Setting Up Horizon 7 for Linux Desktops

DEC 2019

VMware Horizon 7 7.11

vmware®

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

https://docs.vmware.com/

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

docfeedback@vmware.com

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

Contents

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 for Linux desktop. The information includes preparing the Linux guest operating system, installing Horizon Agent on the virtual machine, and configuring the machine in Horizon Console 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 data center operations.

Features and System Requirements

1

With Horizon 6.2.x or later, users can connect to remote desktops that run the Linux operating system.

This chapter includes the following topics:

Features of Horizon Linux Desktops

The following list presents the key features supported for Horizon Linux desktops.

Supported Features on Linux Desktops

Active Directory Integration

Instant-cloned desktops running the following Linux distributions can perform an offline domain join with Active Directory using PowerBroker Identity Services Open (PBISO).

- Ubuntu 16.04 and 18.04
- SLED/SLES 12.x

See the PowerBroker Identity Services Open (PBISO) Authentication section in Integrating Linux with Active Directory for more information.

Instant-cloned desktops running the following Linux distributions can perform an offline domain join with Active Directory using Samba.

- Ubuntu 16.04 and 18.04
- RHEL 7.3 and 8.0

Audio-in

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

	application for the audio input. This feature is supported on the following Linux distributions. ■ Ubuntu 16.04 x64 with MATE or Gnome Flashback (Metacity) desktop environment ■ Ubuntu 18.04 x64 with MATE or Gnome Ubuntu desktop environment ■ RHEL 7.x Workstation x64 with KDE or Gnome desktop environment ■ RHEL 8.0 Workstation x64 with Gnome desktop environment ■ SLED/SLES 12.x SP3 x64
Audio-out	Audio output redirection is supported. This feature is enabled by default. To disable this feature, you must set the `RemoteDisplay.allowAudio` option to **`false`**. When accessed using Chrome and Firefox browsers, VMWare Horizon HTML Access provides audio-out support for Linux desktops.
Automated Full-Clone Desktop Pool	You can create automated full-clone desktop pools for Linux desktops.
Client Drive Redirection	When you enable the Client Drive Redirection (CDR) feature, your local system's shared folders and drives become available for you to access. You use the `tsclient` folder that is located in your home directory in the remote Linux desktop. To use this feature, you must install the CDR components.
Clipboard Redirection	With the clipboard redirection feature, you can copy and paste a rich text or a plain text between a client host and a remote Linux desktop. You can set the copy/paste direction and the maximum text size using Horizon Agent options. This feature is enabled by default. You can disable it during installation.
FIPS 140-2 Mode	The Federal Information Processing Standard (FIPS) 140-2 mode support, although not yet validated with the NIST Cryptographic Module Validation Program (CMVP), is available for Linux desktops. The Horizon 7 Agent for Linux implements cryptographic modules that are designed for FIPS 140-2 compliance. These modules were validated in operational environments listed in CMVP certificate #2839 and #2866, and were ported to this platform. However, the CAVP and CMVP testing requirement to include the new operational environments in VMware's NIST CAVP and CMVP certificates remains to be completed on the product roadmap. **Note** The Transport Layer Security (TLS) protocol version1.2 is required to support FIPS 140-2 mode.

Help Desk Tool

Horizon Help Desk Tool is a Web application that you can use to troubleshoot Linux desktop sessions. You can use Horizon Help Desk Tool to get the status of Horizon 7 user sessions and to perform troubleshooting and maintenance operations. See Using Horizon Help Desk Tool in Horizon Console.

Horizon Smart Policies

You can use VMware Dynamic Environment Manager™ 9.4 or later to create Horizon Smart Policies that control the behavior of the USB redirection, clipboard redirection, and client drive redirection features on specific remote Linux desktops. See Using Smart Policies.

H.264 Encoder

H.264 can improve the Blast Extreme performance for a Horizon desktop, especially under a low-bandwidth network. If the client system has H.264 disabled, Blast Extreme automatically falls back to JPEG/PNG encoding.

The H.264 encoder includes both hardware H.264 support and software encoder support. The hardware H.264 support has the following requirements.

- The vGPU is configured with an NVIDIA graphics card.
- The NVIDIA driver 384 series or later is installed in the NVIDIA graphics card.

When the system meets the preceding requirements, Horizon 7 for Linux uses the hardware H.264 encoder. Otherwise, the software H.264 encoder is used.

Instant-Clone Floating Desktop Pool

You can create instant-clone floating desktop pools for Linux desktops. This feature is supported only on systems with the following Linux distributions installed.

- Ubuntu 16.04 and 18.04
- RHEL 7.1 or later
- RHEL 8.0
- SLED/SLES 12.x

For more information, see Create an Instant-Clone Floating Desktop Pool for Linux.

K Desktop Environment

The K Desktop Environment (KDE) is supported on the following Linux distributions.

- CentOS 6.x and 7.x
- RHEL 6.x and 7.x
- Ubuntu 16.04 and 18.04

Keyboard Layout and Locale Synchronization

This feature specifies whether to synchronize a client's system locale and current keyboard layout with the Horizon Linux Agent desktops. When this setting is enabled or not configured, synchronization is allowed. When this setting is disabled, synchronization is not allowed.

This feature is supported only for VMware Horizon for Windows, and only for the English, French, German, Japanese, Korean, Spanish, Simplified Chinese and Traditional Chinese locales.

Lossless PNG

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

Manual Desktop Pool

Machine source.

- Managed Virtual Machine - Machine source of the vCenter virtual machine. A managed virtual machine is supported for new and upgrade deployment.
- Unmanaged Virtual Machine - Machine source of other sources. An unmanaged virtual machine is only supported when the upgrade is from an unmanaged virtual machine deployment.

Note To ensure the best possible performance, do not use an unmanaged virtual machine.

MATE Desktop Environment

The MATE Desktop Environment is supported on the following Linux distributions.

- Ubuntu 16.04
- Ubuntu 18.04

Multiple Monitors

- vDGA/vGPU desktop supports a maximum resolution of 2560x1600 on four monitors.
- 2D desktop on VMware vSphere® 6.0 or later supports a maximum resolution of 2048x1536 on four monitors or a maximum resolution of 2560x1600 on three monitors.

For Ubuntu 16.04 and 18.04, you must use Gnome, KDE, or the MATE desktop environment to use the multiple monitors feature. See http://kb.vmware.com/kb/2151294 for more information.

For SLES 12 SP1, you must use the default package with kernel level kernel-default-3.12.49-11.1. If you upgraded the package, the multi-monitor feature fails and the desktop is shown in one monitor.

Beginning with VMware Horizon HTML Access™ version 5.0, the multi-monitor feature is supported in Horizon 7 for Linux desktops.

Network Intelligence Support for VMware Blast

The Network Intelligence transport is supported for VMware Blast. This feature is enabled by default.

When User Datagram Protocol (UDP) is enabled, Blast establishes both Transmission Control Protocol (TCP) and UDP connections. Based on the current network conditions, Blast dynamically selects one of the transports for transmitting data to provide the best user experience. For example, in a local area network, TCP performs better than UDP, and so Blast selects TCP to transport data. Similarly, in a wide area network (WAN), UDP performance is better than TCP and Blast selects the UDP transport in that environment.

If one of the inline components used does not support UDP, Blast establishes a TCP connection only. For example, if your connection is using the Blast Security Gateway component of the Horizon Connection Server or Security Server, only a TCP connection is established. Even if both client and agent enabled UDP, the connection uses TCP because Blast Security Gateway does not support UDP. If users are connecting from outside the corporate network, the UDP component requires the VMware Unified Access Gateway (formerly called Access Point), which supports UDP.

Use the following information to establish a UDP-based Blast connection.

- If the client connects to a Linux desktop directly, enable the UDP in both the client and agent. UDP is enabled by default in both the client and agent.
- If the client connects to a Linux desktop using Unified Access Gateway, enable UDP in the client, agent, and Unified Access Gateway.

Session Collaboration

With the Session Collaboration feature, users can invite other users to join an existing remote Linux desktop session, or you can join a collaborative session when you receive an invitation from another user. This feature is supported only on remote Linux desktops with the following Linux distributions installed.

- Ubuntu 18.04 with Gnome desktop environment
- RHEL 7.5 or later with Gnome Classic desktop environment
- RHEL 8.0 with Gnome Classic desktop environment

Single Sign-on

Single sign-on (SSO) is supported on the following Linux distributions.

- RHEL 8.0/7.x/6.x Workstation x64
- CentOS 8.0/7.x/6.x x64
- SLED/SLES 12.x SP3/SP2/SP1
- Ubuntu 18.04/16.04 x64

Smart Card Redirection

Smart card redirection is supported on the following Linux distributions.

- RHEL 8.0
- RHEL 7.1 and later
- RHEL 6.6 and later
- Ubuntu 18.04/16.04
- SLED/SLES 12.x SP3

This feature supports Personal Identity Verification (PIV) cards and Common Access Cards (CAC). For more information, see Setting Up Smart Card Redirection.

True SSO Support

True SSO is supported on the following Linux distributions.

- RHEL 7.x/8.0
- CentOS 7.x/8.0
- SLED/SLES 12.x SP3
- Ubuntu 18.04/16.04

For more information, see Setting Up True SSO for Linux Desktops.

USB Redirection

The USB Redirection feature gives you access to locally attached USB devices from remote Linux desktops. You must install the USB Redirection components and USB VHCI driver kernel module to use the USB feature. Ensure that you have been granted sufficient privileges to use the USB device that you want to redirect.

3Dconnexion Mouse

To begin using your 3Dconnexion mouse, you must install the appropriate device driver and pair the mouse using the Connect USB Device menu on your Linux desktop.

3D Graphics

The 3D graphics feature supports the following combinations of Linux versions and graphics cards:

- vDGA is supported on RHEL 6.x Workstation x64 with NVIDIA GRID K1 or K2 graphics cards.
- vGPU is supported on the Linux distributions and NVIDIA graphics cards listed on https://docs.nvidia.com/grid/9.0/product-support-matrix/index.html.

Limitations of Linux Desktops and Desktop Pools

Linux desktops and desktop pools have the following limitations:

- Virtual Printing, location-based printing, and Real-Time Video are not supported.

- The VMware HTML Access file transfer feature is not supported.

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

Overview of Configuration Steps for Horizon 7 for Linux Desktops

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

2D Graphics - Overview of Configuration Steps

For 2D graphics, take the following steps:

1. Review the system requirements for setting up a Horizon 7 for Linux deployment. See System Requirements For Horizon 7 for Linux.
2. Create a virtual machine in vSphere and install the Linux operating system. See Create a Virtual Machine and Install Linux.
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.
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. See Integrating Linux with Active Directory for more information.
5. Install Horizon Agent on the Linux virtual machine. See Install Horizon Agent on a Linux Virtual Machine.
6. Create a desktop pool that contains the configured Linux virtual machines. See Create a Manual Desktop Pool for Linux.

3D Graphics - Overview of Configuration Steps

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

1. Review the system requirements for setting up a Horizon 7 for Linux deployment. See System Requirements For Horizon 7 for Linux.
2. Create a virtual machine in vSphere and install the Linux operating system. See Create a Virtual Machine and Install Linux.
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.
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. See Integrating Linux with Active Directory for more information.

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 Supported Linux Distributions for vGPU.
   - See Configure RHEL 6.x for vDGA.

6 Install Horizon Agent on the Linux virtual machine. See Install Horizon Agent on a Linux Virtual Machine.

7 Create a desktop pool that contains the configured Linux virtual machines. See Create a Manual Desktop Pool for Linux.

Bulk Deployment

With Horizon Console, you can only deploy Linux virtual machines in a manual desktop pool. With vSphere PowerCLI, you can develop scripts that automate the deployment of a pool of Linux desktop machines. See Chapter 8 Bulk Deployment of Horizon 7 for Manual Desktop Pools.

System Requirements For Horizon 7 for Linux

To install Horizon 7 for Linux, your Linux system must meet certain requirements for the operating system, Horizon 7, and vSphere platform.

Supported Linux Versions for Horizon Agent

The following table lists the Linux operating systems that are supported for Horizon Agent.

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

Linux Distribution	Architecture
Ubuntu 16.04 and 18.04 **Note** You must apply one of the solutions described in VMware KB article http://kb.vmware.com/kb/2151294.	x64
RHEL 6.6, 6.7, 6.8, 6.9, 6.10, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, and 8.0	x64
CentOS 6.6, 6.7, 6.8, 6.9, 6.10, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, and 8.0	x64
NeoKylin 6 Update 1	x64
SLED 12.x SP1/SP2/SP3	x64
SLES 12.x SP1/SP2/SP3	x64

Note Linux agent has dependency packages on some Linux distributions. See Install Dependency Packages for Horizon Agent for more information.

Note On RHEL/CentOS 8.0 systems, Horizon Agent only supports the X11 display server protocol. The Wayland protocol is not supported.

Required Platform and Horizon 7 Software Versions

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

Table 1-2. Required Platform and Horizon 7 Software Versions

Platform and Software	Supported Versions
vSphere platform version	■ vSphere 6.0 U2 or a later release ■ vSphere 6.5 U1 or a later release ■ vSphere 6.7 or later release
Horizon environment	■ Horizon Connection Server 7.11
Horizon Client software	■ Horizon Client 5.3.0 for Android ■ Horizon Client 5.3.0 for Windows ■ Horizon Client 5.3.0 for Linux ■ Horizon Client 5.3.0 for Mac OS X ■ Horizon Client 5.3.0 for iOS (iPad Pro) ■ HTML Access 5.3.0 on Chrome, Firefox, and Internet Explorer ■ Zero clients are not supported.

TCP/UDP Ports Used by Linux Virtual Machines

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

Table 1-3. TCP/UDP Ports Used by Linux Virtual Machines

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

Note For more information on TCP and UDP ports used by clients, see the *Horizon Client and Agent Security* document and the Network Ports in VMware Horizon 7 guide.

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 Horizon 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 an 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 Horizon 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.

Verify the Linux Account Used by Linux Virtual Machines

Table 1-4. Account Name and Account Type lists the account name and account type used by Linux virtual machines.

Table 1-4. Account Name and Account Type

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

Desktop Environment

Horizon 7 for Linux supports multiple desktop environments on different Linux distributions. Table 1-5. Supported Desktop Environments lists the default desktop environments for each Linux distribution and the additional desktop environments supported by Horizon 7 for Linux.

Table 1-5. Supported Desktop Environments

Linux Distribution	Default Desktop Environment	Desktop Environments Supported by Horizon 7 for Linux Desktops
Ubuntu 18.04	Gnome	Gnome Ubuntu, K Desktop Environment (KDE), MATE
Ubuntu 16.04	Unity	Gnome Flashback (Metacity), KDE, MATE
RHEL/CentOS 6.x	Gnome	Gnome, KDE
RHEL/CentOS 7.x	Gnome	Gnome, KDE
RHEL/CentOS 8.0	Gnome	Gnome
SLED 12 SP1/SP2/SP3	Gnome	Gnome
SLES 12 SP1/SP2/SP3	Gnome	Gnome
NeoKylin 6 Update 1	Mate	Mate

To change the default desktop environment used on one of the supported Linux distributions, you must use the following steps and commands appropriate for your Linux desktop.

Note Single sign-on (SSO) for KDE and the MATE Desktop Environment only works when your Linux desktop is using the default greeter (login screen). You must install KDE and MATE using the commands listed in Table 1-6. Commands to Install Desktop Environments.

When using RHEL/CentOS 7.x and Ubuntu 18.04/16.04 distributions, SSO fails to unlock a locked KDE session. You must manually enter your password to unlock the locked session.

1 Install the supported Linux distribution's operating system with the default desktop environment setting.

2 Run the appropriate commands in Table 1-6. Commands to Install Desktop Environments for your specific Linux distribution.

Table 1-6. Commands to Install Desktop Environments

Linux Distribution	New Default Desktop Environment	Commands to Change the Default Desktop Environment
RHEL/CentOS 6.x	KDE	`# yum groupinstall "X Window System" "KDE Desktop"`
RHEL/CentOS 7.x	KDE	`# yum groupinstall "KDE Plasma Workspaces"`
Ubuntu 18.04/16.04	KDE	`# apt install plasma-desktop`
Ubuntu 18.04	MATE 1.225	`# apt install ubuntu-mate-desktop`
Ubuntu 16.04	MATE 1.16	`# apt-add-repository ppa:ubuntu-mate-dev/xenial-mate` `# apt update` `# apt upgrade` `# apt install mate` `# apt install ubuntu-mate-themes`
Ubuntu 16.04	Gnome Flashback (Metacity)	`# apt install gnome-session-flashback`

3 To begin using the new default desktop environment, restart the desktop.

If you enabled SSO on a Linux desktop that has multiple desktop environments installed, use the following information to select the desktop environment to use in an SSO session.

- For Ubuntu 18.04/16.04 and RHEL/CentOS 7.x, use the information in Table 1-7. SSODesktopType Option to set the `SSODesktopType` option in the `/etc/vmware/viewagent-custom.conf` file to specify the desktop environment to use with SSO.

Table 1-7. SSODesktopType Option

Desktop Type	SSODesktopType Option Setting
MATE	SSODesktopType=UseMATE
GnomeUbuntu	SSODesktopType=UseGnomeUbuntu
GnomeFlashback	SSODesktopType=UseGnomeFlashback
KDE	SSODesktopType=UseKdePlasma
GnomeClassic	SSODEsktopType=UseGnomeClassic

- For RHEL/CentOS 6.x, for the SSO login session to use KDE, remove all the desktop startup files, except for the KDE startup file, from the `/usr/share/xsession` directory. Use the following set of commands as an example.

  ```
  # cd /usr/share/xsessions
  # mkdir backup
  # mv *.desktop backup
  # mv backup/kde*.desktop ./
  ```

  After the initial setup, the end user must log out or reboot their Linux desktop to use KDE as the default desktop in their next SSO session.

- For RHEL/CentOS 8.0, for the SSO login session to use Gnome Classic, remove all the desktop startup files, except for the Gnome Classic startup file, from the `/usr/share/xsession` directory. Use the following set of commands as an example.

  ```
  # cd /usr/share/xsessions
  # mkdir backup
  # mv *.desktop backup
  # mv backup/gnome-classic.desktop ./
  ```

  After the initial setup, the end user must log out or reboot their Linux desktop to use Gnome Classic as the default desktop in their next SSO session.

If you disabled SSO on a Linux desktop that has multiple desktop environments installed, you do not need to perform any of the previously described steps. The end users have to select their desired desktop environment when they log in to that Linux desktop.

Network Requirements

VMware Blast Extreme supports both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Network conditions affect the performances of UDP and TCP. To receive the best user experience, select UDP or TCP based on the network condition.

- Select TCP if the network condition is good, such as in a local area network (LAN) environment.
- Select UDP if the network condition is poor, such as in a wide area network (WAN) environment with packet loss and time delay.

Use a network analyzer tool, such as Wireshark, to determine whether VMware Blast Extreme is using TCP or UDP. Use the following set of steps, which use Wireshark, as a reference example.

1 Download and install Wireshark on your Linux VM.

   For RHEL/CentOS 6:

   ```
   sudo yum install wireshark
   ```

   For Ubuntu 18.04/16.04:

   ```
   sudo apt install tshark
   ```

   For SLED/SLES 12:

   ```
   sudo zypper install wireshark
   ```

2 Connect to the Linux desktop using VMware Horizon Client.

3 Open a terminal window and run the following command, which displays the TCP package or UDP package used by VMware Blast Extreme.

   ```
   sudo tshark –i any | grep 22443
   ```

USB Redirection and Client Drive Redirection (CDR) features are sensitive to network conditions. If the network condition is bad, such as a limited bandwidth with time delay and packet loss, the user experience becomes poor. In such condition, the end user might experience one of the following.

- Copying remote files can be slow. In this situation, transmit smaller sized files instead.
- USB device does not appear in the remote Linux desktop.
- USB data does not transfer completely. For example, if you copy a large file, you might get a file smaller in size than the original file.

VHCI Driver for USB Redirection

The USB redirection feature has a dependency on the USB Virtual Host Controller Interface (VHCI) kernel driver. To support USB 3.0 and the USB redirection feature, you must perform the following steps:

1 Download the USB VHCI source code from https://sourceforge.net/projects/usb-vhci/files/linux%20kernel%20module/.

2 To compile the VHCI driver source code and install the resulting binary on your Linux system, use the commands in Table 1-8. Compile and Install USB VHCI Driver.

For example, if you unpack the installation file, VMware-horizonagent-linux-x86_64-*<version>*-*<build-number>*.tar.gz, under the /install_tmp/ directory, the *full-path_to_patch-file* is /install_tmp/VMware-horizonagent-linux-x86_64-*<version>*-*<buildnumber>*/resources/vhci/patch/vhci.patch and the patch command to use is

```
# patch -p1 < /install_tmp/VMware-horizonagent-linux-x86_64-<version>-<build-number>/resources/
vhci/patch/vhci.patch
```

Note The VHCI driver installation must be done before the installation of Horizon for Linux.

Table 1-8. Compile and Install USB VHCI Driver

Linux Distribution	Steps to Compile and Install USB VHCI Driver
Ubuntu 18.04	1 Install the dependency packages. `# apt-get install make` `# apt-get install gcc` `# apt-get install libelf-dev` 2 Compile and install the VHCI drivers. `# tar -xzvf vhci-hcd-1.15.tar.gz` `# cd vhci-hcd-1.15` `# patch -p1 < full-path_to_patch-file` `# make clean && make && make install`
Ubuntu 16.04	Compile and install the VHCI drivers. `# tar -xzvf vhci-hcd-1.15.tar.gz` `# cd vhci-hcd-1.15` `# patch -p1 < full-path_to_patch-file` `# make clean && make && make install`

Table 1-8. Compile and Install USB VHCI Driver (continued)

Linux Distribution	Steps to Compile and Install USB VHCI Driver
RHEL/CentOS 6.9/6.10 RHEL/CentOS 7.x RHEL/CentOS 8.0	1 Install the dependency packages. `# yum install gcc-c++` `# yum install kernel-devel-$(uname -r)` `# yum install kernel-headers-$(uname -r)` `# yum install patch` `# yum install elfutils-libelf-devel` 2 Compile and install the VHCI drivers. `# tar -xzvf vhci-hcd-1.15.tar.gz` `# cd vhci-hcd-1.15` `# patch -p1 < full-path_to_patch-file` `# make clean && make && make install` 3 (RHEL/CentOS 8.0) To ensure that the VHCI drivers work properly with USB redirection, configure signing settings for the USB driver. a Create an SSL key pair for the USB driver. `openssl req -new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out MOK.der -nodes -days 36500 -subj "/CN=Descriptive name/"` b Sign the USB driver. `sudo /usr/src/kernels/$(uname -r)/scripts/sign-file sha256 ./MOK.priv ./MOK.der /lib/modules/$(uname -r)/kernel/drivers/usb/host/usb-vhci-iocifc.ko` `sudo /usr/src/kernels/$(uname -r)/scripts/sign-file sha256 ./MOK.priv ./MOK.der /lib/modules/$(uname -r)/kernel/drivers/usb/host/usb-vhci-hcd.ko` c Register the key for UEFI Secure Boot. `sudo mokutil --import MOK.der` **Note** This command issues a request to set a Machine Owner Key (MOK) password for UEFI Secure Boot. d To set up UEFI Secure Boot in the vSphere console, reboot the system. For more information, see https://sourceware.org/systemtap/wiki/SecureBoot.
SLED/SLES 12 SP2	1 Find out the version of the current kernel package. `# rpm -qa | grep kernel-default-$(echo $(uname -r) | cut -d '-' -f 1,2)` The output is the name of the kernel package currently installed. If, for example, the package name is `kernel-default-3.0.101-63.1`, then the current kernel package version is 3.0.101-63.1. 2 Install the `kernel-devel`, `kernel-default-devel`, `kernel-macros`, and the `patch` packages. `# zypper install --oldpackage kernel-devel-<kernel-package-version> \` `kernel-default-devel-<kernel-package-version> kernel-macros-<kernel-package-version> patch` For example: `# zypper install --oldpackage kernel-devel-4.4.21-90.1 kernel-default-devel-4.4.21-90.1 kernel-macros-4.4.21-90.1 patch`

Table 1-8. Compile and Install USB VHCI Driver (continued)

Linux Distribution	Steps to Compile and Install USB VHCI Driver
	3 Compile and install the VHCI drivers. `# tar –xzvf vhci–hcd–1.15.tar.gz` `# cd vhci–hcd–1.15` `# patch –p1 < full–path_to_patch–file` `# mkdir –p linux/$(echo $(uname –r) \| cut –d '–' –f 1)/drivers/usb/core` `# cp /lib/modules/$(uname –r)/source/include/linux/usb/hcd.h linux/$(echo $(uname –r) | cut –d '–' –f 1)/drivers/usb/core` `# make clean && make && make install`

In addition, observe the following guidelines:

- If your Linux kernel changes to a new version, you must recompile and reinstall the VHCI driver, but you do not need to reinstall Horizon for Linux.
- You can also add Dynamic Kernel Module Support (DKMS) to the VHCI driver using steps similar to the following example for an Ubuntu 18.04/16.04 system.

  a Install the kernel headers.

  ```
  # apt install linux–headers–`uname –r`
  ```

  b Install `dkms` using the following command.

  ```
  # apt install dkms
  ```

  c Extract and patch the VHCI TAR file.

  ```
  # tar xzvf vhci–hcd–1.15.tar.gz
  # cd vhci–hcd–1.15
  # patch –p1 <full–path_to_patch–file>
  # cd ..
  ```

  d Copy the extracted VHCI source files to the `/usr/src` directory.

  ```
  # cp –r vhci–hcd–1.15 /usr/src/usb–vhci–hcd–1.15
  ```

  e Create a file named `dkms.conf` and place it in the `/usr/src/usb–vhci–hcd–1.15` directory.

  ```
  # touch /usr/src/usb–vhci–hcd–1.15/dkms.conf
  ```

  f Add the following contents to the `dkms.conf` file.

  ```
  PACKAGE_NAME="usb–vhci–hcd"
  PACKAGE_VERSION=1.15
  MAKE_CMD_TMPL="make KVERSION=$kernelver"

  CLEAN="$MAKE_CMD_TMPL clean"

  BUILT_MODULE_NAME[0]="usb–vhci–iocifc"
  ```

```
DEST_MODULE_LOCATION[0]="/kernel/drivers/usb/host"
MAKE[0]="$MAKE_CMD_TMPL"

BUILT_MODULE_NAME[1]="usb-vhci-hcd"
DEST_MODULE_LOCATION[1]="/kernel/drivers/usb/host"
MAKE[1]="$MAKE_CMD_TMPL"

AUTOINSTALL="YES"
```

g Add this VHCI driver in `dkms`.

```
# dkms add usb-vhci-hcd/1.15
```

h Build the VHCI driver.

```
# dkms build usb-vhci-hcd/1.15
```

i Install the VHCI driver.

```
# dkms install usb-vhci-hcd/1.15
```

Virtual Machine Settings for 2D Graphics

When you create certain Horizon 7 for Linux virtual machines, you must change the vCPU and virtual memory settings for performance requirements.

Virtual machines that are configured to use NVIDIA vDGA use the NVIDIA physical graphics card. Virtual machines that are configured to use NVIDIA GRID vGPU use the NVIDIA virtual graphics card, which is based on the NVIDIA physical graphics accelerator. You do not need to change the vCPU and virtual memory settings for these virtual machines.

Virtual machines that are configured to use 2D graphics use the VMware virtual graphics card, and you must change vCPU and virtual memory settings to improve the desktop performance. Use the following guidelines:

- For improved performance of a 2D desktop, set more vCPUs and virtual memory for the Linux virtual machine. For example, set 2 vCPUs and 2 GB of virtual memory.
- For the large screen display of multiple monitors, such as four monitors, set 4 vCPUs and 4 GB of virtual memory for the virtual machine.
- For improved video playback in a 2D desktop, set 4 vCPUs and 4 GB of virtual memory for the virtual machine.

Configuring Session Collaboration on Linux Desktops

With the Session Collaboration feature, users can invite other users to join an existing Linux remote desktop session.

System Requirements for Session Collaboration

To support the Session Collaboration feature, your Horizon deployment must meet certain requirements.

Table 1-9. System Requirements for Session Collaboration

Component	Requirements
Client system	Session owners and collaborators must have Horizon Client 4.10 or later for Windows, Mac, or Linux installed on the client system, or must use HTML Access 4.10 or later.
Linux remote desktops	Horizon Agent 7.7 or later must be installed in the Linux virtual desktop. The Session Collaboration feature must be enabled at the desktop pool and the VDI level.
Connection Server	The Connection Server instance uses an Enterprise license.
Display protocol	VMware Blast

Note RHEL 8.0 desktops require additional system configuration to support Session Collaboration. See Configure a RHEL 8.0 Desktop for Session Collaboration.

For information about how to use the Session Collaboration feature, see the Horizon Client documentation.

Setting Session Collaboration Options in Configuration Files

Set the following option in the `/etc/vmware/viewagent-custom.conf` file to enable or disable the Session Collaboration feature.

- `CollaborationEnable`

Set the following options in the `/etc/vmware/config` file to configure the settings used during a collaboration session.

- `collaboration.logLevel`
- `collaboration.maxCollabors`
- `collaboration.enableEmail`
- `collaboration.serverUrl`
- `collaboration.enableControlPassing`

See Setting Options in Configuration Files on a Linux Desktop for more information.

Session Collaboration Feature Limitations

Users cannot use the following remote desktop features in a collaboration session.

- USB redirection
- Audio input redirection
- Client drive redirection
- Smart card redirection
- Clipboard redirection

Users cannot change the remote desktop resolution in a collaborative session.

Users cannot have multiple collaboration sessions on the same client machine.

Note If the Session Collaboration icon in the system tray is unresponsive after a user logs in for the first time to the remote desktop, instruct the user to resize the remote desktop window. The Session Collaboration icon becomes responsive after the desktop window is resized.

Configure a RHEL 8.0 Desktop for Session Collaboration

To use the Session Collaboration feature on a RHEL 8.0 desktop, you must first download and install the GNOME 3.28.26 shell extension.

Procedure

1. Download the required GNOME shell extension to the RHEL 8.0 system from https://extensions.gnome.org/extension/615/appindicator-support/. For the shell version, select **3.28**. For the extension version, select **26**.
2. Untar the downloaded package and rename the directory as `appindicatorsupport@rgcjonas.gmail.com` (the "uuid" value in the `metadata.json` file in the package).
3. Use the `mv` command to move the `appindicatorsupport@rgcjonas.gmail.com` directory to this location: `/usr/share/gnome-shell/extensions`.

   By default, the `metadata.json` file in the `appindicatorsupport@rgcjonas.gmail.com` directory is only readable to the root user. To support Session Collaboration, you must make this file readable to other users as well.
4. Run the command to make `metadata.json` readable to other users, as shown in the following example.

   ```
   chmod a+r metadata.json
   ```
5. Install `gnome-tweaks`.
6. In the desktop environment, restart GNOME shell by pressing the following sequence of keys on the keyboard.

   ```
   Alt+F2
   r
   Enter
   ```
7. In the desktop environment, run `gnome-tweaks` and then enable **KStatusNotifierItem/AppIndicator Support**.

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

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.
- 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 Virtual Desktops in Horizon 7* document.
- Familiarize yourself with the video memory (vRAM) settings requirements for the monitors you plan to use with the virtual machine. See System Requirements For Horizon 7 for Linux.

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 the required settings, follow the guidelines in the installation guide for your Linux distribution.

   For example, Ubuntu 18.04 specifies 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 graphics, which use the VMware driver. The vRAM size has no effect on vDGA or NVIDIA GRID vGPU machines, which use NVIDIA drivers.

   For the required settings, follow the guidelines in Virtual Machine Settings for 2D Graphics. Do not use the Video Memory Calculator.

3 Power on the virtual machine and install the Linux distribution.

4 Configure the desktop environment to use for the specific Linux distribution.

   See the Desktop Environment section in System Requirements For Horizon 7 for Linux for additional information.

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.

To prepare a Linux machine for management by Horizon 7, you must enable communication between the machine and the 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 8.0/7x, CentOS 8.0/7x, and SLED/SLES 12.x machines. If you are preparing any of these machines for use as a remote desktop, you can skip steps 1 through 5 in the following procedure, which describe how to install VMware Tools by manually running the installer.

If you are using an Ubuntu16.04/18.04 machine, install OVT on it. If you are preparing this machine for use as a remote desktop, you can skip steps 1 through 5 in the following procedure and manually install OVT on your Ubuntu 16.04/18.04 machine using the following command:

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

Prerequisites

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

- Familiarize yourself with the steps for mounting and installing VMware Tools on a Linux VM. See "Manually Install or Upgrade VMware Tools in a Linux Virtual Machine" in the *vSphere Virtual Machine Administration* document.
- Familiarize yourself with the steps for configuring your Linux machine to be resolvable through DNS. These steps vary for the different Linux distributions and releases. For instructions, consult the documentation for your Linux distribution and release.

Procedure

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

   The `vmware-tools-distrib` folder is extracted to the desktop.
3. On the VM, log in as root and open a terminal window.
4. Uncompress the VMware Tools tar installer file.

   For example:

   ```
   tar 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. Map the Linux machine's host name to 127.0.0.1 in the `/etc/hosts` file.

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

   Note If you change the Linux machine's host name after installing Horizon Agent, you must map the new host name to 127.0.0.1 in the `/etc/hosts` file. Otherwise, the old host name continues to be used.
7. For RHEL and CentOS, verify that `virbr0` is disabled.

   ```
   virsh net-destroy default
   virsh net-undefine default
   service libvirtd restart
   ```
8. Ensure that the Horizon 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 Horizon Console. Edit the `/etc/hosts` file as follows:

   a `# nano /etc/hosts`

   b Add the fully qualified domain name. For example: `127.0.0.1 hostname.domainname hostname`.

   c Exit and save the file.

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

   a In Yast, click **Network Settings**.

   b Click the **Hostname/DNS** tab.

   c Deselect **Change Hostname via DHCP**.

   d Enter the hostname and the domain name.

   e Click **OK**.

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

Install Dependency Packages for Horizon Agent

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

Prerequisites

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

Procedure

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

Table 2-1. Mandatory Dependency Packages

Linux Distribution	Packages
RHEL 7.5	`yum install libappindicator-gtk3`
SLES 12.x SP1/SLED 12.x SP1 Upgrade `xf86-video-vmware` to a version later than 13.0.2-3.2 from the SUSE repository.	1 Register SUSE 12.x to enable the SUSE repositories. `SUSEConnect -r Registration Code -e Email` 2 Update the `xf86-video-vmware` version. `zypper update xf86-video-vmware`
SLES 12.x	Install python-gobject2 is required for SLES 12.x Linux desktop when you are installing Horizon Agent. 1 Register SUSE 12.x to enable the SUSE repositories. `SUSEConnect -r Registration Code -e Email` 2 Install `python-gobject2`. `zypper install python-gobject2`
Ubuntu 16.04	`apt-get install python-dbus python-gobject`
Ubuntu 18.04	`apt-get install python python-dbus python-gobject`

2 Install the optional package for Horizon Agent.

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

  ```
  sudo yum install glibc
  ```

Setting Up Active Directory Integration for Linux Desktops

3

Horizon 7 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.

Note Horizon Agent expects the Linux desktop and the client user to reside in the same Active Directory domain. If the desktop and user reside in different domains, Horizon Agent might misidentify the desktop domain as being the user domain.

This chapter includes the following topics:

Integrating Linux with Active Directory

Multiple solutions exist to integrate Linux with Microsoft Active Directory (AD) and Horizon 7 for Linux Desktop has no dependency on which solution is used.

The following solutions are known to work in a Horizon 7 for Linux desktop environment.

- OpenLDAP Server Pass-through Authentication
- System Security Services Daemon (SSSD) LDAP Authentication against the Microsoft Active Directory
- Winbind Domain Join
- PowerBroker Identity Services Open (PBISO) Authentication
- Samba Offline Domain Join

If you use the LDAP-based solutions, you must perform the configuration in a template virtual machine and no additional steps are required in the cloned virtual machines.

Note For ease of deployment, use the solution that uses SSSD LDAP authentication against the Microsoft Active Directory.

Use the OpenLDAP Server Pass-Through Authentication

You can set up an OpenLDAP server and use the pass-through authentication (PTA) mechanism to verify the user credentials against Active Directory.

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

Procedure

1. To enable LDAPS (Lightweight Directory Access Protocol over SSL), install Certificate Services on the Active Directory.
2. Set up an OpenLDAP server.
3. Synchronize user information (except password) from the Active Directory to the OpenLDAP server.
4. Configure the OpenLDAP server to delegate password verification to a separate process such as `saslauthd`, which can perform password verification against the Active Directory.
5. Configure the Linux desktops to use an LDAP client to authenticate users with the OpenLDAP server.

Set Up SSSD LDAP Authentication Against the Microsoft Active Directory

You can use LDAP authentication against Windows Active Directory by configuring a System Security Services Daemon (SSSD) in the Linux desktop.

Use the following high-level steps to the SSSD LDAP authentication solution.

Procedure

1. To enable LDAPS (Lightweight Directory Access Protocol Over Secure Socket Layer), install the Certificate Services on the Active Directory server.
2. To use LDAP authentication directly against the Microsoft Active Directory, configure the SSSD in the Linux desktop.

Use the Winbind Domain Join Solution

The Winbind domain join solution, a Kerberos-based authentication solution, is another method of authenticating with Active Directory.

Use the following high-level steps to set up the Winbind domain join solution.

Procedure

1. Install the `winbind`, `samba`, and Kerberos packages on the Linux desktop.

2 Join the Linux desktop to the Microsoft Active Directory.

What to do next

If you use the Winbind Domain Join solution or another Kerberos authentication-based solution, join the template virtual machine to the Active Directory, and rejoin the cloned virtual machine to the Active Directory. For example, use the following command:

```
sudo /usr/bin/net ads join –U <domain_user>%<domain_password>
```

Use the following options to run the domain rejoin command on a cloned virtual machine for the Winbind solution:

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

Configure PowerBroker Identity Services Open (PBISO) Authentication

The PowerBroker Identity Services Open (PBISO) authentication method is one of the supported solutions for performing an offline domain join.

Use the following steps to join a Linux desktop to Active Directory using PBISO.

Procedure

1 Download PBISO 8.5.6 or later from `https://www.beyondtrust.com/products/powerbroker-identity-services-open/`.

2 Install PBISO on your Linux VM.

```
sudo ./pbis-open-8.5.6.2029.linux.x86_64.deb.sh
```

3 Install Horizon 7 Agent for Linux.

4 Use PBISO to join the Linux desktop to the AD domain.

In the following example, **`lxdc.vdi`** is the domain name and **`administrator`** is the domain user name.

```
sudo domainjoin-cli join lxdc.vdi administrator
```

5 Set up the default configuration for domain users.

```
sudo /opt/pbis/bin/config UserDomainPrefix lxdc
sudo /opt/pbis/bin/config AssumeDefaultDomain true
sudo /opt/pbis/bin/config LoginShellTemplate /bin/bash
sudo /opt/pbis/bin/config HomeDirTemplate %H/%U
```

6 Edit the `/etc/pamd.d/common-session` file.

a Locate the line that says **`session sufficient pam_lsass.so`**.

b Replace that line with **`session [success=ok default=ignore] pam_lsass.so`**.

Note This step must be repeated after you reinstall or update the Horizon Agent for Linux.

7 For Ubuntu 16.04, append the following lines to the `/usr/share/lightdm/lightdm.conf.d/50-unity-greeter.conf` configuration file.

```
allow-guest=false
greeter-show-manual-login=true
```

Note If you are using Ubuntu 18.04, no change to the `lightdm` configuration file is required.

8 Reboot your system and log in.

What to do next

Note

- If the `/opt/pbis/bin/config AssumeDefaultDomain` option is set to **`false`**, you must update the `SSOUserFormat=<username>@<domain>` setting in the `/etc/vmware/viewagent-custom.conf` file.
- When using the Horizon instant-clone floating desktop pool feature, to avoid losing the DNS Server setting when the new network adapter is added to the cloned VM, modify the `resolv.conf` file for your Linux system. Use the following example, for an Ubuntu 16.04 system, as a guide for adding the necessary lines in the `/etc/resolvconf/resolv.conf.d/head` file.

```
nameserver 10.10.10.10
search mydomain.org
```

Configure the Samba Offline Domain Join

To support SSO on an instant-cloned VM in a Horizon 7 Linux desktop environment, configure Samba on the master Linux VM.

Use the following procedure as an example for using Samba to offline domain join an instant-cloned Linux desktop to Active Directory. This procedure provides the steps for an Ubuntu system.

Procedure

1 On your master Linux VM, install the `winbind` and `samba` packages, including any other dependent libraries such as `smbfs` and `smbclient`.

2 Install the Samba `tdb-tools` package using the following command.

```
sudo apt-get install tdb-tools
```

3 Install Horizon 7 Agent for Linux.

4 Edit the `/etc/samba/smb.conf` configuration file so that it has content similar to the following example.

```
[global]
security = ads
realm = LAB.EXAMPLE.COM
workgroup = LAB
idmap uid = 10000-20000
idmap gid = 10000-20000
winbind enum users = yes
winbind enum group = yes
template homedir = /home/%D/%U
template shell = /bin/bash
client use spnego = yes
client ntlmv2 auth = yes
encrypt passwords = yes
winbind use default domain = yes
restrict anonymous = 2
```

5 Edit the `/etc/krb5.conf` configuration file so that it has content similar to the following example..

```
[libdefaults]
default_realm = EXAMPLE.COM

krb4_config = /etc/krb.conf
krb4_realms = /etc/krb.realms

kdc_timesync = 1
ccache_type = 4
forwardable = true
proxiable = true

[realms]
YOUR-DOMAIN = {
kdc = 10.111.222.33
}

[domain_realm]
your-domain = EXAMPLE.COM
.your-domain = EXAMPLE.COM
```

6 Edit the `/etc/nsswitch.conf` configuration file, as shown in the following example.

```
passwd: files winbind
group: files winbind
shadow: files winbind
gshadow: files
```

7 Verify that the host name is correct and that the system date and time are synchronized with your DNS system.

8 To inform Horizon Agent that the Linux VM is domain joined using the Samba method, set the following option in the `/etc/vmware/viewagent-custom.conf` file.

```
OfflineJoinDomain=samba
```

9 Reboot your system and log back in.

Use the Realmd Join Solution for RHEL/CentOS 8.0

To ensure the operation of features such as single sign-on for a RHEL/CentOS 8.0 desktop, use the realmd solution to join the desktop to your Active Directory (AD) domain.

Procedure

1 Configure a fully qualified host name for the RHEL/CentOS 8.0 system.

For example, if **`rhel8`** is the unqualified host name of the system and **`LXD.VDI`** is the AD domain, run the following command.

```
# hostnamectl set-hostname rhel8.lxd.vdi
```

2 Verify the network connection with the AD domain, as shown in the following example.

```
# realm discover -vvv LXD.VDI
```

3 Install the required dependency packages, as shown in the following example.

```
 # dnf install -y sssd adcli samba-common-tools oddjob oddjob-mkhomedir
```

4 Join the AD domain, as shown in the following example.

```
# realm join -U Administrator LXD.VDI
```

5 Edit the `/etc/sssd/sssd.conf` so that it resembles the following example. Add `ad_gpo_map_interactive = +gdm-vmwcred` under the *[domain/domain name]* section.

```
[sssd]
domains = LXD.VDI
config_file_version = 2
services = nss, pam

[domain/LXD.VDI]
ad_domain = LXD.VDI
krb5_realm = LXD.VDI
realmd_tags = manages-system joined-with-adcli
cache_credentials = True
id_provider = ad
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
```

```
use_fully_qualified_names = False
fallback_homedir = /home/%u
access_provider = ad
ad_gpo_map_interactive = +gdm-vmwcred
```

6 To ensure that the domain-join takes effect, reboot your system and log back in.

7 Verify that the domain users are configured correctly. The following example shows how to use the `id` command to return the configuration output from domain user **zyc1**.

```
# id zyc1

uid=1084401403(zyc1) gid=1084400513(domain users) groups=1084400513(domain users)
```

8 Using the credentials of a domain user, verify that you can successfully log in to the desktop.

Note Horizon Agent only supports the X11 display server protocol for RHEL/CentOS 8.0 desktops. To configure X11 as the default display server protocol for your system, click the Settings icon on the login screen and select **Classic (X11 display server)** from the drop-down menu.

Setting Up Single Sign-On

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

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

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

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

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

If users must specify the domain name to log in, you must set the `SSOUserFormat` option on the Linux desktop. For more information, see Setting Options in Configuration Files on a Linux Desktop. SSO always uses the short domain name in upper case. For example, if the domain is `mydomain.com`, SSO uses 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 fails. For example, if you have a user, `juser` in AD `mycompany.com`, but the user's UPN is set to `juser123@mycompany.com` instead of `juser@mycompany.com`, SSO fails. The workaround is for the user to log in using the name that is stored in `sAMAccount`. For example, `juser`.

Horizon 7 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
  ```

If your Linux desktop has multiple desktop environments installed on it, refer to Desktop Environment to select the desktop environment to use with SSO.

Setting Up Smart Card Redirection

To set up smart card redirection, you must perform some configuration steps.

Overview of Smart Card Redirection

Smart card redirection is supported on desktops running the following Linux distributions with the specified versions of Horizon Agent installed.

Table 3-1. System Requirements for Smart Card Redirection

Linux Distribution	Horizon Agent
RHEL 8.0	Horizon Agent 7.10 or later
RHEL 7.1 or later	Horizon Agent 7.8 or later
RHEL 6.6 or later	Horizon Agent 6.2.1 or later
Ubuntu 18.04/16.04	Horizon Agent 7.9 or later
SLED/SLES 12.x SP3	Horizon Agent 7.9 or later

When you install Horizon Agent, you must first disable 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.

If the smart card redirection feature is enabled on a virtual machine, vSphere Client's USB redirection does not work with the smart card.

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

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

Note Smart card redirection supports PIV cards on Linux desktops. When you use Horizon Client for Linux to authenticate the broker with a PIV card, you must configure the PIV smart card with TLSv1.2 support to avoid receiving an SSL error. Use the solution described in VMWare Knowledge Base article http://kb.vmware.com/kb/2150470.

Note Smartcard SSO is enabled in Horizon 7 version 7.0.1 or later. RHEL 6.x desktops support Smartcard SSO, but RHEL 7.x and RHEL 8.0 desktops do not support the feature.

Configuring Smart Card Redirection

To configure smart card redirection, perform the following tasks.

1 Set up the smart card for your desktop by following the instructions from the Linux distributor and from the smart card vendor.

2 Integrate your desktop with an Active Directory domain, following the procedure for your Linux distribution.

3 Configure smart card redirection on your desktop, following the procedure for your Linux distribution.

Configuring Smart Card Redirection for RHEL 8.0 Desktops

To set up smart card direction for a RHEL 8.0 desktop, first integrate the desktop with an Active Directory domain. Then install the necessary libraries and root CA certificate before installing Horizon Agent.

Integrate a RHEL 8.0 Desktop with Active Directory for Smart Card Redirection

Use the following procedure to integrate a RHEL 8.0 desktop with an Active Directory (AD) domain for smart card redirection.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
rhel8sc.rzview2.com	Fully qualified host name of your RHEL 8.0 system
rhel8sc	Unqualified host name of your RHEL 8.0 system
rzview2.com	DNS name of your AD domain
RZVIEW2.COM	DNS name of your AD domain, in all capital letters
RZVIEW2	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
rzviewdns.rzview2.com	Host name of your AD server

Procedure

1. On your RHEL 8.0 system, do the following.
   a. Configure network and DNS settings as required by your organization.
   b. Disable **IPv6**.
   c. Disable **Automatic DNS**.
2. Configure the `/etc/hosts` configuration file, so that it resembles the following example.

```
127.0.0.1        rhel8sc.rzview2.com rhel8sc localhost localhost.localdomain localhost4
localhost4.localdomain4
::1              localhost localhost.localdomain localhost6 localhost6.localdomain6

dns_IP_ADDRESS   rzviewdns.rzview2.com
```

3. Configure the `/etc/resolv.conf` configuration file, so that it resembles the following example.

```
# Generated by NetworkManager
search rzview2.com
nameserver dns_IP_ADDRESS
```

4. Install the packages required for the AD integration.

```
# yum install –y samba–common–tools oddjob–mkhomedir
```

5. Enable the `oddjobd` service.

```
# systemctl enable oddjobd.service
# systemctl start oddjobd.service
```

6. Specify the system identity and authentication sources.

```
# authselect select sssd with–smartcard with–mkhomedir
```

7. Start the `oddjobd` service.

```
# systemctl enable oddjobd.service
# systemctl start oddjobd.service
```

8. To support smart card authentication, create the `/etc/sssd/sssd.conf` file.

```
# touch /etc/sssd/sssd.conf
# chmod 600 touch /etc/sssd/sssd.conf
# chown root:root /etc/sssd/sssd.conf
```

9 Add the required content to `/etc/sssd/sssd.conf`, as shown in the following example. Under the **`[pam]`** section, specify **`pam_cert_auth = True`**.

```
[sssd]
config_file_version = 2
domains = rzview2.com
services = nss, pam, pac

[domain/RZVIEW2.COM]
id_provider = ad
auth_provider = ad
chpass_provider = ad
access_provider = ad
cache_credentials = true

[pam]
pam_cert_auth = True
```

10 Enable the `sssd` service.

```
# systemctl enable sssd.service
# systemctl start sssd.service
```

11 Edit the `/etc/krb5.conf` configuration file so that it resembles the following example.

```
# To opt out of the system crypto-policies configuration of krb5, remove the
# symlink at /etc/krb5.conf.d/crypto-policies which will not be recreated.
includedir /etc/krb5.conf.d/

[logging]
    default = FILE:/var/log/krb5libs.log
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log

[libdefaults]
    dns_lookup_realm = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true
    rdns = false
    pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
    spake_preauth_groups = edwards25519
    default_realm = RZVIEW2.COM
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 RZVIEW2.COM = {
     kdc = rzviewdns.rzview2.com
     admin_server = rzviewdns.rzview2.com
     default_domain = rzviewdns.rzview2.com
     pkinit_anchors = FILE:/etc/pki/nssdb/certificate.pem
     pkinit_cert_match = <KU>digitalSignature
     pkinit_kdc_hostname = rzviewdns.rzview2.com
 }
```

```
[domain_realm]
 .rzview2.com = RZVIEW2.COM
 rzview2.com = RZVIEW2.COM
```

12 Edit the `/etc/samba/smb.conf` configuration file so that it resembles the following example.

```
[global]
        workgroup = RZVIEW2
        security = ads
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
        password server = rzviewdns.rzview2.com
        realm = RZVIEW2.COM
        idmap config * : range = 16777216-33554431
        template homedir =/home/RZVIEW2/%U
        template shell = /bin/bash
        kerberos method = secrets and keytab

[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes

[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @printadmin root
        force group = @printadmin
        create mask = 0664
        directory mask = 0775
```

13 Join the AD domain, as shown in the following example.

```
# net ads join -U AdminUser
```

Running the `join` command returns output similar to the following example.

```
Enter AdminUser's password:
Using short domain name -- RZVIEW2
Joined 'RHEL8SC' to dns domain 'rzview2.com'
```

14 Verify that the RHEL 8.0 desktop is successfully joined to the AD domain.

```
# net ads testjoin

Join is OK
```

What to do next

Configure Smart Card Redirection for a RHEL 8.0 Desktop

Configure Smart Card Redirection for a RHEL 8.0 Desktop

To configure smart card redirection on a RHEL 8.0 desktop, install the libraries on which the feature depends, the root CA certificate to support the trusted authentication of smart cards, and the required PC/SC Lite library.

Prerequisites

Integrate a RHEL 8.0 Desktop with Active Directory for Smart Card Redirection

Procedure

1 Install the required libraries.

```
# yum install -y opensc pcsc-lite pcsc-lite-libs pcsc-lite-ccid nss-tools
```

2 Enable the `pcscd` service.

```
# systemctl enable pcscd
# systemctl start pcscd
```

3 Make sure that the `/etc/sssd/sssd.conf` configuration file contains the following lines, which enable smart card authentication.

```
[pam]
pam_cert_auth = True
```

4 Copy the required CA certificate to `/etc/sssd/pki/sssd_auth_ca_db.pem`.

```
# openssl x509 -inform der -in certificate.cer -out certificate.pem
# cp certificate.pem /etc/sssd/pki/sssd_auth_ca_db.pem
```

5 To verify the status of the smart card, run the following `pkcs11-tool` commands and confirm that they return the correct output.

```
# pkcs11-tool -L

# pkcs11-tool --login -O

# pkcs11-tool --test --login
```

6 Set up the PKCS11 module.

```
cp libcmP11.so /usr/lib64/
```

7 Create the `/usr/share/p11-kit/modules/libcmP11.module` file. Add the following content to the file.

```
# This file describes how to load the opensc module
# See: http://p11-glue.freedesktop.org/doc/p11-kit/config.html

# This is a relative path, which means it will be loaded from
# the p11-kit default path which is usually $(libdir)/pkcs11.
# Doing it this way allows for packagers to package opensc for
# 32-bit and 64-bit and make them parallel installable
module: /usr/lib64/libcmP11.so
priority: 99
```

8 Update PC/SC Lite to version 1.8.8.

```
# yum install -y git flex autoconf automake libtool libudev-devel flex
# git clone https://salsa.debian.org/rousseau/PCSC.git
# cd PCSC
# git checkout -b 1.8.8 pcsc-1.8.8
# ./bootstrap
# ./configure --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu
     --program-prefix= --disable-dependency-tracking --prefix=/usr --exec-prefix=/usr
     --bindir=/usr/bin --sbindir=/usr/sbin --sysconfdir=/etc --datadir=/usr/share
     --includedir=/usr/include --libdir=/usr/lib64 --libexecdir=/usr/libexec
     --localstatedir=/var --sharedstatedir=/var/lib --mandir=/usr/share/man
     --infodir=/usr/share/info --disable-static --enable-usbdropdir=/usr/lib64/pcsc/drivers
# make
# make install
```

9 Install Horizon Agent 7.10 or later, with smart card redirection enabled.

10 Reboot your system and log back in.

Configuring Smart Card Redirection for RHEL 7.x/6.x Desktops

To set up smart card direction for a RHEL 7.x/6.x desktop, first integrate the desktop with an Active Directory domain. Then install the necessary libraries and root CA certificate before installing Horizon Agent.

Integrate a RHEL 7.x/6.x Desktop with Active Directory for Smart Card Redirection

To support smart card redirection on a RHEL 7.x/6.x desktop, integrate the desktop with an Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate a RHEL 7.x/6.x desktop with an AD domain for smart card redirection.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server

Note Smart card redirection is supported on desktops running RHEL 6.0 or later, or RHEL 7.1 or later.

Procedure

1 On your RHEL 7.x/6.x desktop, install the required packages.

```
# yum install nscd samba-winbind krb5-workstation pam_krb5 samba-winbind-clients authconfig-gtk
```

2 Edit the network settings for your system connection. Open the NetworkManager control panel and navigate to the **IPv4 Settings** for your system connection. For IPv4 Method, select **Automatic (DHCP)** . In the **DNS** text box, enter the IP address of your DNS name server. Then click **Apply**.

3 Run the following command and verify that it returns the Fully Qualified Domain Name (FQDN) of your RHEL desktop.

```
# hostname –f
```

4 Edit the `/etc/resolve.conf` configuration file, as shown in the following example.

```
search mydomain.com
nameserver dns_IP_ADDRESS
```

5 Disable Security-Enhanced Linux (SELinux) on your RHEL desktop. Edit the `/etc/selinux/config` configuration file, as shown in the following example.

```
SELINUX=disabled
```

6 Edit the `/etc/krb5.conf` configuration file, as shown in the following example.

```
[libdefaults]
      dns_lookup_realm = false
      ticket_lifetime = 24h
      renew_lifetime = 7d
      forwardable = true
      rdns = false
      default_realm = MYDOMAIN.COM
      default_ccache_name = KEYRING:persistent:%{uid}
```

```
[realms]
      MYDOMAIN.COM = {
            kdc = ads-hostname
            admin_server = ads-hostname
            default_domain = ads-hostname
      }

[domain_realm]
      .mydomain.com = MYDOMAIN.COM
      mydomain.com = MYDOMAIN.COM
```

7 Edit the `/etc/samba/smb.conf` configuration file, as shown in the following example.

```
[global]
      workgroup = MYDOMAIN
      password server = ads-hostname
      realm = MYDOMAIN.COM
      security = ads
      idmap config * : range = 16777216-33554431
      template homedir =/home/MYDOMAIN/%U
      template shell = /bin/bash
      kerberos method = secrets and keytab
      winbind use default domain = true
      winbind offline logon = false
      winbind refresh tickets = true

      passdb backend = tdbsam
```

8 Open the `authconfig-gtk` tool and configure settings as follows.

a Select the **Identity & Authentication** tab. For User Account Database, select **Winbind**.

b Select the **Advanced Options** tab, and select the **Create home directories on the first login** check box.

c Select the **Identity & Authentication** tab and then click **Join Domain**. At the alert asking you to save changes, click **Save**.

d When prompted, enter the user name and password of the domain administrator, and click **OK**.

Your RHEL desktop is joined to the AD domain.

9 Set up ticket caching on PAM Winbind. Edit the `/etc/security/pam_winbind.conf` configuration file so that it includes the lines shown in the following example.

```
[global]

# authenticate using kerberos
;krb5_auth = yes

# create homedirectory on the fly
;mkhomedir = yes
```

10 Restart the Winbind service.

```
# sudo service winbind restart
```

11 To verify the AD join, run the following commands and ensure that they return the correct output.

- `net ads testjoin`
- `net ads info`

12 Reboot your system and log back in.

What to do next

Set Up Smart Card Redirection for a RHEL 7.x/6.x Desktop

Set Up Smart Card Redirection for a RHEL 7.x/6.x Desktop

To configure smart card redirection on a RHEL 7.x/6.x desktop, install the libraries on which the feature depends, the root CA certificate required for authentication, and the required PC/SC Lite library. In addition, you must edit some configuration files to complete the authentication setup.

Use the following procedure to set up smart card redirection for a RHEL 7.x/6.x desktop.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server

Smart card redirection is supported on desktops running RHEL 6.0 or later, or RHEL 7.1 or later.

Note If you use the vSphere console to log in to a RHEL 7.x. system that has Horizon Agent installed and smart card redirection enabled, you might experience a delayed logout time of two minutes or longer. This delayed logout only occurs from the vSphere console. The RHEL 7.x logout experience from Horizon Client is not affected.

Prerequisites

Integrate a RHEL 7.x/6.x Desktop with Active Directory for Smart Card Redirection

Procedure

1 Install the required libraries.

```
yum install nss-tools nss-pam-ldapd esc pam_pkcs11 pam_krb5 opensc pcsc-lite-ccid authconfig
      authconfig-gtk krb5-libs krb5-workstation krb5-pkinit pcsc-lite pcsc-lite-libs
```

2 Install a Root Certification Authority (CA) certificate.

a Download a root CA certificate and save it to `/tmp/certificate.cer` on your desktop. See How to Export Root Certification Authority Certificate.

b Locate the root CA certificate that you downloaded, and transfer it to a `.pem` file.

```
openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem
```

c Use the `certutil` command to install the root CA certificate to the system database `/etc/pki/nssdb`.

```
certutil -A -d /etc/pki/nssdb -n "root CA cert" -t "CT,C,C" -i /tmp/certificate.pem
```

d Copy the root CA certificate to the `/etc/pam_pkcs11/cacerts` directory.

```
mkdir -p /etc/pam_pkcs11/cacerts

cp /tmp/certificate.pem /etc/pam_pkcs11/cacerts
```

3 Navigate to **Applications > Sundry > Authentication**, select the **Enable smart card support** check box, and click **Apply**.

4 Copy the smart card drivers and add the drivers library to the system database `/etc/pki/nssdb`.

```
cp libcmP11.so /usr/lib64/
modutil -add "piv card 2.0" -libfile /usr/lib64/libcmP11.so -dbdir /etc/pki/nssdb/
```

5 Edit the `module` setting in the `/etc/pam_pkcs11/pam_pkcs11.conf` configuration file, as shown in the following example.

```
pkcs11_module coolkey {
      module = libcmP11.so;
      description = "Cool Key";
      slot_num = 0;
      ca_dir = /etc/pam_pkcs11/cacerts;
      nss_dir = /etc/pki/nssdb;
      cert_policy = ca, signature;
}
```

6 Edit the `/etc/pam_pkcs11/cn_map` file so that it includes content similar to the following example. For the specific content to include, refer to the user information listed in the smart card certificate.

```
user sc -> user-sc
```

7 Edit the /etc/krb5.conf/ configuration file, as shown in the following example.

```
[libdefaults]
      dns_lookup_realm = false
      ticket_lifetime = 24h
      renew_lifetime = 7d
      forwardable = true
      rdns = false
      default_realm = MYDOMAIN.COM
      default_ccache_name = KEYRING:persistent:%{uid}

[realms]
      MYDOMAIN.COM = {
            kdc = ads-hostname
            admin_server = ads-hostname
            default_domain = ads-hostname
            pkinit_anchors = FILE:/etc/pki/nssdb/certificate.pem
            pkinit_cert_match = <KU>digitalSignature
            pkinit_kdc_hostname = ads-hostname
      }

[domain_realm]
      .mydomain.com = MYDOMAIN.COM
      mydomain.com = MYDOMAIN.COM
```

8 Edit the /etc/pam.d/system-auth configuration file so that it includes the line shown in the following example.

```
auth  optional  pam_krb5.so use_first_pass no_subsequent_prompt
                preauth_options=X509_user_identity=PKCS11:/usr/lib64/libcmP11.so
```

9 Restart the PC/SC daemon.

```
chkconfig pcscd on
service pcscd start
```

10 Install the required PC/SC Lite version for your RHEL distribution.

- For RHEL 7.x, install PC/SC Lite, version 1.8.8.

```
yum install git flex autoconf automake libtool libudev-devel flex
git clone https://salsa.debian.org/rousseau/PCSC.git
cd PCSC
git checkout -b 1.8.8 pcsc-1.8.8
./bootstrap
./configure --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu --program-prefix=
   --disable-dependency-tracking --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --
sbindir=/usr/sbin
   --sysconfdir=/etc --datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64
   --libexecdir=/usr/libexec --localstatedir=/var --sharedstatedir=/var/lib --mandir=/usr/
```

```
share/man
   --infodir=/usr/share/info --disable-static --enable-usbdropdir=/usr/lib64/pcsc/drivers
make
make install
```

- For RHEL 6.x, install PC/SC Lite, version 1.7.4.

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

11 Install the Horizon Agent package, with smart card redirection enabled.

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

Install the required package for your RHEL distribution:

- For RHEL 7.x, install Horizon Agent 7.8 or later.
- For RHEL 6.x, install View Agent 6.2.1 or later.

12 Reboot your system and log back in.

Configuring Smart Card Redirection for Ubuntu Desktops

To set up smart card direction for an Ubuntu desktop, first integrate the desktop with an Active Directory domain. Then install the necessary libraries and root CA certificate before installing Horizon Agent.

Integrate an Ubuntu Desktop with Active Directory for Smart Card Redirection

To support smart card redirection on an Ubuntu desktop, integrate the desktop with an Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate an Ubuntu desktop with an AD domain for smart card redirection.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain

Placeholder Value	Description
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server
ads-hostname.mydomain.com	Fully qualified domain name (FQDN) of your AD server
mytimeserver.mycompany.com	DNS name of your NTP time server
AdminUser	User name of the Linux desktop administrator

Procedure

1. On your Ubuntu desktop, define the host name of the desktop by editing the `/etc/hostname` configuration file.
2. Configure DNS.
   a. Add the DNS server name and IP address to the `/etc/hosts` configuration file.
   b. Add your DNS name server's IP address and the DNS name of your AD domain to the `/etc/network/interfaces` configuration file, as shown in the following example.

```
dns-nameservers dns_IP_ADDRESS
dns-search mydomain.com
```

3. Install the `resolvconfig` package.
   a. Run the installation command.

```
# apt-get install -y resolvconf
```

   Allow the system to install the package and reboot.
   b. Verify your DNS configuration in the `/etc/resolve.conf` file, as shown in the following example.

```
# cat /etc/resolve.conf
…
nameserver dns_IP_ADDRESS
search mydomain.com
```

4. Configure network time synchronization.
   a. Install the `ntpdate` package.

```
# apt-get install -y ntpdate
```

   b. Add the NTP server information to the `/etc/systemd/timesyncd.conf` configuration file, as shown in the following example.

```
[Time]
NTP=mytimeserver.mycompany.com
```

5 Restart the NTP service.

```
sudo service ntpdate restart
```

6 Install the required AD join packages.

a Run the installation command.

```
# apt-get install -y samba krb5-config krb5-user winbind libpam-winbind
    libnss-winbind
```

b At the installation prompt asking for the default Kerberos realm, enter the DNS name of your AD domain in capital letters (for example, `MYDOMAIN.COM`). Then select **Ok**.

7 Edit the `/etc/krb5.conf` configuration file, as shown in the following example.

```
[libdefaults]
      dns_lookup_realm = false
      ticket_lifetime = 24h
      renew_lifetime = 7d
      forwardable = true
      rdns = false
      default_realm = MYDOMAIN.COM
      default_ccache_name = KEYRING:persistent:%{uid}

[realms]
      MYDOMAIN.COM = {
            kdc = ads-hostname.mydomain.com
            admin_server = ads-hostname.mydomain.com
            default_domain = ads-hostname.mydomain.com
            pkinit_anchors = FILE:/etc/pki/nssdb/certificate.pem
            pkinit_cert_match = <KU>digitalSignature
            pkinit_kdc_hostname = ads-hostname.mydomain.com
      }

[domain_realm]
      .mydomain.com = MYDOMAIN.COM
      mydomain.com = MYDOMAIN.COM
```

8 To verify the Kerberos certification, run the following commands.

```
# kinit Administrator@MYDOMAIN.COM

# klist
```

Verify that the commands return output similar to the following example.

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: Administrator@MYDOMAIN.COMValid starting       Expires              Service
principal
2019-05-27T17:12:03   2019-05-28T03:12:03    krbtgt/MYDOMAIN.COM@MYDOMAIN.COM
        renew until 2019-05-28T17:12:03
```

9 Edit the `/etc/samba/smb.conf` configuration file, as shown in the following example.

```
[global]
      workgroup = MYDOMAIN
      realm = MYDOMAIN.COM
      password server = ads-hostname.mydomain.com
      security = ads
      kerberos method = secrets only
      winbind use default domain = true
      winbind offline logon = false
      template homedir =/home/%D/%U
      template shell = /bin/bash
      client use spnego = yes
      client ntlmv2 auth = yes
      encrypt passwords = yes
      passdb backend = tdbsam
      winbind enum users = yes
      winbind enum groups = yes
      idmap uid = 10000-20000
      idmap gid = 10000-20000
```

10 Join the AD domain, and check the integration.

a Run the AD join commands.

```
# net ads join -U AdminUser@mydomain.com
# systemctl stop samba-ad-dc
# systemctl enable smbd nmbd winbind
# systemctl restart smbd nmbd winbind
```

b Modify the `/etc/nsswitch.conf` configuration file, as shown in the following example.

```
passwd:    compat systemd winbind
group:     compat systemd winbind
shadow:    compat
gshadow:   files
```

c To check the results of the AD join, run the following commands and verify that they return the correct output.

```
# wbinfo -u

# wbinfo -g
```

d To check the Winbind Name Service Switch, run the following commands and verify that they return the correct output.

```
# getent group|grep 'domain admins'

# getent passwd|grep 'ads-hostname'
```

11 Enable all PAM profiles.

```
# pam-auth-update
```

In the PAM Configuration screen, select all the PAM profiles, including **Create home directory on login**, and then select **Ok**.

12 On Ubuntu 16.04, enable the user switch in the login screen. Modify the `/usr/share/lightdm/lightdm.conf.d/50-ubuntu.conf` file as shown in the following example.

```
user-session=ubuntu
greeter-show-manual-login=true
```

What to do next

Set Up Smart Card Redirection for an Ubuntu Desktop

Set Up Smart Card Redirection for an Ubuntu Desktop

To configure smart card redirection on an Ubuntu desktop, install the libraries on which the feature depends and the root CA certificate to support the trusted authentication of smart cards. In addition, you must edit some configuration files to complete the authentication setup.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server
ads-hostname.mydomain.com	Fully qualified domain name (FQDN) of your AD server
mytimeserver.mycompany.com	DNS name of your NTP time server
AdminUser	User name of the Linux desktop administrator

Prerequisites

Integrate an Ubuntu Desktop with Active Directory for Smart Card Redirection

Procedure

1 Install the required libraries.

```
# apt-get install -y pcscd pcsc-tools pkg-config libpam-pkcs11 opensc
      libengine-pkcs11-openssl libnss3-tools
```

2 Install a Root Certification Authority (CA) certificate.

a Download a root CA certificate and save it to `/tmp/certificate.cer` on your desktop. See How to Export Root Certification Authority Certificate.

b Locate the root CA certificate that you downloaded, and transfer it to a `.pem` file.

```
# openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem
```

c Use the `certutil` command to install the root CA certificate to the system database `/etc/pki/nssdb`.

```
# certutil -A -d /etc/pki/nssdb -n "root CA cert" -t "CT,C,C" -i /tmp/certificate.pem
```

d Copy the root CA certificate to the `/etc/pam_pkcs11/cacerts` directory.

```
# mkdir -p /etc/pam_pkcs11/cacerts

# cp /tmp/certificate.pem /etc/pam_pkcs11/cacerts
```

3 Create a `pkcs11` hash file.

```
# chmod a+r certificate.pem
# pkcs11_make_hash_link
```

4 Copy the required drivers and add the necessary library files to the `nssdb` directory.

a Run the following commands.

```
# cp libcmP11.so /usr/lib/
# mkdir –p /etc/pki/nssdb
# certutil –N –d /etc/pki/nssdb
# certutil –A –n rootca –i certificate.pem –t "CT,CT,CT" –d /etc/pki/nssdb
# modutil –dbdir /etc/pki/nssdb/  –add "piv card 2.0" –libfile /usr/lib/libcmP11.so
```

b Verify that the expected certificate is loaded successfully.

```
# certutil –L –d /etc/pki/nssdb

Certificate Nickname

rootca
```

c Verify that the expected libraries are added successfully.

```
modutil –dbdir /etc/pki/nssdb –list

Listing of PKCS #11 Modules
-----------------------------------------------------------
  1. NSS Internal PKCS #11 Module
         slots: 2 slots attached
        status: loaded

         slot: NSS Internal Cryptographic Services
        token: NSS Generic Crypto Services

         slot: NSS User Private Key and Certificate Services
        token: NSS Certificate DB

  2. piv card 2.0
        library name: /usr/lib/libcmP11.so
         slots: There are no slots attached to this module
        status: loaded
-----------------------------------------------------------
```

5 Configure the pam_pkcs11 library.

a Create a pam_pkcs11.conf file using default example content.

```
# mkdir /etc/pam_pkcs11
# zcat /usr/share/doc/libpam-pkcs11/examples/pam_pkcs11.conf.example.gz |
       tee /etc/pam_pkcs11/pam_pkcs11.conf
```

b Edit the /etc/pam_pkcs11/pam_pkcs11.conf file as shown in the following example.

```
use_pkcs11_module = mysc;

pkcs11_module mysc {
              module = /usr/lib/libcmP11.so;
              description = "LIBCMP11";
              slot_num = 0;
              ca_dir = /etc/pki/cacerts;
              nss_dir = /etc/pki/nssdb;
              cert_policy = ca;
}
...
use_mappers = cn, null;
...
mapper cn {
      debug = false;
      module = internal;
      # module = /lib/pam_pkcs11/cn_mapper.so;
      ignorecase = true;
      mapfile = file:///etc/pam_pkcs11/cn_map;
      # mapfile = "none";
}
```

c Edit the /etc/pam_pkcs11/cn_map file so that it includes the following line.

```
ads-hostname -> ads-hostname
```

6 Configure the PAM authentication.

a Edit the `/etc/pam.d/gdm-password` configuration file. Place the `pam_pkcs11.so` authorization line before the `common-auth` line, as shown in the following example.

```
#%PAM-1.0
auth    requisite       pam_nologin.so
auth    required        pam_succeed_if.so user != root quiet_success
auth sufficient
pam_pkcs11.so
@include common-auth
auth    optional        pam_gnome_keyring.so
@include common-account
```

b For Ubuntu 16.04, edit the `/etc/pam.d/lightdm` configuration file. Place the `pam_pkcs11.so` authorization line before the `common-auth` line, as shown in the following example.

```
#%PAM-1.0
auth    requisite       pam_nologin.so debug
auth    sufficient      pam_succeed_if.so user ingroup nopasswdlogin debug
auth    [success=3 default=ignore}     pam_pkcs11.so
@include common-auth
auth    optional        pam_gnome_keyring.so
auth    optional        pam_kwallet.so
```

c For Ubuntu 16.04, edit the `/etc/pam.d/unity` configuration file. Place the `pam_pkcs11.so` authorization line before the `common-auth` line, as shown in the following example.

```
auth    [success=3 default=ignore}     pam_pkcs11.so
@include common-auth
auth optional pam_gnome_keyring.so
```

7 To verify the smart card hardware and the certificates installed on the smart card, run the following commands.

```
# pcsc_scan

# pkcs11_listcerts

# pkcs11_inspect
```

8 Configure the Gnome screensaver so that it locks when the smart card is removed.

a Install the screensaver package.

```
# apt-get install gnome-screensaver
```

b To configure the screensaver, edit the `/etc/pam_pkcs11/pkcs11_eventmgr.conf` file, as shown in the following example.

```
pkcs11_eventmgr {
  # Run in background? Implies debug=false if true
  daemon = true;

  # show debug messages?
  debug = false;

  # polling time in seconds
  polling_time = 1;

  # expire time in seconds
  # default = 0 ( no expire )
  expire_time = 0;

  # pkcs11 module to use
  pkcs11_module = /usr/lib/libcmP11.so;

  #
  # list of events and actions
  # Card inserted
  event card_insert {
    # what to do if an action fail?
    # ignore : continue to next action
    # return : end action sequence
    # quit : end program
    on_error = ignore ;

    # You can enter several, comma-separated action entries
    # they will be executed in turn
    action = "gnome-screensaver-command --poke";
  }

  # Card has been removed
  event card_remove {
  on_error = ignore;
    action = "gnome-screensaver-command --lock";
  }

  # Too much time card removed
    event expire_time {
```

```
    on_error = ignore;
    action = "/bin/false";
  }
}
```

c Run `pkcs11_eventmgr`.

```
# /usr/bin/pkcs11_eventmgr &
```

9 Install the Horizon Agent package, with smart card redirection enabled.

```
# sudo ./install_viewagent.sh –m yes
```

Note You must install Horizon Agent 7.9 or later.

10 Reboot your system and log back in.

Configuring Smart Card Redirection for SLED/SLES Desktops

To set up smart card direction for a SLED/SLES desktop, first integrate the desktop with an Active Directory domain. Then install the necessary libraries and root CA certificate before installing Horizon Agent.

Integrate a SLED/SLES Desktop with Active Directory for Smart Card Redirection

To support smart card redirection on a SLED/SLES desktop, integrate the desktop with an Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate a SLED/SLES desktop with an AD domain for smart card redirection.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server
ads-hostname.mydomain.com	Fully qualified domain name (FQDN) of your AD server
mytimeserver.mycompany.com	DNS name of your NTP time server
AdminUser	User name of the Linux desktop administrator

Procedure

1 Configure the network settings for your SLED/SLES desktop.

   a Define the host name of the desktop by editing the `/etc/hostname` and `/etc/hosts` configuration files.

   b Configure the DNS server IP address, and disable **Automatic DNS**. For SLES 12 SP3, also disable **Change Hostname via DHCP**.

   c To configure network time synchronization, add your NTP server information to the `/etc/ntp.conf` file, as shown in the following example.

```
server mytimeserver.mycompany.com
```

2 Install the required AD join packages.

```
# zypper in krb5-client samba-winbind
```

3 Edit the required configuration files.

a Edit the `/etc/samba/smb.conf` file, as shown in the following example.

```
[global]
        workgroup = MYDOMAIN
        usershare allow guests = NO
        idmap gid = 10000-20000
        idmap uid = 10000-20000
        kerberos method = secrets and keytab
        realm = MYDOMAIN.COM
        security = ADS
        template homedir = /home/%D/%U
        template shell = /bin/bash
        winbind use default domain=true
        winbind offline logon = yes
        winbind refresh tickets = yes
[homes]
        ...
```

b Edit the `/etc/krb5.conf` file, as shown in the following example.

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        clockskew = 300

[realms]
        MYDOMAIN.COM = {
                kdc = ads-hostname.mydomain.com
                default_domain = mydomain.com
                admin_server = ads-hostname.mydomain.com
        }

[logging]
        kdc = FILE:/var/log/krb5/krb5kdc.log
        admin_server = FILE:/var/log/krb5/kadmind.log
        default = SYSLOG:NOTICE:DAEMON

[domain_realm]
        .mydomain.com = MYDOMAIN.COM
        mydomain.com = MYDOMAIN.COM

[appdefaults]
        pam = {
                ticket_lifetime = 1d
                renew_lifetime = 1d
                forwardable = true
                proxiable = false
                minimum_uid = 1
        }
```

c Edit the `/etc/security/pam_winbind.conf` file, as shown in the following example.

```
cached_login = yes
krb5_auth = yes
krb5_ccache_type = FILE
```

d Edit the `/etc/nsswitch.conf` file, as shown in the following example.

```
passwd: compat winbind
group: compat winbind
```

4 Join the AD domain, as shown in the following example.

```
# net ads join –U AdminUser
```

5 Enable the Winbind service.

a To enable and start Winbind, run the following sequence of commands.

```
# pam–config ––add ––winbind
# pam–config –a ––mkhomedir
# systemctl enable winbind
# systemctl start winbind
```

b To ensure that AD users can log in to the desktop without having to restart the Linux server, run the following sequence of commands.

```
# systemctl stop nscd
# nscd –i passwd
# nscd –i group
# systemctl start nscd
```

6 To confirm the success of the AD join, run the following commands and check that they return the correct output.

```
# wbinfo –u

# wbinfo –g
```

What to do next

Set Up Smart Card Redirection for a SLED/SLES Desktop

Set Up Smart Card Redirection for a SLED/SLES Desktop

To configure smart card redirection on a SLED/SLES desktop, install the libraries on which the feature depends and the root CA certificate to support the trusted authentication of smart cards. In addition, you must edit some configuration files to complete the authentication setup.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
ads-hostname	Host name of your AD server
ads-hostname.mydomain.com	Fully qualified domain name (FQDN) of your AD server
mytimeserver.mycompany.com	DNS name of your NTP time server
AdminUser	User name of the Linux desktop administrator

Prerequisites

Integrate a SLED/SLES Desktop with Active Directory for Smart Card Redirection

Procedure

1 Install the required library packages.

a Install the PAM library and other packages.

```
# zypper install pam_pkcs11 mozilla-nss mozilla-nss-tools
    pcsc-lite pcsc-ccid opensc coolkey pcsc-tools
```

b To install the PC/SC tools, run the following series of commands.

```
# SUSEConnect --list-extensions
# SUSEConnect -p PackageHub/12.3/x86_64
# zypper in pcsc-tools
```

2 Install a Root Certification Authority (CA) certificate.

a Download a root CA certificate and save it to `/tmp/certificate.cer` on your desktop. See How to Export Root Certification Authority Certificate.

b Locate the root CA certificate that you downloaded, transfer it to a `.pem` file, and create a hash file.

```
# openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem
# cp /tmp/certificate.pem /etc/pam_pkcs11/cacerts
# chmod a+r /etc/pam_pkcs11/cacerts/certificate.pem
# cd /etc/pam_pkcs11/cacerts
# pkcs11_make_hash_link
```

c Install trust anchors to the NSS database.

```
# mkdir /etc/pam_pkcs11/nssdb
# certutil –N –d /etc/pam_pkcs11/nssdb
# certutil –L –d /etc/pam_pkcs11/nssdb
# certutil –A –n rootca –i certificate.pem –t "CT,CT,CT" –d /etc/pam_pkcs11/nssdb
```

d Install the required drivers.

```
# cp libcmP11.so /usr/lib64/
# modutil –add "piv card 2.0" –libfile /usr/lib64/libcmP11.so –dbdir /etc/pam_pkcs11/nssdb/
```

3 Edit the `/etc/pam_pkcs11/pam_pkcs11.conf` file.

a Delete the line `use_pkcs11_module = nss`. In its place, add the line `use_pkcs11_module = mysc`.

b Add the `mysc` module, as shown in the following example.

```
pkcs11_module mysc {
  module = /usr/lib64/libcmP11.so;
  description = "MY Smartcard";
  slot_num = 0;
  nss_dir = /etc/pam_pkcs11/nssdb;
  cert_policy = ca, ocsp_on, signature, crl_auto;
}
```

c Update the Common Name mapper configuration, as shown in the following example.

```
# Assume common name (CN) to be the login
mapper cn {
      debug = false;
      module = internal;
      # module = /usr/lib64/pam_pkcs11/cn_mapper.so;
      ignorecase = true;
      mapfile = file:///etc/pam_pkcs11/cn_map;}
```

d Delete the line `use_mappers = ms`. In its place, add the line `use_mappers = cn, null`.

4 Edit the `/etc/pam_pkcs11/cn_map` configuration file so that it includes the following line.

```
ads-hostname -> ads-hostname
```

5 Modify the PAM configuration.

a To make it possible to configure smart card authentication, first disable the `pam_config` tool.

```
# find /etc/pam.d/ -type l -iname "common-*" -delete
# for X in /etc/pam.d/common-*-pc; do cp -ivp $X ${X:0:-3}; done
```

b Create a file named `common-auth-smartcard` under the `/etc/pam.d/` directory. Add the following content to the file.

```
auth    required        pam_env.so
auth    sufficient      pam_pkcs11.so
auth    optional        pam_gnome_keyring.so
auth    [success=1 default=ignore]      pam_unix.so nullok_secure try_first_pass
auth    required        pam_winbind.so  use_first_pass
```

c For SLED/SLES 12 SP3, replace the line `auth include common-auth` with the line `auth include common-auth-smartcard` in both of these files: `/etc/pam.d/gdm` and `/etc/pam.d/xscreensaver`.

6 Disable the firewall.

```
# rcSuSEfirewall2 stop
# chkconfig SuSEfirewall2_setup off
# chkconfig SuSEfirewall2_init off
```

Note Smart card redirection sometimes fails when the firewall is enabled.

7 Install the library packages required for smart card redirection.

a For SLED/SLES 12 SP3, run the following installation commands.

```
# SUSEConnect -p sle-sdk/12.3/x86_64
# zypper in git autoconf automake libtool flex libudev-devel gcc
```

b For SLES 12 SP3, install `systemd-devel`.

```
# zypper in systemd-devel
```

8 Install the Horizon Agent package, with smart card redirection enabled.

```
# sudo ./install_viewagent.sh -m yes
```

Note You must install Horizon Agent 7.9 or later.

9 Reboot your system and log back in.

Setting Up True SSO for Linux Desktops

The True Single Sign-on (True SSO) feature grants users access to a Linux virtual desktop or a published desktop or application after they first log in to VMware Identity Manager. Users can log in to VMware

Identity Manager using a smart card or RSA SecurID or RADIUS authentication, and then access remote Linux resources without entering their Active Directory credentials.

If a user authenticates by using Active Directory (AD) credentials, the True SSO feature is not necessary. However, you can configure True SSO to be used even in this case, so that the desktop can support both AD credentials and True SSO.

When connecting to a Linux virtual desktop or a published desktop or application, users can select to use either the native Horizon Client or HTML Access.

True SSO has the following limitations:

- The feature is supported only on desktops with the following distributions: RHEL/CentOS 8.0, RHEL/CentOS 7.x, Ubuntu 16.04 and 18.04, and SLED/SLES 12.x SP3.
- For RHEL/CentOS 7.x desktops, the feature is supported only with the following join methods: the default join domain tools, Samba, System Security Services Daemon (SSSD), and the Kerberos network authentication protocol.

To set up True SSO on your Linux environment, perform the following tasks.

1 Set up and configure True SSO in your Horizon 7 environment. See "Setting up True SSO" in the *Horizon 7 Administration* document.

2 Integrate your desktop with an AD domain, following the procedure for your Linux distribution.

3 Configure True SSO on your desktop, following the procedure for your Linux distribution.

Configure True SSO on RHEL/CentOS 8.0 Desktops

To support True SSO on a RHEL/CentOS 8.0 desktop, you must first integrate the system with your Active Directory (AD) domain. Then you must modify certain configurations on the system to support the True SSO feature.

Note True SSO is not supported on instant-clone RHEL 8.0 desktops.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
MYDOMAIN	Name of your NetBIOS domain

Prerequisites

- Verify that the Active Directory (AD) server is resolvable by DNS on the RHEL/CentOS 8.0 system.
- Configure the host name of the system.
- Configure the Network Time Protocol (NTP) on the system.

Procedure

1 On the RHEL/CentOS 8.0 system, verify the network connection to Active Directory.

```
# realm discover mydomain.com
```

2 Install the required dependency packages.

```
# yum install oddjob oddjob-mkhomedir sssd adcli samba-common-tools
```

3 Join the AD domain.

```
# realm join --verbose mydomain.com -U administrator
```

4 Download the root CA certificate and copy it to the required directory as a .pem file.

```
# openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem

# cp /tmp/certificate.pem /etc/sssd/pki/sssd_auth_ca_db.pem
```

5 Modify the `/etc/sssd/sssd.conf` configuration file, as shown in the following example.

```
[sssd]
domains = mydomain.com
config_file_version = 2
services = nss, pam

[domain/mydomain.com]
ad_domain = mydomain.com
krb5_realm = IMYDOMAIN.COM
realmd_tags = manages-system joined-with-adcli
cache_credentials = True
id_provider = ad
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
use_fully_qualified_names = False        <---------------- Use short name for user
fallback_homedir = /home/%u@%d
access_provider = ad
ad_gpo_map_interactive = +gdm-vmwcred    <---------------- Add this line for SSO

[pam]                                    <---------------- Add pam section for certificate logon
pam_cert_auth = True                     <---------------- Add this line to enable certificate
logon for system
pam_p11_allowed_services = +gdm-vmwcred  <---------------- Add this line to enable certificate
logon for VMware Horizon Agent

[certmap/mydomain.com/truesso]          <---------------- Add this section and following lines to
set match and map rule for certificate user
matchrule = <EKU>msScLogin
maprule = (|(userPrincipal={subject_principal})(samAccountName={subject_principal.short_name}))
domains = mydomain.com
priority = 10
```

6 Install the Horizon Agent package, with True SSO enabled.

Note You must install Horizon Agent 7.11 or later.

```
# sudo ./install_viewagent.sh –T yes
```

7 Modify the `/etc/vmware/viewagent–custom.conf` configuration file so that it includes the following line.

```
NetbiosDomain = MYDOMAIN
```

8 Reboot the system and log back in.

Configuring True SSO for RHEL/CentOS 7.x Desktops

To set up True SSO for a RHEL/CentOS 7.x desktop, first integrate the desktop with an Active Directory domain. Then install the required libraries and root CA certificate before installing Horizon Agent.

Integrate a RHEL/CentOS 7.x Desktop with Active Directory for True SSO

To support True SSO on an instant-cloned VM in a Horizon 7 Linux desktop environment on a RHEL/CentOS 7.x system, you must configure Samba on the master Linux VM.

The RHEL/CentOS 7.x `realmd` feature provides a simple way to discover and join identity domains. Instead of connecting the system to the domain itself, `realmd` configures underlying Linux system services, such as SSSD or Winbind, to connect to the domain. The following steps describe how to use `realmd` and Samba to perform an offline domain join of a RHEL/CentOS 7.x desktop to Active Directory.

Prerequisites

- The RedHat Enterprise Linux (RHEL) system is subscribed to Red Hat Network (RHN) or has the `yum` tool installed locally.
- The Active Directory (AD) server is resolvable by DNS on the Linux system.
- The Network Time Protocol (NTP) is configured on the Linux system.

Procedure

1 Verify that the RHEL/CentOS system can discover the AD server. Use the following example, where *ADdomain.example.com* must be replaced with your AD server information.

```
sudo realm discover ADdomain.example.com
```

2 Install the Samba `tdb–tools` package.

The Samba `tdb–tools` package is not available for download from the official Red Hat repository. You must download it manually. For example, use the following command to download it from a CentOS 7.5 system and install the downloaded package in your RHEL system.

```
yumdownloader tdb–tools
```

If you do not have a CentOS system, go to `https://rpmfind.net/linux/rpm2html/search.php?query=tdb-tools&submit=Search+...&system=&arch`, download the `tdb-tools-1.3.15-1.el7.x86_64.rpm` package, and install it on your RHEL system.

3 Install Samba and the dependency packages.

```
sudo yum install sssd-tools sssd adcli samba-common pam_ldap pam_krb5 samba samba-client krb5-workstation
```

4 Run the `join` command, using the following example, where *DNSdomain.example.com* must be replaced with the DNS domain path specific for your environment.

```
sudo realm join DNSdomain.example.com -U administrator
```

When the join command succeeds, you receive the following message.

```
Successfully enrolled machine in realm
```

5 Reboot your system and log back in.

What to do next

Configure True SSO on RHEL/CentOS 7.x Desktops

Configure True SSO on RHEL/CentOS 7.x Desktops

To enable the True SSO feature on a RHEL/CentOS 7.x desktop, install the libraries on which the True SSO feature depends, the root CA certificate to support trusted authentication, and Horizon Agent. In addition, you must edit some configuration files to complete the authentication setup.

Use the following procedure to enable True SSO on RHEL 7.x and CentOS 7.x desktops. To support True SSO on these desktops, you must install Horizon Agent 7.6 or later.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the DNS name of your AD domain. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_server	Path to your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters

Prerequisites

- Configure True SSO for VMware Identity Manager and Horizon Connection Server.
- Integrate a RHEL/CentOS 7.x Desktop with Active Directory for True SSO
- Obtain a root Certificate Authority certificate and save it to `/tmp/certificate.cer` on your RHEL/CentOS 7.x desktop. See How to Export Root Certification Authority Certificate.

Procedure

1 Install the PKCS11 support package group.

```
yum install –y nss–tools nss–pam–ldapd pam_krb5 krb5–libs krb5–workstation krb5–pkinit
```

2 Install a Root Certification Authority (CA) certificate.

a Locate the root CA certificate you downloaded, and transfer it to a `.pem` file.

```
openssl x509 –inform der –in /tmp/certificate.cer –out /tmp/certificate.pem
```

b Use the `certutil` command to install the root CA certificate to the system database `/etc/pki/nssdb`.

```
certutil –A –d /etc/pki/nssdb –n "root CA cert" –t "CT,C,C" –i /tmp/certificate.pem
```

c Add the root CA certificate to the list of CA certificates trusted on your RHEL/CentOS 7.x system and update the system-wide trust store configuration using the `update–ca–trust` command.

```
cp /tmp/certificate.pem /etc/pki/ca–trust/source/anchors/ca_cert.pem
update–ca–trust
```

3 Modify the appropriate section in your system's SSSD configuration file for your domain, as shown in the following example.

```
[domain/mydomain.com]
ad_domain = mydomain.com
krb5_realm = MYDOMAIN.COM
realmd_tags = manages–system joined–with–samba
cache_credentials = True
id_provider = ad
krb5_store_password_if_offline = True
default_shell = /bin/bash
ldap_id_mapping = True
#set the next line to false, so you can use the short name instead of the full domain name.
use_fully_qualified_names = False
fallback_homedir = /home/%u@%d
access_provider = ad
```

4 Modify the Kerberos configuration file `/etc/krb5.conf`, as shown in the following example.

```
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 default_ccache_name = KEYRING:persistent:%{uid}
 # Add following line, if the system doesn't add it automatically
 default_realm = MYDOMAIN.COM

[realms]
```

```
MYDOMAIN.COM = {
  kdc = dns_server
  admin_server = dns_server
  # Add the following three lines for pkinit_*
  pkinit_anchors = DIR:/etc/pki/ca-trust/source/anchors
  pkinit_kdc_hostname = your_org_DNS_server
  pkinit_eku_checking = kpServerAuth
 }
[domain_realm]
 mydomain.com = MYDOMAIN.COM
 .mydomain.com = MYDOMAIN.COM
```

5 Install the Horizon Agent package, with True SSO enabled.

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

Note You must install Horizon Agent 7.6 or later.

6 Add the following parameter to the Horizon Agent custom configuration file `/etc/vmware/viewagent-custom.conf`. Use the following example, where *NETBIOS_NAME_OF_DOMAIN* is the NetBIOS name of your organization's domain.

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

7 Reboot your system and log back in.

Configuring True SSO for Ubuntu Desktops

To set up True SSO for an Ubuntu desktop, first integrate the desktop with an Active Directory domain. Then install the required libraries and root CA certificate before installing Horizon Agent.

Integrate an Ubuntu Desktop with Active Directory for True SSO

To support True SSO on an Ubuntu 16.04 or 18.04 desktop, integrate the desktop with an Active Directory domain using the Samba and Winbind solutions.

Use the following procedure to integrate an Ubuntu 16.04 or 18.04 desktop with an AD domain.

Some examples in the procedure use placeholder values to represent entities in your network configuration, such as the host name of your Ubuntu desktop. Replace the placeholder values with information specific to your configuration, as described in the following table.

Placeholder Value	Description
dns_IP_ADDRESS	IP address of your DNS name server
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain, in all capital letters
myhost	Host name of your Ubuntu desktop
MYDOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters

Placeholder Value	Description
ads-hostname	Host name of your AD server
admin-user	User name of the AD domain administrator

Prerequisites

- The Active Directory (AD) server is resolvable by DNS on the Linux system.
- The Network Time Protocol (NTP) is configured on the Linux system.

Procedure

1 On your Ubuntu 16.04 or 18.04 desktop, install the `samba` and `winbind` packages.

```
sudo apt install samba krb5-config krb5-user winbind libpam-winbind libnss-winbind
```

2 When prompted, configure the Kerberos Authentication settings as follows.

a For **Default Kerberos version 5 realm**, enter the DNS name of your AD domain using all capital letters.

For example, if your AD domain name is **`mydomain.com`**, enter **`MYDOMAIN.COM`**.

b For **Kerberos servers for your realm**, enter the host name of your AD server (represented as **`ads_hostname`** in the examples throughout this procedure).

c For **Administrative server for your Kerberos realm**, enter the host name of your AD server again.

3 Update the PAM configuration.

a Open the PAM configuration page.

```
pam-auth-update
```

b Select **Create home directory on login**, and then select **Ok**.

4 Edit the `/etc/nsswitch.conf` configuration file, as shown in the following example.

```
passwd: compat winbind
group: compat winbind
shadow: compat
gshadow: files
```

5 To ensure that the auto-generated `resolv.conf` file refers to your AD domain as a search domain, edit the NetworkManager settings for your system connection.

a Open the NetworkManager control panel and navigate to the **IPv4 Settings** for your system connection. For Method, select **Automatic (DHCP) addresses only**. In the **DNS servers** text box, enter the IP address of your DNS name server (represented as **`dns_IP_ADDRESS`** in the examples throughout this procedure). Then click **Save**.

b Edit the configuration file for your system connection located in `/etc/NetworkManager/system-connections`. Use the following example.

```
[ipv4]
dns=dns_IP_ADDRESS
dns-search=mydomain.com
ignore-auto-dns=true
method=auto
```

Note A new virtual network adapter is added when a new instant-cloned virtual desktop is created. Any setting in the network adapter, such as the DNS server, in the virtual desktop template is lost when the new network adapter is added to the instant-cloned virtual desktop. To avoid losing the DNS server setting when the new network adapter is added to a cloned virtual desktop, you must specify a DNS server for your Linux system.

c Specify the DNS server by editing the `/etc/resolv.conf` configuration file, as shown in the following example.

```
nameserver dns_IP_ADDRESS

search mydomain.com
```

d Reboot your system and log back in.

6 Edit the `/etc/hosts` configuration file, as shown in the following example.

```
127.0.0.1     localhost
127.0.1.1     myhost.mydomain.com myhost
```

7 Edit the `/etc/samba/smb.conf` configuration file, as shown in the following example.

```
[global]
security = ads
realm = MYDOMAIN.COM
workgroup = MYDOMAIN
idmap uid = 10000-20000
idmap gid = 10000-20000
winbind enum users = yes
winbind enum group = yes
template homedir = /home/%D/%U
template shell = /bin/bash
client use spnego = yes
client ntlmv2 auth = yes
encrypt passwords = yes
```

```
winbind use default domain = yes
restrict anonymous = 2
kerberos method = secrets and keytab
winbind refresh tickets = true
```

8 Restart the `smbd` service.

```
sudo systemctl restart smbd.service
```

9 Edit the `/etc/krb5.conf` configuration file so that it has content similar to the following example.

```
[libdefaults]
      default_realm = MYDOMAIN.COM
      dns_lookup_realm = true
      dns_lookup_kdc = true

[realms]
      MYDOMAIN.COM = {
            kdc = ads-hostname
            admin_server = ads-hostname
      }

[domain_realm]
      .mydomain.com = MYDOMAIN.COM
      mydomain.com = MYDOMAIN.COM
```

10 Join your Ubuntu desktop to the AD domain.

a Initiate a Kerberos ticket.

```
sudo kinit admin-user
```

When prompted, enter your administrator password.

b Verify that the ticket has been created successfully.

```
sudo klist
```

This command returns information about the ticket, including its valid starting time and expiration time.

c Create a Kerberos keytab file.

```
sudo net ads keytab create -U admin-user
```

d Join the AD domain.

```
sudo net ads join -U admin-user
```

11 Restart and verify the Winbind service.

a Restart the Winbind service.

```
sudo systemctl restart winbind.service
```

b To verify the Winbind service, run the following commands and check that they return the correct output.

- `wbinfo –u`
- `wbinfo –g`
- `getend passwd`
- `getend group`

12 Reboot your system and log back in.

What to do next

Configure True SSO on Ubuntu Desktops

Configure True SSO on Ubuntu Desktops

To enable the True SSO feature on an Ubuntu 16.04 or 18.04 desktop, install the libraries on which the True SSO feature depends, the root CA certificate to support trusted authentication, and Horizon Agent. In addition, you must edit some configuration files to complete the authentication setup.

Use the following procedure to enable True SSO on Ubuntu 16.04 and 18.04 desktops. To support True SSO on these desktops, you must install Horizon Agent 7.8 or later.

Prerequisites

- Configure True SSO for VMware Identity Manager and Horizon Connection Server.
- Integrate an Ubuntu Desktop with Active Directory for True SSO
- Obtain a root Certificate Authority certificate and save it to `/tmp/certificate.cer` on your desktop. See How to Export Root Certification Authority Certificate.

Procedure

1 On your Ubuntu 16.04 or 18.04 desktop, install the `pkcs11` support package.

```
sudo apt install libpam-pkcs11
```

2 Install the `libnss3-tools` package.

```
sudo apt install libnss3-tools
```

3 Install a Root Certification Authority (CA) certificate.

a Locate the root CA certificate that you downloaded, and transfer it to a `.pem` file.

```
openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem
```

b Use the `certutil` command to install the root CA certificate to the system database `/etc/pki/nssdb`.

```
certutil -A -d /etc/pki/nssdb -n "root CA cert" -t "CT,C,C" -i /tmp/certificate.pem
```

c Copy the root CA certificate to the `/etc/pam_pkcs11/cacerts` directory.

```
mkdir -p /etc/pam_pkcs11/cacerts

cp /tmp/certificate.pem /etc/pam_pkcs11/cacerts
```

d Create a hash link for the root CA certificate. In the `/etc/pam_pkcs11/cacerts` directory, run the following command.

```
pkcs11_make_hash_link
```

4 Install the Horizon Agent package, with True SSO enabled.

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

Note To use the True SSO feature, you must install Horizon Agent 7.8 or later.

5 Add the following parameter to the Horizon Agent custom configuration file `/etc/vmware/viewagent-custom.conf`. Use the following example, where *NETBIOS_NAME_OF_DOMAIN* is the NetBIOS name of your organization's domain.

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

6 Edit the `/etc/pam_pkcs11/pam_pkcs11.conf` configuration file.

a If needed, create the `/etc/pam_pkcs11/pam_pkcs11.conf` configuration file. Locate the example file in `/usr/share/doc/libpam-pkcs11/examples`, copy it to the `/etc/pam_pkcs11` directory, and rename the file to `pam_pkcs11.conf`. Add your system information to the contents of the file as needed.

b Modify the `/etc/pam_pkcs11/pam_pkcs11.conf` configuration file so that it includes content similar to the following example.

```
use_pkcs11_module = coolkey;
pkcs11_module coolkey {
  module = /usr/lib/vmware/viewagent/sso/libvmwpkcs11.so;
  slot_num = 0;
  ca_dir = /etc/pam_pkcs11/cacerts;
  nss_dir = /etc/pki/nssdb;
}
```

7 Modify the `auth` parameters in the PAM configuration file.

   a Open the PAM configuration file.

      - For Ubuntu 16.04, open `/etc/pam.d/lightdm`.
      - For Ubuntu 18.04, open `/etc/pam.d/gdm-vmwcred`.

   b Edit the PAM configuration file, as shown in the following example.

```
auth requisite pam_vmw_cred.so
auth sufficient pam_pkcs11.so try_first_pass
```

8 Reboot your system and log back in.

Configuring True SSO for SLED/SLES Desktops

To set up True SSO for a SLED/SLES desktop, first integrate the desktop with an Active Directory domain. Then install the required libraries and root CA certificate before installing Horizon Agent.

Integrate a SLED/SLES Desktop with Active Directory for True SSO

To support True SSO on a SLED 12.x SP3 or SLES 12.x SP3 desktop, integrate the desktop with an Active Directory domain using the Samba and Winbind solutions.

Use the following procedure to integrate a SLED/SLES desktop with an AD domain.

Prerequisites

- The Active Directory (AD) server is resolvable by DNS on the Linux system.
- The Network Time Protocol (NTP) is configured on the Linux system.

Procedure

1 On your SLED/SLES desktop, install the `samba` and `winbind` packages.

```
zypper install samba-winbind krb5-client samba-winbind-32bit
```

2 Open the YaST setup tool and navigate to **Network Services > Windows Domain Membership**.

3 On the Windows Domain Membership screen, configure settings as follows.

   a For **Domain or Workgroup**, enter the DNS name of the workgroup or NT domain that includes your Samba server, using all capital letters. For example, if your workgroup name is **`mydomain`**, enter **`MYDOMAIN`**.

   b Select **Also Use SMB Information for Linux Authentication**.

   c Select **Create Home Directory on Login**.

   d Select **Offline Authentication**.

   e Select **Single Sign-on for SSH**.

4 At the prompt asking if you want to join the domain, select **Yes**.

5 Enter the administrator name and password for the specified workgroup, and select **OK**.

A message appears confirming that your SLED/SLES desktop joined the domain successfully. Select **OK**.

6 Edit the `/etc/samba/smb.conf` configuration file so that it includes the following parameter.

```
[global]
...
winbind use default domain = yes
```

7 Reboot your system and log back in.

8 Test and verify your SLED/SLES desktop integration.

Run the following test commands and check that they return the correct output. Replace `mydomain` with the name of your Samba server workgroup or NT domain.

- `net ads testjoin`
- `net ads info`
- `wbinfo –-krb5auth=mydomain\\open%open`
- `ssh localhost –l mydomain\\open`

What to do next

Configure True SSO on SLED/SLES Desktops

Configure True SSO on SLED/SLES Desktops

To enable the True SSO feature on a SLED/SLES 12.x SP3 desktop, install the libraries on which the True SSO feature depends, the root CA certificate to support trusted authentication, and Horizon Agent. In addition, you must edit some configuration files to complete the authentication setup.

Use the following procedure to enable True SSO on SLED 12.x SP3 and SLES 12.x SP3 desktops. To support True SSO on these desktops, you must install Horizon Agent 7.8 or later.

Prerequisites

- Configure True SSO for VMware Identity Manager and Horizon Connection Server.
- Integrate a SLED/SLES Desktop with Active Directory for True SSO
- Obtain a root Certificate Authority certificate and save it to `/tmp/certificate.cer` on your SLED/SLES 12.x SP3 desktop. See How to Export Root Certification Authority Certificate.

Procedure

1 For a SLES 12.x SP3 desktop, install the necessary packages by running the following command.

```
zypper install mozilla-nss-tools pam_krb5 krb5-client krb5-plugin-preauth-pkinit
```

2 For a SLED 12.x SP3 desktop, install the necessary packages by performing the following steps.

   a Download a SLES .iso file to the local disk of your SLED desktop (for example, `/tmp/SLE-12-SP3-Server-DVD-x86_64-GM-DVD1.iso`).

   You must add the SLES .iso file as a package source for your SLED desktop because the necessary `krb5-plugin-preauth-pkinit` package is available only for SLES systems.

   b Mount the SLES .iso file on your SLED desktop, and install the necessary packages.

```
sudo mkdir -p /mnt/sles
sudo mount -t iso9660 /tmp/SLE-12-SP3-Server-DVD-x86_64-GM-DVD1.iso /mnt/sles
sudo zypper ar -f /mnt/sles sles
zypper install mozilla-nss-tools pam_krb5 krb5-client krb5-plugin-preauth-pkinit
```

   c When the installation is complete, unmount the SLES .iso file.

```
sudo unmount /mnt/sles
```

3 Install a Root Certification Authority (CA) certificate.

   a Locate the root CA certificate that you downloaded, and transfer it to a `.pem` file.

```
openssl x509 -inform der -in /tmp/certificate.cer -out /tmp/certificate.pem
```

   b Use the `certutil` command to install the root CA certificate to the system database `/etc/pki/nssdb`.

```
certutil -A -d /etc/pki/nssdb -n "root CA cert" -t "CT,C,C" -i /tmp/certificate.pem
```

   c Add the root CA certificate to `pam_pkcs11`.

```
cp /tmp/certificate.pem /etc/pki/ca-trust/source/anchors/ca_cert.pem
```

4 Edit the `/etc/krb5.conf` configuration file so that it has content similar to the following example.

```
[libdefaults]
      default_realm = MYDOMAIN.COM
      dns_lookup_realm = false
      ticket_lifetime = 24h
      renew_lifetime = 7d
      forwardable = true
      rdns = false
      default_ccache_name = KEYRING:persistent:%{uid}

[realms]
      MYDOMAIN.COM = {
            kdc = ads-hostname
            admin_server = ads-hostname
            pkinit_anchors = DIR:/etc/pki/ca-trust/source/anchors
            pkinit_kdc_hostname = ads-hostname
            pkinit_eku_checking = kpServerAuth
      }
```

```
[domain_realm]
      .mydomain.com = MYDOMAIN.COM
      mydomain.com = MYDOMAIN.COM
```

Replace the placeholder values in the example with information specific to your network configuration, as described in the following table.

Placeholder Value	Description
mydomain.com	DNS name of your AD domain
MYDOMAIN.COM	DNS name of your AD domain (in all capital letters)
ads-hostname	Host name of your AD server (case-sensitive)

5 Install the Horizon Agent package, with True SSO enabled.

```
sudo ./install_viewagent.sh –T yes
```

Note To use the True SSO feature, you must install Horizon Agent 7.8 or later.

6 Add the following parameter to the Horizon Agent custom configuration file `/etc/vmware/viewagent-custom.conf`. Use the following example, where *NETBIOS_NAME_OF_DOMAIN* is the NetBIOS name of your organization's domain.

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

7 Reboot your system and log back in.

Setting Up Graphics for Linux Desktops

4

You can configure the currently supported Linux distributions to take advantage of NVIDIA capabilities on the 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, 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 Supported Linux Distributions for vGPU

You can set up a supported Linux distribution to take advantage of NVIDIA vGPU (shared GPU hardware acceleration) capabilities on the ESXi host.

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

Note For information about the NVIDIA graphics cards and Linux distributions that support vGPU, see https://docs.nvidia.com/grid/9.0/product-support-matrix/index.html.

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

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

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

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

Prerequisites

- Verify that vSphere 6.0 U1 or a later release is installed in your environment.
- Verify that the required vGPU graphics card is installed on the ESXi host.

Note For information about the NVIDIA graphics cards and Linux distributions that support vGPU, see https://docs.nvidia.com/grid/9.0/product-support-matrix/index.html.

Procedure

1 Download the VIB for your NVIDIA GRID vGPU graphics card from the NVIDIA Driver Downloads site.

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

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

2 Uncompress the vGPU software package `.zip` file.

3 Upload the vGPU Manager folder to the ESXi host.

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

4 Power off or suspend all virtual machines on the ESXi host.

5 Connect to the ESXi host using SSH.

6 Stop the `xorg` service.

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

7 Install the NVIDIA VIB.

For example:

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

8 Reboot or update the ESXi host.

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

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

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

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

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

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

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

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

Prerequisites

- Verify that the Linux virtual machine is prepared for use as a desktop. See Create a Virtual Machine and Install Linux and Prepare a Linux Machine for Remote Desktop Deployment.
- Verify that Horizon Agent is not installed on the Linux virtual machine.
- Verify that the NVIDIA VIB is installed on the ESXi host. See Install the VIB for the NVIDIA GRID vGPU Graphics Card on the ESXi Host.
- 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.

Note For information about the NVIDIA graphics cards and Linux distributions that support vGPU, see https://docs.nvidia.com/grid/9.0/product-support-matrix/index.html.

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.

Install the NVIDIA GRID vGPU Display Driver

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

Prerequisites

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

  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 .

Procedure

1 Copy the NVIDIA Linux Display Driver to the virtual machine.

2 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.

3 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)
```

4 Add an executable flag to the NVIDIA GRID vGPU driver package.

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

5 Start the NVIDIA GRID vGPU installer.

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

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

What to do next

Install Horizon Agent on the Linux virtual machine. See Install Horizon Agent on a Linux Virtual Machine.

Create a desktop pool that contains the configured Linux virtual machines. See Create a Manual Desktop Pool for Linux.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a Linux virtual machine by displaying the NVIDIA driver output in a Horizon desktop session.

Prerequisites

- Verify 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.
- Verify that the Linux virtual machine is deployed in a desktop pool. See Create a Manual Desktop Pool for Linux.

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 communicate with Horizon Connection Server. To resolve the problem, reinstall the NVIDIA driver.
- Set the NVIDIA GRID licensing in the Linux VM. See the NVIDIA documentation for more information. If licensing is not set, the Linux desktop does not work correctly. For example, auto-fit does not work.

Configure RHEL 6.x for vDGA

You can set up a RHEL 6.x 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.x Virtual Machine

To configure a RHEL 6.x 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 and Prepare a Linux Machine for Remote Desktop Deployment.

- 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.

Procedure

1. Log in to the RHEL 6.x 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 for vDGA

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

Prerequisites

- Verify that the PCI device was added to the RHEL 6.x virtual machine. See Add a vDGA Pass-Through Device to a RHEL 6.x Virtual Machine.

Procedure

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

      For RHEL 6.x, the file is `/boot/grub/grub.conf`.

RHEL Version	Command
6.x	`sudo vi /boot/grub/grub.conf`

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

c Edit the `blacklist.conf` file.

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

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

```
blacklist nouveau
```

2 Restart the virtual machine.

The display has a changed look and feel.

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

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

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

4 Download the NVIDIA driver from the NVIDIA Driver Downloads site.

Select the appropriate driver version from the NVIDIA drop-down menus:

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

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

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

```
sudo yum install gcc-c++
sudo yum install kernel-devel-$(uname -r)
sudo yum install kernel-headers-$(uname -r)
```

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

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

8 Run the NVIDIA installer.

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

9 Accept the NVIDIA software license agreement and select **Yes** to 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.

Create a desktop pool that contains the configured Linux virtual machines. See Create a Manual Desktop Pool for Linux.

Verify That the NVIDIA Display Driver Is Installed

You can verify that the NVIDIA display driver is installed on a Linux virtual machine by displaying the NVIDIA driver output in a Horizon desktop session.

Prerequisites

- Verify 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.
- Verify that the Linux virtual machine is deployed in a desktop pool. See Create a Manual Desktop Pool for Linux.

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 communicate with Horizon Connection Server. To resolve the problem, reinstall the NVIDIA driver.
- Set the NVIDIA GRID licensing in the Linux VM. See the NVIDIA documentation for more information. If licensing is not set, the Linux desktop does not work correctly. For example, auto-fit does not work.

Installing Horizon Agent

5

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

This chapter includes the following topics:

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.

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

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

Caution If you intend to use NVIDIA GRIDvGPU or vDGA, 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 Supported Linux Distributions for vGPU or Configure RHEL 6.x for vDGA. 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.

Prerequisites

- Verify that the Linux guest operating system is prepared for desktop use. See Prepare a Linux Machine for Remote Desktop Deployment.
- Familiarize yourself with the Horizon Agent installer script for Linux. See install_viewagent.sh Command-Line Options.

Procedure

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

   In the Desktop & End-User Computing section, select View Download Components for VMware Horizon. Under Horizon 7 for Linux, select the Downloads page for VMware Horizon 7 for 64-bit Linux systems.

   The installer filename is `VMware-horizonagent-linux-x86_64-`*`y.y.y-xxxxxxx`*`.tar.gz` for 64-bit Linux where *y.y.y* is the version number and *xxxxxxx* is the build number.
2. Unpack the tarball for your Linux distribution on the guest operating system.

   For example:

   ```
   tar -xzvf VMware-horizonagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz
   ```
3. Navigate to the tar ball folder.
4. Run the `install_viewagent.sh` script as superuser.

   See install_viewagent.sh Command-Line Options for a list of the command-line options.

   For example:

   ```
   sudo ./install_viewagent.sh
   ```
5. Type **Yes** to accept the EULA if you run `install_viewagent.sh` without specifying the `-A` option.

   The installer does not run unless you accept the EULA.
6. Reboot Linux for the changes to take effect.

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

What to do next

Deploy the virtual machine in a desktop pool. See Create a Manual Desktop Pool for Linux.

install_viewagent.sh Command-Line Options

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

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

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

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

Table 5-1. `install_viewagent.sh` Optional but Required Parameter

Optional Parameter (Required Information)	Description
-A yes |no	Accept or refuse the End User License Agreement (EULA) and Federal Information Processing Standards (FIPS) statement. You must specify **yes** for the install to proceed.

Table 5-2. `install_viewagent.sh` Optional Parameters

Optional Parameters	Description
-a yes|no	Install or bypass audio input redirection support. Default is yes.
-f yes|no	Install or bypass support of the cryptographic modules designed for Federal Information Processing Standards (FIPS) 140-2. Default is **no**. For more information, see the FIPS 140-2 Mode description in Features of Horizon Linux Desktops.
-j	JMS SSL keystore password. By default, installer generates a random string.
-m yes|no	Install or bypass the smart card redirection support. Default is **no**.
-r yes|no	Restart the system automatically after installation. Default is **no**.
-s	Self signed cert subject DN. By default, installer uses Blast.
-C yes|no	Install or bypass Clipboard Redirection support. Default is **yes**.
-F yes|no	Install or bypass CDR support. Default is **yes**.
-M yes|no	Upgrade the Linux Agent to managed or unmanaged agent. Default is **yes**.
-S yes|no	Install or bypass Single Sign-on (SSO) support. Default is **yes**.
-T yes|no	Install or bypass True Single Sign-on (True SSO) support. Default is **no**.
-U yes|no	Install or bypass USB support. Default is **no**.

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

Condition	Examples
Fresh Installation	`sudo ./install_viewagent.sh –A yes` Fresh installation always requires a new desktop pool creation.
Upgrade from an unmanaged virtual machine and retain the unmanaged virtual machine style	`sudo ./install_viewagent.sh –A yes –M no` This type of upgrade does not require a new desktop pool creation. You can reuse the existing desktop pool. **Note** To ensure the best possible performance, do not use an unmanaged virtual machine.
Upgrade from an unmanaged virtual machine deployment and convert to a managed virtual machine style. The upgrade requires new desktop pool creation on broker	`sudo ./install_viewagent.sh –A yes` This type of upgrade requires a new desktop pool creation. You must delete the existing desktop pool.

Configure the Certificate for Linux Agent

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

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

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

Procedure

1 Install the private key and the certificate to VMwareBlastServer.

   a Rename the private key to 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.

Upgrading the Horizon Agent on a Linux Virtual Machine

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

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

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

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

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

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

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

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

The following parameters are available for upgrade.

Table 5-4. Optional Parameters for Upgrading the Horizon Agent

Parameter	Description
-A yes	EULA and FIPS statement acceptance. You must specify **yes** for the install to proceed. If this parameter is not specified, the install script prompts for the value.
-a yes|no	Install or bypass audio input redirection support.
-f yes|no	Install or bypass support of the cryptographic modules designed for Federal Information Processing Standards (FIPS) 140-2. Default is **no**. For more information, see the FIPS 140-2 Mode description in Features of Horizon Linux Desktops.
-m yes|no	Install or bypass the smart card redirection support. Default is **no**.
-r yes|no	Reboot the operating system after installation. The default is **no**.
-C yes|no	Install or bypass Clipboard Redirection support. Default is **yes**.
-F yes|no	Install or bypass CDR support. Default is **yes**.

Table 5-4. Optional Parameters for Upgrading the Horizon Agent (continued)

Parameter	Description
-M yes|no	Upgrade the Linux Agent to managed|unmanaged agent. The default value is **yes**.
-S yes|no	Install or bypass SingleSignOn (SSO) support. Default is **yes**.
-U yes|no	Install or Bypass USB support. Default is **no**.

Upgrade Horizon Agent on a Linux Virtual Machine

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

Prerequisites

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

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

Procedure

1. Download the latest installer file for Horizon Agent for Linux from the VMware download site at https://my.vmware.com/web/vmware/downloads.

   Under Desktop & End-User Computing, select to download VMware Horizon 7, which includes the installer for Horizon Agent for Linux.

   The installer filename is `VMware-viewagent-linux-x86_64-`*`y.y.y`*`-`*`xxxxxxx`*`.tar.gz` for 64-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 <Horizon Agent tar ball>
   ```

3. Navigate to the tar ball folder.

4 To upgrade unmanaged virtual machines, run the `install_viewagent.sh` script using one of the following deployment scenarios.

Option	Description
Upgrade an unmanaged virtual machine deployment and retain the unmanaged virtual machine deployment	`sudo ./install_viewagent.sh –A yes –M no` **Note** To ensure the best possible performance, do not use an unmanaged virtual machine.
Upgrade an unmanaged virtual machine deployment and change it to managed virtual machine deployment	`sudo ./install_viewagent.sh –A yes –M yes` **Note** In Horizon Console, delete the existing desktop pool for unmanaged virtual machine deployment and create a desktop pool for a managed virtual machine deployment. For more info, see Create a Manual Desktop Pool for Linux.
Upgrade a managed virtual machine deployment	`sudo ./install_viewagent.sh –A yes –M yes` **Note** After upgrading, your existing desktop pool can be reused.

Uninstall Horizon 7 for Linux Machines

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

Prerequisites

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

Procedure

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

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

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

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

Configuration Options for Linux Desktops

6

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

This chapter includes the following topics:

Setting Options in Configuration Files on a Linux Desktop

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

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

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

```
RemoteDisplay.buildToPNG=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`.

Note The following table includes description for each agent-enforced policy setting for USB in the Horizon Agent configuration file. Horizon Agent uses the settings to decide if a USB can be forwarded to the host machine. Horizon Agent also passes the settings to Horizon Client for interpretation and enforcement. The enforcement is based on whether you specify the merge **(m)** modifier to apply the Horizon Agent filter policy setting in addition to the Horizon Client filter policy setting, or override the **(o)** modifier to use the Horizon Agent filter policy setting instead of the Horizon Client filter policy setting.

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

Option	Value/Format	Default	Description
Clipboard.Direction	`0`, `1`, `2`, or `3`	`2`	Use this option to specify the clipboard redirection policy. Valid values are as follows: ■ 0 - Disable clipboard redirection. ■ 1 - Enable clipboard redirection in both directions. ■ 2 - Enable clipboard redirection from the client to the remote desktop only. ■ 3 - Enable clipboard redirection from the remote desktop to the client only.
RemoteDisplay.allowAudio	`true` or `false`	`true`	Set this option to enable/disable audio out.
RemoteDisplay.allowH264	`true` or `false`	`true`	Set this option to enable or disable H.264 encoding.
RemoteDisplay.buildToPNG	`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 the build to 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. Enabling this option disables the H.264 encoding.
RemoteDisplay.enableNetworkContinuity	`true` or `false`	`true`	Set this option to enable or disable the Network Continuity feature in the Horizon Agent for Linux.
RemoteDisplay.enableNetworkIntelligence	`true` or `false`	`true`	Set this option to enable or disable the Network Intelligence feature in Horizon Agent for Linux.
RemoteDisplay.enableStats	`true` or `false`	`false`	Enables or disables the VMware Blast display protocol statistics in mks log, such as bandwidth, FPS, RTT, and so on.
RemoteDisplay.enableUDP	`true` or `false`	`true`	Set this option to enable or disable UDP protocol support in Horizon Agent for Linux.
RemoteDisplay.maxBandwidthKbps	An integer	1000000	Specifies the maximum bandwidth in kilobits per second (kbps) for a VMware Blast session. The bandwidth includes all imaging, audio, virtual channel, and VMware Blast control traffic. Valid value must be less than 4 Gbps (4096000).

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

Option	Value/Format	Default	Description
RemoteDisplay.minBandwidthKbps	An integer	256	Specifies the minimum bandwidth in kilobits per second (kbps) for a VMware Blast session. The bandwidth includes all imaging, audio, virtual channel, and VMware Blast control traffic.
RemoteDisplay.maxFPS	An integer	30	Specifies the maximum rate of screen updates. Use this setting to manage the average bandwidth that users consume. Valid value must be between 3 and 60. The default is 30 updates per second.
RemoteDisplay.maxQualityJPEG	available range of values: 1–100	90	Specifies the image quality of the desktop display for JPEG/PNG encoding. The high-quality settings are for areas of the screen that are more static, resulting in a better image quality.
RemoteDisplay.midQualityJPEG	available range of values: 1–100	35	Specifies the image quality of the desktop display for JPEG/PNG encoding. Use to set the medium-quality settings of the desktop display.
RemoteDisplay.minQualityJPEG	available range of values: 1–100	25	Specifies the image quality of the desktop display for JPEG/PNG encoding. The low-quality settings are for areas of the screen that change often, for example, when scrolling occurs.
RemoteDisplay.qpmaxH264	available range of values: 0–51	36	Use this option to set the H264minQP quantization parameter, which specifies the best image quality for the remote display configured to use H.264 encoding. Set the value to greater than the value set for RemoteDisplay.qpminH264.
RemoteDisplay.qpminH264	available range of values: 0–51	10	Use this option to set the H264maxQP quantization parameter, which specifies the lowest image quality for the remote display configured to use H.264 encoding. Set the value to less than the value set for RemoteDisplay.qpmaxH264.
UsbRedirPlugin.log.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`	Use this option to set the log level for the USB Redirection plugin.
UsbRedirServer.log.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`	Use this option to set the log level for the USB Redirection server.
VMWPkcs11Plugin.log.enable	`true` or `false`	`false`	Set this option to enable or disable the logging mode for the True SSO feature.
VMWPkcs11Plugin.log.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`	Use this option to set the log level for the True SSO feature.
VVC.RTAV.Enable	`true` or `false`	`true`	Set this option to enable/disable audio input.
VVC.ScRedir.Enable	`true` or `false`	`true`	Set this option to enable/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.

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

Option	Value/Format	Default	Description
cdrserver.cacheEnable	`true` or `false`	`true`	Set this option to enable or disable the write caching feature from the agent towards the client side.
cdrserver.customizedSharedFolderPath	`folder_path`	`/home/`	Use this option to change the Client Drive Redirection (CDR) shared folder location from the default `/home/`*`user`*`/tsclient` directory to a custom directory. For example, if the user `test` wants to place the CDR shared folder at `/mnt/test/tsclient` instead of `/home/test/tsclient`, the user can specify **`cdrserver.customizedSharedFolderPath=/mnt/`**. **Note** In order for this option to take effect, the specified folder must exist and be configured with the correct user permissions.
cdrserver.forcedByAdmin	`true` or `false`	`false`	Set this option to control whether the client can share additional folders that are not specified with the `cdrserver.shareFolders` option.
cdrserver.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`	Use this option to set the log level for the `vmware-CDRserver.log` file.
cdrserver.permissions	`R`	`RW`	Use this option to apply additional read/write permissions that Horizon Agent has on the folders shared by Horizon Client. For example: ■ If the folder shared by Horizon Client has `read` and `write` permissions and you set **`cdrserver.permissions=R`**, then Horizon Agent has only `read` access permissions. ■ If the folder shared by Horizon Client has only `read` permissions and you set **`cdrserver.permissions=RW`**, Horizon Agent still has only `read` access rights. Horizon Agent cannot change the `read` only attribute set by Horizon Client. Horizon Agent can only remove the write access rights. Typical uses are as follows: ■ **`cdrserver.permissions=R`** ■ **`#cdrserver.permissions=R`** (for example, comment it out or delete the entry)
cdrserver.sharedFolders	***`file_path1`*`,R;`*`file-path2`*`,;`*`file_path3`*`,R; ...`	undefined	Specify one or more file paths to the folders that the client can share with the Linux desktop. For example: ■ For a Windows client: **`C:\spreadsheets,;D:\ebooks,R`** ■ For a non-Windows client: **`/tmp/spreadsheets;/tmp/ebooks,;/home/finance,R`**

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

Option	Value/Format	Default	Description
collaboration.logLevel	`error`, `info`, or `debug`	`info`	Use this option to set the log level used for the collaboration session. If the log level is `debug`, all calls made to `collabui` functions and the contents of the `collabor` list are logged.
collaboration.maxCollabors	An integer less than 10	5	Specifies the maximum number of collaborators that you can invite to join a session.
collaboration.enableEmail	`true` or `false`	`true`	Set this option to enable or disable sending of collaboration invitations by using an installed email application. When this option is disabled, you cannot use email to invite collaborators, even if an email application is installed.
collaboration.serverUrl	[URL]	undefined	Specifies the server URLs to include in the collaboration invitations.
collaboration.enableControlPassing	`true` or `false`	`true`	Set this option to permit or restrict collaborators from having control of the Linux desktop. To specify a read-only collaboration session, set this option to **`false`**.
mksVNCServer.useUInputButtonMapping	`true` or `false`	`false`	Set this option to enable the support of a left-handed mouse on Ubuntu or RHEL 7.x. CentOS and RHEL 6.x support a left-handed mouse and you do not need to set this option.
mksvhan.clipboardSize	An integer	1024	Use this option to specify the clipboard maximum size to copy and paste.
vdpservice.log.logLevel	`fatal error`, `warn`, `info`, `debug`, or `trace`	`info`	Use this option to set the log level of the `vdpservice`.
viewusb.AllowAudioIn	**`{m|o}:{true|false}`**	undefined, which equates to `true`	Use this option to allow or disallow audio input devices to be redirected. Example: **`o:false`**
viewusb.AllowAudioOut	**`{m|o}:{true|false}`**	undefined, which equates to `false`	Set this option to allow or disallow redirection of audio output devices.
viewusb.AllowAutoDeviceSplitting	**`{m|o}:{true|false}`**	undefined, which equates to `false`	Set this option to allow or disallow the automatic splitting of composite USB devices. Example: **`m:true`**
viewusb.AllowDevDescFailsafe	**`{m|o}:{true|false}`**	undefined, which equates to `false`	Set this option to allow or disallow devices to be redirected even if Horizon Client fails to get the configuration or device descriptors. To allow a device even if it fails to get the configuration or device descriptors, include it in the Include filters, such as **`IncludeVidPid`** or **`IncludePath`**.

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

Option	Value/Format	Default	Description
viewusb.AllowHIDBootable	**{m|o}: {true|false}**	undefined, which equates to `true`	Use this option to allow or disallow the redirection of input devices other than keyboards or mice that are available at boot time, also known as HID-bootable devices.
viewusb.AllowKeyboardMouse	**{m|o}: {true|false}**	undefined, which equates to `false`	Use this option to allow or disallow the redirection of keyboards with integrated pointing devices (such as a mouse, trackball, or touch pad).
viewusb.AllowSmartcard	**{m|o}: {true|false}**	undefined, which equates to `false`	Set this option to allow or disallow smart card devices to be redirected.
viewusb.AllowVideo	**{m|o}: {true|false}**	undefined, which equates to `true`	Use this option to allow or disallow video devices to be redirected.
viewusb.DisableRemoteConfig	**{m|o}: {true|false}**	undefined, which equates to `false`	Set this option to disable or enable the use of Horizon Agent settings when performing USB device filtering.
viewusb.ExcludeAllDevices	**{true|false}**	undefined, which equates to `false`	Use this option to exclude or include all USB devices from being redirected. If set to **true**, you can use other policy settings to allow specific devices or families of devices to be redirected. If set to **false**, you can use other policy settings to prevent specific devices or families of devices from being redirected. If you set the value of **ExcludeAllDevices** to **true** on Horizon Agent, and this setting is passed to Horizon Client, the Horizon Agent setting overrides the Horizon Client setting.
viewusb.ExcludeFamily	**{m|o}:*family_name_1*[;family_name_2;...]**	undefined	Use this option to exclude families of devices from being redirected. For example: **m:bluetooth;smart-card** If you have enabled automatic device splitting, Horizon examines the device family of each interface of a composite USB device to decide which interfaces must be excluded. If you have disabled automatic device splitting, Horizon examines the device family of the whole composite USB device. **Note** Mice and keyboards are excluded from redirection by default and do not need to be excluded with this setting.

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

Option	Value/Format	Default	Description
viewusb.ExcludePath	**{m|o}:bus-*x1*[/*y1*].../ port-*z1*[;bus-*x2*[/*y2*].../port-*z2*;...]**	undefined	Use this option to exclude devices at specified hub or port paths from being redirected. You must specify bus and port numbers in hexadecimal. You cannot use the wildcard character in paths. For example:**m:bus-1/2/3_port-02;bus-1/1/1/4_port-ff**
viewusb.ExcludeVidPid	**{m|o}:vid-*xxx1*_pid-*yyy1*[;vid-*xxx2*_pid-*yyy2*;..]**	undefined	Set this option to exclude devices with specified vendor and product IDs from being redirected. You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID. For example: **o:vid-0781_pid- ****;vid-0561_pid-554c**
viewusb.IncludeFamily	**{m|o}:*family_name_1*[;*family_name_2*]...**	undefined	Set this option to include families of devices that can be redirected. For example: **o:storage; smart-card**
viewusb.IncludePath	**{m|o}:bus-*x1*[/*y1*].../ port-*z1*[;bus-*x2*[/*y2*].../portz2;...]**	undefined	Use this option to include devices at specified hub or port paths that can be redirected. You must specify bus and port numbers in hexadecimal. You cannot use the wildcard character in paths. For example: **m:bus-1/2_port- 02;bus-1/7/1/4_port-0**f
viewusb.IncludeVidPid	**{m|o}:vid-*xxx1*_pid-*yyy1*[;vid-*xxx2*_pid-*yyy2;...*]**	undefined	Set this option to include devices with specified Vendor and Product IDs that can be redirected. You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID. For example: **o:vid-***_pid-0001;vid-0561_pid-554c**

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

Option	Value/Format	Default	Description
viewusb.SplitExcludeVidPid	**{m|o}:vid-***xxx1***_pid-***yyy1***[;vid-***xxx2***_pid-***yyy2***;...]**	undefined	Use this option to exclude or include a specified composite USB device from splitting by Vendor and Product IDs . The format of the setting is **vid-***xxx1***_pid-***yyy1***[;vid-***xxx2***_pid-***yyy2***;...]**. You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID. Example: **m:vid-0f0f_pid-55****
viewusb.SplitVidPid	**{m|o}: vid-***xxxx***_pid-***yyyy***([exintf:***zz***[;exintf:***ww***]])[;...]**	undefined	Set this option to treat the components of a composite USB device specified by Vendor and Product IDs as separate devices. The format of the setting is **vid-***xxxx***_pid-***yyyy***(exintf:***zz***[;exintf:***ww***])**. You can use the **exintf** keyword to exclude components from redirection by specifying their interface number. You must specify ID numbers in hexadecimal, and interface numbers in decimal including any leading zero. You can use the wildcard character (*) in place of individual digits in an ID. Example: **o:vid-0f0f_pid-***(exintf-01);vid-0781_pid-554c(exintf:01;exintf:02) **Note** Horizon does not automatically include the components that you have not explicitly excluded. You must specify a filter policy such as **Include VidPid Device** to include those components.

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

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

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

Option	Value	Default	Description
CDREnable	`true` or `false`	`true`	Use this option to enable or disable the Client Drive Redirection (CDR) feature.
CollaborationEnable	`true` or `false`	`true`	Use this option to enable or disable the Session Collaboration feature on Linux desktops.
EndpointVPNEnable	`true` or `false`	`false`	Set this option to specify if the client's physical network card IP address or the VPN IP address is to be used when evaluating the endpoint IP address against the range of endpoint IP addresses used in the Dynamic Environment Manager Console. If the option is set to `false`, the client's physical network card IP address is used. Otherwise, the VPN IP address is used.

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

Option	Value	Default	Description
HelpDeskEnable	`true` or `false`	`true`	Set this option to enable or disable the Help Desk Tool feature.
KeyboardLayoutSync	`true` or `false`	`true`	Use this option to specify whether to synchronize a client's system locale list and current keyboard layout with Horizon Agent for Linux desktops. When this setting is enabled or not configured, synchronization is allowed. When this setting is disabled, synchronization is not allowed. This feature is supported only for Horizon Client for Windows, and only for the English, French, German, Japanese, Korean, Spanish, Simplified Chinese, and Traditional Chinese locales.
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.
NetbiosDomain	A text string, in all caps		When configuring True SSO, use this option to set the NetBIOS name of your organization's domain.
OfflineJoinDomain	`pbis` or `samba`	`pbis`	Use this option to set the instant-clone offline domain join. The available methods to perform an offline domain join are the PowerBroker Identity Services Open (PBISO) authentication and the Samba offline domain join. If this property has a value other than `pbis` or `samba`, the offline domain join is ignored.
RunOnceScript			Use this option to rejoin the cloned virtual machine to Active Directory. Set the `RunOnceScript` option after the host name has changed. The specified script is run only once after the first host name change. The script is run with the root permission when the agent service starts and the host name has been changed since the agent installation. For example, for the winbind solution, you must join the base virtual machine to Active Directory with winbind, and set this option to a script path. The script must contain the domain rejoin command `/usr/bin/net ads join -U <ADUserName>%<ADUserPassword>`. After VM Clone, the operating system customization changes the host name. When the agent service starts, the script is run to join the cloned virtual machine to Active Directory.
RunOnceScriptTimeout		120	Use this option to set the timeout time in seconds for the RunOnceScript option. For example, set `RunOnceScriptTimeout=120`

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

Option	Value	Default	Description
SSLCiphers	A text string	!aNULL:kECDH +AESGCM:ECDH +AESGCM:RSA +AESGCM:kECDH +AES:ECDH+AES:RSA +AES	Use this option to specify the list of ciphers. You must use the format that is defined in https://www.openssl.org/docs/manmaster/man1/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.
SSODesktopType	`UseGnomeClassic` or `UseGnomeFlashback` or `UseGnomeUbuntu` or `UseMATE` or `UseKdePlasma`	N/A	This option specifies the desktop environment to use, instead of the default desktop environment, when SSO is enabled. You must first ensure that the selected desktop environment is installed on your desktop before specifying to use it. After this option is set in an Ubuntu 16.04/18.04 desktop, the option takes effect regardless if the SSO feature is enabled or not. If this option is specified in a RHEL.x/CentOS 7.x desktop, the selected desktop environment is used only if SSO is enabled. **Note** This option is not supported on RHEL/CentOS 8.0 and RHEL/CentOS 6.x desktops. Horizon 7 only supports the Gnome desktop environment on RHEL/CentOS 8.0 desktops. See Desktop Environment for more information on how to set up KDE as the default desktop environment when SSO is enabled on RHEL/CentOS 6.x desktops.
SSOEnable	`true` or `false`	`true`	Set this option to enable/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 are as follows: ■ SSOUserFormat=[domain]\\[username] ■ SSOUserFormat=[domain]+[username] ■ SSOUserFormat=[username]@[domain]
Subnet	A value in CIDR IP address format	[subnet]	Set this option to a subnet which other machines can use to connect to the Horizon Agent for Linux. If there is more than one local IP address with different subnets, the local IP address in the configured subnet is used for connecting to the Horizon Agent for Linux. You must specify the value in the CIDR IP address format. For example, Subnet=123.456.7.8/24.

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

Option	Value	Default	Description
UEMEnable	`true` or `false`	`false`	Set this option to enable or disable Dynamic Environment Manager smart policies. If the option is set to enable, and the condition in the Dynamic Environment Manager smart policy is met, then the policies are enforced.
UEMNetworkPath	A text string		This option must be set to the same network path that is set in Dynamic Environment Manager Console. The path must be in the format similar to `//10.111.22.333/view/LinuxAgent/UEMConfig`.

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.

Using Smart Policies

You can use Smart Policies to create policies that control the behavior of the USB redirection, clipboard redirection, and client drive redirection features on specific remote Linux desktops.

You can create policies for user environment settings that control the behavior of the USB redirection, virtual printing, clipboard redirection, client drive redirection, Web and Chrome file transfer features, and bandwidth profiles in a published desktop or application. Horizon Smart Policies for user environment settings are applied during login and can be refreshed during reconnect of a session. To reapply Horizon Smart Policies when a user reconnects to a session, you can configure a triggered task.

You can create policies for computer environment settings that Dynamic Environment Manager applies while end users' computers boot. These Horizon Smart Policies control the behavior of Flash multi-media redirection, integrated printing, and USB redirection. Horizon Smart Policies for computer environment settings are applied during computer boot and can be refreshed during the reconnection of a session.

With Smart Policies, you can create policies that take effect only if certain conditions are met. For example, you can configure a policy that disables the client drive redirection feature if a user connects to a remote desktop from outside your corporate network.

Requirements for Smart Policies

To use Smart Policies, your Horizon 7 environment must meet certain requirements.

- You must install Horizon Agent 7.5 or later and VMware Dynamic Environment Manager 9.4 or later on the remote desktops that you want to manage with Smart Policies.
- Users must use Horizon Client 4.8 or later to connect to remote Linux desktops that you manage with Smart Policies.

- The `DEMEnable` option must be enabled and the `DEMNetworkPath` option must be set in the `/etc/vmware/viewagent-custom.conf` file. See Setting Options in Configuration Files on a Linux Desktop.
- You must install the client packages for accessing network shared storage. On an Ubuntu 18.04 system, for example, install the `nfs-common` package for NFS-enabled shared storage and the `cifs-utils` package for Samba-enabled storage.

Installing Dynamic Environment Manager

To use HorizonSmart Policies to control the behavior of remote desktop features on a remote Linux desktop, you must install Dynamic Environment Manager 9.4 or later on a remote Windows desktop.

You can download the Dynamic Environment Manager installer from the VMware Downloads page. You can install the Dynamic Environment Manager Management Console component on any Windows desktop from which you want to manage the Dynamic Environment Manager environment. From the Dynamic Environment Manager Management Console on a Windows desktop, you can control the behavior of remote desktop features on a remote Linux desktop.

For an RDS desktop pool, you install Dynamic Environment Manager on the RDS host that provides the published desktop sessions.

For Dynamic Environment Manager system requirements and complete installation instructions, see the *Installing and Configuring VMware Dynamic Environment Manager* document.

Configuring Dynamic Environment Manager

You must configure Dynamic Environment Manager before you can use it to create smart policies for remote desktop features.

To configure Dynamic Environment Manager, follow the configuration instructions in the *VMware Dynamic Environment Manager Administration Guide*.

Horizon Smart Policy Settings

You control the behavior of remote features in Dynamic Environment Manager by creating a Horizon smart policy.

You can create policies for user environment settings that control the behavior of the USB redirection, virtual printing, clipboard redirection, client drive redirection, Web and Chrome file transfer features, and bandwidth profiles in a published desktop or application. Horizon Smart Policies for user environment settings are applied during login and can be refreshed during reconnect of a session. To reapply Horizon Smart Policies when a user reconnects to a session, you can configure a triggered task. See the complete list of policies in the topic "Configure Horizon Smart Policies for User Environment Settings" in the *VMware Dynamic Environment Manager Administration Guide*.

You can create policies for computer environment settings that Dynamic Environment Manager applies while end users' computers boot. These Horizon Smart Policies control the behavior of Flash multi-media redirection, integrated printing, and USB redirection. Horizon Smart Policies for computer environment settings are applied during computer boot and can be refreshed during the reconnection of a session. See the complete list of policies in the topic "Configure Horizon Smart Policies for Computer Environment Settings" in the *VMware Dynamic Environment Manager Administration Guide*.

In general, Horizon smart policy settings that you configure for remote features in Dynamic Environment Manager override any equivalent registry key and group policy settings.

Adding Conditions to Horizon Smart Policy Definitions

When you define a Horizon Smart Policy in Dynamic Environment Manager, you can add conditions that must be met for the policy to take effect. For example, you can add a condition that disables the client drive redirection feature only if a user connects to the remote desktop from outside your corporate network.

Important You must add the following conditions to a Horizon Smart Policy definition in order for the supported policy settings to take effect in a remote Linux desktop. These are the only conditions that are currently supported. If other conditions are set, the end result of the condition evaluation is false.

Table 6-3. Required Conditions for Remote Linux Desktops

Condition	Description
`Operating System Architeacture`	Checks the architecture of the operating system. The value must be set to Linux.
`Endpoint IP address`	Checks whether the endpoint IP address is in or not in the specified range. Empty fields at the start of the range are interpreted as 0, and the ones at the end as 255.

You can, however, set multiple `Endpoint IP address` conditions, as shown in the following example.

```
Operating system is Linux
AND Endpoint IP address is in range 11.22.33.44 – 11.22.33.54
OR Endpoint IP address is in range 11.22.33.66 – 11.22.33.77
```

For detailed information about adding and editing conditions in the Dynamic Environment Manager Management Console, see the *VMware Dynamic Environment Manager Administration Guide*.

Create a Horizon Smart Policy in Dynamic Environment Manager

You use the Dynamic Environment Manager Management Console to create a Horizon smart policy in Dynamic Environment Manager. When you define a Horizon smart policy, you can add conditions that must be met for the smart policy to take effect.

Prerequisites

- Install and configure Dynamic Environment Manager. See Installing Dynamic Environment Manager and Configuring Dynamic Environment Manager.

- Become familiar with the conditions that you can add to Horizon Smart Policy definitions. See Adding Conditions to Horizon Smart Policy Definitions.
- Enable the `DEMEnable` option and configure the `DEMNetworkPath` option in the `/etc/vmware/viewagent-custom.conf` file. See Setting Options in Configuration Files on a Linux Desktop.

Note In a high-latency network, after saving your new or updated smart policy, allow Dynamic Environment Manager at least a minute to complete processing the changes before notifying the end users to connect to the affected desktops.

You can create policies for user environment settings that control the behavior of the USB redirection, virtual printing, clipboard redirection, client drive redirection, Web and Chrome file transfer features, and bandwidth profiles in a published desktop or application. Horizon Smart Policies for user environment settings are applied during login and can be refreshed during reconnect of a session. To reapply Horizon Smart Policies when a user reconnects to a session, configure a triggered task.

You can create policies for computer environment settings that Dynamic Environment Manager applies while end users' computers boot. These Horizon Smart Policies control the behavior of Flash multi-media redirection, integrated printing, and USB redirection. Horizon Smart Policies for computer environment settings are applied during computer boot and can be refreshed during the reconnection of a session.

For complete information about using the Dynamic Environment Manager Management Console, see the *VMware Dynamic Environment Manager Administration Guide* document.

Procedure

1. In the Dynamic Environment Manager Management Console, select the **User Environment** to create a policy for user environment settings or the **Computer Environment** tab to create a policy for computer environment settings.

   Existing Horizon smart policy definitions, if any, appear in the Horizon Smart Policies pane.
2. Select **Horizon Smart Policies** and click **Create** to create a new smart policy.
3. Select the **Settings** tab and define the smart policy settings.
   a. In the General Settings section, enter a name for the smart policy in the **Name** text box.

      For example, if the smart policy affects the client drive redirection feature, you might name the smart policy CDR.

   b. In the Horizon Smart Policy Settings section, select the remote desktop features and settings to include in the smart policy.

      You can select multiple remote desktop features.
4. Add the conditions required to use the new smart policy with remote Linux desktops.
   a. Select the **Conditions** tab, click **Add**, and select the **Operating System Architecture** condition.
   b. Set the value to **`Linux`**.

```
Operating System is Linux
```

c Click **Add** and select the **Endpoint IP Address** condition.

The **AND** operator is added by default.

d In the Endpoint IP Address dialog box, set the endpoint IP address range, and click **OK**.

Following is an example of the condition statement.

```
Operating System is Linux
AND Endpoint IP address is in range 11.22.33.44 – 11.22.33.54
```

5 Click **Save** to save the smart policy.

Dynamic Environment Manager processes the Horizon smart policy each time a user connects or reconnects to the remote desktop.

Dynamic Environment Manager processes multiple smart policies in alphabetical order based on the smart policy name. Horizon smart policies appear in alphabetical order in the Horizon Smart Policies pane. If smart policies conflict, the last smart policy processed takes precedence. For example, if you have a smart policy named Sue that enables USB redirection for the user named Sue, and another smart policy named Pool that disables USB redirection for the desktop pool named Ubuntu1604, the USB redirection feature is enabled when Sue connects to a remote desktop in the Ubuntu1604 desktop pool.

Example Blast Settings for Linux Desktops

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

Example VMware Blast Extreme Protocol Settings

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

Table 6-4. Example Blast Configuration Options in `/etc/vmware/config`

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

Table 6-4. Example Blast Configuration Options in `/etc/vmware/config` (continued)

Option name	Parameter	High-speed LAN	LAN	Dedicated WAN	Broadband WAN	Low-speed WAN	Extremely Low speed
Display Quality (H.264)	RemoteDisplay.qpmaxH264	28	36	36	36	36	42
Display Quality (H.264)	RemoteDisplay.qpminH264	10	10	10	10	10	10

Examples of Client Drive Redirection Options for Linux Desktops

Configure client drive redirection (CDR) options to determine whether a local system's shared folders and drives can be accessed from the remote Linux desktops.

Configure CDR settings by adding entries to the `/etc/vmware/config` file.

The following configuration example shares the `d:\ebooks` and `C:\spreadsheets` folders, makes both folders read-only, and prevents the client from sharing more folders.

```
cdrserver.forcedByAdmin=true
cdrserver.sharedFolders=d:\ebooks,;c:\spreadsheets,
cdrserver.permissions=R
```

In the previous example, the comma **","** placed after **ebooks** and **spreadsheets** is mandatory for correct option parsing.

Any **"R"** included in the `cdrserver.sharedFolders`option would impact all the folders listed in that setting. In the following example, the **ebooks** and **spreadsheets** folders are both read-only even if the **R** value is only placed after **/home/jsmith** folder path.

```
cdrserver.sharedFolders=d:\ebooks,;c:\spreadsheets,;/home/jsmith,R
```

Create and Manage Linux Desktop Pools

7

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

Horizon for Linux supports the following desktop pool types:

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

To create a manual desktop pool with a vCenter virtual machine, you must install Horizon Agent on all virtual machines. Then, use the Connection Server desktop pool creation wizard to add the virtual machines to the desktop pool. To clone a large number of virtual machines, see Overview of Bulk Deployment of Linux Desktops.

To create an automated full-clone desktop pool, you must install Horizon 7 Agent on a Linux virtual machine template. Then, use the Connection Server desktop pool creation wizard to clone full virtual machines.

To create an instant clone floating desktop pool, you must install Horizon 7 Agent on a Linux virtual machine with PBIS Open environment setup, and create a template from it. Then, use the Connection Server desktop pool creation wizard to create instant-clone floating desktop pool.

This chapter includes the following topics:

Create a Manual Desktop Pool for Linux

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

The following procedure provides guidelines for configuring the mandatory settings for a Linux-based manual desktop pool. For more information about creating manual desktop pools, see *Setting Up Virtual Desktops in Horizon Console*.

Prerequisites

- Verify that Horizon Agent is installed on the Linux guest operating systems. See Install Horizon Agent on a Linux Virtual Machine.
- Verify that VMware vCenter Server is added to Horizon Connection Server .

Procedure

1. In Horizon Console, add a manual desktop pool.

   Select **Inventory > Desktops > Add**.

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

2. Select **Manual Desktop Pool**.
3. Select virtual machines that are either managed or unmanaged by vCenter Server and click **Next**.
4. Select either dedicated or floating user assignments for the machines in the desktop pool and click **Next**.
5. Follow the prompts in the wizard to create the pool.

   On the Desktop Pool Settings page, set the following options.

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

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

6. After creating the desktop pool, entitle users to the machines in the desktop pool. In Horizon Console, select the desktop pool, select **Entitlements** > **Add entitlement**, and add users or groups.

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

Manage Linux Desktop Pools

When you create a manual desktop pool and add Linux machines to the pool, you can manage the manual desktop pools by configuring the settings. You must add only Linux guest operating systems to the manual desktop pool. If the pool contains both Windows and Linux guest operating systems, the pool is treated as a Windows pool, and you cannot connect to the Linux desktops.

Support for Managing Operations

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

  You can either remove virtual machines from Horizon 7 or delete virtual machines from the disk.

Support for Remote Settings

Table 7-1. Remote Settings

Remote Setting	Options
Remote Machine Power Policy	■ Take no power action ■ Ensure machines are always powered on ■ Suspend ■ Power off
Automatically logoff after disconnect	■ Immediately ■ Never ■ After n minutes
Allow users to reset/restart their machines	■ Yes ■ No
Allow user to initiate separate sessions from different client devices	■ Yes ■ No
"Delete machine after logoff" for Automated Desktop Pool with Full Clone and Floating	■ Yes ■ No

Support for Horizon Console Operations

- Disconnect Session
- Logoff Session
- Reset/Restart Desktop
- Send Message

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

- Update-UserOwnership

- Remove-UserOwnership

Note Do not change **Remote Display Protocol** settings. These settings must remain the same as specified during desktop pool creation.

Setting	Option
Default display protocol	VMware Blast
Allow user to choose protocol	No
3D Renderer	■ Manage using vSphere Client for 2D or vDGA ■ NVIDIA GRID vGPU

For more information, see the *VMware Horizon Console Administration* documentation.

Create an Automated Full-Clone Desktop Pool for Linux

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

The following procedure provides guidelines for configuring the mandatory settings for a Linux-based automated full-clone desktop pool. For more information about creating automated full-clone desktop pools, see *Setting Up Virtual Desktops in Horizon Console*.

Prerequisites

- Verify that Horizon Agent is installed on the Linux guest operating systems. See Install Horizon Agent on a Linux Virtual Machine.
- Before you perform virtual machine cloning, create a virtual machine template that the clones are based on. See Create a Virtual Machine Template for Cloning Linux Desktop Machines.
- If you use the Winbind solution to join the Linux virtual machine to Active Directory, you must finish configuring the Winbind solution in the virtual machine template.
- If you use the Winbind solution, you must run the domain join command on the virtual machine. Include the command in a shell script and specify the script path to the Horizon Agent option `RunOnceScript` in `/etc/vmware/viewagent-custom.conf`. For more information, see Setting Options in Configuration Files on a Linux Desktop.
- Verify that vCenter Server is added to Horizon Connection Server.

Procedure

1 Create a guest customization specification.

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

Setting	Value
Target Virtual Machine OS	Linux
Computer Name	Use the virtual machine name.
Domain	Specify the domain of the Horizon 7 environment.
Network Settings	Use standard network settings.
Primary DNS	Specify a valid address.

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

2 In Horizon Console, add an automated desktop pool.

Select **Inventory > Desktops > Add**.

3 Select **Automated Desktop Pool** and click **Next**.

4 Select **Full Virtual Machines**, select the vCenter Server instance, and click **Next**.

5 Follow the prompts in the wizard to create the pool.

a On the Desktop Pool Settings page, set the following options.

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

b When prompted, set the **Virtual Machine Naming** options.

Option	Description
Specify names manually	Enter names manually.
Naming Pattern	For example, specify `LinuxVM-{n}`. You must also specify the following desktop pool sizing options: ■ Maximum number of machines ■ Number of spare, powered-on machines

c When prompted, select the vCenter Server settings in sequence.

You cannot skip a vCenter Server setting:

1 Template

2 VM folder location

3 Host or cluster

4 Resource pool

5 Datastores

6 After creating the desktop pool, entitle users to the machines in the desktop pool. In Horizon Console, select the desktop pool, select **Entitlements** > **Add entitlement**, and add users or groups.

7 Wait until all the Linux virtual machines in the desktop pool become available.

Create an Instant-Clone Floating Desktop Pool for Linux

You can create an instant-clone floating desktop pool for Linux virtual machines using the **Add Desktop Pool** wizard. After creating an instant-clone floating desktop pool, you can use the Linux virtual machines as remote desktops in a Horizon 7 deployment.

Horizon 7 Agent for Linux supports instant-clone desktop pools only on systems with Ubuntu 18.04/16.04, RHEL 7.1 or later, RHEL 8.0, or SLED/SLES 12.x.

Note vGPU graphics capabilities are not supported on instant-clone desktop pools created from Linux desktops.

The following procedure provides guidelines for configuring the mandatory settings for a Linux-based instant-clone desktop pool. For more information about creating instant-clone desktop pools, see *Setting Up Virtual Desktops in Horizon Console*.

Prerequisites

- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing Linux operating systems. For more information, see Create a Virtual Machine and Install Linux.
- Understand the steps for AD integration using the PBISO authentication solution or Samba Winbind offline join. For more information, see Configure PowerBroker Identity Services Open (PBISO) Authentication or Configure the Samba Offline Domain Join .

  Note To create an instant-clone desktop pool from a Linux virtual machine running RHEL 8.0, perform the AD integration using Samba Winbind offline join. Instant-clone desktop pools are not supported for RHEL 8.0 virtual machines that use PBISO authentication.
- Familiarize yourself with the installation steps for Horizon 7 Agent for Linux. For more information, see Install Horizon Agent on a Linux Virtual Machine.
- Understand the steps to take a snapshot of a powered off Linux VM using VMware vSphere Web Client. See "Take a Snapshot in the VMware Host Client" in *vSphere Single Host Management - VMware Host Client*.
- Verify that vCenter Server is added to Horizon Connection Server.

Procedure

1 Create a Linux virtual machine (VM) with Ubuntu 18.04/16.04, RHEL 7.1 or later, RHEL 8.0, or SLED/SLES 12.x installed.

  For more information, see Create a Virtual Machine and Install Linux.

2 Manually install Open VMware Tools (OVT) on your Ubuntu 18.04/16.04 machine using the following command:

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

See Prepare a Linux Machine for Remote Desktop Deployment for additional information.

3 Install any dependency packages that are required for the Linux distribution.

See Install Dependency Packages for Horizon Agent for more information.

4 Install Horizon Agent for Linux in the Linux VM.

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

See Install Horizon Agent on a Linux Virtual Machine for details.

5 Integrate your Linux VM with Active Directory.

- To use the PBISO authentication solution, perform the following steps:
  a Download PBIS Open 8.5.6 or later from `https://www.beyondtrust.com/products/powerbroker-identity-services-open/` and install it on your Linux VM.

  ```
  # sudo ./pbis-open-8.5.6.2029.linux.x86_64.deb.sh
  ```

  b Integrate your Linux VM with Active Directory using the information in PowerBroker Identity Services Open (PBISO) Authentication section in Integrating Linux with Active Directory.
- To use Samba Winbind offline join, set the `OfflineJoinDomain` to **`samba`** in the `/etc/vmware/viewagent-custom.conf` file.

  Note You must use Samba Winbind to integrate a RHEL 8.0 VM with Active Directory. Otherwise, the creation of the instant-clone floating desktop pool fails.
- If you want to disable offline domain join, you must set the `OfflineJoinDomain` option to **`none`** in the `/etc/vmware/viewagent-custom.conf` file. Otherwise, the creation of the instant-clone floating desktop pool fails.

6 If your DHCP server does not broadcast to a DNS server, specify a DNS server for your Linux system.

A new virtual network adapter is added when a new instant-cloned VM is created. Any setting in the network adapter, such as the DNS server, in the VM template is lost when the new network adapter is added to the instant-cloned VM. PBIS requires a valid DNS server and the FQDN mapping in the `/etc/hosts` is not acceptable. To avoid losing the DNS Server setting when the new network adapter is added to the cloned VM, you must specify a DNS server in your Linux system. For example, in an Ubuntu 16.04 system, specify the DNS server by adding the following lines in the `/etc/resolvconf/resolv.conf.d/head` file.

```
nameserver 10.10.10.10
search mydomain.org
```

7 (Optional) If you want to add an NFS mount in the `/etc/fstab` file of the master Linux VDI instant-clone agent, use one of the following methods.

- Add a 'soft' flag in `/etc/fstab`, such as:

```
10.111.222.333:/share     /home/nfsmount    nfs
rsize=8192,wsize=8192,timeo=14,soft,intr,tcp
```

- If you do not want to use the 'soft' flag in `/etc/fstab`, you cannot configure the `/etc/fstab` in the master Linux VM image. You can write a power-off script to configure the `/etc/fstab` file, and then specify this power-off script for the ClonePrep tool. For more information, see the *VMware Horizon Console Administration* document.

8 Shut down the Linux VM and create a master image by creating a snapshot of your powered off Linux VM using VMware vSphere® Web Client.

See "Take a Snapshot in the VMware Host Client" in *vSphere Single Host Management - VMware Host Client* for information.

9 In Horizon Console, add an automated desktop pool.

Select **Inventory > Desktops > Add**.

10 Select **Automated Desktop Pool** and click **Next**.

11 Select **Instant Clones**, select the vCenter Server instance, and click **Next**.

12 Follow the prompts in the wizard to create the pool.

a When prompted, set the **Virtual Machine Naming** options.

Option	Description
Enable provisioning	Select this option.
Stop provisioning on error	Select this option.
Naming Pattern	Specify a pattern that Horizon 7 uses as a prefix in all the desktop VM names, followed by a unique number. For example, specify **`LinuxVM-{n}`**.
Max number of machines	Specify the total number of machines in the pool.
Number of spare (powered on) machines	Specify the number of desktop VMs to keep available to users.
Provision all machines up front	Select this option to have Horizon 7 provision the number of VMs specified in **Max number of machines**.

b When prompted, select **Use VMware Virtual SAN** for the storage management policy.

c When prompted, specify the Domain setting, AD container, and any extra customization scripts that must be run after the VM is cloned.

Important When you use ClonePrep power-off or post-synchronization scripts, ensure that the scripts are located in the `/var/userScript` folder, owned by the root user, and have the file permissions set to 700.

In Horizon Console, you can view the desktop VMs as they are added to the pool by selecting **Inventory** > **Desktops**.

After you create the pool, do not delete the master image or remove it from the vCenter Server inventory if the pool exists. If you remove the master image VM from the vCenter Server inventory by mistake, you must add it back and then do a push image using the current image.

What to do next

Entitle users to access the pool. See "Add Entitlements to Desktop Pools" in *Setting Up Virtual Desktops in Horizon Console*.

Broker PowerCLI Commands

The Horizon PowerCLI cmdlets, which are used to perform various administration tasks on Connection Server and a Windows desktop, can also be used for Linux desktops.

Create a Manual Desktop Pool

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

The following options and values are mandatory for the Linux desktop.

- `DefaultProtocol Blast`
- `AllowProtocolOverride $false`
- `threedRender usevc|vgpu`. For a vGPU desktop, use `-threedRender vgpu` and for a 2D/DGA desktop, use `-threedRender usevc` .

Examples

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

  ```
  Add-ManualPool -DefaultProtocol Blast -AllowProtocolOverride $false -threedRender usevc -Pool_id
  LinuxDesktop -Id (Get-DesktopVM -Name LinuxVM-01).id -Persistence NonPersistent -Vc_name
  myvc.myorg.org
  ```

- Create a dedicated Linux vGPU desktop pool named LinuxDesktop with all VMs that begin with the VM name as LinuxVM-.

  ```
  Get-DesktopVM | Where-Object {$_.Name.StartsWith("LinuxVM-")} | Add-ManualPool -DefaultProtocol
  Blast -AllowProtocolOverride $false -Persistence Persistent -threedRender vgpu  -Pool_id
  LinuxDesktop
  ```

- Create floating Linux desktop pool LinuxDesktop with the first RHEL 6 x64 VM.

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

Create a Full-Clone Automated Desktop Pool

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

The following options and values are mandatory for Linux desktops.

- `DefaultProtocol Blast`
- `AllowProtocolOverride $false`
- `threedRender usevc|vgpu` For a vGPU desktop, use `-threedRender vgpu` and for a 2D desktop, use `-threedRender usevc`.

Example

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

Add or Remove Desktop Pool Entitlement

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

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

- Remove the entitlement of a domain user group of mydomain.org domain from LinuxDesktop.

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

Assign or Remove User to or from the VM in Dedicated Desktop Pool

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

```
Update-UserOwnership -Machine_id (Get-DesktopVM -Name "LinuxVM-01").machine_id -Sid (Get-User -
Name "myuser" | Where-Object {$_.cn -eq "myuser"}).sid
```

- Remove the **myuser** user from the LinuxVM-01 VM, which is in a dedicated desktop pool.

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

Logoff Desktop Connection

- Log out from the desktop session of myuser.

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

For more information on the broker PowerCLI cmdlet, see "Using the Horizon PowerCLI Module" in *Horizon 7 Integration*.

Bulk Deployment of Horizon 7 for Manual Desktop Pools

8

With Horizon Console, 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

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

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

PowerCLI	SSH
No need to install additional tools.	■ 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 and install Horizon Agent on the virtual machine.

See Create a Virtual Machine Template for Cloning Linux Desktop Machines.

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 Horizon 7 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.

4 Join the cloned VMs to the Active Directory (AD) domain if you are using the winbind solution. You can run the domain join command with example scripts below or use option `RunOnceScript` in `/etc/vmware/viewagent-custom.conf`, configured in the template virtual machine.

   See Sample Script to Join Cloned Virtual Machines to AD Domain or Sample Script to Join Cloned Virtual Machines to AD Domain Using SSH.

5 Update configuration options in virtual machines.

   See Sample Script to Upload Configuration Files to Linux Virtual Machines or Sample Script to Upload Configuration Files to Linux Virtual Machines Using SSH.

6 Create a desktop pool.

   See Create a Manual Desktop Pool for Linux.

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. This script can delete virtual machines from vCenter Server.

Overview of Bulk Upgrade of Linux Desktops

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

Bulk Upgrade Unmanaged Desktop

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

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

Bulk Upgrade Managed Desktop

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

Method	Description
In the template virtual machine, install or upgrade the new Horizon Agent and create a snapshot.	■ The user data and profile are lost since the existing virtual machines are deleted, unless the user data and profile are located on the share server such as NFS server. ■ After the virtual machine replacement, the state of the virtual machine on View Administrator might be missing. You must restart the broker service to fix it.
Use the sample script of upgrade to upload the new Horizon Agent to existing virtual machines and run the upgrade command.	User data and profile is retained.

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.
- 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 Virtual Desktops in Horizon 7* document.
- Familiarize yourself with the required video memory (vRAM) values for the monitors you must use with the virtual machine. See Virtual Machine Settings for 2D Graphics.
- Familiarize yourself with the steps for AD integration. See Chapter 3 Setting Up Active Directory Integration for Linux Desktops.
- Familiarize yourself with the steps to install Horizon Agent on Linux. See Chapter 5 Installing Horizon Agent.
- If necessary, familiarize yourself with the steps to configure options using the Horizon 7 configuration files. See Chapter 6 Configuration Options for Linux Desktops.
- If you plan to set up graphics, familiarize yourself with the steps. See Chapter 4 Setting Up Graphics for Linux Desktops.

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.

      Follow the vCPUs and vMemory size guidelines in the installation guide for your Linux distribution.

      For example, Ubuntu 18.04 specifies 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 graphics, which use the VMware driver. The vRAM size has no effect on vDGA or NVIDIA GRID vGPU machines, which use NVIDIA drivers.

      Follow the guidelines in Virtual Machine Settings for 2D Graphics. 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/CentOS 8.x, RHEL/CentOS 7.x, or SLED/SLES 12.x, install VMware Tools.

RHEL/CentOS 8.0, RHEL/CentOS 7.x, and SLED/SLES 12.x have Open VM Tools installed by default.

8 Install and configure the dependency packages.

a If the Linux distribution is running a version of Open VM Tools earlier than 9.10, install the deployPkg plug-in.

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

b If the Linux distribution is Ubuntu, refer to the following KB articles to determine the dependency packages to install and configure in the VM.

- See KB articles https://kb.vmware.com/s/article/2051469 and https://kb.vmware.com/s/article/59687 for Ubuntu 18.04 and 16.04.
- For Ubuntu 18.04, see KB article https://kb.vmware.com/s/article/56409 also.

9 For RHEL and CentOS, enable the Network Connection setting **Connect automatically**.

10 Perform the AD integration tasks.

11 Perform the steps to set up graphics.

12 Install Horizon agent.

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

See Chapter 5 Installing Horizon Agent.

13 Perform additional configurations using the Horizon 7 configuration files.

14 Shut down the virtual machine and create a snapshot.

Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops

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

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

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

The following example shows what the input file might contain.

```
VMName,Parentvm,CustomSpec,Datastore,Host,FromSnapshot,DeleteIfPresent
linux-001,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-002,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-003,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-004,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-005,Ubuntu1804x64,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://docs.vmware.com/en/VMware-Horizon-7/index.html.

Script Input

This script reads one input file, which is described in Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops. 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 only be full
- Whether to disable a vSphere VM console

Script Content

```
<#
Create Clones from a Master VM

The Tool supports creation of Full clone from Master VM.
#>
#-------------------------- Functions -------------------------
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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 ("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 full clone
if (($CloneType.length >0) -and ($CloneType -ne "full"))
{
    write-host  -ForeGroundColor Red "Clone type supports only '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
    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<"Full"> : Full
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 several 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. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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 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. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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 PowerCLI 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://docs.vmware.com/en/VMware-Horizon-7/index.html.

Script Input

This script reads one input file, which is described in Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops. This script also interactively asks for the following information:

- Acceptance of Horizon Agent EULA (end user license agreement)

- IP address of the vCenter Server
- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Administrator login name for the ESXi host
- Administrator password for the ESXi host
- User login name for the Linux guest operating system
- User password for the Linux guest operating system
- Horizon Agent tar ball path
- Upgrade to managed VM
- Install the Smartcard redirection feature

Script Content

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

#-----------------------------------------------------
Functions-----------------------------------------------------------------
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
        $input = Read-Host -AsSecureString
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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' -IsPassword $false
"------------------------------------------------------"
$UpgradeToManagedVM = GetInput -prompt 'Upgrade to managed VM ("yes" or "no")' -IsPassword $false
if (($UpgradeToManagedVM -ne "yes") -AND $UpgradeToManagedVM -ne "no")
{
    write-host  -ForeGroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host  -ForeGroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"------------------------------------------------------"

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

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

#check if file exists
if (!(Test-Path $csvFile))
{
write-host  -ForeGroundColor Red "CSV File not found"
exit
}
#-----------------------------------------------------
Functions-----------------------------------------------------------------
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;
```

```
    $md5Hash = ($md5HashWithFormat.replace("-","")).ToLower();
    return $md5Hash;
}

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

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

    if($output.Contains($installerMD5Hash))
    {
        Write-Host $VMName": Uploaded installer's MD5Sum matches the local installer's MD5Sum";
        Write-Host $VMName": Extract the installer and do installation";
        $cmd = "tar -xzvf VMware-*-linux-*.tar.gz"
```

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

        $cmd = "sudo setenforce 0";
        Write-Host "Set the selinux to permissive mode: $cmd"
        Invoke-VMScript  -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

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

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

        $cmd = "sudo shutdown -r +1&"
        Write-Host "Reboot to apply the 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: HorizonUser
Your VM guest OS user password: *******
--------------------------------------------------
```

```
Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
--------------------------------------------------------------------------------------------------------
Upgrade to managed VM ("yes" or "no"): yes
Install the Smartcard redirection feature ("yes" or "no"): no
```

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

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

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops. This script also interactively asks for the following information:

- Acceptance of Horizon Agent EULA (end user license agreement)
- IP address of the vCenter Server
- Administrator login name for the vCenter Server
- Administrator password for the vCenter Server
- Administrator login name for the ESXi host
- Administrator password for the ESXi host
- User login name for the Linux guest operating system
- User password for the Linux guest operating system
- Horizon Agent tar ball path
- Upgrade to managed VM
- Install the Smartcard redirection feature

Script Content

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

#-----------------------------------------------------
Functions-----------------------------------------------------------------
function GetInput
{
    Param($prompt, $IsPassword = $false)
```

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

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

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

    $VM= Get-VM $VM_Name
    $IP = $VM.guest.IPAddress[0]
```

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

}

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

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

#-----------------------------------------------------Handle
input-------------------------------------------------------------------
"-------------------------------------------------------"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-------------------------------------------------------"
$acceptEULA = GetInput -prompt 'Accept Linux View Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host  -ForeGroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$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 View Agent tar ball path' -IsPassword $false
"-------------------------------------------------------"
$UpgradeToManagedVM = GetInput -prompt 'Upgrade to managed VM ("yes" or "no")' -IsPassword $false
if (($UpgradeToManagedVM -ne "yes") -AND $UpgradeToManagedVM -ne "no")
{
    write-host  -ForeGroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
```

```
{
    write-host  -ForeGroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"------------------------------------------------------"

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

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

#check if file exists
if (!(Test-Path $csvFile))
{
write-host  -ForeGroundColor Red "CSV File not found"
exit
}
#-----------------------------------------------------
Functions-----------------------------------------------------------------
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;
    $md5Hash = ($md5HashWithFormat.replace("-","")).ToLower();
    return $md5Hash;
}

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

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

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

```
$destFolder = "/home/$guestUser/"

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

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


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

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

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

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

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

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

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

```
        exit;
    }
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit
```

Script Execution

The following messages are from an execution of the script:

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

Sample Script to Perform Operations on Linux Virtual Machines

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

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

To copy and paste the script content without page breaks, use the HTML version of this topic, available from the Horizon 7 documentation page at https://www.vmware.com/support/pubs/view_pubs.html.

Script Input

This script reads one input file, which is described in Input File for the Sample PowerCLI Scripts to Deploy Linux Desktops. 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($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.

Troubleshooting Linux Desktops 9

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:

Using Horizon Help Desk Tool in Horizon Console

Horizon Help Desk Tool is a Web application that you can use to get the status of Horizon 7 user sessions and to perform troubleshooting and maintenance operations.

In Horizon Help Desk Tool, you can look up user sessions to troubleshoot problems and perform desktop maintenance operations such as restart or reset desktops.

To configure Horizon Help Desk Tool, you must meet the following requirements:

- Horizon Enterprise edition license or Horizon Apps Advanced edition license for Horizon 7. To verify that you have the correct license, see the *Horizon 7 Administration* document.
- An event database to store information about Horizon 7 components. For more information about configuring an event database, see the *Horizon 7 Administration* document.
- The Help Desk Administrator role or the Help Desk Administrator (Read Only) role to log in to Horizon Help Desk Tool. For more information on these roles, see the *Horizon 7 Administration* document.
- Enable the timing profiler on each Connection Server instance to view login segments.

Use the following `vdmadmin` command to enable the timing profiler on each Connection Server instance:

```
vdmadmin –I –timingProfiler –enable
```

Use the following `vdmadmin` command to enable the timing profiler on a Connection Server instance that uses a management port:

```
vdmadmin –I –timingProfiler –enable –server {ip/server}
```

- Enable the `HelpDeskEnable` option in the `/etc/vmware/viewagent–custom.conf` configuration file.

Start Horizon Help Desk Tool in Horizon Console

Horizon Help Desk Tool is integrated into Horizon Console. You can search for a user that you want to troubleshoot problems for in Horizon Help Desk Tool.

Procedure

1. You can search for a user name in the User Search text box or navigate directly to the Horizon Help Desk Tool tool.
   - In Horizon Console, enter a user name in the User Search text box.
   - Select **Monitor > Help Desk** and enter a user name in the User Search text box.

   Horizon Console displays a list of users in the search results. The search can return up to 100 matching results.
2. Select a user name.

   The user information appears in a user card.

What to do next

To troubleshoot problems, click the related tabs in the user card.

Troubleshooting Users in Horizon Help Desk Tool

In Horizon Help Desk Tool, you can view basic user information in a user card. You can click tabs in the user card to get more details about specific components.

The user details can sometimes appear in tables. You can sort these user details by table columns.

- To sort a column by ascending order, click the column once.
- To sort a column by descending order, click the column twice.
- To not sort the column, click the column thrice.

Basic User Information

Displays basic user information such as user name, phone number, and email address of the user and the connected or disconnected status of the user. If the user has a desktop session, the status of the user is connected. If the user does not have any desktop sessions, the status of the user is disconnected.

You can click the email address to send a message to the user.

Sessions

The **Sessions** tab displays information about desktop sessions that the user is connected to.

You can use the **Filter** text box to filter desktop sessions.

Note The **Sessions** tab does not display session information for sessions that access VMs from vSphere Client or ESXi.

The **Sessions** tab includes the following information:

Table 9-1. Sessions tab

Option	Description
State	Displays information about the state of the desktop session. ■ Appears green, if the session is connected. ■ L, if the session is a local session or a session running in the local pod.
Computer Name	Name of the desktop session. Click the name to open the session information in a card. You can click the tabs in the session card to view additional information: ■ The **Details** tab displays the user information such as the VM information, CPU, or memory usage. ■ The **Processes** tab displays information about CPU and memory related processes.
Protocol	Display protocol for the desktop session.
Type	Displays whether the desktop is a published desktop or a virtual machine desktop.
Connection Time	The time the session connected to Connection Server.
Session Duration	The duration of time the session remained connected to Connection Server.

Desktops

The **Desktops** tab displays information about the published desktops or virtual desktops that the user is entitled to use.

Table 9-2. Desktops

Option	Description
State	Displays information about the state of the desktop session. ■ Appears green, if the session is connected.
Desktop Pool Name	Name of the desktop pool for the session.
Desktop Type	Displays whether the desktop is a published desktop or virtual machine desktop. **Note** Does not display any information if the session is running in a different pod in the pod federation.
Type	Displays information about the type of desktop entitlement. ■ Local, for a local entitlement.
vCenter	Displays the name of the virtual machine in vCenter Server. **Note** Does not display any information if the session is running in a different pod in the pod federation.
Default Protocol	Default display protocol for the desktop session.

Activities

The **Activities** tab displays the event log information about the user's activities. You can filter activities by a time range such as the Last 12 hours or Last 30 Days or by administrator name. Click **Help Desk Event Only** to filter only by Horizon Help Desk Tool activities. Click the refresh icon to refresh the event log. Click the export icon to export the event log as a file.

Note The event log information is not displayed for users in a Cloud Pod Architecture environment.

Table 9-3. Activities

Option	Description
Time	Select a time range. Default is the last 12 hours. ■ **Last 12 Hours** ■ **Last 24 Hours** ■ **Last 7 Days** ■ **Last 30 Days** ■ **All**
Admins	Name of the administrator user.
Message	Displays messages for a user or administrator that are specific to the activities that the user or administrator performed.
Resource Name	Displays information about the desktop pool or virtual machine name on which the activity was performed.

Session Details for Horizon Help Desk Tool

The session details appear on the **Details** tab when you click a user name in the **Computer Name** option on the **Sessions** tab. You can view details for Horizon Client, the virtual or published desktop, and CPU and memory details.

Client

Displays information that depends on the type of Horizon Client and includes details such as user name, version of Horizon Client, IP address of the client machine, and the operating system of the client machine.

Note If you upgraded Horizon Agent, you must also upgrade Horizon Client to the latest version. Else, no version is displayed for Horizon Client. For more information about upgrading Horizon Client, see the *Horizon 7 Upgrades* document.

VM

Displays information about virtual desktops or published desktops.

Table 9-4. VM Details

Option	Description
Computer Name	Name of the desktop session.
Agent Version	Horizon Agent version.
OS Version	Operating System version.
Connection Server	The Connection Server that the session connects to.
Pool	Name of the desktop pool.
vCenter	IP address of vCenter Server.
Session State	State of the desktop session. The session states can be connected or disconnected.
Session Duration	The time the session remained connected to Connection Server.
State Duration	The time the session remained in the same state.
Logon Time	The logon time of the user who logged in to the session.
Logon Duration	The duration of time that the user is logged on the Linux desktop.

User Experience Metrics

Displays performance details for a virtual or published desktop session that uses the VMware Blast display protocol. To view these performance details, click **More**. To refresh these details, click the refresh icon.

Table 9-5. Blast Display Protocol Details

Option	Description
Frame Rate	The frame rate, in frames per second, in a Blast session.
Skype Status	For Linux desktop sessions, this option appears as N/A .
Blast Session Counters	■ **Estimated Bandwidth (Uplink)**. Estimated bandwidth for an uplink signal. ■ **Packet Loss (Uplink)**. Percentage of packet loss for an uplink signal.
Blast Imaging Counters	■ **Transmitted Bytes**. Total number of bytes for imaging data that have been transmitted for a Blast session. ■ **Received Bytes**. Total number of bytes for imaging data that have been received for a Blast session.
Blast Audio Counters	■ **Transmitted Bytes**. Total number of bytes for audio data that have been transmitted for a Blast session. ■ **Received Bytes**. Total number of bytes for audio data that have been received for a Blast session.
Blast CDR Counters	■ **Transmitted Bytes**. Total number of bytes for Client Drive Redirection data that have been transmitted for a Blast session. ■ **Received Bytes**. Total number of bytes for Client Drive Redirection data that have been received for a Blast session.

CPU and Memory Usage and Network and Disk Performance

Displays charts for CPU and memory usage of the virtual or published desktop and the network or disk performance for the Blast display protocol.

Note Following a start or a restart of Horizon Agent on the desktop, the performance charts might not display the timeline immediately. The timeline appears after a few minutes.

Table 9-6. CPU Usage

Option	Description
Session CPU	CPU usage of the current session.
Host CPU	CPU usage of the virtual machine to which the session is assigned.

Table 9-7. Memory Usage

Option	Description
Session Memory	Memory usage of the current session.
Host Memory	Memory usage of the virtual machine to which the session is assigned.

Table 9-8. Network Performance

Option	Description
Latency	Displays a chart for the latency for the PCoIP or Blast session. The latency time is the Round-Trip Time in milliseconds. The performance counter that tracks this latency time is **VMware Blast Session Counters > RTT**.

Table 9-9. Disk Performance

Option	Description
Read	The number of read Input/Output (I/O) operations per second.
Write	The number of write I/O operations per second.
Disk Latency	Displays a chart for the disk latency. The disk latency is the time in milliseconds from the Input/Output Operations Per Second (IOPS) data retrieved from the Windows performance counters.
Average Read	Average number of random read I/O operations per second.
Average Write	Average number of random write I/O operations per second.
Average Latency	Average latency time in milliseconds from the IOPS data retrieved from the Windows performance counters.

Session Logon Segments

Displays the logon duration and usage segments that are created during logon.

Table 9-10. Session Logon Segments

Option	Description
Logon duration	The length of time calculated from the time the user clicks the desktop pool to the time when the user logged on to the Linux desktop.
Session Logon Time	The length of time that the user was logged in to the session.
Logon Segments	Displays the segments that are created during logon. ■ **Brokering**. Total time for Connection Server to process a session connect or reconnect. Calculated from the time the user clicks the desktop pool to the time when the tunnel connection is set up. Includes the times for Connection Server tasks such as user authentication, machine selection, and machine preparation for setting up the tunnel connection. ■ **Interactive**. Total time for Horizon Agent to process a session connect or reconnect. Calculated from the time when Blast Extreme uses the tunnel connection to the time when the user logged on to the Linux desktop. ■ **Protocol Connection**. Total time taken for the PCoIP or Blast protocol connection to complete during the logon process. ■ **Logon Script**. Total time taken for a logon script to execute from start to completion. ■ **Authentication**. Total time for Connection Server to authenticate the session. ■ **VM Start**. Total time taken to start a VM. This time includes the time for booting the operating system, resuming a suspended machine, and the time it takes Horizon Agent to signal that it is ready for a connection.

Session Processes for Horizon Help Desk Tool

The session processes appear on the **Processes** tab when you click a user name in the **Computer Name** option on the **Sessions** tab.

Processes

For each session, you can view additional details about CPU and memory related processes. For example, if you notice that the CPU and memory usage for a session is abnormally high, you can view the details for the process on the **Processes** tab.

For RDS host sessions, the **Processes** tab displays the current RDS host session processes started by the current user or current system process.

Table 9-11. Session Process Details

Option	Description
Process Name	Name of the session process. For example, chrome.exe.
CPU	CPU usage of the process in percent.

Table 9-11. Session Process Details (continued)

Option	Description
Memory	Memory usage of the process in KB.
Disk	Memory disk IOPs. Calculated using the following formula: (Total I/O bytes of current time) - (Total I/O bytes one second before the current time). This calculation can display a value of 0 KB per second if the Task Manager displays a positive value.
Username	User name of the user who owns the process.
Host CPU	CPU usage of the virtual machine to which the session is assigned.
Host Memory	Memory usage of the virtual machine to which the session is assigned.
Processes	Count of processes in the virtual machine
Refresh	The refresh icon refreshes the list of processes.
End Process	Ends a process that is running. **Note** You must have the Help Desk Administrator role to end a process. To end a process, select a process and click the **End Process** button. You cannot end critical processes such as Windows core processes that might be listed in the **Processes** tab. If you end a critical process, Horizon Help Desk Tool displays a message that states it cannot end the system process.

Troubleshoot Linux Desktop Sessions in Horizon Help Desk Tool

In Horizon Help Desk Tool, you can troubleshoot Linux desktop sessions based on a user's connection status.

Prerequisites

- Start Horizon Help Desk Tool.

Procedure

1. On the user card, click the **Sessions** tab.

   A performance card appears that displays CPU and memory usage and includes information about Horizon Client, and the virtual or published desktop.

2 Choose a troubleshooting option.

Option	Action
Send Message	Sends a message to the user on the published desktop or virtual desktop. You can choose the severity of the message to include Warning, Info, or Error. Click **Send Message** and enter the type of severity and the message details, and then click **Submit**.
Restart	Initiates the Restart process on the virtual desktop. This feature is not available for a published desktop session. Click **Restart VDI**.
Disconnect	Disconnect the desktop or application session. Click **More > Disconnect**.
Log Off	Initiates the log off process for a published desktop or virtual desktop. Click **More > Log Off**.
Reset	Initiates a reset of the virtual machine. This feature is not available for a published desktop. Click **More > Reset VM**. **Note** The user can lose unsaved work.

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).

Horizon Agent Fails to Disconnect on an iPad Pro Horizon Client

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

Problem

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

Cause

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

Solution

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

SLES 12 SP1 Desktop Does Not Auto-Refresh

SLES 12 SP1 does not auto-refresh in a multi-monitor mode when you drag a GNOME terminal.

Problem

When you start SLES 12 SP1 in a multi-monitor mode and return to the window mode, the desktop does not refresh automatically when you drag a GNOME terminal.

Cause

The GNOME terminal does not respond to the drag operation.

Solution

1. End the GNOME Shell session.

   ```
   kill –9 <process id of gnome–shell>
   ```

2. Restart the GNOME Shell session again.

SSO Fails to Connect to a PowerOff Agent

Single Sign-On (SSO) does not connect to a PowerOff agent.

Problem

When you log in as a broker and connect to an agent, SSO fails to connect to the PowerOff agent.

Solution

- Manually log in to the desktop, or disconnect and reconnect to the agent again.

Unreachable VM After Creating a Manual Desktop Pool for Linux

The virtual machine state is not responding.

Problem

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

Cause

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

- Verify that the option `machine.id` exists in the virtual machines vmx configuration file.

  If it does not exist, then verify that the virtual machine was added to the desktop pool correctly. Else recreate the desktop pool to let the broker rewrite the option to the vmx configuration file.

- Verify that the VMware Tool or Open VM Tool is installed correctly.

  If the steps to install VMware Tool or Open VM Tool were not performed correctly, the `vmware-rpctool` command might not exist under `PATH` in the Linux virtual machine. You must follow the guide to install VMware Tool or Open VM Tool.

  Run the command after you finish installing.

  ```
  #vmware-rpctool "machine.id.get"
  ```

  The machine.id values are listed from the virtual machines vmx configuration file.

- Verify if the FQDN of the broker can be resolved to the IP Address in the agent Linux virtual machine.