Linux Virtual Delivery Agent 2203 長期サービスリリース (LTSR)

累積更新プログラム 6 (CU6) は、Linux Virtual Delivery Agent 2203 LTSR の最新リリースです。CU6 は、Linux Virtual Delivery Agent の 2203 LTSR CU5 リリースと比較して、3つの修正を追加しています。

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 5 (CU5)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 4 (CU4)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 3 (CU3)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 2 (CU2)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 1 (CU1)

Linux Virtual Delivery Agent 2203 LTSR (初回リリース)

> **重要:**

> 現在のリリース (CR) および長期サービスリリース (LTSR) の製品ライフサイクル戦略については、Lifecycle Milestonesに記載されています。

> Linux Virtual Delivery Agent (VDA) を使用すると、Citrix Workspace™ アプリがインストールされている任意のデバイスから、どこからでも Linux 仮想アプリとデスクトップにアクセスできます。

サポートされているLinuxディストリビューション に基づいて仮想アプリとデスクトップを配信できます。Linux仮想マシンにVDAソフトウェアをインストールし、Delivery Controllerを構成してから、Citrix Studioを使用してアプリとデスクトップをユーザーに提供します。

新機能

累積更新プログラム 7 (CU7) は、Linux Virtual Delivery Agent 2203 LTSR の最新リリースです。CU7 には、修正された問題は含まれていません。

累積更新プログラム 7 (CU7)

リリース日: Aug 28, 2025

新機能

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 7 (CU7) には、修正は追加されていません。

リナックス バーチャル デリバリー エージェント 2203 LTSR 累積更新 6 (CU6)

リナックス バーチャル デリバリー エージェント 2203 LTSR 累積更新 5 (CU5)

リナックス バーチャル デリバリー エージェント 2203 LTSR 累積更新 4 (CU4)

リナックス バーチャル デリバリー エージェント 2203 LTSR 累積更新 3 (CU3)

Linux仮想デリバリーエージェント 2203 LTSR 累積更新 2 (CU2)

Linux仮想デリバリーエージェント 2203 LTSR 累積更新 1 (CU1)

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

累積更新プログラム 6 (CU6)

リリース日: Jan 08, 2025

新機能

RHEL 8.10およびRocky Linux 8.10のサポート

Linux VDAは、以下のLinuxディストリビューションをサポートするようになりました。

- RHEL 8.10
- ロッキー・リナックス 8.10

サポートされているLinuxディストリビューションの詳細については、「システム要件」を参照してください。

OpenJDKの依存関係がバージョン11からバージョン17にアップグレードされました

このリリースでは、OpenJDKの依存関係がバージョン11からバージョン17にアップグレードされました。Linux VDAをインストールすると、OpenJDK 17が依存関係として自動的にインストールされます。詳細については、「Linux VDAを手動でインストールする」を参照してください。

サービスアカウント「ctxsrvr」が新しい場所に移動されました

このリリースから、Citrixサービスアカウント「ctxsrvr」は/home/ctxsrvrから/var/lib/ctxsrvrに移動されました。

ルートレスXorg

Linux VDAは、非ルートユーザー権限でXorgを実行することをサポートするようになりました。これは「ルートレス」Xorgとも呼ばれます。ルートレスXorgは、ルートとして実行する場合と比較して、セキュリティが大幅に向上します。詳細については、「ルートレスXorg」を参照してください。

グラフィックパフォーマンスの向上

このリリースでは、Linux VDAでJPEG Turboをバージョン3（V3）にアップグレードすることにより、グラフィックパフォーマンスが向上します。

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム6（CU6）は、Linux Virtual Delivery Agentの2203 LTSR累積更新プログラム5リリースと比較して、3つの修正を追加します。

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 5 (CU5)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 4 (CU4)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 3 (CU3)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 2 (CU2)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 1 (CU1)

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

2203 LTSR CU6で修正された問題

Linux Virtual Delivery Agent 2203 LTSR CU5以降で、以下の問題が修正されました:

- リナックス CWAへの再接続またはセッションローミングを実行すると、HDXセッションがハングする場合があります。これは、特にwebcamtest.com/checkでウェブカメラをテストしている際に発生します。
- 断続的な認証エラーが発生し、「AgentAuthInterceptor.HandleMessage: Message failed authentication. Error: Client failed authentication.」というエラーが表示されることがあります。これは、複数のLDAPエラーが立て続けにログに記録された場合に発生し、ログインの問題を引き起こし、システムがStudioで「未登録」状態になります。
- ルートユーザーは、CDMを使用してマップされたクライアントデバイスに読み取り専用ファイルをコピーできません。

累積更新プログラム 5 (CU5)

リリース日: Jun 04, 2024

新機能

LDAPクエリパフォーマンスの最適化

新しいリリースでは、LDAPサーバーへの不要な通信が削除され、インデックス付きクエリキーが使用されることで、LDAPクエリのパフォーマンスが向上し、コストが削減されます。

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 5 (CU5)には、Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 4リリースと比較して、4つの修正が追加されています。

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム4 (CU4)

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム3 (CU3)

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム2 (CU2)

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム1 (CU1)

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

2203 LTSR CU5 で修正された問題

Linux Virtual Delivery Agent 2203 LTSR CU4 以降、以下の問題が修正されました。

- TLS/DTLS が無効になっている場合でも、Linux VDA は TLS/DTLS ポート番号を誤って 443 と報告します。
- SETPOS コマンドのコールバック通知は受信されるものの、以前の MINIMISE 復元操作のコールバックが受信されないため、クライアントウィンドウがまだ最小化状態にあると仮定され、Linux VDA はクライアント側の最新のアプリケーション POS に同期しません。
- VDA が 10 ～ 15 分間使用されていない場合、画面が応答しなくなり、この問題を解決するには VDA を再接続/リセットする必要があります。
- GPF ファイルから大量のグループポリシーセットを処理する際に、グループポリシーパーサーが CPU を 100% 消費する可能性があります。。

累積更新プログラム 4 (CU4)

リリース日: November 16, 2023

新機能

RHEL 8.6、RHEL 8.8、Rocky Linux 8.6、Rocky Linux 8.8のサポート

CU4リリース以降、Linux VDAは以下のLinuxディストリビューションをサポートします。

- RHEL 8.6
- ロッキー Linux 8.6
- RHEL 8.8

- ロッキー Linux 8.8

詳細については、システム要件を参照してください。

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 4 (CU4)は、Linux Virtual Delivery Agentの2203 LTSR Cumulative Update 3リリースと比較して、3つの修正を追加します。

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 3 (CU3)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 2 (CU2)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 1 (CU1)

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

2203 LTSR CU4 で修正された問題

Linux Virtual Delivery Agent 2203 LTSR CU3 以降で、以下の問題が修正されました。

- Linux VDA で、**PDF ユニバーサルプリンターの自動作成**ポリシーが有効になっている場合、Citrix Universal™ Print Driver (UPD) を使用して PDF ファイルを印刷しようとすると失敗する場合があります。この問題は、10 を超える同時ユーザーセッションが VDA に接続されている場合に発生します。
- Linux VDA で構成されている場合、Machine Creation Services™ (MCS) が Citrix Cloud に登録できない場合があります。
- 同時接続数が既に確立されている場合、VDA の登録が失敗する場合があります。

累積更新プログラム 3 (CU3)

リリース日: June 1, 2023

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 3 (CU3) には、Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 2 リリースと比較して、6つの修正が追加されています。

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 2 (CU2)

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 1 (CU1)

Linux Virtual Delivery Agent (初期リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品の Subscription Advantage 資格日

2203 LTSR CU3 で修正された問題

Linux Virtual Delivery Agent 2203 LTSR CU2 以降、以下の問題が修正されました。

- ctxgfx プロセスで発生するセグメンテーション違反エラーにより、Linux 仮想アプリとデスクトップが予期せず終了する可能性があります。
- Linux VDA セッションを開始しようとすると、グレースクリーンになることがあり、その結果、セッションが最終的に切断されます。
- Cloud Connector を再起動すると、Linux VDA が登録解除されることがあります。
- セッション切断後、スマートカードプロセスが予想よりも多くの CPU を使用する可能性があります。
- RPM の問題により、マシンカタログのアップグレードまたは作成が失敗する可能性があります。その結果、Linux VDA バージョン LTSR CU1 または CU2 へのアップグレードが失敗します。
- VDA が登録状態であっても、Linux VDA セッションを開始しようとすると失敗することがあります。
- この修正により、セキュリティの問題が解決されます。詳細については、Knowledge Center の記事 CTX559370 を参照してください。

累積更新プログラム 2 (CU2)

リリース日: December 08, 2022

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 2 (CU2) では、Linux Virtual Delivery Agent 2203 LTSR Cumulative Update 1 リリースと比較して、4つの修正プログラムが追加されています。

Linux Virtual Delivery Agent 2203 LTSR 累積更新プログラム 1 (CU1)

Linux Virtual Delivery Agent（初回リリース）(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/whats-new/2203-ltsr-initial-release.html)

このリリースの既知の問題

非推奨と削除

Citrix製品のサブスクリプション特典の対象日

2203 LTSR CU2で修正された問題

Linux Virtual Delivery Agent 2203 LTSR CU1以降で、以下の問題が修正されました。

- Desktop Viewerからログオフしてデスクトップセッションからログオフしようとすると、永久に続くグレースクリーンになることがあります。この問題は、NFSストレージフォルダーで**AutoFS**サービスが有効になっている場合に発生します。
- デフォルトシェルが/bin/cshに設定されている場合、Linux VDAを使用して公開アプリケーションを開こうとすると失敗することがあります。
- Linuxターゲットデバイスが、ディスク外への読み取りまたは書き込みの試行を示すエラーメッセージが表示されて起動に失敗することがあります。
- Federated Authentication Service (FAS)が使用されている場合、ユーザーは「**無効なログイン**」メッセージを受け取ることがあります。

累積更新プログラム 1 (CU1)

リリース日: August 03, 2022

このリリースについて

Linux Virtual Delivery Agent 2203 LTSR累積更新プログラム1 (CU1) は、Linux VDA 2203 LTSRの初回リリースと比較して、6つの修正(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/whats-new/cumulative-update-1/fixed-issues.html)を追加します。

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix製品のSubscription Advantage資格日

2203 LTSR CU1 で修正された問題

Linux Virtual Delivery Agent 2203 LTSR の初回リリース以降、以下の問題が修正されました。

- Ubuntuを実行しているLinux VDAセッションからログオフして再度ログオンすると、Gnomeデスクトップ拡張機能が保持されない場合があります。
- Linux VDAセッションを起動しようとすると、セッションがグレースクリーンで予期せず終了する場合があります。
- NSAP仮想チャネルが有効になっている場合、Linux VDAでパフォーマンスの問題が発生し、動作が遅くなったり使用できなくなったりする場合があります。
- Citrix Workspace™ app for HTML5を介してLinux VDAからローカルファイルシステムにダウンロードされたファイルの名前が変更される場合があります。
- Machine Creation Services™ (MCS) を使用して作成されたLinux VDAを再起動すると、**winbind**サービスが停止する場合があります。
- この修正により、右クリックのポップアップメニューが表示される際の著しい遅延が解消されます。

このリリースについて

新機能

2203 LTSR の新機能

Linux VDAバージョン2203には、以下の新機能と機能強化が含まれています。

Rendezvous V2の完全サポート

Citrix Gatewayサービスを使用する場合、Rendezvousプロトコルにより、トラフィックはCitrix Cloud™ Connectorをバイパスし、Citrix Cloudコントロールプレーンに直接かつ安全に接続できます。

考慮すべきトラフィックには、1) VDA登録とセッションブローカリングのための制御トラフィック、2) HDX™トラフィックの2種類があります。

Rendezvous V1では、HDXトラフィックはCloud Connectorをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックをプロキシするためにCloud Connectorを必要とします。

標準のADドメイン参加マシンと非ドメイン参加マシンは、シングルセッションおよびマルチセッションLinux VDAでRendezvous V2を使用するためにサポートされています。非ドメイン参加マシ

ンでは、Rendezvous V2により、HDXトラフィックと制御トラフィックの両方がCloud Connectorをバイパスできます。

詳しくは、「Rendezvous V2」を参照してください。

簡易インストールGUIが利用可能に

以前は、簡易インストールを使用するためにコマンドラインインターフェイスしか操作できませんでした。このリリースでは、以下の操作をガイドする簡易インストールGUIが提供されます。

- システム環境の確認
- 依存関係のインストール
- VDAを特定のドメインに参加させる
- ランタイム環境の構成

簡易インストールGUIを使用するには、VDAのデスクトップ環境で`/opt/Citrix/VDA/bin/easyinstall`コマンドを実行します。詳しくは、「簡易インストールを使用したクイックインストール（推奨）」の記事の「**対話モード**」セクションを参照してください。

HDXグラフィックの改善

Thinwireは、特定のタスクを並列化することで、全体的なCPU消費量がわずかに増加するオーバーヘッドを伴いながら、1秒あたりのフレーム数（FPS）を向上させることができるようになりました。この機能はデフォルトで無効になっています。詳しくは、「グラフィックの構成」の記事の「並列処理」セクションを参照してください。

以前のリリースでの新機能

1912 LTSR以降2201 CRまでにリリースされたCRに含まれる新機能については、新機能の履歴を参照してください。

2203 LTSR で修正された問題

Linux Virtual Delivery Agent 2201 以降で、以下の問題が修正されました:

- Remote X11 が有効になっており、X11 ディスプレイ番号に対応する TCP ポートが使用中の場合、セッションを起動できません。

新機能の履歴

この記事では、1912 LTSR以降2201 CRまでにリリースされたCRに含まれる新機能に関する情報を提供します。

2201の新機能

Linux VDAバージョン2201には、以下の新機能と機能強化が含まれています。

SUSE 15.3およびSUSE 15.2でのMATEデスクトップのサポート

SUSE 15.3およびSUSE 15.2セッションのデフォルトデスクトップとしてMATEを指定できるようになりました。デスクトップ環境の詳細については、インストール記事を参照してください。

セッションシャドウ機能がすべてのサポート対象ディストリビューションに拡張

セッションシャドウ機能を使用すると、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。以前はRHEL 7.xおよびUbuntu 16.04でのみ利用可能でしたが、この機能はLinux VDAがサポートするすべてのディストリビューションに拡張されました。 詳細については、「セッションのシャドウ」を参照してください。

ドメインに参加していないVDAで指定された属性を持つローカルユーザーを作成する

ドメインに参加していないVDAでホストされているセッションを開くと、VDAはデフォルト属性を持つローカルユーザーを自動的に作成します。VDAは、Citrix Workspaceアプリにログオンするために使用したユーザー名に基づいてローカルユーザーを作成します。ユーザーのユーザー識別子（UID）、グループID（GID）、ホームディレクトリ、ログインシェルなどのユーザー属性を指定できるようになりました。詳細については、「ドメインに参加していないVDA」を参照してください。

Transport Layer Security (TLS) 1.3のサポート

Linux VDAはTLS 1.3をサポートするようになりました。/opt/Citrix/VDA/sbinディレクトリにある `enable_vdassl.sh` スクリプトを使用すると、TLSを有効にし、最小TLSバージョンを設定できます。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。

> **注:**
>
> このリリース以降、Linux VDAをインストールする前に.NET Runtime 6.0をインストールする必要があります。

2112の新機能

Linux VDAバージョン2112には、以下の新機能と機能強化が含まれています。

Amazon Linux 2のサポート（実験的）

実験的機能として、Amazon Linux 2 をサポート対象のディストリビューションとして追加しました。詳しくは、「システム要件」を参照してください。

HDX™ 画面共有の完全サポート

以前は実験的機能として提供されていた HDX 画面共有が、完全にサポートされるようになりました。仮想デスクトップの画面を、他の仮想デスクトップのセッションユーザーと共有できます。画面共有機能はデフォルトで無効になっています。詳しくは、「HDX 画面共有」を参照してください。

HTTPプロキシを介したRendezvous接続

Linux VDA は、ICA のトランスポートプロトコルとして TCP を使用する HTTP プロキシを介した Rendezvous 接続の確立をサポートするようになりました。詳しくは、「Rendezvous プロトコル」を参照してください。

Rendezvousプロキシ認証

ドメイン参加済みの Linux VDA は、Rendezvous 接続のためにプロキシサーバーに対して認証できるようになりました。詳しくは、「Rendezvous プロトコル」を参照してください。

アダプティブオーディオのサポート

アダプティブオーディオはデフォルトで有効になっています。アダプティブオーディオを使用すると、VDA でオーディオ品質ポリシーを手動で構成する必要がありません。アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、プレミアムなオーディオエクスペリエンスを提供します。詳しくは、「アダプティブオーディオ」を参照してください。

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングでは、以下のオペレーティングシステムがサポートされるようになりました。

- SUSE Linux エンタープライズ サーバー 15 SP2 (SUSE 15.2)
- SUSE Linux エンタープライズ サーバー 15 SP3 (SUSE 15.3)

詳しくは、Citrix Provisioning 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。

キーボードレイアウト同期の機能強化

WFCLIENT.ini ファイルで **KeyboardLayout** を **Server Default** に設定すると、セッションに再接続した後も以前に選択したキーボードレイアウトが保持されます。

キーボード入力の機能強化

いくつかの問題に対処し、より多くの特殊文字とショートカットのサポートを追加しました。

2110の新機能

Linux VDAバージョン2110には、以下の新機能と機能強化が含まれています。

SUSE 15.3およびSUSE 15.2のサポート

SUSE Linux Enterprise Server 15 SP3 (SUSE 15.3) およびSUSE Linux Enterprise Server 15 SP2 (SUSE 15.2) をサポート対象のディストリビューションとして追加しました。詳しくは、「システム要件」を参照してください。

SUSE 15.3、SUSE 15.2、およびSUSE 12.5でのHDX Webカメラビデオ圧縮のサポート

SUSE 15.3、SUSE 15.2、およびSUSE 12.5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 Webカメラビデオ圧縮を使用してWebカメラを使用できるようになりました。詳しくは、「HDX Webカメラビデオ圧縮」を参照してください。

PowerBroker Identity Services (PBIS) を使用するMCSで作成されたVMのFASサポート

PBISを使用してWindowsドメインに参加したMCSで作成されたLinux VDAにログオンするユーザーを認証するために、FASを使用できるようになりました。詳しくは、「フェデレーション認証サービスを構成する」および「MCSで作成されたVMでFASを有効にする」を参照してください。

ctxfascfg.shに2つの環境変数が追加されました

ctxfascfg.shに以下の変数を追加しました。

- CTX_FAS_KDC_HOSTNAME：PBISを選択した場合のKey Distribution Center (KDC) のホスト名を指定します。
- CTX_FAS_PKINIT_KDC_HOSTNAME：PKINIT KDCホスト名を指定します。特に指定がない限り、CTX_FAS_KDC_HOSTNAMEと同じです。

詳しくは、「ctxfascfg.shを実行する」を参照してください。

Citrix Workspace™アプリへのログオンに使用した資格情報とは異なる資格情報でのセッション起動

Citrix Workspaceアプリにログオンした後、Citrix Virtual Apps and Desktops™セッションを起動するために異なる資格情報を使用できます。この機能を有効にするには、Linux VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t
"REG_DWORD" -v "fPromptForDifferentUser" -d "0x00000001" --force
```

この機能を有効にすると、セッションの起動時に資格情報の入力を求められます。この機能は、次のディストリビューションでサポートされています。

- RHEL 8.4 / CentOS 8 (2105)
- RHEL 7.9 / CentOS 7.9
- ウブントゥ 16.04
- デビアン 10.7
- SUSE 15.3
- SUSE 15.2

2109の新機能

Linux VDAバージョン2109には、次の新機能と機能強化が含まれています。

HDX画面共有のサポート（実験的）

Linux VDAでは、仮想デスクトップの画面を他の仮想デスクトップのセッションユーザーと共有できるようになりました。画面共有機能はデフォルトで無効になっています。詳しくは、HDX画面共有を参照してください。

Linux VDA用シトリックス仮想チャネルソフトウェア開発キット (SDK) (実験的)

Linux VDA用Citrix Virtual Channel SDKが利用可能になりました。Virtual Channel SDKを使用すると、VDAで実行するサーバー側アプリケーションを作成できます。詳しくは、Linux VDA用Citrix Virtual Channel SDKドキュメントを参照してください。

Linux VDA用Citrix Virtual Channel SDKは、Citrix Virtual Apps and Desktopsダウンロードページからダウンロードできます。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Linux VDAのダウンロードを選択します。

PBISを使用するVDAのFASサポート

PowerBroker Identity Services (PBIS) を使用してWindowsドメインに参加したLinux VDAにログオンするユーザーを認証するために、FASを使用できるようになりました。ctxfascfg.shの実行中にPBISを選択する場合は、スクリプトの指示に従ってKey Distribution Center (KDC) のホスト名を入力します。詳しくは、「フェデレーション認証サービスを構成する」を参照してください。

RHEL 8.4、RHEL 7.9、およびSUSE 12.5のLinuxストリーミングサポート

Citrix ProvisioningのLinuxストリーミング機能を使用すると、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。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。

MATEデスクトップのキーボードレイアウト同期サポート

キーボードレイアウト同期サポートをMATEデスクトップに拡張しました。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と、MATEデスクトップ環境をインストールしたVDAのレイアウトもそれに合わせて変更されます。詳しくは、「動的キーボードレイアウト同期」を参照してください。

セッションデータクエリユーティリティ

各Linux VDAのセッションデータを照会するために使用できるユーティリティ (`ctxsdcutil`) を提供するようになりました。VDAでホストされているすべてのセッションまたは特定のセッションの次のデータを照会するには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]` コマンドを実行します。`[-c]` 引数は、1秒ごとにデータを照会することを意味します。

- **入力セッション帯域幅**
- **出力セッション帯域幅**
- **出力セッション回線速度**
- **待機時間 - 最終記録**
- **ラウンドトリップ時間**
- **出力ThinWire帯域幅**
- **出力オーディオ帯域幅**
- **出力プリンター帯域幅**
- **入力ドライブ帯域幅**
- **出力ドライブ帯域幅**

新しいLinuxセッションメトリックが利用可能になりました

Linuxセッションのパフォーマンスを追跡、集計、視覚化するために、以下のメトリックが追加されました。

- ログオン期間

  ログオン期間は、ユーザーがCitrix Workspaceアプリから接続してからセッションが使用可能になるまでのログオンプロセスを測定するものです。このメトリックは、Citrix DaaSのMonitorタブで確認できます。Monitorは、Citrix Virtual Apps and DesktopsのCurrent ReleaseおよびLTSR展開を監視およびトラブルシューティングするためのDirectorコンソールとして利用できます。

- セッションの自動再接続数

  Citrix DirectorとMonitorの両方の**トレンド**ビューで、Linuxセッションにおける自動再接続の数を表示できるようになりました。自動再接続は、**セッションの信頼性**または**クライアントの自動再接続**ポリシーが有効な場合に有効になります。セッションの再接続の詳細については、「セッション」を参照してください。ポリシーの詳細については、「クライアントの自動再接続ポリシー設定」および「セッションの信頼性ポリシー設定」を参照してください。

詳細については、「Linux VMとLinuxセッションの監視」を参照してください。

2107の新機能

Linux VDAバージョン2107には、以下の新機能と機能強化が含まれています。

MCSで作成されたRHEL 7/CentOS 7およびRHEL 8/CentOS 8マシンに対するCentrifyのサポート

MCSで作成されたRHEL 7/CentOS 7およびRHEL 8/CentOS 8マシンをWindowsドメインに参加させるためのCentrify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。

USBリダイレクト用のカーネルモジュール再構築の簡素化

現在、Linux VDAソースコードファイルcitrix-linux-vda-sources.zipに`ctxusbcfg.sh`を提供しています。このスクリプトは、Dynamic Kernel Module Support (DKMS) オプションの有無にかかわらず、VHCIカーネルモジュール (`usb-vhci-hcd.ko`および`usb-vhci-iocif.ko`) のビルドを簡素化するのに役立ちます。DKMSは、カーネルがアップグレードされたときにVHCIカーネルモジュールを自動的に再構築するのに役立ちます。詳しくは、「USBリダイレクトの構成」を参照してください。

RHEL 8.4およびCentOS 8 (2105) のサポート

サポート対象のディストリビューションとして、RHEL 8.4およびCentOS 8 (2105) を追加しました。詳しくは、「システム要件」を参照してください。

Debianでのスマートカードのサポート

クライアントデバイスに接続されたスマートカードを使用して、Linux仮想デスクトップセッションへのログオン時に認証を行うことができます。このリリースでは、Debian 10.7セッションでスマートカードを使用することもできます。詳しくは、「スマートカードのサポート」を参照してください。

2106の新機能

Linux VDAバージョン2106には、次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。AzureのVMに対する管理者権限がない場合に役立ちます。詳しくは、「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。詳しくは、「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。詳しくは、「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 Web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 Webカメラビデオ圧縮を使用してWebカメラを使用できるようになりました。HDX Webカメラビデオ圧縮は、**最適化された**Webカメラモードとも呼ばれます。このタイプのWeb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 Web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 Webカメラビデオ圧縮を使用することをお勧めします。詳しくは、「HDX Web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追加されました。このスライダーは、画質と対話性の適切なバランスを見つけるのに役立ちます。詳しくは、「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、Secure Browserサービスで設定したユーザー固有のポリシーを、CGPセッション再接続中に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
  - デスクトップからのファイルのダウンロード
  - デスクトップへのファイルのアップロード

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提供されていたRHEL 8.3およびUbuntu 18.04のLinuxストリーミングが、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。

2104の新機能

Linux VDAバージョン2104には、以下の新機能と機能強化が含まれています。

Citrix DaaS™におけるドメイン非参加Linux VDAのサポート

MCSを使用して、Citrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）でドメイン非参加Linux VDAを作成できるようになりました。詳しくは、「ドメイン非参加」を参照してください。

OpenJDK 11 が必須に

Linux VDA では、OpenJDK 11 が必須になりました。Linux VDA がサポートする Linux ディストリビューションのうち、Ubuntu 16.04 のみ、OpenJDK 11 を手動でインストールする必要があります。その他のサポートされているディストリビューションでは、Linux VDA のインストール時に依存関係として OpenJDK 11 が自動的にインストールされます。

XDPing の変更点

`ctxsetup.sh` を実行しても、`XDPing` はインストールされなくなりました。`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。詳しくは、「XDPing」を参照してください。

Ubuntu のスマートカードサポート

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。このリリースでは、Ubuntu 20.04、Ubuntu 18.04、および Ubuntu 16.04 セッションでスマートカードパススルー認証を使用できるようになりました。詳しくは、「スマートカードを使用したパススルー認証」を参照してください。

MATE デスクトップのサポート

CentOS、RHEL、Ubuntu、Debian で軽量の MATE デスクトップのサポートを追加しました。ctxinstall.sh、ctxsetup.sh、deploymcs.sh スクリプトで利用できる新しい変数を使用して、MATE または GNOME デスクトップを指定できるようになりました。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、この変数の値を mate に設定する必要があります。詳しくは、「簡単なインストール」および「RHEL/CentOS 用 Linux Virtual Delivery Agent のインストール」などの手動インストールに関する記事を参照してください。

RHEL 8、CentOS 8、SUSE 12.5 の PBIS サポート

RHEL 8、CentOS 8、SUSE 12.5 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。

2103 の新機能

Linux VDA のバージョン 2103 には、次の新機能と機能強化が含まれています。

Debian 10.7 および CentOS 8.3 のサポート

サポート対象のディストリビューションとして Debian 10.7 と CentOS 8.3 を追加しました。詳しくは、「システム要件」、「Debian 用 Linux Virtual Delivery Agent のインストール」、および「RHEL/CentOS 用 Linux Virtual Delivery Agent のインストール」を参照してください。

SUSE マシンを Windows ドメインに参加させるための SSSD サポート

SUSEマシンをWindowsドメインに参加させるためのSSSDサポートを追加しました。

ドメイン参加済みおよび非ドメイン参加済みの両方のユースケースに対応する単一の Linux VDAイメージ

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を作成するための単一イメージを提供するようになりました。この機能により、イメージの準備とメンテナンスが簡素化されます。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。

ファイル転送の機能強化

プログレスバーの再設計、ダウンロード速度の高速化、およびいくつかのバグの解決により、ファイル転送機能を強化しました。また、一度に転送できるファイルの数を10個から100個に増やしました。詳細については、「ファイル転送」を参照してください。

Citrix Studioでのセッション接続タイマー構成のサポート

以前は、アイドルセッションと切断されたセッションの制限時間は、各VDAで個別に`/opt/Citrix/VDA/bin/ctxcfg`を介してのみ構成できました。このリリースでは、Citrix Studioでセッション接続タイマーを構成するための次のポリシーが追加されています。

- **セッションアイドルタイマー**: アイドル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**: アイドルセッションの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がユーザー入力を受信しなかった場合、セッションは切断されます。
- **切断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。

ポリシー設定を更新する際は、展開全体で一貫性があることを確認してください。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。

アイドルセッションの時間制限が期限切れになると、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、ユーザー入力を提供してアイドルタイマーをリセットしてください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2103/media/warning-message-idle-session-timeout.png)

Citrix DirectorでLinux VMおよびLinuxセッションの新しいメトリックが利用可能になりました

このリリースでは、Citrix DirectorにLinux VMおよびLinuxセッションの新しいメトリックが追加されています。

各Linux VMの新しいメトリック:

- 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

各Linuxセッションの新しいメトリック:

- アイドル時間

詳しくは、「Citrix DirectorでLinux VMとLinuxセッションを監視する」(/ja-jp/linux-virtual-delivery-agent/2103/configuration/monitor-linux-sessions-in-director.html)を参照してください。

Linux VDAのFAS機能強化

ctxfascfg.shスクリプトを実行する際に、より詳細なログ出力を提供し、ルート証明書とすべての中間証明書の両方を含むパスを指定できるようになりました。構成情報については、「フェデレーション認証サービスを構成する」(/ja-jp/linux-virtual-delivery-agent/2103/configuration/federated-authentication-service.html)を参照してください。

RHEL 8.3およびUbuntu 18.04.5のLinuxストリーミングサポート - 実験的機能

Citrix Provisioningの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、「Linuxストリーミング」(/ja-jp/linux-virtual-delivery-agent/2103/configuration/linux-streaming.html)を参照してください。

2012の新機能

Linux VDAのバージョン2012には、以下の新機能と機能強化が含まれています。

エンライトンド データ トランスポート (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。詳しくは、「EDT MTU検出」(/ja-jp/linux-virtual-delivery-agent/current-release/configure/session/adaptive-transport.html#edt-mtu-discovery)を参照してください。

LDAPチャネルバインディングのサポート

以前は、Active Directoryでチャネルバインディングが有効になっている場合、Linux VDAはDelivery Controllerに登録できませんでした。このLinux VDAリリースでは、Active Directoryドメインコントローラーとの通信のセキュリティを強化するためにチャネルバインディングをサポートしています。Linux VDAでチャネルバインディングを有効にする前に、Python3仮想環境を作成してください。詳細については、「Python3仮想環境の作成」および「LDAPSの構成」を参照してください。

Linux VDAに統合されたXDPing

Linux **XDPing**ツールは、Linux VDA環境における一般的な構成の問題を自動的にチェックするコマンドラインベースのアプリケーションです。以前は、CTX202015ページから**XDPing**をダウンロードし、Linux VDAとは別にツールをインストールする必要がありました。このリリース以降、`XDPing`はLinux VDAに統合されており、Linux VDAのインストール時に自動的にインストールできます。詳細については、「**XDPing**」を参照してください。

RHEL 7.9、RHEL 8.3のサポート

サポート対象のディストリビューションとしてRHEL 7.9およびRHEL 8.3を追加しました。詳細については、「システム要件」および「RHEL/CentOS用Linux Virtual Delivery Agentのインストール」を参照してください。

Ubuntu 20.04のサポート

以前は実験的な機能として提供されていましたが、Ubuntu 20.04はこのリリースで完全にサポートされています。詳細については、「システム要件」および「Ubuntu用Linux Virtual Delivery Agentのインストール」を参照してください。

Rendezvousプロトコルのサポート

Linux VDAはRendezvousプロトコルをサポートするようになりました。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セッションは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ます。詳細については、「Rendezvousプロトコル」を参照してください。

グーグル クラウド プラットフォーム (GCP) での マシン クリエーション サービス (MCS) のサポート

MCSを使用してGCP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、「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。

LinuxマシンでWake on LANが利用可能に

統合されたWake on LAN機能がLinuxマシンで利用可能になりました。詳細については、「Remote PC Access」を参照してください。

> **注:**
>
> このリリース以降、Linux VDAをインストールする前に.NET Core Runtime 3.1をインストールする必要があります。

2009の新機能

Linux VDAバージョン2009には、以下の新機能と機能強化が含まれています。

RHEL 8.2のサポート

Citrixは、RHEL 8.2をサポート対象のディストリビューションとして追加しました。

(実験的) Ubuntu 20.04のサポート

実験的な機能として、Ubuntu 20.04にLinux VDAを配布できます。

ブラウザーコンテンツリダイレクトのサポート

このリリースから、Linux VDAはGoogle Chromeでのブラウザーコンテンツリダイレクトをサポートします。ブラウザーコンテンツリダイレクトは、許可リストにあるWeb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。詳しくは、「ブラウザーコンテンツリダイレクト」を参照してください。

セッションウォーターマークのサポート

Linux VDAはセッションウォーターマークをサポートするようになりました。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す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。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

を阻止し、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。詳しくは、「テキストベースのセッションウォーターマーク」を参照してください。

Citrix Virtual Apps™ and Desktops Standard for Azureにおけるドメインに参加していないLinux VDAのサポート

Citrix Virtual Apps and Desktops Standard for Azureでドメインに参加していない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るようになりました。詳しくは、「Citrix DaaS Standard for AzureでLinux VDAを作成する」および「Citrix Virtual Apps and Desktops Standard for Azure」を参照してください。

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設定するためのポリシーが利用可能に

以前は、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機能を有効または無効にするには、各ターゲットVDAでレジストリを編集する必要がありました。このリリースから、**クライアントキーボードレイアウト同期とIMEの改善**ポリシーを使用してこれらの機能を設定できます。この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と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。詳しくは、「動的キーボードレイアウト同期」および「クライアントIMEユーザーインターフェイス同期」を参照してください。

VDAから追加のデスクトップセッションを開くときに表示されるプロンプト

Citrix HDX、RDP、および直接コンソール接続を介してVDAからデスクトップセッションを開くことができます。ただし、このリリース以前は、同じUbuntu VDAから追加のHDXデスクトップセッションを開こうとすると失敗していました。この問題は、GNOMEデスクトップがインストールされているUbuntu VDAで発生していました。機能強化として、以下のプロンプトで現在のデスクトップセッションから強制的にログオフすることで、新しいデスクトップセッションを続行できるようになりました。このプロンプトは、VDAがインストールされているLinuxプラットフォームに関係なく、VDAから追加のデスクトップセッションを開くたびに表示されます。

注:

- gdm3 を介して VDA にログインする際にプロンプトで **YES** をクリックすると、コンソールデスクトップの画面が黒くなり、応答しないように見えます。Ctrl+Alt+F1 を使用して VDA ログインウィンドウに戻ることができます。

- プロンプトは30秒間表示されます。その後、回答は自動的に **NO** に設定されます。

プロンプトの画像(/en-us/linux-virtual-delivery-agent/current-release/media/prompt.png)

2006の新機能

Linux VDA バージョン2006には、以下の新機能と機能強化が含まれています。

言語サポート

Linux VDA がイタリア語で利用可能になりました。詳細については、「Citrix 製品の言語サポート」を参照してください。

SUSE 12 SP5 のサポート

このリリースから、Citrix は SUSE 12 SP5 をサポート対象のディストリビューションとして追加しました。

クライアントドライブマッピングの機能強化

このリリースでは、クライアントドライブマッピングが強化され、特にネットワーク遅延が大きい状況で、Linux VDA とクライアントデバイス間のファイル転送が高速化されます。

セッションとクライアント間のファイルコピー＆ペーストのサポート

Citrix Virtual Apps and Desktops 2006 および Windows 用 Citrix Workspace app 1903 以降を使用すると、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＆ペーストできるようになりました。コピー＆ペースト機能は、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実行できます。詳細については、「ファイルのコピーと貼り付け」を参照してください。

注:

このリリースから、Linux VDA をインストールする前に .NET Core Runtime 2.1 をインストールする必要があります。

2003の新機能

Linux VDA バージョン2003には、以下の新機能と機能強化が含まれています。

RHEL 8のサポート

このリリースから、CitrixはRHEL 8をサポート対象のディストリビューションとして追加しました。

Citrix Telemetry Serviceとの統合

Citrix Telemetry Service (**ctxtelemetry**) はLinux VDAソフトウェアと統合されています。Delivery ControllerからCitrix Scoutを実行して、Linux VDAに関するログを収集できるようになりました。詳しくは、「Citrix Scout」を参照してください。

軽量NSAP仮想チャネルのサポート

Linux VDAはNSAP仮想チャネルのサポートを追加します。すべてのHDX InsightデータはNSAP仮想チャネルからのみ取得され、非圧縮で送信されるため、スケーラビリティとパフォーマンスが向上します。詳しくは、「HDX Insight」を参照してください。

自動ダウンロード

Citrixは、ファイル転送の機能強化として自動ダウンロードを追加しました。これは、HTML5およびChrome用のCitrix Workspaceアプリで利用できます。この機能強化により、VDA上に **Save to My Device** ディレクトリが提供されます。**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。詳しくは、「File transfer」を参照してください。

NVIDIA Tesla T4のサポート

NVIDIA Tesla T4がサポートされています。

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。

- Cloud ConnectorまたはDelivery Controllerを再起動すると、Linux VDAが登録解除される可能性があります。
- Kerberos で**RC4_HMAC_MD5**暗号化タイプが許可されている場合、Linux VDAがControllerに登録できないことがあり、次のエラーメッセージが表示されます。

  エラー: GSS-APIレベルで指定されていない障害 (メカニズムレベル: HMAC付きRC4暗号化タイプはサポートされていません/有効になっていません)

  この問題に対処するには、Active Directoryドメイン全体で（*または特定のOUで*）**RC4_HMAC_MD5**を無効にするか、Linux VDAで弱い暗号化タイプを許可します。その後、**klist -li 0x3e4 purge**コマンドを使用してControllerおよびCitrix Cloud Connector™上のキャッシュされたKerberosチケットをクリアし、Linux VDAを再起動します。

  アクティブディレクトリドメイン全体で**RC4_HMAC_MD5**を無効にするには、次の手順を実行します。

  a. グループポリシー管理コンソールを開きます。
  b. ターゲットドメインを見つけて、**Default Domain Policy**を選択します。
  c. **Default Domain Policy**を右クリックし、**Edit**を選択します。グループポリシー管理エディターが開きます。
  d. **コンピューターの構成 > ポリシー > Windows の設定 > セキュリティの設定 > ローカル ポリシー > セキュリティ オプション**を選択します。
  e. **ネットワークセキュリティ: Kerberosで許可される暗号化の種類を構成する**をダブルクリックします。
  f. **DES_CBC_CRC**、**DES_CBC_MD5**、および**RC4_HMAC_MD5**のチェックボックスをオフにし、**AES128_HMAC_SHA1**、**AES256_HMAC_SHA1**、および**将来の暗号化タイプ**を選択します。

  Linux VDAで弱い暗号化タイプを許可するには、次の手順を実行します。

> **注:**
>
> 弱い暗号化タイプは、展開を攻撃に対して脆弱にします。

a. Linux VDA 上にある /etc/krb5.conf ファイルを開いてください。

b. ****セクションの下に次の設定を追加します。

```
allow_weak_crypto= TRUE
```

- Linux VDA は、暗号化に SecureICA 1.0 をサポートしていません。Linux VDA で SecureICA 1.0 を有効にすると、セッションの起動に失敗します。
- シームレスではない公開アプリケーションが、起動直後に終了する場合があります。この問題は、Mutter を mutter-3.28.3-4 より新しいバージョンにアップグレードした後に発生します。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 SUSE 12.5 で発生します。その理由は、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 のデフォルト設定では、20 秒間操作がないとサウンドサーバープログラムが終了します。PulseAudio が終了すると、オーディオは機能しません。この問題を回避するには、/etc/pulse/daemon.conf ファイルで exit-idle-time=-1 を設定してください。
- 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。
- 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この問題を修正できます。
- 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。
- 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 印刷リダイレクトによって作成されたプリンターは、セッションからログアウトした後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。
- Citrix Workspace app for Android の CAPS LOCK 状態は、セッションローミング中に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 状態が失われる可能性があります。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 に接続する場合、ALT を含む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 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 への接続を試みると、Broker は VDA への戻りセキュリティコンテキストを確立できない場合があります。一般的な症状は、VDA 登録の失敗です。

  VDAサービスチケットの有効期限が切れて更新されると、この問題はいずれ解決される可能性があります。しかし、サービスチケットは有効期間が長いため、解決には長い時間がかかることがあります。

  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。これにより、他のVDAや、場合によっては他のサービスに対するサービスチケットも削除されます。ただし、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 2203 (PDFダウンロード)

このLinux VDAのリリースには、ドキュメントに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

本記事のお知らせは、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通知し、お客様が適切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判断するために、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	非推奨発表	削除時期
RHEL 8.1、RHEL 8.3のサポート	2203	2206
RHEL 7.8、CentOS 7.8のサポート	2203	2204
CentOS 8.xのサポート	2110	2201
SUSE 12.5のサポート	2109	2204
Ubuntu 16.04のサポート	2109	2203
RHEL 7.7、CentOS 7.7のサポート	2006	2009
SUSE 12.3のサポート	2006	2006
RHEL 6.10、CentOS 6.10のサポート	2003	2003

項目	非推奨発表	削除時期
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 VDA の Current Release は、Citrix Virtual Apps and Desktops に準拠しています。また、ライフサイクルが終了していない以前のバージョンの Citrix Virtual Apps and Desktops とも下位互換性があります。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。どの Delivery Controller™ ファームでも、Windows デスクトップと Linux デスクトップの両方を仲介できます。

ここで説明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で説明されています。

Long Term Service (LTSR) 環境での Current Release (CR) の使用方法やその他の FAQ については、「Knowledge Center の記事」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、Xorg のバージョン、およびデスクトップ環境

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、Xorg のバージョン、およびデスクトップ環境のマトリックスについては、次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 のバージョン	サポートされているデスクトップ
アマゾン Linux 2	1.20	MATE、GNOME
デビアン 10.9	1.20	MATE、GNOME
ロッキー Linux 8.10/8.8/8.6	1.20	メイト、グノーム、グノームクラシック、ケーディーイー
RHEL 8.10、RHEL 8.8、RHEL 8.6、RHEL 8.4、RHEL 8.3、RHEL 8.2、RHEL 8.1	`>= 1.20.8`	MATE、GNOME、KDE

LINUXディストリビューション	XORG のバージョン	サポートされているデスクトップ
RHEL 7.9、RHEL 7.8、CentOS 7.9、CentOS 7.8	1.20	MATE、GNOME、KDE
SUSE 15.3、SUSE 15.2	1.20	MATE、GNOME
SUSE 12.5	1.19	GNOME、KDE
ウブントゥ 20.04	1.20	MATE, GNOME
ウブントゥ 18.04	1.19	MATE, GNOME

注:

- OSベンダーからのサポートが期限切れになった場合、Citrixは問題の修正能力が制限される可能性があります。非推奨または削除されたプラットフォームについては、非推奨を参照してください。
- 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。セッションで使用するGNOMEまたはMATEデスクトップ環境は、ctxinstall.shまたはctxsetup.shスクリプトを介して指定できます。
- Ubuntuで `HWE kernel` または `HWE Xorg` を使用しないでください。
- ユーザー名の形式は、現在のディスプレイマネージャーの `systemd` 構文規則に準拠している必要があります。 `systemd` ユーザー名構文の詳細については、ユーザー/グループ名の構文を参照してください。

ネットワークポート

ネットワークポートに関する包括的な情報については、Citrixテクノロジーで使用される通信ポートを参照してください。

サポートされてい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 vSphere ハイパーバイザー
- マイクロソフト Hyper-V
- ニュータニックス AHV
- マイクロソフト アジュール
- 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)

注:

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。

2203リリース以降、Citrix Virtual Apps and DesktopsおよびCitrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal SubscriptionまたはHybrid Rightsライセンスが必要です。Universal SubscriptionおよびHybrid Rightsライセンスの詳細については、Transition and Trade Up (TTU) with Citrix Universal Subscriptionを参照してください。

Active Directory統合パッケージ

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

注:

Questをドメイン参加方法として使用するには、手動で行う必要があります。詳細については、Linux VDAを手動でインストールするの章にある手動インストール記事を参照してください。簡易インストール方法では、どのLinuxディストリビューションでもQuestをサポートしていません。

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
アマゾン リナックス 2	はい	はい	はい	はい	いいえ
デビアン 10.9	はい	はい	はい	はい	はい
ロッキー リナックス 8.10/8.8/8.6	はい	はい	いいえ	いいえ	はい（Quest v4.1以降）
RHEL 8.10	はい	はい	はい	はい	はい（Quest v4.1以降）
RHEL 8.8	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 8.6	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 8.4	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 8.3	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 8.2	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 8.1	はい	はい	はい	はい	はい (Quest v4.1 以降)

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
RHEL 7.9、CentOS 7.9	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 7.8、CentOS 7.8	はい	はい	はい	はい	はい（Quest v4.1以降）
SUSE 15.3	はい	はい	はい	はい	はい
SUSE 15.2	はい	はい	はい	はい	はい
SUSE 12.5	はい	はい	はい	はい	はい（Quest v4.1以降）
ウブントゥ 20.04	はい	はい	はい	はい	はい（Quest v4.1以降）
ウブントゥ 18.04	はい	はい	はい	はい	はい（Quest v4.1以降）

HDX™ 3D プロ

Citrix Virtual Apps and Desktops™ の HDX 3D Pro は、GPU アクセラレーション対応のデスクトップとアプリケーションを有効にします。

ハイパーバイザー

Linux VDA の場合、HDX 3D Pro は以下のハイパーバイザーと互換性があります。

- Citrix ハイパーバイザー
- VMware vSphere ハイパーバイザー
- ニュータニックス AHV

- マイクロソフト アジュール
- AWS
- グーグル クラウド プラットフォーム (GCP)

> **注:**
>
> 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。 Amazon Linux 2でHDX 3D Proを使用するには、NVIDIAドライバー470をインストールすることをお勧めします。

GPU

お使いのLinuxディストリビューションがサポートするNVIDIA GPUカードを確認するには、NVIDIA製品サポートマトリックスにアクセスし、**Hypervisor or Bare-Metal OS**、**Software Product Deployment**、**Hardware Supported**、**Guest OS Support**の各列を確認してください。

GPUカードには最新のvGPUドライバーをインストールしてください。現在、Linux VDAはvGPU 14までをサポートしています。詳細については、NVIDIA Virtual GPU Software Supported GPUsを参照してください。

インストール概要

このセクションでは、以下の手順について説明します。

- 簡単インストールを使用してドメイン参加済みVDAを作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-domain-joined-vdas-using-easy-install.html)
- MCSを使用してドメインに参加していないLinux VDAを作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-non-domain-joined-linux-vdas.html)
- MCSを使用してLinux VDAを作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html)
- Citrix Provisioningを使用してLinux VDAを作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-pvs.html)
- Citrix DaaS Standard for AzureでLinux VDAを作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-in-citrix-virtual-apps-desktops-standard-azure.html)
- Linux VDAを手動でインストールする(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview.html)
    - Amazon Linux 2、CentOS、およびRHELにLinux VDAを手動でインストールする(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/redhat.html)
    - SUSEにLinux VDAを手動でインストールする(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/suse.html)
    - UbuntuにLinux VDAを手動でインストールする(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/ubuntu.html)
    - DebianにLinux VDAを手動でインストールする(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/debian.html)

ドメイン参加済みVDAをイージーインストールを使用して作成する

> **重要:**

- 新規インストールの場合、迅速なインストールについてはこの記事を参照することをお勧めします。この記事では、イージーインストールを使用してLinux VDAをインストールおよび構成する方法を順を追って説明します。イージーインストールは、手動インストールよりも時間と労力を節約し、エラーが発生しにくいです。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するのに役立ちます。
- ドメインに参加していないVDAを作成するには、Machine Creation Services (MCS)を使用する必要があります。詳細については、「ドメインに参加していないLinux VDAの作成」を参照してください。
- ドメインに参加していないVDAで利用可能な機能については、「ドメインに参加していないVDA」を参照してください。

ステップ1：構成情報とLinuxマシンを準備する

イージーインストールに必要な以下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
- ドメインの完全修飾ドメイン名 (FQDN)

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ます。
- 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ます。

考慮事項

- ワークグループ名は、デフォルトでドメイン名です。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 マシンに /tmp/ctxinstall.conf ファイルを作成してください。

b. そのファイルに workgroup=\<your workgroup\> の行を追加し、変更を保存してください。

- 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。

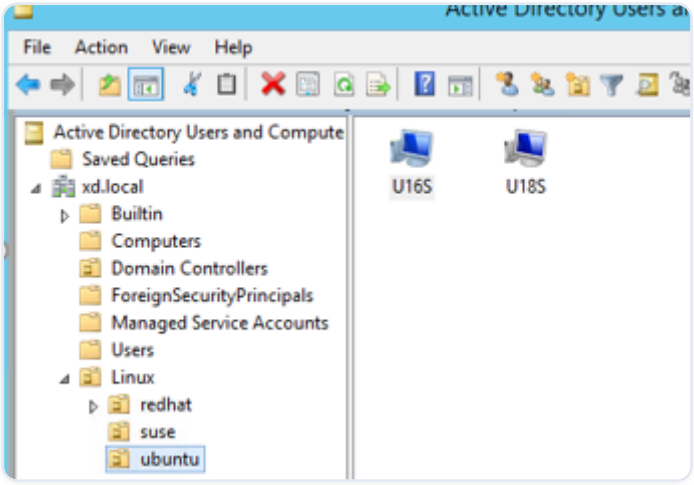
  a. ドメインコントローラーで**管理ツール**を開きます。

  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. ドメインコントローラーに特定のOUが存在することを確認します。

  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\>行を追加します。

OUの値は、ADの方法によって異なります。次の表は、前のスクリーンショットのOU名の例を示しています。組織内で他のOU名を使用することもできます。

OS	ウィンバインド	SSSD	セントリファイ	PBIS
アマゾン リナックス 2	`ou="Linux/amazon"`	`ou="Linux/amazon"`	`ou="XD.LOCAL/Linux/amazon"`	`ou="Linux/amazon"`
デビアン	`ou="Linux/debian"`	`ou="Linux/debian"`	`ou="XD.LOCAL/Linux/debian"`	`ou="Linux/debian"`
RHEL 8、ロッキー リナックス 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"`
SUSE	`ou="Linux/suse"`	`ou="Linux/suse"`	`ou="XD.LOCAL/Linux/suse"`	`ou="Linux/suse"`
ウブントゥ	`ou="Linux/ubuntu"`	`ou="Linux/ubuntu"`	`ou="XD.LOCAL/Linux/ubuntu"`	`ou="Linux/ubuntu"`

- 簡易インストールは、Linux VDA 7.16 以降、純粋な **IPv6** をサポートしています。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 リポジトリが構成されている必要があります。
  - 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。ネットワークがハイブリッド

> **IPv4** と **IPv6** の構成である場合、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。
  - 簡易インストールは、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>`

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

    SUSEの場合: 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>`
  - Centrifyパッケージをローカルディレクトリから取得します。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 サーバー上に /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\>」の行を追加します。

    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、インターネットからPBISパッケージを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。

  CentOS 7、RHEL 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`

  アマゾン リナックス 2、RHEL 8、ロッキー リナックス 8、スーゼ 15.3、スーゼ 15.2： `wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh`

  デビアン、ウブントゥ: `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特定のバージョンのPBISパッケージを取得します。これを行うには、/opt/Citrix/VDA/sbin/ctxinstall.shファイル内の「pbisDownloadPath」行を変更して、PBISパッケージのURLを指定します。

  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 (/en-us/linux-virtual-delivery-agent/2203-ltsr/media/fetching-specific-version-of-pbis-package.png)

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。使用中のハイパーバイザープラットフォームに基づいて、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は、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。

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 - 時刻同期機能は有効になっており、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維持するには、**/etc/sysctl.conf**ファイルを編集し、以下の行を追加します。

`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、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で問題が発生します。両方がシステムクロックを同期しようとします。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：.NET Runtime 6.0を前提条件としてインストールする

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 Runtime 6.0をインストールします。

.NET Runtime 6.0のインストール後、**which dotnet**コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NET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。

ステップ4：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、**コンポーネント**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。

ステップ 5: 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/CentOS/Rocky Linux ディストリビューションの場合:

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

注:

GCP でホストされている RHEL 8.x に Linux VDA をインストールした後、イーサネット接続が失われ、VM の再起動後に Linux VDA に到達できなくなることがあります。この問題を回避するには、VM を再起動する前に次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl NetworkManager restart
```

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、`deb http://deb.debian.org/debian/ oldstable main` の行を /etc/apt/sources.list ファイルに追加します。

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 6: 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 NVIDIA GRID ドライバーをインストールする必要があります。

特定のハイパーバイザーに NVIDIA GRID Virtual GPU Manager (ホストドライバー) をインストールして構成するには、次のガイドを参照してください。

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRID ゲスト VM ドライバーをインストールして構成するには、次の一般的な手順を実行します。

1. ゲスト VM がシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VM に GPU を割り当てます。
3. VM を起動します。
4. ゲストVMドライバーをVMにインストールします。

ステップ7：インストールを完了するためのランタイム環境をセットアップする

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。

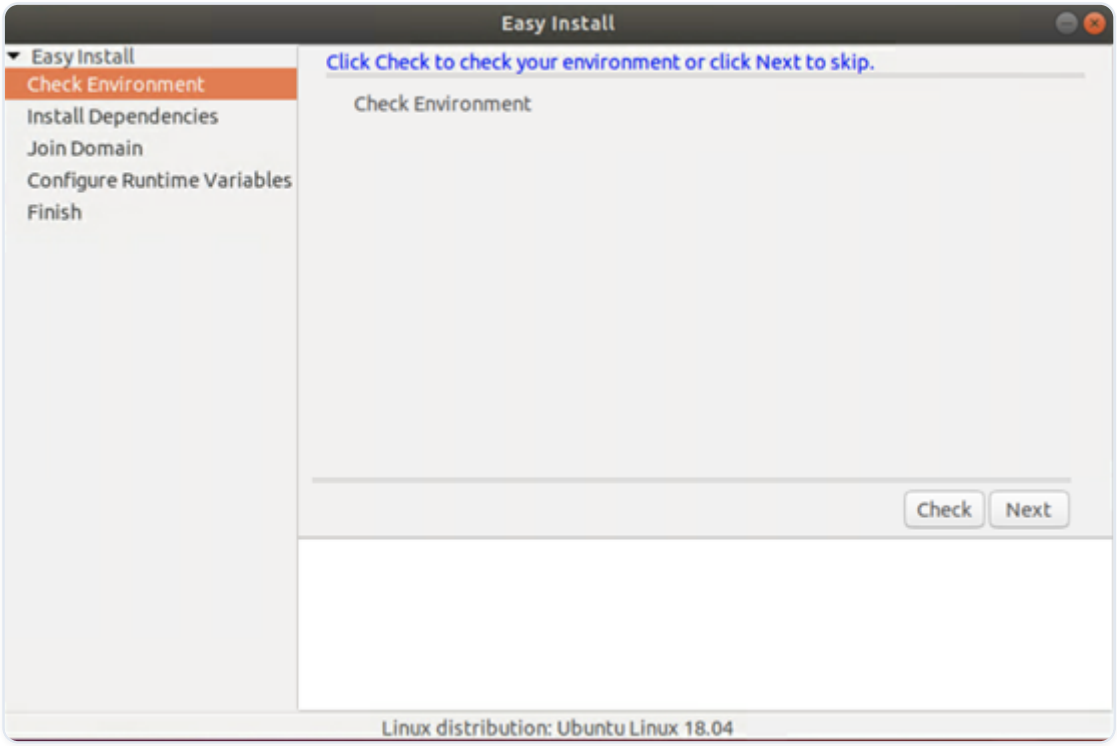
> **注:**

> ランタイム環境をセットアップする前に、OSに`en_US.UTF-8`ロケールが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`コマンドを実行します。Debianの場合、`/etc/locale.gen`ファイルを編集して`# en_US.UTF-8 UTF-8`行のコメントを解除し、`sudo locale-gen`コマンドを実行します。

> **対話モード:**

対話モードで簡易インストールを使用する方法は2つあります。

- `sudo /opt/Citrix/VDA/sbin/ctxinstall.sh`コマンドを実行し、コマンドラインインターフェイスの各プロンプトで関連するパラメーターを入力します。
- VDAのデスクトップ環境で`/opt/Citrix/VDA/bin/easyinstall`コマンドを実行し、簡易インストールGUIの指示に従います。



簡易インストールGUIは、以下の操作をガイドします。

- システム環境の確認
- 依存関係のインストール
- VDAを指定されたドメインに参加させる
- ランタイム環境の構成

サイレントモード:

サイレントモードで簡易インストールを使用するには、ctxinstall.shを実行する前に以下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に登録するために、スペースで区切られたDelivery Controllerの完全修飾ドメイン名（FQDN）の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の場合、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。
- **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 を選択した場合、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限定するには、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 ポートを持つ 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ために、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 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 Runtime 6.0 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。

  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` または `.Xclients` ファイルを作成します。Amazon Linux 2 を使用している場合は、`.Xclients` ファイルを作成します。その他のディストリビューションを使用している場合は、`.xsession` ファイルを作成します。

  b. `.xsession` または `.Xclients` 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。

     - **Amazon Linux 2、Debian、RHEL 8、SUSE 15、および Ubuntu の MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **CentOS、RHEL、SUSE 15、および SUSE 12.5 の GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **Amazon Linux 2、Ubuntu、および Debian の GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session
```

```
fi
```

c. 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** – 設定が完了したときにLinux VDAサービスを開始するかどうか。
- **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。

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 Runtime 6.0をインストールするパスのユーザー入力を求めます。

サイレントモードでは、環境変数を設定してからctxinstall.shスクリプトを実行するために、次のコマンド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ときは、-Eオプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。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 \

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
```

手順8：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、「XDPing」を参照してください。

手順9：Linux VDAを実行する

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` および `ctxhdx` サービスを停止する前に、`systemctl 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 /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
```

手順 10: 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。

手順 11: 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 2203」を参照してください。

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、簡易インストール機能の使用によって発生す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参加の失敗

ドメインに参加しようとすると、以下のような出力（画面表示についてはログを確認）を伴うエラーが発生する場合があります。

```
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 で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後に空白のデスクトップでブロックされた場合に発生します。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.



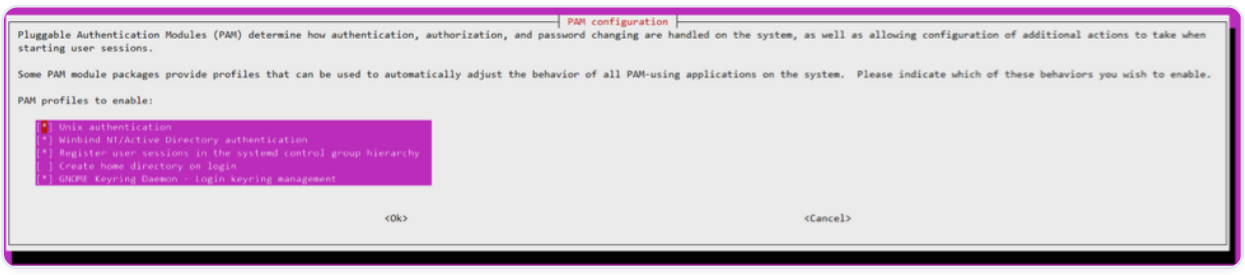
2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting normally.
```

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. 表示されたダイアログで、**Create home directory login** が選択されていることを確認します。



セッションが起動しない、またはdbusエラーですぐに終了する

/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がSSHDのホームディレクトリへのアクセスを妨げる

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。

MCS を使用してドメインに参加していない Linux VDA を作成する

この記事では、Machine Creation Services™ (MCS) を使用して、Citrix DaaS でドメインに参加していない Linux VDA を作成する方法について説明します。

> **重要:**

- ドメインに参加していない VDA は Citrix DaaS でサポートされています。
  - コントロールプレーンは Citrix DaaS 上に展開する必要があります。
  - ドメインに参加していない VDA は、パブリッククラウドまたはオンプレミスのデータセンターに展開できます。ドメインに参加していない VDA は、Citrix DaaS のコントロールプレーンによって管理されます。
  - Citrix Cloud Connector をバイパスするように Rendezvous V2 を構成できます。そうしないと、VDA をコントロールプレーンに接続するために Cloud Connector をインストールする必要があります。
- ドメインに参加していない VDA を作成するには、MCS を使用する必要があります。
  - ベアメタルサーバーは MCS でサポートされていません。
- ドメインに参加していない Linux VDA で利用できる機能は次のとおりです。
  - ドメインに参加していない VDA で指定された属性を持つローカルユーザーを作成する
  - 非SSO認証
  - ランデブー V2

(Nutanix のみ) 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。詳細については、Nutanix Support Portal 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 用 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。

オンプレミスの Delivery Controller 用 Nutanix AHV プラグイン (/en-us/linux-virtual-delivery-agent/current-release/media/nutanix-plug-in-for-cvad.png)

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

Citrix Cloud™ Connector 用の **CWA MCS AHV Plugin** を選択してインストールします。プラグインは、Citrix Cloud テナントに登録されているすべての Citrix Cloud Connector にインストールします。AHV を使用し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する

- `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、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: ホスト接続を作成する

ホストとは、リソースの場所で使用されているハイパーバイザーまたはクラウドサービスです。この手順では、DaaS がホスト上の VM と通信するために使用する情報を指定できます。詳細情報には、リソースの場所、ホストの種類、アクセス資格情報、使用するストレージ方法、およびホスト上の VM が使用できるネットワークが含まれます。

> **重要:**
>
> 接続を作成する前に、リソースの場所にあるホストリソース（ストレージとネットワーク）が利用可能である必要があります。

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS** を選択します。
3. **Manage > Full Configuration** から、左側のペインで **Hosting** を選択します。
4. アクションバーで を選択します。
5. ウィザードは次のページに沿って進みます。特定のページの内容は、選択した接続の種類によって異なります。各ページを完了したら、**Summary** ページに到達するまで **Next** を選択します。

ステップ 2a: 接続

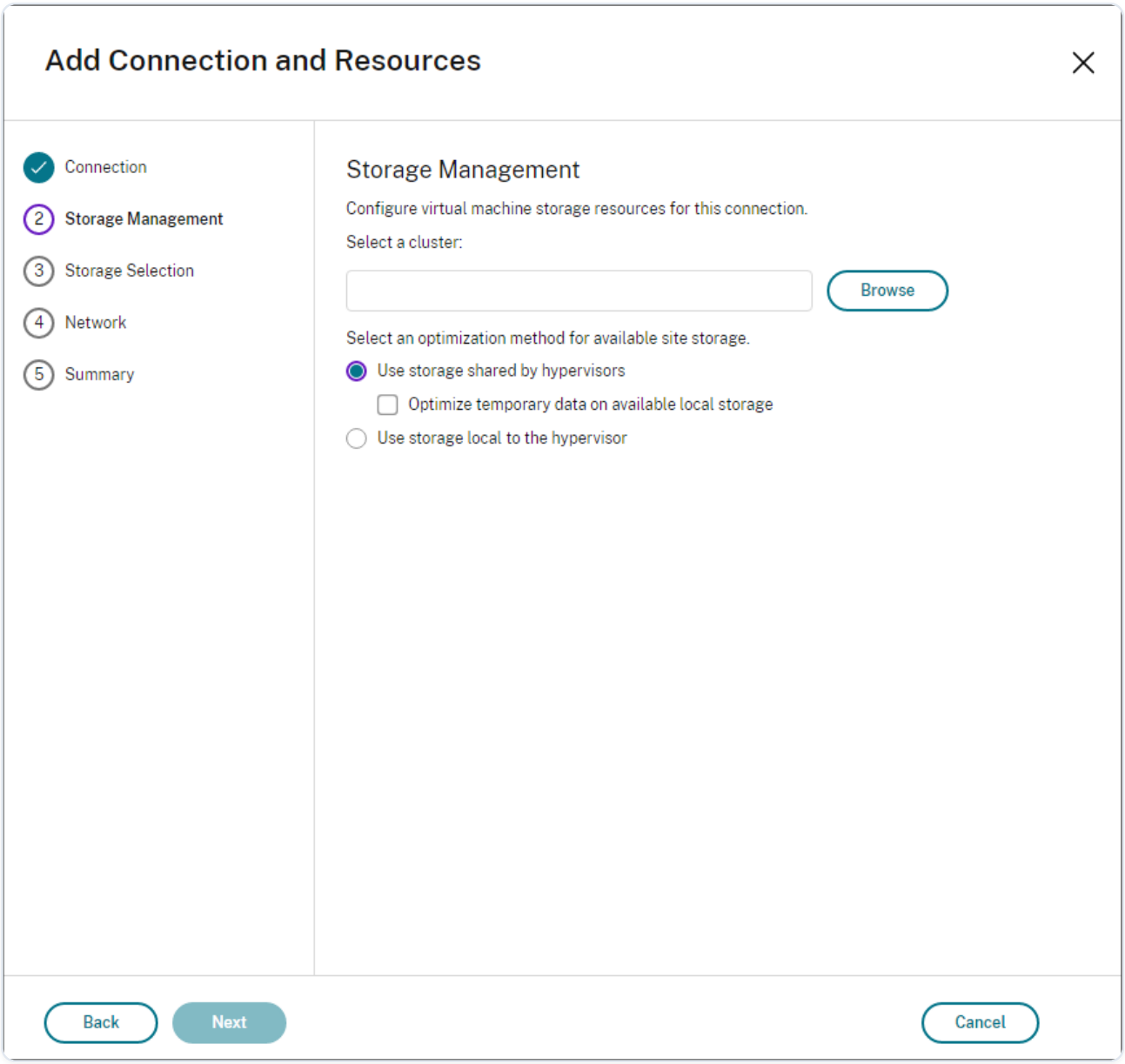
接続の追加イメージ(/en-us/linux-virtual-delivery-agent/current-release/media/add-conn-connection.png)

「**接続**」ページで：

- 接続を作成するには、 を選択します。既存の接続と同じホスト構成に基づいて接続を作成するには、 を選択し、関連する接続を選択します。
- 「**ゾーン名**」フィールドでゾーンを選択します。オプションは、構成したすべてのリソースの場所です。
- 「**接続の種類**」フィールドでハイパーバイザーまたはクラウドサービスを選択します。オプションは、ゾーンにプラグインが適切にインストールされているハイパーバイザーとクラウドサービスです。 または、PowerShellコマンド `Get-HypHypervisorPlugin -ZoneUid` を使用して、選択したゾーンで利用可能なハイパーバイザープラグインのリストを取得できます。
- 接続名を入力します。この名前は、**管理**表示に表示されます。
- 仮想マシンを作成するツールとして、Machine Creation ServicesまたはCitrix Provisioningを選択します。

接続ページの情報は、使用しているホスト（接続の種類）によって異なります。たとえば、Azure Resource Managerを使用する場合、既存のサービスプリンシパルを使用するか、新しいものを作成できます。

ステップ2b：ストレージ管理



ストレージ管理の種類と方法については、「ホストストレージ」を参照してください。

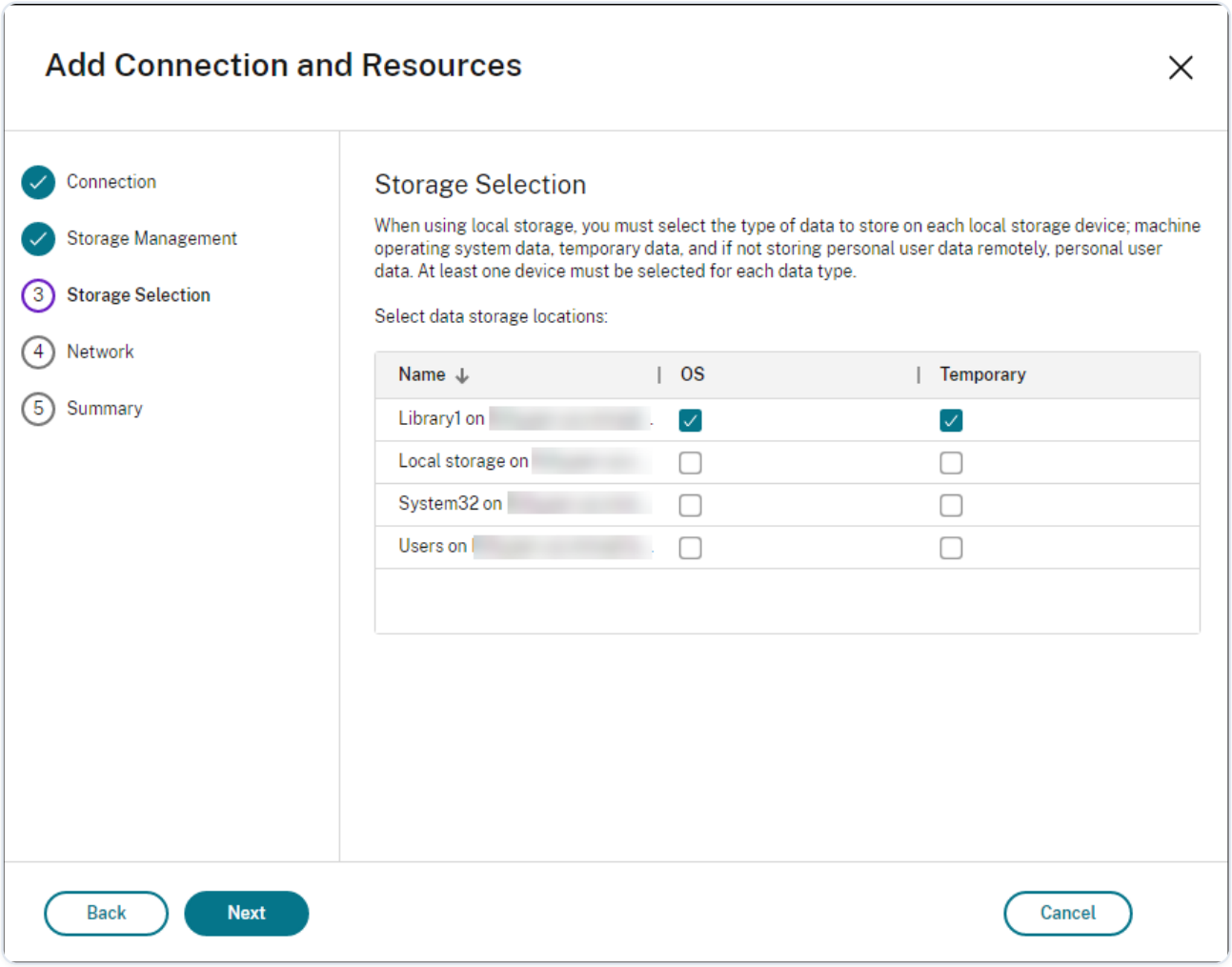
Hyper-V または VMware ホストへの接続を構成している場合は、クラスター名を参照して選択します。その他の接続タイプでは、クラスター名は要求されません。

ストレージ管理方法を選択します。ハイパーバイザーによって共有されるストレージ、またはハイパーバイザーにローカルなストレージ。

- ハイパーバイザーで共有されるストレージを選択した場合、利用可能なローカルストレージに一時データを保持するかどうかを指定します。（この接続を使用するマシンカタログで、デフォルト以外のサイズの一時ストレージを指定できます。）**例外：** クラスター共有ボリューム（CSV）を使用する場合、Microsoft System Center Virtual Machine Managerでは、一時データキャッシュディスクをローカルストレージに作成することはできません。**管理**コンソールでそのストレージ管理設定を構成しようとすると失敗します。

Citrix Hypervisorプールで共有ストレージを使用する場合、共有ストレージデバイスへの負荷を軽減するためにIntelliCacheを使用するかどうかを指定します。「Citrix Hypervisor仮想化環境」を参照してください。

ステップ2c：ストレージの選択



ストレージの選択の詳細については、「ホストストレージ」を参照してください。

利用可能なデータタイプごとに、少なくとも1つのホストストレージデバイスを選択します。前のページで選択したストレージ管理方法によって、このページで選択できるデータタイプが決まります。ウィザードの次のページに進む前に、サポートされているデータタイプごとに少なくとも1つのストレージデバイスを選択する必要があります。

ストレージの選択ページの下部には、ハイパーバイザーで共有されるストレージを選択し、**利用可能なローカルストレージ上の一時データを最適化**を有効にした場合、さらに多くの構成オプションが含まれています。一時データに使用するローカルストレージデバイス（同じハイパーバイザープール内）を選択できます。

現在選択されているストレージデバイスの数が表示されます（グラフィックでは「1 storage device selected」）。そのエントリにカーソルを合わせると、選択されているデバイス名が表示されます（デバイスが構成されていない場合を除く）。

1. 使用するストレージデバイスを変更するには、**「選択」**をクリックします。
2. **ストレージの選択**ダイアログボックスで、ストレージデバイスのチェックボックスをオンまたはオフにしてから、**OK**を選択します。

ステップ2d：地域

（一部のホストタイプでのみ表示されます。）地域の選択は、VMがどこに展開されるかを示します。理想的には、ユーザーがアプリケーションにアクセスする場所に近い地域を選択してください。

ステップ2e：ネットワーク

リソースの名前を入力します。この名前は、**管理**コンソールに表示され、接続に関連付けられたストレージとネットワークの組み合わせを識別します。 VMが使用するネットワークを1つ以上選択します。

一部の接続の種類（Azure Resource Managerなど）では、VMが使用するサブネットも表示されます。1つ以上のサブネットを選択します。

ステップ 2f: 概要

選択内容を確認します。変更する場合は、前のウィザードページに戻ります。確認が完了したら、**完了**を選択します。

注意: 一時データをローカルに保存する場合、この接続を使用するマシンを含むカタログを作成するときに、一時データストレージの非デフォルト値を構成できます。

> **注:**
>
> フルアクセス管理者にはスコープは表示されません。詳細については、「管理者、役割、およびスコープ」を参照してください。
>
> 詳細については、「接続の作成と管理」を参照してください。

ステップ 3: マスターイメージを準備する

> **ヒント:**
>
> ドメイン参加済みVDAと非ドメイン参加済みVDAの両方を作成するために、単一のイメージを使用できます。

(Citrix Hypervisor™ のみ) ステップ 3a: 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` を介して)。
- 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でvCPUの数を調整する。

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

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

2. お使いの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 `xe-guest-utilities` パッケージをインストールします。

   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 の場合:

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

3. XenCenter の **General** 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 **Optimized** になります。

(Azure、AWS、GCP の場合) ステップ 3b: Ubuntu 18.04 用に cloud-init を構成する

1. 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']
   ```

2. AWS 上の MCS で作成された VM にリモートでアクセスするために SSH を使用するには、これらの VM に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、次の操作を行います。

   - `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** の行とその後の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。MCS で作成された VM にログインするために、新しいパスワードを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 systemctl sshd restart` コマンドを実行します。

ステップ 3c: テンプレート VM に .NET と 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 .NET Runtime 6.0 をインストールし、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 をインストールします。

RHEL/CentOS の場合:

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

注:

GCPでホストされているRHEL 8.xにLinux VDAをインストールした後、VMを再起動するとイーサネット接続が失われ、Linux VDAに到達できなくなる場合があります。この問題を回避するには、VMを再起動する前に次のコマンドを実行してください。

```
nmcli dev connect eth0

systemctl NetworkManager restart
```

Ubuntu/Debianの場合:

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



apt-get install -f
```

SUSEの場合:

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

(RHEL 7のみ) ステップ3d: tdb-toolsパッケージをインストールするための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3e: (RHELおよびCentOSのみ) ntfs-3gを提供するEPELリポジトリをインストールする

RHEL 8、RHEL 7、およびCentOS 7にEPELリポジトリをインストールします。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>`の指示を参照してください。

ステップ3f: (SUSEのみ) ntfs-3gを手動でインストールする

SUSE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コンパイラコレクション (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 3g: MCS変数を構成する

`/etc/xdl/mcs/mcs.conf`ファイルを編集して、MCS変数を構成します。

以下は、ドメインに参加していないシナリオ用に構成できるMCS変数です。デフォルトの変数値を使用するか、必要に応じて変数をカスタマイズできます（オプション）。

`DOTNET_RUNTIME_PATH`=dotnetランタイムのインストールパス

`DESKTOP_ENVIRONMENT`=gnome | mate `REGISTER_SERVICE`=はい | いいえ

`ADD_FIREWALL_RULES`=はい | いいえ `VDI_MODE`=はい | いいえ

`START_SERVICE`=はい | いいえ

(オプション) ステップ 3h: MCSのレジストリ値を書き込むか更新する

テンプレートマシンで、`/etc/xdl/mcs/mcs_local_setting.reg`ファイルにコマンドラインを追加して、必要に応じてレジストリ値を書き込むか更新します。このアクションにより、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。

例えば、`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに次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。

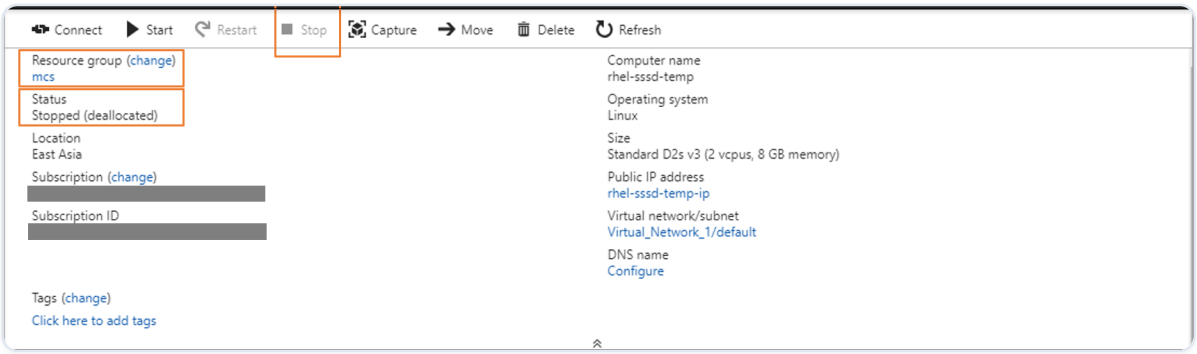
```
create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard\Cl
ipboardSelection" -t "REG_DWORD" -v "Flags" -d "0x00000003" --force
```

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

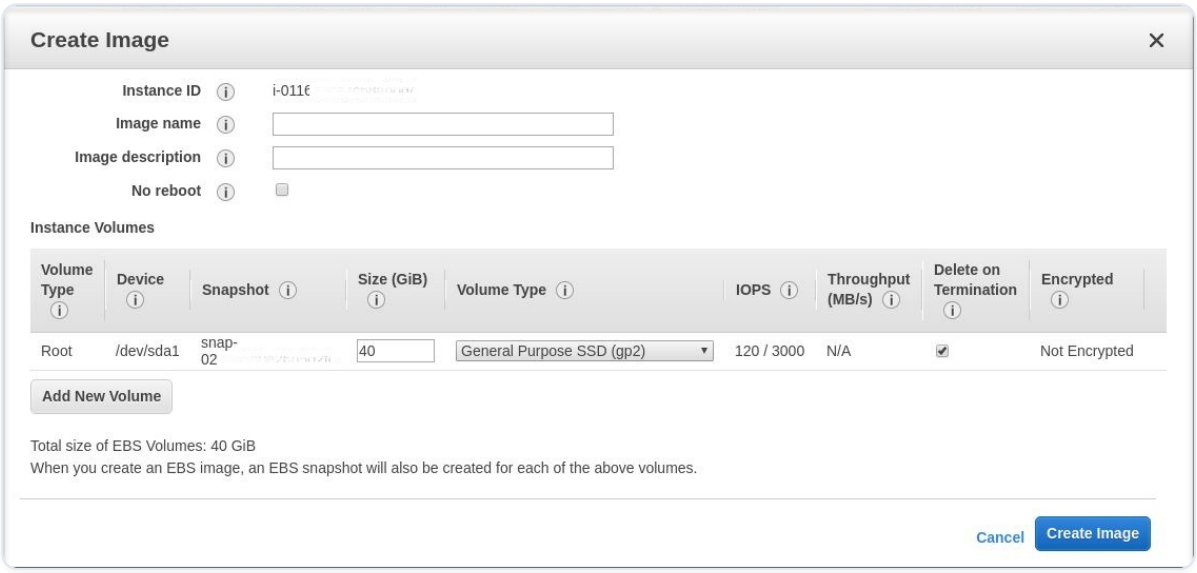
ステップ 3i: マスターイメージを作成する

1. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
2. 使用するパブリッククラウドに基づいて、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。

   - **（Citrix Hypervisor、GCP、およびVMware vSphereの場合）** テンプレートVMにアプリケーションをインストールし、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。
   - **（Azureの場合）** 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を VMをシャットダウンします。テンプレートVMの電源ステータスが**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。



   - **（AWSの場合）** 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を 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ます。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。必要に応じて情報を入力し、設定を行います。**イメージの作成**をクリックします。



- **(Nutanixの場合)** 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、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

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS**を選択します。
3. **Manage > Full Configuration**から、**Machine Catalogs**を選択します。
4. ウィザードの指示に従ってマシンカタログを作成します。

   Nutanix固有の**Container**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。

   Master Imageページで、イメージスナップショットを選択します。

   Virtual Machinesページで、仮想CPUの数とvCPUあたりのコア数を確認します。マシン展開方法としてMCSを選択し、カタログに作成するマシンのIDとして**Non-domain-joined**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、「マシンカタログの作成」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の電源を手動でオンにします。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。どのユーザーがこ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。

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

MCSを使用して、ドメイン参加済みVDAと非ドメイン参加済みVDAを作成できます。Citrix DaaSで非ドメイン参加済みLinux VDAを作成する場合は、専用の記事「Create non-domain-joined Linux VDAs using MCS」も参照してください。

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
デビアン 10.9	はい	はい	いいえ	はい
RHEL 8.10, RHEL 8.8, RHEL 8.6, RHEL 8.4, RHEL 8.3, RHEL 8.2, RHEL 8.1	はい	いいえ	はい	はい
RHEL 7.9, RHEL 7.8,	はい	はい	はい	はい

	ウィンバインド	SSSD	セントリファイ	PBIS
CentOS 7.9, CentOS 7.8				
ロッキー リナックス 8.10/8.8/8.6	はい	いいえ	いいえ	いいえ
SUSE 15.3, SUSE 15.2	はい	はい	いいえ	はい
ウブントゥ 20.04、ウブントゥ 18.04	はい	はい	いいえ	はい

- Citrix®は、関連するLinuxディストリビューションでの初期機能検証のために、以下のCentrifyバージョンを使用します。

LINUXディストリビューション	CENTRIFYバージョン
RHEL	5.8.0
SUSE	5.7.1
デビアン、ウブントゥ	5.6.1

  Centrifyの他のバージョンを使用すると、エラーが発生する可能性があります。テンプレートマシンをドメインに参加させるためにCentrifyを使用しないでください。

- MCSで作成されたマシンをWindowsドメインに参加させるためにPBISまたはCentrify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`ファイルでPBISまたはCentrifyパッケージのダウンロードパスを構成するか、またはPBISまたはCentrify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、その配下にあるすべてのMCSで作成されたマシンに対して書き込みおよびパスワードリセットの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。

サポートされているハイパーバイザー

- AWS
- シトリックス ハイパーバイザー™
- GCP
- マイクロソフト アジュール
- ニュータニックス AHV
- ヴイエムウェア ブイスフィア

サポートされているもの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。

MCSを使用してLinux VMを作成する

考慮事項

- 2203リリース以降、Citrix Virtual Apps and DesktopsおよびCitrix DaaS (旧称 Citrix Virtual Apps and Desktops service) 向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal SubscriptionまたはHybrid Rightsライセンスが必要です。Universal SubscriptionおよびHybrid Rightsライセンスの詳細については、「Transition and Trade Up (TTU) with Citrix Universal Subscription」を参照してください。
- ベアメタルサーバーは、仮想マシンを作成するためのMCSでの使用はサポートされていません。

(Nutanixのみ) ステップ1：Nutanix AHVプラグインのインストールと登録

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。詳細については、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用Nutanix AHVプラグインのインストールと登録

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

オンプレミスのDelivery Controller用Nutanix AHVプラグイン(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/nutanix-plug-in-for-cvad.png)

ステップ1b：クラウドDelivery Controller用Nutanix AHVプラグインのインストールと登録

Citrix Cloud™ Connectors用に**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、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：ホスト接続を作成する

このセクションでは、Azure、AWS、Citrix Hypervisor、GCP、Nutanix AHV、およびVMware vSphereへのホスト接続を作成する方法の例を示します。詳細については、Citrix Virtual Apps and Desktopsドキュメントの接続とリソースの作成および管理と、Citrix DaaSドキュメントの接続の作成および管理を参照してください。

- Citrix StudioでAzureへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-azure-in-citrix-studio)

- Citrix StudioでAWSへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-aws-in-citrix-studio)
- Citrix StudioでCitrix Hypervisorへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-citrix-hypervisor-in-citrix-studio)
- Citrix StudioでGCPへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-gcp-in-citrix-studio)
- Citrix StudioでNutanixへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-nutanix-in-citrix-studio)
- Citrix StudioでVMwareへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-vmware-in-citrix-studio)

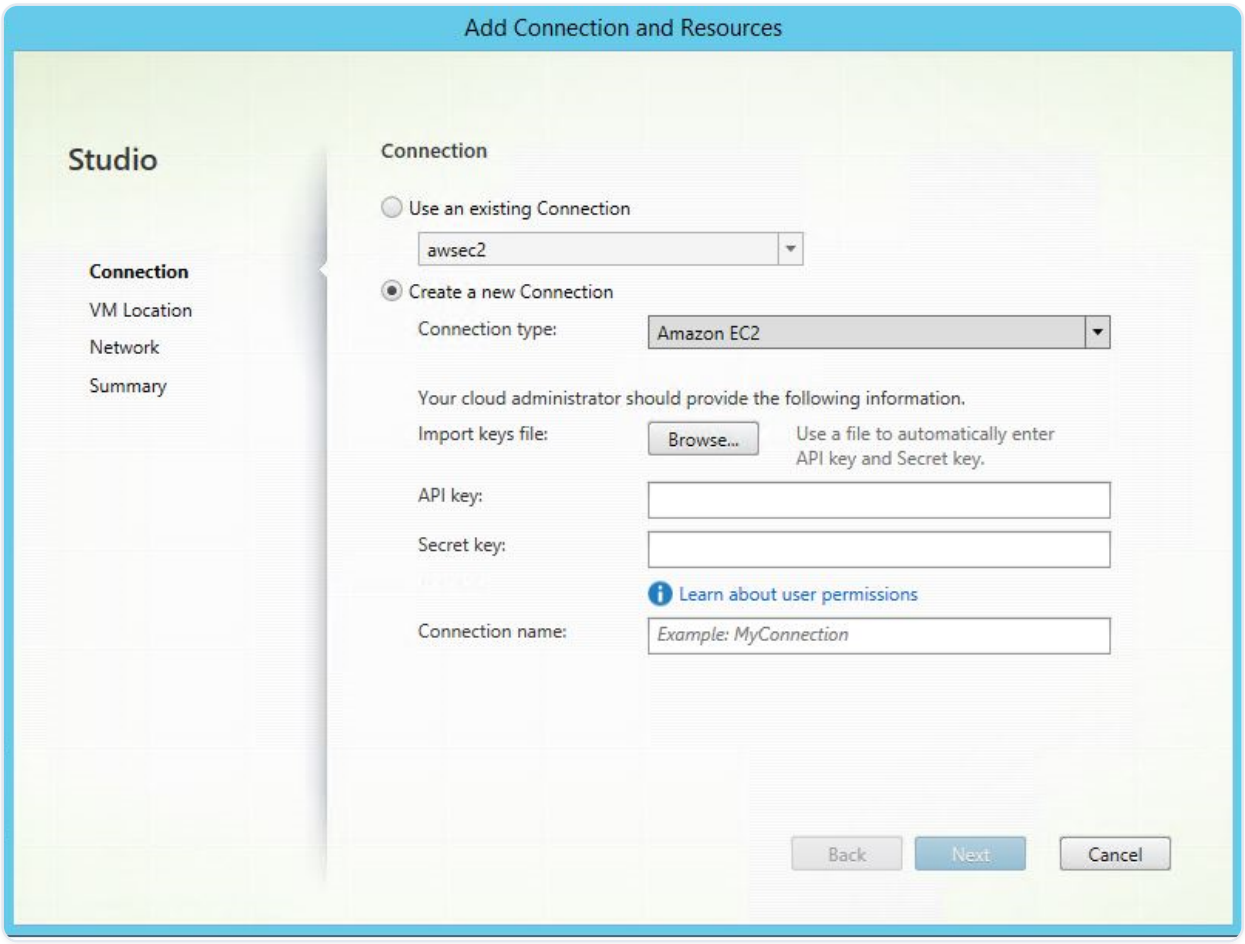
Citrix StudioでAzureへのホスト接続を作成する

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS**を選択します。
3. **Manage > Full Configuration**から、左ペインで**Hosting**を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/current-release/media/add-conn-connection.png)
5. 接続の種類としてMicrosoft Azureを選択します。
6. ウィザードはページを案内します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、**Summary**ページに到達するまで**Next**を選択します。詳細については、ドメインに参加していないLinux VDAの作成の記事の手順2：ホスト接続の作成を参照してください。

Citrix StudioでAWSへのホスト接続を作成する

1. Citrix Cloud の Citrix Studio で、**設定 > ホスティング > 接続とリソースの追加** を選択します。
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。

   アクセスキーペア(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/access-key-pair.png)

   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ものを作成できます。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. **.csvファイルのダウンロード**をクリックし、アクセスキーを安全な場所に保存します。

Citrix StudioでCitrix Hypervisorへのホスト接続を作成する

1. Citrix Cloudにサインインします。

2. 左上のメニューで、**マイサービス > DaaS**を選択します。
3. **管理 > 完全な構成**から、左ペインで**ホスティング**を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。
5. 接続タイプとして**Citrix Hypervisor**を選択します。
6. Citrix Hypervisorアカウントの接続アドレス（Citrix Hypervisor URL）、ユーザー名とパスワード、および接続名を入力します。

Citrix StudioでGCPへのホスト接続を作成する

「Google Cloud Platform 仮想化環境」(/ja-jp/citrix-virtual-apps-desktops-service/install-configure/resource-location/google.html)に従ってGCP環境をセットアップし、その後、以下の手順を完了してGCPへのホスト接続を作成します。

1. Citrix Cloudにサインインします。
2. 左上のメニューで、 **>** を選択します。
3. **>** から、左ペインで を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。
5. 接続の種類として を選択します。
6. 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。
7. ウィザードはページを案内します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、 を選択して ページに到達します。詳細については、「非ドメイン参加Linux VDAの作成」(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-non-domain-joined-linux-vdas.html)の記事の「手順2：ホスト接続の作成」(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/create-non-domain-joined-linux-vdas.html#step-2-create-a-host-connection)を参照してください。

Citrix StudioでNutanixへのホスト接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで **> >** を選択します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで **> >** 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. ウィザードで、 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。 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：

オンプレミスのCitrix StudioでNutanixへのホスティング接続を作成する(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/creating-a-hosting-connection-to-nutanix-in-the-on-premises-citrix-studio.png)

Citrix StudioでVMwareへのホスト接続を作成する

1. vSphere環境にvCenter Serverをインストールします。詳細については、VMware vSphereを参照してください。
2. Citrix Studioで、 **> >** を選択して、VMware vSphereへの接続を作成します。

   (/en-us/linux-virtual-delivery-agent/2203-ltsr/media/create-connection-to-azure.png)
3. 接続の種類としてVMware vSphereを選択します。

   VMware vSphereの選択(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/choose-vsphere.png)
4. VMwareアカウントの接続アドレス（vCenter Server URL）、ユーザー名とパスワード、および接続名を入力します。

   VMware接続名(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/connection-name.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vSphere接続(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/new-vsphere-connection-2.png)

ステップ3：マスターイメージを準備する

（Citrix Hypervisorのみ）ステップ3a：Citrix VM Toolsをインストールする

xe CLIまたはXenCenterを使用する各VMのテンプレート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をクリーンにシャットダウン、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. お使いの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/Rocky Linux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の場合:

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

3. XenCenterの**General**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**Optimized**になります。

(Azure、AWS、GCPの場合) ステップ3b: Ubuntu 18.04向けにcloud-initを構成する

1. VMが再起動または停止されたときに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']
   ```

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、次の操作を行います。

   - `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使用するので、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 systemctl sshd restart`コマンドを実行します。

ステップ3c: テンプレートVMに.NETおよび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ます。

> .NET Runtime 6.0 をテンプレート VM にインストールし、Linux ディストリビューションに基づいて以下のコマンドを実行して Linux VDA をインストールします。

RHEL/CentOS/Rocky Linux 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 の場合:

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

ステップ 3d: tdb-tools パッケージをインストールするための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 3e: (RHEL および CentOS のみ) 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、RHEL 7、および CentOS 7 に EPEL リポジトリをインストールします。EPEL のインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>` の手順を参照してください。

ステップ 3f: (SUSE のみ) ntfs-3g を手動でインストールする

SUSE 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 3g: MCS 構成ファイルを編集する

1. `/etc/xdl/mcs/mcs.conf` 内の変数を変更します。

   - **ドメインに参加していないシナリオの場合**

     ドメインに参加していないシナリオの場合、`/etc/xdl/mcs/mcs.conf` 内の変数を未指定のままにするか、必要に応じて以下の変数を変更できます:

     `DOTNET_RUNTIME_PATH`=dotnet-runtimeのインストールパス

     `DESKTOP_ENVIRONMENT`= gnome | mate `VDA_PORT`=ポート番号

     `REGISTER_SERVICE`=Y | N `ADD_FIREWALL_RULES`=Y | N

     `HDX_3D_PRO`=Y | N `VDI_MODE`=Y | N

     `SITE_NAME`=DNSサイト名 | '<none\>' `SEARCH_BASE`=検索ベースセット | '<none\>'

     `START_SERVICE`=Y | N `TELEMETRY_SOCKET_PORT`=ポート番号

     `TELEMETRY_PORT`=ポート番号

     > **ヒント:**

> `/etc/xdl/mcs/mcs.conf` 内の `AD_INTEGRATION` 変数はデフォルトで `Winbind` に設定されています。デフォルト値は、ドメインに参加していないシナリオには影響しません。

- **ドメインに参加しているシナリオの場合**

  `/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 AD 関連構成ファイル (/etc/krb5.conf、/etc/sssd.conf、および /etc/samba/smb.conf) 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` および `AD_INTEGRATION` 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`: 各 DNS サーバーの IP アドレスを設定します。最大 4 つの DNS サーバーを設定できます。
  - `NTP_SERVER`: NTP サーバーの IP アドレスを設定します。特に指定がない限り、ドメインコントローラーの IP アドレスです。
  - `WORKGROUP`: AD で構成した NetBIOS 名 (大文字と小文字を区別) にワークグループ名を設定します。そうしないと、MCS はマシンホスト名の直後にあるドメイン名の一部をワークグループ名として使用します。たとえば、マシンアカウントが **user1.lvda.citrix.com** の場合、MCS はワークグループ名として **lvda** を使用しますが、正しい選択は **citrix** です。ワークグループ名を正しく設定してください。
  - `AD_INTEGRATION`: Winbind、SSSD、PBIS、または Centrify 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。
  - `CENTRIFY_DOWNLOAD_PATH`: Server Suite Free (旧 Centrify Express) パッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION` 変数を Centrify に設定した場合にのみ有効になります。
  - `CENTRIFY_SAMBA_DOWNLOAD_PATH`: Centrify Sambaパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION` 変数をCentrifyに設定した場合にのみ有効になります。

- `PBIS_DOWNLOAD_PATH`: PBISパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をPBISに設定した場合にのみ有効になります。
- `UPDATE_MACHINE_PW`: マシンアカウントのパスワード更新の自動化を有効または無効にします。詳しくは、「マシンアカウントのパスワード更新を自動化する」を参照してください。
- 次のLinux VDA構成変数:

  `DOTNET_RUNTIME_PATH`=*dotnetランタイムのインストールパス *

  `DESKTOP_ENVIRONMENT`= *gnome | mate ** `SUPPORT_DDC_AS_CNAME`=**Y | N**

  `VDA_PORT`=**ポート番号** `REGISTER_SERVICE`=**Y | N**

  `ADD_FIREWALL_RULES`=**Y | N** `HDX_3D_PRO`=**Y | N**

  `VDI_MODE`=**Y | N** `SITE_NAME`=**DNSサイト名 | '<none\\>'**

  `LDAP_LIST`=**'LDAPサーバーリスト' | '<none\\>'** `SEARCH_BASE`=**検索ベース設定 | '<none\\>'**

  `FAS_LIST`=**'FASサーバーリスト' | '<none\\>'** `START_SERVICE`=**Y | N**

  `TELEMETRY_SOCKET_PORT`=**ポート番号** `TELEMETRY_PORT`=**ポート番号**

`mcs.conf`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。

```
#!/bin/bash

#############################################################################
#
# Citrix Virtual Apps & Desktops For Linux Script: Machine Creation Service
# Copyright (c) Citrix Systems, Inc. All Rights Reserved.
#
# This is the configuration file for mcs scripts.



#########################Template machine check#################################
# If unspecified, the value is N by default, meaning that mcs configuration templates will overwrite configuration items
# If you choose Y, MCS created VMs will use the existing configurations of the current VDA that must running correctly
Use_Existing_Configurations_Of_Current_VDA=N

#########################DNS Configuration######################################
# Provide DNS information
# You can provide 4 DNS servers at most.
# Leave empty if you do not have 4 servers. You may also leave all of them empty
# and configure dns manually.
# Format:
# dns1="xx.xx.xx.xx"
# dns2="xx.xx.xx.xx"
# dns3=
# dns4=
dns1="192.1681.5"
dns2="192.168.3.4"
dns3=
dns4=

#########################NTP Configuration######################################
# Provide NTP server information.
# If not set here, the default value will be the address of domain controller.
# Format:
# NTP_SERVER="xx.xx.xx.xx"
NTP_SERVER="192.168.4.5"

#########################WORKGROUP Configuration################################
# Provide Workgroup information.
# Usually workgroup is the same with domain name and you do not need to configure it here.
# If that is not the case, please config it according to the correct format:
# WORKGROUP="workgroup_name"
WORKGROUP="example"

#########################Domain Join Configuration##############################
# Provide Domain Join method.
# Winbind: support RHEL7/CentOS7, RHEL8/CentOS8, SUSE12, SUSE15, Ubuntu16.04, Ubuntu18.04, Ubuntu20.04
# SSSD: support RHEL7/CentOS7, Ubuntu16.04, Ubuntu18.04, Ubuntu20.04, SUSE12, SUSE15
# Centrify: support RHEL7/CentOS7
# PBIS: support RHEL7/CentOS7, RHEL8/CentOS8, SUSE12, SUSE15, Ubuntu16.04, Ubuntu18.04, Ubuntu20.04, Debian 10
# AD_INTEGRATION="winbind" or AD_INTEGRATION="sssd" or AD_INTEGRATION="centrify" or AD_INTEGRATION="pbis"
AD_INTEGRATION="winbind"

#########################Centrify download path Configuration###################
# When choose Centrify as AD_INTEGRATION, provide Centrify download path with related distribution if Centrify is not installed.
# To find out the correct download url for your os, you may go here:
# https://www.centrify.com/express/linux/download-files/#accordion-download-express-02
CENTRIFY_DOWNLOAD_PATH=


#########################Centrify Samba download path Configuration##############
# When choose Centrify as AD_INTEGRATION, provide Centrify Samba download path with related distribution if Centrify is not installed.
# CENTRIFY_SAMBA_DOWNLOAD_PATH="http://edge.centrify.com/products/opensource/samba-4.5.9/centrify-samba-4.5.9-rhel3-x86_64.tgz
CENTRIFY_SAMBA_DOWNLOAD_PATH=


#########################PBIS download path Configuration########################
# When choose PBIS as AD_INTEGRATION, provide PBIS download path if PBIS is not installed.
# PBIS: support RHEL7/CentOS7, RHEL8/CentOS8, SUSE12/SUSE15, Ubuntu16.04, Ubuntu18.04, Ubuntu20.04
# RHEL7 and SUSE12: "https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh"
# RHEL8: "https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh"
# Ubuntu: "https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh"
# SUSE15: "https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh"
# PBIS_DOWNLOAD_PATH="https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh"
PBIS_DOWNLOAD_PATH=


#########################Machine Password Automate Update########################
# Machine password will expire after 30 days(default), so we have a mechnisum to update
# this password regularly.
# You can set this value to enabled to enable this feature.
UPDATE_MACHINE_PW="disabled"

#########################Linux VDA Configuration#################################
# Provide Linux VDA configuration information.
# Please refer to Linux VDA Documentation for these settings.
DOTNET_RUNTIME_PATH=/opt/dotnet
DESKTOP_ENVIRONMENT=gnome
SUPPORT_DDC_AS_CNAME=N
VDA_PORT=80
REGISTER_SERVICE=Y
ADD_FIREWALL_RULES=Y
```

2. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。
3. 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3h: 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. (現在実行中のVDAをテンプレートVMとして使用している場合は、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。

   Centrifyユーザーの場合は、`/etc/xdl/mcs/centrify_krb5.conf.tmpl`および`/etc/xdl/mcs/centrify_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、$realm、${new_hostname}、$AD_FQDNなどの変数を使用します。

3. 使用しているパブリッククラウドに基づいて、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。

   - **(Citrix Hypervisor、GCP、およびVMware vSphereの場合)** テンプレートVMにアプリケーションをインストールし、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。
   - **(Azureの場合)** 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を VMをシャットダウンします。テンプレートVMの電源ステータスが**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、この名前が必要です。

     テンプレートVMの停止済み電源ステータス(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/stopped-deallocated2.png)
   - **(AWSの場合)** 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ます。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。必要に応じて情報を入力し、設定を行います。**イメージの作成**をクリックします。

     EBSイメージの作成中(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/create-ebs-image.png)
   - **(Nutanixの場合)** 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、名前を付けます。

     > **注:**
     >
     > Citrix Virtual Apps™および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。

(GCPの場合) ステップ3i: RHEL 8.xでイーサネット接続を構成する

GCPでホストされているRHEL 8.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAに到達できなくなる可能性があります。この問題を回避するには、VMを再起動する前に次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl NetworkManager restart
```

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ときは、マスターイメージを選択します。以下に例を示します。

テンプレートVMのVHDを見つける(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/find-vhd-of-template-vm2.png)

マスターイメージの選択(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/choose-master-image-from-the-list.png)

Nutanix固有の**Container**ページで、以前テンプレートVMに指定したコンテナを選択します。**Master Image**ページで、イメージスナップショットを選択します。**Virtual Machines**ページで、仮想CPUの数とvCPUあたりのコア数を確認します。

> **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起動してください。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。

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、Citrix Virtual Apps and Desktopsのドキュメントにあるマシンカタログの作成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。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 Runtime 6.0がインストールされ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:

Linux VDAのセルフアップデート機能を使用して、自動ソフトウェアアップデートをスケジュールすることもできます。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/CentOS 8/Rocky Linux 8の場合：

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

SUSEの場合：

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

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。

マシンアカウントパスワードの更新を自動化する

マシンアカウントのパスワードは、デフォルトでは、マシンカタログが作成されてから30日後に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 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)`

この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	PBIS
RHEL 8、CentOS 8	はい	いいえ	いいえ	はい
RHEL 7、CentOS 7	はい	はい	いいえ	はい
ロッキー Linux 8	はい	いいえ	いいえ	いいえ
ウブントゥ 20.04	はい	いいえ	いいえ	いいえ
ウブントゥ 18.04	はい	いいえ	いいえ	いいえ
デビアン 10.9	はい	いいえ	いいえ	いいえ
SUSE 15.3	はい	いいえ	いいえ	いいえ
SUSE 15.2	はい	いいえ	いいえ	いいえ
SUSE 12.5	はい	いいえ	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。詳しくは、「ctxfascfg.shの実行」を参照してください。
3. `/etc/xdl/mcs/mcs.conf` で変数を設定します。

> **注:**
>
> これらの変数はVMの起動時に呼び出されるため、`/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、または複数のFASサーバーアドレスに設定します。複数のアドレスはセミコロンで区切り、アドレスを単一引用符で囲みます。例: `FAS_LIST='<FAS_SERVER_FQDN>;<FAS_SERVER_FQDN>'`。
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

前述のようにテンプレートマシンでFASが有効になっていない場合、各MCSで作成された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。

> **注:**
>
> これらの変数はVMの起動時に呼び出されるため、`/etc/xdl/mcs/mcs.conf`で必要なすべての変数を設定します。

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、必要に応じて他の変数を設定します。

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。詳しくは、「ctxfascfg.shの実行」を参照してください。

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

Citrix Provisioning を使用して、ドメイン参加済み VDA を作成できます。

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。

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 18.04
- ウブントゥ 20.04
- RHEL 8.4
- RHEL 8.3
- RHEL 7.9
- SUSE 12.5
- SUSE 15.2
- SUSE 15.3

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 `Citrix_Provisioning_2203.iso` を使用することをお勧めします。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2109以降を使用するには、Citrix Provisioning Server 2109以降が必要です。
- Citrix Provisioningを使用してLinuxターゲットデバイスをストリーミングする場合、プロビジョニングされたデバイスが期待どおりに起動できるように、単一の共有ディスクイメージ上に個別のブートパーティションを作成してください。
- `btrfs` でパーティションをフォーマットしないでください。GRUB2には、 `btrfs` パーティションを見つける際に固有の問題があります。GRUBは `GRand Unified Bootloader` の略です。

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。

Citrix DaaS Standard for Azure で Linux VDA を作成する

Citrix DaaS Standard for Azure では、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 を作成して、Microsoft Azure からあらゆる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 DaaS 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

この機能は、以下の Linux ディストリビューションでサポートされています。

- RHEL 8.10
- RHEL 8.8
- RHEL 8.6
- RHEL 8.3
- RHEL 8.2
- RHEL 7.8
- ロッキー・リナックス 8.10/8.8/8.6
- ウブントゥ 20.04
- ウブントゥ 18.04

ステップ 1：Azure でマスターイメージを準備する

注:

自動ソフトウェア更新をスケジュールするには、Linux VDA のセルフアップデート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
```

1. 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。
2. 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。
3. VMに、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。
4. (Ubuntuのみ) `/etc/network/interfaces`ファイルに`source /etc/network/interfaces.d/*`行を追加します。
5. (Ubuntuのみ) `/run/systemd/resolve/stub-resolv.conf`ではなく`/run/systemd/resolve/resolv.conf`を指すように`/etc/resolv.conf`を設定します。

   ```
   unlink /etc/resolv.conf



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

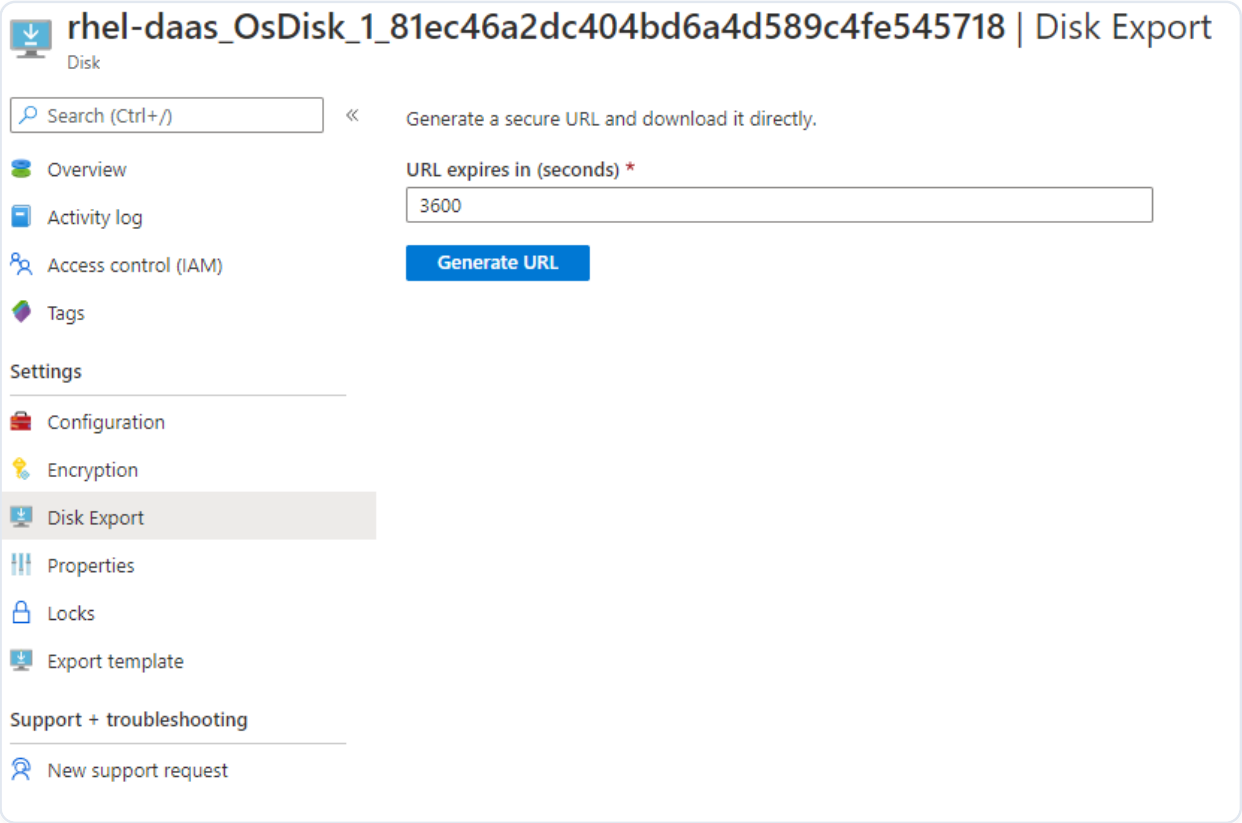
6. Linux VDAパッケージをインストールします。
7. /etc/xdl/mcs/mcs.confの変数を変更します。`mcs.conf`構成ファイルには、MCSとLinux VDAを設定するための変数が含まれています。

   注:

   dns変数は未指定のままにします。 マシンカタログを作成する際に**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。

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

9. 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。



10. (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印刷用の**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。

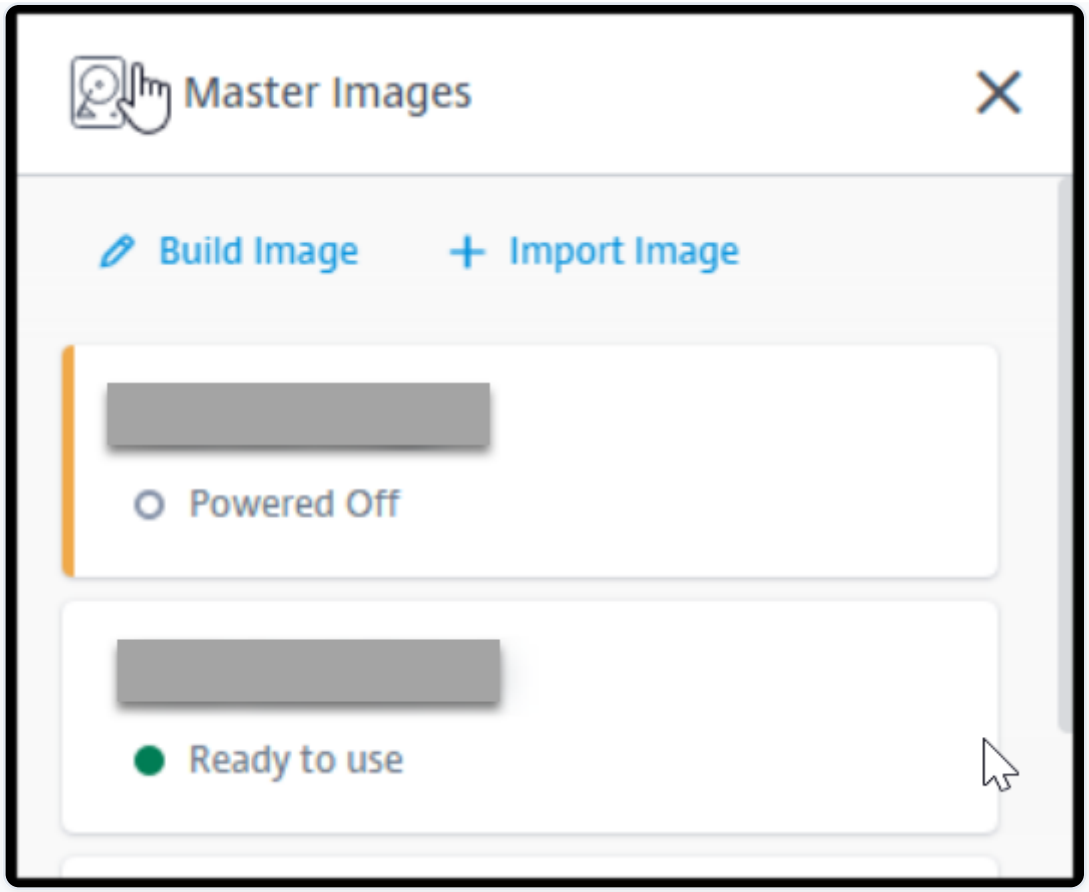
```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\GroupPolicy\Defaults\PrintingPolicies" -t
"REG_DWORD" -v "AutoCreatePDFPrinter" -d "0x00000001" –force
```

ステップ2：Azureからマスターイメージをインポートする

1. 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

> **Managed Desktops**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。



2. **イメージのインポート**をクリックします。
3. 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。

   Azureからイメージをインポートする(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/import-an-image-from-azure.png)
4. 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。

ステップ3：マシンカタログを作成する

管理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ます。

> **注:**

> マスターイメージとして使用されるVMは、SSHまたはRDP経由ではアクセスできません。VMにアクセスするには、Azureポータルでシリアルコンソールを使用します。

Linux VDA を手動でインストールする

以下の Linux ディストリビューションに Linux VDA を手動でインストールできます。

- Amazon Linux 2、CentOS、RHEL、および Rocky Linux に Linux VDA を手動でインストールする
- SUSE に Linux VDA を手動でインストールする
- Ubuntu に Linux VDA を手動でインストールする
- Debian に Linux VDA を手動でインストールする

Amazon Linux 2、CentOS、RHEL に Linux VDA を手動でインストールする

> **重要:**

> 新規インストールの場合、迅速なインストールには**easy install**を使用することをお勧めします。Easy install は、時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: ネットワーク構成を確認する

ネットワークが正しく接続され、構成されていることを確認してください。たとえば、Linux VDA で DNS サーバーを構成する必要があります。

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 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。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。この規則は 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の確認

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1f: クロック同期の構成

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（VM）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、リモート時刻サービスとの時刻同期が推奨されます。

RHEL 8または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サービスの構成

ルートユーザーとして、**/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
```

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、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: PostgreSQLのインストール

Linux VDAにはPostgreSQLが必要です。

- Amazon Linux 2、RHEL 7、および CentOS 7 向け PostgreSQL 9
- RHEL 8 および Rocky Linux 8 向け PostgreSQL 10

PostgreSQL をインストールするには、次のコマンドを実行します。

```
sudo yum -y install postgresql-server



sudo yum -y install postgresql-jdbc
```

データベースを初期化し、マシンの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アクションが必要です。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。

```
sudo postgresql-setup initdb
```

ステップ 1h: PostgreSQL を開始する

マシンの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
```

PostgreSQL のバージョンを確認するには、次を使用します。

```
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 VM として実行する場合、いくつかの変更が必要です。使用中のハイパーバイザープラットフォームに基づいて、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™ 上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 の問題が発生します。両方ともシステムクロックを管理しようとします。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。

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に関する問題が発生します。どちらもシステムクロックを同期しようとします。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。

ステップ3：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・ワインバインド
- 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 Winbind

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およびRocky Linux 8の場合：

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

Amazon Linux 2、CentOS 7、およびRHEL 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認証を構成する

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およびRocky Linux 8の場合：

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

   Amazon Linux 2および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
   ```

   ここで、**REALM**はKerberosレルム名（大文字）、**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`サービスが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ときに`authconfig`が`winbind`サービスを開始しようとすると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およびRocky Linux 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]` セクションで:

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

   `default_realm = REALM` `dns_lookup_kdc = true`

   `[realms]` セクションで:

   `REALM = {`

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

   `[domain_realm]` セクションで:

   `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。

RHEL 8 および Rocky Linux 8 の場合:

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

Amazon Linux 2 および RHEL 7 の場合:

```
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。

WINBIND 用に PAM を構成する

デフォルトでは、Winbind PAM モジュール (pam_winbind) の構成では、Kerberos チケットのキャッシュとホームディレクトリの作成が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。これらの変更には、Winbind デーモンの再起動が必要です。

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

> **ヒント:**
>
> マシンがドメインに参加している場合にのみ、`winbind` デーモンは実行を継続します。
>
> **/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。

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

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
```

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、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
```

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。ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/redhat.html#step-6-install-the-linux-vda)に進んでください。

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も同様に適用されます。

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=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。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）。

ドメイン参加後、Linuxマシンを再起動します。

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 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6：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 `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。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6：Linux VDAをインストールするに進みます。

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。

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、ドメインへの参加や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、Amazon Linux 2およびRHEL 7での**Samba**アプローチと、RHEL 8での**adcli**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 (アマゾン リナックス 2および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 と 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
  ```

  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、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。
> 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、**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。

- **Adcli (RHEL 8 および Rocky Linux 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 と 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:

  `REALM = {`

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

セクションの下:

`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、**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/2203-ltsr/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 のファイルの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 および Rocky Linux 8 の場合：

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
```

Amazon Linux 2、CentOS 7、RHEL 7の場合：

`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 sssd start



sudo chkconfig sssd on
```

KERBEROS構成の確認

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。シェル置換を防ぐために、ドル記号 (**$**) はバックスラッシュ (****) 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
```

ユーザー認証の確認

ログオン形式がサポートされており、NSSが機能す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、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\\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6: 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

CentOS 7およびRHEL 7の場合、例:

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

Amazon Linux 2およびRHEL 8の場合、例:

```
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

CentOS 7およびRHEL 7の場合、例:

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

Amazon Linux 2およびRHEL 8の場合、例:

```
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

CentOS 7およびRHEL 7の場合、例:

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

Amazon Linux 2およびRHEL 8の場合、例:

```
sh pbis-open-9.1.0.551.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り、ドメインにコンピューターを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、Active Directoryドメインにコンピューターを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必要とします。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、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、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6: Linux VDAをインストールするに進みます。

手順4: .NET Runtime 6.0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 Runtime 6.0をインストールします。

.NET Runtime 6.0のインストール後、**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ランタイムバイナリパスを設定します。例えば、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。

手順5: 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。

手順6：Linux VDAをインストールする

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サービスを停止する：

   ```
   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、**systemctl 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する：

```
sudo rpm -e XenDesktopVDA
```

> **注:**
>
> コマンドを実行するには、完全なパスが必要です。または、**/opt/Citrix/VDA/sbin**および**/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。

     Amazon Linux 2の場合：

     ```
     sudo yum install -y XenDesktopVDA-<version>.amzn2.x86_64.rpm
     ```

     RHEL 8およびRocky Linux 8の場合:

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

     CentOS 7およびRHEL 7の場合:

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

     RHEL 8およびRocky Linux 8のRPM依存関係リスト:

```
qt5-qtbase >= 5.5~


ibus >= 1.5


nss-tools >= 3.44.0


gperftools-libs >= 2.4


cyrus-sasl-gssapi >= 2.1


python2 >= 2.7~


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
```

Amazon Linux 2、CentOS 7、およびRHEL 7のRPM依存関係リスト:

```
qt5-qtbase >= 5.5~


libmspack >= 0.5


ibus >= 1.5


cyrus-sasl-gssapi >= 2.1


gperftools-libs >= 2.4


nss-tools >= 3.44.0
```

```
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

xorg-x11-server-Xorg >= 1.20.4
```

```
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、「システム要件」を参照してください。

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、「セッションのシャドウイング」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は、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の下にある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。

- `Yum`を使用してソフトウェアをアップグレードするには：

  Amazon Linux 2の場合:

  ```
  sudo yum install -y XenDesktopVDA-<version>.amzn2.x86_64.rpm
  ```

  RHEL 8およびRocky Linux 8の場合:

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

  CentOS 7およびRHEL 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：

  Amazon Linux 2の場合:

  ```
  sudo rpm -U XenDesktopVDA-<version>.amzn2.x86_64.rpm
  ```

  RHEL 8 の場合:

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

  CentOS 7 および RHEL 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、前述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. .NET ランタイムの正しいパスを設定するには、`/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。

手順 7: 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 NVIDIA GRID ドライバーをインストールする必要があります。

> **注:**

> Amazon Linux 2 で HDX 3D Pro を使用するには、NVIDIA ドライバー 470 をインストールすることをお勧めします。詳細については、「システム要件」を参照してください。

> 特定のハイパーバイザーに NVIDIA GRID Virtual GPU Manager (ホストドライバー) をインストールして構成するには、以下のガイドを参照してください。

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRID ゲスト VM ドライバーをインストールして構成するには、以下の手順を実行します。

1. ゲスト VM がシャットダウンされていることを確認します。
2. XenCenter® で、VM に GPU を割り当てます。
3. VM を起動します。
4. NVIDIA GRIDドライバー用にVMを準備します。

```
yum install gcc



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



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

5. 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（「no」）を選択します。

> **重要:**
>
> GPUパススルーが有効になると、Linux VMには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`

大規模な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。

ステップ8：Linux VDAを構成する

> **注:**
>
> ランタイム環境をセットアップする前に、OSに**en_US.UTF-8**ロケールが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**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集して**# en_US.UTF-8 UTF-8**行のコメントを解除し、**sudo locale-gen**コマンドを実行します。

> 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提供します。必要なすべての変数が存在する場合、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に登録するために、スペースで区切られたDelivery Controllerの完全修飾ドメイン名 (FQDN) の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 \| 5** – 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
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、グラフィックを多用する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DNSをクエリして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を介してLDAPをクエリします（例：DC=mycompany,DC=com）。検索パフォーマンスを向上させるには、検索ベースを指定できます（例：OU=VDI,DC=mycompany,DC=com）。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**\<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。

- **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 Runtime 6.0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。

  次の手順を実行して、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`または`.Xclients`ファイルを作成します。Amazon Linux 2を使用している場合は、`.Xclients`ファイルを作成します。他のディストリビューションを使用している場合は、`.xsession`ファイルを作成します。

  2\. `.xsession`または`.Xclients`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。

  - **Amazon Linux 2およびRHEL 8上のMATEデスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

  - **CentOSおよびRHEL上のGNOMEデスクトップの場合**

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



if [ -n "$GSESSION" ]; then
```

```
export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **Amazon Linux 2上のGNOMEデスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

b. 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され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。

ステップ9：XDPingを実行する

Linux VDA環境で一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳しくは、「XDPing」を参照してください。

ステップ10：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 を起動する:

Linux VDA サービスを起動するには:

```
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` および `ctxhdx` サービスを停止する前に、`systemctl 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 /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 11: 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、「マシンカタログを作成する」および「マシンカタログを管理する」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、**Multi-session OS** オプション。
  - VDI 専用デスクトップ配信モデルの場合は、**Single-session 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12：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 2203」を参照してください。

SUSE に Linux VDA を手動でインストールする

> **重要:**

> 新規インストールの場合、迅速なインストールには**easy install**の使用をお勧めします。easy installは、時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ1：構成情報とLinuxマシンの準備

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に基づいています。テキストベースのYaSTツールも使用できますが、ここでは文書化されていない異なるナビゲーション方法があります。

ホスト名とドメインネームシステム (DNS) の構成

1. UIベースのYaSTツールを起動します。
2. **System**を選択し、次に**Network Settings**を選択します。
3. SLED 12のみ：**Global Options**タブで、**Network Setup Method**を**Wicked Service**に変更します。
4. **Hostname/DNS**タブを開きます。
5. **Assign Hostname to Loopback IP**を選択します。
6. 「**DHCP経由でホスト名を変更**」のチェックを外します。

7. 「**ループバックIPにホスト名を割り当てる**」を選択します。
8. 「**DNS構成の変更**」で「**カスタムポリシーを使用**」オプションを選択します。
9. 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。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、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。

マルチキャストDNSを無効にする

SLEDのみ、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。SLESではmDNSはデフォルトで有効になっていないため、アクションは不要です。

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
```

次のように変更します。

```
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 (VM) 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。

SUSE 15.3およびSUSE 15.2の場合：

1. UIベースのYaSTツールを起動します。
2. **Network Services**を選択し、次に**NTP Configuration**を選択します。
3. **Start NTP Daemon**セクションで、**Now and on Boot**を選択します。
4. **Configuration Source**に**Dynamic**を選択します。

5. 必要に応じてNTPサーバーを追加します。NTP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。
6. /etc/chrony.conf に以下の行が存在する場合は、削除またはコメントアウトします。

   `include /etc/chrony.d/*.conf`

   chrony.conf を編集した後、`chronyd` サービスを再起動します。

   ```
   sudo systemctl restart chronyd.service
   ```

SUSE 12.5 の場合:

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 フィールドは変更せずにそのままにします。
8. NTP サービスに到達可能であることを確認するために、**Test** をクリックします。
9. 一連のウィンドウで **OK** をクリックして、変更を保存します。

> **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可能性があります。詳細については、解決策 を参照してください。

ステップ 1d: Linux VDA 依存パッケージをインストールする

SUSE Linux Enterprise 用の Linux VDA ソフトウェアは、以下のパッケージに依存します。

- PostgreSQL10サーバー 10.12 以降
- Open Motif 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。

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ために、インストール後の手順が必要です。

```
sudo systemctl enable postgresql



ux virtual masudo 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、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するために、以下のコマンドを実行します。

```
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VM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基づいて、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。他のサーバーとのクロックのずれを防ぐため、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。

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 - 時刻同期機能が無効になっており、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維持するには、**/etc/sysctl.conf** ファイルを編集し、以下の行を追加します。

`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正確性を確保するため、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、VMware および Citrix Hypervisor とは異なります。VMware および Citrix Hypervisor では、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 およびハイパーバイザーとの問題が発生します。どちらもシステムクロックを同期しようとします。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: 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。

- Samba Winbind (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。

Samba Winbind (サンバ・ウィンバインド)

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必要があり、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 を起動し、**Network Services** を選択してから **Windows Domain Membership** を選択します。
2. 次の変更を行います。
   - **Domain or Workgroup** 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SMB情報も使用する**」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する**」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ます。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。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、「**インストール**」をクリックします。

YaSTは、これらの変更には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。マシンの再起動をお勧めします。

```
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。ルートユーザーとして、**/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検証するには、次のコマンドを実行します。

```
sudo net ads info
```

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調べるには、次を使用します。

```
sudo net ads status
```

ユーザー認証の検証

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
```

id -u コマンドによって返される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、手順6：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QUESTを構成する

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

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
```

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）。

ドメイン参加後、Linuxマシンを再起動します。

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 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。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、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
```

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も実行できます。ドメイン参加の確認後、手順6：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場合、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-joined** Linuxマシンがドメイン上にあることを確認するには、次のようにします。

```
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
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

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。

SUSEでSSSDをセットアップするには、次の手順を完了します。

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 の構成を確認する
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、ドメインへの参加や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** の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 Kerberos レルム名（例: example.com）です。**REALM** は大文字の Kerberos レルム名（例: EXAMPLE.COM）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 Active Directory ドメインの短い NetBIOS 名（例: EXAMPLE）です。

5. /etc/nsswitch.conf ファイルの **passwd** および **group** エントリを変更して、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。

   - SUSE 15.3およびSUSE 15.2の場合:

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

   - SUSE 12.5の場合:

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

   user は、ドメインにコンピューターを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てください。

```
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 true に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングします。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。Linux セッションでの無効なログオンエラーを防ぐために、**ad_gpo_access_control** は **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. 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
   ```

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

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して実行できます。ドメイン参加の確認後、手順6：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/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

例:

```
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

例:

```
sh pbis-open-9.1.0.551.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6：Linux VDAをインストールするに進みます。

手順4：.NET Runtime 6.0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 Runtime 6.0をインストールします。

.NET Runtime 6.0 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。

ステップ 5: Linux VDA パッケージをダウンロードする

Citrix Virtual Apps and Desktops™ のダウンロードページに移動します。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。

ステップ 6: Linux VDA をインストールする

ステップ 6a: 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リリース以外の古い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、`systemctl 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することもできます。

ステップ 6b: Linux VDA をインストールする

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 15の場合:

```
sudo zypper install XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

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。その前に、以下の依存関係を解決してください。

SUSE 15の場合:

```
sudo rpm -i XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

SUSE 12の場合:

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

ステップ6c: 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の下にある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 15の場合:**

```
sudo rpm -U XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

SUSE 12の場合:

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

SUSE 15のRPM依存関係リスト:

```
postgresql >= 13


postgresql-server >= 13


postgresql-jdbc >= 9.4


java-11-openjdk >= 11


ImageMagick >= 7.0


dbus-1 >= 1.12.2


dbus-1-x11 >= 1.12.2


xorg-x11 >= 7.6_1


libXpm4 >= 3.5.12


libXrandr2 >= 1.5.1
```

```
libXtst6 >= 1.2.3

motif >= 2.3.4

pam >= 1.3.0

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.4

cups >= 2.2

cups-filters >= 1.25

libxml2-2 >= 2.9

libmspack0 >= 0.6
```

```
ibus >= 1.5

libtcmalloc4 >= 2.5

libcap-progs >= 2.26

mozilla-nss-tools >= 3.53.1

libpython2_7-1_0 >= 2.7
```

SUSE 12のRPM依存関係リスト:

```
libQt5Core5 >= 5.5~

mozilla-nss-tools >= 3.47.1

postgresql-server >= 10.12

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

libmspack0 >= 0.4

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。

ステップ7: NVIDIA GRIDドライバーのインストール

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NVIDIA GRIDドライバーをインストールする必要があります。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、以下のガイドを参照してください。

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、以下の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバーをインストールします。

ステップ8：Linux VDAを構成する

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。ロケールがOSで利用できない場合は、**sudo locale-gen en_US.UTF-8**コマンドを実行します。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**コマンドを実行します。

> 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。必要なすべての変数が存在する場合、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の場合、必要なポート（デフォルトではポート80および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・オーセンティケーション・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 ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP をクエリします。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'** – 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で区切られた 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に .NET Runtime 6.0 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。

  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ために、`.xsession` ファイルを編集します。

  - **SUSE 15 の MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

  - **SUSE 15 および SUSE 12.5 の GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic
```

```
exec gnome-session --session=gnome-classic

fi
```

c. 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 のリッスン用ソケットポート。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


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。

あるいは、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 \

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。

ステップ9：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、**XDPing**を参照してください。

ステップ10：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` および `ctxhdx` サービスを停止する前に、`systemctl 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 /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 11: 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、「マシンカタログを作成する」および「マシンカタログを管理する」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:

Active Directoryドメインからマシンを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12：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 2203」を参照してください。

Ubuntu に Linux VDA を手動でインストールする

> **重要:**

> 新規インストールの場合、迅速なインストールには easy install を使用することをお勧めします。Easy install は、時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: ネットワーク構成を確認する

ネットワークが正しく接続され、構成されていることを確認してください。たとえば、Linux VDA で DNS サーバーを構成する必要があります。

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、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`

に変更します。

`hosts: files dns`

ステップ1f: 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1g: クロック同期を設定する (chrony)

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 (VM) 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、リモート時刻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
```

ルートユーザーとして、**/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、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**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: PostgreSQLをインストールする

Linux VDAでは、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1i: Motifをインストールする

```
sudo apt-get install -y libxm4
```

ステップ1j: 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VM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基づいて、以下の変更を行います。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に関する問題が発生します。両方ともシステムクロックを管理しようとします。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。

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 - 時刻同期機能は有効であり、無効にする必要があります。
- 1 - 時刻同期機能は無効であり、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統合サービス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するには、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との間で問題が発生します。両方ともシステムクロックを同期しようとします。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**のチェックを外します。

ステップ 3：Linux 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
```

WINBINDデーモンをマシン起動時に開始するように有効にする

Winbindデーモンは、マシン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、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: 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:

`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

Linux VDAで使用するために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
```

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

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
```

マシンのアカウントの詳細を調べるには、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
```

The domain specified here is the AD domain name, not the Kerberos realm name. For the bash shell, the backslash (\\) character must be escaped with another backslash. This command returns a message indicating success or failure.

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が有効になっていて正しく動作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
```

GnomeまたはKDEコンソールに直接ログオンすることで、同様のテストを実行できます。ドメイン参加の検証後、手順6：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`に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と仮定します。

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にも同様に適用されます。

LINUX VDAでのQUESTの構成

SELinuxポリシーの適用を回避する
RHELのデフォルト環境では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** の設定を変更します。

`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 `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）。

ドメイン参加後、Linuxマシンを再起動します。

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。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
```

ドメイン参加の確認後、手順6：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** 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、手順6：Linux VDAのインストールに進みます。

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、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**に関する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、次の手順を実行します。**

1. **adcli**をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. **adcli**で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、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. ドメインメンバーシップを確認します。

Ubuntu 20.04マシンでは、`adcli testjoin`コマンドを実行します。

Ubuntu 18.04マシンでは、`sudo klist -ket`コマンドを実行します。**adcli**ツールの機能は限られており、マシンがドメインに参加しているかどうかをテストする方法は提供されていません。最善の代替策は、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ことです。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、次の手順を実行します。**

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 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 の構成変更が必要です。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、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 ドメイン名を設定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 デーモンの開始

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 の構成

次のコマンドを実行し、**SSS authentication** と **Create home directory on login** 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 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。

- 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、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。シェル置換を防ぐために、ドル記号 ($) はバックスラッシュ (\\) 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、以下を使用して確認します。

```
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た Kerberos チケットがそのユーザーに対して正しく、有効期限が切れ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たことを確認します。

```
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 6: 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 **アクティブディレクトリ** ユーザーアカウントが必要です。

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

「**ユーザー**」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。「**ドメイン名**」は、たとえばexample.comのようなドメインのDNS名です。

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。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6：Linux VDAのインストールに進みます。

手順4：.NET Runtime 6.0のインストール

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 Runtime 6.0をインストールします。

.NET Runtime 6.0のインストール後、**which dotnet**コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NET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。

手順5：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。

ステップ 6: Linux VDA をインストールする

ステップ 6a: Linux VDA をインストールする

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 20.04 の Debian 依存関係リスト:

```
libqt5widgets5 >= 5.7~


ibus >= 1.5


libsasl2-modules-gssapi-mit >= 2.1.~


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:

```
libqt5widgets5 >= 5.7~

libmspack0 >= 0.6

ibus >= 1.5

libnss3-tools >= 2:3.35

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
```

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 6b: 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 7: 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 NVIDIA GRID ドライバーをインストールする必要があります。

特定のハイパーバイザーに NVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、次のガイドを参照してください。

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールおよび構成するには、次の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバーをインストールします。

ステップ8：Linux VDAを構成する

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

> 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

プロンプトに従って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 (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。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 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 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・ウィンバインド
  - 2 – クエスト・オーセンティケーション・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 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 をクエリします。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 Group Policy` 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で区切られた FAS サーバーのリストを指定できます。シーケンスは `AD Group Policy` 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 Runtime 6.0 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。

  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。

     - **MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

c. 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、**-E** オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。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。

あるいは、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 9: XDPing を実行する

Linux VDA 環境で一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping` を実行します。詳細については、「XDPing」を参照してください。

ステップ 10: 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、`systemctl 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。

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 11: 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示唆されます。**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示唆されます。

> **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12：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 2203」を参照してください。

Debian に Linux VDA を手動でインストールする

> **重要:**

> 新規インストールの場合、迅速なインストールのために **easy install** を使用することをお勧めします。Easy install は時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: ネットワーク構成を確認する

ネットワークが正しく接続され、構成されていることを確認してください。たとえば、Linux VDA で DNS サーバーを構成するには、次のコマンドを実行します。

```
echo "nameserver <dns ip address>" > "/etc/resolv.conf.custom"

echo "#!/bin/bash

    #

    # Description: script to override default resolv.conf file

    # with customized file.

    cp -f /etc/resolv.conf.custom /etc/resolv.conf">"/etc/NetworkManager/
dispatcher.d/15-resolv"



chmod +x /etc/NetworkManager/dispatcher.d/15-resolv
```

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、FQDN とホスト名を最初の 2 つのエントリとして含むように変更します。

```
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。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

マシンを再起動し、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`

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: クロック同期を構成する (chrony)

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 (VM) 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、リモート時刻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
```

ルートユーザーとして、**/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、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: 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1j: PostgreSQLのインストール

Linux VDAは、Debian上でPostgreSQLバージョン11を必要とします。

```
sudo apt-get update



sudo apt-get install -y postgresql



sudo apt-get install -y libpostgresql-jdbc-java
```

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VM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基づいて、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の両方で問題が発生します。両方がシステムクロックを管理しようとします。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。

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を書き込ん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ため、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や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との間で問題が発生します。両方ともシステムクロックを同期しようとします。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。この場合は、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：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(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/debian.html#samba-winbind)
- Quest 認証サービス(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview/manual-installation-overview/debian.html#quest-authentication-service)
- **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

vim /etc/samba/smb.conf コマンドを実行して **/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

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 <Kerberos realm name in uppercase> -U <domain user with
permission to add computers to the domain>
```

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。

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
```

マシンのアカウントの詳細を調べるには、次を使用します。

```
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、手順6：Linux VDAのインストール に進みます。

> **ヒント:**
>
> ドメインアカウントでログオンしたときにユーザー認証に成功しても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：

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も同様に適用されます。

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で使用される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。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や、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
```

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）。

ドメイン参加後、Linuxマシンを再起動します。

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。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、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
```

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6：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**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かLinuxかにかかわらず、すべてのVDAマシンがActive Directory内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、手順6：Linux VDAのインストールに進みます。

SSSD

KERBEROS の設定

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`
>
> `rdns = 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。

ドメインへの参加

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** に関する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。

   古いシステムキー配布ファイル (keytab file) 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. システムキー配布ファイル (keytab) を確認します。

システムキー配布ファイル (keytab file) 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、次の手順を完了してください。**

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. ドメインに **Samba** で参加します。

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
```

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

必要なパッケージをインストールまたは更新する:

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、更新が推奨されます:

```
sudo apt-get install --only-upgrade sssd
```

SSSD の構成

SSSD デーモンを開始する前に、SSSD の構成変更が必要です。SSSD の一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルが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 にマッピングします。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。

構成設定の詳細については、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 デーモンの開始

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 の構成

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つことを要求します。

- 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いません。認証は PAM 経由でのみ実行できます。

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たことを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDE または Gnome ディスプレイマネージャーにログオンして実行できます。ドメイン参加の確認後、手順 6: 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。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、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 6: Linux VDA のインストール に進みます。

手順 4: .NET Runtime 6.0 のインストール

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 Runtime 6.0 をインストールします。

.NET Runtime 6.0 のインストール後、**which dotnet** コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NET 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。

手順 5: Linux VDA パッケージのダウンロード

Citrix Virtual Apps and Desktops™ のダウンロードページに移動します。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 6: Linux VDA のインストール

手順 6a: 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 10.9のDebian依存関係リスト:

```
libqt5widgets5          >= 5.5~

ibus                    >= 1.5

postgresql              >= 11

libpostgresql-jdbc-java >= 42.2

openjdk-11-jdk           >= 11

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」(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/system-requirements.html)を参照してください。

ステップ6b: 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の下にある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7: 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NVIDIA GRIDドライバーをインストールする必要があります。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、次のガイドを参照してください。

- シトリックス ハイパーバイザー

- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、次の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバーをインストールします。

ステップ8：Linux VDAを構成する

> **注:**
>
> ランタイム環境をセットアップする前に、OSに**en_US.UTF-8**ロケールが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**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集して**# en_US.UTF-8 UTF-8**行のコメントを解除し、**sudo locale-gen**コマンドを実行します。

> 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

プロンプトが表示される質問に従って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に登録するために、スペースで区切られたDelivery Controllerの完全修飾ドメイン名（FQDN）の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・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（例：DC=mycompany,DC=com）に設定された検索ベースを介して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\>**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。
- **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 Runtime 6.0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。

  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. `.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。

     - **MATEデスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

◦ **GNOMEデスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

c. 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サービスを開始するかどうかを決定します。デフォルトではYに設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutのリッスン用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との通信用ポートです。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、**-E**オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。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スクリプトと**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ソフトウェアのアンインストール

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設定された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は、PostgreSQLを含む依存パッケージの削除については説明していません。

ステップ9：XDPingを実行する

`sudo /opt/Citrix/VDA/bin/xdping`を実行して、Linux VDA環境における一般的な構成の問題を確認します。詳細については、**XDPing**を参照してください。

ステップ10：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、以下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、モニターサービスデーモンを停止するために `systemctl 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 の状態を確認する:

Linux VDA サービスの実行状態を確認するには:

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

手順 11: 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示唆されます。**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示唆されます。

ヒント:

Active Directoryドメインからマシンを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12：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 2203」を参照してください。

Linux VDA の設定

このセクションでは、Linux VDA の機能について、機能の説明、構成、トラブルシューティングを含めて詳述します。

管理

このセクションには、次のトピックが含まれています。

- CEIP
- HDX インサイト
- Citrix Telemetry Serviceとの統合
- Linux VDAのセルフアップデート
- Linux VMおよびLinuxセッションのメトリック
- ログ収集
- セッションシャドウイング
- モニターサービスデーモン
- トラブルシューティング

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、Citrix製品の品質とパフォーマンスを向上させるために、匿名の統計情報と使用状況情報が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、CEIPに自動的に参加します。データの最初のアップロードは、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

指定されていない場合、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 = 1):

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は無効になります。

クライアントで次のコマンドを実行してGAを有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "GASwitch" -d "0"
```

- **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

次のコマンドを実行してこのパスを設定できます。

```
/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する
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。
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**	HDX 3D Pro がマシンで有効になっているかどうかを示すテキスト文字列。

データポイント	キー名	説明
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 の主要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 タイプ	gpu_type	マシンのGPUタイプを示す
CPUコア	**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した時点でマシン上で起動された（ログオンまたは再接続された）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タイプを示すテキスト文字列: デスクトップまたは `<appname>`

データポイント	キー名	説明
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を持つことができます
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™ アプリの種類を示す整数
Receiverクライアントバージョン	**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
Gfx 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示す整数
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

データポイント	キー名	説明
		リダイレクトが使用された回数を示す整数

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。

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

原因は2つ考えられます。

- 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。

アプリケーションの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実行されていることを確認してください。

Citrix Telemetry Serviceとの統合

Linux VDAソフトウェアにCitrix Telemetry Service (`ctxtelemetry`) が統合されているため、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に関するログを収集できます。

Citrix Scoutの画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/citrix-scout.png)

> **注:**
>
> Linux VDA 1912およびそれ以前のバージョンからアップグレードした後、Citrix Telemetry Service (`ctxtelemetry`) の変数を構成するために、/opt/Citrix/VDA/sbin/ctxsetup.shを再実行する必要があります。変数の詳細については、「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ときに、デフォルトのポートを使用するか、次の変数を使用し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
   ```

   ソケットポートの画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/socket-port.png)

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 を使用している場合は、ポート 7503 を有効にするには、**sudo ufw allow 7503** コマンドを実行します。

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
   ```

デバッグモードを有効にするイメージ (/en-us/linux-virtual-delivery-agent/2203-ltsr/media/enable-debug-mode.png)

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。

   テレメトリを停止するイメージ (/en-us/linux-virtual-delivery-agent/2203-ltsr/media/stop-telemetry.png)

   この例では、Citrix Telemetry Service を停止するには、次のコマンドを実行します。

   ```
   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/ で telemetry-debug.log を見つけます。

サービス待機時間

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、ほとんどリソースを使用しません。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び停止します。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。これらのテストにより、要件が満たされていることが確認されます。マシンのテストが失敗した場合、Scout は推奨される是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。

Azure を介した Linux VDA の自己更新

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。これは、Citrix DaaS Standard for Azure (旧称 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 の管理者権限は必要ありません。詳細については、「Citrix DaaS 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、次の手順を実行します。

ステップ 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

ステップ 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 (匿名読み取りアクセスのみ)** 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、各自己更新後にコンテナのアクセスレベルを **Private (匿名アクセスなし)** に設定してください。

ステップ 1b: VDA 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": "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次の 3 つの項目が含まれます。

- **「TargetOS」**：「RHEL7_9」（RHEL 7、CentOS 7、Amazon Linux 2 用）、「RHEL8_3」、「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 パッケージの完全な名前。

- 「**PackageHash**」：`shasum -a 256 <pkgname>` コマンドを使用して計算する SHA-256 値。

ステップ 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。

ステップ 2: マスターイメージまたは各 VDA で自己更新機能を有効にする

デフォルトでは、自己更新は無効になっています。Citrix DaaS 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実行する必要があります。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。

自己更新を有効にするには、HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\SelfUpdate の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
```

以下の表は、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に設定できます。
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直後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の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あまり速くなく、アップデートが緊急では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`コマンド用の一部のNSSライブラリのリンクが不足しています。証明書の最小限の権限を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**

の設定に従って更新を実行します。オンプレミス展開の場合、`ctxmonitorservice`を直接再起動して更新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 DaaS 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。更新が失敗した場合、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。
- 更新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。
- 更新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。
- 更新には通常、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。

Linux VMとLinuxセッションのメトリック

次の表に、Linux VMおよびLinuxセッションで利用可能なメトリックの一部を示します。

メトリック	必要な最小VDAバージョン	説明	備考
ログオン期間	2109	これは、ユーザーがCitrix Workspace™アプリから接続してからセッションが使用可能になるまでのログオンプロセスを測定するものです。このセッションメトリックを表示するには、Citrix DaaSの**Monitor**タブに移動します。Monitorは、Citrix Virtual Apps and Desktops™の	Monitorでのみ利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
		Current ReleaseおよびLTSR展開を監視およびトラブルシューティングするためのDirectorコンソールとして利用できます。**Monitor**タブで、**Average Logon Duration**セクションの**View Historical Trend**をクリックします。**Logon Performance**ページで、フィルター条件を設定し、**Apply**をクリックしてメトリックを視覚化します。	
セッションの自動再接続数	2109	セッション内の自動再接続数を表示するには、**Trends**ビューにアクセスします。条件を設定し、**Apply**をクリックして検索結果を絞り込みます。**Session Auto Reconnect Count**列には、セッション内の自動再接続数が表示されます。**自動再接続**は、**セッションの信頼性**または**クライアントの自動再接続**ポリシーが有効な場合に有効になります。セッションの再接続とポリシーの詳細については、次の記事	Citrix DirectorとMonitorの両方で利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
		を参照してください。 セッション クライアント自動再接続ポリシー設定 セッションの信頼性ポリシー設定	
アイドル時間	2103	このメトリックにアクセスするには、**Filters > Sessions > All Sessions**を選択して**All Sessions**ページを開きます。	Citrix DirectorとMonitorの両方で利用可能です。
Linux VMのメトリック	2103	Linux VMでは、CPUコア数、メモリサイズ、ハードディスク容量、現在のCPUおよびメモリ使用率、過去のCPUおよびメモリ使用率などのメトリックが利用可能です。	Citrix DirectorとMonitorの両方で利用可能です。
プロトコル	1909	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。	Citrix DirectorとMonitorの両方で利用可能です。
ICA® RTT	1903	ICAラウンドトリップタイム (RTT) は、キーを押してからエンドポイントに応答が表示されるまでの経過時間です。ICA	Citrix DirectorとMonitorの両方で利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
		RTTメトリックを取得するには、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。	

ログ収集

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。トレースオン機能は、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含まれるようになりました。デフォルトでは、Linux VDAをインストールして構成すると、`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

トレースオンは、**/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`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`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**warning**（大文字と小文字を区別しない）に設定されています。

異なるコンポーネントに設定されているログレベルを表示するには、次のコマンドを実行します。

```
setlog levels
```

ログレベル（Disabled、Inherited、Verbose、Information、Warnings、Errors、Fatal Errorsを含む）を設定するには、次のコマンドを実行します。

```
setlog level <class> [<level>]
```

ログレベル	コマンドパラメーター（大文字と小文字を区別しない）
無効	なし
継承済み	継承
詳細	詳細
情報	情報
警告	警告
エラー	エラー
致命的なエラー	致命的

`<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、これを all に設定します。例:

```
setlog level all 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 **/var/xdl/.ctxlog** ファイルを使用して構成されます。このファイルはrootユーザーのみが作成できます。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ユーザーは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、任意のまたは悪意のある構成が `ctxlogd` され、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。

ログ収集

ログを収集するには、`bash /opt/Citrix/VDA/bin/xdlcollect.sh` コマンドを実行します。ログの収集に使用される `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。

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` Bashスクリプトは、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。同意すると、アップロード完了後に `xdlcollect` がupload_IDを返します。アップロードしても、圧縮ログファイルはローカルマシンから削除されません。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。

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している`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

セッションのシャドウイングにより、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ます。

> **注:**
>
> この機能を使用するには、`Citrix Director` 7.16以降を使用してください。

インストールと構成

依存関係

セッションの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。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してください。

RHEL 7.xおよびAmazon Linux2の場合：

`python-websockify` と `x11vnc` (`x11vnc` バージョン0.9.13以降) をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo yum install x11vnc
```

`python-websockify` と `x11vnc` を解決するには、RHEL 7.xでExtra Packages for Enterprise Linux (EPEL) およびオプションのRPMリポジトリを有効にします。

- EPEL

  EPELリポジトリは `x11vnc` 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。

  ```
  yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  ```

- オプションのRPM

  `x11vnc` の一部の依存関係パッケージをインストールするために、オプションの `RPMs` 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。

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

RHEL 8.xおよびRocky Linux 8の場合：

`python-websockify` と `x11vnc` (`x11vnc` バージョン0.9.13以降) をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo yum install x11vnc
```

`x11vnc` を解決するには、EPELおよびCodeReady Linux Builderリポジトリを有効にします。

```
dnf install -y --nogpgcheck https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm



subscription-manager repos --enable "codeready-builder  -for-rhel-8-x86_64-rpms"
```

Ubuntu の場合:

`python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

SUSE の場合:

`python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo zypper install x11vnc
```

Debian の場合:

`python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

ポート

セッションシャドウ機能は、Linux VDAから`Citrix Director`への接続を確立するために、6001～6099の範囲から利用可能なポートを自動的に選択します。そのため、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接続では、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異なります。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://`を使用してLinux VDAセッションシャドウイングを行う場合のセキュリティ上の責任を負いません。

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。代わりに、ドメイン全体にワイルドカード証明書を使用できます。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。

Linux VDAと通信する各Citrix Directorクライアントにもルート証明書が必要です。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。

以下のCAからサーバー証明書とクライアント証明書をインストールできます。

- 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）
- オペレーティングシステムにバンドルされていないCA

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:

- サーバー証明書のコモンネームは、Linux VDA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

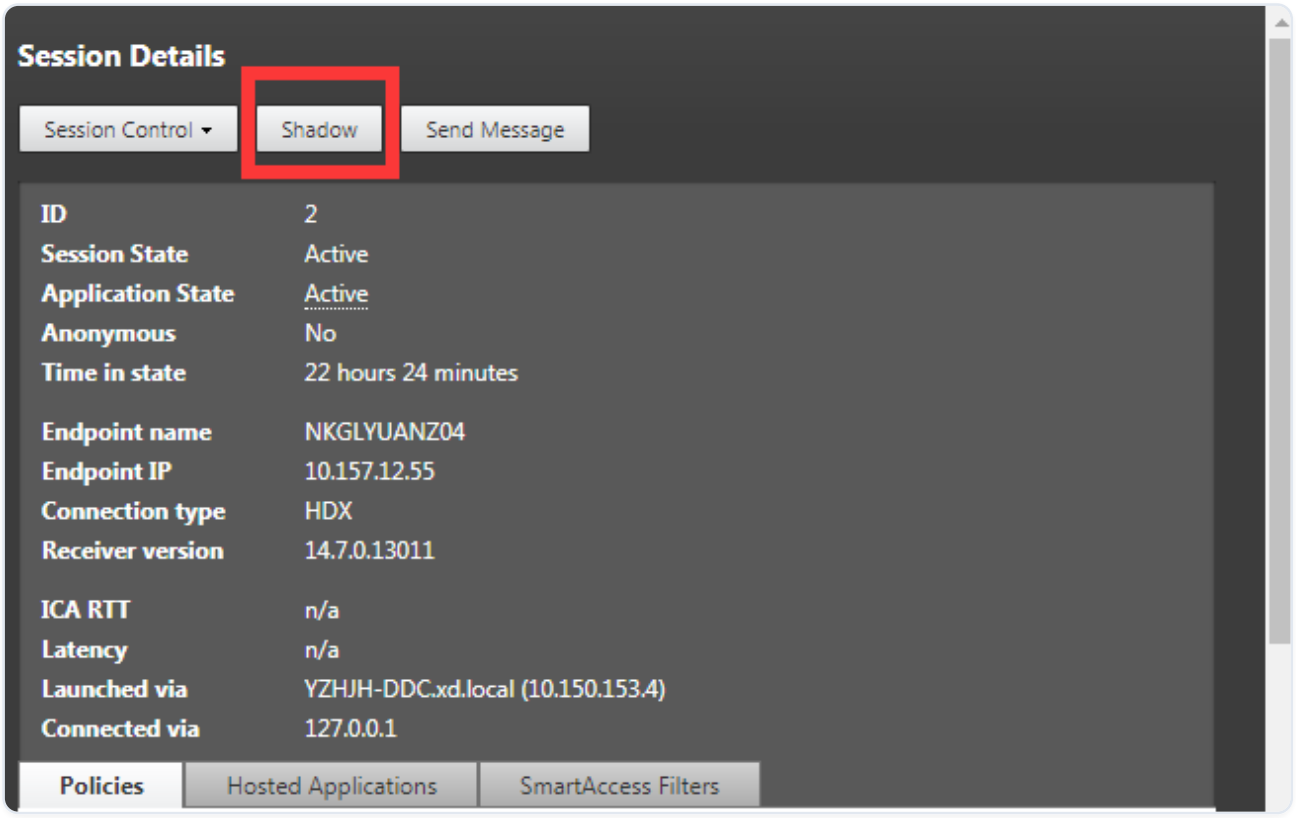
セッションシャドウイングはIISと同じレジストリベースの証明書ストアを使用するため、IISまたはMicrosoft管理コンソール（MMC）の証明書スナップインを使用してルート証明書をインストールできます。CAから証明書を受け取ったら、IISでWebサーバー証明書ウィザードを再起動すると、ウィザードが証明書をインストールします。または、MMCを使用してコンピューター上の証明書を表示およびインポートし、証明書をスタンドアロンスナップインとして追加することもできます。Internet ExplorerとGoogle Chromeは、デフォルトでオペレーティングシステムにインストールされている証明書をインポートします。Mozilla Firefoxの場合、証明書マネージャーの**機関**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します

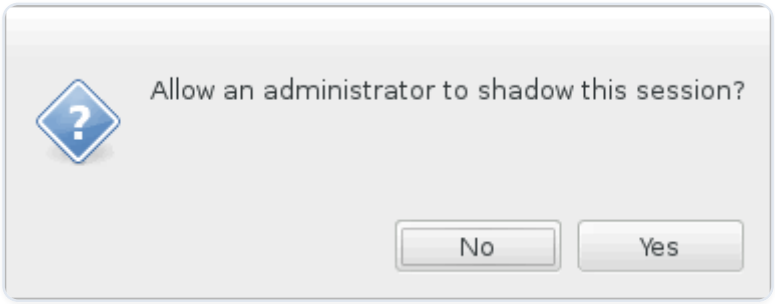
サーバー証明書には「shadowingcert.*」、キーファイルには「shadowingkey.*」という名前を付けます（* は形式を示します。例：shadowingcert.pem、shadowingkey.key）。サーバー証明書とキーファイルはパス **/etc/xdl/shadowingssl** の下に置き、制限された権限で適切に保護してください。名前またはパスが正しくないと、Linux VDAは特定の証明書またはキーファイルを見つけることができず、`Citrix Director` との接続障害が発生します。

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 **Shadow** 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。



接続が初期化されると、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のシャドウイング許可を求めます。



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イング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のドキュメント 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経由で接続する場合でも、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ついて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ず、書き込みや制御の権限はありません。

- 管理者が `Citrix Director` から **Shadow** をクリックすると、ユーザーにセッションのシャドウイング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イングできます。
- 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つのセッションは1人の管理者のみがシャドウできます。たとえば、管理者Bが管理者Aがシャドウしているセッションに対してシャドウイング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のシャドウイング接続は停止し、管理者Bのために新しいシャドウイング接続が確立されます。管理者が同じセッションに対して別のシャドウイング要求を送信した場合も、新しいシャドウイング接続を確立できます。
- 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ます。

Citrix Director クライアント上

ブラウザの開発者ツールを使用して、**コンソール**タブで出力ログを確認します。または、**ネットワーク**タブで ShadowLinuxSession API の応答を確認します。ユーザー権限取得の確認が表示されても接続の確立に失敗する場合は、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 上

シャドウイング要求に応じて、ユーザー権限取得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 および 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 の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」を次のように「trace」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。

接続の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 シナリオの場合は、証明書とキーファイルに適切に名前を付け、正しいパスに配置していることを確認します。
3. 新しいシャドウイング要求のために、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。

モニターサービスデーモン

モニター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骸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

モニターサービスデーモンは、VDAを起動すると自動的に起動します。

管理者権限で **/opt/Citrix/VDA/sbin** の下にある **scanningpolicy.conf**、**rulesets.conf**、および **whitelist.conf** ファイルを通じて機能を構成できます。

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モニターサービスデーモンを再起動します。

```
service ctxmonitorservice restart
```

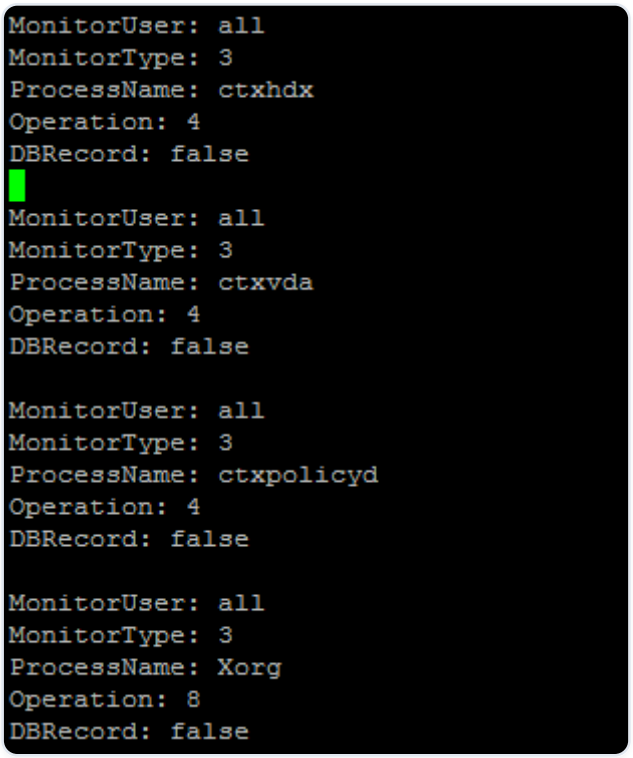
- **scanningpolicy.conf**

  この構成ファイルは、モニター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: 偽
  - ReportAlarm: 偽

- **rulesets.conf**

  この構成ファイルは、監視するターゲットサービスを指定します。次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、次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することはできず、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アします。)
    - **DBRecord**: オフ

- **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。

  ホワイトリストを構成するには、**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、次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(/en-us/linux-virtual-delivery-agent/2203-ltsr/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。

トラブルシューティング

この記事では、**XDPing** を使用したトラブルシューティングと、**ctxsdcutil** ユーティリティを使用したセッションデータのクエリ方法について説明します。

エックスディーピング

Linux **XDPing** ツールはコマンドラインアプリケーションです。これは、Linux VDA 環境における一般的な構成の問題を自動的にチェックします。

Linux **XDPing** ツールは、システム上で150を超える個別のテストを実行し、そ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がないか確認します。
- 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 HyperV
  - 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用のLDAP検索環境を確認します。
- Active Directory統合が適切に設定され、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) の構成を確認します。
- 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**仮想環境を作成するの手順に従って手動で作成してください。

> 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`

> **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、`-h`オプションを使用します。

```
sudo /opt/Citrix/VDA/bin/xdping -h
```

全てのテスト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を実行するには、たとえば、時間テストとKerberosテストなど、`-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
```

出力例

以下は、Kerberosテストを実行したときの出力例です。

Kerberosテストの出力例の最初の部分(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/kerberos-test-sample-output-1.png)

Kerberosテストの出力例の2番目の部分(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/kerberos-test-sample-output-2.png)

Kerberosテストの出力例の3番目の部分(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/kerberos-test-sample-output-3.png)

セッションデータクエリユーティリティ

各Linux VDAでセッションデータをクエリするために使用できるユーティリティ(`ctxsdcutil`)を提供しています。VDAでホストされているすべてのセッションまたは特定のセッションの次のデータをクエリするには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]`コマンドを実行します。`[-c]`引数は、1秒ごとにデータをクエリすることを意味します。

- **入力セッション帯域幅**
- **出力セッション帯域幅**
- **出力セッション回線速度**

- **待機時間 - 最終記録**
- **ラウンドトリップタイム**
- **出力 ThinWire 帯域幅**
- **出力オーディオ帯域幅**
- **出力プリンター帯域幅**
- **入力ドライブ帯域幅**
- **出力ドライブ帯域幅**

その他

このセクションには、次のトピックが含まれています。

- Citrix Workspace app for HTML5 のサポート
- Python3 仮想環境の作成
- NIS と Active Directory の統合
- IPv6
- LDAPS
- **Xオーソリティ**

HTML5向けCitrix Workspace™アプリのサポート

このリリースから、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の詳細については、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接続を有効にする。

   a. Citrix Studioで、**WebSockets接続**ポリシーを**許可**に設定します。

      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. 設定を有効にするには、VDAで`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: TLS暗号化を有効にして、WebSocket接続を保護することもできます。TLS暗号化の有効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。

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` 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 仮想環境の作成に失敗した場合は、ネットワーク接続なしで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には、管理者権限が必要です。
- `Python3` パッケージのホイールファイルが配置されています。ホイールファイルは `<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、次の手順を実行します。

1. Python3 の依存関係をインストールします。

   Amazon Linux 2 の場合:

   ```
   yum -y install python3 python3-devel krb5-devel gcc
   ```

   RHEL および Rocky Linux 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
libffi-dev
```

SUSE の場合:

```
zypper -n install lsb-release python3-devel python3-setuptools krb5-
devel gcc libffi-devel libopenssl-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、次の信頼できるホストを /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、Rocky Linux、Ubuntu、Debian の各ディストリビューションの場合:

```
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. `XDPing` の依存関係をインストールします。

RHEL、Rocky Linux、Ubuntu、Debian の各ディストリビューションの場合:

```
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
```

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環境に密接に適合します。

UIDおよびGIDプロバイダーとしてADの代わりにNISを使用する場合、ADとNISでアカウント情報（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とは何ですか？

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およびNSSモジュールの両方を提供し、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、次の手順を完了します。

ステップ1：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します。

```
yum –y install ypbind rpcbind oddjob-mkhomedir
```

NISドメインを設定します。

```
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**に追加します。

`{NIS server IP address}  server.nis.domain nis.domain`

`authconfig`でNISを構成します。

```
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サービスを設定します:

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS構成が正しいことを確認します:

```
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します:

```
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。」.

ステップ2: SAMBAを使用してドメインに参加し、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。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。以下はその一部です:

- `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、ホストキータブを作成する:

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Kerberosレルム名の大文字表記であり、**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、上記のコマンドに次の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、**\[Global\]**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。

ステップ3：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 構成の例（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`

ステップ4：NSS/PAMを構成する

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。

ステップ 5: 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。シェル置換を防ぐために、ドル記号 ($) はバックスラッシュ (\\) 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
```

ステップ 6: ユーザー認証の確認

ログオン形式がサポートされているか、およびNSSが機能するかどうかを確認するには、**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、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。こ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です。

```
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でないことを確認します。

```
klist
```

IPv6を構成する

Linux VDAは、Citrix Virtual Apps and Desktopsとの整合性を保つためにIPv6をサポートしています。この機能を使用する際は、次の点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に関する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"
```

OnlyUseIPv6ControllerRegistrationを1に設定して、Linux VDAでIPv6を有効に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。

LDAPS の構成

LDAPS は、LDAP 通信が TLS/SSL を使用して暗号化される Lightweight Directory Access Protocol (LDAP) のセキュアバージョンです。

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 通信は暗号化されません。LDAPS を使用すると、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。

次の 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)/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/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認証局（CA）またはMicrosoft以外のCAのいずれかから、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、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、「サードパーティの認証局でLDAP over SSLを有効にする方法」を参照してください。

多層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。

多層認証局階層を使用してドメインコントローラーのLDAPSを有効にする方法については、「LDAP over SSL (LDAPS) 証明書」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「ルート認証局証明書のエクスポート方法」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、**enable_ldaps.sh**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。**enable_ldaps.sh**スクリプトを非対話モードで実行するには、次の変数を環境にエクスポートします。

```
#CTX_LDAPS_KEYSTORE_PASSWORD=



#CTX_LDAPS_LDAP_SERVERS=
```

- 提供されたルートCA証明書を使用してLDAP over SSL/TLSを有効にするには（LDAPチャネルバインディングをサポート）：

  ```
  /opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
  ```

- SSL/TLSなしでLDAPにフォールバックするには

```
/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向けLDAPSの構成

Linux VDAコンポーネントの他に、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（SSSD、Winbind、Centrify、Questなど）もセキュアな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署名およびシーリングを使用してセキュアな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、セキュアな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上のSSL経由のLDAP）はポート389上の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 manページを参照してください。

ウィンバインド

Winbind LDAPクエリはADSメソッドを使用します。Winbindはポート389でのStartTLSメソッドのみをサポートします。影響を受ける構成ファイルは、**/etc/samba/smb.conf** および **/etc/openldap/ldap.conf**（RHELの場合）、または **/etc/ldap/ldap.conf**（Ubuntuの場合）です。ファイルを次のように変更します。

- smb.conf

  `ldap ssl = start tls`

  `ldap ssl ads = yes` `client ldap sasl wrapping = plain`

- ldap.conf

  `TLS_REQCERT never`

あるいは、SASL GSSAPI署名とシーリングによってセキュアなLDAPを構成することもできますが、これはTLS/SSLと共存できません。SASL暗号化を使用するには、**smb.conf**構成を変更します。

`ldap ssl = off`

`ldap ssl ads = no` `client ldap sasl wrapping = seal`

セントリファイ

Centrifyはポート636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ます。ポートはデフォルトで636です。

- **LDAPSが有効な場合にLinux VDAの登録が失敗する**

  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ます。まずルートCA証明書を確認し、AD/LDAPサーバーと一致していることを確認します。

- **誤ってレジストリが変更された**

  enable_ldaps.shを使用せずに誤ってLDAPS関連のキーが更新された場合、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。**/opt/Citrix/VDA/sbin/enable_ldaps.sh**を実行して強制的に有効にします。

- **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なかった**

  IPアドレスの代わりにAD FQDNを使用します。

- **/opt/Citrix/VDA/sbin/enable_ldaps.shスクリプトを実行してルートCA証明書をインポートできなかった**

  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ます。ほとんどのJava Keytoolでサポートされている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証明書を表示できなかった**

  `/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、パスワードはキーストアを保護するために設定されます。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。

Xオーソリティ

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。

この安全な通信の権限を保護するには、次の2つの方法があります。

- **Xhost**。デフォルトでは、Xhost はローカルホストの XClient のみが XServer と通信することを許可します。リモートの XClient が XServer にアクセスすることを許可する場合、特定のマ

シンで Xhost コマンドを実行して権限を付与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、XServer の認証のために xauth が使用する 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) が起動されると、その特定のディスプレイへの接続を認証するために Cookie が使用されます。

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、次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の XServer に設定します。
- Xorg の Cookie 番号のいずれかを含む `.Xauthority` ファイルを取得します。

Xauthority の構成

リモート X11 ディスプレイのために Linux VDA で **Xauthority** を有効にするには、次の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®セッションを起動した後、Linux VDAはXClient用に`.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、`~/.Xauthority`ファイルがデフォルトで使用されます。

- 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、`.Xauthority`ファイルはログオンユーザーのホームディレクトリに作成されます。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が機能しない場合は、以下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. クッキーが同じ場合は、Linux VDAのIPアドレスと公開デスクトップのディスプレイ番号を使用して、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。

   たとえば、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 + \<ディスプレイ番号\>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。

認証

このセクションには、次のトピックが含まれています。

- ダブルホップシングルサインオン認証(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/authentication/single-sign-on.html)
- フェデレーション認証サービス

- スマートカード(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/authentication/support-for-smart-cards.html)
- セッション起動資格情報(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/authentication/non-sso-authentication.html)
- 未認証セッション(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/authentication/access-by-unauthenticated-users.html)
- 未認証（匿名）ユーザーによるアクセス(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/authentication/access-by-unauthenticated-users.html)

ダブルホップシングルサインオン認証

StoreFrontストアにアクセスするためのユーザー資格情報は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10のAuthManagerモジュールに挿入できます。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やアプリケーションにアクセスする際に、ユーザー資格情報を再度入力する必要がありません。

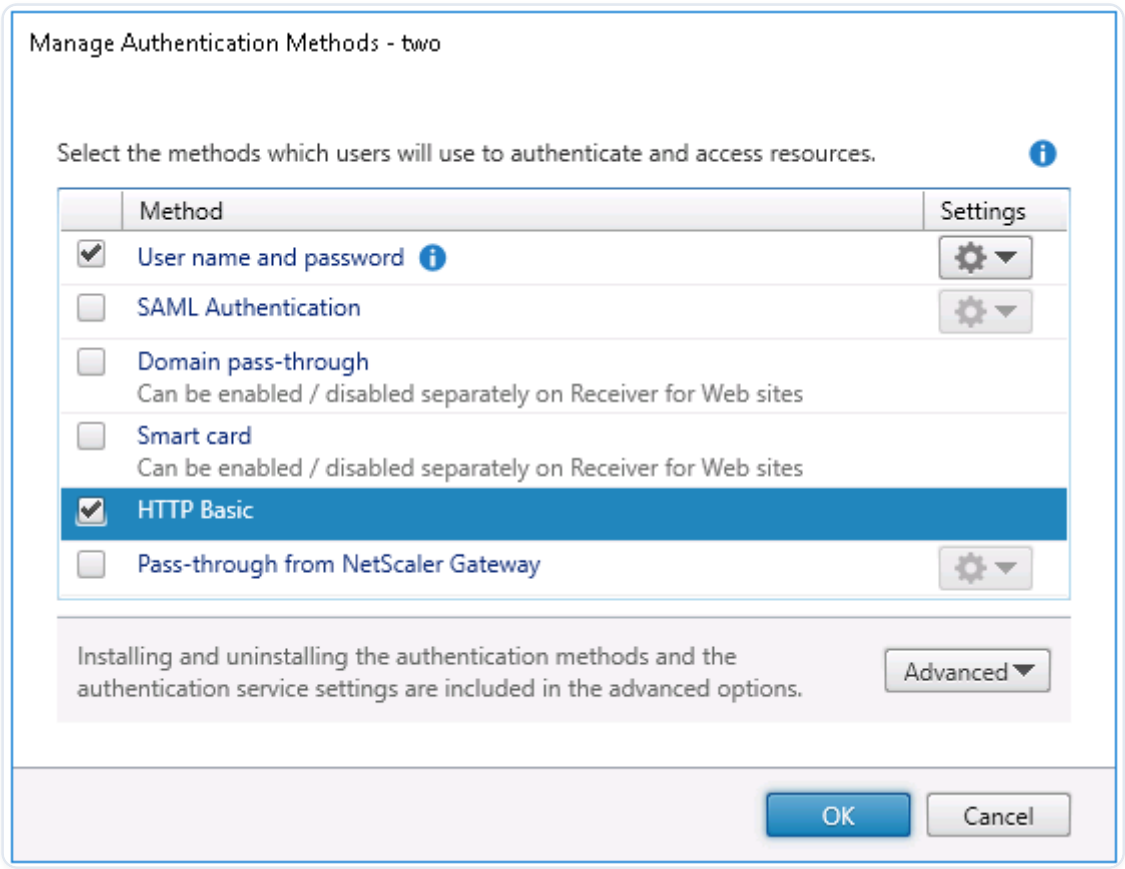
> **注:**
>
> この機能は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10でサポートされています。

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。

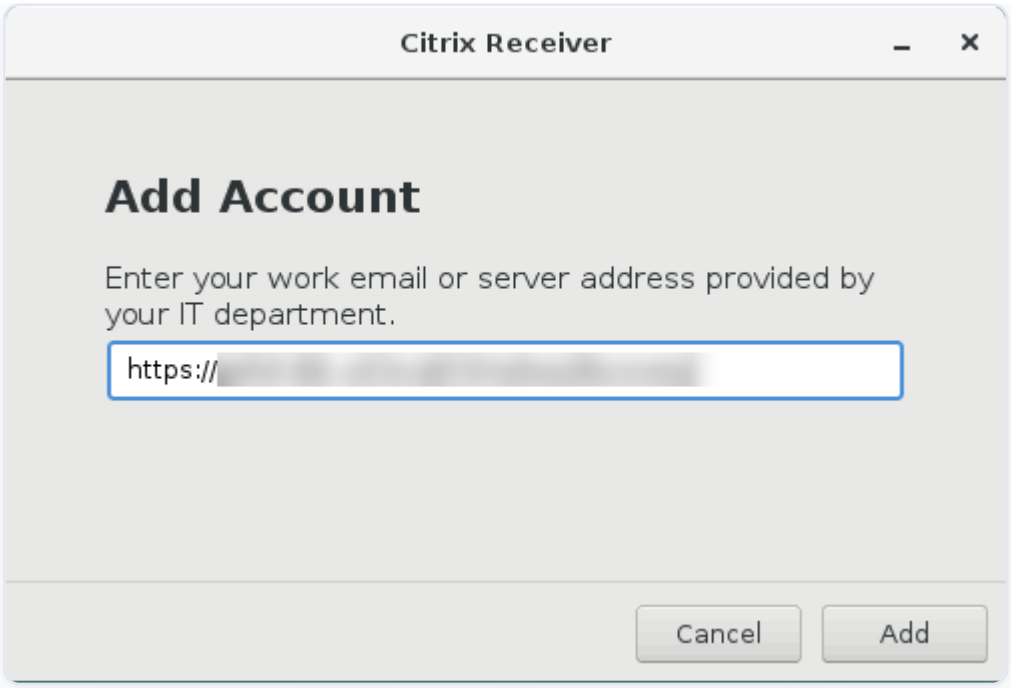
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

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を起動します。

   Citrix Workspace™アプリ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。その後は、以前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。

フェデレーション認証サービスを構成する

Linux VDAにログオンするユーザーを認証するために、フェデレーション認証サービス (FAS) を使用できます。Linux VDAは、FASログオン機能に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転送に短い接続を使用します。

> Linux VDAは、ポート80のみを介してFASサーバーと通信します。

サポートされているディストリビューション

- 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	PBIS	クエスト
アマゾン リナックス 2	はい	はい	はい	はい	いいえ
RHEL 8.10	はい	はい	はい	はい	はい
RHEL 8.8	はい	はい	はい	はい	はい
RHEL 8.6	はい	はい	はい	はい	はい
RHEL 8.4	はい	はい	はい	はい	はい
RHEL 8.3	はい	はい	はい	はい	はい
RHEL 8.2	はい	はい	はい	はい	はい
RHEL 8.1	はい	はい	はい	はい	はい
RHEL 7.9 / CentOS 7.9	はい	はい	はい	はい	はい
RHEL 7.8 / CentOS 7.8	はい	はい	はい	はい	はい
ロッキー・リナックス 8.10/8.8/8.6	はい	はい	いいえ	いいえ	いいえ
SLES 15.3	はい	いいえ	はい	いいえ	はい

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
SLES 15.2	はい	いいえ	はい	いいえ	はい
SLES 12.5	はい	いいえ	はい	いいえ	いいえ
ウブントゥ 20.04	はい	いいえ	はい	いいえ	はい
ウブントゥ 18.04	はい	いいえ	はい	いいえ	はい

Linux VDA で FAS を構成する

FAS サーバーを設定する

新規のLinux VDAインストールでFASを使用するには、ctxinstall.shまたはctxsetup.shを実行するときに、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、代わりにセミコロン区切りのFASサーバーの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には、`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`http://CA-SERVER/certsrv`からDER形式をダウンロードします。

> **注:**
>
> 次のコマンドは、中間証明書を構成する場合にも適用されます。
>
> 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**パスの下に置かないでください。そうしないと、FASはルートCA証明書に対する読み取り権限を持ちません。

ctxfascfg.shを実行する

FASを構成するために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、環境変数が追加されます。

- **CTX_FAS_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis \| quest**: 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が保存されている完全なパスを指定します。
- **CTX_FAS_KDC_HOSTNAME**: PBISを選択した場合のKey Distribution Center (KDC) のホスト名を指定します。
- **CTX_FAS_PKINIT_KDC_HOSTNAME**: PKINIT KDCホスト名を指定します。特に指定がない限り、CTX_FAS_KDC_HOSTNAME と同じです。

適切なActive Directory統合方法を選択し、証明書の正しいパスを入力します（例：`/etc/pki/CA/certs/`）。

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、Federated Authentication Service architectural overviewの記事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セッションが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動作する場合は、**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

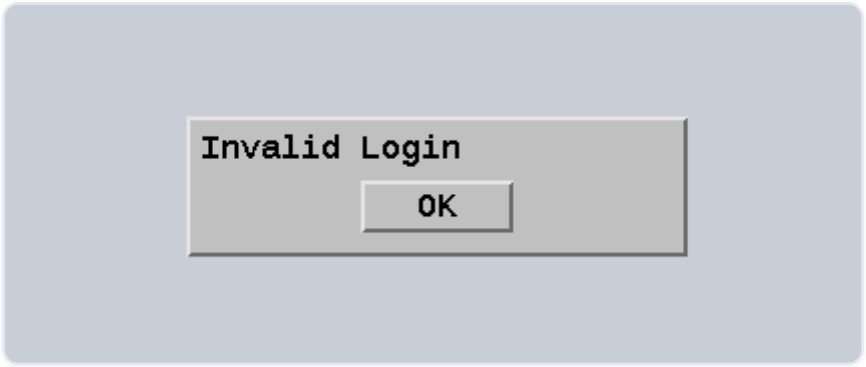
次のコマンドを実行して、Citrixレジストリ値「HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\VirtualDesktopAgent\\Authentication\\UserCredentialService」が\<Your-FAS-Server-List\>に設定されていることを確認します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

既存の設定が正しくない場合は、前述の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証明書構成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

`/etc/krb5.conf` にルート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正しく設定されていることを確認してください。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/

    ......

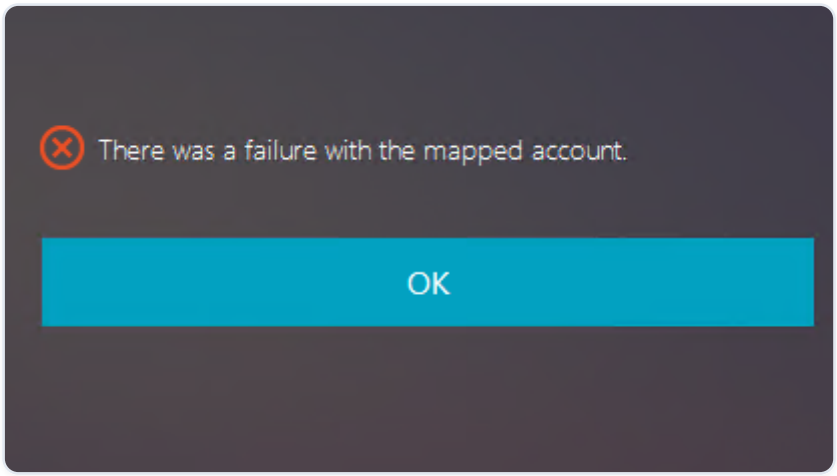
}
```

既存の設定が正しくない場合は、前述の 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を確認してください。

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。



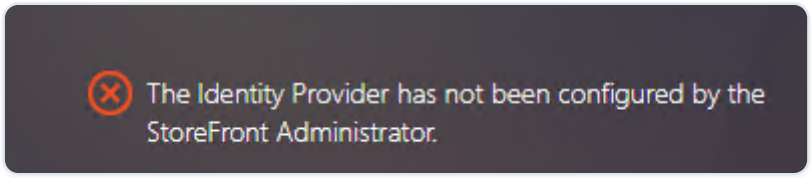
このエラーは、ADFS ユーザーが正常に検証されたものの、AD にシャドウユーザーが構成されていないことを示しています。

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。



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

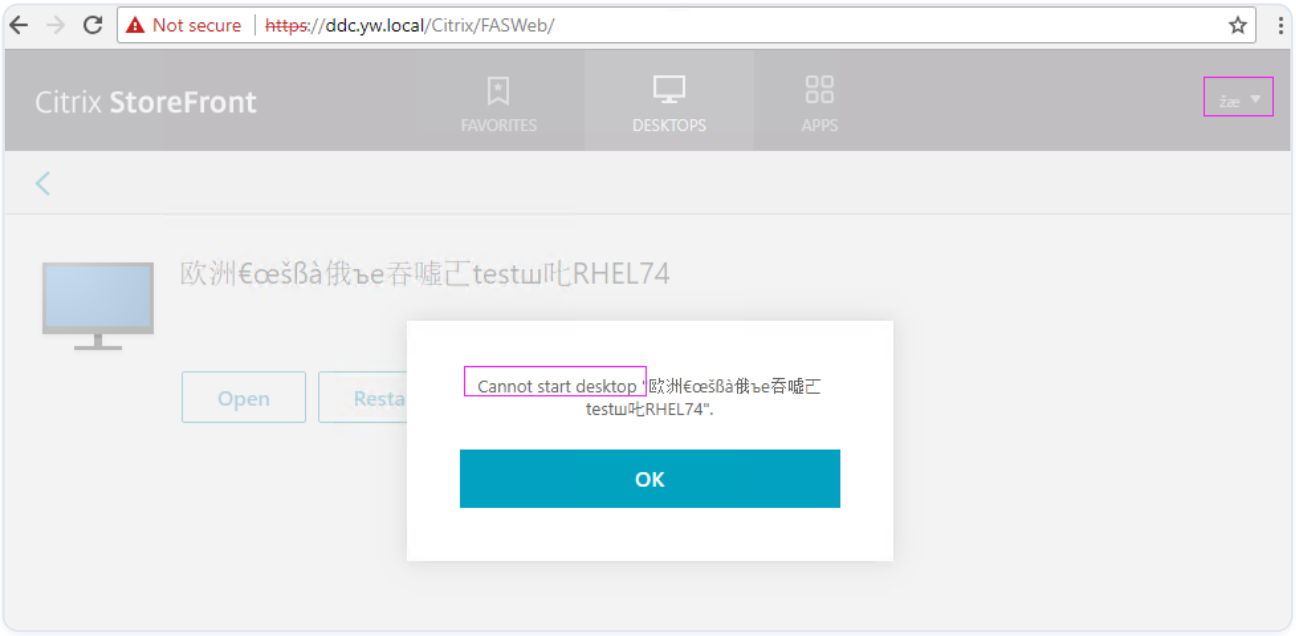
フェデレーション認証サービス用に ADFS IdP を展開します。詳細については、フェデレーション認証サービス ADFS 展開 を参照してください。

関連情報

- 一般的な FAS 展開については、展開アーキテクチャ 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の詳細構成の章で紹介されています。

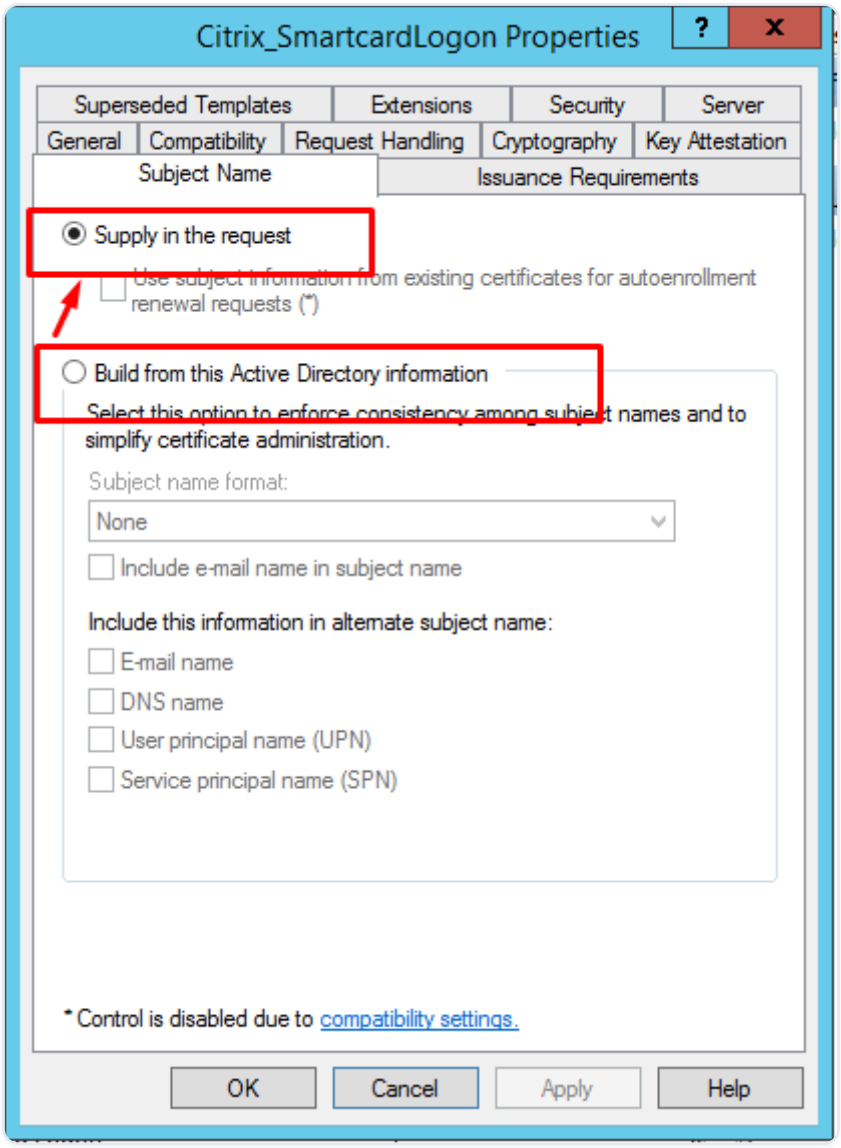
既知の問題

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。



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。



非SSO認証

Citrix Workspace™アプリにログオンした後、異なる資格情報を使用してCitrix Virtual Apps and Desktops™セッションを起動できます。この機能を有効にするには、Linux VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t
"REG_DWORD" -v "fPromptForDifferentUser" -d "0x00000001" --force
```

この機能を有効にすると、セッションの起動時に資格情報の入力を求められます。この機能は、以下のディストリビューションでサポートされています。

- RHEL 8.8
- RHEL 8.6

- RHEL 8.4
- RHEL 7.9 / CentOS 7.9
- ロッキー・リナックス 8.6
- ロッキー・リナックス 8.8
- デビアン 10.9
- SUSE 15.3
- SUSE 15.2

スマートカード

クライアントデバイスに接続されたスマートカードを使用して、Linux仮想デスクトップセッションにログオンする際の認証を行うことができます。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。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、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
  - ロッキー Linux 8
  - RHEL 7/CentOS 7
  - ウブントゥ 20.04
  - ウブントゥ 18.04

- デビアン 10.9

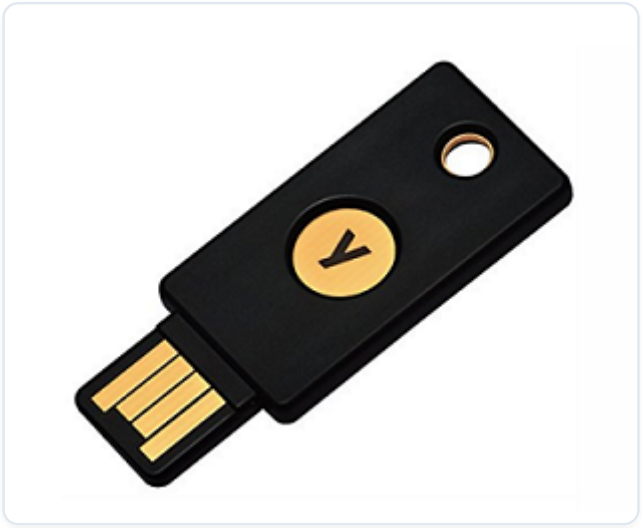
VDAのインストールが完了したら、VDAがDelivery Controller™に登録できること、およびWindows資格情報を使用して公開されたLinuxデスクトップセッションを開けることを確認してください。

- 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と完全に互換性のある代替品として、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、YubiKey 4スマートカードを構成を説明するための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。



組織がより高度なスマートカードを必要とする場合は、サポートされているLinuxディストリビューションと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てください。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を参照してください。スマートカードを挿入し、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ために次のコマンドを実行します。

```
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、次の手順を完了してください。

1. PEM形式のルート証明書を、通常はCAサーバーから取得します。

   DERファイル（*.crt、*.cer、*.der）をPEMに変換するには、次のようなコマンドを実行できます。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。

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、PKINIT情報をデフォルトレルムに追加します。この設定は、**/etc/krb5.conf**構成ファイルを通じて変更できます。

> 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てください。

> **ヒント:**

> SSSDをドメイン参加に使用している場合は、ctxsmartlogon.shを実行した後、SSSDサービスを再起動してください。

```
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、「はい」を選択する(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/smart-card-logon-script1.png)

ctxsmartlogon.shスクリプトを実行して、スマートカードを無効にすることもできます。

```
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、「いいえ」を選択する(/en-us/linux-virtual-delivery-agent/2203-ltsr/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いた構成については、Knowledge Centerの記事CTX206156を参照してください。

次のステップに進む前に、すべてのコンポーネントが正しく構成されていること、秘密鍵とユーザー証明書をスマートカードにダウンロードしていること、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の通信のためのWindowsスマートカードインターフェイスを提供します。Linux VDAにおける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。

RHEL 7/CentOS 7、RHEL 8、およびRocky Linux 8：

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

Ubuntu 20.04、Ubuntu 18.04、Debian 10.9:

```
apt-get install -y libpcsclite1 libccid
```

スマートカードドライバーをインストールする
OpenSCは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。

RHEL 8、Rocky Linux 8、RHEL 7/CentOS 7:

```
yum install opensc
```

ウブントゥ 20.04、ウブントゥ 18.04、デビアン 10.9：

```
apt-get install -y 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
次のコマンドを実行して、pam_krb5およびkrb5-pkinitモジュールをインストールします。

RHEL 7/CentOS 7：

```
yum install pam_krb5 krb5-pkinit
```

RHEL 8およびRocky Linux 8：

```
yum install krb5-pkinit
```

ウブントゥ 20.04、ウブントゥ 18.04：

```
apt-get install libpam-krb5 krb5-pkinit
```

デビアン 10.9：

```
apt-get install -y 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) からチケット保証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 構成ファイルで、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 `&&` のいずれかになります。これは、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、\|\| 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。

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。

```
CTXDEV.LOCAL = {
    kdc = [redacted]
    auth_to_local = RULE:[1:$1@$0]
    pkinit_kdc_hostname = ctx-ad.ctxdev.local
    pkinit_anchors = FILE:/etc/pki/CA/certs/certnew.pem
    pkinit_eku_checking = kpServerAuth
    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
}
```

PAM 認証の構成

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。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_opt=X509_user_identity=PKCS11:/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so
auth        sufficient    pam_permit.so
auth        required      pam_deny.so

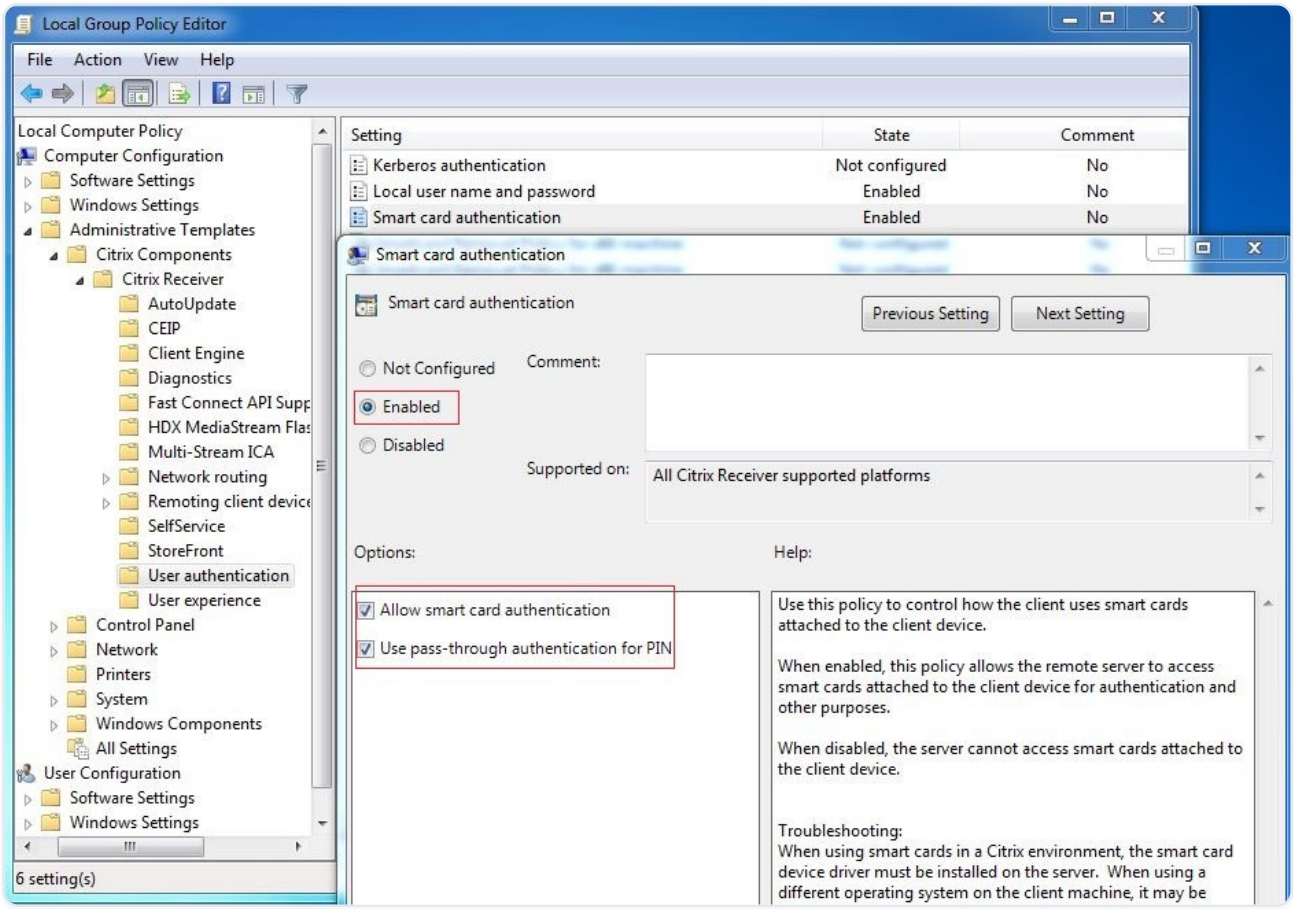
account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore]  pam_krb5.so
account     required      pam_permit.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
#session     optional      pam_oddjob_mkhomedir.so umask=0077
session     optional      pam_mkhomedir.so umask=0077
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_sss.so
session     optional      pam_krb5.so
```

（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。詳細については、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、次のようにスマートカード認証を有効にします。



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、高遅延の WAN 環境でスマートカードを使用する際のパフォーマンスが向上します。詳細については、「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 フォー ウィンドウズ 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、関連付けられた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、関連付けられた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

スマートカードを使用して、SSOおよび非SSOの両方のシナリオで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StoreFront™に自動的にログオンします。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、StoreFrontにログオンするために証明書を選択し、PINを入力するよう求められます。

  StoreFrontにログオンするためのPIN入力(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/pin-new.png)

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、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。

ログオン認証(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/logon-authentication.png)

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と再認証が失敗するため、Linux VDA側で灰色のキャッシュウィンドウが表示され、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促す他のプロンプトは表示されません。

ただし、StoreFront側で、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可能性があります。

スマートカードの挿入(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/please-insert-smart-card.png)

制限事項

限られたLinuxディストリビューションとAD統合方法のサポート

- スマートカードパススルー認証は、限られたLinuxディストリビューションとAD統合方法をサポートしています。以下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	クエスト
RHEL 8	はい	はい	はい	はい
ロッキー Linux 8	はい	はい	いいえ	いいえ
RHEL 7, CentOS 7	はい	はい	はい	はい
ウブントゥ 22.04/18.04	はい	はい	はい	はい
デビアン 10.9	はい	はい	はい	いいえ

スマートカード取り外しポリシー

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ライブラリも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。

認証されていない（匿名）ユーザーによるアクセス

ユーザーがStoreFront™またはCitrix Workspace™アプリに資格情報を提示することなく、アプリケーションやデスクトップにアクセスできるようにすることができます。認証されていないユーザーにアクセスを許可するには、認証されていないStoreFrontストアを持ち、デリバリーグループで認証されていないユーザーのアクセスを有効にする必要があります。

注:

認証されていないユーザーによるアクセスは、ドメイン参加済みVDAでのみサポートされます。

セッションの事前起動は、認証されていないユーザーではサポートされていません。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StoreFrontストアを作成する

1. Citrix StoreFront管理コンソールの左ペインで**［ストア］ノードを選択し、［操作］ペインで［ストアの作成］**をクリックします。
2. **［ストア名］ページで、ストアの名前を指定し、［認証されていないユーザーのみがこのストアにアクセスすることを許可する］を選択して、［次へ］**をクリックします。

詳しくは、「ストアの作成」を参照してください。

デリバリーグループで認証されていないユーザーのアクセスを有効にする

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内のアプリケーションとデスクトップを使用できるユーザーを指定する際に、認証されていないユーザーにアクセスを許可できます。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。

認証されていないユーザーセッションのアイドルタイムアウトを設定する

認証されていないユーザー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、クライアントが切断されると自動的にログオフされます。レジストリ設定**AnonymousUserIdleTime**を使用して、カスタムアイドルタイムアウトを構成できます。たとえば、5分のカスタムアイドルタイムアウトを設定するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\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ユーザー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、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v MaxAnonymousUserNumber
 -d  0x00000020
```

> **重要:**

> 認証されていないユーザー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

認証されていないユーザー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。**

レジストリが次の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てください。

```
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 **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 を使用する認証されていないユーザー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、認証されていないユーザー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 `/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals` のようなグローバルな `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` ファイルを修正して、`[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 ページ を参照してください。

ファイル

このセクションには、次のトピックが含まれています。

- ファイルのコピーと貼り付け(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/file/file-copy-and-paste.html)
- ファイル転送(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/file/file-transfer.html)

ファイルのコピーと貼り付け

ユーザーは、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、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＆ペーストできます。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。

ファイルを正常にコピー＆ペーストするには、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。
- ファイル名にASCII文字のみが含まれ、特殊文字が含まれていないこと。

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は、以下で利用できます。

- RHEL 7.9
- RHEL 7.8
- SLES 12.5
- ウブントゥ 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、この機能の設定に関連しています。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:**
>
> ファイルのコピーアンド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、コピー操作として扱われます。
- 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。
- ファイルのコピーアンドペーストは、順次実行する必要があります。前のファイルが正常にコピーアンドペーストされた後にのみ、次のファイルをコピーできます。

ファイル転送

ファイル転送は、Linux VDAとクライアントデバイスの間で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により、ユーザーはCitrix Workspace™ app for HTML5およびfor Chrome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 app for HTML5およびfor Chrome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、ファイル転送によりLinux VDAとクライアントデバイスの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10個のファイルを転送できます。

ツールバーのアップロードアイコン(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/upload-icon-on-toolbar.png)

> **注:**
>
> Linux VDAとクライアントデバイスの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を有効にします。 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、自動的にクライアントデバイスに転送されます。

> **注:**
>
> 自動ダウンロードには、**File transfer for Citrix Workspace app for ChromeOS/HTML5** および **Download file for Citrix Workspace app for ChromeOS/HTML5** ポリシーが **Allowed** 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ブラウザアプリセッション内で、Webサイトからダウンロードしたファイルは、VDA上の**Save to My Device**ディレクトリに保存され、クライアントデバイスに自動的に転送されます。自動ダウンロードを実現するには、セッション内Webブラウザのデフォルトダウンロードディレクトリを**Save to My Device**に設定し、Citrix Workspace app for HTML5またはfor Chromeを実行するWebブラウザでローカルダウンロードディレクトリを設定します。

- **Save to My Device**にファイルを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、対象ファイルを選択し、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

デフォルトでは、ファイル転送は有効になっています。これらのポリシーは、**ユーザー設定 > ICA\ファイルリダイレクト**の下にあります。Citrix Studioを使用して変更してください。ファイル転送ポリシーを使用する際は、以下を考慮してください。

File transfer for Citrix Workspace app for ChromeOS/HTML5: エンドユーザーがCitrix Virtual Apps and DesktopsセッションまたはCitrix DaaSセッションとエンドユーザーの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。

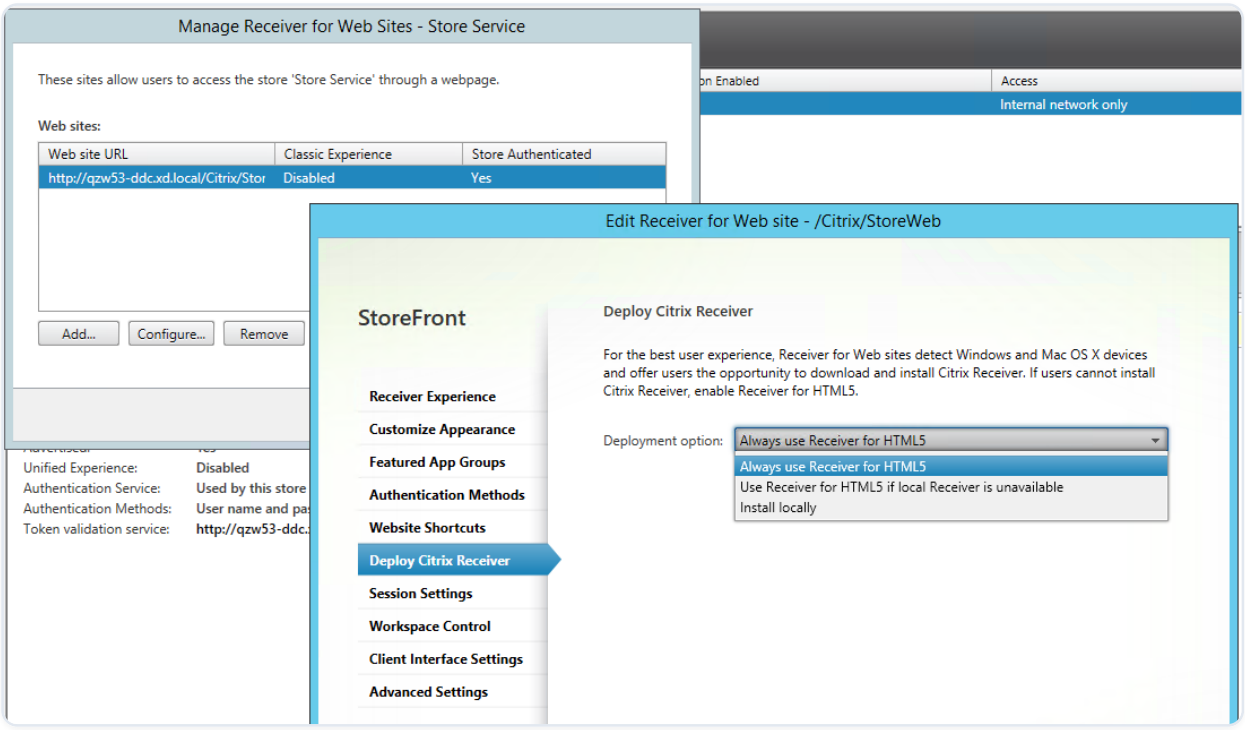
Upload file for Citrix Workspace app for ChromeOS/HTML5: ユーザーがファイルをアップロードすることを許可または禁止します。これは、ユーザーのデバイスからCitrix Virtual Apps and DesktopsセッションまたはCitrix DaaSセッションへのアップロードです。

Download file for Citrix Workspace app for ChromeOS/HTML5: ユーザーが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。これは、Citrix Virtual Apps and DesktopsセッションまたはCitrix DaaSセッションからユーザーのデバイスへのダウンロードです。

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**Allowed**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。

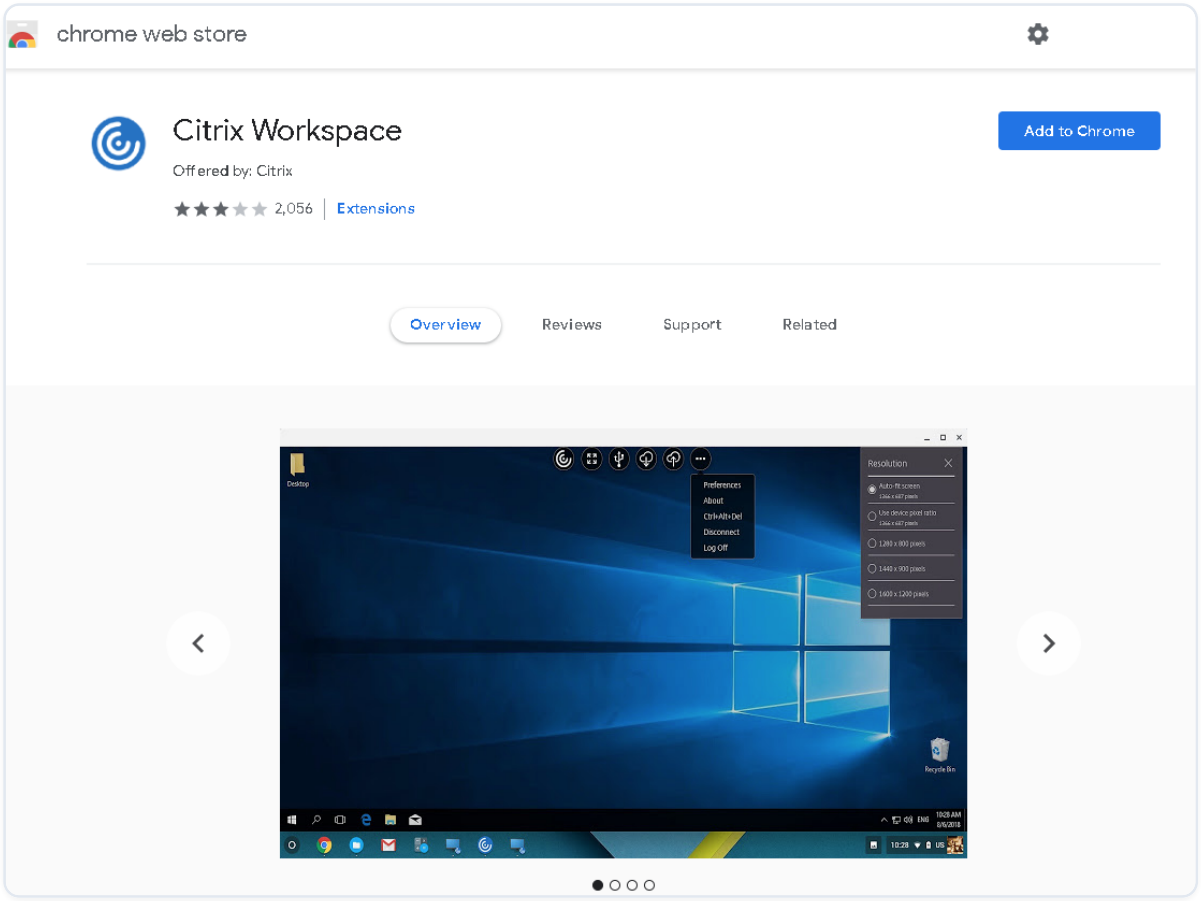
Citrix Workspaceアプリ for Chrome を介してファイル転送機能を使用するには:

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。



c. Citrix Workspaceアプリ をGoogle Chromeに追加するには、**Add to Chrome**をクリックします。

3. Chromeアプリページで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を入力したことがある場合は、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間で1つ以上のファイル転送を実行します。

グラフィックス

このセクションには、次のトピックが含まれています。

- グラフィックの構成と微調整(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/graphics/configuring-graphics.html)
- HDX画面共有(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/graphics/hdx-screen-sharing.html)
- 非GRID 3Dグラフィック(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/graphics/non-vgpu-graphics-cards.html)

- テキストベースのセッションウォーターマーク(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/graphics/text-based-session-watermark.html)
- Thinwireプログレッシブディスプレイ(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/graphics/thinwire-progressive-display.html)

グラフィックの構成と微調整

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。

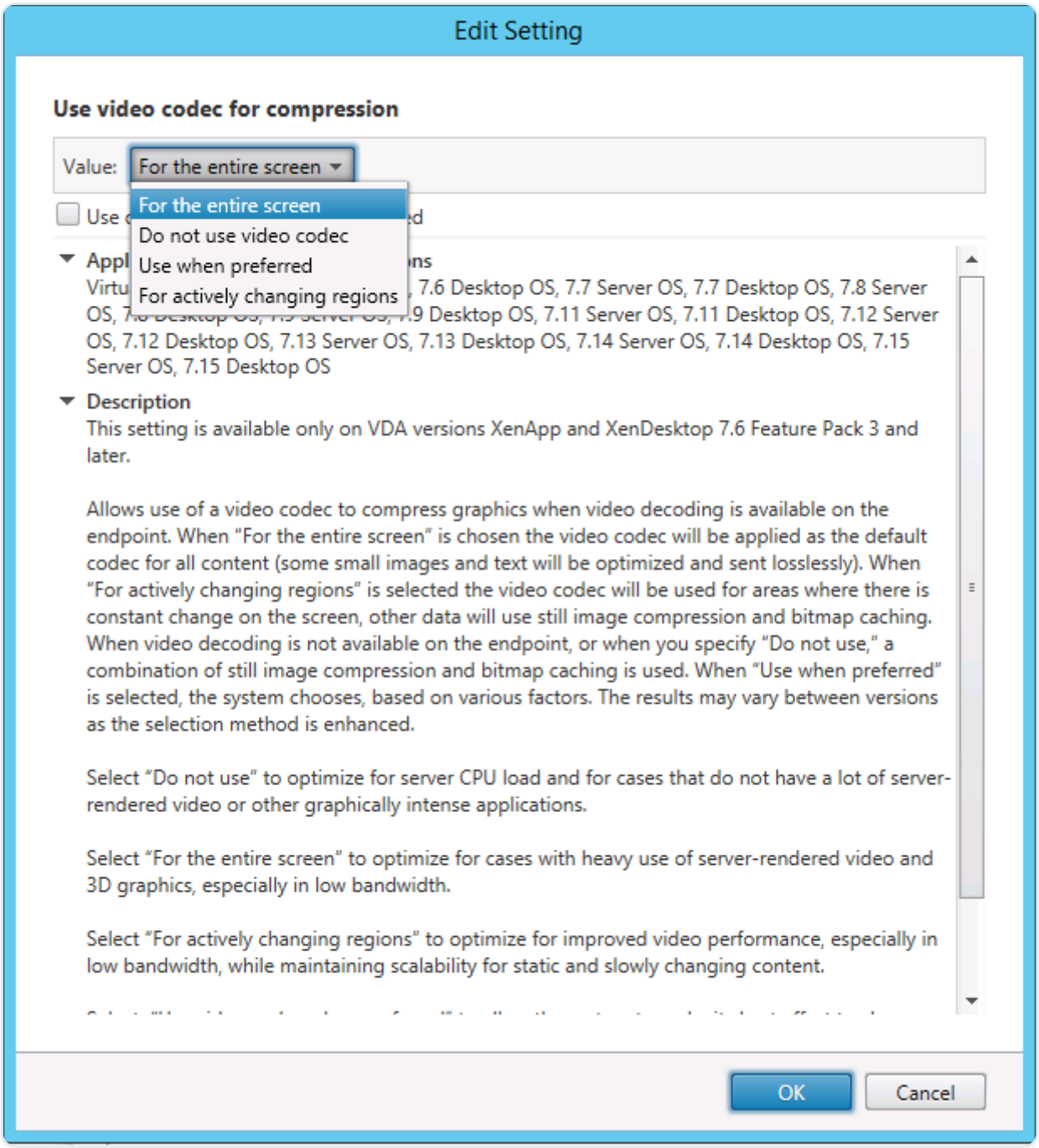
詳しくは、(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/system-requirements.html)および(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/installation-overview.html)セクションを参照してください。

構成

Thinwire は、Linux VDA で使用されるディスプレイリモート処理テクノロジーです。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。

(/ja-jp/citrix-virtual-apps-desktops/policies/reference/ica-policy-settings/graphics-policy-settings.html) 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応する次のオプションを提供します。

- **推奨される場合に使用する**。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。これにより、すべての Citrix® 接続で Thinwire が選択され、スケーラビリティ、帯域幅、および一般的なデスクトップワークロードに対する優れた画質のために最適化されます。
- **画面全体の場合**。Thinwire をフルスクリーン H.264 または H.265 で配信し、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の場合**。Thinwire のアダプティブディスプレイテクノロジーは、動く画像（ビデオ、移動中の 3D）を識別します。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 ビデオコーデックを選択的に使用することで、HDX Thinwire は、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消費量の削減と品質の向上、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メリットを得られます。この機能を有効にするには、ポリシー設定の**ビデオコーデックを圧縮に使用する**を**推奨される場合に使用する**（デフォルト）または**アクティブに変更される領域の場合**に変更します。詳しくは、(/ja-jp/citrix-virtual-apps-desktops/policies/reference/ica-policy-settings/graphics-policy-settings.html)を参照してください。



ディスプレイリモート処理のパフォーマンスを微調整するために、次の視覚表示ポリシー設定を含む、その他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

並列処理

Thinwire は、特定のタスクを並列化することで、1 秒あたりのフレーム数（FPS）を向上させることができますが、全体的な CPU 消費量がわずかに増加するというオーバーヘッドがあります。こ

の機能はデフォルトで無効になっています。この機能を有効にするには、VDA で次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"ParallelProcessing" -d "0x00000001" --force
```

ThinwireでのBuild to LosslessにH.264を使用する

デフォルトでは、**Visual quality**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エンコーディングは、優れた画質を提供します。**Use video codec for compression**ポリシーは、その優先設定を制御し、デフォルトは**Use when preferred**です。**Build to Lossless**でJPEGを強制的に使用するには、**Use video codec for compression**ポリシーを**Do not use video codec**に設定します。クライアントがSelective H.264をサポートしていない場合、ポリシー設定に関係なく**Build to Lossless**はJPEGにフォールバックします。

以下のクライアントはSelective H.264をサポートしています。

- Windows用 シトリックス レシーバー 4.9から4.12まで
- Linux用 シトリックス レシーバー 13.5から13.10まで
- Windows用 Citrix Workspace™ アプリ 1808以降
- Linux用 Citrix Workspace アプリ 1808以降

Visual qualityおよび**Use video codec for compression**ポリシー設定の詳細については、Visual display policy settingsとGraphics policy settings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。

この機能は以下で使用できます。

- Windows用 シトリックス レシーバー 4.10から4.12まで
- Windows向けCitrix Workspaceアプリ1808以降
- 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、H.265 Decoding for graphics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 video encodingを参照してください。

VDAでH.265ハードウェアエンコーディングを有効にするには：

1. 「**ビデオコーデックにハードウェアエンコーディングを使用**」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. 「**圧縮にビデオコーデックを使用**」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「**Build to Lossless**」または「**Always Lossless**」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ディングを有効にするには、「H.265ビデオエンコーディング」を参照してください。

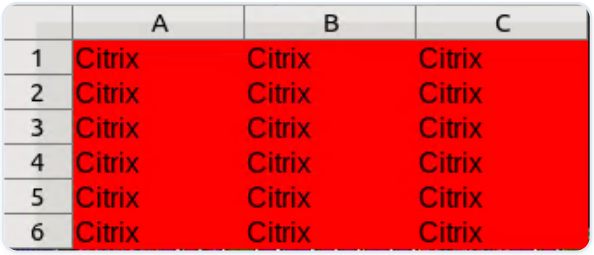
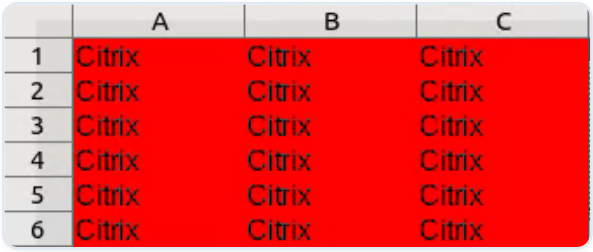
YUV444ソフトウェアエンコーディングのサポート

Linux VDAはYUV444ソフトウェアエンコーディングをサポートしています。YUVエンコーディングスキームは、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`」が輝度または「`luma`」値を表し、「`UV`」が色または「クロマ」値を表します。

この機能は以下で使用できます。

- Windows向けCitrix Receiverのバージョン4.10～4.12
- Windows向けCitrix Workspaceアプリ1808以降

各Y、U、またはVの一意の値は、8ビット（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. 「**圧縮にビデオコーデックを使用**」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. 「**視覚的品質**」ポリシーが「**常にロスレス**」または「**ビルドからロスレス**」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で、HDX™ 3D Pro ハードウェアエンコードを強化します。

HDX 3D Pro ハードウェアエンコード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、次のコマンドを実行してシャープネスと流暢さのバランスを調整することもできます。「**AverageBitRatePercent**」と「**MaxBitRatePercent**」パラメーターは、帯域幅使用率の割合を設定します。値を高く設定するほど、グラフィックはシャープになりますが、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のままになります。シャープ化サポートは、最新のフレームを最高品質で再構成してすぐに送信することで、この問題に対処できます。

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。

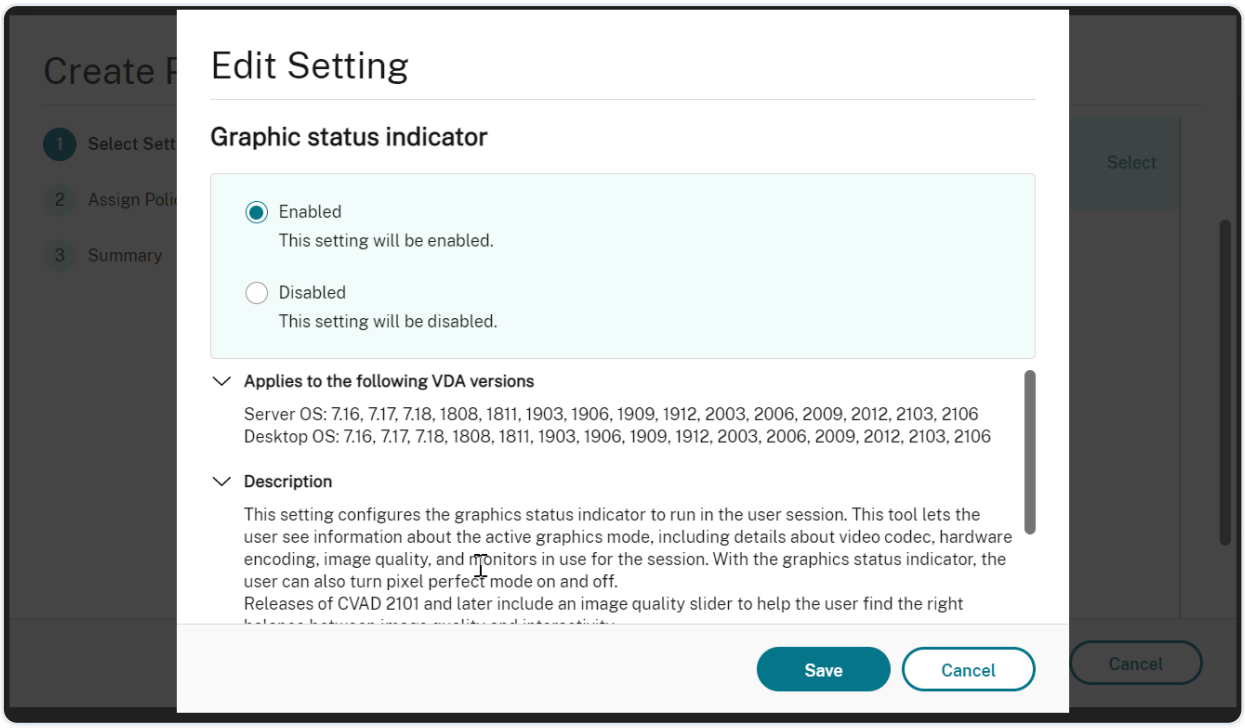
Linux VDA でのマルチモニターサポートの構成については、「CTX220128」を参照してください。

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が含まれています。このスライダーは、画質と対話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:**
>
> **Visual Quality** ポリシーを **Always Lossless** または **Build to Lossless** に設定している場合、スライダーUIは表示されません。

> グラフィック品質スライダー(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/graphic-quality-slider.png)

次の選択肢が利用可能です。

- 画像の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0〜9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、**Let the system decide** を選択します。

- ロスレスモードに切り替えるには、**Pixel perfect** 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

使用中のグラフィックモードを確認するには、次のコマンドを実行します（**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が使用中かどうかを確認する

H.264が使用中かどうかを確認するには、次のコマンドを実行します（**0** は使用中でない、**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が使用中かどうかを確認する

全画面H.265が使用中かどうかを確認するには、次のコマンドを実行します（**0** は使用中でない、**1** は使用中を意味します）。

```
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
```

使用中のYUVエンコードスキームを確認する

使用中のYUVエンコードスキームを確認するには、次のコマンドを実行します（**0**はYUV420、**1**はYUV422、**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**は使用中であ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する

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™の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: 以下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。

Citrix Hypervisor 8.1以降の場合：

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

Citrix Hypervisor 8.1より前のバージョン：

```
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YUV444ハードウェアエンコーディングをサポートしていません

ポリシー設定で **Build to Lossless** を有効にすると、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したときに、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YUV444ハードウェアエンコーディング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起動の制限です。

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うまく描画されないことがあります

Citrix Workspaceアプリのウィンドウサイズを変更すると、画面解像度が変更されます。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。

HDX™ 画面共有

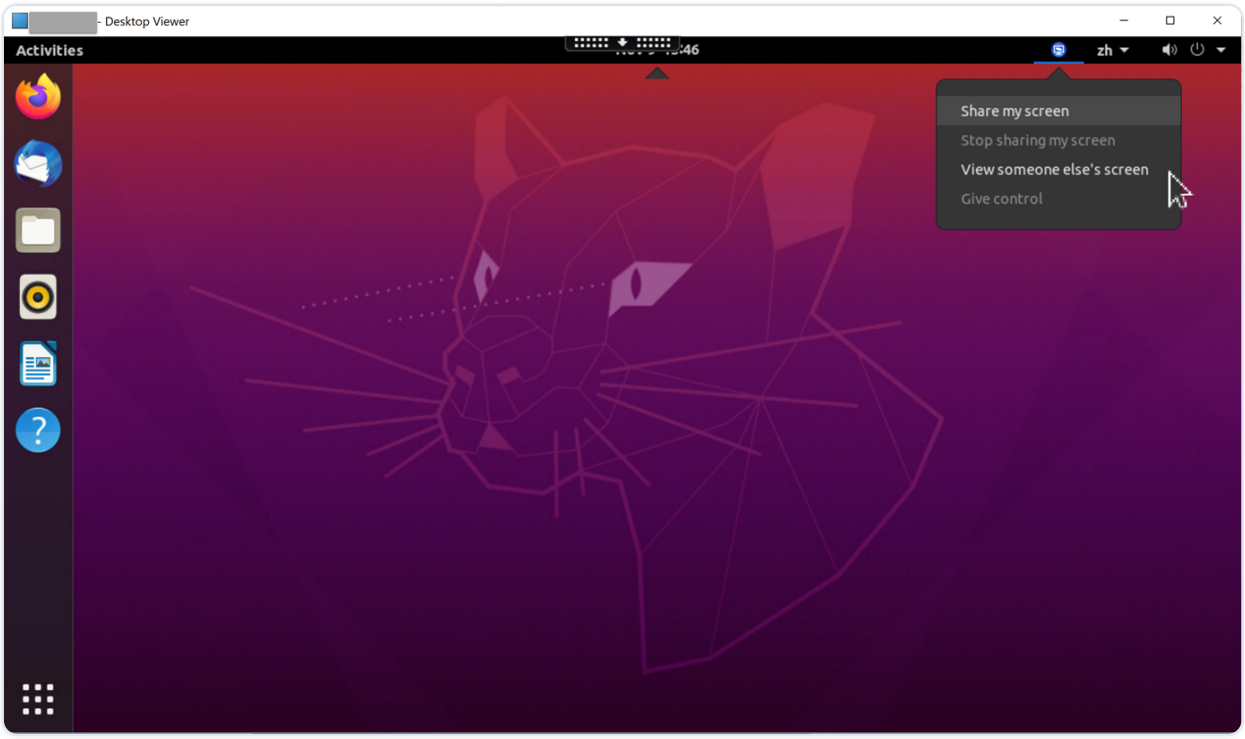
概要

Linux VDAを使用すると、仮想デスクトップの画面を他の仮想デスクトップのセッションユーザーと共有できます。

以下の例では、画面を共有する手順と、他のユーザーの画面を表示する手順を説明します。

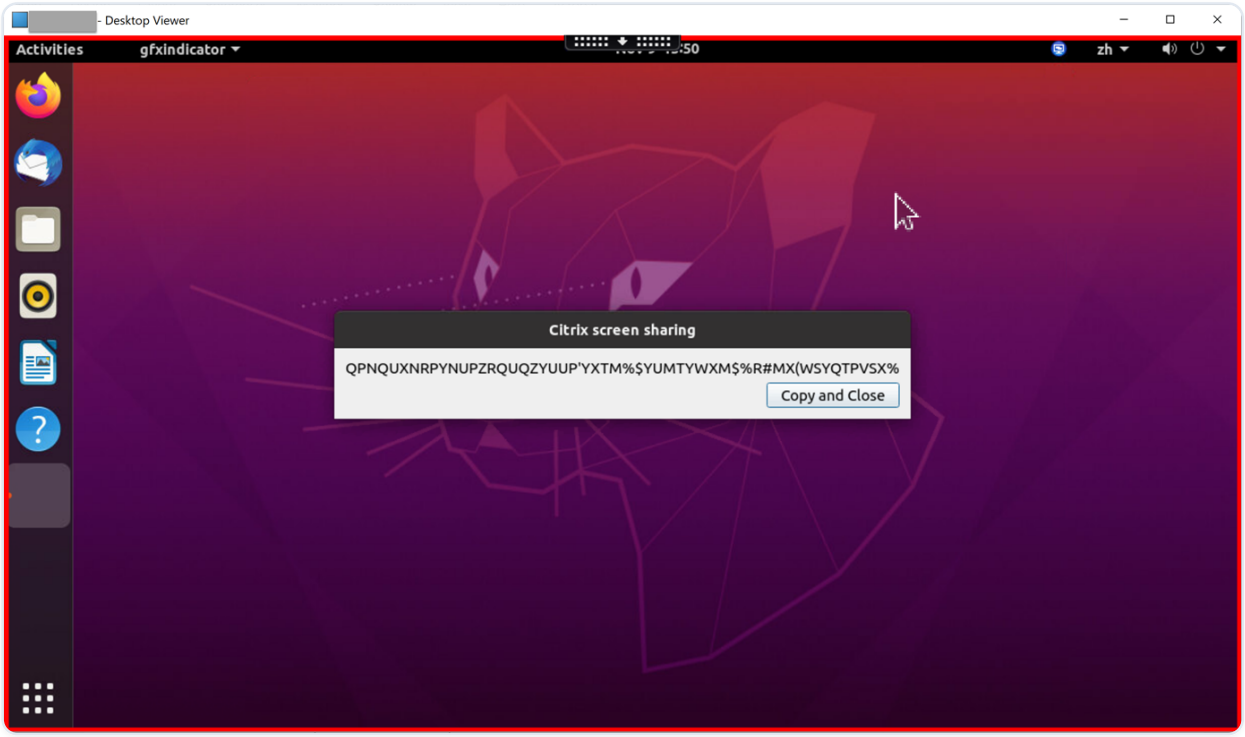
画面を共有するには:

1. 仮想デスクトップの通知領域で、**画面共有**アイコンをクリックし、**画面を共有**を選択します。



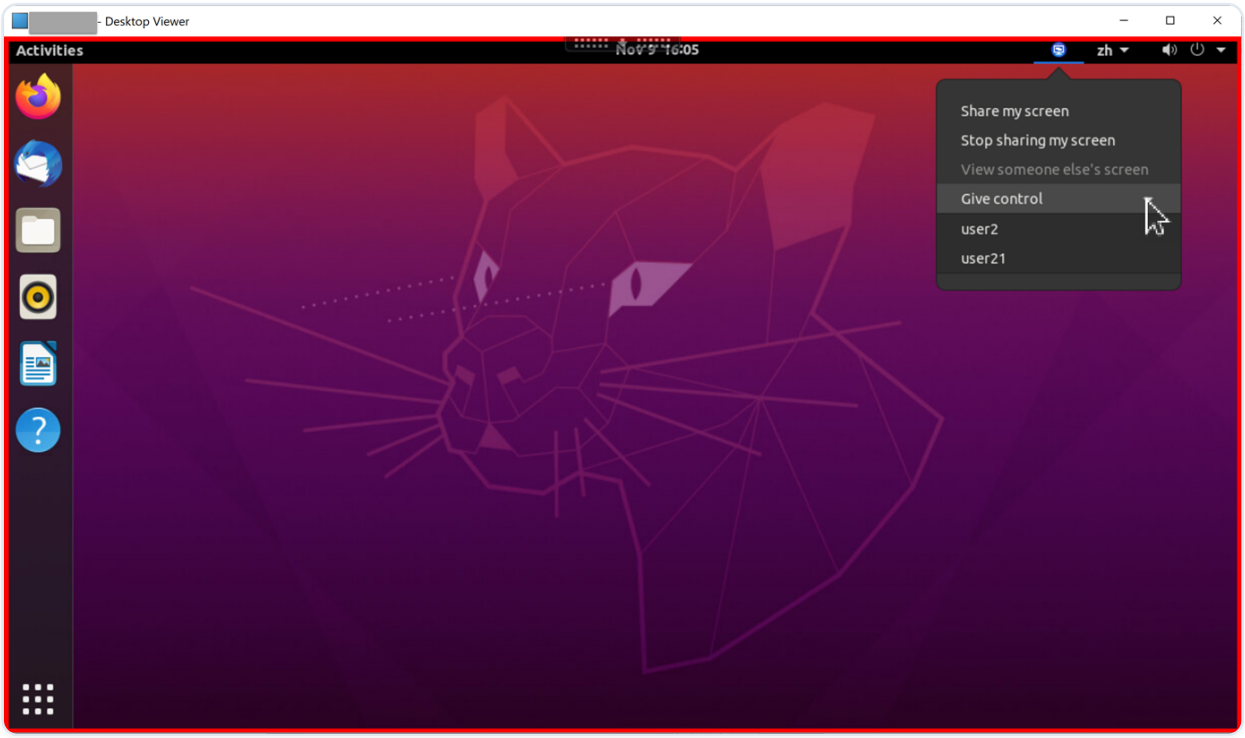
2. **コピーして閉じる**をクリックします。

現在の画面共有コードは、画面共有を停止して再開するまで保持されます。

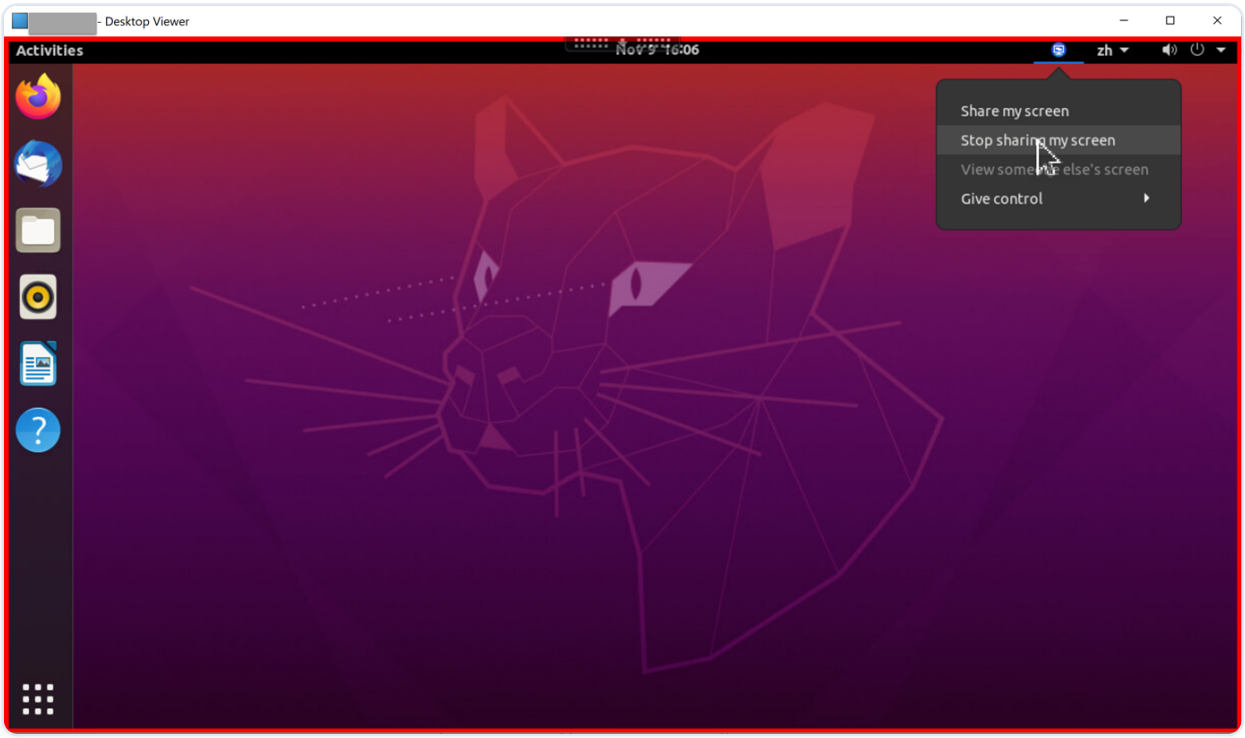


> **ヒント:**
>
> 画面を共有している間、画面の周囲に赤い枠が表示され、共有が進行中であることを示します。

3. 画面を共有したい他の仮想デスクトップのセッションユーザーと、コピーしたコードを共有します。
4. ビューアーに画面を制御させるには、**制御を許可**を選択し、次にビューアーの名前を選択します。制御の許可を停止するには、ビューアーの名前の選択を解除します。

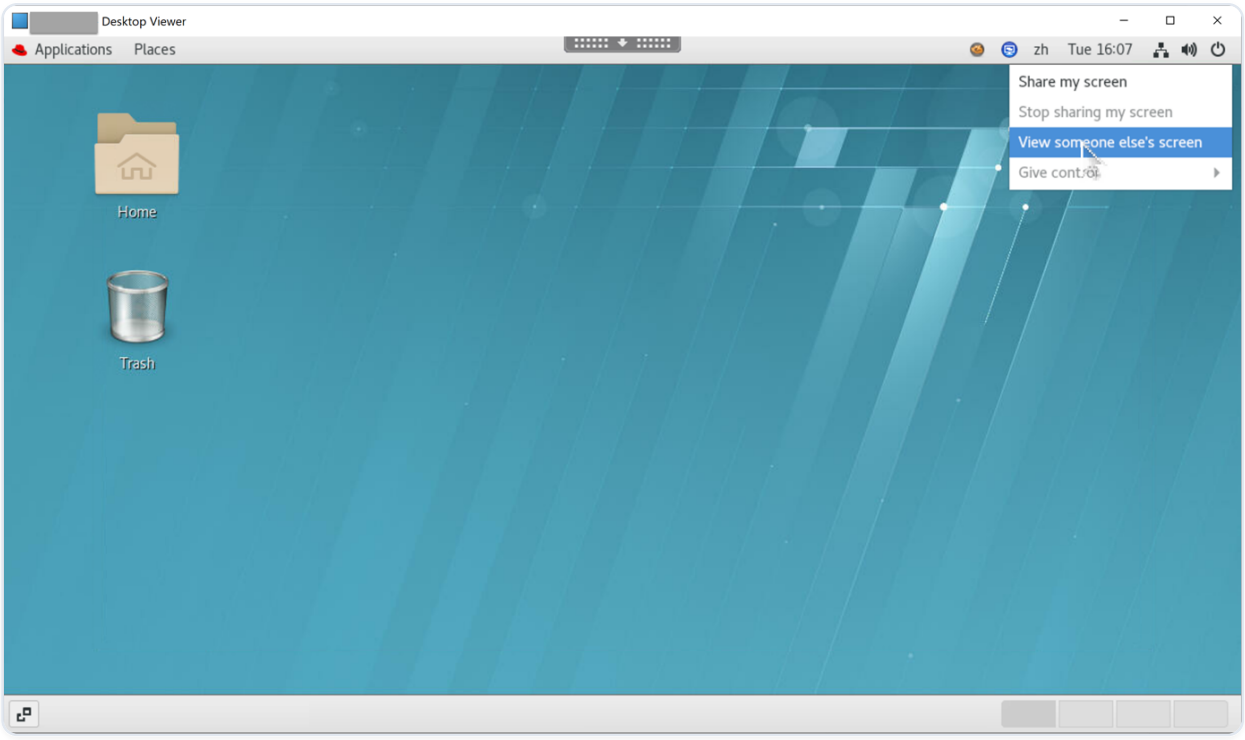


5. 画面共有を停止するには、**画面共有を停止**を選択します。

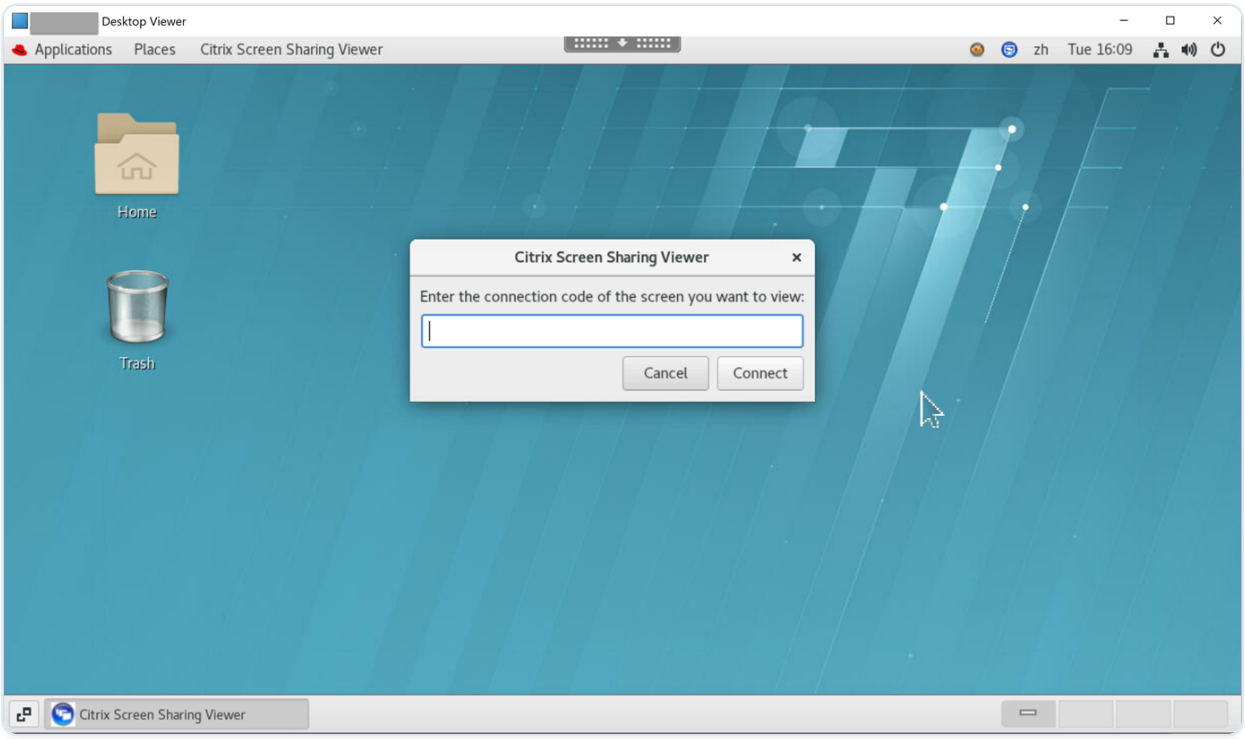


他のユーザーの画面を表示するには:

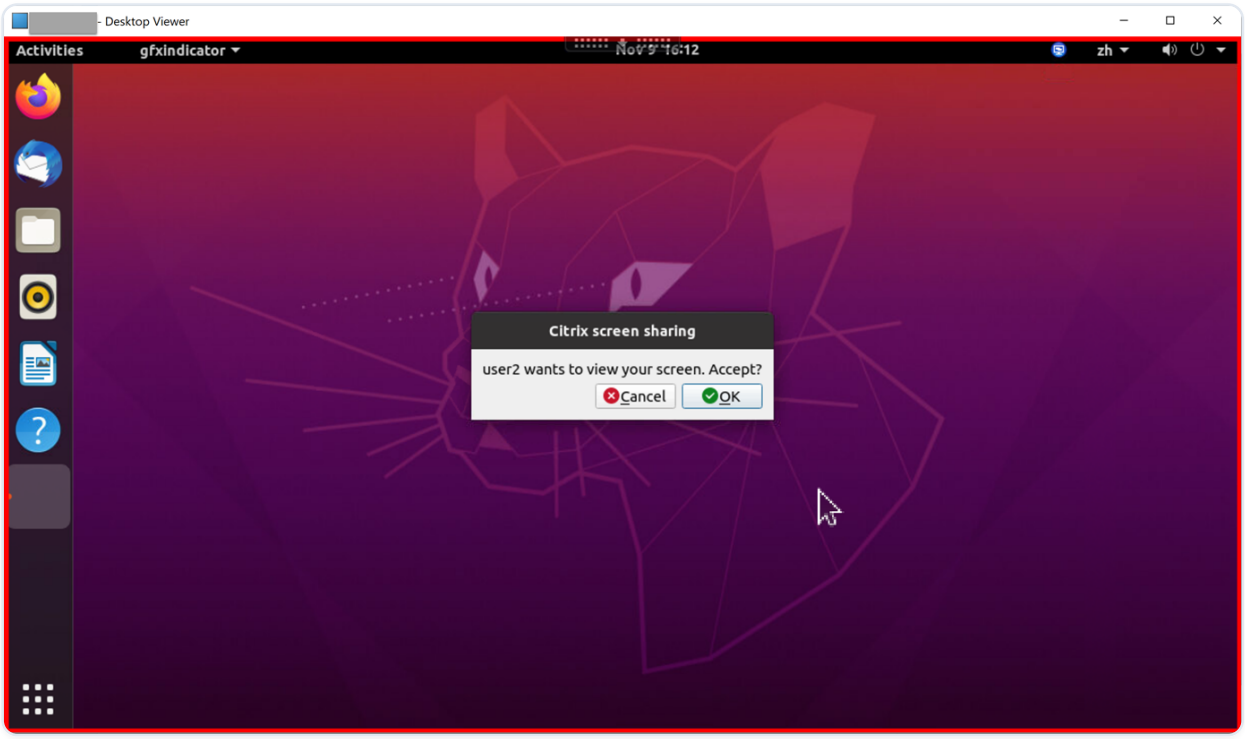
1. 仮想デスクトップの通知領域で、**画面共有**アイコンをクリックし、**他のユーザーの画面を表示**を選択します。



2. 表示したい画面の接続コードを入力し、**接続**をクリックします。



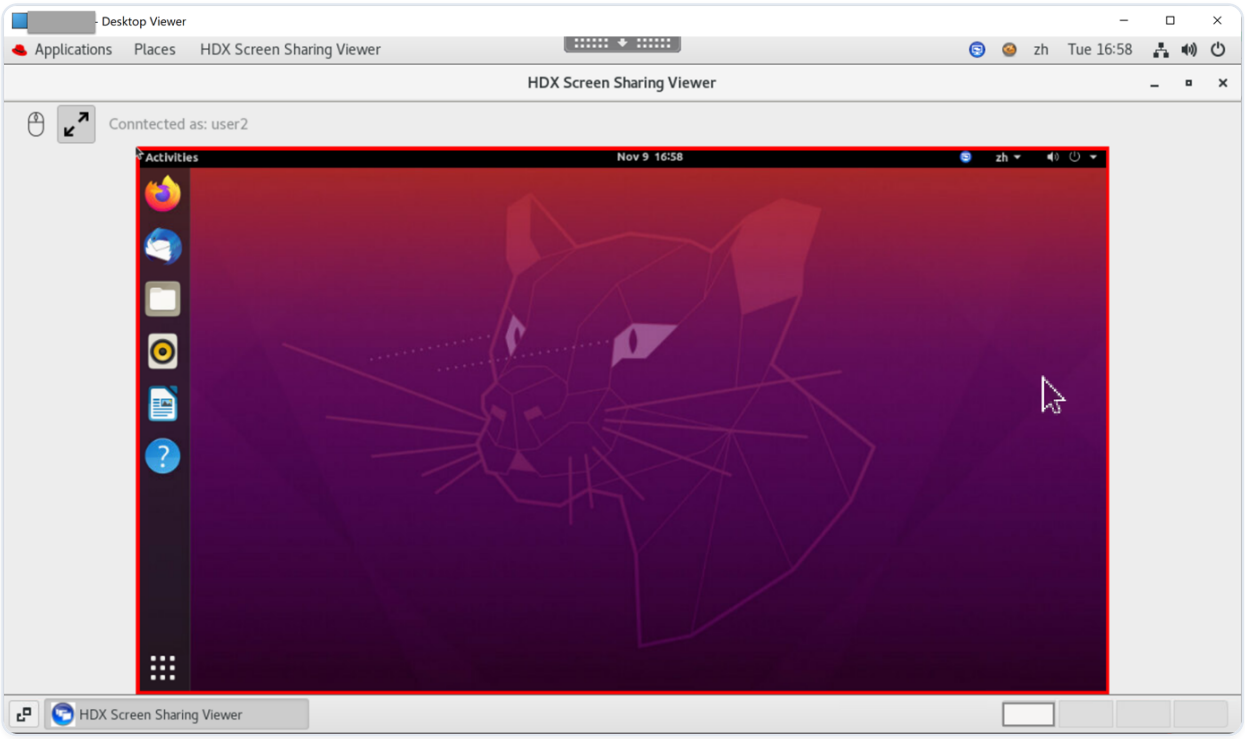
3. 画面共有者がリクエストを承認するまで待ちます。例:



ヒント:

- 共有者側では、Linuxシステムがあなたのリクエストの通知を発行します。
- 共有者が30秒以内にリクエストを承認しない場合、リクエストは期限切れとなり、プロンプトが表示されます。

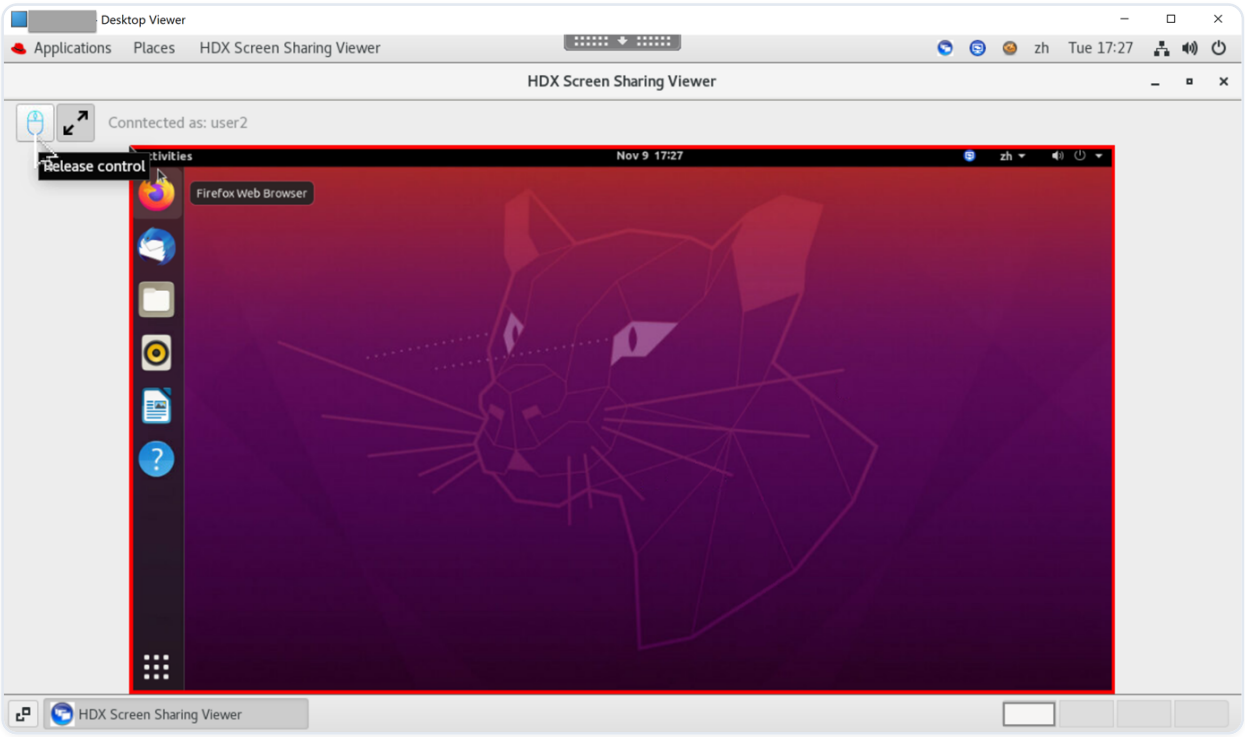
4. 画面共有者が**OK**をクリックしてリクエストを承認すると、共有画面がDesktop Viewerに表示されます。自動的に割り当てられたユーザー名でビューアーとして接続されます。



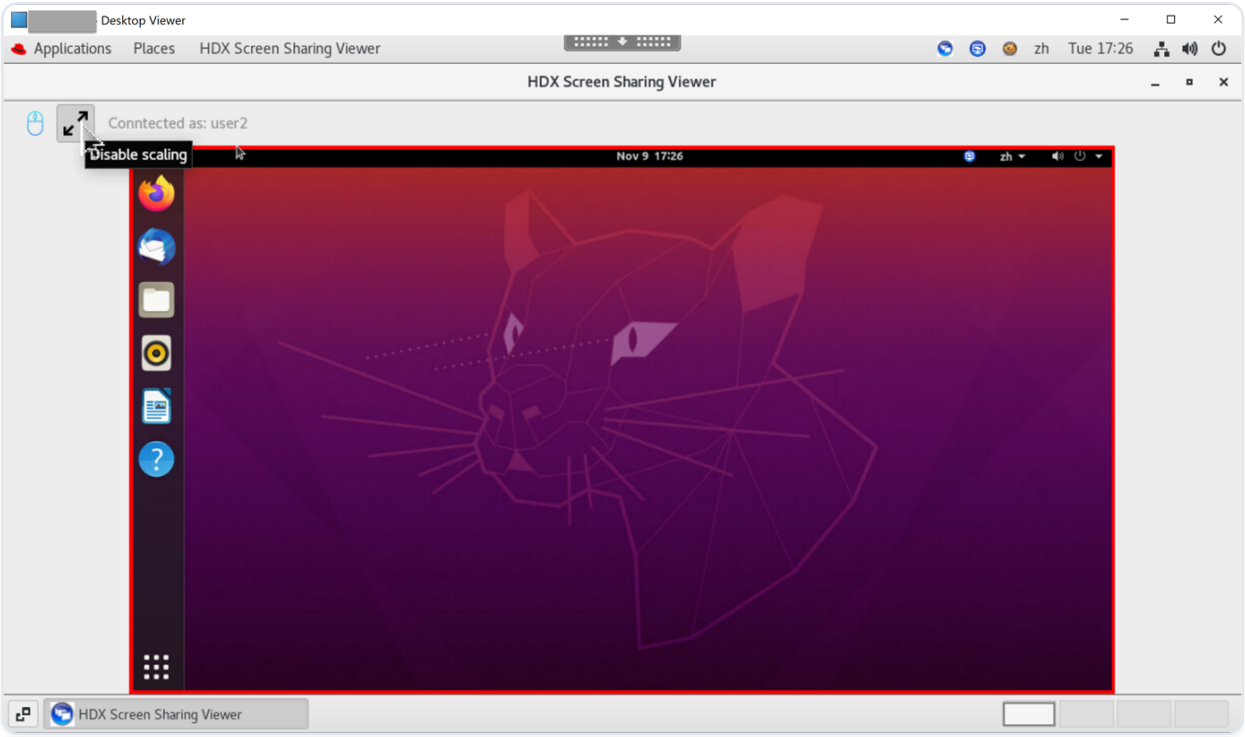
5. 共有画面の制御を要求するには、左上隅にあるマウスアイコンをクリックします。

> **ヒント:**
>
> - 共有者が30秒以内にリクエストを承認しない場合、リクエストは期限切れになります。
> - 一度に1人のビューアーのみが共有画面を制御できます。

> 共有画面の制御を解除するには、マウスアイコンをもう一度クリックします。



6. ディスプレイのスケーリングを無効にするか、ウィンドウサイズに合わせてスケーリングするには、マウスアイコンの隣にあるアイコンをクリックします。



構成

画面共有機能はデフォルトで無効になっています。これを有効にするには、以下の設定を完了してください。

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。

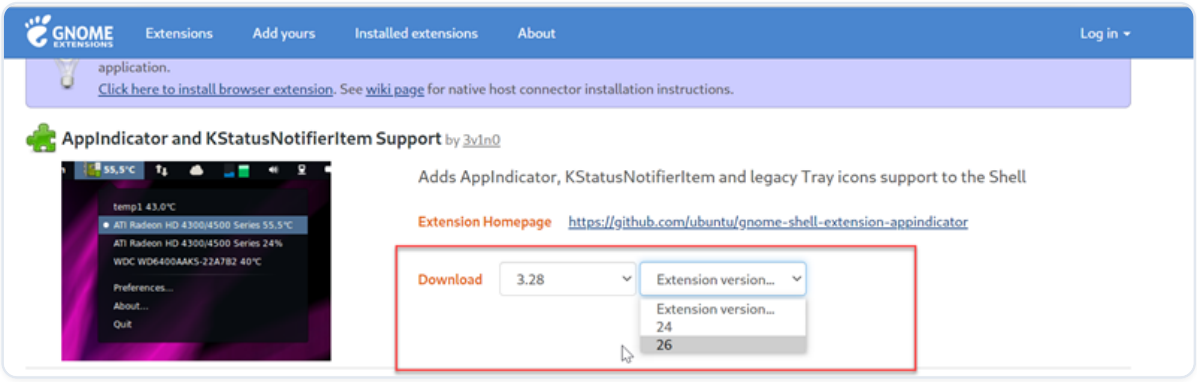
2. Citrix Virtual Apps and Desktops™ 2112以降の場合、Citrix Studioで**ScreenSharing**ポリシーを有効にします。
3. (オプション) Citrix Virtual Apps™ and Desktops 2109以前の場合、以下のコマンドを実行してLinux VDAで画面共有を有効にします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v
"EnableScreenSharing" -d "0x00000001"
```

4. ファイアウォールでポート52525～52625を許可します。
5. (オプション) GNOMEがインストールされたRHEL 8.xまたはSUSE 15.xを使用している場合は、AppIndicatorサポートを有効にするために、GNOMEシェル用の互換性のある拡張機能をインストールします。

   a. `gnome-shell --version`コマンドを実行して、GNOMEシェルバージョンを確認します。

   b. `<https://extensions.gnome.org/extension/615/appindicator-support>`からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能バージョンとして24または26を選択できます。



   c. ダウンロードしたパッケージを解凍します。パッケージ内の**metadata.json**ファイルにある**「uuid」の値が**appindicatorsupport@rgcjonas.gmail.com**に設定されていることを確認します。

   d. `mv`コマンドを実行して、appindicatorsupport@rgcjonas.gmail.comディレクトリを`/usr/share/gnome-shell/extensions/`の下の場所に移動します。

   e. `chmod a+r metadata.json`コマンドを実行して、**metadata.json**ファイルを他のユーザーが読み取れるようにします。

> **ヒント:**

> デフォルトでは、appindicatorsupport@rgcjonas.gmail.comディレクトリ内の**metadata.json**ファイルはrootユーザーのみが読み取り可能です。画面共有をサポートするには、**metadata.json**ファイルを他のユーザーも読み取れるようにします。

f. GNOME Tweaksをインストールします。

g. デスクトップ環境で、`Alt+F2`、`r`、および`Enter`キーを順番に押すか、`killall -SIGQUIT gnome-shell`コマンドを実行してGNOMEシェルをリロードします。

h. デスクトップ環境でGNOME Tweaksを実行し、Tweaksツールで**KStatusNotifierItem/AppIndicator Support**を有効にします。

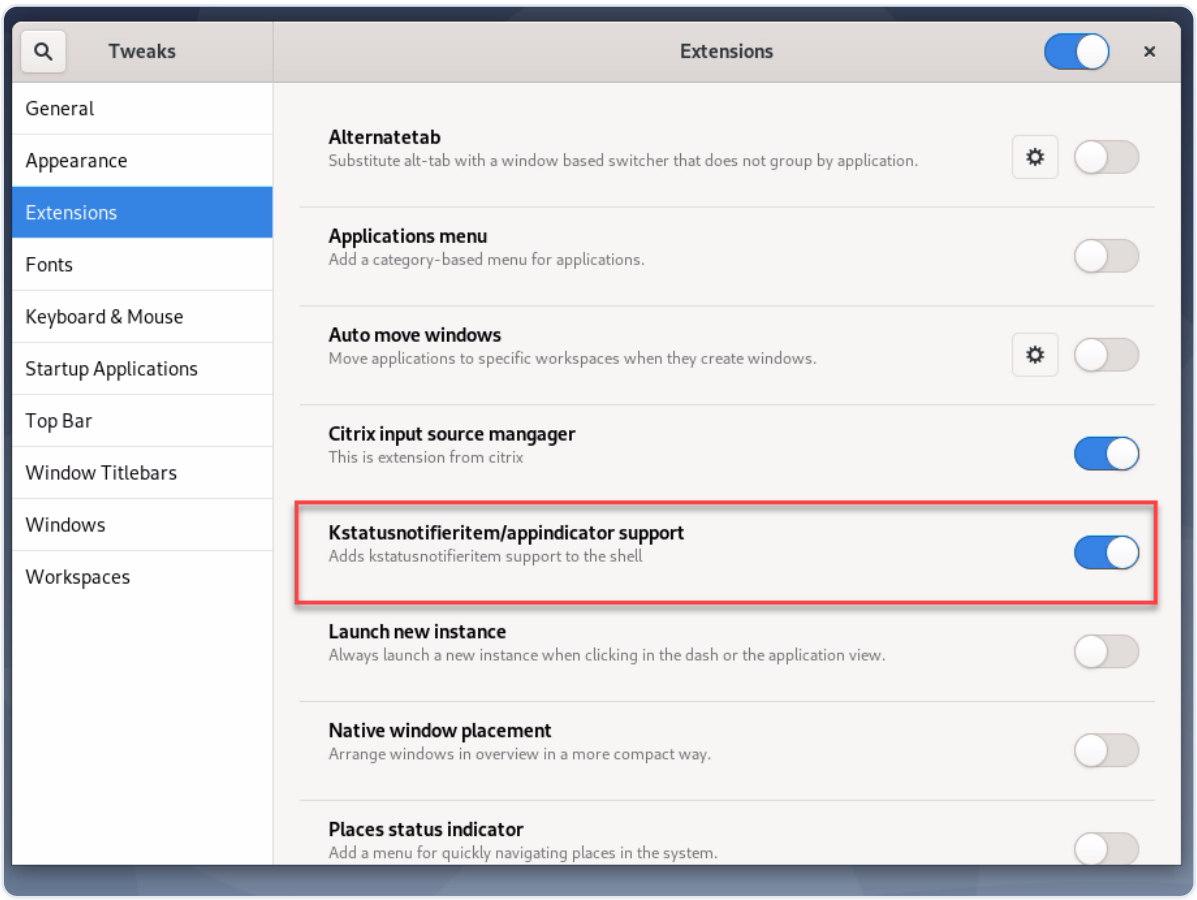
6. （オプション）GNOMEがインストールされたDebian 10を使用している場合は、次の手順を実行してGNOMEシステムトレイアイコンをインストールして有効にします。

a. `sudo apt install gnome-shell-extension-appindicator`コマンドを実行します。GNOMEが拡張機能を表示できるように、一度ログアウトしてから再度ログインする必要がある場合があります。

b. **アクティビティ**画面でTweaksを検索します。

c. Tweaksツールで**拡張機能**を選択します。

d. **Kstatusnotifieritem/appindicator support**を有効にします。



考慮事項

- 画面共有機能はH.265ビデオコーデックをサポートしていません。
- 画面共有機能はアプリセッションでは利用できません。
- デスクトップセッションのユーザーは、デフォルトで最大10人の視聴者とセッション画面を共有できます。最大視聴者数は`ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v "ScreenSharingViewerMaxNum" -d <hex_value>`を介して構成可能です。最大数に達すると、ユーザーが追加の接続要求を受け入れようとしたときにプロンプトが表示されます。

vGPU非対応グラフィックカード

vGPU非対応グラフィックカードとは、NVIDIA仮想GPU (vGPU) ソリューションをサポートしないグラフィックカードを指します。この記事では、vGPU非対応グラフィックカードの使用に関する情報を提供します。

前提条件

vGPU非対応グラフィックカードを使用するには、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。環境変数について詳しくは、「手順7：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構成ファイルを変更する

NVIDIAグラフィックカードの場合

NVIDIAドライバーを使用している場合、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。

その他のグラフィックカードの場合

`/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。このセクションにはビデオドライバー情報が記述されています。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、カードメーカーが提供する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セッションを実行していないときに、ネイティブ構成ファイルを使用してローカル環境でカードが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. 以下のコマンドを実行してレジストリキーを設定し、Linux VDAがステップ1で設定され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vGPUグラフィックを有効にする

非vGPU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の値を1に設定することで、以下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。デバッグ用の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。

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。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者への抑止力として、セッションデスクトップに表示されます。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。このテキストレイヤーは、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には、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

- ブラウザー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、ブラウザー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。
- レガシーNVIDIAドライバーを使用して、セッションが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、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。（この場合、レジストリでNvCaptureTypeが2に設定されています。）
- 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。スクリーン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。

ルートレスXorg

Linux VDAは、非rootユーザー権限でXorgを実行することをサポートしています。これは「ルートレス」Xorgとも呼ばれます。ルートレスXorgは、rootとして実行する場合と比較して、セキュリティが大幅に向上します。

> **注:**
>
> - Xorgは、Amazon Linux 2、RHEL 7、またはRocky Linux 7ではルートレス操作をサポートしていません。
> - ルートレスXorgは、GPUドライバーからのサポートも必要とします。リモートPCアクセスVDAを使用している場合は、GPUドライバーがルートレスXorgをサポートしていることをテストして確認してください。

ルートレスXorgは、GPUなしのVDAおよびNVIDIA GPUを使用するVDAの場合、デフォルトで有効になっています。この機能は、以下のコマンドを使用して手動で有効または無効にできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "1" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "0" --force
```

Thinwire プログレッシブディスプレイ

帯域幅が低い接続や遅延が大きい接続では、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、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**視覚品質に構成したり、低い色深度（16ビットまたは8ビットグラフィック）を設定したりすることで、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。

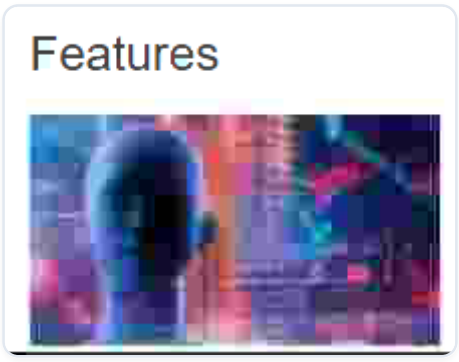
バージョン 7.18 以降、HDX Thinwire は、以下のいずれかのケースでデフォルトでプログレッシブ更新モードに切り替わります。

- 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。

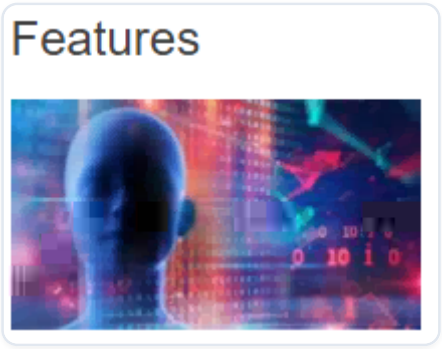
このモードでは：

- 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、文字の**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。



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、プログレッシブにロスレスにシャープ化されます。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を含まなくなり、画像は元の品質で表示されます。



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を回避します。

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Selective H.264で管理されます。

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**視覚的品質**ポリシー設定の**高**、**中**（デフォルト）、**低**でスタンバイ状態になります。

次の場合、プログレッシブモードは強制的にオフ（使用しない）になります。

- **視覚的品質** = **常にロスレス**または**ロスレスにビルド**
- **単純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**（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、デフォルトでは次のいずれかの条件が発生すると有効になります。

- 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

モードが切り替わると、ネットワークの状態が一時的なものであっても、そのモードで最低10秒間は維持されます。

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（ネットワークの状態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、以下のいずれかのコマンドを実行して、プログレッシブモードが切り替わるしきい値を変更できます。

```
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 \<ms単位のしきい値\> です (デフォルト = 200)

例: 100 = ネットワーク遅延が100 msを下回った場合にプログレッシブモードをオンにする.

キーボード

このセクションには、次のトピックが含まれています。

- クライアントIME
- クライアントIMEユーザーインターフェイスの同期
- 動的キーボードレイアウトの同期
- ソフトキーボード
- 複数言語入力のサポート

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。Windowsネイティブ のCJK IMEなど、クライアント側のCitrix Workspace™アプリと互換性のあるIME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、必要に応じてクライアント側で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。

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、テキスト入力領域の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/2203-ltsr/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/2203-ltsr/media/synced-composition-and-candidate-windows.png)

この機能を使用するための前提条件

1. Linux VDAでIntelligent Input Bus (`IBus`) を有効にします。Linux OSで `IBus` を有効にする方法については、OSベンダーのドキュメントを参照してください。例：
   - ウブントゥ: `<https://help.ubuntu.com/community/ibus>`
   - CentOS、RHEL: `<https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods>`
   - デビアン: `<https://wiki.debian.org/I18n/ibus>`
   - SUSE: `<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-gnome-settings.html#sec-gnome-settings-lang>`
2. 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:**
>
> The **Client Keyboard Layout Sync and IME Improvement** policy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。

- 「**クライアントキーボードレイアウトの同期とIMEの改善**」ポリシーを設定して、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の同期とIMEの改善**」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の同期とIMEの改善ポリシーの画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/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の同期とIMEの改善ポリシーの設定画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/setting-client-keyboard-and-ime-policy.png)

次の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。

- `ctxreg`ユーティリティを使用してレジストリを編集し、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"
  ```

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。たとえば、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たが、VDAでは変更されなかった場合、キーマッピングの問題が発生する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™ app 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、**Client Keyboard Layout Sync and IME Improvement** ポリシーを設定するか、`ctxreg` ユーティリティを使用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。

- 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、**Client Keyboard Layout Sync and IME Improvement** 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. **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およびIME改善ポリシーの画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/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。

     Client Keyboard Layout Sync and IME Improvementポリシーの設定画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/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` の **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` ユーティリティを使用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

この機能が有効になっている場合、セッション中にクライアントデバイスでキーボードレイアウトが変更されると、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) に変更すると:

クライアントデバイスでキーボードをフランス語に変更している画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/keyboard-changed-to-french.png)

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:

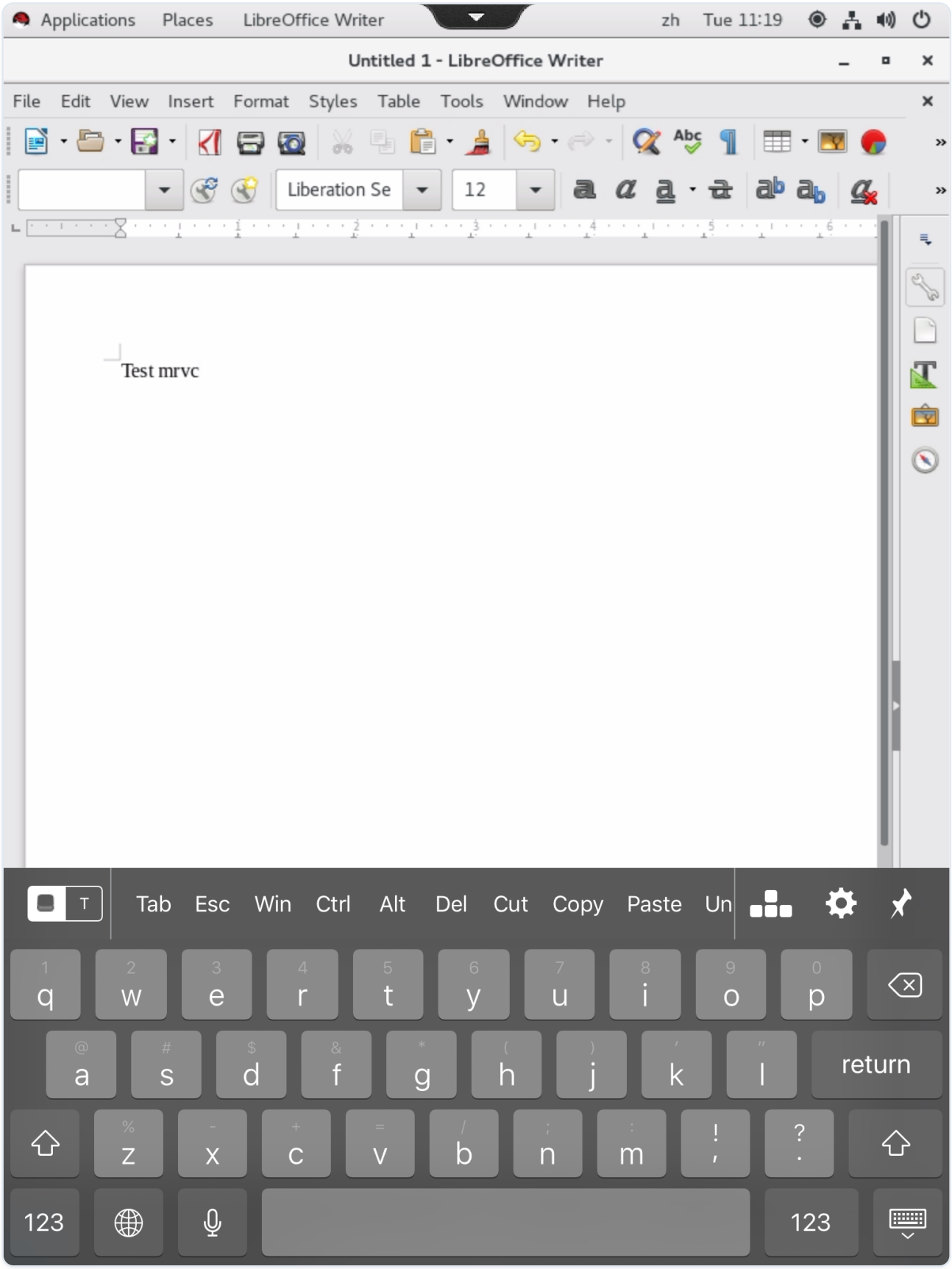
セッションでキーボードがフランス語に変更されている画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/changed-to-fr.png)

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:

タスクバーのフランス語の画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/fr-in-the-task-bar.png)

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自動的に非表示になります。


Applications
Places
LibreOffice Writer
zh
Tue 11:19
Untitled 1 - LibreOffice Writer
File
Edit
View
Insert
Format
Styles
Table
Tools
Window
Help
Liberation Se
12
Test mrvc
Tab
Esc
Win
Ctrl
Alt
Del
Cut
Copy
Paste
Un
return
123
123

注:

この機能は、RHEL 7.8、RHEL 7.9、RHEL 8.1、RHEL 8.2、RHEL 8.3、RHEL 8.6、RHEL 8.8、Rocky Linux 8.6、Rocky Linux 8.8、SUSE 15.3、SUSE 15.2、SUSE 12.5、Ubuntu 18.04、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。**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. 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 "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、次のコマンドを実行して、Intelligent Input Bus (`IBus`) をデフォルトのIMサービスとして構成します。

   ```
   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:

上記の設定は、新しいセッションにログオンしたとき、または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ブラウザーで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。
- ソフトウェアキーボードはパスワード入力フィールドでは使用できません。
- ソフトウェア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。その場合は、アプリ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タブレットにソフトウェアキーボードが表示されます。

複数の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上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用の言語バーを提供します。この言語バーにより、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"
```

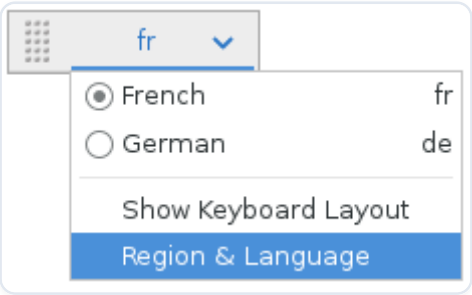
この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000000"
```

使用方法

使い方は簡単です。

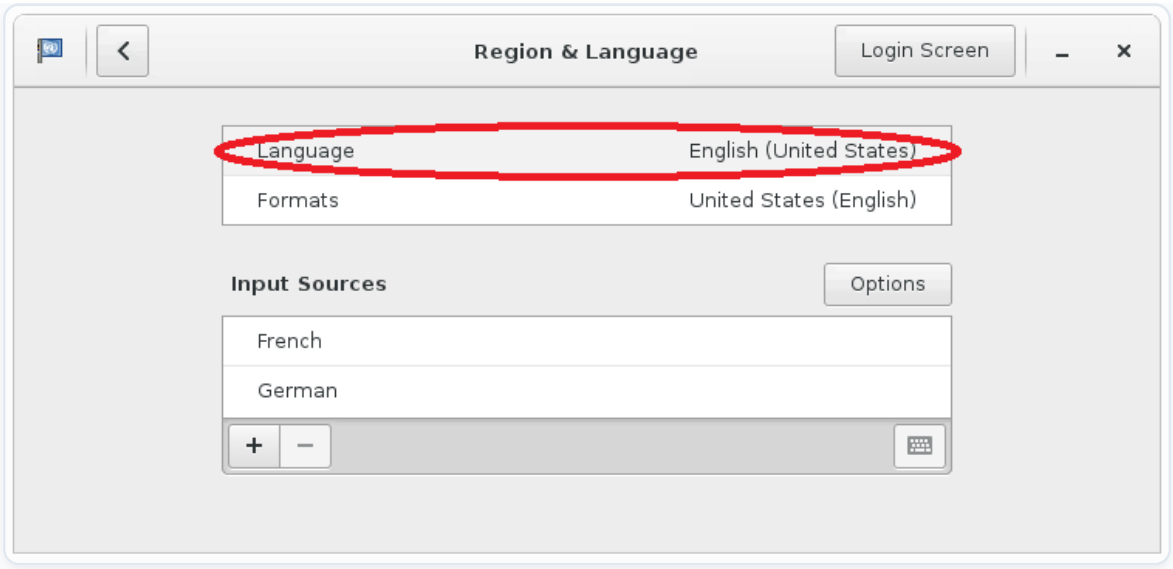
1. 機能を有効にします。
2. テキスト入力を受け付ける公開アプリケーションにアクセスします。アプリケーションの横に言語バーがセッション内に表示されます。
3. ドロップダウンメニューから**Region & Language**を選択し、目的の言語（入力ソース）を追加します。



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場合は、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。
- **地域と言語**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。



マルチメディア

このセクションには、次のトピックが含まれています。

- オーディオ機能
- ブラウザコンテンツリダイレクト
- HDX Webカメラビデオ圧縮

オーディオ機能

アダプティブオーディオ

アダプティブオーディオはデフォルトで有効になっています。次のCitrix Workspace™アプリクライアントをサポートしています。

- Windows向けCitrix Workspaceアプリ – 2109以降のバージョン
- Linux向けCitrix Workspaceアプリ – 2109以降のバージョン
- Mac向けCitrix Workspaceアプリ – 2109以降のバージョン

リストにないクライアントを使用すると、アダプティブオーディオはレガシーオーディオにフォールバックします。

アダプティブオーディオを使用すると、VDAでオーディオ品質ポリシーを手動で構成する必要はありません。アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、プレミアムなオーディオエクスペリエンスを提供します。

次の表は、アダプティブオーディオとレガシーオーディオの比較を示しています。

アダプティブオーディオ	レガシーオーディオ
最大オーディオサンプルレート: 48 kHz	最大オーディオサンプルレート: 8 kHz
ステレオチャネル	モノラルチャネル

ヒント:

RHEL 8.xではPulseAudio 13.99以降を使用してください。

SUSE 15.3ではPulseAudio 14.2以降を、SUSE 15.2ではPulseAudio 13.0以降を使用してください。

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:

許可リストを使用して、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して、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、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/2203-ltsr/media/viewport.png)

システム要件

Windows クライアント:

- Citrix Workspaceアプリ 1809 for Windows以降

リナックス VDA:

- VDA オペレーティングシステム: Ubuntu 20.04, Ubuntu 18.04, RHEL 7.8, RHEL 8.2, RHEL 8.1, SLES 12.5
- VDA上のブラウザ: Citrix® ブラウザコンテンツリダイレクト拡張機能が追加された Google Chrome v66 以降

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のURLの許可リストと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/2203-ltsr/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/2203-ltsr/media/allow-list.png)

   許可リストを設定する画像(/en-us/linux-virtual-delivery-agent/2203-ltsr/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。

ブロックリストを設定するイメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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で**Chromeに追加**をクリックして、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。これにより、VDA上のブラウザが、アクセスしようとしているURL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のに役立ちます。

> **重要:**
>
> 拡張機能はクライアントには不要です。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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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）は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、Webサイトを表示します。

その後、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ブレンドし直します。

ビューポートとクライアントのイメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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

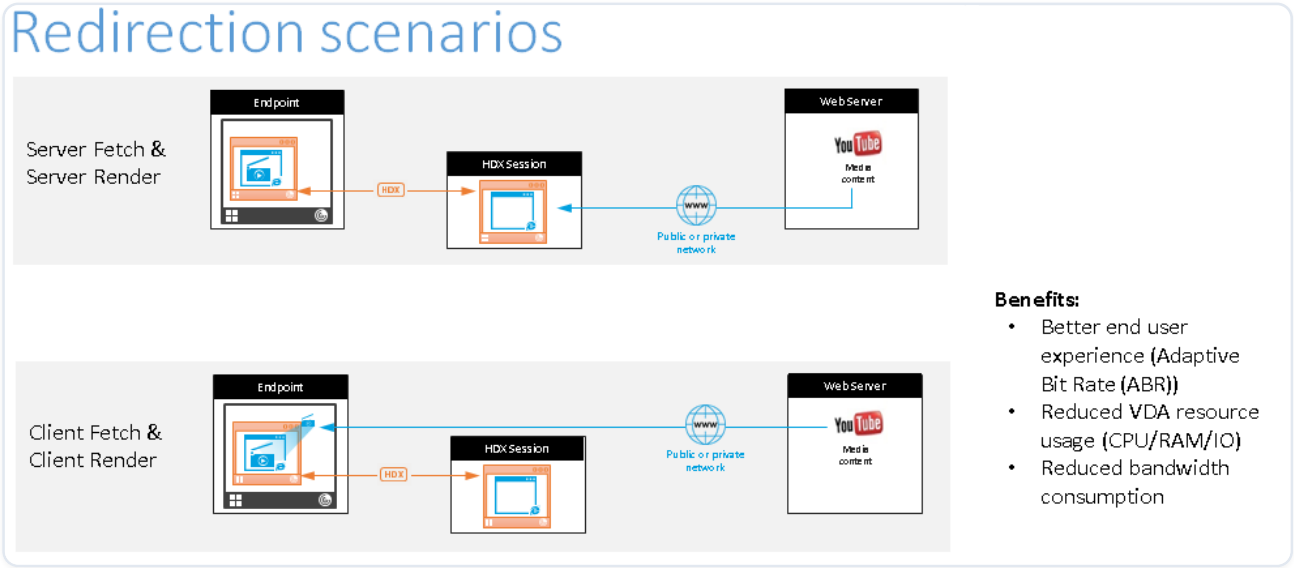
拡張機能アイコンの色は、Chrome拡張機能のステータスを示します。色は次の3つのいずれかです。

- 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. クライアントでレンダリングされたビューポート、または仮想デスクトップにブレンドされたビューポート
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フェッチするシナリオを次に示します。



- **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に戻されることがあります。このような場合、VDA上のブラウザーはページを再読み込みしてサーバー上でレンダリングできます。

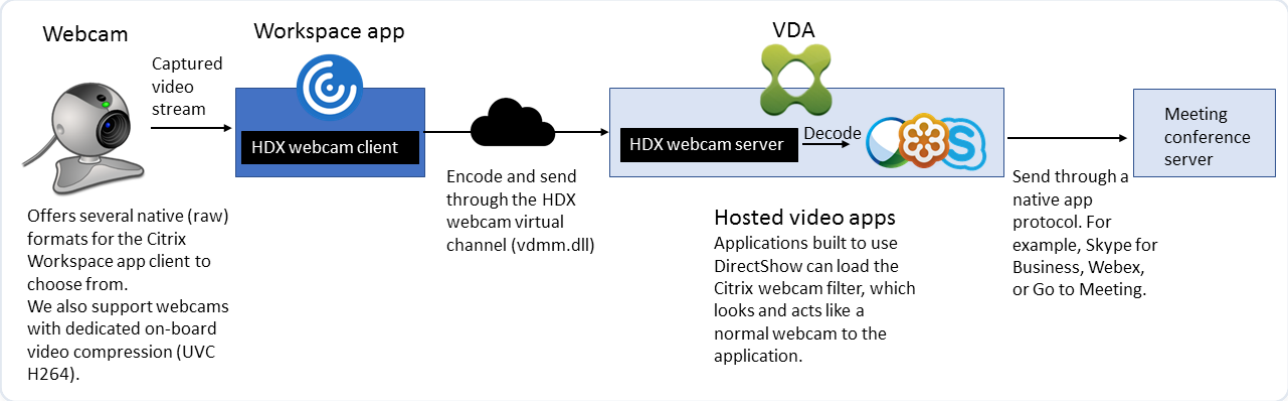
HDX™ Webカメラビデオ圧縮

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 Webカメラビデオ圧縮を使用してWebカメラを使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 Webカメラビデオ圧縮を使用することをお勧めします。

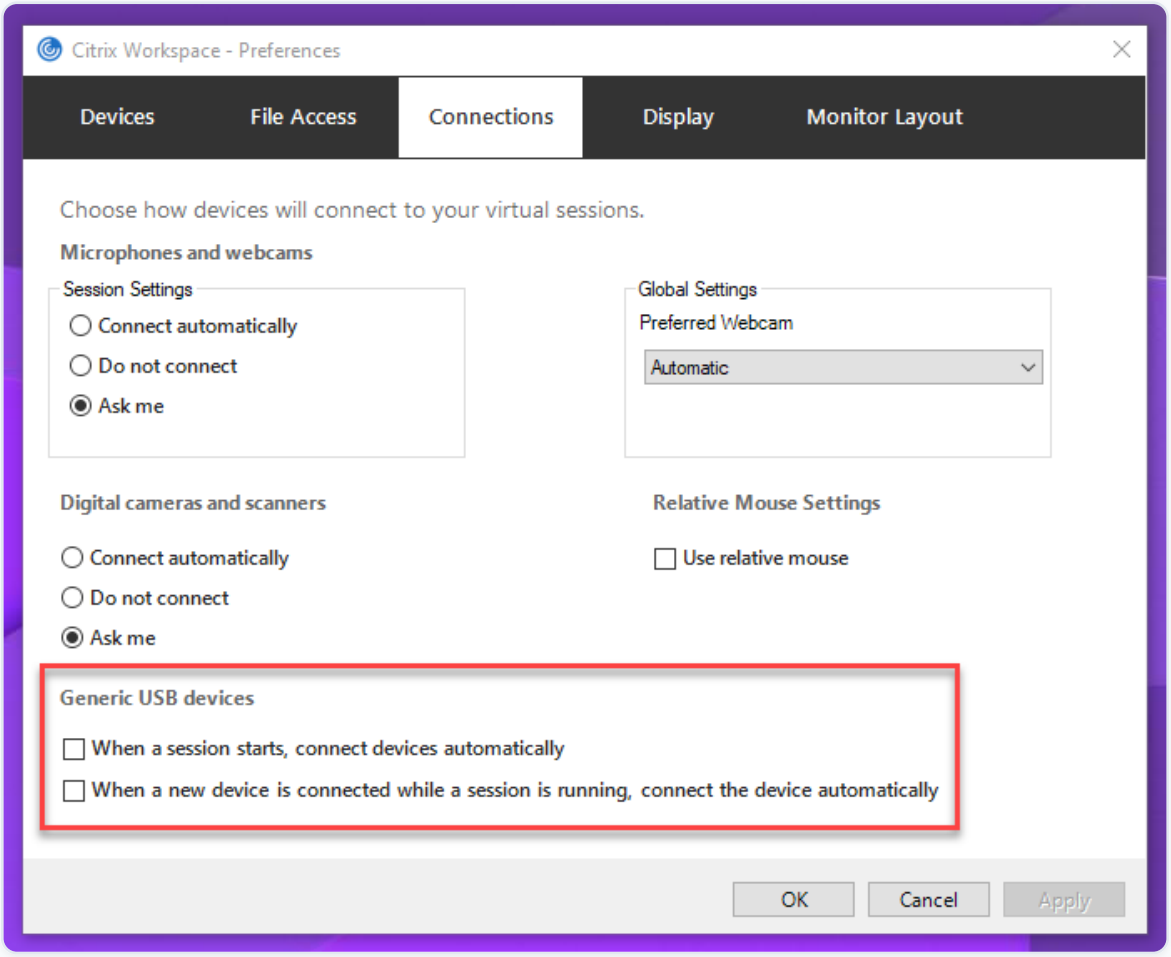
HDX Webカメラビデオ圧縮は、**最適化された**Webカメラモードとも呼ばれます。このタイプのWebカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDX Webカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントはWebカメラとの通信を処理します。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ビデオを送信します。サーバーはWebカメラと直接やり取りしませんが、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Webカメラの解像度は、48x32から1920x1080の範囲です。
- Webカメラを使用しているときは、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 8.4
- RHEL 8.3
- RHEL 7.9/CentOS 7.9
- ウブントゥ 20.04
- ウブントゥ 18.04
- デビアン 10
- SUSE 15.3

- SUSE 15.2
- SUSE 12.5

サポートされているCitrix Workspaceアプリ

HDXウェブカメラビデオ圧縮は、Citrix Workspaceアプリの次のバージョンをサポートしています。

プラットフォーム	プロセッサ
Windows版 Citrix Workspaceアプリ	Windows向けCitrix Workspaceアプリは、XenAppおよびXenDesktop 7.17以降で32ビットおよび64ビットアプリのWebカメラビデオ圧縮をサポートしています。それ以前のバージョンでは、Windows向けCitrix Workspaceアプリは32ビットアプリのみをサポートしています。
Chrome向けCitrix Workspaceアプリ	一部のARM ChromebookはH.264エンコーディングをサポートしていないため、32ビットアプリのみが最適化されたHDX Webカメラビデオ圧縮を使用できます。

完全にテストされたWebカメラ

Web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は異なります。Citrix®は、初期機能検証のために以下のWebカメラを使用しています。

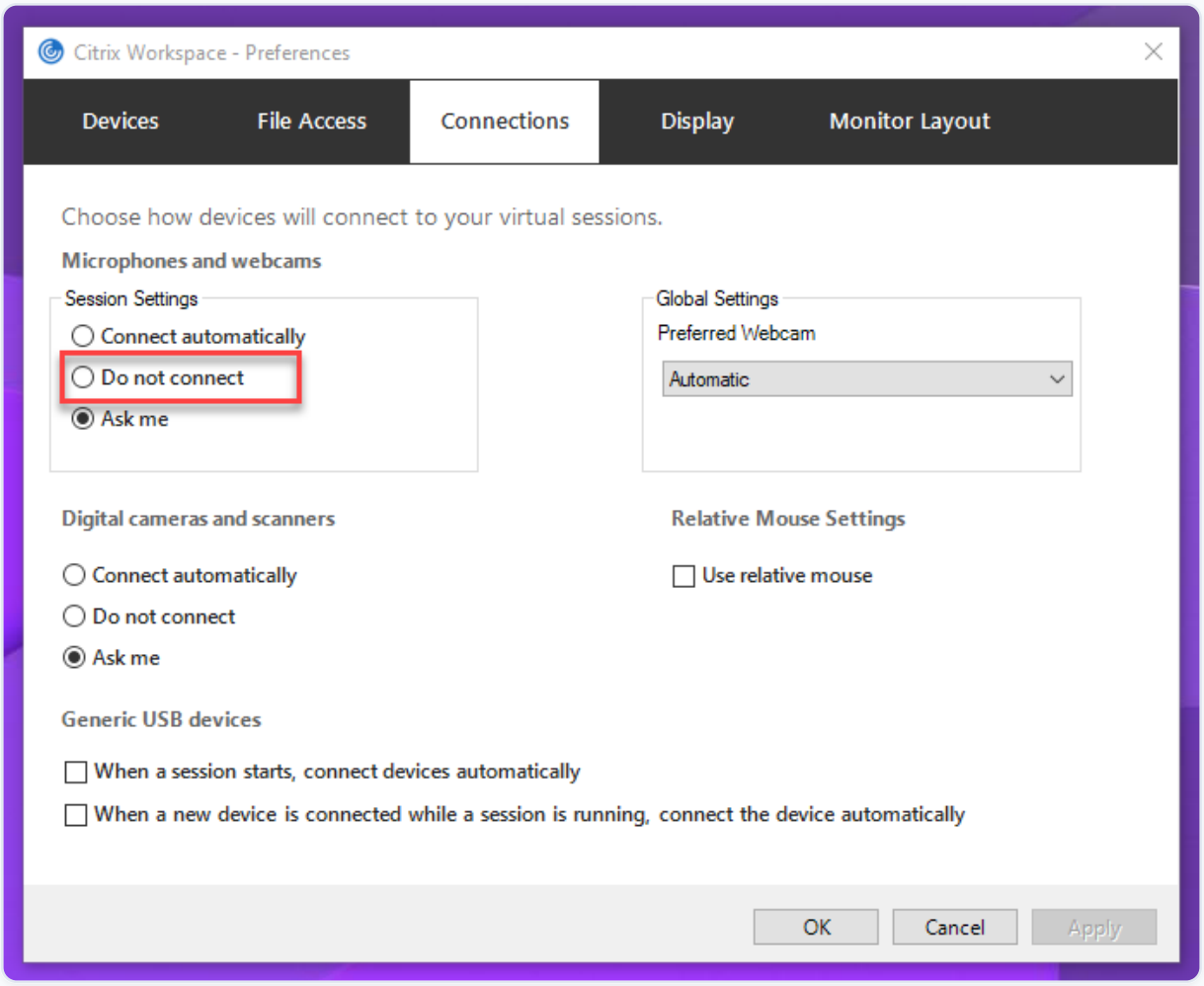
- ロジクール HD ウェブカメラ C270
- ロジクール ウェブカメラ C930e
- マイクロソフト LifeCam HD3000

構成

この機能はデフォルトで有効になっています。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、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、`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、最新のカーネルバージョンに更新するために以下のコマンドを実行します。

   ```
   sudo apt-get update

   sudo apt-get dist-upgrade
   ```

```
sudo reboot
```

VDAがSUSE 15.3、SUSE 15.2、またはSUSE 12.5に展開されている場合は、最新のカーネルバージョンに更新し、再起動するために以下のコマンドを実行します。

```
zypper up kernel-default

reboot
```

ctxwcamcfg.shスクリプトは、次のことを支援します。

a. VDAに`kernel-devel`とDynamic Kernel Module Support (DKMS)プログラムをインストールします。

- `kernel-devel` は、対応するバージョンの仮想ウェブカメラカーネルモジュールの構築に使用されます。
- DKMS は、仮想ウェブ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際、`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。
>
> - Enterprise 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` の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 のウェブカメラサービス - `ctxwcamsd` - が再起動し、HDX ウェブ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 にインターネットアクセスがない場合、別のマシンで v4l2loopback カーネルモジュールを構築し、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 `/var/xdl/` の下のビルドマシンにコピーします。

   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構築します。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ウェブ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。

ドメインに参加していないLinux VDA

概要

ドメインに参加していないVDAは、VDAおよびユーザー認証のためにVDAをActive Directoryドメインに参加させる必要性をなくします。ドメインに参加していないVDAを作成する際、VDAをクラウドコントロールプレーンに登録するための公開鍵と秘密鍵のペアを生成します。これにより、Active Directoryドメインへの参加は不要になります。ユーザーがドメインに参加していないVDAからセッションを起動すると、VDAはCitrix Workspaceアプリにログオンするためにユーザーが使用するユーザー名を使用してローカルマッピングアカウントを作成します。VDAは、ローカルマッピングアカウントがSSOおよびセッション再接続に使用するランダムなパスワードを割り当てます。ランダムなパスワードを変更すると、SSOおよびセッション再接続は失敗します。SSOを無効にするには、「非SSO認証」を参照してください。

> **重要:**
>
> - ドメインに参加していないVDAは、Citrix DaaSでサポートされています。

    - コントロールプレーンはCitrix DaaS上に展開されている必要があります。
    - ドメインに参加していないVDAは、パブリッククラウドまたはオンプレミスのデータセンターに展開できます。Citrix DaaSのコントロールプレーンが、ドメインに参加していないVDAを管理します。
    - Citrix Cloud ConnectorをバイパスするようにRendezvous V2を構成できます。そうしないと、VDAをコントロールプレーンに接続するためにCloud Connectorをインストールする必要があります。

- ドメインに参加していないVDAを作成するには、Machine Creation Services™ (MCS)を使用する必要があります。
    - MCSはベアメタルサーバーをサポートしていません。

ドメインに参加していないLinux VDAで利用可能な機能

ドメインに参加していないVDAで、指定された属性を持つローカルユーザーを作成する

ドメインに参加していないVDAでホストされているセッションを開くと、VDAはデフォルト属性を持つローカルユーザーを自動的に作成します。VDAは、Citrix Workspace™アプリにログオンするために使用したユーザー名に基づいてローカルユーザーを作成します。ユーザーのユーザー識別子（UID）、グループID（GID）、ホームディレクトリ、ログインシェルなどのユーザー属性を指定することもできます。この機能を使用するには、次の手順を完了します。

1. 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 create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\LocalMappedAccount" -t
"REG_DWORD" -v "CreateWithUidGid" -d "0x00000001" --force
```

2. VDAのインストールパスの下にある`/var/xdl/getuidgid.sh`スクリプトで、次の属性を指定します。

属性	必須またはオプション	説明
`uid`	必須	ユーザー識別子 (UID) は、Linux がシステム上の各ユーザーに割り当てる番号です。これにより、ユーザー

属性	必須またはオプション	説明
		がアクセスできるシステムリソースが決まります。
`gid`	必須	グループ識別子 (GID) は、特定のグループを表すために使用される番号です。
`homedir`	オプション	Linux のホームディレクトリは、特定のユーザーのためのディレクトリです。
`shell`	オプション	ログインシェルは、ユーザーがユーザーアカウントにログインしたときに与えられるシェルです。

以下は、`getuidgid.sh` スクリプトの例です。

> **注:**
>
> スクリプトで指定された属性が有効であることを確認してください。

```
#!/bin/bash



################################################################################
########

#

# Citrix Virtual Apps™ & Desktops For Linux Script: Get uid and gid for
the user

#

# Copyright (c) Citrix Systems, Inc. All Rights Reserved.

#
```

```
export LC_ALL="en_US.UTF-8"


function get_uid_gid_for_user()
{
echo "uid:12345"
echo "gid:1003"
echo "homedir:/home/$1"
echo "shell:/bin/sh"
}


get_uid_gid_for_user $1
```

非SSO認証

デフォルトでは、Linux VDA はシングルサインオン (SSO) が有効になっています。ユーザーは、Citrix Workspace アプリと VDA セッションに、1組の資格情報を使用してログオンします。 ユーザーが異なる資格情報を使用して VDA セッションにログオンするようにするには、Linux VDA で SSO を無効にします。詳細については、「非SSO認証」を参照してください。

ランデブー V2

ドメインに参加していない VDA は、Citrix Cloud Connector をバイパスするために Rendezvous V2 を使用することがサポートされています。詳細については、「Rendezvous V2」を参照してください。

ドメインに参加していない Linux VDA の作成

Citrix DaaS でドメインに参加していない Linux VDA を作成するには、MCS を使用します。詳細については、「ドメインに参加していない Linux VDA の作成」を参照してください。

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間を使用	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)
全体的なセッション帯域幅制限	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帯域幅制限の割合	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帯域幅制限	LimitUSBBw	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帯域幅の割合	LimitUSBBwPercent	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
クリップボードリダイレクト帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクト帯域幅制限の割合	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帯域幅制限の割合	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帯域幅制限の割合	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
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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
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\クライアントの自動再接続	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 s
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)
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	AutoCreatePDFPrinter	ユーザー	印刷	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
プリンタードライバーのマッピングと互換性	DriverMappingList	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	AllowClipboardRedir	ユーザー	クリップボード	許可 (1)
クライアントUSBデバイスリダイレクト	AllowUSBRedir	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	USBDeviceRules	ユーザー	USB	"\0"
動画圧縮	MovingImageCompressionConfiguration	ユーザー	シンワイヤー	有効 (1)
追加の色圧縮	ExtraColorCompression	ユーザー	シンワイヤー	無効 (0)
ターゲットの最小フレームレート	TargetedMinimumFramesPerSecond	ユーザー	シンワイヤー	10 フレーム/秒
目標フレームレート	FramesPerSecond	ユーザー	シンワイヤー	30 フレーム/秒
画質	VisualQuality	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する	VideoCodec	ユーザー	シンワイヤー	優先する場合に使用 (3)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ビデオコーデックにハードウェアエンコードを使用する	UseHardwareEncodingForVideoCodec	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	AllowVisuallyLosslessCompression	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	OptimizeFor3dWorkload	ユーザー	シンワイヤー	無効 (0)
シンプルグラフィックの優先色深度	PreferredColorDepth	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	SoundQuality	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	AllowMicrophoneRedir	ユーザー	オーディオ	許可 (1)
セッションの最大数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオンの許容数	ConcurrentLogonsTolerance	コンピューター	負荷管理	2
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

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
Speexの最大品質	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ディスクドライブ	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)
キーボードの自動表示	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	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

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
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)
切断された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、User Datagram Protocol (UDP) 経由のオーディオをサポートしています。Linux VDA はサポートしていません。詳細については、「User Datagram Protocol (UDP) 経由のオーディオリアルタイム転送」を参照してください。

Citrix Studio でセッション接続タイマーを構成するには、以下のCitrixポリシー設定を使用できます。

- **セッションアイドルタイマー**: アイドル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、アクティブなセッションが設定された時間内にユーザー入力を受信しなかった場合、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。

ポリシー設定を更新する際は、展開全体で一貫性があることを確認してください。

アイドルセッションの時間制限が切れると、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、ユーザー入力を提供してアイドルタイマーをリセットしてください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/warning-message-idle-session-timeout.png)

Citrix Studioバージョン7.12以降では、以下のポリシーを構成できます。

- 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コーデックで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でトレードオフを制御することが可能です。Speexのエンコードプロセスは、ほとんどの場合、0から10まで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）に応じて、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、モードが異なるとビットレートも異なります。最適なSpeex品質とは、以下の条件を満たす最大値の場合です。

  - 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

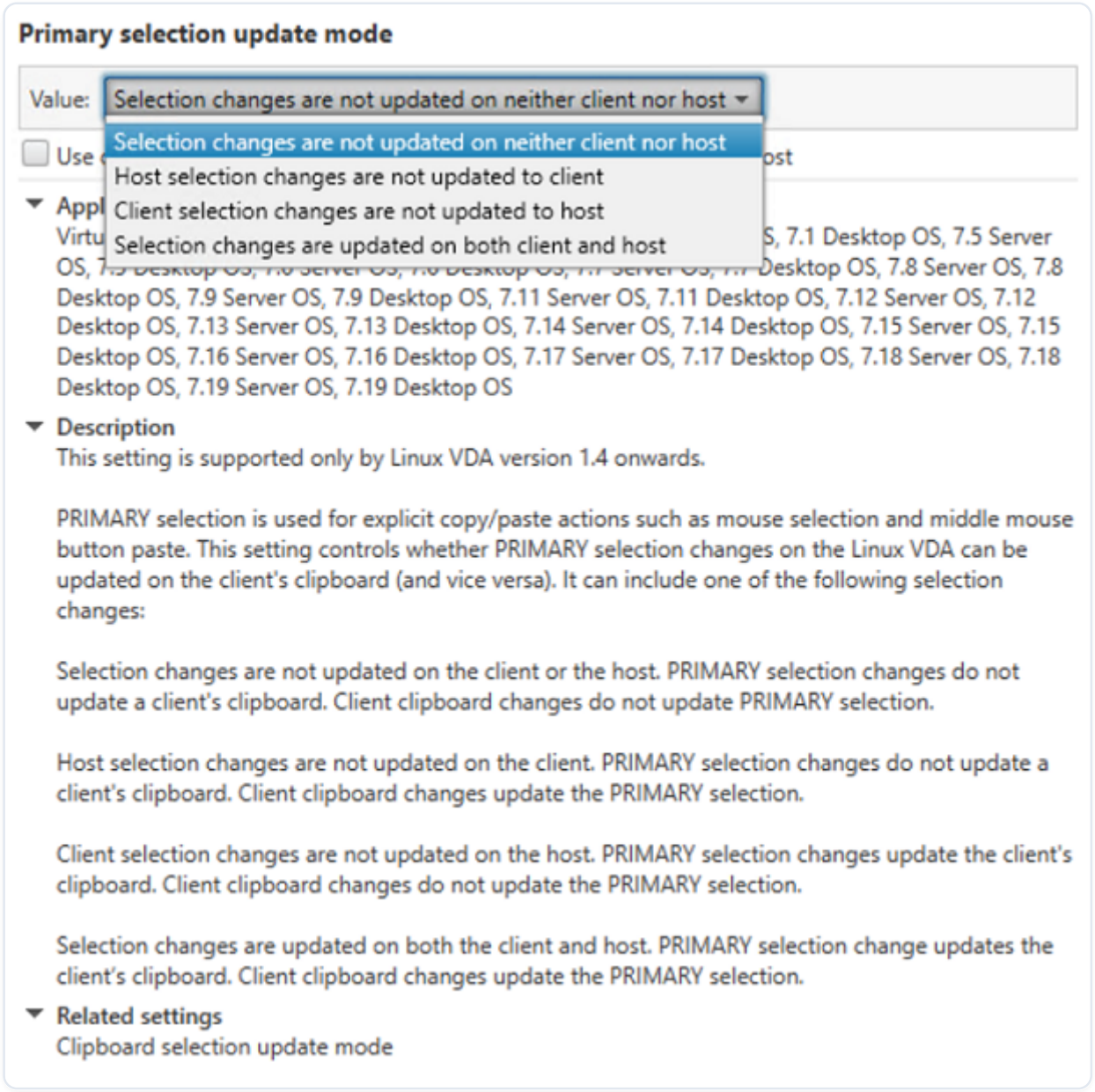
値 (列挙型):

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。



- **クライアントとホストのどちらでも選択の変更は更新されません**

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ません**

  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ません**

  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ます**

  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

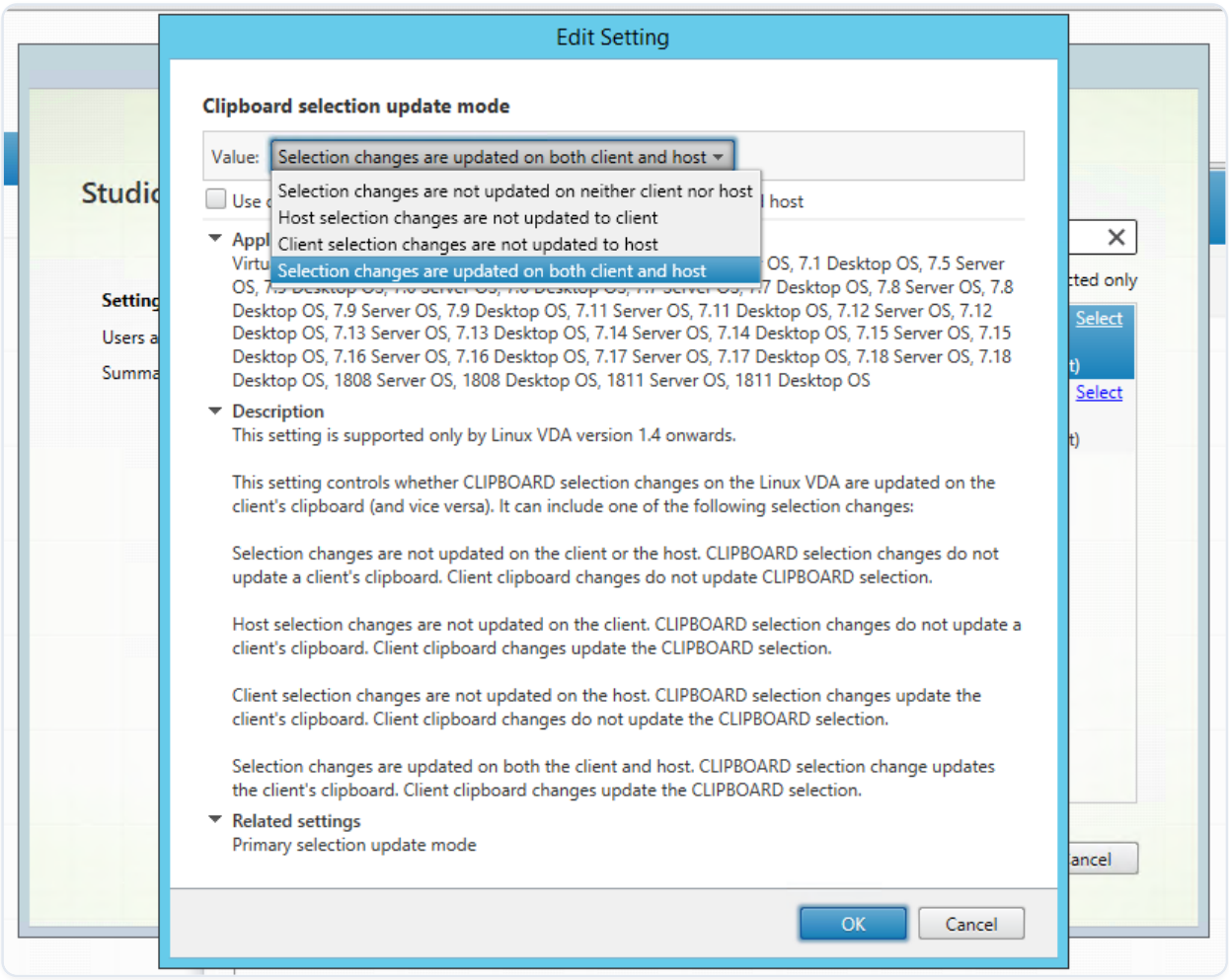
値 (列挙型):

デフォルト値: 3

詳細:

クリップボード選択は、データをいくつか選択し、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」するよう要求する場合に使用されます。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に固有のものです。

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。



- **選択の変更はクライアントとホストのどちらにも更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **ホストの選択変更はクライアントに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、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変更はホストに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDA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る**

Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、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: 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と貼り付け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。

印刷

このセクションには、次のトピックが含まれています。

- 印刷のベストプラクティス
- PDF印刷

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種類あります（postscript、pcl5、pcl6）。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**ポリシーを構成することを選択できます。

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\. **次で置き換える**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/2203-ltsr/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**ポリシーの他のオプションはサポートされていません。**次で置き換える**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いた場合、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したり、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したりすると、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。次のコマンドを実行して、印刷デーモンが実行されていることを確認してください。**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手順

CUPSログを収集するには、次のコマンドを実行してCUPSサービスファイルを構成します。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行われます。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、**~/.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。

> **ppdpath**、**model**、および **drivertype** の各フィールドは、マッピング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、同時に設定することはできません。

- 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 driver** 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種類（PostScript、PCL5、PCL6）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の種類を設定できます。

  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の種類がPCL5の場合、**drivertype** を次のように設定します。

```
drivertype=pcl5
```

出力サイズがゼロになる

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種類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種類が正しい場合は、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. ログオフして再度ログオンし、構成の変更を読み込みます。
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の種類に依存し

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てください。

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の場合は、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** をクリックします。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、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、**.CtxlpProfile$CLIENT_NAME** でドライバーのPPDファイルパスを構成します。

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が遅い

大容量ドキュメントをローカルクライアントプリンターで印刷すると、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、次の行をデフォルトポリシーのXMLブロックに追加します。

```
<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>
```

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、Citrix Workspaceアプリの以下のバージョンで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てください。

- **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。

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 に安全な方法でリモートから簡単にアクセスできるようにします。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およびプロセスは、Citrix Virtual Apps and Desktops の展開に必要な要件および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することはできません。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

オンは管理者による介入なしに行われます。クライアント上の Citrix Workspace™ アプリは、ユーザーが 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にアクセスするのに役立ちます。

- 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

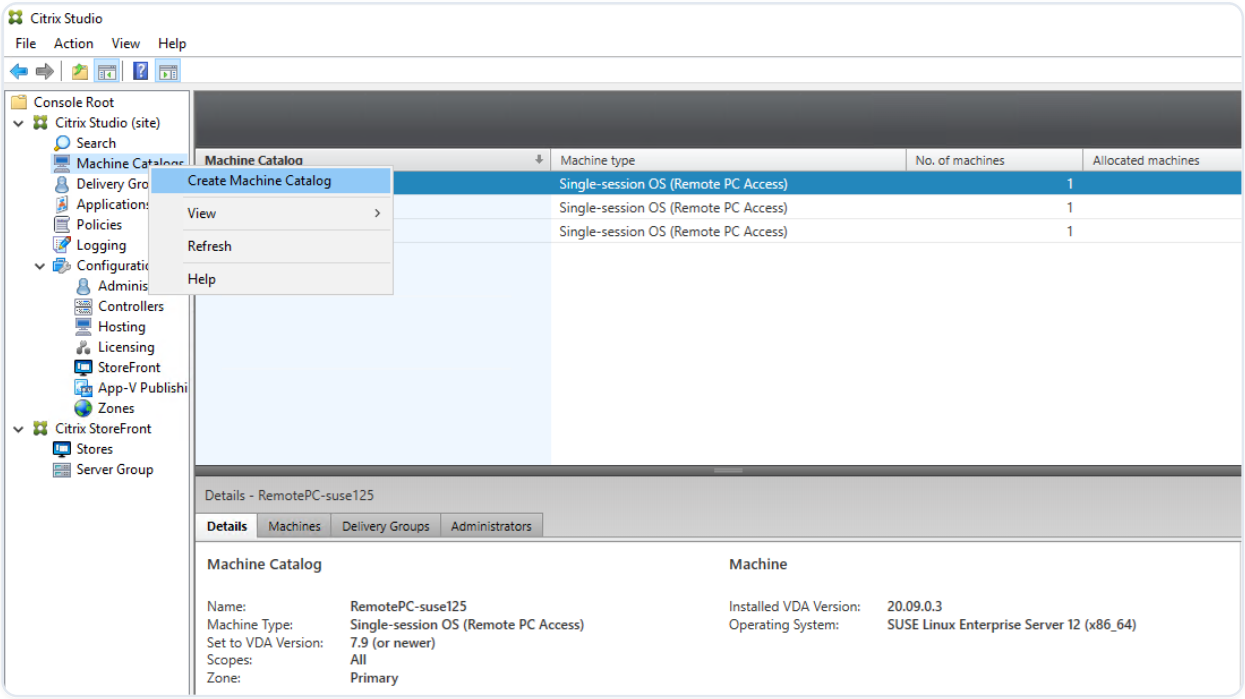
Linux PC セッションを配信するには、ターゲット PC に Linux VDA をインストールし、**Remote PC Access** タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内の PC を利用できるようにするデリバリーグループを作成します。以下のセクションで手順を詳しく説明します。

ステップ 1 - ターゲット PC に Linux VDA をインストールする

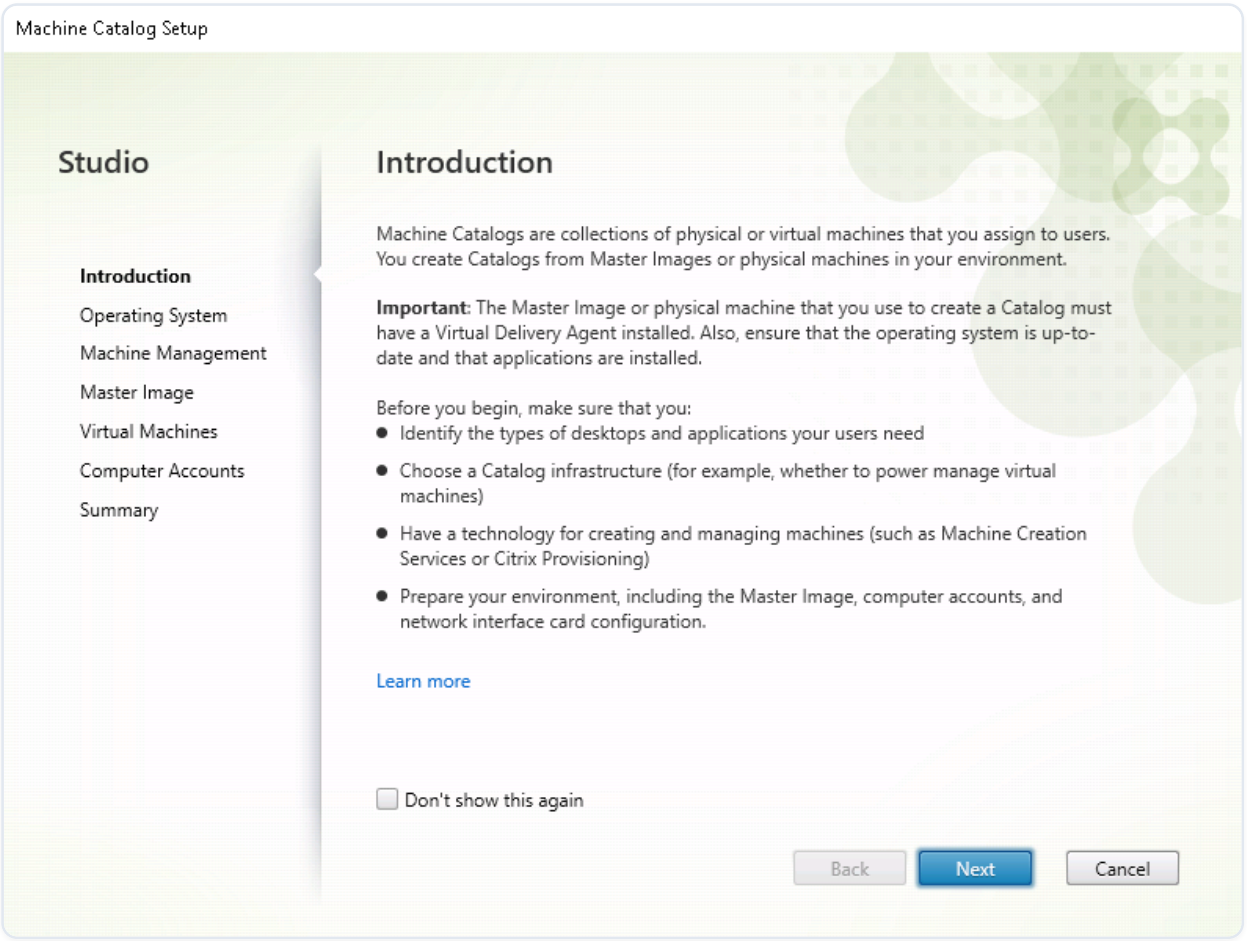
Linux VDA のインストールには、easy install を使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE` 変数の値を `Y` に設定します。

ステップ 2 - Remote PC Access タイプのマシンカタログを作成する

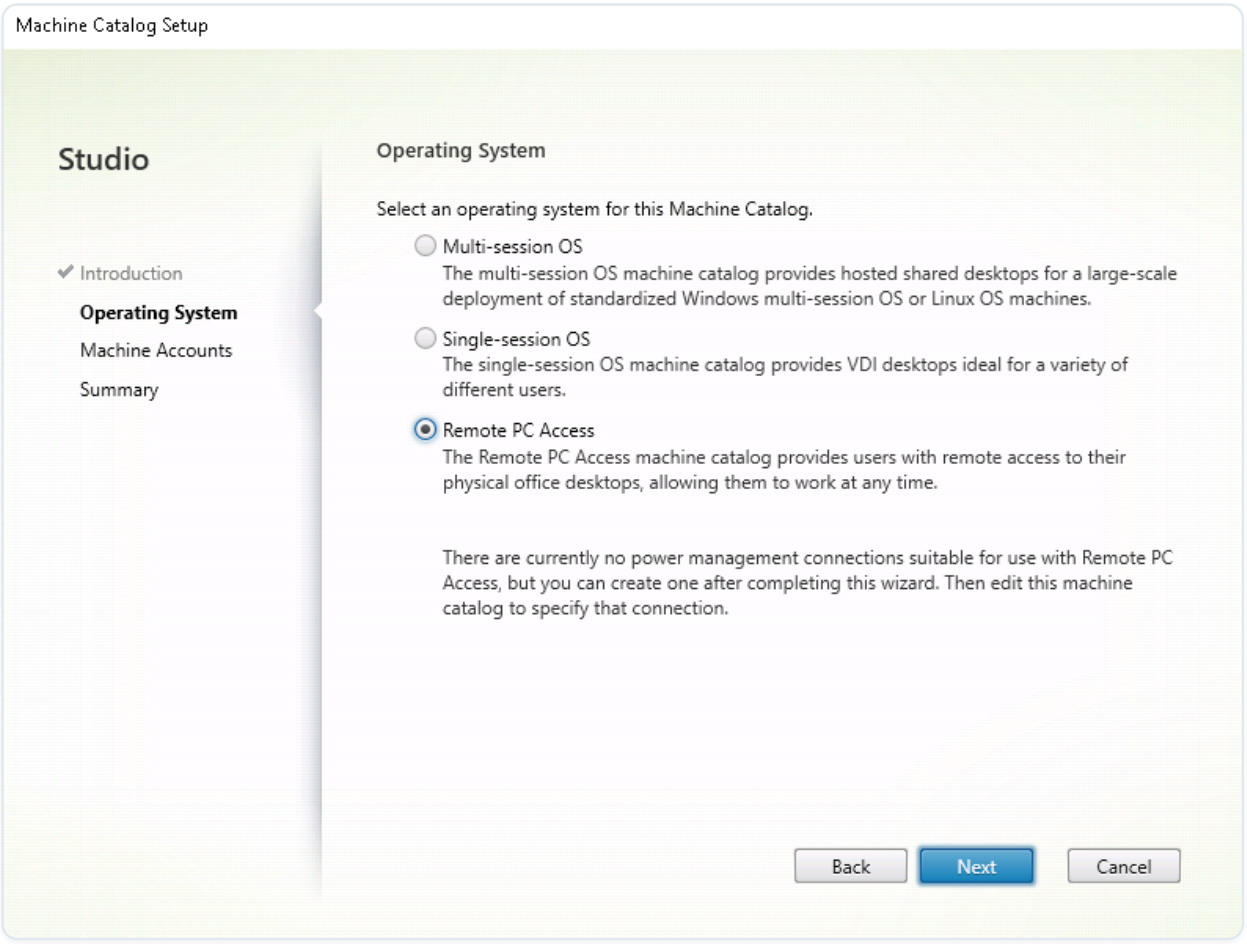
1. Citrix Studio で、**マシンカタログ** を右クリックし、ショートカットメニューから **マシンカタログの作成** を選択します。



2. **はじめに** ページで **次へ** をクリックします。



3. **オペレーティングシステム** 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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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をアクセス要求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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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イメージ(/en-us/linux-virtual-delivery-agent/2203-ltsr/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:
  - Citrix DaaS™ (旧 Citrix バーチャルアプリおよびデスクトップ サービス)
  - Citrix 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 を使用する場合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 の 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. `sudo ethtool <NIC>` コマンドを実行して、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>` のような行があり、その行の `<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>` のような行があり、その行の `<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。

セッション

このセクションには、次のトピックが含まれています。

- アダプティブトランスポート(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/session/adaptive-transport.html)
- アプリケーションの公開(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/session/publish-apps.html)
- セッションの信頼性(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/session/session-reliability.html)
- ランデブー V1
- ランデブー V2
- TLSを使用したユーザーセッションの保護(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/session/secure-user-sessions-using-tls.html)
- DTLSを使用したユーザーセッションの保護(/ja-jp/linux-virtual-delivery-agent/2203-ltsr/configure/session/secure-user-sessions-using-dtls.html)

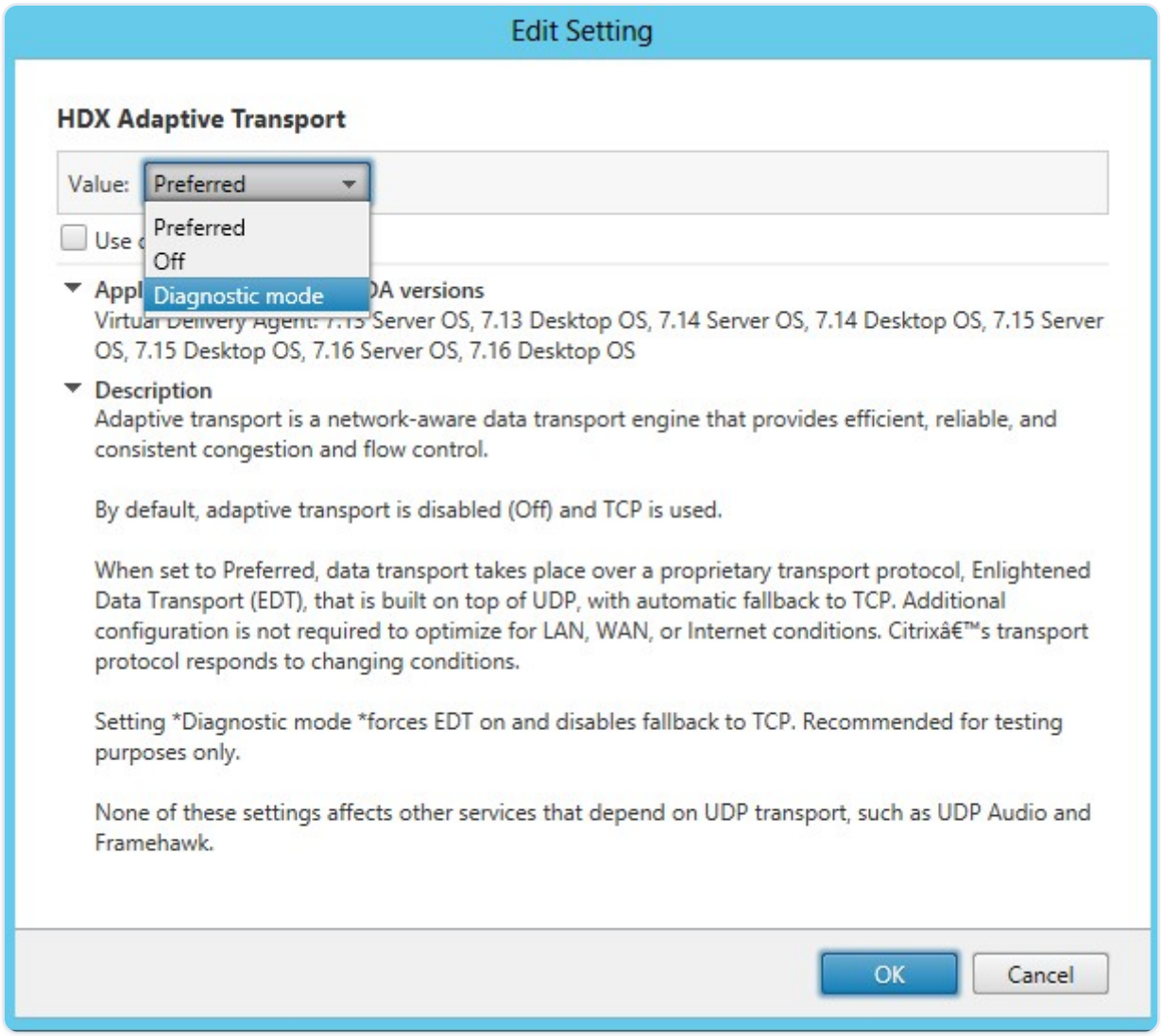
アダプティブトランスポート

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でより対話的です。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。



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どうかを確認するには、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パケットの断片化が防止されます。

最小要件:

- リナックス VDA 2012
- Windows向け Citrix Workspace™ アプリ 1911
- 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を構成できます。詳細については、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されることを保証できません。**レジストリエディター**の使用は、お客様ご自身の責任において行ってください。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でのEDT MTU検出の有効化または無効化

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。

クライアントでのEDT MTU検出の制御

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を機能させるには、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は効果がありません。

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。
- 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、タスクバーボタンを結合しないようにタスクバープロパティを設定します。

ランデブー V1

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セッションはCitrix Cloud Connector™をバイパスし、Citrix Cloudコントロールプレーンに直接かつ安全に接続できます。

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 DaaS (旧称 Citrix Virtual Apps and Desktops™ service)。
- Linux VDAバージョン2112以降。
- CitrixポリシーでRendezvousプロトコルを有効にします。詳細については、「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。詳細については、Citrix Cloudドキュメントのインターネット接続要件セクション（Virtual Apps and Desktopサービスの下）および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 DaaS Remote PowerShell SDKを使用して、コマンド `Set-BrokerSite -DnsResolutionEnabled $true` を実行します。Citrix DaaS Remote PowerShell SDKの詳細については、「SDKとAPI」を参照してください。

プロキシ構成

VDAは、HTTPプロキシを介したRendezvous接続の確立をサポートしています。

プロキシに関する考慮事項

Rendezvousでプロキシを使用する場合は、次の点を考慮してください。

- 非透過型HTTPプロキシがサポートされています。
- パケットの復号化と検査はサポートされていません。VDAとGateway Service間のICA®トラフィックが傍受、復号化、または検査されないように例外を設定してください。そうしないと、接続が切断されます。

- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、Negotiate認証スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

  > **注:**
  >
  > Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名（SPN）を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAは、セッションを確立する際に`HTTP/<proxyURL>`の形式でSPNを生成します。ここで、プロキシURLは**Rendezvousプロキシ**ポリシー設定から取得されます。SPNを作成しない場合、認証は失敗します。

- HTTPプロキシの場合、ICAのトランスポートプロトコルとしてTCPを使用します。

非透過型プロキシ

ネットワークで非透過型プロキシを使用する場合は、Rendezvousプロキシ構成設定を構成します。この設定を有効にすると、VDAが使用するプロキシを認識できるようにHTTPプロキシアドレスを指定します。例：

- プロキシアドレス: `http://<URL or IP>:<port>`

Rendezvousの検証

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、次の手順に従ってください。

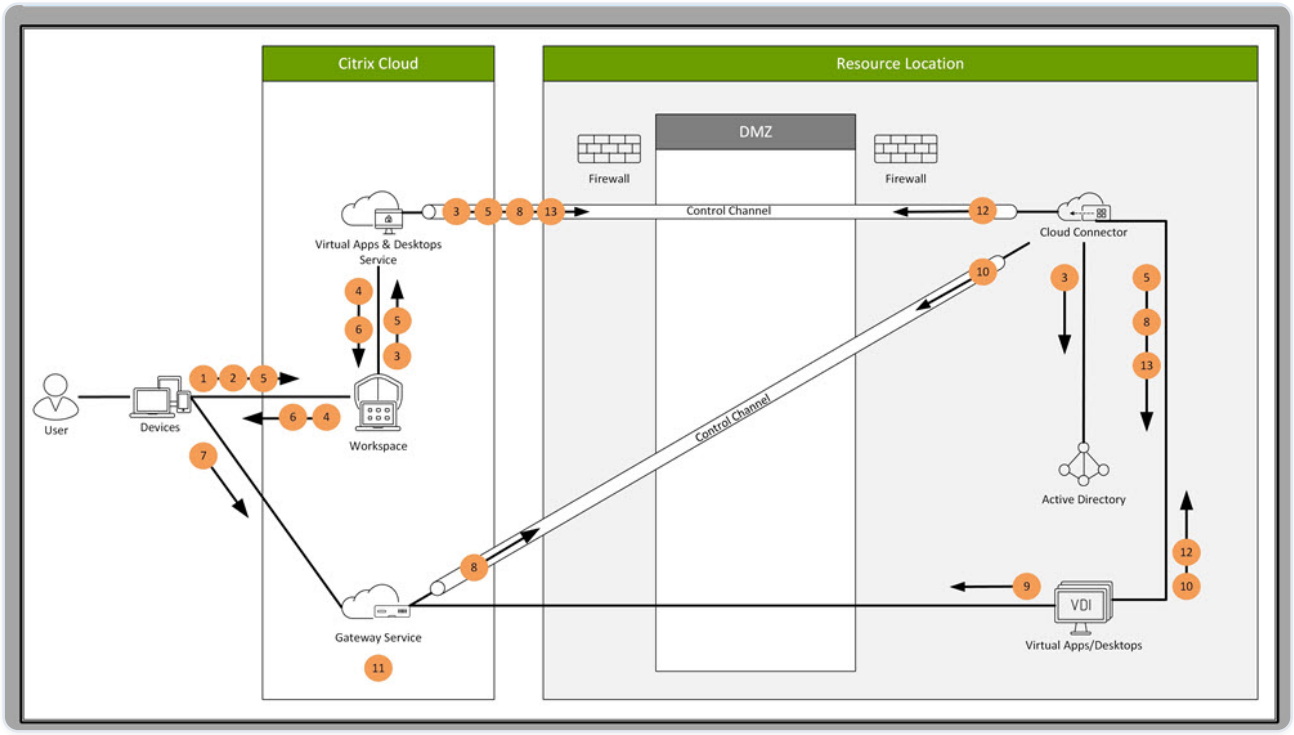
1. VDAでターミナルを起動します。
2. `/opt/Citrix/VDA/bin/ctxquery -f iP`を実行します。
3. トランスポートプロトコルは接続の種類を示します。
   - TCP ランデブー: TCP - TLS - CGP - ICA
   - EDT ランデブー: UDP - DTLS - CGP - ICA
   - Cloud Connectorを介したプロキシ: TCP - CGP - ICA

> **ヒント:**

> Rendezvousが有効になっている状態で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。

Rendezvousの仕組み

この図は、Rendezvous接続フローの概要です。



フローを理解するには、次の手順に従ってください。

1. Citrix Workspaceにアクセスします。
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 DaaS™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 DaaS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 DaaS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、接続情報をCloud Connectorに送信します。Cloud Connectorは、その接続がRendezvous接続であるかどうかを判断し、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 DaaSとライセンスを検証します。
13. Citrix DaaSは、Cloud Connectorを介してセッションポリシーをVDAに送信します。これらのポリシーが適用されます。

ランデブー V2

Citrix Gateway service を使用する場合、Rendezvous プロトコルにより、トラフィックは Citrix Cloud™ Connectors をバイパスし、Citrix Cloud コントロールプレーンに直接かつ安全に接続できます。

考慮すべきトラフィックには2種類あります。1) VDA登録とセッションブローカリングのための制御トラフィック、2) HDX™ セッショントラフィックです。

Rendezvous V1では、HDXセッション トラフィックがCloud Connectorsをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックをCloud Connectorsがプロキシする必要があります。

標準のADドメイン参加マシンとドメイン非参加マシンは、シングルセッションおよびマルチセッションLinux VDAでRendezvous V2を使用する場合にサポートされます。ドメイン非参加マシンでは、Rendezvous V2により、HDXトラフィックと制御トラフィックの両方がCloud Connectorsをバイパスできます。

要件

Rendezvous V2を使用するための要件は次のとおりです。

- Citrix Workspace™ および Citrix Gateway service を使用した環境へのアクセス。
- コントロールプレーン: Citrix DaaS (旧 Citrix Virtual Apps and Desktops™ service)。
- VDAバージョン2201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 Desktopサービスの下）のインターネット接続要件セクションおよびKnowledge Centerの記事CTX270584を参照してください。
- VDAは、前述のアドレスに接続できる必要があります。
  - TCP Rendezvousの場合はTCP 443で。
  - EDT Rendezvousの場合はUDP 443で。

プロキシ構成

Linux VDA は現在、プロキシ経由の Rendezvous V2 接続をサポートしていません。

Rendezvous V2 の構成方法

環境で Rendezvous を構成する手順は次のとおりです。

1. すべての要件 が満たされていることを確認してください。
2. VDA のインストール後、次のコマンドを実行して必要なレジストリキーを設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"GctRegistration" -d "0x00000001" --force
```

3. VDA マシンを再起動します。
4. Citrix ポリシーを作成するか、既存のポリシーを編集します。
   - Rendezvous プロトコル設定を に設定します。
   - Citrix ポリシーフィルターが適切に設定されていることを確認してください。このポリシーは、Rendezvous を有効にする必要があるマシンに適用されます。
   - Citrix ポリシーが別のポリシーを上書きしないように、正しい優先順位を持っていることを確認してください。

Rendezvous の検証

セッションが Rendezvous プロトコルを使用しているかどうかを確認するには、ターミナルで `/opt/Citrix/VDA/bin/ctxquery -f iP` コマンドを実行します。

表示されるトランスポートプロトコルは、接続の種類を示します。

- TCPランデブー: TCP - TLS - CGP - ICA®
- EDTランデブー: UDP - DTLS - CGP - ICA
- Rendezvousではない: TCP - CGP - ICA

Rendezvous V2が使用されている場合、プロトコルバージョンは2.0と表示されます。

> **ヒント:**
>
> Rendezvousが有効な状態で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。

DTLS を使用してユーザーセッションを保護する

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。詳細については、「Transport Layer Security」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™ 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。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**/opt/Citrix/VDA/sbin** にある **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。

TLS を使用したユーザーセッションの保護

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のための 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のために TLS 暗号化を有効にするには、Linux VDA と Delivery Controller™ (Controller) の両方に証明書をインストールし、TLS 暗号化を有効にします。

Linux VDA に証明書をインストールする

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/2203-ltsr/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、**/opt/Citrix/VDA/sbin** ディレクトリにある `enable_vdassl.sh` スクリプトを使用して TLS 暗号化を有効（または無効）にします。スクリプト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。

SSL を有効にするためのスクリプト (/en-us/linux-virtual-delivery-agent/2203-ltsr/media/script-for-enabling-ssl.png)

ヒント: 各 Linux VDA サーバーにサーバー証明書をインストールし、各 Linux VDA サーバーとクライアントにルート証明書を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 TLS 暗号化を有効にすることはできません。

Controller の PowerShell ウィンドウで、ターゲットデリバリーグループの TLS 暗号化を有効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

注:

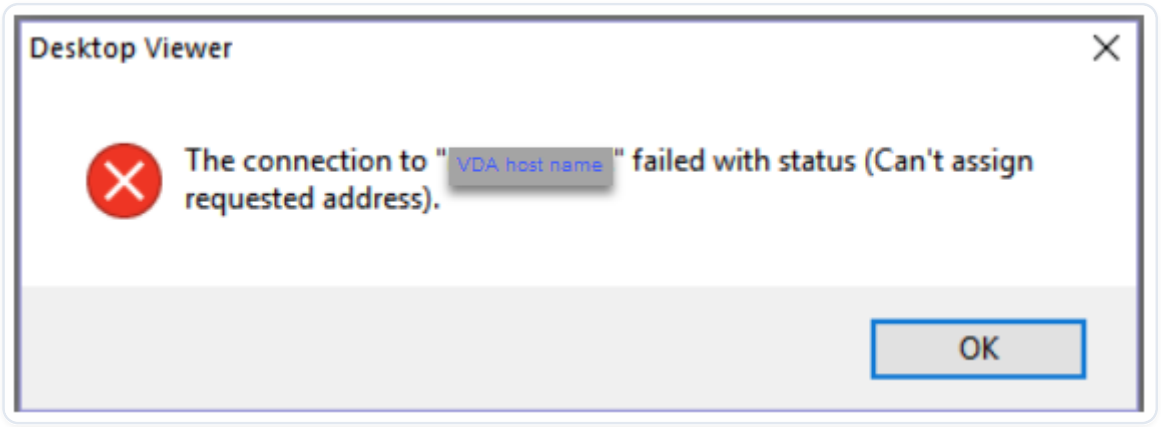
ICA® セッションファイルに VDA FQDN のみが含まれるようにするには、`Set-BrokerSite –DnsResolutionEnabled $true` コマンドも実行できます。このコマンドは DNS 解決を有効にします。DNS 解決を無効にすると、ICA セッションファイルは VDA IP アドレスを開示し、`SSLProxyHost` や `UDPDTLSPort` などの TLS 関連アイテムに対してのみ FQDN を提供します。

Controller で TLS 暗号化を無効にするには、次のコマンドを順番に実行します。

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 用 Citrix Workspace™ アプリで次の「要求されたアドレスを割り当てることができません」エラーが発生する可能性があります。



回避策として、**hosts** ファイルに次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。

セッションの信頼性

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導入しています。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時でも 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。セキュリティ上の目的から、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、次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の透過性レベル

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）を通じ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**

セッションでアクティビティ（マウスの動きや画面の更新など）がない場合、Linux VDAとクライアントの間で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の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でポリシー設定を通じてセッションの信頼性を有効にした後、VDAサービスとHDXサービスがこの順序で再起動されていることを確認してください。
2. VDAで、次のコマンドを実行して、セッションの信頼性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
   ```

   セッションの信頼性ポートに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"
   ```

USBリダイレクトの構成

USBデバイスは、Citrix Workspace™アプリと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、ローカルに接続されているかのようにUSBデバイスを使用できます。

ヒント:

ネットワーク遅延が100ミリ秒未満の場合、USBリダイレクトの使用をお勧めします。ネットワーク遅延が200ミリ秒を超える場合、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アプリ間のすべての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、Citrixが提供するオープンソースサービスであり、VHCIカーネルモジュールと通信します。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モジュールとして機能します。

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セッションで使用できない場合があります。USBデバイスは、適切に機能するためにドライバーに依存しており、一部のデバイスには特別なドライバーが必要です。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デバイスの接続性を確保するには、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。

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、予期しない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ます。

USBマスストレージデバイス	VID:PID	ファイルシステム
`Netac Technology Co., Ltd`	0dd8:173c	FAT32
`Kingston Datatraveler 101 II`	0951:1625	FAT32
`Kingston Datatraveler GT101 G2`	1567:8902	FAT32
`SanDisk SDCZ80 flash drive`	0781:5580	FAT32
`WD HDD`	1058:10B8	FAT32

USB 3D マウス	VID:PID
`3DConnexion SpaceMouse Pro`	046d: c62b

USB スキャナー	VID:PID
`Epson Perfection V330 photo`	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を有効にするか無効にするかを制御します。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

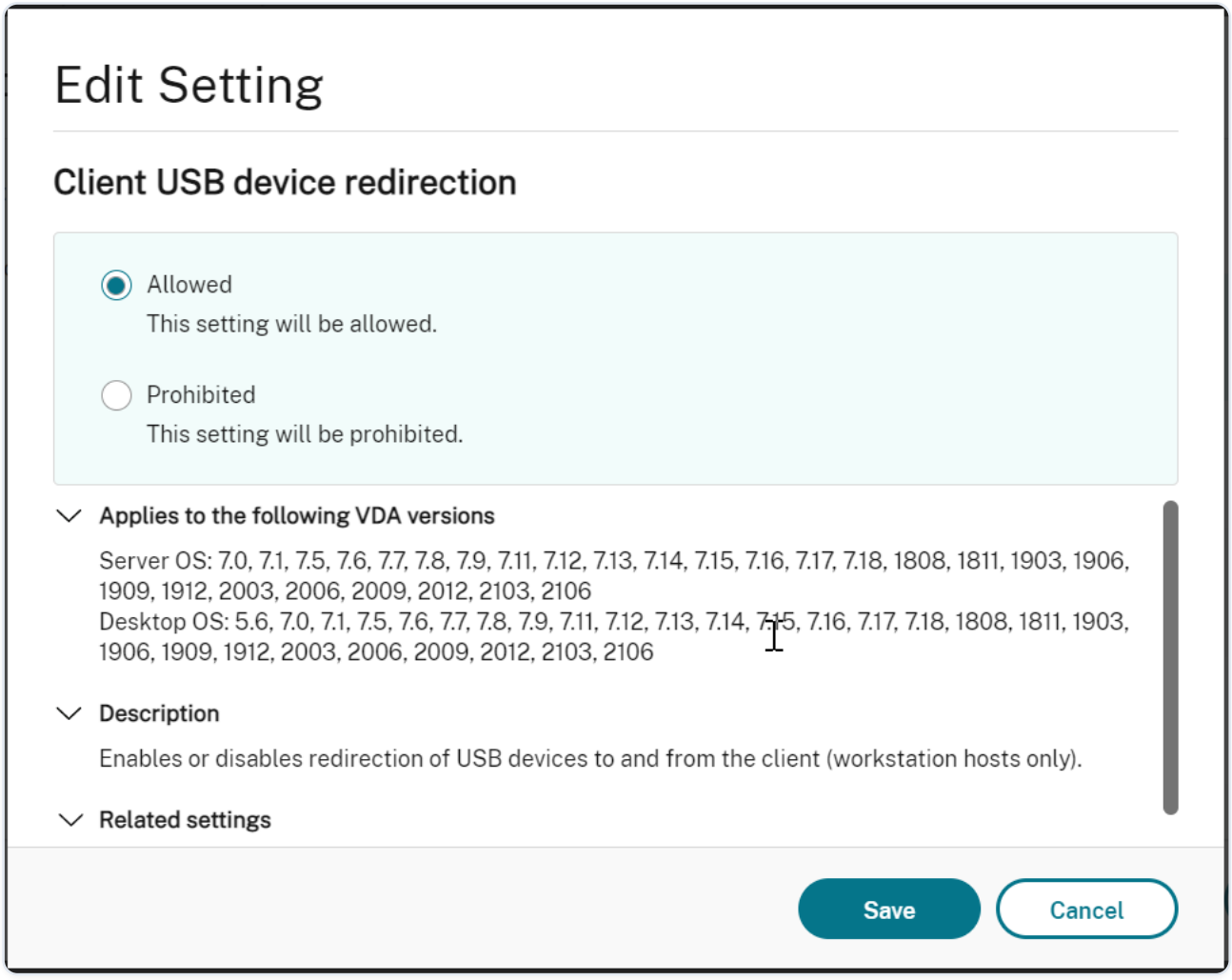
- クライアントUSBデバイスリダイレクト規則

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リダイレクト規則を設定する

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）されるデバイスを指定することにより、リダイレクト規則を設定します。

クライアントUSBデバイスリダイレクト規則ダイアログで：

1. リダイレクト規則を追加するには、**新規**をクリックするか、既存の規則を確認するには**編集**をクリックします。
2. 規則を作成（または編集）した後、**OK**をクリックします。

クライアントUSBリダイレクト規則の設定(/en-us/linux-virtual-delivery-agent/2203-ltsr/media/client-usb-redirection-rule.png)

VHCIカーネルモジュールを構築する

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.x、ロッキー・リナックス 8	4.18.0-240
RHEL 7.9、CentOS 7.9	3.10.0-1160
RHEL 7.8、CentOS 7.8	3.10.0-1160
SUSE 12.5	4.12.14
ウブントゥ 20.04	5.4.0-81
ウブントゥ 18.04	4.15.0-154
デビアン 10	4.19.0-17

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、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バージョンと一致しない場合は、次の操作を行います。

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)**」セクションのファイルを選択します。
2. **citrix-linux-vda-sources.zip**ファイルを展開します。**linux-vda-sources/vhci-hcd-1.15.zip**に移動し、`unzip vhci-hcd-1.15.zip`コマンドを使用してVHCIソースファイルを展開します。
3. Linux VDAパッケージがインストールされていることを確認してから、次のいずれかのコマンドを実行します。

   - `sudo bash ctxusbcfg.sh dkms`

     このコマンドを使用すると、Dynamic Kernel Module Support (DKMS) プログラムを使用してVHCIカーネルモジュールを管理できます。DKMSはSUSEでは利用できません。

     > **注:**
     >
     > `sudo bash ctxusbcfg.sh dkms`コマンドは、VDAに`kernel-devel`および`DKMS`プログラムをインストールします。RHELおよびCentOSにプログラムをインストールする場合、このコマンドはVDAにExtra Packages for Enterprise Linux (EPEL) リポジトリをインストールして有効にします。
     >
     > メジャーなカーネルアップグレード（例: バージョン4.x.yからバージョン5.x.yへ）を実行すると、DKMSがVHCIカーネルモジュール（`usb-vhci-hcd.ko`および`usb-vhci-iocif.ko`）のビルドに失敗する場合があります。DKMSが失敗した場合は、`sudo bash ctxusbcfg.sh dkms`を再度実行してください。

   - `sudo bash ctxusbcfg.sh build`

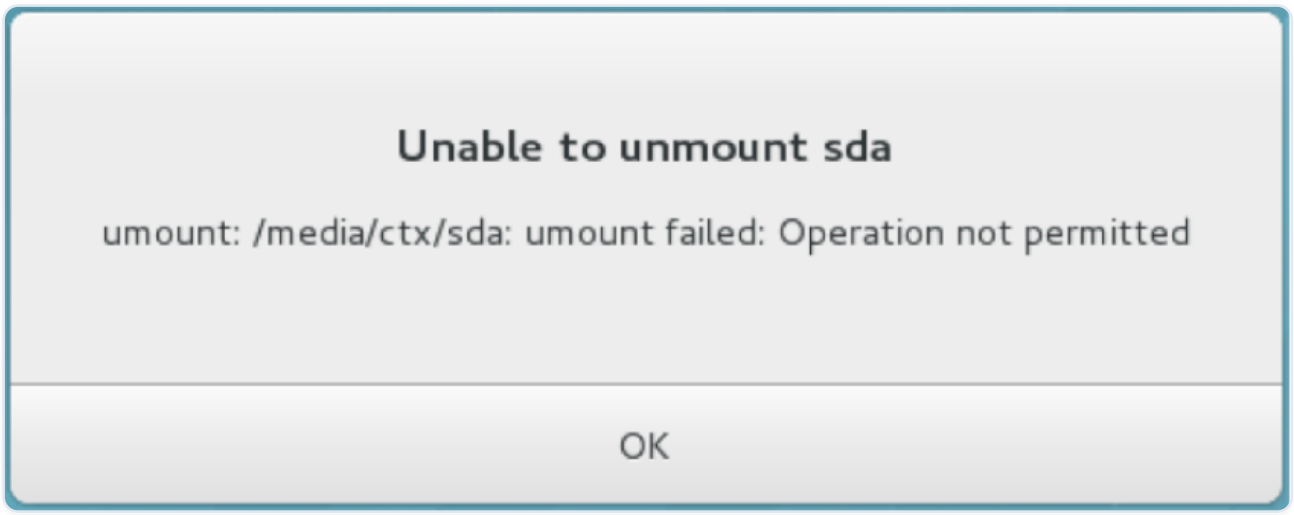
このコマンドは、DKMSオプションなしでVHCIカーネルモジュールをビルドしてインストールします。

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

Linux VDAは、Citrix Workspaceアプリからリダイレクトされたすべてのユーザーディスクを管理者権限で管理し、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します。その結果、デバイスをアンマウントするには管理者権限が必要です。



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

Citrix Workspaceアプリのツールバーを使用してUSBディスクのリダイレクトをすぐに停止すると、ディスク上で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、データがディスク自体に書き込まれないために発生します。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(/en-us/linux-virtual-delivery-agent/2203-ltsr/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場合があります。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。

- USBリダイレクトを許可するようにポリシーが構成されている
- 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/2203-ltsr/media/client-usb-redirection-troubleshoot.png)

> **注:**
>
> Linux版Citrix Workspaceアプリでは、**デバイス**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、*ポリシーによって制限されています*と表示され、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**を確認して、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るが、セッションで使用できない

通常、リダイレクトできるのはサポートされているUSBデバイスのみです。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デバイスごとに、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。

仮想チャネルSDK (実験的)

Linux VDA 用の仮想チャネルソフトウェア開発キット (SDK) を使用すると、VDA 上で実行するサーバー側アプリケーションを作成できます。詳細については、Citrix Virtual Channel SDK for the Linux VDA ドキュメントを参照してください。

Citrix Virtual Channel SDK for the Linux VDA は、Citrix Virtual Apps and Desktops ダウンロードページ からダウンロードできます。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 Linux VDA のダウンロードを選択します。

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 2203のドキュメントに対する重要な変更点を示します。これらの更新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
04 Jun 2024	Linux Virtual Delivery Agent 2203 LTSR CU5のリリース	「新機能」を参照してください。
16 Nov 2023	Linux Virtual Delivery Agent 2203 LTSR CU4のリリース	「新機能」を参照してください。
01 Jun 2023	Linux Virtual Delivery Agent 2203 LTSR CU3のリリース	「新機能」を参照してください。
08 Dec 2022	Linux Virtual Delivery Agent 2203 LTSR CU2のリリース	「新機能」を参照してください。
03 Aug 2022	Linux Virtual Delivery Agent 2203 LTSR CU1のリリース	「新機能」を参照してください。
23 Mar 2022	Linux Virtual Delivery Agent 2203 LTSRの初期リリース	「新機能」を参照してください。