新機能

2106の新機能

Linux VDAバージョン2106には、以下の新機能と機能強化が含まれています。

Linux VDAのセルフアップデート

この機能は、Linux VDAソフトウェアを即座に、またはスケジュールされた時間に自動的に更新するのに役立ちます。特に、Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成する場合、AzureのVMに対する管理者権限がないため、この機能は非常に役立ちます。詳細については、「Linux VDAのセルフアップデート」を参照してください。

Nutanix AHVでのMachine Creation Services™ (MCS)のサポート

MCSを使用してNutanix AHV上にLinux VMを作成できるようになりました。詳細については、「MCSを使用してNutanix AHV上にLinux VMを作成する」を参照してください。

MCSで作成されたマシンをWindowsドメインに参加させるためのPBISサポート

MCSで作成されたマシンをWindowsドメインに参加させるためのPBISサポートを追加しました。詳細については、「Machine Creation Services (MCS)を使用してLinux VMを作成する」を参照してください。

HDX™ウェブカメラビデオ圧縮のサポート

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDXウェブカメラビデオ圧縮を使用してウェブカメラを使用できるようになりました。HDXウェブカメラビデオ圧縮は、**最適化された**ウェブカメラモードとも呼ばれます。このタイプのウェブカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDXウェブカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。この機能はデフォルトで有効になっています。可能な場合は常にHDXウェブカメラビデオ圧縮を使用することをお勧めします。詳細については、「HDXウェブカメラビデオ圧縮」を参照してください。

グラフィック品質スライダー

仮想Linuxセッションで実行されるグラフィックステータスインジケーターツールに、グラフィック品質スライダーが含まれるようになりました。このスライダーは、画質とインタラクティブ性の適切なバランスを見つけるのに役立ちます。詳細については、「グラフィック品質スライダー」を参照してください。

CGPセッション再接続中のSecure Browserログオンユーザーに対するポリシー評価のサポート

Linux VDAは、CGPセッション再接続中に、Secure Browserサービスで設定したユーザー固有のポリシーを評価および適用できるようになりました。以下のポリシーがサポートされています。

- クリップボード
  - クライアントクリップボードリダイレクト
  - クリップボード選択更新モード
  - プライマリ選択更新モード
- 印刷
  - クライアントプリンターリダイレクト
  - PDFユニバーサルプリンターの自動作成
- ファイル転送
  - デスクトップとクライアント間のファイル転送を許可する
  - デスクトップからファイルをダウンロード
  - デスクトップにファイルをアップロード

RHEL 8.3およびUbuntu 18.04のLinuxストリーミングサポート

以前は実験的機能として利用可能でしたが、RHEL 8.3およびUbuntu 18.04向けのLinuxストリーミングは、完全にサポートされるようになりました。Citrix ProvisioningでLinuxストリーミング機能を使用すると、Citrix Virtual Apps and Desktops環境にLinux仮想デスクトップを直接プロビジョニングできます。詳しくは、「Citrix Provisioningを使用してLinux VMを作成する」を参照してください。

修正された問題

Linux Virtual Delivery Agent 2104 以降、以下の問題が修正されています。

- 複数のJavaバージョンを使用している場合、Linux VDAがDelivery Controller™ に登録できないことがあります。
- HDX™ 3D Pro が有効な場合、easy install を使用した Linux VDA のインストールが失敗することがあります。
- CentOS 8.3 でフェデレーション認証サービスを使用するように Linux VDA を構成しようとすると、ctxfascfg.sh が失敗することがあります。
- Linux `XDPing` ツールが SUSE Linux Enterprise Server (SLES) バージョン 12.5 で動作しないことがあります。

既知の問題

このリリースで以下の問題が確認されています。

- シームレスではない公開アプリケーションが、起動後すぐに終了する場合があります。この問題は、Mutterがmutter-3.28.3-4より新しいバージョンにアップグレードされた後に発生します。この問題を回避するには、mutter-3.28.3-4以前を使用してください。
- HDX 3D Proを有効にせずにNVIDIA GRID 3Dカードを使用すると、Linux VDAが期待どおりに動作しません。この問題は、RHEL 7.7以前、SUSE 12.5以前、およびUbuntu 16.04で発生します。これは、これらのLinuxディストリビューションのグラフィックシステムでは、複数のOpenGLライブラリが共存できないためです。
- ファイルのダウンロード中に予期しないウィンドウが表示されます。このウィンドウはファイルのダウンロード機能には影響せず、しばらくすると自動的に消えます。
- PulseAudioのデフォルト設定により、サウンドサーバープログラムは20秒間アイドル状態が続くと終了します。PulseAudioが終了すると、オーディオが機能しなくなります。この問題を回避するには、/etc/pulse/daemon.confファイルでexit-idle-time=-1を設定します。
- SUSE 12.5の `libtcmalloc 4.3.0` により、プロセスが予期せず終了する場合があります。
- Ubuntu 16.04およびSUSE 12.5 VDAで、 `ctxhdx` サービスが予期せず終了する場合があります。この問題は、GNU C Library (`glibc`) バージョン2.22から2.24で発生します。この問題は `glibc 2.25` で修正されています。SUSE 12.5ディストリビューションを使用している場合は、SUSEが提供するパッチをインストールして問題を修正できます。Linux VDAのリリース時点では、Ubuntu 16.04向けの修正はありません。
- SSL暗号化が有効でセッションの信頼性が無効になっている場合、Citrix Workspace™ app for Linuxでセッションを起動できません。

- `indicator-datetime-service`プロセスは$TZ環境変数を消費しません。クライアントとセッションが異なるタイムゾーンにある場合、Ubuntu 16.04 Unity DesktopのUnityパネルにはクライアントの時刻が表示されません。
- Ubuntuグラフィック：HDX™ 3D Proでは、Desktop Viewerのサイズ変更後にアプリケーションの周囲に黒いフレームが表示されたり、背景が黒く表示されたりする場合があります。
- Linux VDAの印刷リダイレクトによって作成されたプリンターが、セッションからログオフしても削除されない場合があります。
- ディレクトリに多数のファイルやサブディレクトリが含まれている場合、CDMファイルが欠落します。この問題は、クライアント側にファイルやディレクトリが多すぎる場合に発生する可能性があります。
- 英語以外の言語では、UTF-8エンコーディングのみがサポートされています。
- セッションローミング中にCitrix Workspace app for AndroidのCAPS LOCKの状態が反転する場合があります。既存の接続をCitrix Workspace app for Androidにローミングすると、CAPS LOCKの状態が失われることがあります。回避策として、拡張キーボードのShiftキーを使用して大文字と小文字を切り替えてください。
- Citrix Workspace app for Macを使用してLinux VDAに接続すると、ALTキーを使用したショートカットキーが常に機能するとは限りません。Citrix Workspace app for Macは、デフォルトで左右のOptions/Altキーの両方に対してAltGrを送信します。この動作はCitrix Workspace appの設定内で変更できますが、結果はアプリケーションによって異なります。
- Linux VDAがドメインに再参加すると、登録に失敗します。再参加により、新しいKerberosキーセットが生成されます。しかし、Brokerは以前のKerberosキーセットに基づいた、キャッシュされた古いVDAサービスチケットを使用する可能性があります。VDAがBrokerに接続しようとすると、BrokerはVDAへの戻りのセキュリティコンテキストを確立できない可能性があります。通常の症状は、VDAの登録が失敗することです。

  この問題は、VDAサービスチケットの有効期限が切れ、更新されたときに最終的に解決されます。しかし、サービスチケットは有効期間が長いため、解決には時間がかかる場合があります。

  回避策として、Brokerのチケットキャッシュをクリアします。Brokerを再起動するか、管理者としてコマンドプロンプトからBrokerで次のコマンドを実行します。

  ```
  klist -li 0x3e4 purge
  ```

  このコマンドは、Citrix Broker Serviceが実行されているNetwork Serviceプリンシパルによって保持されているLSAキャッシュ内のすべてのサービスチケットをパージします。他のVDA

や、場合によっては他のサービスのサービスチケットも削除されます。しかし、これは無害です。これらのサービスチケットは、必要に応じてKDCから再取得できます。

- オーディオのプラグアンドプレイはサポートされていません。ICA®セッションでオーディオの録音を開始する前に、オーディオキャプチャデバイスをクライアントマシンに接続できます。オーディオ録音アプリケーションの起動後にキャプチャデバイスが接続された場合、アプリケーションが応答しなくなる可能性があり、再起動する必要があります。録音中にキャプチャデバイスが取り外された場合、同様の問題が発生する可能性があります。
- Windows向けCitrix Workspaceアプリでは、オーディオ録音中に音声の歪みが発生する可能性があります。

サードパーティ通知

Linux Virtual Delivery Agent バージョン 2106(/ja-jp/linux-virtual-delivery-agent/2106/downloads/lvda-v2106-third-party-notices.pdf) (PDFダウンロード)

このLinux VDAのリリースには、ドキュメントで定義されている条件に基づいてライセンス供与されたサードパーティ製ソフトウェアが含まれる場合があります。

廃止予定

この記事のお知らせは、段階的に廃止されるプラットフォーム、Citrix®製品、および機能について事前にお知らせし、お客様がタイムリーなビジネス上の意思決定を行えるようにすることを目的としています。Citrixは、それらがいつ廃止されるかを決定するために、お客様の使用状況とフィードバックを監視しています。お知らせは、その後のリリースで変更される可能性があり、すべての非推奨の機能や機能が含まれていない場合があります。 製品ライフサイクルサポートの詳細については、「製品ライフサイクルサポートポリシー」の記事を参照してください。

廃止予定と削除

次の表は、非推奨または削除されたプラットフォーム、Citrix製品、および機能を示しています。**非推奨**の項目は、すぐに削除されるわけではありません。Citrixは、このリリースでは引き続きサポートしますが、将来のCurrent Releaseで削除されます。 **削除済み**の項目は、Linux VDAで削除されたか、サポートされなくなりました。

項目	非推奨が発表されたバージョン	削除時期
RHEL 7.7、CentOS 7.7のサポート	2006	2009

項目	非推奨が発表されたバージョン	削除時期
SUSE 12.3のサポート	2006	2006
RHEL 6.10、CentOS 6.10のサポート	2003	2003
RHEL 6.9、CentOS 6.9のサポート	1909	1909
RHEL 7.5、CentOS 7.5のサポート	1903	1903
RHEL 7.4、CentOS 7.4のサポート	1811	1811
RHEL 6.8、CentOS 6.8のサポート	1811	1811
RHEL 7.3、CentOS 7.3のサポート	7.18	7.18
RHEL 6.6、CentOS 6.6のサポート	7.16	7.16
SUSE 11.4	7.16	7.16

システム要件

Linuxディストリビューション

> **注:**
>
> ここに記載されていないコンポーネント（Citrix Workspace™アプリなど）のシステム要件については、それぞれのドキュメントセットに記載されています。

> Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Core Runtime 3.1をインストールしてください。

> このCurrent Release (CR)をLong Term Service (LTSR)環境で使用する方法、およびその他のFAQについては、Knowledge Center articleを参照してください。

Linux VDAは、以下のLinuxディストリビューションをサポートしています。

- SUSE リナックス エンタープライズ:
  - サーバー 12 サービスパック 5 + SUSE Linux Enterprise Workstation Extension 12 SP5
  - サーバー 12 Service Pack 5
- Red Hat エンタープライズ リナックス
  - ワークステーション 8.3
  - ワークステーション 8.2
  - ワークステーション 8.1
  - ワークステーション 7.9
  - ワークステーション 7.8
  - サーバー 8.3
  - サーバー 8.2

    - サーバー 8.1
    - サーバー 7.9
    - サーバー 7.8
- セントオーエス リナックス
    - CentOS 8.3
    - CentOS 8.2
    - CentOS 8.1
    - CentOS 7.9
    - CentOS 7.8
- ウブントゥ リナックス
    - ウブントゥ デスクトップ 20.04
    - ウブントゥ サーバー 20.04
    - ウブントゥ デスクトップ 18.04
    - ウブントゥ サーバー 18.04
    - ウブントゥ ライブサーバー 18.04
    - ウブントゥ デスクトップ 16.04
    - ウブントゥ サーバー 16.04
- デビアン リナックス
    - デビアン 10.7

> **注:**

> CentOSプロジェクトはCentOS Streamに焦点を移します。RHEL 8の再構築版であるCentOS Linux 8は、2021年末に終了します。CentOS Streamはその後も継続され、Red Hat Enterprise Linuxのアップストリーム（開発）ブランチとして機能します。詳細については、`<https://www.redhat.com/en/blog/centos-stream-building-innovative-future-enterprise-linux>`を参照してください。

> Linux VDAは、LinuxマシンをMicrosoft Active Directory (AD) と統合するためのいくつかの方法をサポートしています。

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
RHEL 8.3	はい	はい	はい	はい	いいえ
CentOS 8.3	はい	はい	はい	はい	いいえ
RHEL 8.2	はい	はい	はい	はい	いいえ
CentOS 8.2	はい	はい	はい	はい	いいえ
RHEL 8.1	はい	はい	はい	はい	いいえ
CentOS 8.1	はい	はい	はい	はい	いいえ
RHEL 7.9	はい	はい	はい	はい	はい
CentOS 7.9	はい	はい	はい	はい	はい
RHEL 7.8	はい	はい	はい	はい	はい
CentOS 7.8	はい	はい	はい	はい	はい
ウブントゥ 20.04	はい	はい	はい	はい	はい
ウブントゥ 18.04	はい	はい	はい	はい	はい
ウブントゥ 16.04	はい	はい	はい	はい	はい
デビアン 10.7	はい	はい	はい	はい	いいえ
SUSE 12.5	はい	はい	はい	はい	はい

このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、以下の表を参照してください。詳細については、XorgModuleABIVersionsを参照してください。

LINUXディストリビューション	XORGバージョン
RHEL 8.3, CentOS 8.3	1.20.8
RHEL 8.2, CentOS 8.2	1.20.8
RHEL 8.1, CentOS 8.1	1.20.8
RHEL 7.9, CentOS 7.9	1.20
RHEL 7.8, CentOS 7.8	1.20
ウブントゥ 20.04	1.20
ウブントゥ 18.04	1.19
ウブンツ 16.04	1.18
デビアン 10.7	1.20
SUSE 12.5	1.19

UbuntuではHWEカーネルまたはHWE Xorgを使用しないでください。

RHEL 8.xおよびCentOS 8.xではPulseAudio 13.99を使用してください。

すべての場合において、サポートされるプロセッサアーキテクチャはx86-64です。

注:

- OSベンダーからのサポートが終了すると、Citrix®が問題の解決にできることが制限される場合があります。
- GNOMEおよびKDEデスクトップは、SUSE 12、RHEL 7、CentOS 7、RHEL 8、およびCentOS 8でサポートされています。Unityデスクトップは Ubuntu 16.04でサポートされています。GNOMEデスクトップはUbuntu 20.04、Ubuntu 18.04、およびDebian 10.7でサポートされています。MATEデスクトップは、Linux VDAがサポートするすべてのLinuxディストリビューションでサポートされていますが、SUSE 12.5は除きます。少なくとも1つのデスクトップがインストールされている必要があります。

シトリックス 仮想デスクトップ™

Linux VDAは、現在サポートされているすべてのバージョンのCitrix Virtual Apps and Desktopsと互換性があります。Citrix製品のライフサイクルに関する情報、およびCitrixが特定のバージョンの製品のサポートを終了する時期については、Citrix製品ライフサイクルマトリックスを参照してください。

Linux VDAの構成プロセスはWindows VDAとは若干異なります。ただし、どのDelivery Controller™ファームでもWindowsとLinuxの両方のデスクトップを仲介できます。

サポートされるホストプラットフォームと仮想化環境

- ベアメタルサーバー
- シトリックス ハイパーバイザー™
- VMware ESXおよびESXi
- マイクロソフト ハイパーV
- ニュータニックス AHV
- マイクロソフト アジュール リソースマネージャー
- アマゾン ウェブ サービス (AWS)
- グーグル クラウド プラットフォーム (GCP)

> **ヒント:**
>
> サポートされているプラットフォームのリストについては、ベンダーのドキュメントを参照してください。

> **注:**
>
> Azure、AWS、およびGCPは、Citrix Virtual Apps and Desktops™サービスとのみ互換性があります。MCSを使用して仮想マシンを作成する場合、ベアメタルサーバーはサポートされません。

Active Directory統合パッケージ

Linux VDAは、次のActive Directory統合パッケージまたは製品をサポートしています。

- サンバ ウィンバインド
- クエスト認証サービス v4.1 以降
- セントリファイ ダイレクトコントロール
- SSSD
- PBIS (RHEL 7、Ubuntu、Debianと互換性あり)

> **ヒント:**
>
> サポートされているプラットフォームのリストについては、Active Directory統合パッケージのベンダーのドキュメントを参照してください。

HDX™ 3D プロ

HDX 3D Proをサポートするには、以下のハイパーバイザーとNVIDIA GRID™ GPUが必要です。

ハイパーバイザー

- シトリックス ハイパーバイザー
- VMware ESX および ESXi
- ニュータニックス AHV

> **注:**
>
> これらのハイパーバイザーは、特定のLinuxディストリビューションと互換性があります。

GPU

Linux VDAは、GPUパススルー用に次のGPUをサポートします。

- NVIDIA グリッド - テスラ T4
- NVIDIA ジーティーエックス750Ti
- NVIDIA グリッド - Tesla M60
- NVIDIA GRID - K2
- NVIDIA グリッド - Tesla P40

- NVIDIA グリッド - テスラ P4
- NVIDIA グリッド - テスラ P100

Linux VDAは、vGPU用に次のGPUをサポートします。

- NVIDIA GRID - テスラ T4
- NVIDIA GRID - テスラ V100
- NVIDIA グリッド - テスラ M60
- エヌビディア グリッド — テスラ M10
- エヌビディア グリッド — テスラ P40
- エヌビディア グリッド — テスラ P4
- エヌビディア グリッド — テスラ P100

インストールの概要

Linux VDAをインストールするにはいくつかのオプションがあります。新規インストールを行うことも、以前の2つのバージョンおよびLTSRリリースからの既存のインストールをアップグレードすることもできます。

- 簡易インストール。マシンにLinux VDAパッケージをインストールした後、ctxinstall.shスクリプトを使用して実行環境を構成できます。詳細については、簡易インストールを参照してください。
- Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成する: Citrix Virtual Apps and Desktops Standard for Azureで、ドメイン参加済みおよびドメイン非参加の両方のLinux VDAを作成し、Microsoft Azureから任意のデバイスに仮想アプリとデスクトップを配信できます。詳細については、Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成するを参照してください。
- MCS。MCSを使用して、Linux VDAパッケージもインストールされているLinux VMをバッチで作成できます。詳細については、MCSを使用してLinux VMを作成するを参照してください。
- Citrix Provisioningを使用してLinux VMを作成する: Citrix Virtual Apps and Desktops環境でLinux仮想デスクトップを直接プロビジョニングできます。詳細については、Citrix ProvisioningのドキュメントのLinuxターゲットデバイスのストリーミングを参照してください。
- 手動インストール。以下の一般的な手順を使用してLinux VDAをインストールできます。バリエーションと特定のコマンドは、ディストリビューションごとに文書化されています。詳細については、RHEL/CentOS用Linux Virtual Delivery Agentのインストール、SUSE用Linux Virtual Delivery Agentのインストール、Ubuntu用Linux Virtual Delivery Agentのインストール、およびDebian用Linux Virtual Delivery Agentのインストールを参照してください。

  a. インストールの準備。
  b. ハイパーバイザーの準備。
  c. Linux仮想マシン（VM）をWindowsドメインに追加します。
  d. Linux VDAをインストールします。
  e. Linux VDAを構成します。
  f. Citrix Virtual AppsまたはCitrix Virtual Desktopsでマシンカタログを作成します。

g. Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成します。

XDPing (エックスディーピング)

Linux XDPingツールを使用して、Linux VDA環境における一般的な構成の問題を確認できます。詳細については、XDPingを参照してください。

前提条件として.NET Core Runtime 3.1をインストールする

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Core Runtime 3.1をインストールします。

.NET Core Runtime 3.1をインストールした後、**which dotnet**コマンドを実行してランタイムパスを見つけます。

コマンドの出力に基づいて、.NET Coreランタイムバイナリパスを設定します。たとえば、コマンドの出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

デリバリーコントローラーの構成

XenDesktop 7.6 およびそれ以前のバージョンでは、Linux VDA をサポートするために変更が必要です。これらのバージョンでは、ホットフィックスまたは更新スクリプトが必要です。インストールと検証の手順は、この記事で提供されています。

Delivery Controller™ の構成を更新する

XenDesktop 7.6 SP2 の場合、Hotfix Update 2 を適用して Linux Virtual Desktops 用の Broker を更新します。Hotfix Update 2 はこちらから入手できます。

CTX142438: Hotfix Update 2 - Delivery Controller 7.6 (32ビット) 用 – 英語

XenDesktop 7.6 SP2 より前のバージョンの場合、PowerShell スクリプト **Update-BrokerServiceConfig.ps1** を使用して Broker Service の構成を更新できます。このスクリプトは、次のパッケージで入手できます。

- シトリックス Linux VDA スクリプト.zip

ファーム内のすべての Delivery Controller で次の手順を繰り返します。

1. **Update-BrokerServiceConfig.ps1** スクリプトを Delivery Controller マシンにコピーします。
2. ローカル管理者として Windows PowerShell コンソールを開きます。

3. **Update-BrokerServiceConfig.ps1** スクリプトが含まれているフォルダーに移動します。
4. **Update-BrokerServiceConfig.ps1** スクリプトを実行します。

```
.\Update-BrokerServiceConfig.ps1
```

ヒント:

デフォルトでは、PowerShell は PowerShell スクリプトの実行を防止するように構成されています。スクリプトの実行に失敗した場合は、再試行する前に PowerShell 実行ポリシーを変更してください。

```
Set-ExecutionPolicy Unrestricted
```

Update-BrokerServiceConfig.ps1 スクリプトは、Linux VDA で必要とされる新しい WCF エンドポイントを使用して Broker Service 構成ファイルを更新し、Broker Service を再起動します。スクリプトは、Broker Service 構成ファイルの場所を自動的に決定します。元の構成ファイルのバックアップは、同じディレクトリに作成され、ファイル名に **.prelinux** が追加されます。

これらの変更は、同じ Delivery Controller ファームを使用するように構成されている Windows VDA のブローカー処理には影響しません。単一の Controller ファームで、Windows VDA と Linux VDA の両方のセッションをシームレスに管理およびブローカー処理できます。

注:

Linux VDA は暗号化に Secure ICA® をサポートしていません。Linux VDA で Secure ICA を有効にすると、セッションの起動に失敗します。

Delivery Controller の構成を確認する

必要な構成変更が Delivery Controller に適用されると、**EndpointLinux** 文字列が **%PROGRAMFILES%\\Citrix\\Broker\\Service\\BrokerService.exe.config** ファイルに5回出現します。

Windows コマンドプロンプトから、ローカル管理者としてログオンして確認します:

```
cd "%PROGRAMFILES%"\Citrix\Broker\Service\

findstr EndpointLinux BrokerService.exe.config
```

簡単インストール

簡単インストールは、Linux VDA バージョン 7.13 以降でサポートされています。この機能を使用すると、必要なパッケージをインストールし、構成ファイルを自動的にカスタマイズすることで、Linux VDA の実行環境をセットアップできます。

簡単インストールの使用

この機能を使用するには、次の手順を実行します。

1. 構成情報と Linux マシンを準備します。
2. Linux VDA パッケージをインストールします。

   (https://www.citrix.com/downloads/citrix-virtual-apps-and-desktops/) にある にアクセスします。Citrix Virtual Apps and Desktops の適切なバージョンを展開し、 をクリックして、お使いの Linux ディストリビューションに一致する Linux VDA パッケージをダウンロードします。
3. ランタイム環境をセットアップして、Linux VDA のインストールを完了します。

手順 1：構成情報と Linux マシンを準備する

簡単インストールに必要な次の構成情報を収集します。

- ホスト名 - Linux VDA をインストールするマシンのホスト名
- ドメインネームサーバーの IP アドレス
- NTP サーバーの IP アドレスまたは文字列名
- ドメイン名 - ドメインの NetBIOS 名
- レルム名 - Kerberos レルム名
- Active Directory の FQDN - 完全修飾ドメイン名

重要:

- Linux VDAをインストールするには、リポジトリがLinuxマシンに正しく追加されていることを確認してください。

> - セッションを起動するには、X Windowシステムとデスクトップ環境がインストールされていることを確認してください。

考慮事項

- ワークグループ名は、デフォルトではドメイン名です。環境内でワークグループをカスタマイズするには、次の手順を実行します。

  a. Linux VDA マシン上に /tmp/ctxinstall.conf ファイルを作成します。

  b. ファイルに workgroup=\<your workgroup\> の行を追加し、変更を保存します。

- Centrifyは純粋なIPv6 DNS構成をサポートしていません。`adclient`がADサービスを適切に検出するには、/etc/resolv.confにIPv4を使用するDNSサーバーが少なくとも1つ必要です。

  ログ：

  ```
  ADSITE   : Check that this machine's subnet is in a site known by AD
    : Failed

           : This machine's subnet is not known by AD.

           : We guess you should be in the site Site1.
  ```

  この問題はCentrifyとその構成に固有のものです。この問題を解決するには、次の手順を実行します。

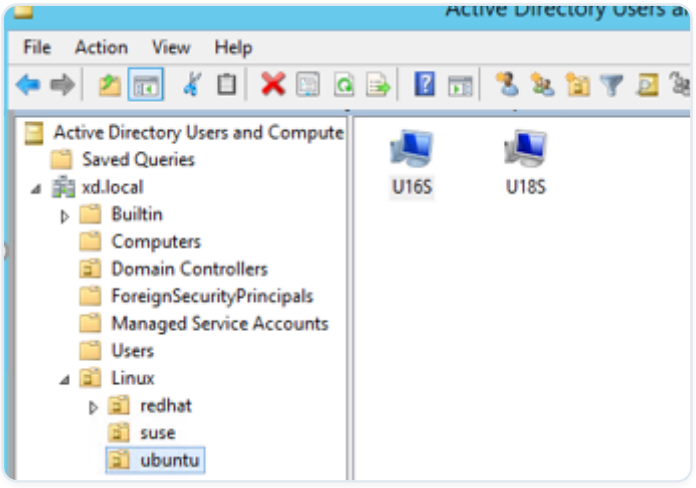
  a. ドメインコントローラーで**管理ツール**を開きます。

  b. **Active Directoryサイトとサービス**を選択します。 c. **サブネット**に適切なサブネットアドレスを追加します。

- VDAを特定のOUに参加させるには、次の手順を実行します。

  a. 特定のOUがドメインコントローラーに存在することを確認してください。

  OUの例については、次のスクリーンショットを参照してください。



b. VDA上に/tmp/ctxinstall.confファイルを作成します。

c. /tmp/ctxinstall.confファイルにou=\<your ou\>の行を追加します。

OUの値はADの方法によって異なります。次の表を参照してください。

OS	ウィンバインド	SSSD	セントリファイ	PBIS
RHEL 8	ou="OU=redhat,OU=Linux"	ou="OU=redhat,OU=Linux"	ou="XD.LOCAL/Linux/redhat"	ou="Linux/redhat"
RHEL 7	ou="Linux/redhat"	ou="Linux/redhat"	ou="XD.LOCAL/Linux/redhat"	ou="Linux/redhat"
ウブントゥ	ou="Linux/ubuntu"	ou="Linux/ubuntu"	ou="XD.LOCAL/Linux/ubuntu"	ou="Linux/ubuntu"
SUSE 12.5	ou="Linux/suse"	ou="Linux/suse"	ou="XD.LOCAL/Linux/suse"	ou="Linux/suse"
デビアン	ou="Linux/debian"	ou="Linux/debian"	ou="XD.LOCAL/Linux/debian"	ou="Linux/debian"

- イージーインストールは、Linux VDA 7.16以降、純粋なIPv6をサポートします。以下の前提条件と制限が適用されます。
  - マシンが純粋なIPv6ネットワーク経由で必要なパッケージをダウンロードできるように、Linuxリポジトリを構成する必要があります。

- Centrify は純粋な IPv6 ネットワークではサポートされていません。

> **注:**
>
> ネットワークが純粋な IPv6 であり、すべての入力が適切な IPv6 形式である場合、VDA は IPv6 を介して Delivery Controller™ に登録されます。ネットワークが IPv4 と IPv6 のハイブリッド構成である場合、最初の DNS IP アドレスのタイプによって、登録に IPv4 または IPv6 のどちらが使用されるかが決まります。

- ドメインに参加する方法として Centrify を選択した場合、ctxinstall.sh スクリプトには Centrify パッケージが必要です。ctxinstall.sh が Centrify パッケージを取得する方法は 2 つあります。
  - Easy install は、Centrify パッケージをインターネットから自動的にダウンロードするのに役立ちます。各ディストリビューションの URL は次のとおりです。

    RHELの場合： wget `<http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-rhel4-x86_64.tgz?_ga=1.178323680.558673738.1478847956>`

    セントOS： wget `<http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-rhel4-x86_64.tgz?_ga=1.186648044.558673738.1478847956>`

    スーゼ： wget `<http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-suse10-x86_64.tgz?_ga=1.10831088.558673738.1478847956>`

    Ubuntu/Debian：wget `<https://downloads.centrify.com/products/infrastructure-services/19.9/centrify-infrastructure-services-19.9-deb8-x86_64.tgz?_ga=2.151462329.1042350071.1592881996-604509155.1572850145>`
  - ローカルディレクトリから Centrify パッケージを取得します。Centrify パッケージのディレクトリを指定するには、次の手順を実行します。

    a. Linux VDA サーバーに /tmp/ctxinstall.conf ファイルが存在しない場合は作成します。

    b. ファイルに「centrifypkgpath=\<path name\>」という行を追加します。

    例:

```
cat /tmp/ctxinstall.conf

set “centrifypkgpath=/home/mydir”

ls  -ls  /home/mydir

    9548 -r-xr-xr-x. 1 root root  9776688 May 13  2016 adcheck-
rhel4-x86_64

    4140 -r--r--r--. 1 root root  4236714 Apr 21  2016
centrifyda-3.3.1-rhel4-x86_64.rpm

    33492 -r--r--r--. 1 root root 34292673 May 13  2016
centrifydc-5.3.1-rhel4-x86_64.rpm

    4 -rw-rw-r--. 1 root root     1168 Dec  1  2015 centrifydc-
install.cfg

    756 -r--r--r--. 1 root root   770991 May 13  2016 centrifydc-
ldapproxy-5.3.1-rhel4-x86_64.rpm

    268 -r--r--r--. 1 root root   271296 May 13  2016 centrifydc-
nis-5.3.1-rhel4-x86_64.rpm

    1888 -r--r--r--. 1 root root  1930084 Apr 12  2016 centrifydc-
openssh-7.2p2-5.3.1-rhel4-x86_64.rpm

    124 -rw-rw-r--. 1 root root   124543 Apr 19  2016 centrify-
suite.cfg

    0 lrwxrwxrwx. 1 root root       10 Jul  9  2012 install-
express.sh -> install.sh

    332 -r-xr-xr--. 1 root root   338292 Apr 10  2016 install.sh

    12 -r--r--r--. 1 root root    11166 Apr  9  2015 release-notes-
agent-rhel4-x86_64.txt

    4 -r--r--r--. 1 root root     3732 Aug 24  2015 release-notes-
da-rhel4-x86_64.txt

    4 -r--r--r--. 1 root root     2749 Apr  7  2015 release-notes-
nis-rhel4-x86_64.txt
```

```
    12 -r--r--r--. 1 root root     9133 Mar 21  2016 release-notes-
openssh-rhel4-x86_64.txt
```

- ドメインに参加する方法として PBIS を選択した場合、ctxinstall.sh スクリプトには PBIS パッケージが必要です。ctxinstall.sh が PBIS パッケージを取得する方法は 2 つあります。
  - Easy install は、PBIS パッケージをインターネットから自動的にダウンロードするのに役立ちます。各ディストリビューションの URL は次のとおりです。

    RHEL 7 / CentOS 7 / SUSE 12.5: `wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh`

    RHEL 8 / CentOS 8: `wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh`

    Ubuntu/Debian: `wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh`
  - インターネットからPBISパッケージの特定のバージョンを取得します。そのためには、/opt/Citrix/VDA/sbin/ctxinstall.shファイル内の「pbisDownloadPath」行をPBISパッケージのURLを指定するように変更します。

    例については、次のスクリーンショットを参照してください。

    PBISパッケージの特定のバージョンを取得する(/en-us/linux-virtual-delivery-agent/2106/media/fetching-specific-version-of-pbis-package.png)

ステップ2: (Ubuntu 16.04のみ) OpenJDK 11をインストールする

Ubuntu 16.04で、次の手順を実行してOpenJDK 11をインストールします。

1. `<https://jdk.java.net/archive/>`から最新のOpenJDK 11をダウンロードします。
2. ダウンロードしたパッケージを解凍するには、`tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz`コマンドを実行します。
3. (オプション) OpenJDKをターゲットディレクトリに保存するには、`mv jdk-11.0.2/ <target directory>`コマンドを実行します。
4. Javaランタイムをセットアップするには、`update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000`コマンドを実行します。
5. Javaのバージョンを確認するには、`java -version`コマンドを実行します。

ステップ3: Linux VDAパッケージをインストールする

Linux VDAの環境をセットアップするには、次のコマンドを実行します。

RHELおよびCentOSディストリビューションの場合:

```
sudo yum  -y localinstall   <PATH>/<Linux VDA RPM>
```

Ubuntu/Debianディストリビューションの場合:

```
sudo dpkg -i <PATH>/<Linux VDA deb>

sudo apt-get install -f
```

注:

Debianディストリビューションに必要な依存関係をインストールするには、/etc/apt/sources.listファイルに`deb http://deb.debian.org/debian/ oldstable main`行を追加します。

SUSEディストリビューションの場合:

```
zypper -i install <PATH>/<Linux VDA RPM>
```

ステップ4：インストールを完了するためにランタイム環境をセットアップする

注:

ランタイム環境をセットアップする前に、`en_US.UTF-8`ロケールがOSにインストールされていることを確認してください。OSでロケールが利用できない場合は、`sudo locale-gen en_US.UTF-8`コマンドを実行してください。Debianの場合、`/etc/locale.gen`ファイルを編集し、`# en_US.UTF-8 UTF-8`行のコメントを解除してから、`sudo locale-gen`コマンドを実行してください。

> Linux VDAパッケージをインストールした後、ctxinstall.shスクリプトを使用して実行環境を構成します。このスクリプトは、対話モードまたはサイレントモードで実行できます。

> **注:**

> Easy Installは、27 MBを超える.NET Core Runtimeをダウンロードしている間、応答しないように見えることがあります。ダウンロードの進行状況については、/var/log/ctxinstall.logを確認してください。

> **対話モード:**

手動で構成するには、次のコマンドを実行し、各プロンプトで関連するパラメーターを入力します。

```
sudo /opt/Citrix/VDA/sbin/ctxinstall.sh
```

サイレントモード:

サイレントモードでEasy Installを使用するには、ctxinstall.shを実行する前に次の環境変数を設定します。

- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDAサーバーのホスト名を示します。
- **CTX_EASYINSTALL_DNS=ip-address-of-dns** – DNSのIPアドレス。
- **CTX_EASYINSTALL_NTPS=address-of-ntps** – NTPサーバーのIPアドレスまたは文字列名。
- **CTX_EASYINSTALL_DOMAIN=domain-name** – ドメインのNetBIOS名。
- **CTX_EASYINSTALL_REALM=realm-name** – Kerberosレルム名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis** – Active Directory 統合方法を示します。
- **CTX_EASYINSTALL_USERNAME=domain-user-name** – ドメインユーザーの名前を示します。ドメインへの参加に使用されます。
- **CTX_EASYINSTALL_PASSWORD=password** – ドメインユーザーのパスワードを指定します。ドメインへの参加に使用されます。

ctxsetup.sh スクリプトは次の変数を使用します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller への登録に使用する Delivery Controller の完全修飾ドメイン名 (FQDN) のスペース区切りリストを必要とします。少なくとも1つの FQDN または CNAME を指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktop サービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktop サービスでは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux Virtual Desktop のシステムファイアウォールで、必要なポート（デフォルトではポート 80 および 1494）を自動的に開くことができます。
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル (VDI) として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境の場合は、値を Y に設定します。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、**\<none\>** に設定します。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例: ad1.mycompany.com:389。不要な場合は、**\<none\>** に設定します。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例: DC=mycompany,DC=com）を介して LDAP を照会します。検索パフォーマンスを向上させるには、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**\<none\>** に設定します。

- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロン区切りの FAS サーバーリストを指定できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白をテキスト文字列 **\<none\>** で埋め、サーバーアドレスの順序を変更しないでください。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために.NET Core Runtime 3.1をインストールするパス。デフォルトのパスは/usr/binです。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/mate** – セッションで使用するGNOMEまたはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、変数の値を**mate**に設定する必要があります。

> **注:**
>
> ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。
>
> a. VDA上の**$HOME/<username\>**ディレクトリの下に `.xsession` ファイルを作成します。
>
> b. ディストリビューションに基づいてデスクトップ環境を指定するように、 `.xsession` ファイルを編集します。
>
> **CentOS、Ubuntu、Debian上のMATEデスクトップの場合**
>
> MSESSION="$(type -p mate-session)"
>
> \> もし ; その場合
>
> ```
> 実行 mate-session
> ```
>
> \> fi

```
**CentOS上のGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> もし ; その場合

```
   エクスポート GNOME_SHELL_SESSION_MODE=classic
```

\> 実行 gnome-session --session=gnome-classic

```
   fi

**UbuntuおよびDebian上のGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> の条件が満たされる場合、

```
   exec gnome-session
```

\> fi

a. ターゲットセッションユーザーと700ファイル権限を共有します。

- **CTX_XDL_START_SERVICE=Y \| N** – 構成完了時にLinux VDAサービスを開始するかどうか。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutのリッスン用ソケットポート。デフォルトポートは7503です。

- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutとの通信用ポート。デフォルトポートは7502です。

いずれかのパラメーターが設定されていない場合、インストールは対話モードに戻り、ユーザー入力が求められます。すべてのパラメーターが環境変数を通じてすでに設定されている場合でも、ctxinstall.shスクリプトは.NET Core Runtime 3.1をインストールするパスについてユーザー入力を求めます。

サイレントモードでは、環境変数を設定するために以下のコマンドを実行し、その後ctxinstall.shスクリプトを実行する必要があります。

```
export CTX_EASYINSTALL_HOSTNAME=host-name


export CTX_EASYINSTALL_DNS=ip-address-of-dns


export CTX_EASYINSTALL_NTPS=address-of-ntps


export CTX_EASYINSTALL_DOMAIN=domain-name


export CTX_EASYINSTALL_REALM=realm-name


export CTX_EASYINSTALL_FQDN=ad-fqdn-name


export CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis


export CTX_EASYINSTALL_USERNAME=domain-user-name


```

```
export CTX_EASYINSTALL_PASSWORD=password

export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N

export CTX_XDL_DDC_LIST=‘list-ddc-fqdns’

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y | N

export CTX_XDL_ADD_FIREWALL_RULES=Y | N

export CTX_XDL_HDX_3D_PRO=Y | N

export CTX_XDL_VDI_MODE=Y | N

export CTX_XDL_SITE_NAME=dns-site-name | '<none>'

export CTX_XDL_LDAP_LIST=‘list-ldap-servers’ | '<none>'

export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'
```

```
export CTX_XDL_FAS_LIST=‘list-fas-servers’ | '<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y | N

sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh
```

sudoコマンドを実行する際は、既存の環境変数を新しく作成されるシェルに渡すために-Eオプションを入力してください。前述のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST=‘list-ddc-fqdns’ \

CTX_XDL_VDA_PORT=port-number \
```

```
CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST=‘list-ldap-servers’ \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST=‘list-fas-servers’ \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \
```

```
CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
```

トラブルシューティング

このセクションの情報を使用して、この機能の使用から生じる可能性のある問題をトラブルシューティングしてください。

SSSDを使用したドメインへの参加が失敗する

ドメインに参加しようとするとエラーが発生し、出力は以下のようになる場合があります（画面表示についてはログを確認してください）：

`Step 6: join Domain!Enter ctxadmin's password:Failed to join domain: failed to lookup DC info for domain 'CITRIXLAB.LOCAL' over rpc: The network name cannot be found`

/var/log/xdl/vda.log:

```
2016-11-04 02:11:52.317 [INFO ] - The Citrix Desktop Service successfully
obtained the following list of 1 delivery controller(s) with which to
register: 'CTXDDC.citrixlab.local (10.158.139.214)'.

2016-11-04 02:11:52.362 [ERROR] -
RegistrationManager.AttemptRegistrationWithSingleDdc: Failed to register
with http://CTXDDC.citrixlab.local:80/Citrix/CdsController/IRegistrar.
Error: General security error (An error occurred in trying to obtain a
TGT: Client not found in Kerberos database (6))

2016-11-04 02:11:52.362 [ERROR] - The Citrix Desktop Service cannot
```

```
connect to the delivery controller 'http://CTXDDC.citrixlab.local:80/
Citrix/CdsController/IRegistrar' (IP Address '10.158.139.214')

Check the following:- The system clock is in sync between this machine and
the delivery controller.

-  The Active Directory provider (e.g. winbind daemon) service is running
and correctly configured.

-  Kerberos is correctly configured on this machine.

If the problem persists, please refer to Citrix Knowledge Base article
CTX117248 for further information.

Error Details:

Exception 'General security error (An error occurred in trying to obtain a
TGT: Client not found in Kerberos database (6))' of type 'class
javax.xml.ws.soap.SOAPFaultException'.

2016-11-04 02:11:52.362 [INFO ] -
RegistrationManager.AttemptRegistrationWithSingleDdc: The current time for
this VDA is Fri Nov 04 02:11:52 EDT 2016.

Ensure that the system clock is in sync between this machine and the
delivery controller.

Verify the NTP daemon is running on this machine and is correctly
configured.

2016-11-04 02:11:52.364 [ERROR] - Could not register with any controllers.
Waiting to try again in 120000 ms. Multi-forest - false

2016-11-04 02:11:52.365 [INFO ] - The Citrix Desktop Service failed to
register with any controllers in the last 470 minutes.
```

/var/log/messages:

`Nov 4 02:15:27 RH-WS-68 [sssd[ldap_child[14867]]]: Failed to initialize credentials using keytab [MEMORY:/etc/krb5.keytab]: Client 'RH-WS-68$@CITRIXLAB.LOCAL' not found in Kerberos database. Unable to create GSSAPI-`

`encrypted LDAP connection.Nov 4 02:15:27 RH-WS-68 [sssd[ldap_child[14867]]]: Client 'RH-WS-68$@CITRIXLAB.LOCAL' not found in Kerberos database`

この問題を解決するには：

1. `rm -f /etc/krb5.keytab`コマンドを実行します。
2. `net ads leave $REALM -U $domain-administrator`コマンドを実行します。
3. Delivery Controller上のマシンカタログとデリバリーグループを削除します。
4. /opt/Citrix/VDA/sbin/ctxinstall.shを実行します。
5. Delivery Controller上にマシンカタログとデリバリーグループを作成します。

Ubuntuデスクトップセッションでグレースクリーンが表示される

この問題は、セッションを起動したときに、そのセッションが空白のデスクトップでブロックされる場合に発生します。さらに、ローカルユーザーアカウントを使用してログオンすると、マシンのコンソールにもグレースクリーンが表示されます。

この問題を解決するには：

1. `sudo apt-get update`コマンドを実行します。
2. `sudo apt-get install unity lightdm`コマンドを実行します。
3. `/etc/lightdm/lightdm.conf`に次の行を追加します。

   `greeter-show-manual-login=true`

ホームディレクトリがないため、Ubuntuデスクトップセッションの起動に失敗する

/var/log/xdl/hdx.log:

```
2016-11-02 13:21:19.015 <P22492:S1> citrix-ctxlogin: StartUserSession:
failed to change to directory(/home/CITRIXLAB/ctxadmin) errno(2)



2016-11-02 13:21:19.017 <P22227> citrix-ctxhdx: logSessionEvent: Session
started for user ctxadmin.



2016-11-02 13:21:19.023 <P22492:S1> citrix-ctxlogin: ChildPipeCallback:
Login Process died: normal.
```

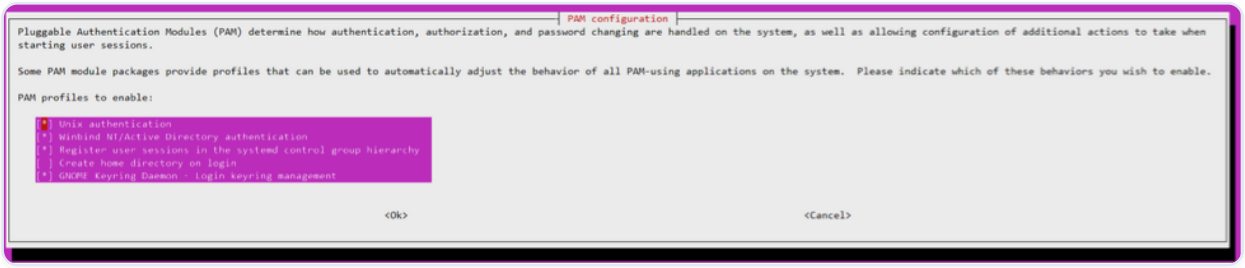
```
2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting normally.
```

ヒント:

この問題の根本原因は、ドメイン管理者用のホームディレクトリが作成されていないことです。

この問題を解決するには：

1. コマンドラインから「**pam-auth-update**」と入力します。
2. 表示されるダイアログで、**Create home directory login**が選択されていることを確認します。



dbusエラーによりセッションが起動しない、またはすぐに終了する

/var/log/messages（RHELまたはCentOSの場合）：

```
Oct 27 04:17:16 CentOS7 citrix-ctxhdx[8978]: Session started for user
CITRIXLAB\ctxadmin.



Oct 27 04:17:18 CentOS7 kernel: traps: gnome-session[19146] trap int3
ip:7f89b3bde8d3 sp:7fff8c3409d0 error:0



Oct 27 04:17:18 CentOS7 gnome-session[19146]: ERROR: Failed to connect to
system bus: Exhausted all available authentication mechanisms (tried:
EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS) (available: EXTERNAL,
DBUS_COOKIE_SHA1, ANONYMOUS)#012aborting...
```

```
Oct 27 04:17:18 CentOS7 gnome-session: gnome-session[19146]: ERROR: Failed
to connect to system bus: Exhausted all available authentication
mechanisms (tried: EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS) (available:
EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS)



Oct 27 04:17:18 CentOS7 gnome-session: aborting...



Oct 27 04:17:18 CentOS7 citrix-ctxgfx[18981]: Exiting normally.



Oct 27 04:17:18 CentOS7 citrix-ctxhdx[8978]: Session stopped for user
CITRIXLAB\ctxadmin.
```

または、Ubuntuディストリビューションの場合は、ログ/var/log/syslogを使用します。

```
Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] pid.c:
Stale PID file, overwriting.



Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] bluez5-
util.c: Failed to get D-Bus connection: Did not receive a reply. Possible
causes include: the remote application did not send a reply, the message
bus security policy blocked the reply, the reply timeout expired, or the
network connection was broken.



Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] hashmap.c:
Assertion 'h' failed at pulsecore/hashmap.c:116, function
pa_hashmap_free(). Aborting.



Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: [pulseaudio] core-
util.c: Failed to connect to system bus: Did not receive a reply. Possible
```

```
causes include: the remote application did not send a reply, the message
bus security policy blocked the reply, the reply timeout expired, or the
network connection was broken.



Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: message repeated 10
times: [ [pulseaudio] core-util.c: Failed to connect to system bus: Did
not receive a reply. Possible causes include: the remote application did
not send a reply, the message bus security policy blocked the reply, the
reply timeout expired, or the network connection was broken.]



Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: [pulseaudio] pid.c:
Daemon already running.Nov  3 11:03:58 user01-HVM-domU citrix-
ctxgfx[24693]: Exiting normally
```

一部のグループまたはモジュールは、再起動するまで有効になりません。ログに**dbus**エラーメッセージが表示される場合は、システムを再起動して再試行することをお勧めします。

SELinuxによりSSHDがホームディレクトリにアクセスできない

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/ctxinstall.log:

```
Jan 25 23:30:31 yz-rhel72-1 setroubleshoot[3945]: SELinux is preventing /
usr/sbin/sshd from setattr access on the directory /root. For complete
SELinux messages. run sealert -l 32f52c1f-8ff9-4566-a698-963a79f16b81



Jan 25 23:30:31 yz-rhel72-1 python[3945]: SELinux is preventing /usr/sbin/
sshd from setattr access on the directory /root.



*****  Plugin catchall_boolean (89.3 confidence) suggests
******************
```

```
If you want to allow polyinstantiation to enabled



   Then you must tell SELinux about this by enabling the
'polyinstantiation_enabled' boolean.



You can read 'None' man page for more details.



    Do



       setsebool -P polyinstantiation_enabled 1



*****  Plugin catchall (11.6 confidence) suggests
*************************



If you believe that sshd should be allowed setattr access on the root
directory by default.



Then you should report this as a bug.



You can generate a local policy module to allow this access.



      Do
```

```
        allow this access for now by executing:


        # grep sshd /var/log/audit/audit.log | audit2allow -M mypol


# semodule -i mypol.pp
```

この問題を解決するには：

1. /etc/selinux/configに次の変更を加えてSELinuxを無効にします。

   SELINUX=無効

2. VDA を再起動します。

Citrix Virtual Apps and Desktops™ Standard for Azure 環境での Linux VDA の作成方法

Microsoft Azure から任意のデバイスに仮想アプリとデスクトップを配信するために、Citrix Virtual Apps and Desktops Standard for Azure でドメイン参加済みおよび非ドメイン参加済みの両方の Linux VDA を作成できます。詳細については、Citrix Virtual Apps and Desktops Standard for Azure を参照してください。

サポートされているLinuxディストリビューション

次の Linux ディストリビューションがこの機能をサポートしています。

- RHEL 8.3
- RHEL 8.2
- RHEL 7.8
- ウブントゥ 20.04
- ウブントゥ 18.04
- ウブントゥ 16.04

手順

Citrix Virtual Apps™ and Desktops Standard for Azure で Linux VDA を作成するには、次の手順を完了します。

1. Azure でマスターイメージを準備します。

> **注:**
>
> 自動ソフトウェア更新をスケジュールするには、Linux VDA self-update 機能を使用することもできます。この目的を達成するには、マスターイメージ上の etc/xdl/mcs/mcs_local_setting.reg ファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。
>
> ```
> create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
> -t "REG_DWORD" -v "fEnabled" -d "0x00000001" –force
>
>
>
> create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
> -t "REG_SZ" -v "ScheduledTime" -d "Immediately" –force
>
>
>
> create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
> -t "REG_SZ" -v "Url" -d "<Your-Azure-Container-Url>" –force
>
>
>
> create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
> -t "REG_SZ" -v "CaCertificate" -d "<Local-Certificate-Path-of-
> PortalAzureCom>" --force
> ```

a. Azure で、サポートされているディストリビューションの Linux VM を作成します。

b. 必要に応じて、Linux VM にデスクトップ環境をインストールします。

c. VM に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` の指示に従って .NET Core Runtime 3.1 をインストールします。

d. (Ubuntuのみ) `/etc/network/interfaces` ファイルに `source /etc/network/interfaces.d/*` 行を追加します。

e. (Ubuntuのみ) `/run/systemd/resolve/stub-resolv.conf`ではなく、`/etc/resolv.conf`を`/run/systemd/resolve/resolv.conf`にポイントします。

```
unlink /etc/resolv.conf



ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

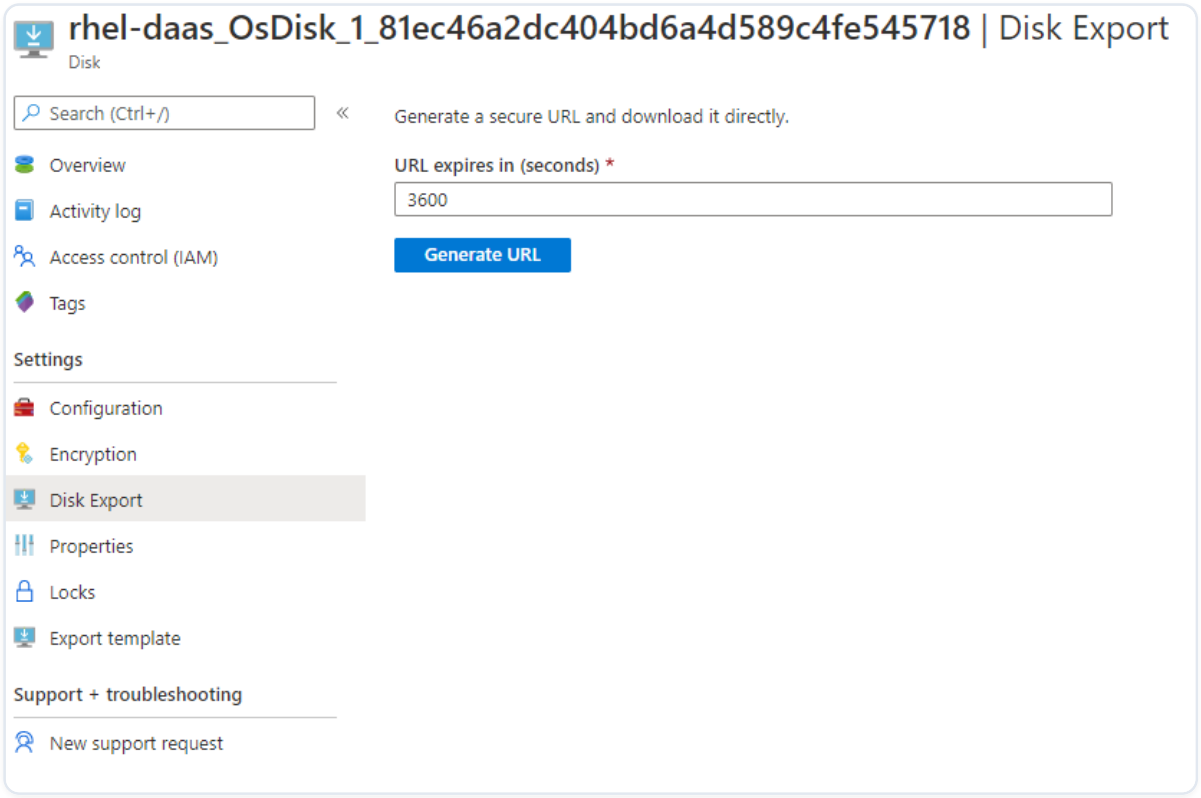
f. Linux VDAパッケージをインストールします。

g. /etc/xdl/mcs/mcs.confの変数を変更します。`mcs.conf`構成ファイルには、MCSとLinux VDAを設定するための変数が含まれています。

> **注:**
>
> dns変数は指定しないままにします。 マシンカタログを作成するときに**Static**または**Random**タイプを選択した場合は、`VDI_MODE=Y`を設定します。

h. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。

i. Azureで、VMを停止（または割り当て解除）します。**ディスクのエクスポート**をクリックして、他のVMを作成するためのマスターイメージとして使用できる仮想ハードディスク（VHD）ファイルのSAS URLを生成します。



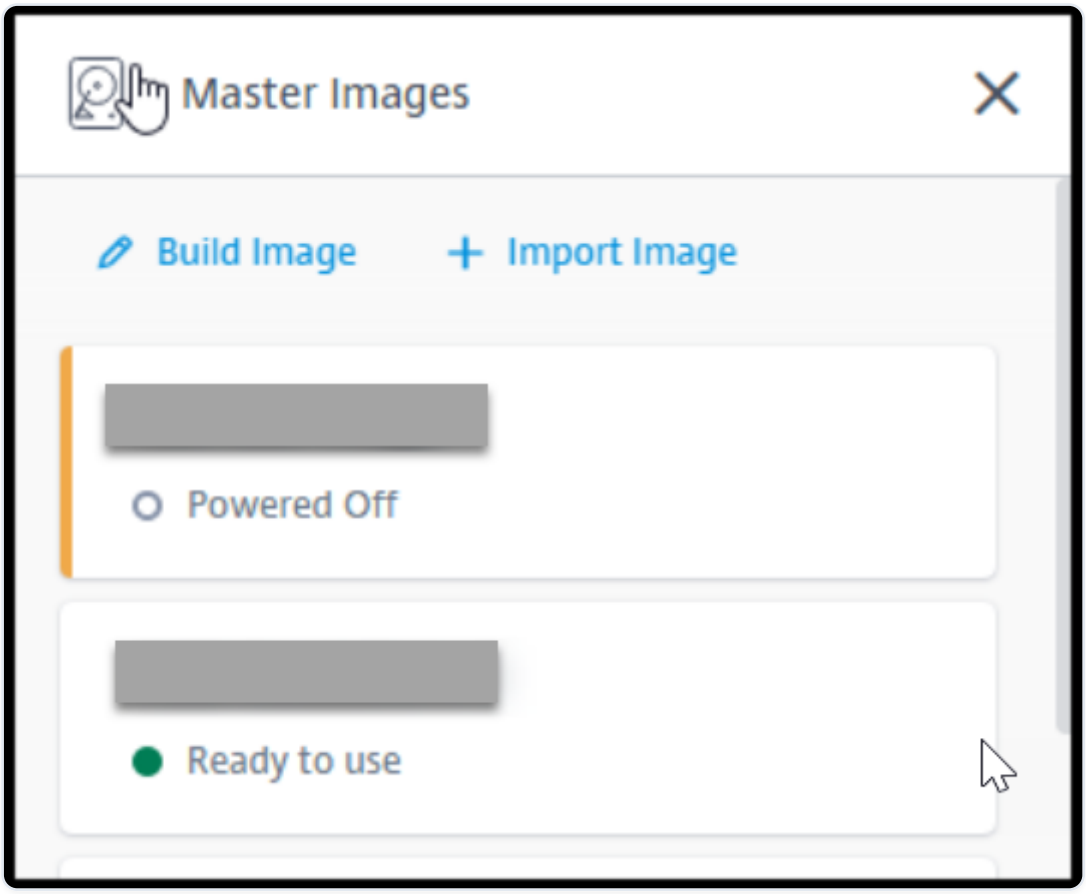
j. (オプション) マスターイメージでグループポリシー設定を行います。`ctxreg`ツールを使用してグループポリシー設定を行うことができます。たとえば、次のコマンドはPDF印刷の**PDFユニバーサルプリンターの自動作成**ポリシーを有効にします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\GroupPolicy\Defaults\PrintingPolicies" -t
"REG_DWORD" -v "AutoCreatePDFPrinter" -d "0x00000001" –force
```

2. Azureからマスターイメージをインポートします。

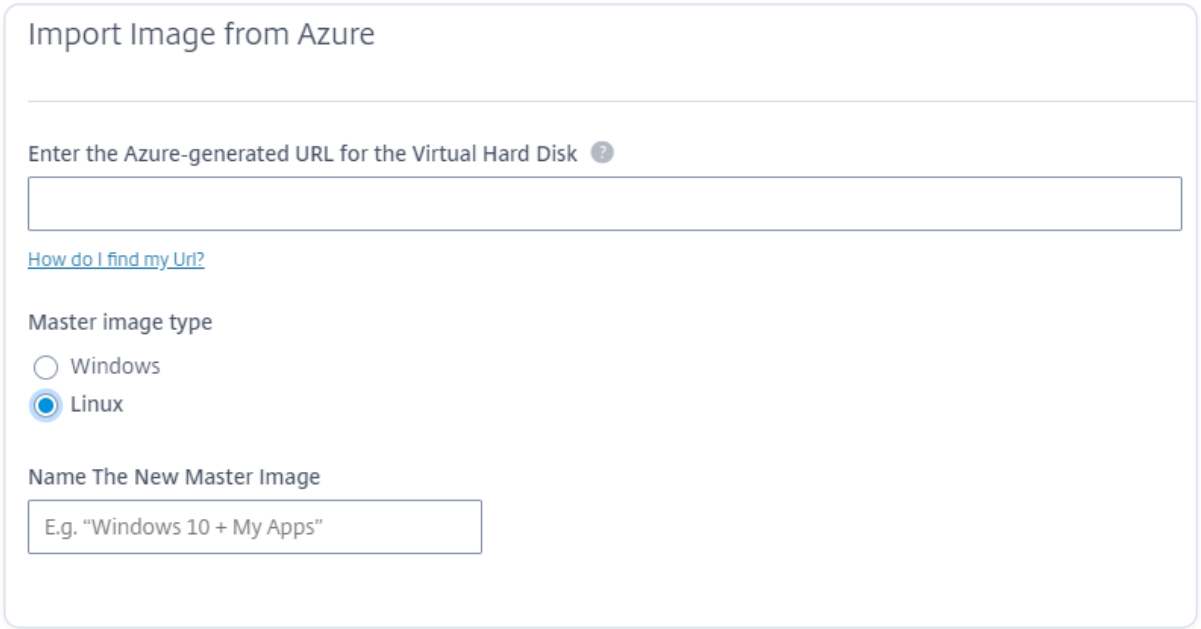
a. 管理ダッシュボードから、右側の**マスターイメージ**を展開します。Citrixが提供するマスターイメージと、作成およびインポートしたイメージが表示されます。

ヒント: このサービスのほとんどの管理者アクティビティは、**管理**および**監視**ダッシュボードを通じて管理されます。最初のカタログを作成すると、Citrix Cloud™にサインインして**管理対象デスクトップ**サービスを選択した後、**管理**ダッシュボードが自動的に起動します。



b. **イメージのインポート**をクリックします。

c. Azureで生成したVHDファイルのSAS URLを入力します。マスターイメージの種類として**Linux**を選択します。



d. ウィザードの指示に従って、マスターイメージのインポートを完了します。

3. マシンカタログを作成します。

Manageダッシュボードにアクセスし、**カタログの作成**をクリックします。マシンカタログを作成する際は、以前に作成したマスターイメージを選択してください。

注： ドメインに参加していないLinuxマシンカタログは、Citrix管理のAzureサブスクリプションでのみ作成できます。

Machine Creation Services™ (MCS) を使用してLinux VMを作成する

MCSを使用してLinux VMを作成するには、ハイパーバイザー上にマスターイメージを準備します。このプロセスには、テンプレートVMへのVDAのインストール、Citrix Studioでのマシンカタログの作成、デリバリーグループの作成、および特定の構成タスクの実行が含まれます。

注:

Citrix Hypervisor™、Microsoft Azure、VMware vSphere、AWS、GCP、またはNutanix AHV以外のハイパーバイザーでマスターイメージを準備しようとすると、予期しない結果が発生する可能性があります。

Microsoft Azure、AWS、およびGCPは、Citrix Virtual Apps and Desktops™ 7 2003以降、サポートされていません。ただし、Citrix Virtual Apps and Desktopsサービスでホストを引き続き使用できます。

サポートされているディストリビューション

	ウィンバインド	SSSD	セントリファイ	PBIS
RHEL 8.3	はい	いいえ	いいえ	はい
CentOS 8.3	はい	いいえ	いいえ	はい
RHEL 8.2	はい	いいえ	いいえ	はい
CentOS 8.2	はい	いいえ	いいえ	はい
RHEL 8.1	はい	いいえ	いいえ	はい
CentOS 8.1	はい	いいえ	いいえ	はい

	ウィンバインド	SSSD	セントリファイ	PBIS
RHEL 7.9	はい	はい	いいえ	はい
CentOS 7.9	はい	はい	いいえ	はい
RHEL 7.8	はい	はい	いいえ	はい
CentOS 7.8	はい	はい	いいえ	はい
ウブントゥ 20.04	はい	はい	いいえ	はい
ウブントゥ 18.04	はい	はい	いいえ	はい
ウブントゥ 16.04	はい	はい	いいえ	はい
デビアン 10.7	はい	はい	いいえ	はい
SUSE 12.5	はい	はい	いいえ	はい

注:

MCSで作成されたマシンをWindowsドメインに参加させるためにPBISを使用している場合は、次のタスクを完了してください。

- テンプレートマシンで、`/etc/xdl/mcs/mcs.conf` ファイル内のPBISパッケージのダウンロードパスを設定するか、PBISパッケージを直接インストールします。
- `/opt/Citrix/VDA/sbin/deploymcs.sh` を実行する前に、その配下にあるMCSで作成されたすべてのマシンに対して書き込み権限とパスワードリセット権限を持つ組織単位 (OU) を作成します。
- `/opt/Citrix/VDA/sbin/deploymcs.sh` の実行が完了した後、MCSで作成されたマシンを再起動する前に、展開に応じてDelivery ControllerまたはCitrix Cloud Connectorで `klist -li 0x3e4 purge` を実行します。

MCSを使用してCitrix Hypervisor上にLinux VMを作成する

ステップ1：マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ1a：Citrix VM Toolsをインストールする

各VMがxe CLIまたはXenCenterを使用できるように、テンプレートVMにCitrix VM Toolsをインストールします。ツールをインストールしないと、VMのパフォーマンスが低下する可能性があります。ツールがないと、次のいずれも実行できません。

- VMを正常にシャットダウン、再起動、または一時停止する。
- XenCenterでVMのパフォーマンスデータを表示する。
- 実行中のVMを移行する（`XenMotion`経由）。
- スナップショットまたはメモリ付きスナップショット（チェックポイント）を作成し、スナップショットに戻す。
- 実行中のLinux VMのvCPU数を調整する。

1. guest-tools.isoという名前のCitrix VM Toolsをマウントするには、次のコマンドを実行します。

   ```
   sudo mount /dev/cdrom /mnt
   ```

2. Linuxディストリビューションに基づいて`xe-guest-utilities`パッケージをインストールするには、次のコマンドを実行します。

   RHEL/CentOSの場合：

   ```
   sudo rpm -i /mnt/Linux/xe-guest-utilities_{package-version}_all.rpm
   ```

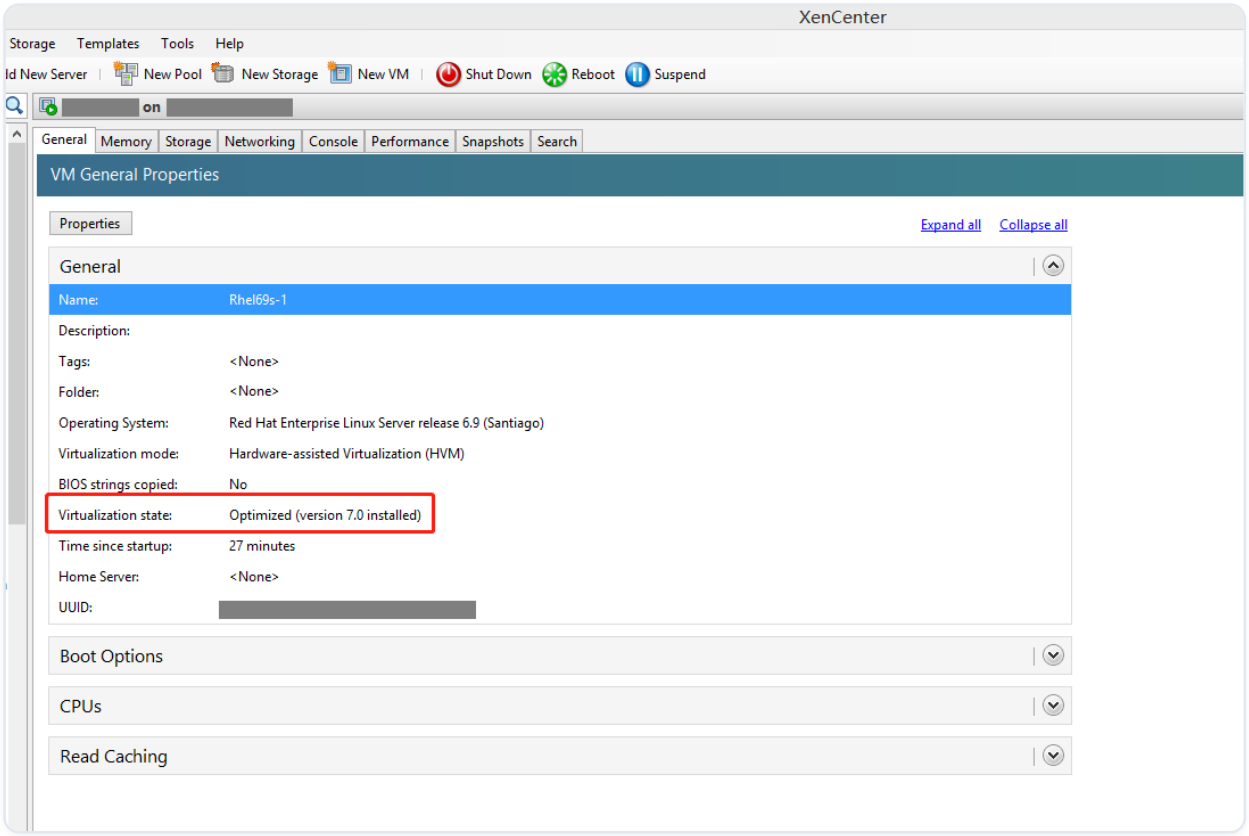
   Ubuntu/Debianの場合：

   ```
   sudo dpkg -i /mnt/Linux/xe-guest-utilities_{package-version}_all.deb
   ```

   SUSE 12の場合：

   ```
   sudo rpm -i /mnt/Linux/xe-guest-utilities_{package-version}_all.rpm
   ```

3. XenCenterの**［全般］タブで、テンプレートVMの仮想化状態を確認します。Citrix VM Tools が正しくインストールされている場合、仮想化状態は［最適化済み］**になります。



ステップ 1b: (Ubuntu 16.04 のみ) OpenJDK 11 をインストールする

Ubuntu 16.04 で、以下の手順を実行して OpenJDK 11 をインストールします。

1. `<https://jdk.java.net/archive/>` から最新の OpenJDK 11 をダウンロードします。
2. ダウンロードしたパッケージを解凍するには、`tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` コマンドを実行します。
3. (オプション) OpenJDK をターゲットディレクトリに保存するには、`mv jdk-11.0.2/ <target directory>` コマンドを実行します。
4. Java ランタイムをセットアップするには、`update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` コマンドを実行します。
5. Java のバージョンを確認するには、`java -version` コマンドを実行します。

ステップ 1c: テンプレート VM に Linux VDA パッケージをインストールする

> **注:**

> 現在実行中の VDA をテンプレート VM として使用する場合は、この手順をスキップしてください。

> テンプレート VM に Linux VDA パッケージをインストールする前に、.NET Core Runtime 3.1 をインストールしてください。詳細については、「インストール概要」を参照してください。

> お使いの Linux ディストリビューションに基づいて、Linux VDA の環境をセットアップするために次のコマンドを実行します。

RHEL/CentOS の場合:

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debian の場合:

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12 の場合:

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ 1d: tdb-tools パッケージをインストールするためにリポジトリを有効にする

RHEL 7サーバーの場合：

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
```

RHEL 7ワークステーションの場合：

```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
```

ステップ1e: ntfs-3gを提供するEPELリポジトリをインストールする

RHEL 8/CentOS 8、RHEL 7/CentOS 7にEPELリポジトリをインストールすると、後でdeploymcs.shを実行したときにEPELリポジトリで利用可能なntfs-3gパッケージがインストールされます。

ステップ1f: SUSE 12にntfs-3gを手動でインストールする

SUSE 12プラットフォームでは、ntfs-3gを提供するリポジトリはありません。ソースコードをダウンロードし、コンパイルして、ntfs-3gを手動でインストールします。

1. GNU Compiler Collection (GCC) コンパイラシステムとmakeパッケージをインストールします。

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

2. ntfs-3gパッケージをダウンロードします。
3. ntfs-3gパッケージを解凍します。

   ```
   sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
   ```

4. ntfs-3gパッケージへのパスを入力します。

   ```
   sudo cd ntfs-3g_ntfsprogs-<package version>
   ```

5. ntfs-3gをインストールします。

   ```
   ./configure

   make

   make install
   ```

ステップ1g: ランタイム環境をセットアップする

`deploymcs.sh`を実行する前に、以下を実行してください。

- `/etc/xdl/mcs/mcs.conf`の変数を変更します。`mcs.conf`構成ファイルは、MCSとLinux VDAを設定するための変数を提供します。以下は、必要に応じて設定できる変数です。

- `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中のVDAの既存の構成を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDAの同等のファイルと同じになります。ただし、`dns`と`AD_INTEGRATION`の変数は引き続き構成する必要があります。デフォルト値はNで、これはマスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。
- `dns`: DNS IPアドレスを設定します。
- `AD_INTEGRATION`: Winbind、SSSD、またはPBISを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされているディストリビューションを参照してください。
- `WORKGROUP`: ADで構成した場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込むまたは更新するためのコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。この操作により、MCSプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg`ファイルの各行は、レジストリ値を設定または更新するためのコマンドです。

  たとえば、レジストリ値をそれぞれ書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加できます。

  ```
  create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
  ```

  ```
  update -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -v "Flags" -d "0x00000003"
  ```

ステップ1h: マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. (オプション) テンプレートVMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する`/etc/krb5.conf`、`/etc/samba/smb.conf`、および`/etc/sssd/sssd.conf`ファイルをカスタマイズします。

Winbindユーザーの場合は、`/etc/xdl/mcs/winbind_krb5.conf.tmpl`および`/etc/xdl/mcs/winbind_smb.conf.tmpl`テンプレートを更新します。

SSSDユーザーの場合は、`/etc/xdl/mcs/sssd.conf.tmpl`、`/etc/xdl/mcs/sssd_krb5.conf.tmpl`、および`/etc/xdl/mcs/sssd_smb.conf.tmpl`テンプレートを更新します。

> **注:**
>
> テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用します。

3. Citrix Hypervisorで、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。

ステップ2：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。必要に応じて、その他の構成タスクを実行します。詳細については、Studioを使用してマシンカタログを作成するを参照してください。

ステップ3: デリバリーグループを作成する

マシンカタログは、1つ以上のマシンカタログから選択されたマシンのコレクションです。マシンカタログは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳細については、デリバリーグループを作成するを参照してください。

MCS を使用して Azure 上に Linux VM を作成する

ステップ 1: Citrix Studio で Azure へのホスティング接続を作成する

1. Citrix Cloud™ の Citrix Studio で、**構成 > ホスティング > 接続とリソースの追加**を選択して、Azure への接続を作成します。

   Azure への接続を作成する(/en-us/linux-virtual-delivery-agent/2106/media/create-connection-to-azure.png)

2. 接続の種類として Microsoft Azure を選択します。

Azure 接続の種類を選択する(/en-us/linux-virtual-delivery-agent/2106/media/choose-connection-type.png)

3. Azure アカウントのサブスクリプション ID と接続名を入力します。

   Azure サブスクリプション ID を入力する(/en-us/linux-virtual-delivery-agent/2106/media/type-subscription-id.png)

   ホスティングペインに新しい接続が表示されます。

   新しい Azure 接続(/en-us/linux-virtual-delivery-agent/2106/media/new-connection-appears-in-hosting-pane.png)

ステップ 2: テンプレート VM でマスターイメージを準備する

マスターイメージには、オペレーティングシステム、仮想化されていないアプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ 2a: Ubuntu 18.04 用に cloud-init を構成する

VM の再起動または停止時に VDA ホスト名が保持されるようにするには、次のコマンドを実行します。

```
echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
```

/etc/cloud/cloud.cfg ファイルの **system_info** セクションに次の行が存在することを確認します。

```
system_info:

   network:

       renderers: ['netplan', 'eni', 'sysconfig']
```

ステップ 2b: (Ubuntu 16.04 のみ) OpenJDK 11 をインストールする

Ubuntu 16.04 で、次の手順を実行して OpenJDK 11 をインストールします。

1. `<https://jdk.java.net/archive/>` から最新の OpenJDK 11 をダウンロードします。
2. `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` コマンドを実行して、ダウンロードしたパッケージを解凍します。
3. (オプション) `mv jdk-11.0.2/ <target directory>` コマンドを実行して、OpenJDK をターゲットディレクトリに保存します。

4. `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` コマンドを実行して、Java ランタイムをセットアップします。
5. `java -version` コマンドを実行して、Java のバージョンを確認します。

ステップ 2c: テンプレート VM に Linux VDA パッケージをインストールする

> **注:**
>
> 現在実行中の VDA をテンプレート VM として使用する場合は、この手順をスキップしてください。
>
> テンプレート VM に Linux VDA パッケージをインストールする前に、.NET Core Runtime 3.1 をインストールします。詳細については、「インストール概要」を参照してください。
>
> Linux ディストリビューションに基づいて、次のコマンドを実行して Linux VDA の環境をセットアップします。

RHEL/CentOS の場合:

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debian の場合:

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12 の場合:

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ 2d: ntfs-3g を提供できる EPEL リポジトリをインストールする

RHEL 8/CentOS 8、RHEL 7/CentOS 7 に EPEL リポジトリをインストールします。これにより、後で deploymcs.sh を実行すると、EPEL リポジトリで利用可能な ntfs-3g パッケージがインストールされます。

手順 2e: SUSE 12 に ntfs-3g を手動でインストールする

SUSE 12 プラットフォームでは、ntfs-3g を提供するリポジトリはありません。ソースコードをダウンロードし、コンパイルして、ntfs-3g を手動でインストールします。

1. GNU Compiler Collection (GCC) コンパイラシステムと make パッケージをインストールします。

```
sudo zypper install gcc

sudo zypper install make
```

2. ntfs-3g パッケージをダウンロードします。
3. ntfs-3g パッケージを解凍します。

```
sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
```

4. ntfs-3g パッケージへのパスを入力します。

```
sudo cd ntfs-3g_ntfsprogs-<package version>
```

5. ntfs-3g をインストールします。

```
./configure

make

make install
```

手順 2f: ランタイム環境をセットアップする

`deploymcs.sh` を実行する前に、以下を実行します。

- `/etc/xdl/mcs/mcs.conf` の変数を変更します。`mcs.conf` 構成ファイルには、MCS と Linux VDA を設定するための変数が用意されています。以下にいくつかの変数を示します。このうち、`dns` と `AD_INTEGRATION` は設定する必要があります。

  注： 変数に複数の値を設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在実行中の VDA の構成ファイルと同じになります。ただし、`dns` と `AD_INTEGRATION` の変数は引き続き構成する必要があります。デフォルト値は N で、これはマスターイメージ上の構成テンプレートが MCS で作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: DNS IP アドレスを設定します。
  - `AD_INTEGRATION`: Winbind、SSSD、または PBIS を設定します。MSC がサポートする Linux ディストリビューションとドメイン参加方法の表については、この記事の サポートされているディストリビューション を参照してください。
  - `WORKGROUP`: ADで構成されている場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込んだり更新したりするために、`/etc/xdl/mcs/mcs_local_setting.reg` ファイルにコマンドラインを追加します。この操作により、MCSプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg` ファイルの各行は、レジストリ値を設定または更新するためのコマンドです。

  たとえば、レジストリ値をそれぞれ書き込んだり更新したりするために、次のコマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加できます。

  ```
  create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
  ```

  ```
  update -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -v "Flags" -d "0x00000003"
  ```

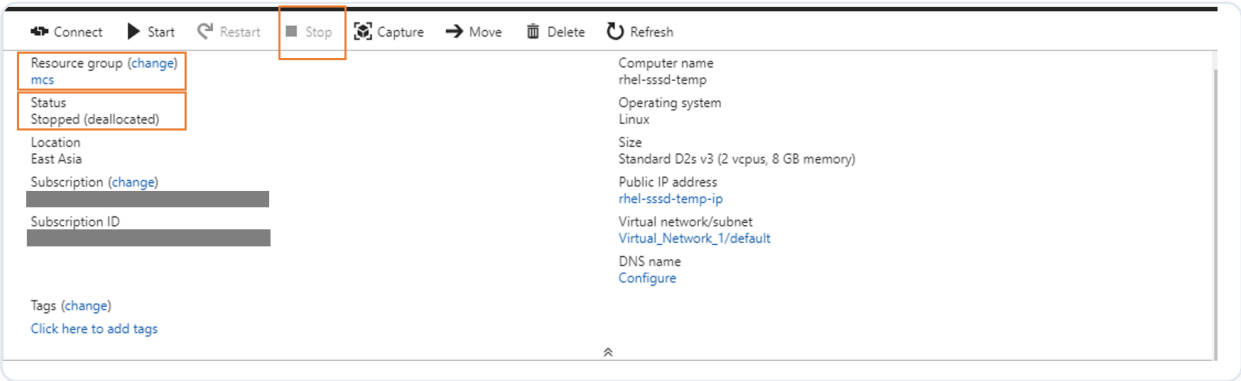
ステップ2g: マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. (オプション) テンプレートVMで、作成されたすべてのVM上の関連する`/etc/krb5.conf`、`/etc/samba/smb.conf`、および`/etc/sssd/sssd.conf`ファイルをカスタマイズするために、構成テンプレートを更新します。

   Winbindユーザーの場合、`/etc/xdl/mcs/winbind_krb5.conf.tmpl`および`/etc/xdl/mcs/winbind_smb.conf.tmpl`テンプレートを更新します。

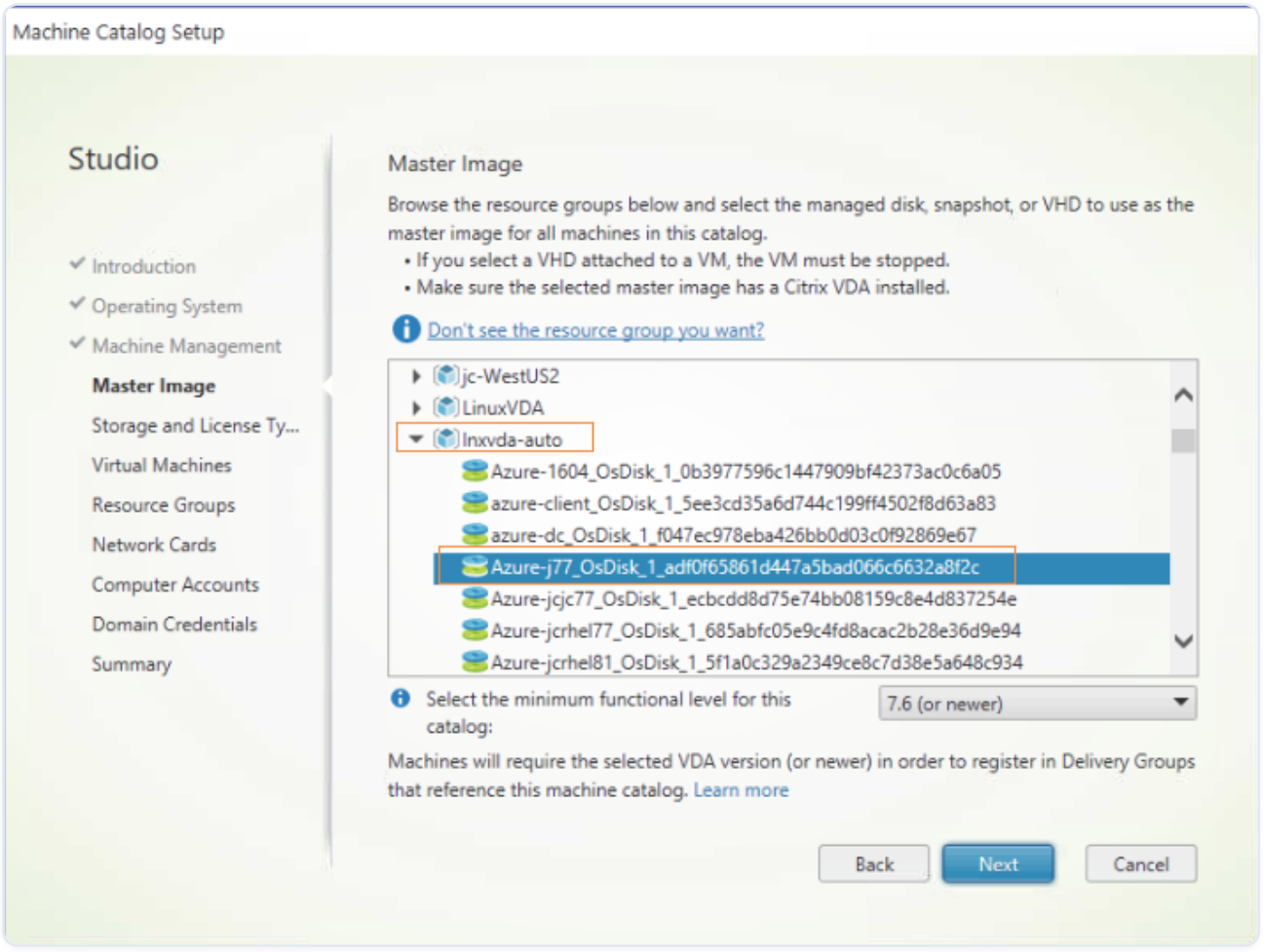
   SSSDユーザーの場合、`/etc/xdl/mcs/sssd.conf.tmpl`、`/etc/xdl/mcs/sssd_krb5.conf.tmpl`、および`/etc/xdl/mcs/sssd_smb.conf.tmpl`テンプレートを更新します。

   注: テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用します。
3. テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレート VMをシャットダウンします。テンプレートVMの電源状態が**停止済み（割り当て解除済み）**であることを確認します。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるにはその名前が必要です。



ステップ3: マシンカタログを作成する

Citrix Studioでマシンカタログを作成し、カタログで作成するVMの数を指定します。マシンカタログを作成する際、テンプレートVMが属するリソースグループからマスターイメージを選択し、テンプレートVMのVHDを見つけます。



必要に応じて他の構成タスクを実行します。詳細については、Studioを使用してマシンカタログを作成するを参照してください。

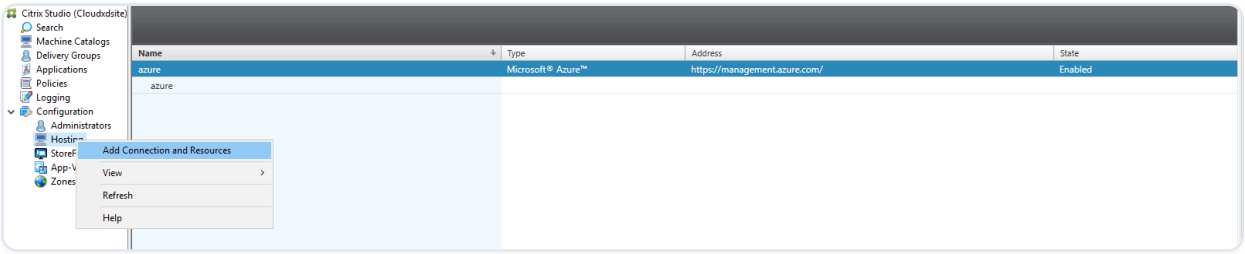
ステップ 4: デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳しくは、「デリバリーグループの作成」を参照してください。

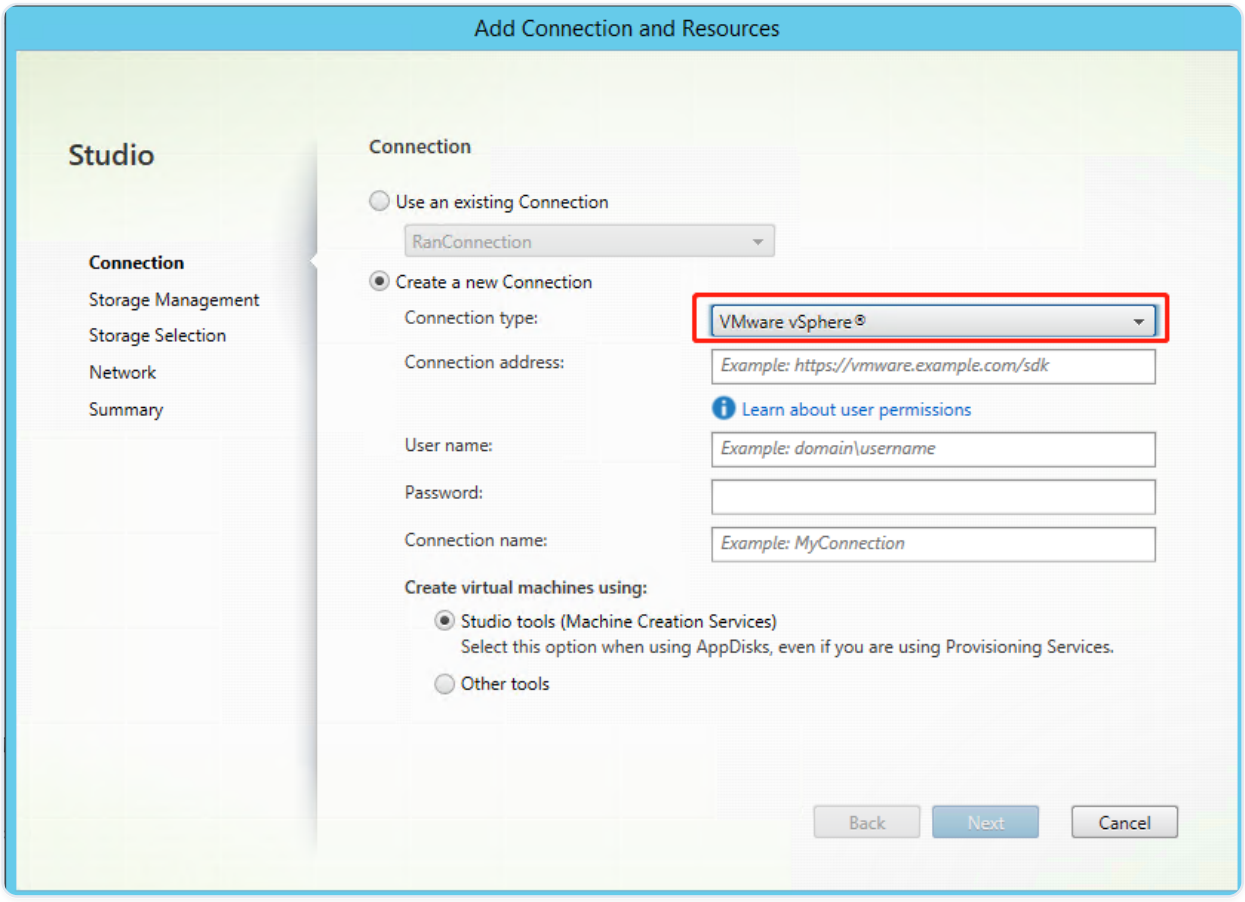
MCSを使用してVMware vSphere上でLinux VMを作成する

ステップ 1: Citrix StudioでVMwareへのホスティング接続を作成する

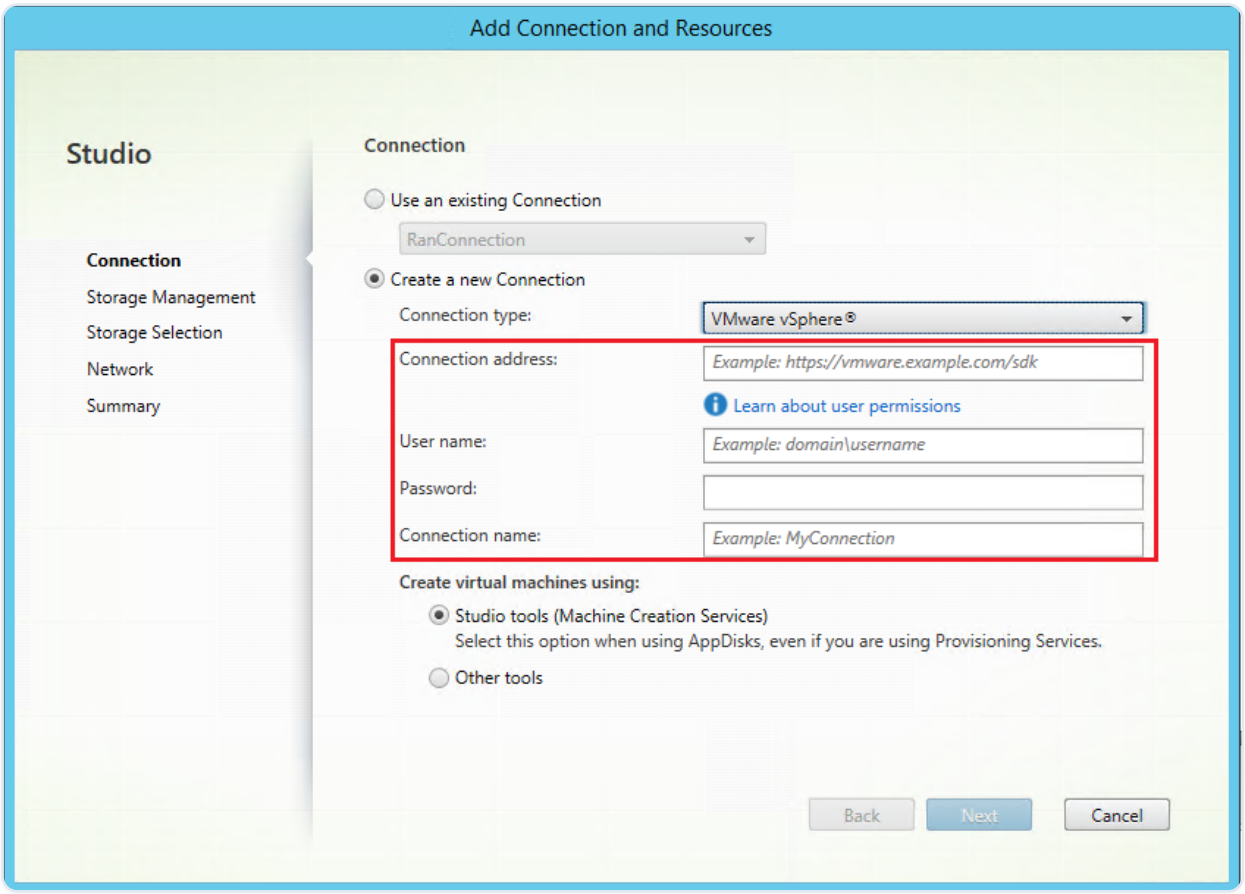
1. vSphere環境にvCenter Serverをインストールします。詳しくは、「VMware vSphere」を参照してください。
2. Citrix Studioで、**［構成］>［ホスティング］>［接続およびリソースの追加］**を選択し、VMware vSphereへの接続を作成します。



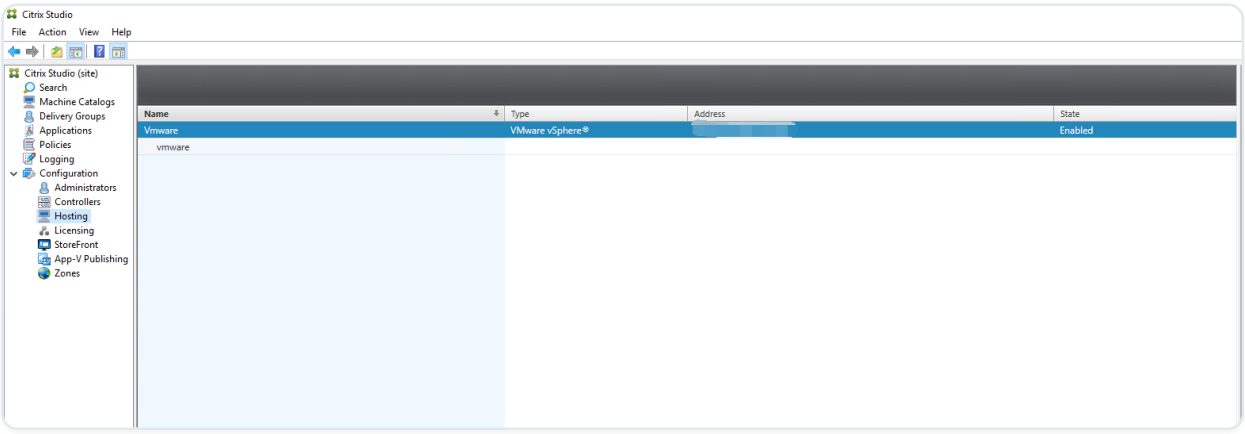
3. 接続の種類としてVMware vSphereを選択します。



4. VMwareアカウントの接続アドレス（vCenter ServerのURL）、ユーザー名とパスワード、および接続名を入力します。



ホスティングペインに新しい接続が表示されます。



ステップ 2: マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ 2a: (Ubuntu 16.04 のみ) OpenJDK 11 をインストールする

Ubuntu 16.04 で、次の手順を実行して OpenJDK 11 をインストールします。

1. `<https://jdk.java.net/archive/>` から最新の OpenJDK 11 をダウンロードします。

2. `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` コマンドを実行して、ダウンロードしたパッケージを解凍します。
3. (オプション) `mv jdk-11.0.2/ <target directory>` コマンドを実行して、OpenJDK をターゲットディレクトリに保存します。
4. `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` コマンドを実行して、Java ランタイムをセットアップします。
5. `java -version` コマンドを実行して、Java のバージョンを確認します。

ステップ 2b: テンプレート VM に Linux VDA パッケージをインストールする

> **注:**
>
> 現在実行中の VDA をテンプレート VM として使用する場合は、この手順をスキップしてください。

> テンプレート VM に Linux VDA パッケージをインストールする前に、.NET Core Runtime 3.1 をインストールします。詳しくは、「インストール概要」を参照してください。

> Linux ディストリビューションに基づいて、次のコマンドを実行して Linux VDA の環境をセットアップします。

RHEL/CentOS の場合:

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debian の場合:

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12 の場合:

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ 2c: ntfs-3g を提供できる EPEL リポジトリをインストールする

EPELリポジトリをRHEL 8/CentOS 8、RHEL 7/CentOS 7にインストールします。これにより、後でdeploymcs.shを実行したときに、EPELリポジトリで利用可能なntfs-3gパッケージがインストールされます。

ステップ2d: SUSE 12にntfs-3gを手動でインストールする

SUSE 12プラットフォームでは、ntfs-3gを提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルして、ntfs-3gを手動でインストールします。

1. GNU Compiler Collection (GCC) コンパイラシステムとmakeパッケージをインストールします。

```
sudo zypper install gcc

sudo zypper install make
```

2. ntfs-3gパッケージをダウンロードします。
3. ntfs-3gパッケージを解凍します。

```
sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
```

4. ntfs-3gパッケージへのパスを入力します。

```
sudo cd ntfs-3g_ntfsprogs-<package version>
```

5. ntfs-3gをインストールします。

```
./configure

make

make install
```

ステップ2e: ランタイム環境をセットアップする

`deploymcs.sh`を実行する前に、以下を実行します。

- `/etc/xdl/mcs/mcs.conf`内の変数を変更します。`mcs.conf`構成ファイルは、MCSとLinux VDAを設定するための変数を提供します。以下はいくつかの変数であり、そのうち`dns`と`AD_INTEGRATION`は設定する必要があります。

  注: 変数に複数の値を設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例: LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中のVDAの既存の構成を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDAの対応するファイルと同じになります。ただし、`dns`と`AD_INTEGRATION`の変数は引き続き構成する必要があります。デフォルト値はNで、これはマスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: DNS IPアドレスを設定します。
  - `AD_INTEGRATION`: Winbind、SSSD、またはPBISを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされているディストリビューションを参照してください。
  - `WORKGROUP`: ADで構成されている場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込むまたは更新するためのコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。この操作により、MCSでプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg`ファイルの各行は、レジストリ値を設定または更新するためのコマンドです。

  たとえば、レジストリ値をそれぞれ書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加できます。

```
create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
\ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
```

```
update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
\ClipboardSelection" -v "Flags" -d "0x00000003"
```

手順2f：マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. （オプション）テンプレートVMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する/etc/krb5.conf、/etc/samba/smb.conf、および/etc/sssd/sssd.confファイルをカスタマイズします。

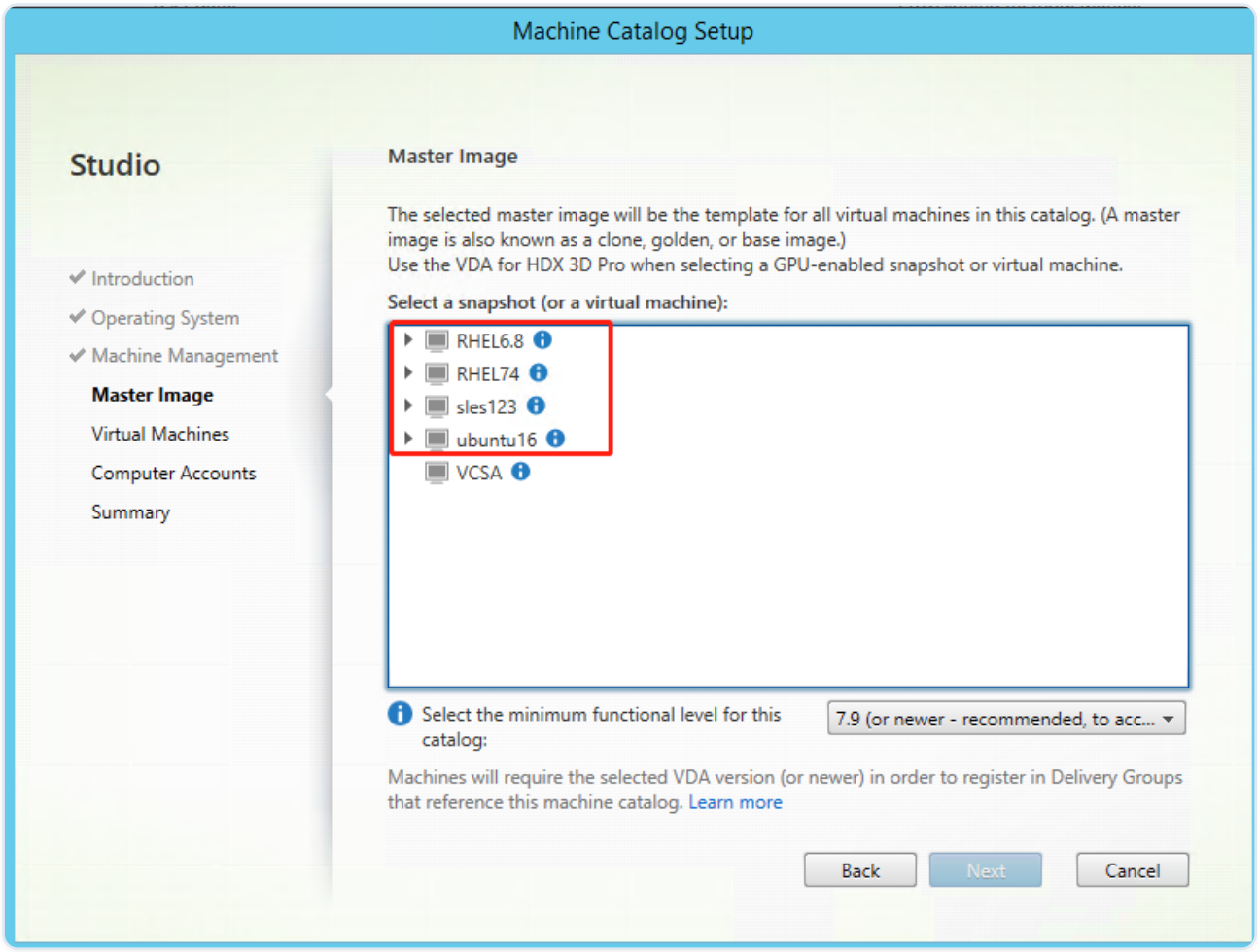
   Winbindユーザーの場合は、/etc/xdl/mcs/winbind_krb5.conf.tmplおよび/etc/xdl/mcs/winbind_smb.conf.tmplテンプレートを更新します。

   SSSDユーザーの場合は、/etc/xdl/mcs/sssd.conf.tmpl、/etc/xdl/mcs/sssd_krb5.conf.tmpl、および/etc/xdl/mcs/sssd_smb.conf.tmplテンプレートを更新します。

   注： テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用してください。
3. テンプレートVMへのアプリケーションのインストールが完了したら、VMwareからテンプレートVMをシャットダウンします。テンプレートVMのスナップショットを作成します。

手順3：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成する際に、スナップショットリストからマスターイメージを選択します。



必要に応じて、その他の構成タスクを実行します。詳細については、「Studioを使用してマシンカタログを作成する」を参照してください。

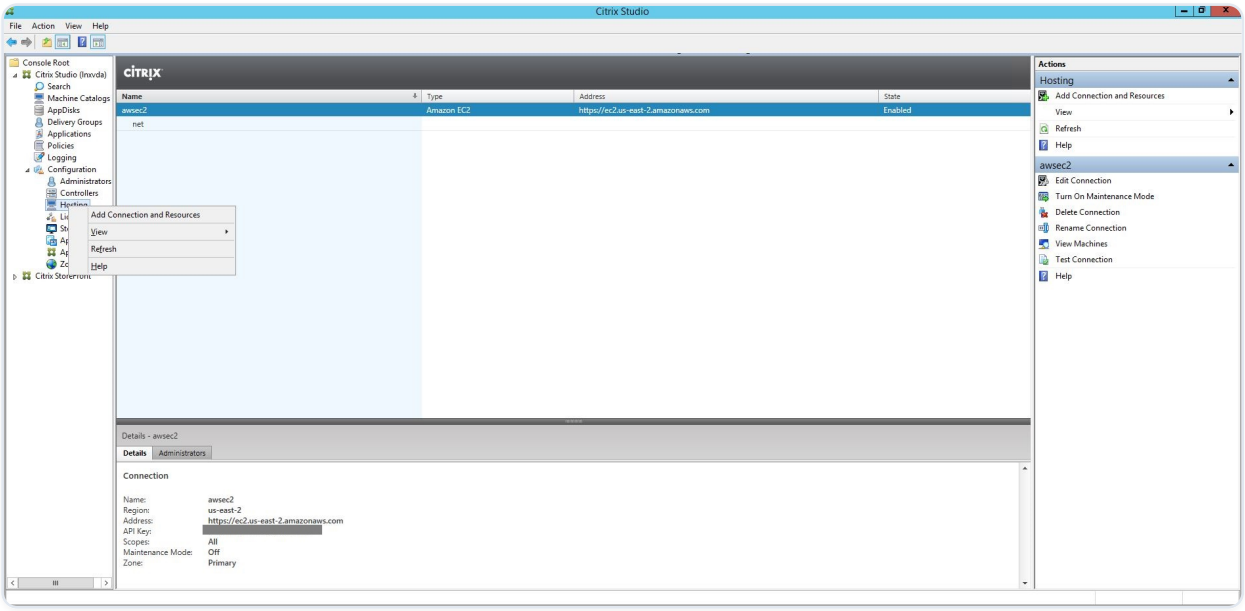
手順4：デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンのコレクションです。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳細については、「デリバリーグループの作成」を参照してください。

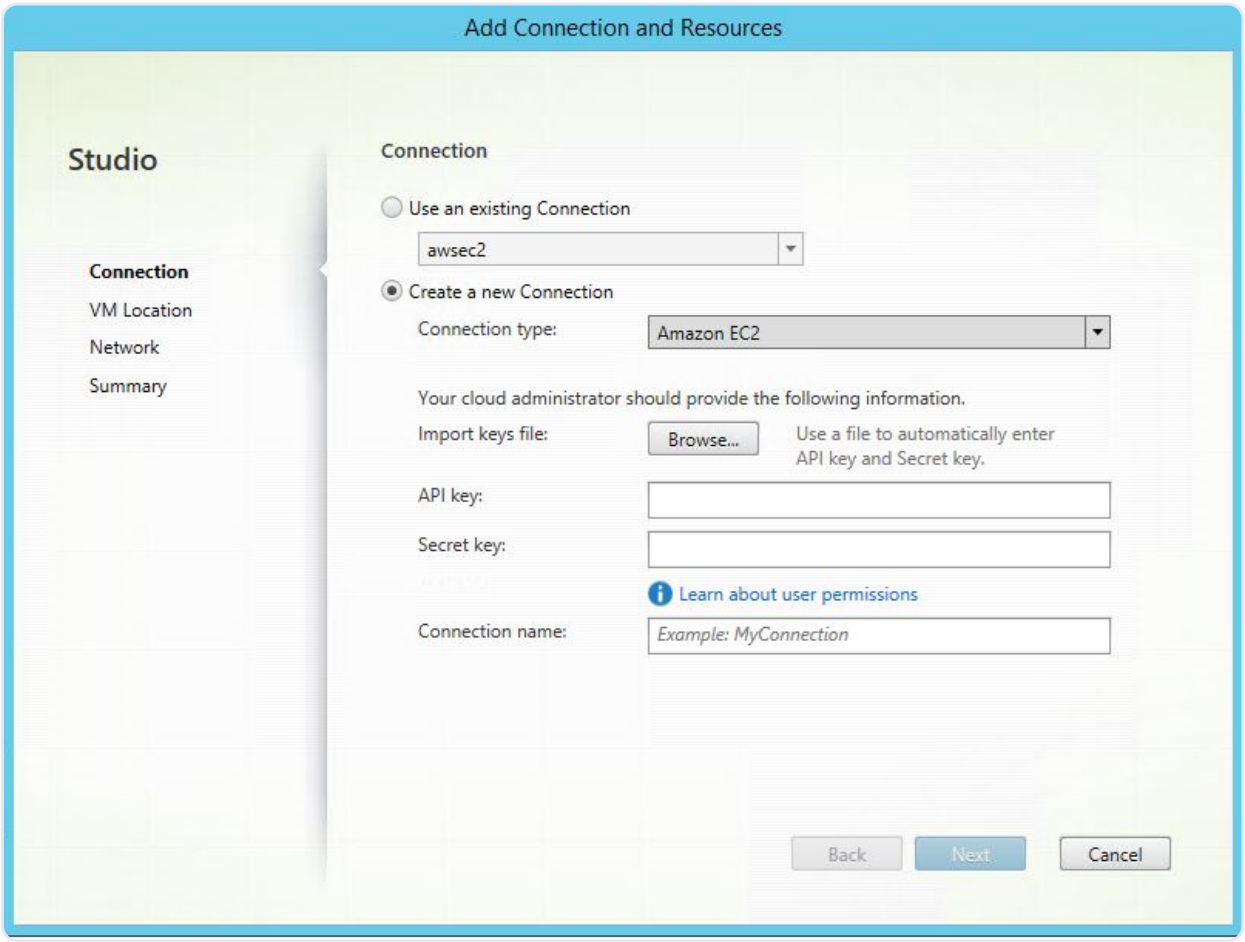
MCSを使用してAWS上にLinux VMを作成する

ステップ1：Citrix StudioでAWSへのホスティング接続を作成する

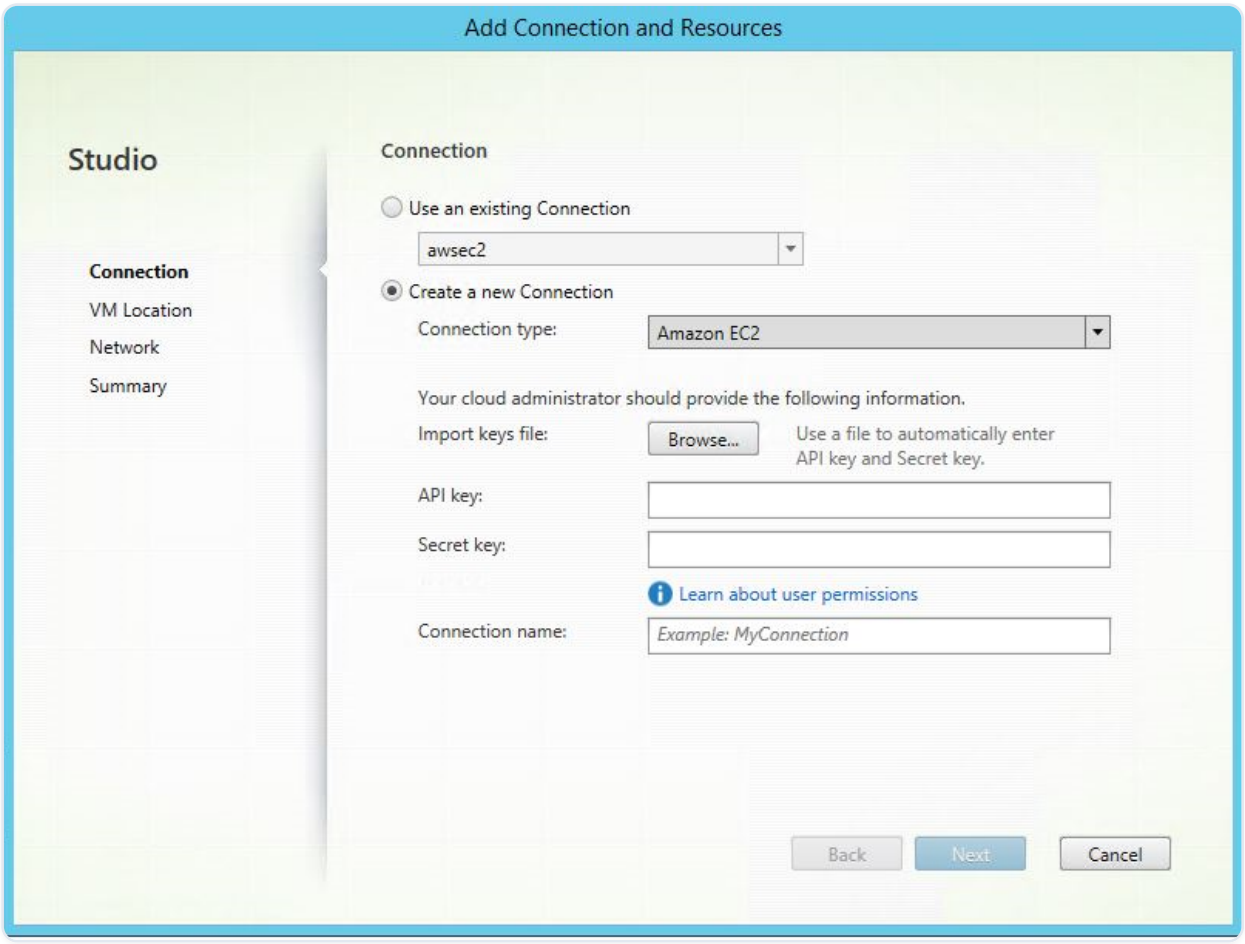
1. Citrix CloudのCitrix Studioで、**構成** > **ホスティング** > **接続とリソースの追加** を選択して、AWSへの接続を作成します。



2. 接続の種類として**Amazon EC2**を選択します。



3. AWSアカウントのAPIキーとシークレットキーを入力し、接続名を入力します。



APIキーはアクセスキーIDであり、**シークレットキー**はシークレットアクセスキーです。これらはアクセスキーペアと見なされます。シークレットアクセスキーを紛失した場合は、アクセスキーを削除して別のキーを作成できます。アクセスキーを作成するには、次の手順を実行します。

a. AWSサービスにサインインします。
b. Identity and Access Management (IAM) コンソールに移動します。
c. 左側のナビゲーションペインで、**ユーザー**を選択します。
d. 対象ユーザーを選択し、下にスクロールして**セキュリティ認証情報**タブを選択します。
e. 下にスクロールして**アクセスキーの作成**をクリックします。新しいウィンドウが表示されます。
f. 「**Download .csv file**」をクリックし、アクセスキーを安全な場所に保存します。

ホスティングペインに新しい接続が表示されます。

新しいAWS接続の画像(/en-us/linux-virtual-delivery-agent/2106/media/new-aws-connection.png)

ステップ2：マスターイメージの準備

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ2a：cloud-initの構成

1. EC2インスタンスの再起動または停止時にVDAホスト名が保持されるようにするには、次のコマンドを実行してVDAホスト名を保持します。

   ```
   echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
   ```

   Ubuntu 18.04の場合、/etc/cloud/cloud.cfg ファイルの system_info セクションに以下の行が存在することを確認してください。

   ```
   system_info:

       network:

           renderers: ['netplan', 'eni', 'sysconfig']
   ```

2. AWSでMCSによって作成されたVMにリモートでアクセスするためにSSHを使用するには、それらのVMにはキー名がアタッチされていないため、パスワード認証を有効にします。必要に応じて、次の手順を実行します。

   - `cloud-init` 構成ファイル /etc/cloud/cloud.cfg を編集します。「**ssh_pwauth: true**」の行が存在することを確認します。「**set-password**」の行とそれに続く行が存在する場合は、削除またはコメントアウトします。

     ```
     users:

     - default
     ```

   - `cloud-init` によって作成されたデフォルトユーザー `ec2-user` または `ubuntu` を使用する予定がある場合は、`passwd` コマンドを使用してユーザーパスワードを変更できます。新しいパスワードは、後でMCSによって作成されたVMにログインするために覚えておいてください。
   - `/etc/ssh/sshd_config` ファイルを編集して、次の行が存在することを確認します。

```
PasswordAuthentication yes
```

ファイルを保存し、`sudo service sshd restart`コマンドを実行します。

ステップ2b：（Ubuntu 16.04のみ）OpenJDK 11のインストール

Ubuntu 16.04でOpenJDK 11をインストールするには、次の手順を実行します。

1. 最新のOpenJDK 11を`<https://jdk.java.net/archive/>`からダウンロードします。
2. ダウンロードしたパッケージを解凍するには、`tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz`コマンドを実行します。
3. (オプション) OpenJDKをターゲットディレクトリに保存するには、`mv jdk-11.0.2/ <target directory>`コマンドを実行します。
4. Javaランタイムをセットアップするには、`update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000`コマンドを実行します。
5. Javaのバージョンを確認するには、`java -version`コマンドを実行します。

ステップ2c: テンプレートVMにLinux VDAパッケージをインストールする

注:

現在実行中のVDAをテンプレートVMとして使用する場合は、この手順をスキップしてください。

テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Core Runtime 3.1をインストールしてください。詳細については、「インストール概要」を参照してください。

お使いのLinuxディストリビューションに基づいて、Linux VDAの環境をセットアップするために次のコマンドを実行します。

RHEL/CentOSの場合:

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debianの場合:

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12の場合:

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ2d: ntfs-3gを提供するEPELリポジトリをインストールする

RHEL 8/CentOS 8、RHEL 7/CentOS 7にEPELリポジトリをインストールすると、後でdeploymcs.shを実行したときにEPELリポジトリで利用可能なntfs-3gパッケージがインストールされます。

ステップ2e: SUSE 12にntfs-3gを手動でインストールする

SUSE 12プラットフォームでは、ntfs-3gを提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルして、ntfs-3gを手動でインストールしてください。

1. GNU Compiler Collection (GCC) コンパイラシステムとmakeパッケージをインストールします。

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

2. ntfs-3gパッケージをダウンロードします。
3. ntfs-3gパッケージを解凍します。

   ```
   sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
   ```

4. ntfs-3gパッケージへのパスを入力します。

   ```
   sudo cd ntfs-3g_ntfsprogs-<package version>
   ```

5. ntfs-3gをインストールします。

```
./configure

make

make install
```

ステップ2f: ランタイム環境をセットアップする

`deploymcs.sh`を実行する前に、以下を実行してください。

- `/etc/xdl/mcs/mcs.conf`内の変数を変更します。`mcs.conf`構成ファイルには、MCSとLinux VDAを設定するための変数が用意されています。以下にいくつかの変数を示しますが、そのうち`dns`と`AD_INTEGRATION`は設定する必要があります。

  注： 変数に複数の値を設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中のVDAの既存の構成を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDAの同等のファイルと同じになります。ただし、`dns`と`AD_INTEGRATION`の変数は引き続き構成する必要があります。デフォルト値はNで、これはマスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: DNS IPアドレスを設定します。
  - `AD_INTEGRATION`: Winbind、SSSD、またはPBISを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされているディストリビューションを参照してください。
  - `WORKGROUP`: ADで構成されている場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込むまたは更新するためのコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。この操作により、MCSプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg`ファイル内の各行は、レジストリ値を設定または更新するためのコマンドです。

  たとえば、レジストリ値をそれぞれ書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加できます。

```
create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
\ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
```

```
update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
\ClipboardSelection" -v "Flags" -d "0x00000003"
```

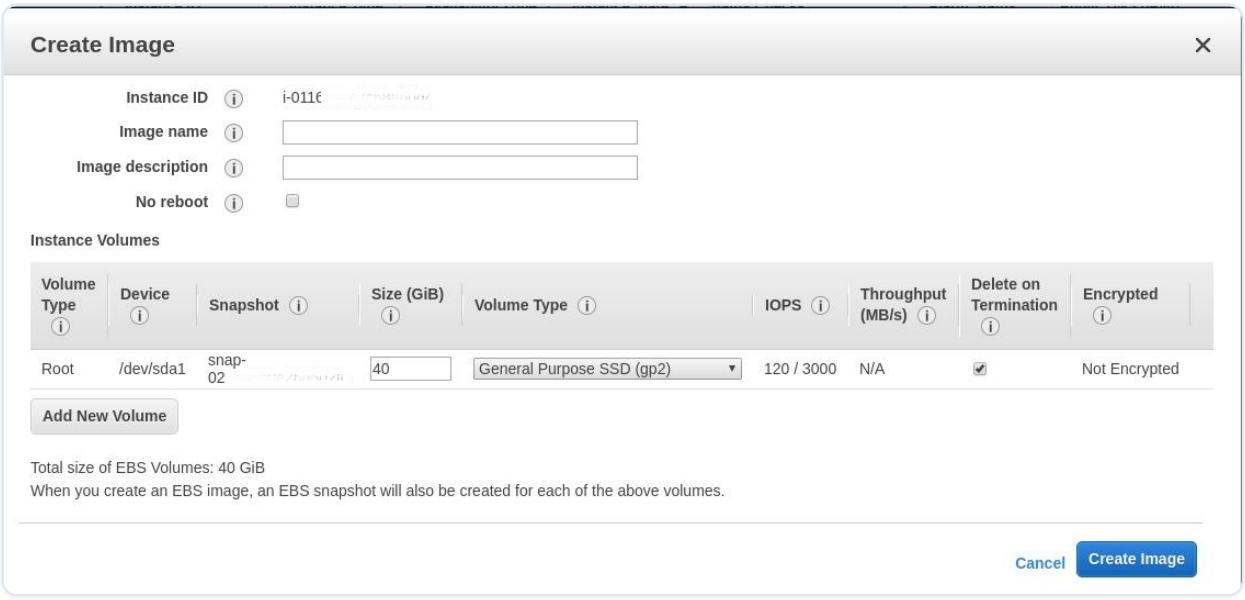
ステップ2g：マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. （オプション）テンプレートVMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する`/etc/krb5.conf`、`/etc/samba/smb.conf`、および`/etc/sssd/sssd.conf`ファイルをカスタマイズします。

   Winbindユーザーの場合は、`/etc/xdl/mcs/winbind_krb5.conf.tmpl`および`/etc/xdl/mcs/winbind_smb.conf.tmpl`テンプレートを更新します。

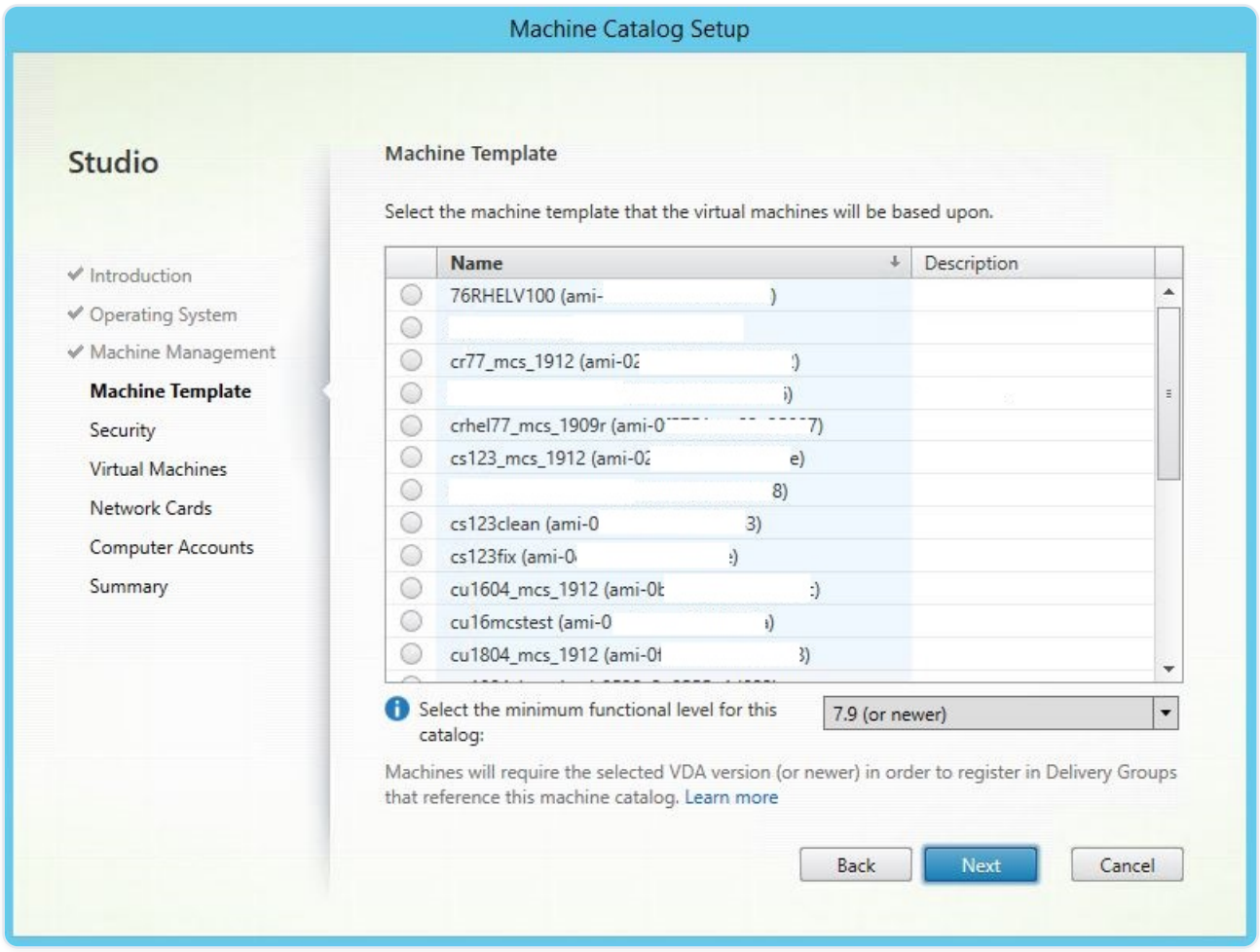
   SSSDユーザーの場合は、`/etc/xdl/mcs/sssd.conf.tmpl`、`/etc/xdl/mcs/sssd_krb5.conf.tmpl`、および`/etc/xdl/mcs/sssd_smb.conf.tmpl`テンプレートを更新します。

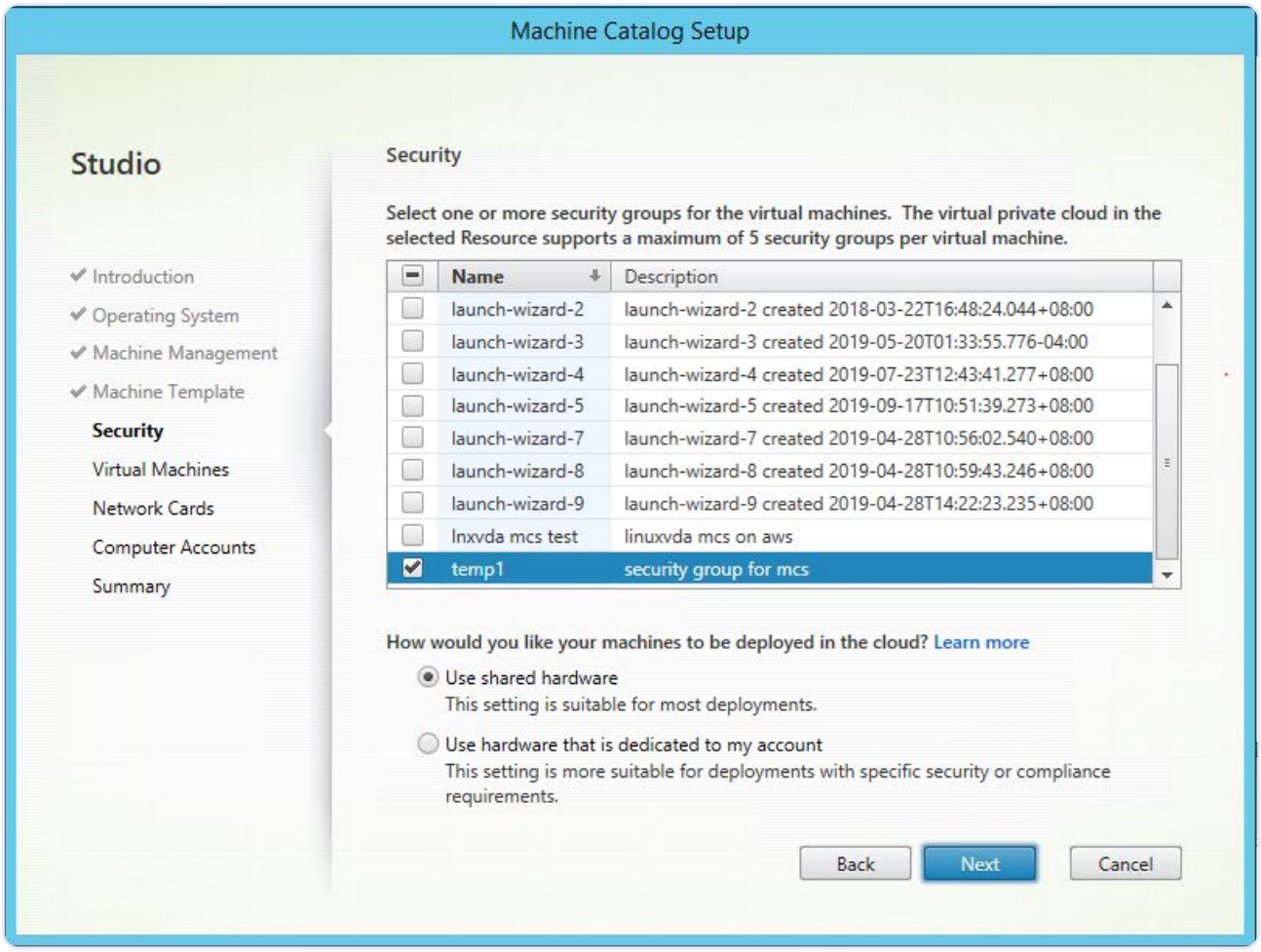
   注： テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用してください。
3. テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**であることを確認してください。
4. テンプレートVMを右クリックし、**イメージ > イメージの作成**を選択します。必要な情報を入力し、設定を行います。**イメージの作成**をクリックします。



ステップ3：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成する際は、マシンのテンプレート（以前に作成したマスターイメージ）を選択し、1つ以上のセキュリティグループを選択します。





必要に応じて、その他の構成タスクを実行します。詳細については、「Studioを使用してマシンカタログを作成する」を参照してください。

ステップ 4: デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳しくは、「デリバリーグループの作成」を参照してください。

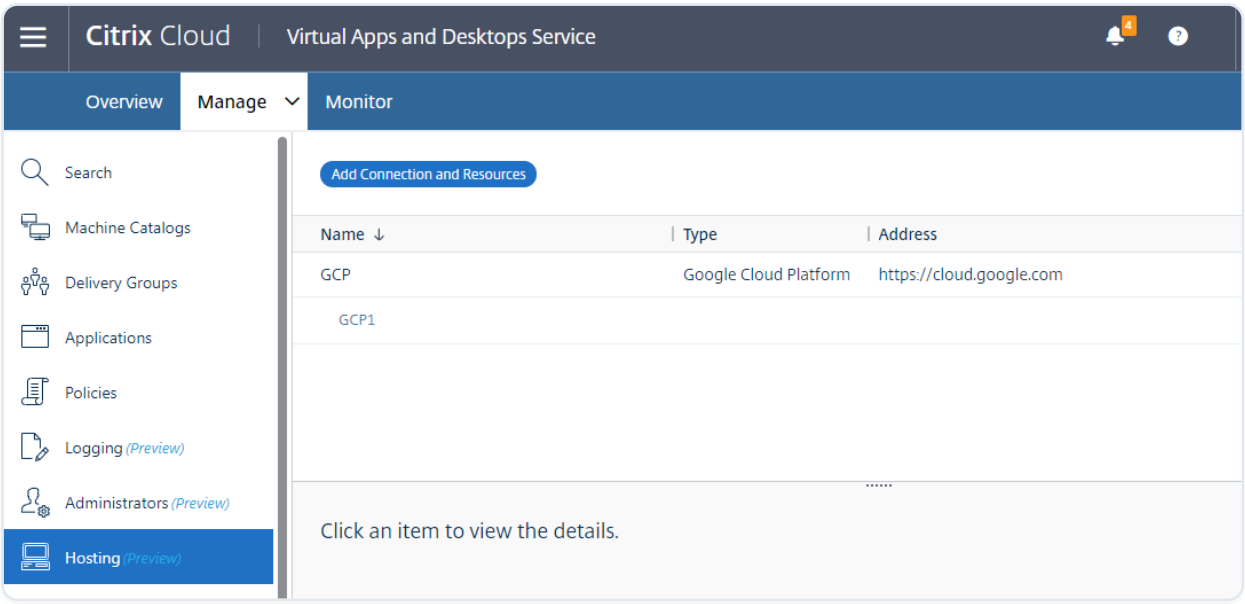
MCSを使用してGCP上にLinux VMを作成する

ステップ 1: GCP環境をセットアップする

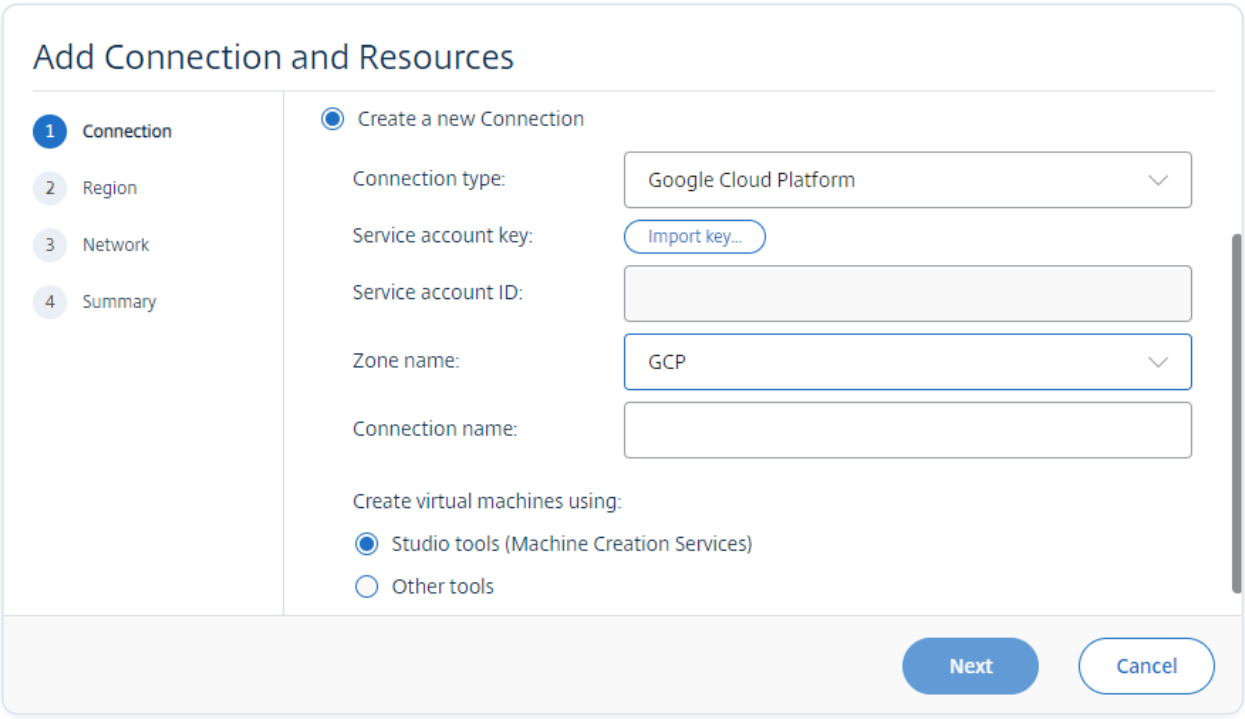
詳しくは、「Google Cloud Platform仮想化環境」を参照してください。

ステップ 2: Citrix StudioでGCPへのホスティング接続を作成する

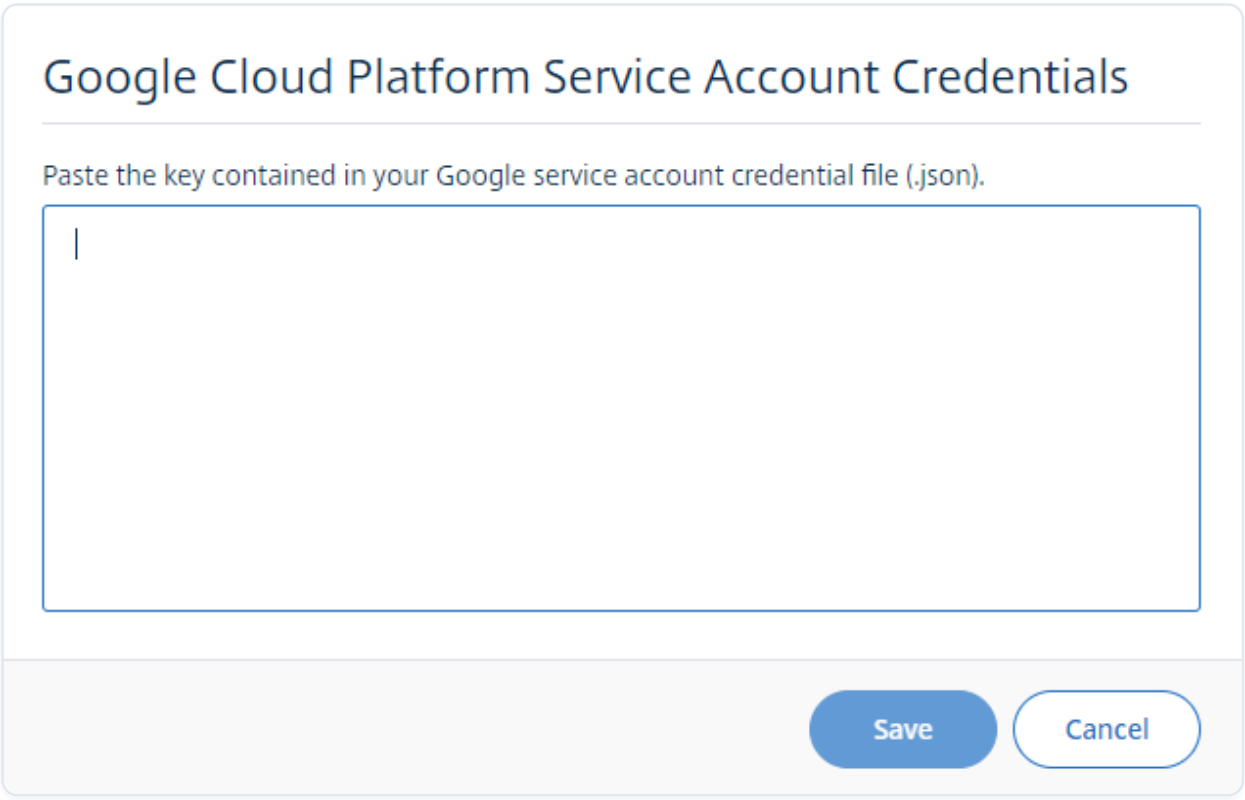
1. Citrix CloudのCitrix Studioで、 **［構成］** > **［ホスティング］** > **［接続とリソースの追加］** を選択して、GCPへの接続を作成します。



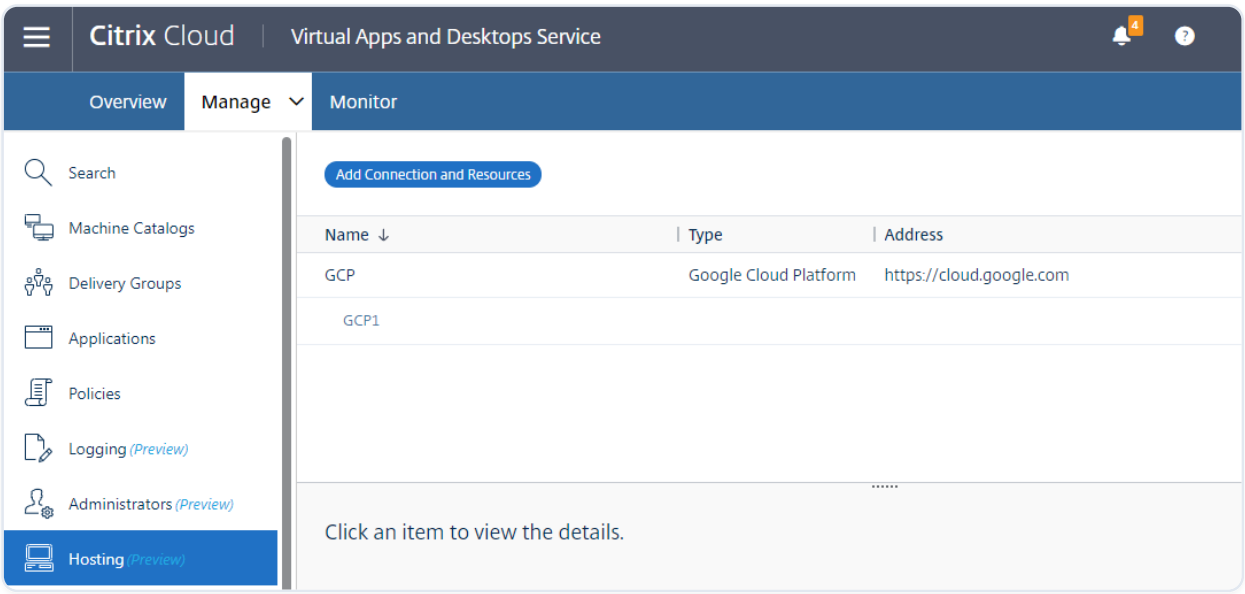
2. 接続の種類として**Google Cloud Platform**を選択します。



3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。



ホスティングペインに新しい接続が表示されます。



ステップ 3: マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ3a：（Ubuntu 16.04のみ）OpenJDK 11をインストールする

Ubuntu 16.04でOpenJDK 11をインストールするには、次の手順を実行します。

1. `<https://jdk.java.net/archive/>`から最新のOpenJDK 11をダウンロードします。
2. ダウンロードしたパッケージを解凍するには、`tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz`コマンドを実行します。
3. （オプション）OpenJDKをターゲットディレクトリに保存するには、`mv jdk-11.0.2/ <target directory>`コマンドを実行します。
4. Javaランタイムをセットアップするには、`update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000`コマンドを実行します。
5. Javaのバージョンを確認するには、`java -version`コマンドを実行します。

ステップ3b：テンプレートVMにLinux VDAパッケージをインストールする

> **注:**
>
> 現在実行中のVDAをテンプレートVMとして使用する場合は、この手順をスキップしてください。
>
> テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Core Runtime 3.1をインストールしてください。詳細については、「インストール概要」を参照してください。
>
> お使いのLinuxディストリビューションに基づいて、Linux VDAの環境をセットアップするために次のコマンドを実行します。

RHEL/CentOSの場合：

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debianの場合：

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12の場合：

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ 3c: ntfs-3g を提供できる EPEL リポジトリをインストールする

RHEL 8/CentOS 8、RHEL 7/CentOS 7 に EPEL リポジトリをインストールします。これにより、後で deploymcs.sh を実行すると、EPEL リポジトリで利用可能な ntfs-3g パッケージがインストールされます。

ステップ 3d: SUSE 12 に ntfs-3g を手動でインストールする

SUSE 12 プラットフォームでは、ntfs-3g を提供するリポジトリはありません。ソースコードをダウンロードし、コンパイルして、ntfs-3g を手動でインストールします。

1. GNU Compiler Collection (GCC) コンパイラシステムと make パッケージをインストールします。

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

2. ntfs-3g パッケージをダウンロードします。
3. ntfs-3g パッケージを解凍します。

   ```
   sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
   ```

4. ntfs-3g パッケージへのパスを入力します。

   ```
   sudo cd ntfs-3g_ntfsprogs-<package version>
   ```

5. ntfs-3g をインストールします。

   ```
   ./configure

   make

   make install
   ```

ステップ 3e: ランタイム環境をセットアップする

`deploymcs.sh` を実行する前に、次の操作を行います。

- `/etc/xdl/mcs/mcs.conf` の変数を変更します。`mcs.conf` 構成ファイルは、MCS と Linux VDA を設定するための変数を提供します。以下にいくつかの変数を示します。そのうち `dns` と `AD_INTEGRATION` は設定する必要があります。

  注：変数を複数の値で設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在実行中の VDA の対応するファイルと同じになります。ただし、`dns` と `AD_INTEGRATION` の変数は引き続き構成する必要があります。デフォルト値は N です。これは、マスターイメージ上の構成テンプレートが、MCS で作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: DNS IP アドレスを設定します。
  - `AD_INTEGRATION`: Winbind、SSSD、またはPBISを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされているディストリビューションを参照してください。
  - `WORKGROUP`: ADで構成されている場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込むか更新するために、`/etc/xdl/mcs/mcs_local_setting.reg`ファイルにコマンドラインを追加します。この操作により、MCSプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg`ファイル内の各行は、レジストリ値を設定または更新するためのコマンドです。

  例えば、`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに以下のコマンドラインを追加して、それぞれレジストリ値を書き込むか更新できます。

  ```
  create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
  ```

  ```
  update -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -v "Flags" -d "0x00000003"
  ```

ステップ3f: マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. (オプション) テンプレートVMで、構成テンプレートを更新して、作成されたすべてのVMで関連する`/etc/krb5.conf`、`/etc/samba/smb.conf`、および`/etc/sssd/sssd.conf`ファイルをカスタマイズします。

   Winbindユーザーの場合、`/etc/xdl/mcs/winbind_krb5.conf.tmpl`および`/etc/xdl/mcs/winbind_smb.conf.tmpl`テンプレートを更新します。

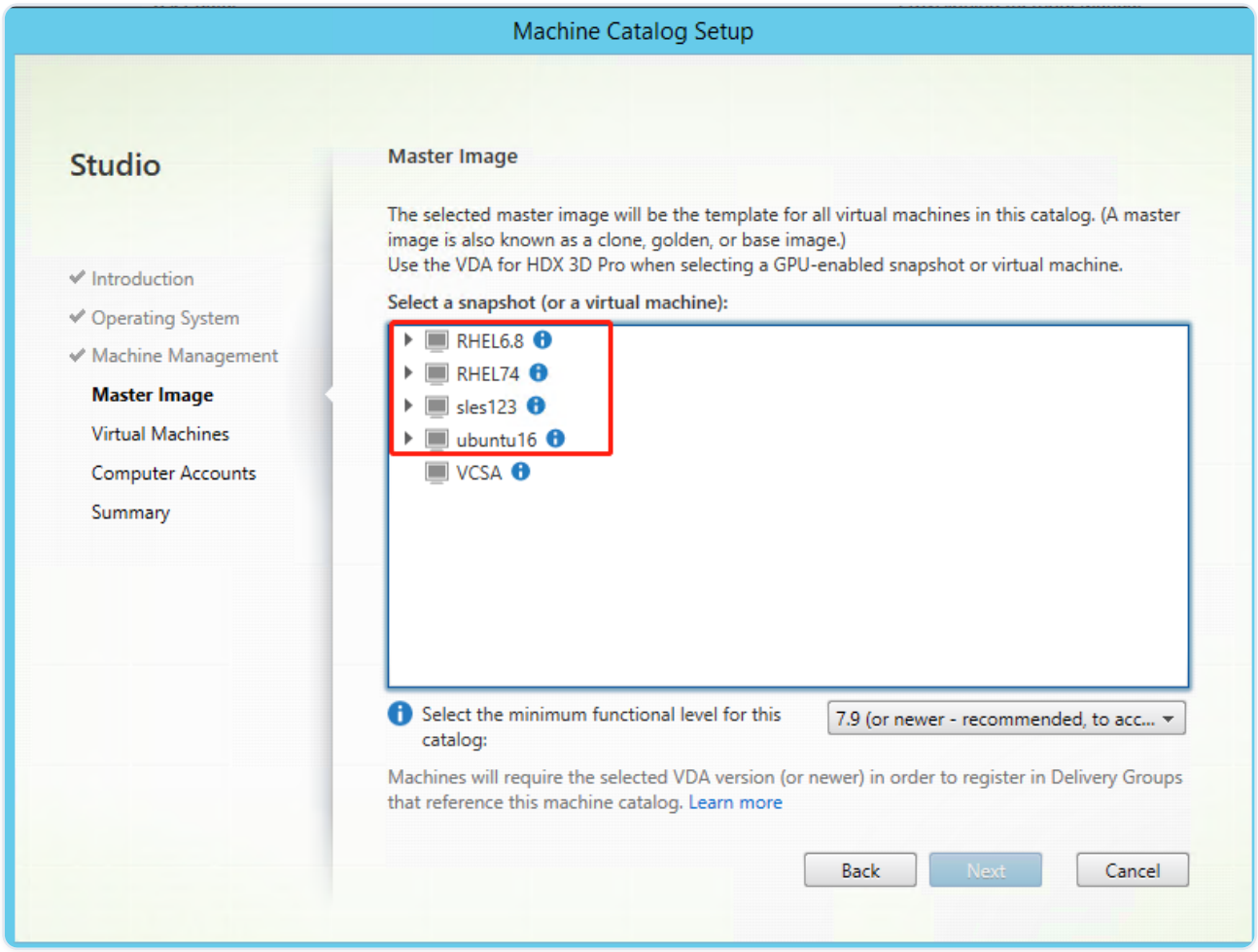
   SSSDユーザーの場合、`/etc/xdl/mcs/sssd.conf.tmpl`、`/etc/xdl/mcs/sssd_krb5.conf.tmpl`、および`/etc/xdl/mcs/sssd_smb.conf.tmpl`テンプレートを更新します。

   > **注:**
   >
   > テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用します。

3. テンプレートVMへのアプリケーションのインストールが完了したら、VMwareからテンプレートVMをシャットダウンします。テンプレートVMのスナップショットを作成します。

ステップ4: マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成する際、スナップショットリストからマスターイメージを選択します。



必要に応じて、その他の構成タスクを実行します。詳しくは、「Studioを使用してマシンカタログを作成する」を参照してください。

ステップ 5: デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳しくは、「デリバリーグループの作成」を参照してください。

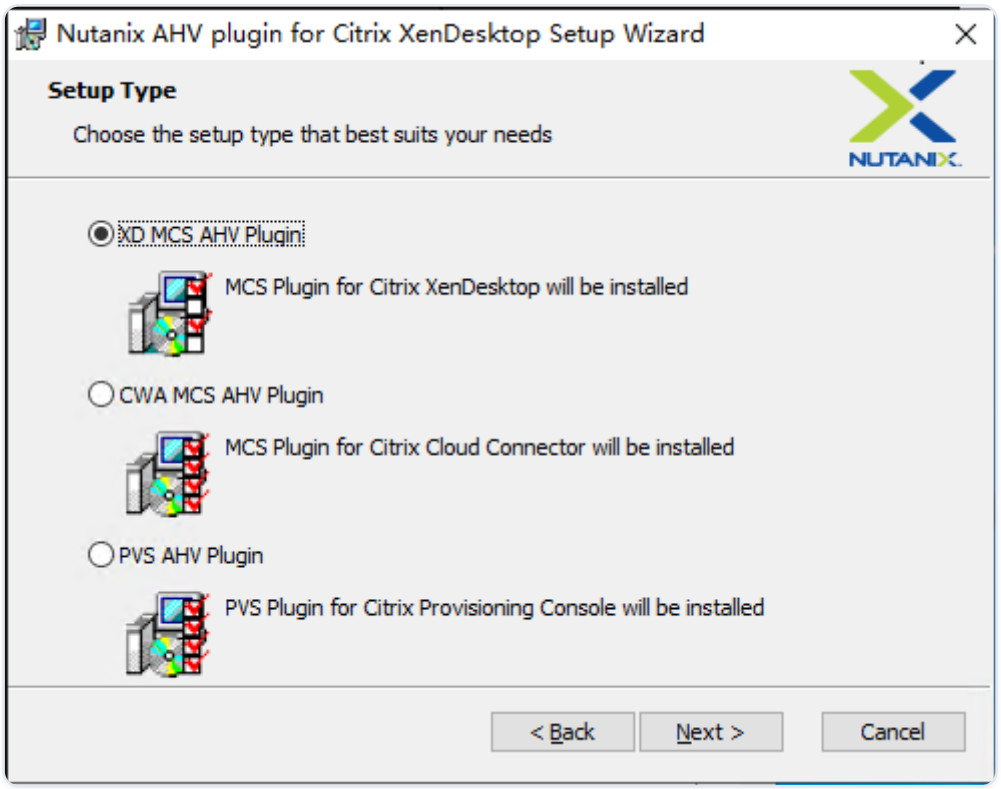
Nutanix AHV上でMCSを使用してLinux VMを作成する

ステップ 1: Nutanix AHVプラグインをインストールして登録する

NutanixからNutanix AHVプラグインパッケージを入手します。Citrix Virtual Apps and Desktops環境にプラグインをインストールして登録します。詳しくは、Nutanixサポートポータルで入手できるNutanix Acropolis MCSプラグインインストールガイドを参照してください。

ステップ 1a: オンプレミスのDelivery Controller向けNutanix AHVプラグインをインストールして登録する

Citrix Virtual Apps™ and Desktopsをインストールした後、Delivery Controllerに**XD MCS AHV Plugin**を選択してインストールします。



ステップ 1b: クラウドDelivery Controller向けNutanix AHVプラグインをインストールして登録する

Citrix Virtual Apps and Desktopsをインストールした後、Citrix Cloud Connector向けに**CWA MCS AHV Plugin**を選択してインストールします。Citrix Cloudテナントに登録されているすべてのCitrix Cloud Connectorにプラグインをインストールします。AHVのないリソースの場所を提供するCitrix Cloud Connectorであっても、登録する必要があります。

ステップ 1c: プラグインのインストール後に次の手順を完了します

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`にNutanix Acropolisフォルダーが作成されていることを確認します。
- `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"`コマンドを実行します。

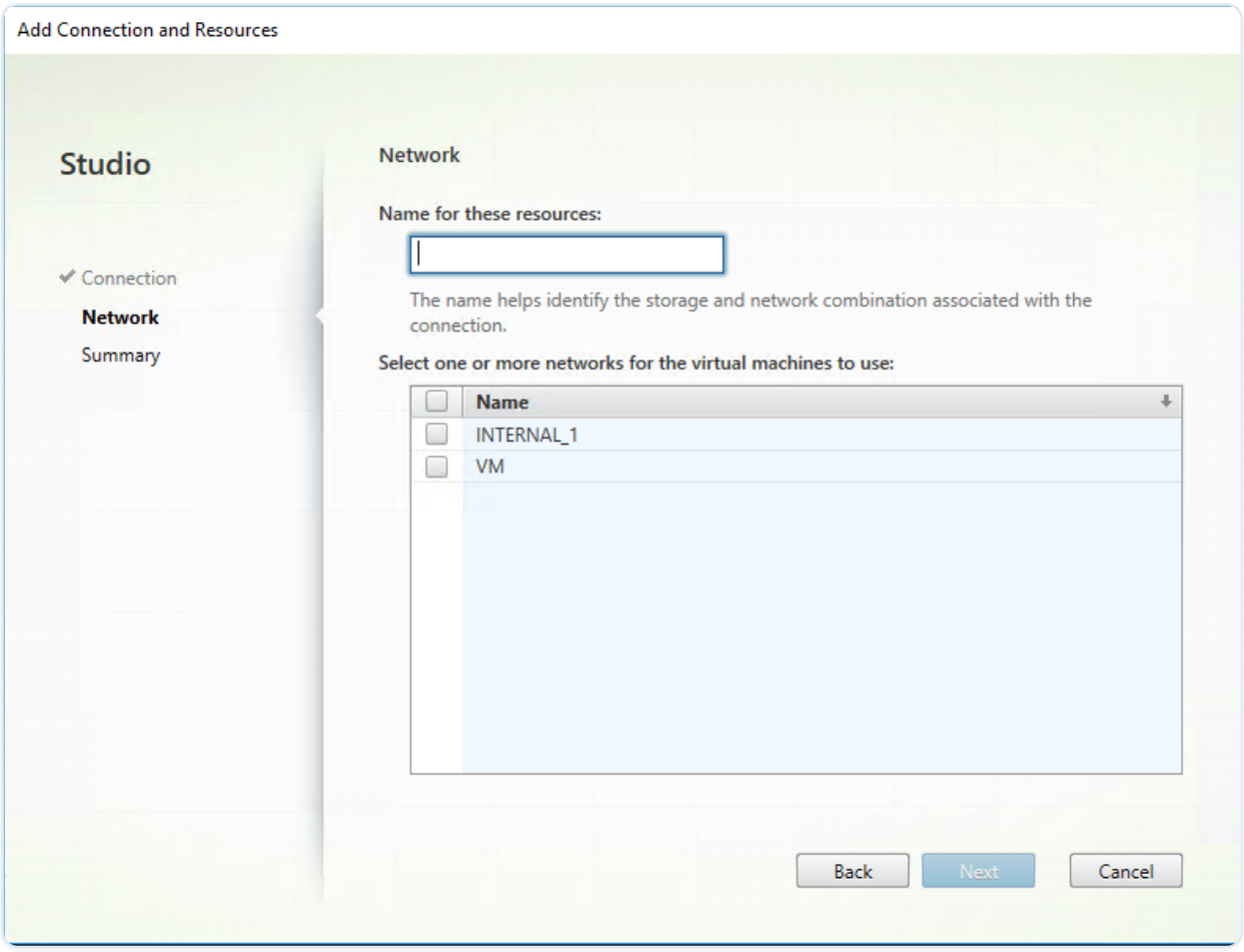
- オンプレミスのDelivery ControllerでCitrix Host、Citrix Broker、およびCitrix Machine Creation Servicesを再起動するか、Citrix Cloud ConnectorでCitrix RemoteHCLServer Serviceを再起動します。

> **ヒント:**
>
> Nutanix AHVプラグインをインストールまたは更新する際は、Citrix Host、Citrix Broker、およびMachine Creation Servicesを停止してから再起動することをお勧めします。

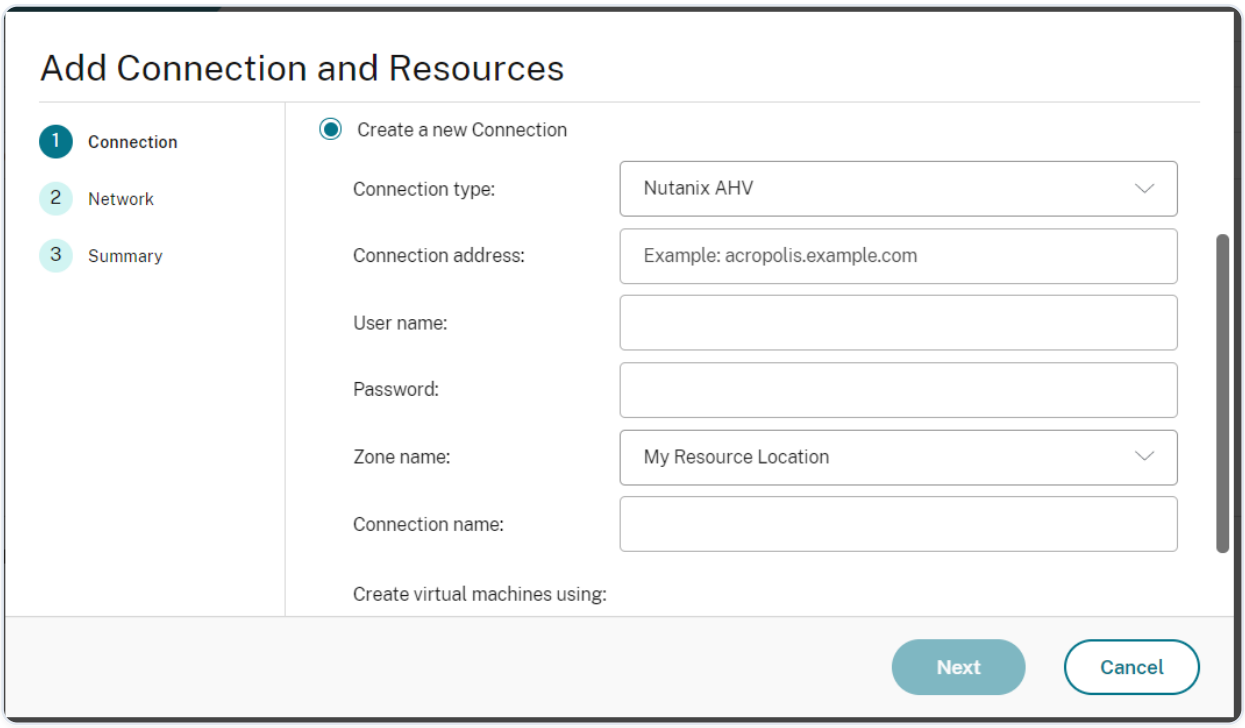
ステップ2：Citrix StudioでNutanixへのホスティング接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続およびリソースの追加**を選択します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続およびリソースの追加**を選択して、Nutanixハイパーバイザーへの接続を作成します。
2. **接続およびリソースの追加**ウィザードで、**接続**ページで接続の種類としてNutanix AHVを選択し、ハイパーバイザーのアドレス、資格情報、および接続名を指定します。**ネットワーク**ページで、ホスティングユニットのネットワークを選択します。

   例：オンプレミスのCitrix Studioの場合：



例：Citrix CloudのWebベースのStudioコンソールの場合：



3. **ネットワーク**ページで、ホスティングユニットのネットワークを選択します。

ステップ3：テンプレートVMでマスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。マスターイメージを準備するには、次の手順を実行します。

ステップ3a：テンプレートVMにLinux VDAパッケージをインストールする

> **注:**

> 実行中のVDAをテンプレートVMとして使用する場合は、この手順をスキップしてください。

> テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Core Runtime 3.1をインストールしてください。詳しくは、「インストール概要」を参照してください。

> Linuxディストリビューションに基づいて、Linux VDAの環境をセットアップするために次のコマンドを実行します。

RHEL/CentOSの場合：

```
sudo yum –y localinstall <PATH>/<Linux VDA RPM>
```

Ubuntu/Debianの場合：

```
sudo dpkg –i  <PATH>/<Linux VDA DEB>



apt-get install -f
```

SUSE 12の場合：

```
sudo zypper –i install <PATH>/<Linux VDA RPM>
```

ステップ3b：ntfs-3gを提供するEPELリポジトリをインストールする

RHEL 8/CentOS 8、RHEL 7/CentOS 7にEPELリポジトリをインストールします。これにより、後でdeploymcs.shを実行したときに、EPELリポジトリで利用可能なntfs-3gパッケージがインストールされます。

ステップ3c：SUSE 12にntfs-3gを手動でインストールする

SUSE 12プラットフォームでは、ntfs-3gを提供するリポジトリはありません。ソースコードをダウンロードし、コンパイルして、ntfs-3gを手動でインストールします。

1. GNU Compiler Collection (GCC) コンパイラシステムとmakeパッケージをインストールします。

```
sudo zypper install gcc

sudo zypper install make
```

2. ntfs-3gパッケージをダウンロードします。
3. ntfs-3gパッケージを解凍します。

```
sudo tar -xvzf ntfs-3g_ntfsprogs-<package version>.tgz
```

4. ntfs-3gパッケージへのパスを入力します。

```
sudo cd ntfs-3g_ntfsprogs-<package version>
```

5. ntfs-3gをインストールします。

```
./configure

make

make install
```

ステップ3d：ランタイム環境をセットアップする

`deploymcs.sh`を実行する前に、以下を実行します。

- `/etc/xdl/mcs/mcs.conf`で変数を変更します。`mcs.conf`構成ファイルは、MCSとLinux VDAを設定するための変数を提供します。以下は変数の一部であり、そのうち`dns`と`AD_INTEGRATION`は設定する必要があります。

  注：変数を複数の値で設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中のVDAの既存の構成を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDA上の同等のファイルと同じになります。ただし、`dns`と`AD_INTEGRATION`の変数は引き続き構成する必要があります。デフォルト値はNで、これはマスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: DNS IPアドレスを設定します。
  - `AD_INTEGRATION`: Winbind、SSSD、またはPBISを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされているディストリビューションを参照してください。
  - `WORKGROUP`: ADで構成されている場合、ワークグループ名（大文字と小文字を区別）を設定します。

- テンプレートマシンで、必要に応じてレジストリ値を書き込むまたは更新するためのコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。この操作により、MCSプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

  `/etc/xdl/mcs/mcs_local_setting.reg`ファイルの各行は、レジストリ値を設定または更新するためのコマンドです。

  たとえば、レジストリ値をそれぞれ書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加できます。

  ```
  create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
  ```

  ```
  update -k
  "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard
  \ClipboardSelection" -v "Flags" -d "0x00000003"
  ```

ステップ3e：Nutanix AHVで、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます

> **注:**
>
> Citrix Virtual Apps and Desktopsで使用するには、Acropolisスナップショット名に`XD_`をプレフィックスとして付ける必要があります。必要に応じて、Acropolisコンソールを使用してスナップショットの名前を変更します。スナップショットの名前を変更した後、**カタログの作成**ウィザードを再起動して、更新されたリストを取得します。

ステップ4：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。Nutanix固有の**コンテナ**ページで、以前にテンプレートVMに指定したコンテナを選択します。**マスターイメージ**ページで、イメージスナップショットを選択します。**仮想マシン**ページで、仮想CPUの数とvCPUあたりのコア数を確認します。

必要に応じて、その他の構成タスクを実行します。詳細については、Studioを使用してマシンカタログを作成するを参照してください。

> **注:**
>
> Delivery Controller™でのマシンカタログ作成プロセスにかなりの時間がかかる場合は、Nutanix Prismに移動し、**Preparation**というプレフィックスが付いたマシンを手動で電源オンにしてください。このアプローチは、作成プロセスを続行するのに役立ちます。

ステップ5：デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンのコレクションです。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳細については、デリバリーグループを作成するを参照してください。

MCS を使用して Linux VDA を更新する

MCS を使用して Linux VDA を更新するには、次の手順を実行します。

1. Linux VDA を現在のリリースに更新する前に、.NET Core Runtime 3.1 をインストールしていることを確認してください。
2. テンプレートマシンで Linux VDA を更新します。

注:

自動ソフトウェア更新をスケジュールするには、Linux VDA self-update 機能も使用できます。この目的を達成するには、テンプレートマシンの etc/xdl/mcs/mcs_local_setting.reg ファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。

```
create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
-t "REG_DWORD" -v "fEnabled" -d "0x00000001" –force



create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
-t "REG_SZ" -v "ScheduledTime" -d "Immediately" –force



create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
-t "REG_SZ" -v "Url" -d "&lt;Your-Azure-Container-Url>" –force



create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
-t "REG_SZ" -v "CaCertificate" -d "&lt;Local-Certificate-Path-of-
PortalAzureCom>" --force
```

RHEL 7 および CentOS 7 の場合:

```
sudo rpm -U XenDesktopVDA-<version>.el7_x.x86_64.rpm
```

RHEL 8 および CentOS 8 の場合:

```
sudo rpm -U XenDesktopVDA-<version>.el8_x.x86_64.rpm
```

SUSE 12 の場合:

```
sudo rpm -U XenDesktopVDA-<version>.sle12_x.x86_64.rpm
```

Ubuntu 16.04 の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu16.04_amd64.deb
```

Ubuntu 18.04 の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu18.04_amd64.deb
```

Ubuntu 20.04 の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu20.04_amd64.deb
```

3. `/etc/xdl/mcs/mcs.conf` と `/etc/xdl/mcs/mcs_local_setting.reg` を編集します。
4. 新しいスナップショットを作成します。
5. Citrix Studio で、新しいスナップショットを選択してマシンカタログを更新します。各マシンが再起動するまで待機します。マシンを手動で再起動しないでください。

マシンアカウントのパスワード更新を自動化する

マシンアカウントのパスワードは、デフォルトで、マシンカタログ作成後30日で期限切れになります。パスワードの期限切れを防ぎ、マシンアカウントのパスワード更新を自動化するには、次の手順を実行します。

1. /opt/Citrix/VDA/sbin/deploymcs.shを実行する前に、/etc/xdl/mcs/mcs.confに次のエントリを追加します。

   `UPDATE_MACHINE_PW="enabled"`

2. /opt/Citrix/VDA/sbin/deploymcs.shを実行した後、/etc/cron.d/mcs_update_password_cronjobを開いて更新時間と頻度を設定します。デフォルト設定では、マシンアカウントのパスワードは毎週日曜日の午前2時30分に更新されます。

マシンアカウントのパスワードが更新されるたびに、Delivery Controller上のチケットキャッシュが無効になり、/var/log/xdl/jproxy.logに次のエラーが表示されることがあります。

`[ERROR] - AgentKerberosServiceAction.Run: GSSException occurred. Error: Failure unspecified at GSS-API level (Mechanism level: Checksum failed)`

エラーを解消するには、チケットキャッシュを定期的にクリアします。すべてのDelivery Controllerまたはドメインコントローラーでキャッシュクリーンアップタスクをスケジュールできます。

MCSで作成されたVMでFASを有効にする

次のディストリビューションで実行されているMCSで作成されたVMでFASを有効にできます。

	ウィンバインド	SSSD	セントリファイ
RHEL 8、CentOS 8	はい	いいえ	いいえ
RHEL 7、CentOS 7	はい	はい	いいえ
ウブントゥ 20.04	はい	いいえ	いいえ
ウブントゥ 18.04	はい	いいえ	いいえ
ウブントゥ 16.04	はい	いいえ	いいえ
デビアン 10.7	はい	いいえ	いいえ
SUSE 12.5	はい	いいえ	いいえ

テンプレートVMでマスターイメージを準備する際にFASを有効にする

1. ルートCA証明書をインポートします。

   ```
   sudo cp root.pem /etc/pki/CA/certs/
   ```

2. ctxfascfg.shを実行します。
3. `/etc/xdl/mcs/mcs.conf`で変数を設定します。
   a. `Use_Existing_Configurations_Of_Current_VDA`の値をYに設定します。
   b. `FAS_LIST`変数をFASサーバーアドレス、またはセミコロンで区切られ二重引用符で囲まれた複数のFASサーバーアドレス（例: `FAS_LIST="<FAS_SERVER_FQDN>;<FAS_SERVER_FQDN>"`）に設定します。
   c. `VDI_MODE`など、必要に応じて他の変数を設定します。

4. スクリプト`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。

MCSで作成されたVMでFASを有効にする

テンプレートマシンで前述のようにFASが有効になっていない場合、MCSで作成された各VMでFASを有効にできます。

MCSで作成されたVMでFASを有効にするには、次の手順を実行します。

1. /etc/xdl/mcs/mcs.confで変数を設定します。
    a. `Use_Existing_Configurations_Of_Current_VDA`の値をYに設定します。
    b. `FAS_LIST`変数をFASサーバーアドレスに設定します。
    c. `VDI_MODE`などの他の必要な変数を設定します。
2. ルートCA証明書をインポートします。

```
sudo cp root.pem /etc/pki/CA/certs/
```

3. `/opt/Citrix/VDA/sbin/ctxfascfg.sh`スクリプトを実行します。

> **注:**
>
> VM起動時にこれらの変数が呼び出されるため、`/etc/xdl/mcs/mcs.conf`ですべての必要な変数を設定する必要があります。

Citrix Provisioning™ を使用してLinux VMを作成する

この記事では、Linuxターゲットデバイスのストリーミングに関する情報を提供します。この機能を使用すると、Citrix Virtual Apps and Desktops™ 環境でLinux仮想デスクトップを直接プロビジョニングできます。

以下のLinuxディストリビューションがサポートされています:

- ウブントゥ 16.04
- ウブントゥ 18.04
- ウブントゥ 20.04
- RHEL 8.3

重要:

Citrix Provisioning実行可能ファイルで利用可能な現在のリリースインストールパッケージを使用することをお勧めします。`Citrix_Provisioning_2106.iso` お使いのLinuxディストリビューションに基づいたパッケージを使用してください。Linuxストリーミングエージェント2106以降を使用するには、Citrix Provisioning Server 2106以降が必要です。

詳細については、Citrix ProvisioningドキュメントのStreaming Linux target devicesを参照してください。

RHEL/CentOS 用 Linux Virtual Delivery Agent をインストールする

手動インストールについてはこの記事の手順に従うか、自動インストールと構成にはeasy installを使用するかを選択できます。Easy installは、手動インストールよりも時間と労力を節約でき、エラーが発生しにくいです。

注:

easy installは新規インストールの場合にのみ使用してください。既存のインストールを更新するためにeasy installを使用しないでください。

ステップ1: VDAインストール用にRHEL 8/CentOS 8、RHEL 7/CentOS 7を準備する

ステップ1a: ネットワーク構成を確認する

続行する前に、ネットワークが正しく接続され、構成されていることをお勧めします。

ステップ1b: ホスト名を設定する

マシンのホスト名が正しく報告されるように、**/etc/hostname** ファイルをマシンのホスト名のみを含むように変更します。

`hostname`

ステップ1c: ホスト名にループバックアドレスを割り当てる

マシンのDNSドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されるように、`/etc/hosts`ファイルの次の行を、最初の2つのエントリとしてFQDNとホスト名を含むように変更します。

127.0.0.1 **hostname-fqdn hostname** localhost localhost.localdomain localhost4 localhost4.localdomain4

例:

127.0.0.1 vda01.example.com vda01 localhost localhost.localdomain localhost4 localhost4.localdomain4

ファイル内の他のエントリから、**hostname-fqdn** または **hostname** への他の参照をすべて削除します。

> **注:**

> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。したがって、ホスト名は15文字を超えてはなりません。

> **ヒント:**

> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダースコア (_)、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは、Delivery Controllerのホスト名にも適用されます。

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、完全修飾ドメイン名 (FQDN) は返しません。

FQDN が正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンの FQDN を返します。

ステップ 1e: 名前解決とサービス到達可能性を確認する

FQDN を解決し、ドメインコントローラーと Delivery Controller™ に ping を実行できることを確認します。

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

FQDN を解決できない場合、またはこれらのマシンのいずれかに ping を実行できない場合は、続行する前に手順を確認してください。

ステップ 1f: クロック同期を構成する

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することは非常に重要です。Linux VDA を仮想マシンとしてホストすると、クロックのずれの問題が発生する可能性があります。このため、リモート時刻サービスとの時刻同期が推奨されます。

RHEL 8/RHEL 7 のデフォルト環境では、クロック同期に Chrony デーモン (`chronyd`) を使用します。

Chrony サービスを構成する

root ユーザーとして、**/etc/chrony.conf** を編集し、各リモートタイムサーバーのサーバーエントリを追加します。

```
server peer1-fqdn-or-ip-address iburst



server peer2-fqdn-or-ip-address iburst
```

一般的な展開では、パブリック NTP プールサーバーから直接ではなく、ローカルドメインコントローラーから時刻を同期します。ドメイン内の各 Active Directory ドメインコントローラーのサーバーエントリを追加します。

ループバック IP アドレス、localhost、およびパブリックサーバーの ***.pool.ntp.org** エントリを含む、リストされている他のすべてのサーバーエントリを削除します。

変更を保存し、Chrony デーモンを再起動します。

```
sudo /sbin/service chronyd restart
```

ステップ 1g: OpenJDK 11 をインストールする

Linux VDA には OpenJDK 11 が必要です。Linux VDA をインストールすると、ランタイム環境は依存関係として自動的にインストールされます。

正しいバージョンであることを確認します。

```
sudo yum info java-11-openjdk
```

プレパッケージされた OpenJDK は以前のバージョンである可能性があります。OpenJDK 11 に更新します。

```
sudo yum -y update java-11-openjdk
```

ステップ 1h: PostgreSQL をインストールする

Linux VDA には、RHEL 8 では PostgreSQL 10.5 以降、RHEL 7 では PostgreSQL 9.2 以降が必要です。

次のパッケージをインストールします。

```
sudo yum -y install postgresql-server



sudo yum -y install postgresql-jdbc
```

データベースを初期化し、マシン起動時にサービスが開始されるようにするには、次のインストール後の手順が必要です。このアクションにより、**/var/lib/pgsql/data** の下にデータベースファイルが作成されます。コマンドは PostgreSQL 10 と 9 で異なります。

```
sudo postgresql-setup initdb
```

ステップ 1i: PostgreSQL を起動する

マシン起動時にサービスを開始し、すぐにサービスを開始します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

次を使用して PostgreSQL のバージョンを確認します。

```
psql --version
```

(RHEL 7 のみ) **psql** コマンドラインユーティリティを使用して、データディレクトリが設定されていることを確認します。

```
sudo -u postgres psql -c 'show data_directory'
```

ステップ 2: ハイパーバイザーを準備する

サポートされているハイパーバイザー上で Linux VDA を仮想マシンとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに応じて、次の変更を行います。Linux マシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™ での時刻同期の修正

Citrix Hypervisor の時刻同期機能が有効になっている場合、準仮想化 Linux VM 内では、NTP と Citrix Hypervisor の両方がシステムクロックを管理しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは変更は不要です。

一部の Linux ディストリビューションでは、Citrix VM Tools がインストールされた準仮想化 Linux カーネルを実行している場合、Linux VM 内から Citrix Hypervisor の時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは 0 または 1 を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合は、ファイルに 1 を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を永続化し、再起動後も維持するには、**/etc/sysctl.conf** ファイルを編集して次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -


```

```
cat /proc/sys/xen/independent_wallclock
```

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux Integration Services がインストールされている Linux VM は、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するには、NTP サービスと並行してこの機能を有効にする必要があります。

管理オペレーティングシステムから:

1. Hyper-V マネージャーコンソールを開きます。
2. Linux VMの設定については、**Integration Services**を選択します。
3. **Time synchronization**が選択されていることを確認します。

注:

このアプローチは、NTPとの競合を避けるためにホストの時刻同期が無効になっているVMwareおよびCitrix Hypervisorとは異なります。Hyper-Vの時刻同期は、NTPの時刻同期と共存し、補完することができます。

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内で、NTPとハイパーバイザーの両方がシステムクロックを同期しようとする問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合：

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **Virtual Machine Properties**ダイアログで、**Options**タブを開きます。
4. **VMware Tools**を選択します。
5. **Advanced**ボックスで、**Synchronize guest time with host**のチェックを外します。

手順3：Linux仮想マシン（VM）をWindowsドメインに追加する

Linux VDAは、LinuxマシンをActive Directory（AD）ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ・ウィンバインド
- クエスト認証サービス
- セントリファイ ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新します。

RHEL 8/CentOS 8 の場合:

```
sudo yum -y install samba-winbind samba-winbind-clients krb5-workstation
oddjob-mkhomedir realmd authselect
```

RHEL 7/CentOS 7 の場合:

```
sudo yum -y install samba-winbind samba-winbind-clients krb5-workstation
authconfig oddjob-mkhomedir
```

マシンの起動時に Winbind デーモンが開始するように有効にする

Winbind デーモンは、マシンの起動時に開始するように構成する必要があります。

```
sudo /sbin/chkconfig winbind on
```

Winbind 認証の構成

Winbind を使用して、Kerberos 認証用にマシンを構成します。

1. 次のコマンドを実行します。

   RHEL 8 の場合:

   ```
   sudo authselect select winbind with-mkhomedir --force
   ```

   RHEL 7 の場合:

   ```
   sudo authconfig --disablecache --disablesssd --disablesssdauth --
   enablewinbind --enablewinbindauth --disablewinbindoffline --
   smbsecurity=ads --smbworkgroup=domain --smbrealm=REALM --
   krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --
   winbindtemplateshell=/bin/bash --enablemkhomedir --updateall
   ```

   REALM は大文字のKerberosレルム名で、**domain** はドメインのNetBIOS名です。

   KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、前のコマンドに次の2つのオプションを追加します。

   `--enablekrb5kdcdns --enablekrb5realmdns`

   `authconfig` コマンドから返される、`winbind` サービスの起動に失敗したというエラーは無視してください。これらのエラーは、マシンがまだドメインに参加していない状態で `authconfig` が `winbind` サービスを起動しようとしたときに発生する可能性があります。

2. **/etc/samba/smb.conf** を開き、\[Global\] セクションの下に、ただし `authconfig` ツールによって生成されたセクションの後に、次のエントリを追加します。

   `kerberos method = secrets and keytab`

   `winbind refresh tickets = true` `winbind offline logon = no`

3. (RHEL 8のみ) **/etc/krb5.conf** を開き、`[libdefaults]`、`[realms]`、および `[domain_realm]` セクションの下にエントリを追加します。

   `[libdefaults]` セクションの下に:

   `default_ccache_name = FILE:/tmp/krb5cc_%{uid}`

   `default_realm = REALM` `dns_lookup_kdc = true`

`[realms]` セクションの下に:

`REALM = {`

`kdc = fqdn-of-domain-controller` `}`

`[domain_realm]` セクションの下に:

`realm = REALM`

`.realm = REALM`

Linux VDAは、Delivery Controllerで認証および登録するために、システムキータブファイル /etc/krb5.keytab を必要とします。以前のKerberosメソッド設定により、マシンが最初にドメインに参加したときにWinbindがシステムキータブファイルを作成するよう強制されます。

Windowsドメインに参加する

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

RHEL 8:

```
sudo realm join -U user --client-software=winbind REALM
```

RHEL 7:

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

REALM は大文字のKerberosレルム名で、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

WinbindのPAMを構成する

デフォルトでは、Winbind PAMモジュール (pam_winbind) の設定では、Kerberosチケットキャッシュとホームディレクトリの作成は有効になっていません。**/etc/security/pam_winbind.conf** を開き、\[Global\] セクションの下に次のエントリを追加または変更します。

`krb5_auth = yes`

`krb5_ccache_type = FILE` `mkhomedir = yes`

各設定の先頭にあるセミコロンが削除されていることを確認してください。これらの変更を適用するには、Winbindデーモンを再起動する必要があります。

```
sudo /sbin/service winbind restart
```

ヒント:

`winbind` デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

/etc/krb5.conf を開き、\[libdefaults\] セクションの下にある次の設定をKEYRINGからFILEタイプに変更します。

```
default_ccache_name = FILE:/tmp/krb5cc_%{uid}
```

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っていることを要求します。

Sambaの **net ads** コマンドを実行して、マシンがドメインに参加していることを確認します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

Kerberos構成の確認

KerberosがLinux VDAで使用するために正しく構成されていることを確認するには、システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

次を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

ドメインユーザーがドメインで認証できることを確認するには、**wbinfo**ツールを使用します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字を別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、これまで使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

Kerberos資格情報キャッシュ内のチケットが有効で期限切れでないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、ステップ4: Linux VDAをインストールするに進みます。

Quest 認証サービス

ドメインコントローラーでQuestを構成する

Active Directoryドメインコントローラーにクエストソフトウェアをインストールして構成済みであり、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには、次の手順を実行します。

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unix対応**をオンにします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合に同等です。

Linux VDA で Quest を構成する

SELINUX ポリシーの適用を回避する

デフォルトの RHEL 環境では、SELinux が完全に適用されています。この適用により、Quest が使用する Unix ドメインソケット IPC メカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinux を無効にすることです。root ユーザーとして、**/etc/selinux/config** を編集し、**SELinux** の設定を変更します。

```
SELINUX=permissive
```

この変更にはマシンの再起動が必要です。

```
reboot
```

重要:

この設定は慎重に使用してください。無効にした後に SELinux ポリシーの適用を再度有効にすると、root ユーザーや他のローカルユーザーであっても、完全にロックアウトされる可能性があります。

VAS デーモンを構成する

Kerberos チケットの自動更新は有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を 9 時間（32,400 秒）に設定します。これは、デフォルトの 10 時間のチケット有効期間よりも 1 時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。

PAM と NSS を構成する

HDX および su、ssh、RDP などの他のサービスを介したドメインユーザーのログオンを有効にするには、次のコマンドを実行して PAM と NSS を手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

Windows ドメインに参加する

Quest の **vastool** コマンドを使用して、Linux マシンを Active Directory ドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターを Active Directory ドメインに参加させる権限を持つ任意のドメインユーザーです。**domain-name** はドメインの DNS 名です（例: example.com）。

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。Questで参加したLinuxマシンがドメイン上にあることを確認するには：

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます：

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します：

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します：

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の確認後、ステップ4：Linux VDAのインストールに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインへの参加

Centrify DirectControl Agentがインストールされている状態で、Centrify `adjoin` コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます：

```
su -

adjoin -w -V -u user domain-name
```

userパラメーターは、コンピューターをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name** は、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。Centrifyで参加したLinuxマシンがドメイン上にあることを確認するには：

```
su -

adinfo
```

「Joined to domain」の値が有効であり、CentrifyDCモードが「connected」を返すことを確認します。モードが開始状態のまま動かない場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all

adinfo –diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2106/installation-overview/redhat.html#step-4-install-the-linux-vda)に進みます。

SSSD

SSSDを使用している場合は、このセクションの指示に従ってください。このセクションには、Linux VDAマシンをWindowsドメインに参加させる手順と、Kerberos認証を構成するためのガイダンスが含まれています。

RHELおよびCentOSでSSSDをセットアップするには、次の手順を実行します。

1. ドメインに参加し、ホストキータブを作成する
2. SSSDをセットアップする
3. SSSDを有効にする
4. Kerberos構成を確認する
5. ユーザー認証を確認する

ドメインに参加し、ホストキータブを作成する

SSSDは、ドメイン参加およびシステムキータブファイルの管理のためのActive Directoryクライアント機能を提供しません。代わりに、`adcli`、`realmd`、または`Samba`を使用できます。

このセクションでは、RHEL 7およびRHEL 8のそれぞれについて、`Samba`と`adcli`のアプローチについて説明します。`realmd`については、RHELまたはCentOSのドキュメントを参照してください。これらの手順は、SSSDを構成する前に実行する必要があります。

- **サンバ (RHEL 7):**

  必要なパッケージをインストールまたは更新します:

  ```
  sudo yum -y install krb5-workstation authconfig oddjob-mkhomedir samba-common-tools
  ```

適切に構成されたファイルを持つLinuxクライアントで:

- /etc/krb5.conf
- /etc/samba/smb.conf:

SambaおよびKerberos認証用にマシンを構成します:

```
sudo authconfig --smbsecurity=ads --smbworkgroup=domain --
smbrealm=REALM --krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --
update
```

ここで、**REALM** は大文字のKerberosレルム名、**domain** はActive Directoryドメインの短いNetBIOS名です。

> **注:**
>
> この記事の設定は、シングルドメイン、シングルフォレストモデルを対象としています。Kerberosは、ADインフラストラクチャに基づいて構成してください。
>
> KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、前のコマンドに次の2つのオプションを追加します:

`--enablekrb5kdcdns --enablekrb5realmdns`

/etc/samba/smb.conf を開き、**\[Global\]** セクションの下に、ただし **authconfig** ツールによって生成されたセクションの後に、次のエントリを追加します:

`kerberos method = secrets and keytab`

`winbind offline logon = no`

Windowsドメインに参加します。ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントがあることを確認してください:

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

REALM は大文字のKerberosレルム名、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

- **アドクリ (RHEL 8):**

  必要なパッケージをインストールまたは更新します:

  ```
  sudo yum -y install samba-common samba-common-tools krb5-workstation
  authconfig oddjob-mkhomedir realmd oddjob authselect
  ```

  SambaおよびKerberos認証用にマシンを構成します:

  ```
  sudo authselect select sssd with-mkhomedir --force
  ```

  /etc/krb5.conf を開き、 および セクションの下にエントリを追加します。

  セクションの下に:

  `REALM = {`

  `kdc = fqdn-of-domain-controller` `}`

  セクションの下に:

  `realm = REALM` `.realm = REALM`

  Windowsドメインに参加します。ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントがあることを確認してください:

  ```
  sudo realm join REALM -U user
  ```

  REALM は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSDをセットアップする

SSSDのセットアップは、次の手順で構成されます:

- `sudo yum -y install sssd` コマンドを実行して、Linux VDAに **sssd-ad** パッケージをインストールします。
- さまざまなファイル（例: sssd.conf）に構成変更を加えます。
- **sssd** サービスを開始します。

RHEL 7の **sssd.conf** 構成例（必要に応じて追加オプションを追加できます）:

RHEL 7のSSSD構成例(/en-us/linux-virtual-delivery-agent/2106/media/example-sssd-conf-for-rhel7.png)

ad.example.com、**server.ad.example.com** を対応する値に置き換えます。詳細については、sssd-ad(5) - Linux man page を参照してください。

(RHEL 8のみ) **/etc/sssd/sssd.conf** を開き、 セクションの下に次のエントリを追加します:

`ad_gpo_access_control = permissive`

`full_name_format = %2$s\%1$s` `fallback_homedir = /home/%d/%u`

`# Kerberos settings` `krb5_ccachedir = /tmp`

`krb5_ccname_template = FILE:%d/krb5cc_%U`

sssd.conf のファイル所有権とアクセス許可を設定します。

`chown root:root /etc/sssd/sssd.conf`

`chmod 0600 /etc/sssd/sssd.conf` `restorecon /etc/sssd/sssd.conf`

SSSD を有効にする

RHEL 8:

SSSD を有効にするには、次のコマンドを実行します。

```
sudo systemctl restart sssd

sudo systemctl enable sssd.service

sudo chkconfig sssd on
```

RHEL 7/CentOS 7:

SSSD を有効にするには `authconfig` を使用します。ホームディレクトリの作成が SELinux と互換性があることを確認するために、**oddjob-mkhomedir** をインストールします。

```
authconfig --enablesssd --enablesssdauth --enablemkhomedir –-update



sudo service sssd start
```

```
sudo chkconfig sssd on
```

Kerberos の構成を確認する

システムの **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos の **kinit** コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 (**$**) は、シェルによる置換を防ぐためにバックスラッシュ (****) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT チケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証を確認する

ログオン形式がサポートされており、NSS が機能していることを確認するには、**getent** コマンドを使用します。

```
sudo getent passwd DOMAIN\\username
```

DOMAIN パラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`

- UPN: `username@domain.com`
- NetBIOSサフィックス形式: `username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost –l DOMAIN\\username



id -u
```

コマンドによって返された**uid**に対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_{uid}
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効であり、期限切れになっていないことを確認します。

```
klist
```

ドメイン参加の検証後、手順4：Linux VDAをインストールするに進みます。

PBIS

必要なPBISパッケージをダウンロードする

たとえば、RHEL 7/CentOS 7の場合：

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/
pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

たとえば、RHEL 8/CentOS 8の場合：

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/
pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行可能にする

たとえば、RHEL 7/CentOS 7の場合：

```
chmod +x pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

たとえば、RHEL 8/CentOS 8の場合：

```
chmod +x pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行する

RHEL 7/CentOS 7 の場合、例:

```
sh pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

RHEL 8/CentOS 8 の場合、例:

```
sh pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

Windowsドメインへの参加

ドメインコントローラーに到達可能であること、およびコンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です:

```
/opt/pbis/bin/domainjoin-cli join domain-name user
```

user は、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

注： Bashをデフォルトシェルとして設定するには、**/opt/pbis/bin/config LoginShellTemplate/bin/bash** コマンドを実行します。

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには:

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します:

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順4：Linux VDAのインストールに進みます。

手順4：Linux VDAのインストール

新規インストールを実行することも、以前の2つのバージョンおよびLTSRリリースから既存のインストールをアップグレードすることもできます。

新規インストールを実行するには

1. (オプション) 古いバージョンをアンインストールする

   以前の2つのバージョンおよびLTSRリリース以外の以前のバージョンをインストールしている場合は、新しいバージョンをインストールする前にアンインストールしてください。

   a. Linux VDAサービスを停止します:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

`ctxvda`および`ctxhdx`サービスを停止する前に、**service ctxmonitorservice stop**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

b. パッケージをアンインストールします:

```
sudo rpm -e XenDesktopVDA
```

注:

コマンドを実行するには、フルパスが必要です。または、**/opt/Citrix/VDA/sbin**と**/opt/Citrix/VDA/bin**をシステムパスに追加することもできます。

2. Linux VDAパッケージをダウンロードする

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

3. Linux VDAをインストールする

- `Yum`を使用してLinux VDAソフトウェアをインストールします:

RHEL 8/CentOS 8の場合:

```
sudo yum install -y XenDesktopVDA-<version>.el8_x.x86_64.rpm
```

RHEL 7/CentOS 7の場合:

```
sudo yum install -y XenDesktopVDA-<version>.el7_x.x86_64.rpm
```

- RPMパッケージマネージャーを使用してLinux VDAソフトウェアをインストールします。インストールする前に、以下の依存関係を解決する必要があります:

  RHEL 8/CentOS 8 の場合:

  ```
  sudo rpm -i XenDesktopVDA-<version>.el8_x.x86_64.rpm
  ```

  RHEL 7/CentOS 7 の場合:

  ```
  sudo rpm -i XenDesktopVDA-<version>.el7_x.x86_64.rpm
  ```

  RHEL 8.2/CentOS 8.2 の RPM 依存関係リスト:

  ```
  postgresql-jdbc >= 42.2.3


  postgresql-server >= 10.6


  java-11-openjdk >= 11


  icoutils >= 0.32


  firewalld  >= 0.8.0


  policycoreutils-python-utils >= 2.9


  python3-policycoreutils >= 2.9
  ```

```
dbus >= 1.12.8

dbus-common  >= 1.12.8

dbus-daemon   >= 1.12.8

dbus-tools  >= 1.12.8

dbus-x11 >= 1.12.8

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12

libXrandr >= 1.5.1

libXtst >= 1.2.3

motif >= 2.3.4
```

```
pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user  >= 2.32.1

xorg-x11-utils  >= 7.5

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.5

cups >= 2.2

foomatic-filters >= 4.0.9

cups-filters >= 1.20.0
```

```
ghostscript  >= 9.25

libxml2 >= 2.9

libmspack >= 0.7
```

RHEL 7/CentOS 7 の RPM 依存関係リスト:

```
postgresql-server >= 9.2

postgresql-jdbc >= 9.2

java-11-openjdk >= 11

ImageMagick >= 6.7.8.9

firewalld >= 0.3.9

policycoreutils-python >= 2.0.83

dbus >= 1.6.12

dbus-x11 >= 1.6.12
```

```
xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

motif >= 2.3.4

pam >= 1.1.8

util-linux >= 2.23.2

bash >= 4.2

findutils >= 4.5

gawk >= 4.0
```

```
sed >= 4.2

cups >= 1.6.0

foomatic-filters >= 4.0.9

openldap >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.6.0

gperftools-libs >= 2.4

rpmlib(FileDigests) <= 4.6.0-1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1
```

```
pmlib(CompressedFileNames) <= 3.0.4-1


rpmlib(PayloadIsXz) <= 5.2-1
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、(/ja-jp/linux-virtual-delivery-agent/2106/system-requirements.html)を参照してください。

注:

RHEL 7.x に Linux VDA をインストールした後、`sudo yum install -y python-websockify x11vnc`コマンドを実行します。これは、セッションシャドウイング機能を使用するために、`python-websockify`と`x11vnc`を手動でインストールすることを目的としています。詳しくは、(/ja-jp/linux-virtual-delivery-agent/2106/configuration/session-shadowing.html)を参照してください。

既存のインストールをアップグレードするには

既存のインストールを、以前の2つのバージョンおよびLTSRリリースからアップグレードできます。

注:

既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、ファイルをバックアップしてください。

- `Yum`を使用してソフトウェアをアップグレードするには:

  RHEL 7/CentOS 7 の場合:

```
sudo yum install -y XenDesktopVDA-<version>.el7_x.x86_64.rpm
```

- RPM パッケージマネージャーを使用してソフトウェアをアップグレードするには:

  RHEL 7/CentOS 7 の場合:

```
sudo rpm -U XenDesktopVDA-<version>.el7_x.x86_64.rpm
```

注:

RHEL 7 を使用している場合は、上記のアップグレードコマンドを実行した後、以下の手順を完了してください:

1. `/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "DotNetRuntimePath" -d "/opt/rh/rh-dotnet31/root/usr/bin/" --force` を実行して、正しい .NET ランタイムパスを設定します。
2. `ctxvda` サービスを再起動します。

重要:

ソフトウェアのアップグレード後、Linux VDA マシンを再起動します。

ステップ 5: NVIDIA GRID ドライバーをインストールする

HDX 3D Pro を有効にするには、ハイパーバイザーと VDA マシンに必要なグラフィックドライバーをインストールするための追加のインストール手順が必要です。

次を設定します。

1. シトリックス ハイパーバイザー
2. ヴイエムウェア イーエスエックス

選択したハイパーバイザーの手順に従ってください。

シトリックス ハイパーバイザー：

この詳細セクションでは、Citrix Hypervisor への NVIDIA GRID ドライバーのインストールと構成について説明します。

ヴイエムウェア イーエスエックス：

このガイドに記載されている情報に従って、VMware ESX 用の NVIDIA GRID ドライバーをインストールおよび構成してください。

VDA マシン:

各 Linux VM ゲストのドライバーをインストールおよび構成するには、次の手順に従ってください。

1. 開始する前に、Linux VM がシャットダウンされていることを確認してください。
2. XenCenter® で、GPU パススルーモードで GPU を VM に追加します。
3. RHEL VMを起動します。

NVIDIA GRIDドライバー用にマシンを準備するには、次のコマンドを実行します。

```
yum install gcc



yum install "kernel-devel-$(uname -r)"



systemctl set-default multi-user.target
```

NVIDIA GRIDドライバーをインストールするには、Red Hat Enterprise Linuxドキュメントの手順に従ってください。

> **注:**

> GPUドライバーのインストール中、各質問に対してデフォルト（「いいえ」）を選択します。

> **重要:**

> GPUパススルーが有効になった後、Linux VMはXenCenter経由でアクセスできなくなります。SSHを使用して接続してください。

```
nvidia-smi

+-----------------------------------------------------+
| NVIDIA-SMI 352.70     Driver Version: 352.70        |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla M60           Off  | 0000:00:05.0     Off |                  Off |
| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

高解像度とマルチモニター機能を利用するには、有効なNVIDIAライセンスが必要です。ライセンスを申請するには、「GRID Licensing Guide.pdf - DU-07757-001 September 2015」の製品ドキュメントに従ってください。

手順6：Linux VDAを構成する

パッケージのインストール後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションを環境変数で指定します。必要な変数がすべて存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）をスペースで区切ったリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。デフォルトではポート80です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux Virtual Desktopサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktopサービスでは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux Virtual Desktopのシステムファイアウォールで、必要なポート（デフォルトではポート80および1494）を自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \| 5** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ Winbind
  - 2 – クエスト認証サービス
  - 3 – セントリファイ DirectControl
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。

- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例: ad1.mycompany.com:389。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP を照会します。検索パフォーマンスを向上させるために、検索ベース (例: OU=VDI,DC=mycompany,DC=com) を指定できます。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロン区切りの FAS サーバーのリストを提供できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするための .NET Core Runtime 3.1 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** – セッションで使用する GNOME または MATE デスクトップ環境を指定します。変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

> **注:**
>
> 以下の手順を実行して、ターゲットセッションユーザーのデスクトップ環境を変更することもできます。

a. VDA 上の **$HOME/<username\>** ディレクトリの下に `.xsession` ファイルを作成します。
b. ディストリビューションに基づいてデスクトップ環境を指定するために、`.xsession` ファイルを編集します。

CentOS、Ubuntu、および Debian 上の MATE デスクトップの場合

MSESSION="$(type -p mate-session)"

\> if ; then

```
exec mate-session
```

\> fi

```
**CentOS 上の GNOME デスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
   export GNOME_SHELL_SESSION_MODE=classic
```

\> exec gnome-session --session=gnome-classic

```
   fi

**Ubuntu および Debian 上のGNOMEデスクトップの場合**
```

```
    GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
    exec gnome-session
```

\> fi

a. ターゲットセッションユーザーと 700 ファイル権限を共有します。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDA の構成が完了したときに Linux VDA サービスが開始されるかどうか。デフォルトで Y に設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutをリッスンするためのソケットポートです。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポートです。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N



export CTX_XDL_DDC_LIST='list-ddc-fqdns'



export CTX_XDL_VDA_PORT=port-number



export CTX_XDL_REGISTER_SERVICE=Y|N
```

```
export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

export CTX_XDL_SITE_NAME=dns-site-name | '<none>'

export CTX_XDL_LDAP_LIST='list-ldap-servers' | '<none>'

export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'

export CTX_XDL_FAS_LIST='list-fas-servers' | '<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number
```

```
export CTX_XDL_TELEMETRY_PORT=port-number


export CTX_XDL_START_SERVICE=Y|N


sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
```

sudoコマンドを実行するときは、既存の環境変数を新しく作成されるシェルに渡すために、**-E**オプションを入力します。上記のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \


CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \


CTX_XDL_HDX_3D_PRO=Y|N \
```

```
CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \
```

```
/opt/Citrix/VDA/sbin/ctxsetup.sh
```

構成の変更を削除する

場合によっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成の変更を削除する必要があるかもしれません。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル**/tmp/xdl.configure.log**に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

手順7：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成した後、次のコマンドを実行してLinux VDAを制御できます。

Linux VDA を開始する:

Linux VDA サービスを開始するには:

```
sudo /sbin/service ctxhdx start
```

```
sudo /sbin/service ctxvda start
```

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

`ctxvda` と `ctxhdx` サービスを停止する前に、モニターサービスデーモンを停止するために `service ctxmonitorservice stop` コマンドを実行します。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDA を再起動する:

Linux VDA サービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDA のステータスを確認する:

Linux VDA サービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ 8: Citrix Virtual Apps または Citrix Virtual Desktops™ でマシンカタログを作成する

マシンカタログを作成し、Linux VDA マシンを追加するプロセスは、従来の Windows VDA のアプローチと同様です。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDA マシンを含むマシンカタログを作成する場合、Windows VDA マシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します:
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッション OS** オプション。
  - VDI 専用デスクトップ配信モデルの場合は、**シングルセッション OS** オプション。
- 同じマシンカタログにLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト型共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに単一ユーザーの配信モデルが意味されます。

ヒント:

マシンをActive Directoryドメインから削除して再参加させる場合、マシンをマシンカタログから削除して再度追加する必要があります。

ステップ9：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

> **重要:**
>
> アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。

> マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2106」を参照してください。

SUSE 用 Linux Virtual Delivery Agent のインストール

手動インストールについては、この記事の手順に従うか、自動インストールと構成には (/ja-jp/linux-virtual-delivery-agent/2106/installation-overview/easy-install.html) を使用できます。Easy install は、手動インストールよりも時間と労力を節約でき、エラーが発生しにくいです。

> **注:**
>
> Easy install は新規インストールにのみ使用してください。既存のインストールを更新するために Easy install を使用しないでください。

ステップ 1: インストールの準備

ステップ 1a: YaST ツールの起動

SUSE Linux Enterprise YaST ツールは、オペレーティングシステムのあらゆる側面を構成するために使用されます。

テキストベースの YaST ツールを起動するには:

```
su -



yast
```

UI ベースの YaST ツールを起動するには:

```
su -



yast2 &
```

ステップ 1b: ネットワークの構成

以下のセクションでは、Linux VDA で使用されるさまざまなネットワーク設定とサービスの構成に関する情報を提供します。ネットワークの構成は、Network Manager などの他の方法ではなく、YaST ツールを介して実行されます。これらの手順は、UI ベースの YaST ツールを使用することを前提としています。テキストベースの YaST ツールも使用できますが、ここでは説明されていない異なるナビゲーション方法があります。

ホスト名と DNS の構成

1. YaST ネットワーク設定を開きます。
2. SLED 12 のみ: **Global Options** タブで、**Network Setup Method** を **Wicked Service** に変更します。
3. **Hostname/DNS** タブを開きます。
4. **Change hostname via DHCP** のチェックを外します。
5. **ループバックIPにホスト名を割り当てる**をオンにします。
6. ネットワーク設定を反映するように、以下を編集します。

- ホスト名 – マシンのDNSホスト名を追加します。
- ドメイン名 – マシンのDNSドメイン名を追加します。
- ネームサーバー – DNSサーバーのIPアドレスを追加します。通常、これはADドメインコントローラーのIPアドレスです。
- ドメイン検索リスト – DNSドメイン名を追加します。

> **注:**
>
> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。そのため、ホスト名は15文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン（-）文字のみを使用してください。アンダースコア（_）、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。この規則は、Delivery Controllerのホスト名にも適用されます。

マルチキャストDNSを無効にする

SLEDのみで、デフォルト設定ではマルチキャストDNS（mDNS）が有効になっており、名前解決の結果が不安定になる可能性があります。mDNSはSLESではデフォルトで有効になっていないため、操作は不要です。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、以下の行を変更します。

```
hosts: files mdns_minimal [NOTFOUND=return] dns
```

変更後：

```
hosts: files dns
```

ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、完全修飾ドメイン名（FQDN）は返しません。

FQDN が正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンの FQDN を返します。

名前解決とサービスの到達可能性を確認する

FQDN を解決し、ドメインコントローラーと Delivery Controller™ に ping できることを確認します。

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

FQDN を解決できない場合、またはこれらのマシンのいずれかに ping できない場合は、続行する前に手順を確認してください。

ステップ 1c: NTP サービスを構成する

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することが重要です。Linux VDA を仮想マシンとしてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモート NTP サービスを使用して時刻を維持することが推奨されます。デフォルトの NTP 設定にいくつかの変更が必要になる場合があります。

1. YaST NTP Configuration を開き、**General Settings** タブを選択します。
2. Start NTP Daemon セクションで、**Now and on Boot** をチェックします。
3. 存在する場合は、**Undisciplined Local Clock (LOCAL)** 項目を選択し、**Delete** をクリックします。
4. **Add** をクリックして、NTP サーバーのエントリを追加します。

5. **Server Type** を選択し、**Next** をクリックします。
6. Address フィールドに NTP サーバーの DNS 名を入力します。このサービスは通常、Active Directory ドメインコントローラーでホストされます。
7. Options フィールドは変更しないでください。
8. **Test** をクリックして、NTP サービスに到達可能であることを確認します。
9. 変更を保存するには、一連のウィンドウで**OK**をクリックします。

> **注:**
>
> SLES 12の実装では、AppArmorポリシーに関する既知のSUSEの問題により、NTPデーモンが起動に失敗する場合があります。詳細については、解決策を参照してください。

手順1d：Linux VDAの依存パッケージをインストールする

SUSE Linux Enterprise用Linux VDAソフトウェアは、以下のパッケージに依存しています。

- PostgreSQL10サーバー 10.12 以降
- OpenJDK 11
- OpenMotif ランタイム環境 2.3.1 以降
- Cups 1.6.0 以降
- Foomatic フィルター 3.0.0 以降
- ImageMagick 6.8 以降

リポジトリを追加する

PostgreSQLやImageMagickなどの一部の必須パッケージは、SUSE Linux Enterprise Software Development Kit (SDK) から入手できます。パッケージを入手するには、YaSTを使用してSDKリポジトリを追加するか、SDKイメージファイルをダウンロードして、以下のコマンドを使用してローカルにマウントします。

```
sudo mkdir -p /mnt/sdk



sudo mount -t iso9660 path-to-iso/SLE-12-SP5-SDK-DVD-x86_64-GM-DVD1.iso /
mnt/sdk
```

```
sudo zypper ar -f /mnt/sdk sdk
```

Kerberosクライアントをインストールする

Linux VDAとDelivery Controller間の相互認証のためにKerberosクライアントをインストールします。

```
sudo zypper install krb5-client
```

Kerberosクライアントの構成は、使用されるActive Directory統合アプローチによって異なります。以下の説明を参照してください。

OpenJDK 11 をインストールする

Linux VDA には OpenJDK 11 が必要です。

OpenJDK 11 をインストールするには、次のコマンドを実行します。

```
```

sudo zypper install java-11-openjdk

```
```

PostgreSQL をインストールする

SLED/SLES 12 で、パッケージをインストールします。

```
sudo zypper install postgresql-init


sudo zypper install postgresql10-server


sudo zypper install postgresql-jdbc
```

データベースサービスを初期化し、マシンの起動時に PostgreSQL が開始されるようにするには、インストール後の手順が必要です。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

データベースファイルは /var/lib/pgsql/data にあります。

リポジトリを削除する

依存パッケージがインストールされたら、次のコマンドを実行して、以前に設定した SDK リポジトリとマウントされたメディアを削除します。

```
sudo zypper rr sdk



sudo umount /mnt/sdk



sudo rmdir /mnt/sdk
```

ステップ 2：ハイパーバイザー用に Linux VM を準備する

サポートされているハイパーバイザー上で Linux VDA を仮想マシンとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに応じて、次の変更を行います。Linux マシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™ での時刻同期を修正する

Citrix Hypervisor の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で、NTP と Citrix Hypervisor の両方がシステムクロックを管理しようとする問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックは NTP と同期させる必要があります。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは変更は不要です。

一部の Linux ディストリビューションでは、Citrix VM Tools がインストールされた準仮想化 Linux カーネルを実行している場合、Linux VM 内から Citrix Hypervisor の時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは 0 または 1 を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は必要ありません。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合は、ファイルに **1** を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を永続化し、再起動後も維持するには、**/etc/sysctl.conf** ファイルを編集して次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
reboot
```

再起動後、設定が正しいことを確認します。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux Integration Services がインストールされている Linux VM は、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、この機能を NTP サービスと併用して有効にしてください。

管理オペレーティングシステムから：

1. Hyper-V マネージャーコンソールを開きます。
2. Linux VM の設定で、**統合サービス** を選択します。
3. **時刻同期** が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTP との競合を避けるためにホスト時刻同期が無効になっている VMware や Citrix Hypervisor とは異なります。Hyper-V の時刻同期は、NTP 時刻同期と共存し、それを補完することができます。

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、準仮想化された各 Linux VM 内で、NTP とハイパーバイザーの両方がシステムクロックを同期しようとする問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックを NTP と同期させます。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools がインストールされた準仮想化 Linux カーネルを実行している場合：

1. vSphere Client を開きます。
2. Linux VM の設定を編集します。
3. **仮想マシンプロパティ**ダイアログで、**オプション**タブを開きます。
4. **VMware Tools** を選択します。
5. **詳細**ボックスで、**ゲストの時刻をホストと同期**のチェックを外します。

ステップ 3：Linux 仮想マシン（VM）を Windows ドメインに追加する

Linux VDA は、Linux マシンを Active Directory (AD) ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ Winbind
- クエスト認証サービス
- セントリファイ DirectControl
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ Winbind

Windows ドメインへの参加

ドメインコントローラーに到達可能であり、マシンをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

1. YaST Windows ドメインメンバーシップを開きます。
2. 次の変更を行います。
   - **ドメインまたはワークグループ**を Active Directory ドメインの名前またはドメインコントローラーの IP アドレスに設定します。ドメイン名が大文字であることを確認してください。
   - **Linux 認証にも SMB 情報を使用**をチェックします。
   - **ログイン時にホームディレクトリを作成**をチェックします。
   - **SSH のシングルサインオン**をチェックします。
   - **オフライン認証**がチェックされていないことを確認してください。このオプションは Linux VDA と互換性がありません。
3. **OK**をクリックします。一部のパッケージのインストールを求められた場合は、**インストール**をクリックします。
4. ドメインコントローラーが見つかった場合、ドメインに参加するかどうかを尋ねられます。**はい**をクリックします。
5. プロンプトが表示されたら、コンピューターをドメインに追加する権限を持つドメインユーザーの資格情報を入力し、**OK**をクリックします。
6. 成功を示すメッセージが表示されます。
7. sambaおよびkrb5パッケージのインストールを求められた場合は、 をクリックします。

YaSTは、これらの変更に一部のサービスまたはマシンの再起動が必要であることを示している場合があります。マシンを再起動することをお勧めします。

```
su -



reboot
```

SUSE 12のみ: Kerberos資格情報キャッシュ名のパッチ適用

SUSE 12では、デフォルトのKerberos資格情報キャッシュ名の指定が、通常の**FILE:/tmp/krb5cc_%{uid}からDIR:/run/user/%{uid}/krb5cc**に変更されました。この新しいDIRキャッシュ方法はLinux VDAと互換性がないため、手動で変更する必要があります。rootユーザーとして**/etc/krb5.conf**を編集し、設定されていない場合は**\[libdefaults\]**セクションに次の設定を追加します。

```
default_ccache_name = FILE:/tmp/krb5cc_%{uid}
```

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。

Sambaの**net ads**コマンドを実行して、マシンがドメインに参加していることを確認します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

Kerberos構成の確認

KerberosがLinux VDAで使用するために正しく構成されていることを確認するには、システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェル置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

マシンアカウントの詳細を確認するには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

`wbinfo`ツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインは、Kerberosレルム名ではなく、ADドメイン名です。bashシェルでは、バックスラッシュ (\\) 文字を別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -uコマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効で期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンして実行できます。ドメイン参加の確認後、手順4：Linux VDAのインストールに進みます。

Quest認証サービス

ドメインコントローラーでのQuestの構成

Active Directoryドメインコントローラーにquestソフトウェアをインストールして構成し、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されていることを前提とします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unix Account**タブを選択します。
3. 「**Unix-enabled**」をチェックします。
4. 「**Primary GID Number**」を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

Linux VDAでQuestを構成する

VASデーモンを構成する

Kerberosチケットの自動更新を有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間のチケット有効期間より1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。

PAMとNSSを構成する

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、次のコマンドを実行してPAMとNSSを手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

Windowsドメインに参加する

Quest `vastool` コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

user は、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。Questに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます。

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証を確認する

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -u コマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、手順4：Linux VDAをインストールするに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインに参加する

Centrify DirectControl Agentがインストールされている状態で、Centrifyの **adjoin** コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su –

adjoin -w -V -u user domain-name
```

user は、コンピューターをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name** は、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップを確認する

Delivery Controllerでは、すべてのVDAマシン（Windows VDAおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
su –



adinfo
```

Joined to domain の値が有効であり、**CentrifyDC** の **mode** が **connected** を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2106/installation-overview/suse.html#step-4-install-the-linux-vda)に進みます。

SSSD

SUSEでSSSDを使用している場合は、このセクションの手順に従ってください。このセクションには、Linux VDAマシンをWindowsドメインに参加させるための手順と、Kerberos認証を構成するためのガイダンスが含まれています。

SUSEでSSSDをセットアップするには、次の手順を実行します。

1. ドメインに参加し、ホストキータブを作成する
2. SSSD用にPAMを構成する
3. SSSDをセットアップする
4. SSSDを有効にする
5. ドメインメンバーシップを確認する
6. Kerberos構成を確認する
7. ユーザー認証を確認する

ドメインに参加し、ホストキータブを作成する

SSSDは、ドメイン参加およびシステムキータブファイルの管理のためのActive Directoryクライアント機能を提供しません。代わりに`Samba`アプローチを使用できます。SSSDを構成する前に、次の手順を完了してください。

1. Name Service Cache Daemon (NSCD) デーモンを停止して無効にします。

   ```
   sudo systemctl stop nscd



   sudo systemctl disable nscd
   ```

2. 必要なパッケージをインストールまたは更新します。

```
sudo zypper install krb5-client



sudo zypper install samba-client
```

3. ルートユーザーとして /etc/krb5.conf ファイルを編集し、`kinit` ユーティリティがターゲットドメインと通信できるようにします。、、および セクションの下に次のエントリを追加します。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]



    dns_canonicalize_hostname = false



    rdns = false



    default_realm = REALM



    forwardable = true



[realms]



    REALM = {
```

```
        kdc = fqdn-of-domain-controller

        default_domain = realm

        admin_server =     fqdn-of-domain-controller
    }
[domain_realm]

    .realm = REALM

    realm = REALM
```

realm は、example.com のような Kerberos レルム名です。**REALM** は、EXAMPLE.COM のような大文字の Kerberos レルム名です。**fqdn-of-domain-controller** は、ドメインコントローラーの FQDN です。

4. ルートユーザーとして /etc/samba/smb.conf を編集し、**net** ユーティリティがターゲットドメインと通信できるようにします。 セクションの下に次のエントリを追加します。

```
[global]
    workgroup = domain

    realm = REALM

    security = ADS
```

```
    kerberos method = secrets and keytab

    client signing = yes

    client use spnego = yes
```

domain は、EXAMPLE のような Active Directory ドメインの短い NetBIOS 名です。

5. ユーザーとグループを解決する際に SSSD を参照するように、/etc/nsswitch.conf ファイルの **passwd** および **group** エントリを変更します。

```
passwd: compat sss

group: compat sss
```

6. Windowsドメインに参加します。ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントがあることを確認してください。

```
sudo realm join REALM -U user
```

user は、コンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD 用に PAM を構成する

SSSD 用に PAM を構成する前に、必要なパッケージをインストールまたは更新します。

```
sudo zypper install sssd sssd-ad
```

SSSD を介したユーザー認証のために PAM モジュールを構成し、ユーザーログオン用のホームディレクトリを作成します。

```
sudo pam-config --add  --sss

sudo pam-config --add --mkhomedir
```

SSSD をセットアップする

1. ルートユーザーとして /etc/sssd/sssd.conf を編集し、SSSD デーモンがターゲットドメインと通信できるようにします。sssd.conf の構成例（必要に応じて追加オプションを追加できます）：

```
[sssd]

    config_file_version = 2

    services = nss,pam

    domains = domain-dns-name



[domain/domain-dns-name]

    id_provider = ad

    auth_provider = ad

    access_provider = ad

    ad_domain = domain-dns-name

    ad_server = fqdn-of-domain-controller

    ldap_id_mapping = true

    ldap_schema = ad



# Kerberos settings

    krb5_ccachedir = /tmp
```

```
    krb5_ccname_template = FILE:%d/krb5cc_%U



# Comment out if the users have the shell and home dir set on the AD
side



    fallback_homedir = /home/%d/%u

    default_shell = /bin/bash



# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available



# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM



    ad_gpo_access_control = permissive
```

domain-dns-name は、example.com のような DNS ドメイン名です。

> **注:**
>
> **ldap_id_mapping** は、SSSD 自体が Windows SID を Unix UID にマッピングするように true に設定されています。そうでない場合、Active Directory は POSIX 拡張機能を提供できる必要があります。**ad_gpo_access_control** は、Linux セッションでの無効なログオンエラーを防ぐために **permissive** に設定されています。sssd.conf および sssd-ad の man ページを参照してください。

2. sssd.conf のファイル所有権とアクセス許可を設定します。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSD を有効にする

システム起動時に SSSD デーモンを有効にして開始するには、次のコマンドを実行します。

```
sudo systemctl enable sssd

sudo systemctl start sssd
```

ドメインメンバーシップを確認する

1. Samba の net ads コマンドを実行して、マシンがドメインに参加していることを確認します。

   ```
   sudo net ads testjoin
   ```

2. 追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

   ```
   sudo net ads info
   ```

Kerberos の構成を確認する

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システム keytab ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 (**$**) は、シェル置換を防ぐためにバックスラッシュ (****) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT チケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証を確認する

SSSD は、デーモンで直接認証をテストするためのコマンドラインツールを提供しておらず、PAM を介してのみ実行できます。

SSSD PAM モジュールが正しく構成されていることを確認するには、以前に使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username



id -u



klist



exit
```

`klist` コマンドによって返された Kerberos チケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

ルートユーザーとして、以前の `id -u` コマンドによって返されたuidに対応するチケットキャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、ステップ4: Linux VDAをインストールする に進みます。

PBIS

必要なPBISパッケージをダウンロードする

例:

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/
pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行可能にする

例:

```
chmod +x pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行する

例:

```
sh pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

Windowsドメインに参加する

ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
/opt/pbis/bin/domainjoin-cli join domain-name user
```

user は、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

注： Bashをデフォルトシェルとして設定するには、**/opt/pbis/bin/config LoginShellTemplate/bin/bash** コマンドを実行します。

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証を確認する

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

「**id -u**」コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順4：Linux VDAのインストールに進みます。

手順4：Linux VDAのインストール

手順4a：古いバージョンのアンインストール

以前の2つのバージョンおよびLTSRリリース以外の以前のバージョンをインストールしている場合は、新しいバージョンをインストールする前にアンインストールしてください。

1. Linux VDAサービスを停止します。

   ```
   sudo /sbin/service ctxvda stop



   sudo /sbin/service ctxhdx stop
   ```

> **注:**
>
> `ctxvda`および`ctxhdx`サービスを停止する前に、`service ctxmonitorservice stop`コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

2. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

> **重要:**
>
> 最新の2つのバージョンからのアップグレードがサポートされています。

> **注:**
>
> インストールコンポーネントは **/opt/Citrix/VDA/** にあります。
>
> コマンドを実行するにはフルパスが必要です。または、**/opt/Citrix/VDA/sbin** と **/opt/Citrix/VDA/bin** をシステムパスに追加できます。

手順4b：Linux VDAパッケージのダウンロード

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

手順4c：Linux VDAのインストール

Zypper を使用して Linux VDA ソフトウェアをインストールします。

SUSE 12 の場合:

```
sudo zypper install XenDesktopVDA-<version>.sle12_x.x86_64.rpm
```

RPM パッケージマネージャーを使用して Linux VDA ソフトウェアをインストールします。インストールする前に、以下の依存関係を解決してください。

SUSE 12 の場合:

```
sudo rpm -i XenDesktopVDA-<version>.sle12_x.x86_64.rpm
```

手順 4d：Linux VDA のアップグレード（オプション）

既存のインストールを、以前の2つのバージョンおよびLTSRリリースからアップグレードできます。

> **注:**
>
> 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。

> **SUSE 12 の場合:**

```
sudo rpm -U XenDesktopVDA-<version>.sle12_x.x86_64.rpm
```

SUSE 12 の RPM 依存関係リスト:

```
postgresql-server >= 9.3


postgresql-jdbc >= 9.2


java-11-openjdk >= 11


ImageMagick >= 6.8
```

```
dbus-1 >= 1.8.8

dbus-1-x11 >= 1.8.8

libXpm4 >= 3.5.11

libXrandr2 >= 1.4.2

libXtst6 >= 1.2.2

motif >= 2.3

pam >= 1.1.8

bash >= 4.2

findutils >= 4.5

gawk >= 4.1

sed >= 4.2
```

```
cups >= 1.6.0

cups-filters-foomatic-rip >= 1.0.0

openldap2 >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.8.1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

rpmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsLzma) <= 4.4.6-1

libtcmalloc4 >= 2.5
```

```
libcap-progs >= 2.22


xorg-x11-server >= 7.6_1.18.3-76.15


ibus >= 1.5


xorg- x11-server = 7.6_1.19.6


xorg-x11 = 7.6_1


postgresql10-server >= 10.12


libgtk-2_0-0 >= 2.24


libgthread-2_0-0 >= 2.48


pulseaudio-utils >= 5.0


lsb-release >= 2.0
```

重要:

アップグレード後、Linux VDA マシンを再起動します。

手順 5：Linux VDA の構成

パッケージをインストールしたら、ctxsetup.sh スクリプトを実行して Linux VDA を構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh –help
```

プロンプトによる構成

プロンプトの質問に従って手動で構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトで必要とされるオプションを環境変数で指定します。必要な変数がすべて存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux Virtual Desktopサービスは、マシンの起動後に開始されます。この値はデフォルトでYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktopサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual

Desktopに必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。

- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4** – Linux VDAは、Delivery Controllerとの認証のためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ ウィンバインド
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーション技術のセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに限定するには、DNS サイト名を指定します。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS にクエリを実行して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例：ad1.mycompany.com:389。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介して LDAP にクエリを実行します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス（FAS）サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりに FAS サーバーのセミコロン区切りリストを指定できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。

サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。

- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための .NET Core Runtime 3.1 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** – セッションで使用する GNOME または MATE デスクトップ環境を指定します。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

> **注:**
>
> ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。
>
> a. VDA の **$HOME/<username\>** ディレクトリの下に `.xsession` ファイルを作成します。
>
> b. `.xsession` ファイルを編集して、ディストリビューションに基づいてデスクトップ環境を指定します。
>
> **CentOS、Ubuntu、Debian の MATE デスクトップの場合**
>
> MSESSION="$(type -p mate-session)"
>
> \> if ; then
>
> ```
> exec mate-session
> ```
>
> \> fi
>
> ```
> **CentOS の GNOME デスクトップの場合**
> ```

```
    GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
    export GNOME_SHELL_SESSION_MODE=classic
```

\> exec gnome-session --session=gnome-classic

```
    fi

**Ubuntu および Debian の GNOME デスクトップの場合**



    GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
    exec gnome-session
```

\> fi

a. ターゲットセッションユーザーと 700 ファイル権限を共有します。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成完了時にLinux VDAサービスを開始するかどうか。デフォルトはYです。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutのリッスン用ソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutとの通信用ポート。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST='list-ddc-fqdns'

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

export CTX_XDL_SITE_NAME=dns-site-name | '<none>'

export CTX_XDL_LDAP_LIST='list-ldap-servers' | '<none>'

export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'
```

```
export CTX_XDL_FAS_LIST=‘list-fas-servers’ | '<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
```

sudoコマンドを実行するときは、既存の環境変数を新しく作成するシェルに渡すために **-E** オプションを入力します。上記のコマンドから、最初の行を **#\!/bin/bash** とするシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST=‘list-ddc-fqdns’ \
```

```
CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \
```

```
CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
```

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh** スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /usr/local/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /usr/local/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.sh および **ctxcleanup.sh** スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイルに書き込まれます。

`/tmp/xdl.configure.log`

変更を有効にするには、Linux VDAサービスを再起動します。

手順6：Linux VDAを実行する

「**ctxsetup.sh** スクリプトを使用してLinux VDAを構成した後、次のコマンドを実行してLinux VDAを制御できます。」.

「**Linux VDAの開始:**」:

「Linux VDAサービスを開始するには:」:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

「**Linux VDAの停止:**」:

「Linux VDAサービスを停止するには:」:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

> **注:**
>
> 「`ctxvda` および `ctxhdx` サービスを停止する前に、`service ctxmonitorservice stop` コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。」.

> 「**Linux VDAの再起動:**」:

「Linux VDAサービスを再起動するには:」:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

「**Linux VDAステータスの確認:**」:

「Linux VDAサービスの実行ステータスを確認するには:」:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

「ステップ7：Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する」

「マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。」.

「Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。」:

- 「オペレーティングシステムには、以下を選択します。」:
  - 「ホスト型共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。」.
  - VDI専用デスクトップ配信モデルの場合の**シングルセッションOS**オプション。
- 同じマシンカタログにLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト型共有デスクトップ配信モデルが適用されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに1人のユーザーが使用する配信モデルが適用されます。

ヒント:

マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除し、再度追加する必要があります。

ステップ8：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループと、Windowsマシンを含むマシンカタログを混在させないでください。

重要:

アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2106」を参照してください。

Ubuntu 用 Linux Virtual Delivery Agent のインストール

手動インストールについてはこの記事の手順に従うか、自動インストールと構成には簡易インストールを使用できます。簡易インストールは時間と労力を節約し、手動インストールよりもエラーが発生しにくいです。

> **注:**
>
> 簡易インストールは新規インストールにのみ使用してください。既存のインストールを更新するために簡易インストールを使用しないでください。

ステップ 1: VDA インストールのために Ubuntu を準備する

ステップ 1a: ネットワーク構成を確認する

続行する前に、ネットワークが正しく接続され、構成されていることをお勧めします。

Ubuntu 18.04 Live Server を使用している場合は、ホスト名を設定する前に **/etc/cloud/cloud.cfg** 構成ファイルで次の変更を行ってください。

```
preserve_hostname: true
```

ステップ 1b: ホスト名を設定する

マシンのホスト名が正しく報告されるように、**/etc/hostname** ファイルをマシンのホスト名のみを含むように変更してください。

```
hostname
```

ステップ 1c: ホスト名にループバックアドレスを割り当てる

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されることを確認してください。そのためには、**/etc/hosts** ファイルの次の行を変更して、FQDN とホスト名を最初の 2 つのエントリとして含めます。

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから `hostname-fqdn` または `hostname` への他の参照をすべて削除してください。

> **注:**
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダースコア (_)、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは Delivery Controller のホスト名にも適用されます。

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

手順1e：マルチキャストDNSを無効にする

デフォルト設定ではマルチキャストDNS（**mDNS**）が有効になっており、名前解決の結果が一貫しない場合があります。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

変更後：

`hosts: files dns`

手順1f：名前解決とサービス到達可能性を確認する

FQDNを解決し、ドメインコントローラーとDelivery Controller™にpingできることを確認します。

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

FQDNを解決できない場合、またはこれらのマシンのいずれかにpingできない場合は、続行する前に手順を確認してください。

手順1g：クロック同期（chrony）を構成する

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することは非常に重要です。Linux VDAを仮想マシンとしてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモートタイムサービスとの時刻同期が推奨されます。

chronyをインストールします。

```
apt-get install chrony
```

rootユーザーとして、**/etc/chrony/chrony.conf**を編集し、各リモートタイムサーバーのサーバーエントリを追加します。

`server peer1-fqdn-or-ip-address iburst`

`server peer2-fqdn-or-ip-address iburst`

一般的な展開では、時刻をローカルのドメインコントローラーから同期し、パブリックNTPプールサーバーから直接同期しないでください。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

ループバックIPアドレス、localhost、およびパブリックサーバーの***.pool.ntp.org**エントリを含む、リストされている他の*server**または*pool**エントリをすべて削除します。

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h: OpenJDK 11をインストールする

Linux VDAにはOpenJDK 11が必要です。

Ubuntu 16.04でOpenJDK 11をインストールするには、次の手順を実行します。

1. `<https://jdk.java.net/archive/>`から最新のOpenJDK 11をダウンロードします。
2. ダウンロードしたパッケージを解凍するには、`tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz`コマンドを実行します。
3. (オプション) ターゲットディレクトリにOpenJDKを保存するには、`mv jdk-11.0.2/ <target directory>`コマンドを実行します。
4. Javaランタイムをセットアップするには、`update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000`コマンドを実行します。
5. Javaのバージョンを確認するには、`java -version`コマンドを実行します。

Ubuntu 20.04およびUbuntu 18.04では、以下を使用してOpenJDK 11をインストールします。

```
sudo apt-get install -y openjdk-11-jdk
```

ステップ1i: PostgreSQLをインストールする

Linux VDAには、UbuntuにPostgreSQLバージョン9.xが必要です。

```
sudo apt-get install -y postgresql



sudo apt-get install -y libpostgresql-jdbc-java
```

ステップ1j: Motifをインストールする

```
sudo apt-get install -y libxm4
```

ステップ1k: その他のパッケージをインストールする

```
sudo apt-get install -y libsasl2-2


sudo apt-get install -y libsasl2-modules-gssapi-mit


sudo apt-get install -y libldap-2.4-2


sudo apt-get install -y krb5-user


sudo apt-get install -y libgtk2.0-0
```

ステップ2: ハイパーバイザーを準備する

サポートされているハイパーバイザー上でLinux VDAを仮想マシンとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに応じて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™での時刻同期の修正

Citrix Hypervisorの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとCitrix Hypervisorの両方がシステムクロックを管理しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

一部のLinuxディストリビューションでは、Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合は、ファイルに1を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を再起動後も永続的に適用するには、**/etc/sysctl.conf ファイル**を編集し、次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration ServicesがインストールされているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックが正確に保たれるように、NTPサービスと並行してこの機能を有効にしてください。

管理オペレーティングシステムから:

1. Hyper-Vマネージャーコンソールを開きます。

2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻の同期**が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTPとの競合を避けるためにホストの時刻同期が無効になっているVMwareおよびCitrix Hypervisorとは異なります。Hyper-Vの時刻同期は、NTPの時刻同期と共存し、補完することができます。

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内で、NTPとハイパーバイザーの両方がシステムクロックを同期しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合:

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。
4. **VMware Tools**を選択します。
5. **詳細設定**ボックスで、**ゲストの時刻をホストと同期**のチェックを外します。

ステップ3: Linux仮想マシン (VM) をWindowsドメインに追加する

Linux VDA は、Linux マシンを Active Directory (AD) ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ・ウィンバインド
- クエスト認証サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新する

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

マシンの起動時に Winbind デーモンが開始するように有効にする

Winbind デーモンは、マシンの起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind` スクリプトが `/etc/init.d` の下にあることを確認してください。

Kerberos を構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

> `[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM = {`

`admin_server = domain-controller-fqdn`

`kdc = domain-controller-fqdn`

`}`

`[domain_realm]`

`domain-dns-name = REALM`

`.domain-dns-name = REALM`

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

Winbind認証を構成する

UbuntuにはRHELの`authconfig`やSUSEのyast2のようなツールがないため、Winbindを手動で構成します。

/etc/samba/smb.conf を開き、以下の設定を行います。

`[global]`

`workgroup = WORKGROUP`

`security = ADS`

`realm = REALM`

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`winbind trusted domains only = no`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字のKerberosレルム名です。

`nsswitch` を構成する

/etc/nsswitch.conf を開き、次の行に `winbind` を追加します。

`passwd: compat winbind`

`group: compat winbind`

Windows ドメインへの参加

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

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

ここで、**REALM** は大文字の Kerberos レルム名、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

`winbind` の再起動

```
sudo systemctl restart winbind
```

Winbind 用の PAM の構成

次のコマンドを実行し、**Winbind NT/Active Directory authentication** および **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ヒント:

`winbind` デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

ドメインメンバーシップの確認

Delivery Controller は、Windows または Linux のいずれであっても、すべての VDA マシンが Active Directory にコンピューターオブジェクトを持つことを要求します。

Samba の **net ads** コマンドを実行して、マシンがドメインに参加していることを確認します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

Kerberos 構成の確認

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐため、バックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT チケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist
```

以下を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

wbinfo ツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインは、Kerberos レルム名ではなく、AD ドメイン名です。bash シェルの場合、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAM モジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username



id -u
```

注:

SSH コマンドを正常に実行するには、SSH が有効になっており、正しく機能していることを確認してください。

id -u コマンドによって返された uid に対して、対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーの Kerberos 資格情報キャッシュ内のチケットが有効で期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンすることで実行できます。ドメイン参加の確認後、手順 4: Linux VDA をインストールする に進みます。

ヒント:

> ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップを表示できない場合は、マシンを再起動してから、もう一度試してください。

Quest 認証サービス

ドメインコントローラーでの Quest の構成

QuestソフトウェアをActive Directoryドメインコントローラーにインストールして構成済みであり、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unixを有効にする**をオンにします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

Linux VDAでQuestを構成する

SELINUXポリシーの適用を回避する

デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinuxを無効にすることです。rootユーザーとして、***/etc/selinux/config***を編集し、**SELinux**設定を変更します。

```
SELINUX=disabled
```

この変更にはマシンの再起動が必要です。

```
reboot
```

重要:

この設定は慎重に使用してください。無効にした後にSELinuxポリシーの適用を再度有効にすると、rootユーザーや他のローカルユーザーであっても、完全にロックアウトされる可能性があります。

VASデーモンを構成する

Kerberosチケットの自動更新を有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります:

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間のチケット有効期間より1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。

PAMとNSSを構成する

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、PAMとNSSを手動で構成するために、次のコマンドを実行します:

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

Windowsドメインに参加する

Quest `vastool` コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます:

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。ドメイン名（domain-name）は、ドメインのDNS名です（例: example.com）。

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを要求します。Questによって参加したLinuxマシンがドメイン上にあることを確認するには:

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます:

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証を確認する

QuestがPAMを介してドメインユーザーを認証できることを確認するには、以前に使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

id -uコマンドによって返されるUIDに対して、対応するKerberos資格情報キャッシュファイルが作成されたことを確認します:

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で期限切れでないことを確認します:

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、Step 4: Install the Linux VDAに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインへの参加

Centrify DirectControl Agentがインストールされている場合、Centrify `adjoin`コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su -

adjoin -w -V -u user domain-name
```

userパラメーターは、コンピューターをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name**パラメーターは、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを要求します。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
su -



adinfo
```

Joined to domainの値が有効であり、**CentrifyDC mode**が**connected**を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo --diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、Step 4: Install the Linux VDAに進みます。

SSSD

Kerberosの構成

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、rootとして**/etc/krb5.conf**を開き、パラメーターを設定します。

注:

Kerberos を AD インフラストラクチャに基づいて構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストでの `domain-dns-name` パラメーターは、example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。

ドメインに参加する

SSSD は、Active Directory を ID プロバイダーとして、Kerberos を認証に使用するように構成する必要があります。ただし、SSSD はドメイン参加やシステム keytab ファイルの管理のための AD クライアント機能を提供しません。代わりに `adcli`、`realmd`、または `Samba` を使用できます。

> **注:**
>
> このセクションでは、`adcli` と `Samba` の情報のみを提供します。

- **`adcli` を使用してドメインに参加する場合は、次の手順を完了してください。**

1. `adcli` をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. `adcli` を使用してドメインに参加します。

   古いシステム keytab ファイルを削除し、以下を使用してドメインに参加します。

   ```
   su -



   rm -rf /etc/krb5.keytab
   ```

```
adcli join domain-dns-name -U user -H hostname-fqdn
```

「**user**」は、マシンをドメインに追加する権限を持つドメインユーザーです。「**hostname-fqdn**」は、マシンの FQDN 形式のホスト名です。

「**-H**」オプションは、Linux VDA が必要とする host/hostname-fqdn@REALM 形式で SPN を生成するために `adcli` に必要です。

3. システム keytab を検証します。

   Ubuntu 20.04 マシンの場合、`adcli testjoin` コマンドを実行して、ドメインに参加しているかどうかをテストします。

   Ubuntu 18.04 または Ubuntu 16.04 マシンの場合、`sudo klist -ket` コマンドを実行して、システム keytab ファイルが作成されていることを確認します。

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **`Samba` を使用してドメインに参加する場合は、次の手順を完了します。**

1. パッケージをインストールします。

```
sudo apt-get install samba krb5-user
```

2. `Samba` を構成します。

   /etc/samba/smb.conf を開き、次の設定を行います。

   `[global]`

   `workgroup =` *`WORKGROUP`*

   `security = ADS`

   `realm =` *`REALM`*

   `client signing = yes`

   `client use spnego = yes`

   `kerberos method = secrets and keytab`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字の Kerberos レルム名です。

3. `Samba` を使用してドメインに参加します。

   ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Windows アカウントが必要です。

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

   ここで、**REALM** は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD をセットアップする

必要なパッケージをインストールまたは更新します:

必要なSSSDおよび構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新をお勧めします。

```
sudo apt-get install --only-upgrade sssd
```

> **注:**
>
> デフォルトでは、Ubuntuのインストールプロセスは**nsswitch.conf**とPAMログインモジュールを自動的に構成します。

SSSD を構成する

SSSDデーモンを開始する前に、SSSD構成の変更が必要です。一部のSSSDバージョンでは、**/etc/sssd/sssd.conf**構成ファイルはデフォルトでインストールされておらず、手動で作成する必要があります。rootとして、**/etc/sssd/sssd.conf**を作成または開き、次の設定を行います。

`[sssd]`

`services = nss, pam`

```
config_file_version = 2

domains = domain-dns-name

[domain/domain-dns-name]

id_provider = ad

access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14
days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

> **注:**
>
> ldap_id_mapping は **true** に設定されており、SSSD自体がWindows SIDをUnix UIDにマッピングする処理を行います。そうでない場合、Active DirectoryはPOSIX拡張機能を提供でき

る必要があります。PAMサービス `ctxhdx` が ad_gpo_map_remote_interactive に追加されます。

このコンテキストでの **domain-dns-name** パラメーターは、example.comのようなDNSドメイン名です。**REALM** は、EXAMPLE.COMのような大文字のKerberosレルム名です。NetBIOSドメイン名を構成する必要はありません。

構成設定の詳細については、sssd.conf および `sssd-ad` のmanページを参照してください。

SSSDデーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSDデーモンを開始する

SSSDデーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM構成

次のコマンドを実行し、**SSS認証**と**ログイン時にホームディレクトリを作成**のオプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。

- ドメインメンバーシップを確認するために`adcli`を使用する場合は、ドメイン情報を表示するために`sudo adcli info domain-dns-name`コマンドを実行します。
- ドメインメンバーシップを確認するために`Samba`を使用する場合は、マシンがドメインに参加していることを確認するために`sudo net ads testjoin`コマンドを、追加のドメインおよびコンピューターオブジェクト情報を確認するために`sudo net ads info`コマンドを実行します。

Kerberos構成の確認

KerberosがLinux VDAで使用するために正しく構成されていることを確認するには、システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit`コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号（$）は、シェル置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

SSSD PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username
```

```
id -u



klist



exit
```

klistコマンドによって返されたKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

rootユーザーとして、以前の**id -u**コマンドによって返されたuidに対応するチケットキャッシュファイルが作成されていることを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDEまたはGnome Display Managerにログオンすることで実行できます。ドメイン参加の検証後、手順4：Linux VDAをインストールするに進みます。

PBIS

必要なPBISパッケージをダウンロードする

例:

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行可能にする

例:

```
sudo chmod +x pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行する

例:

```
sudo sh pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

Windowsドメインに参加する

ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
sudo /opt/pbis/bin/domainjoin-cli join domain-name user
```

userは、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name**はドメインのDNS名です（例：example.com）。

注：Bashをデフォルトシェルとして設定するには、**sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash**コマンドを実行します。

ドメインメンバーシップを確認する

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順4：Linux VDAのインストールに進みます。

手順4：Linux VDAのインストール

手順4a：Linux VDAパッケージのダウンロード

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

手順4b：Linux VDAのインストール

Debianパッケージマネージャーを使用してLinux VDAソフトウェアをインストールします。

Ubuntu 20.04の場合：

```
sudo dpkg -i xendesktopvda_<version>.ubuntu20.04_amd64.deb
```

Ubuntu 18.04の場合：

```
sudo dpkg -i xendesktopvda_<version>.ubuntu18.04_amd64.deb
```

Ubuntu 16.04の場合：

```
sudo dpkg -i xendesktopvda_<version>.ubuntu16.04_amd64.deb
```

Ubuntu 20.04のDebian依存関係リスト：

```
postgresql >= 12



libpostgresql-jdbc-java >= 42.2
```

```
openjdk-11-jdk >= 11

imagemagick >= 8:6.9.10

ufw >= 0.36

ubuntu-desktop >= 1.450

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.34

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.7.0
```

```
sed >= 4.7

cups >= 2.3

libmspack0 >= 0.10

libgoogle-perftools4 >= 2.7~

libpython2.7 >= 2.7~
```

Ubuntu 18.04のDebian依存関係リスト：

```
postgresql >= 9.5

libpostgresql-jdbc-java >= 9.2

openjdk-11-jdk >= 11

gtk3-nocsd >=3

imagemagick >= 8:6.8.9.9

ufw >= 0.35
```

```
ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1

libldap-2.4-2 >= 2.4.42

libsasl2-modules-gssapi-mit >= 2.1.~
```

```
python-requests >= 2.9.1

libgoogle-perftools4 >= 2.4~

xserver-xorg-core >= 2:1.18

xserver-xorg-core << 2:1.19

x11vnc>=0.9.13

python-websockify >= 0.6.1
```

Ubuntu 16.04 の Debian 依存関係リスト:

```
postgresql >= 9.5

libpostgresql-jdbc-java >= 9.2

imagemagick >= 8:6.8.9.9

ufw >= 0.35
```

```
ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1

libldap-2.4-2 >= 2.4.42

libsasl2-modules-gssapi-mit >= 2.1.~
```

```
python-requests >= 2.9.1


libgoogle-perftools4 >= 2.4~


xserver-xorg-core >= 2:1.18


xserver-xorg-core << 2:1.19


x11vnc>=0.9.13


python-websockify >= 0.6.1
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、システム要件を参照してください。

ステップ 4c: Linux VDA をアップグレードする (オプション)

既存のインストールを、以前の2つのバージョンおよびLTSRリリースからアップグレードできます。

```
sudo dpkg -i <PATH>/<Linux VDA deb>
```

注:

> 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、ファイルをバックアップしてください。

ステップ 4d: Linux VDA を構成する

パッケージをインストールした後、ctxsetup.sh スクリプトを実行して Linux VDA を構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトが表示される質問に従って手動で構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求せず、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。デフォルトでは N に設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller への登録に使用する Delivery Controller の完全修飾ドメイン名 (FQDN) のスペース区切りリストを必要とします。少なくとも1つの FQDN または CNAME エイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。デフォルトではポート 80 です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktop サービスは、マシンの起動後に開始されます。デフォルトでは Y に設定されています。

- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktop サービスでは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux Virtual Desktop のシステムファイアウォールで、必要なポート（デフォルトではポート 80 と 1494）を自動的に開くことができます。デフォルトでは Y に設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \|5** – Linux VDA は、Delivery Controller と認証するために Kerberos 構成設定を必要とします。Kerberos 構成は、システムにインストールおよび構成されている Active Directory 統合ツールから決定されます。使用するサポートされている Active Directory 統合方法を指定します。
  - 1 – サンバ・ウィンバインド
  - 2 – クエスト認証サービス
  - 3 – セントリファイ・ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境では、この変数を Y に設定します。この変数はデフォルトで N に設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに限定するには、DNS サイト名を指定します。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は、LDAP サーバーを検出するために DNS を照会します。DNS が LDAP サービスレコードを提供できない場合、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例：ad1.mycompany.com:389。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介して LDAP を照会します。ただし、検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス（FAS）サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサ

ポートしていませんが、代わりにセミコロン区切りの FAS サーバーのリストを提供できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白をテキスト文字列 **\<none\>** で埋め、サーバーアドレスの順序を変更しないでください。

- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするための .NET Core Runtime 3.1 のインストールパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** – セッションで使用するGNOMEまたはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、変数の値を**mate**に設定する必要があります。

> **注:**
>
> ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。
>
> a. VDA上の**$HOME/<username\>**ディレクトリに `.xsession` ファイルを作成します。
>
> b. ディストリビューションに基づいてデスクトップ環境を指定するように、 `.xsession` ファイルを編集します。
>
> **CentOS、Ubuntu、およびDebian上のMATEデスクトップの場合**
>
> MSESSION="$(type -p mate-session)"
>
> \> if ; then
>
> ```
> exec mate-session
> ```
>
> \> fi

```
**CentOS上のGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
   export GNOME_SHELL_SESSION_MODE=classic
```

\> exec gnome-session --session=gnome-classic

```
   fi

**UbuntuおよびDebian上のGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
   exec gnome-session
```

\> fi

a. ターゲットセッションユーザーと700ファイル権限を共有します。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成が完了したときにLinux VDAサービスを開始するかどうか。デフォルトではYに設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutをリッスンするためのソケットポート。デフォルトのポートは7503です。

- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトのポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST='list-ddc-fqdns'

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

export CTX_XDL_SITE_NAME=dns-site-name | '<none>'

export CTX_XDL_LDAP_LIST='list-ldap-servers' | '<none>'
```

```
export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'

export CTX_XDL_FAS_LIST=‘list-fas-servers’ | '<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
```

sudo コマンドを実行するときは、既存の環境変数を新しく作成するシェルに渡すために **-E** オプションを入力します。前のコマンドから、最初の行に **#\!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST=‘list-ddc-fqdns’ \
```

```
CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST=‘list-ldap-servers’ \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST=‘list-fas-servers’ \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \
```

```
CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
```

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル **/tmp/xdl.configure.log** に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

Linux VDAソフトウェアをアンインストールする

Linux VDAがインストールされているかどうかを確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

注:

Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前にセットアップされたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

ヒント:

このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

ステップ5：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成したら、次のコマンドを使用してLinux VDAを制御します。

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

注:

`ctxvda` および `ctxhdx` サービスを停止する前に、`service ctxmonitorservice stop` コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx
```

```
sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ6：Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムについては、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

> **注:**
>
> Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホストされた共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに1人のユーザーという配信モデルが意味されます。

> **ヒント:**

> マシンをActive Directoryドメインから削除して再参加させる場合は、そのマシンをマシンカタログから削除して再度追加する必要があります。

ステップ7：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループと、Windowsマシンを含むマシンカタログを混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2106」を参照してください。

Debian 用 Linux Virtual Delivery Agent をインストールする

手動インストールについてはこの記事の手順に従うか、自動インストールと構成にはeasy installを使用できます。Easy install は時間と労力を節約し、手動インストールよりもエラーが発生しにくいです。

> **注:**

> Easy install は新規インストールにのみ使用してください。既存のインストールを更新するために Easy install を使用しないでください。

ステップ 1: VDA インストール用に Debian を準備する

ステップ 1a: ネットワーク構成を確認する

続行する前に、ネットワークが正しく接続され、構成されていることをお勧めします。

ステップ 1b: ホスト名を設定する

マシンのホスト名が正しく報告されるようにするには、**/etc/hostname** ファイルを、マシンのホスト名のみを含むように変更します。

```
hostname
```

ステップ 1c: ホスト名にループバックアドレスを割り当てる

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されることを確認してください。そのためには、**/etc/hosts** ファイルの次の行を変更して、FQDN とホスト名を最初の 2 つのエントリとして含めます。

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、`hostname-fqdn` または `hostname` への他の参照を削除します。

> **注:**
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダーバー (_)、スペース、およびその他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは、Delivery Controller のホスト名にも適用されます。

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e：マルチキャストDNSを無効にする

デフォルト設定ではマルチキャストDNS（**mDNS**）が有効になっており、名前解決の結果が不安定になる可能性があります。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

次のように変更します。

`hosts: files dns`

ステップ1f：名前解決とサービス到達可能性を確認する

FQDNを解決し、ドメインコントローラーとDelivery Controller™にpingできることを確認します。

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn



nslookup delivery-controller-fqdn



ping delivery-controller-fqdn
```

FQDNを解決できない場合、またはこれらのマシンのいずれかにpingできない場合は、続行する前に手順を確認してください。

ステップ1g：クロック同期（chrony）を構成する

VDA、Delivery Controller、ドメインコントローラー間で正確なクロック同期を維持することは非常に重要です。Linux VDAを仮想マシンとしてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモートタイムサービスとの時刻同期が推奨されます。

chronyをインストールします。

```
apt-get install chrony
```

rootユーザーとして、**/etc/chrony/chrony.conf**を編集し、各リモートタイムサーバーのサーバーエントリを追加します。

`server peer1-fqdn-or-ip-address iburst`

`server peer2-fqdn-or-ip-address iburst`

通常の展開では、パブリックNTPプールサーバーから直接ではなく、ローカルのドメインコントローラーから時刻を同期します。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

ループバックIPアドレス、ローカルホスト、パブリックサーバーの***.pool.ntp.org**エントリなど、リストされている他の**サーバーまたは**プール**エントリをすべて削除します。

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h: パッケージのインストール

```
sudo apt-get install -y libsasl2-2



sudo apt-get install -y libgtk2.0-0
```

ステップ1i: oldstableリポジトリの追加

Debianディストリビューションに必要な依存関係をインストールするには、`/etc/apt/sources.list`ファイルに`deb http://deb.debian.org/debian/ oldstable main`行を追加します。

ステップ2: ハイパーバイザーの準備

サポートされているハイパーバイザー上でLinux VDAを仮想マシンとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに応じて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™での時刻同期の修正

Citrix Hypervisorの時刻同期機能が有効になっている場合、各準仮想化Linux VM内で、NTPとCitrix Hypervisorの両方がシステムクロックを管理しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

一部のLinuxディストリビューションでは、Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclockファイルが存在しない場合、以下の手順は不要です。

有効になっている場合、ファイルに1を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を再起動後も永続的に維持するには、**/etc/sysctl.conf ファイル**を編集し、次の行を追加します。

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration Servicesがインストールされ ているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックが正確であることを保証するために、NTPサービスと並行してこの機能を有効にします。

管理オペレーティングシステムから:

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTPとの競合を避けるためにホスト時刻同期が無効になっているVMwareおよびCitrix Hypervisorとは異なります。Hyper-Vの時刻同期は、NTP時刻同期と共存し、それを補完することができます。

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内で、NTPとハイパーバイザーの両方がシステムクロックを同期しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるために、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホスト時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合:

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. 「**仮想マシンプロパティ**」ダイアログで、「**オプション**」タブを開きます。
4. 「**VMware Tools**」を選択します。

5. 「**詳細設定**」ボックスで、「**ゲストの時刻をホストと同期**」のチェックボックスをオフにします。

ステップ3：Linux仮想マシン（VM）をWindowsドメインに追加する

Linux VDAは、LinuxマシンをActive Directory（AD）ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ・ウィンバインド
- クエスト認証サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新する

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

マシンの起動時にWinbindデーモンが開始するように有効にする

Winbindデーモンは、マシンの起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> winbindスクリプトが `/etc/init.d` の下にあることを確認してください。

Kerberosを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。次の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]

default_realm = REALM

dns_lookup_kdc = false

[realms]

REALM = {

admin_server = domain-controller-fqdn

kdc = domain-controller-fqdn

}

[domain_realm]

domain-dns-name = REALM

.domain-dns-name = REALM
```

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

Winbind認証を構成する

/etc/samba/smb.conf を開き、次の設定を行います。

```
[global]

workgroup = WORKGROUP

security = ADS
```

`realm = REALM`

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`winbind trusted domains only = no`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字の Kerberos レルム名です。

nsswitch を構成する

/etc/nsswitch.conf を開き、次の行に `winbind` を追加します。

`passwd: systemd winbind`

`group: systemd winbind`

Windows ドメインに参加する

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

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

ここで、**REALM** は大文字の Kerberos レルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

Winbind を再起動する

```
sudo systemctl restart winbind
```

Winbind 用に PAM を構成する

次のコマンドを実行し、**Winbind NT/Active Directory authentication** と **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ヒント:

マシンがドメインに参加している場合にのみ、`winbind` デーモンは実行され続けます。

ドメインメンバーシップを確認する

Delivery Controller は、Windows または Linux のすべての VDA マシンが Active Directory にコンピューターオブジェクトを持つことを要求します。

マシンがドメインに参加していることを確認するには、Samba の **net ads** コマンドを実行します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

Kerberos構成の確認

Linux VDAでKerberosが正しく構成されていることを確認するには、システム**keytab**ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェル置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名が

Kerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist
```

以下を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

wbinfoツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっており、正しく機能していることを確認してください。

id -uコマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効であり、期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の確認後、手順4：Linux VDAのインストール に進みます。

ヒント:

ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップが表示されない場合は、マシンを再起動してからもう一度試してください。

Quest認証サービス

ドメインコントローラーでのQuestの構成

Active Directoryドメインコントローラーにインストールして構成済みであり、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unix対応**をオンにします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合と同等です。

Linux VDAでのQuestの構成

SELINUXポリシーの適用を回避する

デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨害され、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinuxを無効にすることです。rootユーザーとして、***/etc/selinux/config***を編集し、**SELinux**設定を変更します。

`SELINUX=disabled`

この変更にはマシンの再起動が必要です:

```
reboot
```

> **重要:**
>
> この設定は慎重に使用してください。SELinuxポリシーの適用を無効にした後に再度有効にすると、rootユーザーや他のローカルユーザーであっても完全にロックアウトされる可能性があります。

VASデーモンの構成

Kerberosチケットの自動更新は有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります:

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400
```

```
sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間のチケット有効期間よりも1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。

PAMとNSSの構成

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、次のコマンドを実行してPAMとNSSを手動で構成します:

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

Windowsドメインへの参加

Quest `vastool` コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます:

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。ドメイン名（domain-name）は、ドメインのDNS名です（例: example.com）。

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持っていることを要求します。Questに参加しているLinuxマシンがドメイン上にあることを確認するには:

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます:

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

「**id -u**」コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順4：Linux VDAをインストールするに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインに参加する

Centrify DirectControl Agentがインストールされたら、Centrify `adjoin` コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su –

adjoin -w -V -u user domain-name
```

userパラメーターは、コンピューターをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name**パラメーターは、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを要求します。Centrifyで参加したLinuxマシンがドメイン上にあることを確認するには：

```
su -



adinfo
```

Joined to domainの値が有効であり、**CentrifyDC mode**が**connected**を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo --diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、手順4：Linux VDAをインストールするに進みます。

SSSD

Kerberos を構成する

Kerberos をインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberos を構成するには、root として **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> AD インフラストラクチャに基づいて Kerberos を構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストでの `domain-dns-name` パラメーターは、example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。

ドメインに参加する

SSSD は、ID プロバイダーとして Active Directory を使用し、認証に Kerberos を使用するように構成する必要があります。ただし、SSSD は、ドメインへの参加やシステムキータブファイルの管理のための AD クライアント機能を提供しません。代わりに `adcli`、`realmd`、または `Samba` を使用できます。

> **注:**

> このセクションでは、`adcli` と `Samba` の情報のみを提供します。

- **adcli を使用してドメインに参加する場合は、次の手順を完了します。**

1. `adcli` をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. `adcli` を使用してドメインに参加します。

   古いシステムキータブファイルを削除し、以下を使用してドメインに参加します。

   ```
   su -



   rm -rf /etc/krb5.keytab



   adcli join domain-dns-name -U user -H hostname-fqdn
   ```

   user は、ドメインにマシンを追加する権限を持つドメインユーザーです。**hostname-fqdn** は、マシンの FQDN 形式のホスト名です。

   -H オプションは、Linux VDA が必要とする host/hostname-fqdn@REALM 形式で SPN を生成するために `adcli` に必要です。

3. システムキータブを確認します。

   システムキータブファイルが作成されていることを確認するために、`sudo klist -ket` コマンドを実行します。

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **`Samba` を使用してドメインに参加する場合は、次の手順を完了します。**

1. パッケージをインストールします。

   ```
   sudo apt-get install samba krb5-user
   ```

2. `Samba` を構成します。

/etc/samba/smb.conf を開き、次の設定を行います。

`[global]`

`workgroup =` *`WORKGROUP`*

`security = ADS`

`realm =` *`REALM`*

`client signing = yes`

`client use spnego = yes`

`kerberos method = secrets and keytab`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字の Kerberos レルム名です。

3. `Samba` でドメインに参加します。

   ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つWindowsアカウントが必要です。

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

   ここで、**REALM** は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD のセットアップ

必要なパッケージのインストールまたは更新:

必要なSSSDおよび構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新をお勧めします。

```
sudo apt-get install --only-upgrade sssd
```

> **注:**
>
> デフォルトでは、Ubuntuのインストールプロセスは**nsswitch.conf**とPAMログインモジュールを自動的に構成します。

SSSD の構成

SSSDデーモンを開始する前に、SSSD構成の変更が必要です。SSSDの一部のバージョンでは、**/etc/sssd/sssd.conf**構成ファイルはデフォルトでインストールされていないため、手動で作成する必要があります。rootとして、**/etc/sssd/sssd.conf**を作成または開いて、次の設定を行います。

```
[sssd]

services = nss, pam

config_file_version = 2

domains = domain-dns-name

[domain/ domain-dns-name ]

id_provider = ad

access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14
days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value
```

`ldap_id_mapping = true`

`override_homedir = /home/%d/%u`

`default_shell = /bin/bash`

`ad_gpo_map_remote_interactive = +ctxhdx`

注:

ldap_id_mapping は **true** に設定されており、SSSD自体がWindows SIDをUnix UIDにマッピングする処理を行います。そうでない場合、Active DirectoryはPOSIX拡張機能を提供できる必要があります。PAMサービス `ctxhdx` は ad_gpo_map_remote_interactive に追加されます。

このコンテキストでの **domain-dns-name** パラメーターは、example.comのようなDNSドメイン名です。**REALM** は、EXAMPLE.COMのような大文字のKerberosレルム名です。NetBIOSドメイン名を構成する必要はありません。

構成設定については、sssd.conf と `sssd-ad` の man ページを参照してください。

SSSD デーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSD デーモンの起動

SSSD デーモンを今すぐ起動し、マシンの起動時にデーモンが起動するようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM 構成

次のコマンドを実行し、**SSS 認証**と**ログイン時にホームディレクトリを作成**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が Active Directory にコンピューターオブジェクトを持っていることを要求します。

- ドメインメンバーシップを確認するために `adcli` を使用する場合は、`sudo adcli info domain-dns-name` コマンドを実行してドメイン情報を表示します。
- ドメインメンバーシップを確認するために `Samba` を使用する場合は、`sudo net ads testjoin` コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info` コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

Kerberos 構成の確認

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システム keytab ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT がキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンで認証を直接テストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

SSSD PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返されるKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

ルートユーザーとして、以前の **id -u** コマンドによって返されたuidに対応するチケットキャッシュファイルが作成されていることを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDEまたはGnome Display Managerにログオンすることで実行できます。ドメイン参加の検証後、手順4：Linux VDAのインストールに進みます。

PBIS

必要なPBISパッケージをダウンロードする

例:

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行可能にする

例:

```
sudo chmod +x pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行する

例:

```
sudo sh pbis-open-8.8.0.506.linux.x86_64.deb.sh
```

Windowsドメインに参加する

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
sudo /opt/pbis/bin/domainjoin-cli join domain-name user
```

user は、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例：example.com）。

注： Bashをデフォルトシェルとして設定するには、**sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** コマンドを実行します。

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が Active Directory にコンピューターオブジェクトを持っていることを要求します。PBIS に参加している Linux マシンがドメイン上にあることを確認するには:

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加している AD ドメインと OU に関する情報を返します。そうでない場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBIS が PAM を介してドメインユーザーを認証できることを確認するには、以前に使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返された UID に対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順 4: Linux VDA のインストール に進みます。

手順 4: Linux VDA のインストール

手順 4a: Linux VDA パッケージのダウンロード

Citrix Virtual Apps and Desktops ダウンロードページ にアクセスします。Citrix Virtual Apps and Desktops の適切なバージョンを展開し、**Components** をクリックして、お使いの Linux ディストリビューションに一致する Linux VDA パッケージをダウンロードします。

手順 4b: Linux VDA のインストール

Debian パッケージマネージャーを使用して Linux VDA ソフトウェアをインストールします。

```
sudo dpkg -i xendesktopvda_<version>.debian10_amd64.deb
```

Debian 10.7 の Debian 依存関係リスト:

```
postgresql              >= 11

libpostgresql-jdbc-java >= 42.2

openjdk-8-jdk           >= 8u252

imagemagick             >= 8:6.9.10
```

```
ufw                     >= 0.36

desktop-base            >= 10.0.2

libxrandr2              >= 2:1.5.1

libxtst6                >= 2:1.2.3

libxm4                  >= 2.3.8

util-linux              >= 2.33

gtk3-nocsd              >= 3

bash                    >= 5.0

findutils               >= 4.6.0

sed                     >= 4.7

cups                    >= 2.2

ghostscript             >= 9.27~

libmspack0              >= 0.10

libgoogle-perftools4    >= 2.7~

libpython2.7            >= 2.7~

libsasl2-modules-gssapi-mit >= 2.1.~
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、システム要件 を参照してください。

手順 4c: Linux VDA のアップグレード (オプション)

既存のインストールを、以前の2つのバージョンおよびLTSRリリースからアップグレードできます。

```
sudo dpkg -i <PATH>/<Linux VDA deb>
```

注:

既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。

ステップ4d: Linux VDAを構成する

パッケージをインストールした後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトに従って手動で構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求しないため、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。

- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。デフォルトではポート80です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktopサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktopサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktopのシステムファイアウォールで、必要なポート（デフォルトではポート80と1494）を自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \|5** – Linux VDAでは、Delivery Controllerで認証するためにKerberos構成設定が必要です。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールによって決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ・ウィンバインド
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するように設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDAはDNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを

含むLDAP FQDNのスペース区切りリストを提供できます。例：ad1.mycompany.com:389。この変数はデフォルトで**\<none\>**に設定されています。

- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを介してLDAPにクエリを実行します。ただし、検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service（FAS）サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていませんが、代わりにFASサーバーのセミコロン区切りリストを提供できます。シーケンスはADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を**\<none\>**テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするために.NET Core Runtime 3.1をインストールするパス。デフォルトのパスは/usr/binです。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** – セッションで使用するGNOMEまたはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、変数の値を**mate**に設定する必要があります。

> **注:**
>
> ターゲットセッションユーザーのデスクトップ環境は、次の手順を実行して変更することもできます。
>
> a. VDA上の**$HOME/<username\>**ディレクトリの下に`.xsession`ファイルを作成します。
>
> b. ディストリビューションに基づいてデスクトップ環境を指定するように、`.xsession`ファイルを編集します。
>
> **CentOS、Ubuntu、およびDebianでのMATEデスクトップの場合**
>
> MSESSION="$(type -p mate-session)"

\> if ; then

```
exec mate-session
```

\> fi

```
**CentOSでのGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
   export GNOME_SHELL_SESSION_MODE=classic
```

\> exec gnome-session --session=gnome-classic

```
   fi

**UbuntuおよびDebian上のGNOMEデスクトップの場合**


   GSESSION="$(type -p gnome-session)"
```

\> if ; then

```
   exec gnome-session
```

\> fi

a. ターゲットセッションユーザーと700のファイル権限を共有します。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成が完了したときにLinux VDAサービスが開始されるかどうか。デフォルトではYに設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutをリッスンするためのソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N


export CTX_XDL_DDC_LIST='list-ddc-fqdns'


export CTX_XDL_VDA_PORT=port-number


export CTX_XDL_REGISTER_SERVICE=Y|N


export CTX_XDL_ADD_FIREWALL_RULES=Y|N


export CTX_XDL_AD_INTEGRATION=1|2|3|4|5


export CTX_XDL_HDX_3D_PRO=Y|N


export CTX_XDL_VDI_MODE=Y|N
```

```
export CTX_XDL_SITE_NAME=dns-site-name | '<none>'

export CTX_XDL_LDAP_LIST=‘list-ldap-servers’ | '<none>'

export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'

export CTX_XDL_FAS_LIST=‘list-fas-servers’ | '<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
```

sudoコマンドを実行するときは、既存の環境変数を新しく作成するシェルに渡すために、**-E**オプションを入力します。前のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定できます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \
```

```
CTX_XDL_FAS_LIST=‘list-fas-servers’ \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
```

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

> **重要:**
>
> このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル**/tmp/xdl.configure.log**に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

Linux VDAソフトウェアをアンインストールする

Linux VDAがインストールされているかどうかを確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

> **注:**
>
> Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前にセットアップされたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

> **ヒント:**
>
> このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

手順5：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成したら、次のコマンドを使用してLinux VDAを制御します。

Linux VDA を起動する:

Linux VDA サービスを起動するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDA を停止する:

Linux VDA サービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

注:

`ctxvda` と `ctxhdx` サービスを停止する前に、モニターサービスデーモンを停止するために `service ctxmonitorservice stop` コマンドを実行します。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDA を再起動する:

Linux VDA サービスを再起動するには：

```
sudo systemctl stop ctxvda
```

```
sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDA のステータスを確認する:

Linux VDA サービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ 6: Citrix Virtual Apps または Citrix Virtual Desktops™ でマシンカタログを作成する

マシンカタログを作成し、Linux VDA マシンを追加するプロセスは、従来の Windows VDA のアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDA マシンを含むマシンカタログを作成する場合、Windows VDA マシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムについては、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッション OS** オプション。
  - VDI 専用デスクトップ配信モデルの場合は、**シングルセッション OS** オプション。
- 同じマシンカタログにLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト型共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに単一ユーザーの配信モデルが意味されます。

> ヒント:
>
> マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除して再度追加する必要があります。

ステップ7：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループを、Windowsマシンを含むマシンカタログと混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2106」を参照してください。

Linux VDA を構成する

このセクションでは、Linux VDAの機能について、その説明、構成、トラブルシューティングを含めて詳しく説明します。

ヒント:

ログ収集に使用される `xdlcollect` Bashスクリプトは、Linux VDAソフトウェアに統合されており、/opt/Citrix/VDA/binにあります。Linux VDAのインストール後、`bash /opt/Citrix/VDA/bin/xdlcollect.sh` コマンドを実行してログを収集できます。

ログ収集が完了すると、スクリプトと同じフォルダーに圧縮ログファイルが生成されます。`xdlcollect` は、圧縮ログファイルをCitrix Insight Services (CIS) にアップロードするかどうかを尋ねることができます。同意した場合、アップロード完了後、`xdlcollect` はupload_IDを返します。アップロードによって、ローカルマシンから圧縮ログファイルが削除されることはありません。他のユーザーは、upload_IDを使用してCIS内のログファイルにアクセスできます。

NIS と Active Directory の統合

この記事では、SSSD を使用して Linux VDA 上で NIS を Windows Active Directory (AD) と統合する方法について説明します。Linux VDA は Citrix Virtual Apps and Desktops のコンポーネントと見なされます。その結果、Windows AD 環境に緊密に適合します。

AD の代わりに NIS を UID および GID プロバイダーとして使用する場合、アカウント情報（ユーザー名とパスワードの組み合わせ）が AD と NIS の両方で同じである必要があります。

注:

認証は引き続き AD サーバーによって実行されます。NIS+ はサポートされていません。NIS を UID および GID プロバイダーとして使用する場合、Windows サーバーからの POSIX 属性は使用されなくなります。

> **ヒント:**

> この方法は、Linux VDA を展開する非推奨の方法であり、特別なユースケースでのみ使用されます。RHEL/CentOS ディストリビューションの場合は、RHEL/CentOS 用 Linux Virtual Delivery Agent のインストール の手順に従ってください。Ubuntu ディストリビューションの場合は、Ubuntu 用 Linux Virtual Delivery Agent のインストール の手順に従ってください。

> SSSD とは？

SSSD はシステムデーモンです。その主な機能は、システムのキャッシュとオフラインサポートを提供できる共通のフレームワークを通じて、リモートリソースを識別および認証するためのアクセスを提供することです。PAM と NSS の両方のモジュールを提供し、将来的には拡張ユーザー情報のための D-BUS ベースのインターフェイスをサポートできます。また、ローカルユーザーアカウントと拡張ユーザーデータを保存するためのより優れたデータベースも提供します。

NIS と AD の統合

NIS を AD と統合するには、次の手順を実行します。

1. Linux VDA を NIS クライアントとして追加する
2. ドメインに参加し、Samba を使用してホストキータブを作成する
3. SSSD をセットアップする
4. NSS/PAM を構成する
5. Kerberos 構成を確認する
6. ユーザー認証を確認する

Linux VDA を NIS クライアントとして追加する

NIS クライアントを構成する:

```
yum –y install ypbind rpcbind oddjob-mkhomedir
```

NIS ドメインを設定する:

```
ypdomainname nis.domain

echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

NIS サーバーとクライアントの IP アドレスを **/etc/hosts** に追加する:

```
{NIS server IP address}   server.nis.domain nis.domain
```

`authconfig` を使用して NIS を構成する:

```
sudo authconfig --enablenis --nisdomain=nis.domain --
nisserver=server.nis.domain --enablemkhomedir --update
```

nis.domain は NIS サーバーのドメイン名を表します。**server.nis.domain** は NIS サーバーのホスト名であり、NIS サーバーの IP アドレスでもかまいません。

NIS サービスを構成する:

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS 構成が正しいことを確認する:

```
ypwhich
```

アカウント情報が NIS サーバーから利用可能であることを検証する:

```
getent passwd nisaccount
```

注:

nisaccount は NIS サーバー上の実際の NIS アカウントを表します。UID、GID、ホームディレクトリ、およびログインシェルが正しく構成されていることを確認してください。

Samba を使用してドメインに参加し、ホスト keytab を作成する

SSSD は、ドメインへの参加やシステム keytab ファイルの管理のための AD クライアント機能を提供しません。これらの機能を実現するには、いくつかの方法があります。

- `adcli`
- `realmd`
- `Winbind`
- `Samba`

このセクションの情報は Samba のアプローチのみを説明しています。`realmd` については、RHEL または CentOS ベンダーのドキュメントを参照してください。これらの手順は SSSD を構成する前に実行する必要があります。

Samba を使用してドメインに参加し、ホスト keytab を作成する:

適切に構成されたファイルを持つ Linux クライアントで:

- /etc/krb5.conf
- /etc/samba/smb.conf:

SambaおよびKerberos認証用にマシンを構成します。

```
sudo authconfig --smbsecurity=ads --smbworkgroup=domain --smbrealm=REALM
--krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --update
```

ここで、**REALM** は大文字のKerberosレルム名であり、**domain** はドメインのNetBIOS名です。

KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、前のコマンドに次の2つのオプションを追加します。

`--enablekrb5kdcdns --enablekrb5realmdns`

/etc/samba/smb.conf を開き、**authconfig** ツールによって生成されたセクションの後に、**\[Global\]** セクションの下に次のエントリを追加します。

`kerberos method = secrets and keytab`

`winbind offline logon = no`

Windowsドメインに参加するには、ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つADユーザーアカウントが必要です。

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

`REALM` は大文字のKerberosレルム名であり、`user` はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSDをセットアップする

SSSDのセットアップは、次の手順で構成されます。

- Linuxクライアントマシンに **sssd-ad** および **sssd-proxy** パッケージをインストールします。
- さまざまなファイル（例：**sssd.conf**）に構成変更を加えます。
- **sssdサービス** を開始します。

/etc/sssd/sssd.conf

sssd.conf の構成例（必要に応じてオプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = EXAMPLE

services = nss, pam



[domain/EXAMPLE]

# Uncomment if you need offline logins

# cache_credentials = true

re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?
P<domain>.+$))|(^(?P<name>[^@\\]+)$))

id_provider = proxy

proxy_lib_name = nis

auth_provider = ad
```

```
access_provider = ad



# Should be specified as the long version of the Active Directory domain.

ad_domain = EXAMPLE.COM



# Kerberos settings

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U



# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



# Comment out if the users have the shell and home dir set on the AD side

default_shell = /bin/bash

fallback_homedir = /home/%d/%u



# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available

# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

「**ad.domain.com**、**server.ad.example.com**」を対応する値に置き換えます。詳細については、sssd-ad(5) - Linux man pageを参照してください。

sssd.conf のファイル所有権と権限を設定します。

`chown root:root /etc/sssd/sssd.conf`

`chmod 0600 /etc/sssd/sssd.conf` `restorecon /etc/sssd/sssd.conf`

NSS/PAM の構成

RHEL/CentOS:

authconfig を使用して SSSD を有効にします。ホームディレクトリの作成が SELinux と互換性があることを確認するために、**oddjob-mkhomedir** をインストールします。

```
authconfig --enablesssd --enablesssdauth --enablemkhomedir --update



sudo systemctl start sssd



sudo systemctl enable sssd
```

ヒント:

Linux VDA の設定を構成する際、SSSD には Linux VDA クライアント用の特別な設定がないことを考慮してください。**ctxsetup.sh** スクリプトの追加ソリューションについては、デフォルト値を使用してください。

Kerberos 構成の検証

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos **kinit** コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT チケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist -ke
```

ユーザー認証の検証

getent コマンドを使用して、ログオン形式がサポートされているか、NSS が機能しているかを確認します。

```
sudo getent passwd DOMAIN\\username
```

DOMAIN パラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOSサフィックス形式: `username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、ドメインユーザーアカウントを使用してLinux VDAにログオンします。そのドメインユーザーアカウントは以前に使用されたことがないものとします。

```
sudo ssh localhost –l DOMAIN\\username



id -u
```

コマンドによって返された**uid**に対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_{uid}
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効で期限切れになっていないことを確認します。

```
klist
```

アプリケーションの公開

Linux VDAバージョン7.13で、Citrixはサポートされているすべてのシームレスアプリケーション機能を追加しました。この機能を使用するために、特定のインストール手順は必要ありません。

ヒント:

Linux VDAバージョン1.4で、Citrixは非シームレス公開アプリケーションとセッション共有のサポートを追加しました。

Citrix Studioを使用したアプリケーションの公開

デリバリーグループを作成するとき、または既存のデリバリーグループにアプリケーションを追加するときに、Linux VDAにインストールされているアプリケーションを公開できます。このプロセスは、Windows VDAにインストールされているアプリケーションを公開するのと同様です。詳しくは、Citrix Virtual Apps and Desktopsのドキュメント（使用しているCitrix Virtual Apps and Desktopsのバージョンに基づく）を参照してください。

注:

- デリバリーグループを構成するときは、デリバリーの種類が**デスクトップとアプリケーション**または**アプリケーション**に設定されていることを確認してください。
- アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。この問題に対処するため、アプリとデスクトップの配信には個別のデリバリーグループを作成することをお勧めします。

- シームレスアプリケーションを使用するには、StoreFrontでシームレスモードを無効にしないでください。シームレスモードはデフォルトで有効になっています。「TWIMode=Off」を設定してすでに無効にしている場合は、「TWIMode=On」に変更するのではなく、この設定を削除してください。そうしないと、公開されたデスクトップを起動できない可能性があります。

制限事項

Linux VDAは、単一ユーザーによる同じアプリケーションの複数の同時インスタンスの起動をサポートしていません。

アプリセッションでは、アプリケーションに固有のショートカットのみが期待どおりに機能します。

既知の問題

アプリケーションの公開中に、以下の既知の問題が確認されています。

- 非矩形ウィンドウはサポートされていません。ウィンドウの角にサーバー側の背景が表示される場合があります。
- 公開されたアプリケーションのウィンドウコンテンツのプレビューはサポートされていません。
- 現在、シームレスモードは以下のウィンドウマネージャーをサポートしています：Mutter、Metacity、およびCompiz (Ubuntu 16.04)。Kwinおよびその他のウィンドウマネージャーはサポートされていません。ウィンドウマネージャーがサポートされているものに設定されていることを確認してください。
- 複数のLibreOfficeアプリケーションを実行すると、これらのアプリケーションはプロセスを共有するため、最初に起動されたものだけがCitrix Studioに表示されます。
- 「Dolphin」のような公開されたQt5ベースのアプリケーションでは、アイコンが表示されない場合があります。この問題を解決するには、`<https://wiki.archlinux.org/title/Qt>`の記事を参照してください。
- 同じICA®セッションで実行されている公開アプリケーションのすべてのタスクバーボタンは、同じグループにまとめられます。この問題を解決するには、タスクバーのプロパティを「タスクバーボタンを結合しない」に設定します。

Linux VDAのセルフアップデート

この機能は、Linux VDAソフトウェアを即時またはスケジュールされた時間に自動的に更新するのに役立ちます。これは、Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成する場合に特に役立ちます。AzureのVMに対する管理者権限はありません。詳細については、「Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成する」を参照してください。

構成

この機能を使用するには、次の手順を完了します。

手順1：更新情報と新しいVDAパッケージをAzureコンテナにアップロードする

手順1a：Azureストレージアカウントの下にコンテナを作成し、コンテナのアクセスレベルを**Blob (Anonymous read access for blobs only)**に設定します。

> **注:**
>
> AzureコンテナとBLOBは、お客様が排他的に保持および管理します。Citrix®は、それらに関するセキュリティ問題について責任を負いません。データセキュリティとコスト効率を確保するため、各セルフアップデート後にコンテナのアクセスレベルを**Private (no anonymous access)**に設定してください。

手順1b：VDAの更新情報をUpdateInfo.jsonという名前のJSONファイルに組み込みます。ファイル形式の例については、次のブロックを参照してください。

```
{

 "Version": "21.04.200.4",

"Distributions":[

{

"TargetOS": "RHEL7_9",

"PackageName": "",

"PackageHash": ""
```

```
},

{

"TargetOS": "RHEL8_3",

"PackageName": "XenDesktopVDA-21.04.200.4-1.el8_x.x86_64.rpm",

"PackageHash":
"a6f2aba23b84bbc3a4640294a8bb92474e0cacbab1e5ae33416c0a4473a28d73"

},

{

"TargetOS": "UBUNTU16_04",

"PackageName": "",

"PackageHash": ""

},

{

"TargetOS": "UBUNTU18_04",

"PackageName": "xendesktopvda_21.04.200.4-1.ubuntu18.04_amd64.deb",

"PackageHash":
"4148cc3f25d3717e3cbc19bd953b42c72bd38ee3fcd7f7034c2cd6f2b15b3c5a"

},

{

"TargetOS": "UBUNTU20_04",

"PackageName": "",

"PackageHash": ""

}
```

```
]

}
```

ここで、**"Version"は新しいVDAバージョンを示し、"Distributions"**は更新オブジェクトの配列です。各オブジェクトには3つの項目が含まれます。

- **"TargetOS"**：「RHEL7_9」（RHEL 7およびCentOS 7の場合）、「RHEL8_3」、「UBUNTU16_04」、「UBUNTU18_04」、または「UBUNTU20_04」である必要があります。`ctxmonitorservice`は他のディストリビューションを認識しません。
- **"PackageName"**：指定されたバージョンのVDAパッケージのフルネーム。
- **"PackageHash"**：`shasum -a 256 <pkgname>`コマンドを使用して計算するSHA-256値。

手順1c：JSONファイルと新しいバージョンのLinux VDAパッケージをAzureコンテナにアップロードします。

手順2：マスターイメージまたは各VDAでセルフアップデート機能を有効にする

デフォルトでは、セルフアップデートは無効になっています。Citrix Virtual Apps and Desktops™ Standard for AzureでLinux VDAを作成する場合、機能の有効化はマスターイメージで行う必要があります。それ以外の場合は、各ターゲットVDAで直接機能を有効にします。

セルフアップデートを有効にするには、HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\SelfUpdateにあるレジストリキーを編集するために、次のようなコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_DWORD"
-v "fEnabled" -d "0x00000001" –force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"ScheduledTime" -d "Immediately" –force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"Url" -d "<Your-Azure-Container-Url>" –force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"CaCertificate" -d "<Local-Certificate-Path-of-PortalAzureCom>" --force
```

次の表は、レジストリ設定について説明しています。

レジストリ設定	説明
fEnabled	この設定は必須です。デフォルトでは値は0で、セルフアップデートが無効であることを意味します。セルフアップデートを有効にするには、1に設定します。
Url	この設定は必須です。更新情報と新しいVDAパッケージを取得するためのAzureコンテナのURLを設定します。
ScheduledTime	この設定は必須です。**Immediately**または**NextStart**に設定できます。**Immediately**は、VDAパッケージのダウンロード後すぐにアップデートを実行することを意味します。このオプションは、ダウンロード速度が速く、アップデートが緊急である場合に適しています。ただし、パッケージのダウンロード中にライブセッションがある場合、ユーザーエクスペリエンスを妨げる可能性があります。**NextStart**は、`ctxmonitorservice`の次回の起動時にアップデートを実行することを意味します。このオプションは、ダウンロード速度が速くなく、アップデートが緊急でない場合に適しています。
CaCertificate	この設定はオプションです。Azureコンテナの URLを検証するためのPEM証明書のフルパスを設定します。Azure BLOBの場合、ブラウザから取得し、PEMに変換されたportal.azure.comの証明書を使用できます。セキュリティのため、このレジストリ設定を追加することをお勧めしますが、Ubuntuでのみサポートされています。RHELでは、`curl`コ

レジストリ設定	説明
	マンド用のNSSライブラリの一部がリンクされていません。証明書の権限を最小限に設定するようにしてください。

`ctxmonitorservice`が再起動すると、まず**Url**を照会してUpdateInfo.jsonファイルを取得し、JSONファイルから更新バージョンを取得します。次に、`ctxmonitorservice`は更新バージョンを現在のバージョンと比較します。現在のバージョンが古い場合、サービスはAzureからVDAパッケージの新しいバージョンをダウンロードし、ローカルに保存します。その後、**ScheduledTime**の設定に従ってアップデートを実行します。オンプレミス展開の場合、`ctxmonitorservice`を直接再起動してアップデートをトリガーできます。ただし、VMに対する管理者権限がないCitrix Virtual Apps and Desktops Standard for Azureでは、`ctxmonitorservice`はVDAマシンが再起動された後にのみ再起動できます。アップデートが失敗した場合、VDAは既存のバージョンにロールバックされます。

注:

- マスターイメージで構成したレジストリ設定は変更できません。
- 環境内のすべてのVMが同時にパッケージをダウンロードすると、ローカルネットワークが混雑する可能性があります。
- アップデートとロールバックの両方が失敗した場合、ユーザーデータは失われます。
- アップデートが失敗してもロールバックが成功した場合、同じネットワーク上のユーザーは異なるバージョンのLinux VDAを使用する可能性があります。このケースは最適ではありません。
- アップデートの完了には通常数分かかります。Citrix Studio にはステータスインジケーターはありません。

リモートPCアクセス

概要

Remote PC Access は Citrix Virtual Apps and Desktops の拡張機能です。これにより、組織は従業員が物理的なオフィス PC に安全にリモートでアクセスできるように簡単に設定できます。ユー

ザーがオフィス PC にアクセスできれば、仕事に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

Remote PC Access は、仮想デスクトップとアプリケーションを提供する Citrix Virtual Apps™ and Desktops と同じコンポーネントを使用します。Remote PC Access の展開と構成の要件とプロセスは、仮想リソースを提供するために Citrix Virtual Apps and Desktops を展開する際に必要な要件とプロセスと同じです。この統一性により、一貫性のある統合された管理エクスペリエンスが提供されます。ユーザーは、Citrix HDX を使用してリモートオフィス PC セッションを配信することで、最高のユーザーエクスペリエンスを得られます。

詳細については、Citrix Virtual Apps and Desktops ドキュメントの Remote PC Access を参照してください。

考慮事項

以下の考慮事項は、Linux VDAに固有のものです。

- 物理マシンでは、Linux VDA は非 3D モードでのみ使用してください。NVIDIA のドライバーの制限により、HDX™ 3D モードが有効になっている場合、PC のローカル画面をブラックアウトできず、セッションのアクティビティが表示されます。この画面が表示されることは、潜在的なセキュリティリスクです。
- 物理Linuxマシンには、シングルセッションOSタイプのマシンカタログを使用します。
- Linux マシンでは、自動ユーザー割り当ては利用できません。自動ユーザー割り当てでは、ユーザーが PC にローカルでログオンすると、自動的にマシンに割り当てられます。このログオンは管理者の介入なしに行われます。クライアントデバイスで実行されている Citrix Workspace™ アプリは、Remote PC Access デスクトップセッション内で、オフィス PC 上のアプリケーションとデータへのアクセスをユーザーに提供します。
- ユーザーがすでにローカルでPCにログオンしている場合、StoreFront™からPCを起動しようとすると失敗します。
- 省電力オプションはLinuxマシンでは利用できません。

構成

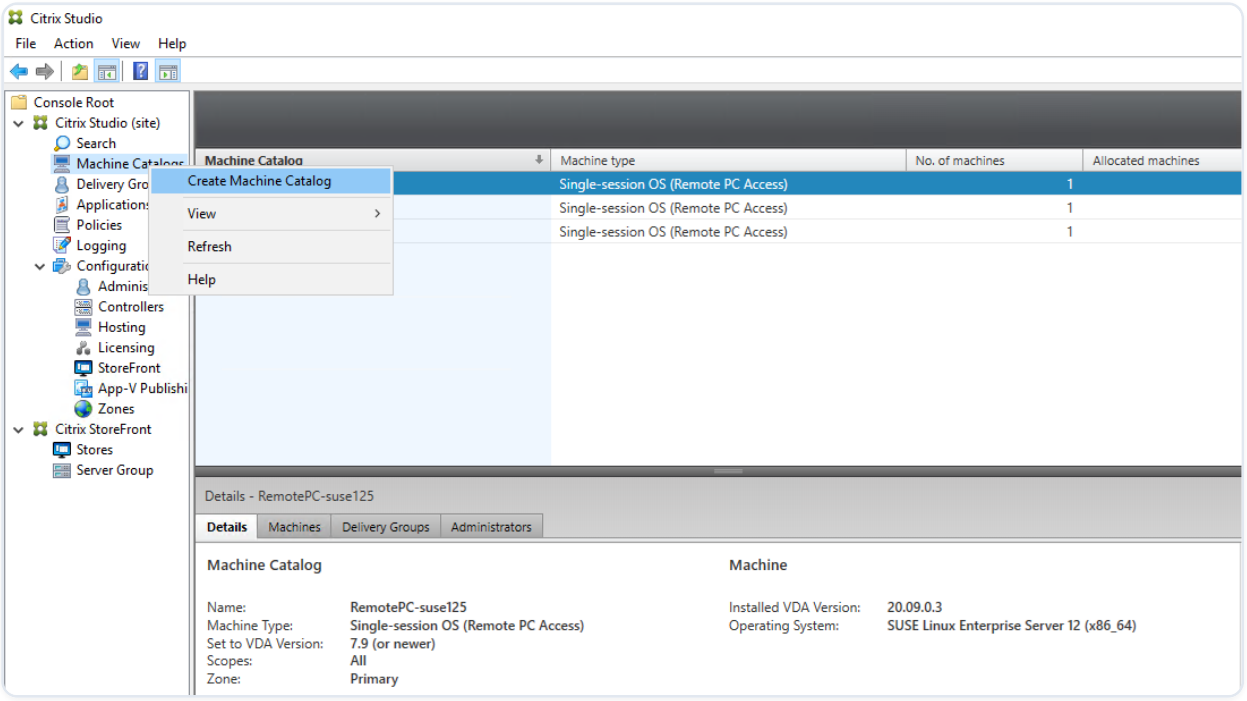
Linux PC セッションを配信するには、ターゲット PC に Linux VDA をインストールし、**Remote PC Access** タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内の PC を利用できるようにデリバリーグループを作成します。以下のセクションで手順を詳しく説明します。

ステップ 1 - ターゲット PC に Linux VDA をインストールする

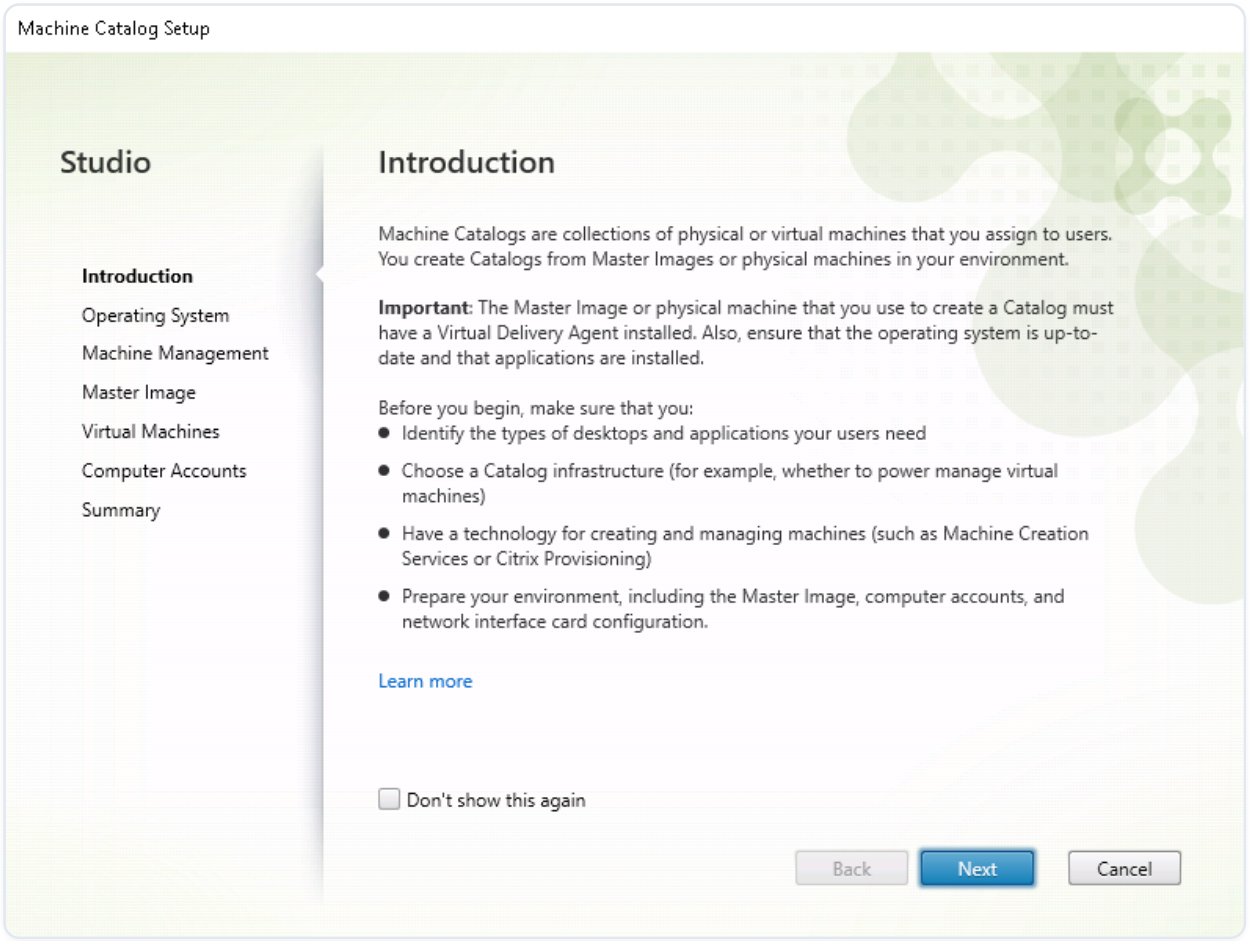
Linux VDA のインストールには、easy install を使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE` 変数の値を `Y` に設定します。

ステップ 2 - Remote PC Access タイプのマシンカタログを作成する

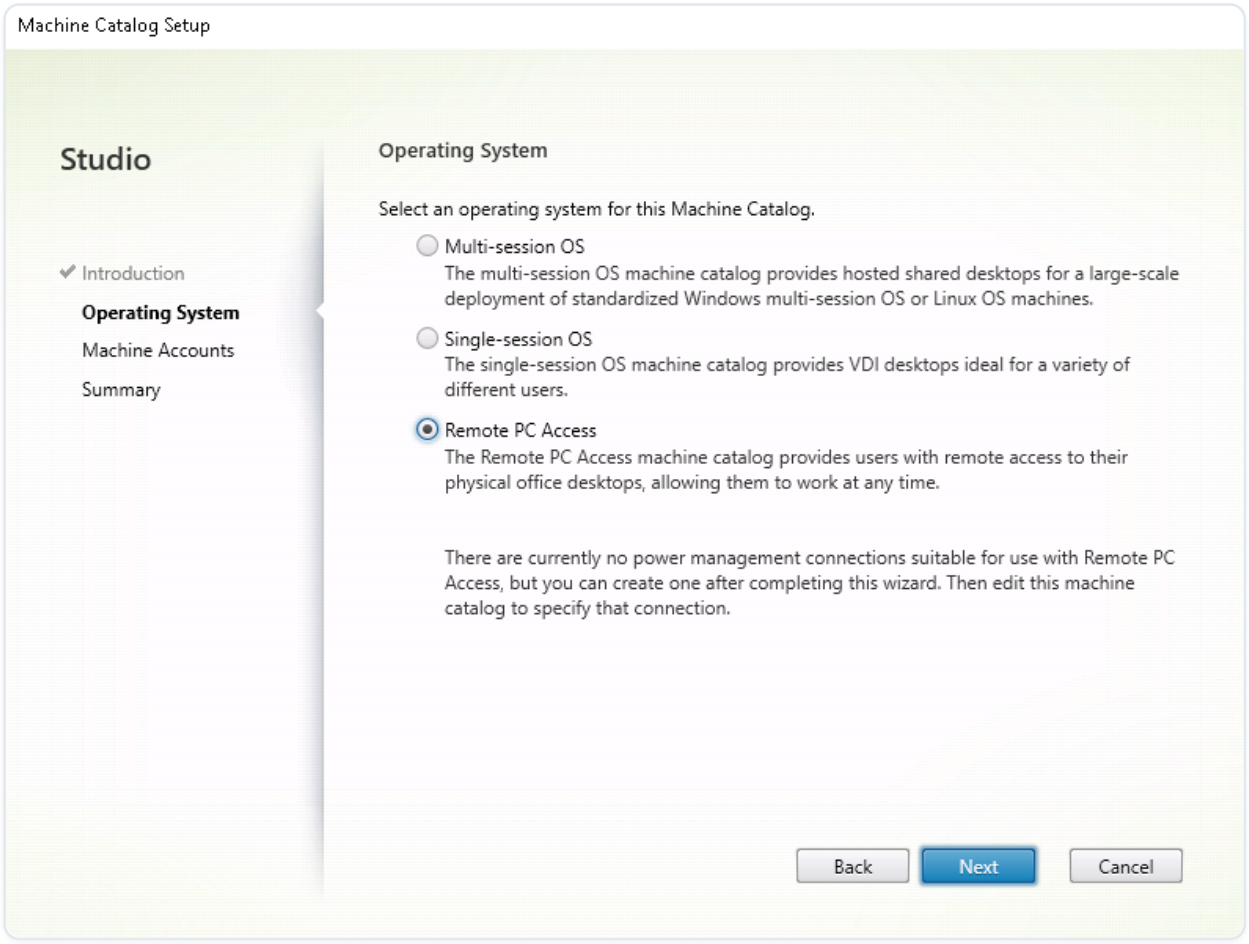
1. Citrix Studio で、**Machine Catalogs** を右クリックし、ショートカットメニューから **Create Machine Catalog** を選択します。



2. **Introduction** ページで **Next** をクリックします。



3. **Operating System** ページで **Remote PC Access** を選択します。



4. 「**OUの追加**」をクリックしてターゲットPCを含むOUを選択するか、「**マシンアカウントの追加**」をクリックして個々のマシンをマシンカタログに追加します。

   OUまたはマシンアカウントを追加する画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-4-add-ous-or-add-machine-accounts.png)

5. マシンカタログに名前を付けます。

   マシンカタログに名前を付ける画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-5-name-the-machine-catalog.png)

6. （オプション）マシンカタログを右クリックして、関連する操作を実行します。

   マシンカタログでの関連操作の画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-6-edit-a-machine-catalog.png)

ステップ3 - マシンカタログ内のPCにアクセスを要求するユーザーが利用できるようにデリバリーグループを作成する

1. Citrix Studioで、**デリバリーグループ**を右クリックし、ショートカットメニューから**デリバリーグループの作成**を選択します。

デリバリーグループの作成を選択する画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-11-create-a-delivery-group.png)

2. **デリバリーグループの概要**ページで**次へ**をクリックします。

   デリバリーグループの概要ページの画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-22-getting-started-with-delivery-groups.png)

3. ステップ2で作成したマシンカタログを選択し、デリバリーグループに関連付けます。

   マシンカタログをデリバリーグループに関連付ける画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-33-assocatiing-the-machine-catalog-with-the-delivery-group.png)

4. マシンカタログ内のPCにアクセスできるユーザーを追加します。追加したユーザーは、クライアントデバイス上のCitrix Workspaceアプリを使用して、PCにリモートでアクセスできます。

   マシンカタログ内のPCにアクセスできるユーザーを追加する画像(/en-us/linux-virtual-delivery-agent/2106/media/remote-pc-access-44-add-uses-to-access-pcs.png)

Wake on LAN (ウェイク・オン・ラン)

Remote PC AccessはWake on LANをサポートしており、ユーザーは物理PCをリモートで起動できます。この機能により、ユーザーは使用していないときにオフィスPCの電源を切ったままにして、エネルギーコストを節約できます。また、誤ってマシンの電源が切れてしまった場合でも、リモートアクセスを可能にします。

Wake on LAN機能を使用すると、デリバリーコントローラーからの指示があった場合、PC上で実行されているVDAから、そのPCが存在するサブネットにマジックパケットが直接送信されます。これにより、追加のインフラストラクチャコンポーネントやマジックパケット配信のためのサードパーティソリューションに依存することなく、この機能が動作します。

Wake on LAN機能は、レガシーなSCCMベースのWake on LAN機能とは異なります。SCCMベースのWake on LANの詳細については、「Wake on LAN – SCCM統合」を参照してください。

システム要件

Wake on LAN機能を使用するためのシステム要件は次のとおりです。

- コントロールプレーン:
  - シトリックス バーチャルアプリおよびデスクトップ サービス
  - シトリックス バーチャルアプリおよびデスクトップ 2012 以降
- 物理PC:

- VDAバージョン2012以降
- BIOSおよびNICでWake on LANが有効になっていること

Wake on LANの構成

現在、統合されたWake on LANの構成は、PowerShellを使用してのみサポートされています。

Wake on LANを構成するには:

1. まだ Remote PC Access マシンカタログがない場合は、作成します。
2. まだ Wake on LAN ホスト接続がない場合は、作成します。

> **注:**
>
> Wake on LAN機能を使用するには、「Microsoft Configuration Manager Wake on LAN」タイプのホスト接続がある場合、ホスト接続を作成します。

3. Wake on LANホスト接続の一意の識別子を取得します。
4. Wake on LAN ホスト接続をマシンカタログに関連付けます。

   Wake on LAN ホスト接続を作成するには：

```
# Load Citrix SnapIns

Add-PSSnapIn -Name "*citrix*"



# Provide the name of the Wake on LAN host connection

[string]$connectionName = "Remote PC Access Wake on LAN"



# Create the hypervisor connection

$hypHc = New-Item -Path xdhyp:\Connections `

            -Name $connectionName `

            -HypervisorAddress "N/A" `
```

```
            -UserName "woluser" `

            -Password "wolpwd" `

            -ConnectionType Custom `

            -PluginId VdaWOLMachineManagerFactory `

            -CustomProperties "<CustomProperties></CustomProperties>" `

            -Persist



$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid



# Wait for the connection to be ready before trying to use it

while (-not $bhc.IsReady)

{

    Start-Sleep -s 5

    $bhc = Get-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid

}
```

ホスト接続の準備ができたら、次のコマンドを実行してホスト接続の一意の識別子を取得します。

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"

$hypUid = $bhc.Uid
```

接続の一意の識別子を取得したら、次のコマンドを実行して、その接続をRemote PC Access マシンカタログに関連付けます。

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog
-RemotePCHypervisorConnectionUid $hypUid
```

5. マシンカタログ内の各VMで、BIOSおよびNICでWake on LANを有効にします。

   注： Wake on LANを有効にする方法は、マシンの構成によって異なります。

   - BIOSでWake on LANを有効にするには：

     a. BIOSに入り、Wake on LAN機能を有効にします。

        BIOSにアクセスする方法は、マザーボードの製造元と、製造元が選択したBIOSベンダーによって異なります。

     b. 設定を保存し、マシンを再起動します。

   - NICでWake on LANを有効にするには：

     a. NICがマジックパケットをサポートしているかどうかを確認するには、`sudo ethtool <NIC>`コマンドを実行します。

        `<NIC>`はNICのデバイス名です（例：`eth0`）。`sudo ethtool <NIC>`コマンドは、NICの機能に関する出力を提供します。

        - 出力に`Supports Wake-on: <letters>`のような行が含まれており、`<letters>`に文字`g`が含まれている場合、NICはWake on LANマジックパケット方式をサポートしています。
        - 出力に`Wake-on: <letters>`のような行が含まれており、`<letters>`に文字`g`が含まれていて、文字`d`が含まれていない場合、Wake on LANマジックパケット方式は有効になっています。ただし、`<letters>`に文字`d`が含まれている場合、Wake on LAN機能が無効になっていることを示します。この場合、`sudo ethtool -s <NIC> wol g`コマンドを実行してWake on LANを有効にします。

     b. ほとんどのディストリビューションでは、起動ごとに`sudo ethtool -s <NIC> wol g`コマンドが必要です。このオプションを永続的に設定するには、お使いのディストリビューションに基づいて次の手順を実行します。

        Ubuntu:

        インターフェイス構成ファイル`/etc/network/interfaces`に`up ethtool -s <NIC> wol g`行を追加します。例：

```
# ifupdown has been replaced by netplan(5) on this system. See

# /etc/netplan for current configuration.

# To re-enable ifupdown on this system, you can run:

# sudo apt install ifupdown

auto eth0

iface eth0 inet static

            address 10.0.0.1

            netmask 255.255.240.0

            gateway 10.0.0.1

            up ethtool -s eth0 wol g
```

RHEL/SUSE:

インターフェイス構成ファイル `/etc/sysconfig/network-scripts/ifcfg-<NIC>` に次の `ETHTOOL_OPTS` パラメーターを追加します。

```
ETHTOOL_OPTS="-s ${DEVICE} wol g"
```

設計上の考慮事項

Remote PC Access で Wake on LAN を使用する予定がある場合は、次の点を考慮してください。

- 複数のマシンカタログで同じ Wake on LAN ホスト接続を使用できます。
- ある PC が別の PC を起動するには、両方の PC が同じサブネット内にあり、同じ Wake on LAN ホスト接続を使用している必要があります。PC が同じマシンカタログ内にあるか、異なるマシンカタログ内にあるかは関係ありません。
- ホスト接続は特定のゾーンに割り当てられます。展開に複数のゾーンが含まれている場合は、各ゾーンに Wake on LAN ホスト接続が必要です。これはマシンカタログにも当てはまります。
- マジックパケットはグローバルブロードキャストアドレス255.255.255.255を使用してブロードキャストされます。このアドレスがブロックされていないことを確認してください。

- そのサブネット内のマシンを起動できるようにするには、Wake on LAN接続ごとに、サブネット内に少なくとも1台のPCが電源オンになっている必要があります。

運用上の考慮事項

Wake on LAN機能を使用する際の考慮事項は次のとおりです。

- 統合されたWake on LAN機能を使用してPCを起動できるようにするには、VDAが少なくとも1回登録されている必要があります。
- Wake on LANはPCを起動するためにのみ使用できます。再起動やシャットダウンなどの他の電源操作はサポートしていません。
- Wake on LAN接続が作成されると、Studioに表示されます。ただし、Studio内でそのプロパティを編集することはサポートされていません。
- マジックパケットは次の2つの方法のいずれかで送信されます。
  - ユーザーが自分のPCへのセッションを起動しようとし、VDAが未登録の場合
  - 管理者がStudioまたはPowerShellから手動で電源オンコマンドを送信した場合
- デリバリーコントローラーはPCの電源状態を認識しないため、Studioでは電源状態の下に**サポートされていません**と表示されます。デリバリーコントローラーは、VDAの登録状態を使用してPCがオンかオフかを判断します。

その他のリソース

以下は Remote PC Access に関するその他のリソースです。

- ソリューション設計ガイダンス: Remote PC Access Design Decisions。
- Remote PC Access アーキテクチャの例: Citrix Remote PC Access ソリューションのリファレンスアーキテクチャ。

印刷

この記事では、印刷のベストプラクティスに関する情報を提供します。

インストール

Linux VDAには、**cups**フィルターと**foomatic**フィルターの両方が必要です。これらのフィルターは、VDAのインストール時にインストールされます。ディストリビューションに基づいて、フィルターを手動でインストールすることもできます。例：

RHEL 7の場合：

```
sudo yum –y install cups



sudo yum -y install foomatic-filters
```

構成

Citrixが提供するユニバーサルプリンタードライバーには、3種類（postscript、pcl5、pcl6）あります。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合があります。この場合、以前のリリースでは、~/.CtxlpProfile$CLIENT_NAME構成ファイルを編集するしかありませんでした。バージョン1906以降では、代わりにCitrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成できます。

Citrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成するには：

1. **プリンタードライバーのマッピングと互換性**ポリシーを選択します。
2. 「**追加**」をクリックします。
3. **ドライバー名**にクライアントプリンターのドライバー名を入力します。Citrix Workspace™ app for Linuxを使用している場合は、代わりにプリンター名を入力します。

   4\. **次で置換**を選択し、VDA上のドライバーファイルの絶対パスを入力します。

   プリンタードライバーのマッピングと互換性ポリシーの画像(/en-us/linux-virtual-delivery-agent/2106/media/printer-driver-mapping-and-compatibili-policy1.png)

注:

- PPDドライバーファイルのみがサポートされています。
- **プリンタードライバーのマッピングと互換性**ポリシーのその他のオプションはサポートされていません。**次で置換**のみが有効になります。

使用方法

公開デスクトップと公開アプリケーションの両方から印刷できます。クライアント側のデフォルトプリンターのみがLinux VDAセッションにマップされます。プリンター名はデスクトップとアプリケーションで異なります。

- 公開デスクトップの場合：

  `CitrixUniversalPrinter:$CLIENT_NAME:dsk$SESSION_ID`

- 公開アプリケーションの場合：

  `CitrixUniversalPrinter:$CLIENT_NAME:app$SESSION_ID`

> **注:**
>
> 同じユーザーが公開デスクトップと公開アプリケーションの両方を開くと、両方のプリンターがセッションで利用可能になります。公開アプリケーションセッションでデスクトッププリンターに印刷すること、または公開デスクトップでアプリケーションプリンターに印刷することは失敗します。

トラブルシューティング

印刷できません

印刷が正しく機能しない場合は、印刷デーモンである**ctxlpmngt**とCUPSフレームワークを確認してください。

印刷デーモンである**ctxlpmngt**はセッションごとのプロセスであり、セッションの期間中実行されている必要があります。印刷デーモンが実行されていることを確認するには、次のコマンドを実行します。**ctxlpmngt**が実行されていない場合は、コマンドラインから手動で**ctxlpmngt**を開始してください。

```
ps –ef | grep ctxlpmngt
```

印刷がまだ機能しない場合は、CUPSフレームワークを確認してください。**ctxcups**サービスはプリンター管理に使用され、Linux CUPSフレームワークと通信します。これはマシンごとに1つのプロセスであり、次のコマンドを実行して確認できます。

```
service ctxcups status
```

CUPSログを収集するための追加の手順

CUPSログを収集するには、CUPSサービスファイルを構成するために次のコマンドを実行します。そうしないと、CUPSログは**hdx.log**に記録できません。

```
sudo service cups stop


sudo vi /etc/systemd/system/printer.target.wants/cups.service


PrivateTmp=false


sudo service cups start


sudo systemctl daemon-reload
```

注:

この構成は、問題が発生したときに完全な印刷ログを収集するためだけに作成されます。通常の状況では、この構成はCUPSのセキュリティを破るため、推奨されません。

印刷出力が文字化けする

互換性のないプリンタードライバーは、文字化けした出力を引き起こす可能性があります。ユーザーごとのドライバー構成が利用可能であり、**~/.CtxlpProfile$CLIENT_NAME**構成ファイルを編集することで構成できます。

```
[DEFAULT_PRINTER]


printername=


model=
```

```
ppdpath=



drivertype=
```

重要:

printernameは、現在のクライアント側のデフォルトプリンターの名前を含むフィールドです。これは読み取り専用の値です。編集しないでください。

フィールド**ppdpath**、**model**、および**drivertype**は、マップされたプリンターに対して1つしか有効にならないため、同時に設定できません。

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、**model**=オプションを使用してネイティブプリンタードライバーのモデルを構成します。プリンターの現在のモデル名は、**lpinfo**コマンドを使用して見つけることができます。

  ```
  lpinfo –m



  …



  xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0



  xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

  xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
  ```

  その後、プリンターに合わせてモデルを設定できます。

```
model=xerox/ph3115.ppd.gz
```

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、ネイティブプリンタードライバーのPPDファイルパスを構成します。**ppdpath**の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  例えば、/home/tester/NATIVE_PRINTER_DRIVER.ppd の下に**ppdドライバー**があります。

  ```
  ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
  ```

- Citrixが提供するユニバーサルプリンタードライバーには、3つのタイプ（Postscript、PCL5、PCL6）があります。プリンターのプロパティに基づいてドライバータイプを設定できます。

  例えば、クライアントのデフォルトプリンタードライバータイプがPCL5の場合、**drivertype**を次のように設定します。

  ```
  drivertype=pcl5
  ```

出力サイズがゼロ

異なる種類のプリンターを試してください。また、CutePDFやPDFCreatorのような仮想プリンターを試して、この問題がプリンタードライバーに関連しているかどうかを確認してください。

印刷ジョブは、クライアントのデフォルトプリンターのプリンタードライバーに依存します。現在アクティブなドライバーのタイプを特定することが重要です。クライアントプリンターがPCL5ドライバーを使用しているのに、Linux VDAがPostscriptドライバーを選択すると、問題が発生する可能性があります。

プリンタードライバーのタイプが正しい場合は、次の手順を実行して問題を特定できます。

1. 公開デスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME**コマンドを実行します。
3. Linux VDAにスプールファイルを保存するために、次のフィールドを追加します。

   ```
   deletespoolfile=no
   ```

4. 構成の変更を読み込むために、ログオフして再度ログオンします。

5. ドキュメントを印刷して問題を再現します。印刷後、スプールファイルは **/var/spool/cups-ctx/$logon_user/$spool_file** に保存されます。
6. スプールが空かどうかを確認します。スプールファイルがゼロの場合、それは問題を示しています。詳細なガイダンスについては、Citrixサポートに連絡してください（印刷ログを提供してください）。
7. スプールサイズがゼロでない場合は、ファイルをクライアントにコピーします。スプールファイルの内容は、クライアントのデフォルトプリンターのプリンタードライバータイプに依存します。マップされたプリンター（ネイティブ）ドライバーがPostscriptの場合、スプールファイルはLinux OSで直接開くことができます。内容が正しいかどうかを確認します。

   スプールファイルがPCLの場合、またはクライアントOSがWindowsの場合、スプールファイルをクライアントにコピーし、別のプリンタードライバーを使用してクライアント側のプリンターで印刷します。
8. マップされたプリンターを別のプリンタードライバーを使用するように変更します。次の例では、postscriptクライアントプリンターを例として使用します。

   a. アクティブなセッションにログオンし、クライアントデスクトップでブラウザーを開きます。

   b. 印刷管理ポータルを開きます。

   ```
   localhost:631
   ```

   c. マップされたプリンター **CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID** を選択し、**プリンターの変更** を選択します。この操作には管理者権限が必要です。

   d. cups-ctx接続を維持し、「続行」をクリックしてプリンタードライバーを変更します。

   e. **製造元** および **モデル** フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。

   f. このプロセスが成功した場合、**.CtxlpProfile$CLIENT_NAME** でドライバーのPPDファイルパスを構成し、マップされたプリンターが新しく選択されたドライバーを使用できるようにします。

既知の問題

Linux VDAでの印刷中に、以下の問題が確認されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブドライバーに設定してください。

大容量ドキュメントの印刷パフォーマンスの低下

ローカルクライアントプリンターで大容量ドキュメントを印刷すると、ドキュメントはサーバー接続を介して転送されます。低速な接続では、転送に時間がかかる場合があります。

他のセッションから見えるプリンターおよび印刷ジョブの通知

Linuxには、Windowsオペレーティングシステムと同じセッションの概念がありません。そのため、すべてのユーザーがシステム全体の通知を受け取ります。これらの通知は、CUPS構成ファイル **/etc/cups/cupsd.conf** を変更することで無効にできます。

ファイルに構成されている現在のポリシー名を見つけます。

`DefaultPolicy default`

ポリシー名が *default* の場合、デフォルトポリシーのXMLブロックに以下の行を追加します。

```
<Policy default>

     # Job/subscription privacy...

     JobPrivateAccess default

     JobPrivateValues default

     SubscriptionPrivateAccess default

     SubscriptionPrivateValues default
```

```
    … …

    <Limit Create-Printer-Subscription>

        Require user @OWNER

        Order deny,allow

    </Limit>

    <Limit All>

        Order deny,allow

    </Limit>

</Policy>
```

ファイルのコピーと貼り付け

ユーザーは、セッションとローカルクライアント間でファイルをコピーして貼り付けることができます。この機能には、Citrix Virtual Apps and Desktops™ 2006以降、およびWindows用Citrix

Workspace™アプリ1903以降が必要です。コピーと貼り付けは、右クリックメニューまたはキーボードショートカットを使用して機能します。

ファイルを正常にコピーして貼り付けるには、以下を確認してください。

- ファイルの最大数が20を超えないこと。
- ファイルの最大サイズが200 MBを超えないこと。

サポートされるプラットフォーム

ファイルのコピーと貼り付け機能は、以下で利用できます。

- RHEL 7.8
- SLES 12.5
- ウブント 16.04
- ウブント 18.04
- デビアン 10

関連するポリシー

以下のクリップボードポリシーは、この機能の構成に関連しています。クリップボードポリシーの詳細については、「ポリシーサポートリスト」を参照してください。

- クライアントクリップボードリダイレクト
- クリップボード選択更新モード
- プライマリ選択更新モード

> **注:**
>
> ファイルのコピー＆ペースト機能を無効にするには、Citrix Studioで**Client clipboard redirection**ポリシーを**Prohibited**に設定します。

制限事項

- 切り取りはサポートされていません。ファイルの切り取り要求はコピーとして扱われます。
- ドラッグアンドドロップはサポートされていません。
- ディレクトリのコピーはサポートされていません。

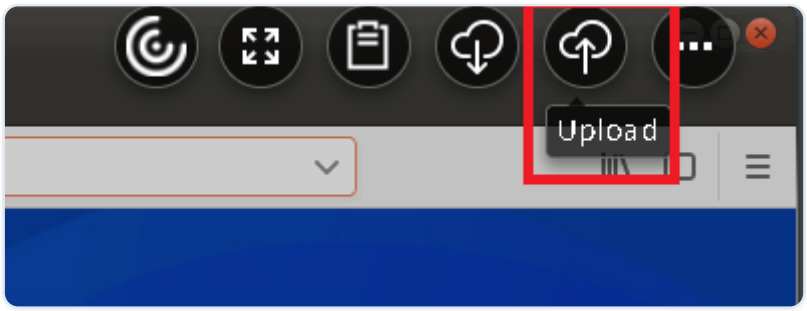
ファイル転送

Linux VDAとクライアントデバイス間でファイル転送がサポートされています。この機能は、クライアントデバイスがHTML5サンドボックス属性をサポートするWebブラウザーを実行している場合に利用できます。HTML5サンドボックス属性を使用すると、ユーザーはCitrix Workspace™アプリ（HTML5およびChrome用）を使用して仮想デスクトップとアプリにアクセスできます。

注:

ファイル転送は、Citrix Workspaceアプリ（HTML5およびChrome用）で利用できます。

公開されたアプリおよびデスクトップセッション内では、ファイル転送によりLinux VDAとクライアントデバイス間でファイルのアップロードとダウンロードが可能です。クライアントデバイスからLinux VDAにファイルをアップロードするには、Citrix Workspaceアプリのツールバーにある**アップロード**アイコンをクリックし、ファイルダイアログから目的のファイルを選択します。Linux VDAからクライアントデバイスにファイルをダウンロードするには、**ダウンロード**アイコンをクリックします。アップロードまたはダウンロード中にファイルを追加できます。一度に最大100個のファイルを転送できます。



注:

Linux VDAとクライアントデバイス間でファイルをアップロードおよびダウンロードするには、Citrix Workspaceアプリのツールバーが有効になっていることを確認してください。

ファイルをドラッグアンドドロップできるバージョンのCitrix Workspaceアプリを使用できます。

> 自動ダウンロードは、ファイル転送の機能強化です。VDA上の**Save to My Device**ディレクトリにダウンロードまたは移動したファイルは、クライアントデバイスに自動的に転送されます。

> **注:**

> 自動ダウンロードには、**デスクトップとクライアント間のファイル転送を許可する**および**デスクトップからファイルをダウンロードする**ポリシーが**許可**に設定されている必要があります。

> 自動ダウンロードのユースケースをいくつか示します。

- **Save to My Device**にファイルをダウンロードする

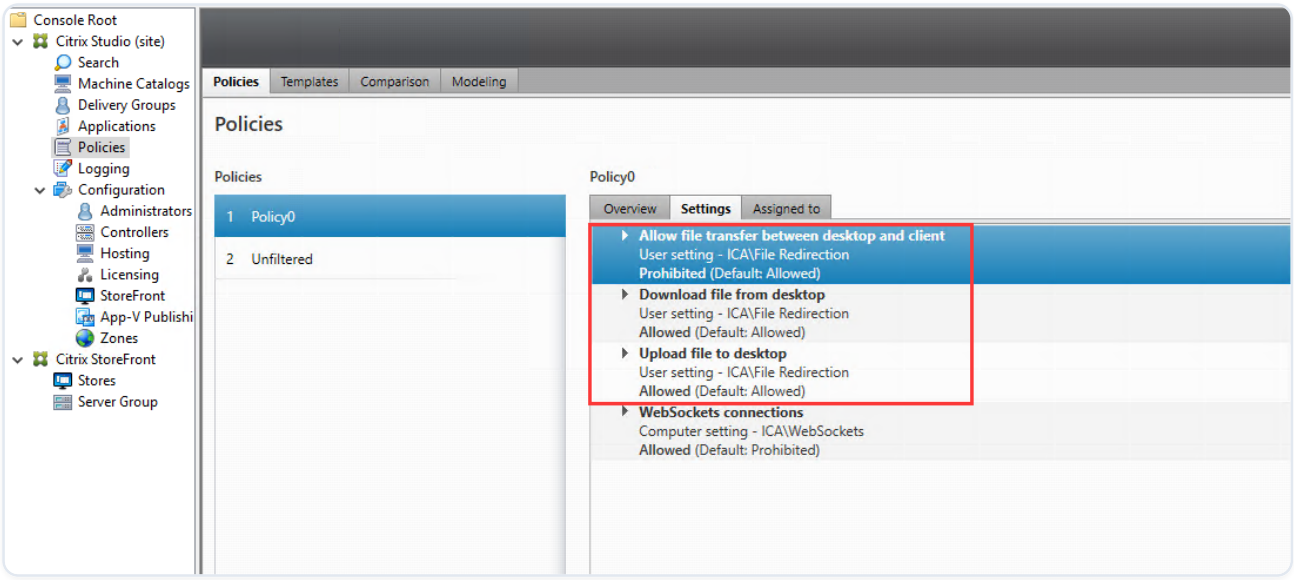
  公開されたデスクトップおよびWebブラウザーアプリセッション内では、Webサイトからダウンロードしたファイルを VDA上の**Save to My Device**ディレクトリに保存して、クライアントデバイスに自動転送できます。自動ダウンロードを実現するには、セッション内Webブラウザーのデフォルトのダウンロードディレクトリを**Save to My Device**に設定し、Citrix Workspaceアプリ（HTML5またはChrome用）を実行するWebブラウザーでローカルダウンロードディレクトリを設定します。

- ファイルを**Save to My Device**に移動またはコピーする

  公開されたデスクトップセッション内では、ターゲットファイルを選択し、それらを**Save to My Device**ディレクトリに移動またはコピーして、クライアントデバイスで利用できるようにします。

ファイル転送ポリシー

Citrix Studioを使用してファイル転送ポリシーを設定できます。デフォルトでは、ファイル転送は有効になっています。



ポリシーの説明：

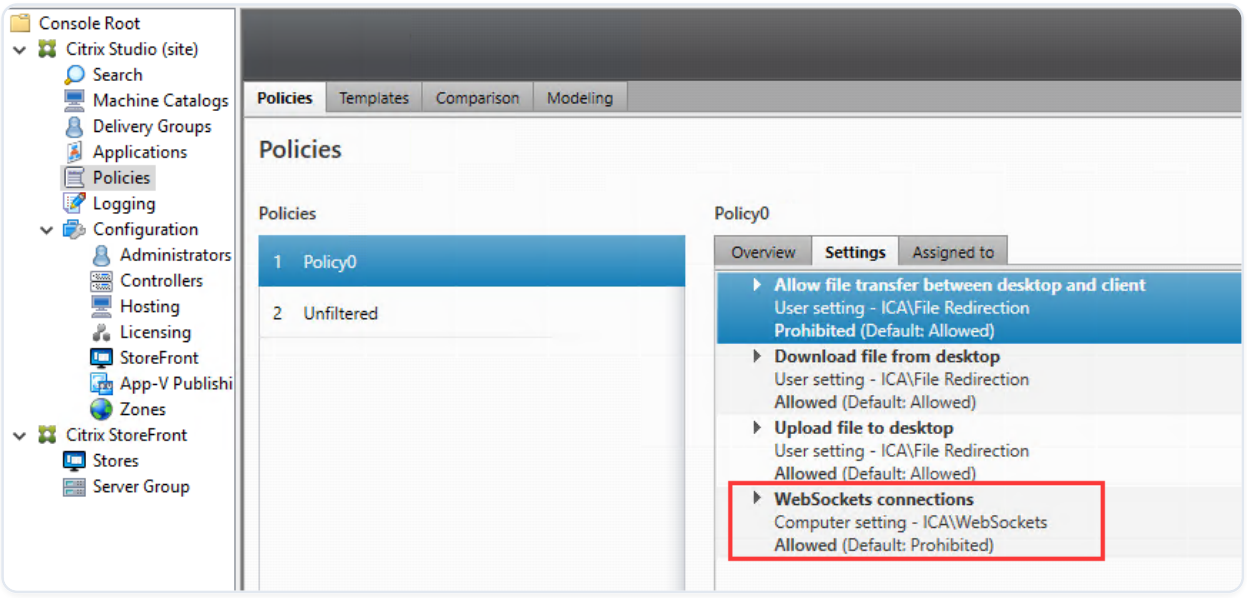
- **デスクトップとクライアント間のファイル転送を許可する。** ユーザーがCitrix Virtual Apps and Desktops™セッションとデバイス間でファイルを転送することを許可または禁止します。
- **デスクトップからファイルをダウンロードする。** ユーザーがCitrix Virtual Apps™ and Desktopsセッションからデバイスにファイルをダウンロードすることを許可または禁止します。
- **デスクトップにファイルをアップロードする。** ユーザーがデバイスからCitrix Virtual Apps and Desktopsセッションにファイルをアップロードすることを許可または禁止します。

> **注:**
>
> **デスクトップからファイルをダウンロードする**および**デスクトップにファイルをアップロードする**ポリシーを有効にするには、**デスクトップとクライアント間のファイル転送を許可する**ポリシーを**許可**に設定します。

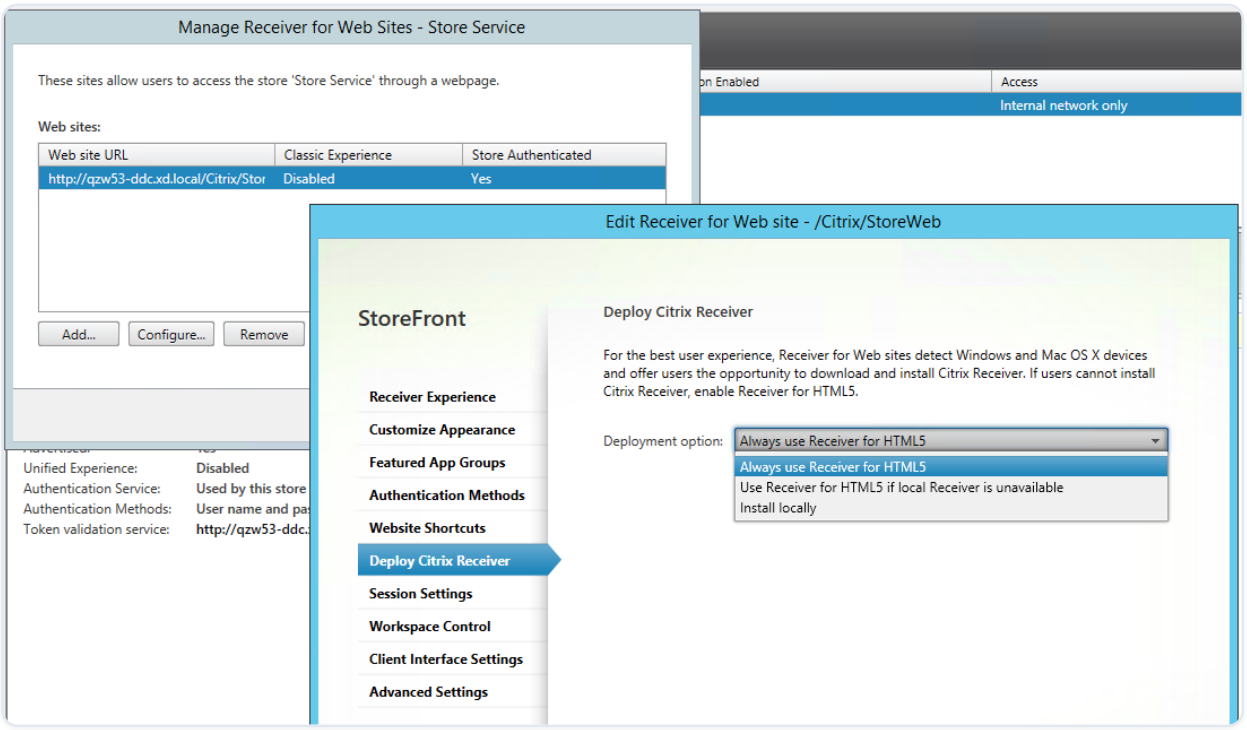
使用方法

Citrix Workspaceアプリ for HTML5を介してファイル転送機能を使用するには：

1. Citrix Studioで、**WebSockets connections**ポリシーを**許可**に設定します。



2. Citrix Studioで、前述のファイル転送ポリシーを介してファイル転送を有効にします。
3. Citrix StoreFront管理コンソールで、**Stores**をクリックし、**Manage Receiver for Web Sites**ノードを選択して、**Always use Receiver for HTML5**オプションを選択してCitrix Receiver™ for HTML5を有効にします。



4. 仮想デスクトップまたはWebブラウザアプリセッションを起動します。Linux VDAとクライアントデバイス間で1つ以上のファイル転送を実行します。

Citrix Workspaceアプリ for Chromeを介してファイル転送機能を使用するには：

1. 前述のファイル転送ポリシーを介してファイル転送を有効にします。
2. ChromeウェブストアからCitrix Workspaceアプリを入手します。

Citrix Workspaceアプリ for ChromeをChromeアプリページにすでに追加している場合は、この手順をスキップしてください。

a. Google Chromeの検索ボックスに「**Citrix Workspace for Chrome**」と入力します。検索アイコンをクリックします。
b. 検索結果の中から、Citrix Workspaceアプリが利用可能なChromeウェブストアへのURLをクリックします。

   Citrix Workspaceアプリ for Chromeの画像(/en-us/linux-virtual-delivery-agent/2106/media/workspace-app-for-chrome.png)

c. 「**Chromeに追加**」をクリックして、Citrix WorkspaceアプリをGoogle Chromeに追加します。

3. Chrome AppsページでCitrix Workspaceアプリ for Chromeをクリックします。
4. 接続するStoreFront™ストアのURLを入力します。

   以前にURLを入力した場合は、この手順をスキップします。

5. 仮想デスクトップまたはアプリセッションを起動します。Linux VDAとクライアントデバイスの間で1つ以上のファイル転送を実行します。

PDF印刷

PDF印刷をサポートするCitrix Workspace™アプリのバージョンを使用すると、Linux VDAセッション内から変換されたPDFを印刷できます。セッションの印刷ジョブは、Citrix Workspaceアプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、任意のPDFビューアを使用してPDFを開き、任意のプリンターで印刷できます。

Linux VDAは、以下のバージョンのCitrix WorkspaceアプリでPDF印刷をサポートしています。

- Citrix Receiver for HTML5バージョン2.4〜2.6.9、Citrix Workspaceアプリ1808 for HTML5以降
- Citrix Receiver for Chromeバージョン2.4〜2.6.9、Citrix Workspaceアプリ1808 for Chrome以降
- Citrix Workspaceアプリ1905 for Windows以降

構成

PDF印刷をサポートするCitrix Workspaceアプリのバージョンを使用する以外に、Citrix Studioで以下のポリシーを有効にします。

- **クライアントプリンターリダイレクト**（デフォルトで有効）

- **PDFユニバーサルプリンターの自動作成**（デフォルトで無効）

これらのポリシーを有効にすると、起動したセッション内で**印刷**をクリックしたときに、プリンターを選択するための印刷プレビューがローカルマシンに表示されます。デフォルトプリンターの設定については、Citrix Workspaceアプリのドキュメントを参照してください。

グラフィックの構成

この記事では、Linux VDAのグラフィック構成と微調整に関するガイダンスを提供します。

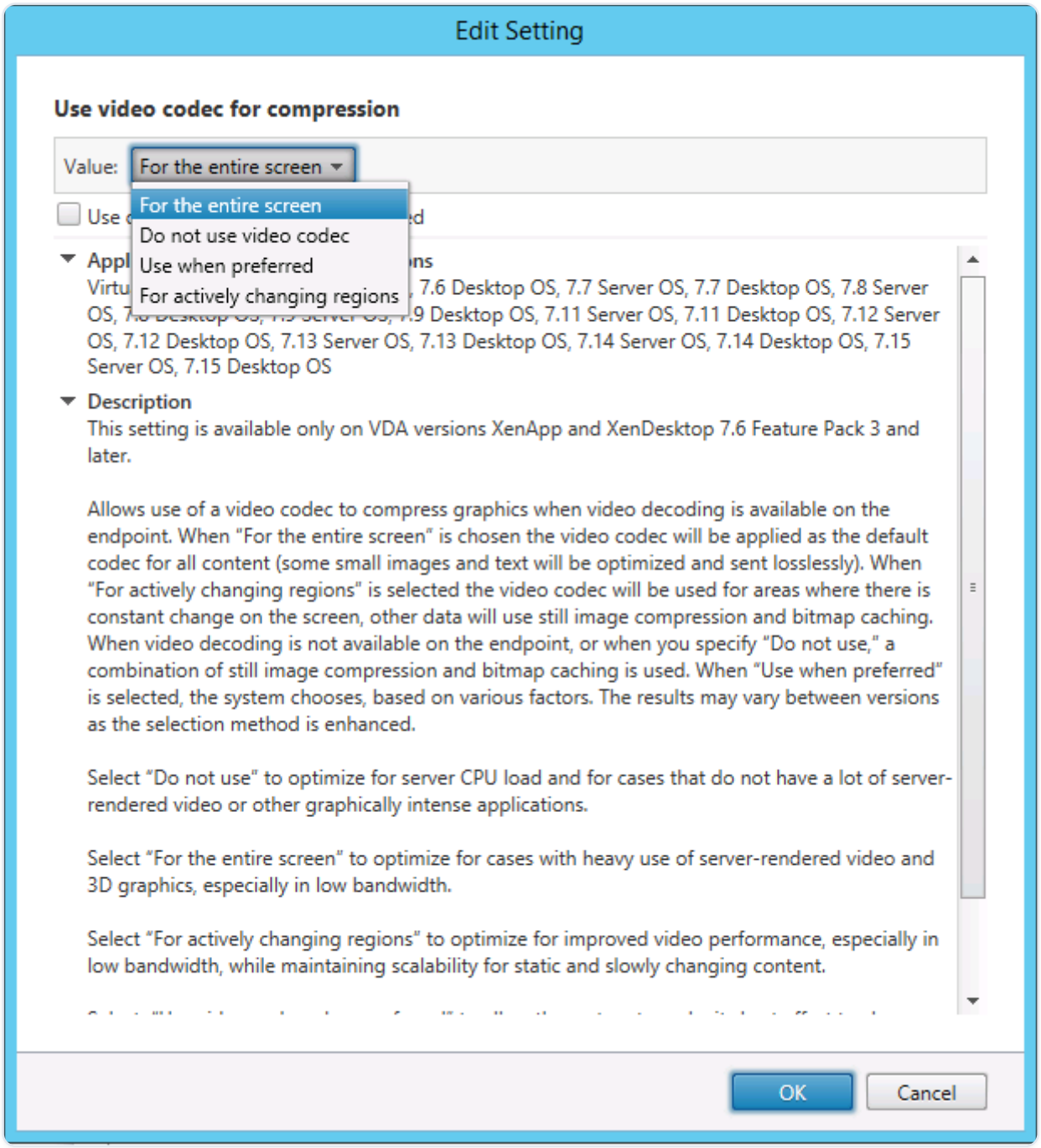
詳細については、「システム要件」および「インストール概要」セクションを参照してください。

構成

Thinwireは、Linux VDAで使用されるディスプレイリモートテクノロジーです。このテクノロジーにより、あるマシンで生成されたグラフィックを、通常はネットワーク経由で別のマシンに送信して表示できます。

「圧縮にビデオコーデックを使用」グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して次のオプションを提供します。

- **優先する場合にのみ使用**。この設定はデフォルトです。追加の構成は必要ありません。この設定を維持することで、すべてのCitrix®接続でThinwireが選択され、一般的なデスクトップワークロードのスケーラビリティ、帯域幅、および優れた画質のために最適化されます。
- **画面全体に適用**。ThinwireをフルスクリーンH.264またはH.265で提供し、特に3Dグラフィックを多用する場合に、ユーザーエクスペリエンスと帯域幅の向上を最適化します。
- **アクティブに変更される領域に適用**。Thinwireのアダプティブディスプレイテクノロジーは、動いている画像（ビデオ、移動中の3D）を識別し、画像が動いている画面の部分でのみH.264を使用します。***H.264ビデオコーデックの選択的な使用*により、HDX Thinwireは、ビデオコンテンツなど、H.264ビデオコーデックを使用して頻繁に更新される画面の一部を検出してエンコードできます。静止画圧縮（JPEG、RLE）とビットマップキャッシュは、テキストや写真画像を含む画面の残りの部分で引き続き使用されます。ユーザーは、ビデオコンテンツの帯域幅の低減と品質の向上、およびロスレステキストまたは高品質の画像を他の場所で組み合わせるという利点を得られます。この機能を有効にするには、ポリシー設定の圧縮にビデオコーデックを使用を優先する場合にのみ使用**（デフォルト）または**アクティブに変更される領域に適用**に変更します。詳細については、「グラフィックポリシー設定」を参照してください。



ディスプレイリモートのパフォーマンスを微調整するために、以下の視覚表示ポリシー設定を含む他のいくつかのポリシー設定を使用できます。

- **シンプルなグラフィックの優先色深度**
- **ターゲットフレームレート**
- **視覚的品質**

ThinwireでBuild to LosslessにH.264を使用する

デフォルトでは、**視覚的品質**ポリシー設定の**Build to Lossless**の優先設定は、動画の場合、JPEGではなくH.264になりました。

H.264エンコードは、優れた画質を提供します。**ビデオコーデックを圧縮に使用**ポリシーは、その設定を制御し、デフォルトは**優先する場合に使用**です。**ロスレスにビルド**でJPEGを使用するように強制するには、**ビデオコーデックを圧縮に使用**ポリシーを**ビデオコーデックを使用しない**に設定します。クライアントが選択的H.264をサポートしていない場合、**ロスレスにビルド**はポリシー設定に関係なくJPEGにフォールバックします。Citrix Receiver for Windows 4.9〜4.12、Citrix Receiver for Linux 13.5〜13.10、Citrix Workspaceアプリ1808 for Windows以降、およびCitrix Workspaceアプリ1808 for Linux以降は、選択的H.264をサポートしています。**視覚的品質**および**ビデオコーデックを圧縮に使用**ポリシー設定の詳細については、視覚表示ポリシー設定とグラフィックポリシー設定を参照してください。

H.265ビデオコーデックのサポート

7.18リリース以降、Linux VDAはリモートグラフィックとビデオのハードウェアアクセラレーションのためにH.265ビデオコーデックをサポートしています。この機能は、Citrix Receiver for Windows 4.10〜4.12、およびCitrix Workspaceアプリ1808 for Windows以降で使用できます。この機能を利用するには、Linux VDAとクライアントの両方で有効にします。クライアントのGPUがDXVAインターフェイスを使用したH.265デコードをサポートしていない場合、グラフィックのH.265デコードポリシー設定は無視され、セッションはH.264ビデオコーデックの使用にフォールバックします。詳細については、H.265ビデオエンコードを参照してください。

VDAでH.265ハードウェアエンコードを有効にするには：

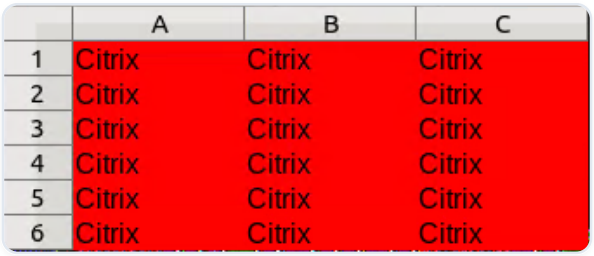
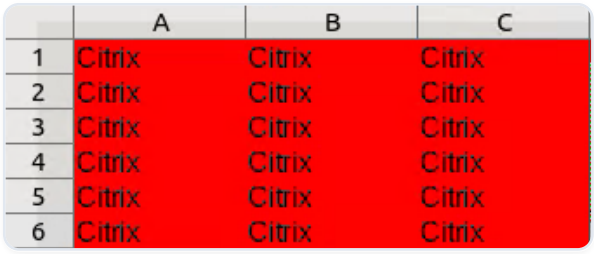
1. **ビデオコーデックにハードウェアエンコードを使用**ポリシーを有効にします。
2. **3Dグラフィックワークロード向けに最適化**ポリシーを有効にします。
3. **ビデオコーデックを圧縮に使用**ポリシーがデフォルトであるか、**画面全体**に設定されていることを確認します。
4. **視覚的品質**ポリシーが、**ロスレスにビルド**または**常にロスレス**に**設定されていない**ことを確認します。

クライアントでH.265ハードウェアエンコードを有効にするには、H.265ビデオエンコードを参照してください。

YUV444ソフトウェアエンコードのサポート

Linux VDAはYUV444ソフトウェアエンコードをサポートしています。YUVエンコードスキームは、各ピクセルに輝度と色の両方の値を割り当てます。YUVでは、「`Y`」は輝度、つまり「`luma`」値を表し、「`UV`」は色、つまり「クロマ」値を表します。このLinux VDAの機能は、Citrix Receiver for Windows 4.10〜4.12、およびCitrix Workspaceアプリ1808 for Windows以降で使用できます。

各Y、U、Vの固有値は8ビット、つまり1バイトのデータで構成されます。YUV444データ形式は、ピクセルあたり24ビットを送信します。YUV422データ形式は、2つのピクセル間でU値とV値を共有するため、ピクセルあたりの平均伝送速度は16ビットになります。次の表は、YUV444とYUV420の直感的な比較を示しています。

YUV444	YUV420
	

VDAでYUV444ソフトウェアエンコーディングを有効にするには：

1. **ビデオコーデックを使用して圧縮**ポリシーが**画面全体**に設定されていることを確認します。
2. **視覚品質**ポリシーが**常にロスレス**または**ロスレスに構築**に設定されていることを確認します。

帯域幅の見積もりに基づいて平均ビットレートを調整する

Citrixは、帯域幅の見積もりに基づいて平均ビットレートを調整することにより、HDX™ 3D Proハードウェアエンコーディングを強化します。

HDX 3D Proハードウェアエンコーディングが使用されている場合、VDAはネットワークの帯域幅を断続的に推定し、帯域幅の見積もりに基づいてエンコードされたフレームのビットレートを調整できます。この新機能は、シャープネスと流暢さのバランスを取るメカニズムを提供します。

この機能はデフォルトで有効になっています。無効にするには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "DisableReconfigureEncoder" -d "0x00000001" --force
```

この機能を使用するだけでなく、シャープネスと流暢さの間で調整するために、次のコマンドを実行することもできます。**AverageBitRatePercent**および**MaxBitRatePercent**パラメーターは、帯域幅使用率のパーセンテージを設定します。設定する値が高いほど、グラフィックはシャープになり、流暢さは低下します。推奨設定範囲は50～100です。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "AverageBitRatePercent" -d "90" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "MaxBitRatePercent" -d "100" --force
```

平均ビットレート調整では、画面が静止している場合、新しいフレームが送信されないため、最新のフレームは低品質の状態を維持します。シャープ化サポートは、最新のフレームを再構成して最高品質で即座に送信することにより、この問題に対処します。

Linux VDA Thinwireでサポートされているポリシーの完全なリストについては、ポリシーサポートリストを参照してください。

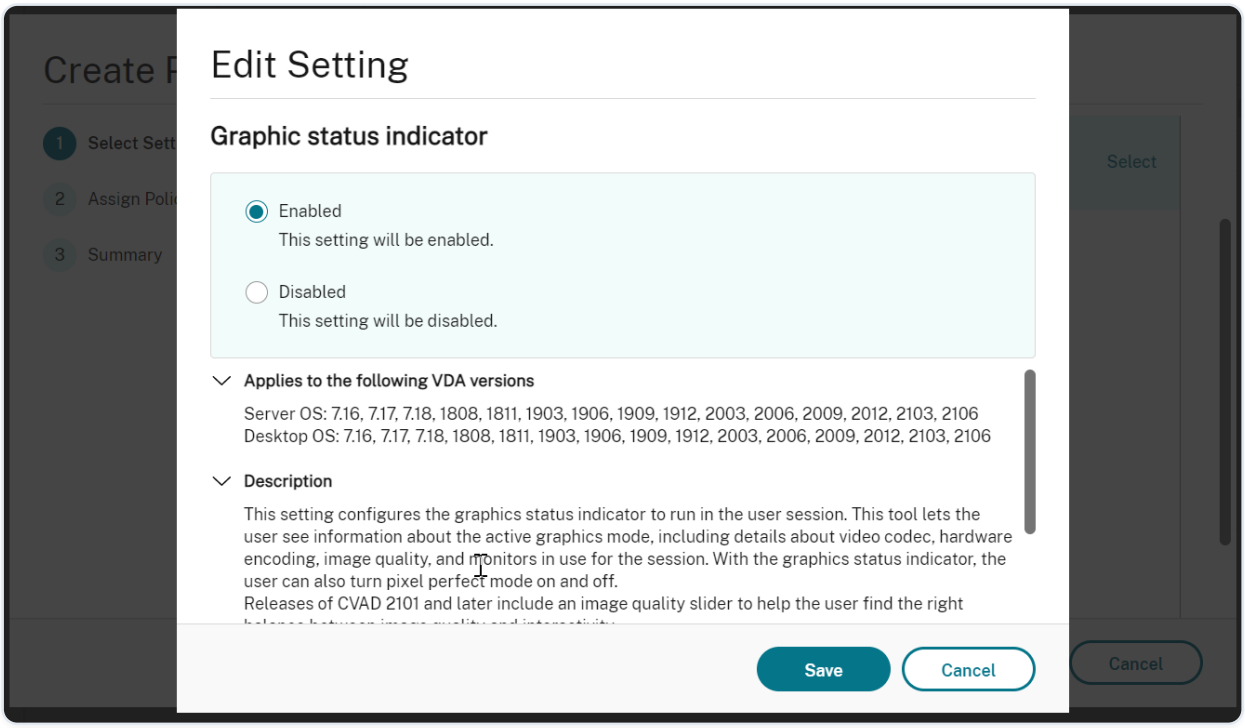
Linux VDAでのマルチモニターサポートの構成については、CTX220128を参照してください。

グラフィック品質スライダー

仮想Linuxセッションで実行されるグラフィックステータスインジケーターツールに、グラフィック品質スライダーを含めました。このスライダーは、画質とインタラクティブ性の適切なバランスを見つけるのに役立ちます。

スライダーを使用するには、次の手順を実行します。

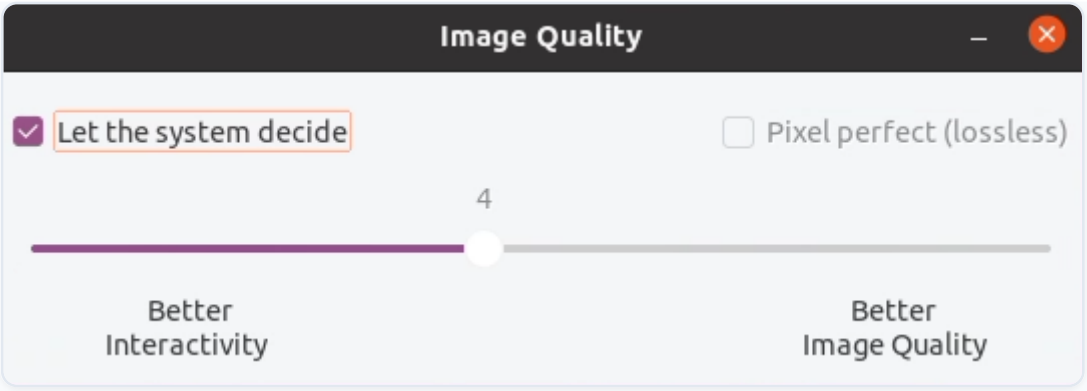
1. Citrix Studioで**グラフィックステータスインジケーター**ポリシーを有効にします。



2. ターミナルを開き、`ctxslider`コマンドを実行します。スライダーUIが表示されます。

注:

視覚品質ポリシーを**常にロスレス**または**ロスレスにビルド**に設定している場合、スライダーUIは表示されません。



次の選択肢が利用可能です。

- 画像品質を変更するには、スライダーを動かします。スライダーは0から9の範囲をサポートしています。
- システム定義の設定を使用するには、**システムに決定させる**を選択します。
- ロスレスモードに切り替えるには、**ピクセルパーフェクト**を選択します。

トラブルシューティング

使用中のグラフィックモードを確認する

次のコマンドを実行して、どのグラフィックモードが使用されているかを確認します（**0**はTW+、**1**は全画面ビデオコーデックを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000000" --force
```

H.264が使用されているかを確認する

次のコマンドを実行して、H.264が使用されているかを確認します（**0**は使用されていない、**1**は使用中を意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000000" --force
```

H.265が使用されているか確認する

全画面H.265が使用されているか確認するには、次のコマンドを実行します（**0**は使用されていないことを意味し、**1**は使用されていることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H265" -d "0x00000000" --force
```

どのYUVエンコードスキームが使用されているか確認する

どのYUVエンコードスキームが使用されているか確認するには、次のコマンドを実行します（**0**はYUV420を意味します。**1**はYUV422を意味します。**2**はYUV444を意味します）。

注： YUVFormatの値は、ビデオコーデックが使用されている場合にのみ意味があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000000" --force
```

YUV444ソフトウェアエンコードが使用されているか確認する

YUV444ソフトウェアエンコードが使用されているか確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
```

YUV444が使用されている場合、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000001" --force
```

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000001" --force create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000000" --force
```

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000002" --force
```

3D Proでハードウェアエンコードが使用されているか確認する

次のコマンドを実行します（**0**は使用されていないことを意味し、**1**は使用されていることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000001" --force
```

別の方法として、**nvidia-smi**コマンドを使用します。ハードウェアエンコードが使用されている場合、出力は次のようになります。

```
Tue Apr 12 10:42:03 2016

+-----------------------------------------------------+

| NVIDIA-SMI 361.28     Driver Version: 361.28         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
===============================+======================+==================
===|

|   0  GRID K1             Off  | 0000:00:05.0     Off |
N/A |

| N/A   42C    P0    14W /  31W |    207MiB /  4095MiB |      8%
Default |

+-------------------------------+----------------------
+----------------------+




+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|
```

```
|    0      2164  C+G   /usr/local/bin/ctxgfx
106MiB |

|    0      2187    G   Xorg
85MiB |

+-----------------------------------------------------------------------
----+
```

NVIDIA GRIDグラフィックドライバーが正しくインストールされていることを確認します

NVIDIA GRIDグラフィックドライバーが正しくインストールされていることを確認するには、**nvidia-smi**を実行します。結果は次のようになります。

```
+-----------------------------------------------------+

| NVIDIA-SMI 352.70     Driver Version: 352.70         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
===============================+======================+==================
===|

|   0  Tesla M60           Off  | 0000:00:05.0     Off |
Off |

| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%
Default |

+-------------------------------+----------------------
+----------------------+
```

```
+-----------------------------------------------------------------------------
----+

| Processes:                                                            GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|

|  No running processes found
                                 |

+-----------------------------------------------------------------------------
----+
```

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

HDX 3D Proマルチモニターの再描画に関する問題

プライマリモニター以外の画面で再描画の問題が発生している場合は、NVIDIA GRIDライセンスが利用可能であることを確認してください。

Xorgエラーログを確認する

Xorgのログファイルは、**/var/log/フォルダーにXorg.{DISPLAY}.log**のような名前で保存されています。

既知の問題と制限事項

vGPUの場合、Citrix Hypervisor™ローカルコンソールにICAデスクトップセッション画面が表示される

回避策: 次のコマンドを実行して、VMのローカルVGAコンソールを無効にします。

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
```

NVIDIA K2グラフィックカードは、パススルーモードでのYUV444ハードウェアエンコードをサポートしていません

ポリシー設定で**Build to Lossless**が有効になっている場合、NVIDIA K2グラフィックカードを使用してアプリ/デスクトップセッションを起動すると、黒または灰色の画面が表示されます。この問題は、NVIDIA K2グラフィックカードがパススルーモードでのYUV444ハードウェアエンコードをサポートしていないために発生します。詳しくは、「Video Encode and Decode GPU Support Matrix」を参照してください。

ログオン時にGnome 3デスクトップのポップアップが遅い

これはGnome 3デスクトップセッションの起動における制限です。

Citrix Workspaceアプリのウィンドウのサイズを変更すると、一部のOpenGL/WebGLアプリケーションが適切にレンダリングされない場合があります

Citrix Workspaceアプリのウィンドウのサイズを変更すると、画面の解像度が変わります。NVIDIA独自のドライバーは一部の内部状態を変更するため、アプリケーションがそれに応じて応答する必要がある場合があります。たとえば、WebGLライブラリ要素**lightgl.js**は、「`Rendering to this texture is not supported (incomplete frame buffer)`」というエラーを生成する可能性があります。

Thinwire プログレッシブ表示

低帯域幅または高遅延の接続では、セッションの対話性が低下する可能性があります。たとえば、帯域幅が 2 Mbps 未満または遅延が 200 ミリ秒を超える接続では、Web ページのスクロールが遅くなったり、応答しなくなったり、途切れたりすることがあります。キーボードとマウスの操作がグラフィックの更新に遅れることがあります。

バージョン 7.17 までは、ポリシー設定を使用して、セッションを **低** 視覚品質に構成するか、より低い色深度 (16 ビットまたは 8 ビットグラフィック) を設定することで、帯域幅の消費を削減できました。ただし、ユーザーが弱い接続を使用していることを知っている必要がありました。HDX Thinwire は、ネットワークの状態に基づいて静止画の品質を動的に調整しませんでした。

バージョン 7.18 以降、HDX Thinwire は、利用可能な帯域幅が 2 Mbps を下回るか、ネットワーク遅延が 200 ミリ秒を超える場合、デフォルトでプログレッシブ更新モードに切り替わります。このモードでは:

- すべての静止画が高度に圧縮されます。
- テキスト品質が低下します。

たとえば、プログレッシブ更新モードがアクティブな以下のグラフィックでは、文字 **F** と **e** に青いアーティファクトがあり、画像は高度に圧縮されています。このアプローチにより、帯域幅の消費が大幅に削減され、画像とテキストをより迅速に受信できるようになり、セッションの対話性が向上します。

Thinwire プログレッシブ表示 1 の画像 (/en-us/linux-virtual-delivery-agent/2106/media/thinwireprogressivedisplay.png)

セッションとの対話を停止すると、劣化した画像とテキストはプログレッシブにシャープ化され、ロスレスになります。たとえば、以下のグラフィックでは、文字に青いアーティファクトがなくなり、画像がソース品質で表示されます。

Thinwire プログレッシブ表示 2 の画像 (/en-us/linux-virtual-delivery-agent/2106/media/thinwireprogressivedisplay2.png)

画像の場合、シャープ化にはランダムなブロック状の方法が使用されます。テキストの場合、個々の文字または単語の一部がシャープ化されます。シャープ化プロセスは複数のフレームにわたって行われます。このアプローチにより、単一の大きなシャープ化フレームによる遅延の発生が回避されます。

一時的な画像 (ビデオ) は、アダプティブディスプレイまたは選択的 H.264 で引き続き管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは **視覚品質** ポリシー設定の **高**、**中** (デフォルト)、および **低** に対してスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフ (使用しない) になります:

- **視覚品質** = **常にロスレス** または **ロスレスに構築**
- **シンプルなグラフィックの優先色深度** = 8ビット
- **圧縮にビデオコーデックを使用** = **画面全体に対して** (全画面H.264が必要な場合)

プログレッシブモードがスタンバイ状態の場合、デフォルトでは以下のいずれかの条件が発生したときに有効になります。

- 利用可能な帯域幅が2 Mbpsを下回る
- ネットワーク遅延が200 msを超える

モード切り替えが発生した後、不利なネットワーク状況が一時的なものであっても、そのモードで最低10秒間費やされます。

プログレッシブモードの動作を変更する

次のコマンドを実行して、プログレッシブモードの動作を変更できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplay" -d "<value>" --force
```

ここで、\<value\> は次のとおりです。

0 = 常にオフ (いかなる状況でも使用しない)

1 = 自動 (ネットワーク状況に基づいて切り替え、デフォルト値)

2 = 常にオン

自動モード (1) の場合、プログレッシブモードが切り替わるしきい値を変更するには、以下のいずれかのコマンドを実行できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

ここで、\<value\> は \<Kbps単位のしきい値\> です (デフォルト = 2,048)

例: 4096 = 帯域幅が4 Mbpsを下回った場合にプログレッシブモードをオンにする

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

ここで、\<value\> は \<threshold in ms\> です（デフォルト = 200）

例: 100 = ネットワーク遅延が 100 ミリ秒を下回った場合、プログレッシブモードをオンにする。

非GRID 3Dグラフィック

概要

この機能強化により、Linux VDAはNVIDIA GRID 3Dカードだけでなく、非GRID 3Dカードもサポートします。

インストール

非GRID 3Dグラフィック機能を使用するには、次の操作を行う必要があります。

- 前提条件としてXDamageをインストールします。通常、XDamageはXServerの拡張機能として存在します。
- Linux VDAのインストール時に`CTX_XDL_HDX_3D_PRO`を`Y`に設定します。環境変数については、「手順4：インストールを完了するためのランタイム環境のセットアップ」を参照してください。

構成

Xorg構成ファイル

3Dカードドライバーが NVIDIAの場合、構成ファイルは自動的にインストールおよび設定されます。

その他の種類の3Dカード

3Dカードドライバーが NVIDIAではない場合、`/etc/X11/`にインストールされている4つのテンプレート構成ファイルを変更する必要があります。

- ctx-driver_name-1.conf
- ctx-driver_name-2.conf
- ctx-driver_name-3.conf
- ctx-driver_name-4.conf

ctx-driver_name-1.confを例として、テンプレート構成ファイルを変更するには、次の手順を実行します。

1. ***driver_name*** を実際のドライバー名に置き換えます。

   たとえば、ドライバー名が `intel` の場合、構成ファイル名を `ctx-intel-1.conf` に変更できます。
2. ビデオドライバー情報を追加します。

   各テンプレート構成ファイルには、「Device」という名前のセクションが含まれており、コメントアウトされています。このセクションでは、ビデオドライバー情報について説明します。ビデオドライバー情報を追加する前に、このセクションを有効にしてください。このセクションを有効にするには：

a. 構成情報については、製造元が提供する3Dカードガイドを参照してください。ネイティブ構成ファイルを生成できます。Linux VDA ICA® セッションを使用していないときに、ネイティブ構成ファイルを使用して3Dカードがローカル環境で動作することを確認してください。

b. ネイティブ構成ファイルの「Device」セクションを **ctx-driver_name-1.conf** にコピーします。

3. 次のコマンドを実行してレジストリキーを設定し、Linux VDA が手順1で設定した構成ファイル名を認識できるようにします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_SZ" -v "DriverName" -d "intel" --force
```

非GRID 3Dグラフィック機能を有効にする

非GRID 3Dグラフィック機能はデフォルトで無効になっています。XDamageEnabled を 1 に設定することで、次のコマンドを実行して有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

トラブルシューティング

グラフィック出力がない、または文字化けしている

3Dアプリケーションをローカルで実行でき、すべての構成が正しい場合、グラフィック出力がない、または文字化けしているのはバグが原因です。デバッグのためにトレース情報を収集するには、/opt/Citrix/VDA/bin/setlog を使用し、GFX_X11 を verbose に設定します。

ハードウェアエンコーディングが機能しない

この機能はソフトウェアエンコーディングのみをサポートしています。

ポリシーを構成する

インストール

インストール記事に従ってLinux VDAを準備します。

依存関係

Linux VDAパッケージをインストールする前に、これらの依存関係をインストールしていることを確認してください。

RHEL/CentOS:

```
sudo yum -y install openldap


sudo yum -y install libxml2


sudo yum -y install cyrus-sasl


sudo yum -y install cyrus-sasl-gssapi
```

SLES/SELD:

```
sudo zypper install openldap2


sudo zypper install libxml2


sudo zypper install cyrus-sasl


sudo zypper install cyrus-sasl-gssapi
```

ウブントゥ:

```
sudo apt-get install -y libldap-2.4-2
```

```
sudo apt-get install -y libsasl2-2
```

```
sudo apt-get install -y libsasl2-modules-gssapi-mit
```

構成

Citrix Studioでのポリシー設定

Citrix Studioでポリシーを設定するには、次の手順を実行します。

1. **Citrix Studio**を開きます。
2. **ポリシー**パネルを選択します。
3. **ポリシーの作成** をクリックします。
4. ポリシーサポートリストに従ってポリシーを設定します。

VDAでのLDAPサーバー設定

Linux VDAでのLDAPサーバー設定は、単一ドメイン環境ではオプションですが、複数ドメインおよび複数フォレスト環境では必須です。この設定は、ポリシーサービスがこれらの環境でLDAP検索を実行するために必要です。

Linux VDAパッケージのインストール後、次のコマンドを実行します。

```
/opt/Citrix/VDA/sbin/ctxsetup.sh
```

提案された形式ですべてのLDAPサーバーを入力してください: LDAPポートを含む、スペース区切りのLDAP完全修飾ドメイン名 (FQDN) のリスト (例: ad1.mycompany.com:389 ad2.mycomany.com:389)。

Linux VDA上のLDAPサーバー設定の画像(/en-us/linux-virtual-delivery-agent/2106/media/ldap-server-setting-on-linux-vda-new.png)

ctxreg コマンドを実行して、この設定をレジストリに直接書き込むこともできます:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
```

```
"ListOfLDAPServers" -d "ad1.mycompany.com:389 ad2.mycomany.com:389" --
force
```

ポリシーサポートリスト

Linux VDA ポリシーサポートリスト

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントのローカル時刻を使用する	UseLocalTimeOfClient	ユーザー	ICA\タイムゾーン制御	サーバーのタイムゾーンを使用する
ICAラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICAラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15
アイドル接続のICAラウンドトリップ計算	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザー監視	無効 (0)
セッション全体の帯域幅制限	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限（パーセント）	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクトの帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントUSB デバイスリダイレクト帯域幅パーセント	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限（パーセント）	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅制限	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクトの帯域幅制限（パーセント）	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの帯域幅制限（パーセント）	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0
WebSockets接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocketsポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
WebSockets信頼済みオリジンサーバーリスト	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICA® キープアライブ	SendICAKeepAlives	コンピューター	ICAキープアライブ	ICAキープアライブメッセージを送信しない (0)
ICAキープアライブタイムアウト	ICAKeepAliveTimeout	コンピューター	ICAキープアライブ	60秒
ICAリスナーポート番号	IcaListenerPortNumber	コンピューター	ICA	1494
HDX™ アダプティブトランスポート	HDXoverUDP	コンピューター	ICA	優先(2)
セッションの信頼性接続	AcceptSessionReliabilityConnections	コンピューター	ICA\セッションの信頼性	許可(1)
再接続UIの透過レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\自動クライアント再接続	80%
セッションの信頼性ポート番号	SessionReliabilityPort	コンピューター	ICA\セッションの信頼性	2598
セッションの信頼性タイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッションの信頼性	180 秒
クライアントの自動再接続	AllowAutoClientReconnect	ユーザー	ICA\自動クライアント再接続	許可 (1)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDFユニバーサルプリンターの自動作成	PDFプリンターの自動作成	ユーザー	印刷	無効 (0)
プリンタードライバーのマッピングと互換性	ドライバーマッピングリスト	ユーザー	印刷	`"Microsoft XPS Document Writer *, Deny;Send to Microsoft OneNote *, Deny"`
クライアントクリップボードリダイレクト	クリップボードリダイレクトを許可	ユーザー	クリップボード	許可 (1)
クライアントUSBデバイスのリダイレクト	USBリダイレクトを許可	ユーザー	USB	禁止 (0)
クライアントUSBデバイスリダイレクト規則	USBデバイスルール	ユーザー	USB	“\0”
動画圧縮	動画圧縮構成	ユーザー	シンワイヤー	有効 (1)
追加の色圧縮	追加の色圧縮	ユーザー	シンワイヤー	無効 (0)
目標最小フレームレート	目標最小フレーム/秒	ユーザー	シンワイヤー	10 フレーム/秒

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ターゲットフレームレート	フレーム/秒	ユーザー	シンワイヤー	30 フレーム/秒
視覚的な品質	視覚品質	ユーザー	シンワイヤー	中 (3)
圧縮にビデオコーデックを使用	ビデオコーデック	ユーザー	シンワイヤー	優先する場合に使用 (3)
ビデオコーデックにハードウェアエンコードを使用	ビデオコーデックにハードウェアエンコードを使用	ユーザー	シンワイヤー	有効 (1)
視覚的にロスレスな圧縮を許可する	視覚的にロスレス圧縮を許可	ユーザー	シンワイヤー	無効 (0)
3Dグラフィックワークロード向けに最適化	3Dワークロード向けに最適化	ユーザー	シンワイヤー	無効 (0)
単純なグラフィックの優先色深度	優先カラー深度	ユーザー	シンワイヤー	24ビット/ピクセル (1)
オーディオ品質	サウンド品質	ユーザー	オーディオ	高 – 高品位オーディオ (2)
クライアントマイクのリダイレクト	マイクのリダイレクトを許可	ユーザー	オーディオ	許可 (1)
最大セッション数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオン許容数	ConcurrentLogonsTolerance	コンピューター	負荷管理	2

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
コントローラーの自動更新を有効にする	EnableAutoUpdateOfControllers	コンピューター	仮想デリバリーエージェント設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3
最大Speex品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブの自動接続	AutoConnectDrives	ユーザー	ファイルリダイレクト/CDM	有効 (1)
クライアント光学ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントフロッピードライブ	AllowFloppyDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントネットワークドライブ	AllowNetworkDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントドライブのリダイレクト	AllowDriveRedir	ユーザー	ファイルリダイレクト/CDM	許可 (1)
読み取り専用クライアントドライブアクセス	ReadOnlyMappedDrive	ユーザー	ファイルリダイレクト/CDM	無効 (0)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
自動キーボード表示	AllowAutoKeyboardPopUp	ユーザー	MRVC	無効 (0)
デスクトップとクライアント間のファイル転送を許可する	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロード	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロード	AllowFileUpload	ユーザー	ファイル転送	許可
セッションアイドルタイマー	EnableSessionIdleTimer	ユーザー	セッションタイマー	有効 (1)
セッションアイドルタイマー間隔	SessionIdleTimerInterval	ユーザー	セッションタイマー	1440分
切断されたセッションタイマー	EnableSessionDisconnectTimer	ユーザー	セッションタイマー	無効 (0)
切断セッションタイマー間隔	SessionDisconnectTimerPeriod	ユーザー	セッションタイマー	1440分

注:

Windows Virtual Delivery Agent (VDA) のみが、ユーザーデータグラムプロトコル (UDP) 経由のオーディオをサポートします。Linux VDA はサポートしません。詳しくは、「ユーザーデータグラムプロトコル (UDP) 経由のオーディオリアルタイムトランスポート」を参照してください。

> Citrix Studioで、以下のCitrixポリシー設定を使用してセッション接続タイマーを構成できます:

- 「**セッションアイドルタイマー**」: アイドル状態のセッションに時間制限を適用するかどうかを決定します。
- 「**セッションアイドルタイマー間隔**」: アイドル状態のセッションに時間制限を設定します。セッションアイドルタイマーが**有効**で、設定された時間内にアクティブなセッションでユーザー入力がなかった場合、セッションは切断されます。
- 「**切断されたセッションタイマー**」: 切断されたセッションに時間制限を適用するかどうかを決定します。
- 「**切断されたセッションタイマー間隔**」: 切断されたセッションがログオフされるまでの間隔を設定します。

いずれかのポリシー設定を更新する場合は、それらが展開全体で一貫していることを確認してください。

アイドルセッションの制限時間が経過すると、警告メッセージが表示されます。例については、次のスクリーンショットを参照してください。**OK**を押すと警告メッセージは閉じますが、セッションをアクティブに保つことはできません。セッションをアクティブに保つには、アイドルタイマーをリセットするためにユーザー入力を提供してください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2106/media/warning-message-idle-session-timeout.png)

以下のポリシーは、Citrix Studioバージョン7.12以降で構成できます。

- MaxSpeexQuality

  値（整数）：

  デフォルト値：5

  詳細：

  オーディオ品質が中または低の場合（ポリシー「オーディオ品質」を参照）、オーディオリダイレクトはSpeexコーデックを使用してオーディオデータをエンコードします。Speexは非可逆コーデックであり、入力音声信号の忠実度を犠牲にして圧縮を実現します。他の音声コーデックとは異なり、品質とビットレートの間のトレードオフを制御することが可能です。Speexエンコードプロセスは、ほとんどの場合、0から10の範囲の品質パラメーターによって制御されます。品質が高いほど、ビットレートも高くなります。

最大Speex品質は、オーディオ品質と帯域幅制限（ポリシー「オーディオリダイレクト帯域幅制限」を参照）に従って、オーディオデータをエンコードするための最適なSpeex品質を選択します。オーディオ品質が中の場合、エンコーダーは広帯域モードになり、サンプリングレートが高くなります。オーディオ品質が低い場合、エンコーダーは狭帯域モードになり、サンプリングレートが低くなります。同じSpeex品質でも、モードによって異なるビットレートになります。最適なSpeex品質とは、最大値が次の条件を満たす場合です。

- 最大Speex品質以下であること。
- そのビットレートが帯域幅制限以下であること。

関連設定：オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

  値（列挙型）：

  デフォルト値：3

  詳細：

  プライマリ選択は、データを選択し、マウスの中央ボタンを押して貼り付けるときに使用されます。

  このポリシーは、Linux VDA とクライアントでのプライマリ選択の変更が、互いのクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。

  プライマリ選択更新モードの画像(/en-us/linux-virtual-delivery-agent/7-15-ltsr/media/primary-selection-update-mode.png)

  - **クライアントにもホストにも選択の変更は更新されない**

    Linux VDA でのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDA のクリップボードを更新しません。

  - **ホストの選択の変更はクライアントに更新されない**

    Linux VDA でのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDA のクリップボードを更新します。

  - **クライアントの選択の変更はホストに更新されない**

Linux VDA でのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDA のクリップボードを更新しません。

- **クライアントとホストの両方で選択の変更が更新される**

  Linux VDA でのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDA のクリップボードを更新します。このオプションがデフォルト値です。

関連設定: クリップボード選択更新モード

- ClipboardSelectionUpdateMode

値 (enum):

デフォルト値: 3

詳細:

クリップボード選択は、データを選択し、ショートカットメニューから「コピー」を選択するなどして、明示的にクリップボードに「コピー」を要求するときに使用されます。クリップボード選択は主に Microsoft Windows のクリップボード操作に関連して使用されますが、プライマリ選択は Linux 固有のものです。

このポリシーは、Linux VDA とクライアントでのクリップボード選択の変更が、互いのクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。

クリップボード選択更新モードの画像(/en-us/linux-virtual-delivery-agent/7-15-ltsr/media/clipboard-selection-update-mode.png)

- **選択の変更はクライアントにもホストにも更新されません**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されません**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されません**

Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **選択の変更はクライアントとホストの両方で更新されます**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

注:

Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピー＆ペーストの動作を制御するには、クリップボード選択更新モードとプライマリ選択更新モードの両方を同じ値に設定することをお勧めします。

IPv6 の構成

Linux VDA は Citrix Virtual Apps and Desktops と連携するために IPv6 をサポートしています。この機能を使用する際は、以下を考慮してください。

- デュアルスタック環境では、IPv6 が明示的に有効になっていない限り、IPv4 が使用されます。
- IPv4 環境で IPv6 が有効になっている場合、Linux VDA は機能しません。

重要:

- Linux VDA だけでなく、ネットワーク環境全体が IPv6 である必要があります。
- Centrify は純粋な IPv6 をサポートしていません。

Linux VDA をインストールする際に、IPv6 の特別なセットアップタスクは必要ありません。

Linux VDA の IPv6 を構成する

Linux VDA の構成を変更する前に、Linux 仮想マシンが以前に IPv6 ネットワークで動作していたことを確認してください。IPv6 構成に関連するレジストリキーは2つあります。

```
“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“OnlyUseIPv6ControllerRegistration”

“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“ControllerRegistrationIPv6Netmask”
```

Linux VDA で IPv6 を有効にするには、**OnlyUseIPv6ControllerRegistration** を 1 に設定する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

Linux VDA に複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask** を使用して、Linux VDA の登録に使用するインターフェイスを指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ControllerRegistrationIPv6Netmask " -d "{IPv6 netmask}" --force
```

{IPv6 netmask} を実際のネットマスク (例: 2000::/64) に置き換えてください。

Citrix Virtual Apps and Desktops での IPv6 展開の詳細については、IPv4/IPv6 のサポート を参照してください。

トラブルシューティング

基本的な IPv6 ネットワーク環境を確認し、ping6 を使用して AD および Delivery Controller™ に到達可能かどうかを確認してください。

Citrixカスタマーエクスペリエンス向上プログラム (CEIP) の構成

CEIPに参加すると、匿名の統計情報と使用状況情報が、Citrix製品の品質とパフォーマンスを向上させるためにCitrixに送信されます。さらに、匿名データのコピーは、迅速かつ効率的な分析のためにGoogle Analytics (GA) に送信されます。

レジストリ設定

デフォルトでは、Linux VDAをインストールすると、自動的にCEIPに参加します。データの初回アップロードは、Linux VDAのインストール後約7日で行われます。このデフォルト設定はレジストリで変更できます。

- **CEIPスイッチ**

CEIPを有効または無効にするレジストリ設定 (デフォルト = 0):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: CEIPSwitch

値: 1 = 無効、0 = 有効

指定しない場合、CEIPは有効になります。

クライアントで次のコマンドを実行してCEIPを無効にできます:

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

- **GAスイッチ**

GAを有効または無効にするレジストリ設定 (デフォルト = 0):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: GASwitch

値: 1 = 無効, 0 = 有効

指定しない場合、GAは有効になります。

クライアントでGAを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "GASwitch" -d "1"
```

- **DataPersistPath**

データ永続化パスを制御するレジストリ設定 (デフォルト = /var/xdl/ceip):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: DataPersistPath

値: 文字列

このパスを設定するには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "DataPersistPath" -d
"your_path"
```

構成されたパスが存在しないか、アクセスできない場合、データはデフォルトパスに保存されます。

Linux VDAから収集されたCEIPデータ

次の表は、収集される匿名情報の種類の例を示しています。このデータには、お客様を特定するような詳細は含まれていません。

データポイント	キー名	説明
マシンGUID	machine_guid	データの発信元となるマシンを識別します
ADソリューション	ad_solution	マシンのドメイン参加方法を示すテキスト文字列
Linuxカーネルバージョン	kernel_version	マシンのカーネルバージョンを示すテキスト文字列
LVDAバージョン	vda_version	インストールされているLinux VDAのバージョンを示すテキスト文字列。

データポイント	キー名	説明
LVDAの更新または新規インストール	update_or_fresh_install	現在のLinux VDAパッケージが新規インストールされているか、更新されているかを示すテキスト文字列
LVDAのインストール方法	install_method	現在のLinux VDAパッケージがMCS、PVS、簡易インストール、または手動インストールを使用してインストールされていることを示すテキスト文字列。
HDX 3D Proが有効かどうか	hdx_3d_pro	マシンでHDX 3D Proが有効になっているかどうかを示すテキスト文字列
VDIモードが有効かどうか	vdi_mode	VDIモードが有効になっているかどうかを示すテキスト文字列
システムロケール	system_locale	このマシンのロケールを示すテキスト文字列
LVDAキーサービスの最終再起動時刻	シーティーエックスエイチディーエックス シーティーエックスブイディーエー	`ctxhdx`および`ctxvda`サービスの最終再起動時刻。形式はdd-hh:mm:ssで、例: 10-17:22:19
GPUの種類	gpu_type	マシンのGPUタイプを示す
CPUコア数	cpu_cores	マシンのCPUコア数を示す整数
CPU周波数	cpu_frequency	MHz単位のCPU周波数を示す浮動小数点数
物理メモリサイズ	memory_size	KB単位の物理メモリサイズを示す整数
起動されたセッション数	session_launch	このデータポイントを収集する時点でマシン上で起動された（ログオンまたは再接続さ

データポイント	キー名	説明
		れた）セッション数を示す整数
Linux OS名とバージョン	os_name_version	マシンのLinux OS名とバージョンを示すテキスト文字列
セッションキー	session_key	データの発信元であるセッションを識別します
リソースの種類	resource_type	起動されたセッションのリソースの種類を示すテキスト文字列: デスクトップまたは `<appname>`
アクティブセッション時間	active_session_time	セッションのアクティブ時間を保存するために使用されます。セッションは切断/再接続できるため、1つのセッションに複数のアクティブ時間がある場合があります
セッション継続時間	session_duration_time	ログオンからログオフまでのセッション継続時間を保存するために使用されます
Receiverクライアントの種類	receiver_type	セッションの起動に使用されたCitrix Workspaceアプリのタイプを示す整数
Receiverクライアントのバージョン	receiver_version	セッションの起動に使用されたCitrix Workspaceアプリのバージョンを示すテキスト文字列
印刷回数	printing_count	セッションが印刷機能を使用した回数を示す整数
USBリダイレクト回数	usb_redirecting_count	セッションがUSBデバイスを使用した回数を示す整数

データポイント	キー名	説明
グラフィックプロバイダーの種類	gfx_provider_type	セッションのグラフィックプロバイダーの種類を示すテキスト文字列
シャドウイング回数	shadow_count	セッションがシャドウされた回数を表す整数
ユーザーが選択した言語	ctxism_select	ユーザーが選択したすべての言語を含む構成された長い文字列
スマートカードのリダイレクト回数	scard_redirecting_count	セッションログオンおよびセッション内アプリのユーザー認証にスマートカードのリダイレクトが使用された回数を表す整数

USBリダイレクトの構成

USBデバイスは、Citrix Workspace™ appとLinux VDAデスクトップ間で共有されます。USBデバイスがデスクトップにリダイレクトされると、ユーザーはそのUSBデバイスをローカルに接続されているかのように使用できます。

> **ヒント:**

> ネットワーク遅延が100ミリ秒未満の場合は、USBリダイレクトを使用することをお勧めします。ネットワーク遅延が200ミリ秒を超える場合は、USBリダイレクトを使用しないでください。

> USBリダイレクトには、主に次の3つの機能領域があります。

- オープンソースプロジェクトの実装（VHCI）
- VHCIサービス
- USBサービス

オープンソースVHCI：

USBリダイレクト機能のこの部分は、IPネットワークを介した一般的なUSBデバイス共有システムを開発します。これは、Linuxカーネルドライバーと、カーネルドライバーと通信してすべてのUSBデータを取得できるいくつかのユーザーモードライブラリで構成されています。Linux VDAの実装では、CitrixはVHCIのカーネルドライバーを再利用します。ただし、Linux VDAとCitrix Workspace app間のすべてのUSBデータ転送は、Citrix ICA®プロトコルパッケージにカプセル化されます。

VHCIサービス：

VHCIサービスは、VHCIカーネルモジュールと通信するためにCitrixが提供するオープンソースサービスです。このサービスは、VHCIとCitrix USBサービス間のゲートウェイとして機能します。

USBサービス：

USBサービスは、USBデバイス上のすべての仮想化とデータ転送を管理するCitrixモジュールを表します。

USBリダイレクトの仕組み

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム/devパスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで利用できない場合があります。USBデバイスは正常に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスできません。USBデバイスの接続を確保するには、ドライバーをインストールし、システムを適切に構成してください。

Linux VDA は、クライアントとの間で正常にリダイレクトされる USB デバイスのリストをサポートしています。さらに、デバイス、特に USB ディスクは適切にマウントされ、ユーザーは追加設定なしでディスクにアクセスできます。

サポートされている USB デバイス

以下のデバイスは、このバージョンの Linux VDA をサポートすることが確認されています。その他のデバイスも自由に使用できますが、予期せぬ結果が生じる可能性があります。

> **注:**
>
> Linux VDA は USB 2.0 プロトコルのみをサポートしています。

USB 大容量記憶装置	VID:PID	ファイルシステム
ネットアックテクノロジー株式会社	0dd8:173c	FAT32
キングストン データトラベラー 101 II	0951:1625	FAT32
キングストン データトラベラー GT101 G2	1567:8902	FAT32
SanDisk SDCZ80 フラッシュドライブ	0781:5580	FAT32
WD HDD	1058:10B8	FAT32

USB 3Dマウス	VID:PID
3Dコネクション スペースマウス プロ	046d: c62b

USB スキャナー	VID:PID
エプソン パーフェクション V330 フォト	04B8: 0142

USBリダイレクトの構成

Citrixポリシーは、USBデバイスリダイレクトが有効か無効かを制御します。さらに、デバイスの種類はDelivery Controller™ポリシーを使用して指定することもできます。Linux VDAのUSBリダイレクトを構成する場合、次のポリシーとルールを構成します。

- クライアントUSBデバイスリダイレクトポリシー
- クライアントUSBデバイスリダイレクト規則

USBリダイレクトポリシーを有効にする

Citrix Studioで、クライアントとの間でUSBデバイスリダイレクトを有効（または無効）にします（ワークステーションホストのみ）。

設定の編集ダイアログで：

1. **許可**を選択します。

2. **OK**をクリックします。

USBリダイレクトを有効にするイメージ(/en-us/linux-virtual-delivery-agent/2106/media/client-usb-redirection-policy.png)

USBリダイレクトルールを設定する

USBリダイレクトポリシーを有効にした後、Citrix Studioを使用して、Linux VDAで許可（または拒否）するデバイスを指定してリダイレクトルールを設定します。

クライアントUSBデバイスリダイレクトルールダイアログで：

1. リダイレクトルールを追加するには**新規**をクリックし、既存のルールを確認するには**編集**をクリックします。
2. ルールを作成（または編集）したら、**OK**をクリックします。

クライアントUSBリダイレクトルール設定のイメージ(/en-us/linux-virtual-delivery-agent/2106/media/client-usb-redirection-rule.png)

VHCIカーネルモジュールをビルドする

USBリダイレクトはVHCIカーネルモジュール（`usb-vhci-hcd.ko`および`usb-vhci-iocif.ko`）に依存します。これらのモジュールはLinux VDAディストリビューションの一部（RPMパッケージの一部）です。これらは公式のLinuxディストリビューションカーネルに基づいてコンパイルされており、次の表に示されています。

サポートされているLINUXディストリビューション	カーネルバージョン
RHEL 8.3	4.18.0-240
RHEL 8.2	4.18.0-240
RHEL 8.1	4.18.0-240
RHEL 7.9	3.10.0-1160
RHEL 7.8	3.10.0-1160
SUSE 12.5	4.12.14
ウブントゥ 20.04	5.4.0-70
ウブントゥ 18.04	4.15.0-140

サポートされているLINUXディストリビューション	カーネルバージョン
ウブントゥ 16.04	4.4.0-206
デビアン 10	4.19.0-16

重要:

マシンのカーネルがLinux VDA用にビルドされたドライバーと互換性がない場合、USBサービスが起動に失敗する可能性があります。この場合、独自のVHCIカーネルモジュールをビルドした場合にのみ、USBリダイレクト機能を使用できます。

カーネルがCitrixによってビルドされたモジュールと整合しているかを確認する

コマンドラインで、カーネルが整合しているかどうかを確認するために、次のコマンドを実行します。

```
insmod /opt/Citrix/VDA/lib64/usb-vhci-hcd.ko
```

コマンドが正常に実行された場合、カーネルモジュールは正常にロードされており、バージョンはCitrixによってインストールされたものと整合しています。

コマンドがエラーで実行された場合、カーネルはCitrixモジュールと整合しておらず、再ビルドする必要があります。

VHCIカーネルモジュールを再ビルドする

カーネルモジュールがCitrixバージョンと整合していない場合は、次の手順を実行します。

1. Citrixダウンロードサイトからの LVDAソースコードをダウンロードします。「**Linux Virtual Delivery Agent (sources)**」セクションに含まれるファイルを選択します。
2. 「**citrix-linux-vda-sources.zip**」ファイルを解凍します。「**linux-vda-souces/vhci-hcd-1.15.tar.bz2**」に移動し、「**tar xvf vhci-hcd-1.15.tar.bz2**」を使用してVHCIソースファイルを解凍します。
3. ヘッダーファイルと「**Module.symvers**」ファイルに基づいてカーネルモジュールをビルドします。適切なLinuxディストリビューションに基づいてカーネルヘッダーファイルをインストールし、「**Module.symvers**」を作成するには、次の手順を使用します。

   RHEL/CentOS:

```
yum install kernel-devel
```

SUSE 12:

```
zypper install kernel-devel



zypper install kernel-source
```

ウブントゥ:

```
apt-get install linux-headers
```

> **ヒント:**
>
> インストールが成功すると、次のようなカーネルフォルダーが作成されます。
>
> /usr/src/kernels/3.10.0-327.10.1.el7.x86_64

4. /usr/src/kernels/3.10.0-327.10.1.el7.x86_64フォルダーで、「**Module.symvers**」ファイルが存在することを確認します。このファイルがフォルダーにない場合は、カーネルをビルドして（次のコマンドを順に実行します: `make oldconfig`; `make prepare`; `make modules`; `make`）このファイルを取得するか、「**/usr/src/kernels/3.10.0-327.10.1.el7.x86_64-obj/x86_64/defaults/module.***」からコピーします。
5. 開発ツールをインストールするには、次のコマンドを実行します。

   RHEL 8、CentOS 8:

   ```
   yum groupinstall 'Development Tools'



   yum install elfutils-libelf-devel
   ```

   RHEL 7、CentOS 7:

```
yum groupinstall 'Development Tools'
```

ウブントゥ 20.04、ウブントゥ 18.04、Debian 10:

```
apt install build-essential flex bison libelf-dev
```

ウブントゥ 16.04:

```
apt install build-essential flex bison
```

6. `vhci-hcd-1.15/Makefile`ファイルで、VCHIのMakefileを変更し、KDIRをカーネルディレクトリに設定します。

```
#KDIR = $(BUILD_PREFIX)/lib/modules/$(KVERSION)/build



KDIR = /usr/src/kernels/3.10.0-327.10.1.el7.x86_64
```

7. `vhci-hcd-1.15/`フォルダーで、`make`を実行してVHCIカーネルをビルドします。

> **注:**
>
> ビルドが成功した場合、`vhci-hcd-1.15/`フォルダーに`usb-vhci-hcd.ko`と`usb-vhci-iocifc.ko`が作成されます。

8. カーネルモジュールを新しくビルドしたものに置き換えます: **cp -f usb-vhci-*.ko /opt/Citrix/VDA/lib64/**
9. USBサービスを再起動します:

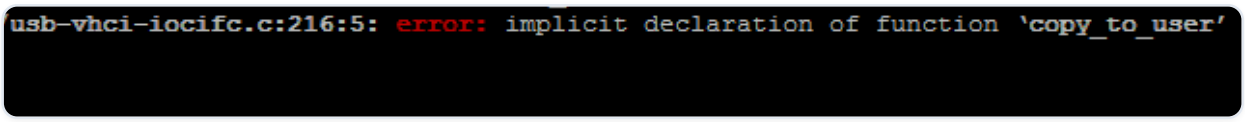
```
service ctxusbsd restart
```

10. セッションからログオフし、再度ログオンします。USBリダイレクトが機能しているか確認します。

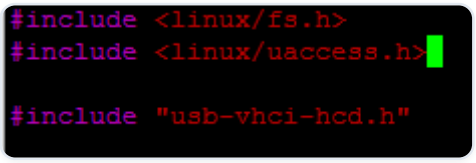
カーネルビルドの問題のトラブルシューティング

特定のカーネルでVHCIモジュールをビルドすると、次のエラーが発生する可能性があります:

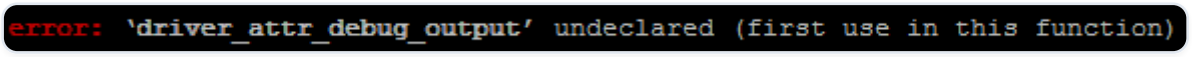
- `implicit declaration of function 'copy\_to\_user'` エラーが発生する可能性があります。次のスクリーンショットを参照してください:

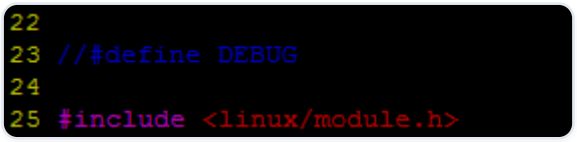
  このエラーは、カーネルのヘッダーファイルの変更が原因で発生します。回避策として、**#include \<linux/uaccess.h\>** の行を `vhci-hcd-1.15/usb-vhci-iocifc.c` ファイルに追加します。

- `'driver\_attr\_debug_output' undeclared` エラーが発生する可能性があります。次のスクリーンショットを参照してください:

  このエラーは、カーネルにシンボルがない場合に発生します。回避策として、`vhci-hcd-1.15/usb-vhci-iocifc.c` ファイルと `vhci-hcd-1.15/usb-vhci-hcd.c` ファイルでDEBUGのマクロ定義を無効にします。

- `'make[3]: *** No rule to make target 'arch/x86/tools/relocs_32.c', needed by 'arch/x86/tools/relocs_32.o'. Stop.` エラーが発生する可能性があります。次のスクリーンショットを参照してください:

  カーネルビルドエラーの画像(/en-us/linux-virtual-delivery-agent/2106/media/kernel-building-error4.png)

  回避策として、`vhci-hcd-1.15/` パスで以下のコマンドを使用して、`SUBDIRS=$(PWD)` を `M=$(shell pwd)` に置き換えます。

  ```
  sed -i 's/SUBDIRS=$(PWD)/M=$(shell pwd)/g' Makefile
  ```

```
sed -i 's/SUBDIRS=$(PWD)/M=$(shell pwd)/g' test/Makefile
```

- `./include/uapi/linux/stat.h:30:17: error: expected ')' before numeric constant`

  `#define S_IRUSR 00400` エラーが発生する可能性があります。次のスクリーンショットを参照してください。

  カーネルビルドエラーの画像(/en-us/linux-virtual-delivery-agent/2106/media/kernel-building-error5.png)

  この問題を回避するには、次のコマンドを実行します。

```
sed -i 's/show_debug_output/debug_output_show/g' usb-vhci-iocifc.c usb-vhci-hcd.c



sed -i 's/store_debug_output/debug_output_store/g' usb-vhci-iocifc.c usb-vhci-hcd.c



sed -i 's/static DRIVER_ATTR(debug_output, S_IRUSR | S_IWUSR, debug_output_show, debug_output_store);/static DRIVER_ATTR_RW(debug_output);/g' usb-vhci-iocifc.c usb-vhci-hcd.c
```

- `./arch/x86/include/asm/uaccess.h:433:29: error: invalid initializer __typeof__(ptr) __pu_ptr = (ptr); \` エラーが発生する可能性があります。次のスクリーンショットを参照してください。

  カーネルビルドエラーの画像(/en-us/linux-virtual-delivery-agent/2106/media/kernel-building-error6.png)

  回避策として、`usb-vhci-iocifc.c` ファイルの 219 行目を `__put_user('\0', arg->bus_id);` から `__put_user('\0', arg->bus_id + 0);` に変更します。

- `error: 'access_ok' undeclared (first use in this function)`

`if(unlikely((_IOC_DIR(cmd) & _IOC_READ) && !access_ok(VERIFY_WRITE, arg, _IOC_SIZE(cmd))))` エラーが発生する可能性があります。次のスクリーンショットを参照してください。

カーネルビルドエラーの画像(/en-us/linux-virtual-delivery-agent/2106/media/kernel-building-error7.png)

この問題を回避するには、次のコマンドを実行します。

```
sed -i 's/VERIFY_READ, //g' usb-vhci-iocifc.c

sed -i 's/VERIFY_WRITE, //g' usb-vhci-iocifc.c
```

USB リダイレクトの問題のトラブルシューティング

このセクションの情報を使用して、Linux VDA の使用時に発生する可能性のあるさまざまな問題のトラブルシューティングを行います。

リダイレクトされた USB ディスクをアンマウントできない

Citrix Workspace アプリからリダイレクトされたすべての USB ディスクのアクセス制御のため、Linux VDA は、所有者のみがリダイレクトされたデバイスにアクセスできるように、これらすべてのデバイスを管理者権限で管理します。その結果、ユーザーは管理者権限なしではデバイスをアンマウントできません。

デバイスをアンマウントできないイメージ(/en-us/linux-virtual-delivery-agent/2106/media/unable-to-mount-disk.png)

USBディスクのリダイレクトを停止するとファイルが失われる

USBディスクをセッションにリダイレクトし、それを変更しようとした場合（たとえば、ディスク上にいくつかのファイルを作成した場合）、その後、Citrix Workspaceアプリのツールバーを使用してすぐにリダイレクトを停止すると、変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込む際に、システムがファイルシステム内のメモリキャッシュをマウントするために発生します。データはディスク自体には書き込まれません。Citrix Workspaceアプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。この問題を解決するには、USBリダイレクトを停止する前に、ターミナルでsyncコマンドを使用してデータをディスクにフラッシュしてください。

USBディスクのリダイレクトを停止するとファイルが失われるイメージ(/en-us/linux-virtual-delivery-agent/2106/media/file-lost-issue.png)

Citrix Workspaceアプリのツールバーにデバイスがない

Citrix Workspaceアプリのツールバーにデバイスが表示されないことがありますが、これはUSBリダイレクトが行われていないことを示します。この問題が発生した場合は、以下を確認してください。

- ポリシーがUSBリダイレクトを許可するように構成されている
- カーネルモジュールがカーネルと互換性がある

Workspaceアプリのツールバーにあるデバイスタブ(/en-us/linux-virtual-delivery-agent/2106/media/client-usb-redirection-troubleshoot.png)

> **注:**
>
> Citrix Workspaceアプリ for Linuxでは、**Devices**タブは利用できません。

Citrix Workspaceアプリのツールバーに*USB*デバイスが表示されるが、*ポリシー制限*と表示されてリダイレクトに失敗する

この問題が発生した場合は、次の操作を行います。

- リダイレクトを有効にするようにLinux VDAポリシーを構成します。
- Citrix Workspaceアプリのレジストリに追加のポリシー制限が構成されていないか確認します。レジストリパスの**DeviceRules**を確認し、この設定によってデバイスへのアクセスが拒否されていないことを確認します。

  HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Citrix\\ICA Client\\GenericUSB

USBデバイスは正常にリダイレクトされますが、セッションで使用できません

通常、サポートされているUSBデバイスのみがリダイレクトできます。その他のデバイスもアクティブなLinux VDAセッションにリダイレクトされる場合があります。リダイレクトされたすべてのデバイスについて、ユーザーが所有するノードがシステム **/dev** パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている（接続されている）がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限のポリシーに追加してください。

注:

USBドライブの場合、Linux VDAはディスクを構成してマウントします。ユーザー（およびそれをインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。これは、サポートされているデバイスリストにないデバイスには当てはまらない場合があります。

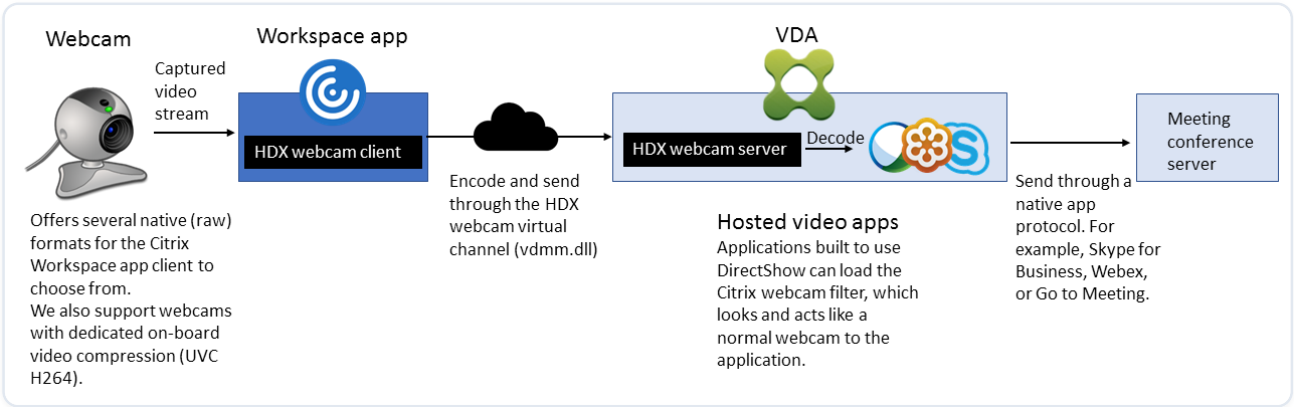
HDX™ウェブカメラビデオ圧縮

概要

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDXウェブカメラビデオ圧縮を使用してウェブカメラを利用できるようになりました。この機能はデフォルトで有効になっています。可能な場合は常にHDXウェブカメラビデオ圧縮を使用することをお勧めします。

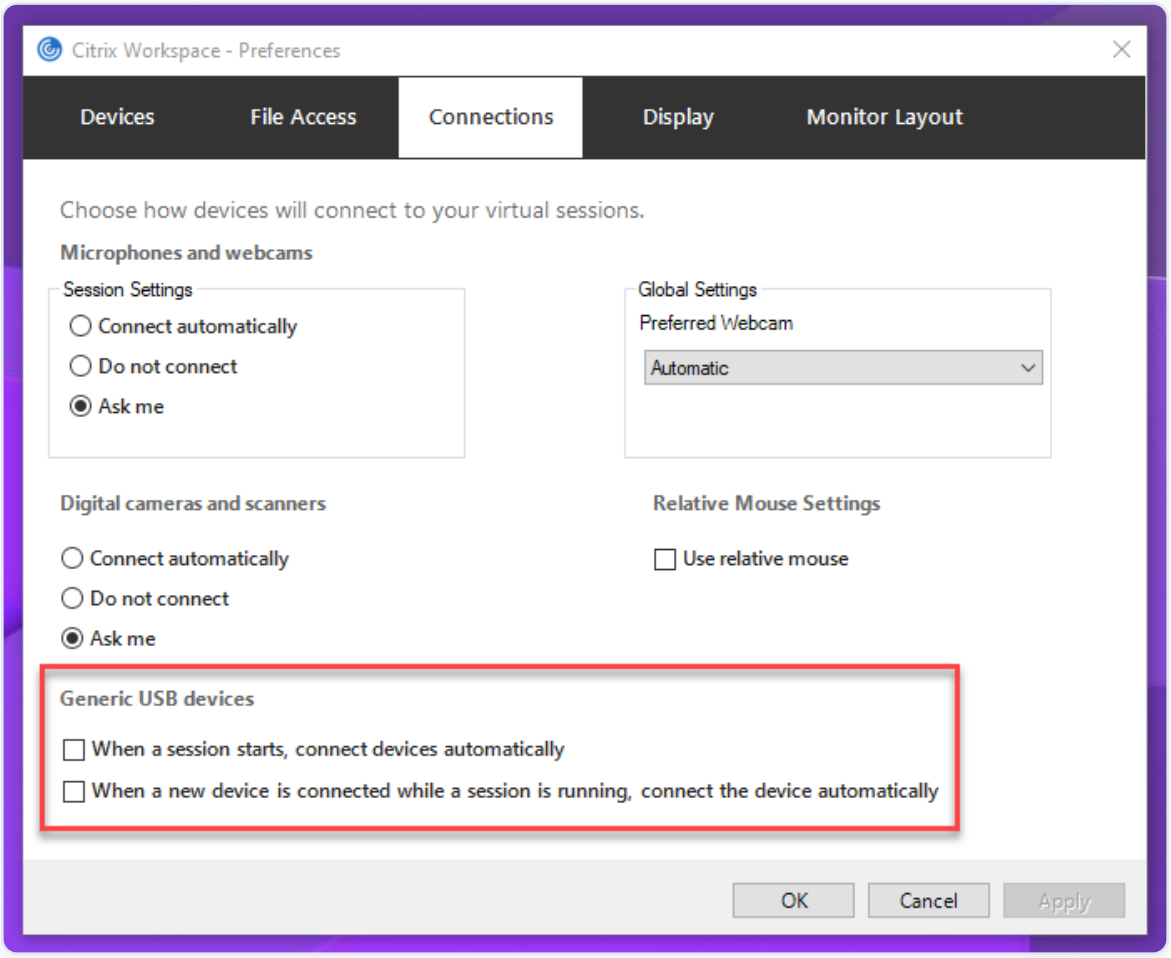
HDXウェブカメラビデオ圧縮は、**最適化された**ウェブカメラモードとも呼ばれます。このタイプのウェブカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDXウェブカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントはウェブカメラとの通信を処理します。その後、クライアントはビデオを適切に表示できるサーバーにのみ送信します。サーバーはウェブカメラを直接処理しませんが、その統合により、デスクトップで同じエクスペリエンスが得られます。Workspaceアプリは、帯域幅を節約し、WANシナリオでの回復力を向上させるためにビデオを圧縮します。



注:

- この機能は、Citrix WorkspaceアプリクライアントからのH.264ビデオのみをサポートします。
- サポートされているウェブカメラの解像度は、48x32から1920x1080の範囲です。
- ウェブカメラを使用しているときに、Citrix Workspace™アプリのツールバーから**汎用USBデバイス**を選択しないでください。そうしないと、予期しない問題が発生する可能性があります。



サポートされているLinuxディストリビューション

- RHEL 8.3/CentOS 8.3
- RHEL 7.9/CentOS 7.9
- ウブントゥ 20.04
- ウブントゥ 18.04
- ウブントゥ 16.04
- デビアン 10

完全にテストされたウェブカメラ

ウェブカメラによってフレームレート、明るさ、コントラストのレベルが異なります。Citrix®は、初期機能検証のために以下のウェブカメラを使用しています。

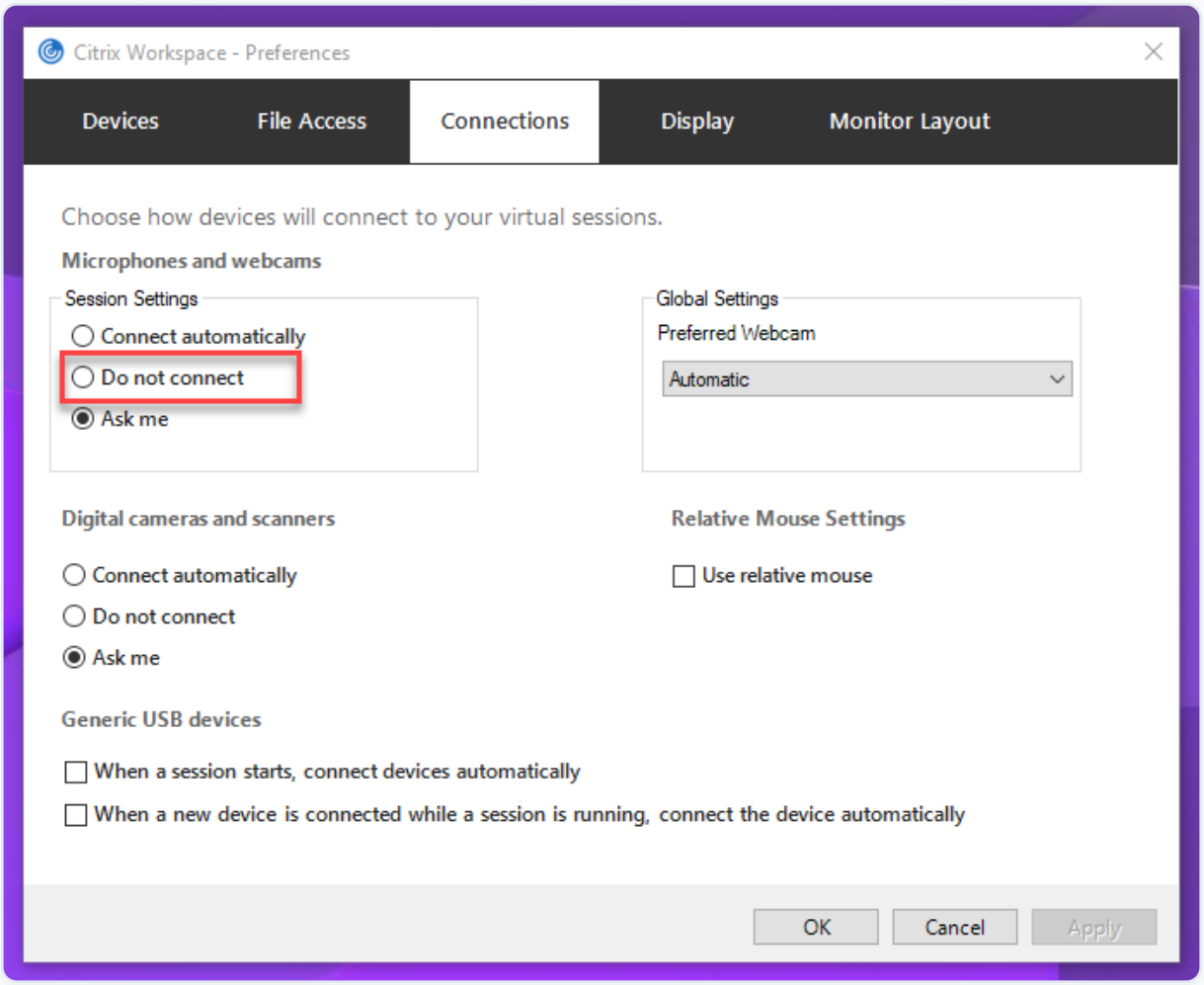
- ロジクール HD ウェブカメラ C270
- ロジクール ウェブカメラ C930e
- マイクロソフト ライフカム HD3000

構成

この機能はデフォルトで有効になっています。これを使用するには、以下の検証と構成を完了してください。

ヒント:

Citrix Workspaceアプリのユーザーは、Desktop Viewerの**マイクとWebカメラ**設定で、**接続しない**を選択することにより、デフォルト設定を上書きできます。



1. VDAのインストールが完了したら、VDAがDelivery Controller™に登録できること、および公開されたLinuxデスクトップセッションがWindows資格情報を使用して正常に起動できることを確認します。
2. VDAがインターネットにアクセスできることを確認し、その後、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行してWebカメラの構成を完了します。VDAがインターネットにアクセスできない場合は、手順3に進んでください。

   VDAがDebian 10に展開されている場合は、最新のカーネルバージョンで実行されていることを確認してください。そうでない場合は、以下のコマンドを実行して最新のカーネルバージョンに更新してください。

   ```
   sudo apt-get update

   sudo apt-get dist-upgrade

   sudo reboot
   ```

   ctxwcamcfg.shスクリプトは、次のことを支援します。

   a. VDAに`FFmpeg`、`kernel-devel`、およびDynamic Kernel Module Support (DKMS) プログラムをインストールします。

      - `FFmpeg`はビデオデコードに使用されます。
      - `kernel-devel`は、対応するバージョンの仮想Webカメラカーネルモジュールを構築するために使用されます。
      - DKMSは、仮想Webカメラカーネルモジュールを動的に管理するために使用されます。

      > **注:**
      >
      > RHELおよびCentOSに上記のプログラムをインストールする場合、`ctxwcamcfg.sh`スクリプトは、VDAに以下のリポジトリをインストールして有効にします。
      >
      > - エンタープライズ Linux 用追加パッケージ (EPEL)
      > - RPM フュージョン

b. `<https://github.com/umlaeute/v4l2loopback>`から`v4l2loopback`オープンソースコードをダウンロードし、DKMSを使用して`v4l2loopback`を管理します。

   `v4l2loopback`は、V4L2ループバックデバイスを作成できるカーネルモジュールです。

c. `sudo service ctxwcamsd restart`コマンドを実行します。Linux VDAのWebカメラサービス（`ctxwcamsd`）が再起動し、HDX Webカメラビデオ圧縮機能のために`v4l2loopback`カーネルモジュールをロードします。

3. VDAがインターネットにアクセスできない場合は、別のマシンでv4l2loopbackカーネルモジュールをビルドし、その後、VDAにコピーします。

   a. インターネットアクセスがあり、VDAと同じカーネルバージョンを持つビルドマシンを準備します。`uname -r`コマンドはカーネルバージョンを見つけるのに役立ちます。

   b. ビルドマシンで、`sudo mkdir -p /var/xdl`コマンドを実行します。

   c. VDAから`/var/xdl/configure_*`をビルドマシンの`/var/xdl/`にコピーします。

   d. ビルドマシンで、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行してカーネルモジュールをビルドします。コマンドが正常に実行されると、`/var/lib/dkms/v4l2loopback/1.81b8df79107d1fbf392fdcbaa051bd227a9c94c1/$(uname -r)/x86_64/module/`パスの下にv4l2loopback.koファイルが作成されます。ctxwcamcfg.shスクリプトの実行時に発生する可能性のあるエラーは無視してください。

   e. v4l2loopback.koをビルドマシンからVDAにコピーし、`/opt/Citrix/VDA/lib64/`の下に配置します。

   f. VDAで、`sudo service ctxwcamsd restart`コマンドを実行してWebカメラサービスを再起動し、v4l2loopbackカーネルモジュールをロードします。

   g. HDX Webカメラビデオ圧縮機能を使用する前に、VDAにFFmpegプログラムをインストールしてください。

セッションの信頼性を構成する

Citrix®は、サポートされているすべてのLinuxプラットフォームにセッションの信頼性機能を提供します。セッションの信頼性はデフォルトで有効になっています。

セッションの信頼性により、ネットワークの中断が発生してもICAセッションがシームレスに再接続されます。セッションの信頼性の詳細については、「自動クライアント再接続とセッションの信頼性」を参照してください。

注: セッションの信頼性接続を介して送信されるデータは、デフォルトでプレーンテキストです。セキュリティ上の理由から、TLS暗号化を有効にすることをお勧めします。TLS暗号化の詳細については、「TLSを使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studioでのポリシー設定

Citrix Studioでセッションの信頼性について、以下のポリシーを設定できます。

- セッションの信頼性接続
- セッションの信頼性タイムアウト
- セッションの信頼性ポート番号
- 再接続UIの透過レベル

詳細については、「セッションの信頼性ポリシー設定」および「自動クライアント再接続ポリシー設定」を参照してください。

注: **セッションの信頼性接続**または**セッションの信頼性ポート番号**ポリシーを設定した後、設定を有効にするには、VDAサービスとHDX™サービスをこの順序で再起動してください。

Linux VDAでの設定

- **セッションの信頼性TCPリスナーの有効化/無効化**

デフォルトでは、セッションの信頼性TCPリスナーは有効になっており、ポート2598でリッスンしています。リスナーを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "fEnableWinStation" -d "0x00000000"
```

注: 設定を有効にするには、HDXサービスを再起動してください。TCPリスナーを無効にしても、セッションの信頼性は無効になりません。**セッションの信頼性接続**ポリシーを通じて機能が有効になっている場合、セッションの信頼性は他のリスナー（SSLなど）を介して引き続き利用できます。

- **セッションの信頼性ポート番号**

次のコマンドを使用して、セッションの信頼性ポート番号を設定することもできます（例としてポート番号2599を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "PortNumber" -d  "2599"
```

注: 設定を有効にするには、HDXサービスを再起動してください。Citrix Studioのポリシー設定でポート番号が設定されている場合、Linux VDAでの設定は無視されます。VDA上のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように構成されていることを確認してください。

- **サーバーからクライアントへのキープアライブ間隔**

セッションにアクティビティがない場合（例：マウスの動きがない、画面の更新がない）、Linux VDAとICA®クライアント間でセッション信頼性キープアライブメッセージが送信されます。キープアライブメッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントからの応答がない場合、クライアントが再接続するまでセッションは中断されます。この設定は、連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpServerToClientKeepAlive"  -d  "10"  --force
```

- **クライアントからサーバーへのキープアライブ間隔**

この設定は、ICAクライアントからLinux VDAに送信される連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpClientToServerKeepAlive"  -d  "10"  --force
```

トラブルシューティング

ポリシー設定でセッション信頼性を有効にした後、セッションを起動できない。

この問題を回避するには、次の手順を実行します。

1. Citrix Studioのポリシー設定でセッション信頼性を有効にした後、VDAサービスとHDXサービスをこの順序で再起動してください。
2. VDAで、次のコマンドを実行して、セッション信頼性TCPリスナーが実行されていることを確認します（例としてポート2598を使用）。

   ```
   netstat -an | grep 2598
   ```

   セッション信頼性ポートにTCPリスナーがない場合は、次のコマンドを実行してリスナーを有効にします。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations
   \cgp"   -v  "fEnableWinStation" -d "0x00000001"
   ```

ソフトキーボード

ソフトキーボード機能は、Linux仮想デスクトップまたはアプリケーションセッションで利用できます。ソフトキーボードは、入力フィールドに入ると自動的に表示され、入力フィールドを離れると非表示になります。

ソフトキーボードの画像(/en-us/linux-virtual-delivery-agent/2106/media/soft-keyboard.png)

注:

この機能は、RHEL 7.8、RHEL 7.9、RHEL 8.1、RHEL 8.2、RHEL 8.3、SUSE 12.5、Ubuntu 16.04、Ubuntu 18.04、およびUbuntu 20.04で利用できます。Citrix Workspaceアプリ（iOSおよびAndroid用）でサポートされています。

機能の有効化と無効化

この機能はデフォルトで無効になっています。この機能を有効または無効にするには、**ctxreg** ユーティリティを使用します。特定のLinux VDAでの機能構成は、そのVDA上のすべてのセッションに適用されます。

この機能を有効にするには：

1. コマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000001"
```

2. Citrix Studioで、**Automatic keyboard display**ポリシーを**Allowed**に設定します。
3. （オプション）RHEL 7およびCentOS 7の場合、Intelligent Input Bus (`IBus`) をデフォルトのIMサービスとして構成するには、次のコマンドを実行します。

```
echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

注:

前述の設定は、新しいセッションにログオンするか、現在のセッションからログオフして再度ログオンしたときに有効になります。

制限事項

- この機能は、Google Chrome、LibreOffice、およびその他のアプリでは、期待どおりに動作しない場合があります。
- ソフトキーボードを手動で非表示にした後、再度表示するには、入力フィールド以外の場所をクリックしてから、現在の入力フィールドを再度クリックします。
- Webブラウザで1つの入力フィールドから別の入力フィールドにクリックしても、ソフトキーボードが表示されない場合があります。この問題を回避するには、入力フィールド以外の場所をクリックしてから、目的の入力フィールドをクリックします。
- この機能は、Unicode文字および2バイト文字（中国語、日本語、韓国語などの文字）をサポートしていません。
- パスワード入力フィールドでは、ソフトキーボードは利用できません。

- ソフトキーボードが現在の入力フィールドと重なる場合があります。この場合、アプリのウィンドウを移動するか、画面をスクロールして入力フィールドにアクセスしやすい位置に移動してください。
- Citrix Workspace™アプリとHuaweiタブレット間の互換性の問題により、物理キーボードが接続されている場合でも、Huaweiタブレットではソフトキーボードが表示されます。

クライアント入力メソッドエディター (IME)

概要

中国語、日本語、韓国語などの2バイト文字は、IMEを介して入力する必要があります。クライアント側でCitrix Workspace™アプリと互換性のある任意のIME（WindowsネイティブのCJK IMEなど）を使用して、これらの文字を入力します。

インストール

この機能は、Linux VDAをインストールすると自動的にインストールされます。

使用方法

通常どおり、Citrix Virtual AppsまたはCitrix Virtual Desktops™セッションを開きます。

クライアントIME機能の使用を開始するには、クライアント側で必要に応じて入力方式を変更します。

既知の問題

- Googleスプレッドシートのセルに文字を入力するためにクライアントIME機能を使用する前に、そのセルをダブルクリックする必要があります。
- クライアントIME機能は、パスワードフィールドでは自動的に無効になりません。
- IMEユーザーインターフェイスは、入力領域でカーソルに追従しません。

複数言語入力のサポート

Linux VDAバージョン1.4以降、Citrixは公開アプリケーションのサポートを追加しました。ユーザーはLinuxデスクトップ環境なしで、目的のLinuxアプリケーションにアクセスできます。

しかし、Linux VDAのネイティブ言語バーは、Linuxデスクトップ環境と高度に統合されているため、公開アプリケーションでは利用できませんでした。その結果、ユーザーは中国語、日本語、

韓国語などのIMEを必要とする言語でテキストを入力できませんでした。また、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け付ける公開アプリケーションに言語バーを提供します。言語バーを使用すると、ユーザーはサーバー側のIMEを選択し、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

ctxregユーティリティを使用して、この機能を有効または無効にできます（デフォルトでは無効）。特定のLinux VDAサーバーでの機能構成は、そのVDAで公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \\SYSTEM\\CurrentControlSet\\Control\\Citrix\\LanguageBar」で、タイプはDWORDです。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE
\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d
"0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE
\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d
"0x00000000"
```

使用方法

使用方法は簡単です。

1. 機能を有効にします。
2. テキスト入力を受け付ける公開アプリケーションにアクセスします。セッション中に、アプリケーションの横に言語バーが表示されます。
3. ドロップダウンメニューから、**地域と言語**を選択して、目的の言語（入力ソース）を追加します。

   地域と言語の画像(/en-us/linux-virtual-delivery-agent/2106/media/select-region-language-dialog-box.png)

4. ドロップダウンメニューからIMEまたはキーボードレイアウトを選択します。
5. 選択したIMEまたはキーボードレイアウトを使用して言語を入力します。

> **注:**
>
> - VDA側の言語バーでキーボードレイアウトを変更する際は、クライアント側（Citrix Workspace™アプリを実行中）でも同じキーボードレイアウトが使用されていることを確認してください。
> - **Region & Language**ダイアログボックスで設定を行う前に、**accountsservice**パッケージをバージョン0.6.37以降にアップグレードする必要があります。

IMEまたはキーボードレイアウトを選択するイメージ(/en-us/linux-virtual-delivery-agent/2106/media/region-language-dialog-box.png)

動的キーボードレイアウト同期

以前は、Linux VDAとクライアントデバイスのキーボードレイアウトは同じである必要がありました。例えば、クライアントデバイスでキーボードレイアウトが英語からフランス語に変更されても、VDAでは変更されない場合、キーマッピングの問題が発生し、VDAもフランス語に変更されるまでその問題が解消されない可能性がありました。

Citrix®は、VDAのキーボードレイアウトをクライアントデバイスのキーボードレイアウトと自動的に同期させることで、この問題に対処します。クライアントデバイスのキーボードレイアウトが変更されるたびに、VDAのレイアウトもそれに追従します。

> **注:**
>
> Citrix Workspace™アプリ for HTML5は、動的キーボードレイアウト同期機能をサポートしていません。

構成

動的キーボードレイアウト同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**クライアントキーボードレイアウト同期とIMEの改善**ポリシーを設定するか、`ctxreg`ユーティリティを介してレジストリを編集します。

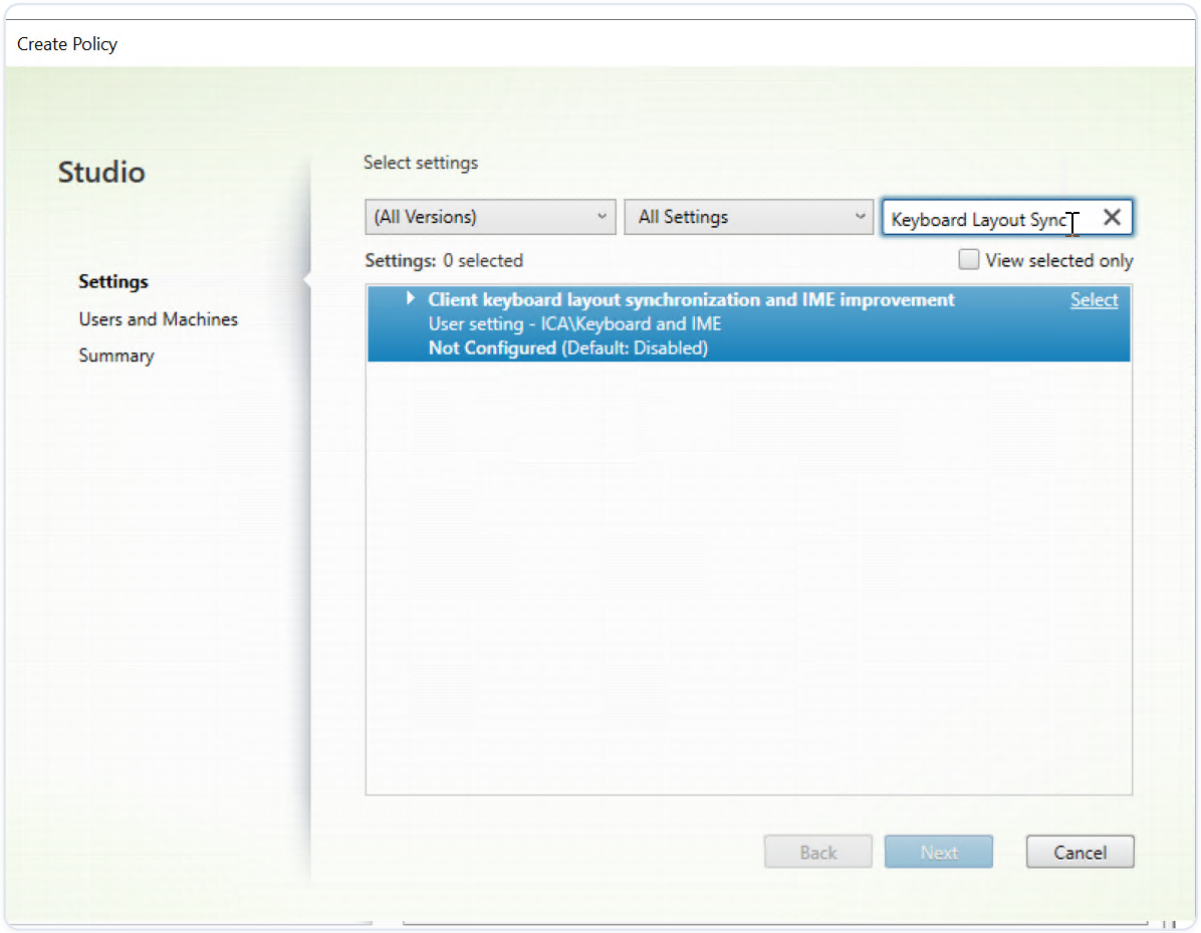
注:

クライアントキーボードレイアウト同期とIMEの改善ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- 動的キーボードレイアウト同期機能を有効または無効にするには、**クライアントキーボードレイアウト同期とIMEの改善**ポリシーを設定します。

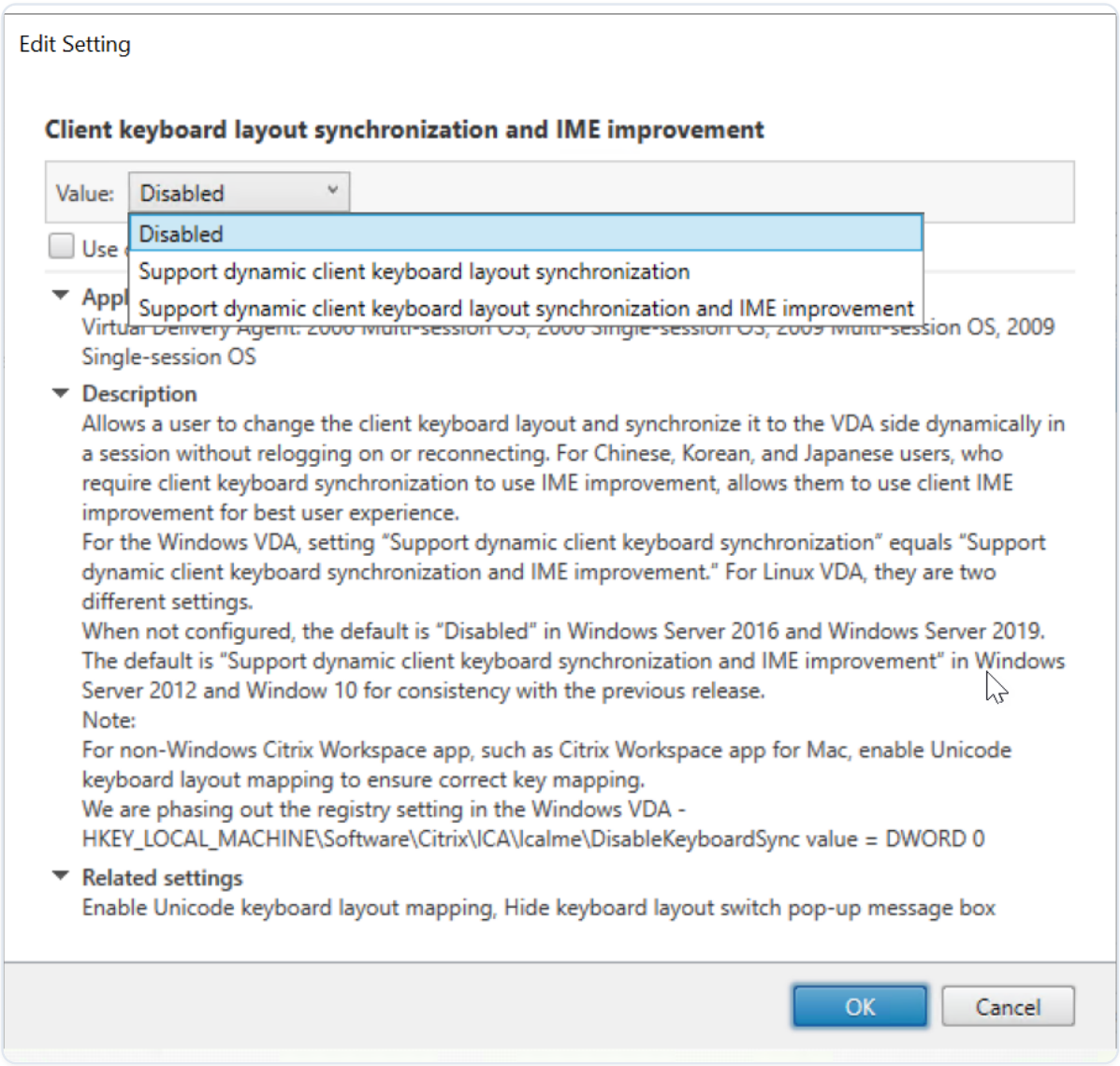
  a. Studioで、**ポリシー**を右クリックし、**ポリシーの作成**を選択します。

  b. **クライアントキーボードレイアウト同期とIMEの改善**ポリシーを検索します。



  c. ポリシー名の横にある**選択**をクリックします。

  d. ポリシーを設定します。



3つのオプションがあります。

- **無効**: 動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期を無効にします。
- **動的クライアントキーボードレイアウト同期をサポート**:
  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期を有効にします。
- **動的クライアントキーボードレイアウト同期とIMEの改善をサポート**：
  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期の両方を有効にします。

- 動的キーボードレイアウト同期機能を有効または無効にするには、`ctxreg`ユーティリティを介してレジストリを編集します。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000001"
  ```

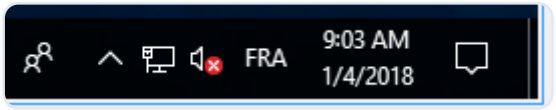
  この機能を無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000000"
  ```

使用方法

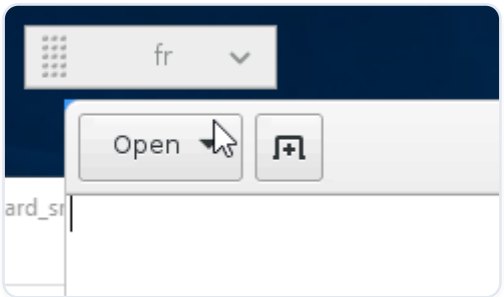
この機能を有効にすると、セッション中にクライアントデバイスのキーボードレイアウトが変更された場合、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語（FR）に変更した場合：



すると、Linux VDAセッションのキーボードレイアウトも「fr」に変更されます。

アプリケーションセッションでは、言語バーを有効にしている場合、この自動変更を確認できます。



デスクトップセッションでは、タスクバーでこの自動変更を確認できます。



クライアントIMEユーザーインターフェイスの同期

概要

これまで、クライアントIMEユーザーインターフェイス（変換ウィンドウと候補ウィンドウを含む）は画面の左上隅に配置されていました。カーソルに追従せず、テキスト入力領域でカーソルから離れた場所に表示されることがありました。

以前のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/2106/media/composition-and-candidate-windows.png)

Citrix®は、クライアントIMEの使いやすさを向上させ、ユーザーエクスペリエンスを次のようにさらに改善します。

現在のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/2106/media/synced-composition-and-candidate-windows.png)

> **注:**

> この機能は、RHEL 7.x、CentOS 7.x、Ubuntu 16.04、Ubuntu 18.04、およびSUSE 12.xで利用できます。Citrix Workspaceアプリ（Windows版およびMac版）でサポートされています。

> RHEL 7.xデスクトップセッションでこの機能を使用するには、`IBus`を有効にする必要があります。たとえば、ユーザーインターフェイス言語をIME入力が必要な言語に設定するか、**${HOME}/.config/imsettings/xinputrc**ファイルに**GTK_IM_MODULE=ibus**を追加します。

> この機能は自動的にインストールされますが、使用する前に有効にする必要があります。

機能の有効化と無効化

クライアントIMEユーザーインターフェイス同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、`ctxreg`ユーティリティを介してレジストリを編集します。

> **注:**

> **Client Keyboard Layout Sync and IME Improvement**ポリシーはレジストリ設定よりも優先され、指定したユーザーオブジェクトおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- クライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定します。

  a. Studioで、**Policies**を右クリックし、**Create Policy**を選択します。

  b. **Client Keyboard Layout Sync and IME Improvement**ポリシーを検索します。

  クライアントキーボードレイアウト同期とIME改善ポリシーの画像(/en-us/linux-virtual-delivery-agent/2106/media/client-keyboard-and-ime-policy.png)

  c. ポリシー名の横にある「**選択**」をクリックします。

  d. ポリシーを設定します。

  クライアントキーボードレイアウト同期とIME改善ポリシーの設定の画像(/en-us/linux-virtual-delivery-agent/2106/media/setting-client-keyboard-and-ime-policy.png)

  3つのオプションがあります。

  - **無効**: 動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期を無効にします。
  - **動的クライアントキーボードレイアウト同期をサポート**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期を有効にします。
  - **動的クライアントキーボードレイアウト同期とIME改善をサポート**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期の両方を有効にします。

- クライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、`ctxreg`ユーティリティを介してレジストリを編集します。

  この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
" -v  "SyncClientIME" -d "0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
" -v "SyncClientIME" -d "0x00000000"
```

HDX™ インサイト

概要

Linux VDA は HDX Insight 機能を部分的にサポートしています。HDX Insight は Citrix Application Delivery Management (ADM) の一部であり、一般的な業界標準である AppFlow に基づいています。これにより、Citrix ADC または Citrix SD-WAN アプリケーションネットワークファブリックを通過する Citrix ICA トラフィックに対する前例のないエンドツーエンドの可視性を提供することで、IT 部門は優れたユーザーエクスペリエンスを提供できます。詳細については、「HDX Insight」を参照してください。(/ja-jp/citrix-application-delivery-management-service/analytics/hdx-insight.html)

インストール

依存パッケージをインストールする必要はありません。

使用方法

HDX Insight は、Citrix Workspace™ アプリと Linux VDA の間で Citrix ADC を介して渡される ICA® メッセージを分析します。すべての HDX Insight データは NSAP 仮想チャネルから取得され、非圧縮で送信されます。NSAP 仮想チャネルはデフォルトで有効になっています。

次のコマンドは、NSAP 仮想チャネルをそれぞれ無効化および有効化します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000000" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000001" --force
```

トラブルシューティング

データポイントが表示されない

2つの原因が考えられます。

- HDX Insight が正しく構成されていません。

  たとえば、Citrix ADC で AppFlow® が有効になっていないか、Citrix ADM で誤った Citrix ADC インスタンスが構成されています。

- Linux VDA で ICA Control Virtual Channel が開始されていません。

  `ps aux | grep -i ctxctl`

  `ctxctl` が実行されていない場合は、管理者に連絡して Citrix にバグを報告してください。

アプリケーションデータポイントが表示されない

シームレス仮想チャネルが有効になっており、シームレスアプリケーションがしばらく起動していることを確認してください。

Rendezvous プロトコル

Citrix Gatewayサービスを使用する環境では、Rendezvousプロトコルにより、HDXセッションがCitrix Cloud Connector™をバイパスし、Citrix Gatewayサービスに直接かつ安全に接続できるようになります。

要件:

- Citrix Workspace™およびCitrix Gatewayサービスを使用した環境へのアクセス。
- コントロールプレーン：Citrix Virtual Apps and Desktops™ サービス (Citrix Cloud)。
- Linux VDAバージョン2012以降。
- CitrixポリシーでRendezvousプロトコルを有効にします。詳しくは、「Rendezvousプロトコルポリシー設定」を参照してください。
- VDAは、すべてのサブドメインを含む`https://*.nssvc.net`にアクセスできる必要があります。その方法ですべてのサブドメインをホワイトリストに登録できない場合は、代わりに

`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳しくは、Citrix Cloudドキュメント（Virtual Apps and Desktopsサービスの下）の「インターネット接続要件」セクションと、Knowledge Centerの記事「CTX270584」を参照してください。

- Cloud Connectorは、セッションを仲介する際にVDAのFQDNを取得する必要があります。この目的を達成するには、サイトのDNS解決を有効にします。Citrix Virtual Apps and Desktops Remote PowerShell SDKを使用して、コマンド`Set-BrokerSite -DnsResolutionEnabled $true`を実行します。Citrix Virtual Apps and Desktops Remote PowerShell SDKについて詳しくは、「SDKとAPI」を参照してください。

> **重要:**

> Rendezvousプロトコルは、透過型プロキシまたは明示的プロキシをサポートしていません。プロキシを使用するには、ICA®トラフィックにCloud Connectorを引き続き使用してください。

> Rendezvousを有効にしていて、VDAがCitrix Gatewayサービスに直接到達できない場合、VDAはCloud Connectorを介してHDX™セッションをプロキシするようにフォールバックします。

すべての要件を満たしている場合は、Rendezvousが使用されているかどうかを検証するために、次の手順に従います。

1. VDAでターミナルを起動します。
2. `su root -c "/opt/Citrix/VDA/bin/ctxquery -f iuStdP"`を実行します。
3. TRANSPORT PROTOCOLSは接続の種類を示します。
    - TCPランデブー: TCP - SSL - CGP - ICA
    - EDTランデブー: UDP - DTLS - CGP - ICA
    - Cloud Connectorを介したプロキシ: TCP - CGP - ICA

この図は、Rendezvous接続フローの概要を示しています。手順に従ってフローを理解してください。

Rendezvousプロトコルの概要(/en-us/linux-virtual-delivery-agent/2106/media/rendezvous-overview.png)

1. Citrix Workspaceに移動します。

2. Citrix Workspaceで資格情報を入力します。
3. オンプレミスのActive Directoryを使用している場合、Citrix Virtual Apps™ and Desktopsサービスは、Cloud Connectorチャネルを使用してActive Directoryで資格情報を認証します。
4. Citrix Workspaceは、Citrix Virtual Apps and Desktopsサービスから列挙されたリソースを表示します。
5. Citrix Workspaceからリソースを選択します。Citrix Virtual Apps and Desktopsサービスは、受信セッションの準備をするためにVDAにメッセージを送信します。
6. Citrix Workspaceは、Citrix Cloudによって生成されたSTAチケットを含むICAファイルをエンドポイントに送信します。
7. エンドポイントはCitrix Gatewayサービスに接続し、VDAに接続するためのチケットを提供し、Citrix Cloudがチケットを検証します。
8. Citrix Gatewayサービスは、Cloud Connectorに接続情報を送信します。Cloud Connectorは、接続がRendezvous接続であるべきかどうかを判断し、その情報をVDAに送信します。
9. VDAはCitrix Gatewayサービスへの直接接続を確立します。
10. VDAとCitrix Gatewayサービス間の直接接続が不可能な場合、VDAはCloud Connectorを介して接続をプロキシします。
11. Citrix Gatewayサービスは、エンドポイントデバイスとVDA間の接続を確立します。
12. VDAは、Cloud Connectorを介してCitrix Virtual Apps and Desktopsサービスでライセンスを検証します。
13. Citrix Virtual Apps and Desktopsサービスは、Cloud Connectorを介してVDAにセッションポリシーを送信します。これらのポリシーが適用されます。

アダプティブトランスポート

アダプティブトランスポートは、Citrix Virtual Apps and Desktops のデータ転送メカニズムです。これは、より高速でスケーラブルであり、アプリケーションの対話性を向上させ、困難な長距離WANおよびインターネット接続においてよりインタラクティブです。アダプティブトランスポートの詳細については、アダプティブトランスポートを参照してください。

アダプティブトランスポートを有効にする

Citrix Studioで、**HDX™ Adaptive Transport**ポリシーが**Preferred**または**Diagnostic mode**に設定されていることを確認します。**Preferred**はデフォルトで選択されています。

- **Preferred**: 可能であればEnlightened Data Transport (EDT) を介したアダプティブトランスポートが使用され、TCPにフォールバックします。

- **Diagnostic mode**: EDTが強制的にオンになり、TCPへのフォールバックは無効になります。

診断モードの画像(/en-us/linux-virtual-delivery-agent/2106/media/diagnostic-mode.jpg)

アダプティブトランスポートを無効にする

アダプティブトランスポートを無効にするには、Citrix Studioで**HDX Adaptive Transport**ポリシーを**Off**に設定します。

アダプティブトランスポートが有効になっているか確認する

UDPリスナーが実行されているか確認するには、次のコマンドを実行します。

```
netstat -an | grep "1494\|2598"
```

通常の状況では、出力は次のようになります。

```
udp        0      0 0.0.0.0:2598            0.0.0.0:*



udp       0      0 :::1494                 :::*
```

EDT MTU検出

EDTは、セッション確立時に最大転送単位（MTU）を自動的に決定します。これにより、パフォーマンスの低下やセッション確立の失敗につながる可能性のあるEDTパケットの断片化を防ぎます。

最小要件:

- リナックス VDA 2012
- Citrix Workspace™ アプリ 1911 Windows 用
- シトリックス ADC:
  - 13.0.52.24
  - 12.1.56.22
- セッションの信頼性を有効にする必要があります

この機能をサポートしていないクライアントプラットフォームまたはバージョンを使用している場合は、環境に適したカスタムEDT MTUを構成する方法の詳細について、Knowledge Centerの記事CTX231821を参照してください。

> **警告:**
>
> レジストリを誤って編集すると、オペレーティングシステムの再インストールが必要になるような深刻な問題が発生する可能性があります。Citrixは、レジストリエディターの誤った使用によって生じる問題が解決できることを保証できません。レジストリエディターは、ご自身の責任において使用してください。編集する前に、必ずレジストリをバックアップしてください。

VDAでEDT MTU検出を有効または無効にする

EDT MTU検出はデフォルトで無効になっています。

- EDT MTU検出を有効にするには、次のコマンドを使用して`MtuDiscovery`レジストリキーを設定し、VDAを再起動して、VDAが登録されるのを待ちます。

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t
  "REG_DWORD" -v "MtuDiscovery" -d "0x00000001" --force
  ```

- EDT MTU検出を無効にするには、`MtuDiscovery`レジストリ値を削除します。

この設定はマシン全体に適用され、サポートされているクライアントから接続するすべてのセッションに影響します。

クライアントでEDT MTU検出を制御する

ICA®ファイルに`MtuDiscovery`パラメーターを追加することで、クライアントでEDT MTU検出を選択的に制御できます。この機能を無効にするには、`Application`セクションの下に以下を設定します。

```
MtuDiscovery=Off
```

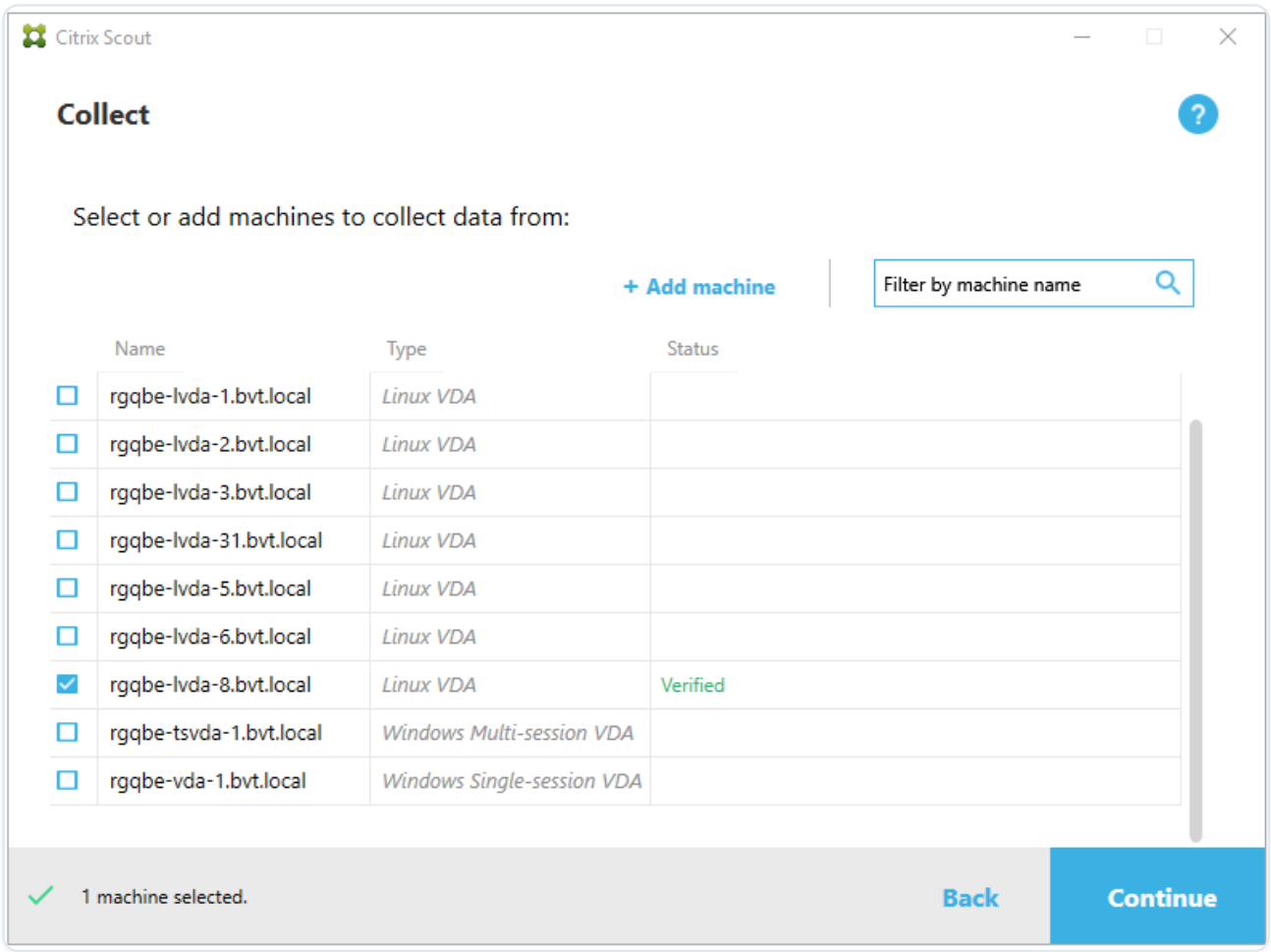
この機能を再度有効にするには、ICAファイルから`MtuDiscovery`パラメーターを削除します。

> **重要:**

このICAファイルパラメーターが機能するには、VDAでEDT MTU検出を有効にします。VDAでEDT MTU検出が有効になっていない場合、ICAファイルパラメーターは効果がありません。

Citrix Telemetry Serviceとの統合

Citrix Telemetry Service (ctxtelemetry) がLinux VDAソフトウェアと統合されている場合、Citrix Scoutを実行できます。これにより、/opt/Citrix/VDA/bin/xdlcollect.shスクリプトを使用してLinux VDAに関するログが収集されます。



注:

Linux VDA 1912以前のバージョンからアップグレードした後、Citrix Telemetry Service (ctxtelemetry) の変数を構成するには、/opt/Citrix/VDA/sbin/ctxsetup.shを再実行する必要があります。変数について詳しくは、「Easy install」を参照してください。

Citrix Telemetry Serviceの有効化と無効化

- サービスを有効にするには、**sudo systemctl enable ctxtelemetry.socket**コマンドを実行します。
- サービスを無効にするには、**sudo systemctl disable ctxtelemetry.socket**を実行します。

ポート

Citrix Telemetry Service (ctxtelemetry) は、デフォルトでTCP/IPポート7503を使用してCitrix Scoutをリッスンします。Delivery ControllerではTCP/IPポート7502を使用してCitrix Scoutと通信します。

Linux VDAをインストールする際に、デフォルトのポートを使用するか、以下の変数を通じてポートを変更できます。

- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutをリッスンするためのソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトポートは7502です。

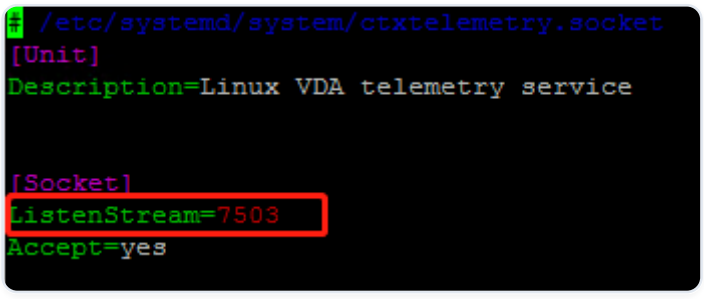
VDAのインストール後にポートを変更するには、次の手順を実行します。

1. Scoutと通信するためのポートを変更するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\Software\Citrix\VirtualDesktopAgent" -v "TelemetryServicePort" -d
   <port number> -t REG_DWORD
   ```

2. Scoutをリッスンするためのソケットポートを変更するには、次のコマンドを実行してctxtelemetry.socketファイルを開いて編集します。

   ```
   sudo vi /etc/systemd/system/ctxtelemetry.socket
   ```

3. ソケットポートを再起動するには、次のコマンドを実行します。

```
sudo systemctl daemon-reload

sudo systemctl stop ctxtelemetry.socket

sudo systemctl start ctxtelemetry.socket
```

4. ファイアウォール構成で新しいポートを有効にします。

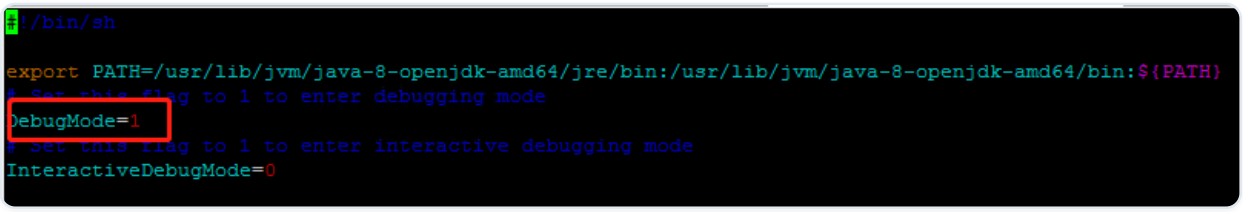
   たとえば、Ubuntuディストリビューションを使用している場合は、**sudo ufw allow 7503**コマンドを実行してポート7503を有効にします。

デバッグモード

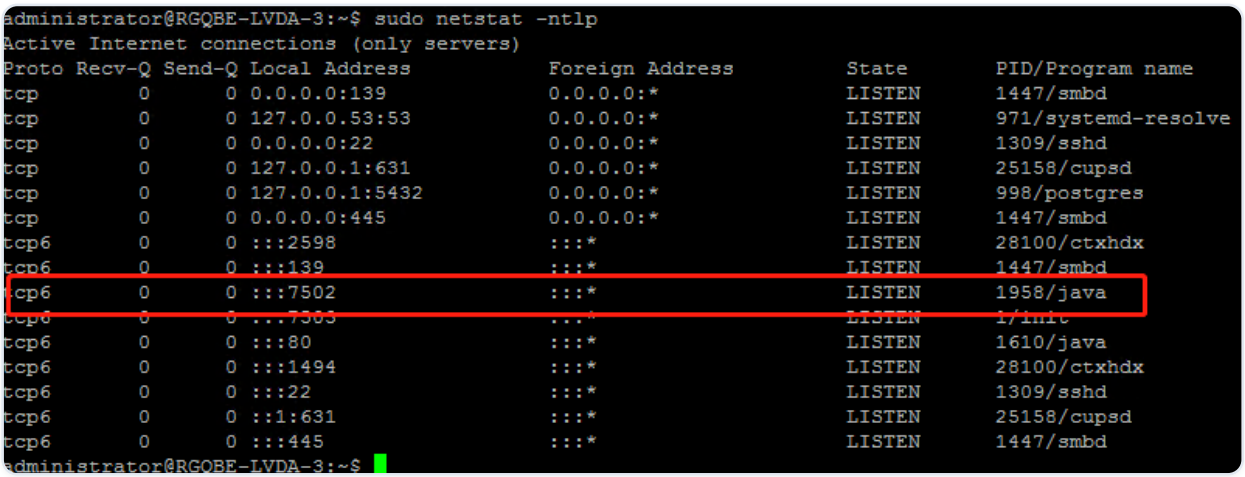
Citrix Telemetry Serviceが期待どおりに動作しない場合は、デバッグモードを有効にして原因を特定できます。

1. デバッグモードを有効にするには、次のコマンドを実行してctxtelemetryファイルを開き、DebugModeの値を1に変更します。

```
sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
```



2. Citrix Telemetry Serviceを手動で停止するか、サービスが自動的に停止するまで15分間待ちます。



この例では、次のコマンドを実行してCitrix Telemetry Serviceを停止できます。

```
sudo netstat -ntlp

Kill -9 1958
```

3. Citrix Telemetry Serviceを再起動するには、ScoutでLinux VDAを選択し、/var/log/xdl/にあるtelemetry-debug.logを見つけます。

サービスの待機時間

ソケットポートを開く`systemd`デーモンはデフォルトで起動し、リソースをほとんど使用しません。Citrix Telemetry Serviceはデフォルトで停止しており、Delivery Controllerからのログ収集要求があった場合にのみ起動します。ログ収集が完了すると、サービスは15分間新しい収集要求を待ち、要求がない場合は再度停止します。待機時間は、次のコマンドで構成できます。最小値は10分です。10分未満の値を設定した場合、最小値である10分が適用されます。待機時間を設定したら、サービスを停止して再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"TelemetryServiceIdleTimeoutInMinutes" -d <number> -t REG_DWORD
```

検証テスト

収集が開始される前に、選択された各マシンに対して検証テストが自動的に実行されます。これらのテストは、要件が満たされていることを確認します。マシンのテストが失敗した場合、Scoutは推奨される修正措置とともにメッセージを表示します。検証テストの詳細については、Citrix Scoutドキュメントの「検証テスト」セクションを参照してください。

トレースの有効化

概要

ログの収集と問題の再現は、診断を遅らせ、ユーザーエクスペリエンスを低下させます。トレースの有効化機能は、このような作業を軽減します。Linux VDAでは、トレースはデフォルトで有効になっています。

構成

`ctxlogd` デーモンと `setlog` ユーティリティは、Linux VDAリリースパッケージに同梱されるようになりました。デフォルトでは、Linux VDAのインストールと構成後に `ctxlogd` デーモンが起動します。

ctxlogd デーモン

トレースされる他のすべてのサービスは、`ctxlogd` デーモンに依存しています。Linux VDAのトレースを継続したくない場合は、`ctxlogd` デーモンを停止できます。

setlog ユーティリティ

トレースの有効化は、**/opt/Citrix/VDA/bin/** パスにある `setlog` ユーティリティを使用して構成されます。rootユーザーのみが実行する権限を持っています。GUIを使用するか、コマンドを実行して構成を表示および変更できます。`setlog` ユーティリティに関するヘルプについては、次のコマンドを実行してください。

```
setlog help
```

値

デフォルトでは、**ログ出力パス**は **/var/log/xdl/hdx.log** に設定され、**最大ログサイズ**は200 MBに設定されており、**ログ出力パス**の下に最大2つの古いログファイルを保存できます。

現在の `setlog` の値を表示します。

```
setlog values


log_path (Log Output Path) = /var/log/xdl/hdx.log


log_size (Max Log Size (MiB)) = 200


log_count (Max Old Log Files) = 2
```

単一の `setlog` の値を表示または設定します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

デフォルトでは、ログレベルは **警告** に設定されています。

さまざまなコンポーネントに設定されているログレベルを表示します。

```
setlog levels
```

次のコマンドを使用して、すべてのログレベル（無効、継承、詳細、情報、警告、エラー、致命的なエラーを含む）を設定できます。

```
setlog level <class> [<level>]
```

`<class>` 変数はLinux VDAの1つのコンポーネントを指定します。すべてのコンポーネントを対象とするには、それを「all」に設定します。

```
setlog level all error



Setting log class ALL to ERROR.
```

フラグ

デフォルトでは、フラグは次のように設定されています。

```
setlog flags



DATE = true
```

```
TIME = true

NAME = true

PID = true

TID = false

SID = true

UID = false

GID = false

CLASS = false

LEVEL = false

FUNC = true

FILE = false
```

現在のフラグを表示します。

```
setlog flags
```

単一のログフラグを表示または設定します。

```
setlog flag <flag> [<state>]
```

デフォルトに戻す

すべてのレベル、フラグ、および値をデフォルト設定に戻します。

```
setlog default
```

重要:

`ctxlogd` サービスは、rootユーザーのみが作成できる **/var/xdl/.ctxlog** ファイルを使用して構成されます。他のユーザーはこのファイルへの書き込み権限を持っていません。rootユーザーが他のユーザーに書き込み権限を与えないことをお勧めします。これに従わない場合、任意または悪意のある構成が `ctxlogd` になる可能性があり、サーバーのパフォーマンス、ひいてはユーザーエクスペリエンスに影響を与える可能性があります。

トラブルシューティング

/var/xdl/.ctxlog ファイルが見つからない場合（誤って削除された場合など）、`ctxlogd` デーモンが失敗し、`ctxlogd` サービスを再起動できません。

/var/log/messages:

```
Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
configuration file.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited,
code=exited, status=1/FAILURE
```

```
Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
```

この問題を解決するには、rootユーザーとして `setlog` を実行して **/var/xdl/.ctxlog** ファイルを再作成します。その後、他のサービスが依存する `ctxlogd` サービスを再起動します。

セッションのシャドウイング

セッションシャドウイング機能により、ドメイン管理者はイントラネットでユーザーのICAセッションを表示できます。この機能はnoVNCを使用してICAセッションに接続し、RHEL 7.xおよびUbuntu 16.04でのみサポートされています。

注:

セッションシャドウイング機能を使用するには、Citrix Directorのバージョンが7.16以降である必要があります。

インストールと構成

依存関係

セッションシャドウイングには、`python-websockify` と `x11vnc` の2つの新しい依存関係が必要です。`python-websockify` と `x11vnc` の依存関係は、Ubuntu 16.04にLinux VDAをインストールすると自動的にインストールされます。RHEL 7.xでは、Linux VDAのインストール後に `python-websockify` と `x11vnc` を手動でインストールする必要があります。

RHEL 7.xで次のコマンドを実行して、`python-websockify` と `x11vnc` (`x11vnc` バージョン0.9.13以降) をインストールします。

```
sudo yum install -y python-websockify x11vnc
```

`python-websockify` と `x11vnc` を解決するには、RHEL 7.xで次のリポジトリを有効にします。

- Enterprise Linux 用追加パッケージ (EPEL)

EPELリポジトリは、`python-websockify` と `x11vnc` の両方に必要です。EPELリポジトリを有効にするには、次のコマンドを実行します。

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(rpm -E '%{rhel}').noarch.rpm
```

- オプションのRPM

  `x11vnc` の一部の依存パッケージをインストールするためのオプションのRPMリポジトリを有効にするには、次のいずれかのコマンドを実行します。

  ワークステーションの場合:

  ```
  subscription-manager repos --enable=rhel-7-workstation-optional-rpms
  ```

  サーバーの場合:

  ```
  subscription-manager repos --enable=rhel-7-server-optional-rpms
  ```

ポート

セッションシャドウイング機能は、Linux VDAからCitrix Directorへの接続を確立するために、6001～6099の範囲内で利用可能なポートを自動的に選択します。したがって、同時にシャドウできるICA®セッションの数は99に制限されます。特にマルチセッションシャドウイングの場合、要件を満たすのに十分なポートが利用可能であることを確認してください。

レジストリ

次の表に、関連するレジストリを示します。

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションシャドウイング機能を有効または無効にします	1 (有効)
ShadowingUseSSL	Linux VDAとCitrix Director間の接続を暗号化するかどうかを決定します	0 (無効)

レジストリ値を変更するには、Linux VDAで `ctxreg` コマンドを実行します。たとえば、セッションシャドウイングを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d
0x00000000
```

SSL

Linux VDAとCitrix Director間のnoVNC接続はWebSocketプロトコルを使用します。セッションシャドウイングの場合、`ws://`と`wss://`のどちらが選択されるかは、前述の「ShadowingUseSSL」レジストリによって決定されます。デフォルトでは、`ws://`が選択されます。ただし、セキュリティ上の理由から、`wss://`を使用し、各Citrix Directorクライアントと各Linux VDAサーバーに証明書をインストールすることをお勧めします。Citrixは、`ws://`を使用するLinux VDAセッションシャドウイングに関するいかなるセキュリティ責任も負いません。

サーバーおよびルートSSL証明書を取得する

証明書は、信頼された認証局（CA）によって署名されている必要があります。

SSLを構成する各Linux VDAサーバーには、個別のサーバー証明書（キーを含む）が必要です。サーバー証明書は特定のコンピューターを識別するため、各サーバーの完全修飾ドメイン名（FQDN）を知っている必要があります。便宜上、ドメイン全体にワイルドカード証明書を使用することもできます。この場合、少なくともドメイン名を知っている必要があります。

各サーバーにサーバー証明書をインストールするだけでなく、Linux VDAサーバーと通信する各Citrix Directorクライアントに、同じCAからのルート証明書をインストールする必要があります。ルート証明書は、サーバー証明書を発行するのと同じCAから入手できます。サーバー証明書とクライアント証明書は、オペレーティングシステムにバンドルされているCA、エンタープライズCA（組織がアクセスを許可しているCA）、またはオペレーティングシステムにバンドルされていないCAからインストールできます。証明書を取得するためにどの方法が必要かについては、組織のセキュリティチームに相談してください。

重要:

- サーバー証明書の共通名は、Linux VDAサーバーの正確なFQDNであるか、少なくとも正しいワイルドカードとドメイン文字である必要があります。例：vda1.basedomain.com または *.basedomain.com。
- SHA1やMD5を含むハッシュアルゴリズムは、一部のブラウザがサポートするデジタル証明書の署名には脆弱すぎます。そのため、SHA-256が最小標準として指定されています。

各Citrix Directorクライアントにルート証明書をインストールする

セッションシャドウイングは、IISと同じレジストリベースの証明書ストアを使用するため、IISまたはMicrosoft管理コンソール (MMC) の証明書スナップインを使用してルート証明書をインストールできます。CAから証明書を受け取ったら、IISでWebサーバー証明書ウィザードを再起動すると、ウィザードが証明書をインストールします。または、MMCを使用してコンピューター上の証明書を表示およびインポートし、スタンドアロンスナップインとして証明書を追加することもできます。Internet ExplorerとGoogle Chromeは、デフォルトでオペレーティングシステムにインストールされている証明書をインポートします。Mozilla Firefoxの場合、証明書マネージャーの**認証局**タブでルートSSL証明書をインポートする必要があります。

各Linux VDAサーバーにサーバー証明書とそのキーをインストールする

サーバー証明書を「shadowingcert.*」、キーファイルを「shadowingkey.*」と名付けます (*は形式を示すことができます。たとえば、shadowingcert.csrおよびshadowingkey.key)。サーバー証明書とキーファイルをパス **/etc/xdl/shadowingssl** の下に置き、制限された権限で適切に保護します。不適切な名前またはパスは、Linux VDAが特定の証明書またはキーファイルを見つけられない原因となり、そのためCitrix Directorとの接続障害を引き起こします。

使用方法

Citrix Directorから、ターゲットセッションを見つけ、**セッションの詳細**ビューで**シャドウ**をクリックして、Linux VDAにシャドウイング要求を送信します。

セッション詳細ビューのシャドウタブの画像(/en-us/linux-virtual-delivery-agent/2106/media/lvda-session-shadowing.png)

接続が初期化された後、ICAセッションクライアント（Citrix Directorクライアントではない）に確認が表示され、ユーザーにセッションをシャドウする許可を求めます。

管理者がこのセッションをシャドウすることを許可するかどうかの画像(/en-us/linux-virtual-delivery-agent/2106/media/allow-administrator-to-shadow-session.png)

ユーザーが**はい**をクリックすると、Citrix Director側にウィンドウが表示され、ICAセッションがシャドウされていることを示します。

使用方法の詳細については、Citrix Directorドキュメントを参照してください。

制限事項

- セッションシャドウイングは、イントラネット内でのみ使用するように設計されています。Citrix Gateway経由で接続しても、外部ネットワークでは機能しません。Citrixは、外部ネットワークでのLinux VDAセッションシャドウイングに対する一切の責任を負いません。

- セッションシャドウイングが有効になっている場合、ドメイン管理者はICAセッションを表示することしかできませんが、書き込みや制御の権限はありません。
- 管理者がCitrix Directorから**シャドウ**をクリックすると、ユーザーにセッションをシャドウする許可を求める確認が表示されます。セッションは、セッションユーザーが許可を与えた場合にのみシャドウできます。
- 前述の確認には20秒のタイムアウト制限があります。時間が経過すると、シャドウイング要求は失敗します。
- 1つのICAセッションは、1つのCitrix Directorウィンドウで1人の管理者のみがシャドウできます。ICAセッションが管理者Aによってシャドウされており、その間に管理者Bがシャドウ要求を送信した場合、ユーザーの許可を得るための確認がユーザーデバイスに再表示されます。ユーザーが同意すると、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が確立されます。同じICAセッションに対して同じ管理者から別のシャドウ要求が送信された場合も同様です。
- セッションシャドウイングを使用するには、Citrix Director 7.16以降をインストールしてください。
- Citrix Directorクライアントは、ターゲットのLinux VDAサーバーに接続するために、IPアドレスではなくFQDNを使用します。したがって、Citrix DirectorクライアントはLinux VDAサーバーのFQDNを解決できる必要があります。

トラブルシューティング

セッションシャドウイングが失敗した場合は、Citrix DirectorクライアントとLinux VDAの両方でデバッグを実行してください。

Citrix Directorクライアントで

ブラウザの開発者ツールを使用して、**Console**タブで出力ログを確認します。または、**Network**タブでShadowLinuxSession APIの応答を確認します。ユーザーの許可を得るための確認が表示されるものの、接続が確立できない場合は、Linux VDAのFQDNを手動でpingして、Citrix DirectorがFQDNを解決できることを確認します。`wss://`接続に問題がある場合は、証明書を確認してください。

Linux VDAでの

シャドウ要求に応じて、ユーザーの許可を得るための確認が表示されることを確認します。表示されない場合は、vda.logファイルとhdx.logファイルで手がかりを探します。vda.logファイルを取得するには、次の手順を実行します。

1. /etc/xdl/ctx-vda.confファイルを見つけます。vda.log構成を有効にするには、次の行のコメントを解除します。

   `Log4jConfig=”/etc/xdl/log4j.xml”`

2. /etc/xdl/log4j.xmlを開き、com.citrix.dmcの部分を見つけて、「info」を「trace」に次のように変更します。

   ```
   <!-- Broker Agent Plugin - Director VDA plugin Logger -->


    <logger name="com.citrix.dmc">


      <level value="trace"/>


    </logger>
   ```

3. `ctxvda`サービスを再起動するには、`service ctxvda restart`コマンドを実行します。

接続確立中にエラーが発生した場合：

1. セッションシャドウイングがポートを開くのを妨げるファイアウォールの制限がないか確認します。
2. SSLシナリオの場合、証明書とキーファイルが適切に命名され、正しいパスに配置されていることを確認します。
3. 新しいシャドウ要求のために、6001～6099の間に十分なポートが残っていることを確認します。

ブラウザコンテンツリダイレクト

概要

Linux VDA は Google Chrome でブラウザコンテンツリダイレクトをサポートしています。ブラウザコンテンツリダイレクトは、許可リスト内のウェブページをクライアント側でレンダリングする機能を提供します。この機能は、Citrix Workspace™ アプリを使用してクライアント側に適切な

レンダリングエンジンをインスタンス化し、URL から HTTP および HTTPS コンテンツを取得します。

> **注:**
>
> 許可リストを使用すると、どのウェブページをクライアント側にリダイレクトするかを指定できます。逆に、ブロックリストを使用すると、どのウェブページをクライアント側にリダイレクトしないかを指定できます。

> このオーバーレイウェブレイアウトエンジンは、VDA上ではなくクライアント上で実行され、クライアントのCPU、GPU、RAM、およびネットワークを使用します。

ブラウザのビューポートのみがリダイレクトされます。ビューポートとは、ブラウザ内でコンテンツが表示される長方形の領域です。ビューポートには、アドレスバー、お気に入りバー、ステータスバーなどの項目は含まれません。これらの項目は、VDA上のブラウザで引き続き実行されます。

ビューポートの画像(/en-us/linux-virtual-delivery-agent/2106/media/viewport.png)

システム要件

Windows クライアント:

- Citrix Workspace アプリ 1809 Windows版 以降

Linux VDA の場合:

- VDA オペレーティングシステム: Ubuntu 16.04, Ubuntu 18.04, RHEL 7.8, RHEL 8.2, RHEL 8.1, SLES 12.5
- VDA上のブラウザ: Google Chrome v66 以降 (Citrix® browser content redirection 拡張機能が追加されていること)

ブラウザコンテンツリダイレクトの構成

1. Citrix Studioで、ブラウザコンテンツリダイレクトを使用できるURLの許可リストと、使用できないURLのブロックリストを指定するポリシーを構成します。ブラウザコンテンツリダイレクトは、デフォルトで**許可**に設定されています。

   ブラウザコンテンツリダイレクトが許可に設定されている画像(/en-us/linux-virtual-delivery-agent/2106/media/bcr-allowed-by-default.png)

Browser Content Redirection ACL Configuration 設定は、ブラウザコンテンツリダイレクトを使用できるURLの許可リストを指定します。

Browser Content Redirection ACL Configuration が選択されている画像(/en-us/linux-virtual-delivery-agent/2106/media/allow-list.png)

許可リストの設定画像(/en-us/linux-virtual-delivery-agent/2106/media/allow-list-confiuration.png)

ブラウザコンテンツリダイレクトブラックリスト構成設定は、ブラウザコンテンツリダイレクトを使用できないURLのブロックリストを指定します。

ブロックリストの設定画像(/en-us/linux-virtual-delivery-agent/2106/media/block-list.png)

> **注:**
>
> Linux VDAは現在、**ブラウザコンテンツリダイレクトプロキシ構成**設定をサポートしていません。

2. VDA上のブラウザが、URL（ナビゲートされている）が許可リストまたはブロックリストと一致するかどうかを検出するには、ChromeウェブストアからCitrixブラウザコンテンツリダイレクト拡張機能を追加します。VDA上で**Chromeに追加**をクリックします。

> **重要:**
>
> クライアントには拡張機能は不要です。VDAにのみ追加してください。

\>

> Chrome拡張機能はユーザーごとにインストールされます。拡張機能を追加または削除するためにゴールデンイメージを更新する必要はありません。

> ChromeウェブストアからCitrixブラウザコンテンツリダイレクト拡張機能を追加する画像(/en-us/linux-virtual-delivery-agent/2106/media/add-to-chrome.png)

URLが許可リスト（例: `https://www.mycompany.com/`）で見つかり、ブロックリストには見つからない場合、仮想チャネル（CTXCSB）がCitrix Workspaceアプリにリダイレクトが必要であることを指示し、URLを中継します。その後、Citrix Workspaceアプリはローカルレンダリングエンジンをインスタンス化し、ウェブサイトを表示します。

Citrix Workspaceアプリは、ウェブサイトを仮想デスクトップのブラウザコンテンツ領域にシームレスに統合します。

ビューポートとクライアントの画像(/en-us/linux-virtual-delivery-agent/2106/media/viewport-desktop-viewer.png)

1. Citrixブラウザコンテンツリダイレクト拡張機能のアイコン

   拡張機能アイコンの色は、Chrome拡張機能のステータスを示します。以下の3色のいずれかです。

   - 緑: アクティブで接続済み
   - 灰色: 現在のタブで非アクティブ/アイドル状態
   - 赤: 破損/動作していません

2. ビューポートはクライアントでレンダリングされるか、仮想デスクトップにブレンドされます
3. リナックス ブイディーエー
4. Windowsクライアント

リダイレクトのシナリオ

Citrix Workspaceアプリがコンテンツを取得するシナリオを次に示します。

リダイレクトのシナリオの画像(/en-us/linux-virtual-delivery-agent/2106/media/browser-viewport-redirection-scenarios.png)

- **サーバーフェッチとサーバーレンダリング:** サイトを許可リストに追加しなかったか、リダイレクトが失敗したため、リダイレクトは行われません。VDAでWebページをレンダリングし、Thinwireを使用してグラフィックをリモート処理するようにフォールバックします。フォールバック動作を制御するには、ポリシーを使用します。このシナリオでは、VDAでCPU、RAM、および帯域幅の消費量が高くなります。
- **クライアントフェッチとクライアントレンダリング:** Citrix WorkspaceアプリがWebサーバーに直接接続するため、インターネットアクセスが必要です。このシナリオでは、Citrix Virtual Apps and Desktops™サイトからすべてのネットワーク、CPU、およびRAMの使用量をオフロードします。

フォールバックメカニズム

クライアントのリダイレクトが失敗する場合があります。たとえば、クライアントマシンに直接インターネットアクセスがない場合、エラー応答がVDAに戻される可能性があります。そのような場合、VDA上のブラウザはページを再読み込みし、サーバー上でレンダリングできます。

Citrix Workspace™アプリ for HTML5のサポート

このリリース以降、Citrix Workspaceアプリ for HTML5を使用して、クライアントをCitrix Gatewayに接続することなく、Linux仮想アプリおよびデスクトップに直接アクセスできます。Citrix Workspaceアプリ for HTML5について詳しくは、Citrixドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトで無効になっています。有効にするには、次の手順を実行します。

1. Citrix StoreFront™で、Citrix Workspaceアプリ for HTML5を有効にします。

   詳細な手順については、Knowledge Centerの記事CTX208163のステップ1を参照してください。

2. WebSocket接続を有効にします。

   a. Citrix Studioで、**WebSockets connections**ポリシーを**Allowed**に設定します。

      その他のWebSocketポリシーも設定できます。WebSocketポリシーの完全なリストについては、「WebSocketsポリシー設定」を参照してください。

   b. VDAで、設定を有効にするために、`ctxvda`サービスと`ctxhdx`サービスをこの順序で再起動します。

   c. VDAで、次のコマンドを実行してWebSocketリスナーが実行されているかどうかを確認します。

      ```
      netstat -an | grep 8008
      ```

      WebSocketリスナーが実行されている場合、コマンド出力は次のようになります。

      ```
      tcp 0 0 :::8008 :::* LISTEN
      ```

注： WebSocket接続を保護するためにTLS暗号化を有効にすることもできます。TLS暗号化の有効化について詳しくは、「TLSを使用したユーザーセッションの保護」を参照してください。

Citrix Director で Linux VM と Linux セッションを監視する

この記事では、Citrix Director で Linux VM および Linux セッションに利用できるメトリックの一部を一覧表示します。

Linux VM のメトリック

Linux VM のメトリックにアクセスするには、Citrix Director でその VM を見つけ、**Machine Details** パネルを確認します。

Linux VDA バージョン 2103 以降、Citrix Director で Linux VM の以下のメトリックが利用可能です。

- CPU コア数
- メモリサイズ
- ハードディスク容量
- 現在および過去の CPU とメモリの使用率

  Linux VM メトリック - 1(/en-us/linux-virtual-delivery-agent/2106/media/linux-vm-metrics-1.png)

  Linux VM メトリック - 2(/en-us/linux-virtual-delivery-agent/2106/media/linux-vm-metrics-2.png)

Linux セッションのメトリック

Linux セッションのメトリックを表示するには、**Filters > Sessions > All Sessions** を選択して **All Sessions** ページを開くか、**Session Details** パネルにアクセスします。**Session Details** パネルにアクセスするには、**All Sessions** ページを開き、ターゲットセッションをクリックしてその **Activity Manager** ビューにアクセスします。例：

Session Details パネル(/en-us/linux-virtual-delivery-agent/2106/media/session-details-panel.png)

- ICA® RTT

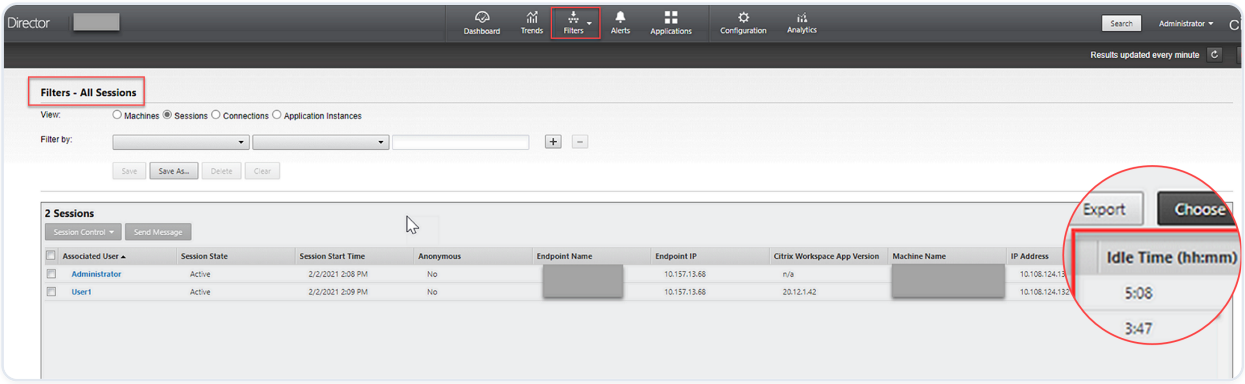
  Linux VDAバージョン1903以降、ICA RTTメトリックが利用可能です。ICA RTTメトリックを表示するには、Citrix Director 1903以降を使用し、Citrix Studioで**ICAラウンドトリップ計算**および**ICAラウンドトリップ計算間隔**ポリシーを作成します。ポリシーの作成については、「Studioを使用したポリシーの作成」を参照してください。

- プロトコル

  Linux VDAバージョン1909以降、プロトコル情報が利用可能です。Linuxセッションのトランスポートプロトコルは、**セッションの詳細**パネルに**UDP**または**TCP**として表示されます。

- アイドル時間

  Linux VDAバージョン2103以降、Linuxセッションのアイドル時間メトリックが利用可能です。このメトリックにアクセスするには、**フィルター > セッション > すべてのセッション**を選択して、**すべてのセッション**ページを開きます。



監視サービスデーモン

監視サービスデーモンは、定期的なスキャンを実行して主要なサービスを監視します。例外を検出すると、デーモンはサービスプロセスを再起動または停止し、リソースを解放するためにプロセスの残存物をクリーンアップします。検出された例外は、**/var/log/xdl/ms.log** ファイルに記録されます。

構成

VDAを起動すると、監視サービスデーモンは自動的に起動します。

管理者権限を持つ**scanningpolicy.conf**、**rulesets.conf**、および**whitelist.conf**ファイルを通じて、この機能を構成できます。構成ファイルは**/opt/Citrix/VDA/sbin**にあります。

scanningpolicy.conf、**rulesets.conf**、および**whitelist.conf**ファイルへの変更を有効にするには、次のコマンドを実行して監視サービスデーモンを再起動します。

```
service ctxmonitorservice restart
```

- **scanningpolicy.conf**

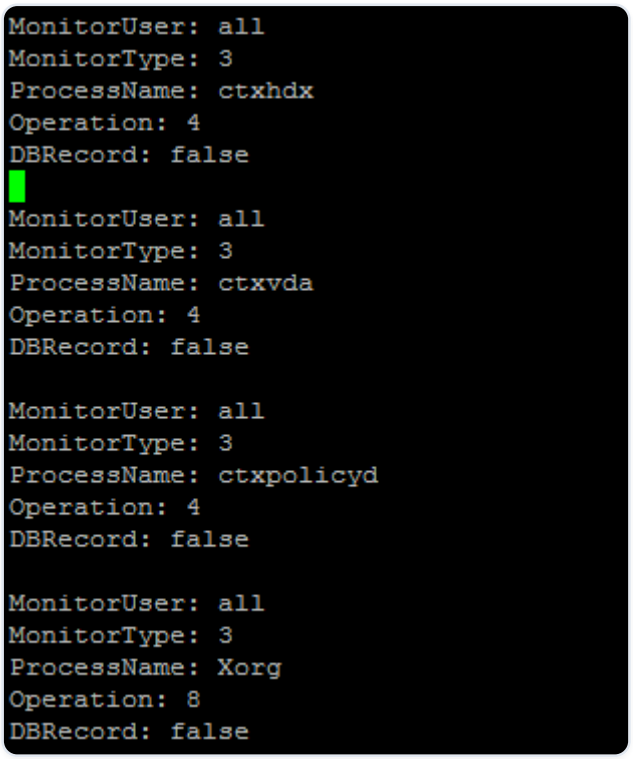
  この構成ファイルは、監視サービスデーモンを有効または無効にします。サービス検出間隔を設定し、検出された例外を修復するかどうかを指定します。

  - MonitorEnable: true/false (デフォルトはtrue)
  - DetectTime: 20 (単位: 秒、デフォルト値: 20、最小値: 5)

- AutoRepair: true/false (デフォルトはtrue)
- MultBalance: 無効
- ReportAlarm: 無効

- **rulesets.conf**

  この構成ファイルは、監視対象サービスを指定します。以下のスクリーンショットに示すように、デフォルトで4つの監視対象サービスがあります。



  監視する各サービスを設定するには、以下のフィールドを設定します。

  - MonitorUser: すべて
  - MonitorType: 3
  - ProcessName: <> (プロセス名は空白にできず、完全に一致する必要があります。)
  - Operation: 1/2/4/8 (1 = 例外が検出されたときにサービスを停止します。2 = 例外が検出されたときにサービスを強制終了します。4 = サービスを再起動します。8 = Xorgプロセスの残存物をクリーンアップします。)
  - DBRecord: 無効

- **whitelist.conf**

  rulesets.confファイルで指定されたターゲットサービスは、**whitelist.conf**ファイルでも構成する必要があります。ホワイトリスト構成は、セキュリティのための二次フィルターです。

ホワイトリストを設定するには、**whitelist.conf**ファイルにプロセス名（完全に一致する必要があります）のみを含めます。例については、以下のスクリーンショットを参照してください。

ホワイトリストの画像(/en-us/linux-virtual-delivery-agent/2106/media/a-white-list.png)

注:

`ctxvda`、`ctxhdx`、および`ctxpolicyd`サービスを停止する前に、`service ctxmonitorservice stop`コマンドを実行してモニターサービスデーモンを停止します。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

TLS を使用したセキュアなユーザーセッション

バージョン 7.16 以降、Linux VDA はセキュアなユーザーセッション向けに TLS 暗号化をサポートしています。TLS 暗号化はデフォルトで無効になっています。

TLS 暗号化を有効にする

セキュアなユーザーセッション向けに TLS 暗号化を有効にするには、証明書を取得し、Linux VDA と Delivery Controller™ (Controller) の両方で TLS 暗号化を有効にします。

証明書を取得する

信頼された認証局 (CA) から、PEM 形式のサーバー証明書と CRT 形式のルート証明書を取得します。サーバー証明書には、次のセクションが含まれています。

- 証明書
- 暗号化されていない秘密キー
- 中間証明書 (オプション)

サーバー証明書の例：

サーバー証明書の例 (/en-us/linux-virtual-delivery-agent/2106/media/example-server-certificate.png)

TLS 暗号化を有効にする

Linux VDA で TLS 暗号化を有効にする

Linux VDA では、`enable_vdassl.sh` ツールを使用して TLS 暗号化を有効 (または無効) にします。このツールは **/opt/Citrix/VDA/sbin** ディレクトリにあります。このツールで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -help` コマンドを実行してください。

ヒント: サーバー証明書は各 Linux VDA サーバーにインストールする必要があり、ルート証明書は各 Linux VDA サーバーとクライアントにインストールする必要があります。

Controller で TLS 暗号化を有効にする

> **注:**
>
> TLS暗号化はデリバリーグループ全体に対してのみ有効にできます。特定のアプリケーションに対しては有効にできません。

Controller上のPowerShellウィンドウで、ターゲットのデリバリーグループに対してTLS暗号化を有効にするには、以下のコマンドを順に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

> **注:**
>
> ICA®セッションファイルにVDA FQDNのみが含まれるようにするには、`Set-BrokerSite –DnsResolutionEnabled $true` コマンドも実行できます。このコマンドはDNS解決を有効にします。DNS解決を無効にすると、ICAセッションファイルはVDAのIPアドレスを開示し、SSLProxyHostやUDPDTLSPortなどのTLS関連項目に対してのみFQDNを提供します。

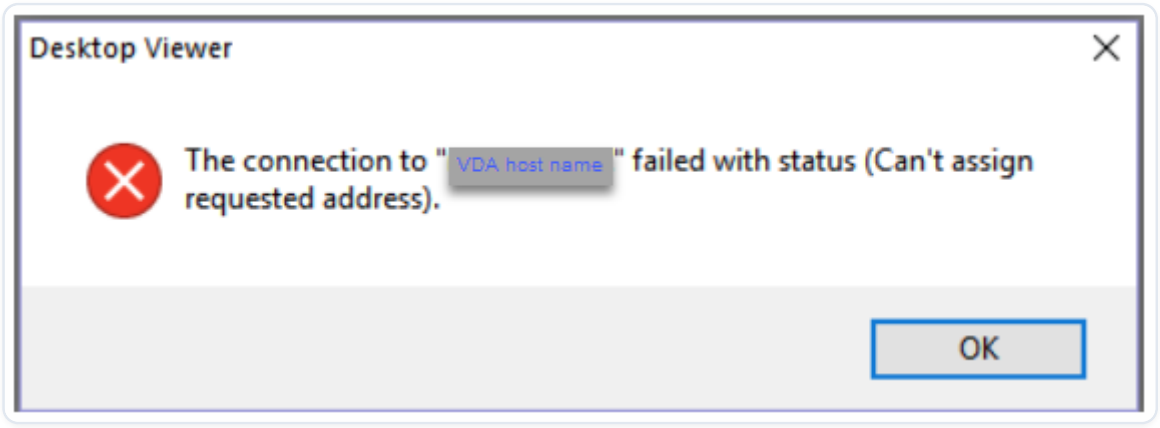
ControllerでTLS暗号化を無効にするには、以下のコマンドを順に実行します。

1. `Add-PSSnapin citrix.*`.

2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

トラブルシューティング

公開されたデスクトップセッションにアクセスしようとすると、Windows向けCitrix Workspace™アプリで以下の「要求されたアドレスを割り当てできません」エラーが発生する場合があります。



回避策として、**hosts**ファイルに以下のようなエントリを追加します。

```
<IP address of the Linux VDA>      <FQDN of the Linux VDA>
```

Windowsマシンでは、**hosts**ファイルは通常`C:\Windows\System32\drivers\etc\hosts`にあります。

DTLS を使用したユーザーセッションの保護

DTLS 暗号化は、7.18 リリース以降で完全にサポートされている機能です。デフォルトでは、この機能は Linux VDA で有効になっています。詳しくは、「トランスポート層セキュリティ」を参照してください。

DTLS 暗号化を有効にする

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、**HDX™ アダプティブトランスポート**ポリシーが**優先**または**診断モード**に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、`enable_vdassl.sh` ツールを使用して SSL 暗号化を有効（または無効）にします。このツールは **/opt/Citrix/VDA/sbin** にあります。このツールで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -h` コマンドを実行してください。

> **注:**
>
> 現在、Linux VDA は DTLS 1.0 と DTLS 1.2 の両方をサポートしています。DTLS 1.2 には、Citrix Receiver for Windows 4.12、または Citrix Workspace app 1808 for Windows 以降が必要です。クライアントが DTLS 1.0 のみをサポートしている場合（例：Citrix Receiver for Windows 4.11）、`enable_vdassl.sh` ツールを使用して、`SSLMinVersion` を `TLS_1.0` に、`SSLCipherSuite` を `COM` または `ALL` に設定します。

テキストベースのセッションウォーターマーク

テキストベースのセッションウォーターマークは、データ盗難を抑止し、追跡を可能にするのに役立ちます。この追跡可能な情報は、写真やスクリーンキャプチャを使用してデータを盗む人々への抑止力として、セッションデスクトップに表示されます。元のドキュメントのコンテンツを変更することなく、セッション画面全体に表示されるテキストのレイヤーであるウォーターマークを指定できます。

> **重要:**
>
> テキストベースのセッションウォーターマークは、セキュリティ機能ではありません。このソリューションはデータ盗難を完全に防ぐものではありませんが、ある程度の抑止力と追跡可能性を提供します。この機能を使用した場合の完全な情報追跡可能性は保証しません。ただし、必要に応じてこの機能を他のセキュリティソリューションと組み合わせて使用することをお勧めします。

> セッションウォーターマークはテキストであり、ユーザーに配信されるセッションに適用されます。セッションウォーターマークには、データ盗難を追跡するための情報が含まれています。最も重要なデータは、スクリーンイメージが取得された現在のセッションのログオンユーザーのIDです。データ漏洩をより効果的に追跡するために、サーバーまたはクライアントのインターネットプロトコルアドレスや接続時間などの他の情報を含めます。

ユーザーエクスペリエンスを調整するには、セッションウォーターマークポリシー設定 を使用して、画面上の配置とウォーターマークの外観を構成します。

制限事項

- ブラウザコンテンツリダイレクトが使用されているセッションでは、セッションウォーターマークはサポートされていません。セッションウォーターマーク機能を使用するには、ブラウザコンテンツリダイレクトが無効になっていることを確認してください。
- セッションがレガシーNVIDIAドライバーでフルスクリーンハードウェアアクセラレーションH.264またはH.265エンコードモードで実行されている場合（この場合、レジストリでNvCaptureTypeが2に設定されている場合）、セッションウォーターマークはサポートされず、表示されません。
- セッションシャドウイングではウォーターマークは表示されません。
- Print Screenキーを押して画面をキャプチャした場合、VDA側でキャプチャされた画面にはウォーターマークは含まれません。画面キャプチャがコピーされないように対策を講じることをお勧めします。

スマートカードを使用したパススルー認証

ユーザーは、Linux仮想デスクトップセッションにログオンする際に、クライアントデバイスに接続されたスマートカードを認証に使用できます。この機能は、ICA®スマートカード仮想チャネルを介したスマートカードリダイレクトによって実装されます。ユーザーはセッション内でスマートカードを使用することもできます。使用例としては、ドキュメントへのデジタル署名の追加、メールの暗号化または復号化、スマートカード認証を必要とするWebサイトへの認証などがあります。

Linux VDAは、この機能に関してWindows VDAと同じ構成を使用します。詳細については、この記事の「スマートカード環境の構成」セクションを参照してください。

> **注:**
>
> Linux VDAセッション内でマップされたスマートカードを使用してCitrix Gatewayにサインオンすることは、公式にはサポートされていません。

前提条件

スマートカードパススルー認証の可用性は、以下の条件に依存します。

- お使いのLinux VDAは、以下のいずれかのディストリビューションにインストールされています。
  - RHEL 8/CentOS 8
  - RHEL 7/CentOS 7
  - ウブントゥ 20.04
  - ウブントゥ 18.04
  - ウブントゥ 16.04

  VDAのインストールが完了したら、VDAがDelivery Controller™に登録でき、公開されたLinuxデスクトップセッションがWindows資格情報を使用して正常に起動できることを確認します。
- OpenSCでサポートされているスマートカードが使用されます。詳細については、「OpenSCがスマートカードをサポートしていることを確認する」を参照してください。
- Citrix Workspace™ app for Windowsが使用されます。

OpenSCがスマートカードをサポートしていることを確認する

OpenSC は RHEL 7.4 以降で広く使用されているスマートカードドライバーです。CoolKey の完全互換の代替として、OpenSC は多くの種類のスマートカードをサポートしています (Red Hat Enterprise Linux でのスマートカードのサポート を参照)。

この記事では、設定を説明するために YubiKey 4 スマートカードを例として使用します。YubiKey 4 は、Amazon やその他の小売業者から簡単に購入できるオールインワンの USB CCID PIV デバイスです。OpenSC ドライバーは YubiKey 4 をサポートしています。



組織でより高度なスマートカードが必要な場合は、RHEL 7 または RHEL 8 と OpenSC パッケージがインストールされた物理マシンを準備します。OpenSC のインストールについては、「スマートカードドライバーのインストール」(/ja-jp/linux-virtual-delivery-agent/2106/configuration/

pass-through-authentication-with-smart-cards-.html#install-the-smart-card-driver) を参照してください。スマートカードを挿入し、次のコマンドを実行して、OpenSC がスマートカードをサポートしていることを確認します。

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. 通常、CA サーバーから PEM 形式のルート証明書を取得します。

   次のコマンドと同様のコマンドを実行して、DER ファイル (*.crt, *.cer, *.der) を PEM に変換できます。次のコマンド例では、**certnew.cer** は DER ファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   ```

2. ルート証明書を `openssl` ディレクトリにインストールします。**certnew.pem** ファイルは例として使用されます。

   ```
   cp certnew.pem <path where you install the root certificate>
   ```

   ルート証明書をインストールするためのパスを作成するには、`sudo mdkir -p <path where you install the root certificate>` を実行します。

RHEL 8/CentOS 8 で pam_krb5 モジュールをビルドする

スマートカード認証は pam_krb5 モジュールに依存しますが、これは RHEL 8/CentOS 8 では非推奨です。RHEL 8/CentOS 8 でスマートカード認証を使用するには、次のように pam_krb5 モジュールをビルドします。

1. `<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>` から pam_krb5-2.4.8-6 ソースコードをダウンロードします。
2. RHEL 8/CentOS 8 で pam_krb5 モジュールをビルドしてインストールします。

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

yum install gcc krb5-devel pam-devel autoconf libtool

rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio –div

tar xvzf pam_krb5-2.4.8.tar.gz

cd pam_krb5-2.4.8

./configure --prefix=/usr

make

make install
```

3. /usr/lib64/security/ の下に pam_krb5.so が存在することを確認します。

```
ls -l /usr/lib64/security | grep pam_krb5
```

スマートカード環境を構成する

ctxsmartlogon.shスクリプトを使用してスマートカード環境を構成するか、手動で構成を完了できます。

(オプション1) ctxsmartlogon.shスクリプトを使用してスマートカード環境を構成する

> **注:**

> ctxsmartlogon.shスクリプトは、デフォルトレルムにPKINIT情報を追加します。この設定は、**/etc/krb5.conf**構成ファイルで変更できます。

> スマートカードを初めて使用する前に、ctxsmartlogon.shスクリプトを実行してスマートカード環境を構成します。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行して有効にする(/en-us/linux-virtual-delivery-agent/2106/media/smart-card-logon-script1.png)

スマートカードを無効にするには：

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行して無効にする(/en-us/linux-virtual-delivery-agent/2106/media/smart-card-logon-script2.png)

(オプション2) スマートカード環境を手動で構成する

Linux VDAは、Windows VDAと同じスマートカード環境を使用します。この環境では、ドメインコントローラー、Microsoft証明機関（CA）、インターネットインフォメーションサービス、Citrix StoreFront、Citrix Workspaceアプリなど、複数のコンポーネントを構成する必要があります。YubiKey 4スマートカードに基づく構成については、Knowledge Centerの記事CTX206156を参照してください。

次の手順に進む前に、すべてのコンポーネントが正しく構成され、秘密キーとユーザー証明書がスマートカードにダウンロードされ、スマートカードを使用してWindows VDAに正常にログオンできることを確認してください。

PC/SC LITEパッケージをインストールする

PCSC Liteは、LinuxにおけるPersonal Computer/Smart Card (PC/SC)仕様の実装です。スマートカードおよびリーダーと通信するためのWindowsスマートカードインターフェイスを提供します。Linux VDAのスマートカードリダイレクトは、PC/SCレベルで実装されています。

PC/SC Lite パッケージをインストールするには、次のコマンドを実行します。

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

スマートカードドライバーをインストールする

OpenSC は RHEL で広く使用されているスマートカードドライバーです。OpenSC がインストールされていない場合は、次のコマンドを実行してインストールします。

```
yum install opensc
```

スマートカード認証用のPAMモジュールをインストールする

pam_krb5 および krb5-pkinit モジュールをインストールするには、次のコマンドを実行します。

RHEL 7/CentOS 7:

```
yum install pam_krb5 krb5-pkinit
```

RHEL 8/CentOS 8:

```
yum install krb5-pkinit
```

Ubuntu 20.04、Ubuntu 18.04、Ubuntu 16.04の各バージョンについて：

```
apt-get install libpam-krb5 krb5-pkinit
```

pam_krb5 モジュールは、PAM 対応アプリケーションがパスワードをチェックし、Key Distribution Center (KDC) からチケット許可チケットを取得するために使用できるプラガブル認証モジュールです。krb5-pkinit モジュールには、クライアントが秘密鍵と証明書を使用して KDC から初期資格情報を取得できるようにする PKINIT プラグインが含まれています。

PAM_KRB5 モジュールを構成する

pam_krb5 モジュールは、スマートカード内の証明書を使用して Kerberos チケットを取得するために KDC と対話します。PAM で pam_krb5 認証を有効にするには、次のコマンドを実行します。

```
authconfig --enablekrb5 --update
```

/etc/krb5.conf 設定ファイルに、実際のレルムに従って PKINIT 情報を追加します。

> **注:**
>
> **pkinit_cert_match** オプションは、PKINIT 認証を試行する前にクライアント証明書が一致する必要がある照合ルールを指定します。照合ルールの構文は次のとおりです。
>
> *コンポーネントルール ...*

ここで、`relation-operator` は `&&` (すべてのコンポーネントルールが一致する必要があることを意味します) または \|\| (1つのコンポーネントルールのみが一致する必要があることを意味します) のいずれかです。

一般的なkrb5.confファイルの例を次に示します。

```
EXAMPLE.COM = {


    kdc = KDC.EXAMPLE.COM


    auth_to_local = RULE:[1:$1@$0]


    pkinit_anchors = FILE:<path where you install the root certificate>/
certnew.pem


    pkinit_kdc_hostname = KDC.EXAMPLE.COM


    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature


    pkinit_eku_checking = kpServerAuth


 }
```

PKINIT情報を追加すると、構成ファイルは次のようになります。

PKINIT情報が追加されました(/en-us/linux-virtual-delivery-agent/2106/media/pkinit-information-new2.png)

PAM認証を構成する

PAM構成ファイルは、PAM認証に使用されるモジュールを示します。pam_krb5を認証モジュールとして追加するには、**/etc/pam.d/smartcard-auth** ファイルに次の行を追加します。

```
auth [success=done ignore=ignore default=die] pam_krb5.so
preauth_options=X509_user_identity=PKCS11:<path to the pkcs11 driver>/opensc-
pkcs11.so
```

SSSDを使用する場合、変更後の構成ファイルは次のようになります。

SSSDを使用する場合の変更された構成ファイル(/en-us/linux-virtual-delivery-agent/2106/media/smartcard-authentication-file.png)

(オプション) スマートカードを使用したシングルサインオン

シングルサインオン (SSO) は、仮想デスクトップおよびアプリケーションの起動でパススルー認証を実装するCitrixの機能です。この機能により、ユーザーがPINを入力する回数が減ります。Linux VDAでSSOを使用するには、Citrix Workspaceアプリを構成します。構成はWindows VDAと同じです。詳しくは、Knowledge Centerの記事CTX133982を参照してください。

Citrix Workspaceアプリでグループポリシーを構成する際に、スマートカード認証を次のように有効にします。

Workspaceアプリでスマートカード認証を有効にする(/en-us/linux-virtual-delivery-agent/2106/media/enable-smart-card-auth-in-receiver.jpg)

高速スマートカードログオン

高速スマートカードは、既存のHDX PC/SCベースのスマートカードリダイレクトを改善したものです。これにより、スマートカードが高遅延WAN環境で使用される場合のパフォーマンスが向上します。詳しくは、スマートカードを参照してください。

Linux VDAは、Citrix Workspaceアプリの次のバージョンで高速スマートカードをサポートしています。

- シトリックス レシーバー for Windows 4.12
- Windows向けCitrix Workspaceアプリ1808以降

クライアントで高速スマートカードログオンを有効にする

高速スマートカードログオンは、VDAではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連付けられているStoreFrontサイトのdefault.icaファイルに次のパラメーターを含めます。

```
[WFClient]

SmartCardCryptographicRedirection=On
```

クライアントで高速スマートカードログオンを無効にする

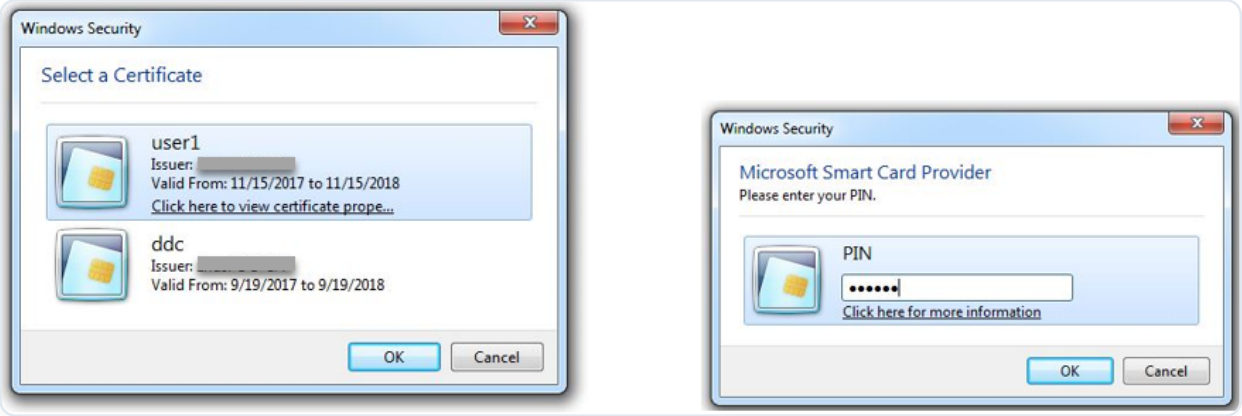
クライアントで高速スマートカードログオンを無効にするには、関連付けられているStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

使用方法

スマートカードを使用してLinux VDAにログオンする

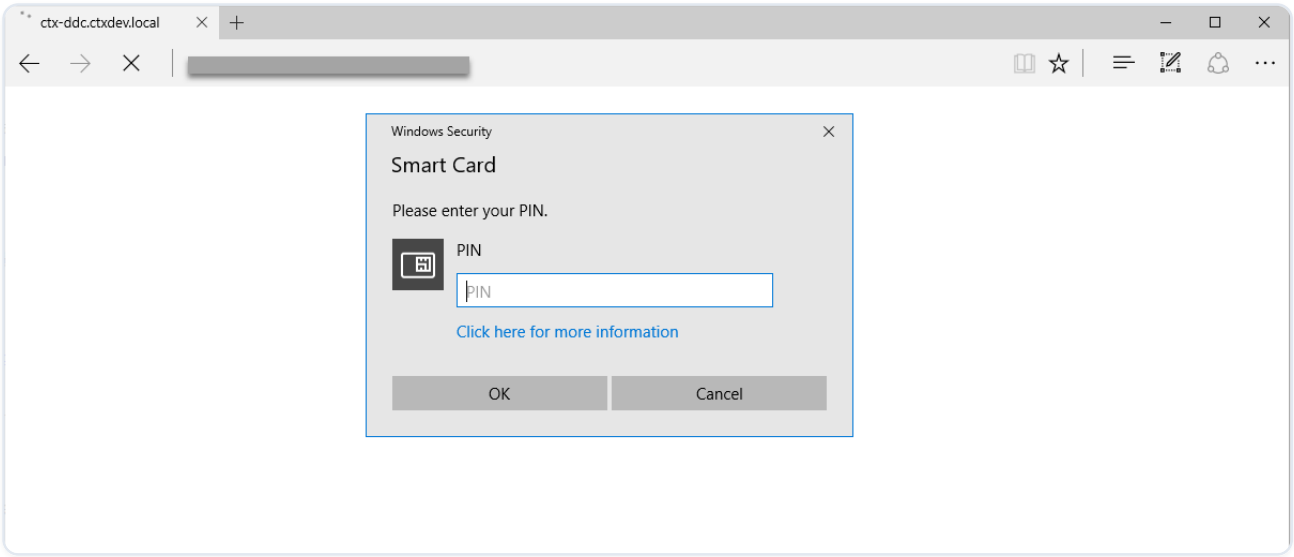
ユーザーは、SSOシナリオと非SSOシナリオの両方で、スマートカードを使用してLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用して、ユーザーはStoreFront™に自動的にログオンします。ユーザーがStoreFrontでLinux仮想デスクトップセッションを起動すると、PINがスマートカード認証のためにLinux VDAに渡されます。
- 非SSOシナリオでは、ユーザーは証明書を選択し、PINを入力してStoreFrontにログオンするように求められます。



ユーザーがStoreFrontでLinux仮想デスクトップセッションを起動すると、Linux VDAへのログオンダイアログボックスが次のように表示されます。ユーザー名はスマートカードの証明書から抽出され、ユーザーはログオン認証のためにPINを再度入力する必要があります。

この動作はWindows VDAと同じです。



スマートカードを使用してセッションに再接続する

セッションに再接続するには、スマートカードがクライアントデバイスに接続されていることを確認してください。そうしないと、スマートカードが接続されていないため再認証が失敗し、Linux VDA側に灰色のキャッシュウィンドウが表示されてすぐに終了します。この場合、スマートカードを接続するように促す他のプロンプトは表示されません。

ただし、StoreFront側では、セッションに再接続しようとしたときにスマートカードが接続されていない場合、StoreFront Webは次のようなアラートを表示することがあります。

スマートカードを挿入するイメージ(/en-us/linux-virtual-delivery-agent/2106/media/please-insert-smart-card.png)

制限事項

スマートカード取り外しポリシー

現在、Linux VDAはスマートカードの取り外しに対してデフォルトの動作のみを使用します。Linux VDAに正常にログオンした後でスマートカードを取り外しても、セッションは接続されたままであり、セッション画面はロックされません。

その他のスマートカードとPKCS#11ライブラリのサポート

サポートリストにはOpenSCスマートカードのみが記載されていますが、Citrixは汎用スマートカードリダイレクトソリューションを提供しているため、他のスマートカードやPKCS#11ライブラリを試すことができます。特定のスマートカードまたはPKCS#11ライブラリに切り替えるには：

1. すべての`opensc-pkcs11.so`インスタンスをPKCS#11ライブラリに置き換えます。
2. PKCS#11ライブラリのパスをレジストリに設定するには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v
"PKCS11LibPath" -d "PATH"
```

ここで、**PATH** は /usr/lib64/pkcs11/opensc-pkcs11.so のようなPKCS#11ライブラリを指します。

3. クライアントで高速スマートカードログオンを無効にします。

ダブルホップシングルサインオン認証

この機能は、StoreFrontストアにアクセスするために入力されたユーザー資格情報を、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10のAuthManagerモジュールに挿入します。挿入後、Linux仮想デスクトップセッション内から仮想デスクトップおよびアプリケーションにアクセスするためにクライアントを使用でき、ユーザー資格情報を再度入力する必要はありません。

注:

この機能は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10でサポートされています。

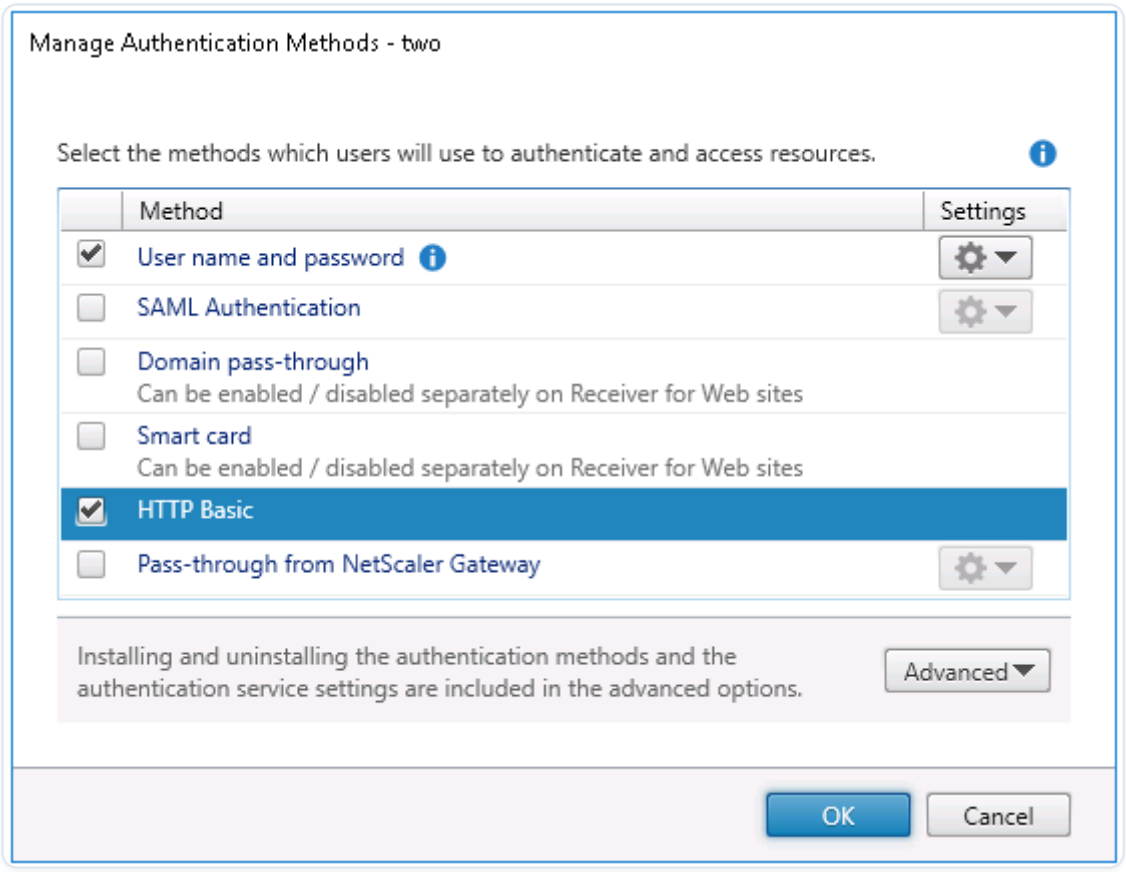
この機能を有効にするには：

1. Linux VDAに、Linux向けCitrix WorkspaceアプリまたはLinux向けCitrix Receiver 13.10をインストールします。

   Citrix WorkspaceアプリまたはCitrix ReceiverのCitrixダウンロードページからアプリをダウンロードします。

   デフォルトのインストールパスは/opt/Citrix/ICAClient/です。アプリを別のパスにインストールする場合は、ICAROOT環境変数を実際のインストールパスを指すように設定します。

2. Citrix StoreFront™管理コンソールで、ターゲットストアに**HTTP Basic**認証方法を追加します。



3. HTTP Basic認証を許可するために、AuthManager構成ファイル（$ICAROOT/config/AuthManConfig.xml）に次のキーを追加します。

```
<Protocols>

     <HTTPBasic>

         <Enabled>True</Enabled>

    </HTTPBasic>

</Protocols>
```

4. 指定されたディレクトリにルート証明書をインストールするには、次のコマンドを実行します。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. この機能を有効にするには、次のコマンドを実行します。

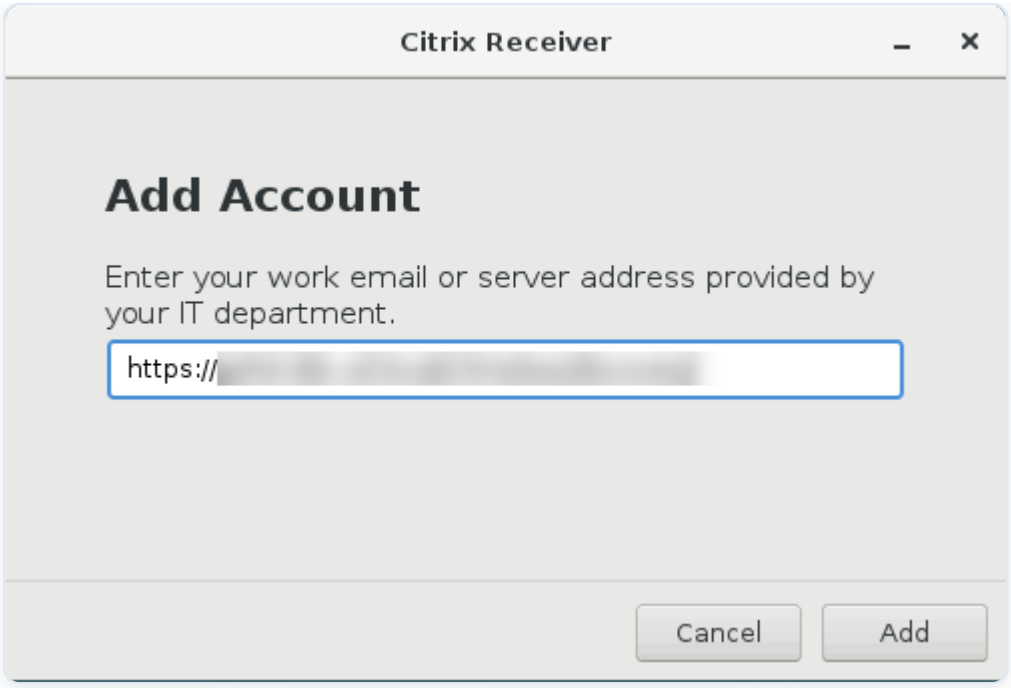
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

6. Linux仮想デスクトップセッションを起動し、そのセッション内でLinux向けCitrix Workspaceアプリまたは Linux向けCitrix Receiver 13.10を開始します。

   Linux仮想デスクトップセッション内でLinux向けCitrix Workspaceアプリまたは Linux向けCitrix Receiver 13.10を初めて起動すると、ストアアカウントの入力を求められます。その後、以前に指定したストアに自動的にログオンされます。

   注:

   ストアアカウントとしてHTTPS URLを入力します。



認証されていないセッションを構成する

この記事の情報を使用して、認証されていないセッションを構成します。この機能を使用するためにLinux VDAをインストールする際に、特別な設定は必要ありません。

注:

> 認証されていないセッションを構成する際、セッションの事前起動はサポートされていないことに注意してください。セッションの事前起動は、Android版Citrix Workspace™アプリでもサポートされていません。

認証されていないストアを作成する

Linux VDAで認証されていないセッションをサポートするには、StoreFrontを使用して認証されていないストアを作成します。

デリバリーグループで認証されていないユーザーを有効にする

認証されていないストアを作成した後、デリバリーグループで認証されていないユーザーを有効にして、認証されていないセッションをサポートします。デリバリーグループで認証されていないユーザーを有効にするには、Citrix Virtual Apps and Desktopsのドキュメントの手順に従ってください。

認証されていないセッションのアイドル時間を設定する

認証されていないセッションのデフォルトのアイドルタイムアウトは10分です。この値は、レジストリ設定 **AnonymousUserIdleTime** を介して構成されます。この値を変更するには、**ctxreg** ツールを使用します。たとえば、このレジストリ設定を5分に設定するには、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v AnonymousUserIdleTime
 -d 0x00000005
```

認証されていないユーザーの最大数を設定する

認証されていないユーザーの最大数を設定するには、レジストリキー **MaxAnonymousUserNumber** を使用します。この設定は、単一のLinux VDAで同時に実行される認証されていないセッションの数を制限します。このレジストリ設定を構成するには、**ctxreg** ツールを使用します。たとえば、値を32に設定するには、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v MaxAnonymousUserNumber
 -d  0x00000020
```

> **重要:**

> 認証されていないセッションの数を制限してください。同時に起動されるセッションが多すぎると、利用可能なメモリが不足するなど、VDAで問題が発生する可能性があります。

トラブルシューティング

認証されていないセッションを構成する際には、次の点を考慮してください。

- **「認証されていないセッションへのログオンに失敗しました。」**.

レジストリが以下を含むように更新されていることを確認してください（0に設定）。

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsd サービスが実行されており、**passwd** キャッシュを有効にするように構成されていることを確認します。

```
ps  uax | grep nscd

cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

passwd キャッシュ変数が有効になっている場合は、それを **no** に設定し、**ncsd** サービスを再起動します。この構成を変更した後、Linux VDA を再インストールする必要がある場合があります。

- **KDE を使用した認証されていないセッションでロック画面ボタンが表示される。**

ロック画面ボタンとメニューは、認証されていないセッションではデフォルトで無効になっています。ただし、KDE では引き続き表示される場合があります。KDE で特定のユーザーのロック画面ボタンとメニューを無効にするには、構成ファイル **$Home/.kde/share/config/kdeglobals** に次の行を追加します。例：

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、`KDE Action Restrictions` パラメーターが `kdeglobals` のようなグローバルな `/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals` ファイルで不変として構成されている場合、ユーザー構成は無効になります。

この問題を解決するには、システム全体の `kdeglobals` ファイルを変更して、`[KDE Action Restrictions]` セクションの `[$i]` タグを削除するか、システム全体の構成を直接使用してロック画面ボタンとメニューを無効にします。KDE の構成の詳細については、「KDE System Administration/Kiosk/Keys page」を参照してください。

LDAPSの構成

セキュアLDAP (LDAPS) を使用すると、Active Directory管理ドメインでセキュア軽量ディレクトリアクセスプロトコルを有効にし、SSL (Secure Socket Layer)/TLS (Transport Layer Security) を介した通信を提供できます。

デフォルトでは、クライアントとサーバーアプリケーション間のLDAP通信は暗号化されていません。SSL/TLS (LDAPS) を使用するLDAPにより、Linux VDAとLDAPサーバー間のLDAPクエリコンテンツを保護できます。

以下のLinux VDAコンポーネントはLDAPSに依存しています:

- ブローカーエージェント: Delivery Controller™へのLinux VDA登録
- ポリシーサービス: ポリシー評価

LDAPSの構成には以下が含まれます:

- Active Directory (AD) および LDAP サーバーで LDAPS を有効にする
- クライアントで使用するためにルートCAをエクスポートする
- Linux VDA で LDAPS を有効または無効にする
- サードパーティプラットフォーム向けにLDAPSを構成する
- SSSDを構成する
- Winbindを構成する
- Centrifyを構成する
- Questを構成する

> **注:**

> LDAPサーバーの監視サイクルを設定するには、次のコマンドを実行します。デフォルト値は15分です。少なくとも10分に設定してください。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --
force
```

AD/LDAPサーバーでLDAPSを有効にする

Microsoftの認証局（CA）またはMicrosoft以外のCAから適切にフォーマットされた証明書をインストールすることで、LDAP over SSL（LDAPS）を有効にできます。

ヒント:

ドメインコントローラーにエンタープライズルートCAをインストールすると、LDAP over SSL/TLS（LDAPS）は自動的に有効になります。

証明書のインストール方法とLDAPS接続の確認方法の詳細については、Microsoftサポートサイトの「How to enable LDAP over SSL with a third-party certification authority」を参照してください。

マルチティア（2層または3層など）の認証局階層を使用している場合、ドメインコントローラー上のLDAPS認証に適した証明書が自動的に提供されるわけではありません。

マルチティア認証局階層を使用してドメインコントローラーでLDAPSを有効にする方法については、Microsoft TechNetサイトの「LDAP over SSL (LDAPS) Certificate」の記事を参照してください。

クライアント使用のためにルート認証局を有効にする

クライアントは、LDAPサーバーが信頼するCAからの証明書を使用する必要があります。クライアントのLDAPS認証を有効にするには、ルートCA証明書を信頼されたキーストアにインポートします。

ルートCAのエクスポート方法の詳細については、MicrosoftサポートWebサイトの「How to export Root Certification Authority Certificate」を参照してください。

Linux VDAでLDAPSを有効または無効にする

Linux VDAでLDAPSを有効または無効にするには、次のスクリプトを（管理者としてログオンした状態で）実行します。

このコマンドの構文には、以下が含まれます。

- 提供されたルートCA証明書を使用してLDAP over SSL/TLSを有効にする:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
```

- チャネルバインディングを使用してLDAP over SSL/TLSを有効にする:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enablecb pathToRootCA
```

> **注:**
>
> チャネルバインディング用のルートCA証明書はPEM形式である必要があります。LDAPSを有効にしてもPython3仮想環境が正常に作成されない場合は、(/ja-jp/linux-virtual-delivery-agent/2106/configuration/create-a-python3-virtual-environment.html)の指示に従って手動で作成してください。

- SSL/TLSなしでLDAPにフォールバックする

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
```

LDAPS専用のJavaキーストアは**/etc/xdl/.keystore**にあります。影響を受けるレジストリキーは次のとおりです。

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers


HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy


HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS
```

```
HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore
```

```
HKLM\Software\Citrix\VirtualDesktopAgent\EnableChannelBinding
```

サードパーティプラットフォームのLDAPSを構成する

Linux VDAコンポーネントの他に、SSSD、Winbind、Centrify、Questなど、VDAに準拠するいくつかのサードパーティ製ソフトウェアコンポーネントもセキュアLDAPを必要とする場合があります。以下のセクションでは、LDAPS、STARTTLS、またはSASL署名とシーリングを使用してセキュアLDAPを構成する方法について説明します。

ヒント:

これらのソフトウェアコンポーネントのすべてが、セキュアLDAPを確保するためにSSLポート636を使用することを好むわけではありません。また、ほとんどの場合、LDAPS（ポート636のSSL経由LDAP）はポート389のSTARTTLSと共存できません。

SSSD

オプションに従って、ポート636またはポート389でSSSDセキュアLDAPトラフィックを構成します。詳細については、SSSD LDAP Linuxのmanページを参照してください。

ウィンバインド

ウィンバインドLDAPクエリはADSメソッドを使用します。ウィンバインドはポート389でのStartTLSメソッドのみをサポートしています。影響を受ける構成ファイルは、**/etc/openldap/ldap.conf**にある**ldap.conf**と、**/etc/samba/smb.conf**にある**smb.conf**です。ファイルを次のように変更します。

```
ldap.conf:
```

```
TLS_REQCERT never
```

```
smb.conf:



ldap ssl = start tls

ldap ssl ads = yes

client ldap sasl wrapping = plain
```

あるいは、セキュアLDAPはSASL GSSAPI署名とシーリングによって構成できますが、TLS/SSLとは共存できません。SASL暗号化を使用するには、**smb.conf**構成を変更します。

```
smb.conf:



ldap ssl = off

ldap ssl ads = no

client ldap sasl wrapping = seal
```

セントリファイ

CentrifyはポートでのLDAPSをサポートしていません。ただし、ポート389でセキュアな暗号化を提供します。詳細については、Centrifyサイトを参照してください。

クエスト

Quest Authentication Serviceはポート636でのLDAPSをサポートしていませんが、異なる方法を使用してポート389でセキュアな暗号化を提供します。

トラブルシューティング

この機能を使用すると、以下の問題が発生する可能性があります。

- **LDAPSサービスの可用性**

  AD/LDAPサーバーでLDAPS接続が利用可能であることを確認してください。ポートはデフォルトで636です。

- **LDAPSが有効な場合にLinux VDAの登録が失敗しました**

  LDAPサーバーとポートが正しく構成されていることを確認してください。まずルートCA証明書を確認し、それがAD/LDAPサーバーと一致していることを確認してください。

- **誤ってレジストリが変更された**

  enable_ldaps.shを使用せずにLDAPS関連のキーが誤って更新された場合、LDAPSコンポーネントの依存関係が壊れる可能性があります。

- **Wiresharkまたはその他のネットワーク監視ツールからSSL/TLS経由でLDAPトラフィックが暗号化されていない**

  デフォルトでは、LDAPSは無効になっています。強制的に有効にするには、**/opt/Citrix/VDA/sbin/enable_ldaps.sh**を実行してください。

- **Wiresharkまたはその他のネットワーク監視ツールからLDAPSトラフィックがない**

  LDAP/LDAPSトラフィックは、Linux VDAの登録とグループポリシーの評価が行われるときに発生します。

- **ADサーバーでldp connectを実行してLDAPSの可用性を確認できませんでした**

  IPアドレスの代わりにAD FQDNを使用してください。

- **/opt/Citrix/VDA/sbin/enable_ldaps.sh スクリプトを実行してルートCA証明書のインポートに失敗しました**

  CA証明書のフルパスを指定し、ルートCA証明書が正しいタイプであることを確認してください。これは、Java Keytoolでサポートされているほとんどのタイプと互換性があるはずです。サポートリストにない場合は、まずタイプを変換できます。証明書形式の問題が発生した場合は、base64エンコードされたPEM形式をお勧めします。

- **Keytool -list でルートCA証明書を表示できませんでした**

  `/opt/Citrix/VDA/sbin/enable_ldaps.sh` を実行して LDAPS を有効にすると、証明書は /etc/xdl/.keystore にインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

Python3仮想環境を作成する

ネットワークに接続している場合、`sudo /opt/Citrix/VDA/bin/xdping`または`/opt/Citrix/VDA/sbin/enable_ldaps.sh`コマンドを実行すると、Python3仮想環境を作成できます。ただし、コマンドでPython3仮想環境を作成できない場合は、ネットワーク接続がなくても手動で作成

できます。この記事では、ネットワーク接続なしでPython3仮想環境を作成するための前提条件と手順について詳しく説明します。

前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3`ディレクトリにアクセスするための管理者権限が必要です。
- `Python3`パッケージのwheelファイルが配置されていること。wheelファイルは`<https://pypi.org/>`からダウンロードできます。

Python3仮想環境を作成する

Python3仮想環境を作成するには、次の手順を完了します。

1. Python3の依存関係をインストールします。

   RHELの場合：

   ```
   yum -y install python36-devel krb5-devel gcc
   ```

   注:

   一部の依存関係をインストールするには、特定のレポジトリを有効にする必要がある場合があります。RHEL 7の場合は、`subscription-manager repos --enable rhel-7-server-optional-rpms`コマンドを実行します。RHEL 8の場合は、`subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms`コマンドを実行します。

   Ubuntu、Debianの場合：

   ```
   apt-get -y install python3-dev python3-pip python3-venv libkrb5-dev
   ```

   SUSEの場合：

   ```
   zypper -i -n install python3-devel python3-setuptools krb5-devel gcc
   libffi48-devel
   ```

> **注:**
>
> 一部の依存関係をインストールするには、`SUSE_Linux_Enterprise_Software_Development_Kit_12_SP5_x86_64`レポジトリを有効にする必要がある場合があります。

2. Python3仮想環境を作成します。

> **注:**
>
> pipツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼済みホストを/etc/pip.confファイルに追加することを検討してください。

\>`[global]`

`trusted-host =`

\>`pypi.org`

`files.pythonhosted.org`

> **RHEL、Ubuntu、Debianの場合：**

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

SUSEの場合:

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
setuptools==40.6.2
```

3. LDAPSの依存関係をインストールします。

   RHEL、Ubuntu、Debianの場合:

   ```
   sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install cffi ==
   1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi==1.6.2
   ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0 psutil == 5.8.0
   pyasn1 == 0.4.8 pycparser == 2.20 scramp == 1.2.0 six == 1.15.0
   termcolor == 1.1.0
   ```

   SUSEの場合:

   ```
   sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install cffi
   == 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi==1.6.2
   ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0 psutil == 5.8.0
   pyasn1 == 0.4.8 pycparser == 2.20 scramp == 1.2.0 six == 1.15.0
   termcolor == 1.1.0
   ```

4. `XDPing`の依存関係をインストールします。

   RHEL、Ubuntu、Debianの場合:

   ```
   sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install cffi ==
   1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi==1.6.2
   ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0 psutil == 5.8.0
   pyasn1 == 0.4.8 pycparser == 2.20 scramp == 1.2.0 six == 1.15.0
   termcolor == 1.1.0



   sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
   Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
   ```

   SUSEの場合:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install cffi
== 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi==1.6.2
ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0 psutil == 5.8.0
pyasn1 == 0.4.8 pycparser == 2.20 scramp == 1.2.0 six == 1.15.0
termcolor == 1.1.0



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

エックスディーピング

説明

Linux XDPingツールは、Linux VDA環境における一般的な構成の問題をチェックするプロセスを自動化する、コマンドラインベースのアプリケーションです。

Linux XDPingツールは、システム上で150以上の個別のテストを実行します。これらは大まかに次のように分類されます。

- Linux VDAのシステム要件が満たされているかを確認します
- Linuxディストリビューションを含むマシン情報を特定して表示します
- Linuxカーネルの互換性を確認します
- Linux VDAの動作に影響を与える可能性のある、既知のLinuxディストリビューションの問題を確認します
- Security-Enhanced Linux (SELinux) のモードと互換性を確認します
- ネットワークインターフェイスを特定し、ネットワーク設定を確認します
- ストレージのパーティション分割と利用可能なディスク容量を確認します
- マシンのホスト名とドメイン名の構成を確認します
- DNS構成を確認し、ルックアップテストを実行します
- 基盤となるハイパーバイザーを特定し、仮想マシンの構成を確認します。サポート対象:
  - シトリックス ハイパーバイザー™
  - マイクロソフト ハイパーV
  - ヴイエムウェア ブイスフィア

- タイム設定を確認し、ネットワーク時刻同期が動作しているか確認します
- PostgreSQLサービスが適切に構成され、動作しているか確認します
- ファイアウォールが有効になっており、必要なポートが開いているか確認します
- Kerberos構成を確認し、認証テストを実行します
- グループポリシーサービスエンジン向けのLDAP検索環境を確認します
- Active Directory統合が適切に設定されており、現在のマシンがドメインに参加しているか確認します。サポート対象:
  - サンバ ウィンバインド
  - デル クエスト オーセンティケーション サービス
  - セントリファイ ダイレクトコントロール
  - SSSD
- Active Directory内のLinuxコンピューターオブジェクトの整合性を確認します
- Pluggable Authentication Module (PAM) 構成を確認します
- コアダンプパターンを確認します
- Linux VDAに必要なパッケージがインストールされているか確認します
- Linux VDAパッケージを特定し、インストールの整合性を確認します
- PostgreSQLレジストリデータベースの整合性を確認します。
- Linux VDAサービスが適切に構成され、動作しているかを確認します。
- VDAおよびHDX™構成の整合性を確認します。
- 各構成済みDelivery Controller™をプローブして、Broker Serviceが到達可能で、動作しており、応答性があることをテストします。
- マシンがDelivery Controllerファームに登録されているかを確認します。
- アクティブまたは切断された各HDXセッションの状態を確認します。
- Linux VDA関連のエラーと警告についてログファイルをスキャンします。
- Xorgのバージョンが適切であるかを確認します。

Linux XDPingツールを使用する

> **注:**

> ctxsetup.shを実行してもXDPingはインストールされません。XDPingをインストールするには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。 このコマンドは、XDPingに必要なPython3仮想環境も作成します。このコマンドでPython3仮想環境の作成に失敗した場合は、Python3仮想環境の作成の手順に従って手動で作成してください。

> XDPingには、コマンドシェルから実行される`xdping`という単一の実行可能ファイルが付属しています。

コマンドラインオプションを表示するには、`--help`オプションを使用します。

```
sudo /opt/Citrix/VDA/bin/xdping --help
```

テストの全スイートを実行するには、コマンドラインオプションなしで`xdping`を実行します。

```
sudo /opt/Citrix/VDA/bin/xdping
```

Linux VDAパッケージをインストールする前に環境を確認するには、`pre-flight`テストを実行します。

```
sudo /opt/Citrix/VDA/bin/xdping --preflight
```

特定のテストカテゴリのみ（たとえば、時間テストやKerberosテスト）を実行するには、`-T`オプションを使用します。

```
sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos
```

特定のXenDesktop® Controllerをプローブするには:

```
sudo /opt/Citrix/VDA/bin/xdping -d myddc.domain.net
```

サンプル出力

以下は、Kerberosテストを実行したときの出力例です:

```
sudo xdping -T kerberos

Root User ------------------------------------------------------------------
    User:         root
    EUID:         0
      Verify user is root                                                [Pass]

Kerberos -------------------------------------------------------------------
    Kerberos version: 5
      Verify Kerberos available                                          [Pass]
      Verify Kerberos version 5                                          [Pass]
    KRB5CCNAME:   [Not set]
                  Distro default FILE:/tmp/krb5cc_%{uid}
    KRB5CCNAME type: [Supported]
    KRB5CCNAME format: [Default]
      Verify KRB5CCNAME cache type                                       [Pass]
      Verify KRB5CCNAME format                                           [Pass]
    Configuration file: /etc/krb5.conf [Exists]
      Verify Kerberos configuration file found                           [Pass]
  Keytab file:  /etc/krb5.keytab [Exists]
  Default realm: XD2.LOCAL
  Default realm KDCs: [NONE SPECIFIED]
  Default realm domains: [NONE SPECIFIED]
  DNS lookup realm: [Enabled]
  DNS lookup KDC: [Enabled]
  Weak crypto:  [Disabled]
  Clock skew limit:  300 s
    Verify system keytab file exists                                     [Pass]
    Verify default realm set                                             [Pass]
    Verify default realm in upper-case                                   [Pass]
    Verify default realm not EXAMPLE.COM                                 [Pass]
    Verify default realm domain mappings                                 [Pass]
    Verify default realm master KDC configured                           [Pass]
    Verify Kerberos weak crypto disabled                                 [Pass]
    Verify Kerberos clock skew setting                                   [Pass]
  Default ccache: [Not set]
                  Distro default FILE:/tmp/krb5cc_%{uid}
  Default ccache type: [Supported]
  Default ccache format: [Default]
    Verify default credential cache cache type                           [Pass]
    Verify default credential cache format                               [Pass]
  UPN system key [MYVDA1$@[redacted]]: [MISSING]
  SPN system key [host/[redacted]]: [Exists]
    Verify Kerberos system keys for UPN exist                            [ERROR]
    No system keys were found for the user principal name (UPN) of
    the machine account. For the Linux VDA to mutually authenticate
    with the Delivery Controller, the system keytab file must
    contain keys for both the UPN and host-based SPN of the machine
    account.
```

```
        Verify Kerberos system keys for SPN exist                    [Pass]
      Kerberos login: [FAILED AUTHENTICATION]
                      Keytab contains no suitable keys for MYVDA1$@[redacted]
                      while getting initial credentials
        Verify KDC authentication                                    [ERROR]
        Failed to authenticate and obtain a Ticket Granting Ticket (TGT)
        from the KDC authentication service for the machine account UPN
        MYVDA1$@[redacted]. Check that the Kerberos configuration is
        valid and the keys in the system keytab are current.


  Summary ------------------------------------------------------------------
    The following tests did not pass:
        Verify Kerberos system keys for UPN exist                    [ERROR]
        Verify KDC authentication                                    [ERROR]
```

Xauthorityを構成する

Linux VDAは、対話型リモート処理のためにX11ディスプレイ機能（`xterm`および`gvim`を含む）を使用する環境をサポートしています。この機能は、XClientとXServer間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信のアクセス許可を保護するには、2つの方法があります。

- **Xhost**。デフォルトでは、XhostはlocalhostのXClientのみがXServerと通信することを許可します。リモートXClientがXServerにアクセスすることを許可する場合は、特定のコンピューターでアクセス許可を付与するためにXhostコマンドを実行する必要があります。または、代わりに**xhost +**を使用して、任意のXClientがXServerに接続できるようにすることもできます。
- **Xauthority**。`.Xauthority`ファイルは、各ユーザーのホームディレクトリにあります。これは、xauthによるXServerの認証に使用されるCookieに資格情報を保存するために使用されます。XServerインスタンス（Xorg）が起動すると、Cookieはその特定のディスプレイへの接続を認証するために使用されます。

仕組み

Xorgが起動すると、`.Xauthority`ファイルがXorgに渡されます。この`.Xauthority`ファイルには、以下の要素が含まれています。

- ディスプレイ番号
- リモート要求プロトコル
- Cookie番号

`xauth`コマンドを使用してこのファイルを参照できます。例：

```
# xauth -f ~/.Xauthority



# > list



# > us01msip06:107  MIT-MAGIC-COOKIE-1  fb228d1b695729242616c5908f11624b
```

XClientがXorgにリモートで接続する場合、2つの前提条件を満たす必要があります。

- **DISPLAY**環境変数をリモートXServerに設定します。
- Xorg内のCookie番号のいずれかが含まれている`.Xauthority`ファイルを取得します。

Xauthorityを構成する

リモートX11ディスプレイ用にLinux VDAでXauthorityを有効にするには、以下の2つのレジストリキーを作成する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"XauthEnabled" -d "0x00000001" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"ListenTCP" -d "0x00000001" --force
```

Xauthority を有効にした後、`.Xauthority` ファイルを XClient に手動で渡すか、共有ホームディレクトリをマウントして渡します。

- `.Xauthority` ファイルを XClient に手動で渡す

  ICA® セッションの起動後、Linux VDA は XClient 用の `.Xauthority` ファイルを生成し、ログオンユーザーのホームディレクトリに保存します。この `.Xauthority` ファイルをリモートの XClient マシンにコピーし、`DISPLAY` および `XAUTHORITY` 環境変数を設定できます。`DISPLAY` は `.Xauthority` ファイルに保存されているディスプレイ番号で、`XAUTHORITY` は Xauthority のファイルパスです。例については、次のコマンドを参照してください。

```
export DISPLAY={Display number stored in the Xauthority file}



export XAUTHORITY={the file path of .Xauthority}
```

注:

`XAUTHORITY` 環境変数が設定されていない場合、デフォルトで `~/.Xauthority` ファイルが使用されます。

- 共有ホームディレクトリをマウントして `.Xauthority` ファイルを XClient に渡す

  便利な方法は、ログオンユーザー用に共有ホームディレクトリをマウントすることです。Linux VDA が ICA セッションを開始すると、ログオンユーザーのホームディレクトリの下に `.Xauthority` ファイルが作成されます。このホームディレクトリが XClient と共有されている場合、ユーザーはこの `.Xauthority` ファイルを XClient に手動で転送する必要はありません。`DISPLAY` および `XAUTHORITY` 環境変数が正しく設定されると、GUI は XServer デスクトップに自動的に表示されます。

トラブルシューティング

Xauthority が機能しない場合、次のトラブルシューティング手順に従ってください。

1. root 権限を持つ管理者として、すべての Xorg クッキーを取得します。

   ```
   ps aux | grep -i xorg
   ```

   このコマンドは、Xorg プロセスと、起動時に Xorg に渡されたパラメーターを表示します。別のパラメーターは、どの `.Xauthority` ファイルが使用されているかを表示します。例：

   ```
   /var/xdl/xauth/.Xauthority110
   ```

   Xauth コマンドを使用してクッキーを表示します。

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   ```

2. `Xauth` コマンドを使用して、`~/.Xauthority` に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーは Xorg と XClient の `.Xauthority` ファイルで同じである必要があります。
3. クッキーが同じである場合、Linux VDA の IP アドレス（例：10.158.11.11）と公開デスクトップのディスプレイ番号（例：160）を使用して、リモートディスプレイポートのアクセシビリティを確認します。

   XClient マシンで次のコマンドを実行します。

   ```
   telnet 10.158.11.11 6160
   ```

   ポート番号は 6000 + \<display number\> の合計です。

   この telnet 操作が失敗した場合、ファイアウォールが要求をブロックしている可能性があります。

フェデレーション認証サービスを構成する

Linux VDAは、FASを使用してCitrix Virtual Apps and Desktops環境にログオンすることをサポートしています。FASログオン機能には、Windows VDAと同じWindows環境を使用します。FASのWindows環境の構成については、「フェデレーション認証サービス」を参照してください。この記事では、Linux VDAに特化した追加情報を提供します。

> **注**
>
> Linux VDAは、**セッション内動作**ポリシーをサポートしていません。

> Linux VDAは、FASサーバーとのデータ送信に短い接続を使用します。

Linux VDAでFASを構成する

RHEL 8/CentOS 8でのFASサポート

FASは、RHEL 8/CentOS 8で非推奨となっているpam_krb5モジュールに依存します。RHEL 8/CentOS 8でFASを使用するには、次のようにpam_krb5モジュールをビルドします。

1. 次のWebサイトからpam_krb5-2.4.8-6のソースコードをダウンロードします。

`<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>`。

2. RHEL 8/CentOS 8にpam_krb5モジュールをビルドしてインストールします。

```
yum install make gcc krb5-devel pam-devel autoconf libtool

rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio -div

tar xvzf pam_krb5-2.4.8.tar.gz

cd pam_krb5-2.4.8

./configure --prefix=/usr

make

make install
```

3. /usr/lib64/security/の下にpam_krb5.soが存在することを確認します。

```
ls -l /usr/lib64/security | grep pam_krb5
```

FASサーバーを設定する

Linux VDAを新規インストールする場合、FASを使用するには、ctxinstall.shまたはctxsetup.shの実行中にCTX_XDL_FAS_LISTを求められたときに、各FASサーバーのFQDNを入力します。Linux VDAはADグループポリシーをサポートしていないため、代わりにFASサーバーのセミコロン区切りリストを指定できます。サーバーアドレスが削除された場合は、その空白を**\<none\>**というテキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。

既存のLinux VDAインストールをアップグレードする場合は、`ctxsetup.sh`を再実行してFASサーバーを設定できます。または、次のコマンドを実行してFASサーバーを設定し、`ctxvda`サービスを再起動して設定を有効にすることもできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

```
service ctxjproxy restart



service ctxvda restart
```

`ctxreg`を介してFASサーバーを更新するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -v "Addresses" -d "<Your-FAS-Server-List>"



service ctxjproxy restart



service ctxvda restart
```

証明書をインストールする

ユーザーの証明書を検証するには、ルートCA証明書とすべての中間証明書をVDAにインストールします。たとえば、ルートCA証明書をインストールするには、前の「**Microsoft CA (AD上) からCA証明書を取得する**」の手順からADルート証明書を取得するか、ルートCAサーバーからそのDER形式をダウンロードします`http://CA-SERVER/certsrv`。

> **注:**
>
> 次のコマンドは、中間証明書の設定にも適用されます。
>
> 次のようなコマンドを実行して、DERファイル（.crt、.cer、.der）をPEMに変換します。

```
sudo openssl x509 -inform der -in root.cer -out root.pem
```

次に、次のようなコマンドを実行して、ルートCA証明書を`openssl`ディレクトリにインストールします。

```
sudo cp root.pem /etc/pki/CA/certs/
```

注:

ルートCA証明書を **/root** パスに配置しないでください。そうしないと、FASはルートCA証明書への読み取り権限を持ちません。

ctxfascfg.sh を実行する

FASパラメーターを設定するには、ctxfascfg.shスクリプトを実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxfascfg.sh
```

`ctxfascfg.sh` をサイレントモードで実行できるように、2つの環境変数が追加されます。

- **CTX_FAS_ADINTEGRATIONWAY=winbind \| sssd \| centrify** – Active Directory統合方法を示します。これは、`CTX_EASYINSTALL_ADINTEGRATIONWAY` が指定されている場合に `CTX_EASYINSTALL_ADINTEGRATIONWAY` と等しくなります。
  `CTX_EASYINSTALL_ADINTEGRATIONWAY` が指定されていない場合、
  `CTX_FAS_ADINTEGRATIONWAY` は独自の値を設定します。
- **CTX_FAS_CERT_PATH \=\<certificate path\>** – ルート証明書とすべての中間証明書を格納する完全なパスを指定します。

正しいActive Directory統合方法を選択し、証明書の正しいパス（例: `/etc/pki/CA/certs/`）を入力します。

スクリプトは、krb5-pkinitおよびpam_krb5パッケージをインストールし、関連する構成ファイルを設定します。

制限事項

- FASは、限られたLinuxプラットフォームとAD統合方法をサポートしています。次のマトリックスを参照してください。

	ウィンバインド	SSSD	セントリファイ
RHEL 8.3 / CentOS 8.3	はい	はい	はい

	ウィンバインド	SSSD	セントリファイ
RHEL 8.2 / CentOS 8.2	はい	はい	はい
RHEL 8.1 / CentOS 8.1	はい	はい	はい
RHEL 7.9 / CentOS 7.9	はい	はい	はい
RHEL 7.8 / CentOS 7.8	はい	はい	はい
ウブントゥ 20.04	はい	いいえ	はい
ウブントゥ 18.04	はい	いいえ	はい
ウブントゥ 16.04	はい	いいえ	はい
SLES 12.5	はい	いいえ	はい

- FAS はまだロック画面をサポートしていません。セッションでロックボタンをクリックすると、FAS を使用してそのセッションに再度ログオンすることはできません。
- このリリースでは、(/ja-jp/federated-authentication-service/2106/deployment-architectures.html)記事「Federated Authentication Service のアーキテクチャの概要」にまとめられている一般的な FAS 展開のみがサポートされており、**Windows 10 Azure AD Join** は含まれていません。

トラブルシューティング

FAS のトラブルシューティングを行う前に、Linux VDA が正しくインストールおよび構成されており、パスワード認証を使用して共通ストアで非 FAS セッションを正常に起動できることを確認してください。

非 FAS セッションが正常に機能する場合は、**Login** クラスの HDX ログレベルを VERBOSE に、VDA ログレベルを TRACE に設定します。Linux VDA のトレースログを有効にする方法については、Knowledge Center の記事 CTX220130 を参照してください。

FAS サーバー構成エラー

FASストアからのセッション起動に失敗します。

/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [Logon
Type] Federated Authentication Logon.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: entry



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
connect to server 0



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0: failed
to connect: Connection refused.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: failed
to connect to server [0], please confirm if fas service list is well
configurated in condb



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit,
43



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: failed
to validate fas credential



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate:
failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER
```

解決策

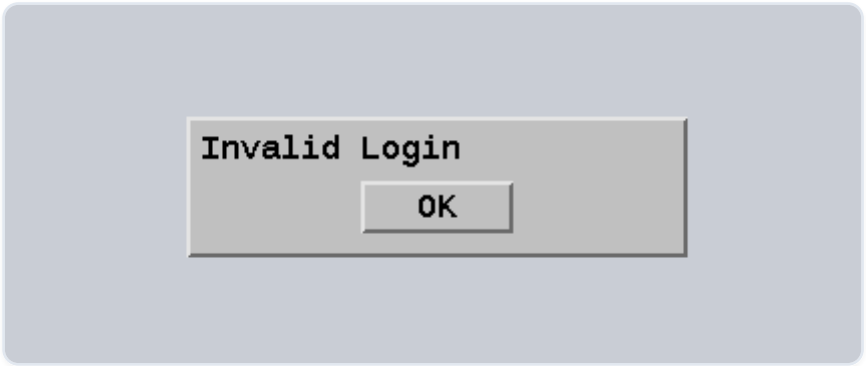
次のコマンドを実行して、Citrixレジストリ値「HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\VirtualDesktopAgent\\Authentication\\UserCredentialService」が\<Your-FAS-Server-List\>に設定されていることを確認します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

既存の設定が正しくない場合は、前の FASサーバーの設定 の手順に従って再度設定してください。

CA証明書の設定が正しくありません

FASストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。



/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: entry



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller: current
process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_public_certificate: entry



2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
for response...
```

```
2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query to
server success

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
get_public_certificate: exit

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
input size 1888

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
output size 1415

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: get logon certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
cache certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: exit, 0

2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策

ルートCA証明書とすべての中間証明書を格納する完全なパスが/etc/krb5.confに正しく設定されていることを確認します。完全なパスは次のようになります。

```
 [realms]

EXAMPLE.COM = {

    ......

    pkinit_anchors = DIR:/etc/pki/CA/certs/

    ......

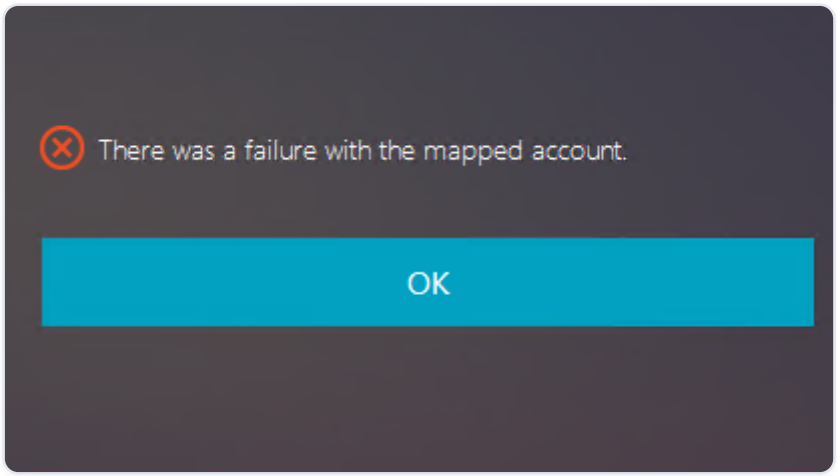
}
```

既存の設定が正しくない場合は、前の 証明書のインストール の手順に従って再度設定してください。

または、ルートCA証明書が有効であるかどうかを確認します。

シャドウアカウントマッピングエラー

FASはSAML認証によって構成されます。ADFSユーザーがADFSログオンページでユーザー名とパスワードを入力した後、次のエラーが発生する可能性があります。



このエラーは、ADFSユーザーは正常に検証されたものの、ADにシャドウユーザーが構成されていないことを示しています。

解決策

ADでシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に、次のエラーが発生します。

ADFSが構成されていません(/en-us/linux-virtual-delivery-agent/2106/media/adfs-not-configured.png)

この問題は、FASストアがSAML認証を使用するように構成されているにもかかわらず、ADFS展開が不足している場合に発生します。

解決策

フェデレーション認証サービス用にADFS IdPを展開します。詳細については、「フェデレーション認証サービスADFS展開」を参照してください。

関連情報

- 一般的なFAS展開については、「フェデレーション認証サービスアーキテクチャの概要」の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービス詳細構成」の章で紹介されています。

既知の問題

FASを使用している場合、非英語文字を含む公開デスクトップまたはアプリセッションを起動しようとすると失敗することがあります。

非英語文字を含むセッションの起動に失敗しました(/en-us/linux-virtual-delivery-agent/2106/media/launching-nonenglish-session-fails.png)

回避策

CAツールで**Manage Templates**を右クリックし、**Citrix_SmartcardLogon**テンプレートを**Build from this Active Directory information**から**Supply in the request**に変更します。

リクエストオプションでの提供(/en-us/linux-virtual-delivery-agent/2106/media/supply-in-the-request.png)

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 2106のドキュメントに対する重要な変更点を示します。これらのアップデートに関する通知を受け取るには、RSSフィードを購読してください。

日付	変更	説明
16 Jun 2021	Linux Virtual Delivery Agent 2106のリリース	Linux Virtual Delivery Agent 2106の新機能