

Setting Up Horizon 6 for Linux Desktops

Horizon 6
Version 6.1.1

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-001796-00

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2015 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Setting Up Horizon 6 for Linux Desktops	5
1 Installing and Configuring Horizon 6 for Linux Desktops	7
Features of a Horizon 6 for Linux Desktop	7
System Requirements for Horizon 6 for Linux	8
Create a Virtual Machine and Install Linux	10
Prepare a Linux Guest Operating System for Remote Desktop Deployment	10
Install View Agent on a Linux Virtual Machine	12
Create a Desktop Pool That Contains Linux Virtual Machines	14
2 Configuring Linux Machines for vDGA	17
Configure RHEL 6.6 for vDGA	17
3 Bulk Deployment of Horizon 6 for Linux Desktops	23
Overview of Bulk Deployment of Linux Desktops	23
Create a Virtual Machine Template for Cloning Linux Desktop Machines	24
Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops	25
Sample PowerCLI Script to Clone Linux Desktop Machines	25
Sample PowerCLI Script to Install View Agent on Linux Desktops	29
Sample PowerCLI Script to Uninstall View Agent on Linux Desktops	32
Sample PowerCLI Script to Perform Operations on Linux Desktop Machines	35
4 Administering Horizon 6 for Linux Desktops	39
Uninstalling and Reinstalling Horizon 6 for Linux Machines	39
Configure Lossless PNG Images on Linux Desktops	42
Suppress the vSphere Console Display of a Linux Desktop	42
Configure a Left-handed Mouse on Ubuntu Desktops	42
How to Perform Power Operations on Linux Desktops from vSphere	42
Gather Information About Horizon 6 for Linux Software	43
Troubleshooting Horizon 6 for Linux	43
Index	49

Setting Up Horizon 6 for Linux Desktops

The *Setting Up Horizon 6 for Linux Desktops* document provides information about setting up a Linux virtual machine for use as a VMware Horizon 6™ desktop, including preparing the Linux guest operating system, installing View Agent on the virtual machine, and configuring the machine in View Administrator for use in a Horizon 6 deployment.

Intended Audience

This information is intended for anyone who wants to configure and use remote desktops that run on Linux guest operating systems. The information is written for experienced Linux system administrators who are familiar with virtual machine technology and datacenter operations.

Installing and Configuring Horizon 6 for Linux Desktops

1

To set up a Linux virtual machine as a remote desktop in a Horizon 6 environment, you must prepare the Linux guest operating system, install View Agent on the virtual machine, and configure the machine in View Administrator.

This chapter includes the following topics:

- [“Features of a Horizon 6 for Linux Desktop,”](#) on page 7
- [“System Requirements for Horizon 6 for Linux,”](#) on page 8
- [“Create a Virtual Machine and Install Linux,”](#) on page 10
- [“Prepare a Linux Guest Operating System for Remote Desktop Deployment,”](#) on page 10
- [“Install View Agent on a Linux Virtual Machine,”](#) on page 12
- [“Create a Desktop Pool That Contains Linux Virtual Machines,”](#) on page 14

Features of a Horizon 6 for Linux Desktop

After you set up a Linux guest operating system for use as a remote desktop, entitled users can launch VDI desktop sessions on the single-user Linux machine just as they do on a Windows machine.

Linux desktops provide features such as audio out, multiple monitors, and autofit.

Certain limitations apply to this release of View Agent for Linux:

- Single Sign-on (SSO) is not supported. After logging in to Horizon 6 and launching the remote desktop, the user must log in to the Linux guest operating system.
- Automated provisioning and other features that are provided only with automated desktop pools are not supported. For example, the refresh on logoff operation is not available.
- Local devices cannot be used on the remote desktop. For example, USB redirection, virtual printing, location-based printing, clipboard redirection, Real-Time Audio-Video, and smart cards are not supported.
- HTML Access is not supported.

NOTE When a security server is used, port 5443 must be open in the internal firewall to allow traffic between the security server and the Linux desktop.

System Requirements for Horizon 6 for Linux

Horizon 6 for Linux must meet certain operating system, Horizon 6, and vSphere platform requirements.

Supported Linux Operating Systems for View Agent

The following table lists the Linux operating systems that are supported on virtual machines in a desktop pool.

Table 1-1. Supported Linux Operating Systems for View Agent

Linux Distribution	Architecture
Ubuntu 12.04	x86 and x64
RHEL 6.6	x86 and x64
CentOS 6.6	x86 and x64
NeoKylin 6, NeoKylin 6 Update 1	x86 and x64

NOTE NeoKylin is supported on a Tech Preview basis.

Other Linux distributions have not been certified to support View Agent, but the View Agent software does not prevent you from using them. You are welcome to try out uncertified distributions. For example, the distributions RHEL 6.5, CentOS 6.5, RHEL 7, Ubuntu 14.04, and UbuntuKylin 14.04 are likely to work.

However, uncertified distributions might not function as fully as supported ones, and VMware cannot guarantee that problems in uncertified distributions will be resolved. For example, performance on Ubuntu 14.04 is poor unless Compiz is disabled.

Required Platform and Horizon 6 Software Versions

To install and use Horizon 6 for Linux, your deployment must meet certain vSphere platform, Horizon 6, and client requirements.

vSphere platform version	vSphere 5.5 U2, vSphere 6.0, or a later release vSphere 6.0 or a later release is required to support virtual machines that use vDGA on NVIDIA GRID graphics cards.
Horizon environment	Horizon 6 version 6.1.1 or a later release
Horizon Client software	Horizon Client 3.4 for Windows, Linux, or Mac OS X Zero clients and mobile clients are not supported

Recommended Video Memory (vRAM) Settings

When you create a Linux virtual machine in vSphere Client, configure the vRAM size as shown in the following table. Set the vRAM size that is recommended for the number and resolution of the monitors that you configure for the virtual machine.

These vRAM sizes are the minimum recommendations. If more resources are available on the virtual machine, set the vRAM to larger values for improved video performance.



CAUTION Horizon 6 does not automatically configure the vRAM settings on Linux virtual machines, as happens on Windows virtual machines. You must configure the vRAM settings manually in vSphere Client.

Table 1-2. Recommended vRAM Settings for Linux Guest Operating Systems

vRAM Size	Number of Monitors	Maximum Resolution
10 MB	1	1600x1200 or 1680x1050
12 MB	1	1920x1440
16 MB	1	2560x1600
32 MB	2	2048x1536 or 2560x1600
48 MB	3	2048x1536
64 MB	3	2560x1600
64 MB	4	2048x1536
128 MB	4	2560x1600

RHEL and CentOS only support this configuration on vSphere 5.5.
To support this configuration on Ubuntu, you must recompile the kernel.
For NeoKylin, this configuration is not supported.

NOTE To connect to a RHEL 6.6 or CentOS 6.6 desktop with multiple monitors, you must specify the number of displays correctly. For information about how to specify the number of displays, see [“Create a Virtual Machine and Install Linux,”](#) on page 10. You must also edit the vmx file and append the following lines:

```
svga.maxWidth="10240"
svga.maxHeight="2048"
```

If these settings are not added, only one monitor displays the desktop. The others display a black screen.

If you encounter an autofit issue with the recommended settings, you can specify a larger vRAM size. vSphere Client permits a maximum vRAM size of 128 MB. If your specified size exceeds 128 MB, you must modify the vmx file manually. The following example specifies a vRAM size of 256 MB:

```
svga.vramSize = "268435456"
```

Recommended vCPU and Shared Memory Settings to Support Multiple Monitors

To improve desktop performance with multiple monitors, configure at least two vCPUs for a Linux virtual machine.

Also make sure that the virtual machine has adequate shared memory to support multiple monitors. Determine the current maximum shared memory size (`shmmx`) with the following command:

```
sysctl -a | grep shm
```

If shared memory is small, increase the maximum size with the following command:

```
sysctl -w "kernel.shmmx=65536000"
```

Recommended vCPU Settings for Video Playback

For a Linux virtual machine that is not using vDGA graphics, video playback on high-resolution monitors might be uneven if too few vCPUs are configured. Configure additional vCPUs, such as four vCPUs, to improve the performance of video playback.

Create a Virtual Machine and Install Linux

You create a new virtual machine in vCenter Server for each remote desktop that is deployed in Horizon 6. You must install your Linux distribution on the virtual machine.

Prerequisites

- Verify that your deployment meets the requirements for supporting Linux desktops. See [“System Requirements for Horizon 6 for Linux,”](#) on page 8.
- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing guest operating systems. See "Creating and Preparing Virtual Machines" in the *Setting Up Desktop and Application Pools in View* document.
- Familiarize yourself with the recommended video memory (vRAM) values for the monitors you will use with the virtual machine. See [“System Requirements for Horizon 6 for Linux,”](#) on page 8.

Procedure

- 1 In vSphere Web Client or vSphere Client, create a new virtual machine.
- 2 Configure custom configuration options.
 - a Right-click the virtual machine and click **Edit Settings**.
 - b Specify the number of vCPUs and the vMemory size.

For recommended values, follow the guidelines in the installation guide for your Linux distribution.

For example, Ubuntu 12.04 recommends configuring 2048 MB for vMemory and 2 vCPUs.
 - c Select **Video card** and specify the number of displays and the total video memory (vRAM).

For recommended values, follow the guidelines in *System Requirements for Horizon 6 for Linux*. Do not use the Video Memory Calculator.
- 3 Power on the virtual machine and install the Linux distribution.
- 4 Configure the virtual machine as a gnome desktop environment.

KDE has not been certified to support View Agent, although basic connections and the use of audio and video work well on some distributions such as Kubuntu.
- 5 Ensure that the system hostname is resolvable.

Prepare a Linux Guest Operating System for Remote Desktop Deployment

You must perform certain tasks to prepare a Linux guest operating system for use as a desktop in a Horizon 6 deployment.

Before a Linux guest operating system can be managed by Horizon 6, the machine must be able to communicate with the other machines in the environment. You must configure the Linux machine to be resolvable through DNS. Otherwise, View Agent cannot be registered as a machine source and the machine cannot be added to a desktop pool.

Prerequisites

- Verify that a new virtual machine was created in vCenter Server and your Linux distribution was installed on the machine

- Familiarize yourself with the steps for mounting and installing VMware Tools on a Linux virtual machine. See "Manually Install or Upgrade VMware Tools in a Linux Virtual Machine" in the *vSphere Virtual Machine Administration* document.
- Familiarize yourself with the steps for configuring your Linux machine to be resolvable through DNS. These steps vary for the different Linux distributions and releases. For instructions, consult the documentation for your Linux distribution and release.
- Familiarize yourself with the required JRE version for your Linux distribution. See "[Required JRE Versions for Linux Guest Operating Systems](#)," on page 12.

Procedure

- 1 In vSphere Web Client or vSphere Client, mount the VMware Tools virtual disk on the guest operating system.
- 2 Right-click the VMware Tools installer file, `VMwareTools.x.x.x-xxxx.tar.gz`, click **Extract to**, and select the desktop for your Linux distribution.

The `vmware-tools-distrib` folder is extracted to the desktop.

- 3 In the virtual machine, log in to the guest operating system and root and open a terminal window.
- 4 Uncompress the VMware Tools tar installer file.

For example:

```
tar xzpf /mnt/cdrom/VMwareTools-x.x.x-yyyy.tar.gz
```

- 5 Run the installer and configure VMware Tools.

The command might vary slightly in different Linux distributions. For example:

```
cd vmware-tools-distrib
sudo ./vmware-install.pl -d
```

Usually, the `vmware-config-tools.pl` configuration file runs after the installer file finishes running.

- 6 Configure the Linux machine to be resolvable through DNS.
- 7 Ensure that the View Connection Server instances in the pod can be resolved through DNS.
- 8 If your deployment uses a Windows Server 2012 or Windows Server 2012 R2 Active Directory (AD) server, enable reversible password encryption.
 - a On the AD server, open AD users and computers.
 - b Right-click the View Admin account, or create a new Linux View Agent account, and select **Properties**.
 - c Select the Account tab.
 - d In Account options, select the **Store password using reversible encryption** setting.
 - e Reset the View Admin password.

- 9 If the Linux virtual machine is not connected to the Internet, manually download and install the required JRE version before you install View Agent.

You do not have to take this step if the machine is connected to the Internet. In this case, the View Agent installer automatically downloads the JRE to the machine.

If necessary, take the following steps to install the JRE:

Linux Distribution	Description
Ubuntu 12.04	Open a terminal window and run the following commands. a <code>sudo add-apt-repository -y ppa:webupd8team/java</code> b <code>sudo apt-get update</code> c <code>sudo apt-get -y install oracle-java7-installer</code>
RHEL 6.6 CentOS 6.6 NeoKylin 6	a Download <code>jre-7u79-linux-i586.rpm</code> or <code>jre-7u79-linux-x64.rpm</code> from http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html b Install the rpm: <code>sudo rpm -i jre-7u79-linux-i586.rpm</code> or <code>sudo rpm -i jre-7u79-linux-x64.rpm</code>

After installing VMware Tools, if you upgrade the Linux kernel, VMware Tools might stop running. To resolve the problem, see <http://kb.vmware.com/kb/2050592>.

Required JRE Versions for Linux Guest Operating Systems

View Agent requires certain JRE versions to be installed on Linux guest operating systems. When you install View Agent on a Linux virtual machine, the installer downloads the required JRE if it is not already present on the machine.

If the Linux virtual machine is not connected to the Internet, you must manually download the required JRE before you install View Agent.

View Agent for Linux supports JRE version 1.7.x. Version 1.8 is not supported.

IMPORTANT Make sure that the JRE version is patched to include the latest security updates. You must use Update 75 or later to address the SKIP-TLS security vulnerabilities.

Table 1-3. JRE Versions Required by View Agent on Linux Distributions

Linux Distribution	JRE Version
Ubuntu 12.04	Oracle Java 1.7.0_80
RHEL 6.6	Oracle jre-7u79 (1.7.0_79)
CentOS 6.6	Oracle jre-7u79 (1.7.0_79)
Neokylin	Oracle jre-7u79 (1.7.0_79)

Install View Agent on a Linux Virtual Machine

You must install View Agent on a Linux virtual machine before you can deploy the machine as a remote desktop.

Prerequisites

- Verify that the Linux guest operating system is prepared for desktop use. See “[Prepare a Linux Guest Operating System for Remote Desktop Deployment](#),” on page 10.
- Familiarize yourself with the View Agent installer script for Linux. See “[install_viewagent.sh Command Usage and Options](#),” on page 14.

- Verify that the View Connection Server administrative user that you provide with the installation command has the **Agent Registration Administrators** or **Administrators** role in View Administrator. Specifically, the **Register Agent** privilege is required to register View Agent with View Connection Server. **Agent Registration Administrators** is a restricted role that provides this minimum privilege.

Procedure

- 1 Download the View Agent for Linux installer file from the VMware download site at <https://my.vmware.com/web/vmware/downloads>.

Under Desktop & End-User Computing, select the VMware Horizon 6 download, which includes the View Agent for Linux installer.

The installer filename is `VMware-viewagent-linux-x86_64-y.y.y-xxxxxxx.tar` for 64-bit Linux or `VMware-viewagent-linux-y.y.y-xxxxxxx.tar` for 34-bit Linux, where `y.y.y` is the version number and `xxxxxxx` is the build number.

- 2 Unpack the tarball for your Linux distribution on the guest operating system.

For example:

```
tar -xvf <View Agent tar ball>
```

- 3 Navigate to the tar ball folder.
- 4 Run the `install_viewagent.sh` script to install View Agent.

For example:

```
sudo ./install_viewagent.sh -b mybroker.mydomain.com -d mydomain.com -u administrator -p password
```

The `-b` parameter specifies the View Connection Server instance to which the Linux machine is registered. You can use an FQDN or IP address with the `-b` parameter.

The `-d`, `-u`, and `-p` parameters specify the domain, user name, and password of the View administrator user. In the password, be sure to escape special characters such as `$`. For example: `ab\$cdef`

If you do not type the `-p` parameter with the command, you are prompted to provide the password after you enter the command. When you type the password at the prompt, the text is hidden.

By default, the hostname of the Linux machine is used to register the machine. You can specify a different machine name with the `-n` parameter.

- 5 Restart or log out of the guest operating system.

This step ensures that the graphics UI changes configured by View Agent take effect on the machine.

The `install_viewagent.sh` script installs the View Agent software on the Linux virtual machine. The script registers the machine with View Connection Server.

The `viewagent` service is started on the Linux virtual machine. You can verify that the service is started by running the `#service viewagent status` command.

If an Agent Registration Failed message appears, see [“Troubleshooting View Agent Registration Failure for a Linux Machine,”](#) on page 44.

If another View Agent issue occurs, see [“Troubleshooting Horizon 6 for Linux,”](#) on page 43.

install_viewagent.sh Command Usage and Options

The `install_viewagent.sh` script installs View Agent on a Linux guest operating system.

Use the following form of the `install_viewagent.sh` script in a command window in the gnome desktop environment.

```
install_viewagent.sh command_option argument [command_option argument] . . .
```

The `install_viewagent.sh` script includes mandatory and optional parameters.

Table 1-4. `install_viewagent.sh` Mandatory Parameters

Mandatory Parameter	Description
-b	FQDN or IP address of the View Connection Server instance. For example: <code>mybroker.mydomain.com</code> or <code>192.0.2.10</code>
-d	Domain name of the View Connection Server instance administrator user. For example: <code>mydomain.com</code>
-u	User name of the View Connection Server administrator user. For example: <code>myadmin</code> The user must have the Agent Registration Administrators or Administrators role in View Administrator. Specifically, the Register Agent privilege is required to register View Agent with View Connection Server. Agent Registration Administrators is a restricted role that provides this minimum privilege.

Table 1-5. `install_viewagent.sh` Password Parameter

Optional Parameter (Required Information)	Description
-p	Password for the View Connection Server administrator user. You do not have to type the <code>-p</code> parameter at the command line if you do not want to expose the password in text. If you type the <code>install_viewagent.sh</code> command without the <code>-p</code> parameter, you are prompted to enter the password. When you type the password at the prompt, the text is hidden. You must provide the password, either with the <code>-p</code> parameter or at the prompt. If the password contains a special character such as <code>\$</code> within the shell in which the installer is executed, make sure the special character is escaped. For example: <code>ab\\$cdef</code>

Table 1-6. `install_viewagent.sh` Optional Parameter

Optional Parameter	Description
-n	Machine name that is registered to View Connection Server. By default, the View Agent installer uses the host name. You can specify your own name.

Create a Desktop Pool That Contains Linux Virtual Machines

To configure Linux virtual machines for use as remote desktops, you create a manual desktop pool and add the Linux machines to the pool.

When you create the desktop pool, add only Linux virtual machines to the pool. If the pool contains both Windows and Linux guest operating systems, the pool is treated as a Windows pool, and you will be unable to connect to the Linux desktops.

When you entitle users to Linux machines in the desktop pool, as a best practice, make sure that the users do not have administrative privileges in the Linux guest operating system. An admin user in Linux can open a terminal window and invoke commands such as `shutdown`, which powers off the virtual machine. The vCenter Server administrator must power on the machine again. Entitling non-admin Linux users ensures that you do not have to manage these power operations manually.

Prerequisites

- Verify that View Agent is installed on the Linux guest operating systems. See [“Install View Agent on a Linux Virtual Machine,”](#) on page 12.
- Verify that the Linux virtual machines are registered in View Connection Server. In View Administrator, select **View Configuration > Registered Machines** and select the **Others** tab. Verify that each machine's state is Available.

Procedure

- 1 In View Administrator, add a manual desktop pool.
Select **Catalog > Desktop Pools > Add** .
- 2 Select **Manual Desktop Pool**.
- 3 Select either dedicated or floating user assignments for the machines in the desktop pool.
- 4 On the Machine Source page, select **Other Sources**.
- 5 On the Add Machines page, select the Linux virtual machines that you configured and complete the Add Desktop Pool wizard.

Do not change the Remote Display Protocol settings. These settings have no effect on Linux desktops. Also, the end user cannot choose the display protocol.

IMPORTANT Add Linux virtual machines only. If you add Windows virtual machines, the Linux desktops in the pool will be unavailable.

- 6 Entitle users to the machines in the desktop pool.
In View Administrator, select the desktop pool, select **Entitlements > Add entitlement**, and add users or groups.
As a best practice, make sure that the users do not have administrative privileges in the Linux guest operating system.

The Linux virtual machines are ready to be used as remote desktops in a Horizon 6 deployment.

Configuring Linux Machines for vDGA

2

You can configure Horizon 6 for Linux virtual machines to use Virtual Dedicated Graphics Acceleration (vDGA) on NVIDIA GRID graphics cards that are installed on ESXi hosts.

Shared GPU hardware acceleration (NVIDIA GRID vGPU) is not supported in Horizon 6 for Linux.

The Autofit feature and multiple monitors are supported on Linux desktops that are configured to use vDGA.

The steps to configure a Linux guest operating system for vDGA vary depending on the Linux distribution.

Configure RHEL 6.6 for vDGA

You can set up an RHEL 6.6 guest operating system so that a Horizon 6 for Linux desktop can take advantage of vDGA capabilities on the ESXi host.

Procedure

- 1 [Enable DirectPath I/O for NVIDIA GRID GPUs on a Host](#) on page 17
Before you configure a Linux virtual machine to use vDGA, you must make the NVIDIA GRID GPU PCI devices available for DirectPath I/O passthrough on the ESXi host.
- 2 [Add a vDGA Pass-Through Device to a RHEL 6.6 Virtual Machine](#) on page 18
To configure a RHEL 6.6 virtual machine to use vDGA, you must add the PCI device to the virtual machine. With this step, the physical device on the ESXi host can be passed through for use on the virtual machine.
- 3 [Install an NVIDIA GRID Driver on RHEL 6.6](#) on page 19
To install an NVIDIA GRID driver on a RHEL 6.6 virtual machine, you must disable the default NVIDIA driver, download the NVIDIA GRID drivers, and configure the PCI device on the virtual machine.

Enable DirectPath I/O for NVIDIA GRID GPUs on a Host

Before you configure a Linux virtual machine to use vDGA, you must make the NVIDIA GRID GPU PCI devices available for DirectPath I/O passthrough on the ESXi host.

Prerequisites

- Verify that vSphere 6.0 or a later release is installed in your environment.
- Verify that the NVIDIA GRID K1 or K2 graphics cards are installed on the ESXi host.

Procedure

- 1 In the vSphere Web Client, browse to the ESXi host.

- 2 Click the **Manage** tab and click **Settings**.
- 3 In the Hardware section, click **PCI Devices**.
- 4 To enable DirectPath I/O passthrough for the NVIDIA GRID GPUs, click **Edit**.

Icon	Description
Green icon	The PCI device is active and can be enabled.
Orange icon	The state of the device has changed. You must reboot the host before you can use the device.

- 5 Select the NVIDIA GRID GPUs and click **OK**.
The PCI devices are added to the table, DirectPath I/O PCI Devices Available to VMs.
- 6 Reboot the host to make the PCI devices available for use by the Linux virtual machines.

Add a vDGA Pass-Through Device to a RHEL 6.6 Virtual Machine

To configure a RHEL 6.6 virtual machine to use vDGA, you must add the PCI device to the virtual machine. With this step, the physical device on the ESXi host can be passed through for use on the virtual machine.

Prerequisites

- Verify that the Linux virtual machine is prepared for use as a desktop, View Agent is installed, and the machine is deployed in a desktop pool. See [Chapter 1, “Installing and Configuring Horizon 6 for Linux Desktops,”](#) on page 7.
- Verify that the NVIDIA GRID GPU PCI device was made available for DirectPath I/O pass-through on the host. See [“Enable DirectPath I/O for NVIDIA GRID GPUs on a Host,”](#) on page 17.

Procedure

- 1 Power off the virtual machine. and log in to the RHEL 6.6 guest operating system as a local user configured with sudo rights.
- 2 In vSphere Web Client, select the virtual machine and, under the **VM Hardware** tab, click **Edit Settings**.
- 3 In the **New device** menu, select **PCI Device**.

Note that the **Shared PCI Device** option is not supported on Linux desktops.

- 4 Click **Add** and select the PCI device from the drop-down menu.
- 5 Click **Reserve all memory** and click **OK**.
You must reserve all virtual machine memory to enable the GPU to support vDGA.
- 6 Power on the virtual machine and open vSphere console to connect to the machine.
- 7 Verify that the NVIDIA GRID device is passed through to the virtual machine.

Open a terminal window and run the following command:

```
lspci | grep NVIDIA
```

The XX:00.0 VGA-compatible controller is displayed. For example:

```
NVIDIA Corporation GK104GL [GRID K2]
```

Install an NVIDIA GRID Driver on RHEL 6.6

To install an NVIDIA GRID driver on a RHEL 6.6 virtual machine, you must disable the default NVIDIA driver, download the NVIDIA GRID drivers, and configure the PCI device on the virtual machine.

Prerequisites

- Verify that the PCI device was added to the RHEL 6.6 virtual machine. See [“Add a vDGA Pass-Through Device to a RHEL 6.6 Virtual Machine,”](#) on page 18.

Procedure

- 1 Disable and blacklist the default NVIDIA Nouveau driver.

You must take this step before you can install the NVIDIA GRID driver.

- a Edit the `grub.conf` file.

```
sudo vi /boot/grub/grub.conf
```

- b Add the `rdblacklist=nouveau` line at the end of the kernel options.

- c Edit the `blacklist.conf` file.

```
sudo vi /etc/modprobe.d/blacklist.conf
```

- d Add the following line anywhere in the `blacklist.conf` file.

```
blacklist nouveau
```

- 2 Restart the virtual machine.

The display has a changed look and feel.

- 3 (Optional) Verify that the Nouveau driver is disabled.

```
/sbin/lsmmod | grep nouveau
```

If the `grep` search does not return any results, the Nouveau driver is disabled.

- 4 Download the NVIDIA drivers from the [NVIDIA Driver Downloads](#) site.

Select the appropriate driver version from the drop-down menus.

Option	Description
Product Type	GRID
Product Series	GRID Series
Product	Select the version (such as GRID K2) that is installed on the ESXi host.
Operating System	Linux 64-bit or Linux 32-bit

- 5 Open a remote terminal to the virtual machine, or switch to a text console by typing `Ctrl-Alt-F2`, log in as root, and run the `init 3` command to disable X Windows.

- 6 Install additional components that are required for the NVIDIA driver.

```
sudo yum install gcc-c++
sudo yum install kernel-devel
sudo yum install kernel-headers
```

- 7 Add an executable flag to the NVIDIA driver package.

```
chmod +x NVIDIA-Linux-x86_64-346.47.run
```

- 8 Determine the kernel version that is running.

```
uname -r
```

The kernel version is displayed. For example:

```
2.6.<kernel version>
```

- 9 Start the NVIDIA installer with a flag pointing to the kernel source location.

For example:

```
sudo ./NVIDIA-Linux-x86_64-346.47.run --kernel-source-  
path=/usr/src/kernels/2.6.<kernel version>
```

- 10 Accept the NVIDIA software license agreement and select **Yes** to automatically update the X configuration settings.

- 11 Run the `lspci | grep NVIDIA` command to identify the PCI device ID of the NVIDIA GRID graphics card.

The terminal window displays the PCI device ID.

For example:

```
13:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K2]
```

The 13:00.0 is the PCI device ID. These digits are displayed in hex.

IMPORTANT In Step 14, you must convert this value from hex to decimal.

- 12 Link the `edid.bin` file to the `/etc/X11` directory.

```
sudo ln -s /usr/lib/vmware/viewagent/resources/X11/edid.bin /etc/X11/edid.bin
```

- 13 Edit the `xorg.conf` file to configure the PCI device ID.

```
sudo nano /etc/X11/xorg.conf
```

- 14 Add the `BusID` and `Option` lines to the Section "Device" section for the driver "NVIDIA".

Add these lines:

```
Section "Device"
    Identifier      "Device0"
    Driver          "nvidia"
    VendorName     "NVIDIA Corporation"
    BusID          "PCI:X:X:X"
    Option         "IgnoreDisplayDevices" "CRT-0"
    Option         "ConnectedMonitor" "DFP-0, DFP-1, DFP-2, DFP-3"
    Option         "CustomEDID" "DFP-0:/etc/X11/edid.bin; DFP-1:/etc/X11/edid.bin;  
DFP-2:/etc/X11/edid.bin; DFP-3:/etc/X11/edid.bin"
    Option         "ModeValidation" "DFP-0: NoVesaModes,NoPredefinedModes,  
NoMaxSizeCheck,NoEDIDModes,NoXserverModes,NoEdidMaxPclkCheck,NoHorizSyncCheck,  
AllowNonEdidModes; DFP-1: NoVesaModes,NoPredefinedModes,NoMaxSizeCheck,NoEDIDModes,  
NoXserverModes,NoEdidMaxPclkCheck,NoHorizSyncCheck,AllowNonEdidModes; DFP-2: NoVesaModes,  
NoPredefinedModes,NoMaxSizeCheck,NoEDIDModes,NoXserverModes,NoEdidMaxPclkCheck,  
NoHorizSyncCheck,AllowNonEdidModes; DFP-3: NoVesaModes,NoPredefinedModes,NoMaxSizeCheck,  
NoEDIDModes,NoXserverModes,NoEdidMaxPclkCheck,NoHorizSyncCheck,AllowNonEdidModes"
EndSection
```

In the `BusID` line, the `X:X:X` digits are the PCI device ID, as displayed by the `lspci` command and converted from hex to decimal.

For example, the `lspci` command might display the PCI device ID as 13:00.0, which is in hex.

For the BusID value, convert the hex to decimal:

```
BusID "PCI:19:0:0"
```

The `Option` lines are required to support the use of multiple monitors with vDGA.

Use colons (`:`) as separators in the `xorg.conf` file.

- 15 Restart the virtual machine.

The View Agent startup script initializes the X server and display topology.

You can no longer view the virtual machine display in the vSphere console.

- 16 From Horizon Client, connect to the Linux desktop.

- 17 In the Linux desktop session, verify that the NVIDIA GRID driver is installed.

Open a terminal window and run the `glxinfo | grep NVIDIA` command.

The NVIDIA driver output is displayed. For example:

```
[root]# glxinfo | grep NVIDIA
server glx vendor string: NVIDIA Corporation
client glx vendor string: NVIDIA Corporation
OpenGL vendor string: NVIDIA Corporation
OpenGL version string: 4.5.0 NVIDIA 346.47
OpenGL shading language version string: 4.50 NVIDIA
```

The user can access the NVIDIA GRID graphics capabilities on the remote desktop.

After installing the NVIDIA driver, if you upgrade the Linux kernel, View Agent might not be able to communicate with View Connection Server. To resolve the problem, reinstall the NVIDIA driver.

Bulk Deployment of Horizon 6 for Linux Desktops

3

With View Administrator, you can create a pool of Windows, but not Linux, desktop machines automatically. With vSphere PowerCLI, you can develop scripts that automate the deployment of a pool of Linux desktop machines.

The sample scripts that are provided are for illustration purposes only. VMware does not accept responsibility for issues that might arise when you use and enhance the sample scripts.

This chapter includes the following topics:

- [“Overview of Bulk Deployment of Linux Desktops,”](#) on page 23
- [“Create a Virtual Machine Template for Cloning Linux Desktop Machines,”](#) on page 24
- [“Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops,”](#) on page 25
- [“Sample PowerCLI Script to Clone Linux Desktop Machines,”](#) on page 25
- [“Sample PowerCLI Script to Install View Agent on Linux Desktops,”](#) on page 29
- [“Sample PowerCLI Script to Uninstall View Agent on Linux Desktops,”](#) on page 32
- [“Sample PowerCLI Script to Perform Operations on Linux Desktop Machines,”](#) on page 35

Overview of Bulk Deployment of Linux Desktops

Deploying Linux desktops involve several steps. If you plan to deploy more than a few desktops, you can automate some of the steps by using PowerCLI scripts.

For more information about vSphere PowerCLI, see <https://www.vmware.com/support/developer/PowerCLI>.

The process of bulk deploying a pool of Linux desktops involves the following steps:

- 1 Create a virtual machine template.

See [“Create a Virtual Machine Template for Cloning Linux Desktop Machines,”](#) on page 24.

- 2 Create a guest customization specification.

See "Create a Customization Specification for Linux in the vSphere Web Client" in the *vSphere Virtual Machine Administration* document. When you create the specification, make sure that you specify the following settings correctly.

Setting	Value
Target Virtual Machine OS	Linux
Computer Name	Use the virtual machine name.

Setting	Value
Network Settings	Use standard network settings.
Primary DNS	Specify a valid address.

- 3 Clone virtual machines.

For a sample script, see [“Sample PowerCLI Script to Clone Linux Desktop Machines,”](#) on page 25.

- 4 Install View Agent in virtual machines.

For a sample script, see [“Sample PowerCLI Script to Install View Agent on Linux Desktops,”](#) on page 29.

- 5 Create a desktop pool.

See [“Create a Desktop Pool That Contains Linux Virtual Machines,”](#) on page 14.

Occasionally, you might need to uninstall View Agent on the Linux virtual machines. For example, you want to install a different version of View agent. For a sample script to uninstall view agent on virtual machines, see [“Sample PowerCLI Script to Uninstall View Agent on Linux Desktops,”](#) on page 32.

For a sample script that performs operations such as power-on or power-off on virtual machines, see [“Sample PowerCLI Script to Perform Operations on Linux Desktop Machines,”](#) on page 35.

Create a Virtual Machine Template for Cloning Linux Desktop Machines

Before you perform virtual machine cloning, you must create a virtual machine template that the clones are based on.

Prerequisites

- Verify that your deployment meets the requirements for supporting Linux desktops. See [“System Requirements for Horizon 6 for Linux,”](#) on page 8.
- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing guest operating systems. See "Creating and Preparing Virtual Machines" in the *Setting Up Desktop and Application Pools in View* document.
- Familiarize yourself with the recommended video memory (vRAM) values for the monitors you will use with the virtual machine. See [“System Requirements for Horizon 6 for Linux,”](#) on page 8.
- Familiarize yourself with the supported JRE versions. See [“System Requirements for Horizon 6 for Linux,”](#) on page 8.

Procedure

- 1 In vSphere Web Client or vSphere Client, create a new virtual machine.
- 2 Configure custom configuration options.

- a Right-click the virtual machine and click **Edit Settings**.

- b Specify the number of vCPUs and the vMemory size.

For recommended values, follow the guidelines in the installation guide for your Linux distribution.

For example, Ubuntu 12.04 recommends configuring 2048 MB for vMemory and 2 vCPUs.

- c Select **Video card** and specify the number of displays and the total video memory (vRAM).

For recommended values, follow the guidelines in *System Requirements for Horizon 6 for Linux*. Do not use the Video Memory Calculator.

- 3 Power on the virtual machine and install the Linux distribution.
- 4 Create a user with root privileges, for example, ViewUser. This user is used to install and uninstall View Agent only.
- 5 Edit `/etc/sudoers` and add the line `ViewUser ALL=(ALL) NOPASSWD:ALL`.
With this line in `/etc/sudoers`, no password is required to run `sudo` as ViewUser. When you run the sample script to install View Agent that is provided in this chapter, you specify ViewUser as an input.
- 6 If the Linux distribution is RHEL, CentOS, or NeoKylin, edit `/etc/sudoers` and comment out the lines `Defaults requiretty` and `Defaults !visiblepw`.
- 7 Install VMware Tools.
- 8 Install a supported version of JRE.
- 9 Shut down the virtual machine and create a snapshot.

Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops

The sample PowerCLI scripts to deploy Linux desktops read one input file that contains information about the desktop machines.

The input file is of type `csv` and contains the following information:

- Desktop virtual machine name
- Parent virtual machine name
- Guest customization specification
- Datastore where the cloned desktop machine resides
- ESXi server that hosts the desktop machine
- Parent virtual machine's snapshot that is used for cloning
- Flag that indicates whether to delete the desktop virtual machine if it exists

The following example shows what the input file may contain.

```
VMName,Parentvm,CustomSpec,Datastore,Host,FromSnapshot,DeleteIfPresent
linux-001,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-002,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-003,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-004,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-005,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
```

The sample scripts assume that the name of this input file is `CloneVMs.csv` and that the file is located in the same folder as the scripts.

Sample PowerCLI Script to Clone Linux Desktop Machines

You can customize and use the following sample script to clone any number of virtual machines.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 6 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in “[Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops](#),” on page 25. This script also interactively asks for the following information:

- IP address of the vCenter Server

- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Clone type, which can be linked or full
- Whether to disable vSphere VM console

Script Content

```
<#
.SYNOPSIS
Create Clones from a Master VM

.DESCRIPTION
The Tool supports creation of Full clone and linked clone from Master VM
The parent VM is required for the linked-clone to work and the parent VMs file cannot be renamed
or Moved

.NOTES
#>

#----- Functions -----

function IsVMExists ($VMExists)
{
    Write-Host "Checking if the VM $VMExists already Exists"
    [bool]$Exists = $false

    #Get all VMS and check if the VMs is already present in VC
    $listvm = Get-vm
    foreach ($lvm in $listvm)
    {
        if($VMExists -eq $lvm.Name )
        {
            $Exists = $true
        }
    }
    return $Exists
}

function Disable_VM_Console($VMToDisableConsole)
{
    $vmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec
    $extra = New-Object VMware.Vim.optionvalue
    $extra.Key="RemoteDisplay.maxConnections"
    $extra.Value=""
    $vmConfigSpec.extraconfig += $extra
    $vm = Get-VM $VMToDisableConsole | Get-View
    $vm.ReconfigVM($vmConfigSpec)
}

function Delete_VM($VMToDelete)
{
    Write-Host "Deleting VM $VMToDelete"
    Get-VM $VMToDelete | where { $_.PowerState -eq "PoweredOn" } | Stop-VM -confirm:$false
    Get-VM $VMToDelete | Remove-VM -DeleteFromDisk -confirm:$false
}
```

```

}

#----- Main Script -----

SvcAddress = Read-Host 'Your vCenter address'
SvcAdmin = Read-Host 'Your vCenter admin user name'
SvcPassword = Read-Host 'Your vCenter admin user password' -AsSecureString
SvcPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($SvcPassword))
$cloneType = Read-Host 'Clone Type ("linked" or "full") '
$disableVMConsole = Read-Host 'Disable vSphere VM Console ("yes" or "no", recommend "yes") '
$csvFile = '.\CloneVMs.csv'

# Check that user passed only linked or full clone
if (($cloneType.length >0) -and ($cloneType -ne "linked" -or $cloneType -ne "full"))
{
    write-host -ForegroundColor Red "Clone type supports only 'linked' or 'full' (case sensitive)"
    exit
}
if (($disableVMConsole.length >0) -and ($disableVMConsole -ne "yes" -or $disableVMConsole -ne "no"))
{
    write-host -ForegroundColor Red "Disable vSphere VM Console supports only 'yes' or 'no' (case sensitive)"
    exit
}

#check if file exists
if (!(Test-Path $csvFile))
{
    write-host -ForegroundColor Red "CSV File $CSVFile not found"
    exit
}

# Connect to the VC (Parameterize VC)
#Write-Host "DisConnecting Previous Connections"
#Disconnect-VIServer * -Confirm:$false -ErrorAction SilentlyContinue

Write-Host "Connecting to VC $vcAddress"
Connect-VIServer $vcAddress -ErrorAction Stop -user $vcAdmin -password $vcPassword

#Read input CSV file
$csvData = Import-CSV $csvFile
#$csvData = Import-CSV $csvFile -
header("VMName","Parentvm","CustomSpec","Datastore","Host","FromSnapshot","DeleteIfPresent")
foreach ($line in $csvData)
{
    $destVMName=$line.VMName
    $srcVM = $line.Parentvm
    $cSpec = $line.CustomSpec
    $targetDSName = $line.Datastore
    $destHost = $line.Host
    $srcSnapshot = $line.FromSnapshot

```

```

$deleteExisting = $line.DeleteIfPresent
if (IsVMExists ($destVMName))
{
    Write-Host "VM $destVMName Already Exists in VC $vcAddress"
    if($deleteExisting -eq "TRUE")
    {
        Delete_VM ($destVMName)
    }
    else
    {
        Write-Host "Skip clone for $destVMName"
        continue
    }
}
$vm = get-vm $srcvm -ErrorAction Stop | get-view -ErrorAction Stop
$cloneSpec = new-object VMware.VIM.VirtualMachineCloneSpec
$cloneSpec.Location = new-object VMware.VIM.VirtualMachineRelocateSpec
if ($CloneType -eq "linked")
{
    $cloneSpec.Location.DiskMoveType =
[VMware.VIM.VirtualMachineRelocateDiskMoveOptions]::createNewChildDiskBacking
}
Write-Host "Using Datastore $targetDSName"
$newDS = Get-Datastore $targetDSName | Get-View
$cloneSpec.Location.Datastore = $newDS.summary.Datastore
Set-VM -vm $srcVM -snapshot (Get-Snapshot -vm $srcVM -Name $srcSnapshot) -confirm:$false
$cloneSpec.Snapshot = $vm.Snapshot.CurrentSnapshot
$cloneSpec.Location.Host = (get-vmhost -Name $destHost).Extensiondata.MoRef
$cloneSpec.Location.Pool = (Get-ResourcePool -Name Resources -Location (Get-VMHost -Name
$destHost)).Extensiondata.MoRef
# Start the Clone task using the above parameters
$task = $vm.CloneVM_Task($vm.parent, $destVMName, $cloneSpec)
# Get the task object
$task = Get-Task | where { $_.id -eq $task }
#Wait for the taks to Complete
Wait-Task -Task $task

$newvm = Get-vm $destVMName
$customSpec = Get-OSCustomizationSpec $cSpec
Set-vm -OSCustomizationSpec $cSpec -vm $newvm -confirm:$false
if ($disableVMConsole -eq "yes")
{
    Disable_VM_Console($destVMName)
}
# Start the VM
Start-VM $newvm
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```
PowerCLI C:\scripts> .\CloneVMs.ps1
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
Clone Type<"linked" or "Full"> : linked
Disable vSphere VM Console ("yes" or "no", recommend "yes") : yes
```

The time that the cloning process takes depends on the number of desktop machines and can range from several minutes to a number of hours. To verify that the process is complete, from vSphere client, make sure that the last desktop virtual machine is powered on and VMware Tools is running.

Sample PowerCLI Script to Install View Agent on Linux Desktops

You can customize and use the following sample script to install View Agent on any number of Linux desktop machines.

Before you run the script, make sure that View Agent is not already installed on the desktop machines.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 6 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in [“Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops,”](#) on page 25. This script also interactively asks for the following information:

- IP address of the vCenter Server
- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Administrator login name for the ESXi host
- Administrator password for the ESXi host
- User login name for the Linux guest operating system
- User password for the Linux guest operating system
- View Agent tar ball path
- IP address of View Connection Server
- Administrator login name for View Connection Server
- Administrator password for View Connection Server
- Domain name of the View Connection Server administrator

Script Content

```
<#
.SYNOPSIS
Create Clones from a Master VM

.DESRIPTION
The Tool is to upload the Linux Agent installer tar ball to destination VM and do installation
```

```

.NOTES
#>

#----- Handle input -----
"-----"
SvcAddress = Read-Host 'Your vCenter address'
SvcAdmin = Read-Host 'Your vCenter admin user name'
SvcPassword = Read-Host 'Your vCenter admin user password' -AsSecureString
SvcPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($SvcPassword))
"-----"
$hostAdmin = Read-Host 'Your ESXi host admin user name, such as root'
$hostPassword = Read-Host 'Your ESXi admin user password' -AsSecureString
$hostPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($hostPassword))
"-----"
$guestUser = Read-Host 'Your VM guest OS user name'
$guestPassword = Read-Host 'Your VM guest OS user password' -AsSecureString
$guestPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($guestPassword))
"-----"
$agentInstaller = Read-Host 'Type the View Agent tar ball path. Please take care the installer arch'
#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

"-----"
$brokerAddress = Read-Host 'Type the View Connection Server address'
$brokerAdminName = Read-Host 'Type the View Connection Server Admin user name'
"Please type the View Connection Server Admin user password."
"Plase note that special character in password may not work with the script"
$brokerAdminPassword = Read-Host 'Your broker admin password' -AsSecureString
$brokerAdminPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($brokerAdminPassword))
$domainName = Read-Host 'Type the View Connection Server Admin user domain name'

#check if file exists
if (!(Test-Path $agentInstaller))
{
write-host -ForegroundColor Red "installer File not found"
exit
}

#check if file exists
if (!(Test-Path $csvFile))
{
write-host -ForegroundColor Red "CSV File not found"
exit
}
"-----"
#----- Functions -----

```

```

function RunCmdInVM($VMName, $cmd)
{
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestpwd -Confirm:$false -ScriptText $cmd -ScriptType Bash
}

function UploadFileToVM($VMName, $localFile, $destFolder)
{
    Write-Host "Upload File '$localFile' to '$destFolder' of VM '$VMName' with user '$guestUser'"
    Copy-VMGuestFile -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestpwd -Confirm:$false -LocalToGuest -Destination $destFolder -
Source $localFile
}

#----- Main -----
#Read input CSV file
Disconnect-VIServer $vcAddress -Confirm:$false
#Connect-VIServer $vcAddress -ErrorAction Stop -user $vcAdmin -password $vcPassword
Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
$csvData = Import-CSV $csvFile
$destFolder = "/home/$guestUser/"
foreach ($line in $csvData)
{
    $VMName = $line.VMName
    $cmd = "rm -rf VMware-viewagent-linux-*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    #Upload installer tar ball to Linux VM
    Write-Host "Upload File '$agentInstaller' to '$destFolder' of VM '$VMName' with user
'$guestUser'"
    Copy-VMGuestFile -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -LocalToGuest -Destination $destFolder -
Source $agentInstaller

    $cmd = "tar -xvf VMware-viewagent-linux-*.tar"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    $cmd = "sudo hostname $VMName"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    $cmd = "cd VMware-viewagent-linux-* && ./install_viewagent.sh -n $VMName -b $brokerAddress -
d $domainName -u $brokerAdminName -p $brokerAdminPassword"
    Write-Host "Run install cmd in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
}

```

```

}

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```

PowerCLI C:\scripts> .\InstallAgent.ps1
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

-----

Your ESXi host admin user name, such as root: root
Your ESXi host admin user password: *****

-----

Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****

-----

Type the View Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-1.0-1234567.tar

-----

Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character in password may not work with the script
Your broker admin password: *****
Type the View Connection Server Admin user domain name: domain1

```

After the script completes, you can log in to View Administrator and view the Linux desktop machines by navigating to **View Configuration > Registered Machines** and clicking the **Others** tab.

Sample PowerCLI Script to Uninstall View Agent on Linux Desktops

You can customize and use the following sample script to uninstall View Agent on any number of Linux desktop machines so that you can install another View Agent build. The script does not remove the desktop machines from View Connection Server.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 6 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in [“Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops,”](#) on page 25. This script also interactively asks for the following information:

- IP address of the vCenter Server
- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Administrator login name for the ESXi host
- Administrator password for the ESXi host
- User login name for the Linux guest operating system
- User password for the Linux guest operating system

Script Content

```

<#
.SYNOPSIS
Create Clones from a Master VM

.DESCRIPTION
The Tool is to uninstall the Linux Agent from destination VM
.NOTES
#>

#----- Handle input -----
"-----"
SvcAddress = Read-Host 'Your vCenter address'
SvcAdmin = Read-Host 'Your vCenter admin user name'
SvcPassword = Read-Host 'Your vCenter admin user password' -AsSecureString
SvcPassword =
[Runtime.InteropServices]::PtrToStringAuto([Runtime.InteropServices]::SecureStringToBSTR($SvcPassword))
"-----"
$HostAdmin = Read-Host 'Your ESXi host admin user name, such as root'
$HostPassword = Read-Host 'Your ESXi admin user password' -AsSecureString
$HostPassword =
[Runtime.InteropServices]::PtrToStringAuto([Runtime.InteropServices]::SecureStringToBSTR($HostPassword))
"-----"
$GuestUser = Read-Host 'Your VM guest OS user name'
$GuestPassword = Read-Host 'Your VM guest OS user password' -AsSecureString
$GuestPassword =
[Runtime.InteropServices]::PtrToStringAuto([Runtime.InteropServices]::SecureStringToBSTR($GuestPassword))
"-----"
#$CsvFile = Read-Host 'Csv File '

$CsvFile = '.\CloneVMs.csv'

#check if file exists
if (!(Test-Path $CsvFile))
{
write-host -ForegroundColor Red "CSV File not found"
exit
}
"-----"

#----- Functions -----
function RunCmdInVM($VMName, $cmd)
{
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$GuestUser'"
    Invoke-VMScript -HostUser $HostAdmin -HostPassword $HostPassword -VM $VMName -GuestUser $GuestUser -GuestPassword $GuestPwD -Confirm:$false -ScriptText $cmd -ScriptType Bash
}

function UploadFileToVM($VMName, $localFile, $destFolder)
{

```

```

    Write-Host "Upload File '$localFile' to '$destFolder' of VM '$VMName' with user '$guestUser'"
    Copy-VMGuestFile -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestpwd -Confirm:$false -LocalToGuest -Destination $destFolder -
Source $localFile
}

#----- Main -----
#Read input CSV file
Disconnect-VIServer $vcAddress -Confirm:$false
#Connect-VIServer $vcAddress -ErrorAction Stop -user $vcAdmin -password $vcPassword
Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
$csvData = Import-CSV $csvFile

foreach ($line in $csvData)
{
    $VMName = $line.VMName
    $cmd = "sudo /usr/lib/vmware/viewagent/bin/uninstall_viewagent.sh"
    Write-Host "Run uninstall cmd in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    $cmd = "sudo rm -rf /var/log/vmware/*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    $cmd = "sudo rm -rf /tmp/vmware-root"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText
$cmd
}

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following screen messages show a sample execution of the script:

```

PowerCLI C:\scripts> .\UninstallAgent.ps1
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

-----
Your ESXi host admin user name, such as root: root
Your ESXi host admin user password: *****

-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****

```

After the script completes, View Agent no longer runs on the desktop machines, but the machines and the desktop pool are not removed from View Connection Server. You can run the View Agent installation script to install a different View Agent build and the same desktop pool and desktop machines will be available.

Sample PowerCLI Script to Perform Operations on Linux Desktop Machines

You can customize and use the following sample script to perform operations on any number of Linux desktop machines.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 6 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in “[Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops](#),” on page 25. This script also interactively asks for the following information:

- IP address of the vCenter Server
- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Action to perform, which can be power-on, power-off, shut down guest, restart machine, or delete virtual machine

Script Content

```
<#
.DESCRIPTION
The Tool supports:
1. Power off VMs
2. Power on VMs
3. Shutdown VMs
4. Restart VMs
5. Restart VM guest
6. Delete VMs from Disk
.NOTES
#>

#----- Functions -----

function IsVMExists ($VMExists)
{
    Write-Host "Checking if the VM $VMExists Exists"
    [bool]$Exists = $false

    #Get all VMS and check if the VMs is already present in VC
    $listvm = Get-vm
    foreach ($lvm in $listvm)
    {
        if($VMExists -eq $lvm.Name )
        {
            $Exists = $true
            Write-Host "$VMExists is Exist"
        }
    }
    return $Exists
}
```

```

function Delete_VM($VMToDelete)
{
    Write-Host "Deleting VM $VMToDelete"
    Get-VM $VMToDelete | where { $_.PowerState -eq "PoweredOn" } | Stop-VM -confirm:$false
    Get-VM $VMToDelete | Remove-VM -DeleteFromDisk -confirm:$false
}

#----- Handle input -----
"-----"
$vcAddress = Read-Host 'Your vCenter address'
$vcAdmin = Read-Host 'Your vCenter admin user name'
$vcPassword = Read-Host 'Your vCenter admin user password' -AsSecureString
$vcPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($vcPassword))
"-----"
#$csvFile = Read-Host 'Csv File '
$action = Read-Host 'Select action: 1). Power On 2). Power Off 3) Shutdown VM Guest 4). Restart VM 5). Restart VM Guest 6). Delete VM'
$sleepTime = Read-Host 'Wait time (seconds) between each VM'
"-----"
switch ($action)
{
    1
    {
        "Your selection is 1). Power On"
    }
    2
    {
        "Your selection is 2). Power Off"
    }
    3
    {
        "Your selection is 3) Shutdown"
    }
    4
    {
        "Your selection is 4). Restart VM"
    }
    5
    {
        "Your selection is 5). Restart VM Guest"
    }
    6
    {
        "Your selection is 6). Delete VM"
    }
    default
    {
        "Invalid selection for action: $action"
        exit
    }
}

$csvFile = '.\CloneVMs.csv'

```

```

#check if file exists
if (!(Test-Path $csvFile))
{
write-host -ForegroundColor Red "CSV File not found"
exit
}
"-----"

#----- Main -----
#Read input CSV file
Disconnect-VIServer $vcAddress -Confirm:$false
#Connect-VIServer $vcAddress -ErrorAction Stop -user $vcAdmin -password $vcPassword
Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
$csvData = Import-CSV $csvFile

foreach ($line in $csvData)
{
    $VMName = $line.VMName
    switch ($action)
    {
        1
        {
            Get-VM $VMName | Start-VM -Confirm:$false
        }
        2
        {
            Get-VM $VMName | Stop-VM -Confirm:$false
        }
        3
        {
            Get-VM $VMName | Shutdown-VMGuest -Confirm:$false
        }
        4
        {
            Get-VM $VMName | Restart-VM -Confirm:$false
        }
        5
        {
            Get-VM $VMName | Restart-VMGuest -Confirm:$false
        }
        6
        {
            if (IsVMExists ($VMName))
            {
                Delete_VM ($VMName)
            }
        }
        default{}
    }
    Start-Sleep -s $sleepTime
}

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```
PowerCLI C:\scripts> .\InstallAgent.ps1
```

```
Your vCenter address: 10.117.44.17
```

```
Your vCenter admin user name: administrator
```

```
Your vCenter admin user password: *****
```

```
-----  
Select action: 1). Power On 2). Power Off 3) Shutdown VM Guest 4). Restart VM 5). Restart VM  
Guest 6). Delete VM: 1
```

```
Wait time (seconds) between each VM: 20
```

For the operations power on, restart VM, and restart VM guest, specify a wait time between virtual machines of at least 20 seconds to avoid a boot storm situation, which might cause some operations to fail.

Administering Horizon 6 for Linux Desktops

4

After you deploy Horizon 6 for Linux desktops, you can manage, uninstall, and troubleshoot the virtual machines in your deployment.

This chapter includes the following topics:

- [“Uninstalling and Reinstalling Horizon 6 for Linux Machines,”](#) on page 39
- [“Configure Lossless PNG Images on Linux Desktops,”](#) on page 42
- [“Suppress the vSphere Console Display of a Linux Desktop,”](#) on page 42
- [“Configure a Left-handed Mouse on Ubuntu Desktops,”](#) on page 42
- [“How to Perform Power Operations on Linux Desktops from vSphere,”](#) on page 42
- [“Gather Information About Horizon 6 for Linux Software,”](#) on page 43
- [“Troubleshooting Horizon 6 for Linux,”](#) on page 43

Uninstalling and Reinstalling Horizon 6 for Linux Machines

To uninstall Horizon 6 for Linux on a virtual machine, you must remove the unmanaged machine from the desktop pool, remove the registered machine from View, uninstall View Agent, and remove configuration files. You can reinstall Horizon 6 for Linux by running the View Agent installation script.

- [Remove an Unmanaged Machine from a Manual Desktop Pool](#) on page 39
You can reduce the size of a manual desktop pool by removing unmanaged machines from the pool.
- [Remove Registered Machines from View](#) on page 40
If you do not plan to use a registered machine again, you can remove it from View.
- [Uninstall View Agent on a Linux Virtual Machine](#) on page 40
To uninstall View Agent on a Linux virtual machine, you use the `uninstall_viewagent.sh` script. You also must remove the Horizon 6 for Linux configuration files from the machine.
- [Reinstall View Agent on a Linux Virtual Machine](#) on page 41
You use different approaches to reinstall View Agent on a Linux virtual machine, depending on whether you use an existing Horizon 6 for Linux configuration or reinstall with a fresh configuration.

Remove an Unmanaged Machine from a Manual Desktop Pool

You can reduce the size of a manual desktop pool by removing unmanaged machines from the pool.

Procedure

- 1 In View Administrator, select **Catalog > Desktop Pools**.

- 2 Double-click the pool ID of the manual pool.
- 3 Select the **Inventory** tab.
- 4 Select the unmanaged machines to remove.
- 5 Click **Remove**.
- 6 If users are logged in to the unmanaged machine-based desktops, choose whether to terminate the sessions or let the sessions remain active.

Option	Description
Leave active	Active sessions remain until the user logs off. View Connection Server does not keep track of these sessions.
Terminate	Active sessions end immediately.

- 7 Click **OK**.

The unmanaged machines are removed from the pool.

Remove Registered Machines from View

If you do not plan to use a registered machine again, you can remove it from View.

There are two types of registered machines in View: RDS Hosts and Others. Unmanaged machines are in the Others category. Unmanaged machines include physical computers and virtual machines that are not managed by vCenter Server, including Horizon 6 for Linux virtual machines. They are used to form manual desktop pools that do not contain vCenter Server virtual machines.

After you remove a registered machine, it becomes unavailable in View. To make the machine available again, you must reinstall View Agent.

Prerequisites

Verify that the registered machines that you want to remove are not being used in any desktop pool.

Procedure

- 1 In View Administrator, select **View Configuration > Registered Machines**.
- 2 Click the **Others** tab.
- 3 Select one or more machines and click **Remove**.

You can select only machines that are not being used by a desktop pool.

- 4 Click **OK** to confirm.

Uninstall View Agent on a Linux Virtual Machine

To uninstall View Agent on a Linux virtual machine, you use the `uninstall_viewagent.sh` script. You also must remove the Horizon 6 for Linux configuration files from the machine.

Prerequisites

- Verify that the machine was removed from the desktop pool and from Horizon 6. See [“Remove an Unmanaged Machine from a Manual Desktop Pool,”](#) on page 39 and [“Remove Registered Machines from View,”](#) on page 40.

Procedure

- 1 Open a terminal window on the virtual machine and run the View Agent uninstall script.

For example:

```
sudo /usr/lib/vmware/viewagent/bin/uninstall_viewagent.sh
```

The script stops View Agent processes, deletes the View Agent service, and deletes the View Agent installation directory, `/usr/lib/vmware/viewagent`.

- 2 Manually delete the Horizon 6 for Linux configuration files, located in the `/etc/vmware/` directory.

If you intend to reinstall View Agent (for example, to upgrade to a later View Agent version), you do not have to delete these configuration files.

```
delete /etc/vmware/viewagent-machine.cfg
delete /etc/vmware/viewagent-config.txt
delete /etc/vmware/jms
delete /etc/vmware/ssl
```

Reinstall View Agent on a Linux Virtual Machine

You use different approaches to reinstall View Agent on a Linux virtual machine, depending on whether you use an existing Horizon 6 for Linux configuration or reinstall with a fresh configuration.

You can reinstall View Agent with an existing configuration if the Horizon 6 for Linux configuration files are present on the virtual machine and the machine is registered in View Connection Server.

If either the configuration files were removed from the machine or the registered machine name was removed from View Connection Server, you must ensure that both the local configuration files and the registered machine name in View Connection Server are removed. You can then perform a fresh installation of View Agent with a new configuration.

Prerequisites

- Verify whether the Horizon 6 for Linux configuration files are present on the machine.
 - `/etc/vmware/viewagent-machine.cfg`
 - `/etc/vmware/viewagent-config.txt`
 - `/etc/vmware/jms`
 - `/etc/vmware/ssl`
- Verify whether the Linux virtual machine is registered in View Connection Server. In View Administrator, select **View Configuration > Registered Machines** and select the **Others** tab.
- If the configuration files and registered machine name were removed, familiarize yourself with the steps for performing a fresh installation of View Agent. See [“Install View Agent on a Linux Virtual Machine,”](#) on page 12.

Procedure

- To reinstall with an existing Horizon 6 for Linux configuration, you can run the View Agent installation script without parameters.

```
sudo ./install_viewagent.sh
```

The View Agent installer reuses the existing configuration files and machine registration in View Connection Server. The installer does not register the machine again.

- To reinstall with a new Horizon 6 for Linux configuration, run the View Agent installation script with the required parameters.

The installation steps are the same as those in a fresh installation. The installer registers the machine with View Connection Server.

Configure Lossless PNG Images on Linux Desktops

Graphic applications, especially graphic design applications, require pixel-exact rendering of images in the client display of a Linux desktop. You can configure a lossless PNG mode for images and video playback that are generated on a Linux desktop and rendered on the client device.

By default, the lossless PNG mode is disabled.

Procedure

- 1 On the Linux remote desktop, edit the `/etc/vmware/config` file.
- 2 Add the following line to enable the lossless PNG mode.

```
RemoteDisplay.alwaysLossless=TRUE
```

Suppress the vSphere Console Display of a Linux Desktop

When a user connects to a Linux desktop, the desktop can also be displayed in the vSphere console for the Linux virtual machine. You can configure Linux virtual machines to ensure that the vSphere console is blank when users connect to their desktops.

Procedure

- ◆ On the ESXi host, add the following line to the Linux virtual machine's vmx file.

```
RemoteDisplay.maxConnections = "0"
```

The vSphere console display remains blank even when you connect to the virtual machine when the user is logged out of the desktop.

Configure a Left-handed Mouse on Ubuntu Desktops

Ubuntu does not support a left-handed mouse by default. You can configure an Ubuntu desktop to support a left-handed mouse.

Procedure

- 1 On the Linux remote desktop, edit the `/etc/vmware/config` file.
- 2 Add the following line to enable a left-handed mouse.

```
mksVNCServer.useUInputButtonMapping=TRUE
```

How to Perform Power Operations on Linux Desktops from vSphere

Occasionally, it might be necessary to perform power operations on Linux desktops from vSphere.

Do not use the operations **Power Off** or **Reset** because they might cause a loss of data or View Agent might not restart properly. Instead, use the operations **Shut Down Guest** or **Restart Guest**.

Gather Information About Horizon 6 for Linux Software

To help manage and troubleshoot the Horizon 6 for Linux software, you can gather information about a View Agent installation on a Linux virtual machine.

Procedure

- Identify the View Agent version that is running on a Linux machine.

Run the following command:

```
cat /usr/lib/vmware/viewagent/Product.txt
```

The output displays the View Agent version. For example:

```
VMware-viewagent-linux-x.x-yyyyyyy
```

where *x.x* is the product version and *yyyyyyy* is the build number.

- Gather information about View Agent software.

Files	Location
View Agent installation directory	/usr/lib/vmware/viewagent
Logs	/var/log/vmware/
View Agent configuration	/etc/vmware/viewagent-config.txt
Machine configuration	/etc/vmware/viewagent-machine.cfg

- Stop, start, or restart the View Agent service.

A View Agent service with these operations is provided on the Linux distributions. For example, on Ubuntu 12.04, you can run the following command:

```
sudo service viewagent <stop/start/restart>
```

Troubleshooting Horizon 6 for Linux

You can take steps to diagnose and resolve problems that you encounter when you install and configure Horizon 6 for Linux desktops.

- [Collect Diagnostic Information for a Horizon 6 for Linux Machine](#) on page 44
You can collect diagnostic information to help VMware Technical Support diagnose and resolve issues with a Horizon 6 for Linux machine. You create a Data Collection Tool (DCT) bundle that gathers the machine's configuration information and logs into a compressed tarball.
- [Troubleshooting View Agent Registration Failure for a Linux Machine](#) on page 44
The View Agent installer fails to register a Linux virtual machine with View Connection Server.
- [Troubleshooting an Unreachable View Agent on a Linux Machine](#) on page 45
View Connection Server cannot communicate with View Agent on a Linux virtual machine.
- [Troubleshooting View Agent on a Linux Machine That Is Not Responding](#) on page 46
View Agent that is installed on a Linux virtual machine is not responding.
- [Configuring the Linux Firewall to Allow Incoming TCP Connections](#) on page 47
To allow users to connect to their Linux desktops, the desktops must be able to accept incoming TCP connections from Horizon Client devices, security server, and View Connection Server.

Collect Diagnostic Information for a Horizon 6 for Linux Machine

You can collect diagnostic information to help VMware Technical Support diagnose and resolve issues with a Horizon 6 for Linux machine. You create a Data Collection Tool (DCT) bundle that gathers the machine's configuration information and logs into a compressed tarball.

Procedure

- 1 Log in to the Linux virtual machine as a user with the required privileges.
- 2 Open a command prompt and run the `dct-debug.sh` script.

```
sudo /usr/lib/vmware/viewagent/bin/dct-debug.sh
```

The script generates a tarball that contains the DCT bundle. For example:

```
ubuntu-12-vdm-sdct-20150201-0606-agent.tgz
```

The tarball is generated in the directory from which the script was executed (the current working directory).

Troubleshooting View Agent Registration Failure for a Linux Machine

The View Agent installer fails to register a Linux virtual machine with View Connection Server.

Problem

When you install View Agent on a Linux virtual machine, the `Agent Registration Failed` error is displayed.

Cause

The View Agent installer might fail to register a Linux machine with View Connection Server for the following reasons:

- The View Connection Server FQDN, domain name, user name, or password that was provided during View Agent installation was incorrect.
- The View administrator user does not have the **Agent Registration Administrators** or **Administrators** role in View Administrator. Specifically, the **Register Agent** privilege is required to register View Agent with View Connection Server. **Agent Registration Administrators** is a restricted role that provides this minimum privilege.
- The View administrator password contains a special character, such as \$, that was not escaped with a backslash.
- Reversible password encryption was not enabled on the Windows Server 2012 or 2012 R2 Active Directory (AD) server.

Solution

- 1 Verify that you have the correct View Connection Server FQDN, domain name, user name, and password.
- 2 Verify that the View administrator user has the privileges to allow View Agent registration.
 - a In View Administrator, select **View Configuration > Administrators**.
 - b In the Administrators and Groups tab, select the View administrator user that you provided during View Agent installation.
 - c Verify that the user has the **Agent Registration Administrators** or **Administrators** role.

- 3 If the View administrator password includes a special character, escape the special character with a backslash.
For example: `ab\$cdef`
- 4 On the Linux guest operating system, open the View Agent registration log.
`/usr/lib/vmware/viewagent/viewagent-registration.log`
- 5 If the View Agent registration log reports LDAP errors, and a message asks if reversible password encryption was enabled on a Windows Server 2012 or Windows Server 2012 R2 Active Directory (AD) server, enable reversible password encryption.
 - a On the AD server, open AD users and computers.
 - b Right-click the View Admin account, or create a new Linux View Agent account, and select **Properties**.
 - c Select the Account tab.
 - d In Account options, select the **Store password using reversible encryption** setting.
 - e Reset the View Admin password.
- 6 Install View Agent on the Linux virtual machine again.

Troubleshooting an Unreachable View Agent on a Linux Machine

View Connection Server cannot communicate with View Agent on a Linux virtual machine.

Problem

In View Administrator, the Linux virtual machine is displayed as Agent `Unreachable`. This problem occurs after View Agent was installed and the View Agent service started successfully on the Linux machine.

Cause

One cause might be that the Linux machine cannot resolve the FQDNs of the View Connection Server instances.

After installation, View Agent stores a list of FQDNs of the View Connection Server instances in the `/etc/vmware/viewagent-machine.cfg` configuration file. View Agent uses FQDNs even if you specify an IP address of a View Connection Server instance with the `-b` parameter during the View Agent installation.

A second cause might be that the hostname of the Linux machine itself cannot be resolved.

Solution

- 1 On the Linux machine, open the View Agent log file.
`/usr/lib/vmware/viewagent/viewagent-debug.log`
- 2 Look for messages that indicate the hostname of the View Connection Server instance cannot be resolved.

For example:

```
2015-01-31T09:21:33.516Z DEBUG <JMS Handler for:svohraUb12x6> [JmsManager]
    Using connection broker sm-15q1-broker.myDomain.com
2015-01-31T09:21:33.518Z DEBUG <JMS Handler for:svohraUb12x6> [JmsManager]
    Unable to resolve hostname for sm-15q1-broker.myDomain.com
```

- 3 Make sure the DNS service is configured properly on the Linux machine.

The steps for configuring DNS vary depending on the Linux distribution and release. For instructions, consult the documentation for your Linux distribution.

As a workaround, you can manually add the View Connection Server FQDNs to the `/etc/hosts` file on the Linux machine. However, this method is not recommended for production use. The proper solution is to configure the DNS service so that the Linux machine can resolve the View Connection Server FQDNs.

After you take these steps, ping the View Connection Server hostnames to verify that they can be resolved. For example:

```
ping sm-15q1-broker.myDomain.com
```

- 4 If the problem continues, make sure that the system hostname of the Linux machine is resolvable.

For example, on CentOS you might take the following steps:

- a Open a terminal window and run the `hostname` command.

The machine's hostname is displayed.

- b Open the `etc/hosts` file and add the hostname.

```
su
nano /etc/hosts
# Add the hostname:
127.0.0.1 <your hostname>
```

- c Ping the hostname to verify that it can be resolved.

```
ping <your hostname>
```

NOTE If you specify the DNS server on the Linux machine by manually editing the `/etc/resolv.conf` file, the setting might become lost in some Linux distributions. The setting is preserved if you specify the DNS server in the `/etc/hosts` file. For complete instructions for configuring DNS and the hostname, consult the documentation for your Linux distribution and release.

Troubleshooting View Agent on a Linux Machine That Is Not Responding

View Agent that is installed on a Linux virtual machine is not responding.

Problem

The Horizon 6 for Linux desktop is not accessible, and View Agent is not responding.

Cause

The View Agent service might not be running, or the X session might not be active on the Linux machine.

Solution

- 1 On the Linux virtual machine, open a terminal window and stop and restart the View Agent service.

```
sudo service viewagent <stop/start/restart>
```

- 2 Verify that the X session is active.

```
ps -A | grep X
```

- 3 Install the X11 Simple VNC server on the virtual machine and verify that the X session is active over VNC.

Configuring the Linux Firewall to Allow Incoming TCP Connections

To allow users to connect to their Linux desktops, the desktops must be able to accept incoming TCP connections from Horizon Client devices, security server, and View Connection Server.

On Ubuntu and Kylin distributions, the `iptables` firewall is configured by default with an input policy of `ACCEPT`.

On RHEL and CentOS distributions, where possible, the View Agent installer script configures the `iptables` firewall with an input policy of `ACCEPT`.

Make sure that `iptables` on a RHEL or CentOS guest operating system has an input policy of `ACCEPT` for new connections from the Blast port, 5443.

When the BSG is enabled, client connections are directed from a Horizon Client device through the BSG on a security server or View Connection Server to the Linux desktop. When the BSG is not enabled, connections are made directly from the Horizon Client device to the Linux desktop.

Index

B

- bulk deployment
 - create a virtual machine template for cloning **24**
 - input file for sample scripts **25**
 - overview **23**

D

- desktop pool creation, Linux virtual machines **14**

F

- firewalls, Linux desktops **47**

G

- glossary **5**

H

- Horizon 6 for Linux
 - bulk deployment **23**
 - gathering information **43**
 - reinstalling **41**
 - troubleshooting View Agent that is not responding **46**
 - uninstalling **40**
- Horizon for Linux
 - reinstalling **39**
 - uninstalling **39**

I

- install_viewagent.sh, command usage and options **14**
- intended audience **5**

J

- JREs, required for View Agent on Linux **12**

L

- left-handed mouse, Ubuntu desktops **42**
- Linux installation
 - troubleshooting **43**
 - View Agent fails to register **44**
 - View Agent is unreachable **45**
- Linux systems
 - adding to desktop pools **14**
 - administering **39**
 - collecting diagnostic information **44**
 - configuring the firewall **47**
 - desktop features **7**

- desktops **5**

- preparing for desktop use **10**
- setting up for use as desktops **7**
- vSphere console display **42**
- lossless PNG display, Linux desktops **42**

N

- NVIDIA GRID driver, installing on RHEL 6.6 **19**

P

- physical computers, removing from a pool **39**
- power operations from vSphere, Linux desktops **42**

R

- registered machines
 - removing **40**
 - removing from View **40**
- removing registered machines **40**
- RHEL, configuring for vDGA **17**

S

- sample script
 - to clone desktop machines **25**
 - to install View Agent **29**
 - to perform operations on VMs **35**
 - to uninstall View Agent **32**
- sample scripts, input file **25**

U

- unmanaged machines, removing from a pool **39**

V

- vDGA
 - adding PCI device to a Linux machine **18**
 - enabling DirectPath I/O on a host **17**
- vDGA (Virtual Dedicated Graphics Acceleration) for Linux machines **17**
 - for RHEL 6.6 **17**
- View Agent
 - installing on a Linux virtual machine **12**
 - Linux installation command options **14**
 - Linux operating system requirements **8**
 - reinstalling on a Linux machine **41**
 - uninstalling on a Linux machine **40**
- virtual machines, creating for Linux desktops **10**
- VMware Tools, configuring on a Linux guest **10**
- vSphere console, suppress display **42**

