Setting Up Horizon 7 for Linux Desktops

VMware Horizon 7 7.0

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 © 2017 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 for Linux Desktops 5

1 Features and System Requirements 7
  Features of Horizon 7 Linux Desktops and Desktop Pools 7
  Overview of Configuration Steps for Horizon 7 for Linux Desktops 7
  System Requirements for Horizon 7 for Linux 8

2 Preparing a Linux Virtual Machine for Desktop Deployment 13
  Create a Virtual Machine and Install Linux 13
  Prepare a Linux Machine for Remote Desktop Deployment 14
  Install Dependency Packages for Horizon Agent 16

3 Setting Up Active Directory Integration for Linux Desktops 19
  Integrating Linux with Active Directory 19
  Setting Up Single Sign-on and Smart Card Redirection 20

4 Configuration Options for Linux Desktops 23
  Setting Options in Configuration Files on a Linux Desktop 23
  Suppress the vSphere Console Display of a Linux Desktop 25

5 Setting Up Graphics for Linux Desktops 27
  Configure RHEL 6.6 and RHEL 7.1 for vGPU 27
  Configure RHEL 6.6 for vDGA 33
  Configure RHEL 7.1 for vSGA 36

6 Installing Horizon Agent and Managing Linux Desktops 41
  Install Horizon Agent on a Linux Virtual Machine 41
  Enable Reversible Password Encryption 45
  Configure the Certificate for Linux Agent 45
  Create a Desktop Pool That Contains Linux Virtual Machines 46
  Upgrade Horizon Agent on a Linux Virtual Machine 47
  Uninstalling and Reinstalling Horizon 7 for Linux Machines 47
  How to Perform Power Operations on Linux Desktops from vSphere 50
  Gather Information About Horizon 7 for Linux Software 50

7 Bulk Deployment of Horizon 7 for Linux Desktops 51
  Overview of Bulk Deployment of Linux Desktops 51
  Create a Virtual Machine Template for Cloning Linux Desktop Machines 53
  Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops 54
  Sample Script to Clone Linux Virtual Machines 55

Sample Script to Join Cloned Virtual Machines to AD Domain 59
Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH 61
Sample Script to Install Horizon Agent on Linux Virtual Machines 65
Sample Script to Install Horizon Agent on Linux Virtual Machines Using SSH 70
Sample Script to Upload Configuration Files to Linux Virtual Machines 75
Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH 79
Sample Script to Upgrade Horizon Agent on Linux Desktop Machines 83
Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH 87
Sample Script to Perform Operations on Linux Virtual Machines 92
Sample Script to Delete Machines from the Connection Server LDAP Database 96

8 Troubleshooting Linux Desktops 99
Collect Diagnostic Information for Horizon 7 for Linux Machine 99
Troubleshooting Horizon Agent Registration Failure for a Linux Machine 99
Troubleshooting an Unreachable Horizon Agent on a Linux Machine 100
Troubleshooting Horizon Agent on a Linux Machine That Is Not Responding 102
Troubleshooting Copy and Paste between Remote Desktop and Client Host 102
Configuring the Linux Firewall to Allow Incoming TCP Connections 102

Index 103

Setting Up Horizon 7 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

1

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 7 Linux Desktops and Desktop Pools," on page 7
- "Overview of Configuration Steps for Horizon 7 for Linux Desktops," on page 7
- "System Requirements for Horizon 7 for Linux," on page 8

Features of Horizon 7 Linux Desktops and Desktop Pools

Horizon 7 introduces a number of new features for Linux desktops. The following list presents the key features.

SLED 11 SP3/SP4	Horizon 7 supports SLED 11 SP3/SP4 release. However, Single Sign-on is not supported.
HTML Access	Horizon 7 supports HTML Access 4.0.0 on Chrome.
Dependency Package	Horizon Agent for Linux requires dependency packages unique to a Linux distribution. These packages must be installed before installing Horizon Agent for Linux.

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 8.
2. Create a virtual machine in vSphere and install the Linux operating system. See "Create a Virtual Machine and Install Linux," on page 13.
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 14.
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 41.
6. Create a desktop pool that contains the configured Linux virtual machines. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

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 8.
2. Create a virtual machine in vSphere and install the Linux operating system. See "Create a Virtual Machine and Install Linux," on page 13.
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 14.
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 and RHEL 7.1 for vGPU," on page 27.
   - See "Configure RHEL 6.6 for vDGA," on page 33.
   - See "Configure RHEL 7.1 for vSGA," on page 36.
6. Install Horizon Agent on the Linux virtual machine. See "Install Horizon Agent on a Linux Virtual Machine," on page 41.
7. Create a desktop pool that contains the configured Linux virtual machines. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

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 7, "Bulk Deployment of Horizon 7 for Linux Desktops," on page 51.

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 **NOTE** Disable Compiz to avoid poor performance. See http://kb.vmware.com/kb/2114809 for more information.	x64
Ubuntu 12.04	x64

Table 1-1. Supported Linux Operating Systems for View Agent (Continued)

Linux Distribution	Architecture
RHEL 6.6 and 7.1	x64
CentOS 6.6 and 7.1	x64
NeoKylin 6 Update 1	x64
SLED 11 SP3/SP4	x64
SLED 12 is supported for installation and connections only.	x64

NOTE Linux agent has dependency packages on some Linux distribution. See "Install Dependency Packages for Horizon Agent," on page 16 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.
Horizon environment	Horizon 6 version 6.2.1 or Horizon 7 version 7.0.0.
Horizon client software	■ Horizon Client 4.0.0 for Windows ■ Horizon Client 4.0.0 for Linux ■ Horizon Client 4.0.0 for Mac OS X ■ HTML Access 4.0.0 on Chrome ■ Zero clients and mobile 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

Recommended 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.

These vRAM size recommendations apply only to virtual machines that are configured to use 2D or vSGA graphics, which use the VMware driver. vDGA and NVIDIA GRID vGPU use NVIDIA drivers. The vRAM size set in vSphere Client has no affect on vDGA or NVIDIA GRID vGPU machines.

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.

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.

CAUTION Horizon 7 does not automatically configure the vRAM settings on Linux virtual machines, as happens on Windows virtual machines. You must configure the vRAM settings manually in vSphere Client.

Table 1-5. 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

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

vRAM Size	Number of Monitors	Maximum Resolution
128 MB	3	2560x1600
128 MB	4	2560x1600 RHEL and CentOS only support this configuration on vSphere 5.5. To support this configuration on Ubuntu, you must recompile the kernel and disable 3D. For information about how to disable 3D, see http://kb.vmware.com/kb/2114809 . For NeoKylin, this configuration is not supported.

NOTE To connect to a RHEL 6.6, CentOS 6.6, or SLED 11 Sp3 or Sp4 desktops with multiple monitors, you must set the number of displays and the amount of video memory as described in "Create a Virtual Machine and Install Linux," on page 13. In addition, with the virtual machine (VM) powered off, you must edit the vmx file and set `svga.maxWidth` and `svga.maxHeight` according to the number and orientation (horizontal or vertical) of the displays. You also must set `svga.autodetect` to false. Then power on the VM. The general rule is that the `svga.maxWidth` and `svga.maxHeight` values must be large enough to support all the displays. To support 4 displays at the maximum resolution of 2560x1600, **Edit Virtual Machine Settings** > **Options** > **General** > **Configuration** > **Parameters** > **Add Row** to add the following parameters. You do not need to modify the parameter `svga.vramSize`.

```
svga.autodetect="false"
svga.maxHeight="3200"
svga.maxWidth="10240"
```

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

- Only one monitor displays correctly. The other monitors might be black or mirror another monitor.
- A keystroke is displayed multiple times.
- The desktop becomes very slow.

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

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

NOTE To configure four monitors for a RHEL 7.1 machine that uses 2D rendering or 3D rendering with vSGA, set a maximum resolution of 2048x1536 for each monitor. To configure a RHEL 7.1 machine to use up to three monitors, set a maximum resolution of 2560x1600 for each monitor.

To improve performance for a RHEL 7.1 machine in a vSGA multiple-monitor environment, set the **3D Memory** setting for the virtual machine to 1 GB or larger, and configure 4 vCPUs for the machine. If you configure four monitors at 2048x1536 resolution on a RHEL 7.1 machine, set **Memory** to 4 GB, and configure 4 vCPUs for the machine.

Recommended vCPU and Shared Memory Settings to Support Multiple Monitors

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

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

```
sysctl –a | grep shm
```

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

```
sysctl –w "kernel.shmmax=65536000"
```

Recommended vCPU Settings for Video Playback

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

Monitor Resolution Supported by vDGA and NVIDIA GRID vGPU

A virtual machine that is configured to use vDGA or NVIDIA GRID vGPU can support up to 4 monitors with a maximum resolution of 2560x1600.

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 13
- "Prepare a Linux Machine for Remote Desktop Deployment," on page 14
- "Install Dependency Packages for Horizon Agent," on page 16

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 8.
- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing guest operating systems. See "Creating and Preparing Virtual Machines" in the *Setting Up Desktop and Application Pools in View* document.
- Familiarize yourself with the recommended video memory (vRAM) values for the monitors you will use with the virtual machine. See "System Requirements for Horizon 7 for Linux," on page 8.

Procedure

1 In vSphere Web Client or vSphere Client, create a new virtual machine.

2 Configure custom configuration options.

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

   b Specify the number of vCPUs and the vMemory size.

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

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

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

   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 as a gnome desktop environment.

KDE has not been certified to support Horizon Agent, although basic connections and the use of audio and video work well on some distributions such as Kubuntu.

5 Ensure that the system hostname is resolvable 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 SLED 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.

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 zxpf /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 For RHEL and CentOS, map the machine's host name to 127.0.0.1 in `/etc/hosts`.

   This step is not necessary for Ubuntu 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.

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 Edit `/etc/nsswitch.conf` and modify the `hosts` entry as follows:

   ```
   hosts: cache db files dns
   ```

   The default setting is `hosts: files dns` and might cause the Linux desktop to fail to recover from a temporary network outage. The desktop might lose communication with View Connection Server permanently and require a reboot to resolve the issue.

12 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 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.
	`yum install mesa-libxatracker xorg-x11-drv-vmware mesa-private-llvm mesa-dri-drivers`
	After updating, packages should be below version or newer:
	■ 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
	Note The update is not available for CentOS default online repository. You must enable the CentOS continuous release (CR) repository.
	`yum-config-manager --enable "CentOS-7 - cr"`
	After upgrading VMware Graphic driver, you can disable the repository again.
	`yum-config-manager --disable "CentOS-7 - cr"`

SLED 11 SP3/SP4 Upgrade xorg-x11-server to a version later than 7.4.27.111.1	`zypper install xorg-x11-server`
SLED 12 SP0 Upgrade VMware Graphic driver to a version later than 13.1.0-5.2	`zypper install xf86-video-vmware` if xf86-video-vmware-13.1.0-5.2.x86_64 or newer is not available in SLED online repository, you must enable the OpenSUSE online repository to install it. 1 zypper addrepo http://download.opensuse.org/distribution/leap/42.1/repo/oss/ oss-42.1 2 zypper ref 3 zypper install xf86-video-vmware 4 zypper removerepo oss-42.1
Ubuntu 1404 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	`wget http://launchpadlibrarian.net/201393830/indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb` `sudo dpkg -i ./indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb`

2 Install the optional packages for Horizon Agent.

Clipboard Redirection feature depends on libexpat.so.1 and libpng12.so.0. The installer displays a warning message for any missing library file and the installation continues. However, Clipboard Redirection feature will not work.

Table 2-2. Optional Dependency Packages

Linux Distribution	Packages
RHEL or CentOS 6.6	■ expat-2.0.1-11.el6_2.x86_64 ■ libpng-1.2.49-1.el6_2.x86_64
RHEL or CentOS 7.1	■ expat-2.1.0-8.el7.x86_64 ■ libpng12-1.2.50-6.el7.x86_64
Ubuntu 1204/1404	■ libexpat1 ■ libpng12-0
SLED 11 SP3/SP4	■ libexpat1-2.0.1-88.34.1 ■ libpng12-0-1.2.31-5.35.1
SLED 12 SP0	■ libexpat1-2.1.0-13.232.x86_64 ■ libpng12-0-1.2.50-10.1.x86_64

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 19
- "Setting Up Single Sign-on and Smart Card Redirection," on page 20

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>
  ```

VMware recommends the OpenLDAP or a similar solution because it does not require an additional step on cloned VMs.

For more information about bulk deploying Linux desktops, see Chapter 7, "Bulk Deployment of Horizon 7 for Linux Desktops," on page 51.

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 need to 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 23. 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/sssd/sssd.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
#service pcscd start
```

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 43.

With smart card redirection enabled, SSO is not implemented if the user logs in with the smart card. That is, after logging in to the Connection Server instance with the smart card, the user needs to type the smart card PIN to log in to the Linux desktop.. In addition, if the smart card redirection feature is installed in 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 are on the card, the one in the first slot is used and the others are ignored. This is a Linux limitation.

Configuration Options for Linux Desktops

4

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 23
- “Suppress the vSphere Console Display of a Linux Desktop,” on page 25

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, `template_config` and `template_viewagent-custom.conf`, to `/etc/vmware`. In addition, if the files `/etc/vmware/config` and `/etc/vmware/viewagent-custom.conf` do not exist, the installer copies `template-config` to `config` and `template_viewagent-custom.conf` 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 4-1. Configuration Options in `/etc/vmware/config`

Option	Value	Default	Description
RemoteDisplay.alwaysLossless	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.
mksVNCServer.useUInputButtonMapping	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 support a left-handed mouse and you do not need to set this option.
RemoteDisplay.allowAudio	TRUE or FALSE	TRUE	Set this option to disable audio out

Table 4-1. Configuration Options in `/etc/vmware/config` (Continued)

Option	Value	Default	Description
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.
Clipboard.Direction	0, 1, 2, or 3	2	This option determines the clipboard redirection policy. ■ 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.useXExtButtonMapping	TRUE or FALSE	FALSE	Set this option to enable or disable the support of a left-handed mouse on SLED 11 SP3.

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

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

Table 4-2. 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	Use this option to set a subnet with which the View Connection Server can connect to the Agent machine if there are more than one local IP addresses with different subnets. You must specify the value in IP address/CIDR format. For example, Subnet=192.168.1.0/24. NULL implies that the Linux Agent randomly selects the IP address.
SSOEnable	TRUE or FALSE	TRUE	Set this option to disable single sign-on (SSO).
SSOUserFormat	A text string	[username]	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: ■ 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.
SSLCiphers	A text string	!aNULL:kECDH+AES:ECDH+AES:RSA+AES:@STRENGTH	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.

Table 4-2. Configuration Options in `/etc/vmware/viewagent-custom.conf` (Continued)

Option	Value	Default	Description
SSLCipherServerPreference	TRUE or FALSE	TRUE	Use this option to enable or disable the option SSL_OP_CIPHER_SERVER_PREFERENCE. For more information, see https://www.openssl.org/docs/manmaster/ssl/SSL_CTX_set_options.html.
LogoutAfterDisconnectTimeout	An integer	2	Use this option to set the timeout value in minutes for `Automatically logoff after disconnect` if the login is in progress.
LogCnt	An integer	-1	Use this option to set the reserved log file count in `/tmp/vmware-root`. ■ -1 - keep all ■ 0 - delete all ■ > 0 - reserved log count.

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.

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.

Setting Up Graphics for Linux Desktops

5

You can configure RHEL 6.6 and 7.1 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 and RHEL 7.1 for vGPU," on page 27
- "Configure RHEL 6.6 for vDGA," on page 33
- "Configure RHEL 7.1 for vSGA," on page 36

Configure RHEL 6.6 and RHEL 7.1 for vGPU

You can set up an RHEL 6.6 and RHEL 7.1 to take advantage of NVIDIA vGPU (shared GPU hardware acceleration) capabilities on the ESXi host.

Important NVIDIA vGPU is supported on NVIDIA Maxwell M60 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 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 drivers 346.69 and 346.42 are now supported for vSGA.

For an NVIDIA GRID vGPU configuration, 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.

For an vSGA configuration, 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 346.69 or 346.42 is installed in your environment.
- For an NVIDIA GRID vGPU configuration, verify that the NVIDIA Maxwell M60 GPUs are installed on the ESXi host.
- For a vSGA configuration, verify that the NVIDIA GRID K1 or K2 GPUs are installed on the ESXi host.

Procedure

1. Download the VIB for your NVIDIA GRID 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	For vGPU, select **NVIDIA GRID vGPU**. For vSGA, 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. For an NVIDIA GRID vGPU configuration, take these steps:
   1. Uncompress the vGPU software package .zip file.
   2. 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.

3. For a vSGA configuration, upload the VMware vSphere ESXi Driver for vSGA to the ESXi 6.0 U1 host.
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 acceptance set --level=CommunitySupported
# esxcli software vib install --no-sig-check -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```

8 Reboot or update the ESXi host.

3D Type	Description

NVIDIA GRID vGPU	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`

vSGA	a Restart `xorg`, which is used for GPU assignment: `# /etc/init.d/xorg start` b Reboot the ESXi host.

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 13 and "Prepare a Linux Machine for Remote Desktop Deployment," on page 14.
- 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 Graphics Card on the ESXi Host," on page 27.
- 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 30.

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.

Table 5-1. Virtual GPU Types Available for NVIDIA GRID 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
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

Install the NVIDIA Display Driver

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

Prerequisites

- For an NVIDIA GRID vGPU configuration, 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 Graphics Card on the ESXi Host," on page 27.

  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 29

- For a vDGA configuration, verify that the PCI device was added to the RHEL 6.6 virtual machine. See "Add a vDGA Pass-Through Device to a RHEL 6.6 Virtual Machine," on page 33.

Procedure

1 Disable and blacklist the default NVIDIA Nouveau driver.

  a Edit the `grub.conf` file.

  For RHEL 6.6, the file is `/boot/grub/grub.conf`. For RHEL 7.1, the file is `/etc/default/grub.conf`.

RHEL Version	Command
6.6	`sudo vi /boot/grub/grub.conf`
7.1	`sudo vi /etc/default/grub.conf`

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

   c. Edit the `blacklist.conf` file.

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

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

      ```
      blacklist nouveau
      ```

2. Restart the virtual machine.

   The display has a changed look and feel.

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

   ```
   /sbin/lsmod | grep nouveau
   ```

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

4. Copy or download the NVIDIA display driver to the virtual machine.

   a. For an NVIDIA GRID vGPU configuration, copy the NVIDIA Linux Display Driver to the virtual machine.

   b. For a vDGA configuration, 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.

   The following example uses a driver package for NVIDIA GRID vGPU:

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

   The following example uses a driver package for vDGA:

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

8. Start the NVIDIA installer.

   For NVIDIA GRID vGPU:

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

   For vDGA:

   ```
   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 41.

Create a desktop pool that contains the configured Linux virtual machines. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a RHEL 6.6. 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 41.
- Verify that the Linux virtual machine is deployed in a desktop pool. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

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 for vDGA

You can set up an RHEL 6.6 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**.
4. To enable DirectPath I/O passthrough for the NVIDIA GRID GPUs, click **Edit**.

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

5. Select the NVIDIA GRID GPUs and click **OK**.

   The PCI devices are added to the table, DirectPath I/O PCI Devices Available to VMs.

6. Reboot the host to make the PCI devices available for use by the Linux virtual machines.

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

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

Prerequisites

- Verify that the Linux virtual machine is prepared for use as a desktop. See "Create a Virtual Machine and Install Linux," on page 13 and "Prepare a Linux Machine for Remote Desktop Deployment," on page 14.
- 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 33.

Procedure

1. Power off the virtual machine. and log in to the RHEL 6.6 guest operating system as a local user configured with `sudo` rights.
2. In vSphere Web Client, select the virtual machine and, under the **VM Hardware** tab, click **Edit Settings**.
3. In the **New device** menu, select **PCI Device**.
4. Click **Add** and select the PCI device from the drop-down menu.
5. Click **Reserve all memory** and click **OK**.

   You must reserve all virtual machine memory to enable the GPU to support vDGA.
6. Power on the virtual machine and open vSphere console to connect to the machine.
7. Verify that the NVIDIA GRID device is passed through to the virtual machine.

   Open a terminal window and run the following command:

   ```
   lspci | grep NVIDIA
   ```

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

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

Install the NVIDIA Display Driver

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

Prerequisites

- For an NVIDIA GRID vGPU configuration, 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 Graphics Card on the ESXi Host," on page 27.

  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 29
- For a vDGA configuration, verify that the PCI device was added to the RHEL 6.6 virtual machine. See "Add a vDGA Pass-Through Device to a RHEL 6.6 Virtual Machine," on page 33.

Procedure

1. Disable and blacklist the default NVIDIA Nouveau driver.
   1. Edit the `grub.conf` file.

      For RHEL 6.6, the file is `/boot/grub/grub.conf`. For RHEL 7.1, the file is `/etc/default/grub.conf`.

RHEL Version	Command
6.6	`sudo vi /boot/grub/grub.conf`
7.1	`sudo vi /etc/default/grub.conf`

   2. Add the `rdblacklist=nouveau` line at the end of the kernel options.
   3. Edit the `blacklist.conf` file.

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

   4. 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/lsmod | grep nouveau
```

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

4 Copy or download the NVIDIA display driver to the virtual machine.

   a For an NVIDIA GRID vGPU configuration, copy the NVIDIA Linux Display Driver to the virtual machine.

   b For a vDGA configuration, 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.

   The following example uses a driver package for NVIDIA GRID vGPU:

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

   The following example uses a driver package for vDGA:

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

8 Start the NVIDIA installer.

   For NVIDIA GRID vGPU:

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

   For vDGA:

```
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 41.

Create a desktop pool that contains the configured Linux virtual machines. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a RHEL 6.6. 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 41.
- Verify that the Linux virtual machine is deployed in a desktop pool. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

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 7.1 for vSGA

You can set up an RHEL 7.1 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 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 drivers 346.69 and 346.42 are now supported for vSGA.

For an NVIDIA GRID vGPU configuration, 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.

For an vSGA configuration, 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 346.69 or 346.42 is installed in your environment.
- For an NVIDIA GRID vGPU configuration, verify that the NVIDIA Maxwell M60 GPUs are installed on the ESXi host.
- For a vSGA configuration, verify that the NVIDIA GRID K1 or K2 GPUs are installed on the ESXi host.

Procedure

1 Download the VIB for your NVIDIA GRID 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	For vGPU, select **NVIDIA GRID vGPU**. For vSGA, 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 For an NVIDIA GRID vGPU configuration, take these steps:

a Uncompress the vGPU software package .zip file.

b 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.

3 For a vSGA configuration, upload the VMware vSphere ESXi Driver for vSGA to the ESXi 6.0 U1 host.

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 acceptance set --level=CommunitySupported
# esxcli software vib install --no-sig-check -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```

8 Reboot or update the ESXi host.

<table>
<tr><th>3D Type</th><th>Description</th></tr>
<tr><td>NVIDIA GRID vGPU</td><td>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.)<pre>
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
</pre></td></tr>
<tr><td>vSGA</td><td>a Restart <code>xorg</code>, which is used for GPU assignment:<pre>
# /etc/init.d/xorg start
</pre>b Reboot the ESXi host.</td></tr>
</table>

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 7.1 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 on the ESXi Host," 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 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 41.

Verify that vSGA is Running on a Linux Virtual Machine

You can verify that vSGA is running on a RHEL 7.1 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 41.

Installing Horizon Agent and Managing Linux Desktops

6

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 41
- "Enable Reversible Password Encryption," on page 45
- "Configure the Certificate for Linux Agent," on page 45
- "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46
- "Upgrade Horizon Agent on a Linux Virtual Machine," on page 47
- "Uninstalling and Reinstalling Horizon 7 for Linux Machines," on page 47
- "How to Perform Power Operations on Linux Desktops from vSphere," on page 50
- "Gather Information About Horizon 7 for Linux Software," on page 50

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.

To install Horizon Agent you must supply the credential of a View administrator. The installer sends the credential to Connection Server for authentication. The following authentication mechanisms are supported:

- DIGEST-MD5. This mechanism requires that the Active Directory policy **Store password using reversible encryption** is enabled. For details about enabling this policy, see "Enable Reversible Password Encryption," on page 45.

- Kerberos. To use this mechanism, you must specify the Active Directory server's address using the **-k** command line option. Also, for the required parameter **-b**, you must provide Connection Server's fully qualified domain name (FQDN) using the real host name and domain name. You cannot use an IP address or a mapped FQDN.

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 and RHEL 7.1 for vGPU," on page 27, "Configure RHEL 6.6 for vDGA," on page 33, or "Configure RHEL 7.1 for vSGA," on page 36. 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 14.

Prerequisites

- Verify that the Linux guest operating system is prepared for desktop use. See "Prepare a Linux Machine for Remote Desktop Deployment," on page 14.
- Familiarize yourself with the Horizon Agent installer script for Linux. See "install_viewagent.sh Command Line Options," on page 43.
- Verify that the View Connection Server administrator that you provide with the installation command has the **Agent Registration Administrators** or **Administrators** role in View Administrator. Specifically, the **Register Agent** privilege is required to register Horizon Agent with View Connection Server. **Agent Registration Administrators** is a restricted role that provides this minimum privilege.

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-viewagent-linux-x86_64-*y.y.y-xxxxxxx*.tar.gz for 64-bit Linux or VMware-viewagent-linux-*y.y.y-xxxxxxx*.tar.gz for 32-bit Linux, where *y.y.y* is the version number and *xxxxxxx* is the build number.

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

   For example:

   ```
   tar -xzvf <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 43 for a list of the command line options.

   For example:

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

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

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

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

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

5 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 machine is registered with View Connection Server, and the `viewagent` service is started. You can verify that the service is started by running the `#service viewagent status` command.

If an `Agent Registration Failed` message appears, see "Troubleshooting Horizon Agent Registration Failure for a Linux Machine," on page 99.

If another Horizon Agent issue occurs, see Chapter 8, "Troubleshooting Linux Desktops," on page 99.

NOTE The `install_viewagent.sh` script installs JRE 8 in the Horizon Agent installation directory, `/usr/lib/vmware/viewagent/jre`, on the Linux virtual machine. Horizon Agent requires JRE 8.

What to do next

Deploy the virtual machine in a desktop pool. See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

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 6-1. `install_viewagent.sh` Mandatory Parameters

Mandatory Parameter	Description
-b	FQDN or IP address of the View Connection Server instance if the authentication mechanism is DIGEST-MD5. If the authentication mechanism is Kerberos, specify the real FQDN and not the IP address or a mapped FQDN.
-d	Domain name of the Broker administrator. For example: `mydomain.com`
-u	User name of the View Connection Server administrator. For example: `myadmin` The administrator must have the **Agent Registration Administrators** or **Administrators** role in View Administrator. Specifically, the **Register Agent** privilege is required to register Horizon Agent with View Connection Server. **Agent Registration Administrators** is a restricted role that provides this minimum privilege.

Table 6-2. `install_viewagent.sh` Password Parameter

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

Table 6-3. `install_viewagent.sh` Optional Parameters

Optional Parameters	Description
-n	Machine name that is registered to View Connection Server. By default, the Horizon Agent installer uses the host name. You can specify your own name.
-s	Subject DN of the self-signed certificate. By default, the Horizon Agent installer generates a self-signed certificate with the following Subject DN value: `'/C=US/ST=California/O=VMware/CN=`*`Hostname.DomainName`*`'`. The default CN specifies the host name and domain name of the machine. You can specify your own Subject DN.
-m yes	Enable smart card redirection.
-j	JMS SSL keystore password. If not specified, the installer will generate a random password.
-k	The Active Directory server's address of the Broker administrator. This is required only for Kerberos authentication.
-r	Reboot the operating system after installation. The allowed values are **`yes`** and **`no`**. The default is **`no`**.
-K	The Active Directory server's address of the Broker host. It is required only for Kerberos authentication when the broker host and the broker admin are from different domains.
-B	Domain name of the Broker host. It is required only for Kerberos authentication when the broker host and the broker admin are from different domains.

Table 6-4. Examples of `install_viewagent.sh` Parameters

Condition	Examples
Broker host and Broker admin are from same domain	■ Digest-MD5. **`sudo install_viewagent.sh -A yes -n mymachine -b mybroker.mydomain.com`** **`-d mydomain.com -u myadmin -p Pa\$ssword`** ■ Kerberos. **`sudo install_viewagent.sh -A yes -n mymachine -b mybroker.mydomain.com`** **`-d mydomain.com -u myadmin -p Pa\$ssword -k mydomain-host.mydomain.com`**
Broker host and Broker admin are from different domains	■ MD5-Digest. **`sudo install_viewagent.sh -A yes -n mymachine -b mybroker.mydomain1.com`** **`-d mydomain2.com -u`** **`myadmin@mydomain2.com-p Pa\$ssword`** ■ Kerberos. **`sudo install_viewagent.sh -A yes -n mymachine -b mybroker.mydomain1.com`** **`-B mydomain1.com -K mydomain1-host.mydomain1.com -d mydomain2.com -u myadmin`** **`-p Pa\$ssword -k mydomain2-host.mydomain2.com`**

Enable Reversible Password Encryption

During installation, Horizon Agent must authenticate itself to View Connection Server as a View administrator. If the authentication mechanism is DIGEST-MD5, you must first enable the Active Directory (AD) setting **Store password using reversible encryption** for the administrator's account.

Procedure

1. On the AD server, open **Administrative Tools > Active Directory Users and Computers**.
2. Right-click the View administrator's account and select **Properties**.
3. Select the Account tab.
4. In Account options, select the **Store password using reversible encryption** setting.
5. Reset the password for the View administrator.

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 rui.key and the certificate to rui.crt .
    b. Run `sudo chmod 550 /etc/vmware/ssl`.
    c. Copy the rui.crt and rui.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.

Create a Desktop Pool That Contains Linux Virtual Machines

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

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

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

Prerequisites

- Verify that Horizon Agent is installed on the Linux guest operating systems. See "Install Horizon Agent on a Linux Virtual Machine," on page 41.
- Verify that the Linux virtual machines are registered in View Connection Server. In View Administrator, select **View Configuration > Registered Machines** and select the **Others** tab. Verify that each machine's state is `Available`.
- On Ubuntu 14.04 machines, verify that user switching is disabled. See step 9 in "Prepare a Linux Machine for Remote Desktop Deployment," on page 14.

Procedure

1. In View Administrator, add a manual desktop pool.

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

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

   Important Add Linux virtual machines only. If you add Windows virtual machines, the Linux desktops in the pool will be unavailable.
6. Entitle users to the machines in the desktop pool.

   In View Administrator, select the desktop pool, select **Entitlements > Add entitlement**, and add users or groups.

   As a best practice, make sure that the users do not have administrative privileges in the Linux guest operating system.

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

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..

Most of the install parameters that were specified in the previous install will be remembered. You can optionally specify the following parameters:

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**.

Prerequisites

- Verify that the `VMwareBlastServer` process is not running.

  To stop this process, make sure 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-xxxxxxx*.tar.gz for 64-bit Linux or VMware-viewagent-linux-*y.y.y-xxxxxxx*.tar.gz for 32-bit Linux, where *y.y.y* is the version number and *xxxxxxx* is the build number.

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

   For example:

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

3. Navigate to the tar ball folder.

4. Run the `install_viewagent.sh` script.

   For example:

   ```
   sudo ./install_viewagent.sh -A yes
   ```

Uninstalling and Reinstalling Horizon 7 for Linux Machines

To uninstall Horizon 7 for Linux on a virtual machine, you must remove the unmanaged machine from the desktop pool, remove the registered machine from View, uninstall Horizon Agent, and remove configuration files. You can reinstall Horizon 7 for Linux by running the Horizon Agent installation script.

Remove an Unmanaged Machine from a Manual Desktop Pool

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

Procedure

1. In View Administrator, select **Catalog > Desktop Pools**.
2. Double-click the pool ID of the manual pool.

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

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

7. Click **OK**.

The unmanaged machines are removed from the pool.

Remove Registered Machines from View

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

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

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

Prerequisites

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

Procedure

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

   You can select only machines that are not being used by a desktop pool.
4. Click **OK** to confirm.

Uninstall Horizon Agent on a Linux Virtual Machine

To uninstall Horizon Agent on a Linux virtual machine, you use the `uninstall_viewagent.sh` script. You also must remove the Horizon 7 for Linux configuration files from the machine.

Prerequisites

- Verify that the machine was removed from the desktop pool and from Horizon 7 . See "Remove an Unmanaged Machine from a Manual Desktop Pool," on page 47 and "Remove Registered Machines from View," on page 48.

Procedure

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

   For example:

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

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

2. Manually delete the Horizon 7 for Linux configuration files, located in the `/etc/vmware/` directory.

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

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

Reinstall Horizon Agent on a Linux Virtual Machine

You use different approaches to reinstall Horizon Agent on a Linux virtual machine, depending on whether you use an existing Horizon 7 for Linux configuration or reinstall with a fresh configuration.

You can reinstall Horizon Agent with an existing configuration if the Horizon 7 for Linux configuration files are present on the virtual machine and the machine is registered in View Connection Server.

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

Prerequisites

- Verify whether the Horizon 7 for Linux configuration files are present on the machine.

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

- Verify whether the Linux virtual machine is registered in View Connection Server. In View Administrator, select **View Configuration > Registered Machines** and select the **Others** tab.

- If the configuration files and registered machine name were removed, familiarize yourself with the steps for performing a fresh installation of Horizon Agent. See "Install Horizon Agent on a Linux Virtual Machine," on page 41.

Procedure

- To reinstall with an existing Horizon 7 for Linux configuration, you can run the Horizon Agent installation script without parameters.

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

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

- To reinstall with a new Horizon 7 for Linux configuration, run the Horizon Agent installation script with the required parameters.

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

How to Perform Power Operations on Linux Desktops from vSphere

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

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

Gather Information About Horizon 7 for Linux Software

To help manage and troubleshoot the Horizon 7 for Linux software, you can gather information about a Horizon Agent installation on a Linux virtual machine.

Procedure

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

  Run the following command:

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

  The output displays the Horizon Agent version. For example:

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

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

- Gather information about Horizon Agent software.

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

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

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

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

Bulk Deployment of Horizon 7 for Linux Desktops

7

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 51
- "Create a Virtual Machine Template for Cloning Linux Desktop Machines," on page 53
- "Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops," on page 54
- "Sample Script to Clone Linux Virtual Machines," on page 55
- "Sample Script to Join Cloned Virtual Machines to AD Domain," on page 59
- "Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH," on page 61
- "Sample Script to Install Horizon Agent on Linux Virtual Machines," on page 65
- "Sample Script to Install Horizon Agent on Linux Virtual Machines Using SSH," on page 70
- "Sample Script to Upload Configuration Files to Linux Virtual Machines," on page 75
- "Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH," on page 79
- "Sample Script to Upgrade Horizon Agent on Linux Desktop Machines," on page 83
- "Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH," on page 87
- "Sample Script to Perform Operations on Linux Virtual Machines," on page 92
- "Sample Script to Delete Machines from the Connection Server LDAP Database," on page 96

Overview of Bulk Deployment of Linux Desktops

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

For 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.	■ For Ubuntu, you need to install the SSH server with the command `sudo apt-get install openssh-server`. 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 `pscp.exe` and `plink.exe` 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.

See "Create a Virtual Machine Template for Cloning Linux Desktop Machines," on page 53.

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 55.

4 Join the cloned VMs to the Active Directory (AD) domain.

See "Sample Script to Join Cloned Virtual Machines to AD Domain," on page 59 or "Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH," on page 61.

5 Install Horizon Agent in virtual machines.

See "Sample Script to Install Horizon Agent on Linux Virtual Machines," on page 65 or "Sample Script to Install Horizon Agent on Linux Virtual Machines Using SSH," on page 70.

6 Update configuration options in virtual machines.

See "Sample Script to Upload Configuration Files to Linux Virtual Machines," on page 75 or "Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH," on page 79.

7 Create a desktop pool.

See "Create a Desktop Pool That Contains Linux Virtual Machines," on page 46.

Occasionally, you might need to upgrade Horizon Agent on the Linux virtual machines. For a sample script to upgrade Horizon Agent on virtual machines, see "Sample Script to Upgrade Horizon Agent on Linux Desktop Machines," on page 83 or "Sample Script to Upgrade Horizon Agent on Linux Virtual Machines Using SSH," on page 87.

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 92. This script can delete virtual machines from vCenter Server. To delete desktop machines from View, see "Sample Script to Delete Machines from the Connection Server LDAP Database," on page 96.

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 8.
- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing guest operating systems. See "Creating and Preparing Virtual Machines" in the *Setting Up Desktop and Application Pools in View* document.
- Familiarize yourself with the recommended video memory (vRAM) values for the monitors you will use with the virtual machine. See "System Requirements for Horizon 7 for Linux," on page 8.
- Familiarize yourself with the steps for AD integration. See Chapter 3, "Setting Up Active Directory Integration for Linux Desktops," on page 19.
- If required, familiarize yourself with the steps to configure options using the View configuration files. See Chapter 4, "Configuration Options for Linux Desktops," on page 23.
- If you plan to set up graphics, familiarize yourself with the steps. See Chapter 5, "Setting Up Graphics for Linux Desktops," on page 27.

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 SLED 12, install VMware Tools .

RHEL 7, CentOS 7, and SLED 12 have Open VM Tools installed by default.

8 If the Linux distribution is RHEL 7, CentOS 7, or SLED 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 additional configurations using the View configuration files.

12 Perform the steps to set up graphics.

13 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 54. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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)
    $vmConfigSpec = 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 54. 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"

.DESCRIPTION
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
        $input = 
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

    [Console]::ResetColor()
    return $input
}
#------------------------- Handle input -------------------------
"_____________________________________________________"
$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
"_____________________________________________________"
$adUser = 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 54. 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

.DESCRIPTION
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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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
""
"`nPlease type the AD user password."
[Console]::ForegroundColor = "Yellow"
"Plase 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 Install Horizon Agent on Linux Virtual Machines

You can customize and use the following sample script to install 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.

Before you run the script, make sure that Horizon Agent is not already installed on the 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 54. 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 VM
- User password for the Linux VM
- Horizon Agent tar ball path

- IP address of View Connection Server
- Administrator login name for View Connection Server
- Administrator password for View Connection Server
- Domain name of the View Connection Server administrator
- The AD server's address if Kerberos is the authentication method for Horizon Agent to register the Linux machine with View Connection Server.

Script Content

```
<#
Install Linux Agent

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

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

    [Console]::ResetColor()
    return $input
}
#------------------------- Handle Input -------------------------
"------------------------------------------------------"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon 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
}
$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
"-----------------------------------------------------"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----------------------------------------------------"
$brokerAddress = GetInput -prompt 'Type the View Connection Server address'  -IsPassword $false
$brokerAdminName = GetInput -prompt 'Type the View Connection Server Admin user name' -
IsPassword $false
"`nPlease type the View Connection Server Admin user password."
"Plase note that special character in password may not work with the script"
$brokerAdminPassword = GetInput -prompt 'Your broker admin password' -AsSecureString  -
IsPassword $true
$domainName = GetInput -prompt 'Type the View Connection Server Admin user domain name' -
IsPassword $false
$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
}
$kdc = GetInput -prompt 'AD address for Kerberos authentication to register. Please type Enter
to skip for MD5-Digest' -IsPassword $false
"-----------------------------------------------------"
#$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"

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

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

    #Check the uploaded installer md5sum
    $cmd = "md5sum VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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";
        $cmd = "tar -xzvf VMware-viewagent-linux-*.tar.gz"
        Write-Host "Run cmd '$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

        #Run the installation command.
        $cmd = "cd VMware-viewagent-linux-* && sudo ./install_viewagent.sh -A yes -n $VMName -b
$brokerAddress -d $domainName -u $brokerAdminName -p $brokerAdminPassword -m $installSmartcard"
        if (!([string]::IsNullOrEmpty($kdc)))
        {
        $cmd = $cmd + " -k $kdc"
        }
        Write-Host "Run install cmd in VM '$VMName' with user '$guestUser'"
        Invoke-VMScript  -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

        $cmd = "sudo shutdown -r +1&"
        Write-Host "Reboot to apply the Horizon 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
--------------------------------------------------
Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character in password may not work with the script
```

```
Your broker admin password: *******
Type the View Connection Server Admin user domain name: domain1
Install the Smartcard redirection feature ("yes" or "no"): no
AD address for Kerberos authentication to register. Please type Enter to skip for MD5 Digest:
10.117.45.1
```

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

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

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

Before you run the script, make sure that Horizon Agent is not already installed on the 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 54. 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
- User login name for the Linux VM
- User password for the Linux VM
- Horizon Agent tar ball path
- IP address of View Connection Server
- Administrator login name for View Connection Server
- Administrator password for View Connection Server
- Domain name of the View Connection Server administrator
- The AD server's address if Kerberos is the authentication method for Horizon Agent to register the Linux machine with View Connection Server.

Script Content

```
<#
Install Linux Agent via SSH

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

#>
#------------------------- Functions -------------------------
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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
"______________________________________________________"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon 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
}
$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
"______________________________________________________"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"______________________________________________________"
$brokerAddress = GetInput -prompt 'Type the View Connection Server address'  -IsPassword $false
$brokerAdminName = GetInput -prompt 'Type the View Connection Server Admin user name' -
IsPassword $false
"`nPlease type the View Connection Server Admin user password."
[Console]::ForegroundColor = "Yellow"
"Plase note that special character should be escaped. For example, $ should be \$"
[Console]::ResetColor()
$brokerAdminPassword = GetInput -prompt 'Your broker admin password' -AsSecureString  -
IsPassword $true
```

```
$domainName = GetInput -prompt 'Type the View Connection Server Admin user domain name' -
IsPassword $false
$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
}
$kdc = GetInput -prompt 'AD address for Kerberos authentication to register. Please type Enter
to skip for MD5-Digest' -IsPassword $false
"-----------------------------------------------------"
#$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"

    $cmd = "rm -rf VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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
    $cmd = "md5sum VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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";
        $cmd = "tar -xzf VMware-viewagent-linux-*.tar.gz"
        Write-Host "Run cmd '$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

        #Run the installation command.
        $cmd = "cd VMware-viewagent-linux-* && sudo ./install_viewagent.sh -r yes -A yes -n 
$VMName -b $brokerAddress -d $domainName -u $brokerAdminName -p $brokerAdminPassword -m 
$installSmartcard"
        if (!([string]::IsNullOrEmpty($kdc)))
        {
        $cmd = $cmd + " -k $kdc"
        }
        Write-Host "Run install cmd in VM '$VMName' with user '$guestUser'"
        RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
        Write-Host -ForeGroundColor Yellow "Linux Agent installer will reboot the Linux VM after
installation, 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
--------------------------------------------------
Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character should be escaped. For example, $ should be \$
Your broker admin password: *******
Type the View Connection Server Admin user domain name: domain1
Install the Smartcard redirection feature ("yes" or "no"): no
AD address for Kerberos authentication to register. Please type Enter to skip for MD5 Digest: 
10.117.45.1
```

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

Sample 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 54. 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
        $input = 
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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 54. 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
        $input = 
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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'
$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
"-----------------------------------------------------"

$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 54. 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

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

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

#------------------------- Handle Input -------------------------
"-----------------------------------------------------"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon 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
}
$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
"-----------------------------------------------------"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----------------------------------------------------"
$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"

    $cmd = "rm -rf VMware-viewagent-linux-*"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript  -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
```

```
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

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

    #Check the uploaded installer md5sum
    $cmd = "md5sum VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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";
        $cmd = "tar -xzvf VMware-viewagent-linux-*.tar.gz"
        Write-Host "Run cmd '$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-viewagent-linux-* && sudo ./install_viewagent.sh -A yes -m
$installSmartcard"
        Write-Host "Run upgrade cmd in VM '$VMName' with user '$guestUser'"
        Invoke-VMScript  -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

        $cmd = "sudo shutdown -r +1&"
        Write-Host "Reboot to apply the Horizon 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
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 54. 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

Script Content

```
<#
Upload the Linux Agent installer tar ball using SSH 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
        $input = 
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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
"_____________________________________________________"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon 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
}
$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
"_____________________________________________________"
```

```
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----------------------------------------------------"
$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"

    $cmd = "rm -rf VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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
    $cmd = "md5sum VMware-viewagent-linux-*"
    Write-Host "Run cmd '$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";

        $cmd = "tar -xzf VMware-viewagent-linux-*.tar.gz"
        Write-Host "Run cmd '$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-viewagent-linux-* && sudo ./install_viewagent.sh -r yes -A yes -m 
$installSmartcard"
        Write-Host "Run upgrade cmd in VM '$VMName' with user '$guestUser'"
        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
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 delete the machines from View, see the sample script "Sample Script to Delete Machines from the Connection Server LDAP Database," on page 96.

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 54. 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
        $input = 
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR($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 ---------------------
"-----------------------------------------------------"
$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
"-----------------------------------------------------"
$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, reatart 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.

Sample Script to Delete Machines from the Connection Server LDAP Database

You can customize and use the following sample script to delete machines that are not in a desktop pool from the Connection Server LDAP database.

This is a PowerShell script and does not require PowerCLI to be installed. To run the script, log in to View Connection Server as an administrator, create the script and input files, and run the script from a PowerShell window. If a machine in the input file is in a desktop pool, this script displays a message and does not delete it from the LDAP database.

Note Before running the script, take a snapshot of the Connection Server virtual machine in case it becomes necessary to undo the changes from running the script.

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

The input file is of type `csv` and contains the names of the machines to be deleted. The following example shows what the input file may contain.

```
MachineName
linux-001
linux-002
linux-003
```

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

Script Content

```
<#
bulk Delete registered machine items from broker LDAP database.
#>

$csvFile = '.\DeleteMachineItems.csv'

#check if file exists
```

```
if (!(Test-Path $csvFile))
{
    write-host  -ForeGroundColor Red "CSV File $CSVFile not found"
    exit
}

function IsNameInCsv($name)
{
    $csvData = Import-CSV $csvFile;
    foreach ($line in $csvData)
    {
        if($line.MachineName -eq $name)
        {
            Write-Host "$name exists in csv"
            return $true;
        }
    }
    Write-Host "$name does not exists in csv"
    return $false;
}

$obj = [adsi]'LDAP://localhost:389/OU=Servers,DC=vdi,DC=vmware,DC=int';
foreach ($child in $obj.psbase.Children)
{
  $machineName = $child.get("pae-DisplayName");
  if(IsNameInCsv($machineName))
  {
      $machineAttributes = $child  | get-member;
      $machineAttributes = $machineAttributes -join ' ';
      if ($machineAttributes -like "*pae-MemberDNOf*")
      {
          $desktopPool = $child.get("pae-MemberDNOf");
          Write-Host "Skip $machineName, which is included by desktop pool $desktopPool";
      }
      else
      {
          Write-Host "Delete $machineName, which is in csv file and not included by any desktop 
pool";
          $child.distinguishedName;
          $child.DeleteObject(0);
      }

  }
  else
  {
      Write-Host "Skip $machineName, which is not in csv file";
  }
  Write-Host '---------------------------------------';
}
```

Troubleshooting Linux Desktops

8

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 99
- "Troubleshooting Horizon Agent Registration Failure for a Linux Machine," on page 99
- "Troubleshooting an Unreachable Horizon Agent on a Linux Machine," on page 100
- "Troubleshooting Horizon Agent on a Linux Machine That Is Not Responding," on page 102
- "Troubleshooting Copy and Paste between Remote Desktop and Client Host," on page 102
- "Configuring the Linux Firewall to Allow Incoming TCP Connections," on page 102

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 Horizon Agent Registration Failure for a Linux Machine

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

Problem

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

Cause

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

- The View Connection Server FQDN, domain name, user name, or password that was provided during Horizon Agent installation was incorrect.
- The View administrator user does not have the **Agent Registration Administrators** or **Administrators** role in View Administrator. Specifically, the **Register Agent** privilege is required to register Horizon Agent with View Connection Server. **Agent Registration Administrators** is a restricted role that provides this minimum privilege.
- The View administrator password contains a special character, such as $, that was not escaped with a backslash.
- Reversible password encryption was not enabled for the administrator's Active Directory (AD) account. For details about enabling this policy, see "Enable Reversible Password Encryption," on page 45.

Solution

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

   For example: `ab\$cdef`
4. On the Linux machine, open the Horizon Agent registration log file and look for error messages.

   ```
   /usr/lib/vmware/viewagent/viewagent-registration.log
   ```
5. If the authentication mechanism is DIGEST-MD5, that is, the **-k** option is not specified, ensure that the AD policy **Store password using reversible encryption** is enabled for the user that you specified.
6. If the authentication mechanism is Kerberos, ensure that the **-k** option specifies the AD server's correct address and that the **-b** option specifies the Connection Server instance's full computer name, including the host name and the domain name.
7. Install Horizon Agent on the Linux virtual machine again.

Troubleshooting an Unreachable Horizon Agent on a Linux Machine

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

Problem

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

Cause

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

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

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

Solution

1 On the Linux machine, open the Horizon Agent log file.

```
/usr/lib/vmware/viewagent/viewagent-debug.log
```

2 Look for messages that indicate the hostname of the View Connection Server instance cannot be resolved.

For example:

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

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

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

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

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

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

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

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

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

The machine's hostname is displayed.

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

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

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

```
ping <your hostname>
```

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

Troubleshooting Horizon Agent on a Linux Machine That Is Not Responding

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

Problem

The Horizon 7 for Linux desktop is not accessible, and Horizon Agent is not responding.

Cause

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

Solution

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

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

2. Verify that the X session is active.

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

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

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.

Index

A

Active Directory, integrating with Linux **19**

B

bulk deployment
  create a virtual machine template for cloning **53**
  input file for sample scripts **54**
  overview **51**

C

configuration options
  audio out **23**
  clipboard redirection **23**
  left-handed mouse **23**
  lossless PNG mode **23**
  single sign-on (SSO) **23**
configuring **45**

D

desktop pool creation, Linux virtual machines **46**
DIGEST-MD5 **41**

F

firewalls, Linux desktops **102**

G

glossary **5**
graphics **27**

H

Horizon 7 for Linux
  bulk deployment **51**
  gathering information **50**
  reinstalling **49**
  troubleshooting Horizon Agent that is not responding **102**
  uninstalling **48**
Horizon Agent
  installing on a Linux virtual machine **41**
  Linux installation command options **43**
  reinstalling on a Linux machine **49**
  uninstalling on a Linux machine **48**
  upgrading on a Linux virtual machine **47**
Horizon for Linux
  reinstalling **47**
  uninstalling **47**

I

install_viewagent.sh, command usage and options **43**
installing **16**
intended audience **5**

K

Kerberos **41**

L

Linux desktops
  configuring **23**
  features **7**
  features and requirements **7**
  installing Horizon Agent **41**
  setting up AD integration **19**
Linux installation
  Horizon Agent fails to register **99**
  Horizon Agent is unreachable **100**
Linux systems
  adding to desktop pools **46**
  collecting diagnostic information **99**
  configuring the firewall **102**
  desktops **5**
  overview of desktop setup steps **7**
  preparing for desktop use **14**
  vSphere console display **25**
Linux virtual machines, preparing for desktop deployment **13**

N

NVIDIA GRID vGPU
  installing the VIB for Linux desktops **27, 36**
  virtual GPU Types **30**
NVIDIA display driver
  installing **30, 34**
  verify the installation **32, 36**
NVIDIA vGPU, configuring a shared PCI device **29**

O

OpenLDAP **19**

P

physical computers, removing from a pool **47**
power operations from vSphere, Linux desktops **50**

R
registered machines
  removing **48**
  removing from View **48**
removing registered machines **48**
reversible password encryption **45**
RHEL
  configuring for vDGA **33**
  configuring for vGPU **27**
  configuring for vSGA **36**

S
sample script
  bulk delete desktop machines **96**
  clone desktop machines using SSH **61**
  install Horizon Agent using SSH **65**
  perform operations on VMs **92**
  to clone desktop machines **55, 59**
  to install Horizon Agent **70**
  to upgrade Horizon Agent **83**
  upgrade Horizon Agent using SSH **87**
  upload configuration files **75**
  upload configuration files using SSH **79**
sample scripts, input file **54**
single sign-on **20**
smart card redirection **20**

T
troubleshooting **99, 102**

U
unmanaged machines, removing from a pool **47**

V
vDGA
  adding PCI device to a Linux machine **33**
  enabling DirectPath I/O on a host **33**
vDGA (Virtual Dedicated Graphics Acceleration), for RHEL 6.6 **33**
vGPU, configuring RHEL 6.6 and RHEL 7.1 **27**
View Agent, Linux operating system requirements **8**
virtual machines, creating for Linux desktops **13**
VMware Tools, configuring on a Linux guest **14**
vSGA
  configuring 3D settings on a Linux video card **38**
  for RHEL 7.1 **36**
  installing the VIB for Linux desktops **27, 36**
  verify on a Linux machine **39**
vSphere console, suppress display **25**

W
Winbind **19**