Linux 仮想配信エージェント 2308

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) は、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へのアクセスを可能にします。

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、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

2308の新機能

Linux VDA バージョン 2308 には、以下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、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 8.8
- 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

詳しくは、「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。詳しくは、「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。詳しくは、「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。詳しくは、「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、サーバーフェッチおよび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。これにより、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。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/2308/configure/multimedia/browser-content-redirection.html)を参照してください。

EDTの輻輳制御の強化（プレビュー）

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、「アダプティブトランスポート」(/ja-jp/linux-virtual-delivery-agent/2308/configure/session/adaptive-transport.html)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できるようにしました。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

詳しくは、「システムトレイ」(/ja-jp/linux-virtual-delivery-agent/2308/configure/graphics/system-tray.html)を参照してください。

H.265ハードウェアエンコードの強化

H.265ハードウェアエンコードを拡張し、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。詳しくは、「グラフィックの構成と微調整」(/ja-jp/linux-virtual-delivery-agent/2308/configure/graphics/configuring-graphics.html)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の**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。詳しくは、「**XDPing**」(/ja-jp/linux-virtual-delivery-agent/2308/configure/administration/tools-and-utilities.html#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、CentOS 7.9で徹底的にテストされました。Amazon Linux 2、RHEL 7.9、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。

以前のリリースでの新機能

1912 LTSR以降2305 CRまでの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 2308 には、以下の修正された問題が含まれています。

- RHEL 8.x または Rocky Linux 8.x にインストールされた Linux VDA からアプリセッションを起動しようとすると、失敗します。この問題は、VDA が System Security Services Daemon (SSSD) を使用してドメインに接続されており、**/etc/sssd/sssd.conf** ファイルで **default_shell** が **/bin/csh** に設定されている場合に発生します。
- Linux VDA で HDX™ 3D Pro を有効にしてセッションを全画面モードに設定すると、セッションが予期せず終了する場合があります。

既知の問題

このリリースで以下の問題が確認されています。

- バッテリーのないクライアントデバイスを使用して仮想デスクトップセッションにアクセスすると、クライアントのバッテリーステータスがセッションに予期せず表示されることがあります。この問題は、以前バッテリーのあるクライアントデバイスを使用して開いたセッション

に、Citrix Workspace™ app for Mac または Citrix Workspace app for Linux を使用してアクセスした場合に発生します。

- Linux VDA をバージョン 2308 に更新して ctxinstall.sh を再実行すると、ホームフォルダーが変更されます。この問題は、Kerberos レルム名が大文字で設定されており、SSSD がドメイン参加方法として使用されている場合に発生します。この問題を解決するには、ctxinstall.sh に以下の変更を加えます。
  - **get_realm** スクリプト関数で、`realm=$(tr '[:upper:]' '[:lower:]' <<<"${CTX_EASYINSTALL_REALM}")` を `realm="${CTX_EASYINSTALL_REALM}"` に、`realm=$(tr '[:upper:]' '[:lower:]' <<<"$val")` を `realm="$val"` に変更します。
  - **get_netbios_domain** スクリプト関数で、`workgroup=$(tr '[:upper:]' '[:lower:]' <<<"$CTX_EASYINSTALL_NETBIOS_DOMAIN")` を `workgroup="$CTX_EASYINSTALL_NETBIOS_DOMAIN"` に変更します。
- HDX™ 3D Pro が有効になっている場合、拡張モニター上のセッションは真っ暗になり、プライマリモニターのみがセッションを適切に表示します。この問題を解決するには、VDA でターミナルを開き、必要に応じて以下のコマンドを実行します。
  - デュアルモニターの場合、以下を実行します。

    ```
    #sed -i "/UseEDID/a \ \ Option \"ConnectedMonitor\" \"DFP, DFP\"" /etc/X11/ctx-nvidia-2.conf
    ```

  - トリプルモニターの場合、以下を実行します。

    ```
    #sed -i "/UseEDID/a \ \ Option \"ConnectedMonitor\" \"DFP, DFP, DFP\"" /etc/X11/ctx-nvidia-3.conf
    ```

  - クアッドモニターの場合、以下を実行します。

    ```
    #sed -i "/UseEDID/a \ \ Option \"ConnectedMonitor\" \"DFP, DFP, DFP, DFP\"" /etc/X11/ctx-nvidia-4.conf
    ```

- PostgreSQL で設定された最大接続数が同時セッションを処理するのに不十分な場合、セッション起動の失敗が発生します。この問題を回避するには、**postgresql.conf** ファイルの **max_connections** 設定を変更して最大接続数を増やします。

- **/var/log/xdl/jproxy.log** で以下の LDAP 例外がスローされるため、VDA 登録が失敗する可能性があります。

  ```
  javax.naming.NamingException: LDAP response read timed out, timeout
  used: 10000 ms.
  ```

  この問題を回避するには、以下を実行します。

  - LDAP タイムアウト値を変更します。たとえば、以下のコマンドを使用して LDAP タイムアウト値を 60 秒に変更します。

    ```
    ctxreg create -k "HKLM\Software\Citrix\GroupPolicy\Defaults" -t
    "REG_DWORD" -v "LDAPTimeout" -d "0x000EA60" --force
    ```

  - 検索ベースを設定して LDAP クエリを高速化します。ctxsetup.sh の CTX_XDL_SEARCH_BASE 変数を使用するか、以下のコマンドを使用して検索ベースを設定できます。

    ```
    ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t
    "REG_SZ" -v "LDAPComputerSearchBase" -d "<specify a search base
    instead of the root of the domain to improve search performance>" --
    force
    ```

- Microsoft は、2022 年 11 月に Windows 10 用の累積更新プログラム KB5019966 および KB5019964 をリリースしました。これらの更新プログラムにより、ドメイン参加と登録に失敗が発生します。この問題を回避するには、Knowledge Center の記事 CTX474888 を参照してください。
- Kerberos で**RC4_HMAC_MD5**暗号化タイプが許可されている場合、Linux VDAはControllerへの登録に失敗し、次のエラーメッセージが表示されることがあります。

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

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

Active Directory ドメインで **RC4_HMAC_MD5** をグローバルに無効にするには、次の手順を実行します。

a. グループポリシー管理コンソールを開きます。
b. ターゲットドメインを見つけて、**Default Domain Policy**を選択します。
c. **Default Domain Policy**を右クリックし、**Edit**を選択します。グループポリシー管理エディターが開きます。
d. **コンピューターの構成 > ポリシー > Windows の設定 > セキュリティの設定 > ローカル ポリシー > セキュリティ オプション**を選択します。
e. **Network security: Configure encryption types allowed for Kerberos**をダブルクリックします。
f. **DES_CBC_CRC**、**DES_CBC_MD5**、および**RC4_HMAC_MD5**のチェックボックスをオフにし、**AES128_HMAC_SHA1**、**AES256_HMAC_SHA1**、および**Future encryption types**を選択します。

Linux VDAで弱い暗号化タイプを許可するには、次の手順を実行します。

> **注:**
>
> 弱い暗号化タイプを使用すると、展開が攻撃に対して脆弱になります。

a. Linux VDA で /etc/krb5.conf ファイルを開きます。
b. ****セクションの下に次のエントリを追加します。

```
allow_weak_crypto= TRUE
```

- Linux VDAは暗号化にSecureICA 1.0をサポートしていません。Linux VDAでSecureICA 1.0を有効にすると、セッション起動が失敗します。
- ファイルのダウンロード中に予期しないウィンドウが表示されます。このウィンドウはファイルダウンロード機能には影響せず、しばらくすると自動的に消えます。
- SSL暗号化が有効でセッションの信頼性が無効になっている場合、Linux版Citrix Workspaceアプリでセッションを起動できません。
- Ubuntuグラフィック: HDX 3D Proでは、Desktop Viewerのサイズ変更後にアプリケーションの周囲に黒いフレームが表示されたり、背景が黒く表示されたりする場合があります。
- Linux VDAの印刷リダイレクトによって作成されたプリンターは、セッションからログアウトしても削除されない場合があります。

- ディレクトリに多数のファイルやサブディレクトリが含まれている場合、CDMファイルが不足します。この問題は、クライアント側にファイルやディレクトリが多すぎる場合に発生する可能性があります。
- 英語以外の言語では、UTF-8エンコーディングのみがサポートされています。
- Android版Citrix WorkspaceアプリのCAPS LOCKの状態は、セッションローミング中に反転する場合があります。既存の接続をAndroid版Citrix Workspaceアプリにローミングすると、CAPS LOCKの状態が失われる可能性があります。回避策として、拡張キーボードのShiftキーを使用して大文字と小文字を切り替えます。
- Mac版Citrix Workspaceアプリを使用してLinux VDAに接続すると、ALTキーを含むショートカットキーが常に機能するとは限りません。Mac版Citrix Workspaceアプリは、デフォルトで左右のOptions/Altキーの両方にAltGrを送信します。この動作はCitrix Workspaceアプリの設定内で変更できますが、結果はアプリケーションによって異なります。
- Linux VDAがドメインに再参加すると、登録に失敗します。再参加により、新しいKerberosキーセットが生成されます。しかし、Brokerは以前のKerberosキーセットに基づいた、キャッシュされた古いVDAサービスチケットを使用する可能性があります。VDAがBrokerへの接続を試みると、BrokerはVDAへの戻りのセキュリティコンテキストを確立できない場合があります。通常の症状は、VDA登録の失敗です。

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

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

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

  このコマンドは、Citrix Broker Serviceが実行されているNetwork Serviceプリンシパルによって保持されているLSAキャッシュ内のすべてのサービスチケットをパージします。これにより、他のVDAや、場合によっては他のサービスのサービスチケットも削除されます。ただし、これは無害です。これらのサービスチケットは、必要に応じてKDCから再取得できます。
- オーディオのプラグアンドプレイはサポートされていません。ICA®セッションでオーディオの録音を開始する前に、オーディオキャプチャデバイスをクライアントマシンに接続できます。オーディオ録音アプリケーションの起動後にキャプチャデバイスが接続された場合、アプリケーションが応答しなくなり、再起動する必要があります。録音中にキャプチャデバイスが取り外された場合も、同様の問題が発生する可能性があります。

- Windows版Citrix Workspaceアプリでは、オーディオ録音中に音声の歪みが発生する場合があります。

サードパーティ通知

Linux Virtual Delivery Agent バージョン 2308(/ja-jp/linux-virtual-delivery-agent/2308/downloads/lvda-v2308-third-party-notices.pdf) (PDFダウンロード)

このLinux VDAリリースには、ドキュメントに定義されている条項に基づいてライセンス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

この記事の発表は、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、顧客の使用状況とフィードバックを監視しています。発表は、その後のリリースで変更される可能性があり、すべての非推奨の機能や機能が含まれているとは限りません。 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、またはサポートされなくなっています。

項目	非推奨が発表された時期	削除時期
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

項目	非推奨が発表された時期	削除時期
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
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

項目	非推奨が発表された時期	削除時期
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2305 CRまでのリリースに含まれる新機能に関する情報を提供します。

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が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**を編集することで、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関する詳細は、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の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/非表示を切り替えられるようになり、誤った入力を簡単に特定できるようになりました。
- 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、認証ダイアログが個別に表示されます。これにより、資格情報が誤って入力された場合にユーザーは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

詳細については、セッションログオン画面で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で利用可能な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設定に役立つ2つのレジストリキーが追加されました。

- CheckUserHomeMountPoint
- CheckMountPointRetryTime

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。

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を使用してウェブ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。詳細については、FIDO2 (プレビュー)を参照してください。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない可能性があり、非実稼働環境での使用が推奨されます。プレビュー機能で発見された問題は、Citrixテクニカルサポートではサポートされません。

HDX 3D PRO向け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。

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るようになりました。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。

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目的で、**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することで、トラブルシューティングを容易にすることができます。詳細については、**XDPing**を参照してください。

SSSDによるMCSを使用したRHEL 8.x/9.xおよびRocky Linux 8.x/9.x VDAの作成のサポート

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ており、簡易インストールは使用されていません。簡易インストールはRHEL 8.x/9.xおよびRocky Linux 8.x/9.xに**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サーバーが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。
>
> 詳しくは、「MCSを使用したLinux VDAの作成」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

詳しくは、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。

2303の新機能

Linux VDAのバージョン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とRocky Linux 9.1が追加されました。詳しくは、「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ストリーミングを以下のディストリビューションに拡張しました。

- RHEL 9.1
- RHEL 9.0
- ロッキー・リナックス 9.1
- ロッキー・リナックス 9.0

詳細については、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。

簡単インストール機能の強化

簡単インストール機能を強化し、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。主な強化点は以下のとおりです。

- 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、同期するための簡単インストールの構成ファイルとして、**/opt/Citrix/VDA/sbin/ctxinstall.conf**を追加しました。
- 堅牢性を向上させるために、変数構成の検証を追加しました。
- 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。

簡単インストールの使用方法の詳細については、簡単インストールを使用したドメイン参加済みVDAの作成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登録ステータスチェックを実行するには、**sudo /opt/Citrix/VDA/bin/xdping -a**コマンドを実行します。

また、**XDPing**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、**sudo /opt/Citrix/VDA/bin/xdping -T dependency,kerberos,pam**コマンドを実行します。

詳しくは、**XDPing**、フェデレーション認証サービス、およびスマートカードを参照してください。

2301の新機能

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

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

> **注:**
>
> このリリースからSUSE 15.3は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、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。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない可能性があり、非実稼働環境での使用が推奨されます。プレビュー機能で発見された問題は、Citrixテクニカルサポートの対象外です。

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。

動的な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。こ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を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のバージョン2212には、以下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関して、以下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です。
- USBデバイスリダイレクトを、**TD-RDF5A Transcend USBデバイス**や複合USBデバイスの**Yubico YubiKey OTP+FIDO+CCID**など、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の特定の非仮想化GPUで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認証のためにpam_krb5をSSSDに置き換える

以前は、RHEL 8およびRocky Linux 8でスマートカードまたはフェデレーション認証サービス（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。詳細については、「フェデレーション認証サービス」および「スマートカード」を参照してください。

MCS構成を含める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ます。GUIで**展開**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。詳細については、「手順3j：マスターイメージの作成」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。詳細については、「セッション記録（プレビュー）」を参照してください。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない場合があり、非実稼働環境での使用が推奨されます。プレビュー機能で見つかった問題は、Citrixテクニカルサポートではサポートされません。

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。これを行うには、次の手順を完了します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略します。
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基づいて、次のいずれかのスクリプトを実行します。

- 簡易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モードでのみSQLite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、手動でインストールする必要があります。簡易インストールおよびMCSの場合、SQLiteとPostgreSQL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`を使用して、PostgreSQLのポート番号を構成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ド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、グラフィックの構成と微調整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

1回のコピーアンドペースト操作で、ユーザーがクライアントと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 Linux 8.6
- ウブントゥ 22.04

詳細については、Citrix Provisioning のドキュメントの「Streaming Linux target devices」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機能強化し、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

簡単インストールでも、特定のドメイン参加方法に関連する構成テンプレートが使用されます。

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でドメイン参加したUbuntuおよびSUSE VDAのフェデレーション認証サポート

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。

これらのVDAにログオンするユーザーの認証にFASを使用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

GPGを使用してLinux VDAパッケージに署名しました。これにより、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ます。

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、公開鍵を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、公開鍵を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、以下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。このリリース以降、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。詳しくは、「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が追加され、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なりました。たとえば、**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>**に設定して、PNGウォーターマークを作成できます。詳しくは、「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て、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＆ペーストが利用可能

以前は、**ファイルコピー＆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。このリリース以降、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。詳しくは、「ファイルコピー＆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、以下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

サポート対象の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資格情報を使用した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 を使用した認証」を参照してください。

クライアントのバッテリー状態表示

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状態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、「クライアントのバッテリー状態表示」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Remote PC Access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Remote PC Access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で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で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値はクライアント側の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側のFASポートをカスタマイズできます。詳しくは、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。

選択的コーデックに拡張されたハードウェアエンコード

以前は画面全体でのみ利用可能だったH.264ハードウェアエンコードは、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のビデオ圧縮消費がハードウェアにオフロードされ、画質と1秒あたりのフレーム数（FPS）が向上します。この機能を有効にする方法については、「グラフィックの構成と微調整」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

このリリース以降、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。詳しくは、「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。詳しくは、「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。詳しくは、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、以下の新機能と機能強化が含まれています。

Rendezvous V2の完全サポート

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

考慮すべきトラフィックには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ためのすべての制御トラフィックをプロキシするには、依然としてCloud Connectorが必要です。

標準のADドメイン参加マシンと非ドメイン参加マシンは、シングルセッションおよびマルチセッションLinux VDAでRendezvous V2を使用するためにサポートされています。非ドメイン参加マシンでは、Rendezvous V2によりHDXトラフィックと制御トラフィックの両方がCloud Connectorをバイパスできます。

詳しくは、「Rendezvous V2」を参照してください。

簡単インストール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、以下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「**対話型モード**」セクションを参照してください。

HDXグラフィックの改善

Thinwire は、特定のタスクを並列化することで、全体的な CPU 消費量がわずかに増加するオーバーヘッドを伴いますが、フレーム/秒 (FPS) の数を向上させることができるようになりました。この機能はデフォルトで無効になっています。詳細については、「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 バージョン 2201 には、以下の新機能と機能強化が含まれています。

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

SUSE 15.3 および SUSE 15.2 セッションのデフォルトデスクトップとして MATE を指定できるようになりました。デスクトップ環境の詳細については、インストールに関する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すべてのサポート対象ディストリビューションに拡張

セッションシャドウ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 アプリへのログオン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

/opt/Citrix/VDA/sbin ディレクトリにある `enable_vdassl.sh` スクリプトを使用すると、TLS を有効にし、最小 TLS バージョンを設定できます。詳細については、「TLS を使用したユーザーセッションの保護」を参照してください。

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

2112 の新機能

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

Amazon Linux 2 のサポート (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。詳しくは、「システム要件」を参照してください。

HDX画面共有の完全サポート

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

HTTPプロキシを介したRendezvous接続

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

Rendezvousプロキシ認証

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

アダプティブオーディオのサポート

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

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングでは、次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 リナックス エンタープライズ サーバー 15 SP2 (SUSE 15.2)
- スーゼ リナックス エンタープライズ サーバー 15 SP3 (SUSE 15.3)

詳しくは、Citrix Provisioningのドキュメントの「Linuxターゲットデバイスのストリーミング」を参照してください。

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

サポート対象の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でのHDX Webカメラビデオ圧縮のサポート

SUSE 15.3、SUSE 15.2、およびSUSE 12.5セッションで実行されているビデオ会議アプリケーションのユーザーは、HDX Webカメラビデオ圧縮を使用してWebカメラを使用できるようになりました。詳しくは、「HDX Webカメラビデオ圧縮」を参照してください。

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

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

ctxfascfg.shに2つの環境変数が追加されました

ctxfascfg.shに以下の変数を追加しました。

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

詳しくは、「ctxfascfg.shを実行する」を参照してください。

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

Citrix Workspaceアプリにログオンした後、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、次の新機能と機能強化が含まれています。

HDX画面共有のサポート（プレビュー）

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

Linux VDA 用 シトリックス バーチャルチャネル 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ドメインに参加している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、スクリプトの指示に従ってKey Distribution Center (KDC)のホスト名を入力します。詳しくは、「フェデレーション認証サービスを構成する」(/ja-jp/linux-virtual-delivery-agent/2109/configuration/federated-authentication-service.html)を参照してください。

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

Citrix Provisioning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」(/ja-jp/linux-virtual-delivery-agent/2109/installation-overview/use-pvs-to-create-linux-vms.html)を参照してください。

MATEデスクトップのキーボードレイアウト同期サポート

キーボードレイアウト同期サポートをMATEデスクトップに拡張しました。クライアントデバイスのキーボードレイアウトが変更されると、MATEデスクトップ環境をインストールしたVDAのレイアウトもそれに従います。詳しくは、「動的なキーボードレイアウト同期」(/ja-jp/linux-virtual-delivery-agent/2109/configuration/dynamic-keyboard-layout-synchronization.html)を参照してください。

セッションデータクエリユーティリティ

各Linux VDAのセッションデータをクエリ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クエリするには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]`コマンドを実行します。`[-c]`引数は、1秒ごとにデータをクエリすることを意味します。

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

新しいLinuxセッションメトリックが利用可能になりました

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

- ログオン期間

  ログオン期間は、ユーザーがCitrix Workspaceアプリから接続してからセッションが使用可能になるまでのログオンプロセスを測定するものです。このメトリックは、Citrix DaaSの

Monitorタブで確認できます。Monitorは、Citrix Virtual Apps and DesktopsのCurrent ReleaseおよびLTSR展開を監視およびトラブルシューティングするためのDirectorコンソールとして利用できます。

- セッションの自動再接続回数

  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**ポリシーが有効な場合に有効になります。セッションの再接続の詳細については、セッションを参照してください。ポリシーの詳細については、クライアントの自動再接続ポリシー設定およびセッションの信頼性ポリシー設定を参照してください。

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

2107の新機能

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

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

USBリダイレクトのカーネルモジュール再構築の簡素化

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

Debian のスマートカードサポート

Linux 仮想デスクトップセッションにログオンする際、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、Debian 10.7 セッションでスマートカードを

使用できるようになりました。詳細については、「スマートカードのサポート」を参照してください。

2106 の新機能

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

Linux VDA のセルフアップデート

この機能は、Linux VDA ソフトウェアを即時またはスケジュールされた時間に自動的に更新するのに役立ちます。Azure の VM に管理者権限がない場合に役立ちます。詳細については、「Linux VDA のセルフアップデート」を参照してください。

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

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

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

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

HDX 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ーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。この機能はデフォルトで有効になっています。可能な場合は常に HDX Webカメラビデオ圧縮を使用することをお勧めします。詳細については、「HDX Webカメラビデオ圧縮」を参照してください。

グラフィック品質スライダー

仮想Linuxセッションで実行されるグラフィックステータスインジケーターツールに、グラフィック品質スライダーを追加しました。このスライダーは、画質と対話性の適切なバランスを見つけるのに役立ちます。詳しくは、グラフィック品質スライダーを参照してください。

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

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

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

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

RHEL 8.3およびUbuntu 18.04のLinuxストリーミングが完全にサポートされるようになりました。Citrix ProvisioningでLinuxストリーミング機能を使用すると、Citrix Virtual Apps and Desktops環境にLinux仮想デスクトップを直接プロビジョニングできます。詳しくは、Citrix Provisioningを使用してLinux VMを作成するを参照してください。

2104の新機能

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

Citrix DaaS™におけるドメイン非参加Linux VDAのサポート

MCSを使用して、Citrix DaaS（旧称Citrix Virtual Apps and Desktopsサービス）でドメイン非参加Linux VDAを作成できるようになりました。

OpenJDK 11が必須に

Linux VDAで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。

その他のサポート対象ディストリビューションでは、Linux VDAのインストール時にOpenJDK 11が依存関係として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で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、「簡単なインストール」および「RHEL/CentOS用Linux Virtual Delivery Agentのインストール」などの手動インストールに関する記事を参照してください。

RHEL 8、CentOS 8、SUSE 12.5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、以下の新機能と機能強化が含まれています。

Debian 10.7およびCentOS 8.3のサポート

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

SUSEマシンをWindowsドメインに参加させるためのSSSDサポート

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

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

Citrix Virtual Apps and Desktops Standard for Azureでは、ドメイン参加済みおよび非ドメイン参加済みの両方のLinux VDAを作成するための単一イメージを提供するようになりました。この機能により、イメージの準備とメンテナンスが簡素化されます。詳細については、「Create Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure」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、ファイル転送機能が強化されました。また、一度に転送できるファイルの数を10個から100個に増やしました。詳細については、「ファイル転送」を参照してください。

Citrix Studioでのセッション接続タイマー構成のサポート

以前は、アイドルセッションと切断されたセッションの時間制限は、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ています。

- **セッションアイドルタイマー**: アイドルセッション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**で、設定された時間内にアクティブなセッションがユーザー入力を受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

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

各Linux VMの新しいメトリック:

- CPUコア数
- メモリサイズ
- ハードディスク容量
- 現在および過去のCPUとメモリの使用率

各Linuxセッションの新しいメトリック:

- アイドル時間

詳細については、Citrix DirectorでLinux VMとLinuxセッションを監視するを参照してください。

Linux VDAのFAS機能強化

ctxfascfg.shスクリプトを実行する際に、より詳細なログ出力を提供し、ルート証明書とすべての中間証明書の両方を含むパスを指定できるようになりました。構成情報については、フェデレーション認証サービスを構成する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ストリーミングを参照してください。

2012の新機能

Linux VDAのバージョン2012には、次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

サポート対象のディストリビューションとして RHEL 7.9 と RHEL 8.3 を追加しました。

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™ をバイパスして、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 を作成できるようになりました。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。

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

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。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、写真やスクリーンショットを使用してデータを盗む人々への抑止力として、セッションデスクトップに表示されます。詳細については、「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 におけるドメイン参加していない Linux VDA のサポート

Citrix Virtual Apps and Desktops Standard for Azure でドメインに参加していない Linux VDA を作成し、Microsoft Azure から任意のデバイスに仮想アプリとデスクトップを配信できるようになりました。詳細については、「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

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 の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、**Client Keyboard Layout Sync and IME Improvement** 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経由で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/2308/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＆ペーストのサポート

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＆ペーストできるようになりました。コピー＆ペースト機能は、右クリックメニューまたはキーボードショートカットを使用して実行できます。詳細については、「ファイルのコピーとペースト」を参照してください。

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

2003の新機能

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

RHEL 8 のサポート

このリリースから、Citrix は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。これで、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。詳細については、「HDX Insight」を参照してください。

自動ダウンロード

Citrix は、ファイル転送の機能強化として自動ダウンロードを追加しました。これは、HTML5 用 Citrix Workspace アプリおよび Chrome 用 Citrix Workspace アプリで利用できます。この機能強化により、VDA 上に **Save to My Device** ディレクトリが提供されます。**Save to My Device** ディレクトリにダウンロードまたは移動したファイルは、クライアントデバイスに自動的に転送されます。詳細については、「ファイル転送」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。

システム要件

Linux VDAの最新リリースは、Citrix Virtual Apps and Desktopsと連携しています。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に関する情報、およびCitrixが特定のバージョンの製品のサポートを終了する時期については、Citrix製品ライフサイクルマトリックスを参照してください。

Linux VDAの構成プロセスは、Windows VDAとは若干異なります。どのDelivery Controller™ファームでも、WindowsとLinuxの両方のデスクトップをブローカーできます。

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に記載されています。

長期サービス（LTSR）環境での最新リリース（CR）の使用に関する情報、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マトリックスについては、次の表を参照してください。詳細については、XorgModuleABIVersionsを参照してください。

LINUXディストリビューション	XORGバージョン	サポートされているデスクトップ
アマゾン リナックス 2	1.20	グノーム、グノームクラシック、マテ
デビアン 11.3	1.20	グノーム、グノームクラシック、ケーディーイー、マテ
RHEL 9.2/9.0	1.20	GNOME
RHEL 8.8/8.6	1.20	グノーム、グノーム クラシック、マテ

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 7.9, CentOS 7.9	1.20	グノーム、グノーム クラシック、ケーディーイー、マテ
ロッキー Linux 9.2/9.0	1.20	GNOME
ロッキー Linux 8.8/8.6	1.20	グノーム、グノームクラシック、ケーディーイー、マテ
SUSE 15.4	1.20	グノーム、グノーム クラシック、マテ
ウブントゥ 22.04	1.21	グノーム、グノーム クラシック、ケーディーイー、マテ
ウブントゥ 20.04	1.20	グノーム、グノーム クラシック、ケーディーイー、マテ

注:

- OSベンダーからのサポートが期限切れになると、Citrixは問題の修復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」(/ja-jp/linux-virtual-delivery-agent/2308/whats-new/removed-features.html)を参照してください。
- 少なくとも1つのデスクトップがインストールされている必要があります。セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境は、ctxinstall.shまたはctxsetup.shスクリプトを介して指定できます。
- 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。
- Ubuntuでは`HWE kernel`または`HWE Xorg`を使用しないでください。
- ユーザー名の形式は、現在のディスプレイマネージャーの`systemd`構文規則に準拠している必要があります。`systemd`ユーザー名構文の詳細については、ユーザー/グループ名の構文を参照してください。

サポートされているホストプラットフォームと仮想化環境

- ベアメタルサーバー
- アマゾン ウェブ サービス (AWS)
- シトリックス ハイパーバイザー™
- グーグル クラウド プラットフォーム (GCP)
- カーネルベース仮想マシン (KVM)
- マイクロソフト アジュール
- マイクロソフト Hyper-V
- VMware vSphere ハイパーバイザー
- ニュータニックス AHV

注:

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

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

Active Directory統合パッケージ

Linux VDAは、次のActive Directory統合パッケージ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
アマゾン Linux 2	はい	はい	はい	はい	いいえ
デビアン 11.3	はい	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	クエスト
RHEL 9.2/9.0、ロッキーリナックス 9.2/9.0/8.8/8.6	はい	はい	いいえ	いいえ	はい (Quest v4.1 以降)
RHEL 8.8/8.6	はい	はい	はい	はい	はい (Quest v4.1 以降)
RHEL 7.9, CentOS 7.9	はい	はい	はい	はい	はい (Quest v4.1 以降)
SUSE 15.4	はい	はい	はい	はい	いいえ
ウブントゥ 22.04/20.04	はい	はい	はい	はい	はい (Quest v4.1 以降)

HDX™ 3D プロ

Citrix Virtual Apps and Desktops™ の HDX 3D Pro を使用すると、ハードウェアアクセラレーションにグラフィック処理ユニット (GPU) を使用して最高のパフォーマンスを発揮するデスクトップとアプリケーションを配信できます。

ハイパーバイザー

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

- シトリックス 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)
- グーグル クラウド プラットフォーム (GCP)

> **注:**

> ハイパーバイザーは特定の Linux ディストリビューションと互換性があります。 Amazon Linux 2 で HDX 3D Pro を使用するには、NVIDIA ドライバー 470 をインストールすることをお勧めします。

GPU

Linux VDAでは、HDX 3D Proは以下の種類の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。

お使いのGPUカードに最新の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 15までをサポートしています。詳細については、NVIDIA Virtual GPU Software Supported GPUsを参照してください。

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

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。

インストール概要

このセクションでは、以下の手順について説明します。

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

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いです。必要なパッケージをインストールし、構成ファイルを自動的にカスタマイズすることで、Linux VDA の実行環境をセットアップするのに役立ちます。
> - easy install は、Quest を使用したドメイン参加をサポートしていません。

- ドメインに参加していない VDA を作成するには、Machine Creation Services (MCS) を使用する必要があります。詳細については、「ドメインに参加していない Linux VDA の作成」を参照してください。
- ドメインに参加していない VDA で利用可能な機能については、「ドメインに参加していない VDA」を参照してください。

ステップ 1：構成情報と Linux マシンを準備する

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 Winbind、SSSD、Centrify、および PBIS 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。
- 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。
- 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、簡易インストールではドメイン参加パスワードは保存されません。インタラクティブモードで簡易インストールスクリプト (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：ハイパーバイザーを準備する

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

Citrix Hypervisor™ での時刻同期の修正

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

Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかを確認できます。

```
su -



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

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能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、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
```

再起動後もこの変更を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 および Citrix Hypervisor とは異なります。Hyper-V の時刻同期は、NTP 時刻同期と共存し、それを補完できます。

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。どちらもシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と同期していることを確認してください。この場合、ホスト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：.NET Runtime 6.0をインストールする

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。

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

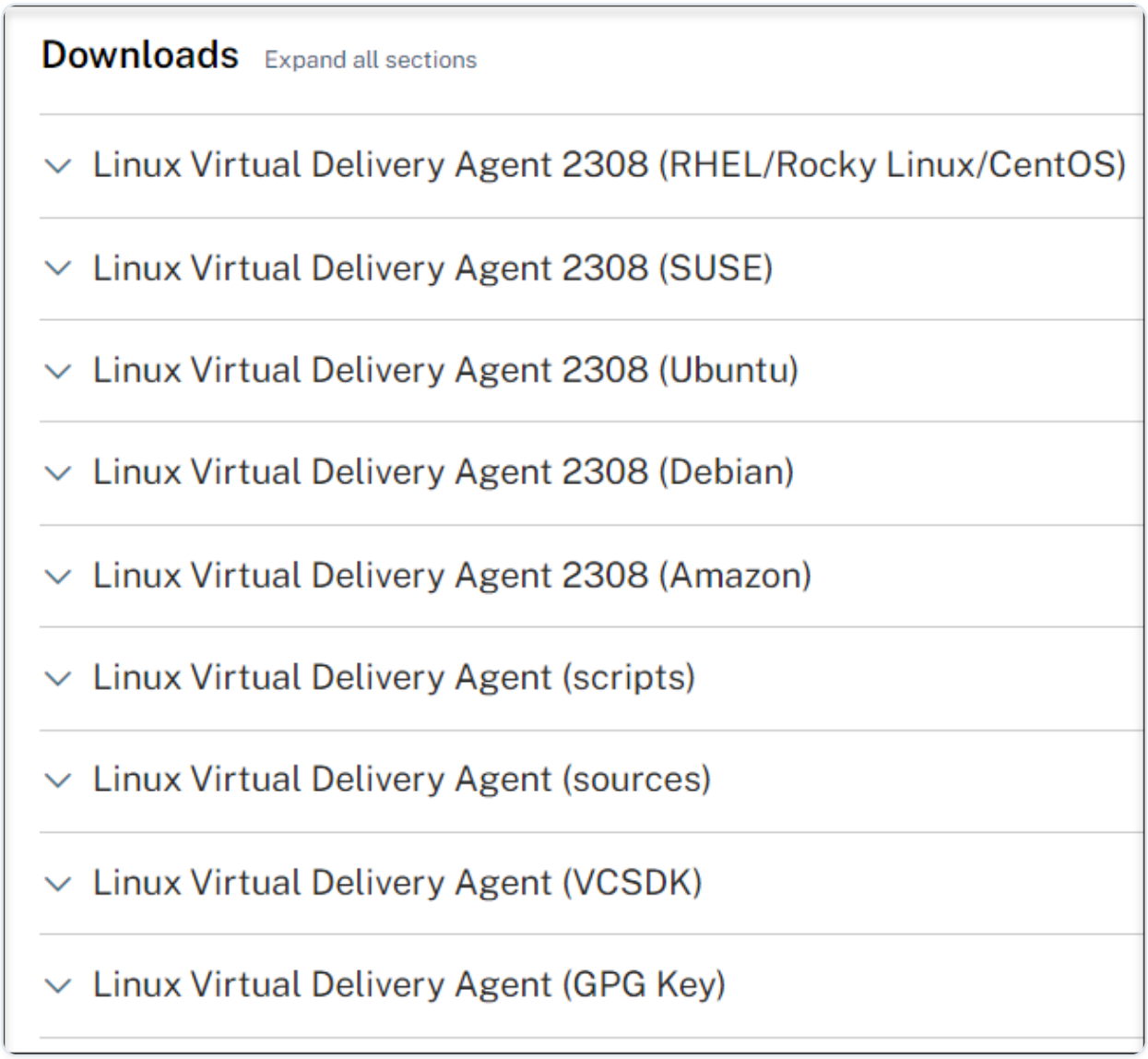
コマンドの出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンドの出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

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. 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：

   GPG公開キー(/en-us/linux-virtual-delivery-agent/2308/media/gpg-key.png)

   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、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。

Amazon Linux 2、CentOS、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 9.2/9.0およびRocky Linux 9.2/9.0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RHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAにアクセス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を再起動した後、コンソールで次のコマンドを実行します：

```
nmcli dev connect eth0

service NetworkManager restart
```

Ubuntu/Debianディストリビューションの場合：

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

sudo apt-get install -f
```

注:

- Debian 11.3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 20.04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 15.4 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。

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

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

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

ステップ 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に、SQLite と PostgreSQL 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。
> - 簡単インストールと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指定して使用できます。**/etc/xdl/db.conf**で特に指定しない限り、Linux VDAはデフォルトでPostgreSQL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。

1. 使用するデータベースを指定するには、**/etc/xdl/db.conf**を編集します。
2. **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：インストールを完了するためのランタイム環境のセットアップ

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**の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、Linux VDAのランタイム環境を構成する簡単インストールスクリプトです。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。

簡単インストールでは、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、同期するための構成ファイルとして**/opt/Citrix/VDA/sbin/ctxinstall.conf**を使用します。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。

> **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、ルートユーザーのみが**/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、次の環境変数を設定します。

- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDAサーバーのホスト名。
- **CTX_EASYINSTALL_DNS=ip-address-of-dns** – DNSのIPアドレス。
- **CTX_EASYINSTALL_NTPS=address-of-ntps** – NTPサーバーのIPアドレスまたは文字列名。
- **CTX_EASYINSTALL_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の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。

次の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、対話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てください。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_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。

ctxinstall.shを実行すると、別のスクリプトctxsetup.shが呼び出されます。ctxsetup.shは次の変数を使用します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller への登録に使用する、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_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktop サービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux 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と1494）を自動的に開くことができます。
- **CTX_XDL_AD_INTEGRATION=winbind \| sssd \| centrify \| pbis** – 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_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル (VDI) として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境の場合は、値を Y に設定します。

- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、**\<none\>** に設定します。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS をクエリして LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合、LDAP ポートを持つ LDAP FQDN のスペース区切りのリストを提供できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268、または LDAPS を使用している場合は ad1.mycompany.com:636 ad2.mycompany.com:3269 ad3.mycompany.com:3269 です。Active Directory フォレスト内でより高速な LDAP クエリを有効にするには、ドメインコントローラーで **Global Catalog** を有効にし、関連する LDAP ポート番号を 3268 または LDAPS を使用している場合は 3269 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 をクエリします。検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**\<none\>** に設定します。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロン区切りの FAS サーバーのリストを提供できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。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付加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするための .NET Runtime 6.0 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/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 に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。
- **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。

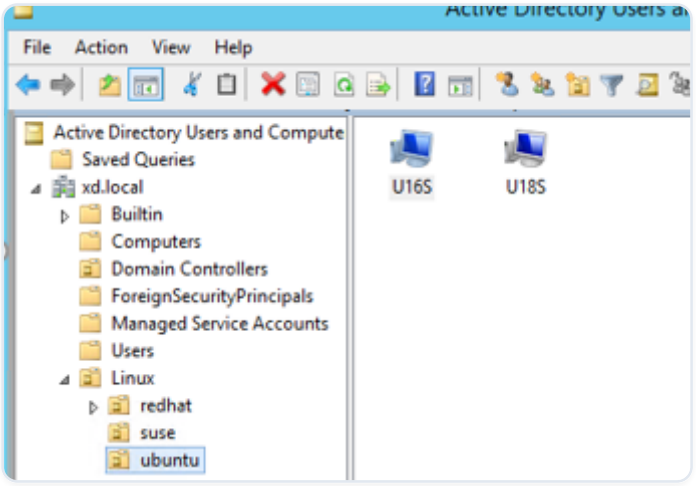
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutがリッスンするソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトポートは7502です。

考慮事項

- NetBIOSドメイン名は通常、DNSドメイン名の最初のコンポーネントであり、ドット (.) で区切られています。環境内で異なる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
アマゾン Linux 2	`"Linux/amazon"`	`"Linux/amazon"`	`"XD.LOCAL/Linux/amazon"`	`"Linux/amazon"`

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 9.2/9.0、Rocky リナックス 9.2/9.0	`"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"`
Rocky リナックス 8.x	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	該当なし	該当なし
RHEL 7	`"Linux/redhat"`	`"Linux/redhat"`	`"XD.LOCAL/Linux/redhat"`	`"Linux/redhat"`
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、純粋な **IPv6** をサポートしています。以下の前提条件と制限が適用されます。
  - マシンが純粋な **IPv6** ネットワーク経由で必要なパッケージをダウンロードできるように、Linux リポジトリを構成する必要があります。
  - Centrify は純粋な **IPv6** ネットワークではサポートされていません。

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

- ターゲットセッションユーザーのデスクトップ環境は、次の手順を実行して変更することもできます。

  a. VDA 上の **$HOME/<username\>** ディレクトリの下に `.xsession` または `.Xclients` ファイルを作成します。Amazon Linux 2 を使用している場合は、`.Xclients` ファイルを作成します。他のディストリビューションを使用している場合は、`.xsession` ファイルを作成します。

  b. ディストリビューションに基づいてデスクトップ環境を指定するには、`.xsession` または `.Xclients` ファイルを編集します。

  - **MATEデスクトップの場合**

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

if [ -n "$MSESSION" ]; then

exec mate-session
```

```
fi
```

- **GNOME Classicデスクトップの場合**

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

if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic

exec gnome-session --session=gnome-classic

fi
```

- **GNOMEデスクトップの場合**

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

if [ -n "$GSESSION" ]; then

exec gnome-session

fi
```

c. ターゲットセッションユーザーと700ファイル権限を共有します。

バージョン2209以降、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ます。この機能を有効にするには、VDAに切り替え可能な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。

- 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。

    Amazon Linux 2/RHEL の場合: wget `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-rhel6-x86_64.tgz>`

CentOS 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 の場合: 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。

    Amazon Linux 2, CentOS 7, RHEL 8, RHEL 7, SUSE 15.4 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>`

    デビアン、ウブントゥ：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PBISパッケージの特定のバージョンを取得します。そのためには、/opt/Citrix/VDA/sbin/ctxinstall.sh ファイル内の「pbisDownloadRelease」および「pbisDownloadExpectedSHA256」行を変更します。

    例として、次のスクリーンショットを参照してください。

    
  - 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パッケージを取得します。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、前述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。

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き、更新されたすべての値は **/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_EASYINSTALL_HOSTNAME=host-name


export CTX_EASYINSTALL_DNS=ip-address-of-dns


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_XDL_SUPPORT_DDC_AS_CNAME=Y | N

export CTX_XDL_DDC_LIST='list-ddc-fqdns'

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y | N

export CTX_XDL_ADD_FIREWALL_RULES=Y | N

export CTX_XDL_AD_INTEGRATION=winbind | centrify | sssd | pbis

export CTX_XDL_HDX_3D_PRO=Y | N

export CTX_XDL_VDI_MODE=Y | N

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

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

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

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


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


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


export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number


export CTX_XDL_TELEMETRY_PORT=port-number


export CTX_XDL_START_SERVICE=Y | N


sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh -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。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変数を指定できます。

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** を実行できます（ここで **s** は **小文字** であることに注意してください）。

GUI

ドメイン参加方法としてSSSDまたはWinbindを使用する場合、GUIを介して簡易インストールを使用できます。VDAのデスクトップ環境で **/opt/Citrix/VDA/bin/easyinstall** コマンドを実行し、簡易インストールGUIの指示に従います。

簡易インストールGUI(/en-us/linux-virtual-delivery-agent/2308/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。

- システム環境を確認する
- 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

> **ヒント:**
>
> - 「**保存**」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設定を保存します。「**読み込み**」をクリックして、指定したファイルから変数設定を読み込みます。MCS変数の構成については、「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。
> - ctxinstall.shスクリプトは、ドメイン参加パスワードを除くすべての変数設定を**/Citrix/VDA/sbin/ctxinstall.conf**に保存します。

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

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

> **注:**

> Citrix Studio の初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS** または **Server OS** オプションを選択すると、同等のホストされた共有デスクトップ配信モデルが意味されます。**Windows Desktop OS** または **Desktop OS** オプションを選択すると、マシンあたり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、再度追加する必要があります。

ステップ 12: デリバリーグループを作成する

デリバリーグループを作成し、Linux VDA マシンを含むマシンカタログを追加するプロセスは、Windows VDA マシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

> **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 2308」(/ja-jp/citrix-virtual-apps-desktops.html)を参照してください。

ステップ13：Linux VDAをアップグレードする（オプション）

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

RHEL 7およびCentOS 7の場合：

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

RHEL 8およびRocky Linux 8の場合：

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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 の場合：

> **注:**
>
> RHEL 9.2/9.0およびRocky Linux 9.2/9.0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。

```
sudo rpm -U XenDesktopVDA-<version>.el9x.x86_64.rpm
```

SUSEの場合：

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

Ubuntu 20.04の場合：

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

Ubuntu 22.04の場合：

```
sudo dpkg -i xendesktopvda_<version>.ubuntu22.04_amd64.deb
```

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時に発生する可能性のある問題をトラブルシューティングします。

SSSD を使用したドメイン参加に失敗する

ドメインへの参加を試行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（画面出力の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にグレースクリーンが表示される

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

この問題を解決するには：

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

   `greeter-show-manual-login=true`

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

/var/log/xdl/hdx.log:

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



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



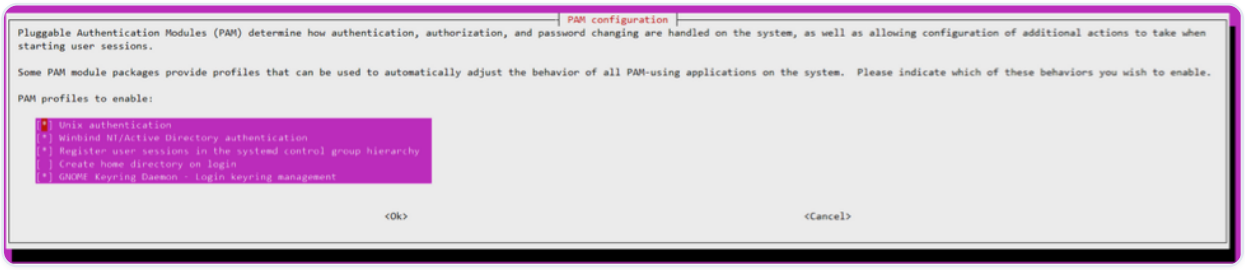
2016-11-02 13:21:19.023 <P22492:S1> citrix-ctxlogin: ChildPipeCallback:
Login Process died: normal.



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

> **ヒント:**
>
> この問題の根本原因は、ドメイン管理者のホームディレクトリが作成されていないことです。
>
> この問題を解決するには:

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



dbusエラーによりセッションが起動しない、またはすぐに終了する

/var/log/messages (RHELまたはCentOSの場合):

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



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



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



Oct 27 04:17:18 CentOS7 gnome-session: gnome-session[19146]: ERROR: Failed
to connect to system bus: Exhausted all available authentication
```

```
mechanisms (tried: EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS) (available:
EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS)



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



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



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

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

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



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



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



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

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



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

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

SELinuxによりSSHDがホームディレクトリにアクセスできない

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/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. /etc/selinux/config に以下の変更を加えて SELinux を無効にします。

   SELINUX=無効

2. VDA を再起動します。

ドメインに参加していないLinux VDAを作成する

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

重要:

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

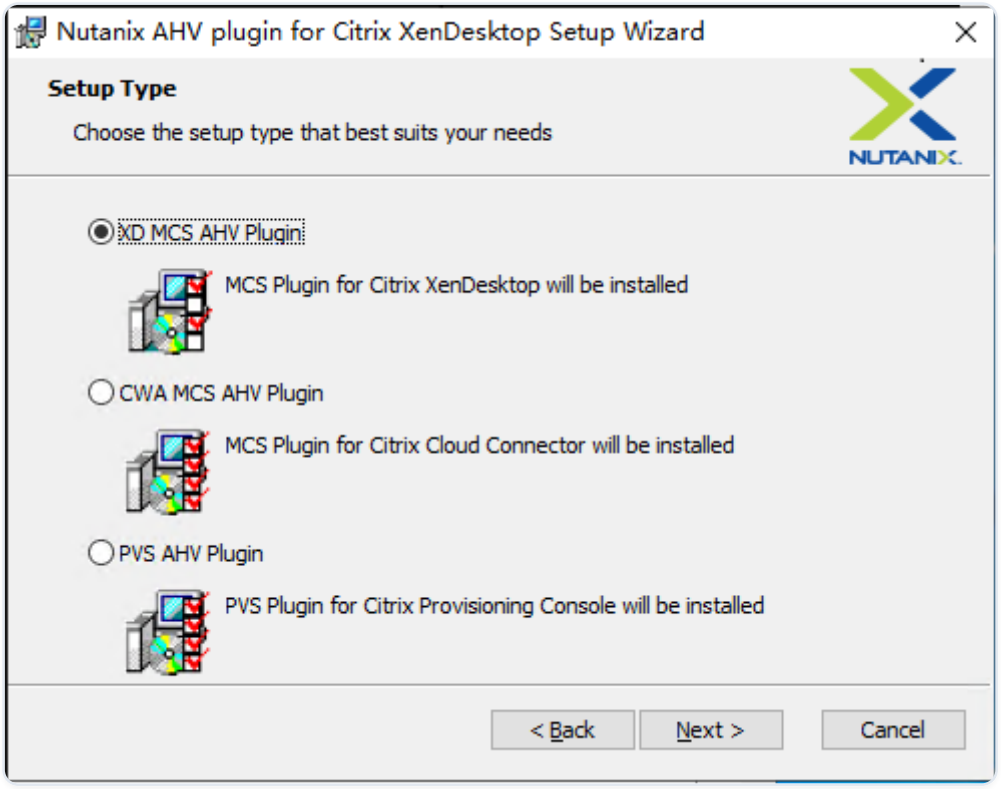
- 非SSO認証
- Azure Active Directoryによる認証
- ランデブー V2

(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Delivery Controller用Nutanix AHVプラグインのインストールと登録

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



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

Citrix Cloud™ Connector用の**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する

- Nutanix Acropolisフォルダーが`C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`に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

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します。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、左ペインで**Hosting**を選択します。

4. アクションバーで を選択します。
5. ウィザードは次のページに案内します。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、**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

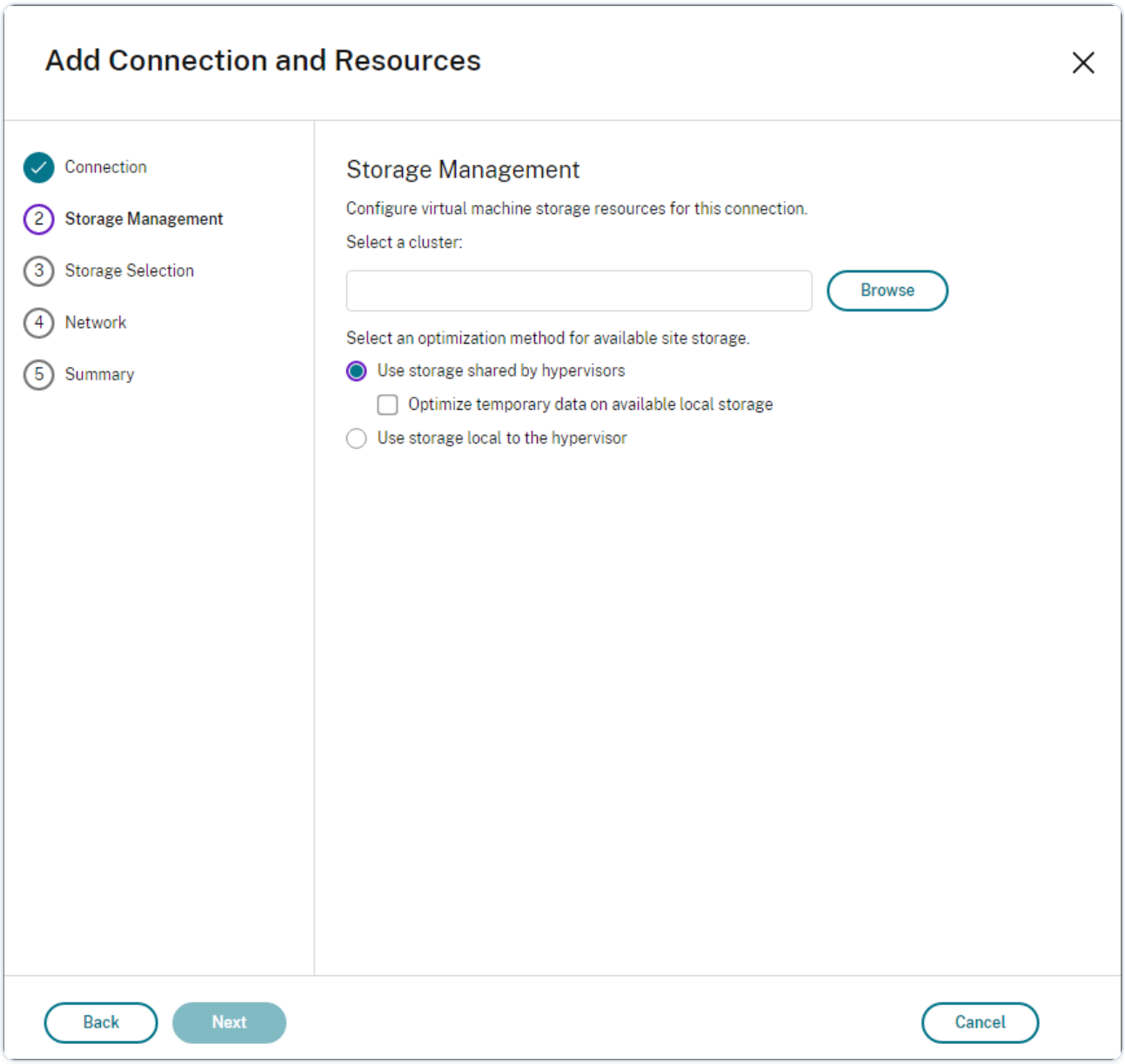
接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2308/media/add-conn-connection.png)

「**接続**」ページで：

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

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

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。

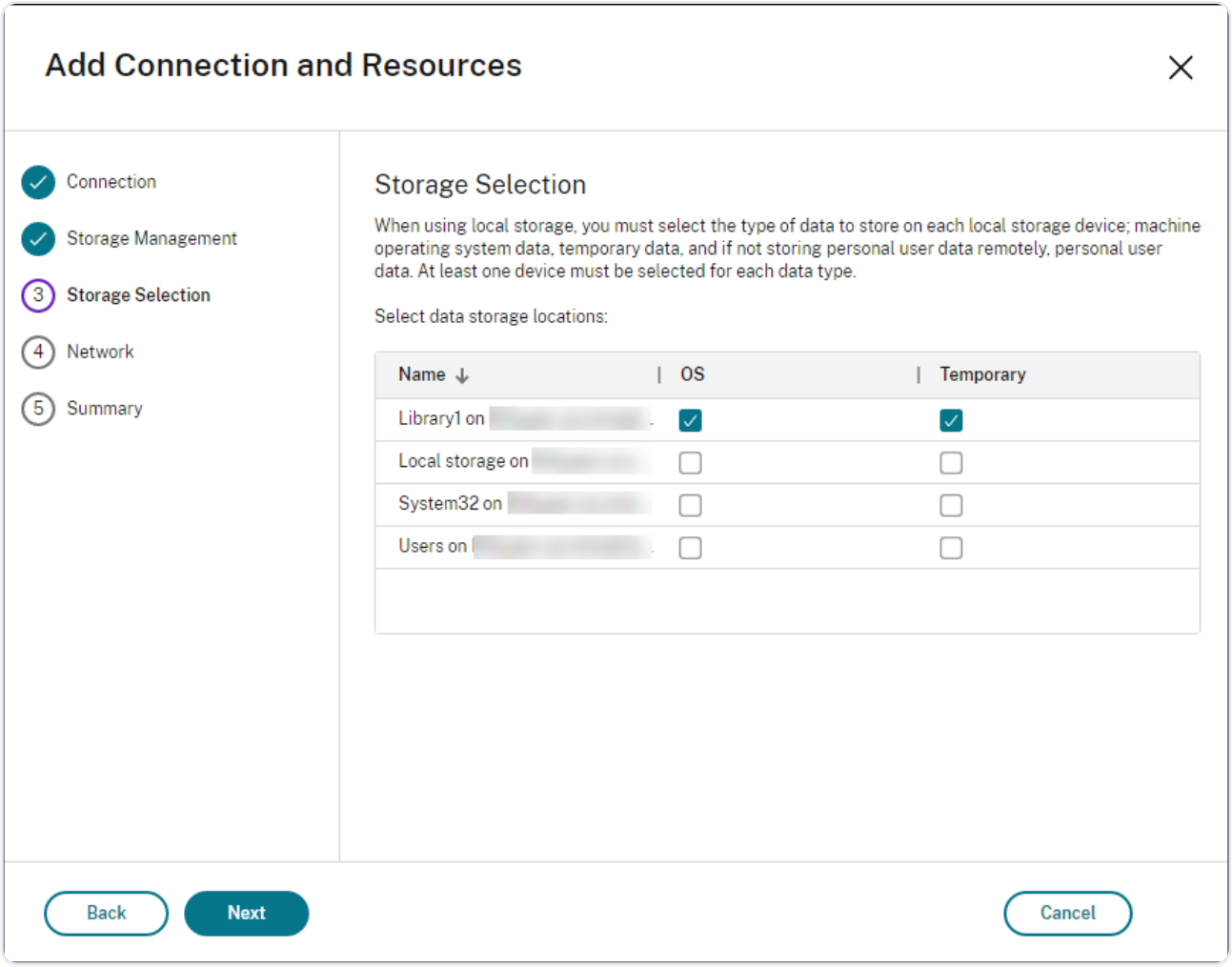
Hyper-V または VMware ホストへの接続を構成している場合は、クラスター名を参照して選択します。その他の接続タイプでは、クラスター名は要求されません。

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。

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

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

ステップ 2c: ストレージの選択



ストレージの選択の詳細については、「ホストストレージ」を参照してください。

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

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

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

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

ステップ 2d: リージョン

（一部のホストタイプでのみ表示されます。）リージョンの選択は、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リージョンを選択してください。

ステップ 2e: ネットワーク

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

一部の接続タイプ（Azure Resource Managerなど）では、VMが使用するサブネットも一覧表示されます。1つ以上のサブネットを選択します。

ステップ2f: 概要

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

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

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

> 詳細については、「接続の作成と管理」を参照してください。

ステップ3: マスターイメージの準備

> **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。

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

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

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

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

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

   RHEL/CentOS/Rocky Linux 向け:

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

   Ubuntu/Debian 向け:

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

   SUSE 向け:

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

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

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

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

> テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Runtime 6.0をインストールしてください。

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

RHEL/CentOS/Rocky Linux 向け:

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

> **注:**
>
> RHELおよびCentOSの場合、Linux VDAをインストールして`deploymcs.sh`を正常に実行する前に、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。

- GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、イーサネット接続が失われ、VMの再起動後にLinux VDAにアクセスできなくなる可能性があります。この問題を回避するには、VMを再起動する前に次のコマンドを実行してください。

  ```
  nmcli dev connect eth0

  service NetworkManager restart
  ```

Ubuntu/Debian 向け:

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



apt-get install -f
```

SUSE 向け:

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

ステップ3c: tdb-toolsパッケージをインストールするためにリポジトリを有効にする (RHEL 7のみ)

RHEL 7サーバー向け:

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

RHEL 7 ワークステーションの場合：

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

ステップ 3d: (SUSE で) 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: 使用するデータベースを指定する

実験的な機能として、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、SQLite と PostgreSQL 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。

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。

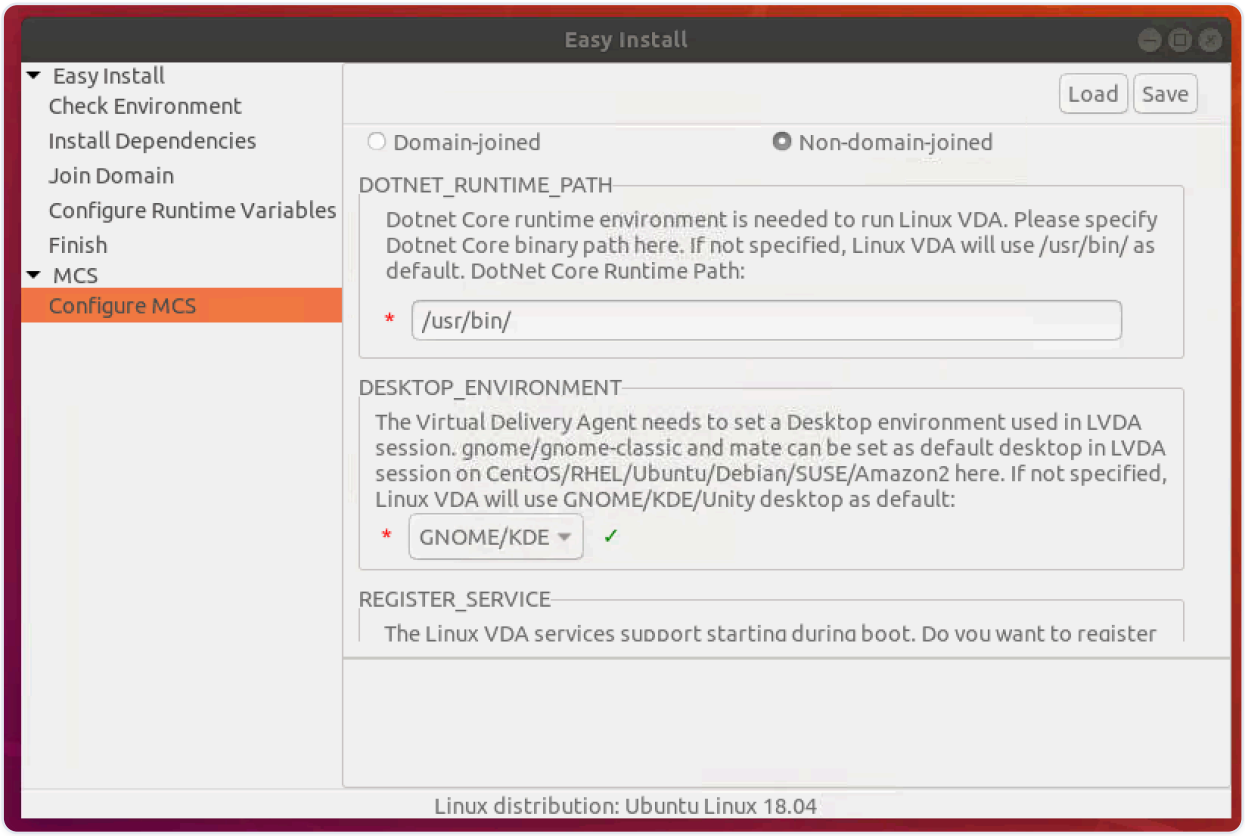
注:

- VDI モードでのみ SQLite 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、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。`/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。
- 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。

ステップ 3f: 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。
- 簡単インストールGUIを使用します。簡単インストール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:**
>
> **保存**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。**読み込み**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で構成できる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` =*dotnetランタイムのインストールパス **

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

`ADD_FIREWALL_RULES` =**Y | N** `VDI_MODE` =**Y | N**

`START_SERVICE` =**Y | N**

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

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

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

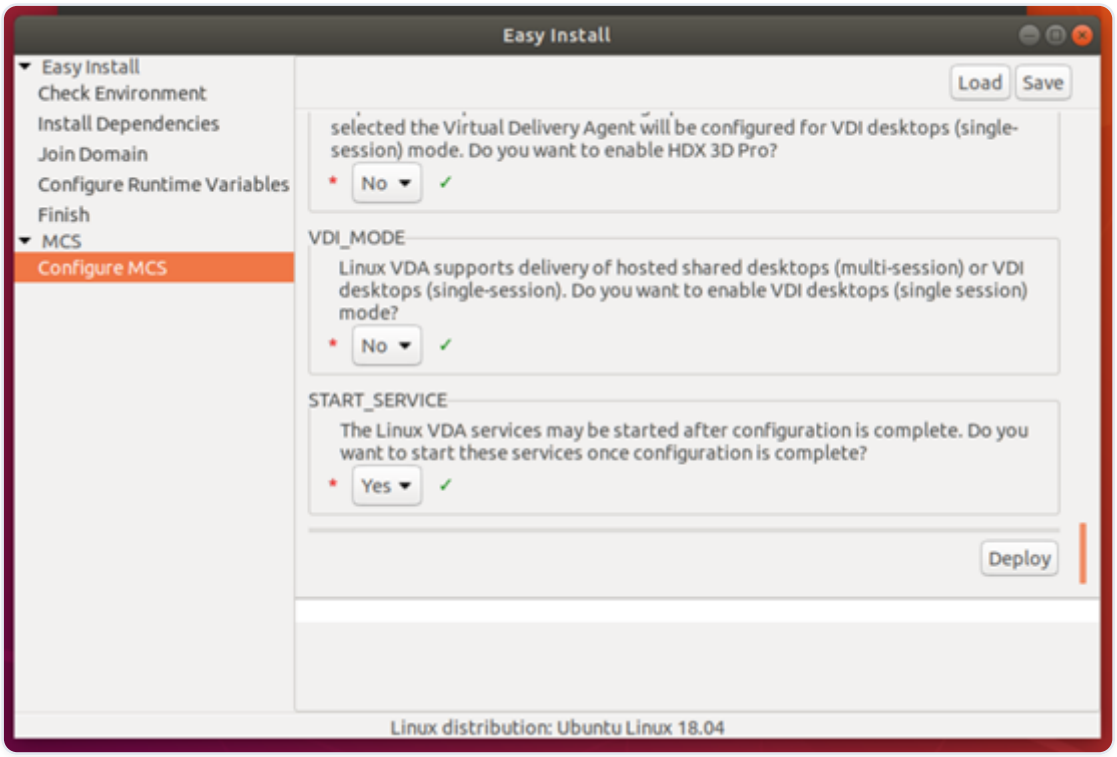
たとえば、レジストリ値を書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルにそれぞれ追加できます。

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

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

ステップ 3h: マスターイメージを作成する

1. `/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、**展開**をクリックします。



GUIで**展開**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。

   - **(Citrix Hypervisor、GCP、VMware vSphereの場合)** テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。
   - **(Azureの場合)** テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを VMをシャットダウンします。テンプレートVMの電源ステータスが**停止済み (割り当て解除済み)** 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、この名前が必要です。

     テンプレートVMの停止済み電源ステータス(/en-us/linux-virtual-delivery-agent/2308/media/stopped-deallocated2.png)
   - **(AWSの場合)** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**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、**イメージ > イメージの作成**を選択します。必要な情報を入力し、設定を行います。**イメージの作成**をクリックします。

     EBSイメージの作成(/en-us/linux-virtual-delivery-agent/2308/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。

ステップ4：マシンカタログを作成する

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**マイサービス > 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. ウィザードに従ってマシンカタログを作成します。

   Nutanixに固有の**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、カタログに作成するマシンの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。詳細については、「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てください。この方法により、作成プロセスを続行できます。

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、「デリバリーグループの作成」を参照してください。

Machine Creation Services™ (MCS) を使用した Linux VDA の作成

MCS を使用して、ドメイン参加済みおよび非ドメイン参加済み VDA を作成できます。

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 3h: MCS 変数の構成」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	セントリファイ	PBIS
デビアン 11.3	はい	はい	いいえ	はい
RHEL 9.2/9.0	はい	はい	いいえ	いいえ
RHEL 8.8/8.6	はい	はい	はい	はい
ロッキー リナックス 9.2/9.0	はい	はい	いいえ	いいえ

	ウィンバインド	SSSD	セントリファイ	PBIS
ロッキー リナックス 8.8/8.6	はい	はい	いいえ	いいえ
RHEL 7.9, CentOS 7.9	はい	はい	はい	はい
SUSE 15.4	はい	はい	いいえ	はい
ウブントゥ 22.04、ウブントゥ 20.04	はい	はい	いいえ	はい

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ており、簡易インストールは使用されていません。簡易インストールは、RHEL 8.x/9.x および Rocky Linux 8.x/9.x に **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 サーバーが AD 認証に SSSD を使用するように構成されていること。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。

サポートされているハイパーバイザー

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

サポートされているもの以外のハイパーバイザーでマスターイメージを準備しようとすると、予期しない結果が発生する可能性があります。

MCS を使用して Linux VM を作成する

考慮事項

- 2203 リリース以降、Citrix Virtual Apps and Desktops および Citrix DaaS (旧 Citrix Virtual Apps and Desktops サービス) 向けに、Microsoft Azure、AWS、GCP で Linux VDA をホストできます。これらのパブリッククラウドホスト接続を Citrix Virtual Apps and Desktops 展開に追加するには、Citrix Universal Subscription または Hybrid Rights ライセンスが必要です。Universal Subscription および Hybrid Rights ライセンスの詳細については、Transition and Trade Up (TTU) with Citrix Universal Subscription を参照してください。
- 仮想マシンを作成するために MCS と組み合わせてベアメタルサーバーを使用することはサポートされていません。
- Citrix® は、関連する Linux ディストリビューションでの初期機能検証のために、以下の Centrify バージョンを使用します:

LINUXディストリビューション	CENTRIFY のバージョン
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、次のタスクを完了してください。
  - テンプレートマシンで、`/etc/xdl/mcs/mcs.conf` ファイルで PBIS または Centrify パッケージのダウンロードパスを構成するか、PBIS または Centrify パッケージを直接インストールします。
  - `/opt/Citrix/VDA/sbin/deploymcs.sh` を実行する前に、その配下にあるすべての MCS で作成されたマシンに対して書き込み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。

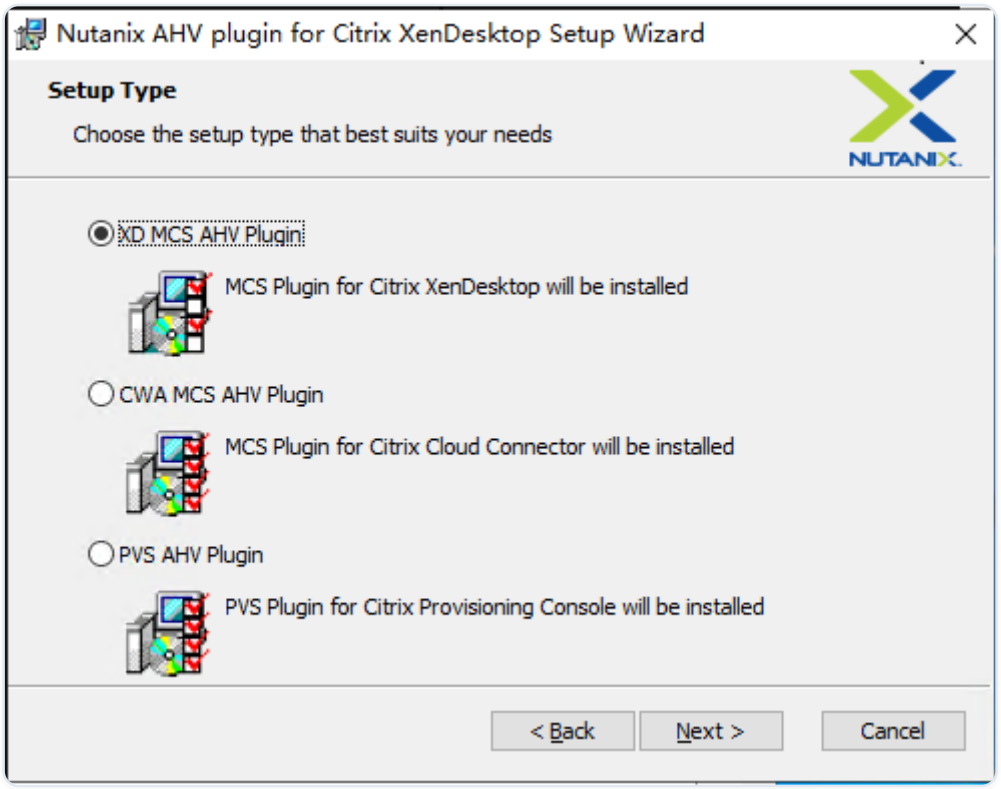
- `/opt/Citrix/VDA/sbin/deploymcs.sh` の実行が完了した後、MCS で作成されたマシンを再起動する前に、展開に応じて、Delivery Controller または Citrix Cloud Connector で `klist -li 0x3e4 purge` を実行します。

(Nutanix のみ) ステップ 1: Nutanix AHV プラグインをインストールして登録する

Nutanix から Nutanix AHV プラグインパッケージを入手します。Citrix Virtual Apps and Desktops 環境にプラグインをインストールして登録します。詳細については、Nutanix Support Portal で入手できる Nutanix Acropolis MCS プラグインインストールガイドを参照してください。

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

Citrix Virtual Apps and Desktops™ をインストールした後、Delivery Controller 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 用 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 Connectors にインストールします。AHV を使用しないリソースの場所を提供する Citrix Cloud Connectors 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 Connectors で Citrix RemoteHCLServer Service を再起動します。

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

ステップ 2: ホスト接続を作成する

このセクションでは、Azure、AWS、Citrix Hypervisor、GCP、Nutanix AHV、および VMware vSphere へのホスト接続を作成する方法の例を示します。詳しくは、Citrix Virtual Apps and Desktops のドキュメントの 接続とリソースの作成と管理 および Citrix DaaS のドキュメントの 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 Citrix Hypervisor 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. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS** を選択します。
3. **Manage > Full Configuration** から、左ペインで **Hosting** を選択します。

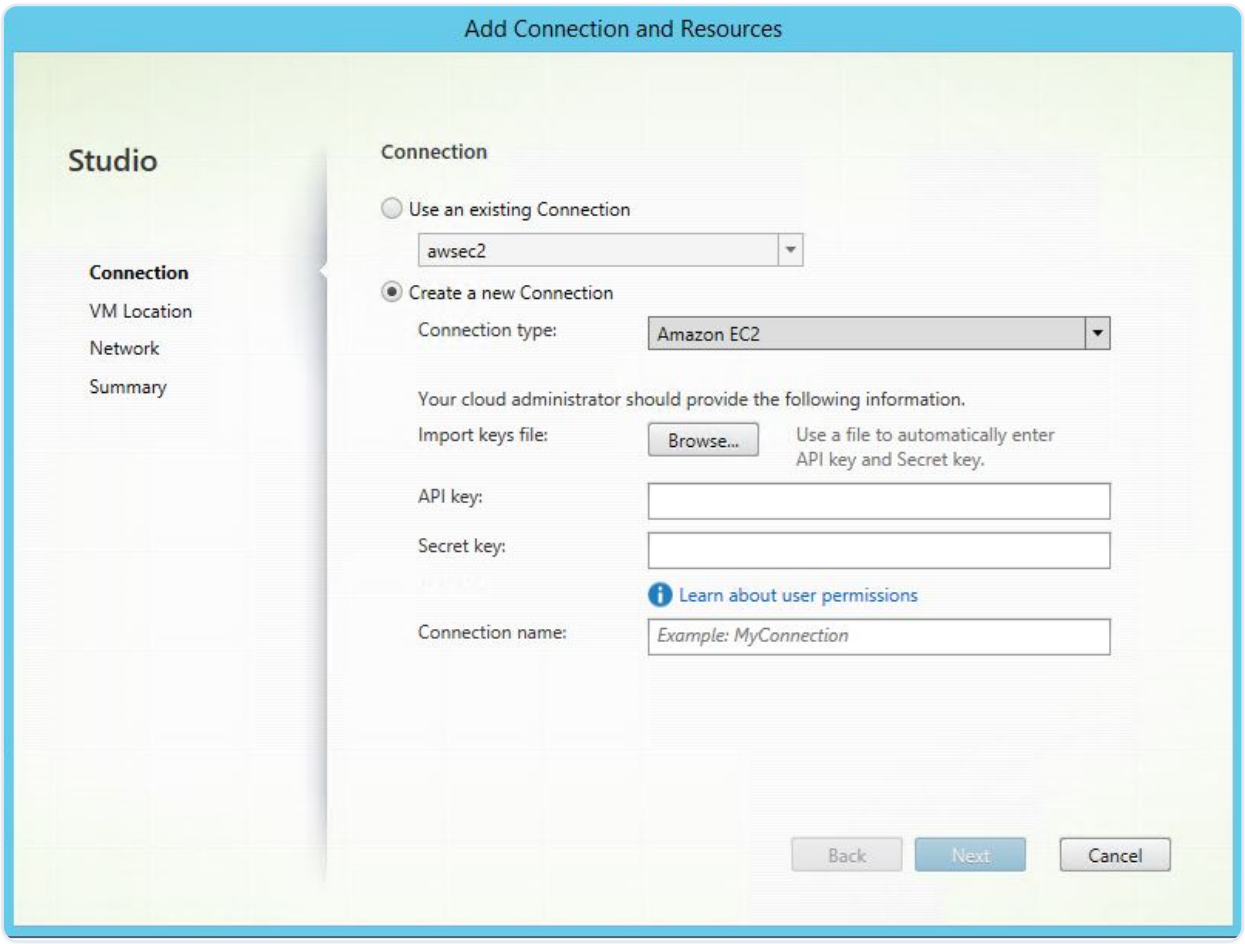
4. アクションバーで**「接続とリソースの追加」**を選択します。

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2308/media/add-conn-connection.png)

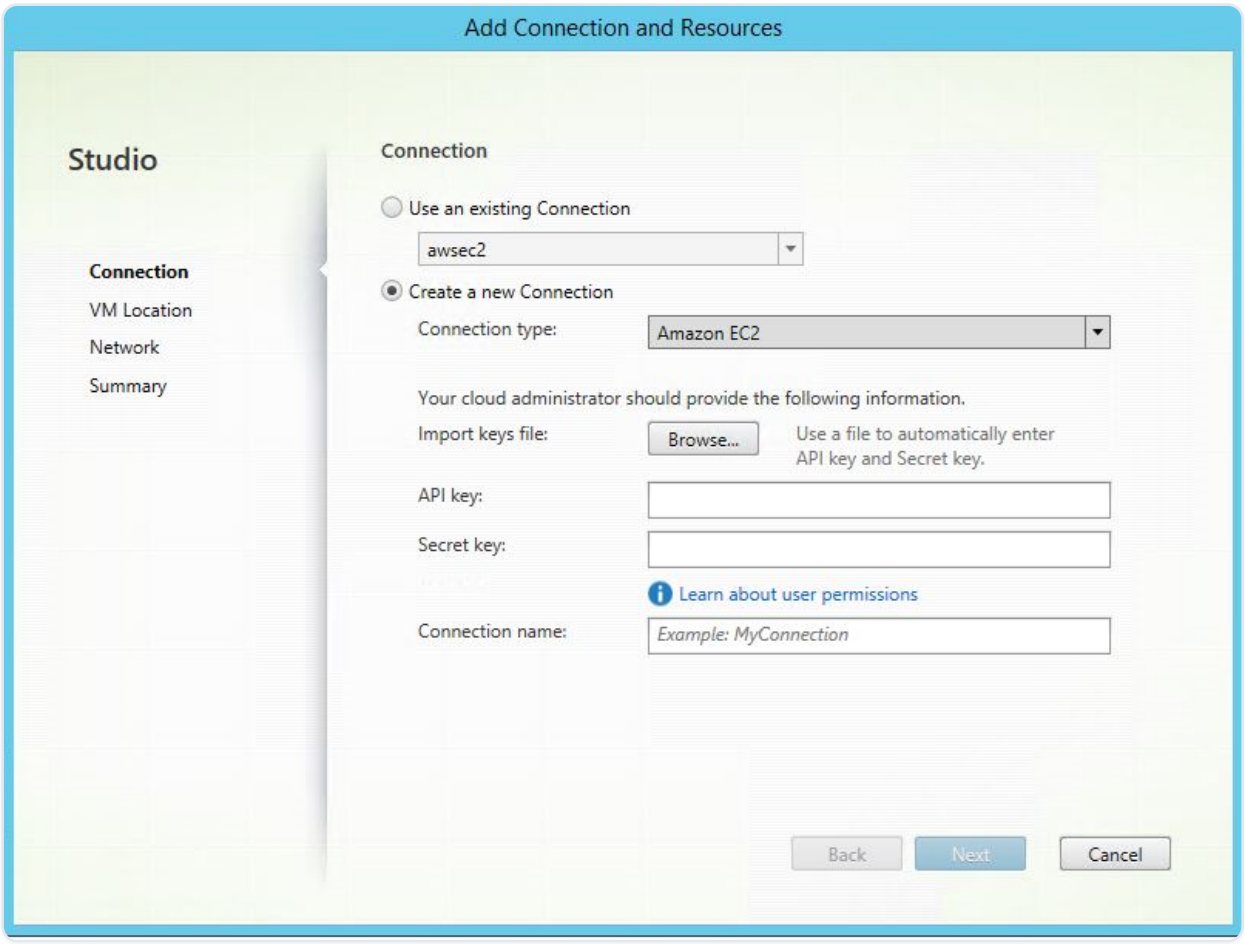
5. 接続の種類として Microsoft Azure を選択します。
6. ウィザードがページを案内します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、**Summary** ページに到達するまで **Next** を選択します。詳細については、ドメインに参加していない Linux VDA の作成 の記事の「**手順 2: ホスト接続を作成する**」を参照してください。

Citrix Studio で AWS へのホスト接続を作成する

1. Citrix Studio で、**構成** > **ホスティング** > **接続およびリソースの追加** を選択します。
2. 接続の種類として **Amazon EC2** を選択します。



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



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

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

4. ウィザードが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、**次へ**を選択して**概要**ページに到達します。

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

1. Citrix Cloudにサインインします。

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

Citrix StudioでGCPへのホスト接続を作成する

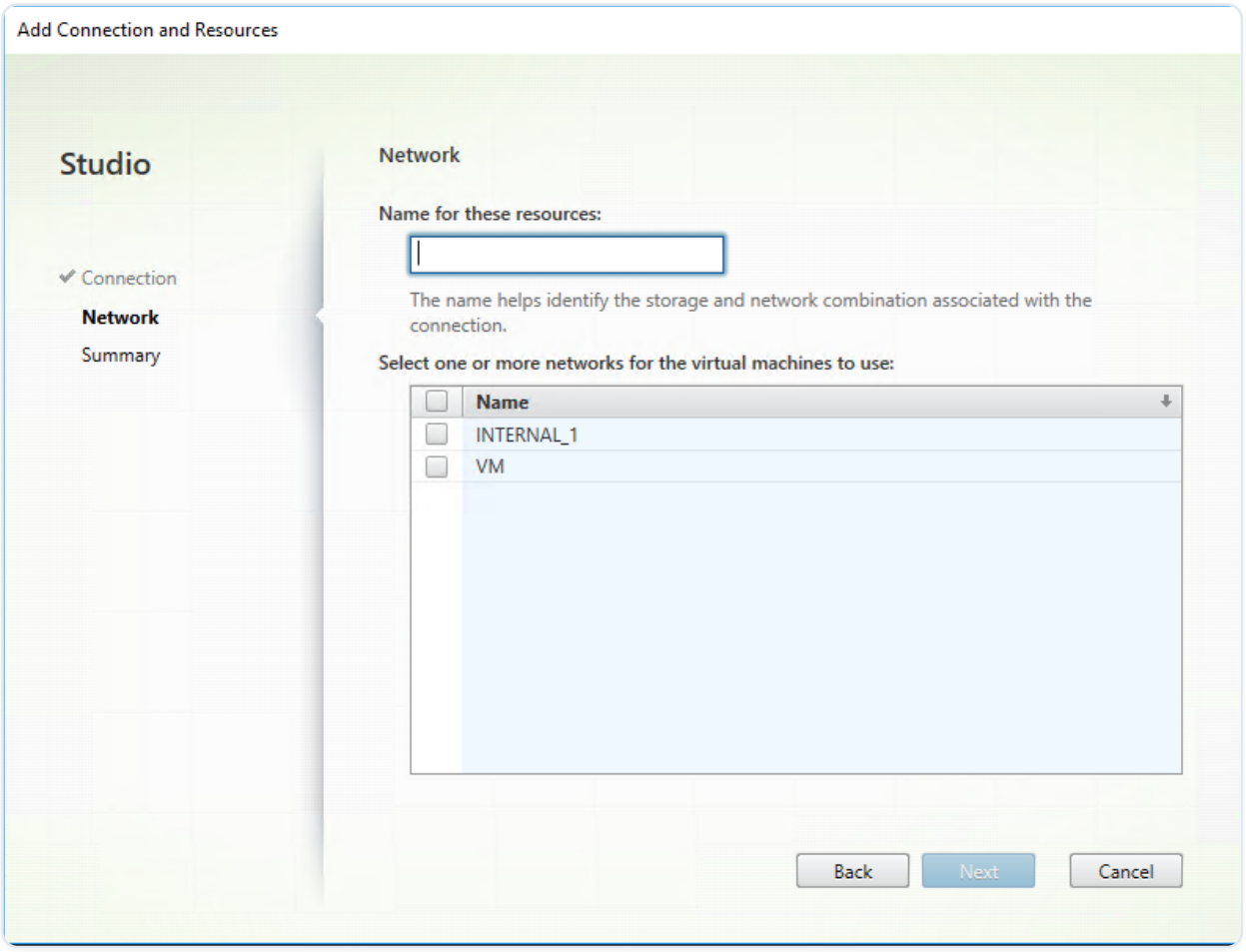
Google Cloud Platform仮想化環境に従ってGCP環境をセットアップし、次に、以下の手順を完了してGCPへのホスト接続を作成します。

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**マイサービス > DaaS**を選択します。
3. **管理 > 完全な構成**から、左ペインで**ホスティング**を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。
5. 接続の種類として**Google Cloud Platform**を選択します。
6. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
7. ウィザードが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、**次へ**を選択して**概要**ページに到達します。詳細については、ドメインに参加していないLinux VDAの作成の記事の**手順2：ホスト接続を作成する**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

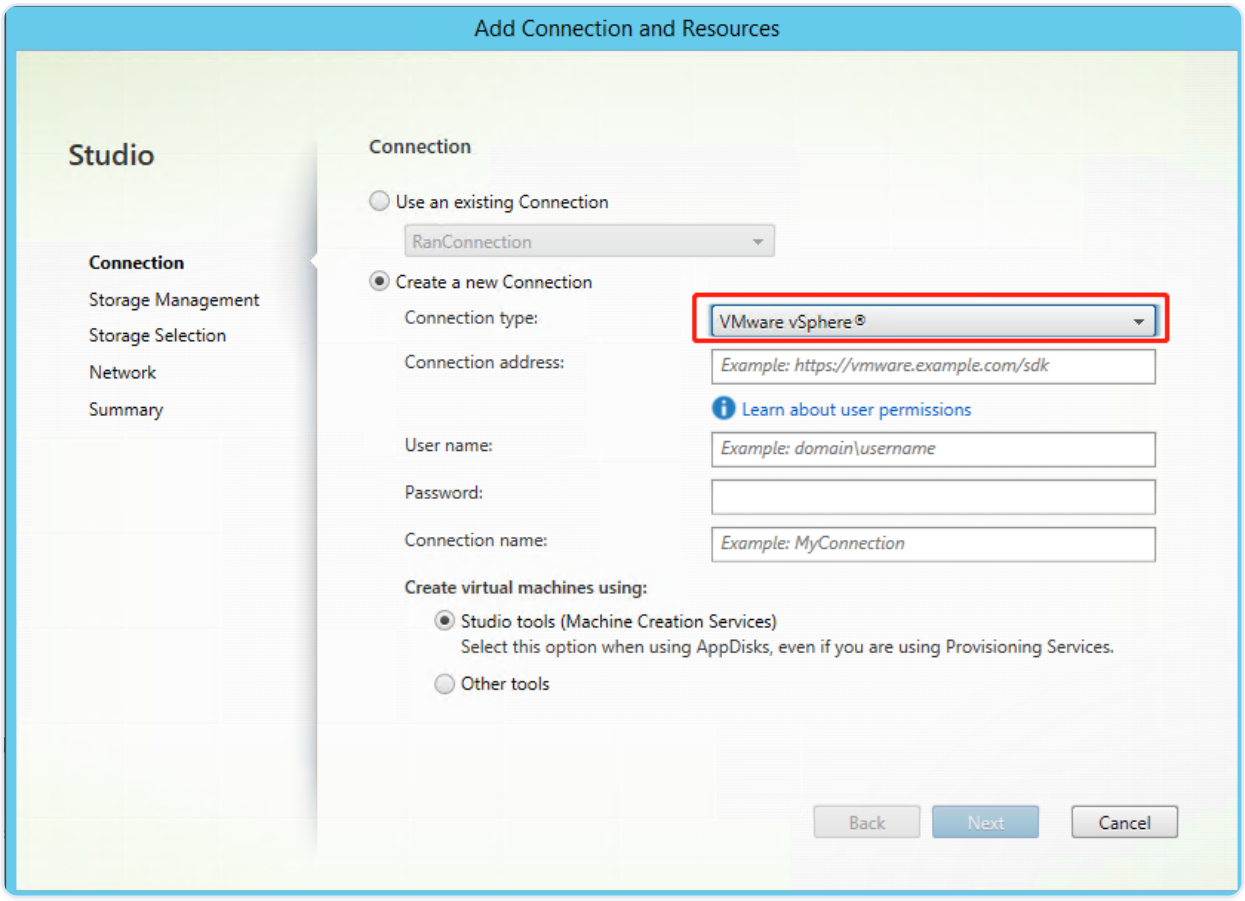
1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続とリソースの追加**を選択します。クラウドDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続とリソースの追加**を選択して、Nutanixハイパーバイザーへの接続を作成します。
2. **接続とリソースの追加**ウィザードで、**接続**ページで接続の種類としてNutanix AHVを選択し、ハイパーバイザーのアドレス、資格情報、および接続名を指定します。**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

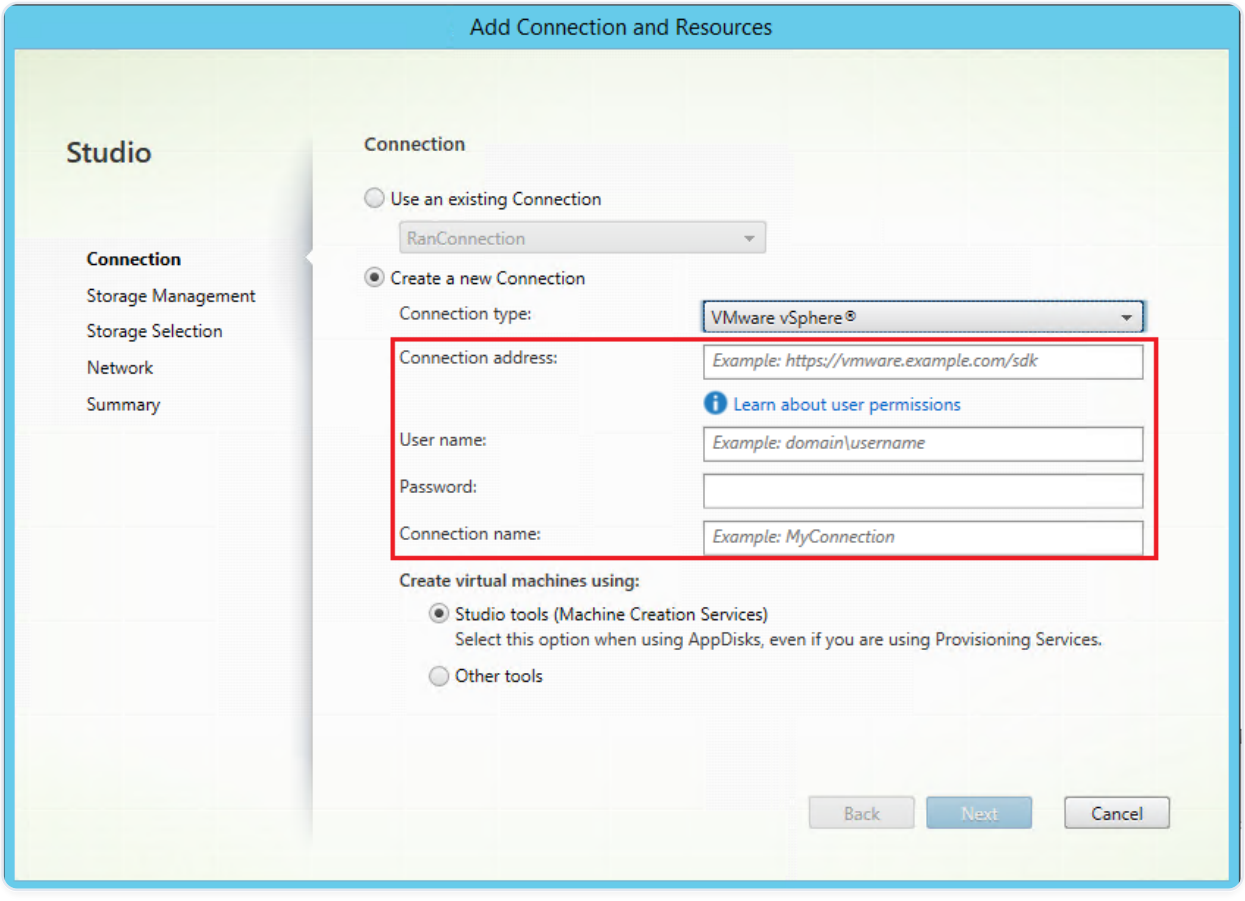


Citrix StudioでVMwareへのホスト接続を作成する

1. vSphere環境にvCenter Serverをインストールします。詳しくは、VMware vSphereを参照してください。
2. Citrix Studioで、**構成 > ホスティング > 接続とリソースの追加**を選択します。
3. 接続の種類としてVMware vSphereを選択します。



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



ステップ3：マスターイメージを準備する

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

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

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

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

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

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

   RHEL/CentOS/Rocky Linuxの場合:

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

   Ubuntu/Debianの場合:

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

   SUSEの場合:

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

3. XenCenterの**General**タブで、テンプレートVMの仮想化状態を確認します。Citrix VM Toolsが正しくインストールされている場合、仮想化状態は**Optimized**と表示されます。

ステップ3b: AWS、Azure、GCP上のSUSE 15.4の構成を確認する

AWS、Azure、GCP上のSUSE 15.4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 20.04 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 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。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をテンプレートVMとして使用する場合は、以下を確認してください。
>   - VDAは手動でインストールされており、簡易インストールは使用されていません。簡易インストールはRHEL 8.x/9.xおよびRocky Linux 8.x/9.xで**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。
> - テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Runtime 6.0をインストールしてください。

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

RHEL/CentOS/Rocky Linuxの場合:

> **注:**

> - RHELおよびCentOSの場合、Linux VDAをインストールして`deploymcs.sh`を正常に実行する前に、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/>` の手順を参照してください。

- RHEL 9.2/9.0およびRocky Linux 9.2/9.0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 3e: tdb-toolsパッケージをインストールするためにリポジトリを有効にする (RHEL 7のみ)

RHEL 7サーバーの場合:

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

RHEL 7ワークステーションの場合:

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

ステップ3f: (SUSEの場合) 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てください:

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

```
sudo zypper install gcc

sudo zypper install make
```

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

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

4. ntfs-3gパッケージへのパスを入力します:

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

5. ntfs-3gをインストールします:

```
./configure

make

make install
```

ステップ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に、SQLiteとPostgreSQL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:

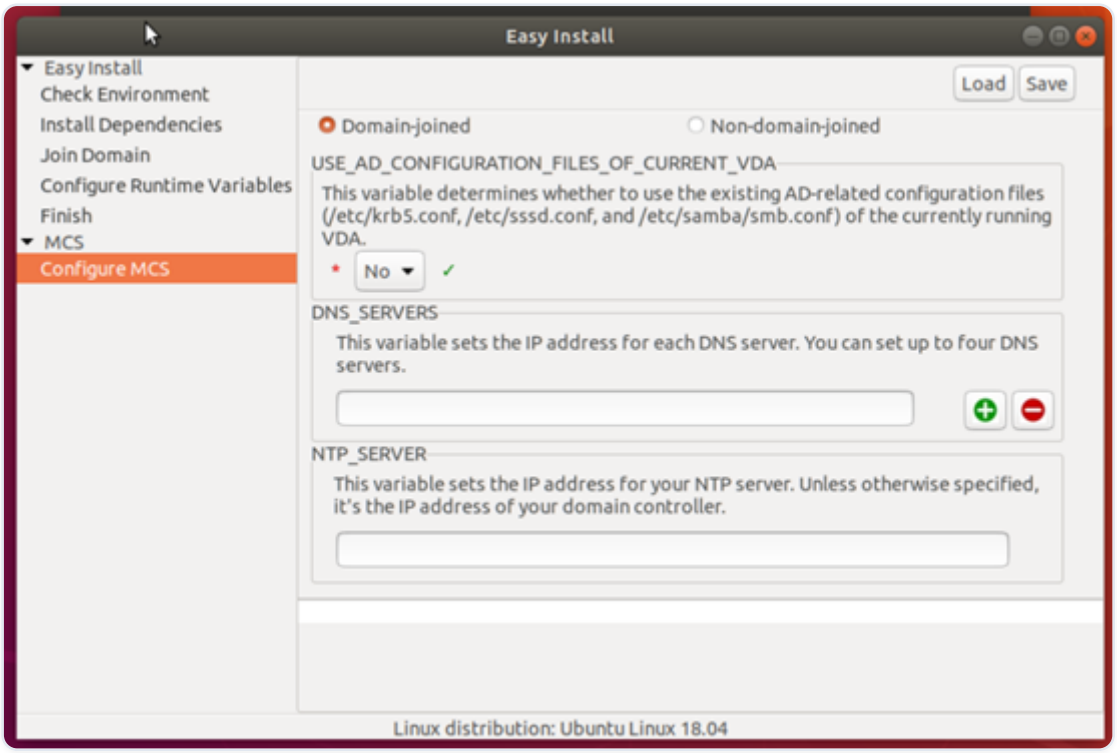
- VDIモードではSQLiteのみを使用し、ホストされた共有デスクトップ配信モデルではPostgreSQLを使用することをお勧めします。
- 簡単なインストールとMCSの場合、手動でインストールすることなく、使用するSQLiteまたはPostgreSQLを指定できます。**/etc/xdl/db.conf**で特に指定がない限り、Linux VDAはデフォルトでPostgreSQLを使用します。
- **/etc/xdl/db.conf**を使用して、PostgreSQLのポート番号を構成することもできます。

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。

ステップ 3h: 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:**
>
> **保存**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。**読み込み**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` =**Y | N**

`ADD_FIREWALL_RULES` =**Y | N** `VDI_MODE` =**Y | N**

`START_SERVICE` =**Y | N**

- **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`と`AD_INTEGRATION`変数を構成する必要があります。デフォルト値はNです。これは、マスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。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。そうでない場合、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`: 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、この記事のサポートされるディストリビューションを参照してください。
  - `CENTRIFY_DOWNLOAD_PATH`: Server Suite Free（旧Centrify Express）パッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。

- `CENTRIFY_SAMBA_DOWNLOAD_PATH`: Centrify Sambaパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。
- `PBIS_DOWNLOAD_PATH`: PBISパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をPBISに設定した場合にのみ有効になります。
- `UPDATE_MACHINE_PW`: マシンアカウントのパスワード更新の自動化を有効または無効にします。詳しくは、マシンアカウントのパスワード更新を自動化するを参照してください。
- 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`=**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 | '<none\>'**

  `LDAP_LIST`=**'list-ldap-servers' | '<none\>'** `SEARCH_BASE`=**search-base-set | '<none\>'**

  `FAS_LIST`=**'list-fas-servers' | '<none\>'** `START_SERVICE`=**Y | N**

  `TELEMETRY_SOCKET_PORT`=**port-number** `TELEMETRY_PORT`=**port-number**

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でプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

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

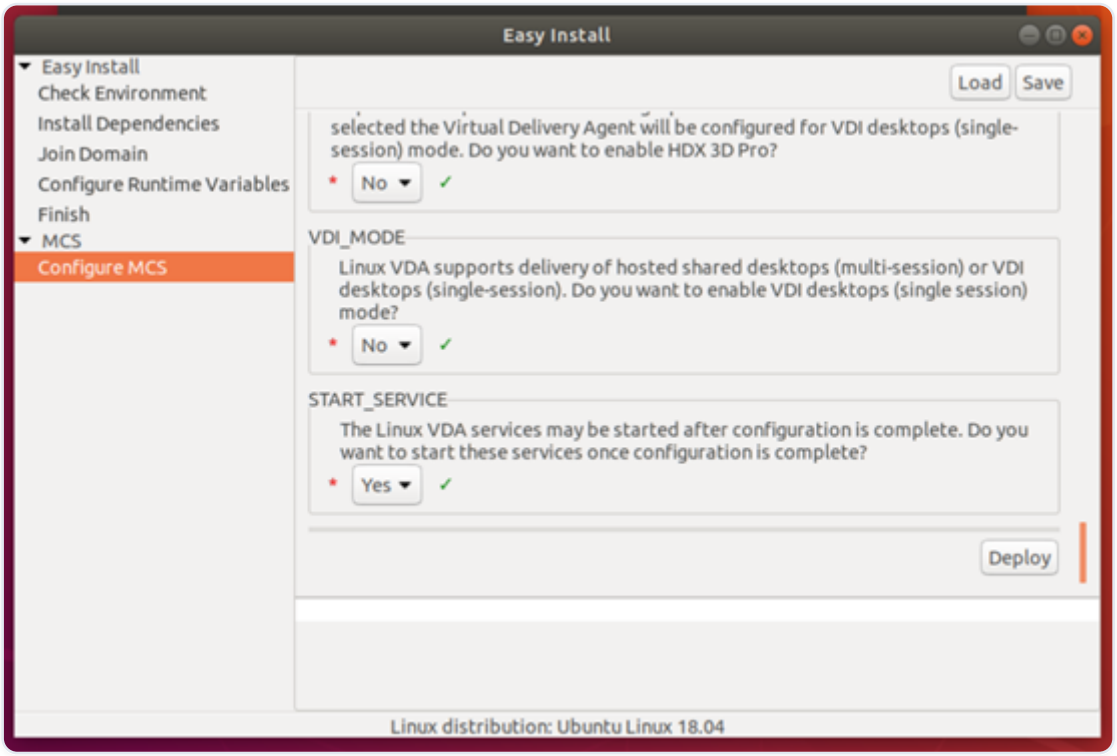
たとえば、`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに次のコマンドラインを追加して、レジストリ値をそれぞれ書き込むまたは更新できます。

```
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\ClipboardSelection" -v "Flags" -d "0x00000003"
```

ステップ3j: 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. `/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、**［展開］**をクリックします。



GUIで**展開**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ユーザーの場合は、`/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。

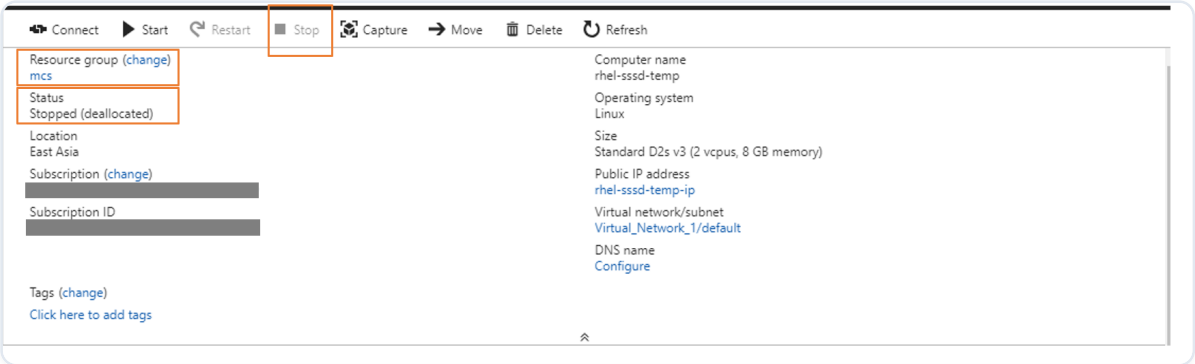
SSSDユーザーの場合は、`/etc/xdl/ad_join/sssd.conf.tmpl`、`/etc/xdl/ad_join/sssd_krb5.conf.tmpl`、および`/etc/xdl/ad_join/sssd_smb.conf.tmpl`テンプレートを更新します。

Centrifyユーザーの場合は、`/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。

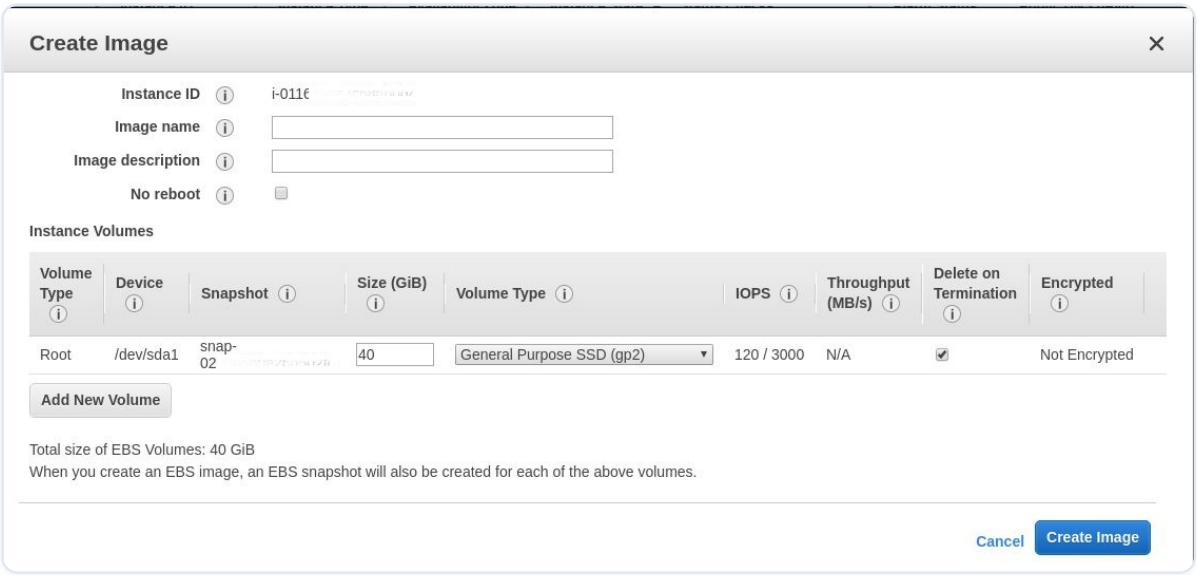
> **注:**
>
> テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、${new_hostname}、$AD_FQDNなどの変数を使用します。

4. 使用しているパブリッククラウドに基づいて、マスターイメージのスナップショットを作成し、名前を付けます。

   - **(Citrix Hypervisor、GCP、およびVMware vSphereの場合)** テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。
   - **(Azureの場合)** テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを**VMをシャットダウンします。テンプレートVMの電源ステータスが**停止済み (割り当て解除済み)**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、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を**VMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**であることを確認します。テンプレートVMを右クリックし、**イメージ > イメージの作成**を選択します。必要な情報を入力し、設定を行います。**イメージの作成**をクリックします。



- **(Nutanixの場合)** Nutanix AHVで、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。

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

(GCPの場合) ステップ3k：RHEL 8.x/9.xおよびRocky Linux 8.x/9.xでのイーサネット接続の構成

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、イーサネット接続が失われ、VMの再起動後にLinux VDAに到達できなくなる可能性があります。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。

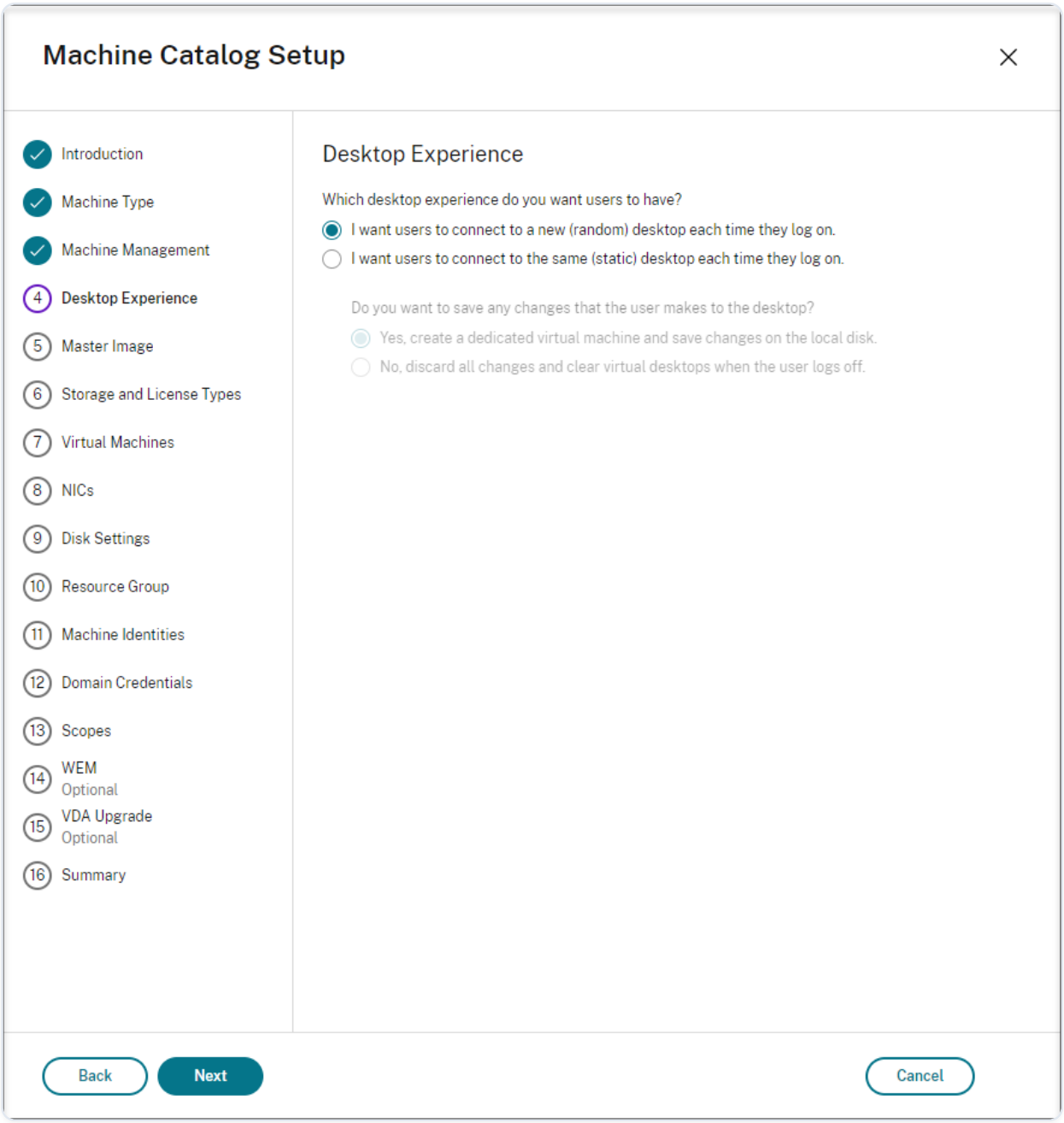
```
nmcli dev connect eth0

service NetworkManager restart
```

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ときは、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**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する（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する（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パワーオンします。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作成を参照してください。

> **注:**

MCSを使用して作成したVMがCitrix Cloud Connector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**に変わったら、手順3〜5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**という行を追加してください。

   ```
   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、.NET Runtime 6.0をインストールしていることを確認してください。
2. テンプレートマシンでLinux VDAをアップグレードします。

注:

Linux VDAセルフアップデート機能を使用して、自動ソフトウェアアップデートをスケジュールすることもできます。この目的を達成するには、テンプレートマシンのetc/xdl/mcs/mcs_local_setting.regファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。

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



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



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



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

RHEL 7およびCentOS 7の場合：

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

RHEL 8.xおよびRocky Linux 8.xの場合：

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

RHEL 9.2/9.0およびRocky Linux 9.2/9.0の場合：

注:

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 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。

```
sudo rpm -U XenDesktopVDA-<version>.el9x.x86_64.rpm
```

SUSE の場合:

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

Ubuntu 20.04 の場合:

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

Ubuntu 22.04 の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu22.04_amd64.deb
```

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

マシンアカウントのパスワード更新を自動化する

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

1. /opt/Citrix/VDA/sbin/deploymcs.sh を実行する前に、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 の実行後、/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。

各マシンアカウントのパスワード更新後、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

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

	ウィンバインド	SSSD	セントリファイ	PBIS
RHEL 9.2/9.0	はい	いいえ	いいえ	いいえ
RHEL 8.x	はい	いいえ	いいえ	はい
ロッキー Linux 9.2/9.0	はい	いいえ	いいえ	いいえ
ロッキー Linux 8.x	はい	いいえ	いいえ	いいえ
RHEL 7, CentOS 7	はい	はい	いいえ	はい
ウブントゥ 22.04, ウブントゥ 20.04	はい	いいえ	いいえ	いいえ
デビアン 11.3	はい	いいえ	いいえ	いいえ
SUSE 15.4	はい	いいえ	いいえ	いいえ

テンプレートVMでマスターイメージを準備する際にFASを有効にする

1. ルートCA証明書をインポートします。

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

2. ctxfascfg.shを実行します。
3. `/etc/xdl/mcs/mcs.conf`で変数を設定します。

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

a. `Use_AD_Configuration_Files_Of_Current_VDA`の値をYに設定します。
b. `FAS_LIST`変数をFASサーバーアドレスまたは複数のFASサーバーアドレスに設定します。複数のアドレスはセミコロンで区切り、アドレスを単一引用符で囲みます。例: `FAS_LIST='<FAS_SERVER_FQDN>;<FAS_SERVER_FQDN>'`.
c. 必要に応じて、`VDI_MODE`などの他の変数を設定します。

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

MCSで作成されたVMでFASを有効にする

前述のようにテンプレートマシンでFASが有効になっていない場合、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:**
>
> `/etc/xdl/mcs/mcs.conf`で必要なすべての変数を設定します。これらの変数はVMの起動時に呼び出されるためです。

a. `Use_AD_Configuration_Files_Of_Current_VDA`の値をYに設定します。
b. `FAS_LIST`変数をFASサーバーアドレスに設定します。
c. 必要に応じて、`VDI_MODE`などの他の変数を設定します。

2. ルートCA証明書をインポートします。

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

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

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 22.04
- ウブントゥ 20.04
- RHEL 9.2
- RHEL 9.0
- RHEL 8.8
- RHEL 8.6
- RHEL 7.9
- ロッキー Linux 9.2
- ロッキー Linux 9.0
- ロッキー Linux 8.8
- ロッキー Linux 8.6
- SUSE 15.4

重要:

- Citrix Provisioning の最新のインストールパッケージを使用することをお勧めします。お使いの Linux ディストリビューションに基づいたパッケージを使用してください。Linux ストリーミングエージェント 2109 以降を使用するには、Citrix Provisioning Server 2109 以降が必要です。
- Citrix Provisioning を使用して Linux ターゲットデバイスをストリーミングする場合、プロビジョニングされたデバイスが期待どおりに起動できるように、単一の共有ディスクイメージ上に個別のブートパーティションを作成します。
- **btrfs** でパーティションをフォーマットすることは避けてください。GRUB2 には、**btrfs** パーティションを見つける際に固有の問題があります。**GRUB** は **GRand Unified Bootloader** の略です。

詳細については、Citrix Provisioning のドキュメントのLinux ターゲットデバイスのストリーミングを参照してください。

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

Citrix DaaS Standard for Azure (旧称 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を作成し、Microsoft Azureから任意のデバイスに仮想アプリとデスクトップを配信できます。詳細については、Citrix DaaS Standard for Azureを参照してください。

サポートされているLinuxディストリビューション

以下のLinuxディストリビューションがこの機能をサポートしています。

- RHEL 9.2
- RHEL 9.0
- RHEL 8.8
- RHEL 8.6
- ロッキー Linux 9.2
- ロッキー Linux 9.0
- ロッキー Linux 8.8
- ロッキー Linux 8.6
- SUSE 15.4
- ウブントゥ 22.04
- ウブントゥ 20.04

手順1：Azureでマスターイメージを準備する

> **注:**
>
> 自動ソフトウェアアップデートをスケジュールするために、Linux VDA セルフアップデート機能も使用できます。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. VM に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` の指示に従って .NET Runtime 6.0 をインストールします。
4. (Ubuntu のみ) `source /etc/network/interfaces.d/*` 行を `/etc/network/interfaces` ファイルに追加します。
5. (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
   ```

6. Linux VDA パッケージをインストールします。
7. 使用するデータベースを指定します。

   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、SQLite と PostgreSQL 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。新規インストールである場合は、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。
> - 簡単なインストールと MCS の場合、SQLite と PostgreSQL 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` を介して別途指定されない限り、Linux VDA はデフォルトで PostgreSQL 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。

8. 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。
- 簡単インストールGUIを使用します。簡単インストール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`変数は指定しないままにします。 マシンカタログを作成するときに**Static**または**Random**タイプを選択した場合は、`VDI_MODE=Y`を設定します。

> `/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。

簡単インストールGUIの展開ボタン(/en-us/linux-virtual-delivery-agent/2308/media/deploy-button-on-easy-install-gui.png)

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。

9. 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。

   VHDファイルのURLを生成する(/en-us/linux-virtual-delivery-agent/2308/media/export-disk.png)

10. （オプション）マスターイメージでグループポリシー設定を行います。`ctxreg`ツールを使用してグループポリシー設定を行うことができます。たとえば、次のコマンドはPDF印刷の**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
    ```

ステップ2：Azureからマスターイメージをインポートする

1. Manageダッシュボードから、右側の**Master Images**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:

   このサービスの管理者アクティビティのほとんどは、**Manage**および**Monitor**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、**Manage**ダッシュボードが自動的に起動します。

   マスターイメージの画像(/en-us/linux-virtual-delivery-agent/2308/media/master-images.png)

2. **Import Image**をクリックします。
3. Azureで生成したVHDファイルのSAS URLを入力します。マスターイメージの種類として**Linux**を選択します。

   Azureからイメージをインポートするイメージ(/en-us/linux-virtual-delivery-agent/2308/media/import-an-image-from-azure.png)

4. ウィザードの指示に従って、マスターイメージのインポートを完了します。

ステップ3：マシンカタログを作成する

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

> **注:**
>
> マスターイメージとして使用されるVMは、SSHまたはRDP経由ではアクセスできません。VMにアクセスするには、Azureポータル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。

- アマゾン リナックス 2、セントオーエス、RHEL、および ロッキー リナックス
- SUSE
- ウブントゥ
- デビアン

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

> **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、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。

ステップ1b：ホスト名を設定する

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

`hostname`

ステップ1c：ホスト名にループバックアドレスを割り当てる

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

`127.0.0.1 hostname-fqdn hostname localhost 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（-）文字のみを使用してください。アンダーバー（_）、スペース、およびその他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。この規則は、Delivery Controllerのホスト名にも適用されます。

ステップ1d：ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

手順1e：名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

手順1f：クロック同期の構成

Maintaining accurate clock synchronization between the VDAs, Delivery Controllers, and domain controllers is crucial. Hosting the Linux VDA as a virtual machine (VM) can cause clock skew problems. For this reason, synchronizing time with a remote time service is preferred.

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サービスの構成

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
```

In a typical deployment, synchronize time from the local domain controllers and not directly from public NTP pool servers. Add a server entry for each Active Directory domain controller in the domain.

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

変更を保存し、Chronyデーモンを再起動します。

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

手順1g：PulseAudioのインストール（RHEL 9.2/9.0およびRocky Linux 9.2/9.0のみ）

pulseaudio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 pulseaudio --allowerasing
```

`/etc/pulse/client.conf` を開き、次のエントリを追加します。

```
autospawn = yes
```

ステップ 1h: OpenJDK 11 をインストールする

Linux VDA には OpenJDK 11 が必要です。

- CentOS または RHEL を使用している場合、Linux VDA のインストール時に OpenJDK 11 は依存関係として自動的にインストールされます。
- Amazon Linux 2 または Rocky Linux を使用している場合は、次のコマンドを実行して OpenJDK 11 を有効にしてインストールします。

  ```
  amazon-linux-extras install java-openjdk11
  ```

正しいバージョンを確認します。

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

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

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

ステップ 1i: 使用するデータベースをインストールして指定する

Linux VDA パッケージのインストール後、**/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。手動インストールの場合、それらを指定できるようになる前に、SQLite と PostgreSQL を手動でインストールする必要があります。

このセクションでは、PostgreSQL および SQLite データベースのインストール方法と、使用するデータベースの指定方法について説明します。

> **注:**
>
> VDI モードでのみ SQLite を使用することをお勧めします。

POSTGRESQL をインストールする

Linux VDA には PostgreSQL が必要です。

- Amazon Linux 2、RHEL 7、および CentOS 7 用の PostgreSQL 9
- RHEL 8.x および Rocky Linux 8.x 用の PostgreSQL 10
- RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 用 PostgreSQL 13

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 9.2/9.0 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 を起動するには、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
```

(RHEL 7 および Amazon Linux 2 のみ) **psql** コマンドラインユーティリティを使用して、データディレクトリが設定されていることを確認します。

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

SQLITE のインストール

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
```

使用するデータベースを指定する

SQLite、PostgreSQL、またはその両方をインストールした後、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、使用するデータベースを指定できます。これを行うには、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. **/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。
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。

ステップ2：ハイパーバイザーを準備する

サポートされているハイパーバイザー上でLinux VDAをVMとして実行する場合、いくつかの変更が必要です。使用中のハイパーバイザープラットフォームに基づいて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™での時刻同期を修正する

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

Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかを確認できます。

```
su -



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

このコマンドは0または1を返します。

- 0 - 時刻同期機能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、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、以下の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期を修正する

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利用できます。システムクロックが正確であることを保証するために、NTPサービスと並行してこの機能を有効に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 および Citrix Hypervisor とは異なります。Hyper-V の時刻同期は、NTP の時刻同期と共存し、補完することができます。

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP とハイパーバイザーに関する問題が発生します。どちらもシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と

同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

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

1. vSphere Client を開きます。
2. Linux VM の設定を編集します。
3. **仮想マシンプロパティ**ダイアログで、**オプション**タブを開きます。
4. **VMware Tools**を選択します。
5. **詳細**ボックスで、**ゲストの時刻をホストと同期**のチェックを外します。

ステップ 3: Linux 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、次の方法が利用できます。

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

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名が使用されている場合、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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 をご利用の場合、**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てください。

```
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 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.x のシステムの場合：

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

Amazon Linux 2、CentOS 7、およびRHEL 7の場合:

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

マシンの起動時にWINBINDデーモンを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されている必要があります:

```
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 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.x の場合:

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

   Amazon Linux 2、CentOS 7、および RHEL 7 の場合:

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

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

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

   `--enablekrb5kdcdns --enablekrb5realmdns`

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

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

   `kerberos method = secrets and keytab`

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

3. (RHEL 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.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]` セクションの下に:

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

   `default_realm = REALM` `dns_lookup_kdc = true`

   `[realms]` セクションの下に:

   `REALM = {`

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

   `[domain_realm]` セクションの下に:

   `realm = REALM`

   `.realm = REALM`

Linux VDA は、Delivery Controller で認証および登録するためにシステム keytab 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 keytab ファイルを作成するよう強制されます。

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、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。

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
```

ヒント:

Amazon Linux 2、RHEL 7.9、CentOS 7.9 で実行されている Linux VM の場合、次のコマンドを使用して Windows ドメインに追加することもできます。

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

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

WINBIND の PAM を構成する

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

`krb5_auth = yes`

`krb5_ccache_type = FILE` `mkhomedir = yes`

各設定の先頭にあるセミコロンが削除されていることを確認してください。これらの変更には、Winbind デーモンの再起動が必要です。

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

ヒント:

`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 9.2/9.0 および Rocky Linux 9.2/9.0 の場合、Winbind 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 keytab ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

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

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

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

マシンアカウントの TGT チケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist
```

以下を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

Winbind PAM モジュールが正しく構成されていることを確認するには、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
```

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

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンすることで実行できます。ドメイン参加の確認後、手順 6: Linux VDA のインストール 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して構成済みであり、Active Directory でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーが LINUX VDA マシンにログオンできるようにする

ドメインユーザーが Linux VDA マシンで HDX™ セッションを確立できるようにするには：

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

> **注:**

> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDAでQUESTを構成する

SELinuxポリシーの適用を回避する
デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨害され、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=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およびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、PAMとNSSを手動で構成するために、次のコマンドを実行します。

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



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

WINDOWSドメインに参加する

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

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

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

ドメインメンバーシップの確認

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

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、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ドメインに参加する

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

```
su –

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

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

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン（Windows VDA および Linux VDA）が Active Directory にコンピューターオブジェクトを持つことを要求します。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「starting」状態のままになっている場合、Centrify クライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all

adinfo –diag
```

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

```
adinfo --test
```

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

SSSD

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

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

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

ドメインに参加し、ホストキータブを作成する

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

このセクションでは、Amazon Linux 2およびRHEL 7向けの**Samba**アプローチと、RHEL 8.x/9.xおよびRocky Linux 8.x/9.x向けの`adcli`アプローチについて説明します。**realmd**については、RHELまたはCentOSのドキュメントを参照してください。これらの手順は、SSSDを構成する前に実行する必要があります。

- **Samba (Amazon Linux 2 および RHEL 7):**

  必要なパッケージをインストールまたは更新します。

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

  適切に構成されたファイルを持つLinuxクライアントで:

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

  Samba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Active Directoryドメインの短いNetBIOS名です。

> **注:**
>
> この記事の設定は、単一ドメイン、単一フォレストモデルを対象としています。Kerberosは、ADインフラストラクチャに基づいて構成してください。
>
> KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、前のコマンドに次の2つのオプションを追加します。

`--enablekrb5kdcdns --enablekrb5realmdns`

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

`kerberos method = secrets and keytab`

`winbind offline logon = no`

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

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

REALM は大文字の Kerberos レルム名であり、**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。

- **Adcli (RHEL 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.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。

セクションの下に:

`REALM = {`

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

セクションの下に:

`realm = REALM` `.realm = REALM`

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

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

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

SSSD をセットアップする

SSSD のセットアップは、次の手順で構成されます。

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

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

RHEL 7 の SSSD 設定例(/en-us/linux-virtual-delivery-agent/2308/media/example-sssd-conf-for-rhel7.png)

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

(RHEL 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.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`

sssd.conf のファイルの所有権とアクセス許可を設定します。

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

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

SSSD を有効にする

RHEL 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.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
```

Amazon Linux 2、CentOS 7、および RHEL 7 の場合:

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

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



sudo service sssd start



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。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

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てください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOSサフィックス形式: `username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、以前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

```
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ドメインに参加する

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

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

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

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

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（Windows VDAおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。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 Runtime 6.0をインストールする

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。

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

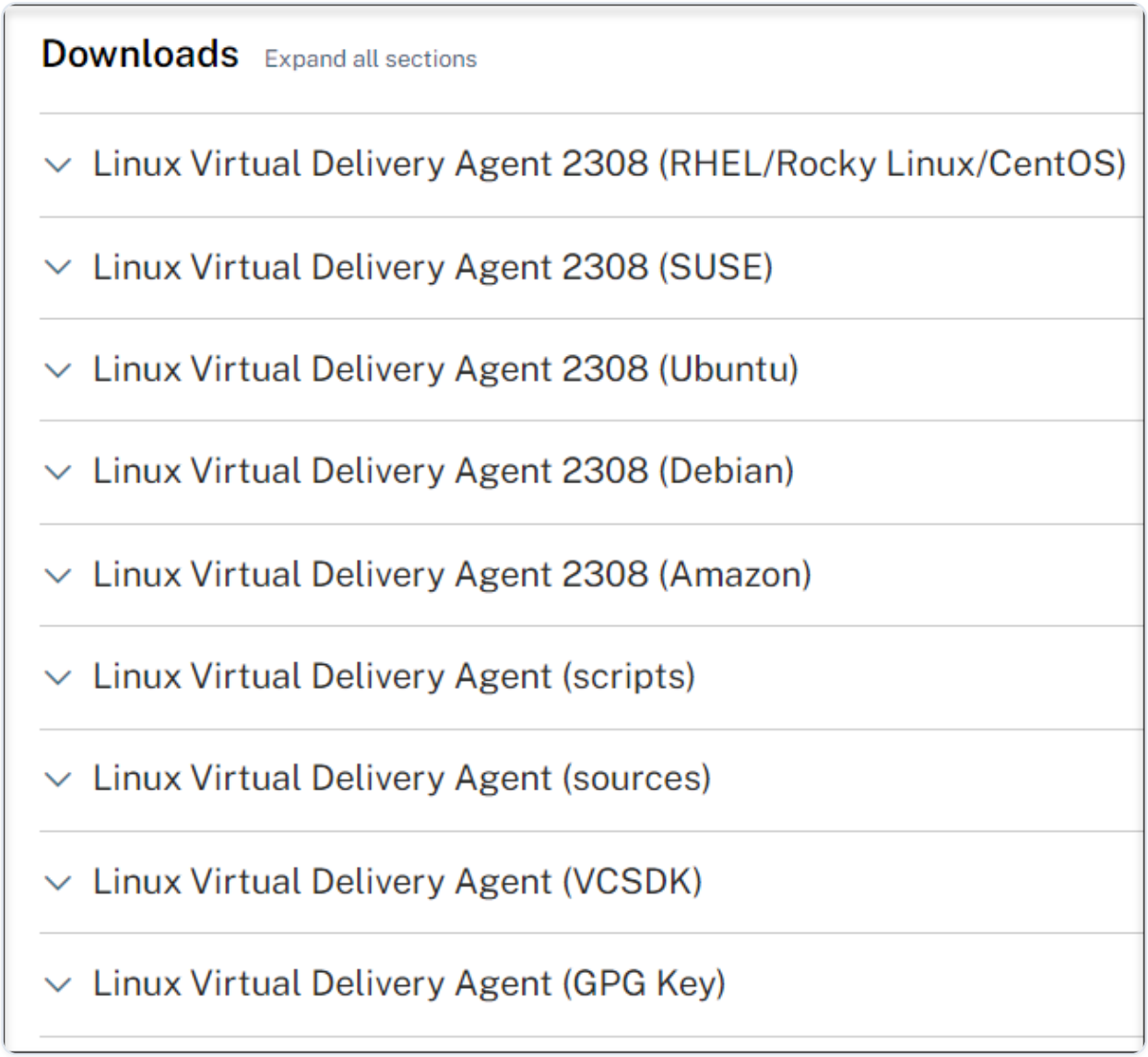
コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

ステップ5：Linux VDAパッケージをダウンロードする

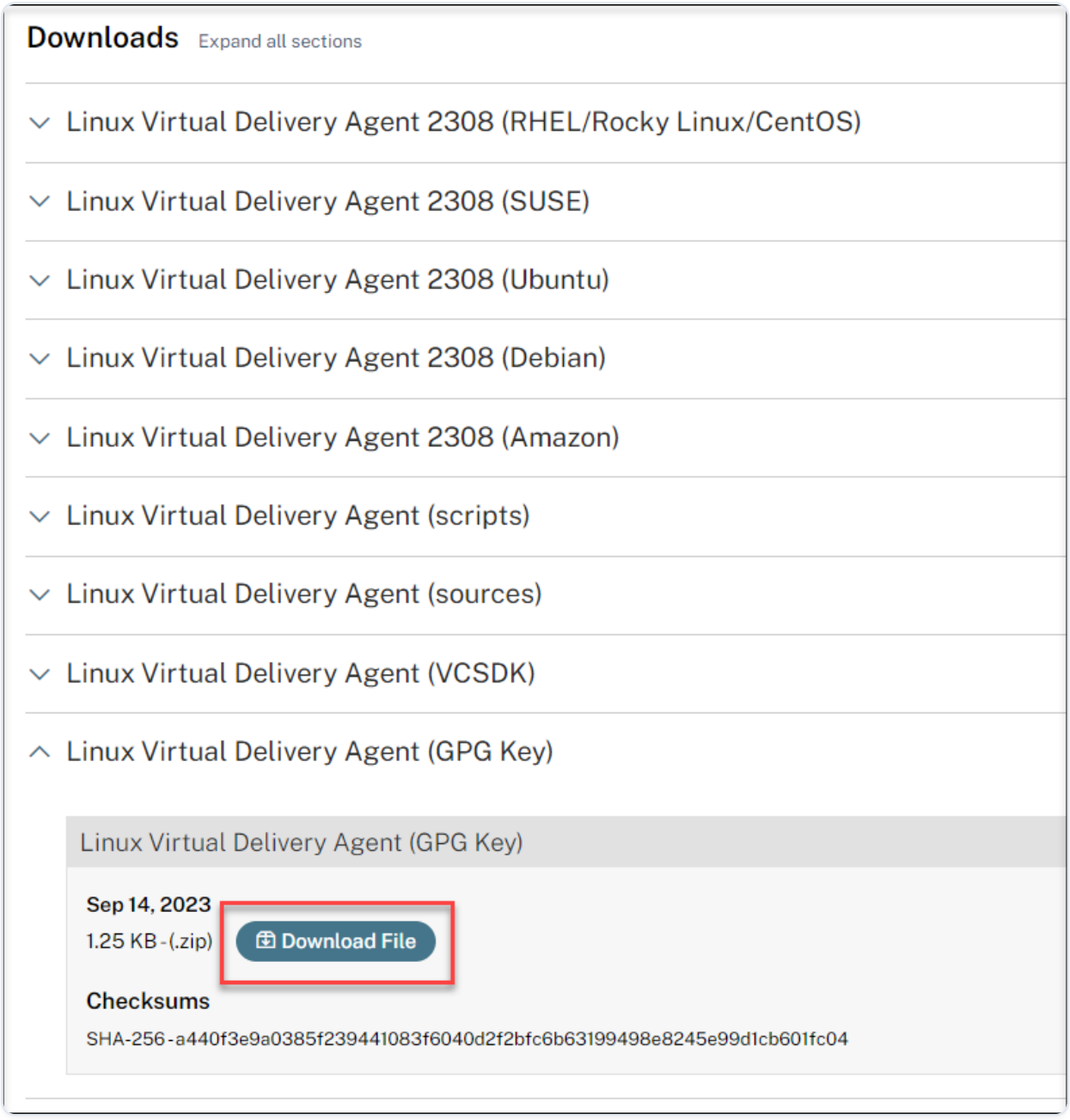
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 適切なバージョンのCitrix Virtual Apps and Desktopsを展開します。
3. **コンポーネント**を展開してLinux VDAを見つけます。例：



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

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

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する

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

   a. Linux VDAサービスを停止します。

   ```
   sudo /sbin/service ctxvda stop



   sudo /sbin/service ctxhdx stop
   ```

   注:

   ctxvda および **ctxhdx** サービスを停止する前に、**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する

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. Linux VDAをインストールする

> **注:**
>
> - CentOS、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 9.2/9.0およびRocky Linux 9.2/9.0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。

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

  Amazon Linux 2の場合:

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

  RHEL 9.2/9.0およびロッキー・リナックス 9.2/9.0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
  ```

  CentOS 7 および RHEL 7 の場合:

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

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

  Amazon Linux 2 の場合:

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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0:

```
sudo rpm -i XenDesktopVDA-<version>.el9_x.x86_64.rpm
```

RHEL 8.x および Rocky Linux 8.x の場合:

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

CentOS 7 および RHEL 7 の場合:

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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 の RPM 依存関係リスト:

```
tzdata-java >= 2022


java-11-openjdk >= 11


 icoutils >= 0.32


 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


foomatic-filters >= 4.0.9


cups-filters >= 1.20.0
```

```
ghostscript >= 9.25


libxml2 >= 2.9


libmspack >= 0.7


krb5-workstation >= 1.13


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
```

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

```
java-11-openjdk >= 11


icoutils >= 0.32


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

foomatic-filters >= 4.0.9

cups-filters >= 1.20.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

krb5-workstation >= 1.13
```

```
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
```

CentOS 7 および RHEL 7 の RPM 依存関係リスト:

```
java-11-openjdk >= 11
```

```
ImageMagick >= 6.7.8.9

firewalld >= 0.3.9

policycoreutils-python >= 2.0.83

dbus >= 1.6.12

dbus-x11 >= 1.6.12

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

xorg-x11-server-Xorg >= 1.20.4

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2
```

```
pam >= 1.1.8


util-linux >= 2.23.2


bash >= 4.2


findutils >= 4.5


gawk >= 4.0


sed >= 4.2


cups >= 1.6.0


foomatic-filters >= 4.0.9


libxml2 >= 2.9


libmspack >= 0.5


ibus >= 1.5
```

```
cyrus-sasl-gssapi >= 2.1


python3 >= 3.6~


gperftools-libs >= 2.4


nss-tools >= 3.44.0


qt5-qtbase >= 5.5~


qt5-qtbase >= 5.5~


imlib2 >= 1.4.5
```

Amazon Linux 2 の RPM 依存関係リスト:

```
java-11-openjdk >= 11


ImageMagick >= 6.7.8.9


firewalld >= 0.3.9


policycoreutils-python >= 2.0.83
```

```
dbus >= 1.6.12

dbus-x11 >= 1.6.12

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

xorg-x11-server-Xorg >= 1.20.4

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

pam >= 1.1.8

util-linux >= 2.23.2

bash >= 4.2
```

```
findutils >= 4.5

gawk >= 4.0

sed >= 4.2

cups >= 1.6.0

foomatic-filters >= 4.0.9

libxml2 >= 2.9

libmspack >= 0.5

ibus >= 1.5

cyrus-sasl-gssapi >= 2.1

gperftools-libs >= 2.4

nss-tools >= 3.44.0
```

```
qt5-qtbase >= 5.5~


qrencode-libs >= 3.4.1


imlib2 >= 1.4.5
```

> **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。
>
> RHEL 7.x に Linux VDA をインストールした後、`sudo yum install -y python-websockify x11vnc` コマンドを実行します。これは、セッションシャドウイング機能を使用するために `python-websockify` と `x11vnc` を手動でインストールすることを目的としています。詳細については、「セッションのシャドウイング」を参照してください。

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）

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

> **注:**
>
> - 既存のインストールをアップグレードすると、/etc/xdl の下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。
> - RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 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。

- `Yum` を使用してソフトウェアをアップグレードするには：

Amazon Linux 2 の場合：

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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 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
```

CentOS 7 および RHEL 7 の場合：

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

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

Amazon Linux 2 の場合：

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

RHEL 9.2/9.0 および Rocky Linux 9.2/9.0 の場合：

```
sudo rpm -U XenDesktopVDA-<version>.el9_x.x86_64.rpm
```

RHEL 8.x および Rocky Linux 8.x の場合：

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

CentOS 7 および RHEL 7 の場合：

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

注:

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

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

重要:

ソフトウェアのアップグレード後、Linux VDA マシンを再起動します。

ステップ 7: NVIDIA GRID ドライバーのインストール

HDX 3D Pro を有効にするには、ハイパーバイザーと VDA マシンに NVIDIA GRID ドライバーをインストールする必要があります。

注:

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

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

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX(https://docs.nvidia.com/vgpu/latest/grid-software-quick-start-guide/index.html#installing-grid-vgpu-manager-vmware-vsphere)
- ニュータニックス AHV

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

1. ゲスト VM がシャットダウンされていることを確認してください。

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. Red Hat Enterprise Linux ドキュメントの手順に従って、NVIDIA GRID ドライバーをインストールします。

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 ('no') 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

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

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

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

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

ステップ8：Linux VDAを構成する

> **注:**

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

> 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を環境変数で指定します。必要なすべての変数が存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。デフォルトではポート80です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux VDAサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux 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。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=winbind \| quest \|centrify \| sssd \| pbis** – Linux VDA が Delivery Controller 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_HDX_3D_PRO=Y \| N** – Linux VDA は、グラフィックを多用するアプリケーションの仮想化を最適化するように設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro を選択すると、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。

- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するかを決定します。HDX 3D Pro 環境の場合は、この変数を Y に設定します。この変数はデフォルトで N に設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS にクエリを実行して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、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、または LDAPS を使用している場合は ad1.mycompany.com:636 ad2.mycompany.com:3269 ad3.mycompany.com:3269 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、または LDAPS を使用している場合は 3269 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（例: DC=mycompany,DC=com）に設定された検索ベースを介して LDAP にクエリを実行します。検索パフォーマンスを向上させるには、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりに FAS サーバーのセミコロン区切りリストを指定できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。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'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために .NET Runtime 6.0 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/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。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現

在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。

a. VDA の **$HOME/<username\>** ディレクトリの下に `.xsession` または `.Xclients` ファイルを作成します。Amazon Linux 2 を使用している場合は、`.Xclients` ファイルを作成します。その他のディストリビューションを使用している場合は、`.xsession` ファイルを作成します。

   2\. `.xsession` または `.Xclients` ファイルを編集して、デスクトップ環境を指定します。

   - **MATE デスクトップの場合**

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


     if [ -n "$MSESSION" ]; then
       exec mate-session


     fi
     ```

   - **GNOME Classic デスクトップの場合**

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


     if [ -n "$GSESSION" ]; then
     export GNOME_SHELL_SESSION_MODE=classic


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

```
fi
```

- **GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

b. 700ファイル権限をターゲットセッションユーザーと共有します。

バージョン2209以降、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ます。この機能を有効にするには、VDAに切り替え可能な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。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成が完了したときにLinux VDAサービスを開始するかどうかを決定します。デフォルト値はYです。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutのリッスン用ソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutとの通信用ポート。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N



export CTX_XDL_DDC_LIST='list-ddc-fqdns'
```

```
export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

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

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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


export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number


export CTX_XDL_TELEMETRY_PORT=port-number


export CTX_XDL_START_SERVICE=Y|N


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

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \
```

```
CTX_XDL_AD_INTEGRATION=winbind | quest |centrify | sssd | pbis \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \
```

```
CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh --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。

ステップ9: XDPingを実行する

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

ステップ10: Linux VDAを実行する

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

Linux VDAを開始する:

Linux VDAサービスを開始するには:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDAを停止する:

Linux VDAサービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

> **注:**
>
> **ctxvda**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 /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart
```

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

Linux VDAのステータスを確認する:

Linux VDAサービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status


sudo /sbin/service ctxhdx status
```

ステップ11: マシンカタログを作成する

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

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

- オペレーティングシステムには、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**［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:**
>
> 削除したマシンを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。

手順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マシンにログオンす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 2308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のために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: 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、ここでは文書化されていない異なるナビゲーション方法があります。

ホスト名とドメインネームシステム (DNS) 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** で **no** オプションを選択します。
5. **Modify DNS Configuration** で **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 <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。このルールは、Delivery Controllerのホスト名にも適用されます。

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。

名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

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設定にいくつかの変更が必要になる場合があります。

SUSE 15.4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。

- OpenJDK 11
- 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

リポジトリを追加する

ほとんどの必要なパッケージは、ImageMagick を除き、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 クライアントをインストールする

Linux VDA と Delivery Controller 間の相互認証のために Kerberos クライアントをインストールします。

```
sudo zypper install krb5-client
```

Kerberos クライアントの構成は、使用される Active Directory 統合アプローチによって異なります。以下の説明を参照してください。

OPENJDK 11 をインストールする

Linux VDA には OpenJDK 11 が必要です。

OpenJDK 11 をインストールするには、次のコマンドを実行します。

```
sudo zypper install java-11-openjdk
```

使用するデータベースをインストールして指定する

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。手動インストールの場合、SQLite と PostgreSQL を指定できるようにするには、それらを事前に手動でインストールする必要があります。

このセクションでは、PostgreSQL および SQLite データベースのインストール方法と、使用するデータベースの指定方法について説明します。

> **注:**
>
> VDI モードでのみ SQLite を使用することをお勧めします。

PostgreSQL をインストールする
`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、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
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
```

使用するデータベースを指定する
SQLite、PostgreSQL、またはその両方をインストールした後、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、使用するデータベースを指定できます。これを行うには、次の手順を完了します。

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. **/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。
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。

ステップ2：ハイパーバイザーを準備する

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

Citrix Hypervisor™での時刻同期を修正する

Citrix Hypervisorの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとCitrix Hypervisorに関する問題が発生します。両方ともシステムクロックを管理しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックを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。

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

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
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. **時刻の同期** が選択されていることを確認します。

> **注:**

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

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、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。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をNTPと同期させてください。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、次の方法が利用できます。

- 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:**
>
> セッションの起動は、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が使用されている場合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、**Network Services**を選択してから、**Windows Domain Membership**を選択します。
2. 次の変更を行います。
    - **Domain or Workgroup**をActive Directoryドメインの名前またはドメインコントローラーのIPアドレスに設定します。ドメイン名が大文字であることを確認してください。
    - **Use SMB information for Linux Authentication**をオンにします。
        - **Create Home Directory on Login**をオンにします。
        - **Single Sign-on for SSH**をオンにします。

- **Offline Authentication**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、**Install**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。**Yes**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では、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。

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keytabファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

マシンアカウントの詳細を調べるには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

Winbind PAMモジュールが正しく構成されていることを確認します。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
```

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

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、手順6: Linux VDAのインストールに進みます。

Quest認証サービス

ドメインコントローラーでの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と仮定します。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

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

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

LINUX VDAでのQUESTの構成

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

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



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

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

PAMとNSS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。そのためには、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るuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します:

```
ls /tmp/krb5cc_uid
```

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
```

GnomeまたはKDEコンソールに直接ログオンして、同様のテストを実行できます。ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

Centrify DirectControl Agentがインストールされている場合、Centrifyの**adjoin**コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます。

```
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内にコンピューターオブジェクトを持つ必要があります。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：

```
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になっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

SSSD

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します
2. SSSD 用に PAM を構成する
3. SSSD をセットアップする
4. SSSD を有効にする
5. ドメインメンバーシップを確認する
6. Kerberos の構成を確認する

7. ユーザー認証を確認する

ドメインに参加し、ホストキー配布ファイルを作成する

SSSD は、ドメインへの参加やシステムキー配布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、以下の手順を完了してください。

1. ネームサービスキャッシュデーモン (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 の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 セクションの下に次のエントリを追加します:

> **注:**
>
> AD インフラストラクチャに基づいて Kerberos を構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]
```

```
    dns_canonicalize_hostname = false

    rdns = false

    default_realm = REALM

    forwardable = true

[realms]

    REALM = {

        kdc = fqdn-of-domain-controller

        default_domain = realm

        admin_server = fqdn-of-domain-controller
    }
[domain_realm]

    .realm = REALM
```

realm は Kerberos レルム名 (例: example.com) です。**REALM** は大文字の Kerberos レルム名 (例: EXAMPLE.COM) です。

5. ルートユーザーとして `/etc/samba/smb.conf` を編集し、**net** ユーティリティがターゲットドメインと通信できるようにします。 セクションの下に次のエントリを追加します:

```
[global]

    workgroup = domain



    client signing = yes



    client use spnego = yes



    kerberos method = secrets and keytab



    realm = REALM



    security = ADS
```

domain は、EXAMPLE のような 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、ターゲットドメインに管理者として認証します。

```
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 keytab ファイルを生成します。

```
net ads join osname="SUSE Linux Enterprise Server" osVersion=15 -U
administrator
```

SSSD 用に PAM を構成する

SSSD 用に PAM を構成する前に、必要なパッケージをインストールまたは更新します。

```
sudo zypper install sssd sssd-ad
```

SSSD を介したユーザー認証のために PAM モジュールを構成し、ユーザーログオン用のホームディレクトリを作成します。

```
sudo pam-config --add  --sss

sudo pam-config --add --mkhomedir
```

SSSD をセットアップする

1. ルートユーザーとして `/etc/sssd/sssd.conf` を編集し、SSSD デーモンがターゲットドメインと通信できるようにします。`sssd.conf` の構成例（必要に応じて追加オプションを追加できます）：

```
[sssd]

    config_file_version = 2

    services = nss,pam

    domains = domain-dns-name



[domain/domain-dns-name]
```

```
    id_provider = ad

    auth_provider = ad

    access_provider = ad

    ad_domain = domain-dns-name

    ad_server = fqdn-of-domain-controller

    ldap_id_mapping = true

    ldap_schema = ad



# Kerberos settings

    krb5_ccachedir = /tmp

    krb5_ccname_template = FILE:%d/krb5cc_%U



# Comment out if the users have the shell and home dir set on the AD
side



    fallback_homedir = /home/%d/%u

    default_shell = /bin/bash



# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available



# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

```
    ad_gpo_access_control = permissive
```

domain-dns-name は、example.com のような DNS ドメイン名です。

> **注:**
>
> **ldap_id_mapping** は true に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングします。そうでない場合、Active Directory は POSIX 拡張機能を提供できる必要があります。Linux セッションでの無効なログオンエラーを防ぐために、**ad_gpo_access_control** は **permissive** に設定されています。`sssd.conf` および `sssd-ad` の man ページを参照してください。

2. `sssd.conf` のファイル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、次のコマンドを実行します。

```
sudo systemctl enable sssd

sudo systemctl start sssd
```

ドメインメンバーシップを確認する

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

```
sudo net ads testjoin
```

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

```
sudo net ads info
```

「KERBEROS の構成を確認する」

「システムキー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、Kerberos の **kinit** コマンドを実行し、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、以下を使用して確認します。」:

```
sudo klist
```

「ユーザー認証を確認する」

「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。」.

「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
```

The **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。
The **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 VDAおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持つことを要求します。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
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

ステップ 4: .NET Runtime 6.0 をインストールする

Linux VDA をインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` の指示に従って .NET Runtime 6.0 をインストールします。

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

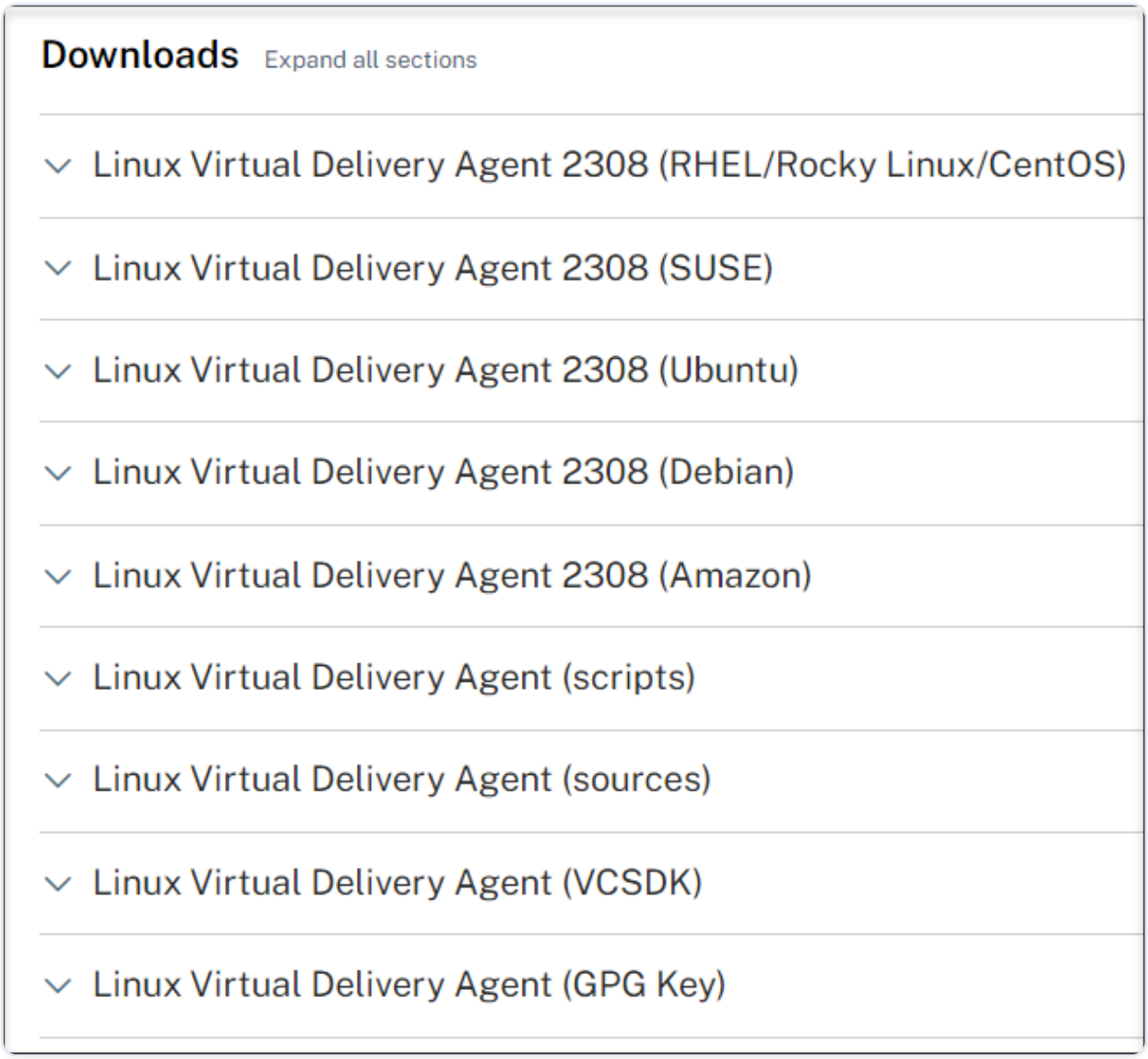
コマンド出力に基づいて、.NET ランタイムバイナリパスを設定します。たとえば、コマンド出力が /aa/bb/dotnet の場合、.NET バイナリパスとして /aa/bb を使用します。

ステップ 5: Linux VDA パッケージをダウンロードする

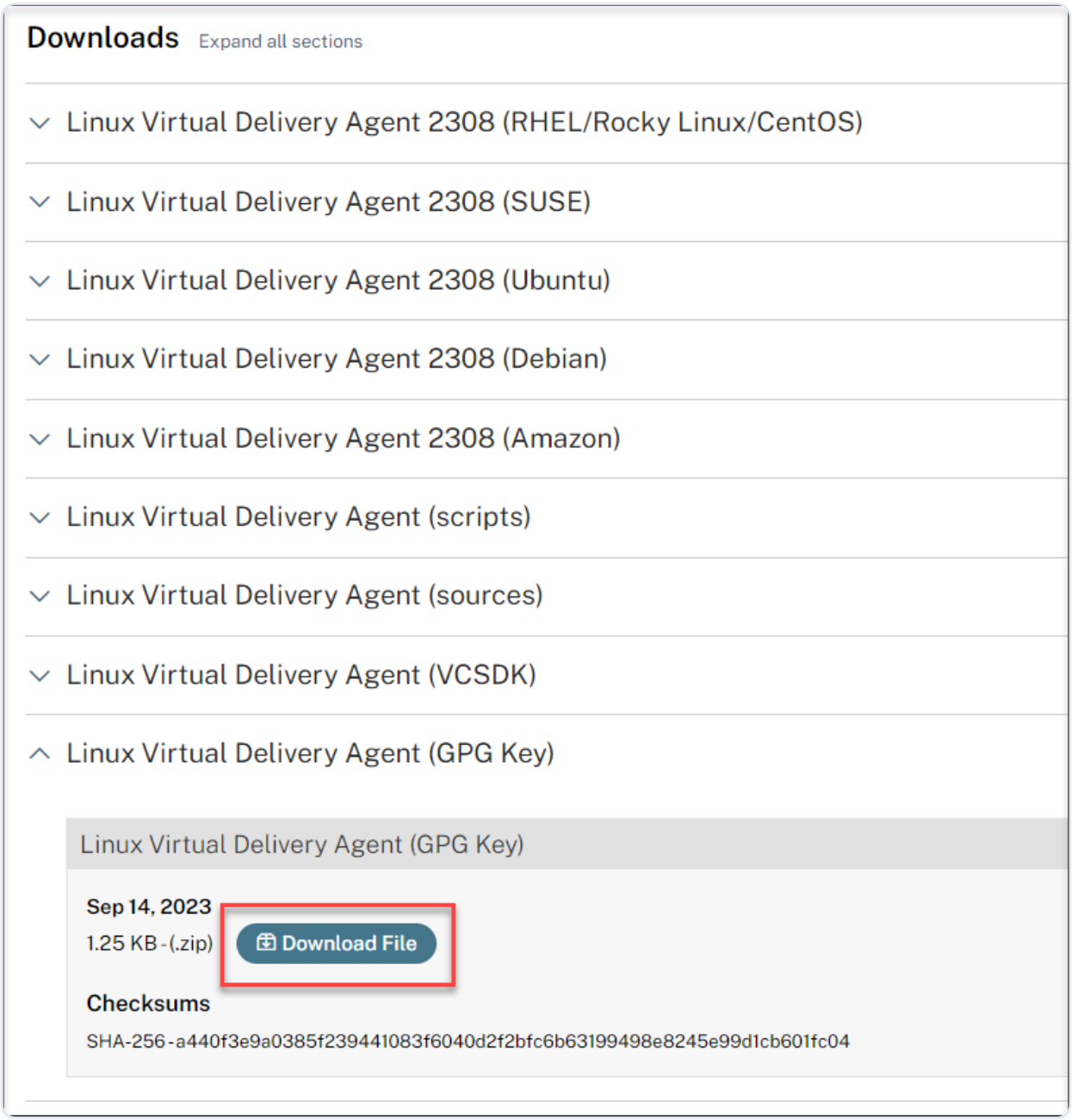
1. Citrix Virtual Apps and Desktops ダウンロードページ にアクセスします。
2. Citrix Virtual Apps and Desktops の適切なバージョンを展開します。
3. Linux VDA を見つけるには、**Components** を展開します。例:

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:



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、次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

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

1. Linux VDAサービスを停止します:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

> **注:**
>
> **ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl 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
```

RPM パッケージマネージャーを使用して Linux VDA ソフトウェアをインストールします:

```
sudo rpm -i 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）

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

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

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

SUSE 15 の RPM 依存関係リスト:

```
java-11-openjdk >= 11


ImageMagick >= 7.0


dbus-1 >= 1.12.2


dbus-1-x11 >= 1.12.2


xorg-x11 >= 7.6_1


libXpm4 >= 3.5.12
```

```
libXrandr2 >= 1.5.1

libXtst6 >= 1.2.3

pam >= 1.3.0

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.4

cups >= 2.2

cups-filters >= 1.25

libxml2-2 >= 2.9

libmspack0 >= 0.6
```

```
ibus >= 1.5

libtcmalloc4 >= 2.5

libcap-progs >= 2.26

mozilla-nss-tools >= 3.53.1

libpython3_6m1_0 >= 3.6~

libQt5Widgets5 >= 5.12

libqrencode4 >= 4.0.0

libImlib2-1 >= 1.4.10
```

重要:

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 ドライバーのインストール

HDX 3D Pro を有効にするには、ハイパーバイザーと VDA マシンに NVIDIA GRID ドライバーをインストールする必要があります。

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

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

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

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

手順 8：Linux VDA の構成

> **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となるオプションを環境変数で指定します。必要な変数がすべて存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用する、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エイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux VDAサービスは、マシンの起動後に開始されます。この値はデフォルトでYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux 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 VDAに必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=winbind \| quest \|centrify \| sssd** – 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_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境では、この変数をYに設定します。この変数はデフォルトでNに設定されています。

- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されます。
- **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ポートを含むスペース区切りのLDAP FQDNのリストを提供できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268、またはLDAPSを使用している場合はad1.mycompany.com:636 ad2.mycompany.com:3269 ad3.mycompany.com:3269です。Active Directoryフォレスト内でより高速なLDAPクエリを有効にするには、ドメインコントローラーで**Global Catalog**を有効にし、関連するLDAPポート番号を3268（LDAPSを使用している場合は3269）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を介して LDAP をクエリします (例: DC=mycompany,DC=com)。検索パフォーマンスを向上させるには、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロンで区切られた FAS サーバーのリストを提供できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。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ているもの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')。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために .NET Runtime 6.0 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/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。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

  ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。

a. VDA の **$HOME/<username\>** ディレクトリの下に `.xsession` ファイルを作成します。

2\. `.xsession` ファイルを編集して、デスクトップ環境を指定します。

- **SUSE 15 の MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **SUSE 15 の GNOME Classic デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **SUSE 15 の GNOME デスクトップの場合**

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



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

```
exec gnome-session



fi
```

b. ターゲットセッションユーザーと 700 ファイル権限を共有します。

バージョン 2209 以降、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ます。この機能を有効にするには、VDA に切り替え可能な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。詳しくは、「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。

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

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N



export CTX_XDL_DDC_LIST='list-ddc-fqdns'



export CTX_XDL_VDA_PORT=port-number



export CTX_XDL_REGISTER_SERVICE=Y|N



export CTX_XDL_ADD_FIREWALL_RULES=Y|N
```

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

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number
```

```
export CTX_XDL_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --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。前のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=winbind | quest |centrify | sssd \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \
```

```
CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh --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を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳しくは、**XDPing**を参照してください。

ステップ10：Linux VDAを実行する

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

Linux VDA を開始する:

Linux VDA サービスを開始するには:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDA を再起動する:

Linux VDA サービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDA のステータスを確認する:

Linux VDA サービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

手順 11: マシンカタログを作成する

マシンカタログを作成し、Linux VDA マシンを追加するプロセスは、従来の Windows VDA のアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログを作成する」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。

注:

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

ヒント:

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

ステップ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 2308」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のために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` への他の参照を削除します。

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

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

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: マルチキャストDNSを無効にする

デフォルト設定ではマルチキャストDNS (**mDNS**) が有効になっており、名前解決の結果が不安定になる可能性があります。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

次のように変更します。

`hosts: files dns`

ステップ1f: 名前解決とサービス到達可能性を確認する

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

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn



nslookup delivery-controller-fqdn



ping delivery-controller-fqdn
```

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

ステップ1g: クロック同期を構成する (chrony)

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

chronyをインストールします。

```
apt-get install chrony
```

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

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

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

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

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

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h: OpenJDK 11をインストールする

Linux VDAにはOpenJDK 11が必要です。

Ubuntu 22.04にはOpenJDK 11が含まれています。

Ubuntu 20.04にOpenJDK 11をインストールするには、次のコマンドを実行します。

```
sudo apt-get install -y openjdk-11-jdk
```

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

Linux VDAパッケージのインストール後、**/etc/xdl/db.conf** を編集することで、SQLiteまたはPostgreSQLを使用するように指定できます。手動インストールの場合、それらを指定できるようになる前に、SQLiteとPostgreSQLを手動でインストールする必要があります。

このセクションでは、PostgreSQLおよびSQLiteデータベースのインストール方法と、使用するデータベースの指定方法について説明します。

> **注:**
>
> VDIモードでのみSQLiteを使用することをお勧めします。

POSTGRESQLをインストールする

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
```

使用するデータベースを指定する

SQLite、PostgreSQL、またはその両方をインストールした後、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、使用するデータベースを指定できます。これを行うには、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. **/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。

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1j: Motifをインストールする

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

手順1k: その他のパッケージをインストールする

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
```

Ubuntu 20.04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-2.4-2

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。

シトリックス ハイパーバイザー™での時刻同期を修正する

シトリックス ハイパーバイザー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とシトリックス 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管理しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます:

```
su -



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

このコマンドは0または1を返します。

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

/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。

```
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は、次の手順を実行します。

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・ワインバインド
- 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
```

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のような DNS ドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字の Kerberos レルム名です。

WINBIND 認証を構成する

Ubuntu には RHEL の **authconfig** や SUSE の yast2 のようなツールがないため、Winbind を手動で構成します。

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: 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 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**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
```

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。

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確認するには、次を使用します。

```
sudo klist
```

マシンのアカウントの詳細を調べるには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっていて正しく動作していることを確認してください。

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

```
ls /tmp/krb5cc_uid
```

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

```
klist
```

セッションを終了します。

```
exit
```

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。ドメイン参加の検証後、手順6：Linux VDAをインストールするに進みます。

> **ヒント:**
>
> ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップ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クエストソフトウェアをインストールして構成済みであり、`Active Directory`でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

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

> **注:**

> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してドメインユーザーのログオン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** の設定を変更します。

`SELINUX=disabled`

この変更にはマシンの再起動が必要です。

```
reboot
```

> **重要:**

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

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

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



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

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

PAM と NSS を構成する

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

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



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

WINDOWS ドメインに参加する

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

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

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

ドメインメンバーシップの確認

Delivery Controller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。Questで参加したLinuxマシンがドメイン上にあることを確認するには:

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用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 Agentがインストールされている状態で、Centrifyの**adjoin**コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます:

```
su –

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

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

ドメインメンバーシップの確認

Delivery Controllerは、WindowsかLinuxかにかかわらず、すべてのVDAマシンが**Active Directory**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
```

ドメインに参加済みの値が有効であり、**CentrifyDCモード**が**接続済み**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
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

SSSD

KERBEROSを構成する

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、rootとして **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**

ADインフラストラクチャに基づいてKerberosを構成します。次の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

`domain-dns-name` パラメーターは、このコンテキストでは example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。

ドメインへの参加

SSSD は、ID プロバイダーとして Active Directory を使用し、認証に Kerberos を使用するように構成する必要があります。ただし、SSSD はドメインへの参加やシステムキーファイル管理のための AD クライアント機能を提供しません。代わりに、**adcli**、**realmd**、または **Samba** を使用できます。

注:

このセクションでは、**adcli** と **Samba** の情報のみを提供します。

- **adcli を使用してドメインに参加する場合は、次の手順を実行します。**

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。

   古いシステムキーファイル (keytab file) を削除し、以下を使用してドメインに参加します。

```
su -



rm -rf /etc/krb5.keytab



adcli join domain-dns-name -U user -H hostname-fqdn
```

   user は、マシンをドメインに追加する権限を持つドメインユーザーです。**hostname-fqdn** は、マシンの FQDN 形式のホスト名です。

   -H オプションは、Linux VDA が必要とする host/hostname-fqdn@REALM 形式で SPN を生成するために **adcli** に必要です。

3. ドメインメンバーシップを確認します。

   Ubuntu 22.04 および Ubuntu 20.04 マシンの場合、`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** を設定します。

   /etc/samba/smb.conf を開いて、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:

まだインストールされていない場合は、必要なSSSDおよび構成パッケージをインストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新が推奨されます。

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

注:

デフォルトでは、Ubuntuのインストールプロセスは **nsswitch.conf** およびPAMログインモジュールを自動的に構成します。

SSSD を構成する

SSSDデーモンを開始する前に、SSSD構成の変更が必要です。SSSDの一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトでインストールされておらず、手動で作成する必要があります。rootとして、**/etc/sssd/sssd.conf** を作成または開いて、次の設定を行います。

```
[sssd]

services = nss, pam

config_file_version = 2

domains = domain-dns-name

[domain/ domain-dns-name ]

id_provider = ad

access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14
days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash
```

`ad_gpo_map_remote_interactive = +ctxhdx`

注:

ldap_id_mapping は **true** に設定され、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 ドメイン名を設定する必要はありません。

構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

SSSD デーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

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

SSSD デーモンの開始

SSSD デーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM の構成

次のコマンドを実行し、**SSS 認証**と**ログイン時にホームディレクトリを作成**のオプションが選択されていることを確認してください。

```
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。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。ドメイン参加の検証後、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

```
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

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

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

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

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

ドメインメンバーシップの確認

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

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

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。そうでない場合は、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。

```
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をインストールするに進みます。

手順4: .NET Runtime 6.0をインストールする

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。

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

コマンド出力に基づいて、.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. Linux VDAを見つけるには、**Components**を展開します。例：

   Citrix Virtual Apps and Desktopsのコンポーネント(/en-us/linux-virtual-delivery-agent/2308/media/components-for-citrix-virtual-apps-and-desktops.png)
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。

   Linux VDAダウンロード(/en-us/linux-virtual-delivery-agent/2308/media/linux-vda-downloads.png)
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：

   GPG公開鍵(/en-us/linux-virtual-delivery-agent/2308/media/gpg-key.png)

   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 20.04では、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 22.04のDebian依存関係リスト:

```
openjdk-11-jdk >= 11



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

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
```

Ubuntu 20.04のDebian依存関係リスト:

```
openjdk-11-jdk >= 11

imagemagick >= 8:6.9.10

libgtkmm-3.0-1v5 >= 3.24.2
```

```
ufw >= 0.36

ubuntu-desktop >= 1.450

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.34

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.7.0

sed >= 4.7

cups >= 2.3
```

```
libmspack0 >= 0.10

ibus >= 1.5

libgoogle-perftools4 >= 2.7~

libpython3.8 >= 3.8~

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

libnss3-tools >= 2:3.49

libqt5widgets5 >= 5.7~

libqrencode4 >= 4.0.0

libimlib2 >= 1.6.1
```

注:

このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、システム要件を参照してください。

ステップ6b: Linux VDAのアップグレード（オプション）

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

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

注:

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

ステップ7: NVIDIA GRIDドライバーのインストール

HDX 3D Proを有効にするには、ハイパーバイザーとVDAマシンにNVIDIA GRIDドライバーをインストールする必要があります。

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

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

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

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

ステップ 8: Linux VDA を構成する

注:

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

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

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

プロンプトによる構成

プロンプトによ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は環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求しないため、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。デフォルトでは N に設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller への登録に使用する Delivery Controller の完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも 1 つの FQDN または CNAME エイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポート（デフォルトではポート 80）を介して Delivery Controller と通信します。

- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux VDA サービスは、マシンの起動後に開始されます。デフォルトでは Y に設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux 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 VDA のシステムファイアウォールで、必要なポート（デフォルトではポート 80 および 1494）を自動的に開くことができます。デフォルトでは Y に設定されています。
- **CTX_XDL_AD_INTEGRATION=winbind \| quest \|centrify \| sssd \| pbis** – 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_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境では、この変数を Y に設定します。この変数はデフォルトで N に設定されます。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS にクエリを実行して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、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。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268、または LDAPS を使用している場合は ad1.mycompany.com:636 ad2.mycompany.com:3269 ad3.mycompany.com:3269 です。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 **Global Catalog** を有効にし、関連する LDAP ポート番号を 3268、または LDAPS を使用している場合は 3269 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。ただし、検索パフォーマンスを向上させるために、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで **\<none\>** に設定されています。

- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロン区切りの FAS サーバーのリストを提供できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。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ている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'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために .NET Runtime 6.0 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/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。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数値を **mate** に設定する必要があります。

  ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。

  a. VDA 上の **$HOME/<username\>** ディレクトリの下に `.xsession` ファイルを作成します。

  2\. `.xsession` ファイルを編集して、ディストリビューションに基づいてデスクトップ環境を指定します。

  - **MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **GNOME Classic デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

b. 700 ファイル権限をターゲットセッションユーザーと共有します。

バージョン 2209 以降、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ます。この機能を有効にするには、VDA に切り替え可能な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。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDA の構成が完了したときに Linux VDA サービスを開始するかどうかを制御します。デフォルトでは Y に設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scout のリッスンに使用するソケットポート。デフォルトのポートは 7503 です。

- **CTX_XDL_TELEMETRY_PORT** – Citrix Scout との通信に使用するポート。デフォルトのポートは 7502 です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N


export CTX_XDL_DDC_LIST='list-ddc-fqdns'


export CTX_XDL_VDA_PORT=port-number


export CTX_XDL_REGISTER_SERVICE=Y|N


export CTX_XDL_ADD_FIREWALL_RULES=Y|N


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


export CTX_XDL_HDX_3D_PRO=Y|N


export CTX_XDL_VDI_MODE=Y|N


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


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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \
```

```
CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

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

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \
```

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


CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \


CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \


CTX_XDL_TELEMETRY_PORT=port-number \


CTX_XDL_START_SERVICE=Y|N \


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

ステップ10：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成したら、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、**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、再度追加する必要があります。

ステップ12: デリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2308」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: 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 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 (-) の文字のみを使用してください。アンダースコア (_)、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは、Delivery Controller のホスト名にも適用されます。

ステップ 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、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を解決できること、および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 (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、リモートタイムサービスとの時刻同期が推奨されます。

chronyをインストールします:

```
apt-get install chrony
```

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

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

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

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

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

変更を保存し、Chronyデーモンを再起動します:

```
sudo systemctl restart chrony
```

ステップ 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
```

Step 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 11.3 の場合、`deb http://deb.debian.org/debian/ bullseye main` の行を /etc/apt/sources.list ファイルに追加します。

Step 1i: 使用するデータベースをインストールして指定する

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。手動インストールの場合、それらを指定する前に SQLite と PostgreSQL を手動でインストールする必要があります。

このセクションでは、PostgreSQL および SQLite データベースのインストール方法と、使用するデータベースの指定方法について説明します。

> **注:**
>
> VDI モードでのみ SQLite を使用することをお勧めします。

POSTGRESQL をインストールする

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、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
```

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、/etc/xdl/db.conf を編集して使用するデータベースを指定できます。これを行うには、次の手順を完了します。

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。
3. 「**ctxsetup.sh**」を実行します。

> **注:**
>
> 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。

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は、変更は不要です。

Citrix Hypervisor™ での時刻同期の修正

Citrix Hypervisor の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP と Citrix Hypervisor の問題が発生します。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。

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

```
su -

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

このコマンドは 0 または 1 を返します。

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

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

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

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

この変更を再起動後も永続的にするには、**/etc/sysctl.conf ファイル**を編集し、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、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 および Citrix Hypervisor とは異なります。Hyper-V の時刻同期は、NTP の時刻同期と共存し、補完することができます。

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP とハイパーバイザーに関する問題が発生します。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. **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 Winbind
- セントリファイ ダイレクトコントロール(/ja-jp/linux-virtual-delivery-agent/2308/installation-overview/manual-installation-overview/debian.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

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、シングルドメイン、シングルフォレストモデルを対象としています。

`[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

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: files systemd winbind`

`group: files systemd winbind`

WINDOWSドメインに参加

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

```
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

```
sudo systemctl restart winbind
```

WINBINDのPAMを構成

次のコマンドを実行し、**Winbind NT/Active Directory authentication**と**Create home directory on login**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

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

ドメインメンバーシップを確認

Delivery Controllerでは、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必要があります。

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構成を確認

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確認するには、次を使用します。

```
sudo klist
```

以下を使用してマシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっており、正しく機能していることを確認してください。

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

```
ls /tmp/krb5cc_uid
```

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

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンして実行できます。ドメイン参加の確認後、手順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再試行してください。

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は、Centrify `adjoin` コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su -

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

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

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを要求します。Centrify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、次の手順を実行します。

```
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を構成する

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、rootとして **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]
```

`default_realm = REALM`

`dns_lookup_kdc = false`

`rdns = false`

`[realms]`

`REALM = {`

`admin_server = domain-controller-fqdn`

`kdc = domain-controller-fqdn`

`}`

`[domain_realm]`

`domain-dns-name = REALM`

`.domain-dns-name = REALM`

このコンテキストでの`domain-dns-name`パラメーターは、example.comのようなDNSドメイン名です。*REALM*は、EXAMPLE.COMのような大文字のKerberosレルム名です。

ドメインへの参加

SSSDは、IDプロバイダーとしてActive Directoryを、認証にKerberosを使用するように構成する必要があります。ただし、SSSDはドメインへの参加やシステムキーファイル管理のための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. システムキーファイルを検証します。

   システムキー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、更新することをお勧めします。

```
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` は ad_gpo_map_remote_interactive に追加されます。

> このコンテキストでの **domain-dns-name** パラメーターは、example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。NetBIOS ドメイン名を設定する必要はありません。

> 構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

> SSSD デーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

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

SSSD デーモンの開始

SSSD デーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM の構成

次のコマンドを実行し、**SSS authentication** と **Create home directory on login** 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構成を確認する

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

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

ユーザー認証を確認する

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返されたKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

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

```
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、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、ステップ6: Linux VDAをインストールするに進みます。

ステップ4: .NET Runtime 6.0をインストールする

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。

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

コマンド出力に基づいて、.NETランタイムの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。

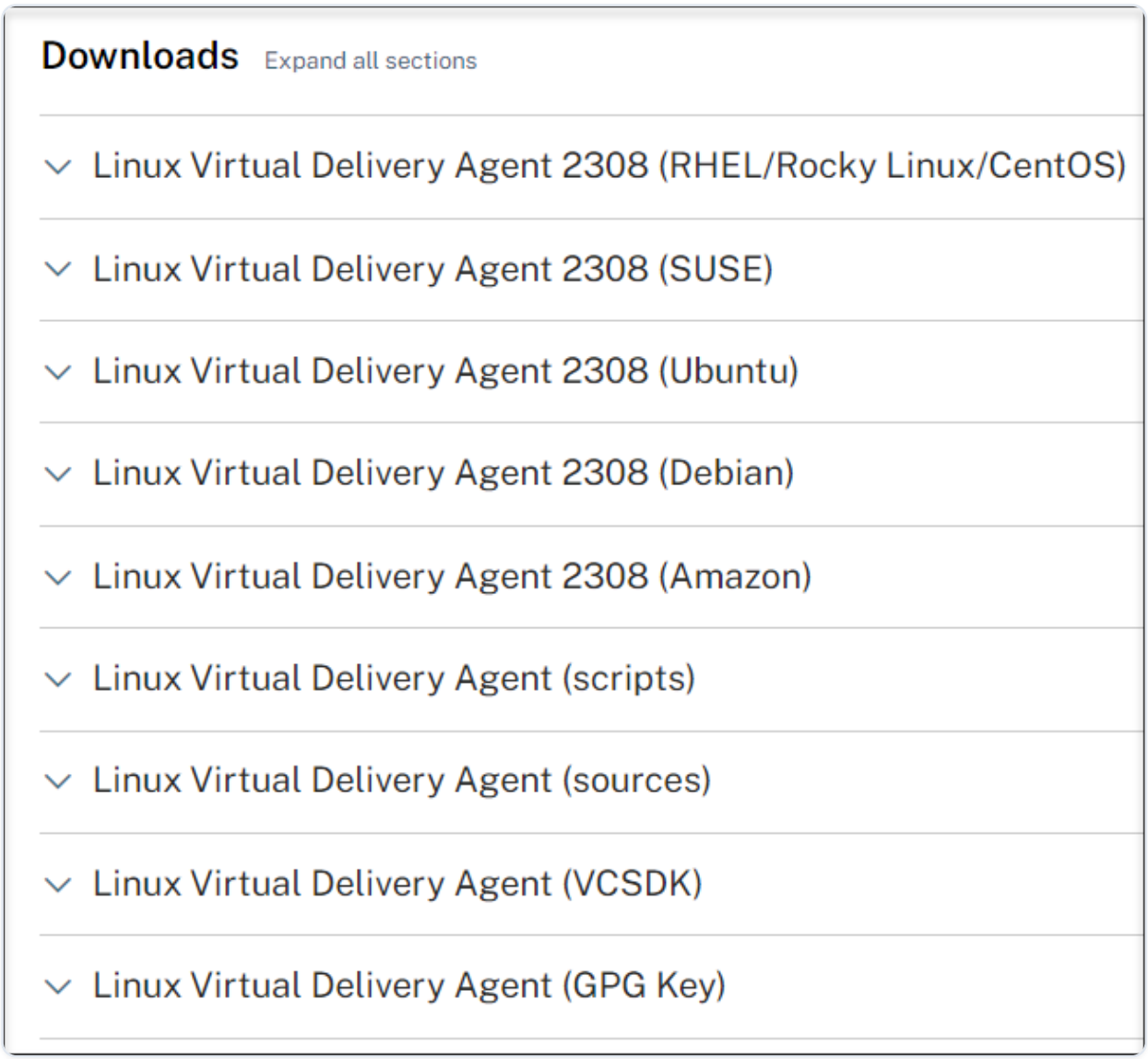
ステップ5：Linux VDAパッケージをダウンロードする

1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。

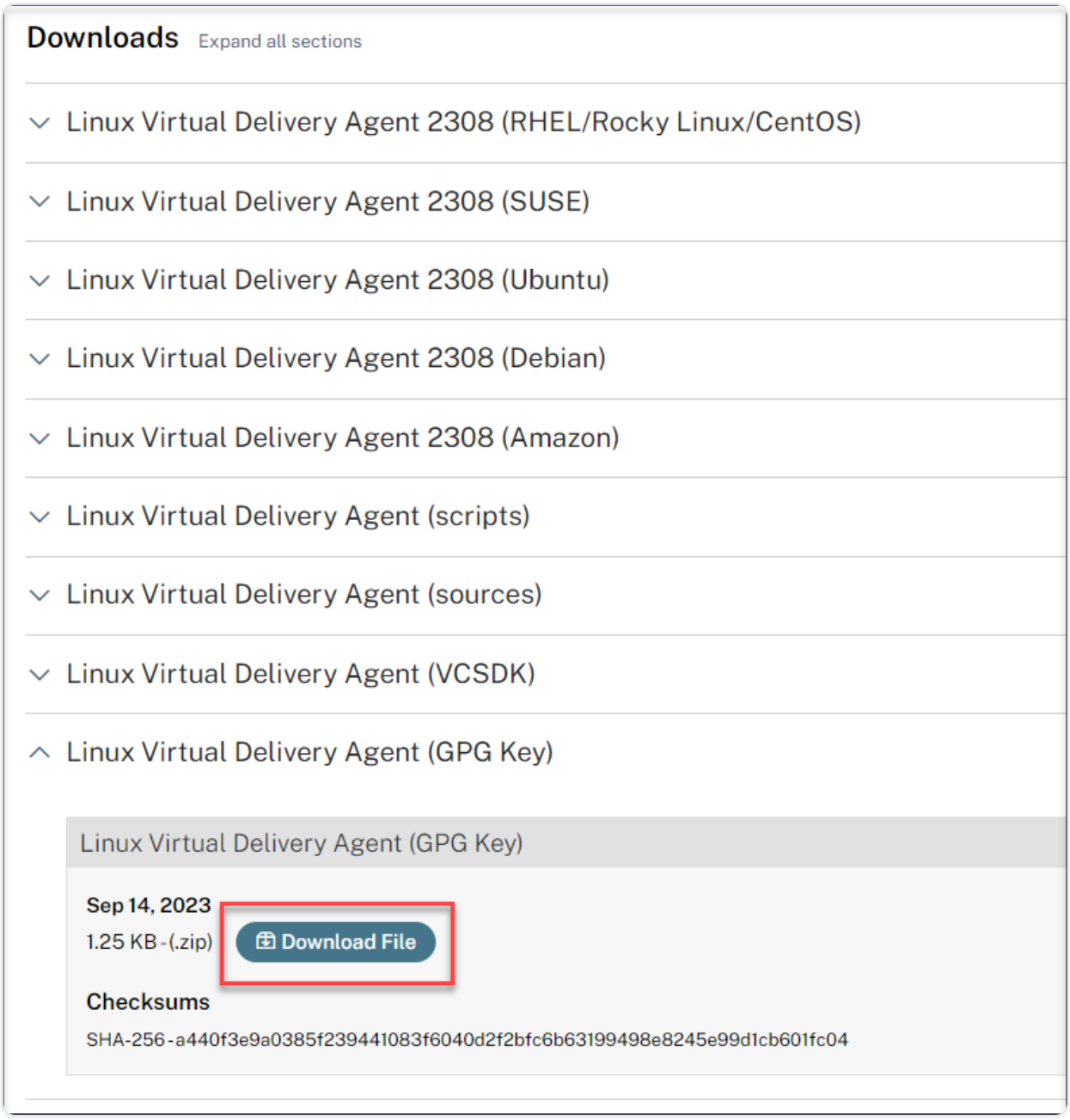
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. Linux VDAを見つけるには、**Components**を展開します。例：



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 xendesktopvda_<version>.debian10_amd64.deb
```

Debian 11.3 の依存関係リスト:

```
openjdk-11-jdk >= 11


imagemagick >= 8:6.9.10


ufw >= 0.36


desktop-base >= 10.0.2


libxrandr2 >= 2:1.5.1


libxtst6 >= 2:1.2.3


libxm4 >= 2.3.8


util-linux >= 2.33


gtk3-nocsd >= 3


bash >= 5.0
```

```
findutils >= 4.6.0

sed >= 4.7

cups >= 2.2

ghostscript >= 9.53~

libmspack0 >= 0.10

ibus >= 1.5

libgoogle-perftools4 >= 2.7~

libpython3.9 >= 3.9~

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

libqt5widgets5 >= 5.5~

mutter >= 3.38.6~
```

```
libqrencode4 >= 4.0.0


libimlib2 >= 1.5.1
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

手順 6b: Linux VDA のアップグレード (オプション)

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

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

注:

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

手順 7: NVIDIA GRID ドライバーのインストール

HDX™ 3D Pro を有効にするには、ハイパーバイザーと VDA マシンに NVIDIA GRID ドライバーをインストールする必要があります。

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

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

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

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

手順 8: Linux VDA の構成

> **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** コマンドを実行してください。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** コマンドを実行してください。

> 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。変更を行う前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

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

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

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求しないため、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。デフォルトでは N に設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller に登録するために、スペースで区切られた Delivery Controller の完全修飾ドメイン名 (FQDN) のリストを必要とします。少なくとも1つの FQDN または CNAME エイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。デフォルトではポート 80 です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux VDA サービスは、マシンの起動後に開始されます。デフォルトでは Y に設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux 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。必要なポート（デフォルトではポート 80 と 1494）を Linux VDA のシステムファイアウォールで自動的に開くことができます。デフォルトでは Y に設定されています。
- **CTX_XDL_AD_INTEGRATION=winbind \| quest \|centrify \| sssd \| pbis** – 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_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- 「**CTX_XDL_VDI_MODE=Y \| N**」 – マシンを専用デスクトップ配信モデル (VDI) として構成するか、ホスト型共有デスクトップ配信モデルとして構成するかを指定します。HDX 3D Pro 環境の場合は、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- 「**CTX_XDL_SITE_NAME=dns-name**」 – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。

- 「**CTX_XDL_LDAP_LIST='list-ldap-servers'**」 – Linux VDAはDNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合は、LDAPポートを含むLDAP FQDNのスペース区切りリストを指定できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268、またはLDAPSを使用している場合はad1.mycompany.com:636 ad2.mycompany.com:3269 ad3.mycompany.com:3269です。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（LDAPSを使用している場合は3269）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。ただし、検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。
- 「**CTX_XDL_FAS_LIST='list-fas-servers'**」 – フェデレーション認証サービス (FAS) サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていませんが、代わりにFASサーバーのセミコロン区切りリストを指定できます。シーケンスはADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白をテキスト文字列**\<none\>**で埋め、サーバーアドレスの順序を変更しないでください。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ているものと一致するポート番号を付加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- 「**CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime**」 – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために.NET Runtime 6.0をインストールするパス。デフォルトのパスは/usr/binです。
- 「**CTX_XDL_DESKTOP _ENVIRONMENT=gnome/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。この変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、変数の値を**mate**に設定する必要があります。

  ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。

  a. VDA上の**$HOME/<username\>**ディレクトリの下に `.xsession` ファイルを作成します。

2\. ディストリビューションに基づいてデスクトップ環境を指定するために、`.xsession` ファイルを編集します。

- **MATEデスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **GNOME Classicデスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **GNOMEデスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session
```

```
fi
```

b. ターゲットセッションユーザーと700ファイル権限を共有します。

バージョン2209以降、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ます。この機能を有効にするには、VDAに切り替え可能な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。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDA の構成が完了したときに Linux VDA サービスを開始するかどうか。デフォルトは Y です。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scout のリッスン用ソケットポート。デフォルトポートは 7503 です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scout との通信用ポート。デフォルトポートは 7502 です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N


export CTX_XDL_DDC_LIST='list-ddc-fqdns'


export CTX_XDL_VDA_PORT=port-number


export CTX_XDL_REGISTER_SERVICE=Y|N


export CTX_XDL_ADD_FIREWALL_RULES=Y|N


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

```
export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number
```

```
export CTX_XDL_START_SERVICE=Y|N


sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --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。前のコマンドから、最初の行に **#\!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \


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


CTX_XDL_HDX_3D_PRO=Y|N \


CTX_XDL_VDI_MODE=Y|N \
```

```
CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh --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、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マシンにログオンするように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

マシンカタログとデリバリーグループの作成方法については、Citrix Virtual Apps and Desktops 7 2308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、その説明、構成、トラブルシューティングを含めて詳しく説明します。

管理

このセクションには、次のトピックが含まれています。

- CEIP
- HDXインサイト
- Citrix Telemetry Serviceとの統合
- Citrix DaaS Standard for Azure向けLinux VDAのセルフアップデート
- Linux VMとLinuxセッションのメトリック
- ログ収集
- セッションシャドウイング
- モニターサービスデーモン
- ツールとユーティリティ
- その他
  - HTML5向けCitrix Workspaceアプリのサポート
  - **Python3**仮想環境の作成
  - NISとActive Directoryの統合
  - **IPv6**
  - LDAPS
  - **Xオーソリティ**

Citrixカスタマーエクスペリエンス向上プログラム (CEIP)

CEIPに参加すると、Citrix製品の品質とパフォーマンスを向上させるために、匿名の統計情報と使用状況情報がCitrixに送信されます。さらに、匿名データのコピーは、迅速かつ効率的な分析のためにGoogle Analytics (GA) に送信されます。GAはデフォルトで無効になっています。

レジストリ設定

デフォルトでは、Linux VDAをインストールするとCEIPに自動的に参加します。データの初回アップロードは、Linux VDAのインストール後約7日で行われます。このデフォルト設定はレジストリで変更できます。

- **CEIPスイッチ**

CEIPを有効または無効にするレジストリ設定 (デフォルト = 0):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: **CEIPSwitch**

値: 1 = 無効、0 = 有効

指定しない場合、CEIPは有効になります。

クライアントで次のコマンドを実行してCEIPを無効にできます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

- **GAスイッチ**

GAを有効または無効にするレジストリ設定 (デフォルト = 1):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: **GASwitch**

値: 1 = 無効, 0 = 有効

指定されていない場合、GA は無効になります。

GA を有効にするには、クライアントで次のコマンドを実行できます:

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

- **DataPersistPath**

データ永続化パスを制御するレジストリ設定 (デフォルト = /var/xdl/ceip):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: DataPersistPath

値: 文字列

このパスを設定するには、次のコマンドを実行できます:

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

構成されたパスが存在しないかアクセスできない場合、データはデフォルトパスに保存されます。

Linux VDA から収集された CEIP データ

次の表は、収集される匿名情報の種類の例を示します。このデータには、お客様を特定するような詳細は含まれていません。

データポイント	キー名	説明
マシン GUID	**machine_guid**	データの発信元であるマシンを識別するGUID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
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
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たことを示す文字列。

データポイント	キー名	説明
HDX™ 3D Proが有効かどうか	**hdx_3d_pro**	HDX 3D Proがマシンで有効になっているかどうかを示す文字列。
VDIモードが有効かどうか	**vdi_mode**	VDIモードが有効になっているかどうかを示す文字列。
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。
LVDAキーサービスの最終再起動時刻	**シーティーエックスエイチディーエックス シーティーエックスブイディーエー**	`ctxhdx`および`ctxvda`サービスの最終再起動時刻。形式はdd-hh:mm:ssで、例: 10-17:22:19
GPUタイプ	**gpu_type**	マシンのGPUタイプを示す
CPUコア	**cpu_cores**	マシンのCPUコア数を示す整数
CPU周波数	**cpu_frequency**	CPU周波数をMHzで示す浮動小数点数
物理メモリサイズ	**memory_size**	物理メモリサイズをKBで示す整数
起動されたセッション数	**session_launch**	このデータポイントを収集した時点でマシン上で起動された（ログオンまたは再接続された）セッション数を示す整数
Linux OS名とバージョン	**os_name_version**	マシンのLinux OS名とバージョンを示す文字列
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する
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:

データポイント	キー名	説明
		デスクトップまたは `<appname>`
アクティブセッション時間	**active_session_time**	セッションのアクティブ時間を保存するために使用されます。セッションは切断/再接続できるため、1つのセッションに複数のアクティブ時間がある場合があります
セッション継続時間	**session_duration_time**	ログオンからログオフまでのセッション継続時間を保存するために使用されます
Receiverクライアントタイプ	**receiver_type**	セッションの起動に使用されたCitrix Workspace™アプリの種類を示す整数
Receiverクライアントのバージョン	**receiver_version**	セッションの起動に使用されたCitrix Workspaceアプリのバージョンを示す文字列
印刷回数	**printing_count**	セッションが印刷機能を使用する回数を示す整数
USBリダイレクト回数	**usb_redirecting_count**	セッションがUSBデバイスを使用する回数を示す整数
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回数	**scard_redirecting_count**	セッションログオンおよびセッション内アプリのユーザー認証にスマートカードリ

データポイント	キー名	説明
		ダイレクトが使用された回数を示す整数

HDX™ インサイト

概要

Linux VDA は、HDX Insight 機能を部分的にサポートしています。

インストール

依存パッケージをインストールする必要はありません。

使用方法

HDX Insight は、Citrix Workspace™ app と Linux VDA の間で Citrix ADC を介して渡される ICA® メッセージを分析します。すべての HDX Insight データは NSAP 仮想チャネルから取得され、非圧縮で送信されます。NSAP 仮想チャネルはデフォルトで有効になっています。

次のコマンドは、NSAP 仮想チャネルをそれぞれ無効化および有効化します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000000" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000001" --force
```

トラブルシューティング

データポイントが表示されない

原因は2つ考えられます。

- HDX Insight が正しく構成されていません。

  たとえば、Citrix ADC で AppFlow® が有効になっていないか、Citrix ADM で誤った Citrix ADC インスタンスが構成されています。

- Linux VDA で ICA Control Virtual Channel が起動していません。

`ps aux | grep -i ctxctl`

`ctxctl` が実行されていない場合は、管理者に連絡して Citrix にバグを報告してください。

アプリケーションデータポイントが表示されない

シームレス仮想チャネルが有効になっており、シームレスアプリケーションが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

Citrix Telemetry Service (`ctxtelemetry`) が Linux VDA ソフトウェアと統合されているため、Citrix Scout を実行できます。Citrix Scout は、/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します。

Citrix スカウト(/en-us/linux-virtual-delivery-agent/2308/media/citrix-scout.png)

> **注:**
>
> Linux VDA 1912 以前のバージョンからアップグレードした後、Citrix Telemetry Service (`ctxtelemetry`) の変数を構成するには、/opt/Citrix/VDA/sbin/ctxsetup.sh を再実行する必要があります。変数について詳しくは、「Easy install」を参照してください。

Citrix Telemetry Service の有効化と無効化

- サービスを有効にするには、**sudo systemctl enable ctxtelemetry.socket** コマンドを実行します。
- サービスを無効にするには、**sudo systemctl disable ctxtelemetry.socket** を実行します。

ポート

Citrix Telemetry Service (`ctxtelemetry`) は、デフォルトで TCP/IP ポート 7503 を使用して Citrix Scout をリッスンします。Delivery Controller 上では TCP/IP ポート 7502 を使用して Citrix Scout と通信します。

Linux VDA のインストール時に、デフォルトのポートを使用するか、以下の変数を使用してポートを変更できます。

- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scout をリッスンするためのソケットポート。デフォルトのポートは 7503 です。

- **CTX_XDL_TELEMETRY_PORT** – Citrix Scout と通信するためのポート。デフォルトのポートは 7502 です。

VDA のインストール後にポートを変更するには、次の手順を実行します。

1. Scout と通信するためのポートを変更するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\Software\Citrix\VirtualDesktopAgent" -v "TelemetryServicePort" -d
   <port number> -t REG_DWORD
   ```

2. Scout をリッスンするためのソケットポートを変更するには、次のコマンドを実行して ctxtelemetry.socket ファイルを開き、編集します。

   ```
   sudo vi /etc/systemd/system/ctxtelemetry.socket
   ```

   ソケットポートの画像(/en-us/linux-virtual-delivery-agent/2308/media/socket-port.png)

3. ソケットポートを再起動するには、次のコマンドを実行します。

   ```
   sudo systemctl daemon-reload

   sudo systemctl stop ctxtelemetry.socket

   sudo systemctl start ctxtelemetry.socket
   ```

4. ファイアウォール構成で新しいポートを有効にします。

   たとえば、Ubuntu を使用している場合は、ポート 7503 を有効にするために **sudo ufw allow 7503** コマンドを実行します。

デバッグモード

Citrix Telemetry Service が期待どおりに動作しない場合は、デバッグモードを有効にして原因を特定できます。

1. デバッグモードを有効にするには、次のコマンドを実行して `ctxtelemetry` ファイルを開き、DebugMode の値を 1 に変更します。

   ```
   sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
   ```

```
#!/bin/sh

export PATH=/usr/lib/jvm/java-8-openjdk-amd64/jre/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin:${PATH}
# Set this flag to 1 to enter debugging mode
DebugMode=1
# Set this flag to 1 to enter interactive debugging mode
InteractiveDebugMode=0
```

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。

```
administrator@RGQBE-LVDA-3:~$ sudo netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      1447/smbd
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      971/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1309/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      25158/cupsd
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      998/postgres
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      1447/smbd
tcp6       0      0 :::2598                 :::*                    LISTEN      28100/ctxhdx
tcp6       0      0 :::139                  :::*                    LISTEN      1447/smbd
tcp6       0      0 :::7502                 :::*                    LISTEN      1958/java
tcp6       0      0 :::7503                 :::*                    LISTEN      1/init
tcp6       0      0 :::80                   :::*                    LISTEN      1610/java
tcp6       0      0 :::1494                 :::*                    LISTEN      28100/ctxhdx
tcp6       0      0 :::22                   :::*                    LISTEN      1309/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      25158/cupsd
tcp6       0      0 :::445                  :::*                    LISTEN      1447/smbd
administrator@RGQBE-LVDA-3:~$
```

この例では、Citrix Telemetry Service を停止するために次のコマンドを実行できます。

```
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、リソースをほとんど使用しません。Citrix Telemetry Service はデフォルトで停止しており、Delivery Controller からのログ収集要求がある場合にのみ起動します。ログ収集が完了すると、サービスは 15 分間新しい収集要求を待ち、要求がない場合は再び停止します。待機時間は次のコマンドで構成できます。最小値は 10 分です。10 分未満の値を設定した場合、最小値である 10 分が適用されます。待機時間を設定したら、サービスを停止して再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"TelemetryServiceIdleTimeoutInMinutes" -d <number> -t REG_DWORD
```

検証テスト

収集が開始される前に、選択された各マシンに対して検証テストが自動的に実行されます。これらのテストにより、要件が満たされていることが確認されます。マシンのテストが失敗した場合、Scout は推奨される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。

Azure を介した 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刻に自動的に更新するのに役立ちます。Citrix DaaS Standard for Azure (旧称 Citrix Virtual Apps and Desktops Standard for Azure) で Linux VDA を作成する場合に便利です。Azure の VM の管理者権限は必要ありません。詳細については、「Citrix DaaS Standard for Azure で Linux VDA を作成する」を参照してください。

構成

この機能を使用するには、次の手順を実行します。

ステップ 1: 更新情報と新しい VDA パッケージを Azure コンテナーにアップロードする

ステップ 1a: Azure ストレージアカウントの下にコンテナーを作成し、コンテナーのアクセスレベルを **Blob (匿名読み取りアクセスのみ)** に設定します。

> **注:**

> Azure コンテナーと BLOB は、お客様が排他的に保持および管理します。Citrix® は、それらに関するセキュリティ問題について責任を負いません。データセキュリティとコスト効率を確保するため、各**セルフアップデート**後にコンテナーのアクセスレベルを **Private (匿名アクセスなし)** に設定してください。

> ステップ 1b: VDA の更新情報を UpdateInfo.json という名前の JSON ファイルに組み込みます。ファイル形式の例については、次のブロックを参照してください。

```
{

 "Version": "21.04.200.4",

"Distributions":[

{
```

```
"TargetOS": "RHEL7_9",

"PackageName": "",

"PackageHash": ""

},

{

"TargetOS": "UBUNTU20_04",

"PackageName": "",

"PackageHash": ""

}

]

}
```

ここで、**「Version」は新しい VDA バージョンを示し、「Distributions」**は更新オブジェクトの配列です。各オブジェクトには次の 3 つの項目が含まれます。

- **「TargetOS」**：「RHEL7_9」（RHEL 7、CentOS 7、Amazon Linux 2 の場合）または「UBUNTU20_04」である必要があります。**ctxmonitord** は、その他のディストリビューションを認識しません。
- **「PackageName」**: 指定されたバージョンの VDA パッケージの完全な名前。
- **「PackageHash」**：`shasum -a 256 <pkgname>` コマンドを使用して計算する SHA-256 値。

ステップ 1c: JSON ファイルと新しいバージョンの Linux VDA パッケージを Azure コンテナーにアップロードします。

ステップ 2: マスターイメージまたは各 VDA でセルフアップデート機能を有効にする

デフォルトでは、**セルフアップデート**は無効になっています。Citrix DaaS Standard for Azure で Linux VDA を作成する場合、機能の有効化はマスターイメージで実行する必要があります。それ以外の場合は、各ターゲット VDA で直接機能を有効にします。

セルフアップデートを有効にするには、HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\SelfUpdate のレジストリキーを編集するために、次のようなコマンドを実行します。

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



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"ScheduledTime" -d "Immediately" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"Url" -d "<Your-Azure-Container-Url>" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"CaCertificate" -d "<Local-Certificate-Path-of-PortalAzureCom>" --force
```

以下の表にレジストリ設定について説明します。

レジストリ設定	説明
fEnabled	この設定は必須です。デフォルトでは、値は0であり、これは**セルフアップデート**が無効になっていることを意味します。1に設定すると、**セルフアップデート**を有効にできます。
Url	この設定は必須です。更新情報と新しいVDAパッケージを取得するためのAzureコンテナのURLを設定します。
ScheduledTime	この設定は必須です。**Immediately**または**NextStart**に設定できます。**Immediately**は、VDAパッケージのダウンロード後すぐに更新

レジストリ設定	説明
	を実行することを意味します。このオプションは、ダウンロード速度が速く、更新が緊急の場合に適しています。ただし、パッケージのダウンロード中にライブセッションがある場合、ユーザーエクスペリエンスを妨げる可能性があります。**NextStart**は、**ctxmonitord**の次回の起動時に更新を実行することを意味します。このオプションは、ダウンロード速度が速くなく、更新が緊急ではない場合に適しています。
CaCertificate	この設定はオプションです。AzureコンテナのURLを検証するためのPEM証明書のフルパスを設定します。Azure Blobの場合、portal.azure.comの証明書をブラウザから取得し、PEMに変換したものを使用できます。セキュリティのため、このレジストリ設定を追加することをお勧めしますが、Ubuntuでのみサポートされています。RHELでは、`curl`コマンド用のNSSライブラリの一部がリンクされていません。証明書の最小限の権限を設定するようにしてください。

ctxmonitordが再起動すると、まず**Url**を照会してUpdateInfo.jsonファイルを取得し、JSONファイルから更新バージョンを取得します。次に、**ctxmonitord**は更新バージョンを現在のバージョンと比較します。現在のバージョンが古い場合、サービスはAzureからVDAパッケージの新しいバージョンをダウンロードし、ローカルに保存します。その後、**ScheduledTime**の設定に従って更新を実行します。オンプレミス展開の場合、**ctxmonitord**を直接再起動して更新をトリガーできます。ただし、VMに対する管理者権限がないCitrix DaaS Standard for Azureでは、**ctxmonitord**はVDAマシンが再起動された後にのみ再起動できます。更新が失敗した場合、VDAは既存のバージョンにロールバックされます。

> **注:**
>
> - マスターイメージで構成したレジストリ設定は変更できません。
> - 環境内のすべてのVMが同時にパッケージをダウンロードすると、ローカルネットワークが混雑する可能性があります。

- 更新とロールバックの両方が失敗した場合、ユーザーデータは失われます。
- 更新が失敗してもロールバックが成功した場合、同じネットワーク上のユーザーは異なるバージョンのLinux VDAを使用している可能性があります。このケースは最適ではありません。
- 更新には通常、数分かかります。Citrix Studio にはステータスインジケーターはありません。

Linux VM と Linux セッションのメトリック

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

メトリック	必要な最小 VDA バージョン	説明	備考
ログオン時間	2109	Citrix Workspace アプリからユーザーが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ます。セッションのメトリックを表示するには、Citrix DaaS (旧 Citrix Virtual Apps and Desktops サービス) の Monitor タブに移動します。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。Monitor タブで、**[平均ログオン時間]** セクションの **[履歴トレンドの表示]**	Monitor でのみ利用可能です。

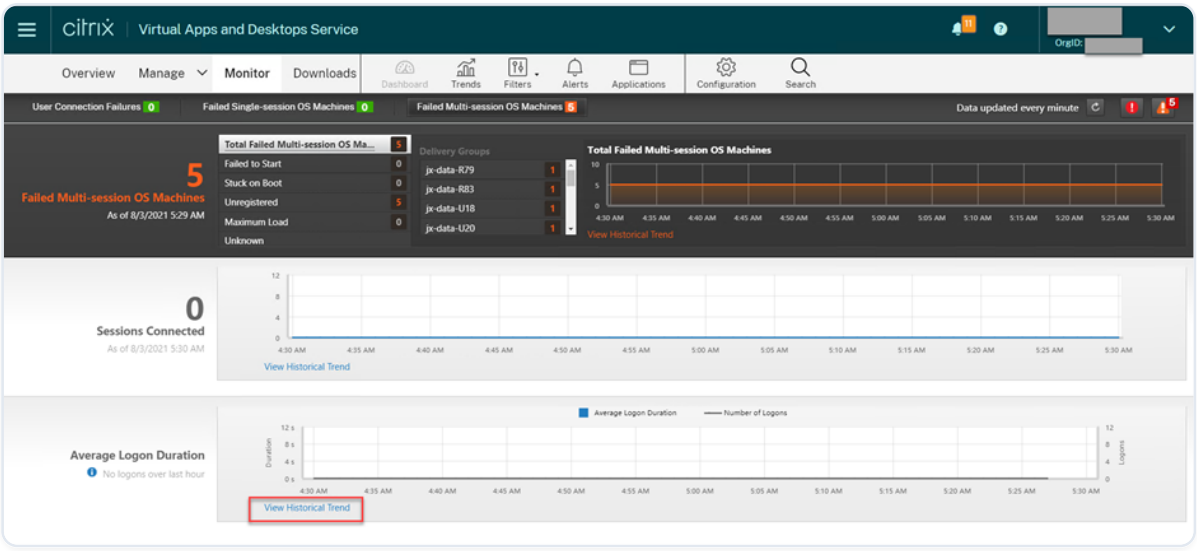
メトリック	必要な最小 VDA バージョン	説明	備考
		をクリックします。**[ログオンパフォーマンス]** ページで、フィルター条件を設定し、**[適用]** をクリックしてメトリックを視覚化します。	
セッションの自動再接続数	2109	セッション内の自動再接続の数を表示するには、**[トレンド]** ビューにアクセスします。条件を設定し、**[適用]** をクリックして検索結果を絞り込みます。**[セッションの自動再接続数]** 列には、セッション内の自動再接続の数が表示されます。**自動再接続**は、**[セッションの信頼性]** または **[クライアントの自動再接続]** ポリシーが有効な場合に有効になります。セッションの再接続について詳しくは、「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。	Citrix Director と Monitor の両方で利用可能です。

メトリック	必要な最小 VDA バージョン	説明	備考
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 と Monitor の両方で利用可能です。
Linux VM のメトリック	2103	Linux VM で利用可能なメトリックは次のとおりです: CPU コア数、メモリサイズ、ハードディスク容量、現在の CPU およびメモリ使用率、履歴 CPU およびメモリ使用率	Citrix Director と Monitor の両方で利用可能です。
プロトコル	1909	Linuxセッションのトランスポートプロトコルは、**セッションの詳細**パネルにUDPまたはTCPとして表示されます。	Citrix DirectorとMonitorの両方で利用可能です。
ICA® RTT	1903	ICAラウンドトリップタイム（RTT）は、キーを押してからエンドポイントに応答が表示されるまでの経過時間です。ICA 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ラウンドトリップ間隔計算**ポリシーを作成します。	Citrix DirectorとMonitorの両方で利用可能です。

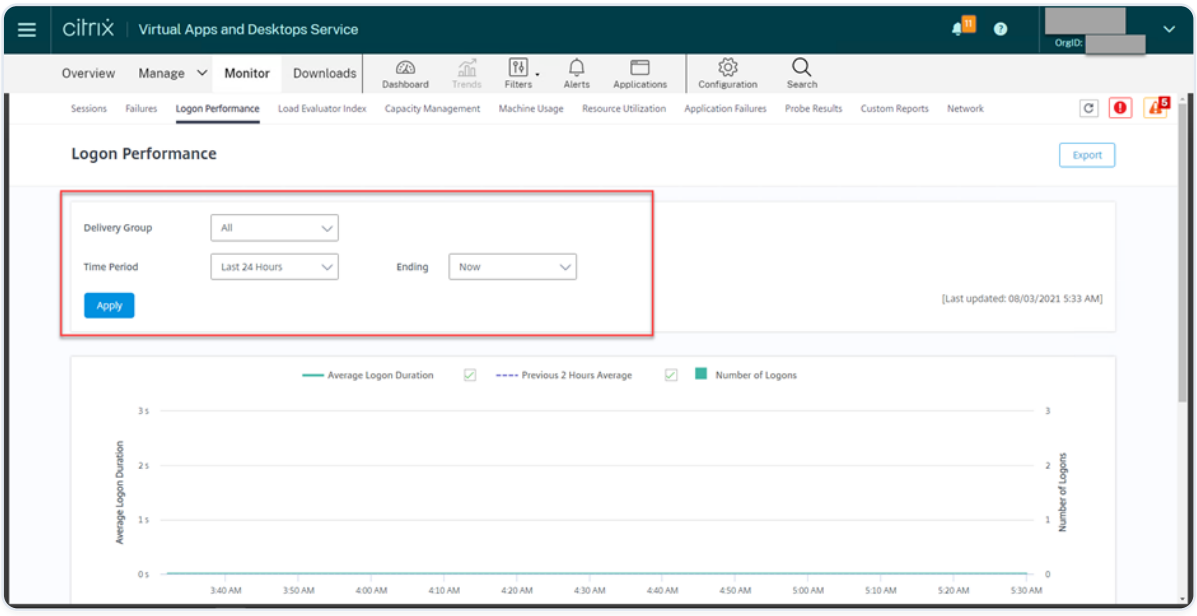
Citrix DirectorおよびMonitorでさまざまなメトリックにアクセスする方法の例

- **ログオン期間**

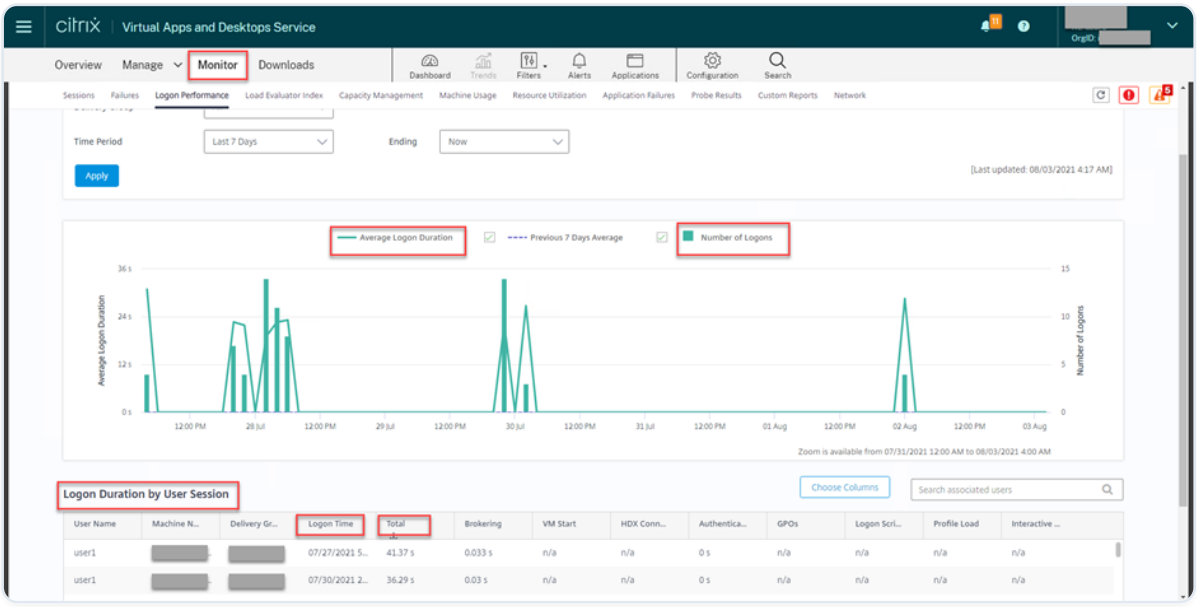
a. Citrix DaaSのMonitorタブで、**平均ログオン期間**セクションの**履歴トレンドの表示**をクリックします。



b. **ログオンパフォーマンス**ページで、フィルター条件を設定します。

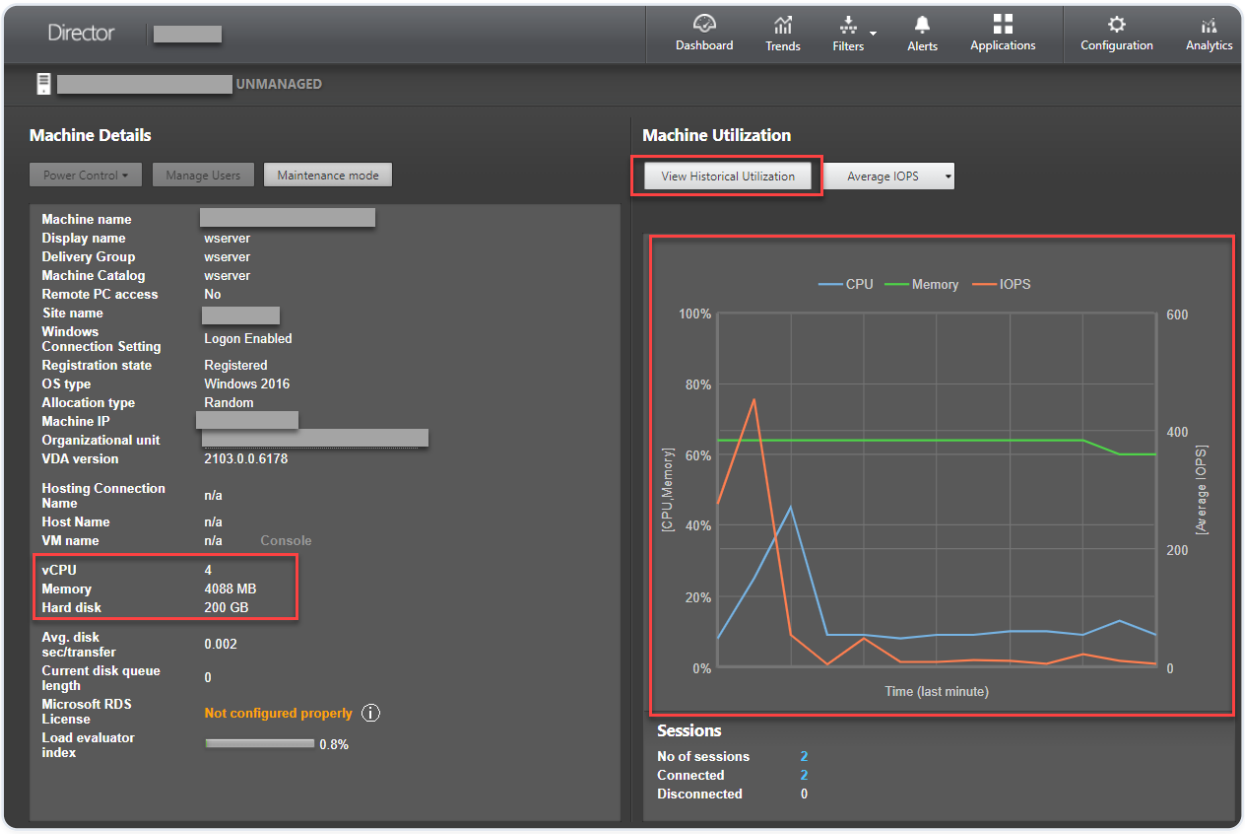


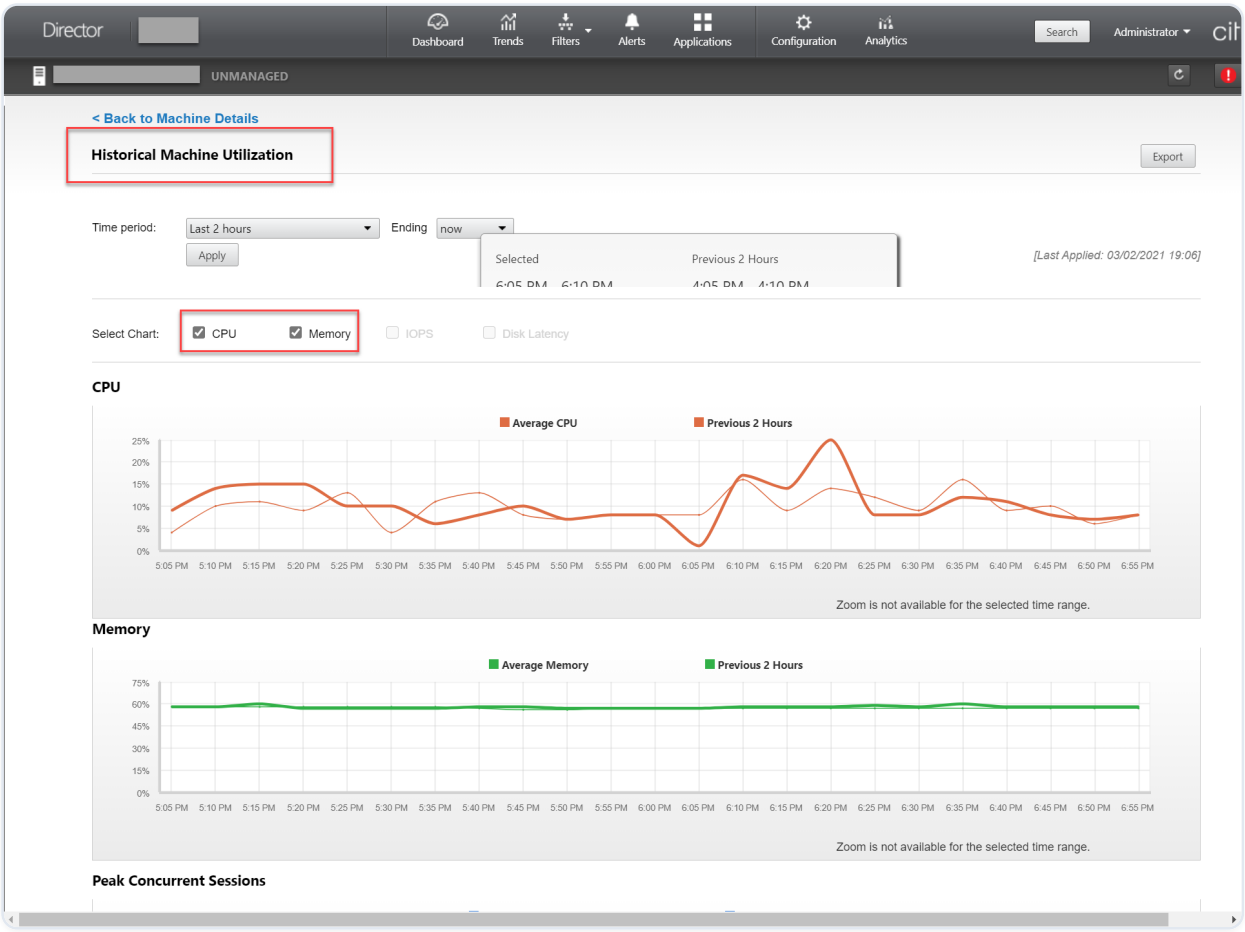
c. ログオン期間メトリックを視覚化するには、**適用**をクリックします。



- **Linux VMの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とメモリ使用率**

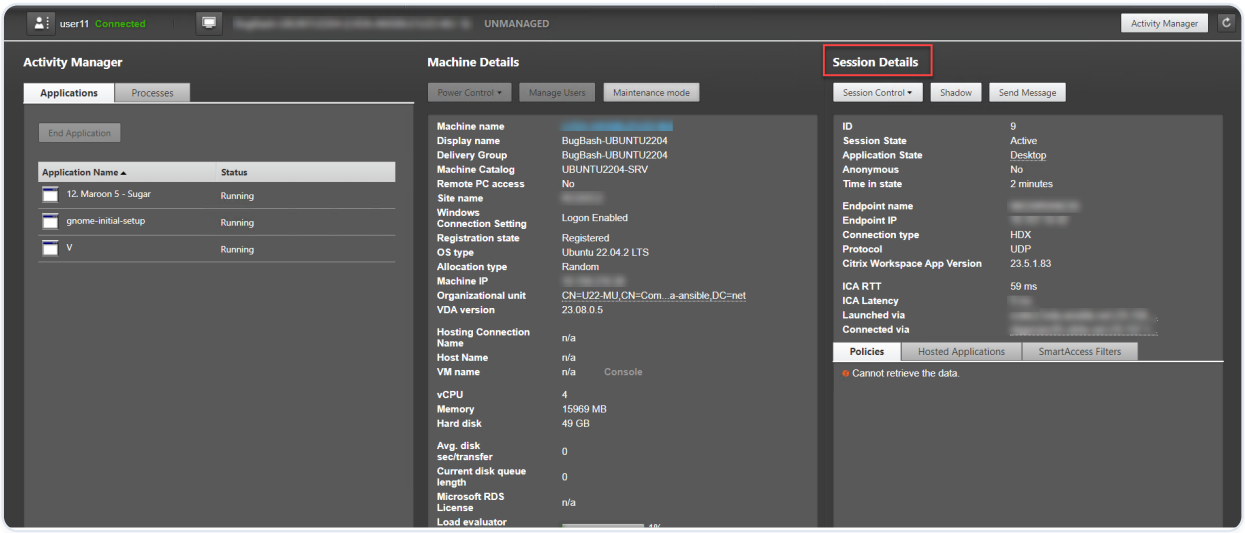
  Linux VMのこれらのメトリックにアクセスするには、Citrix DirectorまたはMonitorでVMを見つけ、**マシンの詳細**パネルを確認します。例：





- **ICA RTT とプロトコル**

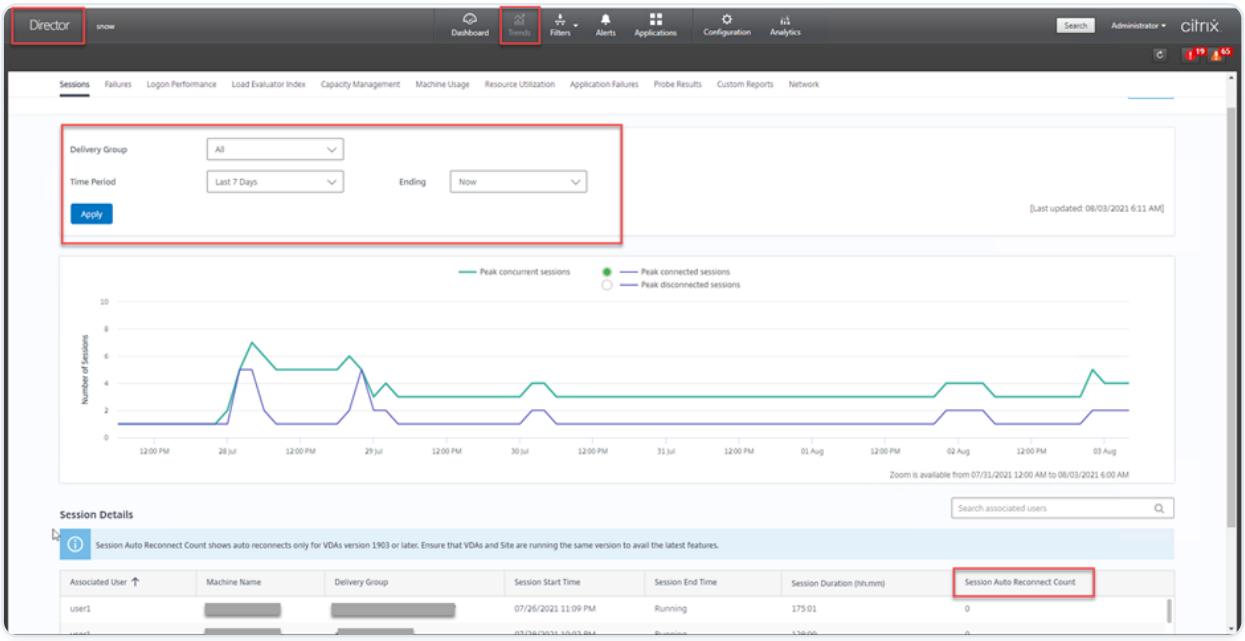
  Linuxセッションのメトリックを表示するには、**フィルター > セッション > すべてのセッション**を選択して**すべてのセッション**ページを開くか、**セッションの詳細**パネルにアクセスします。**セッションの詳細**パネルにアクセスするには、**すべてのセッション**ページを開き、ターゲットセッションをクリックしてその**アクティビティマネージャー**ビューにアクセスします。例:



- **セッションの自動再接続数**

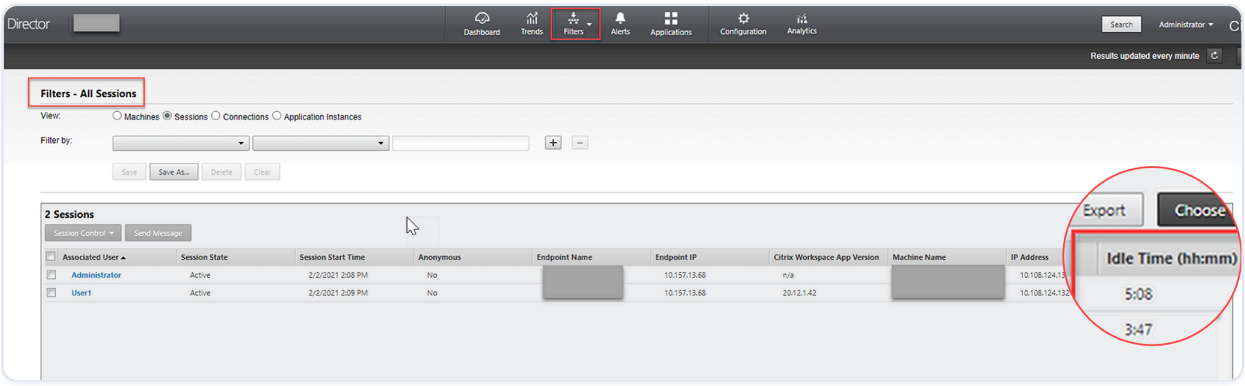
セッション内の自動再接続数を表示するには、**トレンド**ビューにアクセスします。条件を設定し、**適用**をクリックして検索結果を絞り込みます。

セッションの自動再接続数列には、セッション内の自動再接続数が表示されます。例:



- **アイドル時間**

  例:



ログ収集

概要

ログ収集は、デフォルトでLinux VDAに対して有効になっています。

構成

`ctxlogd`デーモンと`setlog`ユーティリティは、Linux VDAリリースパッケージに含まれています。デフォルトでは、Linux VDAのインストールと構成後に`ctxlogd`デーモンが起動します。

ctxlogdデーモン

トレースされる他のすべてのサービスは、`ctxlogd`デーモンに依存します。Linux VDAのトレースを継続したくない場合は、`ctxlogd`デーモンを停止できます。

setlogユーティリティ

ログ収集は、**/opt/Citrix/VDA/bin/**パスにある`setlog`ユーティリティを使用して構成されます。rootユーザーのみが実行する権限を持っています。GUIを使用するか、コマンドを実行して構成を表示および変更できます。`setlog`ユーティリティ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

デフォルトでは、**Log Output Path**は**/var/log/xdl/hdx.log**に、**Max Log Size**は200 MBに設定されており、**Log Output Path**の下に最大2つの古いログファイルを保存できます。

現在の`setlog`値を表示します。

```
setlog values


log_path (Log Output Path) = /var/log/xdl/hdx.log


log_size (Max Log Size (MiB)) = 200


log_count (Max Old Log Files) = 2
```

単一の`setlog`値を表示または設定します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

デフォルトでは、ログレベルは**warning**（大文字と小文字を区別しない）に設定されています。

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

```
setlog levels
```

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

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

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

`<class>` 変数は、Linux VDA のコンポーネントの 1 つを指定します。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
```

フラグ

デフォルトでは、フラグは次のように設定されています。

```
setlog flags


DATE = true


TIME = true


NAME = true


PID = true


TID = false


SID = true


UID = false


GID = false


CLASS = false


LEVEL = false
```

```
FUNC = true



FILE = false
```

現在のフラグを表示する:

```
setlog flags
```

単一のログフラグを表示または設定する:

```
setlog flag <flag> [<state>]
```

デフォルトに戻す

すべてのレベル、フラグ、および値をデフォルト設定に戻します。

```
setlog default
```

> **重要:**
>
> `ctxlogd`サービスは、ルートユーザーのみが作成できる**/var/xdl/.ctxlog**ファイルを使用して構成されます。他のユーザーはこのファイルへの書き込み権限を持っていません。ルートユーザーが他のユーザーに書き込み権限を与えないことをお勧めします。これに従わない場合、任意のまたは悪意のある構成が`ctxlogd`し、サーバーのパフォーマンス、ひいてはユーザーエクスペリエンスに影響を与える可能性があります。

トラブルシューティング

/var/xdl/.ctxlogファイルが見つからない（たとえば、誤って削除された）場合、`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、ルートユーザーとして`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

セッションシャドウイングを使用すると、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。この機能はnoVNCを使用してICAセッションに接続します。

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

インストールと構成

依存関係

セッションシャドウイングには、`python-websockify`と`x11vnc`という2つの新しい依存関係が必要です。Linux VDAのインストール後に、`python-websockify`と`x11vnc`を手動でインストールしてください。

RHEL 7.xおよびAmazon Linux2の場合：

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

```
sudo pip3 install websockify

sudo yum install x11vnc
```

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

- EPEL

  EPELリポジトリは`x11vnc`に必要です。EPELリポジトリを有効にするには、次のコマンドを実行します。

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

- オプションのRPM

  `x11vnc`の一部の依存パッケージをインストールするために、オプションの`RPMs`リポジトリを有効にするには、次のコマンドを実行します。

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

RHEL 9.2/9.0/8.x および Rocky Linux 9.2/9.0/8.x をご利用の場合：

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

```
sudo pip3 install websockify

sudo yum install x11vnc
```

EPELおよびCodeReady Linux Builderリポジトリを有効にして、`x11vnc`を解決します。

```
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` をインストールします (`x11vnc` バージョン 0.9.13 以降):

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

SUSEの場合:

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

```
sudo pip3 install websockify

sudo zypper install x11vnc
```

Debianの場合:

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

```
sudo pip3 install 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の範囲で利用可能なポートを自動的に選択します。したがって、同時にシャドウできるICAセッションの数は99に制限されます。特にマルチセッションシャドウイングの場合、要件を満たすのに十分なポートが利用可能であることを確認してください。

レジストリ

次の表に関連するレジストリを示します:

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションシャドウイング機能を有効または無効にします	1 (有効)
ShadowingUseSSL	Linux VDAとCitrix Director間の接続を暗号化するかどうかを決定します	0 (無効)

レジストリ値を変更するには、Linux VDAで`ctxreg`コマンドを実行します。たとえば、セッションシャドウイングを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d
"0x00000000"
```

SSL

Linux VDA と Citrix Director 間の noVNC 接続は WebSocket プロトコルを使用します。セッションシャドウイングの場合、`ws://` と `wss://` のどちらが選択されるかは、前述の「ShadowingUseSSL」レジストリによって異なります。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上の責任も負いません。

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。

サーバー証明書とクライアント証明書は、次の CA からインストールできます。

- オペレーティングシステムにバンドルされている CA
- エンタープライズ CA（組織がアクセスを許可している CA）
- オペレーティングシステムにバンドルされていない CA

証明書を取得するためにどの方法が必要かについては、組織のセキュリティチームに相談してください。

重要:

- サーバー証明書の共通名 (Common Name) 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 および 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、受け入れなくなりました。`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**機関**タブでルートSSL証明書をインポートする必要があります。

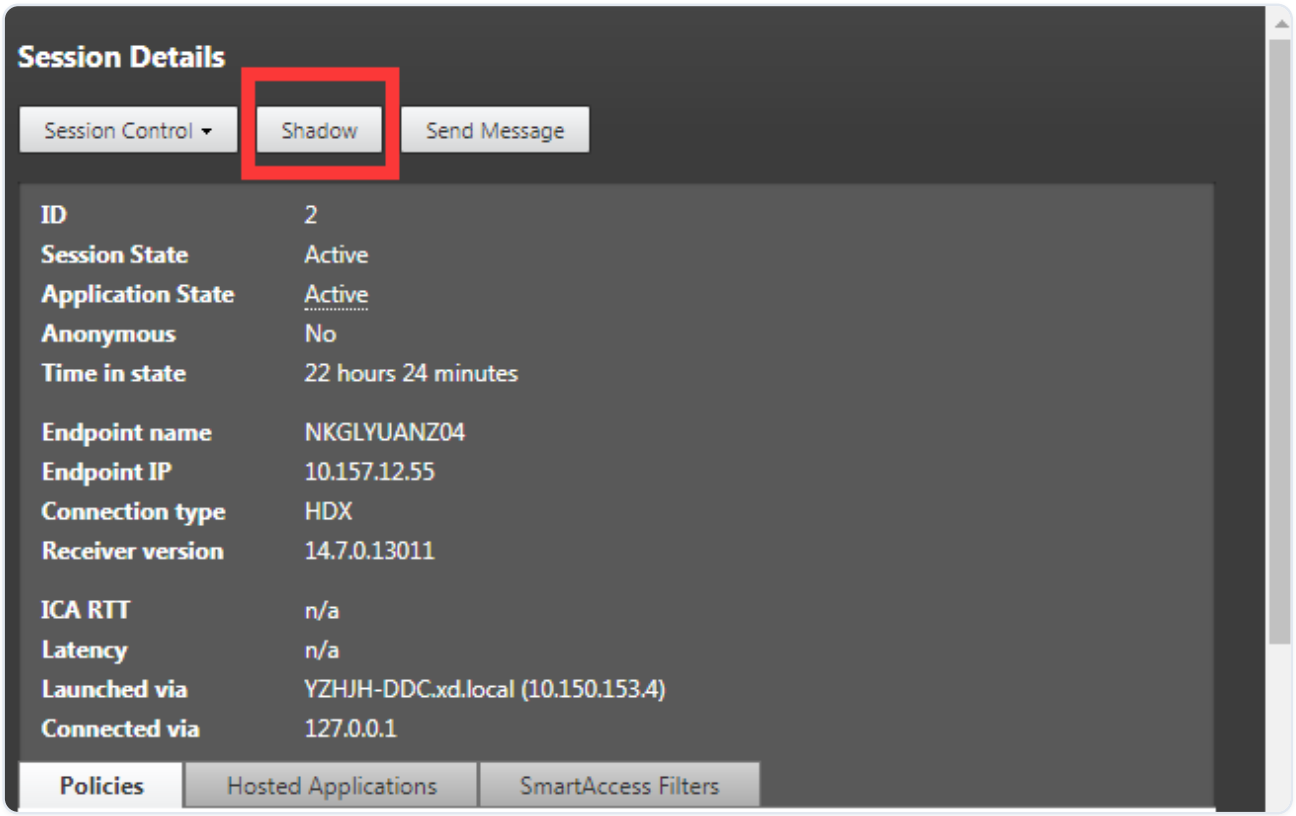
各LINUX VDAサーバーにサーバー証明書とそのキーをインストールする

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はパス **/etc/xdl/shadowingssl** の下に置き、制限された権限で適切に保護してくださ

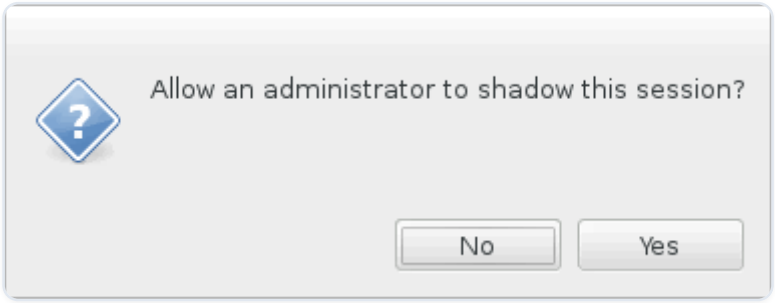
い。名前またはパスが正しくないと、Linux VDAは特定の証明書またはキーファイルを見つけることができず、`Citrix Director`との接続失敗の原因となります。

使用方法

`Citrix Director`から、ターゲットセッションを見つけ、**セッションの詳細**ビューで**シャドウ**をクリックして、Linux VDAにシャドウイング要求を送信します。



接続が初期化されると、ICAセッションクライアント（`Citrix Director`クライアントではない）に確認が表示され、ユーザーにセッションをシャドウする許可を求めます。



ユーザーが**はい**をクリックすると、`Citrix Director`側にウィンドウが表示され、ICAセッションがシャドウされていることを示します。

使用方法の詳細については、Citrix Directorドキュメントを参照してください。

制限事項

- セッションシャドウイングはイントラネット内でのみ使用するように設計されています。Citrix Gateway経由で接続する場合でも、外部ネットワークでは機能しません。Citrixは、外部ネットワークでのLinux VDAセッションシャドウイングについて一切の責任を負いません。
- セッションシャドウイングが有効になっている場合、ドメイン管理者はICAセッションを表示することしかできず、書き込みや制御の権限はありません。
- 管理者が `Citrix Director` から**シャドウ**をクリックすると、セッションをシャドウする許可をユーザーに求める確認が表示されます。セッションは、セッションユーザーが許可を与えた場合にのみシャドウできます。
- 前述の確認には20秒のタイムアウト制限があります。時間が経過すると、シャドウイング要求は失敗します。
- 1つのセッションは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した場合、ユーザー権限取得の確認がユーザーデバイスに再表示されます。ユーザーが同意すると、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が構築されます。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ます。
- `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実行します。

Citrix Directorクライアントで

ブラウザの開発者ツールを使用して、**Console**タブで出力ログを確認します。または、**Network**タブでShadowLinuxSession APIの応答を確認します。ユーザー権限取得の確認が表示されるものの、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 で

1. `/var/log/xdl/vda.log` ファイルで手がかりを確認します。

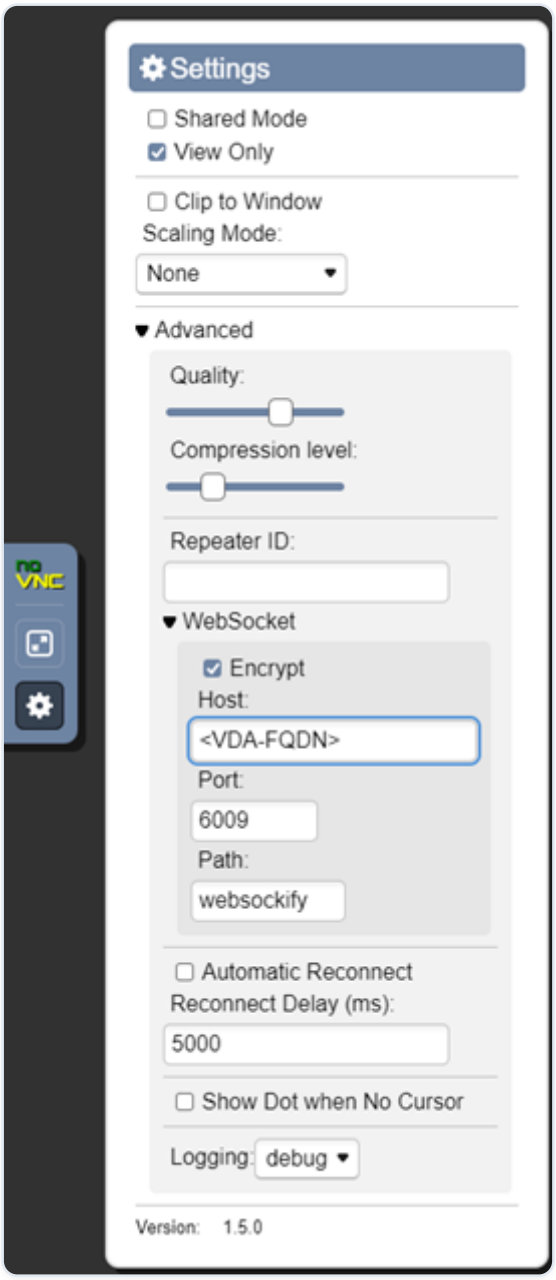
2. `/var/xdl/sessionshadowing.sh`ファイルを編集し、「logFile」変数を変更して、Directorからのセッションシャドウ中に手がかりを追跡できるログファイルを指定します。
3. また、noVNC接続で証明書が正しく機能するかどうかを手動で確認することもできます。

   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。

   ```
   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. `/var/xdl/sessionshadowing.sh`で必要となるため、「netstat」がインストールされていることを確認します。

v. `openssl x509 -in shadowingcert.pem -text -noout`を実行して、証明書が正しく構成されていることを確認します。特にCNフィールドとSANフィールドに注意してください。

vi. 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
```

モニターサービスデーモン

モニターサービスデーモン (**ctxmonitord**) は、定期的なスキャンを実行して主要なサービスを監視します。例外を検出すると、デーモンはサービスプロセスを再起動または停止し、リソースを解放するためにプロセスの残骸をクリーンアップします。検出された例外は、**/var/log/xdl/ms.log** ファイルに記録されます。

構成

モニターサービスデーモンは、VDAを起動すると自動的に開始されます。

この機能は、**/opt/Citrix/VDA/sbin** の下にある **scanningpolicy.conf**、**rulesets.conf**、および **whitelist.conf** ファイルを管理者権限で構成できます。

scanningpolicy.conf、**rulesets.conf**、および **whitelist.conf** ファイルで行った変更を有効にするには、モニターサービスデーモンを再起動するために、次のコマンドを実行します。

```
systemctl restart ctxmonitord
```

- **scanningpolicy.conf**

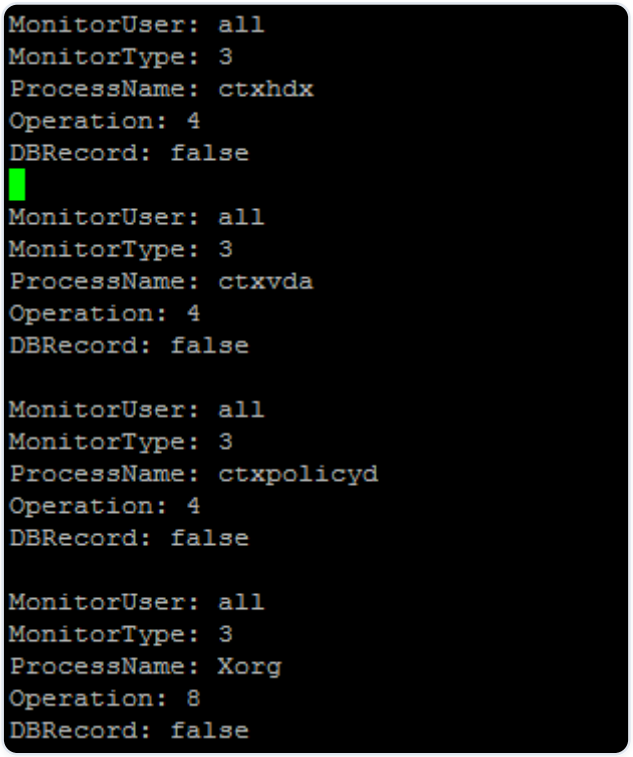
  この構成ファイルは、モニターサービスデーモンを有効または無効にします。サービス検出間隔を設定し、検出された例外を修復するかどうかを指定します。

  - MonitorEnable: true/false (デフォルトはtrue)
  - DetectTime: 20 (単位: 秒、デフォルト値: 20、最小値: 5)
  - AutoRepair: true/false (デフォルトはtrue)
  - MultBalance: 無効

- ReportAlarm: 無効

- **rulesets.conf**

  この構成ファイルは、監視するターゲットサービスを指定します。次のスクリーンショットに示すように、デフォルトで4つのサービスが監視されます。



  監視する各サービスを設定するには、次のフィールドを設定します。

  - **MonitorUser**: すべて
  - **MonitorType**: 3
  - **ProcessName**: <> (プロセス名は空白にできず、完全に一致する必要があります。)
  - **Operation**: 1/2/4/8 (1 = 例外が検出されたときにサービスを停止します。2 = 例外が検出されたときにサービスを強制終了します。4 = サービスを再起動します。8 = Xorg プロセス残存をクリアします。)
  - **DBRecord**: 無効

- **whitelist.conf**

  rulesets.conf ファイルで指定されたターゲットサービスは、**whitelist.conf** ファイルでも構成する必要があります。ホワイトリスト構成は、セキュリティのための二次フィルターです。

  ホワイトリストを構成するには、**whitelist.conf** ファイルにプロセス名のみを含めます（完全に一致する必要があります）。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。

```
ctxcdmd
ctxcdmmount
ctxcdmstat
ctxceip
ctxclipboard
ctxconnect
ctxcredentialctl
ctxctl
ctxcupsd
ctxdisconnect
ctxeuem
ctxfiletransfer
ctxgfx
ctxhdx
ctxism
ctxlogd
ctxlogin
ctxmonitorservice
ctxmrvc
ctxpolicyd
ctxscardsd
ctxvhcid
ctxvda
Xorg
```

> **注:**
>
> **ctxvda**、**ctxhdx**、および **ctxpolicyd** 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。

ツールとユーティリティ

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**) を提供しています。VDAでホストされているすべてのセッションまたは特定のセッションの以下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**

xdlcollect Bashスクリプト

ログ収集に使用される **xdlcollect** Bashスクリプトは、Linux VDAソフトウェアに統合されており、**/opt/Citrix/VDA/bin** にあります。Linux VDAをインストールした後、bash **/opt/Citrix/VDA/bin/xdlcollect.sh** コマンドを実行してログを収集できます。ログ収集が完了すると、スクリプトと同じフォルダーに圧縮ログファイルが生成されます。**xdlcollect** Bashスクリプトは、圧縮ログファイルをCitrix Insight Services (CIS) にアップロードするかどうかを尋ねることができます。同意した場合、アップロード完了後に **xdlcollect** はupload_IDを返します。アップロードによって、ローカルマシンから圧縮ログファイルが削除されることはありません。他のユーザーはupload_IDを使用してCIS内のログファイルにアクセスできます。

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のチェックプロセスを自動化します。

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。
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チェックするには	**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、たとえば、時間、Kerberos、およびデータベースのテスト	**sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos,database**	該当なし

タスク	XDPING コマンド	備考
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>**	該当なし

個別のテスト

Linuxの**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。これらは大まかに次のように分類されます。

- Linux VDAのシステム要件が満たされているかを確認します。
- Linuxディストリビューションを含むマシン情報を特定して表示します。
- Linuxカーネルの互換性を確認します。
- Linux VDAの動作に影響を与える可能性のある既知のLinuxディストリビューションの問題を確認します。
- Security-Enhanced Linux (SELinux) のモードと互換性を確認します。
- ネットワークインターフェイスを特定し、ネットワーク設定を確認します。
- ストレージのパーティション分割と利用可能なディスク容量を確認します。
- マシンのホスト名とドメイン名の構成を確認します。
- DNS構成を確認し、ルックアップテストを実行します。
- 基盤となるハイパーバイザーを特定し、仮想マシンの構成を確認します。サポート対象:
  - シトリックス ハイパーバイザー™
  - マイクロソフト Hyper-V
  - 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を確認します。
- PostgreSQLサービスが適切に構成され、動作しているかを確認します。
- SQLiteが適切に構成され、動作しているかを確認します。
- ファイアウォールが有効になっており、必要なポートが開いているかを確認します。
- Kerberos構成を確認し、認証テストを実行します。
- グループポリシーサービスエンジン用のLDAP検索環境を確認します。
- Active Directory統合が適切に設定され、現在のマシンがドメインに参加しているかを確認します。サポート対象:

    - サンバ Winbind
    - デル クエスト 認証サービス
    - セントリファイ ダイレクトコントロール
    - SSSD

- 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™ の構成の整合性を確認します。
- 構成済みの各 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。
- アクティブまたは切断された各 HDX セッションの状態を確認します。
- ログファイルをスキャンして、Linux VDA 関連のエラーと警告を確認します。
- Xorg のバージョンが適切であるか確認します。
- 必要な依存関係がインストールされているか確認します。

出力例

以下は、Kerberos テストを実行した際の出力例です。

Kerberos テストの出力例の最初の部分(/en-us/linux-virtual-delivery-agent/2308/media/kerberos-test-sample-output-1.png)

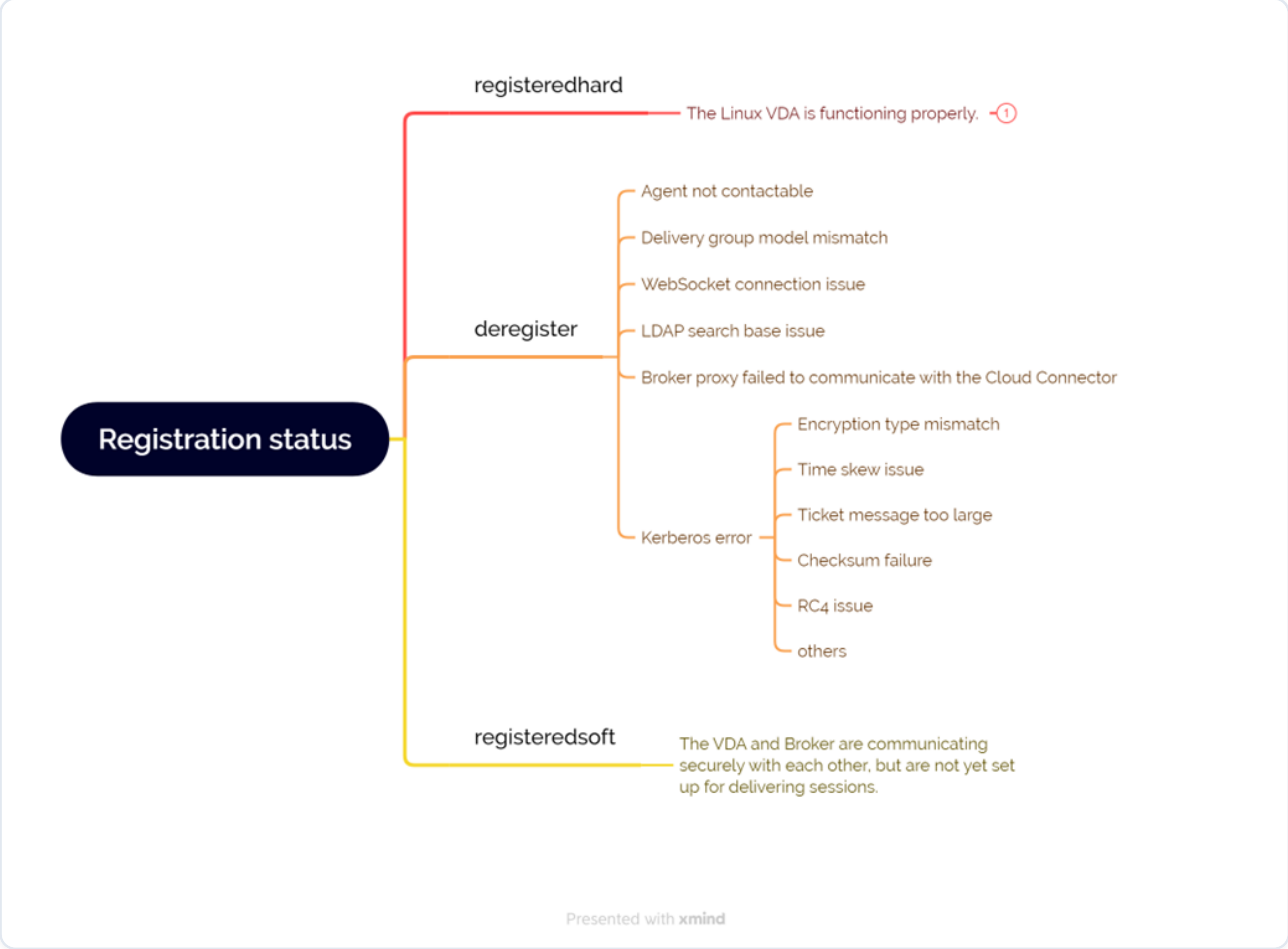
```
  Verify Kerberos configuration file found                    [Pass]
Keytab file:  /etc/krb5.keytab [Exists]
Default realm: XD2.LOCAL
Default realm KDCs: [NONE SPECIFIED]
Default realm domains: [NONE SPECIFIED]
DNS lookup realm: [Enabled]
DNS lookup KDC: [Enabled]
Weak crypto:  [Disabled]
Clock skew 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 **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つのデータの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	**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**形式で現在の日付と時刻で命名されます（例：**2023-02-27-16_31_27**）。デ

タスク	XDPING コマンド	備考
		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`
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**	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=<1つ目のバックアップデータディレクトリ>:<もう1つのバックアップデータディレクトリ>**	該当なし

その他

このセクションには、次のトピックが含まれています。

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

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 仮想アプリと仮想デスクトップに直接アクセスできます。Citrix Workspace app for HTML5 の詳細については、Citrix ドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトで無効になっています。有効にするには、次の手順を実行します。

1. Citrix StoreFront™ で、Citrix Workspace 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。

      その他の WebSocket ポリシーも設定できます。WebSocket ポリシーの完全なリストについては、WebSockets policy settings 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 暗号化を有効にする方法については、Secure user sessions using TLS 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` または `/opt/Citrix/VDA/sbin/enable_ldaps.sh` コマンドを実行すると、**Python3** 仮想環境を作成できます。ただし、コマンドで**Python3** 仮想環境を作成できない場合は、ネットワーク接続がなくても手動で作成できます。この記事では、ネットワーク接続なしで**Python3** 仮想環境を作成するための前提条件と手順について詳しく説明します。

前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3` ディレクトリにアクセスするには、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ること。wheelファイルは `<https://pypi.org/>` からダウンロードできます。

Python3 仮想環境を作成する

次の手順を実行して、**Python3** 仮想環境を作成します。

1. **Python3** の依存関係をインストールします。

   Amazon Linux 2 の場合:

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

   RHEL および Rocky Linux の場合:

   ```
   yum -y install 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 7 の場合は、`subscription-manager repos --enable rhel-7-server-optional-rpms` コマンドを実行します。RHEL 8 の場合は、`subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms` コマンドを実行します。

   Debian、Ubuntu の場合:

   ```
   apt-get -y install python3-dev python3-pip python3-venv libkrb5-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:**
>
> 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、/etc/pip.confファイルに次の信頼できるホストを追加することを検討してください。
>
> \> `[global]`
>
> `trusted-host =`
>
> \> `pypi.org`
>
> `files.pythonhosted.org`

> **アマゾンリナックス2、デビアン、RHEL、ロッキーリナックス、ウブントゥの場合:**

```
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. LDAPS の依存関係をインストールします。

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip==21.3.1



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1 gssapi==1.7.3
ldap3==2.9.1 pyasn1==0.4.8 pycparser==2.21 six==1.16.0
```

4. **XDPing** の依存関係をインストールします。

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip==21.3.1



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



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

NIS と Active Directory を統合する

この記事では、SSSD を使用して Linux VDA 上で NIS と Windows Active Directory (AD) を統合する方法について説明します。Linux VDA は Citrix Virtual Apps and Desktops のコンポーネントと見なされます。その結果、Windows AD 環境に緊密に適合します。

UID および GID プロバイダーとして AD の代わりに NIS を使用するには、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/CentOS ディストリビューションの場合は、Amazon Linux 2、CentOS、RHEL、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、リモートリソースの識別と認証へ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 を統合する

NIS を AD と統合するには、次の手順を実行します。

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` を使用して NIS を構成します。

```
sudo authconfig --enablenis --nisdomain=nis.domain --
nisserver=server.nis.domain --enablemkhomedir --update
```

nis.domain は NIS サーバーのドメイン名を表します。**server.nis.domain** は NIS サーバーのホスト名であり、NIS サーバーの IP アドレスでもかまいません。

NISサービスを設定します。

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS構成が正しいことを確認します。

```
ypwhich
```

アカウント情報がNISサーバーから利用可能であることを検証します。

```
getent passwd nisaccount
```

> **注:**
>
> **nisaccount** は、NISサーバー上の実際のNISアカウントを表します。UID、GID、ホームディレクトリ、およびログインシェルが正しく構成されていることを確認してください。

ステップ2：SAMBAを使用してドメインに参加し、ホストキータブを作成する

SSSDは、ドメインへの参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。これらの機能を実現するには、いくつかの方法があります。

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

KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、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

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 のファイルの所有権とアクセス許可を設定します。

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

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

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はないことに注意してください。**ctxsetup.sh** スクリプトの追加ソリューションには、デフォルト値を使用してください。

ステップ 5: KERBEROS構成の確認

Linux VDAでKerberosが正しく構成されていることを確認するには、システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用して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いるか、およびNSSが機能するかどうかを確認します。

```
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、ドメインユーザーアカウントを使用してLinux VDAにログオンします。このドメインユーザーアカウントは以前に使用されたことがないものとします。

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



id -u
```

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

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

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

```
klist
```

IPv6

Linux VDAは、Citrix Virtual Apps and Desktopsに合わせて**IPv6**をサポートします。この機能を使用する際は、次の点を考慮してください。

- デュアルスタック環境では、**IPv6**が明示的に有効になっていない限り、**IPv4**が使用されます。
- **IPv4**環境で**IPv6**が有効になっている場合、Linux VDAは機能しません。

> **重要:**
>
> - Linux VDAだけでなく、ネットワーク環境全体が**IPv6**である必要があります。
> - **Centrify**は純粋な**IPv6**をサポートしていません。

> Linux VDAをインストールする際に、**IPv6**に関する特別なセットアップタスクは必要ありません。

Linux VDAのIPv6を構成する

Linux VDAの構成を変更する前に、Linux仮想マシンが以前に**IPv6**ネットワークで動作していたことを確認してください。**IPv6**構成に関連するレジストリキーは2つあります。

```
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration"

"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"ControllerRegistrationIPv6Netmask"
```

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 support」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) のセキュアなバージョンです。

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)/LDAP サーバーで LDAPS を有効にする
- クライアントで使用するためにルート CA をエクスポートする
- Linux VDA で LDAPS を有効/無効にする
- サードパーティプラットフォーム向けに LDAPS を構成する
- SSSD を構成する
- Winbind を構成する
- Centrify を構成する
- Quest を構成する

> **注:**

LDAP サーバーの監視サイクルを設定するには、次のコマンドを実行します。デフォルト値は 15 分です。少なくとも 10 分に設定してください。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --
force
```

AD/LDAPサーバーでLDAPSを有効にする

Microsoft 認証局 (CA) または Microsoft 以外の CA のいずれかから、適切にフォーマットされた証明書をインストールすることで、LDAP over SSL (LDAPS) を有効にできます。

ヒント:

ドメイン コントローラーにエンタープライズ ルート CA をインストールすると、LDAPS は自動的に有効になります。

証明書のインストール方法と LDAPS 接続の検証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。

多層認証局階層がある場合、ドメイン コントローラーでの LDAPS 認証のために適切な証明書が自動的に提供されません。

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。

クライアントで使用できるよう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、(管理者としてログオンしている間に) 次のスクリプトを実行します。

このコマンドの構文には、以下が含まれます。

- ルート CA 証明書が提供されている場合、LDAP over SSL/TLS を有効にする:

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

- チャネル バインディングを使用して LDAP over SSL/TLS を有効にする:

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

> **注:**
>
> チャネル バインディング用のルート CA 証明書は PEM 形式である必要があります。LDAPS を有効にしても **Python3** 仮想環境が正常に作成されない場合は、**Python3** 仮想環境を作成するの手順に従って手動で作成してください。
>
> pipツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼済みホストを/etc/pip.confファイルに追加することを検討してください:
>
> \> `[global]`
>
> `trusted-host =`
>
> \> `pypi.org`
>
> `files.pythonhosted.org`

- SSL/TLSなしでLDAPにフォールバックする

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
```

LDAPS専用のJavaキーストアは**/etc/xdl/.keystore**にあります。影響を受けるレジストリキーは次のとおりです:

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers

HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy

HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS

HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore

HKLM\Software\Citrix\VirtualDesktopAgent\EnableChannelBinding
```

サードパーティプラットフォーム向けLDAPSの構成

Linux VDAコンポーネントの他に、VDAに準拠するいくつかのサードパーティ製ソフトウェアコンポーネント（SSSD、Winbind、Centrify、Questなど）もセキュアなLDAPを必要とする場合があります。以下のセクションでは、LDAPS、STARTTLS、またはSASL署名およびシーリングを使用してセキュアなLDAPを構成する方法について説明します。

> **ヒント:**
>
> これらのソフトウェアコンポーネントのすべてが、セキュアなLDAPを確保するためにSSLポート636を使用することを好むわけではありません。また、ほとんどの場合、LDAPS（ポート636上のSSL経由LDAP）はポート389上のSTARTTLSと共存できません。

SSSD

オプションに従って、ポート636またはポート389でSSSDセキュアLDAPトラフィックを構成します。詳細については、SSSD LDAP Linux manページを参照してください。

ウィンバインド

Winbind LDAPクエリはADSメソッドを使用します。Winbindはポート389でのStartTLSメソッドのみをサポートします。影響を受ける構成ファイルは、**/etc/samba/smb.conf** および **/etc/openldap/ldap.conf** (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`

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

`ldap ssl = off`

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

セントリファイ

CentrifyはポートでのLDAPSをサポートしていません。ただし、ポート389では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、別の方法を使用してポート 389 で安全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ます。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ます。まずルート CA 証明書を確認し、それが AD/LDAP サーバーと一致することを確認します。

- **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 経由で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** を実行します。

- **Wireshark またはその他のネットワーク監視ツールから LDAPS トラフィックがない**

  LDAP/LDAPS トラフィックは、Linux VDA の登録とグループポリシーの評価が行われるときに発生します。

- **AD サーバーで ldp connect を実行して LDAPS の可用性を確認できませんでした**

  IP アドレスの代わりに AD FQDN を使用します。

- **/opt/Citrix/VDA/sbin/enable_ldaps.sh スクリプトの実行によるルート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証明書を表示できませんでした**

  `/opt/Citrix/VDA/sbin/enable_ldaps.sh`を実行してLDAPSを有効にすると、証明書は/etc/xdl/.keystoreにインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

エックスオーソリティ

Linux VDA は、インタラクティブなリモート処理のために X11 ディスプレイ機能 (`xterm` および `gvim` を含む) を使用する環境をサポートしています。この機能は、XClient と XServer 間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信のアクセス許可を保護するには、2つの方法があります。

- **Xhost**。デフォルトでは、Xhost は localhost の XClient のみが XServer と通信することを許可します。リモートの 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 Xhost コマンドを実行してアクセス許可を付与する必要があります。または、代

わりに **xhost +** を使用して、任意の XClient が XServer に接続できるようにすることもできます。

- **Xauthority**。`.Xauthority` ファイルは、各ユーザーのホームディレクトリにあります。これは、XServer の認証のために xauth によって使用される Cookie に資格情報を保存するために使用されます。XServer インスタンス (Xorg) が起動すると、Cookie はその特定のディスプレイへの接続を認証するために使用されます。

仕組み

Xorg が起動すると、`.Xauthority` ファイルが Xorg に渡されます。この `.Xauthority` ファイルには、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

このファイルは、`xauth` コマンドを使用して参照できます。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 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 ディスプレイ用に Linux VDA で **Xauthority** を有効にするには、次の2つのレジストリキーを作成する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"XauthEnabled" -d "0x00000001" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"ListenTCP" -d "0x00000001" --force
```

Xauthorityを有効にした後、`.Xauthority`ファイルを**XClient**に手動で渡すか、共有ホームディレクトリをマウントして渡します。

- `.Xauthority`ファイルをXClientに手動で渡す

  ICA®セッションを起動すると、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XClientマシンにコピーし、**DISPLAY**および**XAUTHORITY**環境変数を設定できます。`DISPLAY`は`.Xauthority`ファイルに保存されているディスプレイ番号で、`XAUTHORITY`は**Xauthority**のファイルパスです。例については、次のコマンドを参照してください。

  ```
  export DISPLAY={Display number stored in the Xauthority file}



  export XAUTHORITY={the file path of .Xauthority}
  ```

  注:

  XAUTHORITY環境変数が設定されていない場合、デフォルトで`~/.Xauthority`ファイルが使用されます。

- 共有ホームディレクトリをマウントして`.Xauthority`ファイルをXClientに渡す

  便利な方法は、ログオンユーザー用に共有ホームディレクトリをマウントすることです。Linux VDAがICAセッションを開始すると、`.Xauthority`ファイルはログオンユーザーのホームディレクトリの下に作成されます。このホームディレクトリがXClientと共有されている場合、ユーザーはこの`.Xauthority`ファイルをXClientに手動で転送する必要はありません。**DISPLAY**および**XAUTHORITY**環境変数が正しく設定されると、GUIはXServerデスクトップに自動的に表示されます。

トラブルシューティング

Xauthorityが機能しない場合は、以下のトラブルシューティング手順に従ってください。

1. root権限を持つ管理者として、すべてのXorgクッキーを取得します。

   ```
   ps aux | grep -i xorg
   ```

   このコマンドは、Xorgプロセスと、起動時にXorgに渡されたパラメーターを表示します。別のパラメーターは、どの`.Xauthority`ファイルが使用されているかを表示します。例：

   ```
   /var/xdl/xauth/.Xauthority110
   ```

   Xauthコマンドを使用してクッキーを表示します。

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   ```

2. `Xauth`コマンドを使用して、`~/.Xauthority`に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーはXorgとXClientの`.Xauthority`ファイルで同じである必要があります。
3. クッキーが同じ場合は、Linux VDAのIPアドレスと公開デスクトップのディスプレイ番号を使用して、リモートディスプレイポートのアクセシビリティを確認します。

   たとえば、XClientマシンで次のコマンドを実行します。

   ```
   telnet 10.158.11.11 6160
   ```

   ポート番号は6000 + \<ディスプレイ番号\>の合計です。

   このtelnet操作が失敗した場合、ファイアウォールが要求をブロックしている可能性があります。

認証

このセクションには、次のトピックが含まれています。

- Azure Active Directory を使用した認証(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/aad-authentication.html)

- ダブルホップシングルサインオン認証(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/single-sign-on.html)
- フェデレーション認証サービス
- 非SSO認証(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/non-sso-authentication.html)
- スマートカード(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/support-for-smart-cards.html)
- 未認証（匿名）ユーザーによるアクセス(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/access-by-unauthenticated-users.html)

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 でのみ使用できます。
>
> 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 サービスと統合され、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 22.04、20.04
- RHEL 8.8、8.6、7.9
- SUSE 15.4

詳細については、このMicrosoftの記事を参照してください。

既知の問題と回避策

RHEL 7.9 では、AAD ユーザー認証後に PAM (Pluggable Authentication Module) `pam_loginuid.so` が `loginuid` の設定に失敗します。この問題により、AAD ユーザーは VDA セッションにアクセスできなくなります。

この問題を回避するには、`/etc/pam.d/remote` で行 `Session required pam_loginuid.so` をコメントアウトします。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。

```
#%PAM-1.0
auth       substack     password-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
#session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      password-auth
session    include      postlogin
```

手順 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ポータルでテンプレートを作成します。**Review + create** をクリックする前に、**Management** タブで **Login with Azure AD** を必ず選択してください。

   AADでのログインを有効にする設定(/en-us/linux-virtual-delivery-agent/2308/media/aad-auth-create-vm.png)

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の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、Machine Creation Services (MCS) を使用してLinux VDAを作成するの**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をドメイン非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をドメイン非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、`NonDomainJoined`の値を`2`に変更します。例については、以下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

ドメイン非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、テンプレートVMからLinux VMを作成するためにMachine Creation Servicesを使用することを選択します。詳しくは、Citrix DaaSの記事「ドメインに参加していない」を参照してください。

ステップ5：Linux VMにAADユーザーアカウントを割り当てる

ドメインに参加していないVMを作成したら、それらにAADユーザーアカウントを割り当てます。

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. 「**ID > システム割り当て済み**」タブで、「**システムID**」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/2308/media/aad-auth-identity-setting.png)
3. 「**アクセス制御 (IAM) > ロールの割り当て**」タブで、「**仮想マシンユーザーログイン**」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設定(/en-us/linux-virtual-delivery-agent/2308/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設定(/en-us/linux-virtual-delivery-agent/2308/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、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アカウントまたは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。

ダブルホップシングルサインオン認証

StoreFrontストアにアクセスするためのユーザー資格情報は、Citrix Workspaceアプリ for LinuxおよびCitrix Receiver for Linux 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クライアントを使用して仮想デスクトップとアプリケーションにアクセスでき、ユーザー資格情報を再度入力する必要はありません。

注:

この機能は、Citrix Workspaceアプリ for LinuxおよびCitrix Receiver for Linux 13.10でサポートされています。

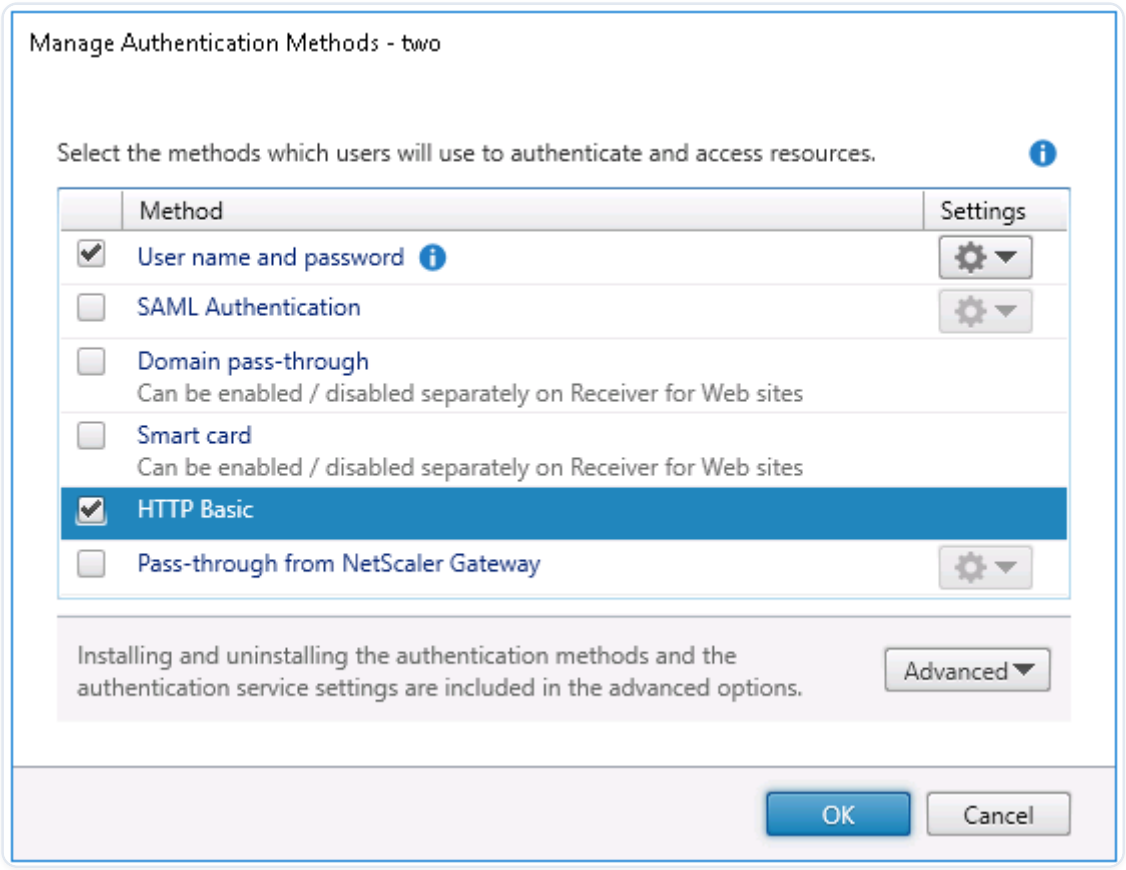
この機能を有効にするには：

1. Linux VDAに、Citrix Workspaceアプリ for LinuxまたはCitrix Receiver for Linux 13.10をインストールします。

   Citrix WorkspaceアプリまたはCitrix ReceiverのCitrixダウンロードページからアプリをダウンロードします。

デフォルトのインストールパスは/opt/Citrix/ICAClient/です。アプリを別のパスにインストールする場合は、ICAROOT環境変数を実際のインストールパスを指すように設定します。

2. Citrix StoreFront™管理コンソールで、ターゲットストアに**HTTP Basic**認証方法を追加します。



3. HTTP Basic認証を許可するために、AuthManager構成ファイル（$ICAROOT/config/AuthManConfig.xml）に次のキーを追加します。

```
<Protocols>

     <HTTPBasic>

         <Enabled>True</Enabled>

    </HTTPBasic>

</Protocols>
```

4. 指定されたディレクトリにルート証明書をインストールするには、次のコマンドを実行します。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. この機能を有効にするには、次のコマンドを実行します。

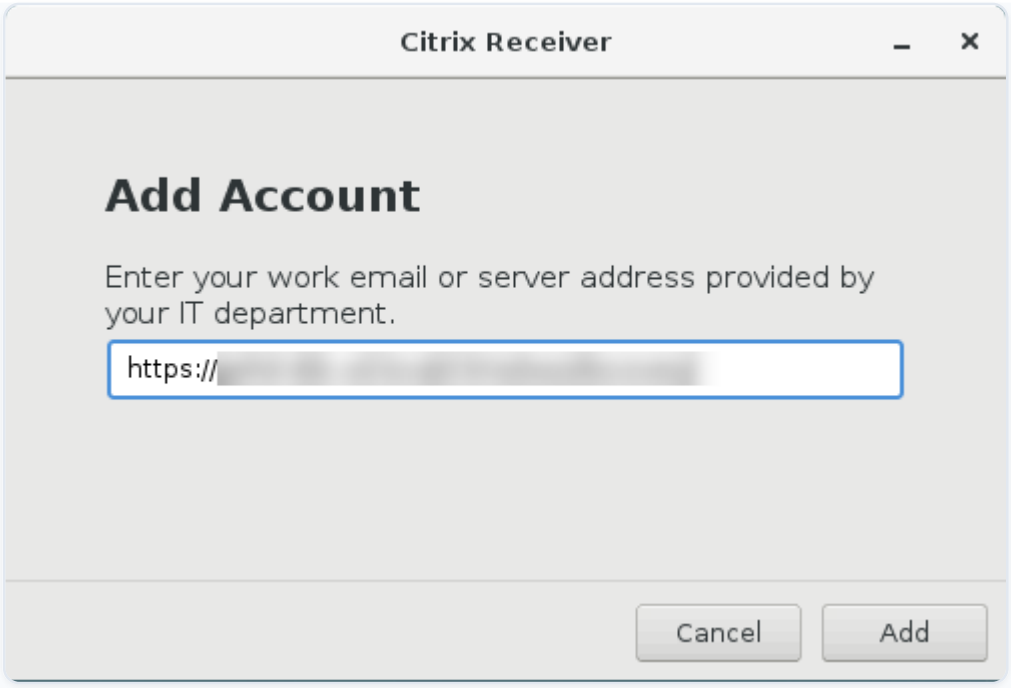
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

6. Linux仮想デスクトップセッションを起動し、そのセッション内でCitrix Workspaceアプリ for LinuxまたはCitrix Receiver for Linux 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、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に特化した追加情報を提供します。

注:

- Linux VDAは、**セッション内動作**ポリシーをサポートしていません。
- Linux VDAは、FASサーバーとのデータ転送に短い接続を使用します。
- 2206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のインストール記事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

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
アマゾン Linux 2	はい	はい	はい	はい
デビアン 11.3	はい	はい	はい	はい
RHEL 9.2/9.0	はい	はい	いいえ	いいえ
RHEL 8.8/8.6	はい	はい	はい	はい
RHEL 7.9、CentOS 7.9	はい	はい	はい	はい
ロッキー Linux 9.2/9.0	はい	はい	いいえ	いいえ
ロッキー Linux 8.8/8.6	はい	はい	いいえ	いいえ

	ウィンバインド	SSSD	セントリファイ	PBIS
SUSE 15.4	はい	はい	はい	いいえ
ウブントゥ 22.04/20.04	はい	はい	はい	はい

Linux VDA での FAS の構成

RHEL 8.x/9.x および Rocky Linux 8.x/9.x での FAS のサポート

FAS は pam_krb5 モジュールに依存しますが、これは RHEL 8.x および Rocky Linux 8.x では非推奨です。マルチセッション OS モードで提供される RHEL 8.x および Rocky Linux 8.x マシンで FAS を使用する場合は、次の手順が必要です。シングルセッション OS (VDI) モードで提供される RHEL 8.x および Rocky Linux 8.x マシンで FAS を使用する場合は、次の手順をスキップできます。

1. 次の Web サイトから pam_krb5-2.4.8-6 ソースコードをダウンロードします:

   `<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>`。

2. RHEL 8.x および Rocky Linux 8.x に pam_krb5 モジュールをビルドしてインストールします。

```
yum install make gcc krb5-devel pam-devel autoconf libtool



rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio -div



tar xvzf pam_krb5-2.4.8.tar.gz



cd pam_krb5-2.4.8



./configure --prefix=/usr
```

```
make



make install
```

3. pam_krb5.so が /usr/lib64/security/ に存在することを確認します。

```
ls -l /usr/lib64/security | grep pam_krb5
```

FAS サーバーの設定

新規の Linux VDA インストールで FAS を使用するには、ctxinstall.sh または ctxsetup.sh を実行するときに、各 FAS サーバーの FQDN を入力します。

注:

FAS サーバーは AD グループポリシーを介して構成されます。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ないため、代わりにセミコロンで区切られた FAS サーバーのリストを指定できます。次の点に注意してください:

- リストの順序は、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インデックスは変更しないでください。

既存のLinux VDAインストールをアップグレードするには、`ctxsetup.sh`を再実行してFASサーバーを設定できます。または、次のコマンドを実行してFASサーバーを設定し、`ctxvda`サービスを再起動して設定を有効にできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

```
service ctxjproxy restart


service ctxvda restart
```

`ctxreg`を介してFASサーバーを更新するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -v "Addresses" -d "<Your-FAS-Server-List>"


service ctxjproxy restart


service ctxvda restart
```

証明書のインストール

ユーザー証明書の検証のために、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**の手順からADルート証明書を取得するか、ルートCAサーバー`http://CA-SERVER/certsrv`からDER形式をダウンロードします。

> **注:**
>
> 次のコマンドは、中間証明書の設定にも適用されます。
>
> DERファイル（.crt、.cer、.der）をPEMに変換するには、次の類似コマンドを実行します。

```
sudo openssl x509 -inform der -in root.cer -out root.pem
```

次に、次の類似コマンドを実行して、ルートCA証明書を`openssl`ディレクトリにインストールします。

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

注:

ルートCA証明書を**/root**パスの下に置かないでください。そうしないと、FASはルートCA証明書への読み取り権限を持ちません。

ctxfascfg.shの実行

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
```

環境変数が追加され、`ctxfascfg.sh`をサイレントモードで実行できるようになります。

- **CTX_FAS_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis**: 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。

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

その後、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。

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



service ctxjproxy restart



service ctxvda restart
```

制限事項

- FAS はまだロック画面をサポートしていません。セッションでロックボタンをクリックすると、FAS を使用してセッションに再度ログオンすることはできません。
- このリリースでは、Federated Authentication Service のアーキテクチャの概要 の記事にまとめられている一般的な FAS 展開のみがサポートされており、**Windows 10 Azure AD Join** は含まれていません。

トラブルシューティング

FAS のトラブルシューティングを行う前に、Linux VDA が正しくインストールおよび構成されており、パスワード認証を使用して共通ストアで非 FAS セッションが正常に起動できることを確認してください。

非 FAS セッションが正常に動作する場合は、**Login** クラスの HDX ログレベルを VERBOSE に、VDA ログレベルを TRACE に設定します。Linux VDA のトレースログを有効にする方法については、Knowledge Center の記事 CTX220130 を参照してください。

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することもできます。詳細については、**XDPing** を参照してください。

FAS サーバー構成エラー

FAS ストアからのセッション起動に失敗します。

/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [Logon
Type] Federated Authentication Logon.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: entry



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
connect to server 0



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0: failed
to connect: Connection refused.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: failed
to connect to server [0], please confirm if fas service list is well
configurated in condb



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit,
43



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: failed
to validate fas credential



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate:
failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER
```

解決策

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

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

既存の設定が正しくない場合は、前述の FAS サーバーの設定 の手順に従って再度設定してください。

CA証明書の構成が正しくありません

FASストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。

ルートCA証明書の構成が正しくないため、ログオンが無効です(/en-us/linux-virtual-delivery-agent/2308/media/incorrect-root-ca-cert-config.png)

/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: entry



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller: current
process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_public_certificate: entry



2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
for response...



2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query to
server success
```

```
2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
get_public_certificate: exit

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
input size 1888

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
output size 1415

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: get logon certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
cache certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: exit, 0

2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策

ルートCA証明書とすべての中間証明書を格納する完全なパスが`/etc/krb5.conf`に正しく設定されていることを確認してください。完全なパスは次のようになります。

```
 [realms]

EXAMPLE.COM = {

    ......

    pkinit_anchors = DIR:/etc/pki/CA/certs/

    ......

}
```

既存の設定が正しくない場合は、前述のの手順(/ja-jp/linux-virtual-delivery-agent/2308/configure/authentication/federated-authentication-service.html#install-certificates)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ています。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。

シャドウアカウントマッピングエラー(/en-us/linux-virtual-delivery-agent/2308/media/shadow-account-mapping-error.png)

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

解決策

ADでシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に次のエラーが発生します。

ADFSが構成されていません(/en-us/linux-virtual-delivery-agent/2308/media/adfs-not-configured.png)

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この問題が発生します。

解決策

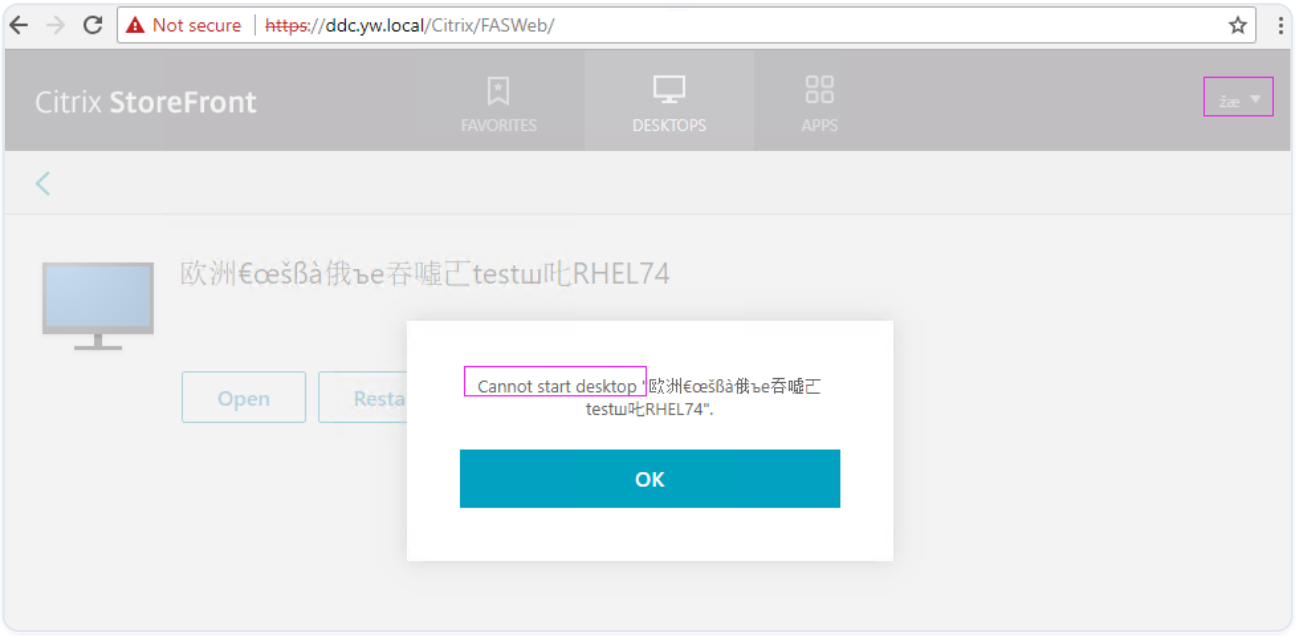
フェデレーション認証サービス用にADFS IdPを展開します。詳細については、「フェデレーション認証サービス ADFS展開」(/ja-jp/federated-authentication-service/current-release/deployment-architectures/adfs.html)を参照してください。

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のアーキテクチャの概要」(/ja-jp/federated-authentication-service/current-release/deployment-architectures.html)の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービスの高度な構成」(/ja-jp/federated-authentication-service/current-release/config-manage.html)の章で紹介されています。

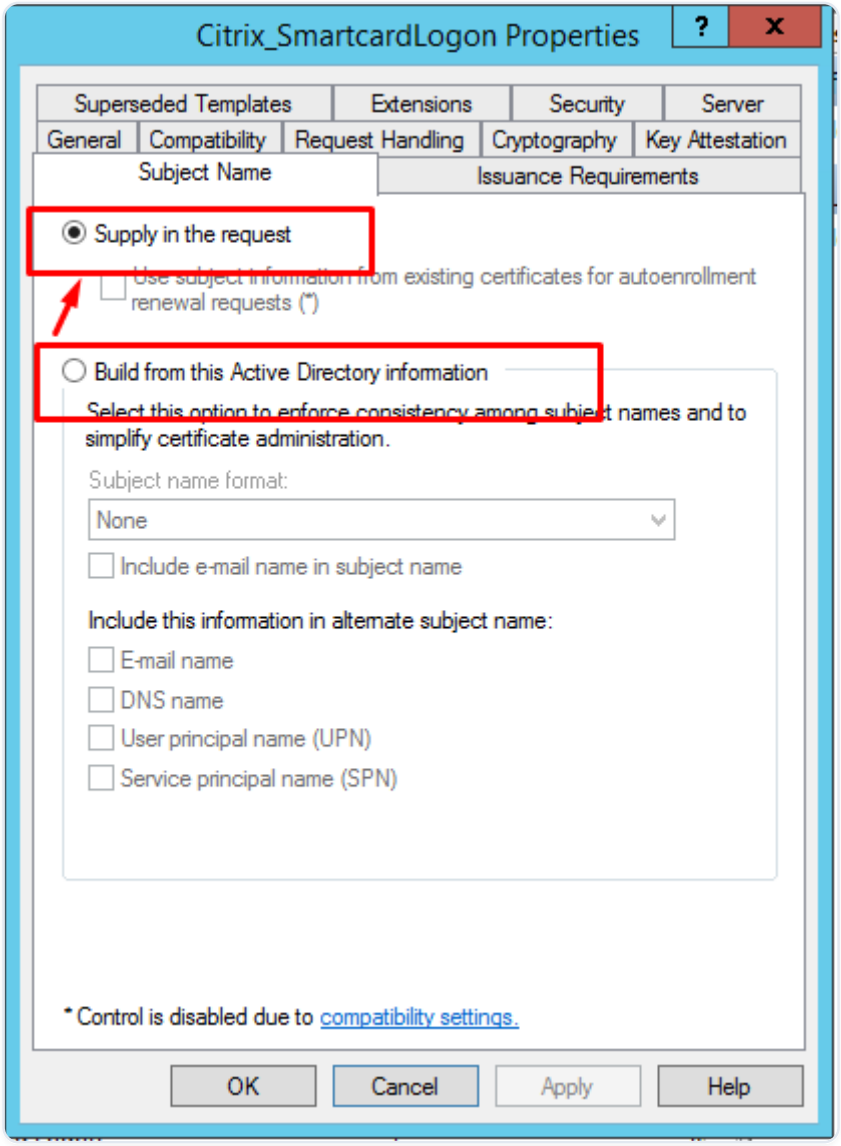
既知の問題

FASを使用している場合、非英語文字を含む公開デスクトップまたはアプリセッションを起動しようとすると失敗することがあります。



回避策

CAツールで**Manage Templates**を右クリックし、**Citrix_SmartcardLogon**テンプレートを**Build from this Active Directory information**から**Supply in the request**に変更します。



FIDO2 (プレビュー)

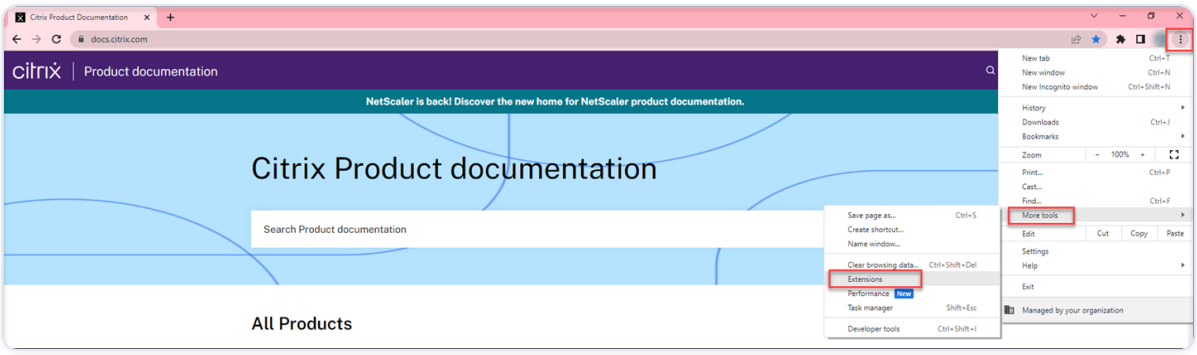
Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするように、FIDO2認証を設定できます。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない可能性があり、非実稼働環境での使用が推奨されます。Citrixテクニカルサポートは、プレビュー機能で発見された問題についてはサポートしません。

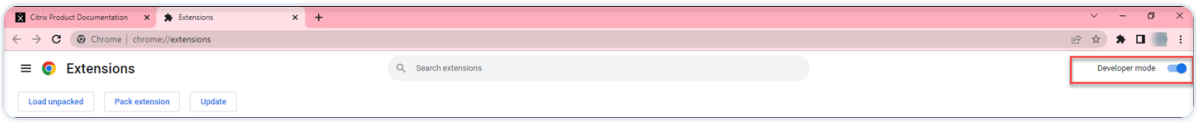
> Linux VDAは、FIDO2とGoogle Chromeの組み合わせのみをサポートしています。

> 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をクリックしてその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。



   c. **デベロッパーモード**をオンに切り替えます。



   d. **パッケージ化されていない拡張機能を読み込む**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ガイダンスを提供します。

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。次の表は、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ています。

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
```

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。使用例は次のとおりです。

- ドキュメントにデジタル署名を追加する
- メールを暗号化または復号化する
- Webサイトへの認証

この機能について、Linux VDAはWindows VDAと同じ構成を使用します。詳細については、この記事の「スマートカード環境の構成」セクションを参照してください。

> **注:**
>
> Linux VDAセッション内でマップされたスマートカードを使用してCitrix Gatewayにサインオンすることはサポートされていません。

前提条件

スマートカードパススルー認証の可用性は、次の条件に依存します。

- お使いのLinux VDAは、次のいずれかのディストリビューションにインストールされています。
  - RHEL 9.2/9.0
  - RHEL 8.8/8.6
  - RHEL 7、CentOS 7
  - ロッキー Linux 9.2/9.0
  - ロッキー Linux 8.8/8.6
  - ウブントゥ 22.04
  - ウブントゥ 20.04
  - デビアン 11.3

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

- OpenSCでサポートされているスマートカードが使用されます。詳細については、「OpenSCがスマートカードをサポートしていることを確認する」を参照してください。
- Citrix Workspace™ app for Windowsが使用されます。

OpenSCがスマートカードをサポートしていることを確認する

OpenSCは、RHEL 7.4以降で広く使用されているスマートカードドライバーです。CoolKey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、構成を説明するための例としてYubiKeyスマートカードを使用します。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ます。OpenSCのインストールについては、「スマートカードドライバーのインストール」を参照してください。スマートカードを挿入し、次のコマンドを実行して、OpenSCがスマートカードをサポートしていることを確認します。

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. 通常、CAサーバーからPEM形式のルート証明書を取得します。

   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。次のコマンド例では、**certnew.cer**がDERファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   ```

2. ルート証明書を`openssl`ディレクトリにインストールします。**certnew.pem**ファイルは例として使用されます。

   ```
   cp certnew.pem <path where you install the root certificate>
   ```

ルート証明書をインストールするためのパスを作成するには、`sudo mdkir -p <path where you install the root certificate>`を実行します。

RHEL 8.x/9.x および Rocky Linux 8.x/9.x 上で pam_krb5 モジュールを構築する

スマートカード認証はpam_krb5モジュールに依存しますが、これはRHEL 8.xおよびRocky Linux 8.xでは非推奨です。マルチセッションOSモードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでスマートカード認証を使用する場合、以下の手順が必要です。シングルセッションOS (VDI) モードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでのスマートカード認証の場合、以下の手順をスキップできます。

1. pam_krb5-2.4.8-6のソースコードを`<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>`からダウンロードします。
2. RHEL 8.xおよびRocky Linux 8.xにpam_krb5モジュールをビルドしてインストールします。

```
yum install -y opensc pcsc-lite pcsc-lite-libs pcsc-lite-ccid nss-tools

yum install gcc krb5-devel pam-devel autoconf libtool

rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio –div

tar xvzf pam_krb5-2.4.8.tar.gz

cd pam_krb5-2.4.8

./configure --prefix=/usr

make

make install
```

3. /usr/lib64/security/の下にpam_krb5.soが存在することを確認します。

```
ls -l /usr/lib64/security | grep pam_krb5
```

スマートカード環境を構成する

ctxsmartlogon.shスクリプトを使用してスマートカード環境を構成するか、または手動で構成を完了できます。

(オプション1) CTXSMARTLOGON.SHスクリプトを使用してスマートカード環境を構成する

> **注:**
>
> ctxsmartlogon.shスクリプトは、PKINIT情報をデフォルトレルムに追加します。この設定は、**/etc/krb5.conf**構成ファイルを通じて変更できます。
>
> スマートカードを初めて使用する前に、ctxsmartlogon.shスクリプトを実行してスマートカード環境を構成します。

> **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。

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

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行し、「はい」を選択する(/en-us/linux-virtual-delivery-agent/2308/media/smart-card-logon-script1.png)

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

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

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行し、「いいえ」を選択する(/en-us/linux-virtual-delivery-agent/2308/media/smart-card-logon-script2.png)

(オプション 2) スマートカード環境を手動で構成する

Linux VDAはWindows VDAと同じスマートカード環境を使用します。この環境では、ドメインコントローラー、Microsoft証明機関 (CA)、インターネットインフォメーションサービス、Citrix StoreFront、Citrix Workspaceアプリなど、複数のコンポーネントを構成する必要があります。YubiKey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。

次の手順に進む前に、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レベルで実装されています。

PC/SC Liteパッケージをインストールするには、次のコマンドを実行します。

RHEL 9.2/9.0/8.x、Rocky リナックス 9.2/9.0/8.x、RHEL 7/CentOS 7の場合：

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

Ubuntu 22.04、Ubuntu 20.04、Debian 11.3のバージョン：

```
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 9.2/9.0/8.x、Rocky Linux 9.2/9.0/8.x、RHEL 7/CentOS 7 の各バージョンに対応しています:

```
yum install opensc
```

Ubuntu 22.04、Ubuntu 20.04、Debian 11.3 の各バージョンに対応しています:

```
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 7/CentOS 7:

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

RHEL 9.2/9.0/8.x、Rocky Linux 9.2/9.0/8.x 向け:

```
yum install krb5-pkinit
```

Ubuntu 22.04 および Ubuntu 20.04:

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

Debian 11.3 の場合:

```
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、キー配布センター（KDC）からチケット許可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構成ファイルで、実際のレルムに従ってPKINIT情報を追加します。

注:

pkinit_cert_matchオプションは、PKINIT認証を試行する前にクライアント証明書が一致する必要がある照合ルールを指定します。照合ルールの構文は次のとおりです。

コンポーネントルール ...

ここで、**relation-operator**は、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。

「追加されたPKINIT情報」(/en-us/linux-virtual-delivery-agent/2308/media/pkinit-information-new2.png)
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。認証モジュールとしてpam_krb5を追加するには、**/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が使用されている場合、変更後の構成ファイルは次のようになります。

「SSSDが使用されている場合の変更された構成ファイル」(/en-us/linux-virtual-delivery-agent/2308/media/smartcard-authentication-file.png)

（オプション）スマートカードを使用したシングルサインオン

シングルサインオン（SSO）は、仮想デスクトップおよびアプリケーションの起動時にパススルー認証を実装するCitrixの機能です。この機能により、ユーザーがPINを入力する回数が減ります。Linux VDAでSSOを使用するには、Citrix Workspaceアプリを構成します。構成はWindows VDAと同じです。詳しくは、Knowledge Centerの記事CTX133982を参照してください。

Citrix Workspaceアプリでグループポリシーを構成する際に、スマートカード認証を次のように有効にします。

「Workspaceアプリでのスマートカード認証の有効化」(/en-us/linux-virtual-delivery-agent/2308/media/enable-smart-card-auth-in-receiver.jpg)

高速スマートカードログオン

高速スマートカードは、既存のHDX PC/SCベースのスマートカードリダイレクトを改善したものです。これにより、高遅延のWAN環境でスマートカードを使用する際のパフォーマンスが向上します。詳しくは、スマートカードを参照してください。

Linux VDAは、Citrix Workspaceアプリの以下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。

- Windows 用 シトリックス レシーバー 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
```

クライアントでの高速スマートカードログオンを無効にする

クライアントでの高速スマートカードログオンを無効にするには、関連付けられたStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

XDPingを実行する

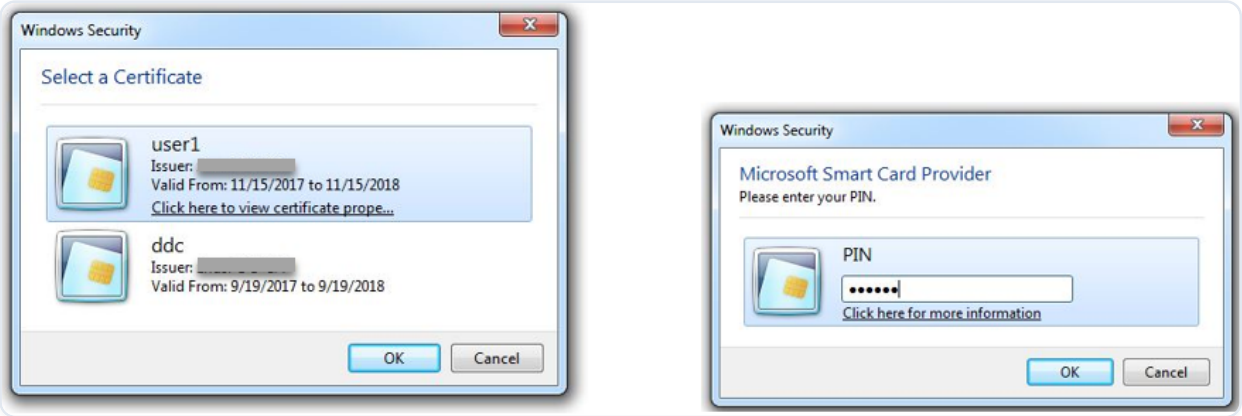
前述の手順を完了した後、Linux VDA環境に存在する可能性のある一般的な構成の問題を確認するために、Linux **XDPing**ツールを使用できます。詳細については、「XDPing」を参照してください。

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

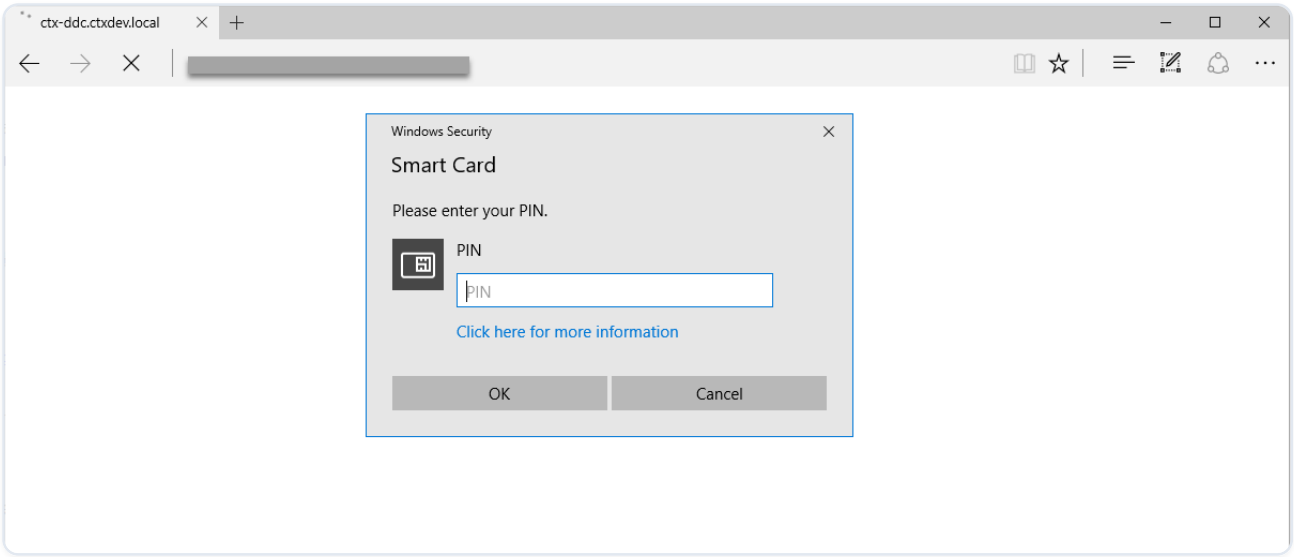
SSOシナリオと非SSOシナリオの両方で、スマートカードを使用してLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用して、StoreFront™に自動的にログオンします。StoreFrontでLinux仮想デスクトップセッションを起動すると、PINはスマートカード認証のためにLinux VDAに渡されます。
- 非SSOシナリオでは、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、スマートカードが接続されていないために再認証が失敗し、Linux VDA側に灰色のキャッシュウィンドウが表示されてすぐに終了します。この場合、スマートカードを接続するように促す他のプロンプトは表示されません。

ただし、StoreFront側では、セッションに再接続するときにスマートカードが接続されていない場合、StoreFront Webは次のようなアラートを表示する場合があります。

スマートカードを挿入している画像(/en-us/linux-virtual-delivery-agent/2308/media/please-insert-smart-card.png)

制限事項

限られたLinuxディストリビューションとAD統合方法のサポート

- スマートカードパススルー認証は、限られたLinuxディストリビューションとAD統合方法をサポートします。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
デビアン 11.3	はい	はい	はい
RHEL 9.2/9.0	はい	はい	いいえ
RHEL 8.8/8.6	はい	はい	はい
RHEL 7.9, CentOS 7.9	はい	はい	はい

	ウィンバインド	SSSD	セントリファイ
ロッキー リナックス 9.2/9.0	はい	はい	いいえ
ロッキー リナックス 8.8/8.6	はい	はい	いいえ
ウブントゥ 22.04/20.04	はい	はい	はい

スマートカード取り外しポリシー

Linux VDAは、スマートカードの取り外しに対して既定の動作のみを使用します。Linux VDAに正常にログオンした後でスマートカードを取り外しても、セッションは接続されたままで、セッション画面はロックされません。

その他のスマートカードとPKCS#11ライブラリのサポート

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ライブラリを試すことができます。特定のスマートカードまたはPKCS#11ライブラリに切り替えるには：

1. すべての `opensc-pkcs11.so` インスタンスを、ご使用のPKCS#11ライブラリに置き換えます。
2. PKCS#11ライブラリのパスをレジストリに設定するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v
   "PKCS11LibPath" -d "PATH"
   ```

   ここで、**PATH** は /usr/lib64/pkcs11/opensc-pkcs11.so のようなPKCS#11ライブラリを指します。
3. クライアントで高速スマートカードログオンを無効にします。

認証されていない（匿名）ユーザーによるアクセス

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

注:

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

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（匿名）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、「認証されていないストアを作成する」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や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することができます。詳しくは、「デリバリーグループを作成する」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には、デフォルトで10分間のアイドルタイムアウトがあり、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**を通じ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に、次の点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。**

レジストリが次のように更新されていることを確認してください (0に設定):

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsd サービスが実行されており、**passwd** キャッシュを有効にするように構成されていることを確認してください:

```
ps  uax | grep nscd

cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

passwd キャッシュ変数が有効になっている場合は **no** に設定し、**ncsd** サービスを再起動します。この構成を変更した後、Linux VDAを再インストールする必要がある場合があります。

- **KDEを使用する認証されていないユーザーセッションでロック画面ボタンが表示されます。**

ロック画面ボタンとメニューは、認証されていないユーザーセッションではデフォルトで無効になっています。ただし、KDEでは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** に次の行を追加します。例:

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、`KDE Action Restrictions` パラメーターが `/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals` のようなグローバルな `kdeglobals` ファイルで不変として構成されている場合、ユーザー構成は効果がありません。

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、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＆ペーストできます。この機能には、Citrix Virtual Apps and Desktops™ 2006以降、およびWindows用Citrix Workspace™アプリ1903以降が必要です。

ファイルを正常にコピー＆ペーストするには、以下を確認してください。

- ファイルの最大数が20を超えないこと。
- ファイルの最大サイズが200 MBを超えないこと。
- 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。

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＆ペースト機能は、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＆ペースト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。

制限事項

- 切り取りはサポートされていません。ファイルを切り取る要求は、コピー操作として扱われます。
- ドラッグ＆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。
- ファイルコピーアンドペーストは、順次実行する必要があります。前のファイルが正常にコピーアンドペーストされてからでないと、次のファイルをコピーできません。

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を使用すると、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のファイルのアップロードとダウンロードが可能になります。クライアントデバイスからLinux VDAにファイルをアップロードするには、Citrix Workspaceアプリのツールバーにある**アップロード**アイコンをクリックし、ファイルダイアログから目的のファイルを選択します。Linux VDAからクライアントデバイスにファイルをダウンロードするには、**ダウンロード**アイコンをクリックします。アップロードまたはダウンロード中にファイルを追加できます。一度に最大100個のファイルを転送できます。

ツールバーのアップロードアイコンの画像(/en-us/linux-virtual-delivery-agent/2308/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、自動的にクライアントデバイスに転送されます。

注:

自動ダウンロードには、**デスクトップとクライアント間のファイル転送を許可**および**デスクトップからファイルをダウンロード**ポリシーが**許可**に設定されている必要があります。

自動ダウンロードのユースケースをいくつか示します。

- **Save to My Device**にファイルをダウンロードする

  公開されたデスクトップおよびWebブラウザアプリセッション内で、Webサイトからダウンロードしたファイルは、VDA上の**Save to My Device**ディレクトリに保存され、クライアント

デバイスに自動的に転送されます。自動ダウンロードを実現するには、セッション内Webブラウザのデフォルトダウンロードディレクトリを**Save to My Device**に設定し、Citrix Workspace app for HTML5またはfor Chromeを実行するWebブラウザでローカルダウンロードディレクトリを設定します。

- **Save to My Device**にファイルを移動またはコピーする

  公開されたデスクトップセッション内で、対象ファイルを選択し、クライアントデバイスで利用できるように**Save to My Device**ディレクトリに移動またはコピーします。

ファイル転送ポリシー

Citrix Studioを使用してファイル転送ポリシーを設定できます。デフォルトでは、ファイル転送は有効になっています。

ファイル転送ポリシーの画像(/en-us/linux-virtual-delivery-agent/2308/media/file-transfer-policies.png)

ポリシーの説明：

- **デスクトップとクライアント間のファイル転送を許可する。** Citrix Virtual Apps and Desktops™セッションとクライアントデバイス間のファイル転送を許可または禁止します。
- **デスクトップからファイルをダウンロードする。** Citrix Virtual Apps™ and Desktopsセッションからクライアントデバイスへのファイルダウンロードを許可または禁止します。
- **デスクトップにファイルをアップロードする。** クライアントデバイスからCitrix Virtual Apps and Desktopsセッションへのファイルダウンロードを許可または禁止します。

> **注:**
>
> **デスクトップからファイルをダウンロードする**および**デスクトップにファイルをアップロードする**ポリシーが有効になるようにするには、**デスクトップとクライアント間のファイル転送を許可する**ポリシーを**許可**に設定します。

使用方法

Citrix Workspace 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接続**ポリシーを**許可**に設定します。

   WebSockets接続ポリシーの設定(/en-us/linux-virtual-delivery-agent/2308/media/websockets-connection.png)

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、**ストア**をクリックし、**Webサイト用Receiverの管理**ノードを選択し、**常にHTML5用Receiverを使用する**オプションを選択してCitrix Receiver™ for HTML5を有効にします。

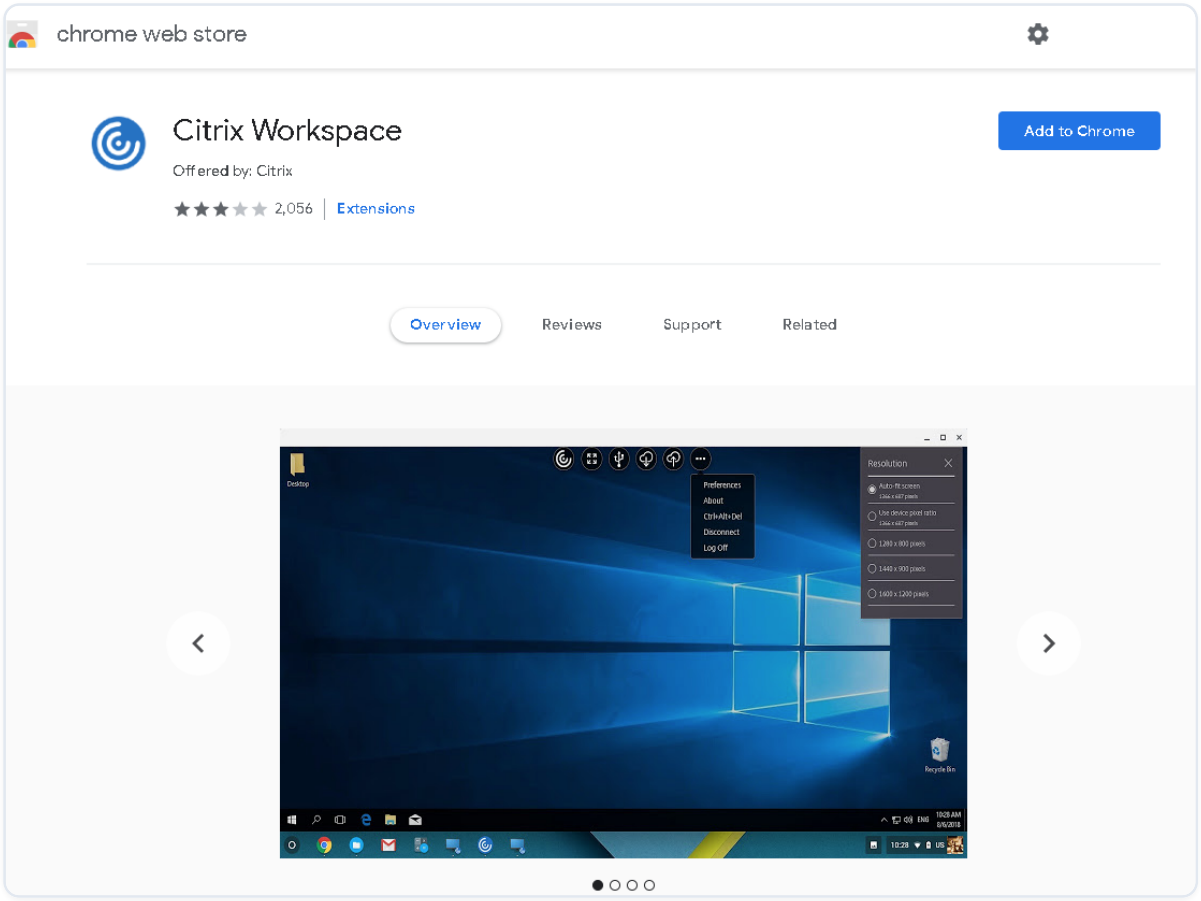
   常にHTML5用Receiverを使用するオプションを選択する図(/en-us/linux-virtual-delivery-agent/2308/media/always-use-receiver-for-html5.png)
4. 仮想デスクトップまたはWebブラウザアプリセッションを起動します。Linux VDAとクライアントデバイス間で1つ以上のファイル転送を実行します。

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ウェブストアからCitrix Workspaceアプリを入手します。

   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ます。

   a. Google Chromeの検索ボックスに「**Citrix Workspace for Chrome**」と入力します。検索アイコンをクリックします。
   b. 検索結果の中から、Citrix Workspaceアプリが利用可能なChromeウェブストアへのURLをクリックします。



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

3. Chromeアプリページで、Chrome向けCitrix Workspaceアプリをクリックします。
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プログレッシブ表示

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関する詳細については、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。DPI設定が混在する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で有効です。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設定に応じ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
174%以下	96 (1 x 96)
175%～274%	192 (2 x 96)
275%～399%	288 (3 x 96)
400%以上	384 (4 x 96)

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
- Citrix Workspaceアプリ for Mac (バージョン 2204.1 はサポートされていません)
- Citrix Workspaceアプリ for Windows (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/2308/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
充電中でバッテリー残量が多いバッテリー(/en-us/linux-virtual-delivery-agent/2308/media/	充電中（「+」記号で表示）	高（緑色で表示）	>=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
battery-charging-high.png)			
充電中で残量が中程度のバッテリー(/en-us/linux-virtual-delivery-agent/2308/media/battery-charging-medium.png)	充電中（「+」記号で表示）	中程度（琥珀色で表示）	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/2308/media/battery-charging-low.png)	充電中（「+」記号で表示）	低（赤色で表示）	< 20%
充電中でなく残量が多いバッテリー(/en-us/linux-virtual-delivery-agent/2308/media/battery-not-charging-high.png)	充電中ではない（「-」記号で表示）	高（緑色で表示）	>=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/2308/media/battery-not-charging-medium.png)	充電中ではない（「-」記号で表示）	中程度（琥珀色で表示）	20%以上80%未満
充電中でなく、残量が少ないバッテリー(/en-us/linux-virtual-delivery-agent/2308/media/battery-not-charging-low.png)	充電中ではない（「-」記号で示される）	低い（赤色で示される）	20%未満

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
不明なバッテリーステータス(/en-us/linux-virtual-delivery-agent/2308/media/unknown-battery-status.png)	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:

上記のコマンドは、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、次の追加手順を実行してください。

1. RHEL 8.x または SUSE 15.x を GNOME とともに使用している場合は、AppIndicator のサポートを有効にするために、GNOME シェル用の互換性のある拡張機能をインストールします。
   a. `gnome-shell --version` コマンドを実行して、GNOME シェルのバージョンを確認します。

b. `<https://extensions.gnome.org/extension/615/appindicator-support>` からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能のバージョンとして24または26を選択できます。

GNOME シェル拡張機能(/en-us/linux-virtual-delivery-agent/2308/media/gnome-extension.png)

c. ダウンロードしたパッケージを解凍します。パッケージ内の **metadata.json** ファイルにある **"uuid"** の値が appindicatorsupport@rgcjonas.gmail.com に設定されていることを確認します。

d. `mv` コマンドを実行して、appindicatorsupport@rgcjonas.gmail.com ディレクトリを `/usr/share/gnome-shell/extensions/` の下の場所に移動します。

e. `chmod a+r metadata.json` コマンドを実行して、**metadata.json** ファイルを他のユーザーが読み取れるようにします。

> **ヒント:**
>
> デフォルトでは、appindicatorsupport@rgcjonas.gmail.com ディレクトリ内の **metadata.json** ファイルは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` キーを順番に押すか、`killall -SIGQUIT gnome-shell` コマンドを実行して GNOME シェルをリロードします。

h. デスクトップ環境で GNOME Tweaks を実行し、Tweaks ツールで **KStatusNotifierItem/AppIndicator Support** を有効にします。

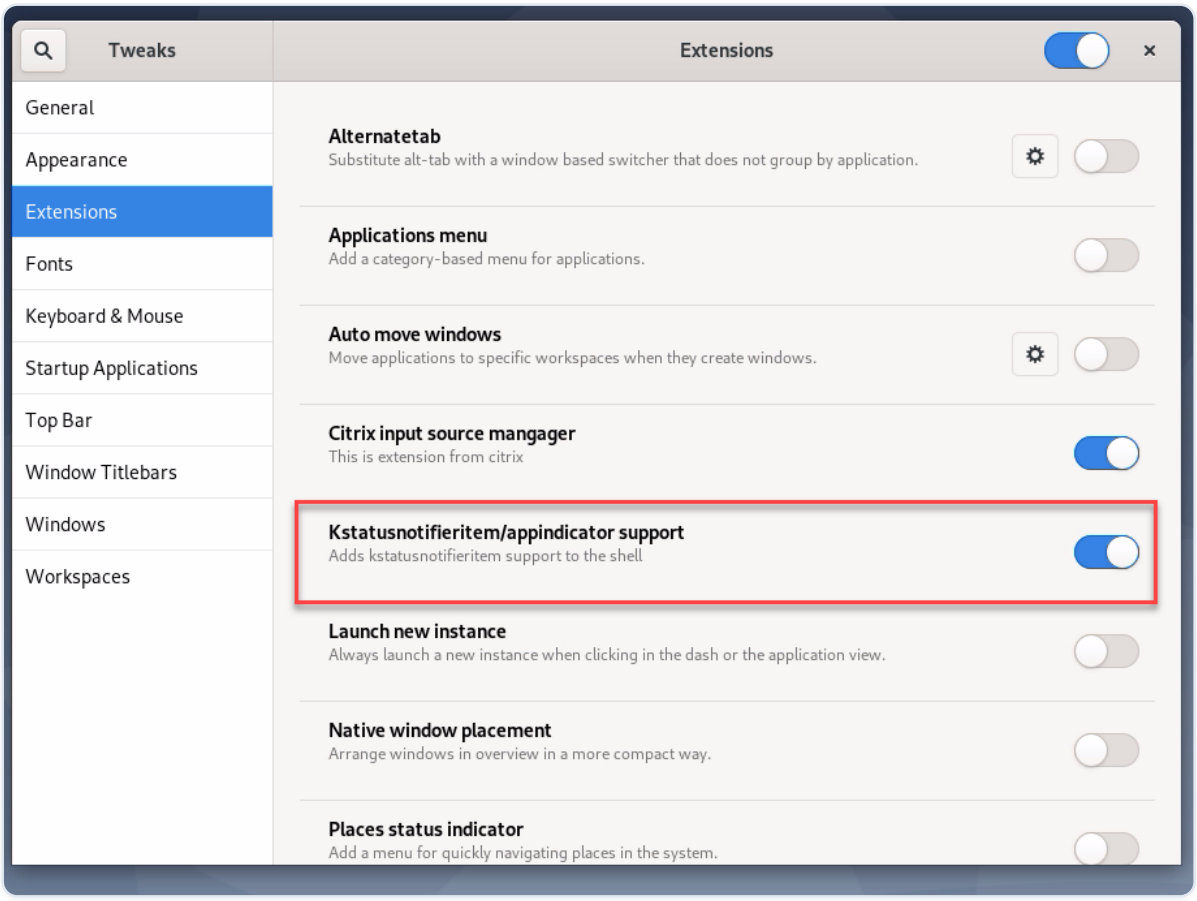
2. GNOME がインストールされた Debian 11.3 を使用している場合は、GNOME システムトレイアイコンをインストールして有効にするには、次の手順を実行します。

a. `sudo apt install gnome-shell-extension-appindicator` コマンドを実行します。GNOME が拡張機能を表示するには、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。



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。

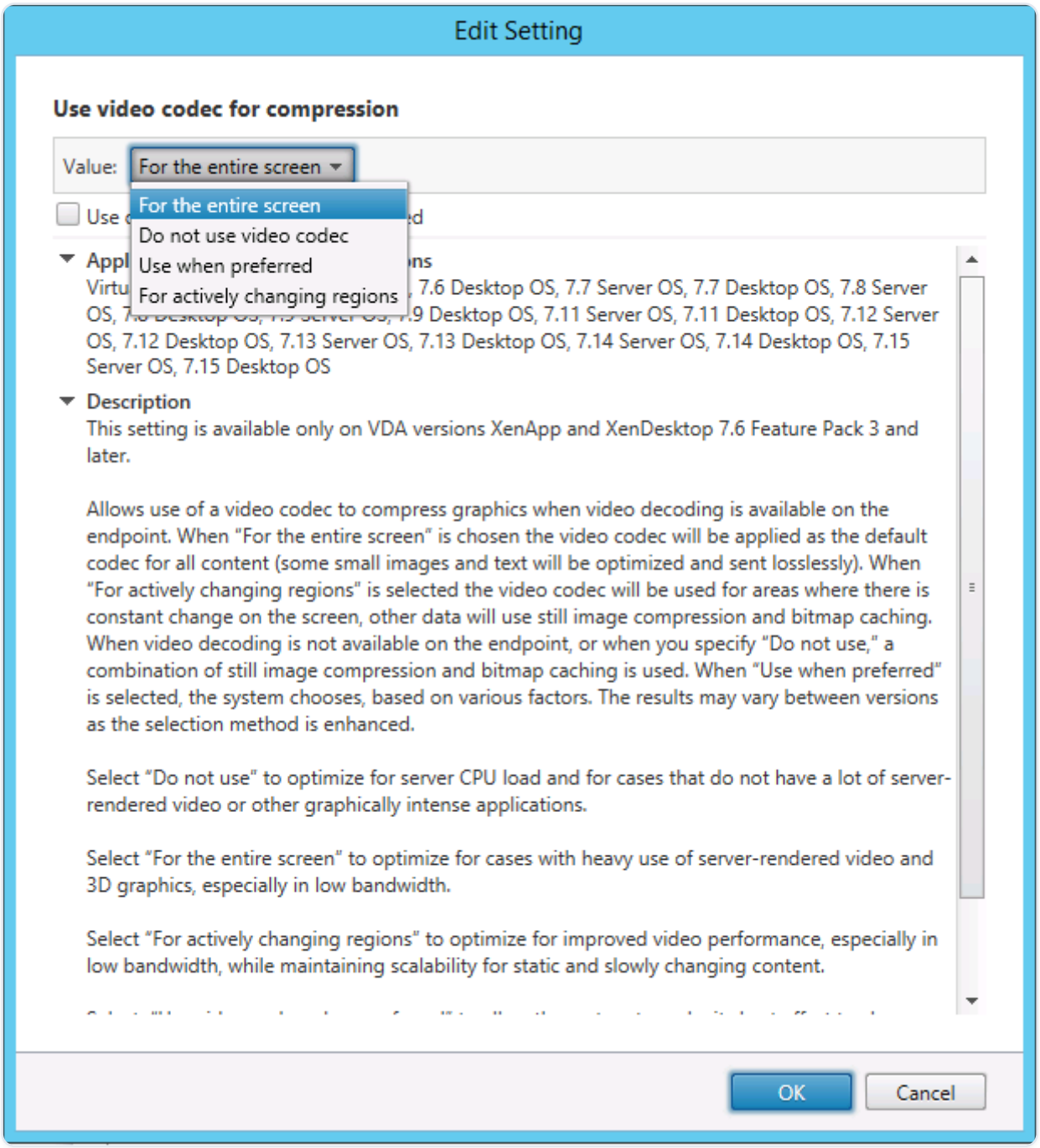
デフォルトでは、**3D グラフィックワークロード向けに最適化**は無効になっています。

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。

The 圧縮にビデオコーデックを使用 グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して以下のオプションを提供します。

- **優先する場合にのみ使用**。この設定はデフォルトです。追加の構成は必要ありません。これにより、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。
- **画面全体に対して**。Thinwire をフルスクリーン H.264 または H.265 で配信し、特に 3D グラフィックを多用する場合に、ユーザーエクスペリエンスと帯域幅の向上を最適化します。**画面全体に対して**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に対して**。Thinwire のアダプティブディスプレイテクノロジーは、動く画像（ビデオ、移動中の 3D）を識別します。画像が動いている画面の一部でのみ H.264 または H.265 を使用します。H.264 または H.265 ビデオコーデックの選択的な使用により、HDX Thinwire は、H.264 または H.265 ビデオコーデックを使用して頻繁に更新される画面の一部を検出し、エンコードできます。静止画圧縮（JPEG、RLE）とビットマップキャッシュは、テキストや写真画像を含む画面の残りの部分で引き続き使用されます。ユーザーは、ビデオコンテンツの帯域幅消費量の削減と品質の向上、およびロスレステキストまたは高品質の画像を他の場所で組み合わせるという利点を得られます。**視覚品質**ポリシーが**常にロスレス**または**ロスレスにビルド**に設定されている場合、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
- 視覚的品質

H.265/H.264 ハードウェアエンコード

ビデオコーデックにハードウェアエンコードを使用ポリシーでは、利用可能な場合に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できます。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で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**
- **アクティブに変更される領域**

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ドを使用**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**、または**アクティブに変更される領域**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。

H.265ビデオコーデックは、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。Linux VDAまたはCitrix Workspaceアプリのいずれかで H.265ビデオコーデックがサポートされていない場合、**グラフィックのH.265デコード**ポリシー設定は無視され、セッションはH.264ビデオコーデックの使用にフォールバックします。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ベースのエンコードにフォールバックします。

クライアントでH.265ハードウェアエンコードを有効にするには、H.265ビデオエンコードを参照してください。

Linux VDAは、以下のクライアントでグラフィックとビデオのハードウェアアクセラレーションにH.265をサポートしています。

- シトリックス レシーバー Windows版 4.10～4.12
- シトリックス ワークスペース™ アプリ 1808 Windows版 以降

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 用 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 用 Citrix Workspace アプリ 2303 以降
- Apple M1 チップを搭載した Mac 用 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、以下の手順を完了します。

1. **ビデオコーデックにハードウェアエンコードを使用**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**ポリシーを**画面全体**に設定します。
3. **視覚品質**ポリシーを**常に損失**または**ロスレスに構築**に設定します。

視覚的にロスレスな圧縮を許可する

視覚的にロスレス圧縮を許可ポリシーは、グラフィックの真のロスレス圧縮の代わりに**視覚的に**ロスレス圧縮を使用することを許可します。視覚的にロスレスは、真のロスレスよりもパフォーマンスを向上させますが、目には見えないわずかな損失があります。この設定は、**視覚品質**設定の値の使用方法を変更します。

視覚的にロスレス圧縮を許可ポリシーはデフォルトで無効になっています。**視覚的に**ロスレス圧縮を有効にするには、**視覚的にロスレス圧縮を許可**を**有効**に設定し、**視覚品質**ポリシーを**ロスレスに構築**に設定します。

圧縮にビデオコーデックを使用ポリシーが**ビデオコーデックを使用しない**に設定されている場合、**視覚的に**ロスレス圧縮は静止画エンコードに適用されます。**圧縮にビデオコーデックを使用**ポリシーが**ビデオコーデックを使用しない**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**ロスレス圧縮は H.264 エンコードに適用されます。

視覚品質および**圧縮にビデオコーデックを使用**ポリシー設定の詳細については、視覚表示ポリシー設定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を含めました。このスライダーは、画質とインタラクティブ性の適切なバランスを見つけるのに役立ちます。

スライダーを使用するには、以下の手順を完了します。

1. Citrix Studioで**グラフィックステータスインジケーター**ポリシーを有効にします。

   グラフィックステータスインジケーターポリシー(/en-us/linux-virtual-delivery-agent/2308/media/graphic-status-indicator-policy.png)
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:**

- **ビジュアル品質**ポリシーを**常にロスレス**または**ロスレスにビルド**に設定している場合、スライダーUIは表示されません。
- 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。

グラフィック品質スライダー(/en-us/linux-virtual-delivery-agent/2308/media/graphic-quality-slider.png)

次の選択肢が利用可能です。

- 画像の品質を変更するには、スライダーを動かします。スライダーは0〜9の範囲をサポートしています。
- システム定義の設定を使用するには、**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ドを強化します。

HDX 3D Proハードウェアエンコードが使用されている場合、VDAはネットワークの帯域幅を断続的に推定し、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ことに加えて、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な状態のままになります。シャープ化サポートは、最新のフレームを最高品質で再構成して即座に送信することで、この問題に対処できます。

Linux VDA Thinwireでサポートされているポリシーの完全なリストについては、「ポリシーサポートリスト」を参照してください。

Linux VDAでのマルチモニターサポートの構成については、「CTX220128」を参照してください。

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、その分、CPU全体の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。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`

H.264が使用されていることを確認する

H.264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（**0**は使用されていないことを意味し、**1**は使用中であることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

例えば、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
```

H.265が使用されていることを確認する

全画面H.265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（**0**は使用されていないことを意味し、**1**は使用中であることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

例えば、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
```

使用中のYUVエンコーディングスキームを確認する

次のコマンドを実行して、使用中のYUVエンコーディングスキームを確認します（**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

次のコマンドを実行して、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

次のコマンドを実行して、HDX 3D Proが有効になっていることを確認します。

```
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エラーログの確認

Xorgのログファイルは、**/var/log/** フォルダーに **Xorg.{DISPLAY}.log** のような名前で保存されています。

既知の問題と制限事項

vGPUの場合、Citrix Hypervisor™のローカルコンソールにICAデスクトップセッション画面が表示される

回避策: 次のコマンドを実行して、VMのローカルVGAコンソールを無効にします。

Citrix Hypervisor 8.1以降の場合：

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

Citrix Hypervisor 8.1より前のバージョン：

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
```

ログオン時にGnome 3デスクトップのポップアップが遅い

これはGnome 3デスクトップセッションの起動における制限です。

Citrix Workspaceアプリのウィンドウのサイズを変更すると、一部のOpenGL/WebGLアプリケーションが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するため、アプリケーションがそれに応じて応答する必要がある場合があります。たとえば、WebGLライブラリ要素の**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。

次の例では、画面を共有する手順と、他のユーザーの画面を表示する手順について説明します。

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/2308/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/2308/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。

   接続コードをコピーします(/en-us/linux-virtual-delivery-agent/2308/media/copy-the-connection-code.png)

   > **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、共有が進行中であることを示します。

3. 画面を共有したい他の仮想デスクトップのセッションユーザーと、コピーしたコードを共有します。
4. ビューアーに画面を制御させるには、**制御を許可**を選択し、ビューアーの名前を選択します。制御を停止するには、ビューアーの名前の選択を解除します。

   制御を許可(/en-us/linux-virtual-delivery-agent/2308/media/give-control.png)
5. 画面共有を停止するには、**自分の画面の共有を停止**を選択します。

   「画面共有を停止」メニューを選択します。(/en-us/linux-virtual-delivery-agent/2308/media/select-the-stop-sharing-my-screen-menu.png)

他のユーザーの画面を表示するには:

1. 仮想デスクトップの通知領域で、**画面共有**アイコンをクリックし、**他のユーザーの画面を表示**を選択します。

   「他のユーザーの画面を表示」メニューを選択します。(/en-us/linux-virtual-delivery-agent/2308/media/select-the-view-someone-else-s-screen-menu.png)
2. 表示したい画面の接続コードを入力し、**接続**をクリックします。

接続コードを入力します。(/en-us/linux-virtual-delivery-agent/2308/media/enter-the-connection-code.png)

3. 画面共有者がリクエストを承認するまで待ちます。例:

   画面共有を承認または拒否します。(/en-us/linux-virtual-delivery-agent/2308/media/accept-or-deny-screen-sharing.png)

   > **ヒント:**
   >
   > - 共有者側では、Linuxシステムがリクエストの通知を発行します。
   > - 共有者が30秒以内にリクエストを承認しない場合、リクエストは期限切れとなり、プロンプトが表示されます。

4. 画面共有者が**OK**をクリックしてリクエストを承認すると、共有画面がDesktop Viewerに表示されます。自動的に割り当てられたユーザー名でビューアーとして接続されます。

   共有画面を表示します。(/en-us/linux-virtual-delivery-agent/2308/media/view-a-shared-screen.png)

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となります。
   > - 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。

   共有画面の制御を解除する(/en-us/linux-virtual-delivery-agent/2308/media/release-control-over-shared-screen.png)

6. ディスプレイ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、マウスアイコンの横にあるアイコンをクリックします。

   ウィンドウのスケーリング(/en-us/linux-virtual-delivery-agent/2308/media/window-scaling.png)

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、次の設定を完了します。

1. Citrix Studioでグラフィックステータスインジケーターポリシーを有効にします。
2. 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。
5. (オプション) RHEL 8.x、Debian 11、またはSUSE 15.xをGNOMEとともに使用している場合は、AppIndicatorサポートを有効にするために、GNOMEシェル用の互換性のある拡張機能をインストールします。

   a. `gnome-shell --version`コマンドを実行して、GNOMEシェルバージョンを確認します。

   b. `<https://extensions.gnome.org/extension/615/appindicator-support>`からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能バージョンとして24または26を選択できます。

   GNOMEシェル拡張機能(/en-us/linux-virtual-delivery-agent/2308/media/gnome-extension.png)

   c. ダウンロードしたパッケージを解凍します。パッケージ内の**metadata.json**ファイルの**「uuid」**値が`appindicatorsupport@rgcjonas.gmail.com`に設定されていることを確認します。

   d. `mv`コマンドを実行して、`appindicatorsupport@rgcjonas.gmail.com`ディレクトリを`/usr/share/gnome-shell/extensions/`の下の場所に移動します。

   e. `chmod a+r metadata.json`コマンドを実行して、**metadata.json**ファイルを他のユーザーが読み取れるようにします。

   > **ヒント:**

> デフォルトでは、`appindicatorsupport@rgcjonas.gmail.com`ディレクトリにある**metadata.json**ファイルはroot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`キーを順番に押すか、`killall -SIGQUIT gnome-shell`コマンドを実行してGNOMEシェルをリロードします。

h. デスクトップ環境でGNOME Tweaksを実行し、Tweaksツールで**KStatusNotifierItem/AppIndicator Support**を有効にします。

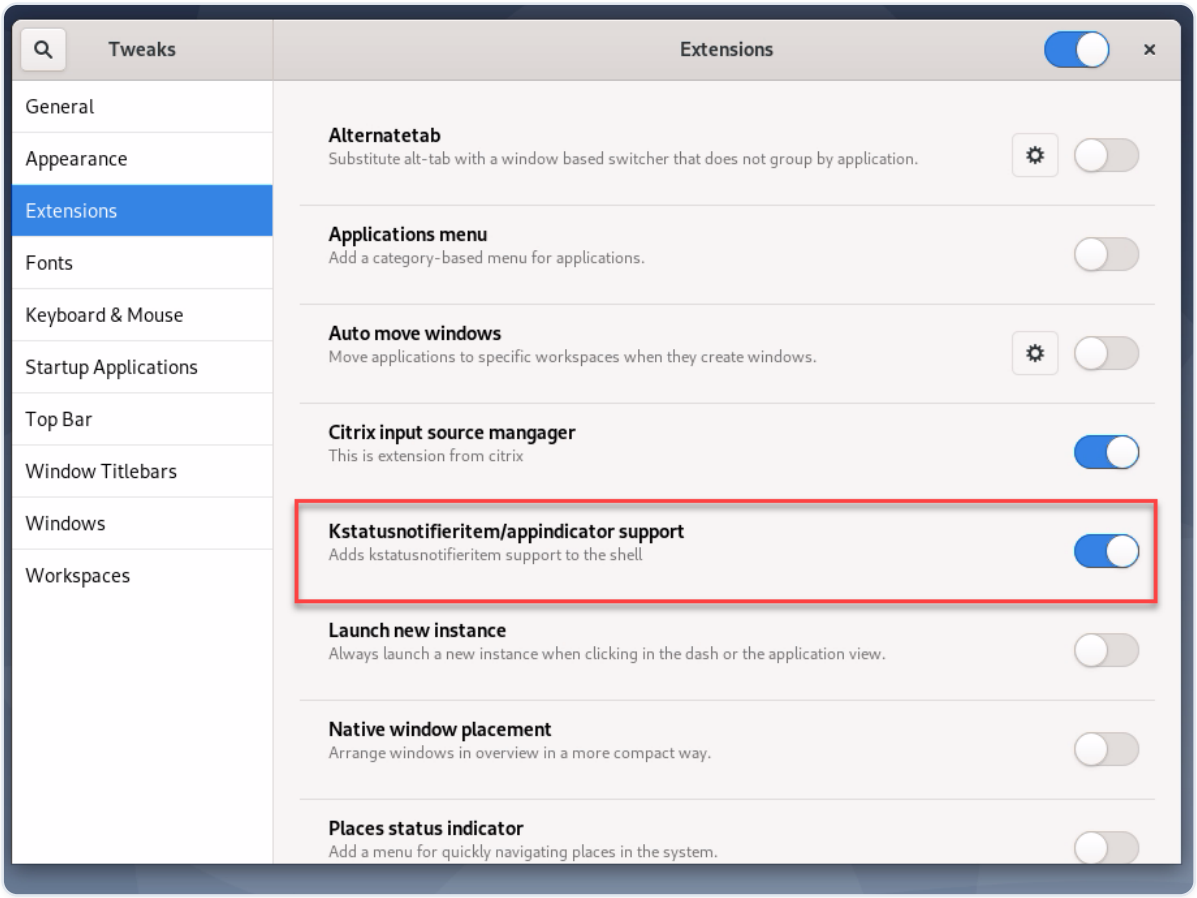
6. (オプション) GNOMEがインストールされたDebian 11.3を使用している場合は、GNOMEシステムトレイアイコンをインストールして有効にするには、次の手順を実行します。

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。



考慮事項

- 画面共有機能はH.265ビデオコーデックをサポートしていません。
- 画面共有機能はアプリセッションでは利用できません。
- デスクトップセッションのユーザーは、デフォルトで最大10人の視聴者とセッション画面を共有できます。最大視聴者数は`ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v "ScreenSharingViewerMaxNum" -d <hex_value>`で構成可能です。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。

マルチモニターのサポート

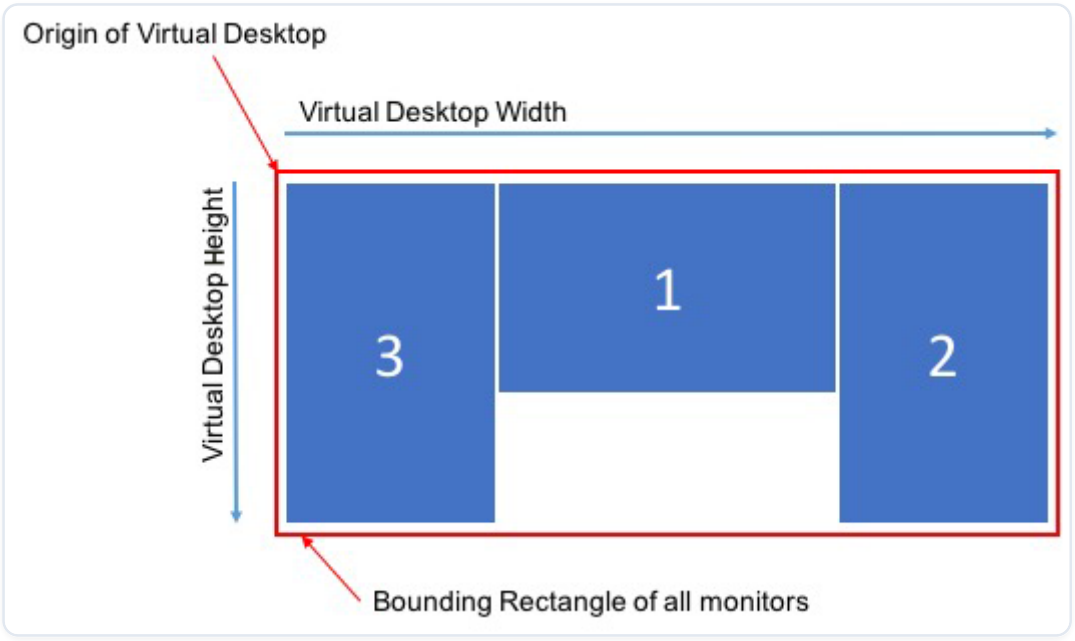
概要

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、さまざまな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外接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いているわけではありません。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外接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、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外接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外接矩形の下左隅までの垂直距離（ピクセル単位）です。

この計算は、次の理由から重要です。

- さまざまな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

さまざまな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柔軟なLinux VDAを構成できます。

次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/2308/media/client-monitor-configuration.png)

上の図は、2台のモニター（それぞれ解像度2560×1600）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/2308/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限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モニターを使用する構成の場合、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モニターを1台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。

モニターは1台のみ(/en-us/linux-virtual-delivery-agent/2308/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。詳しくは、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用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なので、32ビットカラーの場合は4を使用します。

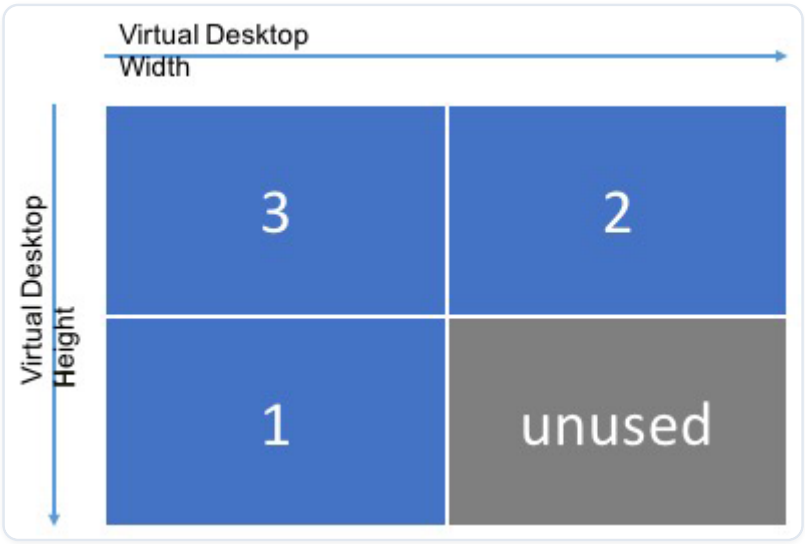
> **注：**
>
> Xサーバーのカラー深度はセッションの開始時に決定され、セッションの存続期間中（**ログインから切断/再接続を経てログオフまで**）変更できません。そのため、Linux VDAは常に仮想セッションデスクトップを32ビットとして割り当て、セッションで要求されたカラー深度にダウンサンプリングします。

> 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 メガバイト = 3 メガバイト

> 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は5120×3200である必要があります。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: 2

  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

  最大値: 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。

次の方法で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
```

`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。

次の方法で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
```

`MaxWidth` は、`1,024` と `16,384` の間の値です。

次の方法で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
```

`MaxHeight` は、`1,024` と `16,384` の間の値です。

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。

- 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。追加の構成は不要です。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：インストールを完了するためのランタイム環境のセットアップ」を参照してください。

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

以下の4つのテンプレート構成ファイルは、**/etc/X11**の下にあります。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、名前に番号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、名前に番号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. Linux VDAが手順1で設定された構成ファイル名を認識できるように、次のコマンドを実行してレジストリキーを設定します。

```
/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およびDebian 11.3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つの手順を完了します。

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、グラフィック出力の欠落または文字化けはバグの結果です。デバッグ用のトレース情報を収集するには、/opt/Citrix/VDA/bin/setlogを使用し、GFX_X11をverboseに設定します。

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、テキストの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を保証するものではありません。代わりに、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IDであり、そのユーザーのログオン資格情報によって追跡されます。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、セッションディスプレイには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

デフォルトでは値は空です。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>`	透かしのレイアウトスタイル	`xstyle`、`single`、`tile`、`horizontal`	`xstyle`	-
`<position>`	透かしの位置	`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>`	透かしの特定の角度への回転	-180〜180	0	-
`<transparency>`	透かしの不透明度	0〜100	17	-
`<font>`	-	システムでサポートされているフォント	サンズ	-
`<fontsize>`	-	20〜50	0 (自動計算)	-
`<fontzoom>`	`<fontsize>` および `<image>` で設定したフォントと画像サイズの割合	0 –	100	-
`<image>`	PNG透かし	VDA上のPNG画像へのパス	該当なし	この構文はPNG透かしを設定します。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透かしを使用している場

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
				合、`<style>`、`<position>`、`<rotation>`、`<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	-

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
`<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	-
`<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:

透かしのカスタムテキストが有効な構文設定で指定されている場合、**セッション透かしを有効にする**を除く、他のすべてのセッション透かしポリシーは無視されます。

構文オプションを指定しない場合、またはサポートされていない値に設定した場合、それらのデフォルト値が使用されます。

制限事項

- セッションウォーターマークは、次のいずれかのケースでサポートされます。
  - **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**推奨される場合に使用**に設定されており、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。
- セッションウォーターマークは、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サポートされません。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、セッションが全画面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は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るのを防ぐための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。

ステップ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てください。

ステップ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。

   - 「VirtualGLで使用するサーバーを構成する（GLX + EGLバックエンド）」には「1」を選択します。
   - 「3D Xサーバーへのアクセスを**vglusers**グループに制限する」には「n」を選択します。

- 「フレームバッファーデバイスへのアクセスを**vglusers**グループに制限する」には「n」を選択します。
- 「XTEST拡張機能を無効にする」には「n」を選択します。

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: OpenGL 3DアプリケーションをGPUアクセラレーションで実行する

Linux VDAセッションでOpenGL 3DアプリケーションをGPUアクセラレーションで実行するには、次の2つの方法があります。

- **方法1:** すべてのOpenGL 3Dアプリケーションに対して共有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アプリケーションに対して共有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アプリケーションに対して共有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上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ために、共有GPUアクセラレーションにはLightDMをディスプレイマネージャーとして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上のFirefoxでのみサポートされています。

スケーラビリティ

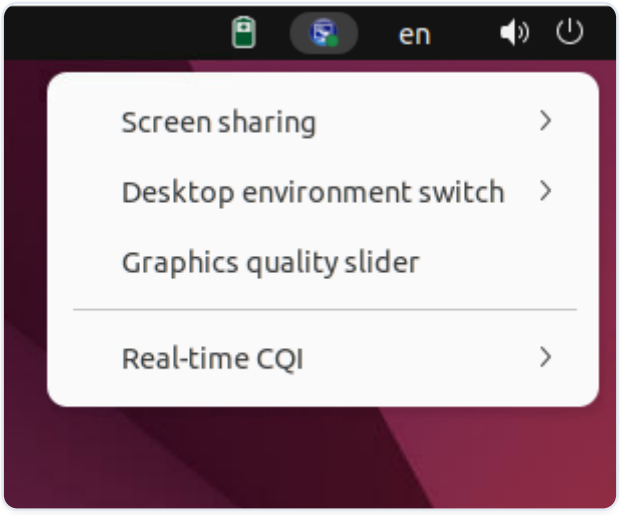
GPUを共有できる同時セッションの最大サポート数は、CPUとシステムメモリに依存します。また、GPUの最大ビデオメモリにも大きく依存します。

例:

条件	結果
NVIDIA **M10-2B** vGPUのビデオメモリが2,048 MBで、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ワークロードに100 MBのビデオメモリを使用する場合、,	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。

システムトレイ

セッションユーザーは、以下のシステムトレイアイコンをクリックして、対応するアクションを実行するためのメニュー項目にアクセスできます。



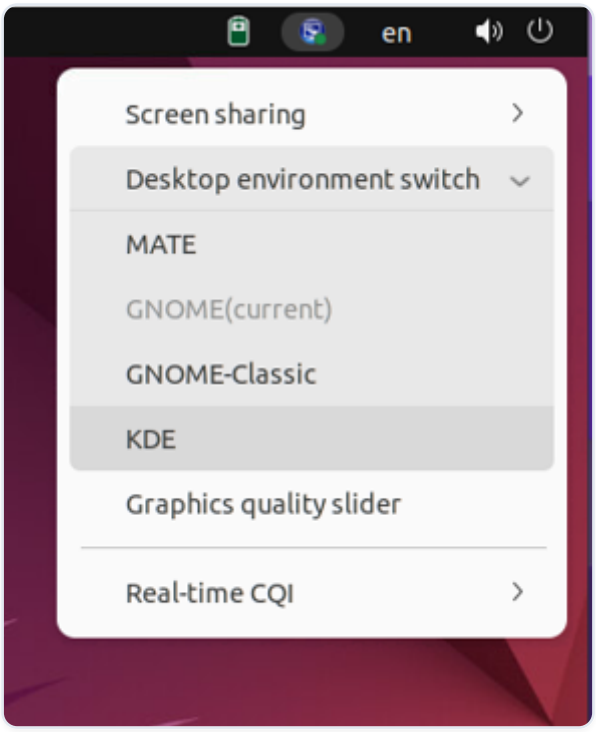
各メニュー項目は、トグル付きの機能に対応しています。メニュー項目に対応する機能が無効になっている場合、そのメニュー項目は非表示になり、表示されません。

- **画面共有**

  この機能の詳細については、「」を参照してください。(/ja-jp/linux-virtual-delivery-agent/2308/configure/graphics/hdx-screen-sharing.html).

- **デスクトップ環境の切り替え**

  このメニュー項目は、**ctxdesktopswitch.sh** のGUIです。詳細については、「」を参照してください。(/ja-jp/linux-virtual-delivery-agent/2308/configure/session/custom-desktop-environments-by-session-users.html).



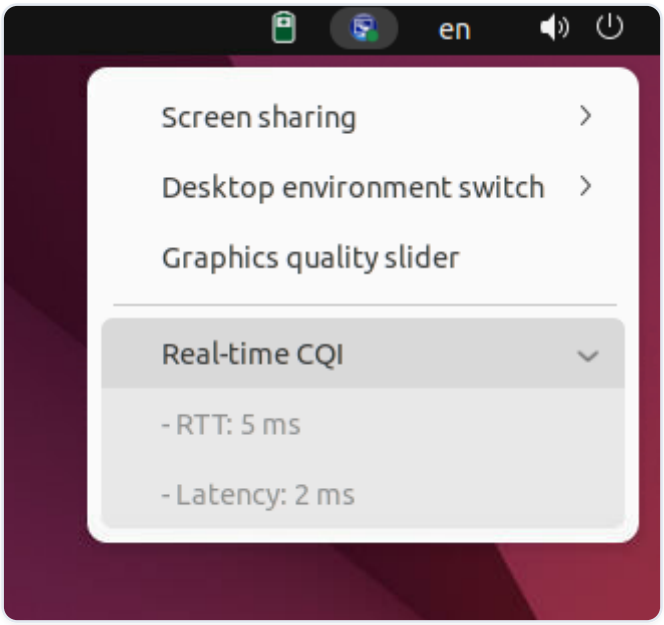
セッションユーザーによるデスクトップ環境のカスタマイズは、デフォルトで有効になっています。これを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "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「」セクションを参照してください。(/ja-jp/linux-virtual-delivery-agent/2308/configure/graphics/configuring-graphics.html#graphics-quality-slider).

- **リアルタイムCQI**



現在、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、「」を参照してください。(/ja-jp/linux-virtual-delivery-agent/2308/configure/administration/tools-and-utilities.html#session-data-query-utility).

リアルタイムCQIステータスに応じて、システムトレイアイコンの表示が異なります。



アイコンの表示が変化す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 "HighRttThreshold" -d "0x00000104" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "LowLatencyThreshold" -d "0x00000078" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "LowRttThreshold" -d "0x00000096" --force
```

実際のRTTが**LowRttThreshold**以下で、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RTTが**HighRttThreshold**より大きい場合、または実際の遅延が**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
```

Thinwire プログレッシブ表示

セッションの対話性は、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可能性があります。

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 ms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、セッションの対話性が向上します。

Thinwireプログレッシブ表示1の画像(/en-us/linux-virtual-delivery-agent/2308/media/thinwireprogressivedisplay.png)

セッションとの対話を停止すると、劣化した画像とテキストはプログレッシブにシャープ化され、ロスレスになります。たとえば、以下のグラフィックでは、文字に青いアーティファクトが含まれなくなり、画像はソース品質で表示されます。

Thinwireプログレッシブ表示2の画像(/en-us/linux-virtual-delivery-agent/2308/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Selective H.264で引き続き管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは**視覚品質**ポリシー設定の**高**、**中**（デフォルト）、および**低**で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したときに有効になります。

- 利用可能な帯域幅が2 Mbpsを下回る
- ネットワーク遅延が200 msを超える

モード切り替えが発生した後、たとえネットワークの状態が悪く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 \<threshold in ms\> です (デフォルト = 200)

例: 100 = ネットワーク遅延が100ミリ秒を下回った場合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

もし	その場合
ホスト (VDA) 上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** 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) 上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) 上でそれを構成すると、,	ユーザーによって指定され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可能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) でそれを構成する場合、,	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ミリ秒未満の場合に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しくみ

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム /dev パスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで使用できない場合があります。USBデバイスは正常に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスできません。USBデバイスの接続を確実にするには、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:

Amazon Linux 2、CentOS、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構成

(CentOS、RHEL、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。ただし、USB/IP カーネルモジュールは CentOS、RHEL、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際は、以下のポリシーとルールを構成してください。

- 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を有効にする

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/2308/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。

クライアントUSBデバイスリダイレクトルールダイアログで、次の操作を行います。

1. **新規**をクリックしてリダイレクトルールを追加するか、**編集**をクリックして既存のルールを確認します。
2. ルールを作成（または編集）した後、**OK**をクリックします。

クライアントUSBデバイスリダイレクトルールの設定(/en-us/linux-virtual-delivery-agent/2308/media/client-usb-redirection-rule.png)

USBデバイスリダイレクトの問題をトラブルシューティングする

このセクションの情報を使用して、Linux VDAの使用中に発生する可能性のあるさまざまな問題をトラブルシューティングします。

リダイレクトされたUSBディスクをアンマウントできない

Linux VDAは、Citrix Workspaceアプリからリダイレクトされたすべてのユーザーディスクを管理者権限で管理し、所有者のみがリダイレクトされたデバイスにアクセスできるようにします。そのため、デバイスをアンマウントするには管理者権限が必要です。

デバイスをアンマウントできない(/en-us/linux-virtual-delivery-agent/2308/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

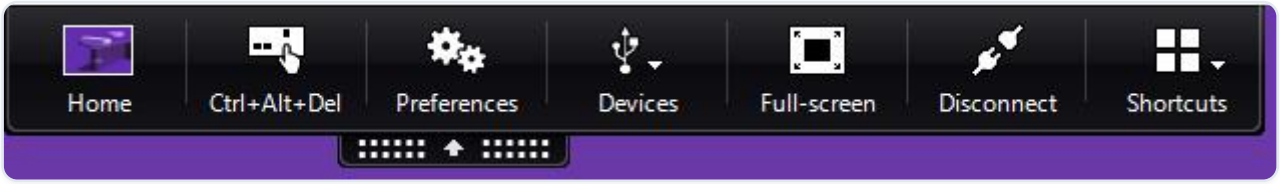
Citrix Workspaceアプリのツールバーを使用してUSBディスクのリダイレクトをすぐに停止すると、ディスク上で変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込む際に、システムがファイルシステム内のメモリキャッシュをマウ

ントするため、データがディスク自体に書き込まれないために発生します。Citrix Workspaceアプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。

この問題を解決するには、USBリダイレクトを停止する前に、ターミナルで**sync**コマンドを使用してデータをディスクにフラッシュします。

Citrix Workspaceアプリのツールバーにデバイスが表示されない

Citrix Workspaceアプリのツールバーにデバイスが一覧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。

- ポリシーがUSBデバイスリダイレクトを許可するように構成されている。
- Citrix USBサービスモジュールが実行されている。

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.
   ```

   - Amazon Linux 2、CentOS、RHEL、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、USB/IPカーネルモジュールをコンパイルしてインストールするために、次の手順を実行します。

  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>`にあるLinuxカーネルリポジトリに移動し、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。

```
service ctxusbsd restart
```

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、リダイレクトできるのは サポートされている USB デバイス のみです。他のデバイスもアクティブな Linux VDA セッションにリダイレクトされる場合があります。リダイレクトされたデバイスごとに、ユーザーが所有するノードがシステム */dev* パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている（接続されている）がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限のポリシーに追加します。

> **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（およびインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。これは、サポートされているデバイスリストにないデバイスには当てはまらない場合があります。

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アンドペーストできます。

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

この設定により、クライアントデバイス上のクリップボードをVDA上のクリップボードにマッピングす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アンド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アンドペーストできます。

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

この設定は、単一のコピーアンド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。

クリップボード転送サイズを制限するには、**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**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

この設定は、単一のコピーアンド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。

クリップボード転送サイズを制限するには、**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**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。

クライアントクリップボード書き込みの制限 AND クライアントクリップボード書き込み許可形式

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アンドペースト（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書き込みの制限 および セッションクリップボード書き込み許可形式

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へコピー＆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、次のトピックが含まれています。

- クライアントIME(/ja-jp/linux-virtual-delivery-agent/2308/configure/keyboard/client-ime.html)
- クライアントIMEユーザーインターフェイスの同期(/ja-jp/linux-virtual-delivery-agent/2308/configure/keyboard/client-ime-user-interface-synchronization.html)

- 動的キーボードレイアウトの同期(/ja-jp/linux-virtual-delivery-agent/2308/configure/keyboard/dynamic-keyboard-layout-synchronization.html)
- ソフトキーボード(/ja-jp/linux-virtual-delivery-agent/2308/configure/keyboard/soft-keyboard.html)
- 複数言語入力のサポート(/ja-jp/linux-virtual-delivery-agent/2308/configure/keyboard/configure-server-ime.html)

クライアント入力メソッドエディター (IME)

概要

中国語、日本語、韓国語などの2バイト文字は、IMEを介して入力する必要があります。これらの文字は、WindowsネイティブのCJK IMEなど、クライアント側でCitrix Workspace™アプリと互換性のある任意のIMEを使用して入力してください。

インストール

この機能は、Linux VDAをインストールすると自動的にインストールされます。

使用方法

通常どおりCitrix Virtual AppsまたはCitrix Virtual Desktops™セッションを開きます。

クライアントIME機能の使用を開始するには、クライアント側で必要に応じて入力方法を変更します。

既知の問題

- Googleスプレッドシートのセルに文字を入力するためにクライアントIME機能を使用する前に、そのセルをダブルクリックする必要があります。
- クライアントIME機能は、パスワードフィールドでは自動的に無効になりません。
- IMEユーザーインターフェイスは、入力領域でカーソルに追従しません。

クライアントIMEユーザーインターフェース同期

概要

これまで、クライアントIMEユーザーインターフェース（変換ウィンドウや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ースの画像(/en-us/linux-virtual-delivery-agent/2308/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ースの画像(/en-us/linux-virtual-delivery-agent/2308/media/synced-composition-and-candidate-windows.png)

この機能を使用するための前提条件

1. Linux VDAでIntelligent Input Bus (`IBus`)を有効にします。Linux OSで `IBus` を有効にする方法については、OSベンダーのドキュメントを参照してください。例：
   - ウブントゥ: `<https://help.ubuntu.com/community/ibus>`
   - CentOS、RHEL: `<https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods>`
   - デビアン: `<https://wiki.debian.org/I18n/ibus>`
   - SUSE: `<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-gnome-settings.html#sec-gnome-settings-lang>`
2. この機能は自動的にインストールされますが、使用する前に有効にする必要があります。

機能の有効化と無効化

クライアントIMEユーザーインターフェース同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、 `ctxreg` ユーティリティを介してレジストリを編集します。

> **注:**
>
> **Client Keyboard Layout Sync and IME Improvement**ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- クライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、**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/2308/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/2308/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。

- クライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、`ctxreg`ユーティリティを使用してレジストリを編集します。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v  "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。例えば、クライアントデバイスでキーボードレイアウトが英語からフランス語に変更されたにもかかわらず、VDAでは変更されなかった場合、キーマッピングの問題が発生する可能性がありました。

Citrix®は、VDAのキーボードレイアウトをクライアントデバイスのキーボードレイアウトと自動的に同期させることで、この問題に対処します。クライアントデバイスのキーボードレイアウトが変更されると、VDAのレイアウトもそれに合わせて変更されます。

> **注:**
>
> Citrix Workspace™ app for HTML5は、動的キーボードレイアウト同期機能をサポートしていません。

構成

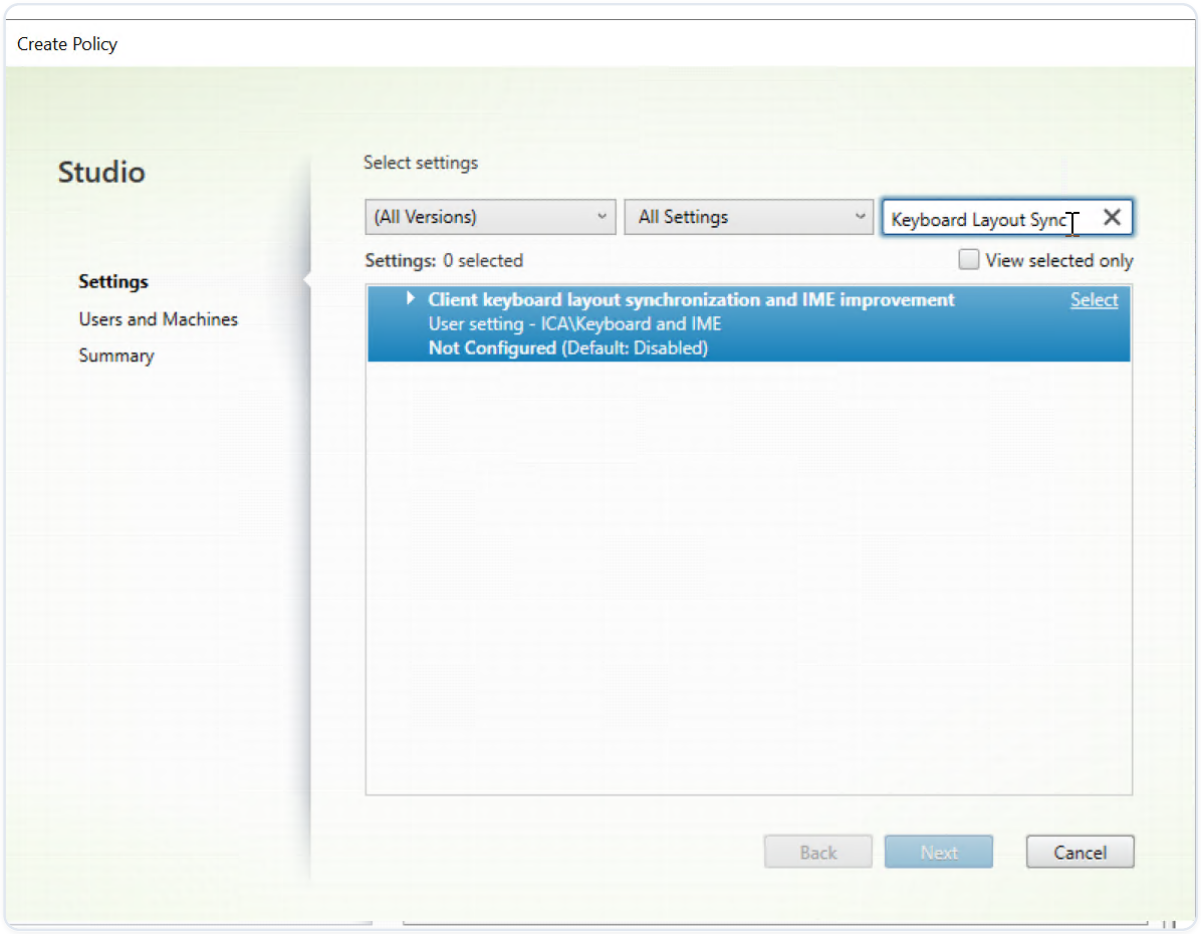
動的キーボードレイアウト同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、`ctxreg`ユーティリティを介してレジストリを編集します。

> **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。

- 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。

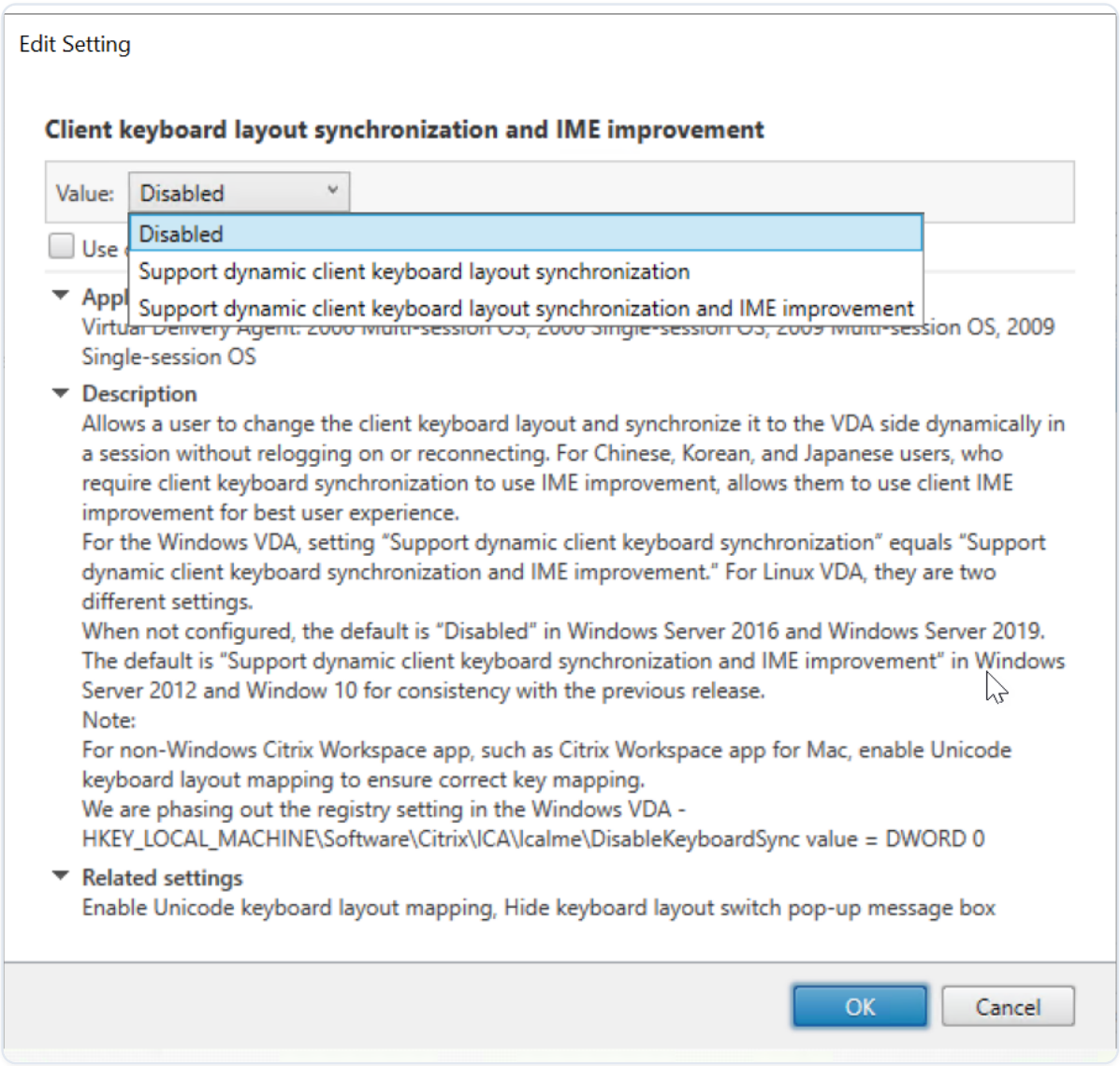
  a. Studioで、**Policies**を右クリックし、**Create Policy**を選択します。

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つのオプションがあります。

- **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。

- `ctxreg`ユーティリティを介してレジストリを編集し、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します。

  この機能を有効にするには、次のコマンドを実行します:

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します:

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000000"
  ```

使用方法

この機能が有効になっている場合、セッション中にクライアントデバイスのキーボードレイアウトが変更されると、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語 (FR) に変更すると、次のようになります:

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/2308/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:

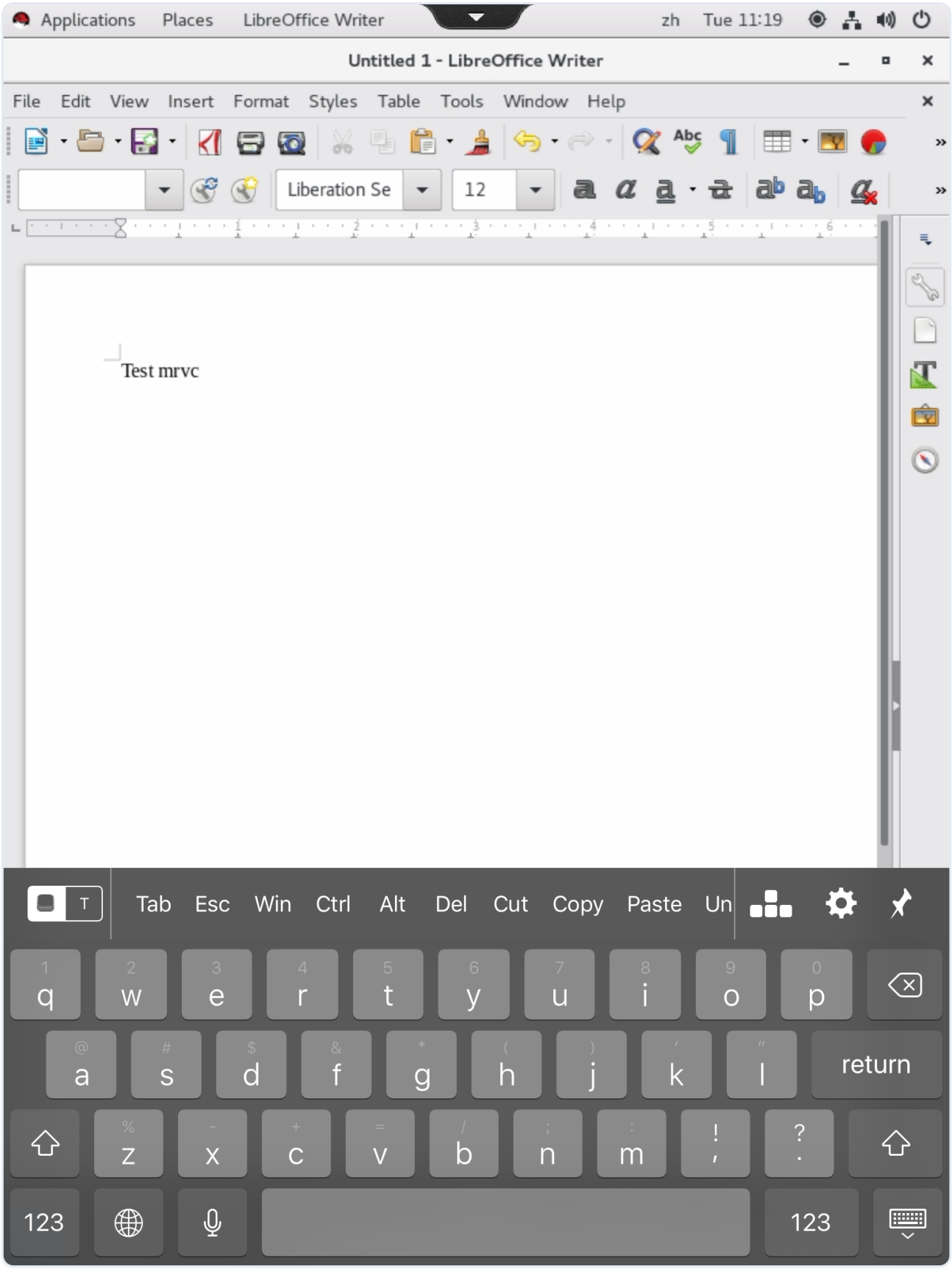
セッションでキーボードがフランス語に変更されている画像(/en-us/linux-virtual-delivery-agent/2308/media/changed-to-fr.png)

デスクトップセッションでは、この自動変更をタスクバーで確認できます:

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/2308/media/fr-in-the-task-bar.png)

ソフトキーボード

ソフトキーボード機能は、Linux仮想デスクトップまたはアプリケーションセッションで利用できます。ソフトキーボードは、入力フィールドに入ると自動的に表示され、入力フィールドを離れると自動的に非表示になります。


Applications
Places
LibreOffice Writer
zh
Tue 11:19
Untitled 1 - LibreOffice Writer
File
Edit
View
Insert
Format
Styles
Table
Tools
Window
Help
Liberation Se
12
Test mrvc
Tab
Esc
Win
Ctrl
Alt
Del
Cut
Copy
Paste
Un
return
123
123

注:

この機能は、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. 次のコマンドを実行します。

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

2. Citrix Studioで、**Automatic keyboard display**ポリシーを**Allowed**に設定します。
3. (オプション) RHEL 7およびCentOS 7の場合、次のコマンドを実行してIntelligent Input Bus (`IBus`)をデフォルトのIMサービスとして構成します。

```
echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

注:

上記の設定は、新しいセッションにログオンするか、現在のセッションからログオフして再度ログオンしたときに有効になります。

制限事項

- この機能は、Google Chrome、LibreOffice、およびその他のアプリでは期待どおりに動作しない場合があります。

- ソフトキーボードを手動で非表示にした後、再度表示するには、入力フィールド以外の場所をクリックしてから、現在の入力フィールドを再度クリックします。
- Webブラウザーで入力フィールドから別の入力フィールドにクリックしても、ソフトキーボードが表示されない場合があります。この問題を回避するには、入力フィールド以外の場所をクリックしてから、目的の入力フィールドをクリックします。
- この機能は、Unicode文字および2バイト文字（中国語、日本語、韓国語などの文字）をサポートしていません。
- パスワード入力フィールドでは、ソフトキーボードは利用できません。
- ソフトキーボードが現在の入力フィールドと重なる場合があります。この場合、アプリのウィンドウを移動するか、画面をスクロールして、入力フィールドにアクセスしやすい位置に移動してください。
- Citrix WorkspaceアプリとHuaweiタブレット間の互換性の問題により、物理キーボードが接続されている場合でも、Huaweiタブレットではソフトキーボードが表示されます。

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上のネイティブ言語バーは、Linuxデスクトップ環境と高度に統合されているため、公開アプリケーションでは利用できませんでした。その結果、ユーザーは中国語、日本語、韓国語などのIMEを必要とする言語でテキストを入力できませんでした。また、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け入れる公開アプリケーション用の言語バーを提供します。言語バーを使用すると、ユーザーはサーバー側のIMEを選択し、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

ctxregユーティリティを使用して、この機能を有効または無効にできます（デフォルトでは無効）。特定のLinux VDAサーバーでの機能構成は、そのVDAで公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \\SYSTEM\\CurrentControlSet\\Control\\Citrix\\LanguageBar」で、タイプはDWORDです。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000001"
```

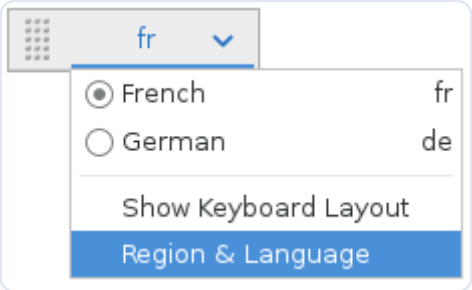
この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000000"
```

使用方法

使い方は簡単です。

1. 機能を有効にします。
2. テキスト入力を受け入れる公開アプリケーションにアクセスします。セッション中に、アプリケーションの横に言語バーが表示されます。
3. ドロップダウンメニューから、**地域と言語**を選択して、目的の言語（入力ソース）を追加します。



4. ドロップダウンメニューからIMEまたはキーボードレイアウトを選択します。
5. 選択したIMEまたはキーボードレイアウトを使用して言語を入力します。

> **注:**
>
> - VDA側の言語バーでキーボードレイアウトを変更する場合は、クライアント側（Citrix Workspace™アプリを実行中）でも同じキーボードレイアウトが使用されていることを確認してください。
> - **Region & Language**ダイアログボックスで設定を行う前に、**accountsservice**パッケージをバージョン0.6.37以降にアップグレードする必要があります。

> IMEまたはキーボードレイアウトを選択するイメージ(/en-us/linux-virtual-delivery-agent/2308/media/region-language-dialog-box.png)

マルチメディア

このセクションには、次のトピックが含まれています。

- オーディオ機能
- ブラウザコンテンツリダイレクト
- HDX Webカメラビデオ圧縮

オーディオ機能

アダプティブオーディオ

アダプティブオーディオはデフォルトで有効になっています。次のCitrix Workspace™アプリクライアントをサポートしています。

- Windows向けCitrix Workspaceアプリ – 2109以降のバージョン
- Linux向けCitrix Workspaceアプリ – 2109以降のバージョン
- Mac向けCitrix Workspaceアプリ – 2109以降のバージョン

リストにないクライアントを使用すると、アダプティブオーディオはレガシーオーディオにフォールバックします。

アダプティブオーディオを使用すると、VDAでオーディオ品質ポリシーを手動で構成する必要はありません。アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、プレミアムなオーディオエクスペリエンスを提供します。

次の表は、アダプティブオーディオとレガシーオーディオの比較を示しています。

アダプティブオーディオ	レガシーオーディオ
最大オーディオサンプルレート: 48 kHz	最大オーディオサンプルレート: 8 kHz
ステレオチャネル	モノラルチャネル

> **ヒント:**

> RHEL 8.xおよびRocky Linux 8.xでPulseAudio 13.99以降を使用してください。

ブラウザコンテンツリダイレクト

概要

ブラウザコンテンツリダイレクトは、クライアント側で許可リスト内のWebページをレンダリングする機能を提供します。この機能は、Citrix Workspace™アプリを使用して、クライアント側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コンテンツをフェッチします。

> **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。

ブラウザのビューポートのみがリダイレクトされます。ビューポートとは、ブラウザ内でコンテンツが表示される長方形の領域です。ビューポートには、アドレスバー、お気に入りバー、ステータスバーなどの項目は含まれません。これらの項目は、VDA上のブラウザで引き続き実行されます。

ビューポートの画像(/en-us/linux-virtual-delivery-agent/2308/media/viewport.png)

リダイレクト用の許可リスト内のURLを含むアクセス制御リストを指定する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。

URLが許可リスト内で見つかり、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、リダイレクトが必要であることをCitrix Workspaceアプリに指示し、URLを中継します。その後、Citrix Workspaceアプリはローカルレンダリングエンジンをインスタンス化し、Webサイトを表示します。

その後、Citrix Workspaceアプリはウェブサイトを仮想デスクトップのブラウザコンテンツ領域にシームレスにブレンドし直します。

ビューポートとクライアント(/en-us/linux-virtual-delivery-agent/2308/media/viewport-desktop-viewer.png)

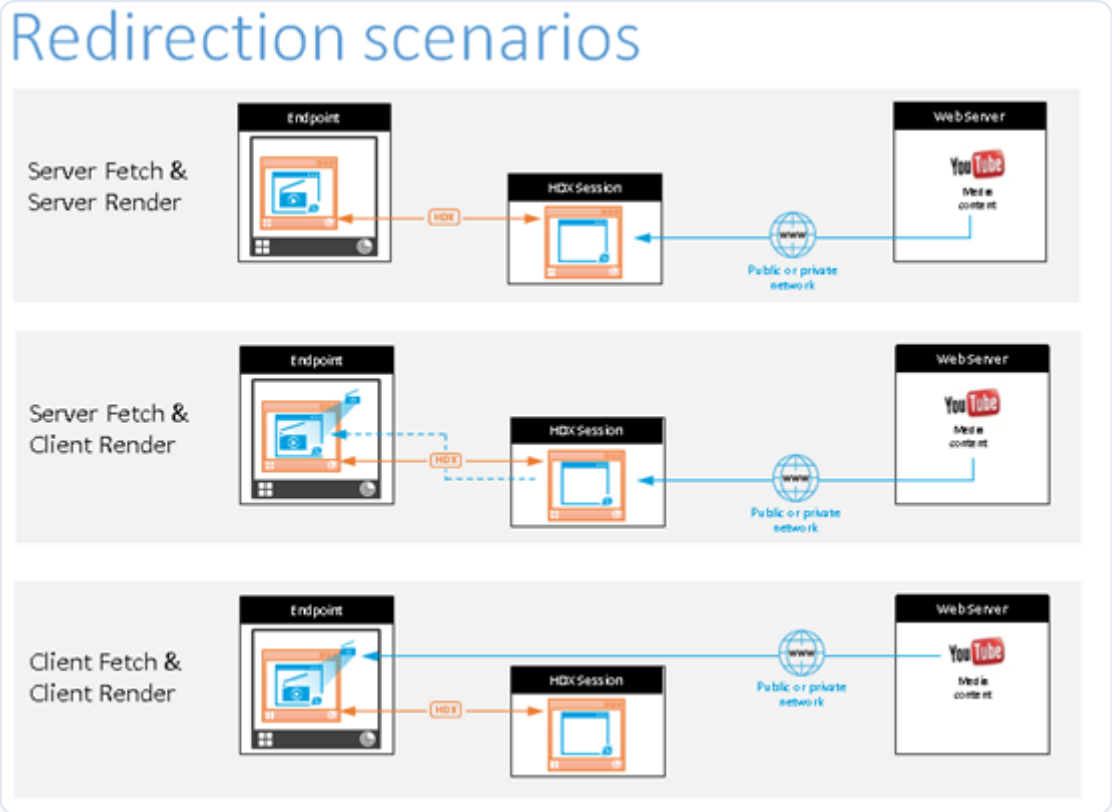
1. Citrix®ブラウザコンテンツリダイレクト拡張機能のアイコン

   拡張機能アイコンの色は、Chrome拡張機能のステータスを示します。以下の3色のいずれかです。

   - 緑: アクティブで接続済み
   - 灰色: 現在のタブで非アクティブ/アイドル状態
   - 赤: 破損/機能していません

2. クライアントでレンダリングされたビューポート、または仮想デスクトップにブレンドバック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でウェブページをレンダリングし、Thinwireを使用してグラフィックをリモート処理します。ポリシーを使用してフォールバック動作を制御します。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。

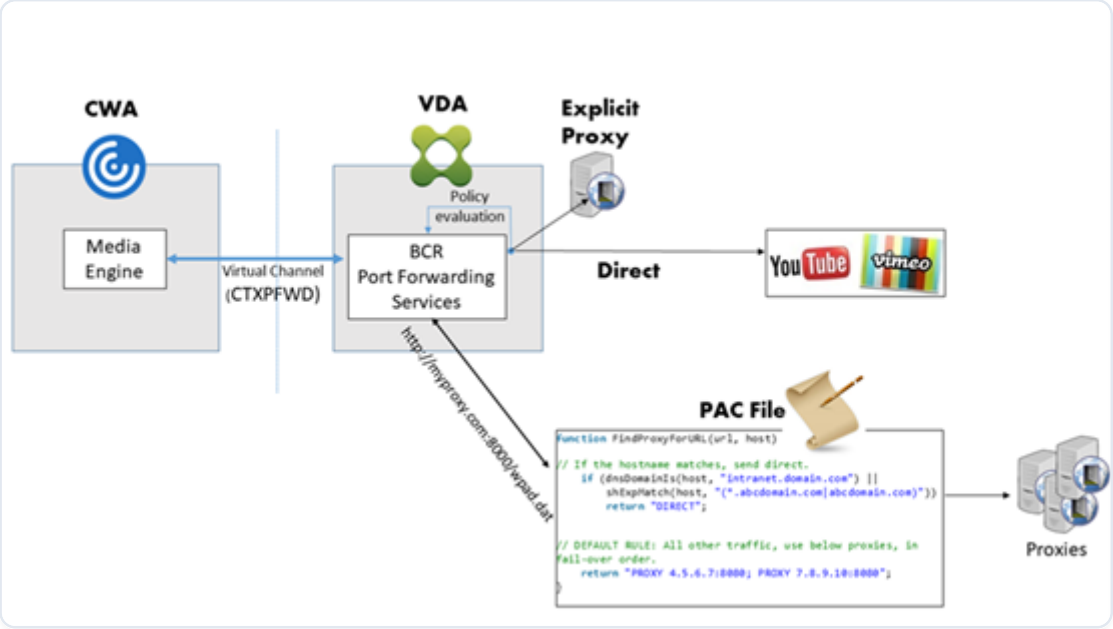
- **サーバーフェッチとクライアントレンダリング**: Citrix Workspaceアプリは、仮想チャネル（CTXPFWD）を使用してVDAを介してWebサーバーからコンテンツにアクセスし、フェッチします。このオプションは、クライアントがWebサーバーにアクセスできない場合（シンクライアントなど）に役立ちます。VDAでのCPUとRAMの消費を抑えますが、ICA®仮想チャネルで帯域幅が消費されます。

  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**Browser Content Redirection Proxy Configuration**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サーバーに直接アクセスするため、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

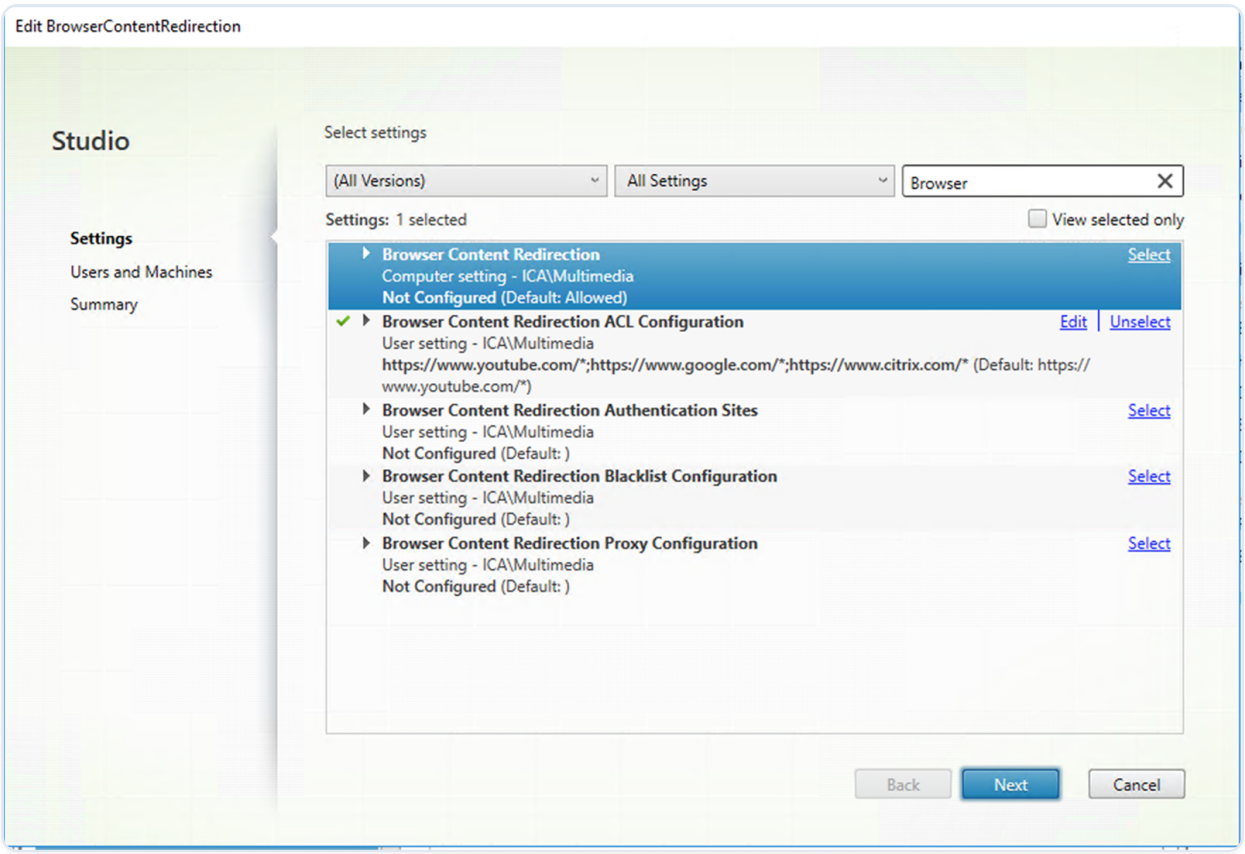
リナックス VDA:

- VDA上のブラウザ: Citrixブラウザコンテンツリダイレクト拡張機能が追加されたGoogle Chrome v66以降

ブラウザコンテンツリダイレクトの構成

ブラウザ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、ブラウザコンテンツリダイレクトを有効にするには、**Browser Content Redirection**を**Allowed**に設定します。



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。

   「**ブラウザーコンテンツリダイレクトACL構成**」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を指定するときは、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。

   ブラウザーコンテンツリダイレクトACL構成が選択されている(/en-us/linux-virtual-delivery-agent/2308/media/allow-list.png)

   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`を指定した場合、`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/`

   ブラウザーコンテンツリダイレクトの許可リストを設定する(/en-us/linux-virtual-delivery-agent/2308/media/allow-list-confiuration.png)

   「**ブラウザーコンテンツリダイレクトブラックリスト構成**」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。

   ブラウザーコンテンツリダイレクトのブロックリストを設定する(/en-us/linux-virtual-delivery-agent/2308/media/block-list.png)

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、「**ブラウザーコンテンツリダイレクトプロキシ構成**」設定を構成します。

   この設定は、ブラウザーコンテンツリダイレクト用の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、または直接/透過設定で有効になっている場合、Citrix Workspaceアプリ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、構成されておらず、デフォルト値を使用している場合、Citrix Workspaceアプリ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**禁止**されています。

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が許可されないことを意味します。
>
> 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、「ブラウザーコンテンツリダイレクト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**Chromeに追加**をクリックして、Chromeウェブストアから Citrixブラウザー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上のブラウザーが、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するのに役立ちます。

> **重要:**
>
> 拡張機能はクライアントには不要です。VDAにのみ追加してください。
>
> \>
>
> Chrome拡張機能はユーザーごとにインストールされます。拡張機能を追加または削除するためにゴールデンイメージを更新する必要はありません。
>
> ChromeウェブストアからCitrixブラウザコンテンツリダイレクト拡張機能を追加する(/en-us/linux-virtual-delivery-agent/2308/media/add-to-chrome.png)

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している場合。**PACファイル** - VDA内のブラウザが指定されたURLをフェッチするために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。
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™ Webカメラビデオ圧縮

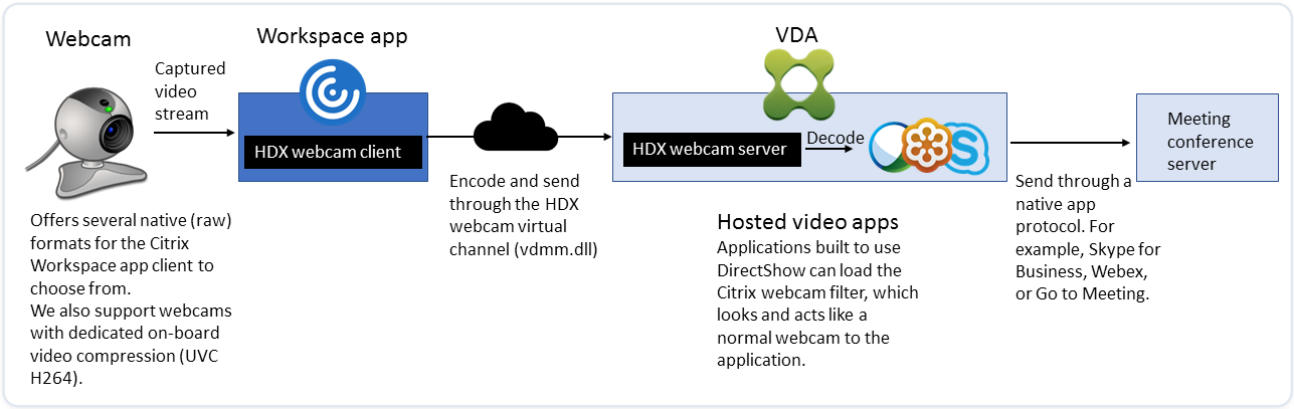
概要

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDX Webカメラビデオ圧縮を使用してWebカメラを使用できるようになりました。この機能はデフォルトで

有効になっています。可能な場合は常にHDX Webカメラビデオ圧縮を使用することをお勧めします。

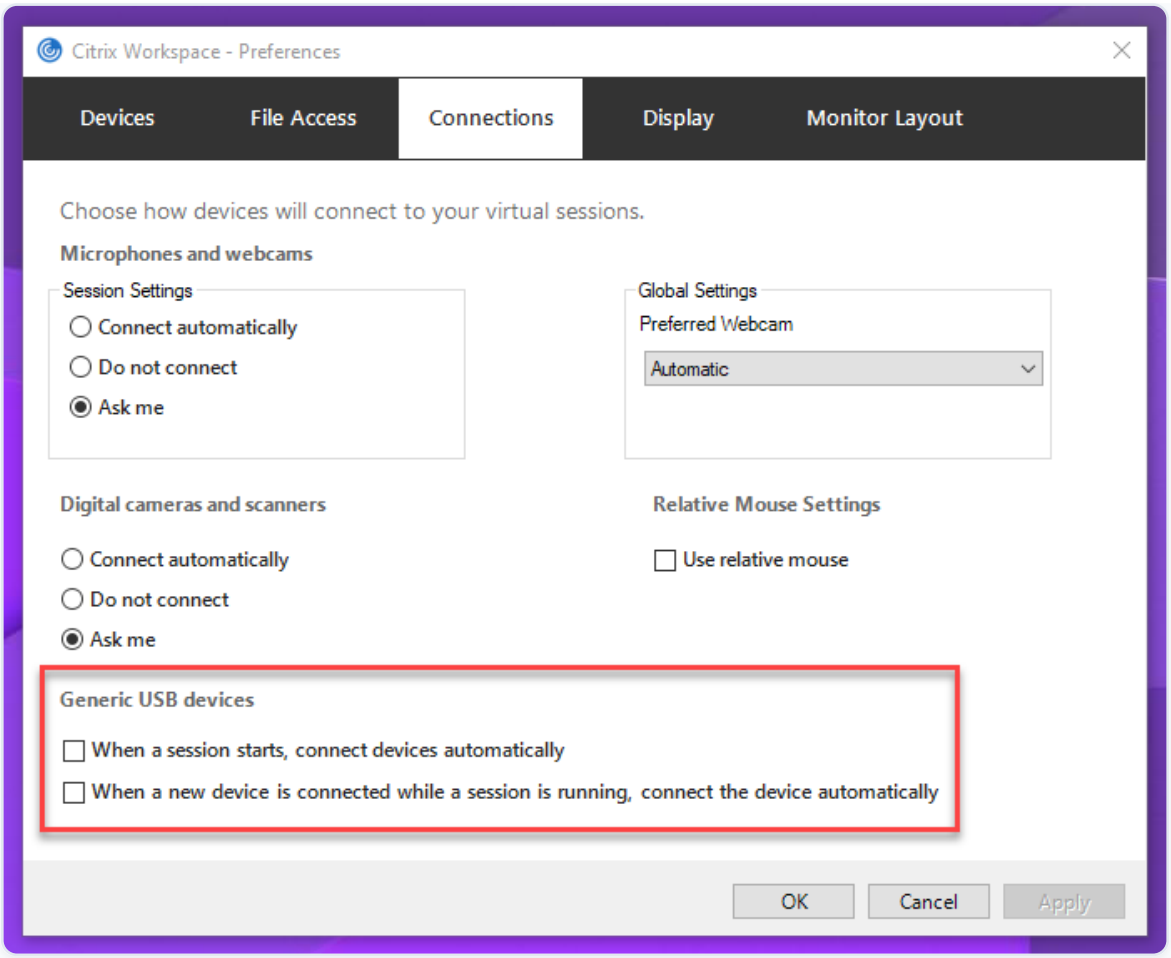
HDX Webカメラビデオ圧縮は、**最適化された**Webカメラモードとも呼ばれます。この種類のWebカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDX Webカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントはWebカメラとの通信を処理します。クライアントは、ビデオを適切に表示できるサーバーにのみビデオを送信します。サーバーはWebカメラを直接処理しませんが、その統合により、デスクトップで同じエクスペリエンスが得られます。Workspaceアプリは、帯域幅を節約し、WANシナリオでの回復力を向上させるためにビデオを圧縮します。



注:

- この機能は、この機能が依存する**videodev**カーネルモジュールがAzureマシンにないため、Azureマシンでは利用できません。
- 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。
- サポートされるWebカメラの解像度は48x32から1920x1080の範囲です。
- Webカメラを使用しているときに、Citrix Workspace™アプリのツールバーから**汎用USBデバイス**を選択しないでください。そうしないと、予期しない問題が発生する可能性があります。



サポートされているCitrix Workspaceアプリ

HDXウェブカメラビデオ圧縮は、Citrix Workspaceアプリの次のバージョンをサポートしています。

プラットフォーム	プロセッサ
Windows版 Citrix Workspaceアプリ	Windows向けCitrix Workspaceアプリは、XenAppおよびXenDesktop 7.17以降で32ビットおよび64ビットアプリのWebカメラビデオ圧縮をサポートしています。それ以前のバージョンでは、Windows向けCitrix Workspaceアプリは32ビットアプリのみをサポートしています。
Chrome向けCitrix Workspaceアプリ	一部のARM ChromebookはH.264エンコーディングをサポートしていないため、32ビットアプリのみが最適化されたHDX Webカメラビデオ圧縮を使用できます。

完全にテストされたWebカメラ

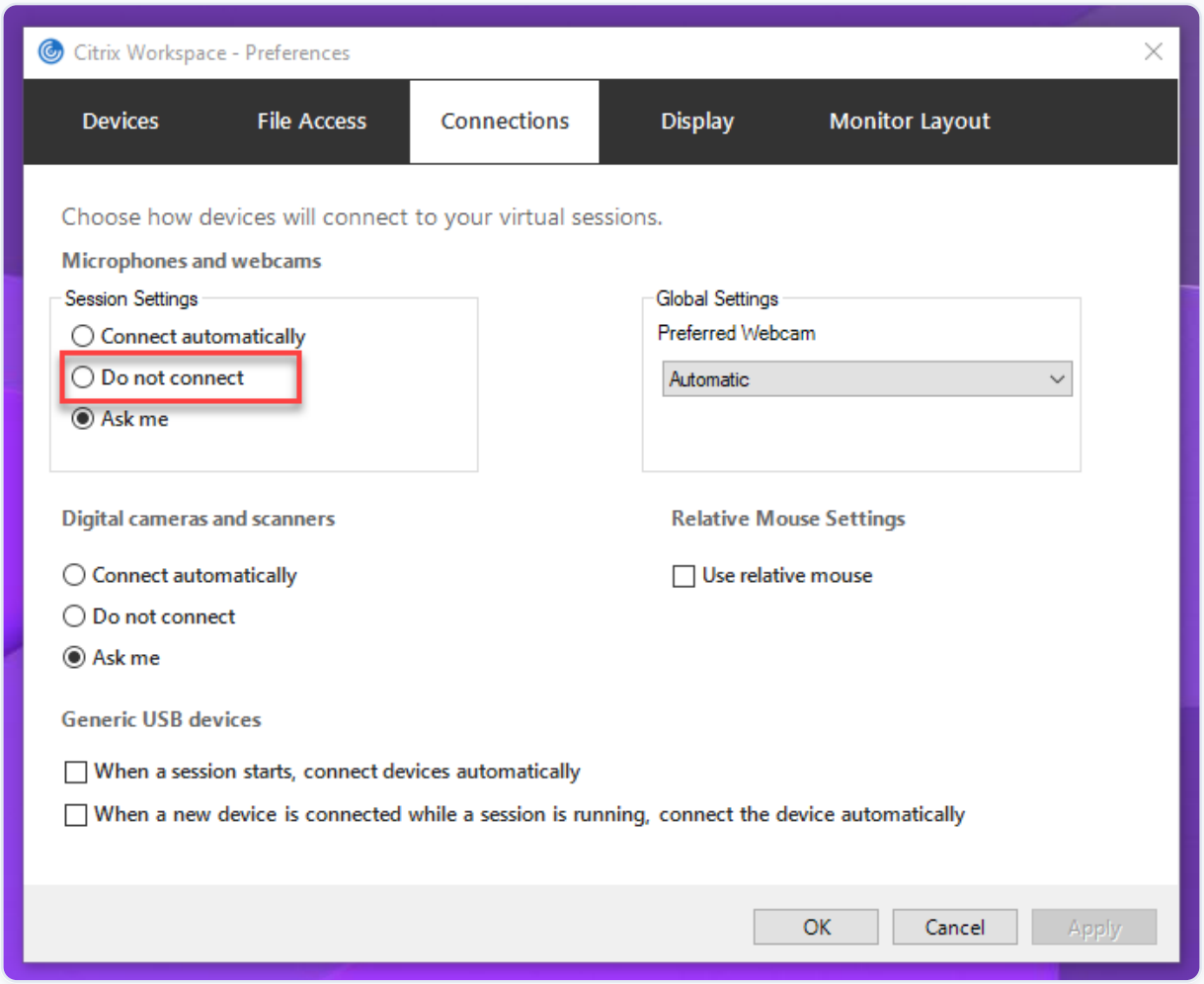
Webカメラによってフレームレート、明るさ、コントラストのレベルが異なります。Citrix®は、初期機能検証のために以下のWebカメラを使用しています。

- ロジクール HD ウェブカメラ C270
- ロジクール ウェブカメラ C930e
- マイクロソフト-LifeCam-HD3000

構成

この機能はデフォルトで有効になっています。これを使用するには、以下の検証と構成を完了してください。

> **ヒント:**
>
> Citrix Workspace app のユーザーは、Desktop Viewer の **Microphones and webcams** 設定で **Do not connect** 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™ に登録できること、および公開された Linux デスクトップセッションが Windows 資格情報を使用して正常に起動できることを確認します。
2. VDA がインターネットにアクセスできることを確認し、`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 に進んでください。

   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てください。

   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、次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 15.3、SUSE 15.2、または SUSE 12.5 に展開されている場合は、次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` および 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、および CentOS に上記のプログラムをインストールする場合、`ctxwcamcfg.sh` スクリプトは VDA に次のリポジトリをインストールして有効にします。
>
> - エンタープライズ Linux 用追加パッケージ (EPEL)
> - RPMフュージョン

b. `<https://github.com/umlaeute/v4l2loopback>` から `v4l2loopback` のオープンソースコードをダウンロードし、DKMS を使用して `v4l2loopback` を管理します。

`v4l2loopback` は、V4L2 ループバックデバイスを作成できるカーネルモジュールです。

c. `sudo service ctxwcamsd restart` コマンドを実行します。Linux VDA の Web カメラサービス (`ctxwcamsd`) が再起動し、HDX Web カメラビデオ圧縮機能のために `v4l2loopback` カーネルモジュールをロードします。

3. VDAにインターネットアクセスがない場合、別のマシンでv4l2loopbackカーネルモジュールをビルドし、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、`/var/xdl/`のパス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. v4l2loopback.koをそのマシンからVDAにコピーし、`/opt/Citrix/VDA/lib64/`の下に配置します。

f. VDAで、`sudo service ctxwcamsd restart`コマンドを実行してWebカメラサービスを再起動し、v4l2loopbackカーネルモジュールをロードします。

非ドメイン参加型Linux VDA

概要

非ドメイン参加型VDAは、VDAおよびユーザー認証のためにVDAをActive Directoryドメインに参加させる必要性をなくします。非ドメイン参加型VDAを作成する際、VDAをクラウドコントロールプレーンに登録するために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。ユーザーが非ドメイン参加型VDAからセッションを起動すると、VDAはCitrix Workspaceアプリにログオンするために使用したユーザー名を使用してローカルマッピングアカウントを作成します。VDAは、ローカルマッピングアカウントがSSOおよびセッション再接続に使用するランダムなパスワードを割り当てます。ランダムなパスワードを変更すると、SSOおよびセッション再接続は失敗します。SSOを無効にするには、非SSO認証を参照してください。

重要:

- 非ドメイン参加型VDAはCitrix DaaSでサポートされています。
  - コントロールプレーンはCitrix DaaS上に展開する必要があります。
  - パブリッククラウドまたはオンプレミスデータセンターに非ドメイン参加型VDAを展開できます。Citrix DaaSのコントロールプレーンは、非ドメイン参加型VDAを管理します。
  - Citrix Cloud ConnectorをバイパスするようにRendezvous V2を構成できます。そうでない場合は、VDAをコントロールプレーンに接続するためにCloud Connectorをインストールする必要があります。
- 非ドメイン参加型VDAを作成するには、Machine Creation Services™ (MCS)を使用する必要があります。
  - MCSはベアメタルサーバーをサポートしていません。

非ドメイン参加型Linux VDAで利用可能な機能

非ドメイン参加型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

非ドメイン参加型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**、および**ログインシェル**を含む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、次の手順を実行します。

1. 機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\LocalMappedAccount" -t
"REG_DWORD" -v "CreateWithUidGid" -d "0x00000001" --force
```

2. VDAのインストールパスの下にある`/var/xdl/getuidgid.sh`スクリプトで、次の属性を指定します。

属性	必須またはオプション	説明
`uid`	必須	ユーザー識別子 (UID) は、Linux がシステム上の各ユーザーに割り当てる番号です。これにより、ユーザー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
```

SSO 以外の認証

デフォルトでは、Linux VDA はシングルサインオン (SSO) が有効になっています。ユーザーは、Citrix Workspace アプリと VDA セッションに、1組の資格情報を使用してログオンします。 ユーザーが異なる資格情報を使用して VDA セッションにログオンするようにするには、Linux VDA で SSO を無効にします。詳しくは、「SSO 以外の認証」を参照してください。

Azure Active Directory を使用した認証

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。詳しくは、「Azure Active Directory を使用した認証」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 を使用することがサポートされています。詳しくは、「Rendezvous V2」を参照してください。

ドメイン参加していない Linux VDA の作成

MCS を使用して、Citrix DaaS でドメインに参加していない Linux VDA を作成します。詳細については、ドメインに参加していない Linux VDA の作成 を参照してください。

ポリシーサポートリスト

Linux VDA ポリシーサポートリスト

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントのローカル時間を使用	UseLocalTimeOfClient	ユーザー	ICA\タイムゾーン制御	サーバーのタイムゾーンを使用する
ICA ラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICA ラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
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帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクト帯域幅制限の割合	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクト帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクト帯域幅の割合	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅の制限	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅の制限の割合	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅の制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅の制限の割合	LimitPrinterBwPercent	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
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
ICAリスナーポート番号	IcaListenerPortNumber	コンピューター	ICA	1494
HDX™アダプティブトランスポート	HDXoverUDP	コンピューター	ICA	優先(2)
セッションの信頼性接続	AcceptSessionReliabilityConnections	コンピューター	ICA\セッションの信頼性	許可(1)
再接続UIの透過レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\自動クライアント再接続	80%
セッション信頼性ポート番号	SessionReliabilityPort	コンピューター	ICA\セッション信頼性	2598
セッション信頼性タイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッション信頼性	180 s

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
自動クライアント再接続	AllowAutoClientReconnect	ユーザー	ICA\自動クライアント再接続	許可 (1)
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDFユニバーサルプリンターの自動作成	AutoCreatePDFPrinter	ユーザー	印刷	無効 (0)
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	LimitClipboardTransferC2H	ユーザー	ICA	無効 (0)
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	LimitClipboardTransferH2C	ユーザー	ICA	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
クリップボードリダイレクトの帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限の割合	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が許可される形式	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	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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
追加の色圧縮	ExtraColorCompression	ユーザー	シンワイヤー	無効 (0)
ターゲットの最小フレームレート	TargetedMinimumFramesPerSecond	ユーザー	シンワイヤー	10 フレーム/秒
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ディングを使用	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)
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)
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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
セッションの最大数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオン許容数	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	仮想配信エージェント設定	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
Speexの最大品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブの自動接続	ドライブの自動接続	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	CD-ROMドライブの許可	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	固定ドライブの許可	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	フロッピードライブの許可	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	ネットワークドライブの許可	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)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
読み取り専用クライアントドライブアクセス	ReadOnlyMappedDrive	ユーザー	ファイルリダイレクト\CDM	無効 (0)
キーボードの自動表示	AllowAutoKeyboardPopUp	ユーザー	MRVC	無効 (0)
デスクトップとクライアント間のファイル転送を許可	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロード	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロード	ファイルアップロードを許可	ユーザー	ファイル転送	許可
セッションアイドルタイマー	EnableSessionIdleTimer	ユーザー	セッションタイマー	有効 (1)
セッションアイドルタイマー間隔	SessionIdleTimerInterval	ユーザー	セッションタイマー	1440分
切断されたセッションタイマー	EnableSessionDisconnectTimer	ユーザー	セッションタイマー	無効 (0)
切断されたセッションタイマー間隔	SessionDisconnectTimerPeriod	ユーザー	セッションタイマー	1440分
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

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
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

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、以下のCitrixポリシー設定を使用できます。

- **セッションアイドルタイマー**: アイドル状態のセッションに時間制限を適用するかどうかを決定します。
- **セッションアイドルタイマー間隔**: アイドル状態のセッションに時間制限を設定します。**セッションアイドルタイマー**が**有効**で、設定された時間内にアクティブなセッション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。

ポリシー設定を更新する際は、展開全体で一貫性があることを確認してください。

アイドルセッションの時間制限が切れると、警告メッセージが表示されます。例については、次のスクリーンショットを参照してください。**OK**を押すと警告メッセージは閉じますが、セッションをアクティブに保つことはできません。セッションをアクティブに保つには、ユーザー入力を提供してアイドルタイマーをリセットしてください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2308/media/warning-message-idle-session-timeout.png)

以下のポリシーは、Citrix Studioバージョン7.12以降で構成できます。

- MaxSpeexQuality

  値（整数）:

  デフォルト値: 5

  詳細:

  オーディオ品質が中または低の場合（ポリシー「オーディオ品質」を参照）、オーディオリダイレクトはSpeexコーデック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、最大値が次の条件を満たす場合です。

  - 最大Speex品質以下であること。
  - そのビットレートが帯域幅制限以下であること。

  関連設定: オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

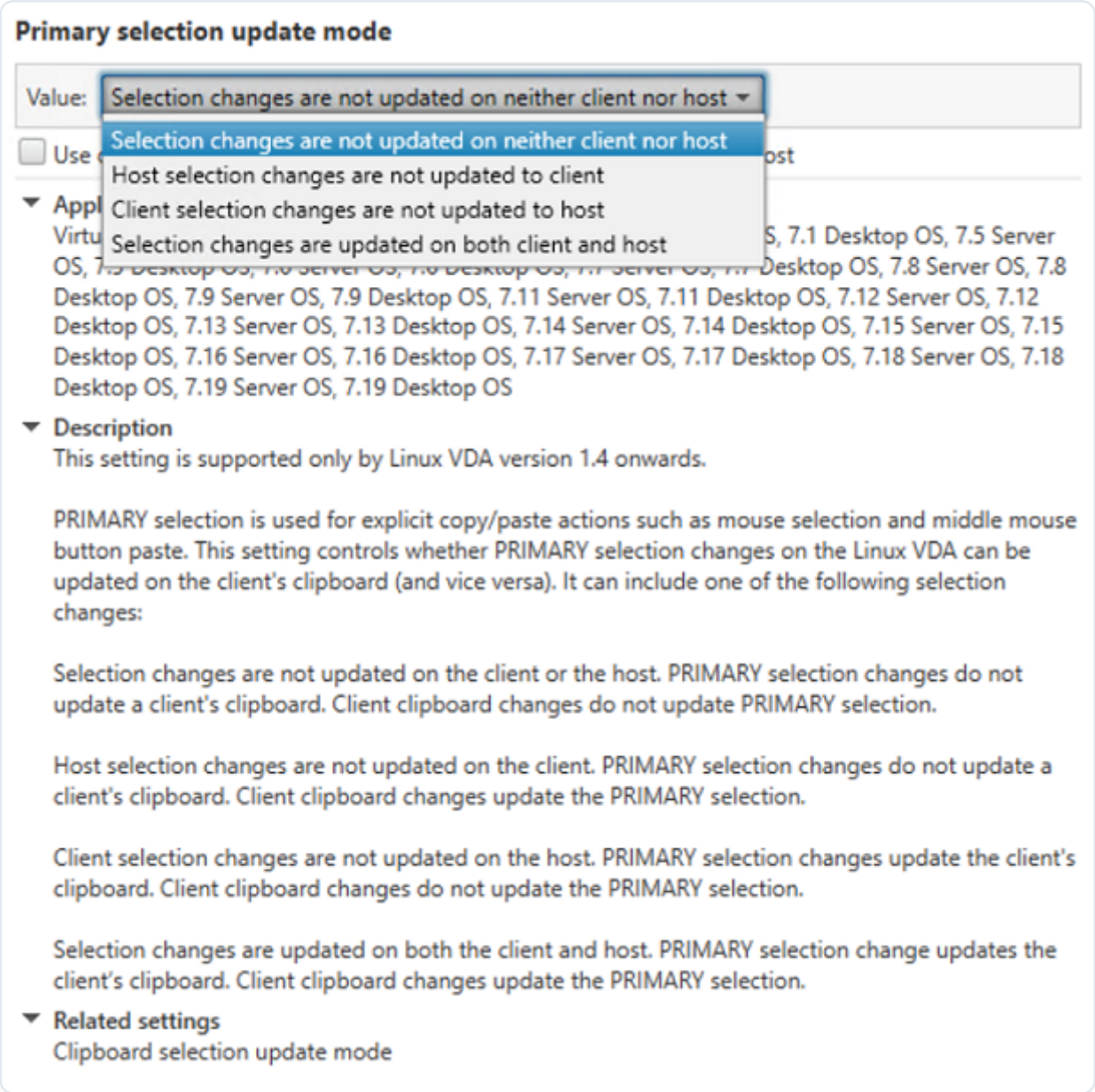
  値 (enum):

  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 (enum):

  デフォルト値: 3

  詳細:

  クリップボード選択は、データをいくつか選択し、ショートカットメニューから「コピー」を選択するなどして、明示的にクリップボードに「コピー」するよう要求する場合に使用されます。クリップボード選択は主にMicrosoft Windowsのクリップボード操作に関連して使用され、プライマリ選択はLinux固有のものです。

  このポリシーは、Linux VDAとクライアントでのクリップボード選択の変更が、互いのクリップボードを更新できるかどうかを制御します。値のオプションは4つあります。

  クリップボード選択更新モードの画像(/en-us/linux-virtual-delivery-agent/7-15-ltsr/media/clipboard-selection-update-mode.png)

  - **クライアントとホストのどちらでも選択の変更は更新されない**

    Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

  - **ホストの選択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、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。

- **クライアントの選択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、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **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、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

> **注:**
>
> Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピー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には、**cups**フィルターと**foomatic**フィルターの両方が必要です。これらのフィルターは、VDAのインストール時にインストールされます。ディストリビューションに基づいて、フィルターを手動でインストールすることもできます。例：

RHEL 7の場合：

```
sudo yum –y install cups



sudo yum -y install foomatic-filters
```

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

この設定は、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かどうかを制御します。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、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のドキュメントにある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。

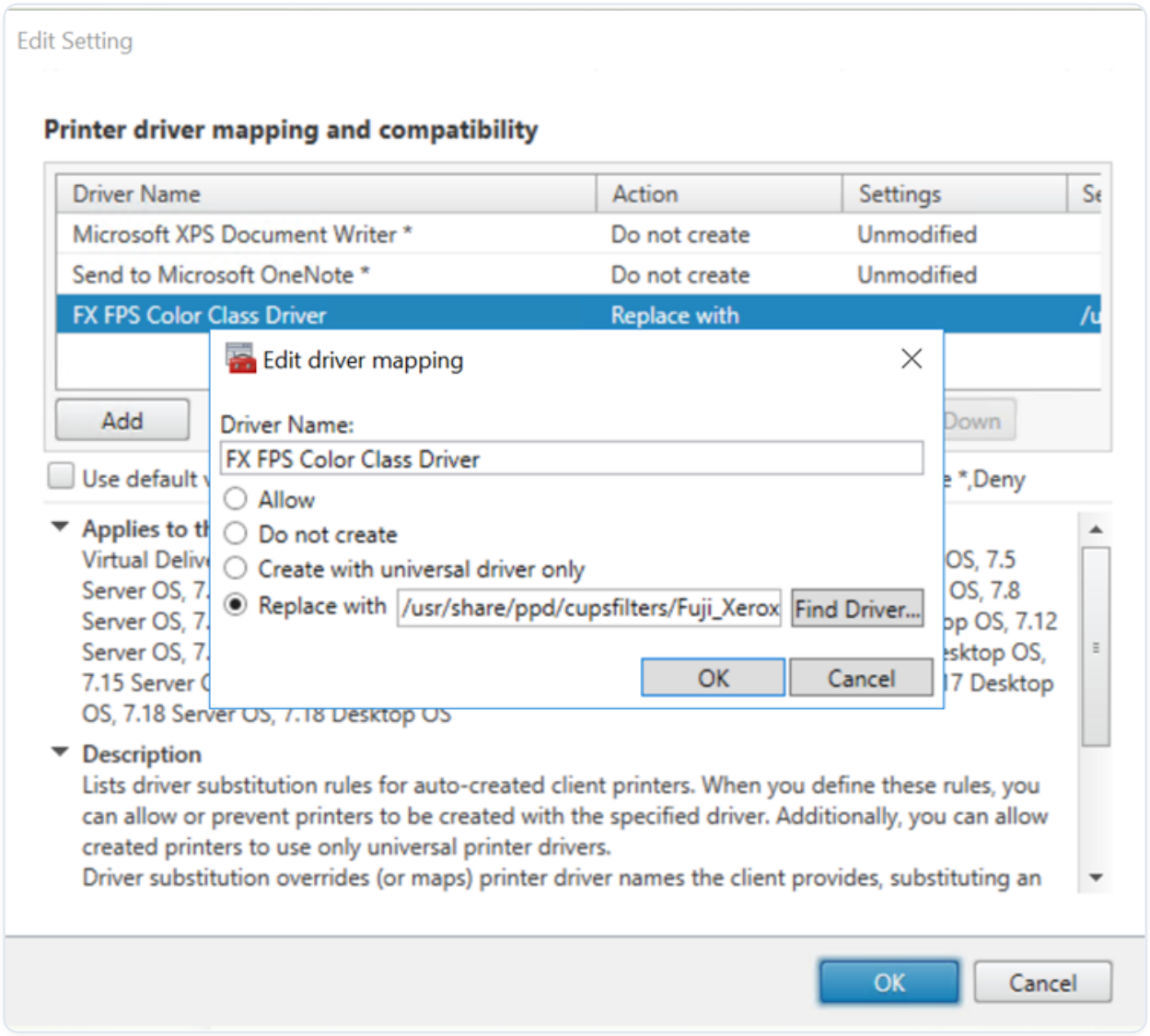
プリンタードライバーのマッピングと互換性

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\. **次で置換**を選択し、VDA上のドライバーファイルの絶対パスを入力します。



注:

- PPDドライバーファイルのみがサポートされています。
- **プリンタードライバーのマッピングと互換性**ポリシーのその他のオプションはサポートされていません。**次で置き換える**のみが有効になります。

使用方法

公開デスクトップと公開アプリケーションの両方から印刷できます。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。プリンター名はデスクトップとアプリケーションで異なります。

- 公開デスクトップの場合：

`<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になります。公開アプリケーションセッションでデスクトッププリンターに印刷したり、公開デスクトップでアプリケーションプリンターに印刷したりすることはできません。

トラブルシューティング

印刷できない

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。次のコマンドを実行して、印刷デーモンが実行されていることを確認します。**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**を開始します。

```
ps –ef | grep ctxlpmngt
```

それでも印刷が機能しない場合は、**CUPS**フレームワークを確認してください。**ctxcups**サービスはプリンター管理に使用され、Linux CUPSフレームワークと通信します。これはマシンごとに1つのプロセスであり、次のコマンドを実行して確認できます。

```
service ctxcups status
```

CUPSログを収集するための追加手順

CUPSログを収集するには、次のコマンドを実行してCUPSサービスファイルを構成します。そうしないと、CUPSログは**hdx.log**に記録されません。

```
sudo service cups stop
```

```
sudo vi /etc/systemd/system/printer.target.wants/cups.service


PrivateTmp=false


sudo service cups start


sudo systemctl daemon-reload
```

注:

この構成は、問題が発生したときに完全な印刷ログを収集するためだけに行われます。通常、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、出力の文字化け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:

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が **printername** です。これは読み取り専用の値です。編集しないでください。

ppdpath、**model**、および **drivertype** の各フィールドは同時に設定できません。**mapped printer** に対して有効になるのは1つだけだからです。

- 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は、**model=** オプションを使用してネイティブプリンタードライバーのモデルを構成します。**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は、ネイティブプリンタードライバーのPPDファイルパスを構成します。**ppdpath** の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  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. ログオフして再度ログオンし、構成の変更を読み込みます。
5. 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。内容が正しいかどうかを確認してください。

   スプールファイルがPCLの場合、またはクライアントOSがWindowsの場合は、スプールファイルをクライアントにコピーし、別のプリンタードライバーを使用してクライアント側のプリンターで印刷します。
8. **マップされたプリンター**を変更して、別のプリンタードライバーを使用します。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** をクリックします。この操作には管理者権限が必要です。

   d. **cups-ctx** 接続を維持し、**続行** をクリックしてプリンタードライバーを変更します。

   e. **製造元** および **モデル** フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。

   f. このプロセスが成功した場合、`.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、以下の問題が確認されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブのもの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構成ファイル **/etc/cups/cupsd.conf** を変更することで無効にできます。

ファイルに構成されている現在のポリシー名を見つけます。

`DefaultPolicy default`

ポリシー名が *default* の場合は、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バージョンのCitrix Workspace™アプリを使用すると、Linux VDAセッション内から変換されたPDFを印刷できます。セッションの印刷ジョブは、Citrix Workspaceアプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、任意のPDFビューアを使用してPDFを開き、任意のプリンターで印刷できます。

Linux VDAは、以下のバージョンのCitrix WorkspaceアプリでPDF印刷をサポートしています。

- Citrix Receiver for HTML5バージョン2.4〜2.6.9、Citrix Workspaceアプリ1808 for HTML5以降
- Citrix Receiver for Chromeバージョン2.4〜2.6.9、Citrix Workspaceアプリ1808 for Chrome以降
- Citrix Workspaceアプリ1905 for Windows以降

構成

PDF印刷をサポートするバージョンのCitrix Workspaceアプリを使用する以外に、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（デフォルトでは**Allowed**）。
- **Auto-create PDF Universal Printer**を**Enabled**に設定します（デフォルトでは**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に安全にリモートアクセスすることを容易にできます。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 のコンポーネントと同じものを使用します。Remote PC Access の展開と構成の要件およびプロセスは、Citrix Virtual Apps and Desktops の展開に必要な要件およびプロセスと同じです。この統一性により、一貫性のある統合された管理エクスペリエンスが提供されます。ユーザーは、Citrix HDX を使用してリモートオフィスPCセッションを配信することで、最高のユーザーエクスペリエンスを得られます。

詳細については、Citrix Virtual Apps and Desktops のドキュメントにある 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 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のローカル画面を非表示にすることはできません。この画面が表示されることは、潜在的なセキュリティリスクとなります。
- 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™ アプリは、ユーザーが Remote PC Access デスクトップセッション内でオフィスPC上のアプリケーションとデータにアクセスするのに役立ちます。
- ユーザーがすでにローカルでPCにログオンしている場合、StoreFront™からPCを起動しようとすると失敗します。
- 省電力オプションはLinuxマシンでは利用できません。

構成

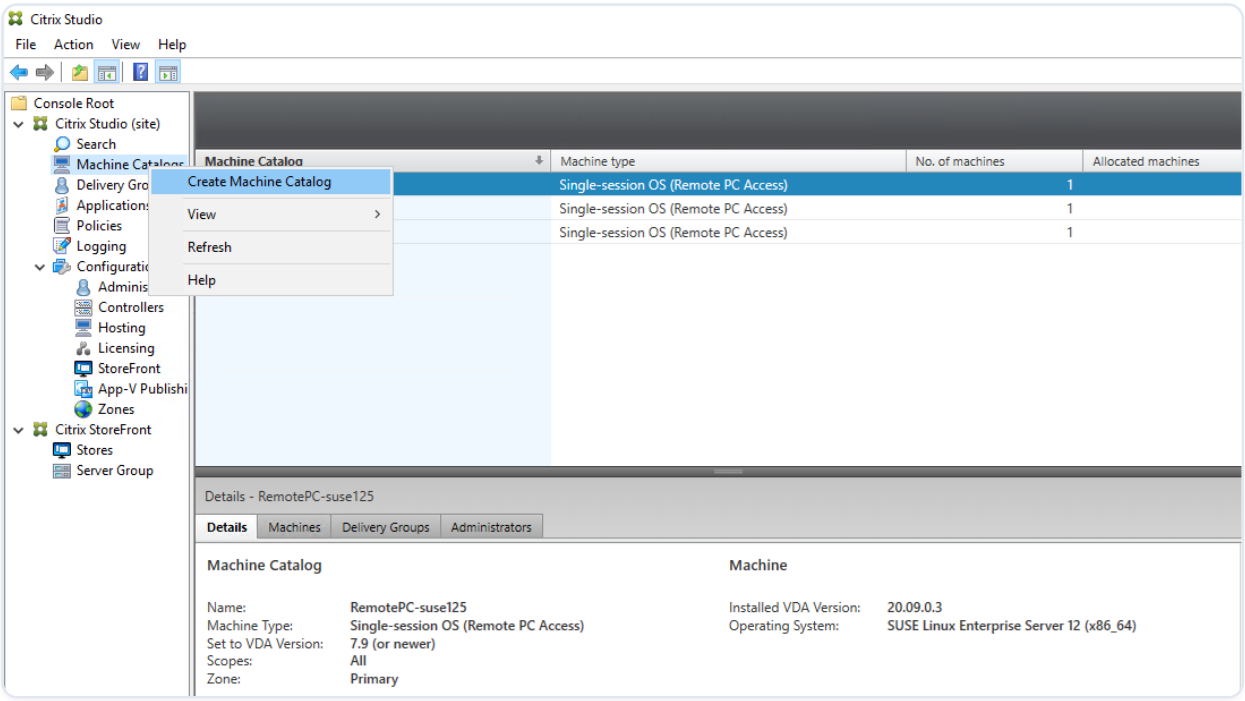
Linux PCセッションを配信するには、ターゲットPCに Linux VDA をインストールし、**Remote PC Access** タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成します。次のセクションで手順を詳しく説明します。

ステップ 1 - ターゲットPCに Linux VDA をインストールする

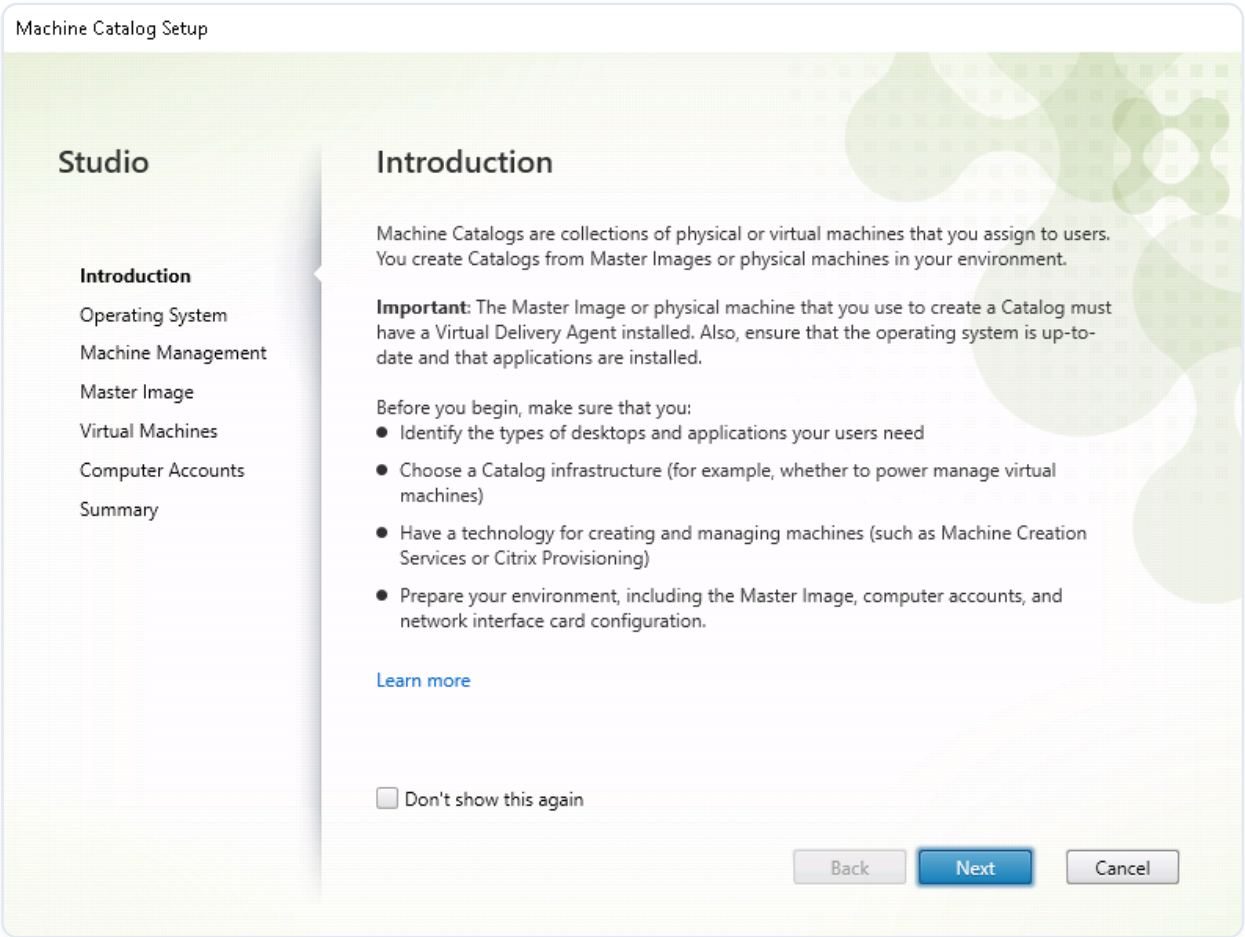
Linux VDA のインストールには、簡易インストール を使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE` 変数の値を `Y` に設定します。

ステップ 2 - Remote PC Access タイプのマシンカタログを作成する

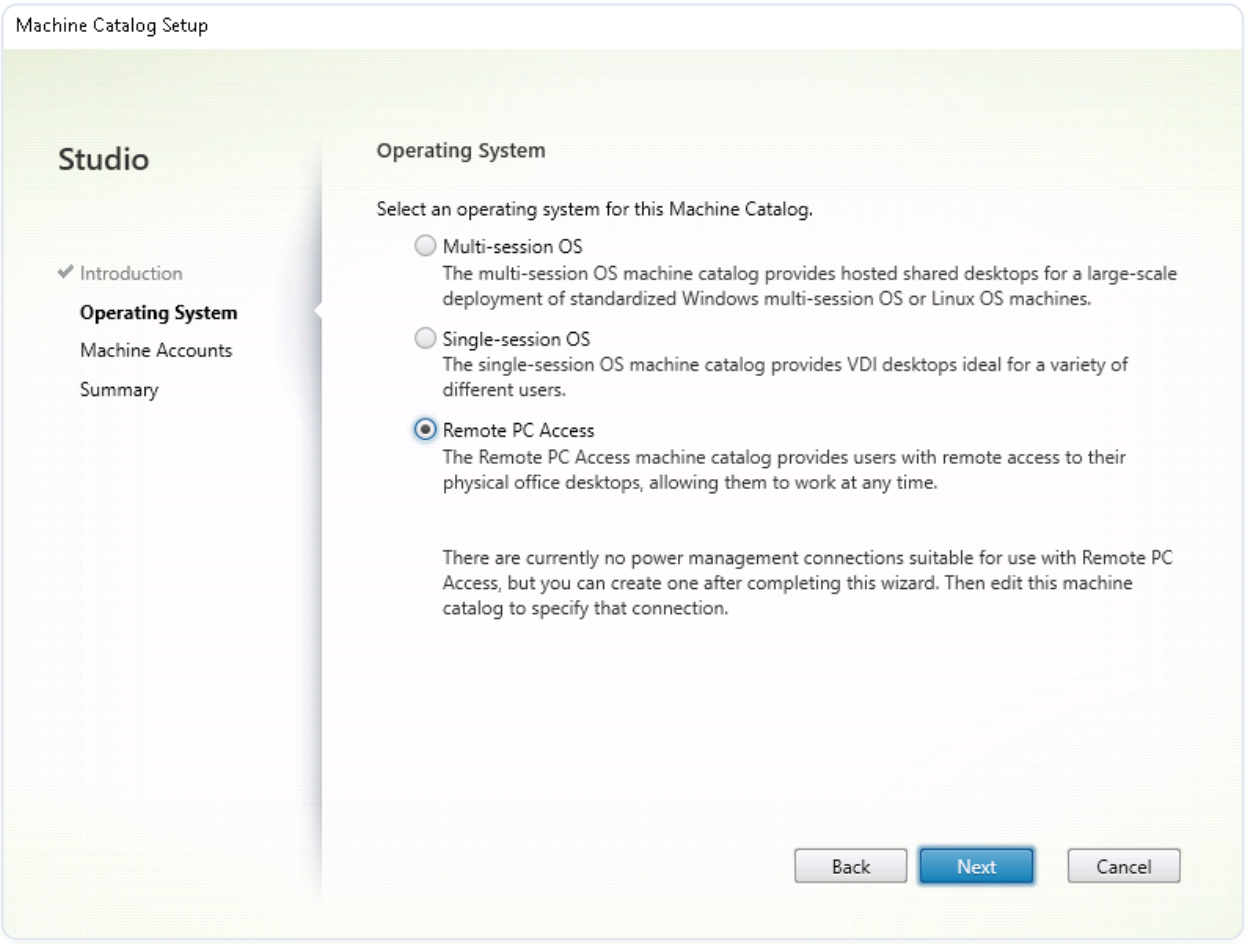
1. Citrix Studio で、**マシンカタログ** を右クリックし、ショートカットメニューから **マシンカタログの作成** を選択します。



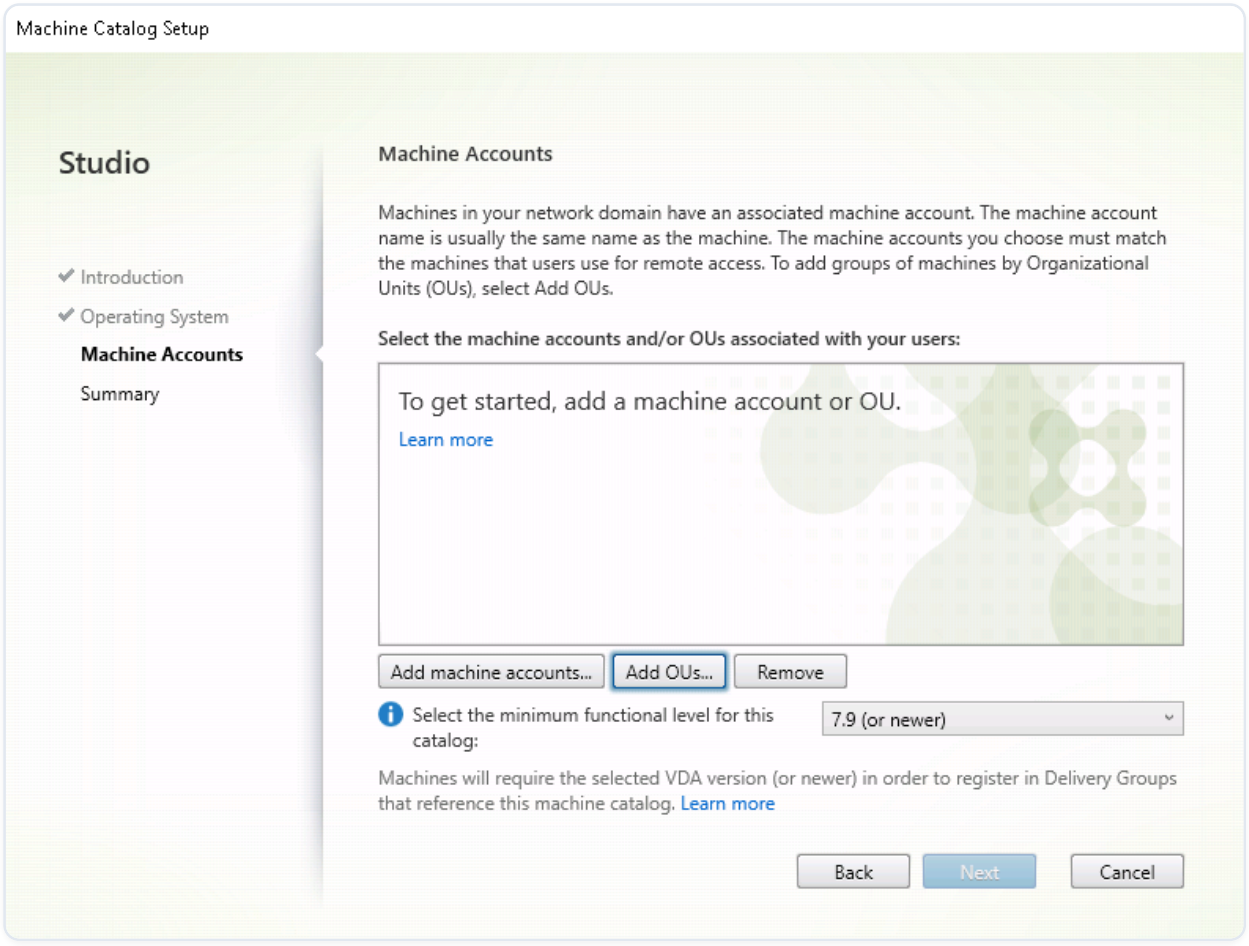
2. **はじめに** ページで **次へ** をクリックします。



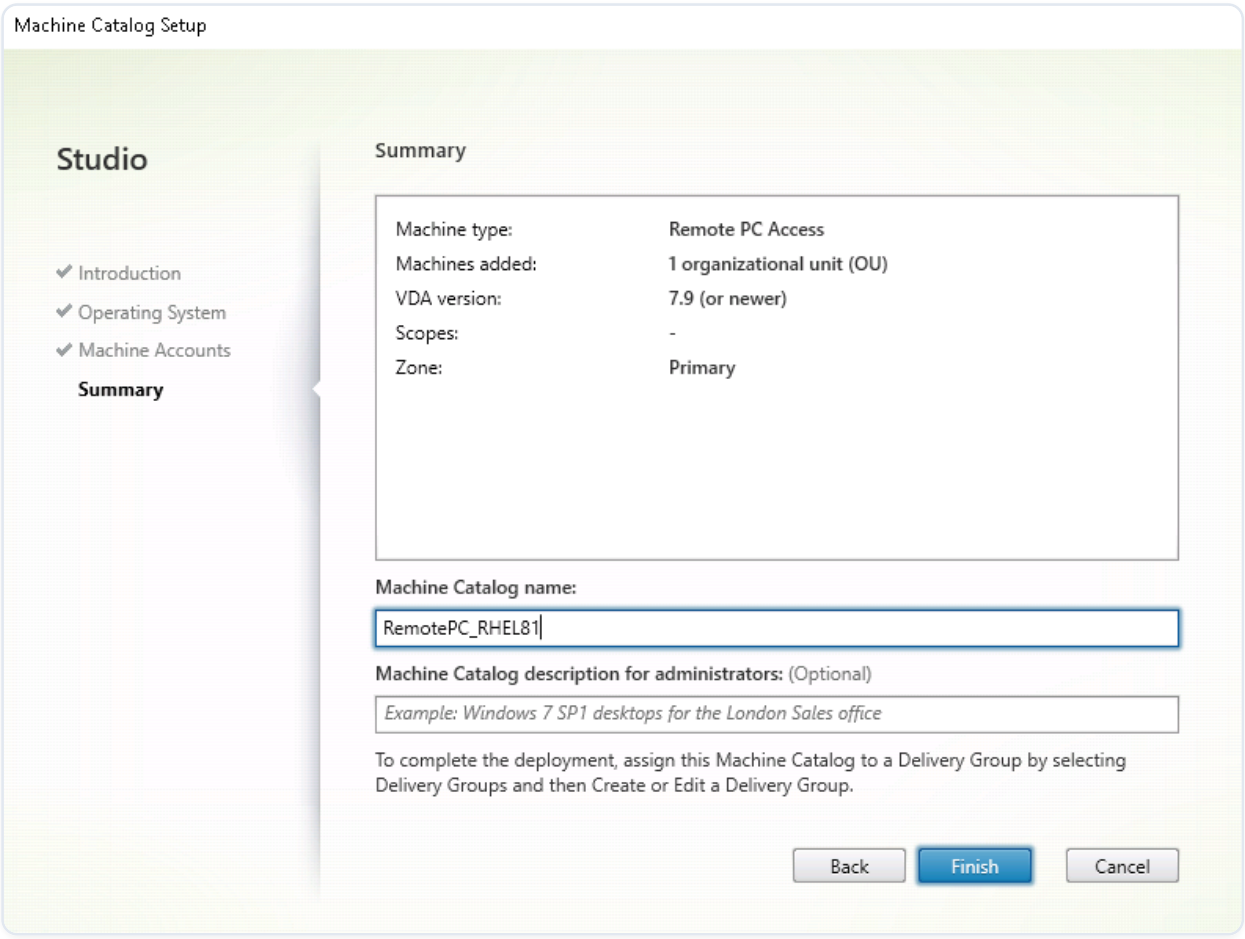
3. **オペレーティングシステム** ページで **Remote PC Access** を選択します。



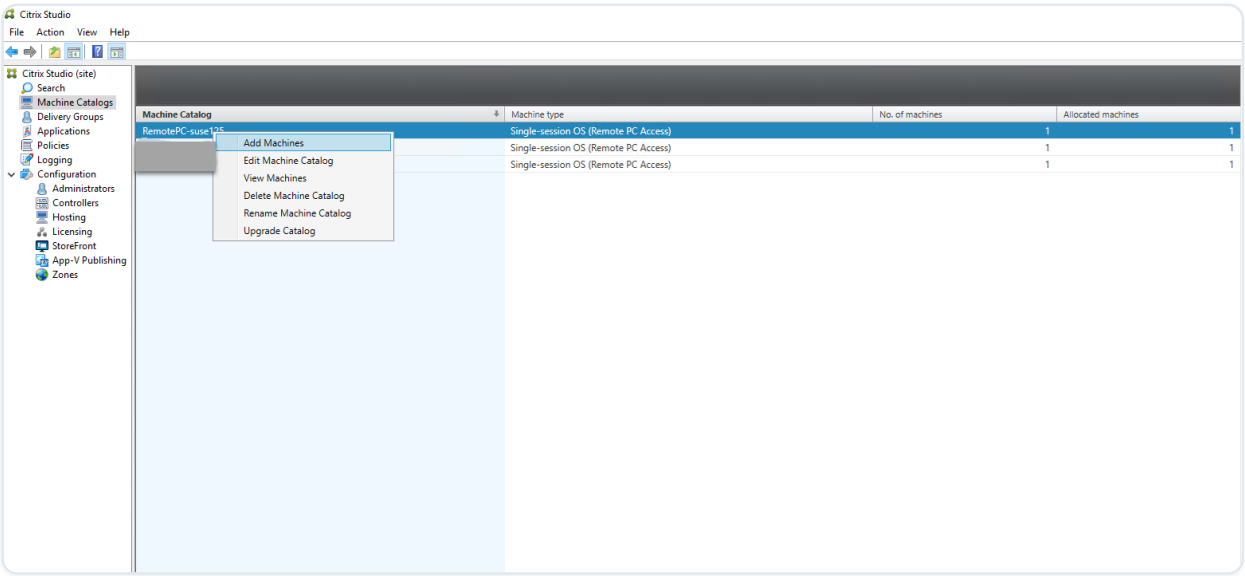
4. ターゲットPCを含むOUを選択するには「**OUの追加**」をクリックするか、個々のマシンをマシンカタログに追加するには「**マシンアカウントの追加**」をクリックします。



5. マシンカタログに名前を付けます。

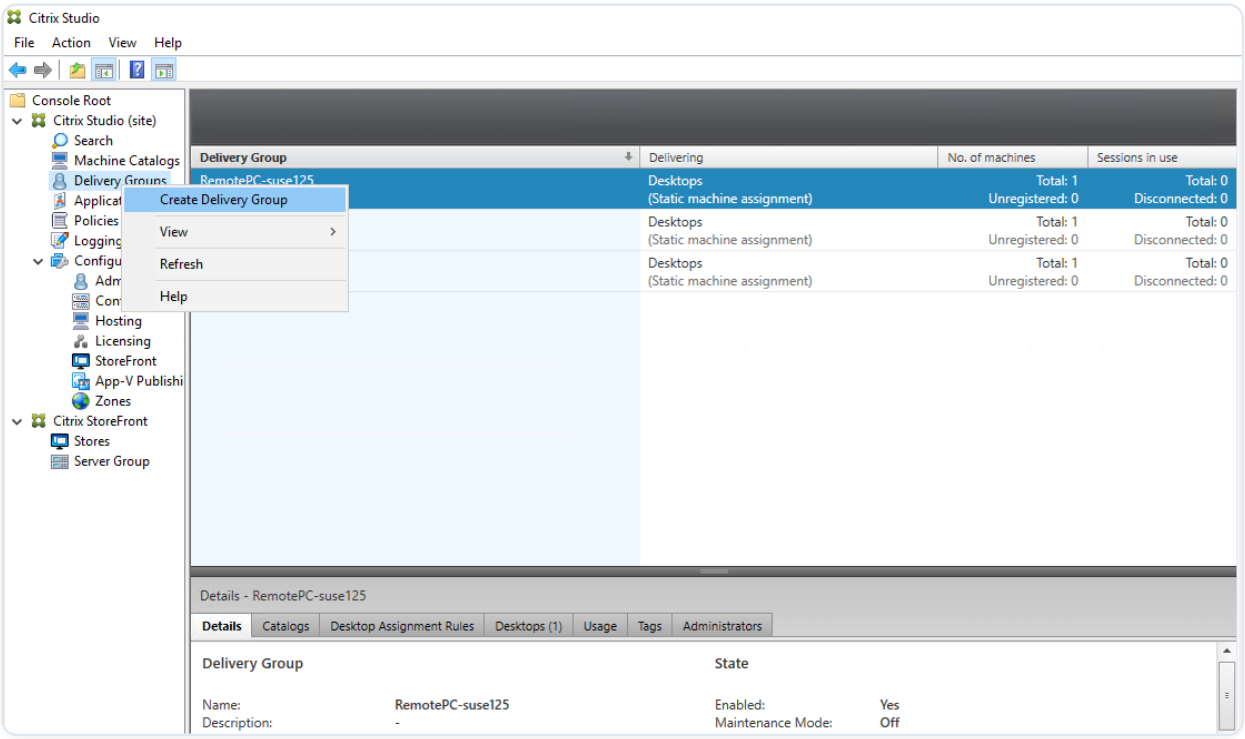


6. （オプション）マシンカタログを右クリックして、関連する操作を実行します。

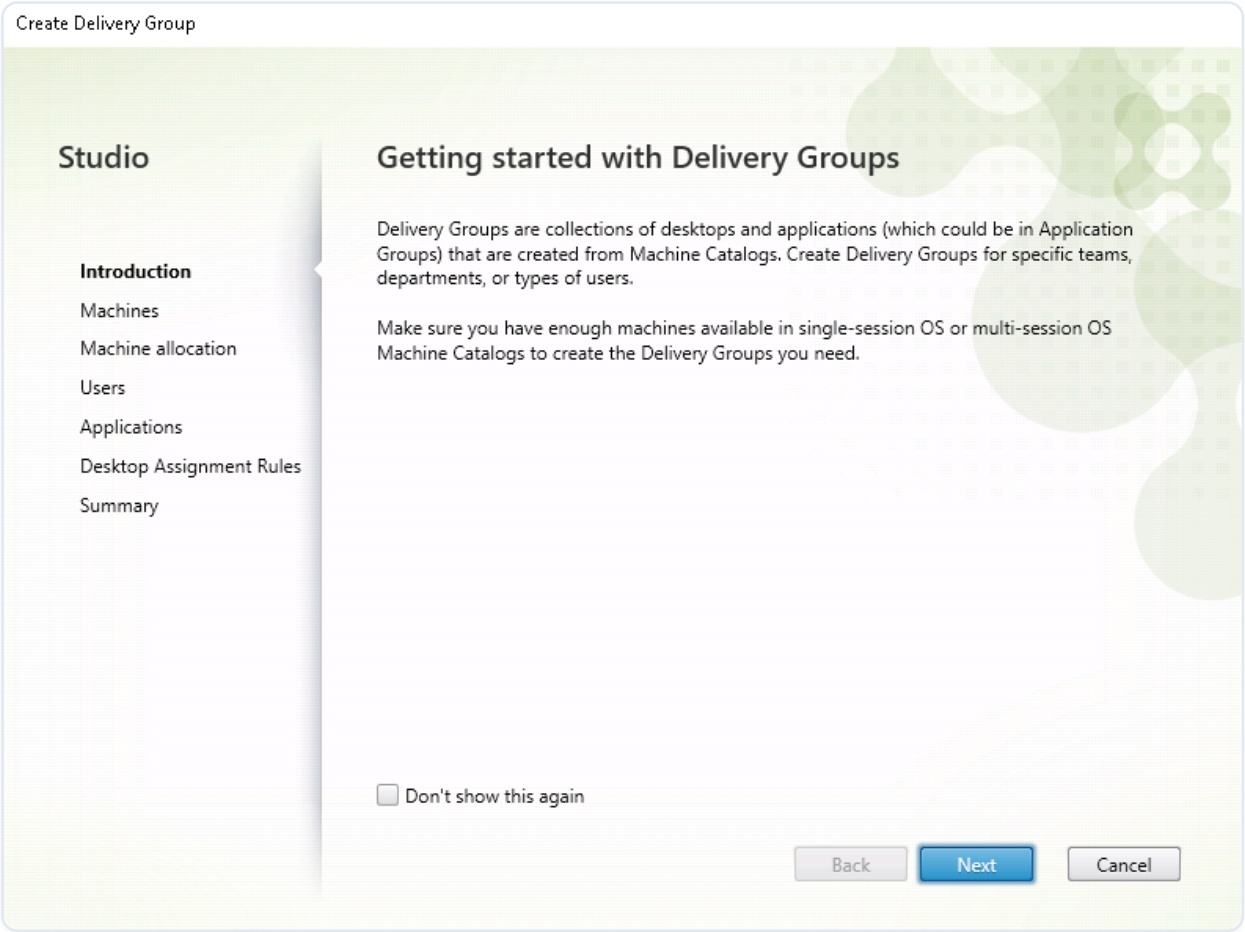


ステップ3 - アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成する

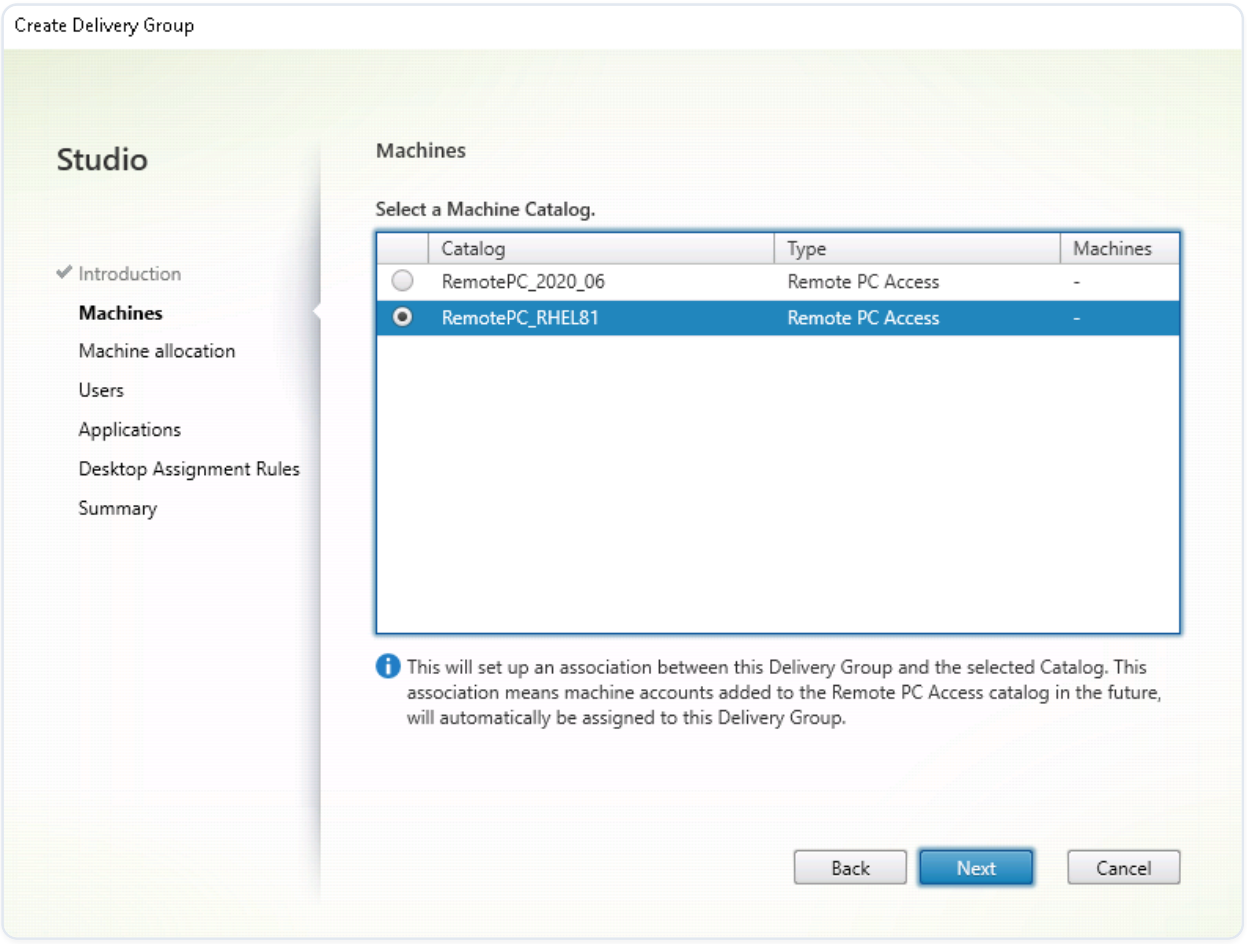
1. Citrix Studioで、「**デリバリーグループ**」を右クリックし、ショートカットメニューから「**デリバリーグループの作成**」を選択します。



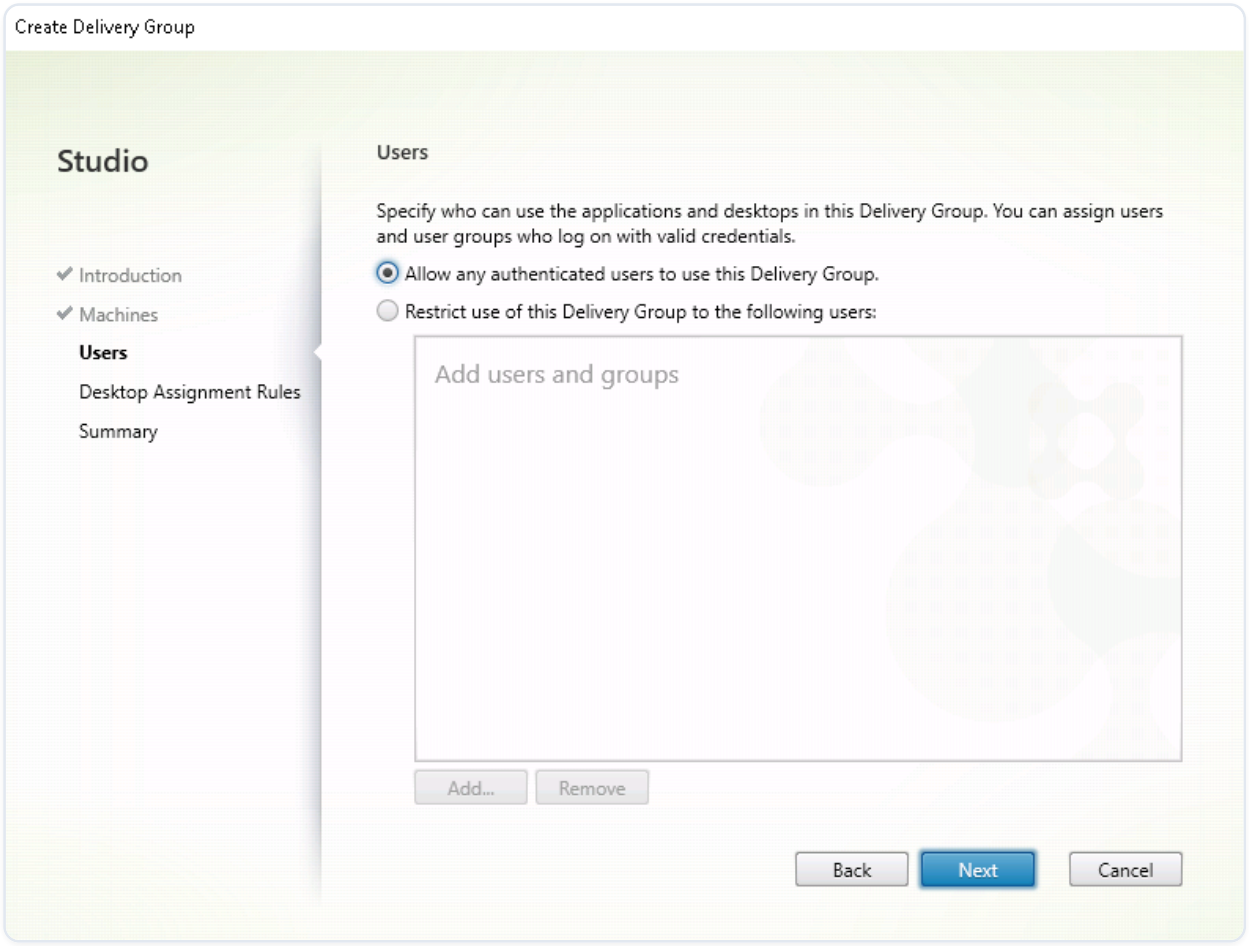
2. 「**デリバリーグループの開始**」ページで「**次へ**」をクリックします。



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。



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。



Wake on LAN (ウェイク・オン・ラン)

Remote PC AccessはWake on LANをサポートしており、ユーザーは物理PCをリモートで起動できます。この機能により、ユーザーは使用していないときにオフィスPCの電源を切ったままにして、エネルギーコストを節約できます。また、誤ってマシンの電源が切れてしまった場合でも、リモートアクセスを可能にします。

Wake on LAN機能を使用すると、デリバリーコントローラーからの指示があった場合、PC上で実行されているVDAから、そのPCが存在するサブネットにマジックパケットが直接送信されます。これにより、追加のインフラストラクチャコンポーネントやマジックパケット配信のためのサードパーティソリューションに依存することなく、この機能が動作します。

Wake on LAN機能は、レガシーなSCCMベースのWake on LAN機能とは異なります。SCCMベースのWake on LANの詳細については、「Wake on LAN – SCCM統合」を参照してください。

システム要件

Wake on LAN機能を使用するためのシステム要件は次のとおりです。

- コントロールプレーン:

    - シトリックス DaaS™（旧称：シトリックス バーチャルアップス アンド デスクトップス サービス）
    - シトリックス バーチャルアップス アンド デスクトップス 2012以降
- 物理PC:
    - VDAバージョン2012以降
    - BIOSおよびNICでWake-on-LANが有効になっていること

Wake on LANの構成

現在、統合されたWake-on-LANの構成は、PowerShellを使用してのみサポートされています。

Wake on LANを構成するには:

1. まだ Remote PC Access マシンカタログがない場合は、作成します。
2. まだ Wake on LAN ホスト接続がない場合は、作成します。

> **注:**
>
> Wake-on-LAN機能を使用するには、「Microsoft Configuration Manager Wake on LAN」タイプのホスト接続がある場合は、ホスト接続を作成します。

3. Wake on LAN ホスト接続の一意の識別子を取得します。
4. Wake on LAN ホスト接続をマシンカタログに関連付けます。

   Wake on LAN ホスト接続を作成するには：

```
# Load Citrix SnapIns

Add-PSSnapIn -Name "*citrix*"



# Provide the name of the Wake on LAN host connection

[string]$connectionName = "Remote PC Access Wake on LAN"
```

```
# Create the hypervisor connection

$hypHc = New-Item -Path xdhyp:\Connections `

             -Name $connectionName `

             -HypervisorAddress "N/A" `

             -UserName "woluser" `

             -Password "wolpwd" `

             -ConnectionType Custom `

             -PluginId VdaWOLMachineManagerFactory `

             -CustomProperties "<CustomProperties></CustomProperties>" `

             -Persist



$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid



# Wait for the connection to be ready before trying to use it

while (-not $bhc.IsReady)

{

    Start-Sleep -s 5

    $bhc = Get-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid

}
```

ホスト接続の準備ができたら、次のコマンドを実行してホスト接続の一意の識別子を取得します。

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"

$hypUid = $bhc.Uid
```

接続の一意の識別子を取得したら、次のコマンドを実行して、その接続を Remote PC Access マシンカタログに関連付けます。

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog
-RemotePCHypervisorConnectionUid $hypUid
```

5. マシンカタログ内の各VMで、BIOSおよびNICでWake-on-LANを有効にします。

   注： Wake-on-LANを有効にする方法は、マシンの構成によって異なります。

   - BIOSでWake-on-LANを有効にするには：

     a. BIOSに入り、Wake-on-LAN機能を有効にします。

        BIOSにアクセスする方法は、マザーボードの製造元および製造元が選択したBIOSベンダーによって異なります。

     b. 設定を保存し、マシンを再起動します。

   - NICでWake-on-LANを有効にするには：

     a. `sudo ethtool <NIC>`コマンドを実行して、NICがマジックパケットをサポートしているかどうかを確認します。

        `<NIC>`はNICのデバイス名です。たとえば、`eth0`です。`sudo ethtool <NIC>`コマンドは、NICの機能に関する出力を提供します：

        - 出力に`Supports Wake-on: <letters>`のような行が含まれており、`<letters>`に`g`の文字が含まれている場合、お使いのNICはWake on LANマジックパケット方式をサポートしています。
        - 出力に`Wake-on: <letters>`のような行が含まれており、`<letters>`に`g`の文字が含まれていて`d`の文字が含まれていない場合、Wake on LANマジックパケット方式は有効になっています。ただし、`<letters>`に`d`の文字が含まれている場合、Wake on LAN機能が無効になっていることを示します。この場合、`sudo ethtool -s <NIC> wol g`コマンドを実行してWake on LANを有効にします。

b. ほとんどのディストリビューションでは、起動するたびに`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ます。

Ubuntu:

インターフェイス構成ファイル`/etc/network/interfaces`に`up ethtool -s <NIC> wol g`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

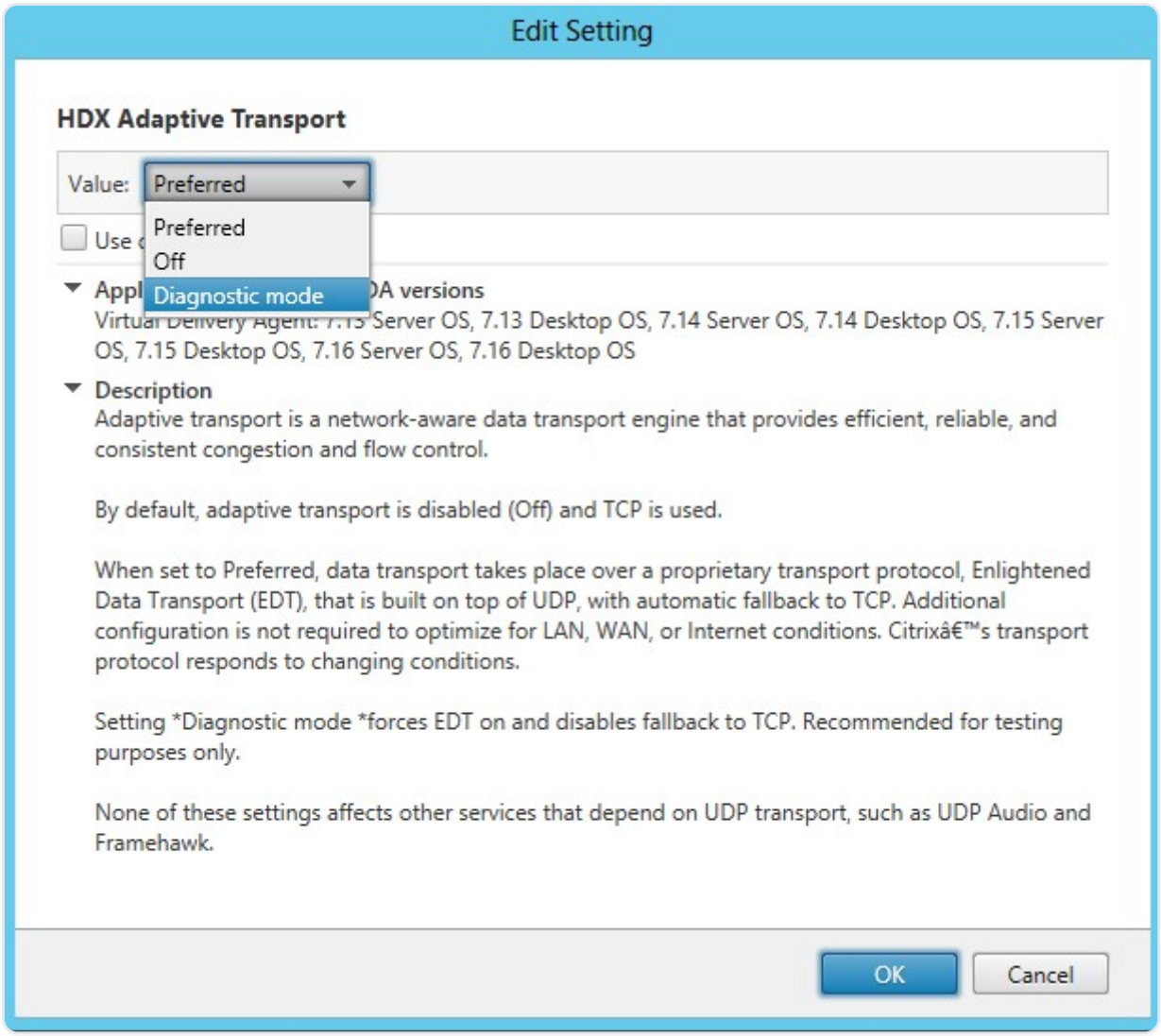
アダプティブトランスポートは、Citrix Virtual Apps and Desktops™のメカニズムであり、ICA接続のトランスポートプロトコルとしてEnlightened Data Transport (EDT)を使用できます。EDTが利用できない場合、アダプティブトランスポートはTCPに切り替わります。

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。これにより、困難な長距離接続でも優れたユーザーエクスペリエンスを提供し、サーバーのスケーラビリティを維持します。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、より優れた一貫性のある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のドキュメントにある「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™ Adaptive Transport**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。



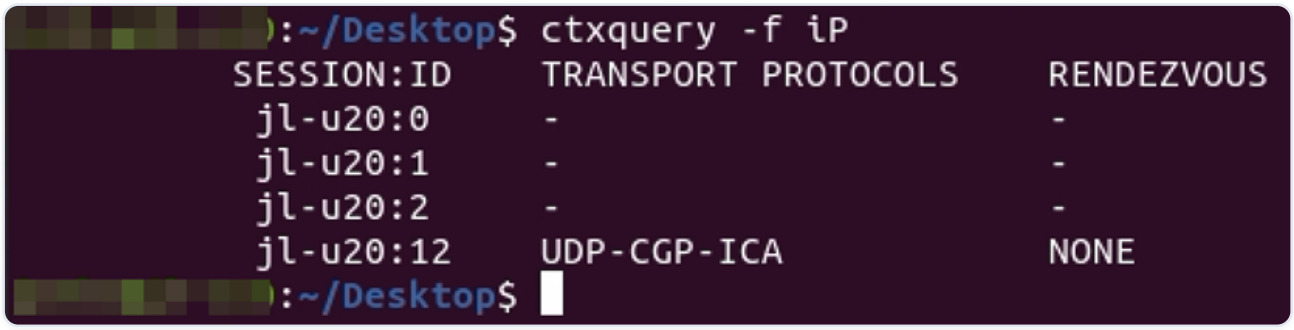
- **優先**: アダプティブトランスポートが有効になり、Enlightened Data Transport (EDT)を優先トランスポートプロトコルとして使用し、TCPにフォールバックします。
- **診断モード**: アダプティブトランスポートが有効になり、EDTの使用が強制されます。TCPへのフォールバックは無効になります。この設定は、テストとトラブルシューティングのみに推奨されます。
- **オフ**: アダプティブトランスポートが無効になり、トランスポートにはTCPのみが使用されます。

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パケットの断片化を防ぎます。

システム要件:

- Linux VDA バージョン 2012 以降
- 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 ディスカバリの制御

EDT MTU ディスカバリはデフォルトで無効になっています。

- EDT MTU ディスカバリを有効にするには、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 ディスカバリを無効にするには、`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**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 ディスカバリの制御

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 ディスカバリを選択的に制御できます。この機能を無効にするには、`Application` セクションの下に以下を設定します。

`MtuDiscovery=Off`

この機能を再度有効にするには、ICA ファイルから `MtuDiscovery` パラメーターを削除します。

> **重要:**
>
> このICAファイルパラメーターが機能するには、VDAでEDT MTUディスカバリを有効にする必要があります。VDAでEDT MTUディスカバリ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。

強化されたEDT輻輳制御

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ました。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。これを有効にするには、次のコマンドを実行してから、`ctxhdx` サービスを再起動してください。

```
/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
```

セッションログオン画面でのカスタム背景とバナーメッセージ

セッションログオン画面にカスタム背景またはバナーメッセージを追加する

> **ヒント:**

> SUSE 15.4でこの機能を使用するには、`<http://download.opensuse.org/distribution/leap/15.3/repo/oss/>`から`imlib2`をインストールします。

> 次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、その後、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、次を実行します:

```
/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。

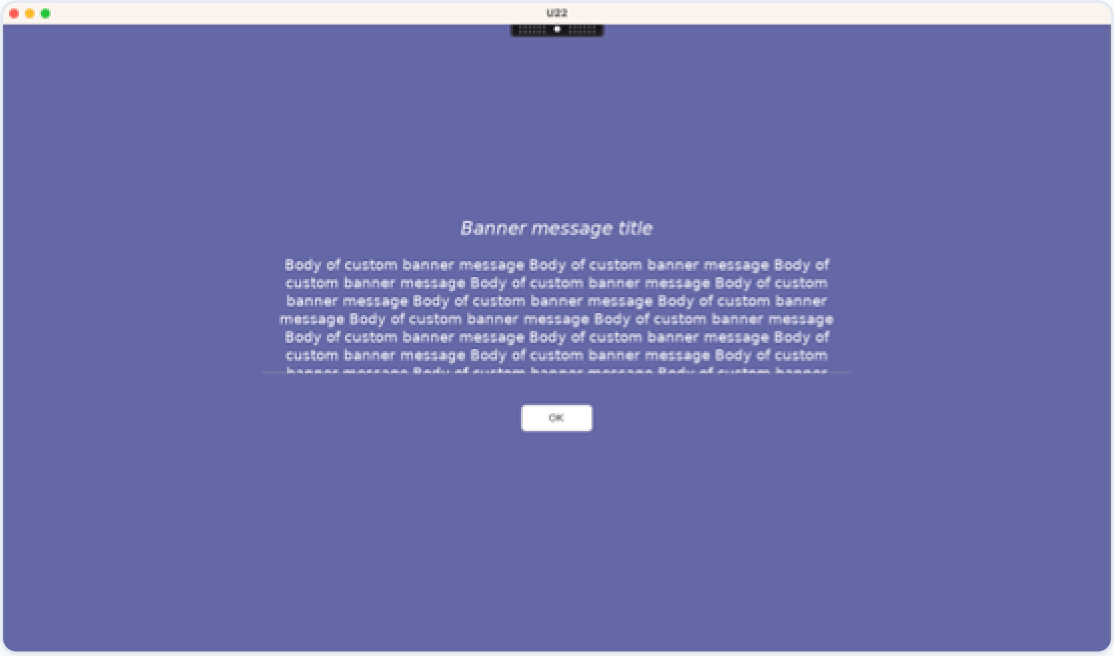
カスタムバナーメッセージの本文を設定するには、次を実行します:

```
/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:

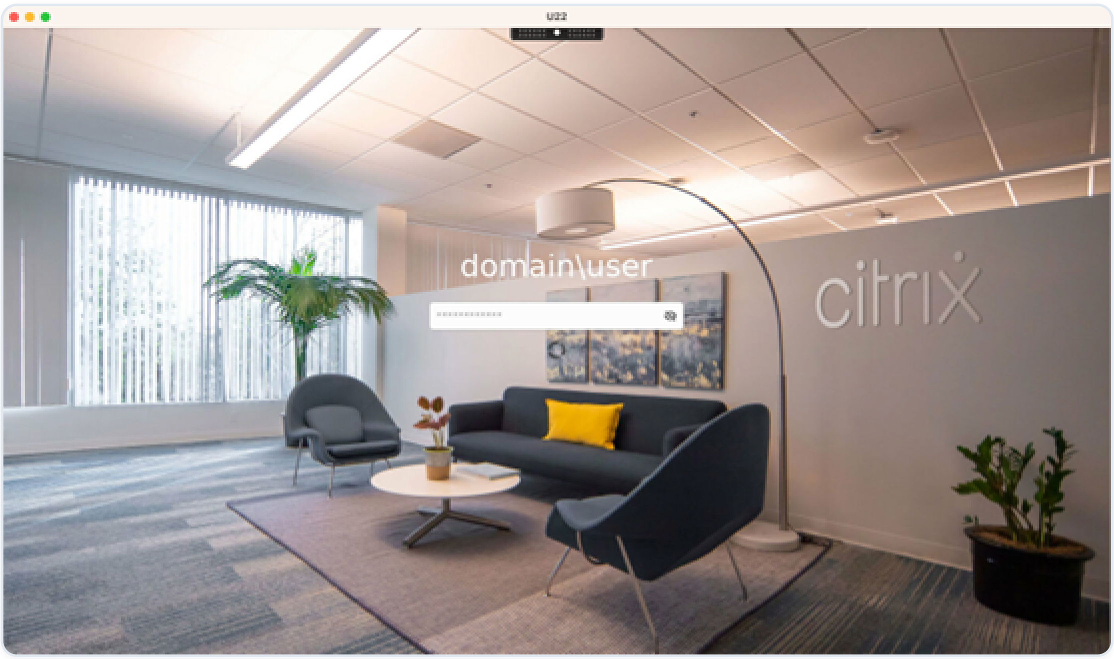


> セッション**ログオン**画面にカスタム背景を追加するには、次を実行します:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"BackgroundImagePath" -d "<path to the background image>" --force
```

カスタム背景を表示するには、セッションユーザーは背景画像の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での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) シナリオでのセッション**ログオン**:

  SSO シナリオでのセッション**ログオン**(/en-us/linux-virtual-delivery-agent/2308/media/session-logon-in-sso-scenarios.png)

  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**:

  一般的な非SSOシナリオでのセッション**ログオン**(/en-us/linux-virtual-delivery-agent/2308/media/session-logon-in-typical-non-sso-scenarios.png)

  - 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:

  非SSOシナリオ: 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(/en-us/linux-virtual-delivery-agent/2308/media/non-sso-scenarios-user-name-and-password-used-for-session-logon.png)

  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: セッションログオンに使用されるスマートカード(/en-us/linux-virtual-delivery-agent/2308/media/non-sso-scenarios-smart-card-used-for-session-logon.png)

  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、非SSO認証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

CTX_XDL_DESKTOP_ENVIRONMENT変数を使用して、セッションユーザーのデスクトップ環境を指定できます。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ます。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、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。

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を示しています。

LINUXディストリビューション	サポートされているデスクトップ
Debian11.3	メイト、グノーム、グノームクラシック、ケイディーイー
RHEL 8.8、RHEL 8.6	メイト、グノーム、グノームクラシック
RHEL 7.9	メイト、グノーム、グノームクラシック、ケイディーイー
ロッキー リナックス 8.8、ロッキー リナックス 8.6	メイト、グノーム、グノームクラシック、ケイディーイー
SUSE 15.4	メイト、グノーム、グノームクラシック
ウブントゥ 22.04、ウブントゥ 20.04	メイト、グノーム、グノームクラシック、ケーディーイー

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	次のコマンドを実行します
GNOME	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME`
グノームクラシック	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME-CLASSIC`
MATE	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh MATE`

目的のデスクトップ環境が次の場合は	次のコマンドを実行します
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では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。その後、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、有効を意味します。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	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "LogNoHome" -d "0x00000001" --force`

レジストリキー	説明	コマンド
	ションログオンは無効になります。	
`HomeMountPoint`	Linux VDA上の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`
`TempHomeDirectoryPath`	マウントポイントが失敗した場合に備えて、Linux VDA上の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	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "TempHomeDirectoryPath" -d "</tmp by default>" --force`

レジストリキー	説明	コマンド
	リは`/tmp/CTXSmf_user_id`です。	
`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は有効を意味します。0は無効を意味します。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "RemoveHomeOnLogoff" -d "0x00000000" --force`

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。
- Publishing applications is supported with Linux VDA Version 1.4 and later. However, the Linux VDA does not support the delivery of desktops and apps to the same machine. To address this issue, we recommend that you create separate delivery groups for app and desktop deliveries.
- 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。

ランデブー V1

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™ 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つの種類があります。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™セッションのトラフィックです。

Rendezvous V1では、HDXセッションのトラフィックが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ためのすべての制御トラフィックをプロキシするには、依然としてCloud Connectorが必要です。

要件

- Citrix Workspace™およびCitrix Gatewayサービスを使用した環境へのアクセス。
- コントロールプレーン: Citrix DaaS (旧Citrix Virtual Apps and Desktops™サービス)。
- 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ホワイトリストに登録できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳細については、Citrix Cloudドキュメントのインターネット接続要件セクション（Virtual Apps and Desktopサービスの下）およびKnowledge Centerの記事CTX270584を参照してください。
- Cloud Connectorは、セッションをブローカリングする際にVDAのFQDNを取得する必要があります。この目的を達成するには、サイトのDNS解決を有効にします。Citrix DaaS Remote PowerShell SDKを使用して、コマンド`Set-BrokerSite -DnsResolutionEnabled $true`を実行します。Citrix DaaS Remote PowerShell SDKの詳細については、「SDKとAPI」を参照してください。

プロキシ構成

VDAは、HTTP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場合は、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、または検査されないように例外を設定してください。そうしないと、接続が切断されます。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、Negotiate認証スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

  > **注:**
  >
  > Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名 (SPN) を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAはセッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。ここで、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に追加の設定は不要です。

非透過型プロキシ

ネットワークで非透過型プロキシを使用する場合は、Rendezvous proxy configuration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

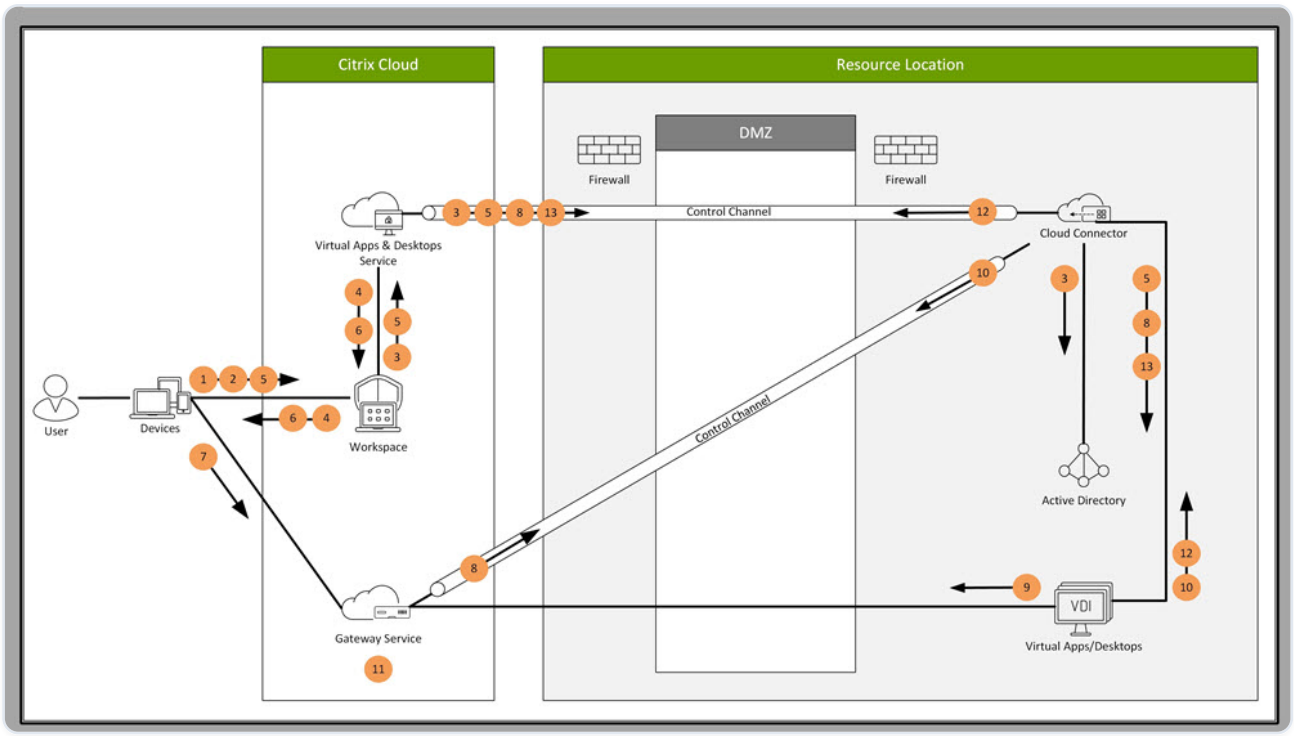
すべての要件を満たしている場合、Rendezvousが使用されているかどうかを検証するには、次の手順に従います。

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. TRANSPORT PROTOCOLS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 HDX セッションを Cloud Connector 経由で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 に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

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™ 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。

There are two types of traffic to consider: 1) control traffic for VDA registration and session brokering; 2) HDX™ session traffic.

Rendezvous V1 allows for HDX session traffic to bypass Cloud Connectors, but it still requires Cloud Connectors to proxy all control traffic for VDA registration and session brokering.

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。

- Access to the environment using Citrix Workspace™ and Citrix Gateway service.
- コントロールプレーン: Citrix DaaS (旧 Citrix Virtual Apps and Desktops™ サービス)。
- VDA バージョン 2201 以降。
  - バージョン2204は、HTTPおよびSOCKS5プロキシに最低限必要なバージョンです。
- 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ホワイトリストに登録できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳細については、Citrix Cloudドキュメントのインターネット接続要件セクション (Virtual Apps and Desktop サービスの下) およびKnowledge Centerの記事CTX270584を参照してください。
- 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セッションの両方のトラフィックに対してプロキシ経由での接続をサポートします。両方の種類のトラフィックの要件と考慮事項は異なるため、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構成してください。そうしないと、接続は失敗します。
- 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構成してください。そうしないと、接続は失敗します。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、Negotiate認証スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

> **注:**
>
> Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名（SPN）を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAは、セッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。この場合、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 で追加の構成は必要ありません。

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ます。
2. VDA のインストール後、次のコマンドを実行して必要なレジストリキーを設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"GctRegistration" -d "0x00000001" --force
```

3. VDA マシンを再起動します。
4. Citrix ポリシーを作成するか、既存のポリシーを編集します。
   - Rendezvous プロトコル設定を に設定します。
   - Citrix ポリシーフィルターが適切に設定されていることを確認します。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 ポリシーが別のポリシーを上書きしないように、正しい優先順位が設定されていることを確認します。

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。

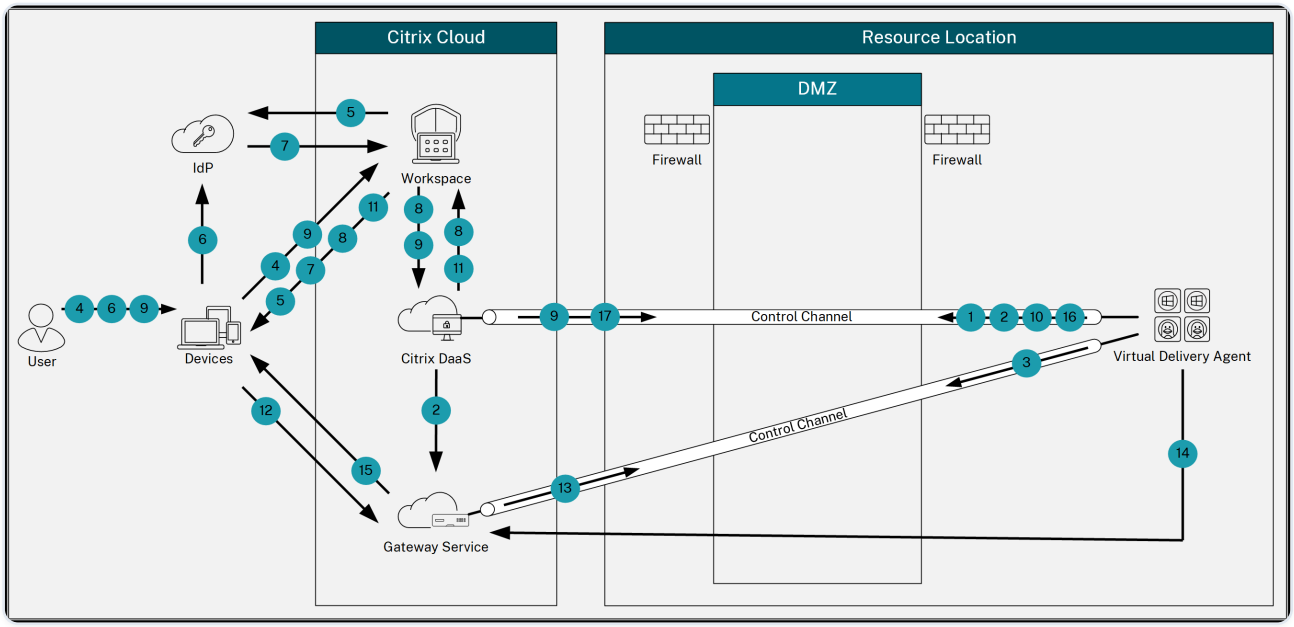
- TCP ランデブー: TCP - TLS - CGP - ICA
- EDT ランデブー: UDP - DTLS - CGP - ICA
- Cloud Connector を介したプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

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 が有効になっている状態で VDA が Citrix Gateway サービスに直接到達できない場合、VDA は HDX セッションを Cloud Connector を介して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ステップを示しています。



1. VDA は Citrix Cloud と 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、認証のためにユーザーを適切な IdP に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成功すると、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™は接続を仲介してVDAにセッションの準備を指示します。
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、認証と接続先リソースの識別のために起動チケットを提供します。
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は適用可能なポリシーをVDAに送信します。

DTLS を使用してユーザーセッションを保護する

DTLS 暗号化は、7.18 リリース以降、完全にサポートされている機能です。デフォルトでは、この機能は Linux VDA で有効になっています。詳細については、「Transport Layer Security」を参照してください。

DTLS 暗号化を有効にする

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、**HDX™ Adaptive Transport** ポリシーが **Preferred** または **Diagnostic mode** に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**/opt/Citrix/VDA/sbin** にある **enable_vdassl.sh** ツールを使用して SSL 暗号化を有効 (または無効) にします。このツールで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -h` コマンドを実行してください。

> **注:**
>
> Linux VDA は DTLS 1.0 と DTLS 1.2 の両方をサポートしており、デフォルトでは DTLS 1.2 を使用します。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てください。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。Citrix Workspace アプリが DTLS 1.0 のみをサポートしている場合 (例: Citrix Receiver for Windows 4.11)、**enable_vdassl.sh** ツールを使用して、**SSLMinVersion** を **TLS_1.0** に、**SSLCipherSuite** を **COM** または **ALL** に設定します。

TLS を使用し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ています。TLS 暗号化はデフォルトで無効になっています。

TLS 暗号化を有効にする

セキュアなユーザーセッション向けに TLS 暗号化を有効にするには、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/2308/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/2308/media/script-for-enabling-ssl.png)

ヒント: サーバー証明書は各 Linux VDA サーバーにインストールする必要があり、ルート証明書は各 Linux VDA サーバーとクライアントにインストールする必要があります。

CONTROLLER で TLS 暗号化を有効にする

> **注:**
>
> TLS 暗号化は、デリバリーグループ全体に対してのみ有効にできます。特定のアプリケーションに対して TLS 暗号化を有効にすることはできません。

> Controller の PowerShell ウィンドウで、ターゲットデリバリーグループの TLS 暗号化を有効にするには、次のコマンドを順に実行します。

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` コマンドを実行することもできます。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、次のコマンドを順に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

トラブルシューティング

公開されたデスクトップセッションにアクセスしようとすると、Windows 用 Citrix Workspace™ アプリで次の「要求されたアドレスを割り当てできません」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の信頼性機能を導入しています。セッションの信頼性はデフォルトで有効になっています。

セッションの信頼性により、ネットワークの中断時にもICAセッションがシームレスに再接続されます。セッションの信頼性の詳細については、「クライアント自動再接続とセッションの信頼性」を参照してください。

注: セッションの信頼性接続を介して送信されるデータは、デフォルトでプレーンテキストです。セキュリティ上の目的から、TLS暗号化を有効にすることをお勧めします。TLS暗号化の詳細については、「TLSを使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studioでのポリシー設定

Citrix Studioでセッションの信頼性について、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、「セッションの信頼性ポリシー設定」および「クライアント自動再接続ポリシー設定」を参照してください。

注: **セッションの信頼性接続**または**セッションの信頼性ポート番号**ポリシーを設定した後、設定を有効にするには、VDAサービスとHDX™サービスをこの順序で再起動してください。

Linux VDAでの設定

- **セッションの信頼性TCPリスナーの有効化/無効化**

デフォルトでは、セッションの信頼性TCPリスナーは有効になっており、ポート2598でリッスンしています。リスナーを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "fEnableWinStation" -d "0x00000000"
```

注: 設定を有効にするには、HDXサービスを再起動してください。TCPリスナーを無効にしても、セッションの信頼性は無効になりません。**セッションの信頼性接続**ポリシーを通じて機能が有効になっている場合、セッションの信頼性は他のリスナー（SSLなど）を通じて引き続き利用できます。

- **セッションの信頼性ポート番号**

次のコマンドを使用して、セッションの信頼性ポート番号を設定することもできます（例としてポート番号2599を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\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。**Citrix Studio**のポリシー設定でポート番号が設定されている場合、Linux VDAでの設定は無視されます。VDA上のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように構成されていることを確認してください。

- **サーバーからクライアントへのキープアライブ間隔**

セッションでアクティビティがない場合（たとえば、マウスの動きや画面の更新がない場合）、Linux VDAとクライアントの間でキープアライブメッセージが送信されます。キープアライブメッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントからの応答がない場合、クライアントが再接続するまでセッションは中断されます。この設定は、連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpServerToClientKeepAlive"  -d  "10"  --force
```

- **クライアントからサーバーへのキープアライブ間隔**

この設定は、ICA®クライアントからLinux VDAに送信される連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpClientToServerKeepAlive"  -d  "10"  --force
```

トラブルシューティング

ポリシー設定でセッションの信頼性を有効にした後、セッションを起動できません。

この問題を回避するには、次の手順を実行します。

1. Citrix Studioのポリシー設定でセッションの信頼性を有効にした後、VDAサービスとHDXサービスがこの順序で再起動されていることを確認してください。
2. VDAで、次のコマンドを実行して、セッションの信頼性TCPリスナーが実行されていることを確認します（例としてポート2598を使用）。

   ```
   netstat -an | grep 2598
   ```

   セッションの信頼性ポートにTCPリスナーがない場合は、次のコマンドを実行してリスナーを有効にします。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations
   \cgp"   -v  "fEnableWinStation" -d "0x00000001"
   ```

セッション録画 (プレビュー)

Linux VDAでホストされているセッションを録画および再生できます。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない場合があり、非実稼働環境での使用が推奨されます。プレビュー機能で見つかった問題は、Citrixテクニカルサポートではサポートされません。

セッション録画の有効化または無効化

Linux VDAのセッション録画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でセッション録画を有効にすると、ユーザーがセッションにログオンしたときに、セッションが録画されていることが通知されます。

録画ファイルのサイズを指定する

録画のサイズが大きくなると、録画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移動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。録画がこの制限に達すると、現在のファイルは閉じられ、録画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**。ファイルが指定されたサイズに達すると現在のファイルが閉じられ、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**。期間に達すると現在のファイルが閉じられ、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時間録画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。

上記の2つの条件のうち、最初に満たされたときにロールオーバーが発生します。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。録画が3時間で17 MBに達すると、セッション録画はファイルを閉じて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。

録画の保存場所を指定する

録画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
```

録画は、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設定し、ユーザー`ctxsrvr`にそのパスへの書き込み権限を付与します。

録画の表示

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と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のドキュメント 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。

仮想チャネルSDK (プレビュー)

Linux VDA 用仮想チャネルソフトウェア開発キット (SDK) を使用すると、VDA 上で実行するサーバー側アプリケーションを作成できます。詳細については、Citrix Virtual Channel SDK for the Linux VDA ドキュメントを参照してください。

Linux VDA 用 Citrix Virtual Channel SDK は、Citrix Virtual Apps and Desktops の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。

> **注:**
>
> この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない場合があり、非実稼働環境での使用が推奨されます。Citrixテクニカルサポートは、プレビュー機能で発見された問題についてはサポートしません。

Wayland (プレビュー)

Linux VDAは、RHEL 9.2/9.0、Rocky Linux 9.2/9.0、およびUbuntu 22.04のGNOMEでWaylandをサポートしています。Waylandでは、以下の機能が完全にテストされています:

- オーディオ
- クリップボード
- クライアントドライブマッピング (CDM)
- 印刷
- USBデバイスリダイレクト

> **注:**
>
> - この機能はプレビュー版です。プレビュー機能は完全にローカライズされていない可能性があり、非本番環境での使用が推奨されます。プレビュー機能で発見された問題は、Citrixテクニカルサポートの**サポート対象外です**。
> - 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はサポートされていません。

Waylandを有効にする

Waylandを使用するには、次のコマンドを実行して、レジストリキー**EnableWayland**を**1**に設定します:

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

デフォルトでは、レジストリキー**EnableWayland**は**0**に設定されており、これはX11が使用されることを意味します。

Waylandが使用されているか確認する

1. Linuxでターミナルウィンドウを開きます。
2. **echo $XDG_SESSION_TYPE** コマンドを実行します。

Waylandが使用されている場合、出力に**'wayland'**が表示されます。

制限事項

Waylandを使用している場合、以下の制限が確認されています:

- クライアントデバイスのキーボードレイアウトは、VDAのキーボードレイアウトと同期されません。
- RHEL 9.1/9.0またはRocky Linux 9.1/9.0でセッションからログオフするのに約20秒かかります。

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 2308のドキュメントに対する重要な変更点を示します。これらの更新に関する通知を受け取るには、RSSフィードを購読してください。

日付	変更	説明
14 Sep 2023	Linux Virtual Delivery Agent 2308の初期リリース	「新機能」を参照してください。