

Setting Up Horizon 7 Version 7.0.3 for Linux Desktops

VMware Horizon 7 7.0.3

VMware Horizon 7 7.0.2

VMware Horizon 7 7.0.1

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

<https://docs.vmware.com/>

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 © 2016 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 7 Version 7.0.3 for Linux Desktops	5
1 Features and System Requirements	7
Features of Horizon Linux Desktops	7
Overview of Configuration Steps for Horizon 7 for Linux Desktops	9
System Requirements for Horizon 7 for Linux	10
2 Preparing a Linux Virtual Machine for Desktop Deployment	15
Create a Virtual Machine and Install Linux	15
Prepare a Linux Machine for Remote Desktop Deployment	16
Install Dependency Packages for Horizon Agent	18
3 Setting Up Active Directory Integration for Linux Desktops	21
Integrating Linux with Active Directory	21
Setting Up Single Sign-on and Smart Card Redirection	22
4 Setting Up Graphics for Linux Desktops	25
Configure RHEL 6.6/6.7/6.8 and RHEL 7.2 for vGPU	25
Configure RHEL 6.6/6.7/6.8 for vDGA	30
Configure RHEL 7.2 for vSGA	34
5 Installing Horizon Agent	37
Install Horizon Agent on a Linux Virtual Machine	37
Configure the Certificate for Linux Agent	39
Upgrading the Horizon Agent on a Linux Virtual Machine	40
Uninstall Horizon 7 for Linux Machines	41
6 Configuration Options for Linux Desktops	43
Setting Options in Configuration Files on a Linux Desktop	43
Example Blast Settings for Linux Desktops	46
Suppress the vSphere Console Display of a Linux Desktop	47
7 Create and Manage Linux Desktop Pools	49
Create a Manual Desktop Pool for Linux	49
Manage Desktop Pool for Linux	50
Create an Automated Full-Clone Desktop Pool for Linux	51
Broker PowerCLI Commands	53
8 Bulk Deployment of Horizon 7 for Manual Desktop Pools	57
Overview of Bulk Deployment of Linux Desktops	57

- Overview of Bulk Upgrade of Linux Desktops 59
- Create a Virtual Machine Template for Cloning Linux Desktop Machines 59
- Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops 61
- Sample Script to Clone Linux Virtual Machines 61
- Sample Script to Join Cloned Virtual Machines to AD Domain 65
- Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH 68
- Sample Script to Upload Configuration Files to Linux Virtual Machines 71
- Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH 75
- Sample Script to Upgrade Horizon Agent on Linux Desktop Machines 79
- Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH 83
- Sample Script to Perform Operations on Linux Virtual Machines 89

- 9 Troubleshooting Linux Desktops 93**
 - Collect Diagnostic Information for Horizon 7 for Linux Machine 93
 - Troubleshooting Copy and Paste between Remote Desktop and Client Host 94
 - Configuring the Linux Firewall to Allow Incoming TCP Connections 94
 - View Agent Fails to Disconnect on an iPad Pro Horizon Client 94
 - SLES 12 SP1 Desktop does not Auto Refresh after Drag and Drop 95
 - SSO Fails to Connect to a PowerOff Agent 95
 - Unreachable VM After Creating a Manual Desktop Pool for Linux 95

- Index 97

Setting Up Horizon 7 Version 7.0.3 for Linux Desktops

The *Setting Up Horizon 7 for Linux Desktops* document provides information about setting up a Linux virtual machine for use as a VMware Horizon[®] 7 desktop, including preparing the Linux guest operating system, installing Horizon Agent on the virtual machine, and configuring the machine in View Administrator for use in a Horizon 7 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.

Features and System Requirements

With Horizon 6 or later, users can connect to remote desktops that run the Linux operating system.

This chapter includes the following topics:

- [“Features of Horizon Linux Desktops,”](#) on page 7
- [“Overview of Configuration Steps for Horizon 7 for Linux Desktops,”](#) on page 9
- [“System Requirements for Horizon 7 for Linux,”](#) on page 10

Features of Horizon Linux Desktops

Horizon 7 introduces a number of new features for Linux desktops.

The following new features are supported for Horizon 7.0.3 release.

- Audio input support
- Ubuntu 16.04 support
- Software H.264 Encoder to support multiple monitors
- Clipboard redirection support on all distributions
- vGPU support with NVIDIA M6 graphics card on RHEL 6.6/6.7/6.8/7.2 Workstation x64

The following list presents the key features for Horizon Linux Desktops.

Automated Full-Clone Desktop Pool	From Horizon 7 version 7.0.2, you can create automated full-clone desktop pools for Linux desktops.
Manual Desktop Pool	Machine source. <ul style="list-style-type: none"> ■ Managed Virtual Machine - Machine source of vCenter virtual machine. Managed virtual machine is supported for new and upgrade deployment. ■ Un-Managed Virtual Machine - Machine source of other sources. Un-Managed virtual machine is only supported when the upgrade from Un-Managed virtual machine deployment.
<hr/>	
NOTE Managed virtual machine requires View Connection Server 7.0.1 or later.	
<hr/>	
Multiple Monitors	<ul style="list-style-type: none"> ■ vDGA/vGPU desktop supports a maximum resolution of 2560x1600 on 4 monitors.

- 2D/vSGA desktop on vSphere 6.0 or later supports a maximum resolution of 2048x1536 on 4 monitors or a maximum resolution of 2560x1600 on 3 monitors.
- 2D desktop on vSphere 5.5 U3
 - RHEL/CentOS 6.6/6.7/6.8 and SLED 11 SP3/SP4 supports a maximum resolution of 2560x1600 on 4 monitors.
 - Ubuntu 12.04/14.04/16.04, RHEL/CentOS 7.2, and SLES 12 SP1 supports a maximum resolution of 2048x1536 on 4 monitors, or maximum resolution of 2560x1600 on 3 monitors.

For Ubuntu 14.04 and 16.04, you must disable Compiz to support multiple monitors. See <http://kb.vmware.com/kb/2114809> for more information.

For SLES 12 SP1, you must use the default package with kernel level kernel-default-3.12.49-11.1. If you upgraded the package, the multi-monitor feature will not work, and the desktop will be shown in one monitor.

Lossless PNG

Images and videos that are generated on a desktop are rendered on the client device in a pixel-exact manner.

Software H.264 Encoder

H.264 can improve the Blast Extreme performance for Horizon desktop, especially under low bandwidth network. If client side disabled H.264, Blast Extreme will fall back to JPEG/PNG encoding automatically.

From Horizon 7 version 7.0.3, multiple monitors are supported.

3D Graphics

3D graphics is supported with the following combinations of Linux versions and graphics cards:

- vSGA is supported on RHEL 7.2 Workstation x64 with NVIDIA GRID K1 or K2 graphics cards.
- vDGA is supported on RHEL 6.6/6.7/6.8 Workstation x64 with NVIDIA GRID K1 or K2 graphics cards.
- vGPU is supported on RHEL 6.6/6.7/6.8 Workstation x64 with NVIDIA Maxwell M60 graphics cards.
- vGPU is supported on RHEL 7.2 Workstation x64 with NVIDIA Maxwell M60 graphics cards.
- vGPU is supported on RHEL 6.6/6.7/6.8 Workstation x64 with NVIDIA M6 graphics cards.
- vGPU is supported on RHEL 7.2 Workstation x64 with NVIDIA M6 graphics cards.

Clipboard Redirection

Clipboard redirection allows you to copy and paste rich text or plain text between a client host and a remote Linux desktop. You can set the copy/paste direction and the maximum text size using Horizon agent options.

Single Sign-on

Single sign-on is supported on the following Linux versions:

- RHEL 6.6/6.7/6.8 Workstation x64
- CentOS 6.6/6.7/6.8 x64
- SLED 11 SP3/SP4 x64

Smart Card Redirection with SSO

Smart card redirection is supported on RHEL 6.6/6.7/6.8 Workstation x64. Personal Identity Verification (PIV) cards and Common Access Cards (CAC) are supported. Mac client is not supported.

Audio-in

Audio input redirection from a client host to a remote Linux desktop is supported. This feature is not based on the USB redirection function. You must select this feature during installation if you want it enabled, and you must select system default audio in device "PulseAudio server (local)" in your application for the audio input. The feature is supported on:

- Ubuntu 14.04 x64
- Ubuntu 16.04 x64
- CentOS 7.2 x64
- RHEL 7.2 workstation x64

Linux desktops and desktop pools have the following limitations:

- USB redirection, virtual printing, location-based printing, and Real-Time Video are not supported.

NOTE When a security server is used, port 22443 must be open in the internal firewall to allow traffic between the security server and the Linux desktop.

Overview of Configuration Steps for Horizon 7 for Linux Desktops

When you install and configure Horizon 7 for Linux desktops, you must follow a different sequence of steps depending on whether you install 2D graphics or 3D graphics on the virtual machines.

2D Graphics - Overview of Configuration Steps

For 2D graphics, take the following steps:

- 1 Review the system requirements for setting up a Horizon 7 for Linux deployment. See ["System Requirements for Horizon 7 for Linux,"](#) on page 10.
- 2 Create a virtual machine in vSphere and install the Linux operating system. See ["Create a Virtual Machine and Install Linux,"](#) on page 15.
- 3 Prepare the guest operating system for deployment as a desktop in a Horizon 7 environment. See ["Prepare a Linux Machine for Remote Desktop Deployment,"](#) on page 16.
- 4 Configure the Linux guest operating system to authenticate with Active Directory. This step is implemented with 3rd-party software, based on the requirements in your environment, and is not described in this guide.
- 5 Install Horizon Agent on the Linux virtual machine. See ["Install Horizon Agent on a Linux Virtual Machine,"](#) on page 37.
- 6 Create a desktop pool that contains the configured Linux virtual machines. See ["Create a Manual Desktop Pool for Linux,"](#) on page 49.

3D Graphics - Overview of Configuration Steps

You must complete the NVIDIA GRID vGPU, vDGA, or vSGA configuration on the Linux virtual machines before you install Horizon Agent on the machines and deploy a desktop pool in View Administrator.

- 1 Review the system requirements for setting up a Horizon 7 for Linux deployment. See ["System Requirements for Horizon 7 for Linux,"](#) on page 10.

- 2 Create a virtual machine in vSphere and install the Linux operating system. See [“Create a Virtual Machine and Install Linux,”](#) on page 15.
- 3 Prepare the guest operating system for deployment as a desktop in a Horizon 7 environment. See [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16.
- 4 Configure the Linux guest operating system to authenticate with Active Directory. This step is implemented with 3rd-party software, based on the requirements in your environment, and is not described in this guide.
- 5 Configure 3D capabilities on your ESXi hosts and the Linux virtual machine. Follow the procedures for the 3D feature you intend to install.
 - See [“Configure RHEL 6.6/6.7/6.8 and RHEL 7.2 for vGPU,”](#) on page 25.
 - See [“Configure RHEL 6.6/6.7/6.8 for vDGA,”](#) on page 30.
 - See [“Configure RHEL 7.2 for vSGA,”](#) on page 34.
- 6 Install Horizon Agent on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.
- 7 Create a desktop pool that contains the configured Linux virtual machines. See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

Bulk Deployment

With View Administrator, you can only deploy Linux virtual machines in a manual desktop pool. With vSphere PowerCLI, you can develop scripts that automate the deployment of a pool of Linux desktop machines. See [Chapter 8, “Bulk Deployment of Horizon 7 for Manual Desktop Pools,”](#) on page 57.

System Requirements for Horizon 7 for Linux

Horizon 7 for Linux must meet certain operating system, Horizon 7, and vSphere platform requirements.

Supported Linux Versions for Horizon 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 14.04 and 16.04	x64
NOTE Disable Compiz to avoid poor performance. See http://kb.vmware.com/kb/2114809 for more information.	
Ubuntu 12.04	x64
RHEL 6.6/6.7/6.8 and 7.2	x64
CentOS 6.6/6.7/6.8 and 7.2	x64
NeoKylin 6 Update 1	x64
SLED 11 SP3/SP4	x64
SLES 12 SP1	x64

NOTE Linux agent has dependency packages on some Linux distribution. See [“Install Dependency Packages for Horizon Agent,”](#) on page 18 for more information.

Required Platform and Horizon 7 Software Versions

To install and use Horizon 7 for Linux, your deployment must meet certain vSphere platform, Horizon 7, and client requirements.

Table 1-2. Required platform and Horizon 7 software versions

Platform and Software	Supported Versions
vSphere platform version	vSphere 5.5 U3, vSphere 6.0 U2 or a later release, vSphere 6.5 or a later release.
Horizon environment	<ul style="list-style-type: none"> ■ Horizon Connection Server 7.0.3.
Horizon client software	<ul style="list-style-type: none"> ■ Horizon Client 4.3.0 for Android ■ Horizon Client 4.3.0 for Windows ■ Horizon Client 4.3.0 for Linux ■ Horizon Client 4.3.0 for Mac OS X ■ Horizon Client 4.3.0 for iOS (iPad Pro) ■ HTML Access 4.3.0 on Chrome ■ Zero clients are not supported

NOTE HTML Access does not support Audio Out.

TCP Ports Used by Linux Virtual Machines

View Agent and Horizon Clients use TCP ports for network access between each other and various View server components.

Table 1-3. TCP ports used by Linux virtual machines

Source	Port	Target	Port	Protocol	Description
Horizon Client	*	Linux Agent	22443	TCP	Blast if Blast Security Gateway is not used
Security Server, View Connection Server, or Access Point appliance	*	Linux Agent	22443	TCP	Blast if Blast Security Gateway is used
View Agent	*	View Connection Server	4001, 4002	TCP	JMS SSL traffic.

NOTE For more information on TCP and UDP ports used by clients, see TCP and UDP Ports Used by Clients and View Agent in the *Horizon Client and View Agent Security* in View document.

Verify the Linux Account Used by Linux Virtual Machines

The following table lists the account name and account type used by Linux virtual machines.

Table 1-4. Account Name and Account Type

Account Name	Account Type	Used By
root	Linux OS built-in	Java Standalone Agent, mksvchanserver, shell scripts
vmwblast	created by Linux Agent installer	VMwareBlastServer
<current login user>	Linux OS built-in or AD user or LDAP user	python script

Desktop Environment

Horizon 7 for Linux desktops support only the GNOME desktop environment, which is the default desktop environment for the Ubuntu, SUSE, RHEL, and CentOS distributions.

If you installed the KDE desktop environment, the installer for Horizon 7 for Linux will fail to start.

Virtual Machine Settings for 2D and vSGA Graphics

When you create certain Horizon 7 for Linux virtual machines, you must change the memory settings and configuration parameters to the minimum values recommended.

Virtual machines that are configured to use NVIDIA vDGA use the NVIDIA physical graphic card. Virtual machines that are configured to use NVIDIA GRID vGPU use the NVIDIA virtual graphic card, which is based on the NVIDIA physical graphic accelerator. You do not need to change the Video Memory (vRAM) settings and configuration parameters for these virtual machines.

Virtual machines that are configured to use 2D or vSGA graphics use the VMware virtual graphic card and you must change the following settings for these types of virtual machines:

- Video Memory (vRAM) settings
- Configuration parameters
- 3D Memory settings
- vCPU and virtual memory settings for performance requirement

Video Memory (vRAM) Settings

When you create a Linux virtual machine in vSphere Client, configure the vRAM size as shown in [Table 1-5](#). Set the vRAM size that is recommended for the number and resolution of the monitors that you configure for the virtual machine.

Table 1-5. Recommended vRAM Settings for 2D or vSGA Graphics

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

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.

10 MB is the minimum vRAM size recommendation for a machine that is configured with a single monitor at the lowest resolution.

You must power off the virtual machine to set the number of displays and the amount of video memory to use, as described in [“Create a Virtual Machine and Install Linux,”](#) on page 15.

Horizon Connection Server 7 does not automatically configure the vRAM settings on Linux virtual machines like it does on Windows virtual machines. You must manually configure the vRAM settings in vSphere Client.

If a Linux virtual machine is configured with a smaller vRAM size than is recommended, the following issues might occur:

- Desktop sessions might be disconnected right after the initial connection is made.
- Autofit might fail to work. The desktop is then displayed in a small area of the screen.

If a Linux virtual machine's **Number of displays** value is less than the actual required count, one or more monitors will display blank for the desktop.

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 configuration file manually. The following example specifies a vRAM size of 256 MB:

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

Configuration Parameters

To display the Linux remote desktop on multiple monitors, you must set certain configuration parameters for the virtual machine. The general steps to set a configuration parameter for a virtual machine are as follows:

- 1 Power off the virtual machine.
- 2 In the vSphere Web Client, right-click the virtual machine and select **Edit Settings**.
- 3 Click the **VM Options** tab and click **Advanced**.
- 4 Click **Edit Configuration** and then **Add Row**.
- 5 Enter the configuration parameter name and value.
- 6 Click **OK** to save the changes.

You must set the following configuration parameters:

- Set `svga.autodetect` to false.


```
svga.autodetect="false"
```
- Calculate the `svga.maxWidth` and `svga.maxHeight` values according to the number and orientation (horizontal or vertical) of the display monitors. The general rule is that the `svga.maxWidth` and `svga.maxHeight` values must be large enough to support all the displays. For example, to support 4 displays at the maximum resolution of 2560x1600, you must set the following values:


```
svga.maxHeight="3200"
svga.maxWidth="10240"
```
- For RHEL 6.8, CentOS 6.8, and Ubuntu 16.04 that are running on vSphere 6.0 or later, the maximum screen size for 2D and vSGA is 4096x4096. Set the following option to get a screen size of 8192x8192.


```
mks.enable3d = TRUE
```
- For RHEL 6.8, CentOS 6.8, and SLED 11 SP3/SP4 that are running vSphere 5.5 U3, set the following option. Do not set this option for other distributions or your virtual machine will fail to start.


```
svga.capabilitiesMask="0xFF7FFFFFFF"
```

If you have multiple monitors, you must set these configuration parameters. Otherwise, you might encounter one or more of the following problems:

- The desktop is displayed on some of the monitors and the other monitors are blank.
- A keystroke is displayed multiple times.
- The desktop becomes very slow.
- The desktop is displayed in a small area of the screen.

vCPU and Memory Settings

To improve the performance of a 2D or vSGA desktop, set more vCPUs and virtual memory for the Linux virtual machine. For example, set two vCPUs and two GB of virtual memory.

For the large screen of multiple monitors, such as four monitors, set 4 vCPUs and 4 GB of virtual memory for the virtual machine.

For the video playback purpose in a 2D or vSGA desktop, set 4 vCPUs and 4 GB of virtual memory for the virtual machine.

3D Memory Settings

To improve performance in a vSGA multiple monitor environment, set the **3D Memory** setting for the virtual machine to 1 GB or larger.

Preparing a Linux Virtual Machine for Desktop Deployment

2

Setting up a Linux desktop involves creating a Linux virtual machine and preparing the operating system for remote desktop deployment.

This chapter includes the following topics:

- [“Create a Virtual Machine and Install Linux,”](#) on page 15
- [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16
- [“Install Dependency Packages for Horizon Agent,”](#) on page 18

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 7. 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 7 for Linux,”](#) on page 10.
- 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 7 for Linux,”](#) on page 10.

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).
 Set the vRAM size in vSphere Web Client for virtual machines that use 2D or vSGA, which use the VMware driver. The vRAM size has no affect on vDGA or NVIDIA GRID vGPU machines, which use NVIDIA drivers.
 For recommended values, follow the guidelines in *System Requirements for Horizon 7 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 only as a GNOME desktop environment.
 The GNOME desktop environment is the default selection when you install the Ubuntu, SUSE, RHEL, and CentOS distribution. Do not select the KDE desktop. environment.
- 5 Ensure that the system hostname is resolvable to 127.0.0.1.

Prepare a Linux Machine for Remote Desktop Deployment

You must perform certain tasks to prepare a Linux machine for use as a desktop in a Horizon 7 deployment.

Before a Linux machine can be managed by Horizon 7, the machine must be able to communicate with Connection Server. You must configure networking on the Linux machine so that the Linux machine can ping the Connection Server instance using its FQDN (fully qualified domain name).

Open VMware Tools (OVT) are pre-installed on RHEL 7, CentOS 7, and SLES 12 machines. If you are preparing either of these machines for use as a remote desktop, you can skip steps 1 through 5 in the following procedure, which describe how to install VMware Tools by manually running the installer.

If you are using an Ubuntu16.04 machine, install OVT on it. If you are preparing this machine for use as a remote desktop, you can skip steps 1 through 5 in the following procedure and manually install OVT on your Ubuntu 16.04 machine using the following command:

```
apt-get install open-vm-tools-desktop
```

Prerequisites

- Verify that a new virtual machine (VM) 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 VM. 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.

Procedure

- 1 In vSphere Web Client or vSphere Client, mount the VMware Tools virtual disk on the VM.

- 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 On the VM, log in as 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 Map the Linux machine's host name to 127.0.0.1 in the `/etc/hosts` file.

For RHEL, CentOS, SLES, and SLED, you must manually map the host name to 127.0.0.1 because it is not automatically mapped. For Ubuntu, this step is not necessary because the mapping is there by default. This step is also not necessary when you bulk deploy desktops because the cloning process adds this mapping.

NOTE If you change the Linux machine's host name after installing Horizon Agent, you must map the new host name to 127.0.0.1 in the `/etc/hosts` file. Otherwise, the old host name will continue to be used.

- 7 For RHEL 7 and CentOS 7, verify that `virbr0` is disabled.

```
virsh net-destroy default
virsh net-undefine default
service libvirtd restart
```

- 8 Ensure that the View Connection Server instances in the pod can be resolved through DNS.
- 9 Configure the Linux machine so that the default runlevel is 5.

The runlevel must be 5 for the Linux desktop to work.

- 10 On an Ubuntu machine that was configured to authenticate with an OpenLDAP server, set the fully qualified domain name on the machine.

This step ensures that the information can be displayed correctly in the User field on the Sessions page in View Administrator. Edit the `/etc/hosts` file as follows:

- a `# nano /etc/hosts`
- b Add the fully qualified domain name. For example: `127.0.0.1 hostname.domainname hostname.`
- c Exit and save the file.

- 11 For SUSE, disable Change Hostname via DHCP. Set the hostname or domain name.

- a In Yast, click **Network Settings**.
- b Click the **Hostname/DNS** tab.
- c Deselect **Change Hostname via DHCP**.
- d Enter the hostname and the domain name.
- e Click **OK**.

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>.

Install Dependency Packages for Horizon Agent

Horizon Agent for Linux has some dependency packages unique to a Linux distribution. You must install these packages before installing Horizon Agent for Linux.

Prerequisites

Verify that a new virtual machine (VM) is created in vCenter Server and your Linux distribution is installed on the machine.

Procedure

- 1 Install the mandatory packages that are not installed or upgraded by default. The installer breaks the installation if any package does not meet the requirement.

Table 2-1. Mandatory Dependency Packages

Linux Distribution	Packages
RHEL 7 or CentOS 7 Upgrade VMware Graphic driver for 2D desktop and vSGA desktop. NOTE This option is not application for vDGA and vGPU as they use NVIDIA Graphic driver in the VM.	<pre>yum install mesa-libxatracker xorg-x11-drv-vmware mesa-private-llvm mesa-dri-drivers</pre> <p>After updating, the packages should have the following versions or later:</p> <ul style="list-style-type: none"> ■ xorg-x11-drv-vmware-13.0.2-7.20150211git8f0cf7c.el7.x86_64 ■ mesa-libxatracker-10.6.5-3.20150824.el7.x86_64 ■ mesa-private-llvm-3.6.2-2.el7.x86_64 ■ mesa-dri-drivers-10.6.5-3.20150824.el7.x86_64 <p>NOTE The update is not available for CentOS default online repository. You must enable the CentOS continuous release (CR) repository.</p> <pre>yum-config-manager --enable "CentOS-7 - cr"</pre> <p>After upgrading VMware Graphic driver, you can disable the repository again.</p> <pre>yum-config-manager --disable "CentOS-7 - cr"</pre>
SLED 11 SP3/SP4 Upgrade xorg-x11-server to a version later than 7.4.27.111.1	<pre>zypper install xorg-x11-server</pre>
SLES 12 SP1 (for Horizon 7 versions 7.0.1/7.0.2) Upgrade VMware Graphic driver, xf86-video-vmware, to a version later than 13.1.0-5.2	<pre>zypper install xf86-video-vmware</pre> <p>You must enable the OpenSUSE online repository to install it.</p> <ol style="list-style-type: none"> 1 <pre>zypper addrepo http://download.opensuse.org/distribution/leap/42.1/repo/oss/oss-42.1</pre> 2 <pre>zypper ref</pre> 3 <pre>zypper install xf86-video-vmware</pre> 4 <pre>zypper removerepo oss-42.1</pre>
SLES 12 SP1 (for Horizon 7 version 7.0.3 or later) Upgrade xf86-video-vmware to a version later than 13.0.2-3.2 from the SUSE repository	<ol style="list-style-type: none"> 1 Register SUSE 12 to enable the SUSE repositories. <pre>SUSEConnect -r <i>Registration Code</i> -e <i>Email</i></pre> 2 Update the xf86-video-vmware version. <pre>zypper install xf86-video-vmware</pre>

Table 2-1. Mandatory Dependency Packages (Continued)

Linux Distribution	Packages
SLES 12 SP1	<p>Install python-gobject2 is required for SLES 12 Linux desktop when you are installing Horizon Agent.</p> <ol style="list-style-type: none"> 1 Remove the OpenSUSE repositories. 2 Register SUSE 12 to enable the SUSE repositories. <pre>SUSEConnect -r <i>Registration Code</i> -e <i>Email</i></pre> <ol style="list-style-type: none"> 3 Install python-gobject2. <pre>zypper install python-gobject2</pre> <p>NOTE python-object2 on OpenSUSE is not compatible with SUSE 12.</p>
Ubuntu 14.04 Upgrade indicator-session to 12.10.5+15.04.20150327, available in https://launchpad.net/ubuntu/wily/amd64/indicator-session/12.10.5+15.04.20150327-0ubuntu1	<pre>wget http://launchpadlibrarian.net/201393830/indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb</pre> <pre>sudo dpkg -i ./indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb</pre>
Ubuntu 16.04	<pre>apt-get install python-dbus python-gobject</pre>

2 Install the optional package for Horizon Agent.

- By default, RHEL or CentOS 6.7 has glibc-2.12-1.166.el6.x86_64 installed which might cause a deadlock issue. As a result, the desktop connection is stuck. To overcome this issue, you must upgrade glibc to the latest version from an online repository.

```
sudo yum install glibc
```
- On Ubuntu 14.04 desktops with multiple monitors, gnome-session-fallback is needed to disable Compiz for better performance..

```
sudo apt-get install gnome-session-fallback
```
- On Ubuntu 16.04 desktops with multi-monitors, gnome-session-flashback is needed to disable Compiz for better performance..

```
sudo apt-get install gnome-session-flashback
```


Setting Up Active Directory Integration for Linux Desktops

3

View uses the existing Microsoft Active Directory (AD) infrastructure for user authentication and management. You can integrate the Linux desktops with Active Directory so that users can log in to a Linux desktop using their Active Directory user account.

This chapter includes the following topics:

- [“Integrating Linux with Active Directory,”](#) on page 21
- [“Setting Up Single Sign-on and Smart Card Redirection,”](#) on page 22

Integrating Linux with Active Directory

Multiple solutions exist to integrate Linux with Active Directory (AD).

The following solutions are known to work in a View environment:

- OpenLDAP Pass-Through Authentication
- Winbind

At a high level, the OpenLDAP pass-through authentication solution involves the following steps:

- Configure the OpenLDAP server to delegate password verification to a separate process such as `saslauthd`, which can perform password verification against Active Directory.
- Configure the Linux desktops to authenticate users with OpenLDAP.

If you plan to bulk deploy Linux desktops, you can set up the template virtual machine (VM) to run the final AD integration task. Be aware of the following considerations:

- The OpenLDAP solution works for cloned VMs without any additional steps.
- With the Winbind solution, the step to join the domain will fail because each cloned VM has a different host name. Each cloned VM needs to run the following command to rejoin the domain:

```
sudo /usr/bin/net ads join -U <domain user>%<domain password>
```

Use the following options to run the domain join command on cloned virtual machine for winbind solution:

- Remote connect such as SSH or vSphere PowerCLI to each virtual machine and run the command. For more information on scripts, see [Chapter 8, “Bulk Deployment of Horizon 7 for Manual Desktop Pools,”](#) on page 57.
- Include the command to a shell script and specify the script path to Horizon agent option `RunOnceScript` in `/etc/vmware/viewagent-custom.conf`. For more information, see [“Setting Options in Configuration Files on a Linux Desktop,”](#) on page 43.

For more information about bulk deploying Linux desktops, see [Chapter 8, “Bulk Deployment of Horizon 7 for Manual Desktop Pools,”](#) on page 57.

Setting Up Single Sign-on and Smart Card Redirection

To set up single sign-on (SSO) and smart card redirection, you must perform some configuration steps.

Single Sign-on

The Horizon View single sign-on module talks to PAM (pluggable authentication modules) in Linux and does not depend on the method that you use to integrate Linux with Active Directory (AD). Horizon View SSO is known to work with the OpenLDAP and Winbind solutions that integrate Linux with AD.

By default, SSO assumes that AD's sAMAccountName attribute is the login ID. To ensure that the correct login ID is used for SSO, you must perform the following configuration steps if you use the OpenLDAP or Winbind solution:

- For OpenLDAP, set sAMAccountName to uid.
- For Winbind, add the following statement to the configuration file `/etc/samba/smb.conf`.

```
winbind use default domain = true
```

If users must specify the domain name to log in, you must set the `SSOUserFormat` option on the Linux desktop. For more information, see [“Setting Options in Configuration Files on a Linux Desktop,”](#) on page 43. Be aware that SSO always uses the short domain name in upper case. For example, if the domain is `mydomain.com`, SSO will use `MYDOMAIN` as the domain name. Therefore, you must specify `MYDOMAIN` when setting the `SSOUserFormat` option. Regarding short and long domain names, the following rules apply:

- For OpenLDAP, you must use short domain names in upper case.
- Winbind supports both long and short domain names.

AD supports special characters in login names but Linux does not. Therefore, do not use special characters in login names when setting up SSO.

In AD, if a user's `UserPrincipalName` (UPN) attribute and `sAMAccount` attribute do not match and the user logs in with the UPN, SSO will fail. The workaround is for the user to log in using the name that is stored in `sAMAccount`.

View does not require the user name to be case-sensitive. You must ensure that the Linux operating system can handle case-insensitive user names.

- For Winbind, the user name is case-insensitive by default.
- For OpenLDAP, Ubuntu uses NSCD to authenticate users and is case-insensitive by default. RHEL and CentOS use SSSD to authenticate users and the default is case-sensitive. To change the setting, edit the file `/etc/sss/sss.conf` and add the following line in the `[domain/default]` section:

```
case_sensitive = false
```

Smart Card Redirection

To set up smart card redirection, first follow the instructions from the Linux distributor and from the smart card vendor. Then update the `pcsc-lite` package to 1.7.4. For example, run the following commands:

```
#yum groupinstall "Development tools"
#yum install libudev-devel
#service pcscd stop
#wget https://alioth.debian.org/frs/download.php/file/3598/pcsc-lite-1.7.4.tar.bz2
#tar -xjvf pcsc-lite-1.7.4.tar.bz2
#cd ./pcsc-lite-1.7.4
#./configure --prefix=/usr/ --libdir=/usr/lib64/ --enable-usbdropdir=/usr/lib64/pcsc/drivers
--enable-confdir=/etc --enable-ipcdir=/var/run --disable-libusb --disable-serial --disable-usb
```

```

--disable-libudev
#make
#make install
#service pcsd start

```

For Winbind, add the following statement to the configuration file `/etc/samba/smb.conf`.

```
winbind use default domain = true
```

When you install the Horizon Agent, you must first disable SELinux or enable permissive mode for SELinux. You must also specifically select the smart card redirection component because the component is not selected by default. For more information, see [“install_viewagent.sh Command Line Options,”](#) on page 38.

Smartcard SSO is enabled in Horizon View 7.0.1 or later. In addition, if the smart card redirection feature is installed on a virtual machine, vSphere Client's USB redirection does not work with the smart card.

Smart card redirection supports only one smart card reader. This feature does not work if two or more readers are connected to the client device.

Smart card redirection supports only one certificate on the card. If more than one certificate is on the card, the one in the first slot is used and the others are ignored. This is a Linux limitation.

NOTE

- Smartcard supports the following winbind value. Else the smartcard SSO and manual login fails.

```
winbind use default domain=true
```
 - When you use Linux client to authenticate the broker with PIV card, which is supported by Linux desktop smartcard redirection, you must add `view.sslProtocolString = "TLSv1.1"` configuration for the Linux client at `~/.vmware/view-preferences` to avoid SSL error.
-

Setting Up Graphics for Linux Desktops

4

You can configure RHEL 6.6/6.7/6.8 and 7.2 to take advantage of NVIDIA capabilities on ESXi host or on a guest operating system.

VM Clone Requirements for Setting Up 3D Graphics

You must consider the following requirements for VM Clone before setting up 3D graphics.

- For vGPU and vSGA, complete the graphic setup in the base VM. Clone the VMs. The graphic settings work for cloned VMs and no further settings are required.
- For vDGA, complete the graphic setup in the base VM. Clone the VMs. However before you power on the cloned VMs, you must remove the existing NVIDIA pass-through PCI device from the cloned VM and add the new NVIDIA pass-through PCI device to the cloned VM. NVIDIA pass-through PCI device cannot be shared between VMs. Each VM uses a dedicated NVIDIA pass-through PCI device.

This chapter includes the following topics:

- [“Configure RHEL 6.6/6.7/6.8 and RHEL 7.2 for vGPU,”](#) on page 25
- [“Configure RHEL 6.6/6.7/6.8 for vDGA,”](#) on page 30
- [“Configure RHEL 7.2 for vSGA,”](#) on page 34

Configure RHEL 6.6/6.7/6.8 and RHEL 7.2 for vGPU

You can set up an RHEL 6.6/6.7/6.8 and RHEL 7.2 to take advantage of NVIDIA vGPU (shared GPU hardware acceleration) capabilities on the ESXi host.

You must use the NVIDIA Linux VM display driver that matches the ESXi host GPU driver(.vib). See the NVIDIA Web site for information about driver packages.

IMPORTANT NVIDIA vGPU is supported on NVIDIA Maxwell M60 graphics cards and NVIDIA M6 graphics cards. This feature does not work on other NVIDIA graphics cards such as GRID K1 or K2.



CAUTION Before you begin, verify that Horizon Agent is not installed on the Linux virtual machine. If you install Horizon Agent before you configure the machine to use NVIDIA vGPU, required configuration parameters in the `xorg.conf` file are overwritten, and NVIDIA vGPU does not work. You must install Horizon Agent after the NVIDIA vGPU configuration is completed.

Install the VIB for the NVIDIA GRID vGPU Graphics Card on the ESXi Host

You must download and install the VIB for your NVIDIA GRID graphics card on the ESXi 6.0 U1 or later host.

NVIDIA provides a vGPU software package that includes a vGPU Manager, which you install on the ESXi host in this procedure, and a Linux Display Driver, which you will install on the Linux virtual machine in a later procedure.

Prerequisites

- Verify that vSphere 6.0 U1 or a later release is installed in your environment.
- Verify that the NVIDIA Maxwell M60 GPUs or M6 GPUs are installed on the ESXi host.

Procedure

- 1 Download the VIB for your NVIDIA GRID vGPU graphics card from the [NVIDIA Driver Downloads](#) site.

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

Option	Description
Product Type	GRID
Product Series	Select NVIDIA GRID vGPU .
Product	Select the version (such as GRID K2) that is installed on the ESXi host.
Operating System	Select the VMware vSphere ESXi version.

- 2 Uncompress the vGPU software package .zip file.
- 3 Upload the vGPU Manager folder to the ESXi 6.0 U1 host.

NOTE You will install the Linux Display Driver on the Linux virtual machine in a later procedure.

- 4 Power off or suspend all virtual machines on the ESXi host.
- 5 Connect to the ESXi host using SSH.
- 6 Stop the xorg service.

```
# /etc/init.d/xorg stop
```

- 7 Install the NVIDIA VIB.

For example:

```
# esxcli system maintenanceMode set --enable true
# esxcli software vib install -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```

- 8 Reboot or update the ESXi host.

- ◆ For an installed ESXi host, reboot the host.
- ◆ For a stateless ESXi host, take the following steps to update the host. (These steps also work on an installed host.)

```
Update vmkdevmgr:
# kill -HUP $(cat /var/run/vmware/vmkdevmgr.pid)
```

```
Wait for the update to complete:
# localcli --plugin-dir /usr/lib/vmware/esxcli/int deviceInternal bind
```

```
This is a new requirement with the NVIDIA 352.* host driver:
# /etc/init.d/nvidia-vgpu start
```

```
Restart xorg, which is used for GPU assignment:
# /etc/init.d/xorg start
```

- 9 Verify that the xorg service is running after the host is restarted.

Configure a Shared PCI Device for vGPU on the Linux Virtual Machine

To use NVIDIA vGPU, you must configure a shared PCI device for the Linux virtual machine.

Prerequisites

- Verify that the Linux virtual machine is prepared for use as a desktop. See [“Create a Virtual Machine and Install Linux,”](#) on page 15 and [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16.
- Verify that Horizon Agent is not installed on the Linux virtual machine.
- Verify that the NVIDIA VIB is installed on the ESXi host. See [“Install the VIB for the NVIDIA GRID vGPU Graphics Card on the ESXi Host,”](#) on page 26.
- Familiarize yourself with the virtual GPU types that are available with NVIDIA vGPU, which you select with the **GPU Profile** setting. The virtual GPU types provide varying capabilities on the physical GPUs installed on the ESXi host. See [“NVIDIA Virtual GPU Types,”](#) on page 27.

Procedure

- 1 Power off the virtual machine.
- 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 **Shared PCI Device**.
- 4 Click **Add** and select **NVIDIA GRID vGPU** from the drop-down menu.
- 5 For the **GPU Profile** setting, select a virtual GPU type from the drop-down menu.
- 6 Click **Reserve all memory** and click **OK**.

You must reserve all virtual machine memory to enable the GPU to support NVIDIA GRID vGPU.

- 7 Power on the virtual machine.

NVIDIA Virtual GPU Types

With the **GPU Profiles** setting on the Virtual Hardware page in vSphere Web Client, you can select a virtual GPU type that provides specific capabilities on the physical NVIDIA GPU on the ESXi host.

On Linux virtual machines, NVIDIA GRID vGPU is supported on NVIDIA Maxwell M60 GPUs or NVIDIA M6 GPUs.

Table 4-1. Virtual GPU Types Available for NVIDIA GRID M60 vGPU on Linux Virtual Machines

Virtual GPU Type	Physical Board	Physical GPUs	FB Per Virtual GPU	Display Heads	Maximum Resolution	Maximum Virtual GPUs Per Physical GPU	Maximum Virtual GPUs Per Physical Board
GRID M60-0q	GRID M60	two	512M	2	2560x1600	16	32
GRID M60-1q	GRID M60	two	1G	2	2560x1600	8	16

Table 4-1. Virtual GPU Types Available for NVIDIA GRID M60 vGPU on Linux Virtual Machines (Continued)

Virtual GPU Type	Physical Board	Physical GPUs	FB Per Virtual GPU	Display Heads	Maximum Resolution	Maximum Virtual GPUs Per Physical GPU	Maximum Virtual GPUs Per Physical Board
GRID M60-2q	GRID M60	two	2G	4	2560x1600	4	8
GRID M60-4q	GRID M60	two	4G	4	3840x2160	2	4
GRID M60-8q	GRID M60	two	8G	4	3840x2160	1	2

Table 4-2. Virtual GPU Types Available for NVIDIA GRID M6 vGPU on Linux Virtual Machines

Virtual GPU Type	Physical Board	Physical GPUs	FB Per Virtual GPU	Display Heads	Maximum Resolution	Maximum Virtual GPUs Per Physical GPU	Maximum Virtual GPUs Per Physical Board
GRID M6-0q	GRID M6	one	512M	2	2560x1600	16	16
GRID M6-1q	GRID M6	one	1G	2	2560x1600	8	8
GRID M6-2q	GRID M6	one	2G	4	2560x1600	4	4
GRID M6-4q	GRID M6	one	4G	4	3840x2160	2	2
GRID M6-8q	GRID M6	one	8G	4	3840x2160	1	1

Install the NVIDIA GRID vGPU Display Driver

To install the NVIDIA GRID vGPU display driver, you must disable the default NVIDIA driver, download the NVIDIA display drivers, and configure the PCI device on the virtual machine.

Prerequisites

- Verify that you downloaded the vGPU software package from the NVIDIA download site, uncompressed the package, and have the Linux Display Driver (a package component) ready. See [“Install the VIB for the NVIDIA GRID vGPU Graphics Card on the ESXi Host,”](#) on page 26.

Also verify that a shared PCI device was added to the virtual machine. See [“Configure a Shared PCI Device for vGPU on the Linux Virtual Machine,”](#) on page 27

Procedure

- 1 Disable and blacklist the default NVIDIA Nouveau driver.

- a Edit the `grub.conf` file.

For RHEL 6.6/6.7/6.8, the file is `/boot/grub/grub.conf`. For RHEL 7.2, the file is `/etc/default/grub.conf`.

RHEL Version	Command
6.6/6.7/6.8	<code>sudo vi /boot/grub/grub.conf</code>
7.2	<code>sudo vi /etc/default/grub.conf</code>

- 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 Copy the NVIDIA Linux Display Driver to the virtual machine.
- 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-$(uname -r)
sudo yum install kernel-headers-$(uname -r)
```
- 7 Add an executable flag to the NVIDIA GRID vGPU driver package.


```
chmod +x NVIDIA-Linux-x86_64-version-grid.run
```
- 8 Start the NVIDIA GRID vGPU installer.


```
sudo ./NVIDIA-Linux-x86_64-version-grid.run
```
- 9 Accept the NVIDIA software license agreement and select **Yes** to automatically update the X configuration settings.

What to do next

Install Horizon Agent on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.

Create a desktop pool that contains the configured Linux virtual machines. See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a RHEL 6.6/6.7/6.8 virtual machine by displaying the NVIDIA driver output in a View desktop session.

Prerequisites

- Check that you installed the NVIDIA display driver.
- Verify that Horizon Agent is installed on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.
- Verify that the Linux virtual machine is deployed in a desktop pool. See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

Procedure

- 1 Restart the Linux virtual machine.

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

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

- 2 From Horizon Client, connect to the Linux desktop.
- 3 In the Linux desktop session, verify that the NVIDIA display 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 graphics capabilities on the remote desktop.

After verifying the installation of NVIDIA display driver, perform the following tasks for installation to work correctly.

- If you upgrade the Linux kernel, Horizon Agent might not be able to communicate with View Connection Server. To resolve the problem, reinstall the NVIDIA driver.
- Set the NVIDIA GRID licensing in the Linux VM. See NVIDIA documentation for more information. Linux desktop will not work correctly if licensing is not set. For example, auto-fit will not work.

Configure RHEL 6.6/6.7/6.8 for vDGA

You can set up an RHEL 6.6/6.7/6.8 guest operating system so that Horizon 7 for Linux desktop can take advantage of vDGA capabilities on the ESXi host.



CAUTION Before you begin, verify that Horizon Agent is not installed on the Linux virtual machine. If you install Horizon Agent before you configure the machine to use vDGA, required configuration parameters in the `xorg.conf` file are overwritten, and vDGA does not work. You must install Horizon Agent after the vDGA configuration is completed.

Enable DirectPath I/O for NVIDIA GRID 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**.

- 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.

- Select the NVIDIA GRID GPUs and click **OK**.
The PCI devices are added to the table, DirectPath I/O PCI Devices Available to VMs.
- 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 6.6/6.7/6.8 Virtual Machine

To configure a RHEL 6.6/6.7/6.8 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. See [“Create a Virtual Machine and Install Linux,”](#) on page 15 and [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16.
- Verify that Horizon Agent is not installed on the Linux virtual machine.
- 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 on a Host,”](#) on page 30.

Procedure

- Log in to the RHEL 6.6/6.7/6.8 guest operating system as a local user configured with sudo rights.
- In vSphere Web Client, select the virtual machine and, under the **VM Hardware** tab, click **Edit Settings**.
- In the **New device** menu, select **PCI Device**.
- Click **Add** and select the PCI device from the drop-down menu.
- Click **Reserve all memory** and click **OK**.

You must reserve all virtual machine memory to enable the GPU to support vDGA.

- Power on the virtual machine and open vSphere console to connect to the machine.
- 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 the NVIDIA Display Driver for vDGA

To install the NVIDIA display driver for vDGA, you must disable the default NVIDIA driver, download the NVIDIA display drivers, and configure the PCI device on the virtual machine.

Prerequisites

- Verify that the PCI device was added to the RHEL 6.6/6.7/6.8 virtual machine. See [“Add a vDGA Pass-Through Device to a RHEL 6.6/6.7/6.8 Virtual Machine,”](#) on page 31.

Procedure

- 1 Disable and blacklist the default NVIDIA Nouveau driver.

- a Edit the `grub.conf` file.

For RHEL 6.6/6.7/6.8, the file is `/boot/grub/grub.conf`. For RHEL 7.2, the file is `/etc/default/grub.conf`.

RHEL Version	Command
6.6/6.7/6.8	<code>sudo vi /boot/grub/grub.conf</code>
7.2	<code>sudo vi /etc/default/grub.conf</code>

- 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 driver from the [NVIDIA Driver Downloads](#) site.

Select the appropriate driver version from the NVIDIA 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-$(uname -r)
sudo yum install kernel-headers-$(uname -r)
```

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

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

- 8 Start the NVIDIA installer.

```
sudo ./NVIDIA-Linux-x86_64-version.run
```

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

What to do next

Install Horizon Agent on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.

Create a desktop pool that contains the configured Linux virtual machines. See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a RHEL 6.6/6.7/6.8 virtual machine by displaying the NVIDIA driver output in a View desktop session.

Prerequisites

- Check that you installed the NVIDIA display driver.
- Verify that Horizon Agent is installed on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.
- Verify that the Linux virtual machine is deployed in a desktop pool. See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

Procedure

- 1 Restart the Linux virtual machine.

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

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

- 2 From Horizon Client, connect to the Linux desktop.
- 3 In the Linux desktop session, verify that the NVIDIA display 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 graphics capabilities on the remote desktop.

After verifying the installation of NVIDIA display driver, perform the following tasks for installation to work correctly.

- If you upgrade the Linux kernel, Horizon Agent might not be able to communicate with View Connection Server. To resolve the problem, reinstall the NVIDIA driver.
- Set the NVIDIA GRID licensing in the Linux VM. See NVIDIA documentation for more information. Linux desktop will not work correctly if licensing is not set. For example, auto-fit will not work.

Configure RHEL 7.2 for vSGA

You can set up an RHEL 7.2 guest operating system so that a Horizon 7 for Linux desktop can take advantage of vSGA capabilities.

Install the VIB for the NVIDIA Graphics Card for vSGA on the ESXi Host

You must download and install the VIB for your NVIDIA GRID graphics card on the ESXi 6.0 U1 or later host.

NVIDIA provides a VMware vSphere ESXi Driver for vSGA. For vSGA, an NVIDIA display driver is not installed on the Linux virtual machine.

Prerequisites

- Verify that vSphere 6.0 U1 or a later release is installed in your environment.
- Verify that NVIDIA driver is installed in your environment.
- Verify that the NVIDIA GRID K1 or K2 GPUs are installed on the ESXi host.

Procedure

- 1 Download the VIB for your NVIDIA GRID vGPU graphics card from the [NVIDIA Driver Downloads](#) site.

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

Option	Description
Product Type	GRID
Product Series	Select GRID Series .
Product	Select the version (such as GRID K2) that is installed on the ESXi host.
Operating System	Select the VMware vSphere ESXi version.

- 2 Upload the VMware vSphere ESXi Driver for vSGA to the ESXi 6.0 U1 host.
- 3 Power off or suspend all virtual machines on the ESXi host.
- 4 Connect to the ESXi host using SSH.

- 5 Stop the xorg service.

```
# /etc/init.d/xorg stop
```

- 6 Install the NVIDIA VIB.

For example:

```
# esxcli system maintenanceMode set --enable true
# esxcli software vib install -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```

- 7 Restart xorg, which is used for GPU assignment:

```
# /etc/init.d/xorg start
```

- 8 Reboot the ESXi host.

- 9 Verify that the xorg service is running after the host is restarted.

Configure 3D Capabilities for vSGA on the Linux Virtual Machine

To configure a RHEL 7.2 virtual machine to use vSGA, you must configure 3D settings for the virtual machine's video card in vSphere Web Client.

Prerequisites

- Verify that the Linux virtual machine is prepared for use as a desktop, Horizon Agent is installed, and the machine is deployed in a desktop pool.
- Verify that the NVIDIA VIB is installed on the ESXi host. See [“Install the VIB for the NVIDIA Graphics Card for vSGA on the ESXi Host,”](#) on page 34.

Procedure

- 1 Power off the virtual machine.
- 2 In vSphere Web Client, select the virtual machine and, under the **VM Hardware** tab, click **Edit Settings**.
- 3 In the Virtual Hardware tab, click **Video card** to expand the menu settings.
- 4 Set the **Total video memory** to 128 MB.
- 5 For **3D Graphics**, select **Enable 3D Support**.
- 6 For **3D Renderer**, select **Hardware** from the drop-down menu.
- 7 For **3D Memory**, select a suitable value for your application requirements.
If your users connect to more than 3 monitors, set this value to at least 1024 MB.
- 8 Click **OK**.
- 9 Power on the virtual machine.

What to do next

Verify that vSGA is running on the Linux virtual machine.

Next, install Horizon Agent on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.

Verify that vSGA is Running on a Linux Virtual Machine

You can verify that vSGA is running on a RHEL 7.2 virtual machine by checking the virtual machine log file and checking the guest operating system.

Procedure

- 1 Open the `vmware.log` file for the virtual machine.

If a supported GPU and NVIDIA VIB are installed correctly, the log file displays lines such as those in the following example:

```
2015-06-24T22:19:25.259Z| mks| I120: OpenGL Version: "4.0.0 NVIDIA 346.69" (4.0.0)
2015-06-24T22:19:25.259Z| mks| I120: GLSL Version: "4.00 NVIDIA" (4.00.0)
2015-06-24T22:19:25.259Z| mks| I120: OpenGL Vendor: "NVIDIA Corporation"
2015-06-24T22:19:25.259Z| mks| I120: OpenGL Renderer: "Quadro 4000/PCIe/SSE2"
```

If a supported GPU and NVIDIA VIB are not installed correctly, the virtual machine uses the Software Renderer. The `vmware.log` file displays lines such as those in the following example:

```
2015-07-06T17:09:26.423Z| vmx| I120: [msg.mks.noGPUResourceFallback] Hardware GPU resources
are not available. The virtual machine uses software rendering.
2015-07-06T17:09:26.423Z| vmx| I120: -----
2015-07-06T17:09:26.425Z| svga| I120: MKS-SWP: plugin started - llvmpipe (LLVM 3.3, 256 bits)
2015-07-06T17:09:26.426Z| svga| I120: Started Shim3D
2015-07-06T17:09:26.426Z| svga| I120: MKS-RenderMain: Starting SWRenderer
```

- 2 In the guest operating system on the virtual machine, type the following command.

```
glxinfo|grep Gallium
```

If vSGA is working, the command returns the following text:

```
OpenGL renderer string: Gallium 0.4 on SVGA3D; build : RELEASE;
```

If vSGA is not working correctly, the command returns the following text:

```
OpenGL renderer string: Gallium 0.4 on llvmpipe (LLVM 3.3, 256 bits)
```

What to do next

Install Horizon Agent on the Linux virtual machine. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.

Installing Horizon Agent

You must install Horizon Agent on the Linux desktops so that View Connection Server can communicate with and manage the desktops.

This chapter includes the following topics:

- [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37
- [“Configure the Certificate for Linux Agent,”](#) on page 39
- [“Upgrading the Horizon Agent on a Linux Virtual Machine,”](#) on page 40
- [“Uninstall Horizon 7 for Linux Machines,”](#) on page 41

Install Horizon Agent on a Linux Virtual Machine

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

With Horizon 7.0.1 release, Horizon Agent for Linux uses vCenter managed virtual machines. The managed virtual machines provide the following enhancements.

- vCenter is a mandatory requirement for Linux desktop deployment.
- Horizon Agent installation on Linux does not require registration.
- For a large number of Linux desktop deployment, you can install the Horizon Agent on the base virtual machine.



CAUTION If you intend to use NVIDIA GRID vGPU, vDGA, or vSGA, you must configure these 3D features on the Linux virtual machine before you install Horizon Agent. If you install Horizon Agent first, required parameters in the `xorg.conf` file are overwritten, and the 3D graphics features do not work.

See [“Configure RHEL 6.6/6.7/6.8 and RHEL 7.2 for vGPU,”](#) on page 25, [“Configure RHEL 6.6/6.7/6.8 for vDGA,”](#) on page 30, or [“Configure RHEL 7.2 for vSGA,”](#) on page 34. Install Horizon Agent after the 3D graphics configuration is completed.

For 2D graphics configuration, you can install Horizon Agent after you complete the steps in [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16.

Prerequisites

- Verify that the Linux guest operating system is prepared for desktop use. See [“Prepare a Linux Machine for Remote Desktop Deployment,”](#) on page 16.
- Familiarize yourself with the Horizon Agent installer script for Linux. See [“install_viewagent.sh Command Line Options,”](#) on page 38.

Procedure

- 1 Download the Horizon 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 7 download, which includes the Horizon Agent for Linux installer.
 The installer filename is VMware-horizonagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz for 64-bit Linux where y.y.y is the version number and xxxxxx is the build number.
- 2 Unpack the tarball for your Linux distribution on the guest operating system.
 For example:

```
tar -xzf <View Agent tar ball>
```
- 3 Navigate to the tar ball folder.
- 4 Run the install_viewagent.sh script as superuser.
 See “install_viewagent.sh Command Line Options,” on page 38 for a list of the command line options.
 For example:

```
sudo ./install_viewagent.sh
```
- 5 Type **Yes** to accept the EULA if you run install_viewagent.sh without specifying the -A option.
 The installer does not run unless you accept the EULA.
- 6 Reboot Linux for the changes to take effect.

After installation, the *viewagent* service is started. Verify that the service is started using `sudo service viewagent status`.

What to do next

Deploy the virtual machine in a desktop pool. See “Create a Manual Desktop Pool for Linux,” on page 49.

install_viewagent.sh Command Line Options

The `install_viewagent.sh` script installs Horizon 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 5-1. install_viewagent.sh Optional but Required Parameter

Optional Parameter (Required Information)	Description
-A yes no	Accept or Refuse EULA. You must specify yes for the install to proceed.

Table 5-2. install_viewagent.sh Optional Parameters

Optional Parameters	Description
-a yes no	Install or bypass audio input support. Default is no .
-s	Self signed cert subject DN. By default, installer uses Blast.

Table 5-2. `install_viewagent.sh` Optional Parameters (Continued)

Optional Parameters	Description
-m yes no	Install or Bypass smartcard redirection support. Default is no .
-j	JMS SSL keystore password. By default, installer generates a random string.
-r yes no	Restart the system automatically after installation. Default is no .
-M yes no	Upgrade the Linux Agent to managed or un-managed agent. Default is yes .

Table 5-3. Examples of `install_viewagent.sh` Parameters

Condition	Examples
Fresh Installation	sudo ./install_viewagent.sh -A yes Fresh installation always requires new desktop pool creation.
Upgrade from unmanaged virtual machine and retain the unmanaged virtual machine style	sudo ./install_viewagent.sh -A yes -M no This type of upgrade does not require new desktop pool creation. You can reuse the existing desktop pool .
Upgrade from unmanaged virtual machine deployment and convert to managed virtual machine style. The upgrade requires new desktop pool creation on broker	sudo ./install_viewagent.sh -A yes This type of upgrade requires a new desktop pool creation. You must delete the existing desktop pool.

Configure the Certificate for Linux Agent

When you install Linux Agent, the installer generates a self-signed certificate for VMwareBlastServer.

- When the Blast Security Gateway is disabled on the broker, VMwareBlastServer presents this certificate to the browser that uses HTML Access to connect to the Linux Desktop.
- When the Blast Security Gateway is enabled on the broker, Blast Security Gateway's certificate presents the certificate to the browser.

To comply with industry or security regulations, you can replace the self-signed certificate with a certificate that is signed by a Certificate Authority (CA).

Procedure

- 1 Install the private key and the certificate to VMwareBlastServer.
 - a Rename the private key to `ru1.key` and the certificate to `ru1.crt` .
 - b Run `sudo chmod 550 /etc/vmware/ssl`.
 - c Copy the `ru1.crt` and `ru1.key` to `/etc/vmware/ssl`.
 - d Run `chmod 440 /etc/vmware/ssl`.

- 2 Install the root and intermediate Certificate Authority into the Linux OS Certificate Authority store.

NOTE Check your Linux distribution documentation for the Linux system settings change.

Upgrading the Horizon Agent on a Linux Virtual Machine

You can upgrade Horizon Agent on a Linux machine by installing the latest version of Horizon Agent.

Unmanaged virtual machine: The agent installer registers the machine to the broker which requires broker admin information. The Desktop Pool Creation wizard uses **Other Sources** in Machine Source page to select the registered machine.

Managed virtual machine: The installer does not communicate with the broker. The Desktop Pool Creation wizard uses **vCenter virtual machines** in the Machine Source page to select the virtual machines through vCenter. The managed virtual machine deployment supports the following functions.

- Remote Machine Power Policy
- Allow users to reset their machines

NOTE Horizon Agent for Linux 7.0.0 and earlier versions functioned as unmanaged virtual machines. The Horizon Agent for Linux 7.0.1 functions as managed virtual machine support.

You can use the following methods to upgrade from unmanaged to managed virtual machine deployment.

- Retain the unmanaged virtual machine deployment and upgrade to the required version. This type of upgrade does not require any configuration modifications in View Connection Server.
- Upgrade from unmanaged virtual machine deployment to managed virtual machine deployment to any version. This type of upgrade requires new desktop pool creation on the View Connection Server.

NOTE For the upgrade from managed virtual machine deployment, you can retain the managed virtual machine deployment and upgrade to the required version. However, to convert the managed virtual machine deployment to an unmanaged virtual machine deployment during an upgrade is not supported.

The following parameters are available for upgrade.

Parameter	Description
-A yes	EULA acceptance. You must specify yes for the install to proceed. If this parameter is not specified, the install script prompts for the value.
-m yes	Enable smart card redirection.
-r	Reboot the operating system after installation. The allowed values are yes and no . The default is no .
-M yes no	Upgrade the Linux Agent to managed unmanaged agent. The default value is yes .

Upgrade Horizon Agent on a Linux Virtual Machine

You can upgrade Horizon Agent on a Linux machine by installing the latest version of Horizon Agent..

Prerequisites

- Verify that the VMwareBlastServer process is not running.

To stop this process, ensure that the user logs off the machine and no desktop session is active, or reboot the machine.

Procedure

- 1 Download the latest Horizon 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 7 download, which includes the Horizon Agent for Linux installer.

The installer filename is VMware-viewagent-linux-x86_64-y.y.y-xxxxxx.tar.gz for 64-bit Linux where y.y.y is the version number and xxxxxx is the build number.

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

For example:

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

- 3 Navigate to the tar ball folder.
- 4 Run the `install_viewagent.sh` script to upgrade unmanaged virtual machines using one of the following deployment scenarios:

Option	Description
Upgrade unmanaged virtual machine deployment and retain the unmanaged virtual machine deployment	<code>sudo ./install_viewagent.sh -A yes -M no</code>
Upgrade unmanaged virtual machine deployment and change it to managed virtual machine deployment	<code>sudo ./install_viewagent.sh -A yes -M yes</code> NOTE In View Administrator, delete the existing desktop pool for unmanaged virtual machine deployment and create a new desktop pool for managed virtual machine deployment. For more info, see “Create a Manual Desktop Pool for Linux,” on page 49.
Upgrade a managed virtual machine deployment	<code>sudo ./install_viewagent.sh -A yes -M yes</code> NOTE After upgrading, your existing desktop pool can be reused.

Uninstall Horizon 7 for Linux Machines

To uninstall Horizon 7 for Linux on a virtual machine, you must uninstall Horizon Agent and remove configuration files.

Prerequisites

Verify that the VMwareBlastServer process is not running. To stop this process, ensure that you log off the machine and no desktop session is active, or reboot the machine.

Procedure

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

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

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

- 2 Manually delete the Horizon 7 for Linux configuration files at `/etc/vmware` directory.

Configuration Options for Linux Desktops

6

You can configure various options to customize the user experience using configuration files.

This chapter includes the following topics:

- [“Setting Options in Configuration Files on a Linux Desktop,”](#) on page 43
- [“Example Blast Settings for Linux Desktops,”](#) on page 46
- [“Suppress the vSphere Console Display of a Linux Desktop,”](#) on page 47

Setting Options in Configuration Files on a Linux Desktop

You can configure certain options by adding entries to the files `/etc/vmware/config` or `/etc/vmware/viewagent-custom.conf`.

During the installation of View Agent or Horizon Agent, the installer copies two configuration template files, `config.template` and `viewagent-custom.conf.template`, to `/etc/vmware`. In addition, if the files `/etc/vmware/config` and `/etc/vmware/viewagent-custom.conf` do not exist, the installer copies `config.template` to `config` and `viewagent-custom.conf.template` to `viewagent-custom.conf`. In the template files, all the configuration options are listed and documented. To set an option, simply remove the comment and change the value as appropriate.

For example, the following line in `/etc/vmware/config` enables the lossless PNG mode.

```
RemoteDisplay.alwaysLossless=TRUE
```

After you make configuration changes, reboot Linux for the changes to take effect.

Configuration Options in `/etc/vmware/config`

VMwareBlastServer and its related plug-ins use the configuration file `/etc/vmware/config`.

Table 6-1. Configuration Options in `/etc/vmware/config`

Option	Value	Default	Description
<code>RemoteDisplay.alwaysLossless</code>	TRUE or FALSE	FALSE	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. This feature uses additional bandwidth between the client and the ESXi host.
<code>mksVNCServer.useUIInputButtonMapping</code>	TRUE or FALSE	FALSE	Set this option to enable the support of a left-handed mouse on Ubuntu or RHEL 7. CentOS and RHEL 6.6/6.7/6.8 support a left-handed mouse and you do not need to set this option.

Table 6-1. Configuration Options in `/etc/vmware/config` (Continued)

Option	Value	Default	Description
RemoteDisplay.allowAudio	TRUE or FALSE	TRUE	Set this option to disable audio out
VVC.ScRedir.Enable	TRUE or FALSE	TRUE	Set this option to disable smart card redirection.
VVC.logLevel	FATAL, ERROR, WARN, INFO, DEBUG, or TRACE	INFO	Use this option to set the log level of the VVC proxy node.
VVC.RTAV.Enable	TRUE or FALSE	TRUE	Set this option to disable audio input.
Clipboard.Direction	0, 1, 2, or 3	2	This option determines the clipboard redirection policy. <ul style="list-style-type: none"> ■ 0 - Disable clipboard redirection. ■ 1 - Enable clipboard redirection in both directions. ■ 2 - Enable clipboard redirection from client to remote desktop only. ■ 3 - Enable clipboard redirection from remote desktop to client only.
mksVNCServer.useXExtButton Mapping	TRUE or FALSE	FALSE	Set this option to enable or disable the support of a left-handed mouse on SLED 11 SP3.
mksvhan.clipboardSize	INTEGER	1024	Use this option to specify the clipboard maximum size to copy and paste.
RemoteDisplay.maxBandwidth Kbps	INTEGER	409600	Specifies the maximum bandwidth in kilobits per second (kbps) for a VMware Blast session. The bandwidth includes all imaging, audio, virtual channel, and VMware Blast control traffic. The max value is 4 Gbps (4096000).
RemoteDisplay.maxFPS	INTEGER	60	Specifies the maximum rate of screen updates. Use this setting to manage the average bandwidth that users consume. Valid value should be between 3 and 60. The default is 60 updates per second.
RemoteDisplay.enableStats	TRUE or FALSE	FALSE	Enable or Disable the Blast protocol statistics in mks log, such as bandwidth, FPS, RTT and so on.
RemoteDisplay.allowH264	TRUE or FALSE	TRUE	Set this option to enable or disable H.264 Encoding.
vdpservice.log.logLevel	FATAL, ERROR, WARN, INFO, DEBUG, or TRACE	INFO	Use this option to set the log level of the vdp service.
RemoteDisplay.qpmaxH264	available range of values: 0-51	36	Use this option to set the H264minQP quantization parameter, which specifies the best image quality for the remote display configured to use H.264 encoding. Set the value to greater than the value set for RemoteDisplay.qpminH264.
RemoteDisplay.qpminH264	available range of values: 0-51	10	Use this option to set the H264maxQP quantization parameter, which specifies the lowest image quality for the remote display configured to use H.264 encoding. Set the value to less than the value set for RemoteDisplay.qpmaxH264.

Table 6-1. Configuration Options in `/etc/vmware/config` (Continued)

Option	Value	Default	Description
RemoteDisplay.minQualityJPEG	available range of values: 1-100	25	Specifies the image quality of the desktop display for JPEG/PNG encoding. The low-quality settings are for areas of the screen that change often, for example, when scrolling occurs.
RemoteDisplay.midQualityJPEG	available range of values: 1-100	35	Specifies the image quality of the desktop display for JPEG/PNG encoding. Use to set the medium-quality settings of the desktop display.
RemoteDisplay.maxQualityJPEG	available range of values: 1-100	90	Specifies the image quality of the desktop display for JPEG/PNG encoding. The high-quality settings are for areas of the screen that are more static, resulting in a better image quality.

Configuration Options in `/etc/vmware/viewagent-custom.conf`

Java Standalone Agent uses the configuration file `/etc/vmware/viewagent-custom.conf`.

Table 6-2. Configuration Options in `/etc/vmware/viewagent-custom.conf`

Option	Value	Default	Description
Subnet	NULL or network address and mask in IP address/CIDR format	NULL	<p>If there are multiple local IP addresses with different subnets, use this option to set the subnet that the Linux Agent provides to the View Connection Server.</p> <p>When multiple subnet configurations are detected on a Linux Agent machine, this option is required to specify the correct subnet that should be used by the Linux Agent. For example, if you installed Docker on the Linux machine, it will be introduced as a virtual network adapter. To avoid Linux Agent from using Docker as a virtual network adapter, you have to set this option to use the real physical network adapter.</p> <p>You must specify the value in IP address/CIDR format. For example, Subnet=192.168.1.0/24.</p> <p>NULL implies that the Linux Agent randomly selects the IP address.</p>
SSOEnable	TRUE or FALSE	TRUE	Set this option to disable single sign-on (SSO).
SSOUserFormat	A text string	[username]	<p>Use this option to specify the format of the login name for single sign-on. The default is the user name only. Set this option if the domain name is also required. Typically the login name is the domain name plus a special character followed by the user name. If the special character is the backslash, you must escape it with another backslash. Examples of login name formats:</p> <ul style="list-style-type: none"> ■ SSOUserFormat=[domain]\\[username] ■ SSOUserFormat=[domain]+[username] ■ SSOUserFormat=[username]@[domain]
StartBlastServerTimeout	An integer	20	This option determines the amount of time, in seconds, that the VMwareBlastServer process has for initialization. If the process is not ready within this timeout value, the user's login will fail.

Table 6-2. Configuration Options in `/etc/vmware/viewagent-custom.conf` (Continued)

Option	Value	Default	Description
SSLCiphers	A text string	!aNULL:kECDH +AESGCM:ECDH +AESGCM:RSA +AESGCM:kECDH +AES:ECDH+AES:RSA +AES	Use this option to specify the list of ciphers. You must use the format that is defined in https://www.openssl.org/docs/manmaster/apps/ciphers.html .
SSLProtocols	A text string	TLSv1_1:TLSv1_2	Use this option to specify the security protocols. The supported protocols are TLSv1.0, TLSv1.1, and TLSv1.2.
SSLCipherServerPreference	TRUE or FALSE	TRUE	Use this option to enable or disable the option <code>SSL_OP_CIPHER_SERVER_PREFERENCE</code> . For more information, see https://www.openssl.org/docs/manmaster/ssl/SSL_CTX_set_options.html .
LogCnt	An integer	-1	Use this option to set the reserved log file count in <code>/tmp/vmware-root</code> . <ul style="list-style-type: none"> ■ -1 - keep all ■ 0 - delete all ■ > 0 - reserved log count.
RunOnceScript			Use this option to rejoin the cloned VM to AD. Set the run once script after the host name has changed. The specified script is executed only once after the first host name change. The script is executed as root permission when the agent service starts and host name has been changed since agent installation. For example, for the winbind solution, you must join the base VM to AD with winbind, and set this option to a script path. This must contain the domain rejoin command <code>/usr/bin/net ads join -U <ADUserName> %<ADUserPassword></code> . After VM Clone, the operating system customization changes the host name. When the agent service starts, the script is executed to join the cloned VM to AD.
RunOnceScriptTimeout		120	Use this option to set the timeout time in seconds for the <code>RunOnceScript</code> option. For example, set <code>RunOnceScriptTimeout=120</code>

NOTE The three security options, `SSLCiphers`, `SSLProtocols`, and `SSLCipherServerPreference` are for the `VMwareBlastServer` process. When starting the `VMwareBlastServer` process, the Java Standalone Agent passes these options as parameters. When Blast Secure Gateway (BSG) is enabled, these options affect the connection between BSG and the Linux desktop. When BSG is disabled, these options affect the connection between the client and the Linux desktop.

Example Blast Settings for Linux Desktops

You can adjust the image quality of your remote desktop display to improve the user experience. Improving image quality is helpful in maintaining a consistent user experience when there is a bad network connection.

Example VMware Blast Extreme Protocol Settings

`VMwareBlastServer` and its related plug-ins use the configuration file `/etc/vmware/config`.

Table 6-3. Example Blast Configuration Options in `/etc/vmware/config`

Option name	Parameter	High-speed LAN	LAN	Dedicated WAN	Broadband WAN	Low-speed WAN	Extremely Low speed
Bandwidth settings	RemoteDisplay.maxBandwidthKbps	1000000 (1 Gbps)	1000000 (1 Gbps)	1000000 (1 Gbps)	5000 (5 Mbps)	2000 (2 Mbps)	1000 (1 Mbps)
Max FPS	RemoteDisplay.maxFPS	60	30	30	20	15	5
Audio Playback	RemoteDisplay.allowAudio	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
Display Quality (JPEG/PNG)	RemoteDisplay.maxQualityJPEG	90	90	90	70	60	50
Display Quality (JPEG/PNG)	RemoteDisplay.midQualityJPEG	35	35	35	35	35	35
Display Quality (JPEG/PNG)	RemoteDisplay.minQualityJPEG	25	25	25	20	20	20
Display Quality (H.264)	RemoteDisplay.qpmaxH264	28	36	36	36	36	42
Display Quality (H.264)	RemoteDisplay.qpminH264	10	10	10	10	10	10

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.

Create and Manage Linux Desktop Pools

7

To configure Linux virtual machines for use as remote desktops, you must create a desktop pool with Linux virtual machines.

Horizon for Linux supports the following desktop pool types:

- Manual desktop pool with vCenter virtual machine
- Automated full-clone desktop pool

To create a manual desktop pool with a vCenter virtual machine, you must install the Horizon agent on all virtual machines. Then, use the Connection Server desktop pool creation wizard to add the virtual machines to the desktop pool. To clone a large number of virtual machines, see [“Overview of Bulk Deployment of Linux Desktops,”](#) on page 57.

To create an automated full-clone desktop pool, you must install the Horizon agent on a Linux virtual machine template. Then, use the Connection Server desktop pool creation wizard to clone full virtual machines.

This chapter includes the following topics:

- [“Create a Manual Desktop Pool for Linux,”](#) on page 49
- [“Manage Desktop Pool for Linux,”](#) on page 50
- [“Create an Automated Full-Clone Desktop Pool for Linux,”](#) on page 51
- [“Broker PowerCLI Commands,”](#) on page 53

Create a Manual Desktop Pool for Linux

You can create a manual desktop pool for Linux virtual machines.

Prerequisites

- Verify that Horizon Agent is installed on the Linux guest operating systems. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.
- Verify that vCenter Server is added to Horizon Connection Server.

Procedure

- 1 In View Administrator, add a manual desktop pool.
Select **Catalog > Desktop Pools > Add** .
- 2 Select **Manual Desktop Pool**.
- 3 On the User assignment page, select either dedicated or floating user assignments for the machines in the desktop pool and click **Next**.

- 4 On the Machine Source page, select **vCenter virtual machines** and click **Next**.
- 5 On the vCenter Server page, select the appropriate vCenter server and click **Next**.
- 6 Enter the **desktop pool id**.
- 7 On the Desktop Pool Settings page, set the following options.

Option	Description
Default display protocol	VMware Blast
Allow users to choose protocol	No
3D Renderer	Manage using vSphere Client for 2D, vSGA, or vDGA desktop and NVIDIA GRID vGPU for vGPU desktop

NOTE The pool settings are mandatory. Else, you might fail to connect to the desktop and get a protocol error or a black screen.

- 8 On the Add vCenter Virtual Machines page, select the linux virtual machine. Click **Add** and click **Next**.

NOTE Do not create Windows and Linux virtual machines in the same desktop pool.

- 9 Follow the prompts to complete the procedure.

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.

The Linux virtual machines are ready to be used as remote desktops in a Horizon 7 deployment.

Manage Desktop Pool for Linux

When you create a manual desktop pool and add Linux machines to the pool, you can manage the manual desktop pools by configuring the settings. You must add only Linux guest operating systems to the manual desktop 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.

Support for Managing Operations

- Disable or Enable desktop pool
- Clone automated desktop pool
- Delete desktop pool

You can either remove virtual machines from View Manager or delete virtual machines from the disk

Support for Remote Settings

Table 7-1. Remote Settings

Remote Setting	Options
Remote Machine Power Policy	<ul style="list-style-type: none"> ■ Take no power action ■ Ensure machines are always powered on ■ Suspend ■ Power off
Automatically logoff after disconnect	<ul style="list-style-type: none"> ■ Immediately ■ Never ■ After n minutes

Table 7-1. Remote Settings (Continued)

Remote Setting	Options
Allow users to reset their machines	<ul style="list-style-type: none"> ■ Yes ■ No
Allow user to initiate separate sessions from different client devices	<ul style="list-style-type: none"> ■ Yes ■ No
"Delete machine after logoff" for Automated Desktop Pool with Full Clone and Floating	<ul style="list-style-type: none"> ■ Yes ■ No

Support for View Administrator Operations

- Disconnect Session
- Logoff Session
- Send Message

For dedicated desktop pool, you can add or remove user assignment for each virtual machine. For large number of operations, you must use View PowerCLI Cmdlets.

- Update-UserOwnership
- Remove-UserOwnership

NOTE Do not change **Remote Display Protocol** settings. This setting must always be the same as desktop pool creation.

Setting	Option
Default display protocol	VMware Blast
Allow user to choose protocol	No
3D Renderer	<ul style="list-style-type: none"> ■ Manage using vSphere Client for 2D or vSGA or vDGA ■ NVIDIA GRID vGPU

For more information, see *View Administration* guide.

Create an Automated Full-Clone Desktop Pool for Linux

You can create an automated full-clone desktop pool for Linux virtual machines. After you create the automated full-clone desktop pool, you can use the Linux virtual machines as remote desktops in a Horizon 7 deployment.

Prerequisites

- Verify that Horizon Agent is installed on the Linux guest operating systems. See [“Install Horizon Agent on a Linux Virtual Machine,”](#) on page 37.
- If you use the Winbind solution to join the Linux virtual machine to Active Directory, you must finish configuring the Winbind solution in the virtual machine template.
- If you use the Winbind solution, you must run the domain join command on the virtual machine. Include the command in a shell script and specify the script path to the Horizon Agent option `RunOnceScript` in `/etc/vmware/viewagent-custom.conf`. For more information, see [“Setting Options in Configuration Files on a Linux Desktop,”](#) on page 43.
- Verify that vCenter Server is added to Horizon Connection Server.

Procedure

- 1 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.
Domain	Specify the domain of the View environment.
Network Settings	Use standard network settings.
Primary DNS	Specify a valid address.

NOTE For more information on Guest OS Customization Support Matrix, see <http://partnerweb.vmware.com/programs/guestOS/guest-os-customization-matrix.pdf>.

- 2 In Horizon Administrator, select **Catalog > Desktop Pools > Add**.
- 3 Select **Automated Desktop Pool** and click **Next**.
- 4 Select either dedicated or floating user assignments for the machines in the desktop pool and click **Next**.
- 5 On the vCenter Server page, select **Full virtual machines** and select the appropriate vCenter server, and click **Next**.
- 6 On the Desktop Pool Identification page, enter the desktop pool ID and click **Next**.
- 7 On the Desktop Pool Settings page, set the following options and click **Next**.

Option	Description
Default display protocol	VMware Blast
Allow users to choose protocol	No
3D Renderer	Manage using vSphere Client for 2D, vSGA, or vDGA desktop and NVIDIA GRID vGPU for vGPU desktop

NOTE The pool settings are mandatory. Else, you might fail to connect to the desktop and get a protocol error or a black screen.

- 8 On the Provisioning Settings page, set the **Virtual Machine Naming** options and click **Next**.

Option	Description
Specify names manually	Enter names manually.
Use a naming pattern	For example, specify <code>LinuxVM-{n}</code> . You must also specify the following desktop pool sizing options: <ul style="list-style-type: none"> ■ Maximum number of machines ■ Number of spare, powered-on machines

- 9 On the Storage Optimization page, select a storage management policy and click **Next**.
- 10 On the vCenter Settings page, you must click Browse and select the vCenter Server settings in sequence and click **Next**.

You cannot skip a vCenter Server setting:

- a Template

- b VM folder location
 - c Host or cluster
 - d Resource pool
 - e Datastores
- 11 On the Advanced Storage Options page, select the appropriate storage options and click **Next**.
 - 12 On the Guest Customization page, select your guest customization for Linux and click **Next**.
 - 13 On the Ready to Complete page, review the details and select **Entitle users after this wizard finishes**.
 - 14 Click **Finish**.
 - 15 To entitle users to the machines in the desktop pool, select the desktop pool and click **Entitlements > Add entitlements** and add the users and groups.
 - 16 Wait till all the Linux virtual machines in the desktop pool become available.

Broker PowerCLI Commands

The View PowerCLI cmdlets which are to perform various administration tasks on Connection Server and Windows desktop, can also work for Linux desktop.

Create a Manual Desktop Pool

```
Add-ManualPool -DefaultProtocol Blast -AllowProtocolOverride $false -threadRender usevc|vgpu -
Pool_id <pool id> [more parameters]
```

For Linux Desktop the following options and values are mandatory.

- DefaultProtocol Blast
- AllowProtocolOverride \$false
- threadRender usevc|vgpu. For vGPU desktop, use -threadRender vgpu and for 2D/vSGA/vDGA desktop -threadRender usevc .

Examples

- Create a floating Linux Desktop pool named LinuxDesktop with a virtual machine, LinuxVM-01.


```
Add-ManualPool -DefaultProtocol Blast -AllowProtocolOverride $false -threadRender usevc -
Pool_id LinuxDesktop -Id (Get-DesktopVM -Name LinuxVM-01).id -Persistence NonPersistent -
Vc_name myvc.myorg.org
```
- Create a dedicated Linux vGPU desktop pool named LinuxDesktop with all VMs that start with VM name as LinuxVM-.


```
Get-DesktopVM | Where-Object {$_.Name.StartsWith("LinuxVM-")} | Add-ManualPool -
DefaultProtocol Blast -AllowProtocolOverride $false -Persistence Persistent -threadRender
vgpu -Pool_id LinuxDesktop
```
- Create floating Linux desktop pool LinuxDesktop with the first RHEL 6 x64 VM.


```
Get-DesktopVM | Where-Object {$_.GuestID -eq "rhel6_64Guest"} | Select-Object -Index 0 |
Add-ManualPool -DefaultProtocol Blast -AllowProtocolOverride $false -Persistence
NonPersistent -threadRender usevc -Pool_id LinuxDesktop
```

Create a Full-Clone Automated Desktop Pool

```
Add-AutomaticPool -DefaultProtocol Blast -AllowProtocolOverride $false -threadRender usevc|vgpu `
-Pool_id <pool id> -Vc_id <vCenter id> `
-NamePrefix <VM Name Prefix>" `
-templatePath <Virtual Machine Template Path> `
-VmFolderPath <Virtual Machine Folder Path> `
-ResourcePoolPath <Resource Pool Path> `
-dataStorePaths <Datastore Path> `
-customizationSpecName <Customization Specification Name> `
[more parameters]
```

For Linux Desktop the following options and values are mandatory.

- DefaultProtocol Blast
- AllowProtocolOverride \$false
- threadRender usevc|vgpu. For vGPU desktop, use -threadRender vgpu and for 2D/vSGA desktop -threadRender usevc.

Example

```
Add-AutomaticPool -DefaultProtocol Blast -AllowProtocolOverride $false -threadRender usevc `
-pool_id FullClone-Linux `
-Vc_id (Get-ViewVC -serverName myvc.myorg.org).vc_id `
-NamePrefix "FullClone-{n:fixed=3}" `
-Persistence NonPersistent -deletePolicy DeleteOnUse `
-VmFolderPath "/LinuxVDI/vm/FullClone" `
-ResourcePoolPath "/LinuxVDI/host/LinuxVDICluster/Resources" `
-templatePath "/LinuxVDI/vm/LinuxTemplate" `
-dataStorePaths "/LinuxVDI/host/LinuxVDICluster/datastore" `
-customizationSpecName "linux-spec" `
-maximumCount 100
```

Add or Remove Desktop Pool Entitlement

- Entitle domain user group of domain mydomain.org to LinuxDesktop.

```
Add-PoolEntitlement -Pool_id LinuxDesktop -Sid (Get-User -Name "domain user" -Domain "mydomain.org").sid
```
- Remove entitlement of domain user group of mydomain.org domain from LinuxDesktop.

```
Remove-PoolEntitlement -Pool_id LinuxDesktop -Sid (Get-User -Name "domain user" -Domain "mydomain.org").sid
```

Assign or Remove User To or From the VM in Dedicated Desktop Pool

- Assign myuser user to LinuxVM-01 VM which is in a dedicated desktop pool.

```
Update-UserOwnership -Machine_id (Get-DesktopVM -Name "LinuxVM-01").machine_id -Sid (Get-User -Name "myuser" | Where-Object {$_.cn -eq "myuser"}).sid
```
- Remove myuser user from LinuxVM-01 VM which is in a dedicated desktop pool.

```
Remove-UserOwnership -Machine_id (Get-DesktopVM -Name "LinuxVM-01").machine_id
```

Logoff Desktop Connection

- Logoff from the desktop session of myuser.

```
Get-RemoteSession -Username "mydomain.org\myuser" | Send-SessionLogoff
```

For more information on broker PowerCLI cmdlet, see Using View PowerCLI in *View Integration*.

Bulk Deployment of Horizon 7 for Manual Desktop Pools

8

With View Administrator, you can create a pool of Windows, but not Linux, desktop machines automatically. However, 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 any responsibility for issues that might arise when you use the sample scripts.

This chapter includes the following topics:

- [“Overview of Bulk Deployment of Linux Desktops,”](#) on page 57
- [“Overview of Bulk Upgrade of Linux Desktops,”](#) on page 59
- [“Create a Virtual Machine Template for Cloning Linux Desktop Machines,”](#) on page 59
- [“Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops,”](#) on page 61
- [“Sample Script to Clone Linux Virtual Machines,”](#) on page 61
- [“Sample Script to Join Cloned Virtual Machines to AD Domain,”](#) on page 65
- [“Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH,”](#) on page 68
- [“Sample Script to Upload Configuration Files to Linux Virtual Machines,”](#) on page 71
- [“Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH,”](#) on page 75
- [“Sample Script to Upgrade Horizon Agent on Linux Desktop Machines,”](#) on page 79
- [“Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH,”](#) on page 83
- [“Sample Script to Perform Operations on Linux Virtual Machines,”](#) on page 89

Overview of Bulk Deployment of Linux Desktops

Deploying manual desktops for Linux 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 some operations, you can choose to have either PowerCLI or SSH execute the commands on the Linux machine. The following table describes the differences between the two approaches.

PowerCLI	SSH
No need to install additional tools.	<ul style="list-style-type: none"> ■ For Ubuntu, you need to install the SSH server with the command <code>sudo apt-get install openssh-server</code>. For RHEL and CentOS, openssh-server is installed by default but you need to ensure that the firewall settings allow ssh. ■ Need to download the SSH client applications <code>pscp.exe</code> and <code>plink.exe</code> and put them in the same folder as the PowerCLI scripts.
Uploading files and command execution are slower.	Uploading files and command execution are faster.
Need to supply the ESXi host's administrator credentials.	No need to supply the ESXi host's administrator credentials.
Cannot handle special characters in the administrator's password when running the script to install Horizon Agent or the AD user's password when running the script to join the domain.	Can handle special characters in the administrator's password when running the script to install Horizon Agent or the AD user's password when running the script to join the domain.

NOTE Both PowerCLI-based and SSH-based scripts can handle special characters in the passwords for the vCenter Server administrator and the Linux administrator. PowerCLI-based scripts can also handle special characters in the ESXi host administrator's password. In all these cases, an escape character is not necessary.

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 and install Horizon Agent on the virtual machine.
See [“Create a Virtual Machine Template for Cloning Linux Desktop Machines,”](#) on page 59.
- 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.
Domain	Specify the domain of the View environment.
Network Settings	Use standard network settings.
Primary DNS	Specify a valid address.

NOTE For more information on Guest OS Customization Support Matrix, see <http://partnerweb.vmware.com/programs/guestOS/guest-os-customization-matrix.pdf>.

- 3 Clone virtual machines.
See [“Sample Script to Clone Linux Virtual Machines,”](#) on page 61.
- 4 Join the cloned VMs to the Active Directory (AD) domain if you are using the winbind solution. You can run the domain join command with example scripts below or use option `RunOnceScript` in `/etc/vmware/viewagent-custom.conf`, configured in the template virtual machine.
See [“Sample Script to Join Cloned Virtual Machines to AD Domain,”](#) on page 65 or [“Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH,”](#) on page 68.
- 5 Update configuration options in virtual machines.

See [“Sample Script to Upload Configuration Files to Linux Virtual Machines,”](#) on page 71 or [“Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH,”](#) on page 75.

- 6 Create a desktop pool.

See [“Create a Manual Desktop Pool for Linux,”](#) on page 49.

For a sample script that performs operations such as powering on, shutting down, restarting, or deleting virtual machines, see [“Sample Script to Perform Operations on Linux Virtual Machines,”](#) on page 89. This script can delete virtual machines from vCenter Server.

Overview of Bulk Upgrade of Linux Desktops

Bulk upgrade of manual desktops for Linux involve several steps. You can automate some of the steps by using PowerCLI scripts.

Bulk Upgrade Unmanaged Desktop

To bulk upgrade the unmanaged virtual machine to managed or unmanaged virtual machine, you must use the sample upgrade script to upload the new Horizon Agent to the existing virtual machines and run upgrade command.

- If you retain the unmanaged virtual machine, your existing desktop pool can be reused.
- If you upgrade from unmanaged virtual machine to managed virtual machine, you must delete the existing desktop pool and create a new desktop pool. For more information, see [“Upgrade Horizon Agent on a Linux Virtual Machine,”](#) on page 40.

Bulk Upgrade Managed Desktop

To bulk upgrade the managed virtual machine, select one of the following methods.

Method	Description
In the template virtual machine, install or upgrade the new Horizon Agent and create a snapshot.	<ul style="list-style-type: none"> ■ The user data and profile are lost since the existing virtual machines are deleted, unless the user data and profile are located on the share server such as NFS server. ■ After the virtual machine replacement, the state of the virtual machine on View Administrator might be missing. You must restart the broker service to fix it. ■ If you are using linked clone, this method avoids duplicate data on each virtual machine.
Use the sample script of upgrade to upload the new Horizon Agent to existing virtual machines and run the upgrade command.	<ul style="list-style-type: none"> ■ User data and profile is retained. ■ If you are using linked clone, this method introduces duplicate data on each virtual machine.

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 7 for Linux,”](#) on page 10.

- 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 7 for Linux](#)," on page 10.
- Familiarize yourself with the steps for AD integration. See [Chapter 3, "Setting Up Active Directory Integration for Linux Desktops,"](#) on page 21.
- Familiarize yourself with the steps to install Horizon Agent on Linux. See Chapter "Installing Horizon Agent and Managing Linux Desktops"
- If required, familiarize yourself with the steps to configure options using the View configuration files. See [Chapter 6, "Configuration Options for Linux Desktops,"](#) on page 43.
- If you plan to set up graphics, familiarize yourself with the steps. See [Chapter 4, "Setting Up Graphics for Linux Desktops,"](#) on page 25.

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).

Set the vRAM size in vSphere Web Client for virtual machines that use 2D or vSGA, which use the VMware driver. The vRAM size has no affect on vDGA or NVIDIA GRID vGPU machines, which use NVIDIA drivers.

For recommended values, follow the guidelines in *System Requirements for Horizon 7 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 Horizon 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 Horizon 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 following lines:


```
Defaults requiretty
Defaults !visiblepw
```
- 7 If the Linux distribution is not RHEL 7, CentOS 7, or SLES 12, install VMware Tools .

RHEL 7, CentOS 7, and SLES 12 have Open VM Tools installed by default.
- 8 If the Linux distribution is RHEL 7, CentOS 7, or SLES 12, install the `deployPkg` plug-in.

The instructions are at <http://kb.vmware.com/kb/2075048>.

- 9 For RHEL and CentOS, enable the Network Connection setting **Connect automatically**.
- 10 Perform the AD integration tasks.
- 11 Perform the steps to set up graphics.
- 12 Install Horizon agent


```
sudo ./install_viewagent.sh -A yes
```
- 13 Perform additional configurations using the View configuration files.
- 14 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 Script to Clone Linux Virtual Machines

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

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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

```
<#
Create Clones from a Master VM

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.
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}

function IsVMExists ()
{
    Param($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()
{
    Param($VMToDisableConsole)
    $svmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec
```

```

    $extra = New-Object VMware.Vim.optionvalue
    $extra.Key="RemoteDisplay.maxConnections"
    $extra.Value="0"
    $vmConfigSpec.extraconfig += $extra
    $vm = Get-VM $VMToDisableConsole | Get-View
    $vm.ReconfigVM($vmConfigSpec)
}

function Delete_VM()
{
    Param($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 -----

$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
$cloneType = GetInput -prompt 'Clone Type ("linked" or "full")' -IsPassword $false
$disableVMConsole = GetInput -prompt 'Disable vSphere VM Console ("yes" or "no", recommend
"yes")' -IsPassword $false
"-----"
$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)
#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{

```

```

    Write-Host 'Exit since failed to login vCenter'
    exit
}
else
{
    Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile
#$csvData = Import-CSV $csvFile -
header("VMName","Parentvm","CustomSpec","Datastore","Host","FromSnapshot","DeleteIfPresent")
foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

    $destVMName=$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, has its own unique host name, and VMware Tools is running.

Sample Script to Join Cloned Virtual Machines to AD Domain

You can customize and use the following sample script to join cloned virtual machines (VMs) to an Active Directory (AD) domain.

You need to run this script if you use the Winbind solution for AD integration because the step to join the domain will fail for the cloned VMs. This script runs a command to join the domain on each VM. You do not need to run this script if you use the OpenLDAP solution.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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 VM
- User password for the Linux VM
- Login name of an AD user that is authorized to join machines to the domain
- Password of the authorized AD user

Script Content

```
<#
.SYNOPSIS
run command "sudo /usr/bin/net ads join"

.DESCRPTION
The tool is to run the command "sudo /usr/bin/net ads join" to join Linux to AD

.NOTES
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}
#----- Handle input -----
"-----"
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$sadUser = GetInput -prompt 'Type the AD user name to join the AD' -IsPassword $false
""
```

```

``nPlease type the AD user password."
"Plase note that special character in password may not work with the script"
$adUserPassword = GetInput -prompt 'Your AD user password' -IsPassword $true
"-----"

#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

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

#Connect to vCenter
#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}
else
{
    Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

#Handle VMs one by one
foreach ($line in $csvData)
{
    ``n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

    $cmd = "sudo /usr/bin/net ads join -U $adUser%$adUserPassword"
    Write-Host "Run cmd 'sudo /usr/bin/net ads join' 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> .\ClonedVMs_JoinDomain.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 AD user name to join the AD: viewadmin
Please type the AD user password.
Please note that special character in password may not work with the script.
Your AD user password: *****

```

Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH

You can customize and use the following sample script to join cloned virtual machines (VMs) to an Active Directory (AD) domain. This script uses SSH to run commands on the Linux VMs.

You need to run this script if you use the Winbind solution for AD integration because the step to join the domain will fail for the cloned VMs. This script runs a command to join the domain on each VM. You do not need to run this script if you use the OpenLDAP solution.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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
- User login name for the Linux VM
- User password for the Linux VM
- Login name of an AD user that is authorized to join machines to the domain
- Password of the authorized AD user

Script Content

```

<#
.SYNOPSIS
run command "sudo /usr/bin/net ads join" via SSH

.DESRIPTION
The tool is to run the command "sudo /usr/bin/net ads join" to join Linux machine to AD via SSH

.NOTES
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)

```

```

    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}

function Check_SSH_Client
{
    Param($IsPlink, $IsPSCP)
    if ($IsPlink)
    {
        if (Test-Path ".\plink.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "plink.exe" found'
        }
        else
        {
            write-host -ForegroundColor Red 'SSH client "plink.exe" not found, please download
from its official web site'
            exit
        }
    }
    if ($IsPSCP)
    {
        if (Test-Path ".\pscp.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "pscp.exe" found'
        }
        else
        {
            write-host -ForegroundColor Red 'SSH client "pscp.exe" not found, please download
from its official web site'
            exit
        }
    }
}

function RunCmdViaSSH
{
    Param($VM_Name, $User, $Password, $Cmd, $returnOutput = $false)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    write-host "Run cmd on $VM_Name ($IP)"
    if($returnOutput)
    {

```

```

        $command = "echo yes | .\plink.exe -ssh -l $user -pw $password $IP " + '"' + $cmd + '"'
        $output = Invoke-Expression $command
        return $output
    }
    else
    {
        echo yes | .\plink.exe -ssh -l $user -pw $password $IP "$cmd"
    }
}

function UploadFileViaSSH
{
    Param($VM_Name, $User, $Password, $LocalPath, $DestPath)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    $command = "echo yes | .\pscp.exe -l $User -pw $Password $LocalPath $IP" + ":" + "$DestPath"
    write-host "Upload file: $command"
    Invoke-Expression $command
}

#----- Handle input -----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $false
"-----"
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$adUser = GetInput -prompt 'Type the AD user name to join the AD' -IsPassword $false
""
"
Please type the AD user password."
[Console]::ForegroundColor = "Yellow"
"Please note that special character should be escaped. For example, $ should be \$"
[Console]::ResetColor()
$adUserPassword = GetInput -prompt 'Your AD user password' -IsPassword $true
"-----"

#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

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

#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}
else

```

```

{
  Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

#Handle VMs one by one
foreach ($line in $csvData)
{
  "-----"
  $VMName = $line.VMName
  write-host -ForegroundColor Yellow "VM: $VMName`n"

  $cmd = "sudo /usr/bin/net ads join -U $adUser%$adUserPassword"
  Write-Host "Run cmd 'sudo /usr/bin/net ads join' in VM '$VMName' with user '$guestUser'"
  RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
}

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```

PowerCLI C:\scripts> .\ClonedVMs_JoinDomain_SSH.ps1
-----
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****
-----
Type the AD user name to join the AD: viewadmin
Please type the AD user password.
Please note that special character should be escaped. For example, $ should be \$
Your AD user password: *****

```

Sample Script to Upload Configuration Files to Linux Virtual Machines

You can customize and use the following sample script to upload the configuration files `config` and `viewagent-custom.conf` to multiple Linux virtual machines (VMs).

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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 VM
- User password for the Linux VM

Script Content

```
<#
Upload the configuration files config and viewagent-custom.conf to Linux VMs
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}

#----- Handle Input -----
"-----"
write-host -ForegroundColor Blue 'Please ensure your config file and viewagent-custom.conf file
are in current working directory'
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"

$csvFile = '.\CloneVMs.csv'
$setConfig = $false
$setCustomConf = $false
```

```

$config_File = "config"
$customConf_File = "viewagent-custom.conf"

#check if config file exists
if(Test-Path $config_File)
{
    $setConfig = $true
    write-host -ForegroundColor Yellow "config" file found'
}
else
{
    write-host -ForegroundColor Yellow "config" file not found, skip it'
}

if(Test-Path $customConf_File)
{
    $setCustomConf = $true
    write-host -ForegroundColor Yellow "viewagent-custom.conf" file found'
}
else
{
    write-host -ForegroundColor Yellow "viewagent-custom.conf" file not found, skip it'
}

if (($setConfig -eq $false)-AND ($setCustomConf -eq $false))
{
    write-host -ForegroundColor Red 'Both file not found, exit'
    exit
}

#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}
else
{
    Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

#Handle VMs one by one
foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"
}

```

```

#Try to delete the configuration file from home folder on destination VM
$cmd = "rm -rf config viewagent-custom.conf"
Write-Host "Run cmd '$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

if ($setConfig)
{
Write-Host "Upload File '$config_File' 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 $config_File

$cmd = "sudo mv ./ $config_File /etc/vmware/";
Write-Host "Move configuraton file: $cmd"
Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
}

if ($setCustomConf)
{
Write-Host "Upload File '$customConf_File' 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 $customConf_File

$cmd = "sudo mv ./ $customConf_File /etc/vmware/";
Write-Host "Move configuraton file: $cmd"
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> .\UpdateOptionFile.ps1
-----
Please ensure your config file and view-agent.conf file are in current working directory.
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: *****

```

Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH

You can customize and use the following sample script to upload the configuration files `config` and `viewagent-custom.conf` to multiple Linux virtual machines (VMs). This script uses SSH to run commands on the Linux VMs.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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
- User login name for the Linux VM
- User password for the Linux VM

Script Content

```
<#
Upload the configuration files config and viewagent-custom.conf to Linux VMs using SSH
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}
function Check_SSH_Client
{
    Param($IsPlink, $IsPSCP)
    if ($IsPlink)
    {
```

```

        if (Test-Path ".\plink.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "plink.exe" found'
        }
        else
        {
            write-host -ForegroundColor Red 'SSH client "plink.exe" not found, please download
from its official web site'
            exit
        }
    }
    if ($IsPSCP)
    {
        if (Test-Path ".\pscp.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "pscp.exe" found'
        }
        else
        {
            write-host -ForegroundColor Red 'SSH client "pscp.exe" not found, please download
from its official web site'
            exit
        }
    }
}

function RunCmdViaSSH
{
    Param($VM_Name, $User, $Password, $Cmd, $returnOutput = $false)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    write-host "Run cmd on $VM_Name ($IP)"
    if($returnOutput)
    {
        $command = "echo yes | .\plink.exe -ssh -l $user -pw $password $IP " + "'" + $cmd + "'"
        $output = Invoke-Expression $command
        return $output
    }
    else
    {
        {
            echo yes | .\plink.exe -ssh -l $user -pw $password $IP "$cmd"
        }
    }
}

function UploadFileViaSSH
{
    Param($VM_Name, $User, $Password, $LocalPath, $DestPath)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    $command = "echo yes | .\pscp.exe -l $User -pw $Password $LocalPath $IP" + ":" + "$DestPath"
    write-host "Upload file: $command"
    Invoke-Expression $command
}

```

```

}

#----- Handle Input -----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-----"
write-host -ForegroundColor Blue 'Please ensure your config file and viewagent-custom.conf file
are in current working directory'
SvcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
SvcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
SvcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"

$csvFile = '.\CloneVMs.csv'
$setConfig = $false
$setCustomConf = $false
$config_File = "config"
$customConf_File = "viewagent-custom.conf"

#check if config file exists
if(Test-Path $config_File)
{
    $setConfig = $true
    write-host -ForegroundColor Yellow '"config" file found'
}
else
{
    write-host -ForegroundColor Yellow '"config" file not found, skip it'
}

if(Test-Path $customConf_File)
{
    $setCustomConf = $true
    write-host -ForegroundColor Yellow '"viewagent-custom.conf" file found'
}
else
{
    write-host -ForegroundColor Yellow '"viewagent-custom.conf" file not found, skip it'
}

if (($setConfig -eq $false)-AND ($setCustomConf -eq $false))
{
    write-host -ForegroundColor Red 'Both file not found, exit'
    exit
}

#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}

```

```

}
else
{
    Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

#Handle VMs one by one
foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

    #Try to delete the configuration file from home folder on destination VM
    $cmd = "rm -rf config viewagent-custom.conf"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

    if ($setConfig)
    {
        Write-Host "Upload File '$config_File' to '$destFolder' of VM '$VMName' with user '$guestUser'"
        UploadFileViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -LocalPath $config_File -DestPath $destFolder

        $cmd = "sudo mv ./ $config_File /etc/vmware/";
        Write-Host "Move configuraton file: $cmd"
        RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
    }

    if ($setCustomConf)
    {
        Write-Host "Upload File '$customConf_File' to '$destFolder' of VM '$VMName' with user '$guestUser'"
        UploadFileViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -LocalPath $customConf_File -DestPath $destFolder

        $cmd = "sudo mv ./ $customConf_File /etc/vmware/";
        Write-Host "Move configuraton file: $cmd"
        RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
    }
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```
PowerCLI C:\scripts> .\UpdateOptionFile.ps1
-----
Please ensure your config file and view-agent.conf file are in current working directory.
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****
```

Sample Script to Upgrade Horizon Agent on Linux Desktop Machines

You can customize and use the following sample script to upgrade Horizon Agent on multiple Linux virtual machines (VMs).

This script uploads the installer tar ball to each VM before installing Horizon Agent. The upload task can be time-consuming, especially when a large number of VMs is involved and the network speed is slow. To save time, you can run the script that uses SSH, or put the installer tar ball in a shared location that is available to each VM so that uploading the file is not necessary.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. This script also interactively asks for the following information:

- Acceptance of Horizon Agent EULA (end user license agreement)
- 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
- Horizon Agent tar ball path
- Upgrade to managed VM
- Install the Smartcard redirection feature

Script Content

```
<#
Upload the Linux Agent installer tar ball and re-install
#>

#-----
Functions-----
```

```

function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}
#-----Handle
input-----
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux View Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the View Agent tar ball path' -IsPassword $false
"-----"
$UpgradeToManagedVM = GetInput -prompt 'Upgrade to managed VM ("yes" or "no")' -IsPassword $false
if (($UpgradeToManagedVM -ne "yes") -AND $UpgradeToManagedVM -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")'
-IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}

```

```

}
"-----"

# $csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

# 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 GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPa
th)));
    $md5Hash = ($md5HashWithFormat.replace("-", "")).ToLower();
    return $md5Hash;
}

#-----
Main-----
# Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

# Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}
else
{
    Write-Host 'vCenter is connected'
}

# Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

# Handle VMs one by one

```

```

foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

    $cmd = "rm -rf VMware-*linux-*"
    Write-Host "Run cmd '$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

    #Check the uploaded installer md5sum
    $cmd = "md5sum VMware-*linux-*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    $output = Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -
    GuestUser $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText
    $cmd

    if($output.Contains($installerMD5Hash))
    {
        Write-Host $VMName": Uploaded installer's MD5Sum matches the local installer's MD5Sum";
        Write-Host $VMName": Extract the installer and do installation";
        $cmd = "tar -xzvf VMware-*linux-*.tar.gz"
        Write-Host "Run cmd '$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 setenforce 0";
        Write-Host "Set the selinux to permissive mode: $cmd"
        Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
        $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

        $cmd = "sudo killall /usr/lib/vmware/viewagent/VMwareBlastServer/VMwareBlastServer"
        Write-Host "Stop VMwareBlastServer before upgrading: $cmd"
        Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
        $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

        #Run the upgrade command.
        $cmd = "cd VMware-*linux-* && sudo ./install_viewagent.sh -A yes -m $installSmartcard -
        M $UpgradeToManagedVM"
        Write-Host "Run upgrade cmd in VM '$VMName' with user '$guestUser': $cmd"
        Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
        $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

        $cmd = "sudo shutdown -r +1&"
        Write-Host "Reboot to apply the View Agent installation"
        Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
        $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
    }
}

```

```

    }
    else
    {
        Write-Host $VMName": Uploaded installer's MD5Sum does NOT match the local installer's
MD5Sum";
        Write-Host $VMName": Skip the installation. Please check your network and VMware Tools
status";
        exit;
    }
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```

PowerCLI C:\scripts> .\InstallAgent.ps1
-----
Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
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 Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
-----
-----
Upgrade to managed VM ("yes" or "no"): yes
Install the Smartcard redirection feature ("yes" or "no"): no

```

Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH

You can customize and use the following sample script to upgrade Horizon Agent on multiple Linux virtual machines (VMs). This script uses SSH to run commands on the Linux VMs.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. This script also interactively asks for the following information:

- Acceptance of Horizon Agent EULA (end user license agreement)
- 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
- Horizon Agent tar ball path
- Upgrade to managed VM
- Install the Smartcard redirection feature

Script Content

```
<#
Upload the Linux Agent installer tar ball and re-install
#>

#-----
Functions-----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}
function Check_SSH_Client
{
    Param($IsPlink, $IsPSCP)
    if ($IsPlink)
    {
        if (Test-Path ".\plink.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "plink.exe" found'
        }
        else
        {
            write-host -ForegroundColor Red 'SSH client "plink.exe" not found, please download
from its official web site'
            exit
        }
    }
}
```

```

}
if ($IsPSCP)
{
    if (Test-Path ".\pscp.exe")
    {
        write-host -ForegroundColor Yellow 'SSH client "pscp.exe" found'
    }
    else
    {
        write-host -ForegroundColor Red 'SSH client "pscp.exe" not found, please download
from its official web site'
        exit
    }
}
}

function RunCmdViaSSH
{
    Param($VM_Name, $User, $Password, $Cmd, $returnOutput = $false)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    write-host "Run cmd on $VM_Name ($IP)"
    if($returnOutput)
    {
        $command = "echo yes | .\plink.exe -ssh -l $user -pw $password $IP " + "'" + $cmd + "'"
        $output = Invoke-Expression $command
        return $output
    }
    else
    {
        echo yes | .\plink.exe -ssh -l $user -pw $password $IP "$cmd"
    }
}

function UploadFileViaSSH
{
    Param($VM_Name, $User, $Password, $LocalPath, $DestPath)

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
    $command = "echo yes | .\pscp.exe -l $User -pw $Password $LocalPath $IP" + ":" + "$DestPath"
    write-host "Upload file $LocalPath to VM $VM_Name with user $User"
    Invoke-Expression $command
}

#-----Handle
input-----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux View Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")

```

```

{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$SvcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$SvcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$SvcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the View Agent tar ball path' -IsPassword $false
"-----"
$UpgradeToManagedVM = GetInput -prompt 'Upgrade to managed VM ("yes" or "no")' -IsPassword $false
if (($UpgradeToManagedVM -ne "yes") -AND $UpgradeToManagedVM -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")'
-IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
}
"-----"

#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

#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 GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPa
th)));
    $md5Hash = ($md5HashWithFormat.replace("-", "")).ToLower();
    return $md5Hash;
}

```

```

}

#-----
Main-----
#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

#Connect to vCenter
$VC_Conn_State = Connect-VIServer $vcAddress -user $vcAdmin -password $vcPassword
if([string]::IsNullOrEmpty($VC_Conn_State))
{
    Write-Host 'Exit since failed to login vCenter'
    exit
}
else
{
    Write-Host 'vCenter is connected'
}

#Read input CSV file
$csvData = Import-CSV $csvFile

$destFolder = "/home/$guestUser/"

#Handle VMs one by one
foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

    $cmd = "rm -rf VMware-*linux-*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

    #Upload installer tar ball to Linux VM
    Write-Host "Upload File '$agentInstaller' to '$destFolder' of VM '$VMName' with user '$guestUser'"
    UploadFileViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -LocalPath $agentInstaller -DestPath $destFolder

    #Check the uploaded installer md5sum
    $cmd = "md5sum VMware-*linux-*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    $output = RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd -
    $returnOutput $true

    if($output.Contains($installerMD5Hash))
    {
        Write-Host $VMName": Uploaded installer's MD5Sum matches the local installer's MD5Sum";
        Write-Host $VMName": Extract the installer and do installation";

        $cmd = "tar -xzf VMware-*linux-*.tar.gz"
        Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    }
}

```

```

RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

$cmd = "sudo setenforce 0";
Write-Host "Set the selinux to permissive mode: $cmd"
RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

$cmd = "sudo killall /usr/lib/vmware/viewagent/VMwareBlastServer/VMwareBlastServer"
Write-Host "Stop VMwareBlastServer before upgrading: $cmd"
RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

#Run the upgrade command.
$cmd = "cd VMware-*--linux-* && sudo ./install_viewagent.sh -r yes -A yes -m
$installSmartcard -M $UpgradeToManagedVM"
Write-Host "Run upgrade cmd in VM '$VMName' with user '$guestUser': $cmd"
RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
Write-Host -ForegroundColor Yellow "Linux Agent installer will reboot the Linux VM after
upgrade, and you may hit the ssh connection closed error message, which is expectation"
}
else
{
Write-Host $VMName": Uploaded installer's MD5Sum does NOT match the local installer's
MD5Sum";
Write-Host $VMName": Skip the installation. Please check your network and VMware Tools
status";
exit;
}
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

Script Execution

The following messages are from an execution of the script:

```

PowerCLI C:\scripts> .\InstallAgent.ps1
-----
Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****
-----
Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
-----
Upgrade to managed VM ("yes" or "no"): yes
Install the Smartcard redirection feature ("yes" or "no"): no

```

Sample Script to Perform Operations on Linux Virtual Machines

You can customize and use the following sample script to perform operations on multiple Linux virtual machines (VMs). The operations include powering on, powering off, shutting down, restarting, and deleting the VMs.

This script can delete virtual machines from vCenter Server but not from View.

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 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 61. 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 VM, restart VM guest, or delete VM.
- The wait time, in seconds, between operations on the VMs.

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 GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
```

```

        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}

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 -----
"-----"
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$action = GetInput -prompt 'Select action: 1). Power On 2). Power Off 3) Shutdown VM Guest 4).
Restart VM 5). Restart VM Guest 6). Delete VM' -IsPassword $false
$sleepTime = GetInput -prompt 'Wait time (seconds) between each VM' -IsPassword $false
"-----"
[Console]::ForegroundColor = "Yellow"
switch ($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
  }
}
[Console]::ResetColor()
$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> .\VMOperations.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
-----
Your selection is 6). Delete VM

```

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.

Troubleshooting Linux Desktops

Certain issues might arise when you manage Linux desktops. You can follow various procedures to diagnose and fix problems.

This chapter includes the following topics:

- [“Collect Diagnostic Information for Horizon 7 for Linux Machine,”](#) on page 93
- [“Troubleshooting Copy and Paste between Remote Desktop and Client Host,”](#) on page 94
- [“Configuring the Linux Firewall to Allow Incoming TCP Connections,”](#) on page 94
- [“View Agent Fails to Disconnect on an iPad Pro Horizon Client,”](#) on page 94
- [“SLES 12 SP1 Desktop does not Auto Refresh after Drag and Drop,”](#) on page 95
- [“SSO Fails to Connect to a PowerOff Agent,”](#) on page 95
- [“Unreachable VM After Creating a Manual Desktop Pool for Linux,”](#) on page 95

Collect Diagnostic Information for Horizon 7 for Linux Machine

You can collect diagnostic information to help VMware Technical Support diagnose and resolve issues with a Horizon 7 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 Copy and Paste between Remote Desktop and Client Host

Copy and Paste between the remote desktop and client host takes more than three seconds for maximum supported data of 1 MB. This issue does not occur if you copy and paste a small data size.

Problem

When you configure 1 vCPU and 1 GB memory for SLED 11 SP3/SP4 desktop, it might take more than three seconds to copy and paste between the remote desktop and local client host.

Cause

The delay in copy and paste might occur due to the old operating system APIs of SLED 11 SP3/SP4.

Solution

- ◆ Configure two vCPUs and 2 GB memory for SLED 11 SP3/SP4.

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 Horizon 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, 22443.

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.

View Agent Fails to Disconnect on an iPad Pro Horizon Client

The SUSE View Agent connection fails to disconnect after a restart or shutdown on a iPad Pro Horizon Client.

Problem

When you restart or shutdown a SUSE virtual machine on an iPad Pro Horizon Client, the desktop does not respond. The View Agent fails to disconnect.

Cause

SUSE machine might not be sending messages correctly to Horizon Client after a restart or shutdown operation.

Solution

- ◆ Disconnect the desktop connection manually from iPad Pro Horizon Client.

SLES 12 SP1 Desktop does not Auto Refresh after Drag and Drop

SLES 12 SP1 does not auto refresh in a multimon mode when you drag and drop a gnome terminal.

Problem

When you launch SLES 12 SP1 in a multimon mode and return to the window mode, the desktop does not refresh automatically when you drag and drop a gnome terminal.

Cause

The gnome terminal does not respond to the drag and drop operation.

Solution

- 1 End the gnome-shell.
 - ◆ `kill -9 `pidof gnome-shell``
- 2 Restart gnome-shell again.

SSO Fails to Connect to a PowerOff Agent

SSO does not connect to a poweroff agent.

Problem

When you login as a broker and connect to an agent, SSO fails to connect to the poweroff agent.

Solution

- ◆ Manually login to the desktop or disconnect and reconnect to the agent again.

Unreachable VM After Creating a Manual Desktop Pool for Linux

The virtual machine state is not responding.

Problem

The virtual machine status might be Waiting for Agent or Unreachable after you create a Manual Desktop Pool.

Cause

There might be several user error configuration or setup causes for the virtual machine state to be Unreachable or Waiting for Agent.

- Verify that the option `machine.id` exists in the virtual machines vmx configuration file.
 - If it does not exist, then verify that the virtual machine was added to the desktop pool correctly. Else recreate the desktop pool to let the broker rewrite the option to the vmx configuration file.
- Verify that the VMware Tool or Open VM Tool is installed correctly.
 - If the steps to install VMware Tool or Open VM Tool were not performed correctly, the `vmware-rpctool` command might not exist under `PATH` in the Linux virtual machine. You must follow the guide to install VMware Tool or Open VM Tool.
 - Run the command after you finish installing.


```
#vmware-rpctool "machine.id.get"
```
 - The `machine.id` values are listed from the virtual machines vmx configuration file.
- Verify if the FQDN of the broker can be resolved to the IP Address in the agent Linux virtual machine.

Index

A

Active Directory, integrating with Linux **21**
automated full-clone desktop pool creation, Linux virtual machines **51**

B

broker powercli **53**
bulk deployment
 create a virtual machine template for cloning **59**
 input file for sample scripts **61**
 overview **57**
bulk upgrade **59**

C

configuration options
 audio out **43**
 clipboard redirection **43**
 example Blast settings for Linux desktops) **46**
 left-handed mouse **43**
 lossless PNG mode **43**
 single sign-on (SSO) **43**
configuring **39**

D

desktop pool creation, Linux virtual machines **49**
DIGEST-MD5 **37**

F

firewalls, Linux desktops **94**

G

glossary **5**
graphics **25**

H

Horizon 7 for Linux, bulk deployment **57**
Horizon Agent
 installing on a Linux virtual machine **37**
 Linux installation command options **38**
 upgrading on a Linux virtual machine **40**

I

install_viewagent.sh, command usage and options **38**
installing **18**
intended audience **5**

K

Kerberos **37**

L

Linux desktops
 configuring **43**
 creating and managing Linux desktop pools **49**
 features **7**
 features and requirements **7**
 installing Horizon Agent **37**
 setting up AD integration **21**
Linux systems
 adding to automated desktop pools **51**
 adding to desktop pools **49**
 collecting diagnostic information **93**
 configuring the firewall **94**
 desktops **5**
 overview of desktop setup steps **9**
 preparing for desktop use **16**
 vSphere console display **47**
Linux virtual machines, preparing for desktop deployment **15**

M

managing manual desktop **50**

N

NVIDIA display driver, verify the installation **29, 33**
NVIDIA display driver for vDGA, installing **31**
NVIDIA GRID vGPU
 installing the VIB for Linux desktops **26**
 virtual GPU Types **27**
NVIDIA GRID vGPU display driver, installing **28**
NVIDIA vGPU, configuring a shared PCI device **27**

O

OpenLDAP **21**

R

RHEL
 configuring for vDGA **30**
 configuring for vGPU **25**
 configuring for vSGA **34**

S

sample script

- clone desktop machines using SSH **68**
- perform operations on VMs **89**
- to clone desktop machines **61, 65**
- to upgrade Horizon Agent **79**
- upgrade Horizon Agent using SSH **83**
- upload configuration files **71**
- upload configuration files using SSH **75**

sample scripts, input file **61**

single sign-on **22**

smart card redirection **22**

sso failure **95**

T

troubleshooting **93–95**

troubleshooting, disconnecting **94**

U

uninstalling **41**

unreachable VM **95**

V

vDGA

- adding PCI device to a Linux machine **31**
- enabling DirectPath I/O on a host **30**

vDGA (Virtual Dedicated Graphics Acceleration), for RHEL 6.6 **30**

vGPU, configuring RHEL 6.6 and RHEL 7.2 **25**

View Agent, Linux operating system requirements **10**

virtual machines, creating for Linux desktops **15**

VMware Tools, configuring on a Linux guest **16**

vSGA

- configuring 3D settings on a Linux video
card **35**

for RHEL 7.2 **34**

- installing the VIB for Linux desktops **34**

- verify on a Linux machine **35**

vSphere console, suppress display **47**

W

Winbind **21**