

# Horizon 7 for Linux デスクト ップのセットアップ

2019 年 7 月

VMware Horizon 7 7.9



vmware®

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>) でご確認いただけます。このドキュメントに関するご意見およびご感想は、[docfeedback@vmware.com](mailto:docfeedback@vmware.com) までお送りください。

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

ヴァイエムウェア株式会社  
105-0013 東京都港区浜松町 1-30-5  
浜松町スクエア 13F  
[www.vmware.com/jp](http://www.vmware.com/jp)

Copyright © 2016-2019 VMware, Inc. All rights reserved. [著作権および商標情報](#)。

# 目次

## Horizon 7 for Linux デスクトップのセットアップ 6

### 1 機能とシステムの要件 7

- Horizon Linux デスクトップの機能 7
- Horizon 7 for Linux デスクトップの構成手順の概要 13
- Horizon 7 for Linux のシステム要件 14
  - 2D または vSGA グラフィックスの仮想マシン設定 24
  - Linux デスクトップでのセッション共同作業の設定 27

### 2 デスクトップ デプロイのための Linux 仮想マシンの準備 29

- 仮想マシンを作成して、Linux をインストールする 29
- リモート デスクトップ デプロイ用の Linux マシンの準備 30
- Horizon Agent 用依存パッケージのインストール 32

### 3 Linux デスクトップの Active Directory 統合のセットアップ 34

- Linux と Active Directory の統合 34
  - OpenLDAP サーバ パススルー認証の使用 35
  - Microsoft Active Directory に対する SSSD LDAP 認証の設定 35
  - Windbind ドメイン参加ソリューションの使用 35
  - PBISO (PowerBroker Identity Services Open) 認証の設定 36
  - Ubuntu デスクトップでの Samba オフライン ドメイン参加の設定 37
- シングル サインオンの設定 39
- スマート カード リダイレクトの設定 40
  - スマート カード リダイレクトでの RHEL デスクトップと Active Directory の統合 41
  - スマート カード リダイレクトでの Ubuntu デスクトップと Active Directory の統合 44
  - スマート カード リダイレクトでの SLED/SLES デスクトップと Active Directory の統合 48
  - RHEL デスクトップのスマート カード リダイレクトの設定 50
  - Ubuntu デスクトップのスマート カード リダイレクトの設定 54
  - SLED/SLES デスクトップのスマート カード リダイレクトの設定 60
- Linux デスクトップでの True SSO のセットアップ 63
  - True SSO での RHEL/CentOS デスクトップと Active Directory の統合 64
  - True SSO での Ubuntu デスクトップと Active Directory の統合 65
  - True SSO での SLED/SLES デスクトップと Active Directory の統合 69
  - RHEL/CentOS デスクトップでの True SSO の設定 70
  - Ubuntu デスクトップでの True SSO の設定 72
  - SLED/SLES デスクトップでの True SSO の設定 74

### 4 Linux デスクトップのグラフィックスのセットアップ 77

サポート対象の RHEL/SLED ディストリビューションを vGPU 用に構成する	77
NVIDIA GRID vGPU グラフィック カードの VIB の ESXi ホストへのインストール	78
Linux 仮想マシンで vGPU を使用するための共有 PCI デバイスの構成	79
NVIDIA GRID vGPU ディスプレイ ドライバのインストール	81
NVIDIA ディスプレイ ドライバがインストールされているかどうかの確認	82
vDGA を使用するための RHEL 6 の構成	83
ホストで NVIDIA GRID を使用するために DirectPath I/O を有効にする	83
vDGA バススルー デバイスの RHEL 6 仮想マシンへの追加	84
vDGA 用の NVIDIA ディスプレイ ドライバのインストール	85
NVIDIA ディスプレイ ドライバがインストールされているかどうかの確認	86
vSGA を使用するための RHEL 7.x の構成	87
vSGA 用の NVIDIA グラフィック カードの VIB の ESXi ホストへのインストール	87
Linux 仮想マシンでの vSGA の 3D 機能の構成	88
vSGA が Linux 仮想マシンで実行されているかどうかの確認	89

## 5 Horizon Agent のインストール 91

Linux 仮想マシンへの Horizon Agent のインストール	91
install_viewagent.sh コマンドライン オプション	92
Linux Agent 用証明書の構成	94
Linux 仮想マシンでの Horizon Agent のアップグレード	95
Linux 仮想マシンでの Horizon Agent のアップグレード	96
Horizon 7 for Linux マシンをアンインストール	97

## 6 Linux デスクトップの構成オプション 98

Linux デスクトップでの構成ファイルのオプション設定	98
スマート ポリシー の使用	107
スマート ポリシー の要件	107
User Environment Manager のインストール	107
User Environment Manager の構成	108
Horizon スマート ポリシー設定	108
Horizon スマート ポリシー定義への条件の追加	109
User Environment Manager の Horizon スマート ポリシーの作成	109
Linux デスクトップの Blast 設定の例	111
Linux デスクトップのクライアント ドライブ リダイレクト オプションの例	111
Linux デスクトップの vSphere コンソールへの表示を抑制する	112

## 7 Linux デスクトップ プールの作成と管理 113

Linux 版手動デスクトップ プールの作成	113
Linux デスクトップ プールの管理	114
Linux の自動化された完全なクローン デスクトップ プールの作成	116
Linux のインスタント クローン フローティング デスクトップ プールの作成	118

[ブローカ PowerCLI コマンド](#) 121

## 8 手動デスクトップ プールのための Horizon 7 の一括デプロイ 124

[Linux デスクトップの一括デプロイの概要](#) 124

[Linux デスクトップの一括アップグレードの概要](#) 126

[Linux デスクトップ マシンのクローンを作成するために仮想マシン テンプレートを作成する](#) 127

[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#) 129

[Linux 仮想マシンのクローンを作成するサンプル スクリプト](#) 130

[クローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト](#) 134

[SSH を使用してクローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト](#) 137

[Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト](#) 141

[SSH を使用して Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト](#) 144

[Linux デスクトップ マシンで Horizon Agent をアップグレードするサンプル PowerCLI スクリプト](#) 148

[SSH を使用して Linux 仮想マシンで Horizon Agent をアップグレードするサンプル スクリプト](#) 153

[Linux 仮想マシンで操作を実行するサンプル スクリプト](#) 158

## 9 Linux デスクトップのトラブルシューティング 163

[Horizon Console での Horizon Help Desk Tool の使用](#) 163

[Horizon Console で Horizon Help Desk Tool を開始します。](#) 164

[Horizon Help Desk Tool でのユーザーのトラブルシューティング](#) 164

[Horizon Help Desk Tool のセッションの詳細](#) 166

[Horizon Help Desk Tool のセッション プロセス](#) 169

[Horizon Help Desk Tool での Linux デスクトップ セッションのトラブルシューティング](#) 170

[Horizon 7 for Linux マシンの診断情報の収集](#) 171

[リモート デスクトップとクライアント ホストの間でのコピーと貼り付けに関するトラブルシューティング](#) 172

[Horizon Agent が iPad Pro Horizon Client で切断できない](#) 172

[SLES 12 SP1 デスクトップが自動更新されない](#) 172

[SSO がパワーオフ エージェントに接続できない](#) 173

[Linux 版手動デスクトップ プール作成後の接続不能な仮想マシン](#) 173

# Horizon 7 for Linux デスクトップのセットアップ

『Horizon 7 for Linux デスクトップのセットアップ』には、Linux 仮想マシンを VMware Horizon<sup>®</sup> 7 for Linux デスクトップとしてセットアップする方法が記載されています。たとえば、Linux ゲスト OS を準備する方法、仮想マシンに Horizon Agent をインストールする方法、Horizon 7 環境で使用するよう Horizon Administrator でマシンを構成する方法などが記載されています。

## 対象読者

この情報は、Linux ゲスト OS で実行するリモート デスクトップを構成および使用するすべてのユーザーを対象にしています。本書に記載されている内容は、仮想マシン テクノロジおよびデータセンターの運用に精通している経験豊富な Linux システム管理者向けに書かれています。

# 機能とシステムの要件

Horizon 6.2.x 以降では、ユーザーは Linux オペレーティング システムを実行しているリモート デスクトップに接続できます。

この章には、次のトピックが含まれています。

- [Horizon Linux デスクトップの機能](#)
- [Horizon 7 for Linux デスクトップの構成手順の概要](#)
- [Horizon 7 for Linux のシステム要件](#)

## Horizon Linux デスクトップの機能

次のリストでは、Horizon Linux デスクトップでサポートされる主な機能を示します。

### Linux デスクトップでサポートされる機能

#### Active Directory の統合

次の Linux ディストリビューションを実行しているインスタント クローン デスクトップでは、PBISO (PowerBroker Identity Services Open) を使用して Active Directory とのオフライン ドメイン参加を実行できます。

- Ubuntu 16.04 および 18.04
- SLED/SLES 11 および 12.x

詳細については、[Linux と Active Directory の統合](#)の「PBISO (PowerBroker Identity Services Open) 認証」セクションを参照してください。

次の Linux ディストリビューションを実行しているインスタント クローン デスクトップでは、Samba を使用して Active Directory とのオフライン ドメイン参加を実行できます。

- Ubuntu 16.04 および 18.04
- RHEL 6.9 および 7.3
- CentOS 6.9/7.3
- SLED 11 SP4/12

#### オーディオ入力

クライアント ホストからリモート Linux デスクトップへのオーディオ入力リダイレクトがサポートされます。この機能は、USB リダイレクト機能をベースにしています。この機能を有効にするには、インストール時にこの機能を選択する必要があります。

あります。オーディオ アプリケーションの「PulseAudio サーバ (ローカル)」デバイスで、システムのデフォルト オーディオを選択する必要があります。この機能は、次の Linux ディストリビューションでサポートされます。

- MATE または Gnome Flashback (Metacity) デスクトップ環境の Ubuntu 16.04 x64
- MATE または Gnome Ubuntu デスクトップ環境の Ubuntu 18.04 x64
- KDE または Gnome デスクトップ環境の RHEL 7 Workstation x64
- SLED 11 SP4 x64
- SLED/SLES 12 SP3 x64

#### オーディオ出力

オーディオ出力リダイレクトがサポートされます。この機能は、デフォルトで有効になっています。この機能を無効にするには、`RemoteDisplay.allowAudio` オプションを **false** に設定する必要があります。Chrome または Firefox ブラウザを使用している場合、VMware Horizon HTML Access により、Linux デスクトップにオーディオ出力サポートが提供されます。

#### 自動化される完全なクローン デスクトップ プール

Linux デスクトップ用に、自動化される完全なクローン デスクトップ プールを作成できます。

#### クライアント ドライブ リダイレクト

クライアント ドライブ リダイレクト (CDR) 機能を有効にすると、ローカル システムの共有フォルダとドライブにアクセスできます。リモート Linux デスクトップのユーザーのホーム ディレクトリにある `tsclient` フォルダを使用します。この機能を使用するには、CDR コンポーネントをインストールする必要があります。

#### クリップボード リダイレクト

クリップボード リダイレクト機能を使用すると、リッチ テキストまたはプレーン テキストをクライアント ホストとリモートの Linux デスクトップ間でコピー アンド ペーストできます。Horizon Agent のオプションを使用して、コピー/ペーストの方向と最大テキスト サイズを設定できます。この機能は、デフォルトで有効になっています。インストール時にこの機能を無効にできます。

#### FIPS 140-2 モード

FIPS (Federal Information Processing Standard) 140-2 モード サポートは、NIST 暗号モジュール認証制度 (CMVP) で検証されていませんが、Linux デスクトップで使用できるようになりました。

Horizon 7 Agent for Linux は、FIPS 140-2 準拠の暗号モジュールを実装します。これらのモジュールは、CMVP 証明書 #2839 および #2866 に記載されている動作環境で検証され、このプラットフォームに移植されました。ただし、VMware の NIST CAVP および CMVP 証明書に新しい動作環境を追加するための CAVP および CMVP テスト要件は、プロダクト ロードマップに従って完了します。

---

**注:** FIPS 140-2 モードを使用するには、TLS (Transport Layer Security) プロトコル バージョン 1.2 が必要です。

---



**ヘルプ デスク ツール**

Horizon Help Desk Tool は、Linux デスクトップ セッションのトラブルシューティングに使用できる Web アプリケーションです。Horizon Help Desk Tool を使用して Horizon 7 ユーザー セッションのステータスを取得したり、トラブルシューティングやメンテナンス操作を実行できます。 [Horizon Console での Horizon Help Desk Tool の使用](#)を参照してください。

**Horizon スマート ポリシー**

VMware User Environment Manager™ 9.4 以降を使用して Horizon スマート ポリシーを作成すると、特定のリモート Linux デスクトップの USB リダイレクト、クリップボードのリダイレクト、クライアント ドライブ リダイレクト機能の動作を制御できます。 [スマート ポリシー の使用](#)を参照してください。

**H.264 エンコーダ**

H.264 は、特に低いバンド幅ネットワークでは、Horizon デスクトップの Blast Extreme のパフォーマンスを改善できます。クライアント システムで H.264 を無効にすると、Blast Extreme は自動的に JPEG/PNG のエンコーディングに戻ります。

H.264 エンコーダには、ハードウェア H.264 のサポートとソフトウェア エンコーダのサポートの両方が含まれます。ハードウェア H.264 のサポートには次の要件があります。

- NVIDIA グラフィック カードで vGPU が構成されていること。
- NVIDIA ドライバ 384 シリーズ以降が NVIDIA グラフィック カードにインストールされていること。

システムが前述の要件を満たしている場合、Horizon 7 for Linux はハードウェア H.264 エンコーダを使用します。それ以外の場合には、ソフトウェア H.264 エンコーダが使用されます。

**インスタント クローン フローティング デスクトップ プール**

Linux デスクトップ用に、インスタント クローン フローティング デスクトップ プールを作成できます。この機能は、次の Linux ディストリビューションがインストールされているシステムでのみサポートされます。

- Ubuntu 16.04 および 18.04
- RHEL 7.1 以降
- SLED/SLES 11 および 12

詳細については、 [Linux のインスタント クローン フローティング デスクトップ プールの作成](#)を参照してください。

**K デスクトップ環境**

次の Linux ディストリビューションで K デスクトップ環境 (KDE) がサポートされています。

- CentOS 6 および 7
- RHEL 6 および 7
- Ubuntu 16.04 および 18.04
- SLED 11 SP4

**キーボード レイアウトおよび言語の同期**

この機能は、クライアントのシステム言語と現在のキーボード レイアウトを Horizon Linux エージェント デスクトップと同期させるかどうかを指定します。この設定を有効にする、あるいは構成しない場合、同期が許可されます。この設定を無効にすると、同期が許可されません。

この機能は、VMware Horizon for Windows のみでサポートされ、英語、フランス語、ドイツ語、日本語、韓国語、スペイン語、簡体字中国語、および繁体字中国語の言語でのみサポートされます。

**可逆圧縮 PNG**

デスクトップで生成される画像とビデオは、クライアント デバイスで正確なピクセル レベルで表示されます。

**手動デスクトップ プール**

マシン ソース

- 管理対象仮想マシン - vCenter Server 仮想マシンのマシン ソース。管理対象仮想マシンは、新規およびアップグレードの展開にサポートされます。
- 管理対象外の仮想マシン - 他のソースのマシン ソース。管理対象外の仮想マシンは、管理対象外の仮想マシンの展開からアップグレードする場合にのみ、サポートされます。

---

**注:** パフォーマンスを維持するため、管理対象外の仮想マシンは使用しないでください。

---

**MATE デスクトップ環境**

次の Linux ディストリビューションで MATE デスクトップ環境がサポートされています。

- Ubuntu 16.04
- Ubuntu 18.04

**マルチモニタ**

- vDGA/vGPU デスクトップは、最大 2560x1600 の解像度を 4 台のモニターでサポートします。
- VMware vSphere<sup>®</sup> 6.0 以降の 2D/vSGA デスクトップは、最大 2048x1536 の解像度を 4 台のモニターでサポートし、最大 2560x1600 の解像度を 3 台のモニターでサポートします。

Ubuntu 16.04 および 18.04 でマルチモニタ機能を使用するには、Gnome、KDE または MATE デスクトップ環境を使用する必要があります。詳細については、<http://kb.vmware.com/kb/2151294> を参照してください。

SLES 12 SP1 では、カーネル レベル kernel-default-3.12.49-11.1 のデフォルト パッケージを使用する必要があります。パッケージをアップグレードしている場合、マルチ モニター機能は動作せず、デスクトップは 1 台のモニターに表示されます。

VMware Horizon HTML Access<sup>™</sup> バージョン 5.0 以降では、Horizon 7 for Linux デスクトップでマルチモニタ機能がサポートされます。

## VMware Blast のネットワーク インテリジェンス サポート

VMware Blast のネットワーク インテリジェンス トランスポートがサポートされます。この機能は、デフォルトで有効になっています。

UDP (ユーザー データグラム プロトコル) を有効にすると、Blast は、TCP (伝送制御プロトコル) と UDP の両方の接続を確立します。Blast は、現在のネットワーク条件に基づいて、データ転送を動的に選択し、最高のユーザー エクスペリエンスを実現します。たとえば、ローカル エリア ネットワークでは UDP よりも TCP のほうが適しているため、Blast はデータ転送に TCP を選択します。また、ワイド エリア ネットワーク (WAN) では、UDP のほうが TCP よりもパフォーマンスが良いため、Blast は UDP 転送を選択します。

使用するインライン コンポーネントのいずれかで UDP がサポートされていない場合、Blast は TCP 接続のみを確立します。たとえば、Horizon Connection Server またはセキュリティ サーバの Blast Security Gateway コンポーネントを使用している場合、TCP 接続のみが確立されます。クライアントとエージェントの両方で UDP が有効な場合でも、Blast Security Gateway が UDP をサポートしていないため、接続では TCP が使用されます。ユーザーが会社のネットワークの外部から接続している場合、UDP コンポーネントは、UDP をサポートする VMware Unified Access Gateway (旧称 Access Point) を必要とします。

UDP ベースの Blast 接続を確立するには、次の情報を使用します。

- クライアントが Linux デスクトップに直接接続している場合には、クライアントとエージェントの両方で UDP を有効にします。デフォルトでは、クライアントとエージェントの両方で UDP が有効になっています。
- クライアントが Unified Access Gateway を介して Linux デスクトップに接続している場合には、クライアント、エージェント、Unified Access Gateway で UDP を有効にします。

## セッション共同作業

セッション共同作業機能により、ユーザーは既存のリモート Linux デスクトップ セッションに参加するユーザーを招待できます。また、別のユーザーから招待を受信したときに、共同作業セッションに参加できます。この機能は、次の Linux ディストリビューションがインストールされているリモート Linux デスクトップでのみサポートされます。

- Gnome デスクトップ環境の Ubuntu 18.04
- Gnome デスクトップ環境の RHEL 7.5 以降

## シングル サインオン

シングル サインオン (SSO) は、次の Linux ディストリビューションでサポートされます。

- RHEL 7/6 Workstation x64
- CentOS 7/6 x64
- SLED/SLES 12.x SP3/SP2/SP1
- SLED 11.x SP4/SP3 x64

	<ul style="list-style-type: none"> <li>■ Ubuntu 18.04/16.04 x64</li> </ul>
SSO でスマート カード リダイレクト	<p>スマート カード リダイレクトは、次の Linux ディストリビューションでサポートされます。</p> <ul style="list-style-type: none"> <li>■ RHEL 7.1 以降</li> <li>■ RHEL 6.6 以降</li> <li>■ Ubuntu 18.04/16.04</li> <li>■ SLED/SLES 12.x SP3</li> <li>■ SLED 11.x SP4</li> </ul> <p>この機能は、PIV (Personal Identity Verification) カードと CAC (Common Access Card) をサポートします。詳細については、<a href="#">スマート カード リダイレクトの設定</a>を参照してください。</p>
True SSO のサポート	<p>True SSO は、次の Linux ディストリビューションでサポートされます。</p> <ul style="list-style-type: none"> <li>■ RHEL 7</li> <li>■ CentOS 7</li> <li>■ SLED/SLES 12.x SP3</li> <li>■ Ubuntu 18.04/16.04</li> </ul> <p>詳細については、<a href="#">Linux デスクトップでの True SSO のセットアップ</a>を参照してください。</p>
USB リダイレクト	<p>USB リダイレクト機能により、リモート Linux デスクトップからローカルに接続された USB デバイスにアクセスできます。USB 機能を使用するには、USB リダイレクト コンポーネントと USB VHCI ドライバカーネル モジュールをインストールする必要があります。リダイレクトする USB デバイスを使用できる十分な権限がユーザーに付与されていることを確認します。</p>
3Dconnexion マウス	<p>3Dconnexion マウスを使用するには、適切なデバイス ドライバをインストールし、Linux デスクトップで USB デバイスの接続メニューを使用してマウスをペアリングする必要があります。</p>
3D グラフィックス	<p>3D グラフィックス機能は、次の Linux バージョンとグラフィック カードの組み合わせをサポートします。</p> <ul style="list-style-type: none"> <li>■ vSGA は、RHEL 7.x Workstation x64 と NVIDIA GRID K1 または K2 のグラフィック カードの組み合わせでサポートされます。</li> <li>■ vDGA は、RHEL 6.x Workstation x64 と NVIDIA GRID K1 または K2 のグラフィック カードの組み合わせでサポートされます。</li> <li>■ vGPU は、NVIDIA Tesla M6、M10、M60 または P40 グラフィック カードを使用する RHEL 6.x、RHEL 7.x、SLED 12.x Workstation x64 でサポートされます。</li> </ul>

## Linux デスクトップとデスクトップ プールの制限

Linux デスクトップとデスクトップ プールには次の制限があります。

- 仮想印刷、ロケーション ベースの印刷、リアルタイム ビデオはサポートされません。
- VMware HTML Access ファイル転送はサポートされていません。

---

**注:** セキュリティ サーバが利用される場合、社内のファイアウォールでポート 22443 を開き、セキュリティ サーバと Linux デスクトップ間のトラフィックを許可する必要があります。

---

## Horizon 7 for Linux デスクトップの構成手順の概要

Horizon 7 for Linux デスクトップをインストールして構成する場合、仮想マシンに 2D グラフィックスまたは 3D グラフィックスをインストールするかどうかによって、実行する必要がある一連の手順が異なります。

### 2D グラフィックス - 構成手順の概要

2D グラフィックスの場合、次の手順を実行します。

- 1 Horizon 7 for Linux のデプロイ環境をセットアップするためのシステム要件を確認します。[Horizon 7 for Linux のシステム要件](#)を参照してください。
- 2 vSphere で仮想マシンを作成し、Linux オペレーティング システムをインストールします。[仮想マシンを作成して、Linux をインストールする](#)を参照してください。
- 3 Horizon 7 環境でデスクトップとしてデプロイするゲスト OS を準備します。[リモート デスクトップ デプロイ用の Linux マシンの準備](#)を参照してください。
- 4 Active Directory で認証するように Linux ゲスト OS を構成します。この手順は、環境内の要件に応じてサードパーティ製ソフトウェアで実装されます。詳細については、[Linux と Active Directory の統合](#)を参照してください。
- 5 Linux 仮想マシンに Horizon Agent をインストールします。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- 6 構成した Linux 仮想マシンを含むデスクトップ プールを作成します。[Linux 版手動デスクトップ プールの作成](#)を参照してください。

### 3D グラフィックス - 構成手順の概要

マシンに Horizon Agent をインストールし、Horizon Administrator にデスクトップ プールをデプロイする前に、NVIDIA GRID vGPU、vDGA または vSGA の設定を完了します。

- 1 Horizon 7 for Linux のデプロイ環境をセットアップするためのシステム要件を確認します。[Horizon 7 for Linux のシステム要件](#)を参照してください。
- 2 vSphere で仮想マシンを作成し、Linux オペレーティング システムをインストールします。[仮想マシンを作成して、Linux をインストールする](#)を参照してください。
- 3 Horizon 7 環境でデスクトップとしてデプロイするゲスト OS を準備します。[リモート デスクトップ デプロイ用の Linux マシンの準備](#)を参照してください。

- 4 Active Directory で認証するように Linux ゲスト OS を構成します。この手順は、環境内の要件に応じてサードパーティ製ソフトウェアで実装されます。詳細については、[Linux と Active Directory の統合](#)を参照してください。
- 5 ESXi ホストと Linux 仮想マシンで 3D 機能を構成します。インストールする 3D 機能に関する手順を実行します。
  - [サポート対象の RHEL/SLED ディストリビューションを vGPU 用に構成する](#)を参照してください。
  - [vDGA を使用するための RHEL 6 の構成](#)を参照してください。
  - [vSGA を使用するための RHEL 7.x の構成](#)を参照してください。
- 6 Linux 仮想マシンに Horizon Agent をインストールします。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- 7 構成した Linux 仮想マシンを含むデスクトップ プールを作成します。[Linux 版手動デスクトップ プールの作成](#)を参照してください。

## 一括デプロイ

Horizon Administrator では、手動デスクトップ プールへの Linux 仮想マシンのデプロイのみを行うことができます。vSphere PowerCLI を使用すると、Linux デスクトップ マシンのプールのデプロイを自動化するスクリプトを開発できます。[8 章 手動デスクトップ プールのための Horizon 7 の一括デプロイ](#)を参照してください。

## Horizon 7 for Linux のシステム要件

Horizon 7 for Linux をインストールするには、Linux システムがオペレーティング システム、Horizon 7 および vSphere プラットフォームの特定の要件を満たしている必要があります。

## Horizon Agent でサポートされる Linux バージョン

[表 1-1. Horizon Agent でサポートされる Linux オペレーティング システム](#)に、デスクトップ プール内の仮想マシンでサポートされる Linux オペレーティング システムの一覧を示します。

表 1-1. Horizon Agent でサポートされる Linux オペレーティング システム

Linux ディストリビューション	アーキテクチャ
Ubuntu 16.04 および 18.04	x64
注: VMware ナレッジベースの記事 <a href="http://kb.vmware.com/kb/2151294">http://kb.vmware.com/kb/2151294</a> で説明されている解決策のいずれかを行う必要があります。	
RHEL 6.6、6.7、6.8、6.9、6.10、7.2、7.3、7.4、7.5 および 7.6	x64
CentOS 6.6、6.7、6.8、6.9、6.10、7.2、7.3、7.4、7.5 および 7.6	x64
Neokylin 6 Update 1	x64
SLED 11.x SP3/SP4	x64

Linux ディストリビューション	アーキテクチャ
SLED 12.x SP1/SP2/SP3	x64
SLES 12.x SP1/SP2/SP3	x64

**注:** Linux エージェントは、一部の Linux ディストリビューションで依存パッケージを使用します。詳細については、[Horizon Agent 用依存パッケージのインストール](#)を参照してください。

## 必須のプラットフォームと Horizon 7 ソフトウェア バージョン

Horizon 7 for Linux をインストールして使用するには、環境が vSphere プラットフォーム、Horizon 7、Horizon Client ソフトウェアの特定の要件を満たしている必要があります。

表 1-2. 必須のプラットフォームと Horizon 7 ソフトウェア バージョン

プラットフォームとソフトウェア	サポートされているバージョン
vSphere プラットフォームのバージョン	<ul style="list-style-type: none"> <li>■ vSphere 6.0 U2 以降のリリース</li> <li>■ vSphere 6.5 U1 以降のリリース</li> <li>■ vSphere 6.7 以降のリリース</li> </ul>
Horizon 環境	<ul style="list-style-type: none"> <li>■ Horizon Connection Server 7.9</li> </ul>
Horizon Client ソフトウェア	<ul style="list-style-type: none"> <li>■ Horizon Client 5.1.0 for Android</li> <li>■ Horizon Client 5.1.0 for Windows</li> <li>■ Horizon Client 5.1.0 for Linux</li> <li>■ Horizon Client 5.1.0 for Mac OS X</li> <li>■ Horizon Client 5.1.0 for iOS (iPad Pro)</li> <li>■ Chrome、Firefox、Internet Explorer での HTML Access 5.1.0</li> <li>■ ゼロ クライアントはサポートされません。</li> </ul>

## Linux 仮想マシンにより使用される TCP/UDP ポート

Horizon Agent と Horizon Client は、互いのネットワーク アクセスや各種 Horizon サーバ コンポーネント間のネットワーク アクセスに TCP または UDP ポートを使用します。

表 1-3. Linux 仮想マシンにより使用される TCP/UDP ポート

送信元	ポート	送信先	ポート	プロトコル	説明
Horizon Client	*	Linux Agent	22443	TCP/UDP	Blast Security Gateway が使用されない場合は Blast
セキュリティ サーバ、Horizon Connection Server、または Access Point アプライアンス	*	Linux Agent	22443	TCP/UDP	Blast Security Gateway が使用される場合は Blast
Horizon Agent	*	Horizon Connection Server	4001、4002	TCP	JMS SSL トラフィック。

**注:** クライアントにより使用される TCP および UDP ポートの詳細については、『Horizon Client および Horizon Agent のセキュリティ』ドキュメントで「クライアントおよび Horizon Agent により使用される TCP および UDP ポート」を参照してください。

ユーザーが自分の Linux デスクトップに接続できるようにするには、Horizon Client デバイス、セキュリティ サーバ、および Horizon Connection Server から受信する TCP 接続をデスクトップが受け入れることができる必要があります。

Ubuntu および Kylin ディストリビューションでは、`iptables` ファイアウォールがデフォルトで構成されており、入力ポリシーが ACCEPT に設定されています。

RHEL および CentOS ディストリビューションでは、可能な場合、Horizon Agent インストーラ スクリプトが、入力ポリシーを ACCEPT にして `iptables` ファイアウォールを構成します。

RHEL や CentOS ゲスト OS の `iptables` では、Blast ポート 22443 からの新しい接続について入力ポリシーが ACCEPT になっていることを確認します。

BSG が有効な場合、クライアント接続はセキュリティ サーバまたは Horizon Connection Server の BSG を介して Horizon Client デバイスから Linux デスクトップに送られます。BSG が有効ではない場合、Horizon Client デバイスは Linux デスクトップに直接接続されます。

## Linux 仮想マシンにより使用される Linux アカウントの確認

表 1-4. アカウント名およびアカウント タイプに、Linux 仮想マシンで使用されるアカウント名とアカウント タイプを示します。

表 1-4. アカウント名およびアカウント タイプ

アカウント名	アカウントタイプ	使用
ルート	Linux OS に組み込み	Java スタンドアローン エージェント、mksvchanserver、シェルスクリプト
vmwblast	Linux Agent インストーラが作成	VMwareBlastServer
<現在のログイン ユーザー>	Linux OS に組み込み、Active Directory ユーザー、または LDAP ユーザー	Python スクリプト



## デスクトップ環境

Horizon 7 for Linux は、異なる Linux ディストリビューションで複数のデスクトップ環境をサポートします。表 1-5. サポート対象のデスクトップ環境に、各 Linux ディストリビューションのデフォルトのデスクトップ環境と Horizon 7 for Linux でサポートされる追加のデスクトップ環境の一覧を示します。

表 1-5. サポート対象のデスクトップ環境

Linux ディストリビューション	デフォルトのデスクトップ環境	Horizon 7 for Linux デスクトップでサポートされるデスクトップ環境
Ubuntu 18.04	Gnome	Gnome Ubuntu、K デスクトップ環境 (KDE)、MATE
Ubuntu 16.04	Unity	Gnome Flashback (Metacity)、KDE、MATE
RHEL/CentOS 6.x	Gnome	Gnome、KDE
RHEL/CentOS 7.x	Gnome	Gnome、KDE
SLED 11 SP4	Gnome	Gnome、KDE
SLED 12 SP1/SP2/SP3	Gnome	Gnome
SLES 12 SP1/SP2/SP3	Gnome	Gnome
NeoKylin 6 Update 1	Mate	Mate

サポートされている Linux ディストリビューションのいずれかで使用するデフォルト デスクトップ環境を変更するには、次の手順に従って、Linux デスクトップに適切なコマンドを使用する必要があります。

**注:** KDE と MATE デスクトップ環境のシングル サインオン (SSO) は、Linux デスクトップがデフォルトのログイン画面を使用している場合にのみ機能します。表 1-6. デスクトップ環境のインストール コマンドにあるコマンドを使用して、KDE と MATE をインストールする必要があります。

SLED 11 SP3/SP4、RHEL/CentOS 7、Ubuntu 18.04/16.04 ディストリビューションを使用している場合、ロックされた KDE セッションを SSO でロック解除することはできません。パスワードを入力して、手動でロックされているセッションを手動でロック解除する必要があります。

- 1 デフォルトのデスクトップ環境の設定を使用して、サポートされている Linux ディストリビューションのオペレーティングシステムをインストールします。
- 2 ご使用の Linux ディストリビューションに適切なコマンドを実行します。実行するコマンドについては、表 1-6. デスクトップ環境のインストール コマンドを参照してください。

表 1-6. デスクトップ環境のインストール コマンド

Linux ディストリビューション	新しいデフォルトのデスクトップ環境	デフォルトのデスクトップ環境を変更するコマンド
RHEL/CentOS 6	KDE	# yum groupinstall "X Window System" "KDE Desktop"
RHEL/CentOS 7	KDE	# yum groupinstall "KDE Plasma Workspaces"
SLED 11 SP4	KDE	# zypper install -t pattern desktop-kde

Linux ディストリビューション	新しいデフォルトのデスクトップ環境	デフォルトのデスクトップ環境を変更するコマンド
Ubuntu 18.04/16.04	KDE	<code># apt install plasma-desktop</code>
Ubuntu 18.04	MATE 1.225	<code># apt install ubuntu-mate-desktop</code>
Ubuntu 16.04	MATE 1.16	<pre># apt-add-repository ppa:ubuntu-mate-dev/xenial-mate # apt update # apt upgrade # apt install mate # apt install ubuntu-mate-themes</pre>
Ubuntu 16.04	Gnome Flashback (Metacity)	<code># apt install gnome-session-flashback</code>

3 新しいデフォルトのデスクトップ環境を開始するには、デスクトップを再起動します。

複数のデスクトップ環境がインストールされている Linux デスクトップで SSO を有効にした場合は、次の情報を使用して、SSO セッションで使用するデスクトップ環境を選択します。

- Ubuntu 18.04/16.04、RHEL/CentOS 7.x の場合には、[表 1-7. SSODesktopType オプション](#)の情報をを使用して、`/etc/vmware/viewagent-custom.conf` ファイルに `SSODesktopType` オプションを設定し、SSO で使用するデスクトップ環境を指定します。

表 1-7. SSODesktopType オプション

デスクトップタイプ	SSODesktopType オプションの設定
MATE	<code>SSODesktopType=UseMATE</code>
GnomeUbuntu	<code>SSODesktopType=UseGnomeUbuntu</code>
GnomeFlashback	<code>SSODesktopType=UseGnomeFlashback</code>
KDE	<code>SSODesktopType=UseKdePlasma</code>
GnomeClassic	<code>SSODesktopType=UseGnomeClassic</code>

- RHEL/CentOS 6.x と SLED 11 SP3/SP4 で SSO ログインセッションに KDE を使用する場合には、KDE スタートアップ ファイルを除く、すべてのデスクトップスタートアップ ファイルを `/usr/share/xsession` ディレクトリから削除します。たとえば、次のコマンドセットを使用します。

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

次の SSO セッションでデフォルトのデスクトップとして KDE を使用するには、初期セットアップの後、エンドユーザーは Linux デスクトップからログアウトするか、システムを再起動する必要があります。

複数のデスクトップ環境がインストールされている Linux デスクトップで SSO を無効にした場合には、前述の手順を行う必要はありません。エンドユーザーが Linux デスクトップにログインするときに、必要なデスクトップ環境を選択します。

## ネットワーク要件

VMware Blast Extreme は、UDP (ユーザー データグラム プロトコル) と TCP (伝送制御プロトコル) の両方をサポートします。ネットワーク条件は、UDP と TCP のパフォーマンスに影響を及ぼします。最高のユーザー エクスペリエンスを実現するには、ネットワーク条件に応じて UDP または TCP を選択します。

- ローカル エリア ネットワーク (LAN) 環境など、ネットワーク条件が良好な場合には TCP を選択します。
- パケット損失や遅延が発生するワイド エリア ネットワーク (WAN) 環境など、ネットワーク条件が良好でない場合には UDP を選択します。

Wireshark などのネットワーク アナライザ ツールを使用して、VMware Blast Extreme が TCP と UDP のどちらを使用するかを確認します。次の手順を行います。ここでは例として Wireshark を使用しています。

- 1 Linux 仮想マシンに Wireshark をダウンロードして、インストールします。

RHEL/CentOS 6 の場合 :

```
sudo yum install wireshark
```

Ubuntu 18.04/16.04 の場合 :

```
sudo apt install tshark
```

SLED/SLES 11/12 の場合:

```
sudo zypper install wireshark
```

- 2 VMware Horizon Client を使用して、Linux デスクトップに接続します。
- 3 ターミナル ウィンドウを開き、次のコマンドを実行します。VMware Blast Extreme が使用する TCP パッケージまたは UDP パッケージが表示されます。

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

USB リダイレクトとクライアント ドライブのリダイレクト (CDR) 機能はネットワーク条件に依存します。パケットロスや遅延でバンド幅に制限があるなど、ネットワーク条件が良好でないと、ユーザー エクスペリエンスが低下します。このような場合、次のいずれかが発生する可能性があります。

- リモート ファイルのコピーが低速になる。この場合、サイズの小さいファイルを送信します。
- USB デバイスがリモートの Linux デスクトップに表示されない。
- USB データの転送が不完全になります。たとえば、サイズの大きいファイルをコピーした場合、元のファイルよりもサイズが小さくなる可能性があります。

## USB リダイレクトのための VHCI ドライバ

USB リダイレクト機能は、USB VHCI (Virtual Host Controller Interface) カーネル ドライバに依存します。USB 3.0 を使用するには、VHCI ドライバにパッチを適用する必要があります。

Horizon for Linux インストーラには、サポートされる Linux ディストリビューションのデフォルト カーネル用の VHCI ドライバ バイナリが含まれています。USB リダイレクト機能を選択すると、VHCI ドライバがインストールされます。表 1-8. デフォルトのカーネル バージョンに、Horizon for Linux インストーラがインストールするデフォルトのカーネル バージョンが記載されています。

表 1-8. デフォルトのカーネル バージョン

Linux ディストリビューション	デフォルトのカーネル バージョン
RHEL/CentOS 6.9	2.6.32-696.el6.x86_64
RHEL/CentOS 6.10	2.6.32-754.el6.x86_64
RHEL/CentOS 7.5	3.10.0-862.el7.x86_64
SUSE 12 SP2	4.4.21-69-default
Ubuntu 16.04	4.4.0-21-generic
Ubuntu 18.04	4.15.0-20-generic

Linux システムで Horizon for Linux インストーラに含まれるデフォルト バージョンと異なるカーネル バージョンを使用している場合には、<https://sourceforge.net/projects/usb-vhci/files/linux%20kernel%20module/> から USB VHCI ソース コードをダウンロードする必要があります。VHCI ドライバのソース コードをコンパイルし、結果のバイナリを Linux システムにインストールする必要があります。

**注:** Horizon for Linux をインストールする前に、VHCI ドライバをインストールする必要があります。

USB VHCI ドライバのソース コードの最新バージョンをダウンロードした後、表 1-9. USB VHCI ドライバのコンパイルとインストールにあるコマンドを使用して、Linux システムをコンパイルしてインストールします。たとえば、インストール ファイル VMware-horizonagent-linux-x86\_64-*<version>*-*<build-number>*.tar.gz を /install\_tmp/ ディレクトリに展開する場合、*full-path-to-patch-file* は /install\_tmp/VMware-horizonagent-linux-x86\_64-*<version>*-*<buildnumber>*/resources/vhci/patch/vhci.patch になり、使用する patch コマンドは次のようになります。

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

表 1-9. USB VHCI ドライバのコンパイルとインストール

Linux ディストリ ビューション	USB VHCI ドライバのコンパイルとインストールの手順
Ubuntu 18.04	<div data-bbox="360 310 858 338">1 依存パッケージをインストールします。</div> <div data-bbox="413 373 751 449"># apt-get install make # apt-get install gcc # apt-get install libelf-dev</div> <div data-bbox="360 476 858 504">2 VHCI ドライバをコンパイルし、インストールします。</div> <div data-bbox="413 539 858 640"># tar -xzf vhci-hcd-1.15.tar.gz # cd vhci-hcd-1.15 # patch -p1 &lt; full-path_to_patch-file # make clean &amp;&amp; make &amp;&amp; make install</div>
Ubuntu 16.04	<div data-bbox="360 678 818 705">VHCI ドライバをコンパイルし、インストールします。</div> <div data-bbox="373 722 818 827"># tar -xzf vhci-hcd-1.15.tar.gz # cd vhci-hcd-1.15 # patch -p1 &lt; full-path_to_patch-file # make clean &amp;&amp; make &amp;&amp; make install</div>

Linux ディストリ ビューション	USB VHCI ドライバのコンパイルとインストールの手順
RHEL/CentOS 6.9/6.10 RHEL/CentOS 7	<ol style="list-style-type: none"> <li>1 依存パッケージをインストールします。 <div data-bbox="400 317 893 426"> <pre># yum install gcc-c++ # yum install kernel-devel-\$(uname -r) # yum install kernel-headers-\$(uname -r) # yum install patch</pre> </div> </li> <li>2 VHCI ドライバをコンパイルし、インストールします。 <div data-bbox="400 510 858 619"> <pre># tar -xzf vhci-hcd-1.15.tar.gz # cd vhci-hcd-1.15 # patch -p1 &lt; full-path_to_patch-file # make clean &amp;&amp; make &amp;&amp; make install</pre> </div> </li> </ol>
SLED/SLES 11 SP4 SLED/SLES 12 SP2	<ol style="list-style-type: none"> <li>1 現在のカーネル パッケージのバージョンを確認します。 <div data-bbox="400 709 1262 741"> <pre># rpm -qa   grep kernel-default-\$(echo \$(uname -r)   cut -d '-' -f 1,2)</pre> </div> <p>現在インストールされているカーネル パッケージの名前が出力されます。たとえば、パッケージ名が <code>kernel-default-3.0.101-63.1</code> の場合、現在のカーネル パッケージのバージョンは <code>3.0.101-63.1</code> になります。</p> </li> <li>2 依存パッケージをインストールします。 <ol style="list-style-type: none"> <li>a SUSE 11 の場合、現在のカーネルに一致する <code>kernel-source</code> パッケージと <code>kernel-default-devel</code> パッケージをインストールします。gcc パッケージと <code>patch</code> パッケージをインストールします。 <div data-bbox="456 982 1303 1035"> <pre># zypper install --oldpackage kernel-source-&lt;kernel-package-version&gt; \ kernel-default-devel-&lt;kernel-package-version&gt; gcc patch</pre> </div> <p>例 :</p> <div data-bbox="456 1129 1329 1182"> <pre># zypper install --oldpackage kernel-source-3.0.101-63.1 kernel-default- devel-3.0.101-63.1 gcc patch</pre> </div> </li> <li>b SUSE 12 の場合、<code>kernel-devel</code>、<code>kernel-default-devel</code>、<code>kernel-macros</code> および <code>patch</code> パッケージをインストールします。 <div data-bbox="400 1304 1326 1381"> <pre># zypper install --oldpackage kernel-devel-&lt;kernel-package-version&gt; \ kernel-default-devel-&lt;kernel-package-version&gt; kernel-macros-&lt;kernel-package- version&gt; patch</pre> </div> <p>例 :</p> <div data-bbox="400 1476 1254 1528"> <pre># zypper install --oldpackage kernel-devel-4.4.21-90.1 kernel-default- devel-4.4.21-90.1 kernel-macros-4.4.21-90.1 patch</pre> </div> </li> </ol> </li> <li>3 VHCI ドライバをコンパイルし、インストールします。 <div data-bbox="400 1612 1409 1801"> <pre># tar -xzf vhci-hcd-1.15.tar.gz # cd vhci-hcd-1.15 # patch -p1 &lt; full-path_to_patch-file # mkdir -p linux/\$(echo \$(uname -r)   cut -d '-' -f 1)/drivers/usb/core # cp /lib/modules/\$(uname -r)/source/include/linux/usb/hcd.h linux/\$(echo \$(uname - r)   cut -d '-' -f 1)/drivers/usb/core # make clean &amp;&amp; make &amp;&amp; make install</pre> </div> </li> </ol>

Linux カーネルのバージョンが変更された場合は、VHCI ドライバを再コンパイルして再インストールする必要がありますが、Horizon for Linux を再インストールする必要はありません。

また、Ubuntu 18.04/16.04 システムの場合、次の例に類似した手順で動的カーネル モジュール サポート (DKMS) を VHCI ドライバに追加できます。

- 1 カーネル ヘッダーをインストールします。

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

- 2 次のコマンドを使用して、dkms をインストールします。

```
# apt install dkms
```

- 3 VHCI TAR ファイルを展開し、パッチを適用します。

```
# tar xzvf vhci-hcd-1.15.tar.gz
# cd vhci-hcd-1.15
# patch -p1 <full-path-to-patch-file>
# cd ..
```

- 4 展開した VHCI ソース ファイルを /usr/src ディレクトリにコピーします。

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

- 5 dkms.conf という名前のファイルを作成し、/usr/src/usb-vhci-hcd-1.15 ディレクトリに配置します。

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

- 6 次の行を dkms.conf ファイルに追加します。

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

CLEAN="$MAKE_CMD_TMPL clean"

BUILT_MODULE_NAME[0]="usb-vhci-iocifc"
DEST_MODULE_LOCATION[0]="/kernel/drivers/usb/host"
MAKE[0]="$MAKE_CMD_TMPL"

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

AUTOINSTALL="YES"
```

- 7 この VHCI ドライバを dkms に追加します。

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

## 8 VHCI ドライバをビルドします。

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

## 9 VHCI ドライバをインストールします。

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

## 2D または vSGA グラフィックスの仮想マシン設定

特定の Horizon 7 for Linux の仮想マシンを作成する場合、メモリ設定と構成パラメータを推奨される最小値に変更する必要があります。

NVIDIA vDGA を使用するように構成された仮想マシンでは、NVIDIA の物理グラフィック カードを使用します。NVIDIA GRID vGPU を使用するように構成されている仮想マシンでは、NVIDIA の物理グラフィック アクセラレータをベースとする NVIDIA 仮想グラフィック カードを使用します。これらの仮想マシンのビデオ メモリ (vRAM) 設定および構成パラメータを変更する必要はありません。

2D または vSGA グラフィックを使用するように構成された仮想マシンでは、VMware 仮想グラフィック カードを使用します。また、これらのタイプの仮想マシンでは次の設定を変更する必要があります。

- ビデオ メモリ (vRAM) 設定
- 構成パラメータ
- 3D メモリ設定
- パフォーマンス要件を満たすための vCPU および仮想メモリ設定

### [ビデオ メモリ (vRAM) 設定]

vSphere Client で Linux 仮想マシンを作成するときには、[表 1-10. 2D または vSGA グラフィックスで推奨される vRAM 設定](#)に示されているように vRAM サイズを構成します。仮想マシンで構成するモニターの数と解像度別に推奨される vRAM サイズを設定します。

表 1-10. 2D または vSGA グラフィックスで推奨される vRAM 設定

VRAM サイズ	モニター数	最大解像度
10 MB	1	1600x1200 または 1680x1050
12 MB	1	1920x1440
32 MB	1	2560x1600
48 MB	2	2048x1536
80 MB	2	2560x1600
128 MB	3	2560x1600
128 MB	4	2048x1536

これらの vRAM サイズは推奨される最小値です。仮想マシンでさらにリソースを利用できる場合は、ビデオ パフォーマンスを向上するために vRAM にさらに大きな値に設定します。

最低解像度の 1 つのモニターで構成されているマシンの推奨 vRAM 最小サイズは 10 MB です。



仮想マシンを作成して、Linux をインストールするで説明されているように、ディスプレイの数と使用するビデオメモリの量を設定するには、仮想マシンをパワーオフする必要があります。

Horizon Connection Server 7 は、Windows 仮想マシンの場合と同じように、Linux 仮想マシンでは vRAM 設定を自動的に構成しません。vSphere Client で vRAM 設定を手動で構成する必要があります。

Linux 仮想マシンが推奨サイズよりも小さい vRAM で構成されている場合、次の問題が発生する可能性があります。

- 最初の接続の直後にデスクトップセッションが切断される。
- 自動調整が機能せず、画面の小さい領域にデスクトップが表示される。

Linux 仮想マシンの [ディスプレイの数] の値が実際に必要な数よりも少ない場合、1 台以上のモニターでデスクトップが空白で表示されます。

推奨された設定で自動調整の問題が発生する場合、さらに大きな vRAM サイズを指定することができます。vSphere Client では、最大で 128 MB の vRAM サイズが許可されます。128 MB を超過するサイズを指定した場合、vmx 構成ファイルを手動で変更する必要があります。次の例では、256 MB の vRAM サイズを指定しています。

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

## [構成パラメータ]

マルチモニタに Linux リモート デスクトップを表示するには、仮想マシンのいくつかの構成パラメータを設定する必要があります。仮想マシンの構成パラメータを設定する一般的な手順は、次のとおりです。

- 1 仮想マシンをパワーオフします。
- 2 vSphere Web Client で仮想マシンを右クリックして、[設定の編集] を選択します。
- 3 [仮想マシン オプション] タブをクリックし、[詳細] をクリックします。
- 4 [設定の編集] をクリックしてから、[行を追加] をクリックします。
- 5 構成パラメータの名前と値を入力します。
- 6 変更内容を保存するには、[OK] をクリックします。

次の構成パラメータを設定する必要があります。

- `svga.autodetect` を `false` に設定します。

```
svga.autodetect="false"
```

- ディスプレイ モニターの数と向き（横または縦）に応じて、`svga.maxWidth` および `svga.maxHeight` の値を計算します。原則では、すべてのディスプレイをサポートするために `svga.maxWidth` 値と `svga.maxHeight` 値を十分な大きさにする必要があります。たとえば、最大解像度 2560x1600 で 4 台のディスプレイをサポートするには、次の値を設定する必要があります。

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

マルチモニタを使用する場合は、これらの構成パラメータを設定する必要があります。これらのパラメータを設定しないと、次の問題が 1 つ以上発生する可能性があります。

- デスクトップはいくつかのモニターに表示され、他のモニターには何も表示されません。
- キーストロークが何回も表示される。
- デスクトップが遅くなる。
- 画面の小さい領域にデスクトップが表示される。

## 画面のサイズが 8192 x 8192 に制限される

RHEL 6.8/6.9/6.10/7.3/7.4, CentOS 6.8/6.9/6.10/7.3/7.4, Ubuntu 16.04, SLED 12 SP2/SP3, SLES 12 SP2/SP3 では、2D および vSGA の最大画面サイズは 4096 x 4096 になります。xrandr コマンドを実行すると、出力の最初の行に maximum 4096x4096 が表示されます。

Ubuntu 18.04, RHEL 7.5, CentOS 7.5 では、2D および vSGA の最大画面サイズは 8192 x 8192 になります。

マルチモニタを接続すると、画面サイズが 4096x4096 より大きくなる場合があります。この制限を回避するには、次のいずれかの方法を使用します。

- 仮想マシンに VMware ハードウェア バージョン 11 (HWv11) 以降を使用する必要がある場合は、仮想マシンの VMX 構成ファイルに次の行を追加します。

```
mks.enable3d = TRUE
```

この解決策より、Linux オペレーティングシステムは Chrome などのソフトウェア アプリケーションで 3D 機能をサポートできるようになります。しかし、これは Linux システムのパフォーマンスに影響を与える可能性があります。

- 仮想マシンが VMware ハードウェア バージョン 10 を使用できる場合は、仮想マシンの VMX ファイルで次の行を使用します。

```
virtualHW.version = "10"
```

このソリューションは RHEL 6.8 および CentOS 6.8 には適用されません。

この解決策を Ubuntu 16.04 システムに適用するには、正規の Ubuntu リポジトリから最新のパッチをインストールする必要があります。

## vCPU とメモリ設定

2D または vSGA デスクトップのパフォーマンスを向上させるには、Linux 仮想マシン用で vCPU と仮想メモリを増設します。たとえば、2 つの vCPU と 2 GB の仮想メモリを設定します。

4 台のモニターがある場合など、マルチモニタを使用して大画面にする場合、仮想マシンに 4 つの vCPU と 4GB の仮想メモリを設定します。

2D または vSGA デスクトップでビデオを再生する場合、仮想マシンに 4 つの vCPU と 4GB の仮想メモリを設定します。

## 3D メモリ設定

vSGA の複数モニター環境でパフォーマンスを向上させるには、仮想マシンの [3D メモリ] 設定を 1GB 以上に設定します。

## Linux デスクトップでのセッション共同作業の設定

セッション共同作業機能を使用すると、他のユーザーを既存の Linux リモート デスクトップ セッションに招待できます。

### セッション共同作業のシステム要件

セッション共同作業機能を使用するには、Horizon 環境が特定の要件を満たしている必要があります。

表 1-11. セッション共同作業のシステム要件

コンポーネント	要件
クライアント システム	セッション オーナーおよびセッション共同作業者が 4.10 以降の Horizon Client for Windows、Mac、または Linux をクライアント システムにインストールしている必要があります。インストールしていない場合は、HTML Access 4.10 以降を使用してください。
Linux リモート デスクトップ	Linux 仮想デスクトップに Horizon Agent 7.7 以降がインストールされている必要があります。セッション共同作業機能をデスクトップ プールまたは VDI レベルで有効にしておく必要があります。
Connection Server	Connection Server インスタンスはエンタープライズライセンスを使用します。
表示プロトコル	VMware Blast

セッション共同作業機能を使用する方法については、『Horizon Client』ドキュメントを参照してください。

### 構成ファイルでのセッション共同作業オプションの設定

セッション共同作業機能を有効または無効にするには、`/etc/vmware/viewagent-custom.conf` ファイルで次のオプションを設定します。

#### ■ CollaborationEnable

共同作業セッションの設定を行うには、`/etc/vmware/config` ファイルで次のオプションを設定します。

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

詳細については、[Linux デスクトップでの構成ファイルのオプション設定](#)を参照してください。

### セッション共同作業機能の制限事項

共同作業セッションでは、次のリモート デスクトップ機能は使用できません。

- USB リダイレクト

- オーディオ入力リダイレクト
- クライアント ドライブのリダイレクト
- スマート カード リダイレクト
- クリップボード リダイレクト

共同作業セッションでは、リモート デスクトップの解像度は変更できません。

同じクライアント コンピュータで複数の共同作業セッションを実行することはできません。

# デスクトップ デプロイのための Linux 仮想マシンの準備

## 2

Linux デスクトップのセットアップには、Linux 仮想マシンの作成およびリモート デスクトップ デプロイのためのオペレーティング システムの準備が含まれます。

この章には、次のトピックが含まれています。

- 仮想マシンを作成して、Linux をインストールする
- リモート デスクトップ デプロイ用の Linux マシンの準備
- Horizon Agent 用依存パッケージのインストール

## 仮想マシンを作成して、Linux をインストールする

Horizon 7 にデプロイする各リモート デスクトップに対して vCenter Server で新しい仮想マシンを作成します。仮想マシンに Linux ディストリビューションをインストールする必要があります。

### 前提条件

- デプロイする環境がサポートする Linux デスクトップの要件を満たしていることを確認します。[Horizon 7 for Linux のシステム要件](#)を参照してください。
- vCenter Server で仮想マシンを作成し、ゲスト OS をインストールする手順について理解しておきます。Horizon 7 での仮想デスクトップのセットアップ ドキュメントの「仮想マシンの作成および準備」を参照してください。
- 仮想マシンで使用するモニターのビデオ メモリ (vRAM) の設定を理解しておきます。[Horizon 7 for Linux のシステム要件](#)を参照してください。

### 手順

- 1 vSphere Web Client または vSphere Client で新しい仮想マシンを作成します。

## 2 カスタム構成オプションを構成します。

- a 仮想マシンを右クリックし、[設定の編集] をクリックします。
- b vCPU の数と vMemory のサイズを指定します。

必要な設定については、お使いの Linux ディストリビューションのインストール ガイドのガイドラインに従ってください。

たとえば、Ubuntu 18.04 では、2048 MB の vMemory と 2 台の vCPU を構成することが指定されています。

- c [ビデオ カード] を選択して、ディスプレイの数とビデオ メモリ (vRAM) の合計を指定します。

VMware のドライバを使用し、2D や vSGA を使用する仮想マシンについては、vSphere Web Client で vRAM のサイズを設定します。vRAM のサイズは、NVIDIA のドライバを使用する vDGA や NVIDIA GRID vGPU マシンには影響しません。

必要な設定については、[Horizon 7 for Linux のシステム要件](#)のガイドラインに従ってください。ビデオ メモリ計算ツールは使用しないでください。

## 3 仮想マシンをパワーオンして、Linux ディストリビューションをインストールします。

## 4 特定の Linux ディストリビューションに使用するデスクトップ環境を設定します。

詳細については、[Horizon 7 for Linux のシステム要件](#)でデスクトップ環境のセクションを参照してください。

## 5 システムのホスト名が 127.0.0.1 に対して解決可能であることを確認してください。

# リモート デスクトップ デプロイ用の Linux マシンの準備

Horizon 7 をデプロイした環境でデスクトップとして使用するために Linux マシンを準備するには、特定のタスクを実行する必要があります。

Linux マシンを Horizon 7 で管理できるようにするには、マシンが接続サーバと通信する必要があります。Linux マシンが完全修飾ドメイン名 (FQDN) を使用して接続サーバ インスタンスに ping を送信できるように、Linux マシンのネットワークを構成する必要があります。

Open VMware Tools (OVT) は、RHEL 7、CentOS 7、SLED 12、および SLES 12 のマシンに事前にインストールされています。リモート デスクトップとして使用できるように、これらのマシンのいずれかを準備する場合は、下記の手順 1 ～ 5 をスキップできます。これらの手順には、手動でインストーラを実行して VMware Tools をインストールする方法が記載されています。

Ubuntu 16.04/18.04 マシンを使用している場合は、OVT をインストールします。このマシンをリモート デスクトップとして使用する場合は、下記の手順 1 ～ 5 をスキップでき、次のコマンドを使用して 16.04/18.04 マシンに OVT を手動でインストールできます。

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

### 前提条件

- 新しい仮想マシン (VM) が vCenter Server で作成され、Linux ディストリビューションがマシンにインストールされていることを確認します。

- Linux 仮想マシンへの VMware Tools のマウントとインストールの手順を理解しておきます。vSphere 仮想マシン管理ドキュメントにある「Linux 仮想マシンでの VMware Tools の手動インストールまたはアップグレード」を参照してください。
- Linux マシンが DNS を介して解決できるように構成する手順を理解しておきます。これらの手順は、Linux ディストリビューションとリリースによって異なります。手順については、Linux ディストリビューションとリリースのドキュメントを参照してください。

## 手順

- 1 vSphere Web Client または vSphere Client で、VMware Tools 仮想ディスクを仮想マシンにマウントします。
- 2 VMware Tools のインストーラ ファイル `VMwareTools.x.x.x-xxxx.tar.gz` を右クリックして、[Extract to (展開先)] をクリックして、Linux ディストリビューションのデスクトップを選択します。

`vmware-tools-distrib` フォルダがデスクトップに展開されます。

- 3 仮想マシンで、root としてログインし、ターミナル ウィンドウを開きます。
- 4 VMware Tools の tar 形式のインストーラ ファイルを解凍します。

例：

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

- 5 インストーラを実行して VMware Tools を構成します。

Linux ディストリビューションによってこのコマンドは若干異なる場合があります。例：

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

通常、インストーラ ファイルの実行が終了した後に、`vmware-config-tools.pl` 構成ファイルが実行されます。

- 6 Linux マシンのホスト名を `/etc/hosts` ファイルの 127.0.0.1 にマッピングします。

RHEL、CentOS、SLES、SLED の場合、ホスト名は自動的にマッピングされないため、127.0.0.1 に手動でマッピングする必要があります。Ubuntu の場合、デフォルトでマッピングされるため、この手順は不要です。この手順はデスクトップを一括デプロイする場合も不要です。このマッピングがクローン作成プロセスによって追加されるためです。

---

**注：** Horizon Agent をインストールした後に Linux マシンのホスト名を変更する場合は、新しいホスト名を `/etc/hosts` ファイルの 127.0.0.1 にマッピングする必要があります。マッピングしないと、古いホスト名が引き続き使用されます。

---

- 7 RHEL 7 および CentOS 7 については、`virbr0` が無効になっていることを確認します。

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

- 8 ポッドにある Horizon Connection Server インスタンスを DNS を介して解決できることを確認します。

- 9 デフォルトのランレベルが 5 になるように Linux マシンを構成します。

Linux デスクトップが機能するには、ランレベルを 5 にする必要があります。

- 10 OpenLDAP サーバを使用して認証するように構成された Ubuntu マシンに、マシンで完全修飾ドメイン名を設定します。

この手順によって、Horizon Administrator の [セッション] ページの [ユーザー] フィールドにこの情報を正しく表示できるようになります。/etc/hosts ファイルを次のように編集します。

a # nano /etc/hosts

b 完全修飾ドメイン名を追加します。たとえば、127.0.0.1 hostname.domainname hostname のように追加します。

c 終了してファイルを保存します。

- 11 SUSE については、[DHCP 経由でホスト名を変更] を無効にします。ホスト名またはドメイン名を設定します。

a Yast では、[ネットワーク設定] をクリックします。

b [ホスト名/DNS] タブをクリックします。

c [DHCP 経由でホスト名を変更] の選択を解除します。

d ホスト名とドメイン名を入力します。

e [OK] をクリックします。

VMware Tools をインストールした後に、Linux カーネルをアップグレードすると、VMware Tools の実行が停止する場合があります。この問題を解決するには、<http://kb.vmware.com/kb/2050592> を参照してください。

## Horizon Agent 用依存パッケージのインストール

Horizon Agent for Linux には、Linux ディストリビューションに一意の依存パッケージがあります。Horizon Agent for Linux をインストールする前に、これらのパッケージをインストールする必要があります。

### 前提条件

新しい仮想マシン (VM) が vCenter Server で作成され、Linux ディストリビューションがマシンにインストールされていることを確認します。



## 手順

- 1 デフォルトではインストールまたはアップグレードされない必須パッケージをインストールします。要件を満たしていないパッケージがあると、インストーラはインストールを中断します。

表 2-1. 必須の依存パッケージ

Linux ディストリビューション	パッケージ
RHEL 7.5	<code>yum install libappindicator-gtk3</code>
SLED 11 SP3/SP4 <code>xorg-x11-server</code> を 7.4.27.111.1 より後のバージョンにアップグレード します。	<code>zypper update xorg-x11-server</code>
SLES 12 SP1/SLED 12 SP1 SUSE リポジトリで <code>xf86-video-vmware</code> を 13.0.2-3.2 以降のバージョンにアップグレードします。	<ol style="list-style-type: none"> <li>1 SUSE 12 を登録して SUSE リポジトリを有効にします。 <code>SUSEConnect -r Registration Code -e Email</code></li> <li>2 <code>xf86-video-vmware</code> のバージョンを更新します。 <code>zypper update xf86-video-vmware</code></li> </ol>
SLES 12	<p>Horizon Agent をインストールする場合、SLES 12 Linux デスクトップに <code>python-gobject2</code> をインストールする必要があります。</p> <ol style="list-style-type: none"> <li>1 SUSE 12 を登録して SUSE リポジトリを有効にします。 <code>SUSEConnect -r Registration Code -e Email</code></li> <li>2 <code>python-gobject2</code> をインストールします。 <code>zypper install python-gobject2</code></li> </ol>
Ubuntu 16.04	<code>apt-get install python-dbus python-gobject</code>
Ubuntu 18.04	<code>apt-get install python python-dbus python-gobject</code>

- 2 Horizon Agent のオプション パッケージをインストールします。

- デフォルトでは、RHEL または CentOS 6.7 に `glibc-2.12-1.166.el6.x86_64` がインストールされていると、デッドロックが発生する可能性があります。その結果、デスクトップ接続は停止します。この問題を解決するには、オンライン リポジトリで `glibc` を最新バージョンにアップグレードする必要があります。

```
sudo yum install glibc
```

# Linux デスクトップの Active Directory 統合のセットアップ

# 3

Horizon 7 は、ユーザーの認証と管理に既存の Microsoft Active Directory (AD) インフラストラクチャを使用します。Linux デスクトップを Active Directory と統合すると、ユーザーは Active Directory ユーザー アカウントを使用して Linux デスクトップにログインできるようになります。

**注:** Horizon Agent は、Linux デスクトップとクライアント ユーザーが同じ Active Directory ドメインに参加していることを前提としています。デスクトップとユーザーが異なるドメインに存在する場合、Horizon Agent がデスクトップドメインをユーザー ドメインとして誤認識する可能性があります。

この章には、次のトピックが含まれています。

- [Linux と Active Directory の統合](#)
- [シングル サインオンの設定](#)
- [スマート カード リダイレクトの設定](#)
- [Linux デスクトップでの True SSO のセットアップ](#)

## Linux と Active Directory の統合

Linux と Microsoft Active Directory (AD) を統合するソリューションは複数ありますが、Horizon 7 for Linux デスクトップには、使用するソリューションと依存関係がありません。

次のソリューションは、Horizon 7 for Linux デスクトップ環境での動作が確認されています。

- OpenLDAP サーバ パススルー認証
- Microsoft Active Directory に対する SSSD (System Security Services Daemon) LDAP 認証
- Winbind ドメイン参加
- PBISO (PowerBroker Identity Services Open) 認証
- Samba オフライン ドメイン参加

LDAP ベースのソリューションを使用する場合は、テンプレート仮想マシンで設定を行う必要があります。クローン作成された仮想マシンで追加の手順を行う必要はありません。

**注:** 展開を簡単に行うため、Microsoft Active Directory に対する SSSD LDAP 認証のソリューションを使用します。

## OpenLDAP サーバ パススルー認証の使用

OpenLDAP サーバをセットアップし、パススルー認証 (PTA) メカニズムを使用して Active Directory でユーザー認証情報を検証できます。

OpenLDAP パススルー認証ソリューションには、おおまかに次のような手順が含まれます。

### 手順

- 1 LDAPS (Lightweight Directory Access Protocol over SSL) を有効にするには、Active Directory に証明書サービスをインストールします。
- 2 OpenLDAP サーバを設定します。
- 3 Active Directory から OpenLDAP サーバにユーザー情報 (パスワードを除く) を同期します。
- 4 パスワード検証を別のプロセス (saslauthd など) に委任するように OpenLDAP サーバを設定します。saslauthd は Active Directory に対してパスワード検証を実行できます。
- 5 LDAP クライアントを使用して OpenLDAP サーバでユーザー認証を行うように Linux デスクトップを設定します。

## Microsoft Active Directory に対する SSSD LDAP 認証の設定

Linux デスクトップに SSSD (System Security Services Daemon) を設定すると、Windows Active Directory に LDAP 認証を使用できます。

SSSD LDAP 認証ソリューションを使用するには、次の手順に従います。

### 手順

- 1 LDAPS (Lightweight Directory Access Protocol Over Secure Socket Layer) を有効にするには、Active Directory サーバに証明書サービスをインストールします。
- 2 Microsoft Active Directory で LDAP 認証を直接使用するには、Linux デスクトップに SSSD を設定します。

## Windbind ドメイン参加ソリューションの使用

Kerberos ベースの認証ソリューションである Windbind ドメイン参加ソリューションでも Active Directory 認証を行うことができます。

Windbing ドメイン参加ソリューションを設定するには、次の手順に従います。

### 手順

- 1 Linux デスクトップに winbind、samba、Kerberos パッケージをインストールします。
- 2 Linux デスクトップを Microsoft Active Directory に参加させます。

## 次のステップ

Winbind ドメイン参加ソリューションまたは他の Kerberos 認証ベースのソリューションを使用する場合は、テンプレート仮想マシンを Active Directory に参加させ、クローン作成された仮想マシンを Active Directory に再度参加させます。たとえば、次のコマンドを使用します。

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

Winbind ソリューション用のクローンが作成された仮想マシンで、ドメイン参加コマンドを実行するには、次のオプションを使用します。

- SSH または vSphere PowerCLI などの各仮想マシンにリモート接続してコマンドを実行します。スクリプトの詳細については、[8 章 手動デスクトップ プールのための Horizon 7 の一括デプロイ](#) を参照してください。
- コマンドをシェル スクリプトに追加し、`/etc/vmware/viewagent-custom.conf` ファイルで `RunOnceScript` オプションに Horizon Agent へのスクリプト パスを指定します。詳細については、[Linux デスクトップでの構成ファイルのオプション設定](#) を参照してください。

## PBISO (PowerBroker Identity Services Open) 認証の設定

PBISO (PowerBroker Identity Services Open) 認証方法は、オフライン ドメイン参加をサポートするソリューションです。

次の手順に従って、PBISO を使用する Active Directory に Linux デスクトップを参加させます。

### 手順

- 1 <https://www.beyondtrust.com/products/powerbroker-identity-services-open/> から PBISO 8.5.6 以降をダウンロードします。
- 2 Linux 仮想マシンに PBISO をインストールします。

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

- 3 Horizon 7 Agent for Linux をインストールします。
- 4 PBISO を使用して、Linux デスクトップを Active Directory ドメインに参加させます。

次の例では、**lxdc.vdi** がドメイン名、**administrator** がドメインのユーザー名です。

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

- 5 ドメイン ユーザーのデフォルト設定を行います。

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

6 `/etc/pamd.d/common-session` ファイルを編集します。

- a `session sufficient pam_lsass.so` という行を探します。
- b この行を `session [success=ok default=ignore] pam_lsass.so` で置き換えます。

**注:** Horizon Agent for Linux を再インストールまたは更新した後に、この手順を繰り返す必要があります。

7 Ubuntu 16.04 の場合、次の行を `/usr/share/lightdm/lightdm.conf.d/50-unity-greeter.conf` 構成ファイルに追加します。

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

**注:** Ubuntu 18.04 を使用している場合、`lightdm` 構成ファイルを変更する必要はありません。

8 システムを再起動してログインします。

#### 次のステップ

#### 注:

- `/opt/pbis/bin/config AssumeDefaultDomain` オプションが **false** に設定されている場合は `/etc/vmware/viewagent-custom.conf` ファイルで `SSOUserFormat=<username>@<domain>` 設定を更新する必要があります。
- Horizon インスタント クローンのフローティング デスクトップ プールの機能を使用する場合は、クローン作成された仮想マシンに新しいネットワーク アダプタを追加するときに DNS サーバの設定が失われるように、Linux システムの `resolv.conf` ファイルを変更します。次の例では、Ubuntu 16.04 システムを使用しています。この例を参考に、`/etc/resolvconf/resolv.conf.d/head` ファイルに必要な行を追加してください。

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

## Ubuntu デスクトップでの Samba オフライン ドメイン参加の設定

Horizon 7Linux デスクトップ環境のインスタント クローン仮想マシンで SSO をサポートするには、Ubuntu システムのマスター Linux 仮想マシンで Samba を設定します。

Ubuntu システムのインスタント クローン Linux デスクトップで、Samba を使用して Active Directory へのオフライン ドメイン参加を行うには、次の手順に従います。

#### 手順

- 1 マスター Linux 仮想マシンで、`winbind` パッケージと `samba` パッケージをインストールします。`smbfs` や `smbclient` など、依存関係のある他のライブラリもインストールします。
- 2 次のコマンドを使用して、Samba `tdb-tools` パッケージをインストールします。

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

- 3 Horizon 7 Agent for Linux をインストールします。
- 4 次の例のように、`/etc/samba/smb.conf` 構成ファイルの内容を編集します。

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

- 5 次の例のように、`/etc/krb5.conf` 構成ファイルの内容を編集します。

```
[libdefaults]
default_realm = EXAMPLE.COM

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

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

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

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

- 6 次の例のように、`/etc/nsswitch.conf` 構成ファイルを編集します。

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

- 7 ホスト名が正しく、システムの日付と時刻が DNS システムと同期されていることを確認します。

- 8 Linux 仮想マシンが Samba を使用してドメインに参加していることを Horizon Agent for Linux に通知できるように、`/etc/vmware/viewagent-custom.conf` ファイルで次のオプションを設定します。

```
OfflineJoinDomain=samba
```

- 9 システムを再起動して再びログインします。

## シングル サインオンの設定

シングル サインオン (SSO) を設定するには、いくつかの手順を実行する必要があります。

Horizon のシングル サインオン モジュールは Linux で PAM (プラグ可能な認証モジュール) と通信を行い、Linux を Active Directory (AD) と連携するために使用する方法には依存しません。Horizon Single Sign-on は、Linux を Active Directory と連携する OpenLDAP と Winbind のソリューションで動作することが分かっています。

デフォルトの場合、SSO では、Active Directory の `sAMAccountName` 属性がログイン ID であると想定されます。OpenLDAP か Winbind ソリューションを使用する場合、正しいログイン ID を SSO に使用するには、次の構成手順を実行する必要があります。

- OpenLDAP では、`sAMAccountName` を `uid` に設定します。
- Winbind では、次のステートメントを構成ファイル `/etc/samba/smb.conf` に追加します。

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

ユーザーがドメイン名を指定してログインする必要がある場合は、`SSOUserFormat` オプションを Linux デスクトップで設定する必要があります。詳細については、[Linux デスクトップでの構成ファイルのオプション設定](#)を参照してください。SSO では常に、大文字で短いドメイン名が使用されます。たとえば、ドメインが `mydomain.com` である場合、SSO では `MYDOMAIN` がドメイン名として使用されます。このため、`SSOUserFormat` オプションを設定するときには、`MYDOMAIN` を指定する必要があります。短いドメイン名と長いドメイン名については、次のルールが適用されます。

- OpenLDAP では、大文字で短いドメイン名を使用する必要があります。
- Winbind では、長いドメイン名と短いドメイン名が両方ともサポートされます。

Active Directory ではログイン名で特殊文字がサポートされますが、Linux ではサポートされません。このため、SSO のセットアップ時には、特殊文字をログイン名に使用しないでください。

Active Directory では、ユーザーの `UserPrincipalName` (UPN) 属性と `sAMAccount` 属性が一致せずに、ユーザーが UPN でログインすると、SSO は失敗します。たとえば、Active Directory の `mycompany.com` にユーザー `juser` が存在しているときに、ユーザーの UPN が `juser@mycompany.com` ではなく `juser123@mycompany.com` の場合、SSO が失敗します。ユーザーが、`sAMAccount` に保存されている名前を使用してログインすると、これを回避できます。たとえば、`juser` のように追加します。

Horizon 7 では、ユーザー名の太文字と小文字を区別する必要はありません。Linux オペレーティングシステムで、太文字と小文字を区別しないユーザー名を処理できることを確認してください。

- Winbind では、ユーザー名の太文字と小文字がデフォルトで区別されません。

- OpenLDAP では、Ubuntu が NSCD を使用してユーザーを認証し、大文字と小文字がデフォルトで区別されません。RHEL と CentOS は SSSD を使用してユーザーを認証し、大文字と小文字がデフォルトで区別されます。設定を変更するには、ファイル `/etc/sss/sss.conf` を編集して次の行を `[domain/default]` セクションに追加します。

```
case_sensitive = false
```

Linux デスクトップに複数のデスクトップ環境がインストールされている場合には、[デスクトップ環境](#)を参照して、SSO で使用するデスクトップ環境を選択してください。

## スマート カード リダイレクトの設定

スマート カード リダイレクトを設定するには、いくつかの手順を実行する必要があります。

### スマート カード リダイレクトの概要

スマート カード リダイレクトは、指定したバージョンの Horizon Agent がインストールされている次の Linux ディストリビューションのデスクトップでサポートされます。

表 3-1. スマート カード リダイレクトのシステム要件

Linux ディストリビューション	Horizon Agent
RHEL 7.1 以降	Horizon Agent 7.8 以降
RHEL 6.6 以降	Horizon Agent 6.2.1 以降
Ubuntu 18.04/16.04	Horizon Agent 7.9 以降
SLED/SLES 12.x SP3	Horizon Agent 7.9 以降
SLED 11.x SP4	Horizon Agent 7.9 以降

Horizon Agent をインストールする前に、SELinux を無効にする必要があります。スマート カード リダイレクトのコンポーネントを明確に選択する必要もあります。コンポーネントはデフォルトでは選択されません。詳細については、[install\\_viewagent.sh コマンドライン オプション](#)を参照してください。

仮想マシンでスマート カード リダイレクト機能が有効になっていると、vSphere Client の USB リダイレクトはスマート カードで動作しません。

スマート カード リダイレクトでは、1 つのスマート カード リーダのみがサポートされます。複数のリーダをクライアント デバイスに接続すると、この機能は動作しません。



スマート カード リダイレクトでは、カードで 1 つの証明書のみがサポートされます。複数の証明書がカードに存在する場合、最初のスロットの証明書が使用され、その他の証明書は無視されます。この動作は Linux の制限です。

**注:** スマート カード リダイレクトは、Linux デスクトップの PIV カードをサポートします。Horizon Client for Linux で、ブローカーの認証を PIV カードで行う場合、TLSv1.2 サポートを PIV スマート カードに設定し、SSL エラーを回避する必要があります。VMware のナレッジベースの記事 <http://kb.vmware.com/kb/2150470> にある解決策に従ってください。

**注:** Horizon 7 バージョン 7.0.1 以降では、スマート カード SSO が有効になっています。RHEL 6 デスクトップはスマート カード SSO をサポートしていますが、RHEL 7 デスクトップはこの機能をサポートしていません。

## スマート カード リダイレクトの設定

スマート カード リダイレクトを設定するには、次のタスクを実行します。

- 1 Linux ディストリビュータとスマート カード ベンダーからの指示に従って、デスクトップ用のスマート カードを設定します。
- 2 [スマート カード リダイレクトでの RHEL デスクトップと Active Directory の統合](#)の説明に従って、デスクトップと Active Directory を統合します。
- 3 [RHEL デスクトップのスマート カード リダイレクトの設定](#)の説明に従って、スマート カード リダイレクトの設定手順を完了します。

## スマート カード リダイレクトでの RHEL デスクトップと Active Directory の統合

RHEL デスクトップでスマート カード リダイレクトをサポートするには、Samba と Winbind ソリューションを使用して、デスクトップと Active Directory (AD) ドメインを統合します。

スマート カード リダイレクトで RHEL デスクトップと Active Directory ドメインを統合するには、次の手順に従います。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名

**注:** スマート カード リダイレクトは、RHEL 6 以降または RHEL 7.1 以降のデスクトップでサポートされています。

## 手順

- 1 RHEL デスクトップに、必要なパッケージをインストールします。

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

- 2 システム接続のネットワーク設定を編集します。NetworkManager コントロール パネルを開き、システム接続の [IPv4 Settings (IPv4 設定)] に移動します。IPv4 の方法で、[Automatic (DHCP) (自動 (DHCP)))] を選択します。[DNS] テキスト ボックスに、DNS ネーム サーバの IP アドレスを入力します。[適用] をクリックします。
- 3 次のコマンドを実行して、RHEL デスクトップの完全修飾ドメイン名 (FQDN) が返されることを確認します。

```
# hostname -f
```

- 4 次の例のように、`/etc/resolve.conf` 構成ファイルを編集します。

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

- 5 RHEL デスクトップで、セキュリティが強化された Linux (SELinux) を無効にします。次の例のように、`/etc/selinux/config` 構成ファイルを編集します。

```
SELINUX=disabled
```

- 6 次の例のように、`/etc/krb5.conf` 構成ファイルを編集します。

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

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

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

- 7 次の例のように、`/etc/samba/smb.conf` 構成ファイルを編集します。

```
[global]
    workgroup = MYDOMAIN
    password server = ads-hostname
    realm = MYDOMAIN.COM
    security = ads
```

```
idmap config * : range = 16777216-33554431
template homedir = /home/MYDOMAIN/%U
template shell = /bin/bash
kerberos method = secrets and keytab
winbind use default domain = true
winbind offline logon = false
winbind refresh tickets = true

passdb backend = tdbsam
```

8 `authconfig-gtk` ツールを開き、次のように設定を行います。

- a [Identity & Authentication (ID と認証)]タブを選択します。ユーザー アカウントのデータベースに [Winbind] を選択します。
- b [Advanced Options (詳細オプション)] タブを選択して、[Create home directories on the first login (最初のログインでホーム ディレクトリを作成)] チェック ボックス選択します。
- c [Identity & Authentication (ID と認証)]タブを選択して、[Join Domain (ドメインに参加)] をクリックします。変更の保存を確認するアラートで、[Save (保存)] をクリックします。
- d プロンプトが表示された、ドメインの管理者のユーザー名とパスワードを入力して、[OK] をクリックします。

RHEL デスクトップが Active Directory ドメインに参加します。

9 PAM Winbind でチケットのキャッシュを設定します。次のような行が含まれるように、`/etc/security/pam_winbind.conf` 構成ファイルを編集します。

```
[global]

# authenticate using kerberos
;krb5_auth = yes

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

10 Winbind サービスを再起動します。

```
# sudo service winbind restart
```

11 Active Directory への参加を確認するには、次のコマンドを実行し、正しい出力が返されていることを確認します。

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

12 システムを再起動して再びログインします。

次のステップ

[RHEL デスクトップのスマート カード リダイレクトの設定](#)

## スマート カード リダイレクトでの Ubuntu デスクトップと Active Directory の統合

Ubuntu デスクトップでスマート カード リダイレクトをサポートするには、Samba と Winbind ソリューションを使用して、デスクトップと Active Directory (AD) ドメインを統合します。

スマート カード リダイレクトで Ubuntu デスクトップと Active Directory ドメインを統合するには、次の手順に従います。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名
ads-hostname.mydomain.com	Active Directory サーバの完全修飾ドメイン名 (FQDN)
mytimeserver.mycompany.com	NTP タイム サーバの DNS 名
AdminUser	Linux デスクトップ管理者のユーザー名

### 手順

- 1 Ubuntu デスクトップで、`/etc/hostname` 構成ファイルを編集して、デスクトップのホスト名を定義します。
- 2 DNS を設定します。
  - a `/etc/hosts` 構成ファイルに DNS サーバ名と IP アドレスを追加します。
  - b 次の例のように、DNS ネーム サーバの IP アドレスと Active Directory ドメインの DNS 名を `/etc/network/interfaces` 構成ファイルに追加します。

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

### 3 resolvconf パッケージをインストールします。

- a インストール コマンドを実行します。

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

システムにパッケージのインストールと再起動を許可します。

- b 次の例のように、`/etc/resolv.conf` ファイルの DNS 設定を確認します。

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

### 4 ネットワークの時刻同期を設定します。

- a ntpdate パッケージをインストールします。

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

- b 次の例のように、NTP サーバの情報を `/etc/systemd/timesyncd.conf` 構成ファイル に追加します。

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

### 5 NTP サービスを再起動します。

```
sudo service ntpdate restart
```

### 6 必要な Active Directory join パッケージをインストールします。

- a インストール コマンドを実行します。

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

- b インストール プロンプトでデフォルトの Kerberos レルムが要求されたら、Active Directory ドメインの DNS 名を大文字で入力します (例: MYDOMAIN.COM)。次に、[OK] を選択します。

### 7 次の例のように、`/etc/krb5.conf` 構成ファイルを編集します。

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

[realms]
    MYDOMAIN.COM = {
        kdc = ads-hostname.mydomain.com
```

```

    admin_server = ads-hostname.mydomain.com
    default_domain = ads-hostname.mydomain.com
    pkinit_anchors = FILE:/etc/pki/nssdb/certificate.pem
    pkinit_cert_match = <KU>digitalSignature
    pkinit_kdc_hostname = ads-hostname.mydomain.com
}

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

```

- 8 Kerberos 認証を確認するには、次のコマンドを実行します。

```

# kinit Administrator@MYDOMAIN.COM

# klist

```

コマンドから次のような出力が返されることを確認します。

```

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

```

- 9 次の例のように、`/etc/samba/smb.conf` 構成ファイルを編集します。

```

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

```

## 10 Active Directory ドメインに参加し、統合を確認します。

- a Active Directory join コマンドを実行します。

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

- b 次の例のように、`/etc/nsswitch.conf` 構成ファイルを変更します。

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

- c Active Directory への参加結果を確認するには、次のコマンドを実行し、正しい出力が返されていることを確認します。

```
# wbinfo -u
# wbinfo -g
```

- d Winbind ネーム サービスのスイッチを確認するには、次のコマンドを実行し、正しい出力が返されていることを確認します。

```
# getent group|grep 'domain admins'
# getent passwd|grep 'ads-hostname'
```

## 11 すべての PAM プロファイルを有効にします。

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

PAM の設定画面で、[Create home directory on login (ログイン時にホーム ディレクトリを作成)] を含むすべての PAM プロファイルを選択して、[OK] を選択します。

## 12 Ubuntu 16.04 では、ログイン画面でユーザー スイッチを有効にします。次の例のように、`/usr/share/lightdm/lightdm.conf.d/50-ubuntu.conf` ファイルを変更します。

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

### 次のステップ

[Ubuntu デスクトップのスマート カード リダイレクトの設定](#)

## スマート カード リダイレクトでの SLED/SLES デスクトップと Active Directory の統合

SLED/SLES デスクトップでスマート カード リダイレクトをサポートするには、Samba と Winbind ソリューションを使用して、デスクトップと Active Directory (AD) ドメインを統合します。

スマート カード リダイレクトで SLED/SLES デスクトップと Active Directory ドメインを統合するには、次の手順に従います。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名
ads-hostname.mydomain.com	Active Directory サーバの完全修飾ドメイン名 (FQDN)
mytimeserver.mycompany.com	NTP タイム サーバの DNS 名
AdminUser	Linux デスクトップ管理者のユーザー名

### 手順

1 SLED/SLES デスクトップのネットワーク設定を行います。

- a Define the host name of the desktop by editing the `/etc/hostname` と `/etc/hosts` 構成ファイルを編集して、デスクトップのホスト名を定義します。
- b DNS サーバの IP アドレスを設定して、[自動 DNS] を無効にします。SLED 11 SP4 および SLES 12 SP3 の場合は、[DHCP 経由でホスト名を変更] も無効にします。
- c ネットワークの時刻同期を設定するには、次の例のように、NTP サーバの情報を `/etc/ntp.conf` ファイルに追加します。

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

2 必要な Active Directory join パッケージをインストールします。

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



### 3 必要な構成ファイルを編集します。

- a 次の例のように、`/etc/samba/smb.conf` ファイルを編集します。

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

[homes]
    ...
```

- b 次の例のように、`/etc/krb5.conf` ファイルを編集します。

```
[libdefaults]
    default_realm = MYDOMAIN.COM
    clocks skew = 300

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

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

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

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

- c 次の例のように、`/etc/security/pam_winbind.conf` ファイルを編集します。

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

- d 次の例のように、`/etc/nsswitch.conf` ファイルを編集します。

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

- 4 次の例のように、Active Directory ドメインに参加します。

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

- 5 Winbind サービスを有効にします。

- a Winbind を有効にして開始するには、次の一連のコマンドを実行します。

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

- b Active Directory ユーザーが Linux サーバを再起動せずにデスクトップにログインできるように、次の一連のコマンドを実行します。

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

- 6 Active Directory に参加できていることを確認するには、次のコマンドを実行し、正しい出力が返されていることを確認します。

```
# wbinfo -u

# wbinfo -g
```

## 次のステップ

### [SLED/SLES デスクトップのスマート カード リダイレクトの設定](#)

## RHEL デスクトップのスマート カード リダイレクトの設定

RHEL デスクトップでスマート カード リダイレクトを設定するには、この機能が依存するライブラリ、スマート カード認証で使用するルート CA 証明書、必要な PC/SC Lite ライブラリをインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

RHEL デスクトップのスマート カード リダイレクトを設定するには、次の手順に従います。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名

**注:** スマート カード リダイレクトは、RHEL 6 または RHEL 7.1 以降のデスクトップでサポートされています。

## 前提条件

### スマート カード リダイレクトでの RHEL デスクトップと Active Directory の統合

#### 手順

- 1 必要なライブラリをインストールします。

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

- 2 ルート CA 証明書をインストールします。

- a ルート CA 証明書をダウンロードし、デスクトップの /tmp/certificate.cer に保存します。[ルート CA 証明書をエクスポートする方法](#)を参照してください。
- b ダウンロードしたルート CA 証明書を .pem ファイルに転送します。

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

- c certutil コマンドを使用して、ルート CA 証明書をシステム データベース /etc/pki/nssdb にインストールします。

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

- d ルート CA 証明書を /etc/pam\_pkcs11/cacerts にディレクトリにコピーします。

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

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

- 3 [アプリケーション] - [Sundry] - [認証] の順に移動し、[スマート カード サポートを有効にする] チェック ボックスを選択して [適用] をクリックします。

- 4 スマート カードのドライバをコピーし、システム データベース (/etc/pki/nssdb) にドライバ ライブラリを追加します。

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

- 5 次の例のように、/etc/pam\_pkcs11/pam\_pkcs11.conf 構成ファイルの モジュール 設定を編集します。

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

- 6 次の例に似たコンテンツが含まれるように、/etc/pam\_pkcs11/cn\_map 構成ファイルを編集します。追加する情報の詳細については、スマート カードの証明書にあるユーザー情報を参照してください。

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

- 7 次の例のように、/etc/krb5.conf/ 構成ファイルを編集します。

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

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

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

- 8 次のような行が含まれるように、/etc/pam.d/system-auth 構成ファイルを編集します。

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

## 9 PC/SC デーモンを再起動します。

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

## 10 RHEL ディストリビューションに必要な PC/SC Lite バージョンをインストールします。

- RHEL 7.x の場合は、PC/SC Lite バージョン 1.8.8 をインストールします。

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

- RHEL 6.x の場合は、PC/SC Lite バージョン 1.7.4 をインストールします。

```
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-conffdir=/etc --enable-ipcdire=/var/run --disable-libusb --disable-serial --disable-
usb
--disable-libudev
make
make install
service pcscd start
```

## 11 Horizon Agent パッケージをインストールして、スマート カード リダイレクトを有効にします。

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

**注:** RHEL 6.x の場合は、View Agent 6.2.1 以降をインストールする必要があります。RHEL 7.x の場合は、Horizon Agent 7.8 以降をインストールする必要があります。

## 12 システムを再起動して再びログインします。

## Ubuntu デスクトップのスマート カード リダイレクトの設定

Ubuntu デスクトップでスマート カード リダイレクトを設定するには、スマート カードの信頼された認証をサポートするため、機能が依存するライブラリとルート CA 証明書をインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名
ads-hostname.mydomain.com	Active Directory サーバの完全修飾ドメイン名 (FQDN)
mytimeserver.mycompany.com	NTP タイム サーバの DNS 名
AdminUser	Linux デスクトップ管理者のユーザー名

### 前提条件

#### スマート カード リダイレクトでの Ubuntu デスクトップと Active Directory の統合

### 手順

- 1 必要なライブラリをインストールします。

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

- 2 ルート CA 証明書をインストールします。

- a ルート CA 証明書をダウンロードし、デスクトップの /tmp/certificate.cer に保存します。[ルート CA 証明書をエクスポートする方法](#)を参照してください。
- b ダウンロードしたルート CA 証明書を .pem ファイルに転送します。

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

- c `certutil` コマンドを使用して、ルート CA 証明書をシステム データベース `/etc/pki/nssdb` にインストールします。

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

- d ルート CA 証明書を `/etc/pam_pkcs11/cacerts` にディレクトリにコピーします。

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

- 3 `pkcs11` ハッシュ ファイルを作成します。

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

#### 4 必要なドライバをコピーし、必要なライブラリ ファイルを nssdb ディレクトリに追加します。

- a 次のコマンドを実行します。

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

- b 予期した証明書が正常に読み込まれていることを確認します。

```
# certutil -L -d /etc/pki/nssdb

Certificate Nickname

rootca
```

- c 予期したライブラリが正常に追加されていることを確認します。

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

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

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

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

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



## 5 pam\_pkcs11 ライブラリを設定します。

- a デフォルトのサンプル コンテンツを使用して、pam\_pkcs11 ファイルを作成します。

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

- b 次の例のように、/etc/pam\_pkcs11/pam\_pkcs11.conf ファイルを編集します。

```
use_pkcs11_module = mysc;

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

- c 次の行を含むように、/etc/pam\_pkcs11/cn\_map ファイルを編集します。

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

## 6 PAM 認証を設定します。

- a /etc/pam.d/gdm-password 構成ファイルを編集します。次の例のように、pam\_pkcs11.so 認証行を common-auth 行の前に配置します。

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

- b Ubuntu 16.04 の場合は、/etc/pam.d/lightdm 構成ファイルを編集します。次の例のように、pam\_pkcs11.so 認証行を common-auth 行の前に配置します。

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

- c Ubuntu 16.04 の場合は、/etc/pam.d/unity 構成ファイルを編集します。次の例のように、pam\_pkcs11.so 認証行を common-auth 行の前に配置します。

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

- 7 スマート カード ハードウェアとスマート カードにインストールされている証明書を確認するには、次のコマンドを実行します。

```
# pcsc_scan

# pkcs11_listcerts

# pkcs11_inspect
```

## 8 スマート カードの取り外し時にロックされるように、Gnome スクリーンセーバーを設定します。

- a スクリーンセーバーのパッケージをインストールします。

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

- b スクリーンセーバーを設定するには、次の例のように、`etc/pam_pkcs11/pkcs11_eventmgr.conf` ファイルを編集します。

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

    # show debug messages?
    debug = false;

    # polling time in seconds
    polling_time = 1;

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

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

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

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

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

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

```

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

```

- C `pkcs11_eventmgr` を実行します。

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

- 9 Horizon Agent パッケージをインストールして、スマート カード リダイレクトを有効にします。

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

**注:** Horizon Agent 7.9 以降をインストールする必要があります。

- 10 システムを再起動して再びログインします。

## SLED/SLES デスクトップのスマート カード リダイレクトの設定

SLED/SLES デスクトップでスマート カード リダイレクトを設定するには、スマート カードの信頼された認証をサポートするため、機能が依存するライブラリとルート CA 証明書をインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
<code>dns_IP_ADDRESS</code>	DNS ネーム サーバの IP アドレス
<code>mydomain.com</code>	Active Directory ドメインの DNS 名
<code>MYDOMAIN.COM</code>	Active Directory ドメインの DNS 名。すべて大文字にします。
<code>MYDOMAIN</code>	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
<code>ads-hostname</code>	Active Directory サーバのホスト名
<code>ads-hostname.mydomain.com</code>	Active Directory サーバの完全修飾ドメイン名 (FQDN)
<code>mytimeserver.mycompany.com</code>	NTP タイム サーバの DNS 名
<code>AdminUser</code>	Linux デスクトップ管理者のユーザー名

### 前提条件

スマート カード リダイレクトでの SLED/SLES デスクトップと Active Directory の統合

## 手順

## 1 必要なライブラリ パッケージをインストールします。

- a PAM ライブラリと他のパッケージをインストールします。

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

- b 該当する手順で PC/SC ツールをインストールします。

- SLED/SLES 12 SP3 の場合、次の一連のコマンドを実行します。

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

- SLED 11 SP4 の場合は、この Web ページから PC/SC ツールをダウンロードしてインストールします。

[https://download.opensuse.org/repositories/security/chipcard/SLE\\_11\\_SP4/](https://download.opensuse.org/repositories/security/chipcard/SLE_11_SP4/)

## 2 ルート CA 証明書をインストールします。

- a ルート CA 証明書をダウンロードし、デスクトップの /tmp/certificate.cer に保存します。[ルート CA 証明書をエクスポートする方法](#)を参照してください。
- b ダウンロードしたルート CA 証明書を .pem ファイルに転送し、ハッシュ ファイルを作成します。

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

- c NSS データベースにトラスト アンカーをインストールします。

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

- d 必要なドライバをインストールします。

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

### 3 /etc/pam\_pkcs11/pam\_pkcs11.conf ファイルを編集します。

- a 行 `use_pkcs11_module = nss` を削除します。この場所に、行 `use_pkcs11_module = mysc` を追加します。
- b 次の例のように、`mysc` モジュールを追加します。

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

- c 次の例のように、共通名のマッパー構成を更新します。

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

- d 行 `use_mappers = ms` を削除します。この場所に、行 `use_mappers = cn, null` を追加します。

### 4 次の行を含むように、/etc/pam\_pkcs11/cn\_map 構成ファイルを編集します。

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

### 5 PAM の設定を変更します。

- a スマート カード認証を構成できるように、まず `pam_config` ツールを無効にします。

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

- b /etc/pam.d/ ディレクトリの下に、`common-auth-smartcard` という名前のファイルを作成します。次の内容をファイルに追加します。

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

- c SLED/SLES 12 SP3 の場合、`/etc/pam.d/gdm` と `/etc/pam.d/xscreensaver` の両方のファイルで、行 `auth include common-auth` を行 `auth include common-auth-smartcard` で置き換えます。
- d SLED 11 SP4 の場合、`/etc/pam.d/gdm` と `/etc/pam.d/xscreensaver` の両方のファイルで、行 `auth include common-auth` を行 `/etc/pam.d/gnome-screensaver` で置き換えます。

## 6 ファイアウォールを無効にします。

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

**注:** ファイアウォールが有効になっていると、スマート カード リダイレクトが失敗することがあります。

## 7 スマート カード リダイレクトに必要なライブラリ パッケージをインストールします。

- a SLED/SLES 12 SP3 および SLED 11 SP4 の場合は、次のインストール コマンドを実行します。

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

- b SLES 12 SP3 の場合は、`systemd-devel` をインストールします。

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

- c SLED 11 SP4 の場合は、次のパッケージをインストールします。

- perl-Error、autoconf、automake、flex rpm
- git、gcc
- libtool

## 8 Horizon Agent パッケージをインストールして、スマート カード リダイレクトを有効にします。

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

**注:** Horizon Agent 7.9 以降をインストールする必要があります。

## 9 システムを再起動して再びログインします。

## Linux デスクトップでの True SSO のセットアップ

True SSO 機能を使用すると、ユーザーは、スマート カード認証や RSA SecurID または RADIUS 認証を使用して VMware Identity Manager にログインした後、Linux 仮想デスクトップ、公開デスクトップまたはアプリケーションを使用するために、さらに Active Directory の認証情報を入力する必要がなくなります。

Active Directory (AD) 認証情報を使用してユーザーを認証する場合、True SSO 機能は必要ありません。この場合でも、True SSO を使用するように設定すると、デスクトップで Active Directory の認証情報と True SSO の両方をサポートできます。

Linux 仮想デスクトップ、公開デスクトップまたはアプリケーションに接続する場合、ユーザーはネイティブ Horizon Client または HTML Access の使用を選択できます。

True SSO には次の制限があります。

- この機能は、RHEL/CentOS 7、Ubuntu 16.04、18.04 および SLED/SLES 12.x SP3 のディストリビューションを実行しているデスクトップでのみサポートされます。

- RHEL/CentOS 7 デスクトップの場合、この機能はデフォルトのドメイン参加ツール、Samba、SSSD (System Security Services Daemon)、Kerberos ネットワーク認証プロトコルでのみサポートされます。

Linux 環境で True SSO をセットアップするには、次のタスクを実行します。

- 1 Horizon 7 環境で True SSO をセットアップして構成します。『Horizon 7 の管理』の「True SSO のセットアップ」を参照してください。
- 2 Linux ディストリビューションの手順に従って、デスクトップと Active Directory ドメインを統合します。
- 3 Linux ディストリビューションの手順に従って、デスクトップで True SSO を構成します。

## True SSO での RHEL/CentOS デスクトップと Active Directory の統合

Horizon 7 Linux デスクトップ環境のインスタンス クローン仮想マシンで True SSO をサポートするには、RHEL/CentOS システムのマスター Linux 仮想マシンで Samba を設定する必要があります。

RHEL 7 の `realmd` 機能により、ID ドメインを簡単に検出し、参加することができます。この機能は、システム自体をドメインに接続するのではなく、基盤となる Linux システムのサービス (SSSD や Winbind など) がドメインに接続するように設定します。Samba を使用して、RHEL/CentOS デスクトップから Active Directory へのオフライン ドメイン参加を実行するには、次の手順に従います。

### 前提条件

- Red Hat Enterprise Linux (RHEL) システムが Red Hat ネットワーク (RHN) に登録されているか、ローカルに `yum` ツールがインストールされている。
- Active Directory (AD) サーバが Linux システムの DNS で解決できる。
- Linux システムで NTP (Network Time Protocol) が設定されている。

### 手順

- 1 RHEL/CentOS システムが Active Directory サーバを検出できることを確認します。次のコマンドを実行します。 `ADdomain.example.com` は、Active Directory サーバの情報で置き換えます。

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

- 2 Samba `tdb-tools` パッケージをインストールします。

Samba `tdb-tools` パッケージは、公式の Red Hat リポジトリからダウンロードできません。手動でダウンロードする必要があります。たとえば、次のコマンドを使用して CentOS 7.5 システムからパッケージをダウンロードし、RHEL システムにダウンロードします。

```
yumdownloader tdb-tools
```

CentOS システムがない場合は、<https://rpmfind.net/linux/rpm2html/search.php?query=tdb-tools&submit=Search+...&system=&arch> に移動して `tdb-tools-1.3.15-1.el7.x86_64.rpm` パッケージをダウンロードし、RHEL システムにインストールします。



### 3 Samba と依存関係パッケージをインストールします。

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

### 4 次の例のように、join コマンドを実行します。DNSdomain.example.com は、環境固有の DNS ドメインパスで置き換えます。

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

join コマンドが成功すると、次のメッセージが表示されます。

```
Successfully enrolled machine in realm
```

### 5 システムを再起動して再びログインします。

#### 次のステップ

[RHEL/CentOS デスクトップでの True SSO の設定](#)

## True SSO での Ubuntu デスクトップと Active Directory の統合

Ubuntu 16.04 または 18.04 デスクトップで True SSO をサポートするには、Samba と Winbind のソリューションを使用して、デスクトップと Active Directory ドメインを統合します。

Ubuntu 16.04 または 18.04 デスクトップと Active Directory ドメインを統合するには、次の手順に従います。

説明の中で、Ubuntu デスクトップのホスト名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_IP_ADDRESS	DNS ネーム サーバの IP アドレス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。
myhost	Ubuntu デスクトップのホスト名
MYDOMAIN	ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名。すべて大文字にします。
ads-hostname	Active Directory サーバのホスト名
admin-user	Active Directory ドメインの管理者のユーザー名

#### 前提条件

- Active Directory (AD) サーバが Linux システムの DNS で解決できる。
- Linux システムで NTP (Network Time Protocol) が設定されている。

## 手順

- 1 Ubuntu 16.04 または 18.04 のデスクトップに、**samba** と **winbind** パッケージをインストールします。

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

- 2 プロンプトが表示されたら、次のように Kerberos Authentication の設定を行います。
  - a [Default Kerberos version 5 realm (デフォルトの Kerberos バージョン 5 レalm)] に、Active Directory ドメインの DNS 名をすべて大文字で入力します。  
たとえば、Active Directory ドメインの名前が **mydomain.com** の場合、**MYDOMAIN.COM** と入力します。
  - b [Kerberos servers for your realm (レalmの Kerberos s サーバ)] で、Active Directory サーバのホスト名を入力します (この手順の例では **ads\_hostname**)。
  - c [Administrative server for your Kerberos realm (Kerberos レalmの管理サーバ)] に、Active Directory ドメインのホスト名を再度入力します。

- 3 PAM の設定を更新します。

- a PAM の設定ページを開きます。

```
pam-auth-update
```

- b [Create Home Directory on Login (ログイン時にホーム ディレクトリを作成)] を選択して、[OK] を選択します。

- 4 次の例のように、**/etc/nsswitch.conf** 構成ファイルを編集します。

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

- 5 自動生成された `resolv.conf` ファイルが検索ドメインとして Active Directory ドメインを参照するように、システム接続で NetworkManager の設定を編集します。

- a NetworkManager コントロール パネルを開き、システム接続の [IPv4 Settings (IPv4 設定)] に移動します。方法として、[Automatic (DHCP) addresses only (自動 DHCP アドレスのみ)] を選択します。[DNS servers (DNS サーバ)] テキスト ボックスで、DNS ネーム サーバの IP アドレスを入力します (この手順の例では `dns_IP_ADDRESS` です)。[Save (保存)] をクリックします。
- b `/etc/NetworkManager/system-connections` に構成ファイルでシステム接続を編集します。次の例を使用します。

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

**注:** 新しいインスタント クローン仮想デスクトップを作成すると、新しい仮想ネットワーク アダプタが追加されます。インスタント クローン仮想デスクトップに新しいネットワーク アダプタを追加すると、仮想デスクトップ テンプレート内の DNS サーバなどのネットワーク アダプタのすべての設定が失われます。クローン作成された仮想デスクトップに新しいネットワーク アダプタを追加したときに DNS サーバ設定が失われないようにするには、Linux システムで DNS サーバを指定する必要があります。

- c DNS サーバを指定するには、次の例のように、`/etc/resolv.conf` 構成ファイルを編集します。

```
nameserver dns_IP_ADDRESS

search mydomain.com
```

- d システムを再起動して再びログインします。

- 6 次の例のように、`/etc/hosts` 構成ファイルを編集します。

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

- 7 次の例のように、`/etc/samba/smb.conf` 構成ファイルを編集します。

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

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

- 8 `smbd` サービスを再起動します。

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

- 9 次の例のように、`/etc/krb5.conf` 構成ファイルの内容を編集します。

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

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

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

- 10 Ubuntu デスクトップを Active Directory ドメインに参加させます。

- a Kerberos チケットを開始します。

```
sudo kinit admin-user
```

プロンプトが表示されたら、管理者パスワードを入力します。

- b チケットが正常に作成されたことを確認します。

```
sudo klist
```

このコマンドは、チケットに関する情報（有効期間の開始時間や有効期限）を返します。

- c Kerberos キータブ ファイルを作成します。

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

- d Active Directory ドメインに参加します。

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

## 11 Winbind サービスを再起動して確認します。

- a Winbind サービスを再起動します。

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

- b Winbind サービスを確認するには、次のコマンドを実行し、正しい出力が返されていることを確認します。

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

## 12 システムを再起動して再びログインします。

### 次のステップ

### [Ubuntu デスクトップでの True SSO の設定](#)

## True SSO での SLED/SLES デスクトップと Active Directory の統合

SLED 12.x SP3 または SLES 12.x SP3 デスクトップで True SSO をサポートするには、Samba と Winbind のソリューションを使用して、デスクトップと Active Directory ドメインを統合します。

SLED/SLES デスクトップと Active Directory ドメインを統合するには、次の手順に従います。

### 前提条件

- Active Directory (AD) サーバが Linux システムの DNS で解決できる。
- Linux システムで NTP (Network Time Protocol) が設定されている。

### 手順

- 1 SLED/SLES デスクトップで、`samba` と `winbind` パッケージをインストールします。

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

- 2 YaST 設定ツールを開き、[Network Services (ネットワーク サービス)] - [Windows Domain Membership (Windows ドメインのメンバーシップ)] の順に移動します。
- 3 Windows ドメインのメンバーシップ画面で、次のように設定します。
  - a [Domain or Workgroup (ドメインまたはワークグループ)] で、ワークグループの DNS 名または Samba サーバが含まれている NT ドメインの DNS 名をすべて大文字で入力します。たとえば、ワークグループ名が **mydomain** の場合、**MYDOMAIN** と入力します。
  - b [Also Use SMB Information for Linux Authentication (Linux 認証に SMB 情報も使用)] を選択します。
  - c [Create Home Directory on Login (ログイン時にホーム ディレクトリを作成)] を選択します。

- d [Offline Authentication (オフライン認証)] を選択します。
  - e [Single Sign-on for SSH (SSH でシングル サインオン)] を選択します。
- 4 ドメインへの参加を確認するプロンプトが表示されたら、[Yes (はい)] を選択します。
  - 5 指定されたワークグループの管理者の名前とパスワードを入力し、[OK] を選択します。  
SLED/SLES デスクトップがドメインを正常に参加していることを確認するメッセージが表示されます。[OK] を選択します。
  - 6 次のパラメータが含まれるように、`/etc/samba/smb.conf` 構成ファイルを編集します。

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

- 7 システムを再起動して再びログインします。
  - 8 SLED/SLES デスクトップ統合をテストして確認します。  
次のテスト コマンドを実行して、正しい出力が返されることを確認します。mydomain は、Samba サーバ ワークグループまたは NT ドメインの名前で置き換えます。
- `net ads testjoin`
  - `net ads info`
  - `wbinfo --krb5auth=mydomain\\open%open`
  - `ssh localhost -l mydomain\\open`

#### 次のステップ

#### SLED/SLES デスクトップでの True SSO の設定

## RHEL/CentOS デスクトップでの True SSO の設定

RHEL/CentOS 7.x デスクトップで True SSO 機能を有効にするには、True SSO 機能が依存するライブラリ、スマート カード認証で使用するルート CA 証明書、Horizon Agent をインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

次の手順に従って、RHEL 7.x または CentOS 7.x デスクトップで True SSO を有効にします。これらのデスクトップで True SSO をサポートするには、Horizon Agent 7.6 以降をインストールする必要があります。

説明の中で、Active Directory ドメインの DNS 名などのネットワーク構成のエンティティをプレースホルダーで表している部分があります。次の表を参考にして、これらのプレースホルダーの値をご使用の環境に合わせて変更してください。

プレースホルダーの値	説明
dns_server	DNS ネーム サーバのパス
mydomain.com	Active Directory ドメインの DNS 名
MYDOMAIN.COM	Active Directory ドメインの DNS 名。すべて大文字にします。

## 前提条件

- VMware Identity Manager と Horizon Connection Server に True SSO を設定します。
- [True SSO での RHEL/CentOS デスクトップと Active Directory の統合](#)
- ルート CA 証明書を取得し、RHEL/CentOS 7.x デスクトップの /tmp/certificate.cer に保存します。[ルート CA 証明書をエクスポートする方法](#)を参照してください。

## 手順

- 1 PKCS11 サポート パッケージ グループをインストールします。

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

- 2 ルート CA 証明書をインストールします。

- a ダウンロードしたルート CA 証明書を .pem ファイルに転送します。

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

- b certutil コマンドを使用して、ルート CA 証明書をシステム データベース /etc/pki/nssdb にインストールします。

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

- c RHEL/CentOS 7.x システムで信頼された認証局 (CA) 証明書のリストにルート CA 証明書を追加し、update-ca-trust コマンドを使用して、システム全体のトラストストアの構成を更新します。

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

- 3 次の例のように、ドメインのシステム SSSD 構成ファイルで該当するセクションを変更します。

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

- 4 次の例のように、Kerberos 構成ファイル (/etc/krb5.conf) を編集します。

```
[libdefaults]
dns_lookup_realm = false
ticket_lifetime = 24h
```

```

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

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

```

- 5 Horizon Agent パッケージをインストールして、True SSO を有効にします。

```
sudo ./install-viewagent.sh -T yes
```

**注:** Horizon Agent 7.6 以降をインストールする必要があります。

- 6 次のパラメータを Horizon Agent カスタム構成ファイル (/etc/vmware/viewagent-custom.conf) に追加します。次の例を使用します。NETBIOS\_NAME\_OF\_DOMAIN は、組織のドメインの NetBIOS 名です。

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

- 7 システムを再起動して再びログインします。

## Ubuntu デスクトップでの True SSO の設定

Ubuntu 16.04 または 18.04 デスクトップで True SSO 機能を有効にするには、True SSO 機能が依存するライブラリ、スマートカード認証で使用するルート CA 証明書、Horizon Agent をインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

次の手順に従って、Ubuntu 16.04 または 18.04 デスクトップで True SSO を有効にします。これらのデスクトップで True SSO をサポートするには、Horizon Agent 7.8 以降をインストールする必要があります。

### 前提条件

- VMware Identity Manager と Horizon Connection Server に True SSO を設定します。
- [True SSO での Ubuntu デスクトップと Active Directory の統合](#)
- ルート CA 証明書を取得し、デスクトップの /tmp/certificate.cer に保存します。[ルート CA 証明書をエクスポートする方法](#)を参照してください。



## 手順

- 1 Ubuntu 16.04 または 18.04 のデスクトップで、pkcs11 サポート パッケージをインストールします。

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

- 2 libnss3-tools パッケージをインストールします。

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

- 3 ルート CA 証明書をインストールします。

- a ダウンロードしたルート CA 証明書を .pem ファイルに転送します。

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

- b certutil コマンドを使用して、ルート CA 証明書をシステム データベース /etc/pki/nssdb にインストールします。

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

- c ルート CA 証明書を /etc/pam\_pkcs11/cacerts にディレクトリにコピーします。

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

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

- d ルート CA 証明書のハッシュ リンクを作成します。/etc/pam\_pkcs11/cacerts ディレクトリで、次のコマンドを実行します。

```
pkcs11_make_hash_link
```

- 4 Horizon Agent パッケージをインストールして、True SSO を有効にします。

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

---

**注:** True SSO 機能を使用するには、Horizon Agent 7.8 以降をインストールする必要があります。

---

- 5 次のパラメータを Horizon Agent カスタム構成ファイル (/etc/vmware/viewagent-custom.conf) に追加します。次の例を使用します。NETBIOS\_NAME\_OF\_DOMAIN は、組織のドメインの NetBIOS 名です。

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

## 6 /etc/pam\_pkcs11/pam\_pkcs11.conf 構成ファイルを編集します。

- a 必要であれば、/etc/pam\_pkcs11/pam\_pkcs11.conf 構成ファイルを作成します。/usr/share/doc/libpam-pkcs11/examples でサンプル ファイルを探し、このファイルを /etc/pam\_pkcs11 ディレクトリにコピーして、ファイル名を pam\_pkcs11.conf に変更します。必要であれば、ファイルのコンテンツにシステム情報を追加します。
- b 次の例に似たコンテンツが含まれるように、/etc/pam\_pkcs11/pam\_pkcs11.conf 構成ファイルを編集します。

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

## 7 PAM 構成ファイルで auth パラメータを変更します。

- a PAM 構成ファイルを開きます。
  - Ubuntu 16.04 の場合は、/etc/pam.d/lightdm を開きます。
  - Ubuntu 18.04 の場合は、/etc/pam.d/gdm-vmwcred を開きます。
- b 次の例のように、PAM 構成ファイルを編集します。

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

## 8 システムを再起動して再びログインします。

# SLED/SLES デスクトップでの True SSO の設定

SLED/SLES 12.x SP3 デスクトップで True SSO 機能を有効にするには、True SSO 機能が依存するライブラリ、スマートカード認証で使用するルート CA 証明書、Horizon Agent をインストールします。また、一部の構成ファイルを編集して、認証設定を完了する必要があります。

次の手順に従って、SLED 12.x SP3 または SLES 12.x SP3 デスクトップで True SSO を有効にします。これらのデスクトップで True SSO をサポートするには、Horizon Agent 7.8 以降をインストールする必要があります。

### 前提条件

- VMware Identity Manager と Horizon Connection Server に True SSO を設定します。
- [True SSO での SLED/SLES デスクトップと Active Directory の統合](#)
- ルート CA 証明書を取得し、SLED/SLES 12.x SP3 デスクトップの /tmp/certificate.cer に保存します。  
[ルート CA 証明書をエクスポートする方法](#)を参照してください。

## 手順

- 1 SLES 12.x SP3 デスクトップの場合は、次のコマンドを実行して、必要なパッケージをインストールします。

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

- 2 SLED 12.x SP3 デスクトップの場合は、次の手順に従って、必要なパッケージをインストールします。

- a SLES .iso ファイルをダウンロードして、SLED デスクトップのローカル ディスクに保存します（例：/tmp/SLE-12-SP3-Server-DVD-x86\_64-GM-DVD1.iso）。

必要な krb5-plugin-preauth-pkinit パッケージは SLES システムでのみ使用可能です。このため、SLES .iso ファイルを SLED デスクトップのパッケージ ソースとして追加する必要があります。

- b SLED デスクトップに SLES .iso ファイルをマウントし、必要なパッケージをインストールします。

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

- c インストールが完了したら、SLES .iso ファイルのマウントを解除します。

```
sudo umount /mnt/sles
```

- 3 ルート CA 証明書をインストールします。

- a ダウンロードしたルート CA 証明書を .pem ファイルに転送します。

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

- b certutil コマンドを使用して、ルート CA 証明書をシステム データベース /etc/pki/nssdb にインストールします。

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

- c ルート CA 証明書を pam\_pkcs11 に追加します。

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

- 4 次の例のように、/etc/krb5.conf 構成ファイルの内容を編集します。

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

[realms]
    MYDOMAIN.COM = {
```

```

    kdc = ads-hostname
    admin_server = ads-hostname
    pkinit_anchors = DIR:/etc/pki/ca-trust/source/anchors
    pkinit_kdc_hostname = ads-hostname
    pkinit_eku_checking = kpServerAuth
}

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

```

次の表を参考にして、サンプルのプレースホルダーの値をご使用のネットワーク環境に合わせて変更してください。

プレースホルダーの値	説明
<i>mydomain.com</i>	Active Directory ドメインの DNS 名
<i>MYDOMAIN.COM</i>	Active Directory ドメインの DNS 名。すべて大文字にします。
<i>ads-hostname</i>	Active Directory サーバのホスト名（大文字と小文字を区別）

- Horizon Agent パッケージをインストールして、True SSO を有効にします。

```
sudo ./install-viewagent.sh -T yes
```

**注:** True SSO 機能を使用するには、Horizon Agent 7.8 以降をインストールする必要があります。

- 次のパラメータを Horizon Agent カスタム構成ファイル (*/etc/vmware/viewagent-custom.conf*) に追加します。次の例を使用します。*NETBIOS\_NAME\_OF\_DOMAIN*は、組織のドメインの NetBIOS 名です。

```
NetbiosDomain=NETBIOS_NAME_OF_DOMAIN
```

- システムを再起動して再びログインします。

# Linux デスクトップのグラフィックス のセットアップ

# 4

ESXi ホストまたはゲスト OS で NVIDIA 機能を活用するように、現在サポートされている RHEL ディストリビューションを構成できます。

[3D グラフィックスのセットアップのための仮想マシンのクローン作成要件]

3D グラフィックスをセットアップする前に、仮想マシンのクローン作成に関する次の要件を考慮する必要があります。

- vGPU および vSGA については、基本仮想マシンのグラフィックスのセットアップを完了させます。仮想マシンのクローンを作成します。グラフィックス設定はクローン作成された仮想マシンについて動作し、さらなる設定は必要ありません。
- vDGA については、基本仮想マシンのグラフィックスのセットアップを完了させます。仮想マシンのクローンを作成します。ただし、クローン作成された仮想マシンをパワーオンする前に、クローン作成された仮想マシンから既存の NVIDIA パススルー PCI デバイスを削除し、クローン作成された仮想マシンに新しい NVIDIA パススルー PCI デバイスを追加する必要があります。NVIDIA パススルー PCI デバイスは、仮想マシン間で共有することはできません。各仮想マシンは、専用の NVIDIA パススルー PCI デバイスを使用します。

この章には、次のトピックが含まれています。

- サポート対象の RHEL/SLED ディストリビューションを vGPU 用に構成する
- vDGA を使用するための RHEL 6 の構成
- vSGA を使用するための RHEL 7.x の構成

## サポート対象の RHEL/SLED ディストリビューションを vGPU 用に構成する

ESXi ホストで NVIDIA vGPU（共有 GPU ハードウェア アクセラレーション）機能を利用するように、サポート対象の RHEL または SLED ディストリビューションをセットアップできます。

ESXi ホスト GPU ドライバ (.vib) と一致する NVIDIA Linux 仮想マシン ディスプレイ ドライバを使用する必要があります。ドライバ パッケージに関する情報は、NVIDIA の Web サイトを参照してください。

**重要:** NVIDIA vGPU は、NVIDIA Tesla M6、M10、M60、P40 のグラフィック カードでサポートされます。この機能は、GRID K1 や K2 などの他の NVIDIA グラフィック カードでは動作しません。

**注意:** 開始する前に、Horizon Agent が Linux 仮想マシンにインストールされていないことを確認します。NVIDIA vGPU を使用するようにマシンを構成する前に Horizon Agent をインストールすると、`xorg.conf` ファイルで必須の構成パラメータが上書きされ、NVIDIA vGPU は動作しません。NVIDIA vGPU の構成が完了した後に、Horizon Agent をインストールする必要があります。

## NVIDIA GRID vGPU グラフィック カードの VIB の ESXi ホストへのインストール

ESXi 6.0 U1 以降のホストに NVIDIA GRID グラフィック カードの VIB をダウンロードしてインストールする必要があります。

NVIDIA から vGPU Manager を含む vGPU ソフトウェア パッケージと Linux ディスプレイ ドライバが提供されます。vGPU ソフトウェア パッケージは、この手順で ESXi ホストにインストールし、Linux ディスプレイ ドライバは、この後の手順で Linux 仮想マシンにインストールします。

### 前提条件

- vSphere 6.0 U1 以降のリリースが環境にインストールされていることを確認します。
- NVIDIA Tesla M6、M10、M60 または P40 GPU が ESXi ホストにインストールされていることを確認します。

### 手順

- 1 [NVIDIA のドライバ ダウンロード](#) サイトから NVIDIA GRID vGPU グラフィック カードの VIB をダウンロードします。

適切な VIB バージョンをドロップダウン メニューから選択します。

オプション	説明
製品タイプ	[GRID]
製品シリーズ	[NVIDIA GRID vGPU] を選択します。
製品	ESXi ホストにインストールされるバージョン ([GRID K2] など) を選択します。
オペレーティング システム	VMware vSphere ESXi のバージョンを選択します。

- 2 vGPU ソフトウェア パッケージの .zip ファイルを解凍します。
- 3 vGPU Manager フォルダを ESXi ホストにアップロードします。

**注:** この後の手順で、Linux ディスプレイ ドライバを Linux 仮想マシンにインストールします。

- 4 ESXi ホスト上のすべての仮想マシンをパワーオフまたはサスペンドします。
- 5 SSH を使用して ESXi ホストに接続します。

## 6 xorg サービスを停止します。

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

## 7 NVIDIA VIB をインストールします。

例：

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

## 8 ESXi ホストを再起動または更新します。

- ◆ インストール済みの ESXi ホストでは、ホストを再起動します。
- ◆ ステートレス ESXi ホストでは、次の手順を実行し、ホストを更新します（これらの手順は、インストール済みのホストにも適用できます）。

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

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

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

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

## 9 ホストを再起動した後に、xorg サービスが実行されていることを確認します。

# Linux 仮想マシンで vGPU を使用するための共有 PCI デバイスの構成

NVIDIA vGPU を使用するには、Linux 仮想マシン用に共有 PCI デバイスを構成する必要があります。

### 前提条件

- Linux 仮想マシンをデスクトップとして使用する準備ができていることを確認します。[仮想マシンを作成して、Linux をインストールする](#)および[リモート デスクトップ デプロイ用の Linux マシンの準備](#)を参照してください。
- Horizon Agent が Linux 仮想マシンにインストールされていないことを確認します。
- NVIDIA VIB が ESXi ホストにインストールされていることを確認します。[NVIDIA GRID vGPU グラフィック カードの VIB の ESXi ホストへのインストール](#)を参照してください。
- NVIDIA vGPU で利用可能な仮想 GPU タイプについて理解しておきます。GPU のタイプは、[GPU プロファイル] 設定で選択します。仮想 GPU タイプは、ESXi ホストにインストールされた物理 GPU でさまざまな機能を提供します。[NVIDIA 仮想 GPU タイプ](#)を参照してください。

## 手順

- 1 仮想マシンをパワーオフします。
- 2 vSphere Web Client で、[仮想マシン ハードウェア] タブで仮想マシンを選択して、[設定編集] をクリックします。
- 3 [新規デバイス] メニューで、[共有 PCI デバイス] を選択します。
- 4 [追加] をクリックして、ドロップダウン メニューから [NVIDIA GRID vGPU] を選択します。
- 5 [GPU プロファイル] 設定で、ドロップダウン メニューから仮想 GPU タイプを選択します。
- 6 [すべてのメモリを予約] をクリックして、[OK] をクリックします。

GPU が NVIDIA GRID vGPU をサポートできるようにするには、すべての仮想マシンのメモリを予約する必要があります。

- 7 仮想マシンをパワーオンします。

## NVIDIA 仮想 GPU タイプ

vSphere Web Client の [仮想ハードウェア] ページにある [GPU プロファイル] 設定で、ESXi ホストの物理 NVIDIA GPU で特別な機能を利用できるようにする仮想 GPU タイプを選択できます。

Linux 仮想マシンでは、NVIDIA Tesla M6、M10、M60 または P40 GPU で NVIDIA GRID vGPU がサポートされています。

表 4-1. Linux 仮想マシンの NVIDIA GRID M60 vGPU で利用可能な仮想 GPU タイプ

仮想 GPU タイプ	物理ボード	物理 GPU の数	仮想 GPU あたりの FB	ディスプレイ数	最大解像度	物理 GPU あたりの最大仮想 GPU 数	物理ボードあたりの最大仮想 GPU 数
GRID M60-1q	Tesla M60	2	1G	2	2560x1600	8	16
GRID M60-2q	Tesla M60	2	2G	4	2560x1600	4	8
GRID M60-4q	Tesla M60	2	4G	4	3840x2160	2	4
GRID M60-8q	Tesla M60	2	8G	4	3840x2160	1	2

表 4-2. Linux 仮想マシンの NVIDIA GRID M6 vGPU で利用可能な仮想 GPU タイプ

仮想 GPU タイプ	物理ボード	物理 GPU の数	仮想 GPU あたりの FB	ディスプレイ数	最大解像度	物理 GPU あたりの最大仮想 GPU 数	物理ボードあたりの最大仮想 GPU 数
GRID M6-1q	Tesla M6	1	1G	2	2560x1600	8	8
GRID M6-2q	Tesla M6	1	2G	4	2560x1600	4	4
GRID M6-4q	Tesla M6	1	4G	4	3840x2160	2	2
GRID M6-8q	Tesla M6	1	8G	4	3840x2160	1	1



表 4-3. Linux 仮想マシンの NVIDIA GRID M10 vGPU で利用可能な仮想 GPU タイプ

仮想 GPU タイプ	物理ボード	物理 GPU の数	仮想 GPU あたりの FB	ディスプレイ数	最大解像度	物理 GPU あたりの最大仮想 GPU 数	物理ボードあたりの最大仮想 GPU 数
GRID M10-1q	Tesla M10	4	1G	2	4096x2160	8	16
GRID M10-2q	Tesla M10	4	2G	4	4096x2160	4	8
GRID M10-4q	Tesla M10	4	4G	4	4096x2160	2	4
GRID M10-8q	Tesla M10	4	8G	4	4096x2160	1	2

表 4-4. Linux 仮想マシンの NVIDIA GRID P40 vGPU で利用可能な仮想 GPU タイプ

仮想 GPU タイプ	物理ボード	物理 GPU の数	仮想 GPU あたりの FB	ディスプレイ数	最大解像度	物理 GPU あたりの最大仮想 GPU 数	物理ボードあたりの最大仮想 GPU 数
GRID P40-1q	Tesla P40	1	1G	2	4096x2160	24	24
GRID P40-2q	Tesla P40	1	2G	4	4096x2160	12	12
GRID P40-3q	Tesla P40	1	3G	4	4096x2160	8	8
GRID P40-4q	Tesla P40	1	4G	4	4096x2160	6	6
GRID P40-6q	Tesla P40	1	6G	4	5120x2880	4	4
GRID P40-8q	Tesla P40	1	8G	4	5120x2880	3	3
GRID P40-12q	Tesla P40	1	12G	4	5120x2880	2	4
GRID P40-24q	Tesla P40	1	24G	4	5120x2880	1	2

## NVIDIA GRID vGPU ディスプレイ ドライバのインストール

NVIDIA GRID vGPU ディスプレイ ドライバをインストールするには、デフォルトの NVIDIA ドライバを無効にし、NVIDIA ディスプレイ ドライバをダウンロードして、仮想マシンで PCI デバイスを構成する必要があります。

### 前提条件

- NVIDIA のダウンロード サイトから vGPU ソフトウェア パッケージをダウンロードして解凍しており、Linux ディスプレイ ドライバ (パッケージ コンポーネント) の準備ができていることを確認します。 [NVIDIA GRID vGPU グラフィック カードの VIB の ESXi ホストへのインストール](#) を参照してください。

また、共有 PCI デバイスが仮想マシンに追加されていることを確認します。 [Linux 仮想マシンで vGPU を使用するための共有 PCI デバイスの構成](#) を参照してください。

### 手順

- 1 NVIDIA Linux ディスプレイ ドライバを仮想マシンにコピーします。
- 2 仮想マシンへのリモート ターミナルを開くか、Ctrl + Alt + F2 キーを押してテキスト コンソールに切り替えて、root としてログインして、init 3 コマンドを実行して X Windows を無効にします。

- 3 NVIDIA ドライバで必要となる追加のコンポーネントをインストールします。

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

- 4 NVIDIA GRID vGPU ドライバ パッケージに実行可能なフラグを追加します。

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

- 5 NVIDIA GRID vGPU インストーラを起動します。

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

- 6 NVIDIA のソフトウェア使用許諾契約書に同意して、[Yes] を選択して、X の設定を自動的に更新します。

#### 次のステップ

Linux 仮想マシンに Horizon Agent をインストールします。 [Linux 仮想マシンへの Horizon Agent のインストール](#) を参照してください。

構成した Linux 仮想マシンを含むデスクトップ プールを作成します。 [Linux 版手動デスクトップ プールの作成](#) を参照してください。

## NVIDIA ディスプレイ ドライバがインストールされているかどうかの確認

Horizon デスクトップ セッションに NVIDIA ドライバの出力を表示して、NVIDIA ディスプレイ ドライバが Linux 仮想マシンにインストールされていることを確認できます。

#### 前提条件

- NVIDIA ディスプレイ ドライバをインストールしていることを確認します。
- Horizon Agent が Linux 仮想マシンにインストールされていることを確認します。 [Linux 仮想マシンへの Horizon Agent のインストール](#) を参照してください。
- Linux 仮想マシンがデスクトップ プールにデプロイされていることを確認します。 [Linux 版手動デスクトップ プールの作成](#) を参照してください。

#### 手順

- 1 Linux 仮想マシンを再起動します。

Horizon Agent 起動スクリプトは、X サーバを初期化し、トポロジを表示します。

vSphere コンソールで、仮想マシンの表示を参照することはできなくなります。

- 2 Horizon Client で、Linux デスクトップに接続します。
- 3 Linux デスクトップセッションで、NVIDIA ディスプレイ ドライバがインストールされていることを確認します。

ターミナル ウィンドウを開き、`glxinfo | grep NVIDIA` コマンドを実行します。

NVIDIA ドライバ出力が表示されます。例：

```
[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
```

ユーザーは、リモート デスクトップで NVIDIA グラフィックスの機能にアクセスできます。

NVIDIA ディスプレイ ドライバのインストールを確認した後、インストールが正しく動作するために、次のタスクを実行します。

- Linux カーネルをアップグレードする場合、Horizon Agent が Horizon Connection Server と通信できないことがあります。この問題を解決するには、NVIDIA ドライバを再インストールします。
- Linux 仮想マシンで NVIDIA GRID のライセンスを設定します。詳細については、NVIDIA のドキュメントを参照してください。ライセンスが設定されていない場合、Linux デスクトップは正しく動作しません。たとえば、自動的に合わせる機能が動作しません。

## vDGA を使用するための RHEL 6 の構成

Horizon 7 for Linux デスクトップが ESXi ホストで vDGA 機能を利用できるように、RHEL 6 ゲスト OS をセットアップできます。

**注意:** 開始する前に、Horizon Agent が Linux 仮想マシンにインストールされていないことを確認します。vDGA を使用するようにマシンを構成する前に Horizon Agent をインストールすると、`xorg.conf` ファイルで必須の構成パラメータが上書きされ、vDGA は動作しません。vDGA の構成が完了した後に、Horizon Agent をインストールする必要があります。

## ホストで NVIDIA GRID を使用するために DirectPath I/O を有効にする

Linux 仮想マシンを構成して vDGA を使用できるようにするには、NVIDIA GRID GPU PCI デバイスを ESXi ホストの DirectPath I/O パススルーで利用できるようにする必要があります。

### 前提条件

- vSphere 6.0 または以降のリリースが環境にインストールされていることを確認します。
- NVIDIA GRID K1 または K2 グラフィック カードが ESXi ホストにインストールされていることを確認します。

### 手順

- 1 vSphere Web Client で、ESXi ホストを参照します。
- 2 [管理] タブをクリックして、[設定] をクリックします。
- 3 [ハードウェア] セクションの [PCI デバイス] をクリックします。

- 4 NVIDIA GRID GPU で DirectPath I/O パススルーを有効にするには、[編集] をクリックします。

アイコン	説明
緑色のアイコン	PCI デバイスはアクティブで、有効にできます。
オレンジ色のアイコン	デバイスの状態が変更されました。デバイスを使用する前にホストを再起動する必要があります。

- 5 NVIDIA GRID GPU を選択して、[OK] をクリックします。

PCI デバイスが表に追加され、仮想マシンで DirectPath I/O PCI デバイスを利用できるようになります。

- 6 ホストを再起動して、Linux 仮想マシンが PCI デバイスを利用できるようにします。

## vDGA パススルー デバイスの RHEL 6 仮想マシンへの追加

vDGA を使用するように RHEL 6 仮想マシンを構成するには、PCI デバイスを仮想マシンに追加する必要があります。この手順によって、ESXi ホストの物理デバイスをパススルーして仮想マシンで使用できるようになります。

### 前提条件

- Linux 仮想マシンをデスクトップとして使用する準備ができていることを確認します。「[仮想マシンを作成して、Linux をインストールする](#)および [リモート デスクトップ デプロイ用の Linux マシンの準備](#)」を参照してください。
- Horizon Agent が Linux 仮想マシンにインストールされていないことを確認します。
- NVIDIA GRID GPU PCI デバイスがホストの DirectPath I/O パススルーで利用可能になっていたか確認します。[ホストで NVIDIA GRID を使用するために DirectPath I/O を有効にする](#)を参照してください。

### 手順

- 1 sudo 権限で構成されたローカル ユーザーとして、RHEL 6 ゲスト OS にログインします。
- 2 vSphere Web Client で、[仮想マシン ハードウェア] タブで仮想マシンを選択して、[設定編集] をクリックします。
- 3 [新規デバイス] メニューで、[PCI デバイス] を選択します。
- 4 [追加] をクリックして、ドロップダウン メニューから PCI デバイスを選択します。
- 5 [すべてのメモリを予約] をクリックして、[OK] をクリックします。  
GPU が vDGA をサポートできるようにするには、すべての仮想マシンのメモリを予約する必要があります。
- 6 仮想マシンをパワーオンして、vSphere コンソールを開いてマシンに接続します。

## 7 NVIDIA GRID デバイスが仮想マシンにパススルーされていることを確認します。

ターミナル ウィンドウを開き、次のコマンドを実行します。

```
lspci | grep NVIDIA
```

XX:00.0 VGA 互換のコントローラが表示されます。例：

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

## vDGA 用の NVIDIA ディスプレイ ドライバのインストール

vDGA 用の NVIDIA ディスプレイ ドライバをインストールするには、デフォルトの NVIDIA ドライバを無効にし、NVIDIA ディスプレイ ドライバをダウンロードして、仮想マシンで PCI デバイスを構成する必要があります。

### 前提条件

- PCI デバイスが RHEL 6.x 仮想マシンに追加されていることを確認します。[vDGA パススルー デバイスの RHEL 6 仮想マシンへの追加](#)を参照してください。

### 手順

#### 1 デフォルトの NVIDIA Nouveau ドライバを無効にしてブラックリストに入れます。

- a grub.conf ファイルを編集します。

RHEL 6 の場合のファイルは /boot/grub/grub.conf です。

RHEL バージョン	コマンド
6	<code>sudo vi /boot/grub/grub.conf</code>

- b rdblacklist=nouveau 行をカーネル オプションの最後に追加します。
- c blacklist.conf ファイルを編集します。

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

- d blacklist.conf ファイルの任意の場所に次の行を追加します。

```
blacklist nouveau
```

#### 2 仮想マシンを再起動します。

表示のルック アンド フィールドが変更されます。

#### 3 (オプション) Nouveau ドライバが無効になっていることを確認します。

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

grep 検索によって何も結果が返されない場合、Nouveau ドライバは無効になっています。

- 4 [NVIDIA のドライバ ダウンロード](#) サイトから NVIDIA のドライバをダウンロードします。

適切なドライバ バージョンを NVIDIA のドロップダウン メニューから選択します。

オプション	説明
製品タイプ	[GRID]
製品シリーズ	[GRID シリーズ]
製品	ESXi ホストにインストールされるバージョン ([GRID K2] など) を選択します。
オペレーティング システム	Linux 64 ビットまたは Linux 32 ビット

- 5 仮想マシンに接続するには、リモート ターミナルを開くか、Ctrl + Alt + F2 キーを押してテキスト コンソールに切り替え、root としてログインして `init 3` コマンドを実行し、X Windows を無効にします。
- 6 NVIDIA ドライバで必要となる追加のコンポーネントをインストールします。

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

- 7 vDGA 用の NVIDIA ドライバ パッケージに実行可能なフラグを追加します。

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

- 8 NVIDIA インストーラを実行します。

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

- 9 NVIDIA のソフトウェア使用許諾契約書に同意して、[Yes] を選択して、X の設定を更新します。

#### 次のステップ

Linux 仮想マシンに Horizon Agent をインストールします。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。

構成した Linux 仮想マシンを含むデスクトップ プールを作成します。[Linux 版手動デスクトップ プールの作成](#)を参照してください。

## NVIDIA ディスプレイ ドライバがインストールされているかどうかの確認

Horizon デスクトップ セッションに NVIDIA ドライバの出力を表示して、NVIDIA ディスプレイ ドライバが Linux 仮想マシンにインストールされていることを確認できます。

#### 前提条件

- NVIDIA ディスプレイ ドライバをインストールしていることを確認します。
- Horizon Agent が Linux 仮想マシンにインストールされていることを確認します。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- Linux 仮想マシンがデスクトップ プールにデプロイされていることを確認します。[Linux 版手動デスクトップ プールの作成](#)を参照してください。

## 手順

- 1 Linux 仮想マシンを再起動します。

Horizon Agent 起動スクリプトは、X サーバを初期化し、トポロジを表示します。

vSphere コンソールで、仮想マシンの表示を参照することはできなくなります。

- 2 Horizon Client で、Linux デスクトップに接続します。

- 3 Linux デスクトップセッションで、NVIDIA ディスプレイ ドライバがインストールされていることを確認します。

ターミナル ウィンドウを開き、`glxinfo | grep NVIDIA` コマンドを実行します。

NVIDIA ドライバ出力が表示されます。例：

```
[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
```

ユーザーは、リモート デスクトップで NVIDIA グラフィックスの機能にアクセスできます。

NVIDIA ディスプレイ ドライバのインストールを確認した後、インストールが正しく動作するために、次のタスクを実行します。

- Linux カーネルをアップグレードする場合、Horizon Agent が Horizon Connection Server と通信できないことがあります。この問題を解決するには、NVIDIA ドライバを再インストールします。
- Linux 仮想マシンで NVIDIA GRID のライセンスを設定します。詳細については、NVIDIA のドキュメントを参照してください。ライセンスが設定されていない場合、Linux デスクトップは正しく動作しません。たとえば、自動的に合わせる機能が動作しません。

## vSGA を使用するための RHEL 7.x の構成

Horizon 7 for Linux デスクトップが vSGA 機能を利用できるように、RHEL 7.x ゲスト OS をセットアップできます。

### vSGA 用の NVIDIA グラフィック カードの VIB の ESXi ホストへのインストール

ESXi 6.0 U1 以降のホストに NVIDIA GRID グラフィック カードの VIB をダウンロードしてインストールする必要があります。

NVIDIA は vSGA 用の VMware vSphere ESXi ドライバを提供しています。vSGA の場合、NVIDIA ディスプレイ ドライバは、Linux 仮想マシンにインストールされません。

#### 前提条件

- vSphere 6.0 U1 以降のリリースが環境にインストールされていることを確認します。
- NVIDIA ドライバが環境にインストールされていることを確認します。

- NVIDIA GRID K1 または K2 GPU が ESXi ホストにインストールされていることを確認します。

#### 手順

- 1 [NVIDIA のドライバ ダウンロード](#) サイトから NVIDIA GRID vGPU グラフィック カードの VIB をダウンロードします。

適切な VIB バージョンをドロップダウン メニューから選択します。

オプション	説明
製品タイプ	[GRID]
製品シリーズ	[GRID シリーズ] を選択します。
製品	ESXi ホストにインストールされるバージョン ([GRID K2] など) を選択します。
オペレーティング システム	VMware vSphere ESXi のバージョンを選択します。

- 2 vSGA 用の VMware vSphere ESXi ドライバを ESXi 6.0 U1 ホストにアップロードします。
- 3 ESXi ホスト上のすべての仮想マシンをパワーオフまたはサスペンドします。
- 4 SSH を使用して ESXi ホストに接続します。
- 5 xorg サービスを停止します。

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

- 6 NVIDIA VIB をインストールします。

例 :

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

- 7 GPU の割り当てに使用される xorg を再起動します。

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

- 8 ESXi ホストを再起動します。
- 9 ホストを再起動した後に、xorg サービスが実行されていることを確認します。

## Linux 仮想マシンでの vSGA の 3D 機能の構成

RHEL 7 仮想マシンで vSGA を使用するように構成するには、vSphere Web Client において仮想マシンのビデオ カードで 3D 設定を構成する必要があります。

#### 前提条件

- Linux 仮想マシンをデスクトップとして使用する準備ができており、Horizon Agent がインストールされ、マシンがデスクトップ プールにデプロイされていることを確認します。



- NVIDIA VIB が ESXi ホストにインストールされていることを確認します。[vSGA 用の NVIDIA グラフィック カードの VIB の ESXi ホストへのインストール](#)を参照してください。

#### 手順

- 1 仮想マシンをパワーオフします。
- 2 vSphere Web Client で、[仮想マシン ハードウェア] タブで仮想マシンを選択して、[設定編集] をクリックします。
- 3 [仮想ハードウェア] タブで、[ビデオ カード] をクリックして、メニュー設定を展開します。
- 4 [ビデオ メモリの合計] を 128 MB に設定します。
- 5 [3D グラフィックス] には、[3D サポートを有効化] を選択します。
- 6 [3D レンダラ] には、ドロップダウン メニューから [ハードウェア] を選択します。
- 7 [3D メモリ] には、アプリケーションの要件に合った値を選択します。  
ユーザーが 3 台以上のモニターに接続する場合、この値は少なくとも 1024 MB に設定します。
- 8 [OK] をクリックします。
- 9 仮想マシンをパワーオンします。

#### 次のステップ

vSGA が Linux 仮想マシンで実行されていることを確認します。

次に、Linux 仮想マシンに Horizon Agent をインストールします。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。

## vSGA が Linux 仮想マシンで実行されているかどうかの確認

仮想マシンのログ ファイルとゲスト OS を確認して、RHEL 7 仮想マシンで vSGA が実行されていることを確認できます。

#### 手順

- 1 仮想マシンの `vmware.log` ファイルを開きます。

サポート対象の GPU および NVIDIA VIB が正しくインストールされている場合、ログ ファイルに次の例のような行が表示されます。

```
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"
```

サポート対象の GPU および NVIDIA VIB が正しくインストールされていない場合、仮想マシンはソフトウェアレンダラーを使用します。vmware.log ファイルには、次の例のような行が表示されます。

```
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 仮想マシンのゲスト OS で、次のコマンドを入力します。

```
glxinfo|grep Gallium
```

vSGA が動作している場合、コマンドは次のテキストを返します。

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

vSGA が正しく動作していない場合、コマンドは次のテキストを返します。

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

#### 次のステップ

Linux 仮想マシンに Horizon Agent をインストールします。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。

# Horizon Agent のインストール

Horizon Connection Server がデスクトップと通信して管理できるように、Linux デスクトップに Horizon Agent をインストールする必要があります。

この章には、次のトピックが含まれています。

- [Linux 仮想マシンへの Horizon Agent のインストール](#)
- [Linux Agent 用証明書の構成](#)
- [Linux 仮想マシンでの Horizon Agent のアップグレード](#)
- [Horizon 7 for Linux マシンをアンインストール](#)

## Linux 仮想マシンへの Horizon Agent のインストール

Linux 仮想マシンをリモート デスクトップとして展開できるようにするには、Horizon Agent を Linux 仮想マシンにインストールする必要があります。

Horizon 7.0.1 のリリースから、Horizon Agent for Linux は vCenter Server の管理対象仮想マシンを使用します。管理対象仮想マシンは、次の機能強化を提供します。

- vCenter Server は、Linux デスクトップ デプロイ環境では必須要件です。
- Linux での Horizon Agent のインストールには、登録は必要ありません。
- ほとんどの Linux デスクトップ デプロイ環境では、ベース仮想マシンに Horizon Agent をインストールできます。

---

**注意:** NVIDIA GRID vGPU、vDGA、または vSGA を使用する場合には、Horizon Agent をインストールする前に、Linux 仮想マシンでこれらの 3D 機能を構成する必要があります。Horizon Agent を最初にインストールしてしまうと、`xorg.conf` ファイルの必須パラメータが上書きされ、3D グラフィックス機能が動作しません。

サポート対象の RHEL/SLED ディストリビューションを vGPU 用に構成する、vDGA を使用するための RHEL 6 の構成、または vSGA を使用するための RHEL 7.x の構成を参照してください。3D グラフィックスの構成が完了したら、Horizon Agent をインストールします。

2D グラフィックスを構成する場合は、[リモート デスクトップ デプロイ用の Linux マシンの準備](#)の手順を完了した後、Horizon Agent をインストールできます。

---

## 前提条件

- Linux ゲスト OS がデスクトップとして使用できるように準備されていることを確認します。 [リモート デスクトップ デプロイ用の Linux マシンの準備](#)を参照してください。
- Linux 用の Horizon Agent インストーラ スクリプトについて理解しておきます。 [install\\_viewagent.sh コマンドライン オプション](#)を参照してください。

## 手順

- 1 VMware ダウンロード サイト (<https://my.vmware.com/web/vmware/downloads>) から、Horizon Agent for Linux インストーラ ファイルをダウンロードします。

[Desktop & End-User Computing] (デスクトップおよびエンドユーザー コンピューティング) セクションで VMware Horizon の View ダウンロード コンポーネントを選択します。Horizon 7 for Linux で、64 ビット Linux システム用の VMware Horizon 7 のダウンロード ページを選択します。

インストーラ ファイル名は、64 ビット Linux で `VMware-horizonagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz` です。y.y.y はバージョン番号、xxxxxxx はビルド番号です。

- 2 お使いの Linux ディストリビューションの tar ボールをゲスト OS に展開します。

例：

```
tar -xzf VMware-horizonagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz
```

- 3 tar ボール フォルダに移動します。
- 4 `install_viewagent.sh` スクリプトをスーパーユーザーとして実行します。

コマンドライン オプションのリストについては、 [install\\_viewagent.sh コマンドライン オプション](#)を参照してください。

例：

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

- 5 `-A` オプションを指定せずに `install_viewagent.sh` を実行する場合、**Yes** と入力して EULA に同意します。  
EULA に同意しない限りインストーラは実行されません。

- 6 Linux を再起動して変更を有効にします。

インストール後に、`viewagent` サービスは開始されます。サービスが、`sudo service viewagent status` を使用して開始されたことを確認します。

## 次のステップ

デスクトップ プールに仮想マシンをデプロイします。 [Linux 版手動デスクトップ プールの作成](#)を参照してください。

## install\_viewagent.sh コマンドライン オプション

`install_viewagent.sh` スクリプトは、Horizon Agent を Linux ゲスト OS にインストールします。

gnome デスクトップ環境のコマンド ウィンドウで、次の形式の `install_viewagent.sh` スクリプトを使用します。

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

`install_viewagent.sh` スクリプトには、必須およびオプションのパラメータが含まれます。

表 5-1. `install_viewagent.sh` 必須のオプション パラメータ

オプション パラメータ (必須情報)	説明
-A yes no	エンドユーザー使用許諾契約書 (EULA) と FIPS (Federal Information Processing Standards) の記載内容に同意するか、拒否します。インストールを続行するには、 <b>yes</b> を指定する必要があります。

表 5-2. `install_viewagent.sh` のオプション パラメータ

オプション パラメータ	説明
-a yes no	オーディオ入力ダイレクト サポートをインストールするかバイパスします。デフォルトは、yes です。
-f yes no	FIPS (Federal Information Processing Standards) 140-2 準拠の暗号モジュールのサポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。詳細については、 <a href="#">Horizon Linux デスクトップの機能</a> で FIPS 140-2 モードの説明を参照してください。
-j	JMS SSL キーストア パスワード。デフォルトでは、インストーラは任意の文列を生成します。
-m yes no	スマート カード リダイレクト サポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。
-r yes no	インストール後にシステムを自動的に再起動します。デフォルトは、 <b>いいえ</b> です。
-s	自己署名証明書 subject DN。デフォルトでは、インストーラは、Blast を使用します。
-C yes no	クリップボード リダイレクト サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-F yes no	CDR サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-M yes no	Linux Agent を管理対象または管理対象外のエージェントにアップグレードします。デフォルトは、 <b>yes</b> です。
-S yes no	シングル サインオン (SSO) サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-T yes no	True SSO サポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。
-U yes no	USB サポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。

表 5-3. install\_viewagent.sh パラメータの例

状況	例
新規インストール	<pre>sudo ./install_viewagent.sh -A yes</pre> <p>新規インストールでは、常に新しいデスクトップ プールの作成が必要です。</p>
管理対象外の仮想マシンからアップグレードし、管理対象仮想マシンのスタイルを保持します。	<pre>sudo ./install_viewagent.sh -A yes-M no</pre> <p>このタイプのアップグレードには、新しいデスクトップ プールの作成は必要ありません。既存のデスクトップ プールを再使用できます。</p> <p><b>注:</b> パフォーマンスを維持するため、管理対象外の仮想マシンは使用しないでください。</p>
管理対象外の仮想マシン展開からアップグレードし、管理対象仮想マシンのスタイルへ変換します。アップグレードには、ブローカに新しいデスクトップ プールの作成が必須です。	<pre>sudo ./install_viewagent.sh -A yes</pre> <p>このタイプのアップグレードには、新しいデスクトップ プールの作成が必要です。既存のデスクトップ プールを削除する必要があります。</p>

## Linux Agent 用証明書の構成

Linux Agent をインストールすると、インストーラによって VMwareBlastServer の自己署名証明書が生成されます。

- Blast Security Gateway がブローカーで無効になっているとき、VMwareBlastServer は、この証明書を HTML Access を使用するブラウザに提示して Linux デスクトップに接続します。
- Blast Security Gateway がブローカーで有効になっているとき、Blast Security Gateway の証明書は、この証明書をブラウザに提示します。

業界またはセキュリティの規制に準拠するために、自己署名証明書に代わって認証局 (CA) が署名した証明書を使用できます。

### 手順

- 1 VMwareBlastServer のプライベート キーと証明書をインストールします。
  - a プライベート キーを rui.key に名前変更し、証明書を rui.crt に名前変更します。
  - b `sudo chmod 550 /etc/vmware/ssl` を実行します。
  - c rui.crt と rui.key を /etc/vmware/ssl にコピーします。
  - d `chmod 440 /etc/vmware/ssl` を実行します。

## 2 ルート認証局と中間認証局を Linux OS の認証局ストアにインストールします。

**注:** Linux システムの設定変更については、お使いの Linux ディストリビューションのドキュメントを確認してください。

## Linux 仮想マシンでの Horizon Agent のアップグレード

Horizon Agent の最新バージョンをインストールすることにより、Linux 仮想マシンで Horizon Agent をアップグレードできます。

管理対象外の仮想マシン：エージェント インストーラは、ブローカ Admin 情報を必要とするブローカに仮想マシンを登録します。[デスクトップ プール作成]ウィザードは、[マシン ソース] ページの [その他のソース] を使用して、登録された仮想マシンを選択します。

管理対象仮想マシン：インストーラはブローカと通信を行いません。[デスクトップ プール作成]ウィザードは、[マシン ソース] ページの [vCenter Server 仮想マシン] を使用して、vCenter Server の仮想マシンを選択します。管理対象仮想マシン展開は、以下の機能をサポートします。

- リモート マシンの電源ポリシー
- ユーザーによるマシンのリセットを許可

**注:** Horizon Agent for Linux 7.0.0 を含む以前のバージョンは、管理対象外の仮想マシンとして機能していました。Horizon Agent for Linux 7.0.1 は、管理対象仮想マシンサポートとして機能します。

管理対象外の仮想マシン展開から管理対象仮想マシン展開にアップグレードする場合、以下の方法を使用できます。

- 管理対象外の仮想マシン展開を保持し、必要なバージョンにアップグレードします。このタイプのアップグレードでは、Horizon Connection Server での構成変更は必要ありません。
- 管理対象外の仮想マシン展開から任意のバージョンの管理対象仮想マシン展開にアップグレードします。このタイプのアップグレードには、Horizon Connection Server で新しいデスクトップ プールの作成が必要です。

**注:** 管理対象仮想マシン展開からのアップグレードの場合、管理対象仮想マシン展開を保持し、必要なバージョンにアップグレードできます。ただし、アップグレード時に管理対象仮想マシン展開から管理対象外の仮想マシン展開への変換はサポートされていません。

アップグレードでは以下のパラメータを利用できます。

表 5-4. Horizon Agent のアップグレードのオプション パラメータ

パラメータ	説明
-A yes	エンド ユーザー使用許諾契約書 (EULA) および FIPS の声明に同意します。インストールを続行するには、 <b>yes</b> を指定する必要があります。このパラメータを指定しないと、インストール スクリプトで値が要求されます。
-a yes no	オーディオ入力ダイレクト サポートをインストールするかバイパスします。
-f yes no	FIPS (Federal Information Processing Standards) 140-2 準拠の暗号モジュールのサポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。詳細については、 <a href="#">Horizon Linux デスクトップの機能</a> で FIPS 140-2 モードの説明を参照してください。
-m yes no	スマート カードリダイレクト サポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。

パラメータ	説明
-r yes no	インストール後にオペレーティング システムを再起動します。デフォルトは、 <b>no</b> です。
-C yes no	クリップボード リダイレクト サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-F yes no	CDR サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-M yes no	Linux Agent を管理対象 管理対象外のエージェントにアップグレードします。デフォルト値は <b>yes</b> です。
-S yes no	シングル サインオン (SSO) サポートをインストールまたはバイパスします。デフォルトは、 <b>yes</b> です。
-U yes no	USB サポートをインストールまたはバイパスします。デフォルトは、 <b>いいえ</b> です。

## Linux 仮想マシンでの Horizon Agent のアップグレード

Horizon Agent の最新バージョンをインストールすることにより、Linux マシンで Horizon Agent をアップグレードできます。

### 前提条件

- VMwareBlastServer プロセスが実行されていないことを確認します。

このプロセスを停止するには、ユーザーがマシンからログオフしていて、アクティブなデスクトップ セッションがないことを確認するか、マシンを再起動します。

### 手順

- 1 <https://my.vmware.com/web/vmware/downloads> の VMware ダウンロード サイトから Horizon Agent for Linux の最新のインストーラ ファイルをダウンロードします。

[Desktop & End-User Computing] (デスクトップおよびエンドユーザー コンピューティング) で、VMware Horizon 7 のダウンロードを選択します。この中に、Horizon Agent for Linux のインストーラが含まれています。

インストーラ ファイル名は、64 ビット Linux で `VMware-viewagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz` です。y.y.yはバージョン番号、xxxxxxxはビルド番号です。

- 2 お使いの Linux ディストリビューションの tar ボールをゲスト OS に展開します。

例：

```
tar -xzf <Horizon Agent の tar ボール>
```

- 3 tar ボール フォルダに移動します。



- 4 管理対象外の仮想マシンをアップグレードするには、次のいずれかの展開シナリオで `install_viewagent.sh` スクリプトを実行します。

オプション	説明
管理対象外の仮想マシン展開をアップグレードし、管理対象外の仮想マシン展開を保持する	<pre>sudo ./install_viewagent.sh -A yes -M no</pre> <p><b>注:</b> パフォーマンスを維持するため、管理対象外の仮想マシンは使用しないでください。</p>
管理対象外の仮想マシン展開をアップグレードし、管理対象仮想マシン展開に変更する	<pre>sudo ./install_viewagent.sh -A yes -M yes</pre> <p><b>注:</b> Horizon Administrator で、管理対象外の仮想マシン展開用の既存のデスクトップ プールを削除し、管理対象仮想マシン展開用のデスクトップ プールを作成します。詳細については、<a href="#">Linux 版手動デスクトップ プールの作成</a>を参照してください。</p>
管理対象仮想マシン展開をアップグレードする	<pre>sudo ./install_viewagent.sh -A yes -M yes</pre> <p><b>注:</b> アップグレード後、既存のデスクトップ プールを再利用できます。</p>

## Horizon 7 for Linux マシンをアンインストール

仮想マシンで Horizon 7 for Linux をアンインストールするには、Horizon Agent をアンインストールし、構成ファイルを削除する必要があります。

### 前提条件

VMwareBlastServer プロセスが実行されていないことを確認します。このプロセスを停止するには、マシンからログオフしていて、アクティブなデスクトップ セッションがないことを確認するか、マシンを再起動します。

### 手順

- 1 仮想マシンでターミナル ウィンドウを開き、Horizon Agent のアンインストール スクリプトを実行します。

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

スクリプトは、Horizon Agent のプロセスを停止し、インストール ディレクトリ `/usr/lib/vmware/viewagent` から Horizon Agent サービスとソフトウェアを削除します。

- 2 `/etc/vmware` のディレクトリから、Horizon 7 for Linux の構成ファイルを手動で削除します。

# Linux デスクトップの構成オプション

構成ファイルを使用してさまざまなオプションを構成し、ユーザーの使用環境をカスタマイズできます。

この章には、次のトピックが含まれています。

- [Linux デスクトップでの構成ファイルのオプション設定](#)
- [スマート ポリシー の使用](#)
- [Linux デスクトップの Blast 設定の例](#)
- [Linux デスクトップのクライアント ドライブ リダイレクト オプションの例](#)
- [Linux デスクトップの vSphere コンソールへの表示を抑制する](#)

## Linux デスクトップでの構成ファイルのオプション設定

/etc/vmware/config ファイルまたは /etc/vmware/viewagent-custom.conf ファイルにエントリを追加して、特定のオプションを構成できます。

インストーラは、Horizon Agent のインストール中に、2 つの構成テンプレート ファイル config.template と viewagent-custom.conf.template を /etc/vmware にコピーします。/etc/vmware/config ファイルと /etc/vmware/viewagent-custom.conf ファイルが存在しない場合、インストーラは config.template を config に、viewagent-custom.conf.template を viewagent-custom.conf にコピーします。テンプレート ファイルではすべての構成オプションがリストされていて、詳細な説明があります。オプションを設定するには、コメントを削除して値を適切に変更します。

たとえば、/etc/vmware/config の次の行により、ビルドで可逆圧縮 PNG モードが有効になります。

```
RemoteDisplay.buildToPNG=TRUE
```

構成を変更したら、Linux を再起動して変更を有効にしてください。

## /etc/vmware/config の構成オプション

VMwareBlastServer およびその関連プラグインでは、構成ファイル `/etc/vmware/config` が使用されます。

**注:** 次の表に、Horizon Agent 構成ファイル中の USB 用の各エージェント適用型ポリシー設定について説明します。Horizon Agent は設定を使用して、USB がホスト マシンに転送できるかどうかを判断します。また、Horizon Agent は Horizon Client に設定を渡し、解釈と適用が行われます。マージ (**m**) 修飾子を指定した場合は、Horizon Agent フィルタ ポリシー設定が Horizon Client フィルタ ポリシー設定に追加適用されます。オーバーライド (**o**) 修飾子を使用した場合は、Horizon Client フィルタ ポリシー設定ではなく Horizon Agent フィルタ ポリシー設定が使用されます。

表 6-1. /etc/vmware/config の構成オプション

オプション	値/形式	デフォルト	説明
Clipboard.Direction	0, 1, 2, または 3	2	このオプションを使用して、クリップボードリダイレクトポリシーを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> <li>■ 0 - クリップボードリダイレクトを無効にします。</li> <li>■ 1 - クリップボードリダイレクトを両方向で有効にします。</li> <li>■ 2 - クリップボードリダイレクトをクライアントからリモートデスクトップのみで有効にします。</li> <li>■ 3 - クリップボードリダイレクトをリモートデスクトップからクライアントのみで有効にします。</li> </ul>
RemoteDisplay.allowAudio	true または false	true	このオプションを設定して、オーディオ出力を有効/無効にします。
RemoteDisplay.allowH264	true または false	true	このオプションを使用して、H.264 エンコードを有効または無効に設定します。
RemoteDisplay.buildToPNG	true または false	false	特にグラフィック設計アプリケーションなどのグラフィックアプリケーションでは、Linux デスクトップのクライアント表示で正確なピクセルレベルの画像処理が必要となります。Linux デスクトップで生成されクライアントデバイスで処理される画像とビデオ再生については、ビルドに可逆圧縮 PNG モードを構成できます。この機能では、クライアントと ESXi ホストの間で追加の帯域幅が使用されます。このオプションを有効にすると、H.264 エンコードが無効になります。
RemoteDisplay.enableNetworkContinuity	true または false	true	このオプションを設定して、Horizon Agent for Linux のネットワーク接続維持機能を有効または無効にします。
RemoteDisplay.enableNetworkIntelligence	true または false	true	このオプションを設定して、Horizon Agent for Linux のネットワークインテリジェンス機能を有効または無効にします。
RemoteDisplay.enableStats	true または false	false	帯域幅、FPS、RTT などでは、VMware Blast 表示プロトコルの統計情報を mks ログで有効または無効にします。
RemoteDisplay.enableUDP	true または false	true	このオプションを設定して、Horizon Agent for Linux で UDP プロトコルサポートを有効または無効にします。

オプション	値/形式	デフォルト	説明
RemoteDisplay.maxBandwidthKbps	整数	1000000	VMware Blast セッションの最大帯域幅をキロビット/秒 (kbps) 単位で指定します。この帯域幅には、イメージ、オーディオ、仮想チャネル、および VMware Blast 制御のすべてのトラフィックが含まれます。有効な値は 4 Gbps (4096000) 未満にする必要があります。
RemoteDisplay.minBandwidthKbps	整数	256	VMware Blast セッションの最小帯域幅をキロビット/秒 (kbps) 単位で指定します。この帯域幅には、イメージ、オーディオ、仮想チャネル、および VMware Blast 制御のすべてのトラフィックが含まれます。
RemoteDisplay.maxFPS	整数	60	画面更新の最大レートを指定します。この設定を使用して、ユーザーが使用する平均帯域幅を管理します。有効値は 3 から 60 までの間にする必要があります。デフォルトは 1 秒あたり 60 回の更新です。
RemoteDisplay.maxQualityJPEG	利用可能な値の範囲：1 ~ 100	90	JPEG/PNG エンコードを使用する場合のデスクトップディスプレイのイメージ品質を指定します。高品質設定は、より静かな画面の領域に適していて、イメージ品質がより高くなります。
RemoteDisplay.midQualityJPEG	利用可能な値の範囲：1 ~ 100	35	JPEG/PNG エンコードを使用する場合のデスクトップディスプレイのイメージ品質を指定します。デスクトップディスプレイの中等度の品質を設定するために使用します。
RemoteDisplay.minQualityJPEG	利用可能な値の範囲：1 ~ 100	25	JPEG/PNG エンコードを使用する場合のデスクトップディスプレイのイメージ品質を指定します。低品質設定は、スクロール発生時など、頻繁に変化する画面の領域に適しています。
RemoteDisplay.qpmaxH264	利用可能な値の範囲：0 ~ 51	36	このオプションを使用して、H264minQP 量子化パラメータを設定します。このパラメータは、H.264 エンコードを使用するように構成されたリモートディスプレイの最高イメージ品質を指定します。RemoteDisplay.qpminH264 に設定した値よりも大きな値を設定します。
RemoteDisplay.qpminH264	利用可能な値の範囲：0 ~ 51	10	このオプションを使用して、H264maxQP 量子化パラメータを設定します。このパラメータは、H.264 エンコードを使用するように構成されたリモートディスプレイの最低イメージ品質を指定します。RemoteDisplay.qpmaxH264 に設定した値よりも小さな値を設定します。
UsbRedirPlugin.log.logLevel	error、warn、info、debug、trace、または verbose	info	このオプションを使用して、USB リダイレクト プラグインのログレベルを設定します。
UsbRedirServer.log.logLevel	error、warn、info、debug、trace、または verbose	info	このオプションを使用して、USB リダイレクト サーバのログレベルを設定します。
VMWPKcs11Plugin.log.enable	true または false	false	このオプションを設定して、True SSO 機能のログ作成モードを有効または無効にします。
VMWPKcs11Plugin.log.logLevel	error、warn、info、debug、trace、または verbose	info	このオプションを使用して、True SSO 機能のログレベルを設定します。
WC.RTAV.Enable	true または false	true	このオプションを設定して、オーディオ入力を有効/無効にします。

オプション	値/形式	デフォルト	説明
VWC.ScRedir.Enable	true または false	true	このオプションを設定して、スマート カード リダイレクトを有効/無効にします。
VWC.logLevel	fatal error、warn、info、debug、または trace	info	このオプションを使用して、VWC プロキシ ノードのログ レベルを設定します。
cdrserver.cacheEnable	true または false	true	このオプションを設定して、エージェントからクライアント側への書き込みキャッシュ機能を有効または無効にします。
cdrserver.forcedByAdmin	true または false	false	このオプションを設定して、cdrserver.shareFolders オプションで指定されていない追加のフォルダをクライアントが共有できるかどうかを制御します。
cdrserver.logLevel	error、warn、info、debug、trace、または verbose	info	このオプションを使用して、vmware-CDRserver.log ファイルのログ レベルを設定します。
cdrserver.permissions	R	RW	<p>このオプションを使用して、Horizon Client によって共有されるフォルダに対する Horizon Agent の追加の読み取り/書き込み権限を適用します。例：</p> <ul style="list-style-type: none"> <li>■ Horizon Client によって共有されるフォルダに read と write 権限があり、<b>cdrserver.permissions=R</b> が設定されている場合には、Horizon Agent には read アクセス権限のみが付与されます。</li> <li>■ Horizon Client によって共有されるフォルダに read 権限があり、<b>cdrserver.permissions=RW</b> が設定されている場合、Horizon Agent には read アクセス権限のみが付与されます。Horizon Agent は、Horizon Client によって設定された read only 属性を変更できません。Horizon Agent は、書き込みアクセス権限のみ削除できます。</li> </ul> <p>一般的な使用法は次のとおりです。</p> <ul style="list-style-type: none"> <li>■ <b>cdrserver.permissions=R</b></li> <li>■ <b>#cdrserver.permissions=R</b>（つまり、コマンドをコメントアウトするか、エントリを削除します）</li> </ul>
cdrserver.sharedFolders	<i>file_path1,R; file_path2,; file_path3,R; . . .</i>	未定義	<p>クライアントが Linux デスクトップと共有できるフォルダへのファイルパスを 1 つ以上指定します。例：</p> <ul style="list-style-type: none"> <li>■ Windows クライアントの場合： <b>C:\spreadsheets,;D:\ebooks,R</b></li> <li>■ Windows 以外のクライアントの場合： <b>/tmp/spreadsheets;/tmp/ebooks,;/home/finance,R</b></li> </ul>
collaboration.logLevel	error、info または debug	info	このオプションを使用して、共同作業セッションのログ レベルを設定します。ログ レベルが debug の場合、collabui 関数の呼び出しと collabor リストのコンテンツがログに記録されます。
collaboration.maxCollabors	10 未満の整数	5	セッションの参加に招待できる共同作業者の最大数を指定します。

オプション	値/形式	デフォルト	説明
collaboration.enableEmail	true または false	true	インストールされている メール アプリケーションでの共同作業の招待を送信するかどうかを設定するには、このオプションを使用します。このオプションを無効にすると、メール アプリケーションがインストールされていても E メールでの共同作業の招待は送信できません。
collaboration.serverUrl	[URL]	未定義	共同作業の招待状に含めるサーバ URL を指定します。
collaboration.enableControlPassing	true または false	true	このオプションは、共同作業者に Linux デスクトップのコントロールを許可または制限する場合に設定します。読み取り専用の共同作業セッションを指定するには、このオプションを <b>false</b> に設定します。
mksVNCServer.useUInputButtonMapping	true または false	false	Ubuntu または RHEL 7 で左手用マウスのサポートを有効にします。CentOS と RHEL 6 では左手用マウスがサポートされており、このオプションを設定する必要はありません。
mksVNCServer.useXExtButtonMapping	true または false	false	このオプションを設定して SLED 11 SP3 での左手用マウスのサポートを有効または無効にします。
mksvhan.clipboardSize	整数	1024	このオプションを使用して、クリップボードの最大サイズをコピー アンド ペーストします。
vdpservice.log.logLevel	fatal error、warn、info、debug、または trace	info	このオプションを使用して、vdpservice のログ レベルを設定します。
viewusb.AllowAudioIn	{m o}: {true false}	未定義、true と同じ	このオプションを使用して、オーディオ入力デバイスのリダイレクトを許可または禁止します。例: <b>o:false</b>
viewusb.AllowAudioOut	{m o}: {true false}	未定義、false と同じ	このオプションを設定して、オーディオ出力デバイスのリダイレクトを許可または禁止します。
viewusb.AllowAutoDeviceSplitting	{m o}: {true false}	未定義、false と同じ	このオプションを設定して、複数 USB デバイスの自動分割を許可または禁止します。 例: <b>m:true</b>
viewusb.AllowDevDescFailsafe	{m o}: {true false}	未定義、false と同じ	このオプションを設定して、Horizon Client が構成またはデバイス記述子を取得できない場合にデバイスのリダイレクトを許可または禁止します。構成またはデバイス記述子を取得できない場合でも、デバイスを許可するには、 <b>IncludeVidPid</b> または <b>IncludePath</b> などの Include フィルタにデバイスを含めます。
viewusb.AllowHIDBootable	{m o}: {true false}	未定義、true と同じ	このオプションを使用して、キーボードとマウス以外で、起動時に利用可能な入力デバイス (HID 起動可能なデバイス) のリダイレクトを許可または禁止します。
viewusb.AllowKeyboardMouse	{m o}: {true false}	未定義、false と同じ	このオプションを使用して、統合型ポインティング デバイス (マウス、トラックボール、タッチ パッドなど) 付きキーボードのリダイレクトを許可または禁止します。
viewusb.AllowSmartcard	{m o}: {true false}	未定義、false と同じ	このオプションを設定して、スマートカード デバイスのリダイレクトを許可または禁止します。
viewusb.AllowVideo	{m o}: {true false}	未定義、true と同じ	このオプションを使用して、ビデオ デバイスのリダイレクトを許可または禁止します。

オプション	値/形式	デフォルト	説明
viewusb.DisableRemoteConfig	<b>{m o}:</b> <b>{true false}</b>	未定義、false と同じ	このオプションを設定して、USB デバイスのフィルタリングを実行するときの Horizon Agent 設定の使用を有効または無効にします。
viewusb.ExcludeAllDevices	<b>{true false}</b>	未定義、false と同じ	このオプションを使用して、リダイレクト対象からすべての USB デバイスを除外したり、すべての USB デバイスをリダイレクト対象に追加したりします。 <b>true</b> に設定すると、その他のポリシー設定を使用して、特定のデバイスまたはデバイス ファミリがリダイレクトされるようにすることができます。 <b>false</b> に設定すると、その他のポリシー設定を使用して、特定のデバイスまたはデバイス ファミリがリダイレクトされるのを防止できます。Horizon Agent で <b>ExcludeAllDevices</b> の値を <b>true</b> に設定し、この設定が Horizon Client に渡されると、Horizon Agent の設定によって Horizon Client の設定はオーバーライドされます。
viewusb.ExcludeFamily	<b>{m o}:</b> <i>family_name_1</i> ; <i>family_name_2</i> ;...	未定義	このオプションを使用して、リダイレクト対象からデバイス ファミリーを除外します。例： <b>m:bluetooth;smart-card</b>  自動デバイス分割を有効にした場合、Horizon は複合 USB デバイスの各インターフェイスのデバイス ファミリーを調べ、除外するインターフェイスを判断します。自動デバイス分割を無効にした場合、Horizon は複合 USB デバイス全体のデバイス ファミリーを調べます。  <b>注:</b> マウスとキーボードはリダイレクト対象からデフォルトで除外されているため、この設定を使用して除外する必要はありません。
viewusb.ExcludePath	<b>{m o}:</b> <i>bus-x1</i> [/ <i>y1</i> ].../ <i>port-z1</i> ; <i>bus-x2</i> [/ <i>y2</i> ].../ <i>port-z2</i> ;...	未定義	このオプションを使用して、特定のハブまたはポートのバスにあるデバイスをリダイレクト対象から除外します。バスやポート番号は 16 進数で指定する必要があります。バスにワイルドカード文字を使用することはできません。 例： <b>m:bus-1/2/3_port- 02;bus-1/1/1/4_port-ff</b>
viewusb.ExcludeVidPid	<b>{m o}:</b> <i>vid-xxx1</i> <i>pid-yyy1</i> ; <i>vid-xxx2</i> <i>pid-yyy2</i> ;...	未定義	このオプションを設定して、指定したベンダーとプロダクト ID のデバイスを、リダイレクト対象から除外します。ID 番号は 16 進数で指定する必要があります。ID の各桁にワイルドカード文字 (*) を使用できます。 例： <b>o:vid-0781_pid- ****;vid-0561_pid-554c</b>
viewusb.IncludeFamily	<b>{m o}:</b> <i>family_name_1</i> ; <i>family_name_2</i> ...	未定義	このオプションを設定して、デバイス ファミリーをリダイレクト対象に含めます。 例： <b>o:storage; smart-card</b>
viewusb.IncludePath	<b>{m o}:</b> <i>bus-x1</i> [/ <i>y1</i> ].../ <i>port-z1</i> ; <i>bus-x2</i> [/ <i>y2</i> ].../ <i>portz2</i> ;...	未定義	このオプションを使用して、特定のハブやポートのバスにあるデバイスをリダイレクト対象に含めます。バスやポート番号は 16 進数で指定する必要があります。バスにワイルドカード文字を使用することはできません。 例： <b>m:bus-1/2_port- 02;bus-1/7/1/4_port-0f</b>

オプション	値/形式	デフォルト	説明
viewusb.IncludeVidPid	<code>{m o}:vid-xxx1_pid-yyy1[;vid-xxx2_pid-yyy2;...]</code>	未定義	<p>このオプションを設定して、指定したベンダーとプロダクト ID のデバイスを、リダイレクト対象に含めます。ID 番号は 16 進数で指定する必要があります。ID の各桁にワイルドカード文字 (*) を使用できます。</p> <p>例 : <code>o:vid-***_pid-0001;vid-0561_pid-554c</code></p>
viewusb.SplitExcludeVidPid	<code>{m o}:vid-xxx1_pid-yyy1[;vid-xxx2_pid-yyy2;...]</code>	未定義	<p>このオプションを使用して、ベンダーとプロダクト ID を基準として特定の複合 USB デバイスをで分割の対象として除外するか追加するかを指定します。この設定の形式は、<code>vid-xxx1_pid-yyy1[;vid-xxx2_pid-yyy2;...]</code> となります。ID 番号は 16 進数で指定する必要があります。ID の各桁にワイルドカード文字 (*) を使用できます。</p> <p>例 : <code>m:vid-0f0f_pid-55**</code></p>
viewusb.SplitVidPid	<code>{m o}: vid-xxxx_pid-yyy([exintf:zz[;exintf:ww]])[;...]</code>	未定義	<p>このオプションを設定して、ベンダーおよびプロダクト ID で指定した複合 USB デバイスのコンポーネントを別のデバイスとして扱います。この設定の形式は、<code>vid-xxxx_pid-yyy(exintf:zz[;exintf:ww])</code> となります。</p> <p><b>exintf</b> というキーワードを使用すれば、インターフェイス番号を指定することで、コンポーネントをリダイレクトから除外することができます。ID 番号は 16 進数で指定し、インターフェイス番号は先行ゼロをすべて含む 10 進数で指定する必要があります。ID の各桁にワイルドカード文字 (*) を使用できます。</p> <p>例 :</p> <p><code>o:vid-0f0f_pid-*** (exintf-01);vid-0781_pid-554c(exintf:01;exintf:02)</code></p> <p><b>注:</b> 明示的に除外しなかったコンポーネントは、Horizon で自動的に含まれることはありません。これらのコンポーネントを含めるには、<b>Include VidPid Device</b> などのフィルタポリシーを指定する必要があります。</p>

## /etc/vmware/viewagent-custom.conf の構成オプション

Java Standalone Agent では、構成ファイル `/etc/vmware/viewagent-custom.conf` が使用されます。

表 6-2. `/etc/vmware/viewagent-custom.conf` の構成オプション

オプション	値	デフォルト	説明
CDREnable	true または false	true	このオプションを使用して、クライアント ドライブのリダイレクト (CDR) 機能を有効/無効にします。
CollaborationEnable	true または false	true	Linux デスクトップのセッション共同作業機能を有効または無効にするには、このオプションを使用します。



オプション	値	デフォルト	説明
EndpointVPNEnable	true または false	false	このオプションは、User Environment Manager コンソールで使用するエンドポイントの IP アドレス範囲とエンドポイントの IP アドレスを比較するときに、クライアントの物理ネットワーク カードの IP アドレスを使用するのか、VPN IP アドレスを使用するのかを指定する場合に設定します。オプションを false に設定すると、クライアントの物理ネットワーク カードの IP アドレスが使用されます。それ以外の場合は、VPN IP アドレスが使用されます。
HelpDeskEnable	true または false	true	このオプションを設定して、ヘルプ デスク ツール機能を有効/無効にします。
KeyboardLayoutSync	true または false	true	<p>このオプションを使用して、クライアントのシステム言語リストと現在のキーボード レイアウトを Horizon Agent for Linux デスクトップと同期させるかどうかを指定します。</p> <p>この設定を有効にする、あるいは構成しない場合、同期が許可されます。この設定を無効にすると、同期が許可されません。</p> <p>この機能は、Horizon Client for Windows のみでサポートされ、英語、フランス語、ドイツ語、日本語、韓国語、スペイン語、簡体字中国語、および繁体字中国語の言語でのみサポートされます。</p>
LogCnt	整数	-1	<p>このオプションを使用して、/tmp/vmware-root に保持するログ ファイルの数を設定します。</p> <ul style="list-style-type: none"> <li>■ -1 - すべて保持</li> <li>■ 0 - すべて削除</li> <li>■ &gt; 0 - 保持するログの数。</li> </ul>
NetbiosDomain	すべて大文字のテキスト文字列		True SSO を設定する場合は、このオプションを使用して、組織のドメインの NetBIOS 名を設定します。
OfflineJoinDomain	pbis または samba	pbis	このオプションを使用すると、インスタント クローンのオフライン ドメイン参加が設定されます。オフライン ドメイン参加の実行方法は、PBISO (PowerBroker Identity Services Open) 認証または Samba オフライン ドメイン参加になります。このプロパティに pbis または samba 以外の値を設定すると、オフライン ドメイン参加が無視されます。
RunOnceScript			<p>このオプションを使用して、Active Directory にクローン作成された仮想マシンに再度参加します。</p> <p>ホスト名が変更された後に、RunOnceScript オプションを設定します。指定されたスクリプトは、最初のホスト名の変更後、一度だけ実行されます。エージェント サービスが開始され、ホスト名がエージェントのインストール後に変更された場合、スクリプトは root 権限で実行されます。</p> <p>たとえば、winbind ソリューションでは、winbind でベース仮想マシンを Active Directory に参加させ、このオプションをスクリプトパスに設定する必要があります。スクリプトには、ドメインへ再度参加させるコマンド <code>/usr/bin/net ads join -U &lt;ADUserName&gt; %&lt;ADUserPassword&gt;</code> が含まれている必要があります。仮想マシンのクローン作成後、オペレーティング システムのカスタマイズによってホスト名が変更されます。Agent サービスが開始されると、クローン作成された仮想マシンを Active Directory へ参加するスクリプトが実行されます。</p>

オプション	値	デフォルト	説明
RunOnceScriptTimeout		120	このオプションを使用して、RunOnceScript オプションのタイムアウト値を秒数で設定します。  たとえば、RunOnceScriptTimeout=120 のように設定します。
SSLCiphers	テキスト文字列	!aNULL:kECDH+AESGCM:ECDH+AESGCM:RSA+AESGCM:kECDH+AES:ECDH+AES:RSA+AES	暗号化のリストを指定します。 <a href="https://www.openssl.org/docs/manmaster/man1/ciphers.html">https://www.openssl.org/docs/manmaster/man1/ciphers.html</a> で定義されている形式を使用する必要があります。
SSLProtocols	テキスト文字列	TLSv1_1:TLSv1_2	セキュリティ プロトコルを指定します。サポートされるプロトコルは、TLSv1.0、TLSv1.1、TLSv1.2 です。
SSODesktopType	UseGnomeClassic 、 UseGnomeFlashback 、 UseGnomeUbuntu 、UseMATE、 UseKdePlasma または		このオプションは、SSO を有効にするときにデフォルトのデスクトップ環境ではなく、他のデスクトップ環境を指定する場合に使用します。このオプションを指定する前に、選択するデスクトップ環境がデスクトップにインストールされていることを確認する必要があります。このオプションを Ubuntu 16.04/18.04 デスクトップで設定すると、SSO 機能が有効かどうかに関わらず、このオプションが有効になります。このオプションを RHEL/CentOS 7.x デスクトップで指定すると、SSO が有効になっている場合にのみ、選択したデスクトップ環境が使用されます。  <b>注:</b> このオプションは、RHEL/CentOS 6 および SLED 11 デスクトップでサポートされていません。デフォルトのデスクトップ環境として KDE をセットアップし、SSO を有効にする方法については、『Horizon 7 for Linux デスクトップのセットアップ』ドキュメントを参照してください。
SSOEnable	true または false	true	このオプションを設定して、シングル サインオン (SSO) を有効/無効にします。
SSOUserFormat	テキスト文字列	[username]	シングル サインオンのログイン名の形式を指定します。デフォルトはユーザー名のみです。ドメイン名も要求する場合は、このオプションを設定します。一般的にログイン名では、ドメイン名と特殊文字にユーザー名を続けます。特殊文字をバックスラッシュにする場合は、別のバックスラッシュを使用してエスケープする必要があります。ログイン名の形式は次のとおりです。 <ul style="list-style-type: none"><li>■ SSOUserFormat=[domain]\\[username]</li><li>■ SSOUserFormat=[domain]+[username]</li><li>■ SSOUserFormat=[username]@[domain]</li></ul>
サブネット	CIDR IP アドレス形式の値	[subnet]	このオプションは、他のマシンが Horizon Agent for Linux との接続に使用するサブネットを設定する場合に使用します。異なるサブネットのローカル IP アドレスが複数ある場合、設定したサブネットのローカル IP アドレスが Horizon Agent for Linux との接続に使用されます。値は、CIDR IP アドレス形式で指定する必要があります。たとえば、Subnet=123.456.7.8/24 と設定します。

オプション	値	デフォルト	説明
UEMEnable	true または false	false	このオプションを使用して、User Environment Manager スマートポリシーを有効または無効にします。オプションを有効に設定し、User Environment Manager スマートポリシーの条件を満たすと、このポリシーが適用されます。
UEMNetworkPath	テキスト文字列		このオプションには、User Environment Manager コンソールで設定されている同じネットワークパスを設定する必要があります。パスは、//10.111.22.333/view/LinuxAgent/UEMConfig のような形式で指定します。

**注:** 3つのセキュリティオプション、SSLCiphers、SSLProtocols、SSLCipherServerPreference は VMwareBlastServer プロセス用です。VMwareBlastServer プロセスが開始されると、Java Standalone Agent はこれらのオプションをパラメータとして渡します。Blast Secure Gateway (BSG) が有効であるとき、これらのオプションは BSG と Linux デスクトップの間の接続に影響します。BSG が無効であるとき、これらのオプションはクライアントと Linux デスクトップの間の接続に影響します。

## スマートポリシーの使用

スマートポリシーを使用して、特定のリモート Linux デスクトップでの USB リダイレクト、クリップボードリダイレクト、クライアントドライブリダイレクト機能の動作を制御できます。

スマートポリシーにより、特定の条件が満たされる場合にのみ有効になるポリシーを作成できます。たとえば、ユーザーが企業のネットワーク以外からリモートデスクトップに接続している場合はクライアントドライブリダイレクト機能を無効にするポリシーを設定できます。

## スマートポリシーの要件

スマートポリシーを使用するには、Horizon 7 環境が特定の要件を満たす必要があります。

- スマートポリシーで管理するリモートデスクトップに、Horizon Agent 7.5 以降と VMware User Environment Manager 9.4 以降をインストールする必要があります。
- スマートポリシーで管理するリモート Linux デスクトップに接続するには、ユーザーが Horizon Client 4.8 以降を使用する必要があります。
- /etc/vmware/viewagent-custom.conf ファイルで、UEMEnable オプションを有効にし、UEMNetworkPath オプションを設定する必要があります。[Linux デスクトップでの構成ファイルのオプション設定](#)を参照してください。

## User Environment Manager のインストール

Horizon スマートポリシーを使用して、リモート Linux デスクトップ機能の動作を制御するには、User Environment Manager 9.4 以降をリモート Linux デスクトップにインストールする必要があります。

User Environment Manager インストーラは、VMware ダウンロード ページからダウンロードできます。User Environment Manager 環境を管理する任意のデスクトップに User Environment Manager 管理コンソール コンポーネントをインストールできます。

リンククローン プールの場合、リンク クローンの基本イメージとして使用する親仮想マシンに User Environment Manager をインストールします。公開デスクトップ プールの場合、RDS デスクトップ セッションを提供する RDS ホストに User Environment Manager をインストールします。

User Environment Manager のシステム要件および完全なインストール手順については、『VMware User Environment Manager のインストールと設定』ドキュメントを参照してください。

## User Environment Manager の構成

リモート デスクトップ機能のスマート ポリシーを作成するには、User Environment Manager を構成してから使用する必要があります。

User Environment Manager を構成するには、『VMware User Environment Manager 管理ガイド』の構成手順に従います。

## Horizon スマート ポリシー設定

User Environment Manager で Horizon スマート ポリシーを作成して、リモート機能の動作を制御します。

[表 6-3. Horizon スマート ポリシー設定](#)では、User Environment Manager でリモート Linux デスクトップに Horizon スマート ポリシーを定義する場合に選択できる設定について説明します。

表 6-3. Horizon スマート ポリシー設定

設定	説明
USB リダイレクト	リモート デスクトップで USB リダイレクトを有効にするかどうかを指定します。USB リダイレクト機能により、ユーザーはリモート デスクトップから小型のフラッシュドライブ、カメラ、プリンタなどのローカルで接続された USB デバイスを使用できます。スマート ポリシーを使用して USB リダイレクトを構成する場合は、User Environment Manager 9.5 以降を使用する必要があります。
クリップボード	<p>クリップボード リダイレクトを許可する方向を決定します。次のいずれかの値を選択できます。</p> <ul style="list-style-type: none"> <li>■ [無効化]。クリップボード リダイレクトは両方の方向で無効になります。</li> <li>■ [すべて許可]。クリップボード リダイレクトが有効になります。ユーザーは、クライアント システムからリモート デスクトップ、およびリモート デスクトップからクライアント システムにコピーして貼り付けることができます。</li> <li>■ [クライアントからエージェントへのコピーを許可]。ユーザーは、クライアント システムからリモート デスクトップにのみコピーして貼り付けることができます。</li> <li>■ [エージェントからクライアントへのコピーを許可]。ユーザーは、リモート デスクトップからクライアント システムにのみコピーして貼り付けることができます。</li> </ul>
クライアント ドライブのリダイレクト	<p>リモート デスクトップでクライアント ドライブ リダイレクトを有効にするかどうかと、共有ドライブおよびフォルダを書き込み可能にするかどうかを指定します。次のいずれかの値を選択できます。</p> <ul style="list-style-type: none"> <li>■ [無効化]。リモート デスクトップでクライアント ドライブ リダイレクトが無効になります。</li> <li>■ [すべて許可]。クライアント ドライブおよびフォルダは、リモート デスクトップと共有され、読み取り/書き込み可能になります。</li> <li>■ [読み取り専用]。クライアント ドライブおよびフォルダは、リモート デスクトップと共有され、読み取り可能になりますが、書き込みはできません。</li> </ul> <p>この設定を行わない場合、共有ドライブとフォルダが書き込み可能になるかどうかは、<code>/etc/vmware/viewagent-custom.conf</code> ファイルの <code>CDREnable</code> オプションで決まります。詳細については、<a href="#">Linux デスクトップでの構成ファイルのオプション設定</a>を参照してください。</p>

通常、User Environment Manager で構成するリモート機能の Horizon スマート ポリシー設定は、対応するレジストリ キーおよびグループ ポリシー設定よりも優先されます。

## Horizon スマート ポリシー定義への条件の追加

User Environment Manager で Horizon スマート ポリシーを定義する場合、ポリシーを有効にするための必要条件を追加できます。たとえば、ユーザーが企業のネットワーク以外からリモート デスクトップに接続した場合にのみクライアント ドライブのリダイレクト機能を無効にする条件を追加できます。

**重要:** サポートされているポリシー設定をリモート Linux デスクトップで有効にするには、Horizon スマート ポリシーの定義に次の条件を追加する必要があります。これらは、現在サポートされている唯一の条件です。他の条件が設定されている場合、条件の評価の結果は false になります。

表 6-4. リモート Linux デスクトップの必須条件

状況	説明
Operating System Architecture	オペレーティング システムのアーキテクチャを確認します。値は Linux に設定する必要があります。
Endpoint IP address	エンドポイント IP アドレスが指定範囲内にあるかどうかをチェックします。範囲の先頭を空のフィールドにすると 0 と解釈され、最後のフィールドを空にすると 255 と解釈されます。

ただし、次の例のように、複数の Endpoint IP address 条件を設定できます。

```
Operating system is Linux
AND Endpoint IP address is in range 11.22.33.44 - 11.22.33.54
OR Endpoint IP address is in range 11.22.33.66 - 11.22.33.77
```

User Environment Manager 管理コンソールで条件を追加および編集する方法の詳細については、『VMware User Environment Manager 管理ガイド』を参照してください。

## User Environment Manager の Horizon スマート ポリシーの作成

User Environment Manager 管理コンソールを使用して、User Environment Manager の Horizon スマート ポリシーを作成します。Horizon スマート ポリシーを定義するときに、スマート ポリシーを有効にするために必要な条件を追加できます。

### 前提条件

- User Environment Manager をインストールして構成します。[User Environment Manager のインストール](#) および [User Environment Manager の構成](#) を参照してください。
- Horizon スマート ポリシー定義を追加できる条件について理解しておきます。[Horizon スマート ポリシー定義への条件の追加](#) を参照してください。
- UEMEnable オプションを有効にして、/etc/vmware/viewagent-custom.conf ファイルで UEMNetworkPath オプションを設定します。[Linux デスクトップでの構成ファイルのオプション設定](#) を参照してください。

**注:** ネットワークの待ち時間が長い場合は、User Environment Manager が変更処理を完了できるように、新しいスマート ポリシーまたは更新されたスマート ポリシーを保存してから 1 分以上経過した後で、影響を受けるデスクトップに接続しているエンドユーザーに通知します。

User Environment Manager 管理コンソールの使用方法の詳細については、『VMware User Environment Manager 管理ガイド』ドキュメントを参照してください。

#### 手順

- 1 User Environment Manager 管理コンソールで、[ユーザー環境] タブを選択し、ツリー ビューで [Horizon スマート ポリシー] をクリックします。

既存の Horizon スマート ポリシー定義がある場合には、[Horizon スマート ポリシー] ペインに表示されます。

- 2 [Horizon スマート ポリシー] を右クリックし、[Horizon スマート ポリシー定義の作成] を選択して新しいスマート ポリシーを作成します。

[Horizon スマート ポリシー] ダイアログ ボックスが表示されます。

- 3 [設定] タブを選択し、スマート ポリシー設定を定義します。

- a [全般設定] セクションで、[名前] テキスト ボックスにスマート ポリシーの名前を入力します。

たとえば、スマート ポリシーがクライアント ドライブ リダイレクト機能に影響する場合、CDR などのスマート ポリシー名を付けます。

- b [Horizon スマート ポリシー設定] セクションで、スマート ポリシーに含めるリモート デスクトップ機能と設定を選択します。

複数のリモート デスクトップ機能を選択できます。

- 4 リモート Linux デスクトップで、新しいスマート ポリシーの使用条件を追加します。

- a [条件] タブを選択して [追加] をクリックし、[オペレーティング システム アーキテクチャ] 条件を選択します。

- b 値を **Linux** に設定します。

```
Operating System is Linux
```

- c [追加] をクリックして、[エンドポイントの IP アドレス] 条件を選択します。

[AND] 演算子がデフォルトで追加されます。

- d [エンドポイントの IP アドレス] ダイアログ ボックスで、エンドポイントの IP アドレス範囲を設定して [OK] をクリックします。

次に、正しい文の例を示します。

```
Operating System is Linux
AND Endpoint IP address is in range 11.22.33.44 - 11.22.33.54
```

- 5 [保存] をクリックしてスマート ポリシーを保存します。

User Environment Manager は、ユーザーがリモート デスクトップに接続または再接続するたびに Horizon スマート ポリシーを処理します。

User Environment Manager は複数のスマート ポリシーをスマート ポリシー名のアルファベット順に処理します。Horizon スマート ポリシーは、[Horizon スマート ポリシー] ペインにアルファベット順に表示されます。スマート ポリシーが競合する場合、最後に処理されたスマート ポリシーが優先されます。たとえば、Sue というユーザーの USB リダイレクトを有効にする Sue というスマート ポリシーがあり Ubuntu1604 というデスクトップ プールの USB リダイレクトを無効にする Pool という別のスマート ポリシーがある場合、Sue が Ubuntu1604 デスクトップ プールのリモート デスクトップに接続したときに USB リダイレクト機能が有効になります。

## Linux デスクトップの Blast 設定の例

リモート デスクトップ ディスプレイのイメージ品質を調整して、ユーザーの使用環境を向上できます。イメージ品質を向上すると、ネットワーク接続が不安定になる場合でも、一貫性のあるユーザーの使用環境を維持できます。

### VMware Blast Extreme プロトコル設定の例

VMwareBlastServer およびその関連プラグインでは、構成ファイル `/etc/vmware/config` が使用されます。

表 6-5. `/etc/vmware/config` の Blast 構成オプションの例

オプション名	パラメータ	高速 LAN	LAN	専用 WAN	ブロードバンド WAN	低速 WAN	超低速
帯域幅設定	RemoteDisplay.maxBandwidthKbps	1000000 (1 Gbps)	1000000 (1 Gbps)	1000000 (1 Gbps)	5000 (5 Mbps)	2000 (2 Mbps)	1000 (1 Mbps)
最大 FPS	RemoteDisplay.maxFPS	60	30	30	20	15	5
オーディオ再生	RemoteDisplay.allowAudio	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
表示品質 (JPEG/PNG)	RemoteDisplay.maxQualityJPEG	90	90	90	70	60	50
表示品質 (JPEG/PNG)	RemoteDisplay.midQualityJPEG	35	35	35	35	35	35
表示品質 (JPEG/PNG)	RemoteDisplay.minQualityJPEG	25	25	25	20	20	20
表示品質 (H.264)	RemoteDisplay.qpmxH264	28	36	36	36	36	42
表示品質 (H.264)	RemoteDisplay.qpmiH264	10	10	10	10	10	10

## Linux デスクトップのクライアント ドライブ リダイレクト オプションの例

クライアント ドライブ リダイレクション (CDR) オプションを構成して、リモートの Linux デスクトップからローカル システムの共有フォルダとドライブにアクセスできるかどうかを決定します。

`/etc/vmware/config` ファイルにエントリを追加して、CDR 設定を構成します。

次の構成例では、`d:\ebooks` と `C:\spreadsheets` フォルダを共有して両方のフォルダを読み取り専用にし、クライアントが他のフォルダを共有できないようにしています。

```
cdserver.forcedByAdmin=true
cdserver.sharedFolders=d:\ebooks,;c:\spreadsheets,
cdserver.permissions=R
```

前の例では、オプションを適切にパースするには、カンマ「`,`」を **ebooks** と **spreadsheets** の後に置く必要があります。

`cdserver.sharedFolders` オプションに含まれる「**R**」は、この設定に表示されているすべてのフォルダに影響を及ぼします。次の例では、**R** 値が `/home/jsmith` フォルダパスの後にのみ配置されていますが、**ebooks** と **spreadsheets** フォルダの両方が読み取り専用になります。

```
cdserver.sharedFolders=d:\ebooks,;c:\spreadsheets,;/home/jsmith,R
```

## Linux デスクトップの vSphere コンソールへの表示を抑制する

ユーザーが Linux デスクトップに接続するときに、このデスクトップを Linux 仮想マシンの vSphere コンソールに表示させることもできます。ユーザーがデスクトップに接続するときには vSphere コンソールを必ず空白にするように Linux 仮想マシンを構成できます。

### 手順

- ◆ ESXi ホストで、Linux 仮想マシンの `vmx` ファイルに次の行を追加します。

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

ユーザーがデスクトップからログアウトしたときに仮想マシンに接続するときでも、vSphere コンソールの表示は空白のままになります。



# Linux デスクトップ プールの作成と管理

# 7

Linux 仮想マシンを構成してリモート デスクトップとして使用するには、Linux 仮想マシンのデスクトップ プールを作成する必要があります。

Horizon for Linux では、次のデスクトップ プール タイプがサポートされます。

- vCenter Server の仮想マシンがある手動デスクトップ プール
- 自動化された完全クローン デスクトップ プール
- インスタント クローン フローティング デスクトップ プール

vCenter Server の仮想マシンがある手動デスクトップ プールを作成するには、すべての仮想マシンに Horizon Agent をインストールする必要があります。次に、接続サーバ デスクトップ プール作成ウィザードを使用して、仮想マシンをデスクトップ プールに追加します。多くの仮想マシンのクローンを作成する方法の詳細は、[Linux デスクトップの一括デプロイの概要](#)を参照してください。

自動化された完全なクローン デスクトップ プールを作成するには、Linux 仮想マシン テンプレートに Horizon 7 Agent をインストールする必要があります。次に、接続サーバ デスクトップ プール作成ウィザードを使用して、完全な仮想マシンのクローンを作成します。

インスタント クローン フローティング デスクトップ プールを作成するには、PBIS Open 環境が設定された Linux 仮想マシンに Horizon 7 Agent をインストールして、テンプレートを作成する必要があります。次に、接続サーバ デスクトップ プール作成ウィザードを使用して、インスタント クローン フローティング デスクトップ プールを作成します。

この章には、次のトピックが含まれています。

- [Linux 版手動デスクトップ プールの作成](#)
- [Linux デスクトップ プールの管理](#)
- [Linux の自動化された完全なクローン デスクトップ プールの作成](#)
- [Linux のインスタント クローン フローティング デスクトップ プールの作成](#)
- [ブローカ PowerCLI コマンド](#)

## Linux 版手動デスクトップ プールの作成

Linux 仮想マシンの手動デスクトップ プールを作成できます。

**前提条件**

- Horizon Agent が Linux ゲスト OS にインストールされていることを確認します。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- VMware vCenter Server が Horizon Connection Server に追加されたことを確認します。

**手順**

- 1 Horizon Administrator で、手動デスクトップ プールを追加します。  
[カタログ] - [デスクトップ プール] - [追加] を選択します。
- 2 [手動デスクトップ プール] を選択します。
- 3 [ユーザー割り当て] ページで、デスクトップ プールのマシンに専用ユーザーまたはフローティング ユーザー割り当てのどちらかを選択して、[次へ] をクリックします。
- 4 [マシン ソース] ページで、[vCenter 仮想マシン] を選択して、[次へ] をクリックします。
- 5 vCenter Server ページで、適切な vCenter Server を選択し、[次へ] をクリックします。
- 6 **デスクトップ プール ID**を入力します。
- 7 [デスクトップ プールの設定] ページで、次のオプションを設定します。

オプション	説明
デフォルト表示プロトコル	VMware Blast
ユーザーがプロトコルを選択できるようにする	いいえ
3D レンダラー	2D 版 vSphere Client、vSGA、または、vDGA デスクトップ、および vGPU デスクトップ版 NVIDIA GRID vGPU を使用して管理します。

**注:** プールの設定は必須です。設定していない場合、デスクトップへの接続が失敗して、プロトコル エラーやブラック スクリーンが生じる場合があります。

- 8 [vCenter 仮想マシンを追加] ページで、[Linux 仮想マシン] を選択します。[追加] をクリックし、[次へ] をクリックします。

**注:** 同じデスクトップ プールに、Windows と Linux の仮想マシンを作成しないでください。

- 9 手順を完了するには、プロンプトの指示に従います。

デスクトップ プールにあるマシンへの資格をユーザーに付与します。Horizon Administrator で、[デスクトップ プール] を選択し、[資格] > [資格を追加] の順に選択して、ユーザーまたはグループを追加します。

Linux 仮想マシンを、Horizon 7 を展開した環境でリモート デスクトップとして使用する準備が整いました。

## Linux デスクトップ プールの管理

手動デスクトップ プールを作成し、プールに Linux マシンを追加すると、設定構成により手動デスクトップ プールを管理できます。手動デスクトップ プールには、Linux ゲスト オペレーティング システムだけを追加する必要があります。

ります。プールに Windows と Linux ゲスト OS の両方が含まれる場合、プールは Windows プールとして扱われ、Linux デスクトップに接続できません。

## 操作管理のサポート

- デスクトップ プールの無効化または有効化
- 自動デスクトップ プールのクローン作成
- デスクトップ プールの削除

Horizon 7 から仮想マシンを取り外したり、ディスクから仮想マシンを削除できます。

## リモート設定のサポート

表 7-1. リモート設定

リモート設定	オプション
リモート マシンの電源ポリシー	<ul style="list-style-type: none"> <li>■ 電源操作を行わない</li> <li>■ マシンは常にパワーオン</li> <li>■ サスペンド</li> <li>■ パワーオフ</li> </ul>
切断後に自動的にログオフ	<ul style="list-style-type: none"> <li>■ 直ちに実行</li> <li>■ 実行しない</li> <li>■ n 分後に</li> </ul>
ユーザーによるマシンのリセット/再起動を許可	<ul style="list-style-type: none"> <li>■ はい</li> <li>■ いいえ</li> </ul>
ユーザーが複数のクライアント デバイスからセッションを個別に開始できるようにする	<ul style="list-style-type: none"> <li>■ はい</li> <li>■ いいえ</li> </ul>
完全なクローンがある自動デスクトップ プールおよびフローティング デスクトップ プールで「ログオフ後にマシンを削除」	<ul style="list-style-type: none"> <li>■ はい</li> <li>■ いいえ</li> </ul>

## Horizon Administrator 操作のサポート

- セッションを切断
- セッションのログアウト
- デスクトップをリセット/再起動
- メッセージを送信

専用デスクトップ プールについては、各仮想マシンのユーザー割り当てを追加したり削除できます。多くの操作を行うには、Horizon PowerCLI コマンドレットを使用する必要があります。

- Update-UserOwnership

## ■ Remove-UserOwnership

**注:** [リモート表示プロトコル] 設定は変更しないでください。この設定は、常にデスクトップ プール作成と同じ必要があります。

設定	オプション
デフォルト表示プロトコル	VMware Blast
ユーザーがプロトコルを選択できるようにする	いいえ
3D レンダラー	<ul style="list-style-type: none"> <li>■ 2D、vSGA、または vDGA の vSphere Client を使用して管理する</li> <li>■ NVIDIA GRID vGPU</li> </ul>

詳細については、『Horizon 7 の管理』ドキュメントを参照してください。

## Linux の自動化された完全なクローン デスクトップ プールの作成

Linux 仮想マシンの自動化された完全なクローン デスクトップ プールを作成できます。自動化された完全なクローン デスクトップ プールを作成した後は、Horizon 7 のデプロイ環境で Linux 仮想マシンをリモート デスクトップとして使用できます。

### 前提条件

- Horizon Agent が Linux ゲスト OS にインストールされていることを確認します。[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- 仮想マシンのクローンを作成する前に、クローンの基準となる仮想マシン テンプレートを作成します。[Linux デスクトップ マシンのクローンを作成するために仮想マシン テンプレートを作成する](#)を参照してください。
- Winbind ソリューションを使用して、Linux 仮想マシンを Active Directory に参加させる場合、仮想マシン テンプレートで Winbind ソリューションの構成を完了する必要があります。
- Winbind ソリューションを使用する場合、仮想マシンでドメインに参加するためのコマンドを実行する必要があります。シェル スクリプトにこのコマンドを追加して、`/etc/vmware/viewagent-custom.conf` にある Horizon Agent のオプション `RunOnceScript` にこのスクリプトのパスを指定します。詳細については、[Linux デスクトップでの構成ファイルのオプション設定](#)を参照してください。
- vCenter Server が Horizon 接続サーバに追加されていることを確認します。

### 手順

- 1 ゲスト カスタマイズの仕様を作成します。

vSphere 仮想マシン管理ドキュメントの「vSphere Web Client での Linux 向けカスタマイズ仕様の作成」を参照してください。仕様を作成する場合、次の設定を必ず正しく指定してください。

設定	値
ターゲット仮想マシンの OS	Linux
コンピュータ名	仮想マシン名を使用します。
ドメイン	Horizon 7 環境のドメインを指定します。

設定	値
ネットワーク設定	標準ネットワーク設定を使用します。
プライマリ DNS	有効なアドレスを指定します。

**注:** ゲスト OS のカスタマイズのサポート一覧の詳細については、<http://partnerweb.vmware.com/programs/guestOS/guest-os-customization-matrix.pdf> を参照してください。

- Horizon Administrator で、[カタログ] - [デスクトップ ツール] - [追加] を選択します。
- [自動化されたデスクトップ プール] を選択して、[次へ] をクリックします。
- デスクトップ プールにあるマシンのユーザー割り当てについて「専用」または「フローティング」のいずれかを選択して、[次へ] をクリックします。
- [vCenter Server] ページで、[フル仮想マシン] を選択し、適切な vCenter Server を選択して、[次へ] をクリックします。
- [デスクトップ プール ID] ページで、デスクトップ プール ID を入力して、[次へ] をクリックします。
- [デスクトップ プールの設定] ページで、次のオプションを設定して、[次へ] をクリックします。

オプション	説明
デフォルト表示プロトコル	VMware Blast
ユーザーがプロトコルを選択できるようにする	いいえ
3D レンダラー	2D 版 vSphere Client、vSGA、または、vDGA デスクトップ、および vGPU デスクトップ版 NVIDIA GRID vGPU を使用して管理します。

**注:** プールの設定は必須です。設定していない場合、デスクトップへの接続が失敗して、プロトコル エラーやブラック スクリーンが生じる場合があります。

- [プロビジョニングの設定] ページで、[仮想マシンの名前付け] オプションを設定して、[次へ] をクリックします。

オプション	説明
名前を手動で指定	名前を手動で入力します。
名前付けパターンを使用	たとえば、LinuxVM-{n} のように指定します。 次のデスクトップ プールのサイズ設定オプションを指定する必要があります。 <ul style="list-style-type: none"> <li>■ マシンの最大数</li> <li>■ スペアのパワーオン状態のマシンの数</li> </ul>

- [ストレージの最適化] ページで、ストレージ管理のポリシーを選択して、[次へ] をクリックします。
- [vCenter 設定] ページで、[参照] をクリックし、vCenter Server の設定を順番に選択し、[次へ] をクリックする必要があります。

vCenter Server の設定を省略することはできません。

- テンプレート

- b 仮想マシンのフォルダの場所
- c ホストまたはクラスタ
- d リソース プール
- e データストア

- 11 [詳細ストレージ オプション] ページで、適切なストレージ オプションを選択して、[次へ] をクリックします。
- 12 [ゲストのカスタマイズ] ページで、Linux のゲスト カスタマイズを選択して、[次へ] をクリックします。
- 13 [設定内容の確認] ページで、詳細を確認して、[このウィザードの終了後にユーザーに資格を割り当てる] を選択します。
- 14 [終了] をクリックします。
- 15 デスクトップ プールにあるマシンへの資格をユーザーに付与するには、デスクトップ プールを選択して、[資格] - [資格を追加] をクリックして、ユーザーとグループを追加します。
- 16 デスクトップ プールのすべての Linux 仮想マシンが利用可能になるまで待機します。

## Linux のインスタント クローン フローティング デスクトップ プールの作成

[デスクトップ プールの追加]ウィザードを使用すると、Linux 仮想マシンにインスタント クローン フローティング デスクトップ プールを作成できます。インスタント クローン フローティング デスクトップ プールの作成後、Horizon 7 環境内でリモート デスクトップとして Linux 仮想マシンを使用できます。

### 前提条件

- vCenter Server で仮想マシンを作成し、Linux オペレーティング システムをインストールする手順について理解しておきます。詳細については、[仮想マシンを作成して、Linux をインストールする](#)を参照してください。
- PBISO 認証ソリューションで Active Directory を統合する手順について理解しておきます。詳細については、[Linux と Active Directory の統合](#)の「PBISO (PowerBroker Identity Services Open) 認証」を参照してください。
- Horizon 7 Agent for Linux のインストール手順について理解しておきます。詳細については、[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。
- VMware vSphere® Web Client を使用して Linux 仮想マシンのパワーオフ状態のスナップショットを作成する手順について理解しておきます。『vSphere 単一ホスト管理 : VMware Host Client』で「VMware Host Client のスナップショットの取得」を参照してください。
- vCenter Server が Horizon Connection Server に追加されていることを確認します。

### 手順

- 1 Ubuntu 18.04/16.04 または RHEL 7.1 以降がインストールされている Linux 仮想マシン (VM) を作成します。

---

**注:** この機能は、Ubuntu 18.04/16.04 または RHEL 7.1 以降のシステムでのみサポートされています。

---

詳細については、[仮想マシンを作成して、Linux をインストールする](#)を参照してください。

- 2 次のコマンドを使用して、Ubuntu 18.04/16.04 のマシンに Open VMware Tools (OVT) を手動でインストールします。

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

詳細については、[リモート デスクトップ デプロイ用の Linux マシンの準備](#)を参照してください。

- 3 Linux ディストリビューションに必要な依存パッケージをすべてインストールします。

詳細については、[Horizon Agent 用依存パッケージのインストール](#)を参照してください。

- 4 Linux 仮想マシンに Horizon Agent for Linux をインストールします。

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

詳細については、[Linux 仮想マシンへの Horizon Agent のインストール](#)を参照してください。

- 5 Linux 仮想マシンを Active Directory と統合します。

- PBISO 認証ソリューションを使用するには、次の手順に従います。

- a PBIS Open 8.5.6 以降を <https://www.beyondtrust.com/products/powerbroker-identity-services-open/> からダウンロードして、Linux 仮想マシンにインストールします。

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

- b [Linux と Active Directory の統合](#)の PBISO (PowerBroker Identity Services Open) 認証セクションの情報をを使用して Linux 仮想マシンを Active Directory と統合します。

- Linux 仮想マシンを Active Directory と統合する際に、PBISO を使用しない場合は、`/etc/vmware/viewagent-custom.conf` ファイル内の `OfflineJoinDomain` オプションを **none** に設定してください。オプションを設定しないと、インスタント クローンのフローティング デスクトップ プールの作成に失敗します。

- 6 DHCP サーバが DNS サーバにブロードキャストしない場合には、Linux システムの DNS サーバを指定します。

新しいインスタント クローン仮想マシンを作成すると、新しい仮想ネットワーク アダプタが追加されます。インスタント クローン仮想マシンに新しいネットワーク アダプタを追加すると、仮想マシン テンプレート内の DNS サーバなどのネットワーク アダプタのすべての設定が失われます。PBIS には有効な DNS サーバが必要で、`/etc/hosts` で FQDN マッピングは使用できません。クローン作成された仮想マシンに新しいネットワーク アダプタを追加したときに DNS サーバ設定が失われないようにするには、Linux システムで DNS サーバを指定する必要があります。たとえば、Ubuntu 16.04 システムでは、`/etc/resolvconf/resolv.conf.d/head` ファイルに次の行を追加して、DNS サーバを指定します。

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

- 7 (オプション) マスターの Linux VDI インスタント クローン エージェントの `/etc/fstab` ファイルに NFS マウントを追加する場合は、次のいずれかの方法を使用します。

- 次のように、`/etc/fstab` に「soft」フラグを追加します。

```
10.111.222.333:/share    /home/nfsmount    nfs
size=8192,wsiz=8192,timeo=14,soft,intr,tcp
```

- `/etc/fstab` で「soft」フラグを使用しない場合は、Linux 仮想マシンのマスタ イメージで、`/etc/fstab` を設定できません。`/etc/fstab` ファイルを設定するパワーオフ スクリプトを作成して、Horizon Administrator でインスタント クローン デスクトップ プールのゲスト カスタマイズ ペインでこのパワーオフ スクリプトを指定します。
- 8 Linux 仮想マシンをシャットダウンし、VMware vSphere® Web Client を使用してパワーオフ状態の Linux 仮想マシンのスナップショットを作成し、マスター イメージを作成します。
- 詳細については、『vSphere 単一ホスト管理 : VMware Host Client』で「VMware Host Client のスナップショットの取得」を参照してください。
- 9 Horizon Administrator で、[カタログ]>[デスクトップ ピール]>[追加] の順に選択します。
- 10 デスクトップ プールを定義します。
- [タイプ] ペインで、[自動デスクトップ プール] を選択して、[次へ] をクリックします。
  - [ユーザーの割り当て] ペインで、[フローティング] を選択します。[次へ] をクリックします。
  - vCenter Server ペインで [インスタント クローン] を選択します。
- 11 [デスクトップ プール ID] ペインでデスクトップ プール ID を入力し、[次へ] をクリックします。
- 12 [デスクトップ プールの設定] ペインで、デフォルトの設定を受け入れ、HTML Access を有効にします。[次へ] をクリックします。
- 13 [プロビジョニングの設定] ペインで次の [仮想マシンの名前付け] オプションを設定し、[次へ] をクリックします。
- [プロビジョニングを有効にする] と [エラーによりプロビジョニングを停止] を選択します。
  - [名前付けパターン] テキスト ボックスに、すべてのデスクトップ仮想マシン名で Horizon 7 がプリフィックスとして使用するパターンとそれに続く一意の番号を指定します。たとえば、**LinuxVM-{n}** と指定します。
  - [マシンの最大数] テキスト ボックスに、プール内のマシンの合計数を指定します。
  - [スベアの (パワーオン状態の) マシンの数] テキスト ボックスに、ユーザーから利用可能な状態を保つデスクトップ仮想マシンの数を指定します。
  - [マシンの最大数] に指定した数の仮想マシンを Horizon 7 がプロビジョニングするように、[全マシンを事前にプロビジョニング] を選択します。
- 14 [ストレージの最適化] ペインでストレージ管理ポリシーに [VMware Virtual SAN を使用する] を選択して、[次へ] をクリックします。



- 15 [vCenter Server の設定] ペインで、各 vCenter Server 設定の [参照] をクリックして適切な値を選択し、[次へ] をクリックします。
- 16 [ゲストのカスタマイズ] ペインで、ドメインの設定、Active Directory コンテナ、仮想マシンのクローン作成後に実行する必要があるカスタム スクリプトを指定します。[次へ] をクリックします。

---

**重要:** ClonePrep パワーオフまたは同期後スクリプトを使用する場合は、スクリプトが /var/userScript フォルダにあり、root ユーザーによって所有され、ファイル権限が 700 に設定されていることを確認します。

---

- 17 [設定内容の確認] ペインで、[このウィザードの終了後にユーザーに資格を割り当てる] を選択して、詳細を確認します。
- 18 [終了] をクリックします。

Horizon Administrator で、[カタログ] > [デスクトップ プール] の順に選択すると、プールに追加されているとおりデスクトップ仮想マシンを表示できます。

プールの作成後、プールが存在しているときに vCenter Server インベントリからマスター イメージを削除したり、取り除いたりしないでください。vCenter Server のインベントリからマスター イメージの仮想マシンを誤って取り除いてしまった場合は、改めて追加し、現在のイメージを使用してプッシュ イメージを実行する必要があります。

#### 次のステップ

プールにアクセスするための資格をユーザーに付与します。『Horizon 7 での仮想デスクトップのセットアップ』の「デスクトップ プールへの資格の追加」を参照してください。

## ブローカ PowerCLI コマンド

接続サーバや Windows デスクトップでさまざまな管理タスクを実行する Horizon PowerCLI cmdlets は、Linux デスクトップで使用できます。

### 手動デスクトップ プールの作成

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

Linux デスクトップの場合、次のオプションと値は必須です。

- DefaultProtocol Blast
- AllowProtocolOverride \$false
- threadRender usevc|vgpu。vGPU デスクトップの場合は -threadRender vgpu を使用します。2D/vSGA/vDGA デスクトップの場合は -threadRender usevc を使用します。

## [例]

- 仮想マシン (VM) LinuxVM-01 で、LinuxDesktop という名のフローティングの Linux デスクトップ プールを作成します。

```
Add-ManualPool -DefaultProtocol Blast -AllowProtocolOverride $false -threadRender usevc -Pool_id
LinuxDesktop -Id (Get-DesktopVM -Name LinuxVM-01).id -Persistence NonPersistent -Vc_name
myvc.myorg.org
```

- LinuxVM- という仮想マシン名で始まるすべての仮想マシンを持つ LinuxDesktop という名の専用 Linux vGPU デスクトップ プールを作成します。

```
Get-DesktopVM | Where-Object {$_.Name.StartsWith("LinuxVM-")} | Add-ManualPool -DefaultProtocol
Blast -AllowProtocolOverride $false -Persistence Persistent -threadRender vgpu -Pool_id
LinuxDesktop
```

- 最初の RHEL 6 x64 仮想マシンでフローティング Linux デスクトップ プール LinuxDesktop を作成します。

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

## 完全なクローンの自動化されたデスクトップ プールを作成

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

Linux デスクトップの場合、次のオプションと値は必須です。

- DefaultProtocol Blast
- AllowProtocolOverride \$false
- threadRender usevc|vgpu vGPU デスクトップの場合は -threadRender vgpu を使用します。2D/VGA デスクトップの場合は -threadRender usevc を使用します。

## [例]

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

```
-templatePath "/LinuxVDI/vm/LinuxTemplate" `
-dataStorePaths "/LinuxVDI/host/LinuxVDICluster/datastore" `
-customizationSpecName "linux-spec" `
-maximumCount 100
```

## デスクトップ プールの資格を追加または削除

- LinuxDesktop に、ドメイン mydomain.org のドメイン ユーザー グループに付与します。

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

- LinuxDesktop から、mydomain.org ドメインのドメイン ユーザー グループの資格を削除してください。

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

## 専用デスクトップ プールの仮想マシンに、または仮想マシンからユーザーを割り当てるか削除します

- 専用デスクトップ プールにある、LinuxVM-01 仮想マシンに **myuser** ユーザーを割り当てます。

```
Update-UserOwnership -Machine_id (Get-DesktopVM -Name "LinuxVM-01").machine_id -Sid (Get-User -
Name "myuser" | Where-Object {$_.cn -eq "myuser"}).sid
```

- 専用デスクトップ プールにある、LinuxVM-01 仮想マシンから **myuser** ユーザーを削除します。

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

## デスクトップ接続をログオフ

- myuser のデスクトップセッションからログアウトします。

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

ブローカ PowerCLI コマンドレットの詳細については、『Horizon 7 の統合』の「Horizon PowerCLI の 使用」を参照してください。

# 手動デスクトップ プールのための Horizon 7 の一括デプロイ

## 8

Horizon Administrator では、Linux ではなく Windows デスクトップ マシンのプールを自動的に作成できます。ただし、Linux デスクトップ マシンのプールのデプロイを自動化するスクリプトを開発できます。

提供されているサンプル スクリプトは、例示のみを目的としています。ユーザーがサンプル スクリプトを使用するときに発生する可能性がある問題について、VMware は一切責任を負いません。

この章には、次のトピックが含まれています。

- [Linux デスクトップの一括デプロイの概要](#)
- [Linux デスクトップの一括アップグレードの概要](#)
- [Linux デスクトップ マシンのクローンを作成するために仮想マシン テンプレートを作成する](#)
- [Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)
- [Linux 仮想マシンのクローンを作成するサンプル スクリプト](#)
- [クローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト](#)
- [SSH を使用してクローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト](#)
- [Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト](#)
- [SSH を使用して Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト](#)
- [Linux デスクトップ マシンで Horizon Agent をアップグレードするサンプル PowerCLI スクリプト](#)
- [SSH を使用して Linux 仮想マシンで Horizon Agent をアップグレードするサンプル スクリプト](#)
- [Linux 仮想マシンで操作を実行するサンプル スクリプト](#)

## Linux デスクトップの一括デプロイの概要

Linux 版手動デスクトップ一括デプロイには、いくつかの手順があります。多数のデスクトップをデプロイする予定がある場合、PowerCLI スクリプトを使用していくつかの手順を自動化できます。

一部の操作では、Linux マシン上で PowerCLI または SSH によってコマンドを実行することを選択できます。次の表では、2 つの手法の違いについて説明します。

PowerCLI	SSH
追加のツールのインストールは不要。	<ul style="list-style-type: none"> <li>■ Ubuntu の場合は、コマンド <code>sudo apt-get install openssh-server</code> を使用して SSH サーバをインストールする必要がある。RHEL と CentOS の場合は、<code>openssh-server</code> がデフォルトでインストールされているが、ファイアウォール設定で <code>ssh</code> が許可されていることを確認する必要がある。</li> <li>■ SSH クライアントアプリケーションの <code>pscp.exe</code> と <code>plink.exe</code> をダウンロードし、PowerCLI スクリプトと同じフォルダに配置する必要がある。</li> </ul>
ファイルのアップロードとコマンド実行は遅い。	ファイルのアップロードとコマンド実行は速い。
ESXi ホストの管理者認証情報を入力する必要がある。	ESXi ホストの管理者認証情報を入力する必要はない。
管理者パスワード（スクリプトを実行して Horizon Agent をインストールする場合）や Active Directory ユーザーのパスワード（スクリプトを実行してドメインに参加する場合）内の特殊文字を処理できない。	管理者パスワード（スクリプトを実行して Horizon Agent をインストールする場合）や Active Directory ユーザーのパスワード（スクリプトを実行してドメインに参加する場合）内の特殊文字を処理できる。

**注:** PowerCLI ベースのスクリプトおよび SSH ベースのスクリプトは、vCenter Server 管理者と Linux 管理者のパスワード内の特殊文字を処理できます。PowerCLI ベースのスクリプトは ESXi ホスト管理者のパスワード内の特殊文字も処理できます。これらのいずれの場合もエスケープ文字は不要です。

vSphere PowerCLI の詳細については、<https://www.vmware.com/support/developer/PowerCLI> を参照してください。

Linux デスクトップ プールの一括デプロイプロセスでは、次の手順を実行します。

- 1 仮想マシン テンプレートを作成し、仮想マシンで Horizon Agent をインストールします。

[Linux デスクトップ マシンのクローンを作成するために仮想マシン テンプレートを作成する](#)を参照してください。

- 2 ゲスト カスタマイズの仕様を作成します。

vSphere 仮想マシン管理ドキュメントの「vSphere Web Client での Linux 向けカスタマイズ仕様の作成」を参照してください。仕様を作成する場合、次の設定を必ず正しく指定してください。

設定	値
ターゲット仮想マシンの OS	Linux
コンピュータ名	仮想マシン名を使用します。
ドメイン	Horizon 7 環境のドメインを指定します。
ネットワーク設定	標準ネットワーク設定を使用します。
プライマリ DNS	有効なアドレスを指定します。

**注:** ゲスト OS のカスタマイズのサポート一覧の詳細については、<http://partnerweb.vmware.com/programs/guestOS/guest-os-customization-matrix.pdf> を参照してください。

- 3 仮想マシンのクローンを作成します。

[Linux 仮想マシンのクローンを作成するサンプル スクリプト](#)を参照してください。

- 4 winbind ソリューションを使用している場合は、クローン作成された仮想マシンを Active Directory (AD) ドメインに参加させます。下の例スクリプトでドメイン参加コマンドを実行したり、テンプレート仮想マシンで構成された `/etc/vmware/viewagent-custom.conf` 内でオプション `RunOnceScript` を使用できます。

クローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプトまたは SSH を使用してクローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプトを参照してください。

- 5 仮想マシンの構成オプションを更新します。

Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプトまたは SSH を使用して Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプトを参照してください。

- 6 デスクトップ プールを作成します。

Linux 版手動デスクトップ プールの作成を参照してください。

仮想マシンのパワーオン、シャットダウン、再起動、または削除などの操作を実行するサンプル スクリプトについては、Linux 仮想マシンで操作を実行するサンプル スクリプトを参照してください。このスクリプトを使用して、vCenter Server から仮想マシンを削除できます。

## Linux デスクトップの一括アップグレードの概要

Linux 版手動デスクトップの一括アップグレードには、いくつかの手順があります。PowerCLI スクリプトの使用により手順のいくつかを自動化できます。

### 非管理対象デスクトップの一括アップグレード

非管理対象仮想マシンを、管理対象または非管理対象仮想マシンへ一括アップグレードするには、サンプルのアップグレード スクリプトを使用して新しい Horizon Agent を既存の仮想マシンにアップロードし、アップグレード コマンドを実行する必要があります。

- 非管理対象仮想マシンを保存すると、既存のデスクトップ プールを再使用できます。
- 非管理対象仮想マシンから管理対象仮想マシンへアップグレードする場合は、既存のデスクトップ プールを削除し、新しいデスクトップ プールを作成する必要があります。詳細については、Linux 仮想マシンでの Horizon Agent のアップグレードを参照してください。

### 管理対象デスクトップの一括アップグレード

管理対象仮想マシンを一括アップグレードするには、次の方法のいずれかを選択します。

方法	説明
テンプレート仮想マシンで、新しい Horizon Agent をインストールするかアップグレードして、スナップショットを作成してください。	<ul style="list-style-type: none"> <li>■ ユーザー データとプロファイルが NFS サーバのような共有サーバに置かれられない限り、既存の仮想マシンが削除されるため、ユーザー データとプロファイルが失われます。</li> <li>■ 仮想マシンの交換後は、View Administrator の仮想マシンの状態が見つからない場合があります。これを修復するにはブローカー サービスを再起動する必要があります。</li> <li>■ リンク クローンを使用すると、この方法は各仮想マシンでのデータの重複を回避できます。</li> </ul>
アップグレードのサンプル スクリプトを使用して、新しい Horizon Agent を既存の仮想マシンにアップロードを行い、アップグレード コマンドを実行してください。	<ul style="list-style-type: none"> <li>■ ユーザー データとプロファイルが保存されます。</li> <li>■ リンク クローンを使用すると、この方法は各仮想マシンでのデータの重複を招きます。</li> </ul>

## Linux デスクトップ マシンのクローンを作成するために仮想マシン テンプレートを作成する

仮想マシンのクローンを作成する前に、クローンの基準となる仮想マシン テンプレートを作成する必要があります。

### 前提条件

- デプロイする環境がサポートする Linux デスクトップの要件を満たしていることを確認します。[Horizon 7 for Linux のシステム要件](#)を参照してください。
- vCenter Server で仮想マシンを作成し、ゲスト OS をインストールする手順について理解しておきます。Horizon 7 での仮想デスクトップのセットアップ ドキュメントの「仮想マシンの作成および準備」を参照してください。
- 仮想マシンで使用するモニターに必要なビデオ メモリ (vRAM) の値を理解しておきます。[Horizon 7 for Linux のシステム要件](#)を参照してください。
- Active Directory 統合の手順について理解しておきます。[3 章 Linux デスクトップの Active Directory 統合のセットアップ](#)を参照してください。
- Linux で Horizon Agent をインストールする手順をよく理解しておいてください。[5 章 Horizon Agent のインストール](#)を参照してください。
- 必要な場合は、Horizon 7 構成ファイルを使用してオプションを構成する手順について理解しておきます。[6 章 Linux デスクトップの構成オプション](#)を参照してください。
- グラフィックスをセットアップする予定がある場合は、その手順について理解しておきます。[4 章 Linux デスクトップのグラフィックスのセットアップ](#)を参照してください。

### 手順

- 1 vSphere Web Client または vSphere Client で新しい仮想マシンを作成します。

## 2 カスタム構成オプションを構成します。

- a 仮想マシンを右クリックし、[設定の編集] をクリックします。
- b vCPU の数と vMemory のサイズを指定します。

お使いの Linux ディストリビューションのインストール ガイドに記載されている vCPU と vMemory サイズのガイドラインに従ってください。

たとえば、Ubuntu 18.04 では、2048 MB の vMemory と 2 台の vCPU を構成することが指定されています。

- c [ビデオ カード] を選択して、ディスプレイの数とビデオ メモリ (vRAM) の合計を指定します。

VMware のドライバを使用し、2D や vSGA を使用する仮想マシンについては、vSphere Web Client で vRAM のサイズを設定します。vRAM のサイズは、NVIDIA のドライバを使用する vDGA や NVIDIA GRID vGPU マシンには影響しません。

Horizon 7 for Linux のシステム要件にあるガイドラインに従ってください。ビデオ メモリ計算ツールは使用しないでください。

- 3 仮想マシンをパワーオンして、Linux ディストリビューションをインストールします。
- 4 たとえば、ViewUser など root 権限のあるユーザーを作成します。このユーザーは、Horizon Agent のインストールとアンインストールにのみ使用されます。
- 5 `/etc/sudoers` を編集して、行 `ViewUser ALL=(ALL) NOPASSWD:ALL` を追加します。

`/etc/sudoers` のこの行では、ViewUser として `sudo` を実行するためにパスワードは必要ありません。この章で説明しているようにサンプル スクリプトを実行して Horizon Agent をインストールする場合、入力として ViewUser を指定します。

- 6 Linux ディストリビューションが RHEL、CentOS、または NeoKylin である場合、`/etc/sudoers` を編集して、次の行をコメントアウトします。

```
Defaults requiretty
Defaults !visiblepw
```

- 7 Linux ディストリビューションが RHEL 7、CentOS 7、SLED 12、または SLES 12 ではない場合、VMware Tools をインストールします。

RHEL 7、CentOS 7、SLED 12、および SLES 12 には、デフォルトで >Open VM Tools がインストールされています。



## 8 依存パッケージをインストールして設定します。

- a Linux ディストリビューションが RHEL 7、CentOS 7、または SLES 12 の場合、deployPkg プラグインをインストールします。

操作手順については、<http://kb.vmware.com/kb/2075048> を参照してください。

- b Linux ディストリビューションが Ubuntu の場合、次のナレッジベースの記事を参照して、仮想マシンにインストールして設定する依存パッケージを確認してください。

- Ubuntu 18.04 および 16.04 の場合は、ナレッジベースの記事 <https://kb.vmware.com/s/article/2051469> および <https://kb.vmware.com/s/article/59687> を参照してください。
- Ubuntu 18.04 の場合は、ナレッジベースの記事 <https://kb.vmware.com/s/article/56409> も参照してください。

## 9 RHEL と CentOS の場合は、[ネットワーク接続] 設定の [自動接続] を有効にします。

## 10 Active Directory 統合タスクを実行します。

## 11 グラフィックスをセットアップする手順を実行します。

## 12 Horizon Agent をインストールします。

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

5 章 [Horizon Agent のインストール](#) を参照してください。

## 13 Horizon 7 構成ファイルを使用して追加構成を実行します。

## 14 仮想マシンをシャットダウンして、スナップショットを作成します。

# Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル

Linux デスクトップを展開するサンプル PowerCLI スクリプトは、デスクトップ マシンに関する情報を含む 1 つの入力ファイルを読み取ります。

入力ファイルのタイプは csv であり、次の情報を含みます。

- デスクトップ仮想マシンの名前
- 親仮想マシンの名前
- ゲスト カスタマイズの仕様
- クローン作成されたデスクトップ マシンが存在するデータストア
- デスクトップ マシンをホストする ESXi サーバ
- クローン作成に使用される親仮想マシンのスナップショット
- 存在している場合、デスクトップ仮想マシンを削除するかどうかを示すフラグ

次の例は、入力ファイルに含まれる可能性がある情報を示しています。

```
VMName,Parentvm,CustomSpec,Datastore,Host,FromSnapshot,DeleteIfPresent
linux-001,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-002,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-003,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-004,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-005,Ubuntu1804x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
```

サンプル スクリプトでは、入力ファイルの名前が CloneVMs.csv であり、このファイルがスクリプトと同じフォルダにある配置されていることを前提としています。

## Linux 仮想マシンのクローンを作成するサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、任意の数の仮想マシン (VM) のクローンを作成できます。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は <https://docs.vmware.com/jp/VMware-Horizon-7/index.html> にある Horizon 7 のドキュメントのページから入手できます。

### スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- クローンのタイプ。リンク クローンまたは完全クローンになります
- vSphere 仮想マシン コンソールを無効にするかどうか

### スクリプトのコンテンツ

```
<#
Create Clones from a Master VM

The Tool supports creation of Full clone and linked clone from Master VM.
The parent VM is required for the linked-clone to work and the parent VMs file cannot be renamed or
moved.
#>
#----- Functions -----
function GetInput
{
    Param($prompt, $IsPassword = $false)
    $prompt = $prompt + ": "
    Write-Host $prompt -NoNewLine
    [Console]::ForegroundColor = "Blue"
    if ($IsPassword)
    {
```

```

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

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

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

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

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

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

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

$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false

```

```

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

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

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

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

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

    $destVMName=$line.VMName
    $srcVM = $line.Parentvm
    $cSpec = $line.CustomSpec
    $targetDSName = $line.Datastore

```

```

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

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

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```
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
```

クローン作成プロセスにかかる時間は、デスクトップ マシンの数によって異なり、数分から数時間になります。プロセスが完了したことを確認するには、vSphere Client で、最後のデスクトップ仮想マシンがパワーオンされていること、一意のホスト名が付いていること、VMware Tools が実行されていることを確認します。

## クローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、クローン作成した仮想マシン (VM) を Active Directory (AD) ドメインに参加させることができます。

Active Directory 統合に Winbind ソリューションを使用する場合は、このスクリプトを実行する必要があります。クローン作成した仮想マシンでは、ドメインに参加する手順がエラーになるためです。このスクリプトでは、ドメインに参加するためのコマンドが各仮想マシンで実行されます。OpenLDAP ソリューションを使用する場合、このスクリプトを実行する必要はありません。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

## スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- ESXi ホスト管理者のログイン名
- ESXi ホスト管理者のパスワード
- Linux 仮想マシンのユーザー ログイン名
- Linux 仮想マシンのユーザー パスワード
- マシンをドメインに参加させる許可を受けた Active Directory ユーザーのログイン名
- 許可された Active Directory ユーザーのパスワード

## スクリプトのコンテンツ

```
<#
.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
```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```
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: *****
```

## SSH を使用してクローン作成した仮想マシンを Active Directory ドメインに参加させるサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、クローン作成した仮想マシン (VM) を Active Directory (AD) ドメインに参加させることができます。このスクリプトでは SSH を使用して、Linux 仮想マシンでコマンドを実行します。

Active Directory 統合に Winbind ソリューションを使用する場合は、このスクリプトを実行する必要があります。クローン作成した仮想マシンでは、ドメインに参加する手順がエラーになるためです。このスクリプトでは、ドメインに参加するためのコマンドが各仮想マシンで実行されます。OpenLDAP ソリューションを使用する場合、このスクリプトを実行する必要はありません。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

### スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- Linux 仮想マシンのユーザー ログイン名
- Linux 仮想マシンのユーザー パスワード
- マシンをドメインに参加させる許可を受けた Active Directory ユーザーのログイン名
- 許可された Active Directory ユーザーのパスワード

### スクリプトのコンテンツ

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

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

.NOTES
#>
#----- Functions -----
```

```

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

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

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

function RunCmdViaSSH
{

```

```

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

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

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

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

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

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

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

#Connect to vCenter

```

```

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

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

$destFolder = "/home/$guestUser/"

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

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

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```

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: *****

```

## Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、構成ファイル config と viewagent-custom.conf を複数の Linux 仮想マシン (VM) にアップロードできます。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

### スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- ESXi ホスト管理者のログイン名
- ESXi ホスト管理者のパスワード
- Linux 仮想マシンのユーザー ログイン名
- Linux 仮想マシンのユーザー パスワード

### スクリプトのコンテンツ

```
<#
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

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```

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: *****

```

## SSH を使用して Linux 仮想マシンに構成ファイルをアップロードするサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、構成ファイル config と viewagent-custom.conf を複数の Linux 仮想マシン (VM) にアップロードできます。このスクリプトでは SSH を使用して、Linux 仮想マシンでコマンドを実行します。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

## スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- Linux 仮想マシンのユーザー ログイン名
- Linux 仮想マシンのユーザー パスワード



## スクリプトのコンテンツ

```
<#
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

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```

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: *****

```

## Linux デスクトップ マシンで Horizon Agent をアップグレードする サンプル PowerCLI スクリプト

次のサンプル スクリプトをカスタマイズして使用し、複数の Linux 仮想マシン (VM) で Horizon Agent をアップグレードできます。

このスクリプトでは、Horizon Agent のインストール前に、各仮想マシンにインストーラの tar ボールをアップロードします。アップロード タスクは、多くの仮想マシンが含まれ、ネットワークのスピードが遅い場合は特に時間がかかることがあります。時間を節約するには、SSH を使用するスクリプトを実行するか、インストーラの tar ボールを共有場所に配置して各仮想マシンで使用できるようにして、ファイルのアップロードを不要にすることができます。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は <https://docs.vmware.com/jp/VMware-Horizon-7/index.html> にある Horizon 7 のドキュメントのページから入手できます。

## スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- Horizon Agent EULA（エンドユーザー使用許諾契約書）の承諾
- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- ESXi ホスト管理者のログイン名
- ESXi ホスト管理者のパスワード
- Linux ゲスト OS のユーザー ログイン名
- Linux ゲスト OS のユーザー パスワード
- Horizon Agent tar ボールのパス
- 管理対象の仮想マシンへアップグレード
- スマートカード リダイレクト機能をインストール

## スクリプトのコンテンツ

```
<#
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' -IsPassword $false
"-----"
$UpgradeToManagedVM = GetInput -prompt 'Upgrade to managed VM ("yes" or "no")' -IsPassword $false
if (($UpgradeToManagedVM -ne "yes") -AND $UpgradeToManagedVM -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"-----"

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

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

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

```

```

write-host -ForegroundColor Red "CSV File not found"
exit
}
#-----
Functions-----
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;
    $md5Hash = ($md5HashWithFormat.replace("-", "")).ToLower();
    return $md5Hash;
}

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

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

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

$destFolder = "/home/$guestUser/"

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

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

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

```

```

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

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

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

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

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

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

```



## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

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

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

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

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

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

-----
Upgrade to managed VM ("yes" or "no"): yes
Install the Smartcard redirection feature ("yes" or "no"): no
```

## SSH を使用して Linux 仮想マシンで Horizon Agent をアップグレードするサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、複数の Linux 仮想マシン (VM) で Horizon Agent をアップグレードできます。このスクリプトでは SSH を使用して、Linux 仮想マシンでコマンドを実行します。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

## スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- Horizon Agent EULA（エンド ユーザー使用許諾契約書）の承諾
- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- ESXi ホスト管理者のログイン名
- ESXi ホスト管理者のパスワード
- Linux ゲスト OS のユーザー ログイン名
- Linux ゲスト OS のユーザー パスワード

- Horizon Agent の tar ボールのパス
- 管理対象の仮想マシンへアップグレード
- スマートカード リダイレクト機能をインストール

## スクリプトのコンテンツ

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

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

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

```

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

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

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

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

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

#-----Handle
input-----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $true
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux View Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForegroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}

```

```

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

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

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

#check if file exists
if (!(Test-Path $csvFile))
{
    write-host -ForegroundColor Red "CSV File not found"
    exit
}
#-----
Functions-----
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;
    $md5Hash = ($md5HashWithFormat.replace("-", "")).ToLower();
    return $md5Hash;
}

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

```

```

#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

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

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

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

```

```

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

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

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```

PowerCLI C:\scripts> .\InstallAgent.ps1

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

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

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

-----
-----
Upgrade to managed VM ("yes" or "no"): yes
Install the Smartcard redirection feature ("yes" or "no"): no

```

## Linux 仮想マシンで操作を実行するサンプル スクリプト

次のサンプル スクリプトをカスタマイズして使用し、複数の Linux 仮想マシン (VM) で操作を実行できます。操作には、仮想マシンのパワーオン、パワーオフ、シャットダウン、および削除が含まれます。

このスクリプトによって、vCenter Server から仮想マシンを削除できますが、View からは削除できません。

改ページせずにスクリプトの内容をコピーして貼り付けるには、このトピックの HTML 版を使用します。HTML 版は [https://www.vmware.com/support/pubs/view\\_pubs.html](https://www.vmware.com/support/pubs/view_pubs.html) にある Horizon 7 のドキュメントのページから入手できます。

## スクリプト入力

このスクリプトは、[Linux デスクトップを展開するサンプル PowerCLI スクリプトの入力ファイル](#)で説明しているように 1 つの入力ファイルを読み取ります。また、このスクリプトは、次の情報をインタラクティブに確認します。

- vCenter Server の IP アドレス
- vCenter Server 管理者のログイン名
- vCenter Server 管理者のパスワード
- 実行するアクション。パワーオン、パワーオフ、ゲストのシャットダウン、仮想マシンの再起動、仮想マシンゲストの再起動、仮想マシンの削除のいずれかです。
- 仮想マシンでの操作間の待機時間（秒単位）

## スクリプトのコンテンツ

```
<#
.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

```

## スクリプトの実行

このスクリプトを実行すると、次のメッセージが表示されます。

```

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

```

パワーオン、仮想マシンの再起動、および仮想マシン ゲストの再起動の操作については、仮想マシン間における待機時間に少なくとも 20 秒を指定して、操作が失敗する原因となることがあるブート ストームの状況を回避してください。

# Linux デスクトップのトラブルシューティング

# 9

Linux デスクトップの管理時に特定の問題が発生する可能性があります。問題を診断および解決するためにさまざまな手順を実行できます。

この章には、次のトピックが含まれています。

- [Horizon Console での Horizon Help Desk Tool の使用](#)
- [Horizon 7 for Linux マシンの診断情報の収集](#)
- [リモート デスクトップとクライアント ホストの間でのコピーと貼り付けに関するトラブルシューティング](#)
- [Horizon Agent が iPad Pro Horizon Client で切断できない](#)
- [SLES 12 SP1 デスクトップが自動更新されない](#)
- [SSO がパワーオフ エージェントに接続できない](#)
- [Linux 版手動デスクトップ プール作成後の接続不能な仮想マシン](#)

## Horizon Console での Horizon Help Desk Tool の使用

Horizon Help Desk Tool は、Horizon 7 ユーザー セッションのステータスを取得し、トラブルシューティングとメンテナンス操作を行う Web アプリケーションです。

Horizon Help Desk Tool では、トラブルシューティングを行うためにユーザー セッションを確認し、デスクトップの再起動やリセットなどのデスクトップ メンテナンス操作を実行できます。

Horizon Help Desk Tool を設定するには、次の要件を満たす必要があります。

- Horizon 7 の Horizon Enterprise Edition ライセンスまたは Horizon Apps Advanced Edition ライセンス正しいライセンスがあることを確認するには、『Horizon 7 の管理』ドキュメントを参照してください。
- Horizon 7 コンポーネントの情報を保存するイベント データベースイベント データベースの設定の詳細については、『Horizon 7 のインストール』ドキュメントを参照してください。
- Horizon Help Desk Tool にログインするヘルプデスク管理者ロールまたはヘルプデスク管理者（読み取り専用）ロールこれらのロールの詳細については、『Horizon 7 の管理』ドキュメントを参照してください。
- ログイン セグメントを表示するには、各接続サーバ インスタンスでタイミング プロファイラを有効にします。

各接続サーバ インスタンスでタイミング プロファイラを有効にするには、次の `vdmadmin` コマンドを使用します。

```
vdmadmin -I -timingProfiler -enable
```

管理ポートを使用している接続サーバ インスタンスでタイミング プロファイラを有効にするには、次の `vdmadmin` コマンドを使用します。

```
vdmadmin -I -timingProfiler -enable -server {ip/server}
```

- `/etc/vmware/viewagent-custom.conf` 構成ファイルで `HelpDeskEnable` オプションを有効にします。

## Horizon Console で Horizon Help Desk Tool を開始します。

Horizon Help Desk Tool は、Horizon Console に統合されています。Horizon Help Desk Tool のトラブルシューティングを行うユーザーを検索できます。

### 手順

- 1 Horizon Console で、ユーザーの検索フィールドにユーザー名を入力します。

Horizon Console では、検索結果にユーザーのリストが表示されます。最大で 100 個までの検索結果が返されます。

- 2 ユーザー名を選択します。

ユーザー カードにユーザー情報が表示されます。

### 次のステップ

問題のトラブルシューティングを行うには、ユーザー カードで関連するタブをクリックします。

## Horizon Help Desk Tool でのユーザーのトラブルシューティング

Horizon Help Desk Tool のユーザー カードを使用すると、ユーザーの基本情報を確認できます。ユーザー カードのタブをクリックすると、特定のコンポーネントの詳細が表示されます。

ユーザーの詳細が表に表示されることがあります。これらのユーザーの詳細は、表の列を使って並べ替えることができます。

- 列を昇順で並べ替えるには、列を 1 回クリックします。
- 列を降順で並べ替えるには、列を 2 回クリックします。
- 列を並べ替えない場合は、列を 3 回クリックします。

### ユーザーの基本情報

ユーザーのユーザー名、電話番号、メールアドレス、ユーザーの接続状態などのユーザーの基本情報が表示されます。ユーザーにデスクトップ セッションがある場合、ユーザーは接続状態になります。ユーザーにデスクトップ セッションがない場合、ユーザーは切断状態になります。

メールアドレスをクリックすると、ユーザーにメッセージを送信できます。

## セッション

[セッション] タブには、ユーザーが接続しているデスクトップの情報が表示されます。

[フィルタ] テキスト ボックスを使用すると、デスクトップ セッションをフィルタリングできます。

**注:** [セッション] タブに、vSphere Client または ESXi から仮想マシンにアクセスするセッションの情報は表示されません。

[セッション] タブには、次の情報が表示されます。

表 9-1. [セッション] タブ

オプション	説明
状態	<p>デスクトップ セッションの状態が表示されます。</p> <ul style="list-style-type: none"> <li>■ セッションが接続されている場合、緑色が表示されます。</li> <li>■ セッションがローカル セッションか、ローカルのポッドで実行されているセッションの場合、L が表示されます。</li> </ul>
コンピュータ名	<p>デスクトップ セッションの名前。名前をクリックすると、カードにセッション情報が表示されます。</p> <p>セッション カードでタブをクリックすると、次の追加情報が表示されます。</p> <ul style="list-style-type: none"> <li>■ [詳細] タブには、仮想マシン、CPU またはメモリ使用量などのユーザー情報が表示されます。</li> <li>■ [プロセス] タブには、CPU およびメモリ関連のプロセスに関する情報が表示されます。</li> </ul>
プロトコル	デスクトップ セッションの表示プロトコルを表示します。
Type	デスクトップの種類（公開デスクトップまたは仮想マシン デスクトップ）が表示されます。
接続時間	セッションが接続サーバに接続した時間。
セッションの期間	セッションが接続サーバに接続していた期間。

## デスクトップ

[デスクトップ] タブには、ユーザーに使用資格が付与されている公開デスクトップまたは仮想デスクトップの情報が表示されます。

表 9-2. デスクトップ

オプション	説明
状態	<p>デスクトップ セッションの状態が表示されます。</p> <ul style="list-style-type: none"> <li>■ セッションが接続されている場合、緑色が表示されます。</li> </ul>
デスクトップ プール名	セッションのデスクトップ プールの名前。
デスクトップ タイプ	<p>デスクトップの種類（公開デスクトップまたは仮想マシン デスクトップ）が表示されます。</p> <p><b>注:</b> セッションでポッド フェデレーションの別のポッド実行されている場合、情報は表示されません。</p>

オプション	説明
Type	デスクトップの資格のタイプが表示されます。 ■ ローカル資格の場合には、Local が表示されます。
vCenter	vCenter Server の仮想マシンの名前が表示されます。  <b>注:</b> セッションでポッド フェデレーションの別のポッド実行されている場合、情報は表示されません。
デフォルトのプロトコル	デスクトップ セッションのデフォルトの表示プロトコル。

## アクティビティ

[アクティビティ] タブには、ユーザーのアクティビティに関するイベント ログ情報が表示されます。過去 12 時間、過去 30 日間などの期間や管理者の名前でアクティビティをフィルタリングできます。[ヘルプデスク イベントのみ] をクリックすると、Horizon Help Desk Tool アクティビティでのみフィルタリングできます。[更新] アイコンをクリックして、イベント ログを更新します。[エクスポート] アイコンをクリックして、イベント ログをファイルにエクスポートします。

**注:** クラウド ポッド アーキテクチャ環境のユーザーのイベント ログ情報は表示されません。

表 9-3. アクティビティ

オプション	説明
[時間]	時間範囲を選択します。デフォルトは、過去 12 時間です。 ■ [過去 12 時間] ■ [過去 24 時間] ■ [過去 7 日間] ■ [過去 30 日間] ■ [すべて]
[管理者]	管理者ユーザーの名前。
[メッセージ]	ユーザーまたは管理者が実行したアクティビティに固有のユーザーまたは管理者のメッセージが表示されます。
[リソース名]	アクティビティの実行対象のデスクトップ プールまたは仮想マシン名に関する情報が表示されます。

## Horizon Help Desk Tool のセッションの詳細

[セッション] タブで [コンピュータ名] オプションのユーザー名をクリックすると、セッションの詳細が [詳細] タブに表示されます。Horizon Client、仮想または公開デスクトップ、CPU とメモリの詳細を確認できます。

## クライアント

Horizon Client のタイプに応じて情報が表示されます。ユーザー名、Horizon Client のバージョン、クライアントマシンの IP アドレス、クライアント マシンのオペレーティング システムなどの詳細が表示されます。

**注:** Horizon Agent をアップグレードした場合、Horizon Client も最新バージョンにアップグレードする必要があります。それ以外の場合、Horizon Client のバージョンは表示されません。Horizon Client のアップグレードの詳細については、『Horizon 7 のアップグレード』ドキュメントを参照してください。

## 仮想マシン

仮想デスクトップまたは公開デスクトップに関する情報が表示されます。

表 9-4. 仮想マシンの詳細

オプション	説明
[コンピュータ名]	デスクトップ セッションの名前。
[エージェント バージョン]	Horizon Agent のバージョン。
[OS バージョン]	オペレーティング システムのバージョン。
[接続サーバ]	セッションが接続している接続サーバ。
[プール]	デスクトップ プールの名前
[vCenter Server]	vCenter Server の IP アドレス。
[セッション状態]	デスクトップ セッションの状態。セッションの状態は接続または切断です。
[セッションの期間]	セッションが接続サーバと接続していた期間。
[状態の継続期間]	セッションが同じ状態を継続した時間。
[ログイン時間]	セッションにログインしたユーザーのログイン時間。
[ログインの継続期間]	ユーザーが Linux デスクトップにログインしている時間。

## ユーザー操作性の評価基準

VMware Blast 表示プロトコルを使用する仮想または公開デスクトップ セッションのパフォーマンスの詳細が表示されます。これらのパフォーマンスの詳細を表示するには、[詳細] をクリックします。これらの詳細を更新するには、更新アイコンをクリックします。

表 9-5. Blast 表示プロトコルの詳細

オプション	説明
[フレーム レート]	Blast セッションのフレーム率 (1 秒あたりのフレーム数)。
[Skype の状態]	Linux デスクトップセッションの場合、このオプションは N/A と表示されます。
[Blast セッション カウンタ]	<ul style="list-style-type: none"> <li>■ [推定バンド幅 (アップリンク)]。アップリンク シグナルの推定バンド幅。</li> <li>■ [パケット損失 (アップリンク)]。アップリンク シグナルのパケット損失率。</li> </ul>

オプション	説明
[Blast イメージング カウンタ]	<ul style="list-style-type: none"> <li>■ [送信バイト]。Blast セッションで転送されたイメージング データの合計バイト数。</li> <li>■ [受信バイト]。Blast セッションで受信したイメージング データの合計バイト数。</li> </ul>
[Blast オーディオ カウンタ]	<ul style="list-style-type: none"> <li>■ [送信バイト]。Blast セッションで転送されたオーディオ データの合計バイト数。</li> <li>■ [受信バイト]。Blast セッションで受信したオーディオ データの合計バイト数。</li> </ul>
[Blast CDR カウンタ]	<ul style="list-style-type: none"> <li>■ [送信バイト]。Blast セッションで転送されたクライアント ドライブ リダイレクトの合計バイト数。</li> <li>■ [受信バイト]。Blast セッションで受信したクライアント ドライブ リダイレクトの合計バイト数。</li> </ul>

## CPU とメモリ使用量、ネットワークとディスクのパフォーマンス

仮想/公開デスクトップの CPU とメモリの使用量や、Blast 表示プロトコルのネットワークまたはディスク パフォーマンスがグラフで表示されます。

**注:** Horizon Agent デスクトップの起動または再起動後すぐに、パフォーマンス グラフにタイムラインが表示されない場合があります。数分後にタイムラインが表示されます。

表 9-6. CPU 使用率

オプション	説明
[セッションの CPU]	現在のセッションの CPU 使用率。
[ホストの CPU]	セッションが割り当てられている仮想マシンの CPU 使用率。

表 9-7. メモリ使用率

オプション	説明
[セッションのメモリ]	現在のセッションのメモリ使用量。
[ホストのメモリ]	セッションが割り当てられている仮想マシンのメモリ使用量。

表 9-8. ネットワークのパフォーマンス

オプション	説明
[遅延]	<p>PCoIP または Blast セッションの遅延がグラフで表示されます。</p> <p>遅延時間はラウンドトリップ時間（ミリ秒単位）です。この遅延時間を追跡するパフォーマンス カウンタは、[VMware Blast セッション カウンタ] - [RTT] です。</p>



表 9-9. ディスクのパフォーマンス

オプション	説明
[読み取り]	1 秒あたりの読み取りの入出力 (I/O) 操作の数。
[書き込み]	1 秒あたりの書き込み I/O 操作の数。
[ディスクの遅延時間]	ディスク遅延のグラフが表示されます。ディスク遅延は、Windows パフォーマンス カウンタから取得した入出力操作/秒 (IOPS) データの時間 (ミリ秒) 時間です。
[平均読み取り]	1 秒あたりのランダム読み取り I/O 操作の平均数。
[平均書き込み]	1 秒あたりのランダム書き込み I/O 操作の平均数。
[平均の遅延時間]	Windows パフォーマンス カウンタから取得した IOPS データの平均遅延時間 (ミリ秒)。

## セッション ログイン セグメント

ログインの継続時間とログイン時に作成されたセグメントが表示されます。

表 9-10. セッション ログイン セグメント

オプション	説明
[ログインの継続期間]	期間は、ユーザーがデスクトップ プールをクリックしてから Linux デスクトップにログインするまでの時間で計算されます。
[セッション ログイン時間]	ユーザーがセッションにログインしていた期間。
[ログイン セグメント]	<p>ログイン時に作成されたセグメントが表示されます。</p> <ul style="list-style-type: none"> <li>■ [仲介]。接続サーバがセッションの接続または再接続を処理する時間の合計。ユーザーがデスクトップ プールをクリックしてからトンネル接続が確立するまでの時間で計算されます。ユーザー認証、マシンの選択、トンネル接続を確立に必要なマシンの準備など、接続サーバのタスクの所要時間が含まれます。</li> <li>■ [インタラクティブ]。Horizon Agent がセッションの接続または再接続を処理する時間の合計。Blast Extreme がトンネル接続を開始してからユーザーが Linux デスクトップにログインするまでの時間で計算されます。</li> <li>■ [プロトコルの接続]。ログインで PCoIP または Blast プロトコル接続の完了にかかった合計時間。</li> <li>■ [ログイン スクリプト]。ログイン スクリプトが開始してから完了するまでの合計時間。</li> <li>■ [認証]。接続サーバがセッションの認証にかかった合計時間。</li> <li>■ [仮想マシンの開始]。仮想マシンの起動にかかった合計時間。この時間には、オペレーティング システムの起動、サスペンド状態のマシンの再開、Horizon Agent が接続準備完了通知の送信にかかる時間が含まれます。</li> </ul>

## Horizon Help Desk Tool のセッション プロセス

[セッション] タブで [コンピュータ名] オプションのユーザー名をクリックすると、セッション プロセスが [プロセス] タブに表示されます。

## プロセス

セッションごとに、CPU やメモリ関連プロセスの詳細情報を表示できます。たとえば、セッションの CPU やメモリ使用率が異常に高い場合、[プロセス] タブでプロセスの詳細を確認できます。

RDS ホスト セッションの場合、現在のユーザーまたはシステム プロセスが開始した RDS ホスト セッション プロセスが [プロセス] タブに表示されます。

表 9-11. セッション プロセスの詳細

オプション	説明
プロセス名	セッション プロセスの名前。たとえば、chrome.exe。
CPU	プロセスの CPU 使用率 (%)。
メモリ	プロセスのメモリ使用量 (KB)。
ディスク	メモリのディスク IOPS。次の式で計算されます。 (現在の時刻の I/O バイト数の合計) - (現在時刻より 1 秒前の I/O バイト数の合計)。 タスク マネージャに正の値が表示されている場合、この計算結果は 1 秒あたり 0 KB と表示されます。
ユーザー名	プロセスを所有するユーザーの名前。
ホストの CPU	セッションが割り当てられている仮想マシンの CPU 使用率。
ホストのメモリ	セッションが割り当てられている仮想マシンのメモリ使用量。
プロセス	仮想マシン内のプロセス数
更新	更新アイコンをクリックすると、プロセスのリストが更新されます。
プロセスの終了	実行中のプロセスを終了します。  <b>注:</b> プロセスを終了するには、ヘルプデスク管理者ロールが必要です。  プロセスを終了するには、プロセスを選択して [プロセスの終了] ボタンをクリックします。  Windows コアのプロセスなどの重要なプロセスは終了できません。これらのプロセスも [プロセス] タブに表示される場合があります。重要なプロセスを終了しようとすると、Horizon Help Desk Tool はメッセージを表示し、システム プロセスを終了できないことを通知します。

## Horizon Help Desk Tool での Linux デスクトップ セッションのトラブルシューティング

Horizon Help Desk Tool では、ユーザーの接続状態に基づいて Linux デスクトップ セッションのトラブルシューティングを行うことができます。

### 前提条件

- Horizon Help Desk Tool を開始します。

## 手順

- 1 ユーザー カードで、[セッション] タブをクリックします。

パフォーマンス カードに CPU とメモリの使用量と、Horizon Client、仮想デスクトップ、公開デスクトップに関する情報が表示されます。

- 2 トラブルシューティングのオプションを選択します。

オプション	アクション
[メッセージを送信]	公開デスクトップまたは仮想デスクトップのユーザーにメッセージを送信します。警告、情報、エラーなどのメッセージの重要度を選択します。 [メッセージの送信] をクリックし、重要度とメッセージの詳細を入力して、[送信] をクリックします。
[再起動]	仮想デスクトップで再起動プロセスを開始します。この機能は、公開デスクトップセッションで使用できません。 [VDI の再起動] をクリックします。
[切断]	デスクトップまたはアプリケーション セッションを切断します。 [詳細] - [切断] の順にクリックします。
[ログオフ]	公開デスクトップまたは仮想デスクトップでログオフ プロセスを開始します。 [詳細] - [ログオフ] の順にクリックします。
[リセット]	仮想マシンのリセットを開始します。この機能は、公開デスクトップで使用できません。 [詳細] - [仮想マシンのリセット] の順にクリックします。
<b>注:</b> 保存していない作業は失われます。	

## Horizon 7 for Linux マシンの診断情報の収集

VMware のテクニカル サポートが Horizon 7 for Linux マシンの問題を診断して解決する際に役立つ診断情報を収集できます。マシンの構成情報を収集して圧縮した tar ボールに記録するデータ収集ツール (DCT) バンドルを作成します。

## 手順

- 1 必要な権限を持つユーザーとして Linux 仮想マシンにログインします。
- 2 コマンド プロンプトを開いて、dct-debug.sh スクリプトを実行します。

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

スクリプトによって、DCT バンドルを含む tar ボールが生成されます。例：

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

tar ボールは、スクリプトが実行されたディレクトリ（現在の作業ディレクトリ）に生成されます。

## リモート デスクトップとクライアント ホストの間でのコピーと貼り付けに関するトラブルシューティング

リモート デスクトップとクライアント ホストの間でのコピーと貼り付けは、最大 1 MB のデータでサポートされ、処理に 3 秒以上かかります。この問題は、サイズの小さなデータのコピーと貼り付けを行う場合には発生しません。

### 問題

SLED 11 SP3/SP4 デスクトップ向けに 1 つの vCPU と 1 GB のメモリを構成するとき、リモート デスクトップとローカル クライアント ホストの間でのコピーと貼り付けに 3 秒以上かかることがあります。

### 原因

コピーと貼り付けの遅延は、SLED 11 SP3/SP4 のオペレーティング システム API が古いために発生することがあります。

### 解決方法

- ◆ SLED 11 SP3/SP4 に 2 つの vCPU と 2 GB のメモリを構成します。

## Horizon Agent が iPad Pro Horizon Client で切断できない

iPad Pro Horizon Client で再起動またはシャットダウンした後に、SUSE Horizon Agent の接続が切断できません。

### 問題

iPad Pro Horizon Client で、SUSE 仮想マシンを再起動またはシャットダウンする際、デスクトップは応答しません。Horizon Agent が切断に失敗します。

### 原因

再起動またはシャットダウンの操作の後に、SUSE マシンは Horizon Client にメッセージを正確に送信していない場合があります。

### 解決方法

- ◆ iPad Pro Horizon Client からデスクトップ接続を手動で切断してください。

## SLES 12 SP1 デスクトップが自動更新されない

マルチモニター モードでは、GNOME ターミナルをドラッグしても SLES 12 SP1 が自動更新されません。

### 問題

マルチモニター モードで SLES 12 SP1 を起動し、ウィンドウ モードに戻り、GNOME ターミナルをドラッグする場合は、デスクトップは自動的に更新しません。

### 原因

GNOME ターミナルは、ドラッグ操作に応答しません。

### 解決方法

- 1 GNOME シェル セッションを終了します。

```
kill -9 <process id of gnome-shell>
```

- 2 GNOME シェル セッションを再起動します。

## SSO がパワーオフ エージェントに接続できない

シングル サインオン (SSO) が PowerOff エージェントに接続しません。

### 問題

ブローカとしてログインし、エージェントに接続する場合、SSO は PowerOff エージェントに接続しません。

### 解決方法

- ◆ 手動でデスクトップへログインするか切断します。次に、エージェントに再接続します。

## Linux 版手動デスクトップ プール作成後の接続不能な仮想マシン

仮想マシン状態が応答状態にない。

### 問題

手動デスクトップ プールの作成後、仮想マシン ステータスは [エージェントの待機]、または [接続不能] になっている可能性があります。

### 原因

仮想マシン状態が [エージェントの待機] または [接続不能] になる場合には、いくつかのユーザー エラー構成またはセットアップに関する原因が考えられます。

- オプション `machine.id` が仮想マシンの `vmx` 構成ファイルに存在していることを確認します。

そのオプションが存在していない場合、仮想マシンがデスクトップ プールに正しく追加されていることを確認します。存在している場合は、デスクトップ プールを再作成し、ブローカで `vmx` 構成ファイルにそのオプションを再度書き込みます。

- VMware Tools または Open VM Tools が正しくインストールされていることを確認します。

VMware Tools または Open VM Tools をインストールするステップが正しく実行されていなかった場合、`vmware-rpctool` コマンドは Linux 仮想マシンの `PATH` に存在しない可能性があります。ガイドに従って VMware Tools または Open VM Tools をインストールする必要があります。

インストールの終了後、以下のコマンドを実行します。

```
#vmware-rpctool "machine.id.get"
```

`machine.id` の値は仮想マシンの `vmx` 構成ファイルから一覧表示されます。

- ブローカの完全修飾ドメイン名 (FQDN) をエージェント Linux 仮想マシンで IP アドレスに名前解決できるかどうかを確認します。