Setting Up Linux Desktops in Horizon

VMware Horizon 2106

vmware®

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

https://docs.vmware.com/

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

Contents

Setting Up Linux Desktops in Horizon 6

1 Features and System Requirements 7
- Features of Horizon Linux Desktops 7
- Overview of Configuration Steps for Setting Up Linux Desktops 14
- System Requirements for Horizon Agent for Linux 15
  - Virtual Machine Settings for 2D Graphics 24
  - Configuring Session Collaboration on Linux Desktops 24

2 Preparing a Linux Virtual Machine for Desktop Deployment 28
- Create a Virtual Machine and Install Linux 28
- Update the GNOME Shell Window List Extension on SLED/SLES Virtual Machines 29
- Prepare a Linux Machine for Remote Desktop Deployment 30
- Install Dependency Packages for Horizon Agent 33
- Upgrade the Operating System of a Linux Virtual Machine 34

3 Setting Up Active Directory Integration and User Authentication Features for Linux Desktops 35
- Integrating Linux Desktops with Active Directory 35
  - Use the OpenLDAP Server Pass-Through Authentication 36
  - Set Up SSSD LDAP Authentication Against the Microsoft Active Directory 36
  - Use the Winbind Domain Join Solution 37
  - Configure PowerBroker Identity Services Open (PBISO) Authentication 37
  - Configure the Samba Offline Domain Join 39
  - Use the Realmd Join Solution for RHEL/CentOS 8.x 40
- Setting Up Single Sign-On 41
- Setting Up Smart Card Redirection 42
  - Configuring Smart Card Redirection for RHEL 8.x Desktops 44
  - Configuring Smart Card Redirection for RHEL 7.x Desktops 49
  - Configuring Smart Card Redirection for Ubuntu Desktops 55
  - Configuring Smart Card Redirection for SLED/SLES Desktops 64
- Setting Up True SSO for Linux Desktops 72
  - Configure True SSO on RHEL/CentOS 8.x Desktops 73
  - Configuring True SSO for RHEL/CentOS 7.x Desktops 75
  - Configuring True SSO for Ubuntu Desktops 79
  - Configuring True SSO for SLED/SLES Desktops 86

4 Setting Up Graphics for Linux Desktops 90

Configure Supported Linux Distributions for vGPU 90
Install the VIB for the NVIDIA GRID vGPU Graphics Card on the ESXi Host 90
Configure a Shared PCI Device for vGPU on the Linux Virtual Machine 92
Install the NVIDIA GRID vGPU Display Driver 93
Verify That the NVIDIA Display Driver Is Installed 93

5 Installing Horizon Agent 95
Install Horizon Agent on a Linux Virtual Machine 95
install_viewagent.sh Command-Line Options 96
Configure the VMwareBlastServer Certificate for Horizon Agent for Linux 98
Upgrading Horizon Agent on a Linux Virtual Machine 99
Upgrade Horizon Agent on a Linux Virtual Machine 99
Uninstall Horizon AgentFrom a Linux Virtual Machine 101

6 Configuration Options for Linux Desktops 102
Setting Options in Configuration Files on a Linux Desktop 102
Using Smart Policies 114
Requirements for Smart Policies 114
Installing Dynamic Environment Manager 115
Configuring Dynamic Environment Manager 115
Horizon Smart Policy Settings 115
Adding Conditions to Horizon Smart Policy Definitions and Environment Variable Definitions 116
Create a Horizon Smart Policy in Dynamic Environment Manager 117
Configure a Digital Watermark Using Environment Variables 119
Using DPI Synchronization with Linux Remote Desktops 122
Example Blast Settings for Linux Desktops 123
Examples of Client Drive Redirection Options for Linux Desktops 125

7 Create and Manage Linux Virtual Desktop Pools 126
Create a Manual Desktop Pool for Linux 126
Manage Linux Desktop Pools 128
Create an Automated Full-Clone Desktop Pool for Linux 129
Create an Instant-Clone Floating Desktop Pool for Linux 131

8 Setting Up Linux Published Desktops and Applications for Multi-Session Use 136
Considerations for Linux Farms, Published Desktops, and Published Applications 137
Create a Manual Farm of Linux Virtual Machines 138
Create an Automated Instant-Clone Farm of Linux Hosts 141
Create a Linux Published Desktop Pool 149
Create a Linux Published Application Pool Manually 152
Create a Linux Published Application Pool from a List of Installed Applications 154

9 Troubleshooting Linux Desktops 157

Using Horizon Help Desk Tool in Horizon Console 157

Start Horizon Help Desk Tool in Horizon Console 158

Troubleshooting Users in Horizon Help Desk Tool 158

Session Details for Horizon Help Desk Tool 161

Session Processes for Horizon Help Desk Tool 164

Troubleshoot Linux Desktop Sessions in Horizon Help Desk Tool 165

Collect Diagnostic Information for a Linux Virtual Machine 166

Horizon Agent Fails to Disconnect on an iPad Pro Horizon Client 167

SSO Fails to Connect to a PowerOff Agent 167

Unreachable VM After Creating a Manual Desktop Pool for Linux 167

Setting Up Linux Desktops in Horizon

The *Setting Up Linux Desktops in Horizon* document provides information about setting up a Linux virtual machine for use as a Linux desktop in VMware Horizon. 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 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 Agent for Linux, users can connect to remote desktops that run the Linux operating system.

This chapter includes the following topics:

- Features of Horizon Linux Desktops
- Overview of Configuration Steps for Setting Up Linux Desktops
- System Requirements for Horizon Agent for Linux

Features of Horizon Linux Desktops

The following list summarizes the key features supported on Horizon Linux desktops.

Note Where applicable, the following entries identify the subset of Linux distributions that support a given feature. For the complete list of Linux distributions supported for Horizon Agent, see System Requirements for Horizon Agent for Linux.

Active Directory Integration

- PowerBroker Identity Services Open (PBISO) Authentication supports offline domain join with Active Directory for instant-cloned desktops running the following Linux distributions.
  - Ubuntu 18.04 and 20.04
  - RHEL 7.x
- Samba supports offline domain join with Active Directory for instant-cloned desktops running the following Linux distributions.
  - Ubuntu 18.04/20.04
  - RHEL Workstation 7.2 or later, and 8.x
  - RHEL Server 7.8, 7.9, 8.3, and 8.4
  - CentOS 7.8, 7.9, 8.3, and 8.4
  - SLED/SLES 12.x/15.x

For more information, see the subtopics under Integrating Linux Desktops with Active Directory.

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 20.04/18.04 with MATE or Gnome Ubuntu desktop environment
- RHEL 7.x with KDE or Gnome desktop environment
- RHEL 8.x with Gnome desktop environment
- SLED 12.x/15.x
- SLES 12.x/15.x

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 of single-session 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 deactivate it during installation.

Digital Watermark

You can create a unique digital watermark as a solution for authenticity, content integrity, and ownership protection of your intellectual property. A watermark shows traceable information that can deter people from potentially stealing your data.

The watermark can be displayed on the following Linux remote sessions:

- Published applications and applications running on a desktop pool
- Virtual desktops and multi-session hosts
- Multiple monitors
- Primary session in a collaborative session

The watermark feature has the following limitations:

- Recorded sessions in Zoom or Webex applications do not include the watermark.
- Screen capture applications and the Print Screen key operated from within the remote desktop do not include the watermark. However, screen capture applications and the Print Screen key operated from the client system do include the watermark.
- If you use an old client version with the latest agent version, the watermark may not display.
- If you use the latest client version with an old agent version, the watermark does not display.
- A shadow session in a collaborative session cannot show the watermark.
- The watermark does not display when the **Search**, **Activities**, or **Show Applications** desktop features are in use
- Watermarks are not supported on the K Desktop Environment (KDE).

You can configure the digital watermark using the following methods:

- Configuration options in the `/etc/vmware/config` file. See Setting Options in Configuration Files on a Linux Desktop.
- Dynamic Environment Manager environment variables. See Configure a Digital Watermark Using Environment Variables. The environment variable settings take priority over the settings in `/etc/vmware/config`.

Display Scaling

With the Display Scaling feature enabled, Linux remote desktops support the client display's scale factor. If the DPI (Dots Per Inch) setting on the remote desktop does not match the DPI setting on the client system, the remote session is displayed using a scale factor that matches the client system.

This feature is turned off by default. You can enable it by setting a configuration option as described in Setting Options in Configuration Files on a Linux Desktop.

DPI Synchronization

The DPI Synchronization feature ensures that the DPI setting in a remote session changes to match the DPI setting of the client system when users connect to a Linux remote desktop or published application. The feature is supported on Linux distributions running Horizon Agent

2012 or later, with one further requirement: RHEL/CentOS 7.x desktops must be running RHEL/CentOS 7.6 or later.

This feature is enabled by default. You can deactivate it by modifying a configuration option as described in Setting Options in Configuration Files on a Linux Desktop.

FIPS 140-2 Mode

The Federal Information Processing Standard (FIPS) 140-2 mode, although not yet validated with the NIST Cryptographic Module Validation Program (CMVP), is available for supported Linux desktops. FIPS mode is not supported on desktops running CentOS.

Horizon 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 version 1.2 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 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 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 and High Efficiency Video Coding (HEVC)

H.264 and HEVC can improve the Blast Extreme performance for a Horizon desktop, especially under a low-bandwidth network. HEVC provides higher image quality than H.264 at the same bandwidth.

If the client system has both H.264 and HEVC turned off, Blast Extreme automatically falls back to JPEG/PNG encoding.

The H.264 and HEVC encoders include both hardware support and software encoder support. The hardware support has the following requirements.

- The vGPU is configured with an NVIDIA graphics card. For specific requirements, see the video codec support matrix on https://developer.nvidia.com.
- The NVIDIA driver 384 series or later is installed in the NVIDIA graphics card.

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

Instant-Clone Floating Desktop Pool

You can create instant-clone floating desktop pools of single-session Linux desktops.

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.

- RHEL/CentOS 7.x
- Ubuntu 18.04 and 20.04

Keyboard Layout and Locale Synchronization

This feature specifies whether to synchronize a client's system locale and current keyboard layout with the 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, Mac, and Linux, 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

When configuring a manual desktop pool of single-session Linux desktops, you can choose from the following options for machine source:

- Managed Virtual Machine - Machine source of the vCenter virtual machine. A managed virtual machine is supported for new and upgrade deployments.
- 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 Ubuntu 18.04 and 20.04.

Multiple Monitors

vGPU desktop supports a maximum resolution of 2560x1600 on four monitors configured in any arrangement.

2D desktop on VMware vSphere® 6.0 or later supports the following maximum resolutions:

- 2560x1600 on three monitors configured in any arrangement
- 2048x1536 on four monitors configured in any arrangement
- 2560x1600 on four monitors configured as follows:
  - Two monitors arranged on the bottom and two monitors arranged on the top
  - Four monitors stacked vertically on top of one another.

    The 2560x1600 resolution is not supported on four monitors arranged side by side.

For Ubuntu 18.04/20.04, 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.

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, 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 VMware Unified Access Gateway, which supports UDP.

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

- If the client connects to a Linux desktop directly, enable 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.

Printer Redirection

With Printer Redirection, users can print from a Linux remote desktop to any local or network printer available on their client computer. Printer Redirection support is enabled by default when you install Horizon Agent. For more information, see install_viewagent.sh Command-Line Options.

Printer Redirection is only supported on Linux desktops running RHEL 7.9, RHEL 8.3, or Ubuntu 20.04.

Published Desktop and Application Pools

You can create published desktop and application pools based on manual or automated instant-clone farms of multi-session Linux host machines. Each published desktop or application can support multiple user sessions at the same time.

Note Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications. vGPU capabilities are not supported for published applications.

For more information, see Chapter 8 Setting Up Linux Published Desktops and Applications for Multi-Session Use.

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 desktops with the following Linux distributions installed.

- Ubuntu 18.04/20.04 with Gnome Ubuntu desktop environment
- RHEL Workstation 7.5 with Gnome Classic or KDE desktop environment
- RHEL Workstation 7.6 or later, or 8.x with Gnome Classic desktop environment
- RHEL Server 7.8 or later, or 8.1 or later with Gnome Classic desktop environment

Single Sign-on

You can configure Active Directory single sign-on (SSO) for Linux desktops.

Smart Card Redirection

Smart card redirection enables users to authenticate into Linux desktops using a smart card reader connected to the local client system. This feature is not supported on desktops running CentOS.

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

You can configure the True SSO feature on Linux desktops.

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

Horizon Agent for Linux supports vGPU graphics capabilities on systems configured with certain NVIDIA graphics cards and running certain operating systems.

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

Limitations of Linux Desktops

Linux desktops have the following limitations:

- Location-based printing and Real-Time Video are not supported.
- The VMware HTML Access file transfer feature is not supported.
- Only the X11 display server protocol is supported. The Wayland protocol is not supported.

Additional limitations apply to multi-session published desktops and published applications. For more information, see Considerations for Linux Farms, Published Desktops, and Published Applications.

Overview of Configuration Steps for Setting Up Linux Desktops

When you configure 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 Linux desktop deployment. See System Requirements for Horizon Agent 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 VMware Horizon 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 Desktops 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 based on the configured Linux virtual machine. See Chapter 7 Create and Manage Linux Virtual Desktop Pools.

3D Graphics - Overview of Configuration Steps

You must complete the NVIDIA GRID vGPU 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 Linux desktop deployment in a VMware Horizon environment. See System Requirements for Horizon Agent 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 VMware Horizon 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 Desktops with Active Directory for more information.

5 Configure 3D capabilities on your ESXi hosts and the Linux virtual machine. For more information, see Configure Supported Linux Distributions for vGPU.

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

7 Create a desktop pool based on the configured Linux virtual machine. See Chapter 7 Create and Manage Linux Virtual Desktop Pools.

System Requirements for Horizon Agent for Linux

To install Horizon Agent for Linux, you must meet certain requirements for the Linux operating system, Linux virtual machine, VMware Horizon system components, 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 20.04 and 18.04	x64
Red Hat Enterprise Linux (RHEL) Workstation 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, and 8.4	x64
Red Hat Enterprise Linux (RHEL) Server 7.8, 7.9, 8.2, 8.3, and 8.4	x64
CentOS 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, and 8.4	x64

Table 1-1. Supported Linux Operating Systems for Horizon Agent (continued)

Linux Distribution	Architecture
SUSE Linux Enterprise Desktop (SLED) 12 SP3, 15 SP1, and 15 SP2	x64
SUSE Linux Enterprise Server (SLES) 12 SP3, 12 SP5, 15 SP1, and 15 SP2	x64

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

Some features are supported on a limited subset of Linux operating systems. For more information, see the section of this document that discusses the specific feature.

Required Platform and Software Versions

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

Table 1-2. Required Platform and VMware Horizon 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 2106
Horizon Client software	■ Horizon Client for Android 2106 ■ Horizon Client for Windows 2106 ■ Horizon Client for Linux 2106 ■ Horizon Client for Mac 2106 ■ Horizon Client for iOS 2106 ■ HTML Access 2106 on Chrome and Firefox ■ Zero clients that support the VMware Blast protocol **Note** Teradici PCoIP 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 used
Horizon Connection Server or Unified Access Gateway 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 Security* document and the Network Ports in VMware Horizon guide.

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

On Ubuntu 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 the 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

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

Table 1-4. Account Name and Account Type

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

Desktop Environment

Horizon Agent for Linux supports multiple desktop environments on different Linux distributions. The following table lists the default desktop environments for each Linux distribution and the additional desktop environments supported by Horizon Agent for Linux.

Table 1-5. Supported Desktop Environments

Linux Distribution	Default Desktop Environment	Desktop Environments Supported by Horizon Agent for Linux
Ubuntu 20.04/18.04	Gnome	Gnome Ubuntu, K Desktop Environment (KDE), MATE
RHEL/CentOS 7.x	Gnome	Gnome, KDE
RHEL/CentOS 8.x	Gnome	Gnome
SLED/SLES	Gnome	Gnome

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 GDM3 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 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 described in the following table 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 7.x	KDE	`# yum groupinstall "KDE Plasma Workspaces"`
Ubuntu 20.04/18.04	KDE	`# apt install plasma-desktop`
Ubuntu 20.04/18.04	MATE 1.225	`# apt install ubuntu-mate-desktop`

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 20.04/18.04 and RHEL/CentOS 7.x, use the information in the following table 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 8.x, 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:

```
sudo yum install wireshark
```

For Ubuntu:

```
sudo apt install tshark
```

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 listed in the following table.

For example, if you unpack the installation file `VMware-horizonagent-linux-x86_64-`*`YYMM-y.y.y-xxxxxxx`*`.tar.gz` under the `/install_tmp/` directory, the *`full-path_to_patch-file`* is `/install_tmp/VMware-horizonagent-linux-x86_64-`*`YYMM-y.y.y-xxxxxxx`*`/resources/vhci/patch/vhci.patch` and the `patch` command to use is

```
# patch –p1 < /install_tmp/VMware-horizonagent-linux-x86_64-YYMM-y.y.y-xxxxxxxi/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 the USB VHCI Driver

Linux Distribution	Steps to Compile and Install USB VHCI Driver

Ubuntu 20.04/18.04	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`

RHEL/CentOS 7.x
RHEL/CentOS 8.x
	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.x) 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/" -addext extendedKeyUsage=1.3.6.1.5.5.7.3.3`
	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.x SLED/SLES 15.x	1 Find the version of the current kernel package. `# rpm -qa | grep kernel-default-$(echo $(uname -r) | cut -d '-' -f 1,2)`

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

Linux Distribution	Steps to Compile and Install USB VHCI Driver

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
```

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
```

4 (SLED/SLES 15.x) 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/" -addext
extendedKeyUsage=1.3.6.1.5.5.7.3.3
```

b Find the path to the signing file for the USB driver.

```
find / -name sign-file
```

This command returns the paths to all the signing files located on the system. The signing file path for the USB driver resembles the following example.

```
/usr/src/linux-5.3.18-24.9-obj/x86_64/default/scripts/
```

c Sign the USB driver. In the following commands, *<sign-file-path>* is the path to the signing file that you found earlier in step 4b.

```
# sudo /<sign-file-path>/sign-file sha256 ./MOK.priv ./MOK.der /lib/modules/$
(uname -r)/kernel/drivers/usb/host/usb-vhci-iocifc.ko
# sudo /<sign-file-path>/src/kernels/$(uname -r)/scripts/sign-file sha256 ./
MOK.priv ./MOK.der /lib/modules/$(uname -r)/kernel/drivers/usb/host/usb-vhci-
hcd.ko
```

d 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.

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

Linux Distribution	Steps to Compile and Install USB VHCI Driver
	e To set up UEFI Secure Boot in the vSphere console, reboot the system. For more information, see https://sourceware.org/systemtap/wiki/SecureBoot.

In addition, follow these 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 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 Linux virtual machines for a VMware Horizon deployment, you must change the vCPU and virtual memory settings for performance requirements.

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 VMware Horizon deployment must meet certain requirements.

Table 1-9. System Requirements for Session Collaboration

Component	Requirements
Linux remote desktops	The Session Collaboration feature is supported on remote desktops running the following Linux distributions and desktop environments: ■ Ubuntu 20.04/18.04 with Gnome desktop environment ■ RHEL 7.5 with Gnome Classic or KDE desktop environment ■ RHEL Workstation 7.6 or later, or 8.x with Gnome Classic desktop environment ■ RHEL Server 7.8 or later, or 8.1 or later with Gnome Classic desktop environment
Connection Server	The Connection Server instance uses an Enterprise license.
Display protocol	VMware Blast

Note RHEL 8.x desktops require additional system configuration to support Session Collaboration. See Configure a RHEL 8.x 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

The following general limitations apply to the Session Collaboration feature:

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

The following limitations apply to collaboration sessions on a RHEL 7.5 desktop with KDE desktop environment:

- To display the Session Collaboration menu, users must right-click the Session Collaboration icon in the system tray. Left-clicking the icon has no effect.
- Users might see the **Email** button, which is used to send invitation emails to collaborators but is initially inactive. To make the button active, a user must first configure a default email application for the KDE desktop environment.

Note Use the following remedies to troubleshoot issues related to Session Collaboration:

- If the Session Collaboration icon fails to appear in the system tray after a user logs in for the first time to the remote desktop, instruct the user to disconnect from and reconnect to the desktop. The Session Collaboration icon usually appears after reconnection to the desktop.
- 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.x Desktop for Session Collaboration

To use the Session Collaboration feature on a RHEL 8.x desktop, you must first download and install the required GNOME Shell extension.

Procedure

1. Download the required GNOME shell extension to the RHEL 8.x system from https://extensions.gnome.org/extension/615/appindicator-support/.
   - For RHEL 8.0, select **3.28** for the shell version and **26** for the extension version.
   - For RHEL 8.1 and later, select **3.32** for the shell version and **29** for the extension version.
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
- Update the GNOME Shell Window List Extension on SLED/SLES Virtual Machines
- Prepare a Linux Machine for Remote Desktop Deployment
- Install Dependency Packages for Horizon Agent
- Upgrade the Operating System of a Linux Virtual Machine

Create a Virtual Machine and Install Linux

You create a virtual machine in vCenter Server for each remote desktop that is deployed in a VMware Horizon environment. 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 Agent for Linux.
- Familiarize yourself with the steps for creating virtual machines in vCenter Server and installing guest operating systems. For more information see the *Setting Up Virtual Desktops in Horizon* 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 Agent for Linux.

Procedure

1 In vSphere Web Client or vSphere Client, create a 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, refer to the following guidelines.

   - If you are preparing the virtual machine for deployment as a single-session virtual desktop pool, 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.

   - If you are preparing the virtual machine to serve as a multi-session host for a published desktop or application pool, specify at least 8 vCPUs and 40 GB of vMemory.

     Important A minimum of 8 vCPUs and 40 GB of vMemory is required to support up to 50 user sessions per published desktop or published application.

3 Power on the virtual machine and install the required Linux distribution. Note the following considerations for instant-clone desktop pools and multi-session hosts.

   Horizon Agent for Linux only supports instant-clone desktop pools created from virtual machines running the following operating systems:

   - Ubuntu 18.04/20.04
   - RHEL Workstation 7.2 or later, and 8.x
   - RHEL Server 7.8, 7.9, 8.3, and 8.4
   - CentOS 7.8, 7.9, 8.3, and 8.4
   - SLED/SLES 12.x/15.x

   Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications.

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

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

5 Ensure that the system hostname is resolvable to 127.0.0.1.

Update the GNOME Shell Window List Extension on SLED/SLES Virtual Machines

Due to a known issue, the default GNOME Shell window list extension on SLED/SLES virtual machines causes the window taskbar to appear in the middle of the screen when the remote desktop is started in full-screen mode. This topic describes how to remedy the display issue.

A client user can correct the location of the taskbar by exiting and reentering full-screen mode. To fix this issue for all provisioned desktops, update the GNOME Shell window list extension to the current version on the SLED/SLES virtual machine.

Prerequisites

Ensure that Firefox version 56 or later is installed on the SLED/SLES machine. The updated GNOME Shell window list extension requires this browser version.

Procedure

1. From Firefox on the SLED/SLES machine, go to the GNOME Shell extensions site at https://extensions.gnome.org/local/.
2. Click **Click here to install browser extension**.
3. Click **Continue to installation**.
4. Click **Add**.
5. Select the **Allow this extension to run in Private Windows** check box and click **Okay, Got it**.
6. Refresh the web page and locate the **Windows List** entry in the extensions list. Click the green update button next to that entry.

   The updated windows list extension is installed on the machine. To confirm the installation, refresh the web page and verify that the green update button no longer appears next to the **Windows List** entry.

Results

The window taskbar appears correctly at the bottom of the screen on remote desktops provisioned from this SLED/SLES machine.

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 VMware Horizon deployment.

To prepare a Linux machine for a VMware Horizon deployment, 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).

If you are preparing the Linux machine for use as a multi-session host for a published desktop or application pool, you must perform several additional preparation steps.

Prerequisites

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

- If you are preparing the Linux machine for use as a multi-session host, verify that one of the following required distributions is installed on the machine:
  - RHEL Workstation 8.x/7.x
  - Ubuntu 20.04/18.04
- 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.

If you are preparing the Linux machine for deployment as an automated full-clone or instant-clone desktop pool or for inclusion in an automated instant-clone farm, you must also do the following:

- Verify that the virtual switch that the instant-clone VMs connect to has enough ports to support the expected number of VMs. Each network card on a VM requires one port.
- To support instant-clone desktop pools or farms, verify that you have added an instant-clone domain administrator in Horizon Console.

Procedure

1 On an Ubuntu machine, manually install VMware Tools by using the following command:

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

Note VMware Tools is pre-installed on RHEL/CentOS and SLED/SLES machines.

Note If you upgrade the Linux kernel after installing VMware Tools, VMware Tools might stop running. To resolve the problem, refer to VMware KB article 2050592.

2 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.

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.

3 To prepare the virtual machine for use in an automated instant-clone farm, in vSphere Client, disable the vApp Options setting on the virtual machine.

4 (RHEL and CentOS only) Verify that `virbr0` is disabled.

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

5 Ensure that the Horizon Connection Server instances in the pod can be resolved through DNS.

6 Configure the Linux machine to run in graphical mode by default.

For example, the following command configures a CentOS machine to run in graphical mode.

```
systemctl set-default graphical.target
```

7 (Ubuntu only) If the machine is configured to authenticate with an OpenLDAP server, set the FQDN 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 FQDN. For example: `127.0.0.1 hostname.domainname hostname`.

c Exit and save the file.

8 (SLED/SLES only) Disable **Change Hostname via DHCP**. Then set the static hostname and 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**.

9 To prepare a virtual machine for use as a multi-session host in a farm, install the required software packages.

- For RHEL Workstation 8.x/7.x:

```
sudo yum install http://mirror.centos.org/centos/7/os/x86_64/Packages/
cpptest-1.1.1-9.el7.x86_64.rpm
sudo yum install https://rpmfind.net/linux/centos/7.8.2003/os/x86_64/Packages/
uriparser-0.7.5-10.el7.x86_64.rpm
```

- For Ubuntu 20.04/18.04:

```
apt-get install liburiparser1
```

10 Install Horizon Agent on the machine, as described in Install Horizon Agent on a Linux Virtual Machine. Ensure that you include the appropriate parameters in the installation script to install or enable required features, as described in install_viewagent.sh Command-Line Options. For example:

   - To prepare the virtual machine for inclusion in an automated instant-clone farm, use the following installation script:

     ```
     sudo ./install_viewagent.sh --multiple-session
     ```

   - To prepare the virtual machine for inclusion in a manual farm, use the following installation script:

     ```
     sudo ./install_viewagent.sh --multiple-session -M no
     ```

11 To prepare the golden-image virtual machine for an instant-clone floating desktop pool or automated instant-clone farm, use vSphere Client or vSphere Web Client to take a snapshot of the virtual machine in its powered-down state. This snapshot is used as the baseline configuration for the first set of instant-clone machines that are anchored to the virtual machine.

    For more information, see "Take a Snapshot in the VMware Host Client" in *vSphere Single Host Management - VMware Host Client*, available from VMware vSphere Documentation.

    Important Before you take a snapshot, completely shut down the golden-image virtual machine by using the shutdown or power-off command in the Linux operating system.

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

- 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.x/8.x	`yum install libappindicator-gtk3`
	Note If the `yum` command does not work, you can try the `dnf` package manager instead.
	`dnf install libappindicator-gtk3`

SLES 12.x	Installation of `python-gobject2` is required for SLES 12.x desktops 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 18.04	`apt-get install python python-dbus python-gobject`

Upgrade the Operating System of a Linux Virtual Machine

This topic explains how to upgrade the operating system of a Linux virtual machine (VM) that has Horizon Agent installed on it. Always follow the order of steps described in this topic when you want to upgrade your Linux VM to a new operating system version.

Prerequisites

Before starting the upgrade procedure, take a snapshot of the current Linux VM.

Procedure

1. Uninstall Horizon Agent from the VM.

   See Uninstall Horizon AgentFrom a Linux Virtual Machine.

2. Upgrade the operating system of the VM, following the upgrade steps for your Linux distribution.

   You can perform the upgrade using the graphical installation interface or installation commands for your Linux distribution.

3. Reinstall Horizon Agent on the VM.

   See Install Horizon Agent on a Linux Virtual Machine.

Setting Up Active Directory Integration and User Authentication Features for Linux Desktops

3

Horizon Agent uses the existing Microsoft Active Directory (AD) infrastructure for user authentication and management. You can integrate your Linux virtual machines with Active Directory so that users can log in to a Linux desktop using their Active Directory user account. You can also configure features for user authentication, such as single sign-on (SSO), smart card redirection, and True SSO.

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 Desktops with Active Directory
- Setting Up Single Sign-On
- Setting Up Smart Card Redirection
- Setting Up True SSO for Linux Desktops

Integrating Linux Desktops with Active Directory

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

The following solutions are known to work for a Linux virtual machine running Horizon Agent for Linux.

- OpenLDAP Server Pass-through Authentication
- System Security Services Daemon (SSSD) LDAP Authentication against the Microsoft Active Directory
- PowerBroker Identity Services Open (PBISO) Authentication supports offline domain join with Active Directory for instant-cloned desktops running the following Linux distributions.
  - Ubuntu 18.04 and 20.04

  - RHEL 7.x
- Samba supports offline domain join with Active Directory for instant-cloned desktops running the following Linux distributions.
  - Ubuntu 18.04/20.04
  - RHEL Workstation 7.2 or later, and 8.x
  - RHEL Server 7.8, 7.9, 8.3, and 8.4
  - CentOS 7.8, 7.9, 8.3, and 8.4
  - SLED/SLES 12.x/15.x

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

Note For ease of deployment, if available, choose 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 virtual machines 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 virtual machine.

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

Note To perform an instant-clone offline domain join, you must use one of the supported authentication methods: PowerBroker Identity Services Open (PBISO) authentication or Samba offline domain join. The SSSD LDAP authentication solution is not supported.

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 virtual machine.

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 virtual machine.
2. Join the Linux desktop to Microsoft Active Directory (AD).

What to do next

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

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

To run the domain rejoin command on a cloned virtual machine for the Winbind solution, include the command to a shell script and set the script path to the `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 virtual machine to Active Directory (AD) using PBISO.

Procedure

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

   Note For Ubuntu 20.04, download PBISO 9.1.0 or later.

2. Install PBISO on your Linux virtual machine.

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

3. Install Horizon Agent for Linux.
4. Use PBISO to join the Linux virtual machine 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. Restart the Linux virtual machine 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 virtual machine, modify the `resolv.conf` file for your Linux system. Use the following example, for an Ubuntu system, as a guide for adding the necessary lines in the `resolv.conf` file.

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

Configure the Samba Offline Domain Join

To support SSO on an instant-cloned Linux virtual machine (VM) in a VMware Horizon desktop environment, configure Samba on the golden-image Linux VM.

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

Procedure

1. On your golden-image 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 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]
```

```
EXAMPLE.COM = {
kdc = 10.111.222.33
}

[domain_realm]
example.com = EXAMPLE.COM
.example.com = 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, configure the following options in the `/etc/vmware/viewagent-custom.conf` file. Replace *YOURDOMAIN* with the NetBIOS name of your domain.

```
OfflineJoinDomain=samba

NetbiosDomain=YOURDOMAIN
```

9 Restart the golden-image Linux VM and log back in.

Use the Realmd Join Solution for RHEL/CentOS 8.x

To ensure the operation of features such as single sign-on for a RHEL/CentOS 8.x desktop, use the realmd solution to join the RHEL/CentOS 8.x virtual machine to your Active Directory (AD) domain.

Procedure

1 Configure a fully qualified host name for the RHEL/CentOS 8.x virtual machine (VM).

For example, if **`rhel8`** is the unqualified host name of the VM 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, restart the VM 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 VM.

Note Horizon Agent only supports the X11 display server protocol for RHEL/CentOS 8.x desktops.

Setting Up Single Sign-On

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

The VMware 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 the Linux virtual machine with Active Directory (AD). VMware Horizon SSO is known to work with the OpenLDAP and Winbind solutions that integrate Linux virtual machines 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 Agent 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 virtual machine 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

When smart card redirection is enabled on a Linux desktop, a user can authenticate into the desktop using a smart card reader connected to the local client system. To set up smart card redirection, you must perform some configuration steps.

Overview of Smart Card Redirection

Smart card redirection is supported on desktops based on virtual machines running the following Linux distributions:

- RHEL 8.x/7.x
- Ubuntu 20.04/18.04
- SLED 12.x/15.x
- SLES 12.x/15.x

Note RHEL 8.x desktops do not support smart card redirection and Active Directory single sign-on (SSO) at the same time. If you set up smart card redirection on a RHEL 8.x desktop, Active Directory SSO does not work.

When you install Horizon Agent, you must 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 the use of PIV cards to authenticate into 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.

Note The Smartcard SSO feature is not supported on Linux desktops.

Configuring Smart Card Redirection

To configure smart card redirection, perform the following tasks.

1 Set up the smart card by following the instructions from the smart card vendor.

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

3 Configure smart card redirection on the base virtual machine, following the procedure for your Linux distribution.

Configuring Smart Card Redirection for RHEL 8.x Desktops

To set up smart card direction for RHEL 8.x desktops, first integrate the RHEL 8.x virtual machine with an Active Directory domain. Then install the necessary libraries and root CA certificate before installing Horizon Agent.

Integrate a RHEL 8.x Virtual Machine with Active Directory for Smart Card Redirection

Use the following procedure to integrate a RHEL 8.x virtual machine (VM) with an Active Directory (AD) domain for smart card redirection.

Note RHEL 8.x desktops do not support smart card redirection and Active Directory single sign-on (SSO) at the same time. If you set up smart card redirection on a RHEL 8.x desktop, Active Directory SSO does not work.

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.domain.com	Fully qualified host name of your RHEL 8.x VM
rhel8sc	Unqualified host name of your RHEL 8.x VM
domain.com	DNS name of your AD domain
DOMAIN.COM	DNS name of your AD domain, in all capital letters
DOMAIN	DNS name of the workgroup or NT domain that includes your Samba server, in all capital letters
dnsserver.domain.com	Host name of your AD server

Procedure

1. On the RHEL 8.x VM, 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.domain.com rhel8sc localhost localhost.localdomain localhost4
localhost4.localdomain4
::1              localhost localhost.localdomain localhost6 localhost6.localdomain6

dns_IP_ADDRESS   dnsserver.domain.com
```

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

```
# Generated by NetworkManager
search domain.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 /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 = domain.com
services = nss, pam, pac

[domain/DOMAIN.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 = DOMAIN.COM
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 DOMAIN.COM = {
     kdc = dnsserver.domain.com
     admin_server = dnsserver.domain.com
     default_domain = dnsserver.domain.com
     pkinit_anchors = FILE:/etc/pki/nssdb/certificate.pem
     pkinit_cert_match = <KU>digitalSignature
     pkinit_kdc_hostname = dnsserver.domain.com
 }

[domain_realm]
 .domain.com = DOMAIN.COM
 domain.com = DOMAIN.COM
```

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

```
[global]
        workgroup = DOMAIN
        security = ads
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
        password server = dnsserver.domain.com
        realm = DOMAIN.COM
        idmap config * : range = 16777216-33554431
```

```
        template homedir =/home/DOMAIN/%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 -- DOMAIN
Joined 'RHEL8SC' to dns domain 'domain.com'
```

14 Verify that the RHEL 8.x VM is successfully joined to the AD domain.

```
# net ads testjoin

Join is OK
```

What to do next

Configure Smart Card Redirection on a RHEL 8.x Virtual Machine

Configure Smart Card Redirection on a RHEL 8.x Virtual Machine

To configure smart card redirection on a RHEL 8.x virtual machine (VM), 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.x Virtual Machine 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 pcsc-1.8.8 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 the Horizon Agent package, with smart card redirection enabled.

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

10 Restart the virtual machine and log back in.

Configuring Smart Card Redirection for RHEL 7.x Desktops

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

Integrate a RHEL 7.x Virtual Machine with Active Directory for Smart Card Redirection

To support smart card redirection on RHEL 7.x desktops, integrate the base virtual machine (VM) with your Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate a RHEL 7.x VM with your 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

Placeholder Value	Description
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 7.2 or later.

Procedure

1 On the RHEL 7.x VM, 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 the RHEL 7.x VM.

```
# hostname –f
```

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

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

5 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
```

6 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
```

7 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**.

   The RHEL 7.x VM is joined to the AD domain.

8 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
```

9 Restart the Winbind service.

```
# sudo service winbind restart
```

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

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

11 Restart the RHEL 7.x VM and log back in.

What to do next

Set Up Smart Card Redirection on a RHEL 7.x Virtual Machine

Set Up Smart Card Redirection on a RHEL 7.x Virtual Machine

To configure smart card redirection on a RHEL 7.x virtual machine (VM), 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 on a RHEL 7.x VM.

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 7.2 or later.

Note If you use the vSphere console to log in to a RHEL 7.x. VM 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 Virtual Machine 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/
```

Note To get the `libcmP11.so` driver, refer to your smart card provider.

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. Ensure that the contents appear in a single line without a carriage return.

```
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 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 pcsc-1.8.8 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
```

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

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

12 Restart the RHEL 7.x VM and log back in.

Configuring Smart Card Redirection for Ubuntu Desktops

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

Integrate an Ubuntu Virtual Machine with Active Directory for Smart Card Redirection

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

Use the following procedure to integrate an Ubuntu VM 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 VM administrator

Procedure

1 On the Ubuntu VM, define the host name of the VM 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**.

What to do next

Set Up Smart Card Redirection on an Ubuntu Virtual Machine

Set Up Smart Card Redirection on an Ubuntu Virtual Machine

To configure smart card redirection on an Ubuntu virtual machine (VM), 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 VM administrator

Prerequisites

Integrate an Ubuntu Virtual Machine with Active Directory for Smart Card Redirection

Procedure

1 Install the required libraries on the Ubuntu VM.

```
# 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 the Ubuntu VM. 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 commands similar to the following example.

These example commands show how to add `libcmP11.so`, the driver file for the Gemalto PIV 2.0 card, to the `nssdb` directory. In place of `libcmP11.so`, you can substitute the driver file for your smart card.

```
# 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.

- For Ubuntu 18.04 and 20.04, run the following command sequence.

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

- For Ubuntu 20.04.1, run the following command sequence.

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

```
Common name -> Login ID
```

6 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
```

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 pcscd service to start automatically after the VM restarts.

Note If the pcscd service does not start after the VM restarts, the first login through pam_pkcs11 fails.

a Edit the `/lib/systemd/system/pcscd.service` file by adding the line `WantedBy=multi-user.target` to the [Install] section.

Verify that the edited file resembles the following example.

```
[Unit]
Description=PC/SC Smart Card Daemon
Requires=pcscd.socket

[Service]
ExecStart=/usr/sbin/pcscd --foreground --auto-exit
ExecReload=/usr/sbin/pcscd --hotplug

[Install]
WantedBy=multi-user.target
Also=pcscd.socket
```

b Enable the pcscd service.

```
# systemctl enable pcscd.service
```

9 Update the PC/SC Lite library to version 1.8.8, using the following sequence of commands.

```
# apt-get install -y git autoconf automake libtool flex libudev-dev
# git clone https://salsa.debian.org/rousseau/PCSC.git
# cd PCSC/
# git checkout -b pcsc-1.8.8 1.8.8
# ./bootstrap
# ./configure --prefix=/usr --sysconfdir=/etc --libdir=/lib/x86_64-linux-gnu/
   CFLAGS="-g -O2 -fstack-protector-strong -Wformat -Werror=format-security"
   LIBS="-ldl" LDFLAGS="-Wl,-Bsymbolic-functions -Wl,-z,relro"
   CPPFLAGS="-Wdate-time -D_FORTIFY_SOURCE=2"
# make
# make install
```

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

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

11 Restart the Ubuntu VM and log back in.

Configuring Smart Card Redirection for SLED/SLES Desktops

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

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

To support smart card redirection on SLED/SLES desktops, integrate the base virtual machine (VM) with an Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate a SLED/SLES VM 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 VM administrator

Prerequisites

Verify that the SLED/SLES VM meets the system requirements described in Setting Up Smart Card Redirection.

Procedure

1 Configure the network settings for the SLED/SLES VM.

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

   b Configure the DNS server IP address, and disable **Automatic DNS**. For a SLES VM, 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 Update the krb5 library, as shown in the following example.

```
# zypper up krb5
```

4 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
```

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

```
# net ads join –U AdminUser
```

6 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 desktops 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
```

7 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

Proceed to Set Up Smart Card Redirection on a SLED/SLES Virtual Machine.

Set Up Smart Card Redirection on a SLED/SLES Virtual Machine

To configure smart card redirection on a SLED/SLES virtual machine (VM), 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 VM administrator

Prerequisites

Complete the steps described in Integrate a SLED/SLES Virtual Machine 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
```

You may need to enable extensions like PackageHub to install all the above packages

b To use the installed packages, enable extensions like PackageHub and install the PC/SC tools. For example, you can run the following commands for SLED/SLES 12 SP3.

```
# 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 the system. 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 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 To configure the pcscd service to start automatically after the VM restarts, edit the appropriate file for your SLED/SLES version.

- (SLED/SLES 12.x) Add the line rcpcscd start to /etc/init.d/after.local so that the file resembles the following example.

```
#! /bin/sh
#
# Copyright (c) 2010 SuSE LINUX Products GmbH, Germany. All rights reserved.
#
# Author: Werner Fink, 2010
#
# /etc/init.d/after.local
#
# script with local commands to be executed from init after all scripts
# of a runlevel have been executed.
#
# Here you should add things, that should happen directly after
# runlevel has been reached.
#
rcpcscd start
```

- (SLED/SLES 15.x) Add the line WantedBy=multi-user.target to /usr/lib/systemd/system/pcscd.service so that the file resembles the following example.

```
[Unit]
Description=PC/SC Smart Card Daemon
Requires=pcscd.socket
```

```
[Service]
ExecStart=/usr/sbin/pcscd --foreground --auto-exit
ExecReload=/usr/sbin/pcscd --hotplug

[Install]
Also=pcscd.socket
WantedBy=multi-user.target
```

After editing the `pcscd.service` file, run the following command.

```
systemctl enable pcscd
```

Note If the pcscd service does not start after the VM restarts, the first login via pam_pkcs11 fails.

7 Disable the firewall.

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

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

8 Update the PC/SC Lite library to version 1.8.8.

a Enable the necessary extensions and modules for the installation of dependent packages.

- (SLED/SLES 12.x) Run the following command, replacing *<SUSE-version>* with the version number of your distribution, for example `12.5` for SLED/SLES 12 SP5.

```
# SUSEConnect -p sle-sdk/<SUSE-version>/x86_64
```

- (SLED/SLES 15.x) Run the following command, replacing *<SUSE-version>* with the version number of your distribution, for example `15.3` for SLED/SLES 15 SP3.

```
# SUSEConnect -p PackageHub/<SUSE-version>/x86_64
```

b Update the PC/SC Lite library.

```
# zypper in git autoconf automake libtool flex libudev-devel gcc
# git clone https://salsa.debian.org/rousseau/PCSC.git
# cd PCSC/
# git checkout -b pcsc-1.8.8 1.8.8
# ./bootstrap
# ./configure -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/
lib64
  --mandir=/usr/share/man --infodir=/usr/share/info --disable-static --enable-usbdropdir=/usr/
lib64/pcsc/drivers
# make
# make install
```

9 (SLED/SLES 15.x) To ensure that the smart card greeter functions properly, modify the `org.gnome.Shell.desktop` file on the VM.

a Open the `/usr/share/applications/org.gnome.Shell.desktop` file.

b In the file, find and replace `Exec=/usr/bin/gnome-shell`with the following line.

```
Exec=sh -c "DISPLAY=:${DISPLAY##*:} exec /usr/bin/gnome-shell"
```

c Save and close the file.

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

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

11 Restart the VM 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 remote desktop after they first log in to VMware Workspace ONE. Users can log in to VMware Workspace ONE using

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

Overview of True SSO

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 remote desktop, users can select to use either the native Horizon Client or HTML Access.

System Requirements for True SSO

True SSO is supported on desktops running the following Linux distributions:

- RHEL/CentOS 8.x
- RHEL/CentOS 7.x
- Ubuntu 20.04/18.04
- SLED 12.x/15.x
- SLES 12.x/15.x

Configuring True SSO

To set up True SSO for Linux desktops, perform the following tasks.

1. Set up and configure True SSO in your VMware Horizonenvironment. For more information, see the *Horizon Administration* document.
2. Integrate the base virtual machine with an AD domain, following the procedure for your Linux distribution.
3. Configure True SSO on the base virtual machine, following the procedure for your Linux distribution.

Configure True SSO on RHEL/CentOS 8.x Desktops

To support True SSO on RHEL/CentOS 8.x desktops, you must first integrate the base virtual machine (VM) 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.x 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

- Configure True SSO for Workspace ONE Access and Horizon Connection Server.
- Verify that the Active Directory (AD) server is resolvable by DNS on the RHEL/CentOS 8.x base VM.
- Configure the host name of the VM.
- Configure the Network Time Protocol (NTP) on the VM.

Procedure

1 On the RHEL/CentOS 8.x VM, 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 Modify the `/etc/krb5.conf` configuration file by setting the mode equal to `644`.

Note If you do not modify `/etc/krb5.conf` as specified, the True SSO feature might not work.

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

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

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

```
NetbiosDomain = MYDOMAIN
```

9 Restart the VM and log back in.

Configuring True SSO for RHEL/CentOS 7.x Desktops

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

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

To support True SSO on instant-cloned RHEL/CentOS 7.x desktops, you must configure Samba on the base virtual machine (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 VM to Active Directory.

Prerequisites

Verify that:

- 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 RHEL/CentOS 7.x VM.
- The Network Time Protocol (NTP) is configured on the VM.

Procedure

1. Verify that the RHEL/CentOS VM can discover the AD server. Use the following example, where *ADdomain.example.com* is 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 on 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. Restart the VM and log back in.

What to do next

Configure True SSO on a RHEL/CentOS 7.x Virtual Machine

Configure True SSO on a RHEL/CentOS 7.x Virtual Machine

To enable the True SSO feature on a RHEL/CentOS 7.x virtual machine (VM), 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 a RHEL 7.x or CentOS 7.x VM.

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 vWorkspace ONE Access and Horizon Connection Server.
- Complete the steps described in Integrate a RHEL/CentOS 7.x Virtual Machine 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 VM. 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 trusted CA certificates on the RHEL/CentOS 7.x VM 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
```

Note You must also set the mode equal to `644` in `/etc/krb5.conf`. Otherwise, the True SSO feature might not work.

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

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

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 Restart the VM and log back in.

Configuring True SSO for Ubuntu Desktops

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

Integrate an Ubuntu Virtual Machine with Active Directory for True SSO

To support True SSO on Ubuntu desktops, integrate the base virtual machine (VM) with an Active Directory (AD) domain using the Samba and Winbind solutions.

Use the following procedure to integrate an Ubuntu VM 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 VM
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

Verify that:

- The AD server is resolvable by DNS on the Ubuntu VM.
- The Network Time Protocol (NTP) is configured on the Ubuntu VM.

Procedure

1 On the Ubuntu VM, 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 Ubuntu VM.

   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 Restart the VM 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 the Ubuntu VM 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 Restart the VM 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 virtual machine (VM), install the libraries on which the True SSO feature depends, the root CA certificate to support trusted authentication, and Horizon Agent. To complete the authentication setup, you must also edit some configuration files.

Use the following procedure to enable True SSO on an Ubuntu VM.

Prerequisites

- Configure True SSO for Workspace ONE Access and Horizon Connection Server.
- Complete the steps described in Integrate an Ubuntu Virtual Machine with Active Directory for True SSO .
- Obtain a root Certificate Authority certificate and save it to `/tmp/certificate.cer` on the Ubuntu VM. See How to Export Root Certification Authority Certificate.

Procedure

1 On the Ubuntu VM, 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 Make an `/etc/pki/nssdb` directory to contain the system database.

   ```
   sudo mkdir -p /etc/pki/nssdb
   ```

   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 Make an `/etc/pam_pkcs11/cacerts` directory and copy the root CA certificate there.

   ```
   mkdir -p /etc/pam_pkcs11/cacerts
   cp /tmp/certificate.pem /etc/pam_pkcs11/cacerts
   ```

   e 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
   ```

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.

Note For Ubuntu 20.04, append ms to the end of the use_mappers line.

```
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;
}

mapper ms {
  debug = false;
  module = internal;
  # module = /usr/$LIB/pam_pkcs11/ms_mapper.so;
  ignorecase = false;
  # ignore domain name
  ignoredomain = true;
  domain = "DOMAIN.COM"; #<== Replace "DOMAIN.COM" with your organization's domain name
}

use_mappers = digest, cn, pwent, uid, mail, subject, null, ms;  #<== For Ubuntu 20.04, append
"ms" at end of use_mappers
```

7 Modify the auth parameters in the PAM configuration file.

a Open the PAM configuration file at /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 Modify the /etc/krb5.conf configuration file by setting the mode equal to 644.

Note If you do not modify /etc/krb5.conf as specified, the True SSO feature might not work.

9 Restart the VM and log back in.

Configuring True SSO for SLED/SLES Desktops

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

Integrate a SLED/SLES Virtual Machine with Active Directory for True SSO

To support True SSO on SLED/SLES desktops, first integrate the base virtual machine (VM) with an Active Directory (AD) domain using the Samba and Winbind solutions.

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

Prerequisites

Verify the following:

- The True SSO feature has been configured for Workspace ONE Access and Horizon Connection Server.
- The SLED/SLES base VM meets the system requirements described in Setting Up True SSO for Linux Desktops.
- The Active Directory server is resolvable by DNS on the VM.
- The Network Time Protocol (NTP) is configured on the VM.

Procedure

1. On the SLED/SLES VM, 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 the system 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 Restart the VM and log back in.

8 Test and verify the AD 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

Proceed to Configure True SSO on a SLED/SLES Virtual Machine.

Configure True SSO on a SLED/SLES Virtual Machine

To enable the True SSO feature on a SLED/SLES virtual machine (VM), 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 a SLED or SLES VM.

Prerequisites

- Configure True SSO for Workspace ONE Access and Horizon Connection Server.
- Complete the steps described in Integrate a SLED/SLES Virtual Machine with Active Directory for True SSO .
- Obtain a root Certificate Authority certificate and save it to `/tmp/certificate.cer` on your SLED/SLES desktop. See How to Export Root Certification Authority Certificate.

Procedure

1 For SLED 15.x or SLES 12.x/15.x, install the necessary packages by running the following command.

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

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

a Download the corresponding SLES .iso file to the local disk of your SLED VM (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 system because the necessary `krb5-plugin-preauth-pkinit` package is available only for SLES systems.

b Mount the SLES .iso file on your SLED system, 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
```

Note You must also set the mode equal to `644` in `/etc/krb5.conf`. Otherwise, the True SSO feature might not work.

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
ADS-HOSTNAME	Host name of your AD server (in all capital letters)

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

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

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

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

Note For SLED/SLES 15.x, always use the long name of the NetBIOS domain, for example `LXD.VDI`. If you use the short name, such as `LXD`, the True SSO feature does not work.

7 Restart the VM 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

For vGPU, complete the graphics setup in the base VM. Clone the VMs. The graphics settings work for cloned VMs and no further settings are required.

This chapter includes the following topics:

- Configure Supported Linux Distributions for vGPU

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 capabilities, see https://docs.nvidia.com/grid/latest/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 capabilities, see https://docs.nvidia.com/grid/latest/product-support-matrix/index.html.

Procedure

1 Download the VIB for your NVIDIA GRID vGPU graphics card from the NVIDIA website.

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 capabilities, see https://docs.nvidia.com/grid/latest/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
```

Results

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 a Linux virtual machine so that Horizon Connection Server can communicate with and manage the desktops based on that virtual machine.

This chapter includes the following topics:

- Install Horizon Agent on a Linux Virtual Machine
- Configure the VMwareBlastServer Certificate for Horizon Agent for Linux
- Upgrading Horizon Agent on a Linux Virtual Machine
- Uninstall Horizon AgentFrom a Linux Virtual Machine

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.

Caution If you intend to use NVIDIA GRID vGPU, you must configure 3D graphics 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. Install Horizon Agent after the 3D graphics configuration is completed.

For a 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.

   Navigate to the download page for the current release ofVMware Horizon. In the product downloads list, find the installer file for Horizon Agent for 64-bit Linux systems.

   The installer filename is `VMware-horizonagent-linux-x86_64-YYMM-y.y.y-xxxxxxx.tar.gz` for 64-bit Linux, where *YYMM* is the marketing version number, *y.y.y* is the internal version number, and *xxxxxxx* is the build number.

2. Unpack the tarball for your Linux distribution.

   For example:

   ```
   tar -xzvf VMware-horizonagent-linux-x86_64-YYMM-y.y.y-xxxxxxx.tar.gz
   ```

3. Navigate to the tarball folder.

4. Run the `install_viewagent.sh` script as a superuser.

   See install_viewagent.sh Command-Line Options for a list of the optional parameters available for this script.

   For example:

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

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

   The installer does not run unless you accept the EULA.

6. Restart the Linux virtual machine for the changes to take effect.

Results

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 terminal window:

```
install_viewagent.sh -optional parameter [parameter argument] . . .
```

The `install_viewagent.sh` script includes the following optional parameters.

Table 5-1. `install_viewagent.sh` Optional Parameters

Optional Parameters	Description
-a yes|no	Install or bypass audio input redirection support. Default is **no**.

--multiple-session	Including this parameter enables support for multi-session published desktop pools and published application pools based on a farm that includes the Linux virtual machine. By default, this parameter is not included.
	■ To prepare the machine for use in an automated instant-clone farm, include the **--multiple-session** parameter in the installation script. For example:
	`sudo ./install_viewagent.sh --multiple-session`
	■ To prepare the machine for use in a manual farm, include both the **--multiple-session** parameter and the managed agent **-M** parameter set to **no**. For example:
	`sudo ./install_viewagent.sh --multiple-session -M no`

-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 feature. Default is **no**.
-r yes|no	Restart the system automatically after installation. Default is **no**.
-s	Generate or bypass the generation of a self-signed certificate for VMwareBlastServer. Default is **yes**.
-A yes |no	Automatically accept or refuse the End User License Agreement (EULA) and Federal Information Processing Standards (FIPS) statement. You must specify **yes** for the installation to proceed. If you do not specify this parameter in the script, you must manually accept the EULA and FIPS statement during the installation.
-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 Horizon Agent to a managed or unmanaged agent. Default is **yes**.
-P yes|no	Install or bypass Printer Redirection support. 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 5-2. Examples of `install_viewagent.sh` Scripts with Parameters

Scenario	Example Script
Perform fresh installation and automatically accept the EULA and FIPS statement	`sudo ./install_viewagent.sh –A yes`
Automatically accept the EULA and FIPS statement and enable smart card redirection	`sudo ./install_viewagent.sh –A yes –m yes`
Bypass SSO support	`sudo ./install_viewagent.sh –S no`

Configure the VMwareBlastServer Certificate for Horizon Agent for Linux

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

- When the Blast Security Gateway is disabled on the Horizon Connection Server, VMwareBlastServer presents the self-signed certificate to the browser that uses HTML Access to connect to the Linux desktop.
- When the Blast Security Gateway is enabled on the Horizon Connection Server, the Blast Security Gateway presents its certificate to the browser.

To comply with industry or security regulations, you can replace the self-signed certificate for VMwareBlastServer 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 CA certificates into the Linux OS Certificate Authority store.

   For information about additional system settings that must be changed to support the CA certificate chain, refer to the documentation for your Linux distribution.

Upgrading 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.

When upgrading Horizon Agent, you can choose between two types of virtual machine deployment.

Unmanaged Virtual Machine Deployment

- This type of upgrade is available for existing unmanaged virtual machines.
- The Horizon Agent installer registers the virtual machine to Horizon Connection Server 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 Deployment

- This type of upgrade is available for unmanaged or managed virtual machines.
- The Horizon Agent installer does not communicate with Horizon Connection Server.
- The Desktop Pool Creation wizard uses **vCenter virtual machines** in the Machine Source page to select the virtual machines through vCenter.
- The deployment supports the following functions:
  - Remote Machine Power Policy
  - Allow users to reset their machines

You can use the following methods to upgrade an unmanaged virtual machine:

- Retain the unmanaged virtual machine deployment while upgrading to the latest version of Horizon Agent. This upgrade scenario does not require any configuration modifications in Horizon Connection Server.
- Upgrade from an unmanaged virtual machine deployment to a managed virtual machine deployment that uses the latest version of Horizon Agent. This upgrade scenario requires the creation of a new desktop pool based on the virtual machine.

Note To ensure the best possible performance, upgrade to a managed virtual machine deployment. The Horizon Agent upgrade does not support conversion of a managed virtual machine deployment to an unmanaged virtual machine deployment.

Upgrade 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.

Prerequisites

Verify that the `VMwareBlastServer` process is not running. To stop this process, use one of the following methods:

- Ensure that the user logs off the machine and no desktop session is active.
- Restart the virtual 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.

   Navigate to the download page for the current release ofVMware Horizon. In the product downloads list, find the installer file for Horizon Agent for 64-bit Linux systems.

   The installer filename is `VMware-horizonagent-linux-x86_64-`*`YYMM-y.y.y-xxxxxxx`*`.tar.gz` for 64-bit Linux, where *YYMM* is the marketing version number, *y.y.y* is the internal version number, and *xxxxxxx* is the build number.

2. Unpack the tarball for your Linux distribution.

   For example:

   ```
   tar -xzvf VMware-horizonagent-linux-x86_64-YYMM-y.y.y-xxxxxxx.tar.gz
   ```

3. Navigate to the tarball folder.

4. To upgrade an unmanaged virtual machine, run the `install_viewagent.sh` script using one of the following upgrade scenarios.

Option	Description
Upgrade an unmanaged virtual machine deployment and retain the unmanaged virtual machine deployment	`sudo ./install_viewagent.sh -A yes -M no` This upgrade scenario does not require the creation of a new desktop pool. You can reuse the existing desktop pool based on the virtual machine. **Note** To ensure the best possible performance, refrain from deploying an unmanaged virtual machine and deploy a managed virtual machine instead.
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 from the unmanaged virtual machine deployment and create a new desktop pool for the 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 Horizon Agent, you can reuse the existing desktop pool based on the virtual machines.

For a detailed list of the optional parameters available for the `install_viewagent.sh` script, see install_viewagent.sh Command-Line Options.

Uninstall Horizon AgentFrom a Linux Virtual Machine

To uninstall Horizon Agent from a Linux virtual machine, you must uninstall the Horizon Agent service and software and remove certain configuration files.

Prerequisites

Verify that the `VMwareBlastServer` process is not running. To stop this process, use one of the following methods:

- Ensure that the user logs off the machine and no desktop session is active.
- Restart the virtual machine.

Procedure

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

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

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

2. Manually delete the Horizon Agent configuration files from the `/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
- Using Smart Policies
- Using DPI Synchronization with Linux Remote Desktops
- Example Blast Settings for Linux Desktops
- Examples of Client Drive Redirection Options for Linux Desktops

Setting Options in Configuration Files on a Linux Desktop

For Linux desktops, you can configure certain options by adding entries to the `/etc/vmware/config` file or the `/etc/vmware/viewagent-custom.conf` file.

During Horizon Agent installation, 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`. All the configuration options are listed and documented in the configuration files. To set an option, 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 to make the changes take effect.

Configuration Options in /etc/vmware/config

The VMware BlastServer and BlastProxy processes, along with their related plug-ins and processes, use the `/etc/vmware/config` configuration file.

Note The following table includes descriptions of each agent-enforced policy setting for USB devices in the Horizon Agent configuration file. Horizon Agent uses these settings to decide whether a USB device can be forwarded to the host machine. Horizon Agent also passes these 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
appScanner	`error`, `warn`, `info`, or `debug`	`info`
BlastProxy.log.logLevel	`error`, `warn`, `info`, `verbose`, `debug`, or `trace`	`info`
BlastProxy.UdpEnabled	`true` or `false`	`true`
cdrserver.cacheEnable	`true` or `false`	`true`
cdrserver.customizedSharedFolderPath	`folder_path`	`/home/`
cdrserver.forcedByAdmin	`true` or `false`	`false`
cdrserver.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`

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

Option	Value/Format	Default
cdrserver.permissions	`R`	`RW`
cdrserver.sharedFolders	*`file_path1`*`,R;`*`file-path2`*`,;` *`file_path3`*`,R; ...`	undefined
Clipboard.Direction	`0`, `1`, `2`, or `3`	`2`
collaboration.enableControlPassing	`true` or `false`	`true`
collaboration.enableEmail	`true` or `false`	`true`
collaboration.logLevel	`error`, `info`, or `debug`	`info`
collaboration.maxCollabors	An integer less than or equal to 20	5
collaboration.serverUrl	[URL]	undefined

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

Option	Value/Format	Default
Desktop.displayNumberMax	An integer	`159`
Desktop.displayNumberMin	An integer	`100`
mksVNCServer.useUInputButtonMapping	`true` or `false`	`false`
mksvhan.clipboardSize	An integer	1024
rdeSvc.allowDisplayScaling	`true` or `false`	`false`
rdeSvc.blockedWindows	List of semicolon-separated paths to application executables	N/A
rdeSvc.enableOptimizedResize	`true` or `false`	`true`
rdeSvc.enableWatermark	`true` or `false`	`false`

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

Option	Value/Format	Default
rdeSvc.watermark.fit	`0`: Tile `1`: Center `2`: Multiple	`0`

rdeSvc.watermark.font	`serif`
	`sans-serif`
	`cursive`
	`fantasy`
	`monospace`
		`serif`

rdeSvc.watermark.fontSize	An integer within the range of values: `8-72`	`12`
rdeSvc.watermark.margin	An integer within the range of values: `0-1024`	`50`
rdeSvc.watermark.opacity	An integer within the range of values: `0-255`	`50`
rdeSvc.watermark.rotation	An integer within the range of values: `0-360`	`45`

rdeSvc.watermark.template	String constructed using any of the available information variables:
	`$BROKER_USER_NAME`
	`$BROKER_DOMAIN_NAME`
	`$USER_NAME`
	`$USER_DOMAIN`
	`$MACHINE_NAME`
	`$REMOTE_CLIENT_IP`
	`$CLIENT_CONNECT_TIME`
		`$USER_DOMAIN\$USER_NAME\n$MACHINE_NAME O`

RemoteDisplay.allowAudio	`true` or `false`	`true`
RemoteDisplay.allowH264	`true` or `false`	`true`
RemoteDisplay.allowH264YUV444	`true` or `false`	`true`
RemoteDisplay.allowHEVC	`true` or `false`	`true`
RemoteDisplay.allowHEVCYUV444	`true` or `false`	`true`

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

Option	Value/Format	Default
RemoteDisplay.allowVMWKeyEvent2Unicode	`true` or `false`	`true`
RemoteDisplay.buildToPNG	`true` or `false`	`false`
RemoteDisplay.enableNetworkContinuity	`true` or `false`	`true`
RemoteDisplay.enableNetworkIntelligence	`true` or `false`	`true`
RemoteDisplay.enableStats	`true` or `false`	`false`
RemoteDisplay.enableUDP	`true` or `false`	`true`
RemoteDisplay.maxBandwidthKbps	An integer	1000000
RemoteDisplay.minBandwidthKbps	An integer	256
RemoteDisplay.maxFPS	An integer	30
RemoteDisplay.maxQualityJPEG	An integer within the range of values: 1–100	90
RemoteDisplay.midQualityJPEG	An integer within the range of values: 1–100	35

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

Option	Value/Format	Default
RemoteDisplay.minQualityJPEG	An integer within the range of values: 1–100	25
RemoteDisplay.qpmaxH264	An integer within the range of values: 0–51	36
RemoteDisplay.qpminH264	An integer within the range of values: 0–51	10
UsbRedirPlugin.log.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`
UsbRedirServer.log.logLevel	`error`, `warn`, `info`, `debug`, `trace`, or `verbose`	`info`
vdpservice.log.logLevel	`fatal error`, `warn`, `info`, `debug`, or `trace`	`info`
viewusb.AllowAudioIn	**`{m|o}:{true|false}`**	undefined, which equates to `true`
viewusb.AllowAudioOut	**`{m|o}:{true|false}`**	undefined, which equates to `false`
viewusb.AllowAutoDeviceSplitting	**`{m|o}:{true|false}`**	undefined, which equates to `false`
viewusb.AllowDevDescFailsafe	**`{m|o}:{true|false}`**	undefined, which equates to `false`
viewusb.AllowHIDBootable	**`{m|o}:{true|false}`**	undefined, which equates to `true`
viewusb.AllowKeyboardMouse	**`{m|o}:{true|false}`**	undefined, which equates to `false`
viewusb.AllowSmartcard	**`{m|o}:{true|false}`**	undefined, which equates to `false`
viewusb.AllowVideo	**`{m|o}:{true|false}`**	undefined, which equates to `true`
viewusb.DisableRemoteConfig	**`{m|o}:{true|false}`**	undefined, which equates to `false`

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

Option	Value/Format	Default
viewusb.ExcludeAllDevices	**{true|false}**	undefined, which equates to `false`
viewusb.ExcludeFamily	**{m|o}:***family_name_1***[;family_name_2;...]**	undefined
viewusb.ExcludePath	**{m|o}:bus-***x1***[/***y1***].../ port-***z1***[;bus-***x2***[/***y2***].../port-***z2***;...]**	undefined
viewusb.ExcludeVidPid	**{m|o}:vid-***xxx1***_ pid-***yyy1***[;vid-***xxx2***_pid-***yyy2***;..]**	undefined
viewusb.IncludeFamily	**{m|o}:***family_name_1***[;***family_name_2***]...**	undefined
viewusb.IncludePath	**{m|o}:bus-***x1***[/***y1***].../ port-***z1***[;bus-***x2***[/***y2***].../portz2;...]**	undefined
viewusb.IncludeVidPid	**{m|o}:vid-***xxx1***_ pid-***yyy1***[;vid-***xxx2***_pid-***yyy2;...***]**	undefined

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

Option	Value/Format	Default
viewusb.SplitExcludeVidPid	{m|o}:vid-*xxx1*_pid-*yyy1*[;vid-*xxx2*_pid-*yyy2*;...]	undefined
viewusb.SplitVidPid	{m|o}: vid-*xxxx*_pid-*yyyy*([exintf:*zz*[;exintf:*ww*]])[;...]	undefined
VMWPkcs11Plugin.log.enable	true or false	false
VMWPkcs11Plugin.log.logLevel	error, warn, info, debug, trace, or verbose	info
VVC.RTAV.Enable	true or false	true
VVC.ScRedir.Enable	true or false	true
VVC.logLevel	fatal error, warn, info, debug, or trace	info

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 deactivate the client drive redirection feature.
CollaborationEnable	true or false	true	Use this option to enable or deactivate the Session Collaboration feature on Linux desktops.

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

Option	Value	Default	Description
DPISyncEnable	`true` or `false`	`true`	Set this option to enable or deactivate the DPI Synchronization feature, which ensures that the DPI setting in the remote desktop matches the client system's DPI setting.
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.
HelpDeskEnable	`true` or `false`	`true`	Set this option to enable or deactivate 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 deactivated, 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.

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

Option	Value	Default	Description
MaxSessionsBuffer	An integer between 1 and the value specified for **Max Sessions Per RDS Host** in the farm configuration wizard. See Create an Automated Instant-Clone Farm of Linux Hosts.	`5`	When configuring farms and multi-session desktop pools, use this option to specify the number of pre-launched desktops per host machine.
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.

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

Option	Value	Default	Description
RunOnceScriptTimeout		120	Use this option to set the timeout time in seconds for the RunOnceScript option. For example, set `RunOnceScriptTimeout=120`
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 by the OpenSSL standard. To find information about the OpenSSL-defined format, type these keywords into an Internet search engine: **`openssl cipher string`**.
SSLProtocols	A text string	TLSv1_1:TLSv1_2	Use this option to specify the security protocols. The supported protocols are 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 desktop, the option takes effect regardless if the SSO feature is enabled or not. If this option is specified in a RHEL/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.x desktops. VMware Horizon supports only the Gnome desktop environment on RHEL/CentOS 8.x desktops.
SSOEnable	`true` or `false`	`true`	Set this option to enable/deactivate 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 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 to connect to 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
DEMEnable	`true` or `false`	`false`	Set this option to enable or deactivate smart policies created in Dynamic Environment Manager. If the option is set to enable, and the condition in a smart policy is met, then the policy is enforced.
DEMNetworkPath	A text string		This option must be set to the same network path that is set in the Dynamic Environment Manager Console. The path must be in the format similar to `//10.111.22.333/view/LinuxAgent/DEMConfig`. The network path must correspond to a public, shared folder which does not require user name and password credentials for access.

Note The VMwareBlastServer process uses the SSLCiphers, SSLProtocols, and SSLCipherServerPreference security options. 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 Linux desktops. To control the behavior of the digital watermark feature on specific Linux desktops, you use environment variables instead of Smart Policies.

You can create policies for user environment settings that control a range of behaviors. 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. 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 VMware Horizon environment must meet certain requirements.

- You must install Horizon Agent and VMware Dynamic Environment Manager 9.4 or later on a remote Windows desktop.

- Users must use Horizon Client 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 Horizon Smart Policies to control the behavior of USB redirection, clipboard redirection, and client drive redirection 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 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 a range of behaviors. 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. 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 and Environment Variable Definitions

When you define a Horizon Smart Policy or environment variable in Dynamic Environment Manager, you can add conditions that must be met for the policy or variable to take effect. For example, you can add a condition that deactivates 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 or environment variable definition in order for the supported settings to take effect in a remote Linux desktop. These are the only conditions that are currently supported. If other conditions are set, the result of the condition evaluation is false.

Table 6-3. Required Conditions for Remote Linux Desktops

Condition	Description
`Operating System Architecture`	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.

Note To control the behavior of the digital watermark feature, you use environment variables instead of Horizon Smart Policies. See Configure a Digital Watermark Using Environment Variables.

You can create policies for user environment settings that control a range of behaviors. 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. 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.

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 and Environment Variable 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.

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.

Results

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 deactivates USB redirection for the desktop pool named Ubuntu1804, the USB redirection feature is enabled when Sue connects to a remote desktop in the Ubuntu1804 desktop pool.

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.

Configure a Digital Watermark Using Environment Variables

You can configure environment variables in Dynamic Environment Manager to control the behavior of the digital watermark feature on specific Linux desktops.

Prerequisites

- Install and configure Dynamic Environment Manager. See Installing Dynamic Environment Manager and Configuring Dynamic Environment Manager.
- 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.

Configure Environment Variables in Dynamic Environment Manager

Use the following steps to configure environment variables that define the settings for a digital watermark on a Linux desktop.

1 In the Dynamic Environment Manager Management Console, click the **User Environment** tab and then select **Environment Variables**.

  Existing environment variable definitions, if any, appear in the Environment Variables pane.

2 To create a new environment variable, click **Create**.

3 Click the **Settings** tab and define the environment variable settings.

  a In the General Settings section, enter a name for the settings definition in the **Name** text box.

  b In the Environment Variable Settings section, enter the variable name and value exactly as described in the "Dynamic Environment Manager Environment Variable Values for the Digital Watermark Feature" section that follows this procedure.

4 Add the conditions required to use the environment variable 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
```

  e To save the environment variable, click **Save**.

Repeat this procedure for each additional environment variable that you want to configure for digital watermark.

Note After saving your new or updated environment variables in a high-latency network, wait at least a minute while Dynamic Environment Manager finishes processing the changes before you make the affected desktops available to your end users.

Dynamic Environment Manager Environment Variable Values for the Digital Watermark Feature

In Dynamic Environment Manager, configure the environment variables described in the following table. Each environment variable maps to a corresponding configuration option in the `/etc/vmware/config` file. The environment variable settings take priority over the settings in `/etc/vmware/config`.

Environment variable	Corresponding option in /etc/vmware/config	Value/format of variable	Default	Description
WATERMARK	rdeSvc.enableWatermark	`0`: Deactivate `1`: Enable	`0`	Enables or deactivates the digital watermark feature. For information about the feature, see Features of Horizon Linux Desktops.

WATERMARK_FONT_NAME	rdeSvc.watermark.font	`serif`
		`sans-serif`
		`cursive`
		`fantasy`
		`monospace`
			`serif`	Defines the font used for the digital watermark.

WATERMARK_FONT_SIZE	rdeSvc.watermark.fontSize	An integer within the range of values: `8-72`	`12`	Defines the font size (in points) of the digital watermark.

Environment variable	Corresponding option in /etc/vmware/config	Value/format of variable	Default	Description
WATERMARK_IMAGE_LAYOUT	rdeSvc.watermark.fit	`0`: Tile `1`: Center `2`: Multiple	`0`	Defines the layout of the digital watermark on the screen, which is divided into nine squares: ■ 0 = Tile: Watermark is positioned in all nine squares. Application sessions always use this layout. ■ 1 = Center: Watermark is positioned in the center square. ■ 2 = Multiple: Watermark is positioned in the center and four corner squares. If the watermark size exceeds the square size, it is scaled to maintain the aspect ratio.
WATERMARK_MARGIN	rdeSvc.watermark.margin	An integer within the range of values: `0-1024`	`50`	Defines the amount of space (in pixels) around the digital watermark for the Tile layout. As the watermark scales, the margin also scales proportionally.
WATERMARK_OPACITY	rdeSvc.watermark.opacity	An integer within the range of values: `0-255`	`50`	Defines the transparency level of the digital watermark text.

Environment variable	Corresponding option in /etc/vmware/config	Value/format of variable	Default	Description

WATERMARK_TEXT	rdeSvc.watermark.template	String constructed using any of the available information variables:
		`$BROKER_USER_NAME`
		`$BROKER_DOMAIN_NAME`
		`$USER_NAME`
		`$USER_DOMAIN`
		`$MACHINE_NAME`
		`$REMOTE_CLIENT_IP`
		`$CLIENT_CONNECT_TIME`
			`$USER_DOMAIN\$USER_NAME\n$MACHINE_NAME\nOn $CLIENT_CONNECT_TIME`	The text that you want to display for the digital watermark. Construct the watermark using any combination and order of the information variables. The character limit is 1024 characters and 4096 characters after expansion. The text is truncated if it exceeds the maximum length.

WATERMARK_TEXT_ROTATION	rdeSvc.watermark.rotation	An integer within the range of values: `0-360`	`45`	Defines the display angle of the digital watermark text.

Processing Order for Environment Variables

Dynamic Environment Manager processes environment variables each time a user connects or reconnects to the remote desktop.

Dynamic Environment Manager processes multiple environment variables in alphabetical order based on the environment variable name. Environment variables appear in alphabetical order in the Environment Variables pane. If several environment variables conflict, the last environment variable processed takes precedence. For example, if you have an environment variable named B that enables the watermark for the user named Sue, and another environment variable named A that deactivates the watermark for the desktop pool named Ubuntu1804, the watermark is enabled when Sue connects to a remote desktop in the Ubuntu1804 desktop pool.

Using DPI Synchronization with Linux Remote Desktops

This topic provides an overview of the DPI Synchronization feature for Linux remote desktops. The DPI Synchronization feature ensures that the DPI value in a remote session changes to match the DPI value of the client system when users connect to a remote desktop or published application.

The following considerations apply to the DPI Synchronization feature for Linux remote desktops and published applications.

- The **DPISyncEnable** configuration option in the `/etc/vmware/viewagent-custom.conf` file determines whether the DPI Synchronization feature is enabled for a desktop. The feature is enabled by default. For more information, see Setting Options in Configuration Files on a Linux Desktop.

- To support the DPI Synchronization feature, RHEL/CentOS 7.x desktops must be running RHEL/CentOS 7.6 or later.
- To use DPI Synchronization on a client system with multiple monitors, configure each monitor with the same DPI setting. DPI Synchronization with a Linux remote desktop does not work if the client monitors have different DPI settings.
- DPI Synchronization is supported when users reconnect to a Linux remote desktop running in a Gnome desktop environment. Linux remote desktops running in a KDE or Mate desktop environment do not support DPI Synchronization upon reconnection. To use DPI Synchronization on a desktop running KDE or Mate, users must log out and log in to a new session.
- Gnome desktops do not support synchronization to the exact DPI values of the client system. Instead, DPI Synchronization rounds the client DPI value down to the nearest multiple of 96 for displaying the remote session. For example, suppose that the client system uses 250 DPI. Since 250 is greater than 192 (2 x 96) but less than 288 (3 x 96), the remote session uses the lower value, 192 DPI.

  Table 6-4. DPI Synchronization Values for Gnome Desktops

Client DPI	Remote Session DPI
96–191	96 (1 x 96)
192–287	192 (2 x 96)
288–383	288 (3 x 96)
384–479	384 (4 x 96)

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 network connection is less than optimal.

Video Encoders Used for Desktop Sessions

The video encoder used for a desktop session depends on the capabilities of both the desktop and client systems. When a user opens a session, Horizon evaluates the hardware and software capabilities of the desktop and client and selects the highest-priority encoder that is supported by both.

For example, Horizon uses HEVC if all the following components are configured to support or allow HEVC:

- The Linux desktop system
- Horizon Agent on the Linux desktop

- The client system
- Horizon Client on the client system

If any one of the components does not support HEVC, Horizon selects the next encoder on the priority list that is supported by all the components.

For a session that uses vGPU technology, Horizon selects from the following encoders listed in order of priority:

1 HEVC YUV 4:4:4

2 H.264 YUV 4:4:4

3 HEVC

4 H.264

5 Switch Encoder

6 BlastCodec

For a non-vGPU session, Horizon selects from the following priority list of encoders:

1 Switch Encoder

2 H.264

3 BlastCodec

Example VMware Blast Extreme Protocol Settings

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

Table 6-5. 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 6-5. 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 or HEVC)	RemoteDisplay.qpmaxH264	28	36	36	36	36	42
Display Quality (H.264 or HEVC)	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 Virtual Desktop Pools 7

To configure Linux virtual machines for use as single-session remote desktops, you must create a virtual desktop pool based on Linux virtual machines. Each virtual desktop in the pool supports a single user session at one time.

Horizon Agent for Linux supports the following desktop pool types for single-session virtual desktops:

- 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 create an automated full-clone desktop pool, you must install Horizon 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 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
- Manage Linux Desktop Pools
- Create an Automated Full-Clone Desktop Pool for Linux
- Create an Instant-Clone Floating Desktop Pool for Linux

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*.

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 desktop and NVIDIA GRID vGPU for 3D 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.

Results

The Linux virtual machines are ready to be used as remote desktops in a Horizon 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 virtual machines to the manual desktop pool. If the pool contains both Windows and Linux machines, the pool is treated as a Windows pool, and users 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 delete desktop pools using Horizon Console 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 ■ 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 ■ NVIDIA GRID vGPU for 3D

For more information, see the *Horizon Administration* document.

Create an Automated Full-Clone Desktop Pool for Linux

You can create an automated full-clone desktop pool from 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 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*.

Prerequisites

- Verify that you have prepared a virtual machine for cloning. See Chapter 2 Preparing a Linux Virtual Machine for Desktop Deployment.
- Verify that Horizon Agent is installed on the Linux guest operating systems. See Install Horizon Agent on a Linux Virtual Machine.
- 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, available from the VMware vSphere Documentation. 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 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 desktop and NVIDIA GRID vGPU for 3D 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 deployment.

Horizon Agent for Linux only supports instant-clone desktop pools created from virtual machines running the following operating systems:

- Ubuntu 18.04/20.04
- RHEL Workstation 7.2 or later, and 8.x

- RHEL Server 7.8, 7.9, 8.3, and 8.4
- CentOS 7.8, 7.9, 8.3, and 8.4
- SLED/SLES 12.x/15.x

Note vGPU graphics capabilities are only supported on instant-clone desktop pools created from Linux machines running the following operating systems:

- Ubuntu 18.04/20.04
- RHEL/CentOS 7.9, 8.3, and 8.4

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*.

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.x, perform the AD integration using Samba Winbind offline join. Instant-clone desktop pools are not supported for RHEL 8.x virtual machines that use PBISO authentication.
- Familiarize yourself with the installation steps for Horizon 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*, available from VMware vSphere Documentation.
- Verify that vCenter Server is added to Horizon Connection Server.

Procedure

1. Create a parent Linux virtual machine (VM) and perform a fresh installation of an operating system that supports the creation of instant-clone desktop pools. See the list of supported operating systems earlier in this article.

   For more information, see Create a Virtual Machine and Install Linux.

   Important Always use a VM equipped with a freshly installed Linux operating system as the parent VM of an instant-clone desktop pool. Do not use an already cloned system as the parent VM.

2 For Ubuntu machines, manually install Open VMware Tools (OVT) 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 on 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 Desktops with Active Directory.
- To use Samba Winbind offline join, configure the following options in the `/etc/vmware/viewagent-custom.conf` file. Replace *YOURDOMAIN* with the NetBIOS name of your domain.

  ```
  OfflineJoinDomain=samba

  NetbiosDomain=YOURDOMAIN
  ```

  Note You must use Samba Winbind to integrate a RHEL 8.x 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 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
```

Note For best results, use NetworkManager instead of WICD for network management. WICD might produce problems when used with instant-cloned SLED/SLES 15.x VMs.

7 (Optional) If you want to add an NFS mount in the `/etc/fstab` file of the Linux golden image, 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 Linux golden 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 *Horizon Administration* document.

8 Shut down the Linux VM and create a golden image by creating a snapshot of your powered off Linux VM using VMware vSphere Client or VMware vSphere Web Client.

For more information, see "Take a Snapshot in the VMware Host Client" in *vSphere Single Host Management - VMware Host Client*, available from VMware vSphere Documentation.

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 uses 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 Agent 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.

Results

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 golden image or remove it from the vCenter Server inventory if the pool exists. If you remove the golden 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*.

Setting Up Linux Published Desktops and Applications for Multi-Session Use

8

You can create published desktop pools and application pools based on farms of Linux virtual machines. Each published desktop or published application can support multiple user sessions at the same time.

To establish multiple user sessions on a published desktop or published application, the BlastProxy process receives each connection request from a client and forwards it to a specific process on VMwareBlastServer. The BlastProxy process also forwards data traffic from VMwareBlastServer to the appropriate client.

About Linux Farms, Published Desktops, and Published Applications

This section describes the entities that you can configure to provide your end users with multi-session Linux desktops and applications.

- **Multi-session Host Machine:** You configure a multi-session host machine by following the steps described in Prepare a Linux Machine for Remote Desktop Deployment and ensuring that you install Horizon Agent with the **`--multiple-session`** parameter included. You can then add the Linux host machine to a farm.
- **Farm:** A farm consists of multi-session Linux host machines and serves as the basis for a published desktop pool or published application pool. You can create manual farms and automated instant-clone farms of Linux host machines.
- **Published Desktop Pool:** A published desktop pool is provisioned from either a manual farm or automated instant-clone farm of multi-session Linux host machines. Each published desktop can support multiple user sessions at the same time. Multi-session published desktops require fewer virtual machine resources than single-session virtual desktops. However, multi-session desktops offer more limited support for Horizon features.
- **Published Application Pool:** With a published application pool, you can deliver a single application to many users. When you create an application pool, you deploy an application in the data center that users can access from anywhere on the network. An application pool runs on either a manual farm or automated instant-clone farm of multi-session host machines.

This chapter includes the following topics:

- Considerations for Linux Farms, Published Desktops, and Published Applications

- Create a Manual Farm of Linux Virtual Machines
- Create an Automated Instant-Clone Farm of Linux Hosts
- Create a Linux Published Desktop Pool
- Create a Linux Published Application Pool Manually
- Create a Linux Published Application Pool from a List of Installed Applications

Considerations for Linux Farms, Published Desktops, and Published Applications

Keep in mind the following feature limitations and considerations when working with Linux farms, published desktop pools, and published application pools.

- Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications.
- vGPU capabilities are not supported for published applications.
- Published desktops and applications are not supported on the KDE desktop environment.
- All the host machines in a farm must be running the same operating system. For example, you can create a farm consisting of all Linux hosts or Windows hosts, but you cannot create a farm consisting of a mix of Linux and Windows hosts.
- All the Linux host machines in a farm must be running the same Linux distribution. For example, you can create a farm consisting of all RHEL Workstation 8.2 hosts or all Ubuntu 18.04 hosts, but you cannot create a farm consisting of a mix of RHEL Workstation 8.2 and Ubuntu 18.04 hosts.
- Published desktops do not support the following Horizon features:
  - USB redirection
  - Smart card redirection
- Each published desktop or published application can support up to 50 user sessions, if the host Linux virtual machine meets the minimum vCPU and vMemory requirements. For more information, see Create a Virtual Machine and Install Linux.
- To make it faster for users to start a remote desktop session, Horizon Agent can pre-launch a specified number of desktops per host machine. You can specify the number of pre-launched desktops by using the **MaxSessionsBuffer** configuration option in `/etc/vmware/viewagent-custom.conf`. See Setting Options in Configuration Files on a Linux Desktop.
- When running a Linux published application from Horizon Client for Windows, users can improve the application performance by setting the preference to hide window contents while dragging. For example, navigate to **Control Panel > System and Security > System > Advanced system settings > Advanced > Settings** and deselect **Show window contents while dragging**.

- When logging in to a RHEL 8.x desktop to connect to a published application, users must enable **Classic (X11 display server)**. Otherwise, the application window is displayed incorrectly, such as without minimize and maximize buttons.
- When connecting to a Linux published application from a client system using a multiple-monitor configuration, verify that all monitors have the identical scale setting. Otherwise, the application window cannot be moved between monitor screens.
- Linux published applications do not support enabling the **Multi-Session Mode** setting in the configuration wizard for application pools. When you create a Linux application pool, you can only configure it in single-session mode. For example, if a user opens a published application on client A and then opens the same published application or another published application based on the same farm on client B, then the session on client A is disconnected and reconnected on client B.
- Horizon Agent for Linux does not support session stealing between published desktops and published applications.

  For example, if a user has opened a published desktop session and then attempts to open an application session based on the same farm, the desktop session remains active and the application session is not established. Likewise, if the user has opened an application session and then attempts to open a published desktop session based on the same farm, the application session remains active and the desktop session is not established.

Create a Manual Farm of Linux Virtual Machines

You create a manual farm as part of the process to give users access to multi-session published applications or desktops.

Note All the host machines in a farm must be running the same Linux distribution. For example, you can create a farm consisting of all RHEL Workstation 8.2 hosts or all Ubuntu 18.04 hosts, but you cannot create a farm consisting of a mix of RHEL Workstation 8.2 and Ubuntu 18.04 hosts.

Prerequisites

- Prepare the multi-session host machines that you want to include in the farm. See the subtopics under Chapter 2 Preparing a Linux Virtual Machine for Desktop Deployment.
- Verify that each host machine is running one of the following Linux operating systems:
  - RHEL Workstation 8.x/7.x
  - Ubuntu 18.04
- Verify that you have installed Horizon Agent on each host machine with the **--multiple-session** parameter included and the managed agent **-M** parameter set to **no**. For example:

```
sudo ./install_viewagent.sh --multiple-session -M no
```

- Verify that all the host machines have the Available status. In Horizon Console, select **Settings > Registered Machines** and check the status of each host machine on the RDS Hosts tab.

Procedure

1 In Horizon Console, select **Inventory > Farms**. Then click **Add**.

The farm configuration wizard appears. As you advance through the wizard, you can go directly back to any prior page by clicking the page name in the navigation pane.

2 In the **Type** page of the wizard, select **Manual Farm** and then click **Next**.

3 In the **Identification and Settings** page of the wizard, configure the required settings.

Setting	Description
ID	Unique name that identifies the farm.
Description	Description of this farm.
Access Group	Select an access group for the farm, or leave the farm in the default root access group.
Default Display Protocol	Select **VMware Blast**. VMware Blast is the only display protocol supported for Linux remote sessions.
Allow Users to Choose Protocol	Select **Yes** or **No**. This setting applies to published desktop pools only, not application pools. If you select **Yes**, users can choose the display protocol when they connect to a published desktop from Horizon Client. The default is **Yes**.
Pre-launch Session Timeout (Applications Only)	Determines the amount of time that an application configured for pre-launch is kept open. The default is **After 10 minutes**. If the end user does not start any application in Horizon Client, the application session is disconnected if the idle session times out or if pre-launch session times out. If you want to end the pre-launch session after timeout, you must set the **Logoff Disconnected Sessions** option to **Immediate**.
Empty Session Timeout (Applications Only)	Determines the amount of time that an empty application session is kept open. An application session is empty when all the applications that run in the session are closed. While the session is open, users can open applications faster. You can save system resources if you disconnect or log off empty application sessions. Select **Never**, **Immediate**, or set the number of minutes as the timeout value. The default is **After 1 minute**. If you select **Immediate**, the session logs off or disconnects within 30 seconds.
When Timeout Occurs	Determines whether an empty application session is disconnected or logged off after the **Empty Session Timeout** limit is reached. Select **Disconnect** or **Log Off**. A session that is logged off frees up resources, but opening an application takes longer. The default is **Disconnect**.

Setting	Description
Logoff Disconnected Sessions	Determines when a disconnected session is logged off. This setting applies to both desktop and application sessions. Select **Never**, **Immediate**, or **After ... minutes**. Use caution when you select **Immediate** or **After ... minutes**. When a disconnected session is logged off, the session is lost. The default is **Never**.
Allow Session Collaboration	Select **Enabled** to allow users of desktop pools based on this farm to invite other users to join their remote desktop sessions. Session owners and collaborators must use the VMware Blast protocol.

Click **Next** to proceed to the next page of the configuration wizard.

4 In the **Load Balancing Settings** page of the wizard, configure the required settings.

Setting	Description
Use Custom Script	Select this setting to use a custom script for load balancing.
Include Session Count	Select this setting to include the session count on the Linux host for load balancing. If none of the settings are selected for load balancing and if the custom script setting is not selected, Horizon uses the session count by default. Disable this setting if you do not need to consider the session count for load balancing.
CPU Usage Threshold	Threshold value for the CPU usage in percentage. Horizon uses the configured CPU threshold to calculate the CPU load index factor. You can set a value from 0 to 100. The recommended value is 90. By default, this setting is not considered for load balancing. The default value is 0.
Memory Usage Threshold	Threshold value for the memory in percentage. Horizon uses the configured memory threshold to calculate the Memory Load Index factor. You can set a value from 0 to 100. The recommended value is 90. By default, this setting is not considered for load balancing. The default value is 0.
Disk Queue Length Threshold	Threshold of the average number of both read and write requests that were queued for the selected disk during the sample interval. Horizon uses the configured threshold to calculate the Disk Load Index factor. You can set the value to any positive integer. By default, this setting is not considered for load balancing. The default value is 0.
Disk Read Latency Threshold	Threshold of the average time of write of data to the disk in milliseconds. Horizon uses the configured threshold to calculate the Disk Load Index factor. You can set the value to any positive integer. By default, this setting is not considered for load balancing. The default value is 0.

Click **Next** to proceed to the next page of the configuration wizard.

5 In the **Select RDS Hosts** page of the wizard, select the Linux virtual machines that you want to add to the farm. Then click **Next**.

6 In the **Ready to Complete** page of the wizard, review the settings and click **Submit** to create the manual farm.

What to do next

Create a published application pool or a published desktop pool. See one of the following topics:

- Create a Linux Published Desktop Pool
- Create a Linux Published Application Pool Manually
- Create a Linux Published Application Pool from a List of Installed Applications

Create an Automated Instant-Clone Farm of Linux Hosts

You create an automated instant-clone farm as part of the process to give users access to published applications or published desktops. An automated farm consists of multi-session Linux hosts that are instant-clone virtual machines (VMs) in vCenter Server.

Overview of Instant-clone Farms

An automated instant-clone farm is created from a golden image using the vmFork technology (called instant clone API) in vCenter Server. In addition to using the instant clone API from vCenter Server, Horizon creates several types of internal VMs (Internal Template, Replica VM, and ParentVM) in order to manage these clones in a more scalable way.

Although helpful in speeding up the provisioning speed, the use of parentVM does increase the memory requirement across the cluster. Sometimes when the benefit of having more memory outweighs the increase in provisioning speed, Horizon automatically chooses to provision instant clones directly from replicaVM, without creating any parentVM. This feature is called Smart Provisioning. A single instant clone farm can have both instant clones that are created with parentVMs or without parentVMs.

When parentVM is used, instant clones share the virtual disk of the parentVM and therefore consume less storage than full VMs. In addition, instant clones share the memory of the parentVM when they are first created, which contributes to fast provisioning. Once the instant clone VM is provisioned and the machine starts to be used, additional memory is utilized.

An instant-clone desktop farm has the following benefits:

- The provisioning of instant clones is fast, with or without using parentVM.
- Instant clones are always created in a powered-on state, ready for use.
- You can patch a farm of instant clones in a rolling process with zero downtime.

Connection Server creates the instant-clone virtual machines based on the parameters that you specify when you create the farm. Instant clones share a virtual disk of a parentVM and therefore consume less storage than full virtual machines. In addition, instant clones share the memory of a parentVM and are created using the vmFork technology.

Process of Creating Instant Clones

Publishing an image is a process by which internal VMs needed for instant cloning are created from a golden image and its snapshot. This process only happens once per image and might take some time.

Horizon performs the following steps to create a pool of instant clones:

1 Horizon publishes the image that you select. In vCenter Server, four folders (`ClonePrepInternalTemplateFolder`, `ClonePrepParentVmFolder`, `ClonePrepReplicaVmFolder`, and `ClonePrepResyncVmFolder`) are created if they do not exist, and some internal VMs that are required for cloning are created. In Horizon Console, you can see the progress of this operation on the **Summary** tab of the desktop pool. During publishing, the Pending Image pane shows the name and state of the image.

   Note Do not tamper with the four folders or the internal VMs that they contain. Otherwise, errors might occur. The internal VMs are removed when they are no longer needed. Normally the VMs are removed within 5 minutes of pool deletion or a push-image operation. However, sometimes the removal can take up to 30 minutes. If there are no internal VMs in all four folders, these folders are unprotected and you can delete these folders.

2 After the image is published, Horizon creates the instant clones. This process is fast. During this process, the Current Image pane in Horizon Console shows the name and state of the image.

After the farm is created, you can change the image through the push-image operation. As with the creation of a farm, the new image is first published. Then the clones are recreated.

When an instant-clone pool farm is created, Horizon spreads the pool across datastores automatically in a balanced way. If you edit a farm to add or remove datastores, rebalancing of the cloned VMs happens automatically when a new clone is created.

Prerequisites

- Verify that Connection Server is installed. See the *Horizon Installation* document.
- Verify that Connection Server settings for vCenter Server are configured in Horizon Console. See the *Horizon Administration* document.
- Verify that you have a sufficient number of ports on the ESXi virtual switch that is used for the virtual machines that are used as remote desktops. The default value might not be sufficient if you create large desktop pools.
- Verify that you have prepared a golden-image host machine. See Chapter 2 Preparing a Linux Virtual Machine for Desktop Deployment. Horizon Agent must be installed on the host machine.
- Verify that each host machine is running one of the following Linux operating systems:
  - RHEL Workstation 8.x/7.x
  - Ubuntu 18.04
- Verify that you have installed Horizon Agent on the golden-image machine with the **`--multiple-session`** parameter included. For example:

```
sudo ./install_viewagent.sh --multiple-session
```

- Take a snapshot of the golden-image host machine in vCenter Server. You must shut down the host machine before you take the snapshot. Connection Server uses the snapshot as the baseline configuration for creating the clones.

  For more information, see "Take a Snapshot in the VMware Host Client" in *vSphere Single Host Management - VMware Host Client*, available from VMware vSphere Documentation.

Procedure

1 In Horizon Console, select **Inventory > Farms**. Then click **Add**.

  The farm configuration wizard appears. As you advance through the wizard, you can go directly back to any prior page by clicking the page name in the navigation pane.

2 In the **Type** page of the wizard, select **Automated Farm** and then click **Next**.

3 In the **vCenter Server** page of the wizard, select **Instant Clone** and click **Next**.

4 In the **Storage Optimization** page of the wizard, configure the required settings.

Setting	Description
Use VMware Virtual SAN/Do not use VMware Virtual SAN	Specify whether to use VMware vSAN, if available. vSAN is a software-defined storage tier that virtualizes the local physical storage disks available on a cluster of ESXi hosts.
Use Separate Datastores for Replica and OS Disks	(Available only if you do not use vSAN) You can place replica and OS disks on different datastores for performance or other reasons. If you select this option, you can select the options to select one or more instant-clone datastores or replica disk datastores.

  Click **Next** to proceed to the next page of the configuration wizard.

5 In the **Identification and Settings** page of the wizard, configure the required settings.

Setting	Description
ID	Unique name that identifies the farm.
Description	Description of this farm.
Access Group	Select an access group for the farm, or leave the farm in the default root access group.
Default Display Protocol	Select **VMware Blast**. VMware Blast is the only display protocol supported for user sessions on Linux desktops.
Allow Users to Choose Protocol	Select **Yes** or **No**. This setting applies to published desktop pools only, not application pools. If you select **Yes**, users can choose the display protocol when they connect to a published desktop from Horizon Client. The default is **Yes**.
3D Renderer	Select 3D graphics rendering for desktops. NVIDIA GRID vGPU is the only 3D rendering option offered for an automated farm of instant-clone hosts.

Setting	Description
Pre-launch Session Timeout (Applications Only)	Determines the amount of time that an application configured for pre-launch is kept open. The default is **After 10 minutes**. If the end user does not start any application in Horizon Client, the application session is disconnected if the idle session times out or if pre-launch session times out. If you want to end the pre-launch session after timeout, you must set the **Logoff Disconnected Sessions** option to **Immediate**.
Empty Session Timeout (Applications Only)	Determines the amount of time that an empty application session is kept open. An application session is empty when all the applications that run in the session are closed. While the session is open, users can open applications faster. You can save system resources if you disconnect or log off empty application sessions. Select **Never**, **Immediate**, or set the number of minutes as the timeout value. The default is **After 1 minute**. If you select **Immediate**, the session logs off or disconnects within 30 seconds.
When Timeout Occurs	Determines whether an empty application session is disconnected or logged off after the **Empty Session Timeout** limit is reached. Select **Disconnect** or **Log Off**. A session that is logged off frees up resources, but opening an application takes longer. The default is **Disconnect**.
Logoff Disconnected Sessions	Determines when a disconnected session is logged off. This setting applies to both desktop and application sessions. Select **Never**, **Immediate**, or **After ... minutes**. Use caution when you select **Immediate** or **After ... minutes**. When a disconnected session is logged off, the session is lost. The default is **Never**.
Allow Session Collaboration	Select **Enabled** to allow users of desktop pools based on this farm to invite other users to join their remote desktop sessions. Session owners and collaborators must use the VMware Blast protocol.
Max Sessions Per RDS Host	Determines the maximum number of sessions that a host machine can support. Select **Unlimited** or **No More Than ...**. The default is **Unlimited**.

Click **Next** to proceed to the next page of the configuration wizard.

6 In the **Load Balancing Settings** page of the wizard, configure the required settings.

Setting	Description
Use Custom Script	Select this setting to use a custom script for load balancing.
Include Session Count	Select this setting to include the session count on the Linux host for load balancing. If none of the settings are selected for load balancing and if the custom script setting is not selected, Horizon uses the session count by default. Disable this setting if you do not need to consider the session count for load balancing.
CPU Usage Threshold	Threshold value for the CPU usage in percentage. Horizon uses the configured CPU threshold to calculate the CPU load index factor. You can set a value from 0 to 100. The recommended value is 90. By default, this setting is not considered for load balancing. The default value is 0.

Setting	Description
Memory Usage Threshold	Threshold value for the memory in percentage. Horizon uses the configured memory threshold to calculate the Memory Load Index factor. You can set a value from 0 to 100. The recommended value is 90. By default, this setting is not considered for load balancing. The default value is 0.
Disk Queue Length Threshold	Threshold of the average number of both read and write requests that were queued for the selected disk during the sample interval. Horizon uses the configured threshold to calculate the Disk Load Index factor. You can set the value to any positive integer. By default, this setting is not considered for load balancing. The default value is 0.
Disk Read Latency Threshold	Threshold of the average time of write of data to the disk in milliseconds. Horizon uses the configured threshold to calculate the Disk Load Index factor. You can set the value to any positive integer. By default, this setting is not considered for load balancing. The default value is 0.

Click **Next** to proceed to the next page of the configuration wizard.

7 In the **Provisioning Settings** page of the wizard, configure the required settings.

Setting	Description
Enable Provisioning	Select this check box to enable provisioning after you finish this wizard. This box is checked by default.
Stop Provisioning on Error	Select this check box to stop provisioning when a provisioning error occurs. This box is checked by default.
Naming Pattern	Specify a prefix or a name format. Horizon will append or insert an automatically generated number starting with 1 to form the machine name. If you want the number at the end, simply specify a prefix. Otherwise, specify **{n}** anywhere in a character string and **{n}** will be replaced by the number. You can also specify **{n:fixed=<number of digits>}**, where **fixed=<number of digits>** indicates the number of digits to be used for the number. For example, specify **vm-{n:fixed=3}-sales** and the machine names will be vm-001-sales, vm-002-sales, and so on. **Note** Each machine name, including the automatically generated number, has a 15-character limit.
Maximum Machines	The number of machines to be provisioned.
Minimum Number of Ready (Provisioned) Machines during Instant Clone Maintenance Operations	This setting lets you keep the specified number of machines available to accept connection requests while Connection Server performs maintenance operations on the machines in the farm. This setting is not honored if you schedule immediate maintenance.

Click **Next** to proceed to the next page of the configuration wizard.

8 In the **vCenter Settings** page of the wizard, configure the required settings.

Setting	Description
Parent VM in vCenter	Click **Browse**, select **Linux** for Operating System, and select a golden-image virtual machine from the list.
Snapshot	Click **Browse** and select the snapshot of the golden-image virtual machine to use as the base image for the farm. Do not delete the snapshot and golden-image virtual machine from vCenter Server, unless no instant clones in the farm use the default image, and no more instant clones will be created from this default image. The system requires the golden-image virtual machine and snapshot to provision new instant clones in the farm, according to farm policies. The golden-image virtual machine and snapshot are also required for Connection Server maintenance operations.
VM Folder Location	Click **Browse** and select the folder in vCenter Server in which the farm resides.
Cluster	Click **Browse** and select the ESXi host or cluster on which the desktop virtual machines run. For the maximum limit on the cluster, see the VMware Knowledge Base (KB) article on Sizing Limits and Recommendations.
Resource Pool	Click **Browse** and select the vCenter Server resource pool in which the farm resides.
Datastores	Click **Browse** and select one or more datastores on which to store the farm. A table on the **Select Instant Clone Datastores** screen provides high-level guidelines for estimating the farm's storage requirements. These guidelines can help you determine which datastores are large enough to store the instant clones. The Storage Overcommit value is always set to Unbounded and is not configurable. **Note** If you use vSAN, there is only one datastore.

Setting	Description
Replica Disk Datastores	Select one or more replica disk datastores on which to store the instant-clones. This setting appears if you selected **Use Separate Datastores for Replica and OS Disks** in the **Storage Optimization** page of the farm configuration wizard. A table on the **Select Replica Disk Datastores** screen provides high-level guidelines for estimate the farm's storage requirements. These guidelines can help you determine which replica disk datastores are large enough to store the instant clones.
Network	Click **Browse** and select the networks to use for the instant-clone farm. You can select multiple vLAN networks to create a larger instant-clone farm. This setting uses the network type from the current golden image configured in vSphere Client and displays networks based on the network type of the parent VM: DVS, NSX-t, and Standard. You can use the same network as the parent VM or select a network from the list of available options. Networks are filtered based on the parent VM network type available in the selected cluster. The **Select Networks** screen provides a list of networks based on the parent VM network type available in the selected cluster. To use multiple networks, you must unselect **Use network from current parent VM image** and then select the networks to use with the instant-clone farm. Use the Filter box to show or hide specific network types. The screen displays error messages for the following incompatible networks: ■ **vmcNetworks**. This network belongs to VMC internal network ■ **dvsUplinkPort**. Cannot use network because it does not meet the naming standards for a virtual switch uplink port. ■ **notConfiguredOnAllHosts**. Cannot use network because it is not configured on all hosts in the cluster. The screen does not list the Standard network type for selection. Therefore, if the parent VM network type is Standard, then you must select **Use network from current parent VM image**. The screen also provides the list of ports and port bindings that are available to use: static (early binding) and ephemeral. Instant clones only support static port group types and ephemeral port group types are dimmed and listed as incompatible. All selected NSX-t network segments must be the same size, such as all /24 networks. Unequally sized segments can result in provisioning errors.

Click **Next** to proceed to the next page of the configuration wizard.

9 In the **Guest Customization** page of the wizard, configure the required settings.

Setting	Description
Domain	Select the Active Directory domain and user name. Connection Server requires certain user privileges to configure the farm. The domain and user account are used by ClonePrep to customize the instant-clone machines. You specify this user when you configure Connection Server settings for vCenter Server. You can specify multiple domains and users when you configure Connection Server settings. In this farm configuration wizard, you must select one domain and user from the list.
AD container	Provide the relative distinguished name for the Active Directory container. For example: `CN=Computers` You can click **Browse** to search your Active Directory tree for the container. You can also cut, copy, or paste in the container name.
Allow Reuse of Existing Computer Accounts	Select this option to use existing computer accounts in Active Directory when the virtual machine names of new instant clones match the existing computer account names. When an instant clone is created, if an existing AD computer account name matches the instant-clone virtual machine name, Horizon uses the existing computer account. Otherwise, a new computer account is created. The existing computer accounts must be located in the Active Directory container that you specify with the AD container setting. When this option is disabled, a new AD computer account is created when Horizon creates an instant clone. This option is disabled by default.

Setting	Description
Image Publish Computer Account	Publishing instant clones requires an additional computer account in the same AD domain as the clones. If you want to use pre-created computer accounts instead of auto-created computer accounts, you must also create the additional computer account and specify its name here. Then you do not need to delegate Create and Delete of computer objects to the provisioning account.
Use ClonePrep	Provide a ClonePrep customization specification to customize the virtual machines. ■ **Power-Off Script Name**. Name of the customization script that ClonePrep runs on instant-clone machines before they are powered off. Provide the path to the script on the golden-image virtual machine. ■ **Power-Off Script Parameters**. Provide parameters that ClonePrep can use to run a customization script on instant-clone machines before they are powered off. For example, use p1. ■ **Post-Synchronization Script Name**. Name of the customization script that ClonePrep runs on instant-clone machines after they are created or an image has been pushed to them. Provide the path to the script on the golden-image virtual machine. ■ **Post-Synchronization Script Parameters**. Provide parameters for the script that ClonePrep runs on instant-clone machines after they are created or an image has been pushed to them. For example, use p2.

Click **Next** to proceed to the next page of the configuration wizard.

10 In the **Ready to Complete** page of the wizard, review the settings and click **Submit** to create the automated instant-clone farm.

What to do next

Create a published application pool or a published desktop pool. See one of the following topics:

- Create a Linux Published Desktop Pool
- Create a Linux Published Application Pool Manually
- Create a Linux Published Application Pool from a List of Installed Applications

Create a Linux Published Desktop Pool

You can create a published desktop pool based on either a manual farm or automated instant-clone farm of multi-session Linux host machines. Each published desktop can support multiple user sessions at the same time.

A published desktop pool and a published desktop have the following characteristics:

- Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications.
- A published desktop pool is associated with a farm, which is a group of Linux host machines that have been enabled for multi-session mode. The farm can be an automated instant-clone farm or a manual farm.
- Horizon provides load balancing of the host machines in a farm by directing connection requests to the host that has the least number of active sessions.

Prerequisites

Create a farm of Linux host machines. See Create a Manual Farm of Linux Virtual Machines or Create an Automated Instant-Clone Farm of Linux Hosts.

Procedure

1. In Horizon Console, select **Inventory > Desktops**. Then click **Add**.

   The pool configuration wizard appears. As you advance through the wizard, you can go directly back to any prior page by clicking the page name in the navigation pane.

2. In the **Type** page of the wizard, select **RDS Desktop Pool** and then click **Next**.

3. In the **Desktop Pool ID** page of the wizard, provide a pool ID, display name, and description.

   The pool ID is the unique name that identifies the pool in Horizon Console. The display name is the name of the published desktop pool that users see when they log in to Horizon Client. If you do not specify a display name, it will be the same as the pool ID.

   Click **Next** to proceed to the next page of the configuration wizard.

4 In the **Desktop Pool Settings** page of the wizard, configure the required settings.

Setting	Description
State	■ **Enabled**: After being created, the desktop pool is enabled and ready for immediate use. ■ **Disabled**: After being created, the desktop pool is disabled and unavailable for use, and provisioning is stopped for the pool. This is an appropriate setting if you want to conduct post-deployment activities such as testing or other forms of baseline maintenance. When this state is in effect, remote desktops are unavailable for use. The default value is **Enabled**.
Connection Server Restrictions	You can restrict access to the desktop pool from certain Connection Servers by clicking **Browse** and selecting one or more Connection Servers. The default value is **No Restrictions**. If you intend to provide access to desktops through VMware Workspace ONE Access and you configure Connection Server restrictions, the VMware Workspace ONE Access application might display desktops to users when those desktops are actually restricted. VMware Workspace ONE Access users will be unable to start these desktops.
Category Folder	You can specify the name of the category folder that contains a Start menu shortcut for the desktop pool entitlement on Windows client devices. To configure this setting, click **Browse**. The default value is **Disabled**.
Client Restrictions	Select whether to restrict access to entitled desktop pools from certain client computers. You must add the names of the computers that are allowed to access the desktop pool in an Active Directory security group. You can select this security group when you add users or groups to the desktop pool entitlement. This setting is disabled by default.

Click **Next** to proceed to the next page of the configuration wizard.

5 In the **Select RDS Farms** page of the wizard, select an existing farm or create a farm for this pool. Then click **Next**.

6 In the **Ready to Complete** page of the wizard, review the settings and click **Submit** to create the published desktop pool.

What to do next

Entitle users to access the pool. For more information, see *Setting Up Published Desktops and Applications in Horizon*.

Create a Linux Published Application Pool Manually

You can manually specify the application that you want to publish as an application pool. If an application that you want to specify manually is not already installed, Horizon displays a warning message. The published application can support multiple user sessions at the same time.

Prerequisites

- Install the application on all the Linux host machines in the base farm.
- Create a farm of the Linux host machines. See Create a Manual Farm of Linux Virtual Machines or Create an Automated Instant-Clone Farm of Linux Hosts.

  Note Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications. vGPU capabilities are not supported for published applications.

Procedure

1. In Horizon Console, select **Inventory > Applications**.
2. Select **Add > Add Manually**.

   The pool configuration dialog box appears.
3. Configure the required settings.

Setting	Description
Application Pool Type	Select **RDS Farm** and then select a Linux-based farm from the drop-down menu. **Note** To create a Linux application pool, you must base the pool on a farm of Linux host machines that are enabled for multi-session capabilities. Horizon does not support the creation of Linux application pools from a virtual desktop pool of single-session Linux machines.
ID	Unique name that identifies the pool in Horizon Console. This field is required.
Display Name	Pool name that users see when they log in to Horizon Client. If you do not specify a display name, it will be the same as **ID**.
Access Group	This is a read-only field. You must specify the access group when you create the farm on which the application pool is based.
Version	Version of the application.
Publisher	Publisher of the application.
Path	Full pathname of the application. For example, `/usr/bin/gnome-terminal`. This field is required.

Setting	Description
Start Folder	Full pathname of the starting directory for the application.
Parameters	Parameters to pass to the application when it starts. For example, you can specify `–username user1 –loglevel 3`.
Description	Description of this application pool.
Pre-Launch	Select this option to configure an application so that an application session is launched before a user opens the application in Horizon Client. When a published application is launched, the application opens more quickly in Horizon Client. If you enable this option, the configured application session is launched before a user opens the application in Horizon Client, regardless of how the user connects to the server from Horizon Client. **Note** Application sessions can be disconnected when the **Pre-launch Session Timeout (Applications Only)** option is set when you add or edit the Linux-based farm.
Connection Server Restrictions	You can restrict access to the application pool from certain Connection Servers by clicking **Browse** and selecting one or more Connection Servers. The default value is **No Restrictions**. If you intend to provide access to published applications through VMware Workspace ONE Access and you configure Connection Server restrictions, VMware Workspace ONE Access application might display applications to users when those applications are actually restricted. VMware Workspace ONE Access users will be unable to start these applications.
Category Folder	You can specify the name of the category folder that contains a Start menu shortcut for the application pool entitlement on Windows client devices. To configure this setting, click **Browse**. The default value is **Disabled**.
Client Restrictions	Select whether to restrict access to entitled application pools from certain client computers. You must add the names of the computers that are allowed to access the application pool in an Active Directory security group. You can select this security group when you add users or groups to the application pool entitlement. This setting is disabled by default.

Setting	Description
Entitle Users After Adding Pool	Select this option to open the **Add Entitlements** dialog box automatically after the pool is created.
Multi-Session Mode	Select **Disabled**. **Note** Linux application pools do not support enabling multi-session mode. If you specify a setting to enable multi-session mode, that setting has no effect. Linux published applications can only be configured in single-session mode. For example, if a user opens a published application on client A and then opens the same published application or another published application based on the same farm on client B, then the session on client A is disconnected and reconnected on client B.

4 Click **Submit** to create the published application pool.

What to do next

Entitle users to access the pool. You can also view the number of entitled users that are using a published application in the **User Count** column in the application pools page.

If you must ensure that Connection Server starts the application only on Linux host machines that have sufficient resources to run the application, configure an anti-affinity rule for the application pool. For more information, see *Setting Up Published Desktops and Applications in Horizon*.

Create a Linux Published Application Pool from a List of Installed Applications

When you create an application pool from a list of installed applications, Horizon automatically displays the applications that are available on all the Linux host machines in the farm. You can publish one or more applications from the list as an application pool. Each published application can support multiple user sessions at the same time.

Prerequisites

- Install the application on all the Linux host machines in the base farm.
- Create a farm of the Linux host machines. See Create a Manual Farm of Linux Virtual Machines or Create an Automated Instant-Clone Farm of Linux Hosts.

  Note Only virtual machines running RHEL Workstation 7.8, 7.9, 8.1, 8.2, 8.3, or 8.4 or Ubuntu 18.04/20.04 can be configured as multi-session hosts for published desktops and published applications. vGPU capabilities are not supported for published applications.

Procedure

1 In Horizon Console, select **Inventory > Applications**.

2 Select **Add > Add from Installed Applications**.

The pool configuration wizard appears.

3 In the **Select Applications** page of the wizard, configure the required settings.

Setting	Description
Application Pool Type	Select **RDS Farm** and then select a Linux-based farm from the drop-down menu. **Note** To create a Linux application pool, you must base the pool on a farm of Linux host machines that are enabled for multi-session capabilities. Horizon does not support the creation of Linux application pools from a virtual desktop pool of single-session Linux machines.
Select installed applications	From the list of installed applications, select the application that you want to publish. If you select multiple applications from the list, a separate application pool is created for each application.
Pre-Launch	Select this option to configure an application so that an application session is launched before a user opens the application in Horizon Client. When a published application is launched, the application opens more quickly in Horizon Client. If you enable this option, the configured application session is launched before a user opens the application in Horizon Client, regardless of how the user connects to the server from Horizon Client. **Note** Application sessions can be disconnected when the **Pre-launch Session Timeout (Applications Only)** option is set when you add or edit the Linux-based farm.
Connection Server Restrictions	You can restrict access to the application pool from certain Connection Servers by clicking **Browse** and selecting one or more Connection Servers. The default value is **No Restrictions**. If you intend to provide access to published applications through VMware Workspace ONE Access and you configure Connection Server restrictions, VMware Workspace ONE Access application might display applications to users when those applications are actually restricted. VMware Workspace ONE Access users cannot start these applications.
Category Folder	You can specify the name of the category folder that contains a Start menu shortcut for the application pool entitlement on Windows client devices. To configure this setting, click **Browse**. The default value is **Disabled**.

Setting	Description
Client Restrictions	Select whether to restrict access to entitled application pools from certain client computers. You must add the names of the computers that are allowed to access the application pool in an Active Directory security group. You can select this security group when you add users or groups to the application pool entitlement. This setting is disabled by default.
Entitle Users After Adding Pool	Select this option to open the **Add Entitlements** dialog box automatically after the pool is created.
Multi-Session Mode	Select **Disabled**. **Note** Linux application pools do not support enabling multi-session mode. If you specify a setting to enable multi-session mode, that setting has no effect. Linux published applications can only be configured in single-session mode. For example, if a user opens a published application on client A and then opens the same published application or another published application based on the same farm on client B, then the session on client A is disconnected and reconnected on client B.

Click **Next** to proceed to the next page of the configuration wizard.

4 In the **Edit Applications** page of the wizard, optionally edit the following settings.

Setting	Description
ID	Unique name that identifies the pool in Horizon Console.
Display Name	Pool name that users see when they log in to Horizon Client. If you do not specify a display name, it will be the same as **ID**.

5 Click **Submit** to create the published application pool.

What to do next

Entitle users to access the pool. You can also view the number of entitled users that are using a published application in the **User Count** column in the application pools page.

If you must ensure that Connection Server starts the application only on Linux host machines that have sufficient resources to run the application, configure an anti-affinity rule for the application pool. For more information, see *Setting Up Published Desktops and Applications in Horizon*.

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
- Collect Diagnostic Information for a Linux Virtual Machine
- Horizon Agent Fails to Disconnect on an iPad Pro Horizon Client
- SSO Fails to Connect to a PowerOff Agent
- Unreachable VM After Creating a Manual Desktop Pool for Linux

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 VMware Horizon 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 VMware Horizon. To verify that you have the correct license, see "Change the Product License Key or License Modes in Horizon Console" in *Horizon Administration*.
- An event database to store information about VMware Horizon components. For more information about configuring an event database, see the *Horizon Installation* 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 "Privileges for Help Desk Tool Tasks" in *Horizon Administration*.
- 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.

Table 9-3. Activities (continued)

Option	Description
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 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.

Table 9-4. VM Details (continued)

Option	Description
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 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 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

To avoid scrolling through the list of session processes, you can search for a session process by name by entering the process name in the search filter text box.

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.
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 a Linux Virtual Machine

You can collect diagnostic information to help VMware Technical Support diagnose and resolve issues with a Linux virtual machine that you are using to provision remote Horizon resources. 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
```

Results

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

```
ubuntu-18-vdm-sdct-20190201-0606-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.

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.