リナックス バーチャル デリバリー エージェント 2607 LTSR

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 (VM) にVDAソフトウェアをインストールし、Delivery Controllerを構成し、その後、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

2607 LTSR の新機能

このリリースには、以下の新機能と機能強化が含まれています。

Debian 13.6 および 12.15、RHEL および Rocky Linux 10.2、RHEL および Rocky Linux 9.8、Ubuntu 26.04 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。

- デビアン 13.6
- デビアン 12.15
- RHEL と ロッキー リナックス 10.2
- RHEL と ロッキー リナックス 9.8
- ウブントゥ 26.04

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。

> **注:**
>
> RHEL および Rocky Linux 10 の場合、LTSR 2607 GA はバージョン 10.2 のみをサポートします。

CTXGFX グラフィックプロセス自動復旧

グラフィックサブシステムで障害が発生した場合、Linux VDA は Thinwire レンダリングとユーザー入力 (キーボードとマウス) を自動的に復元し、セッションを中断することなく続行できるようにします。

セッション間クリップボード共有 (一般提供)

以前はテクニカルプレビューとして提供されていたセッション間クリップボードデータ共有が、一般提供されるようになりました。 以前は、この機能はレジストリキーを介して有効化および構

成されていました。このリリースからは、この機能はクリップボード共有スコープとクリップボード書き込み許可形式のグループポリシーによって制御され、管理者は許可されるクリップボード形式を指定するための集中管理されたポリシー駆動型の方法を利用できます。

移動されたUnixドメインソケットファイル

以前は `$HOME` または `/tmp` に作成されていたUnixドメインソケットファイルは、`/run/user/<uid>` (systemdランタイムディレクトリ) に配置されるようになり、`/var/xdl/user/<uid>` または `/var/xdl/user` への自動フォールバックが可能です。読み取り専用またはNFSマウントされたホームディレクトリでのセッションが失敗することはなくなりました。ソケットが `/run/user/<uid>` にある場合、systemdはセッション終了時に自動的にクリーンアップします。フォールバックディレクトリにあるソケットは自動的にクリーンアップされません。

OSロック画面の制御

Linux VDAは、レジストリキーを介したOSレベルのデスクトップロック画面の管理者制御をサポートするようになりました。 OSロック画面はデフォルトで無効になっています。OSロック画面を有効にするには：

```
#/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Login" -t
"REG_DWORD" -v "EnableOsLockScreen" -d "0x00000001" --force
```

詳しくは、「OSロック画面の制御」を参照してください。

以前のリリースでの新機能

1912 **LTSR** から2603 **CR** までに出荷されたリリース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。

修正された問題

Linux Virtual Delivery Agent 2607 には、以下の修正された問題が含まれています。

- スマートカード認証が有効になっている場合、アプリケーションの起動がハングしたり、タイムアウトしたりすることがありました。この問題は、セッションがアクティブなときに `ctxscardsd` サービスが再起動し、サービスがセッション作成時にのみ記録されたセッション情報を失うことが原因で発生しました。
- Microsoft Azure でホストされているマシンでは、マシンカタログを更新したり、新しいカタログを作成したりした後、セッションの起動が断続的に失敗することがありました。これは、

Xorg が準備完了になるまでに、固定された 30 秒の起動期間よりも長い時間を要することがあったためです。Xorg の起動待機時間は、`WaitXMaxTimeOut` Thinwire 設定で構成できるようになり、デフォルト値は 30 秒です。

- 再接続の試行が、たとえばログオンウィンドウを閉じることによって中断された後、スマートカード認証を使用したセッションは回復できませんでした。その後の再接続の試行では、グレースクリーンが表示されました。セッションは、VDA プロセスを手動で終了することによってのみ解放できました。
- クライアントのネットワーク接続が切断され、その後自動クライアント再接続 (ACR) を使用して再接続された場合、Linux VDA セッションがクラッシュしたり、予期せずログオフしたりすることがありました。
- 韓国語サーバー側 IME を使用している場合、ハングルキーとハングル/漢字キーが機能しませんでした。
- セッション認証に FAS が使用されている場合、セッション内のスマートカード操作が非常に遅くなったり、応答しなくなったりすることがありました。カードポーリングが繰り返し停止したため、クライアント証明書認証とページロードの完了に数分かかることがありました。
- セッションが再接続に失敗し、グラフィックエラーの後に WINSTATION_BUSY エラーが表示され、セッションを終了する必要がありました。これは、PostgreSQL 構成データベースが応答しなくなり、`ctxgfx` プロセスが無期限にブロックされたときに発生しました。

既知の問題

- BCR は最新の Citrix Browser Redirection Extension では動作しません。

Debian 12.11 は Nvidia ドライバー 570 では動作しません。Nvidia の公式ドキュメントによると、リリースされた 570 ドライバーは Debian 12.8 以下の Debian バージョンをサポートしています。データセンターのドキュメントを参照してください。

- HDX™ 3D pro が有効な状態で、複数のモニターで複数のビデオを再生すると、マルチモニターセッションがハングアップ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 2511 (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

この記事のお知らせは、段階的に廃止されるプラットフォーム、Citrix®製品、および機能について事前に通知し、お客様がタイムリーなビジネス上の意思決定を行えるようにするものです。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性を含ま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。

注:

2503リリース以降、最新版より前のDebianマイナーリリースに対するサポートは、通知なしに自動的に非推奨となります。

項目	非推奨が発表された時期	削除時期
Ubuntu 22.04のサポート	2607	2611
RHEL 9.4、9.7のサポート	2603	2607
Rocky Linux 9.4、9.7のサポート	2603	2607
SUSE 15.6のサポート	2507	2511
デビアン 11.x	2503	2511
RHEL 8.8、9.2、9.5のサポート	2503	2507
Rocky Linux 8.8、9.2、9.5のサポート	2503	2507

項目	非推奨が発表された時期	削除時期
Amazon Linux 2のサポート	2407	2503
RHEL 9.3、RHEL 8.9のサポート	2407	2411
Rocky Linux 9.3、Rocky Linux 8.9のサポート	2407	2411
Ubuntu 20.04 のサポート	2407	2503
SUSE 15.5 のサポート	2407	2411
Debian 11.7/11.3 のサポート	2407	2407
RHEL 9.0、RHEL 8.6 のサポート	2402	2407
Rocky Linux 9.0、Rocky Linux 8.6 のサポート	2402	2407
RHEL 7.9、CentOS 7.9 のサポート	2402	2407
SUSE 15.4 のサポート	2308	2311
Rocky Linux 9.1、Rocky Linux 8.7 のサポート	2305	2308
RHEL 9.1、RHEL 8.7 のサポート	2305	2308
RHEL 8.4 のサポート	2303	2308
Ubuntu 18.04 のサポート	2212	2305
SUSE 15.3 のサポート	2210	2301
Debian 10.9 のサポート	2206	2210
SUSE 15.2 のサポート	2206	2209

項目	非推奨が発表された時期	削除時期
RHEL 8.2 のサポート	2206	2209
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
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

新機能の履歴

この記事では、1912 LTSR以降2603 CRまでのリリースに含まれる新機能について説明します。

2603 の新機能

必要な.NETの新しいバージョン

このリリース以降、Linux VDAをインストールする前に、サポートされているすべてのLinuxディストリビューションに.NET Runtime 10をインストールする必要があります。

必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。

新しいハイパーバイザーとクラウドプラットフォームのサポート

Linux VDAの動作は、OpenShift Virtualization、Amazon WorkSpaces Core、およびNutanix Prism Centralで検証されており、最新のインフラストラクチャ展開のサポートを拡大しています。

MCSベースの展開は、Amazon WorkSpaces CoreとNutanix Prism Centralでサポートされています。OpenShift VirtualizationはLinux VDAの動作をサポートしていますが、MCS展開はサポートしていません。

軽量データベースオプションのサポート

リリース2603以降、Linux VDAは、PostgreSQLやSQLiteのようなデータベースを使用する代わりに、構成用の組み込みレジストリ（軽量データベース）サービスをサポートしています。組み込みレジストリサービスを使用するには、インストール後に/etc/xdl/db.confで「DbType」を「ctxregd」に設定します。デフォルトでは、新しくインストールされたLinux VDAの場合、DbTypeはctxregdに設定されます。アップグレードされたLinux VDAの場合、db.confは古いものを使用し、DbTypeは上書きされません。

クリップボードファイルサイズ制御のサポート

クリップボードのセキュリティを強化するために、クリップボードサイズ制御ポリシーにファイルサイズ強制機能を追加しました。

「クリップボードクライアントからセッションへの転送サイズを制限」または**「クリップボードセッションからクライアントへの転送サイズを制限」**ポリシーが有効になっている場合、以下に設定されたファイルサイズ（KB単位）の制限が適用されます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard"
-t "REG_DWORD" -v "FileCopySizeLimitC2H" -d "0x00032000" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard"
-t "REG_DWORD" -v "FileCopySizeLimitH2C" -d "0x00032000" --force
```

Linux VDA GUIベースのセットアップツールが利用可能

プロビジョニングワークフローを簡素化し、手動設定エラーを減らすために、新しいGUIベースのLinux VDAセットアップツールを追加しました。

MCS準備済みイメージワークフローに対応

Linux VDAは、MCSの準備済みイメージワークフローをサポートするようになり、カタログ間で事前に生成されたイメージを再利用できるようになりました。これにより、ストレージコストが削減され、イメージ管理と更新が効率化されます。

MCSワークフローにおけるAdcliベースのマシンパスワード管理

MCSでプロビジョニングされたLinux VDAは、マシンアカウントのパスワードローテーションにSambaに代わって*adcli*を使用するようになり、セキュリティコンプライアンスを強化し、管理オーバーヘッドを最小限に抑えます。

オーディオ音量同期のサポート

リリース2603以降、Linux VDAはVDAとオーディオデバイス間のオーディオ音量同期をサポートするようになりました。VDAのオーディオ音量スライダーを使用して音量を調整できるようになり、デバイスでも同じ音量になります。これは逆の場合も同様です。この機能はデフォルトで有効になっています。

2511 の新機能

Debian 12.12、Debian 13.1、RHEL9.7、Rocky Linux 9.7、SLES 15.7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。

- デビアン 12.12
- Debian 13.1 (注: Debian 13のSambaの制限により、MCSプロビジョニングは現在利用できません。GPUドライバーの制限により、HDX 3D Proは現在利用できません)

- RHEL 9.7
- ロッキー リナックス 9.7
- スーゼ リナックス エンタープライズ サーバー 15.7

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。

RHEL & Rocky Linux 10のサポート（プレビュー）

RHELおよびRocky Linux 10のリリースにより、従来のX Window System (X.Org) からWaylandへのデフォルトディスプレイサーバープロトコルの大きな移行がありました。

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

- RHEL 10
- ロッキー Linux 10

「RHELおよびRocky Linux 10プレビューの制限事項」を参照してください(/ja-jp/linux-virtual-delivery-agent/2511/system-requirements.html#limitations-of-rhel-and-rocky-linux-10-preview)

オプションのLinux VDAサービスの無効化をサポート

この機能により、オプションのLinux VDAサービスを`/etc/xdl/disablelist.conf`を介して無効にできます。これにより、ポリシーおよびローカルレジストリ設定よりも高い優先順位が与えられます。

サービスを無効にするには、Linux VDAパッケージのインストール後、構成ファイルからコメントアウトを解除します。

Azure仮想マシンでのUSBリダイレクトのサポート

このリリースでは、Canonical Ubuntu 22.04および24.04を実行しているAzure仮想マシン向けのUSBリダイレクトのサポートを発表できることを嬉しく思います。これらのプラットフォームには特定のUSBカーネルドライバーがデフォルトで含まれていないため、特定のUSBデバイスに必要なカーネルドライバーモジュールをコンパイルしてロードできる柔軟なソリューションを提供します。

詳細については、「USBデバイスリダイレクト」を参照してください(/ja-jp/linux-virtual-delivery-agent/2511/configure/general-content-redirection/usb-redirection.html#support-usb-redirection-for-ubuntu-on-azure)

HDX画面共有の改善

このリリースから、Linux VDAはHDX画面共有に2つの主要な機能強化を導入します。

- 共有を容易にし、セッション開始を迅速化するためのより短い招待コード
- セキュリティとコンプライアンスの向上を目的として、画面共有アクティビティを追跡および確認するためのシステムログ内の監査ログ情報

詳細については、「画面共有の改善」を参照してください(/ja-jp/linux-virtual-delivery-agent/2511/configure/graphics/hdx-screen-sharing.html#screen-sharing-improvement)

Director 用 HDX 画面共有

HDX画面共有は、Citrix Virtual Apps and Desktopsオンプレミスインストール用のCitrix DirectorおよびCitrix Cloud展開用のCitrix Monitorを使用したシャドウイングで利用できます。HDX画面共有は、シャドウイングの新しい方法であり、現在のシャドウイングに代わるネイティブで低遅延の代替手段を提供します。これには、CR 2511以降のバージョンのCitrix DirectorとLinux VDAの両方が必要です。詳細については、「HDX画面共有の使用」を参照してください(/ja-jp/citrix-virtual-apps-desktops/director/troubleshoot-deployments/user-issues/shadow-users#using-hdx-screen-share)

クロスセッションクリップボード交換サポート (プレビュー)

この機能は、共通のクライアントホストを介してあるセッションから別のセッションにコンテンツをコピーする機能を含みますが、そのコンテンツをクライアントホストマシンに実際に保存することはありません。この機能のテクニカルプレビューリリースでは、CrossSessionClipboard を介してトンネルされるフォーマットのリストを VDA RegKey から取得します。この値は、CSC を介してトンネルされるフォーマットのリストです。

例:

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\wfshell\Virtual Clipboard\CrossSessionClipboard" -t "REG_MULTI_SZ" -v "CrossSessionFormats" -d "CF_TEXT\0CF_BITMAP\0CF_DIB\0CF_UNICODETEXT\0CFX_FILE\0\0" –force
```

強化されたTCP輻輳制御

このリリースから、高遅延ネットワーク下でのセッション内ビデオおよびオーディオパフォーマンスを向上させるために、強化されたTCP輻輳制御が提供されます。これはデフォルトでオフになっており、TCP輻輳制御方法はシステムデフォルトのままです。

```
opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd\HdxPerformance"
-t "REG_DWORD" -v "EnhancedTcpCongestionEnabled" -d "0x00000001" –force
```

オーディオリダイレクトのためのTCPトランスポートによるユーザーエクスペリエンスの向上

テクニカルプレビュー版の2511から、信頼性の高いトランスポート（つまりTCPおよびEDT-Reliable）を介した接続でもオーディオ品質エンハンサーが利用可能になりました。ネットワーク状態が悪い場合でも、オーディオ再生と録音の両方で、遅延が少なく、よりスムーズなエクスペリエンスが得られます。

信頼性の高いトランスポート用オーディオ品質エンハンサー (プレビュー)

テクニカルプレビュー版の2511から、信頼性の高いトランスポート（つまりTCPおよびEDT-Reliable）を介した接続でもオーディオ品質エンハンサーが利用可能になりました。ネットワーク状態が悪い場合でも、オーディオ再生と録音の両方で、遅延が少なく、よりスムーズなエクスペリエンスが得られます。

コネクションレスワークロードに対するService Continuityのサポート

Service Continuityにより、エンドユーザーはビジネスに不可欠なワークロードへのアクセスをサービス中断中も維持できます。この新機能により、Service ContinuityはLinux専用VDI コネクタレスのリソースロケーションにあるワークロードをサポートするようになりました。これには、AD参加済みおよび非ドメイン参加済みワークロードが含まれます。詳細については、Service Continuityを参照してください。

CGS FQDN IPアドレスのキャッシュをサポート

Linux VDA Rendezvous V2は、最初のDNSルックアップが成功した後、CGS FQDN IPアドレスのキャッシュをサポートするようになりました。DNSが一時的に利用できなくなった場合、接続はキャッシュされたIPにフォールバックできます。この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、`/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Ica\EdgeNodeCache" -t "REG_DWORD" -v "EnableDNSCache" -d "0x00000001" –force`を実行します。

FAS起動チケットv2を介したFAS情報送信のサポート

この機能により、FAS起動チケットv2を介したFASサーバーのFQDNとポートの送信が可能になります。Linux VDAは、この情報をチケットから直接デコードするため、VDAにレジストリエントリは不要です。

PVS IPv6のサポート

このリリースでは、純粋なIPv4、純粋なIPv6、およびIPv4とIPv6ネットワークが重複するデュアルスタック展開がサポートされます。詳細については、ドキュメント「IPv4/IPv6のサポート」を参照してください。このリリースでは、VMware用のNVMeディスクがサポートされます。

AOTログの収集とアップロード

このリリースから、Linux VDAはAOTログを集中ログサーバーに収集およびアップロードすることをサポートし、Citrixコンポーネント全体のトラブルシューティングを改善します。

詳細については、「AOTログの収集」を参照してください。

2507 LTSRの新機能

Debian 12.11、RHEL、およびRocky Linux 9.6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。

- デビアン 12.11
- RHEL 9.6
- ロッキー Linux 9.6

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。

SmartcardおよびFAS展開機能の更新

SmartcardおよびFASの展開機能が更新され、Rocky8/9、Ubuntu 24.04、Debian 12/11、SLES 15.6 Linuxプラットフォームに固有の追加の注意事項が含まれています。詳細については、「Smartcardでサポートされているディストリビューション」および「FASで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。

SSSD + FAS + MCS展開機能の更新

SSSD + FAS + MCS展開機能が更新され、RHELおよびRocky Linuxプラットフォームに特化した追加の注意事項が含まれています。詳細については、「FASが有効なLinux VDAをMCSを使用して作成する」(/ja-jp/linux-virtual-delivery-agent/2507-ltsr/installation-overview/create-linux-vdas-with-fas-enabled-using-mcs.html)を参照してください。

セッション再接続時の異なるログオンタイプの許可

以前は、セッションの再接続には元のセッションと同じログオンタイプが必要でした。この機能強化により、ユーザーは異なるログオンタイプを使用して再接続できます。この機能は、ユーザー名/パスワードおよびFAS認証方法にのみ適用されます。この機能を有効にするには、ローカルレジストリキーを設定します。

```
HKLM\System\CurrentControlSet\Control\Citrix\AccessControl\Login -t
REG_DWORD -v allowReconnectCredChange -d 0x00000001
```

ログオンバナー表示タイムアウトの構成

ログオンバナーの表示タイムアウト期間が構成可能になりました。以前は、ログオンバナーは60秒間操作がないと自動的に進行していました。この機能により、ログオンバナーのタイムアウト期間をカスタマイズできるようになりました。「セッションログオン画面のカスタム背景とバナーメッセージ」(/ja-jp/linux-virtual-delivery-agent/2507-ltsr/configure/session/custom-backgrounds-and-banner-messages-on-session-logon-screens.html)を参照してください。

ログインUIでのキーボードレイアウト選択のサポート

ログインインターフェイスにキーボードレイアウトセレクターを追加し、異なる入力言語を切り替えられるようにします。この機能により、適切なキーボードレイアウトを設定することで、ユーザー名とパスワードを正しく入力できます。

最適化されたクライアントドライブマッピング

以下のCDM機能が最適化されています。

- ファイル転送効率の向上と大容量ファイル（4GB超）のアップロードサポート。
- クラッシュした`ctxcdm`進行状況の影響を最小限に抑えます。CDMリンクフォルダーは`$Home/ctxmnt/drives`に変更されます。「クライアントドライブマッピング」(/ja-jp/linux-virtual-delivery-agent/2507-ltsr/configure/general-content-redirection/client-drive-mapping.html)を参照してください。

LDAPSクエリでのCNAMEレコードのサポート

以前は、Linux VDAはLDAPSクエリワークフローでCNAMEレコードをサポートしていませんでした。2507以降、LDAPSクエリでのCNAMEレコードはデフォルトでサポートされます。

マルチドメイン構成のサポート

Easyinstall 用に新しい環境変数 `CTX_EASYINSTALL_TRUSTED_DOMAINS` が、MCS 用に `TRUSTED_DOMAINS` が利用可能になりました。信頼できるドメインのリストを指定できます。

LDAP サーバーリストが指定されていない場合、スクリプトは自動的に `krb5.conf` を更新し、それらの信頼できるドメイン内の LDAP サーバーを自動検出します。

Easy Install GUI の機能強化

Easy Install GUI は、複数のオプションから選択できる新しいインストールタイプ選択ボタンを含む、再構築されたコンポーネント設計を特徴としています。データベース設定専用のページで、データベース設定を簡単に構成できます。新しい「ドメイン非参加」インストールタイプと迅速な概念実証 (POC) インストールタイプは、さまざまな環境に対してより高い柔軟性を提供します。構成を保存およびロードし、オプション設定をカスタマイズし、最小限の構成で完全インストールと POC インストールの両方のセットアッププロセスを効率化できます。詳細については、「Easy Install を使用してドメイン参加済み VDA を作成する」を参照してください。

Director でより多くのセッションメトリックを表示するサポート

セッションパフォーマンスの問題を診断する ために、Director でより多くのセッションメトリックを表示するサポート。

オーディオ診断コマンドラインツール

オーディオ診断コマンドラインツールは、バージョン 2507 で強化され、追加の診断情報を提供するようになりました。詳細については、「オーディオ診断コマンドラインツール」を参照してください。

Adaptive Audio用オーディオ品質エンハンサー

バージョン 2507 以降、適応型オーディオでは Audio Quality Enhancer がデフォルトで有効になっています。この機能は、ネットワーク状況の変化に適応することで、短時間のネットワーク中断中もクリアなオーディオを維持し、再生と録音の両方で一貫したオーディオパフォーマンスを保証します。詳細については、「Audio Quality Enhancer」を参照してください。

2503 の新機能

RHEL 9.5、Rocky Linux 9.5、および Debian 12.9 のサポート

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

- RHEL 9.5
- ロッキー・リナックス 9.5
- デビアン 12.9

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。

ドメイン非参加の Linux VDA が、MCS プロビジョニングを使用してオンプレミス CVAD 展開に拡張されました

以前は、ドメインに参加していないLinux VDAはCitrix DaaS展開でのみ利用可能でした。現在、MCSプロビジョニングテクノロジーにより、オンプレミスのCVAD展開にも拡張できます。この機能は、CVAD 2411およびLinux VDA 2411以降で有効です。詳細については、「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」を参照してください。

MATEデスクトップサポートの拡張

Linux VDAは、サポートされているすべてのRed Hat Enterprise Linux (RHEL) 9.xおよびRocky Linux 9.xディストリビューションでMATEデスクトップ環境をサポートするようになりました。詳細については、「システム要件」を参照してください。

ルートレスXorgはデフォルトで無効になりました

Red Hatは、Xorgサーバーに影響を与えるセキュリティ脆弱性CVE-2024-31083に対処しました。詳細については、「ルートレスXorg」を参照してください。

Secure HDX™が一般提供開始

Citrix Workspace™アプリ（クライアント）とVDA（セッションホスト）の間でICA®セッションをエンドツーエンドで暗号化できます。

エンドツーエンド暗号化（E2EE）機能により、Citrix Gatewayを含む中間ネットワーク要素がICAトラフィックを復号化することはできません。これにより、環境のセキュリティ体制が向上し、構成と管理が容易になります。詳細については、「Secure HDX」を参照してください。

HDX ダイレクト

Citrixが提供するリソースにアクセスする場合、直接通信が可能な場合、HDX Directは内部および外部のクライアントデバイスの両方がセッションホストとの安全な直接接続を確立できるようにします。詳細については、「Linux用HDX Direct」を参照してください。

ドメインに参加しているVDAとDelivery Controller間の通信に対するWebSocketサポート

この機能により、ドメインに参加しているVDAとDelivery Controller間でWebSocket通信を設定でき、Windows Communication Foundation (WCF) の代替手段を提供します。

詳細については、「VDAとDelivery Controller間のWebSocket通信」を参照してください。

カスタマイズ可能なログイン方法

Linux VDAでは、シングルサインオン (SSO) を使用しない場合のユーザーのログイン方法を制御できるようになりました。表示される認証方法を選択することで、ログインエクスペリエンスをカスタマイズできます。以前は、非SSOログイン画面には常にパスワード認証とスマートカード認証の両方がドロップダウンリストに表示されていました。現在、Linux VDAを次のように構成できます。

- パスワード認証のみ
- スマートカード認証のみ
- パスワード認証とスマートカード認証の組み合わせ。どちらかのオプションがデフォルトで表示されます。

VDAで以下のレジストリ設定を調整することで、ログイン方法を簡単にカスタマイズできます。

`System\CurrentControlSet\Control\Citrix\login\NSSOLogonType`

詳しくは、「ログイン画面のカスタマイズ」を参照してください。

シームレスアプリの非SSOログインの改善

SSOが無効な場合のシームレスアプリのログインエクスペリエンスが大幅に改善されました。以前は、アプリの起動時にログオンに時間がかかり、サイズが不適切な全画面表示のログインウィンドウが表示されていました。現在は、画面中央に適切なサイズのログインボックスがすぐに表示され、利用可能なウィンドウスペースに合わせて動的に調整されます。

ユーザー認証の強化

Linux VDAは、複数のユーザープリンシパル名（UPN）を持つユーザーをサポートするようになりました。ユーザーSIDでLDAPを照会することにより、システムはアカウントに関連付けられた複数のUPNを持つユーザーを正しく識別し、認証できます。この機能強化により、Linux VDAユーザーはより柔軟で信頼性が高く、使いやすいログインエクスペリエンスを得られます。

FAS SSO失敗時のセカンダリ認証

Linux VDAは、Federated Authentication Service (FAS) SSOが失敗した場合に特化したセカンダリ認証方法を提供することで、ログインの回復力を強化しました。以前は、FAS SSOで問題が発生すると、ユーザーは認証できず、セッションの起動に失敗していました。この機能により、FAS SSOで問題が発生した場合、ユーザーはパスワード認証のために手動で資格情報を入力するよう求められます。この機能を有効にするには、次のコマンドを実行します。

```
create -k
"HKLM\System\CurrentControlSet\Control\Citrix\AccessControl\Login\Global"
-t "REG_DWORD" -v "SecondaryAuthEnabled" -d "0x00000001" --force
```

詳しくは、「FAS SSO失敗時のセカンダリ認証を有効にする」を参照してください。

スマートカード取り外し時のデスクトップロック

このリリースでは、サインインしているユーザーがアクティブなセッション中にリーダーからスマートカードを取り外すと、デスクトップが自動的にロックされるセキュリティ強化が導入されました。この強化は、VDAで以下のレジストリ設定を行うことで実現できます。

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LocalPolicies\SecurityOptions`

詳しくは、「スマートカード取り外し設定」を参照してください。

JUMBO_ICONSのサポート

Linux VDA は JUMBO_ICONS をサポートするようになりました。この機能強化により、シームレスな公開アプリセッションが、Mac クライアント向け Citrix Workspace アプリで適切な高解像度アイコンとともに正しく表示されるようになります。

オーディオの損失許容モードがデフォルトで有効になりました

詳細については、オーディオ機能を参照してください。

グラフィックのロス耐性モードのサポート

損失許容モードは、パケット損失が検出された場合でもセッションがインタラクティブな状態を維持するように徹底的に再設計されました。 詳細については、「グラフィックの損失許容モード」を参照してください。

インテリジェントなビルド・トゥ・ロスレス機能

インテリジェントなビルド・トゥ・ロスレスは、既存のビルド・トゥ・ロスレス機能の強化版であり、よりスマートで効率的なエクスペリエンスを提供するように設計されています。インテリジェントなビルド・トゥ・ロスレスを使用すると、管理者はビルド・トゥ・ロスレスを手動で構成する必要がなくなり、必要に応じてセッション内のモニターに動的に自動適用されます。必要がなくなると、Thinwire はシームレスにこれを無効にし、最適なパフォーマンスのために通常のエンコードに戻します。この機能は、GPU (HDX 3D Pro) を使用している場合にのみ適用されます。この機能を使用するには、**ビデオコーデックを圧縮に使用**設定が**優先時に使用** (デフォルト) であり、**視覚品質**が**常にロスレス**であってはなりません。

クリップボード同期の強化

このリリースでは、クライアントと VDA セッション間のクリップボード同期を改善するためのクリップボードバッファリング機能が導入されています。以前は、VDA セッション内でテキストを選択すると、クライアントでコピーされたクリップボードデータが上書きされ、ユーザーがそのデータを VDA に貼り付けることができませんでした。この問題は、PRIMARY 選択更新モードが「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」または「クライアントの選択の変更はホストで更新されない」のいずれかに設定されている場合に発生していました。この問題は解決されました。

クリップボードバッファリング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、VDA セッション内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\VirtualChannels\Clipboard\Cl
ipboardSelection" -t "REG_DWORD" -v "BufferData" -d "0x00000001" --force
```

注:

この機能は、テキストデータと BMP データにのみ適用されます。ファイルコピー機能には影響しません。

マウスとトラックパッドのサポート強化

さまざまなクライアントデバイスで、よりスムーズなトラックパッドスクロールとマウスサイドボタンとの互換性を向上させるために、生のマウスイベントの処理を改善しました。

VDA アップグレードの簡素化 (プレビュー)

以前は、VDA のアップグレードには完全な手動介入が必要でした。バージョン 2503 では、VDA アップグレードエージェントを導入することで、DaaS 展開における VDA アップグレードを簡素化します。バージョン 2503 以降のアップグレードは、共有またはローカルのファイルパスから直接実行できるようになります。詳細については、「VDA アップグレード」を参照してください。

2411の新機能

セッション安定性の向上

ユーザーセッションプロセス (`ctxgfx`) は、`ctxhdx` サービスから分離されました。以前は、`ctxhdx` サービスが再起動されると、アクティブまたは切断されたユーザーセッションが終了する可能性がありました。この更新により、`ctxhdx` サービスが再起動されてもユーザーセッションはその状態を維持できるようになり、セッション全体の安定性が向上します。

この機能はデフォルトで有効になっています。無効になっている場合は、次のコマンドを使用して再度有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\HDXSessionRecovery" -t "REG_DWORD" -v
"EnableSessionRecovery" -d "0x1" --force
```

Ubuntu 24.04、Debian 12.7、Debian 11.11、およびSUSE 15.6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。

- ウブントゥ 24.04
- デビアン 12.7
- デビアン 11.11
- SUSE 15.6

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は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。

PostgreSQLバージョンのサポート強化

Linux VDAは、Linuxディストリビューションによって提供されるバージョンに関係なく、カスタムPostgreSQLバージョンの使用をサポートするようになりました。Linuxディストリビューションによって提供されるバージョンと少なくとも同程度の最新のPostgreSQLバージョンを使用することをお勧めします。カスタムバージョンのPostgreSQLを効果的に使用するには、新しいバージョンに合わせて`/etc/xdl/db.conf`が適切に構成されていることを確認してください。詳細につい

ては、インストール記事のデータベース指定手順（例: 「手順7: 使用するデータベースを指定する」および「手順1g: 使用するデータベースをインストールして指定する」）を参照してください。

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に比べて、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。

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** に移動されます。

強化されたファイルコピー＆ペースト機能

このリリースには、ファイル名における特殊文字のサポートを目的とした、ファイルコピー＆ペースト機能の強化が含まれています。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。

クロスプラットフォームキーボード機能の改善

このリリースでは、Linux VDAとWindows VDA間でセッションを共有する際のキーボード機能に関する問題が解決されています。詳細については、「HDX画面共有」を参照してください。

Webカメラ機能の改善

このリリースには、Webカメラ機能に対するいくつかの機能強化が含まれています。

- デバイス作成時のWebカメラの解像度とFPSネゴシエーションの改善。
- Webカメラの作成および削除プロセスの最適化。
- セッション内Webカメラサービスの自己修復のサポートを追加。

Ubuntu 24.04でのPipeWireのサポート

Linux VDAは、デフォルトのオーディオサービスであるUbuntu 24.04にPipeWireのサポートを拡張しました。詳細については、「オーディオ機能」を参照してください。

複数のオーディオデバイスのサポートとオーディオのロス許容モードが一般提供開始

複数のオーディオデバイスのサポートがデフォルトで有効になりました。オーディオのロス許容モードは、レジストリではなくポリシー設定を通じて有効にできるようになりました。詳細については、「オーディオ機能」を参照してください。

アダプティブオーディオのオーディオ品質エンハンサー（プレビュー）

バージョン2411以降、アダプティブオーディオ用のオーディオ品質エンハンサーをプレビュー機能として導入しました。この機能強化により、以前のサンプルからオーディオをインテリジェントに再構築することで、短期間のパケット損失や中断を効果的に管理し、品質の著しい低下を防ぎます。また、必要な場合にのみ失われたオーディオパケットを適応的に回復します。オーディオ品質エンハンサーは、パケット損失の持続的な変化に基づいて自動的に有効/無効を切り替え、良好なネットワーク状態と不良なネットワーク状態の両方でオーディオ再生と録音の品質を最適化します。詳細については、「オーディオ機能」を参照してください。

より多くの自動クライアント再接続ポリシーのサポート

Linux VDAは、セッションの自動再接続を制御するために、自動クライアント再接続ポリシーに加えて、以下のポリシーをサポートするようになりました。

- 自動クライアント再接続認証
- 自動クライアント再接続ログ
- 自動クライアント再接続タイムアウト

ポリシーの詳細については、「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:**
>
> 自動クライアント再接続ログが有効になっている場合、Linux VDAは自動クライアント再接続をシステムログに直接書き込みます。この動作は、ログがイベントログに書き込まれるWindowsとは異なります。

> Linux VDAがサポートするポリシーのリストについては、「ポリシーサポートリスト」を参照してください。

2407 の新機能

RHEL 9.4 / 8.10、Rocky Linux 9.4 / 8.10、およびDebian 12.5 / 11.9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 9.4
- ロッキー Linux 9.4
- RHEL 8.10

- ロッキー Linux 8.10
- デビアン 12.5
- デビアン 11.9

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。

新しい.NETランタイムが必要です

.NET Runtimeに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションに.ASP.NET Core Runtimeをインストールする必要があります。Amazon Linux 2にはバージョン6が、その他のディストリビューションにはバージョン8が必要です。

必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールしてください。含まれていない場合は、Microsoftパッケージフィードから.NETをインストールしてください。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

トークンベースの登録がオンプレミス環境に拡張されました（プレビュー）

このリリースでは、トークンベースの登録機能がオンプレミス環境に拡張されました。簡易インストールを使用してドメインに参加していないVDAを作成する際、次のことができるようになりました。

- ドメインに参加していないVDAを、クラウドとオンプレミスの両方のデリバリーコントローラーのマシンカタログに登録する。
- Citrix Web Studio™を使用して、空のマシンカタログを作成し、登録トークンを生成する。

詳細については、「簡易インストールを使用してドメインに参加していないLinux VDAを作成する（プレビュー）」を参照してください。

SSSDがデフォルトのドメイン参加方法として指定されました

SSSDは、Winbindに代わって、ドメイン参加のデフォルトの方法となりました。この変更は、ctxsetup.shまたは簡易インストールスクリプト（ctxinstall.sh）を対話モードで実行する際の構成動作に影響します。詳細については、「簡易インストールを使用してドメインに参加しているVDAを作成する」を参照してください。

アップグレード中に構成ファイルを処理するパッケージマネージャーへの委任

このリリースでは、アップグレード中に構成ファイルを処理するパッケージマネージャー **rpm** または **dpkg** に委任します。さらに、このリリースではRPMlintによって指摘されたいくつかの問題に対処し、RPMlintの要件にさらに準拠するようにRPMパッケージを再構築します。詳細については、「ステップ13：Linux VDAをアップグレードする（オプション）」を参照してください。

スマートカード取り外し設定のサポート

Linux VDAのレジストリを編集することで、サインインしているユーザーのスマートカードがセッション中にスマートカードリーダーから取り外されたときに何が起こるかを決定できるようになりました。詳細については、「スマートカード取り外し設定」を参照してください。

特定のクライアントプリンターがHDX™セッションにリダイレクトされるのを防ぐ機能

Citrixポリシー設定の**プリンタードライバーのマッピングと互換性**を構成することで、特定のクライアントプリンターがHDXセッションにリダイレクトされるのを防ぐことができるようになりました。詳しくは、「プリンタードライバーのマッピングと互換性」を参照してください。

オーディオの損失許容モード（プレビュー）

オーディオがEnlightened Data Transport（EDT）損失許容プロトコルでサポートされるようになりました。この機能により、ユーザーが高遅延およびパケット損失のあるネットワーク経由で接続している場合のリアルタイムストリーミングのユーザーエクスペリエンスが向上します。この機能を有効にすると、Citrix Virtual Apps and DesktopsのAdaptive Transportは、より良いオーディオエクスペリエンスのためにEDT損失許容トランスポートプロトコルを使用します。この機能はデフォルトで無効になっています。詳しくは、オーディオ機能の記事の「オーディオの損失許容モード（プレビュー）」を参照してください。

オーディオパフォーマンス向上のためのパケット損失隠蔽の実装（プレビュー）

2407リリース以降、ジッターバッファメカニズムが改善され、適応型オーディオコーデックにパケット損失隠蔽（PLC）が追加されました。PLCは、失われたデータパケットの再構築に役立ちます。この機能強化により、パケット損失耐性とジッター耐性が向上し、オーディオの損失許容モード（EDT損失あり）でのオーディオパフォーマンスが向上します。

この機能を有効にするには、オーディオの損失許容モード（プレビュー）機能も有効にする必要があります。詳しくは、オーディオ機能の記事の「オーディオパフォーマンス向上のためのパケット損失隠蔽（プレビュー）」を参照してください。

Debian 12.5、RHEL 9.x、およびRocky Linux 9.xでのPipeWireのサポート

Linux VDAは、Debian 12.5、RHEL 9.x、およびRocky Linux 9.xでPipeWireをサポートするようになりました。これらのOSではPipeWireがデフォルトのオーディオサービスです。詳しくは、「オーディオ機能」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で、グラフィックパフォーマンスが向上しています。

ログインUIに表示されるログインプロセスの詳細

セッションユーザーは、Linux VDAセッションを開く際に、ログインUIで以下のログインプロセスの詳細を確認できるようになりました。

- **Citrix®グループポリシー設定の評価中**
- **ユーザーアクセス認証中**
- **デスクトップ環境の起動中**（仮想デスクトップセッションの場合）または**アプリケーションウィンドウの起動中**（仮想アプリケーションセッションの場合）

KDEおよびXFCEデスクトップ環境のキーボードレイアウト同期のサポート

このリリースでは、動的キーボードレイアウト同期機能がXFCEおよびKDE Plasma 5デスクトップ環境に拡張されました。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。

evdevをデフォルトのX Keyboard Extension (XKB) ルールにする

より応答性が高く予測可能なキーボード動作で優れたユーザーエクスペリエンスを実現するために、Linux VDAではキーボード構成のデフォルトXKBルールとしてevdevを指定するようになりました。

詳しくは、「X Keyboard Extension (XKB) の構成」を参照してください。

LDAPクエリパフォーマンスの最適化

この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。

Cloud ConnectorのVDAスケーラビリティの向上

このリリースでは、VDAのスケーラビリティが改善され、各Citrix Cloud Connector（4 vCPU、10 GBメモリ）で6,000台のLinux VDAをサポートできるようになりました。詳しくは、「Cloud Connectorのサイズとスケールの考慮事項」を参照してください。

HDXアダプティブスループットがデフォルトで有効に

詳細については、(/ja-jp/linux-virtual-delivery-agent/2407/configure/session/hdx-adaptive-throughput.html)を参照してください。

> **注:**
>
> 展開管理にAnsibleの使用に関心のあるお客様向けに、包括的なベストプラクティス記事を提供しています。詳しくは、「Ansibleを使用した展開の管理」を参照してください。

2402 LTSRの新機能

RHEL 8.9、Rocky Linux 8.9、RHEL 9.3、およびRocky Linux 9.3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 8.9
- ロッキー・リナックス 8.9
- RHEL 9.3
- ロッキー・リナックス 9.3

詳細については、システム要件を参照してください。

新しい.NETバージョンが必要

本リリース以降、Linux VDAをインストールする前に、RHEL 7.9およびAmazon Linux 2を除くサポートされているすべてのLinuxディストリビューションに.NET Runtime 8.0をインストールする必要があります。RHEL 7.9およびAmazon Linux 2については、引き続き.NET Runtime 6.0をインストールしてください。

必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールしてください。そうでない場合は、Microsoftパッケージフィードから.NETをインストールしてください。

RHEL、Rocky Linux、およびUbuntu向けのXfceサポート

RHEL 8/9、ロッキー・リナックス 8/9、およびUbuntu 20.04/22.04でホストされているVDAセッションで、**Xfce**デスクトップ環境を使用できるようになりました。

詳細については、システム要件を参照してください。

スマートカードおよびフェデレーション認証サービス (FAS) のサポートが、Quest参加済みLinux VDAに拡張されました

スマートカードおよびFASのサポートは、Questを使用してドメインに参加しているLinux VDAでも利用できるようになりました。この機能は、Linux VDAがスマートカードおよびFAS認証をサポートしているRHEL、CentOS、およびUbuntuディストリビューションで完全にテストされています。

詳細については、スマートカードおよびフェデレーション認証サービスを参照してください。

インストールされている依存関係を確認する機能

XDPingの機能を拡張しました。これにより、Linux VDAにインストールされている依存関係を**XDPing**で確認できるようになりました。これを行うには、**XDPing**実行可能ファイルを実行してインストールされている依存関係を表示するか、表示しながら特定のパスに保存します。

詳細については、**XDPing**で実行できるタスク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ストリーミングを以下のディストリビューションに拡張しました。

- SUSE 15.5

詳細については、Citrix ProvisioningドキュメントのCitrix Provisioningを使用したLinux VDAの作成および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。

ログオン時間の内訳

この機能は、Citrix DirectorおよびMonitorでユーザーログオンプロセスにかかる各ステップの期間を表示します。これにより、ログオンの問題をより効果的にトラブルシューティングできます。ユーザーログオンプロセスは、**HDX接続**、**認証**、**GPO**などのフェーズに分類されます。詳細については、ログオンプロセスフェーズおよびLinux VMとLinuxセッションメトリックを参照してください。

公開アプリケーションの動的ウィンドウプレビュー

公開アプリケーションのウィンドウで画面上のコンテンツをプレビューできるようになりました。たとえば、Windows用Citrix Workspace™アプリを使用してGoogle Chromeセッションの複数のウィンドウを開き、タスクバーのChromeアイコンにカーソルを合わせると、コンテンツをプレビューできます。

この機能は、Citrix Studioで設定できる**動的ウィンドウプレビュー**ポリシーに依存します。詳細については、動的ウィンドウプレビューおよびポリシーサポートリストを参照してください。

新規および最適化されたセッションメトリッククエリユーティリティ

このリリースでは、Windows VDAとの整合性を図り、Windowsユーザーエクスペリエンスを提供するために、新しいセッションメトリッククエリユーティリティ（**ctxsession**）が導入されました。このユーティリティを使用すると、より多くのセッションメトリックにアクセスできます。

このリリースでは、ヘルプ情報も最適化され、**ctxqsession**、**ctxquser**、**ctxqfull**、**ctxquery**などのレガシーセッションメトリッククエリユーティリティにアクセス制御が追加されました。以前はアクセス制御がなかったため、Linux VDAのログインユーザーは、いずれかのツールを介してすべてのセッションのメトリックを照会できました。アクセス制御が追加された現在では、ユーザーは自分のセッションメトリックのみを照会でき、rootユーザーと**ctxadm**グループのユーザーのみが他のユーザーのセッションメトリックを照会する権限を持っています。

詳細については、セッションメトリッククエリユーティリティを参照してください。

2311 の新機能

SUSE 15.5およびDebian 11.7のサポート

Linux VDAは、SUSE 15.5およびDebian 11.7をサポートするようになりました。詳しくは、「システム要件」を参照してください。

HDX アダプティブスループット

Linux VDAは、HDXアダプティブスループットをサポートするようになりました。この機能は、出力バッファを調整することでICAセッションのピークスループットをインテリジェントに微調整し、ユーザーエクスペリエンスを向上させます。詳しくは、「HDXアダプティブスループット」の記事を参照してください。

AOMedia Video 1（AV1）のサポート

このリリースでは、新しいコーデックであるAV1が導入され、同じ帯域幅使用量でより高品質の画像を受信でき、低帯域幅でもより高いFPSを実現できます。AV1は、H.264/H.265と比較して、フレームあたりの帯域幅が少なくなります。詳しくは、「グラフィックの構成と微調整」を参照してください。

> **ヒント:**
>
> NVIDIA GPUでHDX 3D Proを使用するには、NVIDIA Capture SDKバージョン8をサポートするNVIDIAグラフィックドライバーバージョンをインストールする必要があります。詳しくは、「NVIDIA Capture SDKのドキュメント」を参照してください。

システムトレイでグラフィックステータスを利用可能

セッションユーザーは、セッション内のシステムトレイアイコンをクリックすることで、グラフィックステータスを表示できるようになりました。

詳しくは、「システムトレイ」を参照してください。

複数のオーディオデバイスのサポート（プレビュー）

このリリースでは、オーディオリダイレクト機能が導入されました。この機能により、Citrix Workspaceアプリがインストールされているクライアントマシン上の複数のオーディオデバイスを、リモートのLinux VDAセッションにリダイレクトできます。

この機能を有効にすると、次のようになります。

- クライアントマシン上のすべてのローカルオーディオデバイスがセッションに表示されます。CitrixAudioSink（オーディオ出力）またはCitrixAudioSource（オーディオ入力）の代わりに、オーディオデバイスはそれぞれのデバイス名で表示されます。セッション内のアプリでオーディオデバイスを選択するか、セッション中にデフォルトのオーディオデバイス（クライアントマシンのデフォルトオーディオデバイスでもある）を使用できます。必要に応じて、クライアントマシンのシステム設定からデフォルトのオーディオデバイスを変更できます。クライアントマシンのデフォルトオーディオデバイスが更新されると、新しいデバイスがセッションのデフォルトオーディオデバイスとして表示されます。
- オーディオデバイスを接続または取り外すと、セッション内のオーディオデバイスが動的に更新されます。

詳しくは、「複数のオーディオデバイスのサポート」を参照してください。

トークンベースの登録（プレビュー）

ドメインに参加していないVDAをマシンカタログに登録し、トークンファイルを使用してこれらのVDAをCitrix Cloud™コントロールプレーンに認証できるようになりました。

トークンベースの登録は、Citrix以外のプロビジョニング技術を使用して（物理または仮想を問わず）マシンを独自に準備するユースケースに最適です。これにより、以下の利点が得られます。

- Cloud Connectorのインストールとメンテナンスが不要になります。
- ユーザーおよびマシン認証におけるADへの依存を排除し、ドメインに参加していないマシンの認証を可能にします。

詳しくは、「easy install を使用してドメインに参加していないLinux VDAを作成する（プレビュー）」を参照してください。

easy installの機能強化

お客様のエクスペリエンスを向上させるため、easy install機能をリファクタリングし、簡素化しました。これにより、次のことが可能になります。

- **easy install**を使用して、ドメインに参加していないVDAを作成します。この機能はプレビュー版です。詳しくは、「easy install を使用してドメインに参加していないLinux VDAを作成する（プレビュー）」を参照してください。
- easy installスクリプト（ctxinstall.sh）をすべてのモジュールまたは特定のモジュールで実行します。モジュール実行の手順については、**ctxinstall.sh -h**を実行してヘルプ情報を確認してください。例が提供されています。ctxinstall.shの詳細については、「easy install を使用してドメインに参加しているVDAを作成する」の記事の「ctxinstall.sh」セクションを参照してください。
- 変数がユースケースに応じてグループ化されているctxinstall.conf.tmplファイルについて、より深く理解できます。

Linux VDAデータ収集プログラム

Linux VDAのインストール後、データ収集プログラムに自動的に参加するようになりました。データ収集プログラムは、統計情報と使用状況データを収集し、Citrix製品の品質とパフォーマンスの向上に役立てるためにCitrix Analyticsに送信します。プログラムを無効にし、詳細を表示するには、「Linux VDAデータ収集プログラム」を参照してください。

Linuxセッションで利用可能な新しいメトリック

このリリースでは、Citrix DirectorおよびMonitorのLinuxセッションに2つの新しいメトリックが追加されました。

- **ICA®遅延**

  ICA Latency は基本的にネットワーク遅延です。このメトリックは、ネットワークが遅いかどうかを示します。

- **ポリシー**

  現在のセッションで有効なすべてのポリシーは、**セッションの詳細**ビューの「ポリシー」タブに表示されます。

詳しくは、「Linux VM と Linux セッションのメトリック」(/ja-jp/linux-virtual-delivery-agent/2311/configure/administration/linux-vm-and-linux-session-metrics.html)を参照してください。

強化されたEDT輻輳制御が一般提供開始

Enlightened Data Transport (EDT) プロトコルを最適化するために、新しい輻輳制御アルゴリズムが導入されました。この実装により、EDT はより高いスループットを達成し、遅延を削減して、ユーザーエクスペリエンスを向上させることができます。この機能はデフォルトで有効になっています。詳しくは、「アダプティブトランスポート」(/ja-jp/linux-virtual-delivery-agent/2311/configure/session/adaptive-transport.html)を参照してください。

Rendezvous V2 がデフォルトで有効

Rendezvous プロトコルはデフォルトで無効になっています。Rendezvous プロトコルが有効になっている場合、V1 ではなく Rendezvous V2 が適用されます。

EDT MTU 検出がVDAでデフォルトで有効に

MTU 検出により、EDT はセッション確立時に最大伝送単位 (MTU) を自動的に決定できます。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、「アダプティブトランスポート」(/ja-jp/linux-virtual-delivery-agent/2311/configure/session/adaptive-transport.html)を参照してください。

Secure HDX (プレビュー)

Citrix Workspace アプリ (クライアント) と VDA (セッションホスト) の間で、ICA セッションをエンドツーエンドで暗号化できます。

エンドツーエンド暗号化 (E2EE) 機能により、Citrix Gateway を含む中間ネットワーク要素が ICA トラフィックを復号化することはできません。これにより、環境のセキュリティ体制が向上し、構成と管理が容易になります。詳しくは、「Secure HDX (プレビュー)」(/ja-jp/linux-virtual-delivery-agent/2311/configure/session/secure-hdx.html)を参照してください。

2308の新機能

Linux VDA のバージョン 2308 には、次の新機能と機能強化が含まれています。

RHEL 8.8、Rocky Linux 8.8、RHEL 9.2、および Rocky Linux 9.2 のサポート

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

- RHEL 8.8
- ロッキー・リナックス 8.8
- RHEL 9.2
- ロッキー・リナックス 9.2

詳細については、「システム要件」を参照してください。

マルチセッション Linux VDA での共有 GPU アクセラレーション

マルチセッション Linux VDA で共有 GPU アクセラレーションを有効にして、OpenGL 3D アプリケーションを高速化できるようになりました。詳細については、「マルチセッション Linux VDA での共有 GPU アクセラレーション」を参照してください。

特定のデータ形式をセッションとクライアント間でコピー＆ペースト可能

VDA セッションとクライアントデバイス間で特定のデータ形式をコピー＆ペーストできるようになりました。この機能は Citrix ポリシーによって実現されます。詳細については、「クリップボードリダイレクト」を参照してください。

Quest サポートの拡張

Quest サポートを RHEL 8.X、Rocky Linux 8.x、RHEL 9.x、および Rocky Linux 9.x に拡張しました。詳細については、「システム要件」の「**Active Directory 統合パッケージ**」セクションを参照してください。

ブラウザコンテンツリダイレクトのサーバーフェッチとクライアントレンダリング

ブラウザコンテンツリダイレクトを拡張し、サーバーフェッチとクライアントレンダリングのシナリオをサポートしました。このシナリオでは、Citrix Workspace アプリ (クライアント) が仮想チャネル (**CTXPFWD**) を使用して VDA 経由で Web サーバーに接続し、コンテンツを取得します。このオプションは、クライアントが Web サーバーにアクセスできない場合 (シンクライアントなど) に役立ちます。VDA の CPU と RAM の消費量は削減されますが、ICA 仮想チャネルで帯域幅が消費されます。詳細については、「ブラウザコンテンツリダイレクト」を参照してください。

EDT 輻輳制御の強化 (プレビュー)

新しい輻輳制御アルゴリズムが導入され、Enlightened Data Transport (EDT) プロトコルが最適化されました。この実装により、EDTはより高いスループットを達成し、遅延を削減して、ユーザーエクスペリエンスを向上させることができます。この機能はデフォルトで無効になっています。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。

システムトレイのメニュー項目を増やす

画面共有システムトレイツールを拡張し、より多くのメニュー項目に対応しました。現在、セッションユーザーはシステムトレイアイコンをクリックして、以下のすべてのメニュー項目にアクセスし、対応するアクションを実行できます。

- 画面共有

- デスクトップ環境の切り替え
- グラフィック品質スライダー
- リアルタイムCQI

詳細については、「システムトレイ」を参照してください。

強化されたH.265ハードウェアエンコーディング

H.265ハードウェアエンコーディングを拡張し、画面全体のロスレス圧縮と、アクティブに変化する領域のロッシーH.265ハードウェアコーデックを可能にしました。詳細については、「グラフィックの構成と微調整」を参照してください。

SQLiteおよびRC4の問題チェックをサポートするためのXDPingの強化

Linux **XDPing**ツールが実行できる個別のテストとVDA登録ステータスチェックの範囲を拡張し、それぞれSQLiteテストとRC4の問題チェックを含めるようにしました。詳細については、「**XDPing**」を参照してください。

Amazon Linux 2、RHEL 7.9、CentOS 7.9で徹底的にテストされたRealm

realmを使用してLinux VMをActive Directoryドメインに参加させる機能は、Amazon Linux 2、RHEL 7.9、およびCentOS 7.9で徹底的にテストされました。Amazon Linux 2、RHEL 7.9、およびCentOS 7.9で実行されているLinux VMをActive Directoryドメインに参加させるには、引き続き**net ads**コマンドを使用できます。

詳細については、「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」を参照してください。

2305 の新機能

Linux VDAバージョン2305には、以下の新機能と機能強化が含まれています。

SQLiteのサポート

SQLiteは完全にサポートされるようになりました。Linux VDAパッケージのインストール後、***/etc/xdl/db.conf***を編集して、使用するSQLiteまたはPostgreSQLを指定できます。使用するデータベースの指定について詳しくは、簡単なインストールおよび手動インストールの記事を参照してください。

> **注:**
>
> Linux **XDPing**ツールはSQLiteテストを実行しません。

セッションログオンの機能強化

このリリースでは、以下の点からセッションログオンエクスペリエンスが強化されています。

- ユーザーがパスワードとPINコードの表示を切り替えられるようにすることで、誤った入力を簡単に特定できるようにします。
- セッションログオンバナーメッセージと認証ダイアログを別々の画面に表示します。
- セッションログオンプロセスを表示し、非SSOシナリオでのセッション再接続にはパスワードまたはPINコードを要求します。
- ユーザーが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VDAセッションにログオンした場合、認証ダイアログは別々に表示されます。ユーザーは、資格情報が誤って入力された時期を簡単に特定できます。
- 非SSOシナリオにおけるユーザー認証方法の以下の組み合わせをサポートします。

CITRIX WORKSPACEアプリ	VDAセッション
FAS	ユーザー名
FAS	スマートカード

詳しくは、セッションログオン画面のカスタム背景とバナーメッセージおよび非SSO認証を参照してください。

一時ホームディレクトリ設定に利用できるレジストリキーの追加

このリリースでは、一時ホームディレクトリの設定に役立つレジストリキーがさらに2つ導入されました。

- CheckUserHomeMountPoint
- CheckMountPointRetryTime

詳しくは、「一時ホームディレクトリでのログオン」(/ja-jp/linux-virtual-delivery-agent/2305/configure/session/logon-with-temp-home-directory.html)を参照してください。

Fast Identity Online (FIDO2) 認証のサポート (プレビュー)

Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするために、FIDO2認証を設定できるようになりました。詳しくは、「FIDO2 (プレビュー)」(/ja-jp/linux-virtual-delivery-agent/2305/configure/authentication/fido2.html)を参照してください。

HDX 3D PROでのH.264ロスレス圧縮のサポート

NVIDIA GPUによるHDX 3D PROハードウェアアクセラレーションで、H.264ロスレス圧縮のサポートが利用可能になりました。詳しくは、「H.264ロスレス圧縮」(/ja-jp/linux-virtual-delivery-agent/2305/configure/graphics/configuring-graphics.html#h264-lossless-compression)を参照してください。

リダイレクトするURLを指定するためのワイルドカードサポートの拡張

コンテンツをクライアントにリダイレクトできるURLを指定する場合、プロトコルを除くすべてのURLコンポーネントを表すために、`*`ワイルドカードを使用できるようになりました。詳しくは、「ブラウザコンテンツリダイレクト」(/ja-jp/linux-virtual-delivery-agent/2305/configure/multimedia/browser-content-redirection.html)を参照してください。

XDPingを使用したVDAデータバックアップと比較のサポート

トラブルシューティングのために、**XDPing**ツールにVDAバックアップモジュールを追加しました。このモジュールを使用すると、構成、データベース、バイナリ権限データなど、VDAの主要データをいつでもバックアップできます。VDAが正常に動作しているときに、VDAの主要データをバックアップできます。後でVDAが故障した場合に備えて、データの別のコピーをバックアップし、2つのデータコピーを比較してトラブルシューティングを容易にすることができます。詳しくは、「**XDPing**」(/ja-jp/linux-virtual-delivery-agent/2305/configure/administration/tools-and-utilities.html#xdping)を参照してください。

MCSを使用してRHEL 8.x/9.xおよびRocky Linux 8.x/9.x VDAを作成するためのSSSDサポート

MCSを使用してRHEL 8.x/9.xおよびRocky Linux 8.x/9.x VDAを作成する際に、System Security Services Daemon (SSSD) を使用してActive Directoryドメインに参加できるようになりました。

> **注:**
>
> SSSDを使用してドメインに接続されている、現在実行中のRHEL 8.x/9.xまたはRocky Linux 8.x/9.x VDAをMCSのテンプレートVMとして使用するには、以下を確認してください。
>
> - VDAは手動でインストールされ、簡易インストールは使用されません。簡易インストールはRHEL 8.x/9.xおよびRocky Linux 8.x/9.x**Adcli**を使用し、SSSDと**Adcli**の組み合わせはMCSでサポートされていません。

- Sambaサーバーは、AD認証にSSSDを使用するように構成されています。詳細については、`<https://access.redhat.com/solutions/3802321>`のRed Hatの記事を参照してください。

詳細については、「MCSを使用してLinux VDAを作成する」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ストリーミングは以下のディストリビューションに拡張されました。

- RHEL 8.7
- ロッキー Linux 8.7
- SUSE 15.4

詳細については、Citrix Provisioningドキュメントの「Citrix Provisioningを使用してLinux VDAを作成する」および「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。

2303の新機能

Linux VDAバージョン2303には、以下の新機能と機能強化が含まれています。

RHEL 9.1およびRocky Linux 9.1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 9.1およびRocky Linux 9.1を追加しました。詳細については、「システム要件」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ストリーミングは以下のディストリビューションに拡張されました。

- RHEL 9.1
- RHEL 9.0
- ロッキー Linux 9.1
- ロッキー Linux 9.0

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

Easy installの機能強化

Easy install機能を強化し、サイレントモードと対話モードの両方でユーザーエクスペリエンスを向上させました。主な機能強化は次のとおりです。

- GUIおよびctxinstall.shに必要なすべての環境変数の値を設定、保存、同期するために、Easy installの構成ファイルとして **/opt/Citrix/VDA/sbin/ctxinstall.conf** を追加しました。
- 変数構成の検証を追加し、堅牢性を向上させました。
- ctxinstall.shを実行するためのコマンドオプションを追加しました。詳細については、ヘルプコマンド **ctxinstall.sh -h** を使用してください。

Easy installの使用方法の詳細については、Create domain-joined VDAs using easy install を参照してください。

XDPing の機能強化

XDPing ツールに分析モジュールを追加し、VDA登録ステータスを確認および分析できるようにしました。VDA登録ステータスを確認するには、コマンド **sudo /opt/Citrix/VDA/bin/xdping -a** を実行します。

XDPing のテストカテゴリを拡張し、依存関係、Kerberos、PAMのチェックを含めました。特定のチェックを実行するには、コマンド **sudo /opt/Citrix/VDA/bin/xdping -T dependency,kerberos,pam** を実行します。

詳細については、**XDPing**、Federated Authentication Service、および Smart cards を参照してください。

2301の新機能

Linux VDA バージョン 2301 には、次の新機能と機能強化が含まれています。

RHEL 8.7、Rocky Linux 8.7、および SUSE 15.4 のサポート

RHEL 8.7、Rocky Linux 8.7、および SUSE 15.4 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。

> **注:**
>
> SUSE 15.3 は、このリリースから非推奨になりました。SUSE 15.4 で Linux VDA を実行するには、VDA を新規インストールしてください。

Wayland のサポート (プレビュー)

Linux VDA は、RHEL 9.0、Rocky Linux 9.0、および Ubuntu 22.04 の GNOME で Wayland をサポートするようになりました。詳しくは、「Wayland (プレビュー)」を参照してください。

すべてのクライアントプリンターを Linux VDA セッションにマップできるようになりました

以前は、クライアントデバイスのデフォルトプリンターのみを Linux VDA セッションにマップできました。このリリース以降、クライアントデバイスのすべてのプリンターを Linux VDA セッションにマップできます。詳しくは、「印刷のベストプラクティス」および「PDF 印刷」を参照してください。

動的クライアントドライブマッピングとクライアントフォルダーリダイレクト

以前は、セッション開始後にクライアントに接続されたドライブはセッションにマップされませんでした。それらのドライブをセッションでアクセス可能にするには、セッションを切断して再接続する必要がありました。このリリース以降、セッション中にいつでもクライアントに接続されたドライブは自動的にマップできます。さらに、このリリースでは、クライアント上のローカルドライブのカスタム部分をセッションに動的にリダイレクトできるクライアントフォルダーリダイレクトが導入されました。詳しくは、「クライアントドライブマッピング」を参照してください。

2212 の新機能

Linux VDA バージョン 2212 には、次の新機能と機能強化が含まれています。

RHEL 9.0 および Rocky Linux 9.0 のサポート

RHEL 9.0 および Rocky Linux 9.0 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。

USB デバイスリダイレクトの機能強化

USB デバイスリダイレクトについて、次の機能強化を行いました。

- USB デバイスリダイレクトの新しい仮想ホストコントローラーとして USB/IP を使用するようになりました。ほとんどの場合、USB/IP カーネルモジュールは Linux カーネルバージョン 3.17 以降でリリースされているため、デフォルトでカーネルモジュールをビルドする必要はありません。
- USB 3.0 ポートのサポートを追加しました。クライアントデバイスの USB 3.0 ポートに USB 3.0 デバイスを挿入できるようになりました。

- バルク転送を最適化し、転送速度を向上させました。この向上は、高遅延環境でより顕著になります。
- USB デバイスリダイレクトを、**TD-RDF5A Transcend USB device** や複合 USB デバイスである **Yubico YubiKey OTP+FIDO+CCID** などのより多くの USB デバイスに拡張しました。

詳しくは、「USB デバイスリダイレクト」を参照してください。

NVIDIA 製の特定の非仮想化 GPU のハードウェアアクセラレーション

Linux VDA ドキュメントでは、非仮想化 GPU は以下を指します。

- Remote PC Access シナリオで使用される GPU
- ハイパーバイザーからパススルーされた GPU

HDX 3D Pro グラフィックを優れたパフォーマンスでレンダリングするために、NVIDIA 製の特定の非仮想化 GPU のハードウェアアクセラレーションを有効にしました。特定の非仮想化 GPU については、NVIDIA Linux Capture SDK のリリースノートのサポートされているハードウェアのセクションを参照してください。

詳しくは、「非仮想化 GPU」を参照してください。

スマートカードおよび FAS 認証における pam_krb5 の SSSD への置き換え

以前は、RHEL 8 および Rocky Linux 8 でスマートカードまたは Federated Authentication Service (FAS) 認証を使用するには、pam_krb5 モジュールを手動でビルドする必要がありました。このリリース以降、VDI モードで提供される RHEL 8 および Rocky Linux 8 マシンでは、この手動手順を実行する必要がなくなりました。詳しくは、「Federated Authentication Service」および「スマートカード」を参照してください。

MCS 構成を含むように簡易インストール GUI を拡張

以前は、`/etc/xdl/mcs/mcs.conf` ファイルを編集することによってのみ MCS 変数を構成できました。このリリースでは、簡易インストール GUI が MCS 構成を含むように拡張され、GUI を使用して MCS 変数を構成することもできるようになりました。GUI で **Deploy** をクリックすると、GUI で設定した変数が `/etc/xdl/mcs/mcs.conf` ファイルで設定した変数を上書きします。詳しくは、「手順 3：マスターイメージを準備する」を参照してください。

Linuxセッションの記録のサポート (プレビュー)

Linux VDAでホストされているセッションを記録および再生できるようになりました。詳しくは、セッションの記録 (プレビュー)を参照してください。

データベースオプションが利用可能になりました (プレビュー)

PostgreSQLに加えてSQLiteも使用できるようになりました。Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集することで、使用するSQLiteまたはPostgreSQLを指定できます。そのためには、次の手順を実行します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略してください。
2. 使用するデータベースを指定するために`/etc/xdl/db.conf`を編集します。
3. インストール方法に応じて、次のいずれかのスクリプトを実行します。

- イージーインストールを使用している場合は、`/opt/Citrix/VDA/sbin/ctxinstall.sh`または`/opt/Citrix/VDA/bin/easyinstall`
- Linux VDAを手動でインストールしている場合は、`/opt/Citrix/VDA/sbin/ctxsetup.sh`
- MCSを使用してLinux VDAを作成している場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`

注:

- VDIモードではSQLiteのみを使用することをお勧めします。
- 手動インストールの場合、SQLiteとPostgreSQLを切り替える前に、それらを個別に手動でインストールする必要があります。イージーインストールおよびMCSの場合、手動でインストールすることなくSQLiteとPostgreSQLを切り替えることができます。
- PostgreSQLのポート番号を設定するには、`/etc/xdl/db.conf`を使用することもできます。

詳しくは、インストールに関する記事を参照してください。

2210の新機能

Linux VDAバージョン2210には、以下の新機能と機能強化が含まれています。

HDX 3D Pro向けGPUハードウェアアクセラレーションの強化

GPUとLinuxシステムメモリ間のデータ転送効率を改善しました。また、3Dグラフィックレンダリングとハードウェアエンコーディングの遅延を削減しました。これらの機能強化により、ハードウェアリソースの利用が最適化され、フレーム/秒 (FPS) のパフォーマンスが大幅に向上します。詳しくは、「H.264ハードウェアエンコーディング」を参照してください。

クリップボードを介したデータ転送のサイズ制限のサポート

ユーザーが1回のコピーアンドペースト操作でクライアントとLinux仮想セッション間で転送できるクリップボードデータの最大サイズ（KB単位）を指定できます。これを行うには、以下のポリシー設定を使用します。

- クリップボードのクライアントからセッションへの転送サイズを制限する
- クリップボードのセッションからクライアントへの転送サイズを制限する

ポリシー設定について詳しくは、「ICAポリシー設定」の「クライアントクリップボードリダイレクト」セクションを参照してください。

Linux VDAがサポートするポリシー設定については、「ポリシーサポートリスト」を参照してください。

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ストリーミングを以下のディストリビューションに拡張しました。

- RHEL 8.6
- ロッキー リナックス 8.6
- ウブントゥ 22.04

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。

シェルスクリプトの機能強化

シェルスクリプトを保守しやすいように強化し、以下の構成テンプレートを`/etc/xdl/mcs/`から`/etc/xdl/ad_join/`に移動しました。

- winbind_krb5.conf.tmpl
- winbind_smb.conf.tmpl
- sssd.conf.tmpl
- sssd_krb5.conf.tmpl
- sssd_smb.conf.tmpl
- centrify_krb5.conf.tmpl
- centrify_smb.conf.tmpl

簡易インストールでは、特定のドメイン参加方法に関連する構成テンプレートも使用されます。

2209 の新機能

Linux VDA のバージョン 2209 には、次の新機能と機能強化が含まれています。

RHEL 8.6、Rocky Linux 8.6、および Ubuntu 22.04 のサポート

RHEL 8.6、Rocky Linux 8.6、および Ubuntu 22.04 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。

SSSD および PBIS に参加している Ubuntu および SUSE VDA のフェデレーション認証のサポート

フェデレーション認証サービス (FAS) のサポートを以下の種類の VDA に拡張しました。

- SSSD および PBIS を使用してドメインに参加している Ubuntu VDA
- SSSD を使用してドメインに参加している SUSE VDA。

FASを使用して、それらのVDAにログオンするユーザーを認証できるようになりました。詳細については、「フェデレーション認証サービス」を参照してください。

セッションユーザーによるカスタムデスクトップ環境

セッションユーザーは、デスクトップ環境をカスタマイズできるようになりました。この機能を有効にするには、VDAにデスクトップ環境を事前にインストールする必要があります。詳細については、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

GPG署名

公開キーを使用してLinux VDAパッケージの整合性を検証できるように、GPGを使用してLinux VDAパッケージに署名しました。

公開キーを取得するには：

1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. ［**Components**］ をクリックして、公開キーを見つけてダウンロードします。

公開キーを使用してLinux VDAパッケージの整合性を検証するには：

- RPMパッケージの場合、公開キーをRPMデータベースにインポートし、次のコマンドを実行します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

- DEBパッケージの場合、公開キーをDEBデータベースにインポートし、次のコマンドを実行します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

2207の新機能

Linux VDAバージョン2207には、次の新機能と機能強化が含まれています。

非SSOシナリオでのより多くのユーザー認証方法のサポート

以前は、シングルサインオン (SSO) が無効になっている場合、ユーザーは異なるユーザー名を使用してCitrix Workspaceアプリとセッションにログオンできました。このリリースから、ユーザーはユーザー名またはスマートカードを使用してログオンできます。詳細については、「非SSO認証」を参照してください。

セッションウォーターマークを構成するための追加オプション

ウォーターマークカスタムテキストポリシーに新しいオプションを追加し、セッションウォーターマークをさらにカスタマイズできるようにしました。たとえば、PNGウォーターマークを作成するには、**ウォーターマークカスタムテキスト**を**<image=path to a PNG image on the VDA>**に設定できます。詳細については、「セッションウォーターマーク」を参照してください。

KVMハイパーバイザーのサポート

Kernel-based Virtual Machine (KVM) は、Linux向けの主要なオープンソース仮想化テクノロジーです。これはLinuxの一部であり、Linuxマシンをハイパーバイザーに変えることで、複数の分離された仮想マシン (VM) をホストできるようにします。Linux VDAはKVMハイパーバイザーで適切に動作します。ただし、KVMハイパーバイザーは、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。

サポートされているすべてのLinuxディストリビューションでファイルコピーアンドペーストが利用可能

以前は、**ファイルコピーアンドペースト機能**はDebian 10、RHEL 7.9、およびUbuntu 18.04でのみ利用可能でした。このリリースから、この機能はLinux VDAがサポートするすべてのLinuxディストリビューションに拡張されました。詳細については、「ファイルコピーアンドペースト」を参照してください。

2206の新機能

Linux VDAバージョン2206には、次の新機能と機能強化が含まれています。

Debian 11.3のサポート

サポートされているディストリビューションとしてDebian 11.3を追加しました。

Azure AD資格情報によるユーザー認証のサポート

Linux VDAは、ユーザー認証を提供するためにAzure Active Directory (AD) と統合できるようになりました。この機能により、ユーザーはAzure AD資格情報を使用してドメインに参加していないLinuxデスクトップにログオンできます。詳細については、「Azure Active Directoryによる認証」を参照してください。

クライアントバッテリーステータス表示

Linux VDAは、仮想デスクトップ内のクライアントデバイスのバッテリーステータスをリダイレクトして表示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。詳細については、「クライアントバッテリーステータス表示」を参照してください。

非vGPUグラフィックカードを使用するRemote PC Access VDAのマルチモニター機能強化

このリリースでは、ユーザーがリモートPCアクセスセッションにアクセスする際に発生する以下の問題に対処しています。

- リモートPCのモニターをブラックアウトできません。セッションコンテンツを表示すると、セキュリティリスクが生じる可能性があります。
- 複数のモニターを使用してリモートPCアクセスセッションにアクセスすると、モニターにアーティファクトが表示されることがあります。

詳細については、「非vGPUグラフィックカード」を参照してください。

セッションログオン画面のカスタム背景とバナーメッセージ

カスタム背景とバナーメッセージを使用して、セッションログオン画面をカスタマイズできるようになりました。詳細については、「セッションログオン画面のカスタム背景とバナーメッセージ」を参照してください。

自動DPIスケーリング

Linux VDAは自動DPIスケーリングをサポートするようになりました。ユーザーが仮想デスクトップまたはアプリケーションセッションを開くと、セッション内のDPI値がクライアント側のDPI設定と一致するように自動的に変更されます。詳細については、「自動DPIスケーリング」を参照してください。

LDAPクエリの高速化

Linux VDAは、ポーリングモードで各LDAPサーバーをクエリするだけでなく、マルチドメインフォレスト内のLDAPサーバーを同時にクエリできるようになりました。LDAPクエリを高速化するには、ドメインコントローラーで**グローバルカタログ**を有効にし、関連するLDAPポート番号を3268として指定します。LDAPサーバーとLDAPポートの指定方法については、お使いのディストリビューションに基づいたLinux VDAインストール記事のCTX_XDL_LDAP_LIST変数に関する説明を参照してください。

カスタムFASポート

以前は、Linux VDAはポート80を介してFASサーバーと通信するようにハードコードされていました。このリリース以降、ctxsetup.shのCTX_XDL_FAS_LISTを介してLinux VDA側でFASポートをカスタマイズできます。詳細については、お使いのディストリビューションに基づいたLinux VDAインストール記事を参照してください。

2204の新機能

Linux VDAバージョン2204には、以下の新機能と機能強化が含まれています。

ハードウェアエンコーディングが選択的コーデックに拡張されました

以前は画面全体でのみ利用可能でしたが、H.264ハードウェアエンコーディングは、アクティブに変化する領域での選択的な使用をサポートするようになりました。この機能により、CPUのビデオ圧縮処理がハードウェアにオフロードされ、画質とフレームレート（FPS）が向上します。この機能を有効にする方法については、「グラフィック構成と微調整」を参照してください。

一時ホームディレクトリでのログオンのサポート

このリリースから、Linux VDA上のマウントポイントが失敗した場合に備えて、一時ホームディレクトリを指定できるようになりました。一時ホームディレクトリが指定されている場合、マウン

トポイントが失敗すると、セッションログオン中にプロンプトが表示されます。ユーザーデータは一時ホームディレクトリに保存されます。詳細については、「一時ホームディレクトリでのログオン」を参照してください。

Rendezvous HDXトラフィックのSOCKS5プロキシサポート

Linux VDAは、Rendezvous接続を確立するためのSOCKS5プロキシサポートを拡張しました。詳細については、「Rendezvous V1」および「Rendezvous V2」を参照してください。

Rendezvousの透過型プロキシサポート

Rendezvousで透過型HTTPプロキシがサポートされるようになりました。ネットワークで透過型プロキシを使用している場合、VDAに追加の設定は不要です。

GNOME Classicデスクトップのサポート

このリリースでは、ctxsetup.shの`CTX_XDL_DESKTOP _ENVIRONMENT`を介して指定できるデスクトップオプションとしてGNOME Classicが追加されました。詳細については、お使いのディストリビューションに基づいたLinux GNOME Classic VDAインストール記事を参照してください。

アプリのタスクバーボタンのグループ化

以前は、同じ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が1つのグループにまとめられていました。このリリースから、アプリの複数のウィンドウが開いている場合でも、各アプリは単一のタスクバーボタンとして表示されます。

2203の新機能

Linux VDAバージョン2203には、次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2つの種類があります。1) VDA登録およびセッションブローカリング用の制御トラフィック、2) HDXトラフィック。

Rendezvous V1 では HDX トラフィックが 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すべての制御トラフィックをプロキシするには Cloud Connector が引き続き必要です。

Rendezvous V2 をシングルセッションおよびマルチセッションの Linux VDA で使用する場合、標準の AD ドメイン参加済みマシンとドメイン非参加マシンがサポートされます。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 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。

簡単インストール GUI が利用可能になりました

以前は、簡単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、次の操作をガイドする簡単インストール 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

簡単インストール 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、「簡単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、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します。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。

2201 の新機能

Linux VDA のバージョン 2201 には、次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。デスクトップ環境について詳しくは、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イング機能が、サポートされているすべてのディストリビューションに拡張されました

セッションシャドウイング機能を使用すると、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イング」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はデフォルトの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、次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（プレビュー）

サポート対象のディストリビューションとしてAmazon Linux 2を追加しました。詳細については、「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

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上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。詳細については、「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。詳細については、「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が、Rendezvous接続用の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。

- スーゼ Linux Enterprise Server 15 SP2 (スーゼ 15.2)
- スースー リナックス エンタープライズ サーバー 15 SP3 (スースー 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

サポートされるディストリビューションとして、SUSE Linux Enterprise Server 15 SP3 (SUSE 15.3)およびSUSE Linux Enterprise Server 15 SP2 (SUSE 15.2)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。

SUSE 15.3、SUSE 15.2、およびSUSE 12.5向けの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に次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の実行」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、異なる資格情報を使用してCitrix Virtual Apps™および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、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 10.9
- SUSE 15.3
- SUSE 15.2

2109の新機能

Linux VDAバージョン2109には、以下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（プレビュー）

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 Citrix Virtual Channel ソフトウェア開発キット (SDK) (プレビュー)

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のドキュメント」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。適切なバージョンのCitrix Virtual Apps and Desktops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を選択すると、スクリプトのプロンプトに従って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、SUSE 12.5のLinuxストリーミングサポート

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。

MATEデスクトップのキーボードレイアウト同期サポート

MATEデスクトップへのキーボードレイアウト同期のサポートを拡張しました。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追従します。詳細については、「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で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**
- **遅延 - 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**

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、次のメトリックを追加しました。

- **ログオン期間**

  ログオン期間とは、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したものです。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回数**

  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回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**ポリシーが有効になっている場合に有効になります。セッションの再接続について詳しくは、「セッション」を参照してください。ポリシーについて詳しくは、「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。

詳しくは、「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、次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。

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

Linux VDAソースコードファイルcitrix-linux-vda-sources.zipに`ctxusbcfg.sh`を提供するようになりました。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にログオンする際の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、以下の新機能と機能強化が含まれています。

Linux VDAのセルフアップデート

この機能は、Linux VDAソフトウェアを即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便利です。詳しくは、「Linux VDAのセルフアップデート」(/ja-jp/linux-virtual-delivery-agent/2106/configuration/linux-vda-self-update.html)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」(/ja-jp/linux-virtual-delivery-agent/2106/installation-overview/use-mcs-to-create-linux-vms.html#use-mcs-to-create-linux-vms-on-nutanix-ahv)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」(/ja-jp/linux-virtual-delivery-agent/2106/installation-overview/use-mcs-to-create-linux-vms.html)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。この種類の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」(/ja-jp/linux-virtual-delivery-agent/2106/configuration/hdx-webcam-video-compression.html)を参照してください。

グラフィック品質スライダー

仮想Linuxセッションで実行されるグラフィックステータスインジケーターツールに、グラフィック品質スライダーが含まれるようになりました。このスライダーは、画質と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」(/ja-jp/linux-virtual-delivery-agent/2106/configuration/configuring-graphics.html#graphics-quality-slider)を参照してください。

CGPセッションの再接続中にSecure Browserにログオンしているユーザーのポリシーを評価するためのサポート

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セッションの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
  - デスクトップからファイルをダウンロード
  - デスクトップにファイルをアップロード

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

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™でのドメインに参加していない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）でドメインに参加していないLinux VDAを作成できるようになりました。

OpenJDK 11が必須に

Linux VDAでは、OpenJDK 11の存在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。`sudo /opt/Citrix/VDA/bin/xdping`を実行して`XDPing`をインストールできます。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、およびdeploymcs.shスクリプトで利用可能な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である場合は、この変数の値をmateに設定する必要があります。詳しくは、「簡易インストール」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記事を参照してください。

RHEL 8、CentOS 8、SUSE 12.5での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および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ユースケース向けの単一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。詳しくは、「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。詳しくは、「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以下のポリシーが追加されました。

- **セッションアイドルタイマー**: アイドルセッションの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**に設定されており、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け取らなかった場合、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の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、以下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で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ました。

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

詳細については、「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が提供され、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（プレビュー）

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。詳細については、「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年の新機能

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。詳細については、「EDT MTU検出」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/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のリリースでは、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。

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チェックするプロセスを自動化するコマンドラインベースのアプリケーションです。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、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。

RHEL 7.9、RHEL 8.3のサポート

RHEL 7.9とRHEL 8.3がサポート対象のディストリビューションとして追加されました。

Ubuntu 20.04 のサポート

このリリースでは、Ubuntu 20.04 が完全にサポートされています。

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。

Google Cloud Platform (GCP) での マシン作成サービス (MCS) のサポート

MCS を使用して、GCP 上に Linux VM を作成できます。

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。

> **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 のバージョン 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。

Ubuntu 20.04 のサポート (プレビュー)

Ubuntu 20.04 で Linux VDA を配布できます。

ブラウザ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でのブラウザコンテンツリダイレクトをサポートします。ブラウザ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。詳細については、「ブラウザ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内容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。詳細については、「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におけるドメイン非参加Linux VDAのサポート

これで、Citrix Virtual Apps and Desktops Standard for Azureでドメイン非参加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。詳細については、「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。詳細については、「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/2402-ltsr/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 はイタリア語で利用できるようになりました。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 はサポート対象のディストリビューションとして SUSE 12 SP5 を追加しました。

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と貼り付けのサポート

Citrix Virtual Apps and Desktops 2006 および Windows 用 Citrix Workspace アプリ 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アンドペーストできるようになりました。コピーアンドペーストは、右クリックメニューまたはキーボードショートカットを使用して機能します。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 のバージョン 2003 には、次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 はサポート対象のディストリビューションとして RHEL 8 を追加しました。

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。詳細については、「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」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/configure/administration/hdx-insight.html)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 用および 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 上にディレクトリが提供されます。 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、「ファイル転送」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/configure/file/file-transfer.html)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。

Linux VDAテクニカルプレビュー機能のライフサイクル

この記事では、テクニカルプレビュープログラムを通じて導入されたLinux VDA機能を追跡します。これには、アクティブなテクニカルプレビュー機能と、一般提供（GA）に移行した機能の両方が含まれます。各機能について、表にはテクニカルプレビューとして初めて利用可能になったバージョンと、該当する場合はGAに達したバージョンが示されています。

テクニカルプレビューの機能一覧

次の表に、テクニカルプレビューの機能、その説明、およびそれらが初めて利用可能になったバージョンを示します。

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
信頼性の高いトランスポートのためのオーディオ品質エンハンサー	信頼性の高いトランスポートプロトコル（TCPおよびEDT-Reliable）を使用する接続にオーディオ品質の強化を拡張し、困難なネットワーク環境でのオーディオパフォーマンスを向上させます。オーディオ再生と録音の両方で、ネットワーク状態が悪い場合でも遅延が少なく、よりスムーズなエクスペリエンスが得られます。テクニカルプレビューではデフォルトで無効になっています。アダプティブオーディオ	2511	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	が有効になっている必要があり、VDAとCitrix Workspaceアプリの両方で有効にする必要があります。詳細については、信頼性の高いトランスポートのためのオーディオ品質エンハンサー（プレビュー）を参照してください。		
VDAアップグレードの簡素化	以前は、VDAのアップグレードには完全な手動介入が必要でした。バージョン2503ではVDAアップグレードエージェントが導入され、2503以降のアップグレードを共有またはローカルのファイルパスから直接実行できるようになり、以前の複雑な手動介入の要件がなくなりました。詳細については、VDAアップグレードを参照してください。	2503	該当なし
オンプレミス環境に拡張されたトークンベースの登録	簡易インストールを使用してドメイン非参加VDAを作成する場合、クラウドとオンプレミスの両方のデリバリーコントローラーのマシンカタログにドメイン非	2407	2507

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	参加VDAを登録できます。Citrix Web Studioを使用して、空のマシンカタログを作成し、登録トークンを生成できます。詳細については、簡易インストールを使用してドメイン非参加Linux VDAを作成するを参照してください。		
オーディオのロス許容モード	Enlightened Data Transport (EDT) ロス許容プロトコルを介したオーディオがサポートされています。この機能は、ユーザーが高遅延およびパケット損失のあるネットワークを介して接続している場合のリアルタイムストリーミングのユーザーエクスペリエンスを向上させます。この機能が有効になっている場合、Citrix Virtual Apps and DesktopsのAdaptive Transportは、より良いオーディオエクスペリエンスのためにEDTロス許容トランスポートプロトコルを使用します。この機能はデフォルトで無効になっています。詳細	2407	2411

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	については、オーディオのロス許容モードをオーディオ機能の記事で参照してください。		
アダプティブオーディオ用オーディオ品質エンハンサー	バージョン2411以降、アダプティブオーディオ用のオーディオ品質エンハンサーをプレビュー機能として導入しました。この機能強化は、以前のサンプルからオーディオをインテリジェントに再構築することで、短期間のパケット損失や中断を効果的に管理し、品質の著しい劣化を防ぎます。さらに、必要な場合にのみ失われたオーディオパケットを適応的に回復します。オーディオ品質エンハンサーは、パケット損失の持続的な変化に基づいて自動的に有効/無効になり、良好なネットワーク状態と不良なネットワーク状態の両方でオーディオ再生と録音の品質を最適化します。詳細については、オーディオ機能を参照してください。	2411	2507

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
複数のオーディオデバイスのサポート	この機能により、Citrix Workspaceアプリがインストールされているクライアントマシン上の複数のオーディオデバイスをリモートLinux VDAセッションにリダイレクトできます。この機能が有効になっている場合、クライアントマシン上のすべてのローカルオーディオデバイスがセッションに表示されます。CitrixAudioSink（オーディオ出力）またはCitrixAudioSource（オーディオ入力）の代わりに、オーディオデバイスはそれぞれのデバイス名で表示されます。セッション内のアプリでオーディオデバイスを選択したり、セッション中にクライアントマシンのデフォルトオーディオデバイスでもあるデフォルトオーディオデバイスを使用したりできます。必要に応じて、クライアントマシンのシステム設定からデフォルトオーディオデバイスを変更できます。ク	2311	2411

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	ライアントマシンのデフォルトオーディオデバイスが更新されると、新しいデバイスがセッションのデフォルトオーディオデバイスとして表示されます。また、オーディオデバイスは、プラグインまたは取り外し時にセッション内で動的に更新されます。詳細については、複数のオーディオデバイスのサポートを参照してください。		
トークンベースの登録	この機能により、トークンファイルを使用して、ドメインに参加していないVDAをマシンカタログに登録し、これらのVDAをCitrix Cloudコントロールプレーンに対して認証できます。トークンベースの登録は、Citrix以外のプロビジョニングテクノロジーを使用して、マシン（物理または仮想を問わず）を独自に準備するユースケースに最適です。これにより、Cloud Connectorのインストールと保守の必要がなくなり、ユー	2311	2507

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	ザーおよびマシン認証におけるADへの依存が解消され、ドメインに参加していないマシンの認証が可能になります。詳しくは、「Create a non-domain-joined Linux VDA using easy install」を参照してください。		
セキュアな HDX	Citrix Workspaceアプリ（クライアント）とVDA（セッションホスト）間でICAセッションをエンドツーエンドで暗号化できます。エンドツーエンド暗号化（E2EE）機能により、Citrix Gatewayを含む中間ネットワーク要素がICAトラフィックを復号化することはできません。これにより、環境のセキュリティ体制が向上し、構成と管理が容易になります。詳しくは、「Secure HDX」を参照してください。	2311	2503
EDT輻輳制御の強化	Enlightened Data Transport（EDT）プロトコルを最適化するために、新しい輻輳制御アルゴリズムが導入されました。	2308	2311

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	この実装により、EDTはより高いスループットを実現し、遅延を削減して、ユーザーエクスペリエンスを向上させることができます。この機能はデフォルトで無効になっています。詳しくは、「Adaptive transport」を参照してください。		
Fast Identity Online（FIDO2）認証のサポート	Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするために、FIDO2認証を設定できるようになりました。詳しくは、「FIDO2 (preview)」を参照してください。	2305	該当なし
Linuxセッションのローカルストレージへの記録のサポート	Linux VDAでホストされているセッションを記録および再生できます。詳しくは、「Record session to local storage」を参照してください。	2212	2507
データベースオプション	PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後に `/etc/xdl/`	2212	2305

テクニカルプレビュー機能	説明	提供バージョン	一般提供（GA）バージョン
	`db.conf` を編集することで、使用するSQLiteまたはPostgreSQLを指定できます。詳しくは、インストールに関する記事を参照してください。		
Amazon Linux 2のサポート	Amazon Linux 2がサポート対象のディストリビューションとして追加されました。詳しくは、「System requirements」を参照してください。	2112	2203
HDX画面共有	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 screen sharing」を参照してください。	2109	2112
Ubuntu 20.04のサポート	Linux VDA を Ubuntu 20.04 にインストールできます。	2009	2012

システム要件

Linux Virtual Delivery Agent (VDA) の現在のリリースは、Citrix Virtual Apps and Desktops に準拠しています。また、ライフサイクルの終了に達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、および Citrix が特定のバージョンの製品のサポートをいつ終了するかについては、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 および 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、およびその他の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 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 バージョン、およびデスクトップ環境の表については、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バージョン	サポートされているデスクトップ
デビアン 13.6	1.20	グノーム、グノーム クラシック、マテ、エックスエフシーイー
デビアン 12.15	1.20	グノーム、グノーム クラシック、ケーディーイー、マテ、エックスエフシーイー
RHEL 10.2	(Wayland、Xorgなし)	GNOME

LINUXディストリビューション	XORGバージョン	サポートされているデスクトップ
RHEL 9.8/9.6	1.20	グノーム、マテ、Xfce
RHEL 8.10	1.20	グノーム、グノーム クラシック、マテ、Xfce
ロッキーリナックス 10.2	(Wayland、Xorgなし)	GNOME
ロッキーリナックス 9.8/9.6	1.20	グノーム、マテ、Xfce
ロッキーリナックス 8.10	1.20	グノーム、グノームクラシック、KDE、MATE、Xfce
SUSE 15.7	1.20	グノーム、グノームクラシック、MATE
ウブントゥ 26.04	(Wayland、Xorgなし)	GNOME
ウブントゥ 24.04	1.21	グノーム、グノームクラシック、KDE、MATE、Xfce
ウブントゥ 22.04	1.21	グノーム、グノームクラシック、KDE、MATE、Xfce

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デスクトップ環境は、ctxinstall.shまたはctxsetup.shスクリプトを使用して指定できます。
- Red Hat Enterprise Linuxのドキュメントによると、RHEL 9で利用可能なデスクトップ環境はGNOMEのみです。ただし、EPELリポジトリがインストールされている場合は、RHEL 9およびRocky Linux 9でXfceデスクトップ環境も使用できます。

- RHEL 10、Rocky Linux 10、およびUbuntu 26.04でWaylandによって導入された制限を参照してください。
- Ubuntuでは`HWE kernel`または`HWE Xorg`を使用しないでください。
- 場合によっては、カーネルへのサードパーティによる変更により、Linux VDAに必要なモジュールが不足することがあります。この場合、必要なカーネルモジュールをビルドしてインストールする必要があります。新しいカーネルモジュールをインストールする際は、セキュアブートが無効になっていることを確認してください。
- 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。

.Netの要件

.NET Runtimeに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションにASP.NET Core Runtimeをインストールする必要があります。バージョン10が必要です。

Linux VDAはより高いメジャーバージョンの.NETでも動作できますが、互換性は保証されません。より高いメジャーバージョンで実行すると、リスクが生じる可能性があります。詳細については、How the .NET Runtime and SDK are versioned - .NETを参照してください。

お使いのLinuxディストリビューションに必要な.NETバージョンが含まれている場合は、組み込みのフィードからインストールしてください。そうでない場合は、Microsoftパッケージフィードから.NETをインストールしてください。詳細については、Install .NET on Linux distributions - .NET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

- アマゾン ウェブ サービス (AWS EC2およびアマゾン ワークスペース コア)
- グーグル クラウド プラットフォーム (GCP)
- マイクロソフト アジュール

- ベアメタルサーバー
- シトリックス XenServer (旧称 シトリックス Hypervisor™)
- カーネルベース仮想マシン (KVM)
- マイクロソフト Hyper-V
- ニュータニックス AHV
- レッドハット オープンシフト
- ヴイエムウェア vSphere Hypervisor

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。

MCSでサポートされているハイパーバイザーのリストを参照してください

Active Directory統合パッケージ

Linux VDAは、以下のActive Directory統合パッケージおよび製品をサポートしています。

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
デビアン 13.6/12.15	はい	はい	はい	はい	はい
RHEL 10.2、ロッキー Linux 10.2	はい	はい	はい	いいえ	はい（Quest v5.1以降）
アールイーエイチエル 9.8/9.6、ロッキーリナックス 9.8/9.6/8.10	はい	はい	はい	いいえ	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 8.10	はい	はい	はい	はい	はい（Quest v4.1以降）
SUSE 15.7	はい	はい	はい	はい	はい
ウブンツ 26.04	はい	はい	はい	いいえ	はい（Quest v5.0.7以降）
ウブンツ 24.04	はい	はい	はい	いいえ	はい（Quest v4.1以降）
ウブントゥ 22.04	はい	はい	はい	はい	はい（Quest v4.1以降）

Cloud Connectorのサイズとスケーリングに関する考慮事項

Linux VDAをCitrix Cloud™ Connectorでコントロールプレーンに接続する場合、Citrixの社内テストに基づき、以下の点を考慮してください。

- 各Citrix Cloud Connector™（4 vCPU、10 GBメモリ）は、6,000台のLinux VDAをサポートできます。
- 各リソースの場所に2つのCloud Connectorを展開して高可用性を確保し、各リソースの場所に最大6,000台のLinux VDAを展開します。

データベースに関する考慮事項

- VDIモードではSQLiteのみを使用し、ホスト型共有デスクトップ配信モデルではPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
- 簡単なインストールとMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。

- 手動インストールの場合、SQLite、PostgreSQL、またはその両方、あるいは組み込みレジストリサービス関連の依存関係を手動でインストールする必要があります。SQLiteとPostgreSQLの両方をインストールする場合は、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。
- 組み込みレジストリを使用している場合、PostgreSQLまたはSQLiteをインストールする必要はありません。以前のリリースからアップグレードされたLinux VDAの場合、db.confは古いものが保持されます。db.confは、新規インストールの場合にのみ新しいバージョンに更新されます。
- Linux VDA 2607 LTSR以降にアップグレードした後、PostgreSQLまたはSQLiteの代わりに組み込みレジストリサービスを使用する場合は、`/opt/Citrix/VDA/sbin/ctxregd_migrate.sh`を実行して組み込みレジストリサービスに移行できます。移行が可能かどうかを確認するために、移行前に`/opt/Citrix/VDA/sbin/ctxregd_migrate.sh -c`を実行することをお勧めします。

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可能にします。良好な3Dグラフィックエクスペリエンスを確保するために、ネットワーク帯域幅、遅延、およびその他の関連するインフラストラクチャ条件を慎重に検討することをお勧めします。たとえば、可能な限りVDAとCitrix Workspaceアプリを同じリージョンに展開します。

ハイパーバイザー

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

- XenServer®（旧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
- アマゾン ウェブ サービス (AWS)
- Google Cloud プラットフォーム (GCP)

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

GPU

Linux VDAの場合、HDX 3D Proは次の種類のGPUをサポートしています。

NVIDIA vGPU

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カードについては、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**、および**Guest OS Support**の各列を確認してください。

GPUカード用の最新のvGPUドライバーがインストールされていることを確認してください。現在、Linux VDAはvGPU 19までをサポートしています。詳しくは、NVIDIA Virtual GPU Software Supported GPUsを参照してください。

非仮想化GPU

Linux VDAドキュメントでは、非仮想化GPUは以下を指します。

- Remote PC Accessシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

NVIDIA Capture SDK for Linuxをサポートする NVIDIA GPU

NVIDIA Capture SDK for Linuxをサポートする NVIDIA GPUの場合、Linux VDAのインストール時に**CTX_XDL_HDX_3D_PRO**を**Y**に設定することでHDX 3D Proを有効にします。追加の設定は不要です。HDX 3D Proを有効にすると、ハードウェアアクセラレーションはデフォルトで有効になります。

> **ヒント:**
>
> NVIDIA GPUでHDX 3D Proを使用するには、NVIDIA Capture SDKバージョン8をサポートするNVIDIAグラフィックドライバーバージョンをインストールする必要があります。詳しくは、NVIDIA Capture SDKのドキュメントを参照してください。

RHEL 10、Rocky Linux 10、Ubuntu 26.04のWaylandによって導入された制限

- インストール後、rootユーザーで/opt/Citrix/VDA/sbin/gnome_extension_just_perfection_helper.shを実行して、GNOMEトップバーの右上隅にある画面キャストアイコンを非表示にします。

- HDX 3D Proはサポートされていません。
- Linux仮想アプリセッションモードはサポートされていません。
- GUIベースのeasyinstallはサポートされていません
- PVSプロビジョニングは完全にサポートされていません
- マルチモニターシナリオにおける既知の問題：クライアントデバイスの画面配置は、VDAで使用されるすべてのモニターが連続していることを保証する必要があります。そうでない場合、マルチモニター機能は利用できません。

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。

- インストール準備
- 簡易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
- MCSを使用してドメイン非参加のLinux VDAを作成する
- 簡易インストールを使用してドメイン非参加のLinux VDAを作成する（プレビュー）
- MCSを使用してLinux VDAを作成する
- MCSを使用してFASが有効なドメイン参加済みLinux VDAを作成する
- Citrix Provisioningを使用してLinux VDAを作成する
- Citrix DaaS Standard for AzureでLinux VDAを作成する
- Linux VDAを手動でインストールする
    - 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

インストール準備

この記事では、特定の状況に基づいて適切なインストール方法を選択する方法を提示し、関連するインストールガイドに誘導します。Citrix Virtual Apps and DesktopsおよびCitrix DaaSの展開の計画については、(/ja-jp/citrix-virtual-apps-desktops/install-configure/install-prepare#installation-guidance-considerations-and-best-practice)と(/ja-jp/citrix-daas/install-configure)をそれぞれ参照してください。

VDAのインストールを開始する前に、以下に注意してください。

- (/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html)を確認して、ターゲットの展開が公式にサポートされていることを確認してください。
- VDAをドメインに接続するかどうかを決定します。

- プロビジョニング方法とドメイン参加の決定に基づいて、次の表から適切なリンクを確認してください。

	簡易インストールスクリプト	マシンクリエーションサービス™ (MCS)	シトリックス プロビジョニング	完全手動
ドメイン参加済み	簡易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	Machine Creation Services (MCS) を使用してLinux VDAを作成する	Citrix Provisioning を使用してLinux VDAを作成する	Linux VDAを手動でインストールする
ドメイン非参加	簡易インストールを使用して非ドメイン参加Linux VDAを作成する	MCSを使用して非ドメイン参加Linux VDAを作成する	サポートされていません	サポートされていません

注:

- MCSは、リモートPCアクセスユースケースではサポートされていません。
- 大量プロビジョニングの場合、MCS、Citrix Provisioning™、または独自の取り組みによるサードパーティ製自動化を使用できます。サードパーティ製自動化の場合、簡単インストールスクリプト（推奨）を組み込むか、スクリプト内で純粋な手動インストール手順を自動化するオプションがあります。
- 単一のVMでの迅速な概念実証（POC）には、簡単インストールを使用することをお勧めします。
- 簡単インストールスクリプト（ctxinstall.sh -s）を実行して、CTX_XDL_DDC_LISTやCTX_XDL_LDAP_LISTなどの環境変数を迅速に更新することもできます。
- デプロイ管理にAnsibleの使用に関心のあるお客様向けに、包括的なベストプラクティス記事を提供しています。詳細については、「Ansibleを使用したデプロイの管理」を参照してください。

easy install を使用してドメイン参加済みVDAを作成する

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、easy install 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。easy install 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くなります。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) と easy install の両方を使用できます。詳細については、「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」および「easy install を使用してドメインに参加していない Linux VDA を作成する」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マシンの準備

easy install 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)。
- Active Directory (AD) 統合方法 - 現在、easy install は SSSD、Winbind、Centrify、PBIS、および Quest をサポートしています。easy install は、RHEL および Rocky Linux のみで Quest をサポートしています。
- ユーザー名 – マシンをドメインに参加させるユーザーの名前。
- パスワード - マシンをドメインに参加させるユーザーのパスワード。
- OU – 組織単位。オプション。

重要:

- Linux VDAをインストールするには、リポジトリがLinuxマシンに正しく追加されていることを確認してください。
- セッションを起動するには、X Windowシステムとデスクトップ環境がインストールされていることを確認してください。
- セキュリティのため、easy installはドメイン参加パスワードを保存しません。対話モードでeasy installスクリプト (ctxinstall.sh) を実行するたびに、ドメイン参加パスワードを手動で入力する必要があります。サイレントモードでは、ドメイン参加パスワードを */Citrix/VDA/sbin/ctxinstall.conf* で設定するか、パスワードをエクスポートする必要があります。ドメイン参加に管理者アカウントを使用しないことをお勧めします。代わりに、管理者アカウント以外のActive Directoryユーザーにドメイン参加権限を委任してください。そのためには、**Delegation of Control Wizard** を使用して、ドメインコントローラーで制御を委任します。

ステップ2：ハイパーバイザーの準備

サポートされているハイパーバイザー上でLinux VDAを仮想マシンとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに基づいて、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。

XenServer (旧Citrix Hypervisor™) での時刻同期の修正

XenServer®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XenServer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。

XenServer 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XenServerの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration ServicesがインストールされているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するためには、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の時刻同期が無効になっているVMwareやXenServer（旧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、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. 「**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: .NETのインストール

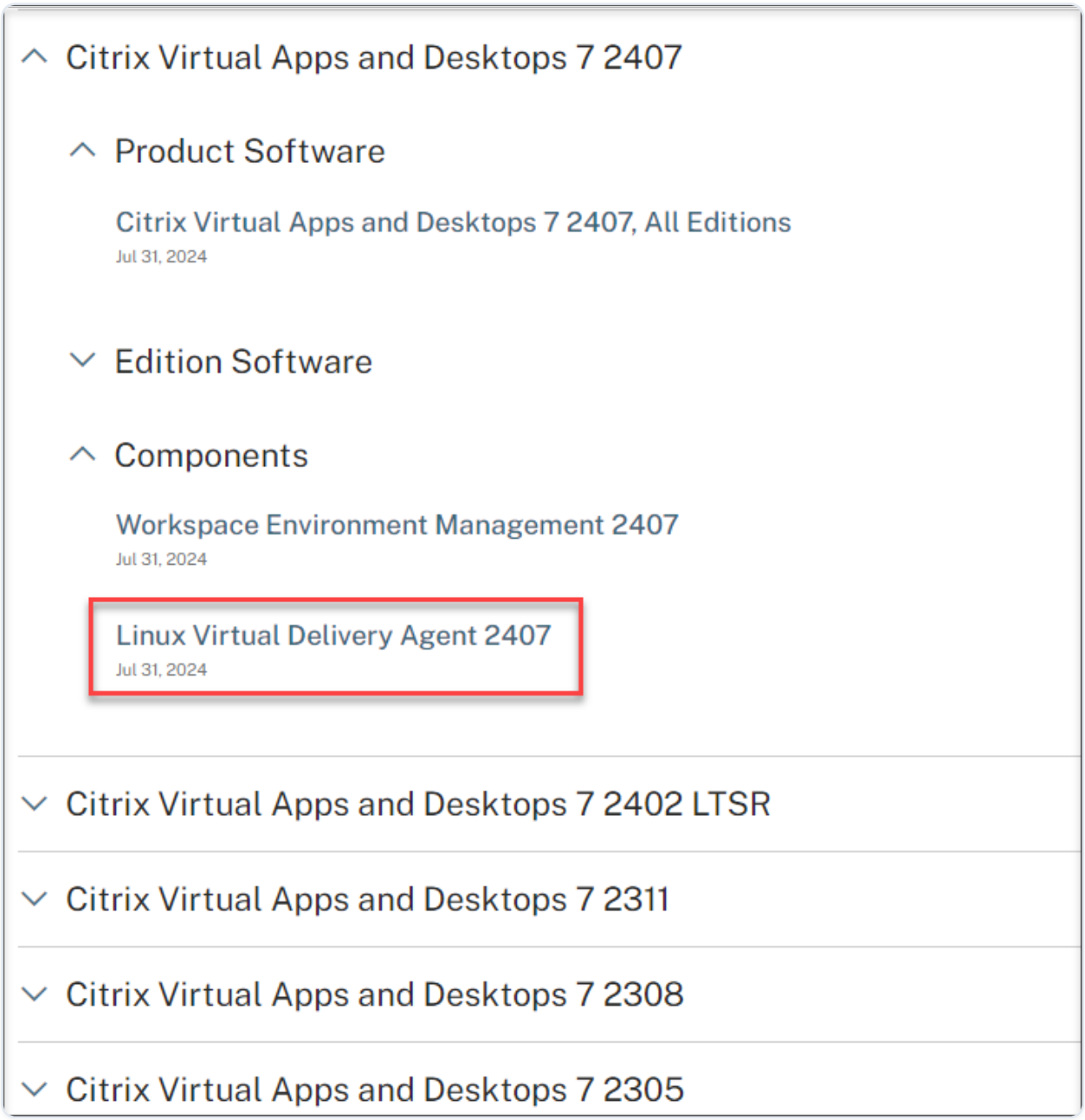
Linux VDAパッケージをインストールする前に、システム要件に従って .NETをインストールします。

.NET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。

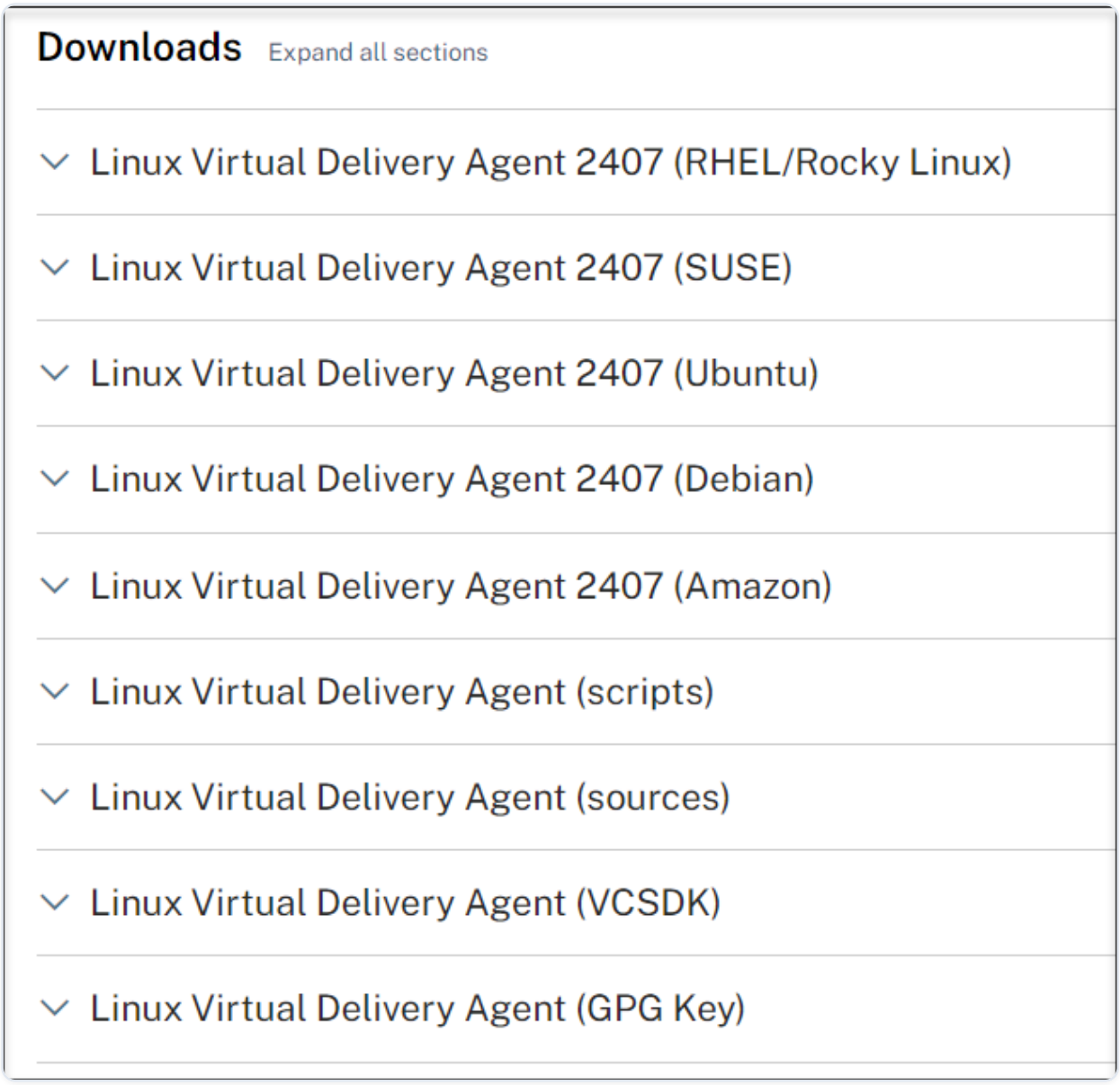
コマンドの出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンドの出力が /aa/bb/dotnet の場合、/aa/bb を .NETバイナリパスとして使用します。

ステップ4: Linux VDAパッケージのダウンロード

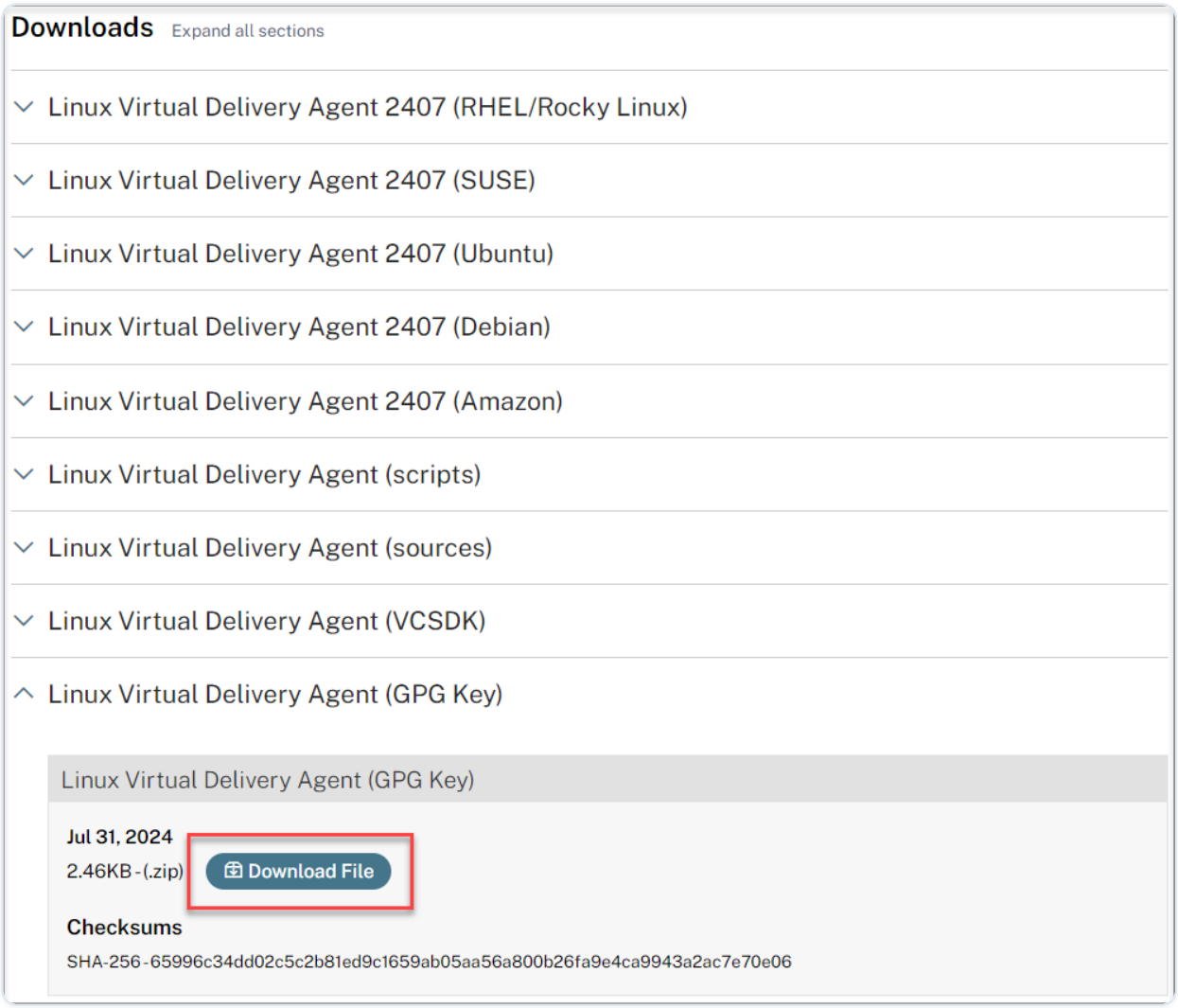
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. 「**Components**」を展開してLinux VDAを見つけます。例:



4. Linux VDAリンクをクリックして、Linux VDAダウンロードにアクセスします。



5. 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。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開キーをダウンロードします。例:



公開キーを使用してLinux VDAパッケージの整合性を検証するには、次のようにします。

- RPMパッケージの場合、公開キーをRPMデータベースにインポートし、パッケージの整合性を確認するには、次のコマンドを実行します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

- DEBパッケージの場合、公開キーをDEBデータベースにインポートし、パッケージの整合性を確認するには、次のコマンドを実行します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

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およびRocky Linuxディストリビューションの場合：

注:

- RHELの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。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。
- RHEL 9.xおよびRocky Linux 9.xにLinux VDAをインストールする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

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

注:

GCPでホストされているサポート対象のRHELまたはRocky Linuxディストリビューションに 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ことがあります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動した後にコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

Ubuntu/Debianディストリビューションの場合：

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

sudo apt-get install -f
```

> **注:**
>
> - Debianディストリビューションに必要な依存関係をインストールするには、`/etc/apt/sources.list`で対応するDebianリポジトリを構成します。たとえば、Debian 13の場合は、次を追加します: `deb http://deb.debian.org/debian/ trixie main`
> - GCP上のUbuntuの場合、RDNSを無効にします。そのためには、/etc/krb5.confの**の下にrdns = false**の行を追加します。

SUSEディストリビューションの場合：

1. AWS、Azure、およびGCP上のSUSE 15.7の場合、以下を確認してください：
   - **libstdc++6** バージョン12以降を使用していること。
   - **/etc/sysconfig/windowmanager** の **Default_WM** パラメーターが **"gnome"** に設定されていること。
2. Linux VDAをインストールするには、次のコマンドを実行します。

   ```
   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。

- XenServer
- 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（クラウドベンダーまたはNVIDIAから提供）をインストールします。

ステップ7：使用するデータベースを指定する

Linux VDAパッケージのインストール後、SQLite、PostgreSQL、および組み込みレジストリサービスを切り替えることができます。これを行うには、次の手順を完了します。

> **注:**
>
> - VDIモードではSQLiteを、ホスト型共有デスクトップ配信モデルではPostgreSQLを使用することをお勧めします。
> - 簡単インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションによって提供されるバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡単インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。
> - また、**/etc/xdl/db.conf**を使用してPostgreSQLのポート番号を構成することもできます。

1. (オプション) Linuxディストリビューションによって提供されるバージョンではなく、カスタムバージョンのPostgreSQLを使用するには、指定されたバージョンを手動でインストールし、PostgreSQLサービスを開始します。
2. **/etc/xdl/db.conf**を編集して、使用するデータベースを指定します。以下は**db.conf**ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite
```

```
#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false
```

```
## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL**をtrueに設定します。

3. **sudo /opt/Citrix/VDA/sbin/ctxinstall.sh** または **/opt/Citrix/VDA/bin/easyinstall** を実行します。

ステップ8：簡単インストールを実行して環境とVDAを構成し、インストールを完了する

Linux VDAパッケージのインストール後、ctxinstall.shスクリプトまたはGUIを使用して実行環境を構成します。

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。

ctxinstall.sh

ctxinstall.shは、いくつかの事前構成とVDA実行環境変数の設定を行うための簡単インストールスクリプトです。

- このスクリプトはrootのみが実行できます。
- 簡単インストールでは、/opt/Citrix/VDA/sbin/ctxinstall.confを構成ファイルとして使用し、使用されるすべての環境変数の値を設定、保存、同期します。テンプレート (ctxinstall.conf.tmpl) を注意深く読み、独自のctxinstall.confをカスタマイズすることをお勧めします。構成ファイルを初めて作成するときは、次のいずれかの方法を使用します。

    - /opt/Citrix/VDA/sbin/ctxinstall.conf.tmplテンプレートファイルをコピーし、/opt/Citrix/VDA/sbin/ctxinstall.confとして保存します。
    - ctxinstall.shを実行します。ctxinstall.shを実行するたびに、入力は/opt/Citrix/VDA/sbin/ctxinstall.confに保存されます。

- Easy installはモジュール実行をサポートしています。モジュールには、事前チェック、インストール、ドメイン構成、セットアップ、検証が含まれます。
- このスクリプトのデバッグの詳細は、/var/log/xdl/ctxinstall.logにあります。

詳細については、ヘルプコマンド **ctxinstall.sh -h** を使用してください。

> **注:**
>
> - 最小特権の原則に従い、ドメイン参加パスワードがファイルに設定されている可能性があるため、rootユーザーのみが **/opt/Citrix/VDA/sbin/ctxinstall.conf** を読み取れるようにしてください。
> - Linux VDAをアンインストールすると、**/opt/Citrix/VDA** 以下のファイルが削除されます。VDAをアンインストールする前に、**/opt/Citrix/VDA/sbin/ctxinstall.conf** をバックアップすることをお勧めします。

ctxinstall.shは、対話モードまたはサイレントモードで実行できます。スクリプトを実行する前に、以下の環境変数を設定してください。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は 'n' です。ドメイン参加シナリオでは、'n' に設定します。
- **CTX_XDL_AD_INTEGRATION='sssd\|winbind\|centrify\|pbis\|quest'** – 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。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerへの登録に使用する、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_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル (VDI) または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どうか。HDX 3D Pro環境の場合は、値を **'y'** に設定します。

- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ (シングルセッション) モード (つまり、CTX_XDL_VDI_MODE=**'y'**) 用に構成されます。
- **CTX_XDL_START_SERVICE='y\|n'** – 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。
- **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 VDAサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktopのシステムファイアウォールで、必要なポート (デフォルトではポート80および1494) を自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または **Xfce** デスクトップ環境を指定します。**'<none\>'** に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。
- **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 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_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_SITE_NAME=<dns-name\>** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、**'<none\>'** に設定します。
- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDA は DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP ポートを持つ 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 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directory フォレスト内で LDAP クエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連する LDAP ポート番号を 3268 として指定します。この変数はデフォルトで **'<none\>'** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP を照会します。検索パフォーマンスを向上させるには、検索ベース (例:

OU=VDI,DC=mycompany,DC=com) を指定できます。不要な場合は、**'<none\>'** に設定します。

- **CTX_XDL_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_EASYINSTALL_DNS='<ip-address-of-dns\>'** – DNS の IP アドレス。
- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDA サーバーのホスト名。
- **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_REALM=realm-name** – Kerberos レルム名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_USERNAME=domain-user-name** – マシンをドメインに参加させるユーザーの名前。
- **CTX_EASYINSTALL_PASSWORD=password** – マシンをドメインに参加させるユーザーのパスワード。

注:

ドメイン参加には管理者アカウントを使用しないことをお勧めします。代わりに、管理者アカウント以外の Active Directory ユーザーにドメイン参加権限を委任してください。これを行うには、**Delegation of Control Wizard** を使用してドメインコントローラーで制御を委任します。

以下の4つの変数はオプションです。これらが設定されていなくても、ctxinstall.sh はサイレントモードで中断せず、インタラクティブモードでユーザー入力を求められることもありません。これらの変数は、値をエクスポートするか、**/Citrix/VDA/sbin/ctxinstall.conf** を編集することによってのみ設定できます。

- **CTX_EASYINSTALL_NETBIOS_DOMAIN=netbios-domain-name** – NetBIOSドメイン名は通常、ドット(.)で区切られたDNSドメイン名の最初のコンポーネントです。それ以外の場合は、別のNetBIOSドメイン名をカスタマイズします。この変数はオプションです。
- **CTX_EASYINSTALL_OU=ou-value** – OUの値は、さまざまな**AD**統合方法によって異なります。OU値の例については、この記事の考慮事項セクションの表を参照してください。この変数はオプションです。

- **CTX_EASYINSTALL_TRUSTED_DOMAINS=trusted-domains** – マルチドメイン環境の場合、信頼されたドメインのスペース区切りのリストを指定します (例: "mycompany1.com mycompany2.com")。これにより、/etc/krb5.conf の信頼されたドメインが更新され、**CTX_XDL_LDAP_LIST** が指定されていない場合、それらのドメイン内のLDAPサーバーの自動検出が有効になります。この変数はオプションです。

> **注**
>
> SSSDは、単一のActive Directoryフォレスト内の信頼されたドメインのみをサポートします。

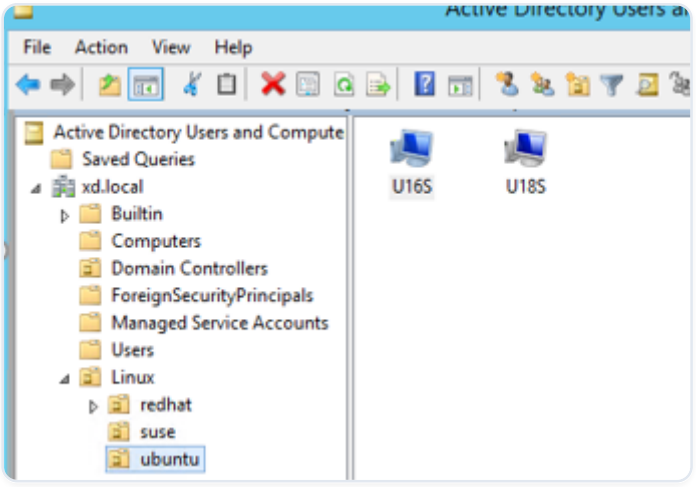
- **CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=centrify-local-path** – Easy installは、インターネットからCentrifyパッケージをダウンロードするのに役立ちます。ただし、Centrifyがすでにインストールされている場合は、この変数で定義されたローカルディレクトリからCentrifyパッケージを取得できます。この変数はオプションです。
- **CTX_EASYINSTALL_PBIS_LOCAL_PATH= pbis-local-path** – Easy installは、インターネットからPBISパッケージをダウンロードするのに役立ちます。ただし、PBISがすでにインストールされている場合は、この変数で定義されたローカルディレクトリからPBISパッケージを取得できます。この変数はオプションです。
- **CTX_EASYINSTALL_QUEST_LOCAL_PATH=quest-local-path** - vastool実行可能ファイルを含むローカルパス。設定されていない場合、デフォルトパス /opt/quest/bin が使用されます。この変数はオプションです。
- **CTX_XDL_DJ_ENROLLMENT_TOKEN_FILE=`'<none>'`** - Delivery ControllerへのWebSocket接続用の登録トークンファイルへのフルパス。この変数はオプションです。
- **CTX_XDL_ENROLLMENT_TOOL_USING_LDAPS='n'** - VdaEnrollmentToolがコンピューターSIDのクエリにLDAPSを使用するかどうかを決定します。LDAPSを使用する場合は、LDAPサーバーがローカルシステム証明書で検証可能であることを確認してください。この変数はオプションです。

考慮事項

- NetBIOSドメイン名は通常、ドット(.)で区切られたDNSドメイン名の最初のコンポーネントです。環境内で別のNetBIOSドメイン名をカスタマイズするには、**/opt/Citrix/VDA/sbin/ctxinstall.conf** で環境変数 **CTX_EASYINSTALL_NETBIOS_DOMAIN** を設定します。
- VDAを特定のOUに参加させるには、次の手順を実行します。

  a. 特定のOUがドメインコントローラー上に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. **/opt/Citrix/VDA/sbin/ctxinstall.conf** で環境変数 **CTX_EASYINSTALL_OU** を設定します。

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
デビアン	`"Linux/debian"`	`"Linux/debian"`	`"XD.LOCAL/Linux/debian"`	`"Linux/debian"`
RHEL 9.8/9.6、ロッキー リナックス 9.8/9.6	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	該当なし	該当なし
RHEL 8.x	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	`"XD.LOCAL/Linux/redhat"`	`"Linux/redhat"`
ロッキー Linux 8.x	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	該当なし	該当なし
SUSE	`"Linux/suse"`	`"Linux/suse"`	`"XD.LOCAL/Linux/suse"`	`"Linux/suse"`
ウブントゥ	`"Linux/ubuntu"`	`"Linux/ubuntu"`	`"XD.LOCAL/Linux/ubuntu"`	`"Linux/ubuntu"`

- 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。

- Easy install は、Linux VDA 7.16 以降で純粋な **IPv6** をサポートします。以下の前提条件と制限が適用されます。

  - Linuxリポジトリは、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**IPv4**と**IPv6**のハイブリッド構成である場合、最初のDNS IPアドレスのタイプによって、登録に**IPv4**または**IPv6**のどちらが使用されるかが決まります。

- 前述のように、CTX_XDL_DESKTOP_ENVIRONMENT変数を使用して、セッションで使用するデスクトップ環境を指定できます。コマンドを実行するか、システムトレイを使用することによって、デスクトップ環境を切り替えることもできます。詳しくは、デスクトップ切り替えコマンドおよびシステムトレイを参照してください。

- 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次のとおりです。
  - Easy Installは、Centrifyパッケージをインターネットから自動的にダウンロードするのに役立ちます。各ディストリビューションのURLは次のとおりです。

    RHEL の場合: wget `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-rhel6-x86_64.tgz>`

    スーゼ：wget `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-suse12-x86_64.tgz>`

    Ubuntu/Debian：wget `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-deb9-x86_64.tgz>`
  - Centrifyがすでにインストールされている場合は、ローカルディレクトリからCentrifyパッケージを取得します。Centrifyパッケージのディレクトリを指定するには、**/opt/Citrix/VDA/sbin/ctxinstall.conf**で**CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=/home/mydir**を設定します。例：

```
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次のとおりです。
  - Easy Installは、PBISパッケージをインターネットから自動的にダウンロードするのに役立ちます。たとえば、各ディストリビューションのURLは次のとおりです。

    RHEL 8, SUSE 15.7 の場合: wget `<https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh>`

    Debian, Ubuntu の場合: wget `<https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.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「pbisDownloadRelease」および「pbisDownloadExpectedSHA256」の行を変更します。

    例については、次のスクリーンショットを参照してください。

    PBISパッケージの特定のバージョンをフェッチする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/fetching-specific-version-of-pbis-package.png)

- PBISがすでにインストールされている場合は、ローカルディレクトリからPBISパッケージをフェッチします。PBISパッケージのディレクトリを指定するには、**/opt/Citrix/VDA/sbin/ctxinstall.conf**で環境変数**CTX_EASYINSTALL_PBIS_LOCAL_PATH**を設定します。

対話モード

対話モードで**ctxinstall.sh**スクリプトを実行するには、**-S**オプションなしで**sudo /opt/Citrix/VDA/sbin/ctxinstall.sh**コマンドを使用します。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変数値を入力します。変数がすでに設定されている場合、ctxinstall.shは変更したい場合に確認を求めます。

サイレントモード

サイレントモードでは、**/opt/Citrix/VDA/sbin/ctxinstall.conf**またはexportコマンドを使用して、上記の変数を設定する必要があります。その後、**ctxinstall.sh -S**を実行します（ここで文字**S**は**大文字**であることに注意してください）。必要な変数がすべて設定されていないか、一部の値が無効な場合、デフォルト値がない限り、**ctxinstall.sh**は実行を中止します。

各変数のエクスポートされた値は、設定されていない場合を除き、**/Citrix/VDA/sbin/ctxinstall.conf**の値を上書きします。ドメイン参加パスワードを除くすべての更新された値は、**/Citrix/VDA/sbin/ctxinstall.conf**に保存されます。したがって、サイレントモードでは、**/Citrix/VDA/sbin/ctxinstall.conf**でドメイン参加パスワードを設定するか、パスワードをエクスポートする必要があります。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'
```

```
export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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_SUPPORT_DDC_AS_CNAME='y|n'

export CTX_EASYINSTALL_DNS='<ip-address-of-dns>'

export CTX_EASYINSTALL_HOSTNAME='<host-name>'

export CTX_EASYINSTALL_NTPS='<address-of-ntps>'

export CTX_EASYINSTALL_REALM='<realm-name>'

export CTX_EASYINSTALL_FQDN='<ad-fqdn-name>'

export CTX_EASYINSTALL_USERNAME='<domain-user-name>'

export CTX_EASYINSTALL_PASSWORD='<password>'

export CTX_EASYINSTALL_NETBIOS_DOMAIN='<netbios-domain>'

export CTX_EASYINSTALL_OU='<organization-unit>'

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

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

または、単一のコマンドを使用してすべての変数を指定することもできます。

VDA実行環境変数（'**CTX_XDL_**'で始まるもの）を設定するには、**ctxinstall.sh -s**を実行します（ここで文字**s**は**小文字**であることに注意してください）。

GUI

Linux VDAパッケージがインストールされると、簡易インストールGUIが自動的に表示されます（ディスプレイが利用可能な場合）。その後、簡易インストールGUIの指示に従ってください。

または、Linux VDAパッケージがインストールされた後であればいつでも、VDAのデスクトップ環境で**/opt/Citrix/VDA/sbin/linux-vda-installer**コマンドを実行できます。 簡易インストールGUI(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/easy-install-gui-1.png)

簡易インストールGUIは、次の操作をガイドします。

- システム環境の確認
- データベースを構成する
- 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
- ランタイム環境を構成する

> **注:**

> マシンがドメインに参加している場合、ユーザーは「Already joined to a domain」をクリックできます。そうでない場合は、マシンをドメインに参加させるために、ドメイン情報と管理者資格情報を提供する必要があります。

> ユーザーは「Advanced Configuration (optional)」ボタンをクリックして、詳細な構成オプションを展開することもできます。

> **ヒント:**

> **Save** をクリックして、指定したパスのローカルファイルに変数の設定を保存します。**Load** をクリックして、指定したファイルから変数の設定を読み込みます。

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

Linux VDA を開始する:

Linux VDA サービスを開始するには:

```
sudo systemctl start ctxhdx.service



sudo systemctl start ctxvda.service
```

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo systemctl stop ctxvda.service



sudo systemctl stop ctxhdx.service
```

注:

ctxvda および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** 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.service



sudo systemctl restart ctxhdx.service
```

```
sudo systemctl start ctxvda.service
```

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.service



sudo systemctl status ctxhdx.service
```

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ついては、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホストされた共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに1人のユーザーという配信モデルが意味されます。

ヒント:

> マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除し、再度追加する必要があります。

ステップ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 2503」を参照してください。

ステップ13：Linux VDAのアップグレード（オプション）

Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311へ、および1912 LTSRから2203 LTSRへアップグレードできます。

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

- 2407リリース以降、Linux VDAは、アップグレード中に構成ファイルを処理するためにパッケージマネージャーの**rpm**または**dpkg**に委任します。以下に、rpmとdpkgが構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**：デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを**.rpmnew**拡張子で保存します。
  - **dpkg**：対話形式で、続行方法の選択肢を提示します。ローカル構成ファイルを保持し、新しいパッケージバージョンを**.dpkg-new**または**.dpkg-dist**として保存しながら、Linux VDAをサイレントアップグレードするには、次のコマンドを使用します。

    ```
    dpkg --force-confold -i package.deb  # Always keep your version, then save new package's version as *.dpkg-new or *.dpkg-dist
    ```

RHELおよび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:

RHEL 9.xおよびRocky Linux 9.xでLinux VDAをアップグレードする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

SUSEディストリビューションの場合：

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

Ubuntu/Debianディストリビューションの場合：

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



sudo apt-get install -f
```

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から生じる可能性のある問題をトラブルシューティングします。

SSSD を使用したドメイン参加に失敗する

ドメインに参加しようとするとエラーが発生する場合があります。出力は次のようになります（画面出力についてはログを確認してください）。

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

/var/log/xdl/vda.log:

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

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

2016-11-04 02:11:52.362 [ERROR] - The Citrix Desktop Service cannot
connect to the delivery controller 'http://CTXDDC.citrixlab.local:80/
Citrix/CdsController/IRegistrar' (IP Address '10.158.139.214')

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

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

-  Kerberos is correctly configured on this machine.

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

Error Details:

Exception 'General security error (An error occurred in trying to obtain a
```

```
TGT: Client not found in Kerberos database (6))' of type 'class
javax.xml.ws.soap.SOAPFaultException'.

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

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

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

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

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

/var/log/messages:

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

この問題を解決するには:

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

Ubuntu デスクトップセッションでグレースクリーンが表示される

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

この問題を解決するには:

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

   `greeter-show-manual-login=true`

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

/var/log/xdl/hdx.log:

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



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



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



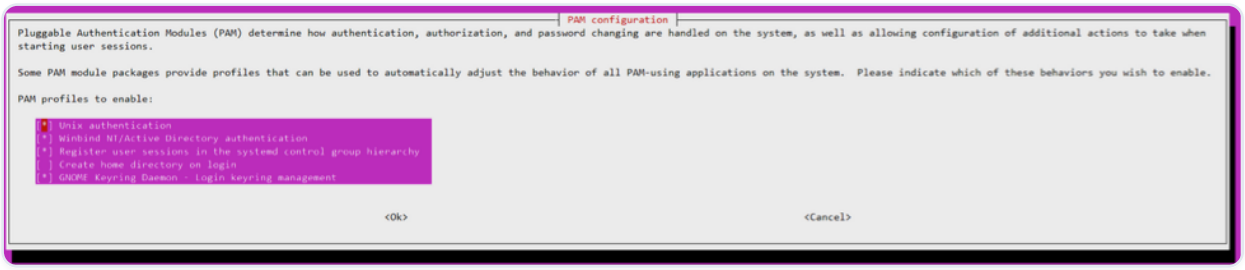
2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting normally.
```

ヒント:

この問題の根本原因は、ドメイン管理者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エラーで迅速に終了する

/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/xdl/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. SELinux を無効にするには、/etc/selinux/config に次の変更を加えます。

   SELINUX=無効

2. VDA を再起動します。

MCSを使用してドメイン非参加のLinux VDAを作成する

この記事では、Machine Creation Services (MCS) メソッドを使用して、Citrix DaaSまたはCitrix Virtual Apps and Desktops™ 2411以降でマシンカタログを持つドメイン非参加のLinux VDAを作成する方法について説明します。

重要:

- Citrix DaaS™のお客様の場合：
  - ドメイン非参加の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。ドメイン非参加の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。
  - ドメイン非参加のVDAを作成するには、Citrix Gatewayサービスを使用しているお客様は、Rendezvous V2が有効になっていることを確認する必要があります。Cloud Connectorは、オンプレミスのハイパーバイザーにマシンをプロビジョニングする場合、またはWorkspaceでActive DirectoryをIDプロバイダーとして使用する場合にのみ必要です。
- CVADのお客様の場合：
  - 以下の手順に従って、DDCでWebSocket機能を有効にします。

    PowerShellを開いて以下のコマンドを実行し、DDCを再起動します。New-ItemProperty `"HKLM:\SOFTWARE\Citrix\DesktopServer\WorkerProxy" -Name "WebSocket_Enabled" -PropertyType "DWord" -Value 1 -Force`
- マスターイメージにCA証明書を配置して更新します。自己署名CA証明書の追加を参照してください。
- ドメイン非参加のVDAを作成するには、MCSと簡易インストールの両方を使用できます。詳細については、MCSを使用したドメイン非参加のLinux VDAの作成および簡易インストールを使用したドメイン非参加のLinux VDAの作成を参照してください。
- MCSはベアメタルサーバーをサポートしていません。
- ドメイン非参加のLinux VDAで利用可能な機能は次のとおりです。

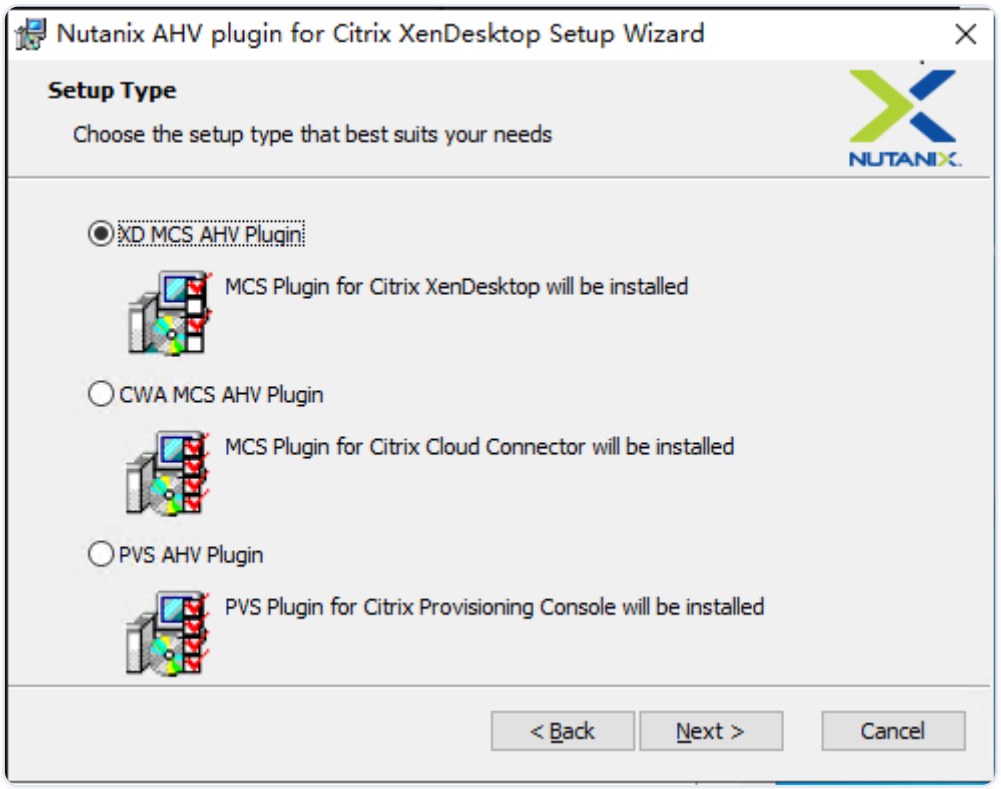
- ドメイン非参加VDAで指定された属性を持つローカルユーザーを作成する
- 非SSO認証
- Azure Active Directoryによる認証
- ランデブー V2

(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向けNutanix AHVプラグインをインストールして登録する

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



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

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

ステップ1c: プラグインのインストール後に次の手順を完了します

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`にNutanix Acropolisフォルダーが作成されていることを確認します。
- `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"`コマンドを実行します。
- オンプレミスのDelivery ControllerでCitrix Host、Citrix Broker、およびCitrix Machine Creation Services™を再起動するか、Citrix Cloud ConnectorでCitrix RemoteHCLServerサービスを再起動します。

> **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. ウィザードは以下の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。

ステップ2a: 接続

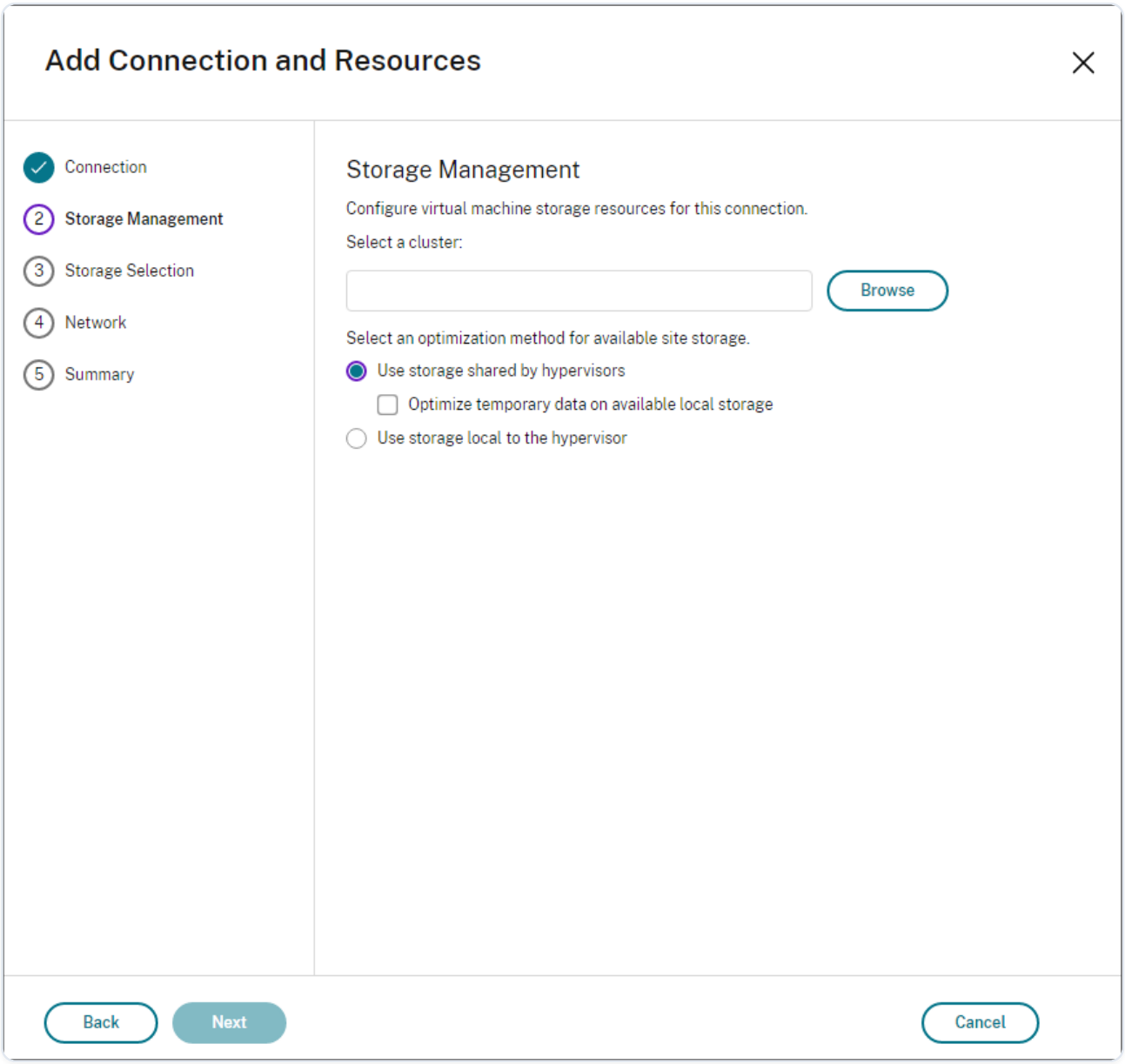
接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/add-conn-connection.png)

「**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。
- **Connection type**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**Manage**表示に表示されます。
- 仮想マシンを作成するツールを選択します: Machine Creation ServicesまたはCitrix Provisioning。

Connection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。

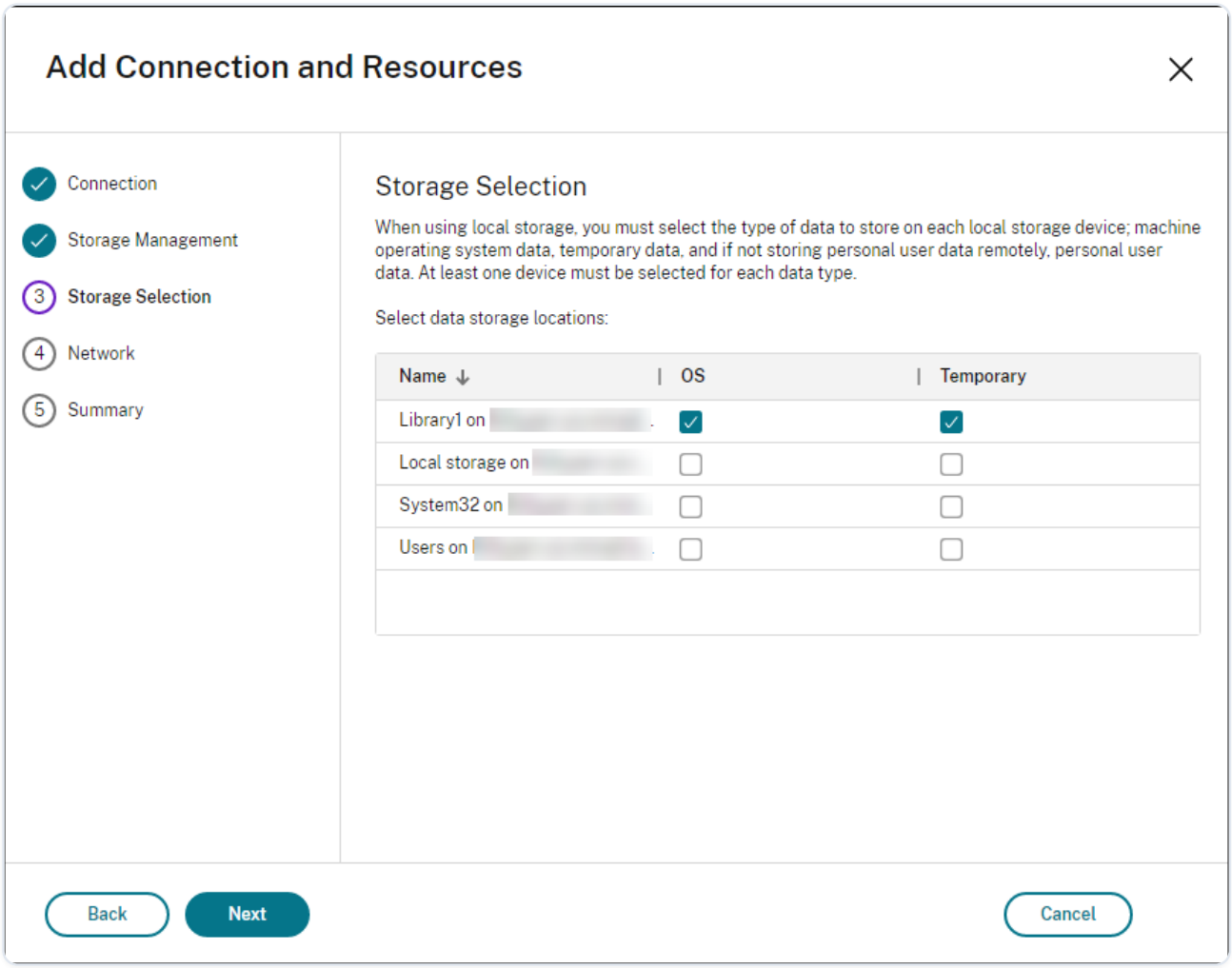
Hyper-V または VMware ホストへの接続を構成している場合は、クラスター名を参照して選択します。その他の接続タイプでは、クラスター名は要求されません。

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示します。（この接続を使用するマシンカタログで、デフォルト以外のテンポラリストレージサイズを指定できます。）**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、ローカルストレージに一時データキャッシュディスクを作成できません。**Manage**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の負荷を軽減するためにIntelliCacheを使用するかどうかを示します。「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。

ハイパーバイザーで共有されるストレージを選択し、**Optimize temporary data on available local storage**を有効にした場合、**Storage Selection**ページの下部にはさらに多くの構成オプションが含まれます。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. **Select Storage**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、その後**OK**を選択します。

ステップ2d: リージョン

(一部のホストタイプにのみ表示されます。) 地域の選択は、VMが展開される場所を示します。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、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するために、**管理**コンソールに表示されます。 VMが使用するネットワークを1つ以上選択します。

一部の接続タイプ（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際に、一時データストレージのデフォルト以外の値を構成できます。

> **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: マスターイメージの準備

> **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。

(XenServer (旧Citrix Hypervisor™) のみ) ステップ3a: XenServer VM Toolsのインストール

xe CLIまたはXenCenterを使用する各VMのテンプレートVMにXenServer 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、以下のいずれも実行できません。

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

1. 使用中のハイパーバイザーのバージョンに基づいて、XenServer Downloads page または Citrix Hypervisor Downloads page からLinux用XenServer VM Toolsファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz` ファイルをLinux VM、またはLinux VMがアクセスできる共有ドライブにコピーします。
3. tarファイルの内容を抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. GCPでホストされているRHEL 10.x/9.x/8.10およびRocky Linux 10.x/9.x/8.10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 restart NetworkManager
   ```

5. 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/SUSEの場合:

```
sudo rpm -i <extract-directory>/xe-guest-utilities_{package-version}
_x86.64.rpm
```

Ubuntu/Debianの場合:

```
sudo dpkg -i <extract-directory>/xe-guest-utilities_{package-version}
_amd64.deb
```

6. 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。XenServer® 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**と表示されます。

ステップ3b：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。

Linux VDAパッケージをインストールする前に、システム要件に従って(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html#net-requirements) .NETをインストールします。

.NETのインストール後、お使いのLinuxディストリビューションに基づいて、Linux VDA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 yum –y localinstall <PATH>/<Linux VDA RPM>
```

注:

GCPでホストされているサポート対象のRHELまたはRocky Linuxディストリビューションに 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 restart NetworkManager
```

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3c：（RHEL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3d：（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3e：（Ubuntuのみ）/etc/network/interfacesファイルを編集する

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

> **ヒント：**
>
> `/etc/network/interfaces`ファイルは、お使いのUbuntuマシンでは利用できない場合があります。ファイルが存在しない場合は、まず **net-tools** および **ifupdown** パッケージをインストールする必要があります。

ステップ 3f: (Ubuntu のみ) /etc/resolv.conf を指定する

`/run/systemd/resolve/stub-resolv.conf` ではなく、`/etc/resolv.conf` を `/run/systemd/resolve/resolv.conf` に指定します。

```
unlink /etc/resolv.conf



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

ステップ 3g: 使用するデータベースを指定する

Linux VDA パッケージのインストール後、SQLite、PostgreSQL、および組み込みレジストリサービスを切り替えることができます。これを行うには、次の手順を完了します。

注:

- VDI モードでは SQLite のみを使用し、ホスト型共有デスクトップ配信モデルでは PostgreSQL または組み込みレジストリサービスを使用することをお勧めします。
- 簡易インストールおよび MCS の場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで組み込みレジストリサービスを使用します。Linux ディストリビューションによって提供されるバージョンではなく、カスタムバージョンの PostgreSQL が必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように `/etc/xdl/db.conf` を編集し、簡易インストールスクリプト (`ctxinstall.sh`) または MCS スクリプト (`deploymcs.sh`) を実行する前に PostgreSQL サービスを開始する必要があります。
- **/etc/xdl/db.conf** を使用して、PostgreSQL のポート番号を構成することもできます。
- RHEL9、RHEL10、および Rocky9 で組み込みレジストリサービスを使用するには、CodeReady Builder リポジトリを有効にする必要があります。 RHEL9: subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms RHEL10: subscription-manager repos --enable codeready-builder-for-rhel-10-$(arch)-rpms Rocky9: dnf config-manager --enable crb

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールである場合は、この手順を省略します。
2. `deploymcs.sh` を実行する前に `/etc/xdl/db.conf` を編集します。以下は **db.conf** ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

#     ctxregd - use the built-in registry service # default choice is
```

```
ctxregd



DbType="ctxregd"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false
```

```
## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンの PostgreSQL を使用するには、**DbCustomizePostgreSQL** を true に設定します。

ステップ 3h: MCS 変数を構成する

MCS 変数を構成するには、次の 2 つの方法があります。

- `/etc/xdl/mcs/mcs.conf` ファイルを編集します。
- 簡易インストール GUI を使用します。簡易インストール GUI を開くには、Linux VDA のデスクトップ環境で `/opt/Citrix/VDA/sbin/linux-vda-easyinstaller` コマンドを実行します。

  「簡易インストールGUI」(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/easy-install-gui-ndj.png)

> **注:**
>
> ユーザーは「Advanced Configuration (optional)」ボタンをクリックして、詳細な構成オプションを展開することもできます。

> **ヒント:**
>
> **Save** をクリックして、指定したパスのローカルファイルに変数の設定を保存します。
> **Load** をクリックして、指定したファイルから変数の設定を読み込みます。

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ことも、必要に応じて変数をカスタマイズすることもできます（オプション）：

`DOTNET_RUNTIME_PATH`=path-to-install-dotnet-runtime

`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3i: 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、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、次のコマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加して、それぞれレジストリ値を書き込むまたは更新できます。

```
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\Cl
ipboardSelection" -v "Flags" -d "0x00000003"
```

ステップ3j: マスターイメージを作成する

1. `/etc/xdl/mcs/mcs.conf` を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh` を実行します。GUIを使用してMCS変数を構成する場合は、**Deploy** をクリックします。GUIで **Deploy** をクリックすると、GUIで設定した変数が `/etc/xdl/mcs/mcs.conf` ファイルで設定した変数を上書きします。
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、名前を付けます。

   - **(XenServer (旧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 **Stopped (deallocated)** と表示されてい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。

  「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/2607-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 **Stopped**と表示されていることを確認します。テンプレートVMを右クリックし、**Image > Create Image**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。**Create Image** をクリックします。

  EBSイメージの作成(/en-us/linux-virtual-delivery-agent/2607-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 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。

手順4a：DaaSで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. **管理 > 完全な構成**から、**マシンカタログ**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**コンテナ**ページで、以前にテンプレートVMに指定したコンテナを選択します。

   マスターイメージページで、イメージスナップショットを選択します。

仮想マシンページで、仮想CPUの数とvCPUあたりのコア数を確認します。マシン展開方法としてMCSを選択し、カタログで作成するマシンのIDとして**ドメイン非参加**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を手動で電源オンにします。この方法により、作成プロセスを続行できます。

手順4b：CVADでマシンカタログを作成する

Web StudioまたはPowerShellを使用して、CVADでドメイン非参加のマシンカタログを作成できます。詳細については、「ドメイン非参加のマシンID」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コレクションです。これにより、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が指定されます。詳細については、DaaSでデリバリーグループを作成するまたはCVADでデリバリーグループを作成するを参照してください。

簡単インストールを使用して、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、Citrix DaaSまたはCitrix Virtual Apps and Desktops™ 2407以降で、簡単インストール方法を使用して、ドメインに参加していないLinux VDAを作成し、マシンカタログに登録する手順を説明します。

> **重要:**
>
> - Citrix DaaS™ のお客様向け:

    - 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、ドメインに参加していないVDAを展開できます。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。
    - ドメインに参加していないVDAを作成するには、Citrix Gatewayサービスを使用しているお客様は、Rendezvous V2 が有効になっていることを確認する必要があります。Cloud Connectorは、オンプレミスのハイパーバイザーにマシンをプロビジョニングする場合、またはWorkspaceでActive DirectoryをIDプロバイダーとして使用する場合にのみ必要です。

- ドメインに参加していないVDAを作成するには、MCSを使用することもできます。詳細については、「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」を参照してください。
    - 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では、次の機能が利用できます。
    - 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
    - Azure Active Directoryによる認証
    - ランデブー V2

ステップ1：空のマシンカタログを作成する

Citrix Web Studio™ にサインインし、マシンを含まない空のマシンカタログを作成します。Linux VDAは、電源管理されたマシンカタログへの登録にトークンファイルを使用することをサポートしていません。

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと同様です。これらのタスクを完了する方法の詳細については、「マシンカタログを作成する」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。
- 同じマシンカタログにドメイン参加マシンと非ドメイン参加マシンを混在させないでください。
- トークン登録マシンカタログを作成するには、**マシンタイプ**ページで**シングルセッションOS**または**マルチセッションOS**を選択し、**マシン管理**ページで**「電源管理されていないマシン（物理マシンなど）」**を選択します。

> **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。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。

手順2：登録トークンを作成する

簡易インストールを使用して非ドメイン参加VDAを作成するには、VDAをマシンカタログに登録し、VDAをクラウドまたはオンプレミスのデリバリーコントローラーに認証するためのトークンファイルが必要です。Linux VDAは、電源管理されたマシンカタログへの登録にトークンファイルを使用することをサポートしていません。

登録トークンを作成するには、Citrix Web Studioで次の手順を実行します。

> **ヒント:**
>
> Citrix DaaSでは、Web Studioは「完全な構成」として知られています。

1. 以前に作成した空のマシンカタログを選択し、アクションバーで**登録トークンの管理**を選択します。
2. **登録トークンの管理**ページで、**生成**をクリックして登録トークンを作成します。または、有効な状態の既存のトークンを選択することもできます。
3. ウィザードに従って設定を完了します。
4. 正常に作成されたトークンをダウンロードし、最小限の特権で安全な場所に保存します。後で簡易インストールスクリプトを実行する際に、CTX_XDL_NDJ_ENROLLMENT_TOKEN_FILE変数を使用して、VDAを登録するためのトークンファイルへのパスを指定できます。

ステップ3：.NETのインストール

Linux VDAパッケージをインストールする前に、システム要件に従って .NETをインストールします。

.NET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。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4：自己署名CA証明書の追加

自己署名CA証明書の設定の詳細については、「WebSocket 用の自己署名証明書を構成する」を参照してください。このセクションでは、Linux VDAにCA証明書を配置および更新する方法について説明します。

- RHELおよびRocky Linuxの場合：

  自己署名CA証明書をLinux VDA上の **/etc/pki/ca-trust/source/anchors** ディレクトリに保存し、次のコマンドを実行して証明書を更新します。

  ```
  sudo update-ca-trust
  ```

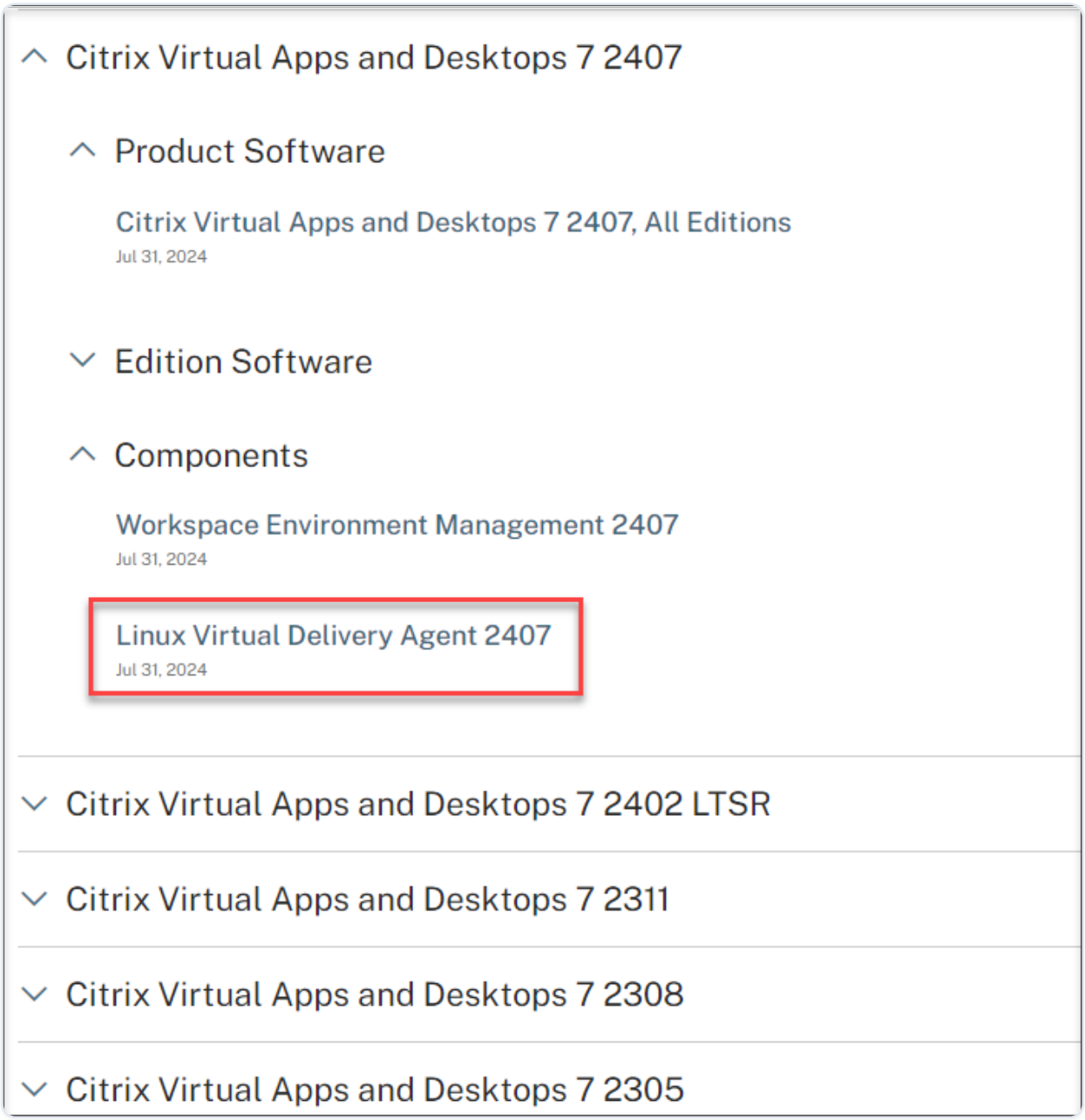
- SUSE、Ubuntu、およびDebianの場合：

  ルートCA証明書を **/usr/local/share/ca-certificates** ディレクトリに配置します。その後、**update-ca-certificate** コマンドを実行します。

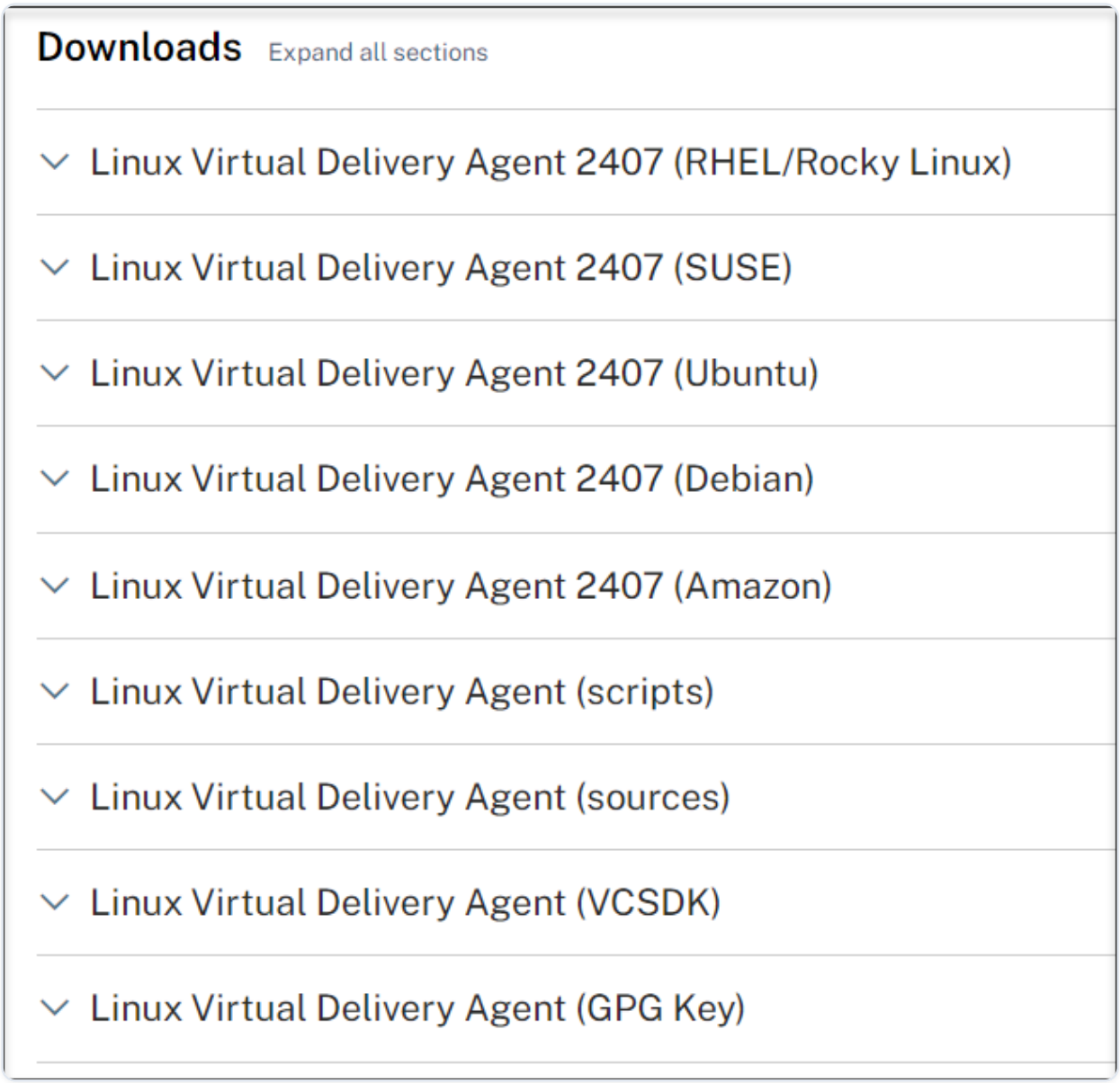
  ```
  sudo update-ca-certificates
  ```

ステップ5：Linux VDAパッケージのダウンロード

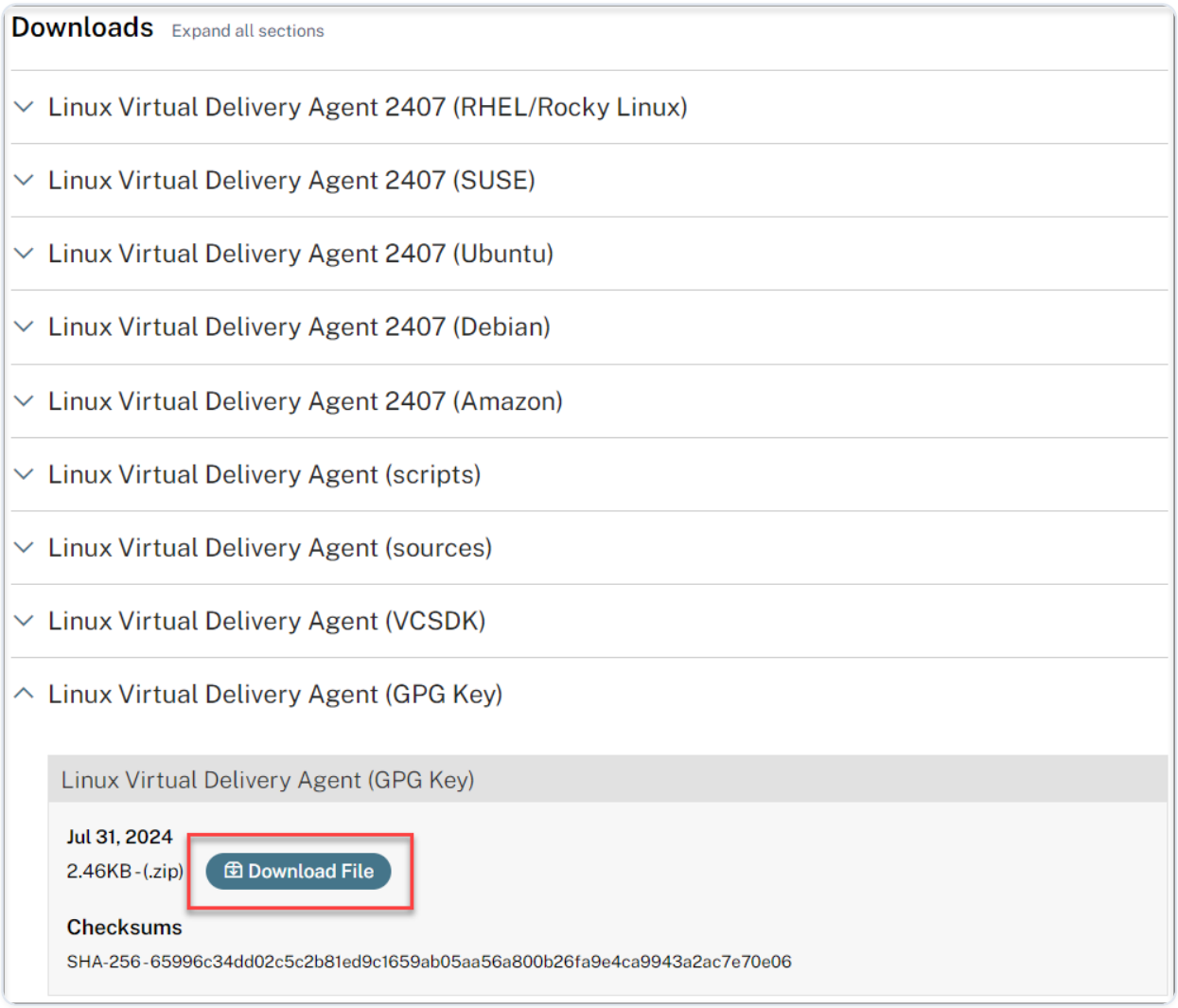
1. Citrix Virtual Apps and Desktopsダウンロードページに移動します。
2. Citrix Virtual Apps™ and Desktopsの適切なバージョンを展開します。
3. **Components** を展開してLinux VDAを見つけます。例：



4. Linux VDAのダウンロードにアクセスするには、Linux VDAリンクをクリックします。



5. お使いのLinuxディストリビューションに合ったLinux VDAパッケージをダウンロードします。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：



公開鍵を使用してLinux VDAパッケージの整合性を検証するには：

- RPMパッケージの場合、公開鍵をRPMデータベースにインポートし、パッケージの整合性を確認するには、次のコマンドを実行します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

- DEBパッケージの場合、公開鍵をDEBデータベースにインポートし、パッケージの整合性を確認するには、次のコマンドを実行します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

手順6：Linux VDAパッケージをインストールする

Linux VDAの環境をセットアップするには、次のコマンドを実行します。

RHELおよびRocky Linuxディストリビューションの場合：

注:

- RHELおよびCentOSの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。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。
- RHEL 9.xおよびRocky Linux 9.xにLinux VDAをインストールする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

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

注:

GCPでホストされているサポート対象のRHELまたはRocky Linuxディストリビューションに 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に初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

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 11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/ bullseye main`行を追加します。
- GCP上のUbuntuの場合、RDNSを無効にします。そのためには、/etc/krb5.confの**の下にrdns = false**行を追加します。

SUSEディストリビューションの場合：

1. AWS、Azure、GCP上のSUSE 15.7の場合、以下を確認してください。
   - **libstdc++6**バージョン12以降を使用していること。
   - **/etc/sysconfig/windowmanager** 内の **Default_WM** パラメーターが **"gnome"** に設定されています。
2. Linux VDAをインストールするには、次のコマンドを実行します。

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

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（クラウドベンダーまたはNVIDIA製）をインストールします。

ステップ8：使用するデータベースを指定する

Linux VDAパッケージのインストール後、SQLite、PostgreSQL、および組み込みレジストリサービスを切り替えることができます。

- そのためには、以下の手順を完了してください。
  - VDIモードではSQLiteのみを使用し、ホスト型共有デスクトップ配信モデルではPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
  - Easy InstallおよびMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように`/etc/xdl/db.conf`を編集し、Easy Installスクリプト (`ctxinstall.sh`) またはMCSスクリプト (`deploymcs.sh`) を実行する前にPostgreSQLサービスを開始する必要があります。
  - PostgreSQLのポート番号を設定するには、**/etc/xdl/db.conf**を使用することもできます。
  - RHEL9、RHEL10、およびRocky9で組み込みレジストリサービスを使用するには、CodeReady Builderリポジトリを有効にする必要があります。

  RHEL9: subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms
  RHEL10: subscription-manager repos --enable codeready-builder-for-rhel-10-$(arch)-rpms
  Rocky9: dnf config-manager --enable crb

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略してください。
2. `deploymcs.sh`を実行する前に`/etc/xdl/db.conf`を編集します。以下はdb.confファイルの例です。

```
# database configuration file for Linux VDA
```

```
## database choice


# possible choices are:


#     SQLite


#     PostgreSQL


#     ctxregd - use the built-in registry service # default choice is
ctxregd


DbType="ctxregd"









## database port


# specify database port for the database.


# if not specified, default port will be used:


# SQLite: N/A
```

```
# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false
```

```
DbCustomizePostgreSQL=false




## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

Linux VDAパッケージをインストールした後、**/etc/xdl/db.conf** を編集することで、SQLite、PostgreSQL、または組み込みのレジストリサービスを使用するように指定できます。

そのためには、**sudo /opt/Citrix/VDA/sbin/ctxinstall.sh**または**/opt/Citrix/VDA/bin/easyinstall**を実行する前に、**/etc/xdl/db.conf**を編集します。

注:

- VDIモードではSQLiteのみを使用することをお勧めします。
- イージーインストールと MCS の場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで組み込みレジストリサービスを使用します。
- **/etc/xdl/db.conf** を使用して PostgreSQL のポート番号を構成することもできます。

- RHEL9、RHEL10、Rocky9 で組み込みレジストリサービスを使用するには、CodeReady Builder リポジトリを有効にする必要があります。 RHEL9: subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms RHEL10: subscription-manager repos --enable codeready-builder-for-rhel-10-$(arch)-rpms Rocky9: dnf config-manager --enable crb

ステップ 9: イージーインストールを実行して環境と VDA を構成し、インストールを完了する

Linux VDA パッケージのインストール後、ctxinstall.sh スクリプトまたは GUI を使用して実行環境を構成します。

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。前述のコマンドから、最初の行に **#!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

または、1 つのコマンドを使用してすべての変数を指定することもできます。

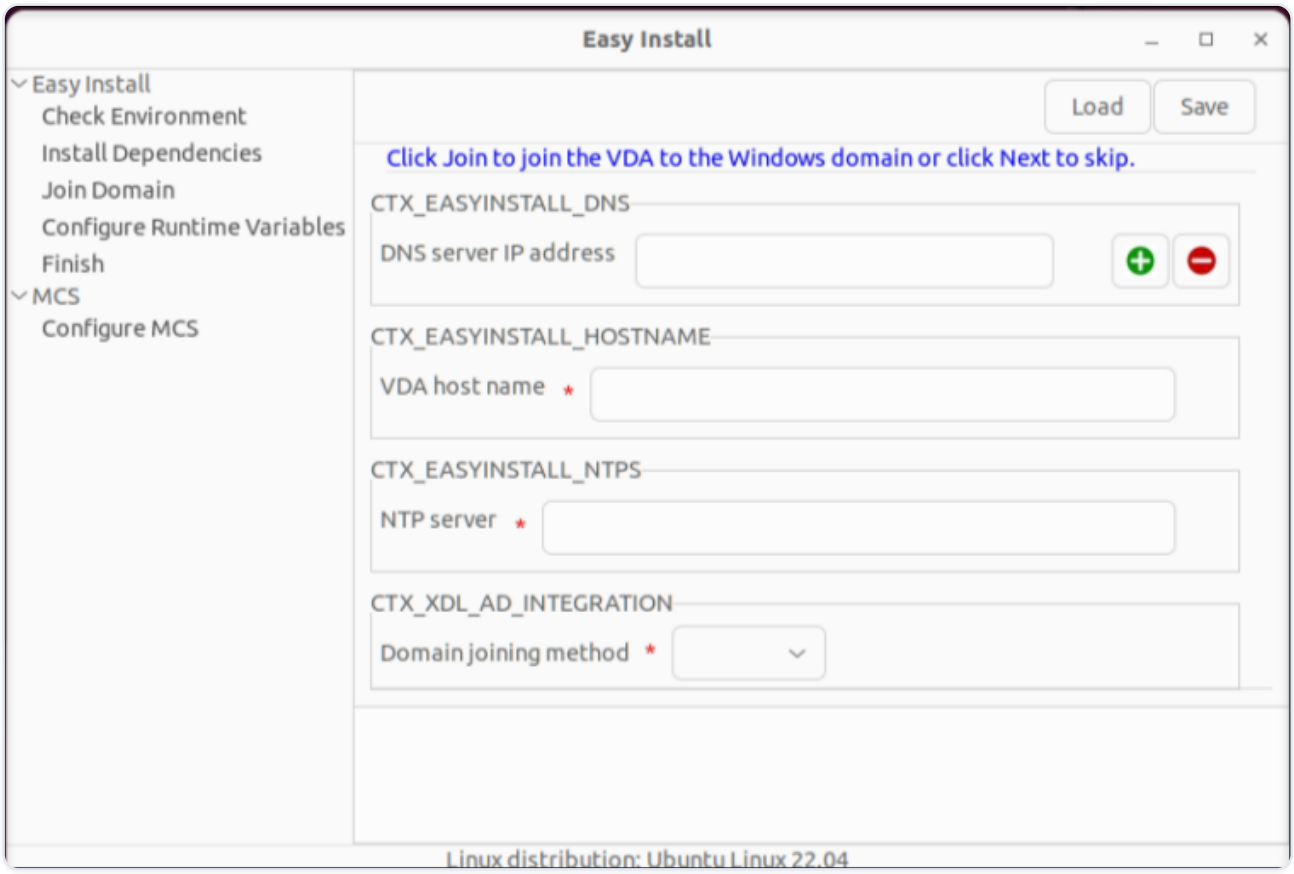
VDA 実行環境変数 (**'CTX_XDL_'** で始まる変数) を設定するには、**ctxinstall.sh -s** を実行します (ここで **s** は小文字であることに注意してください)。

注:

ランタイム環境を設定する前に、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 の場合は、**/etc/locale.gen** ファイルを編集し、**# en_US.UTF-8 UTF-8** の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。

GUI

GUI を使用してイージーインストールを実行できます。VDA のデスクトップ環境で **/opt/Citrix/VDA/sbin/linux-vda-installer** コマンドを実行し、イージーインストール GUI の指示に従います。



イージーインストール GUI は、次の操作をガイドします。

- システム環境の確認
- データベースの構成
- 依存関係のインストール
- ランタイム環境の構成

> **注:**
>
> ユーザーは「Advanced Configuration (optional)」ボタンをクリックして、詳細な構成オプションを展開することもできます。

ctxinstall.sh

ctxinstall.sh は、いくつかの事前構成と VDA 実行環境変数の設定を行うための簡易インストールスクリプトです。

- このスクリプトは root のみが実行できます。

- 簡易インストールでは、使用されるすべての環境変数の値を設定、保存、同期するための構成ファイルとして /opt/Citrix/VDA/sbin/ctxinstall.conf を使用します。テンプレート (ctxinstall.conf.tmpl) を注意深く読み、独自の ctxinstall.conf をカスタマイズすることをお勧めします。構成ファイルを初めて作成する場合は、次のいずれかの方法を使用してください。
  - /opt/Citrix/VDA/sbin/ctxinstall.conf.tmpl テンプレートファイルをコピーし、/opt/Citrix/VDA/sbin/ctxinstall.conf として保存します。
  - ctxinstall.sh を実行します。ctxinstall.sh を実行するたびに、入力は /opt/Citrix/VDA/sbin/ctxinstall.conf に保存されます。
- 簡易インストールはモジュール実行をサポートしています。モジュールには、事前チェック、インストール、ドメイン構成、セットアップ、検証が含まれます。
- このスクリプトのデバッグの詳細は、/var/log/xdl/ctxinstall.log で確認できます。

詳細については、ヘルプコマンド **ctxinstall.sh -h** を使用してください。

注:

- 最小特権の原則に従い、ドメイン参加パスワードがファイルに設定されている可能性があるため、root ユーザーのみが **/opt/Citrix/VDA/sbin/ctxinstall.conf** を読み取れるようにしてください。
- Linux VDA をアンインストールすると、**/opt/Citrix/VDA** の下のファイルが削除されます。VDA をアンインストールする前に、**/opt/Citrix/VDA/sbin/ctxinstall.conf** をバックアップすることをお勧めします。

ctxinstall.sh は対話モードまたはサイレントモードで実行できます。スクリプトを実行する前に、次の環境変数を設定してください。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は「n」です。ドメインに参加しないシナリオでは、「y」に設定します。
- **CTX_XDL_NDJ_ENROLLMENT_TOKEN_FILE='<path-to-token-file-on-vda-machine\>'** – 簡易インストールを使用してドメインに参加していない VDA を作成するには、Delivery Controller のマシンカタログに VDA を登録するためのトークンファイルが必要です。トークンは最小特権で安全な場所に保存してください。

- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル (VDI) または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どうか。HDX 3D Pro環境の場合、値を「y」に設定します。
- **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_START_SERVICE='y\|n'** – 構成完了時にLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシンの起動後にLinux Virtual Desktopサービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA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_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用するGNOME、GNOME Classic、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で構成されているデフォルトのデスクトップが使用されます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、またはXfceデスクトップ環境を指定します。'<none\>'に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。CTX_XDL_DESKTOP_ENVIRONMENT変数を使用して、セッションで使用するデスクトップ環境を指定できます。コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- **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_VDA_PORT='<port-number\>'** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。

対話モード

対話モードで**ctxinstall.sh**スクリプトを実行するには、**-S**オプションなしで**sudo /opt/Citrix/VDA/sbin/ctxinstall.sh**コマンドを使用します。コマンドラインインターフェイスの各プロンプトで、

関連する変数の値を入力します。変数がすでに設定されている場合、**ctxinstall.sh**は変更するかどうかを確認します。

サイレントモード

サイレントモードでは、**/opt/Citrix/VDA/sbin/ctxinstall.conf**またはexportコマンドを使用して、上記の変数を設定する必要があります。その後、**ctxinstall.sh -S**を実行します（ここで**S**は**大文字**であることに注意してください）。必要な変数がすべて設定されていないか、一部の値が無効な場合、デフォルト値がない限り、**ctxinstall.sh**は実行を中止します。

設定した場合、各変数のエクスポートされた値は、**/Citrix/VDA/sbin/ctxinstall.conf**内の値を上書きします。更新されたすべての値は、**/Citrix/VDA/sbin/ctxinstall.conf**に保存されます。

```
export CTX_XDL_NON_DOMAIN_JOINED='y'

export CTX_XDL_NDJ_ENROLLMENT_TOKEN_FILE='<token-file-path>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_START_SERVICE='y|n'

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_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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

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変数を指定できます。

VDA実行環境変数（「**CTX_XDL_**」で始まる変数）を設定するには、**ctxinstall.sh -s** を実行します（ここでの文字 **s** は**小文字**であることに注意してください）。

ステップ10: 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11: 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 systemctl start ctxhdx.service



sudo systemctl start ctxvda.service
```

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo systemctl stop ctxvda.service



sudo systemctl stop ctxhdx.service
```

> **注:**

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

> **Linux VDA を再起動する:**

Linux VDA サービスを再起動するには:

```
sudo systemctl stop ctxvda.service



sudo systemctl restart ctxhdx.service



sudo systemctl start ctxvda.service
```

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.service



sudo systemctl status ctxhdx.service
```

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、次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 2503」を参照してください。

ステップ13：ローカルアカウントマッピングを有効にする

ローカルアカウントマッピングが有効になっていない場合

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は、ログオン時に失敗し、パスワードプロンプトが表示されますが、正しいパスワードを受け付けません。次のようなエラーが`hdx.log`で見つかることがあります。

```
2024-09-25 06:40:30.767979 [LOGIN_BOX:ERROR] <P124211:T58675:S4> citrix-
ctxlogin: PamAuthenticate: pam authentication: Authentication failure. Can
retry for user 'user3'

2024-09-25 06:40:30.768431 [LOGIN:ERROR] <P124211:T58675:S4> citrix-
ctxlogin: LoginBoxValidate: Failed to validate user 'user3'. Due to
INVALID_PASSWORD

2024-09-25 06:40:30.768947 [LOGIN_AUTH:INFO] <P124218:T58682:S4> citrix-
ctxloginui: CalLabelFormat: Update username label on LoginUI from 'user3'
to 'user3' (18)

2024-09-25 06:41:18.766196 [GFX_SLIDER:ERROR] <P124161:T58699:S4> citrix-
ctxgfx: GfxCreateSliderListener: Failed to get user home dir.
```

ローカルアカウントマッピングを有効にする

ユーザーがドメインに参加していないLinux VDAにログオンできるようにするには、次のコマンドを使用してローカルアカウントマッピングを有効にします。

```
Set-BrokerDesktopGroup -Name "<your delivery group name>"
-MachineLogOnType LocalMappedAccount
```

オンプレミス展開の場合、Delivery Controllerで直接コマンドを実行します。Citrix DaaS展開の場合、Citrix Virtual Apps and Desktops Remote PowerShell SDKを介してコマンドを実行します。

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

MCS を使用して、ドメイン参加済みおよびドメイン非参加の VDA を作成できます。Citrix DaaS でドメイン非参加の Linux VDA を作成する場合は、専用の記事「ドメイン非参加の Linux VDA を MCS を使用して作成する」も参照してください。

重要:

2212 リリース以降の重要な変更点は次のとおりです。

- /etc/xdl/mcs/mcs.conf ファイル内、または簡易インストール GUI 上の **AD_INTEGRATION** 変数には、デフォルト値がなくなりました。必要に応じて値を設定する必要があります。詳細については、この記事の「手順 3i: MCS 変数を構成する」セクションを参照してください。
- /etc/xdl/mcs/mcs.conf 内の **UPDATE_MACHINE_PW** エントリの有効な値は、**enabled** または **disabled** ではなく、**Y** または **N** です。詳細については、この記事の「マシンアカウントのパスワード更新を自動化する」セクションを参照してください。

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 13.6	はい	はい	いいえ	はい
デビアン 12.15	はい	はい	いいえ	はい
RHEL 10.2/9.8/9.6	はい	はい	はい	いいえ
RHEL 8.10	はい	はい	はい	はい
ロッキー Linux 10.2/9.8/9.6	はい	はい	はい	いいえ
ロッキー リナックス 8.10	はい	はい	はい	いいえ

	ウィンバインド	SSSD	セントリファイ	PBIS
SUSE 15.7	はい	はい	いいえ	はい
ウブントゥ 26.04	はい	はい	いいえ	いいえ
ウブントゥ 24.04	はい	はい	いいえ	はい
ウブントゥ 22.04	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 7/8	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、以下のタスクを完了してください。

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

サポートされているハイパーバイザー

- AWS
- GCP
- マイクロソフト アジュール
- シトリックス XenServer
- 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ハイパーバイザー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、予期しない結果が生じ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

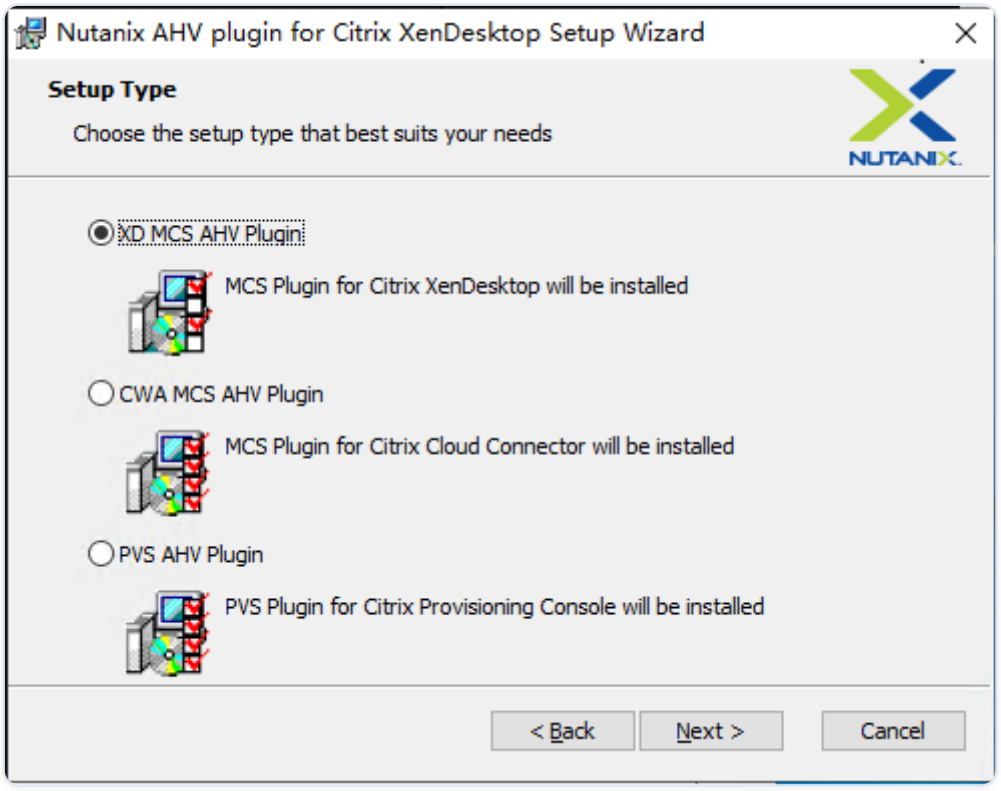
- 2203リリース以降、Citrix Virtual Apps and Desktops™およびCitrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal Hybrid Multi-Cloud（HMC）ライセンスが必要です。
- ベアメタルサーバーは、仮想マシンを作成するためのMCSでの使用はサポートされていません。

（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向けNutanix AHVプラグインをインストールして登録する

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



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

Citrix Cloud™ Connectors 用の **CWA MCS AHV プラグイン** 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 にインストールしてください。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します

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0` に Nutanix Acropolis フォルダーが作成されていることを確認します。
- `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"` コマンドを実行します。
- オンプレミスの Delivery Controller で Citrix Host、Citrix Broker、および Citrix Machine Creation Services を再起動するか、Citrix Cloud Connector で Citrix RemoteHCLServer Service を再起動します。

> ヒント:

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

ステップ 2: ホスト接続を作成する

このセクションでは、Azure、AWS、XenServer® (旧 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。

> **注:**
>
> オンプレミスの Delivery Controller の場合、オンプレミスの Citrix Studio で **> >** を選択してホスト接続を作成します。クラウド Delivery Controller の場合、Citrix Cloud の Web ベースの Studio コンソールで **> >** を選択してホスト接続を作成します。

> 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と管理 および Citrix DaaS ドキュメントの 接続の作成と管理 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
- Citrix Studio で AWS へのホスト接続を作成する
- Citrix Studio で XenServer へのホスト接続を作成する
- Citrix Studio で GCP へのホスト接続を作成する
- Citrix Studio で Nutanix へのホスト接続を作成する
- Citrix StudioでVMwareへのホスト接続を作成する

Citrix StudioでAzureへのホスト接続を作成する

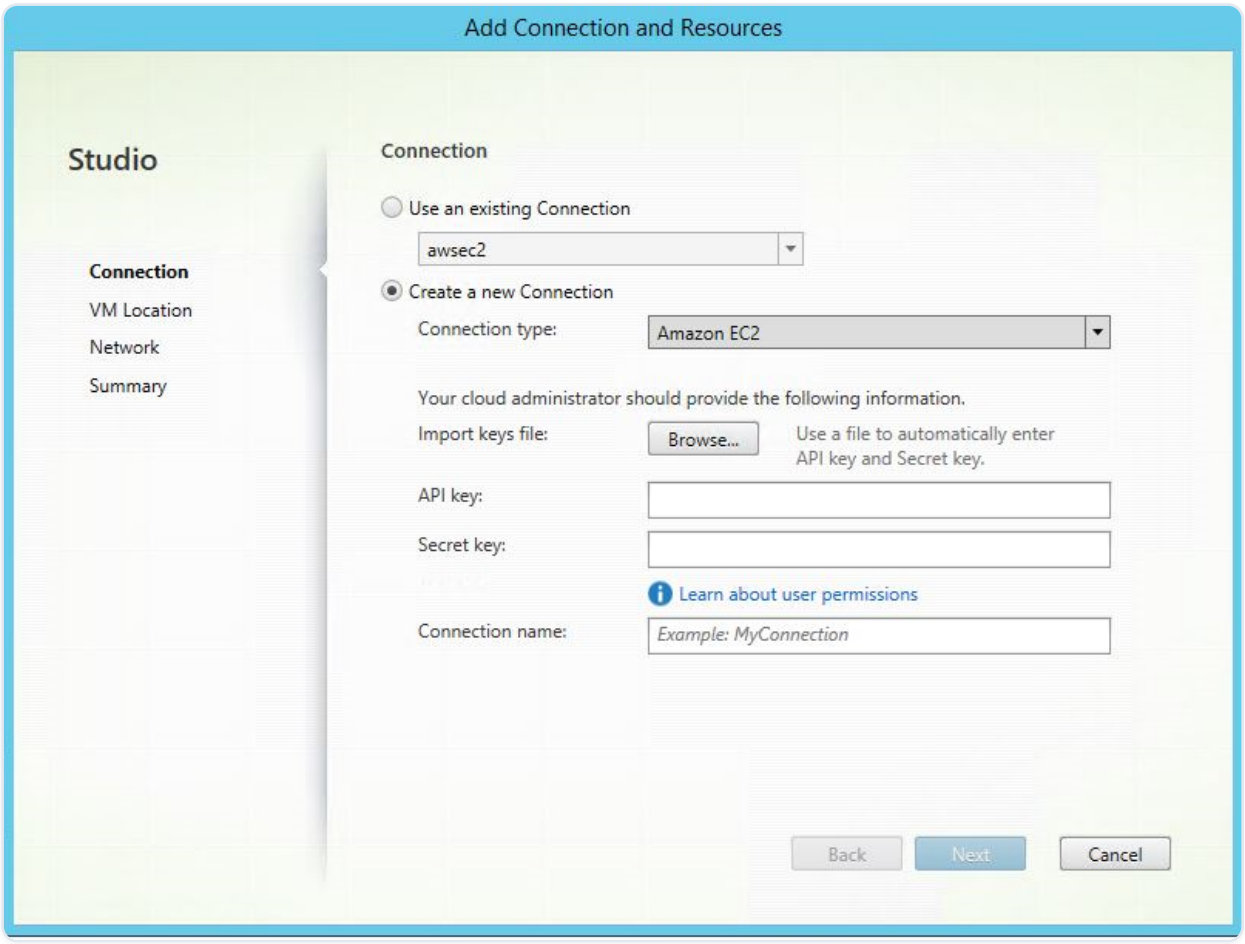
1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
2. **Add Connection and Resources**ウィザードで、接続の種類としてMicrosoft Azureを選択します。
3. 接続の種類としてMicrosoft Azureを選択します。
4. ウィザードのページに従って操作します。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、Create non-domain-joined Linux VDAs using MCSの記事の「**手順2：ホスト接続を作成する**」を参照してください。

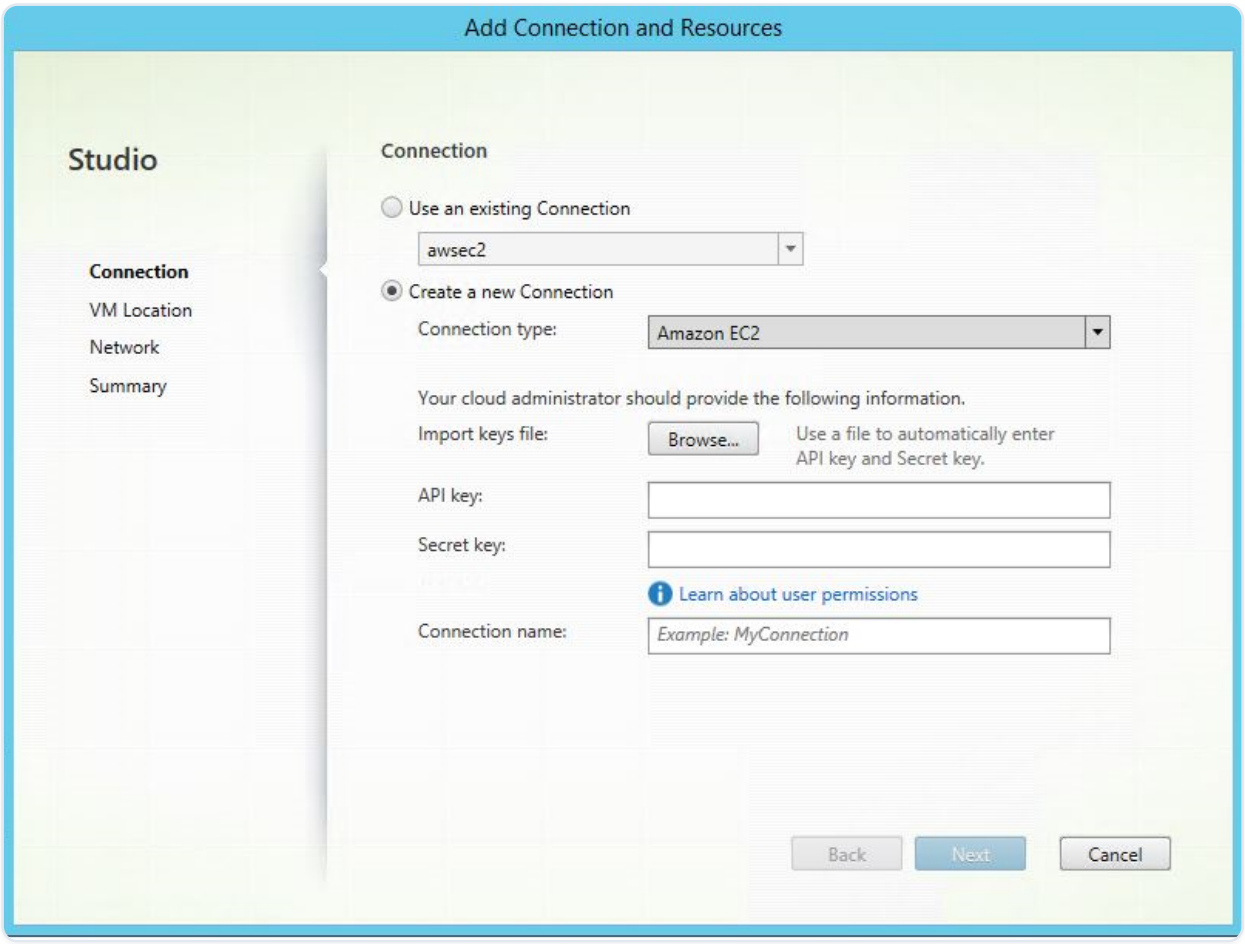
Citrix StudioでAWSへのホスト接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
2. **Add Connection and Resources**ウィザードで、接続の種類として**Amazon EC2**を選択します。Amazon WorkSpaces Core Managed Instancesへの接続を作成するには、Connection to Amazon WorkSpaces Core Managed Instancesを参照してください。

   ここでは、オンプレミスのCitrix StudioでのAmazon EC2接続を例にとります。



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



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

a. AWSサービスにサインインします。

b. Identity and Access Management (IAM) コンソールに移動します。 1 左側のナビゲーションペインで、**ユーザー**を選択します。

c. 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。

d. 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。

e. **.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。

4. ウィザードの指示に従ってページを進めます。特定のページコンテンツは、選択した接続タイプによって異なります。各ページを完了したら、**次へ**を選択して**概要**ページに到達します。

Citrix StudioでXenServer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**構成 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。クラウドDelivery Controllerの場

合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。

2. **接続とリソースの追加**ウィザードで、**接続の種類**フィールドでXenServer（旧Citrix Hypervisor）を選択します。
3. 接続アドレス（XenServer URL）と資格情報を入力します。
4. 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仮想化環境に従ってGCP環境をセットアップし、次の手順を実行してGCPへのホスト接続を作成します。

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。
2. **接続とリソースの追加**ウィザードで、接続の種類として**Google Cloud Platform**を選択します。

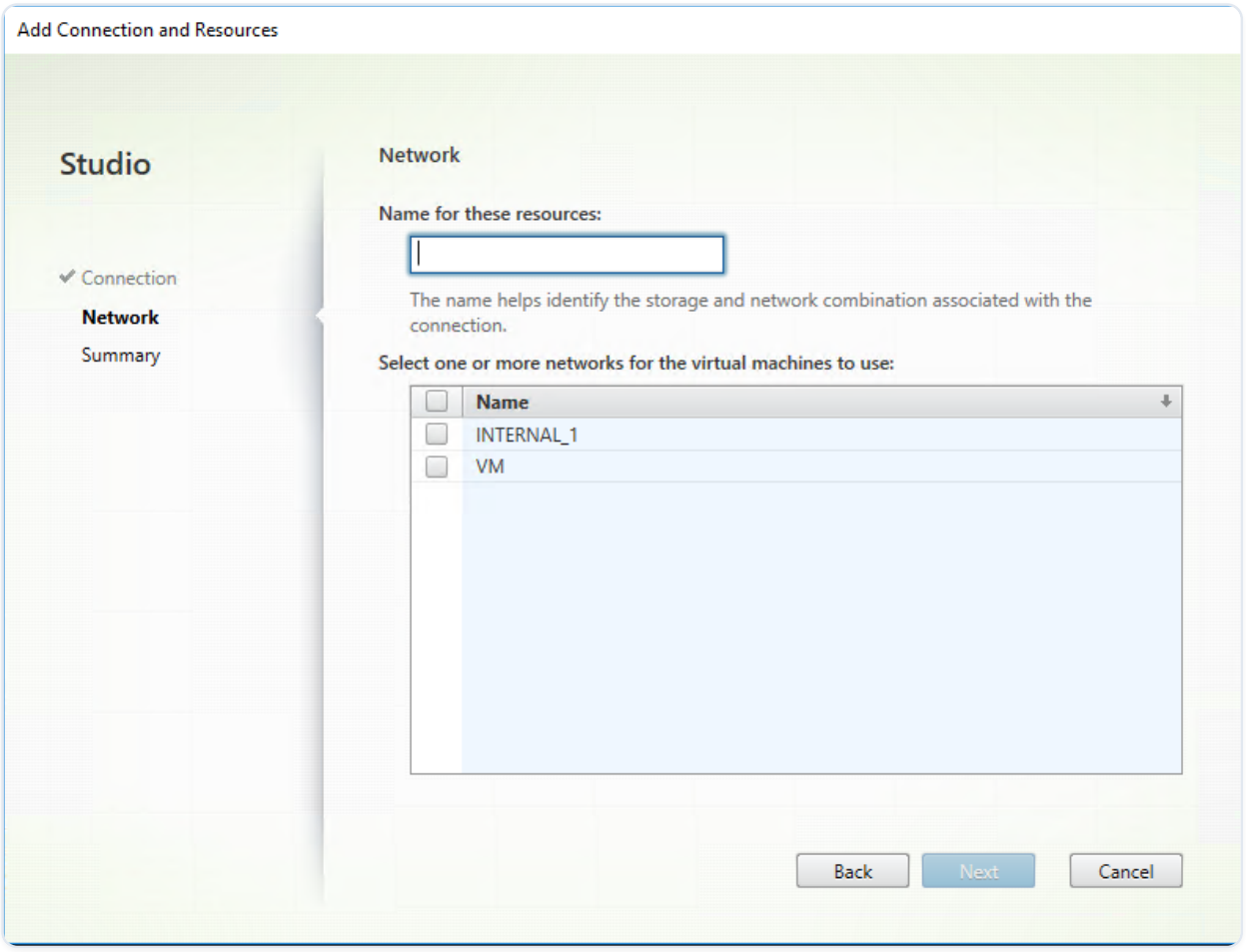
   たとえば、Citrix CloudのWebベースのStudioコンソールで：

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/add-conn-connection.png)
3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードがページを案内します。各ページのコンテンツは、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、Create non-domain-joined Linux VDAs using MCSの記事の**手順2：ホスト接続を作成する**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**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
2. **Add Connection and Resources**ウィザードの**Connection**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。**Network**ページで、ユニットのネットワークを選択します。

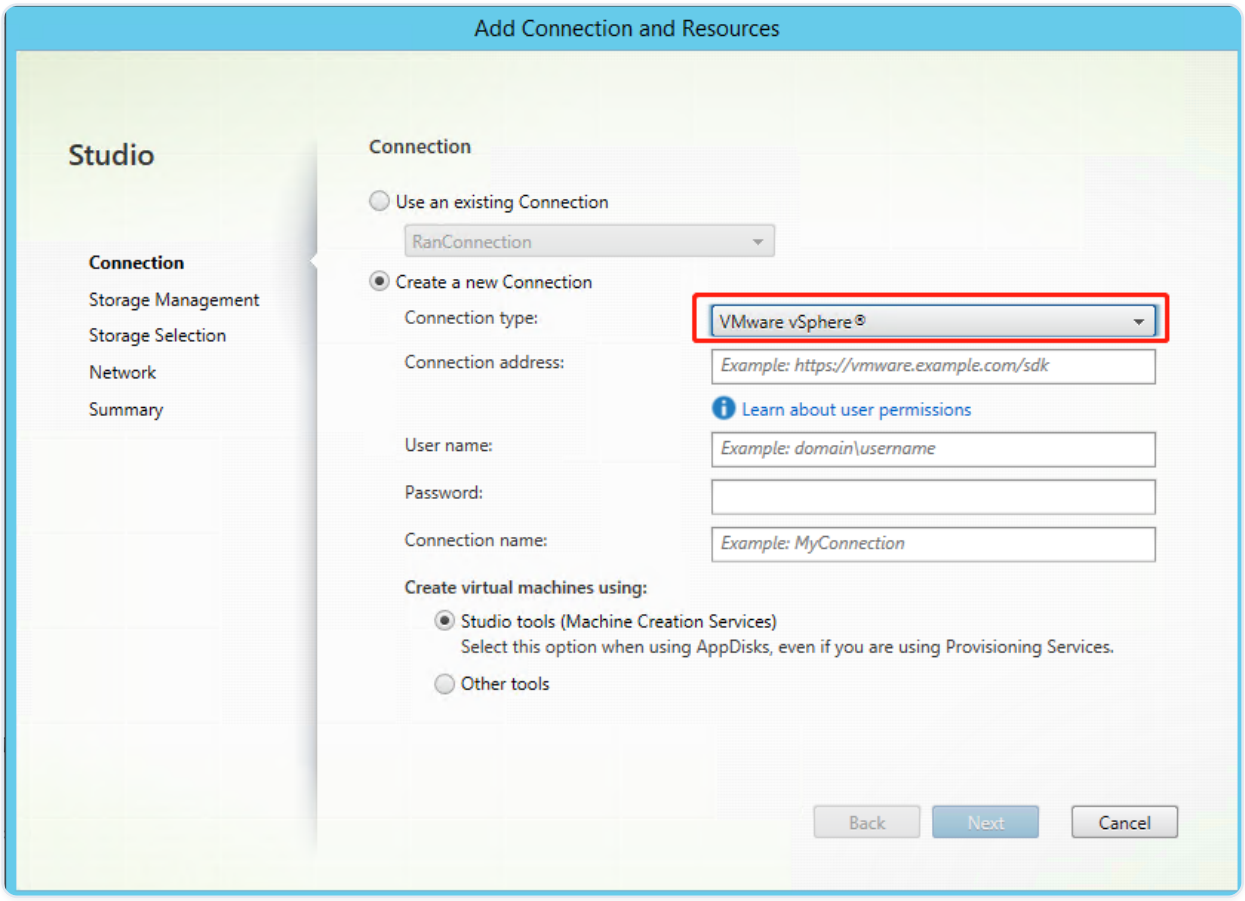
   例：オンプレミスのCitrix Studioの場合：



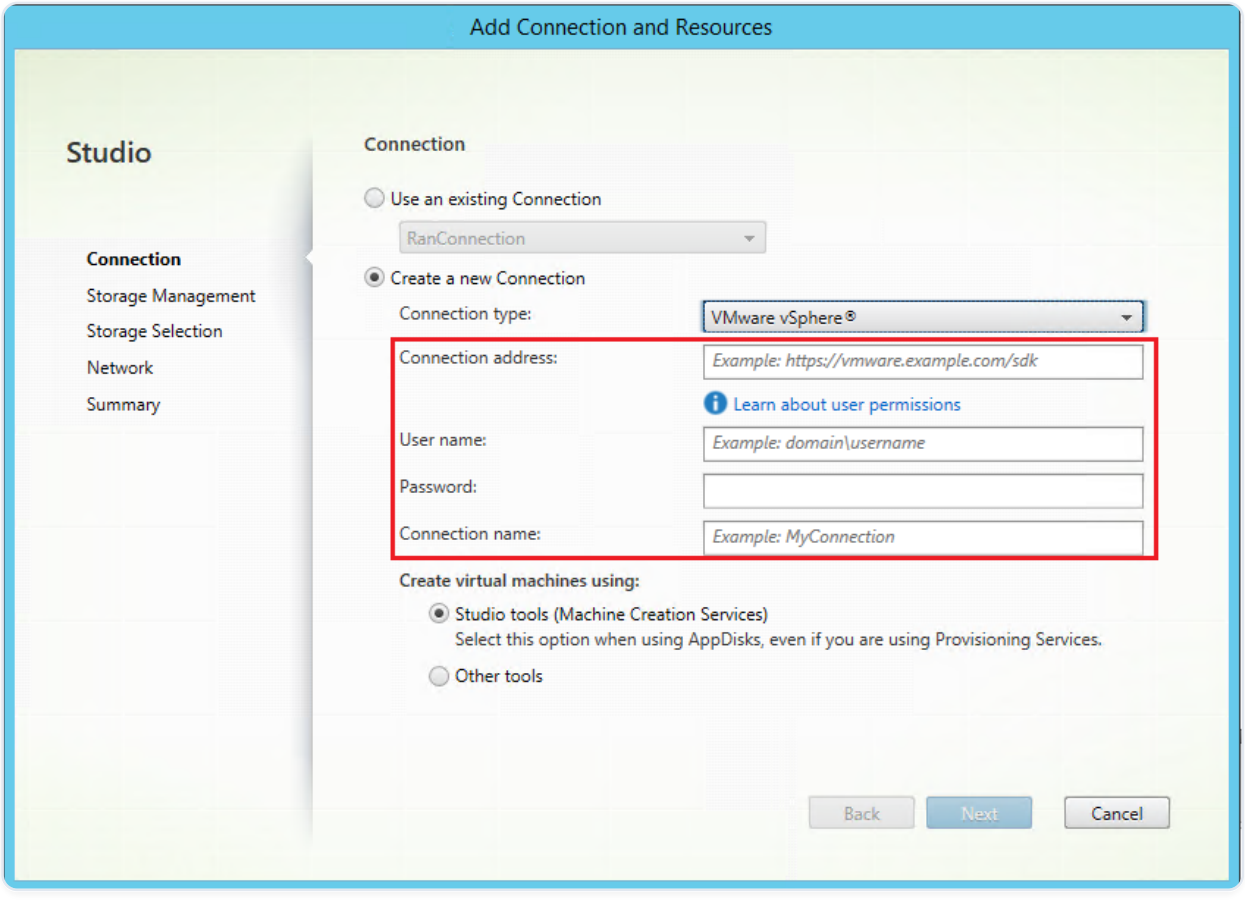
Citrix StudioでVMwareへのホスト接続を作成する

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. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
3. 接続の種類としてVMware vSphereを選択します。

   例：オンプレミスのCitrix Studioの場合：



4. VMwareアカウントの接続アドレス（vCenter ServerのURL）、資格情報、および接続名を入力します。



手順3：マスターイメージを準備する

(XenServer のみ) ステップ 3a: XenServer 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 XenServer VM Tools をインストールします。ツールをインストールしないと、VM のパフォーマンスが低下する可能性があります。ツールがないと、次のいずれも実行できません。

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

1. 使用しているハイパーバイザーのバージョンに基づいて、XenServer ダウンロードページ または Citrix Hypervisor ダウンロードページ から Linux 用 XenServer VM Tools ファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz` ファイルを Linux VM または Linux VM がアクセスできる共有ドライブにコピーします。
3. tar ファイルの内容を抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. 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/SUSE の場合:

   ```
   sudo rpm -i <extract-directory>/xe-guest-utilities_{package-version}
   _x86.64.rpm
   ```

   Ubuntu/Debian の場合:

   ```
   sudo dpkg -i <extract-directory>/xe-guest-utilities_{package-version}
   _amd64.deb
   ```

5. XenCenter の タブでテンプレート VM の仮想化状態を確認します。XenServer 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 と表示されます。

ステップ 3b: AWS、Azure、GCP 上の SUSE 15.7 の構成を確認する

AWS、Azure、GCP 上の SUSE 15.7 の場合、以下を確認してください。

- **libstdc++6** バージョン 12 以降を使用していること。
- **/etc/sysconfig/windowmanager** 内の **Default_WM** パラメーターが **"gnome"** に設定されていること。

ステップ 3c: GCP 上の Ubuntu で RDNS を無効にする

テンプレート VM で、/etc/krb5.conf の の下に **rdns = false** の行を追加します。

ステップ 3d: テンプレート VM に .NET をインストールする

Linux VDA パッケージをインストールする前に、システム要件に従って .NET をインストールします(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html#net-requirements)

.NET 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 3e: テンプレート VM に Linux VDA パッケージをインストールする

.NET のインストール後、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、Linux VDA をインストールします。

RHEL/CentOS/Rocky Linux の場合:

> **注:**
>
> RHEL 9.x および Rocky Linux 9.x に Linux VDA をインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

```
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 3f: (RHEL 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 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 3g: (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 3h: 使用するデータベースを指定する

Linux VDA パッケージのインストール後、SQLite、PostgreSQL、および組み込みのレジストリサービスを切り替えることができます。これを行うには、次の手順を完了します。

注:

- VDI モードでは SQLite を、ホスト型共有デスクトップ配信モデルでは PostgreSQL を使用することをお勧めします。
- 簡易インストールおよび MCS の場合、SQLite または PostgreSQL を手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** で別途指定されていない限り、Linux VDA はデフォルトで PostgreSQL を使用します。Linux ディストリビューションによって提供されるバージョンではなく、カスタムバージョンの PostgreSQL が必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf` を編集して新しいバージョンを反映させ、簡易インストールスクリプト (`ctxinstall.sh`) または MCS スクリプト (`deploymcs.sh`) を実行する前に PostgreSQL サービスを開始する必要があります。
- **/etc/xdl/db.conf** を使用して、PostgreSQL のポート番号を構成することもできます。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールの場合、この手順は省略してください。
2. `deploymcs.sh` を実行する前に `/etc/xdl/db.conf` を編集します。以下に**db.conf**ファイルの例を示します。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite
```

```
#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false
```

```
## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

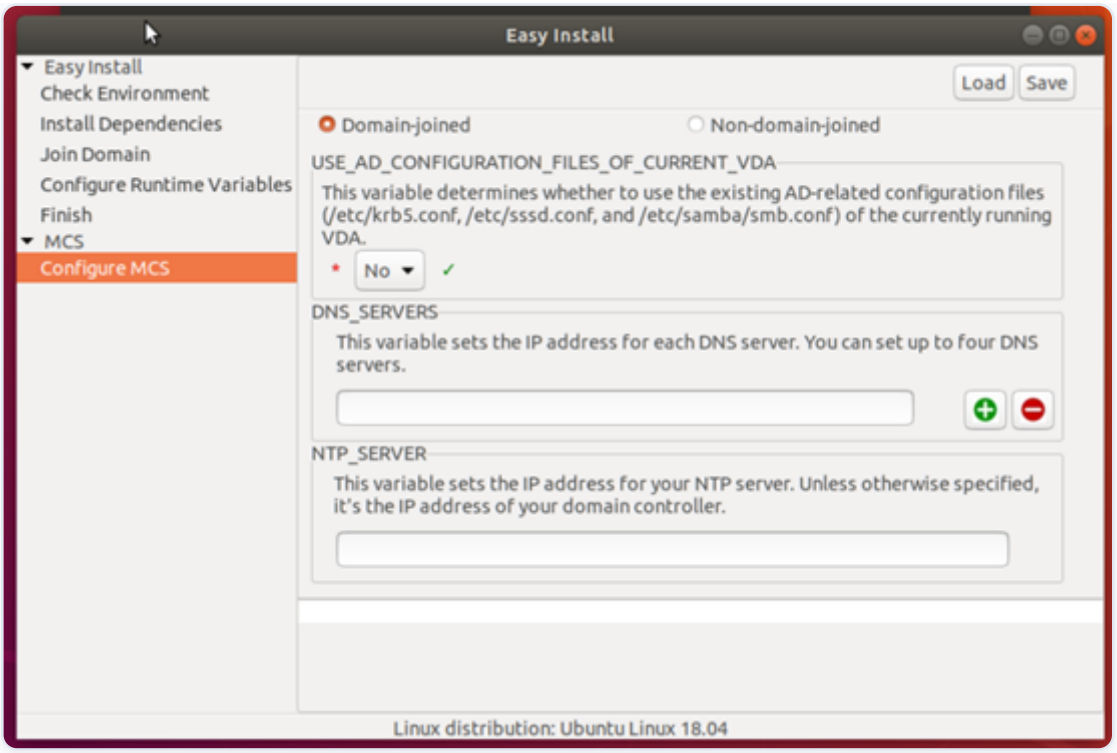
PostgreSQLのカスタムバージョンを使用するには、**DbCustomizePostgreSQL**をtrueに設定します。

ステップ 3i: MCS変数を構成する

MCS変数を構成する方法は2つあります。

- `/etc/xdl/mcs/mcs.conf`ファイルを編集します。
- 簡易インストールGUIを使用します。簡易インストールGUIを開くには、Linux VDAのデスクトップ環境で/opt/Citrix/VDA/sbin/linux-vda-installerコマンドを実行します。

> **注:**
>
> ユーザーは「Advanced Configuration (optional)」ボタンをクリックして、詳細な構成オプションを展開することもできます



ヒント:

Saveをクリックして、指定したパスのローカルファイルに変数の設定を保存します。
Loadをクリックして、指定したファイルから変数の設定を読み込みます。

以下は、ドメインに参加していないシナリオとドメインに参加しているシナリオで構成できるMCS変数です。

- **ドメインに参加していないシナリオの場合**

  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`=path-to-install-dotnet-runtime

  `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

- **ドメインに参加しているシナリオの場合**

- `Use_AD_Configuration_Files_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の同等のファイルと同じになります。ただし、`dns`および`AD_INTEGRATION`変数を構成する必要があります。デフォルト値はNで、これはマスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。現在実行中のVDAをテンプレートVMとして使用するには、値をYに設定します。それ以外の場合は、Nに設定します。
- `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 **lvda** をワークグループ名として使用しますが、**citrix** が正しい選択です。ワークグループ名が正しく設定されていることを確認してください。
- `AD_INTEGRATION`: SSSD、Winbind、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対応表については、この記事の「サポートされているディストリビューション」を参照してください。
- `TRUSTED_DOMAINS`: マルチドメイン環境の場合、信頼されたドメインのスペース区切りリスト（例: 「mycompany1.com mycompany2.com」）を指定します。これにより、/etc/krb5.conf の信頼されたドメインが更新され、LDAP_LISTが指定されていない場合、それらのドメイン内のLDAPサーバーの自動検出が有効になります。この変数はオプションです。

> **注**
>
> SSSDは、単一のActive Directoryフォレスト内の信頼されたドメインのみをサポートします。

- `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。詳細については、「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。
- Linux VDA構成変数:

  `DOTNET_RUNTIME_PATH`=.NET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サーバーのリスト' | '<none\>' `SEARCH_BASE`=検索ベースセット | '<none\>'

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

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

(オプション) ステップ3j: MCSのレジストリ値を書き込むまたは更新する

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。

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

```
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"
```

注

MCS の設定を変更するには、/etc/xdl/ad_join および /etc/xdl/mcs/ の下のファイルを編集できますが、/var/xdl/mcs の下のファイルを編集することは禁止されています。

ステップ 3k: マスターイメージを作成する

1. (SSSD + RHEL 10.x/9.x/8.10 または Rocky Linux 10.x/9.x/8.10 のみ) `update-crypto-policies --set DEFAULT:AD-SUPPORT` コマンドを実行し、その後、テンプレート VM を再起動します。
2. `/etc/xdl/mcs/mcs.conf` を編集して MCS 変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh` を実行します。GUI を使用して MCS 変数を構成する場合は、 をクリックします。GUI で をクリックすると、GUI で設定した変数が `/etc/xdl/mcs/mcs.conf` ファイルで設定した変数を上書きします。
3. (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、またはドメインに参加していないシナリオの場合は、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、`/etc/xdl/ad_join/winbind_krb5.conf.tmpl` および `/etc/xdl/ad_join/winbind_smb.conf.tmpl` テンプレートを更新します。

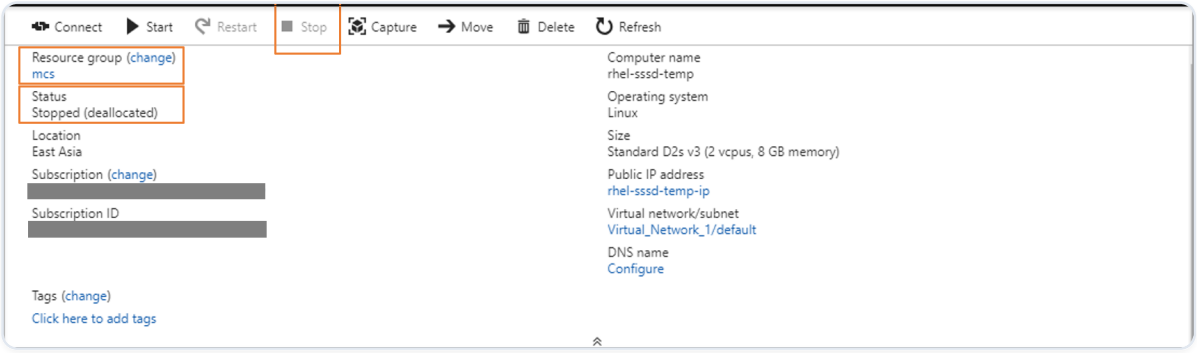
   SSSD ユーザーの場合、`/etc/xdl/ad_join/sssd.conf.tmpl`、`/etc/xdl/ad_join/sssd_krb5.conf.tmpl`、および `/etc/xdl/ad_join/sssd_smb.conf.tmpl` テンプレートを更新します。

   Centrify ユーザーの場合、`/etc/xdl/ad_join/centrify_krb5.conf.tmpl` および `/etc/xdl/ad_join/centrify_smb.conf.tmpl` テンプレートを更新します。

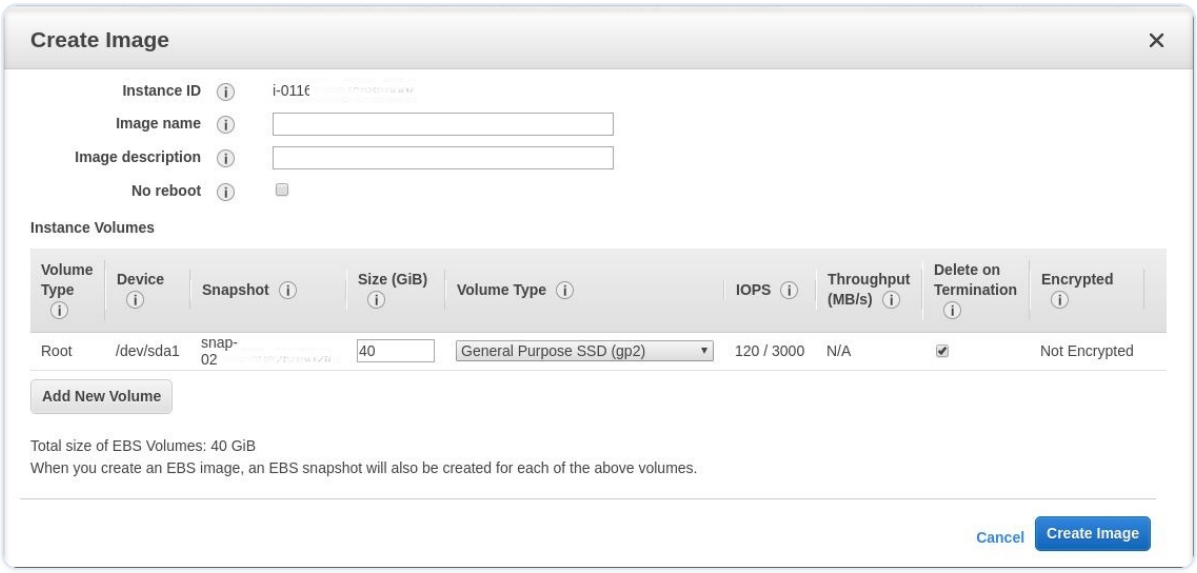
> **注:**
>
> テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、${new_hostname}、および $AD_FQDN などの変数を使用します。

4. 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。

   - **(XenServer、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 VM をシャットダウンします。テンプレート VM の電源状態が 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。その名前は、Azure でマスターイメージを見つけるために必要です。

   - **(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 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 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、 **>** を選択します。必要な情報を入力し、設定を行います。 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て名前を付けます。

> **注:**
>
> Citrix Virtual Apps and Desktopsで使用するには、Acropolisスナップショット名に`XD_`をプレフィックスとして付ける必要があります。必要に応じて、Acropolisコンソールを使用してスナップショットの名前を変更します。スナップショットの名前を変更したら、**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) 手順3l: RHEL 10.x/9.x/8.10およびRocky Linux 10.x/9.x/8.10でのイーサネット接続の構成

GCPでホストされているRHEL 10.x/9.x/8.10およびRocky Linux 10.x/9.x/8.10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に初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動してから、コンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

手順4: 準備済みイメージのマシンカタログを作成する

Citrix StudioまたはWeb Studioで、準備済みイメージを作成し、その準備済みイメージを使用して、次の方法でMCSマシンカタログを作成します。

- ウェブスタジオ
- PowerShellコマンド

現在、AzureおよびVMware仮想化環境で、準備済みイメージマシンカタログを作成できます。

詳細については、準備済みイメージのマシンカタログを作成するを参照してください。

ここではWeb Studioを例として、イメージ定義と初期イメージバージョンを作成するには、次の手順を実行します。

1. Studioから、**イメージ**ノードに移動し、**イメージ定義の作成**をクリックします。**はじめに**ページで**次へ**をクリックします。
2. **イメージ定義**ページで、イメージ定義の**OSタイプ**と**セッションタイプ**を指定します。
3. **［イメージ］ページで、［リソース］**（設定された接続に適用可能なリソースのみが一覧表示されます）と、イメージバージョンを作成するためのテンプレートとして使用するマスターイメージを選択します。**［マシンプロファイルを使用する］**チェックボックスをオンにして、マシンプロファイルを選択できます。

   > **注:**
   >
   > イメージを選択する前に、マスターイメージにVDA 2311以降がインストールされており、MCSIOドライバーがVDAにインストールされていることを確認してください。

4. 「**マシン仕様**」ページで、マシンサイズを選択します。マシンプロファイルを選択した場合、そのマシンプロファイルから派生した仮想CPUカウントが表示され、変更することはできません。マシンプロファイルを選択しない場合、マスターイメージから派生したメモリサイズのみが表示されます。
5. **NIC** ページで、準備イメージの NIC を選択または追加します。各 NIC について、関連付けられた仮想ネットワークを選択します。マシンプロファイルを選択しない場合、マスターイメージに関連付けられた NIC がデフォルトで選択されます。マシンプロファイルを選択した場合、NIC はマシンプロファイルから派生し、その数は変更できません。
6. 「**Version Description**」ページで、作成された初期イメージバージョンの説明を入力します。

7. 「**概要**」ページで、イメージ定義と作成された初期イメージバージョンの詳細を確認します。イメージ定義の名前と説明を入力します。「**完了**」をクリックします。

イメージバージョンを作成する

イメージバージョンを使用すると、特定のイメージに対するさまざまなイテレーションや更新を管理できます。この機能により、さまざまな目的のためにイメージの複数のバージョンを維持できます。

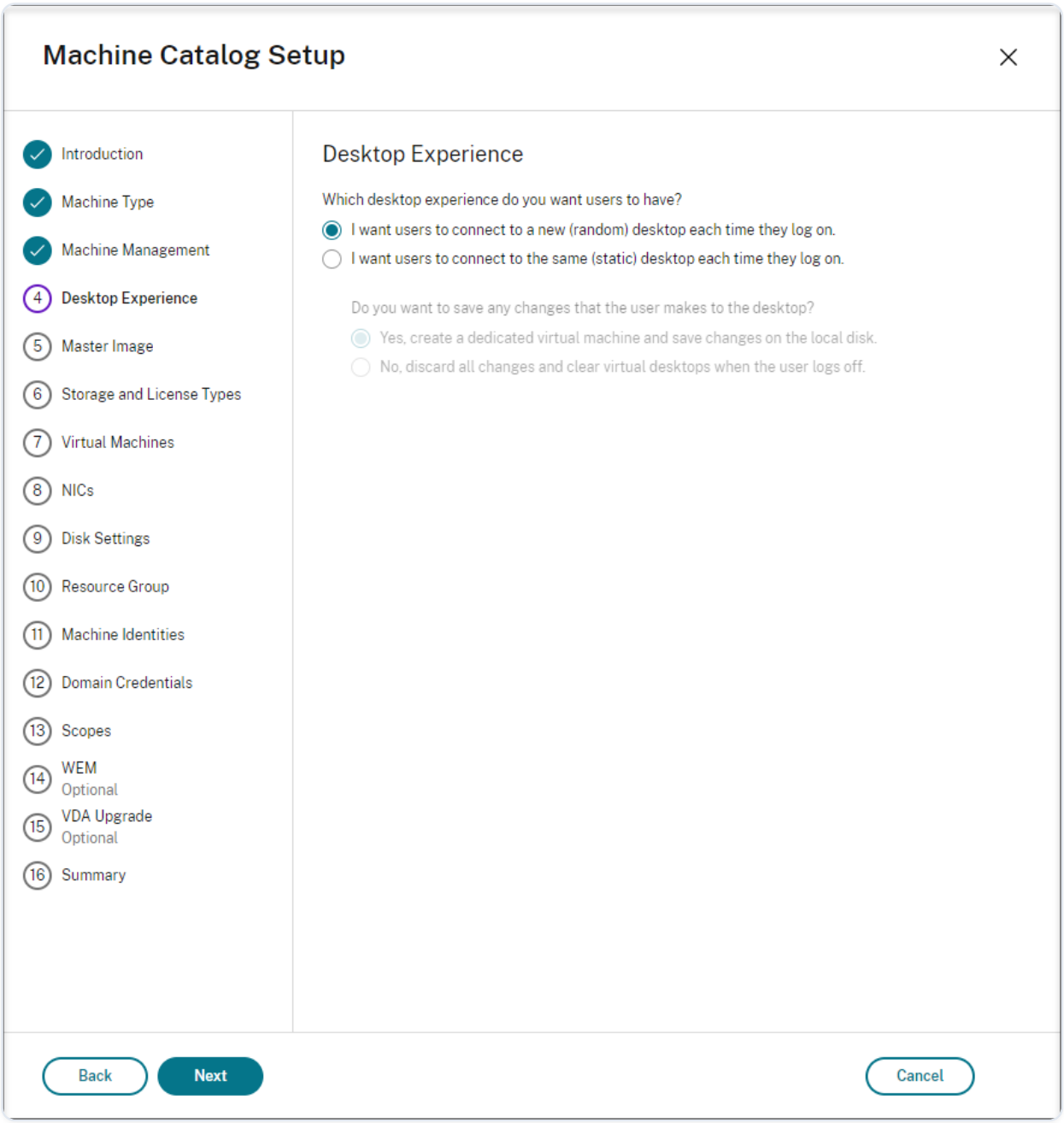
初期イメージバージョンからイメージバージョンを作成するには、次の手順を実行します。

1. 「**Images**」ノードに移動し、イメージバージョンまたはイメージ定義を選択して、「**Create Image Version**」をクリックします。
2. **イメージ** ページで、同じホスティング接続内のホスティングユニットを変更し、そのイメージバージョン用のマスターイメージとマシンプロファイルを再選択できます。
3. イメージバージョンの構成を初期構成イメージバージョンと異なるものにする場合は、「**Create Image Version**」ダイアログの「**Machine Specification**」ページと「**NICs**」ページで設定を構成します。
4. イメージバージョンの説明を追加します。「**Finish**」をクリックします。

手順5: マシンカタログを作成する

Citrix StudioまたはWeb Studioで、マシンカタログを作成し、カタログで作成するVMの数を指定します。マシンカタログを作成するときは、手順4で作成した準備済みイメージ（推奨）/マスターイメージを選択し、次の点を考慮してください。

- Nutanix固有の**コンテナ**ページで、以前にテンプレートVMに指定したコンテナを選択します。
- **シングルセッションOS**マシンを含むカタログを作成すると、**デスクトップエクスペリエンス**ページが表示され、ユーザーがログオンするたびに何が起こるかを決定できます。



デスクトップエクスペリエンスページで、次のいずれかを選択します。

- ユーザーはログオンするたびに新しい（ランダムな）デスクトップに接続します。
- ユーザーはログオンするたびに同じ（静的な）デスクトップに接続します。

最初のオプションを選択した場合、ユーザーがデスクトップに加えた変更は破棄されます（非永続的）。

2番目のオプションを選択し、MCSを使用してマシンをプロビジョニングしている場合は、デスクトップに対するユーザーの変更がどのように処理されるかを構成できます。

- ユーザーの変更をローカルディスク上のデスクトップに保存します（永続的）。

    - ユーザーがログオフしたときにユーザーの変更を破棄し、仮想デスクトップをクリアします（非永続的）。ユーザーパーソナライゼーションレイヤーを使用している場合は、このオプションを選択します。

- 永続マシンを含むMCSカタログのマスターイメージを更新する場合、カタログに追加された新しいマシンは更新されたイメージを使用します。既存のマシンは元のマスターイメージを引き続き使用します。

詳細については、(/ja-jp/citrix-virtual-apps-desktops/install-configure/machine-catalogs-create.html)ドキュメントのCitrix Virtual Apps and Desktopsおよび(/ja-jp/citrix-daas/install-configure/machine-catalogs-create.html)ドキュメントのCitrix DaaSにあるマシンカタログの作成を参照してください。

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

ステップ6：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。

詳細については、(/ja-jp/citrix-daas/install-configure/delivery-groups-create.html)ドキュメントのCitrix Virtual Apps and Desktopsおよび(/ja-jp/citrix-daas/install-configure/delivery-groups-create.html)ドキュメントのCitrix DaaSにあるデリバリーグループの作成を参照してください。

> **注:**
>
> MCSを使用して作成したVMは、Citrix Cloud Connectorsに登録できず、**未登録**と表示される場合があります。この問題は、AzureでVMをホストし、Samba Winbindを使用してADドメインに参加している場合に発生します。この問題を回避するには、次の手順を実行します。

1. ADSI Editコンソールに移動し、未登録のVMを選択し、そのマシンアカウントの**msDS-SupportedEncryptionTypes**属性を編集します。
2. VMで**ctxjproxy**サービスと**ctxvda**サービスを再起動します。VMのステータスが**登録済み**に変わった場合は、手順3から5に進みます。
3. テンプレートVMで**/var/xdl/mcs/ad_join.sh**ファイルを開きます。
4. **/var/xdl/mcs/ad_join.sh** ファイル内の次の行の後に、**net ads enctypes set $NEW_HOSTNAME\$ <Decimal value of encryption type attribute, for example, 28\> -U $NEW_HOSTNAME\$ -P password** の行を追加します。

   ```
   if [ "$AD_INTEGRATION" == "winbind" ]; then

                   join_domain_samba

                   restart_service winbind /usr/bin/systemctl
   ```

5. 新しいスナップショットを取得し、新しいテンプレートを使用してVMを作成します。

MCS を使用して Linux VDA をアップグレードする

MCS を使用して Linux VDA をアップグレードするには、次の手順を実行します。

1. Linux VDA を現在のリリースにアップグレードする前に、(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html#net-requirements)システム要件に従って .NET 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 をアップグレードします。

> **注:**
>
> - (/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/administration/vda-upgrades.html)Azure を介した Linux VDA の自己更新機能を使用して、自動ソフトウェア更新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
```

- 2407 リリース以降、Linux VDA はアップグレード中に構成ファイルを処理するためにパッケージマネージャーの **rpm** または **dpkg** に委任します。以下に、rpm と dpkg が構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを **.rpmnew** 拡張子で保存します。
  - **dpkg**: 続行方法の選択肢を対話形式で求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを **.dpkg-new** または **.dpkg-dist** として保存しながら Linux VDA をサイレントアップグレードするには、次のコマンドを使用します。

    ```
    dpkg --force-confold -i package.deb  # Always keep your
    version, then save new package's version as *.dpkg-new or
    *.dpkg-dist
    ```

RHEL および 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:

RHEL 9.x および Rocky Linux 9.x で Linux VDA をアップグレードする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

SUSE ディストリビューションの場合:

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

Ubuntu/Debian ディストリビューションの場合:

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



sudo apt-get install -f
```

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てください。マシンを手動で再起動しないでください。

マシンアカウントのパスワード更新を自動化する

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

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

   `UPDATE_MACHINE_PW="Y"`

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)
```

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。

Machine Creation Services™ (MCS) を使用してFASが有効なドメイン参加済みLinux VDAを作成する

> **重要:**
>
> 2212リリース以降、以下の重要な変更点があります。
>
> - /etc/xdl/mcs/mcs.conf ファイル内または簡易インストールGUI上の**AD_INTEGRATION** 変数には、デフォルト値がなくなりました。必要に応じて値を設定する必要があります。詳細については、この記事の「手順3j: MCS変数を構成する」セクションを参照してください。
> - /etc/xdl/mcs/mcs.conf 内の **UPDATE_MACHINE_PW** エントリの有効な値は、**enabled** または **disabled** ではなく、**Y** または **N** になりました。詳細については、この記事の「マシンアカウントパスワードの自動更新」セクションを参照してください。

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	PBIS
デビアン 13.x/12.15	はい	はい	はい
RHEL 10.x/9.x	はい	はい	いいえ

	ウィンバインド	SSSD	PBIS
RHEL 8.10	はい	はい	はい
ロッキー リナックス 10.x/9.x	はい	はい	いいえ
ロッキー リナックス 8.10	はい	はい	いいえ
ウブントゥ 26.04/24.04/22.04	はい	はい	いいえ
SUSE 15.7	はい	はい	いいえ

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

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

サポートされているハイパーバイザー

- AWS
- GCP
- マイクロソフト アジュール
- シトリックス ゼノサーバー
- 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ハイパーバイザー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、予期しない結果が生じる可能性があります。

MCSを使用してFASが有効なLinux VDAを作成する

このセクションでは、MCSを使用してLinux VDAを作成し、テンプレートVMでマスターイメージを準備する際にFASを有効にする手順について説明します。テンプレートVMでFASが有効になっていない場合は、MCSで作成されたVMでFASを有効にするを参照して、後でMCSで作成された各VMで有効にすることができます。

考慮事項

- 2203リリース以降、Citrix Virtual Apps and Desktops™およびCitrix DaaS（旧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 Hybrid Multi-Cloud（HMC）ライセンスが必要です。
- 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サポートポータルで入手できるNutanix Acropolis MCSプラグインインストールガイドを参照してください。

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

Citrix Virtual Apps™ and Desktopsをインストールした後、Delivery Controller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Delivery Controller向けNutanix AHVプラグイン(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/nutanix-plug-in-for-cvad.png)

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

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

ステップ1c: プラグインのインストール後に次の手順を完了します

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`にNutanix Acropolisフォルダーが作成されていることを確認します。
- `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"`コマンドを実行します。
- オンプレミスのDelivery ControllerでCitrix Host、Citrix Broker、およびCitrix Machine Creation Servicesを再起動するか、Citrix Cloud ConnectorでCitrix RemoteHCLServer Serviceを再起動します。

> **ヒント:**
>
> Nutanix AHVプラグインをインストールまたは更新するときは、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、XenServer® (旧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。

> **注:**
>
> オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。

> 詳細については、Citrix Virtual Apps and Desktopsの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の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
- Citrix StudioでAWSへのホスト接続を作成する
- Citrix StudioでXenServerへのホスト接続を作成する

- Citrix Studio で GCP へのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2607-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/2607-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/2607-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. オンプレミスの Delivery Controller の場合、オンプレミスの Citrix Studio で **> >** を選択して、ホスト接続を作成します。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、ホスト接続を作成します。
2. ウィザードで、接続の種類として Microsoft Azure を選択します。
3. 接続の種類として Microsoft Azure を選択します。
4. ウィザードのページに従って操作します。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 ページに進みます。詳細については、MCS を使用してドメインに参加していない Linux VDA を作成するの記事の「**手順 2：ホスト接続を作成する**」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. オンプレミスの Delivery Controller の場合、オンプレミスの Citrix Studio で **> >** を選択して、ホスト接続を作成します。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、ホスト接続を作成します。
2. ウィザードで、接続の種類として を選択します。

   たとえば、オンプレミスの Citrix Studio では、次のようになります。

   Amazon EC2 の選択(/en-us/linux-virtual-delivery-agent/2210/media/choose-amazon-ec2.png)
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/2210/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キーを作成できます。アクセスキーを作成するには、次の手順を実行します。

a. AWS サービスにサインインします。
b. Identity and Access Management (IAM) コンソールに移動します。 1 左側のナビゲーションペインで、 を選択します。
c. ターゲットユーザーを選択し、下にスクロールして タブを選択します。
d. 下にスクロールして をクリックします。新しいウィンドウが表示されます。
e. をクリックし、アクセスキーを安全な場所に保存します。

4. ウィザードのページを順に進めます。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、 を選択して ページに進みます。

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

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で **> >** を選択してホスト接続を作成します。クラウド Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで **> >** を選択してホスト接続を作成します。
2. ウィザードで、 フィールドで XenServer (旧称 Citrix Hypervisor) を選択します。
3. 接続アドレス (XenServer URL) と資格情報を入力します。
4. 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

(/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、次の手順を完了して GCP へのホスト接続を作成します。

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で **> >** を選択してホスト接続を作成します。クラウド Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで **> >** を選択してホスト接続を作成します。
2. ウィザードで、接続の種類として を選択します。

   例えば、Citrix CloudのWebベースのStudioコンソールで:

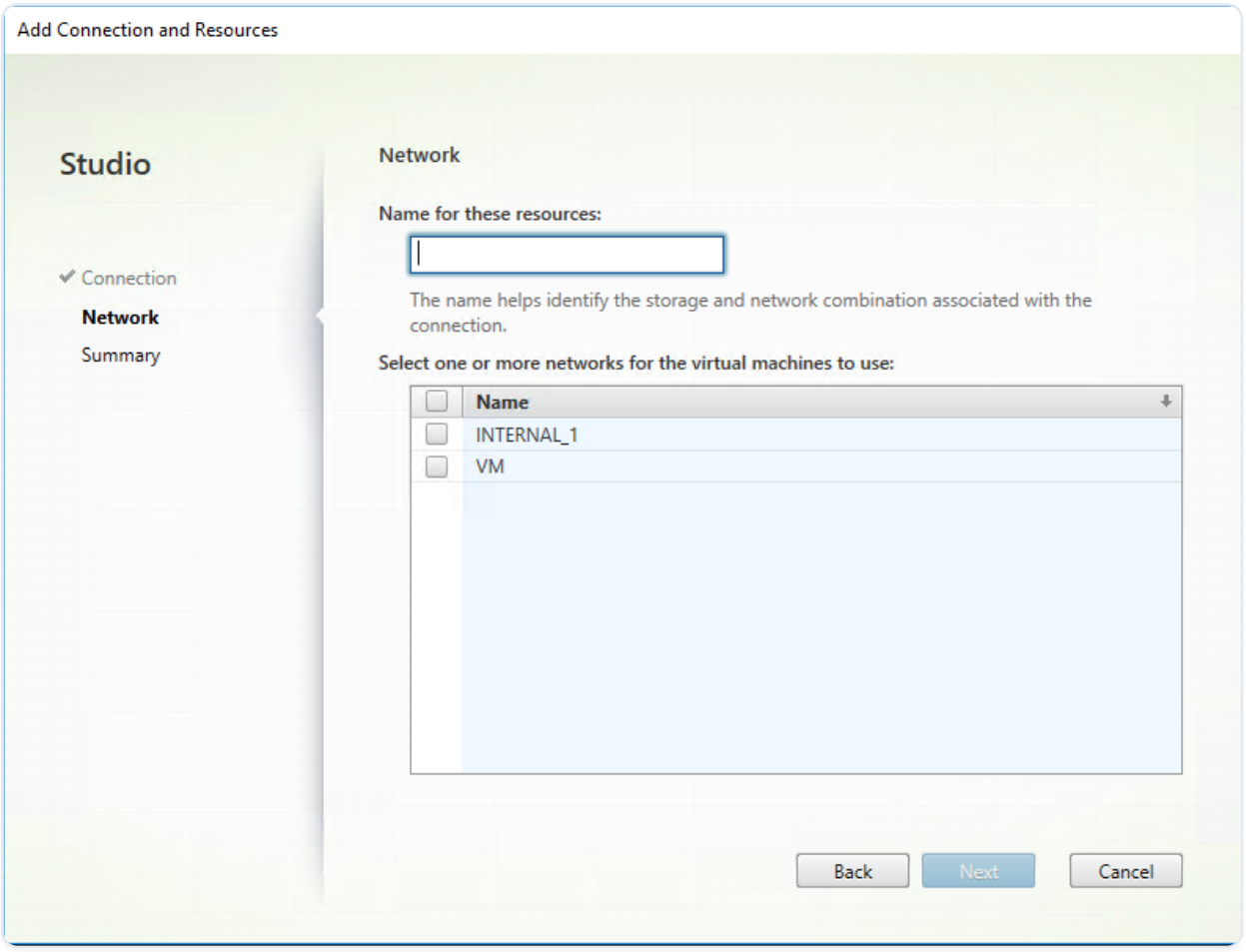
   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/add-conn-connection.png)
3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードの指示に従ってページを進めます。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、**概要**ページに到達するまで**次へ**を選択しま

す。詳細については、MCSを使用してドメインに参加していないLinux VDAを作成するの記事の「**ステップ2：ホスト接続を作成する**」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**構成 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。
2. **接続とリソースの追加**ウィザードの**接続**ページで、接続の種類としてNutanix AHVを選択し、ハイパーバイザーのアドレス、資格情報、および接続名を指定します。**ネットワーク**ページで、ユニットのネットワークを選択します。

   例えば、オンプレミスのCitrix Studioで:

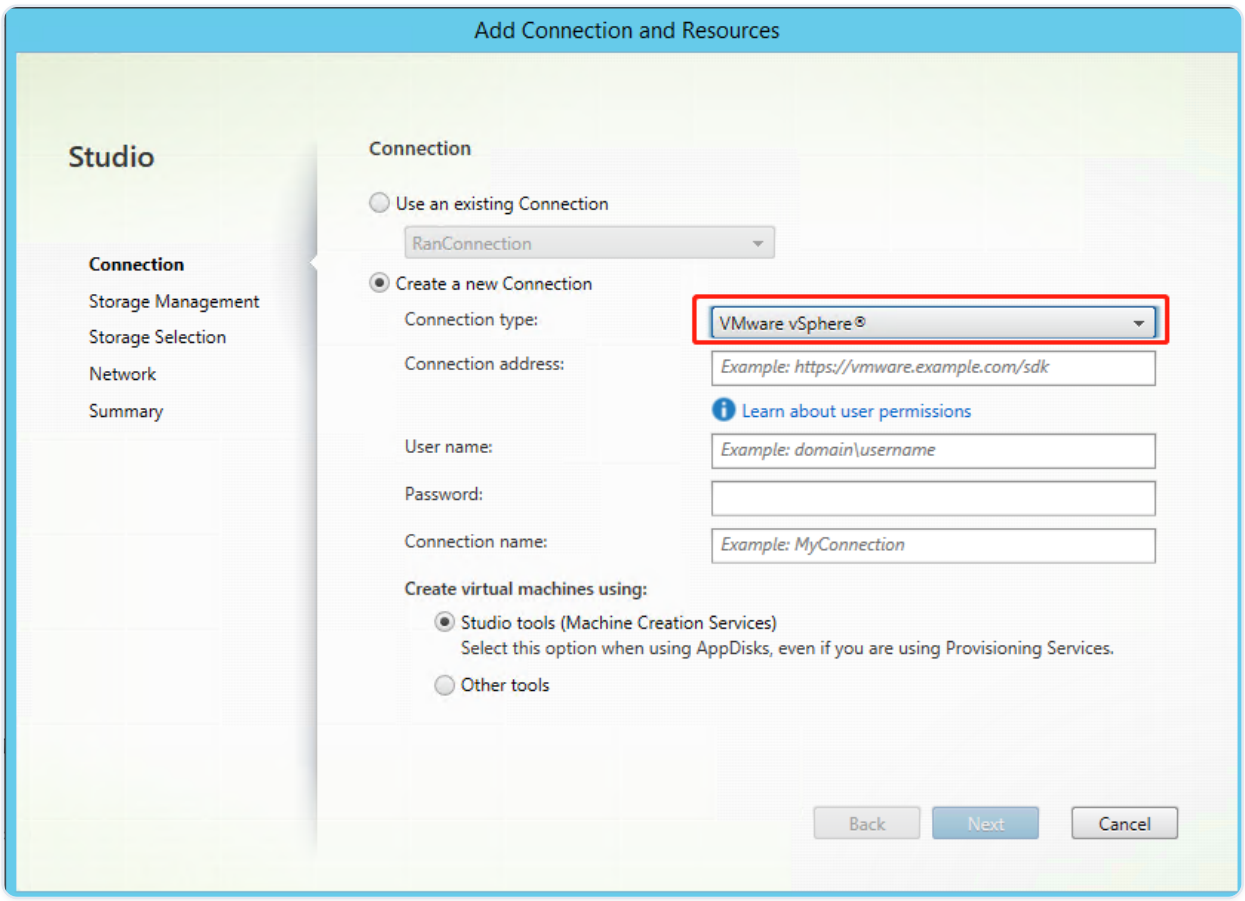


Citrix StudioでVMwareへのホスト接続を作成する

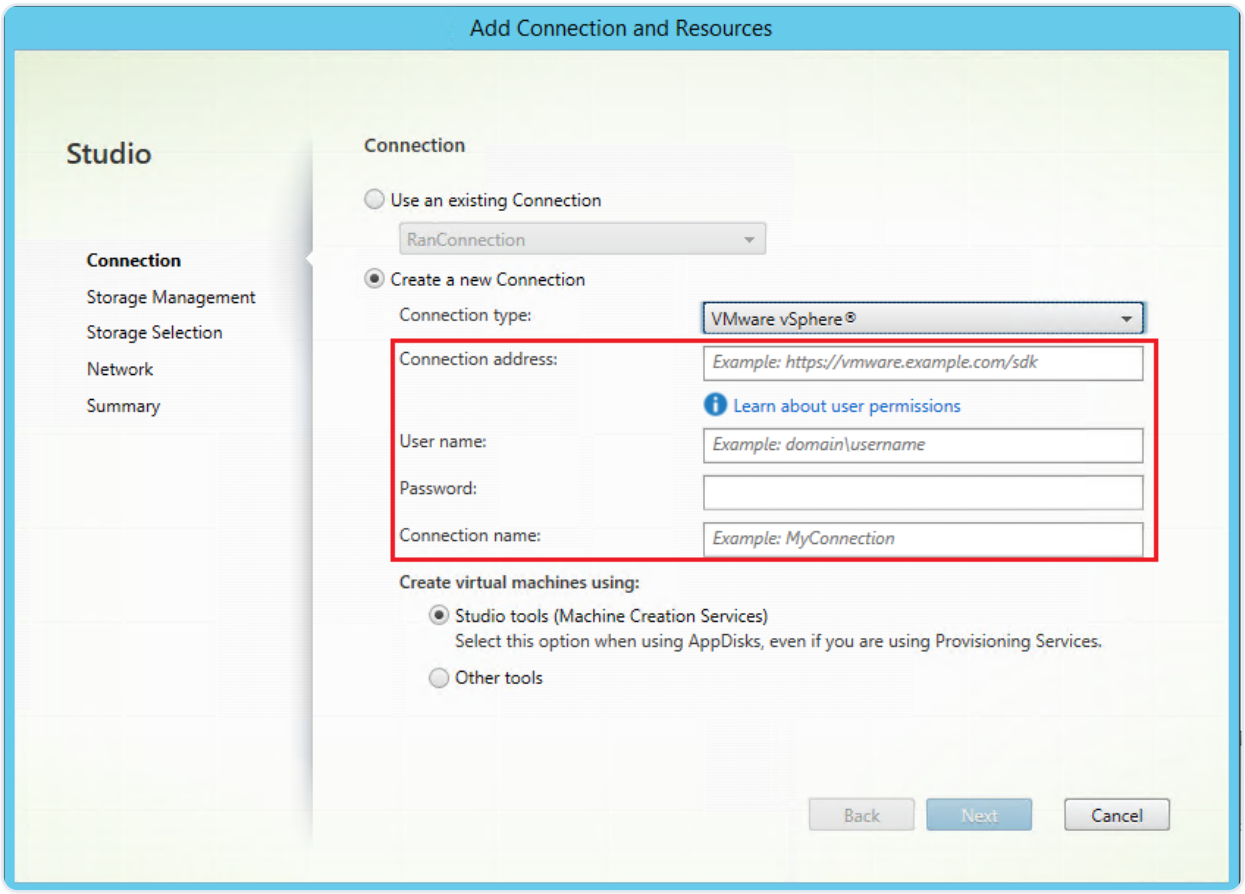
1. vSphere環境にvCenter Serverをインストールします。詳細については、VMware vSphereを参照してください。

2. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択してホスト接続を作成します。
3. 接続の種類としてVMware vSphereを選択します。

   例えば、オンプレミスのCitrix Studioで:



4. VMwareアカウントの接続アドレス（vCenter Server URL）、資格情報、および接続名を入力します。



ステップ3：マスターイメージを準備する

(XenServer のみ) ステップ 3a: XenServer 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 XenServer 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、次の操作はできません。

- VMをきれいにシャットダウン、再起動、または一時停止します。
- XenCenterでVMのパフォーマンスデータを表示します。
- 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します。
- 実行中のLinux VMでvCPUの数を調整します。

1. 使用中のハイパーバイザーのバージョンに基づいて、XenServerダウンロードページまたはCitrix HypervisorダウンロードページからXenServer VM Tools for Linuxファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz` ファイルを、お使いの Linux VM または Linux VM がアクセスできる共有ドライブにコピーします。

3. tar ファイルのコンテンツを抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. 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/SUSEの場合：

   ```
   sudo rpm -i <extract-directory>/xe-guest-utilities_{package-version}
   _x86.64.rpm
   ```

   Ubuntu/Debianの場合:

   ```
   sudo dpkg -i <extract-directory>/xe-guest-utilities_{package-version}
   _amd64.deb
   ```

5. 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。XenServer 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** と表示されます。

ステップ 3b: AWS、Azure、GCP 上の SUSE 15.7 の構成を確認する

AWS、Azure、GCP 上の SUSE 15.7 の場合、以下を確認してください。

- **libstdc++6** バージョン 12 以降を使用していること。
- **/etc/sysconfig/windowmanager** 内の **Default_WM** パラメーターが **"gnome"** に設定されていること。

ステップ 3c: GCP 上の Ubuntu で RDNS を無効にする

テンプレート VM で、/etc/krb5.conf の の下に **rdns = false** の行を追加します。

ステップ 3d: テンプレート VM に .NET をインストールする

Linux VDA パッケージをインストールする前に、.NET を システム要件に従って インストールします。

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

.NET のインストール後、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、Linux VDA をインストールします。

RHEL/Rocky Linux の場合:

> **注:**

> RHEL 9.x および Rocky Linux 9.x に Linux VDA をインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

```
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 3f: (RHEL 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 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 3g: (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 3h: VM を Windows ドメインに参加させる

VM を Windows ドメインに参加させます。このプロセスには、時間と労力を節約し、手動インストールと比較してエラーが発生しにくい簡単なインストールスクリプト (ctxinstall.sh) を使用することをお勧めします。詳細については、「ステップ 8: 環境と VDA を構成してインストールを完了するための簡易インストールを実行する」を参照してください。手動インストールを希望する場合は、「Linux VDA を手動でインストールする」の記事のステップ 1 から 3 を参照してください。

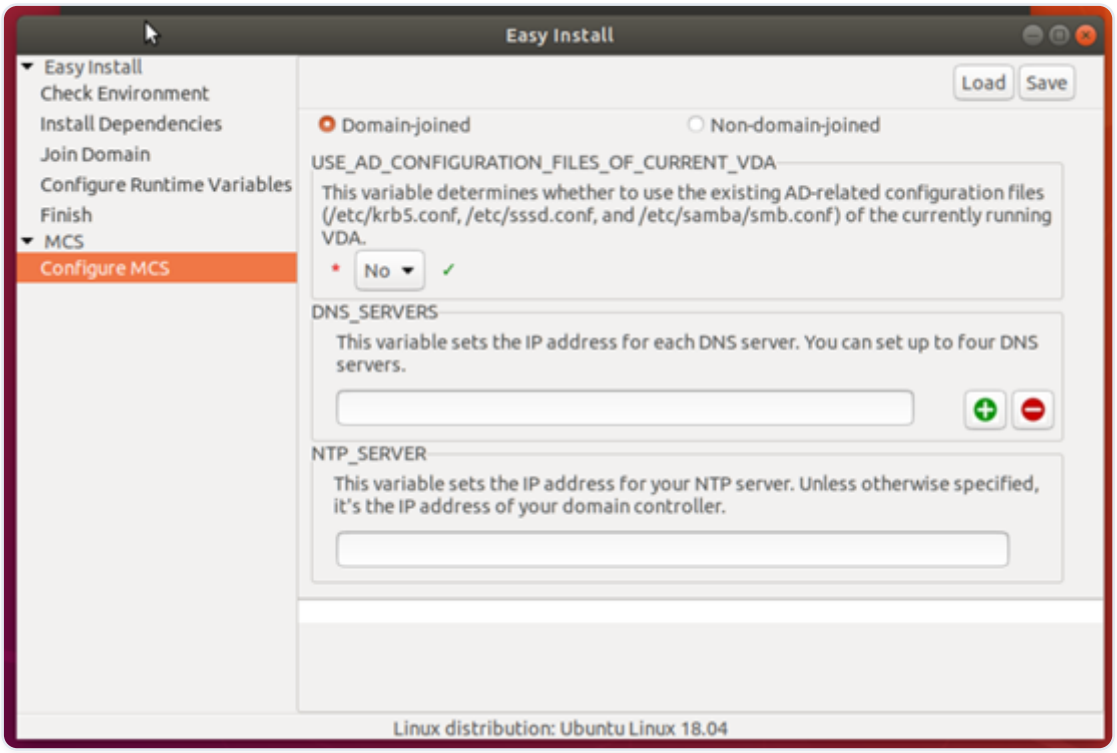
ステップ 3i: VM で FAS を構成する

詳細な手順については、「Linux VDA で FAS を構成する」を参照してください。

ステップ 3j: MCS 変数を構成する

MCS 変数を構成する方法は 2 つあります。

- `/etc/xdl/mcs/mcs.conf` ファイルを編集します。
- 簡易インストールGUIを使用します。簡易インストールGUIを開くには、Linux 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。



> ヒント:
>
> **Save**をクリックして、指定したパスのローカルファイルに変数の設定を保存します。
> **Load**をクリックして、指定したファイルから変数の設定を読み込みます。

以下は、FASが有効なシナリオで構成できるMCS変数です。

```
-  `Use_AD_Configuration_Files_Of_Current_VDA`: Determines whether to use the existing AD-related configuration files (/etc/krb5.conf, /etc/sssd.conf, and /etc/samba/smb.conf) of the currently running VDA. Set the value to Y when FAS is enabled.



-  `dns`: Sets the IP address for each DNS server. You can set up to four DNS servers.



-  `NTP_SERVER`: Sets the IP address for your NTP server. Unless otherwise specified, it's the IP address of your domain controller.
```

- `WORKGROUP`: Sets the workgroup name to the NetBIOS name (case-sensitive) that you configured in AD. Otherwise, MCS uses the part of the domain name that immediately follows the machine hostname as the workgroup name. For example, if the machine account is **user1.lvda.citrix.com**, MCS uses **lvda** as the workgroup name while **citrix** is the correct choice. Ensure that you set the workgroup name correctly.

- `AD_INTEGRATION`: Sets SSSD, Winbind, or PBIS. For a matrix of the Linux distributions and domain joining methods that MSC supports, see [Supported distributions](#サポートされているディストリビューション) in this article.

- `PBIS_DOWNLOAD_PATH`: Sets the path for downloading the PBIS package. The value takes effect only when you set the `AD_INTEGRATION` variable to PBIS.

- `UPDATE_MACHINE_PW`: Enables or disables automating machine account password updates. For more information, see [Automate machine account password updates](https://docs.citrix.com/us/en/linux-virtual-delivery-agent/2607-ltsr/installation-overview/create-linux-vdas-using-mcs.html#マシンアカウントのパスワード更新を自動化する).

- Linux VDA configuration variables:

    `DOTNET_RUNTIME_PATH`=path-to-install-dotnet-runtime

    `DESKTOP_ENVIRONMENT`=gnome | mate

    `SUPPORT_DDC_AS_CNAME`=Y | N

    `VDA_PORT`=port-number

    `REGISTER_SERVICE`=Y | N

```
    `ADD_FIREWALL_RULES`=Y | N

    `HDX_3D_PRO`=Y | N

    `VDI_MODE`=Y | N

    `SITE_NAME`=dns-site-name | '&lt;none\>'

    `LDAP_LIST`='list-ldap-servers' | '&lt;none\>'

    `SEARCH_BASE`=search-base-set | '&lt;none\>'

    `START_SERVICE`=Y | N

    `TELEMETRY_SOCKET_PORT`=port-number

    `TELEMETRY_PORT`=port-number
```

手順3k：MCSのレジストリ値を書き込むか更新する

FASサーバーアドレスを設定するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。

```
create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

注

MCSの設定を変更する場合、/etc/xdl/ad_joinおよび/etc/xdl/mcs/配下のファイルを編集することは許可されていますが、/var/xdl/mcs配下のファイルを編集することは禁止されています。

手順3l：マスターイメージを作成する

1. （SSSD + RHELおよびRocky Linuxのみ）`AD-SUPPORT`暗号化ポリシーを有効にしてから、テンプレートVMを再起動します。

    a. RHELおよびRocky Linux 10.x/9.xの場合、`AD-SUPPORT`暗号化ポリシーを有効にします。

```
`update-crypto-policies --set DEFAULT:AD-SUPPORT`
```

Active DirectoryがKerberos認証中にSHA-1を使用する場合、SHA-1のサポートも有効にする必要があります。

RHELおよびRocky Linux 9.xでは、組み込みの `SHA1` サブポリシーを有効にします。

```
`update-crypto-policies --set DEFAULT:AD-SUPPORT:SHA1`
```

RHELおよびRocky Linux 10.xでは、組み込みの `SHA1` サブポリシーは提供されなくなりました。可能な場合は、Active Directory CAおよびドメインコントローラー証明書をSHA-256で再発行することをお勧めします。上記のサブポリシーは、SHA-1を継続して使用する必要がある環境向けです。`AD-SUPPORT-LEGACY` は代替手段ではありません。Kerberos暗号化タイプを広げるだけで、SHA-1署名を再度有効にするものではありません。

SHA-1を使い続けるには、独自の同等のサブポリシーを作成して適用します。

```
sudo tee /etc/crypto-policies/policies/modules/AD-SHA1.pmod > /dev/
null <<'EOF'

hash = SHA1+

sign = ECDSA-SHA1+ RSA-PSS-SHA1+ RSA-SHA1+

sha1_in_certs = 1

EOF

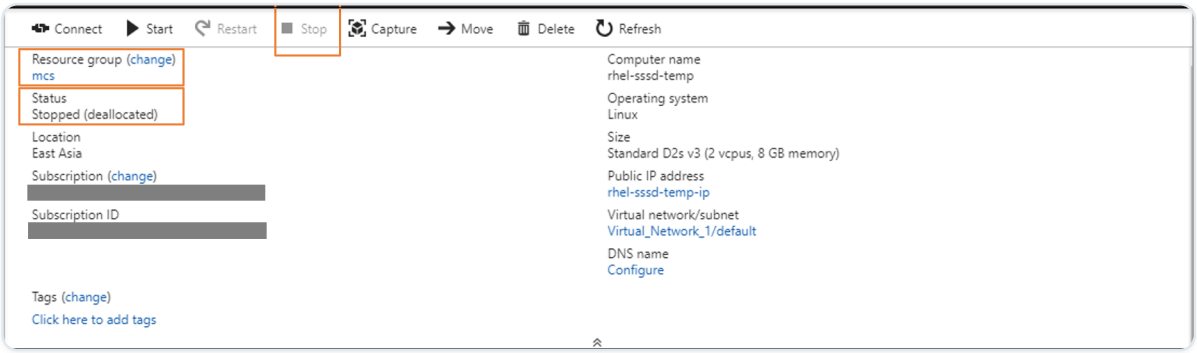
sudo update-crypto-policies --set DEFAULT:AD-SUPPORT:AD-SHA1
```

新しいポリシーは、その後起動されたアプリケーションに適用されます。すべての場所に適用するには再起動してください。

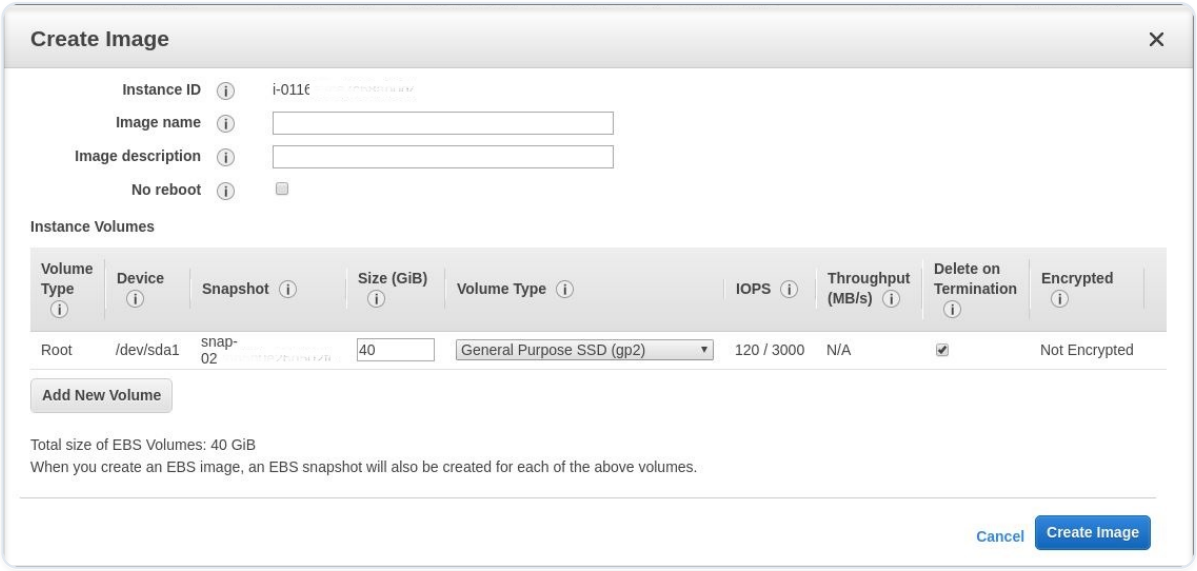
2. MCS 変数を `/etc/xdl/mcs/mcs.conf` を編集して構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh` を実行します。GUI を使用して MCS 変数を構成する場合は、**Deploy** をクリックします。GUI で **Deploy** をクリックすると、GUI で設定した変数が `/etc/xdl/mcs/mcs.conf` ファイルで設定した変数を上書きします。

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。

   - **(XenServer、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 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 **Stopped (deallocated)** であることを確認してください。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。



   - **(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 **Stopped** であることを確認してください。テンプレート VM を右クリックし、**Image > Create Image** を選択します。必要な情報を入力し、設定を行います。**Create Image** 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 で使用するために、Acropolis スナップショット名には `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てください。スナップショットの名前を変更した後、**Create Catalog** 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 3m: RHEL 10.x/9.x/8.10 および Rocky Linux 10.x/9.x/8.10 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 10.x/9.x/8.10 および Rocky Linux 10.x/9.x/8.10 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 に初めてログオンするときにルートパスワードを設定し、ルートとして VM にログオンできることを確認してください。その後、VM を再起動した後、コンソールで次のコマンドを実行します。

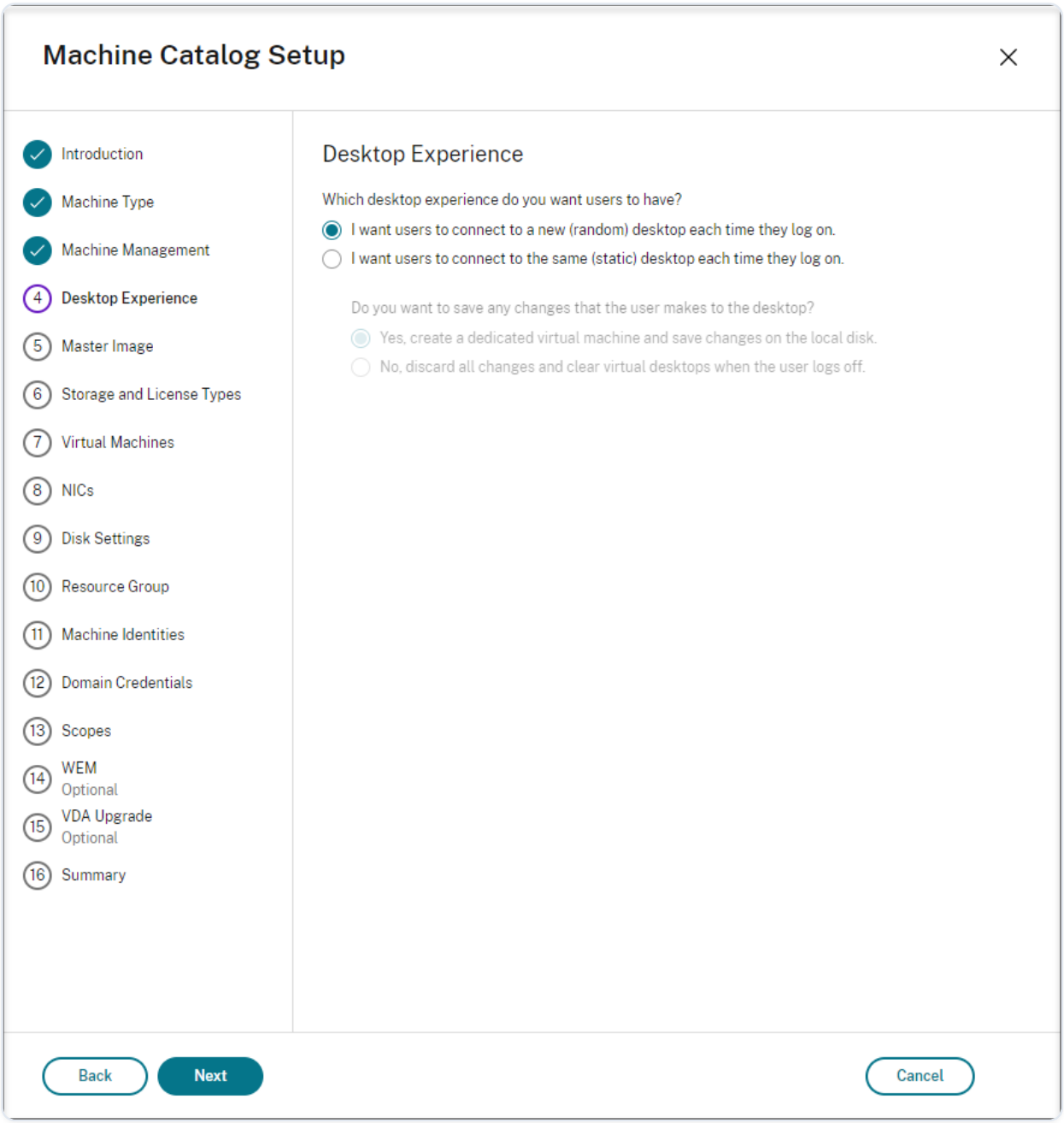
```
nmcli dev connect eth0

systemctl restart NetworkManager
```

ステップ 4: マシンカタログを作成する

Citrix Studio または Web Studio で、マシンカタログを作成し、カタログに作成する VM の数を指定します。マシンカタログを作成する際は、マスターイメージを選択し、次の点を考慮してください。

- Nutanix 固有の **Container** ページで、以前にテンプレート VM 用に指定したコンテナを選択します。
- **シングルセッションOS**マシンを含むカタログを作成すると、**デスクトップエクスペリエンス** ページが表示され、ユーザーがログオンするたびに何が起こるかを決定できます。



デスクトップエクスペリエンスページで、次のいずれかを選択します。

- ユーザーはログオンするたびに新しい（ランダムな）デスクトップに接続します。
- ユーザーはログオンするたびに同じ（静的な）デスクトップに接続します。

最初のオプションを選択した場合、ユーザーがデスクトップに加えた変更は破棄されます（非永続的）。

2番目のオプションを選択し、MCSを使用してマシンをプロビジョニングしている場合は、デスクトップに対するユーザーの変更がどのように処理されるかを構成できます。

- ユーザーの変更をローカルディスク上のデスクトップに保存します（永続的）。

    - ユーザーがログオフしたときに、ユーザーの変更を破棄し、仮想デスクトップをクリアします（非永続的）。ユーザーパーソナライゼーションレイヤーを使用している場合は、このオプションを選択します。

- 永続マシンを含むMCSカタログのマスターイメージを更新する場合、カタログに追加された新しいマシンは更新されたイメージを使用します。既存のマシンは元のマスターイメージを引き続き使用します。

詳細については、Citrix Virtual Apps and DesktopsドキュメントおよびCitrix DaaSドキュメントの「マシンカタログの作成」を参照してください。

> **注:**
>
> Nutanix環境の場合、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。どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。

詳細については、Citrix Virtual Apps and DesktopsドキュメントおよびCitrix DaaSドキュメントの「デリバリーグループの作成」を参照してください。

> **注:**
>
> MCSを使用して作成したVMは、Citrix Cloud Connectorsに登録できず、**未登録**と表示される場合があります。この問題は、AzureでVMをホストし、Samba Winbindを使用してADドメインに参加している場合に発生します。この問題を回避するには、次の手順を実行します。
>
> 1. ADSI Editコンソールに移動し、未登録のVMを選択して、そのマシンアカウントの**msDS-SupportedEncryptionTypes**属性を編集します。
> 2. VMで**ctxjproxy**サービスと**ctxvda**サービスを再起動します。VMのステータスが**登録済み**に変わった場合は、手順3から5に進みます。

3. テンプレートVMで **/var/xdl/mcs/ad_join.sh** ファイルを開きます。
4. **/var/xdl/mcs/ad_join.sh** ファイル内の以下の行の後に、**net ads enctypes set $NEW_HOSTNAME\$ <Decimal value of encryption type attribute, for example, 28\> -U $NEW_HOSTNAME\$ -P password** の行を追加します。

```
if [ "$AD_INTEGRATION" == "winbind" ]; then

                join_domain_samba

                restart_service winbind /usr/bin/systemctl
```

5. 新しいスナップショットを取得し、新しいテンプレートを使用してVMを作成します。

マシンアカウントのパスワード更新を自動化する

マシンアカウントのパスワードは、デフォルトで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、次のエントリを /etc/xdl/mcs/mcs.conf に追加します。

   `UPDATE_MACHINE_PW="Y"`

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、毎週日曜日の午前2時30分にマシンアカウントのパスワードが更新されます。

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)`

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

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、MCSで作成された各VMでFASを有効にできます。

MCSで作成されたVMでFASを有効にするには、次の手順を実行します。

1. /etc/xdl/mcs/mcs.conf で変数を設定します。

   > **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_AD_Configuration_Files_Of_Current_VDA` の値をYに設定します。
   b. 必要なその他の変数を、`VDI_MODE` のように設定します。

2. FASサーバーアドレスを設定するために、次のコマンドラインを **/etc/xdl/mcs/mcs_local_setting.reg** ファイルに追加します。

   ```
   create -k
   "HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
   Service" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
   ```

3. ルートCA証明書をインポートします。

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

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

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 24.04/22.04
- RHEL 9.8/9.6/8.10
- ロッキー リナックス 9.8/9.6/8.10

重要:

- 該当する Linux ターゲットおよび PVS Server LTSR のバージョンについては、以下の表を参照してください。
- 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ます。
- いずれのパーティションも **btrfs** で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という本質的な問題があります。**GRUB** は **GRand Unified Bootloader** の略です。

PVS LINUX ターゲットバージョン	PVS SERVER バージョン	起動モードのサポート
2511より前	2203 LTSR	BIOS+UEFI
2511より前	2402 LTSR	BIOS
2511より前	2507 LTSR	UEFI起動の問題があるため、いずれもサポートされていません
2511	2203 LTSR	BIOS+UEFI
2511	2402 LTSR	BIOS+UEFI
2511	2507 LTSR	UEFI (IPv4+IPv6対応)

詳細については、Citrix Provisioningのドキュメントにある「Streaming Linux target devices」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（旧称 Citrix Virtual Apps and Desktops Standard for Azure）では、ドメイン参加済みおよびドメイン非参加の 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。Linux VDA と Azure の両方でサポートされている Linux ディストリビューションでこの機能を使用できます。詳細については、「Citrix DaaS Standard for Azure」を参照してください。

ステップ 1：Azure でマスターイメージを準備する

注:

また、Linux VDA セルフアップデート 機能を使用して、ソフトウェアの自動更新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、マスターイメージ上の etc/xdl/mcs/mcs_local_setting.reg ファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。

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



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



create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t
"REG_SZ" -v "Url" -d "&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. Linux VDA パッケージをインストールする前に、システム要件に従って .NET をインストールします。

.NET 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。

1. (Ubuntu のみ) `source /etc/network/interfaces.d/*` 行を `/etc/network/interfaces` ファイルに追加します。
2. (Ubuntu のみ) `/etc/resolv.conf` を `/run/systemd/resolve/stub-resolv.conf` にポイントする代わりに、`/run/systemd/resolve/resolv.conf` にポイントします。

```
unlink /etc/resolv.conf



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

3. Linux VDA パッケージをインストールします。
4. 使用するデータベースを指定します。

   PostgreSQL に加えて、SQLite または組み込みのレジストリサービスを使用できます。Linux VDA パッケージのインストール後に、SQLite、PostgreSQL、および組み込みのレジストリサービスを切り替えることもできます。これを行うには、次の手順を完了します。

   a. `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールの場合は、この手順を省略してください。

   b. `deploymcs.sh` を実行する前に `/etc/xdl/db.conf` を編集します。

   > **注:**
   >
   > - VDIモードでのみSQLiteを使用することをお勧めします。
   > - 簡易インストールとMCSの場合、SQLite、PostgreSQL、および組み込みレジストリサービスを手動でインストールすることなく切り替えることができます。`/etc/xdl/db.conf` で特に指定しない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションで提供されるバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定され

> たバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡易インストールスクリプト (`ctxinstall.sh`) またはMCSスクリプト (`deploymcs.sh`) を実行する前にPostgreSQLサービスを開始する必要があります。**db.conf**ファイルの例については、手順7：使用するデータベースを指定するを参照してください。
>
> - `/etc/xdl/db.conf`を使用してPostgreSQLのポート番号を構成することもできます。
> - RHEL9、RHEL10、およびRocky9で組み込みレジストリサービスを使用するには、CodeReady Builderリポジトリを有効にする必要があります。 RHEL9: subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms RHEL10: subscription-manager repos --enable codeready-builder-for-rhel-10-$(arch)-rpms Rocky9: dnf config-manager --enable crb

5. MCS変数を変更します。

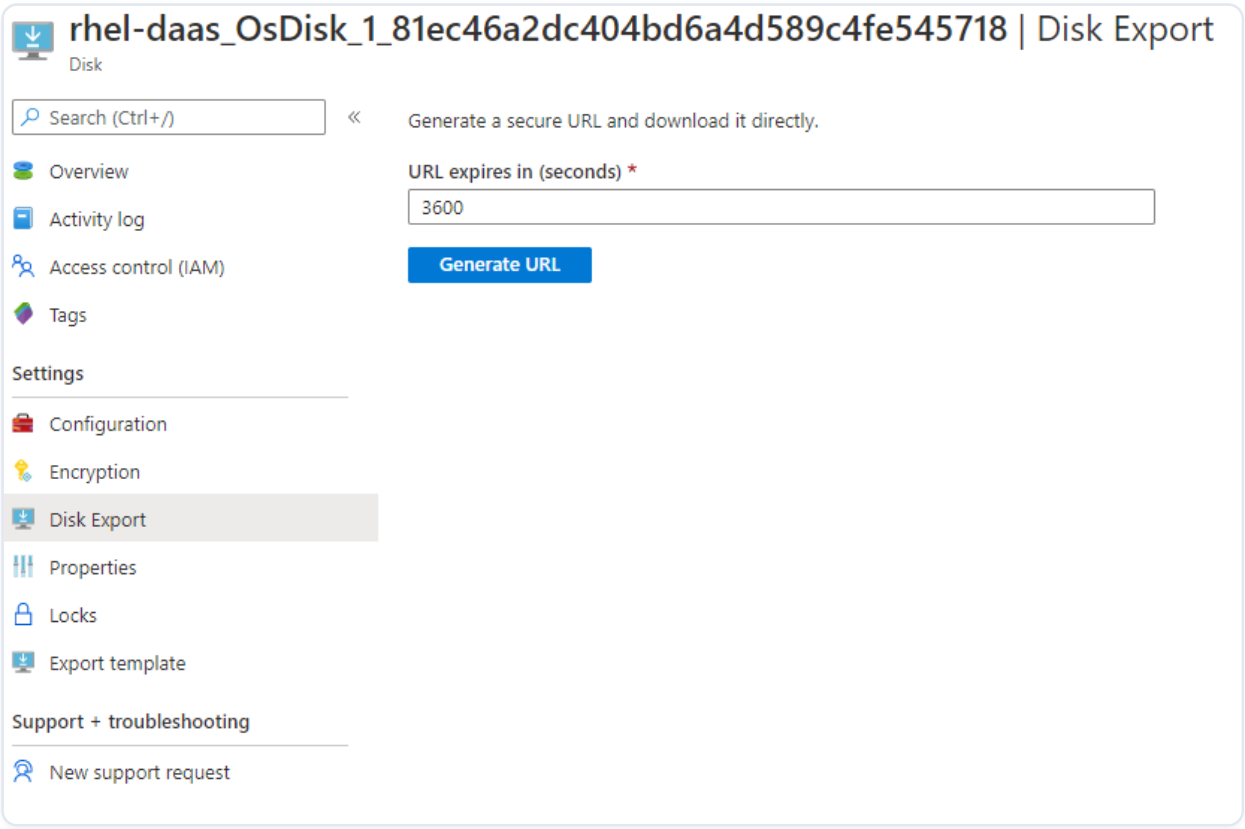
   MCS変数を構成する方法は2つあります。

   - `/etc/xdl/mcs/mcs.conf`ファイルを編集します。
   - 簡易インストールGUIを使用します。簡易インストールGUIを開くには、Linux 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。

   > **注:**
   >
   > `dns`変数は指定しないでください。 マシンカタログを作成するときに**Static**または**Random**タイプを選択する場合は、`VDI_MODE=Y`を設定します。

   > `/etc/xdl/mcs/mcs.conf`を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、**Deploy**をクリックします。GUIで**Deploy**をクリックすると、GUIで設定した変数が`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。

6. Azureで、VMを停止（または割り当て解除）します。**Disk Export**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。



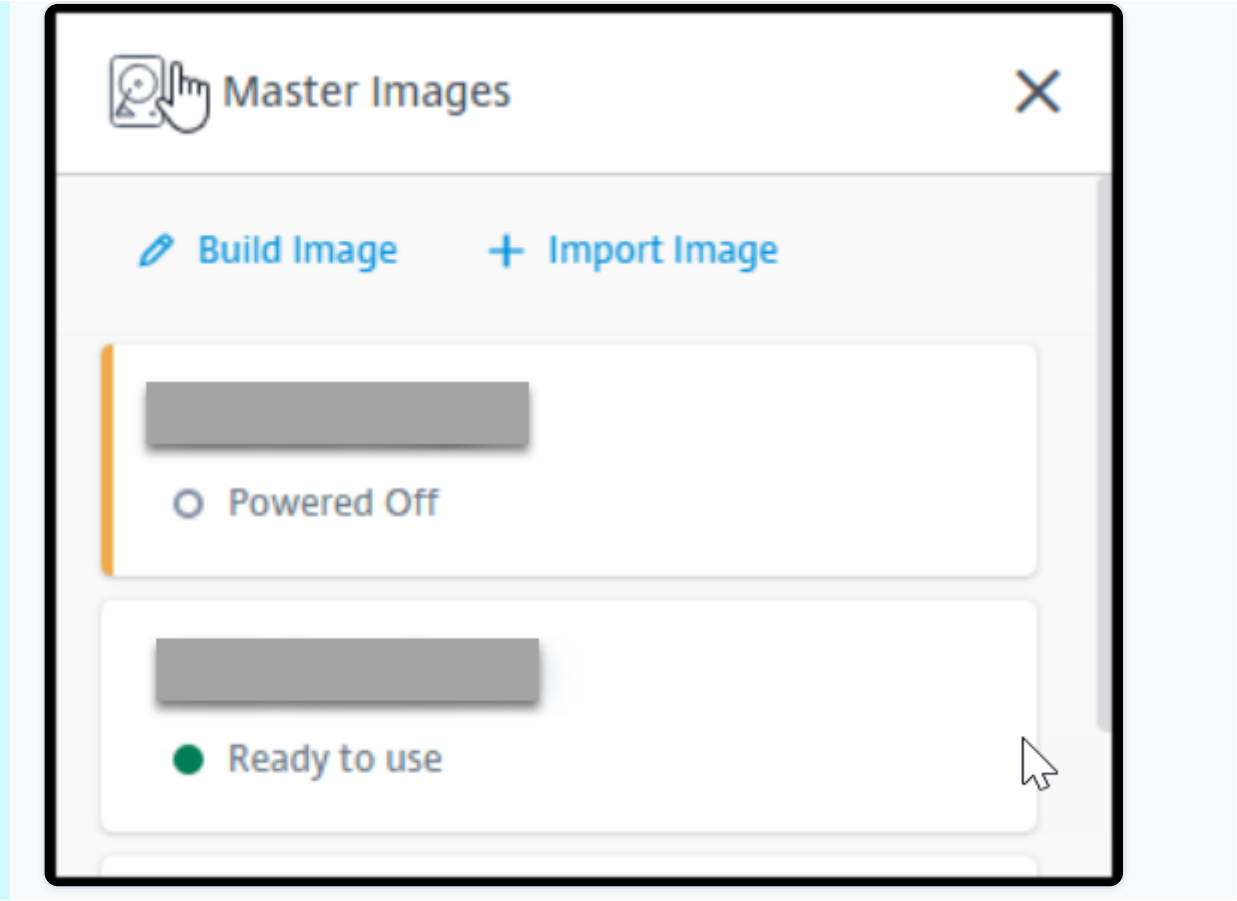
7. （オプション）マスターイメージでグループポリシー設定を行います。`ctxreg`ツールを使用してグループポリシー設定を行うことができます。たとえば、次のコマンドはPDF印刷の**Auto-create PDF Universal Printer**ポリシーを有効にします。

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

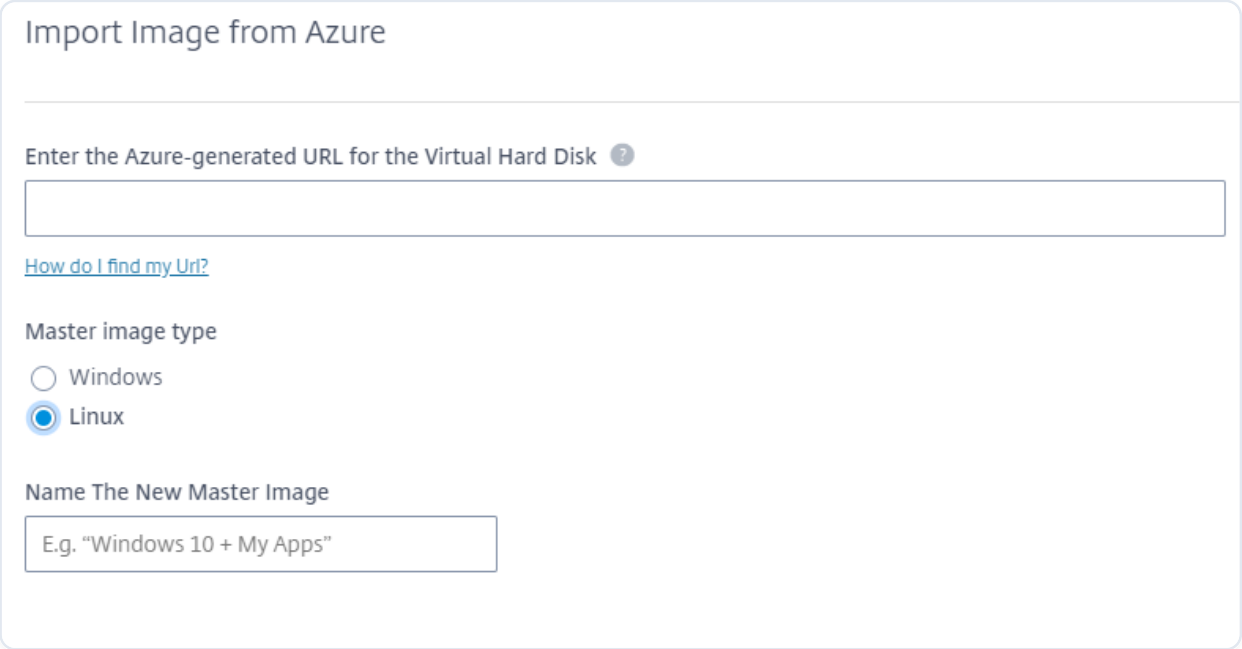
手順2：Azureからマスターイメージをインポートする

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に、「**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。



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ポータルの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。

- RHEL および Rocky Linux
- SUSE
- ウブントゥ
- デビアン

RHELおよびRocky Linux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簡易インストールを使用することをお勧めします。簡易インストールは、この記事で詳述されている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くなります。

ステップ1：構成情報とLinuxマシンの準備

ステップ1a：ネットワーク構成の確認

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：ホスト名の設定

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

`hostname`

ステップ1c：ホスト名にループバックアドレスを割り当てる

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

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

例:

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

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

> **注:**
>
> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。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：ホスト名の確認

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1f: クロック同期を構成する

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

CHRONYサービスを構成する

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

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

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

一般的な展開では、パブリックNTPプールサーバーから直接ではなく、ローカル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 systemctl restart chronyd
```

ステップ1g: 使用するデータベースをインストールして指定する

注:

- VDIモードではSQLiteのみを使用し、ホスト型共有デスクトップ配信モデルではPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
- 簡単なインストールとMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** を通じて別途指定されない限り、Linux VDA はデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように `/etc/xdl/db.conf` を編集し、簡単なインストールスクリプト (`ctxinstall.sh`) またはMCSスクリプト (`deploymcs.sh`) を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方、あるいは組み込みレジストリサービス関連の依存関係を手動でインストールする必要があります。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールした場合、Linux VDAパッケージのインストール後に **/etc/xdl/db.conf** を編集して、どちらか一方を使用するように指定できます。
- RHEL9、RHEL10、Rocky9で組み込みレジストリサービスを使用するには、CodeReady Builderリポジトリを有効にする必要があります。 RHEL9: subscription-manager repos --

> enable codeready-builder-for-rhel-9-$(arch)-rpms RHEL10: subscription-manager repos --enable codeready-builder-for-rhel-10-$(arch)-rpms Rocky9: dnf config-manager --enable crb

POSTGRESQLをインストールする

このセクションでは、Linuxディストリビューションが提供するPostgreSQLのバージョンをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

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
```

RHEL 8.x および RHEL 9.8/9.6 の場合、PostgreSQL 用の `libpq` 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 libpq
```

データベースを初期化するには、次のコマンドを実行します。この操作により、**/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
```

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

次のコマンドを使用してPostgreSQLのバージョンを確認します。

```
psql --version
```

SQLITEをインストールする

SQLite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 sqlite
```

組み込みレジストリサービスをインストールする

組み込みレジストリサービス関連の依存関係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 lmdb
```

使用するデータベースを指定する

SQLite、PostgreSQL、および組み込みレジストリサービスの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、使用するサービスをいずれか1つ指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh**を実行します。新規インストールの場合、この手順は省略してください。
2. 使用するデータベースを指定するために、**/etc/xdl/db.conf**を編集します。以下は**db.conf**ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

#     ctxregd - use the built-in registry service

# default choice is ctxregd
```

```
DbType="ctxregd"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name
```

```
# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

3. カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL**をtrueに設定します。
4. **ctxsetup.sh**を実行します。

注:

PostgreSQLのポート番号を設定するために、**/etc/xdl/db.conf**を使用することもできます。

ステップ2：ハイパーバイザーの準備

サポートされているハイパーバイザー上でLinux VDAを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。

XenServer (旧Citrix Hypervisor™) での時刻同期の修正

XenServer®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XenServer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。

XenServer 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XenServer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統合サービスがインストールされているLinux VMは、Hyper-V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には、NTPサービスと並行してこの機能を有効にする必要があります。

管理オペレーティングシステムから：

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定については、**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XenServer（旧Citrix Hypervisor）とは異なります。Hyper-V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。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、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 マシンを Active Directory (AD) ドメインに追加するには、次の方法があります。

- 「サンバ・ウィンバインド」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/installation-overview/manual-installation-overview/redhat.html#samba-winbind)
- 「クエスト・オーセンティケーション・サービス」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/installation-overview/manual-installation-overview/redhat.html#quest-authentication-services)
- 「セントリファイ・ダイレクトコントロール」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/installation-overview/manual-installation-overview/redhat.html#centrify-directcontrol)
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

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

サンバ・ウィンバインド

RHEL 9.x および Rocky Linux 9.x の場合、**pam_winbind** がルートディレクトリの所有権を変更するのを防ぐために、次のコマンドを実行します。

```
usermod -d /nonexistent nobody
```

必要なパッケージをインストールまたは更新します。

RHEL 9.x/8.x および Rocky Linux 9.x/8.x の場合:

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

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 9.x/8.x および Rocky Linux 9.x/8.x の場合:

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

2. **/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 refresh tickets = true` `winbind offline logon = no`

3. (RHEL 9.x/8.xおよびRocky Linux 9.x/8.xのみ) **/etc/krb5.conf** を開き、`[libdefaults]`、`[realms]`、および`[domain_realm]` セクションの下にエントリを追加します。

   `[libdefaults]` セクションの下：

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

`default_realm = REALM` `dns_lookup_kdc = true`

`[realms]` セクションの下：

`REALM = {`

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

`[domain_realm]` セクションの下：

`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、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。

Linux VMをWindowsドメインに追加するには、次のコマンドを実行します。

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

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

WINBINDのPAMを構成する

デフォルトでは、Winbind PAMモジュール (pam_winbind) の構成では、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 systemctl restart winbind
```

ヒント:

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

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

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

RHEL 9.xおよびRocky Linux 9.xの場合、WinbindでのSELinuxの問題を解決するには、次のコマンドを実行します。

```
ausearch -c 'winbindd' --raw | audit2allow -M my-winbindd -p /etc/selinux/
targeted/policy/policy.*



semodule -X 300 -i my-winbindd.pp
```

ドメインメンバーシップの確認

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

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ドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

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

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なります。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。

次のコマンドを使用して、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します。

```
sudo klist
```

次のコマンドを使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

Winbind PAM モジュールが正しく構成されていることを確認するには、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
```

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

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンして実行できます。ドメイン参加の確認後、(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/installation-overview/manual-installation-overview/redhat.html#step-6-install-the-linux-vda) に進みます。

クエスト認証サービス

ドメインコントローラーでの QUEST の構成

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

ドメインユーザーが LINUX VDA マシンにログオンできるようにする

ドメインユーザーが Linux VDA マシンで HDX™ セッションを確立できるようにするには：

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

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

LINUX VDAでQUESTを構成する

SELinuxポリシーの適用を回避する

デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨害され、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。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、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、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場合、Centrify `adjoin` コマンドを使用して Linux マシンを Active Directory ドメインに参加させます。

```
su -

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

user パラメーターは、コンピューターを Active Directory ドメインに参加させる権限を持つ Active Directory ドメインユーザーです。**domain-name** 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
```

ドメイン参加の確認後、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 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 の構成を確認します
5. ユーザー認証を確認します

ドメインに参加して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、RHEL 8.x/9.x および Rocky Linux 8.x/9.x の `adcli` アプローチについて説明します。**realmd** については、RHEL 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。

- **Adcli (RHEL 9.x/8.x および Rocky Linux 9.x/8.x):**

  必要なパッケージをインストールまたは更新します。

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

Samba および Kerberos 認証用にマシンを構成します。

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

/etc/krb5.conf を開き、 セクションと 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 レルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD をセットアップする

SSSD のセットアップは、次の手順で構成されます。

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

(RHEL 9.x/8.x および Rocky Linux 9.x/8.x 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`

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。

sssd.conf のファイル所有権とアクセス許可を設定します。

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

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

SSSD を有効にする

RHEL 9.x/8.x および Rocky Linux 9.x/8.x 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
```

KERBEROS 構成を確認する

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。シェルによる置換を防ぐため、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

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

```
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ます。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOS サフィックス形式: `username@DOMAIN`

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

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



id -u
```

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

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

ユーザーの Kerberos 資格情報キャッシュ内のチケットが有効で、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

```
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

```
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

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

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の確認

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

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

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

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前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
```

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

手順4: .NETのインストール

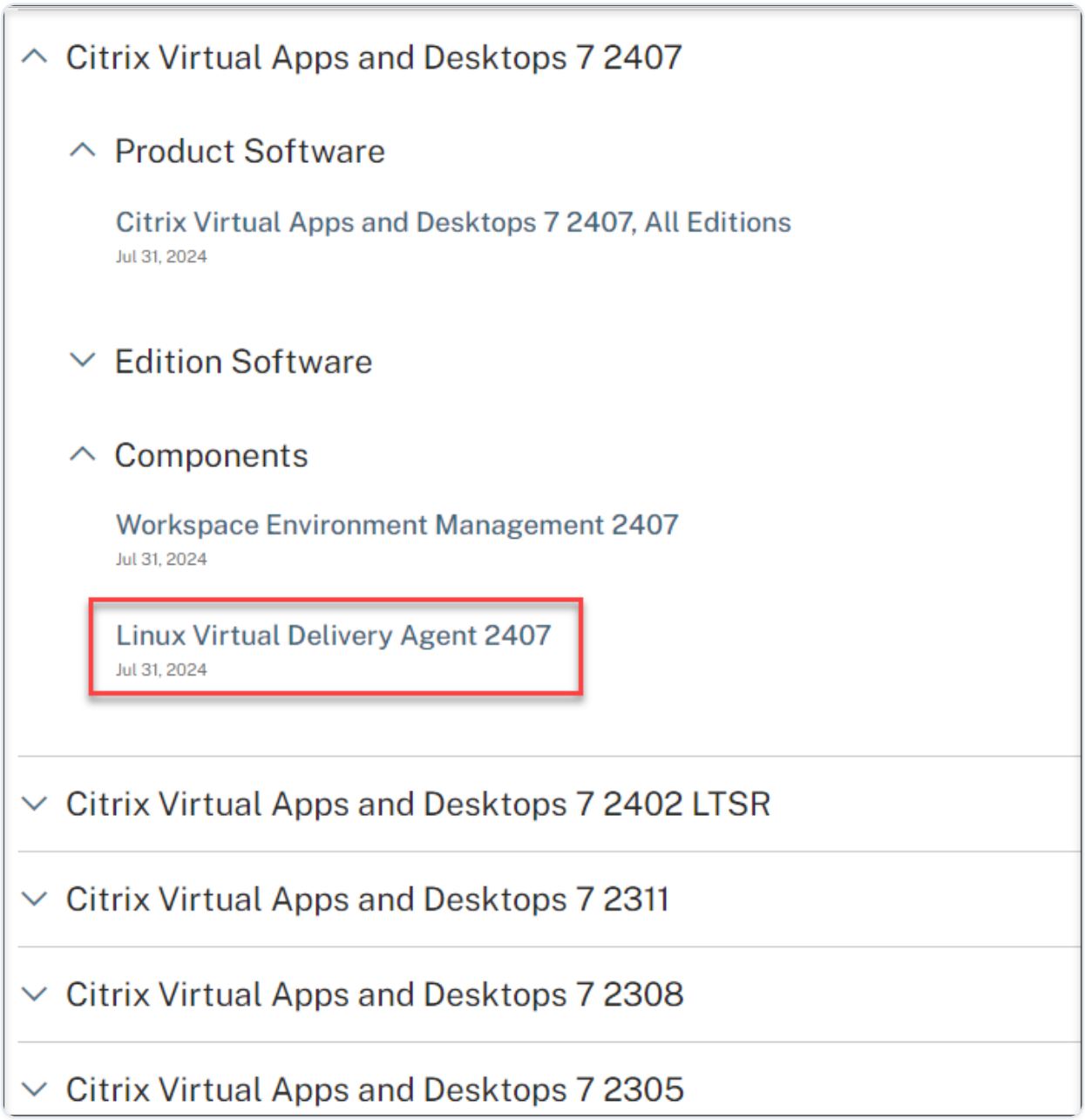
Linux VDAパッケージをインストールする前に、システム要件に従って .NETをインストールします。

.NET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。

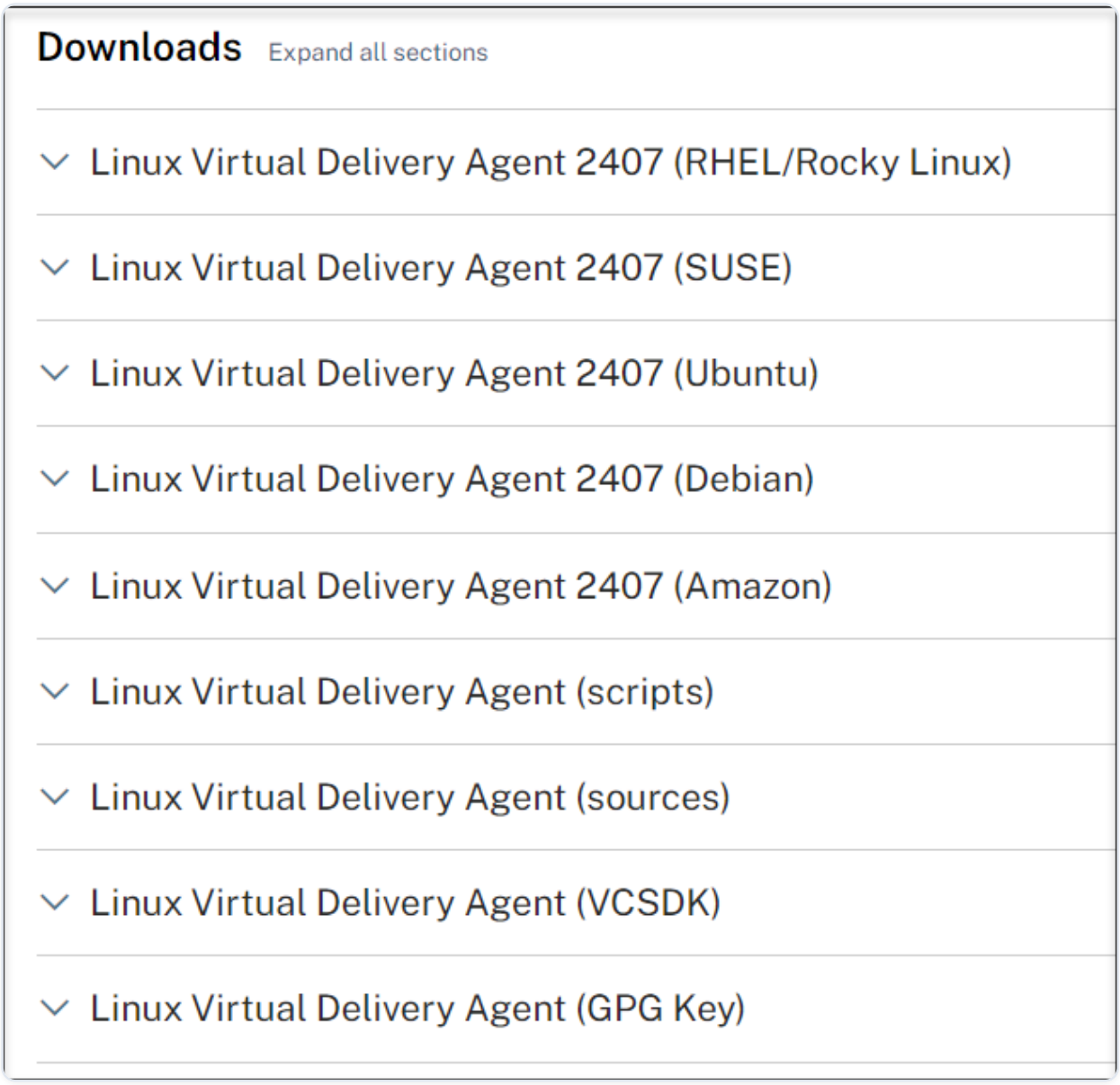
コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。例えば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

ステップ5：Linux VDAパッケージをダウンロードする

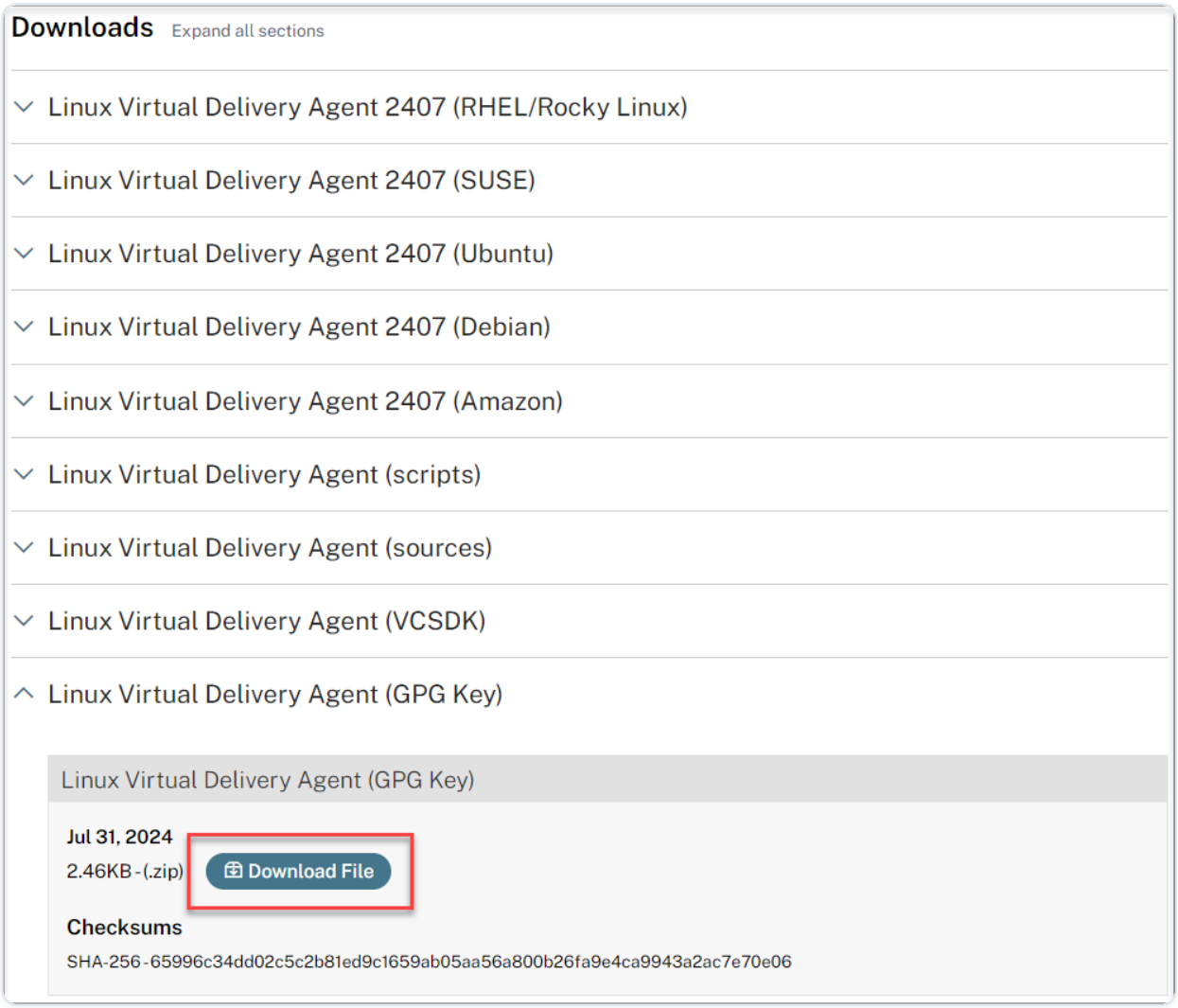
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 適切なバージョンのCitrix Virtual Apps and Desktopsを展開します。
3. **Components**を展開してLinux VDAを見つけます。例：



4. Linux VDAリンクをクリックして、Linux VDAのダウンロードにアクセスします。



5. 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。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：



Linux VDAパッケージの整合性を検証するには、以下のコマンドを実行して公開鍵をRPMデータベースにインポートし、パッケージの整合性を確認します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

ステップ6：Linux VDAをインストールする

新規インストールまたは既存のインストールのアップグレードが可能です。Linux VDAは、最新バージョンからのアップグレードをサポートしています。例えば、Linux VDAを2308から2311へ、1912 LTSRから2203 LTSRへアップグレードできます。

ステップ6a：新規インストールを実行する

1. (オプション) 古いバージョンをアンインストールします。

   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。

a. Linux VDAサービスを停止します。

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

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

b. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

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

2. Linux VDAパッケージをダウンロードします。

   Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。適切なバージョンのCitrix Virtual Apps and Desktops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. `Yum`を使用してLinux VDAソフトウェアをインストールします。

> **注:**

- RHELおよびRocky Linuxの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。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。
- RHEL 9.xおよびRocky Linux 9.xにLinux VDAをインストールする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

RHEL 10.xおよびRocky Linux 10.xの場合：

```
sudo yum install -y XenDesktopVDA-<version>.el10_x.x86_64.rpm
```

RHEL 9.xおよびRocky Linux 9.xの場合：

```
sudo yum install -y XenDesktopVDA-<version>.el9_x.x86_64.rpm
```

RHEL 8.xおよびRocky Linux 8.xの場合：

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

RHEL 10.xおよびRocky Linux 10.xのRPM依存関係リスト：

```
java-21-openjdk >= 21

tzdata-java >= 2025

ImageMagick >= 7.1

gtk2 >= 2.24.33

firewalld >= 0.6.3

policycoreutils-python-utils >= 3.8

python3-policycoreutils >= 3.8

dbus >= 1.14.10

dbus-common >= 1.14.10
```

```
dbus-daemon >= 1.14.10

dbus-tools >= 1.14.10

dbus-x11 >= 1.14.10

libXpm >= 3.5.17

libXrandr >= 1.5.4

libXtst >= 1.2.4

pam >= 1.6.1

util-linux >= 2.40.2

xprop >= 1.2.7

bash >= 5.2.26

findutils >= 4.10.0

gawk >= 5.3.0

sed >= 4.9

pciutils >= 3.13

cups >= 2.4.10

ghostscript >= 10.02.1

libxml2 >= 2.12.5

libmspack >= 0.10.1

ibus >= 1.5.31

nss-tools >= 3.101.0

cyrus-sasl-gssapi >= 2.1.28

python3 >= 3.12.9
```

```
qt5-qtbase >= 5.15.15

qt5-qtbase-gui >= 5.15.15

qrencode-libs >= 4.1.1

fuse >= 2.9.9

fuse-libs >= 2.9.9

pulseaudio-utils >= 17.0

traceroute >= 2.1.6

webkit2gtk4.1 >= 2.50
```

RHEL 9.x および Rocky Linux 9.x の RPM 依存関係リスト:

```
gtk2 >= 2.24.33

java-17-openjdk >= 17

tzdata-java >= 2022

ImageMagick >= 6.9

firewalld >= 0.6.3

policycoreutils-python >= 2.8.9

policycoreutils-python-utils >= 2.8

python3-policycoreutils >= 2.8

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

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6

gawk >= 4.2

sed >= 4.5

cups >= 1.6.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

ibus >= 1.5

nss-tools >= 3.44.0

cyrus-sasl-gssapi >= 2.1

python3 >= 3.6~
```

```
qt5-qtbase >= 5.5~

qt5-qtbase-gui >= 5.5~

qrencode-libs >= 3.4.4

imlib2 >= 1.4.9

fuse-libs >= 2.9

pulseaudio-utils >= 15.0
```

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

```
java-17-openjdk >= 17

ImageMagick >= 6.9

firewalld >= 0.6.3

policycoreutils-python >= 2.8.9

policycoreutils-python-utils >= 2.8

python3-policycoreutils >= 2.8

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

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6

gawk >= 4.2

depends_on sed >= 4.5

pulseaudio >= 14.0

pulseaudio-module-x11 >= 14.0

pulseaudio-module-bluetooth >= 14.0

alsa-plugins-pulseaudio >= 1.1.9

cups >= 1.6.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

ibus >= 1.5

nss-tools >= 3.44.0

gperftools-libs >= 2.4

cyrus-sasl-gssapi >= 2.1
```

```
python3 >= 3.6~

qt5-qtbase >= 5.5~

qt5-qtbase-gui >= 5.5~

qrencode-libs >= 3.4.4

imlib2 >= 1.4.9

fuse-libs >= 2.9
```

注:

このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、システム要件を参照してください。

手順 6b: 既存のインストールをアップグレードする（オプション）

Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311へ、1912 LTSRから2203 LTSRへアップグレードできます。

RHELおよび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:

- 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。
- RHEL 9.x および Rocky Linux 9.x でLinux VDAをアップグレードする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。
- 2407リリース以降、Linux VDAは、アップグレード中の構成ファイルの処理をパッケージマネージャーの**rpm**または**dpkg**に委任します。以下に、rpmとdpkgが構成ファイルの変更とどのように相互作用するかを説明します。

- **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを`.rpmnew`拡張子で保存します。
- **dpkg**: どのように続行するかを対話形式で尋ねます。ローカル構成ファイルを保持し、新しいパッケージバージョンを**.dpkg-new**または**.dpkg-dist**として保存しながら、Linux VDAをサイレントアップグレードするには、次のコマンドを使用します。

```
dpkg --force-confold -i package.deb  # Always keep your version,
then save new package's version as *.dpkg-new or *.dpkg-dist
```

- ソフトウェアのアップグレード後、Linux VDAマシンを再起動してください。

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。

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。

- XenServer
- ヴイエムウェア イーエスエックス
- 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に、各質問に対してデフォルト（「いいえ」）を選択します。

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

> NVIDIA smiコードスニペット(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/linux-vda-nvidia-smi-code-snippet.png)

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、**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

プロンプトが表示される質問に従って手動で構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプト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、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は「n」です。ドメイン参加シナリオでは、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – 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。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerに登録するために、Delivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_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、値を**「y」**に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード用に構成されます（つまり、CTX_XDL_VDI_MODE=**「y」**）。

- **CTX_XDL_START_SERVICE='y\|n'** – 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。
- **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 VDA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_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または**Xfce**デスクトップ環境を指定します。**'<none\>'**に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳細については、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- **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 のインストールパス。デフォルトのパスは **'/usr/bin'** です。
- **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_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 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directory フォレスト内で LDAP クエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連する LDAP ポート番号を 3268 として指定します。この変数は、デフォルトで **'<none\>'** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP を照会します。検索パフォーマンスを向上させるには、検索ベース (例: OU=VDI,DC=mycompany,DC=com) を指定できます。不要な場合は、**'<none\>'** に設定します。

- **CTX_XDL_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。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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_SUPPORT_DDC_AS_CNAME='y|n'

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

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。前述のコマンドから、最初の行に **#\!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

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

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

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

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

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

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

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

構成の変更を削除する

一部のシナリオでは、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。

手順9：XDPingを実行する

Linux VDA環境で一般的な構成の問題がないか確認するには、`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。

手順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 systemctl restart ctxhdx



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

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの稼働状況を確認するには：

```
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。
  - ホスト型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」という概念をサポートしていませんでした。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1人のユーザーという配信モデルが意味されます。

> **ヒント:**
>
> 削除されたマシンをActive Directoryドメインに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、再度追加してください。

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 2503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、この記事で詳述されている手動インストールよりも時間と労力を節約でき、エラーも発生しにくくなります。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: YaST ツールを起動する

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: ネットワークを構成する

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を構成する方法について説明します。ネットワークの構成は、Network Manager などの他の方法ではなく、YaST ツールを介して実行されます。これらの手順は、UI ベースの YaST ツールを使用することを前提としています。テキストベースの YaST ツールも使用できますが、ここでは説明されていない異なるナビゲーション方法があります。

ホスト名とドメインネームシステム (DNS) を構成する

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. **Hostname/DNS** タブを開きます。
4. **Set Hostname via DHCP** の **no** オプションを選択します。
5. **Modify DNS Configuration** の **Use Custom Policy** オプションを選択します。
6. ネットワーク設定を反映するように、以下を編集します。
   - **静的ホスト名** – マシンのDNSホスト名を追加します。
   - **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。
7. `/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 <FQDN of the VDA> <hostname of the VDA> localhost`

注:

Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。そのため、ホスト名は15文字を超えてはなりません。 **ヒント:**

> a～z、A～Z、0～9、およびハイフン (-) 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。

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。

Step 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設定にはいくつかの変更が必要になる場合があります。

SUSE 15.7の場合:

1. UIベースのYaSTツールを起動します。
2. **ネットワークサービス**を選択し、次に**NTP設定**を選択します。
3. **NTPデーモンの開始**セクションで、**今すぐ起動してブート時にも起動**を選択します。
4. **構成ソース**には**動的**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
   ```

Step 1d: Linux VDAの依存パッケージをインストールする

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

- Open Motif ランタイム環境 2.3.1 以降
- Cups 1.6.0以降
- ImageMagick 6.8以降

リポジトリの追加

ImageMagickを除くほとんどの必要なパッケージは、公式リポジトリから入手できます。ImageMagickパッケージを入手するには、YaSTまたは次のコマンドを使用して`sle-module-desktop-applications`リポジトリを有効にします。

```
SUSEConnect -p sle-module-desktop-applications/<version number>/x86_64
```

KERBEROSクライアントのインストール

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。次の説明を参照してください。

使用するデータベースのインストールと指定

注:

- VDIモードではSQLiteのみを使用し、ホスト型共有デスクトップ配信モデルではPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
- 簡易インストールおよびMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションで提供されるバージョンではなく、PostgreSQLのカスタムバージョンが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように`/etc/xdl/db.conf`を編集し、簡易インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方、あるいは組み込みレジストリサービス関連の依存関係を手動でインストールする必要があります。Linuxディストリビューションで提供されるバージョンではなく、PostgreSQLのカスタムバージョンを使用できます。SQLiteとPostgreSQLの両方をインストールする場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

PostgreSQLのインストール
このセクションでは、Linuxディストリビューションで提供されるPostgreSQLのバージョンをインストールする方法について説明します。PostgreSQLのカスタムバージョンが必要な場合は、特定の要件に基づいてインストールできます。

`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 zypper install postgresql-server



sudo zypper install postgresql-jdbc
```

マシンの起動時またはすぐにPostgreSQLを開始するには、次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLiteのインストール
SUSEの場合、SQLiteをインストールするには次のコマンドを実行します。

```
sudo zypper install sqlite3
```

組み込みレジストリサービスをインストールする
組み込みレジストリサービス関連の依存関係をインストールするには、次のコマンドを実行します。

```
sudo zypper install lmdb
```

使用するデータベースを指定する
SQLite、PostgreSQL、および組み込みレジストリサービスをインストールした場合、Linux VDA パッケージのインストール後に **/etc/xdl/db.conf** を編集して、そのうちの1つを使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省略してください。
2. **/etc/xdl/db.conf** を編集して、使用するデータベースを指定します。以下は **db.conf** ファイルの例です。

   ```
   # database configuration file for Linux VDA



   ```

```
## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

#     ctxregd - use the built-in registry service



# default choice is ctxregd



DbType="ctxregd"









## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432









## PostgreSQL customized

# only the following value means true, otherwise false:

#     true
```

```
#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

3. カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL** をtrueに設定します。
4. **ctxsetup.sh** を実行します。

> **注:**
>
> **/etc/xdl/db.conf** を使用して、PostgreSQLのポート番号を構成することもできます。

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、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。

XenServer（旧Citrix Hypervisor™）での時刻同期を修正する

XenServer®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XenServerの両方で問題が発生します。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。HVMモードでは変更は不要です。

XenServer 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XenServerの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

この変更を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。

```
reboot
```

再起動後、設定が正しいことを確認します。

```
su -
```

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

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux Integration Services がインストールされている Linux VM は、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、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、NTP との競合を避けるためにホスト時刻同期が無効になっている VMware や XenServer (旧 Citrix Hypervisor) とは異なります。Hyper-V の時刻同期は、NTP 時刻同期と共存し、それを補完することができます。

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、準仮想化された各 Linux VM 内で NTP とハイパーバイザー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. **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：Linux VM を Windows ドメインに追加する

Linux マシンを Active Directory (AD) ドメインに追加するには、次の方法があります。

- サンバ・ウィンバインド
- 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

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、**ネットワークサービス**、次に**Windowsドメインメンバーシップ**を選択します。
2. 次の変更を行います。
   - **ドメインまたはワークグループ**をActive Directoryドメインの名前またはドメインコントローラーのIPアドレスに設定します。ドメイン名が大文字であることを確認してください。
   - **Linux認証にSMB情報を使用**をオンにします。
     - **ログイン時にホームディレクトリを作成**をオンにします。
     - **SSHのシングルサインオン**をオンにします。
     - **オフライン認証**がオフになっていることを確認します。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。一部のパッケージのインストールを求められた場合は、**インストール**をクリックします。

4. ドメインコントローラーが見つかった場合、ドメインに参加するかどうかを尋ねられます。**はい**をクリックします。
5. プロンプトが表示されたら、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の資格情報を入力し、**OK**をクリックします。
6. サービスを手動で再起動するか、マシンを再起動します。マシンを再起動することをお勧めします。

```
su -

reboot
```

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。

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

システムのキー配布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
```

マシンアカウントの詳細を確認するには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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します。これを行うには、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クレデンシャル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/2607-ltsr/installation-overview/manual-installation-overview/suse.html#step-6-install-the-linux-vda)に進みます。

Quest認証サービス

ドメインコントローラーでのQUESTの構成

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

VASデーモンの構成
Kerberosチケットの自動更新を有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

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



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

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間チケットの有効期間より1時間短いです。チケットの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を手動で構成します。

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



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

WINDOWSドメインに参加する

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

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

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

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

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

`ERROR: No domain could be found.`

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

ユーザー認証を確認する

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します。これを行うには、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
```

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/2607-ltsr/installation-overview/manual-installation-overview/suse.html#step-6-install-the-linux-vda)に進みます。

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

Centrify DirectControlエージェント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。

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

userは、マシンを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では、すべてのVDAマシン（Windows VDAおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。Centrify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 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動かない場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/installation-overview/manual-installation-overview/suse.html#step-6-install-the-linux-vda)に進みます。

SSSD

SUSEでSSSDを使用している場合は、このセクションの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のPAMを構成する
3. SSSDをセットアップする
4. SSSDを有効にする
5. ドメインメンバーシップを確認する
6. Kerberos構成を確認する
7. ユーザー認証を確認する

ドメインに参加し、ホストキータブを作成する

SSSDは、ドメイン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**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する。

```
sudo systemctl stop nscd

sudo systemctl disable nscd
```

2. ホスト名とChrony時刻同期を確認する。

```
hostname

hostname -f

chronyc traking
```

3. 必要なパッケージをインストールまたは更新する：

```
sudo zypper install samba-client sssd-ad
```

4. 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はKerberosレルム名（例：example.com）です。**REALM**はKerberosレルム名の大文字表記（例：EXAMPLE.COM）です。

5. root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。****セクションの下に以下のエントリを追加します。

```
[global]

    workgroup = domain


    client signing = yes


    client use spnego = yes


    kerberos method = secrets and keytab


    realm = REALM


    security = ADS
```

domainは、EXAMPLEなどのActive Directoryドメインの短いNetBIOS名です。

6. ユーザーとグループを解決する際にSSSDを参照するように、**/etc/nsswitch.conf**ファイル内の**passwd**および**group**エントリを変更します。

```
passwd: compat sss



group: compat sss
```

7. 構成済みのKerberosクライアントを使用して、管理者としてターゲットドメインに認証します。

```
kinit administrator
```

8. **net**ユーティリティを使用して、システムをドメインに参加させ、システムキータブファイルを生成します。

```
net ads join osname="SUSE Linux Enterprise Server" osVersion=15 -U
administrator
```

SSSDのPAMを構成する

SSSDのPAMを構成する前に、必要なパッケージをインストールまたは更新します。

```
sudo zypper install sssd sssd-ad
```

SSSDを介したユーザー認証のためにPAMモジュールを構成し、ユーザーログオン用のホームディレクトリを作成します。

```
sudo pam-config --add  --sss

sudo pam-config --add --mkhomedir
```

SSSDを設定する

1. root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`構成の例（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などの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にマッピングする処理を行います。そうしないと、Active DirectoryはPOSIX拡張機能を提供できる必要があります。**ad_gpo_access_control**は**permissive**に設定されており、Linuxセッションでの無効なログオンエラーを防ぎます。`sssd.conf`および`sssd-ad`のマニュアルページを参照してください。

2. `sssd.conf`のファイルの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、以下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の確認

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構成の確認

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、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:

```
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、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。

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

user は、Active Directoryドメインにマシンを追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

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

ドメインメンバーシップを確認する

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

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

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。そうでない場合は、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します。これを行うには、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のインストールに進みます。

手順4：.NETのインストール

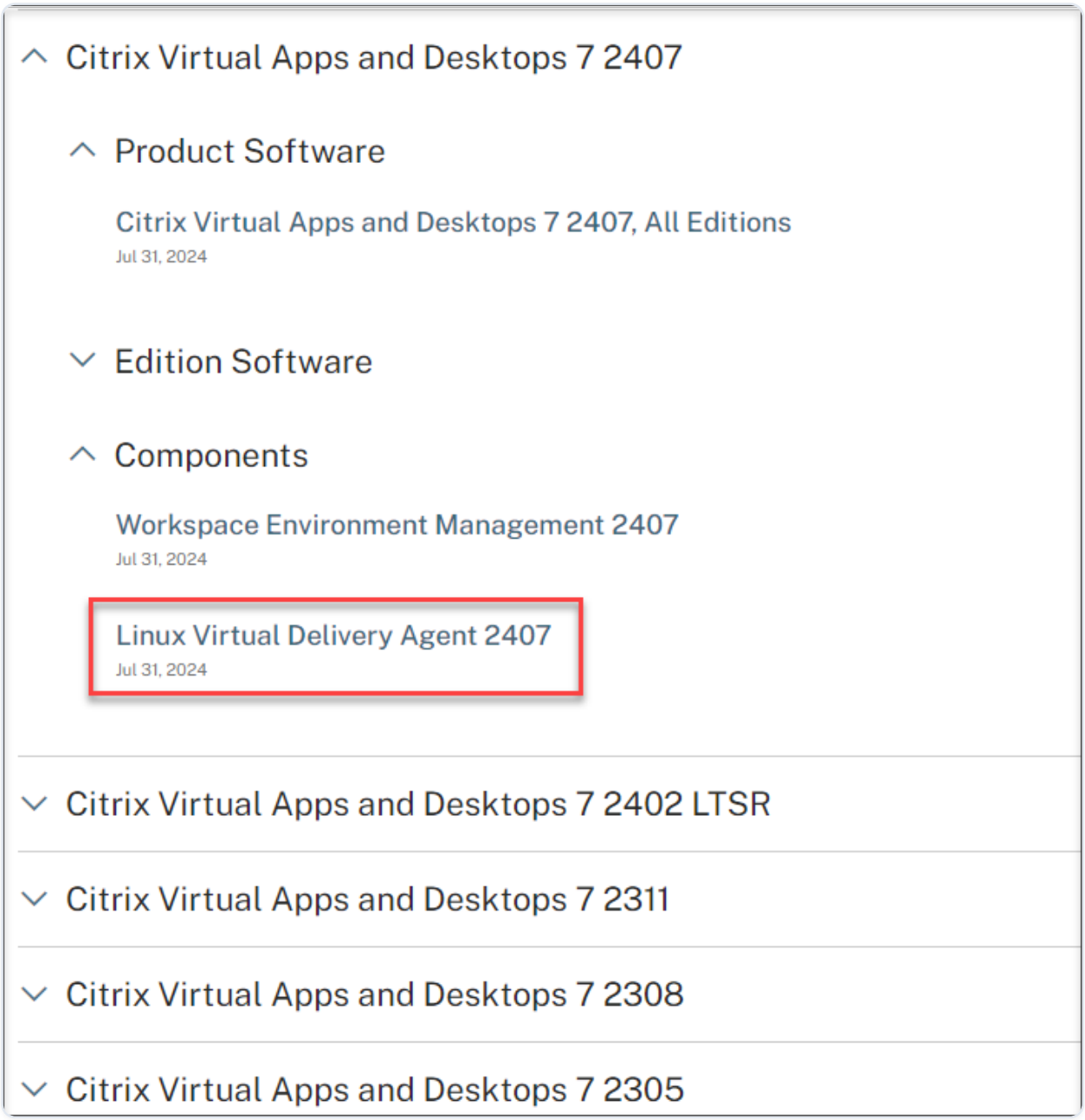
Linux VDAパッケージをインストールする前に、システム要件に従って.NETをインストールします。

.NET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。

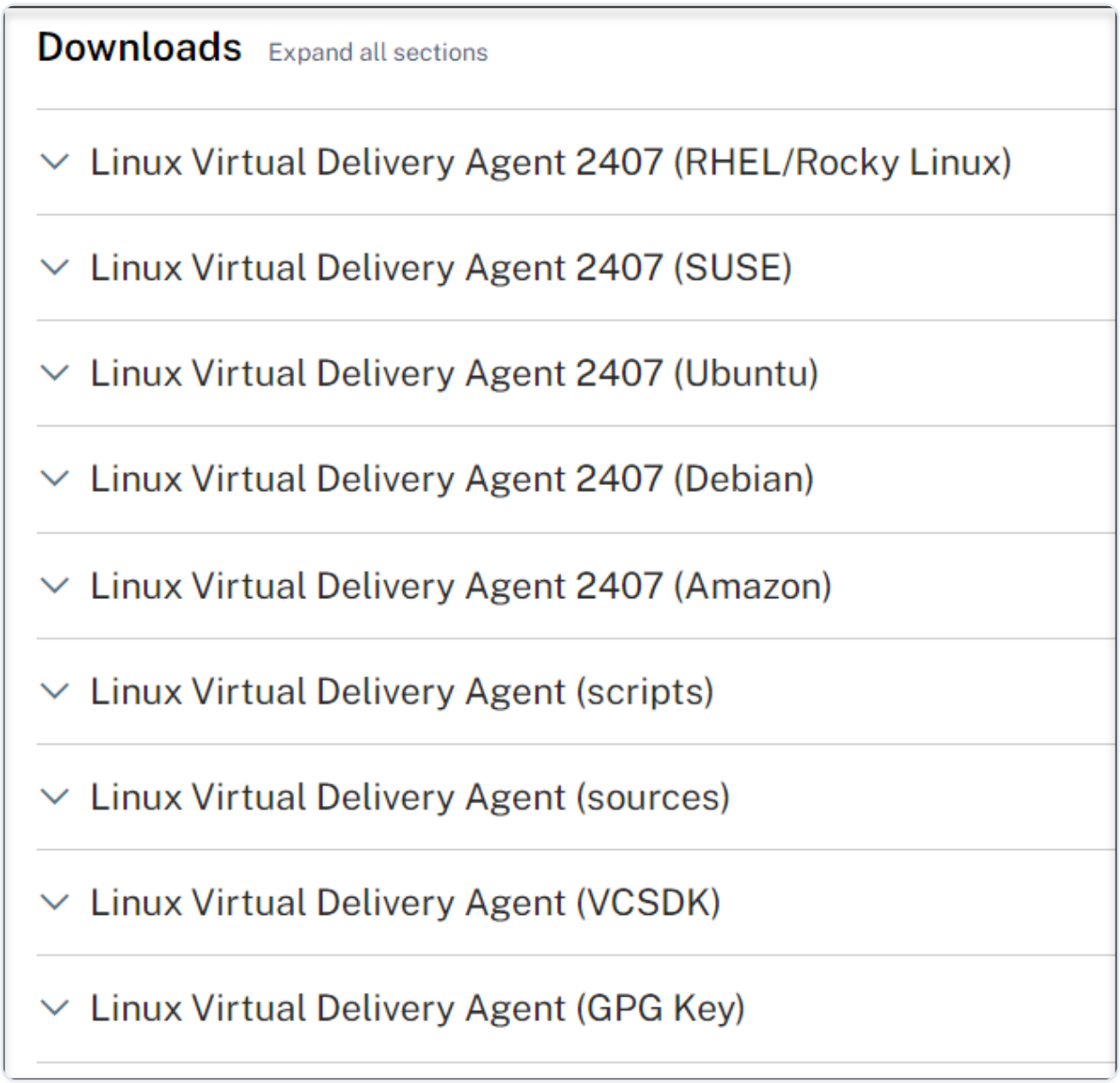
コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

手順5：Linux VDAパッケージのダウンロード

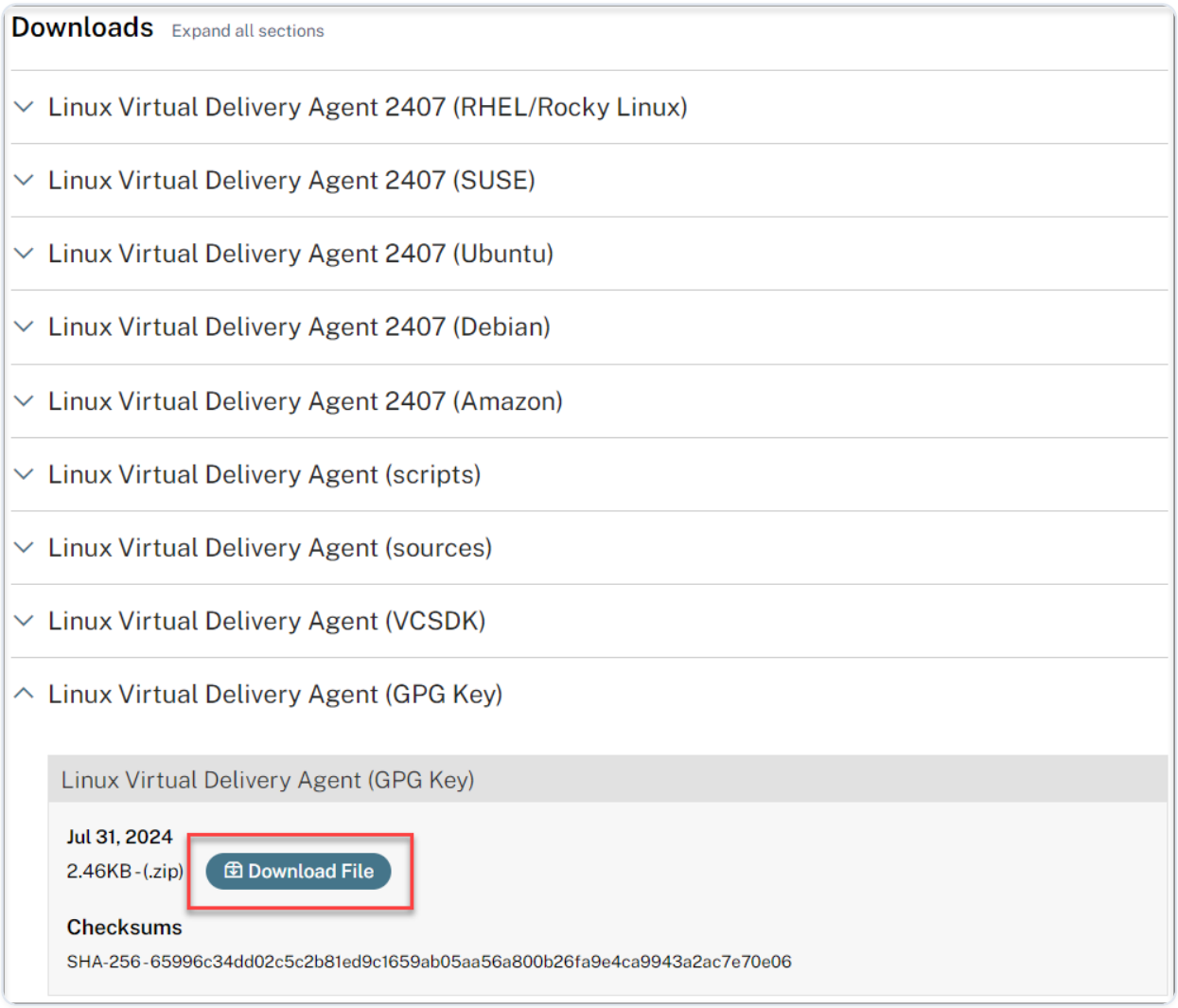
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. **コンポーネント**を展開してLinux VDAを見つけます。例：



4. Linux VDAリンクをクリックして、Linux VDAのダウンロードにアクセスします。



5. お使いのLinuxディストリビューションに合ったLinux VDAパッケージをダウンロードします。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：



公開キーを使用してLinux VDAパッケージの整合性を検証するには、以下のコマンドを実行して公開キーをRPMデータベースにインポートし、パッケージの整合性を確認します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

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 systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

注:

ctxvdaサービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**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:

インストールされているコンポーネントは **/opt/Citrix/VDA/** の下にあります。

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。

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

ステップ6c：Linux VDAをアップグレードする（オプション）

Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311に、1912 LTSRから2203 LTSRにアップグレードできます。

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

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

SUSE 15 の RPM 依存関係リスト:

```
java-17-openjdk >= 17

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

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

libQt5DBus5 >= 5.12

libtcmalloc4 >= 2.5

libcap-progs >= 2.26

mozilla-nss-tools >= 3.53.1

libpython3_6m1_0 >= 3.6~

libQt5Widgets5 >= 5.12

libqrencode4 >= 4.0.0

libImlib2-1 >= 1.4.10

libgtk-2_0-0 >= 2.24

libgthread-2_0-0 >= 2.48

pulseaudio-utils >= 5.0

lsb-release >= 2.0

pkexec >= 121

cyrus-sasl-gssapi >= 2.1

libfuse2 >= 2.9
```

重要:

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。

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。

- XenServer
- 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 (クラウドベンダーまたは NVIDIA から提供) をインストールします。

手順 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 の場合は、**/etc/locale.gen** ファイルを編集して **# en_US.UTF-8 UTF-8** の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。スクリプトは変更を加える前に、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

プロンプトで質問に答える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を環境変数で指定します。必要な変数がすべて存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンをドメインに結合するかどうか。 デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – 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。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_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_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_START_SERVICE='y\|n'** – 構成完了時にLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシンの起動後にLinux Virtual Desktopサービスが開始されます。

- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA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_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate/'<none\>'** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。**'<none\>'** に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、デスクトップ切り替えコマンドおよびシステムトレイを参照してください。
- **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 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_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_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 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 ポートを持つ LDAP FQDN のスペース区切りリストを指定できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268 のように指定します。Active Directory フォレスト内で LDAP クエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連する LDAP ポート番号を 3268 として指定します。この変数はデフォルトで **'<none\>'** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP にクエリを実行します。検索パフォーマンスを向上させるには、検索ベース (例: OU=VDI,DC=mycompany,DC=com) を指定できます。不要な場合は、**'<none\>'** に設定します。
- **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。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

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

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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_SUPPORT_DDC_AS_CNAME='y|n'

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

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。最初の行に **#\!/bin/bash** を指定して、上記のコマンドから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、単一のコマンドで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_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \
```

```
CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

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

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

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

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

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

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環境で一般的な構成の問題がないか確認するには、`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 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サービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDAを再起動する:

Linux VDAサービスを再起動するには:

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する:

Linux VDAサービスの実行ステータスを確認するには:

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ 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、**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:**
>
> マシンをActive Directoryドメインから削除して再参加させる場合は、そのマシンをマシンカタログから削除して再度追加する必要があります。

ステップ 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:**

> Linux VDA バージョン 1.4 以降では、アプリケーションの公開がサポートされています。ただし、Linux VDA は、同じマシンへのデスクトップとアプリの配信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 2503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簡易インストールを使用することをお勧めします。簡易インストール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 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います。

```
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。その方法は、**/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ることです。

```
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。

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: マルチキャストDNSを無効にする

デフォルト設定ではマルチキャストDNS (**mDNS**) が有効になっており、名前解決の結果が不安定になる可能性があります。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

次のように変更します。

`hosts: files dns`

ステップ1f: 名前解決とサービス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
```

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

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

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

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

ループバックIPアドレス、ローカルホスト、パブリックサーバーの***.pool.ntp.org**エントリなど、リストされている他の**サーバー**または**プール**エントリをすべて削除します。

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

手順1h：使用するデータベースをインストールして指定する

注:

- VDIモードではSQLiteのみを使用し、ホストされた共有デスクトップ配信モデルではPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
- 簡易インストールおよびMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションで提供されているバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡易インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方、あるいは組み込みレジストリサービス関連の依存関係を手動でインストールする必要があります。Linuxディストリビューションで提供されているバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールする場合は、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

POSTGRESQLをインストールする

このセクションでは、Linuxディストリビューションで提供されているPostgreSQLのバージョンをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

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 apt-get install -y postgresql



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

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLITEをインストールする

Ubuntuの場合、SQLite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 -y sqlite3
```

組み込みレジストリサービスをインストールする

組み込みレジストリサービス関連の依存関係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 -y liblmdb0
```

使用するデータベースを指定する

SQLite、PostgreSQL、および組み込みレジストリサービスの両方をインストールした場合、Linux VDAパッケージのインストール後に **/etc/xdl/db.conf** を編集して、そのうちの1つを使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省略してください。
2. **/etc/xdl/db.conf** を編集して、使用するデータベースを指定します。以下は **db.conf** ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

#     ctxregd - use the built-in registry service



# default choice is ctxregd



DbType="ctxregd"







## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432







## PostgreSQL customized
```

```
# only the following value means true, otherwise false:
#     true
#     yes
#     y
#     YES
#     Y
# default is false
DbCustomizePostgreSQL=false



## PostgreSQL service name
# specify the service name of PostgreSQL for Linux VDA
# default is "postgresql"
DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL** を true に設定します。

3. **ctxsetup.sh** を実行します。

注:

/etc/xdl/db.conf を使用して、PostgreSQLのポート番号を構成することもできます。

手順1i：Motifをインストールする

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

手順1j：その他のパッケージをインストールする

Ubuntu 26.04の場合：

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap2

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0t64
```

Ubuntu 24.04の場合：

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap2

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

Ubuntu 22.04の場合：

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.5-0

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

手順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。

XenServer（旧Citrix Hypervisor™）での時刻同期を修正する

XenServer®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XenServerの両方で問題が発生します。どちら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。

XenServer 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XenServer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、それ以上の操作は必要ありません。

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

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

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

この変更を再起動後も永続的に適用するには、**/etc/sysctl.conf ファイル**を編集し、次の行を追加します。

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration Servicesがインストールされ ているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックが正確に保たれるように、NTPサービスと並行してこの機能を有効にしてください。

管理オペレーティングシステムから：

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**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XenServer（旧Citrix Hypervisor）とは異なります。Hyper-Vの時刻同期は、NTP時刻同期と共存し、それを補完することができます。

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとハイパーバイザー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、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マシンをActive Directory (AD) ドメインに追加するには、以下の方法が利用可能です。

- サンバ・ウィンバインド
- Quest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する

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

マシンの起動時にWINBINDデーモンが開始するように有効にする

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

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind`スクリプトが`/etc/init.d`の下にあることを確認してください。

KERBEROSを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストにおける**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。

vim /etc/samba/smb.confコマンドを実行して**/etc/samba/smb.conf**を開き、以下の設定を行います。

`[global]`

`workgroup =` *`WORKGROUP`*

`security = ADS`

`realm = REALM`

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`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、以下の行に**winbind**を追加します。

`passwd: compat winbind`

`group: compat winbind`

WINDOWSドメインに参加する

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

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

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

`WINBIND`を再起動する

```
sudo systemctl restart winbind
```

WINBIND 用に PAM を構成する

次のコマンドを実行し、**Winbind NT/Active Directory 認証**および**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
```

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 のすべての VDA マシンが Active Directory にコンピューターオブジェクトを持っていることを要求します。

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
```

次の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。

```
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。Kerberos の **kinit** コマンドを実行して、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
```

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

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

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

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

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

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっており、適切に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、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`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-enabled**」をチェックします。
4. **Primary GID Number**を、実際のドメインユーザーグループのグループIDに設定します。

注:

これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDAでの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。この適用は、Questが使用するUnixドメインソケットIPCメカニズムと干渉し、ドメインユーザーのログオンを妨げます。

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、完全なロックアウトを引き起こす可能性があります。

VASデーモンの構成
Kerberosチケットの自動更新は有効にし、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の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、WindowsまたはLinuxにかかわらず、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で参加させたLinuxマシンがドメイン上にあることを確認するには:

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用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
```

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エージェント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では、WindowsまたはLinuxにかかわらず、すべてのVDAマシンが**Active Directory**にコンピューターオブジェクトを持っている必要があります。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
su -



adinfo
```

「**Joined to domain**」の値が有効であり、「**CentrifyDC mode**」が「**connected**」を返すことを確認します。モードが開始状態で停止したままの場合、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サービスへの接続性をテストします。

```
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の構成

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、rootとして **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> 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、Active DirectoryをIDプロバイダーとして、Kerberosを認証に使用するように構成する必要があります。ただし、SSSDはドメイン参加やシステム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**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 22.04マシンでは、`adcli 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かどうかをテストします。

- **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 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** 構成ファイルはデフォルトでインストールされていないため、手動で作成する必要があります。root として、**/etc/sssd/sssd.conf** を作成または開き、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` が ad_gpo_map_remote_interactive に追加されます。

このコンテキストでの **domain-dns-name** パラメーターは、example.com のような DNS ドメイン名です。**REALM** は、EXAMPLE.COM のような大文字の Kerberos レルム名です。NetBIOS ドメイン名を構成する必要はありません。

構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

SSSD デーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

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

SSSD デーモンの開始

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** 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。

- 「**adcli** を使用してドメインメンバーシップを確認する場合は、`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。」.
- 「**Samba** を使用してドメインメンバーシップを確認する場合は、`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。Kerberos `kinit` コマンドを実行して、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します。

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

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

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

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u



klist

```

```
exit
```

klist コマンドによって返されるKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

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

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
9.1.0/pbis-open-9.1.0.551.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

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

PBISインストールスクリプトを実行する

```
sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

WINDOWSドメインに参加する

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

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

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

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

ドメインメンバーシップを確認する

Delivery Controller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、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。

```
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をインストールする に進みます。

手順4：.NETをインストールする

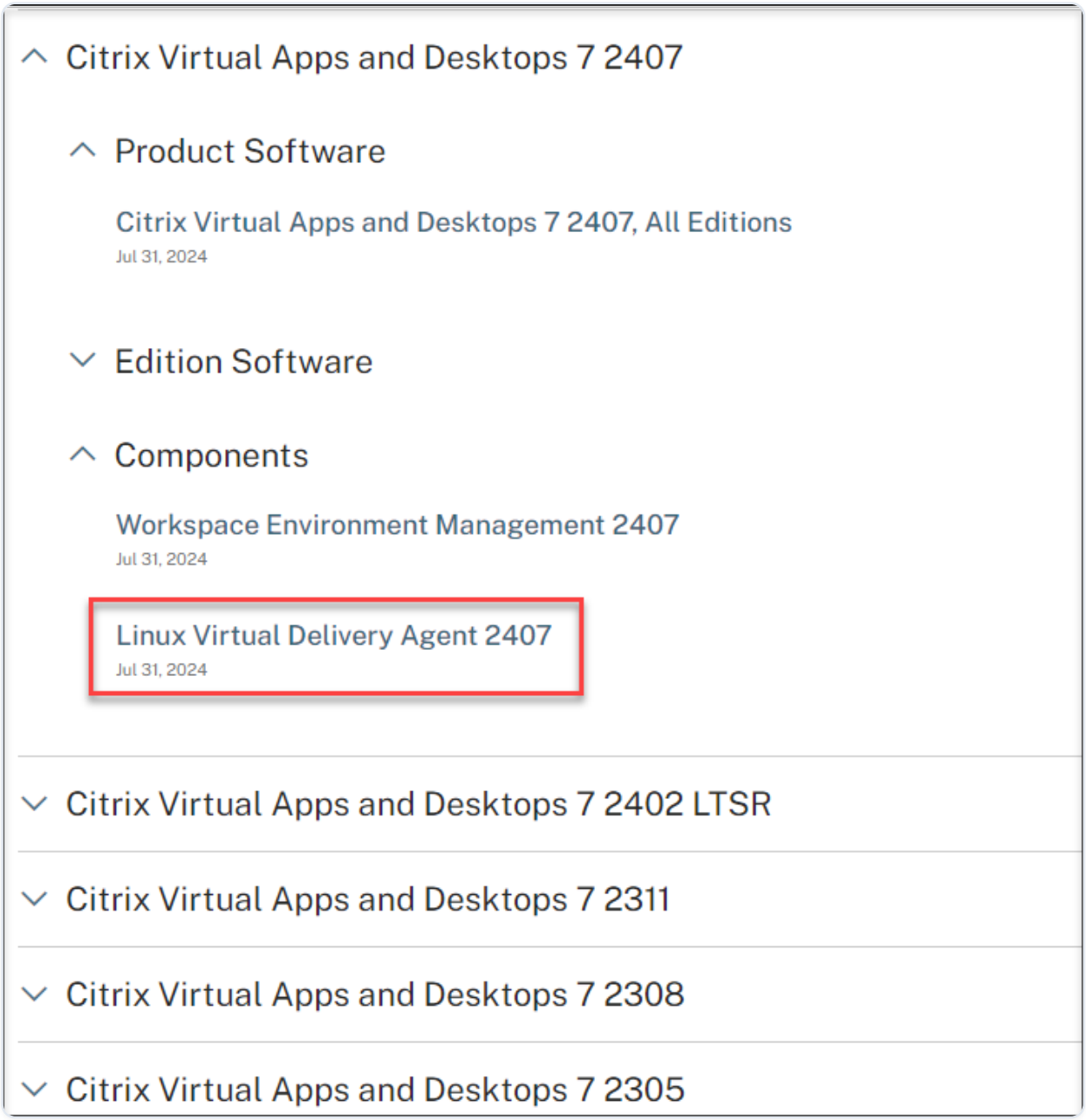
Linux VDAパッケージをインストールする前に、システム要件(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html#net-requirements)に従って.NETをインストールします。

.NET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。

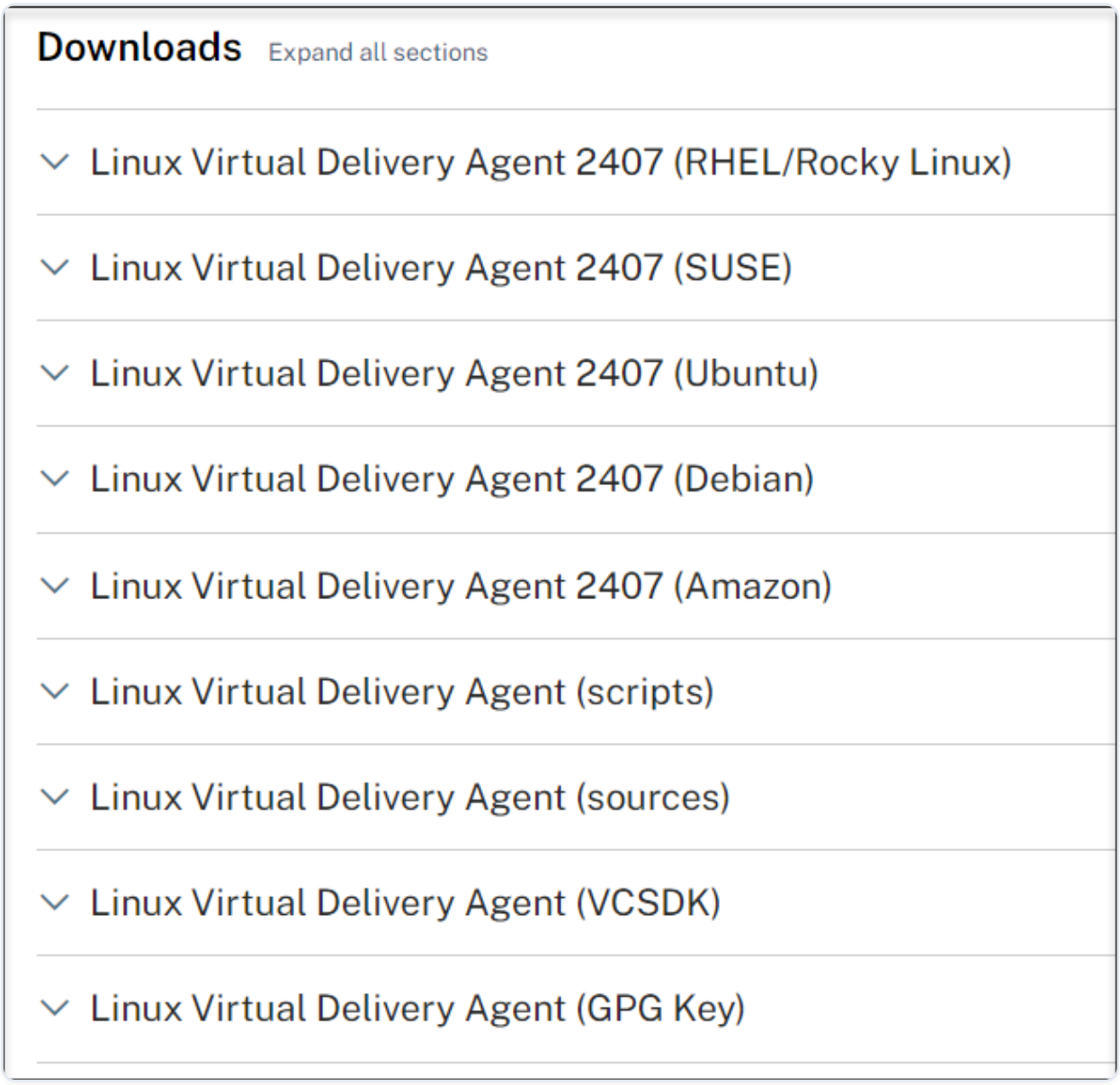
コマンドの出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンドの出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

ステップ5：Linux VDAパッケージをダウンロードする

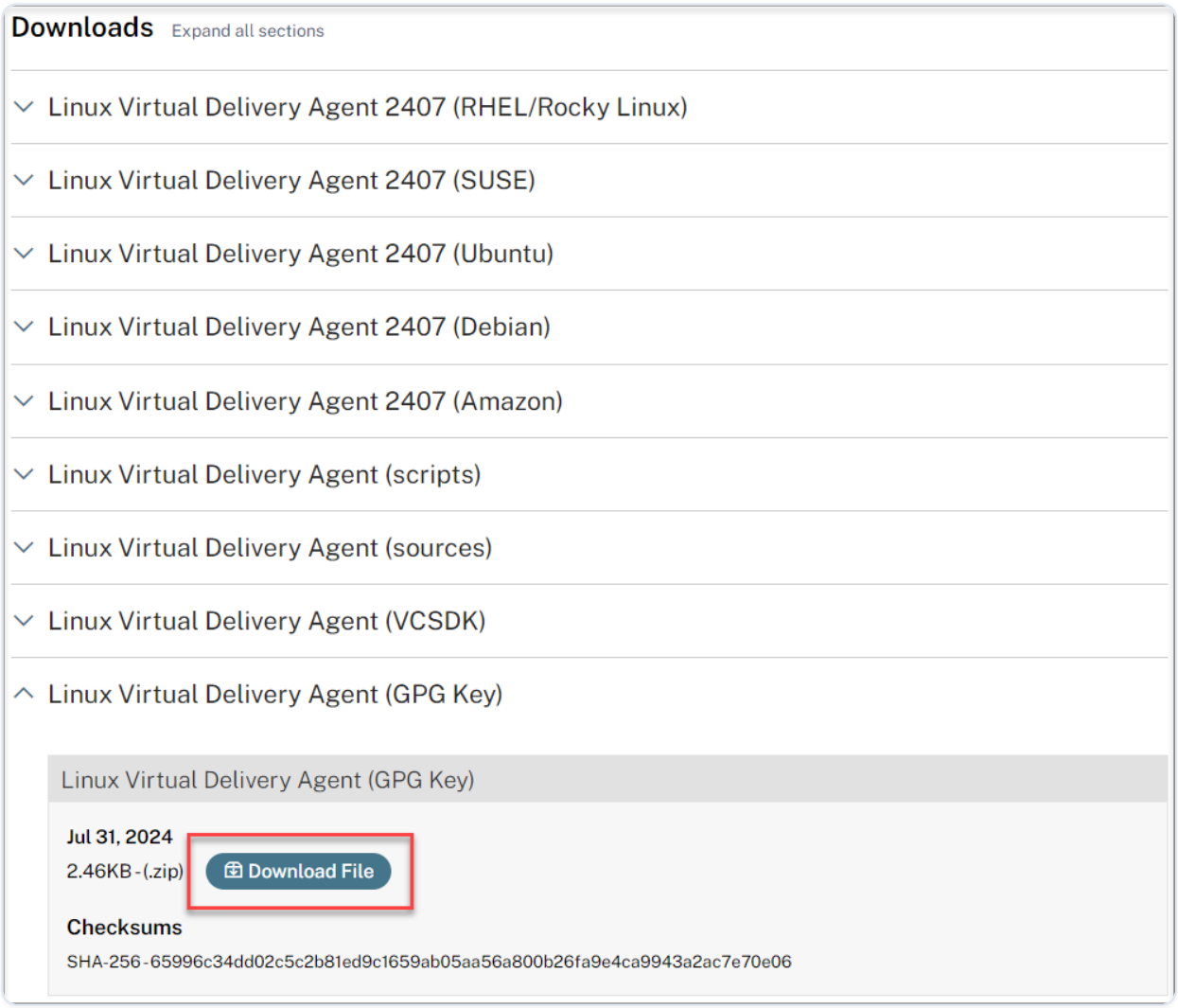
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 適切なバージョンのCitrix Virtual Apps and Desktopsを展開します。
3. **Components**を展開してLinux VDAを見つけます。例：



4. Linux VDAリンクをクリックして、Linux VDAダウンロードにアクセスします。



5. 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。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：



Linux VDAパッケージの整合性を検証するには、次のコマンドを実行して公開鍵をDEBデータベースにインポートし、パッケージの整合性を確認します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

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:

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

apt-get install -f
```

注:

GCP 上の Ubuntu の場合、RDNS を無効にします。そのためには、/etc/krb5.conf の の下に **rdns = false** の行を追加します。

Ubuntu 26.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:7.1.2

libgtk2.0-0t64 >= 2.24.33

libgtkmm-3.0-1t64 >= 3.24.10

libxrandr2 >= 2:1.5.4

libxtst6 >= 2:1.2.5

libxm4 >= 2.3.8

libmspack0t64 >= 0.11

libpython3.14 >= 3.14~

libgoogle-perftools4t64 >= 2.16

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

libnss3-tools >= 2:3.114

libqt5dbus5t64 >= 5.15~

libqt5widgets5t64 >= 5.15~

libfuse2t64 >= 2.9

libqrencode4 >= 4.1.1

libwebkit2gtk-4.1-0 >= 2.50
```

```
mutter >= 50.0

dbus-x11 >= 1.16

pulseaudio-utils >= 1:17.0

ibus >= 1.5

ufw >= 0.36

ubuntu-desktop >= 1.560

cups >= 2.4

curl >= 8.14

bash >= 5.2

findutils >= 4.10.0

sed >= 4.9

pciutils >= 3.13

util-linux >= 2.41

gtk3-nocsd >= 3

traceroute >= 1:2.1

libbsd-dev >= 0.12.2-2build1
```

Ubuntu 24.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.12

libgtkmm-3.0-1t64 >= 3.24.9

ufw >= 0.36
```

```
ubuntu-desktop >= 1.539

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.39

gtk3-nocsd >= 3

bash >= 5.2

findutils >= 4.9.0

sed >= 4.9

cups >= 2.4

libmspack0t64 >= 0.11

curl >= 8.5.0

libbsd-dev >= 0.12.1-1build1

ibus >= 1.5

libqt5dbus5 >= 5.15~

libgoogle-perftools4 >= 2.15

libpython3.12 >= 3.12~

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

libnss3-tools >= 2:3.98

libqt5widgets5 >= 5.15~

libqrencode4 >= 4.1.1

libimlib2 >= 1.12.1
```

```
libfuse2 >= 2.9

mutter >= 46.0

pulseaudio-utils >= 16.1
```

Ubuntu 22.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.11

libgtkmm-3.0-1v5 >= 3.24.5

ufw >= 0.36

ubuntu-desktop >= 1.481

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.37

gtk3-nocsd >= 3

bash >= 5.1

findutils >= 4.8.0

sed >= 4.8

cups >= 2.4

libmspack0 >= 0.10

ibus >= 1.5

libqt5dbus5 >= 5.15~
```

```
libgoogle-perftools4    2.9~

libpython3.10 >= 3.10~

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

libnss3-tools >= 2:3.68

libqt5widgets5 >= 5.15~

libqrencode4 >= 4.1.1

libimlib2 >= 1.7.4

libfuse2 >= 2.9

mutter >= 42.5
```

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)

Linux VDA は、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDA を 2308 から 2311 へ、および 1912 LTSR から 2203 LTSR へアップグレードできます。

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



sudo apt-get install -f
```

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。
- 2407 リリース以降、Linux VDA は、アップグレード中の構成ファイルの処理をパッケージマネージャーの **rpm** または **dpkg** に委任します。以下に、rpm と dpkg が構成ファイルの変更とどのように相互作用するかを説明します:
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを .rpmnew 拡張子で保存します。
  - **dpkg**: 対話形式で続行方法の選択を求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを **.dpkg-new** または **.dpkg-dist** として保存しながら Linux VDA をサイレントアップグレードするには、次のコマンドを使用します:

    ```
    dpkg --force-confold -i package.deb  # Always keep your version,
    then save new package's version as *.dpkg-new or *.dpkg-dist
    ```

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。

- XenServer
- 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（クラウドベンダーまたはNVIDIAから提供）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**ファイルを編集して**# en_US.UTF-8 UTF-8**の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で必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求せず、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – 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。

- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerへの登録に使用する、スペース区切りの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_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_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_START_SERVICE='y\|n'** – 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。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシンの起動後にLinux Virtual Desktopサービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA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。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または**Xfce**デスクトップ環境を指定します。**'<none\>'**に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、デスクトップ切り替えコマンドおよびシステムトレイを参照してください。
- **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_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_SITE_NAME=<dns-name\>** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。不要な場合は、**'<none\>'**に設定します。

- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDAはDNS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ポートを含むスペース区切りのLDAP FQDNのリストを指定できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数はデフォルトで**'<none\>'**に設定されます。
- **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。不要な場合は、**'<none\>'**に設定します。
- **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。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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_SUPPORT_DDC_AS_CNAME='y|n'

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

sudoコマンドを実行するときは、既存の環境変数を新しく作成されるシェルに渡すために、**-E**オプションを入力します。上記のコマンドから、最初の行に**#\!/bin/bash**を記述した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_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

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

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

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

```
CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

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

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

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する:

Linux VDAサービスの稼働状況を確認するには:

```
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ついては、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

> **注:**

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

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は許可しないでください。
- デリバリーグループとWindowsマシンを含むマシンカタログを混在させないでください。

マシンカタログとデリバリーグループの作成方法については、Citrix Virtual Apps and Desktops 7 2503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：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**ファイルを変更して、マシンのホスト名のみが含まれるようにします。

```
hostname
```

ステップ1b：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。ホスト名は15文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン（-）文字のみを使用してください。アンダーバー（_）、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。

ステップ1c：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 1d: マルチキャスト DNS を無効にする

デフォルト設定ではマルチキャスト DNS (**mDNS**) が有効になっており、名前解決の結果が一貫しなくなる可能性があります。

mDNS を無効にするには、**/etc/nsswitch.conf** を編集して、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

変更後:

`hosts: files dns`

ステップ 1e: 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 を実行できることを確認します。

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

ステップ 1f: クロック同期を構成する (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
```

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

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

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

一般的な展開では、パブリック NTP プールサーバーから直接ではなく、ローカル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 **server** または **pool** 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 systemctl restart chrony
```

ステップ 1g: 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 libgtk2.0-0
```

ステップ1h: 必要な依存関係をインストールするためのリポジトリを追加する

Debian 11の場合、/etc/apt/sources.list ファイルに `deb http://deb.debian.org/debian/ bullseye main` 行を追加します。

ステップ1i: 使用するデータベースをインストールして指定する

注:

- VDIモードでのみSQLiteを使用し、ホスト型共有デスクトップ配信モデルにはPostgreSQLまたは組み込みレジストリサービスを使用することをお勧めします。
- 簡易インストールおよびMCSの場合、SQLite、PostgreSQL、または組み込みレジストリサービスを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** で特に指定がない限り、Linux VDAはデフォルトで組み込みレジストリサービスを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように `/etc/xdl/db.conf` を編集し、簡易インストールスクリプト (`ctxinstall.sh`) またはMCSスクリプト (`deploymcs.sh`) を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方、あるいは組み込みレジストリサービス関連の依存関係を手動でインストールする必要があります。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールする場合は、Linux VDA

> パッケージのインストール後に **/etc/xdl/db.conf** を編集して、どちらか一方を使用するように指定できます。

POSTGRESQLをインストールする

このセクションでは、Linuxディストリビューションが提供するバージョンのPostgreSQLをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

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 apt-get update



sudo apt-get install -y postgresql



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

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

SQLITEをインストールする

Debianの場合、SQLite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 -y sqlite3
```

組み込みレジストリサービスをインストールする

組み込みレジストリサービス関連の依存関係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 -y liblmdb0
```

使用するデータベースを指定する

SQLite、PostgreSQL、および組み込みレジストリサービスの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、使用するサービスをいずれか1つ指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh**を実行します。新規インストールである場合は、この手順を省略してください。
2. 使用するデータベースを指定するために **/etc/xdl/db.conf** を編集します。以下は、**db.conf** ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

#     ctxregd - use the built-in registry service



# default choice is ctxregd



DbType="ctxregd"







## database port

# specify database port for the database.

# if not specified, default port will be used:
```

```
# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL**をtrueに設定します。

3. **ctxsetup.sh**を実行します。

> **注:**
>
> **/etc/xdl/db.conf** を使用して、PostgreSQLのポート番号を構成することもできます。

手順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。

XenServer（旧Citrix Hypervisor™）での時刻同期の修正

XenServer® Time Sync機能が有効になっている場合、各パラ仮想化Linux VM内でNTPとXenServer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。

XenServer VM Toolsがインストールされたパラ仮想化Linuxカーネルを実行している場合、Linux VM内からXenServer Time Sync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XenServer（旧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。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、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：Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory (AD) ドメインに追加するには、次の方法が利用可能です。

- サンバ・ウィンバインド
- Quest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、マシンの起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind`スクリプトが`/etc/init.d`の下にあることを確認してください。

KERBEROSを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

WINBIND認証を構成する

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`

`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: files systemd winbind`

`group: files systemd winbind`

WINDOWS ドメインに参加する

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

```
sudo net ads join <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** および **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ヒント:

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

ドメインメンバーシップを確認する

Delivery Controller は、Windows または Linux のすべての VDA マシンが `Active Directory` にコンピューターオブジェクトを持つことを要求します。

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

```
sudo net ads testjoin
```

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

```
sudo net ads info
```

KERBEROS構成の確認

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、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
```

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、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 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-enabled** をオンにします。
4. **Primary GID Number** を実際のドメインユーザーグループのグループ ID に設定します。

注:

これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

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

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

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

WINDOWSドメインに参加する

Quest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`

ユーザー認証を確認する

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

```
ssh localhost -l domain\\username



id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

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** パラメーターは、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinux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によって参加したLinuxマシンがドメイン上にあることを確認するには：

```
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をインストールするに進みます。

SSSD

KERBEROS の構成

Kerberos をインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberos を構成するには、root として **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> 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。

ドメインに参加する

SSSD は、ID プロバイダーとして Active Directory を使用し、認証に Kerberos を使用するように構成する必要があります。ただし、SSSD はドメインへの参加やシステム keytab ファイルの管理のための AD クライアント機能を提供しません。代わりに、**adcli**、**realmd**、または **Samba** を使用できます。

> **注:**
>
> このセクションでは、**adcli** と **Samba** 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** を使用してドメインに参加します。

   古いシステムキータブファイルを削除し、以下を使用してドメインに参加します。

```
su -



rm -rf /etc/krb5.keytab



adcli join domain-dns-name -U user -H hostname-fqdn
```

user は、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. システムキータブを確認します。

   システムキータブファイルが作成されていることを確認するために、`sudo klist -ket` コマンドを実行します。

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **Samba を使用してドメインに参加する場合は、次の手順を実行します。**

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

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
```

SSSD を構成する

SSSD デーモンを開始する前に、SSSD 構成の変更が必要です。SSSD の一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトでインストールされていないため、手動で作成する必要があります。root として、**/etc/sssd/sssd.conf** を作成または開き、次の設定を行います。

`[sssd]`

`services = nss, pam`

`config_file_version = 2`

`domains =` *`domain-dns-name`*

`[domain/`*`domain-dns-name`*`]`

`id_provider = ad`

```
access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14
days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

注:

ldap_id_mapping は **true** に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングする処理を行います。そうでない場合、`Active Directory` は POSIX 拡張機能を提供できる必要があります。PAM サービス `ctxhdx` が ad_gpo_map_remote_interactive に追加されます。

このコンテキストでの **domain-dns-name** パラメーターは、example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。NetBIOS ドメイン名を構成する必要はありません。

> 構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

> SSSD デーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

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

SSSD デーモンを起動する

SSSD デーモンを今すぐ起動し、マシンの起動時にデーモンが起動するようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM 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。

- **adcli** を使用してドメインメンバーシップを確認する場合は、`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。
- **Samba** を使用してドメインメンバーシップを確認する場合は、`sudo net ads testjoin` コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info` コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

KERBEROS 構成の確認

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

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンで認証を直接テストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返されるKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

ルートユーザーとして、以前の **id -u** コマンドによって返されたuidに対応するチケットキャッシュファイルが作成され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

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
9.1.0/pbis-open-9.1.0.551.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

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

PBISインストールスクリプトを実行する

```
sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

WINDOWSドメインに参加する

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

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

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

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

ドメインメンバーシップを確認する

Delivery Controller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
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

手順4：.NETのインストール

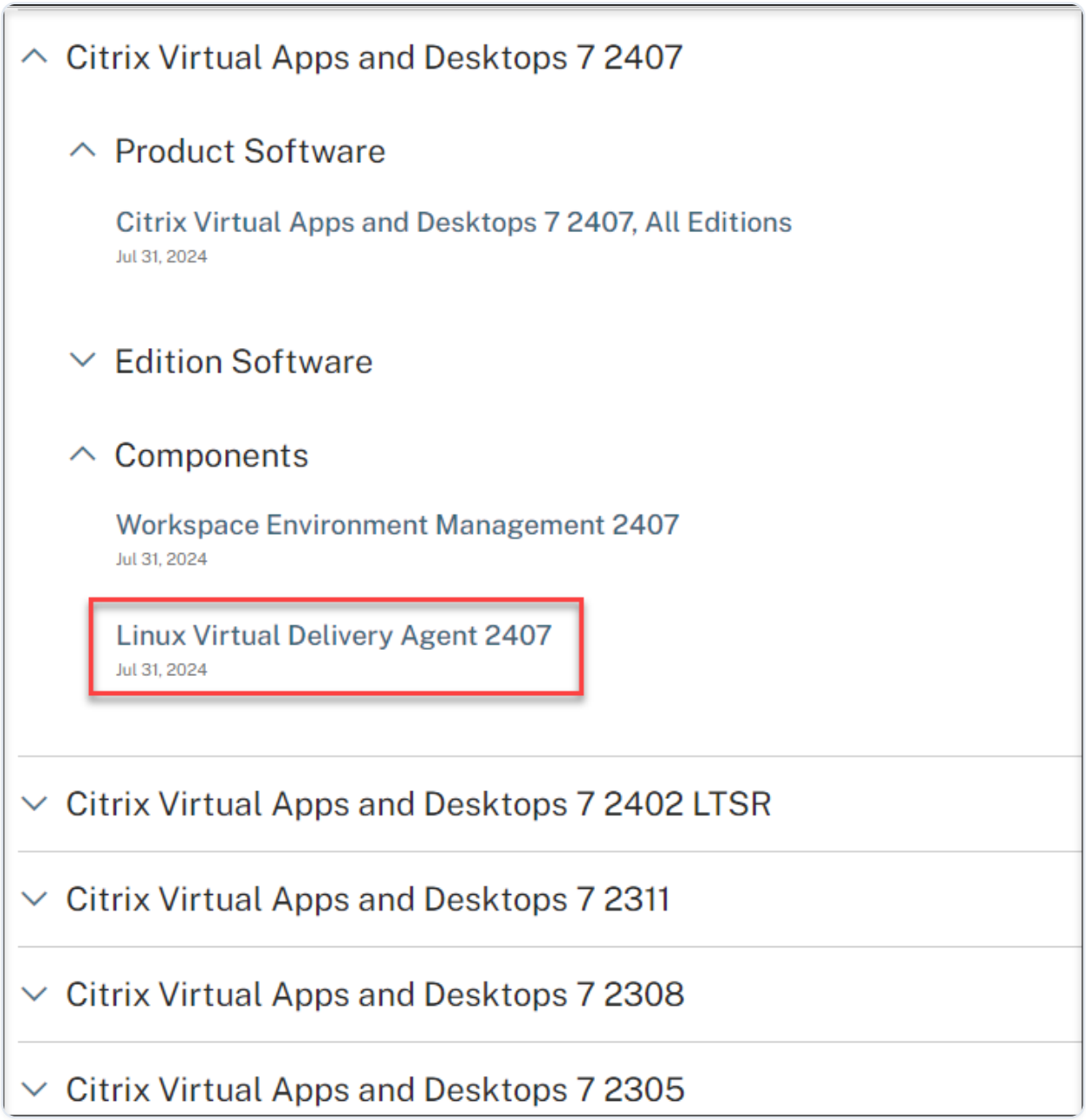
Linux VDAパッケージをインストールする前に、システム要件に従って .NETをインストールします。

.NET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。

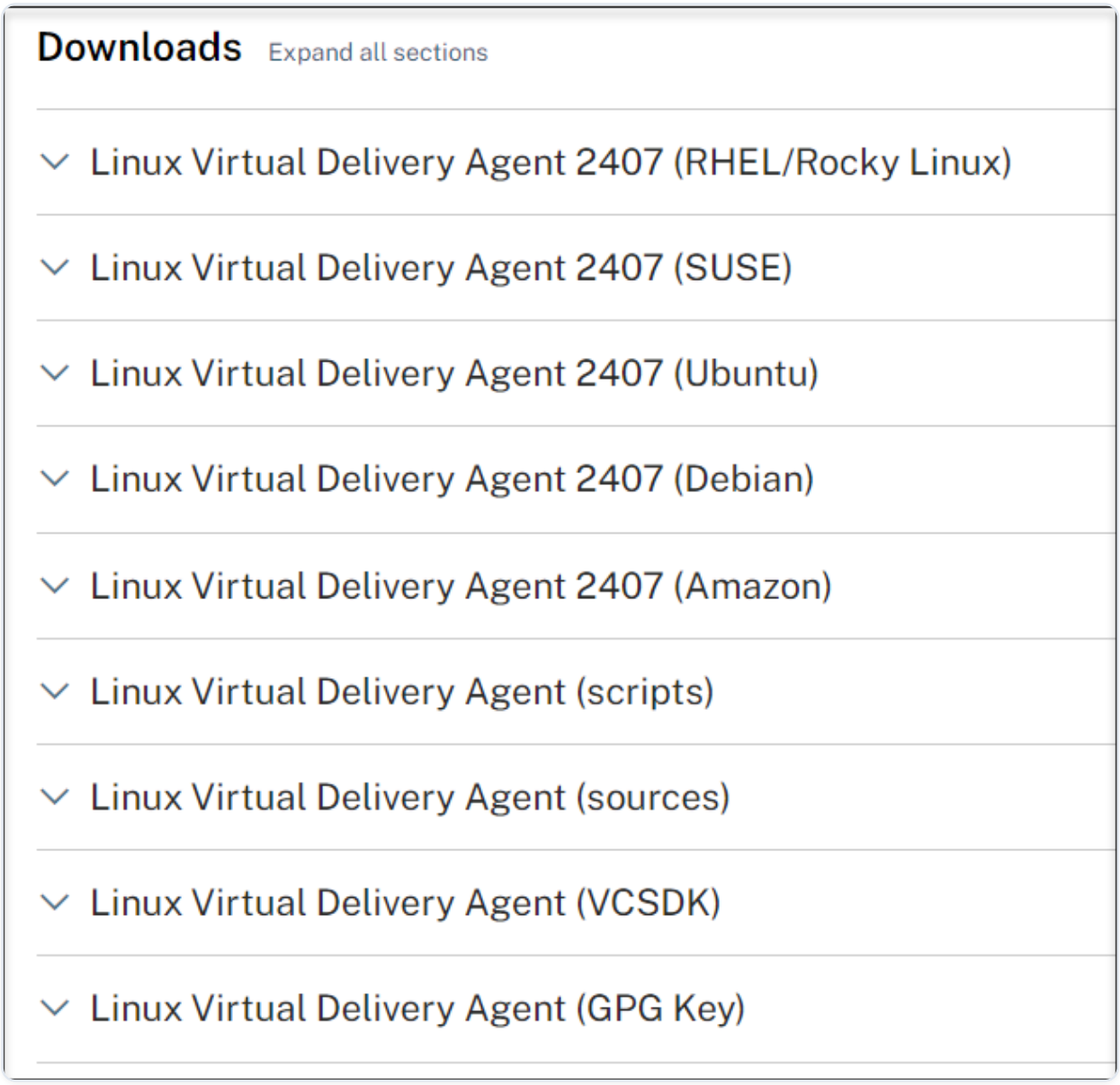
コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が /aa/bb/dotnet の場合、/aa/bb を.NETバイナリパスとして使用します。

手順5：Linux VDAパッケージのダウンロード

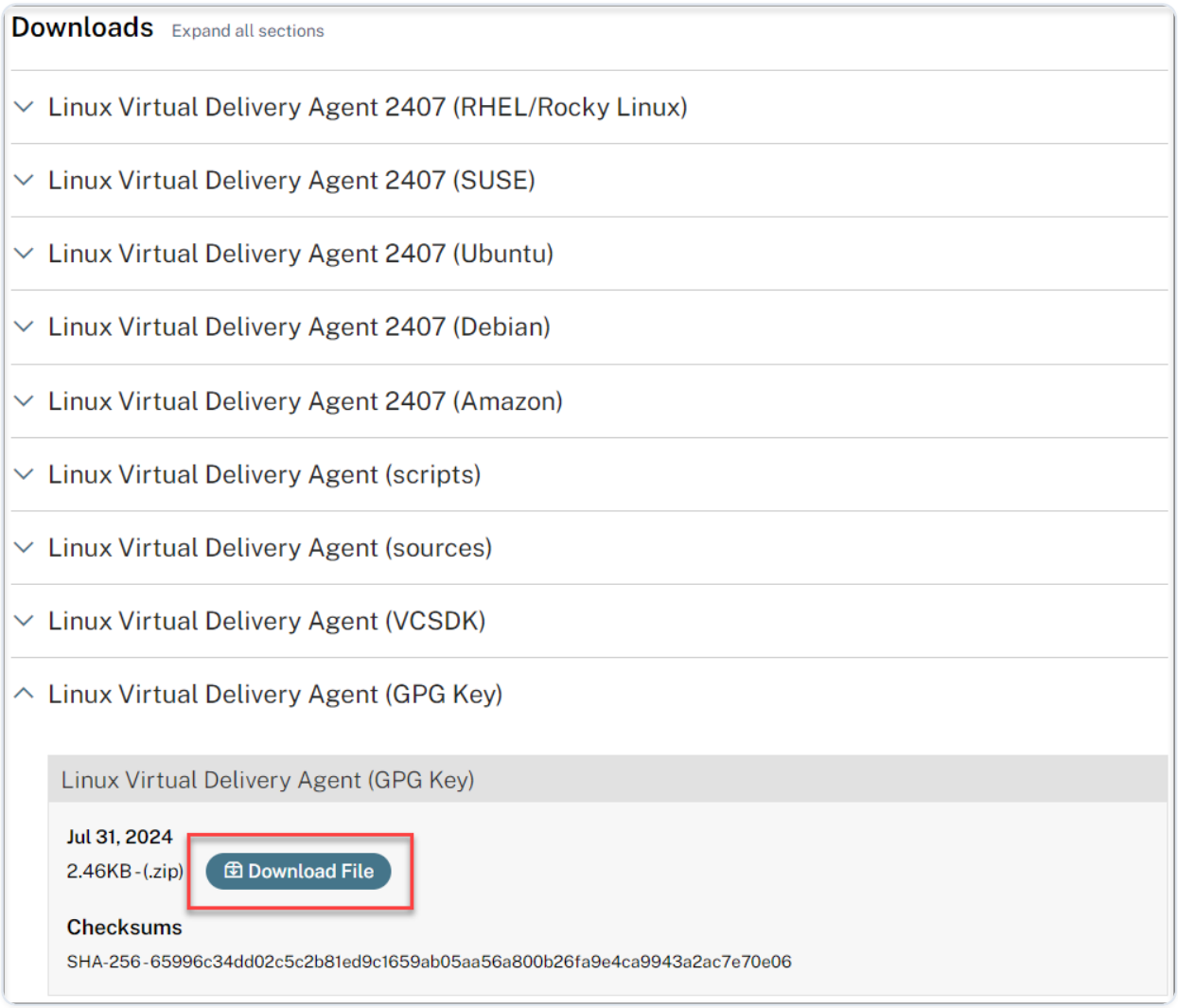
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 適切なバージョンのCitrix Virtual Apps and Desktopsを展開します。
3. **Components** を展開してLinux VDAを見つけます。例：



4. Linux VDAダウンロードにアクセスするには、Linux VDAリンクをクリックします。



5. 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。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開キーをダウンロードします。例：



Linux VDAパッケージの整合性を検証するには、次のコマンドを実行して公開キーをDEBデータベースにインポートし、パッケージの整合性を確認します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

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。

Debian 13の場合：

```
sudo dpkg -i xendesktopvda_<version>.debian13_amd64.deb
```

Debian 12の場合：

```
sudo dpkg -i xendesktopvda_<version>.debian12_amd64.deb
```

Debian 13の依存関係リスト：

```
openjdk-21-jdk >= 21

imagemagick >= 8:6.9.11

ufw >= 0.36

desktop-base >= 12.0.6

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.38

gtk3-nocsd >= 3

bash >= 5.2

findutils >= 4.9.0

sed >= 4.9

pciutils >= 3.13

cups >= 2.4

ghostscript >= 10.0.0~

libmspack0 >= 0.11

ibus >= 1.5

libgoogle-perftools4 >= 2.10~
```

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

libnss3-tools >= 2:3.87

libqt5widgets5 >= 5.15~

mutter >= 43.8

libqrencode4 >= 4.1.1

libfuse2 >= 2.9.9

pulseaudio-utils >= 16.1

traceroute >= 1:2.1.6

libwebkit2gtk-4.1-0 >= 2.50
```

Debian 12の依存関係リスト：

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.11

ufw >= 0.36

desktop-base >= 12.0.6

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.38

gtk3-nocsd >= 3

bash >= 5.2

findutils >= 4.9.0
```

```
sed >= 4.9

cups >= 2.4

ghostscript >= 10.0.0~

libmspack0 >= 0.11

ibus >= 1.5

libgoogle-perftools4 >= 2.10~

libpython3.11 >= 3.11~

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

libnss3-tools >= 2:3.87

libqt5widgets5 >= 5.15~

mutter >= 43.8

libqrencode4 >= 4.1.1

libimlib2 >= 1.10.0

libfuse2 >= 2.9.9

pulseaudio-utils >= 16.1
```

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）

Linux VDA は最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDA を 2308 から 2311 へ、1912 LTSR から 2203 LTSR へアップグレードできます。

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



sudo apt-get install -f
```

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。
- 2407 リリース以降、Linux VDA はアップグレード中に構成ファイルを処理するためにパッケージマネージャーの **rpm** または **dpkg** に委任します。以下に、**rpm** と **dpkg** が構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを **.rpmnew** 拡張子で保存します。
  - **dpkg**: どのように続行するかを対話形式で選択するよう求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを **.dpkg-new** または **.dpkg-dist** として保存しながら、Linux VDA をサイレントアップグレードするには、次のコマンドを使用します。

    ```
    dpkg --force-confold -i package.deb  # Always keep your version,
    then save new package's version as *.dpkg-new or *.dpkg-dist
    ```

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。

- XenServer
- ヴイエムウェア イーエスエックス
- 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 (クラウドベンダーまたは NVIDIA から提供) を VM に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_NON_DOMAIN_JOINED='y\|n'** – マシンをドメインに 参加させるかどうか。デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – Linux VDAが配信コントローラーで認証するには、Kerberos構成設定が必要です。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAが配信コントローラーに登録するために使用する、スペース区切りの配信コントローラーの完全修飾ドメイン名（FQDN）のリストが必要です。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_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_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_START_SERVICE='y\|n'** – 構成完了時にLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシン起動後にLinux Virtual Desktopサービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA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。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/'<none\>'** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。「**<none\>**」に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、デスクトップ切り替えコマンドおよびシステムトレイを参照してください。

- **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_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_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、LDAPサーバーを検出するためにDNSを照会します。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを含むLDAP FQDNのスペース区切りリストを提供できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数は、デフォルトで「**<none\>**」に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPを照会します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、「**<none\>**」に設定します。
- **CTX_XDL_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。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'
```

```
export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

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_SUPPORT_DDC_AS_CNAME='y|n'

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

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。前述のコマンドから、最初の行に「**#\!/bin/bash**」を含むシェルスクリプトファイルを作成することをお勧めします。

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

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \
```

```
CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

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

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

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

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

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

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
```

設定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 stop ctxmonitord** コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

> **Linux VDA を再起動する:**

Linux VDA サービスを再起動するには:

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx
```

```
sudo systemctl restart ctxvda
```

Linux VDA のステータスを確認する:

Linux VDA サービスの実行ステータスを確認するには:

```
sudo systemctl status ctxvda


sudo systemctl status ctxhdx
```

ステップ 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。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッション OS** オプション。
  - VDI 専用デスクトップ配信モデルの場合は、**シングルセッション OS** オプション。
- 同じマシンカタログにLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

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

ヒント:

> マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除して再度追加する必要があります。

ステップ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。

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 2503」を参照してください。

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しく説明します。

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。

- Linux VDA データ収集プログラム
- HDX インサイト
- Citrix Telemetry Service との統合
- Citrix DaaS Standard for Azure 向け Linux VDA のセルフアップデート
- Linux VM および Linux セッションメトリック
- 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
- その他
  - 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**

VDAのアップグレード（プレビュー）

はじめに

以前は、VDAsのアップグレードには完全な手動介入が必要でした。バージョン2503では、VDAアップグレードエージェントを導入することで、DaaS展開におけるVDAアップグレードを簡素化します。バージョン2503以降のアップグレードは、共有またはローカルのファイルパスから直接実行できます。

VDAアップグレードエージェント**ctxvua**は、VDAアップグレードサービスとの通信を担当し、以下の機能を実行します。

- **スケジュールされたチェック**: VDAアップグレードエージェントは、15分ごとにVDAアップグレードサービスにスケジュールされたアップグレード情報を照会します。
- **自動アップグレード**: アップグレード指示を受信すると、VDAアップグレードエージェントはVDAを自動的にアップグレードします。
- **ステータスレポート**: VDAアップグレードエージェントは、アップグレード結果（成功または失敗）をVDAアップグレードサービスに報告します。

VDAアップグレードサービスの詳細については、「Tech Brief: Citrix VDA Upgrade service」を参照してください。そこでは、サービスの概要、その仕組みに関する詳細情報、およびその他の役立つリソースを見つけることができます。

考慮事項

- Linux VDAは、基盤となるパッケージ管理コマンド（rpmやaptなど）を使用してアップグレードされ、手動アップグレードプロセスを反映し、コマンドラインアップグレード中に構成ファイルが自動的に処理されます。
- Windowsとは異なり、Linux VDAには組み込みのVDAアップグレードエージェントが含まれています。これにより、エージェントがすでに存在するため、アップグレードプロセスが簡素化されます。VDAアップグレードエージェントのバージョンは、VDAのバージョンに紐付けられます。
- デフォルトでは、VDAアップグレードエージェントは無効になっています。エージェントを有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\UpdateServices\UpdateAgent" -t "REG_DWORD" -v
"fEnabled" -d "0x00000001" --force
```

```
systemctl start ctxvua.service
```

- VDAアップグレードエージェントサービス（**ctxvua**）はデフォルトで無効になっています。このサービスを有効にして開始するには、**systemctl**を使用できます。
- ベストプラクティスとして、本番環境に移行する前にVDAアップグレードを徹底的にテストすることをお勧めします。
- Windowsとは異なり、Linux VDAのアップグレードはファイルパスからのみサポートされます。これは、Azure CDN URLやその他のオンラインリポジトリを直接使用できないことを意味します。VDAパッケージは自分で管理する必要があります。これは、メジャーバージョンアップグレードとマイナーバージョンアップグレードの両方に適用されます。
- VDAアップグレードサービスにおける「Latest VDA Version」と「Upgrade State」は無視してください。「VDA Upgrade State」のみがLinuxに関連します。
- VDAパッケージのファイルパスは、VDAマシンにローカルであるか、共有場所（たとえば、VDAにマウントされたネットワーク共有）であることができます。システムはパッケージを自動的にダウンロードするようには設計されていません。完全なパッケージファイルを提供する必要があります。
- StudioまたはCitrix DaaS Remote PowerShell SDKを使用する際にパス検証を通過させるには、Windows UNCパス形式（****で始まる）でパスを指定します。例：**/mnt/pkg\/\<package-name>** は **\\\mnt\pkg\\<package-name\>** と入力する必要があります。
- 「サーバー」VDAと「ワークステーション」VDAの区別はLinuxには適用されません。StudioまたはPowerShellでどちらのオプションを使用しても、アップグレードに影響はありません。
- VDAのダウングレードはサポートされていません。

前提条件

- コントロールプレーン：Citrix DaaS™
- VDAバージョン：2503以降

> **注:**
>
> 最新のCR VDAを使用することをお勧めします。

- VDAにはVDAアップグレードエージェントがインストールされており、サービスが実行されている必要があります。

- VDAをアップグレードする権限があること。
- Studioで、VDAアップグレードが適切なCRまたはLTSRトラックで構成されていること。
- VDAが使用中でないこと。（ユーザーはVDAからサインオフする必要があります。）
- VDAがメンテナンスモードでないこと。（VDAは管理者によってメンテナンスモードに設定できます。また、最大許容登録試行回数を超えた場合、VDAは自動的にメンテナンスモードになることがあります。）
- VDAはデリバリーグループに属し、DaaSに登録されている必要があります。
- 移行先のVDAが現在のVDAのオペレーティングシステムをサポートしていること。

Studioを使用したVDAのアップグレード

一般的なワークフロー

Studioを使用してVDAをアップグレードする一般的なワークフローは次のとおりです。

1. カタログのVDAアップグレードを有効にします。
   - カタログの作成時にVDAアップグレードを有効にできます。
   - カタログの編集時にVDAアップグレードを有効にできます。
2. VDAはカタログ単位でアップグレードします。マシンごとのVDAアップグレードは現在利用できません。詳しくは、「VDAの自動アップグレードの構成」を参照してください。

   > **注:**
   >
   > カタログのVDAアップグレードをスケジュールすると、カタログ内のすべてのマシンがアップグレードの対象に含まれます。そのため、アップグレードを開始する前に、これらのマシンをバックアップすることをお勧めします。

3. VDAアップグレードプロセスでは、追加コンポーネントのアップグレードや、復元などの機能の使用はサポートされていません。これら2つの手順はスキップしてください。
4. アップグレード時間やアップグレード失敗のしきい値など、スケジューリングオプションを構成します。失敗のしきい値は、プロセスが停止されるかアラートがトリガーされるまでに許容される失敗したアップグレードの数を決定する可能性があります。
5. VDAインストーラーの場所として「Use local file share」を選択します。パスはWindows UNC形式で指定します（例: **\\\\server\share\path**）。

6. 「Force logoff sessions」オプションは、VDAアップグレード中にユーザーセッションがどのように処理されるかを制御します。Studio UIでは切断されたセッションのみログオフできますが、PowerShellではすべてのセッション（接続済みおよび切断済み）をログオフできます。ログオフは即座には行われません。VDAアップグレードエージェントがアップグレードスケジュールを照会しようとして切断されたセッションを見つけた後、VDAアップグレードサービスがログオフを開始します。その後、エージェントはクエリを再試行する前に15分間待機します。

PowerShellを使用したVDAのアップグレード

WindowsでRemote PowerShell SDKを使用してVDAアップグレードを構成できます。Remote PowerShell SDKの詳細については、「Citrix DaaS Remote PowerShell SDK」を参照してください。

次のPowerShellコマンドレットがあります。

- Get-VusCatalog

  このコマンドレットを使用して、**Name**、**Uid**、**Uuid**、**UpgradeState**（**Available**、**UpToDate**、**Scheduled**、**Unknown**）、**Upgrade scheduled**、および**StateId**（**Upgrade scheduled**のステータス）などのカタログの詳細を取得します。

- Get-VusMachine

  このcmdletを使用して、**MachineName**、**Uid**、**Uuid**、**UpgradeState** (**Available**、**UpToDate**、**Scheduled**、**Unknown**)、および**StateId** (**Upgrade scheduled** のステータス) などのマシンの詳細を取得します。

- Get-VusComponentVersion

  このcmdletを使用して、VDAがコンポーネントバージョンを報告したかどうかを確認します。**MachineId**を使用してVDAをフィルター処理します。**MachineId**は**Get-BrokerMachine**からのUUIDです。

- New-VusMachineUpgrade

  このcmdletを使用して、マシンレベルでVDAアップグレードを構成します。

- New-VusCatalogSchedule

  このcmdletを使用して、マシンカタログレベルでVDAアップグレードをスケジュールします。

例:

```
Get-BrokerMachine -DNSName 'u22-test*'



New-VusCatalogSchedule -CatalogName "test-catalog" -UpgradeNow
-DurationInHours 2 -LogoffOption ActiveAndDisconnectedSessions
-VdaServerPackageUri "\
\root\xendesktopvda_24.11.0.1-1.ubuntu22.04_amd64.deb"



Get-VusComponent -CatalogName 'test-catalog'



Get-VusCatalog -Name 'test-catalog'
```

トラブルシューティング

アップグレードプロセスの核は、VDAアップグレードエージェントサービス（**ctxvua**）を中心に展開されます。これは仲介役として機能し、VDAアップグレードサービスと通信し、OS関連の操作のために**/opt/Citrix/VDA/sbin/update_helper.sh**スクリプトを実行します。アップグレード中、プロセスに関する情報はレジストリに保存されます。

レジストリ

コマンド**/opt/Citrix/VDA/bin/ctxreg dump |grep -i UpdateAgent**を使用して、VDAアップグレードエージェントに関連するレジストリ設定を調べます。これにより、構成の問題やアップグレードプロセス自体の問題が明らかになる可能性があります。

1. **構成の確認: ctxvua**サービスの構成ファイルは**/etc/xdl/updateagent**.confにあります。このファイルを確認することで、誤った構成を特定できます。

ログ

トラブルシューティングには、次のログファイルが重要です。

- **/var/log/xdl/vua.log**: **ctxvua**サービスのログファイル。これは、アップグレードエージェントの操作に関連する問題を確認するための主要なログです。**ctxvua**サービスの構成ファイルは**/etc/xdl/updateagent.conf**にあります。このファイルを確認することで、誤った構成を特定できます。
- **/var/log/xdl/update_helper.log**: **update_helper.sh**スクリプトのログファイル。このログは、アップグレード中のOSレベルのタスクに関連する問題を診断するために不可欠です。

一般的な問題

このセクションでは、VDAアップグレード中に発生する一般的な問題について説明します。特に、Studioで無効になっているオプションと「**Upgrade Unknown**」の状態に焦点を当てています。

一般的な問題 1: 無効なアップグレードオプション

症状: 特定のカタログのStudioで、「アップグレードタイプの選択」と「VDAのアップグレード」のオプションが無効になっています（グレー表示）。

解決策: 使用しているカタログタイプでVDAアップグレードサービスがサポートされているか確認してください。サポートされていない場合、これらの自動アップグレード機能は使用できず、アップグレードを手動で管理する必要があります。

一般的な問題 2: 「アップグレード不明」の状態

症状: マシンカタログでVDAアップグレードサービスを有効にした後、「アップグレード状態」が期待される「利用可能」または「最新」に変わらず、「不明」のままになります。「アップグレード不明」は一時的な状態です。最終的には「利用可能」または「最新」のいずれかに更新されるはずです。

「アップグレード不明」のトラブルシューティング手順：

1. VDAアップグレードエージェントがバージョンを報告していることを確認します。
   - ステップ1a: マシンのUUIDを取得します。

     ```
     Get-BrokerMachine -DNSName '<hostname>'
     ```

   - ステップ1b: エージェントによって報告されたコンポーネントバージョンを確認します。

     ```
     Get-VusComponentVersion -MachineId "<UUID>"
     ```

     Get-VusComponentVersionコマンドが空白を返す場合、VDAアップグレードエージェントがそのバージョンを報告していないことを意味します。これは、VDAが「ハード登録」されている（マシンカタログとデリバリーグループの両方の設定を確認してください）ことを示している可能性があります。また、VDAアップグレードエージェントがターゲットVDAにインストールされていないか、実行されていない可能性も示しています。
2. VDAアップグレードサービスの同期を確認します。

ステップ2a: VDAアップグレードサービスがBrokerデータベースからマシンを同期したかどうかを確認します。

```
```

Get-VusEntityUnit -EntityUUID ""

```
```

実際の**EntityUUID**が分かっている場合は`""`を置き換えるか、すべて取得するために指定せずに実行します。これが空白であると確認された場合、マシンがVDAアップグレードサービスサーバーと同期されていないことを示している可能性があります。

ステップ2b: マシンが同期されていない場合、VDAアップグレードサービスが同期するまでしばらく時間を置きます。その後、「アップグレードタイプ」が設定されていることを確認します。

参照

- 技術概要: Citrix VDAアップグレードサービス
- VDAをアップグレードする(/ja-jp/citrix-daas/upgrade-migrate.html)

Linux VDA データ収集プログラム

Linux VDA のインストール後、データ収集プログラムに自動的に参加します。データ収集プログラムは、統計情報と使用状況データを収集し、Citrix 製品の品質とパフォーマンスの向上に役立てるため、そのデータを Citrix Analytics に送信します。データのアップロードは、セッションを起動したときに発生します。

レジストリ設定

Linux VDA データ収集プログラムは、デフォルトで有効になっています。次のレジストリ設定でプログラムを有効または無効にできます。

場所: HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

名前: **CEIPSwitch**

値: 1 = 無効、0 = 有効（デフォルト）

指定しない場合、Linux VDA データ収集プログラムは有効になります。

次のコマンドを実行して、Linux VDA データ収集プログラムを無効にできます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

Linux VDA から収集されるデータ

次の表に、収集されるデータの種類の例を示します。

データポイント	キー名	説明
マシン GUID	**machine_guid**	データの発信元であるマシンを識別するGUID文字列
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文字列
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文字列
GPUタイプ	**gpu_type**	VDA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
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モードが有効かどうかを示す文字列
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文字列。
HDX™ 3D Proが有効かどうか	**hdx_3d_pro**	マシンでHDX 3D Proが有効になっているかどうかを示す文字列
BCRが有効かどうか	**ビーシーアール**	このマシンでブラウザコンテンツリダイレクト（BCR）が有効になっているかどうかを示す文字列
システムロケール	**system_locale**	このマシンのロケールを示す文字列
ファームID	**farm_id**	ファームIDを示す文字列。
VDA仮想化タイプ	**vda_virtualization**	VDAを作成したハイパーバイザーを示す文字列。
セッションキー	**session_key**	データの発信元とな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文字列: デスクトップまたは `<appname>`
Receiverクライアントタイプ	**receiver_type**	セッションの起動に使用されたCitrix Workspace™アプリのタイプを示す整数。

データポイント	キー名	説明
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文字列
シャドウイング回数	**shadow_count**	セッションがシャドウイング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複合長文字列
スマートカードリダイレクト数	**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リダイレクトが使用された回数を示す整数
ビデオコーデックの種類	**grahpic_video_codec_type**	Thinwire で使用されているビデオコーデックを示す文字列。
ログオン資格情報の種類	**credentials_type**	Linux VDA へのログオンに使用された資格情報の種類を示す整数
透かし	**ウォーターマーク**	セッションの透かしが有効か無効かを示す文字列
透かしの透明度	**watermark_transparency**	透かしの透明度を示す整数
ウォーターマークカスタムテキスト長	**watermark_custom_text_len**	ウォーターマークカスタムテキストの長さを表す整数

データポイント	キー名	説明
MTU	**エムティーユー**	このセッションで最大伝送単位 (MTU) が使用されているかどうかを示す文字列
MTU MSS	**mtu_mss**	最大セグメントサイズ (MSS) を表す整数
ファイル転送	**ファイル転送**	ファイル転送ポリシー設定を表す整数
ファイル転送アップロード回数	**filetrans_upload_count**	セッション中にファイル転送で「アップロード」アイコンが使用された回数を表す整数
ファイル転送ダウンロード回数	**filetrans_download_count**	セッション内でファイル転送のために「ダウンロード」アイコンが使用された回数を示す整数

HDX™ インサイト

概要

Linux VDA は、(/ja-jp/citrix-application-delivery-management-service/analytics/hdx-insight.html)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間の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仮想チャネルの無効化と有効化をそれぞれ行います。

```
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、またはCitrix ADMで誤ったCitrix ADCインスタンスが構成されている場合などです。

- ICA制御仮想チャネルがLinux VDAで開始されていません。

  `ps aux | grep -i ctxctl`

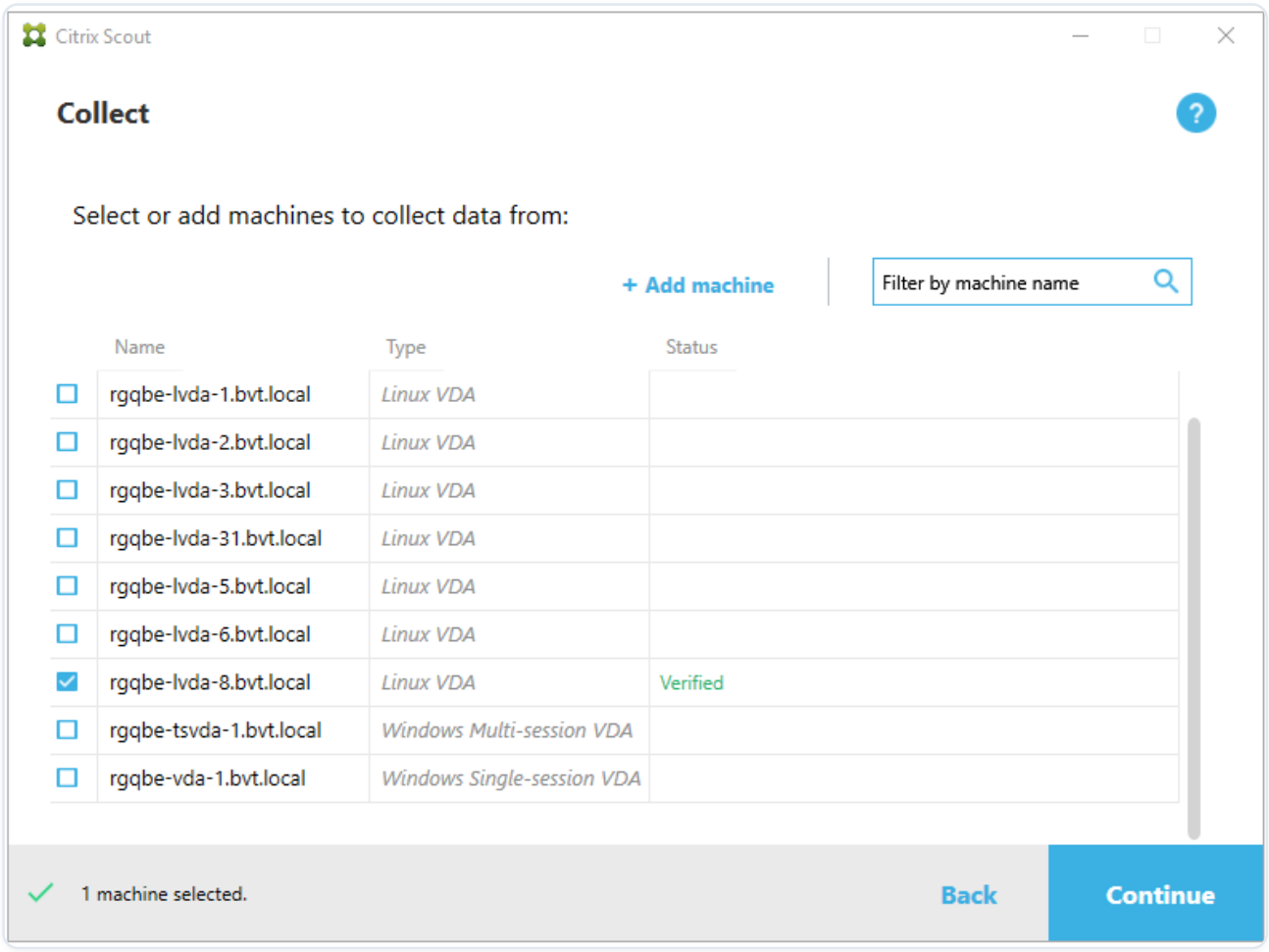
  `ctxctl`が実行されていない場合は、管理者に連絡してCitrixにバグを報告してください。

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`) を統合することで、Citrix Scout を実行し、/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 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 上では TCP/IP ポート 7502 を使用して Citrix Scout と通信します。

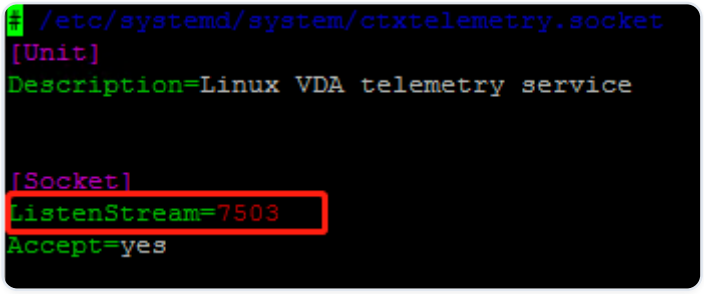
VDA のインストール後にポートを変更するには、次の手順を実行します。

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 create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t REG_DWORD -v
"TelemetryServicePort" -d <port number> --force
```

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 ファイルを開き、編集します。

```
sudo vi /etc/systemd/system/ctxtelemetry.socket
```



3. ソケットポートを再起動するには、次のコマンドを実行します。

```
sudo systemctl daemon-reload

sudo systemctl stop ctxtelemetry.socket

sudo systemctl start ctxtelemetry.socket
```

4. ファイアウォール構成で新しいポートを有効にします。

   たとえば、Ubuntu を使用している場合は、**sudo ufw allow 7503** コマンドを実行してポート 7503 を有効にします。

> **注:**
>
> 代わりに ctxsetup.sh を実行して、前述の手順 3 と 4 を自動化することもできます。

デバッグモード

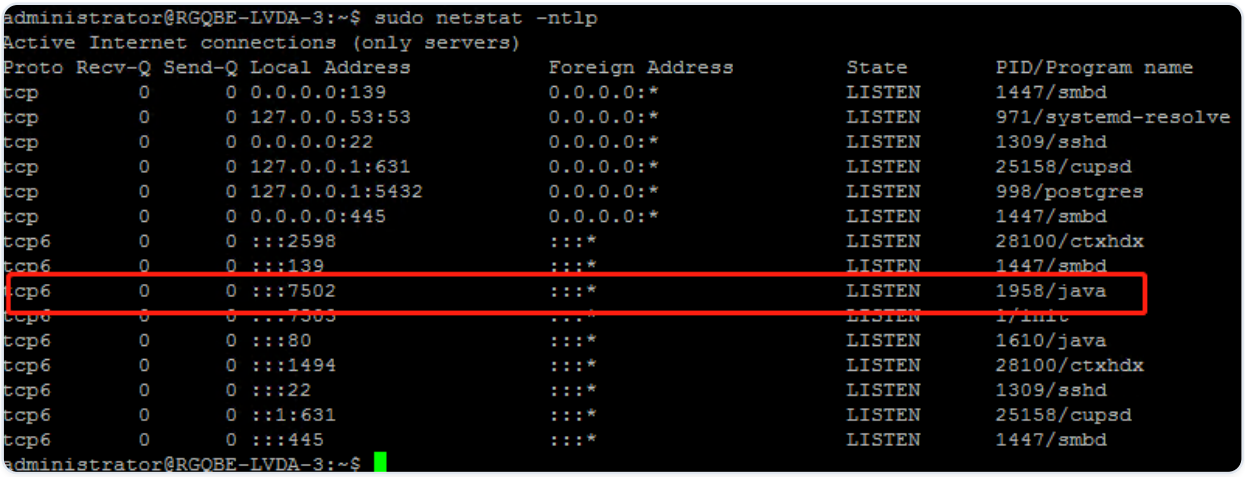
Citrix Telemetry Serviceが期待どおりに動作しない場合は、デバッグモードを有効にして原因を特定できます。

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(/en-us/linux-virtual-delivery-agent/2607-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。



この例では、次のコマンドを実行してCitrix Telemetry Serviceを停止できます。

```
sudo netstat -ntlp

Kill -9 1958
```

3. Citrix Telemetry Serviceを再起動するには、ScoutでLinux VDAを選択し、/var/log/xdl/にあるtelemetry-debug.logを見つけます。

サービス待機時間

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、使用するリソースはわずかです。Citrix Telemetry Serviceはデフォルトで停止しており、Delivery Controllerからのログ収集要求があった場合にのみ起動します。ログ収集が完了すると、サービスは15分間新しい収集要求を待ち、要求がない場合は再度停止します。待機時間は、次のコマンドで構成できます。最小値は10分です。10分未満の値を設定した場合、最小値である10分が適用されます。待機時間を設定したら、サービスを停止して再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"TelemetryServiceIdleTimeoutInMinutes" -d <number> -t REG_DWORD
```

検証テスト

収集が開始される前に、選択された各マシンに対して検証テストが自動的に実行されます。これらのテストにより、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は推奨される修正措置を含む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。

Linux VMおよび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	2402	総ログオン時間は、Linux VDA 2109以降で利用できます。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したものです。**HDX™接続**、**認証**、**GPO**など、ログオンプロセスの各フェーズにかかる時間は、Linux VDA 2402以降で利用できます。ログオン期間のメトリックを表示するには、**トレンド**ビューで**ログオンパフォーマンス**タブを選択します。	Citrix DirectorおよびMonitorで利用可能。
ICA®待機時間	2311	ICA待機時間は、ネットワークが遅いかどうかを示すネットワーク待機時間です。このメトリックにアクセスするに	Citrix DirectorおよびMonitorで利用可能。

メトリック	必要な最小VDAバージョン	説明	備考
		は、**セッションの詳細**ビューを開きます。	
ポリシー	2311	現在のセッションに適用されているすべてのポリシーは、**セッションの詳細**ビューの**ポリシー**タブに表示されます。	Citrix DirectorおよびMonitorで利用可能。
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の数を表示するには、**トレンド**ビューにアクセスします。条件を設定し、**適用**をクリックして検索結果を絞り込みます。**セッション自動再接続数**列には、セッション内の自動再接続の数が表示されます。**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、**自動再接続**が有効になります。セッションの再接続と関連ポリシーの詳細については、次の記事を参照してください。	Citrix DirectorおよびMonitorで利用可能。
		セッション	
		クライアント自動再接続ポリシー設定	

メトリック	必要な最小VDAバージョン	説明	備考
		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、「**フィルター > セッション > すべてのセッション**」を選択して「**すべてのセッション**」ページを開きます。	Citrix DirectorおよびMonitorで利用可能です。
Linux VMのメトリック	2103	Linux VMのメトリックは、CPUコア数、メモリサイズ、ハードディスク容量、および現在と過去のCPUおよびメモリ使用率です。	Citrix DirectorおよびMonitor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**」ビューでUDPまたはTCPとして表示されます。	Citrix DirectorおよびMonitor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 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**	Citrix DirectorおよびMonitor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
		間隔」ポリシーを作成します。	
L7遅延	2507	アプリケーション層（L7）遅延は、NetScalerまたはCGSとCWA/VDAエンドポイントの間でのICAプローブと応答の交換を介して評価されます。	Citrix DirectorおよびMonitorで利用可能です。

ログ収集

ロギングメカニズムの概要

次の表は、Linux VDA のロギングメカニズムの概要を示しています。

ロギングモジュール	ログファイル名の形式	ログファイル名の例	ロギングスコープ	最大単一ファイルサイズ	ローテーションしきい値（デフォルト）	構成
HDX	hdx.log[.n]	hdx.log、hdx.log.1	グラフィック、ログイン、オーディオ、キーボード、マウス	200 MiB	現在1つ、古いもの2つ	setlogユーティリティで構成可能
Jプロキシ	jproxy.log[.n.log]	jproxy.log, jproxy.log.1.log	VDA登録、ユーザー認証	20 MiB	現在の1つ、古い10個	setlogユーティリティまたは/etc/xdl/log4j2.xmlを介して設定可能

ロギングモジュール	ログファイル名の形式	ログファイル名の例	ロギングスコープ	最大単一ファイルサイズ	ローテーションしきい値（デフォルト）	構成
VDA	vda.YYYY-MM-DD.hh.mm.ss.log	vda.2024-05-06.20.18.40.log	VDA登録	50 MiB	現在の1つ、古い1つ	setlogユーティリティまたは/etc/xdl/brokeragent.confを介して設定可能

注:

- 最初のログファイルには名前に番号がなく、後続のファイルには「.n」という番号が付けられます。ここで「n」はファイル番号を表します。例えば、「hdx.log」は最初のHDXログファイルであり、「hdx.log.1」は2番目です。
- 単一のログファイルの最大サイズはメビバイト (MiB) で測定されます。
- 生成中で、まだ単一ファイルの最大サイズに達していないログファイルは「現在の」ログファイルと呼ばれます。「現在の」ログファイルが単一ファイルの最大サイズに達すると、ロールオーバーされて「古い」ログファイルになります。
- 保持できる「古い」ログファイルの数を制限するために、ローテーションしきい値を設定できます。制限に達すると、最も古いログファイルが削除されます。

ログ構成

このセクションでは、上記の表に記載されている詳細を補完する形で、ログ構成に関する追加情報を提供します。

Linux VDAでデフォルトで有効になっているログ

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ています。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。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のトレースを維持したくない場合は、ctxlogdデーモンを停止できます。

/etc/xdl/brokeragent.confを介したVDAログの構成

注:

単一ログファイルの最大サイズなど、他のログパラメーターに深く立ち入ることなく、VDAログのログレベルのみを構成したい場合は、この記事の後半で説明するsetlogユーティリティを使用できます。それ以外の場合は、/etc/xdl/brokeragent.confを使用してください。

VDA上の/etc/xdl/brokeragent.confファイルは、VDAログの構成に利用できます。例:

```
<!-- VDA trace settings -->
<add key="TraceProvider.Type" value="QueuedLogFileProvider"/>
<add key="TraceProvider.BaseFilename" value="/var/log/xdl/vda.log"/>
<add key="TraceProvider.AppendMode" value="true"/>
<add key="TraceProvider.SizeLimit" value="50"/>
<add key="TraceProvider.TrailCount" value="2"/>
<add key="TraceProvider.ProcessSafe" value="true"/>
<add key="TraceProvider.CategoryFilter" value="All" />
```

/etc/xdl/log4j2.xmlを介したJproxyログの構成

注:

単一ログファイルの最大サイズなど、他のログパラメーターに深く立ち入ることなく、Jproxyログのログレベルのみを構成したい場合は、この記事の後半で説明するsetlogユーティリティを使用できます。それ以外の場合は、/etc/xdl/log4j2.xmlを使用してください。

以下は、/etc/xdl/log4j2.xmlを介したJproxyログの構成例です。この例では、**SizeBasedTriggeringPolicy**パラメーターが単一のJproxyログファイルの最大サイズを指定し、**DefaultRollerStrategy**パラメーターが保持できるJproxyログファイルの総数を設定します。

```
<!-- Logging Properties -->
<Properties>
    <Property name="LOG_PATTERN">%d{yyyy-MM-dd HH:mm:ss.SSS}{GMT} [%-5p] [%tid] - %m%n</Property>
    <Property name="logfile">/var/log/xdl/jproxy.log</Property>
    <Property name="perflogfile">/var/log/xdl/jproxyperf.log</Property>
</Properties>

<Appenders>

  <Console name="CONSOLE" target="SYSTEM_OUT">
    <PatternLayout pattern="${LOG_PATTERN}"/>
  </Console>

  <RollingFile name="logfile" fileName="${sys:logfile}"
    filePattern="${sys:logfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
        <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>

  <RollingFile name="perflogfile" fileName="${sys:perflogfile}"
    filePattern="${sys:perflogfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
        <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>
```

setlogユーティリティを介したHDX™ログの構成

setlogユーティリティは/opt/Citrix/VDA/bin/パスにあります。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、値（ログファイルパス、単一ファイルサイズ制限、ローテーションしきい値）やログレベルを含む構成オプションを表示および変更できます。setlogユーティリティのヘルプについては、次のコマンドを実行してください:

```
setlog help
```

値

既定では、HDXログは**/var/log/xdl/hdx.log**に保存され、単一のHDXログファイルのサイズ制限は200 MiBです。また、最大2つの「古い」HDXログファイルを**/var/log/xdl/hdx.log**に保存できます。

現在のすべてのsetlog値を表示するには、次のコマンドを実行します。

```
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するには、次のコマンドを実行します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

既定では、ログレベルは**info** (大文字と小文字を区別しない) に設定されています。

トラブルシューティングが必要な問題に遭遇した場合は、ほとんどのシナリオでログレベルを**verbose**に設定することをお勧めします。その後、問題を再現し、ログを収集してください。

ログレベル (Disabled、Inherited、Trace、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 (大文字と小文字を区別しない)
無効	なし
継承	継承
トレース	トレース
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、all に設定します。例:

```
setlog level all error
```

サポートされているすべてのクラスまたはコンポーネントを表示するには、次のコマンドを実行します:

```
setlog levels
```

デフォルトに戻す

すべてのレベルと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、**ctxlogd** に任意の、または悪意のある構成が適用され、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、VDA の **/tmp/xdlcollect** の下に ZIP ファイルが生成されます。

AOTログの収集とアップロード

概要

この機能は、**Virtual Delivery Agent (VDA)** 上のコンポーネントから **Always-On Tracing (AOT)** ログを収集します。以下をサポートします:

- 再現が困難な問題のトラブルシューティング

- 日常的なヘルスチェック
- 集中ログ管理

ログは構造化されたイベントに変換され、クエリと分析のために一元化された**ログサーバー**にアップロードされます。

構成

グループポリシーの構成

グループポリシーを介してこの機能を有効にするには、**ログサーバー**が**Delivery Controller (DDC)**で適切に構成されていることを確認してください。手順は次のとおりです。

1. Citrix Studioで、 -> -> をクリックします。

AOTログサーバー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/aot-log-server.png)

1. ログサーバーへのログ転送を有効にするには、ログサーバーのアドレスとポートを設定します。

AOTログ構成(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/aot-log-config.png)

ローカルレジストリの構成

ローカルレジストリはポリシーよりも優先されます。ローカルレジストリを介して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（`<ip>`と`<port>`を独自の値に置き換えてください）。

AOTデータ収集を有効にする:

ログサーバーのエンドポイントを設定する:

```
sudo /opt/Citrix/VDA/bin/ctxreg create \

  -k "HKLM\System\CurrentControlSet\Control\Citrix\AOT" \

  -t "REG_DWORD" \

  -v "EnableAotDataCollection" \

  -d "0x00000001" --force



sudo /opt/Citrix/VDA/bin/ctxreg create \
```

```
-k "HKLM\System\CurrentControlSet\Control\Citrix\AOT" \

-t "REG_SZ" \

-v "AotDataStoreEndpoint" \

-d "<http/https>://<fqdn-or-ip-address>:<port>" --force
```

Studioからプッシュされたポリシー設定が変更された場合、またはローカルレジストリ設定が変更された場合、ctxvectorサービスは1分以内に自動的に変更を適用します。手動で再起動する必要はありません。

TLS認証

この機能は、ログサーバーへのHTTPおよびHTTPS接続をサポートしています。HTTPSの場合、ルートCA証明書をLinuxシステムキーストアにインストールします。VDAは、ログサーバー証明書を認証した後にのみHTTPS接続を正常に確立できます。ログサーバーはVDAを認証しません。

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内でユーザーの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:**
>
> HDX画面共有は、シャドウイングの新しい方法であり、現在のシャドウイングに代わるネイティブで低遅延の代替手段を提供します。これには、CR 2511以降のバージョンのCitrix DirectorとLinux VDAの両方が必要です。詳細については、「HDX画面共有の使用」を参照してください。

インストールと構成

依存関係

セッションシャドウイングには、`python-websockify`と`x11vnc`の2つの新しい依存関係が必要です。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 9.x/8.x および Rocky Linux 9.x/8.x の場合：

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

```
sudo 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：

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

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

SUSEの場合：

まず、YaSTまたは次のSUSEConnectコマンドのいずれかを使用して、「SUSE Linux Enterprise Workstation Extension 15 SP7」モジュールを有効にします。

```
suseconnect -p sle-we/15.7/x86_64 -r <regcode>
```

詳細については、SUSEのドキュメントを参照してください。

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

```
sudo pip3 install websockify

sudo zypper install x11vnc
```

Debian 12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。

```
apt install python3-websockify

sudo apt-get install x11vnc
```

ポート

セッションシャドウイング機能は、Linux VDA から `Citrix Director` への接続を確立するために、6001～6099 の範囲から利用可能なポートを自動的に選択します。したがって、同時にシャドウできるICAセッションの数は99に制限されます。特にマルチセッションシャドウイングの場合、要件を満たすのに十分なポートが利用可能であることを確認してください。

レジストリ

次の表に、関連するレジストリを示します。

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションシャドウイング機能を有効または無効にします	1 (有効)
ShadowingUseSSL	Linux VDA と Citrix Director 間の接続を暗号化するかどうかを決定します	0 (無効)

レジストリ値を変更するには、Linux VDA で `ctxreg` コマンドを実行します。たとえば、セッションシャドウイングを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d
"0x00000000"
```

SSL

Linux VDA と Citrix Director 間の noVNC 接続は WebSocket プロトコルを使用します。セッションシャドウイングの場合、`ws://` または `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://` を使用した Linux VDA セッションシャドウイングに関するいかなるセキュリティ責任も負いません。

SSL 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\Software\Citrix\VirtualDesktopAgent" -v "ShadowingUseSSL" -d
"0x00000001"
```

サーバーおよびルートSSL証明書を取得する

証明書は、信頼された認証局 (CA) によって署名されている必要があります。

SSLを構成する各Linux VDAサーバーには、個別のサーバー証明書（キーを含む）が必要です。サーバー証明書は特定のコンピューターを識別するため、各サーバーの完全修飾ドメイン名（FQDN）を知っている必要があります。便宜上、ドメイン全体にワイルドカード証明書を使用することを検討してください。

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組織がどの方法を要求しているかについては、組織のセキュリティチームに相談してください。

> **重要:**
>
> - サーバー証明書の共通名は、Linux VDA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、一部のブラウザーがサポートするにはデジタル証明書の署名として弱すぎます。そのため、SHA-256が最小標準として指定されています。
> - Chromeは、自己署名SSL証明書を安全でないとみなし、受け入れを停止しました。`NET::ERR_CERT_COMMON_NAME_INVALID`エラーは、生成された証明書にSAN（subjectAltName）フィールドがないために発生します。この問題を解決するには、SANフィールドを含む拡張属性（X509 v3拡張）を持つ証明書を提供してください。

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の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、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デフォルトでインポートします。Mozilla Firefoxの場合は、証明書マネージャーの**認証局**タブでルートCA証明書をインポートする必要があります。

各LINUX VDAサーバーにサーバー証明書とそのキーをインストールする

サーバー証明書には「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をパス **/etc/xdl/shadowingssl** の下に置き、**ctxsrvr** のみが読み取りアクセス権を持つように、制限された権限で適切に保護します。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`で接続エラーが発生します。コマンドは次のとおりです。

```
cp <vda's-public-key> /etc/xdl/shadowingssl/shadowingcert.pem

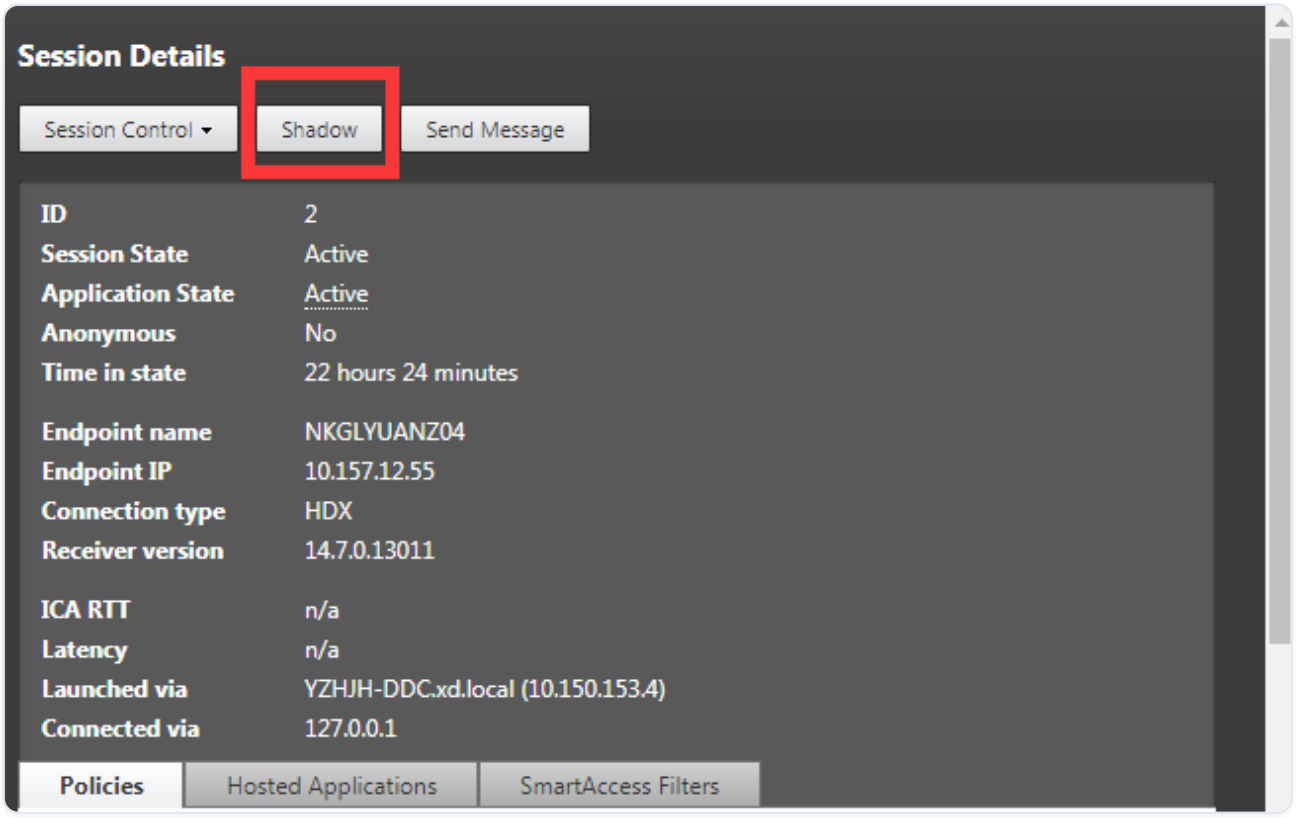
cp <vda's-server-private-key> /etc/xdl/shadowingssl/shadowingkey.key

sudo chown ctxsrvr:ctxadm  /etc/xdl/shadowingssl/shadowingcert.pem

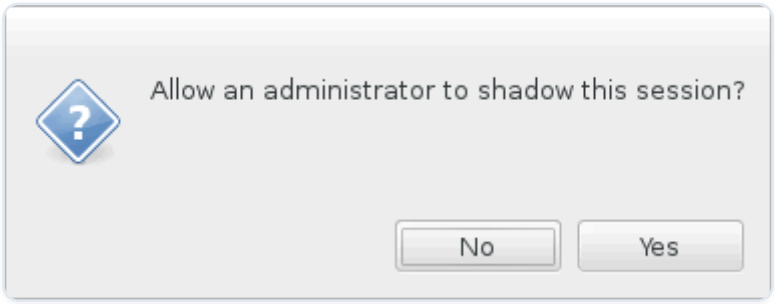
sudo chown ctxsrvr:ctxadm  /etc/xdl/shadowingssl/shadowingkey.key
```

使用方法

`Citrix Director`から、ターゲットセッションを見つけ、**セッションの詳細**ビューで**シャドウ**をクリックして、Linux VDAにシャドウイング要求を送信します。



接続が初期化されると、ICAセッションクライアント (`Citrix Director` クライアントではありません) に確認が表示され、ユーザーにセッションのシャドウを許可するよう求めます。



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 Documentationを参照してください。

制限事項

- VDAがドメインに参加しており、認証にAzure Active Directory (AAD) を使用してMicrosoft Azureでホストされている場合、セッションシャドウ機能は動作しません。
- セッションシャドウは、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に関するいかなる責任も負いません。
- セッションシャドウが有効になっている場合、ドメイン管理者はICAセッションを表示することしかできませんが、書き込みや制御の権限はありません。

- 管理者が`Citrix Director`から**シャドウ**をクリックすると、確認が表示され、ユーザーにセッションのシャドウを許可するよう求めます。セッションは、セッションユーザーが許可を与えた場合にのみシャドウできます。
- 前述の確認には、20秒のタイムアウト制限があります。時間が経過すると、シャドウ要求は失敗します。
- 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のシャドウ要求を送信すると、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。ユーザーが同意した場合、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が構築されます。管理者が同じセッションに対して別のシャドウ要求を送信した場合でも、新しいシャドウ接続を構築できます。
- セッションシャドウを使用するには、`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行ってください。

Citrix Directorクライアントで

ブラウザの開発者ツールを使用して、**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るにもかかわらず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。

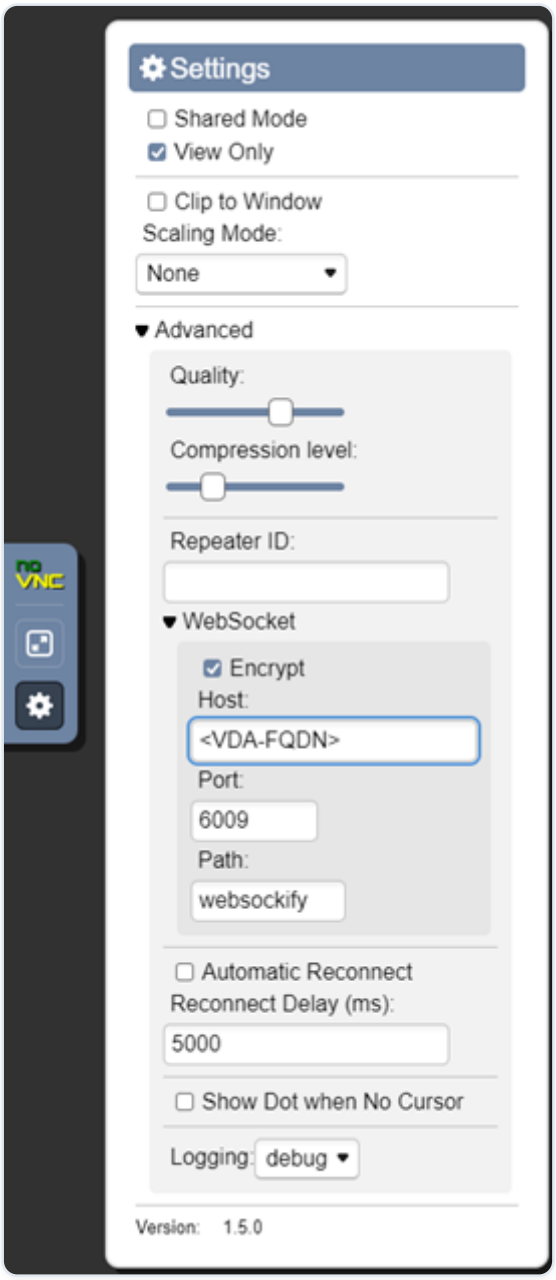
Linux VDA 上で

1. `/var/log/xdl/vda.log`ファイルで手がかりを確認します。
2. `/var/xdl/sessionshadowing.sh`ファイルを編集し、「logFile」変数を変更して、ディレクターからのセッションシャドウイング中に手がかりを追跡できるログファイルを指定します。
3. また、noVNC接続で証明書が正しく機能するかどうかを手動で確認することもできます。
    a. **ps aux \| grep xorg**を実行して、現在のセッションのXorgディスプレイ番号**$display-num**（例: **:3**）を見つけます。
    b. 次のコマンドを実行してx11vncサーバーを起動し、着信接続を待ちます。

> **注:**
>
> 次のコマンドを実行する前に、**$passwd**、**$port**、**$display-num**変数を設定します。

```
runuser -l "ctxsrvr" -s /bin/bash -c "websockify <port> -v --cert  /
etc/xdl/shadowingssl/shadowingcert.pem --key  /etc/xdl/shadowingssl/
shadowingkey.key -- x11vnc  -viewonly  -shared -passwd $passwd
-rfbport $port -display $display-num -many -o /var/log/xdl/
x11vnc.log"
```

c. 次のようにnoVNCを使用して接続し、SSLモードを確認します。VDAのFQDNとポート番号を入力します。この例では、ポート番号は6009です。



d. VDA上のWebsockifyによって出力されたエラー、またはクライアント上のブラウザによって報告されたエラーを解決します。

   接続確立中の主要なチェックポイント：

   i. 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制限がないか確認します。

   ii. SSLシナリオの場合、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たことを確認します。

   iii. 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。

iv. 証明書が正しく構成されていることを確認するために`openssl x509 -in shadowingcert.pem -text -noout`を実行し、特にCNフィールドとSANフィールドに注意してください。

v. RHEL 8では、`rebind.so`が見つからないという問題が発生する可能性があります。この問題を解決するには、次のコマンドを実行します。

```
ln -s /usr/bin/rebind.so /usr/local/bin/rebind.so
```

VDAとデリバリーコントローラー間のWebSocket通信

この記事では、Windows Communication Foundation (WCF) の代替として、VDAsとDelivery Controllers間のWebSocket通信を確立する手順について説明します。

ステップ1：Delivery ControllersでWebSocketを有効にする

1. サイトを構成します。詳しくは、「サイトの作成」を参照してください。
2. サイト上の各Delivery ControllerにTLS証明書をインストールします。詳細については、「ControllerにTLSサーバー証明書をインストールする」を参照してください。
3. 次のコマンドを使用して、各Delivery Controller™でWebSocket通信を有効にします。

```
New-ItemProperty "HKLM:\SOFTWARE\Citrix\DesktopServer\WorkerProxy"
-Name "WebSocket_Enabled" -PropertyType "DWord" -Value 1 -Force
```

注:

WebSocketを有効にした後、Delivery Controllersを再起動してください。

ステップ2：VDAsでWebSocketを有効にする

1. Delivery Controllersを信頼するために、ルートおよび中間証明機関 (CA) をVDAsにインストールします。
2. VDAの作成方法に基づいて、VDAsのWebSocket通信を有効にします。

- **ドメインに参加していないVDA**：

  WebSocket通信はデフォルトで有効になっています。追加の構成は不要です。

- **簡易インストールを使用して作成されたドメイン参加済みVDA**：

簡易インストールスクリプト (ctxinstall.sh) を最初に実行する前に、`/opt/Citrix/VDA/sbin/ctxinstall.conf` で次の環境変数を設定してWebSocketを有効にします。

- **CTX_XDL_DJ_ENROLLMENT_TOKEN_FILE=<file-path\>|'<none\>'** – WebSocketの有効化を制御し、VDA登録用のトークンファイルを指定します。デフォルト値は **'<none\>'**で、WebSocketが無効であることを意味します。ドメイン参加済みVDAでWebSocketを有効にするには、トークンファイルへのパスを入力します。
- **CTX_XDL_ENROLLMENT_TOOL_USING_LDAPS='y|n'** – 登録ツールがLDAPまたはLDAPSのどちらをクエリするかを構成します。デフォルトではLDAP (**'n'**) をクエリします。LDAPSを使用するには、値を**'y'**に設定します。

- **Machine Creation Services™ (MCS) を使用して作成されたドメイン参加済みVDA**:

  テンプレートマシンで `/etc/xdl/mcs/mcs_local_setting.reg` を開き、WebSocketを有効にするために、次のようなコマンドラインを追加します。

  ```
  create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD"
  -v "CbpTransportVersion2" -d "0x00000001" --force
  ```

  このコマンドは、**CbpTransportVersion2**レジストリキーを作成または変更します。デフォルトでは、値は**0**（WCF通信）です。ドメイン参加済みVDAでWebSocket通信を有効にするには、値をゼロ以外の値に設定します。レジストリを変更した後、**ctxvda**サービスを再起動して変更を適用します。

  > **ヒント：**
  >
  > 作成方法に関係なく、任意のドメイン参加済みVDAで、**CbpTransportVersion2**レジストリキーを使用してWCFとWebSocketを切り替えることができます。

監視サービスデーモン

監視サービスである **ctxmonitord** は、Linux VDA のヘルスステータスを監視するデーモンです。これは次の3つのモジュールで構成されています。

- 主要プロセス監視モジュール: このモジュールは、予期せぬ障害発生時に主要サービスを自動的に再起動するよう Systemd に委任し、ターゲットプロセスのステータスを定期的にスキャンして記録し、Xorg の残存ファイルをタイムリーにクリーンアップします。ログは **/var/log/xdl/ms.log** に保存されます。

- XDPing 統合モジュール: このモジュールは、XDPing 分析およびバックアップタスクを定期的に実行します。これらのタスクからのすべての出力は **/var/log/xdl/msxdping.log** で確認できます。
- Linux VDA 自己更新モジュール: 詳しくは、「Azure を介した Linux VDA の自己更新」を参照してください。

構成

デフォルトでは、VDA を起動すると監視サービス **ctxmonitord** が自動的に開始されます。管理者権限があれば、/etc/xdl/ctxmonitord.conf および /etc/xdl/whitelist.conf を介して構成できます。

ctxmonitord.conf

この構成ファイルは、監視サービス ctxmonitord とそのモジュールの動作を指定します。デフォルトでは、次のように構成されています。

```
     ; This is the configuration file for ctxmonitord service




; Section 'service' configures the key process monitor module

[service]

MonitorEnable=true     ; true or false

DetectInterval=300      ; in seconds, minimum is 60



; Section 'xdping' configures the XDPing integration module

[xdping]

XdpingEnable=true      ; true or false

AnalysisInterval=600    ; in seconds, minimum is 60, 0 means disable

BackupInterval=1        ; in days, 0 means disable
```

```
; Section 'rules' configures the rules about how to monitor each key
process

; Each rule should be named as rules.<ProcessName> where <ProcessName> is
the name of the process to be monitored

; 'ProcessName' should be the name of the target process that is picked up
from whitelist.conf

; 'MonitorType' should be 'HealthCheck' or 'ResidueCleanup'. All
'MonitorType' should be 'HealthCheck' except for 'Xorg'

[rules]

[rules.ctxhdx]

ProcessName=ctxhdx

MonitorType=HealthCheck



[rules.ctxvda]

ProcessName=ctxvda

MonitorType=HealthCheck



[rules.ctxjproxy]

ProcessName=ctxjproxy

MonitorType=HealthCheck



[rules.ctxpolicyd]

ProcessName=ctxjproxy

MonitorType=HealthCheck
```

```
[rules.ctxlogd]
ProcessName=ctxlogd
MonitorType=HealthCheck

[rules.xorg]
ProcessName=Xorg
MonitorType=ResidueCleanup
```

whitelist.conf

この構成ファイルは、監視するターゲットプロセスのホワイトリストを指定します。ctxmonitord.conf ファイルで指定されたターゲットサービスは、whitelist.conf ファイルにもリストされている必要があります。デフォルトでは、次のように構成されています。

```
ctxhdx
ctxvda
ctxjproxy
ctxlogd
ctxcdm
ctxcups
ctxpolicyd
ctxusbsd
ctxceip
ctxsdcd
```

```
ctxrunatboot

ctxgdtd

ctxbcrd

ctxfidod

ctxpfwd

ctxwcamsd

ctxcertmgr

ctxscardsd  # Except for SLES

ctxfedsd    # Only for RHEL 8/9

Xorg
```

注:

- ctxmonitord.conf および whitelist.conf ファイルでの変更を有効にするには、**systemctl restart ctxmonitord** コマンドを実行して監視サービスデーモンを再起動します。
- 予期せぬ障害発生時にサービスが再起動する方法を構成するには、**/usr/lib/systemd/system/ctx*.service** ファイルを編集します。たとえば、以下はデフォルトの再起動オプションです。

  ```
  Restart=on-failure

  RestartSec=5

  StartLimitInterval=60

  StartLimitBurst=3
  ```

Citrix Linux Virtual Delivery Agent (LVDA) – ネットワークポートマトリックス

コアVDAの登録と制御

送信元	宛先	プロトコル	ポート	説明
VDA	デリバリーコントローラー	TCP	**80**	VDAの登録とCXF/WCF通信 *1
デリバリーコントローラー	VDA	TCP	**80 / カスタム**	ブローカーが開始するVDA通信 *2

*1 デフォルトでは、VDA登録はVDAとDelivery Controller間のCXF/WCF通信に**TCPポート80**を使用します。このポートが利用できないか制限されている場合、すべてのDelivery Controllerでブローカーサービスを再構成し、VDAが同じポートを使用するように更新することで、登録ポートをカスタマイズできます。VDAとDelivery Controller間のネットワークファイアウォールは、構成されたポートを許可する必要があります。詳細については、Citrixドキュメントを参照してください。

*2 VDAでカスタムCXFサービスポートを構成するには、ctxregコマンドを使用して次のレジストリ値を設定します: `/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "VdaCxfServicesPort" -d "<port>" --force`

サービスを再起動して適用します:

```
systemctl restart ctxjproxy
```

```
systemctl restart ctxvda
```

ICA / HDXセッショントラフィック

送信元	宛先	プロトコル	ポート	説明
クライアント	VDA	TCP	**1494**	ICA / HDX (デフォルト) *3

送信元	宛先	プロトコル	ポート	説明
クライアント	VDA	TCP	**2598**	セッションの信頼性を備えたICA
クライアント	VDA	TCP	**443**	TLS (SSL) 経由のICA
クライアント	VDA	UDP	**443**	HDXアダプティブトランスポート (EDT)

*3 注記 – ICA / HDXのデフォルトポート

ICA / HDX接続は、デフォルトでTCPポート1494を使用します。セッションの信頼性が有効な場合はTCP 2598を使用します。これらのポートは固定されており、任意に変更することはできません。TCP 1494が制限されている場合、CitrixはICAポートを変更する代わりに、TLS経由のICA (TCP 443) またはHDXアダプティブトランスポート (UDP 443) を使用することを推奨します。Citrixドキュメントを参照してください。

HTML5 / Workspace Webアクセス

送信元	宛先	プロトコル	ポート	説明
クライアント	VDA	TCP	**8008**	HTML5 ICAプロキシ (内部アクセス)
クライアント	VDA	TCP	**443**	TLS経由のHTML5 ICA

Linux用HDX™ ダイレクト

以下の表は、内部ユーザーと外部ユーザー向けのデータセンターネットワークについて説明しています。

内部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
直接内部接続	TCP	クライアントネットワーク	1024-65535	VDAネットワーク	443
直接内部接続	UDP	クライアントネットワーク	1024-65535	VDAネットワーク	443

外部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
STUN (外部ユーザーのみ)	UDP	VDAネットワーク	1024-65535	インターネット (下記の注記を参照)	3478, 19302
外部ユーザー接続	UDP	DMZ / 内部ネットワーク	1024-65535	VDAネットワーク	55000-55250
外部ユーザー接続	UDP	VDAネットワーク	55000-55250	クライアントのパブリックIP	1024-65535

VDA ↔ Citrix Scout / テレメトリ (ctxtelemetry)

送信元	宛先	方向	プロトコル	ポート	目的
デリバリーコントローラー	VDA	インバウンド	TCP	**7503**	Citrix Scout が VDA 上の Citrix Telemetry Service (ctxtelemetry) に接続します

送信元	宛先	方向	プロトコル	ポート	目的
VDA	デリバリーコントローラー	アウトバウンド	TCP	**7502**	VDA から Delivery Controller へのテレメトリ通信

注:

- Citrix Telemetry Service (ctxtelemetry) は、デフォルトで **TCP 7503** をリッスンします。
- ポート **7502** は、Scout 通信のために Delivery Controller 側で使用されます。
- これらのポートは、**Citrix Scout / テレメトリ収集**が使用されている場合にのみ必要です。

VDA ローカルデータベースサービス (PostgreSQL)

送信元	宛先	方向	プロトコル	ポート	目的
ローカル VDA	ローカル VDA	ローカル	TCP	**5432**	VDA コンポーネントによって使用されるローカル PostgreSQL サービス

注:

- PostgreSQL は、**デフォルトで TCP ポート 5432** をリッスンします。
- このポートはPostgreSQLによって**VDAホストでローカルに使用されます**。
- 明示的に構成されていない限り、このポートへの外部からのインバウンドアクセスは必要ありません。

- ポートまたはセキュリティスキャン中に、このポートが**VDAシステムでリッスンしている**と表示される場合があります。

HDX画面共有

送信元	宛先	方向	プロトコル	ポート範囲	説明
クライアント	VDA	インバウンド	TCP	**52525-52624**	HDX画面共有データチャネル

注:

- HDX画面共有はデフォルトで無効になっており、Citrixポリシーを介して有効にする必要があります。
- 使用可能なポート範囲は構成可能です。デフォルトの範囲は**TCP 52525–52624**です。
- これらのポートは、標準のICAポート（1494 / 2598）に加えて使用されます。

VDA ↔ シトリックス ディレクター

セッションシャドウイング (Linux VDA)

送信元	宛先	方向	プロトコル	ポート範囲	説明
VDA	Citrix Director (シトリックス ディレクター)	アウトバウンド	TCP	**6001-6099**	セッションシャドウイング (noVNC / WebSocket ベース)
VDA	Citrix Director (シトリックス ディレクター)	アウトバウンド	TCP	**52525-52624**	リモートアシスタンス シャドウ (画面共有ベース)

注:

- セッションシャドウイングはLinux VDAのみでサポートされています。
- この機能は、**6001〜6099**の範囲から利用可能なポートを動的に選択します。
- 同時シャドウセッションの数は、この範囲で利用可能なポートによって制限されます。
- これらのポートは、Linux VDAからCitrix Directorクライアントに到達可能である必要があります。

Citrix Linux VDA – ネットワークポートマトリックスの概要

コアおよび機能依存ポート

ポート / 範囲	プロトコル	スコープ	機能 / 目的
80 / カスタム	TCP	VDA ↔ デリバリーコントローラー	VDA登録 (CXF / WCF)
1494	TCP	クライアント ↔ VDA	ICA / HDX (デフォルト)
2598	TCP	クライアント ↔ VDA	セッションの信頼性を持つICA
443	TCP	クライアント ↔ VDA	セキュアICA / HTML5 ICA
443	UDP	クライアント ↔ VDA	HDXアダプティブトランスポート (EDT)
8008	TCP	クライアント ↔ VDA	HTML5 ICA (内部アクセス)
52525-52624	TCP	VDA ↔ VDA	HDX画面共有。グループポリシー「**HDX screen sharing ports**」で構成可能
52525-52624	TCP	VDA ↔ シトリックス ディレクター	**リモートアシスタンスシャドウ**。グルー

ポート / 範囲	プロトコル	スコープ	機能 / 目的
			プポリシー「Remote assistance ports」で構成可能
6001-6099	TCP	VDA ↔ シトリックス ディレクター	**セッションシャドウイング (Linux VDA)**
7503	TCP	デリバリーコントローラー ↔ VDA	シトリックス スカウト / テレメトリ (ctxtelemetry リスンポート)
7502	TCP	VDA ↔ デリバリーコントローラー	Citrix Scout / テレメトリー通信
5432	TCP	ローカルVDA	ローカルPostgreSQL サービス

VDAでネットワークポートを開く

カスタムCXFサービスポートを設定した後、通信を成功させるために、対応するネットワークポートをVDAホストファイアウォールで許可する必要があります。

以下のセクションでは、サポートされているLinuxディストリビューションでTCPポートを開く方法について説明します。

RHEL 8 / RHEL 9 (ロッキー・リナックス、アルマ・リナックスを含む)

RHELベースのディストリビューションでは、デフォルトのファイアウォールサービスとして **firewalld**を使用します。

- firewalldが実行中であることを確認します。

```
sudo systemctl enable --now firewalld
```

- 必要なTCPポートを開きます (例: `<port>`):

```
sudo firewall-cmd --zone=public --add-port=<port>/tcp --permanent
```

- ファイアウォール設定をリロードします。

```
sudo firewall-cmd --reload
```

- (オプション) ポートが開いていることを確認します。

```
sudo firewall-cmd --query-port=<port>/tcp
```

Ubuntuは、デフォルトのファイアウォール管理ツールとして**ufw (Uncomplicated Firewall)**を使用します。

- ufwが有効になっていることを確認します。

```
sudo ufw enable
```

- 必要なTCPポートを許可します (例: `<port>`):

```
sudo ufw allow <port>/tcp
```

- ファイアウォール規則を再ロードまたは確認します:

```
sudo ufw reload
```

```
sudo ufw status
```

> **注:**
>
> - ファイアウォール構成は、構成されたCXFサービスポートで**インバウンドTCPトラフィック**を許可する必要があります。
> - ファイアウォールの変更は、再ロード後すぐに有効になります。
> - ファイアウォールでポートを開いてもサービスは開始されず、ネットワークトラフィックがVDAに到達することを許可するだけです。

その他

- 上記にリストされているすべてのポートが、すべての展開で必要とされるわけではありません。
- **有効な機能**および**ローカルサービス**に対応するポートのみが使用されます。
- 一部のポート (例: 画面共有およびセッションシャドウイング) は**構成可能または動的な範囲**を使用し、同時セッションのために十分なサイズである必要があります。
- ポート**5432**は**VDAホスト上のローカルPostgreSQLサービス**によって使用され、外部データベース接続を意味するものではありません。

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方法と、**ctxsdcutil** ユーティリティを使用したセッションメトリック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。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 ツールのインストール

ctxsetup.sh を実行しても **XDPing** はインストールされません。**XDPing** をインストールするには、`sudo /opt/Citrix/VDA/bin/xdping` を実行します。 このコマンドは、**XDPing** に必要な **Python3** 仮想環境も作成します。このコマンドで **Python3** 仮想環境の作成に失敗した場合は、**Python3** 仮想環境の作成 の手順に従って手動で作成してください。

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 で実行できるタスク

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。

次の表は、対応する **XDPing** コマンドで実行できるさまざまなタスクについて説明しています。

タスク	XDPING コマンド	備考
コマンドラインオプションを表示するには	**sudo /opt/Citrix/VDA/bin/xdping -h**	該当なし
テストスイート全体を実行するには	**sudo /opt/Citrix/VDA/bin/xdping**（コマンドラインオプションなしで **XDPing** を実行）	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ます。詳細については、この記事の後半にある「個別のテスト」を参照してください。

タスク	XDPING コマンド	備考
VDA登録ステータスチェックを実行するには	**sudo /opt/Citrix/VDA/bin/xdping -a**	詳細については、この記事の後半にある「登録ステータスチェックの範囲」を参照してください。
VDAの主要データをバックアップするには	**sudo /opt/Citrix/VDA/bin/xdping -b**	詳細については、この記事の後半にある「VDAデータのバックアップと比較」を参照してください。
VDAバックアップデータの最新の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff**	詳細については、この記事の後半にある「VDAデータのバックアップと比較」を参照してください。
VDAバックアップデータの特定の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff=<one backup data directory>:<the other backup data directory>**	詳細については、この記事の後半にある「VDAデータのバックアップと比較」を参照してください。
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	**sudo /opt/Citrix/VDA/bin/xdping --preflight**	該当なし
特定のテストカテゴリのみを実行する場合（例：時間、Kerberos、データベーステスト）	**sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos,database** を実行します。	該当なし
特定のDelivery Controllerをプローブする場合	**sudo /opt/Citrix/VDA/bin/xdping -d <Delivery Controller の FQDN>** を実行します。	該当なし
VDAにインストールされている依存関係を表示する場合	**sudo /opt/Citrix/VDA/bin/xdping --query-pkgs** または **sudo /opt/Citrix/VDA/bin/xdping -q**	該当なし
VDAにインストールされている依存関係を表示し、クエリ	**sudo /opt/Citrix/VDA/bin/xdping -f <クエリ結果を保存**	該当なし

タスク	XDPING コマンド	備考
結果を特定のパスに保存する場合	**する特定のパス>** または **sudo /opt/Citrix/VDA/bin/xdping --query-pkgs-to <クエリ結果を保存する特定のパス>**	

個別テスト

Linux **XDPing**ツールは、システム上で150を超える個別テストを実行します。これらは大まかに次のように分類されます。

- Linux VDAのシステム要件が満たされているかを確認します。
- Linuxディストリビューションを含むマシン情報を特定して表示します。
- Linuxカーネルの互換性を確認します。
- Linux VDAの動作に影響を与える可能性のある既知のLinuxディストリビューションの問題を確認します。
- Security-Enhanced Linux (SELinux) のモードと互換性を確認する。
- 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する。
- 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する。
- マシンのホストおよびドメイン名の構成を確認する。
- DNS構成を確認し、ルックアップテストを実行する。
- 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する。サポート対象:
  - ゼノサーバー (旧 シトリックス ハイパーバイザー™)
  - マイクロソフト ハイパーブイ
  - ヴイエムウェア ブイスフィア
- 時刻設定を確認し、ネットワーク時刻同期が動作しているか確認する。
- PostgreSQLサービスが適切に構成され、動作しているか確認する。
- SQLiteが適切に構成され、動作しているか確認する。
- 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する。
- Kerberos構成を確認し、認証テストを実行する。
- グループポリシーサービスエンジンのLDAP検索環境を確認する。

- Active Directory統合が適切に設定されており、現在のマシンがドメインに参加しているか確認する。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確認します。
- 各アクティブまたは切断された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かどうかを確認します。
- 必要な依存関係がインストールされているかどうかを確認します。

サンプル出力

Kerberos テストを実行した際のサンプル出力は次のとおりです。

```
sudo xdping -T kerberos

Root User ------------------------------------------------------------------
    User:         root
    EUID:         0
      Verify user is root                                                [Pass]

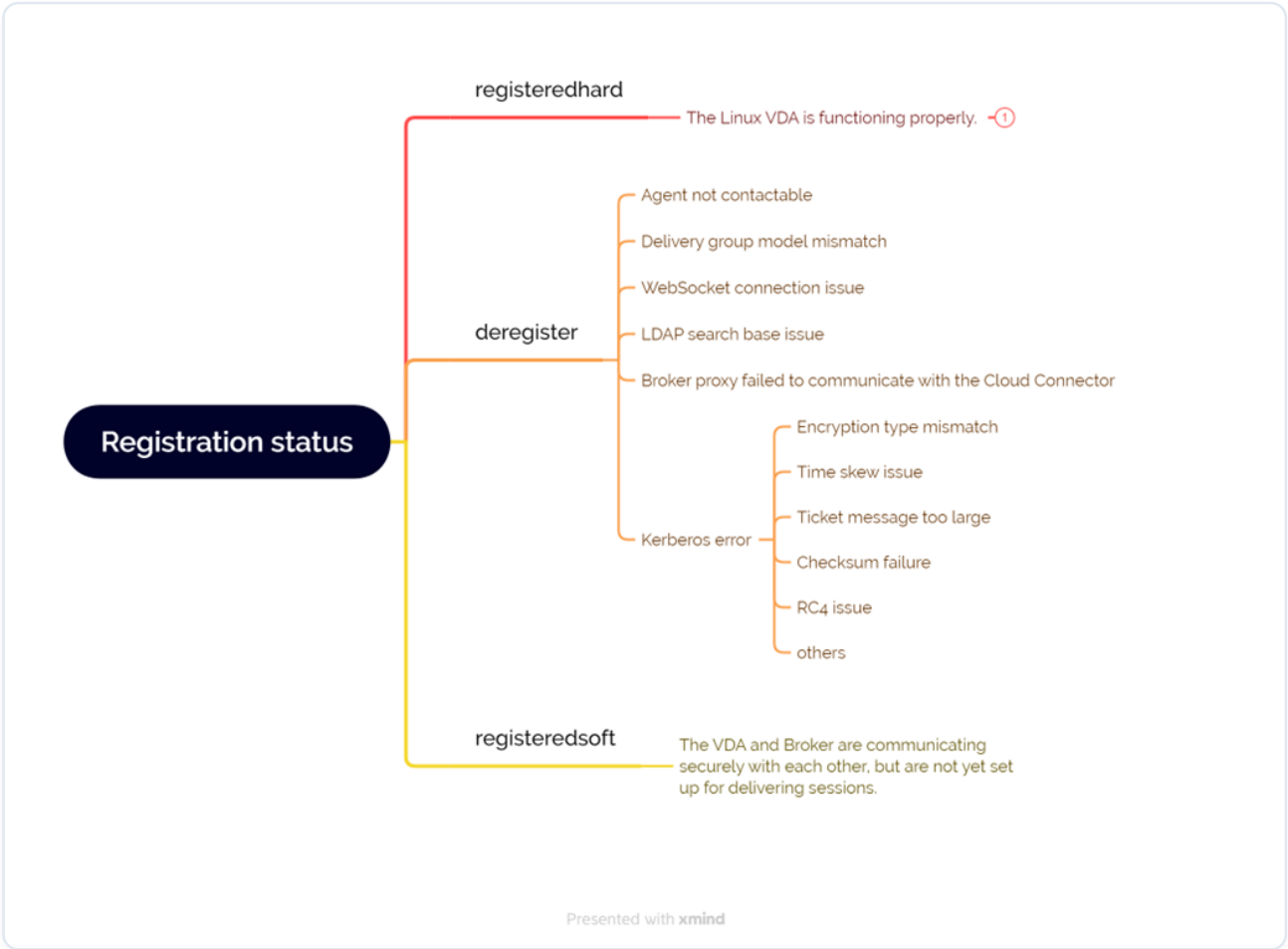
Kerberos -------------------------------------------------------------------
    Kerberos version: 5
      Verify Kerberos available                                          [Pass]
      Verify Kerberos version 5                                          [Pass]
    KRB5CCNAME:    [Not set]
                   Distro default FILE:/tmp/krb5cc_%{uid}
    KRB5CCNAME type: [Supported]
    KRB5CCNAME format: [Default]
      Verify KRB5CCNAME cache type                                       [Pass]
      Verify KRB5CCNAME format                                           [Pass]
    Configuration file: /etc/krb5.conf [Exists]
      Verify Kerberos configuration file found                           [Pass]
  Keytab file:  /etc/krb5.keytab [Exists]
  Default realm: XD2.LOCAL
  Default realm KDCs: [NONE SPECIFIED]
  Default realm domains: [NONE SPECIFIED]
  DNS lookup realm: [Enabled]
  DNS lookup KDC: [Enabled]
  Weak crypto:  [Disabled]
  Clock skew limit:  300 s
    Verify system keytab file exists                                     [Pass]
    Verify default realm set                                             [Pass]
    Verify default realm in upper-case                                   [Pass]
    Verify default realm not EXAMPLE.COM                                 [Pass]
    Verify default realm domain mappings                                 [Pass]
    Verify default realm master KDC configured                           [Pass]
    Verify Kerberos weak crypto disabled                                 [Pass]
    Verify Kerberos clock skew setting                                   [Pass]
  Default ccache: [Not set]
                  Distro default FILE:/tmp/krb5cc_%{uid}
  Default ccache type: [Supported]
  Default ccache format: [Default]
    Verify default credential cache cache type                           [Pass]
    Verify default credential cache format                               [Pass]
  UPN system key [MYVDA1$@[redacted]]: [MISSING]
  SPN system key [host/[redacted]]: [Exists]
    Verify Kerberos system keys for UPN exist                            [ERROR]
    No system keys were found for the user principal name (UPN) of
    the machine account. For the Linux VDA to mutually authenticate
    with the Delivery Controller, the system keytab file must
    contain keys for both the UPN and host-based SPN of the machine
    account.
```

```
      Verify Kerberos system keys for SPN exist                    [Pass]
    Kerberos login: [FAILED AUTHENTICATION]
                    Keytab contains no suitable keys for MYVDA1$@[redacted]
                    while getting initial credentials
      Verify KDC authentication                                    [ERROR]
      Failed to authenticate and obtain a Ticket Granting Ticket (TGT)
      from the KDC authentication service for the machine account UPN
      MYVDA1$@[redacted]. Check that the Kerberos configuration is
      valid and the keys in the system keytab are current.


Summary ------------------------------------------------------------------------
  The following tests did not pass:
      Verify Kerberos system keys for UPN exist                    [ERROR]
      Verify KDC authentication                                    [ERROR]
```

VDA 登録ステータスチェックの範囲

Linux の **XDPing** ツールには、VDA 登録ステータスを確認および分析するのに役立つ分析モジュールも用意されています。登録ステータスチェックの範囲については、次のスクリーンショットを参照してください。



VDA データのバックアップと比較

Linux VDA 2305 以降、**XDPing** ツールに VDA バックアップモジュールが導入されました。このモジュールを使用すると、構成、データベース、バイナリ権限データなど、VDA の主要データをいつでもバックアップできます。VDA が正常に動作しているときに、VDA の主要データをバックアップできます。後で VDA に障害が発生した場合に備えて、データの別のコピーをバックアップし、2つのデータコピーを比較してトラブルシューティングを容易にすることができます。次の表に、対応する **XDPing** コマンドを使用した VDA データのバックアップと比較について説明します。

タスク	XDPING コマンド	備考
VDA の主要データをバックアップするには	VDAデータのバックアップ: **sudo /opt/Citrix/VDA/bin/xdping -b**	バックアップコマンドを実行するたびに、バックアップデータのコピーが生成され、**/var/ctxbackup** の下のディレクトリに保存されます。バックアップデータディレクトリは、**yyyy-mm-dd-hh_mm_ss** 形式の現在の日時で命名されます（例: **2023-02-27-16_31_27**）。デフォルトでは、バックアップデータディレクトリの最大数は30であり、この数を超えると**XDPing**ツールは古いバックアップデータディレクトリをローテーションまたは削除します。ディレクトリローテーションの数をカスタマイズするには、次のコマンドを実行します: `sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Backup" -t "REG_DWORD" -v "MaxDirRotationCount" -d "0x0000005" --force`

タスク	XDPING コマンド	備考
VDAバックアップデータの最新の2つのコピーを比較するには	VDAデータの比較: **sudo /opt/Citrix/VDA/bin/xdping -diff**	該当なし
VDAバックアップデータの特定の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff=<あるバックアップデータディレクトリ>:<もう一方のバックアップデータディレクトリ>**	該当なし

セッションメトリッククエリユーティリティ

ctxセッション

このユーティリティは、Windowsユーザーエクスペリエンスを提供します。このユーティリティを使用すると、次のスクリーンショットに示すようにセッションメトリックにアクセスできます。

```
root@u22:~# /opt/Citrix/VDA/bin/ctxsession -v

            Session Id:    4
             Protocols:    TCP-TLS-CGP-ICA
             User Name:    st\wcx
           Client Name:    BLR3-EWS-WPD030
        Client Address:    10.109.143.143:58535
         Local Address:    10.158.180.61:443
        Remote Address:    10.109.143.143:58535
     Security Protocol:    TLSv1.2
       Security Cipher:    ECDHE-RSA-AES256-GCM-SHA384
       Cipher Strength:    256 bits
        ICA Encryption:    Basic
    Rendezvous Version:    None
       Reducer Version:    4

     ICA Statistics:
        ICA RTT: 0 ms
        Sent Bandwidth: 112 bps
        Received Bandwidth: 0 bps
```

ctxqsession, ctxquser, ctxqfull, ctxquery

これらのユーティリティの使用方法については、ヘルプコマンドを実行してください（例:

```
root@ubuntu2204-33:~# /opt/Citrix/VDA/bin/ctxqsession --help
Usage: ctxqsession [-f short_format_options | -o long_format_options] [-m] [user user_name]
Display information about ICA connections to the local server.

Options:
    -f          Change the displayed information specified by short format options
    -o          Change the displayed information specified by long format options
    -m          Alter the column headers to remove spaces
    user        Display information about the specified user

Short format options:
    d    Client device name (client name)
    i    Session ID
    n    Session name
    p    Published application name
    P    Protocols
    S    Session state
    u    User name
    x    X display number

Long format options(comma separated):
    app      Published application name
    dev      Client device name (client name)
    id       Session ID
    proto    Protocols
    sess     Session name
    state    Session state
    user     User name
    xdpy     X display number
```

ユーザーは自分のセッションメトリックのみを照会できます。ルートユーザーと**ctxadm**グループのユーザーのみが、他のユーザーのセッションメトリックを照会する権限を持っています。

ctxsdcutil

このユーティリティは、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メトリックを照会するのに役立ちます。そのためには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]`コマンドを実行します。`[-c]`引数は、1秒ごとにメトリック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**

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。

- HTML5 対応の Citrix Workspace アプリ
- Python3 仮想環境の作成
- NIS と Active Directory の統合
- IPv6
- LDAPS
- **Xオーソリティ**

Citrix Workspace™ app for HTML5 のサポート

Citrix Workspace app for HTML5 を使用すると、クライアントを Citrix Gateway に接続せずに、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 app for HTML5 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 app 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 の手順 1 を参照してください。

2. WebSocket 接続を有効にします。

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

      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仮想環境を作成できない場合は、ネットワーク接続がなくても手動で作成できます。この記事では、ネットワーク接続なしでPython3仮想環境を作成するための前提条件と手順について詳しく説明します。

前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3`ディレクトリにアクセスするには、管理者権限が必要です。
- `Python3`パッケージのwheelファイルが配置されています。wheelファイルは`<https://pypi.org/>`からダウンロードできます。

Python3仮想環境を作成する

Python3仮想環境を作成するには、次の手順を完了します。

1. Python3の依存関係をインストールします。

   RHELおよびRocky Linuxの場合：

   ```
   yum -y install python3-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 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。RHEL 9の場合は`subscription-manager repos --enable=rhel-9-for-x86_64-appstream-rpms`コマンドを実行します。RHEL 10の場合は`subscription-manager repos --enable=rhel-10-for-x86_64-appstream-rpms`コマンドを実行します。

   Debian、Ubuntuの場合：

```
apt update



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
```

2. Python3仮想環境を作成します。

注:

次のステップでパッケージをダウンロードする際に、pipがSSLエラー（例：**SSLError**、**CERTIFICATE_VERIFY_FAILED**、または**There was a problem confirming the ssl certificate**）で失敗した場合、お使いのマシンはPyPI証明書チェーンを検証できません。これは通常、検査プロキシの背後で発生します。pipがそれらの証明書検証をスキップするように、2つのPyPIホストを信頼済みとしてマークしてください。

`/etc/pip.conf` が存在しない場合は作成し、次を追加します。

```
[global]

trusted-host =

    pypi.org

    files.pythonhosted.org
```

システム全体ではなく単一のコマンドに免除を適用するには、インラインで渡します。

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
trusted-host pypi.org --trusted-host files.pythonhosted.org
&lt;packages>
```

Debian、RHEL、Rocky Linux、Ubuntu をお使いの場合:

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

SUSE の場合:

```
sudo ln -s /usr/lib/mit/bin/krb5-config /usr/bin/krb5-config



export PATH=$PATH:/usr/lib/mit/bin:/usr/lib/mit/sbin



sudo mkdir -p /usr/lib/mit/include/gssapi/



sudo ln -s /usr/include/gssapi/gssapi_ext.h /usr/lib/mit/include/
gssapi/gssapi_ext.h



sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

3. **XDPing** の依存関係をインストールします。

   XDPing は、ディストリビューションではなく、仮想環境を作成したインタープリターの Python3 マイナーバージョンによって選択される2つの異なる依存関係セットに固定されています。まずバージョンを確認してください:

   ```
   /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -V
   ```

   Python 3.11 以降の場合:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
asn1crypto==1.5.1 cffi==2.0.0 cryptography==46.0.6 decorator==5.1.1
gssapi==1.8.3 ldap3==2.9.1 netifaces==0.11.0 packaging==24.1
pg8000==1.31.5 prettytable==3.11.0 psutil==6.0.0 pyasn1==0.6.4
pycparser pyparsing==3.1.4 python-dateutil==2.9.0.post0 scramp==1.4.5
six==1.16.0 termcolor==2.4.0 wcwidth==0.2.13



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

Python 3.10 以前の場合:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
asn1crypto==1.5.1 cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1
gssapi==1.7.3 ldap3==2.9.1 netifaces==0.11.0 packaging==21.3
pg8000==1.26.0 psutil==5.9.0 pyasn1==0.4.8 pycparser==2.21
pyparsing==3.0.8 scramp==1.4.1 six==1.16.0 termcolor==1.1.0
prettytable==2.5.0 wcwidth



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

次の表に、Linux VDA 2607でサポートされている各LinuxディストリビューションのデフォルトのPython3バージョンを示します。デフォルト以外のPython3をインストールした場合は、代わりにこの手順のコマンドで報告されたバージョンを使用してください。

ディストリビューション	デフォルトのPYTHON3
デビアン 13	3.13
デビアン 12	3.11
RHEL 10 および Rocky Linux 10	3.12
RHEL 9.8/9.6 および Rocky Linux 9.8/9.6	3.9
RHEL 8.10 および Rocky Linux 8.10	3.6
SUSE 15.7	3.6
Ubuntu 26.04 および Ubuntu 24.04	3.12 以降
ウブンツ 22.04	3.10

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、アカウント情報（ユーザー名とパスワードの組み合わせ）が AD と NIS で同じである必要があります。

注:

認証は引き続き AD サーバーによって実行されます。NIS+ はサポートされていません。NIS を UID および GID プロバイダーとして使用する場合、Windows サーバーからの POSIX 属性は使用されなくなります。

ヒント:

この方法は、Linux VDA を展開する非推奨の方法であり、特別なユースケースでのみ使用されます。RHEL ディストリビューションの場合は、RHEL および Rocky Linux に Linux VDA を手動でインストールする 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 に Linux VDA を手動でインストールする の手順に従ってください。

> SSSD とは？

SSSD はシステムデーモンです。その主な機能は、システムに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 モジュールと 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サービスを構成します。

```
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。

- `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（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、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に、**\[Global\]**セクションの下に次のエントリを追加します。

`kerberos method = secrets and keytab`

`winbind offline logon = no`

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

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

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

ステップ3：SSSDをセットアップする

> **注**
>
> SSSDをName Service Cache Daemon (NSCD) と一緒に使用すると、予期しない動作が発生する可能性があります。詳細については、「Using NSCD with SSSD」を参照してください。

SSSDのセットアップは、次の手順で構成されます。

- Linuxクライアントマシンに **sssd-ad** および **sssd-proxy** パッケージをインストールします。
- さまざまなファイル（例：**sssd.conf**）に構成変更を加えます。
- **sssdサービス**を開始します。

/etc/sssd/sssd.conf

sssd.conf の構成例（必要に応じてオプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = EXAMPLE

services = nss, pam



[domain/EXAMPLE]

# Uncomment if you need offline logins

# cache_credentials = true

re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?
P<domain>.+$))|(^(?P<name>[^@\\]+)$))

id_provider = proxy

proxy_lib_name = nis

auth_provider = ad

access_provider = ad



# Should be specified as the long version of the Active Directory domain.

ad_domain = EXAMPLE.COM



# Kerberos settings

krb5_ccachedir = /tmp

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

```
# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



# Comment out if the users have the shell and home dir set on the AD side

default_shell = /bin/bash

fallback_homedir = /home/%d/%u



# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available

# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

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

sssd.conf のファイル所有権と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の場合、Linux VDAクライアントに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構成の検証

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

マシンアカウントのTGTチケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist -ke
```

ステップ6：ユーザー認証の検証

getentコマンドを使用して、ログオン形式がサポートされているか、およびNSSが機能しているかを確認します。

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

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

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名：`DOMAIN\username`
- UPN：`username@domain.com`
- NetBIOSサフィックス形式：`username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、ドメインユーザーアカウントを使用してLinux VDAにログオンします。そのドメインユーザーアカウントは以前に使用されたことがないものとします。

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



id -u
```

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

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

ユーザーのKerberosクレデンシャル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

Linux VDA は、Citrix Virtual Apps and Desktops と連携するために IPv6 をサポートしています。この機能を使用する際は、以下を考慮してください。

- デュアルスタック環境では、IPv6 が明示的に有効になっていない限り、IPv4 が使用されます。
- IPv4 環境で IPv6 が有効になっている場合、Linux VDA は機能しません。

> **重要:**

- Linux VDA だけでなく、ネットワーク環境全体が IPv6 である必要があります。
- Centrify は純粋な IPv6 をサポートしていません。

Linux VDA をインストールする際に、IPv6 の特別なセットアップタスクは必要ありません。

Linux VDA の IPv6 の構成

Linux VDA の構成を変更する前に、Linux 仮想マシンが以前に IPv6 ネットワークで動作していたことを確認してください。IPv6 構成に関連するレジストリキーは 2 つあります。

```
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration"

"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"ControllerRegistrationIPv6Netmask"
```

Linux VDA で IPv6 を有効にするには、**OnlyUseIPv6ControllerRegistration** を 1 に設定する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

Linux VDA に複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask** を使用して、Linux VDA の登録に使用するインターフェイスを指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ControllerRegistrationIPv6Netmask " -d "{IPv6 netmask}" --force
```

{IPv6 netmask} を実際のネットマスク (例: 2000::/64) に置き換えます。

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ます。

LDAPS

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。

以下のLinux VDAコンポーネントはLDAPSに依存しています。

- ブローカーエージェント：Delivery Controller™へのLinux VDA登録
- ポリシーサービス：ポリシー評価

LDAPSの構成には以下が含まれます。

- Active Directory (AD) または LDAP サーバーで LDAPS を有効にする
- クライアントで使用するためにルートCAをエクスポートする
- Linux VDA で LDAPS を有効または無効にする
- サードパーティプラットフォーム向けにLDAPSを構成する
- SSSDを構成する
- Winbindを構成する
- Centrifyを構成する
- Questを構成する

注:

LDAPサーバーの監視サイクルを設定するには、次のコマンドを実行します。デフォルト値は15分です。少なくとも10分に設定してください。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --
force
```

AD/LDAPサーバーでLDAPSを有効にする

Microsoftの認証局 (CA) またはMicrosoft以外のCAから適切にフォーマットされた証明書をインストールすることで、LDAP over SSL (LDAPS) を有効にできます。

> **ヒント:**
>
> ドメインコントローラーにエンタープライズルートCAをインストールすると、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で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で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をエクスポートする方法」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**スクリプトを実行します（管理者としてログオンしている間に）。**enable_ldaps.sh**スクリプトを非対話型モードで実行するには、以下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にLDAPSを構成する

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署名と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ています。影響を受ける構成ファイルは、**/etc/samba/smb.conf** および **/etc/openldap/ldap.conf**（Amazon Linux 2、RHEL、Rocky Linux、CentOS、SUSEの場合）、または **/etc/ldap/ldap.conf**（Debianおよび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** の構成を変更します。

`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、次の問題が発生する可能性があります。

- **LDAPSサービスの可用性**

  AD/LDAPサーバーでLDAPS接続が利用可能であることを確認してください。ポートはデフォルトで636です。

- **LDAPSが有効になっている場合にLinux VDAの登録に失敗しました**

  LDAPサーバーとポートが正しく構成されていることを確認してください。まずルートCA証明書を確認し、それがAD/LDAPサーバーと一致していることを確認してください。

- **誤ったレジストリ変更**

  enable_ldaps.shを使用せずに誤ってLDAPS関連のキーを更新した場合、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、LDAPトラフィックがSSL/TLSを介して暗号化されていない**

  デフォルトでは、LDAPSは無効になっています。強制的に有効にするには、**/opt/Citrix/VDA/sbin/enable_ldaps.sh**を実行してください。

- **Wiresharkまたはその他のネットワーク監視ツールからLDAPSトラフィックがない**

  LDAP/LDAPSトラフィックは、Linux VDAの登録とグループポリシーの評価が行われるときに発生します。

- **ADサーバーでldp connectを実行してLDAPSの可用性の確認に失敗しました**

  IPアドレスの代わりにAD FQDNを使用してください。

- **/opt/Citrix/VDA/sbin/enable_ldaps.shスクリプトを実行してルートCA証明書のインポートに失敗しました**

  CA証明書のフルパスを指定し、ルートCA証明書が正しいタイプであることを確認してください。これは、サポートされているほとんどのJava Keytool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証明書の表示に失敗しました**

  `/opt/Citrix/VDA/sbin/enable_ldaps.sh`を実行してLDAPSを有効にすると、証明書は/etc/xdl/.keystoreにインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

エックスオーソリティ

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ています。この機能は、XClient と XServer 間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信のアクセス許可を保護するには、2つの方法があります。

- **Xhost**。デフォルトでは、Xhost はローカルホストの 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は、特定のコンピューターでアクセス許可を付与するために Xhost コマンドを実行する必要があります。または、代わりに **xhost +** を使用して、任意の XClient が XServer に接続できるようにすることもできます。
- **Xauthority**。`.Xauthority` ファイルは、各ユーザーのホームディレクトリにあります。これは、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) が起動すると、Cookie はその特定のディスプレイへの接続を認証するために使用されます。

仕組み

Xorg が起動すると、`.Xauthority` ファイルが Xorg に渡されます。この `.Xauthority` ファイルには、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 XServer に設定します。

- Xorg 内の 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に手動で渡します

  ICA®セッションを起動すると、Linux VDAはXClient用の`.Xauthority`ファイルを生成し、ログオンユーザーのホームディレクトリに保存します。この`.Xauthority`ファイルをリモートのXClientマシンにコピーし、**DISPLAY**および**XAUTHORITY**環境変数を設定できます。`DISPLAY`は`.Xauthority`ファイルに保存されているディスプレイ番号で、`XAUTHORITY`は**Xauthority**のファイルパスです。例については、次のコマンドを参照してください。

  ```
  export DISPLAY={Display number stored in the Xauthority file}



  export XAUTHORITY={the file path of .Xauthority}
  ```

  注:

  XAUTHORITY環境変数が設定されていない場合、デフォルトで`~/.Xauthority`ファイルが使用されます。

- 共有ホームディレクトリをマウントして`.Xauthority`ファイルをXClientに渡します

  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`ファイルが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、次のトラブルシューティング手順に従ってください。

1. root権限を持つ管理者として、すべてのXorgクッキーを取得します。

   ```
   ps aux | grep -i xorg
   ```

   このコマンドは、Xorgプロセスと、起動時にXorgに渡されたパラメーターを表示します。別のパラメーターは、どの`.Xauthority`ファイルが使用されているかを表示します。例：

   ```
   /var/xdl/xauth/.Xauthority110
   ```

   Xauthコマンドを使用してクッキーを表示します。

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   ```

2. `Xauth`コマンドを使用して、`~/.Xauthority`に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーはXorgとXClientの`.Xauthority`ファイルで同じである必要があります。
3. クッキーが同じ場合、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。

   例えば、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。

オーディオ

- オーディオ診断ツール

オーディオ診断コマンドラインツール

VDA上のオーディオ診断コマンドラインツールは、オーディオポリシー、構成、データ転送に関連するセッションデータを照会するために使用できます。

使用方法

1. コマンドプロンプトを開き、`/opt/Citrix/VDA/bin`フォルダーから`ctxaudiosession`を実行します。
    - ツールを実行すると、現在のユーザーのアクティブなICAセッションのオーディオ情報とデバイスリダイレクトステータスがすべて表示されます。

出力

このツールは、セッション内のオーディオ関連の問題の診断に役立つさまざまな構成設定を出力します。

セクション	説明
DDCポリシー情報	現在のセッションに適用されているオーディオポリシー。
ローカルレジストリ情報	レジストリに保存されているオーディオ関連の構成設定。
状態情報	現在のセッションに適用されるオーディオの状態、バージョン、コーデック、トランスポート、およびデバイス番号。
デバイス情報	セッションで使用されているデバイス名、その役割、およびステータス。

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。

- Azure Active Directory を使用した認証
- ダブルホップシングルサインオン認証

- フェデレーション認証サービス
- SSO 以外の認証
- スマートカード
- 認証されていない（匿名）ユーザーによるアクセス

Azure Active Directory を使用した認証

> **注:**
>
> この機能は、Azure でホストされている VDA でのみ利用可能です。

> 必要に応じて、Azure に 2 種類の Linux VDA を展開できます。

- Azure AD DS に参加している VM。VM は Azure Active Directory (AAD) Domain Services (DS) マネージド ドメインに参加しています。ユーザーはドメイン資格情報を使用して VM にログオンします。
- ドメインに参加していない VM。VM は AAD ID サービスと統合してユーザー認証を提供します。ユーザーは AAD 資格情報を使用して VM にログオンします。

AAD DS および AAD の詳細については、この Microsoft の記事を参照してください。

この記事では、ドメインに参加していない VDA で AAD ID サービスを有効にして構成する方法について説明します。

サポートされているディストリビューション

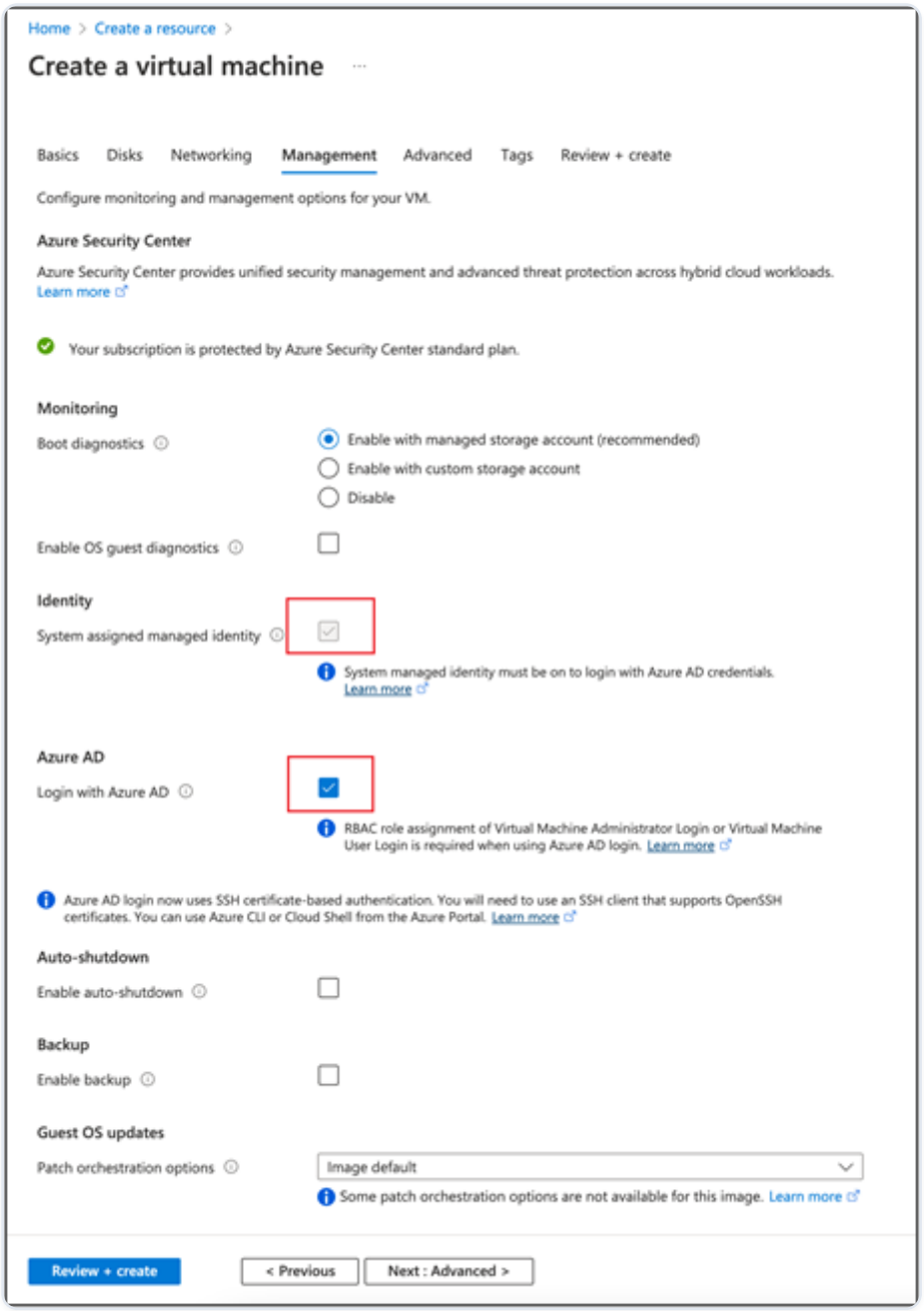
- ウブントゥ ２４．０４、２２．０４
- RHEL 8.10
- SUSE 15.7

詳細については、このMicrosoftの記事を参照してください。

ステップ 1：Azure portal でテンプレート VM を作成する

テンプレート VM を作成し、VM に Azure CLI をインストールします。

1. Azure portal でテンプレート VM を作成します。**Review + create** をクリックする前に、必ず **Management** タブで **Login with Azure AD** を選択してください。



2. テンプレートVMにAzure CLIをインストールします。 詳細については、このMicrosoft記事を参照してください。

ステップ2：テンプレートVMでマスターイメージを準備する

マスターイメージを準備するには、Use MCS to create Linux VMs on Azureの「**ステップ3：マスターイメージを準備する**」に従ってください。

ステップ3：テンプレートVMを非ドメイン参加モードに設定する

マスターイメージを作成したら、以下の手順に従ってVMを非ドメイン参加モードに設定します。

1. コマンドプロンプトから次のスクリプトを実行します。

```
Modify /var/xdl/mcs/mcs_util.sh
```

2. `function read_non_domain_joined_info()`を見つけて、`NonDomainJoined`の値を`2`に変更します。例については、次のコードブロックを参照してください。

```
function read_non_domain_joined_info()

{

log "Debug: Enter read_non_domain_joined_info"

# check if websocket enabled

TrustIdentity=`cat ${id_disk_mnt_point}${ad_info_file_path} | grep '\
[TrustIdentity\]' | sed 's/\s//g'`

if [ "$TrustIdentity" == "[TrustIdentity]" ]; then

NonDomainJoined=2

fi

...

}
```

3. 変更を保存します。
4. テンプレートVMをシャットダウンします。

ステップ4：テンプレートVMからLinux VMを作成する

非ドメイン参加のテンプレートVMの準備ができたら、以下の手順に従ってVMを作成します。

1. Citrix Cloudにサインインします。
2. Citrix DaaS™をダブルクリックし、Full Configuration管理コンソールにアクセスします。
3. **マシンカタログ**で、Machine Creation Servicesを使用してテンプレートVMからLinux VMを作成することを選択します。詳細については、Citrix DaaSの記事非ドメイン参加を参照してください。

ステップ5：AADユーザーアカウントをLinux VMに割り当てる

非ドメイン参加のVMを作成したら、AADユーザーアカウントをそれらに割り当てます。

VMにAADユーザーアカウントを割り当てるには、次の手順に従います。

1. 管理者アカウントを使用してVMにアクセスします。
2. **>** タブで、 を有効にします。

   AADでのログインを有効にする設定(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/aad-auth-identity-setting.png)
3. **>** タブで、 領域を見つけ、必要に応じてAADユーザーアカウントを追加します。

   AADでのログインを有効にする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/aad-auth-assign-aad-account.png)

ドメインに参加していないVDAへのログオン

組織のエンドユーザーは、ドメインに参加していないVDAに2つの方法でログオンできます。詳細な手順は次のとおりです。

1. Workspaceアプリを起動し、AADユーザー名とパスワードを入力してワークスペースにログオンします。Workspaceページが表示されます。
2. ドメインに参加していないデスクトップをダブルクリックします。AADログインページが表示されます。

   ページは、VDAに設定されているログインモード（デバイスコードまたはAADアカウント/パスワード）によって異なります。デフォルトでは、Linux VDAは次のようにデバイスコードログインモードを使用してAADユーザーを認証します。管理者として、必要に応じてログインモードをAADアカウント/パスワードに変更できます。詳細な手順については、次のセクションを参照してください。

   AADでのログインを有効にする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/aad-auth-device-code.png)
3. 画面の指示に従って、次のいずれかの方法でデスクトップセッションにログオンします。

   - QRコードをスキャンしてコードを入力します。
   - AADユーザー名とパスワードを入力します。

AADアカウント/パスワードログインモードへの変更

デフォルトでは、Linux VDAはデバイスコードを使用してAADユーザーを認証します。詳細については、このMicrosoftの記事を参照してください。ログインモードを*AADアカウント/パスワード*に変更するには、次の手順に従います。

VDAで次のコマンドを実行し、キー`AADAcctPwdAuthEnable`を見つけて、その値を`0x00000001`に変更します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Services\CitrixBrokerAgent\WebSocket" -t
"REG_DWORD" -v "AADAcctPwdAuthEnable" -d "0x00000001" --force
```

注:

このアプローチは、Microsoftアカウントや2要素認証が有効になっているアカウントでは機能しません。

二重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、ユーザー資格情報を再度入力することなく、クライアントを使用して仮想デスクトップおよびアプリケーションにアクセスできます。

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。

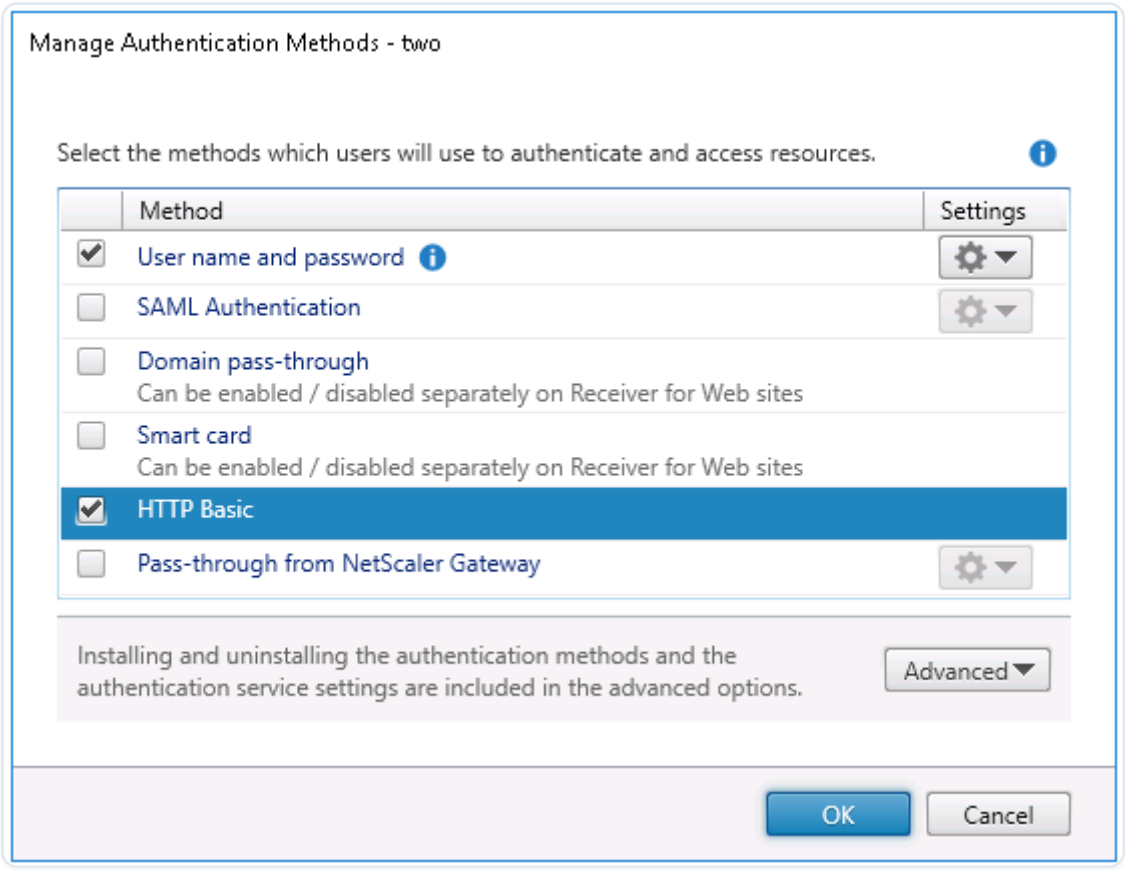
この機能を有効にするには：

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、ターゲットストアの**HTTP Basic**認証方法を追加します。



3. HTTP基本認証を許可するために、AuthManager構成ファイル（$ICAROOT/config/AuthManConfig.xml）に次のキーを追加します。

```
<Protocols>

     <HTTPBasic>

         <Enabled>True</Enabled>

    </HTTPBasic>

</Protocols>
```

4. 指定されたディレクトリにルート証明書をインストールするには、次のコマンドを実行します。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. この機能を有効にするには、次のコマンドを実行します。

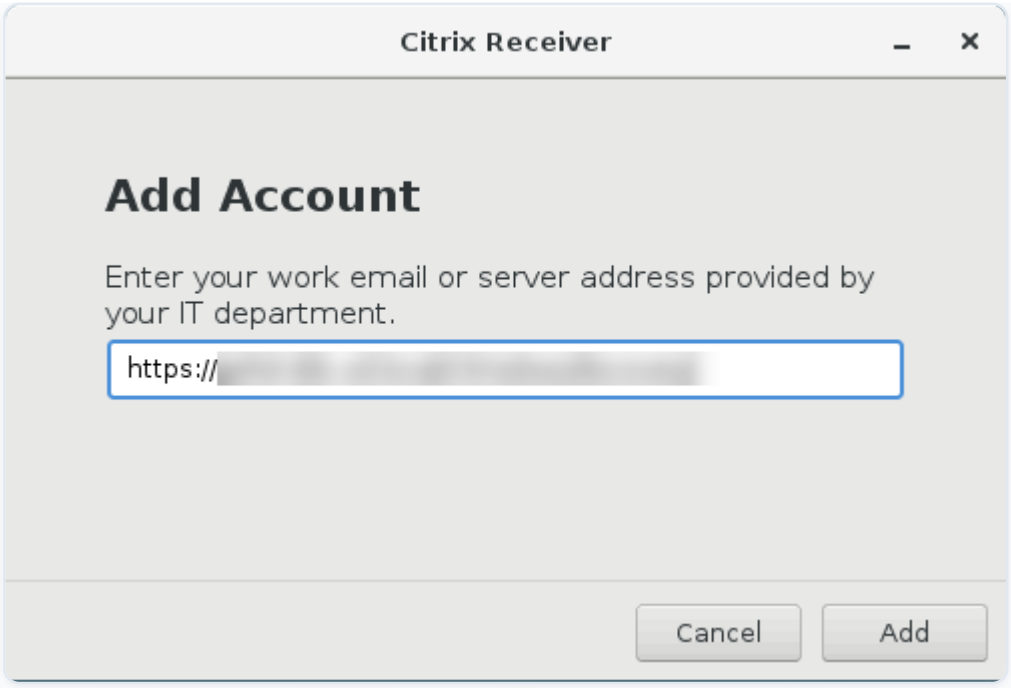
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

6. Linux仮想デスクトップセッションを起動し、そのセッション内でLinux向けCitrix Workspaceアプリまたは Linux向けCitrix Receiver 13.10を開始します。

   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、以前に指定したストアに自動的にログオンされます。

   注:

   ストアアカウントとしてHTTPS URLを入力します。



フェデレーション認証サービス

フェデレーション認証サービス (FAS) を使用して、Linux VDAにログオンするユーザーを認証できます。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に固有の追加情報を提供します。

注:

- Linux VDAは、**セッション内動作**ポリシーをサポートしていません。
- Linux VDAは、FASサーバーとのデータ送受信に短い接続を使用します。

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ディストリビューションとドメイン参加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
デビアン 13.6	はい	はい	はい	はい	いいえ
デビアン 12.15	はい	はい	はい	はい	はい
RHEL 10.2/9.8/9.6	はい	はい	はい	いいえ	はい
RHEL 8.10	はい	はい	はい	はい	はい
ロッキー Linux 10.2/9.8/9.6	はい	はい	はい	いいえ	はい
ロッキー リナックス 8.10	はい	はい	はい	いいえ	はい
SUSE 15.7	はい	はい	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
ウブントゥ 26.04	はい	はい	いいえ	いいえ	いいえ
ウブントゥ 24.04	はい	はい	はい	いいえ	はい
ウブントゥ 22.04	はい	はい	はい	はい	はい

注:

1. `/etc/krb5.conf`が存在しない場合。そのようなシナリオでは、まず「krb5-pkinit」と「libpam-krb5」を手動でインストールしてください（これによりKerberos情報が求められ、krb5.confに書き込まれます）。その後、fasスクリプト「/opt/Citrix/VDA/sbin/ctxfascfg.sh」を実行します。
2. RHELおよびRocky Linux 10.x/9.xの場合、`AD-SUPPORT`暗号化ポリシーを有効にします。

   `update-crypto-policies --set DEFAULT:AD-SUPPORT`

Active DirectoryがKerberos認証中にSHA-1を使用する場合、SHA-1のサポートも有効にする必要があります。

RHELおよびRocky Linux 9.xでは、組み込みの`SHA1`サブポリシーを有効にします。

```
`update-crypto-policies --set DEFAULT:AD-SUPPORT:SHA1`
```

RHELおよびRocky Linux 10.xでは、組み込みの`SHA1`サブポリシーは提供されなくなりました。可能な場合は、Active Directory CAおよびドメインコントローラー証明書をSHA-256で再発行することをお勧めします。上記のサブポリシーは、SHA-1の使用を継続する必要がある環境向けです。`AD-SUPPORT-LEGACY`は代替手段ではありません。これはKerberos暗号化タイプを広げるだけであり、SHA-1署名を再度有効にするものではありません。

SHA-1の使用を継続するには、独自の同等のサブポリシーを作成し、それを適用します。

```
sudo tee /etc/crypto-policies/policies/modules/AD-SHA1.pmod > /dev/
null <<'EOF'

hash = SHA1+

sign = ECDSA-SHA1+ RSA-PSS-SHA1+ RSA-SHA1+

sha1_in_certs = 1

EOF

sudo update-crypto-policies --set DEFAULT:AD-SUPPORT:AD-SHA1
```

新しいポリシーは、その後起動されたアプリケーションに適用されます。すべての場所に適用するには、再起動してください。

Linux VDAでのFASの構成

証明書のインストール

ユーザーの証明書を検証するために、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**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`からダウンロードします。

注:

以下のコマンドは、中間証明書の設定にも適用されます。

たとえば、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 mkdir -p /etc/pki/CA/certs/

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

注:

ルートCA証明書を **/root** パスに配置しないでください。そうしないと、FASはルートCA証明書に対する **read** 権限を持ちません。

ctxfascfg.sh を実行する

ctxfascfg.sh スクリプトを実行してFASを構成します。

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

`ctxfascfg.sh` をサイレントモードで実行できます。スクリプトをサイレントモードで実行する前に、以下の環境変数を設定してください。

- **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を格納する完全なパスを指定します。
- **CTX_FAS_KDC_HOSTNAME**: PBISとQuest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。複数のDelivery Controllerがある場合は、ドメインのすべてのKDCのホスト名を **/etc/krb5.conf** ファイルのpkinit_kdc_hostnameに追加します。詳細については、Knowledge Centerの記事 CTX322129を参照してください。
- **CTX_FAS_SERVER_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、ADグループポリシーを通じて構成されます。ドメインGPOでのFASポリシー設定については、グループポリシーの構成を参照してください。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\>'** 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。FASサーバーと適切に通信するには、FASサーバーで指定されたポート番号と一致するポート番号を追加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。

既存のLinux VDAインストールをアップグレードするには、以下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ができます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
Service" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force



systemctl restart ctxjproxy



systemctl restart ctxvda
```

`ctxreg`を介してFASサーバーを更新するには、以下のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
Service" -v "Addresses" -d "<Your-FAS-Server-List>"



systemctl restart ctxjproxy



systemctl restart ctxvda
```

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

スクリプトはkrb5-pkinitおよびpam_krb5パッケージをインストールし、関連する構成ファイルを設定します。RHEL 8以降では、PAM_KRB5がEPELリポジトリに移動されたため、スクリプトはこれらのディストリビューションでEPELを有効にしようとします。

FASを無効にする

Linux VDAでFASを無効にするには、次のコマンドを使用してConfDBからすべてのFASサーバーを削除します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "" --force



systemctl restart ctxjproxy



systemctl restart ctxvda
```

FAS SSOの失敗時にセカンダリ認証を有効にする

Linux VDAは、FASシングルサインオン（SSO）が失敗した場合に特にセカンダリ認証方法を提供することで、ログインの回復力を強化します。この機能により、FAS SSOで問題が発生した場合、ユーザーはパスワード認証のために手動で資格情報を入力するよう求められます。この機能を有効にするには、次のコマンドを実行します。

```
create -k
"HKLM\System\CurrentControlSet\Control\Citrix\AccessControl\Login\Global"
-t "REG_DWORD" -v "SecondaryAuthEnabled" -d "0x00000001" --force
```

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、(/ja-jp/federated-authentication-service/2607-ltsr/deployment-architectures.html)記事の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、以下を確認してください。

- Linux VDAが正しくインストールされ、構成されていること。
- パスワード認証を使用して、共通ストアでFAS以外のセッションを正常に起動できること。

FAS以外のセッションが正常に機能する場合は、**Login**クラスのHDXログレベルをVERBOSEに、VDAログレベルをTRACEに設定します。Linux VDAのトレースログを有効にする方法については、Knowledge Centerの記事CTX220130を参照してください。

Linux VDA環境に存在する可能性のある一般的な構成の問題を確認するには、Linuxの**XDPing**ツールを使用することもできます。

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

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

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

既存の設定が正しくない場合は、前の「FAS サーバーを設定する」の手順に従って再度設定してください。

CA 証明書の設定が正しくありません

FAS ストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。



/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: entry



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller: current
process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]
```

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_public_certificate: entry



2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
for response...



2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query to
server success



2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
get_public_certificate: exit



2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
input size 1888



2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
output size 1415



2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: get logon certificate success



2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
cache certificate success



2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: exit, 0
```

```
2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策

`/etc/krb5.conf` で、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/

    ......

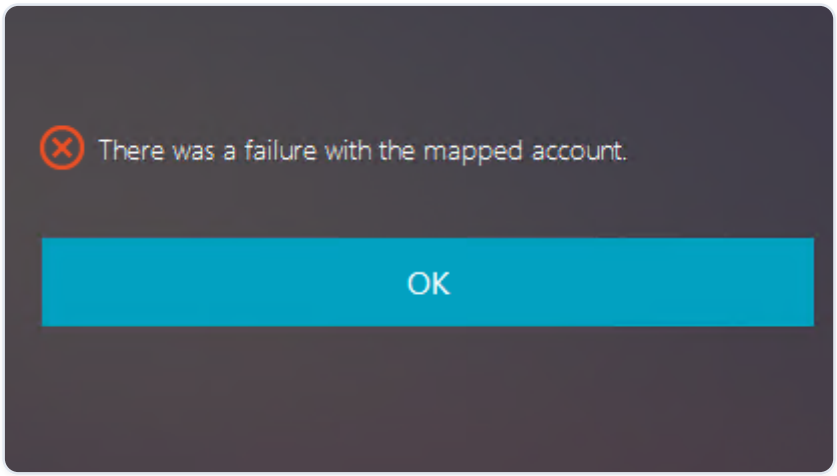
}
```

既存の設定が正しくない場合は、前の「証明書をインストールする」の手順に従って再度設定してください。

または、ルート CA 証明書が有効であるかどうかを確認してください。

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。



このエラーは次のことを示しています。

- ADFS ユーザーは正常に検証されましたが、AD にシャドウユーザーが構成されていません。

解決策

AD でシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に、次のエラーが発生します。

ADFSが構成されていません(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/adfs-not-configured.png)

この問題は、FASストアをSAML認証を使用するように構成しているにもかかわらず、ADFS展開が不足している場合に発生します。

解決策

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

関連情報

- 一般的なFAS展開については、「フェデレーション認証サービス アーキテクチャの概要」の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービス 高度な構成」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、英語以外の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。

英語以外の文字を含むセッションの起動に失敗しました(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/launching-nonenglish-session-fails.png)

回避策

CAツールで **Manage Templates** を右クリックし、**Citrix_SmartcardLogon** テンプレートを **Build from this Active Directory information** から **Supply in the request** に変更します。

リクエストオプションで提供(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/supply-in-the-request.png)

FIDO2 (プレビュー)

Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするために、FIDO2認証を設定できます。

> **注:**
>
> Linux VDAは、FIDO2とGoogle Chromeの組み合わせのみをサポートします。

> FIDO2認証を設定するには、次の手順を実行します。

1. Citrix FIDO2拡張機能パッケージをダウンロードします。
    a. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
    b. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
    c. **Components**をクリックしてLinux VDAを見つけます。
    d. Linux VDAをクリックしてダウンロードページを開きます。
    e. ソースパッケージをダウンロードします。
    f. ソースパッケージを解凍して、**FIDO2-JavaScript-Extensions.zip**を見つけます。
    g. FIDO2拡張機能パッケージを解凍します。FIDO2拡張機能ディレクトリは、**extensions > chrome > fido2**にあります。
2. Google ChromeにCitrix FIDO2拡張機能を追加します。
    a. Linux VDAでホストされているGoogle Chromeを開きます。
    b. アドレスバーの右側にある3点メニューをクリックし、**その他のツール > 拡張機能**を選択します。

「拡張機能」メニューの場所(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/more-tools-extensions.png)

c. **開発者モード**をオンにします。

開発者モード(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/developer-mode.png)

d. **Load unpacked**をクリックし、**extensions > chrome > fido2**にある拡張機能ディレクトリを選択します。

3. FIDO2認証を使用したいWebサイトで、FIDO2認証を使用するためにFIDO2セキュリティキーを登録します。

a. Citrix Workspace™アプリがインストールされているクライアントにFIDO2セキュリティキーを挿入します。

b. 多要素認証を有効にし、FIDO2を認証方法として追加します。

FIDO2認証が設定されると、Webサイトに正常にアクセスするためにセキュリティキーに触れるよう求められます。

非SSO認証

この記事では、Linux VDAで非SSO認証を有効にする方法について説明します。

概要

デフォルトでは、Linux VDAで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。次の表に、非SSOシナリオでサポートされるユーザー認証方法の組み合わせを示します。

CITRIX WORKSPACEアプリ	VDAセッション
ユーザー名	ユーザー名
スマートカード	ユーザー名
ユーザー名	スマートカード
FAS	ユーザー名
FAS	スマートカード

SSOを無効にする

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
```

ログイン画面をカスタマイズする

Linux VDAでは、SSOを使用しない場合のユーザーのログイン方法を制御できます。表示される認証方法を選択することで、ログインエクスペリエンスをカスタマイズできます。以前は、非SSOログイン画面には常にパスワード認証とスマートカード認証の両方がドロップダウンリストに表示されていました。現在、Linux VDAを次のように構成できます。

- パスワード認証のみ
- スマートカード認証のみ
- パスワード認証とスマートカード認証の組み合わせ（いずれかのオプションがデフォルトで表示される）

VDAで以下のレジストリ設定を調整することで、ログイン方法をカスタマイズできます。

`System\CurrentControlSet\Control\Citrix\login\NSSOLogonType`

`NSSOLogonType` レジストリキーの値は、非SSOユーザー向けのLinux VDAログイン画面に表示されるログイン方法を制御します。

- **1**: スマートカード認証のみ。ユーザーはスマートカードを使用してログインする必要があります。
- **2**: パスワード認証とスマートカード認証の組み合わせ（パスワード認証がデフォルトで表示される）
- **3**: パスワード認証とスマートカード認証の組み合わせ（スマートカード認証がデフォルトで表示される）。
- **その他の値**: パスワード認証のみ。ユーザーはユーザー名とパスワードでログインします。

Linux VDAでカスタマイズ可能なログイン機能が正しく構成されていることを確認するには、次の手順に従います。

1. 以下のコマンドを使用して、VDAで非SSOを有効にします。

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

2. VDAで以下のコマンドと同様のコマンドを使用して、デフォルトのログインタイプを変更します。

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

   前述のとおり、`NSSOLogonType` レジストリキーの値を `1` に設定すると、Linux VDAはログイン画面にスマートカード認証オプションのみを表示するように構成されます。

3. ログイン画面を確認し、ドロップダウンリストに構成したスマートカード認証のみが表示されていることを確認します。

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へのログオン時に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。ユースケースには以下が含まれます。

- 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

Linux VDAは、この機能について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。

制限された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、制限された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	クエスト
デビアン 12.15	はい	はい	はい	はい
デビアン 13.6	はい	はい	はい	いいえ
RHEL 10.2/9.8/9.6	はい	はい	はい	はい
RHEL 8.10	はい	はい	はい	はい
ロッキー Linux 10.2/9.8/9.6	はい	はい	はい	はい
ロッキー Linux 8.10	はい	はい	はい	はい
ウブントゥ 26.04	はい	はい	いいえ	いいえ
ウブントゥ 24.04	はい	はい	はい	はい
ウブントゥ 22.04	はい	はい	はい	はい

前提条件

スマートカードパススルー認証の利用可能性は、以下の条件に依存します。

- お使いのLinux VDAがDelivery Controller™に登録でき、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くことができること。
- OpenSCでサポートされているスマートカードが使用されていること。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。

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スマートカードを構成の説明の例として使用します。YubiKey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をサポートしています。

組織でより高度なスマートカードが必要な場合は、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、次のコマンドを実行して、OpenSCがスマートカードをサポートしていることを確認します。

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. ルート証明書をPEM形式で取得します。通常、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、次のコマンドに似たコマンドを実行します。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ています。

   ```
   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情報をデフォルトの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ます。
>
> **ヒント:**
>
> ドメイン参加にSSSDを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。 RHEL 8以降では、PAM_KRB5がEPELリポジトリに移動されたため、スクリプトはこれらのディストリビューションでEPELを有効にしようとします。

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

結果は次のようになります。

```
#*********************************************************************
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*********************************************************************
Step 1:Enable smart card logon.
  Do you want enable smart card logon? (y/n)[y] y
Step 2:Select the Active Directory integration method.
  Please select which Active Directory integration method to use:
     1: Winbind
     2: SSSD
     3: Centrify
  Select one of the above options (1-3)[1] 1
Step 3:Install dependent packages.
  Installing, please wait...
    [krb5-pkinit][pam_krb5] already installed.
    [pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][coolkey] already installed.
  Packages installed. [Success]
Step 4:Configure krb5.conf.
  Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
  /etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
  Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/libcoolkeypk11.so):/usr/lib64/pkcs11/libcoolkeypk11.so
  /etc/pam.d/ctxfsc configure successfully.
  /etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

注:

ドメイン参加方法としてQuestを使用する場合、キー配布センター（KDC）のホスト名を指定する必要があります。例：

```
[root@jx-rh79qu110 ~]# /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
#*********************************************************************
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*********************************************************************
Step 1:Enable smart card logon.
  Do you want enable smart card logon? (y/n)[y]
Step 2:Select the Active Directory integration method.
  Please select which Active Directory integration method to use:
     1: Winbind
     2: SSSD
     3: Centrify
     4: Quest
  Select one of the above options (1-4)[1] 4
Step 3:Install dependent packages.
  Installing, please wait...
    [krb5-pkinit][pam_krb5] already installed.
    [pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][opensc] already installed.
  Packages installed. [Success]
Step 4:Configure krb5.conf.
  Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
  Specify the KDC hostname:kdc.test.local
  /etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
  Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/opensc-pkcs11.so):/usr/lib64/pkcs11/opensc-pkcs11.so
  /etc/pam.d/ctxfsc configure successfully.
  /etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

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 sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*******************************************************************************
Step 1:Enable smart card logon.
  Do you want enable smart card logon? (y/n)[y] n
ctxsmartlogon.sh exit.
```

(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）、Internet Information Services、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、以下を確認してください。

- すべてのコンポーネントが正しく構成されていること。
- 秘密キー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VDAに正常にログオンできること。

PC/SC Liteパッケージをインストールする
PCSC Liteは、LinuxにおけるPersonal Computer/Smart Card (PC/SC) 仕様の実装です。スマートカードおよびリーダーと通信するためのWindowsスマートカードインターフェイスを提供します。Linux VDAでのスマートカードリダイレクトは、PC/SCレベルで実装されます。

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およびRocky Linux:

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

UbuntuおよびDebian:

```
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 および Rocky Linux:

```
yum install -y opensc
```

Ubuntu および Debian:

```
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
pam_krb5およびkrb5-pkinitモジュールをインストールするには、次のコマンドを実行します。

RHEL および Rocky Linux:

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

ウブントゥ:

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

デビアン:

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

pam_krb5モジュールは、プラガブル認証モジュールです。PAM対応アプリケーションは、pam_krb5を使用して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できます。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構成ファイルに、実際のレルムに従ってPKINIT情報を追加します。

> **注:**

pkinit_cert_matchオプションは、PKINIT認証を試行する前にクライアント証明書が一致する必要がある一致ルールを指定します。一致ルールの構文は次のとおりです。

コンポーネント規則 ...

ここで、**relation-operator**は、すべてのコンポーネントルールが一致する必要があることを意味する**&&、または1つのコンポーネントルールのみが一致する必要があることを意味する\|\|**のいずれかです。

一般的なkrb5.confファイルの例を次に示します。

```
EXAMPLE.COM = {


    kdc = KDC.EXAMPLE.COM


    auth_to_local = RULE:[1:$1@$0]


    pkinit_anchors = FILE:<path where you install the root certificate>/
certnew.pem


    pkinit_kdc_hostname = KDC.EXAMPLE.COM


    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature


    pkinit_eku_checking = kpServerAuth
```

```
}
```

PKINIT情報を追加すると、構成ファイルは次のようになります。

```
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

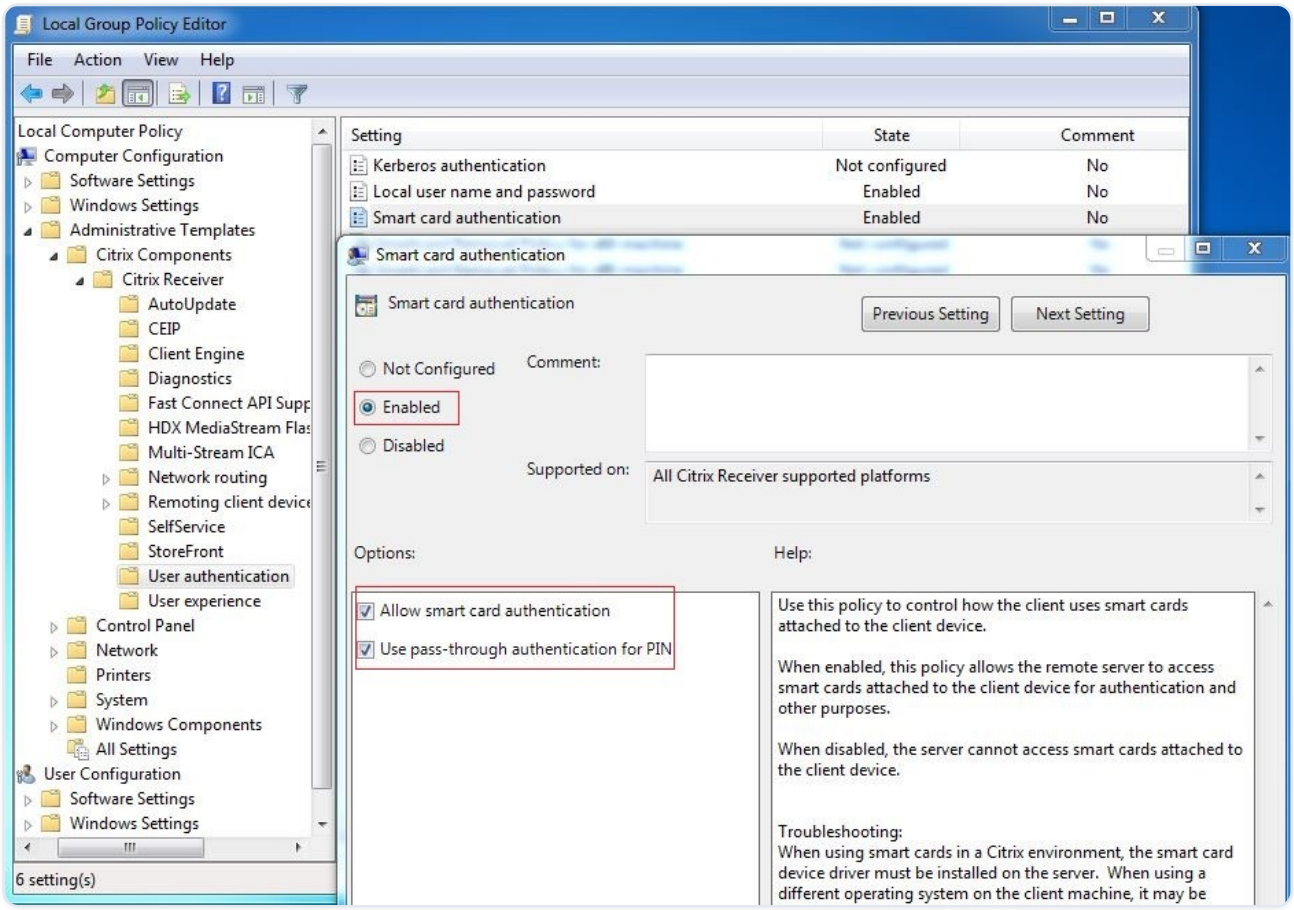
account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore]  pam_krb5.so
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、スマートカード認証を次のように有効にします。



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、関連するStoreFrontサイトのdefault.icaファイルに次のパラメーターを含めます。

```
[WFClient]

SmartCardCryptographicRedirection=On
```

クライアントで高速スマートカードログオンを無効にする

クライアントで高速スマートカードログオンを無効にするには、関連するStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

XDPingを実行する

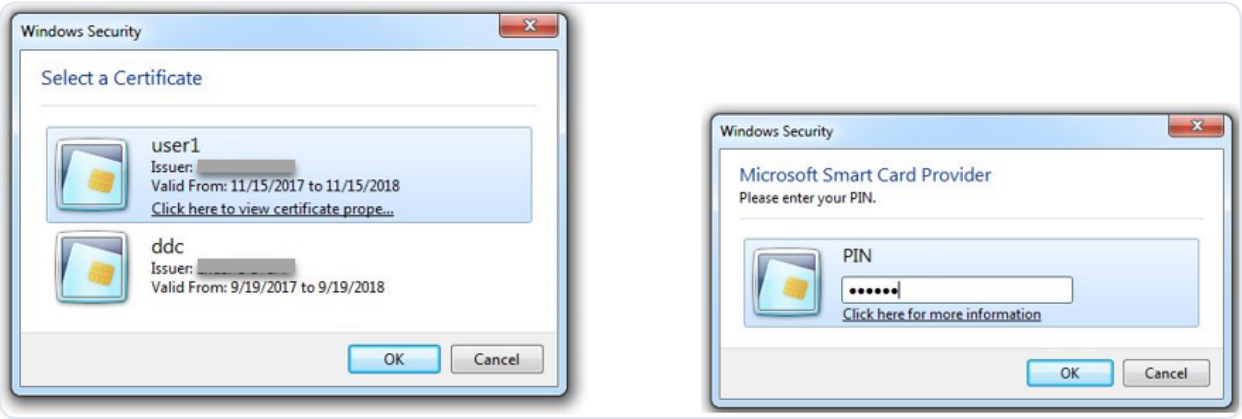
前述の手順を完了した後、Linux **XDPing**ツールを使用して、Linux VDA環境に存在する可能性のある一般的な構成の問題を確認できます。

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

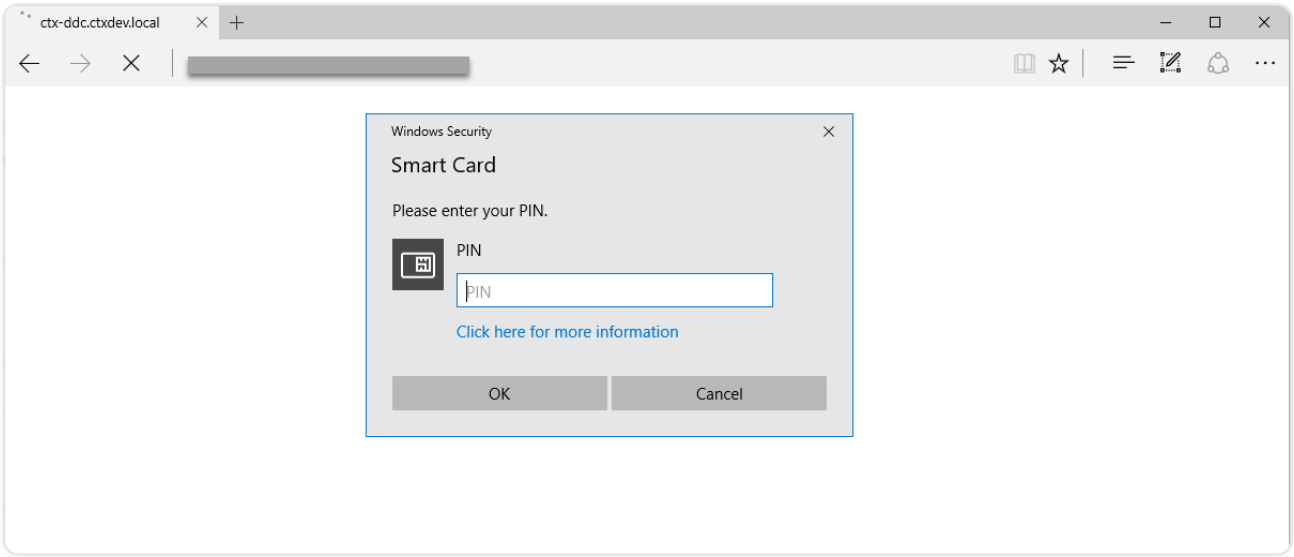
SSOと非SSOの両方のシナリオで、スマートカードを使用してLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用して、StoreFront™に自動的にログオンされます。StoreFrontでLinux仮想デスクトップセッションを起動すると、スマートカード認証のためにPINがLinux VDAに渡されます。
- 非SSOシナリオでは、証明書を選択し、PINを入力してStoreFrontにログオンするよう求められます。



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。



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、Linux VDA側で灰色のキャッシュウィンドウが表示され、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、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通知する他のプロンプトは表示されません。

ただし、StoreFront側では、セッションに再接続する際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/2607-ltsr/media/please-insert-smart-card.png)

制限事項

スマートカード取り外し設定

セッション中にサインインしているユーザーのスマートカードがスマートカードリーダーから取り外されたときの動作を指定するには、Linux VDAで次のレジストリキーを編集します。

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LocalPolicies\SecurityOptions

値の名前	タイプ	デフォルト	説明
SCardRemoveOption	REG_DWORD	0x00000000	この設定は、セッション中にサインインしているユーザーのスマートカードがスマートカードリーダーから取り外されたときに何が起こる

値の名前	タイプ	デフォルト	説明
			かを決定します。オプションは次のとおりです。
\^	\^	\^	0 **アクションなし**
\^	\^	\^	1 **強制ログオフ**: スマートカードが取り外されると、ユーザーは自動的にサインアウトされます。
\^	\^	\^	2 **セッション切断**: スマートカードが取り外されてもユーザーはサインアウトされず、セッションが切断されます。ユーザーは後でスマートカードを再挿入してセッションに再接続できます。
\^	\^	\^	3 **ワークステーションのロック**: 仮想デスクトップがロックされます。ユーザーはその後、スマートカードを再挿入し、PINコードを入力して画面のロックを解除できます。
SCardRemoveActionDelaySeconds	REG_DWORD	15	削除オプションが**強制ログオフ**または**セッション切断**に設定されている場合、アクションが実行されるまでに数秒の遅延をさらに指定できます。この設定により、スマートカード

値の名前	タイプ	デフォルト	説明
			が取り外された後、ユーザーセッションにメッセージボックスが表示され、指定された秒数後にセッションが強制ログオフまたは切断されることが示されます。その時間内にスマートカードを再挿入すると、セッションは中断されずに続行されます。

制限事項

Linux VDAは、一度に1つのスマートカードリーダーのみをサポートします。

その他のスマートカードとPKCS#11ライブラリのサポート

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ます。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、他のスマートカードやPKCS#11ライブラリを使用することもできます。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. `opensc-pkcs11.so`のすべてのインスタンスをお使いのPKCS#11ライブラリに置き換えます。
2. お使いの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`のようなお使いの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集合です。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**を介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、カスタムアイドルタイムアウトを5分に設定するには、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 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、利用可能なメモリ不足など、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設定する際は、以下を考慮してください。

- **認証されていないユーザーセッションへのログオンに失敗しました。**

レジストリが以下を含むように更新されていることを確認してください（0に設定）。

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsdサービスが実行中で、**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ます。**

認証されていないユーザーセッションでは、ロック画面ボタンとメニューはデフォルトで無効になっています。ただし、KDEでは引き続き表示されることがあります。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**に以下の行を追加します。例:

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、`/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`KDE Action Restrictions`パラメーターが不変として構成されている場合、ユーザー構成は効果がありません。

この問題を解決するには、システム全体の`kdeglobals`ファイルを変更して、`[KDE Action Restrictions]`セクションの`[$i]`タグを削除するか、または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
- ファイルの転送

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アンドペーストできます。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アンドペーストするには、以下を確認してください。

- ファイル数が20を超えないこと。
- ファイルサイズがデフォルトで200 MBを超えないこと。

- Linux VDAをインストールしたマシンでNautilusファイルマネージャーが利用可能であること。
- 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。

サポートされているLinuxディストリビューション

ファイルのコピーと貼り付け機能は、Linux VDAがサポートするすべてのLinuxディストリビューションで利用できます。

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:**
>
> ファイルのコピーと貼り付け機能を無効にするには、Citrix Studioで**クライアントクリップボードリダイレクト**ポリシーを**禁止**に設定します。 **クリップボードクライアントからセッションへの転送サイズを制限**および**クリップボードセッションからクライアントへの転送サイズを制限**ポリシーは、転送されるファイルのサイズではなく、クリップボードバッファーのサイズを制御します。サポートされている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。

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。
- ファイルのコピーと貼り付け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および貼り付けされた後でのみ、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/2607-ltsr/media/upload-icon-on-toolbar.png)

> **注:**
>
> Linux VDAとクライアントデバイス間でファイルをアップロードおよびダウンロードするには、Citrix Workspaceアプリのツールバーを有効にします。 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、クライアントデバイスに自動的に転送されます。

> **注:**

> 自動ダウンロードを使用するには、**File transfer for Citrix Workspace app for ChromeOS/HTML5**および**Download file for Citrix Workspace app for ChromeOS/HTML5**ポリシーを**Allowed**に設定する必要があります。

> 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を VDA上の**Save to My Device**ディレクトリに保存し、クライアントデバイスに自動転送できます。自動ダウンロードを実現するには、セッション内Webブラウザ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、ターゲットファイルを選択し、**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。

Citrix Workspace app for ChromeOS/HTML5のファイル転送: 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。

Citrix Workspace app for ChromeOS/HTML5のファイルアップロード: 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。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が可能です。

Citrix Workspace app for ChromeOS/HTML5のファイルダウンロード: 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。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が可能です。

使用方法

Citrix Workspace app 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。

   常にReceiver for HTML5を使用するオプションを選択する(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/always-use-receiver-for-html5.png)
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。

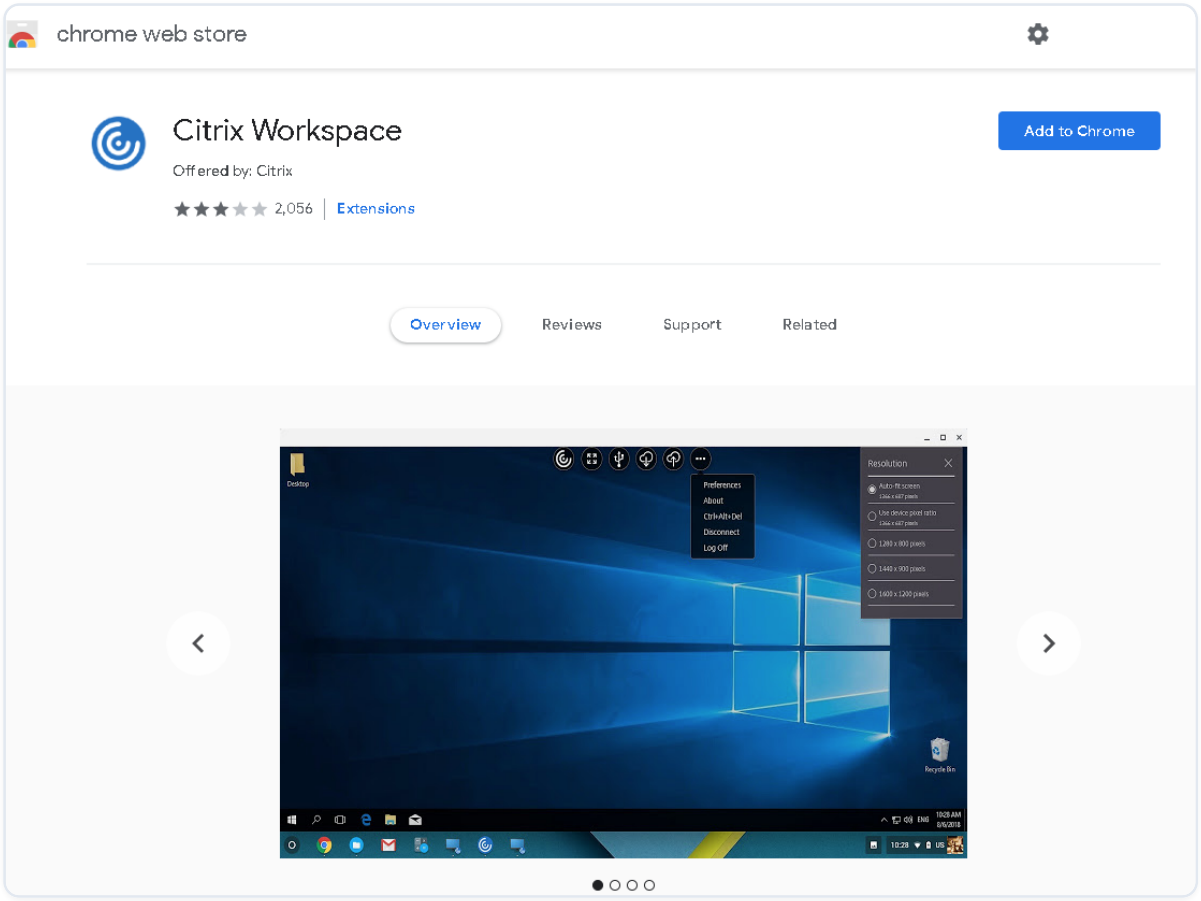
Citrix Workspace app for Chrome を介してファイル転送機能を使用するには:

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 Web StoreからCitrix Workspace appを入手します。

   Citrix Workspace app 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 appが利用可能なChrome Web StoreへのURLをクリックします。



c. Citrix Workspace app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 app for Chromeをクリックします。
4. 接続するStoreFront™ストアのURLを入力します。

   以前にURLを入力した場合は、この手順をスキップします。
5. 仮想デスクトップまたはアプリセッションを起動します。Linux VDAとクライアントデバイス間で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。

- 自動DPIスケーリング
- クライアントのバッテリー状態表示
- グラフィックの構成と微調整
- HDX画面共有
- 非仮想化GPU

- セッションウォーターマーク
- Thinwireプログレッシブディスプレイ

自動DPIスケーリング

Linux VDAは自動DPIスケーリングをサポートしています。ユーザーが仮想デスクトップまたはアプリケーションセッションを開くと、セッション内のDPI値はクライアント側のDPI設定に合わせて自動的に変更されます。

この機能に関する考慮事項は次のとおりです。

- この機能を使用するには、Citrix WorkspaceでDPIマッチングを有効にする必要があります。Citrix Workspaceアプリfor Windowsの場合、**いいえ、ネイティブ解像度を使用します**オプションが選択されていることを確認してください。Citrix Workspaceアプリfor WindowsのDPIスケーリングの構成について詳しくは、「DPIスケーリング」を参照してください。
- マルチモニター環境でこの機能を使用するには、各モニターが同じDPI設定で構成されている必要があります。混在DPIシナリオはサポートされていません。モニターが異なるDPI設定で構成されている場合、Linux VDAはすべての画面に最小のDPI値を適用します。
- この機能は、MATE、GNOME、GNOME Classic、およびKDEで有効になっています。KDEまたはMATEを使用する場合は、次の点を考慮してください。
  - KDEデスクトップ環境で実行されているLinux仮想デスクトップの場合：
    - KDE Plasma 5以降を使用することをお勧めします。
    - セッションの実行中にクライアント側でDPI設定を変更した場合、ユーザーはログオフして再度ログオンする必要があります。
  - MATEデスクトップ環境で実行されているLinux仮想デスクトップの場合：
    - スケールファクター1および2のみがサポートされています。
    - セッションの実行中にクライアント側でDPI設定を変更した場合、ユーザーはログオフして再度ログオンする必要があります。
- 仮想セッションのDPI値は、クライアント側のDPI設定に従って自動的に変更されます。現在、この機能は、100%や200%などの整数型のスケールファクターのみをサポートしています。クライアント側で構成されているスケールファクターが小数型の場合、仮想セッションのDPIは次の表に従って整数型のスケールファクターに変更されます。例：スケールファクターが125%の場合、DPI値は100%に変更されます。

クライアント側のスケールファクター	リモートセッションDPI	ディスプレイのスケーリング
174%以下	96 (1 x 96)	100%
175%～274%	192 (2 x 96)	200%
275%～399%	288 (3 x 96)	300%
400%以上	384 (4 x 96)	400%

注:

Linux VDAは、MATEデスクトップで最大200%のスケーリングをサポートします。

クライアントのバッテリーステータス表示

Linux VDAは、仮想デスクトップでクライアントデバイスのバッテリーステータスをリダイレクトして表示できます。この機能はデフォルトで有効になっており、Citrix Workspace™アプリの以下のバージョンで利用できます。

- iOS向けCitrix Workspaceアプリ
- Linux向けCitrix Workspaceアプリ
- Mac用Citrix Workspaceアプリ（バージョン2204.1はサポートされていません）
- Windows用Citrix Workspaceアプリ（バージョン2204.1はサポートされていません）

概要

ユーザーが仮想デスクトップを開くと、Linuxシステムトレイにバッテリーアイコンが表示されます。バッテリーアイコンは、クライアントデバイスのバッテリーステータスを示します。バッテリー残量の割合を確認するには、バッテリーアイコンをクリックします。たとえば、次のスクリーンショットを参照してください。

バッテリーステータスアイコン(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-status-icon.png)

異なるバッテリーアイコンは、異なるバッテリーステータスを示します。概要については、次の表を参照してください。

バッテリーアイコン	充電ステータス	バッテリー残量レベル	バッテリー残量パーセンテージ
充電中で残量レベルが高いバッテリー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-charging-high.png)	充電中（「+」記号で表示）	高（緑色で表示）	>=80%
充電中で、残量が中程度のバッテリー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-charging-medium.png)	充電中、「+」記号で示されます	中、琥珀色で示されます	20%以上かつ80%未満
充電中で、残量が少ないバッテリー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-charging-low.png)	充電中、「+」記号で示されます	低、赤色で示されます	< 20%
充電中でなく、残量が多いバッテリー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-not-charging-high.png)	充電中ではない、「-」記号で示されます	高、緑色で示されます	>=80%
充電中でなく、残量が中程度のバッテリー(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/battery-not-charging-medium.png)	充電中ではない、「-」記号で示されます	中、琥珀色で示されます	>=20% かつ <80%

バッテリーアイコン	充電ステータス	バッテリー残量レベル	バッテリー残量パーセンテージ
	充電中ではない（「-」記号で示される）	低い（赤色で示される）	< 20%
	不明	不明	不明

構成

クライアントのバッテリーステータス表示は、デフォルトで有効になっています。

この機能を無効にするには、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"
```

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\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000001"
```

注:

前述のコマンドは、クライアントのバッテリーステータス表示とモバイルレシーバー仮想チャネル (MRVC) を共有するソフトキーボード機能に影響します。

お使いのディストリビューションに基づいて、この機能を有効にするための追加手順を完了してください。詳細については、「システムトレイを有効にする」セクションを参照してください。

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ついて説明します。

詳細については、「システム要件」および「インストール概要」セクションを参照してください。

構成

3Dグラフィックワークロード向けに最適化

この設定は、グラフィックを多用するワークロードに最適なデフォルト値を構成します。グラフィックを多用するアプリケーションに重点を置くユーザーに対して、この設定を有効にします。このポリシーは、セッションでGPUが利用可能な場合にのみ適用してください。このポリシーによって設定されたデフォルト設定を明示的に上書きする他の設定は、優先されます。

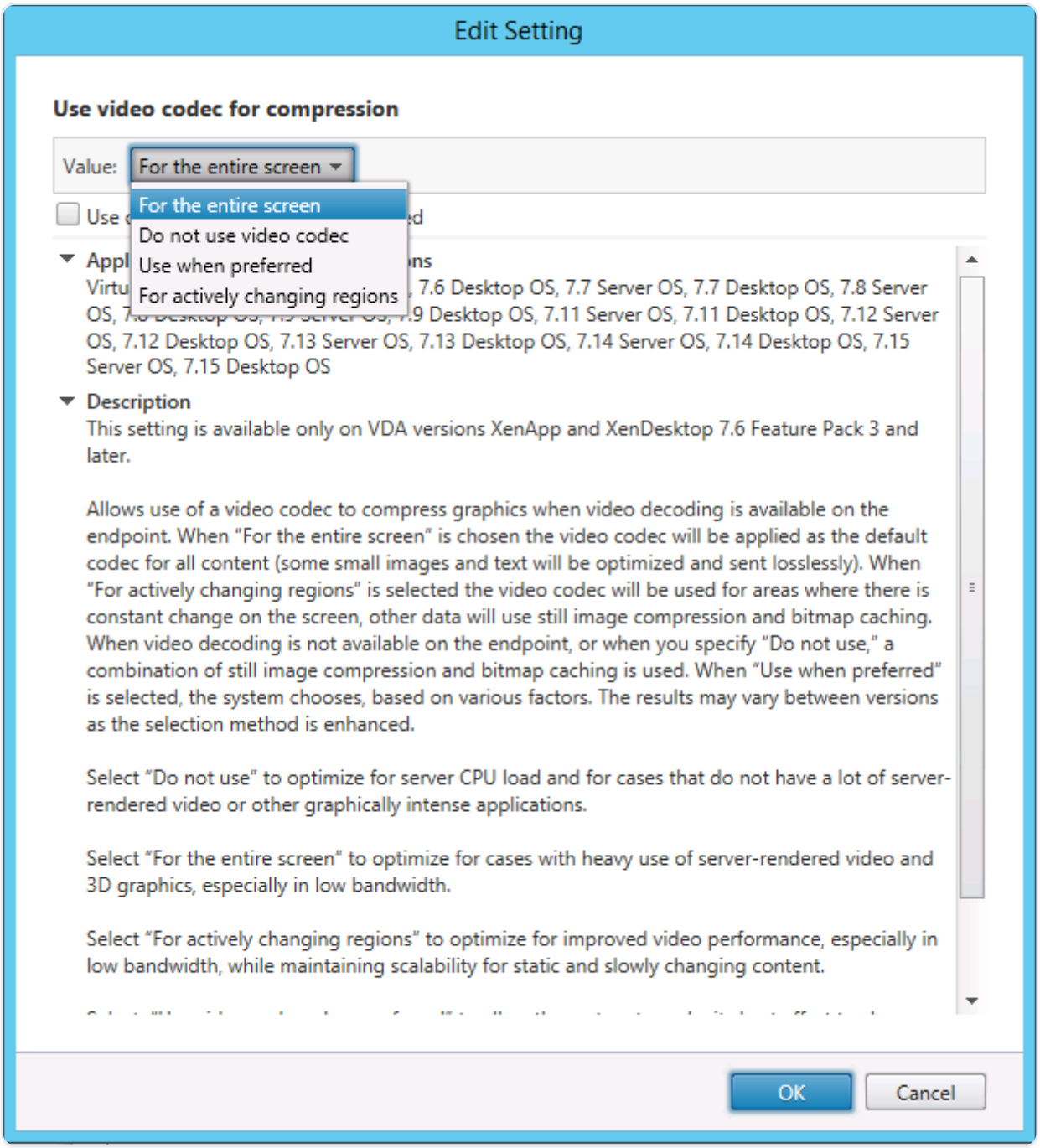
デフォルトでは、**3Dグラフィックワークロード向けに最適化**は無効になっています。

圧縮用のビデオコーデック

Thinwireは、Linux VDAで使用されるディスプレイリモートテクノロジーです。このテクノロジーにより、あるマシンで生成されたグラフィックを、通常はネットワーク経由で別のマシンに送信して表示できます。

圧縮にビデオコーデックを使用 グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対応する次のオプションを提供します。

- **優先する場合にのみ使用**。この設定はデフォルトです。追加の構成は不要です。これにより、すべてのCitrix®接続でThinwireが選択され、一般的なデスクトップワークロードのスケーラビリティ、帯域幅、および優れた画質のために最適化されます。
- **画面全体の場合**。ThinwireをフルスクリーンH.264またはH.265で提供し、特に3Dグラフィックを多用する場合に、ユーザーエクスペリエンスと帯域幅の向上を最適化します。「**画面全体の場合**」が選択されている場合、または「**優先する場合にのみ使用**」が選択され、「3Dグラフィックワークロード向けに最適化」が有効になっている場合に、セッションウォーターマークがサポートされます。
- **アクティブに変化する領域の場合**。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、モーション内の3D）を識別します。画像が動いている画面の部分でのみ、AV1、H.265、またはH.264を使用します。AV1、H.265、またはH.264ビデオコーデックを選択的に使用することで、HDX Thinwireは頻繁に更新される画面の部分を検出し、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、およびその他の場所でのロスレステキストまたは高品質画像の恩恵を受けます。**視覚品質**ポリシーが**常にロスレス**または**ロスレスに構築**に設定されている場合、AV1およびH.265の選択的な使用はサポートされません。



ディスプレイリモートのパフォーマンスを微調整するために、以下の視覚表示ポリシー設定を含む他の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
- 画質

AV1/H.265/H.264 ハードウェアエンコーディング

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は、利用可能な場合、GPU ハードウェアアクセラレーションを使用してビデオコーデックで画面要素を圧縮することを可能

にします。GPUハードウェアアクセラレーションは、ハードウェアリソースの利用を最適化し、フレーム/秒 (FPS) のパフォーマンスを大幅に向上させます。

GPUハードウェアアクセラレーションは、**圧縮にビデオコーデックを使用** ポリシーによって設定されたすべてのグラフィックモードをカバーします。

- **可能な場合に使用**
- **画面全体に**
- **アクティブに変化する領域に**

ハードウェアビデオ圧縮を有効にするには、次の手順を実行します。

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**に設定します。
2. **圧縮にビデオコーデックを使用**を**可能な場合に使用**、**画面全体に**、または**アクティブに変化する領域に**設定します。**ビデオコーデックを使用しない**に設定されていないことを確認してください。

使用可能にするには、AV1またはH.265ビデオコーデックがVDAとCitrix Workspaceアプリの両方でサポートされ、有効になっている必要があります。コーデックネゴシエーション中、AV1はH.265およびH.264よりも優先されます。AV1がサポートされていない場合、H.265がネゴシエートされます。AV1とH.265の両方がサポートされていない場合、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。GPUハードウェアが利用できない場合、VDAはソフトウェアビデオコーデックを使用したCPUベースのエンコーディングにフォールバックします。

AV1ハードウェアエンコーディングの要件

VDA

- VDA: 2311以降
- GPU: **NVIDIA Ada Lovelace以降** (NVIDIA GPUがサポートするビデオコーデックのマトリックスについては、`<https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new>`のNVIDIAドキュメントを参照してください。)
- NVIDIAグラフィックドライバー 522.25以降 (Video Codec SDK v12.0)

クライアント

- Windows版 Citrix Workspace™アプリ 2305以降
- AV1デコードをサポートするクライアントGPU:
  - NVIDIA Ampere以降
  - Intel 第11世代 / Arc以降

- AMD Radeon RX 6000 / Radeon Pro W6000シリーズ (RDNA2) 以降

H.265ハードウェアエンコードの要件

クライアント

- Windows 用 Citrix Receiver 4.10 から 4.12 まで
- Windows向けCitrix Workspaceアプリ1808以降

クライアントでH.265ハードウェアエンコードを有効にするには、「H.265ビデオエンコード」を参照してください。

H.265/H.264ロスレス圧縮

H.265/H.264ロスレス圧縮は、NVIDIA GPUによるHDX 3D PROハードウェアアクセラレーションで利用できます。H.265ロスレス圧縮には、Windows版 Citrix Workspaceアプリ 2305以降が必要です。H.264ロスレス圧縮には、以下のクライアントが必要です:

- Windows版 Citrix Workspaceアプリ 2303以降
- Apple M1チップを搭載したMac版 Citrix Workspaceアプリ 2301以降

H.265/H.264ロスレス圧縮を有効にするには、以下の手順を実行します:

1. **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**ポリシーを**有効**に設定します。
2. **圧縮にビデオコーデックを使用する**ポリシーを**画面全体**に設定します。
3. 「**Visual quality**」ポリシーを「**Always losses**」または「**Build to Lossless**」に設定します。

視覚的にロスレスな圧縮を許可する

「**Allow visually lossless compression**」ポリシーでは、グラフィックに対して真のロスレス圧縮の代わりに**視覚的に**ロスレスな圧縮を使用できます。視覚的にロスレスな圧縮は、真のロスレス圧縮よりもパフォーマンスが向上しますが、目には見えないわずかな損失があります。この設定は、「**Visual quality**」設定の値の使用方法を変更します。

「**Allow visually lossless compression**」ポリシーは、デフォルトで無効になっています。**視覚的に**ロスレスな圧縮を有効にするには、「**Allow visually lossless compression**」を「**Enabled**」に、「**Visual quality**」ポリシーを「**Build to Lossless**」に設定します。

「**Use video codec for compression**」ポリシーが「**Do not use video codec**」に設定されている場合、**視覚的に**ロスレスな圧縮は静止画エンコードに適用されます。「**Use video codec for compression**」ポリシーが「**Do not use video codec**」以外のグラフィックモードに設定されている場合、**視覚的に**ロスレスな圧縮はH.264エンコードに適用されます。

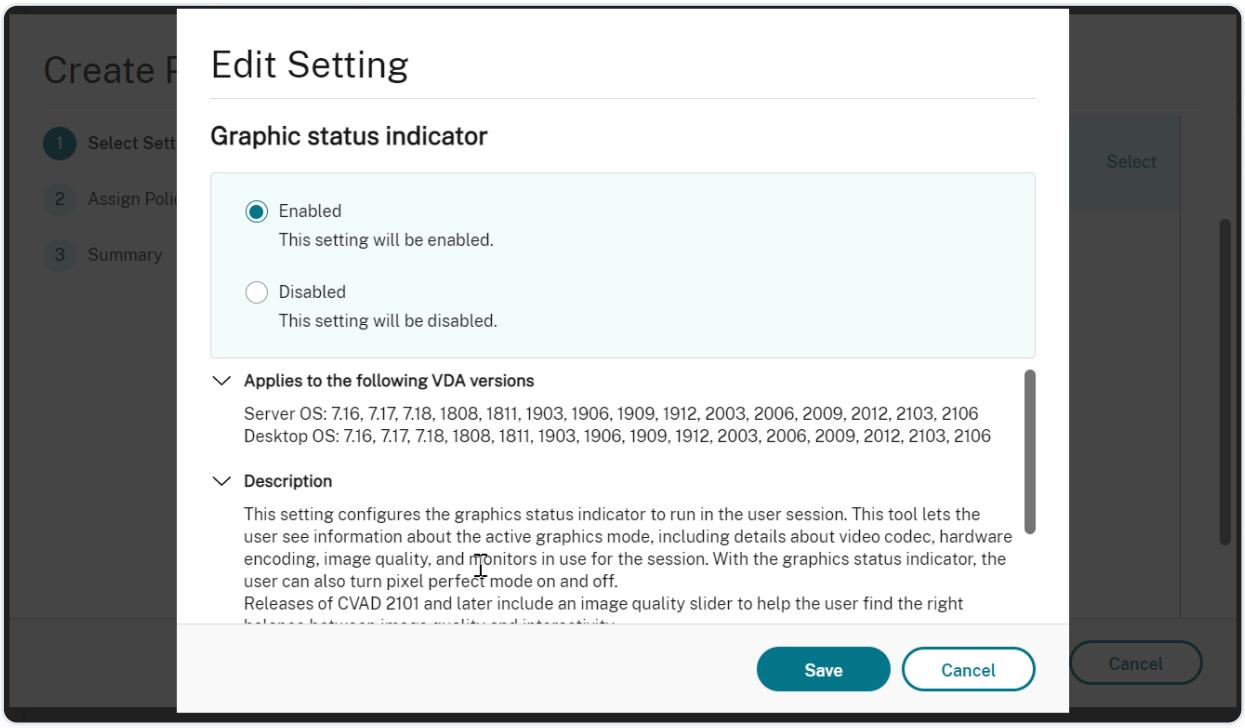
「**Visual quality**」および「**Use video codec for compression**」ポリシー設定の詳細については、「視覚表示ポリシー設定」および「グラフィックポリシー設定」を参照してください。

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、画質とインタラクティブ性の適切なバランスを見つけるのに役立ちます。

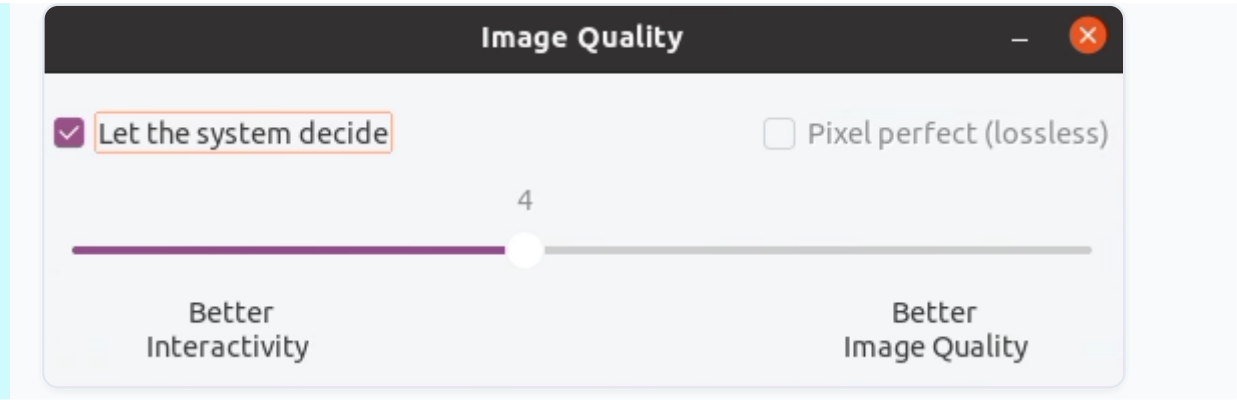
スライダーを使用するには、次の手順を実行します。

1. Citrix Studioで「**Graphic status indicator**」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。
> - スライダーUIは、ターミナルとシステムトレイの両方から起動できます。



次の選択肢が利用可能です。

- 画像品質を変更するには、スライダーを動かします。スライダーは0～9の範囲をサポートしています。
- システム定義の設定を使用するには、**システムに決定させる**を選択します。
- ロスレスモードに切り替えるには、**ピクセルパーフェクト**を選択します。

帯域幅の見積もりに基づいて平均ビットレートを調整する

Citrixは、帯域幅の見積もりに基づいて平均ビットレートを調整することにより、HDX™ 3D Pro ハードウェアエンコーディングを強化します。

HDX 3D Proハードウェアエンコーディングが使用されている場合、VDAはネットワークの帯域幅を断続的に推定し、それに応じてエンコードされたフレームのビットレートを調整できます。この新機能は、シャープネスと流暢さのバランスを取るメカニズムを提供します。

この機能はデフォルトで有効になっています。無効にするには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "DisableReconfigureEncoder" -d "0x00000001" --force
```

この機能を使用するだけでなく、シャープネスと流暢さの間で調整するために、次のコマンドを実行することもできます。**AverageBitRatePercent**と**MaxBitRatePercent**パラメーターは、帯域幅使用率のパーセンテージを設定します。設定する値が高いほど、グラフィックはシャープになり、流暢さは低下します。推奨される設定範囲は50～100です。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "AverageBitRatePercent" -d "90" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "MaxBitRatePercent" -d "100" --force
```

平均ビットレート調整では、画面が静止している場合、新しいフレームが送信されないため、最新のフレームは低品質の状態を維持します。シャープ化サポートは、最新のフレームを最高品質で再構成してすぐに送信することにより、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。

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消費量がわずかに高くな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
```

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`

AV1が使用中であることを確認する

> **注:**
>
> 現在のセッションでどのビデオコーデックが使用されているかを確認するには、以下のコマンドを実行するか、システムトレイからグラフィックステータスを確認します。

> AV1が使用中であることを確認するには、次のコマンドを実行します（**0**は未使用を意味します。**1**は使用中を意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep AV1
```

たとえば、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 "AV1" -d "0x00000000" --force
```

H.265が使用中であることを確認する

全画面H.265が使用中であることを確認するには、次のコマンドを実行します（**0**は未使用を意味します。**1**は使用中を意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

たとえば、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
```

H.264が使用中であることを確認する

H.264が使用中であることを確認するには、次のコマンドを実行します（**0**は未使用を意味します。**1**は使用中を意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

たとえば、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
```

どのYUVエンコーディングスキームが使用中であるかを確認する

どのYUVエンコーディングスキームが使用中であるかを確認するには、次のコマンドを実行します（**0**はYUV420を意味します。**1**はYUV422を意味します。**2**はYUV444を意味します）。

> **注:**
>
> **YUVFormat**の値は、ビデオコーデックが使用されている場合にのみ意味があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

たとえば、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ディングが使用されていることを確認する

YUV444 ソフトウェアエンコーディングが使用されていることを確認するには、次のコマンドを実行します。

```
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
```

HDX 3D Pro が有効になっていることを確認する

HDX 3D Pro が有効になっていることを確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep ProductEdition



sudo /opt/Citrix/VDA/bin/ctxreg dump | grep StackSessionMode



sudo /opt/Citrix/VDA/bin/ctxreg dump | grep 3DPro
```

HDX 3D Pro が有効になっている場合、結果は次のようになります。

`create -k "HKLM\Software\Citrix\VirtualDesktopAgent\State" -t "REG_SZ" -v "ProductEdition" -d "<PLT or ENT>" --force`

`create -k "HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t "REG_DWORD" -v "StackSessionMode" -d "0x00000000" --force`

`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "3DPro" -d "0x00000000" --force`

HDX 3D Pro に必要な NVIDIA ライブラリがロードされていることを確認するには、Linux VDA で **nvidia-smi** コマンドを実行します。結果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 |

+-----------------------------------------------------------------------------
----+
```

3D Pro でハードウェアエンコーディングが使用されていることを確認する

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ます。**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
```

3D Pro 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\1\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000001" --force`

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**という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、XenServer（旧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ます

回避策: VMのローカルVGAコンソールを無効にするには、次のコマンドを実行します。

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

ログオン時にGnome 3デスクトップのポップアップが遅い

これはGnome 3デスクトップセッションの起動に関する制限です。

Citrix Workspaceアプリのウィンドウ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表示されないことがあります

Citrix Workspaceアプリのウィンドウのサイズを変更すると、画面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し、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ライブラリ要素の**lightgl.js**は、**このテクスチャへのレ

ンダリングはサポートされていません（不完全なフレームバッファー）**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。HDX画面共有は、オンプレミス環境のCitrix Virtual Apps and DesktopsではCitrix Directorを、Citrix Cloud展開ではCitrix Monitorを使用してシャドウイングに利用できます。

注:

- 複数のLinux VDA間でセッション共有を行う場合、キーボード入力の問題を回避するために、同じVDAバージョンにアップグレードすることをお勧めします。
- 閲覧者のセッションと共有者のセッションで異なるキーボードレイアウトが存在する場合、共有者のキーボードレイアウトが有効になります。
- Unicodeモードを使用するとより多くの制限があるため、閲覧者のCitrix Workspace™アプリではスキャンコードモードを使用することをお勧めします。
- Citrix WorkspaceアプリクライアントのUnicodeモードでは、一部の文字が認識されない可能性があります。
- Unicodeモードを使用する場合、混在した出力になるのを避けるため、共有者と閲覧者が同じキーボードレイアウトを使用することをお勧めします。
- 閲覧者と共有者の両方でショートカットキーの組み合わせがアクティブな場合、それらは閲覧者でのみ有効になります。
- ショートカットキーにSuperキーが含まれている場合、それは閲覧者で有効になり、共有者にも送信されます。

次の例では、画面を共有し、他の人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：

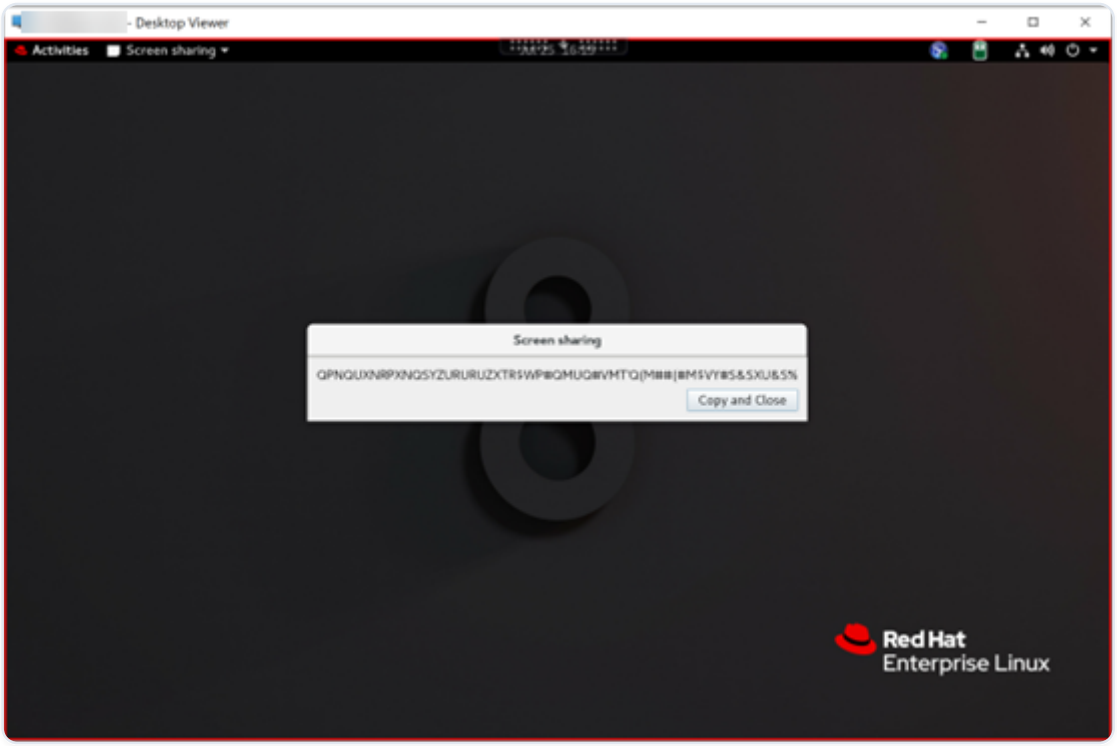
1. 仮想デスクトップの通知領域で、次のシステムトレイアイコンをクリックし、**画面共有 > 自分の画面を共有**を選択します。

   「自分の画面を共有」メニューを選択します(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/select-the-share-my-screen-menu.png)

「自分の画面を共有」メニューを選択します(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/select-the-share-my-screen-menu-2.png)

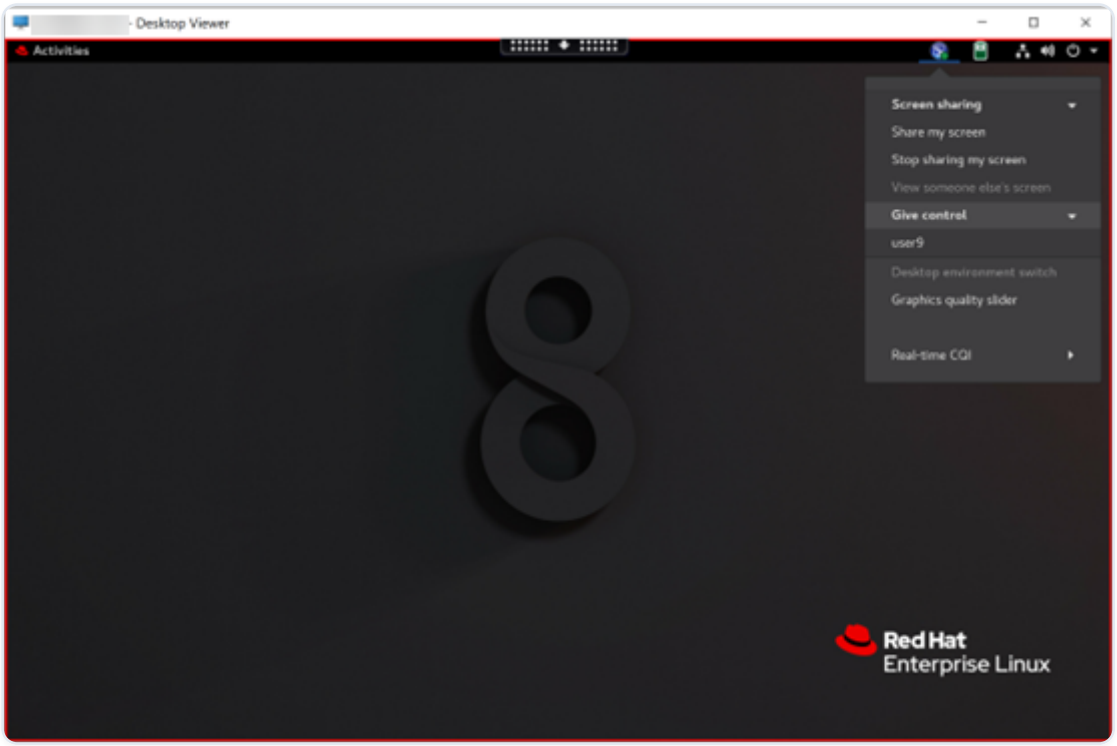
2. **コピーして閉じる**をクリックします。

   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は、画面の周囲に赤い枠が表示され、共有中であることを示します。

3. コピーしたコードを、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共有します。
4. 閲覧者に画面の制御を許可するには、 を選択し、閲覧者の名前を選択します。制御の許可を停止するには、閲覧者の名前の選択を解除します。



5. 画面の共有を停止するには、 を選択します。

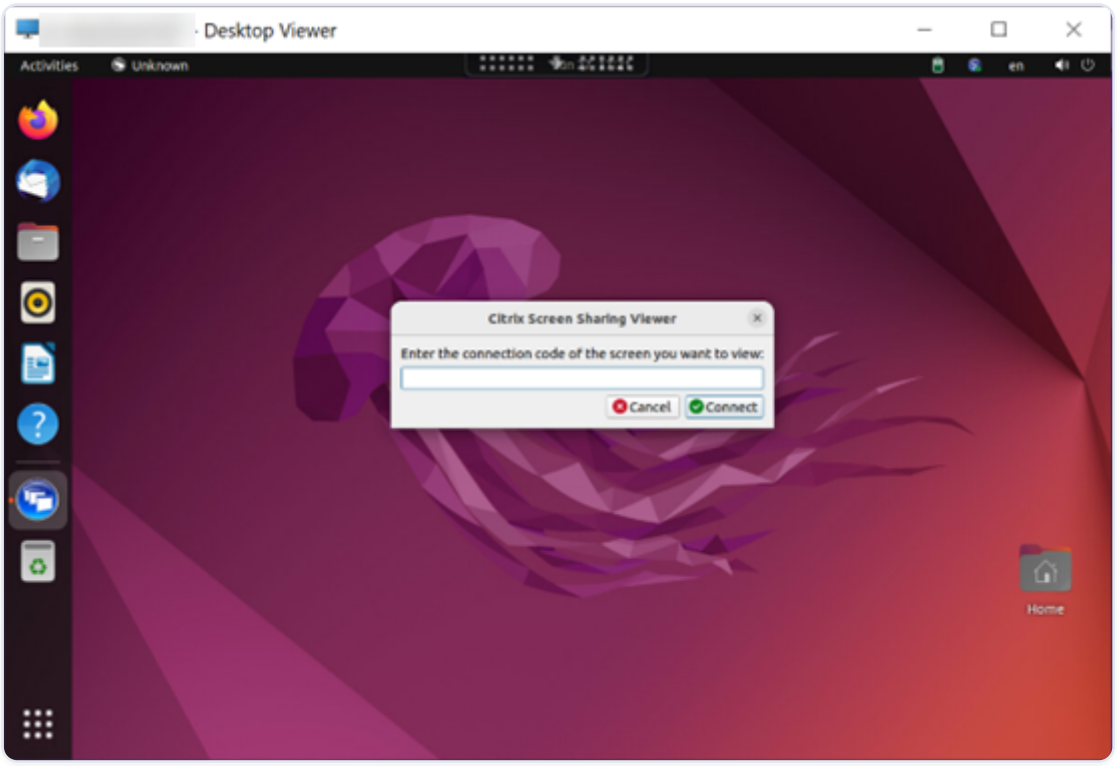


他のユーザーの画面を表示するには：

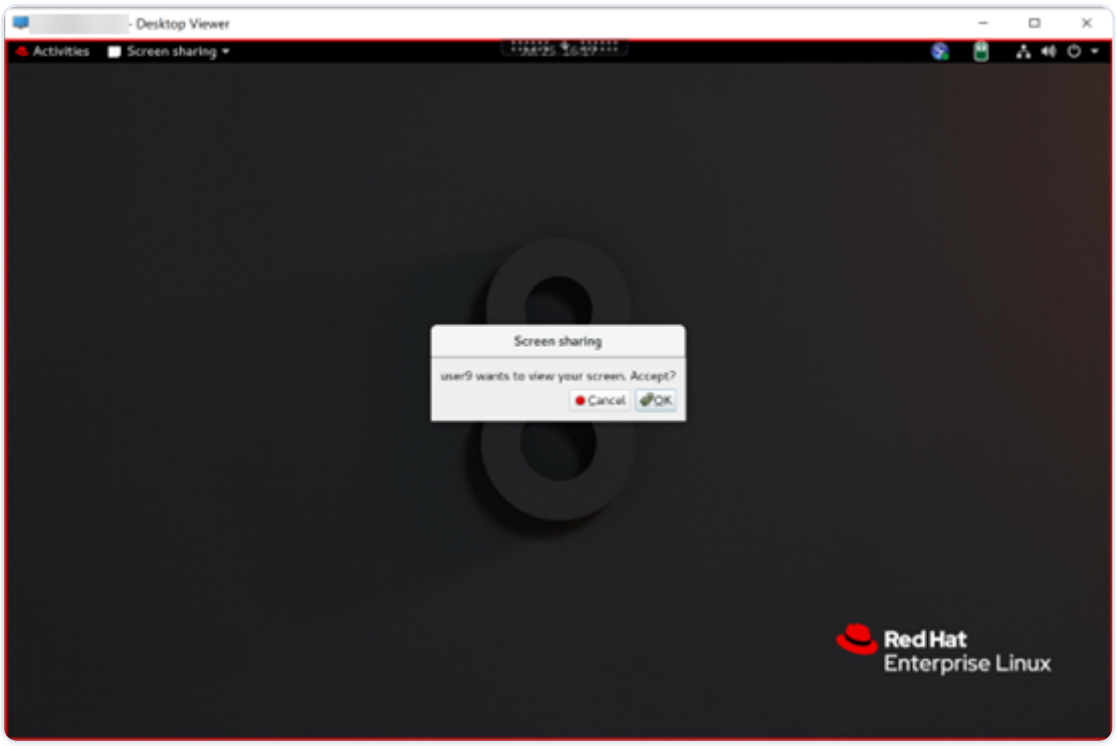
1. 仮想デスクトップの通知領域で、 アイコンをクリックし、 を選択します。



2. 表示したい画面の接続コードを入力し、 をクリックします。



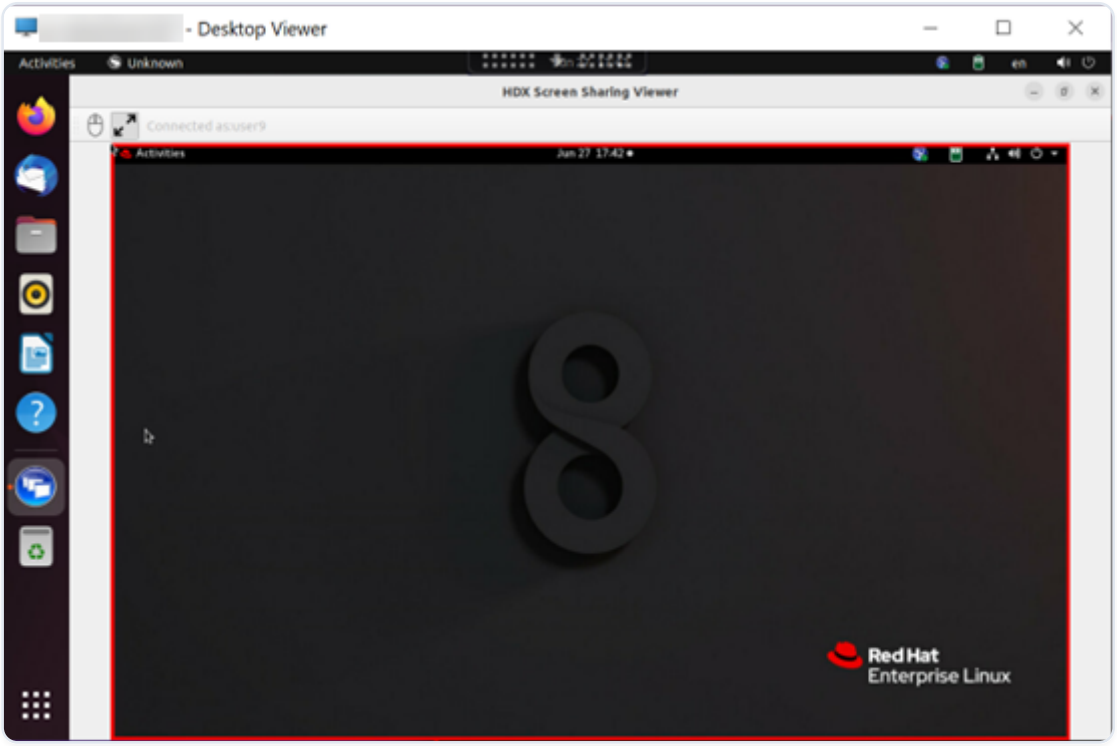
3. 画面共有者が要求を承認するまで待ちます。例：



> **ヒント:**
>
> - 共有者側では、Linuxシステムが要求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になり、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。

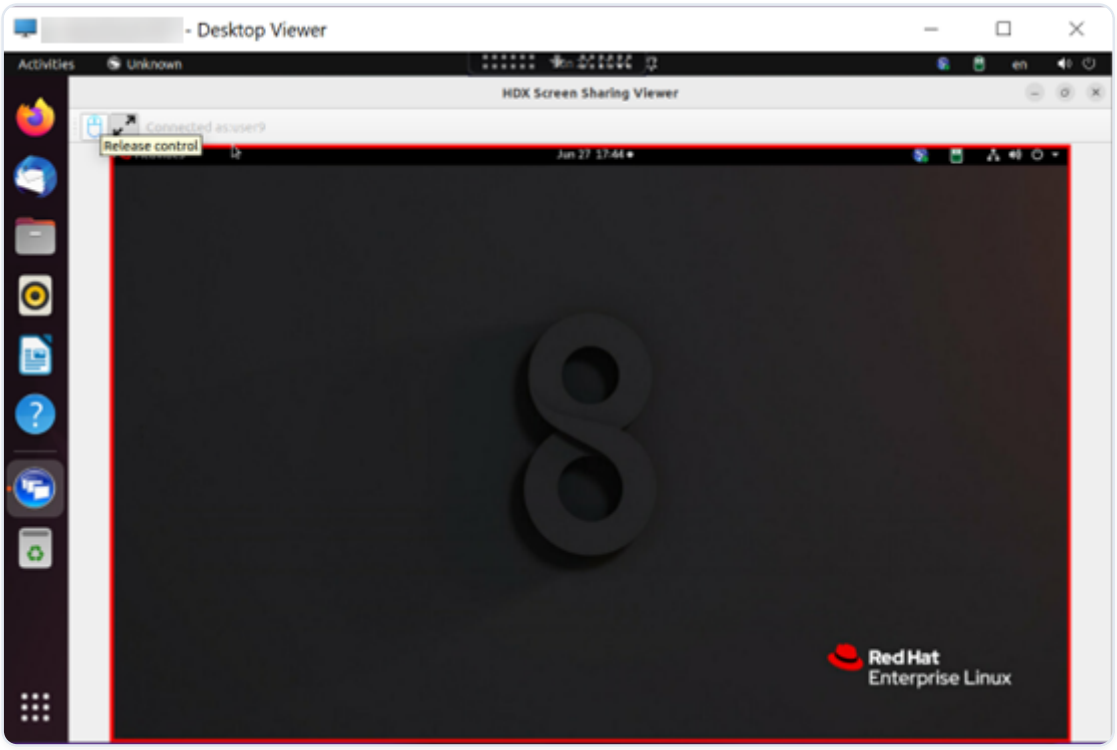


5. 共有画面の制御を要求するには、左上隅にあるマウスアイコンをクリックします。

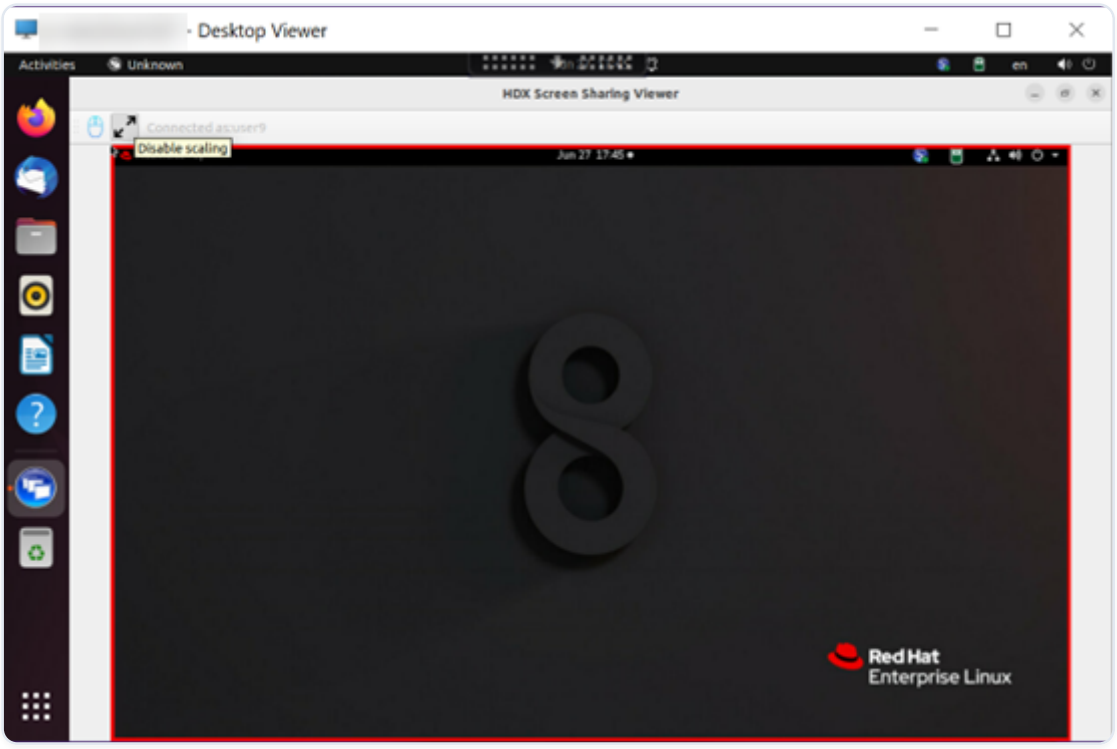
ヒント:

- 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. 表示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. システムトレイを有効にする。

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、次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。

画面共有の改善

Linux VDAでは、以下のレジストリキーを使用して、要件に応じて画面共有の改善機能を構成できます。

これらの設定を有効にするには、Linux VDAセッションを再起動する必要があります（ログオフしてログイン）。

1. **ScreenShareCodeType** - 使用する接続コードを構成します。デフォルトでは、短縮された接続コードは使用されません。

   短縮された接続コードを有効にする

```
#/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
-t "REG_DWORD" -v "ScreenShareCodeType" -d "0x01" --force
```

```
**Disable shortened connection code**
```

```
#/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
-t "REG_DWORD" -v "ScreenShareCodeType" -d "0x0" --force
```

2. **ScreenSharingUsers** - 参加を許可される視聴者の数を制御します。デフォルトでは、参加を許可されるユーザー数は10人です。

   例: 参加を許可される視聴者の数を32に設定します。

```
#/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
-t "REG_DWORD" -v "ScreenSharingViewerMaxNum" -d "0x20" --force
```

3. **ScreenShareJoinTimeoutInS** - ユーザーが参加しなかった場合、または接続されているすべてのユーザーが切断した場合に、画面共有が自動的に停止するまでのタイムアウトを制御します。現在、このレジストリキーは、ユーザーが参加した場合に接続コードがいつ期限切れになるかも制御します。コードが期限切れになると、再度使用することはできず、共有者は新しいコードを再生成する必要があります。デフォルト値は600（10分）です。

   例: 参加タイムアウトを300秒（5分）に設定します。

   ```
   #/opt/Citrix/VDA/bin/ctxreg create -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
   -t "REG_DWORD" -v "ScreenShareJoinTimeoutInS" -d "0x12C" --force
   ```

4. **ScreenShareRequestTimeoutInS** - 共有者が応答しない場合に、ユーザー接続または画面制御要求が自動的に拒否されるまでのタイムアウトを制御します。デフォルト値は30秒です。

   例: 要求タイムアウトを60秒に設定します。

   ```
   #/opt/Citrix/VDA/bin/ctxreg create -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
   -t "REG_DWORD" -v "ScreenShareRequestTimeoutInS" -d "0x3C" --force
   ```

5. ScreenSharePortStart - 使用するポート範囲の開始を制御します。デフォルトの開始ポートは52525（VDAインストーラーによって構成）で、デフォルトの範囲は100です。カスタムポート範囲が指定されている場合、画面共有データがそれらのポートで流れることを保証するために、適切なファイアウォールルールを設定する必要があります。

   例: 開始ポートを4096に変更します。

   ```
   #/opt/Citrix/VDA/bin/ctxreg create -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
   -t "REG_DWORD" -v "ScreenSharingPortBegin" -d "0x1000" --force
   ```

6. ScreenSharePortEnd - 使用するポート範囲の終了を制御します。デフォルトの終了ポートは52625（VDAインストーラーによって構成）です。

例: 終了ポートを4196に変更します。

```
#/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire"
-t "REG_DWORD" -v "ScreenSharingPortBegin" -d "0x1064" --force
```

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人のビューアーとセッション画面を共有できます。ビューアーの最大数は `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。

グラフィック用ロス許容モード

グラフィックの損失許容モードは、パケット損失が検出されたときにセッションがインタラクティブな状態を維持するように徹底的に再設計されました。ネットワークの状態が、事前に定義された帯域幅、遅延、パケット損失のしきい値を超えて悪化すると、Citrix®グラフィックエンコーダーは、パケット損失の影響を克服するために、より積極的なパケット配信モードに自動的に切り替わります。その結果、帯域幅の使用量はパケット損失量に比例して増加します。その後、状態が改善すると、Citrixグラフィックエンコーダーはシームレスに元の状態に戻ります。しきい値はポリシーを介して構成でき、デフォルトは300ミリ秒の遅延と5%のパケット損失です。

Windows版Citrix Workspace™アプリバージョン2311以降がサポートされています。その他のプラットフォームのサポートは、今後のCitrix Workspace™アプリリリースで追加される予定です。この機能の以前のバージョンと同様に、この機能が動作するにはHDX™アダプティブトランスポート (EDT) を有効にする必要があります。さらに、Citrix Gatewayサービス経由で接続する場合は、Gatewayでグラフィックのロス許容モードも有効にする必要があります。

マルチモニターのサポート

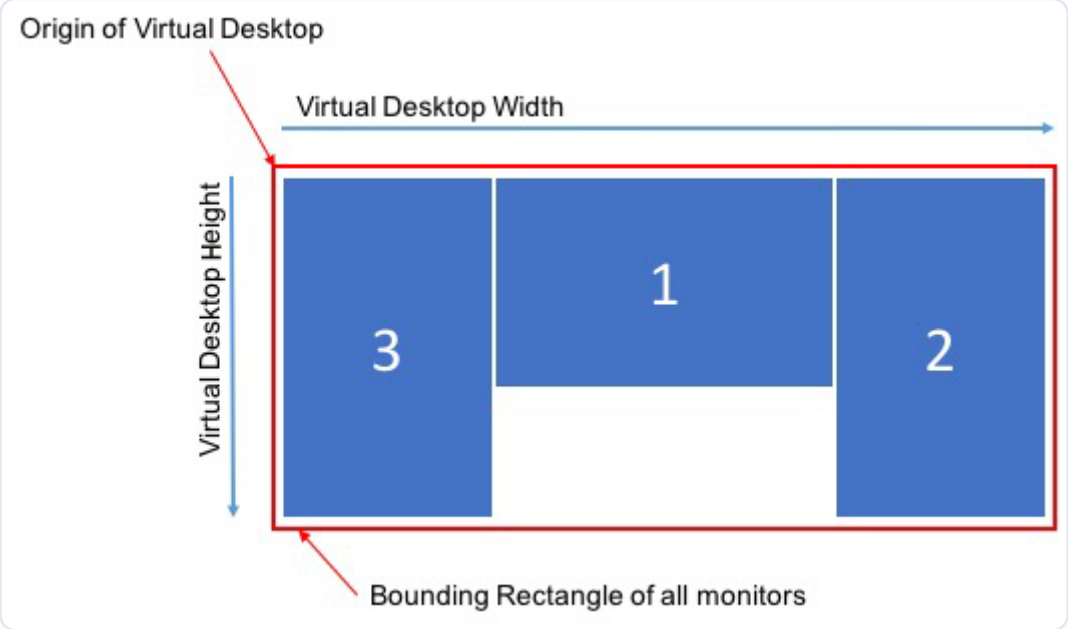
概要

Linux VDAは、モニターあたり2560×1600のデフォルト解像度で、すぐに使えるマルチモニターサポートを提供します。標準VDAは最大9台のモニターをサポートし、HDX™ 3D Pro VDAは最大4台のモニターをサポートします。

この記事では、異なるモニター解像度とレイアウトに対応するようにLinux VDAを構成する方法について説明します。

仮想セッションデスクトップ

Windows VDAと同様に、Linux VDAにはマルチモニター仮想デスクトップの概念があります。これは、すべてのモニターの境界矩形に基づいており、モニターの実際のレイアウトに基づくものではありません。したがって、仮想デスクトップの領域は、理論的にはクライアントのモニターがカバーする領域よりも大きくなる可能性があります。



仮想セッションデスクトップのサイズ

仮想セッションデスクトップの原点は、すべてのモニターの境界矩形の左上隅から計算されます。その点はX = 0、Y = 0に位置し、ここでXとYはそれぞれ水平軸と垂直軸です。

仮想セッションデスクトップの幅は、原点からすべてのモニターの境界矩形の右上隅までの水平距離（ピクセル単位）です。

同様に、仮想セッションデスクトップの高さは、原点からすべてのモニターの境界矩形の左下隅までの垂直距離（ピクセル単位）です。

この計算は、以下の理由から重要です。

- 異なるクライアントモニターレイアウトに対応するため
- Linux VDAでのメモリ使用量を理解するため

異なるクライアントモニター構成に対応する

さまざまなクライアントモニター構成における最大仮想デスクトップサイズを知ることで、クライアントモニター構成に関してLinux VDAを柔軟に構成できます。

以下のクライアントモニター構成を検討してください。

「クライアントモニター構成 1」(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/client-monitor-configuration.png)

上の図は、それぞれ2560×1600の解像度を持つ2台のモニターを備えた、既成のマルチモニター構成を示しています。

次に、以下のクライアントモニター構成で同じLinux VDAに接続することを検討します。

「クライアントモニター構成 2」(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/client-monitor-configuration-2.png)

上の図の各モニターの解像度が2560×1600の場合、既成のマルチモニター構成パラメーターでは不十分です。このモニターレイアウトの仮想セッションデスクトップに対応するには、最大高さが小さすぎます。この例のクライアントモニター構成に対応するには、Linux VDA仮想デスクトップのサイズを4160×2560に設定する必要があります。

マルチモニター構成で最大の柔軟性を得るには、サポートしたいすべてのモニターレイアウトの最小バウンディングボックスを見つけます。2台の2560×1600モニター構成の場合、可能なレイアウトは次のとおりです。

- **モニター1** 2560×1600 および **モニター2** 2560×1600
- **モニター1** 1600×2560 および **モニター2** 2560×1600
- **モニター1** 2560×1600 および **モニター2** 1600×2560
- **モニター1** 1600×2560 および **モニター2** 1600×2560

上記のすべてのレイアウトに対応するには、5120×2560の仮想セッションデスクトップが必要です。これは、目的のすべてのレイアウトを包含できる最小のバウンディングボックスです。

すべてのユーザーが一般的な横向きレイアウトで1台のモニターしか持っていない場合は、仮想デスクトップの最大サイズをモニターの最高解像度に設定します。

「モニター1台のみ」(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/only-one-monitor.png)

この例では、仮想デスクトップのサイズを2560×1600に設定する必要があります。デフォルト構成は5120×1600でモニター2台であるため、シングルモニター展開のメモリ使用量を最適化するには構成変更が必要です。

> **注:**
>
> マルチモニター設定でデスクトップが不適切な解像度で表示される場合は、Citrix WorkspaceアプリでDPI（Dots Per Inch）設定を調整してください。詳細については、Knowledge Centerの記事CTX230017を参照してください。

Linux VDAでのメモリ使用量の理解

仮想デスクトップのサイズを知ることで、各HDXセッションで使用されるメモリ量を計算できます。このメモリは、セッション開始時にグラフィックデータのために各セッションに割り当てられるメモリです。セッションの存続期間中、このメモリは変化しません。このメモリはセッションで使用されるメモリの総量ではありませんが、セッションごとのメモリ使用量を計算する最も簡単な方法です。

各HDXセッションに割り当てられるメモリ量を計算するには、次の式を使用します。

M = X × Y × Z,

ここで：

- **M** はセッショングラフィックに使用されるメモリ量です。
- **X** は仮想セッションデスクトップの幅です。
- **Y** は仮想セッションデスクトップの高さです。
- **Z** はHDXセッションウィンドウのカラー深度です。値はビットではなくバイト単位であるため、32ビットカラーの場合は4を使用します。

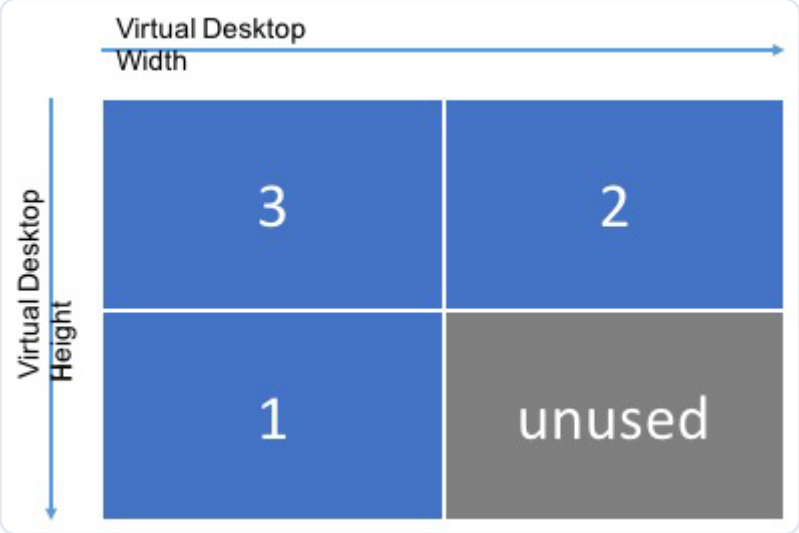
> **注：**
>
> Xサーバーのカラー深度はセッションの存続期間中（**ログインから切断/再接続を経てログオフまで**）変更できません。したがって、Linux VDAは常に仮想セッションデスクトップを32ビットとして割り当て、セッションで要求されたカラー深度にダウンサンプリングします。

> たとえば、1024×768セッションの場合、使用されるメモリは次のとおりです。

> 1024 × 768 × 4 / 2\^20 MB = 3 MB

> メモリ使用量を理解することは、各Linux VDAでのセッション密度を高める上で重要です。

次のクライアントモニター構成を検討してください。



各モニターの解像度が2560×1600の場合、このクライアントモニター構成に対応するには、仮想セッションデスクトップのサイズを5120×3200にする必要があります。灰色の領域は未使用であり、16,384,000バイト（つまり、2560 x 1600 x 4）の無駄なメモリに相当することに注意してください。

Citrix® マルチモニター構成パラメーター

以下の構成パラメーターを使用して、Linux VDA のマルチモニター機能を制御できます。

- `MaxScreenNum`

  パラメーター: HKEY_LOCAL_MACHINE/System/CurrentControlSet/Control/Citrix/Thinwire/MaxScreenNum

  説明: サポートするモニターの数

  種類: DWORD

  デフォルト: 4

  最大: 標準 VDA の場合は 9、HDX 3D Pro VDA の場合は 4

- `MaxFbWidth`

パラメーター: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFbWidth

説明: 仮想セッションデスクトップの最大幅

種類: DWORD

デフォルト: 5,120

最大: 16,384 (8,192 x 2)

- `MaxFbHeight`

  パラメーター: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFbHeight

  説明: 仮想セッションデスクトップの最大高さ

  種類: DWORD

  デフォルト: 1,600

  最大: 16,384 (8,192 x 2)

Linux VDAのマルチモニター構成を変更する

以下のセクションでは、Linux VDAでマルチモニター機能を有効化、構成、および無効化する方法について説明します。

次を使用して、モニターの最大数を設定します。

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxScreenNum" -d "NumMons" --force
```

ここで、`NumMons` は、標準VDAの場合は1〜9、HDX 3D Pro VDAの場合は1〜4の間の値です。

次を使用して、仮想セッションデスクトップの最大幅を設定します。

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxFbWidth" -d " MaxWidth" --force
```

ここで、`MaxWidth` は**`1,024`** と**`16,384`** の間の値です。

次を使用して、仮想セッションデスクトップの最大高さを設定します。

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxFbHeight" -d " MaxHeight" --force
```

ここで、`MaxHeight` は** `1,024` ** と** `16,384` ** の間の値です。

非仮想化GPU

Linux VDAのドキュメントでは、非仮想化GPUは以下を指します。

- リモートPCアクセスシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

この記事では、非仮想化GPUのサポートに関する情報を提供します。

NVIDIA Capture SDK for LinuxをサポートするNVIDIA GPUでHDX™ 3D Proを有効にする

NVIDIA Capture SDK for LinuxをサポートするNVIDIA GPUの場合、Linux VDAのインストール時に**CTX_XDL_HDX_3D_PRO**を**Y**に設定するだけで、HDX 3D Proを有効にできます。追加の設定は不要です。HDX 3D Proを有効にすると、ハードウェアアクセラレーションはデフォルトで有効になります。

NVIDIA Capture SDK for LinuxをサポートしないNVIDIA GPU、およびAMDやIntelなどの他のメーカーのGPUとの互換性

> **注:**
>
> このシナリオでは、ソフトウェアエンコーディングのみがサポートされます。

ステップ1：Linux VDAのインストール時にCTX_XDL_HDX_3D_PROをYに設定する

環境変数については、「ステップ8：easy installを実行して環境とVDAを構成し、インストールを完了する」を参照してください。

ステップ2：Xdamageをインストールする

たとえば、Ubuntu 20.04にXDamageをインストールするには、**sudo apt-get install -y libxdamage1**を実行します。通常、XDamageはXServerの拡張機能として存在します。

ステップ3：次のコマンドを実行してXDamageを有効にする

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

ステップ4：Xorg構成ファイルを変更する

/etc/X11 の下に、次の4つのテンプレート構成ファイルがあります。接続されているモニターの数に基づいて、名前に対応する番号が含まれるテンプレート構成ファイルのいずれかを変更します。たとえば、モニターが1台のみ接続されている場合は、名前に1が含まれるテンプレート構成ファイル、つまり ctx-driver_name-1.conf を変更します。モニターが2台接続されている場合は、名前に2が含まれるテンプレート構成ファイル、つまり ctx-driver_name-2.conf を変更します。

- 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、これは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ます。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、GPU 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。ネイティブ構成ファイルを使用して、GPU がローカル環境で動作することを確認してください。

   b. ネイティブ構成ファイルの「Device」セクションを **ctx-driver_name-1.conf** にコピーします。

3. 次のコマンドを実行してレジストリキーを設定し、Linux VDA が手順1で設定した構成ファイル名を認識できるようにします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_SZ" -v "DriverName" -d "intel" --force
```

Remote PC Access VDA のモニターブランキング

Linux VDA は、非仮想化GPUを使用するRemote PC Access VDA の物理モニターブランキングをサポートしています。

この機能をサポートする完全にテスト済みのLinuxディストリビューションには、Ubuntu 20.04とDebian 11が含まれます。

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2つの手順を実行します。

1. お使いのLinuxディストリビューションに基づいて、`evdi-dkms`パッケージをインストールします。

```
sudo apt install evdi-dkms
```

2. EVDIへのグラフィックディスプレイオフロードを有効にします。

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

3. Intel GPUを使用している場合は、ディスプレイマネージャーを無効にします。そうしないと、ディスプレイマネージャーがIntel GPUを占有し、Citrix®リモートセッションで利用できなくなります。

```
sudo systemctl disable --now gdm
```

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の結果です。デバッグのためにトレース情報を収集するには、/opt/Citrix/VDA/bin/setlogを使用してGFX_X11をverboseに設定します。

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:**
>
> - 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。Remote PC Access 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はデフォルトで無効になっています。Red Hatは、Xorgサーバーに影響を与えるCVE-2024-31083のセキュリティ脆弱性に対処しました。

ルートレスXorgの機能を維持するには、次の手順に従って、ICA®セッションでMIT-SHMが機能することを確認してください。

1. ラッパースクリプトではなく、Xorg実行可能ファイルに直接`cap_ipc_owner`機能を適用するには、次のディストリビューション固有のコマンドを使用します。
   - ウブントゥ、デビアン: `sudo setcap 'cap_ipc_owner=+ep' /usr/lib/xorg/Xorg`
   - RHEL、ロッキー: `sudo setcap 'cap_ipc_owner=+ep' /usr/libexec/Xorg`
   - SUSE: `sudo setcap 'cap_ipc_owner=+ep' /usr/libexec/Xorg /usr/bin/Xorg`
2. 次のコマンドを使用してルートレスXorgを有効にします。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "1" --force
```

ルートレスXorgがすでに有効になっている場合に無効にするには、次の手順を実行します。

1. 次のコマンドを実行して機能を無効にします。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "0" --force
```

2. Xorg実行可能ファイルから`cap_ipc_owner`機能を削除するには、ディストリビューション固有のコマンドを使用します。
   - ウブントゥ、デビアン: `sudo setcap 'cap_ipc_owner=-ep' /usr/lib/xorg/Xorg`
   - RHEL、ロッキー: `sudo setcap 'cap_ipc_owner=-ep' /usr/libexec/Xorg`
   - SUSE: `sudo setcap 'cap_ipc_owner=-ep' /usr/libexec/Xorg /usr/bin/Xorg`

セッションウォーターマーク

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ユーザーへの抑止力として、セッションデスクトップに表示されます。ウォーターマークは、テキストレイヤーまたはアルファチャネル付きのPNG画像として指定できます。ウォーターマークは、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:

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。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。代わりに、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、画面イメージが取得されたセッションのユーザーのログオン資格情報によって追跡される、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、以下の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。

セッションウォーターマークポリシー設定

セッションウォーターマークを有効にする

この設定を有効にすると、セッション表示にセッション固有の情報を表示する不透明なウォーターマークが表示されます。他のウォーターマーク設定は、この設定が有効になっているかどうかに依存します。

デフォルトでは、セッションウォーターマークは無効になっています。

クライアントIPアドレスを含める

この設定を有効にすると、セッションは現在のクライアントIPアドレスをウォーターマークとして表示します。

デフォルトでは、**クライアントIPアドレスを含める**は無効になっています。

接続時間を含める

この設定を有効にすると、セッションウォーターマークに接続時間が表示されます。形式はyyyy/mm/dd hh:mmです。表示される時刻は、システムクロックとタイムゾーンに基づいています。

デフォルトでは、**接続時間を含める**は無効になっています。

ログオンユーザー名を含める

この設定を有効にすると、セッションは現在のログオンユーザー名をウォーターマークとして表示します。表示形式はUSERNAME@DOMAINNAMEです。ユーザー名は最大20文字にすることをお勧めします。ユーザー名が20文字を超えると、フォントサイズが小さくなったり、切り捨てられたりする可能性があり、ウォーターマークの効果が低下します。

デフォルトでは、**ログオンユーザー名を含める**は有効になっています。

VDAホスト名を含める

この設定を有効にすると、セッションは現在のICA®セッションのVDAホスト名をウォーターマークとして表示します。

デフォルトでは、**VDAホスト名を含める**は有効になっています。

VDA IPアドレスを含める

この設定を有効にすると、セッションは現在のICAセッションのVDA IPアドレスをウォーターマークとして表示します。

デフォルトでは、**VDA IPアドレスを含める**は無効になっています。

セッションウォーターマークのスタイル

この設定は、単一のウォーターマークテキストラベルを表示するか、複数のラベルを表示するかを制御します。**値**ドロップダウンメニューから**複数**または**単一**を選択します。 追加のスタイルオプションについては、この記事の**ウォーターマークカスタムテキスト**セクションを参照してください。

「**複数**」は、セッションに5つのウォーターマークラベルを表示します。中央に1つ、四隅に4つです。

「**単一**」は、セッションの中央に単一のウォーターマークラベルを表示します。

デフォルトでは、**セッションウォーターマークスタイル**は**複数**です。

透かしの透明度

ウォーターマークの不透明度を0から100まで指定できます。指定する値が大きいほど、ウォーターマークは不透明になります。

デフォルトでは、値は17です。

ウォーターマークのカスタムテキスト

値はデフォルトで空です。空でない文字列を入力するか、文字列を形成する構文を設定するか、またはその組み合わせを使用してセッションウォーターマークに表示できます。空でない文字列は、1行あたり最大25個のUnicode文字をサポートします。長い文字列は25文字に切り捨てられます。

例えば、ポリシーを次の値に設定できます。

```
<date>
<time><newline><username><style=single><fontsize=40><font=Ubuntu><position=center><rotation
Linux VDA<newline>Version 2207
```

すべての構文オプションの説明については、次の表を参照してください。

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<style>`	ウォーターマークのレイアウトスタイル	`xstyle`、`single`、`tile`、`horizontal`	`xstyle`	-

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<position>`	ウォーターマークの位置	`center`, `topleft`, `topright`, `bottomleft`, `bottomright`	`center`	レイアウトスタイルが**single**に設定されている場合にのみ有効です。
`<rotation>`	ウォーターマークの特定の角度への回転	-180–180	0	-
`<transparency>`	ウォーターマークの不透明度	0–100	17	-
`<font>`	-	システムでサポートされているフォント	サン	-
`<fontsize>`	-	20–50	0 (自動計算)	-
`<fontzoom>`	`<fontsize>` と `<image>` で設定したフォントサイズと画像サイズの割合	0 –	100	-
`<image>`	PNGウォーターマーク	VDA上のPNG画像へのパス	該当なし	この構文はPNGウォーターマークを構成します。アルファチャネルを持つPNGのみがサポートされています。PNGウォーターマークが使用されている場合、`<style>`、`<position>`、`<rotation>`

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
				、`<transparency>`、および `<fontzoom>` の構文オプションのみが有効になります。
`<date>`	セッション接続日 (YYYY/MM/DD) のプレースホルダー	該当なし	該当なし	-
`<time>`	セッション接続時刻 (HH:MM) のプレースホルダー	該当なし	該当なし	-
`<domain>`	ユーザーアカウントドメインのプレースホルダー	該当なし	該当なし	-
`<username>`	現在のログオンユーザー名 (ユーザーアカウントドメインを除く) のプレースホルダー	該当なし	該当なし	-
`<hostname>`	VDAのホスト名のプレースホルダー	該当なし	該当なし	-
`<clientip>`	クライアントのIPアドレスのプレースホルダー	該当なし	該当なし	-

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<serverip>`	VDAのIPアドレスのプレースホルダー	該当なし	該当なし	-

注:

ウォーターマークカスタムテキストが少なくとも1つの有効な構文オプションで指定されている場合、**セッションウォーターマークを有効にする**を除く他のすべてのセッションウォーターマークポリシーは無視されます。

構文オプションが指定されていないか、サポートされていない値に設定されている場合、既定値が使用されます。

制限事項

- セッションウォーターマークは、以下のいずれかのケースでサポートされています。
  - **ビデオコーデックを圧縮に使用**が**画面全体**に設定されている場合。
  - **ビデオコーデックを圧縮に使用**が**優先する場合に使用**に設定されており、(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/graphics/configuring-graphics.html#)が有効になっている場合。
- ブラウザコンテンツリダイレクトが使用されているセッションでは、セッションウォーターマークはサポートされていません。セッションウォーターマーク機能を使用するには、ブラウザコンテンツリダイレクトが無効になっていることを確認してください。
- レガシーNVIDIAドライバーを使用し、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。画面キャプチャがコピーされないように対策を講じることをお勧めします。

マルチセッションLinux VDAでの共有GPUアクセラレーション

HDX™ 3D PROは、VDIデスクトップ（シングルセッションモード）用に構成されたLinux VDAのみをサポートします。マルチセッションLinux VDAの場合、共有GPUアクセラレーションを有効にしてOpenGL 3Dアプリケーションを高速化できます。

> **注:**
>
> Waylandディスプレイサーバーは、共有GPUアクセラレーションではサポートされていません。

構成

マルチセッションLinux VDAで共有GPUアクセラレーションを有効にしてOpenGL 3Dアプリケーションを高速化するには、次の構成手順を完了します。

手順1：VirtualGLをインストールする

`<https://sourceforge.net/projects/virtualgl/files>`から**VirtualGL**をダウンロードしてインストールします。DebianベースのLinuxディストリビューションには**.deb**パッケージを、RHELベースのLinuxディストリビューションには**.rpm**パッケージをダウンロードします。

手順2：VirtualGLを構成する

1. Linuxディスプレイマネージャー（例：LightDMまたはGNOME Display Manager（GDM））を停止します。
2. 次のコマンドを実行して、VirtualGL構成スクリプトを実行します。

   ```
   #/opt/VirtualGL/bin/vglserver_config
   ```

   スクリプトの実行中に、次の選択を行うことをお勧めします。

   - 「1」を選択して「VirtualGLで使用するサーバーを構成する（GLX + EGLバックエンド）」
   - 「n」を選択して「3D Xサーバーアクセスを**vglusers**グループに制限する」
   - 「n」を選択して「フレームバッファーデバイスアクセスを**vglusers**グループに制限する」
   - 「n」を選択して「XTEST拡張機能を無効にする」
3. 構成スクリプトを終了し、Linuxディスプレイマネージャーを再起動します。

ステップ 3: GPU アクセラレーションを使用して OpenGL 3D アプリケーションを実行する

Linux VDA セッションで GPU アクセラレーションを使用して OpenGL 3D アプリケーションを実行するには、2つの方法があります。

- **方法 1:** すべての OpenGL 3D アプリケーションで共有 GPU アクセラレーションを有効にする

  これを行うには、Linux VDA で bash ターミナルを開き、次のコマンドを実行してから、bash ターミナルを再起動します。この方法により、bash ターミナルから起動されたすべての OpenGL 3D アプリケーションで共有 GPU アクセラレーションが有効になります。

  ```
  #/opt/Citrix/VDA/sbin/ctxgpushare.sh enable
  ```

- **方法 2:** 特定の OpenGL 3D アプリケーションで共有 GPU アクセラレーションを有効にする:

  これを行うには、Linux VDA でターミナルを開き、アプリケーション名を指定して次のコマンドを実行します:

  ```
  #vglrun <AppName>
  ```

制限事項

- 共有 GPU アクセラレーションは、Linux VDA のディスプレイマネージャーと密接に連携します。期待される機能とパフォーマンスを実現するには、共有 GPU アクセラレーションのディスプレイマネージャーとして LightDM を使用することをお勧めします。
- WebGL ハードウェアアクセラレーションは、Ubuntu および Debian の Firefox でのみサポートされています。

スケーラビリティ

GPU を共有できる同時セッションの最大サポート数は、CPU とシステムメモリに依存します。また、GPU の最大ビデオメモリにも大きく依存します。

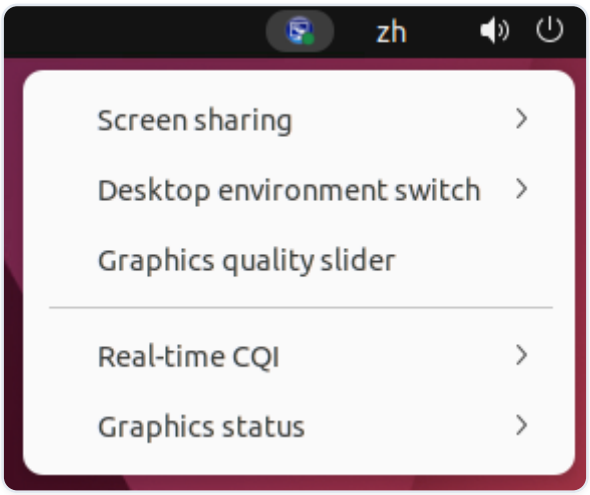
例:

もし	ならば
NVIDIA **M10-2B** vGPU が 2,048 MB のビデオメモリを持ち、VariCAD ビューアなどの OpenGL アプリケーションが各セッションで	理論上、同時セッションの最大サポート数は 20 を超えることはできません。

もし	ならば
そのワークロードに 100 MB のビデオメモリを使用する場合、,	

システムトレイ

セッションユーザーは、次のシステムトレイアイコンをクリックして、アクションを実行したり、インジケーターを表示したりできます。



システムトレイ内のアイテムの紹介

各アイテムは、トグル付きの機能に対応しています。アイテムに対応する機能が無効になっている場合、そのアイテムは非表示になり、表示されません。

- **画面共有**

  この機能の詳細については、HDX画面共有を参照してください。

- **デスクトップ環境の切り替え**

  このアイテムは、**ctxdesktopswitch.sh** のGUIです。詳細については、セッションユーザーによるカスタムデスクトップ環境を参照してください。

  セッションユーザーによるデスクトップ環境のカスタマイズ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。

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
  -v "EnableDesktopSwitch" -d "0x00000000" --force
  ```

- **グラフィック品質スライダー**

詳細については、グラフィック構成の記事のグラフィック品質スライダーセクションを参照してください。

- **リアルタイムCQI**

  現在、ICAラウンドトリップタイム（RTT）と遅延データが表示されます。詳細については、セッションメトリッククエリユーティリティを参照してください。

  システムトレイアイコンは、リアルタイムCQIの遅延に応じて異なって表示されます。

  アイコンの表示が変更されるタイミングを制御するしきい値があります。デフォルトでは、次のように設定されています。

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
  -v "HighLatencyThreshold" -d "0x000000dc" --force
  ```

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
  -v "LowLatencyThreshold" -d "0x00000078" --force
  ```

  実際の遅延が**LowLatencyThreshold**以下の場合、アイコンは緑色で表示されます。実際の遅延が**HighLatencyThreshold**より大きい場合、アイコンは赤色で表示されます。その他の状況では、アイコンは黄色で表示されます。リアルタイムCQIが無効になっている場合、アイコンには色マークがありません。

  リアルタイムCQIはデフォルトで有効になっており、表示されます。これを無効にして非表示にし、トレイアイコンに色マークを付けないようにするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
  -v "EnableCqiShow" -d "0x00000000" --force
  ```

- **グラフィックステータス**

  このインジケーターは、現在のセッションのグラフィック設定を表示します。デフォルトで有効になっています。これを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "EnableGfxInfo" -d "0x00000000" --force
```

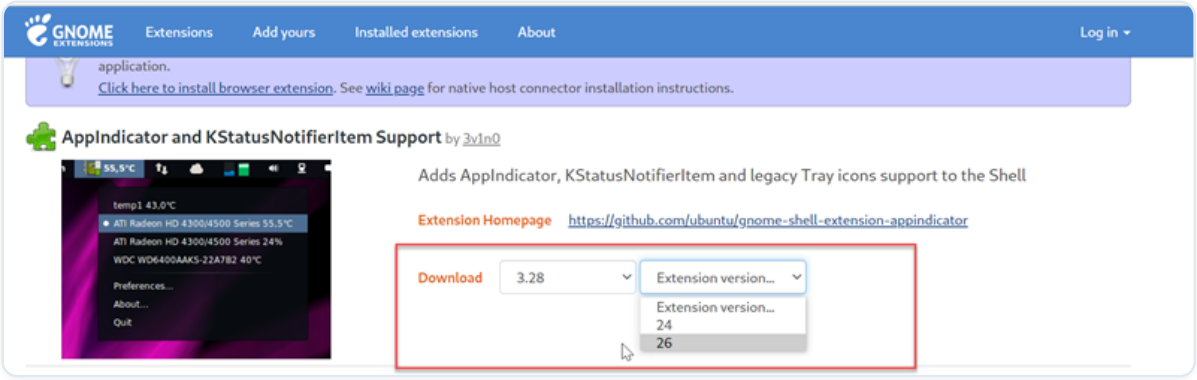
システムトレイを有効にする

システムトレイとクライアントバッテリーステータス表示はデフォルトで有効になっています。ただし、特定の条件下では、システムトレイとクライアントバッテリーステータス表示を有効にするために追加の設定を行う必要があります。詳細は次のとおりです。

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. (**この手順は、RHELおよびRocky Linux 10.x/9.x/8.10、またはGNOMEがインストールされたSUSE 15.7を使用している場合にのみ必要です。**) AppIndicatorサポートを有効にするには、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シェルのバージョンを確認するには、`gnome-shell --version`コマンドを実行します。

   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、拡張機能のバージョンとして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、解凍したディレクトリの名前を`appindicatorsupport@rgcjonas.gmail.com`に変更します。パッケージ内の**metadata.json**ファイルの**"uuid"**値が`appindicatorsupport@rgcjonas.gmail.com`に設定されていることを確認します。

   d. `appindicatorsupport@rgcjonas.gmail.com`ディレクトリを`/usr/share/gnome-shell/extensions/`の下の場所に移動するには、`mv`コマンドを実行します。

   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** ファイルはルートユーザーのみが読み取れます。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`、`Enter` キーを順に押すか、`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 または Extensions ツールで **KStatusNotifierItem/AppIndicator Support** を有効にします。

> **注:**
>
> GNOME がインストールされている RHEL および Rocky Linux 10.x/9.x、SUSE 15.7 の場合、拡張機能の管理は Tweaks から移動されました。GNOME Software アプリストアから GNOME Extensions をダウンロードし、GNOME Extensions で AppIndicator サポートを有効にすることをお勧めします。

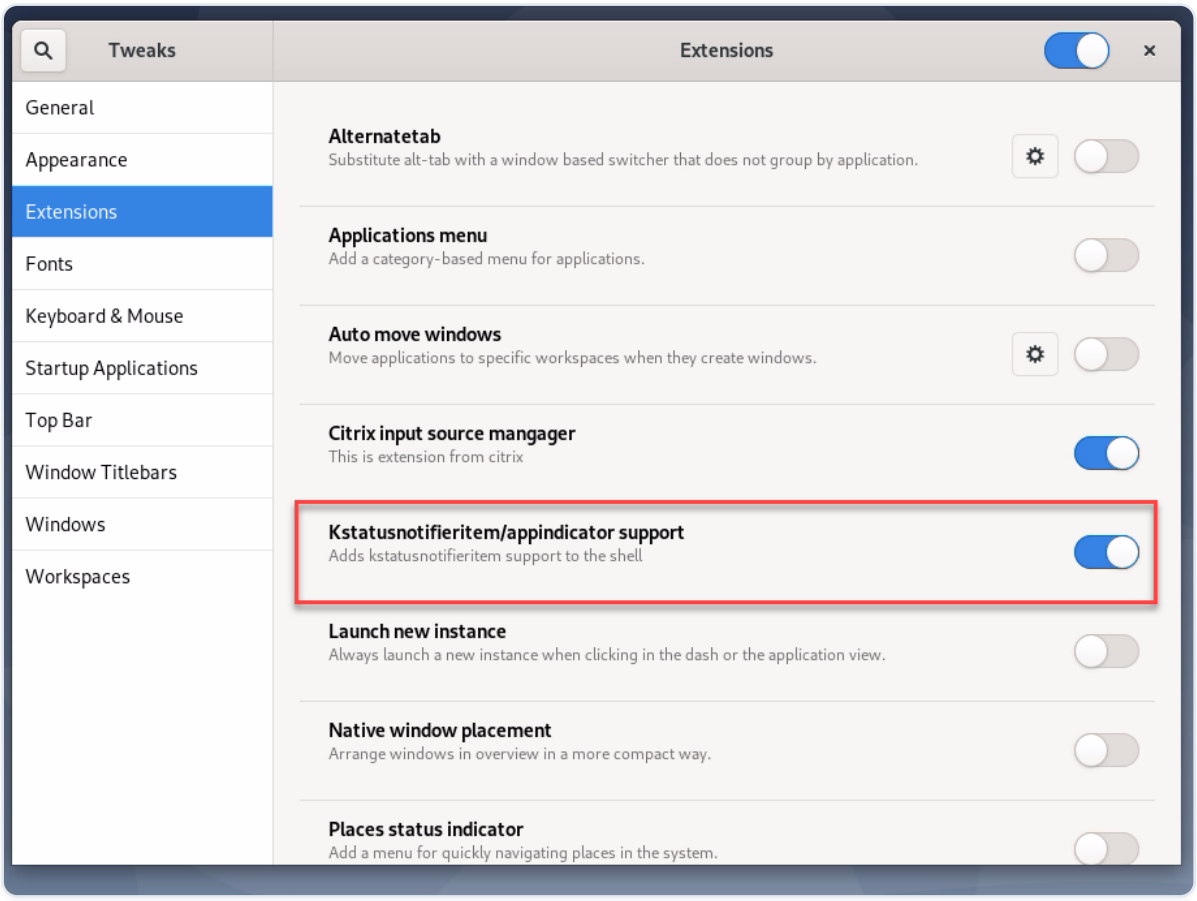
3. (**この手順は、GNOME がインストールされている RHEL および Rocky Linux 8.10 を使用している場合にのみ必要です。**) GNOME システムトレイアイコンをインストールして有効にするには、次の手順を実行します。

a. `sudo apt install gnome-shell-extension-appindicator` コマンドを実行します。GNOME が拡張機能を認識するには、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 **Extensions** を選択します。

d. **Kstatusnotifieritem/appindicator support** を有効にします。



4. GNOME システムトレイアイコンをインストールして有効にするには、次の手順を実行します。

   a. アクティビティ画面で Extentions を検索します。

   b. Extentions を構成して、**AppIndicator and KstatusNotifierItem Support** を有効にします。

   c. ログアウトしてセッションに再ログインします。

Thinwireプログレッシブ表示

セッションのインタラクティブ性は、低帯域幅または高遅延の接続で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は、グラフィックの更新に遅れることがあります。

バージョン7.17までは、ポリシー設定を使用して、セッションを低視覚品質に構成するか、より低い色深度（16ビットまたは8ビットグラフィック）を設定することで、帯域幅の消費を削減できました。ただし、ユーザーが弱い接続を使用していることを知っている必要がありました。HDX Thinwireは、ネットワークの状態に基づいて静止画の品質を動的に調整しませんでした。

バージョン7.18以降、HDX Thinwireは、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ミリ秒を超える。

このモードでは:

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、セッションのインタラクティブ性が向上します。

Thinwireプログレッシブ表示1の画像](/en-us/linux-virtual-delivery-agent/2607-ltsr/media/thinwireprogressivedisplay.png)

セッションとの対話を停止すると、劣化した画像とテキストは、徐々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が含まれなくなり、画像は元の品質で表示されます。

Thinwireプログレッシブ表示2の画像](/en-us/linux-virtual-delivery-agent/2607-ltsr/media/thinwireprogressivedisplay2.png)

画像の場合、シャープ化にはランダムなブロック状の方法が使用されます。テキストの場合、個々の文字または単語の一部がシャープ化されます。シャープ化プロセスは複数のフレームにわたって行われます。このアプローチにより、単一の大きなシャープ化フレームによる遅延の発生が回避されます。

一時的な画像（ビデオ）は、アダプティブディスプレイまたは選択的H.264で引き続き管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは、**視覚品質**ポリシー設定の**高**、**中**（デフォルト）、**低**に対してスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフになります（使用されません）。

- **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

モード切り替えが発生した後、たとえネットワークの状態が悪化しても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、次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 \<しきい値 (ミリ秒)\> (デフォルト = 200)

例: 100 = ネットワーク遅延が100ミリ秒を下回った場合にプログレッシブモードをオンにする。

一般的なコンテンツリダイレクト

クライアントドライブマッピングとクライアントフォルダーリダイレクト

IF	次に
ホスト (VDA) でクライアントドライブマッピングのみを有効にすると、,	クライアント側のフルボリュームは、ホームディレクトリ内の **ctxmnt/drivers** サブディレクトリの下のセッションに自動的にマップされます。
ホスト (VDA) でクライアントフォルダーリダイレクトを有効にし、ユーザーがユーザーデバイス (クライアント) でそれを構成すると、,	ユーザーが指定したローカルボリュームの一部がリダイレクトされます。

USBデバイスリダイレクト

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。

クライアントドライブマッピング

クライアントドライブマッピングとクライアントフォルダーリダイレクトを使用して、クライアント側のファイルをホスト側のセッションでアクセスできるようにすることができます。クライアントドライブマッピングとクライアントフォルダーリダイレクトの比較は次のとおりです。

条件	結果
ホスト（VDA）でクライアントドライブマッピングのみを有効にする場合、,	クライアント側のフルボリュームは、ホームディレクトリの**ctxmnt**サブディレクトリの下のセッションに自動的にマップされます。
ホスト（VDA）でクライアントフォルダーリダイレクトを有効にし、ユーザーがユーザーデバイス（クライアント）で構成する場合、,	ユーザーが指定したローカルボリュームの一部がリダイレクトされます。

クライアントドライブマッピングを有効にする

クライアントドライブマッピングを有効にするには、Citrix Studioで**Client drive redirection**ポリシーを**Allowed**に設定します。このポリシーについて詳しくは、「ファイルリダイレクトポリシー設定」を参照してください。

クライアントフォルダーリダイレクトを有効にし、リダイレクトするフォルダーを指定する

クライアントフォルダーリダイレクトを有効にするには、VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Client Folder
Redirection" -t "REG_DWORD" -v "CFROnlyModeAvailable" -d "0x00000001" --
force
```

クライアントからホスト側のセッションにリダイレクトするフォルダーを指定するには、ユーザーデバイスで次の手順を実行します。

1. Citrix Workspace™アプリの最新バージョンがインストールされていることを確認してください。
2. Citrix Workspaceアプリのインストールディレクトリから、**CtxCFRUI.exe**を起動します。
3. **Custom** ラジオボタンを選択し、フォルダーを追加、編集、または削除します。
4. 設定を有効にするには、セッションを切断して再接続します。

USBデバイスリダイレクト

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ミリ秒未満の場合は、USBデバイスリダイレクトを使用することをお勧めします。ネットワーク遅延が200ミリ秒を超える場合は、USBデバイスリダイレクトを使用しないでください。

> USBデバイスリダイレクトには、主に次の3つの機能領域があります。

- オープンソースUSB/IPプロジェクト

- Citrix USBセッションモジュール
- Citrix USBサービスモジュール

オープンソースUSB/IPプロジェクト：

USB/IPプロジェクトは、Linuxカーネルドライバーと、カーネルドライバーと通信してすべてのUSBデータを取得できるいくつかのユーザーモードライブラリで構成されています。

Linux VDAは、オープンソースのUSB/IPプロジェクトに基づいてUSBデバイスリダイレクトを実装し、USB/IPのカーネルドライバーとユーザーモードライブラリを再利用します。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 USBプロトコルによってカプセル化されます。

Citrix USBセッションモジュール：

Citrix USBセッションモジュールは、USB/IPカーネルモジュールとCitrix Workspaceアプリ間の通信ブリッジとして機能します。

Citrix USBサービスモジュール：

Citrix USBサービスモジュールは、USBデバイスの接続や切断など、USBデバイスに対するすべての操作を管理します。

USBデバイスリダイレクトの仕組み

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム/devパスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで利用できない場合があります。USBデバイスは正常に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスできません。USBデバイスの接続性を確保するには、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、クライアントから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

> **ヒント:**

USB 3.0ポートのサポートを追加しました。クライアントデバイスのUSB 3.0ポートにUSB 3.0デバイスを挿入できます。

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検証されています。その他のデバイスは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。

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, NTFS
`Kingston Datatraveler 101 II`	0951:1625	FAT32, NTFS
`Kingston Datatraveler GT101 G2`	1567:8902	FAT32, NTFS
`SanDisk SDCZ80 flash drive`	0781:5580	FAT32, NTFS
`WD HDD`	1058:10B8	FAT32, NTFS
`Toshiba Kingston DataTraveler 3.0 USB device`	0930:6545	FAT32, NTFS
`Taiwan OEM – OBSOLETE VendorCo ProductCode Disk 2.0`	FFFF:5678	FAT32, NTFS
`TD-RDF5A Transcend USB device`	8564:4000	FAT32, NTFS

注:

RHEL、Rocky Linux、SUSE で NTFS を使用するには、まずこれらのディストリビューションで NTFS サポートを有効にしてください。

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

ユビコ USB	VID:PID
`Yubico YubiKey OTP+FIDO+CCID -Keyboard, HID`	1050:0407

ウェブカメラ USB	VID:PID
`Logitech composite USB device – WebCam, Audio`	0460:0825

USBデバイスリダイレクトの設定

(RHELおよびRocky Linuxのみ) USB/IPカーネルモジュールのインストールまたはコンパイル

Linux VDA は、USB デバイスリダイレクトの仮想ホストコントローラーとして USB/IP を使用します。ほとんどの場合、USB/IP カーネルモジュールは Linux カーネルバージョン 3.17 以降でリリースされているため、デフォルトでカーネルモジュールをビルドする必要はありません。しかし、USB/IP カーネルモジュールは RHEL および Rocky Linux では利用できません。これらの Linux ディストリビューションで USB デバイスリダイレクトを使用するには、USB/IP カーネルモジュールをインストールまたはコンパイルする必要があります。お使いの Linux ディストリビューションに基づいて、`<https://pkgs.org/download/kmod-usbip>` から USB/IP をダウンロードしてインストールしてください。

USBデバイスリダイレクトポリシーの設定

Citrixポリシーは、USBデバイスリダイレクトが有効か無効かを制御します。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デバイスリダイレクトを構成する場合、以下のポリシーを構成します。

- 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デバイスリダイレクトの有効化

Citrix Studioで、クライアントからの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デバイスリダイレクトを有効にする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/client-usb-redirection-policy.png)

USBデバイスリダイレクトルールを設定する

USBリダイレクトポリシーを有効にした後、Citrix Studioを使用して、Linux VDAで許可（または拒否）するデバイスを指定することで、リダイレクトルールを設定します。

Client USB device redirection rulesダイアログで、次の操作を行います。

1. リダイレクトルールを追加するには**New**をクリックするか、既存のルールを確認するには**Edit**をクリックします。
2. ルールを作成（または編集）したら、**OK**をクリックします。

クライアントUSBデバイスリダイレクトルール設定(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/client-usb-redirection-rule.png)

> **注:**
>
> USBデバイスリダイレクトを構成する際は、**Client USB device redirection**オプションを**Allowed**に設定し、**Client USB device redirection rules**を**Allow:#all ok**として構成してください。両方の設定を構成しないと、セッション内でクリックしたときに外部マウスが消える可能性があります。

Azure上のUbuntuでのUSBリダイレクトのサポート

デフォルトでは、Azure上のUbuntuには特定のUSBカーネルドライバーが含まれていない場合があります。以下の手順に従って、汎用デバイスと特定のデバイスの両方でUSBリダイレクトを有効にできます。

汎用デバイスのUSBリダイレクトを有効にする

標準的なUSBストレージデバイス、YubiKey USB HIDデバイス、ASIXネットワークアダプターなどの汎用USBデバイスの場合、以下の構成スクリプトを実行するだけで済みます。

```
sudo /opt/Citrix/VDA/sbin/usbazure/ctxusbcfg.sh
```

スクリプトを実行した後、Citrix USBサービスを再起動してサポートをアクティブにします。

```
sudo service ctxusbsd restart
```

特定のデバイス用のUSBカーネルドライバーモジュールを構築する

すぐにサポートされていないデバイスの場合、以下の一般的なワークフローを使用して、必要なカーネルドライバーモジュールを構築およびインストールできます。以下の例では、説明のために `Kingston DataTraveler 2.0 Stick (VID=0930, PID=6544)` を使用しています。

前提条件:

- Linux VDAパッケージがインストールされたVDAマシン。
- 汎用Linuxカーネルを実行しているLinux物理マシン。

Linux物理マシンで必要なカーネルドライバーを特定する

ステップ1 - デバイスを接続する

USBデバイスを、汎用Linuxカーネルを実行している物理マシンに接続します。

ステップ2 - USBカーネルドライバー識別スクリプトを準備する

`/opt/Citrix/VDA/sbin/usbazure/ctxusbkodriver.sh` を物理マシンにコピーし、実行可能にします。

ステップ3 - デバイスのVIDとPIDを取得する

接続されているUSBデバイスを特定するには、**lsusb**を使用します。

```
lsusb

Example output:

Bus 001 Device 040: ID 0930:6544 Toshiba Corp. TransMemory-Mini / Kingston
DataTraveler 2.0 Stick
```

ステップ4 - 必要なカーネルドライバーを特定する

デバイスの `VID:PID` を使用してヘルパースクリプトを実行します。

```
./ctxusbkodriver.sh 0930:6544

Example output:

All USB Kernel drivers for 0930:6544:

usb-storage
```

VDAマシンにカーネルドライバーをビルドしてインストールする

ステップ1 - ビルド環境を構成する

構成スクリプトを実行してビルド環境をセットアップし、必要なカーネルソースコードを`/root/.ctxusb`にダウンロードします。

`sudo /opt/Citrix/VDA/sbin/usbazure/ctxusbcfg.sh`

ステップ2 - カーネルビルド構成を決定する

構成スクリプトを使用して、適切なカーネルモジュールビルドオプションを取得します。

```
sudo /opt/Citrix/VDA/sbin/usbazure/ctxusbkoconfig.sh /root/.ctxusb/linux
usb-storage

Example output:

Config option: CONFIG_USB_STORAGE

Folder: /root/.ctxusb/linux/drivers/usb/storage
```

ステップ3 - カーネル構成オプションの検査と記録

```
cd /root/.ctxusb/linux

make menuconfig
```

- 検索機能（/ を押す）を使用してオプション（例：CONFIG_USB_STORAGE）を見つけます。
- 関連する設定（例：CONFIG_USB_SUPPORT=y、CONFIG_USB_STORAGE=m）を記録します。
- 変更を保存せずに終了します。

ステップ4 - カーネルドライバーのビルドとインストール

カーネルモジュールをコンパイルしてインストールします。

```
sudo /opt/Citrix/VDA/sbin/usbazure/ctxusbkobuild.sh /root/.ctxusb/linux/drivers/usb/storage "CONFIG_USB_SUPPORT=y CONFIG_USB_STORAGE=m"
```

ステップ5 - カーネルドライバーの検証

```
modinfo usb-storage
```

正常に完了すると、指定したUSBデバイス用のカーネルドライバーがビルドされ、インストールされます。

USBデバイス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usb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。そのため、デバイスは管理者権限でのみアンマウントできます。

デバイスをアンマウントできません(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/unable-to-mount-disk.png)

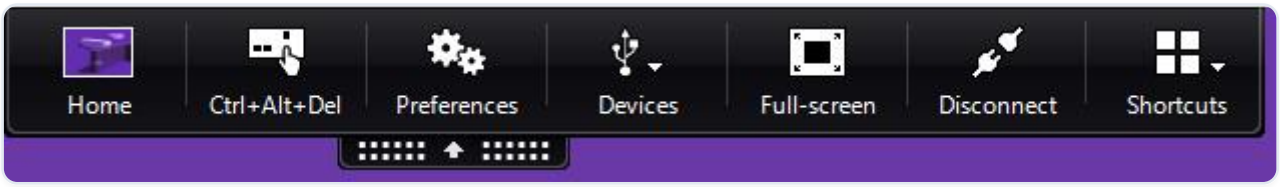
USBディスクのリダイレクトを停止するとファイルが失われる

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ときに、システムがファイルシステムにメモリキャッシュをマウントするために発生します。データはディスク自体には書き込まれません。Citrix Workspaceアプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。

この問題を解決するには、USBリダイレクトを停止する前に、ターミナルで **sync** コマンドを使用してデータをディスクにフラッシュします。

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デバイス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こと。
- Citrix USBサービスモジュールが実行されていること。

ポリシーが正しく設定されていない場合は、この記事の「USBデバイスリダイレクトポリシーの設定」セクションを参照して修正してください。

Citrix USBサービスモジュールが実行されていない場合は、次の手順を実行します。

1. 次のコマンドを使用して、お使いのLinuxディストリビューションでUSB/IPカーネルモジュールが利用可能かどうかを確認します。

   ```
   modinfo usbip-core
   ```

2. 出力が次のように表示される場合は、お使いのLinuxディストリビューションに基づいてUSB/IPカーネルモジュールをインストールまたはコンパイルします。

   ```
   modinfo: ERROR: Module usbip-core not found.
   ```

   - RHELおよびRocky Linuxについては、この記事の「USB/IPカーネルモジュールのインストールまたはコンパイル」セクションを参照してください。
   - SUSEの場合は、`<https://software.opensuse.org/package/usbip>`からUSB/IPパッケージをダウンロードしてインストールします。
   - Ubuntu/Debianの場合は、次の手順を実行してUSB/IPカーネルモジュールをコンパイルおよびインストールします。

     a. USB/IP カーネルモジュールのソースコードをダウンロードします。

     `<https://github.com/torvalds/linux/tree/master/drivers/usb/usbip>` の Linux カーネルリポジトリにアクセスし、ターゲットの Linux カーネルバージョン (v4.15 以降) タグを選択して、`<https://github.com/torvalds/linux/tree/v4.15/drivers/usb/usbip>` のようなリンクを取得します。

     DownGit にアクセスし、上記のリンクを入力して USB/IP ソースコードをダウンロードするためのダウンロードリンクを作成します。

b. 次のコマンドを使用してソースファイルを解凍します。

```
unzip ${USBIP_SRC}.zip



cd usbip
```

c. **Makefile** ファイルを次のように変更します。

```
# SPDX-License-Identifier: GPL-2.0



ccflags-$(CONFIG_USBIP_DEBUG) := -DDEBUG



obj-$(CONFIG_USBIP_CORE) += usbip-core.o



usbip-core-y := usbip_common.o usbip_event.o



obj-$(CONFIG_USBIP_VHCI_HCD) += vhci-hcd.o



vhci-hcd-y := vhci_sysfs.o vhci_tx.o vhci_rx.o vhci_hcd.o



#obj-$(CONFIG_USBIP_HOST) += usbip-host.o



#usbip-host-y := stub_dev.o stub_main.o stub_rx.o stub_tx.o
```

```
#obj-$(CONFIG_USBIP_VUDC) += usbip-vudc.o



#usbip-vudc-y := vudc_dev.o vudc_sysfs.o vudc_tx.o vudc_rx.o
vudc_transfer.o vudc_main.o
```

d. ソースコードをコンパイルします。

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



make -C /lib/modules/`uname -r`/build M=$PWD
```

e. USB/IP カーネルモジュールをインストールします。

```
cp usbip-core.ko vhci-hcd.ko /opt/Citrix/VDA/lib64/
```

f. USB/IP カーネルモジュールをロードするために、**ctxusbsd** サービスを再起動します。

```
systemctl restart ctxusbsd
```

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」と表示される場合のリダイレクトの失敗

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ますが、セッションで使用できません

通常、サポートされている USB デバイス のみがリダイレクトできます。他のデバイスもアクティブな Linux VDA セッションにリダイレクトされる場合があります。リダイレクトされたすべての

デバイスについて、ユーザーが所有するノードがシステム **/dev** パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている (接続されている) がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限のポリシーに追加します。

> **注:**
>
> USBドライブの場合、Linux VDAはディスクを構成してマウントします。ユーザー（およびそれをインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。サポートされているデバイスリストにないデバイスの場合、そうではない可能性があります。

クリップボードリダイレクト

クリップボードリダイレクトを使用すると、VDAセッションで実行されているアプリケーションとクライアントデバイスで実行されているアプリケーションの間でデータをコピー＆ペーストできます。

この記事では、クリップボードリダイレクトを実現するために利用できるCitrix®ポリシーについて説明します。

クリップボードリダイレクトのCitrixポリシー

クライアントクリップボードリダイレクト

この設定は、クライアントデバイス上のクリップボードがVDA上のクリップボードにマップされることを許可または禁止します。

デフォルトでは、クリップボードリダイレクトは**許可**に設定されています。

セッションとローカルクリップボード間のコピー＆ペーストデータ転送を禁止するには、**禁止**を選択します。ユーザーは、セッションで実行されているアプリケーション間では引き続きデータをコピー＆ペーストできます。

クリップボードリダイレクトの帯域幅制限

この設定は、セッションとローカルクリップボード間のデータ転送に許可される最大帯域幅（kbps）を指定します。

クリップボードリダイレクトの帯域幅制限（パーセント）

この設定は、セッションとローカルクリップボード間のデータ転送に許可される最大帯域幅を、総セッション帯域幅の割合として指定します。

クリップボードのクライアントからセッションへの転送サイズを制限する（ファイルタイプ以外の制限）

この設定は、単一のコピー＆ペースト操作でクライアントデバイスから仮想セッションに転送できるクリップボードデータの最大サイズを指定します。

クリップボードの転送サイズを制限するには、**クリップボードのクライアントからセッションへの転送サイズを制限する**設定を有効にします。次に、**サイズ制限**フィールドにキロバイト単位の値を入力して、ローカルクリップボードとセッション間のデータ転送サイズを定義します。

デフォルトでは、この設定は無効になっています。

クリップボードのセッションからクライアントへの転送サイズを制限する（ファイルタイプ以外の制限）

この設定は、単一のコピー＆ペースト操作で仮想セッションからクライアントデバイスに転送できるクリップボードデータの最大サイズを指定します。

クリップボードの転送サイズを制限するには、**クリップボードのセッションからクライアントへの転送サイズを制限する**設定を有効にします。次に、**サイズ制限**フィールドにキロバイト単位の値を入力して、セッションとローカルクリップボード間のデータ転送サイズを定義します。

デフォルトでは、この設定は無効になっています。

クリップボードのクライアントからセッションへのファイルタイプ転送サイズを制限する

この設定は、クリップボードのコピー＆ペーストを介してクライアントからセッションに転送できるファイルの最大サイズを指定します。

この制限を適用するには、VDAローカルレジストリで対応する値（KB単位）を設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard"
-t "REG_DWORD" -v "FileCopySizeLimitC2H"  -d "<YourLimitSizeInKB>" --force
```

デフォルト値は0x00032000（200MB）です。

クリップボードのセッションからクライアントへのファイルタイプ転送サイズを制限する

この設定は、クリップボードのコピー＆ペーストを介してセッションからクライアントに転送できるファイルの最大サイズを指定します。

この制限を適用するには、VDAローカルレジストリで対応する値（KB単位）を設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Clipboard"
-t "REG_DWORD" -v "FileCopySizeLimitH2C" -d "<YourLimitSizeInKbyte>" --
force
```

デフォルト値は0x00032000（200MB）です。

クライアントクリップボードへの書き込みを制限 AND クライアントクリップボードへの書き込み許可形式

これら2つの設定を有効にすると、特定のデータ形式をセッションからクライアントにコピー＆ペースト（クライアントへの書き込み）できるようになります。

次のクリップボード形式はシステム定義です。

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

セッションクリップボードへの書き込みを制限 AND セッションクリップボードへの書き込み許可形式

これら2つの設定を有効にすると、特定のデータ形式をクライアントからセッションにコピー＆ペースト（セッションへの書き込み）できるようになります。

次のクリップボード形式はシステム定義です。

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

キーボード

このセクションには、以下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

Xキーボード拡張 (XKB) の構成

2407リリース以降、Linux VDAはキーボード構成のデフォルトのXKBルールとしてevdevを指定します。この決定にはいくつかの理由があります。

- ほとんどの最新のLinuxディストリビューションは、デフォルトでevdev XKBルールを使用しています。
- ほとんどすべてのアプリケーションが、evdev XKBルールのキーコードパターンに従っています。

- evdev XKBルールは、基本的な英数字や一般的なファンクションキーを超えて、「Cancel」、「Redo」、「Undo」、「XF86Copy」、「XF86Open」、「XF86Paste」などの幅広いキーシンボルに対して拡張サポートを提供します。
- Linuxのevdev XKBルールは、さまざまなキーボード関連の問題、特にキー入力への誤った応答に関する問題に対処します。

XFree86ルールに戻したい場合は、次のレジストリ設定を行うことができます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "XkbRuleName" -d "xorg"
```

このコマンドでは、大文字と小文字を区別する正確なテキスト「xorg」のみが「xfree86」ルールセットとして認識され、その他の入力値は無効です。

クライアント入力方式エディター (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のインストール時に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をダブルクリックすることは、クライアントIME機能を使用してセルに文字を入力する前に必須です。
- 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、入力領域の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/2607-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次のようにさらに改善します。

現在のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/2607-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：
   - ウブンツ: `<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:**

> **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同期とIME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同期とIME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同期とIME改善ポリシーの画像(/en-us/linux-virtual-delivery-agent/2607-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同期とIME改善ポリシーを設定する画像(/en-us/linux-virtual-delivery-agent/2607-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`ユーティリティを介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するには:

  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す。

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:**
>
> HTML5向けCitrix Workspace™アプリは、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

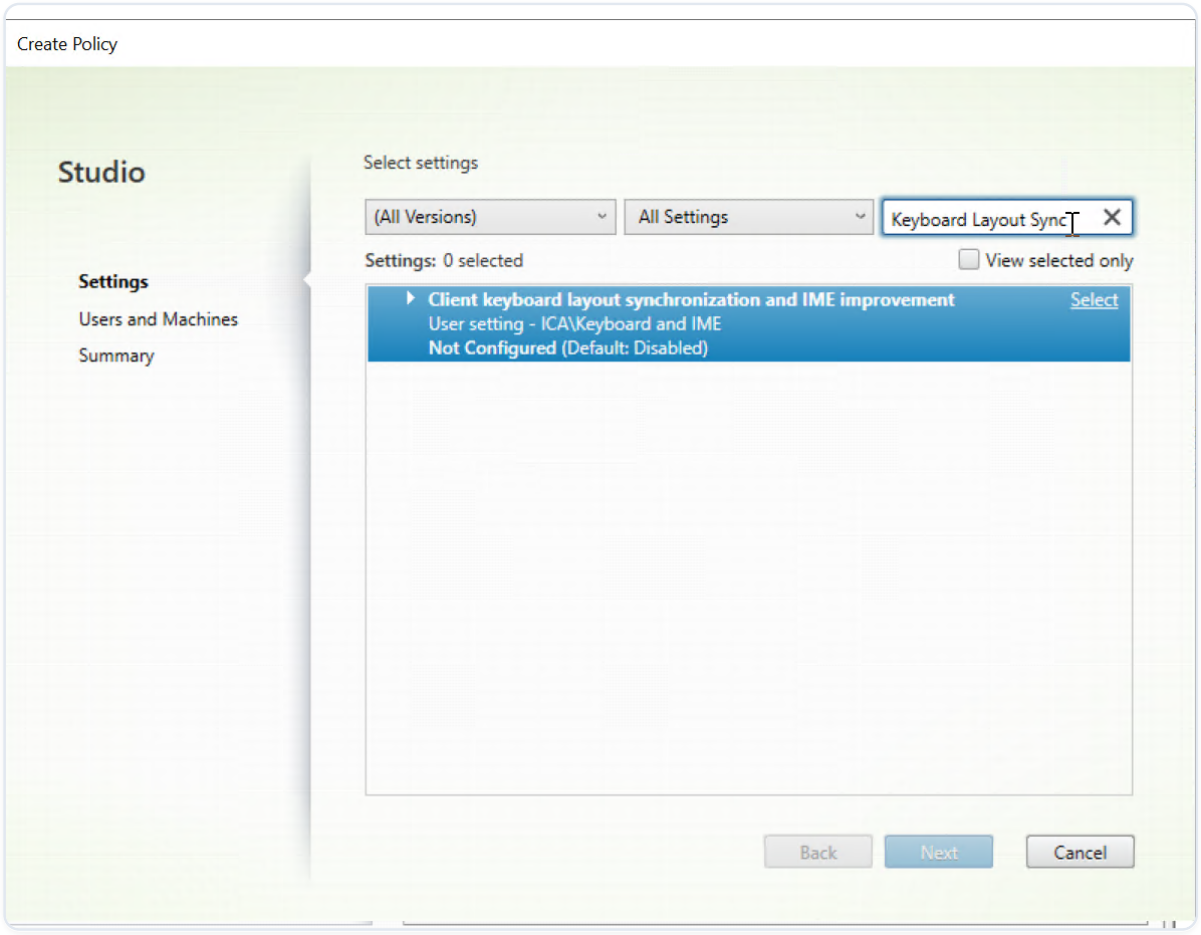
動的キーボードレイアウト同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、`ctxreg`ユーティリティを介してレジストリを編集します。

> **注:**
>
> **Client Keyboard Layout Sync and IME Improvement**ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- 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。

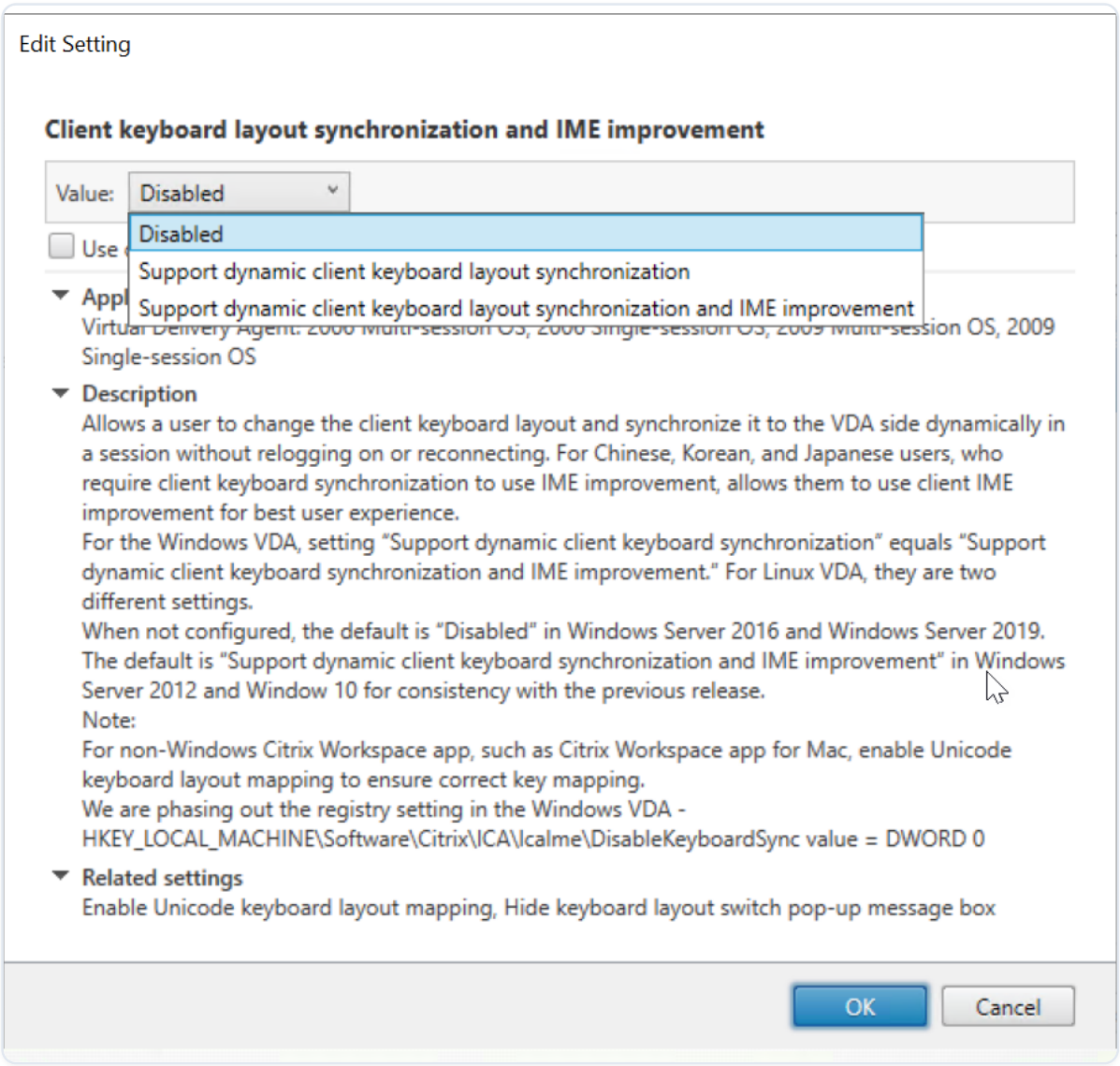
  a. Studioで、**Policies**を右クリックし、**Create Policy**を選択します。

  b. **Client Keyboard Layout Sync and IME Improvement**ポリシーを検索します。



  c. ポリシー名の横にある**Select**をクリックします。

  d. ポリシーを設定します。



3つのオプションが利用可能です。

- **Disabled**：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。
- **Support dynamic client keyboard layout synchronization**：`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。
- **動的な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、動的な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。

- 動的なキーボードレイアウト同期機能を有効または無効にするには、`ctxreg`ユーティリティを介してレジストリを編集します。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000000"
  ```

使用方法

この機能を有効にすると、セッション中にクライアントデバイスでキーボードレイアウトが変更された場合、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語 (FR) に変更した場合：

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/2607-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/2607-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。

タスクバーにフランス語が表示されている画像(/en-us/linux-virtual-delivery-agent/2607-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。

ソフトキーボード(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/soft-keyboard.png)

> **注:**
>
> この機能は、iOSおよびAndroid版のCitrix Workspace™アプリ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. 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:**
>
> 前述の設定は、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、ある入力フィールドから別の入力フィールドをクリックしても、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、韓国語の文字など）をサポートしていません。
- ソフトキーボードはパスワード入力フィールドでは利用できません。
- ソフトキーボードが現在の入力フィールドと重なる場合があります。この場合、アプリ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させてください。
- 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キーボードが表示されます。

複数言語入力のサポート

Linux VDAバージョン1.4以降、Citrixは公開アプリケーションのサポートを追加しました。ユーザーは、Linuxデスクトップ環境なしで目的のLinuxアプリケーションにアクセスできます。

しかし、Linux VDAのネイティブ言語バーは、Linuxデスクトップ環境と高度に統合されているため、公開アプリケーションでは利用できませんでした。その結果、ユーザーは中国語、日本語、韓国語などのIMEを必要とする言語でテキストを入力できませんでした。また、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け付ける公開アプリケーションに言語バーを提供します。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

使用方法は簡単です。

1. 機能を有効にします。
2. テキスト入力を受け付ける公開アプリケーションにアクセスします。セッションに、アプリケーションの横に言語バーが表示されます。
3. ドロップダウンメニューから、**地域と言語**を選択して、目的の言語（入力ソース）を追加します。

   地域と言語の画像(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/select-region-language-dialog-box.png)
4. ドロップダウンメニューからIMEまたはキーボードレイアウトを選択します。
5. 選択したIMEまたはキーボードレイアウトを使用して言語を入力します。

> **注:**

- VDA側の言語バーでキーボードレイアウトを変更する場合は、クライアント側（Citrix Workspace™アプリを実行している）で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。
- **accountsservice** パッケージは、**地域と言語**ダイアログボックスで設定を行う前に、バージョン0.6.37以降にアップグレードする必要があります。

IMEまたはキーボードレイアウトを選択します(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/region-language-dialog-box.png)

マルチメディア

このセクションには、以下のトピックが含まれています。

- オーディオ機能(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/multimedia/audio-features.html)
- 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(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/multimedia/hdx-webcam-video-compression.html)

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、環境に合わせて設定を最適化し、古いオーディオ圧縮形式を置き換えて、優れたユーザーエクスペリエンスを提供します。

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

リストに含まれていないクライアントを使用する場合、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。

オーディオサービスに関する考慮事項

Linux VDAは、次の場合にPipeWireをデフォルトのオーディオサービスとして使用します。

- VDA バージョン 2407 以降が、Debian 12.x、RHEL 9.x/10.x、または Rocky Linux 9.x/10.x にインストールされていること。
- Ubuntu 24.04にインストールされているVDAバージョン2411以降。
- Debian 13.xまたはUbuntu 26.04にインストールされているVDAバージョン2607以降。

その他のすべてのディストリビューションでは、Linux VDAは引き続きPulseAudioを使用します。RHEL 8.xおよびRocky Linux 8.xでは、PulseAudio 13.99以降を使用してください。

次の表は、サポートされている各ディストリビューションで使用されるオーディオバックエンドの概要を示しています。

ディストリビューション	オーディオバックエンド	サポートされるVDAバージョン
デビアン 12.x	PipeWire	2407
デビアン 13.x	PipeWire	2607
RHEL 9.x/10	PipeWire	2407
ロッキー Linux 9.x/10	PipeWire	2407
ウブントゥ 24.04	PipeWire	2411
ウブントゥ 26.04	PipeWire	2607
RHEL 8.x、ロッキー Linux 8.x	PulseAudio (13.99以降)	—
その他のディストリビューション (SUSE Linux Enterprise 15.xを含む)	PulseAudio	—

PulseAudio から PipeWire への切り替え

RHEL 9.x/10 または Rocky Linux 9.x/10 に 2407 より前のバージョンの Linux VDA がインストールされている場合、PulseAudio を使用しています。PulseAudio を引き続き使用するには、何もする必要はありません。PipeWire に切り替えるには、次の手順を実行します。

1. PulseAudio を PipeWire に置き換えます。

```
dnf swap --allowerasing pulseaudio pipewire-pulseaudio
```

2. PipeWire サービスをグローバルに有効にします。

```
systemctl --global --now enable pipewire pipewire-pulse pipewire-
pulse.socket wireplumber
```

3. Linux VDA を再起動します。

英語以外のシステム言語で PipeWire を使用する

システム言語が英語以外で PipeWire が使用されている場合は、/etc/xdg/autostart/ctxaudio.desktop を編集し、次の行を見つけます。

```
Exec=/opt/Citrix/VDA/bin/ctxaudio
```

英語以外の環境で PipeWire との互換性を確保するため、C ロケールを強制的に適用するには、次の行に置き換えます。

```
Exec=env LC_ALL=C /opt/Citrix/VDA/bin/ctxaudio
```

オーディオのロス許容モード

損失許容モードはオーディオをサポートします。この機能は、リアルタイムストリーミングのユーザーエクスペリエンスを向上させ、高遅延およびパケット損失のあるネットワークを介してユーザーが接続している場合の EDT 経由のオーディオ品質を改善します。

損失許容モードとEDTの詳細については、Citrix Virtual Apps and Desktopsドキュメントの「追加情報」を参照してください。

オーディオ機能の損失許容モードを有効にする

オーディオの損失許容モードはデフォルトで有効になっています。無効になっている場合は、次の手順を実行して再度有効にします。

1. HDX アダプティブトランスポート ポリシーを設定して、アダプティブトランスポートを有効にします。アダプティブトランスポートはデフォルトで有効になっています。
2. アダプティブオーディオ ポリシーを設定して、アダプティブオーディオを有効にします。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。

3. オーディオの損失許容モード ポリシーを設定して、損失許容モード（EDT 信頼性の低いトランスポート）を有効にします。
4. 直接接続の場合、VDA での DTLS の有効化 が必要です。
5. リモート接続の場合、EDTロス許容モードは、(/ja-jp/citrix-gateway-service/hdx-edt-support-for-gateway-service.html)または(/ja-jp/netscaler-gateway/2607-ltsr/hdx-enlightened-data-transport-support/configuring-citrix-gateway%23configure-loss-tolerant-mode-for-audio#configure-loss-tolerant-mode-for-audio)でもサポートされている必要があります。

クライアントの要件と設定

オーディオでロス許容モードを使用するには、Citrix Workspaceアプリがこの機能をサポートし、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。そうでない場合、オーディオはEDT Reliable transportにデフォルト設定されます。

損失許容モードをサポートするCitrix Workspaceアプリの最小バージョンは次のとおりです。

- Windows向けCitrix Workspaceアプリの最小バージョン2309
- Linux向けCitrix Workspaceアプリの最小バージョン2311
- Mac向けCitrix Workspaceアプリの最小バージョン2311

オーディオ診断コマンドラインツール

VDA上のオーディオ診断コマンドラインツールは、オーディオポリシー、構成、データ転送に関連するセッションデータを照会するために使用できます。

使用方法

コマンドプロンプトを開き、`/opt/Citrix/VDA/bin`フォルダーから`ctxaudiosession`を実行します。

このツールを実行すると、現在のユーザーのアクティブなICA®セッションのオーディオ情報とデバイスリダイレクトステータスがすべて表示されます。

出力

このツールは、セッション内のオーディオ関連の問題の診断に役立つさまざまな構成設定を出力します。

セクション	説明
警告	デバイスステータス、トランスポートタイプ、オーディオコーデックなどに関するオーディオサービスの警告メッセージ.

セクション	説明
状態情報	オーディオの状態、バージョン、コーデック、現在のセッションに適用されるトランスポートなど.
ポリシー設定	現在のセッションに適用されているオーディオポリシー。
ローカル設定	レジストリまたはローカル設定に保存されているオーディオ関連の構成。
機能	CWAとVDA間のオーディオ機能の結果。
サウンドデバイス	セッション内のデバイス名、その役割、および実行ステータス。

EDT損失許容モード用オーディオ品質エンハンサー

バージョン2507以降、(/ja-jp/citrix-virtual-apps-desktops/multimedia/audio.html#loss-tolerant-mode-for-audio)を介したアダプティブオーディオでは、オーディオ品質エンハンサーがデフォルトで有効になっています。

オーディオ品質エンハンサーは、短時間のネットワーク中断中もクリアなオーディオを維持します。この機能は、ネットワークの状態に適応し、再生中および録音中の一貫したオーディオパフォーマンスを保証します。

> **注:**
>
> この機能が動作するには、適応型オーディオを有効にする必要があります。

信頼性の高いトランスポート用オーディオ品質エンハンサー (プレビュー)

テクニカルプレビュー版2511以降、オーディオ品質エンハンサーは信頼性の高いトランスポート（つまりTCPおよびEDT-Reliable）を介した接続でも利用できるようになりました。ネットワークの状態が悪い場合でも、オーディオの再生と録音の両方で、遅延が少なく、よりスムーズなエクスペリエンスが得られます。

この機能を**有効にする**には、次の手順を実行します。

1. Linux VDAでこの機能を有効にします。

ターミナル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\VirtualChannels\Audio" -v
"fAlwaysUseSpeexJitterBuffer" -t REG_DWORD -d 0 –force
```

2. Citrix Workspaceアプリ for Linuxでこの機能を有効にします。

   a. `<ICAROOT>/config`フォルダーに移動し、`module.ini`ファイルを開きます。

   b. セクションに移動し、次のエントリを追加します。

   `AlwaysUseSpeexJitterBuffer=FALSE`

この機能を**無効にする**には、次の手順を実行します。

1. Linux VDAでこの機能を無効にします。

   ターミナルで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\VirtualChannels\Audio" -v "fAlwaysUseSpeexJitterBuffer" -t REG_DWORD -d 1`

2. Citrix Workspaceアプリ for Linuxでこの機能を無効にします。

   a. `<ICAROOT>/config`フォルダーに移動し、`module.ini`ファイルを開きます。

   b. セクションに移動し、次のエントリを追加します。

   `AlwaysUseSpeexJitterBuffer=TRUE`

注:

この機能が動作するには、アダプティブオーディオを有効にする必要があります。

この機能は、テクニカルプレビュー版（信頼性の高いトランスポート経由）ではデフォルトで無効になっています。

この機能は、VDAとCWAの両方で有効にする必要があります。

> この機能は、Citrix Workspaceアプリ for Linuxバージョン2511以降、およびCitrix Workspaceアプリ for Macバージョン2508以降でサポートされています。

複数のオーディオデバイスのサポート

バージョン2311以降、Linux VDAは複数のクライアントオーディオデバイスをセッションにリダイレクトすることをサポートしています。リダイレクトされたデバイスは、CitrixAudioSinkおよびCitrixAudioSourceではなく、実際のデバイス名でセッションに表示され、デバイスの抜き差しに応じてデバイスリストが動的に更新されます。

既知の問題

Ubuntu 22.04では、PulseAudioのバグにより、オーディオデバイスの切り替えが失敗する可能性があります。この問題を回避するには、ユーザーのPulseAudio設定を削除し、セッションを再開してください。

```
rm -r ~/.config/pulse
```

オーディオ音量同期のサポート

リリース2607 LTSR以降、Linux VDAは、VDAとオーディオデバイス間のオーディオ音量同期をサポートするようになりました。VDAのオーディオ音量スライダーを使用して音量を調整できるようになり、デバイスでも同じ音量になります。これは逆の場合も同様です。この機能はデフォルトで有効になっています。

ブラウザコンテンツリダイレクト

概要

ブラウザコンテンツリダイレクトは、許可リスト内の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。この機能は、Citrix Workspace™ appを使用して、クライアント側で対応する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:**
>
> Linux VDAは、Google Chromeでの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。

> このオーバーレイWebレイアウトエンジンは、VDA上ではなくクライアント上で実行され、クライアントのCPU、GPU、RAM、およびネットワークを使用します。

リダイレクトされるのはブラウザのビューポートのみです。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/2607-ltsr/media/viewport.png)

リダイレクト用の許可リストに含まれるURLを指定するアクセス制御リストを含むStudioポリシーを構成します。特定のURLのリダイレクトを無効にするブロックリストを構成します。

許可リスト内でURLに一致するものが見つかり、ブロックリスト内には見つからない場合、仮想チャネル（CTXCSB）はCitrix Workspace app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 app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(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/viewport-desktop-viewer.png)

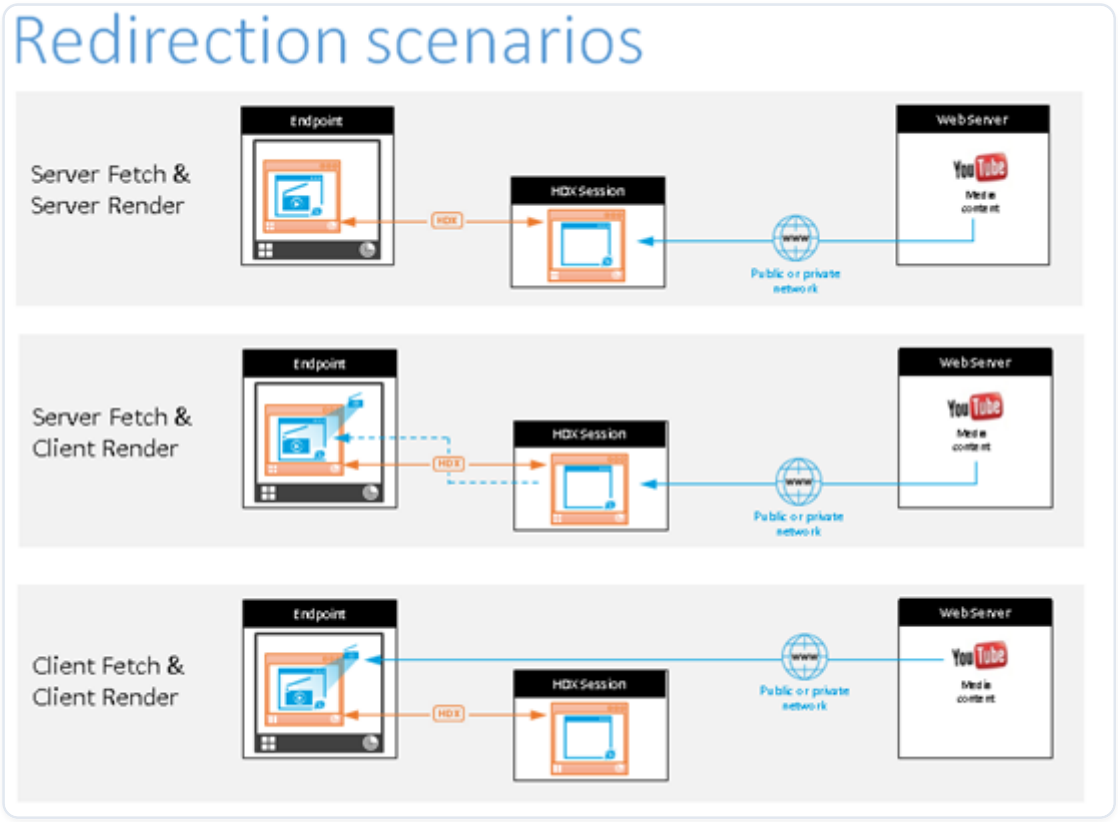
1. Citrix®ブラウザコンテンツリダイレクト拡張機能のアイコン

   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、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

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。



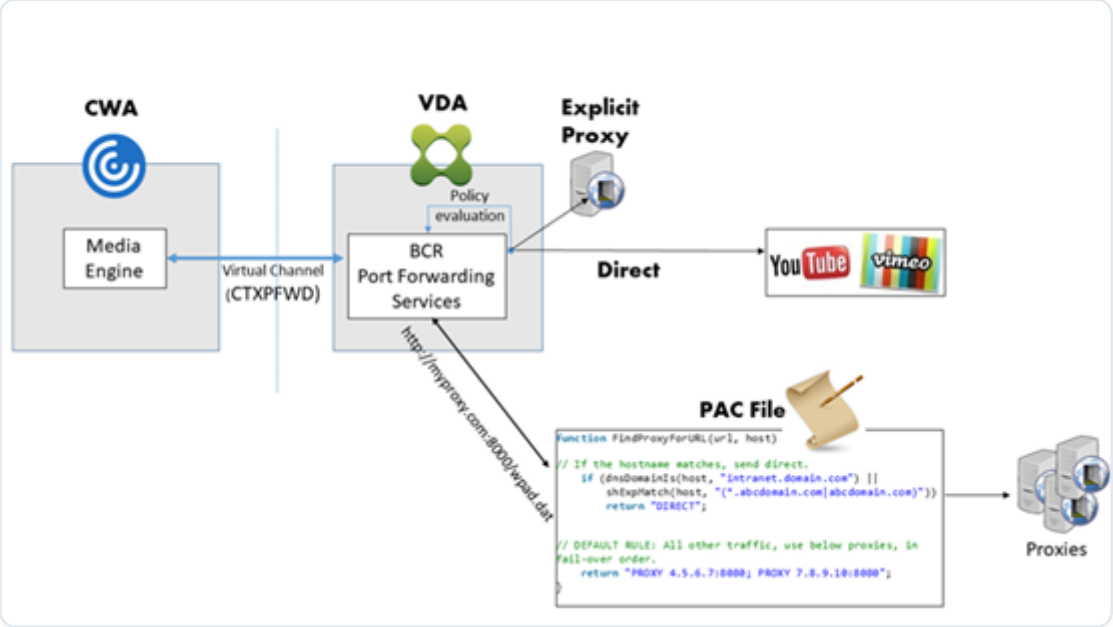
- **サーバーフェッチとサーバーレンダリング:** サイトを許可リストに追加しなかったか、リダイレクトが失敗したため、リダイレクトは行われません。VDAでWebページをレンダリングするフォールバックを行い、Thinwireを使用してグラフィックをリモート処理します。フォールバック動作を制御するには、ポリシーを使用します。このシナリオでは、VDAで高いCPU、RAM、および帯域幅が消費されます。
- **サーバーフェッチとクライアントレンダリング**: Citrix Workspaceアプリは、仮想チャネル（CTXPFWD）を使用してVDA経由でWebサーバーに接続し、コンテンツをフェッチします。このオプションは、クライアントがWebサーバーにアクセスできない場合（シンクライアントなど）に役立ちます。VDAでのCPUとRAMの消費は抑えられますが、ICA®仮想チャネルで帯域幅が消費されます。

  このシナリオには3つの動作モードがあります。CTXPFWDは、VDAがWebサーバーにアクセスするために使用するプロキシデバイスにデータを転送します。

  選択するポリシーオプション：

  - 明示的プロキシ - データセンターに単一の明示的プロキシがある場合。
  - 直接または透過的 - プロキシがない場合、または透過的プロキシを使用している場合。
  - PACファイル - 指定されたURLをフェッチするために、VDA内のブラウザが適切なプロキシサーバーを自動的に選択できるようにPACファイルに依存している場合。

詳細については、この記事の後半にある**ブラウザーコンテンツリダイレクトプロキシ構成**設定を参照してください。



- **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。

  利点:

  - より良いエンドユーザーエクスペリエンス（アダプティブビットレート（ABR））
  - VDAリソース使用量の削減（CPU/RAM/IO）
  - 帯域幅消費量の削減

システム要件

Windowsクライアント:

- Citrix Workspaceアプリ 1809 for Windows以降

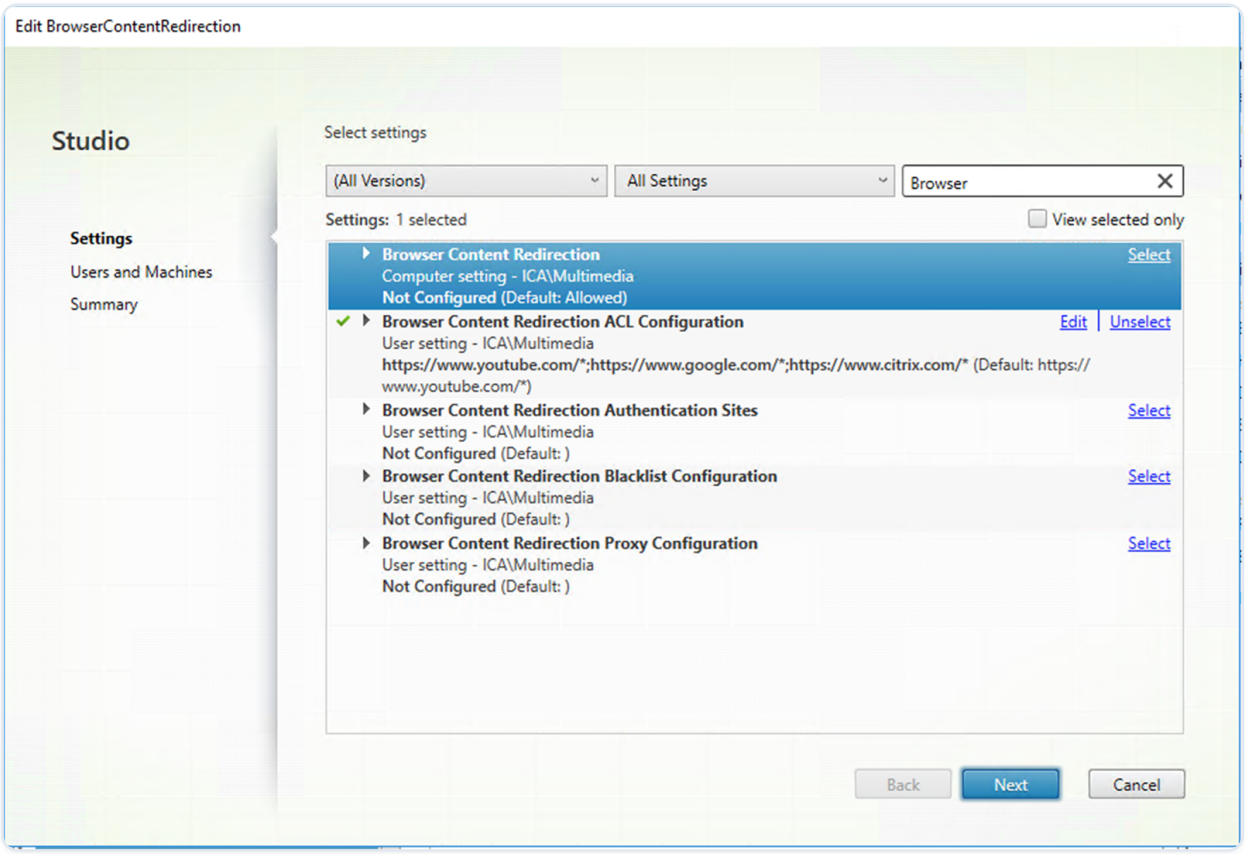
リナックス VDA:

- VDA上のブラウザー: Citrixブラウザー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

ブラウザーコンテンツリダイレクトの構成

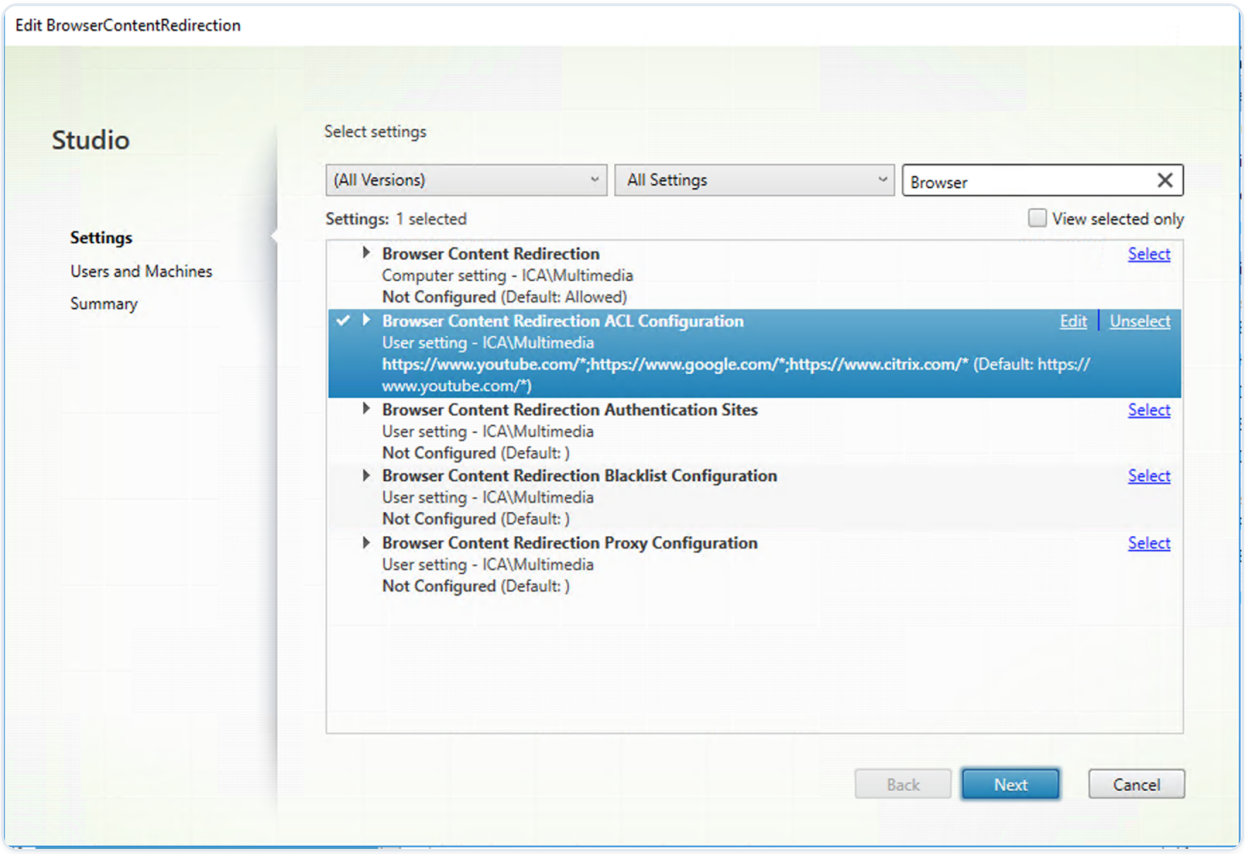
ブラウザーコンテンツリダイレクトを使用するには、関連するポリシーを構成し、Google Chromeにブラウザーコンテンツリダイレクト拡張機能をインストールします。これを行うには、次の手順を完了します。

1. Citrix Studioで、**ブラウザーコンテンツリダイレクト**を**許可**に設定して、ブラウザーコンテンツリダイレクトを有効にします。



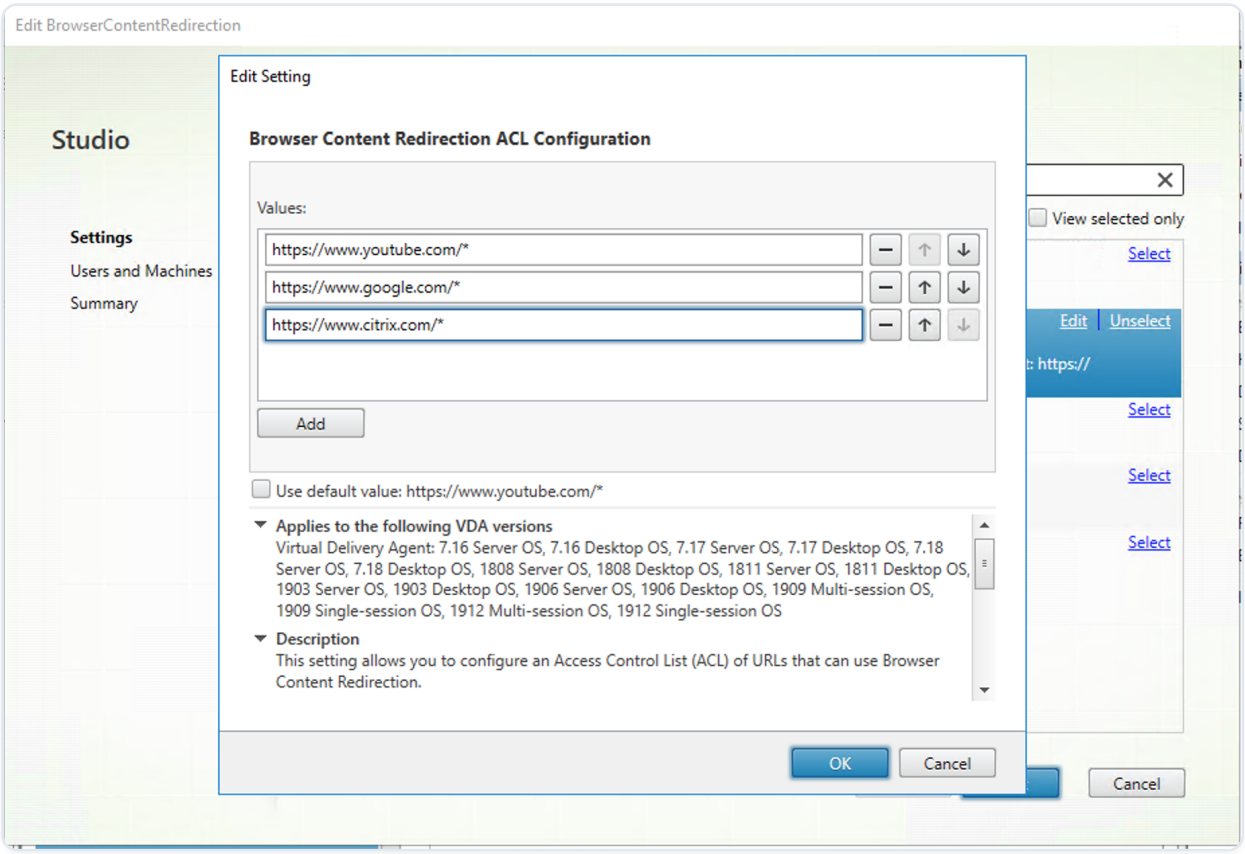
2. コンテンツをクライアントにリダイレクトできるURLの許可リストと、特定のURLのリダイレクトを無効にするブロックリストを指定します。ブロックリストの構成はオプションです。

   Browser Content Redirection ACL Configuration 設定は、コンテンツをクライアントにリダイレクトできるURLの許可リストを指定します。URLを指定する際、プロトコルを除くすべてのURLコンポーネントを表すために `*` ワイルドカードを使用できます。

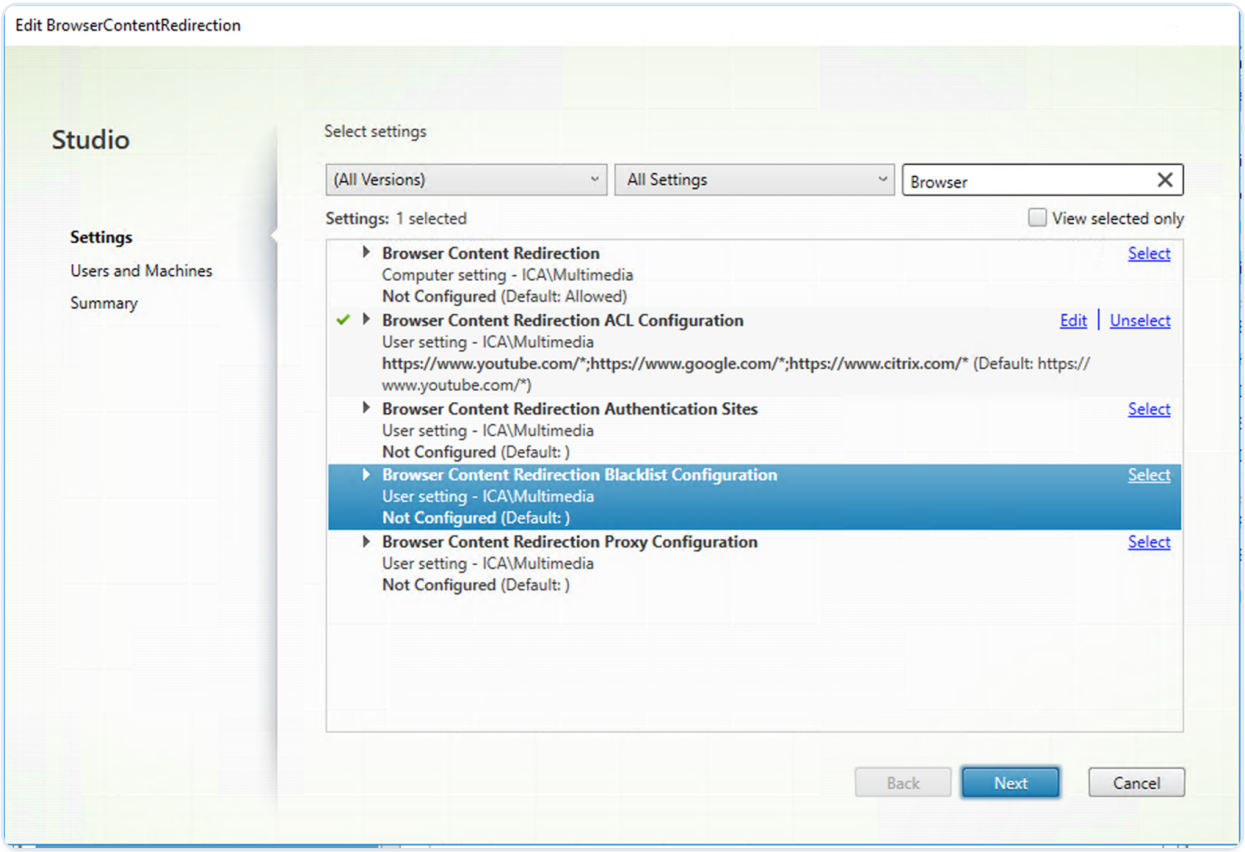


以下は許可される例です。

- `http://www.xyz.com/index.html` (URLでパスを指定することで、より詳細な粒度を実現できます。たとえば、`https://www.xyz.com/sports/index.html` を指定すると、`index.html` ページのみがリダイレクトされます。)
- `https://www.xyz.com/*`
- `http://www.xyz.com/*videos*`
- `http://*.xyz.com/`
- `http://*.*.com/`



Browser Content Redirection Blacklist Configuration 設定は、特定のURLのリダイレクトを無効にするブロックリストを指定します。



3. サーバーフェッチとクライアントレンダリングを有効にするには、 **Browser Content Redirection Proxy Configuration** 設定を構成します。

   この設定は、ブラウザコンテンツリダイレクトのためのVDA上のプロキシ設定の構成オプションを提供します。有効なプロキシアドレスとポート番号、PAC/WPAD URL、またはDirect/Transparent設定で有効になっている場合、Citrix Workspace app は常に最初にサーバーフェッチとクライアントレンダリングを試行します。詳細については、「フォールバックメカニズム」を参照してください。

   無効になっているか、構成されていない場合、またはデフォルト値を使用している場合、Citrix Workspace app はクライアントフェッチとクライアントレンダリングを試行します。

   デフォルトでは、この設定は **Prohibited** です。

   明示的なプロキシの許可パターン:

   `http://<hostname/ip address\>:\<port\>`

   例:

   `http://proxy.example.citrix.com:80 http://10.10.10.10:8080`

   PAC/WPADファイルの許可パターン:

   `http://<hostname/ip address>:<port>/<path>/<Proxy.pac>`

   例: `http://wpad.myproxy.com:30/configuration/pac/Proxy.pac`

   `https://<hostname/ip address>:<port>/<path>/<wpad.dat>`

   例: `http://10.10.10.10/configuration/pac/wpad.dat`

   直接または透過プロキシの許可パターン:

   ポリシーテキストボックスに**DIRECT**と入力します。

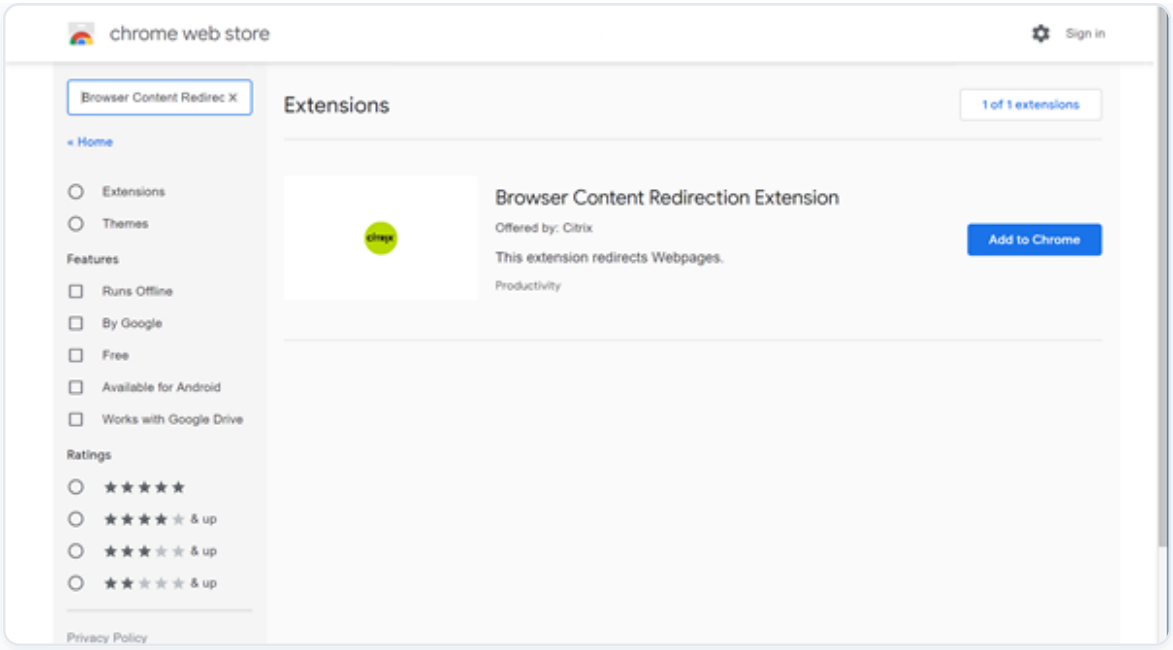
   > **注:**
   >
   > レジストリ値 `HKLM\Software\Citrix\HdxMediastream\WebBrowserRedirectionProxyAddress.` を編集してプロキシを設定することもできます。さらに、 `HKLM\Software\Citrix\HdxMediastream\AllowNonTlsPacUri` レジストリ値を使用

> すると、HTTP経由でのPACファイルのダウンロードを許可するかどうかを決定できます。デフォルト値は0で、HTTPは許可されません。

> レジストリはポリシー設定のオプションを上書きします。関連するレジストリキーのリストについては、「ブラウザコンテンツリダイレクトのレジストリキーのオーバーライド」を参照してください。

4. VDAで **Add to Chrome** をクリックして、Chrome Web Store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上のブラウザは、ナビゲートされている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できます。

> **重要:**
>
> 拡張機能はクライアントには必要ありません。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。



フォールバックメカニズム

ブラウザコンテンツリダイレクトプロキシ構成ポリシーを有効にすると、Citrix Workspaceアプリはサーバーフェッチとクライアントレンダリングを試行します。サーバーフェッチとクライアントレンダリングが失敗した場合、クライアントフェッチとクライアントレンダリングにフォールバックします。クライアントマシンがWebサーバーにアクセスできない場合、VDA上のブラウザはページを再読み込みし、サーバー上でレンダリングできます（サーバーフェッチとサーバーレンダリング）。

ブラウザコンテンツリダイレクトのレジストリキーのオーバーライド

> **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。

`HKLM\Software\Citrix\HdxMediastream`

名前	タイプ	値
WebBrowserRedirection	DWORD	1=許可、0=禁止
WebBrowserRedirectionAcl	REG_MULTI_SZ	/
WebBrowserRedirectionProxyAddress	REG_SZ	以下のいずれかのモードに設定すると、サーバーフェッチクライアントレンダリングが有効になります。**明示的プロキシ** - データセンターに単一の明示的プロキシがある場合。**直接または透過的** - プロキシがない場合、または透過的プロキシを使用する場合。**PACファイル** - PACファイルに依存して、VDA内のブラウザが指定されたURLを取得するための適切なプロキシサー

名前	タイプ	値
		バーを自動的に選択できるようにする場合。
WebBrowserRedirectionBlacklist	REG_MULTI_SZ	/
AllowNonTlsPacUri	DWORD	HTTP経由でのPACファイルのダウンロードを許可するかどうかを決定します。デフォルト値は0で、HTTPは許可されません。1に設定すると、**HDXWebProxy.exe**はHTTP経由でPACファイルをダウンロードできます（厳密にはHTTPS経由ではありません）。

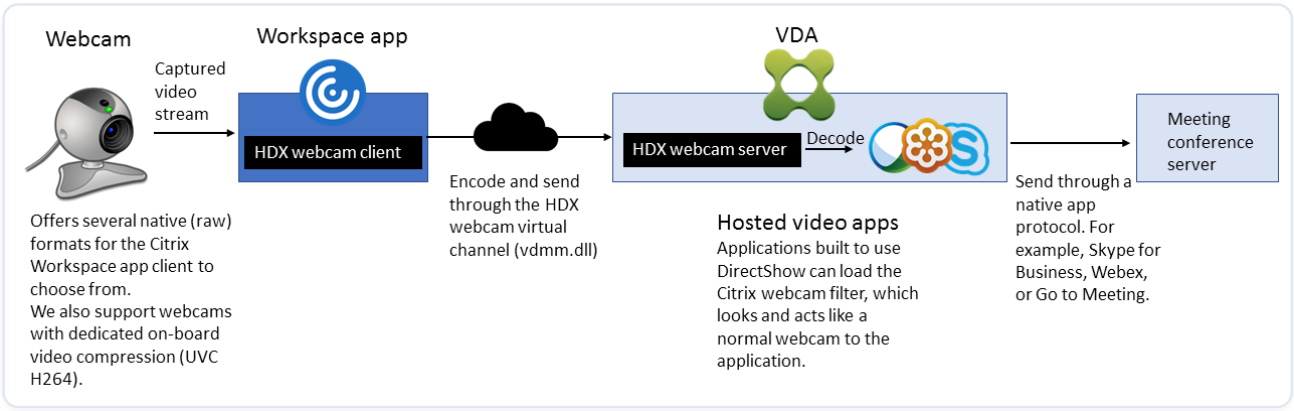
HDX™ ウェブカメラビデオ圧縮

概要

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDXウェブカメラビデオ圧縮を使用してウェブカメラを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ウェブカメラビデオ圧縮を使用することをお勧めします。

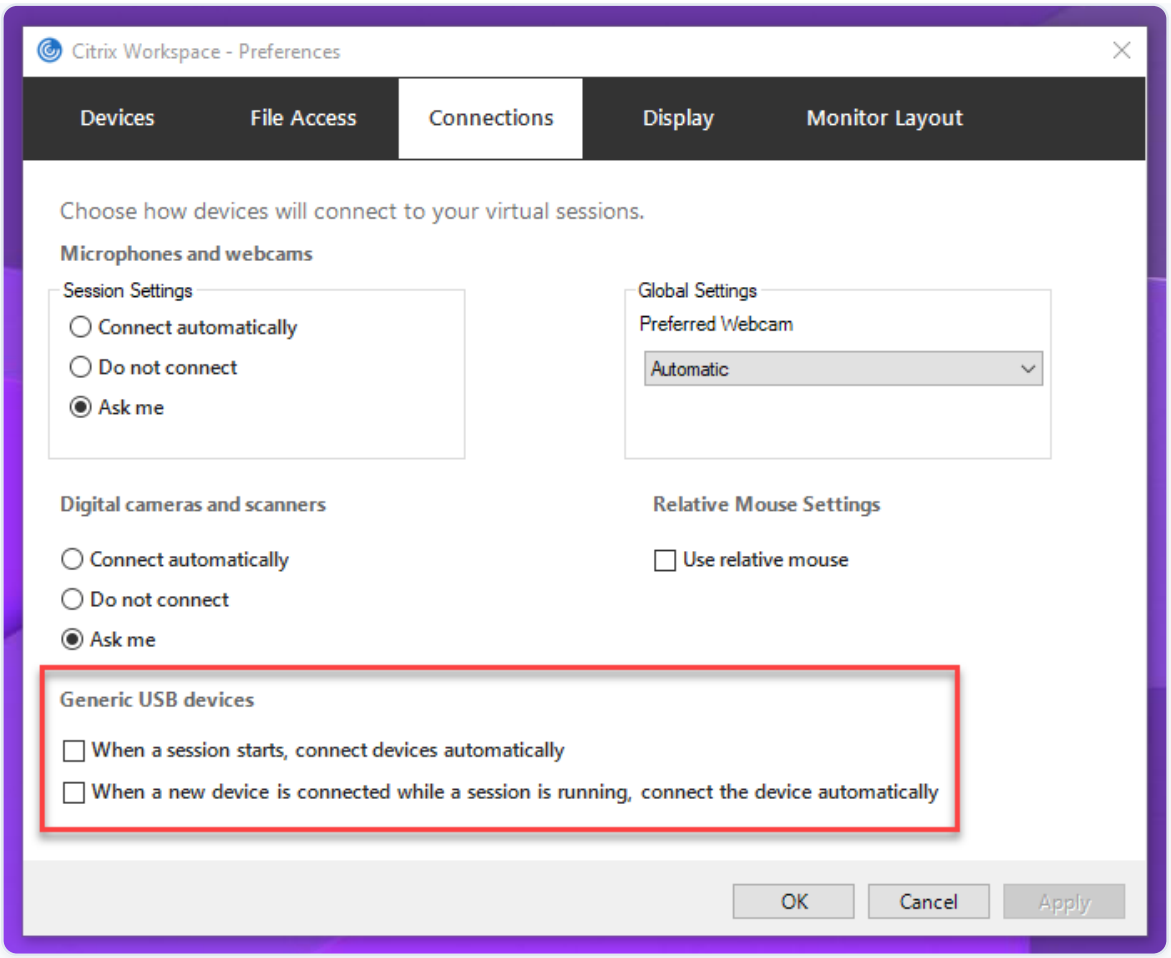
HDXウェブカメラビデオ圧縮は、**最適化された**ウェブカメラモードとも呼ばれます。このタイプのウェブカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDXウェブカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントはウェブカメラとの通信を処理します。その後、クライアントはビデオを適切に表示できるサーバーにのみ送信します。サーバーはウェブカメラを直接扱いませんが、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、Azureマシンでは利用できません。これは、この機能が依存する**videodev**カーネルモジュールがAzureマシンに存在しないためです。
- この機能は、Citrix WorkspaceアプリクライアントからのH.264ビデオのみをサポートします。
- サポートされているウェブカメラの解像度は、48x32から1920x1080の範囲です。
- ウェブカメラを使用しているときに、Citrix Workspace™アプリのツールバーから**Generic USB devices**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ウェブカメラ

ウェブカメラによってフレームレート、明るさ、コントラストのレベルが異なります。Citrix®は、初期機能検証のために以下のウェブカメラを使用しています。

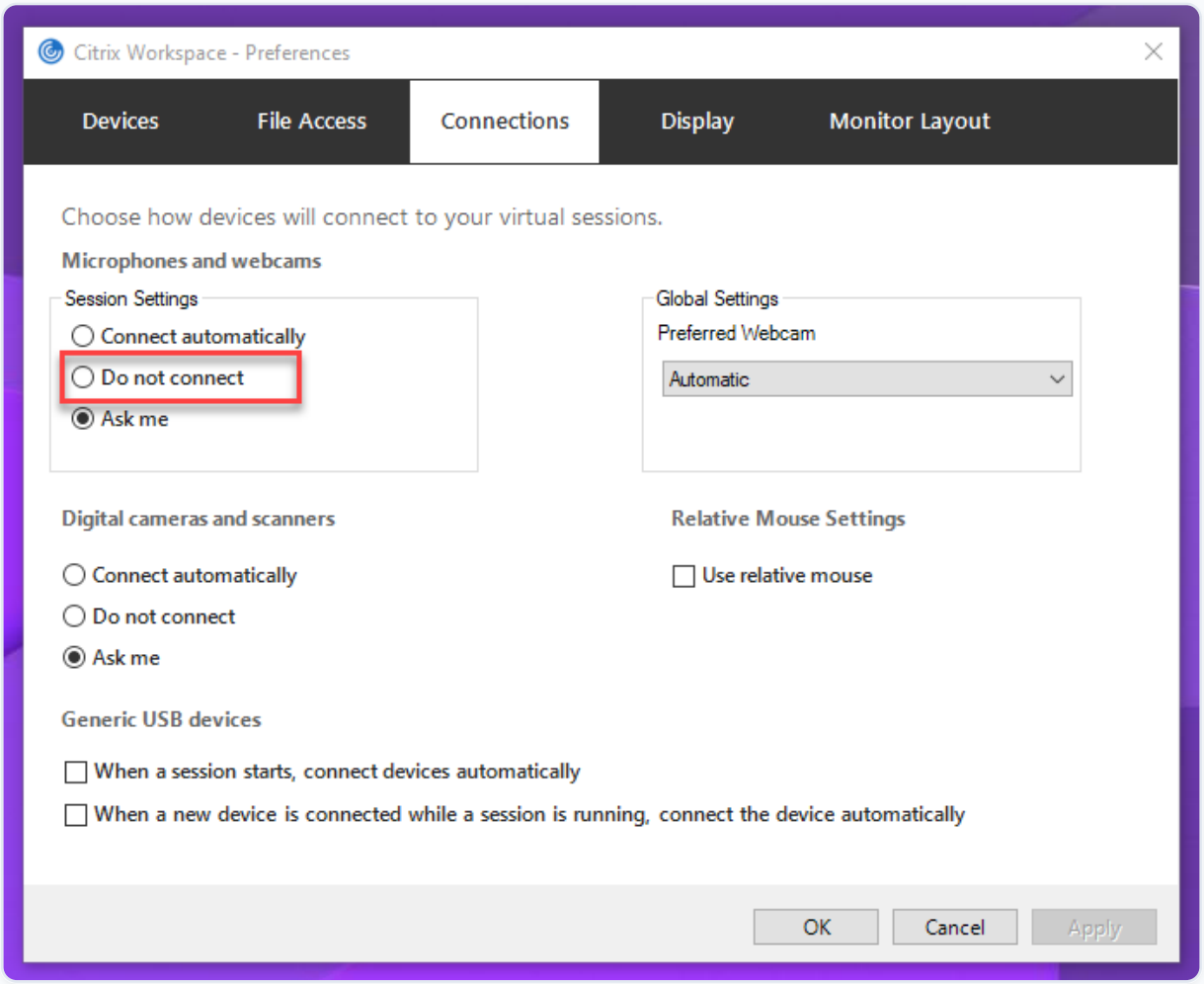
- ロジクール HD ウェブカメラ C270
- ロジクール ウェブカメラ C930e
- マイクロソフト ライフカム HD3000

構成

この機能はデフォルトで有効になっています。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**を選択することで、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カメラの設定を完了します。VDAがインターネットにアクセスできない場合は、手順3に進みます。

   > **注:**
   >
   > `uname -r`とカーネルヘッダーの間でカーネルの不一致が発生する可能性があります。この不一致により、ctxwcamcfg.shスクリプトが失敗します。HDX Webカメラビデオ圧縮を適切に使用するには、**sudo apt-get dist-upgrade**を実行し、VDAを再起動してから、ctxwcamcfg.shスクリプトを再実行します。

   VDAがDebianに展開されている場合は、最新のカーネルバージョンで実行されていることを確認してください。そうでない場合は、以下のコマンドを実行して最新のカーネルバージョンに更新してください。

   ```
   sudo apt-get update

   sudo apt-get dist-upgrade

   sudo reboot
   ```

   VDAがSUSE 15に展開されている場合は、以下のコマンドを実行して最新のカーネルバージョンに更新し、再起動してください。

   ```
   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Webカメラカーネルモジュールを構築するために使用されます。

- DKMSは、仮想Webカメラカーネルモジュールを動的に管理するために使用されます。

> **注:**
>
> RHEL、Rocky Linux、CentOSに上記のプログラムをインストールする場合、`ctxwcamcfg.sh`スクリプトはVDAに以下のリポジトリをインストールして有効にします。
>
> - 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 systemctl restart ctxwcamsd`コマンドを実行します。Linux VDAのWebカメラサービスである`ctxwcamsd`が再起動し、HDX Webカメラビデオ圧縮機能のために`v4l2loopback`カーネルモジュールをロードします。

3. VDAにインターネットアクセスがない場合は、別のマシンでv4l2loopbackカーネルモジュールをビルドし、それを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. そのマシンで、`sudo mkdir -p /var/xdl`コマンドを実行します。

c. VDAから`/var/xdl/configure_*`をコピーし、`/var/xdl/`の下にあるマシンにコピーします。

d. そのマシンで、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行してカーネルモジュールをビルドします。コマンドが正常に実行されると、`/var/lib/dkms/v4l2loopback/1.81b8df79107d1fbf392fdcbaa051bd227a9c94c1/$(uname -r)/x86_64/module/`パスの下にv4l2loopback.koファイルが作成されます。ctxwcamcfg.shスクリプトの実行時に発生する可能性のあるエラーは無視してください。

e. マシンからVDAにv4l2loopback.koをコピーし、`/opt/Citrix/VDA/lib64/`の下に配置します。

f. VDAで、`sudo systemctl restart ctxwcamsd`コマンドを実行してWebカメラサービスを再起動し、v4l2loopbackカーネルモジュールをロードします。

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と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を作成するときは、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。したがって、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はユーザーがCitrix Workspaceアプリにログオンするために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と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とセッション再接続は失敗します。SSOを無効にするには、「SSO以外の認証」を参照してください。

重要:

- Citrix DaaS™のお客様の場合：
  - 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。
  - ドメインに参加していないVDAを作成するには、Citrix Gatewayサービスを使用しているお客様は、Rendezvous V2が有効になっていることを確認する必要があります。Cloud Connectorは、オンプレミスのハイパーバイザーにマシンをプロビジョニングする場合、またはActive DirectoryをWorkspaceのIDプロバイダーとして使用する場合にのみ必要です。
- CVADのお客様の場合：
  - 以下の手順に従って、DDCでWebSocket機能を有効にします。`"HKLM:\SOFTWARE\Citrix\DesktopServer\WorkerProxy" -Name "WebSocket_Enabled" -PropertyType "DWord" -Value 1 -Force`

PowerShellを開いて次のコマンドを実行し、DDCを再起動します。New-ItemProperty

- ドメインに参加していないVDAを作成するには、MCSと簡易インストールの両方を使用できます。詳しくは、「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」および「簡易インストールを使用してドメインに参加していないLinux VDAを作成する」を参照してください。
- 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では、次の機能が利用できます。
  - 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
  - SSO以外の認証
  - Azure Active Directoryによる認証
  - ランデブー V2

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

属性	必須またはオプション	説明
		ザーに割り当てる番号です。これにより、ユーザー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で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。 ユーザーが別の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」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/authentication/non-sso-authentication.html)を参照してください。

Azure Active Directoryによる認証

Azureに展開するドメイン参加していないVDAは、AAD IDサービスと統合してユーザー認証を提供します。詳細については、「Azure Active Directoryによる認証」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/authentication/aad-authentication.html)を参照してください。

ランデブー V2

ドメイン参加していないVDAは、Citrix Cloud Connectorをバイパスするために Rendezvous V2の使用がサポートされています。詳細については、「Rendezvous V2」(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/session/rendezvous-v2.html)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**
ICAラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICAラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15
アイドル接続のICAラウンドトリップ計算	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザー監視	無効 (0)
セッション全体の帯域幅制限	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限（パーセント）	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクトの帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントUSBデバイスリダイレクト帯域幅パーセント	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
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パーセント	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限パーセント	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0
WebSockets接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocketsポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008
WebSockets信頼済みオリジンサーバーリスト	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICA®キープアライブ	SendICAKeepAlives	コンピューター	**ICAキープアライブ**	ICAキープアライブメッセージを送信しない (0)
ICAキープアライブタイムアウト	ICAKeepAliveTimeout	コンピューター	**ICAキープアライブ**	60秒

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
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	推奨(2)
Rendezvousプロトコル	RendezvousProtocol	コンピューター	ICA	禁止
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 秒
クライアントの自動再接続	AllowAutoClientReconnect	ユーザー	ICA\クライアントの自動再接続	許可(1)
クライアント自動再接続認証	AllowAutoClientReconnectAuthentication	コンピューター	ICA\自動クライアント再接続	無効 (0)
クライアント自動再接続ログ	AllowAutoClientReconnectLogging	コンピューター	ICA\自動クライアント再接続	無効 (0)
クライアント自動再接続タイムアウト	AllowAutoClientReconnectTimeout	コンピューター	ICA\自動クライアント再接続	120秒

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDFユニバーサルプリンターの自動作成	AutoCreatePDFPrinter	ユーザー	印刷	無効 (0)
プリンタードライバーのマッピングと互換性	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)
クリップボードのクライアントからセッションへの転送サイズを制限する	LimitClipboardTransferC2H	ユーザー	ICA	無効 (0)
クリップボードセッションのクライアント転送サイズを制限	LimitClipboardTransferH2C	ユーザー	ICA	無効 (0)
クリップボードリダイレクトの帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0

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の帯域幅制限の割合	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0
クライアントのクリップボード書き込みを制限	RestrictClientClipboardWrite	ユーザー	クリップボード	無効 (0)
クライアントのクリップボード書き込み許可形式	ClientClipboardWriteAllowedFormats	ユーザー	クリップボード	該当なし
セッションクリップボードへの書き込みを制限する	RestrictSessionClipboardWrite	ユーザー	クリップボード	無効 (0)
セッションクリップボードへの書き込み許可形式	SessionClipboardWriteAllowedFormats	ユーザー	クリップボード	該当なし
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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ターゲット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
ターゲットフレームレート	FramesPerSecond	ユーザー	シンワイヤー	30 フレーム/秒
視覚的品質	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	VideoCodec	ユーザー	シンワイヤー	優先する場合に使用 (3)
ビデオコーデックにハードウェアエンコードを使用	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する	OptimizeFor3dWorkload	ユーザー	シンワイヤー	無効 (0)
シンプルな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)
グラフィックスステータスインジケーター	DisplayLosslessIndicator	ユーザー	シンワイヤー	無効 (0)
オーディオ品質	SoundQuality	ユーザー	オーディオ	高 – 高精細オーディオ (2)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントマイクのリダイレクト	AllowMicrophoneRedir	ユーザー	オーディオ	許可 (1)
最大セッション数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオン許容数	ConcurrentLogonsTolerance	コンピューター	負荷管理	2
Controller の自動更新を有効にする	EnableAutoUpdateOfControllers	コンピューター	バーチャル デリバリー エージェント設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3
最大Speex品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブの自動接続	AutoConnectDrives	ユーザー	ファイルリダイレクト\CDM	有効 (1)
クライアント光学ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアント固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアントフロッピーディスクドライブ	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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントリムーバブルドライブ	AllowRemoveableDrives	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する	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロードする	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロードする	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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
切断されたセッションのタイマー間隔	SessionDisconnectTimerPeriod	ユーザー	セッションタイマー	1440分
ブラウザコンテンツリダイレクト	WebBrowserRedirection	コンピューター	ICA\マルチメディア	許可
ブラウザコンテンツリダイレクトACL構成	WebBrowserRedirectionAcl	ユーザー	ICA\マルチメディア	`https://www.youtube.com/*`
ブラウザコンテンツリダイレクト認証サイト	WebBrowserRedirectionAuthenticationSites	ユーザー	ICA\マルチメディア	ヌル
ブラウザコンテンツリダイレクトブラックリスト構成	WebBrowserRedirectionBlacklist	ユーザー	ICA\マルチメディア	なし
ブラウザコンテンツリダイレクトプロキシ構成	WebBrowserRedirectionProxy	ユーザー	ICA\マルチメディア	なし
セッションウォーターマークを有効にする	EnableSessionWatermark	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	無効
クライアントIPアドレスを含める	IncludeClientIPAddress	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	無効
接続時間を含める	IncludeConnectTime	ユーザー	ICA\セッションウォーターマーク\ウォーター	無効

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
			マークコンテンツ	
ログオンユーザー名を含める	IncludeLogonUsername	ユーザー	ICA\セッション透かし\透かしコンテンツ	有効
VDAホスト名を含める	IncludeVDAHostname	ユーザー	ICA\セッション透かし\透かしコンテンツ	有効
VDA IPアドレスを含める	IncludeVDAIPAddress	ユーザー	ICA\セッション透かし\透かしコンテンツ	無効
セッションウォーターマークのスタイル	WatermarkStyle	ユーザー	ICA\セッション透かし\透かしスタイル	複数
ウォーターマークの透明度	WatermarkTransparency	ユーザー	ICA\セッション透かし\透かしスタイル	17
透かしカスタムテキスト	WatermarkCustomText	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	有効なカスタムテキスト構成は、`<font=Arial Unicode MS><domain><newline><fontsize=47><username><newline><fontzoom=120>Citrix<newline><style=tile>Najing <clientip><newline><rotation=45><serverip><position=`

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
				`bottomleft>` のようになります。詳細については、「セッションの透かし」を参照してください。
セッションメトリックの収集	UsageDataCollectionThroughClient	コンピューター	ICA	有効 (1)
動的ウィンドウプレビュー	EnableDynamicPreview	コンピューター	ICA\グラフィック	この設定は、シームレスウィンドウの表示を有効または無効にします。この設定を有効にすると、公開アプリケーションのウィンドウで画面上のコンテンツをプレビューできます。たとえば、Windows 用 Citrix Workspace アプリを使用して Google Chrome セッションの複数のウィンドウを開き、タスクバーの Chrome アイコンにカーソルを合わせると、コンテンツをプレビューできます。詳細については、「動的ウィンドウプ

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
				レビュー」を参照してください。デフォルトでは、この設定は**有効 (1)**です。

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) リアルタイムトランスポート経由のオーディオ」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、次の 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**に設定されており、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。
- **切断されたセッションタイマー**：切断されたセッションに時間制限を適用するかどうかを決定します。
- **切断されたセッションタイマー間隔**：切断されたセッションがログオフされるまでの間隔を設定します。

いずれかのポリシー設定を更新する際は、展開全体でそれらが一貫していることを確認してください。

アイドルセッションの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/2607-ltsr/media/warning-message-idle-session-timeout.png)

以下のポリシーは、Citrix Studioバージョン7.12以降で構成できます。

- MaxSpeexQuality

  値（整数）：

  デフォルト値：5

  詳細：

  オーディオリダイレクトは、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）、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の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、モードによってビットレートが異なります。最適なSpeex品質とは、最大値が以下の条件を満たす場合です。

  - 最大Speex品質以下であること。
  - そのビットレートが帯域幅制限以下であること。

  関連設定：オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

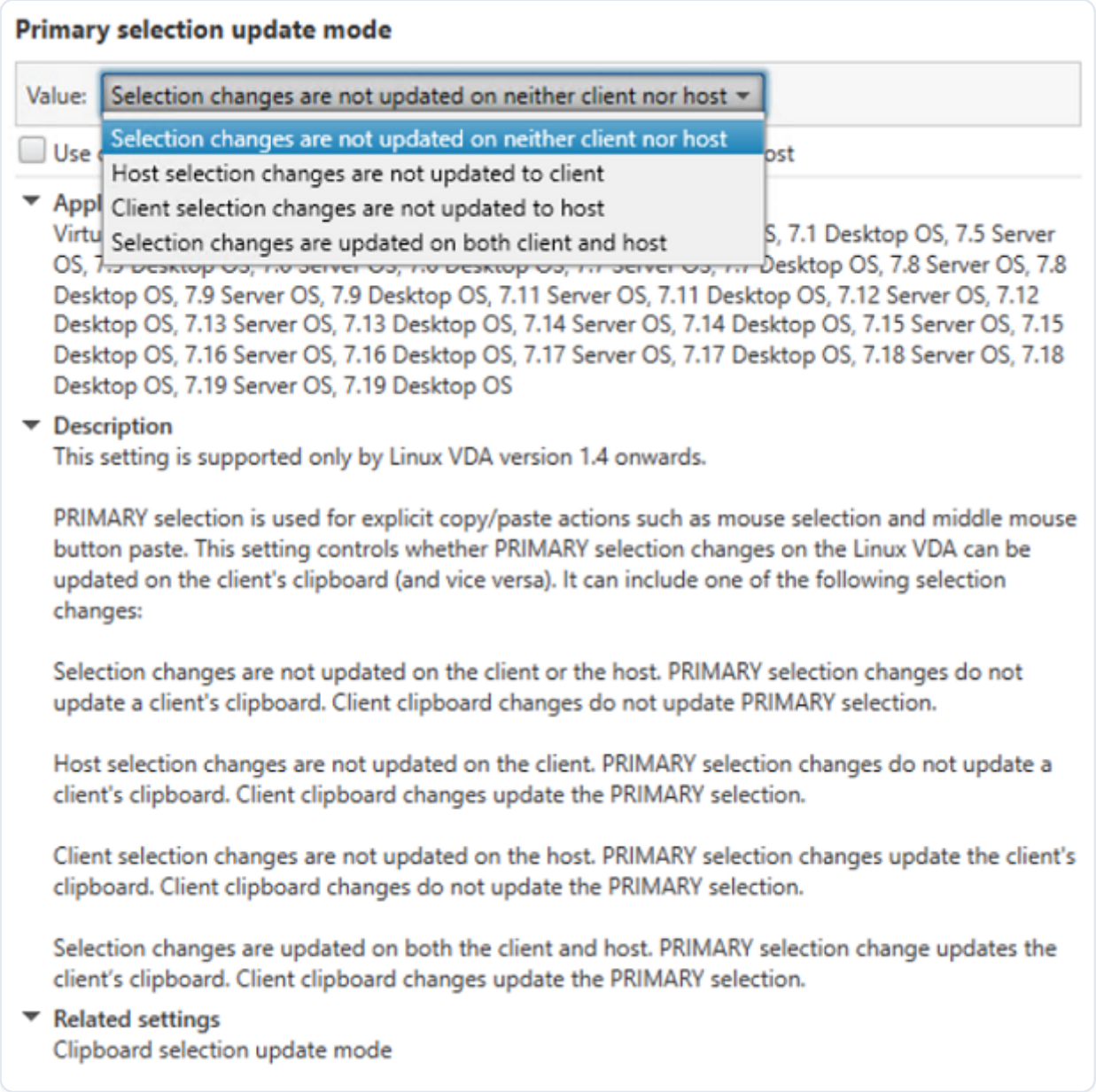
  値 (列挙型):

  デフォルト値: 3

  詳細:

  プライマリ選択は、データを選択し、マウスの中央ボタンを押して貼り付けるときに使用されます。

このポリシーは、Linux VDAとクライアント間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。値のオプションは4つあります。



- **クライアントとホストのどちらに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

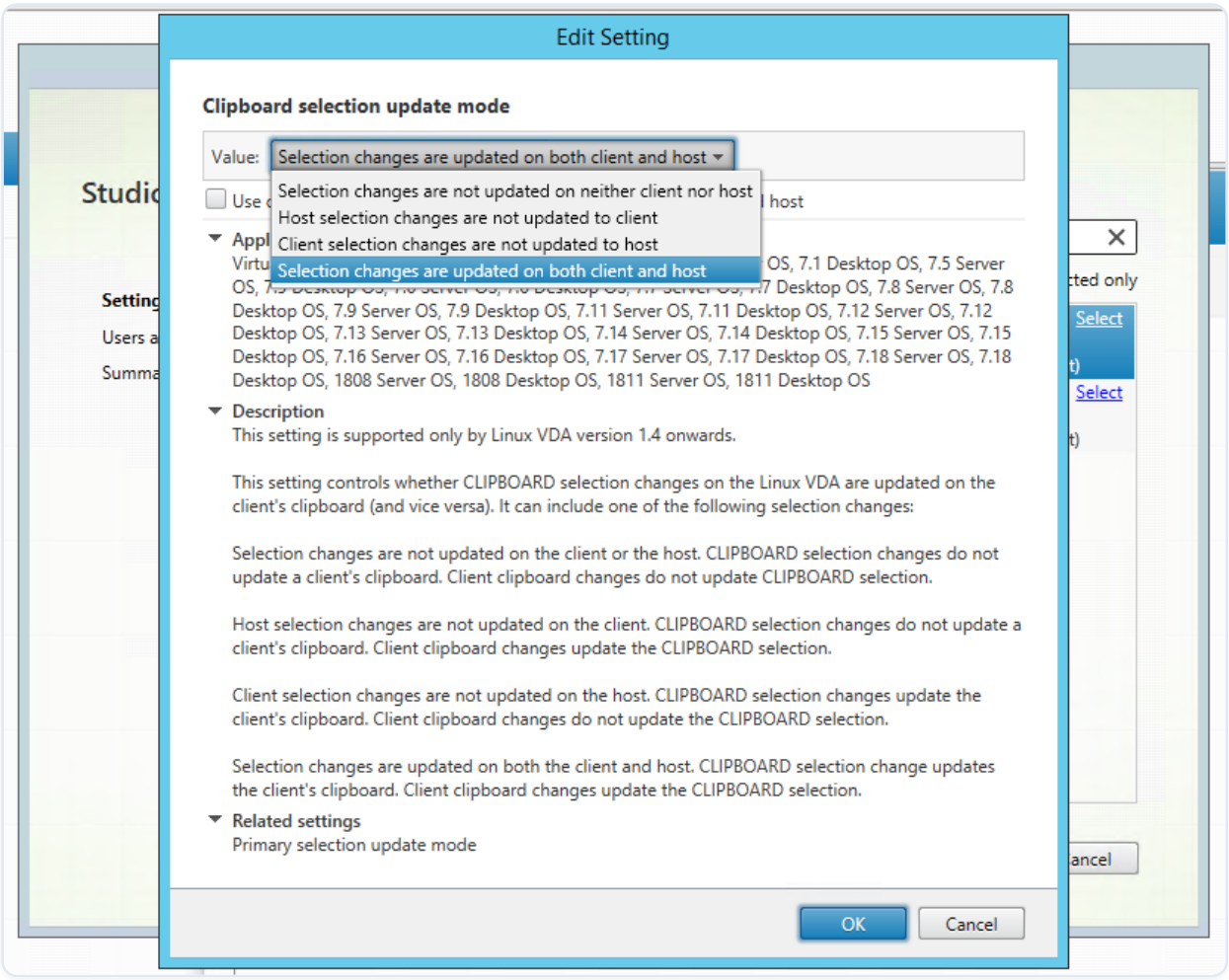
値 (列挙型):

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、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付けられていますが、プライマリ選択はLinux固有のものです。

このポリシーは、Linux VDAとクライアント間のクリップボード選択の変更が相互に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にも選択の変更は更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **クライアントとホストの両方で選択の変更が更新される**

Linux VDAでのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

> **注:**
>
> Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピーアンドペーストの動作を制御するには、クリップボード選択更新モードとプライマリ選択更新モードの両方を同じ値に設定することをお勧めします。

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

印刷のベストプラクティス

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では、**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。

印刷ポリシー設定

クライアントプリンターリダイレクト

この設定により、クライアントプリンターをVDAセッションにマップするかどうかを決定できます。デフォルトでは、クライアントプリンターのマッピングは許可されています。

クライアントプリンターを自動作成

この設定では、VDAセッションにマップできるクライアントプリンターを指定します。デフォルトでは、**すべてのクライアントプリンターを自動作成**に設定されており、これはすべてのクライアントプリンターがVDAセッションにマップされることを意味します。この設定の詳細について

は、Citrix Virtual Apps and Desktopsのドキュメントでクライアントプリンターの自動作成を参照してください。

PDFユニバーサルプリンターの自動作成

PDF印刷機能を使用するには、このポリシーを**有効**に設定します。

プリンタードライバーのマッピングと互換性

Citrixは、3種類のユニバーサルプリンタードライバー（postscript、pcl5、pcl6）を提供しています。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合があります。この場合、以前のリリースでは、~/.CtxlpProfile$CLIENT_NAME構成ファイルを編集するしかありませんでした。バージョン1906以降では、代わりにCitrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成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. 必要に応じて、次のアクションを実行します。

    - クライアントプリンターをVDAセッションにリダイレクトし、ユニバーサル印刷ドライバーのみを使用できるようにするには、**許可**または**ユニバーサルドライバーのみで作成**を選択します。
    - 指定されたドライバー名のクライアントプリンターがVDAセッションにリダイレクトされないようにするには、**作成しない**を選択します。
    - サーバーと同じドライバーを持つがドライバー名が異なるクライアントプリンターにサーバーアプリケーションがアクセスできるようにするには、**Replace with** 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(/en-us/linux-virtual-delivery-agent/2607-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。

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。すべてのクライアントプリンターをVDAセッションにマップできます。デスクトップとアプリケーションではプリンター名が異なります。

- 公開デスクトップの場合:

  `<client printer name>:$CLIENT_NAME:dsk$SESSION_ID`

- 公開アプリケーションの場合:

  `<client printer name>:$CLIENT_NAME:app$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です。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ことは失敗します。

トラブルシューティング

印刷できません

印刷が正しく機能しない場合は、印刷デーモン **ctxlpmngt** と **CUPs framework** を確認してください。

印刷デーモン **ctxlpmngt** はセッションごとのプロセスであり、セッションの期間中実行されている必要があります。印刷デーモンが実行されていることを確認するには、次のコマンドを実行します。**ctxlpmngt** が実行されていない場合は、コマンドラインから手動で **ctxlpmngt** を開始してください。

```
ps –ef | grep ctxlpmngt
```

印刷がまだ機能しない場合は、**CUPS** フレームワークを確認してください。**ctxcups** サービスはプリンター管理に使用され、Linux CUPSフレームワークと通信します。これはマシンごとに1つのプロセスであり、次のコマンドを実行して確認できます。

```
systemctl status ctxcups
```

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** に記録されません。

```
sudo systemctl stop cups



sudo vi /etc/systemd/system/printer.target.wants/cups.service



PrivateTmp=false



sudo systemctl start cups



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**、**drivertype** の各フィールドは同時に設定できません。これは、**マップされたプリンター**に対して1つしか有効にならないためです。

- 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、**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。**lpinfo** コマンドを使用して、プリンターの現在のモデル名を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、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。

  たとえば、/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種類あります (postscript、pcl5、および pcl6)。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構成できます。

  たとえば、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

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、次の手順を実行して問題を特定できます。

1. 公開デスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME** コマンドを実行します。
3. Linux VDAにスプールファイルを保存するために、次のフィールドを追加します。

   ```
   deletespoolfile=no
   ```

4. 構成の変更を読み込むには、ログオフしてから再度ログオンします。
5. 問題を再現するためにドキュメントを印刷します。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よって異なります。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。内容が正しいこと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. 別のプリンタードライバーを使用するように**マップされたプリンター**を変更します。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ています。

   a. アクティブなセッションにログオンし、クライアントデスクトップでブラウザーを開きます。

   b. 印刷管理ポータルを開きます。

   ```
   localhost:631
   ```

   c. **マップされたプリンター** `CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID`を選択し、**プリンターの変更**を選択します。この操作には管理者権限が必要です。

   d. **cups-ctx**接続を保持し、**続行**をクリックしてプリンタードライバーを変更します。

   e. **製造元**および**モデル**フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。

   f. このプロセスが成功した場合、`.CtxlpProfile$CLIENT_NAME`でドライバーのPPDファイルパスを構成して、マップされたプリンターが新しく選択されたドライバーを使用できるようにします。

既知の問題

Linux VDAでの印刷中に、次の問題が特定されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブドライバーに設定してください。

大容量ドキュメントの印刷パフォーマンスの低下

ローカルクライアントプリンターで大容量ドキュメントを印刷すると、ドキュメントはサーバー接続を介して転送されます。低速な接続では、転送に時間がかかる場合があります。

他のセッションから表示され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設定ファイルである **/etc/cups/cupsd.conf** を変更することで無効にできます。

ファイルに設定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* の場合は、以下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 のバージョン2.4～2.6.9、および Citrix Workspace app 1808 for HTML5 以降
- Citrix Receiver for Chrome のバージョン2.4～2.6.9、および Citrix Workspace app 1808 for Chrome 以降
- Citrix Workspaceアプリ Windows版1905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設定します。

- **Client Printer Redirection**を**Allowed**に設定します（デフォルトで**Allowed**）
- **Auto-create PDF Universal Printer**を**Enabled**に設定します（デフォルトで**Disabled**）
- 「**クライアントプリンターの自動作成**」を「**すべてのクライアントプリンターを自動作成**」に設定します。

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**Print**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アクセスすることを容易に許可できます。ユーザーがオフィスPCにアクセスできれば、業務に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

Remote PC Accessは、仮想デスクトップとアプリケーションを提供するCitrix Virtual Apps™ and Desktopsと同じコンポーネントを使用します。Remote PC Accessの展開と構成の要件とプロセスは、Citrix Virtual Apps and Desktopsの展開に必要な要件とプロセスと同じです。この一貫性により、一貫した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を非3Dモードでのみ使用してください。NVIDIAのドライバーの制限により、HDX™ 3Dモードが有効になっている場合、PCのローカル画面をブラックアウトできません。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、管理者の介入なしに行われます。クライアント上のCitrix Workspace™ appは、Remote PC Accessデスクトップセッション内で、ユーザーがオフィス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

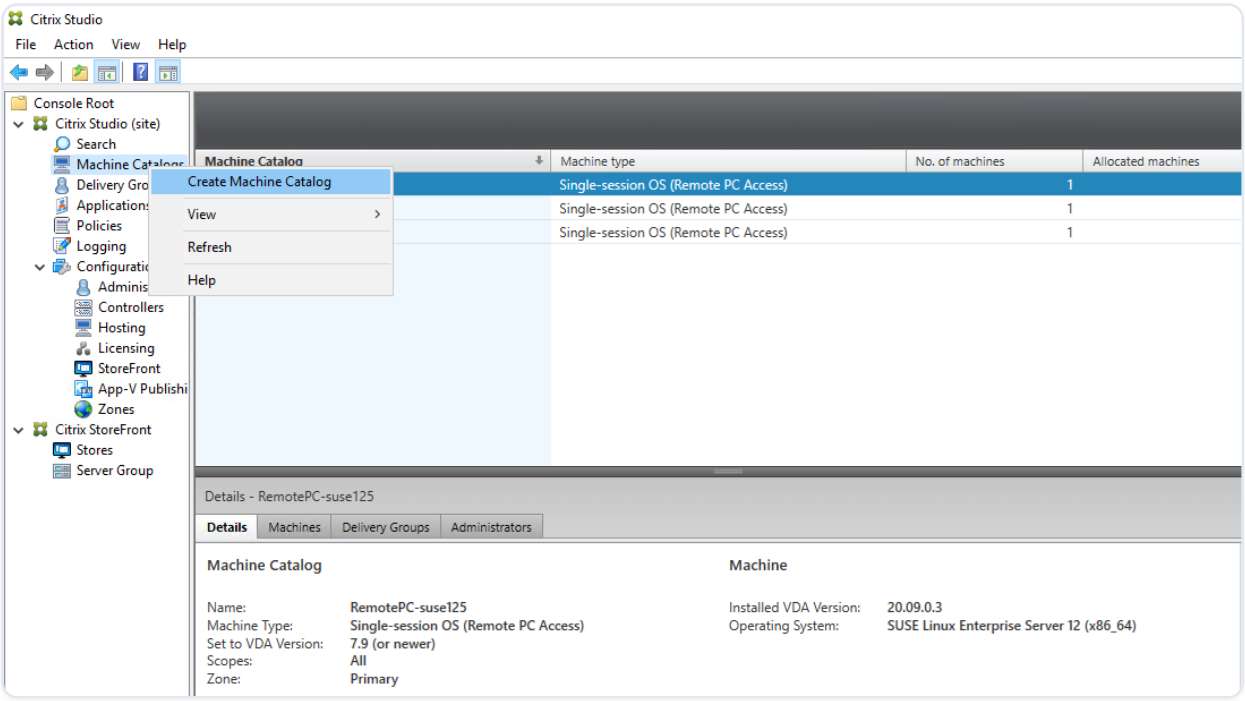
Linux PCセッションを配信するには、ターゲットPCにLinux VDAをインストールし、**Remote PC Access**タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成します。以下のセクションで手順を詳しく説明します。

手順1 - ターゲットPCにLinux VDAをインストールする

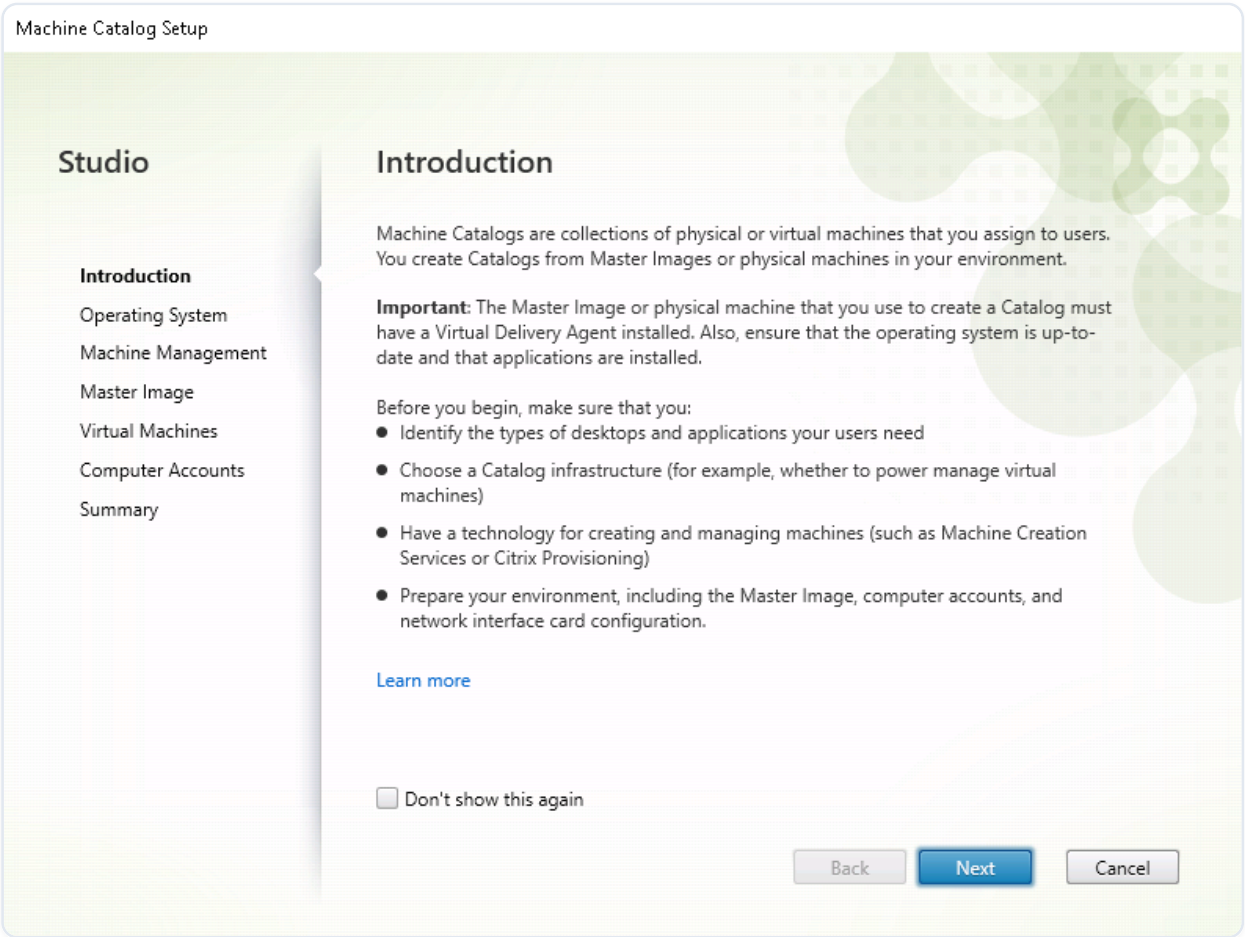
Linux VDAのインストールには、easy installを使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE`変数の値を`Y`に設定します。

手順2 - Remote PC Accessタイプのマシンカタログを作成する

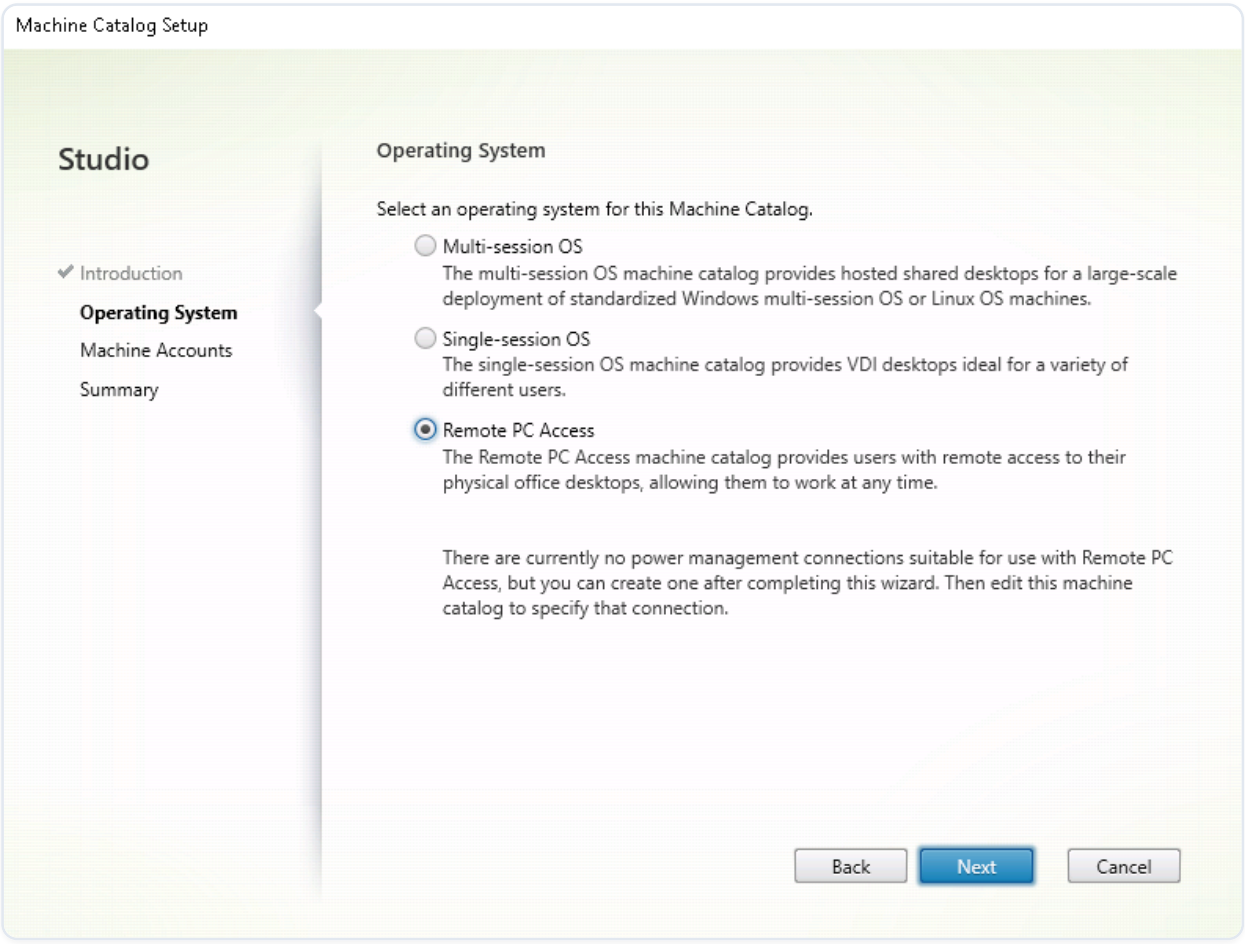
1. Citrix Studioで、**Machine Catalogs**を右クリックし、ショートカットメニューから**Create Machine Catalog**を選択します。



2. **Introduction**ページで**Next**をクリックします。



3. **Operating System**ページで**Remote PC Access**を選択します。



4. ターゲットPCを含むOUを選択するには「**OUの追加**」をクリックするか、個々のマシンをマシンカタログに追加するには「**マシンアカウントの追加**」をクリックします。

   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/2607-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/2607-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/2607-ltsr/media/remote-pc-access-6-edit-a-machine-catalog.png)

ステップ3 - アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成する

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/2607-ltsr/media/remote-pc-access-11-create-a-delivery-group.png)

2. 「**デリバリーグループの開始**」ページで「**次へ**」をクリックします。

   「デリバリーグループの開始」ページのイメージ(/en-us/linux-virtual-delivery-agent/2607-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/2607-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/2607-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™ (旧称 Citrix バーチャル アプリケーションズ アンド デスクトップス サービス)
  - Citrix Virtual Apps and Desktops 2012 以降のバージョン

- 物理PC:
  - VDA バージョン 2012 以降
  - BIOS および NIC で Wake on LAN が有効になっていること

Wake on LANの構成

現在、統合されたWake on LANの構成はPowerShellを使用してのみサポートされています。

Wake on LANを構成するには:

1. まだ Remote PC Access マシンカタログがない場合は、作成します。
2. まだ Wake on LAN ホスト接続がない場合は、作成します。

> **注:**
>
> Wake on LAN機能を使用するには、「Microsoft Configuration Manager Wake on LAN」タイプのホスト接続がある場合、ホスト接続を作成します。

3. Wake on LAN ホスト接続の一意の識別子を取得します。
4. Wake on LAN ホスト接続をマシンカタログに関連付けます。

   Wake on LAN ホスト接続を作成するには：

```
# Load Citrix SnapIns

Add-PSSnapIn -Name "*citrix*"



# Provide the name of the Wake on LAN host connection

[string]$connectionName = "Remote PC Access Wake on LAN"



# Create the hypervisor connection

$hypHc = New-Item -Path xdhyp:\Connections `

             -Name $connectionName `
```

```
            -HypervisorAddress "N/A" `

            -UserName "woluser" `

            -Password "wolpwd" `

            -ConnectionType Custom `

            -PluginId VdaWOLMachineManagerFactory `

            -CustomProperties "<CustomProperties></CustomProperties>" `

            -Persist



$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid



# Wait for the connection to be ready before trying to use it

while (-not $bhc.IsReady)

{

    Start-Sleep -s 5

    $bhc = Get-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid

}
```

ホスト接続の準備ができたら、次のコマンドを実行してホスト接続の一意の識別子を取得します。

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"

$hypUid = $bhc.Uid
```

接続の一意の識別子を取得したら、次のコマンドを実行して、その接続を Remote PC Access マシンカタログに関連付けます。

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog
-RemotePCHypervisorConnectionUid $hypUid
```

5. マシンカタログ内の各VMのBIOSおよびNICでWake on LANを有効にします。

   注: Wake on LANを有効にする方法は、マシンの構成によって異なります。

   - BIOSでWake on LANを有効にするには：

     a. BIOSに入り、Wake on LAN機能を有効にします。

        BIOSにアクセスする方法は、マザーボードの製造元と、製造元が選択したBIOSベンダーによって異なります。

     b. 設定を保存し、マシンを再起動します。

   - NICでWake on LANを有効にするには：

     a. NICがマジックパケットをサポートしているかどうかを確認するには、`sudo ethtool <NIC>` コマンドを実行します。

        `<NIC>` はNICのデバイス名です（例：`eth0`）。`sudo ethtool <NIC>` コマンドは、NICの機能に関する出力を提供します。

        - 出力に`Supports Wake-on: <letters>`のような行があり、その`<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`<letters>`に文字`g`が含まれ、かつ文字`d`が含まれていない場合、Wake on LANマジックパケット方式が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を有効にしてください。

     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以下の手順を実行してください。

        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` の行を追加します。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

- 一時ホームディレクトリでのログオン
- アプリケーションの公開
- セッションの信頼性
- 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
- DTLS を使用したユーザーセッションの保護

アダプティブトランスポート

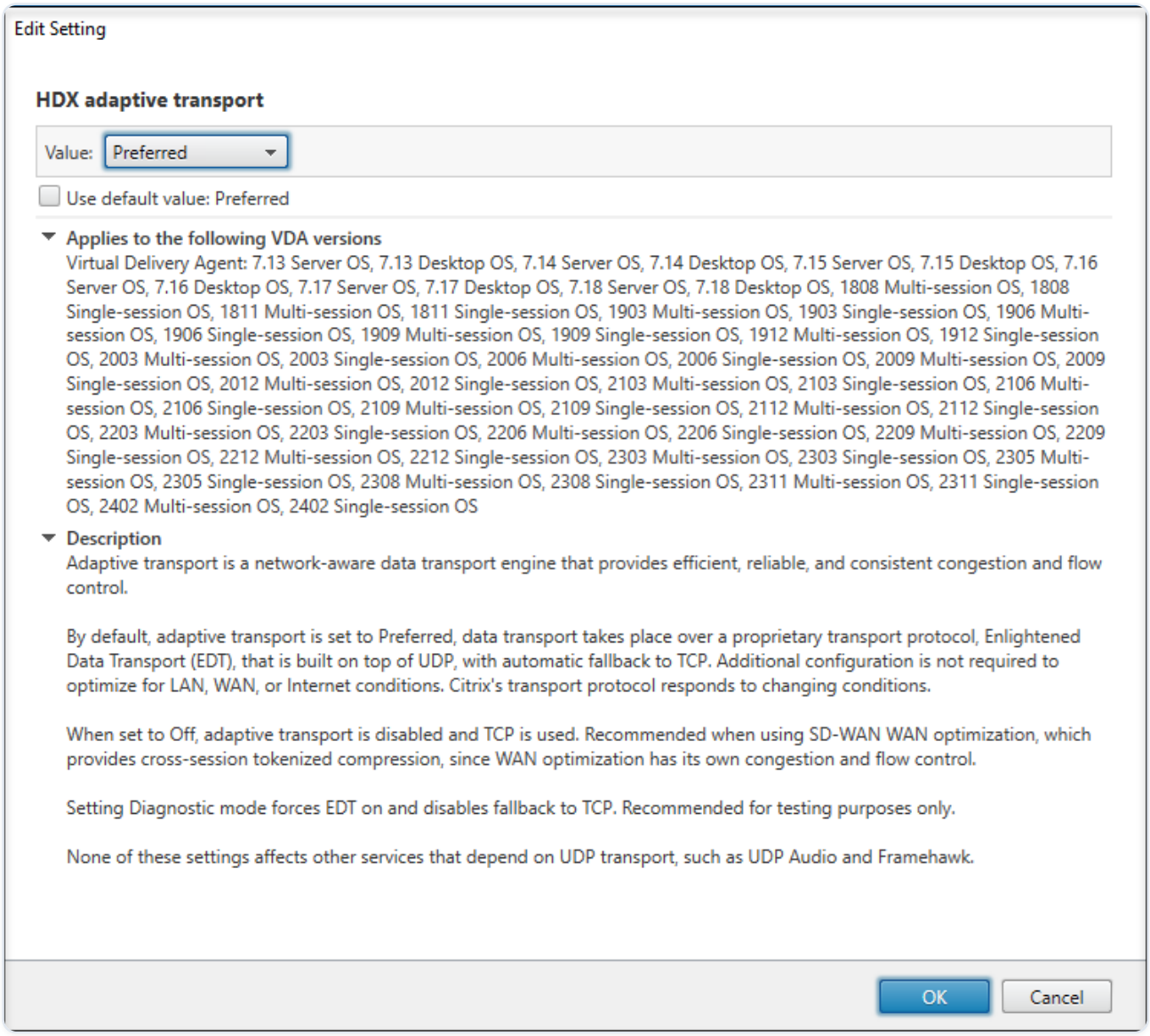
アダプティブトランスポートは、Citrix Virtual Apps and Desktops™のメカニズムであり、ICA接続のトランスポートプロトコルとしてEnlightened Data Transport (EDT)を使用できます。EDTが利用できない場合、アダプティブトランスポートはTCPに切り替わります。

EDTは、User Datagram Protocol (UDP)上に構築されたCitrix独自のトランスポートプロトコルです。サーバーのスケーラビリティを維持しながら、困難な長距離接続において優れたユーザーエクスペリエンスを提供します。EDTは、信頼性の低いネットワーク上のすべてのICA®仮想チャネルのデータスループットを向上させ、より良く、より一貫したユーザーエクスペリエンスを提供します。

詳細については、Citrix Virtual Apps and Desktopsのドキュメントの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または無効にする

アダプティブトランスポートはデフォルトで有効になっています。**HDX™アダプティブトランスポート**ポリシー設定を使用して、次のオプションを構成できます。



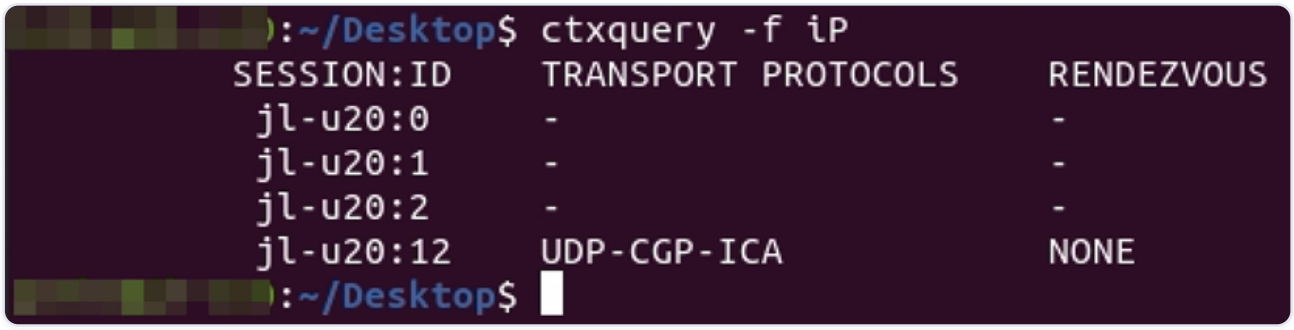
- **優先**: アダプティブトランスポートが有効になり、Enlightened Data Transport (EDT)を優先トランスポートプロトコルとして使用し、TCPへのフォールバックを伴います。
- **診断モード**: アダプティブトランスポートが有効になり、EDTの使用を強制します。TCPへのフォールバックは無効になります。この設定は、テストおよびトラブルシューティングのみに推奨されます。
- **オフ**: アダプティブトランスポートは無効になり、トランスポートにはTCPのみが使用されます。

アダプティブトランスポートが使用されているかどうかを確認する

現在のセッションでEDTがトランスポートプロトコルとして使用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。

```
/opt/Citrix/VDA/bin/ctxquery -f iP
```

EDTが使用されている場合、表示されるトランスポートプロトコルにはUDPが含まれます。例:



EDT MTU検出

MTU検出により、EDTはセッション確立時に最大転送単位（MTU）を自動的に決定できます。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。

システム要件:

- Linux VDA の最小バージョン 2012
- Citrix Workspace™ アプリ:
  - Windows: 1911 以降
- シトリックス ADC:
  - 13.0.52.24 以降
  - 12.1.56.22 以降
- 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の構成の詳細について、「CTX231821」を参照してください。

VDA での EDT MTU 検出の制御

EDT MTU 検出は、VDA でデフォルトで有効になっています。`MtuDiscovery` レジストリキーを次のように設定することで、有効または無効にできます。

- 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 が登録されるまで待ちます。

```
/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。

> **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発生した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**は自己責任で使用してください。編集する前に必ずレジストリをバックアップしてください。

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` セクションで次のように設定します。

`MtuDiscovery=Off`

この機能を再度有効にするには、ICA ファイルから `MtuDiscovery` パラメーターを削除します。

> **重要:**
>
> この ICA ファイルパラメーターが機能するには、VDA で EDT MTU 検出を有効にする必要があります。VDA で EDT MTU 検出が有効になっていない場合、ICA ファイルパラメーターは効果がありません。

EDT 輻輳制御の強化

EDTプロトコルを最適化するために、輻輳制御アルゴリズムが導入されています。この実装により、EDTはより高いスループットを実現し、レイテンシーを削減して、ユーザーエクスペリエンスを向上させることができます。

この機能はデフォルトで有効になっています。この機能を無効化および再有効化するには、それぞれ次のコマンドを実行し、その後`ctxhdx`サービスを再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal
Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d
"0x00000000" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal
Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d
"0x00000001" --force
```

HDX™アダプティブスループット

HDXアダプティブスループットは、出力バッファを調整することで、ICA®セッションのピークスループットをインテリジェントに微調整します。出力バッファの数は、初期値として高い値に設定されます。この高い値により、特に高遅延ネットワークにおいて、データがより迅速かつ効率的にクライアントに送信されるようになります。この機能は、ユーザーエクスペリエンスを向上させます。これにより、インタラクティブ性の向上、ファイル転送の高速化、ビデオ再生のスムーズ化、フレームレートと解像度の向上が実現します。セッションのインタラクティブ性は、ICAセッション内のデータストリームがインタラクティブ性に悪影響を与えていないかを判断するために常に測定されます。そのような事態が発生した場合、大規模なデータストリームがセッションに与える影響を軽減し、インタラクティブ性を回復させるために、スループットが低下されます。

重要:

HDXアダプティブスループットは、メカニズムをクライアントからVDAに移動することで、出力バッファの設定方法を変更します。手動での設定は不要です。

この機能には、VDAの最小バージョン2311が必要です。デフォルトで有効になっています。無効になっている場合は、VDAで次のコマンドを実行して有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t
"REG_DWORD" -v "AdaptiveScalingEnabled" -d "0x00000001" --force
```

この機能は、機能が有効になった後に起動されたセッションにのみ適用されます。

HDX™ Direct Linux向け

Citrixが提供するリソースにアクセスする場合、HDX Directは、直接通信が可能な場合、内部および外部のクライアントデバイスがセッションホストとの安全な直接接続を確立することを可能にします。

システム要件

HDX Directを使用するためのシステム要件は以下のとおりです。

- コントロールプレーン
  - シトリックス DaaS™
  - シトリックス Virtual Apps and Desktops™ 2503 以降
- 仮想配信エージェント (VDA)
  - Linux: バージョン 2503 以降
- Workspace アプリ
  - Windows: バージョン 2503 以降
  - Linux: バージョン 2411 以降
  - Mac: バージョン 2411 以降
- アクセス層
  - シトリックス ワークスペース™
  - Citrix StoreFront™ 2503 以降
  - シトリックス ゲートウェイ サービス
  - Citrix NetScaler® ゲートウェイ

ネットワーク要件

HDX Directを使用するためのネットワーク要件は以下のとおりです。

セッションホスト

セッションホストにファイアウォールがある場合、内部接続に対して以下の受信トラフィックを許可する必要があります。

説明	送信元	プロトコル	ポート
直接内部接続	クライアント	TCP	443
直接内部接続	クライアント	UDP	443

クライアントネットワーク

以下の表は、内部および外部ユーザー向けのクライアントネットワークについて説明しています。

内部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
直接内部接続	TCP	クライアントネットワーク	1024–65535	VDAネットワーク	443
直接内部接続	UDP	クライアントネットワーク	1024–65535	VDAネットワーク	443

外部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
STUN (外部ユーザーのみ)	UDP	クライアントネットワーク	1024–65535	インターネット (下記の注記を参照)	3478, 19302
外部ユーザー接続	UDP	クライアントネットワーク	1024–65535	データセンターのパブリックIPアドレス	1024–65535

データセンターネットワーク

以下の表は、内部ユーザーと外部ユーザー向けのデータセンターネットワークについて説明しています。

内部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
直接内部接続	TCP	クライアントネットワーク	1024–65535	VDAネットワーク	443
直接内部接続	UDP	クライアントネットワーク	1024–65535	VDAネットワーク	443

外部ユーザー

説明	プロトコル	送信元	ソースポート	宛先	宛先ポート
STUN (外部ユーザーのみ)	UDP	VDAネットワーク	1024–65535	インターネット (下記の注記を参照)	3478, 19302
外部ユーザー接続	UDP	DMZ / 内部ネットワーク	1024–65535	VDAネットワーク	55000–55250
外部ユーザー接続	UDP	VDAネットワーク	55000–55250	クライアントのパブリックIP	1024–65535

注:

VDAとWorkspaceアプリの両方が、以下のサーバーに同じ順序でSTUNリクエストを送信しようとします。

- stun.cloud.com:3478
- stun.cloudflare.com:3478
- stun.l.google.com:19302

> **HDX Directポート範囲**ポリシー設定を使用して外部ユーザー接続のデフォルトポート範囲を変更する場合、対応するファイアウォール規則はカスタムポート範囲と一致する必要があります。

構成

HDX Directはデフォルトで無効になっています。この機能は、Citrixポリシーの**HDX Direct**設定を使用して構成できます。

- **HDX Direct**: 機能を有効または無効にする。
- 「**HDX Directモード**」：**HDX Direct**が内部クライアントのみで利用可能か、内部クライアントと外部クライアントの両方で利用可能かを決定します。
- 「**HDX Directポート範囲**」：VDAが外部クライアントからの接続に使用するポート範囲を定義します。

考慮事項

HDX Directを使用する際の考慮事項は次のとおりです。

- 外部ユーザー向けのHDX Directは、トランスポートプロトコルとしてEDT (UDP) を使用する場合にのみ利用可能です。したがって、**アダプティブトランスポート**を有効にする必要があります。
- **HDX Insight**を使用している場合、**HDX Direct**を使用すると、セッションがNetScaler Gatewayを介してプロキシされなくなるため、HDX Insightのデータ収集が妨げられることに注意してください。
- HDX Direct で独自の証明書を使用することは現在サポートされていません。

仕組み

HDX Directを使用すると、直接通信が可能な場合にクライアントがセッションホストへの直接接続を確立できます。HDX Directを使用して直接接続が行われる場合、自己署名証明書が使用され、ネットワークレベルの暗号化 (TLS/DTLS) で直接接続を保護します。

内部ユーザー

次の図は、内部ユーザーのHDX Direct接続プロセスの概要を示しています。



1. クライアントはGateway Serviceを介してHDXセッションを確立します。

2. 接続が成功すると、VDAはHDX接続を介して、VDAマシンのFQDN、そのIPアドレスのリスト、およびVDAマシンの証明書をクライアントに送信します。
3. クライアントは、VDAに直接到達できるかどうかを確認するためにIPアドレスをプローブします。
4. クライアントが共有されたいずれかのIPアドレスでVDAに直接到達できる場合、クライアントはVDAとの直接接続を確立します。この接続は、手順(2)で交換された証明書と一致する証明書を使用して(D)TLSで保護されます。
5. 直接接続が正常に確立されると、セッションは新しい接続に転送され、Gateway Serviceへの接続は終了します。

> **注:**
>
> 上記の手順2で接続を確立した後、セッションはアクティブになります。その後の手順は、ユーザーが仮想アプリケーションまたはデスクトップを使用する能力を遅らせたり、妨げたりすることはありません。その後の手順のいずれかが失敗した場合でも、Gateway を介した接続はユーザーのセッションを中断することなく維持されます。

外部ユーザー

次の図は、外部ユーザー向けのHDX Direct接続プロセスの概要を示しています。



1. クライアントはGateway Serviceを介してHDXセッションを確立します。
2. 接続が成功すると、クライアントとVDAの両方がSTUNリクエストを送信して、それぞれのパブリックIPアドレスとポートを検出します。
3. STUNサーバーは、クライアントとVDAにそれぞれのパブリックIPアドレスとポートで応答します。
4. HDX接続を介して、クライアントとVDAはそれぞれのパブリックIPアドレスとUDPポートを交換し、VDAはクライアントに証明書を送信します。
5. VDAは、クライアントのパブリックIPアドレスとUDPポートにUDPパケットを送信します。クライアントは、VDAのパブリックIPアドレスとUDPポートにUDPパケットを送信します。
6. VDAからのメッセージを受信すると、クライアントはセキュアな接続要求で応答します。
7. DTLS ハンドシェイク中に、クライアントは証明書が手順4で交換された証明書と一致することを確認します。検証後、クライアントは認証トークンを送信します。これで安全な直接接続が確立されます。

8. 直接接続が正常に確立されると、セッションは新しい接続に転送され、Gateway Serviceへの接続は終了します。

> **注:**
>
> 上記の手順2で接続を確立した後、セッションはアクティブになります。その後の手順は、ユーザーが仮想アプリケーションまたはデスクトップを使用する能力を遅らせたり、妨げたりすることはありません。その後の手順のいずれかが失敗した場合でも、Gateway を介した接続はユーザーのセッションを中断することなく維持されます。

NAT互換性

外部ユーザーデバイスとセッションホスト間の直接接続を確立するために、HDX Direct は、NAT トラバーサルのためのホールパンチングと、クライアントデバイスおよびセッションホストのパブリック IP アドレスとポートマッピングの交換を容易にするための STUN を活用します。これは、VoIP、ユニファイドコミュニケーション、P2P ソリューションの仕組みに似ています。

ファイアウォールやその他のネットワークコンポーネントが、STUNリクエストおよびHDXセッションのUDPトラフィックを許可するように構成されている限り、外部ユーザー向けのHDX Directは機能すると予想されます。ただし、ユーザーおよびセッションホストネットワークのNATタイプが互換性のない組み合わせとなり、HDX Directが失敗する特定のシナリオがあります。

検証

クライアントおよびセッションホスト上のNATタイプとフィルタリングは、STUNTMANのSTUNクライアントユーティリティを使用して検証できます。

1. stunprotocol.orgからターゲットプラットフォームに適したパッケージをダウンロードし、その内容を抽出します。
2. ターミナルプロンプトを開き、内容が抽出されたディレクトリに移動します。
3. 次のコマンドを実行します: `./stunclient stunserver2024.stunprotocol.org --mode behavior`
4. 出力を記録します。

   バインディングテストと動作テストが成功した場合、**バインディングテスト**と**動作テスト**の両方が成功を報告し、NAT動作が指定されます。

   NAT の成功(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/nat-compatibility-report-success.png)

テストが失敗した場合、**バインディングテスト**および/または**動作テスト**が失敗を報告します。

NAT の失敗(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/nat-compatibility-report-failure.png)

5. 次のコマンドを実行します: `./stunclient stunserver2024.stunprotocol.org --mode filtering`
6. 出力を記録します。

NAT フィルタリング(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/nat-compatibility-report-filtering.png)

クライアントとセッションホストの両方のテスト結果に基づいて、外部ユーザー向けのHDX Directが機能するかどうかを判断するには、次の表を参照してください。

クライアントNATの動作	クライアントNATのフィルタリング	セッションホストNATの動作	セッションホストNATのフィルタリング	動作が期待されますか？
エンドポイント独立マッピング	任意	エンドポイント独立マッピング	任意	はい
エンドポイント独立マッピング	エンドポイント独立フィルタリング	アドレス依存マッピング	任意	はい
エンドポイント独立マッピング	アドレス依存フィルタリング	アドレス依存マッピング	任意	いいえ
エンドポイント独立マッピング	アドレスおよびポート依存フィルタリング	アドレス依存マッピング	任意	いいえ
エンドポイント独立マッピング	エンドポイント独立フィルタリング	アドレスおよびポート依存マッピング	エンドポイント独立フィルタリング	はい
エンドポイント独立マッピング	アドレス依存フィルタリング	アドレス依存マッピング	任意	いいえ
エンドポイント独立マッピング	アドレスおよびポート依存フィルタリング	アドレス依存マッピング	任意	いいえ

クライアントNATの動作	クライアントNATのフィルタリング	セッションホストNATの動作	セッションホストNATのフィルタリング	動作が期待されますか？
アドレス依存マッピング	任意	エンドポイント独立マッピング	エンドポイント独立フィルタリング	はい
アドレス依存マッピング	任意	エンドポイント独立マッピング	アドレス依存フィルタリング	いいえ
アドレス依存マッピング	任意	エンドポイント独立マッピング	アドレスおよびポート依存フィルタリング	いいえ
アドレス依存マッピング	任意	アドレス依存マッピング	任意	いいえ
アドレス依存マッピング	任意	アドレスおよびポート依存マッピング	任意	いいえ
アドレスおよびポート依存マッピング	任意	エンドポイント独立マッピング	エンドポイント独立フィルタリング	はい
アドレスおよびポート依存マッピング	任意	エンドポイント独立マッピング	アドレス依存フィルタリング	いいえ
アドレスおよびポート依存マッピング	任意	エンドポイント独立マッピング	アドレスおよびポート依存フィルタリング	いいえ
アドレスおよびポート依存マッピング	任意	アドレス依存マッピング	任意	いいえ
アドレスおよびポート依存マッピング	任意	アドレスおよびポート依存マッピング	任意	いいえ
失敗	任意	任意	任意	いいえ

クライアントNATの動作	クライアントNATのフィルタリング	セッションホストNATの動作	セッションホストNATのフィルタリング	動作が期待されますか？
任意	任意	失敗	任意	いいえ
失敗	任意	失敗	任意	いいえ

カスタムセッションログオン画面とOSロック画面の制御

セッションログオン画面にカスタムの背景またはバナーメッセージを追加する

以下のコマンドを使用して、セッション**ログオン**画面にカスタムの背景またはバナーメッセージを追加できます。セッション**ログオン**画面に背景とバナーメッセージの両方を追加するには、バナーメッセージを背景画像に埋め込むことができます。セッションを開くと、まずバナーメッセージページが表示され、次に認証ダイアログが表示されます。

カスタムバナーメッセージのタイトルを設定するには、以下を実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"LogonDisplayStringTitle" -d "<Banner message title>" --force
```

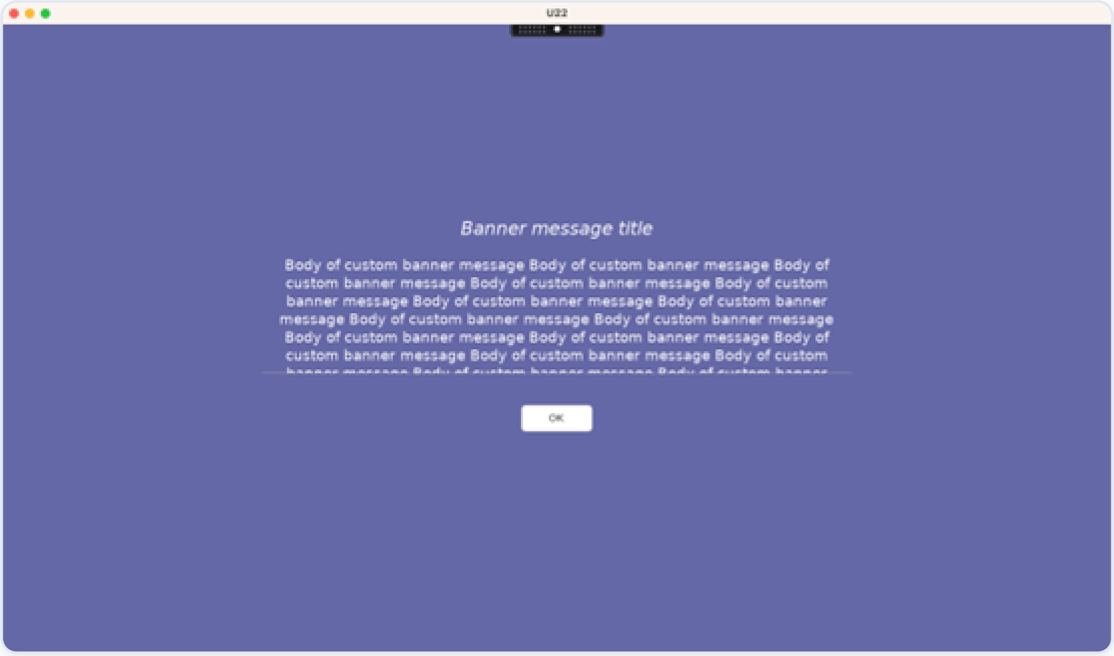
バナーメッセージのタイトルの最大長は64バイトです。

カスタムバナーメッセージの本文を設定するには、以下を実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"LogonDisplayString" -d "Body of custom banner message\nBody of custom
banner message\nBody of custom banner message\n" --force
```

バナーメッセージの本文の最大長は1,024バイトです。

> **ヒント:**
>
> `\n` 要素は改行を作成します。この例では、バナーメッセージ画面は次のようになります。

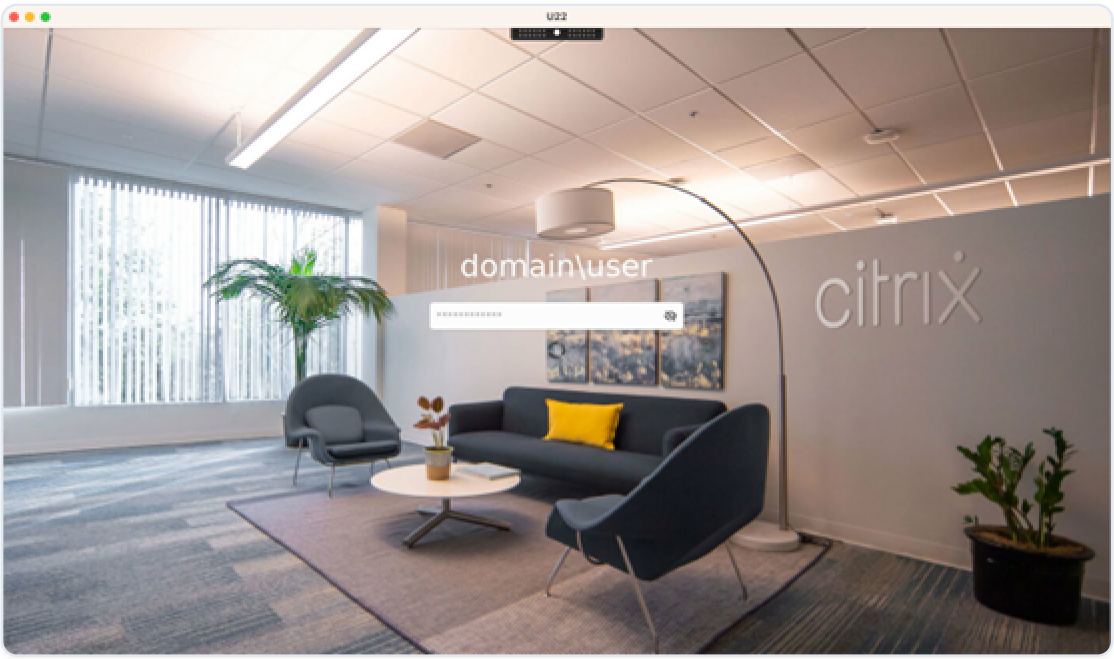


セッション**ログオン**画面にカスタムの背景を追加するには、以下を実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"BackgroundImagePath" -d "<path to the background image>" --force
```

カスタムの背景を表示するには、セッションユーザーは背景画像パスにアクセスできる必要があります。

例:



セッションログオン画面の例

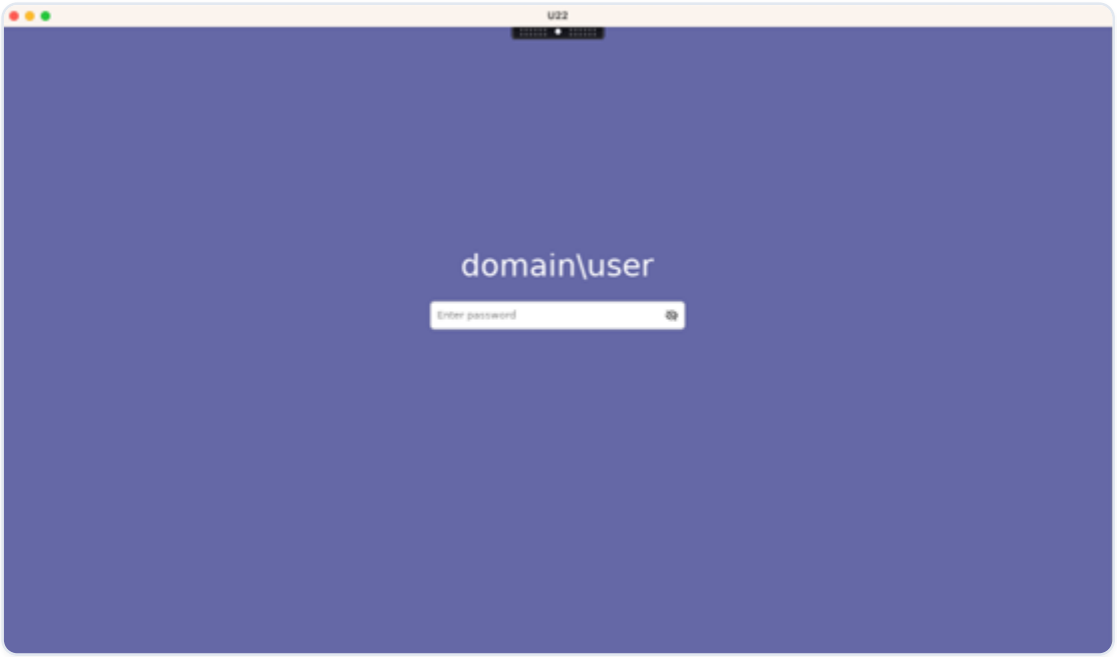
以下は、さまざまなシナリオにおけるセッションログオン画面の例です。

- シングルサインオン (SSO) シナリオにおけるセッション**ログオン**:



ログオンプロセスが表示されます。

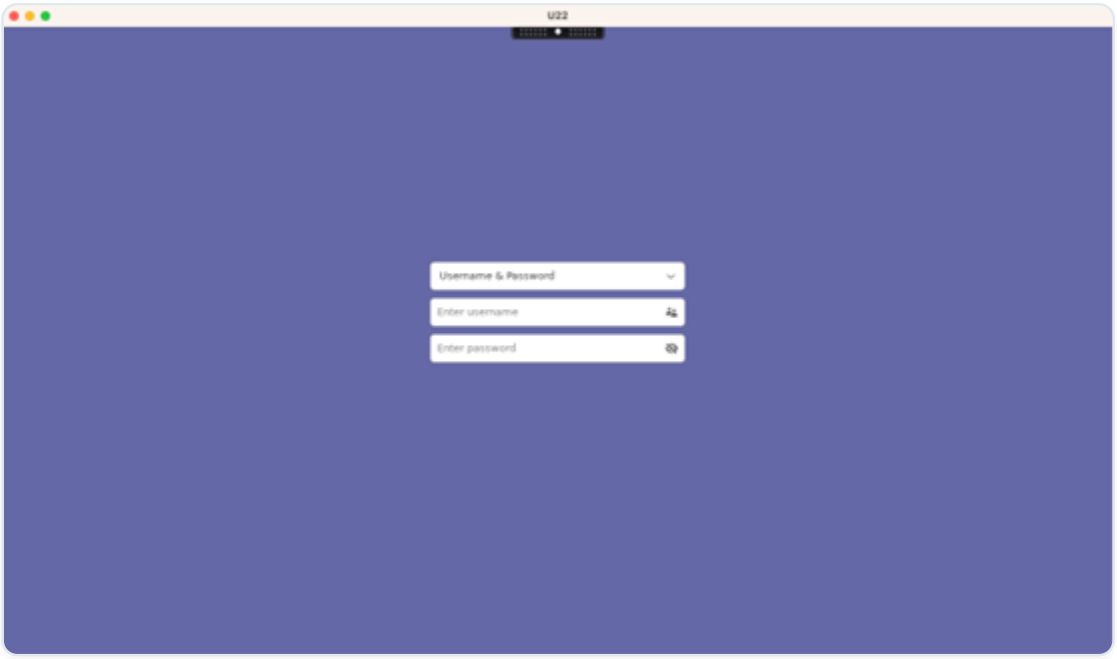
- 一般的な非SSOシナリオでのセッション**ログオン**:



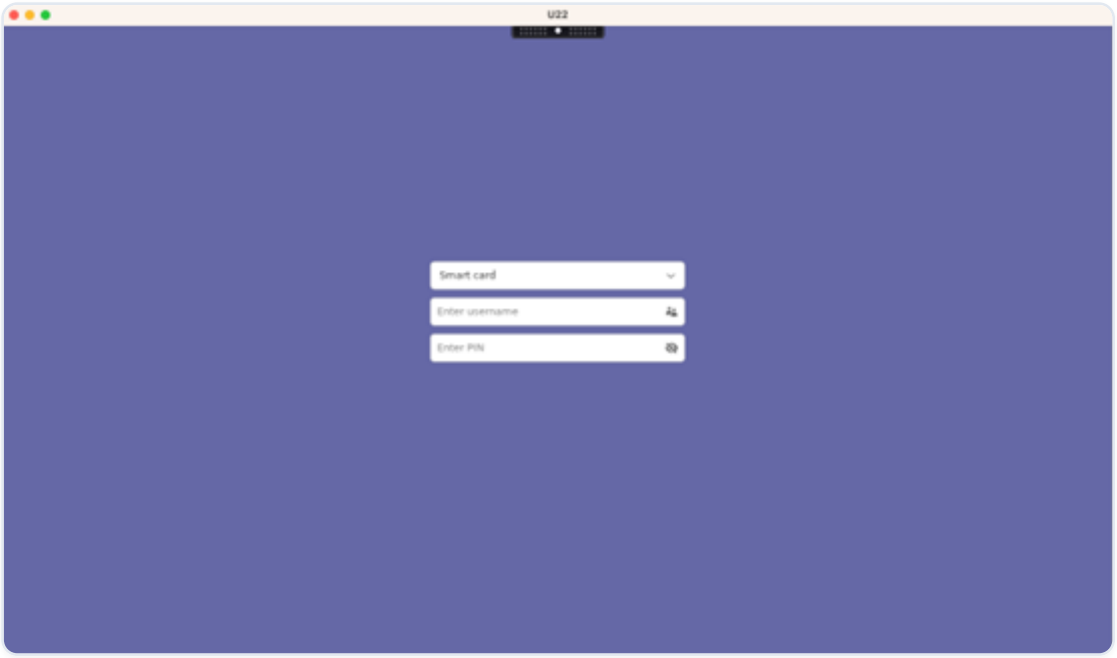
  - 非SSOシナリオでは、パスワードまたはPINコードが必要です。
  - ユーザーはパスワードとPINコードの表示を切り替えることができ、誤った入力を簡単に見つけられるようになります。

- ユーザーが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VDAセッションにログオンする際の、非SSOシナリオでのセッションログオン:

  セッションログオンに使用されるユーザー名とパスワード:

  セッションログオンに使用されるスマートカード:

  非SSOシナリオでサポートされているユーザー認証方法の組み合わせについては、「」を参照してください(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/configure/authentication/non-sso-authentication.html)。

構成可能なログオンバナー表示タイムアウト

カスタムタイムアウト値（秒単位）を設定するには、次のコマンドを使用します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v
"LogonDisplayTimeoutS" -d "<timeout_value>" --force
```

パラメーター

- 値を0にするとタイムアウトが無効になり、続行するにはユーザーの操作が必要になります。
- ゼロ以外の値は、タイムアウト期間を秒単位で指定します。有効な範囲: **1〜1800秒**。
- デフォルト値は**60秒**です。

OSロック画面の制御

概要

Linux VDA 2607では、**EnableOsLockScreen**レジストリキーが導入され、管理者はネイティブOSロック画面（GNOME、MATE、KDE Plasma、XFCE）がHDXセッション内で利用可能かどうかを明示的に制御できるようになりました。

以前は、Linux VDAはセッションの種類と認証方法に基づいてOSロック画面を自動的に有効または無効にしていました。2607以降、この動作は管理者制御の設定に置き換えられました。

OSロック画面はデフォルトで無効になっています。OSロック画面を有効にするには：

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Login" -t
"REG_DWORD" -v "EnableOsLockScreen" -d "0x00000001" --force
```

現在の値を確認するには：

```
/opt/Citrix/VDA/bin/ctxreg read -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Login" -v
"EnableOsLockScreen"
```

この設定は、**新しいセッションが開始されたとき**に適用されます。有効にすると、Linux VDAはデスクトップのロック画面、自動ロック設定、ロックショートカットを復元します。無効にすると、これらの機能は抑制されます。

> **注:**

> 匿名セッションでは、OSロック画面は常に無効になっています。

2607での動作変更

Linux VDA 2603以前では：

- ユーザー名/パスワードを使用するドメイン参加セッションでは、OSロック画面が自動的に有効になりました。
- その他の認証方法では無効になっていました。

2607以降、この自動動作は削除されました。

> **重要:**
>
> ドメイン参加ユーザーがOSロック画面に依存している場合は、アップグレード後に `EnableOsLockScreen=1` を設定してください。

注記と要件

- 設定はユーザープロファイルに保存され、将来のHDXログオンとローカルログオンの両方に影響を与える可能性があります。
- OSロック画面は、セッションで使用されるデスクトップ環境のネイティブスクリーンロッカーに依存します。この機能を有効にする前に、デスクトップ環境のスクリーンロッカーパッケージがインストールされていることを確認してください。たとえば、MATEの場合は `mate-screensaver`、Xfceの場合は `xfce4-screensaver` です。スクリーンロッカーがインストールされていない場合、**EnableOsLockScreen**はそのデスクトップ環境では効果がありません。
- PAMベースのロック解除には、**GNOMEはGDMを必要とします**。GDMが実行されていない場合、ロック解除に失敗する可能性があります。
- GNOME 3では、OSロック画面を無効のままにすると、VDAのウィンドウ装飾調整（ウィンドウシャドウの抑制、明示的な最小化/最大化/閉じるボタンのレイアウト）も適用されます。OSロック画面を有効にすると、デスクトップ独自のウィンドウ装飾のデフォルト設定が維持されます。
- Citrixセッションロック（ctxlockおよびスマートカード取り外しロックアクション）は影響を受けません。

セッションユーザーによるカスタムデスクトップ環境

セッションユーザーのデスクトップ環境は、**CTX_XDL_DESKTOP_ENVIRONMENT** 変数を使用して指定できます。2209リリース以降、セッションユーザーは独自のデスクトップ環境をカスタマイズできるようになりました。セッションユーザーがこの機能を使用できるようにするには、事前にVDAにデスクトップ環境をインストールする必要があります。

次の表は、セッションユーザーによるカスタムデスクトップ環境をサポートするLinuxディストリビューションとデスクトップ環境の対応表です。

LINUXディストリビューション	サポートされるデスクトップ
デビアン 12.12	メイト、グノーム、グノームクラシック、ケーディーイー、エックスエフシーイー
デビアン 13	メイト、グノーム、グノームクラシック、エックスエフシーイー
RHEL 10	GNOME
RHEL 9.8/9.6	メイト、グノーム、エックスエフシーイー
RHEL 8.10	メイト、グノーム、グノームクラシック、エックスエフシーイー
ロッキーリナックス 10	GNOME
ロッキー Linux 9.8/9.6	メイト、グノーム、Xfce
ロッキー Linux 8.10	メイト、グノーム、グノームクラシック、KDE、Xfce
SUSE 15.7	メイト、グノーム、グノームクラシック
ウブントゥ 22.04	メイト、グノーム、グノームクラシック、KDE、Xfce
ウブントゥ 24.04	メイト、KDE、Xfce

デスクトップ切り替えコマンド

> **注:**

> ターミナルとシステムトレイの両方からデスクトップ環境を切り替えることができます。

> ターミナルから目的のデスクトップ環境に切り替えるには、セッション内で対応するコマンドを実行します。

目的のデスクトップ環境が次の場合は、	コマンドを実行します。
GNOME	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME`
GNOME クラシック	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME-CLASSIC`
MATE	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh MATE`
KDE	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh KDE`

KDE のヒント

- KDE の起動時に Magnus がロードされることがあります。回避策として、`sudo apt remove magnus` を実行して Magnus パッケージを削除できます。
- KDE の起動中に発生する QT 警告を無効にするには、ルートユーザーとして `/usr/share/qt5/qtlogging.ini` を編集し、次のエントリを追加します。

  ```
  qt.qpa.xcb.xcberror.error=false

  qt.qpa.xcb.warning=false

  qt.qpa.xcb.error=false
  ```

- KDE で画面のロック解除に失敗する場合があります。回避策として、デスクトップの自動ロック機能を無効にすることをお勧めします。

一時ホームディレクトリでのログオン

Linux VDA上のマウントポイントが失敗した場合に備えて、一時ホームディレクトリを指定できます。一時ホームディレクトリが指定されている場合、マウントポイントが失敗すると、セッション

ログオン中にプロンプトが表示されます。ユーザーデータは一時ホームディレクトリに保存されます。

次の表は、ホームディレクトリ設定に役立つレジストリキーについて説明しています。

レジストリキー	説明	コマンド
`LogNoHome`	ユーザーがホームディレクトリなしでセッションにログオンできるかどうかを制御します。デフォルト値は1で、これは「はい」を意味します。値が0に設定されている場合、ホームディレクトリなしでのセッションログオンは無効になります。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "LogNoHome" -d "0x00000001" --force`
`HomeMountPoint`	Linux VDA上にローカルマウントポイントを設定します。たとえば、`/mnt/home` がマウントポイントの場合、ユーザーのホームディレクトリは `/mnt/home/domain/<user_name>` です。マウントポイントが環境内のユーザーホームディレクトリと同じであることを確認してください。この設定は、`CheckUserHomeMountPoint` が0に設定されている場合にのみ有効になります。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "HomeMountPoint" -d "<A directory where the NFS share is to be mounted>" --force`
`CheckUserHomeMountPoint`	Linux VDA上のマウントポイントとしてユーザー固有のホームディレクトリを設定するかどうかを制御します。ユーザー固有のホームディレクトリをマウントポイントとして設定する場合は、値を **1** に設定します。デフォルト値は **0** です。	`ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "CheckUserHomeMountPoint" -d "0x00000001" --force`

レジストリキー	説明	コマンド
`TempHomeDirectoryPath`	マウントポイントが失敗した場合に備えて、Linux VDA上に一時ホームディレクトリを設定します。デフォルト値は `/tmp` です。一時ホームディレクトリの設定は、`HomeMountPoint` と `CheckUserHomeMountPoint` によって決定されるマウントポイントが利用できない場合にのみ有効になります。ユーザーの一時ホームディレクトリは `/tmp/CTXSmf_user_id` です。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "TempHomeDirectoryPath" -d "</tmp by default>" --force`
`CheckMountPointRetryTime`	マウントが成功したかどうかを1秒に1回の頻度でチェックする回数を設定します。デフォルト値は5です。	`ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "CheckMountPointRetryTime" -d "0x000000010" --force`
`RemoveHomeOnLogoff`	ユーザーログオフ時に一時ホームディレクトリを削除するかどうかを制御します。1は「はい」を意味します。0は「いいえ」を意味します。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "RemoveHomeOnLogoff" -d "0x00000000" --force`

例: Linux VDAでのホームディレクトリマウント失敗の処理

シナリオ

`/mnt/home` のNFS共有からユーザーホームディレクトリをマウントするように構成されたLinux VDAがあります。ネットワークの問題により、マウントが時々失敗することがあります。ユーザーが引き続きログインでき、一時ホームディレクトリが自動的に作成されるようにしたいと考えています。

段階的な構成

1. マウントポイントを設定する

```
ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" \ -t "REG_SZ" -v "HomeMountPoint" -d "/mnt/home" --force
```

これは、ユーザーホームディレクトリ用のNFS共有をどこにマウントするかをVDAに指示します。

2. 共通のマウントポイントを使用する（ユーザー固有ではない）

```
ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" \ -t "REG_DWORD" -v "CheckUserHomeMountPoint" -d "0x00000000" --force
```

これにより、ユーザー固有のマウントポイントが無効になり、共有の `/mnt/home/domain/` パスが使用されます。ユーザー固有のマウントポイントのチェックが必要な場合は、値を1に設定します。

3. 一時ホームディレクトリのフォールバックを設定する

```
ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" \ -t "REG_SZ" -v "TempHomeDirectoryPath" -d "/tmp" --force
```

マウントが失敗した場合、ユーザーは一時ホームとして /tmp/CTXSmf_ を取得します。

4. マウントチェックを10回再試行する（1秒に1回）

```
ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" \ -t "REG_DWORD" -v "CheckMountPointRetryTime" -d "0x0000000A" --force
```

これにより、システムはフォールバックする前にマウントを10秒間待機します。

5. ログオフ時に一時ホームディレクトリを削除する

```
ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" \ -t "REG_DWORD" -v "RemoveHomeOnLogoff" -d "0x00000001" --force
```

セッション終了後に `/tmp/CTXSmf_` をクリーンアップします。

動作

ユーザーがログインし、`/mnt/home` がマウントされていない場合、システムは次のようになります。

- マウントを最大10秒間待機します
- セッションログオン中に、一時ホームフォルダーが使用されていることを示すプロンプトが表示されます

- ホームディレクトリ `/tmp/CTXSmf_` が作成された状態でログインを許可し、ユーザーデータは一時ホームディレクトリに保存されます
- ログアウト後に一時ディレクトリをクリーンアップします

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プラットフォームに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。
> - アプリケーションとデスクトップのデリバリーには、個別のVDAと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。
- Linuxアプリケーションには、使用中のアプリケーションに関する情報を含む「バージョン情報」ウィンドウがよくあり、詳細情報へのWebリンクがそれらのウィンドウによく見られます。「バージョン情報」ウィンドウのWebリンクをクリックすると、**calc**、**gedit**、**calendar**、**LibreOffice Suite**などの公開アプリケーション内からWebブラウザが起動する可能性があります。意図しないWebブラウザの起動は、アプリケーションの分離を迂回し、セキュリティを侵害する可能性があります。この問題を解決するには、以下の手順を実行してデフォルトのWebブラウザを変更します。

  a. カスタムの場所にnone.shファイルを作成します。例:

```
sudo mkdir /home/none



sudo touch /home/none/none.sh
```

```
sudo chmod +x /home/none/none.sh
```

b. none.shファイルに次の行を追加します。

```
#!/bin/bash


echo "NONE"
```

c. sudoを使用して/etc/xdg/mimeapps.listファイルを作成し、その後、mimeapps.listファイルに次の行を追加します。

```
[Default Applications]


text/html=none.desktop


x-scheme-handler/http=none.desktop


x-scheme-handler/https=none.desktop


x-scheme-handler/about=none.desktop


x-scheme-handler/unknown=none.desktop
```

d. sudoを使用して/usr/share/applications/none.desktopファイルを作成し、その後、none.desktopファイルに次の行を追加します。

```
[Desktop Entry]


Encoding=UTF-8


Version=1.0


Type=Application


Terminal=false


Exec=/home/none/none.sh


Name=None


Icon=/home/none/none.sh
```

none.shファイルをカスタムの場所に配置できるため、none.desktopがnone.shファイルを正しく参照できることを確認してください。

ランデブー V1

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。

要件

- 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 (旧称 Citrix Virtual Apps and Desktops™ service)。
- Linux VDA バージョン 2112 以降。
  - バージョン 2112 は、非透過型HTTPプロキシに必要な最小バージョンです。
  - バージョン 2204 は、透過型およびSOCKS5プロキシに必要な最小バージョンです。
- 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め、`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許可リストに追加できない場合は、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 serviceの下）およびKnowledge Centerの記事「CTX270584」を参照してください。
- Cloud Connectorsは、セッションをブローカーする際にVDAのFQDNを取得する必要があります。このタスクは、次の2つの方法のいずれかで実行します。
  - **サイトのDNS解決を有効にする。** **Full Configuration > Settings** に移動し、**Enable DNS resolution** 設定をオンにします。または、Citrix Virtual Apps and Desktops Remote PowerShell SDK を使用してコマンド `Set-BrokerSite -DnsResolutionEnabled $true` を実行します。Citrix Virtual Apps and Desktops Remote PowerShell SDK の詳細については、「SDKとAPI」を参照してください。
  - **VDAのPTRレコードを含むDNS逆引き参照ゾーン。** このオプションを選択する場合、VDAが常にPTRレコードの登録を試みるように構成することをお勧めします。これを行うには、グループポリシーエディターまたはグループポリシーオブジェクトを使用し、**Computer Configuration > Administrative Templates > Network > DNS Client** に移動して、**Register PTR Records** を **Enabled and Register** に設定します。接続のDNSサフィックスがドメインのDNSサフィックスと一致しない場合は、マシンがPTRレコードを正常に登録するために、**Connection-specific DNS suffix** 設定も構成する必要があります。

> **注:**

> DNS解決オプションを使用する場合、Cloud ConnectorsはVDAマシンの完全修飾ドメイン名（FQDN）を解決できる必要があります。内部ユーザーがVDAマシンに直接接続する場合、クライアントデバイスもVDAマシンのFQDNを解決できる必要があります。

> DNS逆引き参照ゾーンを使用する場合、PTRレコード内のFQDNはVDAマシンのFQDNと一致する必要があります。PTRレコードに異なるFQDNが含まれている場合、Rendezvous接続は失敗します。たとえば、マシンのFQDNが `vda01.domain.net` の場合、PTRレコードには `vda01.domain.net` が含まれている必要があります。`vda01.sub.domain.net` のような異なるFQDNは機能しません。

プロキシ構成

VDAは、HTTPおよびSOCKS5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際は、次の点を考慮してください。

- 非透過型HTTPプロキシとSOCKS5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構成してください。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て、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Kerberosプロトコルを自動的に選択します。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、セッション確立時に `HTTP/<proxyURL>` の形式でSPNを生成します。このとき、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。

- SOCKS5プロキシでの認証は現在サポートされていません。SOCKS5プロキシを使用する場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように例外を構成する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

透過型プロキシ

Rendezvousでは透過型HTTPプロキシがサポートされています。ネットワークで透過型プロキシを使用している場合、VDAに追加の構成は必要ありません。

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。この設定が有効になっている場合、VDAが使用するプロキシを認識できるように、HTTPまたはSOCKS5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>` または `socks5://<URL or IP>:<port>`

Rendezvousの検証

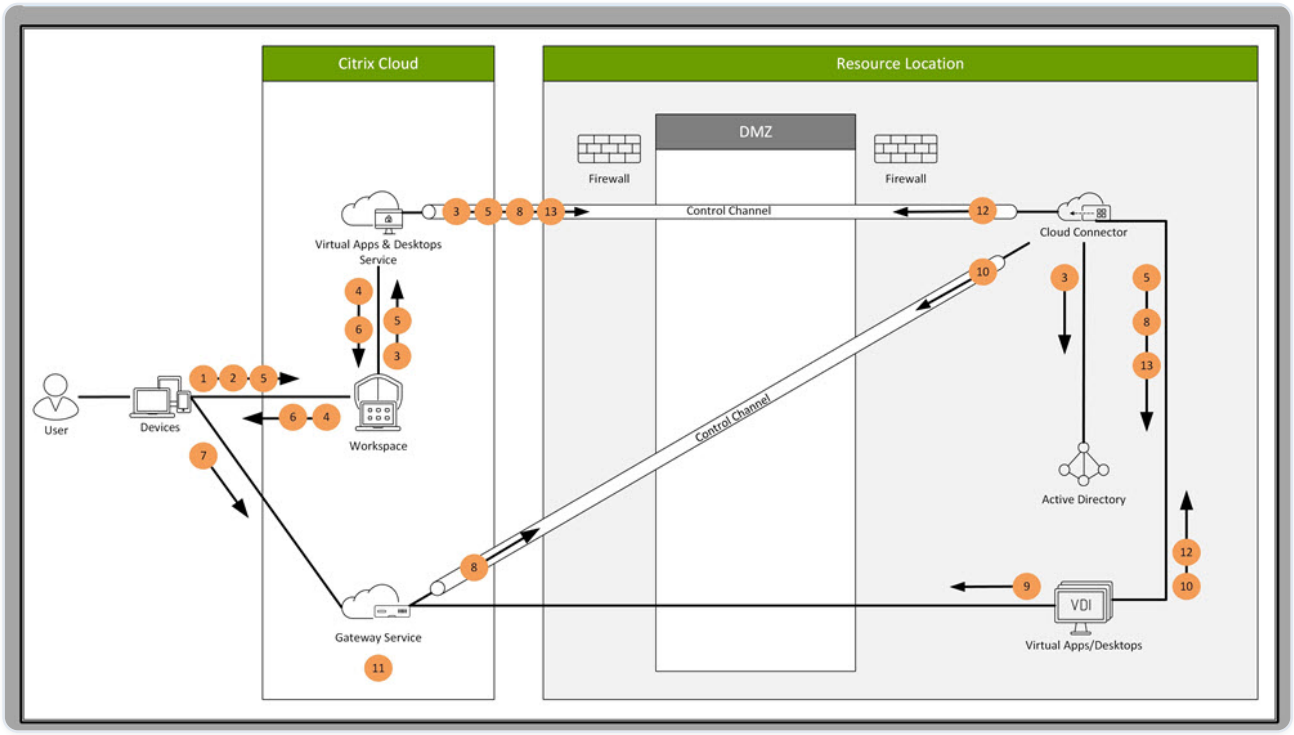
すべての要件を満たしている場合は、Rendezvousが使用されているかどうかを検証するために、次の手順に従ってください。

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経由のプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

> **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。

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、以下の手順に従ってください。

1. Citrix Workspaceに移動します。
2. Citrix Workspaceで資格情報を入力します。
3. オンプレミスのActive Directoryを使用している場合、Citrix 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よう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がその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、その情報をVDAに送信します。
9. VDAはCitrix Gatewayサービスへの直接接続を確立します。
10. VDAとCitrix Gatewayサービス間の直接接続が不可能な場合、VDAはCloud Connectorを介して接続をプロキシします。

11. Citrix Gatewayサービスは、エンドポイントデバイスとVDA間の接続を確立します。
12. VDAは、Cloud Connectorを介してCitrix 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

When using the Citrix Gateway service, the Rendezvous protocol allows traffic to bypass the Citrix Cloud™ Connectors and connect directly and securely with the Citrix Cloud control plane.

考慮すべき通信には、1) VDA登録とセッションブローカリングのための制御通信、2) HDX™セッション通信の2種類があります。

Rendezvous V1では、HDXセッション通信は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通信をプロキシするために、Cloud Connectorが引き続き必要です。

Standard AD domain joined machines and non-domain joined machines are supported for using Rendezvous V2 with single-session and multi-session Linux VDAs. With non-domain joined machines, Rendezvous V2 allows for both HDX traffic and control traffic to bypass the Cloud Connectors.

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。
    - HTTPおよびSOCKS5プロキシにはバージョン2204が最低限必要です。
- 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は以下にアクセスできる必要があります。
    - `https://*.xendesktop.net` on `TCP 443`。この方法ですべてのサブドメインを許可できない場合は、`https://<customer_ID>.xendesktop.net`を使用できます。ここで、`<customer_ID>`はCitrix Cloud管理者ポータルに表示されるCitrix Cloudカスタマーidです。

  - `https://*.nssvc.net`（すべてのサブドメインを含む）。こ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ドキュメントの「Internet Connectivity Requirements」セクション（Virtual Apps and Desktop serviceの下）と、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 443で、TCP Rendezvous用。
  - UDP 443で、EDT Rendezvous用。

プロキシ構成

Rendezvousを使用する場合、VDAは制御トラフィックとHDXセッショントラフィックの両方でプロキシ経由の接続をサポートします。両方のトラフィックタイプで要件と考慮事項が異なるため、注意深く確認してください。

制御トラフィックのプロキシに関する考慮事項

- HTTPプロキシのみ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Citrix Cloudコントロールプレーン間の制御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、接続が失敗します。
- プロキシ認証はサポートされていません。
- 制御トラフィックのプロキシを構成するには、次のようにレジストリを編集します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ProxySettings" -d "http://<URL or IP>:<port>" --force
```

HDXトラフィックのプロキシに関する考慮事項

- HTTPおよびSOCKS5プロキシがサポートされています。
- EDTはSOCKS5プロキシでのみ使用できます。
- HDXトラフィックのプロキシを構成するには、Rendezvousプロキシ構成ポリシー設定を使用します。
- 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Citrix Cloudコントロールプレーン間のHDX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、接続が失敗します。

- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用して、マシンベースの認証をサポートします。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、セッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。このとき、プロキシURLは**Rendezvous proxy**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。

- SOCKS5プロキシでの認証は現在サポートされていません。SOCKS5プロキシを使用している場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように、例外を設定する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

透過型プロキシ

Rendezvousでは透過型HTTPプロキシがサポートされています。ネットワークで透過型プロキシを使用している場合、VDAに追加の設定は必要ありません。

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. (オプション) CGS FQDN IPアドレスのキャッシュを有効にするオプションのレジストリキーを設定します `/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Ica\EdgeNodeCache" -t "REG_DWORD" -v "EnableDNSCache" -d "0x00000001" --force`

4. VDAマシンを再起動します。
5. 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**Allowed**に設定します。Rendezvousプロトコルはデフォルトで無効になっています。Rendezvousプロトコルが有効（**Allowed**）になっている場合、V1ではなくRendezvous V2が有効になります。
    - 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ます。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ポリシーが正しい優先順位を持っていることを確認し、他のポリシーを上書きしないようにします。

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。

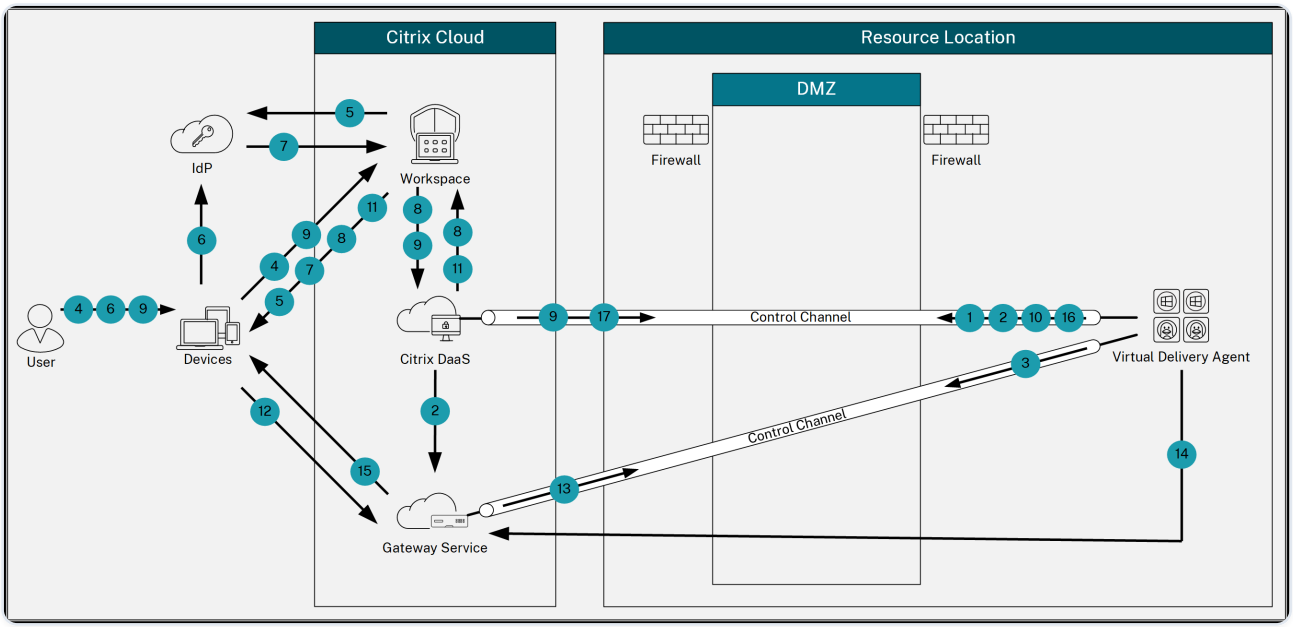
- TCP ランデブー: TCP - TLS - CGP - ICA
- EDT ランデブー: UDP - DTLS - CGP - ICA
- Cloud Connectorを介したプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

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。

Rendezvousトラフィックフロー

次の図は、Rendezvousトラフィックフローに関する一連の手順を示しています。



1. VDAはCitrix CloudとのWebSocket接続を確立し、登録します。
2. VDAはCitrix Gateway Serviceに登録し、専用のトークンを取得します。
3. VDAはGateway Serviceとの永続的な制御接続を確立します。
4. ユーザーはCitrix Workspaceに移動します。
5. Workspaceは認証構成を評価し、認証のために適切なIdPにユーザーをリダイレクトします。
6. ユーザーは資格情報を入力します。
7. ユーザー資格情報の検証が正常に完了すると、ユーザーはWorkspaceにリダイレクトされます。
8. Workspaceはユーザーのリソースをカウントし、それらを表示します。
9. ユーザーはWorkspaceからデスクトップまたはアプリケーションを選択します。WorkspaceはCitrix DaaS™にリクエストを送信し、Citrix DaaS™は接続を仲介し、VDAにセッションの準備をするよう指示します。
10. VDAはRendezvous機能とそのIDで応答します。
11. Citrix DaaSは起動チケットを生成し、Workspaceを介してユーザーデバイスに送信します。
12. ユーザーのエンドポイントはGateway Serviceに接続し、起動チケットを提供して、接続するリソースを認証および識別します。
13. Gateway Serviceは接続情報をVDAに送信します。
14. VDAはGateway Serviceとのセッションのために直接接続を確立します。
15. Gateway ServiceはエンドポイントとVDA間の接続を完了します。
16. VDAはセッションのライセンスを検証します。
17. Citrix DaaSは該当するポリシーをVDAに送信します。

セキュアHDX™

Citrix Workspace™アプリ（クライアント）とVDA（セッションホスト）の間で、ICA®セッションをエンドツーエンドで暗号化できます。

エンドツーエンド暗号化（E2EE）機能により、Citrix Gatewayを含む中間ネットワーク要素がICAトラフィックを復号化することはできません。これにより、環境のセキュリティ体制が向上し、設定と管理が容易になります。

システム要件

- Linux VDA 最小バージョン 2311
- Delivery Controller™ 最小バージョン 2308
- StoreFront™ 最小バージョン 2308
- Windows向けCitrix Workspaceアプリ 最小バージョン 2308

構成

エンドツーエンド暗号化を有効にする

エンドツーエンド暗号化（E2EE）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、Citrix Studioで**Secure HDX**ポリシーを**有効**に設定します。

証明書の更新をスケジュールする

エンドツーエンド暗号化（E2EE）機能には、Linux VDA上の**ctxcertmgr**サービスが管理する自己署名証明書とその秘密キーが必要です。

ctxcertmgrサービスが開始または再起動すると、新しい自己署名証明書が作成されます。デフォルトでは、**ctxcertmgr**サービスは2年ごとに午前2時に証明書（秘密キーを含む）を更新します。次のようなレジストリ設定で証明書の更新をスケジュールすることもできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_SZ" -v "CaRotationStartDate" -d "2023-10-19" --force



/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_SZ" -v "CaRotationTime" -d "00:45:30" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_DWORD" -v "CaRotationPeriod" -d "0x00000005" --force
```

上記の例では、最初の証明書更新時刻は2023-10-19 00:45:30に設定されています。その後、**ctxcermgr**サービスは5日ごとに00:45:30に証明書を更新します。スケジュールされた日時とは、Linux VDA上の日時です。

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、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 で無効になっています。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:**
>
> Citrix Workspace™ アプリでどのバージョンの DTLS が使用されているかを確認します。Linux VDA と Citrix Workspace アプリの両方で同じバージョンの DTLS が使用されていることを確認してください。

TLS を使用してユーザーセッションを保護する

バージョン 7.16 以降、Linux VDA はセキュアなユーザーセッション向けに TLS 暗号化をサポートします。TLS 暗号化はデフォルトで無効になっています。

TLS 暗号化を有効にする

セキュアなユーザーセッション向けに TLS 暗号化を有効にするには、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/2607-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、**/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 を有効にするスクリプト (/en-us/linux-virtual-delivery-agent/2607-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 サーバーおよび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、ターゲットのデリバリーグループに対してTLS暗号化を有効にするには、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関連項目に対してのみFQDNを提供します。

ControllerでTLS暗号化を無効にするには、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

公開デスクトップセッションにアクセスしようとすると、Windows向けCitrix Workspace™アプリで次の「要求されたアドレスを割り当てできません」エラーが発生する場合があります。



回避策として、**hosts**ファイルに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信頼性機能を提供します。セッション信頼性はデフォルトで有効になっています。

セッション信頼性により、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信頼性」を参照してください。

> **注**:
>
> セッション信頼性接続を介して送信されるデータは、デフォルトではプレーンテキストです。セキュリティ上の理由から、TLS暗号化を有効にすることをお勧めします。TLS暗号化の詳細については、「TLSを使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studioのポリシー設定

Citrix Studioでセッション信頼性について以下のポリシーを設定できます。

- セッションの信頼性接続
- セッションの信頼性タイムアウト
- セッションの信頼性ポート番号
- 再接続UIの透過レベル

詳しくは、「セッション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:**
>
> **セッション信頼性接続**または**セッション信頼性ポート番号**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\cgp" -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、セッション信頼性は無効になりません。**セッション信頼性接続**ポリシーを通じて機能が有効になっている場合、セッション信頼性は他のリスナー（SSLなど）を通じて引き続き利用可能です。

- **セッション信頼性ポート番号**

次のコマンドを使用して、セッション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\cgp" -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。ポート番号が**Citrix Studio**のポリシー設定を通じて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

ポリシー設定を通じて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ません。

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、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するために、次のコマンドを実行します（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、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"
   ```

セッションをローカルストレージに記録する

Linux VDAでホストされているセッションを記録および再生できます。

セッション記録を有効または無効にする

Linux VDAのセッション記録を有効または無効にするには、**SmAudAllowed**をそれぞれ**1**または**0**に設定します。次のコマンドを使用できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000001" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000000" --force
```

注:

Linux VDAでセッション記録を有効にすると、ユーザーがセッションにログオンしたときに、セッションが記録されていることが通知されます。

記録のファイルサイズを指定する

記録のサイズが大きくなると、記録ファイルのダウンロードに時間がかかり、再生中にシークスライダーを使用してナビゲートする際の応答が遅くなります。ファイルサイズを制御するには、ファイルのしきい値制限を指定します。記録がこの制限に達すると、現在のファイルは閉じられ、記録を続行するために追加のファイルが作成されます。このアクションは「ロールオーバー」と呼ばれます。

次のコマンドを使用して、ロールオーバーの2つのしきい値を指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverFileSizeInMB" -d "0x00000032" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
```

```
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverTimeInHours" -d "0x0000000c" --force
```

- **RolloverFileSizeInMB**。現在のファイルは、指定されたサイズに達すると閉じられ、新しいファイルが開きます。デフォルトでは、サイズが50 MBを超えるとロールオーバーが発生します。サポートされる値：10～300。
- **RolloverTimeInHours**。指定された期間に達すると、現在のファイルは閉じられ、新しいファイルが開きます。デフォルトでは、セッションが12時間記録されるとロールオーバーが発生します。サポートされる値：1～24。

ロールオーバーは、上記の2つの条件のうち最初に満たされたときに発生します。たとえば、ファイルサイズに17 MB、期間に6時間を指定したとします。記録が3時間で17 MBに達すると、セッション記録はそのファイルを閉じ、新しいファイルを開きます。

多数の小さなファイルが作成されるのを防ぐため、ファイルサイズに指定された値に関係なく、少なくとも1時間が経過するまでロールオーバーは発生しません。この規則の例外は、ファイルサイズが300 MBを超えた場合です。

記録の保存場所を指定する

記録ファイルは、デフォルトで`/var/xdl/session_recordings`に保存されます。別のパスを指定するには、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\SmartAuditor" -t "REG_SZ" -v
"Path" -d "<your custom storage path>" --force
```

記録は、ローカルドライブまたはネットワークパスを指すマウントポイントに保存できます。設定したストレージパスに適切なアクセス許可を構成し、ユーザー`ctxsrvr`にそのパスへの書き込み許可を付与します。

録画を表示

録画を表示するには、以下の手順に従ってSession Recording PlayerまたはSession Recording Web Playerをインストールします。

1. Citrixアカウントの資格情報を使用して、Citrix Virtual Apps and Desktopsダウンロードページにアクセスし、製品ファイルをダウンロードします。ファイルを解凍します。
2. SessionRecordingPlayer.msiおよびSessionRecordingWebPlayer.msiをダブルクリックし、指示に従ってインストールを完了します。

> **ヒント:**
>
> Session Recording Web Playerを使用するには、Session Recordingサーバーにのみインストールし、録画がSession Recordingサーバーで利用可能であることを確認してください。詳細については、Citrix Session Recordingドキュメントを参照してください。

制限事項

- 仮想アプリセッションの場合、録画通知が中央に表示されないことがあります。

ベストプラクティス

このセクションでは、次の手順について説明します。

- WebSocket 用の自己署名証明書を構成する(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/best-practices/configure-self-signed-certificates-for-websocket.html)
- Machine Creation Services (MCS) を使用して Google Cloud Platform (GCP) 上に Linux VDA を作成する(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/best-practices/create-linux-vda-on-gcp-using-mcs.html)
- Ansible を使用して展開を管理する(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/best-practices/manage-your-deployment-using-ansible.html)

WebSocket 用の自己署名証明書を構成する

2402 LTSR の初回リリース以降、Citrix Virtual Apps and Desktops では、Citrix Brokering Protocol (CBP) を介した WebSocket テクノロジーを使用して、VDA と Delivery Controller 間の通信を容易にすることができます。この機能では、VDA から Delivery Controller への通信に TLS ポート 443 のみが必要です。詳細については、Citrix Virtual Apps and Desktops のドキュメントにある「VDA と Delivery Controller 間の WebSocket 通信」を参照してください。

WebSocket は、クライアントとサーバー間のリアルタイムで双方向の通信を可能にする強力なテクノロジーです。ただし、特に **wss://** を使用する場合に安全な接続を確保するには、特に開発環境やテスト環境では、自己署名証明書を構成することがしばしば必要になります。この記事では、WebSocket 用の自己署名証明書を構成するためのベストプラクティスについて説明します。

ステップ 1: (ドメインに参加していない VDA のみ) DNS サーバーを構成する

- **Ubuntu および Debian VDA の場合:**

1. **/etc/systemd/resolved.conf** を次のように編集して DNS 設定を変更します。

```
[Resolve]

# Some examples of DNS servers which may be used for DNS= and
FallbackDNS=:
```

```
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com
2606:4700:4700::1111#cloudflare-dns.com
2606:4700:4700::1001#cloudflare-dns.com

# Google:     8.8.8.8#dns.google 8.8.4.4#dns.google
2001:4860:4860::8888#dns.google 2001:4860:4860::8844#dns.google

# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net
2620:fe::fe#dns.quad9.net 2620:fe::9#dns.quad9.net

DNS=<DNS IP address>

#FallbackDNS=

#Domains=

#DNSSEC=no

#DNSOverTLS=no

#MulticastDNS=no

#LLMNR=no

#Cache=no-negative

#CacheFromLocalhost=no

#DNSStubListener=yes

#DNSStubListenerExtra=

#ReadEtcHosts=yes

#ResolveUnicastSingleLabel=no
```

2. systemd-resolved サービスを再起動します。

```
sudo service systemd-resolved restart
```

詳細については、`<https://notes.enovision.net/linux/changing-dns-with-resolve>` を参照してください。

- **RHEL、Rocky Linux、および SUSE VDA の場合:**

1. 次の **nmcli** コマンドを実行して、接続名のリストを取得します。

```
sudo nmcli connection
```

2. 別の **nmcli** コマンドを実行して、DNS IP アドレスを設定します。

```
sudo nmcli con mod {connectionNameHere} ipv4.dns "<dns ip address>"
```

たとえば、次のコマンドを使用して DNS IP アドレスを 192.168.2.254 に設定できます。

```
sudo nmcli con mod eth0 ipv4.dns "192.168.2.254"
```

3. 次のいずれかのコマンドを実行して、新しい DNS 設定を再読み込みします。

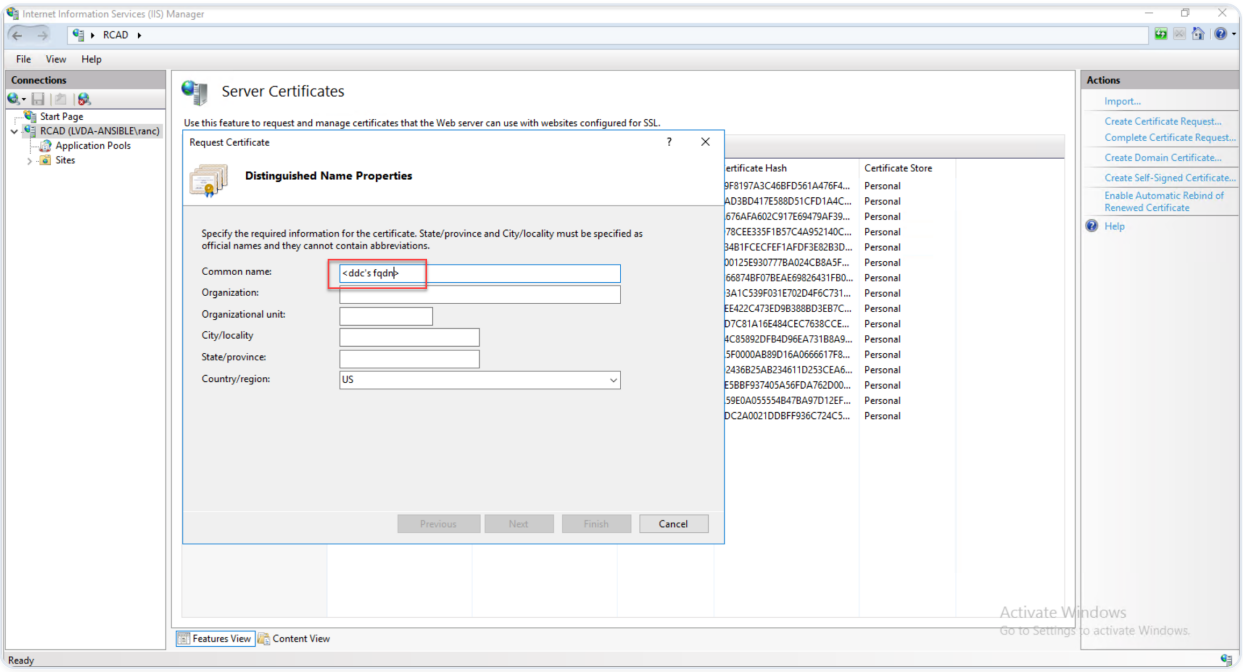
```
sudo systemctl restart NetworkManager.service
```

```
sudo nmcli connection reload
```

詳細については、`<https://www.cyberciti.biz/faq/change-dns-ip-address-rhel-redhat-linux/>` を参照してください。

ステップ 2: 認証局 (CA) に証明書を要求する

1. 証明書要求を開始します。証明書要求を開始するときは、Delivery Controller の完全修飾ドメイン名 (FQDN) を入力します。



2. 証明書の堅牢なセキュリティを確保するために、2048以上のビット長を選択します。

   ビット長を選択する(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/select-a-bit-length.png)

3. 識別しやすいように、証明書要求ファイルにわかりやすい名前を割り当てます。

   証明書要求ファイルに名前を付ける(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/name-the-certificate-request-file.png)

4. Notepadのようなテキストエディターを使用して生成された証明書要求ファイルを開き、その中のすべてのコンテンツを選択します。

   生成された証明書要求ファイルを開く(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/open-the-generated-certificate-request-file.png)

5. Web証明書サーバーにログインし、証明書を要求する手順に進みます。

   Web証明書サーバーにログインする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/log-in-to-your-web-certificate-server.png)

   高度な証明書要求(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/advanced-certificate-request.png)

6. コピーした要求ファイルの内容をWebサーバーの適切なフィールドに貼り付け、**Webサーバー**証明書テンプレートを選択します。

   証明書要求を送信する(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/certificate-template.png)

7. Base 64エンコード形式で証明書をダウンロードします。

証明書をダウンロードする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/download-the-certificate.png)

8. 証明書要求を完了します。ダウンロードが完了すると、証明書要求プロセスは完了です。

   証明書要求を完了する(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/complete-the-certificate-request.png)

ステップ3：Delivery Controller FQDN証明書をバインドする

1. Delivery Controller FQDN証明書をPFXファイルとしてエクスポートします。

   Delivery Controller FQDN証明書をエクスポートする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/export-the-delivery-controller-fqdn-certificate.png)

2. エクスポートされたPFX証明書を、Delivery ControllerサーバーのInternet Information Services (IIS) にインポートします。

   エクスポートされたPFX証明書をインポートする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/import-the-exported-pex-certificate.png)

3. インポートされた証明書をIIS内のデフォルトのWebサイトにバインドします。

   インポートされた証明書をバインドする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/bind-the-imported-certificate.png)

4. バインドプロセス中に、インポートした特定の証明書を選択するようにしてください。

   バインド中にインポートされた証明書を選択する(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/select-the-imported-certificate-during-binding.png)

ステップ4：Linux VDAでCA証明書を保存して更新する

1. CA証明書をダウンロードします。例：

   CA証明書をダウンロードする(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/download-ca-certificate.png)

2. CA証明書を配置して更新します。

   - **RHELおよびRocky Linuxの場合：**

     trust anchor <path/CA certificate> コマンドを使用してCA証明書を追加します。**/etc/pki/ca-trust/source/anchors** ディレクトリに証明書が手動で配置されていないことを確認してください。読み取り専用フィールドに関連するエラーが発生した場合は、そのディレクトリに存在する証明書をすべて削除してください。

- **SUSE、Ubuntu、およびDebianの場合：**

  ルートCA証明書を **/usr/local/share/ca-certificates** ディレクトリに配置します。証明書に **.crt** 拡張子がない場合は、それに応じて名前を変更します。その後、**update-ca-certificate** コマンドを実行します。

Machine Creation Services™ (MCS) を使用して、Google Cloud Platform (GCP) 上に Linux VDA を作成する

MCS を使用して GCP 上に Linux VDA を作成するには、次の手順を実行します。

ステップ 1: GCP 上に Linux 仮想マシン (VM) を作成する

ステップ 2: GCP サービスアカウントを作成する

ステップ 3: Citrix Studio で GCP へのホスト接続を作成する

ステップ 4: Linux VDA マスターイメージを準備する

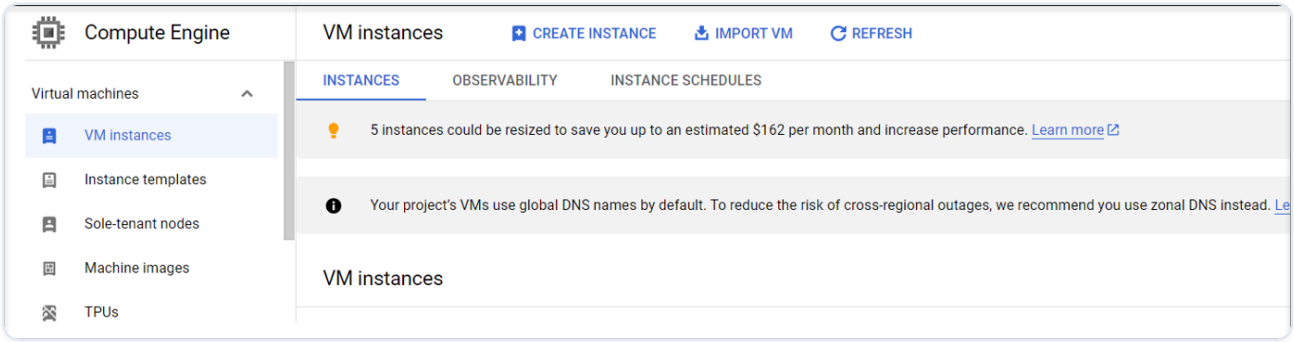
ステップ 5: マシンカタログを作成する

ステップ 6: デリバリーグループを作成する

ステップ 1: GCP 上に Linux VM を作成する

ステップ 1a: Google Cloud コンソールで、**Compute Engine > VM インスタンス**に移動します。

ステップ 1b: VM インスタンスページで、**インスタンスを作成**をクリックして VM インスタンスを作成します。



ステップ 1c: 次の構成を行い、その他の構成はデフォルトのままにします。

- VM インスタンスの名前を入力します。
- VM をホストするリージョンとゾーンを選択します。

- （オプション）VMにGPUを追加します。詳細については、この記事の後半にある**手順4c**を参照してください。
- **ブートディスク**セクションで、VMのオペレーティングシステムとディスクサイズを選択します。例：

  ブートディスク(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/boot-disk.png)
- **詳細オプション > ネットワーク**に移動し、**ホスト名**フィールドをFQDNに設定します。

  ホスト名フィールドを設定(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/hostname-field.png)

手順1d： **作成**をクリックしてVMインスタンスを作成します。

手順1e： VMが作成されたら、**Compute Engine**ダッシュボードに戻り、リストからVMインスタンスを見つけ、SSHボタンをクリックしてVMに接続します。

GCPでVMに接続するためにSSHボタンをクリック(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/connect-vm-on-gcp-via-ssh.png)

手順1f： WebベースのSSHクライアントを介して、Linux VDAパッケージをVMにアップロードします。

GCPでLinux VDAパッケージをVMにアップロード(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/upload-linux-vda-package-to-your-vm-on-gcp.png)

手順1g： SSHを使用したVMへのアクセス障害を防止します。

VMは再起動後に到達不能になる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認します。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

シリアルコンソールに接続(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/connect-to-serial-console.png)

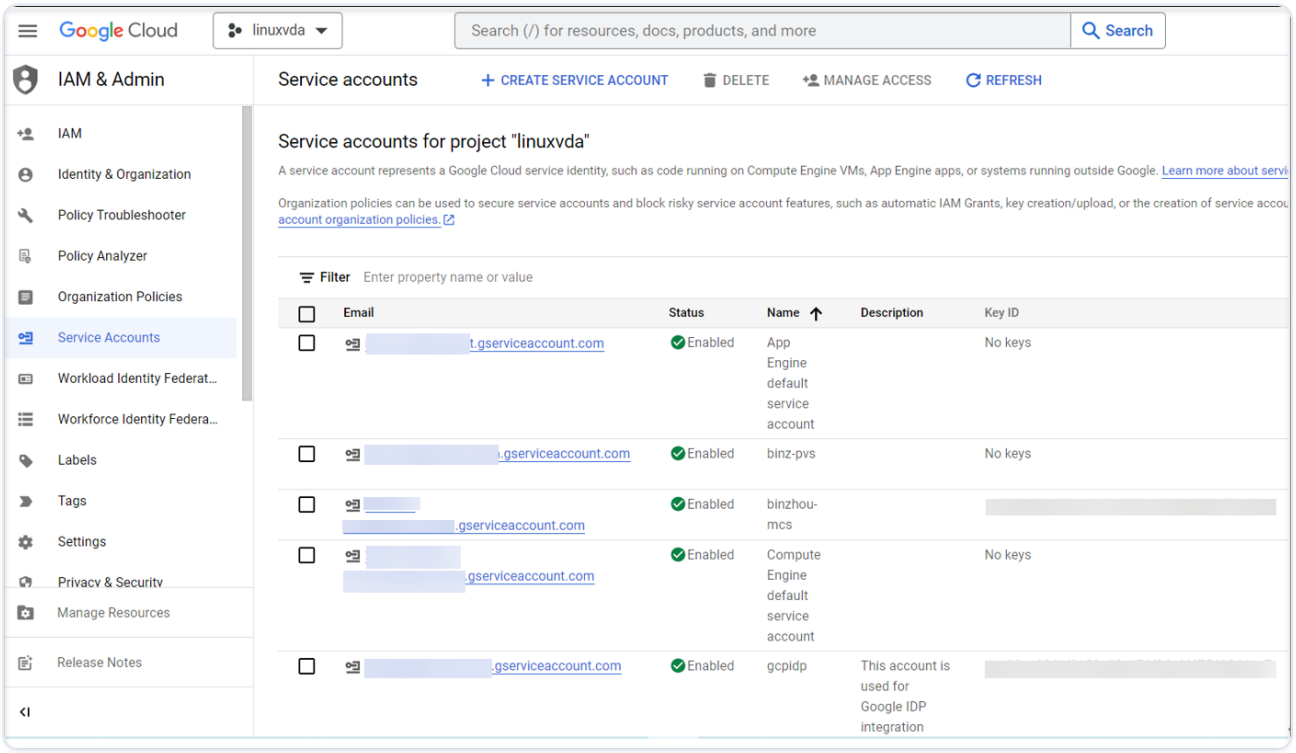
手順2：GCPサービスアカウントを作成する

このセクションでは、GCPサービスアカウントの作成について説明します。これには、サービスアカウントキーの作成と、サービスアカウントに必要な役割の付与が含まれます。

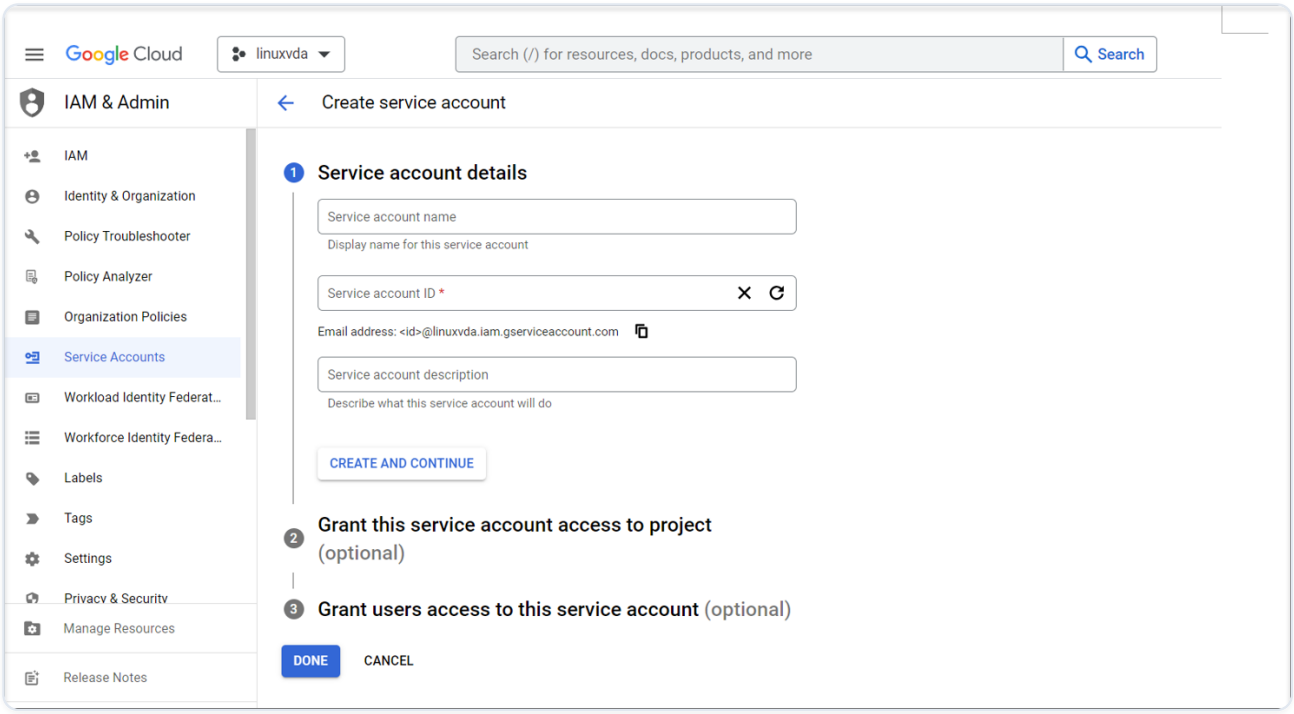
注:

GCP サービスアカウントを作成するには、Service Account Admin (roles/iam.serviceAccountAdmin) IAM ロールが付与されていることを確認してください。

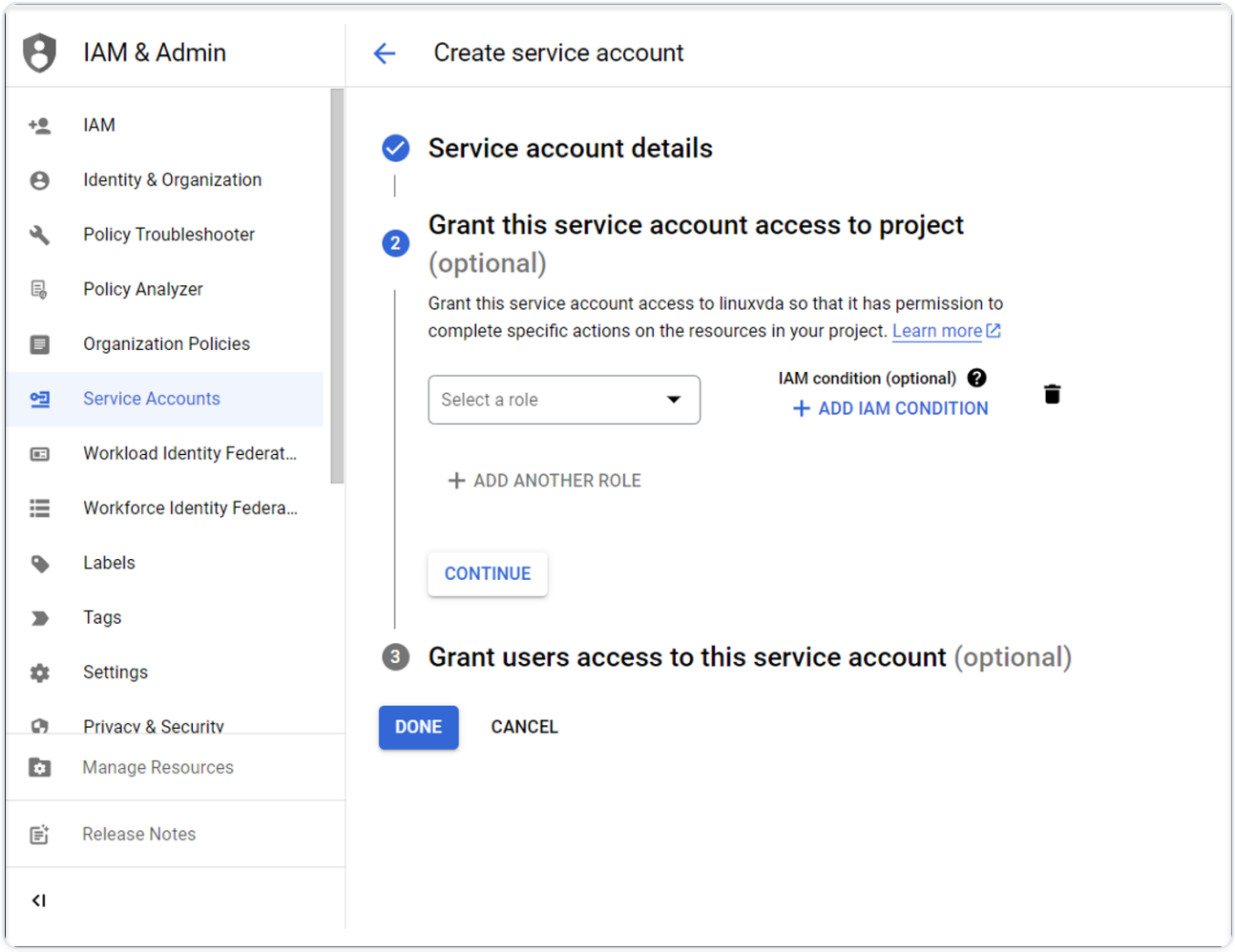
ステップ 2a: Google Cloud コンソールで、**IAM と管理 > サービスアカウント**に移動し、**サービスアカウントを作成**タブをクリックします。



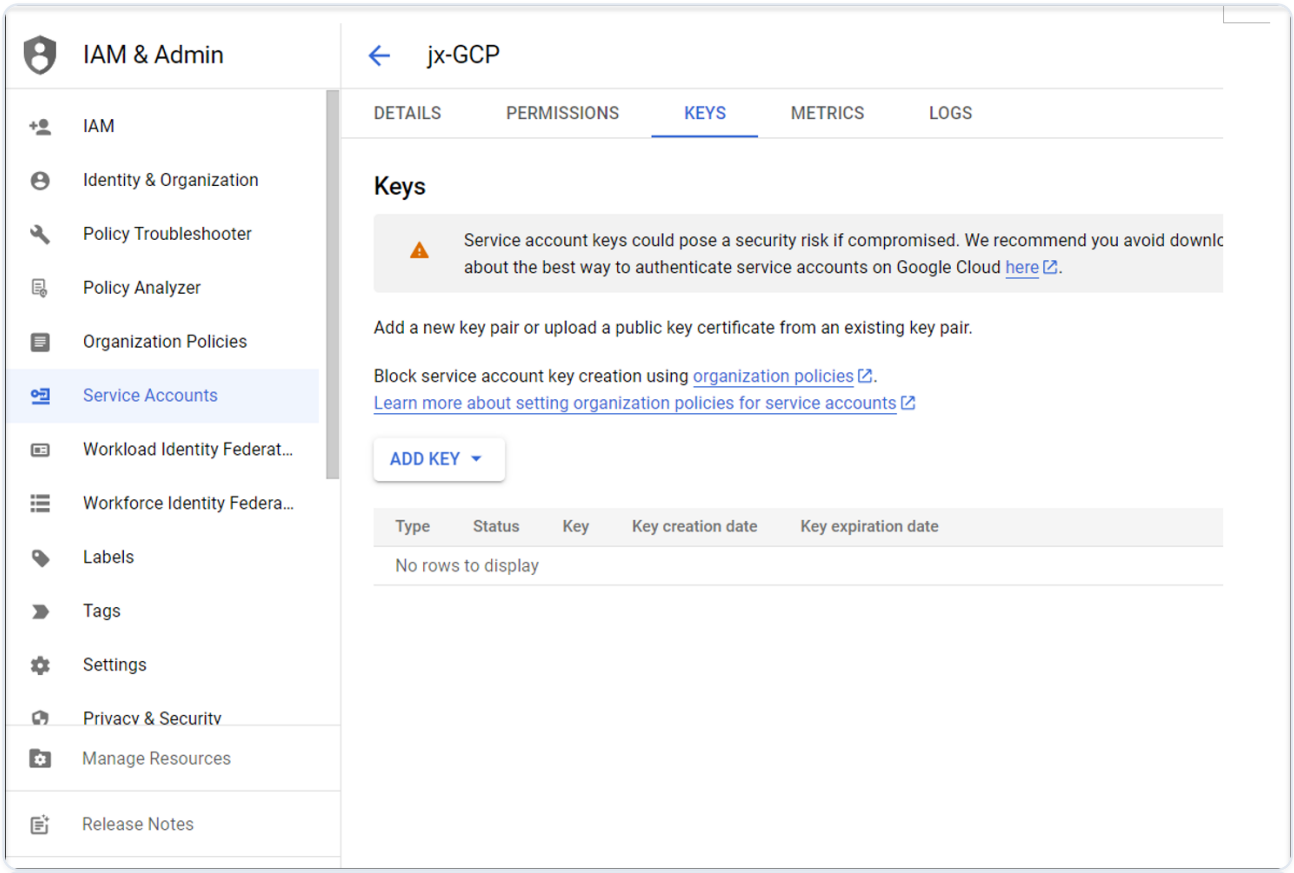
ステップ 2b: サービスアカウントの詳細ステップで、フィールドに値を入力し、**作成して続行**をクリックします。

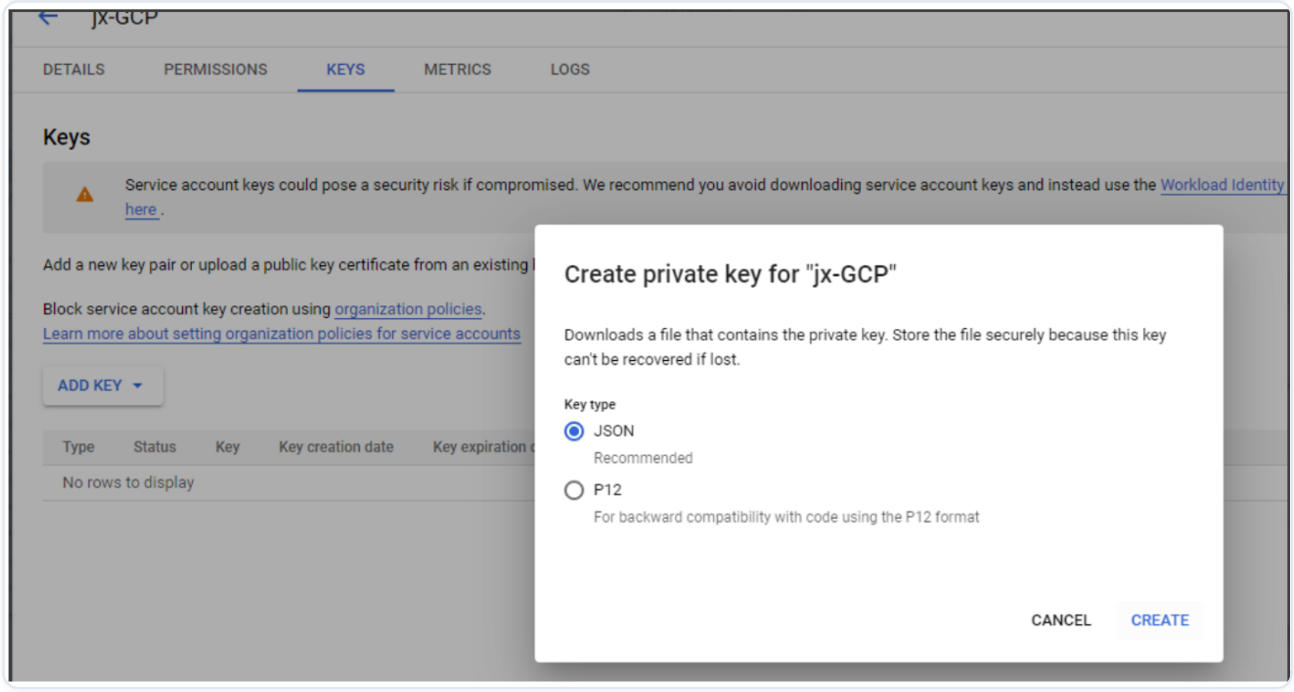


ステップ 2c: オプションの手順をスキップし、下部にある**完了**をクリックします。



ステップ 2d: 再度**IAM と管理 > サービスアカウント**に移動し、**サービスアカウント**タブをクリックします。新しく作成したサービスアカウントを見つけ、**キー**タブに移動し、**キーを追加 > 新しいキーを作成 > JSON > 作成**をクリックします。





注:

キーファイルをダウンロードすると、再度ダウンロードすることはできません。

ステップ 2e: Google Cloud コンソールで、**IAM と管理 > IAM**に移動し、**追加**をクリックします。**新しいメンバー**フィールドで新しく作成したサービスアカウントを検索して選択し、そのサービスアカウントにリソースへのアクセス権を付与するためのロールを選択します。**別のロールを追加**をクリックしてロールの付与を続け、新しく作成したサービスアカウントに以下のすべてのロールが付与されていることを確認します。

- **Compute 管理者**
- **Storage 管理者**
- **Cloud Build 編集者**
- **サービスアカウントユーザー**
- **クラウド データストア ユーザー**
- **Compute Instance 管理者 (ベータ版)**
- **所有者**

例:



ステップ 3: Citrix Studio で GCP へのホスト接続を作成する

Google Cloud Platform 仮想化環境 に従って GCP 環境をセットアップし、次の手順を実行して GCP へのホスト接続を作成します。

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で **> >** を選択してホスト接続を作成します。クラウド Delivery Controller の場合は、Citrix Cloud™ の Web ベースの Studio コンソールで **> >** を選択してホスト接続を作成します。
2. ウィザードで、接続の種類として を選択します。

   たとえば、Citrix Cloud の Web ベースの Studio コンソールでは次のようになります。

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/add-conn-connection.png)
3. GCP アカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードはページを順に案内します。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、 を選択して ページに到達します。詳細については、

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 の記事の ステップ 2: ホスト接続を作成する を参照してください。

ステップ 4: Linux VDA マスターイメージを準備する

ステップ 4a: (RHEL 8.x/9.x および Rocky Linux 8.x/9.x の場合) イーサネット接続を構成します。

GCP でホストされているサポート対象の RHEL または Rocky Linux ディストリビューションに 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ことがあります。この問題を回避するには、VM に初めてログオンするときに root パスワードを設定し、root として VM にログオンできることを確認します。その後、VM を再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

シリアルコンソールに接続(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/connect-to-serial-console.png)

Step 4b: テンプレートVMにLinux VDAパッケージをインストールします。

テンプレートVMで、Linux VDAパッケージをインストールするには、次の手順を実行します。

1. .NETをインストールします。

   Linux VDAパッケージをインストールする前に、システム要件に従って.NETをインストールします。(/ja-jp/linux-virtual-delivery-agent/2607-ltsr/system-requirements.html#net-requirements)

   .NET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。

2. Linux VDAパッケージをインストールします。

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

3. EPELリポジトリを有効にします。

   ```
   sudo rpm -ivh  https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   ```

Step 4c: GCP上のテンプレートVMにグラフィックス処理ユニット（GPU）を追加します（オプション）。

1. Google Cloudコンソールで、テンプレートVMに1つ以上のGPUを追加します。GCPでのGPUの追加と削除については、以下を参照してください。`<https://cloud.google.com/compute/docs/gpus/add-remove-gpus>`.

   GCP上のVMに1つ以上のGPUを追加(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/add-gpus-to-vm-on-gcp.png)

2. テンプレートVMに適切なGPUドライバーをインストールします。詳細については、以下を参照してください。`<https://cloud.google.com/compute/docs/gpus/install-drivers-gpu>`.

   必須のNVIDIAドライバーバージョン:

   Compute Engineで実行されているNVIDIA GPUは、以下のNVIDIAドライバーバージョンを使用する必要があります。

   - L4 GPUの場合：
     - Linux : 525.60.13 以降
   - A100 GPUの場合：
     - Linux : 450.80.02 以降
   - T4、P4、P100、およびV100 GPUの場合：
     - Linux : 410.79 以降
   - K80 GPUの場合（サポート終了）：
     - Linux : 410.79 - 最新のR470バージョン

   K80 GPUについて、NVIDIAはR470ドライバーブランチがデバッグサポートを受ける最終ドライバーバージョンであることを発表しました。この更新を確認するには、NVIDIAソフトウェアサポートマトリックスを参照してください。

   インストールスクリプト：

   インストールプロセスを自動化するには、次のスクリプトを使用できます。

```
https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

サポートされているオペレーティングシステム：

インストールスクリプトは、以下のLinuxディストリビューションでテストされました。

- デビアン 11
- レッドハット エンタープライズ リナックス (RHEL) 8
- ロッキー・リナックス 8
- ウブントゥ 20/22

このスクリプトを他のLinuxディストリビューションで使用すると、インストールは失敗します。Linux VMの場合、このスクリプトはNVIDIAドライバーのみをインストールします。

a. インストールスクリプトをダウンロードします。

```
curl https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

b. スクリプトに完全なアクセス権を付与します。

```
chmod 777 install_gpu_driver.py
```

c. インストールスクリプトを実行します。

```
python3 install_gpu_driver.py
```

d. gdm3でWaylandを無効にします。

- お使いのディストリビューションのWayland構成ファイルを次のいずれかの場所で見つけます。
  - /etc/gdm3/custom.conf (ウブントゥ)
  - /etc/gdm/custom.conf (CentOS、RHEL、ロッキー・リナックス)
- sudo/root権限でファイルを開きます。

- 行頭の # を削除して、**WaylandEnable=false** のコメントを解除します。
- VMを再起動します。

e. NVIDIA 510以降のドライバーをインストールした場合は、GSPファームウェアを無効にします。

GSPファームウェアが有効になっている場合は、NVIDIAモジュールパラメーター **NVreg_EnableGpuFirmware** を0に設定して無効にします。

このパラメータは、/etc/modprobe.d/nvidia.conf ファイルに次のエントリを追加して設定します。

- オプション nvidia NVreg_EnableGpuFirmware=0
- /etc/modprobe.d/nvidia.conf ファイルが存在しない場合は、作成します。

この手順を完了する際には、次の点に注意してください。

- sudo を使用して、構成ファイルの作成および更新コマンドを実行します。
- VM を再起動するには、Linux ターミナルで **sudo reboot** を使用するか、VM を停止して起動します。

手順 4d: MCS 変数を構成します。

/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とドメインに参加しているシナリオで構成できる MCS 変数です。

- **ドメインに参加していないシナリオの場合**

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` =*path-to-install-dotnet-runtime **

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

`ADD_FIREWALL_RULES` =**Y | N** `VDI_MODE` =**Y | N**

`START_SERVICE` =**Y | N**

```
#########################Linux VDA Configuration#####################################
# Provide Linux VDA configuration information.
# Please refer to Linux VDA Documentation for these settings.
DOTNET_RUNTIME_PATH=/usr/bin
DESKTOP_ENVIRONMENT=gnome
SUPPORT_DDC_AS_CNAME=N
VDA_PORT=80
REGISTER_SERVICE=Y
ADD_FIREWALL_RULES=Y
HDX_3D_PRO=N
VDI_MODE=N
SITE_NAME='<none>'
LDAP_LIST='<none>'
SEARCH_BASE='<none>'
START_SERVICE=Y
```

- **ドメインに参加しているシナリオの場合**

  - `Use_AD_Configuration_Files_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` 変数を構成する必要があります。デフォルト値は N で、これはマスターイメージ上の構成テンプレートが MCS で作成されたマシンの構成ファイルを決定することを意味します。現在実行中の VDA をテンプレート VM として使用するには、値を Y に設定します。それ以外の場合は、N に設定します。
  - `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`: ワークグループ名を、ADで構成したNetBIOS名（大文字と小文字を区別）に設定します。設定しない場合、MCSはマシンの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、**citrix** が正しい選択です。ワークグループ名が正しく設定されていることを確認してください。
  - `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対応表については、「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の自動更新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の自動更新」を参照してください。
- Linux VDA構成変数:

  `DOTNET_RUNTIME_PATH`=*path-to-install-dotnet-runtime **

  `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`=**検索ベースセット | '<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**

手順4e: マスターイメージを作成する

1. (SSSD + RHEL 8.x/9.xまたはRocky Linux 8.x/9.xの場合のみ) `update-crypto-policies --set DEFAULT:AD-SUPPORT`コマンドを実行し、テンプレートVMを再起動します。
2. `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。

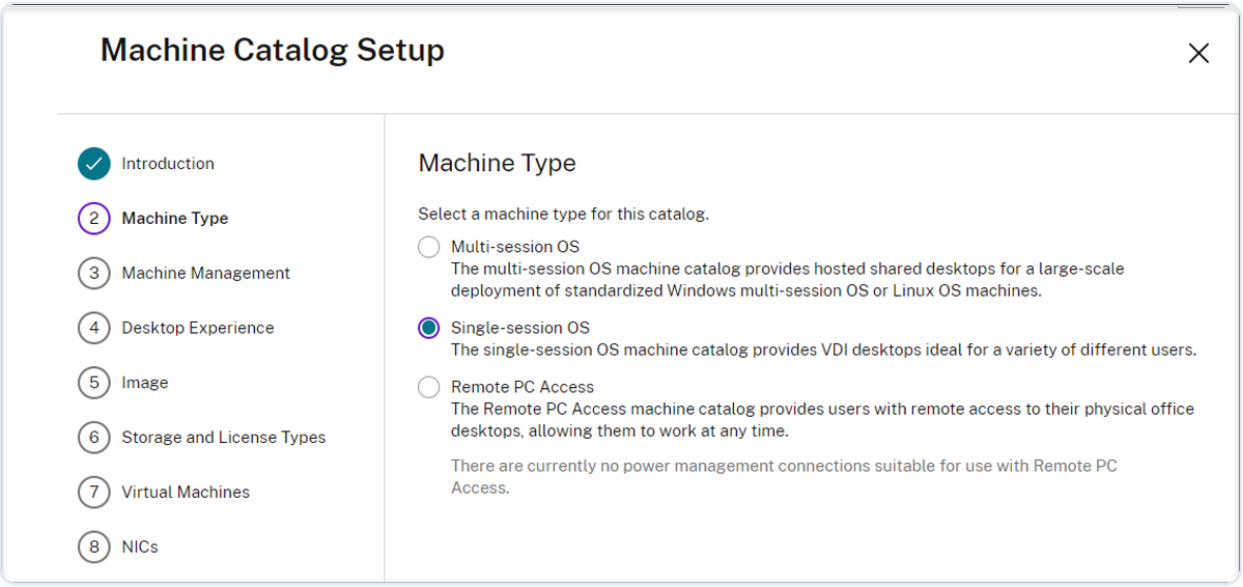
   ドメインに参加していないシナリオの場合、以下のエラーは正常であり、続行を妨げるものではありません。

```
[root@localhost ~]#
[root@localhost ~]# /opt/Citrix/VDA/sbin/deploymcs.sh
Installing Linux VDA dependency packages ...
Installing package redhat-lsb-core
Installing package nautilus
Installing package totem-nautilus
Installing package brasero-nautilus
Installing package pulseaudio
Installing package pulseaudio-module-x11
Installing package pulseaudio-gdm-hooks
Installing package pulseaudio-module-bluetooth
Installing package alsa-plugins-pulseaudio
Installing package pciutils
Installing package openssh
Installing package openssh-clients
Installing package java-11-openjdk
Installing package chrony
Installing package krb5-workstation
Installing package oddjob-mkhomedir
Starting PostgreSQL database ...
Installing package tdb-tools
Installing package ntfs-3g
/opt/Citrix/VDA/lib64/mcs ~
installing mcs systemd unit file: ad_join.service ...
~
/opt/Citrix/VDA/lib64/mcs ~
~
Installing winbind dependency packages ...
Installing package samba-winbind
Installing package samba-winbind-clients
ERROR: Exit funtion conf_dns, dns not configured, please do it manually.
[root@localhost ~]#
```

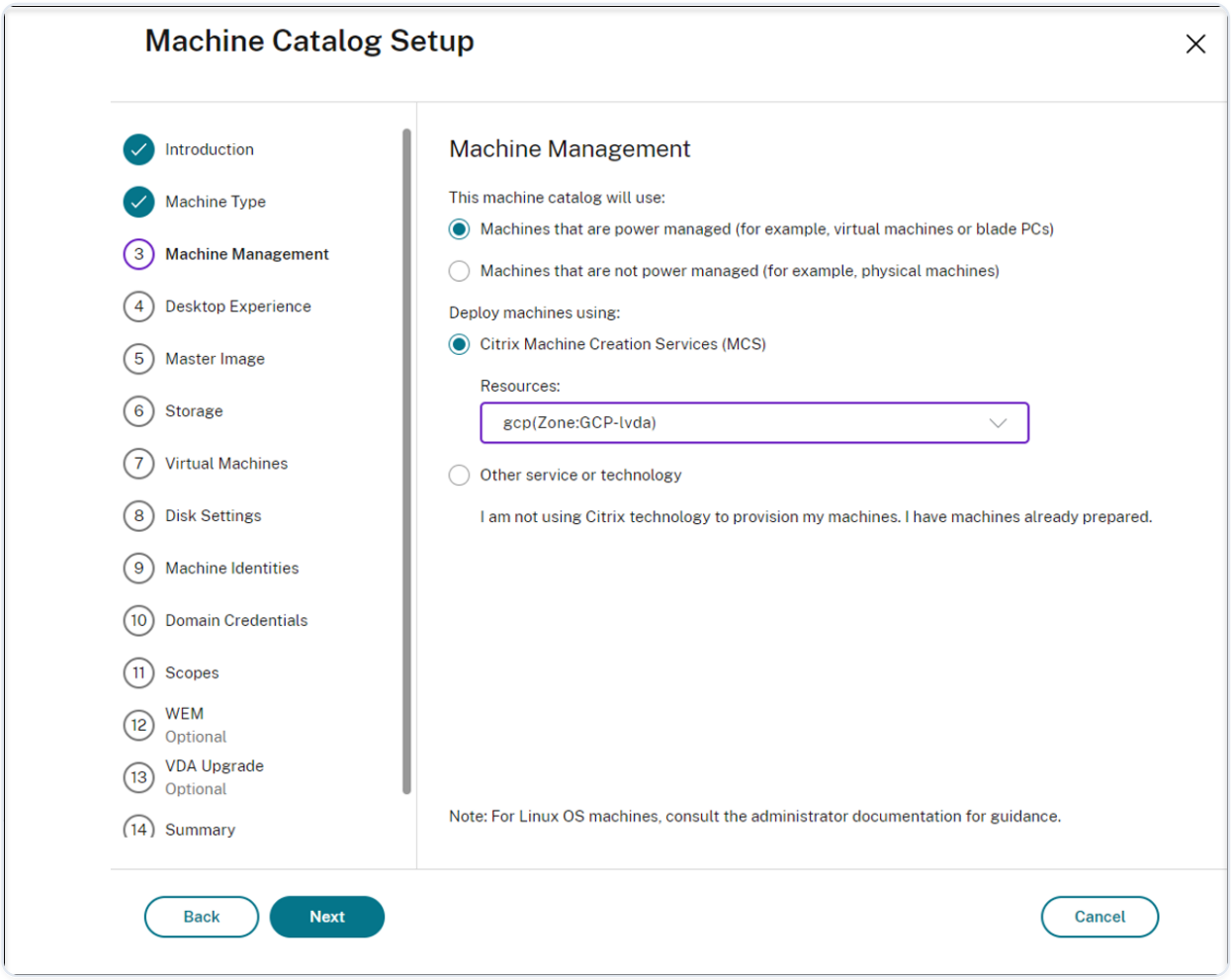
3. テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。

手順5: マシンカタログを作成する

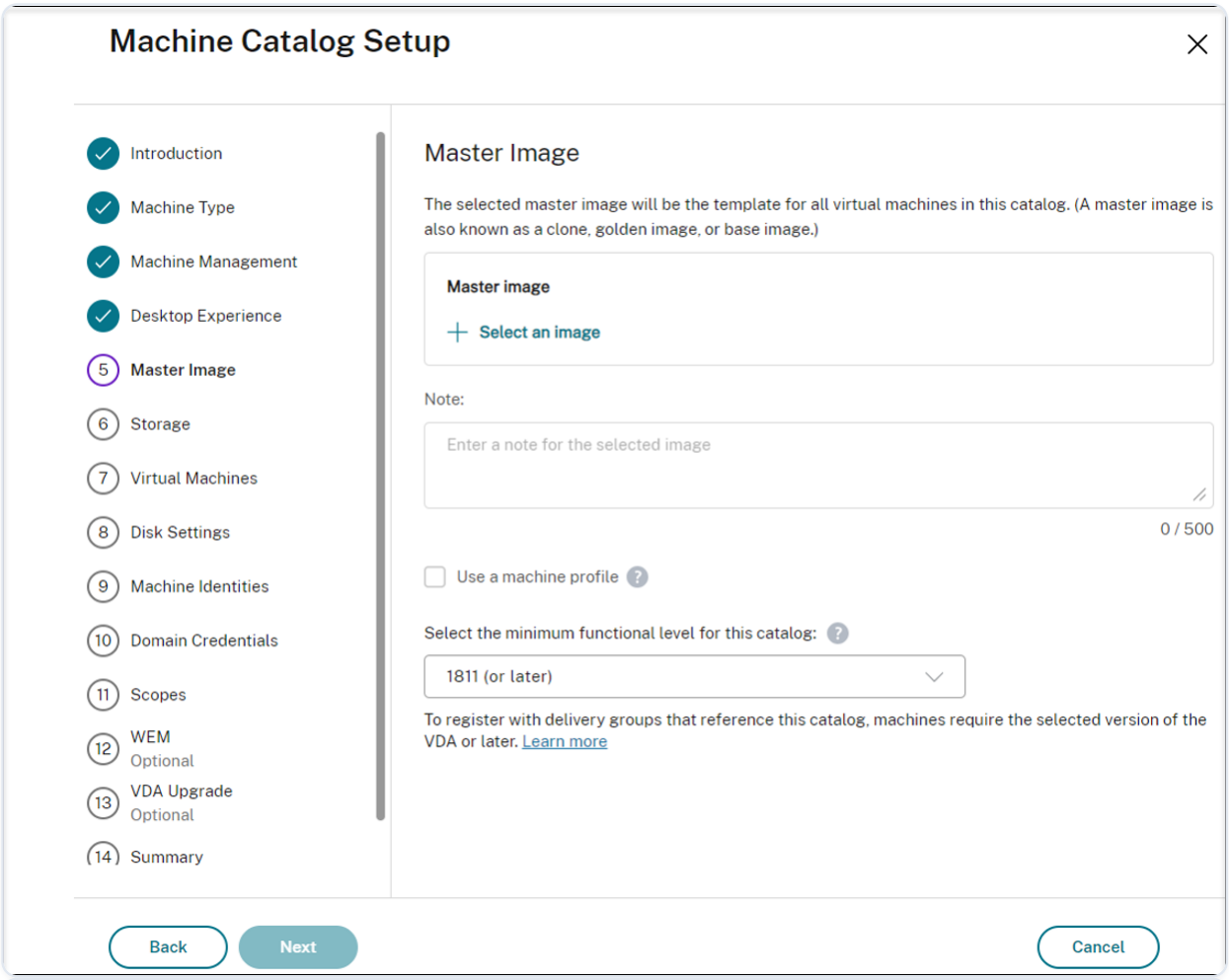
1. Citrix Cloud にサインインします。左上のメニューで、**マイサービス > DaaS** を選択します。
2. **管理 > フル構成**から、左側のペインで**マシンカタログ**を選択します。
3. アクションバーで を選択します。
4. **マシンタイプ**ページで、**マルチセッションOS**または**シングルセッションOS**を選択し、**次へ**を選択します。

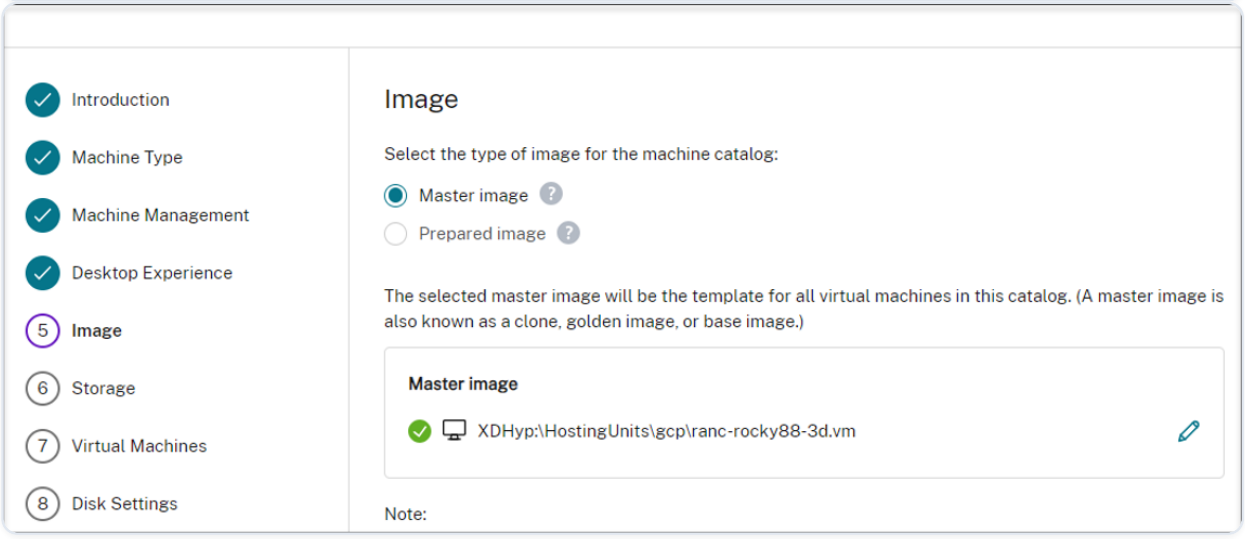


5. **マシン管理**ページで、**電源管理されているマシン**オプションと**Citrix Machine Creation Services**オプションを選択し、**次へ**を選択します。複数のリソースがある場合は、メニューから1つ選択します。



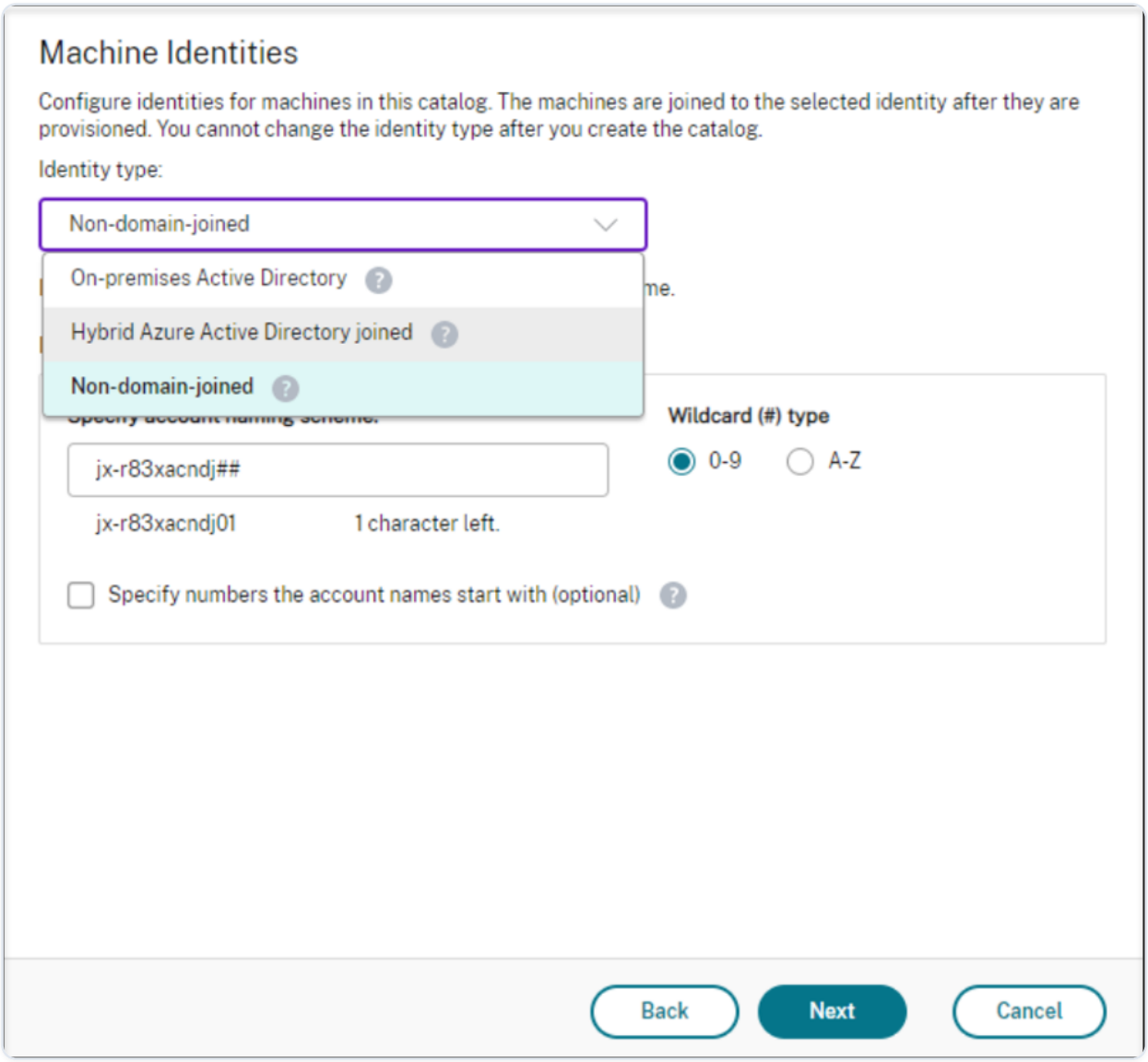
6. **マスターイメージ**ページで、以前に作成したマスターイメージを選択します。



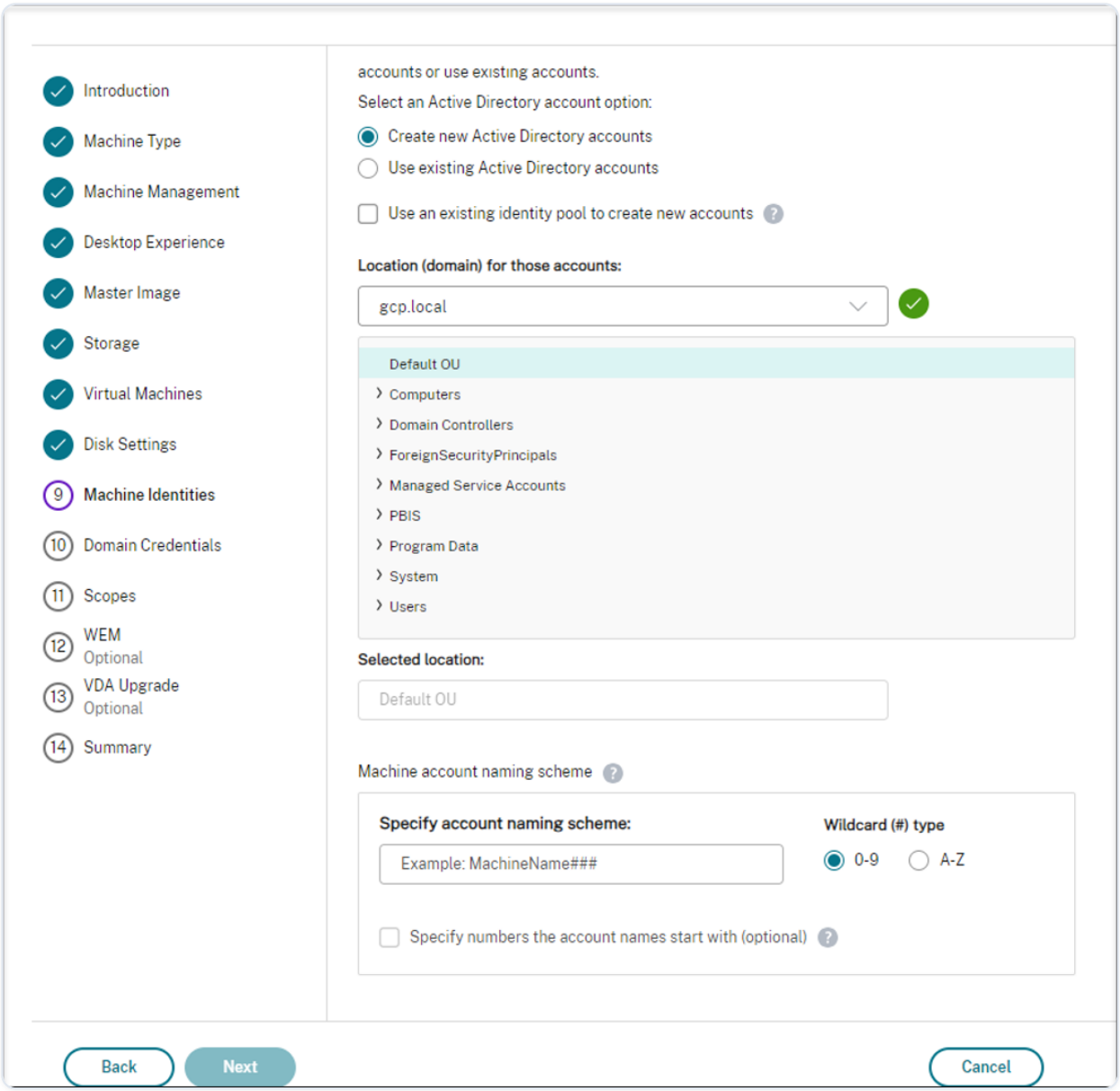


7. **マシンID**ページで、マスターイメージがどのドメインにも参加していない場合は**ドメインに参加していません**を選択するか、マスターイメージをドメインに参加させた場合はActive Directoryアカウントを選択します。

   ドメインに参加していないシナリオの場合:

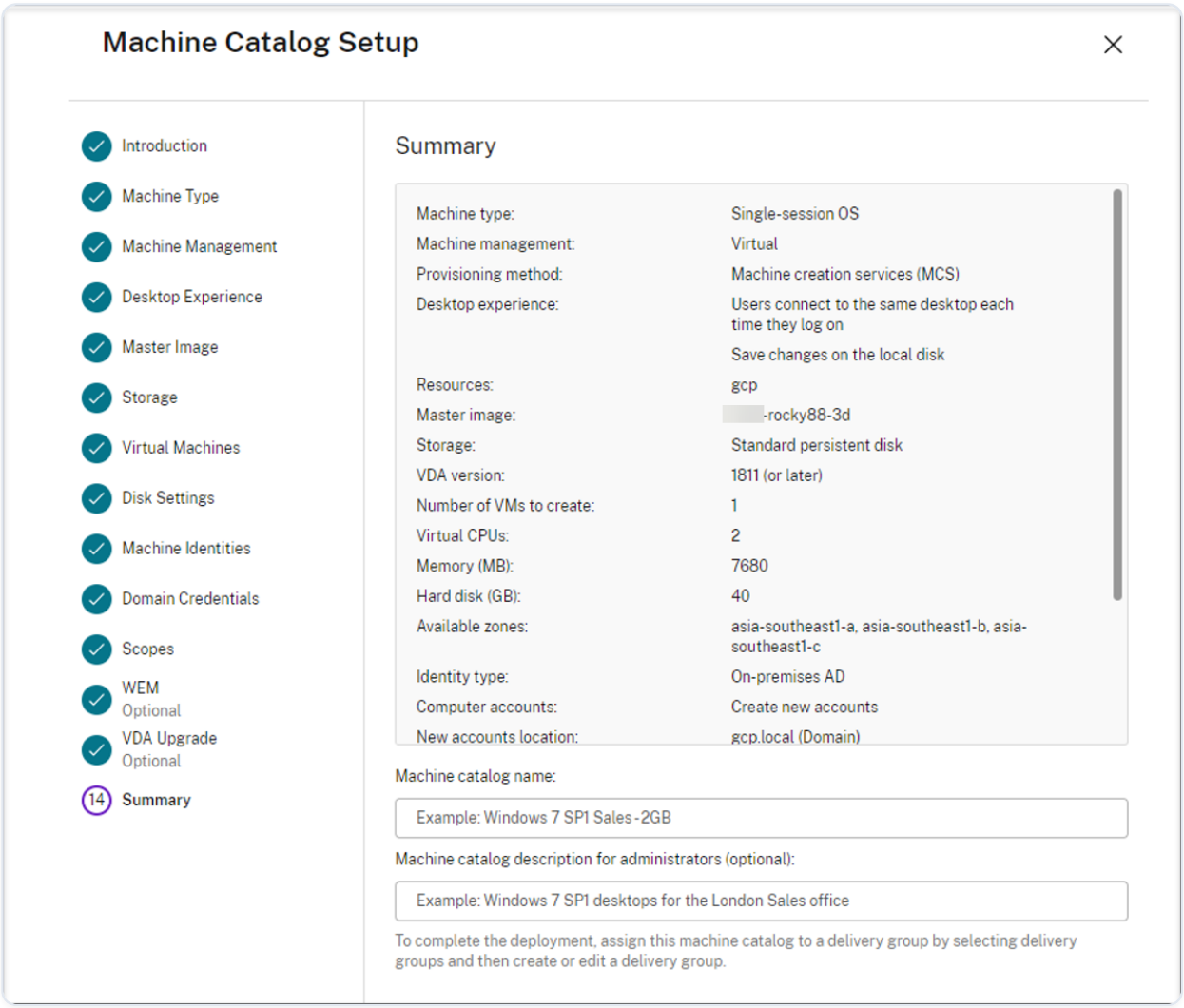


ドメインに参加しているシナリオの場合:



- **新しいActive Directoryアカウントを作成**を選択した場合は、ドメインを選択し、Active Directoryで作成されるプロビジョニングされたVMコンピューターアカウントの命名スキームを表す文字のシーケンスを入力します。アカウントの命名スキームは1〜64文字で、空白スペース、非ASCII文字、または特殊文字を含めることはできません。
- **既存のActive Directoryアカウントを使用**を選択した場合は、**参照**を選択して、選択したマシン用の既存のActive Directoryコンピューターアカウントに移動します。
- **ドメイン資格情報**ページで、**資格情報の入力**を選択し、ユーザー名とパスワードを入力し、**保存**を選択してから**次へ**を選択します。入力する資格情報には、Active Directoryアカウント操作を実行する権限が必要です。

8. 他のページで追加設定を構成します。詳細については、Google Cloud Platformカタログの作成を参照してください。
9. **概要**ページで、情報を確認し、カタログの名前を指定して、**完了**を選択します。



マシンカタログの作成には時間がかかる場合があります。完了すると、カタログが一覧表示されます。Google Cloudコンソールで、マシンがターゲットノードグループに作成されていることを確認できます。

ステップ6：デリバリーグループを作成する

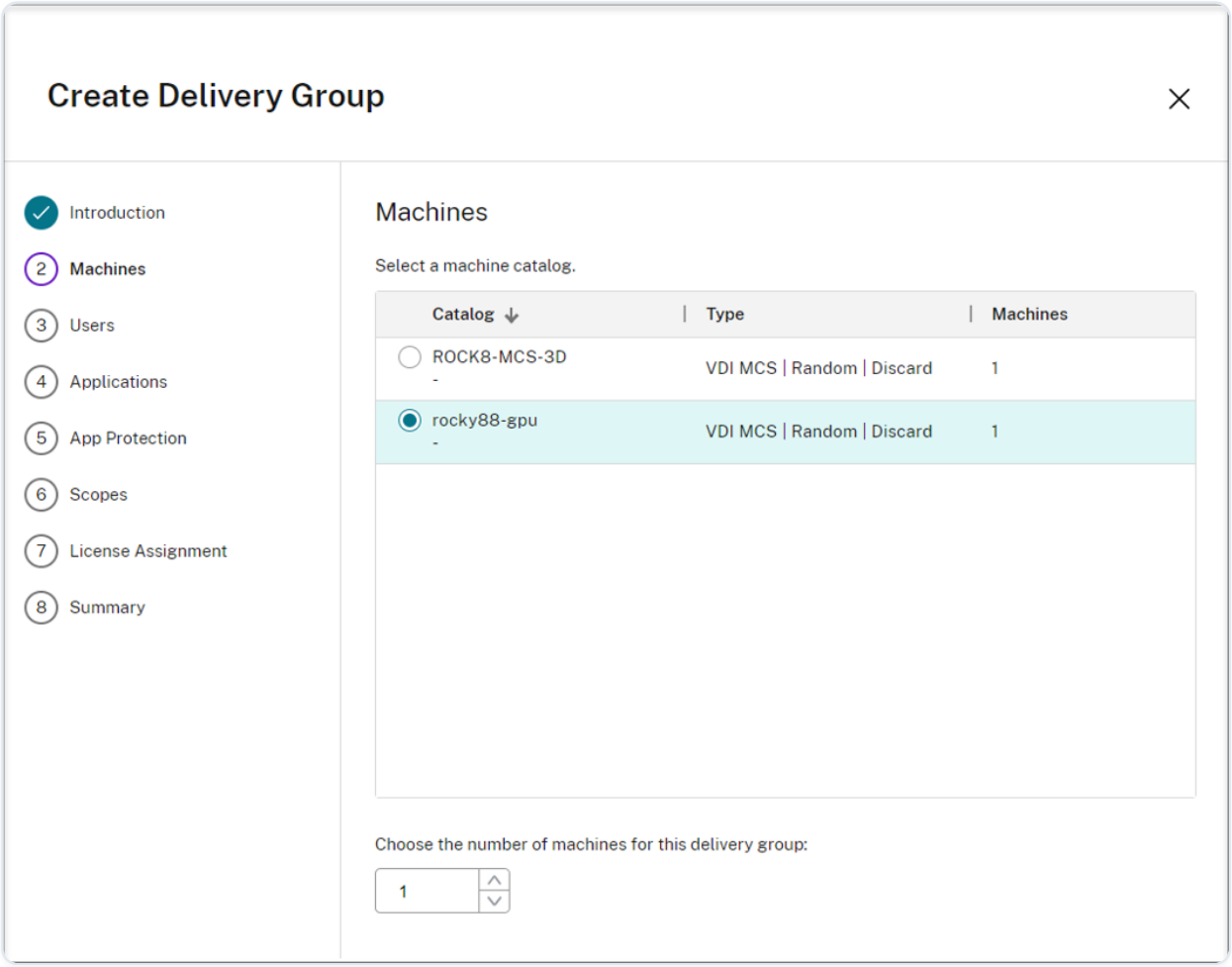
デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。

詳細については、Citrix DaaSのドキュメントでデリバリーグループの作成を参照してください。

1. Citrix Cloudにサインインします。左上のメニューで、**My Services > DaaS**を選択します。
2. **Manage > Full Configuration**から、左ペインで**Delivery Groups**を選択します。
3. アクションバーで**Create Delivery Group**を選択します。デリバリーグループ作成ウィザードが開きます。

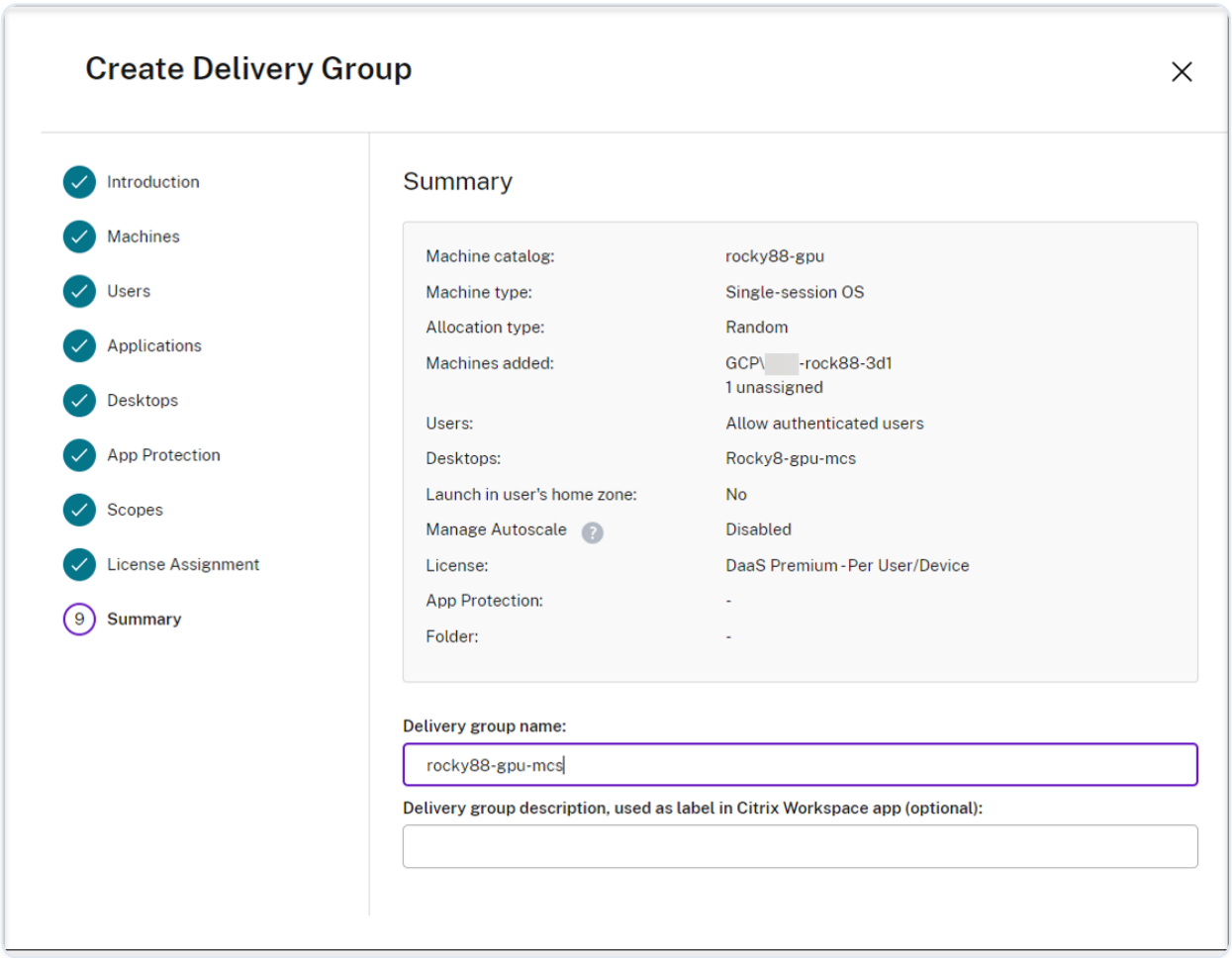
   表示されるウィザードページは、選択内容によって異なる場合があります。

4. **Machines**ページで、マシンカタログを選択し、そのカタログから使用するマシンの数を選択します。



5. 他のページで追加設定を構成します。詳細については、デリバリーグループの作成を参照してください。
6. **Summary**ページで、デリバリーグループの名前を入力します。（オプションで）説明を入力することもできます。これはCitrix Workspace™アプリと**Full Configuration**管理インターフェイスに表示されます。例：

   デリバリーグループの名前を入力します。



Full Configuration 管理インターフェイスの配信グループのリスト:

Full Configuration 管理インターフェイスの配信グループのリスト(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/list-of-delivery-groups.png)

Citrix Workspaceアプリ内の配信されたマシン

Citrix Workspaceアプリ内の配信されたマシン(/en-us/linux-virtual-delivery-agent/2607-ltsr/media/delivered-machine.png)

Ansible を使用して展開を管理する

Ansible は、アプリケーション、構成、および更新を展開全体にわたって展開するプロセスを自動化するのに役立ちます。この記事では、Ansible を使用して展開を効率的に管理するための段階的な手順を説明します。

ステップ 1: 展開するものを決定する

開始する前に、アプリケーション、サービス、構成、環境変数など、展開する必要があるものを特定します。

ステップ 2: Ansible プロジェクトをセットアップする

Ansible プロジェクトのディレクトリ構造を作成します。プレイブックのコンテンツを整理する重要な方法の 1 つは、Ansible の「ロール」機能です。詳細については、Ansible ドキュメントのロールを参照してください。

以下に、参考として 2 つのディレクトリ構造の例を示します。

ディレクトリ構造の例 #1

```
production                # inventory file for production servers

staging                   # inventory file for staging environment



group_vars/

   group1.yml             # here we assign variables to particular groups

   group2.yml

host_vars/

   hostname1.yml          # here we assign variables to particular systems

   hostname2.yml



library/                  # if any custom modules, put them here
(optional)

module_utils/             # if any custom module_utils to support modules,
put them here (optional)

filter_plugins/           # if any custom filter plugins, put them here
(optional)



site.yml                  # master playbook

webservers.yml            # playbook for webserver tier
```

```
dbservers.yml             # playbook for dbserver tier



roles/

    common/               # this hierarchy represents a "role"

        tasks/            #

            main.yml      #  <-- tasks file can include smaller files if
warranted

        handlers/         #

            main.yml      #  <-- handlers file

        templates/        #  <-- files for use with the template resource

            ntp.conf.j2   #  <------- templates end in .j2

        files/            #

            bar.txt       #  <-- files for use with the copy resource

            foo.sh        #  <-- script files for use with the script
resource

        vars/             #

            main.yml      #  <-- variables associated with this role

        defaults/         #

            main.yml      #  <-- default lower priority variables for this
role

        meta/             #

            main.yml      #  <-- role dependencies

        library/          # roles can also include custom modules
```

```
        module_utils/     # roles can also include custom module_utils
        lookup_plugins/   # or other types of plugins, like lookup in this
case



    webtier/              # same kind of structure as "common" was above,
done for the webtier role
    monitoring/           # ""
    fooapp/               # ""
```

ディレクトリ構造の例 #2

```
inventories/
   production/
      hosts               # inventory file for production servers
      group_vars/
         group1.yml       # here we assign variables to particular groups
         group2.yml
      host_vars/
         hostname1.yml    # here we assign variables to particular systems
         hostname2.yml


   staging/
      hosts               # inventory file for staging environment
      group_vars/
```

```
        group1.yml       # here we assign variables to particular groups
        group2.yml
    host_vars/
        stagehost1.yml   # here we assign variables to particular systems
        stagehost2.yml


library/
module_utils/
filter_plugins/


site.yml
webservers.yml
dbservers.yml


roles/
    common/
    webtier/
    monitoring/
    fooapp/
```

ステップ 3: インベントリを構成する

インベントリファイル (inventory.ini) を定義します。インベントリファイルには通常、Ansible を使用して管理するホストが、ホスト名、IP アドレス、グループメンバーシップなどの必要な詳細とともにリストされます。例:

```
# Hostname and ip address

[UBUNTU2004]

<ip address>

[UBUNTU2204]

<ip address>

[RHEL8]

<ip address>

[RHEL9]

<ip address>

[DEBIAN11]

<ip address>

[DEBIAN12]

<ip address>

[SUSE15]

<ip address>



[all:children]

UBUNTU2004

UBUNTU2204
```

```
RHEL8

RHEL9

DEBIAN11

DEBIAN12

SUSE15



[all:vars]

ansible_user=<ansible execute user e.g root>

ansible_password=<>

ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

ステップ 4: Ansible プレイブックを作成する

展開タスクを自動化するためのプレイブック (.yml ファイル) を作成します。このセクションでは、さまざまな展開タスクを自動化するためのプレイブックの例を示します。

Linux ディストリビューションをパッチするためのプレイブックの例

Ansible を使用してさまざまな Linux ディストリビューションにパッチを適用するには、patch-for-different-distribution.yml という名前の YAML プレイブックファイルを作成し、そのファイルに次のようなタスクを設定します。hosts ディレクティブは、プレイブックタスクを実行する対象となるインベントリ (このコンテキストでは inventory.ini) からのターゲットホストを定義します。

```
- name: Upgrade and Reboot RHEL & Debian family Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    reboot_connect_timeout: 5

    reboot_post_reboot_delay: 15
```

```
    reboot_timeout: 600

  tasks:

    # Upgrade RHEL family OS packages

    - name: Upgrade RHEL Family OS packages

      ansible.builtin.yum:

        name: '*'

        state: latest

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"



    # Upgrade RHEL family OS packages

    - name: Upgrade RHEL Family OS packages

      ansible.builtin.yum:

        name: '*'

        state: latest

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "8"



    # Upgrade RHEL family OS packages

    - name: Upgrade RHEL Family OS packages
```

```
    ansible.builtin.yum:
      name: '*'
      state: latest
    when:
      - ansible_facts['distribution'] == "RedHat"
      - ansible_facts['distribution_major_version'] == "9"


# Ubuntu Family upgrade
- name: Update repositories cache
  apt:
    update_cache: yes
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "20"


- name: Update all packages to their latest version
  apt:
    name: "*"
    state: latest
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "22"
```

```
# Debian Family upgrade
- name: Upgrade the OS (apt-get dist-upgrade)
  apt:
    upgrade: dist
  when:
    - ansible_facts['distribution'] == "Debian"
    - ansible_facts['distribution_major_version'] == "11"

- name: Upgrade the OS (apt-get dist-upgrade)
  apt:
    upgrade: dist
  when:
    - ansible_facts['distribution'] == "Debian"
    - ansible_facts['distribution_major_version'] == "12"

# Reboot after upgrade
- name: Reboot host
  reboot:
    connect_timeout: "{{ reboot_connect_timeout }}"
    post_reboot_delay: "{{ reboot_post_reboot_delay }}"
```

```
        reboot_timeout: "{{ reboot_timeout }}"
```

.Net環境をインストールするためのプレイブックの例

以下のプレイブックの例は、特定のLinuxディストリビューションに異なるバージョンの.Net環境をインストールします。

```
- name: Install dotnet runtime environment on Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  tasks:

    # Install dotnet runtime environment on RHEL7

    - name: Enable the rhel-7-server-dotnet-rpms repository

      command: subscription-manager repos --enable=rhel-7-server-dotnet-
rpms

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"



    - name: Install dotnet runtime environment on RHEL7

      ansible.builtin.yum:

        name: rh-dotnet60-aspnetcore-runtime-6.0

        state: present

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"
```

```
- name: Remove /usr/bin/dotnet if it exists
  file:
    path: /usr/bin/dotnet
    state: absent

- name: Create a symbolic link
  file:
    src: /opt/rh/rh-dotnet60/root/usr/lib64/dotnet/dotnet
    dest: /usr/bin/dotnet
    state: link

# RHEL8 linux vda install dotnet runtime environment
- name: Install dotnet-runtime-8.0
  ansible.builtin.dnf:
    name: dotnet-runtime-8.0
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "8"

- name: Install aspnetcore-runtime-8.0
```

```
    ansible.builtin.dnf:
      name: aspnetcore-runtime-8.0
      state: present
    when:
      - ansible_facts['distribution'] == "RedHat"
      - ansible_facts['distribution_major_version'] == "8"


  # RHEL9 linux vda install dotnet runtime environment
  - name: Install dotnet-runtime-8.0
    ansible.builtin.dnf:
      name: dotnet-runtime-8.0
      state: present
    when:
      - ansible_facts['distribution'] == "RedHat"
      - ansible_facts['distribution_major_version'] == "9"


  - name: Install aspnetcore-runtime-8.0
    ansible.builtin.dnf:
      name: aspnetcore-runtime-8.0
      state: present
    when:
      - ansible_facts['distribution'] == "RedHat"
```

```
        - ansible_facts['distribution_major_version'] == "9"



    # Ubuntu20.04 linux vda install dotnet runtime environment

    - name: Register Microsoft key and feed

      shell: |

        wget https://packages.microsoft.com/config/ubuntu/20.04/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb

        dpkg -i packages-microsoft-prod.deb

        rm packages-microsoft-prod.deb

      when:

        - ansible_facts['distribution'] == "Ubuntu"

        - ansible_facts['distribution_major_version'] == "20"



    - name: Install dotnet-runtime-8.0

      ansible.builtin.apt:

        name: dotnet-runtime-8.0

        state: present

        update_cache: yes

      when:

        - ansible_facts['distribution'] == "Ubuntu"

        - ansible_facts['distribution_major_version'] == "20"
```

```
  - name: Install aspnetcore-runtime-8.0
    ansible.builtin.apt:
      name: aspnetcore-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "20"


  # Ubuntu22.04 linux vda install dotnet runtime environment
  - name: Install dotnet-runtime-8.0
    ansible.builtin.apt:
      name: dotnet-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "22"


  - name: Install aspnetcore-runtime-8.0
    ansible.builtin.apt:
      name: aspnetcore-runtime-8.0
```

```
        state: present

        update_cache: yes

      when:

        - ansible_facts['distribution'] == "Ubuntu"

        - ansible_facts['distribution_major_version'] == "22"



    # Debian11 linux vda install dotnet runtime environment

    - name: Register Microsoft key and feed

      shell: |

        wget https://packages.microsoft.com/config/debian/11/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb

        dpkg -i packages-microsoft-prod.deb

        rm packages-microsoft-prod.deb

      when:

        - ansible_facts['distribution'] == "Debian"

        - ansible_facts['distribution_major_version'] == "11"



    - name: Install dotnet-runtime-8.0

      ansible.builtin.apt:

        name: dotnet-runtime-8.0

        state: present

        update_cache: yes
```

```
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "11"


    - name: Install aspnetcore-runtime-8.0
      ansible.builtin.apt:
        name: aspnetcore-runtime-8.0
        state: present
        update_cache: yes
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "11"


    # Debian12 linux vda install dotnet runtime environment
    - name: Register Microsoft key and feed
      shell: |
        wget https://packages.microsoft.com/config/debian/12/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb
        dpkg -i packages-microsoft-prod.deb
        rm packages-microsoft-prod.deb
      when:
        - ansible_facts['distribution'] == "Debian"
```

```
      - ansible_facts['distribution_major_version'] == "12"


  - name: Install dotnet-runtime-8.0
    ansible.builtin.apt:
      name: dotnet-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "Debian"
      - ansible_facts['distribution_major_version'] == "12"


  - name: Install aspnetcore-runtime-8.0
    ansible.builtin.apt:
      name: aspnetcore-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "Debian"
      - ansible_facts['distribution_major_version'] == "12"


  # Sles15 linux vda install dotnet runtime environment
  - name: Register Microsoft key and feed
```

```
      shell: |

        sudo rpm -Uvh https://packages.microsoft.com/config/sles/15/
packages-microsoft-prod.rpm

        sudo ln -s /etc/yum.repos.d/microsoft-prod.repo /etc/zypp/repos.d/
microsoft-prod.repo

      when:

        - ansible_facts['distribution'] == "SLES"

        - ansible_facts['distribution_major_version'] == "15"



    - name: Install dotnet-runtime-8.0

      community.general.zypper:

        name: dotnet-runtime-8.0

        state: present

        update_cache: yes

      when:

        - ansible_facts['distribution'] == "SLES"

        - ansible_facts['distribution_major_version'] == "15"



    - name: Install aspnetcore-runtime-8.0

      community.general.zypper:

        name: aspnetcore-runtime-8.0

        state: present

        update_cache: yes
```

```
    when:
      - ansible_facts['distribution'] == "SLES"
      - ansible_facts['distribution_major_version'] == "15"


  # Amazon2 linux vda install dotnet runtime environment
  - name: Install dotnet-runtime-8.0
    ansible.builtin.yum:
      name: dotnet-runtime-8.0
      state: present
    when:
      - ansible_facts['distribution'] == "Amazon"
      - ansible_facts['distribution_major_version'] == "2"


  - name: Install aspnetcore-runtime-8.0
    ansible.builtin.yum:
      name: aspnetcore-runtime-8.0
      state: present
    when:
      - ansible_facts['distribution'] == "Amazon"
      - ansible_facts['distribution_major_version'] == "2"
```

Linux VDAをアップグレードするためのプレイブックの例

Ansibleを使用してLinux VDAのアップグレードを自動化するには、2つの別々のプレイブックを作成できます。get_the_build.ymlのような1つのプレイブックは、Linux VDAパッケージをターゲットマシン（ホスト）にダウンロードして転送することに特化しています。もう1つのプレイブック、例えばlinux_upgrade.ymlは、以前にダウンロードしたパッケージを使用してターゲットマシン上のLinux VDAをアップグレードするように設計されたタスクを含んでいます。

プレイブックの例 get_the_build.yml

```
- hosts: localhost

  name: Get the latest release build to local

  vars:

    build_url: <linux vda download link>  # replace with your actual
value.

    local_tmp: "/tmp/"  # replace with your actual value.

    remote_tmp: "/tmp/"  # replace with your actual value.

    linuxvda_file_name : "linux vda rpm/deb file name" # replace with your
actual value.

  tasks:

  - name: Download the file

    get_url:

      url: "{{ build_url }}"

      dest: "{{ local_tmp }}"

    tags:

      - get
```

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  name: Copy a file to remote location

  tasks:

  - name: Copy vda to the remote machine

    ansible.builtin.copy:

      src: "{{ local_tmp }}{{ linuxvda_file_name }}"

      dest: "{{ remote_tmp }}"

      remote_src: no

    tags:

      - copy
```

プレイブックの例 linux_upgrade.yml

```
- name: Upgrade Linux VDA and Reboot RHEL & Debian Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    remote_tmp: "/path/to/remote/tmp"  # replace with your actual path

    rhel7_file_name: "rhel7_file.rpm"  # replace with your actual file
name

    rhel8_file_name: "rhel8_file.rpm"  # replace with your actual file
name

    rhel9_file_name: "rhel9_file.rpm"  # replace with your actual file
name

    ubuntu2004_file_name: "ubuntu2004_file.deb"  # replace with your
```

```
actual file name

    ubuntu2204_file_name: "ubuntu2204_file.deb"  # replace with your
actual file name

    debian11_file_name: "debian11_file.deb" # replace with your actual
file name

    debian12_file_name: "debian12_file.deb" # replace with your actual
file name

    suse15_file_name: "suse15_file.deb" # replace with your actual file
name

    amazon2_file_name: "amazon2_file.rpm" # replace with your actual file
name

  tasks:

    # Upgrade RHEL linux vda packages

    - name: Upgrade RHEL7 linux vda packages

      ansible.builtin.yum:

        name: "{{ remote_tmp }}{{ rhel7_file_name }}"

        state: present

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"



    # Upgrade RHEL linux vda packages

    - name: Upgrade RHEL8 linux vda packages

      ansible.builtin.yum:

        name: "{{ remote_tmp }}{{ rhel8_file_name }}"
```

```
      state: present
    when:
      - ansible_facts['distribution'] == "RedHat"
      - ansible_facts['distribution_major_version'] == "8"


  # Upgrade RHEL linux vda packages
  - name: Upgrade RHEL9 linux vda packages
    ansible.builtin.yum:
      name: "{{ remote_tmp }}{{ rhel9_file_name }}"
      state: present
    when:
      - ansible_facts['distribution'] == "RedHat"
      - ansible_facts['distribution_major_version'] == "9"


  # Ubuntu20.04 linux vda upgrade
  - name: Ubuntu20.04 linux vda upgrade
    ansible.builtin.apt:
      deb: "{{ remote_tmp }}{{ ubuntu2004_file_name }}"
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "20"
```

```
- name: Ubuntu22.04 linux vda upgrade
  ansible.builtin.apt:
    deb: "{{ remote_tmp }}{{ ubuntu2204_file_name }}"
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "22"


# Debian Linux VDA upgrade
- name: Debian11 Linux VDA upgrade
  ansible.builtin.apt:
    deb: "{{ remote_tmp }}{{ debian11_file_name }}"
  when:
    - ansible_facts['distribution'] == "Debian"
    - ansible_facts['distribution_major_version'] == "11"


- name: Debian12 Linux VDA upgrade
  ansible.builtin.apt:
    deb: "{{ remote_tmp }}{{ debian12_file_name }}"
  when:
    - ansible_facts['distribution'] == "Debian"
    - ansible_facts['distribution_major_version'] == "12"
```

```
    # Sles15 Linux VDA upgrade
    - name: Sles15 Linux VDA upgrade
      community.general.zypper:
        name: "{{ remote_tmp }}{{ suse15_file_name }}"
        state: present
      when:
        - ansible_facts['distribution'] == "SLES"
        - ansible_facts['distribution_major_version'] == "15"


    # Amazon2 Linux VDA upgrade
    - name: Amazon2 Linux VDA upgrade
      ansible.builtin.yum:
        name: "{{ remote_tmp }}{{ amazon2_file_name }}"
      when:
        - ansible_facts['distribution'] == "Amazon"
        - ansible_facts['distribution_major_version'] == "2"
    # Reboot after upgrade
    - name: Reboot host
      reboot:
        connect_timeout: "{{ reboot_connect_timeout }}"
        post_reboot_delay: "{{ reboot_post_reboot_delay }}"
```

```
        reboot_timeout: "{{ reboot_timeout }}"
```

ネットワークファイルシステム (NFS) サーバーをホームディレクトリとしてマウントするためのプレイブックの例

以下のプレイブックの例は、ターゲットホストにNFSサーバーをホームディレクトリとしてマウントします。

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    nfs_server = <nfsserver ip address> # replace with your actual values

    mount_points = /home/<domain realm>/user1,/home/<domain realm>user2  #
replace with your actual values

    nfs_shares = user1,user2  # replace with your actual values

    owners = user1,user2   # replace with your actual values

    groups = group1,group2  # replace with your actual values

  tasks:

    - name: Enable NFS as home directory

      ansible.builtin.command:

        cmd: "/opt/Citrix/VDA/bin/ctxreg create -k 'HKLM\\System\
\CurrentControlSet\\Control\\Citrix' -t 'REG_DWORD' -v
'CheckUserHomeMountPoint' -d '0x00000001' --force"

      register: result

      failed_when: result.rc != 0

      check_mode: no
```

```
    - name: Mount NFS shares
      ansible.builtin.mount:
        path: "{{ item.0 }}"
        src: "{{ nfs_server }}:{{ item.1 }}"
        fstype: nfs
        opts: rw,nolock
        state: mounted
      loop: "{{ mount_points | zip(nfs_shares) | list }}"


    - name: Set owner, group and mode for NFS client paths
      ansible.builtin.file:
        path: "{{ item.0 }}"
        owner: "{{ item.1 }}"
        group: "{{ item.2 }}"
        mode: "{{ mode }}"
      loop: "{{ mount_points | zip(owners, groups) | list }}"
```

リモートコマンド実行のためのプレイブックの例

レジストリ設定を変更するためのプレイブックの例

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    registry_key: "your_registry_key"      #  E.g. registry_key =
```

```
HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd

    registry_type: "your_registry_type"    #  E.g. registry_type =
REG_DWORD

    registry_value: "your_registry_value"  #  E.g. registry_value =
AdaptiveScalingEnabled

    registry_data: "your_registry_data"    #  E.g. registry_data =
0x00000000

  tasks:

  - name: Execute AdaptiveScaling redirection script

    ansible.builtin.command:

      cmd: "/opt/Citrix/VDA/bin/ctxreg create -k \"{{ registry_key }}\" -t
\"{{ registry_type }}\" -v \"{{ registry_value }}\" -d
\"{{ registry_data }}\" --force"

    register: result

    failed_when: result.rc != 0

    check_mode: no
```

RHELマイナーバージョンをロックするためのプレイブックの例

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    rhel_minor_version: "9.3"  # replace with your actual minor version
such as 9.3, 8.8

  tasks:

    - name: Lock system to a specific minor version

      ansible.builtin.command:
```

```
        cmd: "subscription-manager release --set={{ rhel_minor_version }}"
      register: result
      failed_when: "'Error' in result.stderr"
```

非ドメイン参加Linux VDAとRed Hat IdMの統合

Red Hat Identity Management (IdM) は、Linux環境でIDを管理するために広く採用されているソリューションです。LinuxシステムをActive Directory (AD) に直接参加させるのと比較して、IdMは、特にLinuxのフットプリントが大きいインフラストラクチャにおいて、戦略的および運用上の大きな利点を提供します。

Citrix Linux VDA が **非ドメイン参加 (NDJ)** 展開をサポートするようになったことで、組織はNDJ Linux VDAとIdMの統合を検討できます。このアプローチにより、利用可能な認証方法の範囲が広がり、ID管理の柔軟性が向上します。 ユーザーは、ADドメインの資格情報、またはADとの信頼関係が確立されているIdMドメインの資格情報のいずれかを使用してワークスペースに認証する必要があります。認証されると、**シングルサインオン (SSO)** を介してIdMに参加しているLinuxデスクトップにシームレスにアクセスできます。

以下に、IdMでLinux VDAを構成するための検証済みの統合およびテスト手順の概要を示します。

1. **前提条件と環境設定**

- **ID管理 (IdM) の統合:**
  - **IdMサーバーのインストール:** IdMサーバーは適切にインストールされ、そのサービスが稼働している必要があります。Red Hatドキュメントを参照してください。
- **IdMとADの統合:**
  - **双方向信頼:** 両方のドメインのユーザーが互いのサービスにアクセスできるように、IdMとADの間に双方向の信頼関係を確立する必要があります。Red Hatドキュメントを参照してください。
- **IdMサーバーでの信頼関係の検証:**
  - **IdMユーザー権限の確認:** `getentpasswd idmuser1@idm.example.com`
  - **ADユーザー権限の確認:** `getentpasswd aduser1@domain`

1. **非ドメイン参加Linux VDAマシンの作成**

非ドメイン参加Linux VDAを作成するには、MCSを使用した非ドメイン参加Linux VDAの作成を参照してください。

1. **DDCでのデフォルトログオンタイプの変更**

MCSによって作成されたデリバリーグループのデフォルトの`MachineLogOnType`は`LocalMappedAccount`です。これは、DDCで以下のPowerShellコマンドを実行して`ActiveDirectory`に設定する必要があります。

```
Set-BrokerDesktopGroup -Name "<your delivery group name>" -MachineLogOnType ActiveDirectory
```

1. **IdMクライアントをインストールし、VDAをIdMに参加させる**

- **Linux VDAにIdMクライアントをインストールして構成する:**
  - **IdMクライアントのセットアップ:** IdMサーバーによってマシンを管理できるようにするには、Linux VDAにIdMクライアントをインストールする必要があります。Red Hatドキュメントを参照してください。
  - 非対話型インストールモードの使用をお勧めします。
- **AD/IDMユーザー認証の確認:**
  - **IdMユーザー権限の確認:** `ssh localhost -l idmuser1@idm.example.com`
  - **ADユーザー権限の確認:** `ssh localhost -l admuser1@domain`

1. **セッションを起動して統合を確認する**

- **オンプレミスシナリオの場合:** ADユーザーとIdMユーザーの両方が、IdMと統合されたLinux VDAでセッションを起動できます。
- **DaaSシナリオの場合:** IdMユーザーはADドメインに属するサービスを利用できないため、DaaS環境ではADユーザーのみがIdMと統合されたLinux VDAでセッションを起動できます。

1. **Linux VDAマシンの一括プロビジョニング**

- ステップ2「MCSを介してLinux VDAをさらにプロビジョニングする」を参照してください。
- ステップ4「サードパーティの自動化を活用して各VDAをIdMに参加させる」を参照してください。