst gescheiden.

Tekst gezonden naar een terminal wordt getoond in Mono-spaced Roman en als volgt gepresenteerd:

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

Opsommingen van broncode worden ook vertoond in Mono-spaced Roman maar worden als volgt gepresenteerd en benadrukt:

```
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"));
    }
}
```

2.3. Noten en waarschuwingen

Tenslotte gebruiken we drie visuele stijlen om aandacht te vestigen op informatie die anders misschien over het hoofd zou worden gezien.



Noot

Een noot is een tip of handigheidje of een alternatieve benadering voor de taak die uitgevoerd gaat worden. Het negeren van een noot zou geen ernstige gevolgen moeten hebben, maar het leven kan een stuk makkelijker worden indien de noot gevolgd wordt.



Belangrijk

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.



Waarschuwing

Een waarschuwing dient niet genegeerd te worden. Waarschuwingen negeren zal ongetwijfeld leiden tot data- en haarverlies.

3. We hebben terugkoppeling nodig!

Indien u een typografische fout in deze handleiding vindt, of u weet een manier om deze handleiding te verbeteren, zouden wij dat graag van u horen! Meldt u alstublieft fouten in de uitgave **Fedora Documentation** via Bugzilla: <http://bugzilla.redhat.com/bugzilla/>.

Indien u fouten meldt, vergeet dan alstublieft niet het kenmerk: *Virtualization_Guide* te vermelden.

Indien u suggesties hebt om de documentatie te verbeteren, probeer dan zo duidelijk mogelijk deze suggesties te omschrijven. Indien u fouten hebt ontdekt, vermeldt dan alstublieft het sectienummer en wat omringende tekst, opdat we de fout makkelijker kunnen vinden.

Deel I. Installation


Virtualisatie installatie onderwerpen

Deze hoofdstukken beschrijven het instellen van de host en het installeren van gevirtualiseerde gasten in Fedora. Het wordt aanbevolen om deze hoofdstukken zorgvuldig te lezen om een succesvolle installatie van gevirtualiseerde gast besturingssystemen te verzekeren.

Het installeren van de virtualisatie pakketten

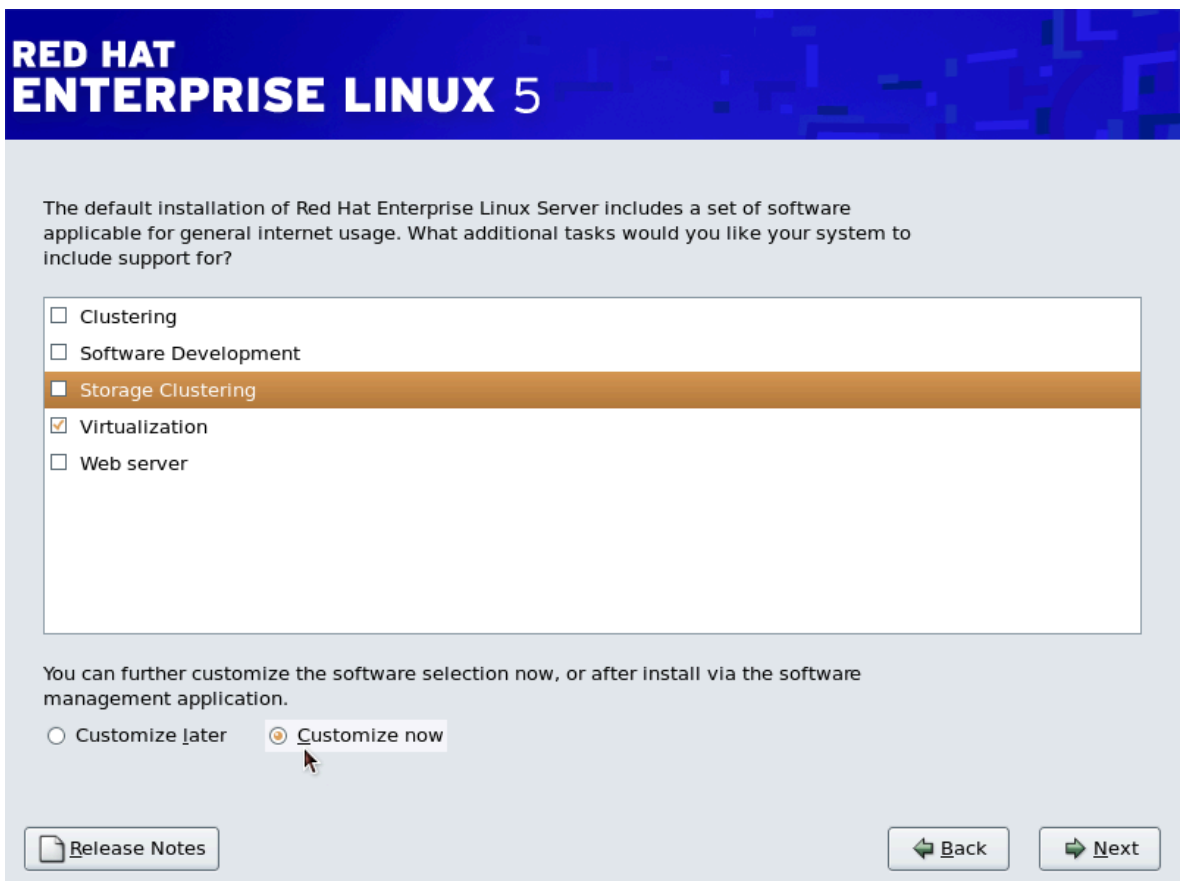
1.1. KVM installeren tijdens een nieuwe Fedora installatie

Deze paragraaf behandelt het installeren van virtualisatie gereedschappen en het KVM pakket als onderdeel van een nieuwe Fedora 12 installatie.



Hulp nodig bij het installeren?
De *Fedora 12 Installatie gids* (beschikbaar op <http://docs.fedoraproject.org>) behandelt het installeren van Fedora 12 tot in alle details.

1. Begin een interactieve Fedora installatie met de Fedora 12 installatie CD-ROM, DVD of PXE.
2. Maak de andere stappen af tot de pakket selectie stap.



**RED HAT
ENTERPRISE LINUX 5**

The default installation of Red Hat Enterprise Linux Server includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?

- Clustering
- Software Development
- Storage Clustering
- Virtualization
- Web server

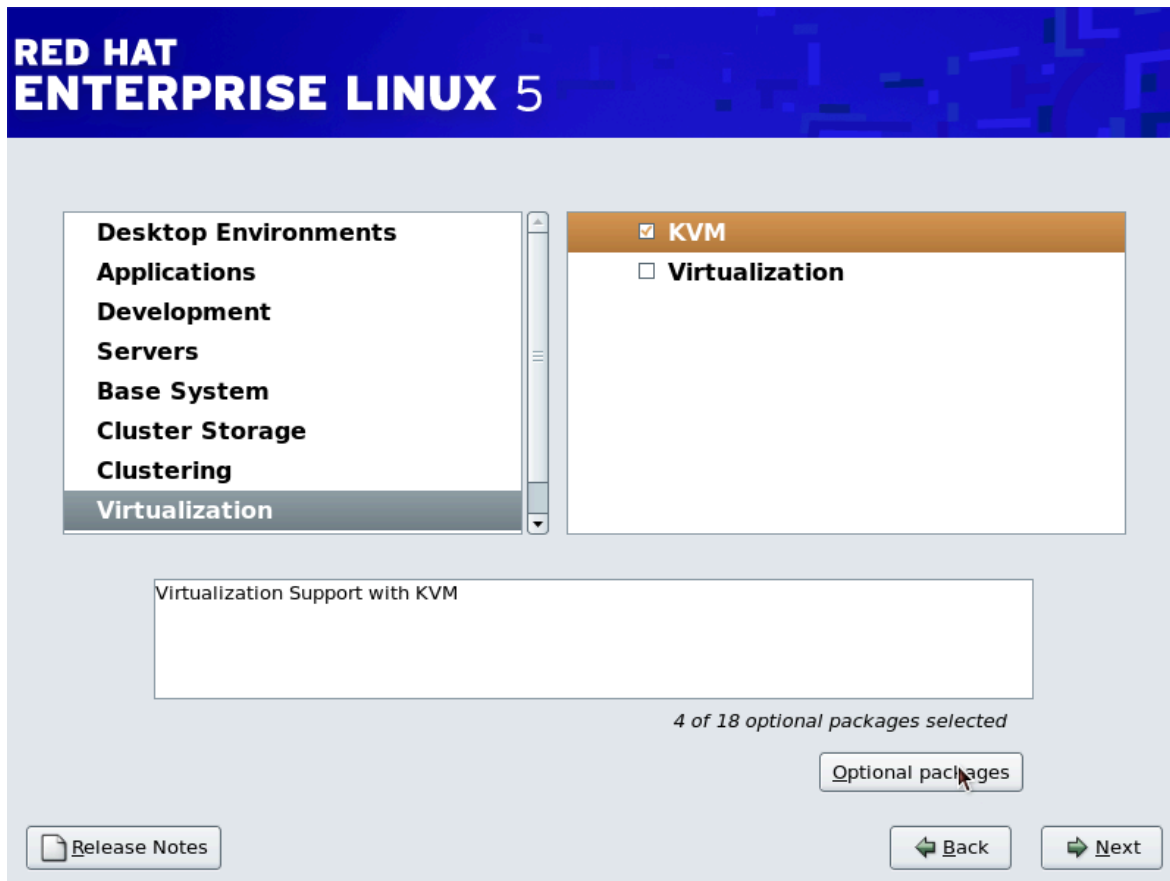
You can further customize the software selection now, or after install via the software management application.

Customize later Customize now

[Release Notes](#) [Back](#) [Next](#)

Selecteer de **Virtualisatie** pakketgroep en de **Nu aanpassen** knop.

3. Selecteer de **KVM** pakketgroep. De-selecteer de **Virtualisatie** pakketgroep. Dit selecteert de KVM hypervisor, **virt-manager**, **libvirt** en **virt-viewer** voor installatie.



4. **Pas de pakketten aan (indien nodig)**

Pas de **Virtualisatie** groep aan als je andere virtualisatie pakketten nodig hebt.



Klik op **Sluiten**, gevolgd door **Volgende** om verder te gaan met de installatie.

KVM pakketten installeren met Kickstart bestanden

Deze paragraaf beschrijft hoe je een Kickstart bestand kunt gebruiken om Fedora te installeren met de KVM hypervisor pakketten. Kickstart bestanden laten grote, automatische installaties toe zonder dat een gebruiker ieder individueel systeem handmatig moet installeren. De stappen in deze paragraaf zullen je helpen een Kickstart bestand te maken en te gebruiken om Fedora te installeren met de virtualisatie pakketten.

In de %packages sectie van je Kickstart bestand, voeg je de volgende pakketgroep toe:

```
%packages
@kvm
```

Meer informatie over Kickstart bestanden kan gevonden worden op de Fedora Project website, <http://docs.fedoraproject.org>, in de *Fedora 12 Installatie gids*.

1.2. KVM pakketten installeren op een bestaand Fedora systeem

Deze paragraaf beschrijft de stappen die nodig voor het installeren van de KVM hypervisor op een werkende Fedora 12 of nieuwer.

De KVM hypervisor installeren met yum

Om virtualisatie te gebruiken op Fedora heb je het **kvm** pakket nodig. Het **kvm** pakket bevat de KVM kernel module die de KVM hypervisor aanbiedt voor de standaard Linux kernel.

Om het **kvm** pakket te installeren, voer je uit:

```
# yum install kvm
```

Installeer nu extra virtualisatie beheer pakketten.

[Aanbevolen virtualisatie pakketten:](#)

python-virtinst

Levert het **virt-install** commando voor het aanmaken van virtuele machines.

libvirt

libvirt is een API bibliotheek voor interactie met hypervisors. **libvirt** gebruikt het **xm** virtualisatie raamwerk en het **virsh** commando-regel gereedschap om virtuele machines te beheren en te controleren.

libvirt-python

Het **libvirt-python** pakket bevat een module die toestaat dat toepassingen geschreven in de Python programmeertaal de interface gebruiken die geleverd wordt door de **libvirt** API.

virt-manager

virt-manager, ook bekend als **Virtual Machine Manager**, biedt een grafisch gereedschap voor het beheren van virtuele machines. Het gebruikt de **libvirt** bibliotheek voor de beheer API.

Installeer de andere aanbevolen virtualisatie pakketten:

```
# yum install virt-manager libvirt libvirt-python python-virtinst
```

Gevirtualiseerde guest installatie

overzicht

Nadat je de virtualisatie pakketten op het host systeem geïnstalleerd hebt, kun je guest besturingssystemen aanmaken. Dit hoofdstuk beschrijft het algemene proces voor het installeren van guest besturingssystemen op virtuele machines. Je kunt guests aanmaken met gebruik van de **New** knop in **virt-manager** of je kunt de commando-regel interface **virt-install** gebruiken. Beide manieren worden in dit hoofdstuk behandeld.

Gedetailleerde instructies zijn beschikbaar voor specifieke versies van Fedora, andere Linux distributies, Solaris en Windows. Refereer naar [Hoofdstuk 3, Guest besturingssysteem installatie procedures](#) voor deze procedures.

2.1. Guests aanmaken met virt-install

Je kunt het **virt-install** commando gebruiken om gevirtualiseerde guests aan te maken op de commando-regel. **virt-install** wordt of interactief gebruikt, of als onderdeel van een script om het aanmaken van virtuele machines te automatiseren. Gebruik van **virt-install** met Kickstart staat niet begeleide installatie van virtuele machines toe.

Het **virt-install** gereedschap biedt een aantal opties die doorgegeven kunnen worden op de commando-regel. Om een complete lijst van opties te zien, voer je uit:

```
$ virt-install --help
```

De **virt-install** manual pagina beschrijft ook iedere commando optie en belangrijke variabelen.

gemu-img is een gerelateerd commando welke voor **virt-install** gebruikt kan worden om opslag opties in te stellen.

Een belangrijke optie is de `--vnc` optie welke een grafisch venster opent voor de installatie van de guest.

Dit voorbeeld maakt een Red Hat Enterprise Linux 3 guest, met de naam *rhel3support*, vanaf een CD-ROM, met virtueel netwerk en een 5 GB bestand-gebaseerd blok apparaat image. Dit voorbeeld gebruikt de KVM hypervisor.

```
# virt-install --accelerate --hvm --connect gemu:///system \  
  --network network:default \  
  --name rhel3support --ram=756\  
  --file=/var/lib/libvirt/images/rhel3support.img \  
  --file-size=6 --vnc --cdrom=/dev/sr0
```

Voorbeeld 2.1. Het gebruik van `virt-install` met KVM om een Red Hat Enterprise Linux 3 guest aan te maken

```
# virt-install --name Fedora11 --ram 512 --file=/var/lib/libvirt/images/  
Fedora11.img \  
  --file-size=3 --vnc --cdrom=/var/lib/libvirt/images/Fedora11.iso
```

Voorbeeld 2.2. Het gebruik van `virt-install` om een Fedora 11 guest aan te maken

2.2. Guests aanmaken met virt-manager

virt-manager, ook bekend als Virtual Machine Manager, is een grafisch gereedschap voor het aanmaken en beheren van gevirtualiseerde guests.

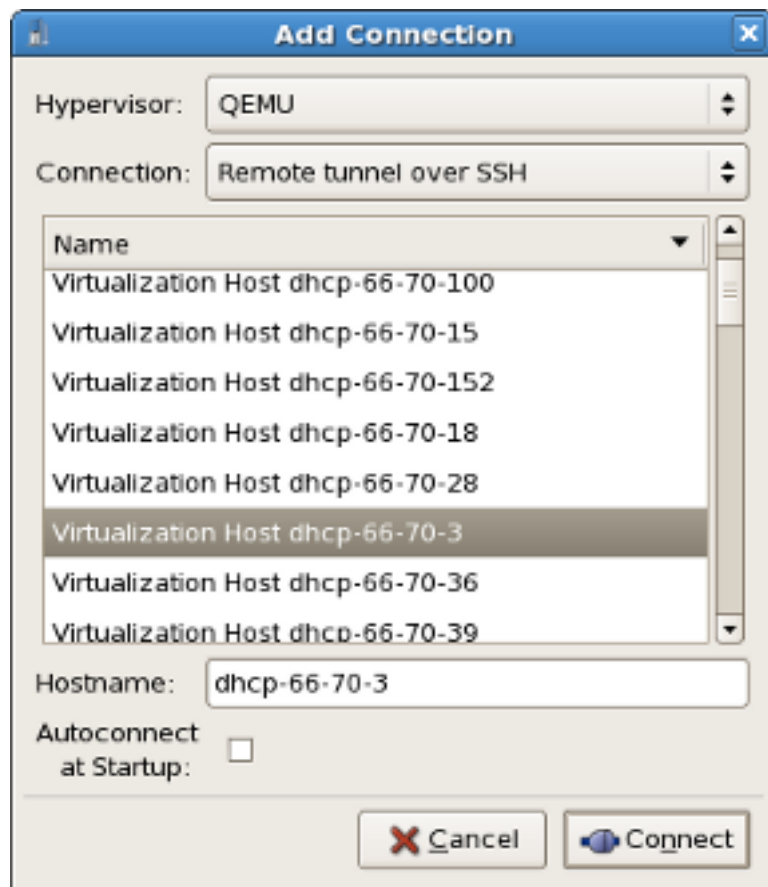
Procedure 2.1. Het aanmaken van een gevirtualiseerde guest met virt-manager

1. Om **virt-manager** op te starten voer je het volgende commando uit als root:

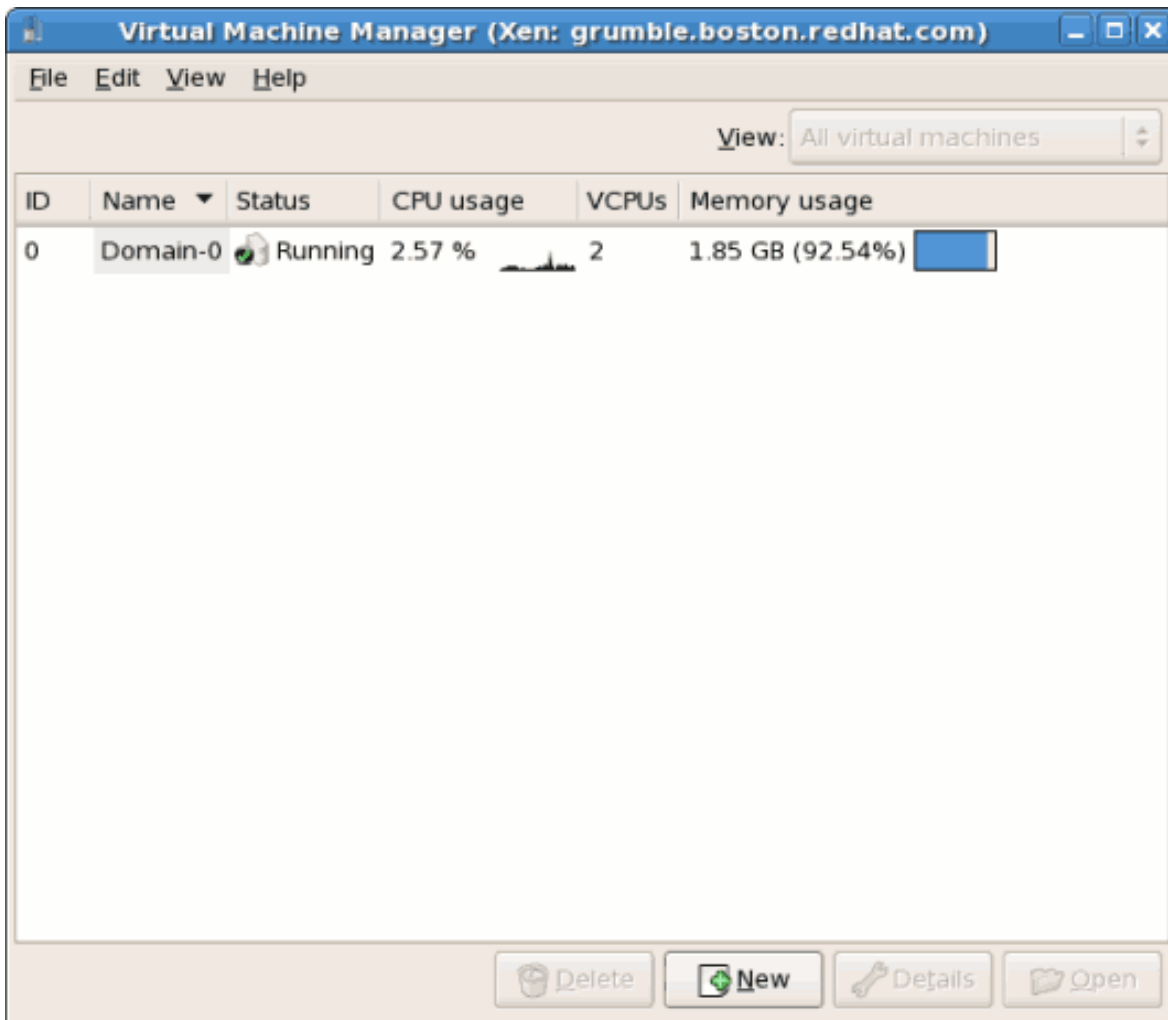
```
# virt-manager &
```

Het **virt-manager** commando opent een grafisch gebruiker interface scherm. Verscheidene functies zijn niet beschikbaar voor gebruikers zonder root rechten of ingesteld met **sudo**, waaronder de **New** knop en je zult niet in staat zijn om een nieuwe gevirtualiseerde guest aan te maken.

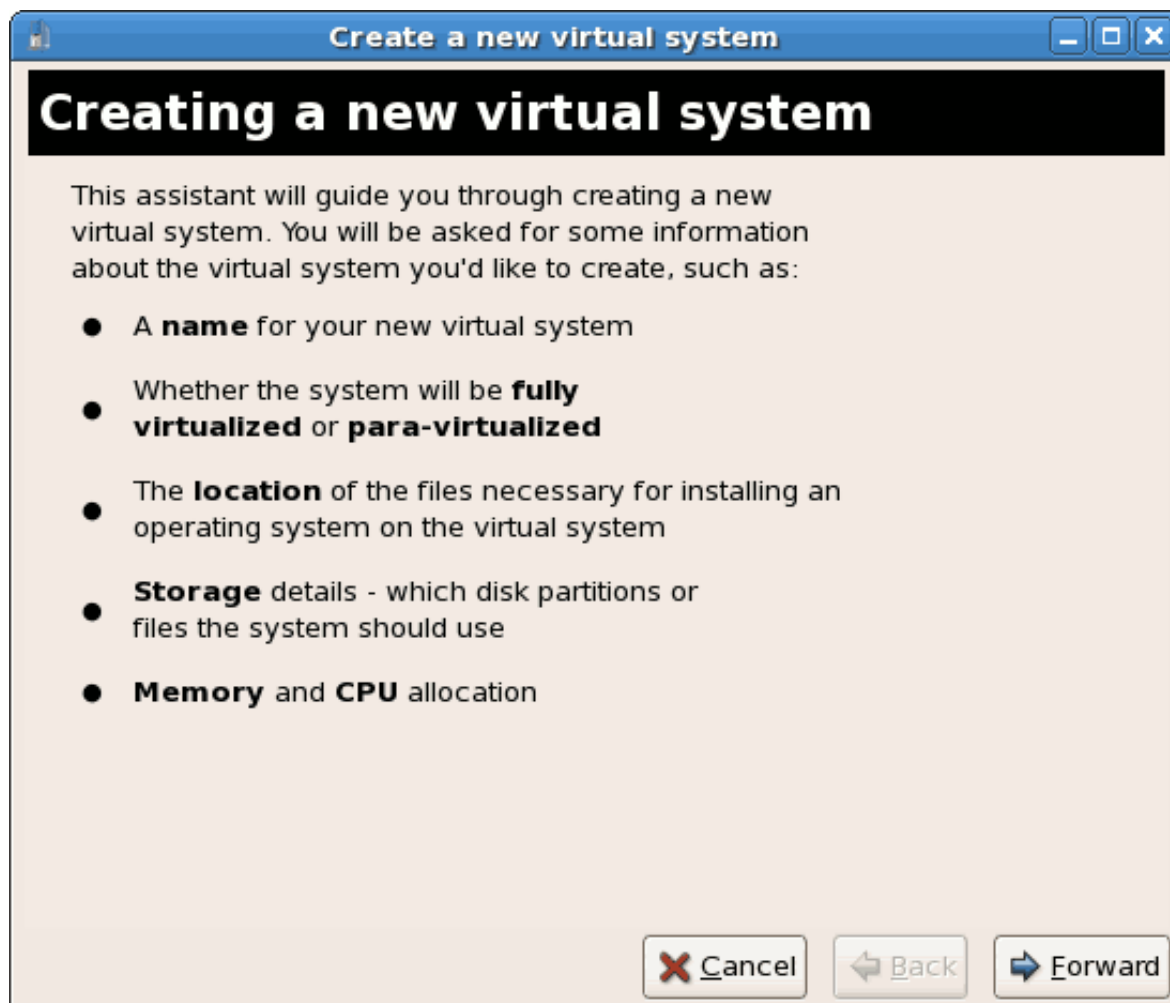
2. Klik op **File -> Open Connection**. Het dialoog venster hieronder getoond verschijnt. Selecteer een hypervisor en klik op de **Connect** knop:



3. Het **virt-manager** venster laat je een nieuwe virtuele machine aanmaken. Klik op de **New** knop om een nieuwe guest aan te maken. Dit opent instel hulp getoond in de schermafdruck.



4. Het **Create a new virtual system** venster biedt een samenvatting van de informatie die je moet opgeven om een virtuele machine aan te maken:

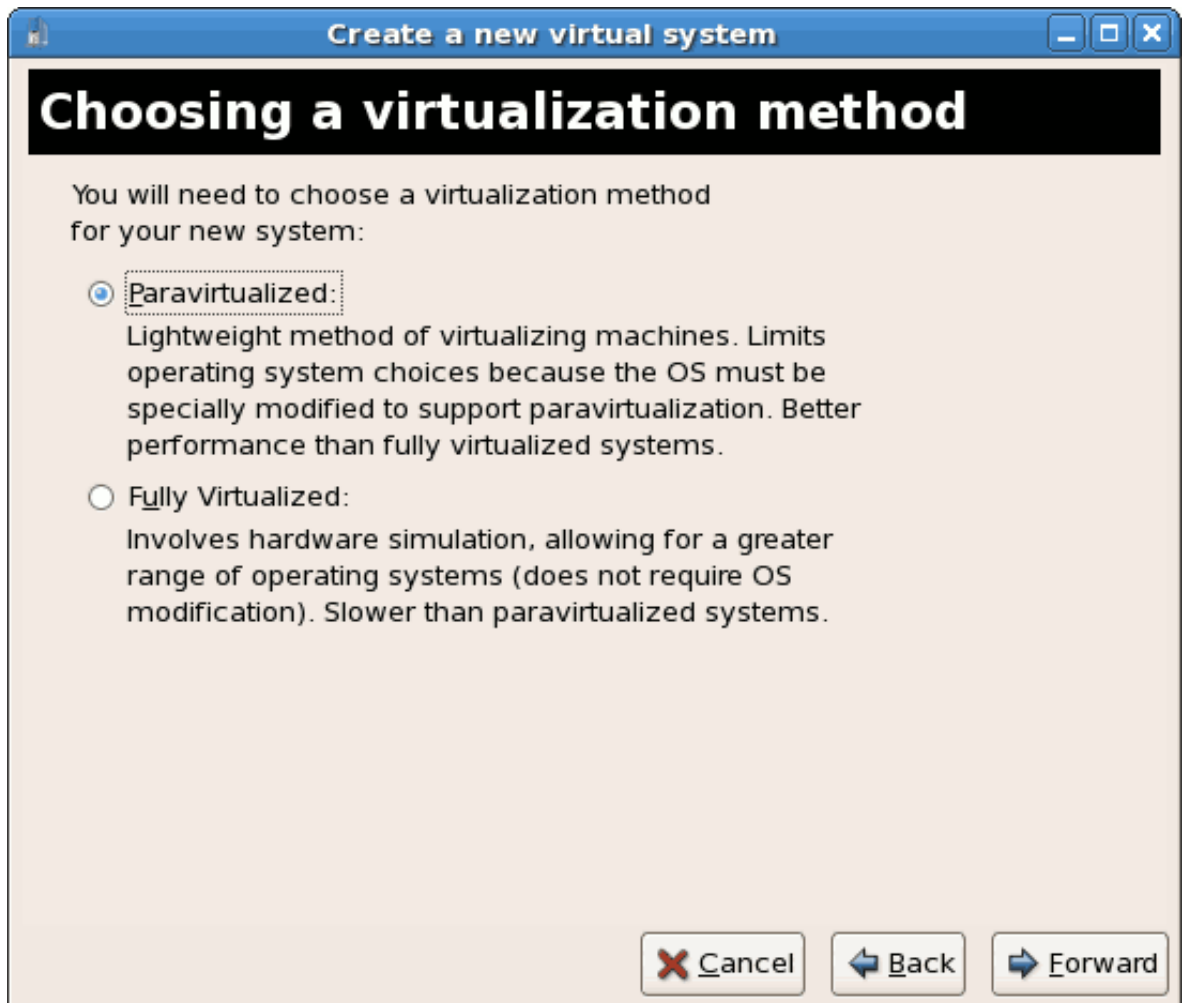


Bekijk de informatie voor jouw installatie en klik op de **Forward** knop.

5. Het **Choosing a virtualization method** venster verschijnt. Kies tussen **Para-virtualized** of **Fully virtualized**.

Volledige virtualisatie vereist een systeem met een Intel® VT of AMD-V processor. Als de virtualisatie uitbreidingen niet aanwezig zijn, zal de **fully virtualized** knop of **Enable kernel/hardware acceleration** niet selecteerbaar zijn. De **Para-virtualized** optie zal grijs zijn als de **kernel-xen** niet de kernel is die op dit moment draait.

Als je verbond met een KVM hypervisor is alleen volledige virtualisatie beschikbaar.

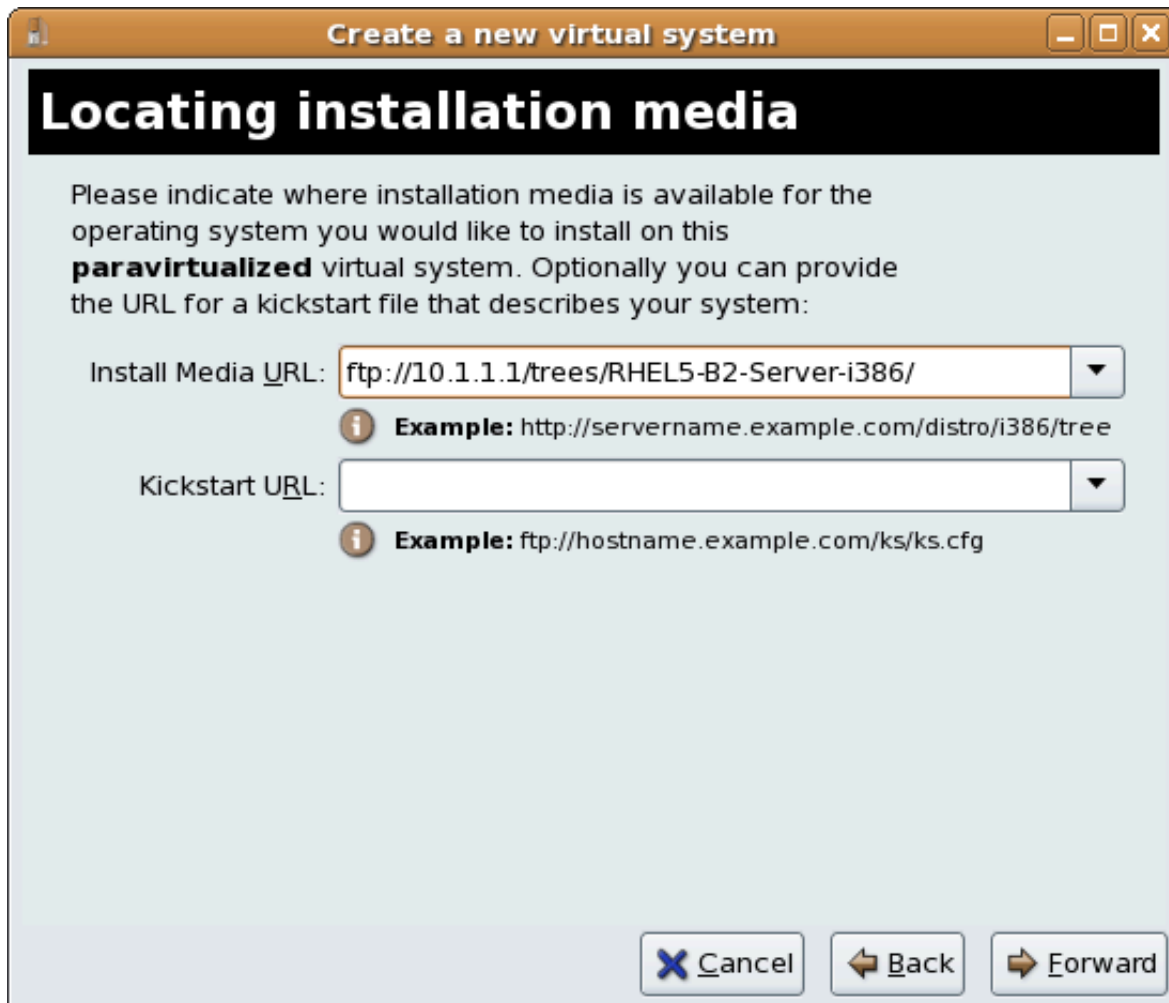


Kies het virtualisatie type en klik op de **Next** knop.

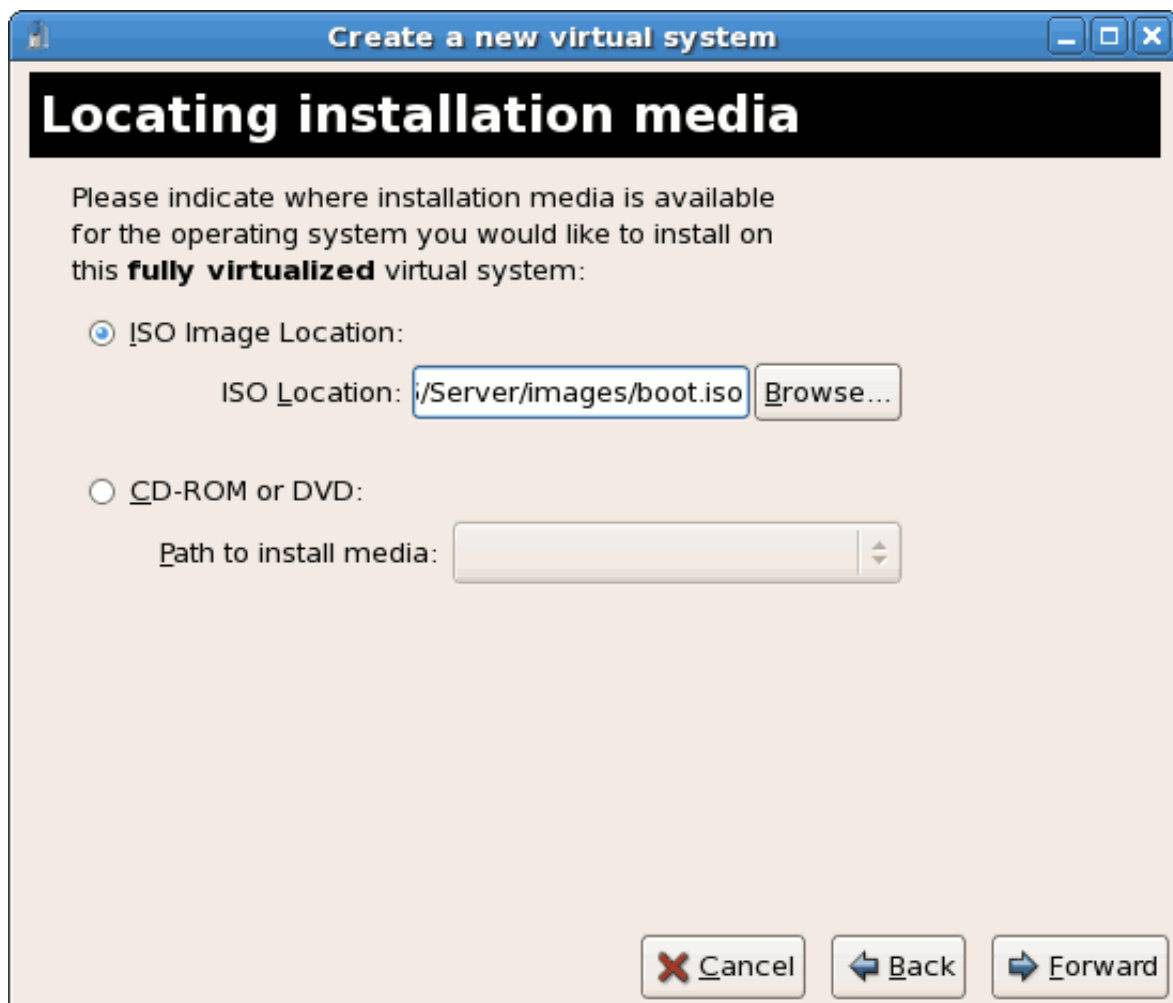
6. De **Locating installation media** prompt vraagt om de installatie media voor het type installatie die je selecteerde. Dit scherm is afhankelijk van de selectie in de vorige stap.
 - a. De para-gevirtualiseerde installatie vereist een installatie boom die toegankelijk is met een van de volgende netwerk protocollen: HTTP, FTP of NFS. De installatie media URL moet een Fedora installatie boom bevatten. Deze boom wordt geschikt gesteld met gebruik van NFS, FTP of HTTP. De netwerk services en bestanden kunnen beschikbaar gesteld worden met gebruik van netwerk services op de host of een andere spiegel.

Met gebruik van een CD-ROM of DVD image (gemarkt als een **.iso** bestand), koppel je de CD-ROM image aan en je maakt de aangekoppelde bestanden beschikbaar met een van de genoemde protocollen.

Als alternatief kopieer je de installatie boom van een Fedora spiegel.



- b. Een volledig gevirtualiseerde guest installatie vereist lokaal opstartbare installatie DVD's, CD-ROM's of images van opstartbare DVD's of CD-ROM's (met het .iso of .img bestandstype). Windows installaties gebruiken DVD, CD-ROM of iso bestand. Vele Linux en UNIX-achtige besturingssystemen gebruiken een .iso bestand om een basis systeem te installeren voordat de installatie afgemaakt wordt met een netwerk-gebaseerde installatie boom.



Na het selecteren van de juiste installatie media, klik je op de **Forward** knop.

7. The **Assigning storage space** window displays. Choose a disk partition, LUN or create a file based image for the guest storage.

De conventie voor op bestand gebaseerde images in Fedora is dat alle bestand gebaseerde guest images zich bevinden in de `/var/lib/xen/images/` map. Andere map locaties voor op bestand gebaseerde images zijn verboden door SELinux. Als je SELinux draait in de afdwingende mode, refereer je naar [Paragraaf 7.1, "SELinux en virtualisatie"](#) voor meer informatie over het installeren van guests.

Your guest storage image should be larger than the size of the installation, any additional packages and applications, and the size of the guests swap file. The installation process will choose the size of the guest's swap file based on size of the RAM allocated to the guest.

Allocate extra space if the guest needs additional space for applications or other data. For example, web servers require additional space for log files.



Choose the appropriate size for the guest on your selected storage type and click the **Forward** button.

Opmerking

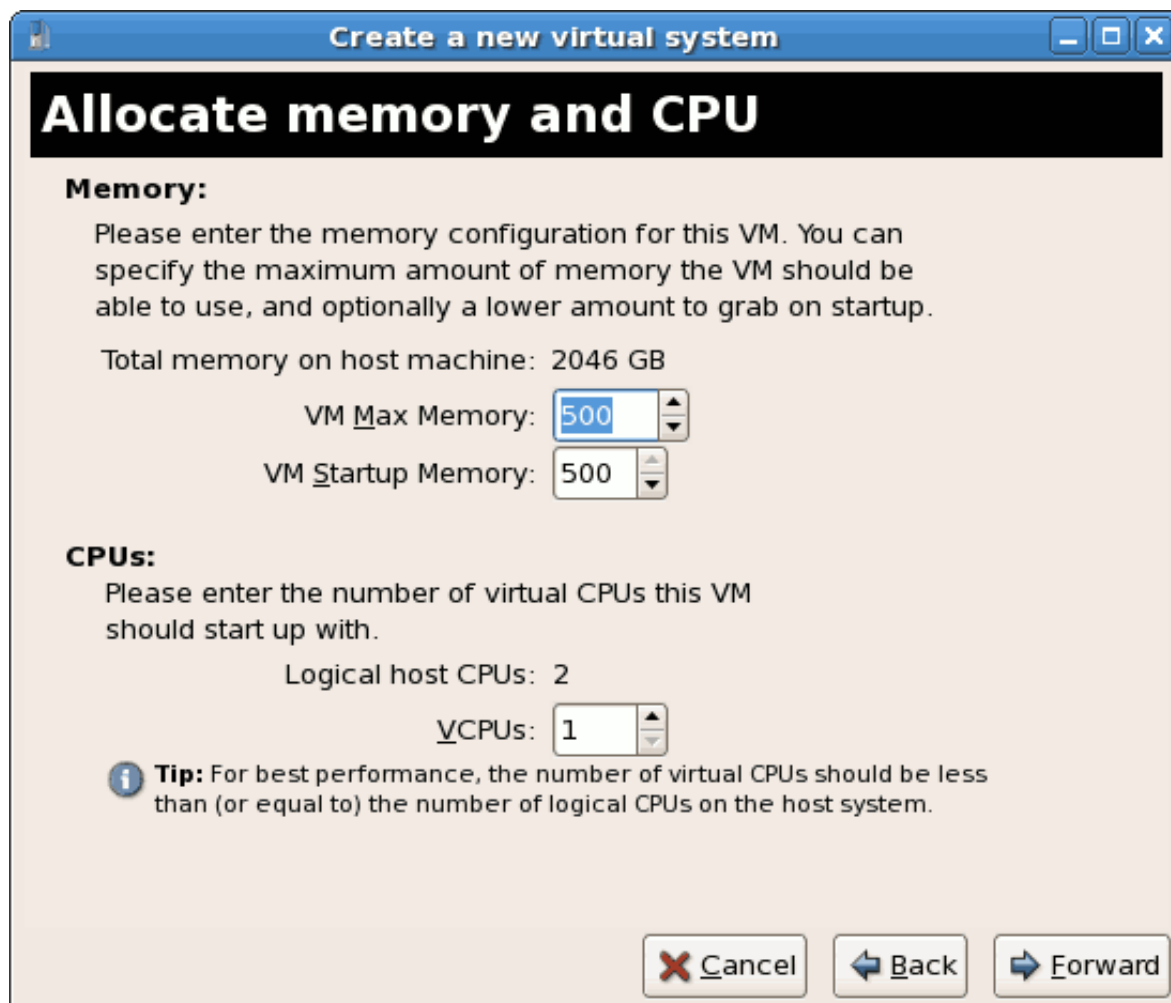
Het wordt aanbevolen dat je de standaard map gebruikt voor virtuele machine images, `/var/lib/xen/images/`. Als je een andere locatie gebruikt (zoals `/xen/images/` in dit voorbeeld) wees er dan zeker van dat het wordt toegevoegd aan jouw SELinux richtlijnen en dat het opnieuw gelabeld wordt voordat je verder gaat met de installatie (later in dit document zul je informatie vinden over het veranderen van jouw SELinux richtlijnen).

8. The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

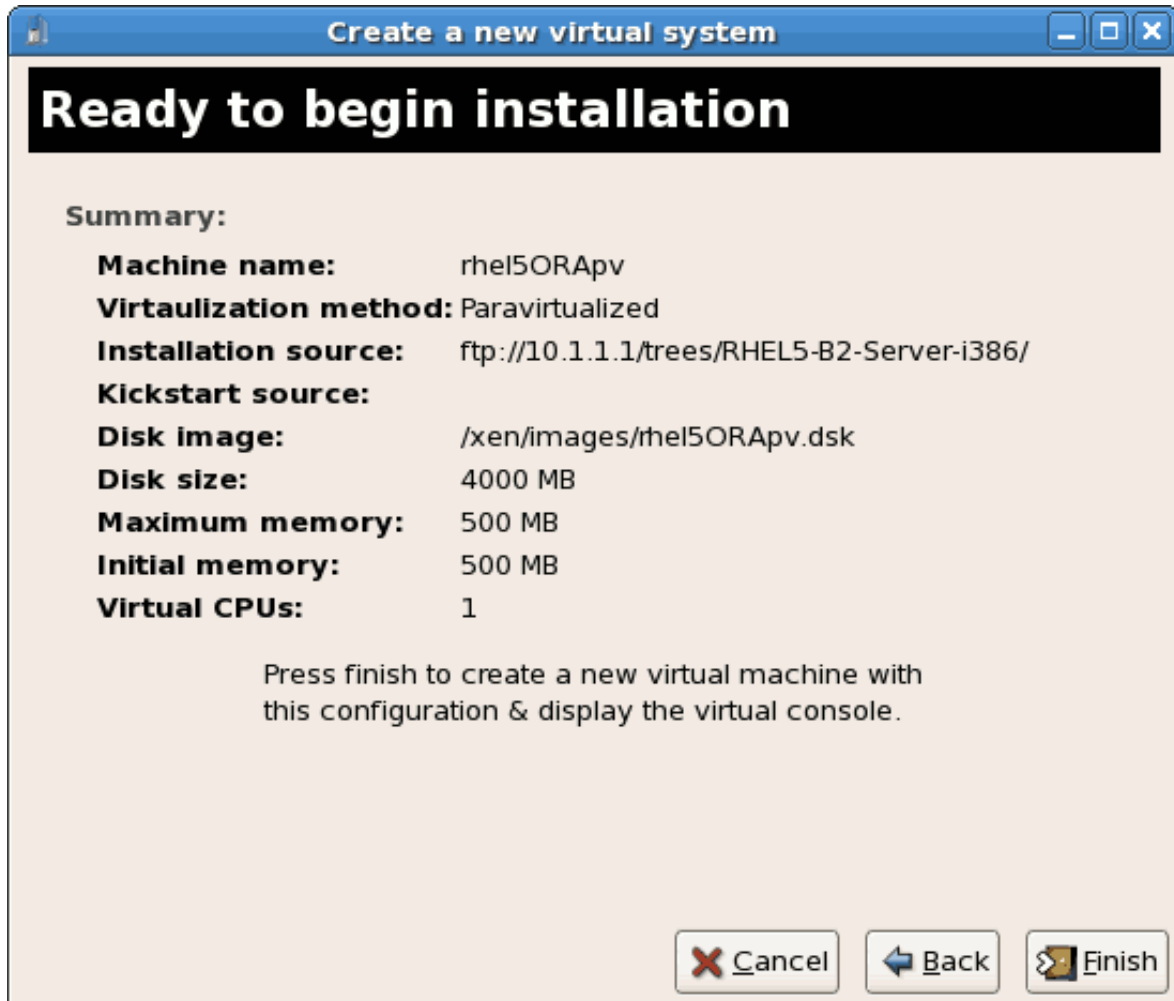
Guests vereisen voldoende fysiek geheugen (RAM) om efficiënt en effectief te draaien. Kies een geheugen waarde die past bij je guest besturingssysteem en toepassing behoeften. De meeste besturingssystemen vereisen ten minste 512 MB RAM om voldoende snel te werken. Denk eraan dat guests fysiek RAM gebruiken. Het draaien van te veel guests of het overlaten van onvoldoende geheugen voor het host systeem resulteert in een significant gebruik van virtueel geheugen. Virtueel geheugen is significant langzamer wat degradatie van de systeem prestatie

en snelheid veroorzaakt. Wees er zeker van om voldoende geheugen toe te kennen voor alle guests en de host om effectief te kunnen werken.

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



9. Het "Ready to begin installation" venster laat een overzicht zien van alle configuratie informatie die je hebt opgegeven. Bekijk de gepresenteerde informatie en gebruik de **Back** knop om veranderingen te maken als dat nodig is. Als je tevreden bent klik je op de **Finish** knop om het installatie proces te beginnen.



Een VNC venster opent die het begin van het guest besturingssysteem installatie proces laat zien.

Dit beëindigt het algemene proces voor het aanmaken van guests met **virt-manager**. [Hoofdstuk 3, Guest besturingssysteem installatie procedures](#) bevat stap-voor-stap instructies voor het installeren van een aantal veel voorkomende besturingssystemen.

2.3. Guests installeren met PXE

Deze paragraaf behandelt de stappen die nodig zijn om guests te installeren met PXE. PXE guest installatie vereist een gedeeld netwerk apparaat, ook bekend als een brug. De procedure hier beneden behandelt het maken van een brug en de stappen die nodig zijn om de brug te gebruiken voor een PXE installatie.

1. **Maak een nieuwe brug**
 - a. Maak een nieuw netwerk script bestand in de `/etc/sysconfig/network-scripts/` map. Dit voorbeeld maakt een bestand met de naam `ifcfg-installation` welke een brug maakt met de naam `installation`.

```
# cd /etc/sysconfig/network-scripts/  
# vim ifcfg-installation
```

```
DEVICE=installation
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
```



Warning

The line, `TYPE=Bridge`, is case-sensitive. It must have uppercase 'B' and lower case 'ridge'.

- b. Start de nieuwe brug

```
# ifup installation
```

- c. Er zijn nog geen interfaces toegevoegd aan de nieuwe brug. Gebruik het **brctl show** commando, om details te bekijken van de netwerk bruggen op het systeem.

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
installation     8000.00000000000000     no
virbr0           8000.00000000000000     yes
```

De **virbr0** brug is de standaard brug die gebruikt wordt door **libvirt** voor netwerk adres vertaling (NAT) op het standaard Ethernet apparaat.

2. Voeg een interface toe aan de nieuwe brug

Bewerk het configuratie bestand voor de interface. Voeg de **BRIDGE** parameter toe aan het configuratie bestand met de naam van de brug die in de vorige stappen aangemaakt werd.

```
# Intel Corporation Gigabit Network Connection
DEVICE=eth1
BRIDGE=installation
BOOTPROTO=dhcp
HWADDR=00:13:20:F7:6E:8E
ONBOOT=yes
```

Na het bewerken van het configuratie bestand, start je het netwerk opnieuw op of je herstart de computer.

```
# service network restart
```

Controleer of de interface aangebracht is met het **brctl show** commando:

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
installation     8000.001320f76e8e       no              eth1
virbr0           8000.00000000000000     yes
```

3. Beveiliging configuratie

Configure **iptables** to allow all traffic to be forwarded across the bridge.

```
# iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
# service iptables save
# service iptables restart
```



Disable iptables on bridges

Alternatively, prevent bridged traffic from being processed by **iptables** rules. In `/etc/sysctl.conf` append the following lines:

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Reload the kernel parameters configured with **sysctl**

```
# sysctl -p /etc/sysctl.conf
```

4. Herstart libvirt voor de installatie

Restart the **libvirt** daemon.

```
# service libvirtd reload
```

De brug is ingesteld, je kunt nu beginnen met een installatie.

PXE installatie met virt-install

Aan **virt-install** voeg je de `--network=bridge:BRIDGENAME` installatie parameter toe waarin installatie de naam van jouw brug is. Gebruik voor PXE installaties de `--pxe` parameter.

```
# virt-install --accelerate --hvm --connect qemu:///system \
  --network=bridge:installation --pxe \
  --name EL10 --ram=756 \
  --vcpus=4
  --os-type=linux --os-variant=rhel5
  --file=/var/lib/libvirt/images/EL10.img \
```

Voorbeeld 2.3. PXE installatie met virt-install

PXE installatie met virt-manager

De stappen hier beneden zijn de stappen die anders zijn dan die in de standaard virt-manager installatie procedure. Voor de standaard installaties refereer je naar [Hoofdstuk 3, Guest besturingssysteem installatie procedures](#).

1. Selecteer PXE

Selecteer PXE als de installatie methode.



The screenshot shows a window titled "Create a new virtual machine" with a sub-header "Installation Method". The window contains the following text and controls:

Please indicate where installation media is available for the operating system you would like to install on this virtual machine:

- Local install media (ISO image or CDROM)
- Network install tree (HTTP, FTP, or NFS)
- Network boot (PXE)

Please choose the operating system you will be installing on the virtual machine:

OS Type: Linux

OS Variant: Red Hat Enterprise Linux 5

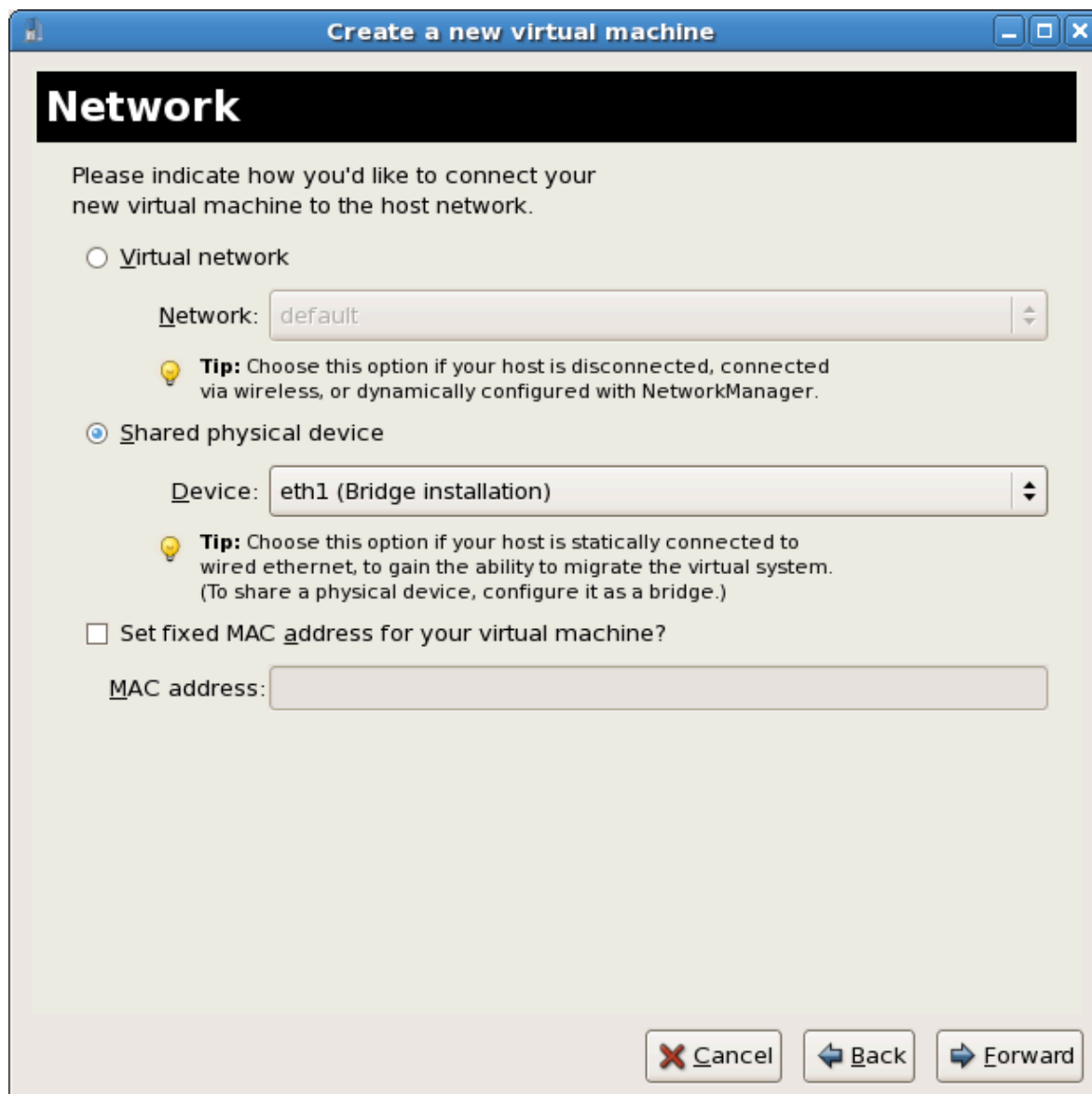
⚠ Not all operating system choices are supported by Red Hat. Please see the link below for supported configurations:

[Red Hat Enterprise Linux 5 virtualization support](#)

Buttons: Cancel, Back, Forward

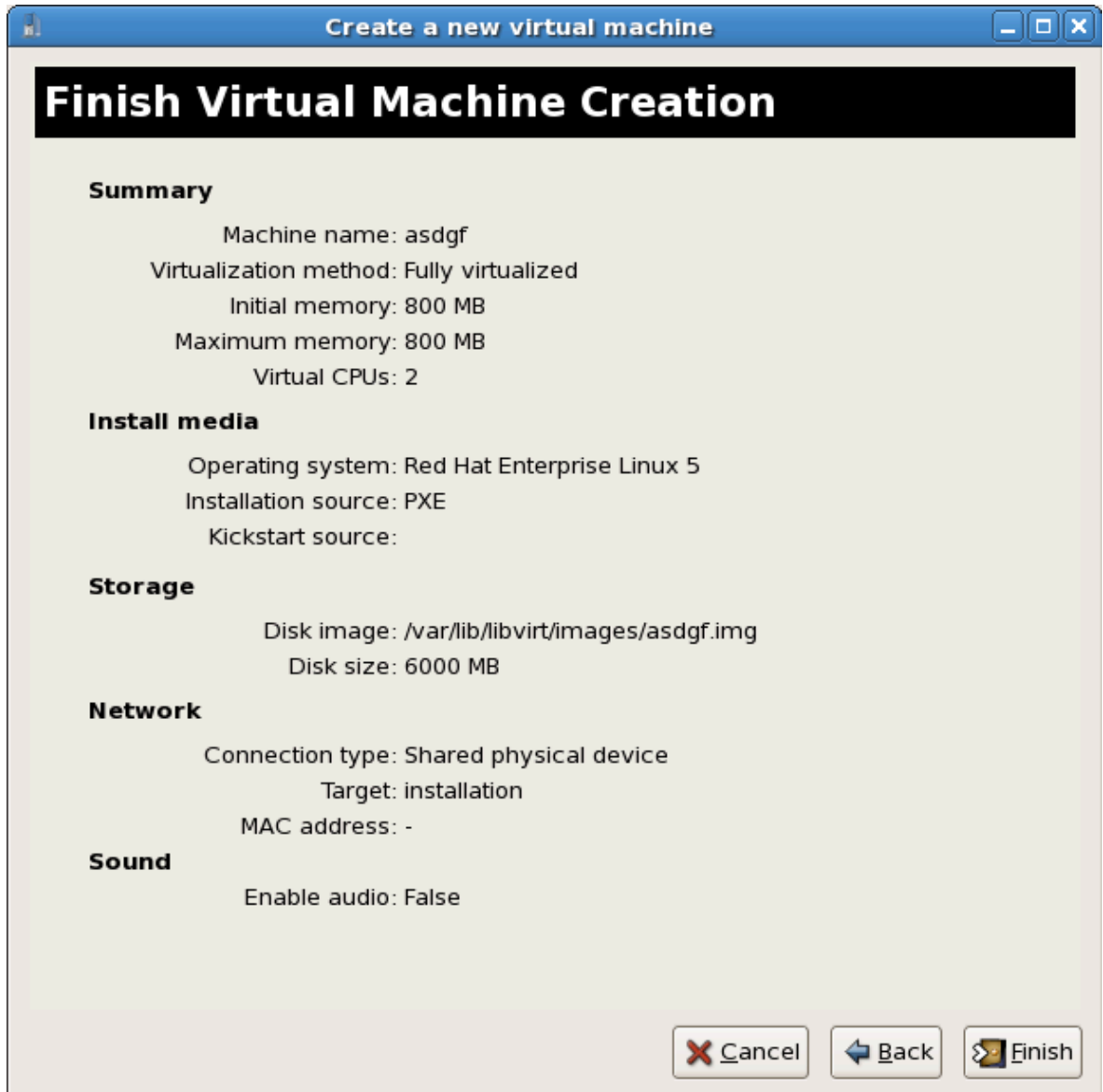
2. **Selecteer de brug**

Selecteer **Shared physical device** en selecteer de brug die in de vorige procedure gemaakt is.



3. **Start de installatie**

De installatie is klaar om te beginnen.



Een DHCP verzoek wordt verstuurd en als een geldige PXE server wordt gevonden, zullen de guest installatie processen beginnen.

Guest besturingssysteem installatie procedures

Dit hoofdstuk behandelt hoe je verschillende guest besturingssystemen installeert in een gevirtualiseerde omgeving in Fedora. Om de basis processen te begrijpen, refereer je naar [Hoofdstuk 2, Gevirtualiseerde guest installatie overzicht](#).

3.1. Red Hat Enterprise Linux 5 installeren als een para-gevirtualiseerde guest

Deze paragraaf beschrijft hoe je Red Hat Enterprise Linux 5 installeert als een para-gevirtualiseerde guest. Para-virtualisatie is sneller dan volledige virtualisatie en ondersteunt alle voordelen van volledige virtualisatie. Para-virtualisatie vereist een speciale, ondersteunde kernel, de **kernel-xen** kernel.



Belangrijke opmerking over para-virtualisatie

Para-virtualisatie werkt alleen met de Xen hypervisor. Para-virtualisatie werkt niet met de KVM hypervisor.

Verzeker je ervan dat je root toegang hebt voordat je met de installatie begint.

Deze methode installeert Red Hat Enterprise Linux van een server op afstand. De installatie instructies die hier getoond worden zijn vergelijkbaar met die van een minimale installatie van een live CD-ROM.

Maak para-gevirtualiseerde Red Hat Enterprise Linux 5 guests aan met gebruik van `virt-manager` of `virt-install`. Voor instructies over **virt-manager**, refereer je naar de procedure in [Paragraaf 2.2, "Guests aanmaken met virt-manager"](#).

Maak een para-gevirtualiseerde guest aan met het commando-regel gereedschap **virt-install**. De `--vnc` optie laat de grafische installatie zien. De naam van de guest is in dit voorbeeld `rhe15PV`, het schijf image bestand is `rhe15PV.dsk` en een locale spiegel van de Red Hat Enterprise Linux 5 installatie boom is `ftp://10.1.1.1/trees/CentOS5-B2-Server-i386/`. Vervang deze waarden met waarden die geldig zijn voor jouw systeem en netwerk.

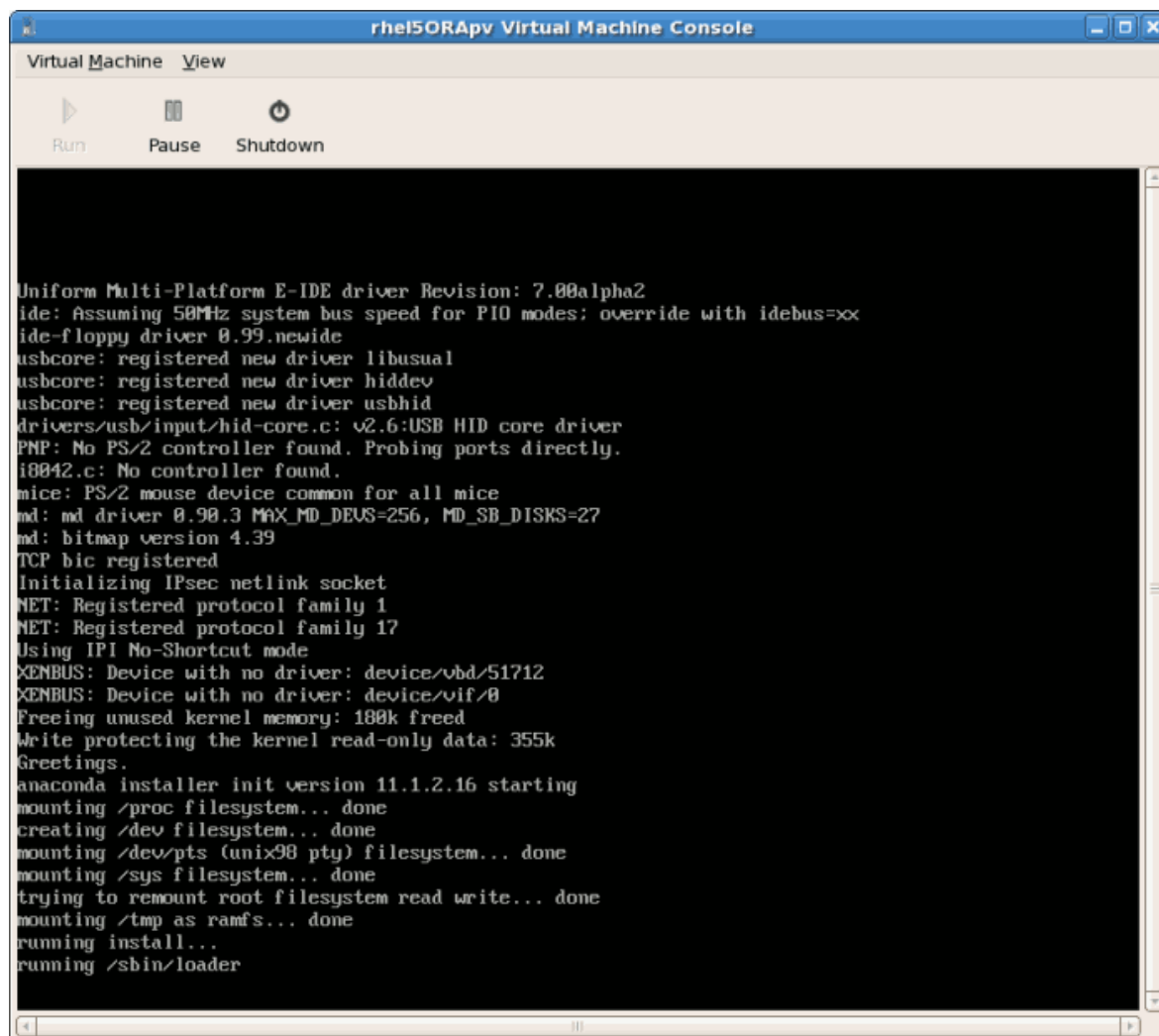
```
# virt-install -n rhe15PV -r 500 \  
-f /var/lib/libvirt/images/rhe15PV.dsk -s 3 --vnc -p \  
-l ftp://10.1.1.1/trees/CentOS5-B2-Server-i386/
```



Automatische installatie

Red Hat Enterprise Linux kan geïnstalleerd worden zonder een grafische interface of handmatige input. Gebruik Kickstart bestanden om het installatie proces te automatiseren.

Het gebruik van beide methodes opent dit venster, welke de initiële opstart fases van je guest laat zien:

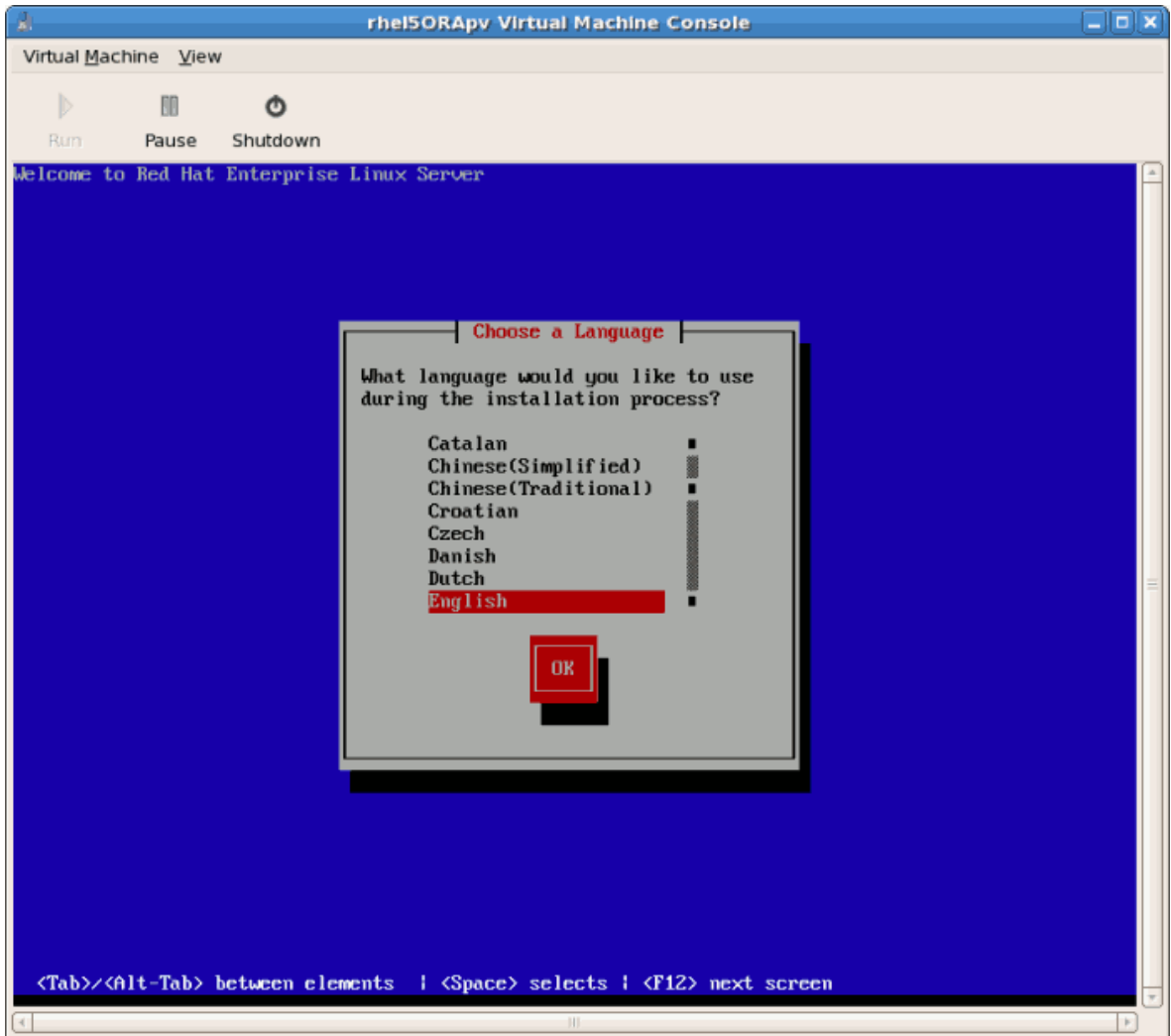


```
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
ide-floppy driver 0.99.newide
usbcore: registered new driver libusual
usbcore: registered new driver hiddev
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
PNP: No PS/2 controller found. Probing ports directly.
i8042.c: No controller found.
mouse: PS/2 mouse device common for all mice
md: md driver 0.90.3 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 4.39
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI No-Shortcut mode
XENBUS: Device with no driver: device/vbd/51712
XENBUS: Device with no driver: device/vif/0
Freeing unused kernel memory: 180k freed
Write protecting the kernel read-only data: 355k
Greetings.
anaconda installer init version 11.1.2.16 starting
mounting /proc filesystem... done
creating /dev filesystem... done
mounting /dev/pts (unix98 pty) filesystem... done
mounting /sys filesystem... done
trying to remount root filesystem read write... done
mounting /tmp as ramfs... done
running install...
running /sbin/loader
```

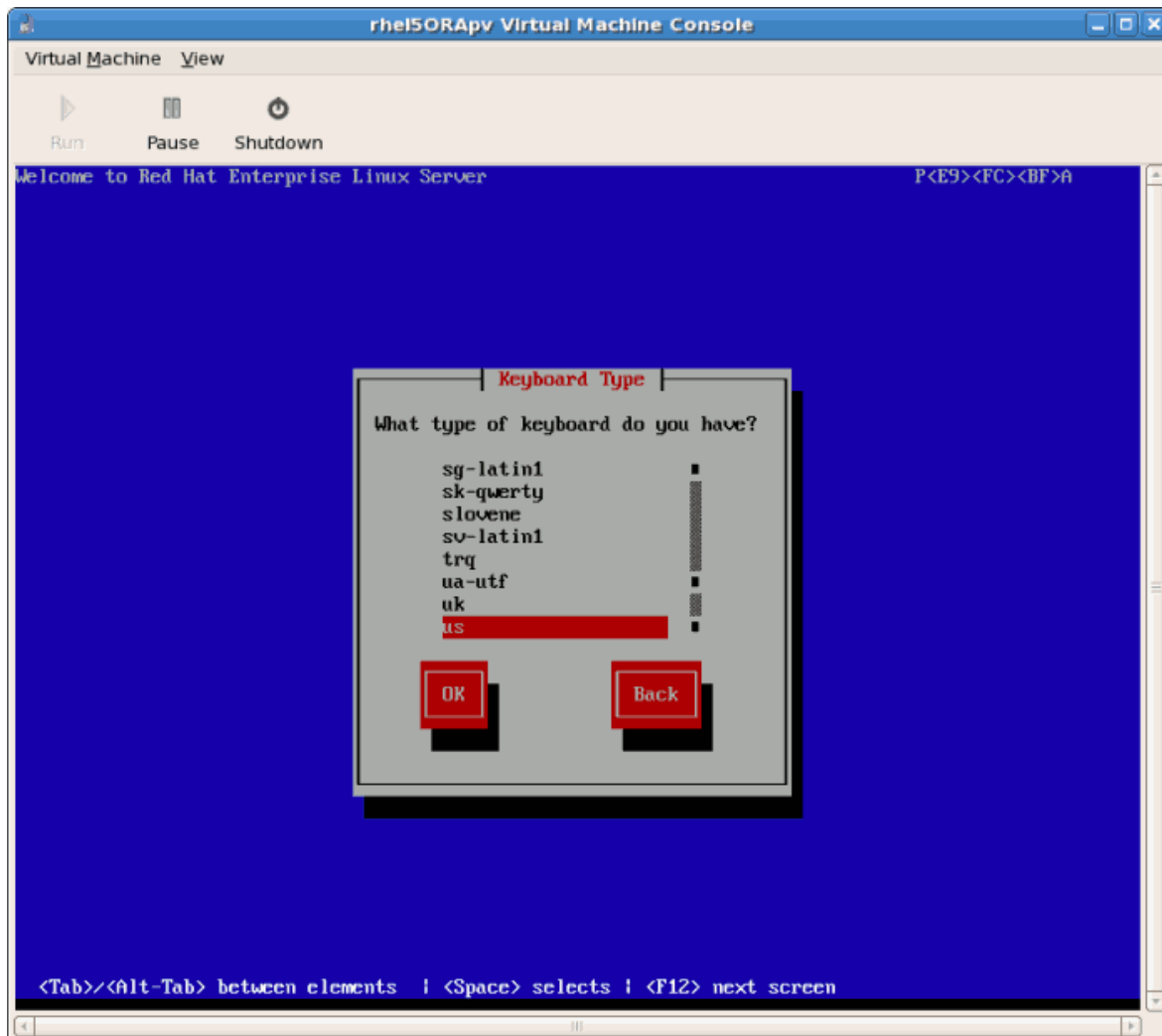
Nadat jouw systeem zijn initiële opstart heeft afgemaakt, begint het standaard installatie proces voor Red Hat Enterprise Linux. Voor de meeste systemen zijn de standaard antwoorden acceptabel.

[Procedure 3.1. Para-gevirtualiseerde Red Hat Enterprise Linux guest installatie procedure](#)

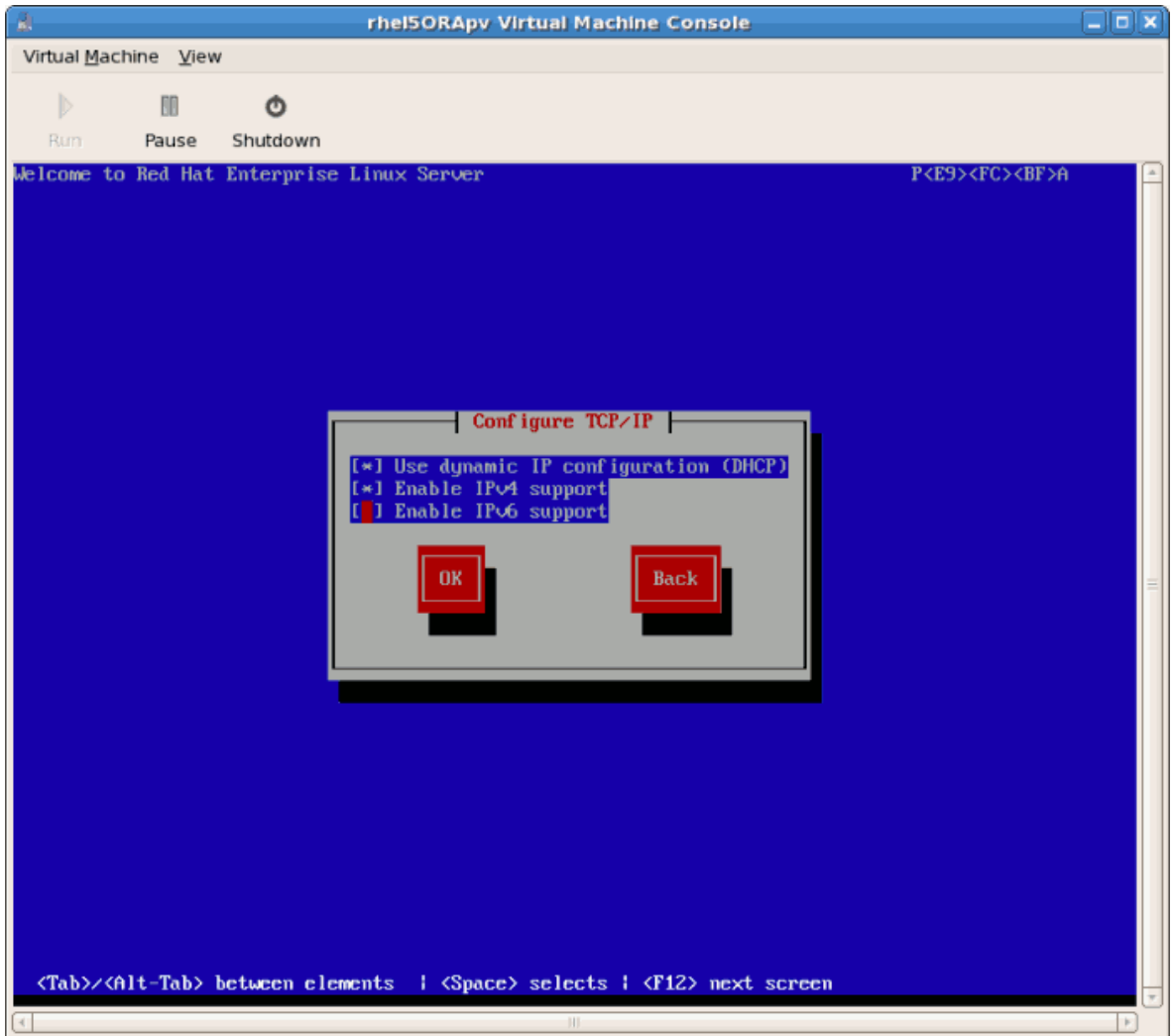
1. Selecteer de taal en klik op **OK**.



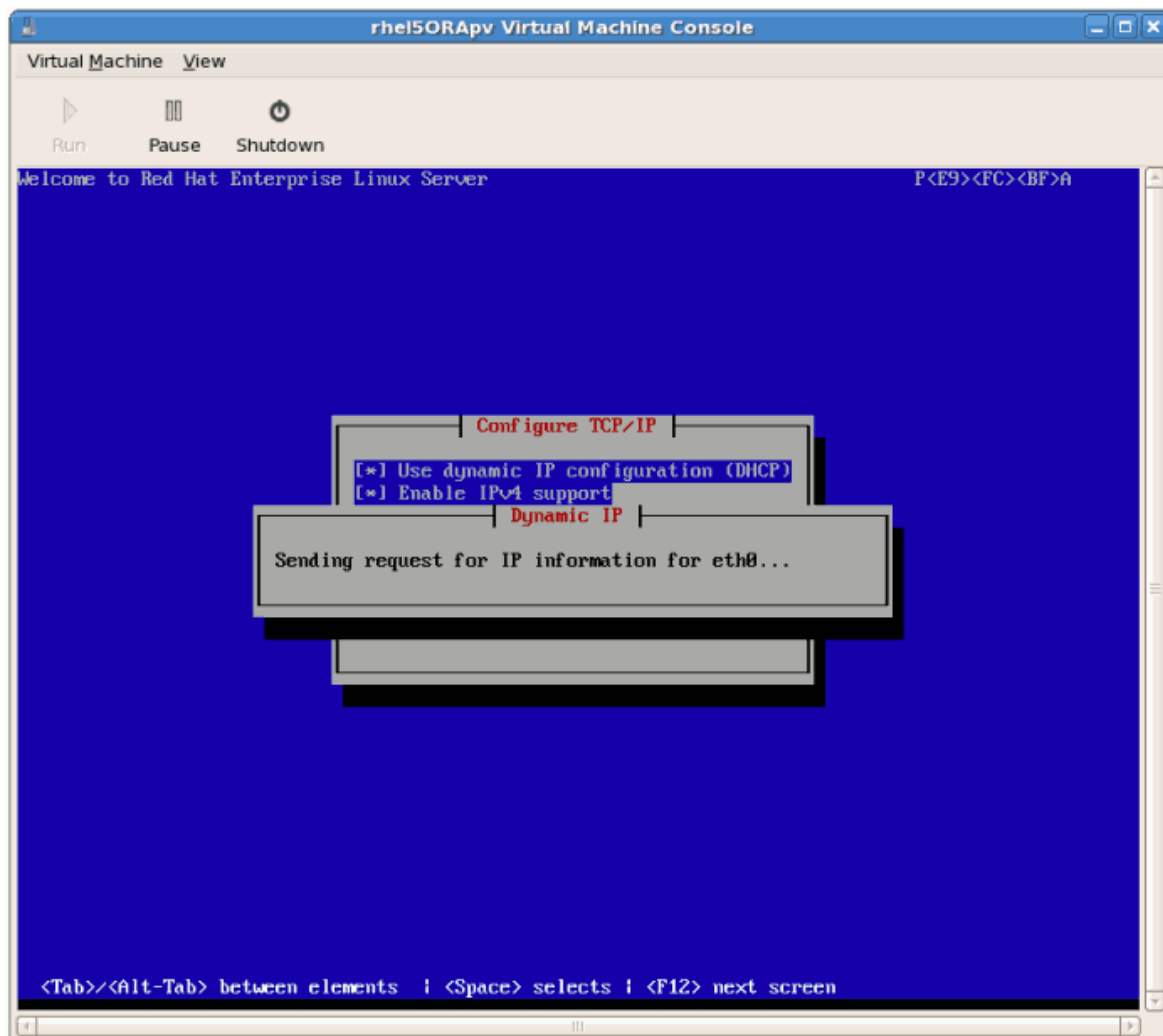
2. Selecteer het toetsenbord en klik op **OK**.



3. Ken het netwerk adres van de guest toe. Kies ervoor om DHCP (zoals hieronder getoond) te gebruiken of kies een statisch IP adres:

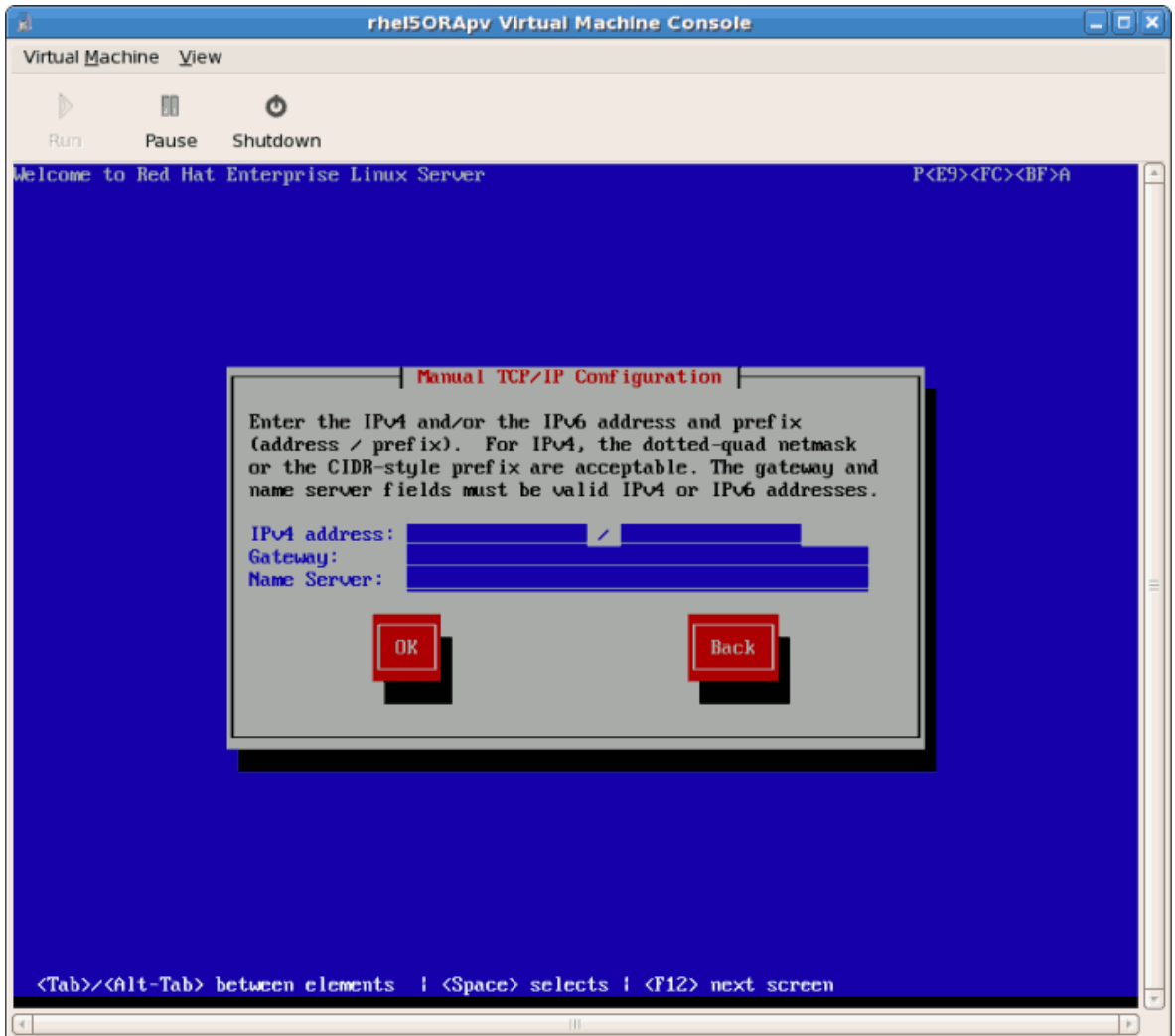


4. Als je DHCP selecteerde zal het installatie proces nu proberen om een IP adres te verkrijgen:

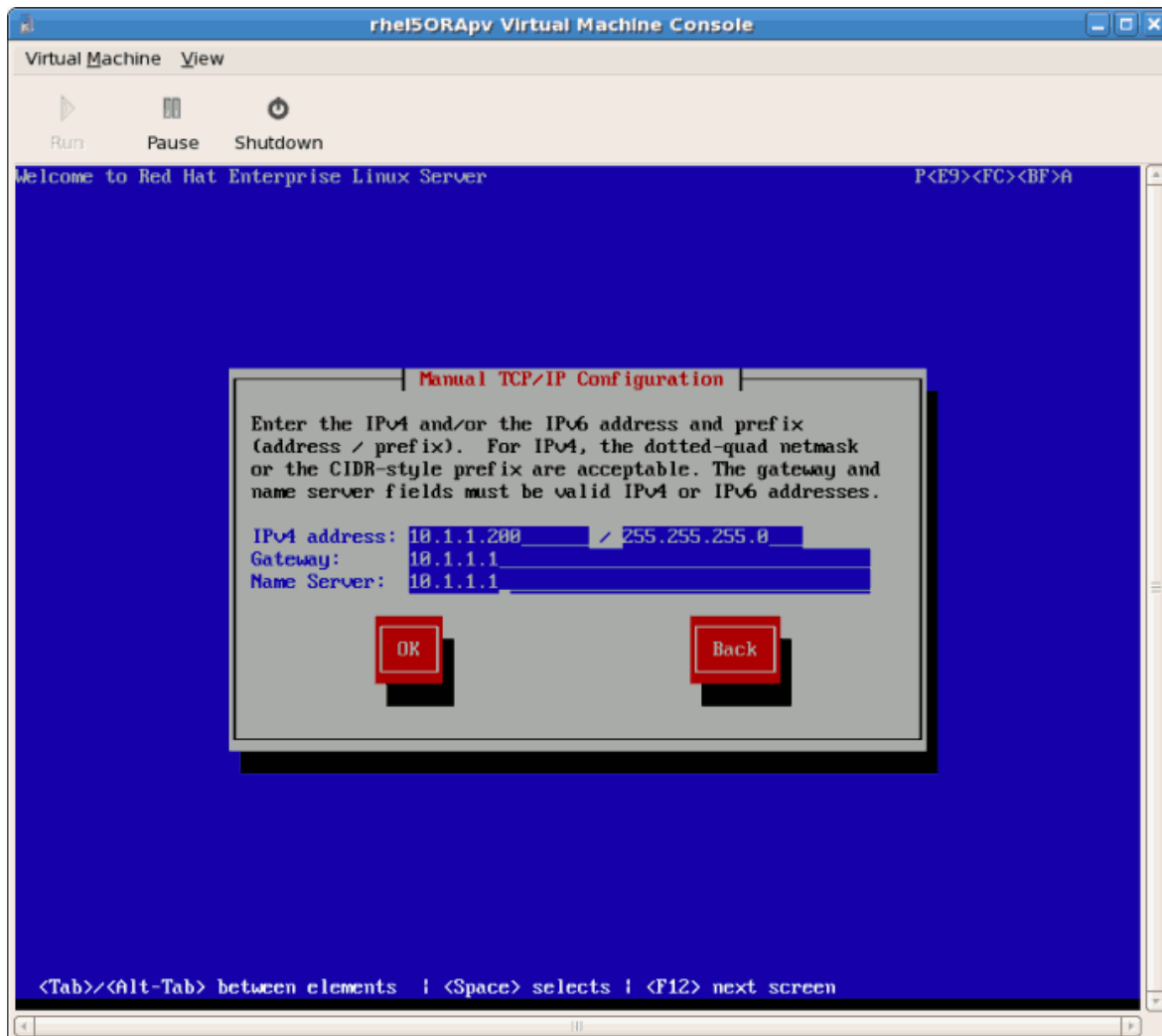


5. Als je een statisch IP adres voor je guest koos verschijnt deze prompt. Vul de details in op de netwerk configuratie van de guest:
 - a. Vul een geldig IP adres in. Verzeker je ervan dat het adres dat je opgeeft de server met de installatie boom kan bereiken.
 - b. Vul een geldig sub-net masker in, standaard gateway en naamserver adres.

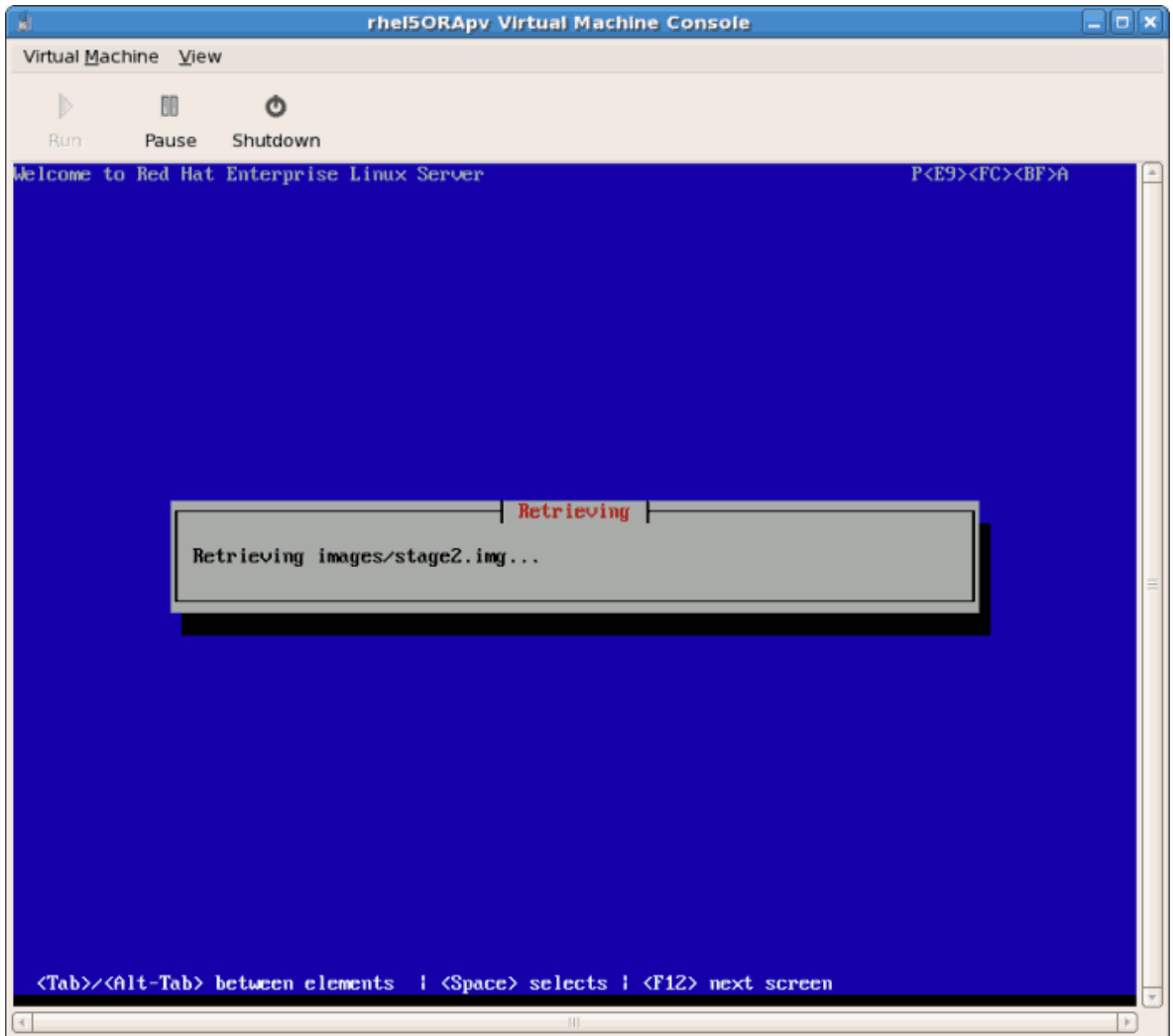
Selecteer de taal en klik op **OK**.



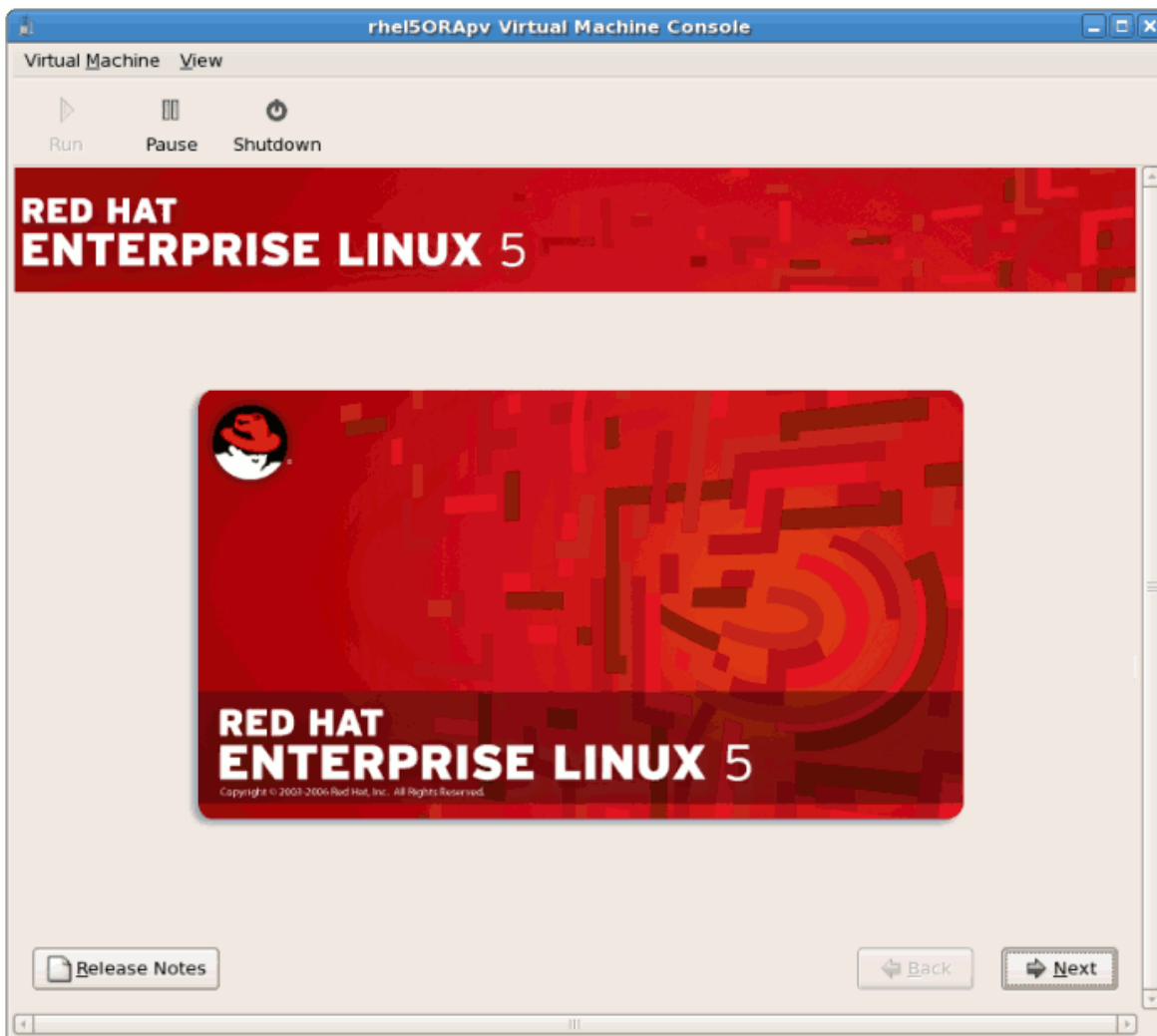
6. Dit is een voorbeeld van een statische IP adres configuratie:



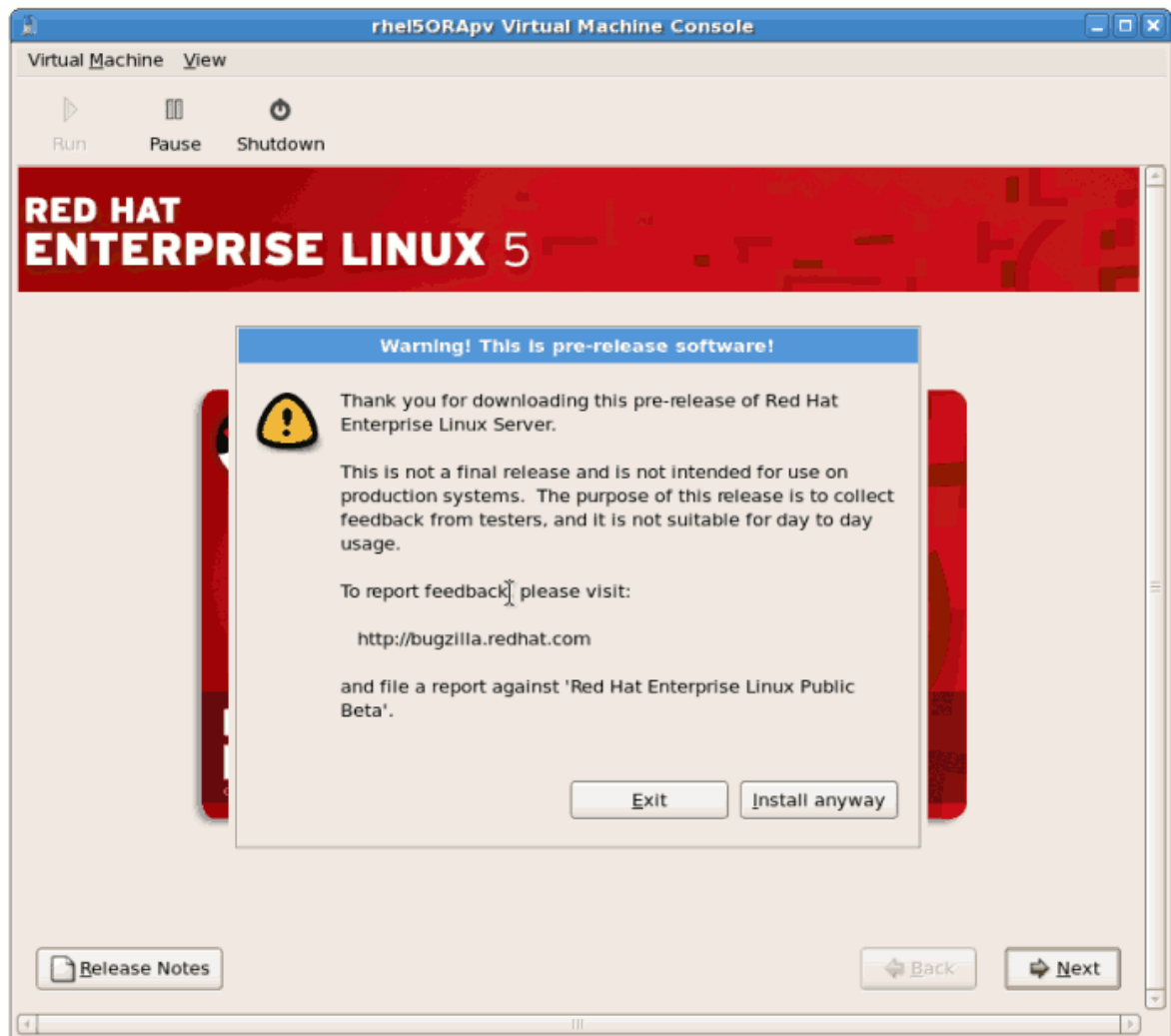
7. Het installatie proces haalt nu de bestanden die het nodig heeft op van de server:



Zodra de initiële stappen klaar zijn start het grafische installeer proces.

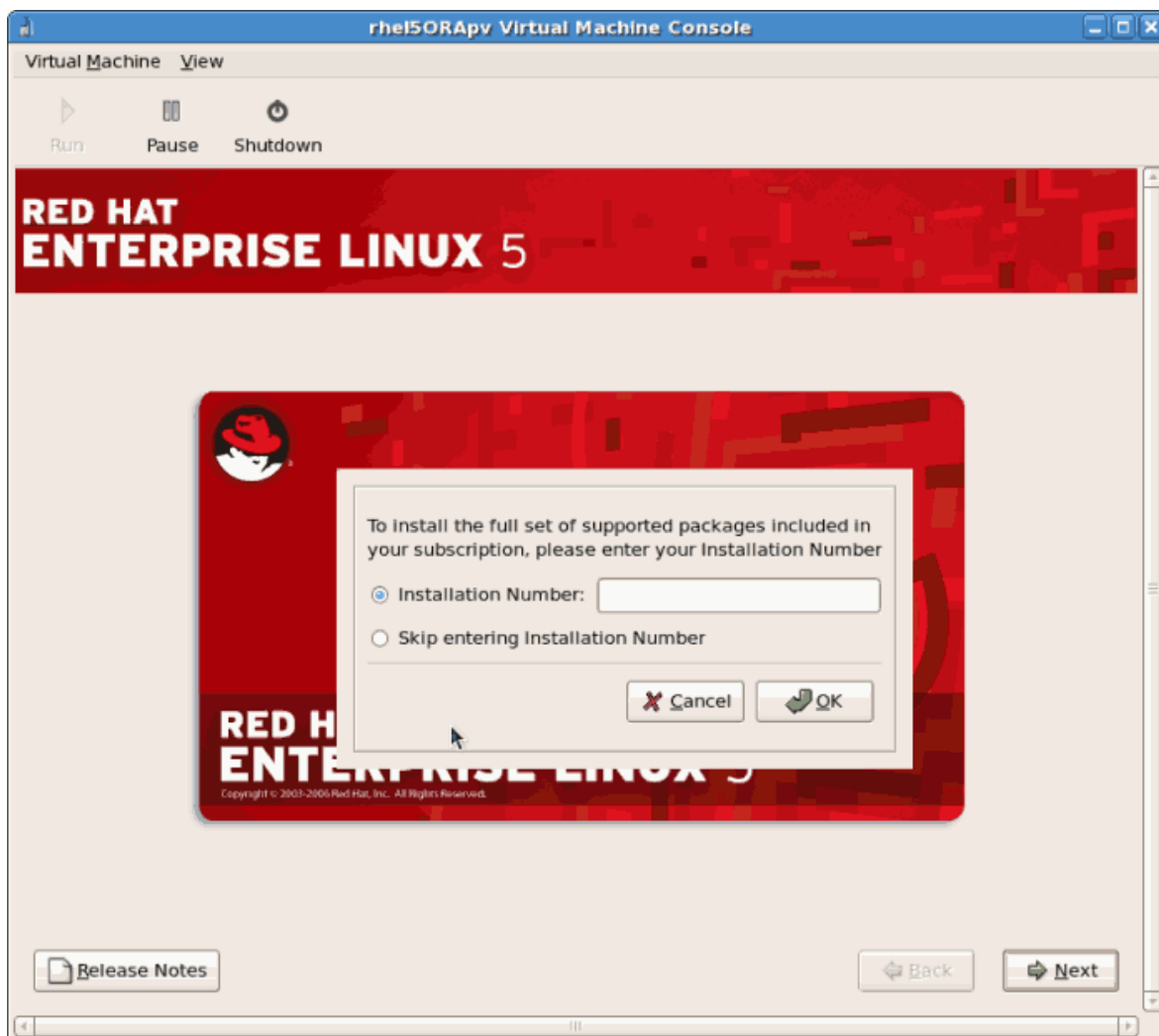


Als je een beta of vroege vrijgave distributie installeert, bevestig dan dat je het bedrijfssysteem wilt installeren. Klik op **Install Anyway**, en klik dan op **OK**:



Procedure 3.2. Het grafische installatie proces

1. Vul een geldige registratie code in. Als je een geldige RHN abonnement sleutel hebt vul je die in het Installation Number veld in:

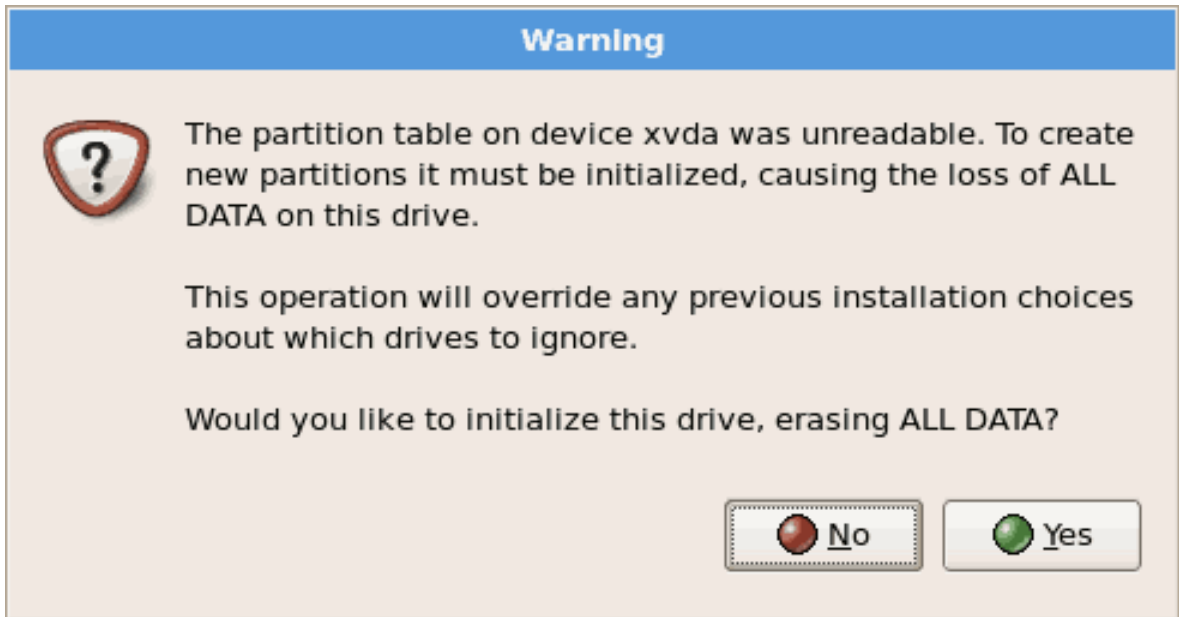


Note

Als je de registratie stap overslaat kun je jouw Fedora netwerk account details bevestigen na de installatie met het `rhnc_register` commando. Het `rhnc_register` commando vereist root toegang.

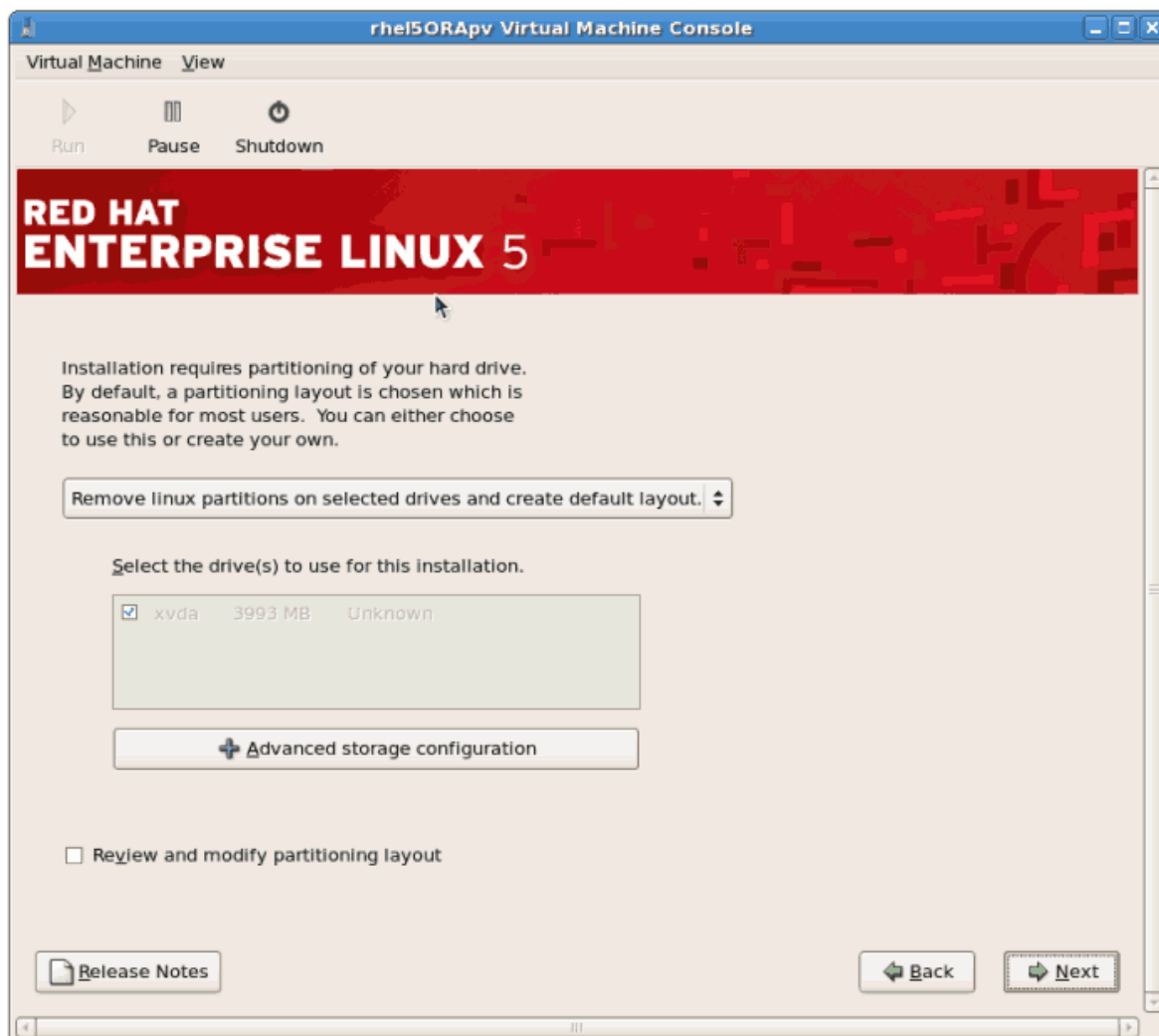
```
# rhnc_register
```

2. De installatie vraagt je om het wissen van alle data te bevestigen voor de opslag die je selecteerde voor de installatie:



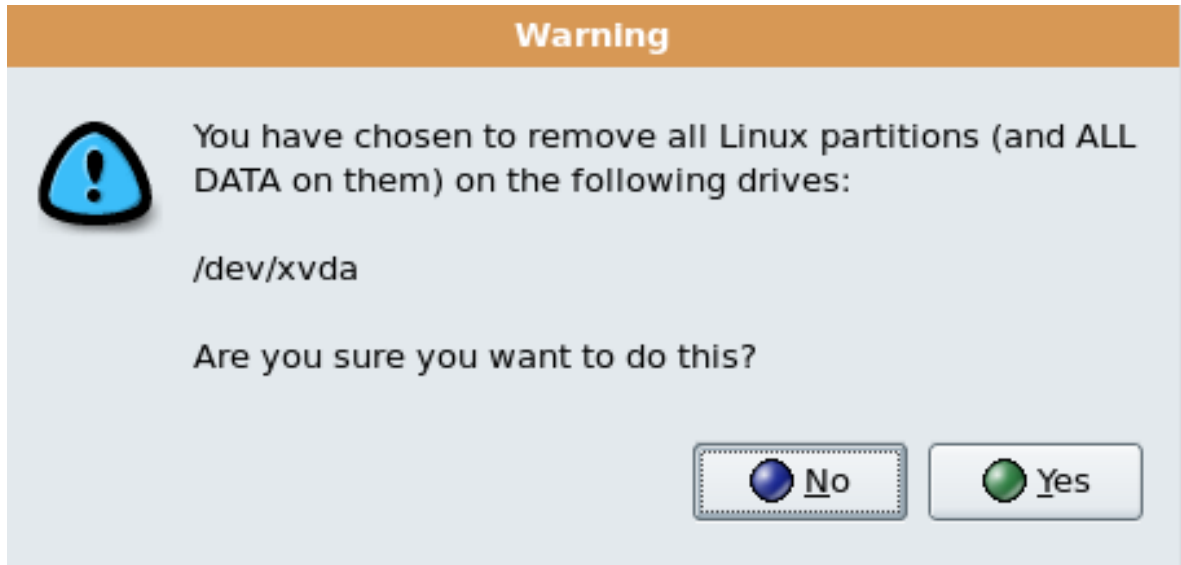
Klik op **Yes** om verder te gaan.

3. Bekijk de opslag configuratie en partitie indeling. Je kunt geavanceerde opslag configuratie te selecteren als je iSCSI wilt gebruiken voor de opslag voor de guest.



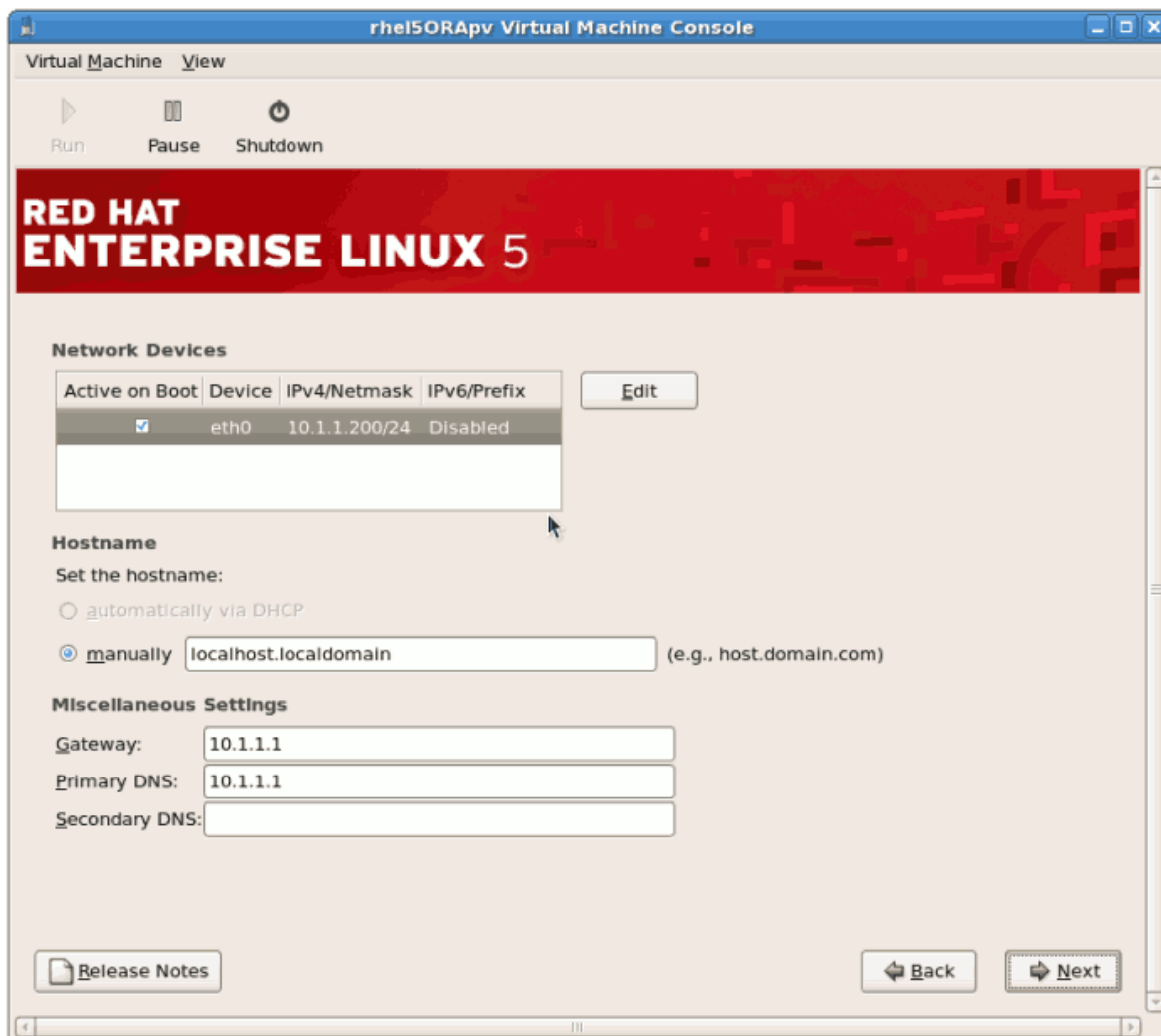
Maak je selecties en klik dan op **Next**.

4. Bevestig de geselecteerde opslag voor de installatie.



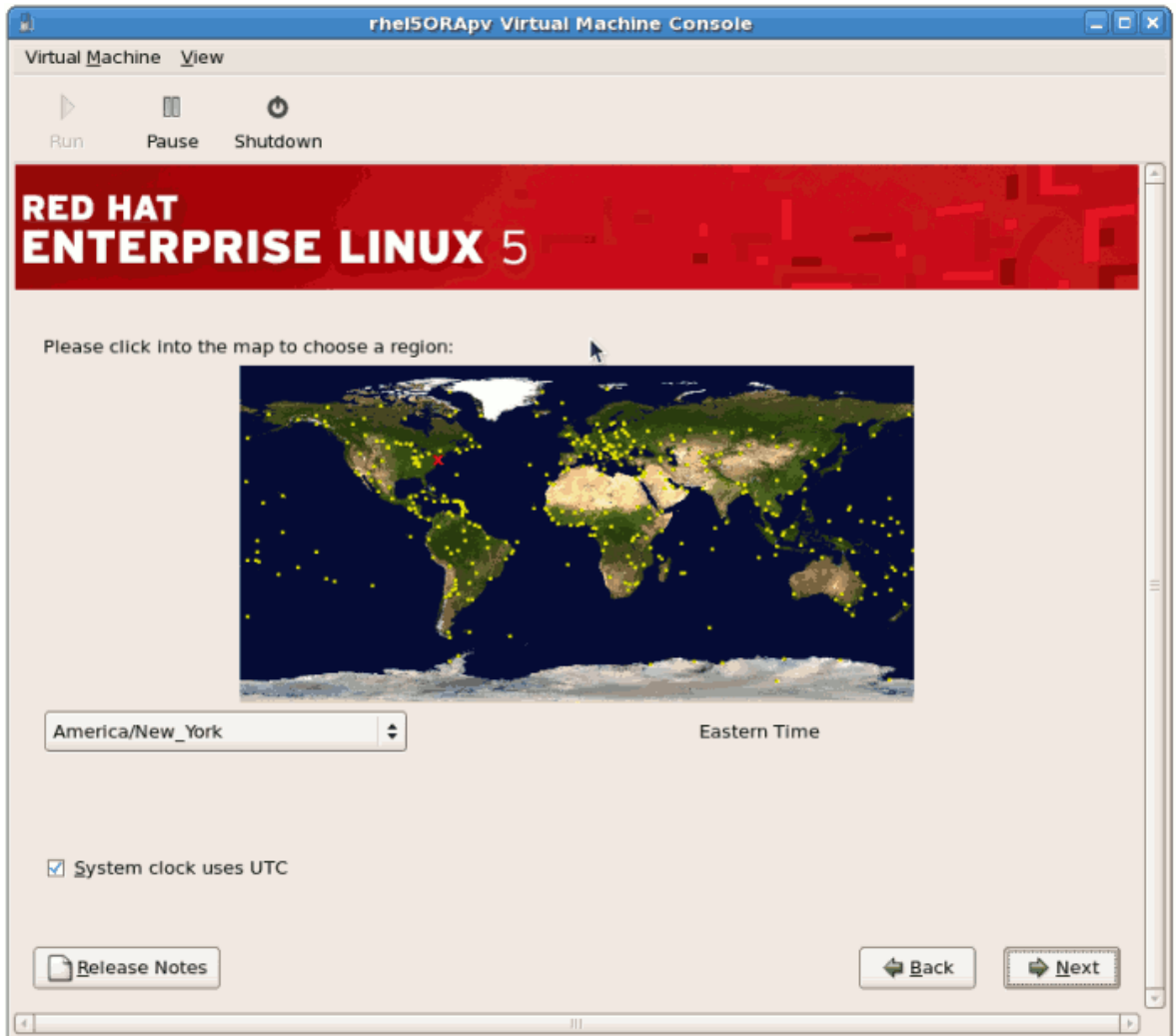
Klik op **Yes** om verder te gaan.

5. Configureer netwerk en hostnaam instellingen. Deze instellingen zijn ingevuld met de data die eerder in het installatie proces zijn opgegeven. Verander indien nodig deze instellingen.

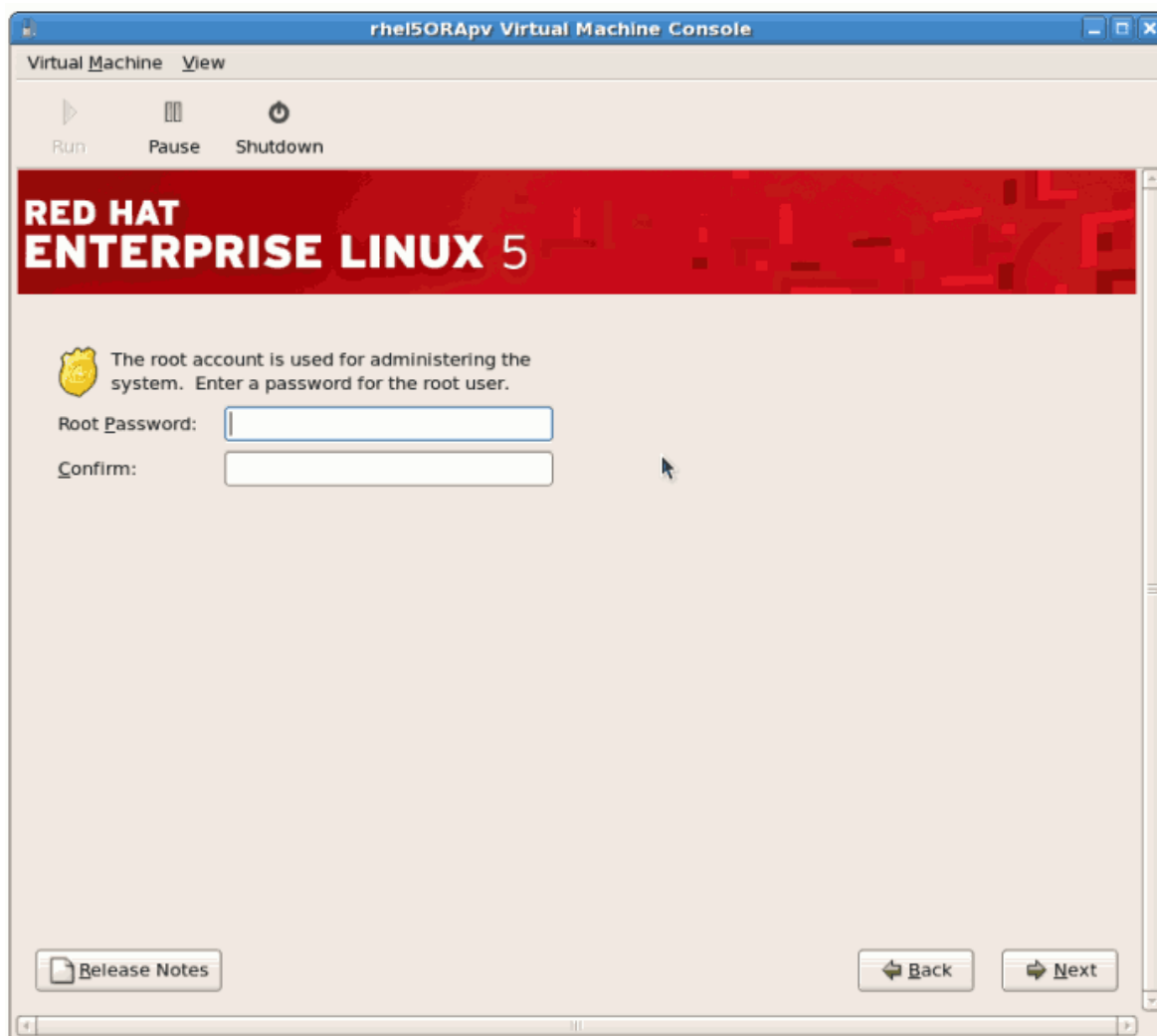


Klik op **OK** om verder te gaan.

6. Selecteer de juiste tijdzone voor jouw omgeving.

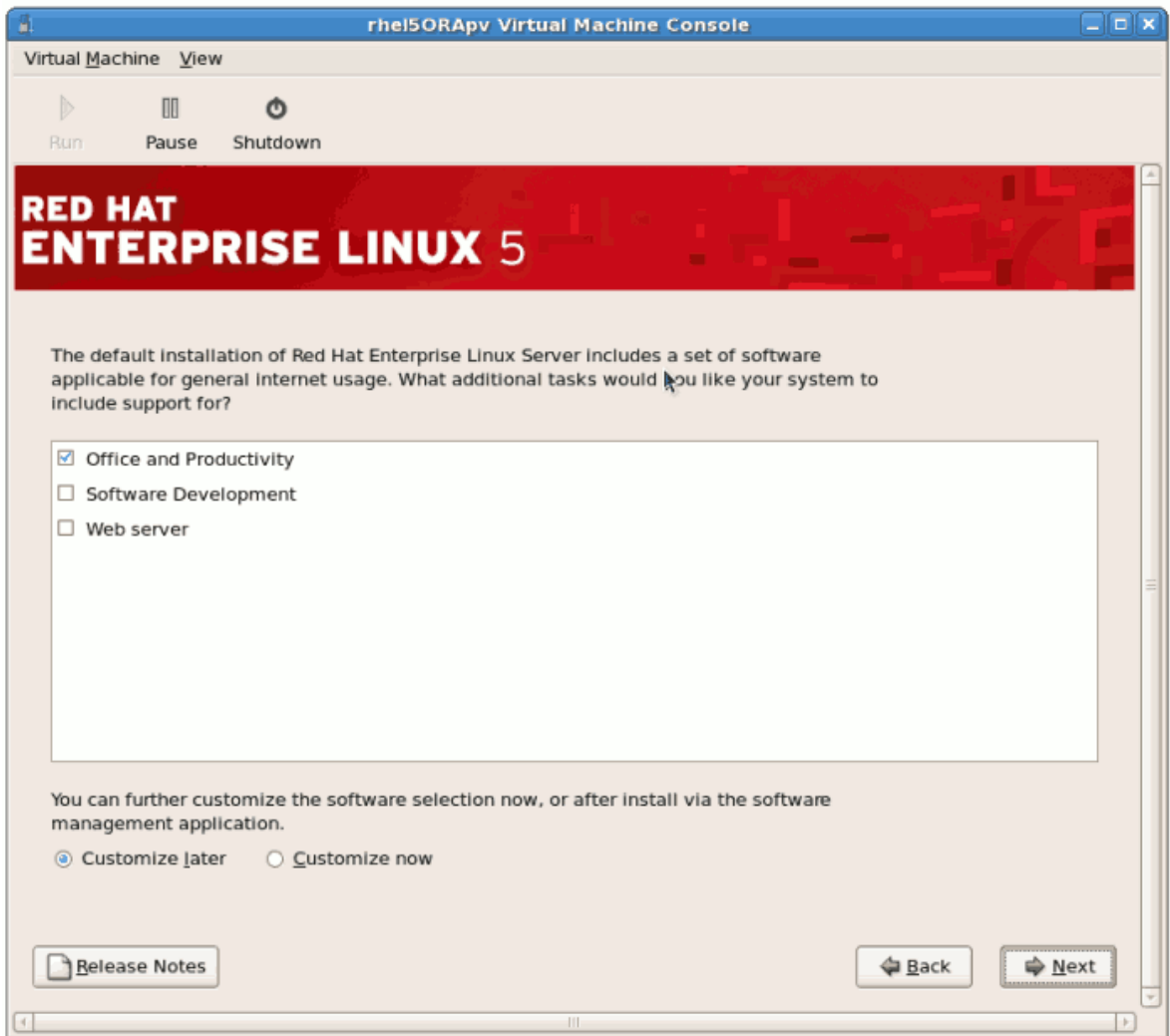


7. Vul het root wachtwoord van de guest in.



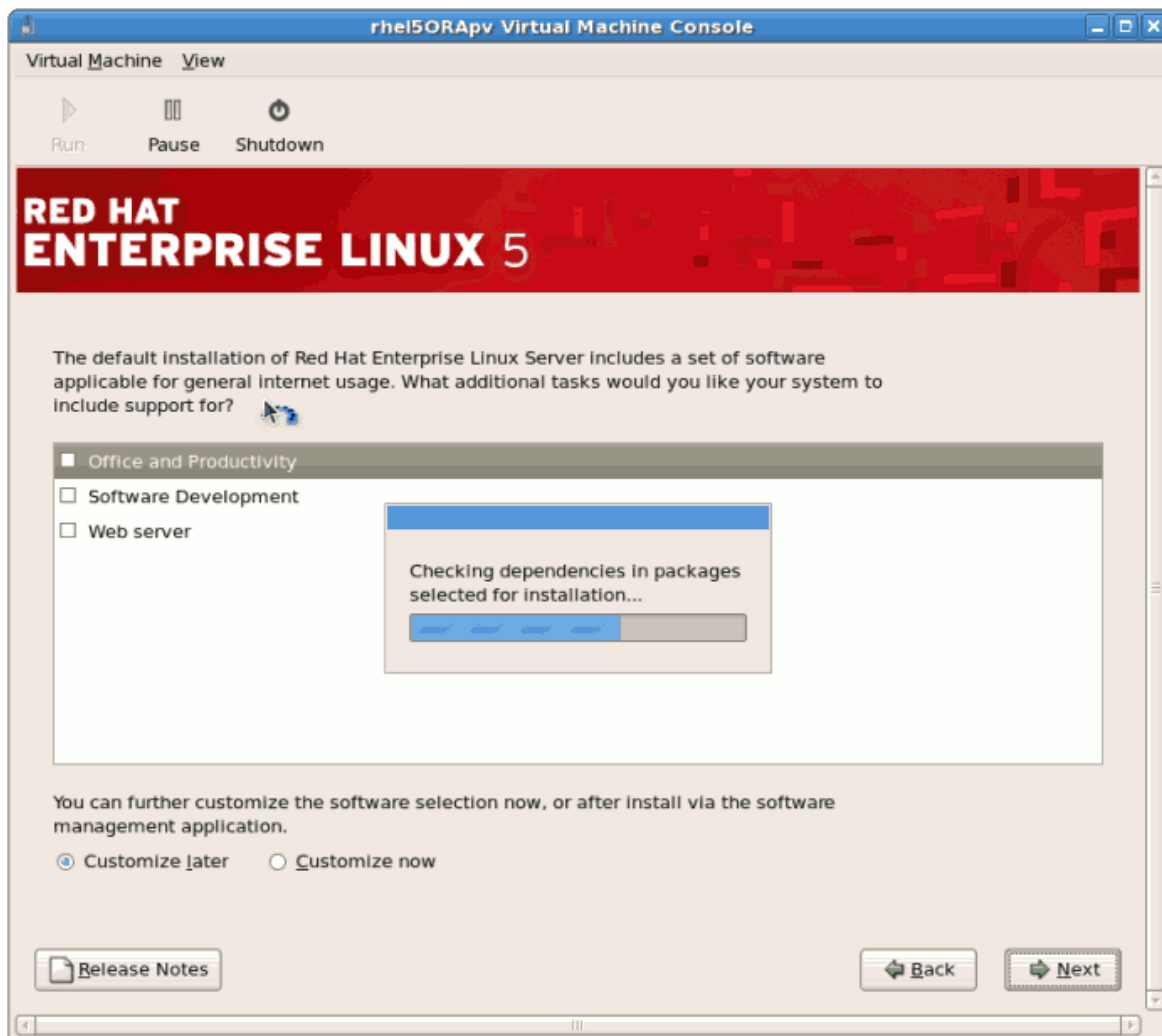
Klik op **Next** om verder te gaan.

8. Selecteer de te installeren software pakketten. Selecteer de **Customize Now** knop. Je moet het **kernel-xen** pakket in de **System** map installeren. Het **kernel-xen** pakket is vereist voor para-virtualisatie.

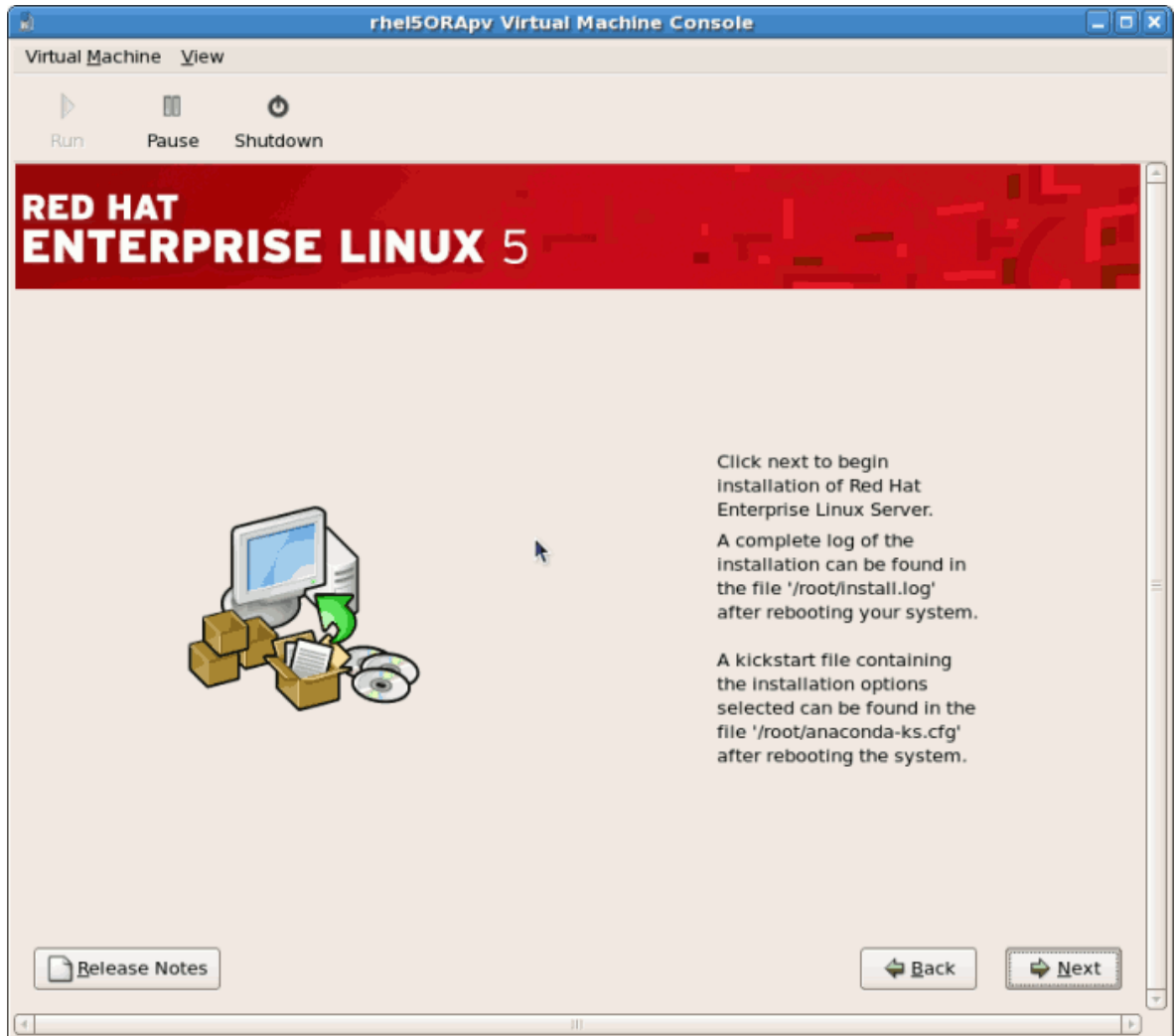


Klik op **Next**.

9. Afhankelijkheden en ruimte vereisten worden berekend.



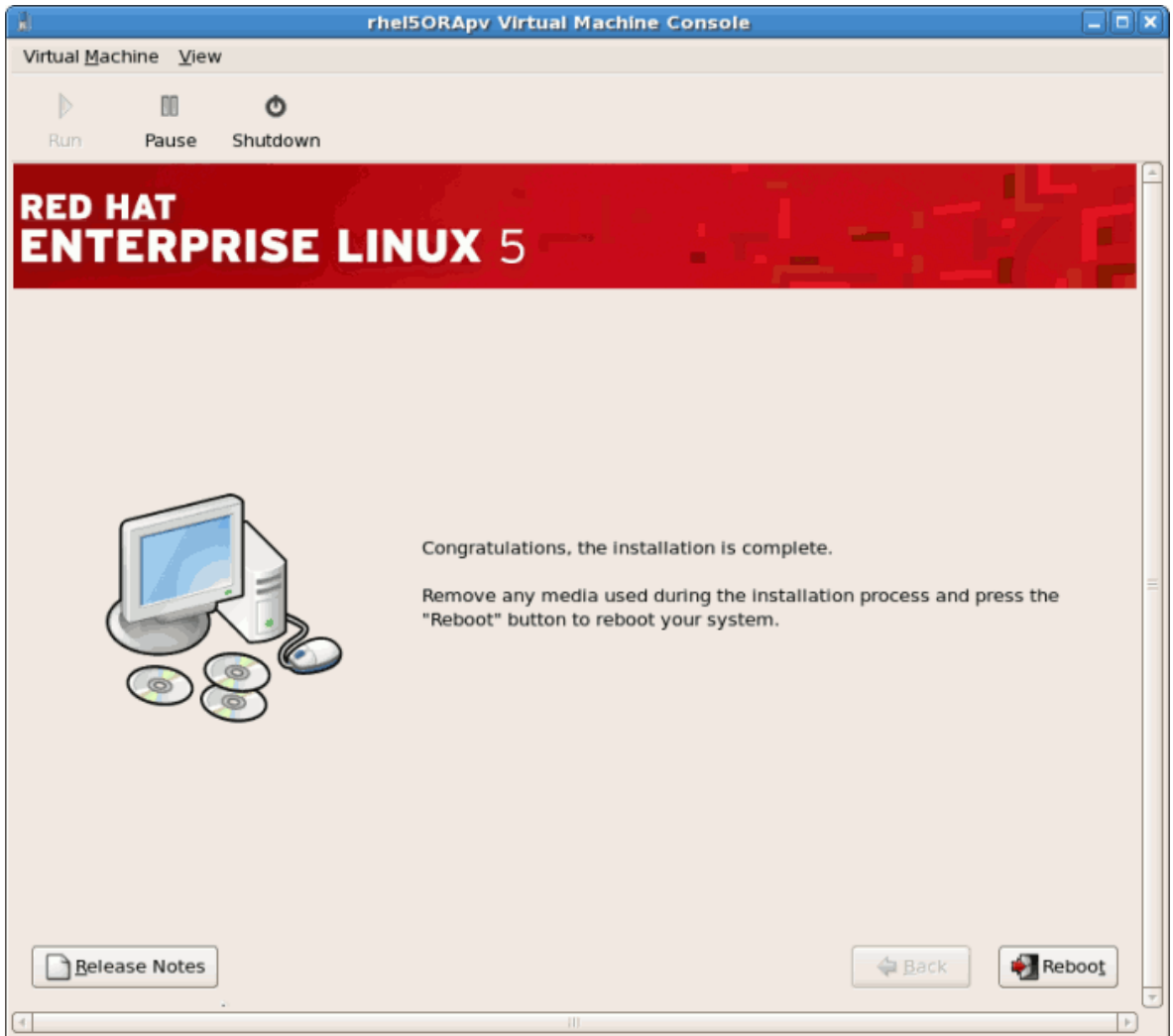
10. Nadat de installatie afhankelijkheden en ruimte vereisten geverifieerd zijn, klik je op **Next** om de feitelijke installatie te beginnen.



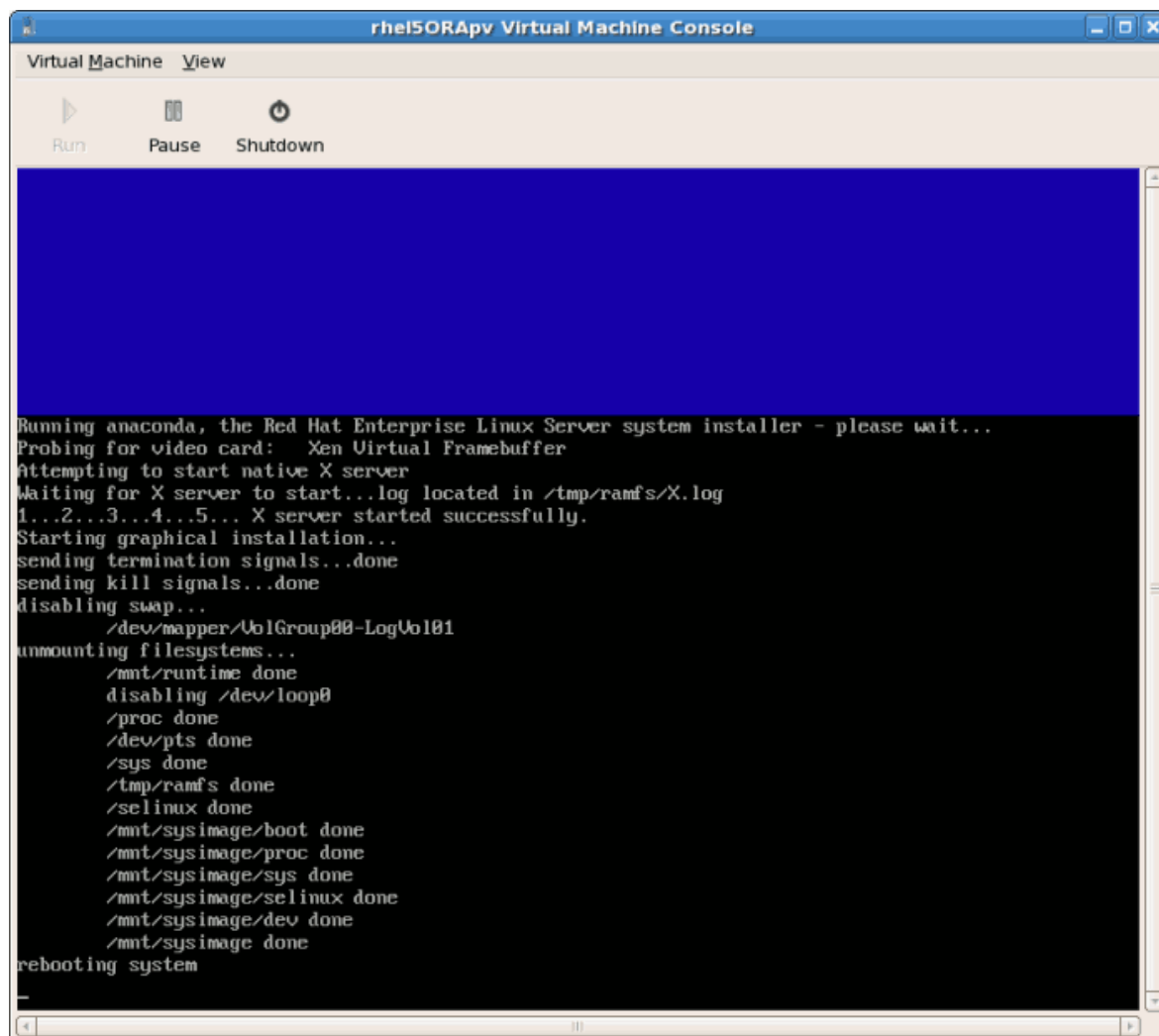
11. Alle geselecteerde software pakketten worden automatisch geïnstalleerd.



12. Zodra de installatie klaar is start je de guest opnieuw op:



13. De gast zal niet opnieuw opstarten, maar het zal afsluiten..



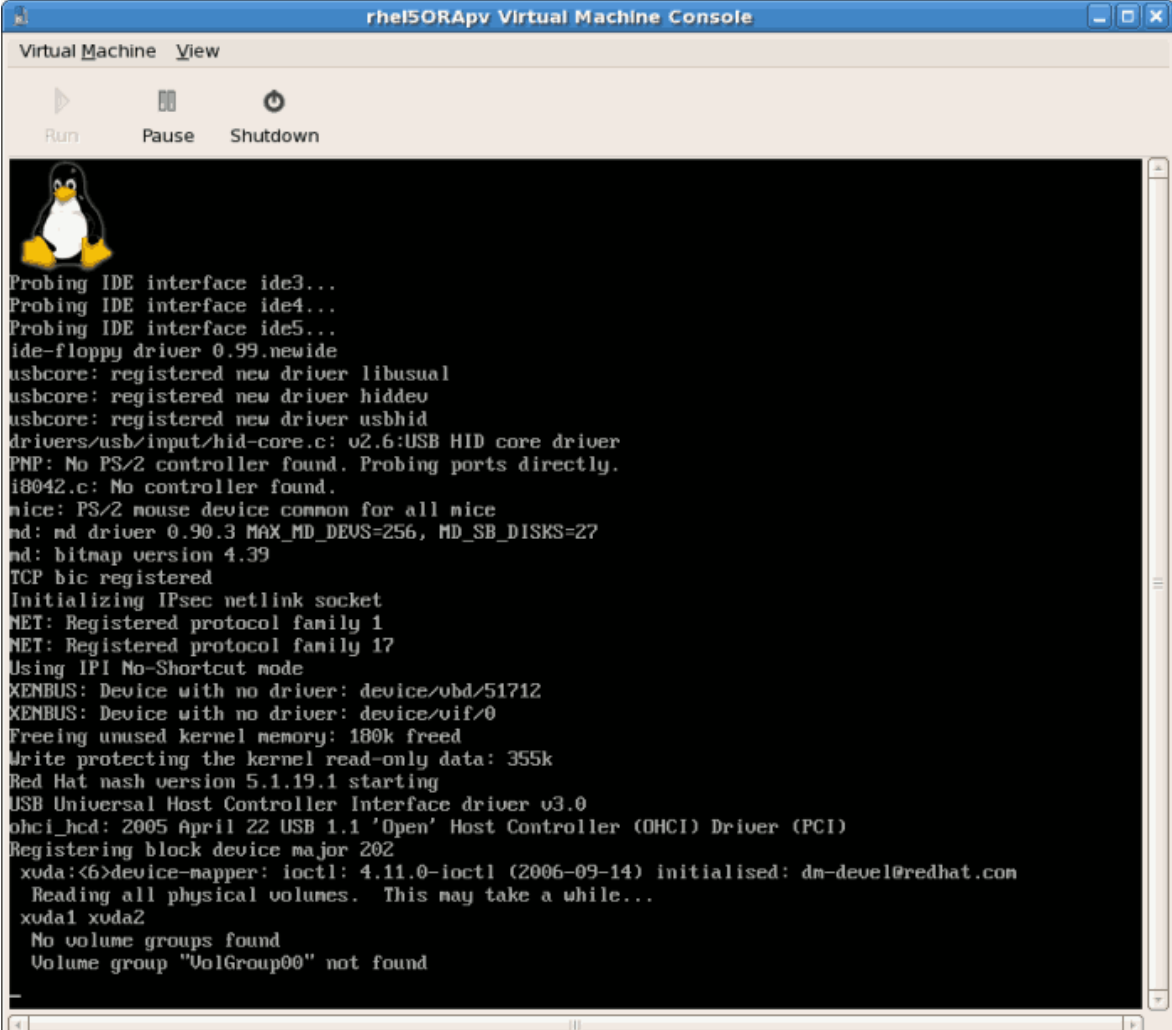
14. Start de guest op. De naam van de guest werd gekozen toen je **virt-install** gebruikte in [Paragraaf 3.1, "Red Hat Enterprise Linux 5 installeren als een para-gevirtualiseerde guest"](#). Als je het standaard voorbeeld gebruikt hebt is de naam *rhe15PV*.

Voer uit:

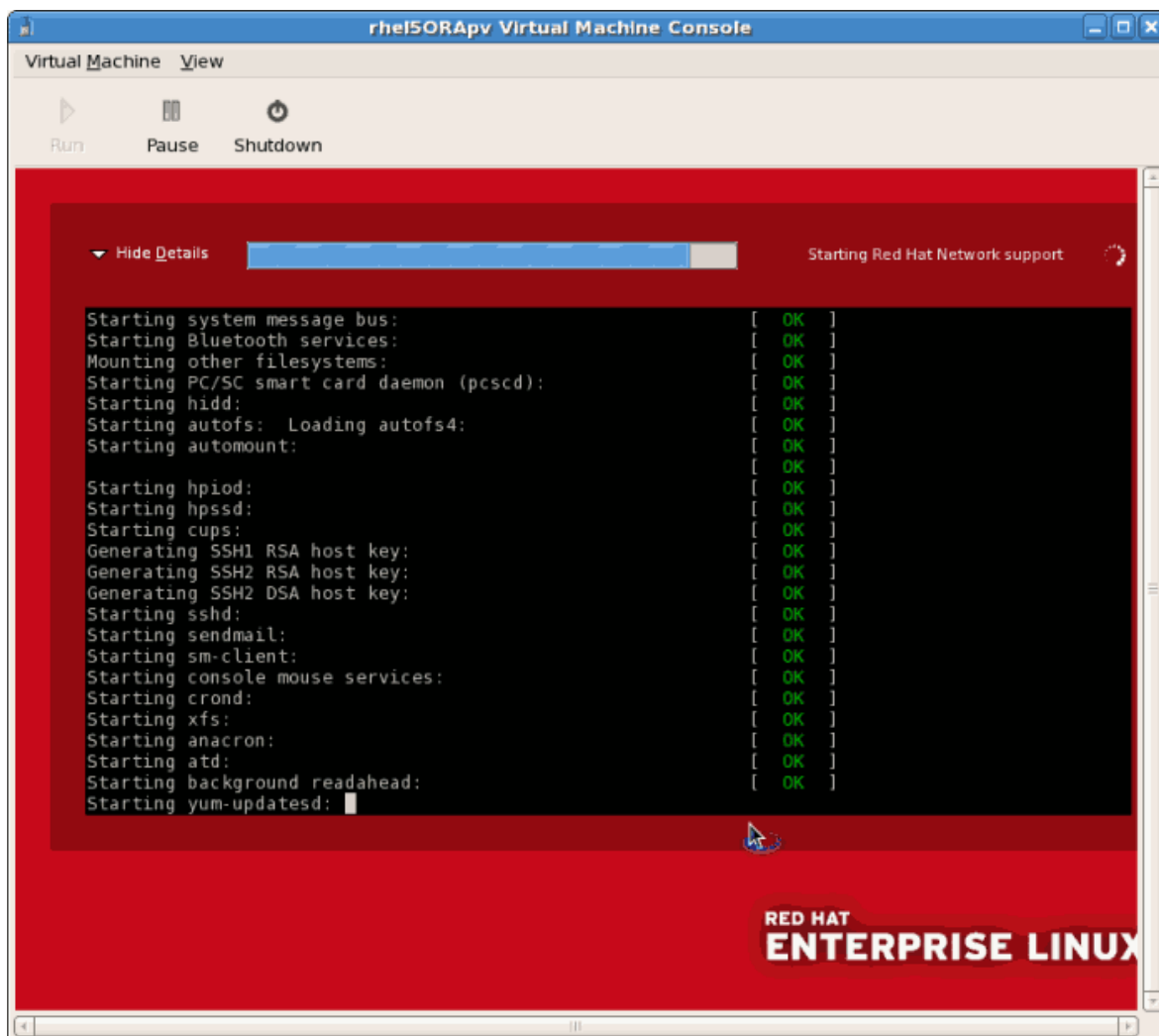
```
virsh reboot rhe15PV
```

Als alternatief open je **virt-manager**, je selecteert de naam van jouw guest, je klikt op **Open**, en daarna op **Run**.

Een **VNC** venster opent nu die de opstart processen van de guest laat zien.



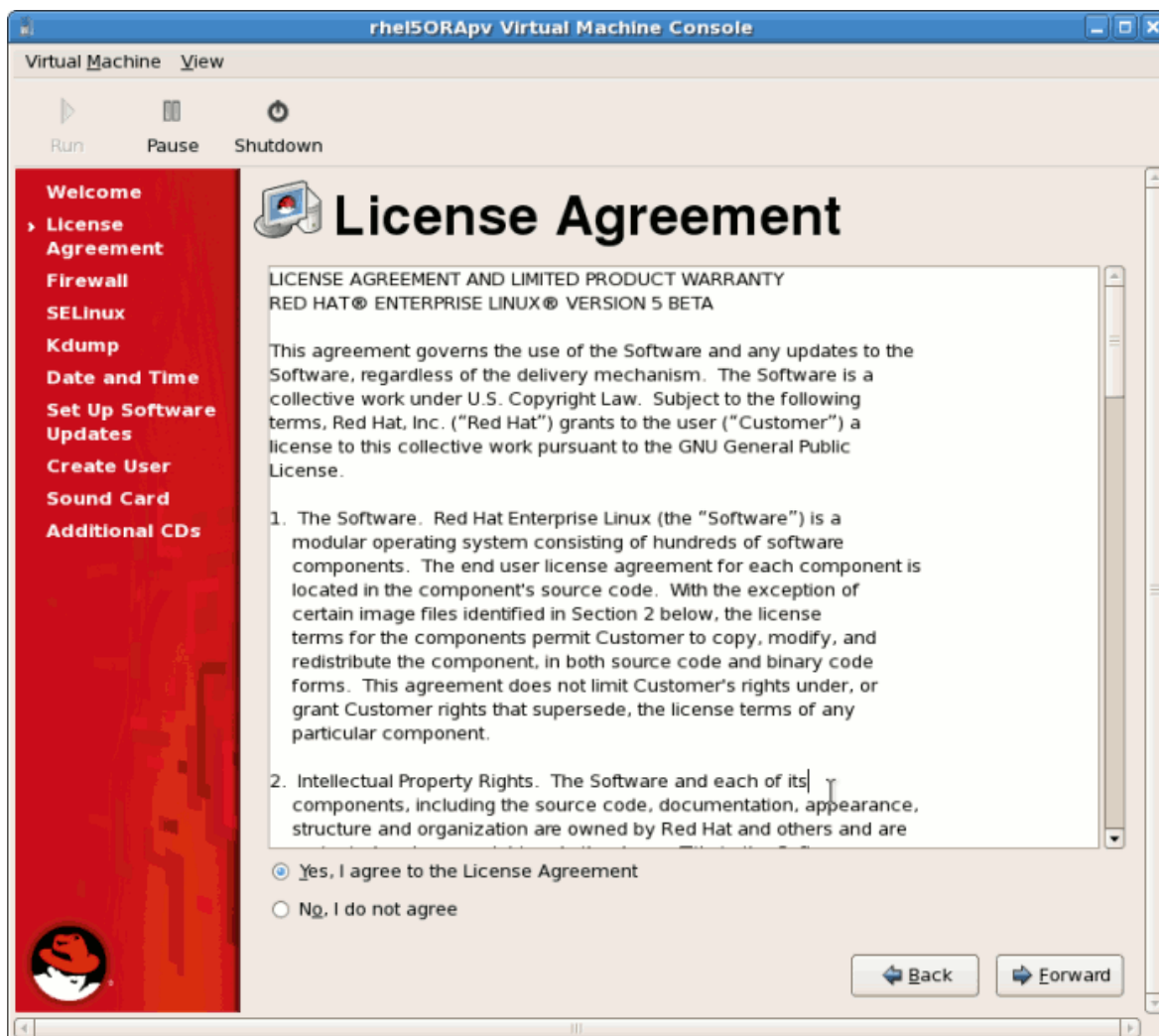
```
rheISOApv Virtual Machine Console
Virtual Machine View
Run Pause Shutdown
Probing IDE interface ide3...
Probing IDE interface ide4...
Probing IDE interface ide5...
ide-floppy driver 0.99.newide
usbcore: registered new driver libusual
usbcore: registered new driver hiddev
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
PNP: No PS/2 controller found. Probing ports directly.
i8042.c: No controller found.
mice: PS/2 mouse device common for all mice
md: md driver 0.90.3 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 4.39
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI No-Shortcut mode
XENBUS: Device with no driver: device/vbd/51712
XENBUS: Device with no driver: device/vif/0
Freeing unused kernel memory: 180k freed
Write protecting the kernel read-only data: 355k
Red Hat nash version 5.1.19.1 starting
USB Universal Host Controller Interface driver v3.0
ohci_hcd: 2005 April 22 USB 1.1 'Open' Host Controller (OHCI) Driver (PCI)
Registering block device major 202
xoda:<6>device-mapper: ioctl: 4.11.0-ioctl (2006-09-14) initialised: dm-devel@redhat.com
Reading all physical volumes. This may take a while...
xoda1 xoda2
No volume groups found
Volume group "VolGroup00" not found
```



15. Opstarten van de guest opent het *First Boot* configuratie scherm. Dit hulpprogramma vraagt je naar een paar basis configuratie keuzes voor je guest.



16. Lees de licentie overeenkomst en verklaar je hiermee akkoord.



Klik op **Forward** in het licentie overeenkomst venster.

17. Stel de firewall in.

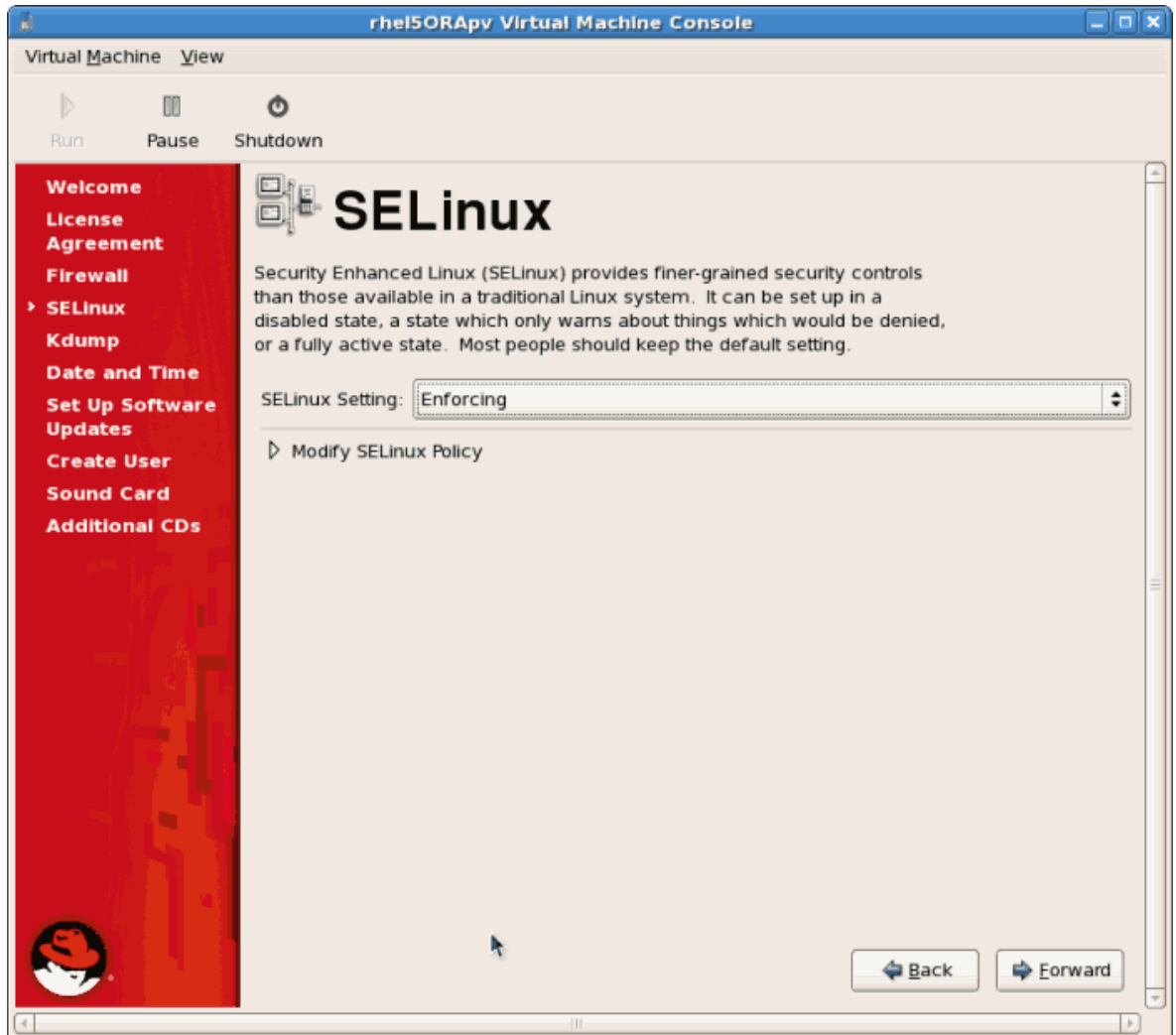


Click **Forward** to continue.

- Als je de firewall uitzet wordt je gevraagd om je keuze te bevestigen. Klik op **Yes** voor de bevestiging en ga verder.



18. Stel SELinux in. Het wordt sterk aanbevolen om SELinux in de **afdwingende mode** te draaien. Je kunt ervoor kiezen om SELinux in de toelatende mode te draaien of om het uit te zetten.



Click **Forward** to continue.

- Als je kiest voor het uitzetten van SELinux verschijnt deze waarschuwing. Klik op **Yes** om SELinux uit te zetten.

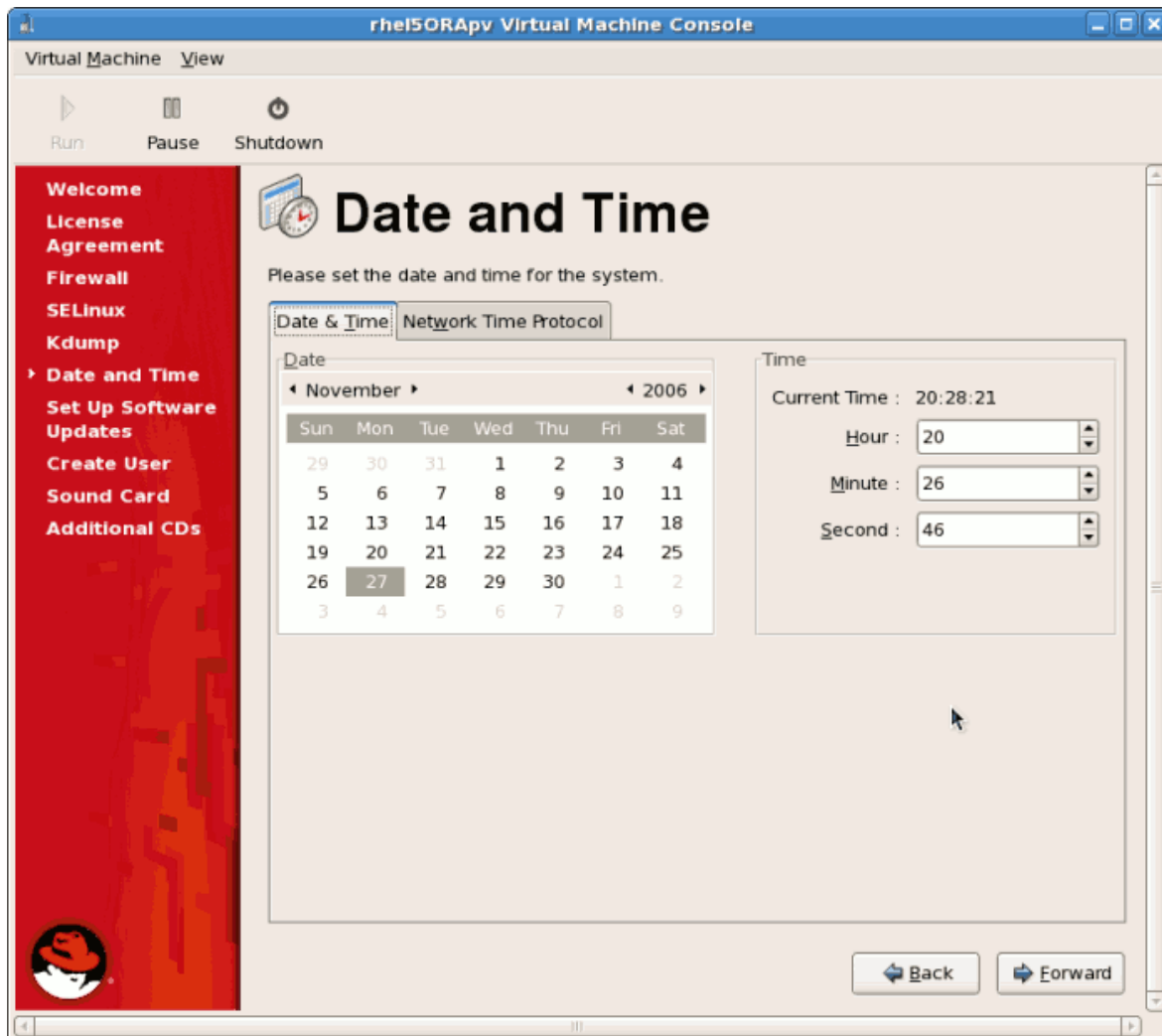


19. Zet indien nodig **kdump** aan.



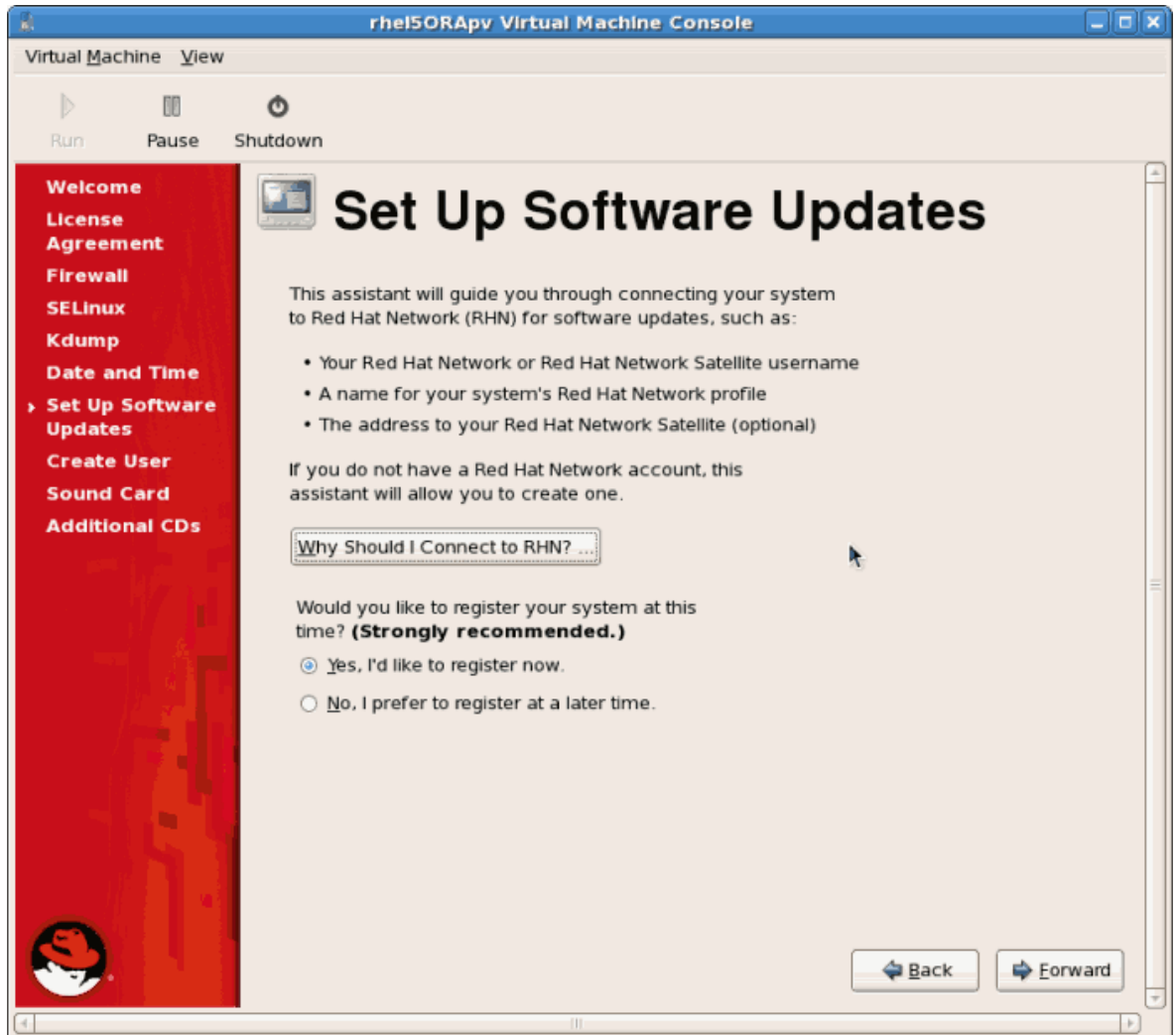
Click **Forward** to continue.

20. Bevestig dat de tijd en datum correct zijn ingesteld voor je guest. Als je een para-gevirtualiseerde guest installeert moeten tijd en datum gesynchroniseerd zijn met de hypervisor.



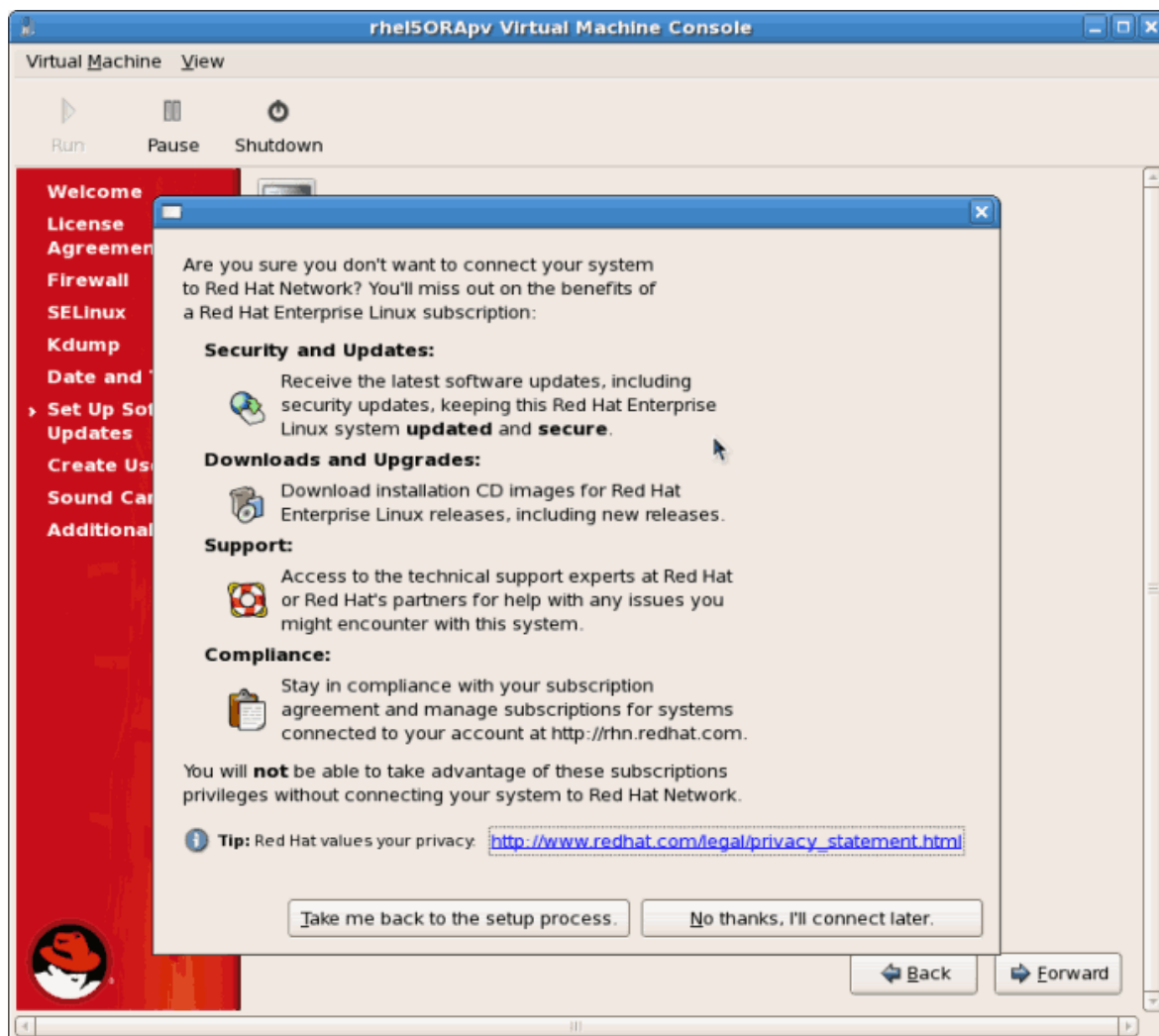
Click **Forward** to continue.

21. Stel software vernieuwingen in. Als je een Fedora netwerk abonnement hebt of je wilt er een proberen, gebruik je het scherm hieronder voor het registreren van je nieuw geïnstalleerde guest in RHN.

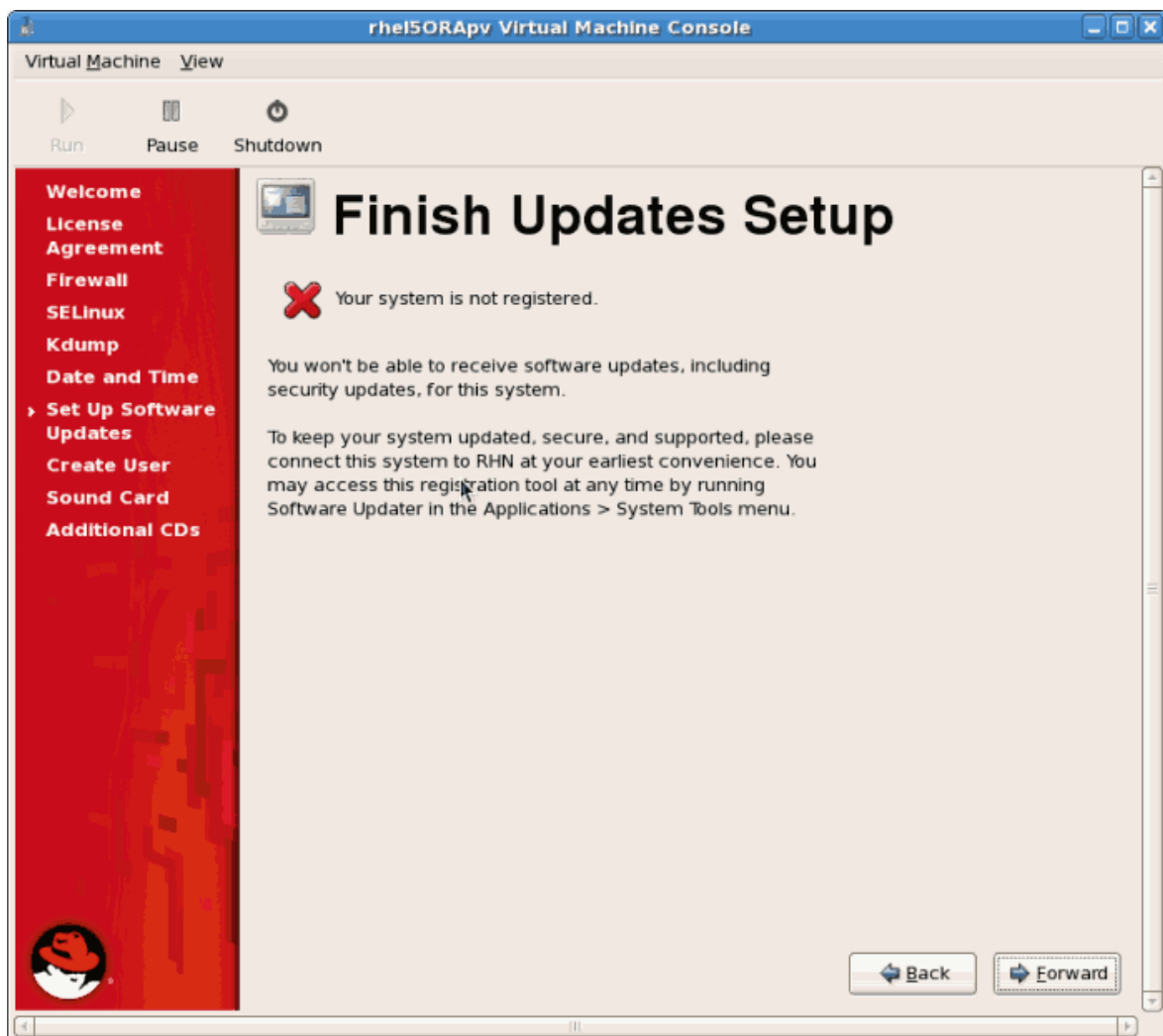


Click **Forward** to continue.

- a. Bevestig je keuzes voor RHN.

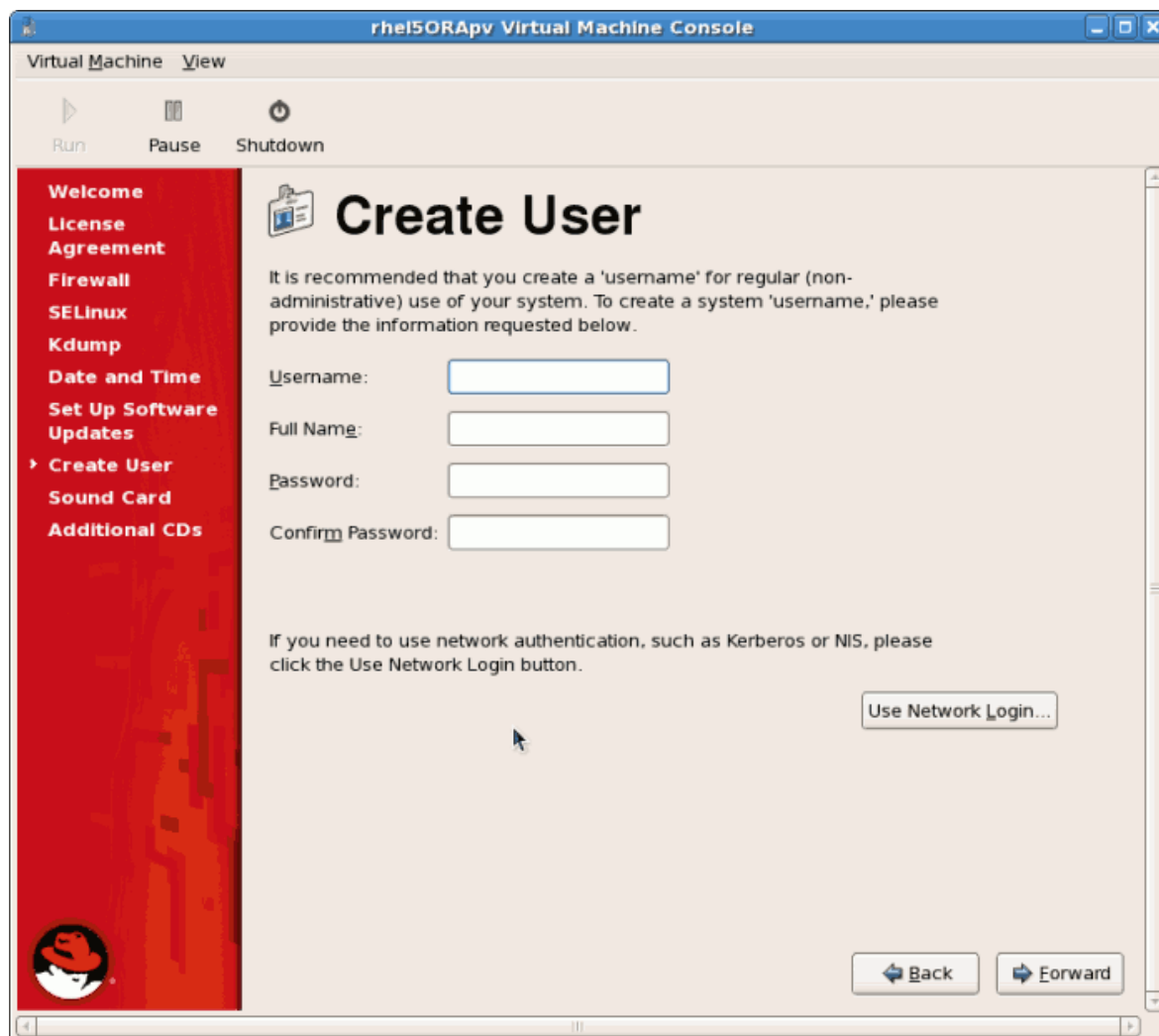


- b. Zodra de instelling beëindigd is zie je misschien nog een venster als je je niet registreerde bij RHN. Je zult geen software vernieuwingen ontvangen.



Klik op de **Forward** knop.

22. Maak een gebruikersaccount aan anders dan root. Het wordt aanbevolen om een gebruikersaccount anders dan root aan te maken voor normaal gebruik en verbeterde beveiliging. Vul de gebruikersnaam en wachtwoord in.

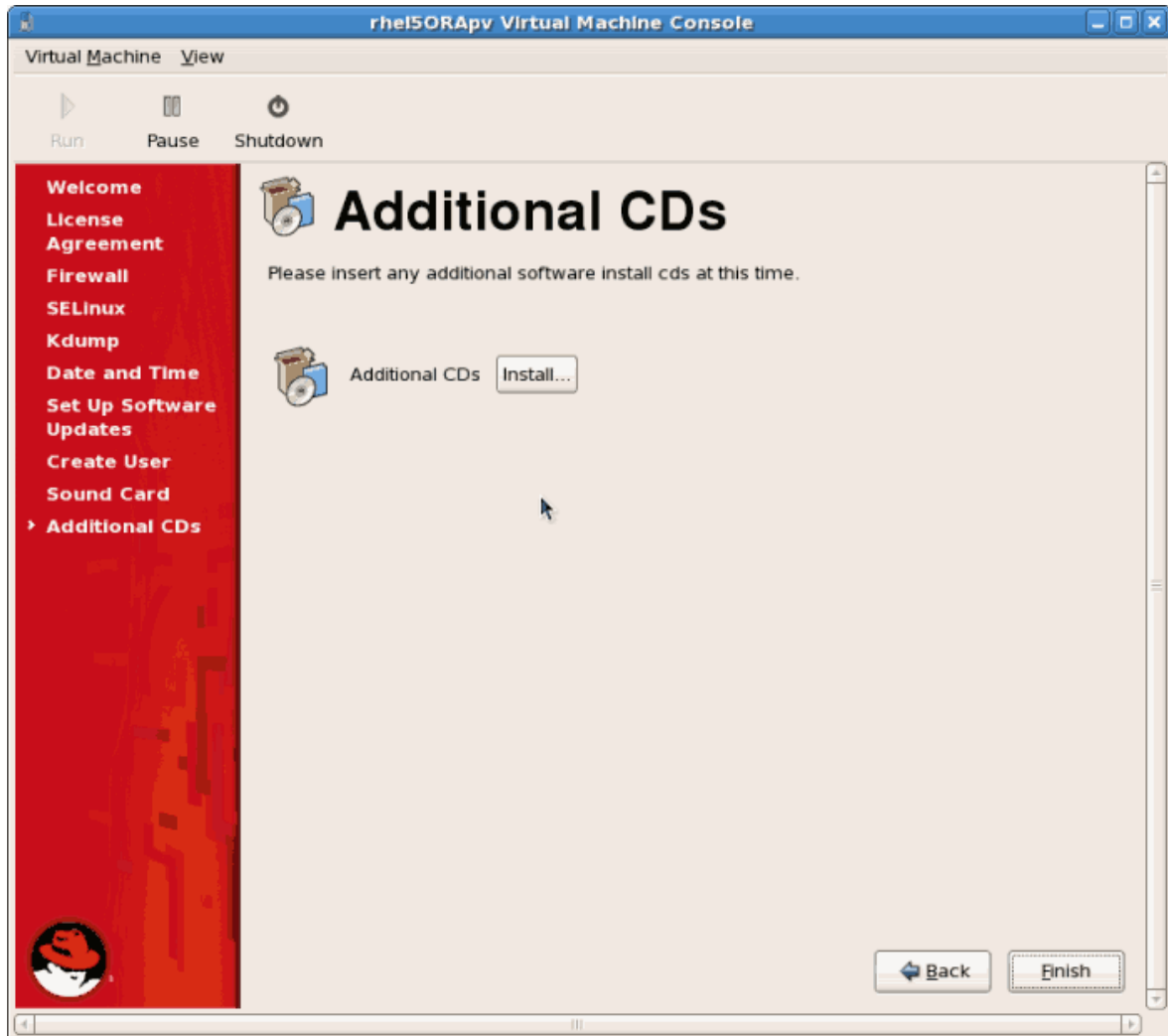


Klik op de **Forward** knop.

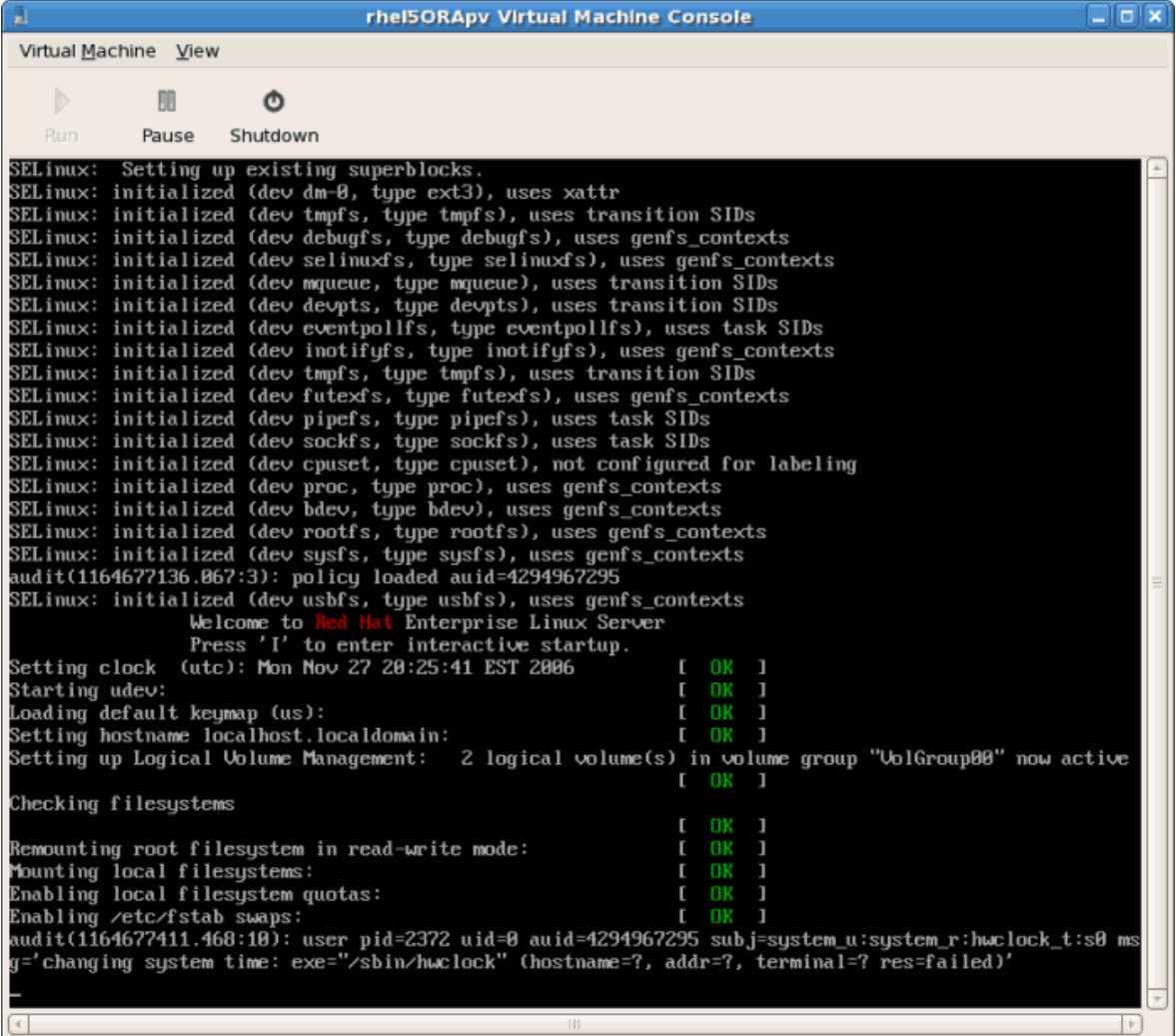
23. Als een geluid apparaat is ontdekt en je wilt geluid gebruiken, moet je het instellen. Maak dit proces af en klik op **Forward**.



24. Je kunt extra software pakketten van CD installeren met gebruik van dit scherm. Het is vaak efficiënter om op dit moment geen extra software te installeren, maar om ze later toe te voegen met gebruik van yum. Klik op **Finish**.



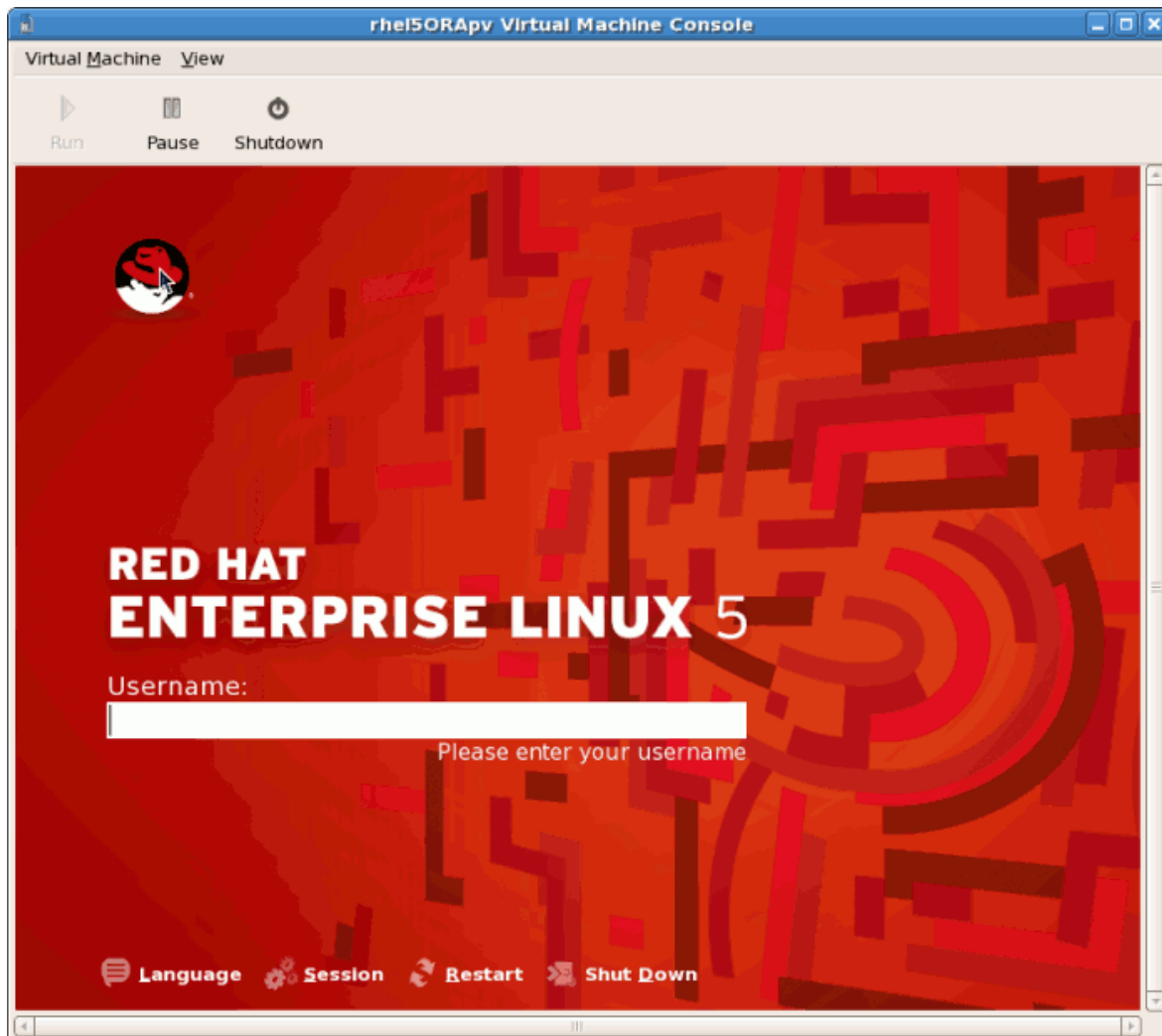
25. De guest configureert nu de instellingen die je veranderde en gaat verder met het opstart proces.



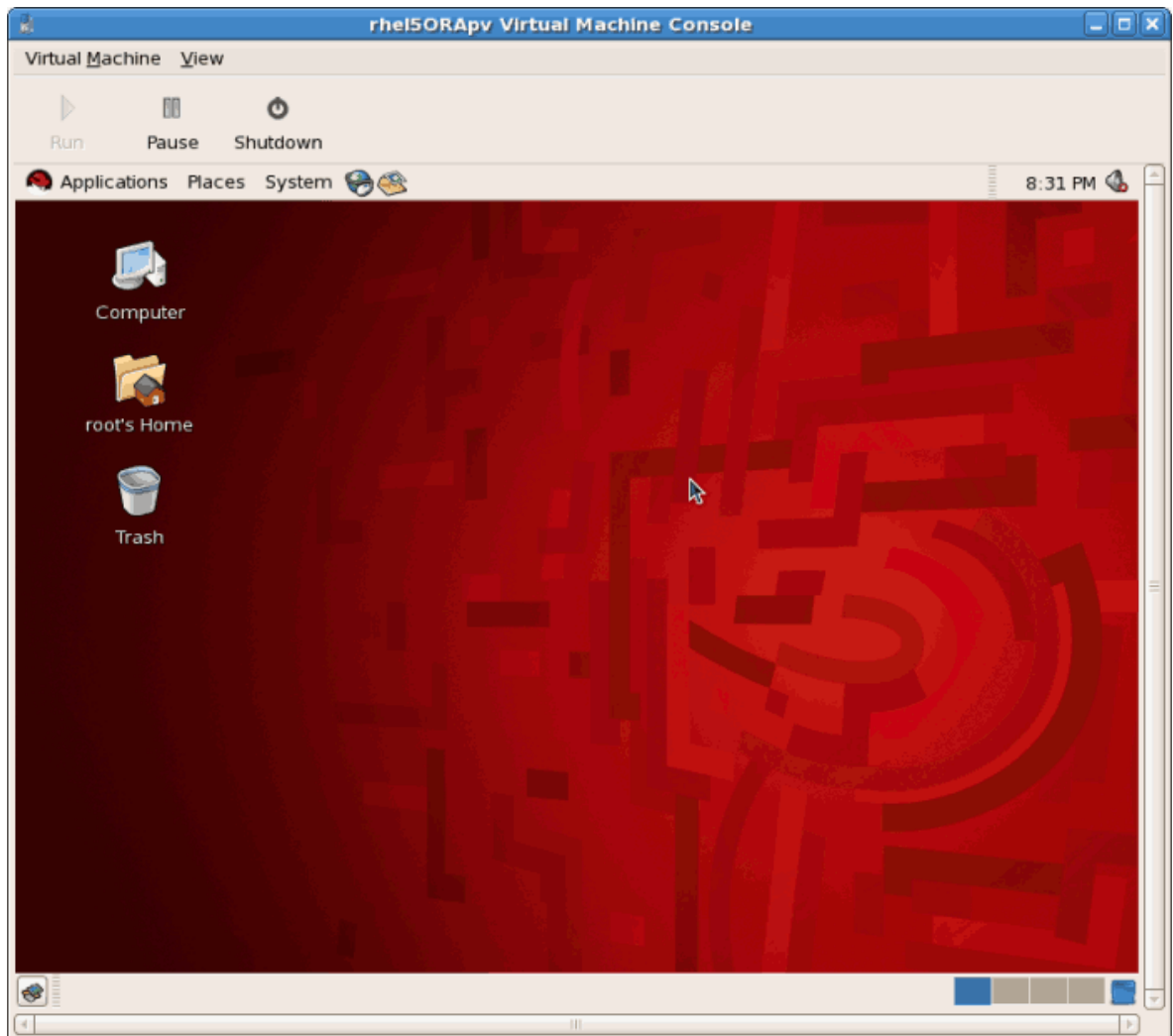
```
Virtual Machine Console
Virtual Machine View
Run Pause Shutdown

SELinux: Setting up existing superblocks.
SELinux: initialized (dev dm-0, type ext3), uses xattr
SELinux: initialized (dev tmpfs, type tmpfs), uses transition SIDs
SELinux: initialized (dev debugfs, type debugfs), uses genfs_contexts
SELinux: initialized (dev selinuxfs, type selinuxfs), uses genfs_contexts
SELinux: initialized (dev mqueue, type mqueue), uses transition SIDs
SELinux: initialized (dev devpts, type devpts), uses transition SIDs
SELinux: initialized (dev eventpollfs, type eventpollfs), uses task SIDs
SELinux: initialized (dev inotifyfs, type inotifyfs), uses genfs_contexts
SELinux: initialized (dev tmpfs, type tmpfs), uses transition SIDs
SELinux: initialized (dev futexfs, type futexfs), uses genfs_contexts
SELinux: initialized (dev pipefs, type pipefs), uses task SIDs
SELinux: initialized (dev sockfs, type sockfs), uses task SIDs
SELinux: initialized (dev cpuset, type cpuset), not configured for labeling
SELinux: initialized (dev proc, type proc), uses genfs_contexts
SELinux: initialized (dev bdev, type bdev), uses genfs_contexts
SELinux: initialized (dev rootfs, type rootfs), uses genfs_contexts
SELinux: initialized (dev sysfs, type sysfs), uses genfs_contexts
audit(1164677136.067:3): policy loaded auid=4294967295
SELinux: initialized (dev usbfs, type usbfs), uses genfs_contexts
Welcome to Red Hat Enterprise Linux Server
Press 'I' to enter interactive startup.
Setting clock (utc): Mon Nov 27 20:25:41 EST 2006 [ OK ]
Starting udev: [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname localhost.localdomain: [ OK ]
Setting up Logical Volume Management: 2 logical volume(s) in volume group "VolGroup00" now active [ OK ]
Checking filesystems [ OK ]
Remounting root filesystem in read-write mode: [ OK ]
Mounting local filesystems: [ OK ]
Enabling local filesystem quotas: [ OK ]
Enabling /etc/fstab swaps: [ OK ]
audit(1164677411.468:10): user pid=2372 uid=0 auid=4294967295 subj=system_u:system_r:hwclock_t:s0 msg='changing system time: exe="/sbin/hwclock" (hostname=?, addr=?, terminal=? res=failed)'
-
```

26. Het Red Hat Enterprise Linux 5 login scherm verschijnt. Log in met de gebruikersnaam die je in de vorige stappen aangemaakt hebt.



27. Je hebt nu met succes een para-gevirtualiseerde Red Hat Enterprise Linux 5 guest geïnstalleerd.



3.2. Red Hat Enterprise Linux installeren als een volledig gevirtualiseerde guest

Deze paragraaf behandelt het installeren van een volledig gevirtualiseerde Red Hat Enterprise Linux 5 guest.

[Procedure 3.3. Het maken van een volledig gevirtualiseerde Red Hat Enterprise Linux 5 guest met virt-manager](#)

1. **Open virt-manager**
Start **virt-manager**. Launch the **Virtual Machine Manager** application from the **Applications** menu and **System Tools** submenu. Alternatively, run the **virt-manager** command as root.
2. **Select the hypervisor**
Select the hypervisor. If installed, select Xen or KVM. For this example, select KVM. Note that presently KVM is named qemu.

Verbindt met een hypervisor als je dat nog niet gedaan hebt. Open het **File** menu en selecteer de **Add Connection...** optie. Refereer naar [Paragraaf 16.1, "Het open connection venster"](#).

Zodra een hypervisor verbinding geselecteerd is komt de **New** knop beschikbaar. Klik op de **New** knop.

3. Start the new virtual machine wizard

Pressing the **New** button starts the virtual machine creation wizard.



Press **Forward** to continue.

4. Name the virtual machine

Geef een naam op voor je gevirtualiseerde guest. Leestekens en spaties zijn niet toegestaan.



Klik op **Forward** om verder te gaan.

5. **Choose a virtualization method**

Kies de virtualisatie methode voor de gevirtualiseerde guest. Merk op dat je alleen een geïnstalleerde virtualisatie methode kunt kiezen. Als je eerder KVM of Xen selecteerde ([Stap 4](#)) moet je de hypervisor gebruiken die je selecteerde. Dit voorbeeld gebruikt de KVM hypervisor.



Druk op **Forward** om verder te gaan.

6. Select the installation method

Selecteer **Local install media** voor het installeren vanaf een optische schijf of ISO image; **Network install tree** om te installeren van een HTTP, FTP, or NFS server; of **Network boot** om te installeren van een PXE server.

Zet **OS Type** op **Linux** en **OS Variant** op **Red Hat Enterprise Linux 5** zoals getoond in de schermafdruck.



Klik op **Forward** om verder te gaan.

7. **Locate installation media**

Selecteer ISO image locatie of CD-ROM of DVD station. Dit voorbeeld gebruikt een ISO image bestand van de Red Hat Enterprise Linux 5 installatie DVD.

- Press the **Browse** button.
- Zoek de locatie van het ISO bestand op en selecteer de ISO image. Klik op **Open** om je keuze te bevestigen.
- Het bestand is geselecteerd en klaar om van te installeren.



Druk op **Forward** om verder te gaan.



Image files and SELinux

Voor ISO image bestanden en guest opslag images, gebruik je de `/var/lib/libvirt/images/` map. Elke andere locatie vereist extra configuratie voor SELinux, refereer naar *Paragraaf 7.1, "SELinux en virtualisatie"* voor details.

8. Storage setup

Ken een fysiek opslag apparaat (**Block device**) of een op bestand-gebaseerde image (**File**) toe. Op bestand-gebaseerde images moeten opgeslagen worden in de `/var/lib/libvirt/images/` map. Ken voldoende opslag toe aan jouw gevirtualiseerde guest en elke toepassing die het vereist.

Create a new virtual machine

Storage

Please indicate how you'd like to assign space from the host for your new virtual machine. This space will be used to install the virtual machine's operating system.

Block device (partition):

Location:

i **Example:** /dev/hdc2

File (disk image):

Location:

Size: MB

Allocate entire virtual disk now

⚠ Warning: If you do not allocate the entire disk now, space will be allocated as needed while the virtual machine is running. If sufficient free space is not available on the host, this may result in data corruption on the virtual machine.

i Tip: You may add additional storage, including network-mounted storage, to your virtual machine after it has been created using the same tools you would on a physical system.

Klik op **Forward** om verder te gaan.

Naar deze guest migreren

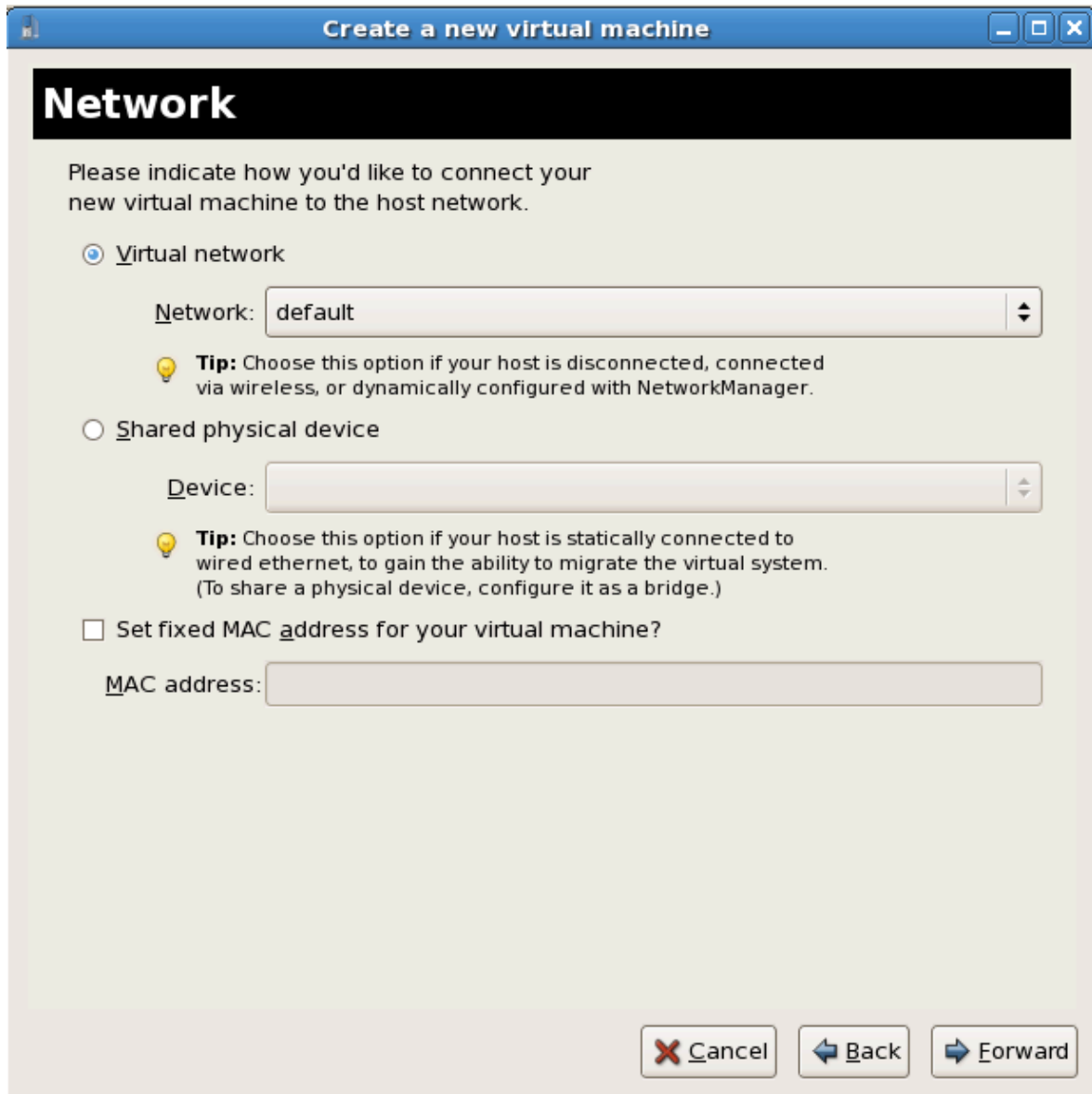
Live en off-line migraties vereisen dat guests op gedeelde netwerk opslag geïnstalleerd zijn. Voor informatie over het opzetten van gedeelde opslag voor guests refereer je naar *Hoofdstuk 5, Gedeelde opslag en virtualisatie*.

9. Network setup

Select either **Virtual network** or **Shared physical device**.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.



Press **Forward** to continue.

10. **Memory and CPU allocation**

The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

Virtualized guests require sufficient physical memory (RAM) to run efficiently and effectively. Choose a memory value which suits your guest operating system and application requirements. Windows Server 2008. Remember, guests use physical RAM. Running too many guests or leaving insufficient memory for the host system results in significant usage of virtual memory and swapping. Virtual memory is significantly slower causing degraded system performance and responsiveness. Ensure to allocate sufficient memory for all guests and the host to operate effectively.

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on

the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.

Memory:

Please enter the memory configuration for this virtual machine. You can specify the maximum amount of memory the virtual machine should be able to use, and optionally a lower amount to grab on startup. Warning: setting virtual machine memory too high will cause out-of-memory errors in your host domain!

Total memory on host machine: 2.89 GB

Max memory (MB): 1024

Startup memory (MB): 1024

CPUs:

Please enter the number of virtual CPUs this virtual machine should start up with.

Logical host CPUs: 4

Maximum virtual CPUs: 16

Virtual CPUs: 2

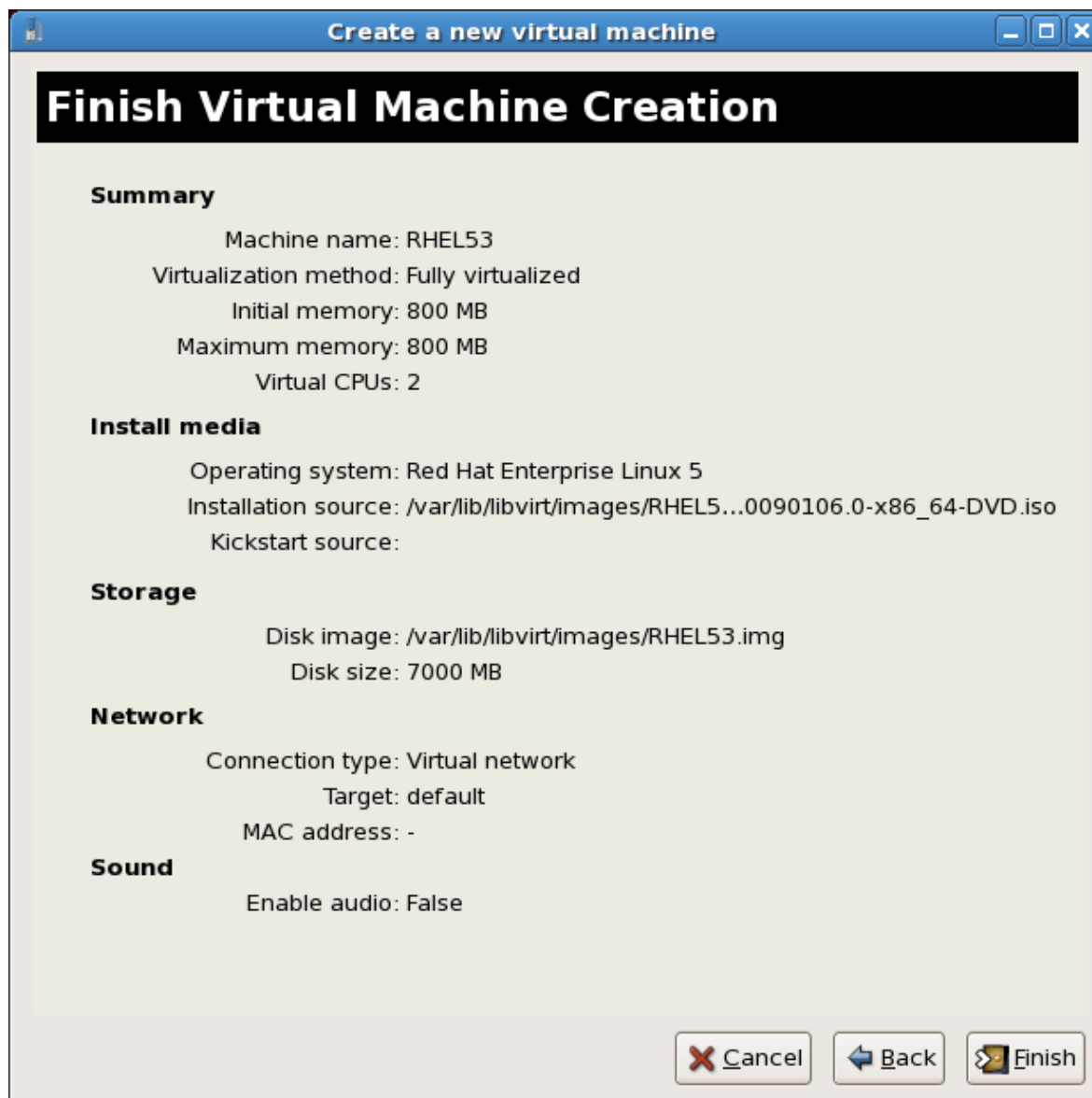
Tip: For best performance, the number of virtual CPUs should be less than (or equal to) the number of physical CPUs on the host system.

Cancel Back Forward

Press **Forward** to continue.

11. Verify and start guest installation

Controleer de configuratie.



Klik op **Finish** om de guest installatie procedure te starten.

12. Linux installeren

Maak de Red Hat Enterprise Linux 5 installatie af. De installatie wordt behandeld in de *Red Hat Enterprise Linux Installation Guide*, beschikbaar op <http://redhat.com/docs>.

Een volledig gevirtualiseerde Red Hat Enterprise Linux 5 guest is nu geïnstalleerd.

3.3. Windows XP installeren als een volledig gevirtualiseerde guest

Windows XP kan geïnstalleerd worden als een volledig gevirtualiseerde guest. Deze paragraaf beschrijft hoe Windows XP te installeren is als een volledig gevirtualiseerde guest op Linux.

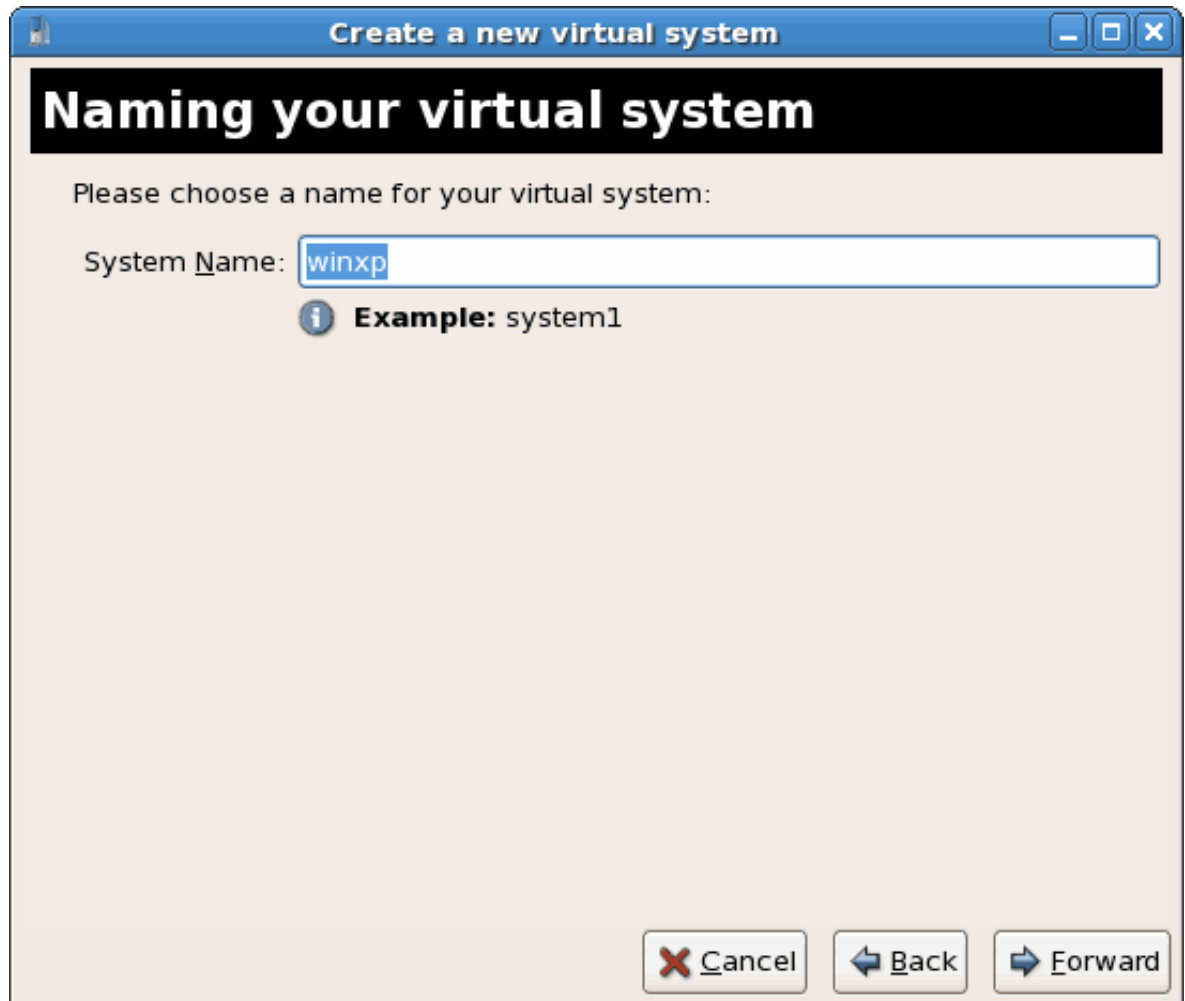
Voordat je aan deze procedure begint moet je er zeker van zijn dat je root toegang hebt.

1. **Starting virt-manager**

Open **Applications > System Tools > Virtual Machine Manager**. Open een verbinding met de host (klik op **File > Open Connection**). Klik op de **New** knop om een nieuwe virtuele machine aan te maken.

2. **Jouw systeem een naam geven**

Vul de **System Name** in en klik op de **Forward** knop.



3. **Kies een virtualisatie methode**

Als je KVM of Xen eerder selecteerde (stap [Stap 1](#)) moet je de hypervisor gebruiken die je koos. Dit voorbeeld gebruikt de KVM hypervisor.

Windows kan alleen geïnstalleerd worden met gebruik van volledige virtualisatie.



4. Kies een installatie methode

Dit scherm laat je de installatie methode en het type besturingssysteem opgeven.

Voor CD-ROM or DVD installatie selecteer je het station waarin de Windows installatie schijf zich bevindt. Als je **ISO Image Location** kiest vul het pad naar een Windows installatie .iso image in.

Selecteer **Windows** van de **OS Type** lijst en **Microsoft Windows XP** van de **OS Variant** lijst.

PXE installatie wordt niet behandeld in dit hoofdstuk.

Create a new virtual system

Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **fully virtualized** virtual system:

ISO Image Location:

ISO Location:

CD-ROM or DVD:

Path to install media:

Network PXE boot

Please choose the type of guest operating system you will be installing:

OS Type:

OS Variant:

Press **Forward** to continue.



Image files and SELinux

Voor ISO images en guest opslag images wordt het aanbevolen om de `/var/lib/libvirt/images/` map te gebruiken. Elke andere locatie kan extra configuratie voor SELinux nodig hebben, refereer naar *Paragraaf 7.1, "SELinux en virtualisatie"* voor details.

- The **Assigning storage space** window displays. Choose a disk partition, LUN or create a file based image for the guest storage.

De conventie voor bestand-gebaseerde images in Fedora is dat alle bestand-gebaseerde guest images zich bevinden in de `/var/lib/libvirt/images/` map. Andere map locaties voor bestand-gebaseerde images zijn verboden door SELinux. Als je SELinux draait in de afdwingende

mode, refereer je naar [Paragraaf 7.1, "SELinux en virtualisatie"](#) voor meer informatie over het installeren van guests.

Your guest storage image should be larger than the size of the installation, any additional packages and applications, and the size of the guests swap file. The installation process will choose the size of the guest's swap file based on size of the RAM allocated to the guest.

Allocate extra space if the guest needs additional space for applications or other data. For example, web servers require additional space for log files.



Choose the appropriate size for the guest on your selected storage type and click the **Forward** button.



Opmerking

Het wordt aanbevolen dat je de standaard map gebruikt voor virtuele machine images, `/var/lib/libvirt/images/`. Als je een andere locatie gebruikt (zoals `/images/` in dit voorbeeld) wees er dan zeker van dat het toegevoegd wordt aan

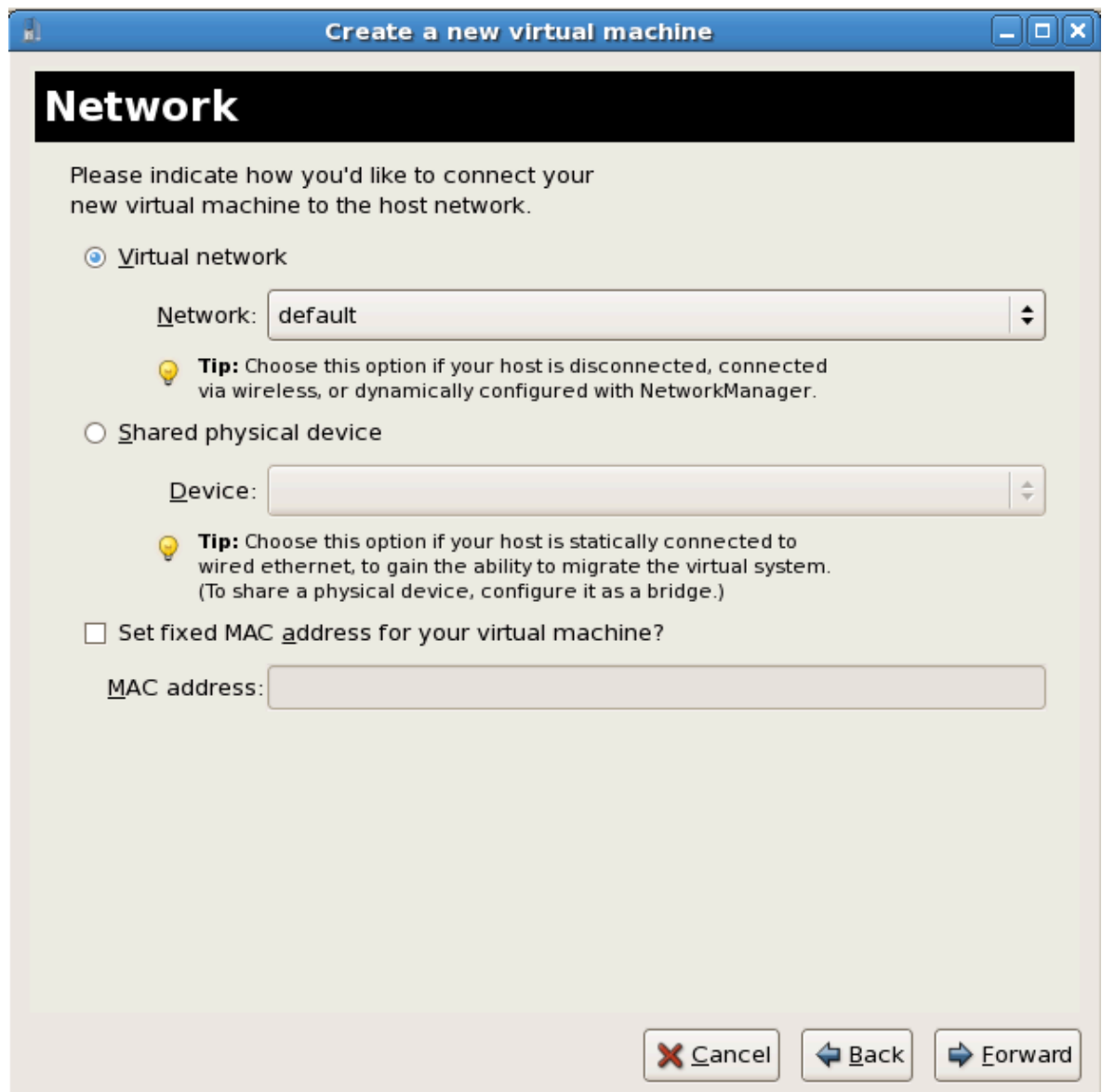
de SELinux gedragslijnen en opnieuw gelabeld wordt voordat je verder gaat met de installatie (later in dit document zul je informatie vinden over het veranderen van jouw SELinux gedragslijnen)

6. Network setup

Select either **Virtual network** or **Shared physical device**.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.



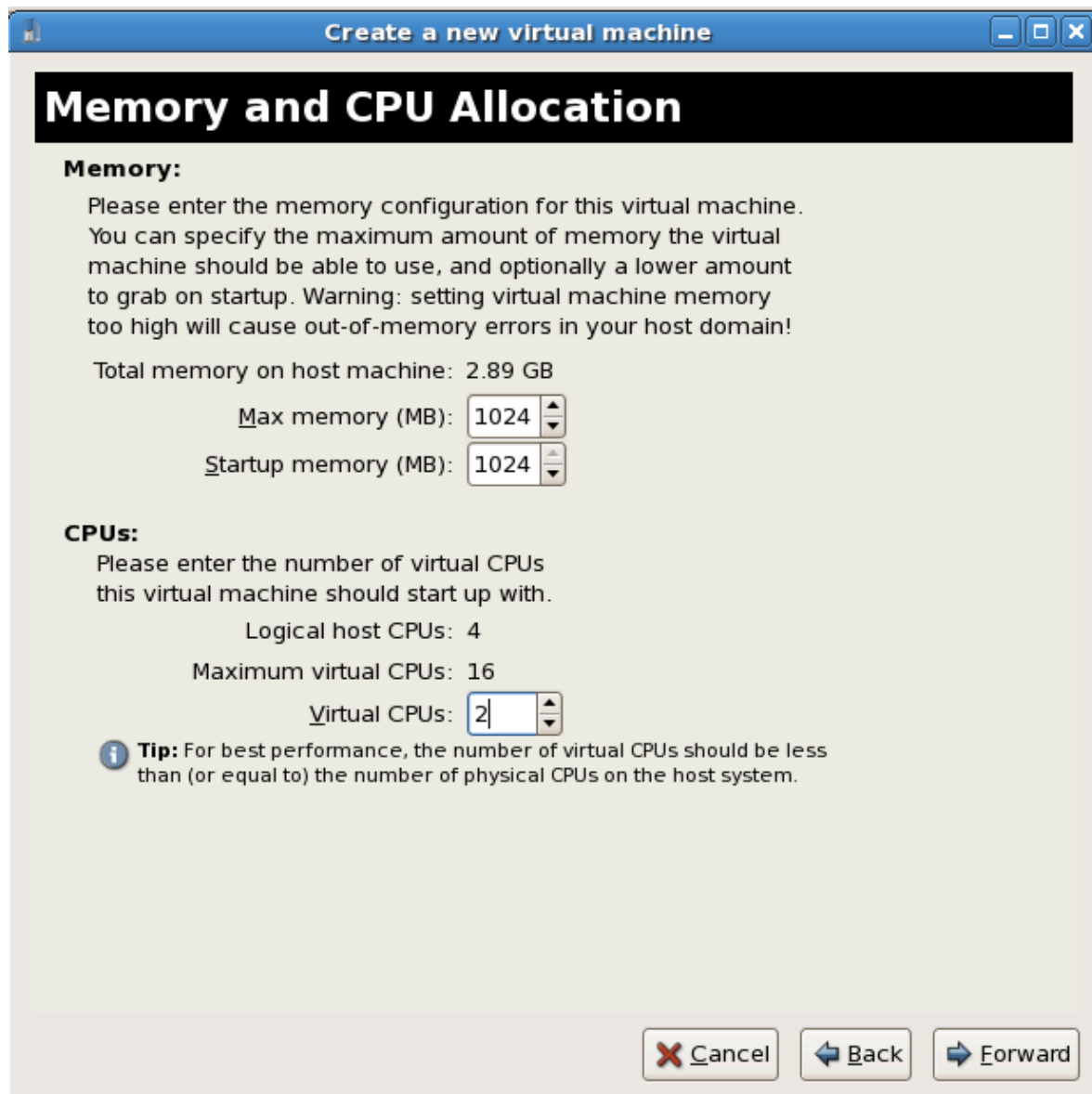
The screenshot shows a window titled "Create a new virtual machine" with a "Network" sub-header. The main text asks the user to indicate how to connect the new virtual machine to the host network. There are two radio button options: "Virtual network" (selected) and "Shared physical device". The "Virtual network" option has a "Network:" dropdown menu set to "default" and a tip: "Choose this option if your host is disconnected, connected via wireless, or dynamically configured with NetworkManager." The "Shared physical device" option has a "Device:" dropdown menu and a tip: "Choose this option if your host is statically connected to wired ethernet, to gain the ability to migrate the virtual system. (To share a physical device, configure it as a bridge.)" There is also a checkbox for "Set fixed MAC address for your virtual machine?" with an empty "MAC address:" text box below it. At the bottom right, there are three buttons: "Cancel", "Back", and "Forward".

Press **Forward** to continue.

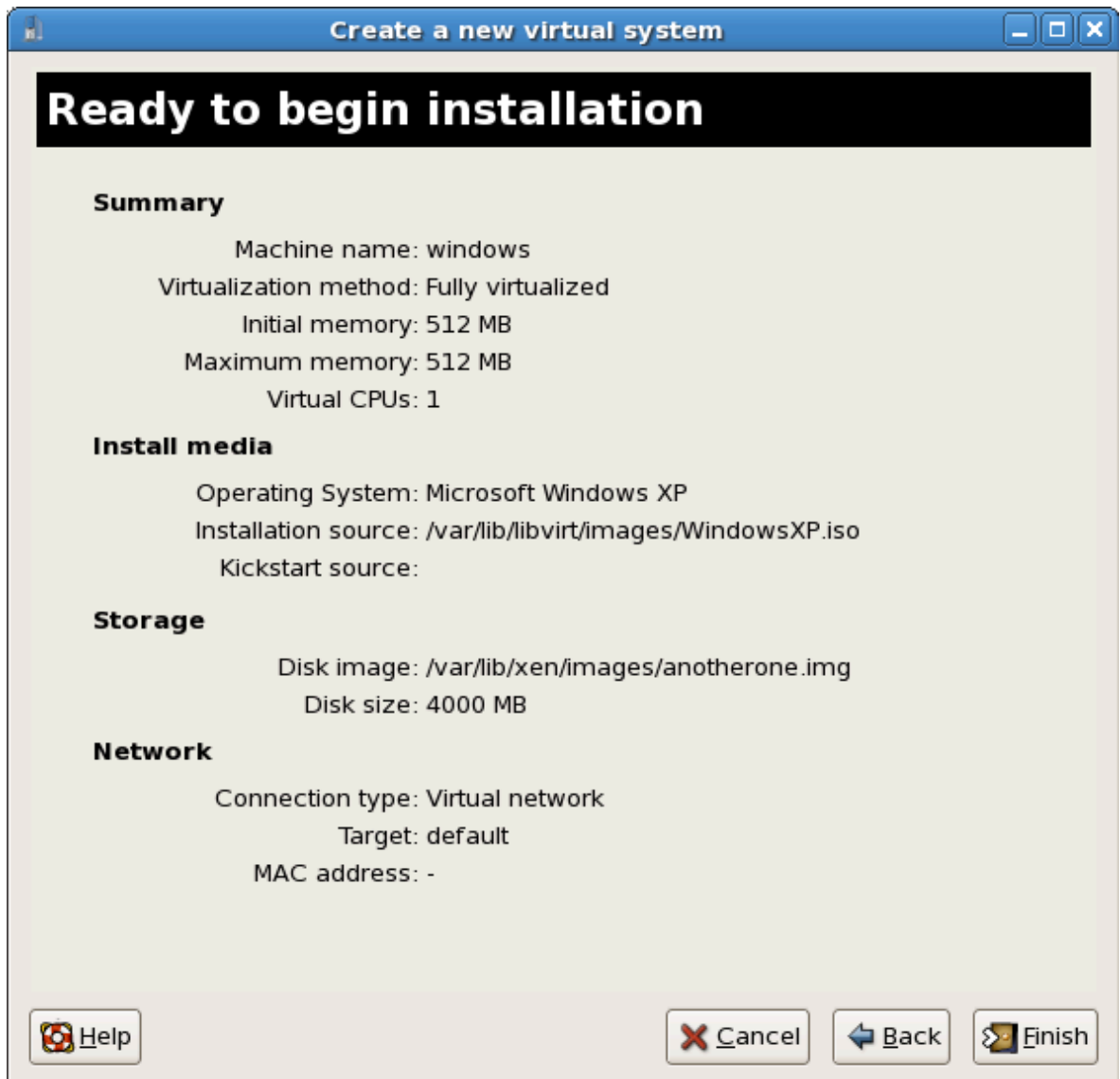
7. The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

Gevirtualiseerde guest vereisen voldoende fysiek geheugen (RAM) om efficiënt en effectief te draaien. Kies een geheugen waarde die past bij je guest besturingssysteem en toepassing behoeften. De meeste besturingssystemen vereisen ten minste 512 MB RAM om snel te kunnen werken. Denk eraan dat guest fysieke RAM gebruiken. Het draaien van te veel guests of onvoldoende geheugen overlaten voor het host systeem resulteert in significant gebruik van virtueel geheugen en swappen. Virtueel geheugen is veel langzamer wat afnemende systeem prestaties en snelheid veroorzaakt. Wees er zeker van om voldoende geheugen toe te kennen aan alle guests en de host om effectief te kunnen werken.

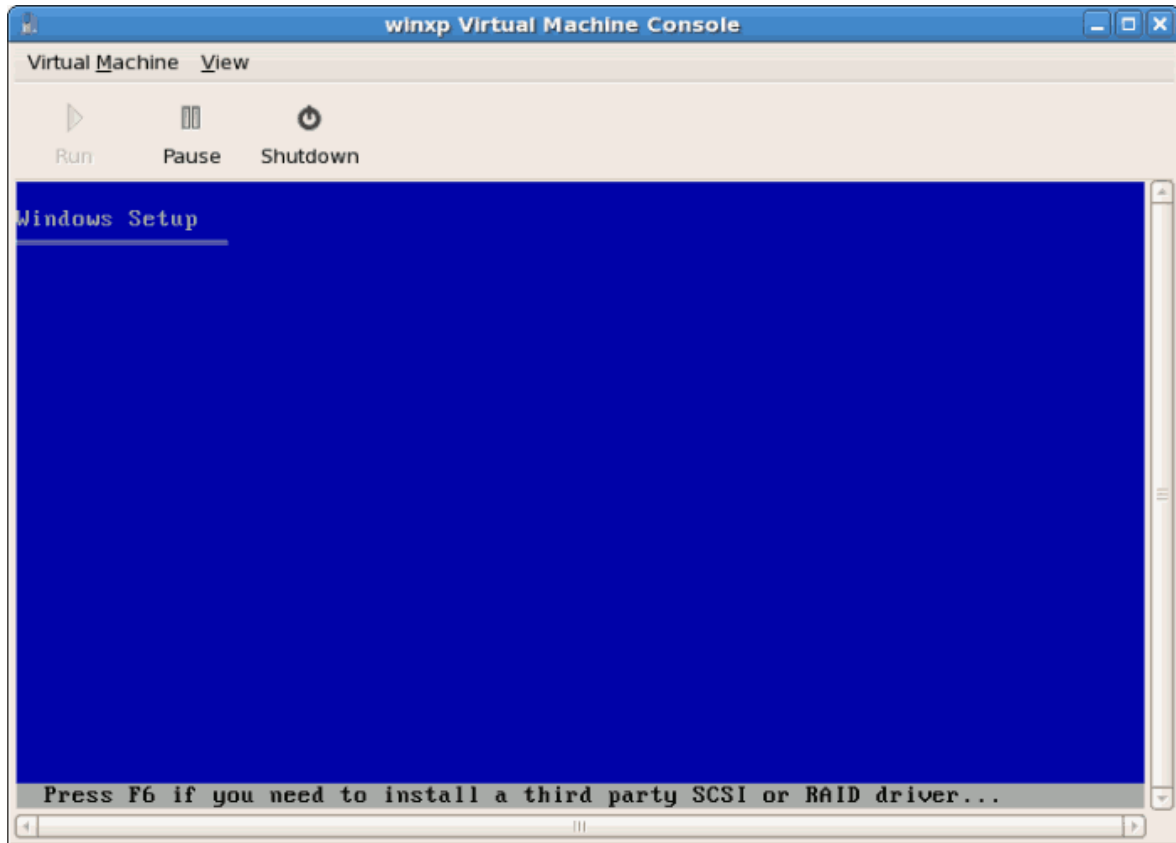
Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



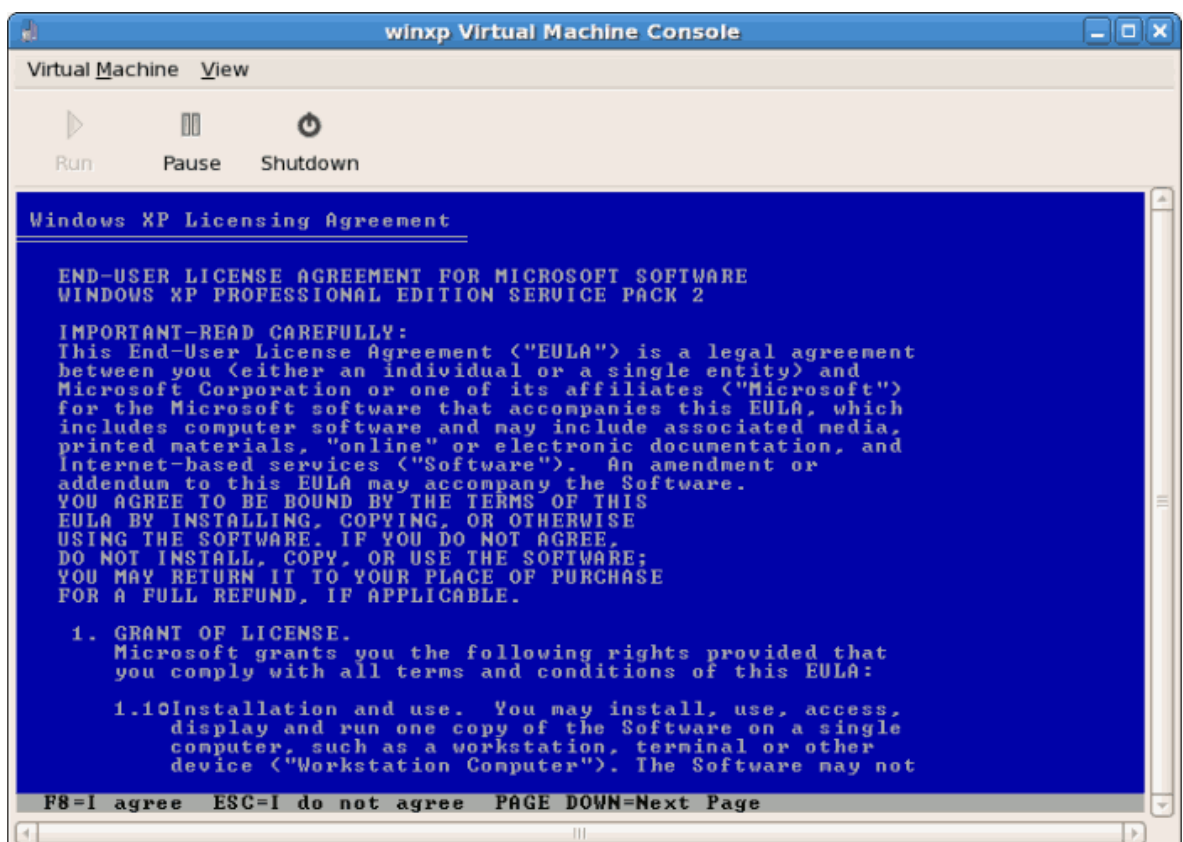
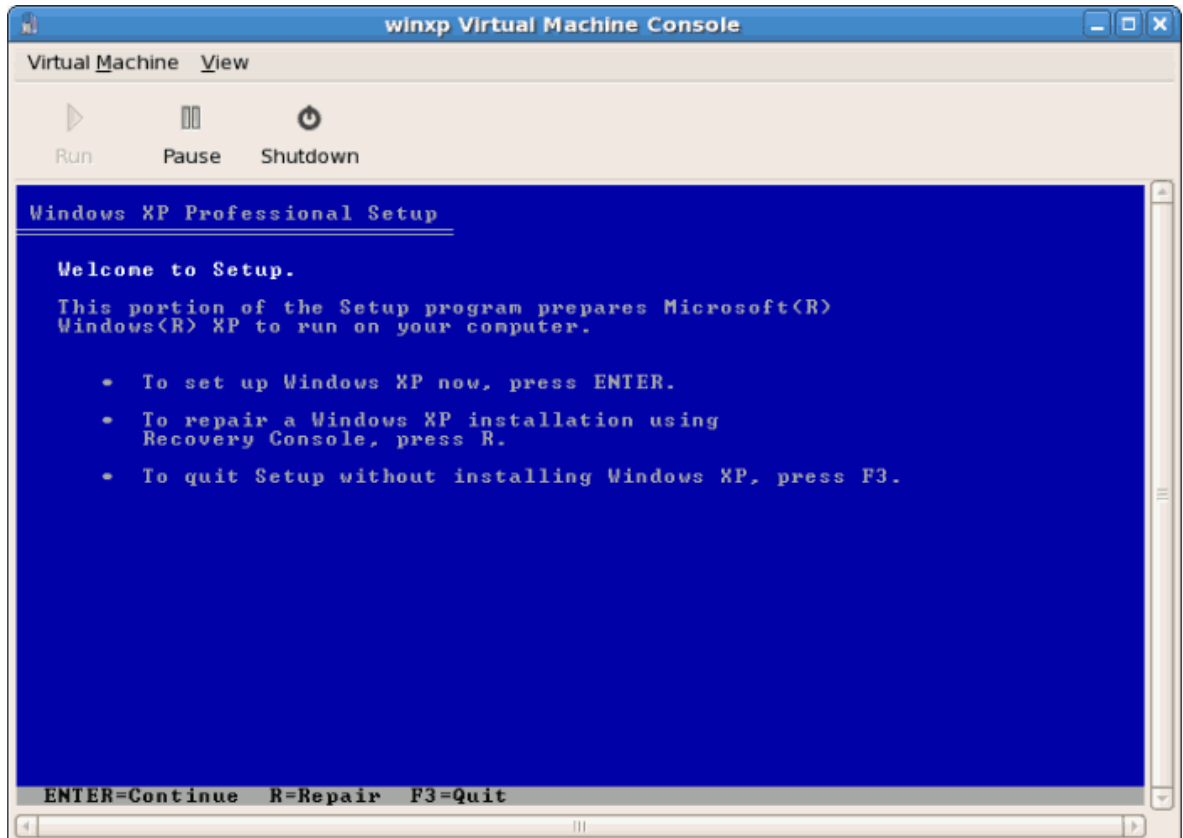
8. Voordat de installatie verdergaat zul je het overzicht scherm zien. Klik op **Finish** om verder te gaan met de guest installatie:



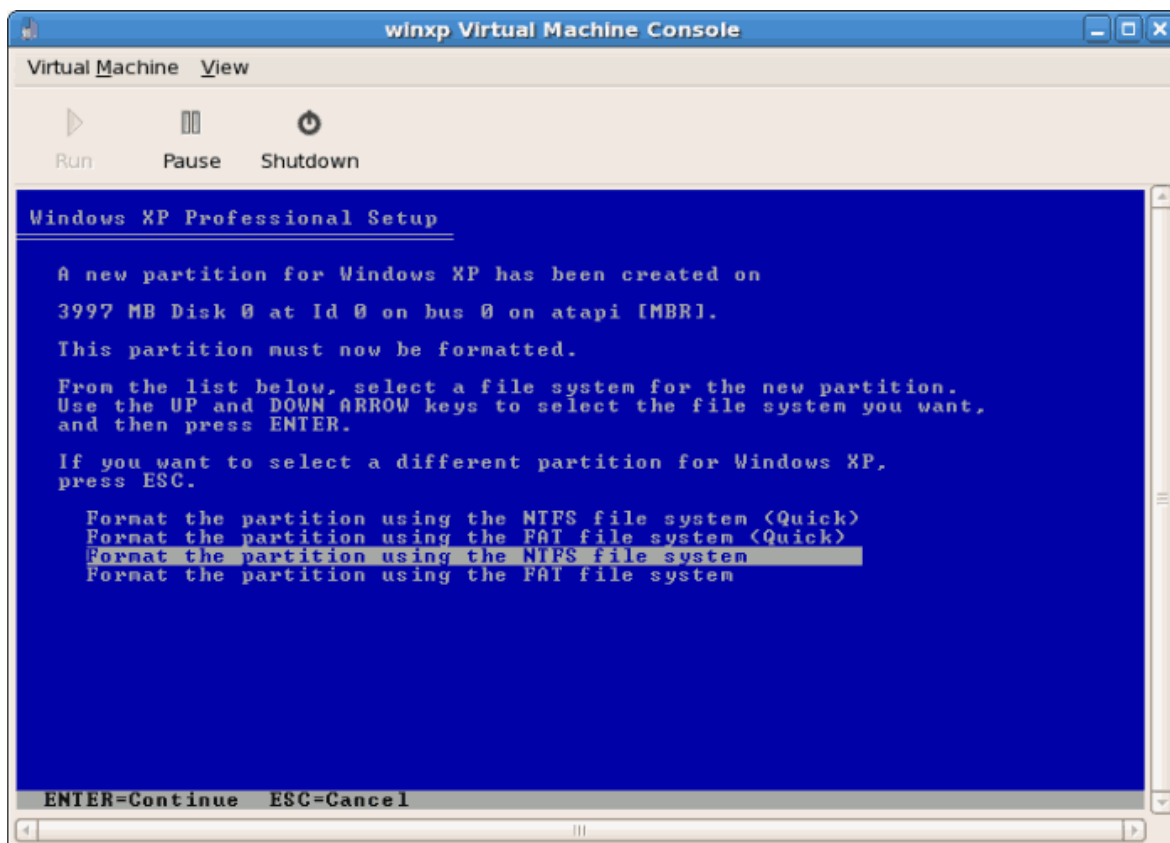
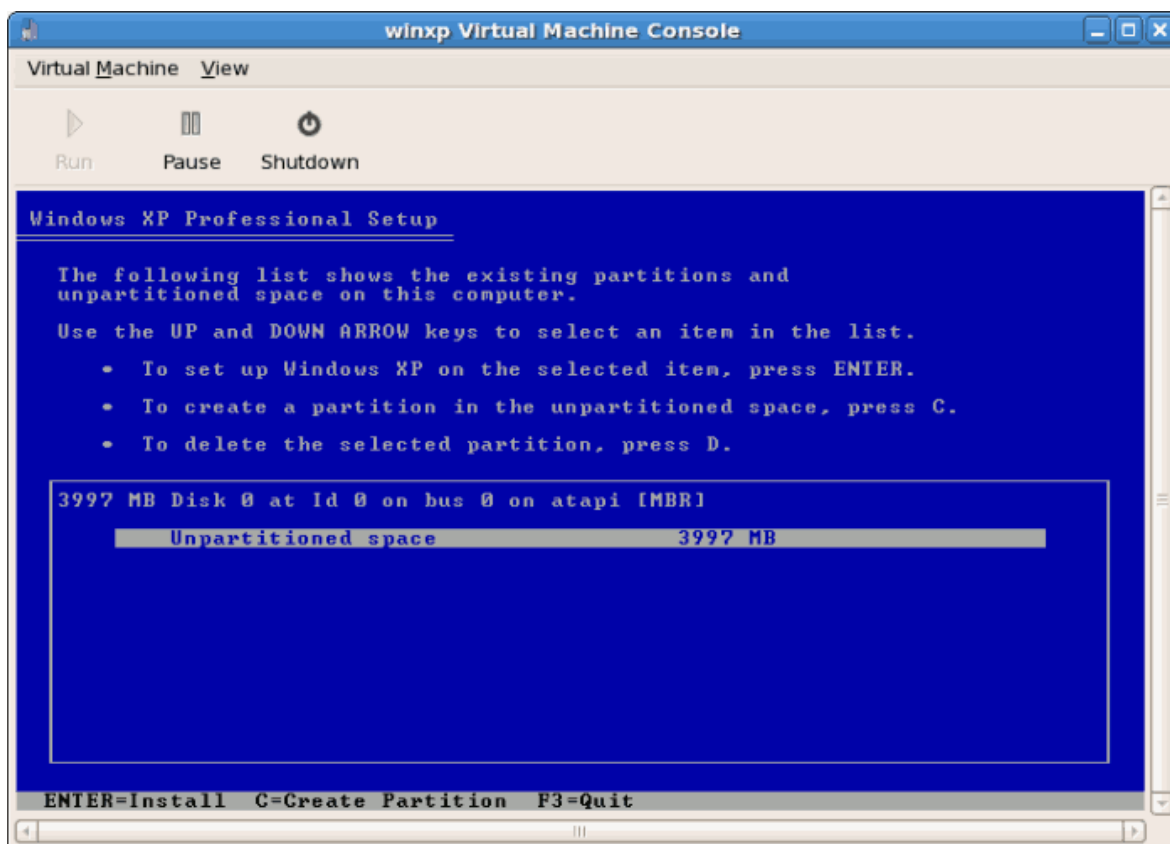
9. Je moet een hardware selectie maken dus open een console venster meteen nadat de installatie start. Klik op de **Finish** knop en ga dan naar het **virt-manager** overzicht scherm en selecteer je pas opgestarte Windows guest. Dubbel-klik op de systeem naam en het console venster opent. Duw snel en herhaald op **F5** om een nieuwe HAL te selecteren, zodra je het dialoog scherm in de Windows installatie krijgt, selecteer je de 'Generic i486 Platform' tab (blader door de selecties met de **Up** en **Down** pijlen).



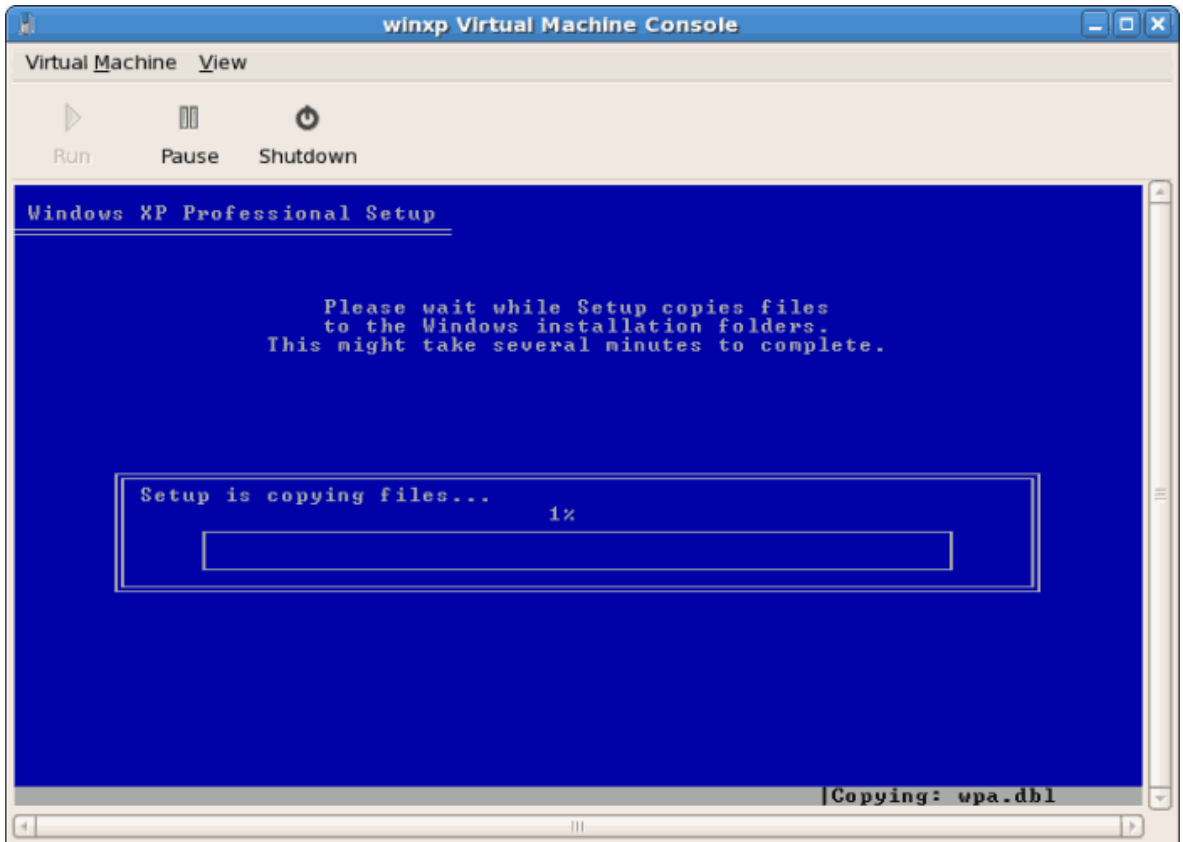
10. De installatie vervolgt met de standaard Windows installatie.

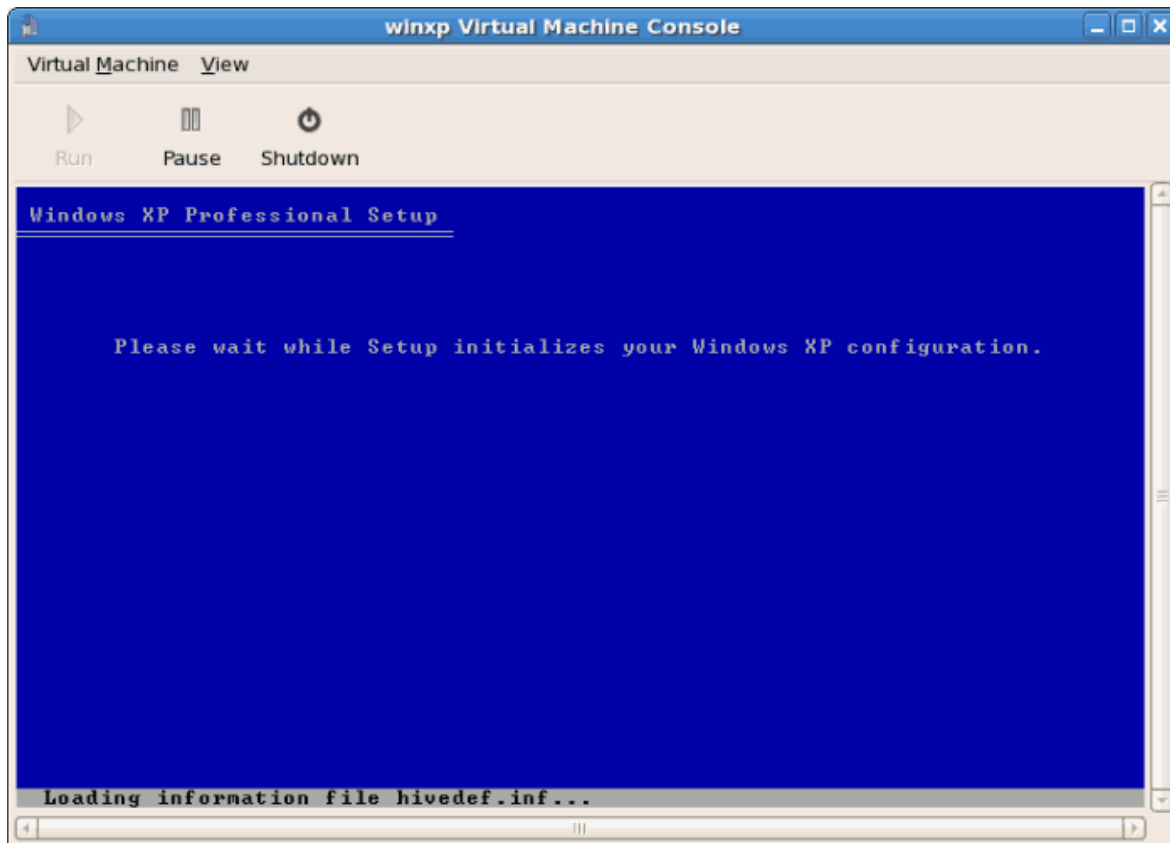


11. Partioneer de harde schijf als er om gevraagd wordt.

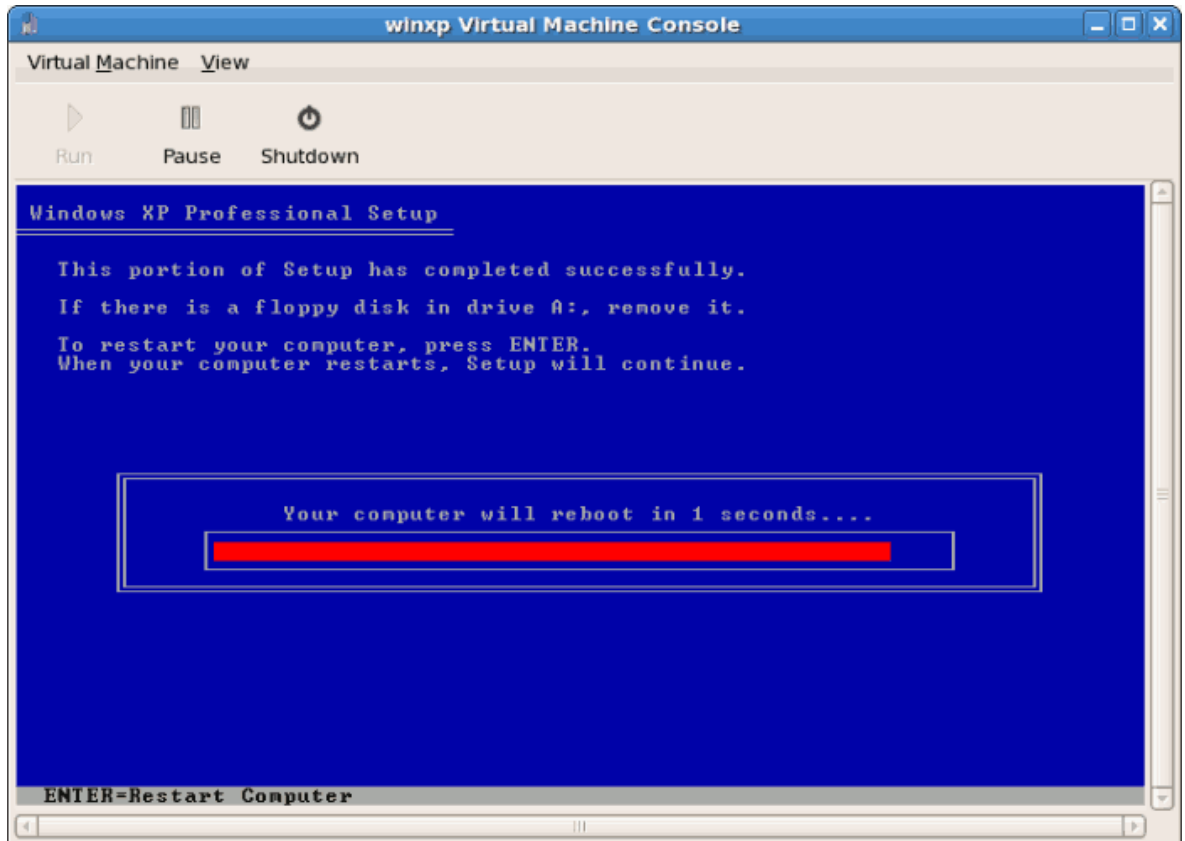


12. Nadat de schijf geformatteerd is begint Windows met het kopiëren van de bestanden naar de harde schijf.





13. De bestanden worden gekopieerd naar het opslag apparaat, Windows gaat nu opnieuw opstarten.

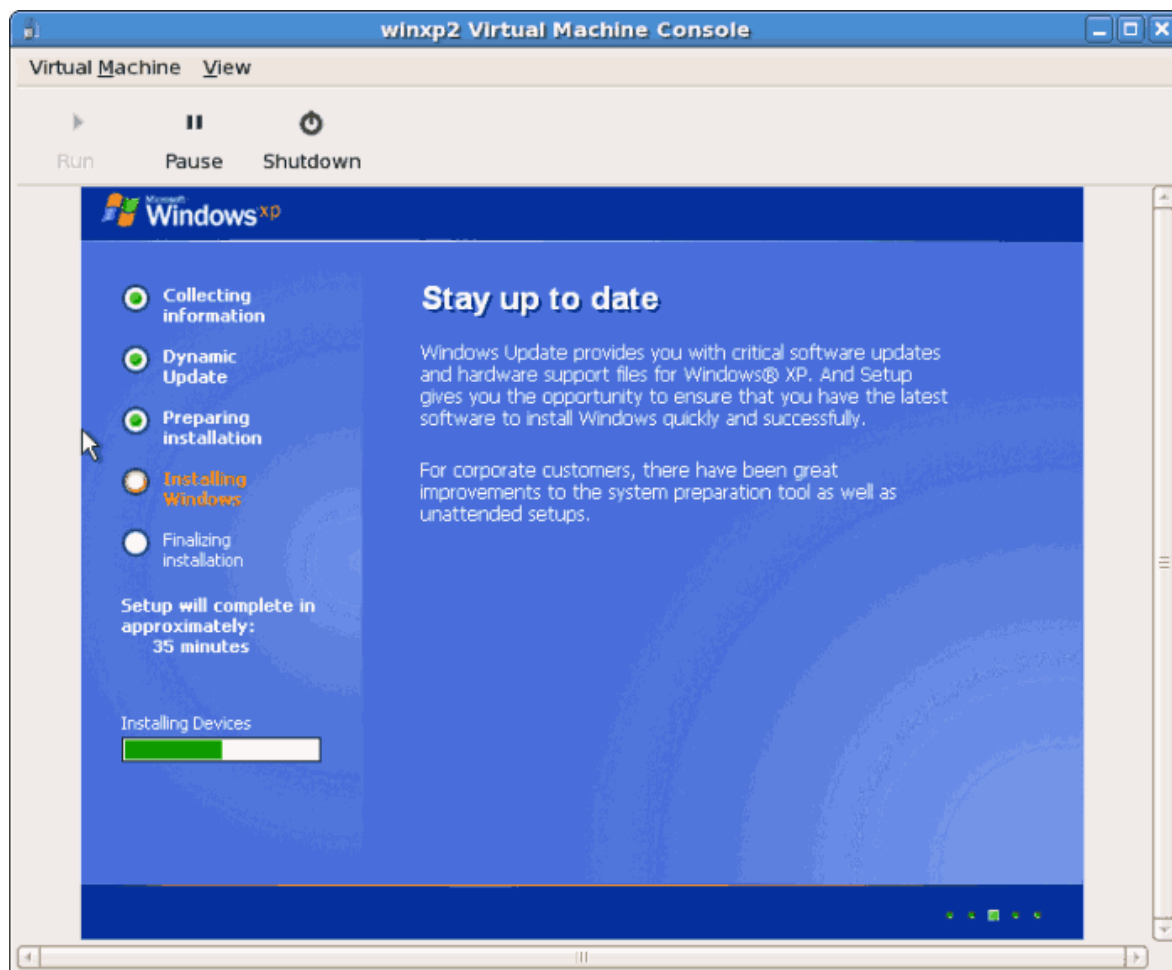


14. Start je Windows guest opnieuw op:

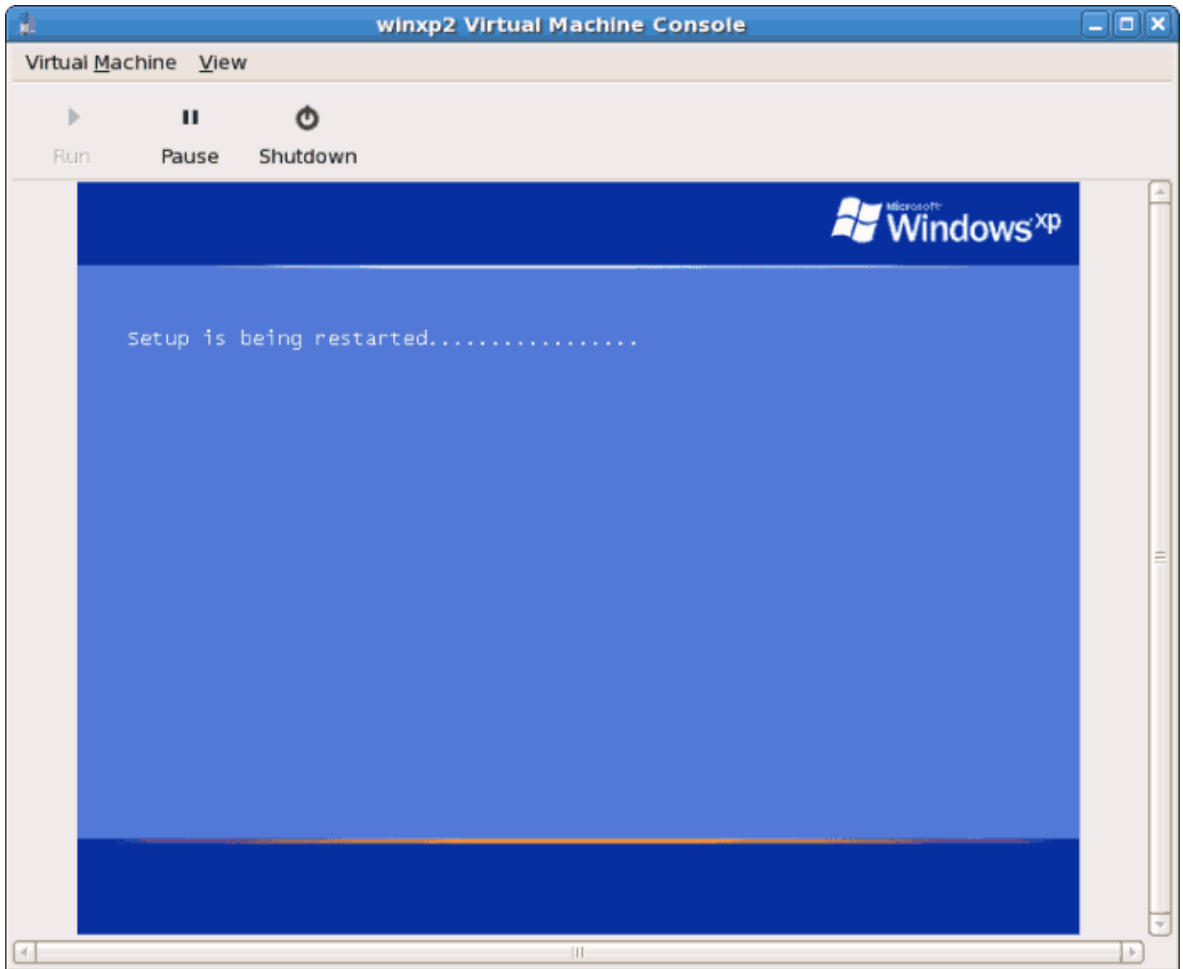
```
# virsh start WindowsGuest
```

Waarin *WindowsGuest* de naam van je virtuele machine is.

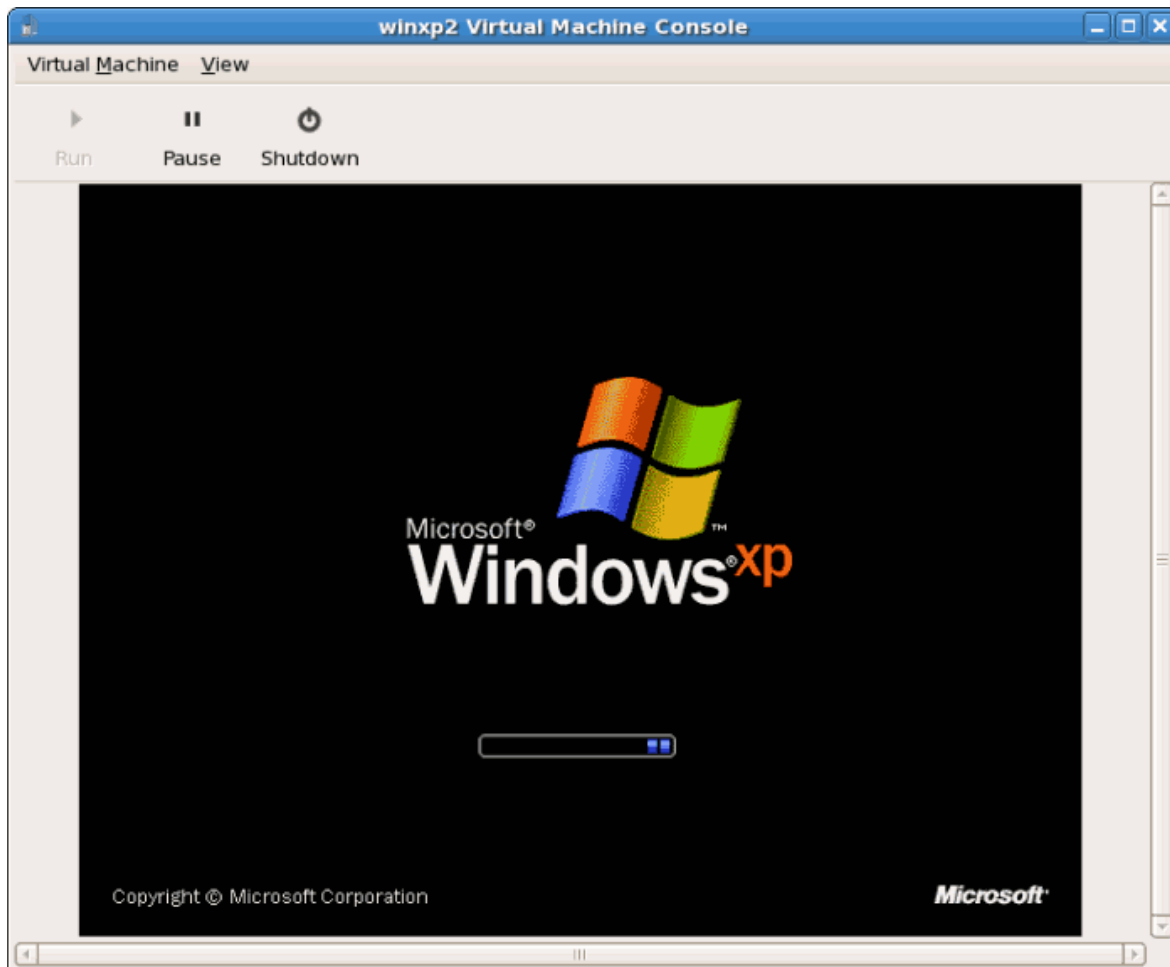
15. Als het console venster opent, zie je de instel fase van de Windows installatie.



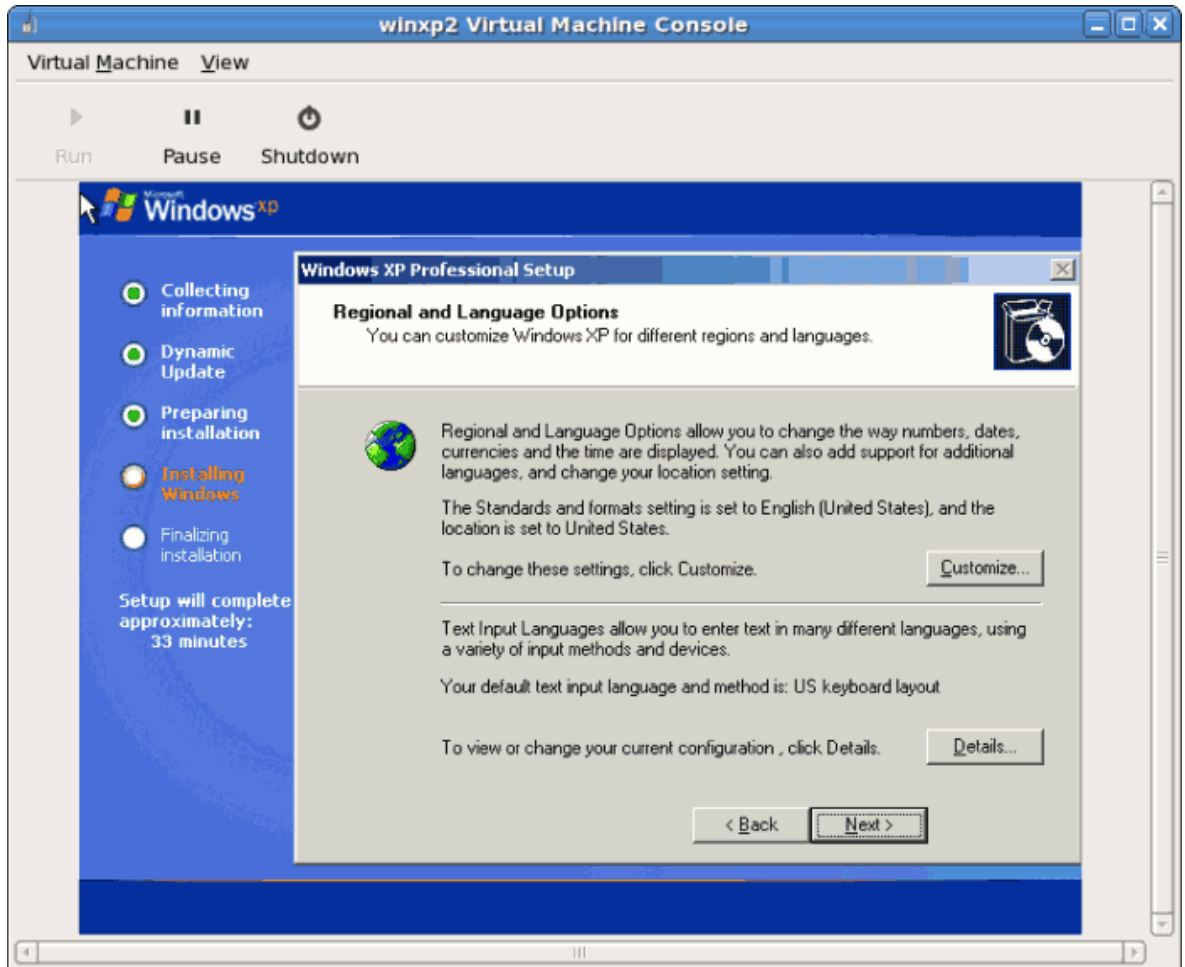
16. Als het lijkt dat je installatie vastloopt tijdens de instel fase, start dan de guest opnieuw op met **virsh reboot *WindowsGuestName***. Hiermee zal de installatie meestal verdergaan. Als je de virtuele machine opnieuw opstart zul een Setup is being restarted boodschap zien:



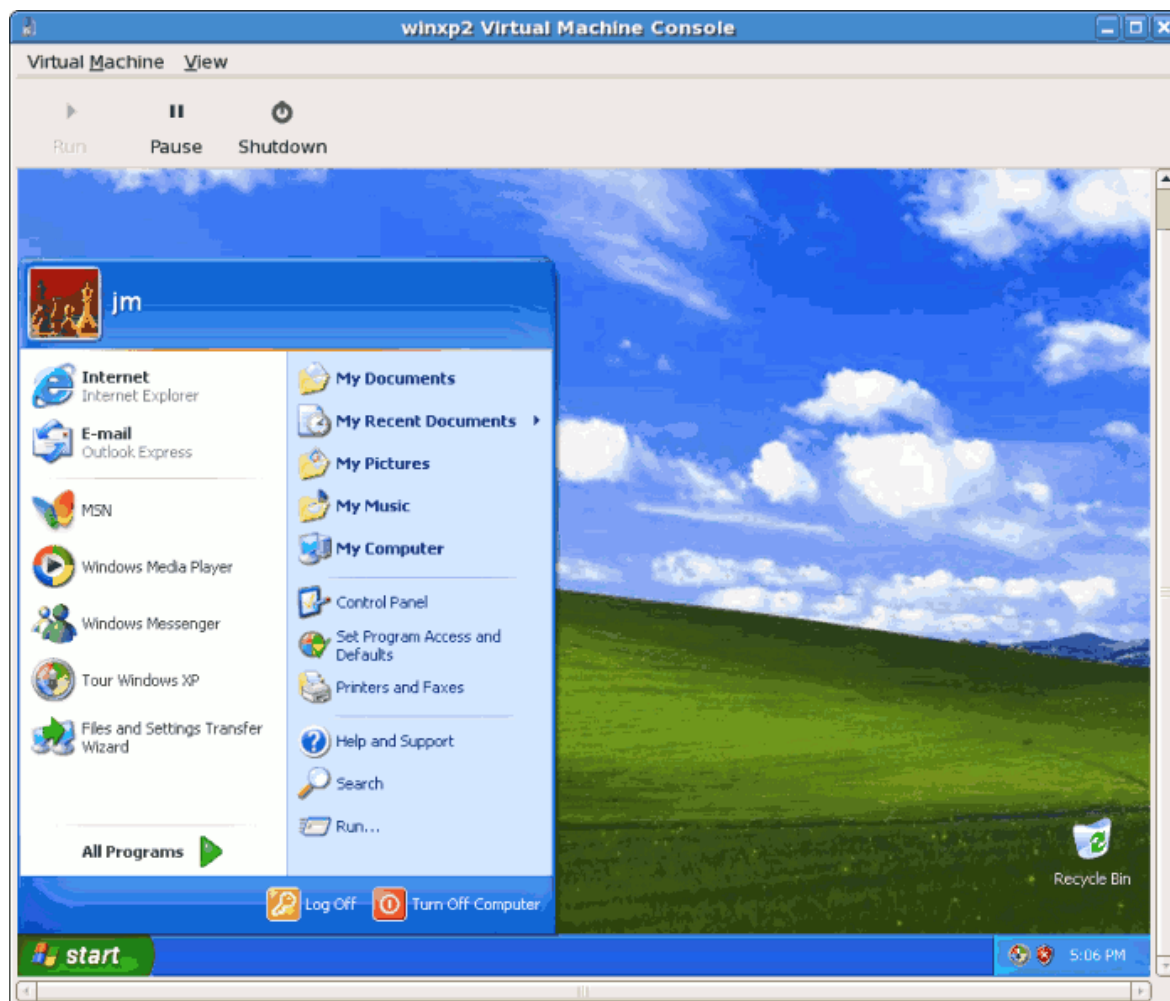
17. Nadat setup is beëindigd, zul je het Windows boot scherm zien:



18. Je kunt nu verder gaan met de standaard setup van jouw Windows installatie:



19. Het setup proces is klaar, er verschijnt een Windows bureaublad.



3.4. Windows Server 2003 installeren als een volledig gevirtualiseerde guest

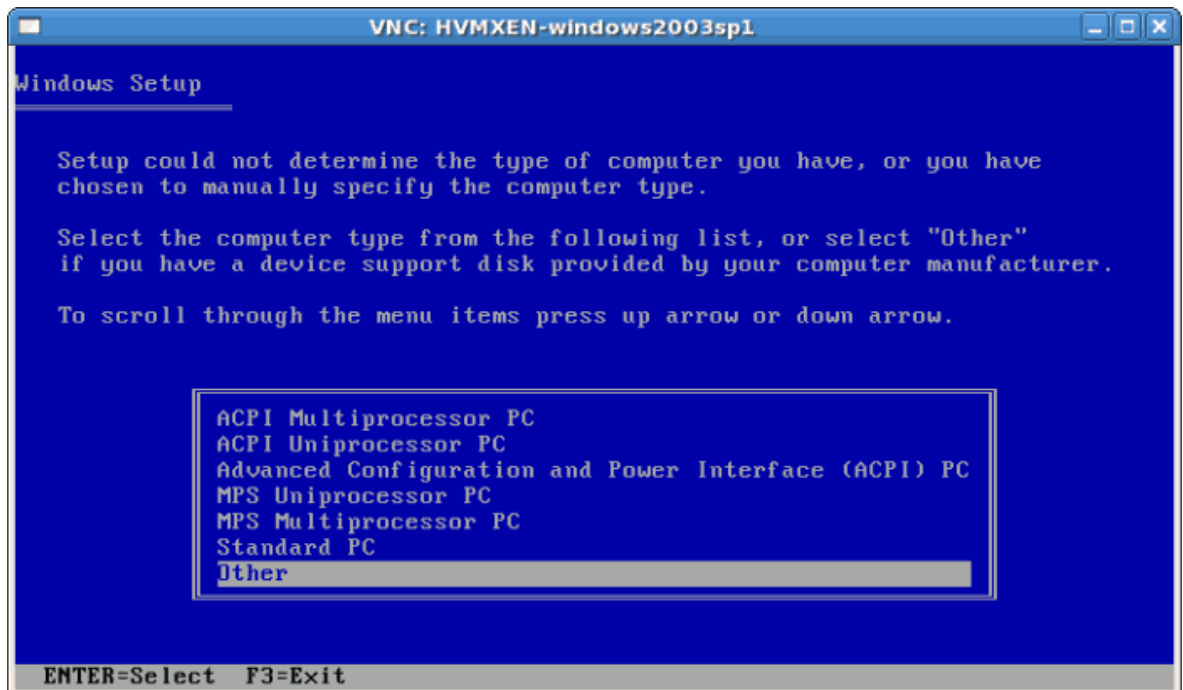
Dit hoofdstuk beschrijft de installatie van een volledig gevirtualiseerde Windows Server 2003 guest met het **virt-install** commando. **virt-install** kan gebruikt worden in plaats van virt-manager. Dit proces is vergelijkbaar met de Windows XP installatie behandeld in [Paragraaf 3.3, "Windows XP installeren als een volledig gevirtualiseerde guest"](#).

1. Het gebruiken van **virt-install** voor het installeren van Windows Server 2003 als de console voor de Windows guest opent snel het virt-viewer venster. Een voorbeeld van het gebruik van **virt-install** voor het installeren van een Windows Server 2003 guest:

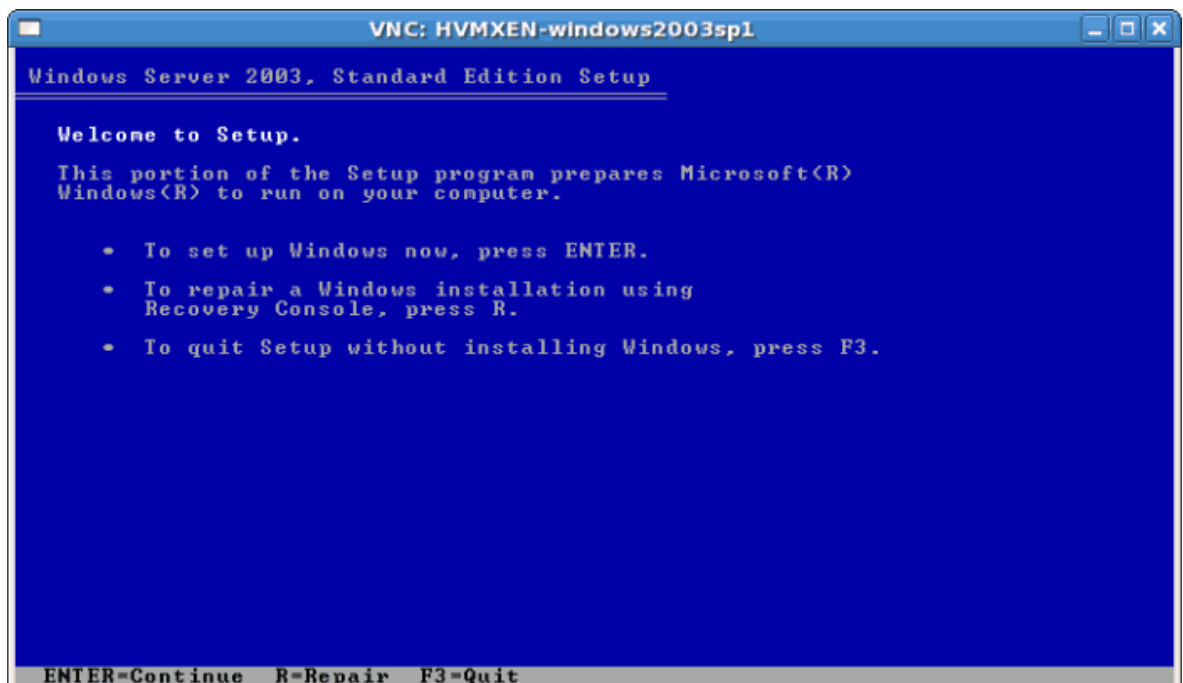
Start de installatie met het **virt-install** commando.

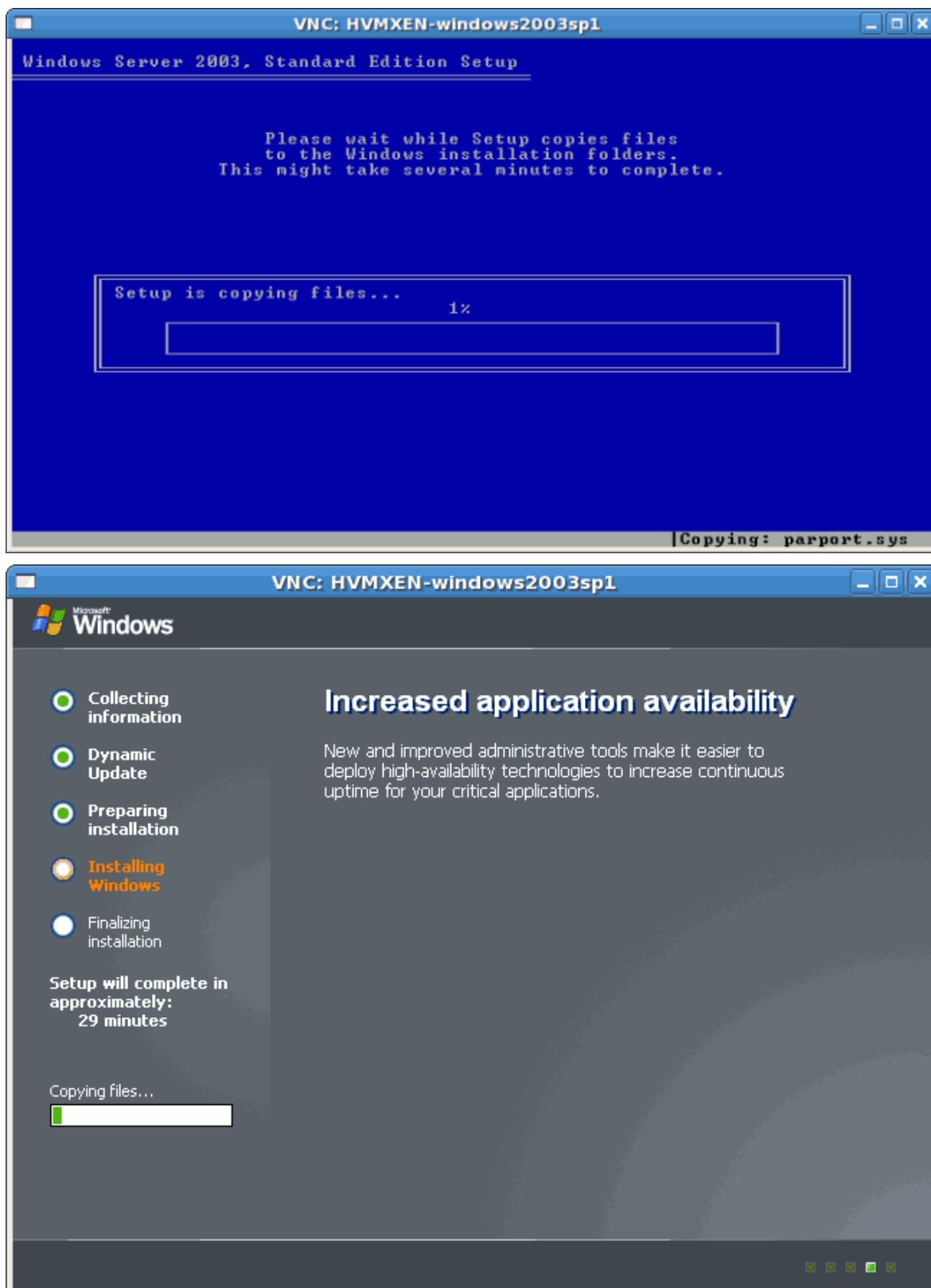
```
# virt-install -hvm -s 5 -f /var/lib/libvirt/images/windows2003spi1.dsk \
-n windows2003sp1 -cdrom=/ISOs/WIN/en_windows_server_2003_sp1.iso \
-vnc -r 1024
```

- Zodra de guest opstart in de installatie moet je snel **F5** induwen. Als je **F5** niet op het juiste moment induwt moet je de installatie opnieuw opstarten. Het induwen van **F5** laat je een andere **HAL** of **Computer Type** selecteren. Kies Standard PC als het Computer Type. Dit is de enigste niet-standaard stap die vereist is.



- Maak de rest van de installatie af.





4. Windows Server 2003 is nu geïnstalleerd als een volledig gevirtualiseerde guest.

3.5. Windows Server 2008 installeren als een volledig gevirtualiseerde guest

Deze paragraaf behandelt het installeren van een volledig gevirtualiseerde Windows Server 2008 guest.

Procedure 3.4. Windows Server 2008 installeren met virt-manager

1. **Open virt-manager**

Start **virt-manager**. Launch the **Virtual Machine Manager** application from the **Applications** menu and **System Tools** submenu. Alternatively, run the **virt-manager** command as root.

2. **Select the hypervisor**

Select the hypervisor. If installed, select Xen or KVM. For this example, select KVM. Note that presently KVM is named qemu.

Zodra de optie geselecteerd is, komt de **New** knop beschikbaar. Klik op de **New** knop.

3. **Start the new virtual machine wizard**

Pressing the **New** button starts the virtual machine creation wizard.



Press **Forward** to continue.

4. **Name the virtual machine**

Geef een naam op voor jouw gevirtualiseerde guest. Leestekens en spaties zijn niet toegelaten.



Klik op **Forward** om verder te gaan.

5. **Choose a virtualization method**

Kies de virtualisatie methode voor de gevirtualiseerde guest. Merk op dat je alleen een geïnstalleerde virtualisatie methode kan kiezen. Als je KVM of Xen eerder selecteerde (stap 2) moet je de hypervisor gebruiken die je selecteerde. Dit voorbeeld gebruikt de KVM hypervisor.



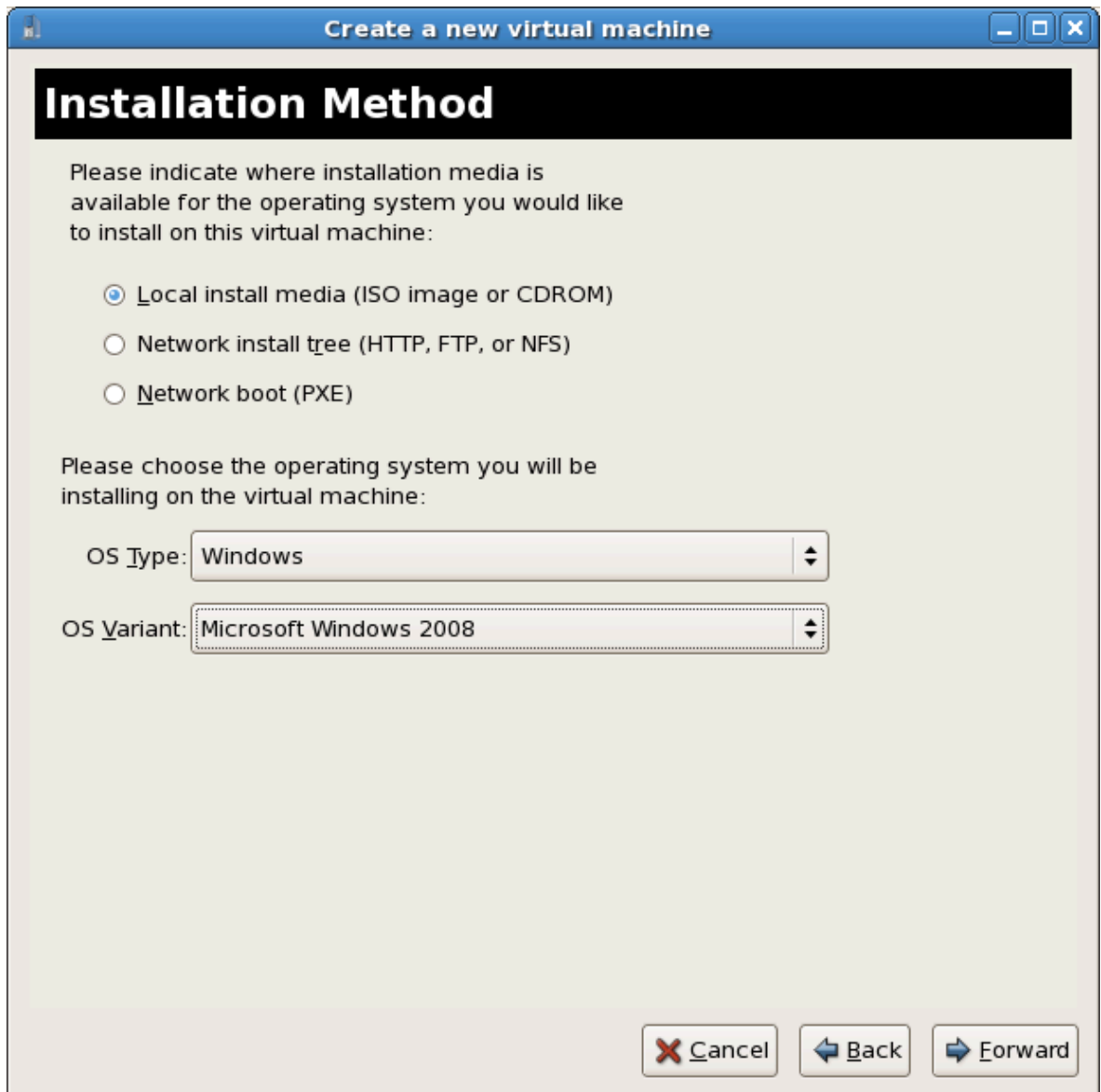
Klik op **Forward** om verder te gaan.

6. Select the installation method

Voor alle versies van Windows moet je **local install media** gebruiken, met of een ISO image, of een fysieke optische media.

PXE kan gebruikt worden als je een PXE server hebt die ingesteld is voor Windows netwerk installatie. PXE Windows installatie wordt niet behandeld in deze gids.

Stel **OS Type** in op **Windows** en **OS Variant** op **Microsoft Windows 2008** zoals getoond in de schermafdruck.

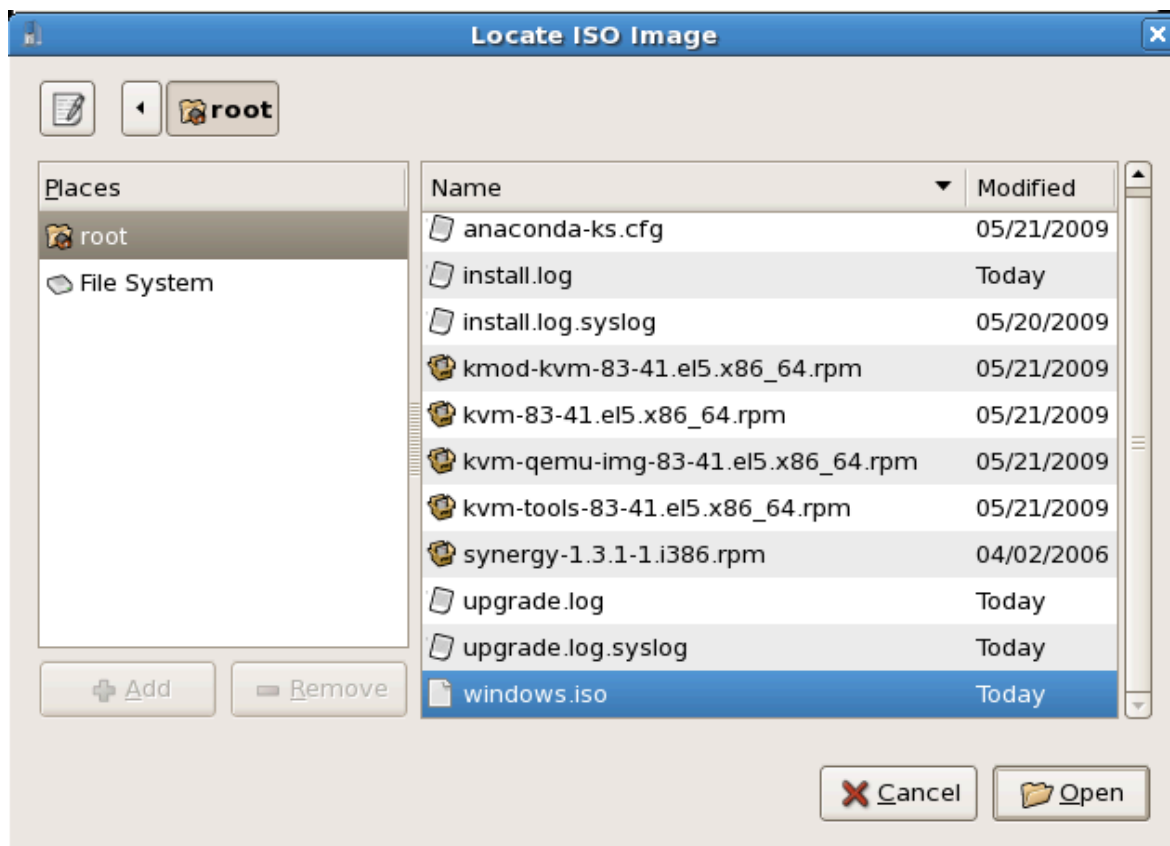


Klik op **Forward** om verder te gaan.

7. **Locate installation media**

Selecteer de ISO image locatie of het CD-ROM of DVD station. Dit voorbeeld gebruikt een ISO bestand image van de Windows Server 2008 installatie CD.

- a. Press the **Browse** button.
- b. Zoek naar de locatie van het ISO bestand en selecteer het.



Klik op **Open** om je selectie te bevestigen.

- c. Het bestand is geselecteerd en klaar om van te installeren.

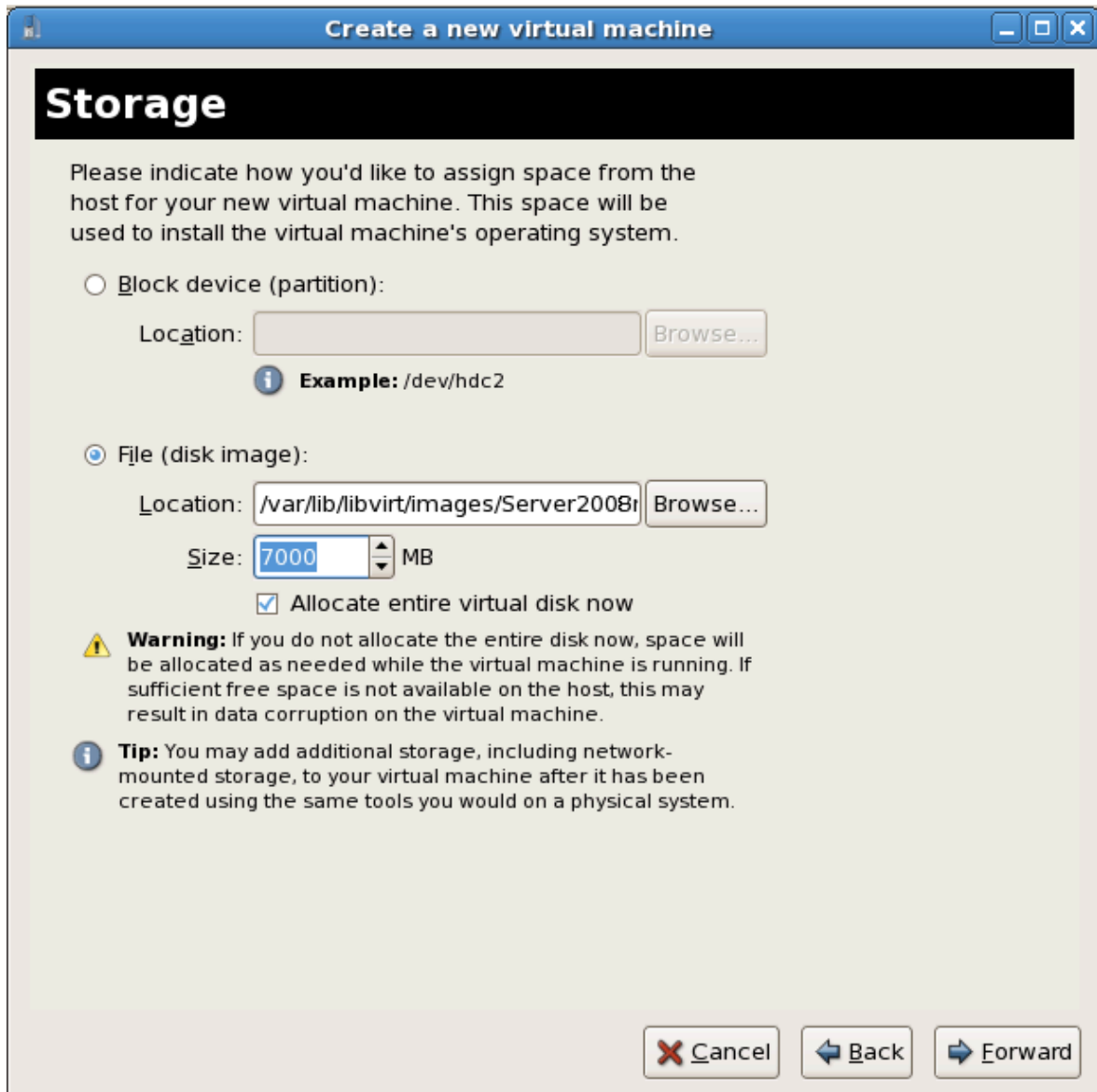


Klik op **Forward** om verder te gaan.

 **Image files and SELinux**
Voor ISO image bestanden en guest opslag images, wordt het aanbevolen om de `/var/lib/libvirt/images/` map te gebruiken. Elke andere locatie heeft misschien extra configuratie nodig voor SELinux, refereer naar *Paragraaf 7.1, "SELinux en virtualisatie"* voor details.

8. Storage setup

Ken een fysiek opslag apparaat (**Block device**) of een bestand-gebaseerde image (**File**) toe. Bestand-gebaseerde images moeten opgeslagen worden in de `/var/lib/libvirt/images/` map. Ken voldoende opslag toe aan je gevirtualiseerde guest. Ken voldoende ruimte toe aan je gevirtualiseerde guest en alle toepassingen die het nodig heeft.



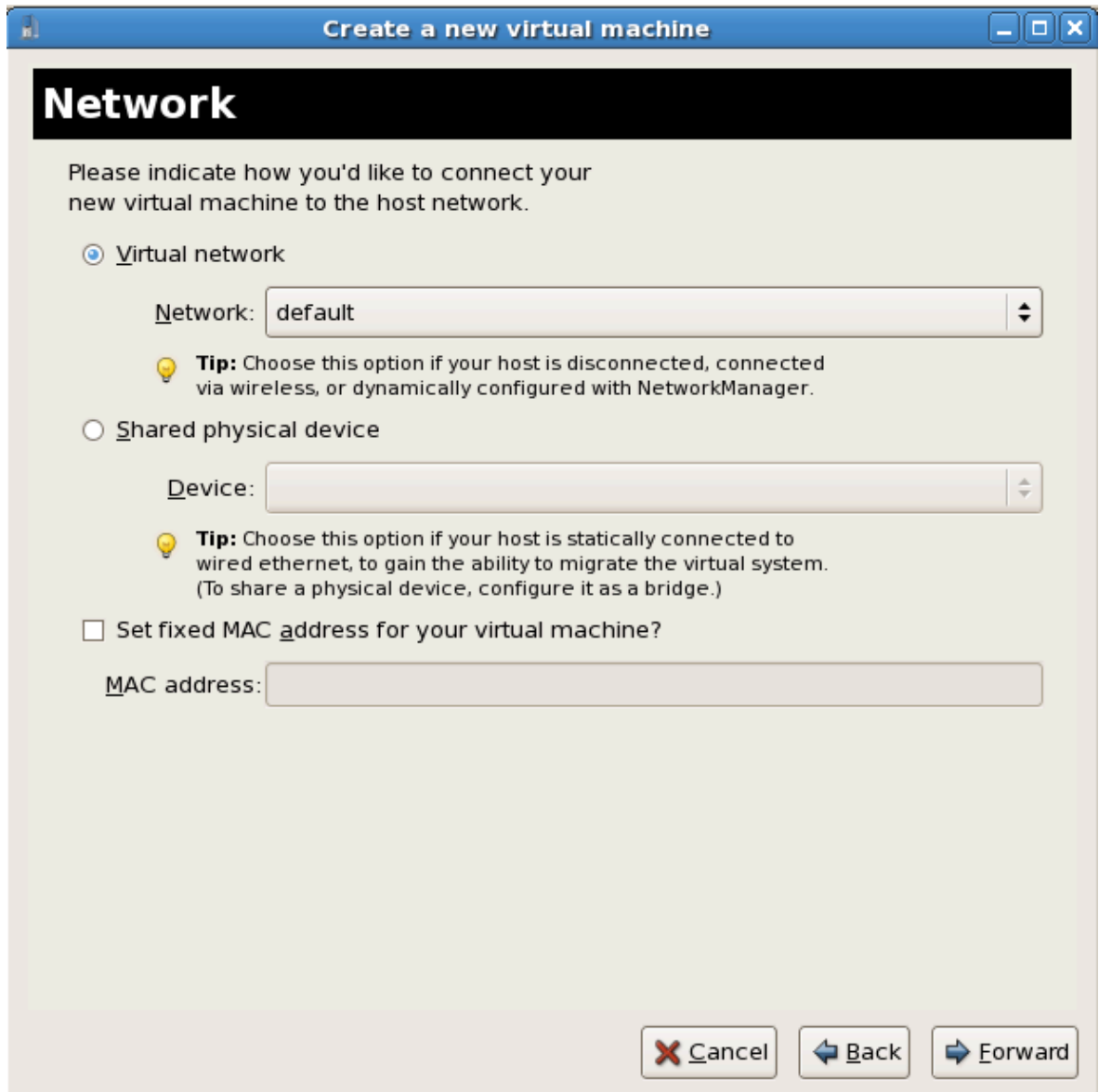
Klik op **Forward** om verder te gaan.

9. **Network setup**

Select either **Virtual network** or **Shared physical device**.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.



Press **Forward** to continue.

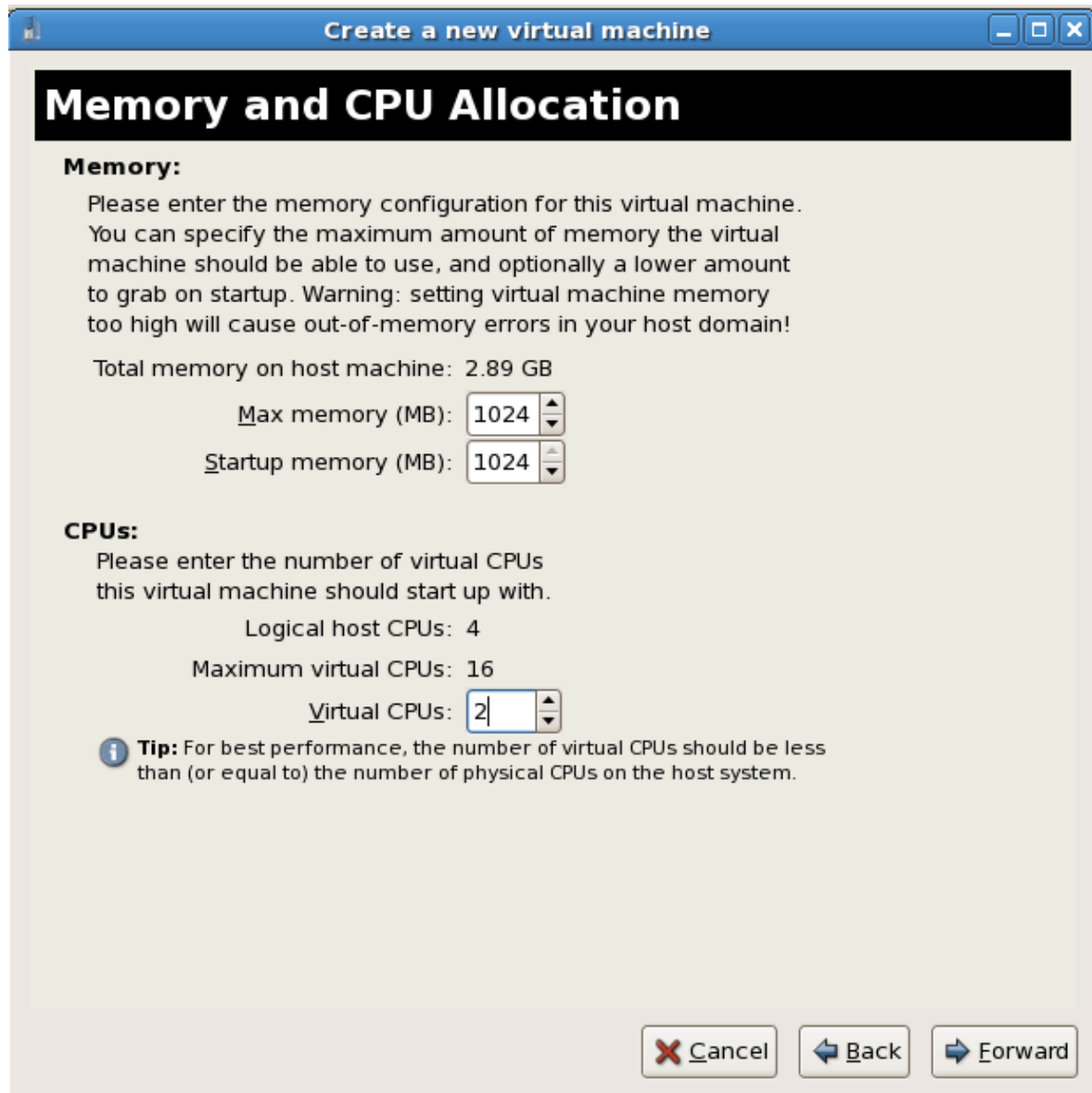
10. Memory and CPU allocation

The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

Virtualized guests require sufficient physical memory (RAM) to run efficiently and effectively. Choose a memory value which suits your guest operating system and application requirements. Windows Server 2008. Remember, guests use physical RAM. Running too many guests or leaving insufficient memory for the host system results in significant usage of virtual memory and swapping. Virtual memory is significantly slower causing degraded system performance and responsiveness. Ensure to allocate sufficient memory for all guests and the host to operate effectively.

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on

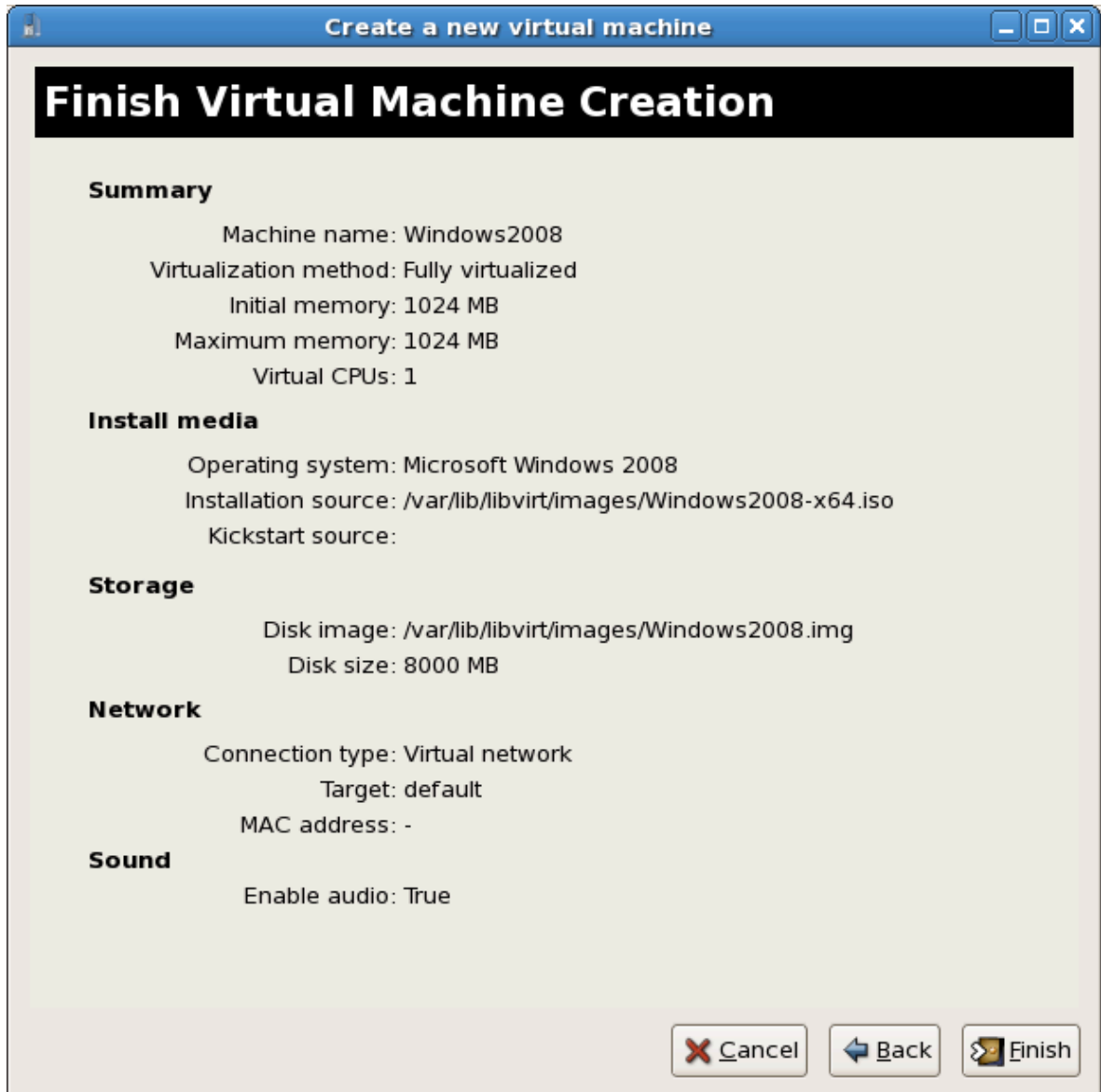
the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



Press **Forward** to continue.

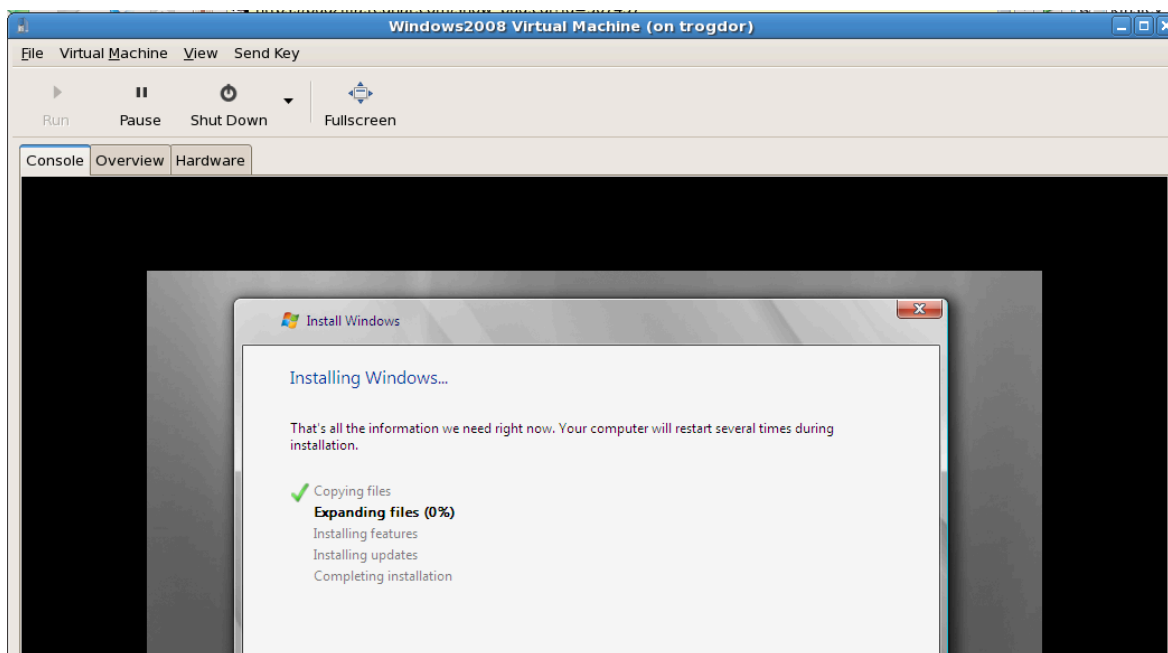
11. Verify and start guest installation

Verifieer de configuratie.



Klik op **Finish** om de guest installatie procedure te beginnen.

12. Windows installeren



Maak de Windows Server 2008 installatie volgorde af. De installatie volgorde wordt in deze gids niet behandeld, refereer naar Microsoft's [documentatie](#)¹ voor informatie over het installeren van Windows.

Deel II. Configuration

Virtualisatie instellen in Fedora

Deze hoofdstukken behandelen instel procedures voor verscheidene geavanceerde virtualisatie taken. De taken omvatten netwerk en opslag apparaten toevoegen, beveiliging verbeteren, prestaties verbeteren, en het gebruik van para-gevirtualiseerde drivers op volledig gevirtualiseerde guests.

Gevirtualiseerde blok apparaten

Dit hoofdstuk behandelt het installeren en instellen van blok apparaten in gevirtualiseerde guests. De term blok apparaten refereert naar verschillende vormen van opslag apparaten.

4.1. Een gevirtualiseerde floppy disk controller aanmaken

Floppy disk controllers zijn nodig voor een aantal oudere besturingssystemen, in het bijzonder voor het installeren van drivers. Op dit moment kunnen fysieke floppy disk stations niet benaderd worden vanuit gevirtualiseerde guests. Echter het aanmaken en benaderen van floppy disk images vanaf gevirtualiseerde floppy stations wordt ondersteund. Deze paragraaf behandelt het aanmaken van een gevirtualiseerde floppy station.

Een image bestand van een floppy disk is vereist. Maak floppy disk image bestanden met het **dd** commando. Vervang `/dev/fd0` met de naam van een floppy disk station en geef de schijf een geschikte naam.

```
# dd if=/dev/fd0 of=~/legacydrivers.img
```



Para-gevirtualiseerde drivers opmerking

De para-gevirtualiseerde drivers kunnen fysieke floppy stations afbeelden naar volledig gevirtualiseerde guests

Dit voorbeeld gebruikt een guest aangemaakt met **virt-manager** en draait een volledig gevirtualiseerde Linux installatie met een image geplaatst in `/var/lib/libvirt/images/rhel5FV.img`. De Xen hypervisor wordt in het voorbeeld gebruikt.

1. Maak het XML configuratie bestand voor je guest image met gebruik van het **virsh** commando op een draaiende guest.

```
# virsh dumpxml rhel5FV > rhel5FV.xml
```

Dit slaat de configuratie instellingen op als een XML bestand welke bewerkt kan worden om de operaties en apparaten gebruikt door de guest aan te passen. Voor meer informatie over het gebruik van de virsh XML configuratie bestanden, refereer je naar [Hoofdstuk 18, Aangepaste libvirt scripts aanmaken](#).

2. Maak een floppy disk image voor de guest.

```
# dd if=/dev/zero of=/var/lib/libvirt/images/rhel5FV-floppy.img bs=512 count=2880
```

3. Voeg de onderstaande regels, met veranderingen waar nodig, toe aan het configuratie XML bestand van jouw guest. Dit voorbeeld maakt een guest met een floppy station als een bestand-gebaseerd virtueel apparaat.

```
<disk type='file' device='floppy'>
  <source file='/var/lib/libvirt/images/rhel5FV-floppy.img' />
```

```
<target dev='fda' />
</disk>
```

4. Stop de guest

```
# virsh stop rhe15FV
```

5. Start de guest opnieuw op met gebruik van het XML bestand.

```
# virsh create rhe15FV.xml
```

Het floppy station is nu beschikbaar in de guest en opgeslagen als een image bestand op de host.

4.2. Opslag apparaten toevoegen aan guests

Deze paragraaf behandelt het toevoegen van opslag apparaten aan gevirtualiseerde guests. Extra opslag kan alleen toegevoegd worden nadat guests zijn aangemaakt. De ondersteunde opslag apparaten en protocollen zijn:

- locale harde schijf partities,
- logische volumes,
- Fibre Channel of iSCSI direct verbonden met de host.
- Bestand containers die zich bevinden in een bestandssysteem op de host.
- **NFS** bestandssystemen direct aangekoppeld door de virtuele machine.
- iSCSI opslag direct toegankelijk voor de guest.
- Cluster bestandssystemen (**GFS**).

Bestand-gebaseerde opslag toevoegen aan een guest

Bestand-gebaseerde opslag of bestand-gebaseerde containers zijn bestanden op het host bestandssysteem welke optreden als gevirtualiseerde harde schijven voor gevirtualiseerde guests. Om een bestand-gebaseerde container toe te voegen voer je de volgende stappen uit:

1. Maak een leeg container bestand of gebruik een bestaande bestand container (zoals een ISO bestand).
 - a. Maak een sparse bestand met gebruik van het **dd** commando. Sparse bestanden worden niet aanbevolen door problemen met data integriteit en prestaties. Sparse bestanden kun veel sneller aangemaakt worden en kunnen gebruikt worden voor testen, maar moeten niet gebruikt worden in productie omgevingen.

```
# dd if=/dev/zero of=/var/lib/libvirt/images/FileName.img bs=1M
seek=4096 count=0
```

- b. Niet-sparse, voor-toegekende bestanden worden aanbevolen voor bestand-gebaseerde opslag containers. Om een niet-sparse bestand te maken, voer je uit:


```
# dd if=/dev/zero of=/var/lib/libvirt/images/FileName.img bs=1M
count=4096
```

Beide commando's maken een 400 MB bestand aan welke gebruikt kan worden als extra opslag voor een gevirtualiseerde guest.

- Schrijf de configuratie weg voor de guest. In dit voorbeeld wordt de guest *Guest1* genoemd en het bestand wordt opgeslagen in de persoonlijke map van de gebruiker.

```
# virsh dumpxml Guest1 > ~/Guest1.xml
```

- Open het configuratie bestand (*Guest1.xml* in dit voorbeeld) met een tekstverwerker. Zoek de regels op die beginnen met "disk=". Deze regels lijken op:

```
>disk type='file' device='disk'<
  >driver name='tap' type='aio'</>
  >source file='/var/lib/libvirt/images/Guest1.img'</>
  >target dev='xvda'</>
</disk<
```

- Voeg de extra opslag toe door het aanpassen van het einde van de disk= regels. Wees er zeker van dat je een apparaat naam opgeeft voor het virtuele blok apparaat die nog niet in gebruik is in het configuratie bestand. Het volgende voorbeeld voegt een bestand, met de naam **FileName.img**, toe als een bestand-gebaseerde opslag container.

```
>disk type='file' device='disk'<
  >driver name='tap' type='aio'</>
  >source file='/var/lib/libvirt/images/Guest1.img'</>
  >target dev='xvda'</>
</disk<
>disk type='file' device='disk'<
  >driver name='tap' type='aio'</>
  >source file='/var/lib/libvirt/images/FileName.img'</>
  >target dev='hda'</>
</disk<
```

- Start de guest opnieuw op met het vernieuwde configuratie bestand.

```
# virsh create Guest1.xml
```

- De volgende stappen zijn specifiek voor een Linux guest. Andere besturingssystemen behandelen nieuwe opslag apparaten op andere manieren. Voor niet-Linux systemen refereer je naar de documentatie van je guest besturingssysteem.

De guest gebruikt nu het bestand **FileName.img** als een apparaat met de naam **/dev/hdb**. Dit apparaat vereist formattering van de guest. Op de guest partitioneer je het apparaat als een primaire partitie voor het gehele apparaat en formatteer dan het apparaat.

- Type *n* voor een nieuwe partitie.

```
# fdisk /dev/hdb  
Command (m for help):
```

- b. Type *p* voor een primaire partitie.

```
Command action  
  e   extended  
  p   primary partition (1-4)
```

- c. Kies een beschikbaar partitie nummer. In dit voorbeeld wordt de eerste partitie gekozen door een *1* in te typen.

```
Partition number (1-4): 1
```

- d. Kies de standaard eerste cilinder door op *Enter* te duwen.

```
First cylinder (1-400, default 1):
```

- e. Selecteer de grootte van de partitie. In dit voorbeeld wordt de gehele schijf toegekend door op *Enter* te duwen.

```
Last cylinder or +size or +sizeM or +sizeK (2-400, default 400):
```

- f. Stel de partitie type in door *t* in te typen.

```
Command (m for help): t
```

- g. Kies de partitie die je in de vorige stappen aangemaakt hebt. In dit voorbeeld is dat partitie *1*.

```
Partition number (1-4): 1
```

- h. Type *83* in voor een Linux partitie.

```
Hex code (type L to list codes): 83
```

- i. Schrijf de veranderingen naar schijf en sluit af.

```
Command (m for help): w  
Command (m for help): q
```

- j. Formateer de nieuwe partitie met het ext3 bestandssysteem.

```
# mke2fs -j /dev/hdb
```

7. Koppel de schijf aan op de guest.

```
# mount /dev/hdb1 /mijnbestanden
```

De guest heeft nu een extra gevirtualiseerd bestand-gebaseerd opslag apparaat.

Harde schijven en andere blok apparaten toevoegen aan een guest

Systeem beheerders gebruiken extra harde schijven voor het aanbieden van meer opslag ruimte of om systeem data te scheiden van gebruikers data. Deze procedure, [Procedure 4.1, "Fysieke blok apparaten toevoegen aan gevirtualiseerde guests"](#), beschrijft hoe je een harde schijf op de host toevoegt aan een gevirtualiseerde guest.

De procedure werkt voor alle fysieke blok apparaten, zoals CD-ROM, DVD, en floppy disk stations.

Procedure 4.1. Fysieke blok apparaten toevoegen aan gevirtualiseerde guests

1. Verbindt de harde schijf fysiek met de host. Configureer de host als de schijf standaard niet bereikbaar is.
2. Configureer de schijf op de host met **multipath** en vasthoudendheid als dat nodig is.
3. Gebruik het **virsh attach** commando. Vervang: *myguest* met de naam van jouw guest, /*dev/hdb1* met de schijf die je toevoegt, en *hdc* met de locatie voor de schijf op de guest. *hdc* moet een niet gebruikte apparaat naam zijn. Gebruik de *hd** notatie ook voor Windows guests, de guest zal het apparaat correct herkennen.

Voeg de `--type hdd` parameter toe aan het commando voor CD-ROM en DVD stations.

Voeg de `--type floppy` parameter toe aan het commando voor floppy stations.

```
# virsh attach-disk myguest /dev/hdb1 hdc --driver tap --mode readonly
```

4. De guest heeft nu een nieuwe harde schijf met de naam **/dev/hdb** onder Linux of **D: drive**, of iets dergelijks, onder Windows. Deze schijf kan formattering vereisen.

4.3. Blijvende opslag configureren

Deze paragraaf is voor systemen met externe opslag of opslag via het netwerk; dat betekent, Fibre Channel of iSCSI gebaseerde opslag apparaten. Het wordt aanbevolen dat deze systemen blijvende apparaat namen instellen voor jouw hosts. Dit helpt bij live migratie en biedt ook consistente apparaat namen en opslag voor meerdere gevirtualiseerde systemen.

Universele Unieke Identifiers(UUID's) is een standaard methode voor het identificeren van computers en apparaten in verspreide computer omgevingen. Deze paragraaf gebruikt UUID's voor het identificeren van iSCSI of Fibre Channel LUN's. UUID's zijn blijvend na opnieuw opstarten, los koppelen en apparaat omwisselen. De UUID is vergelijkbaar met een label op het apparaat.

Systemen die **multipath** niet draaien, moeten [Enkel-pad configuratie](#) gebruiken. Systemen die **multipath** draaien kunnen [Meervoudig-pad configuratie](#) gebruiken.

Enkel-pad configuratie

Deze procedure implementeert *LUN* apparaat vasthoudendheid met gebruik van **udev**. Gebruik deze procedure alleen voor hosts die **multipath** niet gebruiken.

1. Bewerk het **/etc/scsi_id.config** bestand.
 - a. Verzeker je ervan dat de **options=-b** uitgecommentarieerd is.

```
# options=-b
```

- b. Voeg de volgende regel toe:

```
options=-g
```

Deze optie stelt **udev** in om aan te nemen dat alle aangekoppelde SCSI apparaten een UUID teruggeven.

2. Om de UUID voor een bepaald apparaat te tonen voer je het **scsi_id -g -s /block/sd*** commando uit. Bijvoorbeeld:

```
# scsi_id -g -s /block/sd*
3600a0b800013275100000015427b625e
```

De output kan anders zijn dan in het voorbeeld hierboven. De output laat de UUID zien van het apparaat **/dev/sdc**.

3. Controleer of de UUID output identiek is met het **scsi_id -g -s /block/sd*** commando van de computer die toegang heeft tot het apparaat.
4. Maak een regel om het apparaat een naam te geven. Maak een bestand aan met de naam **20-names.rules** in re **/etc/udev/rules.d** map. Voeg nieuwe regels toe aan dit bestand. Alle regels worden toegevoegd aan hetzelfde bestand met gebruik van hetzelfde formaat. De regels volgen dit formaat:

```
KERNEL="sd*", BUS="scsi", PROGRAM="/sbin/scsi_id -g -s", RESULT=UUID,
NAME=apparaatnaam
```

Vervang *UUID* en *apparaatnaam* met de UUID hierboven verkregen, en een naam voor het apparaat. Dit is een regel voor het voorbeeld hierboven:

```
KERNEL="sd*", BUS="scsi", PROGRAM="/sbin/scsi_id -g -s",
RESULT="3600a0b800013275100000015427b625e", NAME="rack4row16"
```

De **udev** daemon zoekt nu alle apparaten met de naam **/dev/sd*** af voor de UUID in de regel. Zodra een overeenkomend apparaat is verbonden met het systeem krijgt het apparaat de naam uit de regel toegekend. In het voorbeeld verschijnt een apparaat met een UUID van **3600a0b800013275100000015427b625e** als **/dev/rack4row16**.

5. Voeg deze regel toe aan **/etc/rc.local**:

```
/sbin/start_udev
```

6. Kopieer de veranderingen in de `/etc/scsi_id.config`, `/etc/udev/rules.d/20-names.rules`, en `/etc/rc.local` bestanden naar alle relevante hosts.

```
/sbin/start_udev
```

Netwerk opslag apparaten met ingestelde regels hebben nu blijvende namen op alle hosts waar de bestanden vernieuwd waren. Dit betekent dat je kunt migreren tussen hosts met gebruik van de gedeelde opslag en de guests hebben toegang tot de opslag apparaten in hun configuratie bestanden.

Meervoudig-pad configuratie

Het **multipath** pakket wordt gebruikt voor systemen met meer dan een fysiek pad van de computer naar opslag apparaten. **multipath** biedt fout tolerantie, failover en verbeterde prestaties voor netwerk opslag apparaten verbonden aan een Linux systeem.

Het implementeren van LUN vasthoudendheid in een **multipath** omgeving vereist gedefinieerde alias namen voor jouw meervoudig-pad apparaten. Elk opslag apparaat heeft een UUID dat optreedt als een sleutel voor de alias namen. Identificeer de UUID van een apparaat met gebruik van het **scsi_id** commando.

```
# scsi_id -g -s /block/sdc
```

De meervoudig-pad apparaten worden aangemaakt in de `/dev/mpath` map. In het voorbeeld hieronder worden vier apparaten gedefinieerd in `/etc/multipath.conf`:

```

multipaths {
    multipath {
        wwid          3600805f300159870000000000768a0019
        alias         oramp1
    }
    multipath {
        wwid          3600805f300159870000000000d643001a
        alias         oramp2
    }
    mulitpath {
        wwid          3600805f30015987000000000086fc001b
        alias         oramp3
    }
    mulitpath {
        wwid          3600805f300159870000000000984001c
        alias         oramp4
    }
}

```

Deze instelling zal vier LUN's aanmaken met de namen `/dev/mpath/oramp1`, `/dev/mpath/oramp2`, `/dev/mpath/oramp3` en `/dev/mpath/oramp4`. Zodra ze opgegeven zijn, wordt de afbeelding van de WWID van de apparaten naar hun namen blijvend na opnieuw opstarten.

4.4. Voeg een gevirtualiseerde CD-ROM of DVD station toe aan een guest

Om een ISO bestand te koppelen aan een guest terwijl de guest on-line is, gebruik je **virsh** met de *attach-disk* parameter.

```
# virsh attach-disk [domain-id] [source] [target] --driver file --type  
  cdrom --mode readonly
```

De *source* en *target* parameters zijn paden voor de bestanden en apparaten, respectievelijk op de host en de guest. De *source* parameter kan een pad naar een ISO bestand zijn of het apparaat van de **/dev** map.

Gedeelde opslag en virtualisatie

Dit hoofdstuk behandelt gedeelde netwerk opslag met virtualisatie in Fedora.

De volgende methodes worden ondersteund voor virtualisatie:

- Fibre Channel
- iSCSI
- NFS
- GFS2

Netwerk opslag is essentieel voor live en off-line guest migraties. Je kunt geen guests migreren zonder gedeelde opslag.

5.1. iSCSI gebruiken voor het opslaan van guests

Deze paragraaf behandelt het gebruik van iSCSI-gebaseerde apparaten om gevirtualiseerde guests op te slaan.

5.2. NFS gebruiken voor het opslaan van guests

Deze paragraaf behandelt het gebruik van NFS om gevirtualiseerde guests op te slaan.

5.3. GFS2 gebruiken voor het opslaan van guests

Deze paragraaf behandelt het gebruik van het Fedora Global File Systeem 2 (GFS2) om gevirtualiseerde guests op te slaan.

Server beste praktijken

De volgende taken en tips kunnen je helpen met het beveiligen en het verzekeren van de betrouwbaarheid van je Fedora server host (dom0).

- Draai SELinux in de afdwingende mode. Je kunt dit doen door het volgende commando uit te voeren.

```
# setenforce 1
```

- Verwijder alle onnodige services of zet ze uit, zoals **AutoFS**, **NFS**, **FTP**, **HTTP**, **NIS**, **telnetd**, **sendmail** enzovoort.
- Voeg alleen het minimale aantal gebruikeraccounts toe die nodig zijn voor het platform beheer op de server en verwijder onnodige gebruikeraccounts.
- Vermijd het draaien van alle niet-essentiële toepassingen op je host. Het draaien van toepassingen op de host kan de virtuele machine prestaties beïnvloeden en kan de server stabiliteit aantasten. Elke toepassing welke de server laat crashen zal ook alle virtuele machines op de server buiten werking stellen.
- Gebruik een centrale locatie voor virtuele machine installaties en images. Virtuele machine images moeten opgeslagen worden in **/var/lib/libvirt/images/**. Als je een andere map gebruikt voor je virtuele machine images wees er dan zeker van dat je de map toevoegt aan je SELinux richtlijnen en het opnieuw labelt voordat je de installatie opstart.
- Installatie bronnen, bomen, en images moeten opgeslagen worden in een centrale locatie, gewoonlijk de locatie van je vsftpd server.

Beveiliging voor virtualisatie

Bij het gebruik van virtualisatie technologieën in je bedrijfs infrastructuur moet je er zeker van zijn dat de host niet in gevaar kan worden gebracht. De host, voor de Xen hypervisor, is een domein met rechten dat systeem beheer afhandelt en alle virtuele machines beheert. Als de host onveilig is, zijn alle andere domeinen in het systeem kwetsbaar. Er zijn verschillende manieren om de beveiliging te verbeteren van systemen die virtualisatie gebruiken. Jij of jouw onderneming moet een *Inzet plan* maken die de werk specificaties bevat een specificeert welke services nodig zijn op je gevirtualiseerde guests on host servers en ook welke ondersteuning nodig is voor die services. Hier volgen een paar beveiligings problemen die overwogen moeten worden tijdens het maken van een inzet plan.

- Draai alleen de nodige services op hosts. Hoe minder processen en services op de host draaien, des te hoger het beveiliging niveau en prestaties zijn.
- Zet [SELinux](#) aan op de hypervisor. Lees [Paragraaf 7.1, "SELinux en virtualisatie"](#) voor meer informatie over het gebruik van SELinux en virtualisatie.
- Gebruik een firewall om het verkeer naar dom0 te beperken. Je kunt een firewall instellen met standaard-weigerings regels die zullen helpen om aanvallen op dom0 te voorkomen. Het is ook belangrijk om het aantal services die contact hebben met het netwerk te beperken.
- Sta normale gebruikers geen toegang toe tot dom0. Als je normale gebruikers toegang tot dom0 toestaat, loop je het risico dom0 kwetsbaar te maken. Derk eraan dat dom0 rechten heeft, en het toestaan van accounts zonder rechten kan het niveau van beveiliging in gevaar brengen.

7.1. SELinux en virtualisatie

Security Enhanced Linux werd ontwikkeld door de NSA met hulp van de Linux gemeenschap om een betere beveiliging voor Linux te bieden. SELinux beperkt de mogelijkheden van een aanvaller en helpt om vele vaak voorkomende uitbuitingen zoals buffer overflow aanvallen en rechten escalatie te voorkomen. Deze voordelen zijn de reden dat Fedora aanbeveelt om alle Linux systemen moeten draaien met SELinux aangezet en werkend in de afdwingende mode.

SELinux belet guest images om te laden als SELinux is aangezet en de images zich niet in de juiste map bevinden. SELinux vereist dat alle guest images zijn opgeslagen in **`/var/lib/libvirt/images`**.

LVM gebaseerde opslag toevoegen met SELinux in de afdwingende mode

De volgende paragraaf is een voorbeeld van het toevoegen van een logische volume aan een gevirtualiseerde guest met SELinux aangezet. Deze instructies werken ook voor harde schijf partities.

[Procedure 7.1. Het aanmaken en aankoppelen van een logische volume op een gevirtualiseerde guest met SELinux aangezet](#)

1. Maak een logische volume aan. Dit voorbeeld maakt een 5 Gigabyte logische volume met de naam *NewVolumeName* op de volume groep met de naam *volumegroup*.

```
# lvcreate -n NewVolumeName -L 5G volumegroup
```

2. Formateer de *NewVolumeName* logische volume met een bestandssysteem dat uitgebreide attributen ondersteunt, zoals ext3.

```
# mke2fs -j /dev/volumegroup/NewVolumeName
```

3. Maak een nieuwe map aan voor het aankoppelen van de nieuwe volume. Deze map kan overal op je bestandssysteem zijn. Het wordt aanbevolen het niet in belangrijke systeem mappen te plaatsen (**/etc**, **/var**, **/sys**) or in home directories (**/home** of **/root**). Dit voorbeeld gebruikt een map met de naam **/virtstorage**

```
# mkdir /virtstorage
```

4. Koppel de logische volume aan.

```
# mount /dev/volumegroup/NewVolumeName /virtstorage
```

5. Stel het juiste SELinux type in voor de Xen map.

```
semanage fcontext -a -t xen_image_t "/virtualization(/.*)?"
```

Of, stel het het juiste SELinux type in voor een KVM map.

```
semanage fcontext -a -t virt_image_t "/virtualization(/.*)?"
```

Als de gerichte tactiek wordt gebruikt (gericht is de standaard tactiek) voegt het commando een regel toe aan het **/etc/selinux/targeted/contexts/files/file_contexts.local** bestand wat de verandering blijvend maakt. De toegevoegde regel kan hier op lijken:

```
/virtstorage(/.*)?    system_u:object_r:xen_image_t:s0
```

6. Voer het commando uit om het type van het aankoppelpunt (**/virtstorage**) en alle bestanden hierin te veranderen naar **xen_image_t** (**restorecon** en **setfiles** lezen de bestanden in **/etc/selinux/targeted/contexts/files/**).

```
# restorecon -R -v /virtualization
```

7.2. SELinux overwegingen

Deze paragraaf bevat een paar dingen die je moet overwegen als je SELinux implementeert in je virtualisatie inzet. Als je systeem veranderingen uitvoert of apparaten toevoegt, moet je jouw SELinux tactiek hierop aanpassen. Om een LVM volume in te stellen voor een guest, moet je de SELinux context veranderen voor de respectievelijke onderliggende blok apparaat en volume groep.

```
# semanage fcontext -a -t xen_image_t -f -b /dev/sda2
# restorecon /dev/sda2
```

De boolean parameter **xend_disable_t** kan **xend** instellen in de onbeperkte mode na het opnieuw opstarten van de daemon. Het is beter om bescherming uit te zetten voor een enkele daemon dat voor het gehele systeem. Het wordt aanbevolen dat je mappen niet opnieuw moet labelen als **xen_image_t**, als je ze ook op andere plaatsen gebruikt.

Netwerk configuratie

Deze pagina biedt een inleiding voor de algemene netwerk instellingen die gebruikt worden door op libvirt gebaseerde toepassingen. Deze informatie is van toepassing op alle hypervisors, of dit nu Xen, KVM, of een andere is. Voor meer informatie raadpleeg de de libvirt netwerk architectuur documentatie.

De twee algemene instellingen zijn "virtueel netwerk" of "gedeeld fysiek apparaat". De eerste is identiek voor alle distributies en kant en klaar beschikbaar. De laatste heeft handmatige instelling specifiek voor de distributie nodig.

8.1. Netwerk adres vertaling (NAT) met libvirt

Een van de meest algemene methodes voor het delen van netwerk verbindingen is om netwerk adres vertaling (NAT) doorsturen te gebruiken (ook bekend als virtuele netwerken).

Host configuratie

Elke standaard libvirt installatie biedt op NAT gebaseerde verbindingen naar virtuele machines kant en klaar aan. Dit is het zogenaamde 'standaard virtuele netwerk'. Controleer of het beschikbaar is met het **virsh net-list --all** commando.

```
# virsh net-list --all
Name                State      Autostart
-----
default             active    yes
```

Als het ontbreekt, kan het voorbeeld XML configuratie bestand opnieuw geladen worden en geactiveerd:

```
# virsh net-define /usr/share/libvirt/networks/default.xml
```

Het standaard netwerk wordt gedefinieerd door **/usr/share/libvirt/networks/default.xml**

Markeer het standaard netwerk om automatisch te starten:

```
# virsh net-autostart default
Network default marked as autostarted
```

Start het standaard netwerk:

```
# virsh net-start default
Network default started
```

Zodra het libvirt standaard netwerk draait, zul je een geïsoleerd brug apparaat zien. Aan dit apparaat heeft *geen* fysieke interfaces toegekend, omdat het NAT en IP doorsturen gebruikt om te verbinden met de buitenwereld. Voeg geen nieuwe interfaces toe.

```
# brctl show
bridge name      bridge id          STP enabled      interfaces
```

```
virbr0          8000.00000000000000    yes
```

libvirt voegt **iptables** regels toe welke verkeer toestaan naar en van de guests verbonden met het virbr0 apparaat in de **INPUT**, **FORWARD**, **OUTPUT** en **POSTROUTING** ketens. **libvirt** probeert dan om de **ip_forward** parameter aan te zetten. Sommige andere toepassingen kunnen **ip_forward** uitzetten, dus de beste optie is het volgende aan **/etc/sysctl.conf** toe te voegen.

```
net.ipv4.ip_forward = 1
```

Guest configuratie

Zodra de host configuratie klaar is, kan een guest verbonden worden met het virtuele netwerk op basis van zijn naam. Om een guest te verbinden het 'standaard' virtuele netwerk, kan de volgende XML gebruikt worden in de guest:

```
<interface type='network'>
  <source network='default' />
</interface>
```

Note

Het definiëren van een MAC adres is optioneel. Een MAC adres wordt automatisch aangemaakt als het weggelaten wordt. Het MAC adres handmatig instellen kan in bepaalde situaties nuttig zijn.

```
<interface type='network'>
  <source network='default' />
  <mac address='00:16:3e:1a:b3:4a' />
</interface>
```

8.2. Brug netwerken met libvirt

Netwerk bruggen (ook bekend als fysieke apparaat deling) wordt gebruikt om een fysiek apparaat toe te kennen aan een virtuele machine. Een brug wordt vaak gebruikt voor meer geavanceerde opstellingen en op servers met meerdere netwerk interfaces.

Xen netwerk uitzetten scripts

Als jouw systeem een Xen brug gebruikt, is het aan te bevelen om de standaard Xen netwerk brug uit te zetten door het bewerken van **/etc/xen/xend-config.sxp** en de volgende regel te veranderen van:

```
(network-script network-bridge)
```

naar:

```
(network-script /bin/true)
```

NetworkManager uitzetten

NetworkManager ondersteunt geen bruggen. NetworkManager moet uitgezet worden om de oudere netwerk instelling scripts te gebruiken.

```
# chkconfig NetworkManager off
# chkconfig network on
# service NetworkManager stop
# service network start
```



Note

In plaats van het uitzetten van NetworkManager, kun je "*NM_CONTROLLED=no*" toevoegen aan de *ifcfg-** scripts gebruikt in de voorbeelden.

Netwerk init scripts aanmaken

Maak of bewerk de volgende twee netwerk configuratie bestanden. Deze stap kan herhaald worden (met andere namen) voor extra netwerk bruggen.

Ga naar de `/etc/sysconfig/network-scripts` map:

```
# cd /etc/sysconfig/network-scripts
```

Open het netwerk script voor het apparaat die je gaat toevoegen aan de brug. In dit voorbeeld, definieert `ifcfg-eth0` het fysieke netwerk interface welke ingesteld is als onderdeel van de brug:

```
DEVICE=eth0
# change the hardware address to match the hardware address your NIC uses
HWADDR=00:16:76:D6:C9:45
ONBOOT=yes
BRIDGE=br0
```



Tip

Je kunt de Maximum Transfer Unit (MTU) van het apparaat instellen door een *MTU* variabele toe te voegen aan het einde van het configuratie bestand.

```
MTU=9000
```

Maak een nieuw netwerk script in de `/etc/sysconfig/network-scripts` map met de naam `ifcfg-br0` of iets dergelijks. De `br0` is de naam van de brug, dit kan van alles zijn zolang de naam van het bestand overeenkomt met de `DEVICE` parameter.

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
```

```
DELAY=0
```



Warning

The line, `TYPE=Bridge`, is case-sensitive. It must have uppercase 'B' and lower case 'ridge'.

Na het instellen, start je het netwerk opnieuw op of je start de computer opnieuw op.

```
# service network restart
```

Configure **iptables** to allow all traffic to be forwarded across the bridge.

```
# iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
# service iptables save
# service iptables restart
```



Disable iptables on bridges

Alternatively, prevent bridged traffic from being processed by **iptables** rules. In `/etc/sysctl.conf` append the following lines:

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Reload the kernel parameters configured with **sysctl**

```
# sysctl -p /etc/sysctl.conf
```

Restart the **libvirt** daemon.

```
# service libvirtd reload
```

Je moet nu een "gedeeld fysiek apparaat" hebben, welke verbonden worden kan met guests en die volledige LAN toegang hebben. Controleer je nieuwe brug:

```
# brctl show
bridge name      bridge id                STP enabled    interfaces
virbr0           8000.00000000000000     yes            eth0
br0              8000.000e0cb30550      no
```

Merk op dat de brug geheel onafhankelijk is van de **virbr0** brug. Probeer *niet* om een fysiek apparaat te verbinden met **virbr0**. De **virbr0** brug alleen voor Netwerk adres vertaling (NAT) verbinding.

KVM para-gevirtualiseerde drivers

Para-gevirtualiseerde drivers zijn beschikbaar voor gevirtualiseerde Windows guests die draaien op KVM hosts. Deze para-gevirtualiseerde drivers bevinden zich in het virtio pakket. Het virtio pakket ondersteunt blok (opslag) apparaten en netwerk interface controllers.

Para-gevirtualiseerde drivers verbeteren de prestaties van volledig gevirtualiseerde guests. Met de para-gevirtualiseerde drivers neemt guest I/O latentie af en de verwerkingscapaciteit neemt toe naar het bare-metal niveau. Het wordt aanbevolen de para-gevirtualiseerde drivers te gebruiken voor volledig gevirtualiseerde guests die zware I/O taken en toepassingen draaien.

De KVM para-gevirtualiseerde drivers worden automatisch geladen en geïnstalleerd op nieuwere versies van Fedora. Deze Fedora versies detecteren en installeren de drivers zodat extra installatie stappen niet nodig zijn.

Net als de KVM module, zijn de virtio drivers alleen beschikbaar op hosts die nieuwere versies van Fedora draaien.



Note

Er zijn slechts 28 PCI plaatsen beschikbaar voor extra apparaten per guest. Elk para-gevirtualiseerd netwerk of blok apparaat gebruikt een plaats. Elke guest kan tot 28 extra apparaten gebruiken welke opgebouwd kunnen worden met elke combinatie van para-gevirtualiseerde netwerk, para-gevirtualiseerde schijf stations, of andere PCI apparaten die VTd gebruiken.

De volgende Microsoft Windows versies hebben ondersteunde KVM para-gevirtualiseerde drivers:

- Windows XP,
- Windows Server 2003,
- Windows Vista, en
- Windows Server 2008.

9.1. De KVM Windows para-gevirtualiseerde drivers installeren

Deze paragraaf behandelt het installatie proces voor KVM Windows para-gevirtualiseerde drivers. De KVM para-gevirtualiseerde drivers kunnen geladen worden tijdens de Windows installatie of geïnstalleerd worden nadat de guest geïnstalleerd is.

Je kunt de para-gevirtualiseerde drivers installeren op je guest met een van de volgende methodes:

- de installatie bestanden te plaatsen op een netwerk toegankelijk voor de guest,
- het gebruik van een gevirtualiseerd CD-ROM station van de driver installatie schijf .iso bestand of
- het gebruik van een gevirtualiseerd floppy station om de drivers te installeren tijdens het opstarten (voor Windows guests).

Deze gids beschrijft de installatie van de para-gevirtualiseerde installeer schijf als een gevirtualiseerd CD-ROM station.

1. Download de drivers

De drivers zijn beschikbaar van Microsoft (windowsservercatalog.com¹).

Het *virtio-win* pakket installeert een CD-ROM image, **virtio-win.iso**, in de `/usr/share/virtio-win/` map.

2. Installeer de para-gevirtualiseerde drivers

Het wordt aanbevolen om de drivers te installeren op de guest voor het bevestigen of veranderen van een apparaat om de para-gevirtualiseerde drivers te gebruiken.

Voor blok apparaten die root bestandssystemen opslaan of andere blok apparaten vereist voor het opstarten van de guest, moeten de drivers geïnstalleerd worden voordat het apparaat wordt veranderd. Als de drivers niet geïnstalleerd worden op de guest en de driver is ingesteld voor de virtio driver zal de guest niet opstarten.

De image aankoppelen met virt-manager

Volg [Procedure 9.1, “virt-manager gebruiken om een CD-ROM image aan te koppelen voor een Windows guest”](#) om een CD-ROM image toe te voegen met **virt-manager**.

Procedure 9.1. virt-manager gebruiken om een CD-ROM image aan te koppelen voor een Windows guest

1. Open **virt-manager**, selecteer je gevirtualiseerde guest in de lijst van virtuele machines en klik op de **Details** knop.
2. Klik op de **Add** knop in het **Details** paneel.
3. Dit opent een instel-hulp voor het toevoegen van het nieuwe apparaat. Selecteer **Storage device** van het uitklap menu, en klik op **Forward**.



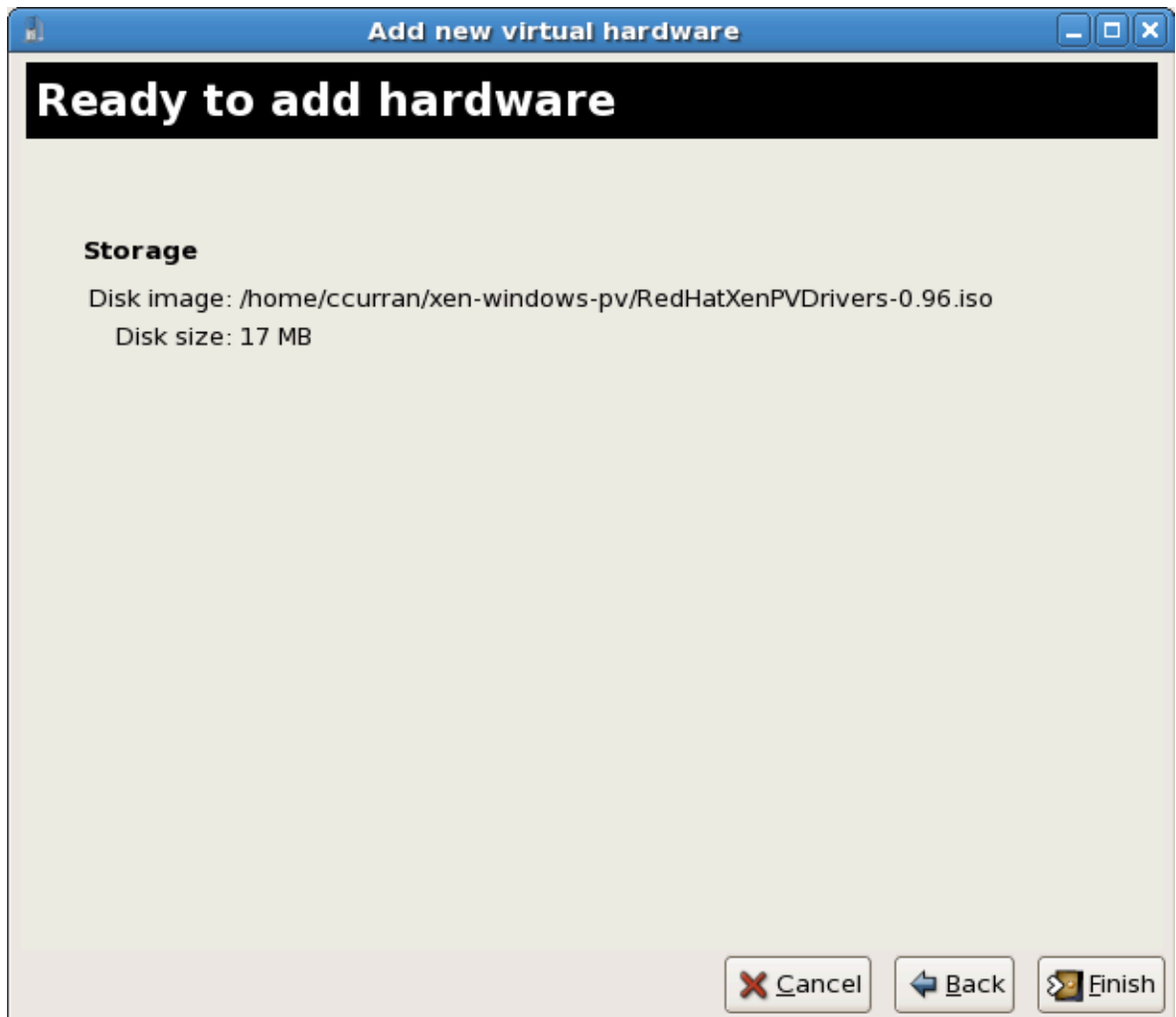
4. Kies de **File (disk image)** optie en stel de bestand locatie in van de para-gevirtualiseerde drivers .iso bestand. De locatie van de .iso bestanden is `/usr/share/xenpv-win` als je **yum** gebruikt hebt om de para-gevirtualiseerde driver pakketten te installeren.

Als de drivers zijn opgeslagen op een fysieke CD, gebruik je de **Normal Disk Partition** optie.

Stel het **Device type** in naar **IDE cdrom** en klik op **Forward** om verder te gaan.



5. De schijf is toegekend en is beschikbaar voor de guest zodra de guest wordt opgestart. Klik op **Finish** om de instel-hulp af te sluiten of ga terug als je een fout gemaakt hebt.



Installeren met een gevirtualiseerde floppy

Deze procedure behandelt het installeren van de para-gevirtualiseerde drivers tijdens een Windows installatie.

- Tijdens het installeren van de Windows VM voor de eerste keer met gebruik van het eenmaal-uitvoeren menu verbindt **viostor.vfd** als een floppy
 - a. **Windows Server 2003**

Als Windows vraagt om op F6 te duwen voor drivers van derden, doe dat dan en volg de instructies op het scherm op.
 - b. **Windows Server 2008**

Als de installer je naar de driver vraagt, klik dan op "Load Driver", verwijst de installer naar drive A: en kies de driver die past bij je OS en bit lengte.

KVM para-gevirtualiseerde drivers gebruiken voor bestaande apparaten

Verander een bestaande harde schijf aangesloten op de guest om de **virtio** driver te gebruiken in plaats van de gevirtualiseerde IDE driver. Dit voorbeeld bewerkt libvirt configuratie bestanden. Als alternatief kunnen **virt-manager**, **virsh attach-disk** of **virsh attach-interface** een

nieuw apparaat toevoegen met gebruik van para-gevirtualiseerde drivers [KVM para-gevirtualiseerde drivers gebruiken voor nieuwe apparaten](#).

1. Hieronder is een bestand-gebaseerd blok apparaat die de gevirtualiseerde IDE driver gebruikt. Dit is een typische regel voor een gevirtualiseerde guest die de para-gevirtualiseerde drivers niet gebruikt.

```
<disk type='file' device='disk'>
  <source file='/var/lib/libvirt/images/disk1.img' />
  <target dev='hda' bus='ide' />
</disk>
```

2. Verander de regel om het para-gevirtualiseerde apparaat te gebruiken door het veranderen van de **bus=** regel naar **virtio**.

```
<disk type='file' device='disk'>
  <source file='/var/lib/libvirt/images/disk1.img' />
  <target dev='hda' bus='virtio' />
</disk>
```

KVM para-gevirtualiseerde drivers gebruiken voor nieuwe apparaten

Deze procedure behandelt het maken van nieuwe apparaten met gebruik van de KVM para-gevirtualiseerde drivers met **virt-manager**.

Als alternatief kunnen **devirsh attach-disk** of **virsh attach-interface** commando's gebruikt worden apparaten aan te sluiten die de para-gevirtualiseerde drivers gebruiken.



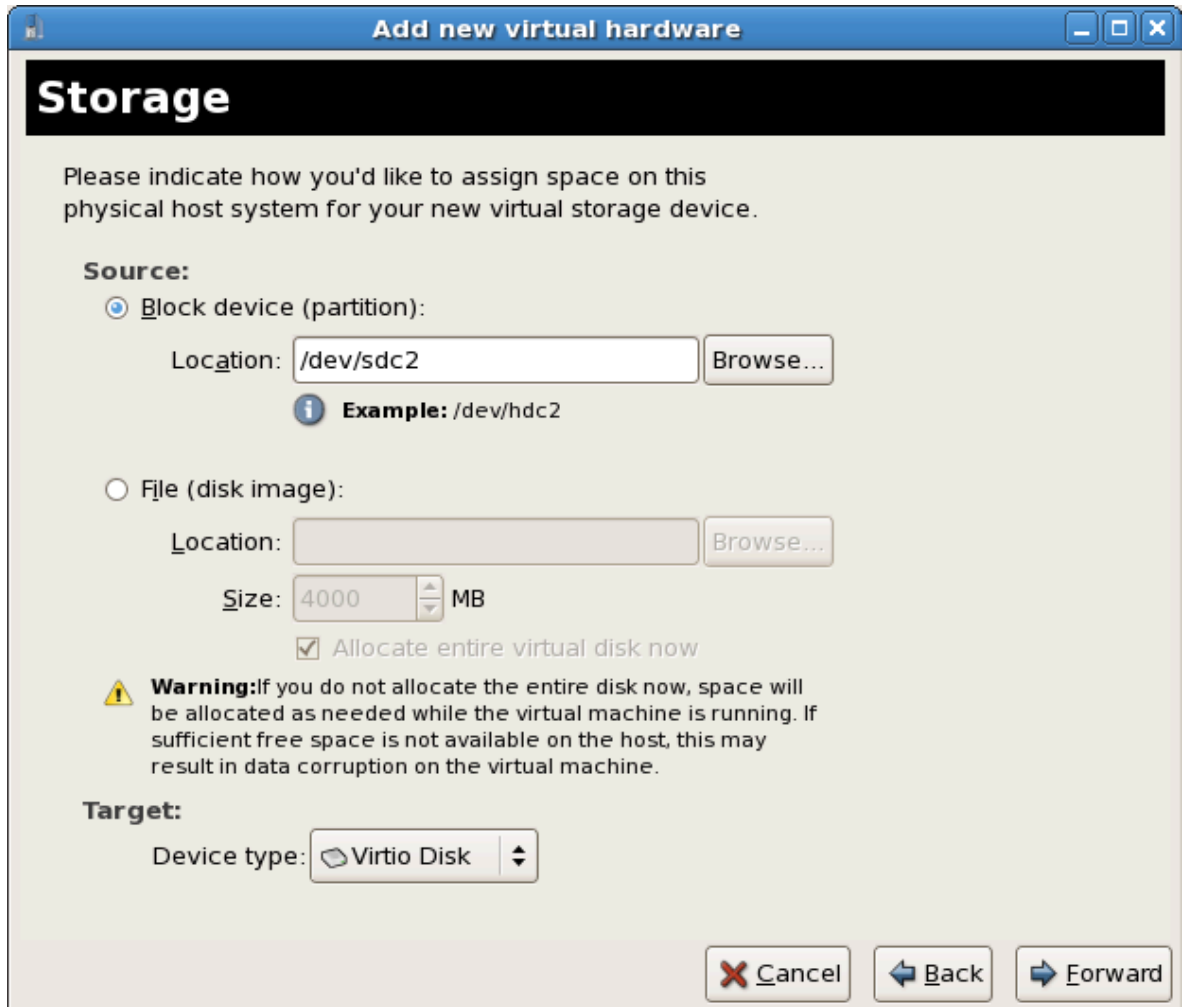
Installeer eerst de drivers

Wees er zeker van dat de drivers geïnstalleerd zijn op de Windows guest voordat je verder gaat met het installeren van de nieuwe apparaten. Als de drivers niet beschikbaar zijn zal het apparaat niet herkend worden en zal niet werken.

1. Opne de gevirtualiseerde guest door dubbel te klikken op de naam van de guest in **virt-manager**.
2. Open de **Hardware** tab.
3. Klik op de **Add Hardware** knop.
4. In de Adding Virtual Hardware tab selecteer je **Storage** of **Network** voor het type apparaat.

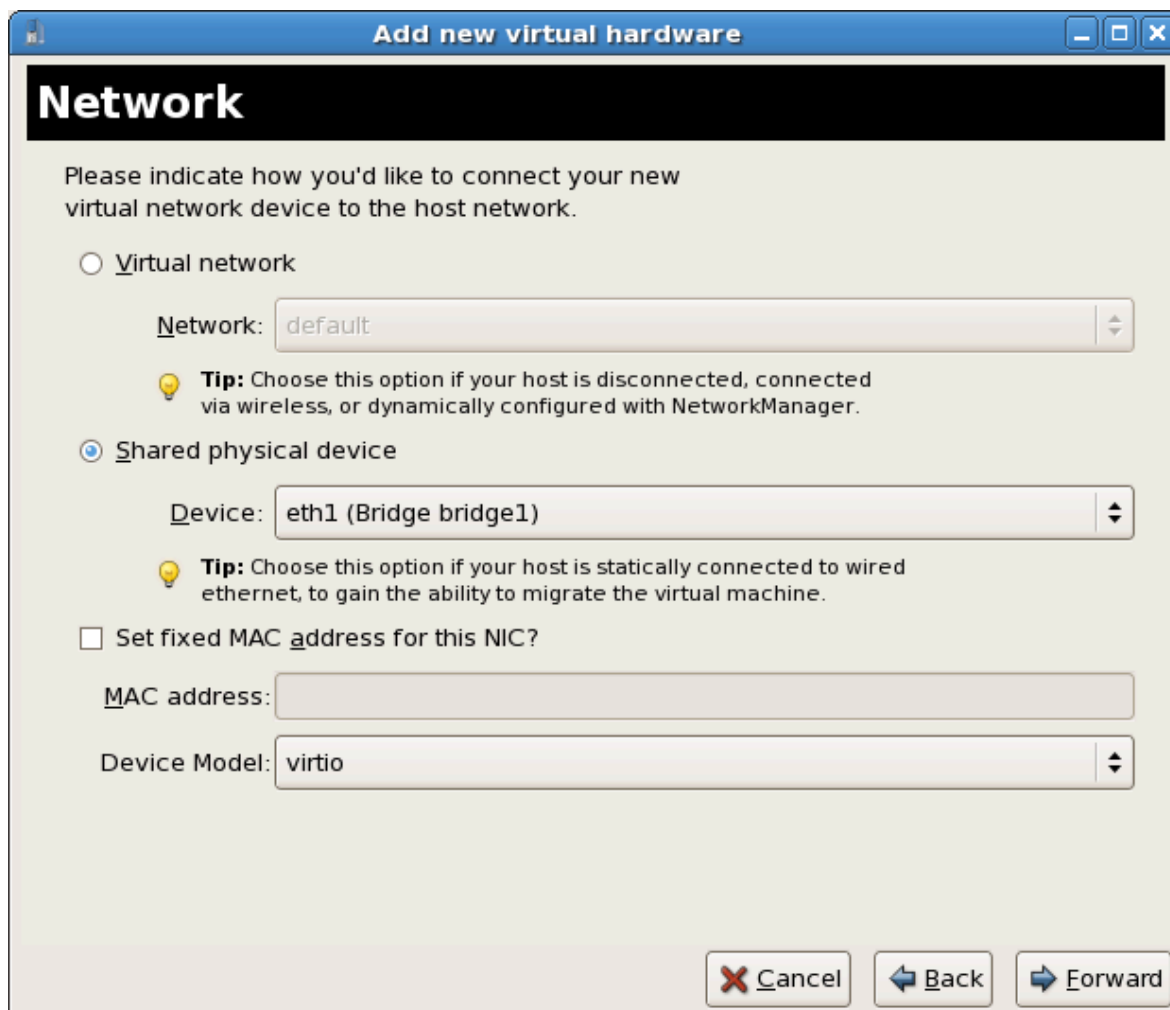
1. **Nieuw schijf station**

Selecteer het opslag apparaat of bestand-gebaseerde image. Selecteer **Virtio Disk** als het **Device type** en klik op **Forward**.



2. Nieuw netwerk apparaat

Selecteer **Virtual network** of **Shared physical device**. Selecteer **virtio** als het **Device type** en klik op **Forward**.



5. Klik op **Finish** om het apparaat op te slaan.



6. Start de guest opnieuw op. Het apparaat kan herkend worden door de Windows guest totdat het opnieuw opstart.

Deel III. Administration

Gevirtualiseerde systemen beheren

Deze hoofdstukken bevatten informatie over het beheren van de host en gevirtualiseerde guests met gebruik van de gereedschappen die onderdeel zijn van Fedora.

Guests beheren met xend

De **xend** node controle daemon voert bepaalde systeem beheer functies uit die gerelateerd zijn aan virtuele machines. Deze daemon controleert de gevirtualiseerde hulpbronnen, en **xend** moet draaien om interactie te hebben met virtuele machines. Voordat je **xend** opstart, moet je de werk parameters specificeren door het bewerken van het **xend** configuratie bestand **/etc/xen/xend-config.sxp**. Hier zijn de parameters die je kunt aan of uitzetten in het **xend-config.sxp** configuratie bestand:

Item	Description
(console-limit)	Bepaalt de geheugen buffer limiet xend_unix_server van de console server en kent waarden toe op een per domein basis.
(min-mem)	Bepaalt het minimum aantal megabytes dat gereserveerd wordt voor domain0 (als je 0 invult, verandert de waarde niet).
(dom0-cpus)	Bepaalt het aantal CPU's dat gebruikt wordt door domain0 (tenminste 1 CPU wordt standaard toegekend).
(enable-dump)	Bepaalt of er een dump komt als een crash optreedt (de standaard is 0).
(external-migration-tool)	Bepaalt het script of de toepassing die externe apparaat migratie afhandelt. Scripts moeten zich bevinden in etc/xen/scripts/external-device-migrate .
(logfile)	Bepaalt de locatie van het log bestand (standaard is /var/log/xend.log).
(loglevel)	Bepaalt de log mode instelling: DEBUG, INFO, WARNING, ERROR, of CRITICAL (standaard is DEBUG).
(network-script)	Bepaalt het script dat de netwerk omgeving aanzet (script moet zich bevinden in etc/xen/scripts directory).
(xend-http-server)	Zet de http stroom pakket beheer server aan (de standaard is nee)
(xend-unix-server)	Zet de unix domein socket server aan, de socket server is een communicatie eindpunt die laag-niveau netwerk verbindingen afhandelt en binnenkomende verbindingen accepteert of verwerpt. De standaard waarde is ja).
(xend-relocation-server)	Zet relocatie server aan voor cross-machine migraties (de standaard is nee).
(xend-unix-path)	Bepaalt de locatie waar het xend-unix-server commando data naar toe stuurt (standaard is var/lib/xend/xend-socket).
(xend-port)	Bepaalt de poort die de http beheer server gebruikt (de standaard is 8000).

Item	Description
(xend-relocation-port)	Bepaalt de poort die de relocatie server gebruikt (de standaard is 8002).
(xend-relocation-address)	Bepaalt de host adressen toegestaan voor migratie. De standaard waarde is de waarde van xend-address.
(xend-address)	Bepaalt het adres waarnaar de domein socket server verbindt. De standaard waarde staat alle verbindingen toe.

Tabel 10.1. xend configuratie parameters

Na het instellen van deze werk parameters, moet je verifiëren dat xend draait en indien niet, initialiseer je de daemon. Op de commando prompt kun je de **xend** daemon starten door het volgende in te typen:

```
service xend start
```

Je kunt **xend** gebruiken om de daemon te stoppen:

```
service xend stop
```

Dit stopt de daemon.

Je kunt **xend** gebruiken om de daemon te starten:

```
service xend restart
```

De daemon start weer op.

Je controleert de status van **xend** daemon:

```
service xend status
```

De output laat de status van de daemon zien.



xend aanzetten tijdens het opstarten

Gebruik het **chkconfig** commando om xend toe te voegen aan het **initscript**.

```
chkconfig --level 345 xend
```

xend zal nu starten in de runlevels 3, 4, en 5.

KVM guest tijds beheer

KVM gebruikt de constante Time Stamp Counter (TSC) eigenschap van vele moderne CPU's. Sommige CPU's hebben geen constante Time Stamp Counter wat de manier zal beïnvloeden die guest die met KVM draaien gebruiken om de tijd bij te houden. Guests die draaien zonder nauwkeurige tijdstelling hebben een serieus effect op sommige netwerk toepassingen omdat jouw guest sneller of langzamer zal draaien dan de actuele tijd.

Guests kunnen verscheidene problemen hebben veroorzaakt door onnauwkeurige klokken en tellers:

- Klokken kunnen synchronisatie met de actuele tijd verliezen wat sessies ongeldig maakt en netwerken beïnvloedt.
- Guests met langzame klokken kunnen problemen hebben met een migratie.
- Guests kunnen stoppen of crashen.

Deze problemen bestaan op andere virtualisatie platforms en de tijd moet altijd getest worden.



NTP

De Netwerk Tijd Protocol (NTP) daemon moet draaien op de host en de guests. Zet de ntpd service aan:

```
# service ntpd start
```

Voeg de ntpd service toe aan de opstart procedure:

```
# chkconfig ntpd on
```

Het gebruiken van de ntpd service moet de effecten van klok ongelijkheid in alle gevallen minimaliseren.

Bepalen of jouw CPU de contante Time Stamp Counter heeft

Houw CPU heeft een constante Time Stamp Counter als de constant_tsc vlag aanwezig is. Om te bepalen of jouw CPU de constant_tsc vlag heeft, voer je het volgende commando uit:

```
$ cat /proc/cpuinfo | grep constant_tsc
```

als er output wordt gegeven dan heeft jouw CPU de constant_tsc bit. Als geen output wordt gegeven volg je de instructies hieronder.

Hosts zonder een constante Time Stamp Counter instellen

Systemen zonder een constante tijd stempel teller vereisen extra instelling. Vermogensbeheer eigenschappen interfereren met nauwkeurig tijd bijhouden en moeten uitgezet worden voor guests om de tijd met KVM nauwkeurig bij te houden.



Note

Deze instructies zijn alleen voor AMD revisie F cpu's.

Als de CPU de `constant_tsc` bit mist, zet je alle vermogensbeheer eigenschappen uit ([BZ#513138](https://bugzilla.redhat.com/show_bug.cgi?id=513138)¹). Elk systeem heeft meerdere tellers die het gebruikt om de tijd bij te houden. De TSC is niet stabiel op de host, wat soms veroorzaakt wordt door `cpufreq` veranderingen, diepe C toestanden, of migratie naar een host met een snellere TSC. Om diepe C toestanden te stoppen, welke de TSC kunnen stoppen, voeg je "`processor.max_cstate=1`" toe aan de kernel boot opties in grub op de host:

```
term Fedora (vmlinuz-2.6.29.6-217.2.3.fc11)
  root (hd0,0)
  kernel /vmlinuz-vmlinuz-2.6.29.6-217.2.3.fc11 ro root=/dev/
VolGroup00/LogVol00 rhgb quiet processor.max_cstate=1
```

Zet `cpufreq` (alleen nodig op hosts zonder de `constant_tsc`) uit door het bewerken van het `/etc/sysconfig/cpuspeed` configuratie bestand en verander de `MIN_SPEED` en `MAX_SPEED` variabelen naar de hoogste beschikbare frequentie. Geldige waarden kunnen gevonden worden in de `/sys/devices/system/cpu/cpu*/cpufreq/scaling_available_frequencies` bestanden.

De para-gevirtualiseerde klok gebruiken met Red Hat Enterprise Linux guests

Voor bepaalde Red Hat Enterprise Linux guests zijn extra kernel parameters nodig. Deze parameters kunnen ingesteld worden door ze toe te voegen aan het einde van de `/kernel` regel in het `/boot/grub/grub.conf` bestand op de guest.

De tabel hieronder laat versies van Red Hat Enterprise Linux zien en de parameters die vereist zijn voor guest systemen zonder een constante Time Stamp Counter.

Red Hat Enterprise Linux	Extra guest kernel parameters
5.4 AMD64/Intel 64 met de para-gevirtualiseerde klok	Extra parameters zijn niet vereist
5.4 AMD64/Intel 64 zonder de para-gevirtualiseerde klok	<code>divider=10 notsc lpj=n</code>
5.4 x86 met de para-gevirtualiseerde klok	Extra parameters zijn niet vereist
5.4 x86 zonder de para-gevirtualiseerde klok	<code>divider=10 clocksource=acpi_pm lpj=n</code>
5.3 AMD64/Intel 64	<code>divider=10 notsc</code>
5.3 x86	<code>divider=10 clocksource=acpi_pm</code>
4.8 AMD64/Intel 64	<code>notsc divider=10</code>
4.8 x86	<code>clock=pmtmr divider=10</code>
3.9 AMD64/Intel 64	Extra parameters zijn niet vereist
3.9 x86	Extra parameters zijn niet vereist

¹ https://bugzilla.redhat.com/show_bug.cgi?id=513138

De para-gevirtualiseerde klok gebruiken met Windows guests

Zet de para-gevirtualiseerde klok op Windows guests aan door het bewerken van de boot parameters. Windows boot instellingen worden opgeslagen in het boot.ini bestand. Om the para-gevirtualiseerde klok aan te zetten voeg je de volgende regel toe:

```
/use pmtimer
```

Voor meer informatie over Windows boot instellingen en de pmtimer optie, refereer je naar [Available switch options for the Windows XP and the Windows Server 2003 Boot.ini files](#)².

² <http://support.microsoft.com/kb/833721>

KVM live migratie

Dit hoofdstuk behandelt migratie van guests die draaien onder een KVM hypervisor naar een andere KVM host.

Migratie is de naam voor het proces van het verplaatsen van een gevirtualiseerde guest van een host naar een andere. Migratie is de sleutel kenmerk van virtualisatie omdat software geheel gescheiden is van hardware. Migratie is nuttig voor:

- Load balancing - guests can be moved to hosts with lower usage when a host becomes overloaded.
- Hardware failover - when hardware devices on the host start to fail, guests can be safely relocated so the host can be powered down and repaired.
- Energy saving - guests can be redistributed to other hosts and host systems powered off to save energy and cut costs in low usage periods.
- Geographic migration - guests can be moved to another location for lower latency or in serious circumstances.

Migraties kunnen live of off-line uitgevoerd worden. Om guests te migreren moet de opslag gedeeld worden. Migratie werkt door het sturen van het guest geheugen naar de bestemming host. De gedeelde opslag slaat het standaard bestandssysteem van de guest op. De bestandssysteem image wordt niet over het netwerk verstuurd van de bron host naar de bestemming host.

An offline migration suspends the guest then moves an image of the guests memory to the destination host. The guest is resumed on the destination host and the memory the guest used on the source host is freed.

De tijd die een off-line migratie duurt hangt af van de netwerk bandbreedte en latentie. Een guest met 2 GB geheugen zal ongeveer tien seconden duren op een 1Gbit Ethernet link.

Een live migratie houdt de guest draaiende op de bron host en begint met het verhuizen van het geheugen zonder de guest te stoppen. Alle veranderde geheugen pagina's worden bijgehouden voor veranderingen en naar de bestemming gestuurd terwijl de image wordt verzonden. Het geheugen wordt vernieuwd met de veranderde pagina's. Dit proces vervolgt totdat de toegestane pauze tijd voor de guest gelijk is aan de voorspelde tijd om de laatste paar pagina's te versturen. KVM berekent dat en probeert om de maximale hoeveelheid pagina's van de bron naar de bestemming te sturen totdat we berekenen dat de hoeveelheid overblijvende pagina's overgebracht kan worden in de ingestelde tijd terwijl de VM wordt gepauzeerd. De registers worden op de nieuwe host geladen en de guest wordt dan hervat op de nieuwe host. Als de guest niet samengevoegd kan worden (wat gebeurt als guests een extreme werkdruk hebben), wordt de guest gepauzeerd en dat wordt een off-line migratie opgestart.

De tijd die een off-line migratie duurt hangt af van de netwerk bandbreedte en latentie. Als het netwerk zwaar belast is of een lage bandbreedte heeft zal de migratie veel langer duren.

12.1. Live migratie vereisten

Guests migreren vereist het volgende:

Migratie vereisten

- Een gevirtualiseerde guest geïnstalleerd op gedeelde netwerk opslag met gebruik van een van de volgende protocollen:
 - Fibre Channel
 - iSCSI
 - NFS
 - GFS2
- Twee of meer Fedora systemen van dezelfde versie met dezelfde vernieuwingen.
- Beide systemen moeten de juiste poorten open hebben staan.
- Beide systemen moeten identieke netwerk instellingen hebben. Alle brug en netwerk configuraties moeten exact hetzelfde zijn op beide hosts.
- Gedeelde opslag moet op dezelfde locatie aankoppelen op de bron en bestemming systemen. De aangekoppelde map naam moet identiek zijn.

Netwerk opslag configureren

Configureer gedeelde opslag en installeer een guest op de gedeelde opslag. gedeelde opslag instructies refereer je naar [Hoofdstuk 5, Gedeelde opslag en virtualisatie](#).

Als alternatief gebruik je het NFS voorbeeld in [Paragraaf 12.2, “Gedeelde opslag voorbeeld: NFS voor een eenvoudige migratie”](#).

12.2. Gedeelde opslag voorbeeld: NFS voor een eenvoudige migratie

Dit voorbeeld gebruikt NFS om guest images te delen met andere KVM hosts. dit voorbeeld is niet praktisch voor grote installaties, dit voorbeeld is alleen bedoeld om de migratie technieken te laten zien en voor toepassing in kleine opstellingen. Gebruik dit voorbeeld niet voor het migreren en draaien voor meer dan een paar gevirtualiseerde guests.

Voor gevorderde een meer robuuste opslag instructies, refereer je naar [Hoofdstuk 5, Gedeelde opslag en virtualisatie](#)

1. Exporteer je libvirt image map

Voeg de standaard image map toe aan het `/etc/exports` bestand:

```
/var/lib/libvirt/images *.bne.redhat.com(rw,no_root_squash,async)
```

Verander de host parameter zoals vereist door jouw omgeving.

2. Start NFS

a. Installeer de NFS pakketten als ze nog niet geïnstalleerd zijn:

```
# yum install nfs
```

- b. Open de poorten voor NFS in **iptables** en voeg NFS toe aan het **/etc/hosts.allow** bestand.
- c. Start de NFS service:

```
# service nfs start
```

3. Koppel de gedeelde opslag aan op de bestemming

Op het bestemming systeem koppel je de **/var/lib/libvirt/images** map aan:

```
# mount sourceURL:/var/lib/libvirt/images /var/lib/libvirt/images
```



Locaties moeten hetzelfde zijn op de bron en bestemming.

De map die gekozen is voor de guests moet exact overeenkomen op host en guest. Dit geldt voor alle types gedeelde opslag. De map moet hetzelfde zijn of de migratie zal mislukken.

12.3. Live KVM migratie met virsh

Een guest kan gemigreerd worden naar een andere host met het **virsh** commando. Het **migrate** commando accepteert parameters met het volgende formaat:

```
# virsh migrate --live GuestName DestinationURL
```

The *GuestName* parameter represents the name of the guest which you want to migrate.

The *DestinationURL* parameter is the URL or hostname of the destination system. The destination system must run the same version of Fedora, be using the same hypervisor and have **libvirt** running.

Once the command is entered you will be prompted for the root password of the destination system.

Voorbeeld: live migratie met virsh

Dit voorbeeld migreert van `test1.bne.redhat.com` naar `test2.bne.redhat.com`. Verander de host namen voor jouw omgeving. Dit voorbeeld migreert een virtuele machine met de naam **CentOS4test**.

Dit voorbeeld veronderstelt dat je volledig ingestelde gedeelde opslag hebt en voldoet aan alle voorwaarden (hier opgesomd: [Migratie vereisten](#)).

1. Verifieer of de guest draait

Vanaf het bron systeem, `test1.bne.redhat.com`, verifieer je of `CentOS4test` draait:

```
[root@test1 ~]# virsh list
Id Name                               State
-----
```

```
10 CentOS4          running
```

2. Migreer de guest

Voer het volgende commando uit om de guest live te migreren naar de bestemming, `test2.bne.redhat.com`. Voeg `/system` toe aan het eind van de bestemming URL om aan libvirt door te geven dat je volledige toegang nodig hebt.

```
# virsh migrate --live CentOS4test qemu+ssh://test2.bne.redhat.com/system
```

Once the command is entered you will be prompted for the root password of the destination system.

3. Wacht

De migratie kan enige tijd duren afhankelijk van de werkdruk en de grootte van de guest. `virsh` rapporteert alleen fouten. De guest vervolgt met het draaien op de bron host totdat de migratie voltooid is.

4. Verifieer of de guest aangekomen is op de bestemming host

Vanaf het bestemming systeem, `test2.bne.redhat.com`, verifieer je of `CentOS4test` draait:

```
[root@test2 ~]# virsh list
Id Name                State
-----
10 CentOS4            running
```

De live migratie is nu klaar.



Andere netwerk methodes

libvirt ondersteunt een verscheidenheid van netwerk methodes waaronder TLS/SSL, unix sockets, SSH, en niet-versleutelde TCP. Refereer naar [Hoofdstuk 13, Beheer op afstand van gevirtualiseerde guests](#) voor informatie over het gebruik van andere methodes.

12.4. Migreren met virt-manager

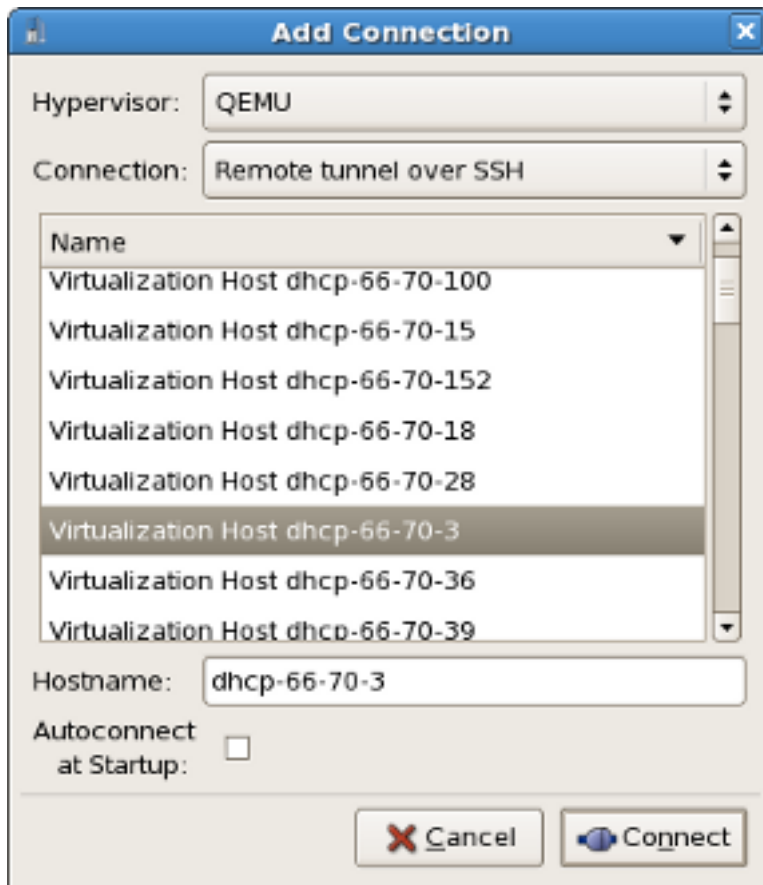
Deze paragraaf behandelt het migreren van op KVM gebaseerde guests met `virt-manager`.

1. Verbindt met de bron en doel hosts. In het **File** menu, klik je op **Add Connection**, en het **Add Connection** venster verschijnt.

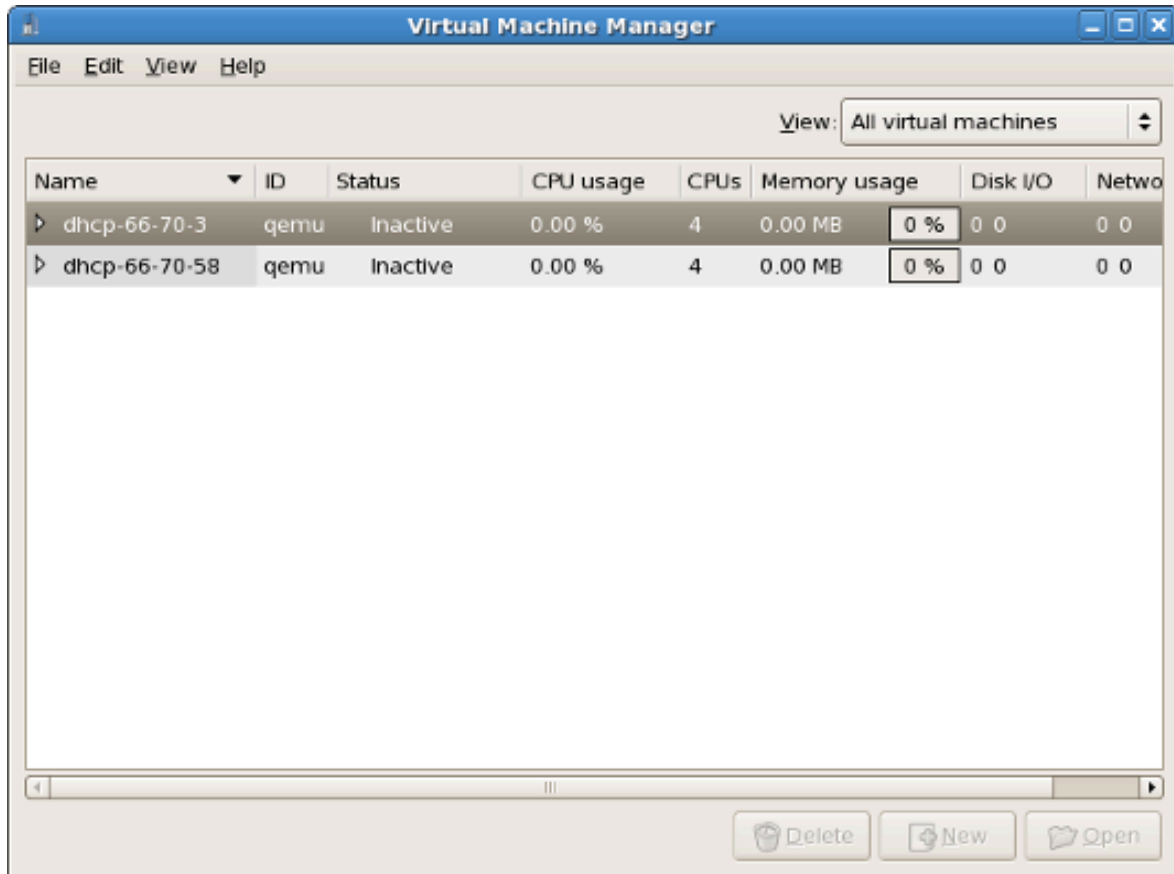
Vul de volgende details in:

- **Hypervisor:** Selecteer **QEMU**.
- **Connection:** Selecteer het verbindings type.
- **Hostname:** Vul de host naam in.

Klik op **Connect**.



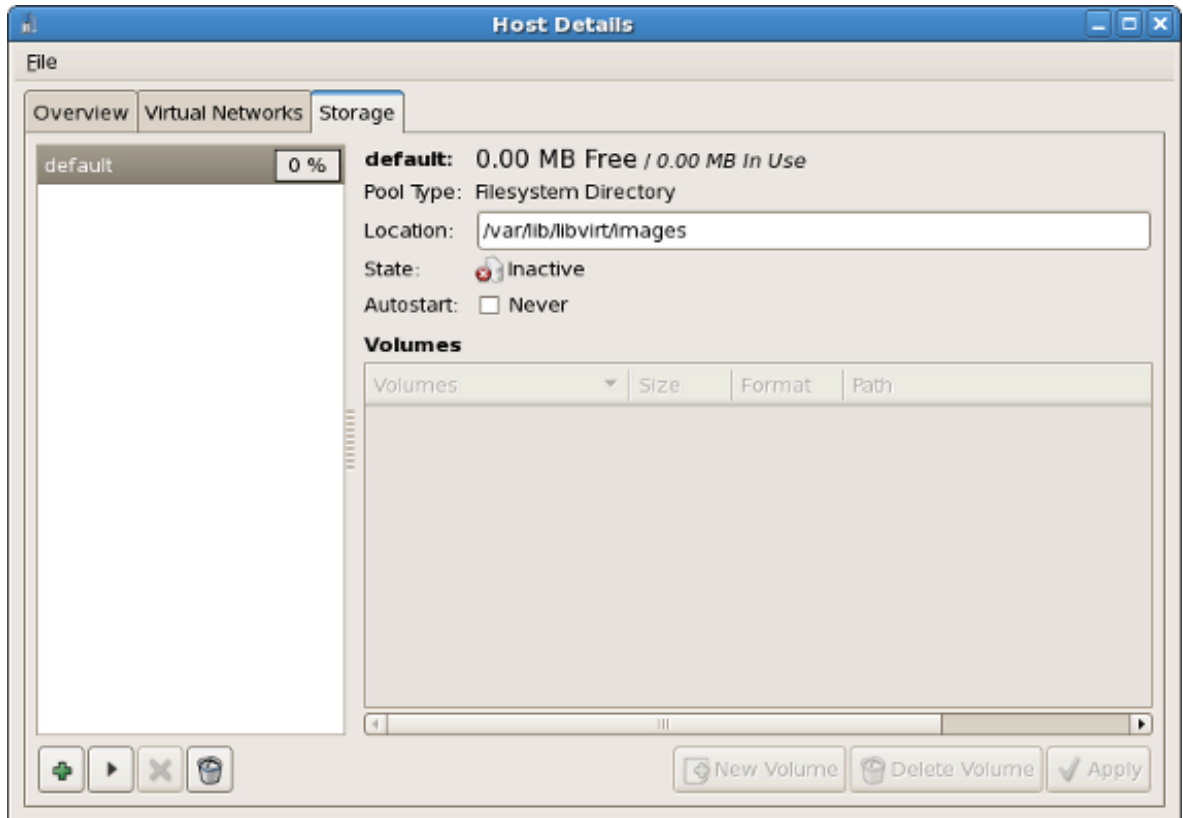
De Virtuele machine beheerder laat een lijst van verbonden hosts zien.



2. Voeg een opslag pool toe met dezelfde NFS aan de bron en bestemming hosts.

In het **Edit** menu, klik je op **Host Details**, het Host details venster verschijnt.

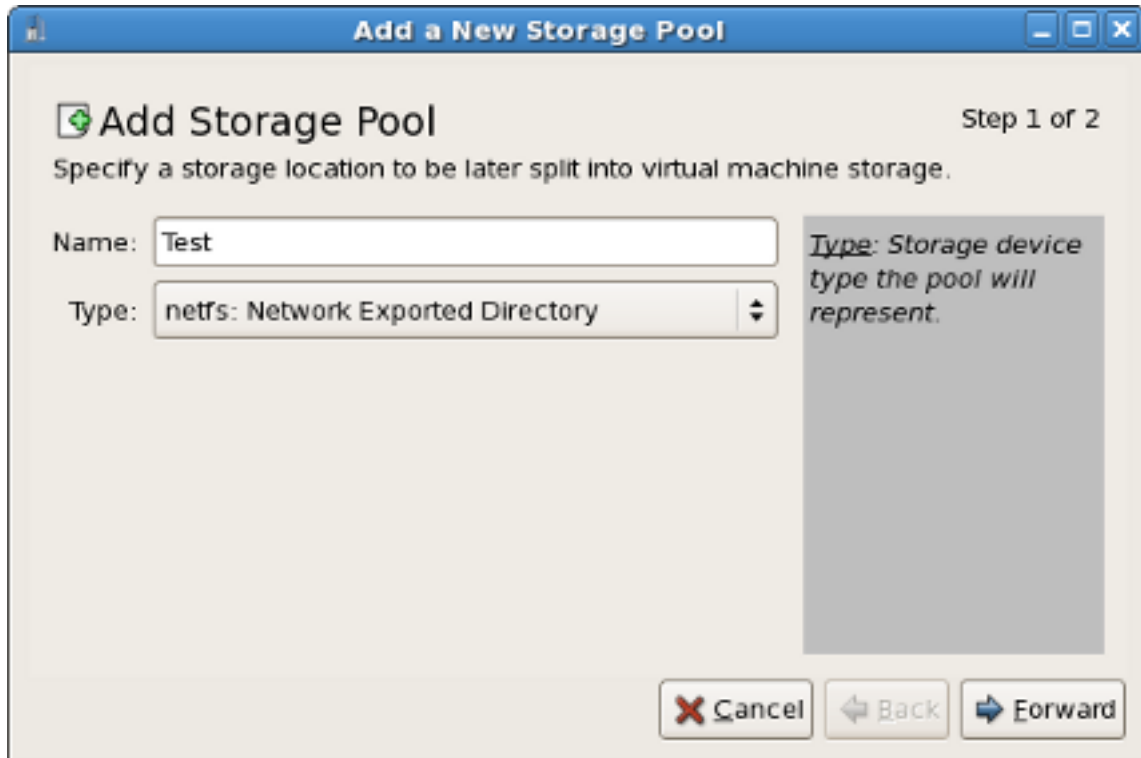
Klik op de **Storage** tab.



3. Voeg een nieuwe opslag pool toe. In de linker onder hoek van het venster, klik je op de + knop. Het Add a new storage pool venster verschijnt.

Vul de volgende details in:

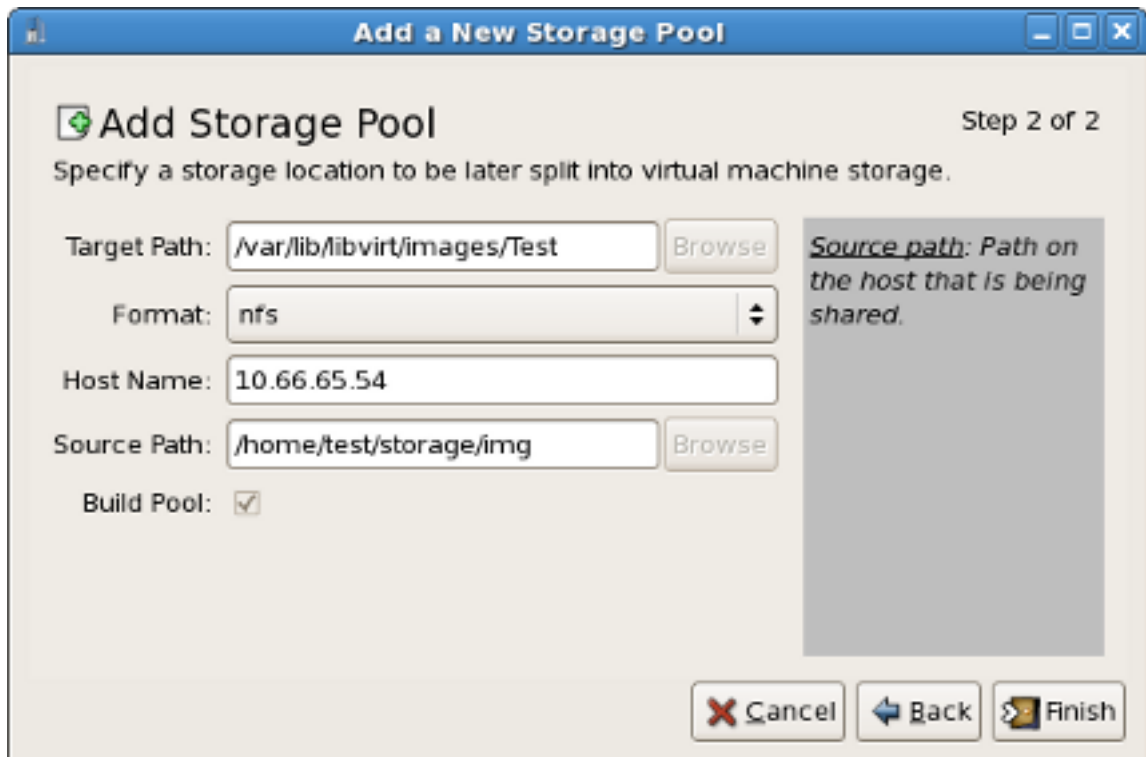
- **Name:** Vul de naam van de opslag pool in.
- **Type:** Selecteer **netfs: Network Exported Directory**.



Klik op **Forward**.

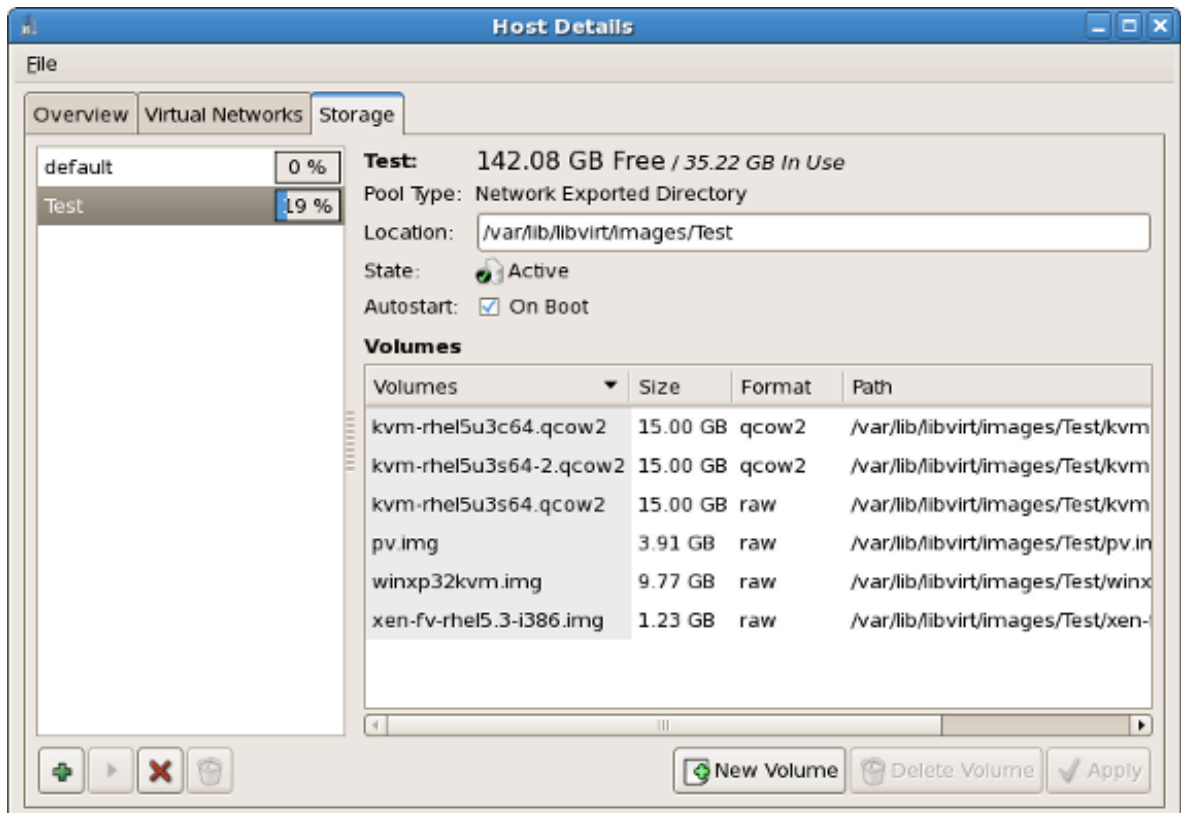
4. Vul de volgende details in:

- **Format:** Selecteer het opslag type. Dit moet NFS of iSCSI zijn voor live migraties.
- **Host Name:** Vul het IP adres of volledig gekwalificeerde domein naam van de opslag server in.

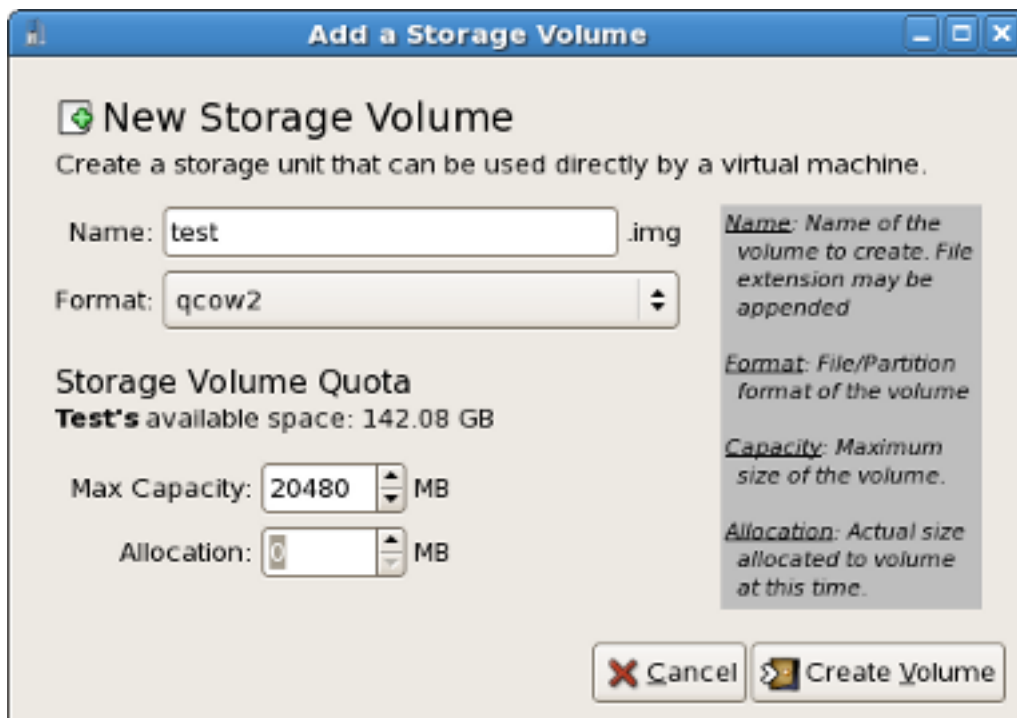


Klik op **Finish**.

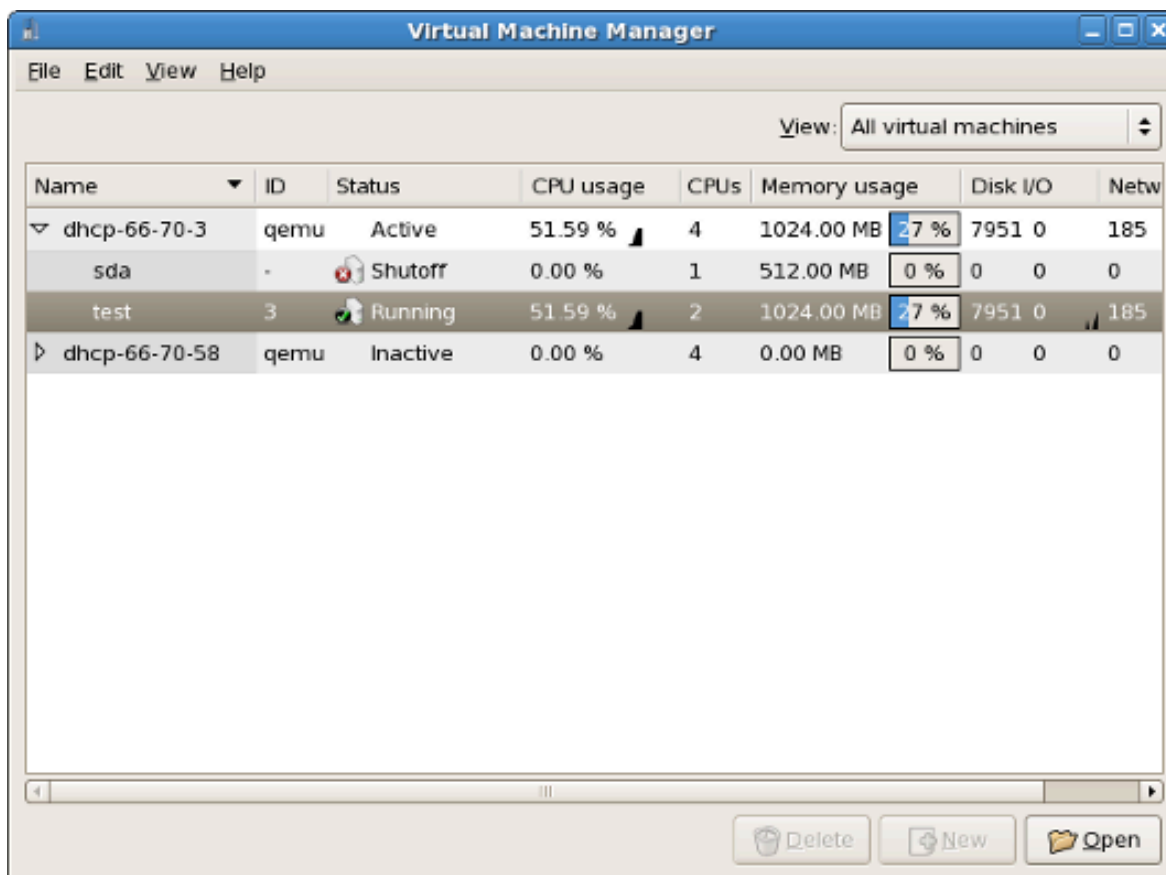
5. Maak een nieuw volume aan in de gedeelde opslag pool, klik op **New Volume**.



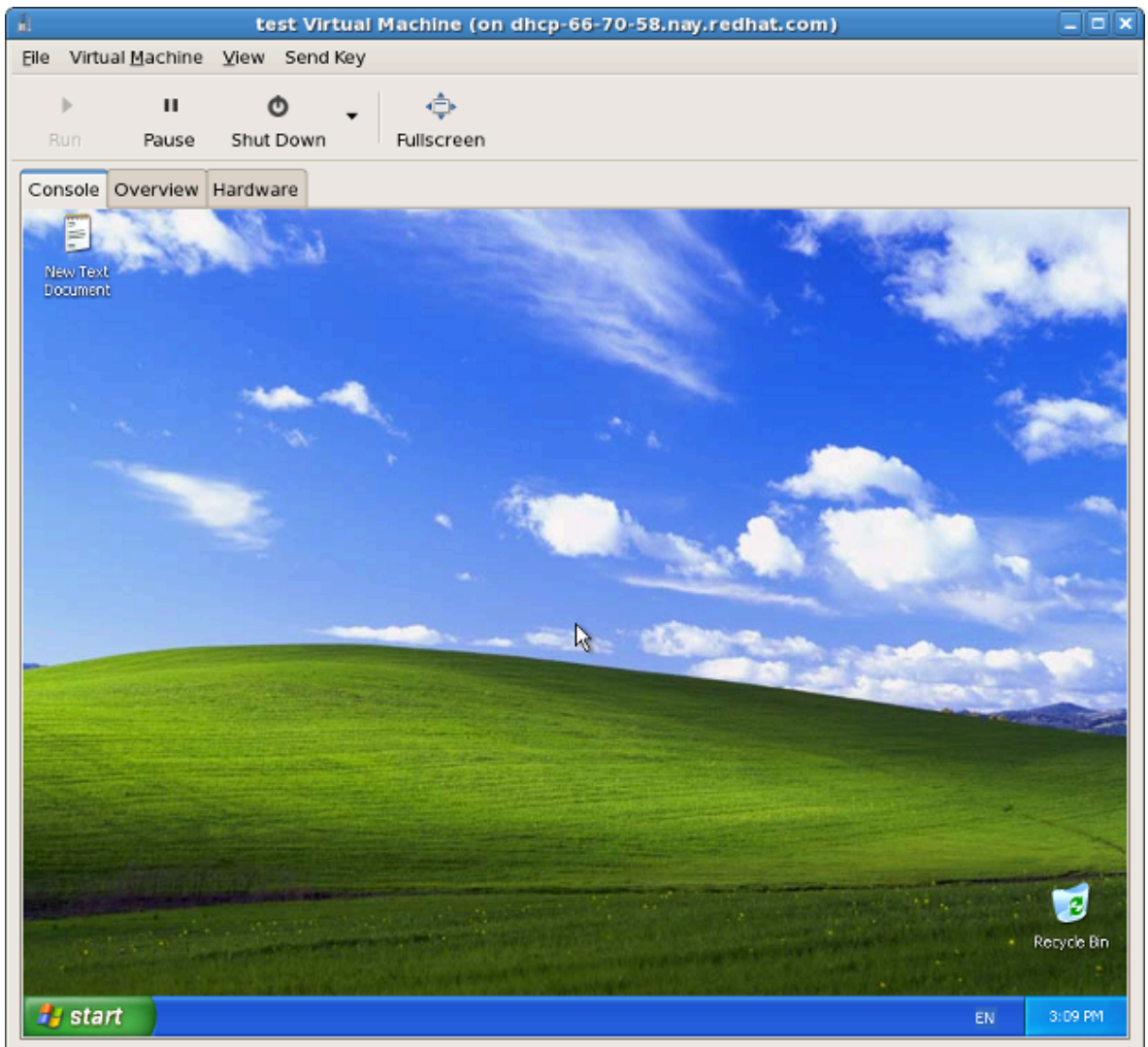
- 6. Vul de details in, en klik dan op **Create Volume**.



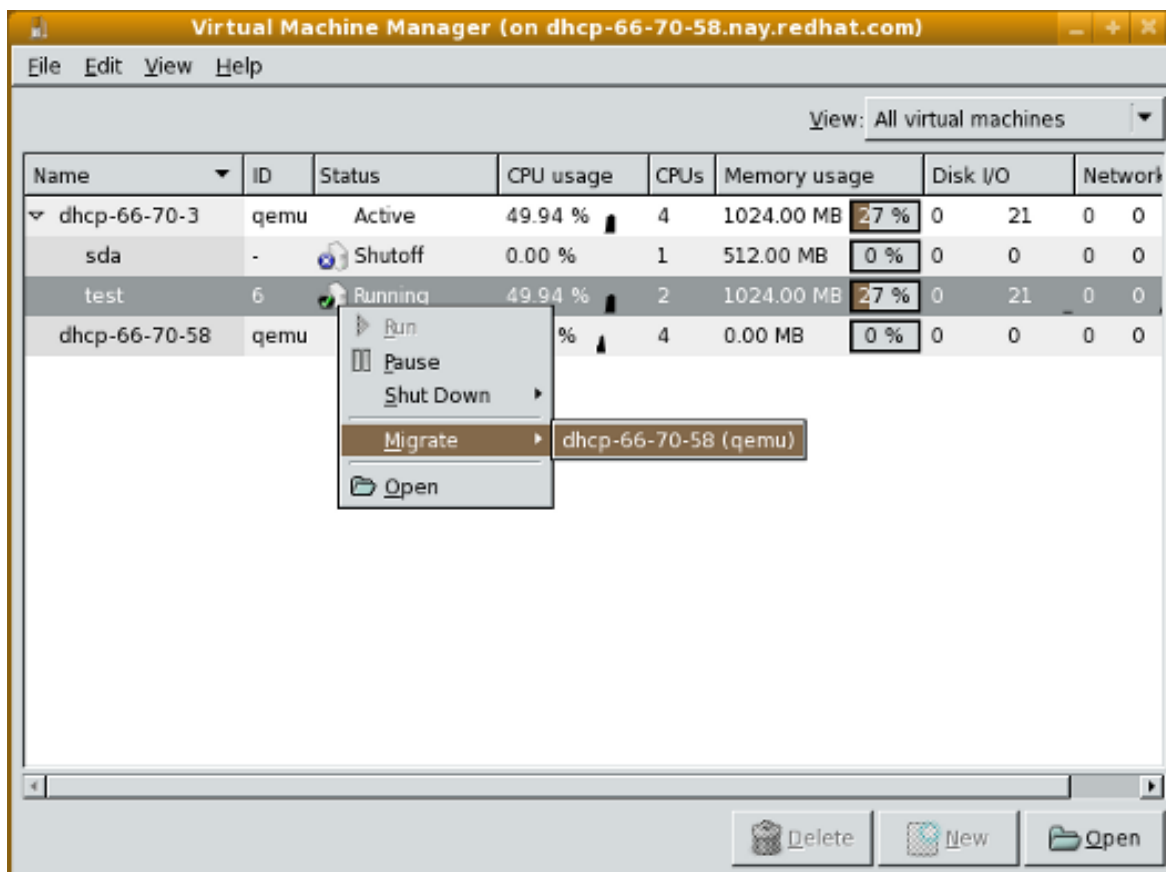
- 7. Maak een virtuele machine met de nieuwe volume, en voer de virtuele machine uit.



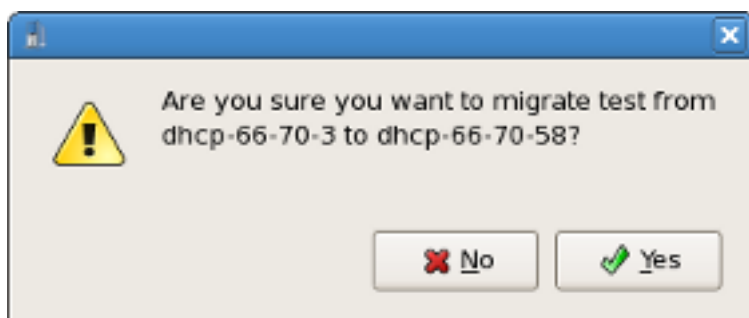
Het Virtuele Machine venster verschijnt.



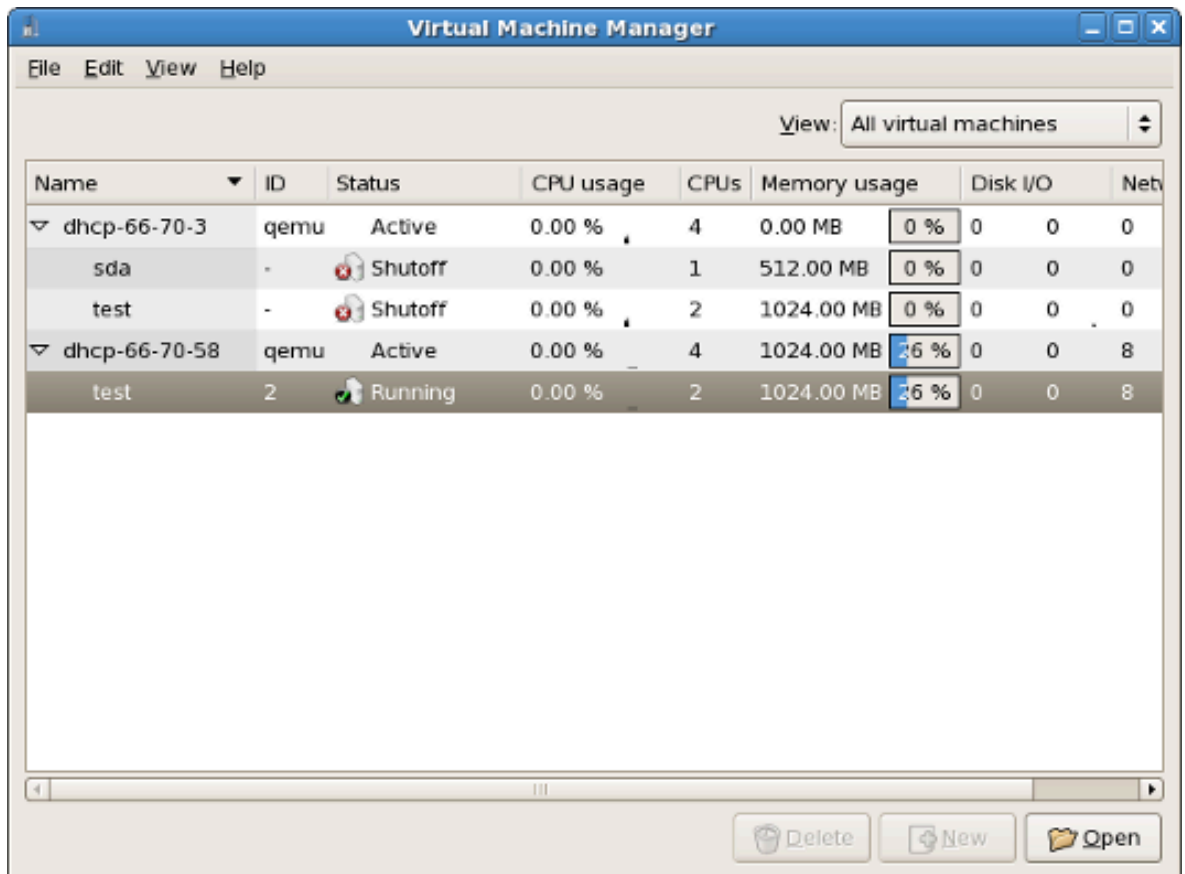
8. In het Virtuele machine beheer venster, klik je rechts op de virtuele machine, je selecteert **Migrate**, en daarna klik je op de migratie locatie.



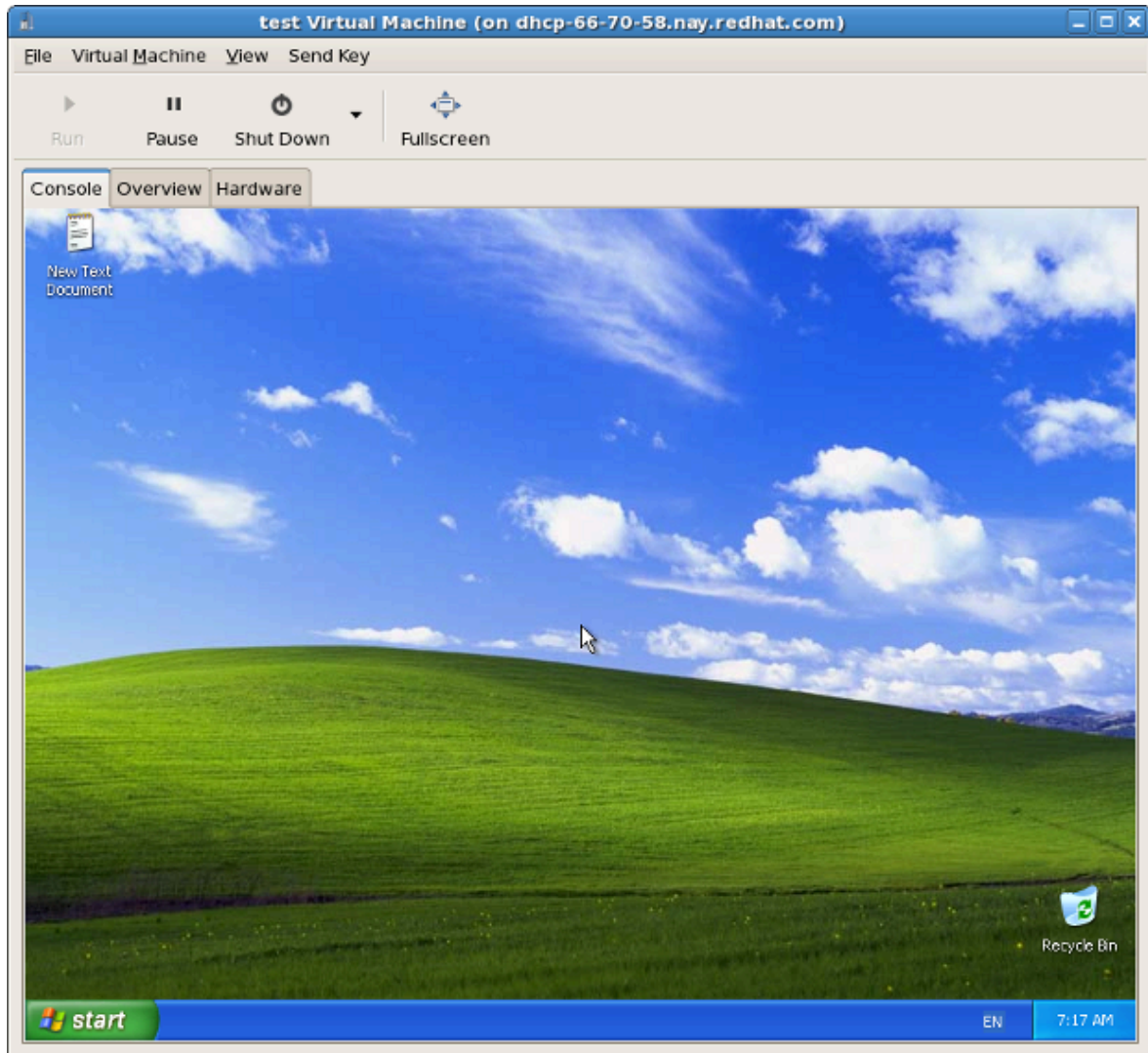
9. Klik op **Yes** om de migratie te bevestigen.



De Virtuele machine beheerder laat de virtuele machine op zijn nieuwe locatie zien.



Het Virtuele machine venster laat de nieuwe virtuele machine locatie zien.



Beheer op afstand van gevirtualiseerde guests

Deze paragraaf legt uit hoe je jouw gevirtualiseerde guests op afstand beheert met gebruik van **ssh** of TLS en SSL.

13.1. Beheer op afstand met SSH

Het **ssh** pakket biedt een versleuteld netwerk protocol welke veilig beheer functies kan versturen naar virtualisatie server op afstand. De beschreven methode gebruikt de **libvirt** beheer verbinding welke veilig met een tunnel een **SSH** verbinding maakt om machines op afstand te beheren. Alle authenticatie wordt gedaan met het gebruik van **SSH** publieke sleutel versleuteling en wachtwoorden of wachzinnen verzameld door je lokale **SSH** agent. Bovendien wordt de **VNC** console voor iedere guest virtuele machine getunneld met **SSH**.

SSH wordt gewoonlijk standaard geconfigureerd dus je hebt waarschijnlijk al SSH sleutels ingesteld en zijn er geen extra firewall regels nodig om toegang te krijgen tot de beheer service of **VNC** console.

Wees bedacht op de problemen met het gebruik van **SSH** voor het beheren op afstand van je virtuele machines, waaronder:

- je hebt root login toegang nodig voor de machine op afstand voor het beheren van virtuele machines,
- de initiële verbinding instelling kan traag zijn,
- er is geen standaard of triviale manier om de sleutel van een gebruiker op alle hosts of guests in te trekken, en
- ssh schaal niet goed met grote aantallen machines op afstand.

SSH configureren voor toegang tot virt-manager

De volgende instructies nemen aan je van nul af begint en nog geen **SSH** sleutels ingesteld hebt.

1. Je hebt een publieke sleutel paar nodig op de machine waarop **virt-manager** wordt gebruikt. Als **ssh** al ingesteld is kun je dit commando overslaan.

```
$ ssh-keygen -t rsa
```

2. Om login op afstand toe te staan, heeft **virt-manager** een kopie van de publieke sleutel nodig voor elke machine op afstand welke **libvirt** draait. Kopieer het bestand **\$HOME/.ssh/id_rsa.pub** van de machine die je wilt gebruiken voor beheer op afstand met het **scp** commando:

```
$ scp $HOME/.ssh/id_rsa.pub root@somehost:/root/key-dan.pub
```

3. Nadat het gekopieerd is, gebruik je **ssh** om als root met de machines op afstand te verbinden en je voegt het bestand dat je gekopieerd hebt naar de lijst van goedgekeurde sleutels. Als de root gebruiker op de host op afstand nog geen lijst van goedgekeurde sleutels heeft, wees er dan zeker van dat je de rechten correct instelt.

```
$ ssh root@somehost
# mkdir /root/.ssh
# chmod go-rwx /root/.ssh
# cat /root/key-dan.pub >> /root/.ssh/authorized_keys
# chmod go-rw /root/.ssh/authorized_keys
```

De libvirt daemon (libvirtd)

De libvirt daemon biedt een interface voor het beheren van virtuele machines. Je moet de libvirtd daemon gebruiken welke geïnstalleerd moet zijn en draait op elke host op afstand die je moet beheren. Het gebruik van het Fedora **kernel-xen** pakket een speci

```
$ ssh root@somehost
# chkconfig libvirtd on
# service libvirtd start
```

Nadat libvirtd en **SSH** zijn ingesteld moet je in staat zijn om op afstand toegang te krijgen tot je virtuele machines en te kunnen beheren. Je moet op dit punt ook in staat zijn om toegang tot jouw guests te krijgen met **VNC**.

13.2. Beheer op afstand met TLS en SSL

Je kunt virtuele machine beheren met gebruik van TSL en SSL. TLS en SSL biedt een betere schaalbaarheid maar is gecompliceerder dan ssh (refereer naar [Paragraaf 13.1, "Beheer op afstand met SSH"](#)). TLS en SSL is dezelfde technologie als gebruikt door web browsers voor veilige verbindingen. De **libvirt** beheer verbinding open een TCP poort voor binnenkomende verbindingen, welke veilig wordt versleuteld en voor echt verklaard op basis van x509 certificaten. bovendien zal de VNC console voor elke guest virtuele machine ingesteld worden om TLS te gebruiken met x509 certificaat authenticatie.

Deze methode vereist geen shell accounts op de machine op afstand die beheerd wordt. Er zijn echter extra firewall regels nodig om toegang te krijgen to de beheer service of VNC console. Certificaat herroeping lijsten kunnen toegang van gebruikers intrekken.

Stappen om TLS/SSL toegang voor virt-manager instellen

De volgende korte gids neemt aan dat je van nul af begint en je geen enkele TLS/SSL certificaat kennis hebt. Als je het geluk hebt om een certificaat beheer server te hebben, kun je waarschijnlijk de eerste stappen overslaan.

libvirt server instelling

Voor meer informatie over het aanmaken van certificaten, refereer je naar de **libvirt** website, <http://libvirt.org/remote.html>.

Xen VNC Server

De Xen VNC server kan TLS aangezet hebben door het bewerken van het configuratie bestand, **/etc/xen/xend-config.sxp**. Verwijder het commentaar bij de **(vnc-tls 1)** configuratie parameter in het configuratie bestand.

De **/etc/xen/vnc** heeft de volgende drie bestanden nodig:

- **ca-cert.pem** - De CA certificaat
- **server-cert.pem** - De Server certificaat getekend door de CA
- **server-key.pem** - De server privé sleutel

Dit biedt versleuteling voor het data kanaal. Het is geschikt om te eisen dat cliënten hun eigen x509 certificaat presenteren als een vorm van authenticatie. Om dit aan te zetten verwijder je het commentaar op de (**vnc-x509-verify 1**) parameter.

virt-manager en **virsh** cliënt instelling

De instelling voor cliënten is licht inconsistent op dit moment. Om de **libvirt** beheer API over TLS aan te zetten, moeten de CA en cliënt certificaten geplaatst worden in **/etc/pki**. voor details hierover raadpleeg je <http://libvirt.org/remote.html>

In de **virt-manager** gebruikers interface, gebruik je de '**SSL/TLS**' transport mechanisme optie bij het verbinden naar een host.

Voor **virsh** heeft de URI het volgende formaat:

- **qemu://hostname.guestname/system** voor KVM.
- **xen://hostname.guestname/** voor Xen.

Om SSL en TLS aan te zetten voor VNC, is het nodig om de certificaat autoriteit en cliënt certificaten te plaatsen in **\$HOME/.pki**, dat wil zeggen de volgende drie bestanden:

- CA of **ca-cert.pem** - De CA certificaat.
- **libvirt-vnc** of **clientcert.pem** - De cliënt certificaat getekend door de CA.
- **libvirt-vnc** of **clientkey.pem** - De cliënt privé sleutel.

13.3. Transport modes

Voor beheer op afstand ondersteunt **libvirt** de volgende transport modes:

Transport Layer Security (TLS)

Transport Layer Security TLS 1.0 (SSL 3.1) is een TCP/IP socket met authenticatie en versleuteling die gewoonlijk op een publieke poort luistert. Om dit te gebruiken zul je een cliënt en server certificaten moeten aanmaken. De standaard poort is 16514.

UNIX sockets

Unix domein sockets zijn alleen toegankelijk op de locale machine. Sockets zijn niet versleuteld, en gebruiken UNIX rechten of SELinux voor authenticatie. De standaard socket namen zijn **/var/run/libvirt/libvirt-sock** en **/var/run/libvirt/libvirt-sock-ro** (voor alleen-lezen verbindingen).

SSH

Transport over een Secure Shell protocol (SSH) verbinding. Vereist dat Netcat (het *nc* pakket) geïnstalleerd is. De libvirt daemon (**libvirtd**) moet op de machine op afstand draaien. Poort

22 moeten open zijn voor SSH toegang. Je moet een vorm van ssh sleutel beheer gebruiken (bijvoorbeeld, het **ssh-agent** programma) of je zult gevraagd worden naar een wachtwoord.

ext

De ext parameter wordt gebruikt voor elk extern programma dat een verbinding kan maken naar de machine op afstand op een manier die buiten de reikwijdte van libvirt ligt. Dit zijn gewoonlijk niet-ondersteunde beveiliging toepassingen van derden.

tcp

Niet-versleutelde TCP/IP socket. Wordt niet aanbevolen voor productie gebruik. Het is normaal uitgezet, maar een beheerder kan het aanzetten voor testen of voor gebruik op een vertrouwd netwerk. De standaard poort is 16509.

Het standaard transport gebeurt met tls, behalve als iets anders opgegeven is.

URI's op afstand

Een Uniform Resource Identifier (URI) wordt gebruikt door virsh en libvirt om te verbinden met een host op afstand. URI's kunnen ook gebruikt worden met de --connect parameter voor het virsh commando om enkele commando's uit te voeren of voor migraties op hosts op afstand.

libvirt URI's hebben de algemene vorm (inhoud tussen rechte haken, "[]", representeren optionele functies):

```
driver[+transport]://[gebruikersnaam@][hostnaam][:poort]/[pad][?extraparameters]
```

Of de transport methode, of de host naam moet opgegeven worden om dit te onderscheiden van een locale URI.

Voorbeelden van beheer parameters op afstand

- Verbindt met een Xen hypervisor op afstand op de host met de naam towada, met gebruik van SSH transport en de SSH gebruikersnaam ccurran.

```
xen+ssh://ccurran@towada/
```

- Verbindt met een Xen hypervisor op afstand op een host met de naam towada met gebruik van TLS.

```
xen://towada/
```

- Verbindt met een Xen hypervisor op afstand op host towada met gebruik van TLS. De `no_verify=1` parameter vertelt libvirt om het certificaat van de server niet te verifiëren.

```
xen://towada/?no_verify=1
```

- Verbindt met een KVM hypervisor op afstand op host towada met gebruik van SSH.

```
qemu+ssh://towada/system
```

Test voorbeelden

- Verbindt met de lokale KVM hypervisor met een niet-standaard UNIX socket. Het volledige pad naar de Unix socket wordt in dit geval expliciet opgegeven.

```
qemu+unix:///system?socket=/opt/libvirt/run/libvirt/libvirt-sock
```

- Verbindt met de libvirt daemon met een niet-versleutelde TCP/IP verbinding naar de server met het IP adres 10.1.1.10 op poort 5000. dit gebruikt de test driver met standaard instellingen.

```
test+tcp://10.1.1.10:5000/default
```

Extra URI parameters

Extra parameters kunnen toegevoegd worden aan URI's op afstand. De tabel hieronder [Tabel 13.1, "Extra URI parameters"](#) behandelt de herkende parameters, Alle andere parameters worden genegeerd. Merk op dat parameter waarden een URI escape karakter moeten hebben (dat betekent, voorafgegaan door een vraagteken (?)) en speciale karakters worden geconverteerd naar het URI formaat)

Naam	Transport mode	Description	Voorbeeld gebruik
name	alle modes	De naam doorgegeven aan de virConnectOpen functie op afstand. De naam wordt normaal gevormd door het verwijderen van transport, host naam, poort nummer, gebruikersnaam en extra parameters van de URI op afstand, maar in bepaalde complexe gevallen kan het beter zijn om de naam expliciet op te geven.	name=qemu:///system
command	ssh en ext	Het externe commando. Voor ext transport is dit vereist. Voor ssh is de standaard ssh. Het PATH wordt afgezocht voor het commando.	command=/opt/openssh/bin/ssh
socket	unix en ssh	Het pad naar de UNIX domein socket, welke	socket=/opt/libvirt/run/libvirt/libvirt-sock

Naam	Transport mode	Description	Voorbeeld gebruik
		de standaard terzijde schuift. Voor ssh transport wordt dit doorgegeven aan het netcat commando op afstand (zie netcat).	
netcat	ssh	De naam van het netcat commando op de machine op afstand. De standaard is nc. Voor ssh transport maakt libvirt een ssh commando die eruit ziet als: command -p poort [-l gebruikersnaam] hostnaam netcat -U socket waarin poort, gebruikersnaam, hostnaam opgegeven kunnen worden als onderdeel van de URI op afstand URI, en command, netcat en socket komen van extra parameters (of zinnige standaarden).	netcat=/opt/netcat/bin/nc
no_verify	tls	Als het op een niet-nul waarde wordt ingesteld, zet dit de controle uit die de cliënt uitvoert naar het certificaat van de server. Merk op dat het uitzetten van de controle die de server uitvoert naar het certificaat of IP adres van de cliënt je de libvirt configuratie moet veranderen.	no_verify=1
no_tty	ssh	Als het op een niet-nul waarde wordt ingesteld, stopt ssh met het vragen naar een wachtwoord als het niet automatisch kan inloggen op de machine op afstand	no_tty=1

Naam	Transport mode	Description	Voorbeeld gebruik
		(voor met gebruik van ssh-agent of soortgelijk). Gebruik dit als je geen toegang tot een console hebt - bijvoorbeeld in grafische programma's die libvirt gebruiken.	

Tabel 13.1. Extra URI parameters

Deel IV. Virtualisatie referentie gids

Virtualisatie commando's, systeem gereedschappen, toepassingen en aanvullende systeem referentie

Deze hoofdstukken bieden gedetailleerde beschrijvingen van virtualisatie commando's, systeem gereedschappen, en toepassingen die onderdeel zijn van Fedora. Deze hoofdstukken zijn gemaakt voor gebruikers die informatie nodig hebben over geavanceerde functionaliteit en andere kenmerken.

Virtualisatie gereedschappen

Het volgende is een lijst van gereedschappen voor virtualisatie beheer, debuggen, en netwerk gereedschappen die nuttig zijn voor systemen die Xen draaien.

Systeem beheer gereedschappen

- **vmstat**
- **iostat**
- **lsof**

```
# lsof -i :5900
xen-vncfb 10635 root 5u IPv4 218738 TCP
grumble.boston.redhat.com:5900 (LISTEN)
```

- **qemu-img**

Geavanceerde debug gereedschappen

- **systemTap**
- **crash**
- **xen-gdbserver**
- **sysrq**
- **sysrq t**
- **sysrq w**
- **sysrq c**

Netwerk

brctl

- ```
brctl show
bridge name bridge id STP enabled interfaces
xenbr0 8000.fefffffffffff no vif13.0
 pdummy0
 vif0.0
```
- ```
# brctl showmacs xenbr0
port no  mac addr          is local?  aging timer
  1      fe:ff:ff:ff:ff:ff  yes        0.00
```
- ```
brctl showstp xenbr0
xenbr0
bridge id 8000.fefffffffffff
designated root 8000.fefffffffffff
```

|                       |                     |                      |
|-----------------------|---------------------|----------------------|
| root port             | 0                   | path cost            |
| 0                     |                     |                      |
| max age               | 20.00               | bridge max age       |
| 20.00                 |                     |                      |
| hello time            | 2.00                | bridge hello time    |
| 2.00                  |                     |                      |
| forward delay         | 0.00                | bridge forward delay |
| 0.00                  |                     |                      |
| aging time            | 300.01              |                      |
| hello timer           | 1.43                | tcn timer            |
| 0.00                  |                     |                      |
| topology change timer | 0.00                | gc timer             |
| 0.02                  |                     |                      |
| flags                 |                     |                      |
| vif13.0 (3)           |                     |                      |
| port id               | 8003                | state                |
| forwarding            |                     |                      |
| designated root       | 8000.feffffffffffff | path cost            |
| 100                   |                     |                      |
| designated bridge     | 8000.feffffffffffff | message age timer    |
| 0.00                  |                     |                      |
| designated port       | 8003                | forward delay timer  |
| 0.00                  |                     |                      |
| designated cost       | 0                   | hold timer           |
| 0.43                  |                     |                      |
| flags                 |                     |                      |
| pdummy0 (2)           |                     |                      |
| port id               | 8002                | state                |
| forwarding            |                     |                      |
| designated root       | 8000.feffffffffffff | path cost            |
| 100                   |                     |                      |
| designated bridge     | 8000.feffffffffffff | message age timer    |
| 0.00                  |                     |                      |
| designated port       | 8002                | forward delay timer  |
| 0.00                  |                     |                      |
| designated cost       | 0                   | hold timer           |
| 0.43                  |                     |                      |
| flags                 |                     |                      |
| vif0.0 (1)            |                     |                      |
| port id               | 8001                | state                |
| forwarding            |                     |                      |
| designated root       | 8000.feffffffffffff | path cost            |
| 100                   |                     |                      |
| designated bridge     | 8000.feffffffffffff | message age timer    |
| 0.00                  |                     |                      |
| designated port       | 8001                | forward delay timer  |
| 0.00                  |                     |                      |

---

```
designated cost 0 hold timer
0.43
flags
```

- **ifconfig**
- **tcpdump**

KVM gereedschappen

- **ps**
- **pstree**
- **top**
- **kvmtrace**
- **kvm\_stat**

Xen gereedschappen

- **xentop**
- **xm dmesg**
- **xm log**



---

# Guests beheren met virsh

**virsh** is een commando-regel interface gereedschap voor het beheren van guests en de hypervisor.

Het **virsh** gereedschap is gebouwd op de **libvirt** beheer API en werkt als een alternatief voor het **xm** commando en de grafische guest manager (**virt-manager**). **virsh** kan gebruikt worden in de alleen-lezen mode door gebruikers zonder rechten. Je kunt **virsh** gebruiken om scripts uit te voeren voor de guest machines.

## virsh commando snelle referentie

De volgende tabellen bieden een snelle referentie voor alle virsh commando-regel opties

| Commando        | Description                                                                |
|-----------------|----------------------------------------------------------------------------|
| <b>help</b>     | Laat de basis hulp informatie zien.                                        |
| <b>list</b>     | Laat alle guests zien.                                                     |
| <b>dumpxml</b>  | Output het XML configuratie bestand van de guest.                          |
| <b>create</b>   | Maakt een guest van een XML configuratie bestand en start de nieuwe guest. |
| <b>start</b>    | Start een niet-actieve guest                                               |
| <b>destroy</b>  | Forceert een guest om te stoppen.                                          |
| <b>define</b>   | Output een XML configuratie bestand van een guest.                         |
| <b>domid</b>    | Laat de ID van de guest zien.                                              |
| <b>domuuid</b>  | Laat de UUID van de guest zien.                                            |
| <b>dominfo</b>  | Laat guest informatie zien.                                                |
| <b>domname</b>  | Laat de naam van de guest zien.                                            |
| <b>domstate</b> | Laat de toestand van een guest zien,                                       |
| <b>quit</b>     | Verlaat de interactieve terminal.                                          |
| <b>reboot</b>   | Start een guest opnieuw op.                                                |
| <b>restore</b>  | Laadt een vroeger opgeslagen quest vanuit het opslag bestand opnieuw in.   |
| <b>resume</b>   | Vervolg een in pauze gezette guest.                                        |
| <b>save</b>     | Sla de huidige toestand van een guest op in een bestand.                   |
| <b>shutdown</b> | Sluit een guest netjes af.                                                 |
| <b>suspend</b>  | Zet een guest in pauze stand.                                              |
| <b>undefine</b> | Verwijder alle bestanden geassocieerd met een guest.                       |
| <b>migrate</b>  | Migreer een guest naar een andere host.                                    |

Tabel 15.1. Guest beheer commando's

De volgende zijn **virsh** commando opties voor het beheren van guests en hypervisor hulpbronnen.

| Commando                | Description                                                                                                          |
|-------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>setmem</b>           | Stel het toegewezen geheugen in voor een guest.                                                                      |
| <b>setmaxmem</b>        | Stel de maximum geheugen limiet in voor de hypervisor.                                                               |
| <b>setvcpus</b>         | Verander het aantal virtuele CPU's toegekend aan een guest.                                                          |
| <b>vcpuinfo</b>         | Laat virtuele CPU informatie van een guest zien.                                                                     |
| <b>vcpupin</b>          | Controleert de virtuele CPU affiniteit van een guest.                                                                |
| <b>domblockstat</b>     | Laat blok apparaat statistieken zien van een draaiende guest.                                                        |
| <b>domifstat</b>        | Laat netwerk interface statistieken zien van een draaiende guest.                                                    |
| <b>attach-device</b>    | Ken een apparaat toe aan een guest, met gebruik van een apparaat definitie in een XML bestand.                       |
| <b>attach-disk</b>      | Ken een nieuw schijf station toe aan een guest.                                                                      |
| <b>attach-interface</b> | Ken een nieuw netwerk interface toe aan een guest.                                                                   |
| <b>detach-device</b>    | Maak een apparaat los van een guest, gebruikt hetzelfde soort XML beschrijvingen als commando <b>attach-device</b> . |
| <b>detach-disk</b>      | Maak een schijf station los van een guest.                                                                           |
| <b>detach-interface</b> | Maak een netwerk interface los van een guest.                                                                        |

Tabel 15.2. Hulpbron beheer opties

Dit zijn gevarieerde **virsh** opties:

| Commando        | Description                           |
|-----------------|---------------------------------------|
| <b>version</b>  | Laat de versie van <b>virsh</b> zien. |
| <b>nodeinfo</b> | Output informatie over de hypervisor. |

Tabel 15.3. Gevarieerde opties

### Verbinden met de hypervisor

Verbindt naar een hypervisor sessie met **virsh**:

```
virsh connect {hostnaam OF URL}
```

Waarin **<hostnaam>** de machine naam van de hypervisor is. Om een alleen-lezen verbinding op te zetten, voeg je **-readonly** toe aan het commando hierboven.

### Aanmaken van een virtuele machine XML dump (configuratie bestand)

Output het XML configuratie bestand van een guest met **virsh**:



---

```
virsh dumpxml {domein-id, domein-naam of domein-uuid}
```

Dit commando stuurt het XML configuratie bestand van de guest naar standaard uit (**stdout**). Je kunt de data opslaan door de output om te leiden naar een bestand. Een voorbeeld van de output omleiden naar een bestand met de naam *guest.xml*:

```
virsh dumpxml GuestID > guest.xml
```

Dit bestand **guest.xml** kan de guest opnieuw opbouwen, refereer naar [Bewerk het configuratie bestand van een guest](#). Je kunt dit XML configuratie bestand bewerken om extra apparaten in te stellen of extra guests in te zetten. Refereer naar [Paragraaf 18.1, "XML configuratie bestanden gebruiken met virsh"](#) voor meer informatie over het veranderen van bestanden aangemaakt met **virsh dumpxml**.

Een voorbeeld van **virsh dumpxml** output:

```
virsh dumpxml r5b2-mysql01
<domain type='xen' id='13'>
 <name>r5b2-mysql01</name>
 <uuid>4a4c59a7ee3fc78196e4288f2862f011</uuid>
 <bootloader>/usr/bin/pygrub</bootloader>
 <os>
 <type>linux</type>
 <kernel>/var/lib/libvirt/vmlinuz.2dgnU_</kernel>
 <initrd>/var/lib/libvirt/initrd.UQafMw</initrd>
 <cmdline>ro root=/dev/VolGroup00/LogVol00 rhgb quiet</cmdline>
 </os>
 <memory>512000</memory>
 <vcpu>1</vcpu>
 <on_poweroff>destroy</on_poweroff>
 <on_reboot>restart</on_reboot>
 <on_crash>restart</on_crash>
 <devices>
 <interface type='bridge'>
 <source bridge='xenbr0' />
 <mac address='00:16:3e:49:1d:11' />
 <script path='vif-bridge' />
 </interface>
 <graphics type='vnc' port='5900' />
 <console tty='/dev/pts/4' />
 </devices>
</domain>
```

### Een guest aanmaken van een configuratie bestand

Guest kunnen aangemaakt worden van XML configuratie bestanden. Je kunt bestaande XML kopiëren van eerder aangemaakt guests of gebruik de **dumpxml** optie (refereer naar [Aanmaken van een virtuele machine XML dump \(configuratie bestand\)](#)). Om een guest aan te maken van een XML bestand met **virsh**

```
virsh create configuratie_bestand.xml
```

### Bewerk het configuratie bestand van een guest

In plaats van het gebruik van de **dumpxml** optie (refereer naar [Aanmaken van een virtuele machine XML dump \(configuratie bestand\)](#)) kunnen guest bewerkt worden of terwijl ze draaien, of als ze off-line zijn. Het **virsh edit** commando biedt deze functionaliteit. Bijvoorbeeld, om de guest met de naam *softwaretesting* te bewerken:

```
virsh edit softwaretesting
```

Dit opent een tekstverwerker. De standaard tekstverwerker is de **\$EDITOR** shell parameter (standaard ingesteld op **vi**).

### Een guest pauzeren

Pauzeer een guest met **virsh**:

```
virsh suspend {domein-id, domein-naam of domein-uuid}
```

Als een guest in pauze staat, verbruikt het systeem RAM maar geen processor capaciteit. Schijf en netwerk I/O treedt niet op als de guest in pauze staat. Deze operatie treedt onmiddellijk in en de guest kan weer in werking gebracht worden met de **resume** ([Een guest weer in werking stellen](#)) optie.

### Een guest weer in werking stellen

Stel in guest in pauze weer in werking met **virsh** met gebruik van de **resume** optie:

```
virsh resume {domein-id, domein-naam of domein-uuid}
```

Deze operatie heeft onmiddellijk effect en de guest parameters worden bewaard voor **suspend** en **resume** operaties.

### Een guest opslaan

Sla de huidige toestand van een guest op in een bestand met gebruik van het **virsh** commando:

```
virsh save {domein-naam, domein-id of domein-uuid} bestandsnaam
```

Dit stopt de guest die je specificeert en slaat de data op in een bestand, wat enige tijd kan duren afhankelijk van de hoeveelheid geheugen in gebruik bij jouw guest. Je kunt de toestand van de guest weer inladen met de **restore** ([Een guest herladen](#)) optie. Opslaan is vergelijkbaar met pauzeren, in plaats van alleen maar pauzeren wordt de huidige toestand ook opgeslagen.

### Een guest herladen

Een eerder opgeslagen guest kan herladen worden met het **virsh save** commando ([Een guest opslaan](#)) met gebruik van **virsh**:

```
virsh restore bestandsnaam
```

---

Dit herstart de opgeslagen guest, wat even kan duren. De naam en UUID van de guest blijven bewaard maar toegekend een een nieuwe ID.

### Een guest afsluiten

Een guest afsluiten met gebruik van het **virsh** commando:

```
virsh shutdown {domein-id, domein-naam of domein-uuid}
```

Je kunt het gedrag van de opnieuw opstartende guest controleren met het veranderen van de **on\_shutdown** parameter in het configuratie bestand van de guest.

### Een guest opnieuw opstarten

Start een guest opnieuw op met gebruik van het **virsh** commando:

```
#virsh reboot {domein-id, domein-naam of domein-uuid}
```

Je kunt het gedrag van de opnieuw opstartende guest controleren door het veranderen van de **on\_reboot** parameter in het configuratie bestand van de guest.

### Een guest forceren te stoppen

Een guest forceren te stoppen met het **virsh** commando:

```
virsh destroy {domein-id, domein-naam or domein-uuid}
```

Dit commando sluit onmiddellijk en onverbiddeijk af en stopt de opgegeven guest. Het gebruik van **virsh destroy** kan guest bestandssystemen corrupt maken. Gebruik de **destroy** optie alleen als de guest niet meer reageert. Voor para-gevirtualiseerde guests gebruik je in plaats hiervan de **shutdown** optie ([Een guest afsluiten](#)).

### De domein ID van een guest verkrijgen

Om de domein ID van een guest te verkrijgen:

```
virsh domid {domein-naam of domein-uuid}
```

### De domein naam van een guest verkrijgen

Om de domein naam van een guest te verkrijgen:

```
virsh domname {domein-id of domein-uuid}
```

### De UUID van een guest verkrijgen

Om de Universally Unique Identifier (UUID) voor een guest te verkrijgen:

```
virsh domuuid {domein-id or domein-naam}
```

Een voorbeeld van **virsh domuuid** output:

```
virsh domuuid r5b2-mysql01
4a4c59a7-ee3f-c781-96e4-288f2862f011
```

### Guest informatie laten zien

Met gebruik van **virsh** en de domein ID, domein naam of UUID van de guest kun je informatie laten zien over de gespecificeerde guest:

```
virsh dominfo {domein-id, domein-naam of domein-uuid}
```

Dit is een voorbeeld van **virsh dominfo** output:

```
virsh dominfo r5b2-mysql01
id: 13
name: r5b2-mysql01
uuid: 4a4c59a7-ee3f-c781-96e4-288f2862f011
os type: linux
state: blocked
cpu(s): 1
cpu time: 11.0s
max memory: 512000 kb
used memory: 512000 kb
```

### Host informatie laten zien

Om informatie over de host te laten zien:

```
virsh nodeinfo
```

Een voorbeeld van **virsh nodeinfo** output:

```
virsh nodeinfo
CPU model x86_64
CPU (s) 8
CPU frequency 2895 Mhz
CPU socket(s) 2
Core(s) per socket 2
Threads per core: 2
Numa cell(s) 1
Memory size: 1046528 kb
```

Dit laat de node informatie zien en de machines die het virtualisatie proces ondersteunen.

### De guests laten zien

Om een lijst van de guests en hun huidige toestand te laten zien met **virsh**:

```
virsh list
```

---

Andere beschikbare opties zijn:

de **--inactive** optie om een lijst van de niet-actieve guests (dat betekent, guests die gedefinieerd zijn maar op dit moment niet actief) te tonen, en

de **--all** optie om alle guests te tonen. Bijvoorbeeld:

```
virsh list --all
 Id Name State

 0 Domain-0 running
 1 Domain202 paused
 2 Domain010 inactive
 3 Domain9600 crashed
```

De output van **virsh list** wordt in een van zes toestanden ingedeeld (hieronder getoond).

- De **running** toestand refereert naar guests die op dit moment actief zijn op een CPU.
- Guests getoond met **blocked** zijn geblokkeerd, en draaien niet of zijn niet draaibaar. Dit wordt veroorzaakt door een guest die wacht op I/O (een traditionele wacht toestand) of guests in een slaap mode.
- De **paused** toestand laat domeinen zien die in pauze zijn. Dit gebeurt als de beheerder de **pause** knop in **virt-manager**, **xm pause** of **virsh suspend** gebruikt. Als een guest in pauze is verbruikt het geheugen en andere hulpbronnen maar het is ongeschikt voor inplannen en CPU en hulpbronnen van de hypervisor.
- De **shutdown** toestand is voor guests die bezig zijn met afsluiten. De guest wordt een afsluit signaal gestuurd en moet bezig zijn om zijn operaties zorgvuldig te stoppen. Dit werkt misschien niet met alle besturingssystemen; sommige besturingssystemen reageren niet op deze signalen.
- Domeins in de **dying** toestand zijn in het proces van afsterven, wat een toestand is waarin het domein nog niet helemaal uitgezet is of gecrashed is.
- **crashed** guest hebben gefaald toen ze draaiden en draaien nu niet meer. Deze toestand kan alleen optreden als de guest is ingesteld om niet opnieuw op te starten na een crash.

### Virtuele CPU informatie tonen

Om virtuele CPU informatie te tonen met **virsh**:

```
virsh vcpuinfo {domein-id, domein-naam of domein-uuid}
```

Een voorbeeld van **virsh vcpuinfo** output:

```
virsh vcpuinfo r5b2-mysql01
VCPU: 0
CPU: 0
State: blocked
CPU time: 0.0s
CPU Affinity: yy
```

### Virtuele CPU verwantschap instellen

Om de verwantschap van virtuele CPU's met fysieke CPU's in te stellen:

```
virsh vcpupin {domein-id, domein-naam of domein-uuid} vcpu, cpulist
```

Waarin **vcpu** het virtuele VCPU nummer is en **cpulist** een lijst van fysieke CPU's.

### Virtuele CPU aantal instellen

Om het aantal CPU's toegekend aan een guest in te stellen met **virsh**:

```
virsh setvcpu {domein-naam, domein-id of domein-uuid} count
```

De nieuwe *count* waarde kan niet groter zijn dan de hoeveelheid die gespecificeerd werd voor *count* toen de guest werd aangemaakt.

### Geheugen toekenning instellen

Om de geheugen toekenning van een guest te veranderen met **virsh** :

```
virsh setmem {domein-id of domein-naam} count
```

Je moet de waarde van *count* in kilobytes opgeven. De nieuwe *count* waarde kan niet groter zijn dan de waarde van *count* die je opgaf bij het aanmaken van de guest. Waardes lager dan 64 MB zullen waarschijnlijk niet werken met moderne besturingssystemen. Een hogere maximum geheugen waarde zal een actieve guest niet beïnvloeden behalve als de nieuwe waarde lager is, dit zal het beschikbare geheugen gebruik laten krimpen.

### Guest blok apparaat informatie tonen

Gebruik **virsh domblkstat** om blok apparaat statistieken te tonen voor een draaiende guest.

```
virsh domblkstat GuestNaam blok-apparaat
```

### Guest netwerk apparaat informatie tonen

Gebruik **virsh domifstat** om netwerk interface statistieken te tonen voor een draaiende guest.

```
virsh domifstat GuestNaam interface-apparaat
```

### Guests migreren met virsh

Een guest kan gemigreerd worden naar een andere host met **virsh**. Migreer domein naar een andere host. Voeg `--live` toe voor live migratie. Het **migrate** commando accepteert parameters in het volgende formaat:

```
virsh migrate --live GuestName DestinationURL
```

De `--live` parameter is optioneel. Voeg de `--live` parameter toe voor live migraties.

---

The *GuestName* parameter represents the name of the guest which you want to migrate.

The *DestinationURL* parameter is the URL or hostname of the destination system. The destination system must run the same version of Fedora, be using the same hypervisor and have **libvirt** running.

Once the command is entered you will be prompted for the root password of the destination system.

### Virtuele netwerken beheren

Deze paragraaf behandelt het beheren van virtuele netwerken met het **virsh** commando. Om een lijst te laten zien van virtuele netwerken:

```
virsh net-list
```

Dit commando geeft output die lijkt op:

```
virsh net-list
Name State Autostart

default active yes
vnet1 active yes
vnet2 active yes
```

Om netwerk informatie van een specifiek virtueel netwerk te bekijken:

```
virsh net-dumpxml NetwerkNaam
```

Dit laat informatie zien van een specifiek virtueel netwerk in XML formaat:

```
virsh net-dumpxml vnet1
<network>
 <name>vnet1</name>
 <uuid>98361b46-1581-acb7-1643-85a412626e70</uuid>
 <forward dev='eth0' />
 <bridge name='vnet0' stp='on' forwardDelay='0' />
 <ip address='192.168.100.1' netmask='255.255.255.0'>
 <dhcp>
 <range start='192.168.100.128' end='192.168.100.254' />
 </dhcp>
 </ip>
</network>
```

Andere **virsh** commando's die gebruikt worden in het beheer van virtuele netwerken zijn:

- **virsh net-autostart *netwerk-naam*** — Start een netwerk gespecificeerd als *netwerk-naam* automatisch op.
- **virsh net-create *XMLbestand*** — genereert en start een nieuw netwerk op met gebruik van een bestaand XML bestand.

- **virsh net-define *XMLbestand*** — genereert een nieuw netwerk apparaat van een bestaand XML bestand zonder het op te starten.
- **virsh net-destroy *netwerk-naam*** — vernietig een netwerk opgegeven als *netwerk-naam*.
- **virsh net-name *netwerkUUID*** — converteer een gespecificeerde *netwerkUUID* naar een netwerk naam.
- **virsh net-uuid *netwerk-naam*** — converteer een gespecificeerde *netwerk-naam* naar een netwerk UUID.
- **virsh net-start *naamVanInactiefNetwerk*** — start een inactief netwerk.
- **virsh net-undefine *naamVanInactiefNetwerk*** — verwijdert de definitie van een inactief netwerk.



---

# Guests beheren met de Virtual Machine Manager (virt-manager)

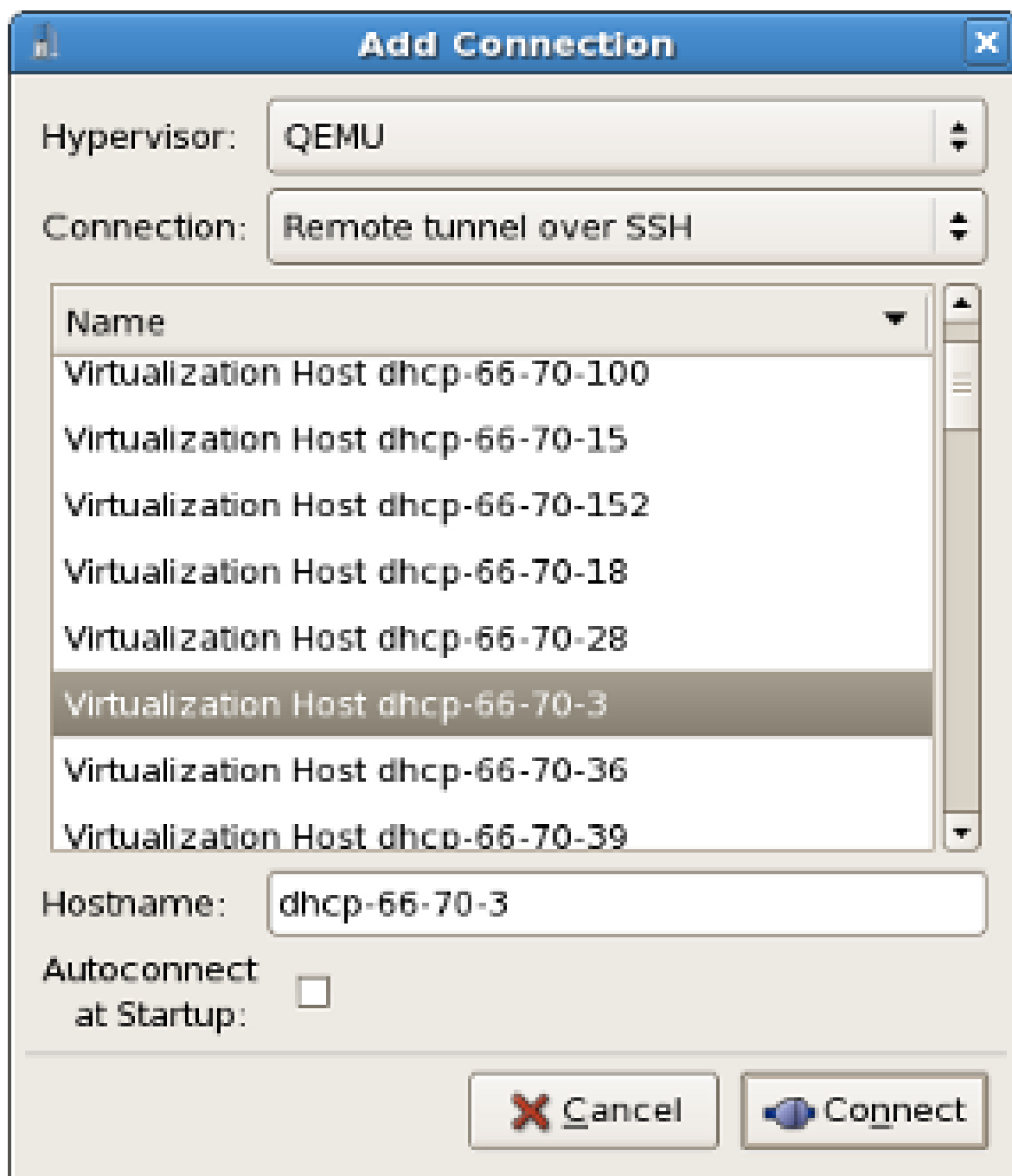
Deze paragraaf beschrijft de Virtual Machine Manager (**virt-manager**) vensters, dialogen, en verschillende GUI controlepanelen.

**virt-manager** biedt een grafisch overzicht van hypervisors en guests op jouw systeem en op machines op afstand. Je kunt **virt-manager** gebruiken voor het definiëren van zowel para-gevirtualiseerde als volledig gevirtualiseerde guests. **virt-manager** kan virtualisatie beheer taken uitvoeren, zoals:

- geheugen toekennen,
- virtuele CPU's toekennen,
- operationele prestaties volgen,
- gevirtualiseerde guests opslaan en laden, pauzeren en vervolgen, en afsluiten en opstarten,
- linken naar tekst en grafische consoles, en
- live en off-line migraties.

## 16.1. Het open connection venster

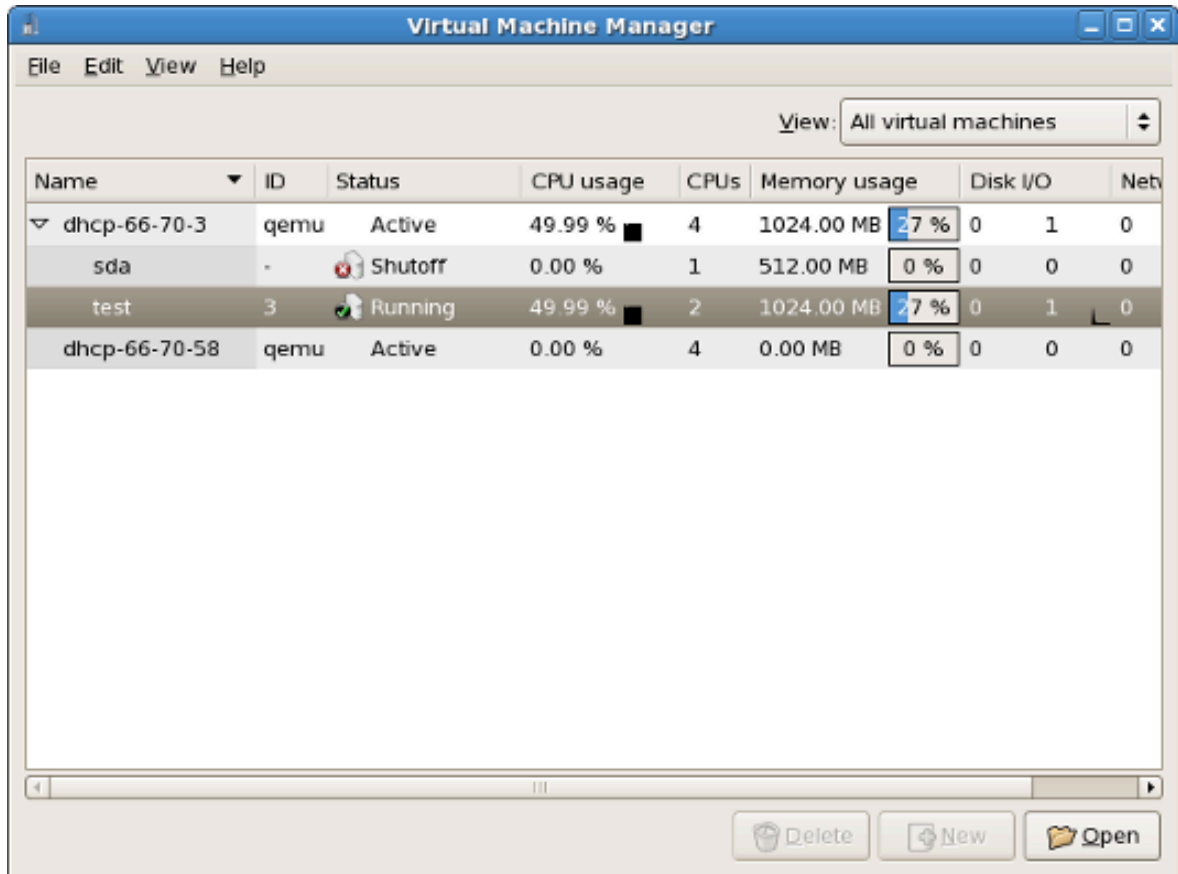
Dit venster verschijnt eerst en vraagt de gebruiker om een hypervisor sessie te kiezen. Gebruikers zonder rechten kunnen een alleen-lezen sessie opstarten. Root gebruikers kunnen een sessie opstarten met volledige lees-schrijf rechten. Voor normaal gebruik selecteer je de **Local Xen host** optie of QEMU (voor KVM).



Figuur 16.1. Virtual Machine Manager verbinding venster

## 16.2. Het Virtual Machine Manager hoofd scherm

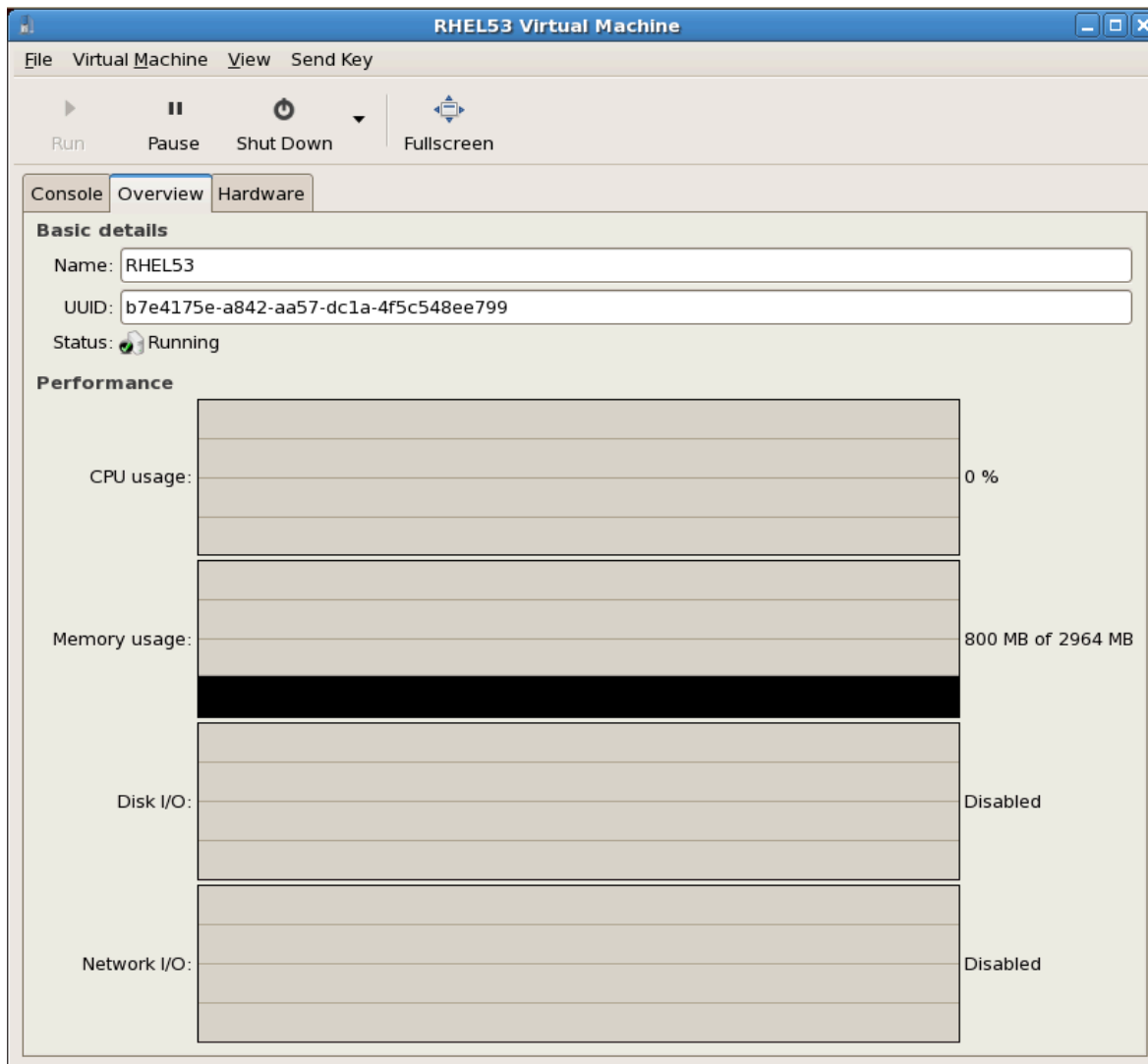
Dit hoofd scherm laat alle draaiende virtuele machines zien en de hulpbronnen die er op dit moment aan toegekend zijn (inclusief domein0). Je kunt beslissen welke velden je wilt tonen. Dubbel-klikken op de gewenste virtuele machine laat de console voor die machine verschijnen. Het selecteren van een virtuele machine en dubbel-klikken op de **Details** knop laat het Details venster voor die machine verschijnen. Je kunt ook toegang krijgen tot het **File** menu om een nieuwe virtuele machine aan te maken.



Figuur 16.2. Virtual Machine Manager hoofd venster

### 16.3. Het Virtual Machine Manager details venster

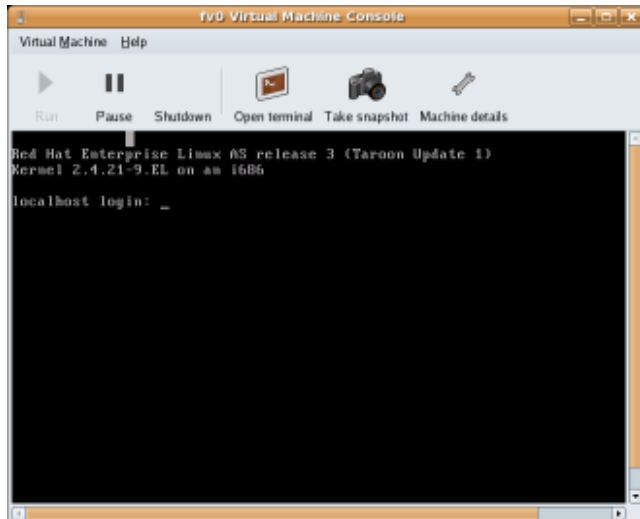
Dit venster laat de grafieken en statistieken van huidige hulpbronnen gebruik data van een guest zien, welke beschikbaar is van **virt-manager**. Het UUID veld laat de globale unieke identifier zien voor de virtuele machines.



Figuur 16.3. virt-manager details venster

### 16.4. Virtual Machine grafische console

Dit venster laat de grafische console van een virtuele machine zien. Para-gevirtualiseerde en volledig gevirtualiseerde guests gebruiken verschillende technieken om hun lokale virtuele framebuffers te exporteren, maar beide technologieën gebruiken **VNC** om ze beschikbaar te maken voor het console venster van de Virtual Machine Manager. Als jouw virtuele machine is ingesteld om authenticatie te vereisen, vraagt de Virtual Machine Manager je om een wachtwoord voordat het scherm verschijnt.



Figuur 16.4. Grafische console venster



### Een opmerking over beveiliging en VNC

VNC wordt als onveilig beschouwd door vele beveiligings deskundigen, er zijn echter vele veranderingen gemaakt om het veilig gebruik van VNC voor virtualisatie in Fedora mogelijk te maken. De guest machines luisteren alleen naar het loopback adres (127.0.0.1) van de locale host (dom0). Dit verzekert dat alleen diegenen met shell rechten op de host met VNC toegang kunnen krijgen naar virt-manager en de virtuele machine.

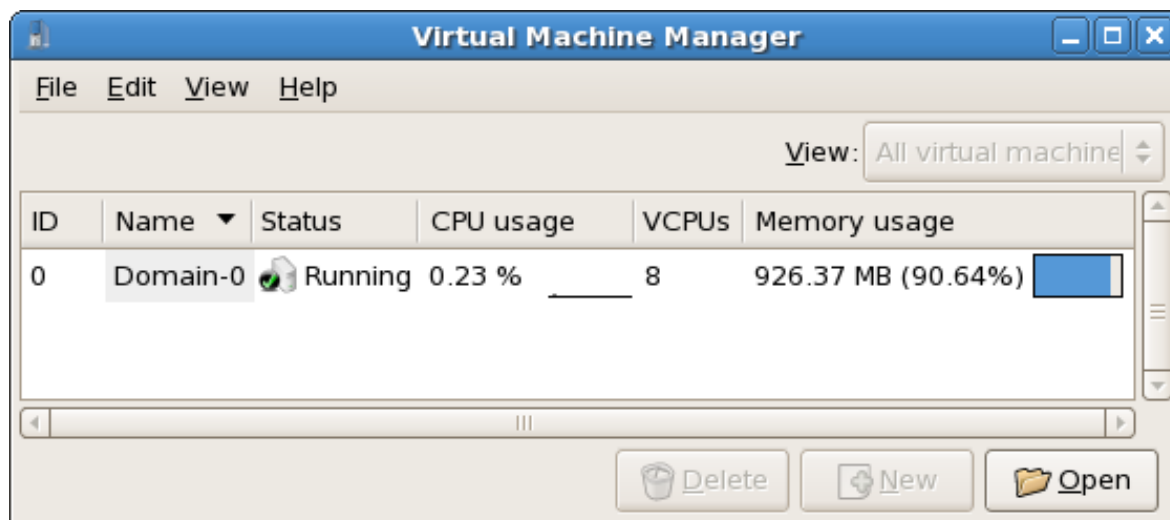
Beheer op afstand kan uitgevoerd worden door de instructies in *Hoofdstuk 13, Beheer op afstand van gevirtualiseerde guests* op te volgen. TLS kan productie niveau beveiliging bieden voor het beheer van guest en host systemen.

Jouw locale desktop van toetscombinaties (bijvoorbeeld, Ctrl+Alt+F11) onderscheppen om te voorkomen dat ze naar de guest machine gestuurd worden. Je kunt de **virt-manager** sticky-toets mogelijkheid gebruiken om deze aanslagen te versturen. Je moet een verander toets (Ctrl of Alt) 3 keer induwen en de toets die specificeert wordt behandeld als actief totdat de volgende niet-verander toets ingedruwd wordt. Dus je kunt Ctrl-Alt-F11 naar de guest sturen door het intypen van de toets volgorde 'Ctrl Ctrl Ctrl Alt+F1'.

## 16.5. Starting virt-manager

Om een **virt-manager** sessie op te starten, open je het **Applications** menu, dan het **System Tools** menu, en je selecteert **Virtual Machine Manager (virt-manager)**.

Het **virt-manager** hoofd venster verschijnt.



Figuur 16.5. **virt-manager** opstarten

Als alternatief kan **virt-manager** op afstand opgestart worden met gebruik van ssh zoals het volgende commando laat zien:

```
ssh -X adres van host[remotehost]# virt-manager
```

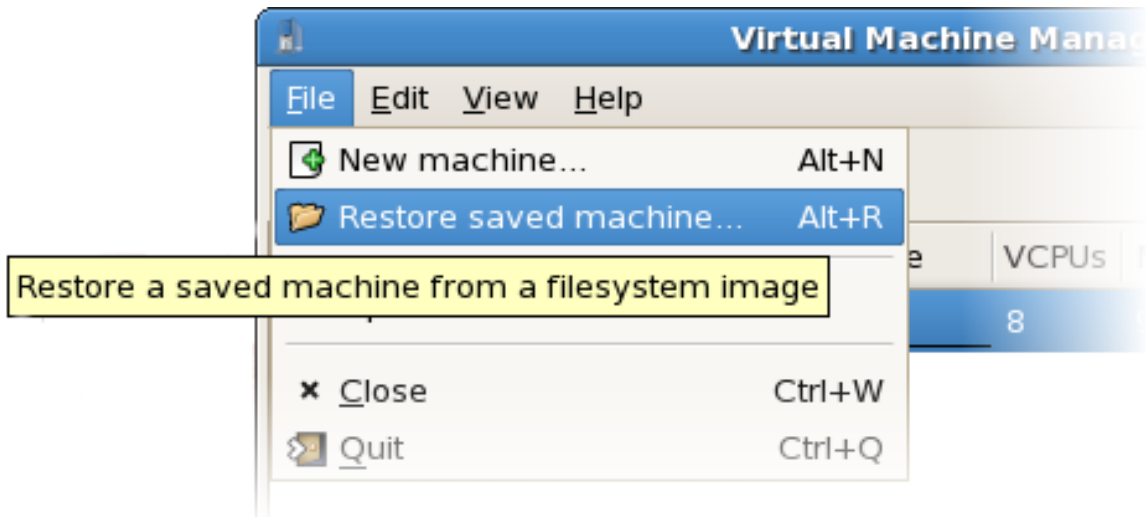
Het gebruik van **ssh** om virtuele machines en hosts te beheren wordt verder behandeld in [Paragraaf 13.1, "Beheer op afstand met SSH"](#).

## 16.6. Een opgeslagen machine herladen

Nadat je de Virtual Machine Manager opgestart hebt, worden alle virtuele machines op jouw systeem getoond in het hoofd venster. Domein0 is je host systeem. Als er geen machines aanwezig zijn, betekent dat er op dit moment geen machines op het systeem draaien.

Om een eerder opgeslagen systeem te herladen:

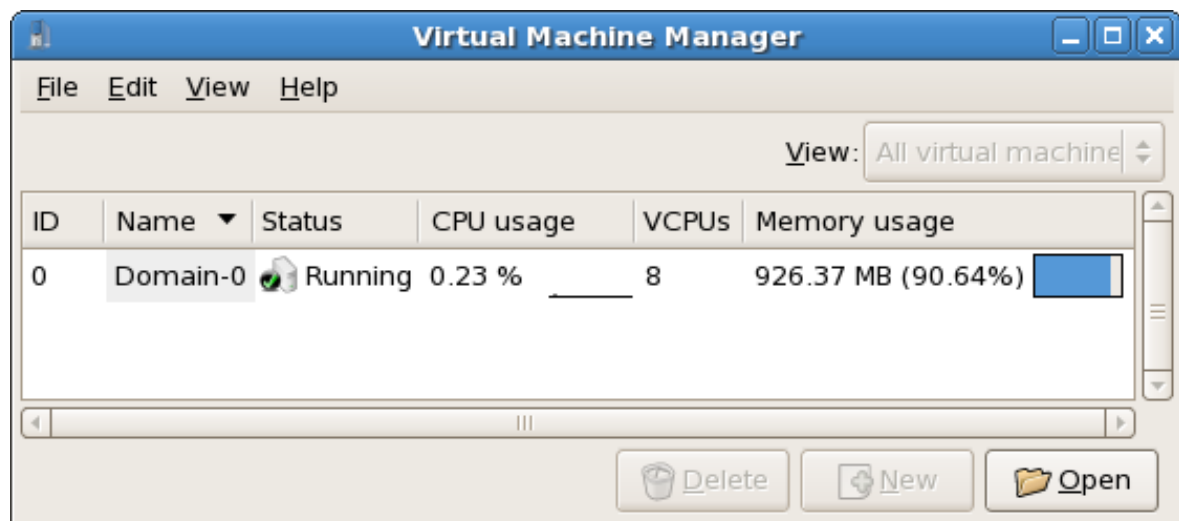
1. In het **File** menu, selecteer je **Restore a saved machine**.



Figuur 16.6. Een virtuele machine herladen

2. Het **Restore Virtual Machine** hoofd venster verschijnt,
3. Ga naar de juiste map en selecteer het opgeslagen sessie bestand.
4. Klik op **Open**.

Het opgeslagen virtuele systeem verschijnt in het Virtual Machine Manager hoofd venster.



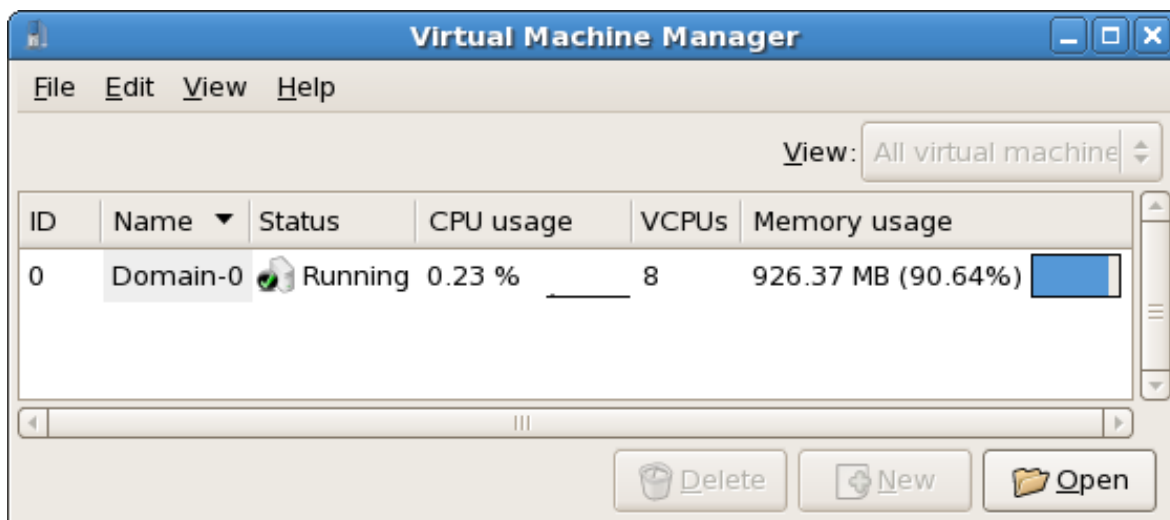
Figuur 16.7. Een herladen virtuele machine beheer sessie

## 16.7. Guest details laten zien

Je kunt de Virtual Machine Monitor gebruiken om activiteit data informatie te bekijken voor elke virtuele machine op jouw systeem.

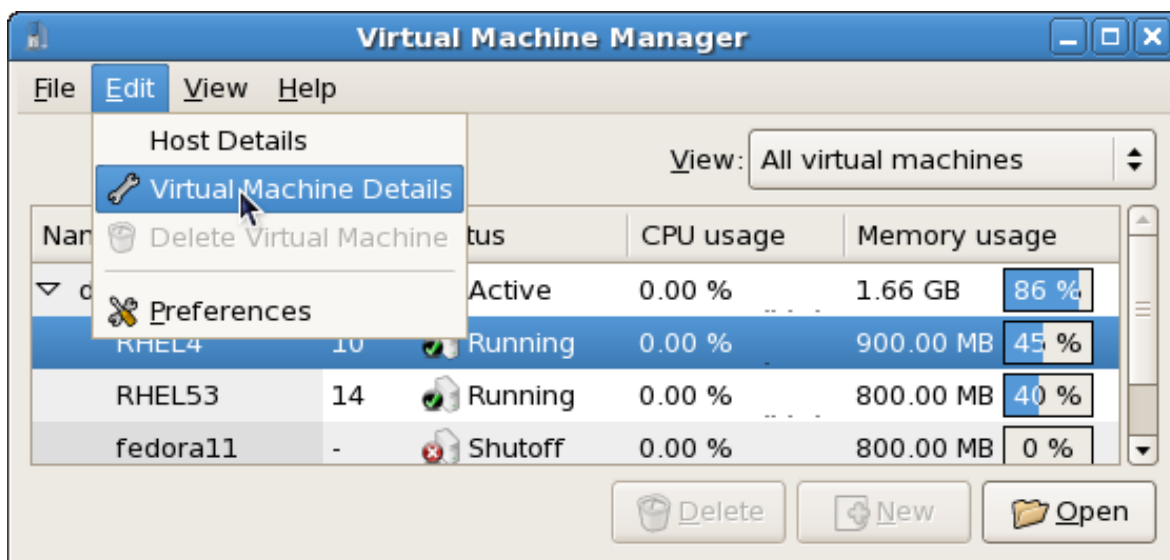
Om de details van een virtueel systeem te bekijken:

1. In het Virtual Machine Manager hoofd venster selecteer je de virtuele machine die je wilt bekijken.



Figuur 16.8. Een virtuele machine selecteren om te bekijken

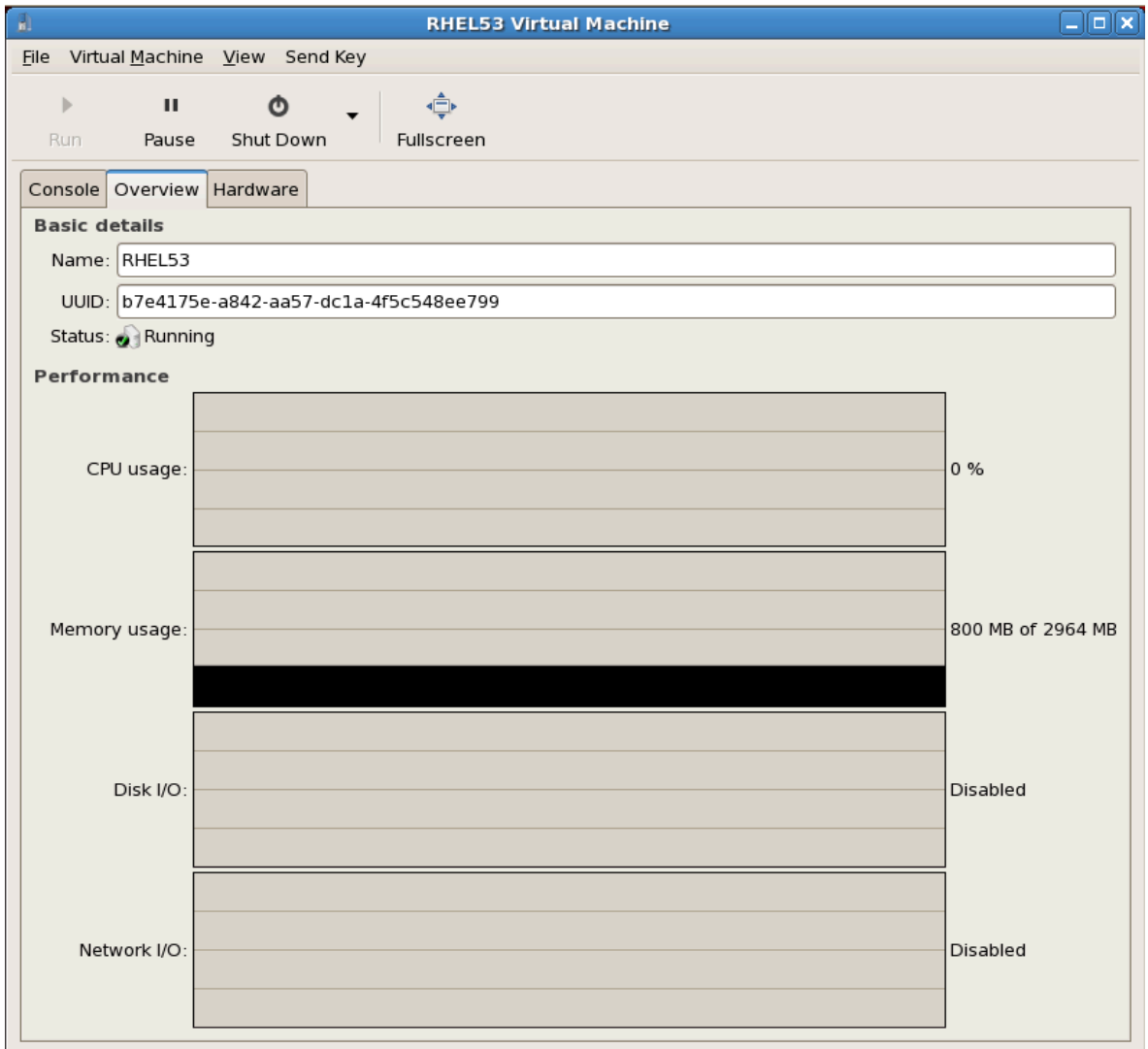
2. In het Virtual Machine Manager **Edit** menu, selecteer je **Machine Details** (of klik op de **Details** knop onder in het Virtual Machine Manager hoofd venster).



Figuur 16.9. Het virtuele machine details menu laten zien

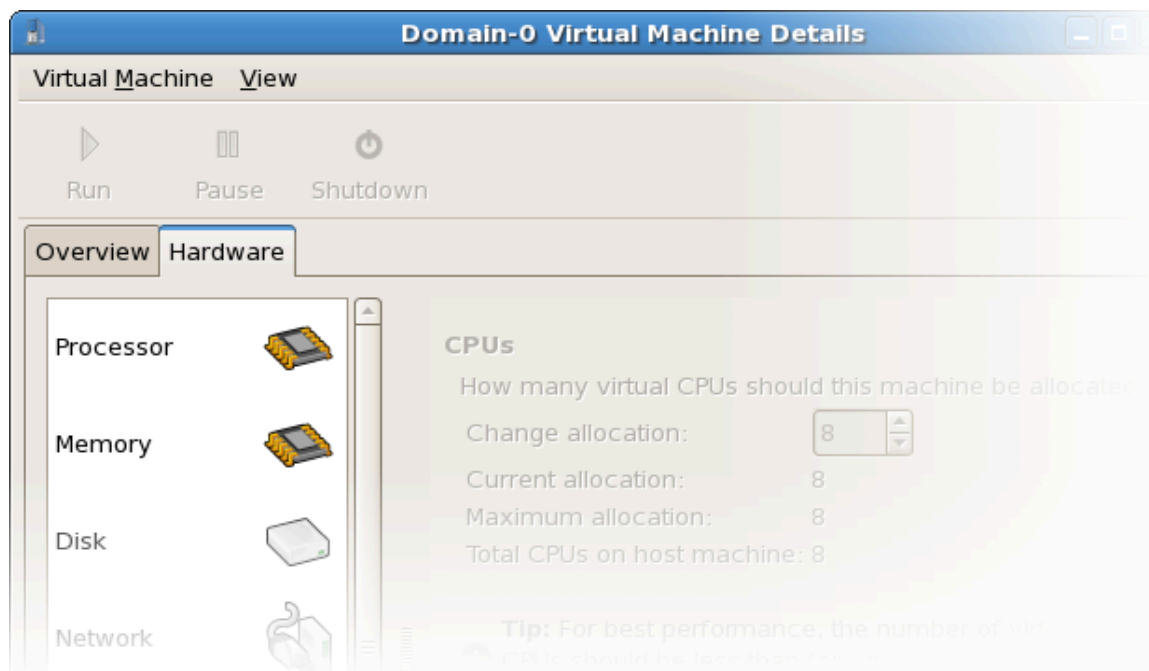
Het Virtual Machine Details Overview venster verschijnt. Dit venster geeft een samenvatting van CPU en geheugen gebruik voor de domeinen die je opgegeven hebt.





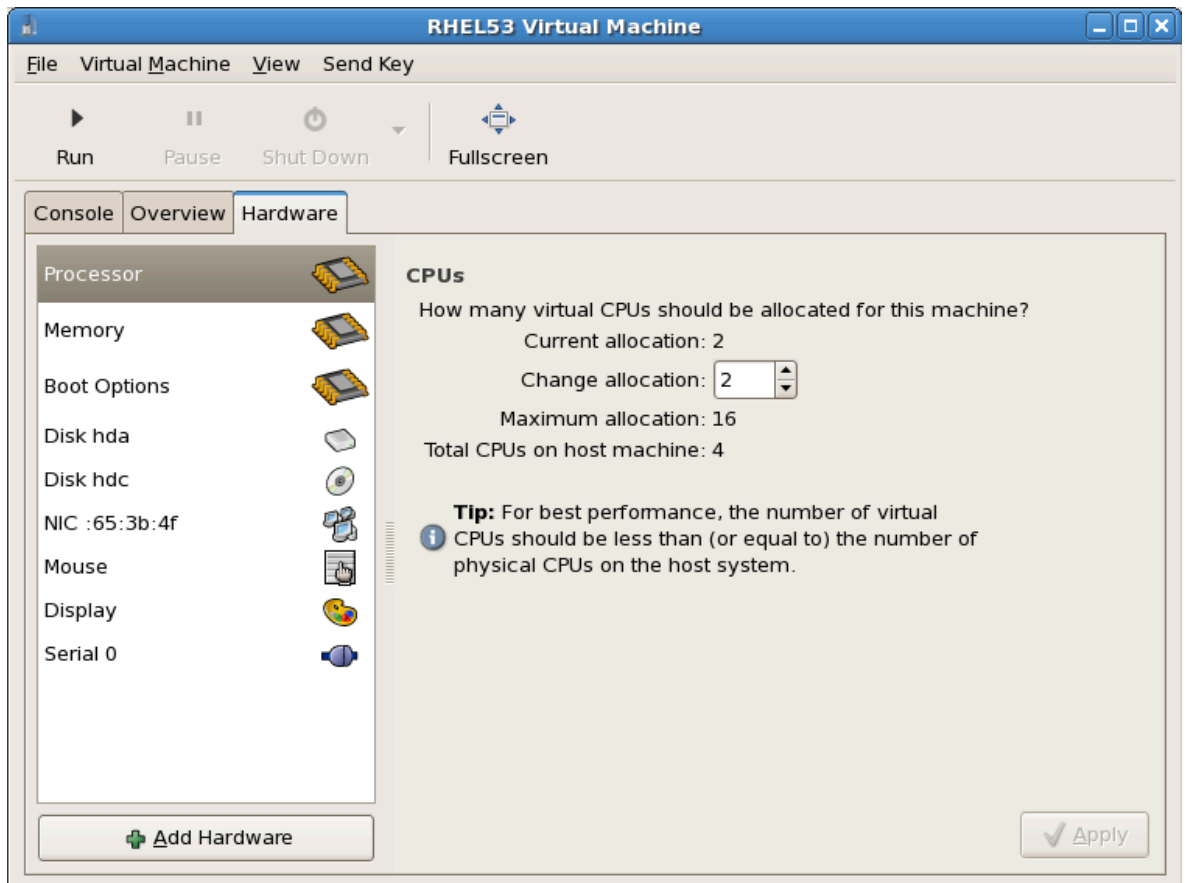
Figuur 16.10. Guest details overzicht laten zien

3. In het **Virtual Machine Details** venster klik je op de **Hardware** tab.  
Het **Virtual Machine Details Hardware** venster verschijnt.



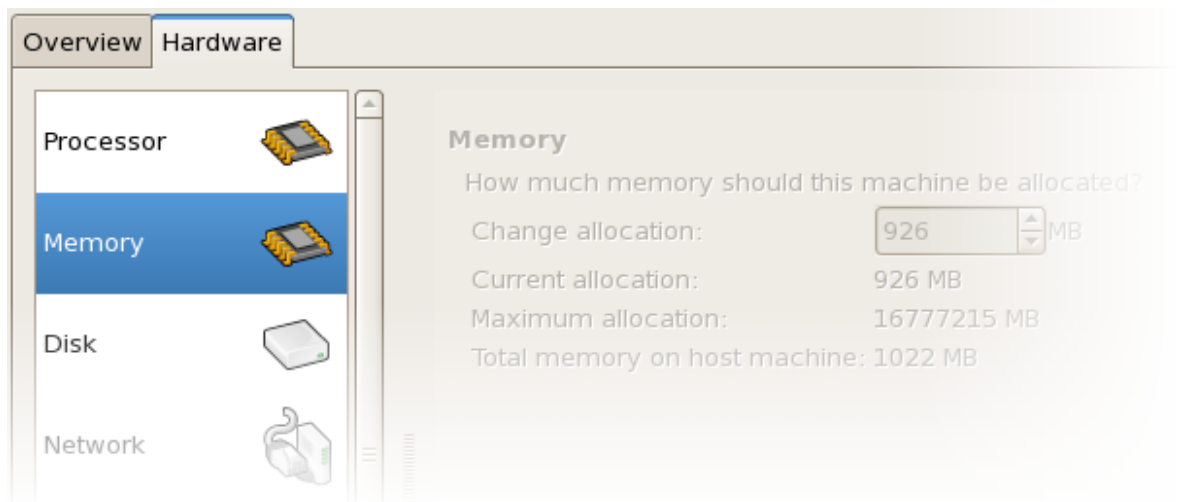
Figuur 16.11. Guest hardware details laten zien

4. In de **Hardware** tab klik je op **Processor** om de huidige processor toekenning te bekijken of te veranderen.



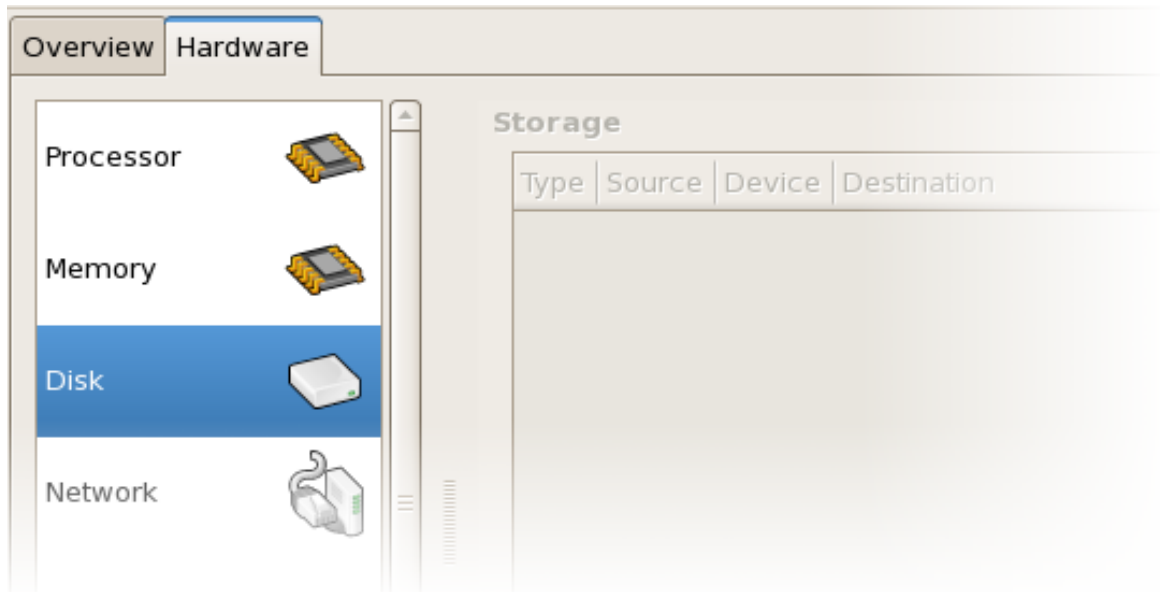
Figuur 16.12. Processor toekenning paneel

- In de **Hardware** tab klik je op **Memory** om de huidige RAM geheugen toekenning te bekijken of te veranderen.



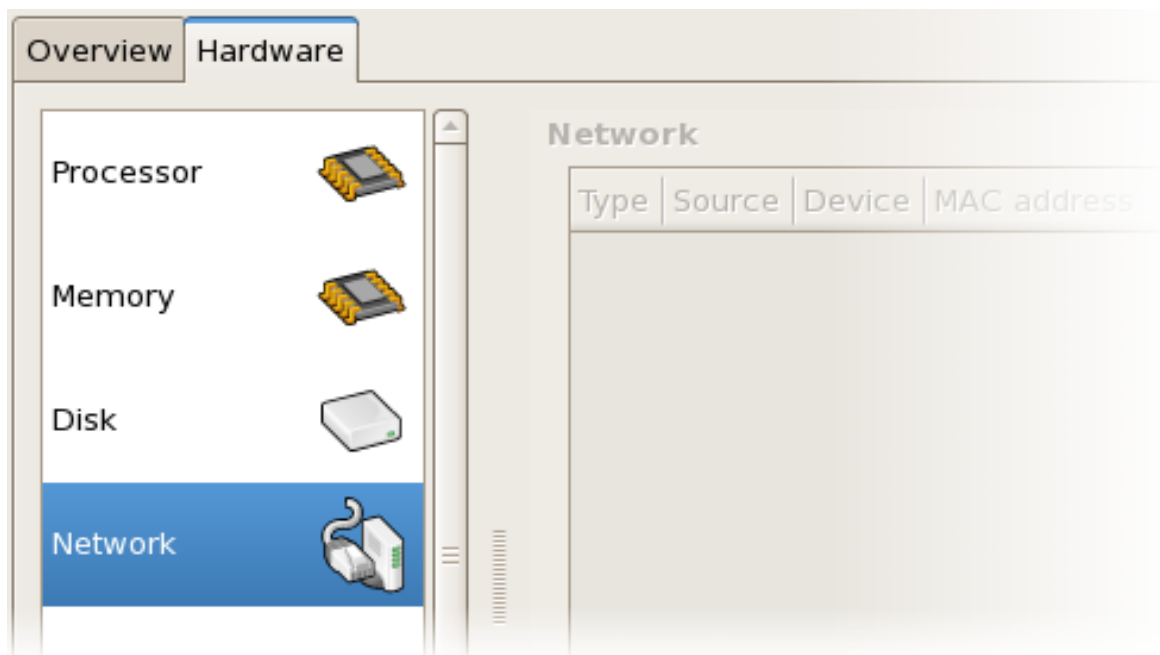
Figuur 16.13. Geheugen toekenning laten zien

- In de **Hardware** tab klik je op **Disk** om de huidige harde schijf configuratie te bekijken of te veranderen.



Figuur 16.14. Schijf configuratie laten zien

7. In de **Hardware** tab klik je op **Network** om de huidige netwerk configuratie te bekijken of te veranderen.



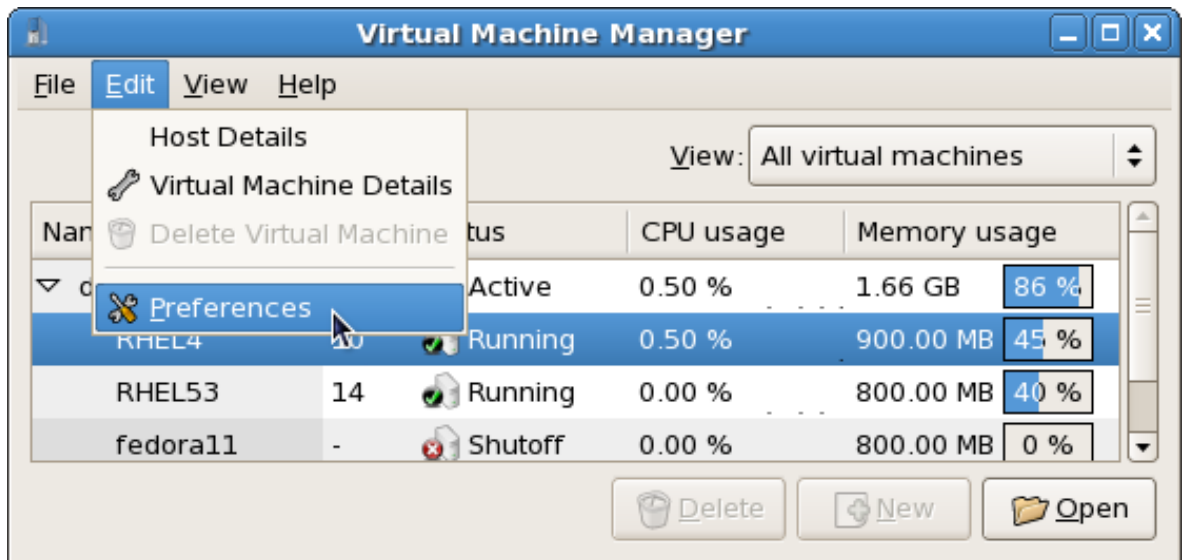
Figuur 16.15. Netwerk configuratie laten zien

## 16.8. Status volgen

Je kunt de Virtual Machine Manager gebruiken om het volgen van de virtuele systeem status te veranderen.

Om het volgen van de status en het aanzetten van consoles in te stellen:

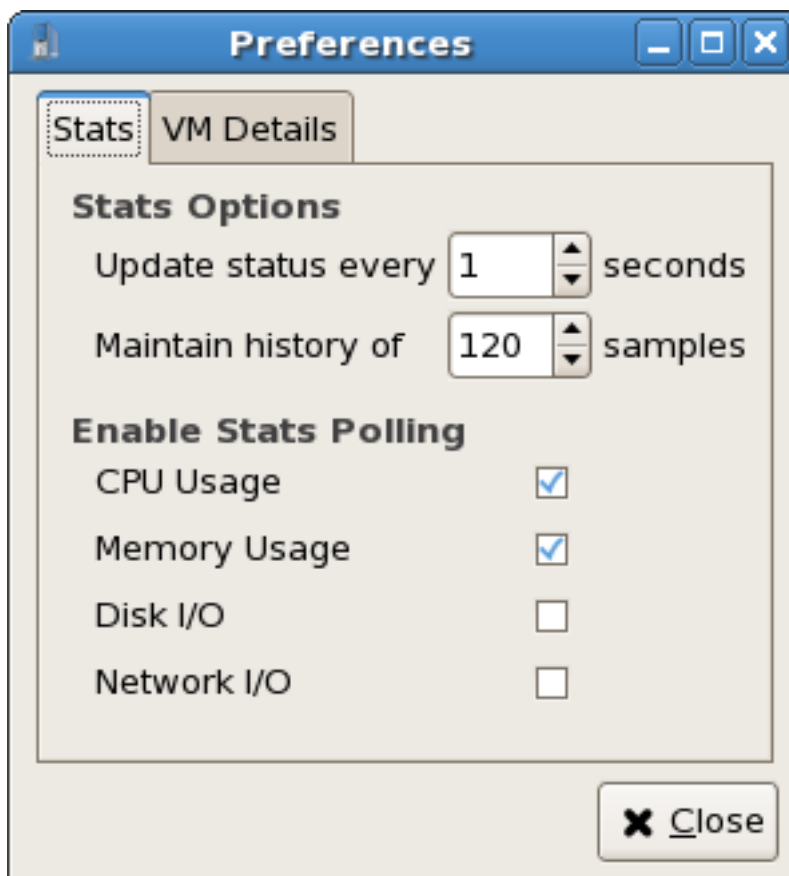
1. In het **Edit** menu selecteer je **Preferences**.



Figuur 16.16. Guest voorkeuren veranderen

Het Virtual Machine Manager Preferences venster verschijnt.

- In het Status volgen selectie gebied specificeer je de tijd (in secondes) waarna je het systeem wilt vernieuwen.



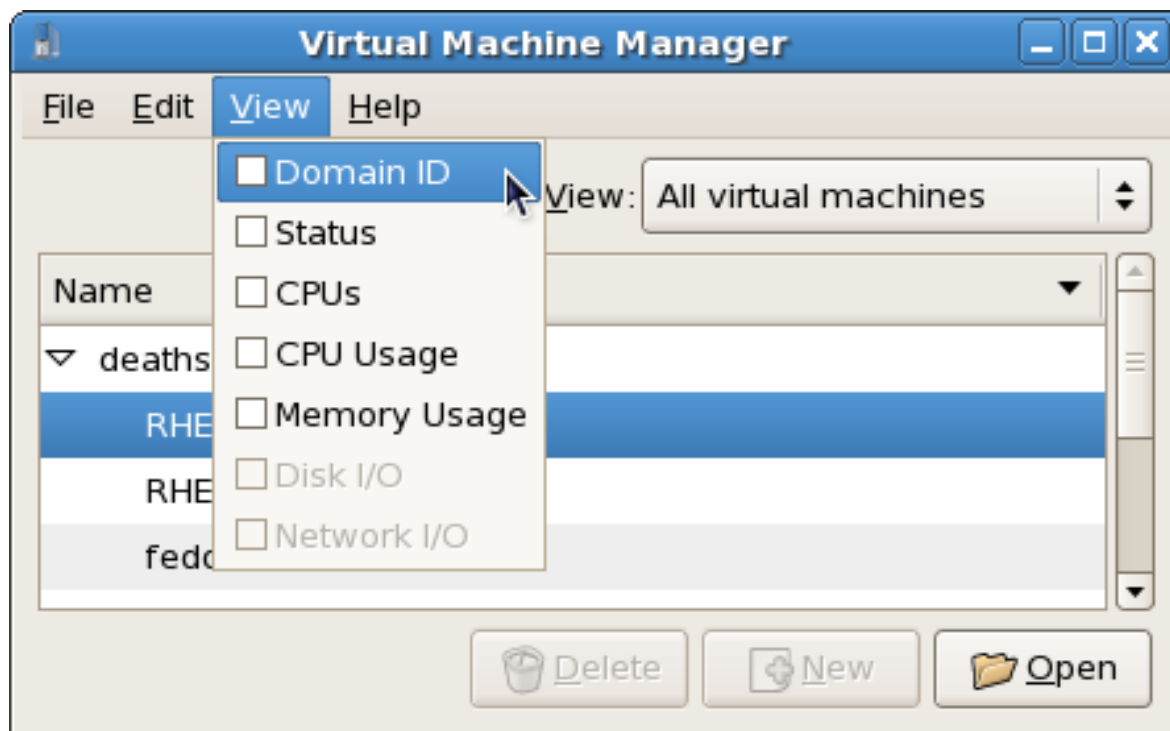
Figuur 16.17. Status volgen instellen

3. In het Consoles gebied specificeer je hoe een console geopend wordt en je specificeert een input apparaat.

### 16.9. Guest identifiers laten zien

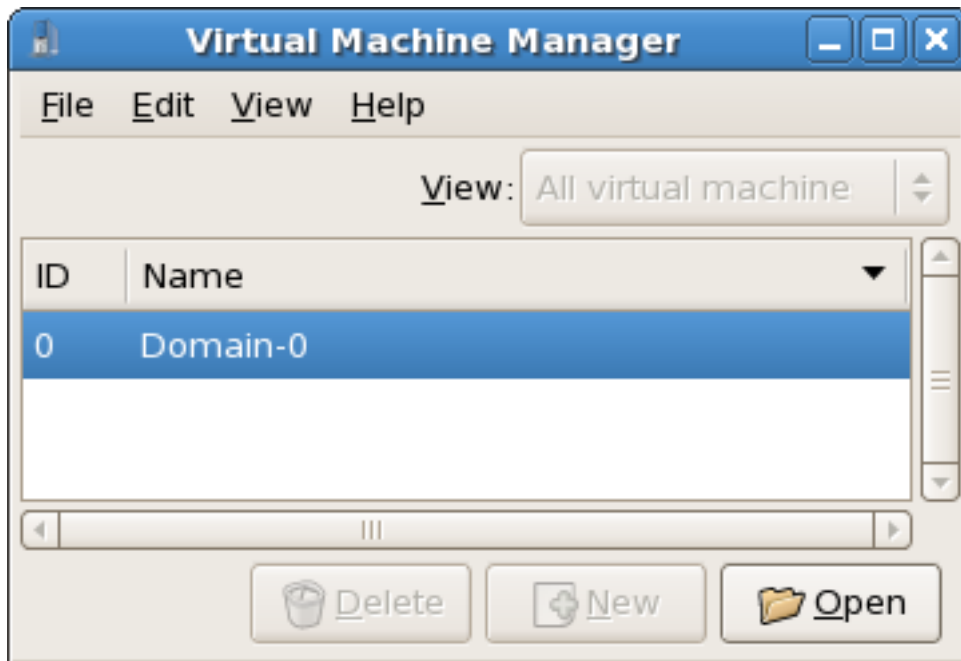
Om de guest ID's van alle virtuele machines op jouw systeem te bekijken:

1. In het **View** menu selecteer je het **Domain ID** afvink hokje.



Figuur 16.18. Guest ID's bekijken

2. De Virtual Machine Manager laat een lijst zien van de Domein ID's van alle domeinen op jouw systeem.

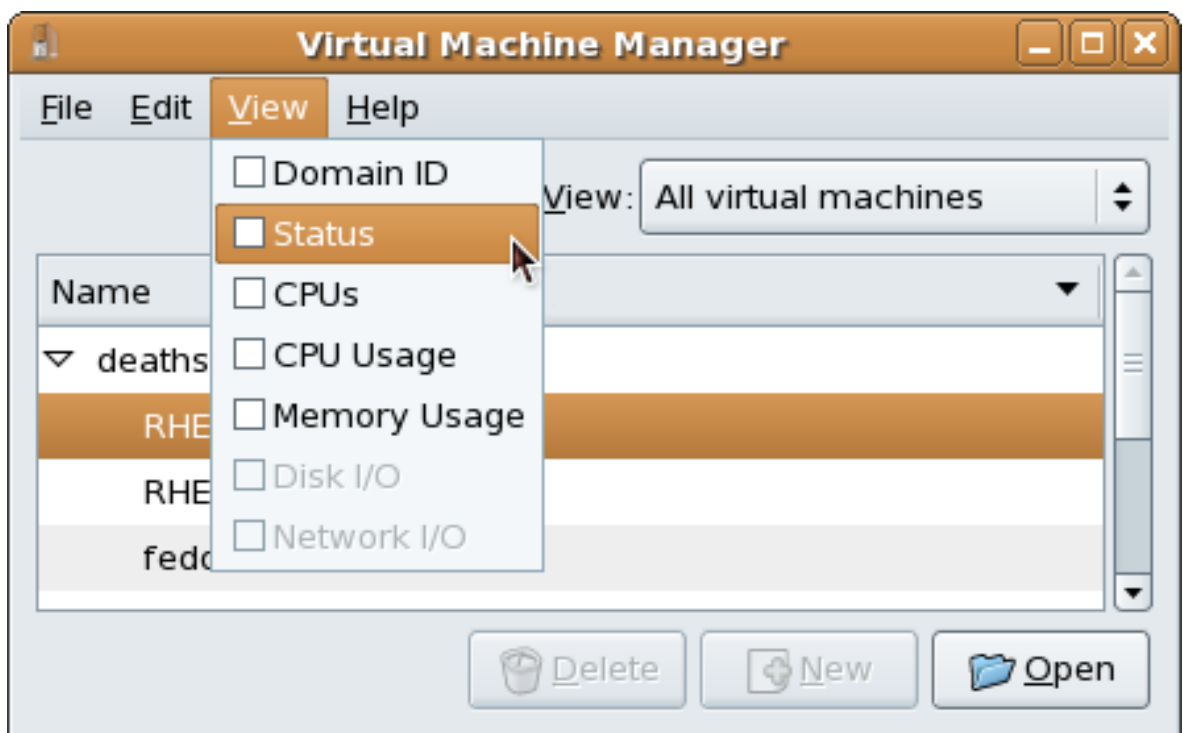


Figuur 16.19. Domein ID's laten zien

## 16.10. De status van een guest laten zien

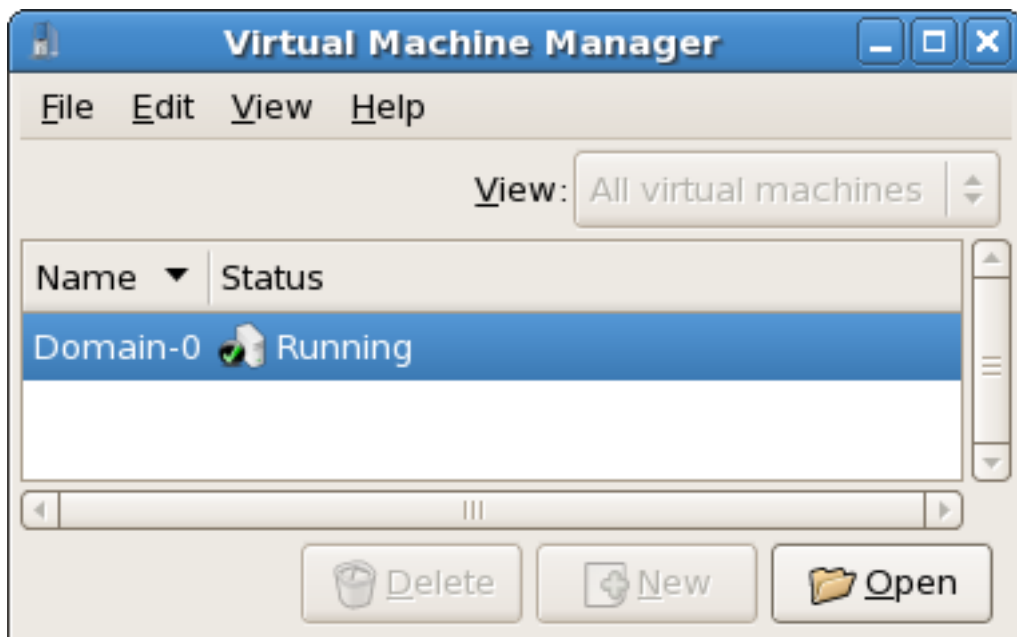
Om de status van alle virtuele machines op jouw systeem te bekijken:

1. In het **View** menu selecteer je het **Status** aanvink hokje.



Figuur 16.20. De status van een virtuele machine selecteren

2. De Virtual Machine Manager laat de status van alle virtuele machines op jouw systeem zien.

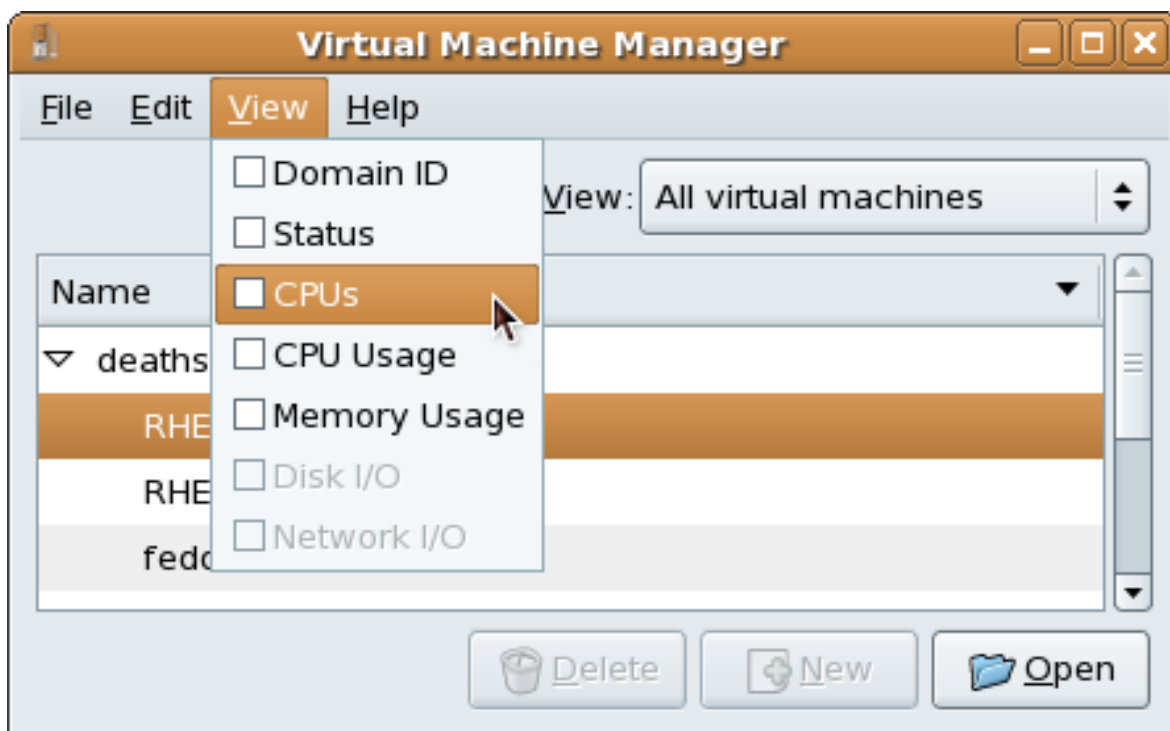


Figuur 16.21. De status van een virtuele machine laten zien

## 16.11. Virtuele CPU's laten zien

Om het aantal virtuele CPU's voor alle virtuele machines op jouw systeem te laten zien:

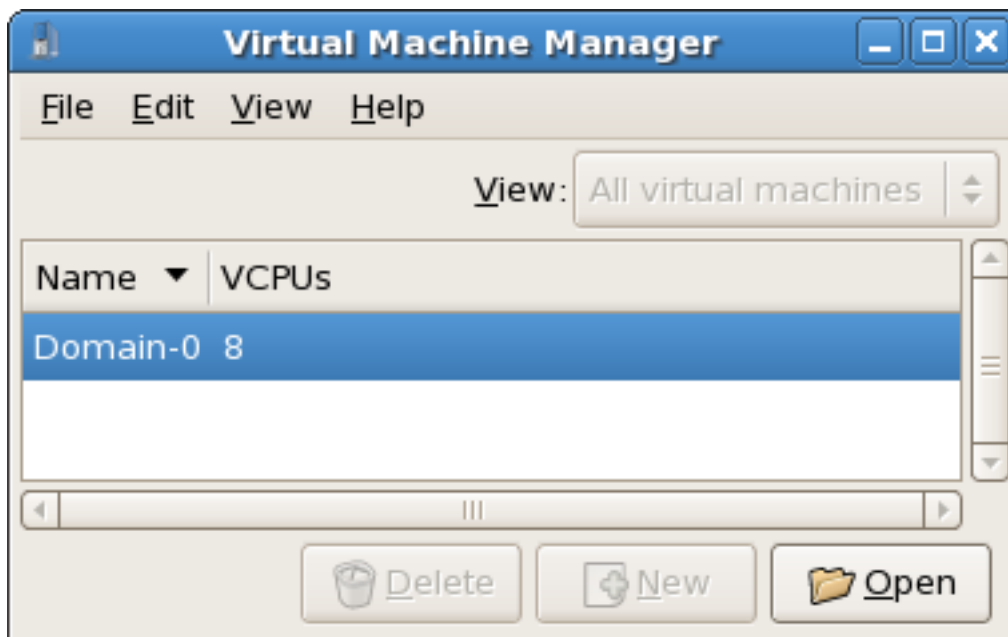
1. In het **View** menu selecteer je het **Virtual CPUs** aanvink hokje.



Figuur 16.22. De virtuele CPU's optie selecteren



2. De Virtual Machine Manager laat een lijst van de virtuele CPU's zien voor alle virtuele machines op jouw systeem.

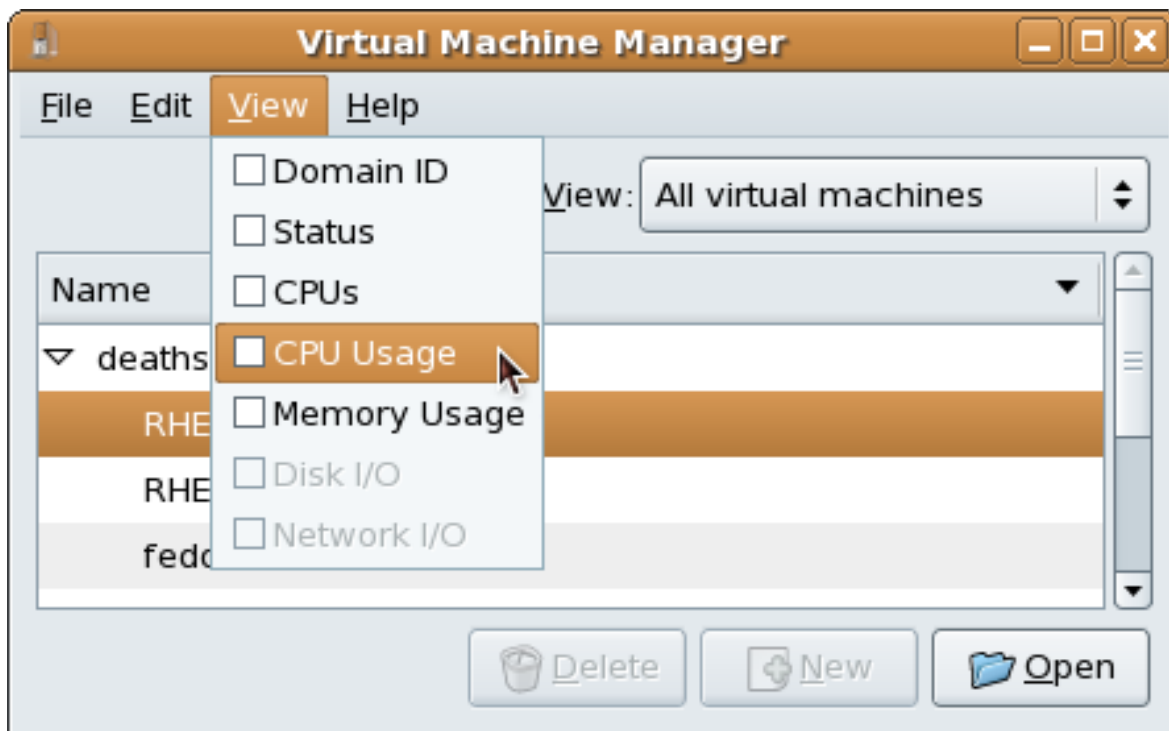


Figuur 16.23. Virtuele CPU's laten zien

## 16.12. CPU gebruik laten zien

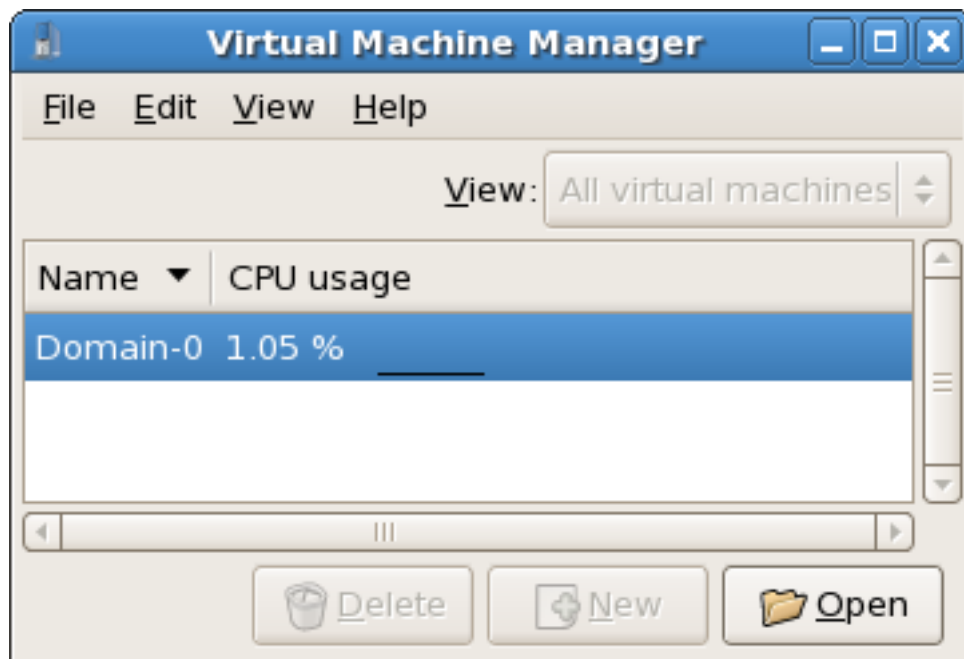
Om het CPU gebruik van alle virtuele machines op jouw systeem te laten zien:

1. In het **View** menu selecteer je het **CPU Usage** aanvink hokje.



Figuur 16.24. CPU gebruik selecteren

2. De Virtual Machine Manager laat een lijst zien van de percentages CPU gebruik voor alle virtuele machines op jouw systeem.

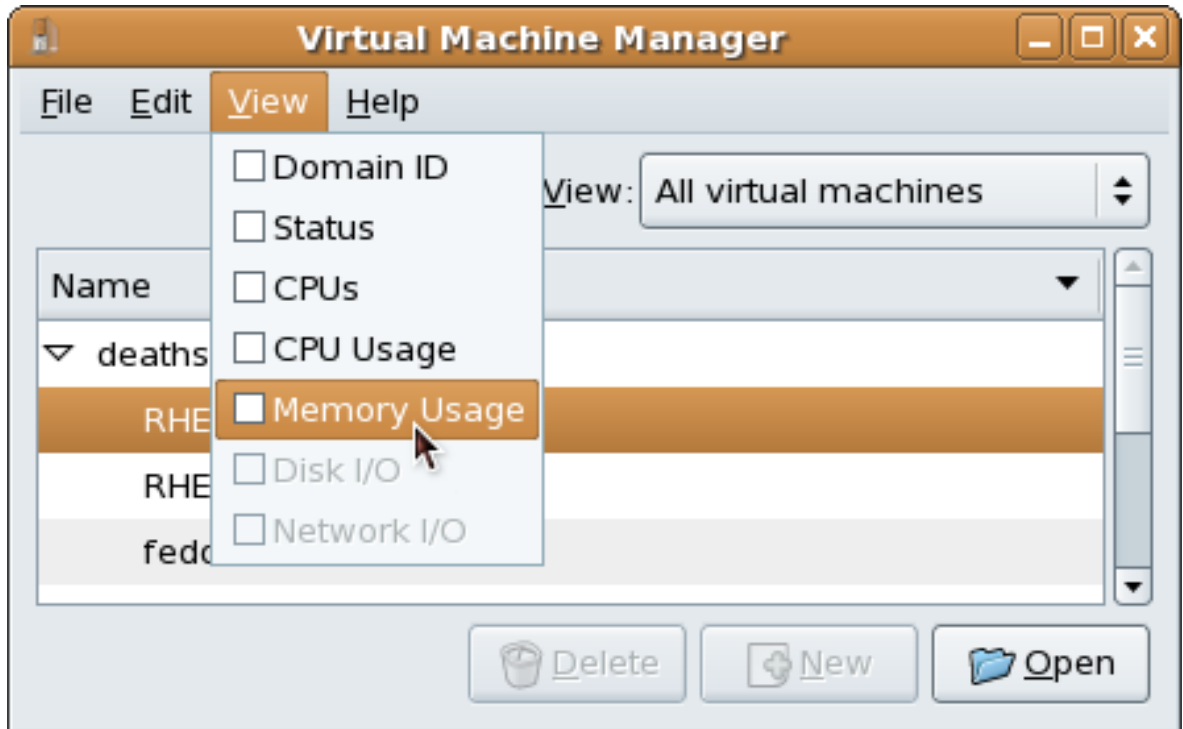


Figuur 16.25. CPU gebruik laten zien

### 16.13. Geheugen gebruik laten zien

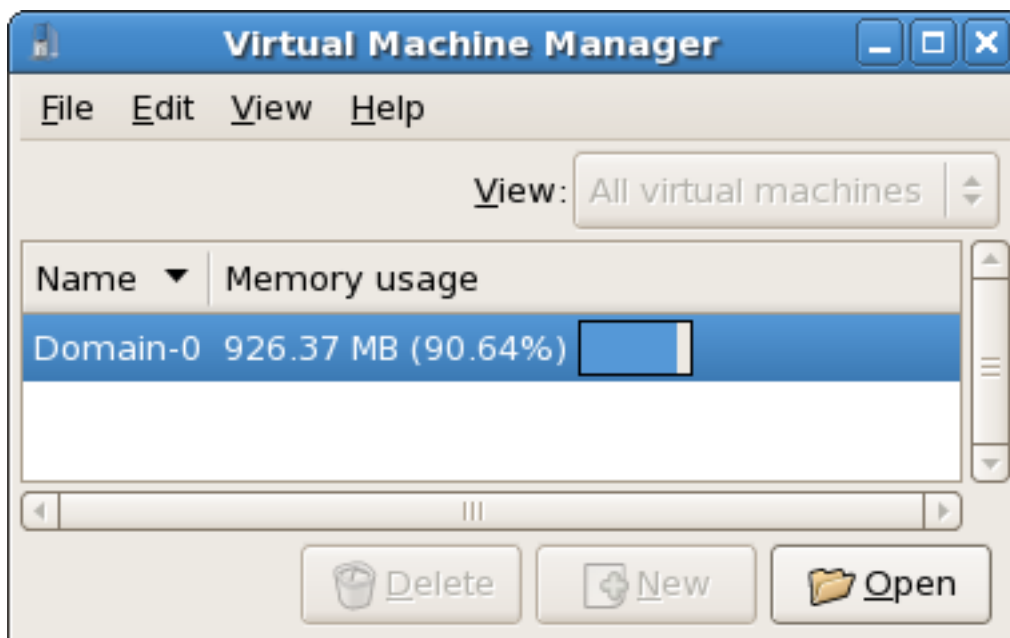
Om het geheugen gebruik van alle virtuele machines op jouw systeem te laten zien:

1. In het **View** menu selecteer je het **Memory Usage** aanvink hokje.



Figuur 16.26. Geheugen gebruik selecteren

- De Virtual Machine Manager laat een lijst zien van het percentage geheugen gebruik (in megabytes) voor alle virtuele machines op jouw systeem.

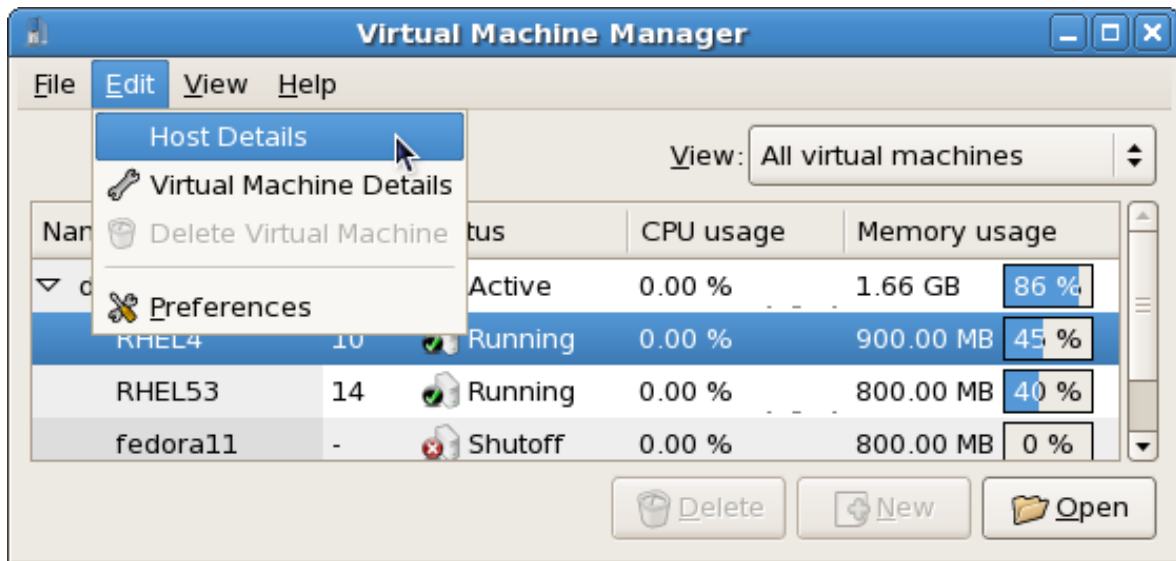


Figuur 16.27. Geheugen gebruik laten zien

## 16.14. Een virtueel netwerk beheren

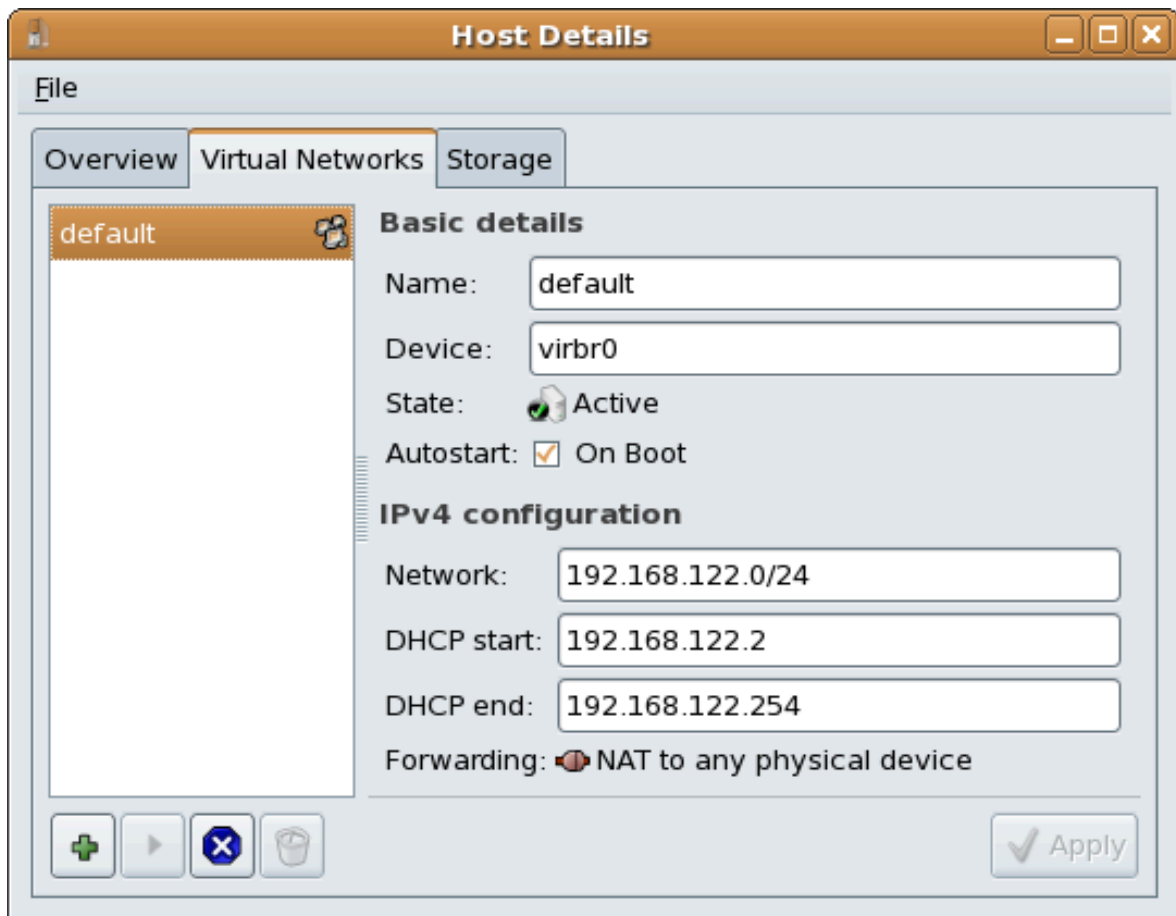
Om een virtueel netwerk op jouw systeem te configureren:

1. In het **Edit** menu selecteer je **Host Details**.



Figuur 16.28. De details van een host selecteren

2. Dit opent het **Host Details** menu. Klik op de **Virtual Networks** tab.



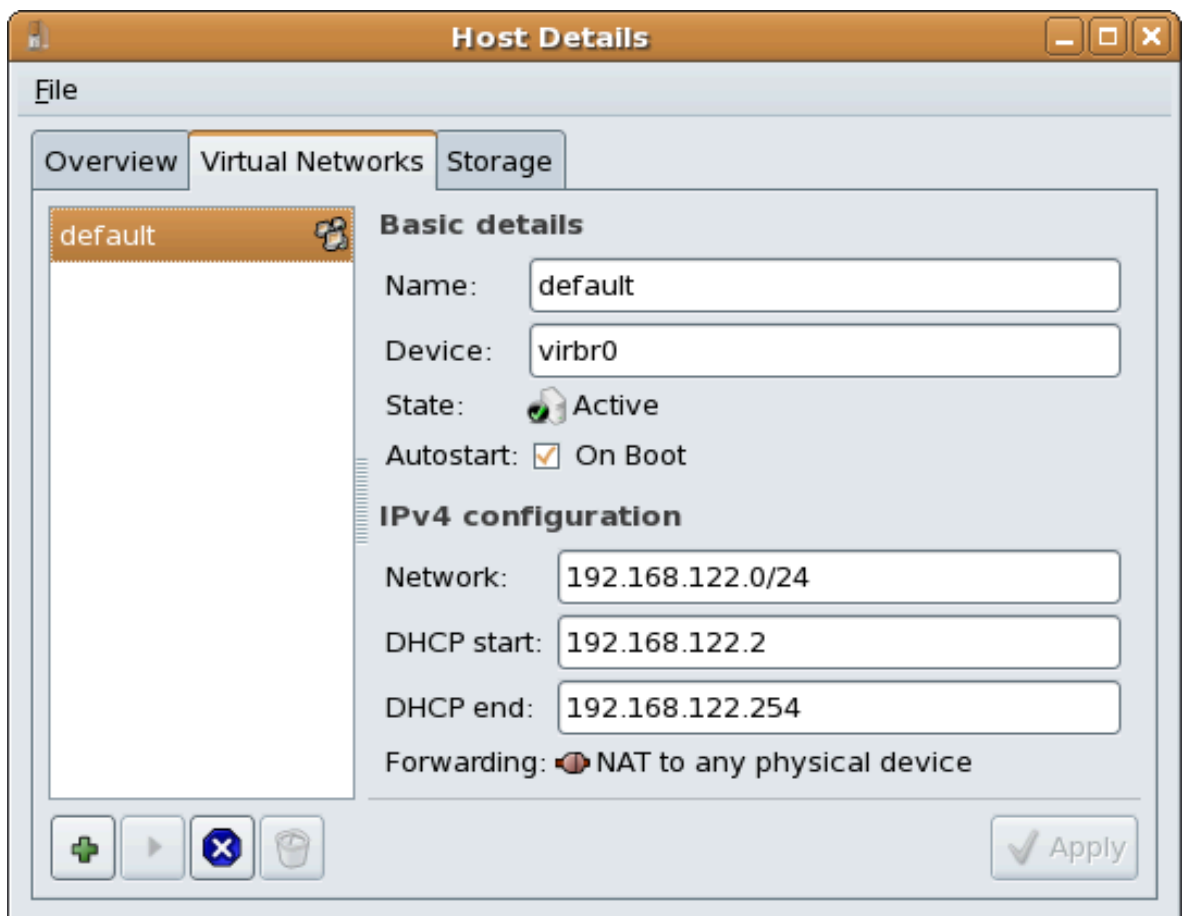
Figuur 16.29. Virtueel netwerk configuratie

3. Alle beschikbare virtuele netwerken worden in de linker rechthoek getoond. Je kunt de instelling van een virtueel netwerk bewerken door het hier te selecteren en naar eigen goeddunken bewerken.

## 16.15. Een virtueel netwerk aanmaken

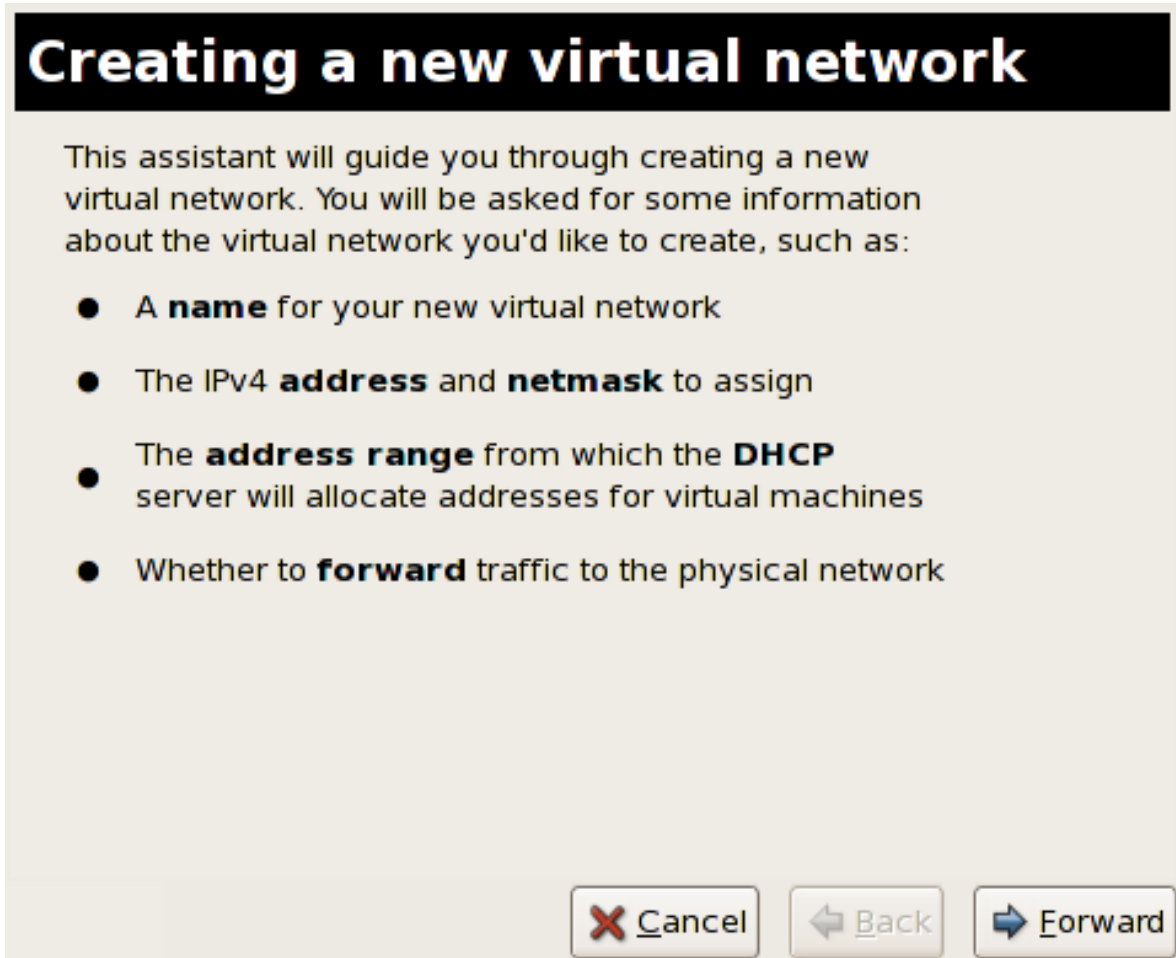
Om een virtueel netwerk op jouw systeem te maken:

1. Open het **Host Details** menu (refereer naar [Paragraaf 16.14](#), “Een virtueel netwerk beheren”) en klik op de **Add** knop.



Figuur 16.30. Virtueel netwerk configuratie

Dit opent het **Create a new virtual network** menu. Klik op **Forward** om verder te gaan.



Figuur 16.31. Een nieuw virtueel netwerk aanmaken

2. Vul een toepasselijke naam in voor jouw virtuele netwerk en klik op **Forward**.

## Naming your virtual network

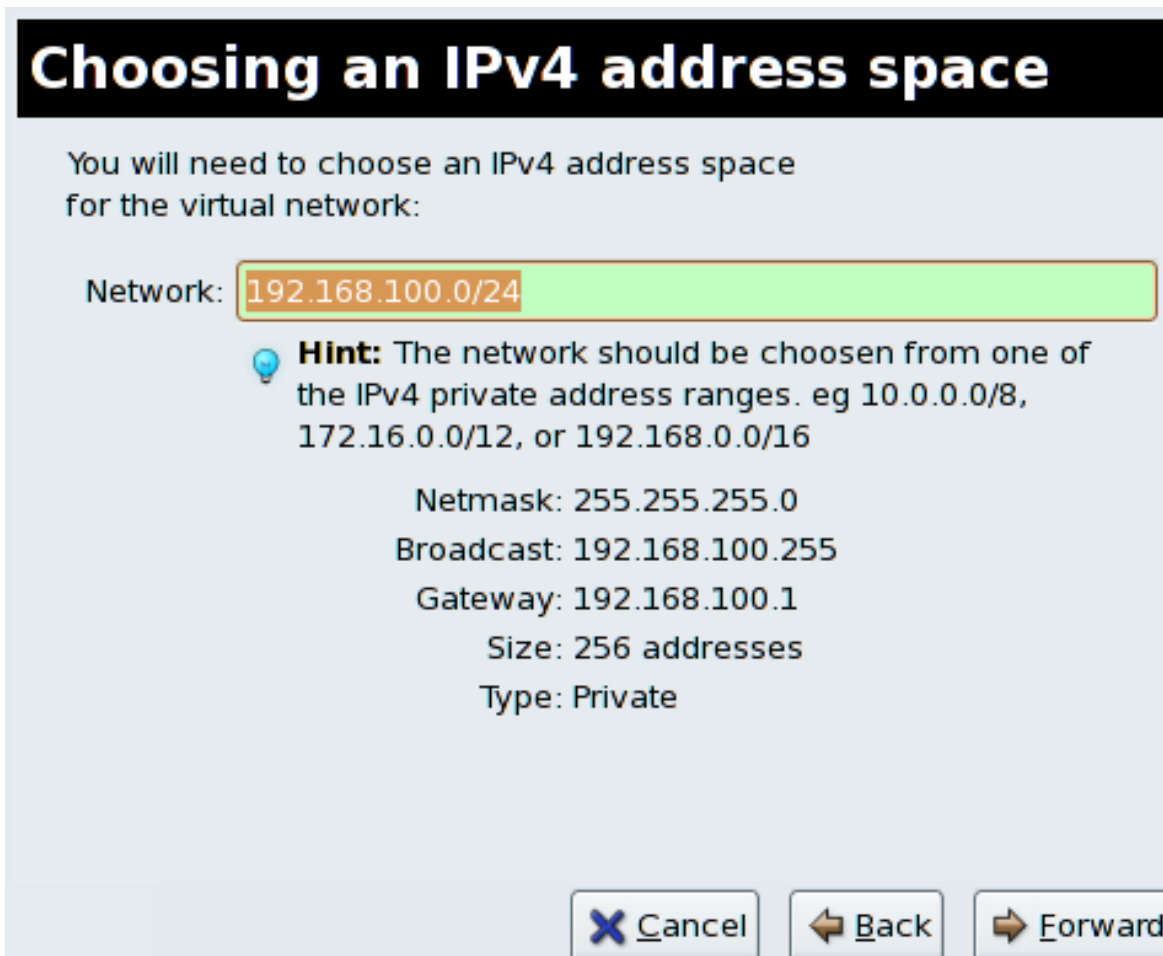
Please choose a name for your virtual network:

Network Name:

 **Example:** network1

Figuur 16.32. Jow virtueel netwerk een naam geven

3. Vul een IPv4 adres ruimte in voor jow virtuele netwerk en klik op **Forward**.



Figuur 16.33. Een IPv4 adres ruimte kiezen

4. Definieer de DHCP reeks voor jouw virtuele netwerk door het specificeren van een **Start** en **End** reeks van IP adressen. Klik op **Forward** om verder te gaan




## Selecting the DHCP range

Please choose the range of addresses the DHCP server can use to allocate to guests attached to the virtual network

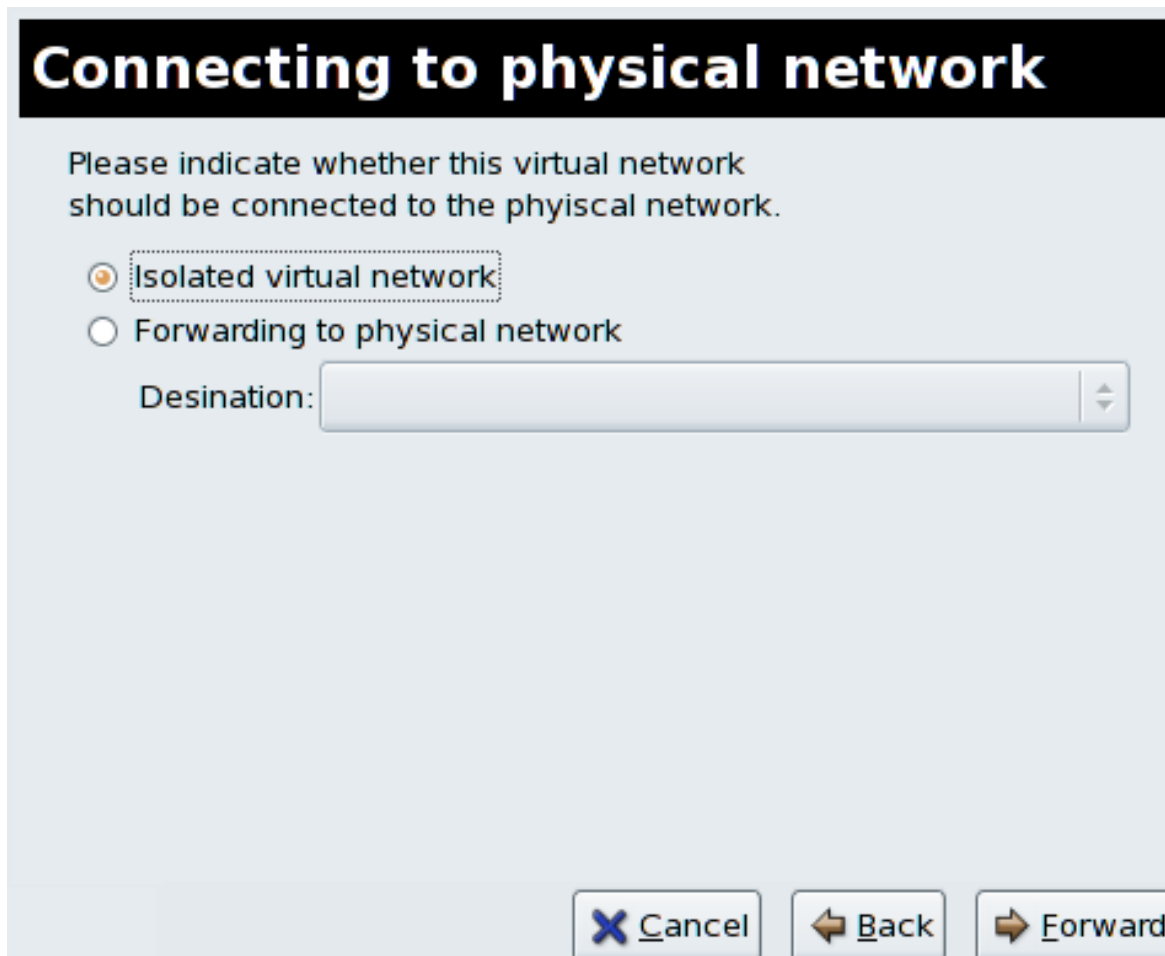
Start:

End:

 **Tip:** Unless you wish to reserve some addresses to allow static network configuration in virtual machines, these parameters can be left with their default values.

Figuur 16.34. Het selecteren van de DHCP reeks

5. Selecteer hoe het virtuele netwerk moet verbinden met het fysieke netwerk.

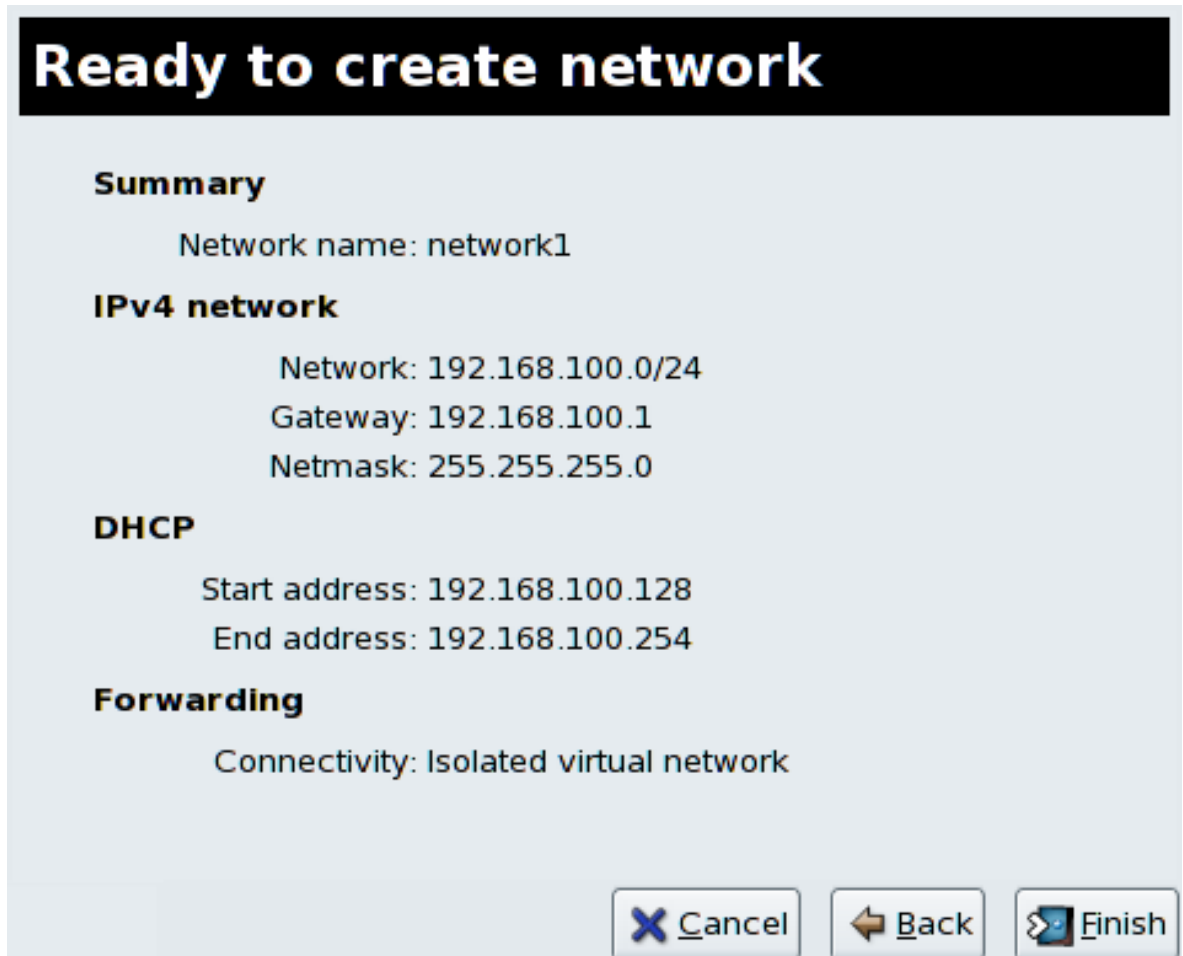


Figuur 16.35. Verbinden met het fysieke netwerk

Als je **Forwarding to physical network** selecteert, kies je of de **Destination** een **NAT to any physical device** of **NAT to physical device eth0** moet zijn.

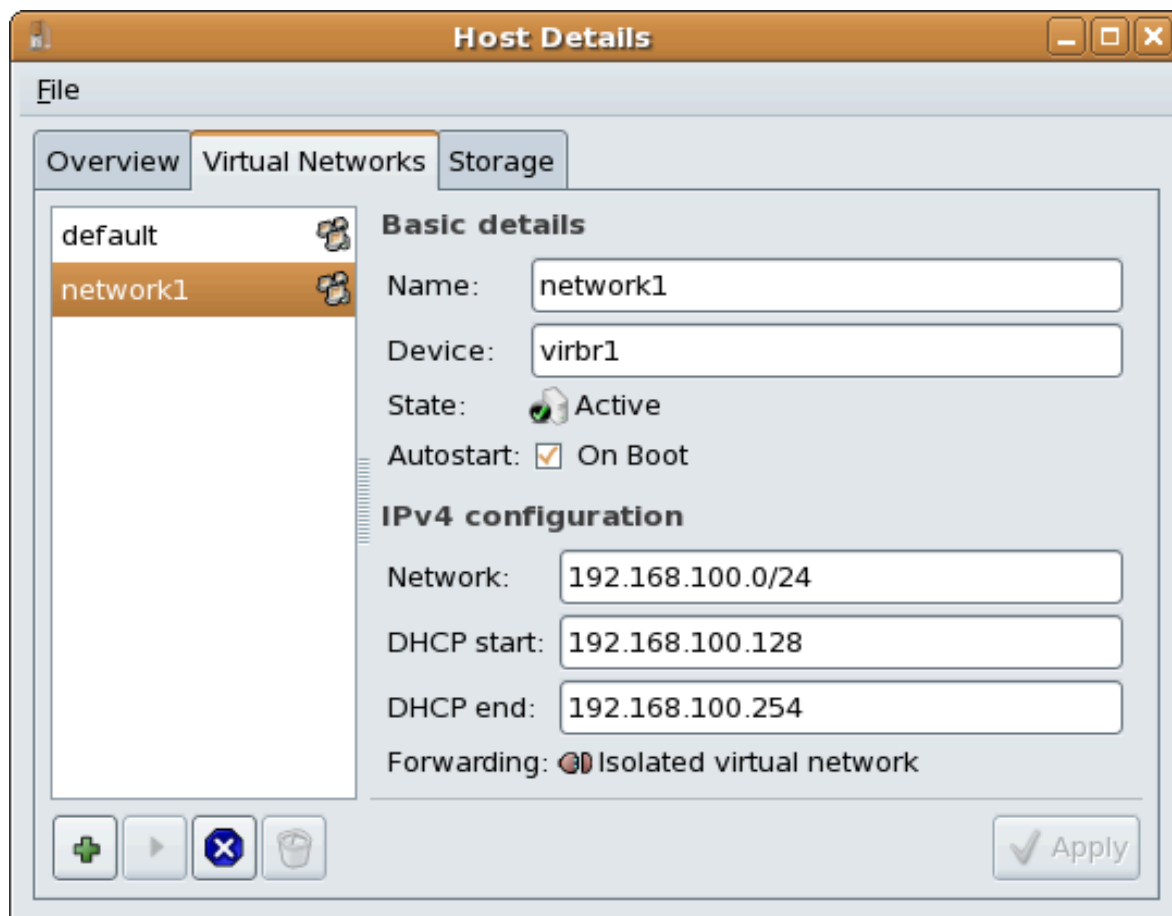
Click **Forward** to continue.

6. Je bent er nu klaar voor om het netwerk aan te maken. Controleer de instelling van jouw netwerk en klik op **Finish**.



Figuur 16.36. Klaar om het netwerk aan te maken

7. Het nieuwe virtuele netwerk is nu beschikbaar in de **Virtual Network** tab van het **Host Details** menu.



Figuur 16.37. Nieuwe virtuele netwerk is nu beschikbaar

---

## Deel V. Tips and Tricks

# Tips en trucjes om de productiviteit te verbeteren

Deze hoofdstukken bevatten nuttige hints en tips voor het verbeteren van virtualisatie prestaties, schaalbaarheid en stabiliteit.

---

---

---

# Tips en trucjes

Dit hoofdstuk bevat nuttige hints en tips voor het verbeteren van virtualisatie prestaties, schaalbaarheid en stabiliteit.

## 17.1. Guests automatisch starten

Deze paragraaf behandelt hoe je gevirtualiseerde guests automatisch op kunt starten tijdens de opstart fase van het host systeem.

Dit voorbeeld gebruikt **virsh** om een guest, *TestServer*, in te stellen om automatisch te starten als de host opgestart wordt.

```
virsh autostart TestServer
Domain TestServer marked as autostarted
```

De guest start nu automatisch op met de host.

Om het automatisch opstarten te stoppen gebruik je de `--disable` parameter

```
virsh autostart --disable TestServer
Domain TestServer unmarked as autostarted
```

De guest start niet meer automatisch op met de host.

## 17.2. Omschakelen tussen de KVM en Xen hypervisors

Deze paragraaf behandelt het omschakelen tussen de KVM en Xen hypervisors.

Fedora ondersteunt maar een actieve hypervisor.



### Gevirtualiseerde guests migreren tussen hypervisors

Op dit moment is er geen toepassing voor het omschakelen van Xen-gebaseerde guests naar KVM of KVM-gebaseerde guests naar Xen. Guests kunnen alleen gebruikt worden met de hypervisor waarmee ze aangemaakt zijn.

### 17.2.1. Xen naar KVM

De volgende procedure beschrijft het veranderen van de Xen hypervisor naar de KVM hypervisor. Deze procedure veronderstelt dat het *kernel-xen* pakket geïnstalleerd is en aangezet.

1. **Installeer het KVM pakket**

Installeer het *kvm* pakket als je dat nog niet gedaan hebt.

```
yum install kvm
```

2. **Controleer welke kernel in gebruik is**

Het *kernel-xen* pakket kan geïnstalleerd zijn. Gebruik het **uname** commando om te bepalen welke kernel draait:

```
$ uname -r
2.6.23.14-107.fc8xen
```

De "2.6.23.14-107.fc8xen" kernel draait op het systeem. Als de standaard kernel, "2.6.23.14-107.fc8" draait kun je deze tussenstap overslaan.

- **Het veranderen van de Xen kernel naar de standaard kernel**

Het **grub.conf** bestand bepaalt welke kernel opgestart wordt. Om de standaard kernel te veranderen bewerk je het **/boot/grub/grub.conf** bestand zoals hieronder getoond.

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
 root (hd0,0)
 kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/
LogVol00 rhgb quiet
 initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8
 module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/
VolGroup00/LogVol00 rhgb quiet
 module /initrd-2.6.23.14-107.fc8xen.img
```

Let op de **default=1** parameter. Deze instrueert de GRUB boot loader om de tweede regel, de Xen kernel, op te starten. Verander de default naar 0 (of het nummer voor de standaard kernel):

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
 root (hd0,0)
 kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/
LogVol00 rhgb quiet
 initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8
 module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/
VolGroup00/LogVol00 rhgb quiet
 module /initrd-2.6.23.14-107.fc8xen.img
```

### 3. Start opnieuw op om de nieuwe kernel te laden

Start het systeem opnieuw op. De computer zal herstarten met de standaard kernel. De KVM module moet automatisch geladen worden met de kernel. Verifieer of KVM draait:



```
$ lsmod | grep kvm
kvm_intel 85992 1
kvm 222368 2 ksm,kvm_intel
```

De `kvm` module en de `kvm_intel` module of de `kvm_amd` module zijn aanwezig als alles werkt.

## 17.2.2. KVM naar Xen

De volgende procedure behandelt het veranderen van de KVM hypervisor naar de Xen hypervisor. Deze procedure neemt aan dat het `kvm` pakket geïnstalleerd is en aangezet.

### 1. Installeer de Xen pakketten

Installeer de `kernel-xen` en `xen` pakketten als je dat nog niet gedaan hebt.

```
yum install kernel-xen xen
```

Het `kernel-xen` kan geïnstalleerd zijn maar uitgez.

### 2. Controleer welke kernel in gebruik is

Gebruik het `uname` commando om te bepalen welke kernel draait.

```
$ uname -r
2.6.23.14-107.fc8
```

De "**2.6.23.14-107.fc8**" kernel draait op het systeem. Dit is de standaard kernel. Als de kernel naam `xen` op het einde heeft (bijvoorbeeld, **2.6.23.14-107.fc8xen**) dan draait de Xen kernel en kun je deze tussenstap overslaan.

- **Het veranderen van de standaard kernel naar de Xen kernel**

Het `grub.conf` bestand bepaalt welke kernel opgestart wordt. Om de standaard kernel te veranderen bewerk je het `/boot/grub/grub.conf` bestand zoals hieronder getoond.

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
 root (hd0,0)
 kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/
LogVol00 rhgb quiet
 initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8
 module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/
VolGroup00/LogVol00 rhgb quiet
 module /initrd-2.6.23.14-107.fc8xen.img
```

Let op de **default=0** parameter. Deze instrueert de GRUB boot loader om de eerste regel, de standaard kernel, op te starten. Verander default naar **1** (of het nummer voor de Xen kernel):

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
 root (hd0,0)
 kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/
LogVol00 rhgb quiet
 initrd /initrd-2.6.23.14-107.fc82.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8
 module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/
VolGroup00/LogVol00 rhgb quiet
 module /initrd-2.6.23.14-107.fc8xen.img
```

### 3. Start opnieuw op om de nieuwe kernel te laden

Start het systeem opnieuw op. De computer zal herstarten met de Xen kernel. Verifieer dit met het **uname** commando:

```
$ uname -r
2.6.23.14-107.fc8xen
```

Als de output xen op het einde heeft dan draait de Xen kernel.

## 17.3. Het gebruik van qemu-img

Het **qemu-img** commando-regel gereedschap wordt gebruikt voor het formatteren van verschillende bestandssystemen die gebruikt worden door Xen en KVM. **qemu-img** moet gebruikt worden voor het formatteren van gevirtualiseerde guest images, extra opslag apparaten en netwerk opslag. **qemu-img** opties en gebruik worden hieronder beschreven.

### Het formatteren en aanmaken van nieuwe images of apparaten

Maakt een nieuw schijf image bestand-naam met opgegeven grootte en formaat.

```
qemu-img create [-6] [-e] [-b base_image] [-f formaat] bestand-naam
[grootte]
```

Als **base\_image** opgegeven is, zal de image alleen de verschillen met **base\_image** opslaan. In dat geval hoeft de grootte niet opgegeven worden. **base\_image** zal nooit veranderd worden behalve als je het "commit" monitor commando gebruikt.

### Converteer een bestaande image naar een ander formaat

De **convert** optie wordt gebruikt voor het converteren van een herkend formaat naar een ander image formaat.

Commando formaat:

```
qemu-img convert [-c] [-e] [-f formaat] bestandsnaam [-O output_formaat]
output_bestandsnaam
```

converteert de schijf image bestandsnaam naar schijf image output\_bestandsnaam met gebruik van het formaat output\_formaat. Het kan optioneel versleuteld ("-e" option) of gecomprimeerd ("-c" option) worden.

Alleen het formaat "qcow" ondersteunt versleuteling of compressie. De compressie is alleen-lezen. Dit betekent dat als een gecomprimeerde sector opnieuw wordt geschreven, dan wordt het geschreven als niet-gecomprimeerde data.

De versleuteling gebruikt het AES formaat met erg veilige 128 bit sleutels. Gebruik een lang wachtwoord (16 karakters) om de maximale bescherming te krijgen.

Image conversie is ook nuttig om een kleinere image te krijgen als een formaat wordt gebruikt dat kan groeien, zoals **qcow** of **cow**. De lege sectoren worden ontdekt en onderdrukt in de bestemmings image.

### Image informatie verkrijgen

De **info** parameter laat informatie zien over een schijf image. Het formaat voor de **info** optie is als volgt:

```
qemu-img info [-f formaat] bestandsnaam
```

Geeft informatie over de schijf image bestandsnaam. Gebruik het in het bijzonder om te weten te komen welke grootte voor de schijf gereserveerd is omdat die kan verschillen van de getoonde grootte. Als vm momentopnames in de schijf image zijn opgeslagen, worden deze ook getoond.

### Ondersteunde formaten

Het formaat van een image wordt gewoonlijk automatisch vastgesteld. De volgende formaten worden ondersteund:

#### raw

Raw schijf image formaat (standaard). Dit formaat heeft het voordeel dat het eenvoudig is en gemakkelijk exporteerbaar naar alle andere emulatoren. Als jouw bestandssysteem gaten ondersteunt (bijvoorbeeld in ext2 of ext3 in Linux of NTFS in Windows), dan zullen alleen geschreven sectoren ruimte reserveren. Gebruik **qemu-img info** om de werkelijke grootte gebruikt door de image te weten te komen of **ls -ls** in Unix/Linux.

#### qcow2

QEMU image formaat, het meest veelzijdige formaat. Gebruik het om kleinere images te krijgen (nuttig als jouw bestandssysteem geen gaten ondersteunt, bijvoorbeeld: in Windows), optionele AES versleuteling, zlib gebaseerde compressie en ondersteuning voor meerdere VM momentopnames.

#### qcow

Oud QEMU image formaat. Alleen toegevoegd voor compatibiliteit met oudere versies.

### cow

User Mode Linux Copy On Write image formaat. Het **cow** formaat is alleen toegevoegd voor compatibiliteit met vorige versies. Het werkt niet met Windows.

### vmdk

Image formaat compatibel met VMware 3 en 4.

### cloop

Linux Compressed Loop image, alleen nuttig voor hergebruik van direct gecomprimeerde CD-ROM images zoals bijvoorbeeld aanwezig in de Knoppix CD-ROM's.

## 17.4. Overcommit met KVM

De KVM hypervisor ondersteunt CPU overcommit en geheugen overcommit. Overcommit is het toekennen van meer gevirtualiseerde CPU's of geheugen dan er fysiek op het systeem aanwezig is. Met CPU overcommit kunnen zwak belaste gevirtualiseerde servers of desktops draaien op minder servers, wat vermogen en geld bespaart.



### Xen ondersteuning

CPU overcommit wordt niet ondersteund door de Xen hypervisor. CPU overcommit met de Xen hypervisor kan systeem instabiliteit en het crashen van de host en gevirtualiseerde guests veroorzaken.

### Overcommit van geheugen

De meeste besturingssystemen en toepassingen gebruiken niet de gehele tijd 100% van het beschikbare RAM. Dit gedrag kan uitgebuit worden met KVM om voor gevirtualiseerde guests meer geheugen te gebruiken dan dat er fysiek beschikbaar is.

In KVM zijn virtuele machines Linux processen. Guests op de KVM hypervisor krijgen geen blokken fysiek RAM toegekend maar werken in plaats daarvan als processen. Elk proces wordt geheugen toegekend als het meer geheugen vraagt. KVM gebruikt dit om geheugen toe te kennen aan guests als het guest besturingssysteem meer of minder geheugen verzoekt. De guest gebruikt slechts een beetje meer fysiek geheugen dan wat het gevirtualiseerde besturingssysteem lijkt te gebruiken.

Als het fysieke geheugen bijna op is of een proces al enige tijd niet actief, dan verhuist Linux het geheugen van het proces naar het wisselgeheugen. Het wisselgeheugen is gewoonlijk een partitie op een harde schijf of solid state station dat Linux gebruikt om virtueel geheugen uit te breiden. Wisselgeheugen is veel langzamer dan RAM.

Omdat KVM virtuele machines Linux processen zijn, kan geheugen gebruikt door gevirtualiseerde guests in het wisselgeheugen geplaatst worden als de guest niets doet of niet zwaar belast is. Geheugen kan toegekend worden voor het totaal van het wisselgeheugen en fysieke RAM. Als er niet voldoende wisselgeheugen aanwezig is voor de virtuele machines als ze dit nodig hebben dan wordt **pdflush** opgestart. **pdflush** sluit processen af om geheugen vrij te maken zodat het systeem niet crasht. **pdflush** kan gevirtualiseerde guests of andere systeem processen afsluiten wat bestandssysteem fouten kan veroorzaken en gevirtualiseerde guests in een niet-opstartbare toestand achterlaten.



## Warning

Als geen voldoende wisselgeheugen beschikbaar is zullen guest besturingssystemen met harde hand afgesloten worden. Dit kan de guest onbruikbaar achterlaten. Vermijd dit door nooit meer geheugen overcommit te doen dan dat er beschikbaar is aan wisselgeheugen.

De wisselgeheugen partitie wordt gebruikt voor het wegschrijven van weinig gebruikt geheugen naar de harde schijf om de geheugen prestaties te versnellen. De standaard grootte van de wisselgeheugen partitie wordt berekend uit de hoeveelheid RAM en de overcommit verhouding. Het wordt aanbevolen om jouw wisselgeheugen partitie groter te maken als je van plan bent om geheugen overcommit te gebruiken met KVM. Een aanbevolen overcommit verhouding is 50% (0.5). De gebruikte formule is:

$$(0.5 * \text{RAM}) + (\text{overcommit verhouding} * \text{RAM}) = \text{Aanbevolen wisselgeheugen grootte}$$

De Red Hat Knowledgebase heeft een artikel over het veilig en efficiënt bepalen van de grootte van de wisselgeheugen partitie— refereer naar [Knowledgebase](#)<sup>1</sup>.

Het is mogelijk om de draaien met een overcommit verhouding van tien keer het aantal gevirtualiseerde guests over de hoeveelheid fysieke RAM. Dit werkt alleen bij bepaalde toepassings belastingen (bijvoorbeeld desktop virtualisatie met minder dan 100% gebruik). Het instellen van overcommit verhoudingen is geen harde formule, je moet de verhouding testen en aanpassen voor jouw omgeving.

## Overcommit van gevirtualiseerde CPU's

De KVM hypervisor ondersteunt overcommit van gevirtualiseerde CPU's. Overcommit van gevirtualiseerde CPU's kan gebruikt worden voor zover de belasting limieten van gevirtualiseerde guests dit toestaan. Wees voorzichtig met overcommit van VCPU's als de belasting 100% nadert wat het laten vallen van verzoeken en onbruikbare response tijden kan veroorzaken.

Overcommit van gevirtualiseerde CPU's werkt het best als iedere gevirtualiseerde guest slechts een enkele VCPU heeft. De Linux scheduler is erg efficiënt met dit type belasting. KVM kan veilig guests met een belasting van onder de 100% ondersteunen met een verhouding van 5 VCPU's. Overcommit van een enkele VCPU gevirtualiseerde guests is geen probleem.

Je kunt geen overcommit uitvoeren voor symmetrische multi-proces guests voor meer dan het fysieke aantal processor kernen. Bijvoorbeeld een guest met vier VCPU's moet niet draaien op een host met een dual core processor. Overcommit van symmetrische multi-proces guests meer dan het fysieke aantal processor kernen zal belangrijke prestatie degradatie veroorzaken.

Guest VCPU's toekennen tot aan het aantal fysieke kernen is juist en werkt zoals verwacht. Bijvoorbeeld, het draaien van gevirtualiseerde guests met vier VCPU's op een host met vier fysieke kernen. Guests met minder dan 100% belasting moeten in deze opstelling effectief werken.

<sup>1</sup> <http://kbase.redhat.com/faq/docs/DOC-15252>



### Altijd eerst testen

Voer geen overcommit van geheugen of CPU's uit in een productie omgeving zonder uitvoerig te testen. Toepassingen welke 100% geheugen of proces hulpbronnen gebruiken kunnen onstabiel worden in overcommit omgevingen. Test voor het toepassen.

## 17.5. Het veranderen van `/etc/grub.conf`

Deze paragraaf beschrijft hoe je veilig en correct jouw `/etc/grub.conf` bestand kunt veranderen voor het gebruik van de virtualisatie kernel. Je moet de xen kernel gebruiken voor het gebruik van de Xen hypervisor. Bij het kopiëren van je bestaande xen kernel ingang wees er zeker van dat je alle belangrijke regels kopieert of jouw systeem zal paniekeren bij het opstarten (`initrd` zal een lengte van '0' hebben). Als je specifieke waarden voor de xen hypervisor nodig hebt, moet je deze toevoegen achteraan de xen regel van jouw grub ingang.

De output hieronder is een voorbeeld van een `grub.conf` ingang van een systeem dat het `kernel-xen` pakket draait. De `grub.conf` op jouw systeem kan anders zijn. Het belangrijke gedeelte in het voorbeeld hieronder is de sectie vanaf de `title` regel tot aan de volgende nieuwe regel.

```
#boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grub/splash.xpm.gz hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console

title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8 com1=115200,8n1
 module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/VolGroup00/
LogVol100
 module /initrd-2.6.23.14-107.fc8xen.img
```



### Een belangrijk punt wat betreft het bewerken van `grub.conf`...

Jouw `grub.conf` kan er heel anders uitzien als het handmatig bewerkt is of gekopieerd van een voorbeeld.

Om de hoeveelheid geheugen toegekend aan je host systeem tijdens het opstarten in te stellen op 256MB moet je `dom0_mem=256M` toevoegen van de xen regel in jouw `grub.conf`. Een veranderde versie van het grub configuratie bestand uit het vorige voorbeeld:

```
#boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console
```

```

title Fedora (2.6.23.14-107.fc8xen)
 root (hd0,0)
 kernel /xen.gz-2.6.23.14-107.fc8 com1=115200,8n1 dom0_mem=256MB
 module /vmlinuz-2.6.23.14-107.fc8xen ro
 root=/dev/VolGroup00/LogVol100
 module /initrd-2.6.23.14-107.fc8xen.img

```

## 17.6. Virtualisatie uitbreidingen verifiëren

Gebruik deze paragraaf om te bepalen of jouw systeem hardware virtualisatie uitbreidingen heeft. Virtualisatie uitbreidingen (Intel VT of AMD-V) zijn vereist voor volledige virtualisatie.



### Kan ik virtualisatie gebruiken zonder de virtualisatie uitbreidingen?

Als hardware virtualisatie uitbreidingen niet aanwezig zijn, kun je Xen para-virtualisatie gebruiken met het Fedora *kernel-xen* pakket.

Voer het volgende commando uit om te verifiëren of de CPU virtualisatie uitbreidingen beschikbaar zijn:

```
$ grep -E 'svm|vmx' /proc/cpuinfo
```

De volgende output bevat een vmx ingang wat een Intel processor met de Intel VT uitbreidingen aangeeft:

```

flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush
 dts acpi mmx fxsr sse sse2 ss ht tm syscall lm constant_tsc pni
 monitor ds_cpl
 vmx est tm2 cx16 xtpr lahf_lm

```

De volgende output bevat een svm ingang wat een AMD processor met de AMD-V uitbreidingen aangeeft:

```

flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush
 mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt lm 3dnowext 3dnow
 pni cx16
 lahf_lm cmp_legacy svm cr8legacy ts fid vid ttp tm stc

```

De "flags:" inhoud kan meerdere keren verschijnen voor iedere hyperthread, kern, of CPU in het systeem.

De virtualisatie uitbreidingen kunnen uitgezet zijn in de BIOS. Als de uitbreidingen niet verschijnen of volledige virtualisatie niet werkt, refereer je naar [Procedure 19.1, "Virtualisatie uitbreidingen aanzetten in de BIOS"](#).

### 17.7. Guest type en implementatie identificeren

Het script hieronder kan identificeren of de omgeving waarin een toepassing of script op een para-gevirtualiseerde, een volledig gevirtualiseerde guest, of op de hypervisor draait.

```
#!/bin/bash
declare -i IS_HVM=0
declare -i IS_PARA=0
check_hvm()
{
 IS_X86HVM="$(strings /proc/acpi/dsdt | grep int-xen)"
 if [x"${IS_X86HVM}" != x]; then
 echo "Guest type is full-virt x86hvm"
 IS_HVM=1
 fi
}
check_para()
{
 if $(grep -q control_d /proc/xen/capabilities); then
 echo "Host is dom0"
 IS_PARA=1
 else
 echo "Guest is para-virt domU"
 IS_PARA=1
 fi
}
if [-f /proc/acpi/dsdt]; then
 check_hvm
fi

if [${IS_HVM} -eq 0]; then
 if [-f /proc/xen/capabilities] ; then
 check_para
 fi
fi

if [${IS_HVM} -eq 0 -a ${IS_PARA} -eq 0]; then
 echo "Baremetal platform"
fi
```



#### Hosts onderzoeken

Om hosts te onderzoeken gebruik je het `virsh capabilities` commando.

### 17.8. Een nieuw uniek MAC adres aanmaken

In sommige gevallen moet je een nieuw en uniek *MAC adres* aanmaken voor een guest. Er is op dit moment geen commando-regel gereedschap beschikbaar voor het aanmaken van een MAC adres. De hieronder aangeboden script kan een nieuw MAC adres voor jouw guests aanmaken. Sla het script op in jouw guest als `macgen.py`. Vanuit die map kun je het script uitvoeren met gebruik van `./`



**macgen.py** en het zal een nieuw MAC adres aanmaken. Een voorbeeld output kan op het volgende lijken:

```
$./macgen.py
00:16:3e:20:b0:11

#!/usr/bin/python
macgen.py script to generate a MAC address for virtualized guests on Xen
#
import random
#
def randomMAC():
 mac = [0x00, 0x16, 0x3e,
 random.randint(0x00, 0x7f),
 random.randint(0x00, 0xff),
 random.randint(0x00, 0xff)]
 return ':'.join(map(lambda x: "%02x" % x, mac))
#
print randomMAC()
```

### Een andere manier om een nieuw MAC adres voor jouw guest te maken

Je kunt ook de ingebouwde modules van **python-virtinst** gebruiken om een nieuw MAC adres en **UUID** aan te maken voor gebruik in een guest configuratie bestand:

```
echo 'import virtinst.util ; print\
virtinst.util.uuidToString(virtinst.util.randomUUID())' | python
echo 'import virtinst.util ; print virtinst.util.randomMAC()' | python
```

Het script hierboven kan ook geïmplementeerd worden als een script bestand zoals hieronder te zien is.

```
#!/usr/bin/env python
-*- mode: python; -*-
print ""
print "New UUID:"
import virtinst.util ; print
virtinst.util.uuidToString(virtinst.util.randomUUID())
print "New MAC:"
import virtinst.util ; print virtinst.util.randomMAC()
print ""
```

## 17.9. Heel veilig ftpd

vsftpd kan toegang bieden tot installatie bomen voor para-gevirtualiseerde guests of andere data. Als je vsftpd niet hebt geïnstalleerd tijdens de server installatie kun je het RPM pakket uit de **Server** map van jouw installatie media halen en het installeren met het gebruik van **rpm -ivh vsftpd\*.rpm** (merk op dat het RPM pakket in je huidige map moet zijn).

1. Om vsftpd te configureren, bewerk je **/etc/passwd** met gebruik van **vi** en je verandert de persoonlijke map van de ftp gebruiker naar de map waar je de installatie bomen voor jouw para-

gevirtualiseerde guests gaat opslaan. Een voorbeeld ingang voor de FTP gebruiker kan lijken op het volgende:

```
ftp:x:14:50:FTP User:/xen/pub:/sbin/nologin
```

2. Om vsftpd automatisch te laten starten tijdens de systeem opstart gebruik je het chkconfig programma om de automatische opstart van vsftpd aan te zetten.
3. Verifieer dat vsftpd niet aangezet wordt met gebruik van **chkconfig --list vsftpd**:

```
$ chkconfig --list vsftpd
vsftpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

4. Voer **chkconfig --levels 345 vsftpd on** uit om vsftpd automatisch op te starten voor run levels 3, 4, en 5.
5. Gebruik het **chkconfig --list vsftpd** commando om te verifiëren dat vsftpd aangezet is om te starten tijdens de systeem opstart:

```
$ chkconfig --list vsftpd
vsftpd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

6. Gebruik **service vsftpd start vsftpd** om de vsftpd service te starten:

```
$service vsftpd start vsftpd
Starting vsftpd for vsftpd: [OK]
```

### 17.10. LUN vasthoudendheid instellen

Deze paragraaf behandelt het implementeren van [LUN](#) vasthoudendheid in guests en op de host machine met en zonder multipath.

#### LUN vasthoudendheid implementeren zonder multipath

Als jouw systeem multipath niet gebruikt, kun je **udev** gebruiken om LUN vasthoudendheid te implementeren. Voordat je LUN vasthoudendheid in jouw systeem installeert, verzeker je je ervan dat je de juiste UUID's hebt verkregen. Zodra je deze hebt verkregen, kun je LUN vasthoudendheid configureren door het bewerken van het **scsi\_id** bestand die zich in de **/etc** map bevindt. Als je dit bestand geopend hebt in een tekstverwerker, moet je de volgende regel uitcommentariëren:

```
options=-b
```

Daarna vervang je het met deze parameter:

```
options=-g
```

Dit zegt udev om alle systeem SCSI apparaten te controleren of ze UUID's teruggeven. Om de systeem UUID's te ontdekken, gebruik je het **scsi\_id** commando:

```
scsi_id -g -s /block/sdc
3600a0b80001327510000015427b625e
```

De lange reeks karakters in de output is de UUID. De UUID verandert niet als je een apparaat aan je systeem toevoegt. Verkrijg de UUID voor elk apparaat om regels voor de apparaten te maken. Om nieuwe regels te maken, bewerk je het **20-names.rules** bestand in de **/etc/udev/rules.d** map. De apparaat naam regels volgen dit formaat:

```
KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id", RESULT="UUID",
NAME="apparaatnaam"
```

Vervang je bestaande *UUID* en *apparaatnaam* met die hierboven verkregen UUID. De regel moet op het volgende lijken:

```
KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id",
RESULT="3600a0b80001327510000015427b625e", NAME="mijnapparaat"
```

Cit zorgt ervoor dat alle apparaten die overeenkomen met het **/dev/sd\*** patroon de gegeven UUID gaan inspecteren. Als het een overeenkomend apparaat vindt, maakt het een apparaat node met de naam */dev/devicename*. Voor dit voorbeeld wordt de apparaat node */dev/mijnapparaat*. Tenslotte voeg je deze regel toe aan het **/etc/rc.local** bestand:

```
/sbin/start_udev
```

### LUN vasthoudendheid implementeren met multipath

Om LUN vasthoudendheid te implementeren in een multipath omgeving, moet je de alias namen voor de multipath apparaten definiëren. Bijvoorbeeld, je moet vier apparaat aliases definiëren door het bewerken van het **multipath.conf** bestand dat zich in de **/etc/** map bevindt:

```
multipath {
 wwid 3600a0b80001327510000015427b625e
 alias oramp1
}
multipath {
 wwid 3600a0b80001327510000015427b6
 alias oramp2
}
multipath {
 wwid 3600a0b80001327510000015427b625e
 alias oramp3
}
multipath {
 wwid 3600a0b80001327510000015427b625e
 alias oramp4
}
```

Dit definieert 4 LUN's: **/dev/mpath/oramp1**, **/dev/mpath/oramp2**, **/dev/mpath/oramp3**, en **dev/mpath/oramp4**. De apparaten zullen zich in de **/dev/mpath** map bevinden. Deze LUN

namen zijn vasthoudend bij opnieuw opstarten omdat het de alias namen aanmaakt op de wwid van de LUN's.

### 17.11. SMART schijf controleren uitzetten

SMART schijf controleren kan uitgezet worden omdat we op virtuele schijven draaien en de fysieke opslag wordt door de host beheerd.

```
/sbin/service smartd stop
/sbin/chkconfig --del smartd
```

### 17.12. Guest configuratie bestanden klonen

Je kunt een bestaand configuratie bestand kopiëren om een geheel nieuwe guest aan te maken. Je moet de name parameter van het configuratie bestand van de guest veranderen. De nieuwe, unieke naam verschijnt in de hypervisor en is zichtbaar voor de beheer programma's. Je moet ook een geheel nieuwe UUID genereren met gebruik van het **uuidgen** commando. Daarna moet je voor de **vif** ingangen een uniek MAC adres definiëren voor iedere guest (als je een guest configuratie van een bestaande guest kopieert, kun je een script maken om dit af te handelen). Voor de xen brug informatie, indien een bestaand guest configuratie bestand verhuist naar een nieuwe host, moet je de **xenbr** ingang vernieuwen om overeen te komen met je locale netwerk configuratie. Voor de Device ingangen moet je de ingangen veranderen in de '**disk=**' sectie om te wijzen naar de correcte guest image.

Je moet ook deze systeem configuratie instellingen op je guest veranderen. Je moet de HOSTNAME ingang van het **/etc/sysconfig/network** bestand veranderen om overeen te komen met de hostnaam van de nieuwe guest.

Je moet het **HWADDR** adres van het **/etc/sysconfig/network-scripts/ifcfg-eth0** bestand veranderen om overeen te komen met de output van **ifconfig eth0** en als je statische IP adressen gebruikt, moet je de **IPADDR** ingang veranderen.

### 17.13. Een bestaande guest dupliceren en zijn configuratie bestand

Deze paragraaf beschrijft het kopiëren van een bestaand configuratie bestand om een nieuwe guest aan te maken. Er zijn sleutel parameters in het configuratie bestand van jouw guest waarvan je op de hoogte moet zijn om een guest succesvol te dupliceren.

name

De naam van jouw guest is bekend aan de hypervisor en wordt getoond in de beheer programma's. Deze ingang moet uniek zijn op jouw systeem.

uuid

Een uniek handvat voor de guest, een nieuwe UUID kan aangemaakt worden met gebruik van het **uuidgen** commando. Een voorbeeld UUID output:

```
$ uuidgen
a984a14f-4191-4d14-868e-329906b211e5
```

vif

- Het *MAC adres* moet een uniek MAC adres definiëren voor iedere guest. Dit wordt automatisch gedaan als de standaard gereedschappen gebruikt worden. Als je een guest configuratie kopieert van een bestaande guest kun je het script in [Paragraaf 17.8, “Een nieuw uniek MAC adres aanmaken”](#) gebruiken.
- Als je een bestaand guest configuratie bestand verplaatst of dupliceert naar een nieuwe host moet je er zeker van zijn dat je de `xenbr` ingang aanpast om overeen te komen met jouw locale netwerk configuratie (je kunt de brug informatie verkrijgen door gebruik van het `brctl show` commando).
- Wees er zeker van dat je de apparaat ingangen in de `disk=` sectie aanpast om te wijzen naar de juiste guest image.

Pas nu de systeem configuratie instellingen van je guest aan:

#### **/etc/sysconfig/network**

Verander de `HOSTNAME` ingang naar de nieuwe **hostnaam** van de guest.

#### **/etc/sysconfig/network-scripts/ifcfg-eth0**

- Verander het `HWADDR` adres naar de output van `ifconfig eth0`
- Verander de `IPADDR` ingang als een statisch IP adres gebruikt wordt.



---

# Aangepaste libvirt scripts aanmaken

Deze paragraaf geeft enige informatie welke nuttig kan zijn voor programmeurs en systeem beheerders die van plan zijn om aangepaste scripts te schrijven om hun leven wat gemakkelijker te maken met gebruik van **libvirt**.

*Hoofdstuk 17, Tips en trucjes* wordt aanbevolen om te lezen voor programmeurs die erover denken om nieuwe toepassingen te schrijven die **libvirt** gebruiken.

## 18.1. XML configuratie bestanden gebruiken met virsh

**virsh** kan XML configuratie bestanden afhandelen. Je wilt dit misschien tot jouw voordeel gebruiken voor het maken van scripts voor grotere inzet met speciale opties. Je kunt apparaten toevoegen die gedefinieerd zijn in een XML bestand om een para-gevirtualiseerde guest te draaien. Bijvoorbeeld, om een ISO bestand toe te voegen als **hdc** aan een draaiende guest, maak je een XML bestand:

```
cat satelliteiso.xml
<disk type="file" device="disk">
 <driver name="file"/>
 <source file="/var/lib/libvirt/images/rhn-satellite-5.0.1-11-
redhat-linux-as-i386-4-embedded-oracle.iso"/>
 <target dev="hdc"/>
 <readonly/>
</disk>
```

Voer **virsh attach-device** uit om het ISO bestand aan te koppelen als **hdc** voor een guest met de naam "satellite" :

```
virsh attach-device satellite satelliteiso.xml
```





---

## Deel VI. Troubleshooting

# Inleiding tot fout zoeken en probleem oplossen

De volgende hoofdstukken bieden informatie om je te helpen met het oplossen van problemen die tegen bent gekomen met het gebruik van virtualisatie.



### Belangrijke opmerking over virtualisatie problemen

Jouw persoonlijke probleem komt misschien niet in dit boek voor door de voortdurende ontwikkeling welke problemen veroorzaakt en oplost. Voor de laatste stand van zaken voor bekende fouten, problemen en fout reparaties lees je de *Fedora Vrijgave informatie* voor jouw versie en hardware architectuur. De *Vrijgave informatie* kan gevonden worden in de documentatie sectie van de Fedora website, <http://docs.fedoraproject.org>.

---

---

---

# Troubleshooting

Dit hoofdstuk behandelt algemene problemen en oplossingen voor Fedora virtualisatie.

## 19.1. Loop apparaat fouten

Als bestand-gebaseerde guest images worden gebruikt moet je misschien het aantal ingestelde loop apparaten verhogen. De standaard instelling laat tot 8 actieve loop apparaten toe. Als meer dan 8 bestand-gebaseerde guests of loop apparaten nodig zijn, dan kan het aantal ingestelde loop apparaten aangepast worden in `/etc/modprobe.conf`. Bewerk `/etc/modprobe.conf` en voeg de volgende regel toe:

```
options loop max_loop=64
```

Dit voorbeeld gebruikt 64, maar je kunt een ander getal opgeven om het maximum aantal loop apparaten in te stellen. Je moet misschien ook loop apparaat ondersteunde guests op je systeem implementeren. Om loop apparaat ondersteunde guests in te zetten voor een para-gevirtualiseerde guest, gebruik je de **phy: block device** of **tap:aio** commando's. Om loop apparaat ondersteunde guests voor een volledig gevirtualiseerd systeem toe te passen, gebruik je de **phy: device** of **file: file** commando's.

## 19.2. Intel VT en AMD-V virtualisatie hardware uitbreidingen aanzetten in de BIOS

Deze paragraaf beschrijft hoe je hardware virtualisatie uitbreidingen identificeert en hoe je ze in de BIOS aanzet als ze uitgezet zijn.

De Intel VT uitbreidingen kunnen uitgezet zijn in de BIOS. Bepaalde laptop leveranciers hebben de Intel VT uitbreidingen standaard uitgezet voor hun CPU's.

De virtualisatie uitbreidingen kunnen niet uitgezet worden in de BIOS voor AMD-V (voor processoren geïnstalleerd in een Rev 2 voetje).

De virtualisatie uitbreidingen zijn soms uitgezet in de BIOS, gewoonlijk door laptop leveranciers. Refereer naar [Paragraaf 19.2, "Intel VT en AMD-V virtualisatie hardware uitbreidingen aanzetten in de BIOS"](#) voor instructies over het aanzetten van de uitgezette virtualisatie uitbreidingen.

Verifieer of de virtualisatie uitbreidingen aangezet zijn in de BIOS. De BIOS instellingen voor Intel® VT of AMD-V zijn gewoonlijk in het **Chipset** of **Processor** menu. De menu namen kunnen anders zijn dan genoemd in deze gids, de virtualisatie uitbreidingen instellingen kunnen misschien gevonden worden in **Security Settings** of andere niet standaard menu namen.

### Procedure 19.1. Virtualisatie uitbreidingen aanzetten in de BIOS

1. Start de computer opnieuw op en open het BIOS menu van het systeem. Dit kan gewoonlijk gedaan worden door het induwen van **delete** of **Alt + F4**.
2. Selecteer **Restore Defaults**, en selecteer dan **Save & Exit**.
3. Zet de machine uit en koppel de voeding af.

4. Zet de machine weer aan en open de **BIOS Setup Utility**. Open de **Processor** sectie en zet **Intel®Virtualization Technology** of **AMD-V** aan. Deze kunnen op sommige machines ook **Virtualization Extensions** genoemd worden. Selecteer **Save & Exit**.
5. Zet de machine uit en koppel de voeding af.
6. Voer `cat /proc/cpuinfo | grep vmx svm` uit. Als het commando output geeft, dan zijn de virtualisatie uitbreidingen nu aangezet. Als er geen output is, dan heeft jouw systeem de virtualisatie uitbreidingen misschien niet of is de correcte BIOS instelling niet aangezet.

---

# Bijlage A. Extra hulpbronnen

Om meer te weten te komen over virtualisatie en Linux, refereer je naar de volgende hulpbronnen.

## A.1. On-line hulpbronnen

- <http://www.cl.cam.ac.uk/research/srg/netos/xen/> De project website van de Xen™ para-gevirtualiseerde machine beheerder waarvan het Fedora *kernel-xen* pakket is afgeleid. De site onderhoudt de upstream xen project binaire programma's en bron code en bevat ook informatie, architectuur overzichten, documentatie, en gerelateerde verwijzingen war betreft xen en aanverwante technologie.
- De Xen Community website  
<http://www.xen.org/>
- <http://www.libvirt.org/> is de officiële website voor de **libvirt** virtualisatie API.
- <http://virt-manager.et.redhat.com/> is de project website voor de **Virtual Machine Manager** (virt-manager), de grafische toepassing voor het beheren van virtuele machines.
- Open Virtualization Center  
<http://www.openvirtualization.com><sup>1</sup>
- Fedora documentatie  
<http://docs.fedoraproject.org>
- Virtualisatie technologie overzicht  
<http://virt.kernelnewbies.org><sup>2</sup>
- Red Hat Emerging Technologies group  
<http://et.redhat.com><sup>3</sup>

## A.2. Geïnstalleerde documentatie

- `/usr/share/doc/xen-<versie-nummer>/` is de map welke informatie bevat over de Xen para-gevirtualiseerde hypervisor en aanverwante beheer gereedschappen, inclusief verschillende voorbeeld configuraties, hardware specifieke informatie, en de huidige Xen upstream gebruikers documentatie.
- `man virsh` en `/usr/share/doc/libvirt-<versie-nummer>` — Bevatten sub-commando's en opties voor het **virsh** virtuele machine beheer programma en uitgebreide informatie over de **libvirt** virtualisatie bibliotheek API.
- `/usr/share/doc/gnome-applet-vm-<versie-nummer>` — Documentatie voor de GNOME grafische paneel applet die lokaal draaiende virtuele machines controleert en beheert.
- `/usr/share/doc/libvirt-python-<versie-nummer>` — Biedt details over de Python verwijzingen voor de **libvirt** bibliotheek. Het **libvirt-python** pakket staat python

ontwikkelaars toe om programma's te maken die samenwerken met de **libvirt** virtualisatie beheer bibliotheek.

- `/usr/share/doc/python-virtinst-<versie-nummer>` — Biedt documentatie over het **virt-install** commando dat helpt met het starten van installaties van Fedora en Linux gerelateerde distributies binnen virtuele machines.
- `/usr/share/doc/virt-manager-<versie-nummer>` — Biedt documentatie over de Virtual Machine Manager, welke een grafisch gereedschap levert voor het beheren van virtuele machines.

---

# Bijlage B. Herzieningsgeschiedenis

Herziening  
12.1.3

Mon Oct 12 2009

Christopher Curran [ccurran@redhat.com](mailto:ccurran@redhat.com)

Afsplitsen van Red Hat Enterprise Linux 5.4 Virtualization Guide version 5.4-61.





---

## Bijlage C. Colofon

Deze handleiding werd geschreven in het DocBook XML v4.3 formaat.

Dit boek is gebaseerd op het werk van Jan Mark Holzer en Chris Curran.

Andere verdiensten gaan naar:

- Don Dutile leverde technische opmaak voor de para-virtualisatie driver paragraaf.
- Barry Donahue leverde technische opmaak voor de para-virtualisatie driver paragraaf.
- Rick Ring leverde technische opmaak voor de virtuele machine beheerder paragraaf
- Michael Kearey leverde technische opmaak voor de paragrafen over het gebruik van XML configuratie bestanden met virsh en gevirtualiseerde floppy disk stations.
- Marco Grigull leverde technische opmaak voor de software compatibiliteit en prestatie paragraaf.
- Eugene Teo leverde technische opmaak voor de Gasten beheren met virsch paragraaf.

Publican, het uifgeef gereedschap die dit boek produceerde, werd geschreven door Jeffrey Fearn.

Het Red Hat Localisatie Team bestaat uit de volgende personen:

### Oost Aziatische talen

- Vereenvoudigd Chinees
  - Leah Wei Liu
- Traditioneel Chinees
  - Chester Cheng
  - Terry Chuang
- Japans
  - Junko Ito
- Koreaans
  - Eun-ju Kim

### Latijnse talen

- Frans
  - Sam Friedmann
- Duits
  - Hedda Peters
- Italiaans

- Francesco Valente
- Braziliaans Portugees
  - Glaucia de Freitas
  - Leticia de Lima
- Spaans
  - Angela Garcia
  - Gladys Guerrero
- Russisch
  - Yuliya Poyarkova

---

# Woordenlijst

Deze verklarende woordenlijst is bedoeld om de begrippen te definiëren die in deze Virtualisatie gids gebruikt worden.

Bare-metal	Het begrip bare-metal refereert naar de onderliggende fysieke architectuur van de computer. Het uitvoeren van een besturingssysteem op bare-metal is een andere manier om te refereren naar het draaien van een onveranderde versie van het besturingssysteem op de fysieke hardware. Voorbeelden van besturingssystemen die op bare-metal draaien zijn <i>dom0</i> of een normaal geïnstalleerd besturingssysteem.
dom0	Ook bekend als de <i>Host</i> of host besturingssysteem.  <b>dom0</b> refereert naar de host versie van Linux die de <i>Hypervisor</i> draait, welke de virtualisatie van de guest besturingssystemen mogelijk maakt. Dom0 draait op en beheert de fysieke hardware en hulpbronnen toewijzing voor zichzelf en de guest besturingssystemen.
Domeinen	<i>domU</i> en <i>Domeinen</i> zijn beide domeinen. Domeinen draaien op de <i>Hypervisor</i> . De term domeinen heeft een betekenis vergelijkbaar met <i>Virtuele machines</i> en de twee zijn technisch uitwisselbaar. Een domein is een Virtuele machine.
domU	<b>domU</b> refereert naar het guest besturingssysteem welke op het host systeem draait ( <i>Domeinen</i> ).
Volledige virtualisatie	Xen en KVM kunnen volledige virtualisatie gebruiken. Volledige virtualisatie gebruikt hardware kenmerken van de processor om een totale abstractie van het onderliggende fysieke systeem ( <i>Bare-metal</i> ) aan te bieden en maakt een nieuw virtueel systeem waarin de guest besturingssystemen kunnen draaien. Geen veranderingen zijn nodig in het guest besturingssysteem. Het guest besturingssysteem en alle toepassingen op de guest zijn zich niet bewust van de gevirtualiseerde omgeving en draaien normaal. Para-virtualisatie vereist een aangepaste versie van het Linux besturingssysteem.
Volledig gevirtualiseerd	Zie <i>Volledige virtualisatie</i> .
Guest systeem	Staat ook bekend als guest, virtuele machine of <i>domU</i> .
Hardware virtuele machine	Zie <i>Volledige virtualisatie</i>
Hypervisor	De hypervisor is de software laag die de hardware abstraheert van het besturingssysteem wat toestaat om meerdere besturingssystemen op dezelfde hardware te draaien. De hypervisor draait op een host besturingssysteem wat toestaat dat andere gevirtualiseerde besturingssystemen op de hardware van de host draaien.
Host	Het host besturingssysteem, ook bekend als <i>dom0</i> .

De host besturingssysteem omgeving draait de virtualisatie software voor *Volledig gevirtualiseerd* en *Para-gevirtualiseerd* guest systemen.

I/O

Afkorting van input/output (uitgesproken als "aai-oh") De term I/O beschrijft elk programma, operatie of apparaat dat data overbrengt naar of van een computer en naar of van een randapparaat. Elke overdracht is een output van een apparaat en een input voor een andere. Apparaten zoals toetsenborden en muizen zijn alleen-input apparaten, terwijl apparaten zoals printers alleen-output zijn. Een beschrijfbare CD-ROM is zowel een input als een output apparaat.

Kernel-gebaseerde virtuele machine

KVM (Kernel-gebaseerde virtuele machine) is een *Volledige virtualisatie* oplossing voor Linux op AMD64 en Intel 64 hardware. VM is een Linux kernel module gebouwd voor de standaard Linux kernel. KVM kan meerdere, onveranderde gevirtualiseerde Windows en Linux bedrijfssystemen draaien. KVM is een hypervisor welke de libvirt virtualisatie gereedschappen (virt-manager en virsh) gebruikt.

KVM is een set Linux kernel modules voor het beheer van apparaten, geheugen en beheers API's voor de Hypervisor module zelf. Gevirtualiseerde guests worden gedraaid als Linux processen en threads die gecontroleerd worden door deze modules.

LUN

Een Logische unit nummer (LUN) is een getal toegekend aan een logische unit (een SCSI protocol eenheid).

Migratie

Migratie is de naam voor het proces van het verplaatsen van een gevirtualiseerde guest van een host naar een andere. Migratie kan off-line uitgevoerd worden (waarbij de guest gestopt wordt en dan verplaatst) of live (waarbij een guest verplaatst wordt zonder te stoppen). Xen para-gevirtualiseerde guests en KVM volledig gevirtualiseerde guests kunnen alle gemigreerd worden.

Migratie is een sleutel kenmerk van virtualisatie omdat software geheel gescheiden is van hardware. Migratie is nuttig voor:

- Load balancing - guests can be moved to hosts with lower usage when a host becomes overloaded.
- Hardware failover - when hardware devices on the host start to fail, guests can be safely relocated so the host can be powered down and repaired.
- Energy saving - guests can be redistributed to other hosts and host systems powered off to save energy and cut costs in low usage periods.
- Geographic migration - guests can be moved to another location for lower latency or in serious circumstances.

Gedeelde, netwerk opslag wordt gebruikt voor het opslaan van guest images. Zonder gedeelde opslag is migratie niet mogelijk.

---

An offline migration suspends the guest then moves an image of the guests memory to the destination host. The guest is resumed on the destination host and the memory the guest used on the source host is freed.

De tijd die een off-line migratie kost hangt af van de netwerk bandbreedte en latentie. Een guest met 2GB geheugen zal meerdere seconden duren op een 1 Gbit Ethernet link.

Een live migratie laat de guest draaiende op de bron host en begint met het verplaatsen van het geheugen zonder de guest te stoppen. Alle veranderde geheugen pagina's worden bijgehouden en naar de bestemming gestuurd nadat de image verzonden is. Het geheugen wordt vernieuwd met de veranderde pagina's. Het proces gaat door totdat het een heuristiek bereikt, of het kopieert alle pagina's met succes, of de bron verandert te snel en de doel host kan geen vordering maken. Als de heuristiek bereikt wordt, wordt de guest kort gestopt op de bron host en de registers en buffers worden verstuurd. De registers worden geladen op de nieuwe host en de guest vervolgt dan op de doel host. Als de guest niet kan fuseren (wat gebeurt als guests onder zware belasting werken), wordt de guest gestopt en wordt een off-line migratie opgestart.

De tijd die een off-line migratie duurt hangt af van de netwerk bandbreedte en latentie zowel als van de activiteit op de guest. Als de guest significant veel I/O of CPU gebruikt zal de migratie veel langer duren.

MAC adressen

Het Media Access Control adres is het hardware adres van een netwerk interface controller. In de context van virtualisatie moeten MAC adressen aangemaakt worden voor virtuele netwerk interfaces waarbij iedere MAC op je locale systeem uniek moet zijn.

Para-virtualisatie

Para-virtualisatie gebruikt een speciale kernel, waarnaar soms wordt gerefereerd als de Xen kernel of het *kernel-xen* pakket. Para-gevirtualiseerde guest kernels worden tegelijkertijd op de host gedraaid met het gebruik van de bibliotheken en apparaten van de host. Een para-gevirtualiseerde installatie kan complete toegang hebben tot alle apparaten op het systeem wat beperkt kan worden met beveiliging instellingen (SELinux en bestand controles). Para-virtualisatie is sneller dan volledige virtualisatie. Para-virtualisatie kan effectief gebruikt worden voor werklust balanceren, levering, beveiliging en consolidatie voordelen.

Sinds Fedora 9 zal niet langer een speciale kernel nodig zijn. Zodra deze aanpassing wordt geaccepteerd in de hoofd Linux boom zullen alle kernels na die versie para-virtualisatie aangezet hebben of het zal beschikbaar zijn.

Para-gevirtualiseerd

Zie [Para-virtualisatie](#).

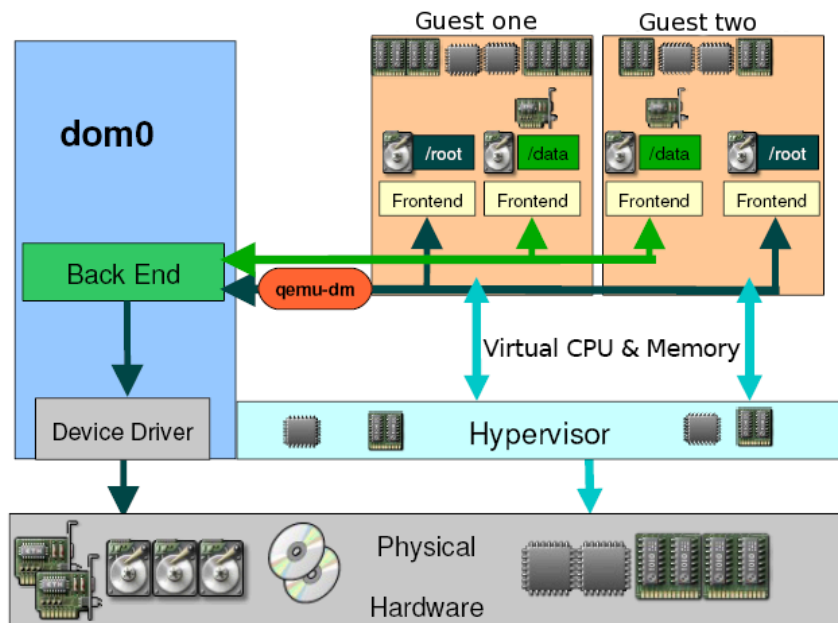
Para-gevirtualiseerde drivers

Para-gevirtualiseerde drivers zijn apparaat drivers die op volledig gevirtualiseerde Linuxguests werken. Deze drivers verhogen

	de prestaties van netwerk en blok apparaat i/O voor volledig gevirtualiseerde guests.
Security Enhanced Linux	SELinux, afkorting van Security Enhanced Linux, gebruikt Linux security modules (LSM) in de Linux kernel om een reeks van minimale rechten vereist voor beveiliging richtlijnen aan te bieden.
Universele unieke identificatie	Een Universele unieke identificatie (UUID) is een standaard methode voor het nummers geven aan apparaten, systemen en bepaalde software objecten in verspreide computer omgevingen. Types van UUID's in virtualisatie zijn: ext2 en ext3 bestandssysteem identifiers, RAID opstelling identifiers, iSCSI en LUN apparaat identifiers, MAC adressen en virtuele machine identifiers.
Virtualization	<p>Virtualisatie is een brede computer term voor het draaien van software, gewoonlijk besturingssystemen, tegelijkertijd en geïsoleerd van andere programma's op een systeem. De meeste bestaande implementaties van virtualisatie gebruiken een hypervisor, een software laag bovenop een besturingssysteem, om hardware te abstraheren. De hypervisor laat meerdere besturingssystemen draaien op hetzelfde fysieke systeem door het guest besturingssysteem gevirtualiseerde hardware te geven. Er zijn verscheidene methodes voor het virtualiseren van besturingssystemen:</p> <ul style="list-style-type: none"><li>• Hardware-ondersteunde virtualisatie is de techniek gebruikt voor volledige virtualisatie met Xen en KVM (definitie: <a href="#">Volledige virtualisatie</a>)</li><li>• Para-virtualisatie is een techniek gebruikt door Xen om Linux guests te draaien (definitie: <a href="#">Para-virtualisatie</a>)</li><li>• Software virtualisatie of emulatie. Software virtualisatie gebruikt binaire vertaling en andere emulatie technieken om onveranderde besturingssystemen te draaien. Software virtualisatie is belangrijk langzamer dan hardware-ondersteunde virtualisatie of para-virtualisatie.</li></ul>
Gevirtualiseerde CPU	Een systeem heeft een aantal virtuele CPU's (VCPU's) gerelateerd aan het aantal fysieke processor kernen. Het aantal virtuele CPU's is eindig en representeert het totale aantal virtuele CPU's die toegekend kunnen worden aan guest virtuele machines.
Virtuele machines	Een virtuele machine is een software implementatie van een fysieke machine of programmeertaal (bijvoorbeeld de Java Runtime omgeving of LISP). Virtuele machines in de context van virtualisatie zijn besturingssystemen die draaien op gevirtualiseerde hardware.
Xen	Fedora ondersteunt de Xen hypervisor en de KVM hypervisor (refereer naar <a href="#">Kernel-gebaseerde virtuele machine</a> ). Beide hypervisors hebben verschillende architecturen en ontwikkel aanpak. De Xen hypervisor draait in een Linux besturingssysteem welke optreedt als een host beheer systeem hulpbronnen en virtualisatie API's. Naar de host wordt soms gerefereerd als een <i>dom0</i> or Domein0.

# Xen Full Virtualization Architecture

With the para-virtualized drivers



# Xen Para-virtualization Architecture

