

# 设置桌面 Horizon 7 for Linux

VMware Horizon 7 7.0



vmware®

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

如果您对本文档有任何意见或建议, 请将反馈信息发送至:

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**威睿信息技术(中国)有限公司**  
北京办公室  
北京市  
朝阳区新源南路 8 号  
启皓北京东塔 8 层 801  
[www.vmware.com/cn](http://www.vmware.com/cn)

上海办公室  
上海市  
淮海中路 333 号  
瑞安大厦 804-809 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

广州办公室  
广州市  
天河路 385 号  
太古汇一座 3502 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

版权所有 © 2019 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

# 目录

## 设置 Horizon 7 for Linux 桌面 5

### 1 功能和系统要求 6

Horizon 7 Linux 桌面和桌面池的功能 6

Horizon 7 for Linux 桌面的配置步骤概述 6

Horizon 7 for Linux 的系统要求 7

### 2 为桌面部署准备 Linux 虚拟机 12

创建虚拟机并安装 Linux 12

为远程桌面部署准备 Linux 计算机 13

安装 Horizon Agent 的依赖项包 15

### 3 为 Linux 桌面设置 Active Directory 集成 17

将 Linux 与 Active Directory 集成 17

设置单点登录和智能卡重定向 18

### 4 Linux 桌面的配置选项 20

在 Linux 桌面上的配置文件中设置选项 20

禁止在 vSphere 控制台中显示 Linux 桌面 22

### 5 为 Linux 桌面设置图形 23

为 vGPU 配置 RHEL 6.6 和 RHEL 7.1 23

为 ESXi 主机上的 NVIDIA 显卡安装 VIB 23

在 Linux 虚拟机上为 vGPU 配置共享 PCI 设备 25

安装 NVIDIA 显卡驱动程序 26

验证是否安装了 NVIDIA 显卡驱动程序 28

配置 RHEL 6.6 以支持 vDGA 29

在主机上为 NVIDIA GRID 启用 DirectPath I/O 29

向 RHEL 6.6 虚拟机添加 vDGA 直通设备 30

安装 NVIDIA 显卡驱动程序 31

验证是否安装了 NVIDIA 显卡驱动程序 33

配置 RHEL 7.1 以支持 vSGA 34

为 ESXi 主机上的 NVIDIA 显卡安装 VIB 34

在 Linux 虚拟机上为 vSGA 配置 3D 功能 35

验证是否正在 Linux 虚拟机上运行 vSGA 36

### 6 安装 Horizon Agent 和管理 Linux 桌面 38

在 Linux 虚拟机上安装 Horizon Agent	38
install_viewagent.sh 命令行选项	40
启用可逆密码加密	42
配置 Linux 代理的证书	42
创建包含 Linux 虚拟机的桌面池	42
在 Linux 虚拟机上升级 Horizon Agent	43
卸载和重新安装 Horizon 7 for Linux 计算机	44
从手动桌面池中移除未受管计算机	44
从 View 中移除已注册的计算机	45
在 Linux 虚拟机上卸载 Horizon Agent	45
在 Linux 虚拟机上重新安装 Horizon Agent	46
如何在 Linux 桌面上从 vSphere 执行电源操作	47
收集有关 Horizon 7 for Linux 软件的信息	47

## 7 批量部署 Horizon 7 for Linux 桌面 49

Linux 桌面批量部署概览	49
为克隆 Linux 桌面计算机创建虚拟机模板	51
用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件	52
用于克隆 Linux 虚拟机的示例脚本	53
用于将克隆的虚拟机加入 AD 域的示例脚本	57
使用 SSH 将克隆的虚拟机加入 Active Directory 域的示例脚本	60
用于在 Linux 虚拟机上安装 Horizon Agent 的示例脚本	63
使用 SSH 在 Linux 虚拟机上安装 Horizon Agent 的示例脚本	68
用于将配置文件上载至 Linux 虚拟机的示例脚本	74
使用 SSH 将配置文件上载至 Linux 虚拟机的示例脚本	77
用于在 Linux 桌面计算机上升级 Horizon Agent 的示例脚本	81
使用 SSH 在 Linux 虚拟机上升级 Horizon Agent 的示例脚本	86
用于在 Linux 虚拟机上执行操作的示例脚本	91
用于从连接服务器 LDAP 数据库中删除计算机的示例脚本	95

## 8 对 Linux 桌面进行故障排除 97

收集 Horizon 7 for Linux 计算机的诊断信息	97
对 Linux 计算机的 Horizon Agent 注册失败问题进行故障排除	97
对无法访问 Linux 计算机上的 Horizon Agent 问题进行故障排除	98
对 Linux 计算机上的 Horizon Agent 没有响应问题进行故障排除	100
对远程桌面和客户端主机之间的复制和粘贴问题进行故障排除	101
配置 Linux 防火墙以允许传入的 TCP 连接	101

# 设置 Horizon 7 for Linux 桌面

《设置 Horizon 7 for Linux 桌面》文档提供了有关设置 Linux 虚拟机以用作 VMware Horizon 7™ 桌面的信息，其中包括准备 Linux 客户机操作系统，在虚拟机上安装 Horizon Agent，以及在 View Administrator 中配置计算机以用于 Horizon 7 部署。

## 目标读者

本文档中的信息适用于希望配置并使用在 Linux 客户机操作系统上运行的远程桌面的任何人。本文档中的信息专门为已熟练掌握虚拟机技术和数据中心操作、并具有丰富经验的 Linux 系统管理员编写。

# 功能和系统要求

使用 Horizon 6 或更高版本，用户可以连接至运行 Linux 操作系统的远程桌面。

本章讨论了以下主题：

- [Horizon 7 Linux 桌面和桌面池的功能](#)
- [Horizon 7 for Linux 桌面的配置步骤概述](#)
- [Horizon 7 for Linux 的系统要求](#)

## Horizon 7 Linux 桌面和桌面池的功能

Horizon 7 引入了多种适用于 Linux 桌面的新功能。以下列表显示了主要功能。

<b>SLED 11 SP3/SP4</b>	Horizon 7 支持 SLED 11 SP3/SP4 版本。但不支持单点登录。
<b>HTML Access</b>	Horizon 7 支持 Chrome 上的 HTML Access 4.0.0。
<b>依赖项包</b>	Horizon Agent for Linux 需要 Linux 分发包特有的依赖项包。在安装 Horizon Agent for Linux 之前，必须先安装这些依赖项包。

## Horizon 7 for Linux 桌面的配置步骤概述

在安装和配置 Horizon 7 for Linux 桌面时，您必须按照一组不同的步骤进行操作，具体取决于在虚拟机上安装 2D 图形还是 3D 图形。

### 2D 图形 - 配置步骤概述

对于 2D 图形，请执行以下步骤：

- 1 查看设置 Horizon 7 for Linux 部署的系统要求。请参阅 [Horizon 7 for Linux 的系统要求](#)。
- 2 在 vSphere 中创建一个虚拟机并安装 Linux 操作系统。请参阅[创建虚拟机并安装 Linux](#)。
- 3 准备客户机操作系统以在 Horizon 7 环境中部署为桌面。请参阅[为远程桌面部署准备 Linux 计算机](#)。
- 4 配置 Linux 客户机操作系统以使用 Active Directory 进行身份验证。该步骤是根据您的环境中的要求使用第三方软件执行的，本指南未介绍该步骤。
- 5 在 Linux 虚拟机上安装 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。
- 6 创建一个桌面池以包含配置的 Linux 虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

## 3D 图形 - 配置步骤概述

您必须先在 Linux 虚拟机上完成 NVIDIA GRID vGPU、vDGA 或 vSGA 配置，然后再在计算机上安装 Horizon Agent，并在 View Administrator 中部署桌面池。

- 1 查看设置 Horizon 7 for Linux 部署的系统要求。请参阅 [Horizon 7 for Linux 的系统要求](#)。
- 2 在 vSphere 中创建一个虚拟机并安装 Linux 操作系统。请参阅[创建虚拟机并安装 Linux](#)。
- 3 准备客户机操作系统以在 Horizon 7 环境中部署为桌面。请参阅[为远程桌面部署准备 Linux 计算机](#)。
- 4 配置 Linux 客户机操作系统以使用 Active Directory 进行身份验证。该步骤是根据您的环境中的要求使用第三方软件执行的，本指南未介绍该步骤。
- 5 在 ESXi 主机和 Linux 虚拟机上配置 3D 功能。按照要安装的 3D 功能的步骤进行操作。
  - 请参阅[为 vGPU 配置 RHEL 6.6 和 RHEL 7.1](#)。
  - 请参阅[配置 RHEL 6.6 以支持 vDGA](#)。
  - 请参阅[配置 RHEL 7.1 以支持 vSGA](#)。
- 6 在 Linux 虚拟机上安装 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。
- 7 创建一个桌面池以包含配置的 Linux 虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

## 批量部署

使用 View Administrator，您只能在手动桌面池中部署 Linux 虚拟机。使用 vSphere PowerCLI，您可以开发用于自动部署 Linux 桌面计算机池的脚本。请参阅[第 7 章 批量部署 Horizon 7 for Linux 桌面](#)。

## Horizon 7 for Linux 的系统要求

Horizon 7 for Linux 必须满足特定的操作系统、Horizon 7 和 vSphere 平台要求。

## Horizon Agent 支持的 Linux 版本

下表列出了桌面池中的虚拟机上支持的 Linux 操作系统。

**表 1-1. View Agent 支持的 Linux 操作系统**

Linux 分发版	架构
Ubuntu 14.04	x64
<b>注</b> 禁用 Compiz 可避免性能降低。请参阅 <a href="http://kb.vmware.com/kb/2114809">http://kb.vmware.com/kb/2114809</a> 了解更多信息。	
Ubuntu 12.04	x64
RHEL 6.6 和 7.1	x64
CentOS 6.6 和 7.1	x64
NeoKylin 6 Update 1	x64

Linux 分发包	架构
SLED 11 SP3/SP4	x64
仅支持安装和连接 SLED 12。	x64

**注** Linux 代理在某些 Linux 分发包中具有依赖项包。请参阅[安装 Horizon Agent 的依赖项包](#)了解更多信息。

## 所需的平台和 Horizon 7 软件版本

要安装并使用 Horizon 7 for Linux，您的部署必须满足特定的 vSphere 平台、Horizon 7 和客户端要求。

**表 1-2. 所需的平台和 Horizon 7 软件版本**

平台和软件	支持的版本
vSphere 平台版本	vSphere 5.5 U3、vSphere 6.0 U2 或更高版本。
Horizon 环境	Horizon 6 版本 6.2.1 或 Horizon 7 版本 7.0.0。
Horizon Client 软件	<ul style="list-style-type: none"> <li>■ 适用于 Windows 的 Horizon Client 4.0.0</li> <li>■ 适用于 Linux 的 Horizon Client 4.0.0</li> <li>■ 适用于 Mac OS X 的 Horizon Client 4.0.0</li> <li>■ Chrome 上的 HTML Access 4.0.0</li> <li>■ 不支持零客户端和移动客户端</li> </ul>

**注** HTML Access 不支持音频输出。

## Linux 虚拟机使用的 TCP 端口

View Agent 和 Horizon Client 使用 TCP 端口进行相互之间以及它们与各种 View Server 组件之间的网络访问。

**表 1-3. Linux 虚拟机使用的 TCP 端口**

源	端口	目标	端口	协议	说明
Horizon Client	*	Linux 代理	22443	TCP	如果未使用 Blast 安全网关，则为 Blast
安全服务器、View 连接服务器或 Access Point 设备	*	Linux 代理	22443	TCP	如果使用了 Blast 安全网关，则为 Blast
View Agent	*	View 连接服务器	4001、4002	TCP	JMS SSL 流量。

**注** 有关客户端使用的 TCP 和 UDP 端口的详细信息，请参阅 View 文档中《Horizon Client 和 View Agent 安全指南》的“客户端和 View Agent 使用的 TCP 和 UDP 端口”部分。

## 验证 Linux 虚拟机使用的 Linux 帐户

下表列出了 Linux 虚拟机使用的帐户名称和帐户类型。



表 1-4. 帐户名称和帐户类型

帐户名称	帐户类型	使用者
根	Linux 操作系统内置	Java Standalone Agent、 mksvchanserver、Shell 脚本
vmwblast	由 Linux 代理安装程序创建	VMwareBlastServer
<当前登录用户>	Linux 操作系统内置或 AD 用户或 LDAP 用户	python 脚本

## 建议的显存 (vRAM) 设置

在 vSphere Client 中创建 Linux 虚拟机时，请按照表 1-5. 2D 或 vSGA 图形的推荐 vRAM 设置 所示配置 vRAM 大小。针对您为虚拟机配置的显示器数量和分辨率设置建议的 vRAM 大小。

这些 vRAM 大小建议仅适用于配置为使用 2D 或 vSGA 图形的虚拟机（使用 VMware 驱动程序）。vDGA 和 NVIDIA GRID vGPU 使用 NVIDIA 驱动程序。在 vSphere Client 中设置的 vRAM 大小对 vDGA 或 NVIDIA GRID vGPU 计算机无效。

这些 vRAM 大小是最小的建议值。如果虚拟机上有更多资源可供使用，请将 vRAM 设置为更大的值以提高视频性能。

对于在最低分辨率下配置了单个显示器的虚拟机，10 MB 是建议的最低 vRAM 大小。

如果为 Linux 虚拟机配置的 vRAM 大小比建议值小，则可能会出现以下问题：

- 在最初建立连接后，可能会立即断开桌面会话。
- 自动适应可能无法正常工作。将在较小的屏幕区域中显示桌面。

**小心** Horizon 7 不会在 Linux 虚拟机上自动配置 vRAM 设置，这与在 Windows 虚拟机上的情形一样。您必须在 vSphere Client 中手动配置 vRAM 设置。

表 1-5. 2D 或 vSGA 图形的推荐 vRAM 设置

vRAM 大小	显示器数量	最大分辨率
10 MB	1	1600x1200 或 1680x1050
12 MB	1	1920x1440
32 MB	1	2560x1600
48 MB	2	2048x1536
80 MB	2	2560x1600

vRAM 大小	显示器数量	最大分辨率
128 MB	3	2560x1600
128 MB	4	2560x1600

RHEL 和 CentOS 只在 vSphere 5.5 上支持此配置。

要在 Ubuntu 上支持此配置，您必须重新编译内核并禁用 3D。有关如何禁用 3D 的信息，请参阅 <http://kb.vmware.com/kb/2114809>。

NeoKylin 上不支持此配置。

**注** 要连接到配有多个显示器的 RHEL 6.6、CentOS 6.6 或者 SLED 11 Sp3 或 Sp4 桌面，您必须按[创建虚拟机并安装 Linux](#) 中所述来设置显示器的数量和显存大小。此外，在虚拟机 (VM) 电源关闭后，您必须编辑 `vmx` 文件，并根据显示器的数量和方向（横向或纵向）设置 `svga.maxWidth` 和 `svga.maxHeight`。您还必须将 `svga.autodetect` 设置为 `false`。然后打开虚拟机电源。一般而言，`svga.maxWidth` 和 `svga.maxHeight` 值必须足够大才能支持所有显示器。要以最大分辨率 2560x1600 支持 4 个显示器，请使用[编辑虚拟机设置 > 选项 > 常规 > 配置 > 参数 > 添加行](#)添加以下参数。您无需修改参数 `svga.vramSize`。

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

如果有多个显示器，您必须设置这些参数。否则，可能遇到以下一个或多个问题：

- 只有一个显示器正常显示。其他显示器可能黑屏或显示与其他显示器相同的内容。
- 一个按键显示多次。
- 桌面速度变得非常缓慢。

如果您在使用建议的设置时遇到自动适应问题，可以指定更大的 vRAM 大小。vSphere Client 允许的最大 vRAM 大小为 128 MB。如果您指定的大小超过 128 MB，则必须手动修改 `vmx` 文件。以下示例指定的 vRAM 大小为 256 MB：

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

**注** 要为将 2D 呈现或 3D 呈现与 vSGA 一起使用的 RHEL 7.1 计算机配置四个显示器，请将每个显示器的最大分辨率设置为 2048x1536。要配置 RHEL 7.1 计算机以使用最多三个显示器，请将每个显示器的最大分辨率设置为 2560x1600。

要提高 vSGA 多显示器环境中的 RHEL 7.1 计算机的性能，请将该虚拟机的 **3D 内存** 设置设为 1 GB 或更大并为该计算机配置 4 个 vCPU。如果在 RHEL 7.1 计算机上配置四个具有 2048x1536 分辨率的显示器，请将**内存**设置为 4 GB 并为该计算机配置 4 个 vCPU。

## 可支持多个显示器的建议 vCPU 和共享内存设置

要提升配有多个显示器的桌面的性能，请为 Linux 虚拟机配置至少两个 vCPU。

此外，请确保虚拟机有足够的共享内存来支持多个显示器。使用以下命令来确定当前最大的共享内存大小 (shmmax):

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

如果共享内存较小，请使用以下命令增加最大大小：

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

## 针对视频回放的建议 vCPU 设置

对于未使用 vDGA 图形的 Linux 虚拟机，如果配置的 vCPU 数量过少，则在高分辨率显示器上的视频回放效果可能不佳。配置更多 vCPU（例如，四个 vCPU）可提高视频回放的性能。

## vDGA 和 NVIDIA GRID vGPU 支持的显示器分辨率

配置为使用 vDGA 或 NVIDIA GRID vGPU 的虚拟机最多可以支持四个最大分辨率为 2560x1600 的显示器。

# 为桌面部署准备 Linux 虚拟机

设置 Linux 桌面需要创建 Linux 虚拟机，并准备操作系统以进行远程桌面部署。

本章讨论了以下主题：

- 创建虚拟机并安装 Linux
- 为远程桌面部署准备 Linux 计算机
- 安装 Horizon Agent 的依赖项包

## 创建虚拟机并安装 Linux

您可以在 vCenter Server 中为在 Horizon 7 中部署的每个远程桌面创建新虚拟机。您必须在虚拟机上安装 Linux 分发包。

### 前提条件

- 确认您的部署满足支持 Linux 桌面的相关要求。请参阅 [Horizon 7 for Linux 的系统要求](#)。
- 熟悉有关在 vCenter Server 中创建虚拟机和安装客户机操作系统的步骤。请参阅《在 View 中设置桌面和应用程序池》文档中的“创建和准备虚拟机”。
- 熟悉您将用于虚拟机的监视器的建议显存 (vRAM) 值。请参阅 [Horizon 7 for Linux 的系统要求](#)。

### 步骤

1 在 vSphere Web Client 或 vSphere Client 中，创建新虚拟机。

2 配置自定义配置选项。

- a 右键单击虚拟机，然后单击**编辑设置**。
- b 指定 vCPU 数量和显存大小。

有关建议的值，请遵循您的 Linux 分发包安装指南中的准则。

例如，Ubuntu 12.04 建议配置 2048 MB 的显存和 2 个 vCPU。

- c 选择**视频卡**并指定显示器数量和显存 (vRAM) 总大小。

对于使用 2D 或 vSGA 的虚拟机（使用 VMware 驱动程序），请在 vSphere Web Client 中设置 vRAM 大小。vRAM 大小对 vDGA 或 NVIDIA GRID vGPU 计算机（使用 NVIDIA 驱动程序）无效。

有关建议的值，请遵循《Horizon 7 for Linux 的系统要求》中的准则。请勿使用显存计算器。

- 3 打开虚拟机的电源并安装 Linux 分发版。
- 4 将虚拟机配置为 **gnome** 桌面环境。

虽然可以在某些分发版（例如 **Kubuntu**）上进行基本连接以及使用音频和视频，但 **KDE** 仍尚未通过支持 **Horizon Agent** 的认证。

- 5 确保系统主机名可解析为 **127.0.0.1**。

## 为远程桌面部署准备 Linux 计算机

要准备 Linux 计算机以使其在 **Horizon 7** 部署中用作桌面，您必须执行特定的任务。

Linux 计算机必须可以与连接服务器通信，然后 **Horizon 7** 才可以管理此计算机。您必须在 Linux 计算机上配置网络，以便 Linux 计算机可以使用其完全限定域名 (FQDN) 对连接服务器实例执行 **ping** 操作。

**Open VMware Tools (OVT)** 预安装在 **RHEL 7**、**CentOS 7** 和 **SLED 12** 计算机上。如果准备其中任何一种计算机以用作远程桌面，您可以在以下过程中跳过步骤 1 至 5，该过程介绍了如何通过手动运行安装程序来安装 **VMware Tools**。

### 前提条件

- 确认已在 **vCenter Server** 中创建了新虚拟机 (VM)，并且已在虚拟机上安装了您的 Linux 分发版
- 熟悉有关在 Linux 虚拟机上挂载和安装 **VMware Tools** 的步骤。请参阅《**vSphere 虚拟机管理**》文档中的“在 Linux 虚拟机中手动安装或升级 **VMware Tools**”。
- 熟悉有关将 Linux 虚拟机配置为可通过 **DNS** 解析的步骤。这些步骤因不同的 Linux 分发版和版本而异。有关说明，请参考您的 Linux 分发版和版本的文档。

### 步骤

- 1 在 **vSphere Web Client** 或 **vSphere Client** 中，将 **VMware Tools** 虚拟磁盘挂载到虚拟机上。
- 2 右键单击 **VMware Tools** 安装程序文件 **VMwareTools-x.x.x-xxxx.tar.gz**，单击**提取到**，然后为您的 Linux 分发版选择桌面。

**vmware-tools-distrib** 文件夹将提取到桌面。

- 3 在虚拟机上，以 **root** 用户身份登录，然后打开终端窗口。
- 4 解压缩 **VMware Tools tar** 安装程序文件。

例如：

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

- 5 运行安装程序并配置 **VMware Tools**。

该命令在不同的 Linux 分发版中可能略有不同。例如：

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

通常情况下，安装程序文件运行完毕后会运行 `vmware-config-tools.pl` 配置文件。

- 6 对于 RHEL 和 CentOS，将计算机的主机名映射到 `/etc/hosts` 中的 127.0.0.1。

对于 Ubuntu，不必执行此步骤，因为已默认执行此映射。在批量部署桌面时也不必执行此步骤，因为克隆过程中会添加此映射。

- 7 对于 RHEL 7 和 CentOS 7，确认已禁用 `virbr0`。

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

- 8 确保容器中的 View 连接服务器实例可以通过 DNS 解析。

- 9 配置 Linux 计算机，以使默认运行级别为 5。

运行级别必须为 5，Linux 桌面才能正常工作。

- 10 在配置为使用 OpenLDAP 服务器进行身份验证的 Ubuntu 计算机上，在计算机上设置完全限定域名。

该步骤确保可以在 View Administrator 的“会话”页上的“用户”字段中正确显示该信息。按如下方式编辑 `/etc/hosts` 文件：

- a `# nano /etc/hosts`
- b 添加完全限定域名。例如：`127.0.0.1 hostname.domainname hostname.`
- c 退出并保存该文件。

- 11 编辑 `/etc/nsswitch.conf` 并修改以下 `hosts` 条目：

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

默认设置为 `hosts: files dns`，该设置可能会导致 Linux 桌面无法从临时的网络中断中恢复。桌面与 View 连接服务器的通信可能永久中断，需要重新引导来解决此问题。

- 12 对于 SUSE，禁用“通过 DHCP 更改主机名”。设置主机名或域名。

- a 在“Yast”中，单击**网络设置**。
- b 单击**主机名/DNS**选项卡。
- c 取消选择**通过 DHCP 更改主机名**。
- d 输入主机名和域名。
- e 单击**确定**。

在安装 VMware Tools 之后，如果您升级 Linux 内核，则 VMware Tools 可能会停止运行。要解决此问题，请参阅 <http://kb.vmware.com/kb/2050592>。

## 安装 Horizon Agent 的依赖项包

Horizon Agent for Linux 具有一些 Linux 分发包特有的依赖项包。在安装 Horizon Agent for Linux 之前，必须先安装这些依赖项包。

### 前提条件

确认已在 vCenter Server 中创建新虚拟机 (Virtual Machine, VM)，并且已在该虚拟机上安装 Linux 分发包。

### 步骤

- 1 安装默认情况下未安装或升级的必需依赖项包。如果任何依赖项包不满足要求，安装程序便会中断安装。

**表 2-1. 必需依赖项包**

Linux 分发包	依赖项包
RHEL 或 CentOS 7 升级 2D 桌面和 vSGA 桌面的 VMware 图形驱动程序。	<pre>yum install mesa-libxatracker xorg-x11-drv-vmware mesa-private-llvm mesa-dri-drivers</pre> <p>更新后，依赖项包应为以下版本或更高版本：</p> <ul style="list-style-type: none"> <li>■ xorg-x11-drv-vmware-13.0.2-7.20150211git8f0cf7c.el7.x86_64</li> <li>■ mesa-libxatracker-10.6.5-3.20150824.el7.x86_64</li> <li>■ mesa-private-llvm-3.6.2-2.el7.x86_64</li> <li>■ mesa-dri-drivers-10.6.5-3.20150824.el7.x86_64</li> </ul> <p><b>注</b> 此更新不适用于 CentOS 默认联机存储库。您必须启用 CentOS 持续发布 (Continuous Release, CR) 存储库。</p> <pre>yum-config-manager --enable "CentOS-7 - cr"</pre> <p>升级 VMware 图形驱动程序后，您可以再次禁用此存储库。</p> <pre>yum-config-manager --disable "CentOS-7 - cr"</pre>
SLED 11 SP3/SP4 将 xorg-x11-server 升级到高于 7.4.27.111.1 的版本	<pre>zypper install xorg-x11-server</pre>

Linux 分发	依赖项包
<b>SLED 12 SP0</b> 将 VMware 图形驱动程序升级到高于 13.1.0-5.2 的版本	<pre>zypper install xf86-video-vmware</pre> <p>如果 SLED 联机存储库中没有 xf86-video-vmware-13.1.0-5.2.x86_64 或更高版本，您必须启用 OpenSUSE 联机存储库进行安装。</p> <ol style="list-style-type: none"> <li>zypper addrepo <a href="http://download.opensuse.org/distribution/leap/42.1/repo/oss/">http://download.opensuse.org/distribution/leap/42.1/repo/oss/</a> oss-42.1</li> <li>zypper ref</li> <li>zypper install xf86-video-vmware</li> <li>zypper removerepo oss-42.1</li> </ol>
<b>Ubuntu 1404</b> 将 indicator-session 升级到 12.10.5+15.04.20150327（可在 <a href="https://launchpad.net/ubuntu/wily/amd64/indicator-session/12.10.5+15.04.20150327-0ubuntu1">https://launchpad.net/ubuntu/wily/amd64/indicator-session/12.10.5+15.04.20150327-0ubuntu1</a> 中获取）	<pre>wget http://launchpadlibrarian.net/201393830/indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb</pre> <pre>sudo dpkg -i ./indicator-session_12.10.5+15.04.20150327-0ubuntu1_amd64.deb</pre>

## 2 安装 Horizon Agent 的可选依赖项包。

剪贴板重定向功能依赖于 libexpat.so.1 和 libpng12.so.0。如果缺少任何库文件，安装程序便会显示一条警告消息，然后继续安装。但是，剪贴板重定向功能将无法正常使用。

**表 2-2. 可选依赖项包**

Linux 分发	依赖项包
RHEL 或 CentOS 6.6	<ul style="list-style-type: none"> <li>■ expat-2.0.1-11.el6_2.x86_64</li> <li>■ libpng-1.2.49-1.el6_2.x86_64</li> </ul>
RHEL 或 CentOS 7.1	<ul style="list-style-type: none"> <li>■ expat-2.1.0-8.el7.x86_64</li> <li>■ libpng12-1.2.50-6.el7.x86_64</li> </ul>
Ubuntu 1204/1404	<ul style="list-style-type: none"> <li>■ libexpat1</li> <li>■ libpng12-0</li> </ul>
SLED 11 SP3/SP4	<ul style="list-style-type: none"> <li>■ libexpat1-2.0.1-88.34.1</li> <li>■ libpng12-0-1.2.31-5.35.1</li> </ul>
SLED 12 SP0	<ul style="list-style-type: none"> <li>■ libexpat1-2.1.0-13.232.x86_64</li> <li>■ libpng12-0-1.2.50-10.1.x86_64</li> </ul>



# 为 Linux 桌面设置 Active Directory 集成

# 3

View 使用现有的 Microsoft Active Directory (AD) 基础架构来进行用户身份验证和管理。您可以将 Linux 桌面与 Active Directory 集成，以便用户可以使用其 Active Directory 用户帐户登录到 Linux 桌面。

本章讨论了以下主题：

- 将 Linux 与 Active Directory 集成
- 设置单点登录和智能卡重定向

## 将 Linux 与 Active Directory 集成

有多种解决方案可用于将 Linux 与 Active Directory (AD) 集成。

众所周知，以下解决方案适用于 View 环境：

- OpenLDAP 直通身份验证
- Winbind

OpenLDAP 直通身份验证解决方案主要包含以下步骤：

- 配置 OpenLDAP 服务器以将密码验证委托给一个单独的进程（例如 `saslauthd`），此进程可以对 Active Directory 执行密码验证。
- 配置 Linux 桌面，以通过 OpenLDAP 对用户进行身份验证。

如果您计划批量部署 Linux 桌面，则可以设置模板虚拟机 (VM) 来运行最终的 AD 集成任务。请注意以下事项：

- OpenLDAP 解决方案在用于克隆的虚拟机时不需要执行任何额外步骤。
- 对于 Winbind 解决方案，由于每个克隆虚拟机的主机名不同，加入域的步骤将失败。每个克隆的虚拟机需要运行以下命令来重新加入域：

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

VMware 建议采用 OpenLDAP 或类似的解决方案，因为它不需要在克隆的虚拟机上执行额外步骤。

有关批量部署 Linux 桌面的更多信息，请参阅第 7 章 批量部署 [Horizon 7 for Linux 桌面](#)。

## 设置单点登录和智能卡重定向

要设置单点登录 (SSO) 和智能卡重定向，您必须执行一些配置步骤。

### 单点登录

Horizon View 单点登录模块会与 Linux 中的 PAM（可插入身份验证模块）通讯，这不依赖于您将 Linux 与 Active Directory (AD) 集成所使用的方法。Horizon View SSO 可以用于 OpenLDAP 及 Winbind 这两种将 Linux 与 AD 集成的解决方案。

默认情况下，SSO 假定 AD 的 `sAMAccountName` 属性是登录 ID。如果使用 OpenLDAP 或 Winbind 解决方案，要确保 SSO 使用正确的登录 ID，您必须执行以下配置步骤：

- 对于 OpenLDAP，将 `sAMAccountName` 设置为 `uid`。
- 对于 Winbind，将以下语句添加到配置文件 `/etc/samba/smb.conf`。

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

如果用户必须指定登录的域名，您必须在 Linux 桌面上设置 `SSOUserFormat` 选项。有关更多信息，请参阅在 [Linux 桌面上的配置文件中设置选项](#)。注意，SSO 始终使用大写的短域名。例如，如果域是 `mydomain.com`，SSO 将使用 `MYDOMAIN` 作为域名。因此，在设置 `SSOUserFormat` 选项时必须指定 `MYDOMAIN`。关于短域名和长域名，请遵循以下规则：

- 对于 OpenLDAP，您必须使用大写的短域名。
- Winbind 同时支持长短两种域名。

AD 支持在登录名称中使用特殊字符，而 Linux 则不支持。因此，在设置 SSO 时，请不要在登录名称中使用特殊字符。

在 AD 中，如果用户的 `UserPrincipalName (UPN)` 属性和 `sAMAccount` 属性不匹配，而用户使用 UPN 登录，则 SSO 将失败。解决办法是，用户使用 `sAMAccount` 中存储的名称登录。

View 不要求用户名区分大小写。您必须确保 Linux 操作系统可以处理不区分大小写的用户名。

- 对于 Winbind，用户名默认不区分大小写。
- 对于 OpenLDAP，Ubuntu 使用 NSCD 对用户进行身份验证，并且默认不区分大小写。RHEL 和 CentOS 使用 SSSD 对用户进行身份验证，并且默认区分大小写。要更改该设置，请编辑文件 `/etc/sss/sss.conf`，并在 `[domain/default]` 部分添加以下行：

```
case_sensitive = false
```

## 智能卡重定向

要设置智能卡重定向，请先按照 Linux 分销商和智能卡供应商的说明操作。然后将 **pcsc-lite** 软件包更新至 1.7.4。例如，运行以下命令：

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

在安装 Horizon Agent 时，您必须先禁用 SELinux 或为 SELinux 启用许可模式。您还必须专门选择智能卡重定向组件，因为该组件默认未被选中。有关更多信息，请参阅 [install\\_viewagent.sh 命令行选项](#)。

启用智能卡重定向后，如果用户使用智能卡登录，则不会实施 SSO。也就是说，在使用智能卡登录连接服务器实例后，用户需要键入智能卡 PIN 才能登录 Linux 桌面。此外，如果虚拟机中已安装智能卡重定向功能，vSphere Client 的 USB 重定向将无法与智能卡一起使用。

智能卡重定向仅支持一个智能卡读卡器。如果两个或更多读卡器连接客户端设备，此功能将无法正常使用。

智能卡重定向只支持卡上的一个证书。如果卡上有多个证书，将使用第一个插槽中的证书，其他证书将被忽略。这是 Linux 的一个限制。

# Linux 桌面的配置选项

您可以使用配置文件配置各种选项，以便自定义用户体验。

本章讨论了以下主题：

- 在 Linux 桌面上的配置文件中设置选项
- 禁止在 vSphere 控制台中显示 Linux 桌面

## 在 Linux 桌面上的配置文件中设置选项

您可以向文件 `/etc/vmware/config` 或 `/etc/vmware/viewagent-custom.conf` 添加条目，以配置某些选项。

在 View Agent 或 Horizon Agent 安装过程中，安装程序将两个配置模板文件 `template_config` 和 `template_viewagent-custom.conf` 复制到 `/etc/vmware` 中。此外，如果 `/etc/vmware/config` 和 `/etc/vmware/viewagent-custom.conf` 文件不存在，安装程序将 `template_config` 复制到 `config` 并将 `template_viewagent-custom.conf` 复制到 `viewagent-custom.conf`。在模板文件中会列出并记录所有配置选项。要设置某个选项，只需移除注释和更改相关值即可。

例如，`/etc/vmware/config` 中的以下行用于启用无损 PNG 模式。

```
RemoteDisplay.alwaysLossless=TRUE
```

在进行配置更改后，重新引导 Linux 以使更改生效。

## `/etc/vmware/config` 中的配置选项

VMwareBlastServer 及其相关插件使用配置文件 `/etc/vmware/config`。

**表 4-1. `/etc/vmware/config` 中的配置选项**

选项	值	默认	说明
<code>RemoteDisplay.alwaysLossless</code>	TRUE 或 FALSE	FALSE	图形应用程序（尤其是图形设计应用程序）要求图像在 Linux 桌面的客户端显示器中呈现时精确至像素级别。您可以为 Linux 桌面上生成并在客户端设备上呈现的图像和视频回放配置无损 PNG 模式。该功能会占用客户端与 ESXi 主机之间额外的带宽。
<code>mksVNCServer.useUIInputButton Mapping</code>	TRUE 或 FALSE	FALSE	设置该选项以在 Ubuntu 或 RHEL 7 上启用左手鼠标支持。CentOS 和 RHEL 6.6 支持左手鼠标，您不需要设置该选项。
<code>RemoteDisplay.allowAudio</code>	TRUE 或 FALSE	TRUE	设置该选项以禁用音频输出

选项	值	默认	说明
VVC.ScRedir.Enable	TRUE 或 FALSE	TRUE	设置该选项以禁用智能卡重定向。
VVC.logLevel	FATAL、ERROR、WARN、INFO、DEBUG 或 TRACE	INFO	使用该选项设置 VVC 代理节点的日志级别。
Clipboard.Direction	0、1、2 或 3	2	该选项确定剪贴板重定向策略。 <ul style="list-style-type: none"> <li>■ 0 - 禁用剪贴板重定向。</li> <li>■ 1 - 启用双向剪贴板重定向。</li> <li>■ 2 - 仅启用从客户端到远程桌面的剪贴板重定向。</li> <li>■ 3 - 仅启用从远程桌面到客户端的剪贴板重定向。</li> </ul>
mksVNCServer.useXExtButtonMapping	TRUE 或 FALSE	FALSE	设置该选项以在 SLED 11 SP3 上启用或禁用左手鼠标支持。

## /etc/vmware/viewagent-custom.conf 中的配置选项

Java Standalone Agent 使用配置文件 /etc/vmware/viewagent-custom.conf。

**表 4-2. /etc/vmware/viewagent-custom.conf 中的配置选项**

选项	值	默认	说明
Subnet	NULL 或采用 IP 地址/CIDR 格式的网络地址和掩码	NULL	如果存在多个具有不同子网的本地 IP 地址，可以使用该选项设置 View 连接服务器连接到代理计算机时使用的子网。 您必须指定 IP 地址/CIDR 格式的值。例如，Subnet=192.168.1.0/24。 NULL 意味着 Linux 代理随机选择 IP 地址。
SSOEnable	TRUE 或 FALSE	TRUE	设置该选项以禁用单点登录 (SSO)。
SSOUserFormat	文本字符串	[username]	使用该选项以指定用于单点登录的登录名称格式。默认值只包含用户名。如果还需要域名，请设置该选项。通常，登录名称是域名加上一个特殊字符，再加上用户名。如果特殊字符是反斜杠，则必须使用另一个反斜杠对其进行转义。登录名称格式示例： <ul style="list-style-type: none"> <li>■ SSOUserFormat=[domain]\\[username]</li> <li>■ SSOUserFormat=[domain]+[username]</li> <li>■ SSOUserFormat=[username]@[domain]</li> </ul>
StartBlastServerTimeout	整数	20	该选项决定 VMwareBlastServer 进程初始化的时间长短（以秒为单位）。如果进程在此超时值内未准备就绪，用户登录将失败。
SSLCiphers	文本字符串	!aNULL:kECDH+AES:ECDH+AES:RSA+AES:@STRENGTH	使用该选项以指定密码列表。您必须使用 <a href="https://www.openssl.org/docs/manmaster/apps/ciphers.html">https://www.openssl.org/docs/manmaster/apps/ciphers.html</a> 中定义的格式。
SSLProtocols	文本字符串	TLSv1_1:TLSv1_2	使用该选项以指定安全协议。支持的协议是 TLSv1.0、TLSv1.1 和 TLSv1.2。

选项	值	默认	说明
SSLCipherServerPreference	TRUE 或 FALSE	TRUE	使用该选项以启用或禁用选项 SSL_OP_CIPHER_SERVER_PREFERENCE。有关更多信息，请参阅 <a href="https://www.openssl.org/docs/manmaster/ssl/SSL_CTX_set_options.html">https://www.openssl.org/docs/manmaster/ssl/SSL_CTX_set_options.html</a> 。
LogoutAfterDisconnectTimeout	整数	2	如果正在进行登录，可以使用该选项设置 Automatically logoff after disconnect 超时值（分钟）。
LogCnt	整数	-1	使用该选项设置在 /tmp/vmware-root 中保留的日志文件数。 <ul style="list-style-type: none"> <li>■ -1 - 全部保留</li> <li>■ 0 - 全部删除</li> <li>■ &gt; 0 - 保留的日志计数。</li> </ul>

**注** VMwareBlastServer 进程有三个安全选项：SSLCiphers、SSLProtocols 和 SSLCipherServerPreference。在启动 VMwareBlastServer 进程时，Java Standalone Agent 将这些选项作为参数传递。启用 Blast 安全网关 (BSG) 时，这些选项会影响 BSG 和 Linux 桌面之间的连接。当 BSG 被禁用时，这些选项会影响客户端和 Linux 桌面之间的连接。

## 禁止在 vSphere 控制台中显示 Linux 桌面

用户连接到 Linux 桌面时，该桌面还可以显示在 Linux 虚拟机的 vSphere 控制台中。您可以对 Linux 虚拟机进行配置，以确保在用户连接到其桌面时 vSphere 控制台中不显示任何桌面。

### 步骤

- ◆ 在 ESXi 主机上，将以下行添加到 Linux 虚拟机的 vmx 文件中。

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

甚至当您在用户已从桌面注销的情况下连接到虚拟机时，vSphere 控制台仍不会显示任何桌面。

## 为 Linux 桌面设置图形

您可以配置 RHEL 6.6 和 7.1，以在 ESXi 主机或客户机操作系统上利用 NVIDIA 功能。

### 设置 3D 图形的虚拟机克隆要求

在设置 3D 图形之前，您必须考虑以下虚拟机克隆要求。

- 对于 vGPU 和 vSGA，在基础虚拟机上完成图形设置。克隆这些虚拟机。图形设置可用于克隆的虚拟机，无需执行进一步设置。
- 对于 vDGA，在基础虚拟机上完成图形设置。克隆这些虚拟机。但是，在打开克隆虚拟机的电源之前，您必须先从克隆的虚拟机中移除现有 NVIDIA 直通 PCI 设备，然后再向克隆的虚拟机中添加新的 NVIDIA 直通 PCI 设备。无法在虚拟机之间共享 NVIDIA 直通 PCI 设备。每个虚拟机都使用一个专用的 NVIDIA 直通 PCI 设备。

本章讨论了以下主题：

- 为 vGPU 配置 RHEL 6.6 和 RHEL 7.1
- 配置 RHEL 6.6 以支持 vDGA
- 配置 RHEL 7.1 以支持 vSGA

## 为 vGPU 配置 RHEL 6.6 和 RHEL 7.1

您可以设置 RHEL 6.6 和 RHEL 7.1，以在 ESXi 主机上利用 NVIDIA vGPU（共享 GPU 硬件加速）功能。

---

**重要事项** 在 NVIDIA Maxwell M60 显卡上支持 NVIDIA vGPU。该功能在其他 NVIDIA 显卡（如 GRID K1 或 K2）上无法正常工作。

---

**小心** 在开始之前，请确认未在 Linux 虚拟机上安装 Horizon Agent。如果在配置计算机以使用 NVIDIA vGPU 之前安装 Horizon Agent，则会覆盖 xorg.conf 文件中的所需配置参数，并且 NVIDIA vGPU 无法正常工作。您必须在完成 NVIDIA vGPU 配置后安装 Horizon Agent。

---

## 为 ESXi 主机上的 NVIDIA 显卡安装 VIB

您必须为 ESXi 6.0 U1 或更高版本的主机上的 NVIDIA GRID 显卡下载并安装 VIB。vSGA 现在支持 NVIDIA 驱动程序 346.69 和 346.42。

对于 NVIDIA GRID vGPU 配置，NVIDIA 提供了一个包含 vGPU 管理器（在此步骤中在 ESXi 主机上安装）和 Linux 显卡驱动程序（在后面的步骤中在 Linux 虚拟机上安装）的 vGPU 软件包。

对于 vSGA 配置，NVIDIA 提供了一个用于 vSGA 的 VMware vSphere ESXi 驱动程序。对于 vSGA，不会在 Linux 虚拟机上安装 NVIDIA 显卡驱动程序。

### 前提条件

- 确认在您的环境中安装了 vSphere 6.0 U1 或更高版本。
- 确认在您的环境中安装了 NVIDIA 驱动程序 346.69 或 346.42。
- 对于 NVIDIA GRID vGPU 配置，请确认在 ESXi 主机上安装了 NVIDIA Maxwell M60 GPU。
- 对于 vSGA 配置，请确认在 ESXi 主机上安装了 NVIDIA GRID K1 或 K2 GPU。

### 步骤

- 1 从 [NVIDIA 驱动程序下载](#) 站点下载用于您的 NVIDIA GRID 显卡的 VIB。

从下拉菜单中选择相应的 VIB 版本。

选项	说明
产品类型	GRID
产品系列	对于 vGPU，请选择 <b>NVIDIA GRID vGPU</b> 。 对于 vSGA，请选择 <b>GRID 系列</b> 。
产品	选择 ESXi 主机上安装的版本（例如 <b>GRID K2</b> ）。
操作系统	选择 VMware vSphere ESXi 版本。

- 2 对于 NVIDIA GRID vGPU 配置，请执行以下步骤：

- a 解压缩 vGPU 软件包 .zip 文件。
- b 将 vGPU 管理器文件夹上载到 ESXi 6.0 U1 主机中。

**注** 将在后面的步骤中在 Linux 虚拟机上安装 Linux 显卡驱动程序。

- 3 对于 vSGA 配置，请将用于 vSGA 的 VMware vSphere ESXi 驱动程序上载到 ESXi 6.0 U1 主机中。
- 4 关闭电源或挂起 ESXi 主机上的所有虚拟机。
- 5 使用 SSH 连接到 ESXi 主机。
- 6 停止 xorg 服务。

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

- 7 安装 NVIDIA VIB。

例如：

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



## 8 重新引导或更新 ESXi 主机。

3D 类型	说明
<b>NVIDIA GRID vGPU</b>	<p>对于安装的 ESXi 主机，请重新引导该主机。</p> <p>对于无状态 ESXi 主机，请执行以下步骤以更新该主机。（这些步骤也适用于安装的主机。）</p> <pre>Update vmkdevmgr: # kill -HUP \$(cat /var/run/vmware/vmkdevmgr.pid)  Wait for the update to complete: # localcli --plugin-dir /usr/lib/vmware/esxcli/int deviceInternal bind  This is a new requirement with the NVIDIA 352.* host driver: # /etc/init.d/nvidia-vgpu start  Restart xorg, which is used for GPU assignment: # /etc/init.d/xorg start</pre>
<b>vSGA</b>	<p>a 重新启动 xorg（用于 GPU 分配）：</p> <pre># /etc/init.d/xorg start</pre> <p>b 重新引导 ESXi 主机。</p>

## 9 在重新启动主机后，验证是否正在运行 xorg 服务。

## 在 Linux 虚拟机上为 vGPU 配置共享 PCI 设备

要使用 NVIDIA vGPU，您必须为 Linux 虚拟机配置共享 PCI 设备。

### 前提条件

- 确认准备好将 Linux 虚拟机作为桌面。请参阅[创建虚拟机并安装 Linux](#)和[为远程桌面部署准备 Linux 计算机](#)。
- 确认未在 Linux 虚拟机上安装 Horizon Agent。
- 确认在 ESXi 主机上安装了 NVIDIA VIB。请参阅[为 ESXi 主机上的 NVIDIA 显卡安装 VIB](#)。
- 熟悉适用于 NVIDIA vGPU 的虚拟 GPU 类型，您可以通过 **GPU 配置文件** 设置进行选择。虚拟 GPU 类型在 ESXi 主机上安装的物理 GPU 中提供了各种不同的功能。请参阅[NVIDIA 虚拟 GPU 类型](#)。

### 步骤

- 1 关闭虚拟机的电源，
- 2 在 vSphere Web Client 中，选择虚拟机，然后在**虚拟机硬件**选项卡下，单击**编辑设置**。
- 3 在**新设备**菜单中，选择**共享 PCI 设备**。
- 4 单击**添加**，然后从下拉菜单中选择 **NVIDIA GRID vGPU**。
- 5 对于 **GPU 配置文件** 设置，请从下拉菜单中选择一种虚拟 GPU 类型。

**6** 单击**保留所有内存**，然后单击**确定**。

您必须保留所有虚拟机内存，以使 GPU 能够支持 NVIDIA GRID vGPU。

**7** 启动虚拟机。**NVIDIA 虚拟 GPU 类型**

通过使用 vSphere Web Client 的“虚拟硬件”页中的 **GPU 配置文件** 设置，您可以选择一种虚拟 GPU 类型以在 ESXi 主机上的物理 NVIDIA GPU 中提供特定的功能。

在 Linux 虚拟机上，在 NVIDIA Maxwell M60 GPU 上支持 NVIDIA GRID vGPU。

**表 5-1. 适用于 Linux 虚拟机上的 NVIDIA GRID vGPU 的虚拟 GPU 类型**

虚拟 GPU 类型	物理主板	物理 GPU	每个虚拟 GPU 的 FB 数	显示器数	最大分辨率	每个物理 GPU 的最大虚拟 GPU 数	每个物理主板的最大虚拟 GPU 数
GRID M60-0q	GRID M60	2	512M	2	2560x1600	16	32
GRID M60-1q	GRID M60	2	1G	2	2560x1600	8	16
GRID M60-2q	GRID M60	2	2G	4	2560x1600	4	8
GRID M60-4q	GRID M60	2	4G	4	3840x2160	2	4
GRID M60-8q	GRID M60	2	8G	4	3840x2160	1	2

**安装 NVIDIA 显卡驱动程序**

要安装 NVIDIA 显卡驱动程序，您必须禁用默认 NVIDIA 驱动程序，下载 NVIDIA 显卡驱动程序，然后在虚拟机上配置 PCI 设备。

**前提条件**

- 对于 NVIDIA GRID vGPU 配置，请确认从 NVIDIA 下载站点下载了 vGPU 软件包，解压缩该软件包，然后将 Linux 显卡驱动程序（一个软件包组件）准备就绪。请参阅[ESXi 主机上的 NVIDIA 显卡安装 VIB](#)。

还要确认在虚拟机中添加了共享 PCI 设备。请参阅[在 Linux 虚拟机上为 vGPU 配置共享 PCI 设备](#)

- 对于 vDGA 配置，请确认在 RHEL 6.6 虚拟机中添加了 PCI 设备。请参阅[向 RHEL 6.6 虚拟机添加 vDGA 直通设备](#)。

**步骤****1** 禁用默认的 NVIDIA Nouveau 驱动程序并将其列入黑名单。**a** 编辑 `grub.conf` 文件。

对于 RHEL 6.6，此文件是 `/boot/grub/grub.conf`。对于 RHEL 7.1，此文件是 `/etc/default/grub.conf`。

RHEL 版本	命令
6.6	<code>sudo vi /boot/grub/grub.conf</code>
7.1	<code>sudo vi /etc/default/grub.conf</code>

**b** 在内核选项的末尾添加 `rdblacklist=nouveau` 行。**c** 编辑 `blacklist.conf` 文件。

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

**d** 在 `blacklist.conf` 文件中的任意位置添加以下行。

```
blacklist nouveau
```

**2** 重新启动虚拟机。

显示器的外观将发生更改。

**3** （可选） 确认已禁用 Nouveau 驱动程序。

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

如果 `grep` 搜索没有返回任何结果，则表示 Nouveau 驱动程序已被禁用。

**4** 将 NVIDIA 显卡驱动程序复制或下载到虚拟机中。**a** 对于 NVIDIA GRID vGPU 配置，请将 NVIDIA Linux 显卡驱动程序复制到虚拟机中。**b** 对于 vDGA 配置，请从 [NVIDIA 驱动程序下载](#) 站点下载 NVIDIA 驱动程序。

从 NVIDIA 下拉菜单中选择相应的驱动程序版本：

选项	说明
产品类型	GRID
产品系列	GRID 系列
产品	选择 ESXi 主机上安装的版本（例如 <b>GRID K2</b> ）。
操作系统	Linux 64 位或 Linux 32 位

**5** 打开虚拟机的远程终端，或通过键入 `Ctrl-Alt-F2` 切换到文本控制台，以 `root` 用户身份登录，然后运行 `init 3` 命令以禁用 X Windows。

## 6 安装 NVIDIA 驱动程序所需的其他组件。

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

## 7 向 NVIDIA 驱动程序包添加可执行标记。

以下示例使用一个用于 NVIDIA GRID vGPU 的驱动程序包：

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

以下示例使用一个用于 vDGA 的驱动程序包：

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

## 8 启动 NVIDIA 安装程序。

对于 NVIDIA GRID vGPU：

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

对于 vDGA：

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

## 9 接受 NVIDIA 软件许可协议，然后选择是以自动更新 X 配置设置。

### 后续步骤

在 Linux 虚拟机上安装 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。

创建一个桌面池以包含配置的 Linux 虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

## 验证是否安装了 NVIDIA 显卡驱动程序

您可以在 View 桌面会话中显示 NVIDIA 驱动程序输出以验证是否在 RHEL 6.6 虚拟机上安装了 NVIDIA 显卡驱动程序。

### 前提条件

- 确认安装了 NVIDIA 显卡驱动程序。
- 确认在 Linux 虚拟机上安装了 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。
- 确认在桌面池中部署了 Linux 虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

### 步骤

#### 1 重新启动 Linux 虚拟机。

Horizon Agent 启动脚本将初始化 X 服务器和显示器拓扑。

您无法再通过 vSphere 控制台查看虚拟机显示器。

- 2 从 Horizon Client 中连接到 Linux 桌面。
- 3 在 Linux 桌面会话中，验证是否安装了 NVIDIA 显卡驱动程序。

打开终端窗口，运行 `glxinfo | grep NVIDIA` 命令。

此时将显示 NVIDIA 驱动程序输出。例如：

```
[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
```

用户可以在远程桌面上访问 NVIDIA 图形功能。

在确认已安装 NVIDIA 显卡驱动程序后，请执行以下任务来使安装的程序正常运行。

- 如果升级 Linux 内核，Horizon Agent 可能无法与 View 连接服务器通信。要解决此问题，请重新安装 NVIDIA 驱动程序。
- 在 Linux 虚拟机中设置 NVIDIA GRID 许可。有关更多信息，请参阅 NVIDIA 文档。如果不设置许可，Linux 桌面将无法正常工作。例如，自动调整功能将无法使用。

## 配置 RHEL 6.6 以支持 vDGA

您可以设置 RHEL 6.6 客户机操作系统，以便 Horizon 7 for Linux 桌面可以利用 ESXi 主机上的 vDGA 功能。

**小心** 在开始之前，请确认未在 Linux 虚拟机上安装 Horizon Agent。如果在配置计算机以使用 vDGA 之前安装 Horizon Agent，则会覆盖 `xorg.conf` 文件中的所需配置参数，并且 vDGA 无法正常工作。您必须在完成 vDGA 配置后安装 Horizon Agent。

## 在主机上为 NVIDIA GRID 启用 DirectPath I/O

在您将 Linux 虚拟机配置为使用 vDGA 之前，必须使 NVIDIA GRID GPU PCI 设备可用于 ESXi 主机上的 DirectPath I/O 传送。

### 前提条件

- 确认您的环境中安装了 vSphere 6.0 或更高版本。
- 确认 ESXi 主机上安装了 NVIDIA GRID K1 或 K2 显卡。

### 步骤

- 1 在 vSphere Web Client 中，浏览到 ESXi 主机。
- 2 单击**管理**选项卡，然后单击**设置**。
- 3 在“硬件”部分中，单击 **PCI 设备**。

- 4 要为 NVIDIA GRID GPU 启用 DirectPath I/O 传送，请单击**编辑**。

图标	说明
绿色图标	PCI 设备处于活动状态并且可以启用。
橙色图标	设备的状态已更改。您必须先重新引导主机，然后才能使用该设备。

- 5 选择 NVIDIA GRID GPU，然后单击**确定**。

PCI 设备将添加到表“可用于虚拟机的 DirectPath I/O PCI 设备”中。

- 6 重新引导主机，以便使 PCI 设备可用于 Linux 虚拟机。

## 向 RHEL 6.6 虚拟机添加 vDGA 直通设备

要将 RHEL 6.6 虚拟机配置为使用 vDGA，您必须向虚拟机添加 PCI 设备。通过此步骤，可以将 ESXi 主机上的物理设备进行传送，以便在虚拟机上使用。

### 前提条件

- 确认准备好将 Linux 虚拟机作为桌面。请参阅[创建虚拟机并安装 Linux](#)和[为远程桌面部署准备 Linux 计算机](#)。
- 确认未在 Linux 虚拟机上安装 Horizon Agent。
- 确认 NVIDIA GRID GPU PCI 设备在主机上可用于 DirectPath I/O 直通。请参阅[在主机上为 NVIDIA GRID 启用 DirectPath I/O](#)。

### 步骤

- 1 关闭虚拟机的电源，并以配置了 `sudo` 权限的本地用户身份登录 RHEL 6.6 客户机操作系统。
- 2 在 vSphere Web Client 中，选择虚拟机，然后在**虚拟机硬件**选项卡下，单击**编辑设置**。
- 3 在**新设备**菜单中，选择 **PCI 设备**。
- 4 单击**添加**并从下拉菜单选择 PCI 设备。
- 5 单击**保留所有内存**，然后单击**确定**。

您必须保留所有虚拟机内存才能使 GPU 支持 vDGA。

- 6 打开虚拟机的电源，然后打开 vSphere 控制台以连接到计算机。
- 7 确认 NVIDIA GRID 设备已传送到虚拟机。

打开终端窗口，运行以下命令：

```
lspci | grep NVIDIA
```

此时会显示与 VGA 兼容的控制器 XX:00.0。例如：

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

## 安装 NVIDIA 显卡驱动程序

要安装 NVIDIA 显卡驱动程序，您必须禁用默认 NVIDIA 驱动程序，下载 NVIDIA 显卡驱动程序，然后在虚拟机上配置 PCI 设备。

### 前提条件

- 对于 NVIDIA GRID vGPU 配置，请确认从 NVIDIA 下载站点下载了 vGPU 软件包，解压缩该软件包，然后将 Linux 显卡驱动程序（一个软件包组件）准备就绪。请参阅[ESXi 主机上的 NVIDIA 显卡安装 VIB](#)。

还要确认在虚拟机中添加了共享 PCI 设备。请参阅[在 Linux 虚拟机上为 vGPU 配置共享 PCI 设备](#)

- 对于 vDGA 配置，请确认在 RHEL 6.6 虚拟机中添加了 PCI 设备。请参阅[向 RHEL 6.6 虚拟机添加 vDGA 直通设备](#)。

### 步骤

- 1 禁用默认的 NVIDIA Nouveau 驱动程序并将其列入黑名单。

- a 编辑 `grub.conf` 文件。

对于 RHEL 6.6，此文件是 `/boot/grub/grub.conf`。对于 RHEL 7.1，此文件是 `/etc/default/grub.conf`。

RHEL 版本	命令
6.6	<code>sudo vi /boot/grub/grub.conf</code>
7.1	<code>sudo vi /etc/default/grub.conf</code>

- b 在内核选项的末尾添加 `rdblacklist=nouveau` 行。

- c 编辑 `blacklist.conf` 文件。

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

- d 在 `blacklist.conf` 文件中的任意位置添加以下行。

```
blacklist nouveau
```

- 2 重新启动虚拟机。

显示器的外观将发生更改。

- 3 （可选） 确认已禁用 Nouveau 驱动程序。

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

如果 `grep` 搜索没有返回任何结果，则表示 Nouveau 驱动程序已被禁用。

#### 4 将 NVIDIA 显卡驱动程序复制或下载到虚拟机中。

- a 对于 NVIDIA GRID vGPU 配置，请将 NVIDIA Linux 显卡驱动程序复制到虚拟机中。
- b 对于 vDGA 配置，请从 [NVIDIA 驱动程序下载](#) 站点下载 NVIDIA 驱动程序。

从 NVIDIA 下拉菜单中选择相应的驱动程序版本：

选项	说明
产品类型	GRID
产品系列	GRID 系列
产品	选择 ESXi 主机上安装的版本（例如 <b>GRID K2</b> ）。
操作系统	Linux 64 位或 Linux 32 位

- 5 打开虚拟机的远程终端，或通过键入 **Ctrl-Alt-F2** 切换到文本控制台，以 **root** 用户身份登录，然后运行 **init 3** 命令以禁用 X Windows。
- 6 安装 NVIDIA 驱动程序所需的其它组件。

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

#### 7 向 NVIDIA 驱动程序包添加可执行标记。

以下示例使用一个用于 NVIDIA GRID vGPU 的驱动程序包：

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

以下示例使用一个用于 vDGA 的驱动程序包：

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

#### 8 启动 NVIDIA 安装程序。

对于 NVIDIA GRID vGPU：

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

对于 vDGA：

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

#### 9 接受 NVIDIA 软件许可协议，然后选择是以自动更新 X 配置设置。

#### 后续步骤

在 Linux 虚拟机上安装 Horizon Agent。请参阅在 [Linux 虚拟机上安装 Horizon Agent](#)。

创建一个桌面池以包含配置的 Linux 虚拟机。请参阅 [创建包含 Linux 虚拟机的桌面池](#)。



## 验证是否安装了 NVIDIA 显卡驱动程序

您可以在 View 桌面会话中显示 NVIDIA 驱动程序输出以验证是否在 RHEL 6.6 虚拟机上安装了 NVIDIA 显卡驱动程序。

### 前提条件

- 确认安装了 NVIDIA 显卡驱动程序。
- 确认在 Linux 虚拟机上安装了 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。
- 确认在桌面池中部署了 Linux 虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

### 步骤

- 1 重新启动 Linux 虚拟机。

Horizon Agent 启动脚本将初始化 X 服务器和显示器拓扑。

您无法再通过 vSphere 控制台查看虚拟机显示器。

- 2 从 Horizon Client 中连接到 Linux 桌面。

- 3 在 Linux 桌面会话中，验证是否安装了 NVIDIA 显卡驱动程序。

打开终端窗口，运行 `glxinfo | grep NVIDIA` 命令。

此时将显示 NVIDIA 驱动程序输出。例如：

```
[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
```

用户可以在远程桌面上访问 NVIDIA 图形功能。

在确认已安装 NVIDIA 显卡驱动程序后，请执行以下任务来使安装的程序正常运行。

- 如果升级 Linux 内核，Horizon Agent 可能无法与 View 连接服务器通信。要解决此问题，请重新安装 NVIDIA 驱动程序。
- 在 Linux 虚拟机中设置 NVIDIA GRID 许可。有关更多信息，请参阅 NVIDIA 文档。如果不设置许可，Linux 桌面将无法正常工作。例如，自动调整功能将无法使用。

## 配置 RHEL 7.1 以支持 vSGA

您可以设置 RHEL 7.1 客户机操作系统，以便 Horizon 7 for Linux 桌面可以利用 vSGA 功能。

### 为 ESXi 主机上的 NVIDIA 显卡安装 VIB

您必须为 ESXi 6.0 U1 或更高版本的主机上的 NVIDIA GRID 显卡下载并安装 VIB。vSGA 现在支持 NVIDIA 驱动程序 346.69 和 346.42。

对于 NVIDIA GRID vGPU 配置，NVIDIA 提供了一个包含 vGPU 管理器（在此步骤中在 ESXi 主机上安装）和 Linux 显卡驱动程序（在后面的步骤中在 Linux 虚拟机上安装）的 vGPU 软件包。

对于 vSGA 配置，NVIDIA 提供了一个用于 vSGA 的 VMware vSphere ESXi 驱动程序。对于 vSGA，不会在 Linux 虚拟机上安装 NVIDIA 显卡驱动程序。

#### 前提条件

- 确认在您的环境中安装了 vSphere 6.0 U1 或更高版本。
- 确认在您的环境中安装了 NVIDIA 驱动程序 346.69 或 346.42。
- 对于 NVIDIA GRID vGPU 配置，请确认在 ESXi 主机上安装了 NVIDIA Maxwell M60 GPU。
- 对于 vSGA 配置，请确认在 ESXi 主机上安装了 NVIDIA GRID K1 或 K2 GPU。

#### 步骤

- 1 从 [NVIDIA 驱动程序下载](#) 站点下载用于您的 NVIDIA GRID 显卡的 VIB。

从下拉菜单中选择相应的 VIB 版本。

选项	说明
产品类型	GRID
产品系列	对于 vGPU，请选择 <b>NVIDIA GRID vGPU</b> 。 对于 vSGA，请选择 <b>GRID 系列</b> 。
产品	选择 ESXi 主机上安装的版本（例如 <b>GRID K2</b> ）。
操作系统	选择 VMware vSphere ESXi 版本。

- 2 对于 NVIDIA GRID vGPU 配置，请执行以下步骤：

- a 解压缩 vGPU 软件包 .zip 文件。
- b 将 vGPU 管理器文件夹上加载到 ESXi 6.0 U1 主机中。

**注** 将在后面的步骤中在 Linux 虚拟机上安装 Linux 显卡驱动程序。

- 3 对于 vSGA 配置，请将用于 vSGA 的 VMware vSphere ESXi 驱动程序上加载到 ESXi 6.0 U1 主机中。
- 4 关闭电源或挂起 ESXi 主机上的所有虚拟机。
- 5 使用 SSH 连接到 ESXi 主机。

## 6 停止 xorg 服务。

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

## 7 安装 NVIDIA VIB。

例如：

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

## 8 重新引导或更新 ESXi 主机。

3D 类型	说明
<b>NVIDIA GRID vGPU</b>	<p>对于安装的 ESXi 主机，请重新引导该主机。</p> <p>对于无状态 ESXi 主机，请执行以下步骤以更新该主机。（这些步骤也适用于安装的主机。）</p> <pre>Update vmkdevmgr: # kill -HUP \$(cat /var/run/vmware/vmkdevmgr.pid)  Wait for the update to complete: # localcli --plugin-dir /usr/lib/vmware/esxcli/int deviceInternal bind  This is a new requirement with the NVIDIA 352.* host driver: # /etc/init.d/nvidia-vgpu start  Restart xorg, which is used for GPU assignment: # /etc/init.d/xorg start</pre>
<b>vSGA</b>	<p>a 重新启动 xorg（用于 GPU 分配）：</p> <pre># /etc/init.d/xorg start</pre> <p>b 重新引导 ESXi 主机。</p>

## 9 在重新启动主机后，验证是否正在运行 xorg 服务。

# 在 Linux 虚拟机上为 vSGA 配置 3D 功能

要配置 RHEL 7.1 虚拟机以使用 vSGA，您必须在 vSphere Web Client 中为虚拟机显卡配置 3D 设置。

### 前提条件

- 确认 Linux 虚拟机已准备好用作桌面，已安装 Horizon Agent，并且已在桌面池中部署计算机。
- 确认在 ESXi 主机上安装了 NVIDIA VIB。请参阅[为 ESXi 主机上的 NVIDIA 显卡安装 VIB](#)。

### 步骤

- 1 关闭虚拟机的电源，
- 2 在 vSphere Web Client 中，选择虚拟机，然后在[虚拟机硬件](#)选项卡下，单击[编辑设置](#)。

- 3 在“虚拟硬件”选项卡中，单击**显卡**以展开菜单设置。
- 4 将**总视频内存**设置为 128 MB。
- 5 对于 **3D 图形**，请选择**启用 3D 支持**。
- 6 对于 **3D 呈现器**，请从下拉菜单中选择**硬件**。
- 7 对于 **3D 内存**，请选择符合您的应用程序要求的值。

如果您的用户连接到超过 3 个显示器，请将该值设置为至少 1024 MB。

- 8 单击**确定**。
- 9 启动虚拟机。

### 后续步骤

验证是否正在 Linux 虚拟机上运行 vSGA。

接下来，在 Linux 虚拟机上安装 Horizon Agent。请参阅在 [Linux 虚拟机上安装 Horizon Agent](#)。

## 验证是否正在 Linux 虚拟机上运行 vSGA

您可以检查虚拟机日志文件和客户机操作系统以验证是否正在 RHEL 7.1 虚拟机上运行 vSGA。

### 步骤

- 1 打开虚拟机的 `vmware.log` 文件。

如果正确安装了支持的 GPU 和 NVIDIA VIB，日志文件将显示类似以下示例中的行：

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

如果未正确安装支持的 GPU 和 NVIDIA VIB，则虚拟机使用软件呈现器。`vmware.log` 文件显示类似以下示例中的行：

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

- 2 在虚拟机上的客户机操作系统中，键入以下命令。

```
glxinfo|grep Gallium
```

如果 vSGA 正常工作，该命令将返回以下文本：

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

如果 vSGA 无法正常工作，该命令将返回以下文本：

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

### 后续步骤

在 Linux 虚拟机上安装 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。

# 安装 Horizon Agent 和管理 Linux 桌面

# 6

您必须在 Linux 桌面上安装 Horizon Agent，这样 View 连接服务器才能与桌面通信并管理桌面。

本章讨论了以下主题：

- 在 Linux 虚拟机上安装 Horizon Agent
- 启用可逆密码加密
- 配置 Linux 代理的证书
- 创建包含 Linux 虚拟机的桌面池
- 在 Linux 虚拟机上升级 Horizon Agent
- 卸载和重新安装 Horizon 7 for Linux 计算机
- 如何在 Linux 桌面上从 vSphere 执行电源操作
- 收集有关 Horizon 7 for Linux 软件的信息

## 在 Linux 虚拟机上安装 Horizon Agent

您必须先要在 Linux 虚拟机上安装 Horizon Agent，然后才能将该虚拟机部署为远程桌面。

要安装 Horizon Agent，您必须提供 View 管理员的凭据。安装程序会将该凭据发送至连接服务器进行身份验证。支持以下身份验证机制：

- DIGEST-MD5。此机制需要启用 Active Directory 策略使用可逆加密存储密码。有关启用此策略的详细信息，请参阅[启用可逆密码加密](#)。
- Kerberos。要使用此机制，您必须使用 **-k** 命令行选项指定 Active Directory 服务器的地址。此外，对于所需的参数 **-b**，您必须使用真实主机名和域名提供连接服务器的完全限定域名 (Fully Qualified Domain Name, FQDN)。您不能使用 IP 地址或映射的 FQDN。

**小心** 如果要使用 NVIDIA GRID vGPU、vDGA 或 vSGA，必须先要在 Linux 虚拟机上配置这些 3D 功能，然后再安装 Horizon Agent。如果先安装 Horizon Agent，则会覆盖 `xorg.conf` 文件中的所需参数，并且 3D 图形功能无法正常工作。

请参阅[为 vGPU 配置 RHEL 6.6 和 RHEL 7.1](#)、[配置 RHEL 6.6 以支持 vDGA](#) 或 [配置 RHEL 7.1 以支持 vSGA](#)。在 3D 图形配置完成后，安装 Horizon Agent。

对于 2D 图形配置，您可以在完成[为远程桌面部署准备 Linux 计算机](#)中的步骤后安装 Horizon Agent。

## 前提条件

- 确认已准备好 Linux 客户机操作系统以供桌面使用。请参阅[为远程桌面部署准备 Linux 计算机](#)。
- 熟悉用于 Linux 的 Horizon Agent 安装程序脚本。请参阅[install\\_viewagent.sh 命令行选项](#)。
- 确认您为其提供安装命令的 View 连接服务器管理员在 View Administrator 中具有代理注册管理员或管理员角色。具体而言，需要注册代理特权才能向 View 连接服务器注册 Horizon Agent。代理注册管理员是提供此最低特权的受限角色。

## 步骤

- 1 从 VMware 下载站点 <https://my.vmware.com/web/vmware/downloads> 下载 Horizon Agent for Linux 安装程序文件。

在“桌面和最终用户计算”下，选择 VMware Horizon 7 下载，其中包括 Horizon Agent for Linux 安装程序。

安装程序文件名是 VMware-viewagent-linux-x86\_64-y.y.y-xxxxxxx.tar.gz（对于 64 位 Linux）或 VMware-viewagent-linux-y.y.y-xxxxxxx.tar.gz（对于 32 位 Linux），其中 y.y.y 是版本号，xxxxxxx 是内部版本号。

- 2 在客户机操作系统上解压缩 Linux 分发包的 tarball。

例如：

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

- 3 浏览到 tar ball 文件夹。
- 4 以超级用户身份运行 install\_viewagent.sh 脚本。

请参阅[install\\_viewagent.sh 命令行选项](#)，获取命令行选项列表。

例如：

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

**-b** 参数用于指定向其注册 Linux 计算机的 View 连接服务器实例。您可以将 FQDN 或 IP 地址与 **-b** 参数一起使用。

**-d**、**-u** 和 **-p** 参数用于指定 View 管理员的域、用户名和密码。密码中若存在特殊字符（例如 \$），请务必将其转义。例如：ab\<\$cdef

如果您没有在该命令中键入 **-p** 参数，则当您输入命令后，系统将提示您提供密码。在提示符处键入密码时，文字会隐藏起来。

默认情况下，将使用 Linux 计算机的主机名来注册计算机。您可以使用 **-n** 参数指定其他计算机名称。

- 5 如果您运行 install\_viewagent.sh 而未指定 **-A** 选项，请键入 **Yes** 以接受 EULA。

除非接受 EULA，否则安装程序不会运行。

- 6 重新引导 Linux 以使更改生效。

安装后，计算机将通过 View 连接服务器注册，并且 viewagent 服务将启动。您可以通过运行 `#service viewagent status` 命令来验证此服务是否已启动。

如果显示“代理注册失败 (Agent Registration Failed)”消息，请参阅[对 Linux 计算机的 Horizon Agent 注册失败问题进行故障排除](#)。

如果出现有关 Horizon Agent 的其他问题，请参阅[第 8 章 对 Linux 桌面进行故障排除](#)。

**注** `install_viewagent.sh` 脚本将在 Linux 虚拟机上的 Horizon Agent 安装目录 `/usr/lib/vmware/viewagent/jre` 中安装 JRE 8。Horizon Agent 需要使用 JRE 8。

后续步骤

在桌面池中部署虚拟机。请参阅[创建包含 Linux 虚拟机的桌面池](#)。

install\_viewagent.sh 命令行选项

`install_viewagent.sh` 脚本会在 Linux 客户机操作系统上安装 Horizon Agent。

在 gnome 桌面环境中的命令窗口中使用以下形式的 `install_viewagent.sh` 脚本。

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

`install_viewagent.sh` 脚本包括必需参数和可选参数。

表 6-1. install\_viewagent.sh 必需参数

必需参数	说明
-b	View 连接服务器实例的 FQDN 或 IP 地址（如果身份验证机制是 DIGEST-MD5）。如果身份验证机制是 Kerberos，则指定实际 FQDN，而非 IP 地址或映射的 FQDN。
-d	代理管理员的域名。例如：mydomain.com
-u	View 连接服务器管理员的用户名。例如：myadmin 该管理员在 View Administrator 中必须拥有代理注册管理员或管理员角色。具体而言，需要注册代理特权才能向 View 连接服务器注册 Horizon Agent。代理注册管理员是提供此最低特权的受限角色。

表 6-2. install\_viewagent.sh 密码参数

可选参数（必需信息）	说明
-p	View 连接服务器管理员的密码。 如果您不希望以文本形式显示密码，则不必在命令行中键入 <code>-p</code> 参数。如果您键入 <code>install_viewagent.sh</code> 命令且没有附带 <code>-p</code> 参数，则系统将提示您输入密码。在提示符处键入密码时，文字会隐藏起来。 您必须使用 <code>-p</code> 参数或在提示符处提供密码。 如果密码包含特殊字符（例如，在其中执行安装程序的 shell 内的 <code>\$</code> ），请确保对特殊字符进行转义。例如：ab \ <code>\$cdef</code>
-A yes	接受 EULA。您必须指定 <b>yes</b> 才能继续进行安装。如果未指定该参数，安装脚本将提示输入该值。



表 6-3. install\_viewagent.sh 可选参数

可选参数	说明
-n	向 View 连接服务器注册的计算机名称。默认情况下，Horizon Agent 安装程序使用主机名。您可以指定自己的名称。
-s	自签名证书的主体 DN。 默认情况下，Horizon Agent 安装程序会生成一个具有以下主体 DN 值的自签名证书：'/C=US/ST=California/O=VMware/CN=Hostname.DomainName'。默认 CN 指定计算机的主机名和域名。 您可以指定自己的主体 DN。
-m yes	启用智能卡重定向。
-j	JMS SSL 密钥库密码。如果未指定，安装程序将生成随机密码。
-k	Active Directory 服务器的代理管理员地址。只有 Kerberos 身份验证需要使用此参数。
-r	安装后重新引导操作系统。允许的值是 <b>yes</b> 和 <b>no</b> 。默认值为 <b>no</b> 。
-K	Active Directory 服务器的代理主机地址。只有 Kerberos 身份验证在代理主机和代理管理员来自不同域的情况下才需要使用此参数。
-B	代理主机的域名。只有 Kerberos 身份验证在代理主机和代理管理员来自不同域的情况下才需要使用此参数。

表 6-4. install\_viewagent.sh 参数的示例

情况	示例
代理主机和代理管理员来自相同的域	<ul style="list-style-type: none"> <li>■ Digest-MD5。 <pre>sudo install_viewagent.sh -A yes-n mymachine-b mybroker.mydomain.com-d mydomain.com-u myadmin-p Pa\$\$sword</pre></li> <li>■ Kerberos。 <pre>sudo install_viewagent.sh -A yes-n mymachine-b mybroker.mydomain.com-d mydomain.com-u myadmin-p Pa\$\$sword -k mydomain-host.mydomain.com</pre></li> </ul>
代理主机和代理管理员来自不同的域	<ul style="list-style-type: none"> <li>■ MD5-Digest。 <pre>sudo install_viewagent.sh -A yes-n mymachine-b mybroker.mydomain1.com-d mydomain2.com-u myadmin@mydomain2.com-p Pa\$\$sword</pre></li> <li>■ Kerberos。 <pre>sudo install_viewagent.sh -A yes-n mymachine-b mybroker.mydomain1.com-B mydomain1.com-K mydomain1-host.mydomain1.com -d mydomain2.com-u myadmin-p Pa\$\$sword-k mydomain2-host.mydomain2.com</pre></li> </ul>

## 启用可逆密码加密

在安装过程中，Horizon Agent 必须向 View 连接服务器验证其 View 管理员的身份。如果身份验证机制是 DIGEST-MD5，您必须先为管理员帐户启用 Active Directory (AD) 设置使用可逆加密存储密码。

### 步骤

- 1 在 AD 服务器上，打开**管理工具 > Active Directory 用户和计算机**。
- 2 右键单击 View 管理员的帐户并选择**属性**。
- 3 选择“帐户”选项卡。
- 4 在“帐户”选项中，选中**使用可逆加密存储密码**设置。
- 5 重置 View 管理员密码。

## 配置 Linux 代理的证书

安装 Linux 代理时，安装程序会为 VMwareBlastServer 生成一个自签名证书。

- 在代理上禁用 Blast 安全网关后，VMwareBlastServer 会向使用 HTML Access 连接到 Linux 桌面的浏览器提供该证书。
- 在代理上启用 Blast 安全网关后，Blast 安全网关的证书会将该证书提供给浏览器。

为了遵循行业或安全法规，您可以将该自签名证书替换为由证书颁发机构 (Certificate Authority, CA) 签名的证书。

### 步骤

- 1 向 VMwareBlastServer 安装私有密钥和证书。
  - a 将私有密钥重命名为 rui.key，将证书重命名为 rui.crt。
  - b 运行 `sudo chmod 550 /etc/vmware/ssl`。
  - c 将 rui.crt 和 rui.key 复制到 /etc/vmware/ssl。
  - d 运行 `chmod 440 /etc/vmware/ssl`。
- 2 将根和中间证书颁发机构安装到 Linux 操作系统证书颁发机构存储中。

---

**注** 查看您的 Linux 分发包文档，了解 Linux 系统设置更改信息。

---

## 创建包含 Linux 虚拟机的桌面池

要将 Linux 虚拟机配置为用作远程桌面，您需要创建手动桌面池并将 Linux 计算机添加到池中。

当您创建桌面池时，仅将 Linux 虚拟机添加到池中。如果池中同时包含 Windows 和 Linux 客户机操作系统，则该池将被视为 Windows 池，您将无法连接到 Linux 桌面。

当您授权用户使用桌面池中的 Linux 计算机时，最佳做法是确保用户在 Linux 客户机操作系统中没有管理特权。Linux 中的管理员用户可以打开终端窗口并调用命令（例如 `shutdown`，该命令用于关闭虚拟机的电源）。vCenter Server 管理员必须重新打开此虚拟机的电源。对非管理员的 Linux 用户授权可确保您不必手动管理这些电源操作。

### 前提条件

- 确认在 Linux 客户机操作系统上已安装 Horizon Agent。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。
- 确认已在 View 连接服务器中注册 Linux 虚拟机。在 View Administrator 中，选择 **View 配置 > 已注册的计算机**，然后选择**其他**选项卡。确认每个计算机的状态均为“可用”。
- 在 Ubuntu 14.04 计算机上，确认禁用了用户切换。请参阅[为远程桌面部署准备 Linux 计算机](#)中的步骤 9。

### 步骤

- 1 在 View Administrator 中，添加手动桌面池。  
选择**目录 > 桌面池 > 添加**。
- 2 选择**手动桌面池**。
- 3 为桌面池中的计算机选择专用或浮动用户分配。
- 4 在“计算机源”页面上，选择**其他源**。
- 5 在“添加计算机”页面上，选择您配置的 Linux 虚拟机，然后完成**添加桌面池**向导。  
请勿更改“远程显示协议”设置。这些设置在 Linux 桌面上无效。而且，最终用户无法选择显示协议。

---

**重要事项** 请仅添加 Linux 虚拟机。如果您添加 Windows 虚拟机，则池中的 Linux 桌面将不可用。

---

- 6 授权用户使用桌面池中的虚拟机。  
在 View Administrator 中，选择桌面池，选择**授权 > 添加授权**，然后添加用户或组。  
最佳做法是确保用户在 Linux 客户机操作系统中没有管理特权。

此时，Linux 虚拟机便已准备好在 Horizon 7 部署中用作远程桌面。

## 在 Linux 虚拟机上升级 Horizon Agent

您可以通过安装最新版本的 Horizon Agent 来升级 Linux 计算机上的 Horizon Agent。

之前安装时指定的大多数安装参数均会保留。您可以选择性地指定以下参数：

参数	说明
-A yes	接受 EULA。您必须指定 <b>yes</b> 才能继续进行安装。如果未指定该参数，安装脚本将提示输入该值。
-m yes	启用智能卡重定向。
-r	安装后重新引导操作系统。允许的值是 <b>yes</b> 和 <b>no</b> 。默认值为 <b>no</b> 。

**前提条件**

- 确认 VMwareBlastServer 进程未运行。

要停止该进程，请确保用户注销虚拟机并且没有活动桌面会话，或者重新引导虚拟机。

**步骤**

- 1 从 VMware 下载站点 <https://my.vmware.com/web/vmware/downloads> 下载最新的 Horizon Agent for Linux 安装程序文件。

在“桌面和最终用户计算”下，选择 VMware Horizon 7 下载，其中包括 Horizon Agent for Linux 安装程序。

安装程序文件名是 VMware-viewagent-linux-x86\_64-y.y.y-xxxxxxx.tar.gz（对于 64 位 Linux）或 VMware-viewagent-linux-y.y.y-xxxxxxx.tar.gz（对于 32 位 Linux），其中 y.y.y 是版本号，xxxxxxx 是内部版本号。

- 2 在客户机操作系统上解压缩 Linux 分发包的 tarball。

例如：

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

- 3 浏览到 tar ball 文件夹。

- 4 运行 install\_viewagent.sh 脚本。

例如：

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

**卸载和重新安装 Horizon 7 for Linux 计算机**

要在虚拟机上卸载 Horizon 7 for Linux，您必须从桌面池中移除未受管的计算机，从 View 中移除已注册的计算机，卸载 Horizon Agent，并移除配置文件。您可以通过运行 Horizon Agent 安装脚本来重新安装 Horizon 7 for Linux。

**从手动桌面池中移除未受管计算机**

您可以通过从池中移除未受管计算机来减少手动桌面池的大小。

**步骤**

- 1 在 View Administrator 中，选择目录 > 桌面池。
- 2 双击手动池的池 ID。
- 3 选择清单选项卡。
- 4 选择要移除的未受管计算机。
- 5 单击**移除**。

- 6 如果有用户已登录到基于计算机的未受管桌面中，请选择是终止其会话，还是允许会话继续活动。

选项	描述
保留为活动状态	活动会话将保留，直到用户注销为止。View 连接服务器不会跟踪这些会话。
终止	立即终止活动会话。

- 7 单击**确定**。

未受管计算机将从池中移除。

## 从 View 中移除已注册的计算机

如果您不打算再次使用某个已注册的计算机，可以从 View 中移除该计算机。

View 中有两种类型的已注册计算机：“RDS 主机”和“其他”。未受管的计算机属于“其他”类别。未受管的计算机包括不由 vCenter Server 管理的物理机和虚拟机，包括 Horizon 6 for Linux 虚拟机。它们用于构成不含 vCenter Server 虚拟机的手动桌面池。

移除某个已注册的计算机后，该计算机在 View 中将不可用。要使计算机再次可用，您必须重新安装 Horizon Agent。

### 前提条件

确认任何桌面池中均未使用您要移除的已注册的计算机。

### 步骤

- 1 在 View Administrator 中，选择 **View 配置 > 已注册的计算机**。
- 2 单击**其他**选项卡。
- 3 选择一个或多个计算机，然后单击**删除**。  
您只能选择未被桌面池使用的计算机。
- 4 单击**确定**进行确认。

## 在 Linux 虚拟机上卸载 Horizon Agent

要在 Linux 虚拟机上卸载 Horizon Agent，您可以使用 `uninstall_viewagent.sh` 脚本。您还必须从计算机中移除 Horizon 7 for Linux 配置文件。

### 前提条件

- 确认已从桌面池和 Horizon 7 中移除计算机。请参阅[从手动桌面池中移除未受管计算机](#)和[从 View 中移除已注册的计算机](#)。

## 步骤

- 1 在虚拟机上打开终端窗口，然后运行 Horizon Agent 卸载脚本。

例如：

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

此脚本将停止 Horizon Agent 进程，删除 Horizon Agent 服务和 Horizon Agent 安装目录 `/usr/lib/vmware/viewagent`。

- 2 手动删除位于 `/etc/vmware/` 目录中的 Horizon 7 for Linux 配置文件。

如果您要重新安装 Horizon Agent（例如，升级到更高的 Horizon Agent 版本），则不必删除这些配置文件。

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

## 在 Linux 虚拟机上重新安装 Horizon Agent

您可以采用不同的方法在 Linux 虚拟机上重新安装 Horizon Agent，具体采用哪种方法取决于您使用现有的 Horizon 7 for Linux 配置，还是使用全新的配置重新安装。

如果虚拟机中存在 Horizon 7 for Linux 配置文件，并且已在 View 连接服务器中注册该计算机，则您可以使用现有的配置来重新安装 Horizon Agent。

如果已从计算机中移除配置文件，或已从 View 连接服务器中移除注册的计算机名称，则您必须确保已同时移除本地配置文件和 View 连接服务器中注册的计算机名称。随后，您可以使用新的配置来执行 Horizon Agent 的全新安装。

### 前提条件

- 确认计算机上是否存在 Horizon 7 for Linux 配置文件。

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

- 确认是否已在 View 连接服务器中注册 Linux 虚拟机。在 View Administrator 中，选择 **View 配置 > 已注册的计算机**，然后选择其他选项卡。
- 如果已移除配置文件和注册的计算机名称，请了解有关执行 Horizon Agent 全新安装的步骤。请参阅[在 Linux 虚拟机上安装 Horizon Agent](#)。

**步骤**

- ◆ 要使用现有的 Horizon 7 for Linux 配置重新安装，您可以在不带任何参数的情况下运行 Horizon Agent 安装脚本。

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

Horizon Agent 安装程序将重新使用现有的配置文件和 View 连接服务器中的计算机注册。安装程序不会重新注册计算机。

- ◆ 要使用新的 Horizon 7 for Linux 配置重新安装，请运行带有必要参数的 Horizon Agent 安装脚本。  
安装步骤与全新安装的步骤相同。安装程序将向 View 连接服务器注册计算机。

## 如何在 Linux 桌面上从 vSphere 执行电源操作

有时，可能需要在 Linux 桌面上从 vSphere 执行电源操作。

请勿使用**关闭**或**重置**操作，因为这些操作可能会导致数据丢失或 Horizon Agent 无法正常重新启动。应使用**关闭客户机**或**重新启动客户机**操作。

## 收集有关 Horizon 7 for Linux 软件的信息

为了帮助对 Horizon 7 for Linux 软件进行管理和故障排除，您可以收集有关 Linux 虚拟机上 Horizon Agent 安装的信息。

**步骤**

- ◆ 识别在 Linux 计算机上运行的 Horizon Agent 版本。

运行以下命令：

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

输出将显示 Horizon Agent 版本。例如：

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

其中 **x.x** 是产品版本，**yyyyyyy** 是内部版本号。

- ◆ 收集有关 Horizon Agent 软件的信息。

文件	位置
Horizon Agent 安装目录	/usr/lib/vmware/viewagent
日志	/var/log/vmware/
Horizon Agent 配置	/etc/vmware/viewagent-config.txt
计算机配置	/etc/vmware/viewagent-machine.cfg

- ◆ 停止、启动或重新启动 Horizon Agent 服务。

Linux 分发版中提供了包含这些操作的 Horizon Agent 服务。例如，在 Ubuntu 12.04 上，您可以运行以下命令：

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



## 批量部署 Horizon 7 for Linux 桌面

使用 View Administrator，您可以自动创建 Windows（而不是 Linux）桌面计算机池。但是，您可以开发用于自动部署 Linux 桌面计算机池的脚本。

提供的示例脚本仅作为说明之用。对于在使用示例脚本时可能出现的问题，VMware 概不负责。

本章讨论了以下主题：

- [Linux 桌面批量部署概览](#)
- [为克隆 Linux 桌面计算机创建虚拟机模板](#)
- [用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)
- [用于克隆 Linux 虚拟机的示例脚本](#)
- [用于将克隆的虚拟机加入 AD 域的示例脚本](#)
- [使用 SSH 将克隆的虚拟机加入 Active Directory 域的示例脚本](#)
- [用于在 Linux 虚拟机上安装 Horizon Agent 的示例脚本](#)
- [使用 SSH 在 Linux 虚拟机上安装 Horizon Agent 的示例脚本](#)
- [用于将配置文件上载至 Linux 虚拟机的示例脚本](#)
- [使用 SSH 将配置文件上载至 Linux 虚拟机的示例脚本](#)
- [用于在 Linux 桌面计算机上升级 Horizon Agent 的示例脚本](#)
- [使用 SSH 在 Linux 虚拟机上升级 Horizon Agent 的示例脚本](#)
- [用于在 Linux 虚拟机上执行操作的示例脚本](#)
- [用于从连接服务器 LDAP 数据库中删除计算机的示例脚本](#)

### Linux 桌面批量部署概览

部署 Linux 桌面涉及若干步骤。如果您计划部署多个桌面，则可以使用 PowerCLI 脚本来自动完成某些步骤。

对于某些操作，您可以选择使用 PowerCLI 或 SSH 在 Linux 计算机上执行命令。下表介绍了这两种方法的区别。

PowerCLI	SSH
无需安装额外工具。	<ul style="list-style-type: none"> <li>■ 对于 Ubuntu，您需要使用命令 <code>sudo apt-get install openssh-server</code> 安装 SSH 服务器。对于 RHEL 和 CentOS，<code>openssh-server</code> 是默认安装的，但您需要确保防火墙设置允许 SSH。</li> <li>■ 需要下载 SSH 客户端应用程序 <code>pscp.exe</code> 和 <code>plink.exe</code>，并将它们和 PowerCLI 脚本置于同一个文件夹。</li> </ul>
上载文件和命令执行速度较慢。	上载文件和命令执行速度较快。
需要提供 ESXi 主机的管理员凭据。	不需要提供 ESXi 主机的管理员凭据。
在运行用于安装 Horizon Agent 的脚本时无法处理管理员密码中的特殊字符，或在运行用于加入域脚本时无法处理 AD 用户密码中的特殊字符。	在运行用于安装 Horizon Agent 的脚本时可以处理管理员密码中的特殊字符，或在运行用于加入域脚本时可以处理 AD 用户密码中的特殊字符。

**注** 基于 PowerCLI 的脚本和基于 SSH 的脚本均可以处理 vCenter Server 管理员和 Linux 管理员的密码中的特殊字符。基于 PowerCLI 的脚本还可以处理 ESXi 主机管理员密码中的特殊字符。在以上所有情况下，均不必使用转义字符。

有关 vSphere PowerCLI 的更多信息，请参阅 <https://www.vmware.com/support/developer/PowerCLI>。

Linux 桌面池的批量部署过程涉及以下步骤：

#### 1 创建虚拟机模板。

请参阅[为克隆 Linux 桌面计算机创建虚拟机模板](#)。

#### 2 创建客户机自定义规范。

请参阅《vSphere 虚拟机管理指南》文档中的“在 vSphere Web Client 中为 Linux 创建自定义规范”。创建规范时，确保您正确指定以下设置。

设置	值
目标虚拟机操作系统	Linux
计算机名称	使用虚拟机名称。
域	指定 View 环境的域。
网络设置	使用标准网络设置。
主 DNS	指定有效地址。

**注** 有关客户机操作系统自定义支持表的更多信息，请参阅 <http://partnerweb.vmware.com/programs/guestOS/guest-os-customization-matrix.pdf>。

#### 3 克隆虚拟机。

请参阅[用于克隆 Linux 虚拟机的示例脚本](#)。

#### 4 将克隆的虚拟机加入 Active Directory (AD) 域中。

请参阅[用于将克隆的虚拟机加入 AD 域的示例脚本](#)或使用 SSH 将克隆的虚拟机加入 Active Directory 域的示例脚本。

## 5 在虚拟机中安装 Horizon Agent。

请参阅[用于在 Linux 虚拟机上安装 Horizon Agent 的示例脚本](#)或使用 SSH 在 Linux 虚拟机上安装 Horizon Agent 的示例脚本。

## 6 更新虚拟机中的配置选项。

请参阅[用于将配置文件上载至 Linux 虚拟机的示例脚本](#)或使用 SSH 将配置文件上载至 Linux 虚拟机的示例脚本。

## 7 创建桌面池。

请参阅[创建包含 Linux 虚拟机的桌面池](#)。

有时，您可能需要在 Linux 虚拟机上升级 Horizon Agent。有关用于在虚拟机上升级 Horizon Agent 的示例脚本，请参阅[用于在 Linux 桌面计算机上升级 Horizon Agent 的示例脚本](#)或使用 SSH 在 Linux 虚拟机上升级 Horizon Agent 的示例脚本。

有关执行打开电源、关闭、重新启动或删除虚拟机等操作的示例脚本，请参阅[用于在 Linux 虚拟机上执行操作的示例脚本](#)。该脚本可以从 vCenter Server 中删除虚拟机。要从 View 中删除桌面计算机，请参阅[用于从连接服务器 LDAP 数据库中删除计算机的示例脚本](#)。

# 为克隆 Linux 桌面计算机创建虚拟机模板

在执行虚拟机克隆之前，您必须创建克隆所基于的虚拟机模板。

### 前提条件

- 确认您的部署满足支持 Linux 桌面的相关要求。请参阅[Horizon 7 for Linux 的系统要求](#)。
- 熟悉有关在 vCenter Server 中创建虚拟机和安装客户机操作系统的步骤。请参阅《在 View 中设置桌面和应用程序池》文档中的“创建和准备虚拟机”。
- 熟悉您将用于虚拟机的监视器的建议显存 (vRAM) 值。请参阅[Horizon 7 for Linux 的系统要求](#)。
- 熟悉 AD 集成的步骤。请参阅[第 3 章 为 Linux 桌面设置 Active Directory 集成](#)。
- 如果需要，熟悉使用 View 配置文件配置选项的步骤。请参阅[第 4 章 Linux 桌面的配置选项](#)。
- 如果您打算设置图形，请熟悉相关步骤。请参阅[第 5 章 为 Linux 桌面设置图形](#)。

### 步骤

- 1 在 vSphere Web Client 或 vSphere Client 中，创建新虚拟机。

**2 配置自定义配置选项。**

a 右键单击虚拟机，然后单击**编辑设置**。

b 指定 vCPU 数量和显存大小。

有关建议的值，请遵循您的 Linux 分发安装指南中的准则。

例如，Ubuntu 12.04 建议配置 2048 MB 的显存和 2 个 vCPU。

c 选择**视频卡**并指定显示器数量和显存 (vRAM) 总大小。

对于使用 2D 或 vSGA 的虚拟机（使用 VMware 驱动程序），请在 vSphere Web Client 中设置 vRAM 大小。vRAM 大小对 vDGA 或 NVIDIA GRID vGPU 计算机（使用 NVIDIA 驱动程序）无效。

有关建议的值，请遵循《Horizon 7 for Linux 的系统要求》中的准则。请勿使用显存计算器。

**3 打开虚拟机的电源并安装 Linux 分发包。****4 创建具有 root 特权的用户，例如 ViewUser。此用户仅用于安装和卸载 Horizon Agent。****5 编辑 /etc/sudoers 并添加行 ViewUser ALL=(ALL) NOPASSWD:ALL。**

如果 /etc/sudoers 中包含此行，则无需密码就能以 ViewUser 身份运行 sudo。当您通过运行本章中提供的示例脚本来安装 Horizon Agent 时，需要将 ViewUser 指定为输入。

**6 如果 Linux 分发包是 RHEL、CentOS 或 NeoKylin，请编辑 /etc/sudoers 并注释掉以下行：**

```
Defaults requiretty
Defaults !visiblepw
```

**7 如果 Linux 分发包不是 RHEL 7、CentOS 7 或 SLED 12，请安装 VMware Tools。**

默认情况下，RHEL 7、CentOS 7 和 SLED 12 已安装 Open VM Tools。

**8 如果 Linux 分发包是 RHEL 7、CentOS 7 或 SLED 12，请安装 deployPkg 插件。**

<http://kb.vmware.com/kb/2075048> 中提供了相应的说明。

**9 对于 RHEL 和 CentOS，启用网络连接设置“自动连接”。****10 执行 AD 集成步骤。****11 使用 View 配置文件执行其他配置。****12 执行步骤设置图形。****13 关闭虚拟机并创建快照。**

## 用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件

用于部署 Linux 桌面的示例 PowerCLI 脚本会读取一个包含桌面计算机信息的输入文件。

输入文件的类型为 csv，其中包含以下信息：

- 桌面虚拟机名称

- 父虚拟机名称
- 客户机自定义规范
- 克隆的桌面计算机所在的数据存储
- 托管桌面计算机的 ESXi Server
- 用于克隆的父虚拟机的快照
- 用于指示是否删除桌面虚拟机（如果存在）的标记

以下示例显示了输入文件可能包含的内容。

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

在示例脚本中，假定此输入文件名称为 `CloneVMs.csv`，并且该文件与脚本位于同一文件夹中。

## 用于克隆 Linux 虚拟机的示例脚本

您可以自定义和使用以下示例脚本来克隆任意数量的虚拟机 (VM)。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

### 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- 克隆类型，可以是链接克隆或完整克隆
- 是否禁用 vSphere 虚拟机控制台

### 脚本内容

```
<#
Create Clones from a Master VM

The Tool supports creation of Full clone and linked clone from Master VM.
The parent VM is required for the linked-clone to work and the parent VMs file cannot be renamed or
moved.
#>
#----- Functions -----
```

```

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

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

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

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

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

function Delete_VM()
{
    Param($VMToDelete)

```

```

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

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

$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
$cloneType = GetInput -prompt 'Clone Type ("linked" or "full")' -IsPassword $false
$disableVMConsole = GetInput -prompt 'Disable vSphere VM Console ("yes" or "no", recommend "yes")' -
IsPassword $false
"-----"
$csvFile = '.\CloneVMs.csv'

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

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

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

#Read input CSV file
$csvData = Import-CSV $csvFile
#$csvData = Import-CSV $csvFile -
header("VMName","Parentvm","CustomSpec","Datastore","Host","FromSnapshot","DeleteIfPresent")
foreach ($line in $csvData)
{

```

```

" `n-----"
$VMName = $line.VMName
write-host -ForegroundColor Yellow "VM: $VMName`n"

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

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

```



```
Start-VM $newvm
}
Disconnect-VM $vcAddress -Confirm:$false
exit
```

## 脚本执行

下面是执行脚本时显示的消息：

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

克隆过程所用的时间取决于桌面计算机的数量，可能会持续几分钟至数小时。要验证此过程是否已完成，请在 **vSphere Client** 中确保最后一个桌面虚拟机的电源已开启并具有唯一的主机名，同时 **VMware Tools** 处于运行状态。

## 用于将克隆的虚拟机加入 AD 域的示例脚本

您可以自定义并使用以下示例脚本来将克隆的虚拟机 (VM) 加入 **Active Directory (AD)** 域。

如果使用 **Winbind** 解决方案进行 **AD** 集成，您需要运行此脚本，因为将克隆的虚拟机加入域的步骤会失败。此脚本会运行一个命令以在每个虚拟机上加入域。如果使用 **OpenLDAP** 解决方案，则您无需运行此脚本。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 **HTML** 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 **Horizon 7** 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- ESXi 主机的管理员登录名称
- ESXi 主机的管理员密码
- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码
- 有权将计算机加入域中的 **AD** 用户的登录名称
- 获授权的 **AD** 用户的密码

## 脚本内容

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

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

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

    [Console]::ResetColor()
    return $input
}
#----- Handle input -----
"-----"
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$adUser = GetInput -prompt 'Type the AD user name to join the AD' -IsPassword $false
""
"Please type the AD user password."
"Please note that special character in password may not work with the script"
$adUserPassword = GetInput -prompt 'Your AD user password' -IsPassword $true
"-----"

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

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

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

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

$destFolder = "/home/$guestUser/"

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

    $cmd = "sudo /usr/bin/net ads join -U $adUser%$adUserPassword"
    Write-Host "Run cmd 'sudo /usr/bin/net ads join' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
}

Disconnect-VIServer $vcAddress -Confirm:$false
exit
```

## 脚本执行

下面是执行脚本时显示的消息：

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

-----

Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

-----

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

-----

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

-----
```

```
Type the AD user name to join the AD: viewadmin
Please type the AD user password.
Please note that special character in password may not work with the script.
Your AD user password: *****
```

## 使用 SSH 将克隆的虚拟机加入 Active Directory 域的示例脚本

您可以自定义并使用以下示例脚本来将克隆的虚拟机 (VM) 加入 Active Directory (AD) 域。此脚本会使用 SSH 在 Linux 虚拟机上运行命令。

如果使用 Winbind 解决方案进行 AD 集成，您需要运行此脚本，因为将克隆的虚拟机加入域的步骤会失败。此脚本会运行一个命令以在每个虚拟机上加入域。如果使用 OpenLDAP 解决方案，则您无需运行此脚本。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

### 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码
- 有权将计算机加入域中的 AD 用户的登录名称
- 获授权的 AD 用户的密码

### 脚本内容

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

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

.NOTES
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
```

```

[Console]::ForegroundColor = "Blue"
if ($IsPassword)
{
    $input = Read-Host -AsSecureString
    $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
}
else
{
    $input = Read-Host
}

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

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

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

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

```

```

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

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

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

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

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

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

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

```

```

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

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

$destFolder = "/home/$guestUser/"

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

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

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

## 脚本执行

下面是执行脚本时显示的消息：

```

PowerCLI C:\scripts> .\ClonedVMs_JoinDomain_SSH.ps1

-----
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

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

-----
Type the AD user name to join the AD: viewadmin
Please type the AD user password.
Please note that special character should be escaped. For example, $ should be \$
Your AD user password: *****

```

## 用于在 Linux 虚拟机上安装 Horizon Agent 的示例脚本

您可以自定义并使用以下示例脚本在多个 Linux 虚拟机 (Virtual Machine, VM) 上安装 Horizon Agent。

此脚本会先将安装程序 tar ball 上载至每个虚拟机，然后再安装 Horizon Agent。上载任务可能非常耗时，在涉及大量虚拟机和网络速度缓慢时尤其如此。为节省时间，您可以运行使用 SSH 的脚本，或者将安装程序 tar ball 置于每个虚拟机均可访问的共享的位置，这样就不必上载文件。

在运行脚本之前，请确保尚未在虚拟机上安装 Horizon Agent。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- 接受 Horizon Agent EULA（最终用户许可协议）
- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- ESXi 主机的管理员登录名称
- ESXi 主机的管理员密码
- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码
- Horizon Agent tar ball 路径
- View 连接服务器的 IP 地址
- View 连接服务器的管理员登录名称
- View 连接服务器的管理员密码
- View 连接服务器管理员的域名
- AD 服务器地址（如果 Horizon Agent 在向 View 连接服务器注册 Linux 计算机时使用 Kerberos 身份验证方法）。

## 脚本内容

```
<#
Install Linux Agent

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

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



```

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

[Console]::ResetColor()
return $input
}
#----- Handle Input -----
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----"
$brokerAddress = GetInput -prompt 'Type the View Connection Server address' -IsPassword $false
$brokerAdminName = GetInput -prompt 'Type the View Connection Server Admin user name' -IsPassword
$false
"$nPlease type the View Connection Server Admin user password."
"Plase note that special character in password may not work with the script"
$brokerAdminPassword = GetInput -prompt 'Your broker admin password' -AsSecureString -IsPassword
$true
$domainName = GetInput -prompt 'Type the View Connection Server Admin user domain name' -IsPassword
$false
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$skdc = GetInput -prompt 'AD address for Kerberos authentication to register. Please type Enter to
skip for MD5-Digest' -IsPassword $false
"-----"
#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

```

```

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

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

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

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

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

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

$destFolder = "/home/$guestUser/"

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

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

```

```

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

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

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

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

    $cmd = "sudo setenforce 0";
    Write-Host "Set the selinux to permissive mode: $cmd"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

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

    $cmd = "sudo shutdown -r +1&"
    Write-Host "Reboot to apply the Horizon Agent installation"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
}
else
{
    Write-Host $VMName": Uploaded installer's MD5Sum does NOT match the local installer's MD5Sum";
    Write-Host $VMName": Skip the installation. Please check your network and VMware Tools
status";
    exit;
}
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

## 脚本执行

下面是执行脚本时显示的消息：

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

-----
Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

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

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

-----
Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz

-----
Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character in password may not work with the script
Your broker admin password: *****
Type the View Connection Server Admin user domain name: domain1
Install the Smartcard redirection feature ("yes" or "no"): no
AD address for Kerberos authentication to register. Please type Enter to skip for MD5 Digest:
10.117.45.1
```

脚本执行完成之后，您可以登录到 View Administrator，然后通过浏览到 **View 配置 > 已注册的计算机** 并单击 **其他** 选项卡来查看 Linux 桌面计算机。

## 使用 SSH 在 Linux 虚拟机上安装 Horizon Agent 的示例脚本

您可以自定义并使用以下示例脚本在多个 Linux 虚拟机 (Virtual Machine, VM) 上安装 Horizon Agent。此脚本会使用 SSH 在 Linux 虚拟机上运行命令。

在运行脚本之前，请确保尚未在虚拟机上安装 Horizon Agent。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- 接受 Horizon Agent EULA（最终用户许可协议）
- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称

- vCenter Server 的管理员密码
- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码
- Horizon Agent tar ball 路径
- View 连接服务器的 IP 地址
- View 连接服务器的管理员登录名称
- View 连接服务器的管理员密码
- View 连接服务器管理员的域名
- AD 服务器地址（如果 Horizon Agent 在向 View 连接服务器注册 Linux 计算机时使用 Kerberos 身份验证方法）。

## 脚本内容

```
<#
Install Linux Agent via SSH

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

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

    [Console]::ResetColor()
    return $input
}
function Check_SSH_Client
{
    Param($IsPlink, $IsPSCP)
    if ($IsPlink)
    {
        if (Test-Path ".\plink.exe")
        {

```

```

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

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

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

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

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

```

```

#----- Handle Input -----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----"
$brokerAddress = GetInput -prompt 'Type the View Connection Server address' -IsPassword $false
$brokerAdminName = GetInput -prompt 'Type the View Connection Server Admin user name' -IsPassword
$false
"$nPlease type the View Connection Server Admin user password."
[Console]::ForegroundColor = "Yellow"
"Plase note that special character should be escaped. For example, $ should be \$\"
[Console]::ResetColor()
$brokerAdminPassword = GetInput -prompt 'Your broker admin password' -AsSecureString -IsPassword
$true
$domainName = GetInput -prompt 'Type the View Connection Server Admin user domain name' -IsPassword
$false
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$skdc = GetInput -prompt 'AD address for Kerberos authentication to register. Please type Enter to
skip for MD5-Digest' -IsPassword $false
"-----"
#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

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

#check if file exists
if (!(Test-Path $csvFile))
{

```

```

write-host -ForegroundColor Red "CSV File not found"
exit
}

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

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

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

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

$destFolder = "/home/$guestUser/"

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

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

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

```



```

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

if($output.Contains($installerMD5Hash))
{
    Write-Host $VMName": Uploaded installer's MD5Sum matches the local installer's MD5Sum";
    Write-Host $VMName": Extract the installer and do installation";
    $cmd = "tar -xzf VMware-viewagent-linux-*.tar.gz"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

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

    #Run the installation command.
    $cmd = "cd VMware-viewagent-linux-* && sudo ./install_viewagent.sh -r yes -A yes -n $VMName -
b $brokerAddress -d $domainName -u $brokerAdminName -p $brokerAdminPassword -m $installSmartcard"
    if (![string]::IsNullOrEmpty($kdc))
    {
        $cmd = $cmd + " -k $kdc"
    }
    Write-Host "Run install cmd in VM '$VMName' with user '$guestUser'"
    RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd
    Write-Host -ForegroundColor Yellow "Linux Agent installer will reboot the Linux VM after
installation, and you may hit the ssh connection closed error message, which is expectation"
}
else
{
    Write-Host $VMName": Uploaded installer's MD5Sum does NOT match the local installer's MD5Sum";
    Write-Host $VMName": Skip the installation. Please check your network and VMware Tools
status";
    exit;
}
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

## 脚本执行

下面是执行脚本时显示的消息：

```

PowerCLI C:\scripts> .\InstallAgent.ps1
-----
Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****

```

```

-----
Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
-----
Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character should be escaped. For example, $ should be \$
Your broker admin password: *****
Type the View Connection Server Admin user domain name: domain1
Install the Smartcard redirection feature ("yes" or "no"): no
AD address for Kerberos authentication to register. Please type Enter to skip for MD5 Digest:
10.117.45.1

```

脚本执行完成之后，您可以登录到 **View Administrator**，然后通过浏览到 **View 配置 > 已注册的计算机** 并单击 **其他** 选项卡来查看 Linux 桌面计算机。

## 用于将配置文件上载至 Linux 虚拟机的示例脚本

您可以自定义并使用以下示例脚本来将配置文件 `config` 和 `viewagent-custom.conf` 上载至多个 Linux 虚拟机 (VM)。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

### 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#) 中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- ESXi 主机的管理员登录名称
- ESXi 主机的管理员密码
- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码

### 脚本内容

```

<#
Upload the configuration files config and viewagent-custom.conf to Linux VMs
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "

```

```

Write-Host $prompt -NoNewLine
[Console]::ForegroundColor = "Blue"
if ($IsPassword)
{
    $input = Read-Host -AsSecureString
    $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($input))
}
else
{
    $input = Read-Host
}

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

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

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

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

if(Test-Path $customConf_File)
{
    $setCustomConf = $true
    write-host -ForegroundColor Yellow '"viewagent-custom.conf" file found'
}

```

```

else
{
    write-host -ForegroundColor Yellow '"viewagent-custom.conf" file not found, skip it'
}

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

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

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

$destFolder = "/home/$guestUser/"

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

    #Try to delete the configuration file from home folder on destination VM
    $cmd = "rm -rf config viewagent-custom.conf"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
    $guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

    if ($setConfig)
    {
        Write-Host "Upload File '$config_File' to '$destFolder' of VM '$VMName' with user '$guestUser'"
        Copy-VMGuestFile -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
    $guestUser -GuestPassword $guestPassword -Confirm:$false -LocalToGuest -Destination $destFolder -
    Source $config_File

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

```

```

    if ($setCustomConf)
    {
        Write-Host "Upload File '$customConf_File' to '$destFolder' of VM '$VMName' with user '$guestUser'"
        Copy-VMGuestFile -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser $guestUser -GuestPassword $guestPassword -Confirm:$false -LocalToGuest -Destination $destFolder -Source $customConf_File

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

```

## 脚本执行

下面是执行脚本时显示的消息：

```

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

```

## 使用 SSH 将配置文件上载至 Linux 虚拟机的示例脚本

您可以自定义并使用以下示例脚本来将配置文件 `config` 和 `viewagent-custom.conf` 上载至多个 Linux 虚拟机 (VM)。此脚本会使用 SSH 在 Linux 虚拟机上运行命令。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码

- Linux 虚拟机的用户登录名称
- Linux 虚拟机的用户密码

## 脚本内容

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

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

```

        {
            write-host -ForegroundColor Red 'SSH client "pscp.exe" not found, please download from its
official web site'
            exit
        }
    }
}

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

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

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

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

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

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

```

```

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

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

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

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

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

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

$destFolder = "/home/$guestUser/"

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

```



```

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

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

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

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

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

```

## 脚本执行

下面是执行脚本时显示的消息：

```

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

```

## 用于在 Linux 桌面计算机上升级 Horizon Agent 的示例脚本

您可以自定义并使用以下示例脚本在多个 Linux 虚拟机 (Virtual Machine, VM) 上升级 Horizon Agent。

此脚本会先将安装程序 tar ball 上载至每个虚拟机，然后再安装 Horizon Agent。上载任务可能非常耗时，在涉及大量虚拟机和网络速度缓慢时尤其如此。为节省时间，您可以运行使用 SSH 的脚本，或者将安装程序 tar ball 置于每个虚拟机均可访问的共享的位置，这样就不必上载文件。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- 接受 Horizon Agent EULA（最终用户许可协议）
- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- ESXi 主机的管理员登录名称
- ESXi 主机的管理员密码
- Linux 客户机操作系统的用户登录名称
- Linux 客户机操作系统的用户密码
- Horizon Agent tar ball 路径

## 脚本内容

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

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

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

```

#----- Handle Input -----
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$svcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$svcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$svcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$hostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
$hostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----"
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"-----"

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

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

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

#----- Functions -----
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =

```

```

[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;
    $md5Hash = ($md5HashWithFormat.replace("-", "").ToLower());
    return $md5Hash;
}

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

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

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

```

```

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

    $cmd = "sudo setenforce 0";
    Write-Host "Set the selinux to permissive mode: $cmd"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd

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

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

    $cmd = "sudo shutdown -r +1&"
    Write-Host "Reboot to apply the Horizon Agent installation"
    Invoke-VMScript -HostUser $hostAdmin -HostPassword $hostPassword -VM $VMName -GuestUser
$guestUser -GuestPassword $guestPassword -Confirm:$false -ScriptType Bash -ScriptText $cmd
}
else
{
    Write-Host $VMName": Uploaded installer's MD5Sum does NOT match the local installer's MD5Sum";
    Write-Host $VMName": Skip the installation. Please check your network and VMware Tools
status";
    exit;
}
}
Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

## 脚本执行

下面是执行脚本时显示的消息：

```

PowerCLI C:\scripts> .\InstallAgent.ps1
-----
Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****
-----
Your ESXi host admin user name, such as root: root
Your ESXi host admin user password: *****
-----
Your VM guest OS user name: ViewUser
Your VM guest OS user password: *****

```

```

-----
Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
Install the Smartcard redirection feature ("yes" or "no"): no

```

## 使用 SSH 在 Linux 虚拟机上升级 Horizon Agent 的示例脚本

您可以自定义并使用以下示例脚本在多个 Linux 虚拟机 (Virtual Machine, VM) 上升级 Horizon Agent。此脚本会使用 SSH 在 Linux 虚拟机上运行命令。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

### 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- 接受 Horizon Agent EULA（最终用户许可协议）
- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- ESXi 主机的管理员登录名称
- ESXi 主机的管理员密码
- Linux 客户机操作系统的用户登录名称
- Linux 客户机操作系统的用户密码
- Horizon Agent tar ball 路径

### 脚本内容

```

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

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

```

```

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

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

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

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

```

```

else
{
    echo yes | .\plink.exe -ssh -l $user -pw $password $IP "$cmd"
}
}

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

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

#----- Handle Input -----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"
$guestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
$guestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
$agentInstaller = GetInput -prompt 'Type the Horizon Agent tar ball path. Please take care the
installer arch' -IsPassword $false
"-----"
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"-----"

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

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

```



```

exit
}

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

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

#----- Main -----

#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

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

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

$destFolder = "/home/$guestUser/"

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

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

```

```

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

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

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

    $cmd = "tar -xzf VMware-viewagent-linux-*.tar.gz"
    Write-Host "Run cmd '$cmd' in VM '$VMName' with user '$guestUser'"
    RunCmdViaSSH -VM_Name $VMName -User $guestUser -Password $guestPassword -Cmd $cmd

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

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

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

```

## 脚本执行

下面是执行脚本时显示的消息：

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

-----

Accept Linux Horizon Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

-----

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

-----

Type the Horizon Agent tar ball path. Please take care of the installer arch: .\VMware-viewagent-
linux-x86_64-x.y.z-1234567.tar.gz
Install the Smartcard redirection feature ("yes" or "no"): no
```

## 用于在 Linux 虚拟机上执行操作的示例脚本

您可以自定义和使用以下示例脚本来在多个虚拟机 (VM) 上执行操作。这些操作包括打开电源、关闭电源、关闭、重新启动和删除虚拟机。

该脚本可以从 vCenter Server 中删除虚拟机，但不能从 View 中删除虚拟机。要从 View 中删除计算机，请参阅示例脚本[用于从连接服务器 LDAP 数据库中删除计算机的示例脚本](#)。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

## 脚本输入

此脚本将读取一个输入文件，[用于部署 Linux 桌面的示例 PowerCLI 脚本的输入文件](#)中对此做了介绍。此脚本还会以交互方式要求提供以下信息：

- vCenter Server 的 IP 地址
- vCenter Server 的管理员登录名称
- vCenter Server 的管理员密码
- 要执行的操作，可以是打开电源、关闭电源、关闭客户机、重新启动虚拟机、重新启动虚拟机客户机或删除虚拟机。
- 虚拟机上的操作之间的等待时间（以秒为单位）。

## 脚本内容

```
<#
.DESCRIPTION
The Tool supports:
1. Power off VMs
2. Power on VMs
3. Shutdown VMs
```

```

4. Restart VMs
5. Restart VM guest
6. Delete VMs from Disk
.NOTES
#>

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

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

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

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

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

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

```

```

#----- Handle input -----
"-----"
SvcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
SvcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
SvcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"

$action = GetInput -prompt 'Select action: 1). Power On 2). Power Off 3) Shutdown VM Guest 4).
Restart VM 5). Restart VM Guest 6). Delete VM' -IsPassword $false
$sleepTime = GetInput -prompt 'Wait time (seconds) between each VM' -IsPassword $false
"-----"

[Console]::ForegroundColor = "Yellow"
switch ($action)
{
    1
    {
        "Your selection is 1). Power On"
    }
    2
    {
        "Your selection is 2). Power Off"
    }
    3
    {
        "Your selection is 3) Shutdown"
    }
    4
    {
        "Your selection is 4). Restart VM"
    }
    5
    {
        "Your selection is 5). Restart VM Guest"
    }
    6
    {
        "Your selection is 6). Delete VM"
    }
    default
    {
        "Invalid selection for action: $action"
        exit
    }
}

[Console]::ResetColor()
$csvFile = '.\CloneVMs.csv'

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

#----- Main -----

```

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

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

Disconnect-VIServer $vcAddress -Confirm:$false
exit
```

## 脚本执行

下面是执行脚本时显示的消息：

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

```
-----
Select action: 1). Power On 2). Power Off 3) Shutdown VM Guest 4). Restart VM 5). Restart VM Guest
6). Delete VM: 1
Wait time (seconds) between each VM: 20
-----
Your selection is 6). Delete VM
```

对于打开电源、重新启动虚拟机和重新启动虚拟机客户机操作，将虚拟机之间的等待时间指定为至少 20 秒可避免出现引导风暴的情况，否则可能导致某些操作失败。

## 用于从连接服务器 LDAP 数据库中删除计算机的示例脚本

您可以自定义并使用以下示例脚本从连接服务器 LDAP 数据库中删除未位于桌面池中的计算机。

这是一个 PowerShell 脚本，不需要安装 PowerCLI。要运行该脚本，请以管理员身份登录到 View 连接服务器，创建该脚本和输入文件，然后从 PowerShell 窗口中运行该脚本。如果输入文件中的计算机位于桌面池中，该脚本将显示一条消息，并且不会从 LDAP 数据库中删除该计算机。

**注** 在运行该脚本之前，请拍摄连接服务器虚拟机的快照，以防止需要撤消运行该脚本所做的更改。

要复制和粘贴不包含分页符的脚本内容，请使用此主题的 HTML 版本，您可以从位于 [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) 的 Horizon 7 文档页面找到该版本。

### 脚本输入

输入文件的类型为 csv，并包含要删除的虚拟机名称。以下示例显示了输入文件可能包含的内容。

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

此示例脚本假定该输入文件的名称为 DeleteMachineItems.csv，并且该文件位于与脚本相同的文件夹中。

### 脚本内容

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

$csvFile = '.\DeleteMachineItems.csv'

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

function IsNameInCsv($name)
```

```

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

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

```



## 对 Linux 桌面进行故障排除

在管理 Linux 桌面时可能出现某些问题。您可以按照不同的步骤诊断并解决问题。

本章讨论了以下主题：

- 收集 Horizon 7 for Linux 计算机的诊断信息
- 对 Linux 计算机的 Horizon Agent 注册失败问题进行故障排除
- 对无法访问 Linux 计算机上的 Horizon Agent 问题进行故障排除
- 对 Linux 计算机上的 Horizon Agent 没有响应问题进行故障排除
- 对远程桌面和客户端主机之间的复制和粘贴问题进行故障排除
- 配置 Linux 防火墙以允许传入的 TCP 连接

### 收集 Horizon 7 for Linux 计算机的诊断信息

您可以收集诊断信息以协助 VMware 技术支持部门诊断并解决 Horizon 7 for Linux 计算机存在的问题。您可以创建数据收集工具 (Data Collection Tool, DCT) 捆绑包，用来收集计算机的配置信息并记录到压缩的 tarball 中。

#### 步骤

- 1 以具有必需特权的用户身份登录 Linux 虚拟机。
- 2 打开命令提示符，然后运行 `dct-debug.sh` 脚本。

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

该脚本将生成包含 DCT 捆绑包的 tarball。例如：

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

将在执行该脚本的目录（当前的工作目录）中生成 tarball。

### 对 Linux 计算机的 Horizon Agent 注册失败问题进行故障排除

Horizon Agent 安装程序未能向 View 连接服务器注册 Linux 虚拟机。

#### 问题

在 Linux 虚拟机上安装 Horizon Agent 时，显示代理注册失败 (Agent Registration Failed) 错误。

## 原因

Horizon Agent 安装程序可能由于以下原因未能向 View 连接服务器注册 Linux 计算机：

- 在 Horizon Agent 安装过程中提供的 View 连接服务器 FQDN、域名、用户名或密码不正确。
- View Administrator 用户在 View Administrator 中没有代理注册管理员或管理员角色。具体而言，需要注册代理特权才能向 View 连接服务器注册 Horizon Agent。代理注册管理员是提供此最低特权的受限角色。
- View Administrator 密码包含未使用反斜杠进行转义的特殊字符（例如 \$）。
- 未对管理员的 Active Directory (AD) 帐户启用可逆密码加密。有关启用此策略的详细信息，请参阅[启用可逆密码加密](#)。

## 解决方案

- 1 确认您拥有正确的 View 连接服务器 FQDN、域名、用户名和密码。
- 2 确认 View 管理员用户拥有可允许进行 Horizon Agent 注册的特权。
  - a 在 View Administrator 中，选择 **View 配置 > 管理员**。
  - b 在“管理员和组”选项卡中，选择您在 Horizon Agent 安装过程中提供的 View 管理员用户。
  - c 确认该用户拥有代理注册管理员或管理员角色。
- 3 如果 View Administrator 密码包含特殊字符，请使用反斜杠对其进行转义。  
例如：ab\ \$cdef
- 4 在 Linux 计算机上，打开 Horizon Agent 注册日志文件，并查找错误消息。

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

- 5 如果身份验证机制是 DIGEST-MD5，即未指定 **-k** 选项，请确保已为指定的用户启用 AD 策略使用可逆加密存储密码。
- 6 如果身份验证机制是 Kerberos，请确保 **-k** 选项指定 AD 服务器的正确的地址，且 **-b** 选项指定连接服务器实例的计算机全名（包括主机名和域名）。
- 7 在 Linux 虚拟机上重新安装 Horizon Agent。

# 对无法访问 Linux 计算机上的 Horizon Agent 问题进行故障排除

View 连接服务器无法与 Linux 虚拟机上的 Horizon Agent 通信。

## 问题

在 View Administrator 中，Linux 虚拟机显示为无法访问代理 (Agent Unreachable)。在 Linux 计算机上安装了 Horizon Agent 并成功启动了 Horizon Agent 服务后，会发生此问题。

## 原因

一个原因可能是 Linux 计算机无法解析 View 连接服务器实例的 FQDN。

安装后，Horizon Agent 会将 View 连接服务器实例的 FQDN 列表存储在 `/etc/vmware/viewagent-machine.cfg` 配置文件中。即使您在 Horizon Agent 安装过程中使用 `-b` 参数指定 View 连接服务器实例的 IP 地址，Horizon Agent 也会使用 FQDN。

第二个原因可能是无法解析 Linux 计算机自身的主机名。

### 解决方案

- 1 在 Linux 计算机上，打开 Horizon Agent 日志文件。

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

- 2 查找表明无法解析 View 连接服务器实例的主机名的消息。

例如：

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

- 3 确保在 Linux 计算机上正确配置了 DNS 服务。

配置 DNS 的步骤因 Linux 分发版和版本而异。有关说明，请参考您的 Linux 分发版的文档。

要解决此问题，您可以手动将 View 连接服务器 FQDN 添加到 Linux 计算机上的 `/etc/hosts` 文件中。但是，建议您不要将此方法用于生产用途。正确的解决方案是配置 DNS 服务，以便使 Linux 计算机可以解析 View 连接服务器 FQDN。

完成这些步骤之后，对 View 连接服务器主机名执行 `ping` 操作以验证是否可以解析这些主机名。例如：

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

- 4 如果问题继续存在，请确保可以解析 Linux 计算机的系统主机名。

例如，在 CentOS 上，您可能需要执行以下步骤：

- a 打开终端窗口，运行 `hostname` 命令。

此时会显示计算机的主机名。

- b 打开 `etc/hosts` 文件并添加主机名。

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

- c 对主机名进行 `ping` 操作以验证是否可以解析此主机名。

```
ping <your hostname>
```

**注** 如果您通过手动编辑 `/etc/resolv.conf` 文件在 Linux 计算机上指定 DNS 服务器，则此设置在某些 Linux 分发版中可能会丢失。如果您在 `/etc/hosts` 文件中指定 DNS 服务器，此设置将会保留。有关配置 DNS 和主机名的完整说明，请参考您的 Linux 分发版和版本的文档。

## 对 Linux 计算机上的 Horizon Agent 没有响应问题进行故障排除

在 Linux 虚拟机上安装的 Horizon Agent 没有响应。

### 问题

无法访问 Horizon 7 for Linux 桌面，并且 Horizon Agent 没有响应。

### 原因

Horizon Agent 服务可能未运行，或 X 会话可能在 Linux 计算机上未处于活动状态。

### 解决方案

- 1 在 Linux 虚拟机上，打开终端窗口，然后停止并重新启动 Horizon Agent 服务。

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

- 2 确认 X 会话处于活动状态。

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

- 3 在虚拟机上安装 X11 Simple VNC 服务器，并确认 X 会话在 VNC 上处于活动状态。

## 对远程桌面和客户端主机之间的复制和粘贴问题进行故障排除

在远程桌面和客户端主机之间复制和粘贴所支持的最大 1 MB 数据时，需要超过 3 秒钟时间。如果复制和粘贴少量数据，则不会出现此问题。

### 问题

如果为 SLED 11 SP3/SP4 桌面配置 1 个 vCPU 和 1 GB 内存，则在远程桌面和本地客户端主机之间进行复制和粘贴操作所需的时间可能超过 3 秒钟。

### 原因

复制和粘贴操作发生延迟可能是由于 SLED 11 SP3/SP4 的操作系统 API 较旧所导致。

### 解决方案

- ◆ 为 SLED 11 SP3/SP4 配置两个 vCPU 和 2 GB 内存。

## 配置 Linux 防火墙以允许传入的 TCP 连接

要允许用户连接到其 Linux 桌面，这些桌面必须可以接受来自 Horizon Client 设备、安全服务器和 View 连接服务器的传入 TCP 连接。

在 Ubuntu 和 Kylin 分发版中，已默认配置 iptables 防火墙并设置了一条“ACCEPT”的输入策略。

在 RHEL 和 CentOS 分发版中，如果可能，Horizon Agent 安装程序脚本将配置 iptables 防火墙并设置一条“ACCEPT”的输入策略。

请确保 RHEL 或 CentOS 客户机操作系统上的 iptables 具有一条“ACCEPT”的输入策略，用于接受来自 Blast 端口 22443 的新连接。

启用 BSG 时，将通过安全服务器或 View 连接服务器上的 BSG 在 Horizon Client 设备与 Linux 桌面之间建立连接。未启用 BSG 时，将直接在 Horizon Client 设备与 Linux 桌面之间建立连接。