

Horizon 6 for Linux デスクト ップのセットアップ

VMware Horizon 6 6.2



vmware®

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>) でご確認ください。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 までお送りください。

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

ヴァイエルムウェア株式会社
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

Copyright © 2015 年～ 2017 年 VMware, Inc. All rights reserved. [著作権および商標情報](#)。

目次

Horizon 6 for Linux デスクトップのセットアップ 5

1 機能とシステムの要件 6

Horizon 6 Linux デスクトップとデスクトップ プールの機能 6

Horizon 6 for Linux デスクトップの構成手順の概要 8

Horizon 6 for Linux のシステム要件 9

2 デスクトップ デプロイのための Linux 仮想マシンの準備 13

仮想マシンを作成して、Linux インストールする 13

リモート デスクトップ デプロイ用の Linux マシンの準備 14

3 Linux デスクトップの Active Directory 統合のセットアップ 17

Linux と Active Directory の統合 17

シングル サインオンとスマート カード リダイレクトのセットアップ 18

4 Linux デスクトップの構成オプション 20

Linux デスクトップでの構成ファイルのオプション設定 20

Linux デスクトップの vSphere コンソールへの表示を抑制する 22

5 Linux デスクトップのグラフィックスのセットアップ 23

vGPU を使用するための RHEL 6.6 と RHEL 7.1 の構成 23

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 23

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 25

NVIDIA ディスプレイ ドライバのインストール 26

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 28

VDGA を使用するために RHEL 6.6 を構成する 29

ホストで NVIDIA GRID GPU を使用するために DirectPath I/O を有効にする 29

AvDGA バススルー デバイスを RHEL 6.6 仮想マシンに追加する 30

NVIDIA ディスプレイ ドライバのインストール 31

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 33

vSGA を使用するための RHEL 7.1 の構成 34

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 34

Linux 仮想マシンでの vSGA の 3D 機能の構成 35

vSGA が Linux 仮想マシンで実行されているかどうかの確認 36

6 View Agent のインストールと Linux デスクトップの管理 38

Linux 仮想マシンに View Agent をインストールする 38

install_viewagent.sh コマンドライン オプション 40

暗号化を元に戻せるパスワードの有効化	42
Linux 仮想マシンを含むデスクトップ プールを作成する	42
Linux 仮想マシンで View Agent をアップグレードする	43
Horizon 6 for Linux マシンのアンインストールと再インストール	44
手動デスクトップ プールからの非管理対象マシンを削除する	44
登録済みのマシンを View から削除する	45
Linux 仮想マシンで View Agent をアンインストールする	45
Linux 仮想マシンに View Agent を再インストールする	46
vSphere からの Linux デスクトップの電源操作の実行方法	47
Horizon 6 for Linux ソフトウェアに関する情報を収集する	47

7 Horizon 6 for Linux デスクトップの一括デプロイ 49

Linux デスクトップの一括デプロイの概要	49
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	51
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	53
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	53
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	57
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	60
Linux 仮想マシンで View Agent をインストールするサンプル スクリプト	64
SSH を使用して Linux 仮想マシンで View Agent をインストールするサンプル スクリプト	69
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	75
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	78
Linux デスクトップ マシンで View Agent をアップグレードするサンプル スクリプト	82
SSH を使用して Linux 仮想マシンで View Agent をアップグレードするサンプル スクリプト	87
Linux 仮想マシンで操作を実行するサンプル スクリプト	92
接続サーバの LDAP データベースからマシンを削除するサンプル スクリプト	96

8 Linux デスクトップのトラブルシューティング 99

Horizon 6 for Linux マシンの診断情報を収集する	99
Linux マシンでの View Agent 登録失敗のトラブルシューティング	99
Linux マシンの到達できない View Agent のトラブルシューティング	101
応答しない Linux マシンの View Agent のトラブルシューティング	102
TCP 接続の受信を許可するための Linux ファイアウォールの構成	103

Horizon 6 for Linux デスクトップのセットアップ

Horizon 6 for Linux デスクトップのセットアップドキュメントは、Linux ゲスト OS の準備、仮想マシンへの View Agent のインストール、Horizon 6 の展開環境で使用するための View Administrator でのマシンの構成など、Linux 仮想マシンをセットアップして、VMware Horizon 6™ デスクトップとして使用するための情報について説明します。

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。これらの情報は、仮想マシン テクノロジーおよびデータ 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記述されています。

機能とシステムの要件

Horizon 6 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 6 Linux デスクトップとデスクトップ プールの機能](#)
- [Horizon 6 for Linux デスクトップの構成手順の概要](#)
- [Horizon 6 for Linux のシステム要件](#)

Horizon 6 Linux デスクトップとデスクトップ プールの機能

Horizon 6 バージョン 6.2.1 では、Linux デスクトップの多くの新機能が導入されます。次のリストで主な機能について説明します。

仮想マシンのクローンのタイプ

次のクローンのタイプがサポートされています。

- リンクされたクローン
- 完全なクローン

リンクされたクローンの仮想マシン (VM) は、親仮想マシンのオペレーティング システム イメージを共有し、必要なストレージは完全なクローンの仮想マシンより少なくなります。

デスクトップ プール タイプ

次のデスクトップ プール タイプがサポートされています。

- 専用
- 流動

専用プールでは、ユーザーをデスクトップに割り当てると、その割り当ては永続的になります。ユーザーがログアウトして再度ログインすると、同じデスクトップが表示されます。流動プールでは、デスクトップの割り当ては永続的ではありません。ユーザーがログアウトして再度ログインすると、異なるデスクトップが表示されることがあります。

オーディオ出力

ユーザーは、デスクトップで生成されるオーディオを聞くことができます。

複数モニタ

4 台のモニタが、それぞれ最大解像度 2560x1600 でサポートされます。

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。
3D グラフィックス	<p>3D グラフィックスは、次の Linux バージョンとグラフィック カードの組み合わせでサポートされます。</p> <ul style="list-style-type: none">■ vSGA は、RHEL 7.1 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。■ vDGA は、RHEL 6.6 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。■ vGPU は、RHEL 6.6 Workstation x64 と NVIDIA Maxwell M60 グラフィック カードの組み合わせでサポートされます。■ vGPU は、RHEL 7.1 Workstation x64 と NVIDIA Maxwell M60 グラフィック カードの組み合わせでサポートされます。
クリップボード リダイレクト	<ul style="list-style-type: none">■ リッチ テキストとプレーン テキストは RHEL 6.6 Workstation x64 でサポートされます。■ リッチ テキストとプレーン テキストは CentOS 6.6 x64 でサポートされます。
シングル サインオン	<p>シングル サインオンは、次の Linux バージョンでサポートされます。</p> <ul style="list-style-type: none">■ RHEL 6.6 Workstation x64■ CentOS 6.6 x64
スマート カード リダイレクト	スマート カード リダイレクトは RHEL 6.6 Workstation x64 でサポートされます。Personal Identity Verification (PIV) カードと Common Access Card (CAC) がサポートされます。Mac クライアントはサポートされません。

注: クリップボード リダイレクトは、Ubuntu では正式にはサポートされません。ただし、この機能は、libexpat1 コンポーネントをコマンド `apt-get install libexpat1` でインストールすると動作します。

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。

- 自動デスクトップ プールでのみ提供される自動プロビジョニングおよびその他の機能は、サポートされません。たとえば、ログオフ操作の更新は、利用できません。
- USB リダイレクト、仮想印刷、ロケーションベースの印刷、リアルタイム オーディオビデオはサポートされません。
- HTML アクセスはサポートされません。

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 6 for Linux デスクトップの構成手順の概要

Horizon 6 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 6 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 6 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 インストールする](#) を参照してください。
- 3 Horizon 6 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、ユーザー環境の要件を基準に、サードパーティ ソフトウェアを使用して実施します。この手順についてはこのガイドでは説明しません。
- 5 Linux 仮想マシンに View Agent をインストールします。 [Linux 仮想マシンに View Agent をインストールする](#) 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 仮想マシンを含むデスクトップ プールを作成する](#) 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 View Agent をインストールし、View 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 の構成を完了します。

- 1 Horizon 6 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 6 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 インストールする](#) を参照してください。
- 3 Horizon 6 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、ユーザー環境の要件を基準に、サードパーティ ソフトウェアを使用して実施します。この手順についてはこのガイドでは説明しません。
- 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。
 - [vGPU を使用するための RHEL 6.6 と RHEL 7.1 の構成](#) を参照してください。
 - [vDGA を使用するために RHEL 6.6 を構成する](#) を参照してください。
 - [vSGA を使用するための RHEL 7.1 の構成](#) を参照してください。

- 6 Linux 仮想マシンに View Agent をインストールします。[Linux 仮想マシンに View Agent をインストールする](#)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 仮想マシンを含むデスクトップ プールを作成する](#)を参照してください。

一括デプロイ

View 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。[7 章 Horizon 6 for Linux デスクトップの一括デプロイ](#)を参照してください。

Horizon 6 for Linux のシステム要件

Horizon 6 for Linux は、特定のオペレーティング システム、Horizon 6、および vSphere プラットフォームの要件に適合している必要があります。

View Agent でサポートされる Linux バージョン

次の表は、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 システムを一覧で示しています。

表 1-1. View 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 14.04	x64
Ubuntu 12.04	x64
RHEL 7.1	x64
RHEL 6.6	x86 および x64
CentOS 6.6	x64
NeoKylin 6 Update 1	x64

注: Ubuntu 14.04 では、パフォーマンスが低下しないように Compiz を無効にしてください。

必須のプラットフォームと Horizon 6 ソフトウェア バージョン

Horizon 6 for Linux をインストールして使用するには、デプロイする環境が特定の vSphere プラットフォーム、Horizon 6、およびクライアント要件を満たしている必要があります。

vSphere プラットフォーム のバージョン	vSphere 5.5 U3、vSphere 6.0 U1 または以降のリリース NeoKylin をサポートするには、vSphere 6.0 以降のリリースが必要です。
------------------------------------	--

NVIDIA GRID vGPU、vDGA、および vSGA などの 3D グラフィックスを実装する仮想マシンをサポートするには、vSphere 6.0 U1 以降のリリースが必要です。

Horizon 環境	Horizon 6 バージョン 6.2.1 または以降のリリース
Horizon Client ソフトウェア	Horizon Client 3.5.2 (Windows 用)、3.5.0 (Linux 用)、または 3.5.2 (Mac OS X 用)
	ゼロ クライアントとモバイル クライアントはサポートされません

推奨されるビデオ 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 1-2. 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の数と解像度別に推奨される vRAM サイズを設定します。

推奨されるこれらの vRAM サイズは、VMware のドライバを使用する 2D または vSGA グラフィックスを使用するように構成された仮想マシンのみに適用されます。vDGA および NVIDIA GRID vGPU は、NVIDIA ドライバを使用します。vSphere Client で設定される vRAM サイズは、vDGA や NVIDIA GRID vGPU マシンには影響しません。

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。

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。

注意: Horizon 6 は、Windows 仮想マシンの場合とは異なり、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。

表 1-2. 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	モニタ数	最大解像度
10 MB	1	1600x1200 または 1680x1050
12 MB	1	1920x1440
16 MB	1	2560x1600
32 MB	2	2048x1536 または 2560x1600
48 MB	3	2048x1536
64 MB	3	2560x1600

VRAM サイズ	モニタ数	最大解像度
64 MB	4	2048x1536
128 MB	4	2560x1600

RHEL および CentOS は、vSphere 5.5 でのみこの構成をサポートします。

この構成を Ubuntu でサポートするには、カーネルを再コンパイルして 3D を無効にする必要があります。3D を無効にする方法については、<http://kb.vmware.com/kb/2114809> を参照してください。

NeoKylin では、この構成はサポートされません。

注: RHEL 6.6 または CentOS 6.6 デスクトップに複数のモニタを接続するには、[仮想マシンを作成して、Linux インストールする](#)に説明されているとおりに、ディスプレイの数とビデオ メモリの量を設定する必要があります。また、仮想マシン (VM) をパワーオフし、ディスプレイの数と方向 (水平または垂直) に従って vmx ファイルを編集して `svga.maxWidth` と `svga.maxHeight` を設定する必要があります。 `svga.autodetect` を `false` に設定する必要もあります。次に、仮想マシンをパワーオンします。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。4 つのディスプレイを最大解像度 2560x1600 でサポートするには、次の行を vmx ファイルに追加します。パラメータ `svga.vramSize` の変更は不要です。

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

複数のモニタを使用する場合は、これらの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。

- 1 つのモニタのみが正常に表示される。その他のモニタは黒い画面になるか別のモニタをミラーリングすることがある。
- キーストロークが何回も表示される。
- デスクトップが非常に遅くなる。

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 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"
```

注: vSGA で 2D レンダリングや 3D レンダリングを使用する RHEL 7.1 マシンで 4 台のモニタを構成するには、各モニタの最大解像度を 2048x1536 に設定します。最大で 3 台のモニタを使用するように RHEL 7.1 マシンを構成するには、各モニタの最大解像度を 2560x1600 に設定します。

vSGA の複数モニタ環境で RHEL 7.1 マシンのパフォーマンスを向上するには、仮想マシンの [3D メモリ] 設定を 1 GB 以上に設定し、4 個の vCPU をマシンに構成します。解像度が 2048x1536 の 4 台のモニタを RHEL 7.1 マシンで構成するには、[メモリ] を 4 GB に設定し、4 個の vCPU をマシンに構成します。

複数のモニタをサポートするために推奨される vCPU および共有メモリ設定

複数のモニタが接続されるデスクトップのパフォーマンスを向上するために、Linux 仮想マシンでは少なくとも 2 台の vCPU を構成します。

また、仮想マシンに複数のモニタをサポートできるだけの十分な共有メモリがあることを確認してください。次のコマンドを使用して、現在の最大共有メモリ サイズ (shmmax) を判別します。

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

共有メモリが小さい場合、次のコマンドを使用して最大サイズを増加します。

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

ビデオ再生で推奨される vCPU 設定

vDGA グラフィックスを使用しない Linux 仮想マシンで、構成される vCPU の数が少なすぎると、高解像度モニタでのビデオ再生が不規則になる場合があります。ビデオ再生のパフォーマンスを向上するには、4 台の vCPU など、追加の vCPU を構成します。

vDGA および NVIDIA GRID vGPU でサポートされるモニタ解像度

vDGA や NVIDIA GRID vGPU を使用するように構成された仮想マシンは、最大解像度が 2560x1600 のモニタを最大 4 台までサポートできます。

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 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

仮想マシンを作成して、Linux インストールする

Horizon 6 にデプロイする各リモート デスクトップについて 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 6 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。View でのデスクトップ プールとアプリケーション プールの設定 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について推奨されるビデオ メモリ (vRAM) の値を理解しておきます。 [Horizon 6 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。

推奨値については、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 12.04 では、2048 MB の vMemory と 2 台の vCPU を構成することが推奨されています。

- 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。

推奨値については、Horizon 6 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 gnome デスクトップ環境として仮想マシンを構成します。

基本的な接続およびオーディオとビデオは Kubuntu などのいくつかのディストリビューションで正しく動作しますが、KDE による View Agent のサポートは認定されていません。

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 6 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 6 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 マシンにプレ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 RHEL 7 マシンを準備する場合、下記の手順 1 ～ 5 はスキップできます。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。

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 RHEL と CentOS の場合は、`/etc/hosts` でマシンのホスト名を 127.0.0.1 にマップします。

この手順は、Ubuntu の場合はマッピングがデフォルトで存在するため不要です。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。

- 7 ポッドにある View 接続サーバ 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。
- 8 Ubuntu 14.04 マシンで、ユーザー切り替えを無効にして、エンド ユーザーが Linux デスクトップ内からユーザー セッションを切り替えることができないようにします。
 - a ユーザー切り替えを正常に無効にするには、<https://launchpad.net/ubuntu/wily/amd64/indicator-session/12.10.5+15.04.20150327-0ubuntu1> にある indicator-session パッケージの最新バージョンをダウンロードしてインストールします。
 - b マシンを再起動します。

- 9 RHEL 7.1 マシンの場合は、インストールされている最新のグラフィック ドライバをダウングレードします。

```
yum update
yum downgrade mesa-libxatracker mesa-dri-drivers
yum downgrade mesa-libxatracker.x86_64 xorg-x11-drv-vmware mesa-private-llvm mesa-dri-drivers
```

- 10 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。

- 11 [HD オーディオ] デバイスが仮想マシンの仮想ハードウェア設定にないことを確認します。

[HD オーディオ] がマシンで構成されている場合、ユーザーが Linux デスクトップに接続するときに、オーディオ出力が適切に動作しない場合があります。

- 12 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、View 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。

- 13 RHEL と CentOS の場合は、/etc/cron.daily/tmpwatch を編集し、パラメータ -X '/tmp/view*' をコマンド tmpwatch に追加します。

次に例を示します。

```
/usr/sbin/tmpwatch "$flags" -x /tmp/.X11-unix -x /tmp/.XIM-unix \
-x /tmp/.font-unix -x /tmp/.ICE-unix -x /tmp/.Test-unix \
-X '/tmp/hsperfdata_*' -X '/tmp/view*' 10d /tmp
```

- 14 /etc/nsswitch.conf を編集し、hosts エントリを次のように変更します。

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

デフォルト設定は hosts: files dns です。このため、Linux デスクトップが一時的なネットワーク停止からリカバリできないことがあります。Linux デスクトップで View 接続サーバとの通信が完全に失われ、問題を解決するために再起動が必要になる場合があります。

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。

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

3

View 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。

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 サインオンとスマート カード リダイレクトのセットアップ](#)

Linux と Active Directory の統合

Linux と Active Directory (AD) を統合するためのソリューションは複数あります。

次のソリューションは View 環境で動作することが分かっています。

- OpenLDAP パススルー認証
- Winbind

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。

- 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 サーバを構成します。saslauthd は Active Directory に対してパスワード検証を実行できます。
- OpenLDAP でユーザーを認証するように Linux デスクトップを構成します。

Linux デスクトップを一括デプロイする予定がある場合は、最後の Active Directory 統合タスクを実行するようにテンプレート仮想マシン (VM) をセットアップできます。次の考慮事項を認識しておく必要があります。

- OpenLDAP ソリューションは追加の手順なしでクローン仮想マシンで動作します。
- Winbind ソリューションでは、各クローン仮想マシンには別のホスト名があるため、ドメインに参加する手順が失敗します。各クローン仮想マシンは、次のコマンドを実行してドメインに再参加する必要があります。

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

クローン仮想マシンで追加の手順を行う必要がないため、OpenLDAP または同様のソリューションをお勧めします。

Linux デスクトップの一括デプロイに関する詳細については、[7 章 Horizon 6 for Linux デスクトップの一括デプロイ](#)を参照してください。

シングル サインオンとスマート カード リダイレクトのセットアップ

シングル サインオン (SSO) とスマート カード リダイレクトをセットアップするには、構成手順をいくつか実行する必要があります。

シングル サインオン

View 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) と対話し、Linux を Active Directory (AD) と統合するために使用する方法には依存しません。View の SSO は、Linux を Active Directory と統合する 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ることに注意してください。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。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。

View では、ユーザー名の大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、大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、ユーザー名の大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
```

スマート カード リダイレクト

スマート カード リダイレクトをセットアップするには、最初に Linux ディストリビュータとスマート カード ベンダーの指示に従ってください。次に、pcsc-lite パッケージを 1.7.4 に更新します。たとえば、次のコマンドを実行します。

```
#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-confdir=/etc --enable-ipcdire=/var/run --disable-libusb --disable-serial --disable-usb
--disable-libudev
#service pcscd start
```

View Agent をインストールするときは、最初に SELinux を無効にするか、SELinux の permissive モードを有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。

スマート カード リダイレクトが有効である場合、ユーザーがスマート カードでログインすると、SSO は実装されません。つまりユーザーは、スマート カードで接続サーバ インスタンスにログインした後で、スマート カード PIN を入力して Linux デスクトップにログインする必要があります。スマート カード リダイレクト機能を仮想マシンにインストールすると、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。これは Linux の制限です。

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
- 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、View 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 `template_config` と `template_viewagent-custom.conf` 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 `template-config` を `config` に、`template_viewagent-custom.conf` 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、可逆圧縮 PNG モードが有効になります。

```
RemoteDisplay.alwaysLossless=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。

表 4-1. /etc/vmware/config の構成オプション

オプション	値	デフォルト	説明
RemoteDisplay.alwaysLossless	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、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。
mksVNCServer.useUInputButton Mapping	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 6.6 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。
RemoteDisplay.allowAudio	TRUE または FALSE	TRUE	オーディオ出力を無効にします。
WVC.ScRedir.Enable	TRUE または FALSE	TRUE	スマート カード リダイレクトを無効にします。
Clipboard.Direction	0、1、2、3	2	このオプションにより、クリップボード リダイレクト ポリシーを決めます。 <ul style="list-style-type: none"> ■ 0 - クリップボード リダイレクトを無効にします。 ■ 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。 ■ 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。 ■ 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。

/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 4-2. /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
SSOEnable	TRUE または FALSE	TRUE	シングル サインオン (SSO) を無効にします。
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。ログイン名の形式の例を次に挙げます。 <ul style="list-style-type: none"> ■ SSOUserFormat=[domain]\\[username] ■ SSOUserFormat=[domain]+[username] ■ SSOUserFormat=[username]@[domain]
StartBlastServerTimeout	整数	20	VMwareBlastServer プロセスで初期化に使用する時間を秒単位で決めます。このタイムアウト値以内にプロセスの準備ができない場合、ユーザーのログインは失敗します。
SSLCiphers	テキスト文字列	!aNULL:kECDH+AES:EC DH+AES:RSA+AES:@STRENGTH	暗号化のリストを指定します。 https://www.openssl.org/docs/manmaster/apps/ciphers.html で定義されている形式を使用する必要があります。

オプション	値	デフォルト	説明
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。
SSLCipherServerPreference	TRUE または FALSE	TRUE	オプション SSL_OP_CIPHER_SERVER_PREFERENCE を有効または無効にします。詳細については、 https://www.openssl.org/docs/manmaster/ssl/SSL_CTX_set_options.html を参照してください。

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。

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 デスクトップのグラフィックス のセットアップ

5

この章には、次のトピックが含まれています。

- [vGPU を使用するための RHEL 6.6 と RHEL 7.1 の構成](#)
- [vDGA を使用するために RHEL 6.6 を構成する](#)
- [vSGA を使用するための RHEL 7.1 の構成](#)

vGPU を使用するための RHEL 6.6 と RHEL 7.1 の構成

RHEL 6.6 と RHEL 7.1 をセットアップすると、NVIDIA vGPU（共有 GPU ハードウェア アクセラレーション）機能を ESXi ホストで利用できます。

重要: NVIDIA vGPU は、NVIDIA Maxwell M60 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、View 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 View 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、View Agent をインストールする必要があります。

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 GRID vGPU を構成する場合、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。

vSGA を構成する場合、NVIDIA から vSGA 用の VMware vSphere ESXi ドライバが提供されます。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 GRID vGPU を構成する場合、NVIDIA Maxwell M60 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。

- vSGA を構成する場合、NVIDIA GRID K1 や 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 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	vGPU では、[NVIDIA GRID vGPU] を選択します。 vSGA では、[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 NVIDIA GRID vGPU を構成する場合、次の手順を実行します。
 - a vGPU ソフトウェア パッケージの .zip ファイルを解凍します。
 - b vGPU Manager フォルダを ESXi 6.0 U1 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。

- 3 vSGA を構成する場合、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。
- 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 acceptance set --level=CommunitySupported
# esxcli software vib install --no-sig-check -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```


8 ESXi ホストを再起動または更新します。

3D タイプ	説明
NVIDIA GRID vGPU	<p>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。</p> <p>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）。</p> <pre>Update vmkdevmgr: # kill -HUP \$(cat /var/run/vmware/vmkdevmgr.pid) Wait for the update to complete: # localcli --plugin-dir /usr/lib/vmware/esxcli/int deviceInternal bind This is a new requirement with the NVIDIA 352.* host driver: # /etc/init.d/nvidia-vgpu start Restart xorg, which is used for GPU assignment: # /etc/init.d/xorg start</pre>
vSGA	<p>a GPU の割り当てに使用される xorg を再起動します。</p> <pre># /etc/init.d/xorg start</pre> <p>b ESXi ホストを再起動します。</p>

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 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。
- View 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 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 GRID vGPU は、NVIDIA Maxwell M60 GPU でサポートされます。

表 5-1. Linux 仮想マシンの NVIDIA GRID 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 M60-0q	GRID M60	2	512M	2	2560x1600	16	32
GRID M60-1q	GRID M60	2	1G	2	2560x1600	8	16
GRID M60-2q	GRID M60	2	2G	4	2560x1600	4	8
GRID M60-4q	GRID M60	2	4G	4	3840x2160	2	4
GRID M60-8q	GRID M60	2	8G	4	3840x2160	1	2

NVIDIA ディスプレイ ドライバのインストール

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

- NVIDIA GRID vGPU を構成する場合、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 グラフィックス 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。

- vDGA を構成する場合、PCI デバイスが RHEL 6.6 仮想マシンに追加されていることを確認します。 [AvDGA パススルー デバイスを RHEL 6.6 仮想マシンに追加する](#) 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 6.6 の場合のファイルは `/boot/grub/grub.conf` です。RHEL 7.1 の場合のファイルは `/etc/default/grub.conf` です。

RHEL バージョン	コマンド
6.6	<code>sudo vi /boot/grub/grub.conf</code>
7.1	<code>sudo vi /etc/default/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 ディスプレイ ドライバを仮想マシンにコピーまたはダウンロードします。

- a NVIDIA GRID vGPU を構成する場合、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。
- b vDGA を構成する場合、[NVIDIA のドライバ ダウンロード](#) サイトから NVIDIA 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。
- 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 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。

次の例では、NVIDIA GRID vGPU のドライバ パッケージが使用されています。

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

次の例では、vDGA のドライバ パッケージが使用されています。

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

- 8 NVIDIA インストーラを起動します。

NVIDIA GRID vGPU の場合：

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

vDGA の場合：

```
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 の構成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 View Agent をインストールします。[Linux 仮想マシンに View Agent をインストールする](#)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 仮想マシンを含むデスクトップ プールを作成する](#)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

View 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 RHEL 6.6 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。
- View 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 仮想マシンに View Agent をインストールする](#)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 仮想マシンを含むデスクトップ プールを作成する](#)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。

View 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 ドライバをインストールした後、Linux カーネルをアップグレードすると、View Agent が View 接続サーバと通信できなくなる場合があります。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。

vDGA を使用するために RHEL 6.6 を構成する

Horizon 6 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 6.6 ゲスト OS をセットアップできます。

注意： 開始する前に、View 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 View 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、View Agent をインストールする必要があります。

ホストで NVIDIA GRID GPU 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。

AvDGA パススルー デバイスを RHEL 6.6 仮想マシンに追加する

vDGA を使用するように RHEL 6.6 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 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。
- View 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 GPU 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 仮想マシンをパワーオフします。sudo 権限で構成されたローカル ユーザーとして、RHEL 6.6 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]
```

NVIDIA ディスプレイ ドライバのインストール

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

- NVIDIA GRID vGPU を構成する場合、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 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。
- vDGA を構成する場合、PCI デバイスが RHEL 6.6 仮想マシンに追加されていることを確認します。 [AvDGA パススルー デバイスを RHEL 6.6 仮想マシンに追加する](#)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 6.6 の場合のファイルは `/boot/grub/grub.conf` です。RHEL 7.1 の場合のファイルは `/etc/default/grub.conf` です。

RHEL バージョン	コマンド
6.6	<code>sudo vi /boot/grub/grub.conf</code>
7.1	<code>sudo vi /etc/default/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 ディスプレイ ドライバを仮想マシンにコピーまたはダウンロードします。

a NVIDIA GRID vGPU を構成する場合、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。

b vDGA を構成する場合、[NVIDIA のドライバ ダウンロード](#) サイトから NVIDIA 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。

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 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。

次の例では、NVIDIA GRID vGPU のドライバ パッケージが使用されています。

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

次の例では、vDGA のドライバ パッケージが使用されています。

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

8 NVIDIA インストーラを起動します。

NVIDIA GRID vGPU の場合：

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


vDGA の場合 :

```
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 の構成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 View Agent をインストールします。[Linux 仮想マシンに View Agent をインストールする](#)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 仮想マシンを含むデスクトップ プールを作成する](#)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

View 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 RHEL 6.6 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。
- View 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 仮想マシンに View Agent をインストールする](#)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 仮想マシンを含むデスクトップ プールを作成する](#)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。

View 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 ドライバをインストールした後、Linux カーネルをアップグレードすると、View Agent が View 接続サーバと通信できなくなる場合があります。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。

vSGA を使用するための RHEL 7.1 の構成

Horizon 6 for Linux デスクトップが vSGA 機能を利用できるように、RHEL 7.1 ゲスト OS をセットアップできます。

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 GRID vGPU を構成する場合、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。

vSGA を構成する場合、NVIDIA から vSGA 用の VMware vSphere ESXi ドライバが提供されます。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 GRID vGPU を構成する場合、NVIDIA Maxwell M60 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。
- vSGA を構成する場合、NVIDIA GRID K1 や 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 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	vGPU では、[NVIDIA GRID vGPU] を選択します。 vSGA では、[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 NVIDIA GRID vGPU を構成する場合、次の手順を実行します。
 - a vGPU ソフトウェア パッケージの .zip ファイルを解凍します。
 - b vGPU Manager フォルダを ESXi 6.0 U1 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。

- 3 vSGA を構成する場合、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。
- 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 acceptance set --level=CommunitySupported
# esxcli software vib install --no-sig-check -v /path-to-vib/NVIDIA-VIB-name.vib
# esxcli system maintenanceMode set --enable false
```

- 8 ESXi ホストを再起動または更新します。

3D タイプ	説明
NVIDIA GRID vGPU	<p>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。</p> <p>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）。</p> <pre>Update vmkdevmgr: # kill -HUP \$(cat /var/run/vmware/vmkdevmgr.pid) Wait for the update to complete: # localcli --plugin-dir /usr/lib/vmware/esxcli/int deviceInternal bind This is a new requirement with the NVIDIA 352.* host driver: # /etc/init.d/nvidia-vgpu start Restart xorg, which is used for GPU assignment: # /etc/init.d/xorg start</pre>
vSGA	<p>a GPU の割り当てに使用される xorg を再起動します。</p> <pre># /etc/init.d/xorg start</pre> <p>b ESXi ホストを再起動します。</p>

- 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 7.1 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いるか、View Agent がインストールされているか、そして、マシンがデスクトップ プールにデプロイされているか確認します。
- 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 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 View Agent をインストールします。[Linux 仮想マシンに View Agent をインストールする](#)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 7.1 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| svga| I125: OpenGL Version: "4.5.0 NVIDIA 346.69" (4.5.0)
2015-06-24T22:19:25.259Z| svga| I125: GLSL Version: "4.50 NVIDIA" (4.50.0)
2015-06-24T22:19:25.259Z| svga| I125: OpenGL Vendor: "NVIDIA Corporation"
2015-06-24T22:19:25.259Z| svga| I125: 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| I125: [msg.mks.noGPUResourceFallback] Hardware GPU resources are
not available. The virtual machine will use software rendering.
2015-07-06T17:09:26.423Z| vmx| I125: -----
2015-07-06T17:09:26.425Z| svga| I125: MKS-SWP: plugin started - llvmpipe (LLVM 3.3, 256 bits)
2015-07-06T17:09:26.426Z| svga| I125: Started Shim3D
2015-07-06T17:09:26.426Z| svga| I125: 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 View Agent をインストールします。[Linux 仮想マシンに View Agent をインストールする](#)を参照してください。

View Agent のインストールと Linux デスクトップの管理

View 接続サーバがデスクトップと通信し管理することができるように、View Agent を Linux デスクトップに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 仮想マシンに View Agent をインストールする](#)
- [暗号化を元に戻せるパスワードの有効化](#)
- [Linux 仮想マシンを含むデスクトップ プールを作成する](#)
- [Linux 仮想マシンで View Agent をアップグレードする](#)
- [Horizon 6 for Linux マシンのアンインストールと再インストール](#)
- [vSphere からの Linux デスクトップの電源操作の実行方法](#)
- [Horizon 6 for Linux ソフトウェアに関する情報を収集する](#)

Linux 仮想マシンに View Agent をインストールする

Linux 仮想マシンをリモート デスクトップとしてデプロイできるようにするには、View Agent を Linux 仮想マシンにインストールする必要があります。

View Agent をインストールするには、View 管理者の認証情報を入力する必要があります。インストーラはその認証情報を認証のため接続サーバに送信します。次の認証メカニズムがサポートされています。

- DIGEST-MD5。このメカニズムでは、Active Directory ポリシーの [暗号化を元に戻せる状態でパスワードを保存する] が有効になっている必要があります。このポリシーの有効化の詳細については、[暗号化を元に戻せるパスワードの有効化](#)を参照してください。

- Kerberos。このメカニズムを使用するには、[-k] コマンドライン オプションで Active Directory サーバのアドレスを指定する必要があります。また、必須パラメータ [-b] には、実際のホスト名とドメイン名を使用して接続サーバの完全修飾ドメイン名 (FQDN) を指定する必要があります。IP アドレスまたはマップされた FQDN は使用できません。

注意: NVIDIA GRID vGPU、vDGA、または vSGA を使用する場合には、View 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。View 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。

vGPU を使用するための RHEL 6.6 と RHEL 7.1 の構成、vDGA を使用するために RHEL 6.6 を構成する、または vSGA を使用するための RHEL 7.1 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、View 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、View 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 向けの View 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。
- インストール コマンドで指定した View 接続サーバの管理者に View Administrator で Agent 登録管理者 または 管理者 のロールが関連付けられていることを確認します。特に、View Agent を View 接続サーバに登録するには、エージェントを登録 の権限が必要です。Agent 登録管理者 は、この最小権限を提供する制限付きのロールです。
- インストール環境で複数のドメインを使用している場合には、View Administrator のユーザー アカウントが View 接続サーバと同じドメインになければなりません。
- View Agent のインストール時に Linux View Agent が View 接続サーバと通信できるように、ポート 389 が使用可能であることを確認します。

手順

- 1 VMware ダウンロード サイト (<https://my.vmware.com/web/vmware/downloads>) から、View 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] にある [VMware Horizon 6] のダウンロードを選択します。このダウンロードには、View Agent for Linux のインストーラが含まれます。

64 ビットの Linux ではインストーラのファイル名は `VMware-viewagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz` となり、32 ビットの Linux では `VMware-viewagent-linux-y.y.y-xxxxxxx.tar.gz` となります。`y.y.y` はバージョン番号で、`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 <View Agent の tar ボール>
```

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 -b mybroker.mydomain.com -d mydomain.com -u administrator -p password
```

`-b` パラメータは、Linux マシンが登録されている View 接続サーバ インスタンスを指定します。`-b` パラメータでは FQDN や IP アドレスを使用できます。

`-d`、`-u`、および `-p` パラメータは、View 管理者のドメイン、ユーザー名、パスワードを指定します。パスワードを指定するときには、\$ などの特殊文字は必ずエスケープしてください。たとえば、`ab\$cdef` のようにエスケープします。

このコマンドで `-p` パラメータを入力しないと、コマンドを入力した後にパスワードを指定するプロンプトが表示されます。プロンプトでパスワードを入力するときには、テキストは表示されません。

デフォルトでは、Linux マシンのホスト名がマシンを登録するときに使用されます。`-n` パラメータを使用して別のマシン名を指定することができます。

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。

インストール後にマシンが View 接続サーバに登録され、`viewagent` サービスが開始されます。`#service viewagent status` コマンドを実行して、サービスが起動していることを確認できます。

エージェントの登録に失敗しました というメッセージが表示される場合には、[Linux マシンでの View Agent 登録失敗のトラブルシューティング](#)を参照してください。

その他の View Agent の問題が発生する場合には、[8 章 Linux デスクトップのトラブルシューティング](#)を参照してください。

注: `install_viewagent.sh` スクリプトによって、Linux 仮想マシンの View Agent インストール ディレクトリ `/usr/lib/vmware/viewagent/jre` に JRE 8 がインストールされます。View Agent には JRE 8 が必要です。

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 仮想マシンを含むデスクトップ プールを作成する](#)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、View 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 6-1. `install_viewagent.sh` の必須パラメータ

必須パラメータ	説明
-b	認証メカニズムが DIGEST-MD5 の場合は、View 接続サーバ インスタンスの FQDN または IP アドレス。認証メカニズムが Kerberos の場合は、IP アドレスやマップされた FQDN ではなく実際の FQDN を指定します。
-d	View 接続サーバ インスタンスのドメイン名。例: mydomain.com
-u	View 接続サーバ管理者のユーザー名。例: myadmin 管理者は、View Administrator で Agent 登録管理者 または 管理者 のロールを持っている必要があります。特に、View Agent を View 接続サーバに登録するには、エージェントを登録 の権限が必要です。Agent 登録管理者 は、この最小権限を提供する制限付きのロールです。

表 6-2. `install_viewagent.sh` のパスワード パラメータ

オプションパラメータ (必須情報)	説明
-p	View 接続サーバ管理者のパスワード。 テキストとしてパスワードを表示させない場合、コマンドラインで -p パラメータを入力する必要はありません。-p パラメータを指定せずに <code>install_viewagent.sh</code> コマンドを入力すると、パスワードを入力するプロンプトが表示されます。プロンプトでパスワードを入力するときには、テキストは表示されません。 -p パラメータまたはプロンプトのいずれかでパスワードを指定する必要があります。 インストーラが実行されるシェル内で \$ のような特殊文字がパスワードに含まれる場合は、必ず特殊文字をエスケープします。たとえば、ab\cdef のようにエスケープします。
-A yes	EULA への同意。インストールを続行するには、 yes 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。

表 6-3. `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
-n	View 接続サーバに登録されるマシン名。デフォルトでは、View Agent インストーラはホスト名を使用します。独自の名前を指定できます。
-s	自己署名証明書のサブジェクト DN。 デフォルトでは、View Agent インストーラによって、サブジェクト DN 値が '/C=US/ST=California/O=VMware/CN=Hostname.DomainName' の自己署名証明書が生成されます。デフォルトの CN では、マシンのホスト名とドメイン名が指定されます。 独自のサブジェクト DN を指定できます。
-m yes	スマート カード リダイレクトを有効にします。
-j	JMS SSL キーストア パスワード。これを指定しない場合、インストーラによってランダム パスワードが生成されます。

オプション パラメータ	説明
-k	Kerberos 認証用の Active Directory サーバのアドレス。このパラメータを設定すると、インストーラによって認証用の Kerberos のみが使用されます。
-r	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。使用可能な値は、 yes および no です。デフォルトは、 no です。

暗号化を元に戻せるパスワードの有効化

インストール中、View Agent は、自身を View 管理者として View 接続サーバに認証する必要があります。認証方式が DIGEST-MD5 である場合、最初に Active Directory (AD) の設定 [暗号化を元に戻せる状態でパスワードを保存する] を管理者のアカウントで有効にする必要があります。

手順

- 1 Active Directory サーバで、[管理ツール] - [Active Directory ユーザーおよびコンピュータ] を選択します。
- 2 View 管理者のアカウントを右クリックし、[プロパティ] を選択します。
- 3 [アカウント] タブを選択します。
- 4 [アカウント] オプションで、[暗号化を元に戻せる状態でパスワードを保存する] 設定を選択します。
- 5 View 管理者のパスワードをリセットします。

Linux 仮想マシンを含むデスクトップ プールを作成する

リモート デスクトップとして使用するよう Linux 仮想マシンを構成するには、手動デスクトップ プールを作成して、Linux マシンをプールに追加します。

デスクトップ プールを作成するときには、Linux 仮想マシンのみをプールに追加します。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なくなります。

デスクトップ プールの Linux マシンに対する資格をユーザーに付与するときには、ベスト プラクティスとして、Linux ゲスト OS で管理者権限をユーザーが持っていないことを確認してください。Linux の管理者ユーザーは、ターミナル ウィンドウを開いて、仮想マシンをパワーオフする `shutdown` などのコマンドを起動できます。vCenter Server 管理者は、マシンを再度パワーオンする必要があります。管理者以外の Linux ユーザーに資格を付与すると、これらの電源操作を手動で管理する必要がなくなります。

前提条件

- View 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 仮想マシンに View Agent をインストールする](#)を参照してください。
- Linux 仮想マシンが View 接続サーバに登録されていることを確認します。View Administrator で、[View 構成] - [登録済みのマシン] を選択して、[その他] タブを選択します。各仮想マシンの状態が **使用可能** であることを確認します。

- Ubuntu 14.04 マシンで、ユーザー切り替えが無効になっ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 マシンの準備の手順 9](#) を参照してください。

手順

- 1 View 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 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流動のいずれかを選択します。
- 4 [マシン ソース] ページで、[その他のソース] を選択します。
- 5 [マシンを追加] ページで、構成した Linux 仮想マシンを選択して、[デスクトップ プールを追加] ウィザードを完了します。

[リモート表示プロトコル] 設定は変更しないでください。これらの設定は、Linux デスクトップに影響しません。
また、エンド ユーザーは表示プロトコルを選択できません。

重要: Linux 仮想マシンのみを追加します。Windows 仮想マシンを追加すると、プールの Linux デスクトップは利用できなくなります。

- 6 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。

View Administrator で、デスクトップ プールを選択して、[資格] - [資格を追加] を選択して、ユーザーまたはグループを追加します。

ベスト プラクティスとして、Linux ゲスト OS ではユーザーに管理者権限を付与しないようにしてください。

Linux 仮想マシンを、Horizon 6 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 仮想マシンで View Agent をアップグレードする

View 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 View Agent をアップグレードできます。

以前のインストールで指定したほとんどのインストール パラメータは記憶されます。 オプションで次のパラメータを指定できます。

パラメータ	説明
-A yes	EULA への同意。インストールを続行するには、 yes 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。
-m yes	スマート カード リダイレクトを有効にします。
-r	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。 使用可能な値は、 yes および no です。デフォルトは、 no です。

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 VMware ダウンロード サイト (<https://my.vmware.com/web/vmware/downloads>) から、最新の Linux 版 View Agent インストーラ ファイルをダウンロードします。

[Desktop & End-User Computing] にある [VMware Horizon 6] のダウンロードを選択します。このダウンロードには、Linux 版の View Agent のインストーラが含まれます。

64 ビットの Linux ではインストーラのファイル名は VMware-viewagent-linux-x86_64-y.y.y-xxxxxxx.tar.gz となり、32 ビットの Linux では VMware-viewagent-linux-y.y.y-xxxxxxx.tar.gz となります。y.y.y はバージョン番号で、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 <View Agent の tar ボール>
```

- 3 tar ボール フォルダに移動します。
- 4 `install_viewagent.sh` スクリプトを実行します。

例：

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

Horizon 6 for Linux マシンのアンインストールと再インストール

仮想マシンの Horizon 6 for Linux をアンインストールするには、デスクトップ プールから非管理対象マシンを削除し、View から登録済みマシンを削除し、View Agent をアンインストールして、構成ファイルを削除する必要があります。View Agent のインストール スクリプトを実行して、Horizon 6 for Linux を再インストールできます。

手動デスクトップ プールからの非管理対象マシンを削除する

非管理対象マシンをプールから削除することによって、手動デスクトップ プールのサイズを減らすことができます。

手順

- 1 View Administrator で、[カタログ] - [デスクトップ プール] を選択します。
- 2 手動プールのプール ID をダブルクリックします。
- 3 [インベントリ] タブを選択します。
- 4 削除する非管理対象マシンを選択します。
- 5 [削除] をクリックします。

- 6 ユーザーが非管理対象マシンベースのデスクトップにログインしている場合は、セッションを終了するか、それともセッションをアクティブなままにするかを選択します。

オプション	説明
アクティブなままにする	アクティブなセッションをユーザーがログオフするまで続行します。View 接続サーバはこれらのセッションを追跡しません。
終了する	アクティブなセッションをすぐに終了します。

- 7 [OK] をクリックします。

非管理対象マシンがプールから削除されます。

登録済みのマシンを View から削除する

登録済みマシンを再度使用する予定がない場合は、View から削除できます。

View には [RDS ホスト] と [その他] の 2 つのタイプの登録済みのマシンがあります。非管理対象マシンはその他のカテゴリに含まれます。非管理対象マシンには、Horizon 6 for Linux 仮想マシンなど、vCenter Server により管理されていない物理コンピュータおよび仮想マシンが含まれます。これらを使用して、vCenter Server 仮想マシンを含まない手動デスクトップ プールが形成されます。

削除した登録済みのマシンは、View で使用できなくなります。マシンを再度使用できるようにするには、View Agent を再インストールする必要があります。

前提条件

削除する登録済みマシンが、どのデスクトップ プールでも使用されていないことを確認します。

手順

- 1 View Administrator で、[View 構成] - [登録済みのマシン] を選択します。
- 2 [その他] タブをクリックします。
- 3 1 つ以上のマシンを選択し、[削除] をクリックします。

選択できるマシンは、デスクトップ プールで使用されていないものだけです。

- 4 [OK] をクリックして確定します。

Linux 仮想マシンで View Agent をアンインストールする

Linux 仮想マシンで View Agent をアンインストールするには、`uninstall_viewagent.sh` スクリプトを使用します。また、マシンから Horizon 6 for Linux の構成ファイルを削除する必要があります。

前提条件

- マシンがデスクトップ プールおよび Horizon 6 から削除されていることを確認します。 [手動デスクトップ プールからの非管理対象マシンを削除する](#) および [登録済みのマシンを View から削除する](#) を参照してください。

手順

- 1 仮想マシンでターミナル ウィンドウを開き、View Agent のアンインストール スクリプトを実行します。

例：

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

このスクリプトによって、View Agent のプロセスが停止し、View Agent サービスが削除され、View Agent のインストール ディレクトリ `/usr/lib/vmware/viewagent` が削除されます。

- 2 `/etc/vmware/` ディレクトリにある Horizon 6 for Linux の構成ファイルを手動で削除します。

View Agent を再インストールする予定がある場合（たとえば、さらに新しい View Agent バージョンにアップグレードする場合）、これらの構成ファイルを削除する必要はありません。

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

Linux 仮想マシンに View Agent を再インストールする

既存の Horizon 6 for Linux の構成を使用するか、新規構成で再インストールするかどうかによって、Linux 仮想マシンへ View Agent を再インストールするアプローチは異なります。

Horizon 6 for Linux の構成ファイルが仮想マシンに存在し、マシンが View 接続サーバで登録されている場合は、既存の構成で View Agent を再インストールできます。

構成ファイルがマシンから削除されている、あるいは登録済みのマシン名が View 接続サーバから削除されている場合、ローカルの構成ファイルと View 接続サーバの登録済みのマシン名の両方が削除されていることを確認する必要があります。次に、新しい構成で View Agent の新規インストールを実行できます。

前提条件

- Horizon 6 for Linux の構成ファイルがマシンに存在しているかどうかを確認します。

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

- Linux 仮想マシンが View 接続サーバに登録されているかどうかを確認します。View Administrator で、[View 構成] - [登録済みのマシン] を選択して、[その他] タブを選択します。
- 構成ファイルと登録済みのマシン名が削除されている場合、View Agent の新規インストールを実行する手順について理解しておいてください。[Linux 仮想マシンに View Agent をインストールする](#)を参照してください。

手順

- ◆ 既存の Horizon 6 for Linux の構成で再インストールする場合、パラメータを指定せずに View Agent のインストール スクリプトを実行できます。

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

View Agent インストーラは View 接続サーバの既存の構成ファイルとマシン登録を再利用します。インストーラによってマシンは再度登録されません。

- ◆ 新しい Horizon 6 for Linux 構成を使用してインストールするには、必須のパラメータを指定して View Agent のインストール スクリプトを実行します。

インストールの手順は、新規インストールの手順と同じです。インストーラによって、View 接続サーバにマシンが登録されます。

vSphere からの Linux デスクトップの電源操作の実行方法

vSphere から Linux デスクトップの電源操作を実行する必要がある場合があります。

[パワーオフ] や [リセット] の操作は、データが損失する、あるいは、View Agent が正しく再起動しないことがあるため、使用しないでください。代わりに、[ゲストをシャットダウン] または [ゲストを再起動] の操作を使用します。

Horizon 6 for Linux ソフトウェアに関する情報を収集する

Horizon 6 for Linux ソフトウェアを管理およびトラブルシューティングするために、Linux 仮想マシンにインストールされている View Agent に関する情報を収集できます。

手順

- ◆ Linux マシンで実行している View Agent のバージョンを特定します。

次のコマンドを実行します。

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

出力に View Agent のバージョンが表示されます。例：

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

x.x は製品バージョンで、*yyyyyyy* はビルド番号です。

- ◆ View Agent ソフトウェアに関する情報を収集します。

ファイル	場所
View Agent インストールディレクトリ	/usr/lib/vmware/viewagent
ログ	/var/log/vmware/
View Agent の構成	/etc/vmware/viewagent-config.txt
マシン構成	/etc/vmware/viewagent-machine.cfg

- ◆ View Agent サービスを停止、開始、または再起動します。

Linux ディストリビューションでは、View Agent サービスにこれらの操作を実行できます。たとえば、Ubuntu 12.04 では、次のコマンドを実行できます。

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


Horizon 6 for Linux デスクトップの一括デプロイ

7

View 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 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 仮想マシンで View Agent をインストールするサンプル スクリプト](#)
- [SSH を使用して Linux 仮想マシンで View Agent をインストールするサンプル 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 View Agent をアップグレードするサンプル スクリプト](#)
- [SSH を使用して Linux 仮想マシンで View 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](#)
- [接続サーバの LDAP データベースからマシンを削除するサンプル スクリプト](#)

Linux デスクトップの一括デプロイの概要

Linux デスクトップをデプロイするには、いくつかの手順を実行します。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"> ■ 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。 ■ 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。
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 View 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 View 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	View 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。

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 クローン仮想マシンを Active Directory (AD) ドメインに参加させます。

[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 仮想マシンに View Agent をインストールします。

Linux 仮想マシンで View Agent をインストールするサンプル スクリプトまたは SSH を使用して Linux 仮想マシンで View Agent をインストールするサンプル スクリプトを参照してください。

- 6 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。

- 7 デスクトップ プールを作成します。

Linux 仮想マシンを含むデスクトップ プールを作成するを参照してください。

Linux 仮想マシンで View Agent をアップグレードする必要がある場合があります。仮想マシンで View Agent をアップグレードするサンプル スクリプトについては、Linux デスクトップ マシンで View Agent をアップグレードするサンプル スクリプトまたは SSH を使用して Linux 仮想マシンで View Agent をアップグレードするサンプル スクリプト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。View からデスクトップ マシンを削除するには、接続サーバの LDAP データベースからマシンを削除するサンプル スクリプトを参照してください。

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 6 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。View でのデスクトップ プールとアプリケーション プールの設定 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について推奨されるビデオ メモリ (vRAM) の値を理解しておきます。 [Horizon 6 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。
- 必要な場合は、View 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。 [4 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。 [5 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。

推奨値については、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 12.04 では、2048 MB の vMemory と 2 台の vCPU を構成することが推奨されています。

- 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。

推奨値については、Horizon 6 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 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、View 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 View 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 ではない場合、VMware Tools をインストールします。
RHEL 7 には、Open VM Tools がデフォルトでインストールされます。
- 8 Linux ディストリビューションが RHEL 7 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。
- 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 View 構成ファイルを使用して追加構成を実行します。
- 12 グラフィックスをセットアップする手順を実行します。
- 13 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,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-002,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-003,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-004,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
linux-005,Ubuntu1204x64,linuxagent,datastore1,10.117.44.172,snapshot1,TRUE
```

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBS
TR($input))
    }
    else
    {
        $input = Read-Host
    }

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

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

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

function Disable_VM_Console()
{
    Param($VMToDisableConsole)
```

```

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

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

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

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

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

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

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

```

```

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

#Read input CSV file
$csvData = Import-CSV $csvFile
#$csvData = Import-CSV $csvFile -
header("VMName","Parentvm","CustomSpec","Datastore","Host","FromSnapshot","DeleteIfPresent")
foreach ($line in $csvData)
{
    "`n-----"
    $VMName = $line.VMName
    write-host -ForegroundColor Yellow "VM: $VMName`n"

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

```



```
# Get the task object
$task = Get-Task | where { $_.id -eq $task }
#Wait for the taks to Complete
Wait-Task -Task $task

$newvm = Get-vm $destVMName
$customSpec = Get-OSCustomizationSpec $cSpec
Set-vm -OSCustomizationSpec $cSpec -vm $newvm -confirm:$false
if ($disableVMConsole -eq "yes")
{
    Disable_VM_Console($destVMName)
}
# Start the VM
Start-VM $newvm
}
Disconnect-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 が動作し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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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
        $input =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($input))
    }
    else
    {
        $input = Read-Host
    }

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

```
#----- 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
"-----"
HostAdmin = GetInput -prompt 'Your ESXi host admin user name, such as root' -IsPassword $false
HostPassword = GetInput -prompt "Your ESXi admin user password" -IsPassword $true
"-----"
GuestUser = GetInput -prompt 'Your VM guest OS user name' -IsPassword $false
GuestPassword = GetInput -prompt 'Your VM guest OS user password' -IsPassword $true
"-----"
AdUser = GetInput -prompt 'Type the AD user name to join the AD' -IsPassword $false
""
"\nPlease type the AD user password."
"Plase note that special character in password may not work with the script"
AdUserPassword = GetInput -prompt 'Your AD user password' -IsPassword $true
"-----"

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

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

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

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

$destFolder = "/home/$guestUser/"

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

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

```
}

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

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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

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

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

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

function Check_SSH_Client
{
    Param($IsPlink, $IsPSCP)
    if ($IsPlink)
    {
        if (Test-Path ".\plink.exe")
        {
            write-host -ForegroundColor Yellow 'SSH client "plink.exe" found'
        }
        else
        {

```

```

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

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

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

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

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

#----- Handle input -----
"-----"
Check_SSH_Client -IsPlink $true -IsPSCP $false
"-----"

```

```

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
"-----"
$adUser = GetInput -prompt 'Type the AD user name to join the AD' -IsPassword $false
""
"Please 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 仮想マシンで View Agent をインストールするサンプル 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) に View Agent をインストールできます。

このスクリプトでは、View 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。

スクリプトを実行する前に、View Agent が仮想マシンにすでにインストールされていないことを確認します。

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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。

- View 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 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
- View Agent の tar ボールのパス
- View 接続サーバの IP アドレス
- View 接続サーバ管理者のログイン名
- View 接続サーバ管理者のパスワード
- View 接続サーバ管理者のドメイン名
- Kerberos が認証方式であり、View Agent が Linux マシンを View 接続サーバに登録する場合の Active Directory サーバのアドレス

スクリプトのコンテンツ

```
<#
Install Linux Agent

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

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

    [Console]::ResetColor()
    return $input
}
#----- Handle Input -----
"-----"
$acceptEULA = GetInput -prompt 'Accept Linux View Agent EULA in tar bundle ("yes" or "no")' -
IsPassword $false
if ($acceptEULA -ne "yes")
{
    write-host -ForeGroundColor Red "You need accept the EULA with 'yes'(case sensitive)"
    exit
}
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
```

```

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 View Agent tar ball path. Please take care the installer
arch' -IsPassword $false
"-----"
$brokerAddress = GetInput -prompt 'Type the View Connection Server address' -IsPassword $false
$brokerAdminName = GetInput -prompt 'Type the View Connection Server Admin user name' -IsPassword
$false
"$nPlease type the View Connection Server Admin user password."
"Plase note that special character in password may not work with the script"
$brokerAdminPassword = GetInput -prompt 'Your broker admin password' -AsSecureString -IsPassword
$true
$domainName = GetInput -prompt 'Type the View Connection Server Admin user domain name' -IsPassword
$false
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
$skdc = GetInput -prompt 'AD address for Kerberos authentication to register. Please type Enter to
skip for MD5-Digest' -IsPassword $false
"-----"
#$csvFile = Read-Host 'Csv File '
$csvFile = '.\CloneVMs.csv'

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

#check if file exists
if (!(Test-Path $csvFile))
{
    write-host -ForegroundColor Red "CSV File not found"
    exit
}
#----- Functions -----
function GetSourceInstallerMD5()
{
    $agentInstallerPath = Convert-Path $agentInstaller;
    $md5 = New-Object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider;
    $md5HashWithFormat =
[System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($agentInstallerPath)))
;

```

```

    $md5Hash = ($md5HashWithFormat.replace("-", "").ToLower());
    return $md5Hash;
}
#----- Main -----
#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

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

```

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

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

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

    $cmd = "sudo shutdown -r +1&"
    Write-Host "Reboot to apply the View 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 View Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

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

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

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

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

スクリプトの実行が完了したら、View Administrator にログインして、[View 構成] - [登録済みのマシン]に移動して、[その他] タブをクリックして、Linux デスクトップ マシンを表示できます。

SSH を使用して Linux 仮想マシンで View Agent をインストールする 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) に View Agent をインストールできます。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。

スクリプトを実行する前に、View Agent が仮想マシンにすでにインストールされていないことを確認します。

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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。

- View 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
- 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
- View Agent の tar ボールのパス
- View 接続サーバの IP アドレス
- View 接続サーバ管理者のログイン名
- View 接続サーバ管理者のパスワード
- View 接続サーバ管理者のドメイン名
- Kerberos が認証方式であり、View Agent が Linux マシンを View 接続サーバに登録する場合の Active Directory サーバのアドレス

スクリプトのコンテンツ

```
<#
Install Linux Agent via SSH

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

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

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

```

```

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

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

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

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

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

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

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

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

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

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

```



```

#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

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

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

```

```

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

```

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

-----
Type the View Connection Server address: 10.117.45.93
Type the View Connection Server Admin user name: viewadmin
Please type the View Connection Server Admin user password.
Please note that special character should be escaped. For example, $ should be \$
Your broker admin password: *****
Type the View Connection Server Admin user domain name: domain1
Install the Smartcard redirection feature ("yes" or "no"): no
AD address for Kerberos authentication to register. Please type Enter to skip for MD5 Digest:
10.117.45.1

```

スクリプトの実行が完了したら、View Administrator にログインして、[View 構成] - [登録済みのマシン]に移動して、[その他] タブをクリックして、Linux デスクトップ マシンを表示できます。

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

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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 View 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 View Agent をアップグレードできます。

このスクリプトでは、View 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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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。

- View 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
- View Agent の tar ボールのパス

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 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
"-----"
$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 View Agent tar ball path. Please take care the installer
arch' -IsPassword $false
"-----"
$installSmartcard = GetInput -prompt 'Install the Smartcard redirection feature ("yes" or "no")' -
IsPassword $false
if (($installSmartcard -ne "yes") -AND $installSmartcard -ne "no")
{
    write-host -ForegroundColor Red "You need select 'yes' or 'no'(case sensitive)"
    exit
}
"-----"

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

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

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

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

```

```

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

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

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

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

$destFolder = "/home/$guestUser/"

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

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

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

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

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

```

```

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

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

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

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

    $cmd = "sudo shutdown -r +1&"
    Write-Host "Reboot to apply the View 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 View Agent EULA in tar bundle ("yes" or "no"): yes
Your vCenter address: 10.117.44.17
Your vCenter admin user name: administrator
Your vCenter admin user password: *****

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

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

```

```
Your VM guest OS user password: *****
```

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

```
Install the Smartcard redirection feature ("yes" or "no"): no
```

SSH を使用して Linux 仮想マシンで View 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 View 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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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。

- View 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
- View Agent の tar ボールのパス

スクリプトのコンテンツ

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

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

```

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

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

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

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

```



```

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

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

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

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

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

```

```

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

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

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

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

#Get installer MD5Sum
$installerMD5Hash = GetSourceInstallerMD5;

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

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

$destFolder = "/home/$guestUser/"

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

```

```

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

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

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

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

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

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

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

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

```

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

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。View からマシンを削除するには、サンプル スクリプト [接続サーバの LDAP データベースからマシンを削除するサンプル スクリプト](#) を参照してください。

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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 -----
"-----"
$vcAddress = GetInput -prompt "Your vCenter address" -IsPassword $false
$vcAdmin = GetInput -prompt "Your vCenter admin user name" -IsPassword $false
$vcPassword = GetInput -prompt "Your vCenter admin user password" -IsPassword $true
"-----"

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

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

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

```

```

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

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

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

Disconnect-VIServer $vcAddress -Confirm:$false
exit

```

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。

接続サーバの LDAP データベースからマシンを削除するサンプル スクリプト

次のサンプル スクリプトをカスタマイズおよび使用して、デスクトップ プールにないマシンを接続サーバの LDAP データベースから削除できます。

これは PowerShell スクリプトであり、インストールのために PowerCLI は必要ありません。スクリプトを実行するには、View 接続サーバに管理者としてログインして、スクリプトと入力ファイルを作成し、PowerShell ウィンドウからスクリプトを実行します。入力ファイルに記述されているマシンがデスクトップ プールにある場合、このスクリプトはメッセージを表示し、LDAP データベースからマシンを削除しません。

注: スクリプトの実行によって生じた変更を元に戻すことが求められることがあるため、スクリプトを実行する前に、接続サーバの仮想マシンのスナップショットを取得します。

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 版を使用してください。HTML 版は https://www.vmware.com/support/pubs/view_pubs.html にある Horizon 6 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

入力ファイルのタイプは、csv であり、削除するマシンの名前が含まれます。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。

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

サンプル スクリプトでは、入力ファイルの名前が DeleteMachineItems.csv であり、このファイルがスクリプトと同じフォルダに配置されていることを前提としています。

スクリプトのコンテンツ

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

$csvFile = '.\DeleteMachineItems.csv'

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

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

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

```

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

Linux デスクトップのトラブルシューティング

8

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 6 for Linux マシンの診断情報を収集する](#)
- [Linux マシンでの View Agent 登録失敗のトラブルシューティング](#)
- [Linux マシンの到達できない View Agent のトラブルシューティング](#)
- [応答しない Linux マシンの View Agent のトラブルシューティング](#)
- [TCP 接続の受信を許可するための Linux ファイアウォールの構成](#)

Horizon 6 for Linux マシンの診断情報を収集する

VMware のテクニカル サポートが Horizon 6 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。

Linux マシンでの View Agent 登録失敗のトラブルシューティング

View Agent インストーラが View 接続サーバへの Linux 仮想マシンの登録に失敗する。

問題

View Agent を Linux 仮想マシンにインストールするときに、エージェントの登録に失敗しました というエラーが表示されます。

原因

View Agent インストーラは、次の理由で View 接続サーバへの Linux 仮想マシンの登録に失敗する場合があります。

- View Agent のインストール時に指定された View 接続サーバの FQDN、ドメイン名、ユーザー名、またはパスワードが不正であった。
- View 管理者ユーザーが、View Administrator で Agent 登録管理者 または 管理者 ロールを持っていない。特に、View Agent を View 接続サーバに登録するには、エージェントを登録 の権限が必要です。Agent 登録管理者 は、この最小権限を提供する制限付きのロールです。
- View 管理者パスワードに、\$ などの特殊文字がバックスラッシュでエスケープされずに含まれている。
- 管理者の Active Directory (AD) アカウントで元に戻せるパスワードの暗号化が有効になっていなかった。このポリシーの有効化の詳細については、[暗号化を元に戻せるパスワードの有効化](#)を参照してください。

解決方法

- 1 正しい View 接続サーバ FQDN、ドメイン名、ユーザー名、およびパスワードになっているかを確認します。
- 2 View 管理者ユーザーに View Agent を登録できる権限があることを確認します。
 - a View Administrator で、[View 構成] - [管理者] を選択します。
 - b [管理者とグループ] タブで、View Agent のインストール時に指定した View 管理者ユーザーを選択します。
 - c ユーザーが Agent 登録管理者 または 管理者 ロールを持っていることを確認します。
- 3 View 管理者パスワードに、特殊文字が含まれている場合には、バックスラッシュで特殊文字をエスケープします。

たとえば、`ab\ $cdef` のようにエスケープします。
- 4 Linux マシンで View Agent の登録ログ ファイルを開き、エラー メッセージを探します。

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

- 5 認証メカニズムが DIGEST-MD5 である ([-k] オプションが指定されていない) 場合、指定したユーザーで Active Directory ポリシーの [暗号化を元に戻せる状態でパスワードを保存する] が有効になっていることを確認します。
- 6 認証メカニズムが Kerberos である場合、[-k] オプションで Active Directory サーバの正しいアドレスが指定されていることと、[-b] オプションで接続サーバインスタンスのフルコンピュータ名 (ホスト名やドメイン名など) が指定されていることを確認します。
- 7 Linux 仮想マシンに View Agent を再度インストールします。

Linux マシンの到達できない View Agent のトラブルシューティング

View 接続サーバは Linux 仮想マシン上の View Agent と通信できません。

問題

View Administrator で、Linux 仮想マシンが エージェントに到達できません として表示されます。View Agent がインストールされ、View Agent サービスが Linux マシンで正常に起動した後に、この問題は発生します。

原因

Linux マシンが View 接続サーバ インスタンスの FQDN を解決できないことが原因の 1 つとして考えられます。

インストール後に、View Agent は View 接続サーバ インスタンスの FQDN のリストを構成ファイル `/etc/vmware/viewagent-machine.cfg` に保存します。View Agent をインストールするときに `-b` パラメータで View 接続サーバ インスタンスの IP アドレスを指定している場合であっても、View Agent は FQDN を使用します。

2 番目の原因としては、Linux マシン自体のホスト名を解決できないことが考えられます。

解決方法

- 1 Linux マシンで View Agent のログ ファイルを開きます。

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

- 2 View 接続サーバ インスタンスのホスト名を解決できないことを示すメッセージを探します。

例：

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

- 3 DNS サービスが Linux マシンで適切に構成されていることを確認します。

DNS を構成する手順は、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のドキュメントを参照してください。

回避策として、View 接続サーバの FQDN を Linux マシンの `/etc/hosts` ファイルに手動で追加することができます。しかし、この方法を本番環境で使用することは推奨されません。適切な解決策は、Linux マシンが View 接続サーバの FQDN を解決できるように DNS サービスを構成することです。

これらの手順を実行したら、View 接続サーバのホスト名に ping を送信して、解決できるかどうかを確認します。例：

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

- 4 問題が継続する場合、Linux マシンのシステム ホスト名が解決可能であることを確認します。

たとえば、CentOS では、次の手順を実行できます。

- a ターミナル ウィンドウを開き、`hostname` コマンドを実行します。

マシンのホスト名が表示されます。

- b `etc/hosts` ファイルを開いて、ホスト名を追加します。

```
su
nano /etc/hosts
# Add the hostname:
127.0.0.1 <ホスト名>
```

- c このホスト名に `ping` を送信し、解決できるかどうかを検証します。

```
ping <ホスト名>
```

注: `/etc/resolv.conf` ファイルを手動で編集して Linux マシンで DNS サーバを指定している場合、この設定はいくつかの Linux ディストリビューションで失われることがあります。DNS サーバを `/etc/hosts` ファイルで指定していると、この設定は維持されます。DNS とホスト名の構成に関する詳細な手順については、お使いの 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。

応答しない Linux マシンの View Agent のトラブルシューティング

Linux 仮想マシンにインストールされた View Agent が応答しない。

問題

Horizon 6 for Linux デスクトップにアクセスできず、View Agent が応答していない。

原因

View Agent サービスが実行していない、あるいは X セッションが Linux マシンでアクティブになっていない可能性があります。

解決方法

- 1 Linux 仮想マシンで、ターミナル ウィンドウを開いて、View Agent サービスを停止して再起動します。

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

- 2 X セッションがアクティブであることを確認します。

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

- 3 X11 Simple VNC サーバを仮想マシンにインストールして、VNC を介して X セッションがアクティブであることを確認します。

TCP 接続の受信を許可するための Linux ファイアウォールの構成

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 View 接続サーバから受信する 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、View 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 View 接続サーバの 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。