



# Linux Virtual Delivery Agent 2407

## Contents

<b>Linux Virtual Delivery Agent 2407</b>	<b>6</b>
新機能	6
解決された問題	10
既知の問題	10
サードパーティ製品についての通知	14
廃止	14
システム要件	16
インストールの概要	21
インストールの準備	21
簡単インストールによるドメイン参加済み <b>VDA</b> の作成	23
<b>MCS</b> を使用したドメイン非参加 <b>Linux VDA</b> の作成	46
簡単インストールを使用したドメイン非参加 <b>Linux VDA</b> の作成 ( <b>Technical Preview</b> )	59
<b>Machine Creation Services (MCS)</b> を使用した <b>Linux VDA</b> の作成	69
<b>Citrix Provisioning</b> を使用した <b>Linux VDA</b> の作成	93
<b>Citrix DaaS Standard for Azure</b> で <b>Linux VDA</b> を作成	94
<b>Linux VDA</b> の手動インストール	99
<b>Amazon Linux 2</b> 、 <b>RHEL</b> 、および <b>Rocky Linux</b> への <b>Linux VDA</b> の手動インストール	100
<b>SUSE</b> への <b>Linux VDA</b> の手動インストール	134
<b>Ubuntu</b> への <b>Linux VDA</b> の手動インストール	161
<b>Debian</b> への <b>Linux VDA</b> の手動インストール	191
構成	221
管理	222
<b>Linux VDA</b> データ収集プログラム	222

<b>HDX Insight</b>	<b>225</b>
<b>Citrix Telemetry Service</b> との統合	<b>227</b>
<b>Azure</b> を使用した <b>Linux VDA</b> の自動更新	<b>230</b>
<b>Linux VM</b> および <b>Linux</b> セッションのメトリック	<b>233</b>
ログ収集	<b>235</b>
セッションのシャドウ	<b>240</b>
監視サービスデーモン	<b>246</b>
トラブルシューティング	<b>249</b>
その他	<b>258</b>
<b>HTML5</b> 向け <b>Citrix Workspace</b> アプリのサポート	<b>258</b>
<b>Python 3</b> 仮想環境の作成	<b>259</b>
<b>NIS</b> の <b>Active Directory</b> との統合	<b>261</b>
<b>IPv6</b>	<b>266</b>
<b>LDAPS</b>	<b>268</b>
<b>Xauthority</b>	<b>272</b>
認証	<b>275</b>
<b>Azure Active Directory</b> を使用した認証	<b>275</b>
ダブルホップシングルサインオン認証	<b>279</b>
フェデレーション認証サービス	<b>281</b>
<b>FIDO2</b> (プレビュー)	<b>290</b>
<b>SSO</b> 以外の認証	<b>292</b>
スマートカード	<b>292</b>
認証が不要なユーザー (匿名ユーザー) のアクセス	<b>305</b>
ファイル	<b>307</b>

ファイルのコピーと貼り付け	307
ファイル転送	308
グラフィック	312
自動 <b>DPI</b> スケーリング	313
クライアントのバッテリー状態の表示	314
グラフィックの構成と微調整	316
<b>HDX</b> 画面共有	328
マルチモニターサポート	335
非仮想化 <b>GPU</b>	341
セッションウォーターマーク	343
マルチセッション <b>Linux VDA</b> での共有 <b>GPU</b> アクセラレーション	348
システムトレイ	350
<b>Thinwire</b> のプログレッシブ表示	355
一般コンテンツリダイレクト	357
クライアントドライブマッピング	358
<b>USB</b> デバイスリダイレクト	359
クリップボードリダイレクト	368
キーボード	370
<b>X</b> キーボード拡張 ( <b>XKB</b> ) の構成	370
クライアント入力システム ( <b>IME</b> )	371
クライアント <b>IME</b> ユーザーインターフェイスの同期	372
動的なキーボードレイアウトの同期	376
ソフトキーボード	379
多言語入力サポート	382

マルチメディア	384
オーディオ機能	384
ブラウザコンテンツリダイレクト	388
<b>HDX Web</b> カメラビデオ圧縮	398
ドメイン非参加の <b>Linux VDA</b>	403
ポリシーサポート一覧	405
印刷	431
印刷のベストプラクティス	431
<b>PDF</b> 印刷	438
リモート <b>PC</b> アクセス	439
セッション	451
アダプティブトランスポート	452
<b>HDX</b> アダプティブスループット	455
セッションログオン画面のカスタム背景とバナーメッセージ	455
セッションユーザーによるカスタムデスクトップ環境	459
一時的なホームディレクトリを使用したログオン	460
アプリケーションの公開	462
<b>Rendezvous V1</b>	464
<b>Rendezvous V2</b>	468
<b>Secure HDX (Technical Preview)</b>	473
<b>DTLS</b> によるユーザーセッションの保護	474
<b>TLS</b> によるユーザーセッションの保護	475
セッション画面の保持	478
セッションの録画 (プレビュー)	481

仮想チャネル <b>SDK</b> (プレビュー)	<b>483</b>
<b>Wayland</b> (プレビュー)	<b>484</b>
ベストプラクティス	<b>485</b>
<b>Google Cloud Platform (GCP)</b> で <b>Machine Creation Services (MCS)</b> を使用した <b>Linux VDA</b> の作成	<b>485</b>
<b>Ansible</b> を使用した展開の管理	<b>506</b>

## Linux Virtual Delivery Agent 2407

August 20, 2024

**重要:**

最新リリース (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は、[Lifecycle Milestones](#) で説明しています。

Linux Virtual Delivery Agent (VDA) によって、場所を問わず、Citrix Workspace アプリがインストールされたどのデバイスからでも、Linux 仮想アプリおよびデスクトップにアクセスできるようになります。

[サポートされているディストリビューション](#) をベースとした仮想アプリおよび仮想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 を構成します。次に、Citrix Studio を使ってユーザーがアプリおよびデスクトップを使用できるようにします。

### 新機能

August 20, 2024

#### 2407 の新機能

##### **RHEL 9.4 / 8.10、Rocky Linux 9.4 / 8.10、Debian 12.5 / 11.9** のサポート

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

- RHEL 9.4
- Rocky Linux 9.4
- RHEL 8.10
- Rocky Linux 8.10
- Debian 12.5
- Debian 11.9

サポートされている Linux ディストリビューションについて詳しくは、「[システム要件](#)」を参照してください。

#### 新しい .NET ランタイムが必要

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

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

### トークンベースの登録をオンプレミス環境に拡張 (**Technical Preview**)

このリリースでは、トークンベースの登録機能がオンプレミス環境に拡張されます。簡単インストールでドメイン非参加 VDA を作成する場合、次のことが可能になります：

- ドメイン非参加の VDA を、クラウドとオンプレミスの両方の Delivery Controller のマシンカタログに登録する。
- Citrix Web Studio を使用して空のマシンカタログを作成し、登録トークンを生成する。

詳しくは、「[簡単インストールを使用したドメイン非参加 Linux VDA の作成 \(Technical Preview\)](#)」を参照してください。

### **SSSD** をデフォルトのドメイン参加方法として指定

SSSD は、Winbind に代わって、ドメインに参加するためのデフォルトの方法になりました。この変更によって、`ctxsetup.sh` または簡単インストールスクリプト (`ctxinstall.sh`) を対話モードで実行するときの構成の動作が影響を受けます。詳しくは、「[簡単インストールによるドメイン参加済み VDA の作成](#)」を参照してください。

### アップグレード中に構成ファイルの処理をパッケージマネージャーに委任

このリリースでは、アップグレード中に構成ファイルを処理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。さらに、このリリースでは、RPMLint によってフラグが付けられたいくつかの問題に対処し、RPM パッケージを再構築して、より適切に RPMLint の要件に準拠するようにしています。詳しくは、「[手順 13: Linux VDA のアップグレード \(オプション\)](#)」を参照してください。

### スマートカードの取り出し設定をサポート

Linux VDA で次のレジストリキーを編集することで、セッション中にサインインしたユーザーのスマートカードがスマートカードリーダーから取り出されたときに何が起こるかを決定できるようになりました。詳しくは、「[スマートカードの取り出し設定](#)」を参照してください。



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

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

オーディオの損失耐性モード (**Technical Preview**)

Enlightened Data Transport (EDT) 損失耐性プロトコルでオーディオがサポートされるようになりました。この機能により、ユーザーが遅延やパケット損失が大きいネットワーク経由で接続している場合に、リアルタイムストリーミングのユーザーエクスペリエンスが向上します。この機能が有効になっている場合、Citrix Virtual Apps and Desktops のアダプティブトランスポートは、オーディオエクスペリエンスを向上させるために EDT 損失耐性トランスポートプロトコルを使用します。この機能はデフォルトでは無効になっています。詳しくは、「[オーディオ機能](#)」記事の「[オーディオの損失耐性モード](#)」を参照してください。

オーディオパフォーマンスを向上させるために実装されたパケット損失補間 (**Technical Preview**)

2407 リリース以降では、ジッターバッファメカニズムが向上し、アダプティブオーディオコーデックにパケット損失補間 (PLC) が追加されました。PLC は、失われたデータパケットの再構築に役立ちます。この機能強化により、パケット損失耐性とジッター耐性が向上し、損失耐性モード (EDT Lossy) でのオーディオパフォーマンスが向上します。

この機能を有効にするには、[オーディオの損失耐性モード](#)機能も有効にする必要があります。詳しくは、「[オーディオ機能](#)」記事の「[オーディオパフォーマンスを向上させるパケット損失補間 \(Technical Preview\)](#)」を参照してください。

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

Linux VDA は Debian 12.5、RHEL 9.x、Rocky Linux 9.x で PipeWire をサポートするようになり、PipeWire はデフォルトのオーディオサービスとなっています。詳しくは、「[オーディオ機能](#)」を参照してください。

強化されたグラフィックパフォーマンス

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

ログイン **UI** に表示されるログインプロセスの詳細

セッションユーザーは、Linux VDA セッションを開いたときに、ログイン UI で次のログインプロセスの詳細を確認できるようになりました:

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

#### **KDE** および **XFCE** デスクトップ環境におけるキーボードレイアウト同期のサポート

このリリースでは、動的なキーボードレイアウト同期機能が XFCE および KDE Plasma 5 デスクトップ環境に拡張されています。詳しくは、「[動的なキーボードレイアウトの同期](#)」を参照してください。

#### **evdev** をデフォルトの **X** キーボード拡張 (**XKB**) 規則にする

より応答性が高く予測可能なキーボード動作による優れたユーザーエクスペリエンスを実現するために、Linux VDA では、キーボード構成のデフォルトの XKB 規則として evdev が指定されるようになりました。

詳しくは、「[X キーボード拡張 \(XKB\) の構成](#)」を参照してください。

#### **LDAP** クエリのパフォーマンス最適化

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

#### **Cloud Connector** の **VDA** のスケーラビリティが向上

このリリースでは、VDA のスケーラビリティが向上し、Citrix Cloud Connector (4 つの仮想 CPU、10GB のメモリ) ごとに 6,000 の Linux VDA をサポートできるようになりました。詳しくは、「[Cloud Connector のサイズおよびスケールの考慮事項](#)」を参照してください。

#### **HDX** アダプティブスループットがデフォルトで有効化

詳しくは、「[HDX アダプティブスループット](#)」を参照してください。

注:

このリリースから、展開管理に Ansible を使用することに関心のあるお客様向けに、総合的なベストプラクティスに関する記事を提供します。詳しくは、「[Ansible を使用した展開の管理](#)」を参照してください。

#### 以前のリリースの新機能

1912 LTSR~2402 LTSR の後に出荷されたリリースの新機能については、「[新機能の履歴](#)」を参照してください。

## 解決された問題

August 20, 2024

Linux Virtual Delivery Agent 2407 には次の修正された問題があります:

- ルートユーザーは、CDM を使用してマップされたクライアントデバイスに読み取り専用ファイルをコピーできません。[CVADHELP-25124]
- グループポリシーパーサーは、GPF ファイルからのサイズの大きなグループポリシーセットを処理するときに、CPU を 100% 消費する可能性があります。[CVADHELP-25323]
- セッションが切断されると ctxnsap プロセスが予期せず終了し、Citrix Application Delivery Management (ADM) が HDX Insight 情報を表示できなくなります。[CVADHELP-25493]
- テンプレートマシンで誤った ad\_join.sh を実行すると、mcs\_setting\_flag というフラグが生成され、新しく MCS で作成されたマシンがドメインに参加できなくなります。この修正プログラムにより、deploymcs.sh スクリプトにチェックインが導入され、フラグが検出された場合はそれを識別して削除するようになりました。[CVADHELP-25117]

## 既知の問題

August 20, 2024

このリリースでは、次の問題が確認されています:

- スマートカードサービスは、スマートカード認証中にファイル記述子をリークし、新しいスマートカードへのアクセスをブロックします。この問題は、ほとんどの Linux ディストリビューションがデフォルトで、プロセスごとに開くファイルの最大数を 1,024 に制限しているために発生します。スマートカードサービスがこの制限に達すると、新しい接続を確立できなくなり、その後のスマートカードアクセスが事実上ブロックされます。

---

この問題は、スマートカードログオンが有効になっている `grep PID**`。  
る VDA に影響します。現象としては、

**/var/log/xdl/hdx.log** で多数の **Failed to accept new connection: Too many open files** エラーが発生し、**/proc/\${pid}/fd/** でファイル記述子が蓄積されます。この **\${pid}** は、**ctxscardsd** のプロセス ID を表します。PID を確認するには、次のコマンドを使用します: **\*\*systemctl status ctxscardsd**

---

この問題を軽減するには、スマートカードサービスのオープンファイルの最大数制限を増やすか、スマートカードサービスを再起動します。サービスを再起動する前に、アクティブなセッションがないことを確認してください。制限を増やすか、サービスを再起動するには、次のコマンドを使用します：

- スマートカードサービスを再起動するには：

```
1 systemctl restart ctxscardsd
```

- 現在のサービスに関してオープンファイルの最大数を照会するには：

```
1 cat /proc/${
2 PID }
3 /limits
```

- スマートカードサービスのオープンファイルの最大数を設定するには：

1. 現在の設定を確認するには、**ctxscardsd.service** ファイルを読み取り専用モードで開きます：

```
1 vim -R /lib/systemd/system/ctxscardsd.service
```

2. 制限を増やすには、**ctxscardsd.service** の **Service** セクションに次の行を追加します：

```
1 LimitNOFILE=65536
```

3. **systemd** デーモンを再読み込みし、**ctxscardsd** サービスを再起動します：

```
1 systemctl daemon-reload
2 systemctl restart ctxscardsd
```

4. 新しい制限を確認します：

```
1 cat /proc/${
2 PID }
3 /limits
```

注：

オープンファイルの最大数を増やすと、ファイル記述子が不足するまでの時間を延長できますが、最終的には **ctxscardsd** の再起動が必要になる可能性があります。

[LNXVDA-17784]

- CTX\_XDL\_DESKTOP\_ENVIRONMENT 変数を指定しないままにすると、その後のプロセスでどのデスクトップが使用されるかを知ることができなくなります。その結果、デスクトップ関連の環境変数が構成されず、これらの変数に依存するアプリやプラグインが正常に動作しない可能性があります。[LNXVDA-16212]
- GNOME の問題により、RHEL 8.X、Rocky Linux 8.x、RHEL 9.x、および Rocky Linux 9.x で **samba-winbind** をバージョン 4.18.6 にアップグレードすると、Linux VDA が正常に動作しません。詳しくは、<https://issues.redhat.com/browse/RHEL-17122>を参照してください。

- セッション起動エラーは、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には不十分な場合に発生します。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** で次の LDAP 例外が発生して、VDA 登録が失敗する場合があります：

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

この問題を解決するには、以下の手順に従います。

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

```
1 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 変数を使用して、または次のコマンドを使用して検索ベースを設定できます：

```
1 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
```

[CVADHELP-20895]

- 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。この更新プログラムにより、ドメインへの参加と登録に障害が発生します。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。
- 暗号化の種類 **RC4\_HMAC\_MD5** を Kerberos に許可している場合、Linux VDA を Controller に登録できず、次のエラーメッセージが表示されることがあります：

**Error: Failure unspecified at GSS-API level (Mechanism level: Encryption type RC4 with HMAC is not supported/enabled)**

この問題に対応するには、Active Directory ドメイン（具体的には **OU**）で **RC4\_HMAC\_MD5** を無効にするか、Linux VDA で弱い暗号化の種類を許可してください。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：

1. グループポリシー管理コンソールを開きます。
2. 対象のドメインを見つけて、[既定のドメインポリシー] を選択します。
3. [既定のドメインポリシー] を右クリックし、[編集] 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。

4. [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。
5. [ネットワークセキュリティ: Kerberos で許可する暗号化の種類を構成する] をダブルクリックします。
6. [DES\_CBC\_CRC]、[DES\_CBC\_MD5]、[RC4\_HMAC\_MD5] のチェックボックスをオフにして、[AES128\_HMAC\_SHA1]、[AES256\_HMAC\_SHA1]、[将来使用する暗号化の種類] をオンにします。

Linux VDA で弱い暗号化の種類を許可するには、次の手順を実行します：

注：

暗号化の種類が弱いと、展開は攻撃に対して脆弱になります。

1. Linux VDA で/etc/krb5.conf ファイルを開きます。
2. [libdefaults] セクションに次の設定を追加します。

```
allow_weak_crypto= TRUE
```

- Linux VDA では、暗号化で SecureICA はサポートされていません。Linux VDA で SecureICA を有効にすると、セッションの起動に失敗します。
- GNOME デスクトップセッションでは、キーボードレイアウトを変更しようとする場合がありま  
す。[CVADHELP-15639]
- Ubuntu のグラフィック：HDX 3D Pro で、Desktop Viewer をサイズ変更した後、アプリケーションの周  
囲に黒い枠が表示されたり、まれに背景が黒く表示される場合があります。
- Linux VDA 印刷リダイレクトで作成されたプリンターは、セッションからログアウト後、削除されることがあ  
ります。
- ディレクトリにファイルやサブディレクトリが多数含まれているときに、CDM ファイルが欠落します。クラ  
イアント側のファイルやディレクトリが非常に多い場合、この問題が生じることがあります。
- このリリースでは、英語以外の言語では UTF-8 エンコードのみがサポートされます。
- セッションのローミング時、Android 向け Citrix Workspace アプリで CapsLock が通常とは反対の状  
態になる場合があります。Android 向け Citrix Workspace アプリへの既存の接続をローミングすると、  
CapsLock 状態が失われる場合があります。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 向け Citrix Workspace アプリを使用して Linux VDA に接続している場合、Alt キーを使用するシ  
ョートカットキーが機能しないことがあります。Mac 向け Citrix Workspace アプリでは、左右どちらの  
option/alt キーを押しても、デフォルトでは AltGr が送信されます。Citrix Workspace アプリの設定でこ  
の動作を変更することはできますが、結果はアプリケーションによって異なります。
- Linux VDA をドメインに再度追加すると、登録できません。再度追加することにより、Kerberos キーの申し  
いセットが生成されます。しかし、ブローカーは、Kerberos キーの以前のセットに基づいた、キャッシュに

存在する期限切れの VDA サービスチケットを使用する可能性があります。VDA がブローカーに接続しようとするときに、ブローカーは VDA に返すセキュリティコンテキストを確立できないことがあります。通常見られる現象は、VDA 登録の失敗です。

この問題は、VDA サービスチケットが最終的に期限切れとなって更新されると自動的に解決します。ただし、サービスチケットの期限は長いので、それまでに時間がかかることがあります。

この問題を回避するには、ブローカーのチケットキャッシュを消去します。ブローカーを再起動するか、管理者としてコマンドプロンプトからブローカーで次のコマンドを実行します。

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

このコマンドにより、Citrix Broker Service を実行する Network Service プリンシパルが LSA キャッシュに保持するサービスチケットはすべて削除されます。これにより、ほかの VDA のサービスチケットが削除されます。また、その他のサービスのサービスチケットも削除される可能性があります。ただし、この処理は悪影響を及ぼしません。これらのサービスチケットは、再度必要になった時に 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がサポートされません。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開始後にキャプチャデバイスを接続した場合は、アプリケーションが応答しなくなって再起動する必要が生じる可能性があります。録音中にキャプチャデバイスが取り外されると、同様の問題が発生する可能性があります。
- Windows 向け Citrix Workspace アプリでオーディオ録音中にオーディオの歪みが生じることがあります。

## サードパーティ製品についての通知

August 20, 2024

[Linux Virtual Delivery Agent バージョン 2407](#) (PDF ダウンロード)

Linux VDA のこのリリースには、ドキュメント内で定義された条件の下でライセンスが有効になったサードパーティのソフトウェアが含まれている可能性があります。

## 廃止

August 20, 2024

この記事の告知は、お客様が適宜ビジネス上の決定を下せるように、段階的に廃止されるプラットフォーム、Citrix 製品、機能について前もってお知らせするためのものです。Citrix ではお客様の使用状況とフィードバックをチェックして、各プラットフォーム、Citrix 製品、機能を撤廃するかどうかを判断しています。お知らせする内容は以降のリリースで変わることがあり、廃止される機能がすべて含まれるわけではありません。

製品ライフサイクルサポートについて詳しくは、「[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。

## 廃止と削除

廃止または削除されるプラットフォーム、Citrix 製品、機能を以下の表に示します。

廃止されたアイテムはすぐには削除されません。このリリースでは Citrix が引き続きサポートしていますが、今後のリリースでは削除される予定です。

削除されたアイテムは Linux VDA で削除されたか、サポートされなくなりました。

---

アイテム	廃止が発表されたリリース	削除されたリリース
RHEL 9.3、RHEL 8.9 のサポート	2407	2411
Rocky Linux 9.3、Rocky Linux 8.9 のサポート	2407	2411
Ubuntu 20.04 のサポート	2407	2502
SUSE 15.5 のサポート	2407	2411
Debian 11.7/11.3 のサポート	2407	2407
RHEL 9.0、RHEL 8.6 のサポート	2402	2407
Rocky Linux 9.0、Rocky Linux 8.6 のサポート	2402	2407
RHEL 7.9、CentOS 7.9 のサポート	2402	2407
SUSE 15.4 のサポート	2308	2311
Rocky Linux 9.1、Rocky Linux 8.7 のサポート	2305	2308
RHEL 9.1、RHEL 8.7 のサポート	2305	2308
RHEL 8.4 のサポート	2303	2308
Ubuntu 18.04 のサポート	2212	2305
SUSE 15.3 のサポート	2210	2301
Debian 10.9 のサポート	2206	2210
SUSE 15.2 のサポート	2206	2209
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



アイテム	廃止が発表されたリリース	削除されたリリース
SUSE 12.5 のサポート	2109	2204
Ubuntu 16.04 のサポート	2109	2203
RHEL 7.7、CentOS 7.7 のサポート	2006	2009
SUSE 12.3 のサポート	2006	2006
RHEL 6.10、CentOS 6.10 のサポート	2003	2003
RHEL 6.9、CentOS 6.9 のサポート	1909	1909
RHEL 7.5、CentOS 7.5 のサポート	1903	1903
RHEL 7.4、CentOS 7.4 のサポート	1811	1811
RHEL 6.8、CentOS 6.8 のサポート	1811	1811
RHEL 7.3、CentOS 7.3 のサポート	7.18	7.18
RHEL 6.6、CentOS 6.6 のサポート	7.16	7.16
SUSE 11.4	7.16	7.16

---

## システム要件

August 20, 2024

Linux VDA の最新リリースは、Citrix Virtual Apps and Desktops に対応しています。また、ライフサイクルの終わりにまだ達していない、以前のバージョンの Citrix Virtual Apps and Desktops との後方互換性もあります。Citrix 製品のライフサイクル、および製品のバージョンごとのサポートが停止される時期について詳しくは、[Citrix 製品マトリックス](#)を参照してください。

Linux VDA の構成手順は、Windows VDA と多少異なります。Delivery Controller ファームは Windows デスクトップと Linux デスクトップを両方とも仲介できます。

このトピックで説明されていないシステム要件コンポーネント（Citrix Workspace アプリなど）については、各コンポーネントのドキュメントを参照してください。

長期サービスリリース（LTSR）環境での最新リリース（CR）の使用について、およびその他のよくある質問については、[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、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、デスクトップ環境については、次の表を参照してください。詳しくは、「[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	サポートされるデスクトップ
Amazon Linux 2	1.20	MATE
Debian 12.5	1.20	GNOME、GNOME クラシック、KDE、MATE、 <b>Xfce</b> *
Debian 11.9	1.20	GNOME、GNOME クラシック、KDE、MATE
RHEL 9.4/9.3/9.2	1.20	GNOME、 <b>Xfce</b>
RHEL 8.10/8.9/8.8	1.20	GNOME、GNOME クラシック、MATE、 <b>Xfce</b>
Rocky Linux 9.4/9.3/9.2	1.20	GNOME、 <b>Xfce</b> *
Rocky Linux 8.10/8.9/8.8	1.20	GNOME、GNOME クラシック、KDE、MATE、 <b>Xfce</b>
SUSE 15.5	1.20	GNOME、GNOME クラシック、MATE
Ubuntu 22.04	1.21	GNOME、GNOME クラシック、KDE、MATE、 <b>Xfce</b>
Ubuntu 20.04	1.20	GNOME、GNOME クラシック、KDE、MATE、 <b>Xfce</b>

注:

- ご利用の OS ベンダーのサポートが期限切れになると、問題の修正において Citrix の機能が制限される場合があります。廃止された、または削除されたプラットフォームについては、「[廃止](#)」を参照してください。
- 1 つまたは複数のデスクトップをインストールする必要があります。ctxinstall.sh または ctxsetup.sh スクリプトを使用して、セッションで使用する GNOME、GNOME クラシック、または MATE デスクトップ環境を指定できます。
- [Red Hat Enterprise Linux のドキュメント](#)によると、GNOME は RHEL 9 で利用できる唯一のデスクトップ環境です。ただし、EPEL リポジトリがインストールされている場合は、RHEL 9 および Rocky Linux 9 でも **Xfce** デスクトップ環境を使用できます。
- 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のユーザー名の構文について詳しくは、[User/Group Name Syntax](#)を参照してください。

### .NET の要件

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

Linux ディストリビューションに必要な.NET バージョンが含まれている場合は、組み込みフィールドからインストールします。それ以外の場合は、Microsoft パッケージフィードから.NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>を参照してください。

### サポートされるホストプラットフォームおよび仮想化環境

- ベアメタルサーバー
- Amazon Web Services (AWS)
- XenServer (旧称 Citrix Hypervisor)
- Google Cloud Platform (GCP)
- カーネルベースの仮想マシン (KVM)
- Microsoft Azure
- Microsoft Hyper-V
- VMware vSphere Hypervisor
- Nutanix AHV

#### 注:

すべての場合で、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、Microsoft Azure、AWS、および GCP で Linux VDA をホストすることは、Citrix DaaS (旧称 Citrix Virtual Apps and Desktops サービス) でサポートされていました。これらのパブリッククラウドホスト接続を Citrix Virtual Apps and Desktops 展開環境に追加する場合は、Citrix Universal ハイブリッドマルチクラウド (HMC) ライセンスが必要です。

### Active Directory 統合パッケージ

Linux VDA では、以下の Active Directory 統合パッケージおよび製品がサポートされています:

	Winbind	SSSD	Centrify	PBIS	Quest
Amazon Linux 2	はい	はい	はい	はい	いいえ
Debian 12.5/11.9	はい	はい	はい	はい	はい
RHEL 9.4/9.3/9.2、Rocky Linux 9.4/9.3/9.2/8.10/8.9/8.8	はい	はい	いいえ	いいえ	はい (Quest v4.1 以降)
RHEL 8.10/8.9/8.8	はい	はい	はい	はい	はい (Quest v4.1 以降)
SUSE 15.5	はい	はい	はい	はい	はい
Ubuntu 22.04/20.04	はい	はい	はい	はい	はい (Quest v4.1 以降)

### Cloud Connector のサイズおよびスケールの考慮事項

Citrix Cloud Connector を使用して Linux VDA をコントロールプレーンに接続する場合は、Citrix の内部テストに基づいて次の点を考慮してください:

- 各 Citrix Cloud Connector (4 つの仮想 CPU、10GB のメモリ) は、6,000 の Linux VDA をサポートできます。
- 高可用性を実現するために、各 [リソースの場所](#) に 2 つの Cloud Connector を展開し、各リソースの場所に最大 6,000 の Linux VDA を展開します。

### データベースに関する考慮事項

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合は、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

## HDX 3D Pro

Citrix Virtual Apps and Desktops の HDX 3D Pro を使用すると、グラフィック処理装置 (GPU) によるハードウェアアクセラレーションで最高の性能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 は次のハイパーバイザーと互換性があります：

- XenServer (旧称 Citrix Hypervisor)
- VMware vSphere Hypervisor
- Nutanix AHV
- Microsoft Azure
- Amazon Web Services (AWS)
- Google Cloud Platform (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 の場合、HDX 3D Pro は次の種類の GPU をサポートします：

**NVIDIA vGPU** Linux ディストリビューションがサポートする NVIDIA GPU カードを確認するには、[NVIDIA 製品サポートマトリックス](#)に移動し、ハイパーバイザーまたはベアメタル **OS**、ソフトウェア製品の展開、ハードウェアサポート、およびゲスト **OS** サポートの列を確認してください。

GPU カード用の最新の vGPU ドライバーをインストールしていることを確認してください。現在、Linux VDA は vGPU 17 までをサポートしています。詳しくは、「[NVIDIA Virtual GPU Software Supported GPUs](#)」を参照してください。

**非仮想化 GPU** Linux VDA のドキュメントでは、非仮想化 GPU とは次を指します：

- リモート PC アクセスのシナリオで使用される GPU
- ハイパーバイザーから渡された 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** に設定して HDX 3D Pro を有効にできます。追加の構成は必要ありません。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がデフォルトで有効になります。

ヒント:

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

## インストールの概要

July 16, 2024

このセクションでは、次の手順について説明します:

- [インストールの準備](#)
- [簡単インストールによるドメイン参加済み VDA の作成](#)
- [MCS を使用したドメイン非参加 Linux VDA の作成](#)
- [簡単インストールを使用したドメイン非参加 Linux VDA の作成 \(Technical Preview\)](#)
- [MCS を使用した Linux VDA の作成](#)
- [Citrix Provisioning を使用した Linux VDA の作成](#)
- [Citrix DaaS Standard for Azure で Linux VDA を作成](#)
- [Linux VDA の手動インストール](#)
  - [Amazon Linux 2、CentOS、RHEL、および Rocky Linux への Linux VDA の手動インストール](#)
  - [SUSE への Linux VDA の手動インストール](#)
  - [Ubuntu への Linux VDA の手動インストール](#)
  - [Debian への Linux VDA の手動インストール](#)

## インストールの準備

July 16, 2024

この記事では、特定の状況に基づいた適切なインストール方法をどのように選択するかを示し、関連するインストールガイドを紹介します。Citrix Virtual Apps and Desktops および Citrix DaaS の展開の計画について詳しくは、「[インストールの準備](#)」および「[はじめに：展開の計画と構築](#)」を参照してください。

VDA のインストールを開始する前に、次の点に注意してください：

- [Linux VDA のシステム要件](#)を確認して、ターゲット展開が正式にサポートされていることを確認します。
- VDA をドメインに接続するかどうかを決定します。
- プロビジョニング方法とドメイン参加の決定に基づいて、次の表から適切なリンクを確認してください：

	簡単なインストールスクリプト	Machine Creation Services (MCS)	Citrix Provisioning	完全な手動	注
ドメイン参加	<a href="#">簡単インストールによるドメイン参加済み VDA の作成</a>	<a href="#">Machine Creation Services (MCS) を使用した Linux VDA の作成</a>	<a href="#">Citrix Provisioning を使用した Linux VDA の作成</a>	<a href="#">Linux VDA の手動インストール</a>	-
ドメイン非参加	<a href="#">簡単インストールを使用したドメイン非参加 Linux VDA の作成 (Technical Preview)</a>	<a href="#">MCS を使用したドメイン非参加 Linux VDA の作成</a>	未サポート	未サポート	DaaS のみ

注：

- MCS は、リモート PC アクセスのユースケースではサポートされていません。
- 一括プロビジョニングの場合は、MCS、Citrix Provisioning、またはサードパーティの自動化を独自に使用できます。サードパーティの自動化の場合、簡単なインストール スクリプトを組み込む (推奨) か、スクリプト内で完全な手動インストール手順を自動化するかを選択できます。
- 単一の仮想マシンで概念実証 (POC) を迅速に行うには、簡単インストールを使用することをお勧めします。
- 簡単インストールスクリプト (ctxinstall.sh -s) を実行して、CTX\_XDL\_DDC\_LIST や CTX\_XDL\_LDAP\_LIST のような環境変数をすばやく更新することもできます。

## 簡単インストールによるドメイン参加済み VDA の作成

August 21, 2024

**重要:**

- 新規インストールの場合、簡単インストールについてはこの記事参照することをお勧めします。この記事では、簡単インストールを使用して Linux VDA をインストールおよび構成する方法について説明します。簡単インストールは時間と労力を節約するだけでなく、手動のインストールよりもエラーを減らすことができます。必要なパッケージをインストールして、構成ファイルを自動的にカスタマイズすることで、Linux VDA の実行環境をセットアップできます。
- ドメイン非参加の VDA を作成する場合、Machine Creation Services (MCS) と簡単インストールの両方を使用できます。詳しくは、「[MCS を使用したドメイン非参加 Linux VDA の作成](#)」および「[簡単インストールを使用したドメイン非参加 Linux VDA の作成 \(Technical Preview\)](#)」を参照してください。
- ドメイン非参加の VDA で利用可能な機能について詳しくは、「[ドメイン非参加の VDA](#)」を参照してください。

### 手順 1: 構成ファイル情報および Linux マシンの準備

簡単インストールに必要な以下の構成情報を収集します。

- ホスト名 - Linux VDA がインストールされ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。
- 領域名 - Kerberos 領域名。
- ドメインの完全修飾ドメイン名 (FQDN)。
- Active Directory (AD) 統合方法 - 現在、簡単インストールは SSSD、Winbind、Centrify、PBIS および Quest をサポートしています。簡単インストールは、RHEL および Rocky Linux の Quest のみをサポートします。
- ユーザー名 - マシンをドメインに参加させるユーザーの名前。
- パスワード - マシンをドメインに参加させるユーザーのパスワード。
- OU - 組織単位。オプションです。

**重要:**

- Linux VDA をインストールするには、Linux マシンでリポジトリが正しく追加されていることを確認します。
- セッションを起動するには、X Window システムおよびデスクトップ環境がインストールされていることを確認します。



- セキュリティ上の理由で、簡単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。簡単インストールスクリプト (ctxinstall.sh) を対話モードで実行するたびに、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、**/Citrix/VDA/sbin/ctxinstall.conf** でドメイン参加パスワードを設定する、またはパスワードをエクスポートする必要があります。ドメイン参加には管理者アカウントを使用しないことをお勧めします。代わりに、管理者アカウント以外の Active Directory ユーザーにドメイン参加権限を委任してください。このためには、制御の委任ウィザードを使用して、ドメインコントローラーの制御を委任します。

## 手順 2: ハイパーバイザーの準備

サポートされるハイパーバイザー上で仮想マシンとして Linux VDA を実行する場合、いくつかの変更が必要です。使用するハイパーバイザーのプラットフォームに合わせて、次の変更を行います。ベアメタルハードウェアで Linux マシンを実行する場合、変更は必要ありません。

### XenServer (旧称 Citrix Hypervisor) での時刻同期の修正

XenServer の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP と XenServer の問題が発生します。これは、NTP と Citrix Hypervisor の両方がシステムの時間を管理しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは、変更は必要ありません。

XenServer 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、XenServer の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
1 su -
2
3 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いるため、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います。こ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、以下の手順は必要ありません。

時刻同期機能が有効になっている場合は、ファイルに「1」と書き込んで無効にします:

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

この変更を永続化し、再起動後も保持するには、**/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ため、次のようにしてシステムを再起動します:

```
1 su -  
2  
3 cat /proc/sys/xen/independent_wallclock
```

このコマンドは 1 を返します。

### Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想ホストでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともに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で、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、[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:

この方法は VMware および XenServer (旧称 Citrix Hypervisor) の場合とは異なります。VMware および XenServer では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期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な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーで問題が発生します。これは、NTP とハイパーバイザーの両方がシステムの時間を同期しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools をインストールした状態で準仮想化 Linux カーネルを実行している場合は、次の操作を行います。

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. [仮想マシンのプロパティ] ダイアログボックスで、[オプション] タブをクリックします。
4. [VMware Tools] を選択します。
5. [詳細] ボックスで、[ホストとゲスト時刻を同期] チェックボックスをオフにします。

### 手順 3: .NET のインストール

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

Linux 2 にはバージョン 6 が必要です。他のディストリビューションにはバージョン 8 が必要です。

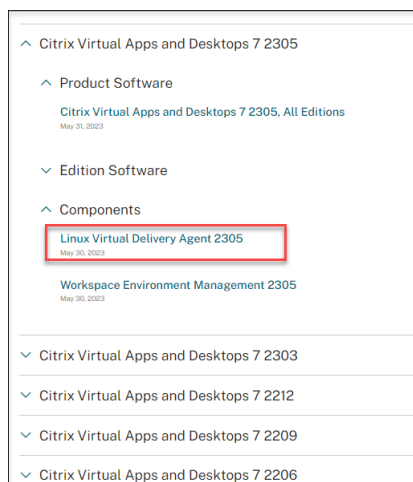
Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

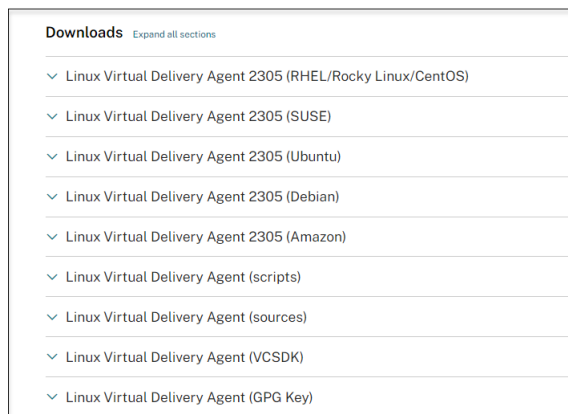
コマンド出力に基づいて、.NET ランタイムのバイナリパスを設定します。たとえば、コマンド出力が /aa/bb/dotnet の場合、/aa/bb を .NET バイナリパスとして使用します。

### 手順 4: Linux VDA パッケージのダウンロード

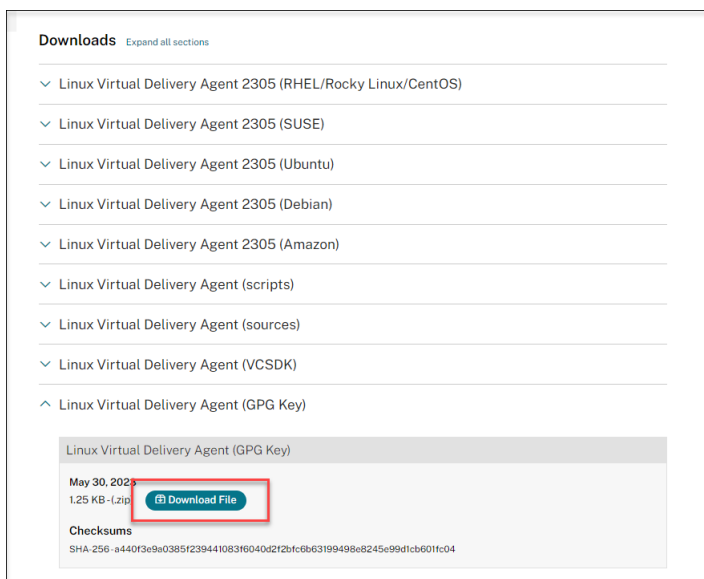
1. [Citrix Virtual Apps and Desktops のダウンロードページ](#) にアクセスします。
2. 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
3. **Components** を展開して Linux VDA を見つけます。例:



4. Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



5. 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
6. Linux VDA パッケージの整合性を検証するために使用できる GPG 公開キーをダウンロードします。例:



公開キーを使用して Linux VDA パッケージの整合性を検証するには:

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

```
1 rpmkeys --import <path to the public key>
2 rpm --checksig --verbose <path to the Linux VDA package>
```

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

```
1 sudo apt-get install dpkg-sig
2 gpg --import <path to the public key>
3 dpkg-sig --verify <path to the Linux VDA package>
```

### 手順 5: Linux VDA パッケージのインストール

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、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、LinuxVDA 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ます。EPEL のインストール方法については、<https://docs.fedoraproject.org/en-US/epel/>の説明を参照してください。

- Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 にインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新します。

```
1 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 をインストールすると、イーサネット接続が失われ、仮想マシンの再起動後に Linux VDA にアクセスできなくなることがあります。この問題を回避するには、仮想マシンに初めてログオンするときにルートパスワードを設定し、ルートとして仮想マシンにログオンできることを確認します。次に、仮想マシンを再起動した後、コンソールで次のコマンドを実行します:

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```

## Ubuntu/Debian ディストリビューション

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2 sudo apt-get install -f
```

注:

- Debian 11 ディストリビューションに必要な依存関係をインストールするには、`/etc/apt/sources.list` ファイルに「`deb http://deb.debian.org/debian/ bullseye main`」行を追加します。
- GCP 上の Ubuntu 20.04 の場合、RDNS を無効にします。これを行うには、`/etc/krb5.conf` の **[libdefaults]** 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、および GCP の SUSE 15.5 の場合は、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。
- `/etc/sysconfig/windowmanager` の **Default\_WM** パラメーターが **gnome** に設定されている。

2. 次のコマンドを実行して、Linux VDA をインストールします:

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

## 手順 6: 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、次のガイドを参照してください：

- [XenServer](#)
- [VMware ESX](#)
- [Nutanix 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、GPU を VM に割り当てます。
3. 仮想マシンを起動します。
4. ゲスト VM ドライバーを VM にインストールします。

#### 手順 7：使用するデータベースの指定

Linux VDA パッケージをインストールした後は、SQLite と PostgreSQL を切り替えることができます。このためには、次の手順を実行します：

注：

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで 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. **/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。
2. **sudo /opt/Citrix/VDA/sbin/ctxinstall.sh** または **/opt/Citrix/VDA/bin/easyinstall** を実行します。

#### 手順 8：簡単インストールを実行して環境と VDA を構成し、インストールを完了します

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、**en\_US.UTF-8** ロケールが OS にインストールされていることを確認します。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して **/etc/locale.gen** ファイルを編集してから、**sudo locale-gen** コマンドを実行します。

## ctxinstall.sh

ctxinstall.sh は、いくつかの事前構成と VDA 実行環境変数のセットアップを行うための、簡単インストールのスク립トです。

- このスク립トは root のみが実行できます。
- 簡単インストールでは、使用されるすべての環境変数の値を設定、保存、および同期するための構成ファイルとして、`/opt/Citrix/VDA/sbin/ctxinstall.conf` を使用します。テンプレート (`ctxinstall.conf.tpl`) をよく読んで上で、独自の `ctxinstall.conf` をカスタマイズすることをお勧めします。構成ファイルを初めて作成するときは、次のいずれかの方法を使用します：
  - `/opt/Citrix/VDA/sbin/ctxinstall.conf.tpl` テンプレートファイルをコピーして、`/opt/Citrix/VDA/sbin/ctxinstall.conf` として保存する。
  - `ctxinstall.sh` を実行する。`ctxinstall.sh` を実行するたびに、入力値は `/opt/Citrix/VDA/sbin/ctxinstall.conf` に保存されます。
- 簡単インストールではモジュール形式の実行をサポートします。モジュールには、事前チェック、インストール、ドメイン構成、セットアップ、検証が含まれます。
- このスク립トのデバッグについて詳しくは、`/var/log/xdl/ctxinstall.log` で確認できます。

詳しくは、ヘルプコマンド **ctxinstall.sh -h** を使用してご確認ください。

注：

- 最小特権の原則に従い、ルートユーザーのみが `/opt/Citrix/VDA/sbin/ctxinstall.conf` の読み取りができるようにします。これは、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です。
- Linux VDA をアンインストールすると、`/opt/Citrix/VDA` にある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` のバックアップを作成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립トを実行する前に、次の環境変数を設定します：

- **CTX\_XDL\_NON\_DOMAIN\_JOINED= ' y|n'**  
- マシンをドメインに参加させるかどうか。デフォルト値は ' n' です。ドメイン参加済みシナリオの場合は ' n' に設定します。
- **CTX\_XDL\_AD\_INTEGRATION= ' sssd|winbind|centrify|pbis|quest'** - Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。
- **CTX\_XDL\_DDC\_LIST= ' <list-ddc-fqdns>'** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の、完全修飾ドメイン名 (FQDN) のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。

- **CTX\_XDL\_VDI\_MODE=' y|n'** -専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を **'y'** に設定します。
- **CTX\_XDL\_HDX\_3D\_PRO=' y|n'** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= **'y'** となります)。
- **CTX\_XDL\_START\_SERVICE = 'y|n'** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE = 'y|n'** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。
- **CTX\_XDL\_ADD\_FIREWALL\_RULES = 'y|n'** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート (デフォルトではポート 80 およびポート 1494) を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/ <none>** '-セッションで使用する GNOME、GNOME クラシック、KDE、MATE、または **Xfce** デスクトップ環境を指定します。'**<none>**' に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。
- **CTX\_XDL\_DOTNET\_RUNTIME\_PATH=path-to-install-dotnet-runtime** -新しいブローカーエージェントサービス (**ctxvda**) をサポートするための .NET をインストールするパス。デフォルトのパスは **'usr/bin'** です。
- **CTX\_XDL\_VDA\_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX\_XDL\_SITE\_NAME =<dns-name>** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、'**<none>**' に設定します。
- **CTX\_XDL\_LDAP\_LIST=' <list-ldap-servers>'** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。例: 「ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268」。Active Directory フォレストでより高速な LDAP クエリを有効にするには、ドメインコントローラーで [グローバルカタログ] を有効にし、関連する LDAP ポート番号で 3268 を指定します。この変数は、デフォルトでは **'<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使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。不要な場合は、'**<none>'** に設定します。



- **CTX\_XDL\_SUPPORT\_DDC\_AS\_CNAME=' y|n'** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。
- **CTX\_EASYINSTALL\_DNS=' <ip-address-of-dns>'** - DNS の IP アドレス。
- **CTX\_EASYINSTALL\_HOSTNAME=host-name** - Linux VDA サーバーのホスト名。
- **CTX\_EASYINSTALL\_NTPS=address-of-ntp** - NTP サーバーの IP アドレスまたは文字列名。
- **CTX\_EASYINSTALL\_REALM=realm-name** - Kerberos 領域名。
- **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ているユーザーの名前。
- **CTX\_EASYINSTALL\_PASSWORD=password** - マシンをドメインに参加させているユーザーのパスワード。

注:

ドメイン参加には管理者アカウントを使用しないことをお勧めします。代わりに、管理者アカウント以外の Active Directory ユーザーにドメイン参加権限を委任してください。このためには、制御の委任ウィザードを使用して、ドメインコントローラーの制御を委任します。

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い場合でも、サイレントモードで `ctxinstall.sh` は中止せず、対話モードでユーザー入力を求めるプロンプトも表示されません。これらは値をエクスポートすることで、または `/Citrix/VDA/sbin/ctxinstall.conf` を編集することでのみ設定できます。

- **CTX\_EASYINSTALL\_NETBIOS\_DOMAIN=netbios-domain-name** - NetBIOS ドメイン名は、通常はドット (.) で区切られた DNS ドメイン名の最初のコンポーネントです。それ以外の場合は、別の NetBIOS ドメイン名をカスタマイズします。この変数はオプションです。
- **CTX\_EASYINSTALL\_OU=ou-value** - OU の値は、**Active Directory** 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「注意事項」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** - 簡単インストールは、インターネットから Centrify パッケージをダウンロードするために役立ちます。ただし、Centrify が既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\_PDIS\_LOCAL\_PATH= pdis-local-path** - 簡単インストールは、インターネットから PDIS パッケージをダウンロードするために役立ちます。ただし、PDIS が既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 PDIS パッケージを取得できます。この変数はオプションです。

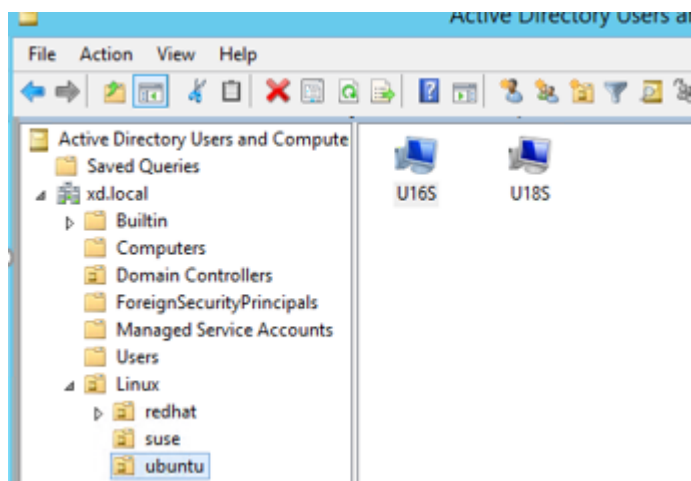
注意事項

- NetBIOS ドメイン名は通常、ドット (.) で区切られた DNS ドメイン名の最初のコンポーネントです。環境で別の NetBIOS ドメイン名をカスタマイズするには、`/opt/Citrix/VDA/sbin/ctxinstall.conf` で環境変数 **CTX\_EASYINSTALL\_NETBIOS\_DOMAIN** を設定します。

- VDA を特定の OU に追加するには、次の手順を実行します：

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

OU の例として、以下のスクリーンショットを参照してください：



2. 環境でワークグループをカスタマイズするには、`/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、上記のスクリーンショットにおける OU 名の例に基づいています。これ以外の、所属する組織の任意の OU 名を使用することができます。

OS	Winbind	SSSD	Centrify	PBIS
Amazon Linux 2	"Linux/ amazon"	"Linux/ amazon"	"XD.LOCAL/ Linux/amazon "	"Linux/ amazon"
Debian	"Linux/ debian"	"Linux/ debian"	"XD.LOCAL/ Linux/debian "	"Linux/ debian"
RHEL 9.4/9.3/9.2、 Rocky Linux 9.4/9.3/9.2	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	-	-
RHEL 8.x	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	"XD.LOCAL/ Linux/redhat "	"Linux/ redhat"

OS	Winbind	SSSD	Centrify	PBIS
Rocky Linux 8.x	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	-	-
SUSE	"Linux/suse"	"Linux/suse"	"XD.LOCAL/ Linux/suse"	"Linux/suse"
Ubuntu	"Linux/ ubuntu"	"Linux/ ubuntu"	"XD.LOCAL/ Linux/ubuntu "	"Linux/ ubuntu"

- Centrify ではピュア **IPv6** DNS 構成をサポートしていません。**Active Directory** クライアントで Active Directory サービスを適切に検索するためには、**IPv4** を使用する DNS サーバーが/etc/resolv.conf に少なくとも 1 つ存在している必要があります。

ログ:

```

1  ADSITE      : Check that this machine's subnet is in a site known by
   AD         : Failed
2           : This machine's subnet is not known by AD.
3           : We guess you should be in the site Site1.
```

この問題は、Centrify およびその構成に特有のもので、この問題を解決するには、次の手順を実行します:

- ドメインコントローラーの [管理ツール] を開きます。
  - [**Active Directory** のサイトとサービス] を選択します。
  - [サブネット] の適切なサブネットアドレスを追加します。
- 簡単インストールは、Linux VDA 7.16 以降のピュア **IPv6** をサポートしています。以下のような前提条件と制限事項があります:
    - お使いのマシンがピュア **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 リポジトリを設定する必要があります。
    - Centrify は、ピュア **IPv6** ネットワークではサポートされていません。

注:

ご使用のネットワークがピュア **IPv6** で、すべての入力が必要な **IPv6** 形式である場合、VDA は **IPv6** を使用して Delivery Controller に登録します。ご使用のネットワークが **IPv4** と **IPv6** のハイブリッド構成である場合、最初の DNS IP アドレスの種類によって、**IPv4** または **IPv6** のどちらが登録に使用されるかが決まります。

- 前述の CTX\_XDL\_DESKTOP\_ENVIRONMENT 変数を使用して、セッションで使用するデスクトップ環境を指定できます。コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。

- ドメインに参加させる方式として Centrify を選択する場合、`ctxinstall.sh` スクリプトでは Centrify パッケージが必要です。`ctxinstall.sh` で Centrify パッケージを取得する方法:

- 簡単インストールは、インターネットから Centrify パッケージを自動でダウンロードするために役立ちます。ディストリビューションごとの URL は次のとおりです:

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

CentOS: `wget https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-rhel6-x86_64.tgz`

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

Ubuntu/Debian: `wget https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-deb9-x86_64.tgz`

- Centrify が既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委任するには、`/opt/Citrix/VDA/sbin/ctx-install.conf` で `CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=/home/mydir` を指定します。  
例:

```
1  ls -ls /home/mydir
2  9548 -r-xr-xr-x. 1 root root 9776688 May 13 2016
   adcheck-rhel4-x86_64
3  4140 -r--r--r--. 1 root root 4236714 Apr 21 2016
   centrifyda-3.3.1-rhel4-x86_64.rpm
4  33492 -r--r--r--. 1 root root 34292673 May 13 2016
   centrifydc-5.3.1-rhel4-x86_64.rpm
5  4 -rw-rw-r--. 1 root root 1168 Dec 1 2015
   centrifydc-install.cfg
6  756 -r--r--r--. 1 root root 770991 May 13 2016
   centrifydc-ldaproxy-5.3.1-rhel4-x86_64.rpm
7  268 -r--r--r--. 1 root root 271296 May 13 2016
   centrifydc-nis-5.3.1-rhel4-x86_64.rpm
8  1888 -r--r--r--. 1 root root 1930084 Apr 12 2016
   centrifydc-openssh-7.2p2-5.3.1-rhel4-x86_64.rpm
9  124 -rw-rw-r--. 1 root root 124543 Apr 19 2016
   centrify-suite.cfg
10 0 lrwxrwxrwx. 1 root root 10 Jul 9 2012 install-
   express.sh -> install.sh
11 332 -r-xr-xr--. 1 root root 338292 Apr 10 2016 install
   .sh
12 12 -r--r--r--. 1 root root 11166 Apr 9 2015 release-
   notes-agent-rhel4-x86_64.txt
13 4 -r--r--r--. 1 root root 3732 Aug 24 2015 release-
   notes-da-rhel4-x86_64.txt
14 4 -r--r--r--. 1 root root 2749 Apr 7 2015 release-
   notes-nis-rhel4-x86_64.txt
```

```
15      12 -r--r--r--. 1 root root      9133 Mar 21 2016 release-
      notes-openssh-rhel4-x86_64.txt
```

- ドメインに参加させる方式として PBIS を選択する場合、ctxinstall.sh スクリプトでは PBIS パッケージが必要です。ctxinstall.sh で PBIS パッケージを取得する方法:

- 簡単インストールは、インターネットから PBIS パッケージを自動でダウンロードするために役立ちます。たとえば、ディストリビューションごとの URL は次のとおりです:

Amazon Linux 2、RHEL 8、SUSE 15.5: `wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh`

Debian、Ubuntu: `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。取得するには、`/opt/Citrix/VDA/sbin/ctxinstall.sh` ファイルの「pbisDownloadRelease」行と「pbisDownloadExpectedSHA256」行を変更します。

例として、以下のスクリーンショットを参照してください:

```
pbisDownloadPath_RHEL="https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh"
pbisDownloadPath_Ubuntu="https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh"
```

- PBIS が既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、`sudo /opt/Citrix/VDA/sbin/ctxinstall.sh` コマンドを **-S** オプションなしで使用します。コマンドラインインターフェイスのプロンプトごとに、関連する変数値を入力します。変数が既に設定されている場合、ctxinstall.sh は、変数の変更が必要かの確認を求めます。

**サイレントモード** サイレントモードでは、`/opt/Citrix/VDA/sbin/ctxinstall.conf` または `export` コマンドを使用して上記の変数を設定する必要があります。その後、`ctxinstall.sh -S` を実行します（ここでの **S** の文字は大文字であることに注意してください）。必要なすべての変数が設定されていないか一部の値が無効である場合、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、既に設定されていない限り `/Citrix/VDA/sbin/ctxinstall.conf` の値を上書きします。すべての更新された値は、ドメイン参加パスワード以外は `/Citrix/VDA/sbin/ctxinstall.conf` に保存されます。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、またはパスワードをエクスポートする必要があります。

```
1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
4 export CTX_XDL_VDI_MODE='y|n'
```

```
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<
  none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 export CTX_EASYINSTALL_DNS='<ip-address-of-dns>'
17 export CTX_EASYINSTALL_HOSTNAME='<host-name>'
18 export CTX_EASYINSTALL_NTFS='<address-of-ntfs>'
19 export CTX_EASYINSTALL_REALM='<realm-name>'
20 export CTX_EASYINSTALL_FQDN='<ad-fqdn-name>'
21 export CTX_EASYINSTALL_USERNAME='<domain-user-name>'
22 export CTX_EASYINSTALL_PASSWORD='<password>'
23 export CTX_EASYINSTALL_NETBIOS_DOMAIN='<netbios-domain>'
24 export CTX_EASYINSTALL_OU='<organization-unit>'
25 sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh -S
```

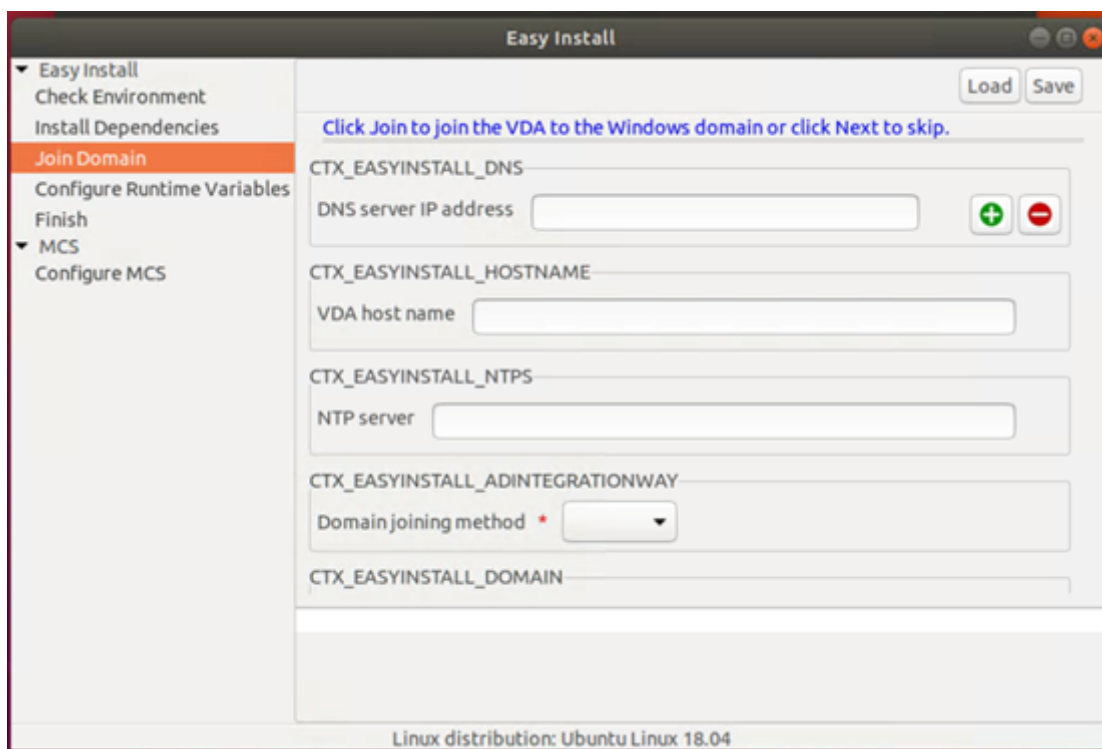
sudo コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべての変数を指定することができます。

VDA 実行環境変数（「**CTX\_XDL\_**」で始まる）を設定するには、**ctxinstall.sh -s** を実行できます（文字 **s** は小文字であることに注意してください）。

## GUI

GUI による簡単インストールを利用できます。VDA のデスクトップ環境で **/opt/Citrix/VDA/bin/easyinstall** コマンドを実行してから、簡単インストールの GUI の指示に従います。



簡単インストールの GUI は、次の操作をガイドします：

- システム環境を確認する
- 依存関係をインストールする
- 指定されたドメインに VDA を参加させる
- ランタイム環境を構成する

ヒント：

[保存] をクリックすると、指定したパスにあるローカルファイルに変数設定が保存されます。[読み込み] をクリックすると、指定したファイルから変数設定が読み込まれます。MCS 変数の構成については、「[手順 3: マスターイメージの準備](#)」を参照してください。

### 手順 9: XDPing の実行

`sudo /opt/Citrix/VDA/bin/xdping`を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「[XDPing](#)」を参照してください。

### 手順 10: Linux VDA の実行

**Linux VDA** の起動：

Linux VDA サービスを起動するには、次のコマンドを実行します：

```
1 sudo systemctl start ctxhdx.service
2
3 sudo systemctl start ctxvda.service
```

#### Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda.service
2
3 sudo systemctl stop ctxhdx.service
```

注:

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

#### Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda.service
2
3 sudo systemctl restart ctxhdx.service
4
5 sudo systemctl start ctxvda.service
```

#### Linux VDA の状態の確認:

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda.service
2
3 sudo systemctl status ctxhdx.service
```

### 手順 11: マシンカタログの作成

マシンカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明については、「[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。

次のように、Linux VDA マシンを含むマシンカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシンカタログの作成手順と異なる点があります:

- オペレーティングシステムには、次を選択します:
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。

- 同じマシンカタログで、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 の以前のバージョンは、「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** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **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暗黙的に選択されません。

ヒント:

マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

## 手順 12: デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明については、「[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は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで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、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよび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 2407](#)」を参照してください。

## 手順 13: Linux VDA のアップグレード（オプション）

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

注:

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

- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイルを処理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。以下では、rpm と dpkg が構成ファイルの変更でどのように機能するかについて説明しています：

- **rpm**: by default keeps the local version and saves the new version from the package with a .rpmnew extension.
- **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as **.dpkg-new** or **.dpkg-dist**, use the following command:

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

### Amazon Linux 2、RHEL、Rocky Linux ディストリビューションの場合：

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

注：

Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 で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 以降に更新します。

### SUSE ディストリビューションの場合：

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

### Ubuntu/Debian ディストリビューション：

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2
3 sudo apt-get install -f
```

## トラブルシューティング

このセクションの情報を参照して、簡単インストール機能を使用することで発生する可能性のある問題のトラブルシューティングを実行できます。

### SSSD を使用してドメインに参加できない

ドメインに参加しようとすると、次のような出力のエラーが発生することがあります（画面印刷のログを確認する）：

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:

```
1 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)'.
2 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))
3 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')
4 Check the following:- The system clock is in sync between this machine
  and the delivery controller.
5 - The Active Directory provider (e.g. winbind daemon) service is
  running and correctly configured.
6 - Kerberos is correctly configured on this machine.
7 If the problem persists, please refer to Citrix Knowledge Base article
  CTX117248 for further information.
8 Error Details:
9 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'.
10 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.
11 Ensure that the system clock is in sync between this machine and the
  delivery controller.
12 Verify the NTP daemon is running on this machine and is correctly
  configured.
13 2016-11-04 02:11:52.364 [ERROR] - Could not register with any
  controllers. Waiting to try again in 120000 ms. Multi-forest - false
14 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、次の手順を実行します:

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 でマシンカタログおよびデリバリーグループを削除します。
4. `/opt/Citrix/VDA/sbin/ctxinstall.sh` を実行します。
5. Delivery Controller でマシンカタログおよびデリバリーグループを作成します。

### Ubuntu のデスクトップセッションで灰色の画面が表示される

セッションを起動すると、空のデスクトップでブロックされる問題が発生します。また、マシンのコンソールでも、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灰色の画面が表示されます。

この問題を解決するには、次の手順を実行します：

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. 次の行を `/etc/lightdm/lightdm.conf` に追加します。  
`greeter-show-manual-login=true`

### Ubuntu のデスクトップセッションを起動しようとするときホームディレクトリがないため失敗する

`/var/log/xdl/hdx.log`:

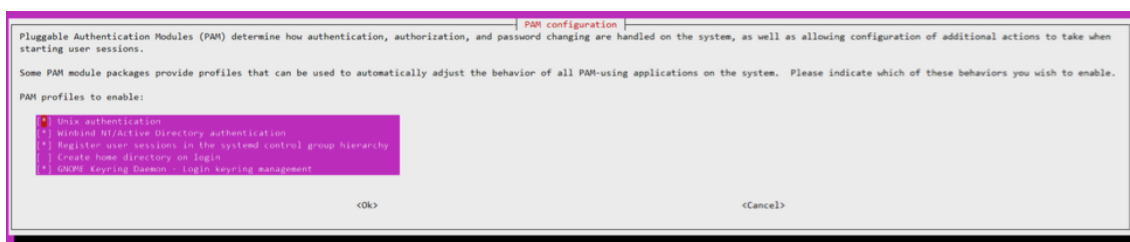
```
1 2016-11-02 13:21:19.015 <P22492:S1> citrix-ctxlogin: StartUserSession:
   failed to change to directory(/home/CITRIXLAB/ctxadmin) errno(2)
2
3 2016-11-02 13:21:19.017 <P22227> citrix-ctxhdx: logSessionEvent:
   Session started for user ctxadmin.
4
5 2016-11-02 13:21:19.023 <P22492:S1> citrix-ctxlogin: ChildPipeCallback:
   Login Process died: normal.
6
7 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、次の手順を実行します：

1. コマンドラインで、**pam-auth-update** を入力します。
2. 表示されたダイアログで、[ログイン時にホームディレクトリを作成する] 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を起動または終了できない

/var/log/messages (RHEL または CentOS の場合)

```
1 Oct 27 04:17:16 CentOS7 citrix-ctxhdx[8978]: Session started for user
  CITRIXLAB\ctxadmin.
2
3 Oct 27 04:17:18 CentOS7 kernel: traps: gnome-session[19146] trap int3
  ip:7f89b3bde8d3 sp:7fff8c3409d0 error:0
4
5 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...
6
7 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)
8
9 Oct 27 04:17:18 CentOS7 gnome-session: aborting...
10
11 Oct 27 04:17:18 CentOS7 citrix-ctxgfx[18981]: Exiting normally.
12
13 Oct 27 04:17:18 CentOS7 citrix-ctxhdx[8978]: Session stopped for user
  CITRIXLAB\ctxadmin.
```

Ubuntu ディストリビューションの場合は、log /var/log/syslog を使用

```
1 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] pid.c:
  Stale PID file, overwriting.
2
3 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.
4
5 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.
6
7 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.
8
9 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.]
10
11 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

```

再起動するまで機能しないグループまたはモジュールがあります。**dbus** エラーメッセージがログに表示される場合、システムを再起動してから再試行することをお勧めします。

**SELinux** で **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:

```

1 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
2
3 Jan 25 23:30:31 yz-rhel72-1 python[3945]: SELinux is preventing /usr/
  sbin/sshd from setattr access on the directory /root.
4
5 ***** Plugin catchall_boolean (89.3 confidence) suggests
  *****
6
7 If you want to allow polyinstantiation to enabled
8
9 Then you must tell SELinux about this by enabling the '
  polyinstantiation_enabled' boolean.
10
11 You can read 'None' man page for more details.
12
13 Do
14
15     setsebool -P polyinstantiation_enabled 1
16
17 ***** Plugin catchall (11.6 confidence) suggests
  *****
18
19 If you believe that sshd should be allowed setattr access on the root
  directory by default.

```

```
20
21 Then you should report this as a bug.
22
23 You can generate a local policy module to allow this access.
24
25     Do
26
27         allow this access for now by executing:
28
29         # grep sshd /var/log/audit/audit.log | audit2allow -M mypol
30
31 # semodule -i mypol.pp
```

この問題を解決するには、次の手順を実行します：

1. /etc/selinux/config に次の変更を加えることで、SELinux を無効にします。  
SELINUX=disabled
2. VDA を再起動します。

## MCS を使用したドメイン非参加 Linux VDA の作成

August 20, 2024

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

### 重要：

- MCS を使用したドメイン非参加 VDA の作成は、Citrix DaaS でのみサポートされます。Citrix DaaS のお客様の場合：
  - コントロールプレーンは、Citrix DaaS 経由で展開する必要があります。
  - ドメイン非参加 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せん。ドメイン非参加 VDA は、Citrix DaaS のコントロールプレーンによって管理されます。
  - ドメイン非参加の VDA を作成する場合、Citrix Gateway サービスを利用しているお客様は、[Rendezvous V2](#)が有効になっていることを確認する必要があります。Cloud Connector: オンプレミスハイパーバイザーでマシンをプロビジョニングする予定の場合、または Workspace で Active Directory を ID プロバイダーとして使用する場合にのみ必要です。
- ドメイン非参加の VDA を作成する場合、MCS と簡単インストールの両方を使用できます。詳しくは、「[MCS を使用したドメイン非参加 Linux VDA の作成](#)」および「[簡単インストールを使用したドメイン非参加 Linux VDA の作成 \(Technical Preview\)](#)」を参照してください。
- MCS はベアメタルサーバーをサポートしていません。
- ドメイン非参加の Linux VDA では、次の機能を使用できます：

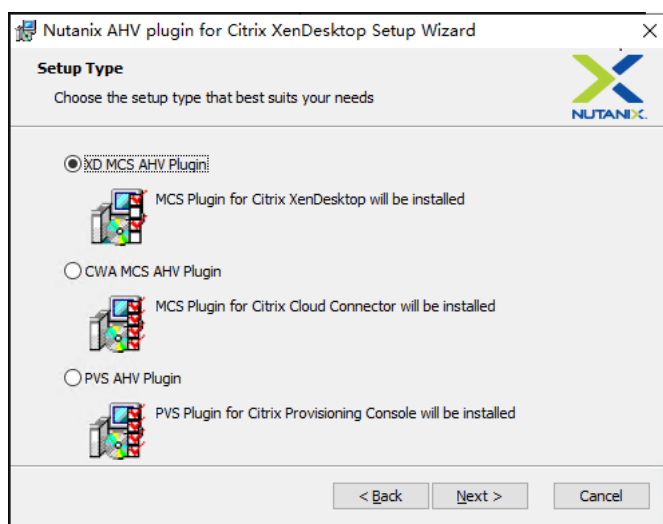
- ドメイン非参加の VDA で指定された属性を持つローカルユーザーを作成する
- SSO 以外の認証
- Azure Active Directory を使用した認証
- Rendezvous V2

(Nutanix の場合のみ) 手順 1: Nutanix AHV プラグインのインストールと登録

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

手順 1a: オンプレミス **Delivery Controller** 用の **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、**[XD MCS AHV Plugin]** を選択して Delivery Controller にインストールします。



手順 1b: クラウド **Delivery Controller** 用の **Nutanix AHV** プラグインをインストールして登録する

Citrix Cloud Connector 用に **[CWA MCS AHV Plugin]** を選択してインストールします。Citrix Cloud テナントに登録されているすべての Citrix Cloud Connector にプラグインをインストールします。AHV なしでリソースの場所にサービスを提供する場合でも、Citrix Cloud Connector を登録する必要があります。

手順 1c: プラグインをインストールした後、次の手順を実行する

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

ヒント:

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

## 手順 2: ホスト接続の作成

ホストは、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 がホスト上の仮想マシンと通信するために使用する情報を指定できます。詳細情報には、リソースの場所、ホストの種類、アクセス資格情報、使用するストレージ方法、およびホスト上の仮想マシン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、リソースの場所にホストリソース（ストレージとネットワーク）を用意する必要があります。

1. Citrix Cloud にサインインします。
2. 左上のメニューで、[マイサービス] > [DaaS] を選択します。
3. [管理] > [完全な構成] の左側ペインで [ホスト] を選択します。
4. 操作バーの [接続およびリソースの追加] を選択します。
5. ウィザードの指示に従って、以下のページの操作を行います。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。各ページの操作を終えたら、[概要] ページに到達するまで [次へ] を選択します。

## 手順 2a: 接続

[接続] ページで以下を実行します:

- 接続を作成するには、[新しい接続を作成する] をクリックします。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場合は、[既存の接続を使用する] を選択してから該当の接続を選択します。
- [ゾーン名] フィールドでゾーンを選択します。選択できるオプションは、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。選択できるオプションは、プラグインがゾーンに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-HypervisorPlugin -ZoneUid` を使用して、選択したゾーンで使用できるハイパーバイザープラグインの一覧を取得することもできます。
- 接続名を入力します。この名前は管理画面に表示されます。
- 仮想マシンを作成するツール、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、サービスプリンシパルを作成できます。

手順 **2b**: ストレージの管理

**Add Connection and Resources** [X]

1 Connection  
2 **Storage Management**  
3 Storage Selection  
4 Network  
5 Summary

**Storage Management**  
Configure virtual machine storage resources for this connection.

Select a cluster:

[Text Input] [Browse]

Select an optimization method for available site storage.

Use storage shared by hypervisors  
 Optimize temporary data on available local storage  
 Use storage local to the hypervisor

[Back] [Next] [Cancel]

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 ホストに対する接続を構成している場合は、クラスター名を参照してから選択します。他の接続の種類では、クラスター名は要求されません。

ストレージ管理方法（ハイパーバイザー間で共有されるストレージまたはハイパーバイザーのローカルに配置するストレージ）を選択します。

- ハイパーバイザー間で共有されるストレージを選択する場合、一時データを使用可能なローカルストレージで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、デフォルトではない一時ストレージのサイズを指定できます）。例外：クラスターストレージボリューム（CSV）を使用する場合、Microsoft System Center Virtual Machine Manager は、ローカルストレージでの一時データキャッシュディスクを許可しません。[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 のプール上で共有ストレージを使用する場合は、IntelliCache を使用して共有ストレージデバイスにかかる負荷を減らす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。

## 手順 2c: ストレージの選択

**Add Connection and Resources** [Close]

Connection  
 Storage Management  
 **Storage Selection**  
 Network  
 Summary

**Storage Selection**

When using local storage, you must select the type of data to store on each local storage device; machine operating system data, temporary data, and if not storing personal user data remotely, personal user data. At least one device must be selected for each data type.

Select data storage locations:

Name ↓	OS	Temporary
Library1 on [redacted]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Local storage on [redacted]	<input type="checkbox"/>	<input type="checkbox"/>
System32 on [redacted]	<input type="checkbox"/>	<input type="checkbox"/>
Users on [redacted]	<input type="checkbox"/>	<input type="checkbox"/>

[Back] [Next] [Cancel]

ストレージの選択について詳しくは、「[ホストストレージ](#)」を参照してください。

使用可能なデータの種類のごとに 1 つ以上の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の種類が変化します。ウィザードの次のページに進むには、サポートされる各データの種類に対して 1 つ以上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で一時データを最適化します] を有効にした場合、[ストレージの選択] ページの下部に表示される構成オプションが増えます。一時データに使用する（同じハイパーバイザープールにある）ローカルストレージデバイスを選択できます。

現在選択中のストレージデバイスの数が表示されます（上図では「1 個のストレージデバイスが選択されました」）。このエントリの上にマウスを合わせると、選択したデバイスの名前が表示されます（構成されたデバイスがある場合のみ）。

1. 使用するストレージデバイスを変更するには [選択] を選択します。
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 [OK] を選択します。

## 手順 2d: リージョン

（一部のホストの種類にのみ表示）リージョンの選択により、仮想マシンが展開される場所を指定します。可能であれば、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。

### 手順 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、これにより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できます。

仮想マシンで使用するネットワークを 1 つまたは複数選択します。

一部の接続の種類 (Azure Resource Manager など) では、仮想マシンが使用するサブネットも表示されます。サブネットを 1 つまたは複数選択します。

### 手順 2f: まとめ

選択内容を確認します。変更を行う場合は、[戻る] を使って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にデフォルト以外の値を設定できます。

注:

フルアクセス権管理者については、スコープは表示されません。詳しくは、「[管理者、役割、およびスコープ](#)」を参照してください。

詳しくは、「[接続の作成と管理](#)」を参照してください。

### 手順 3: マスターイメージの準備

ヒント:

単一のイメージを使用して、ドメイン参加済み VDA とドメイン非参加 VDA の両方を作成できます。

(XenServer (旧称 Citrix Hypervisor) の場合のみ) 手順 3a: **XenServer VM Tools** をインストールする

xe CLI または XenCenter を使用するために、仮想マシンごとにテンプレート仮想マシンに XenServer VM Tools をインストールします。このツールがインストールされていないと、仮想マシンのパフォーマンスが低下する可能性があります。ツールがなければ、次のいずれも実行できません:

- 仮想マシンを正しくシャットダウン、再起動、または一時停止する。
- XenCenter でその仮想マシンのパフォーマンスデータを表示する。
- 実行中の仮想マシンを移行する (XenMotionを使用)。
- スナップショットまたはメモリを含んだスナップショット (チェックポイント) を作成したり、スナップショットを復元したりする。
- 実行中の Linux 仮想マシン上の vCPU の数を調整する。

1. [XenServer ダウンロードページ](#)または[Citrix Hypervisor ダウンロードページ](#)から、使用中のハイパーバイザーバージョンに応じた Linux 向け XenServer VM Tools ファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz` ファイルを、Linux 仮想マシン、または Linux 仮想マシンがアクセスできる共有ドライブにコピーします。
3. `tar` ファイルの内容を展開します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. 次のコマンドを実行して、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します。

**RHEL/CentOS/Rocky Linux/SUSE** の場合:

```
1 sudo rpm -i <extract-directory>/xe-guest-utilities_{
2   package-version }
3   _x86_64.rpm
```

**Ubuntu/Debian** の場合:

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

5. XenCenter の [全般] タブで、テンプレート仮想マシンの仮想化状態を確認します。XenServer VM Tools が正しくインストールされている場合、仮想化の状態は [最適化済み] を示します。

手順 **3b**: テンプレート仮想マシンに **.NET** および **Linux VDA** パッケージをインストールする

注:

現在実行中の VDA をテンプレート仮想マシンとして使用するには、この手順を省略します。

Linux VDA パッケージをインストールする前に、テンプレート仮想マシンに .NET をインストールし、以下について確認します:

- Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリビューションに、.NET ランタイムに加えて .ASP.NET Core ランタイムをインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。他のディストリビューションにはバージョン 8 が必要です。
- Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィールドからインストールします。それ以外の場合は、Microsoft パッケージフィールドから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

.NET をインストールした後、Linux ディストリビューションに応じて次のコマンドを実行し、Linux VDA をインストールします:

**RHEL/CentOS/Rocky Linux** の場合:

```
1 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 をインストールすると、イーサネット接続が失われ、仮想マシンの再起動後に Linux VDA にアクセスできなくなることがあります。この問題を回避するには、仮想マシンの起動前に次のコマンドを実行します:

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```

**Ubuntu/Debian** の場合:

```
1 sudo dpkg -i <PATH>/<Linux VDA DEB>
2
3 apt-get install -f
```

**SUSE** の場合:

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

手順 **3c**: (**RHEL** の場合のみ) **ntfs-3g** を提供できる **EPEL** リポジトリをインストールする RHEL 8 に EPEL リポジトリをインストールします。EPEL のインストール方法については、<https://docs.fedoraproject.org/en-US/epel/> の説明を参照してください。

手順 **3e**: (**SUSE** の場合のみ) **ntfs-3g** を手動でインストールする

SUSE プラットフォームには、**ntfs-3g** を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、**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:

```
1 sudo zypper install gcc
2 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** パッケージを展開します。

```
1 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:

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

5. **ntfs-3g** をインストールします:

```
1 ./configure
2 make
3 make install
```

手順 **3f**: (Ubuntu の場合のみ) **/etc/network/interfaces** ファイルを編集する

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

ヒント:

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

手順 **3f**: (Ubuntu の場合のみ) **/etc/resolv.conf** を参照する

**/etc/resolv.conf** が **/run/systemd/resolve/stub-resolv.conf** で は な く **/run/systemd/resolve/resolv.conf** を参照するようにします:

```
1 unlink /etc/resolv.conf
2
3 ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

手順 **3h**: 使用するデータベースを指定する

Linux VDA パッケージをインストールした後は、SQLite と PostgreSQL を切り替えることができます。このためには、次の手順を実行します:

注:

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで 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。

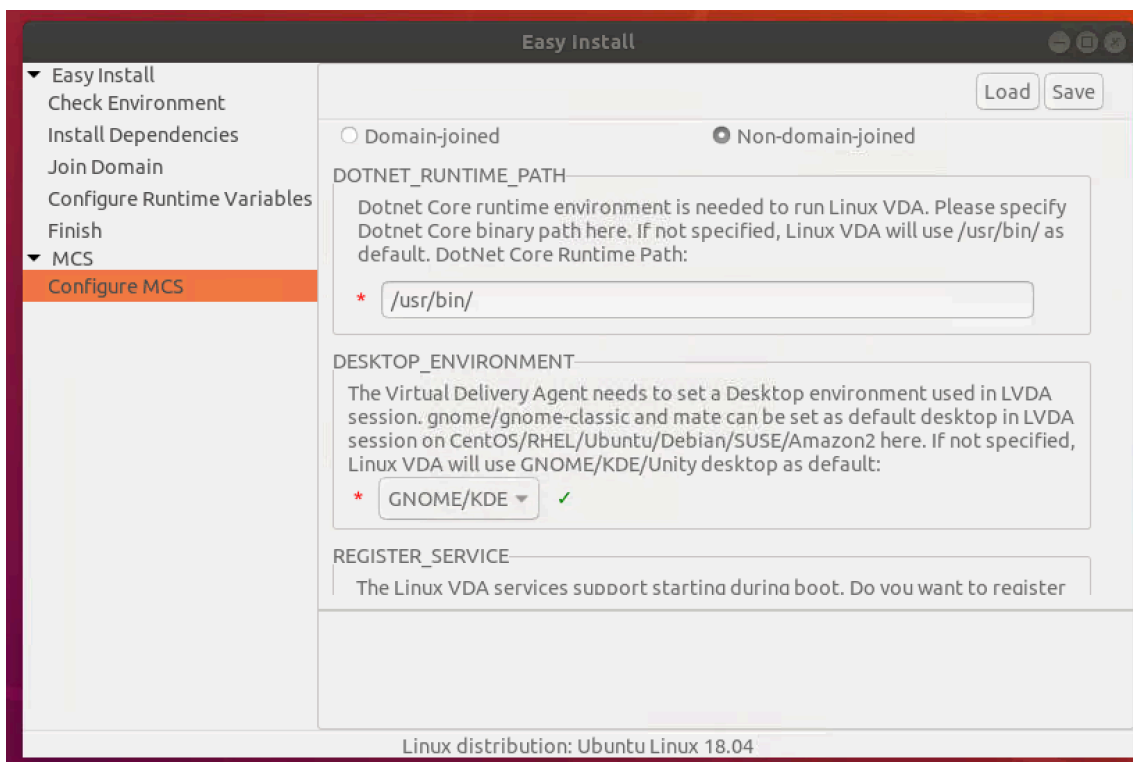
1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省きます。
2. **deploymcs.sh** を実行する前に **/etc/xdl/db.conf** を編集します。



## 手順 3i: MCS 変数を構成する

MCS 変数を構成するには、次の 2 つの方法があります：

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



## ヒント：

[保存] をクリックすると、指定したパスにあるローカルファイルに変数設定が保存されます。[読み込み] をクリックすると、指定したファイルから変数設定が読み込まれます。

以下は、ドメイン非参加のシナリオにおいて構成できる MCS 変数です。デフォルトの変数値を使用するか、必要に応じて変数をカスタマイズできます（オプション）：

`DOTNET_RUNTIME_PATH=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`

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

テンプレートマシンで、コマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加して、必要なレジストリ値を作成または更新します。この操作によって、MCS でプロビジョニングされたマシンを再起動するたびにデータと設定が失われないようにします。

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

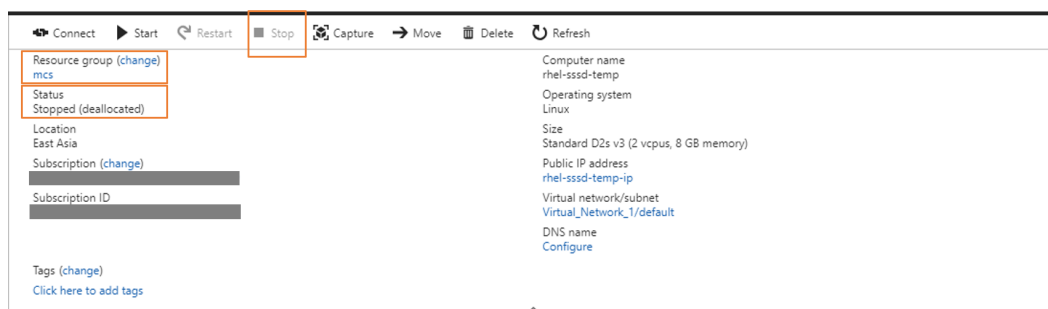
たとえば、次のそれぞれのコマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加して、レジストリ値を作成または更新できます：

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

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

手順 **3k**: マスターイメージを作成する

- `/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よりも優先されます。
- 使用するパブリッククラウドに基づき、マスターイメージのスナップショットを作成して名前を付けます。
  - (**XenServer** (旧称 **Citrix Hypervisor**)、**GCP**、および **VMware vSphere** の場合) テンプレート仮想マシンにアプリケーションをインストールし、テンプレート仮想マシンをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。
  - (**Azure** の場合) テンプレート仮想マシンにアプリケーションをインストールし、Azure Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンの電源状態が、[**Stopped (deallocated)**] になっ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検索する際に名前が必要です。



- (AWS の場合) テンプレート仮想マシンにアプリケーションをインストールし、AWS EC2 Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンのインスタンス状態の表示が **[Stopped]** であることを確認します。テンプレート仮想マシンを右クリックし、**[Image] > [Create Image]** を選択します。必要に応じて情報を入力し、設定を行います。**[Create Image]** をクリックします。

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

注:

Citrix Virtual Apps and Desktops で使用するには、Acropolis スナップショット名を「XD\_」で始める必要があります。必要に応じて、Acropolis コンソールを使用してスナップショットの名前を変更します。スナップショットの名前を変更したら、カタログ作成ウィザードを再起動して、更新された一覧を取得します。

#### 手順 4: マシンカタログの作成

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 固有の [コンテナ] ページで、前にテンプレート仮想マシン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、イメージの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 の数と仮想 CPU あたりのコア数を確認します。マシンの展開方法として MCS を選択し、カタログで作成するマシンの ID としてドメイン非参加を選択します。

必要に応じて他の構成タスクを実行します。詳しくは、「[マシンカタログの作成](#)」を参照してください。

注:

Delivery Controller でのマシンカタログの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**」という接頭辞が付いたマシンの電源を手動でオンにします。このアプローチは、作成プロセスを継続するのに役立ちます。

## 手順 5: デリバリーグループの作成

デリバリーグループは、いくつかのマシンカタログから選択したマシンをグループ化したものです。それらのマシンを使用できるユーザーと、そのユーザーに提供するアプリケーションおよびデスクトップを指定します。詳しくは、「[デリバリーグループの作成](#)」を参照してください。

## 簡単インストールを使用したドメイン非参加 **Linux VDA** の作成 (**Technical Preview**)

August 20, 2024

この記事では、簡単インストールを使用して、Citrix DaaS または Citrix Virtual Apps and Desktops 2407 以降でドメイン非参加の Linux VDA を作成し、マシンカタログに登録する方法について説明します。

重要:

- Citrix DaaS のお客様の場合:
  - ドメイン非参加 VDA は、パブリッククラウドまたはオンプレミスのデータセンターに展開できます。ドメイン非参加 VDA は、Citrix DaaS のコントロールプレーンによって管理されます。
  - ドメイン非参加の VDA を作成する場合、Citrix Gateway サービスを利用しているお客様は、[Rendezvous V2](#)が有効になっていることを確認する必要があります。Cloud Connector: オンプレミスハイパーバイザーでマシンをプロビジョニングする予定の場合、または Workspace で Active Directory を ID プロバイダーとして使用する場合にのみ必要です。
- ドメイン非参加 VDA を作成するには、MCS を使用することもできます。詳しくは、「[MCS を使用したドメイン非参加 Linux VDA の作成](#)」を参照してください。
  - MCS はベアメタルサーバーをサポートしていません。
- ドメイン非参加の Linux VDA では、次の機能を使用できます:
  - [ドメイン非参加の VDA で指定された属性を持つローカルユーザーを作成する](#)
  - [SSO 以外の認証](#)
  - [Azure Active Directory を使用した認証](#)
  - [Rendezvous V2](#)

## 手順 1: 空のマシナカタログの作成

Citrix Web Studio にサインインし、マシンを含まない空のマシナカタログを作成します。Linux VDA は、電源管理対象のマシナカタログへの登録にトークンファイルの使用をサポートしていないことに注意してください。

マシナカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明については、「[マシナカタログの作成](#)」および「[マシナカタログの管理](#)」を参照してください。

次のように、Linux VDA マシンを含むマシナカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシナカタログの作成手順と異なる点があります：

- オペレーティングシステムには、次を選択します：
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。
- 同じマシナカタログで、Linux VDA マシンと Windows VDA マシンを混在させないでください。
- 同じマシナカタログ内にドメインに参加済みマシンとドメイン非参加マシンを混在させないでください。
- リモート PC アクセスのマシナカタログは、ドメイン参加済みマシンのみでサポートされており、ドメイン非参加マシンではサポートされません。

### 注：

Citrix Studio の以前のバージョンは、「Linux OS」という概念をサポートしていません。ただし、**[Windows サーバー OS]** オプションまたは **[サーバー OS]** オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。**[Windows デスクトップ OS]** オプションまたは **[デスクトップ 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暗黙的に選択されます。

## 手順 2: 登録トークンの作成

簡単インストールを使用してドメイン非参加 VDA を作成するには、VDA をマシナカタログに登録し、その VDA をクラウドまたはオンプレミス Delivery Controller に対して認証するためのトークンファイルが必要です。Linux VDA は、電源管理対象のマシナカタログへの登録にトークンファイルの使用をサポートしていません。

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

### ヒント：

Citrix DaaS では、Web Studio は **[完全な構成]** と呼ばれます。

1. 先ほど作成した空のマシナカタログを選択し、操作バーで **[登録トークンを管理する]** を選択します。
2. **[登録トークンを管理する]** ページで、**[生成]** をクリックして登録トークンを作成します。または、有効な状態の既存のトークンを選択することもできます。

3. ウィザードに従って設定を完了します。
4. 正常に作成されたトークンをダウンロードし、最小限の権限で安全な場所に保存します。後で簡単インストールスクリプトを実行すると、CTX\_XDL\_NDJ\_ENROLLMENT\_TOKEN\_FILE 変数の使用によって、VDA を登録するためのトークンファイルへのパスを指定できます。

### 手順 3: .NET のインストール

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

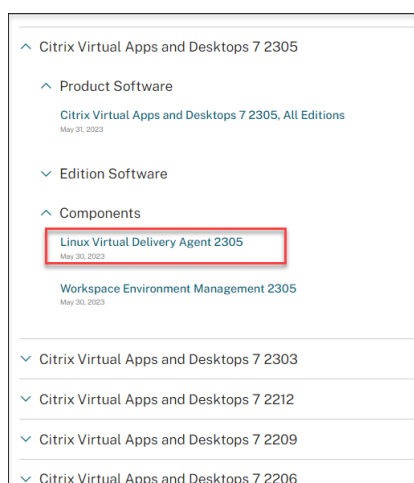
Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィールドからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

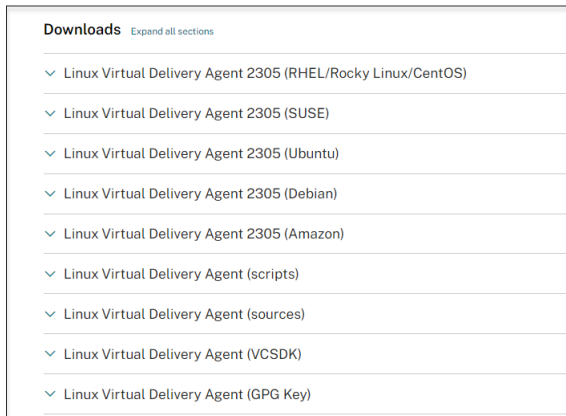
コマンド出力に基づいて、.NET ランタイムのバイナリパスを設定します。たとえば、コマンド出力が /aa/bb/dotnet の場合、/aa/bb を .NET バイナリパスとして使用します。

### 手順 4: Linux VDA パッケージのダウンロード

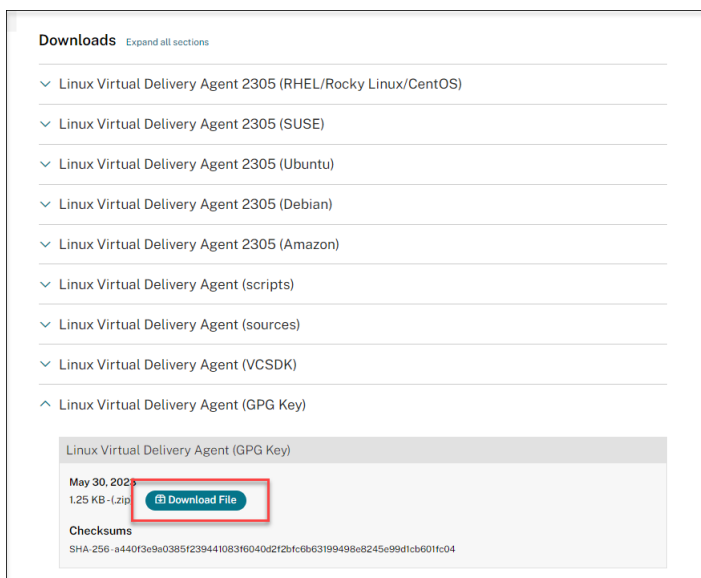
1. [Citrix Virtual Apps and Desktops のダウンロードページ](#) にアクセスします。
2. 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
3. **Components** を展開して Linux VDA を見つけます。例：



4. Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



5. 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
6. Linux VDA パッケージの整合性を検証するために使用できる GPG 公開キーをダウンロードします。例:



公開キーを使用して Linux VDA パッケージの整合性を検証するには:

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

```
1 rpmkeys --import <path to the public key>
2 rpm --checksig --verbose <path to the Linux VDA package>
```

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

```
1 sudo apt-get install dpkg-sig
2 gpg --import <path to the public key>
3 dpkg-sig --verify <path to the Linux VDA package>
```

## 手順 5: Linux VDA パッケージのインストール

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、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ます。EPEL のインストール方法については、<https://docs.fedoraproject.org/en-US/epel/>の説明を参照してください。
- Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 にインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新します。

```
1 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 をインストールすると、イーサネット接続が失われ、仮想マシンの再起動後に Linux VDA にアクセスできなくなることがあります。この問題を回避するには、仮想マシンに初めてログオンするときにルートパスワードを設定し、ルートとして仮想マシンにログオンできることを確認します。次に、仮想マシンを再起動した後、コンソールで次のコマンドを実行します:

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```

## Ubuntu/Debian ディストリビューション

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2 sudo apt-get install -f
```

注:

- Debian 11 ディストリビューションに必要な依存関係をインストールするには、`/etc/apt/sources.list` ファイルに「`deb http://deb.debian.org/debian/ bullseye main`」行を追加します。
- GCP 上の Ubuntu 20.04 の場合、RDNS を無効にします。これを行うには、`/etc/krb5.conf` の `[libdefaults]` 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、および GCP の SUSE 15.5 の場合は、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。



- **/etc/sysconfig/windowmanager** の **Default\_WM** パラメーターが **gnome** に設定されている。

2. 次のコマンドを実行して、Linux VDA をインストールします:

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

## 手順 6: 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、次のガイドを参照してください:

- [Citrix Hypervisor](#)
- [VMware ESX](#)
- [Nutanix 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、GPU を VM に割り当てます。
3. 仮想マシンを起動します。
4. ゲスト VM ドライバーを VM にインストールします。

## 手順 7: 使用するデータベースの指定

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。

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

注:

- SQLite は VDI モードにのみ使用することをお勧めします。
- 簡単インストールと MCS のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで 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。

## 手順 8: 簡単インストールを実行して環境と VDA を構成し、インストールを完了します

Linux VDA パッケージのインストール後、ctxinstall.sh スクリプトを使用して、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、**en\_US.UTF-8** ロケールが OS にインストールされていることを確認します。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して **/etc/locale.gen** ファイルを編集してから、**sudo locale-gen** コマンドを実行します。

**ctxinstall.sh**

ctxinstall.sh は、いくつかの事前構成と VDA 実行環境変数のセットアップを行うための、簡単インストールのスク립トです。

- このスク립トは root のみが実行できます。
- 簡単インストールでは、使用されるすべての環境変数の値を設定、保存、および同期するための構成ファイルとして、**/opt/Citrix/VDA/sbin/ctxinstall.conf** を使用します。テンプレート (**ctxinstall.conf.tpl**) をよく読んで、独自の **ctxinstall.conf** をカスタマイズすることをお勧めします。構成ファイルを初めて作成するときは、次のいずれかの方法を使用します:
  - **/opt/Citrix/VDA/sbin/ctxinstall.conf.tpl** テンプレートファイルをコピーして、**/opt/Citrix/VDA/sbin/ctxinstall.conf** として保存する。
  - **ctxinstall.sh** を実行する。**ctxinstall.sh** を実行するたびに、入力値は **/opt/Citrix/VDA/sbin/ctxinstall.conf** に保存されます。
- 簡単インストールではモジュール形式の実行をサポートします。モジュールには、事前チェック、インストール、ドメイン構成、セットアップ、検証が含まれます。
- このスク립トのデバッグについて詳しくは、**/var/log/xdl/ctxinstall.log** で確認できます。

詳しくは、ヘルプコマンド **ctxinstall.sh -h** を使用してご確認ください。

**注:**

- 最小特権の原則に従い、ルートユーザーのみが **/opt/Citrix/VDA/sbin/ctxinstall.conf** の読み取りができるようにします。これは、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です。
- Linux VDA をアンインストールすると、**/opt/Citrix/VDA** にある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** のバックアップを作成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립トを実行する前に、次の環境変数を設定します:

- **CTX\_XDL\_NON\_DOMAIN\_JOINED=' y|n'**
  - マシンをドメインに参加させるかどうか。デフォルト値は ' n ' です。ドメイン非参加シナリオの場合は ' y ' に設定します。

- **CTX\_XDL\_NDJ\_ENROLLMENT\_TOKEN\_FILE=** **<path-to-token-file-on-vda-machine>** - 単純インストールを使用してドメインに非参加 VDA を作成するには、VDA を Delivery Controller のマシンカタログに登録するためのトークンファイルが必要です。最小限の権限を持つトークンを安全な場所に保存します。
- **CTX\_XDL\_VDI\_MODE=** **y|n** - 専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を 'y' に設定します。
- **CTX\_XDL\_HDX\_3D\_PRO=** **y|n** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= 'y' となります)。
- **CTX\_XDL\_START\_SERVICE =** **y|n** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE =** **y|n** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。
- **CTX\_XDL\_ADD\_FIREWALL\_RULES =** **y|n** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート (デフォルトではポート 80 およびポート 1494) を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/mate** - セッションで使用する GNOME、GNOME クラシック、または MATE デスクトップ環境を指定します。変数を指定しないままにすると、VDA で構成済みのデフォルトデスクトップが使用されます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/** **<none>** - セッションで使用する GNOME、GNOME クラシック、KDE、MATE、または Xfce デスクトップ環境を指定します。' <none>' に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。CTX\_XDL\_DESKTOP\_ENVIRONMENT 変数を使用して、セッションで使用するデスクトップ環境を指定できます。コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。
- **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 をインストールするパス。デフォルトのパスは '/usr/bin' です。
- **CTX\_XDL\_VDA\_PORT=** **<port-number>** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。

対話モード 対話モードで **ctxinstall.sh** スクリプトを実行するには、**sudo /opt/Citrix/VDA/sbin/ctxinstall.sh** コマンドを **-S** オプションなしで使用します。コマンドラインインターフェイスのプロンプトごとに、関連する変数値を入力します。変数が既に設定されている場合、**ctxinstall.sh** は、変数の変更が必要かの確認を求めます。

サイレントモード サイレントモードでは、**/opt/Citrix/VDA/sbin/ctxinstall.conf** または `export` コマンドを使用して上記の変数を設定する必要があります。その後、**ctxinstall.sh -S** を実行します（ここでの **S** の文字は大文字であることに注意してください）。必要なすべての変数が設定されていないか一部の値が無効である場合、デフォルト値がない限り、**ctxinstall.sh** は実行を中止します。

これを設定すると、各変数のエクスポートされた値は、既に設定されていない限り **/Citrix/VDA/sbin/ctxinstall.conf** の値を上書きします。すべての更新された値は、**/Citrix/VDA/sbin/ctxinstall.conf** に保存されます。

```

1 export CTX_XDL_NON_DOMAIN_JOINED='y'
2 export CTX_XDL_NDJ_ENROLLMENT_TOKEN_FILE='<token-file-path>'
3 export CTX_XDL_VDI_MODE='y|n'
4 export CTX_XDL_START_SERVICE='y|n'
5 export CTX_XDL_REGISTER_SERVICE='y|n'
6 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
7 export CTX_XDL_HDX_3D_PRO='y|n'
8 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<
  none>'
9 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
10 export CTX_XDL_VDA_PORT='<port-number>'
11 sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh -S

```

`sudo` コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべての変数を指定することができます。

VDA 実行環境変数（「**CTX\_XDL\_**」で始まる変数）を設定するには、**ctxinstall.sh -s** を実行できます（文字 **s** は小文字であることに注意してください）。

## 手順 9: XDPing の実行

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「**XDPing**」を参照してください。

## 手順 10: Linux VDA の実行

### Linux VDA の起動:

Linux VDA サービスを起動するには、次のコマンドを実行します:

```

1 sudo systemctl start ctxhdx.service
2
3 sudo systemctl start ctxvda.service

```

### Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda.service
2
3 sudo systemctl stop ctxhdx.service
```

注:

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

#### Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda.service
2
3 sudo systemctl restart ctxhdx.service
4
5 sudo systemctl start ctxvda.service
```

#### Linux VDA の状態の確認:

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda.service
2
3 sudo systemctl status ctxhdx.service
```

#### 手順 11: デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「[デリバリーグループの作成](#)」を参照してください。

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は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで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、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「[Citrix Virtual Apps and Desktops 7 2407](#)」を参照してください。

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

August 21, 2024

MCS を使用して、ドメイン参加済み VDA とドメイン非参加 VDA を作成できます。Citrix DaaS でドメイン非参加の Linux VDA を作成する場合は、専用の記事「[MCS を使用したドメイン非参加 Linux VDA の作成](#)」も参照してください。

### 重要:

2212 リリース以降の重要な変更点は次のとおりです:

- /etc/xdl/mcs/mcs.conf ファイルまたは簡単インストールの GUI で **AD\_INTEGRATION** 変数にデフォルト値はなくなりました。必要に応じて値を設定する必要があります。詳しくは、この記事の「[手順 3h: MCS 変数を構成する](#)」セクションを参照してください。
- /etc/xdl/mcs/mcs.conf の **UPDATE\_MACHINE\_PW** エントリに対する有効な値は、**enabled** や **disabled** ではなく、**Y** または **N** になりました。詳しくは、この記事の「[マシンアカウントのパスワードの更新を自動化](#)」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

	Winbind	SSSD	Centrify	PBIS
Debian 12.5/11.9	はい	はい	いいえ	はい
RHEL 9.4/9.3/9.2	はい	はい	いいえ	いいえ
RHEL 8.10/8.9/8.8	はい	はい	はい	はい
Rocky Linux 9.4/9.3/9.2	はい	はい	いいえ	いいえ
Rocky Linux 8.10/8.9/8.8	はい	はい	いいえ	いいえ
SUSE 15.5	はい	はい	いいえ	はい
Ubuntu 22.04、 Ubuntu 20.04	はい	はい	いいえ	はい

- Citrix 製品では、関連する Linux ディストリビューションの初期の機能検証に次の Centrify バージョンを使用します:

Linux ディストリビューション	Centrify バージョン
RHEL 7/8	5.8.0
SUSE	5.7.1
Debian、Ubuntu	5.6.1

---

ほかのバージョンの Centrify を使用すると、エラーが発生する可能性があります。Centrify を使用してテンプレートマシンをドメインに追加しないでください。

- 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、次のタスクを実行してください：
  - テンプレートマシンで、`/etc/xdm/mcs/mcs.conf`ファイルに PBIS または Centrify パッケージのダウンロードパスを設定するか、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、MCS で作成された下位のすべてのマシンに対する書き込みおよびパスワードの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`の実行が終了した後、MCS で作成されたマシンを再起動する前に、環境に応じて、Delivery Controller または Citrix Cloud Connector で `klist -li 0x3e4 purge`を実行します。
- SSSD を MCS のテンプレート仮想マシンとして使用して、ドメインに接続された現在実行中の RHEL 8.x/9.x または Rocky Linux 8.x/9.x VDA を使用するには、次のことを確認してください：
  - VDA は簡単インストールではなく手動でインストールします。簡単インストールは RHEL 8.x/9.x および Rocky Linux 8.x/9.x で **Adcli** を使用します。SSSD と **Adcli** の組み合わせは MCS ではサポートされていません。
  - Samba サーバーは、AD 認証に SSSD を使用するよう構成されています。詳しくは、<https://access.redhat.com/solutions/3802321>の Red Hat の記事を参照してください。

#### サポートされるハイパーバイザー

- AWS
- XenServer (旧称 Citrix Hypervisor)
- GCP
- Microsoft Azure
- Nutanix AHV
- VMware vSphere

サポート対象ではない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、予期しない問題が発生することがあります。

## MCS を使用した Linux 仮想マシンの作成

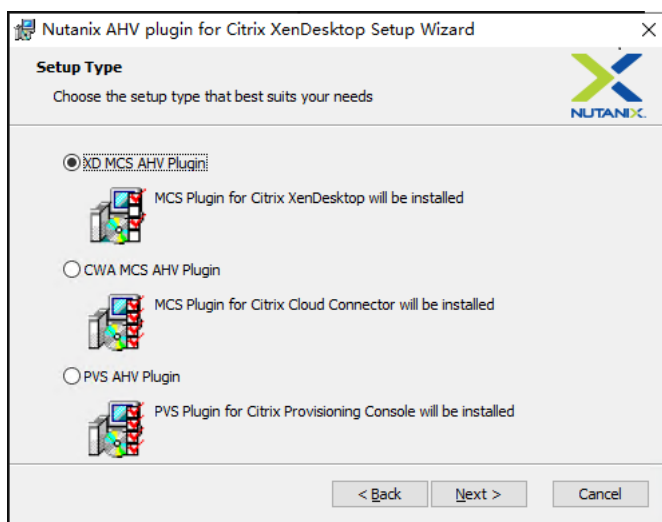
### 注意事項

- 2203 リリース以降、Microsoft Azure、AWS、および GCP で Linux VDA をホストすることは、Citrix DaaS (旧称 Citrix Virtual Apps and Desktops サービス) でサポートされていました。これらのパブリッククラウドホスト接続を Citrix Virtual Apps and Desktops 展開環境に追加する場合は、Citrix Universal ハイブリッドマルチクラウド (HMC) ライセンスが必要です。
- MCS を使用して仮想マシンを作成する場合、ベアメタルサーバーはサポートされません。

### (Nutanix の場合のみ) 手順 1: Nutanix AHV プラグインのインストールと登録

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

手順 **1a**: オンプレミス **Delivery Controller** 用の **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、**[XD MCS AHV Plugin]** を選択して Delivery Controller にインストールします。



手順 **1b**: クラウド **Delivery Controller** 用の **Nutanix AHV** プラグインをインストールして登録する Citrix Cloud Connector 用に **[CWA MCS AHV Plugin]** を選択してインストールします。Citrix Cloud テナントに登録されているすべての Citrix Cloud Connector にプラグインをインストールします。AHV なしでリソースの場所にサービスを提供する場合でも、Citrix Cloud Connector を登録する必要があります。



手順 **1c**: プラグインをインストールした後、次の手順を実行する

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

ヒント:

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

手順 **2**: ホスト接続の作成

このセクションでは、Azure、AWS、XenServer (旧称 Citrix Hypervisor)、GCP、Nutanix AHV、および VMware vSphere へのホスト接続を作成する方法の例を示します。

注:

オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。

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 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 での Azure へのホスト接続の作成](#)
- [Citrix Studio での AWS へのホスト接続の作成](#)
- [Citrix Studio での XenServer へのホスト接続の作成](#)
- [Citrix Studio での GCP へのホスト接続の作成](#)
- [Citrix Studio での Nutanix へのホスト接続の作成](#)
- [Citrix Studio での VMware へのホスト接続の作成](#)

### Citrix Studio での Azure へのホスト接続の作成

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
2. 接続およびリソースの追加ウィザードで、接続の種類として Microsoft Azure を選択します。
3. 接続の種類として [Microsoft Azure] を選択します。
4. ウィザードの指示に従って、各ページの操作を行います。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。各ページの操作を終えたら、[概要] ページに到達するまで [次へ] を選択します。詳しくは、「[MCS を使用したドメイン非参加の Linux VDA の作成](#)」の記事にある「手順 2: ホスト接続の作成」を参照してください。

### Citrix Studio での AWS へのホスト接続の作成

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
2. 接続およびリソースの追加ウィザードで、接続の種類として **Amazon EC2** を選択します。  
たとえば、オンプレミスの Citrix Studio では次のようになります：

The screenshot shows a software interface titled "Add Connection and Resources". On the left, a sidebar labeled "Studio" contains a list of navigation items: "Connection", "VM Location", "Network", and "Summary". The "Connection" item is selected. The main area is titled "Connection" and contains two radio button options: "Use an existing Connection" (unselected) and "Create a new Connection" (selected). Under "Use an existing Connection", there is a dropdown menu showing "awsec2". Under "Create a new Connection", there is a "Connection type:" dropdown menu showing "Amazon EC2". Below this, a text prompt reads: "Your cloud administrator should provide the following information." This is followed by an "Import keys file:" label, a "Browse..." button, and a note: "Use a file to automatically enter API key and Secret key." There are two empty text input fields labeled "API key:" and "Secret key:". Below these is a link: "Learn about user permissions" with an information icon. At the bottom, there is a "Connection name:" label and a text input field containing the text "Example: MyConnection". At the bottom right of the dialog, there are three buttons: "Back", "Next" (which is highlighted in blue), and "Cancel".

3. AWS アカウントの API キーと秘密キーを入力し、接続名を入力します。

**API** キーはアクセスキー ID で、**秘密キー**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アクセスキーを作成できます。アクセスキーを作成するには、次の手順を実行します：

- a) AWS サービスにサインインします。
  - b) ID およびアクセス管理 (IAM) コンソールに移動します。
    - 1 左側のナビゲーションペインで、**[Users]** を選択します。
  - c) 対象ユーザーを選択して下にスクロールして、**[Security credentials]** タブを選択します。
  - d) 下にスクロールして、**[Create access key]** をクリックします。新しいウィンドウが開きます。
  - e) **[Download .csv file]** をクリックし、アクセスキーを安全な場所に保存します。
4. ウィザードの指示に従って、各ページの操作を行います。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。各ページの操作を終えたら、**[概要]** ページに到達するまで **[次へ]** を選択します。

#### Citrix Studio での XenServer へのホスト接続の作成

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で **[構成] > [ホスト] > [接続およびリソースの追加]** の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで **[管理] > [ホスト] > [接続およびリソースの追加]** の順に選択し、ホスト接続を作成します。

2. 接続およびリソースの追加ウィザードの [接続の種類] フィールドで XenServer (旧称 Citrix Hypervisor) を選択します。
3. 接続アドレス (XenServer URL) と資格情報を入力します。
4. 接続名を入力します。

**Citrix Studio** での **GCP** へのホスト接続の作成 [Google Cloud Platform 仮想化環境](#)に合わせて GCP 環境をセットアップしてから、次の手順を実行して GCP へのホスト接続を作成します。

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
2. 接続およびリソースの追加ウィザードで、接続の種類として **Google Cloud Platform** を選択します。

たとえば、Citrix Cloud の Web ベースの Studio コンソールでは次のようになります：

The screenshot shows the 'Add Connection and Resources' wizard in Citrix Studio. The 'Connection' step is selected in the left-hand navigation pane. The main area is titled 'Connection' and contains the following options and fields:

- Use an existing connection
  - Field: BingTest (dropdown)
- Create a new connection
  - Field: Zone name (dropdown)
  - Field: Connection type: Google Cloud Platform (dropdown)
  - Field: Service account key: Import key... (button)
  - Field: Service account ID: (text input)
  - Field: Connection name: (text input)
  - Section: Create virtual machines using:
    - Citrix provisioning tools (Machine Creation Services or Citrix Provisioning)
    - Other tools

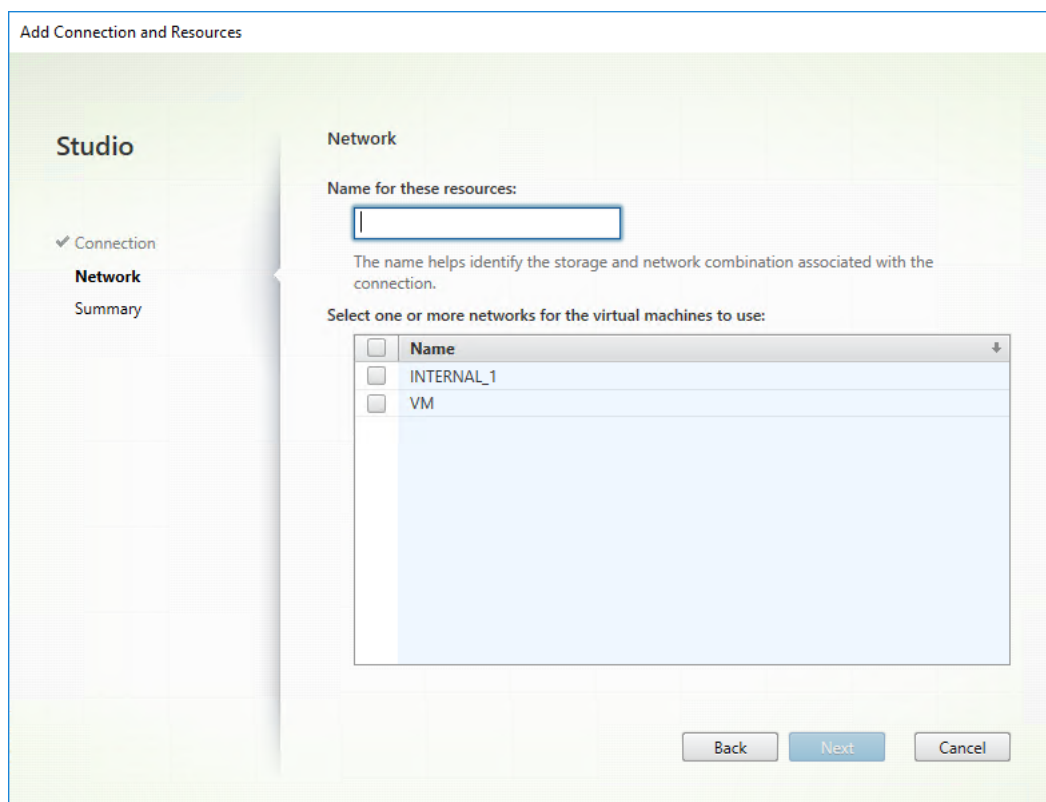
At the bottom of the wizard, there are three buttons: 'Next', 'Cancel', and a circular icon with the number '7'.

3. GCP アカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードの指示に従って、各ページの操作を行います。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。各ページの操作を終えたら、[概要] ページに到達するまで [次へ] を選択します。詳しくは、「[MCS を使用したドメイン非参加の Linux VDA の作成](#)」の記事にある「手順 2: ホスト接続の作成」を参照してください。

## Citrix Studio での Nutanix へのホスト接続の作成

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
2. 接続とリソースの追加ウィザードの [接続] ページで、接続の種類として [Nutanix AHV] を選択し、ハイパーバイザーのアドレスと資格情報、接続の名前を指定します。[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。

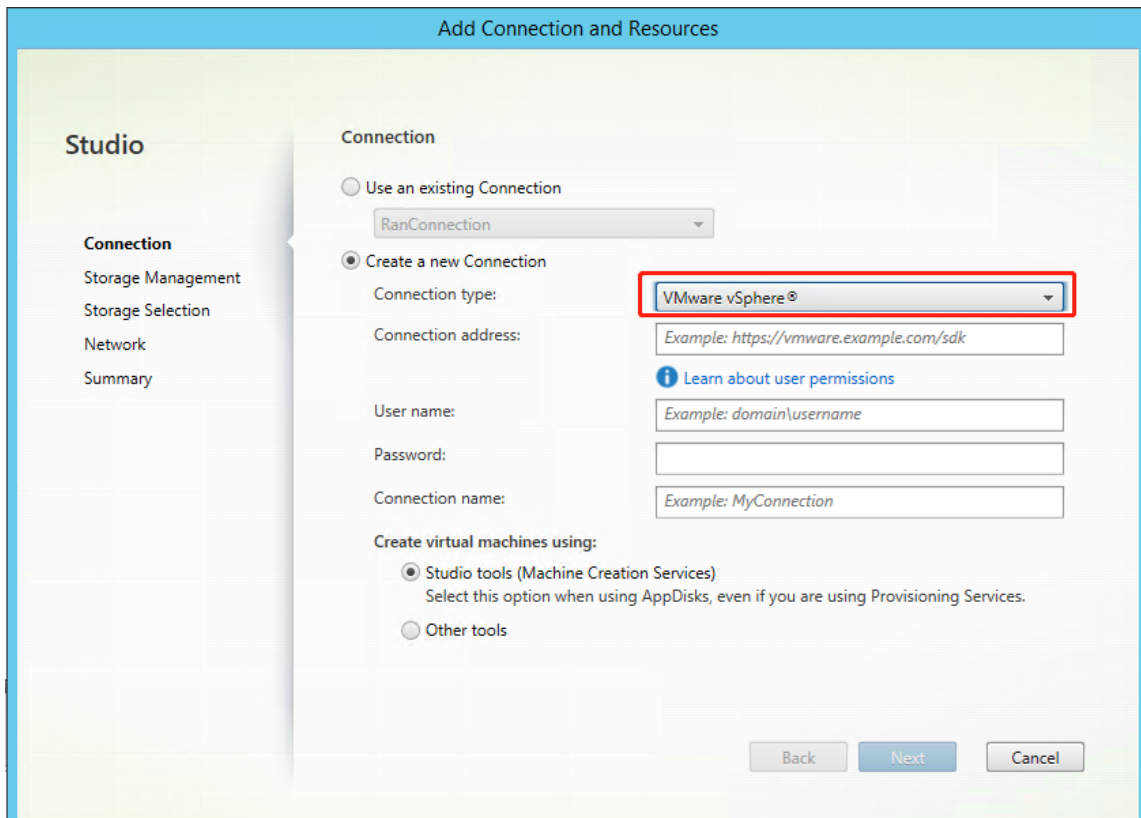
たとえば、オンプレミスの Citrix Studio では次のようになります：



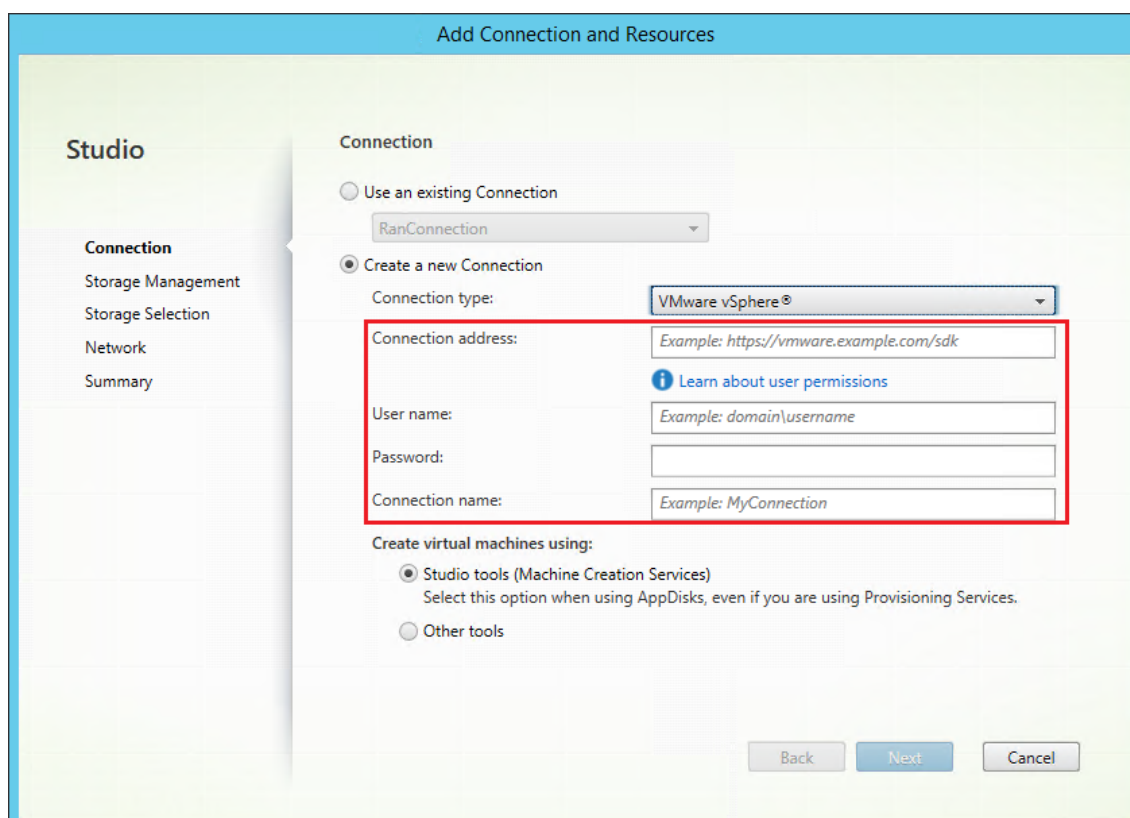
## Citrix Studio での VMware へのホスト接続の作成

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. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
3. 接続の種類として [VMware vSphere] を選択します。

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



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



### 手順 3: マスターイメージの準備

(XenServer の場合のみ) 手順 3a: **XenServer VM Tools** をインストールする xe CLI または XenCenter を使用するために、仮想マシンごとにテンプレート仮想マシンに XenServer VM Tools をインストールします。このツールがインストールされていないと、仮想マシンのパフォーマンスが低下する可能性があります。ツールがなければ、次のいずれも実行できません:

- 仮想マシンを正しくシャットダウン、再起動、または一時停止する。
  - XenCenter でその仮想マシンのパフォーマンスデータを表示する。
  - 実行中の仮想マシンを移行する (XenMotion を使用)。
  - スナップショットまたはメモリを含んだスナップショット (チェックポイント) を作成したり、スナップショットを復元したりする。
  - 実行中の Linux 仮想マシン上の vCPU の数を調整する。
1. [XenServer ダウンロードページ](#) または [Citrix Hypervisor ダウンロードページ](#) から、使用中のハイパーバイザーバージョンに応じた Linux 向け XenServer VM Tools ファイルをダウンロードします。
  2. `LinuxGuestTools-xxx.tar.gz` ファイルを、Linux 仮想マシン、または Linux 仮想マシンがアクセスできる共有ドライブにコピーします。
  3. `tar` ファイルの内容を展開します: `tar -xzf LinuxGuestTools-xxx.tar.gz`



4. 次のコマンドを実行して、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します。

**RHEL/CentOS/Rocky Linux/SUSE** の場合:

```
1 sudo rpm -i <extract-directory>/xe-guest-utilities_{
2   package-version }
3   _x86_64.rpm
```

**Ubuntu/Debian** の場合:

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

5. XenCenter の [全般] タブで、テンプレート仮想マシンの仮想化状態を確認します。XenServer VM Tools が正しくインストールされている場合、仮想化の状態は [最適化済み] を示します。

手順 **3b**: **AWS**、**Azure**、**GCP** の **SUSE 15.5** の構成を確認する AWS、Azure、および GCP の SUSE 15.5 の場合は、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。
- **/etc/sysconfig/windowmanager** の **Default\_WM** パラメーターが **gnome** に設定されている。

手順 **3c**: **GCP** 上の **Ubuntu 20.04** で **RDNS** を無効にする テンプレート仮想マシンで、**/etc/krb5.conf** の **[libdefaults]** に **rdns = false** 行を追加します。

手順 **3d**: テンプレート仮想マシンに **.NET** および **Linux VDA** パッケージをインストールする

注:

現在実行中の VDA をテンプレート仮想マシンとして使用するには、この手順を省略します。SSSD をテンプレート仮想マシンとして使用して、ドメインに接続された現在実行中の RHEL 8.x/9.x または Rocky Linux 8.x/9.x VDA を使用するには、次のことを確認してください:

- VDA は簡単インストールではなく手動でインストールします。簡単インストールは RHEL 8.x/9.x および Rocky Linux 8.x/9.x で **Adcli** を使用します。SSSD と **Adcli** の組み合わせは MCS ではサポートされていません。
- Samba サーバーは、AD 認証に SSSD を使用するよう構成されています。詳しくは、<https://access.redhat.com/solutions/3802321> の Red Hat の記事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、テンプレート仮想マシンに **.NET** をインストールし、以下について確認します:

- Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリビューションに、.NET ランタイムに加えて、ASP.NET Core ランタイムをインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。他のディストリビューションにはバージョン 8 が必要です。
- Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

.NET をインストールした後、Linux ディストリビューションに応じて次のコマンドを実行し、Linux VDA をインストールします：

#### **RHEL/CentOS/Rocky Linux** の場合：

注：

Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 にインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新します。

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

#### **Ubuntu/Debian** の場合：

```
1 sudo dpkg -i <PATH>/<Linux VDA DEB>
2
3 apt-get install -f
```

#### **SUSE** の場合：

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

(**RHEL 7** の場合のみ) 手順 **3e**：リポジトリを有効にして **tdb-tools** パッケージをインストールする **RHEL 7** サーバーの場合：

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

#### **RHEL 7** ワークステーションの場合：

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

手順 **3f**：(**RHEL** の場合のみ) **ntfs-3g** を提供できる **EPEL** リポジトリをインストールする **RHEL 8** に EPEL リポジトリをインストールします。EPEL のインストール方法については、<https://docs.fedoraproject.org/en-US/epel/> の説明を参照してください。

手順 **3g**: (**SUSE** の場合のみ) **ntfs-3g** を手動でインストールする SUSE プラットフォームには、ntfs-3g を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、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:

```
1 sudo zypper install gcc
2 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 パッケージを展開します。

```
1 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:

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

5. ntfs-3g をインストールします:

```
1 ./configure
2 make
3 make install
```

手順 **3h**: 使用するデータベースを指定する Linux VDA パッケージをインストールした後は、SQLite と PostgreSQL を切り替えることができます。このためには、次の手順を実行します:

注:

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで 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。

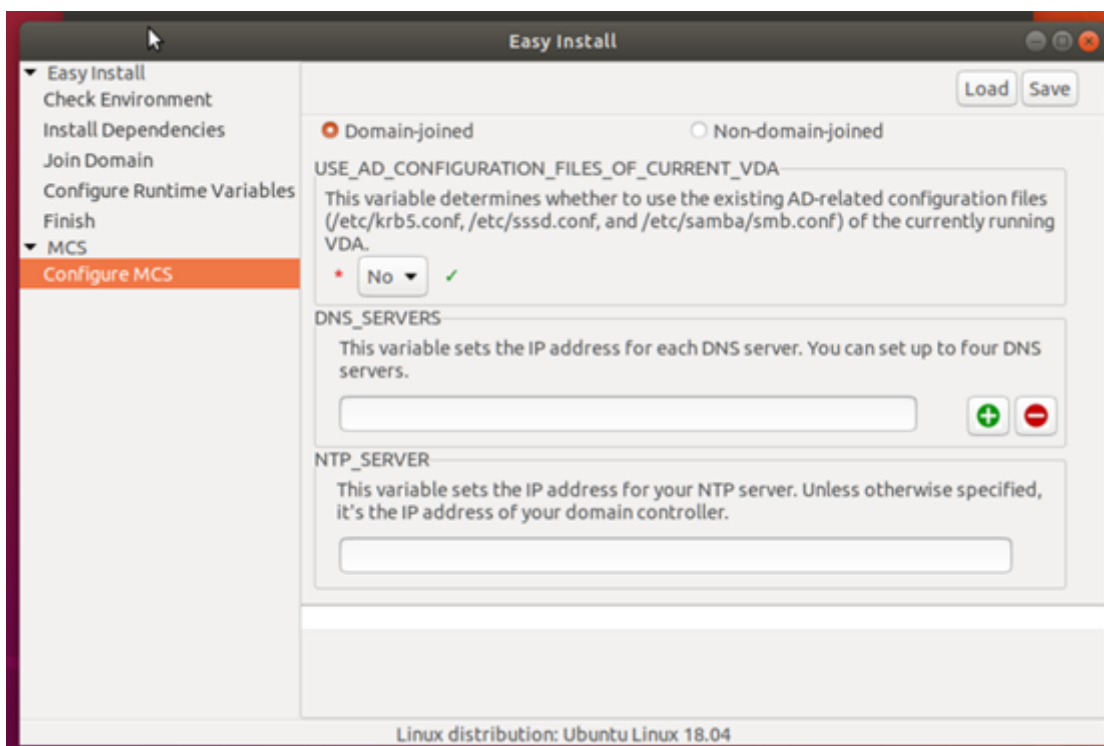
1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省きます。

2. **deploymcs.sh** を実行する前に **/etc/xdl/db.conf** を編集します。

手順 **3i**: **MCS** 変数を構成する MCS 変数を構成するには、次の 2 つの方法があります:

- **/etc/xdl/mcs/mcs.conf** ファイルを編集します。

- 簡単インストールの GUI を使用します。簡単インストールの GUI を開くには、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。



ヒント:

[保存] をクリックすると、指定したパスにあるローカルファイルに変数設定が保存されます。[読み込み] をクリックすると、指定したファイルから変数設定が読み込まれます。

以下は、ドメイン非参加シナリオとドメイン参加済みシナリオで構成できる MCS 変数です:

- ドメイン非参加シナリオの場合

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`

- ドメイン参加済みシナリオの場合

- `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/sss.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

の構成ファイルと同じファイルになります。ただし、`dns`変数と`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、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。現在実行中の VDA をテンプレート仮想マシン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、値を 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。特に指定のない限り、これはドメインコントローラーの IP アドレスです。
- `WORKGROUP`: ワークグループ名を、AD で構成した NetBIOS 名（大文字と小文字を区別）に設定します。設定しなかった場合、MCS はマシンのホスト名の直後に続くドメイン名の部分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、ワークグループ名として `citrix` が正しい選択であるにもかかわらず、MCS は `lvda` を使用することになります。ワークグループ名を正しく設定するようにしてください。
- `AD_INTEGRATION`: SSSD、Winbind、PBIS、または Centrify を設定します。Linux ディストリビューションのマトリックスと MSC がサポートするドメイン参加方法については、この記事の「サポートされているディストリビューション」を参照してください。
- `CENTRIFY_DOWNLOAD_PATH`: Server Suite Free（旧称 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の更新を自動化](#)」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
```

(オプション) 手順 **3j**: **MCS** のレジストリ値を書き込むまたは更新する テンプレートマシンで、コマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加して、必要なレジストリ値を作成または更新します。この操作によって、MCS でプロビジョニングされたマシンを再起動するたびにデータと設定が失われなないようにします。

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

たとえば、次のそれぞれのコマンドラインを `/etc/xdl/mcs/mcs_local_setting.reg` ファイルに追加して、レジストリ値を作成または更新できます：

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

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

#### 注

MCS の設定を変更する場合、「`/etc/xdl/ad_join`」および「`/etc/xdl/mcs/`」のファイルの編集は許可されていますが、「`/var/xdl/mcs`」のファイルの編集は許可されていません。

手順 **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 のみの場合) `update-crypto-policies --set DEFAULT:AD-SUPPORT` コマンドを実行してテンプレート仮想マシンを再起動します。
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よりも優先されます。
3. (現在実行中の VDA をテンプレート仮想マシンとして使用している、またはドメイン非参加のシナリオである場合は、この手順をスキップしてください。) テンプレート仮想マシン上で、構成テンプレートを更新して、作成されたすべての仮想マシン上の関連する `/etc/krb5.conf` ファイル、`/etc/samba/smb.conf` ファイル、および `/etc/sss/sss.conf` ファイルをカスタマイズします。

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

SSSD ユーザーの場合、`/etc/xdl/ad_join/sss.conf.tpl`、`/etc/xdl/ad_join/sss_krb5.conf.tpl`、および`/etc/xdl/ad_join/sss_smb.conf.tpl`テンプレートを更新します。

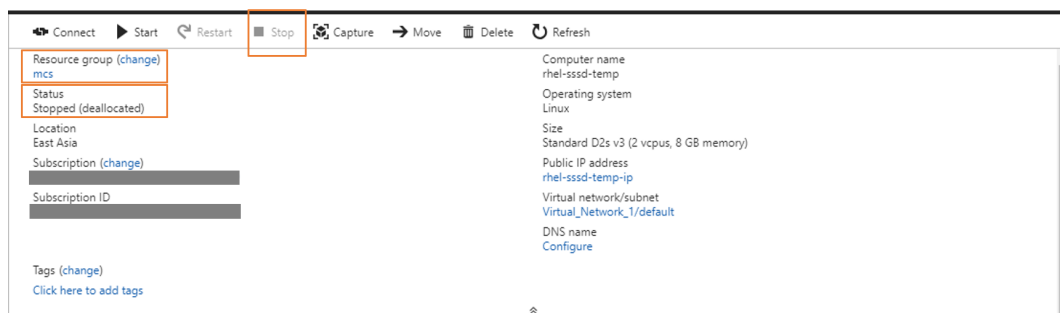
Centrify ユーザーの場合、`/etc/xdl/ad_join/centrify_krb5.conf.tpl`および`/etc/xdl/ad_join/centrify_smb.conf.tpl`テンプレートを更新します。

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保持し、`$WORKGROUP`、`$REALM`、`$realm`、`${new_hostname}`、および `$AD_FQDN` などの変数を使用してください。

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

- **(XenServer、GCP、および VMware vSphere の場合)** テンプレート仮想マシンにアプリケーションをインストールし、テンプレート仮想マシンをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。
- **(Azure の場合)** テンプレート仮想マシンにアプリケーションをインストールし、Azure Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンの電源状態が、**[Stopped (deallocated)]** になっ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検索する際に名前が必要です。



- **(AWS の場合)** テンプレート仮想マシンにアプリケーションをインストールし、AWS EC2 Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンのインスタンス状態が、**[Stopped]** になっていることを確認します。テンプレート仮想マシンを右クリックし、**[Image] > [Create Image]** を選択します。必要に応じて情報を入力し、設定を行います。**[Create Image]** をクリックします。

**Create Image**
✕

Instance ID ⓘ i-011f...

Image name ⓘ

Image description ⓘ

No reboot ⓘ

**Instance Volumes**

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/sda1	snap-02...	40	General Purpose SSD (gp2)	120 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Total size of EBS Volumes: 40 GiB  
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Cancel
Create Image

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

注:

Citrix Virtual Apps and Desktops で使用するには、Acropolis スナップショット名を「XD\_」で始める必要があります。必要に応じて、Acropolis コンソールを使用してスナップショットの名前を変更します。スナップショットの名前を変更したら、カタログ作成ウィザードを再起動して、更新された一覧を取得します。

**(GCP の場合)** 手順 **3I: RHEL 8.x/9.x** および **Rocky Linux 8.x/9.x** でイーサネット接続を構成する GCP でホストされている RHEL 8.x/9.x および Rocky Linux 8.x/9.x に Linux VDA をインストールすると、イーサネット接続が失われ、仮想マシンの再起動後に Linux VDA にアクセスできなくなることがあります。この問題を回避するには、仮想マシンに初めてログオンするときにルートパスワードを設定し、ルートとして仮想マシンにログオンできることを確認します。次に、仮想マシンを再起動した後、コンソールで次のコマンドを実行します:

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```

#### 手順 4: マシンカタログの作成

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仮想マシンの数を指定します。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、以下に注意してください:

- Nutanix 固有の [コンテナ] ページで、前にテンプレート仮想マシン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ときの毎回の動作を指定できます。



The screenshot shows the 'Machine Catalog Setup' wizard. The left sidebar contains a list of steps: Introduction, Machine Type, Machine Management, Desktop Experience (highlighted with a purple circle and the number 4), Master Image, Storage and License Types, Virtual Machines, NICs, Disk Settings, Resource Group, Machine Identities, Domain Credentials, Scopes, WEM Optional, VDA Upgrade Optional, and Summary. The main content area is titled 'Desktop Experience' and contains the following text and options:

Which desktop experience do you want users to have?

- I want users to connect to a new (random) desktop each time they log on.
- I want users to connect to the same (static) desktop each time they log on.

Do you want to save any changes that the user makes to the desktop?

- Yes, create a dedicated virtual machine and save changes on the local disk.
- No, discard all changes and clear virtual desktops when the user logs off.

At the bottom of the wizard, there are three buttons: 'Back', 'Next', and 'Cancel'.

[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：

- ユーザーがログオンするたびに新しい（ランダム）デスクトップに接続します。
- ユーザーが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すると、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 つ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、デスクトップへのユーザーによる変更の処理方法を構成できます：

- ユーザーによる変更をローカルディスク上のデスクトップに保存します（永続的）。
- ユーザーがログオフしたらユーザーによる変更を破棄し、仮想デスクトップをクリアします（非永続的）。ユーザー個人設定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と、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。既存のマシンは引き続き元のマスターイメージを使用します。

詳しくは、[Citrix Virtual Apps and Desktops](#)ドキュメントおよび[Citrix DaaS](#)ドキュメントでマシンカタログの作成を参照してください。

注:

Nutanix 環境では、Delivery Controller でのマシンカタログの作成プロセスに大幅に時間がかかる場合、Nutanix Prism に移動し、「**Preparation**」というプレフィックスが付いたマシンの電源を手動でオンにします。このアプローチは、作成プロセスを継続するのに役立ちます。

### 手順 5: デリバリーグループの作成

デリバリーグループは、いくつかのマシンカタログから選択したマシンをグループ化したものです。それらのマシンを使用できるユーザーと、そのユーザーに提供するアプリケーションおよびデスクトップを指定します。

詳しくは、[Citrix Virtual Apps and Desktops](#)ドキュメントおよび[Citrix DaaS](#)ドキュメントでデリバリーグループの作成を参照してください。

注:

MCS を使用して作成した仮想マシンは、Citrix Cloud Connector に登録できず、未登録として表示される場合があります。この問題は、Azure で仮想マシンをホストし、Samba Winbind を使用して AD ドメインに参加すると発生します。この問題を解決するには、以下の手順を実行します:

1. ADSI Edit コンソールに移動し、未登録の仮想マシンを選択して、そのマシンアカウントの **msDS-SupportedEncryptionTypes** 属性を編集します。
2. 仮想マシンで **ctxjproxy** および **ctxvda** サービスを再起動します。仮想マシンの状態が登録済みになった場合は、手順 3~5 を実行します。
3. テンプレート仮想マシンで **/var/xdl/mcs/ad\_join.sh** ファイルを開きます。
4. **/var/xdl/mcs/ad\_join.sh** ファイルで次の行の後に **net ads encytypes set \$NEW\_HOSTNAME\$** < 暗号化の種類属性の 10 進数値、例: **28** > **-U \$NEW\_HOSTNAME\$ -P password** の行を追加します:

```
1 if [ "$AD_INTEGRATION" == "winbind" ]; then
2     join_domain_samba
3     restart_service winbind /usr/bin/systemctl
```

5. 新しいスナップショットを作成し、新しいテンプレートを使用して仮想マシンを作成します。

### MCS を使用した Linux VDA のアップグレード

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 を最新リリースにアップグレードする前に、.NET がインストールされていることを確認してください。

- Amazon Linux 2 を除く、サポートされているすべての Linux ディストリビューションに .NET Runtime 8.0 をインストールします。
- Amazon Linux 2 の場合は、.NET Runtime 6.0 のインストールを続行します。

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

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：

- Azure を使用した Linux VDA の自動更新機能を使用して、ソフトウェアの自動更新をスケジューリングすることもできます。これを行うには、テンプレートマシン上の `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：

```
1 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_DWORD" -v "fEnabled" -d "0x00000001" --force
2
3 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v "ScheduledTime" -d "Immediately" --force
4
5 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v "Url" -d "<Azure コンテナの URL>" - force
6
7 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v "CaCertificate" -d "<PortalAzureComのローカル証明書パス>" --force
```

- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイルを処理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。以下では、rpm と dpkg が構成ファイルの変更でどのように機能するかについて説明しています：

- **rpm**： デフォルトでは、ローカルバージョンが保持され、パッケージから新しいバージョンが `rpmnew` 拡張子で保存されます。
- **dpkg**： どのように続行するかを選択するよう対話形式で促します。ローカル構成ファイルを保持し、新しいパッケージバージョンを **dpkg-new** または **dpkg-dist** として保存しながら Linux VDA を自動アップグレードするには、次のコマンドを使用します：

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

**RHEL** および **Rocky Linux** ディストリビューションの場合:

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

注:

Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 で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 以降に更新します。

**SUSE** ディストリビューションの場合:

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

**Ubuntu/Debian** ディストリビューション:

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2
3 sudo apt-get install -f
```

3. `/etc/xdl/mcs/mcs.conf`と`/etc/xdl/mcs/mcs_local_setting.reg`を編集します。
4. 新しいスナップショットを作成します。
5. Citrix Studio で新しいスナップショットを選択し、マシンカタログを更新します。各マシンが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の更新を自動化

マシンアカウントのパスワードは、デフォルトではマシンカタログの作成後 30 日で有効期限切れになります。パスワードの有効期限を無効にし、マシンアカウントのパスワードの更新を自動化するには、以下を実行します:

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

```
UPDATE_MACHINE_PW="Y"
```

2. `/opt/Citrix/VDA/sbin/deploymcs.sh` を実行後、`/etc/cron.d/mcs_update_password_cronjob` を開いて更新の時刻と頻度を設定します。デフォルトの設定では、マシンアカウントのパスワードを毎週日曜日、午前 2 時 30 分に更新します。

各マシンアカウントのパスワードの更新後、Delivery Controller のチケットキャッシュが無効になり、次のエラーが`/var/log/xdl/jproxy.log`に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、定期的にチケットキャッシュを消去します。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のクリーンアップタスクをスケジュールできます。

## MCS が作成した仮想マシンで FAS を有効化

次のディストリビューションで実行される MCS で作成した仮想マシンで FAS を有効にできます：

	Winbind	SSSD	Centrify	PBIS
RHEL 9.4/9.3/9.2	はい	いいえ	いいえ	いいえ
RHEL 8.x	はい	いいえ	いいえ	はい
Rocky Linux 9.4/9.3/9.2	はい	いいえ	いいえ	いいえ
Rocky Linux 8.x	はい	いいえ	いいえ	いいえ
Ubuntu 22.04、 Ubuntu 20.04	はい	いいえ	いいえ	いいえ
Debian 12.5/11.9	はい	はい	いいえ	はい
SUSE 15.5	はい	いいえ	いいえ	いいえ

テンプレート仮想マシンでマスターイメージを準備するときに **FAS** を有効にする

1. ルート CA 証明書をインポートします。

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

2. 簡単インストールスクリプト (ctxinstall.sh) を実行し、テンプレート仮想マシンがドメインに適切に参加していることを確認します。
3. [ctxfascfg.sh](#) を実行します。
4. `/etc/xdl/mcs/mcs.conf` に変数を設定します。

注：

`/etc/xdl/mcs/mcs.conf` に必要なすべての変数を設定します。これらの変数は仮想マシン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) `VDI_MODE`など、必要に応じて他の変数を設定します。
5. FAS サーバーのアドレスを設定するために、以下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:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialService" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

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

### MCS が作成した仮想マシンで 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は、MCS で作成された各仮想マシンで FAS を有効にできます。

MCS が作成した仮想マシンで FAS を有効にするには、次を実行します:

1. `/etc/xdl/mcs/mcs.conf` の変数を設定します。

注:

`/etc/xdl/mcs/mcs.conf` に必要なすべての変数を設定します。これらの変数は仮想マシン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) `VDI_MODE` など、必要に応じて他の変数を設定します。
2. FAS サーバーのアドレスを設定するために、以下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:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialService" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

3. ルート CA 証明書をインポートします。

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

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

## Citrix Provisioning を使用した Linux VDA の作成

August 20, 2024

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

ここでは、Linux ターゲットデバイスのストリーミングについて説明します。この機能を使用すると、Citrix Virtual Apps and Desktops 環境で直接 Linux 仮想デスクトップをプロビジョニングできます。

サポートされている Linux ディストリビューションは次のとおりです。

- Ubuntu 22.04
- Ubuntu 20.04
- RHEL 9.2
- RHEL 8.8
- Rocky Linux 9.2
- Rocky Linux 8.8
- SUSE 15.5

**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。使用する Linux ディストリビューションに応じたパッケージを使用します。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 サーバー 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は、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起動パーティションを作成します。
- パーティションを **btrfs** でフォーマットすることは避けてください。GRUB2 には、**btrfs** パーティションの検索で本質的な問題があります。**GRUB** は **GRand Unified Bootloader** の略です。

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

## Citrix DaaS Standard for Azure で Linux VDA を作成

August 20, 2024

Citrix DaaS Standard for Azure (Citrix Virtual Apps and Desktops Standard for Azure の新名称) でドメイン参加とドメイン非参加の両方の Linux VDA を作成して、Microsoft Azure から任意のデバイスに仮想アプリおよび仮想デスクトップを配信できます。詳しくは、「[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

次の Linux ディストリビューションはこの機能をサポートしています:

- RHEL 9.4

- RHEL 9.3
- RHEL 9.2
- RHEL 8.10
- RHEL 8.9
- RHEL 8.8
- Rocky Linux 9.4
- Rocky Linux 9.3
- Rocky Linux 9.2
- Rocky Linux 8.10
- Rocky Linux 8.9
- Rocky Linux 8.8
- SUSE 15.5
- Ubuntu 22.04
- Ubuntu 20.04

#### 手順 1: Azure でのマスターイメージの準備

注:

[Linux VDA の自動更新機能](#)を使用して、ソフトウェアの自動更新をスケジュールすることもできます。これを行うには、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 "" --force
```

```
create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v "Certificate" -d "" --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 仮想マシンを作成します。
2. 必要に応じて、Linux 仮想マシンにデスクトップ環境をインストールします。
3. 仮想マシンに、Linux ディストリビューションに応じた .NET をインストールします。

Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリ



ビューションに、.NET ランタイムに加えて ASP.NET Core ランタイムをインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。他のディストリビューションにはバージョン 8 が必要です。

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

4. (Ubuntu の場合のみ) `/etc/network/interfaces` ファイルに `source /etc/network/interfaces.d/*` 行を追加します。
5. (Ubuntu の場合のみ) `/etc/resolv.conf` で `/run/systemd/resolve/stub-resolv.conf` ではなく `/run/systemd/resolve/resolv.conf` を指定します:

```
1 unlink /etc/resolv.conf
2
3 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。

試験段階の機能として、PostgreSQL に加えて SQLite も使用できます。Linux VDA パッケージをインストールした後は、SQLite と PostgreSQL を切り替えることもできます。このためには、次の手順を実行します:

- a) `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールの場合、この手順は省きます。
- b) `deploymcs.sh` を実行する前に `/etc/xdl/db.conf` を編集します。

注:

- SQLite は VDI モードにのみ使用することをお勧めします。
- 簡単なインストールと MCS のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく切り替えることができます。`/etc/xdl/db.conf` で特に指定しない限り、Linux VDA はデフォルトで 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 変数を構成するには、次の 2 つの方法があります:

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

注:

`dns`変数は指定しないでください。

マシンカタログの作成時に静的タイプまたはランダムタイプを選択する場合は、`VDI_MODE=Y`を設定します。

`/etc/xdl/mcs/mcs.conf`を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、[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よりも優先されます。

- Azureで仮想マシンを停止（または割り当て解除）します。[ディスクのエクスポート]をクリックして、他の仮想マシン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。

- (オプション) マスターイメージでグループポリシーを設定します。`ctxreg`ツールを使用してグループポリシーを設定できます。たとえば、次のコマンドは、PDF印刷のPDFユニバーサルプリンターを自動作成するポリシーを有効にします。

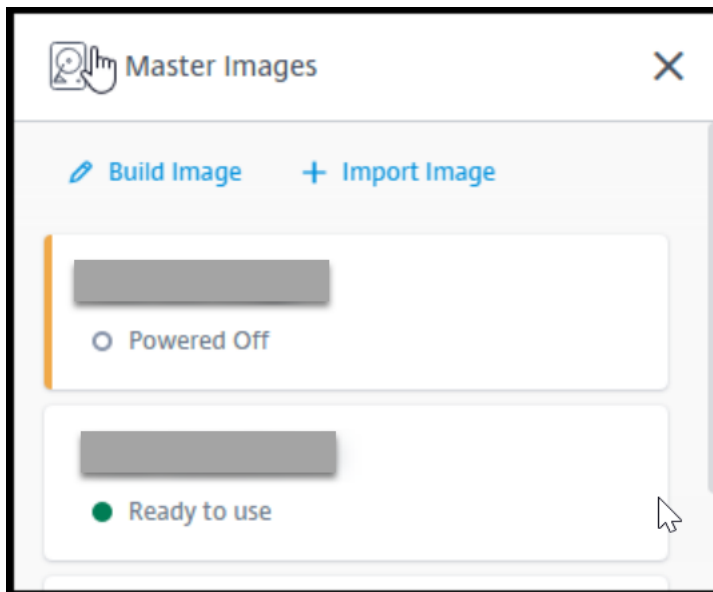
```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\GroupPolicy\Defaults\PrintingPolicies" -t "REG_DWORD" -v "AutoCreatePDFPrinter" -d "0x00000001" --force
```

## 手順 2: Azure からのマスターイメージのインポート

1. **[管理]** ダッシュボードから、右側の **[マスターイメージ]** を展開します。ディスプレイには、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一覧表示されます。

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、**[管理]** ダッシュボードと **[監視]** ダッシュボードで管理されます。最初のカatalogを作成後、Citrix Cloud にサインインして **[Managed Desktops]** サービスを選択すると、管理 ダッシュボードが自動的に起動します。



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

## Import Image from Azure

---

Enter the Azure-generated URL for the Virtual Hard Disk 

[How do I find my Uri?](#)

Master image type

- Windows  
 Linux

Name The New Master Image

4. ウィザードの指示に従い、マスターイメージをインポートします。

### 手順 3: マシンカタログの作成

**[管理]** ダッシュボードにアクセスし、**[カタログを作成する]** をクリックします。マシンカタログを作成するときは、上記で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仮想マシンには、SSH または RDP を介してアクセスすることはできません。仮想マシンにアクセスするには、Azure Portal のシリアルコンソールを使用します。

## Linux VDA の手動インストール

August 20, 2024

Linux VDA は、次の Linux ディストリビューションに手動でインストールできます:

- [Amazon Linux 2](#)、[CentOS](#)、[RHEL](#)、[Rocky Linux](#)
- [SUSE](#)
- [Ubuntu](#)
- [Debian](#)

## Amazon Linux 2、RHEL、および Rocky Linux への Linux VDA の手動インストール

August 20, 2024

**重要:**

新規インストールの場合は、[簡単インストール](#)を使用して簡易インストールを行うことをお勧めします。簡単インストールは時間と労力を節約するだけでなく、本記事に記載されている手動インストールよりもエラーを減らすことができます。

### 手順 1: 構成ファイル情報および Linux マシンの準備

#### 手順 1a: ネットワーク構成の確認

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

#### 手順 1b: ホスト名の設定

マシンのホスト名が確実に正しく報告されるようにするには、**/etc/hostname** ファイルを変更してマシンのホスト名のみを記述します。

`hostname`

#### 手順 1c: ホスト名へのループバックアドレスの割り当て

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が確実に正しく報告されるようにするには、**/etc/hosts** ファイルの以下の行を変更し、最初の 2 つのエントリとして完全修飾ドメイン名とホスト名を記述します:

```
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** に対するその他の参照を削除します。

注:

Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。ホスト名は 15 文字以内である必要があります。

ヒント:

a~z、A~Z、0~9、およびハイフン (-) の文字のみ使用してください。アンダースコア (\_)、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了し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。

### 手順 **1d**: ホスト名の確認

次のコマンドで、ホスト名が正しく設定されていることを確認します:

```
1 hostname
```

このコマンドにより、そのマシンの完全修飾ドメイン名 (FQDN) ではなく、そのホスト名のみが返されます。

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します:

```
1 hostname -f
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

### 手順 **1e**: 名前解決とサービス到達可能性の確認

次のコマンドで、完全修飾ドメイン名が解決できることと、ドメインコントローラーと Delivery Controller から ping に応答があることを確認します:

```
1 nslookup domain-controller-fqdn
2
3 ping domain-controller-fqdn
4
5 nslookup delivery-controller-fqdn
6
7 ping delivery-controller-fqdn
```

完全修飾ドメイン名を解決できない、またはこれらのマシンのいずれかから ping に応答がない場合は、手順を確認してから次に進んでください。

### 手順 **1f**: 時刻同期の構成

VDA、Delivery Controller、ドメインコントローラーの間で正確な時刻同期を維持することは重要です。仮想マシン (VM) として Linux VDA をホストすると、時刻が不正確になる問題が発生する可能性があります。したがって、リモートのタイムサービスを使用して時刻を維持することをお勧めします。

RHEL のデフォルト環境では、時刻同期に Chrony デーモン (`chronyd`) を使用します。

**Chrony** サービスの構成 ルートユーザーとして `/etc/chrony.conf` を編集し、次のように各リモートタイムサーバーに対応するサーバーエントリを追加します：

```
1 server peer1-fqdn-or-ip-address iburst
2
3 server peer2-fqdn-or-ip-address iburst
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応する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、パブリックサーバーの `*.pool.ntp.org` エントリなど、一覧にあるその他の `server` エントリを削除します。

変更を保存してから、次のコマンドで Chrony デーモンを再起動します：

```
1 sudo systemctl restart chronyd
```

#### 手順 1g: OpenJDK 11 のインストール

Linux VDA には、OpenJDK 11 が必要です。

- RHEL を使用している場合は、Linux VDA をインストールすると、依存関係として OpenJDK 11 が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 を有効にしインストールします：

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

正しいバージョンを確認します。

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

事前にパッケージされた 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：

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

#### 手順 1h: 使用するデータベースのインストールと指定

注：

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合は、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから**/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

このセクションでは、PostgreSQL と SQLite をインストールする方法と、どちらを使用するかを指定する方法について説明します。

**PostgreSQL** のインストール Linux VDA には PostgreSQL が必要です：

- PostgreSQL 9: Amazon Linux 2 の場合
- PostgreSQL 10: RHEL 8.x、Rocky Linux 8.x の場合
- PostgreSQL 13: RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 の場合

次のコマンドを実行して、PostgreSQL をインストールします：

```
1 sudo yum -y install postgresql-server
2
3 sudo yum -y install postgresql-jdbc
```

RHEL 8.x と RHEL 9.4/9.3/9.2 の場合、次のコマンドを実行して PostgreSQL の **libpq** をインストールします：

```
1 sudo yum -y install libpq
```

次のコマンドを実行して、データベースを初期化します。この操作により、**/var/lib/pgsql/data** にデータベースファイルが作成されます。

```
1 sudo postgresql-setup initdb
```

マシンの起動時または即時で PostgreSQL 起動するには、それぞれ次のコマンドを実行します：

```
1 sudo systemctl enable postgresql
2
3 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：

```
1 psql --version
```

(Amazon Linux 2 の場合のみ) 次のように **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：

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



**SQLite** をインストールする 次のコマンドを実行して SQLite をインストールします:

```
1 sudo yum -y install sqlite
```

使用するデータベースを指定する SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省きます。
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:

**/etc/xdl/db.conf** を使用して PostgreSQL のポート番号を構成することもできます。

## 手順 2: ハイパーバイザーの準備

サポートされるハイパーバイザー上で VM として Linux VDA を実行する場合、いくつかの変更が必要です。使用するハイパーバイザーのプラットフォームに合わせて、次の変更を行います。ベアメタルハードウェアで Linux マシンを実行する場合、変更は必要ありません。

## XenServer (旧称 Citrix Hypervisor) での時刻同期の修正

XenServer の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP と XenServer の問題が発生します。これは、NTP と Citrix Hypervisor の両方がシステムの時間を管理しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは、変更は必要ありません。

XenServer 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、XenServer の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
1 su -
2
3 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いるため、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います。こ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、以下の手順は必要ありません。

時刻同期機能が有効になっている場合は、ファイルに「1」と書き込んで無効にします:

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

この変更を永続化し、再起動後も保持するには、**/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ため、次のようにしてシステムを再起動します：

```
1 su -
2
3 cat /proc/sys/xen/independent_wallclock
```

このコマンドは 1 を返します。

### Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想ホストでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともに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で、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、[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：

この方法は VMware および XenServer (旧称 Citrix Hypervisor) の場合とは異なります。VMware および XenServer では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期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な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーで問題が発生します。これは、NTP とハイパーバイザーの両方がシステムの時間を同期しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools をインストールした状態で準仮想化 Linux カーネルを実行している場合は、次の操作を行います。

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. [仮想マシンのプロパティ] ダイアログボックスで、[オプション] タブをクリックします。
4. [VMware Tools] を選択します。
5. [詳細] ボックスで、[ホストとゲスト時刻を同期] チェックボックスをオフにします。

### 手順 3: Linux VM を Windows ドメインに追加

以下は、Linux マシンを Active Directory (AD) ドメインに追加する方法です:

- [Samba Winbind](#)
- [Quest Authentication Services](#)
- [Centrify DirectControl](#)
- [SSSD](#)
- [PBIS](#)

選択した方法の手順に従います。

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ことがあります。

#### Samba Winbind

RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2, の場合、次のコマンドを実行して、**pam\_winbind** がルートディレクトリの所有権を変更しないようにします:

```
1 usermod -d /nonexistent nobody
```

次のようにして、必要なパッケージをインストールまたは更新します:

RHEL 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x の場合:

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

Amazon Linux 2 の場合:

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

マシンの起動時に **Winbind** デーモンを開始できるようにする 次のコマンドで、マシン起動時に Winbind デーモンが開始するように構成する必要があります:

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

**Winbind** 認証の構成 次のようにして、Winbind を使用した 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 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x の場合:

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

Amazon Linux 2 の場合:

```
1 sudo authconfig --disablecache --disableldap --disableldapauth --
  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 領域名で、**domain** はドメインの NetBIOS 名です。

KDC サーバーおよび領域名を DNS ベースで参照する必要がある場合は、次の 2 つのオプションを前述のコマンドに追加します:

```
--enablekrb5kdcdns --enablekrb5realmdns
```

`authconfig` コマンドから返される、開始に失敗した `winbind` サービスに関するエラーは無視します。これらのエラーは、マシンがドメインにまだ参加していない状態で `authconfig` が `winbind` サービスを開始しようとするとき発生することがあります。

2. `/etc/samba/smb.conf` を開いて、[Global] セクションに次のエントリを追加します。ただし、追加するのは、`authconfig` ツールによって生成されたセクションの後です:

```
kerberos method = secrets and keytab
winbind refresh tickets = true
winbind offline logon = no
```

3. (RHEL 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x の場合のみ) `/etc/krb5.conf` を開いて、[libdefaults]、[realms]、[domain\_realm] セクションにエントリを追加します:

[libdefaults] セクション:

```
default_ccache_name = FILE:/tmp/krb5cc_%{ uid }
default_realm = REALM
dns_lookup_kdc = true
```

[realms] セクション:

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

[domain\_realm] セクション:

```
realm = REALM
.realm = REALM
```

Delivery Controller に対する認証と登録には、Linux VDA にシステムの keytab ファイル `/etc/krb5.keytab` が必要です。前述の `kerberos` を使用した設定により、マシンが初めてドメインに参加するときに、`Winbind` によって

システムの `keytab` ファイルが強制的に作成されます。

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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 仮想マシンを Windows ドメインに追加するには、次のコマンドを実行します。

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

ヒント:

Amazon Linux 2 で実行されている Linux 仮想マシンの場合は、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:

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

**REALM** は大文字の Kerberos 領域名で、**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** 用の **PAM** の構成 デフォルトでは、Winbind PAM モジュール (`pam_winbind`) の構成で、Kerberos チケット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` を開いて、`[Global]` セクションで次のとおりにエントリを追加または変更します:

```
krb5_auth = yes
krb5_ccache_type = FILE
mkhomedir = yes
```

各設定の先頭のセミコロンが削除されていることを確認します。これらを変更するには、次のようにして Winbind デーモンを再起動する必要があります:

```
1 sudo systemctl restart winbind
```

ヒント:

マシンがドメインに参加済みの場合にのみ、`winbind` デーモンは実行を続けます。

`/etc/krb5.conf` を開いて、`[libdefaults]` セクションで次の設定を `KEYRING` から `FILE` タイプに変更します:

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

RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 の場合、次のコマンドを実行して Winbind で SELinux の問題を解決します:

```
1 ausearch -c 'winbindd' --raw | audit2allow -M my-winbindd -p /etc/
  selinux/targeted/policy/policy.*
2
3 semodule -X 300 -i my-winbindd.pp
```

ドメインメンバーシップの確認 Delivery Controllerを使用するには、すべてのVDAマシン (WindowsとLinux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。

次のように、**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：

```
1 sudo net ads testjoin
```

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

```
1 sudo net ads info
```

**Kerberos** 構成の確認 Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドにより、システムの keytab ファイルが作成済みで keytab ファイルに有効なキーが含まれていることを確認します：

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します：

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します：

```
1 sudo klist
```

次のコマンドを使用して、マシンアカウントの詳細を調査します：

```
1 sudo net ads status
```

ユーザー認証の確認 次のように、**wbinfo** ツールを使用して、ドメインユーザーがドメインに対して認証できることを確認します：

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

ここで指定するドメインは AD ドメイン名で、Kerberos 領域名ではありません。bash シェルの場合、バックスラッシュ文字 (\) は、もう 1 つバックスラッシュ文字を指定してエスケープする必要があります。このコマンドにより、成功または失敗を示すメッセージが返されます。

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

```
1 ssh localhost -l domain\username
2 id -u
```

次のコマンドで、Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

```
1 klist
```

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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。

## Quest Authentication Services

ドメインコントローラーでの **Quest** の構成 次の操作は、Quest ソフトウェアを Active Directory ドメインコントローラーにインストールし、構成していることと、管理者特権が付与され、Active Directory にコンピューターオブジェクトを作成できることを前提としています。

**Linux VDA** マシンにドメインユーザーがログオンできるようにする Linux VDA マシンで HDX セッションを確立する必要がある各ドメインユーザーに対して、次の操作を行います。

1. [Active Directory ユーザーとコンピューター] 管理コンソールで、目的のユーザーアカウントの Active Directory ユーザーの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] チェックボックスをオンにします。
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：

この手順は、ドメインユーザーがコンソール、RDP、SSH、またはその他のリモート処理プロトコルを使用してログオンできるように設定する場合も同じです。

## Linux VDA での **Quest** の構成

**SELinux** ポリシー適用の回避策 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 ドメインソケットの IPC のメカニズムに干渉し、ドメインユーザーのログオンを妨げます。

この問題を回避するための便利な方法は、SELinux の無効化です。ルートユーザーとして、**/etc/selinux/config** を編集し、**SELinux** 設定を次のとおりに変更します：

## SELINUX=permissive

この変更にはマシンの再起動が必要です:

```
1 reboot
```

### 重要:

この設定は注意して使用してください。SELinux ポリシーの適用を無効にした後に再度有効にすると、ルートユーザーやその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てしまう可能性があります。

**VAS** デーモンの構成 次のように Kerberos チケットの自動更新を有効にして、切断する必要があります。認証（オフラインログイン）は無効にする必要があります。

```
1 sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-  
   interval 32400  
2  
3 sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-  
   auth false
```

このコマンドにより、更新間隔が9時間（32,400秒）に設定されます。すなわち、チケットのデフォルトの有効期間である10時間よりも1時間短くなります。チケットの有効期間がさらに短いシステムでは、より小さい値をこのパラメーターに設定します。

**PAM** および **NSS** の構成 HDX や、su、ssh、RDP などのその他のサービスを介したドメインユーザーのログイ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:

```
1 sudo /opt/quest/bin/vastool configure pam  
2  
3 sudo /opt/quest/bin/vastool configure nss
```

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

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

**user** は、コンピューターを Active Directory ドメインに追加す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（example.com など）です。

ドメインに追加後、Linux マシンを再起動します。

ドメインメンバーシップの確認 Delivery Controller を使用するには、すべての VDA マシン（Windows と Linux VDA）で **Active Directory** にコンピューターオブジェクトが必要です。Quest によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します:



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

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下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** PAMを使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\username  
2 id -u
```

次のコマンドで、**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:

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します:

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

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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。

## Centrify DirectControl

**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、次のように Centrify の **adjoin** コマンドを使用して、Linux マシンを Active Directory ドメインに追加します:

```
1 su -  
2 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 ドメインに追加する権限を持つ任意の Active Directory ドメインユーザーです。**domain-name** は、Linux マシンを追加す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を使用するには、すべてのVDAマシン (WindowsとLinux VDA) でActive Directoryにコンピューターオブジェクトが必要です。Centrifyにより追加されたLinuxマシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 su -
2 adinfo
```

Joined to domain 値が有効であることと、CentrifyDC mode で connected が返されることを確認します。CentrifyDC mode が starting のまま変化しない場合は、Centrifyクライアントにサーバーとの接続の問題、または認証の問題が発生しています。

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
1 adinfo --sysinfo all
2 adinfo -diag
```

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

```
1 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、このセクションの指示に従ってください。このセクションでは、Linux VDAマシンのWindowsドメインへの参加手順、およびKerberos認証の構成について説明します。

SSSDをRHELでセットアップするには、次の作業を行います:

1. ドメインに参加してホストのkeytabを作成
2. SSSDのセットアップ
3. SSSDの有効化
4. Kerberos構成の確認
5. ユーザー認証の確認

ドメインに参加してホストの**keytab**を作成 SSSDでは、ドメイン参加とシステムのkeytabファイルの管理に関するActive Directoryのクライアント機能が提供されていません。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の場合の**Samba**のアプローチと、RHEL 8.x/9.xとRocky Linux 8.x/9.xの場合の**adcli**のアプローチについて説明します。**realmd**に関しては、RHELのドキュメントを参照してください。SSSDを構成する前に、以下の手順に従う必要があります。

- **Samba (Amazon Linux 2):**

次のようにして、必要なパッケージをインストールまたは更新します:

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

Linux クライアントで、適切に構成されたファイルを使用します：

- /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：

```
1 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 領域名で、**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。AD インフラストラクチャに基づいて Kerberos を構成します。

KDC サーバーおよび領域名を DNS ベースで参照する必要がある場合は、次の 2 つのオプションを前述のコマンドに追加します：

```
--enablekrb5kdcdns --enablekrb5realmdns
```

**/etc/samba/smb.conf** を開いて、**[Global]** セクションに次のエントリを追加します。ただし、追加するのは、**authconfig** ツールによって生成されたセクションの後です：

```
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できることと、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ます：

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

**REALM** は大文字の Kerberos 領域名で、**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。

• **Adcli (RHEL 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x)：**

次のようにして、必要なパッケージをインストールまたは更新します：

```
1 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：

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

**/etc/krb5.conf** を開いて、[realms] および [domain\_realm] セクションにエントリを追加します。

[realms] セクション:

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

[domain\_realm] セクション:

```
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できることと、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ます:

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

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

**SSSD** のセットアップ SSSD のセットアップは、以下の手順で構成されています:

- `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に構成の変更を行う (sssd.conf など)。
- **sssd** サービスを開始します。

(RHEL 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x のみ)

**/etc/sss/sss.conf** を開いて、[domain/ad.example.com] セクションに次のエントリを追加します:

```
ad_gpo_access_control = permissive  
full_name_format = %2$s\\%1$s  
fallback_homedir = /home/%d/%u  
# Kerberos settings  
krb5_ccachedir = /tmp  
krb5_ccname_template = FILE:%d/krb5cc_%U
```

**ad.example.com** と **server.ad.example.com** を対応する値で置き換えます。詳しくは、「[sssd-ad\(5\) - Linux man page](#)」を参照してください。

ファイルの所有権およびアクセス権を sssd.conf で設定します:

```
chown root:root /etc/sss/sss.conf
chmod 0600 /etc/sss/sss.conf
restorecon /etc/sss/sss.conf
```

**SSSD** の有効化 **RHEL 9.4/9.3/9.2/8.x** および **Rocky Linux 9.4/9.3/9.2/8.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:

```
1 sudo systemctl restart sssd
2 sudo systemctl enable sssd.service
3 sudo chkconfig sssd on
```

**Amazon Linux 2** の場合:

**authconfig** を使用して SSSD を有効にします。**odjjob-mkhomedir** をインストールして、このホームディレクトリの作成機能が SELinux に対応していることを確認します:

```
1 authconfig --enablesssd --enablesssdauth --enablemkhomedir --update
2
3 sudo systemctl start sssd
4
5 sudo chkconfig sssd on
```

**Kerberos** 構成の確認 システムの **keytab** ファイルが作成され、このファイルに有効なキーが含まれていることを確認します:

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します:

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します:

```
1 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いること、および NSS が機能するかを確認します:

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

**DOMAIN** パラメーターは短い形式のドメイン名です。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通りです：

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

SSSD PAM モジュールが正しく構成されていることを確認するには、以前に使用したことが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。

```
1 sudo ssh localhost -l DOMAIN\username
2
3 id -u
```

次のコマンドで、次の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：

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

次のコマンドで、ユーザーの Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します。

```
1 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

```
1 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

```
1 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

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

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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：

```
1 /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 (example.com など) です。

注: Bash をデフォルトのシェルとして設定するには、**/opt/pbis/bin/configLoginShellTemplate/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 を使用するには、すべての VDA マシン (Windows と Linux VDA) で Active Directory にコンピューターオブジェクトが必要です。PBIS によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 /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の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

**ユーザー認証の確認** PAM を使用した PBIS のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\user
2
3 id -u
```

次のコマンドで、**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:

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、セッションを終了します。

```
1 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 をインストールする

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

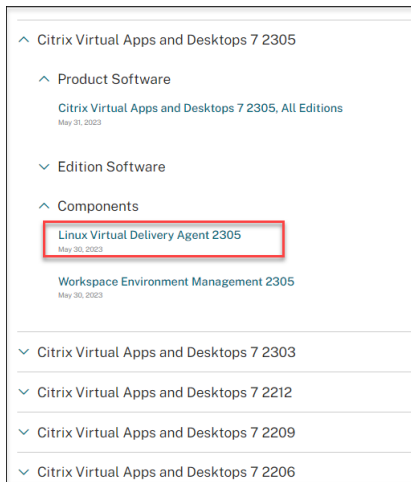
Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

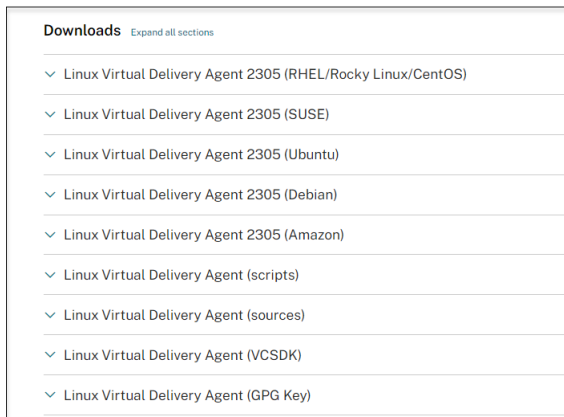
コマンド出力に基づいて、.NET ランタイムのバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnet の場合、/aa/bb を .NET バイナリパスとして使用します。

### 手順 5: **Linux VDA** パッケージのダウンロード

1. [Citrix Virtual Apps and Desktops のダウンロードページ](#)にアクセスします。
2. 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
3. **Components** を展開して Linux VDA を見つけます。例:

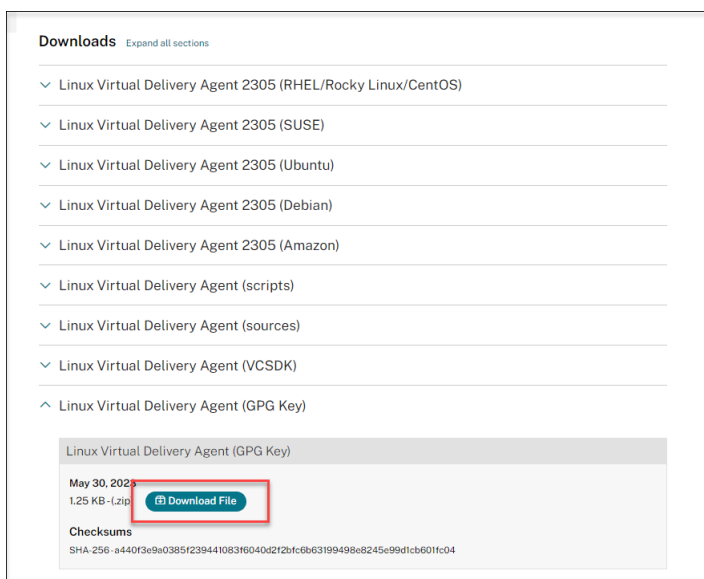


4. Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



5. 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
6. Linux VDA パッケージの整合性を検証するために使用できる GPG 公開キーをダウンロードします。例:





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

```
1 rpmkeys --import <path to the public key>
2 rpm --checksig --verbose <path to the Linux VDA package>
```

### 手順 6: Linux VDA のインストール

新規インストールを実行することも、既存のインストールをアップグレードすることもできます。Linux VDA は、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDA を 2308 から 2311 に、および 1912 LTSR から 2203 LTSR にアップグレードできます。

#### 手順 6a: 新規インストールを実行する

##### 1. (オプション) 古いバージョンのアンインストール

最新の 2 バージョンおよび LTSR リリース以外の古いバージョンの Linux VDA がインストールされている場合は、それをアンインストールしてから新しいバージョンをインストールする必要があります。

a) 次のコマンドで、Linux VDA サービスを停止します：

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

注：

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停

止したサービスを再起動します。

b) 次のコマンドで、パッケージをアンインストールします：

```
1 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。代わりに、システムパスに **/opt/Citrix/VDA/sbin** および **/opt/Citrix/VDA/bin** を追加することもできます。

## 2. Linux VDA パッケージのダウンロード

[Citrix Virtual Apps and Desktops のダウンロードページ](#)にアクセスします。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、使用中の Linux ディストリビューションに対応する Linux VDA パッケージをダウンロードします。

## 3. Linux VDA のインストール

注：

- RHEL および Rocky Linux の場合、EPEL リポジトリを先にインストールしておかないと、Linux VDA を正常にインストールできません。EPEL のインストール方法については、<https://docs.fedoraproject.org/en-US/epel/>の説明を参照してください。
- Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 にインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新します。

- 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：

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

**RHEL 9.4/9.3/9.2** および **Rocky Linux 9.4/9.3/9.2** の場合：

```
1 sudo yum install -y XenDesktopVDA-<version>.el9_x.x86_64.rpm
```

**RHEL 8.x** および **Rocky Linux 8.x** の場合：

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

- RPM Package Manager を使用して、Linux VDA ソフトウェアをインストールします。その前に、次の依存関係を解決する必要があります。

**Amazon Linux 2** の場合：

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

**RHEL 9.4/9.3/9.2** および **Rocky Linux 9.4/9.3/9.2** の場合：

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

**RHEL 8.x** および **Rocky Linux 8.x** の場合:

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

**RPM** 依存関係一覧 (**RHEL 9.4/9.3/9.2** および **Rocky Linux 9.4/9.3/9.2** の場合):

```
1 gtk2 >= 2.24.33
2
3 java-11-openjdk >= 11
4
5 tzdata-java >= 2022
6
7 ImageMagick >= 6.9
8
9 firewalld >= 0.6.3
10
11 polycoreutils-python-utils >= 2.8
12
13 python3-polycoreutils >= 2.8
14
15 dbus >= 1.12.8
16
17 dbus-common >= 1.12.8
18
19 dbus-daemon >= 1.12.8
20
21 dbus-tools >= 1.12.8
22
23 dbus-x11 >= 1.12.8
24
25 xorg-x11-server-utils >= 7.7
26
27 xorg-x11-xinit >= 1.3.4
28
29 libXpm >= 3.5.12
30
31 libXrandr >= 1.5.1
32
33 libXtst >= 1.2.3
34
35 pam >= 1.3.1
36
37 util-linux >= 2.32.1
38
39 util-linux-user >= 2.32.1
40
41 xorg-x11-utils >= 7.5
42
43 bash >= 4.3
44
45 findutils >= 4.6
```

```
46
47 gawk >= 4.2
48
49 sed >= 4.5
50
51 cups >= 1.6.0
52
53 cups-filters >= 1.20.0
54
55 ghostscript >= 9.25
56
57 libxml2 >= 2.9
58
59 libmspack >= 0.7
60
61 ibus >= 1.5
62
63 nss-tools >= 3.44.0
64
65 chkconfig >= 1.20
66
67 cyrus-sasl-gssapi >= 2.1
68
69 python3 >= 3.6~
70
71 qt5-qtbase >= 5.5~
72
73 qt5-qtbase-gui >= 5.5~
74
75 qrencode-libs >= 3.4.4
76
77 imlib2 >= 1.4.9
78
79 fuse-libs >= 2.9
80
81 pulseaudio-utils >= 15.0
```

**RPM 依存関係一覧 (RHEL 8.x および Rocky Linux 8.x の場合):**

```
1 java-11-openjdk >= 11
2
3 icoutils >= 0.32
4
5 firewalld >= 0.6.3
6
7 polycoreutils-python >= 2.8.9
8
9 polycoreutils-python-utils >= 2.8
10
11 python3-polycoreutils >= 2.8
12
13 dbus >= 1.12.8
14
```

```
15  dbus-common >= 1.12.8
16
17  dbus-daemon >= 1.12.8
18
19  dbus-tools >= 1.12.8
20
21  dbus-x11 >= 1.12.8
22
23  xorg-x11-server-utils >= 7.7
24
25  xorg-x11-xinit >= 1.3.4
26
27  libXpm >= 3.5.12
28
29  libXrandr >= 1.5.1
30
31  libXtst >= 1.2.3
32
33  pam >= 1.3.1
34
35  util-linux >= 2.32.1
36
37  util-linux-user >= 2.32.1
38
39  xorg-x11-utils >= 7.5
40
41  bash >= 4.3
42
43  findutils >= 4.6
44
45  gawk >= 4.2
46
47  sed >= 4.5
48
49  cups >= 1.6.0
50
51  foomatic-filters >= 4.0.9
52
53  cups-filters >= 1.20.0
54
55  ghostscript >= 9.25
56
57  libxml2 >= 2.9
58
59  libmspack >= 0.7
60
61  krb5-workstation >= 1.13
62
63  ibus >= 1.5
64
65  nss-tools >= 3.44.0
66
67  gperftools-libs >= 2.4
```

```
68
69  cyrus-sasl-gssapi >= 2.1
70
71  python3 >= 3.6~
72
73  qt5-qtbase >= 5.5~
74
75  qt5-qtbase-gui >= 5.5~
76
77  qrencode-libs >= 3.4.4
78
79  imlib2 >= 1.4.9
```

**RPM 依存関係一覧 (Amazon Linux 2 の場合):**

```
1  java-11-openjdk >= 11
2
3  ImageMagick >= 6.7.8.9
4
5  firewalld >= 0.3.9
6
7  polycoreutils-python >= 2.0.83
8
9  dbus >= 1.6.12
10
11  dbus-x11 >= 1.6.12
12
13  xorg-x11-server-utils >= 7.7
14
15  xorg-x11-xinit >= 1.3.2
16
17  xorg-x11-server-Xorg >= 1.20.4
18
19  libXpm >= 3.5.10
20
21  libXrandr >= 1.4.1
22
23  libXtst >= 1.2.2
24
25  pam >= 1.1.8
26
27  util-linux >= 2.23.2
28
29  bash >= 4.2
30
31  findutils >= 4.5
32
33  gawk >= 4.0
34
35  sed >= 4.2
36
37  cups >= 1.6.0
38
```

```
39  foomatic-filters >= 4.0.9
40
41  libxml2 >= 2.9
42
43  libmspack >= 0.5
44
45  ibus >= 1.5
46
47  cyrus-sasl-gssapi >= 2.1
48
49  gperftools-libs >= 2.4
50
51  nss-tools >= 3.44.0
52
53  qt5-qtbase >= 5.5~
54
55  qrencode-libs >= 3.4.1
56
57  imlib2 >= 1.4.5
```

**注:**

このバージョンの Linux VDA でサポートされている Linux ディストリビューションと Xorg のバージョンについては、「[システム要件](#)」を参照してください。

手順 **6b**: 既存のインストールをアップグレードする (オプション)

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

**Amazon Linux 2、RHEL、Rocky Linux** ディストリビューションの場合:

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

**注:**

- 既存のインストールをアップグレードすると、`/etc/xdl` にある構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。
- Linux VDA を RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 で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 以降に更新します。
- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイルを処理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。以下では、`rpm` と `dpkg` が構成ファイルの変更でどのように機能するかについて説明しています:
  - **rpm**: デフォルトでは、ローカルバージョンが保持され、パッケージから新しいバージョンが `.rpmnew` 拡張子で保存されます。

- **dpkg**: どのように続行するかを選択するよう対話形式で促します。ローカル構成ファイルを保持し、新しいパッケージバージョンを**.dpkg-new** または**.dpkg-dist** として保存しながら Linux VDA を自動アップグレードするには、次のコマンドを使用します:

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

- ソフトウェアをアップグレードした後、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。

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。詳しくは、「[システム要件](#)」を参照してください。

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

- [XenServer](#)
- [VMware ESX](#)
- [Nutanix AHV](#)

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

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

```
1 yum install gcc
2
3 yum install "kernel-devel-$(uname -r)"
4
5 systemctl set-default multi-user.target
```

5. [Red Hat Enterprise Linux のドキュメント](#) 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時は、すべての質問でデフォルト（「いいえ」）を選択してください。

## 重要:

GPU パススルーを有効にすると、XenCenter を利用して Linux 仮想マシンにアクセスできなくなります。SSH を使用して接続します。

## 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
+-----+-----+-----+-----+-----+-----+

```

次のコマンドで、カードに適切な構成を設定します:

```
etc/X11/ctx-nvidia.sh
```

高い解像度や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。この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の指示に従ってください。

## 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、**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ます。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して **/etc/locale.gen** ファイルを編集してから、**sudo locale-gen** コマンドを実行します。

パッケージのインストール後、**ctxsetup.sh** スクリプトを実行して、Linux VDA を構成する必要があります。このスクリプトは、変更を行う前に環境を確認し、すべての依存コンポーネントがインストールされていることが確認されます。必要に応じて、いつでもこのスクリプトを再実行して設定を変更できます。

このスクリプトは、手動で質問に回答しながら、または事前に構成した回答を使用して自動で実行できます。続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

質問に回答する構成

次のようにして、質問に回答する手動構成を実行します：

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

自動化された構成

インストールを自動化するために、環境変数を使用して、セットアップスクリプトで必要となるオプションを指定できます。必要な変数がすべて指定されていると、スクリプトによってユーザーに情報の入力を求めるメッセージが表示されることがなくなります。

サポートされる環境変数には次のようなものがあります：

- **CTX\_XDL\_NON\_DOMAIN\_JOINED=' y|n'**  
-マシンをドメインに参加させるかどうか。デフォルト値は ' n' です。ドメイン参加済みシナリオの場合は ' n' に設定します。
- **CTX\_XDL\_AD\_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** -Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。
- **CTX\_XDL\_DDC\_LIST=' <list-ddc-fqdns>'** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の、完全修飾ドメイン名 (FQDN) のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。
- **CTX\_XDL\_VDI\_MODE=' y|n'** -専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を ' y' に設定します。
- **CTX\_XDL\_HDX\_3D\_PRO=' y|n'** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= ' y' となります)。
- **CTX\_XDL\_START\_SERVICE = ' y|n'** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE = ' y|n'** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。

- **CTX\_XDL\_ADD\_FIREWALL\_RULES = 'y|n'** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/ <none>** '-セッションで使用する GNOME、GNOME クラシック、KDE、MATE、または **Xfce** デスクトップ環境を指定します。' <none> 'に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。  
コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。
- **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 をインストールするパス。デフォルトのパスは '/usr/bin' です。
- **CTX\_XDL\_VDA\_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX\_XDL\_SITE\_NAME =<dns-name>** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、' <none> 'に設定します。
- **CTX\_XDL\_LDAP\_LIST=' <list-ldap-servers>'** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。例: 「ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268」。Active Directory フォレストでより高速な LDAP クエリを有効にするには、ドメインコントローラーで [グローバルカタログ] を有効にし、関連する LDAP ポート番号で 3268 を指定します。この変数は、デフォルトでは ' <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使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。不要な場合は、' <none> 'に設定します。
- **CTX\_XDL\_SUPPORT\_DDC\_AS\_CNAME=' y|n'** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。

次のようにして、環境変数を設定し、構成スクリプトを実行します:

```
1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
```

```

9 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<
  none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

sudo コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべてのパラメーターを指定することができます：

```

1 sudo CTX_XDL_NON_DOMAIN_JOINED='n' \
2 CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \
3 CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \
4 CTX_XDL_VDI_MODE='y|n' \
5 CTX_XDL_HDX_3D_PRO='y|n' \
6 CTX_XDL_START_SERVICE='y|n' \
7 CTX_XDL_REGISTER_SERVICE='y|n' \
8 CTX_XDL_ADD_FIREWALL_RULES='y|n' \
9 CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \
  \
10 CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \
11 CTX_XDL_VDA_PORT='<port-number>' \
12 CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \
13 CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \
14 CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \
15 CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \
16 /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

#### 構成変更の削除

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しないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

構成変更を削除するには：

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

#### 重要：

このスクリプトにより、すべての構成データがデータベースから削除され、Linux VDA を操作できなくなります。

## 構成ログ

**ctxsetup.sh** および **ctxcleanup.sh** スクリプトでは、コンソールにエラーが表示され、構成ログファイル **/tmp/xdl.config.log** に追加情報が書き込まれます。

Linux VDA サービスを再起動し、変更を反映させます。

## 手順 9: XDPing の実行

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「**XDPing**」を参照してください。

## 手順 10: Linux VDA の実行

**ctxsetup.sh** スクリプトを使用して Linux VDA を構成したら、次のコマンドを使用して Linux VDA を制御します。

### Linux VDA の起動:

Linux VDA サービスを起動するには、次のコマンドを実行します:

```
1 sudo systemctl restart ctxhdx
2
3 sudo systemctl restart ctxvda
```

### Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

#### 注:

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

### Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl restart ctxhdx
4
5 sudo systemctl restart ctxvda
```

### Linux VDA の状態の確認:

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda
2
3 sudo systemctl status ctxhdx
```

### 手順 11: マシンカタログの作成

マシンカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明については、「[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。

次のように、Linux VDA マシンを含むマシンカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシンカタログの作成手順と異なる点があります:

- オペレーティングシステムには、次を選択します:
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。
- 同じマシンカタログで、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 の以前のバージョンは、「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** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **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暗黙的に選択されます。

#### ヒント:

削除されたマシンを Active Directory ドメインに再度追加する場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

### 手順 12: デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明については、「[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は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。

- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで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、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「[Citrix Virtual Apps and Desktops 7 2407](#)」を参照してください。

## SUSE への Linux VDA の手動インストール

August 20, 2024

**重要:**

新規インストールの場合は、[簡単インストール](#)を使用して簡易インストールを行うことをお勧めします。簡単インストールは時間と労力を節約するだけでなく、本記事に記載されている手動インストールよりもエラーを減らすことができます。

### 手順 1: 構成ファイル情報および Linux マシンの準備

#### 手順 1a: YaST ツールの起動

SUSE Linux Enterprise YaST ツールを使用して、オペレーティングシステムのすべての要素を構成します。

テキストベースの YaST ツールを起動する方法

```
1 su -
2
3 yast
```

UI ベースの YaST ツールを起動する方法:

```
1 su -
2
3 yast2 &
```

#### 手順 1b: ネットワークの構成

以降のセクションでは、Linux VDA で使用するさまざまなネットワーク設定およびサービスの構成方法に関する情報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 ツールで実行す

る必要があります。次の手順は、UI ベースの YaST ツールを使用することが前提となっています。テキストベースの YaST ツールも使用できますが、ナビゲーション方法が異なり、ここでは説明していません。

#### ホスト名とドメインネームシステム (DNS) の構成

1. UI ベースの YaST ツールを起動します。
2. [システム]、[ネットワーク設定] の順に選択します。
3. [ホスト名/DNS] タブを開きます。
4. [DHCP でホスト名を設定する] オプションでいいえを選択します。
5. [DNS 構成の変更] で [カスタムポリシーを使用する] オプションをオンにします。
6. 以下を編集してネットワーク設定に反映させます。
  - 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] 一覧-DNS ドメイン名を追加します。
7. `/etc/hosts` ファイルの次の行の最初の 2 つのエントリに FQDN とホスト名が含まれるように編集します:

```
127.0.0.1 <FQDN of the VDA> <hostname of the VDA> localhost
```

#### 注:

Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字以内である必要があります。

#### ヒント:

a~z、A~Z、0~9、およびハイフン (-) の文字のみ使用してください。アンダースコア (\_)、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了し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** 次のコマンドで、ホスト名が正しく設定されていることを確認します:

```
1 hostname
```

このコマンドにより、そのマシンの完全修飾ドメイン名 (FQDN) ではなく、そのホスト名のみが返されます。

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します:

```
1 hostname -f
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。



名前解決とサービス到達可能性の確認 次のコマンドで、完全修飾ドメイン名が解決できることと、ドメインコントローラーと Delivery Controller から ping に応答があることを確認します：

```
1 nslookup domain-controller-fqdn
2
3 ping domain-controller-fqdn
4
5 nslookup delivery-controller-fqdn
6
7 ping delivery-controller-fqdn
```

完全修飾ドメイン名を解決できない、またはこれらのマシンのいずれかから ping に応答がない場合は、手順を確認してから次に進んでください。

### 手順 1c: NTP サービスの構成

VDA、Delivery Controller、ドメインコントローラーの間で正確な時刻同期を維持することが重要です。仮想マシン (VM) として Linux VDA をホストすると、時刻が不正確になる問題が発生する可能性があります。したがって、リモート NTP サービスを使用して時刻を維持することをお勧めします。次のように、デフォルト NTP 設定にいくつかの変更が必要な場合があります。

**SUSE 15.5** 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]、[NTP 設定] の順に選択します。
3. [NTP デーモンを起動する] セクションで、[今すぐ開始し、システム起動時に開始するよう設定] を選択します。
4. [設定元] で [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。この 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 に次の行があれば、削除するかコメントを付けます。

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

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

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

### 手順 1d: Linux VDA に依存するパッケージのインストール

SUSE Linux Enterprise 用の Linux VDA ソフトウェアは、次のパッケージに依存しています：

- OpenJDK 11

- Open Motif Runtime Environment 2.3.1 以降
- Cups 1.6.0 以降
- ImageMagick 6.8 以降

リポジトリの追加 ImageMagick を除くほとんどの必要なパッケージは、公式リポジトリから入手できます。ImageMagick パッケージを入手するには、YaST または次のコマンドを使用して `sle-module-desktop-applications` リポジトリを有効にします:

```
SUSEConnect -p sle-module-desktop-applications/<version number>/x86_64
```

**Kerberos** クライアントのインストール 次のコマンドで、Linux VDA と Delivery Controller 間の相互認証用に Kerberos クライアントをインストールします。

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

Kerberos クライアントの構成は、使用する Active Directory 統合の方法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** のインストール 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:

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

使用するデータベースのインストールと指定

注:

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。`/etc/xdl/db.conf` で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合は、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから `/etc/xdl/db.conf` を編集すると、どちらを使用するかを指定できます。

このセクションでは、PostgreSQL と SQLite をインストールする方法と、どちらを使用するかを指定する方法について説明します。

**PostgreSQL** のインストール `Postgresql` をインストールするには、次のコマンドを実行します：

```
1 sudo zypper install postgresql-server
2
3 sudo zypper install postgresql-jdbc
```

マシンの起動時または即時で PostgreSQL 起動するには、それぞれ次のコマンドを実行します：

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl restart postgresql
```

**SQLite** をインストールする SUSE の場合、次のコマンドを実行して SQLite をインストールします：

```
1 sudo zypper install sqlite3
```

使用するデータベースを指定する SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから `/etc/xdl/db.conf` を編集すると、どちらを使用するかを指定できます。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールの場合、この手順は省きます。
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：

`/etc/xdl/db.conf` を使用して PostgreSQL のポート番号を構成することもできます。

## 手順 2：ハイパーバイザーの準備

サポートされるハイパーバイザー上で VM として Linux VDA を実行する場合、いくつかの変更が必要です。使用するハイパーバイザーのプラットフォームに合わせて、次の変更を行います。ベアメタルハードウェアで Linux マシンを実行する場合、変更は必要ありません。

## XenServer（旧称 Citrix Hypervisor）での時刻同期の修正

XenServer の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP と XenServer の問題が発生します。これは、NTP と Citrix Hypervisor の両方がシステムの時間を管理しようとするのが原因です。システムの時刻と他のサーバーの時刻との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時刻を 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、変更は必要ありません。

XenServer 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、XenServer の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます：

```
1 su -
2
3 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いるため、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います。こ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、以下の手順は必要ありません。

時刻同期機能が有効になっている場合は、ファイルに「**1**」と書き込んで無効にします:

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

この変更を永続化し、再起動後も保持するには、**/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ため、次のようにしてシステムを再起動します:

```
1 reboot
```

再起動後、設定が正しいことを確認します:

```
1 su -
2
3 cat /proc/sys/xen/independent_wallclock
```

このコマンドは 1 を返します。

## Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想マシンでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にします。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、[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:

この方法は VMware および XenServer (旧称 Citrix Hypervisor) の場合とは異なります。VMware および XenServer では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期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な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーで問題が発生します。これは、NTP とハイパーバイザーの両方がシステムの時間を同期しようとするのが原因です。システムの時刻と他のサーバーの時刻との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時刻を NTP と同期させます。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools をインストールした状態で準仮想化 Linux カーネルを実行している場合は、次の操作を行います。

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. [仮想マシンのプロパティ] ダイアログボックスで、[オプション] タブをクリックします。
4. [VMware Tools] を選択します。
5. [詳細] ボックスで、[ホストとゲスト時刻を同期] チェックボックスをオフにします。

## 手順 3: Linux VM を Windows ドメインに追加

以下は、Linux マシンを Active Directory (AD) ドメインに追加する方法です：

- [Samba Winbind](#)
- [Centrify DirectControl](#)
- [SSSD](#)
- [PBIS](#)

選択した方法の手順に従います。

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ことがあります。

## Samba Winbind

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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。

1. YaST を起動し、[ネットワークサービス]、[**Windows** ドメインメンバーシップ] の順に選択します。
2. 以下の変更を行います。
  - [ドメイン/ワークグループ] に 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 アドレスを設定します。ドメイン名は必ず大文字にします。
  - [Linux の認証に **SMB** の情報を使用する] チェックボックスをオンにします。
    - [Create Home Directory on Login] チェックボックスをオンにします。

- [SSH 向けのシングルサインオン] チェックボックスをオンにします。
  - [オフライン認証] チェックボックスがオフになっていることを確認します。Linux VDA は、このオプションに対応していません。
3. [OK] をクリックします。いくつかのパッケージのインストールを促すメッセージが表示された場合は、[インストール] をクリックします。
  4. ドメインコントローラーが見つかると、ドメインに参加するかどうかを確認するメッセージが表示されます。[はい] をクリックします。
  5. メッセージが表示されたら、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、マシンを再起動してください。マシンを再起動することをお勧めします：

```
1 su -
2 reboot
```

**ドメインメンバーシップの確認** Delivery Controller を使用するには、すべての VDA マシン (Windows と Linux VDA) で Active Directory にコンピューターオブジェクトが必要です。

次のように、**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：

```
1 sudo net ads testjoin
```

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

```
1 sudo net ads info
```

**Kerberos 構成の確認** システムの keytab ファイルが作成され、このファイルに有効なキーが含まれていることを確認します：

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します：

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します：

```
1 sudo klist
```

次のコマンドを使用して、マシンアカウントの詳細を調査します。

```
1 sudo net ads status
```

ユーザー認証の確認 次のように、**wbinfo** ツールを使用して、ドメインユーザーがドメインに対して認証できることを確認します：

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

ここで指定するドメインは AD ドメイン名で、Kerberos 領域名ではありません。bash シェルの場合、バックスラッシュ文字 (\) は、もう 1 つバックスラッシュ文字を指定してエスケープする必要があります。このコマンドにより、成功または失敗を示すメッセージが返されます。

Winbind PAM モジュールが正しく構成されていることを確認します。これを行うには、以前に使用したことが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。

```
1 ssh localhost -l domain\username
2 id -u
```

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、ユーザーの Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

```
1 klist
```

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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。

## Quest Authentication Service

ドメインコントローラーでの **Quest** の構成 次の操作は、Quest ソフトウェアをドメインコントローラーにインストールし、構成していることと、管理者特権が付与され、**Active Directory** にコンピューターオブジェクトを作成できることを前提としています。

**Linux VDA** マシンにドメインユーザーがログオンできるようにする Linux VDA マシンで HDX セッションを確立する必要がある各ドメインユーザーに対して、次の操作を行います。

1. [Active Directory ユーザーとコンピューター] 管理コンソールで、目的のユーザーアカウントの Active Directory ユーザーの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] チェックボックスをオンにします。
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:

この手順は、ドメインユーザーがコンソール、RDP、SSH、またはその他のリモート処理プロトコルを使用してログオンできるように設定する場合も同じです。

### Linux VDA での Quest の構成

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

```
1 sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-  
   interval 32400  
2  
3 sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-  
   auth false
```

このコマンドにより、更新間隔が9時間（32,400秒）に設定されます。すなわち、チケットのデフォルトの有効期間である10時間よりも1時間短くなります。チケットの有効期間がさらに短いシステムでは、より小さい値をこのパラメーターに設定します。

**PAM** および **NSS** の構成 HDX や、su、ssh、RDP などのその他のサービスを介したドメインユーザーのログオンを有効にするには、PAM と NSS を手動で構成します:

```
1 sudo /opt/quest/bin/vastool configure pam  
2  
3 sudo /opt/quest/bin/vastool configure nss
```

**Windows** ドメインへの参加 Quest `vastool` コマンドを使用して、Linux マシンを Active Directory ドメインに参加させます:

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

**user** は、マシンを Active Directory ドメインに追加す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（example.com など）です。

ドメインに追加後、Linux マシンを再起動します。



**ドメインメンバーシップの確認** Delivery Controller を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。Quest によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します：

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

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下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します。これを行うには、以前に使用したことが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。

```
1 ssh localhost -l domain\username  
2 id -u
```

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

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

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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。

## Centrify DirectControl

**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、次のように Centrify の **adjoin** コマンドを使用して、Linux マシンを Active Directory ドメインに追加します：

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

**user** は、マシンを Active Directory ドメインに追加する権限を持つ任意の Active Directory ドメインユーザーです。**domain-name** は、Linux マシンを追加す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を使用するには、すべてのVDAマシン (WindowsとLinux VDA) でActive Directoryにコンピューターオブジェクトが必要です。Centrifyにより追加されたLinuxマシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 sudo adinfo
```

**Joined to domain** 値が有効であることと、**CentrifyDC mode** で **connected** が返されることを確認します。CentrifyDC mode が starting のまま変化しない場合は、Centrifyクライアントにサーバーとの接続の問題、または認証の問題が発生しています。

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
1 adinfo --sysinfo all
2
3 adinfo -diag
```

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

```
1 adinfo --test
```

ドメイン参加の確認後、「[手順6: Linux VDAのインストール](#)」に進みます。

## SSSD

SUSEでSSSDを使用している場合は、このセクションの指示に従ってください。このセクションでは、Linux VDAマシンのWindowsドメインへの参加手順、およびKerberos認証の構成について説明します。

SUSEでSSSDをセットアップするには、次の手順を実行します:

1. ドメインに参加してホストのkeytabを作成
2. SSSD用のPAMの構成
3. SSSDのセットアップ
4. SSSDの有効化
5. ドメインメンバーシップの確認
6. Kerberos構成の確認
7. ユーザー認証の確認

ドメインに参加してホストの**keytab**を作成 SSSDでは、ドメイン参加とシステムのkeytabファイルの管理に関するActive Directoryのクライアント機能が提供されていません。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、以下の手順を実行してください。

1. Name Service Cache Daemon (NSCD) デーモンを停止して無効にします。

```
1 sudo systemctl stop nscd
2 sudo systemctl disable nscd
```

2. ホスト名と Chrony の時間同期を確認してください。

```
1 hostname
2 hostname -f
3 chronyc traking
```

3. 次のようにして、必要なパッケージをインストールまたは更新します:

```
1 sudo zypper install samba-client sssd-ad
```

4. `/etc/krb5.conf` ファイルをルートユーザーとして編集し、**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。**[libdefaults]**、**[realms]**、**[domain\_realm]** セクションに次のエントリを追加します:

注:

AD インフラストラクチャに基づいて Kerberos を構成します。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。

```
1 [libdefaults]
2
3     dns_canonicalize_hostname = false
4
5     rdns = false
6
7     default_realm = REALM
8
9     forwardable = true
10
11 [realms]
12
13     REALM = {
14
15
16         kdc = fqdn-of-domain-controller
17
18         default_domain = realm
19
20         admin_server = fqdn-of-domain-controller
21     }
22
23 [domain_realm]
24
25     .realm = REALM
```

**realm** は、Kerberos 領域名 (example.com など) です。**REALM** は、大文字の Kerberos 領域名 (EXAMPLE.COM など) です。

5. `/etc/samba/smb.conf` をルートユーザーとして編集し、**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。**[global]** セクションで次のとおりにエントリを追加します:
-

```
1 [global]
2   workgroup = domain
3
4   client signing = yes
5
6   client use spnego = yes
7
8   kerberos method = secrets and keytab
9
10  realm = REALM
11
12  security = ADS
```

**domain** は、EXAMPLE などの Active Directory ドメインの短い NetBIOS 名です。

6. **/etc/nsswitch.conf** ファイルで **passwd** および **group** エントリを変更して、ユーザーとグループの解決時に SSSD を参照します。

```
1 passwd: compat sss
2
3 group: compat sss
```

7. 構成済みの Kerberos クライアントを使用して、管理者としてターゲットドメインに対して認証します。

```
1 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、システムの keytab ファイルを生成します。

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

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

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

SSSD 経由のユーザー認証用に 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。

```
1 sudo pam-config --add --sss
2 sudo pam-config --add --mkhomedir
```

**SSSD** のセットアップ

1. **/etc/sss/sss.conf** をルートユーザーとして編集し、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。sss.conf の設定の例（必要に応じて追加の設定を行うことができます）：

```
1 [sss]
2   config_file_version = 2
3   services = nss,pam
```

```
4     domains = domain-dns-name
5
6     [domain/domain-dns-name]
7         id_provider = ad
8         auth_provider = ad
9         access_provider = ad
10        ad_domain = domain-dns-name
11        ad_server = fqdn-of-domain-controller
12        ldap_id_mapping = true
13        ldap_schema = ad
14
15    # Kerberos settings
16        krb5_ccachedir = /tmp
17        krb5_ccname_template = FILE:%d/krb5cc_%U
18
19    # Comment out if the users have the shell and home dir set on the
20    # AD side
21
22        fallback_homedir = /home/%d/%u
23        default_shell = /bin/bash
24
25    # Uncomment and adjust if the default principal SHORTNAME$@REALM
26    # is not available
27
28        ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
29
30        ad_gpo_access_control = permissive
```

**domain-dns-name** は、example.com などの DNS ドメイン名です。

注:

**ldap\_id\_mapping** は true に設定されるため、SSSD 自体が Windows SID を Unix UID にマッピングします。それ以外の場合、Active Directory は POSIX 拡張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のエラーを防ぐために、**ad\_gpo\_access\_control** は **permissive** に設定されます。[sssd.conf](#) および [sssd-ad](#) の man ページを参照してください。

2. ファイルの所有権およびアクセス権限を `sssd.conf` で設定します。

```
1 sudo chmod 0600 /etc/sssd/sssd.conf
```

**SSSD** の有効化 次のコマンドを実行して、SSSD デーモンを有効にし、システムの起動時に起動できるようにします。

```
1 sudo systemctl enable sssd
2 sudo systemctl start sssd
```

ドメインメンバーシップの確認

1. 次のように、**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:

```
1 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検証するには、次のコマンドを実行します:

```
1 sudo net ads info
```

**Kerberos** 構成の確認 システムの `keytab` ファイルが作成され、このファイルに有効なキーが含まれていることを確認します:

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。

Kerberos の **kinit** コマンドを実行し、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に対して認証します:

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します:

```
1 sudo klist
```

ユーザー認証の確認 SSSD は、デーモンで直接認証をテストするコマンドラインツールを提供しません。PAM 経由でのみ完了できます。

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が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。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

ユーザーの `klist` コマンドで返される Kerberos チケットが正しく、期限切れではないことを確認します。

ルートユーザーとして、前述の `id -u` コマンドで返された 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:

```
1 ls /tmp/krb5cc_uid
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「[手順 6: Linux VDA のインストール](#)」に進みます。

## PBIS

必要な **PBIS** パッケージをダウンロードする 例:

```
1 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:

```
1 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:

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

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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。

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

**user** は、マシンを 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 (example.com など) です。

注: Bash をデフォルトのシェルとして設定するには、**/opt/pbis/bin/configLoginShellTemplate/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 を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。PBIS によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 /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の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

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します。これを行うには、以前に使用したことが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。

```
1 ssh localhost -l domain\user
2
3 id -u
```

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、セッションを終了します。

```
1 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 をインストールする

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

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

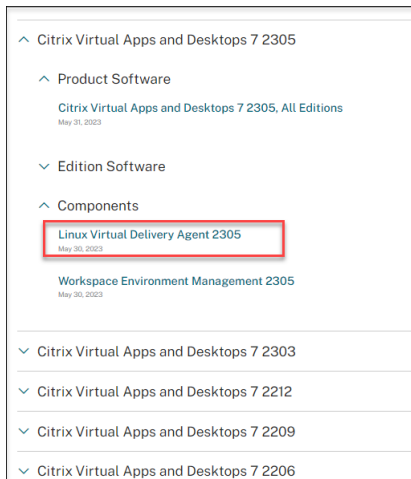
.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

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。

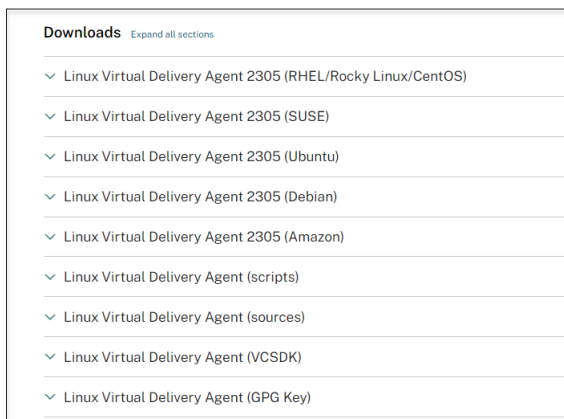
### 手順 5: Linux VDA パッケージのダウンロード

1. [Citrix Virtual Apps and Desktops のダウンロードページ](#) にアクセスします。
2. 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
3. **Components** を展開して Linux VDA を見つけます。例：



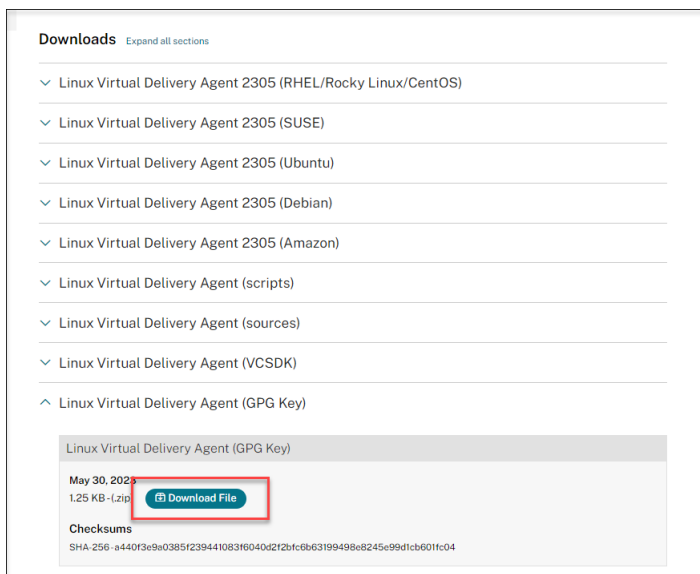


4. Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



5. 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。

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



公開キーを使用して Linux VDA パッケージの整合性を確認するには、次のコマンドを実行して公開キーを RPM データベースにインポートし、パッケージの整合性を確認します：

```
1 rpmkeys --import <path to the public key>
2 rpm --checksig --verbose <path to the Linux VDA package>
```

### 手順 6: Linux VDA のインストール

#### 手順 6a: 古いバージョンのアンインストール

最新の 2 バージョンおよび LTSR リリース以外の古いバージョンの Linux VDA がインストールされている場合は、それをアンインストールしてから新しいバージョンをインストールする必要があります。

1. 次のコマンドで、Linux VDA サービスを停止します：

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

注：

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

2. 次のコマンドで、パッケージをアンインストールします：

```
1 sudo rpm -e XenDesktopVDA
```

重要：

最新の 2 バージョンからのアップグレードがサポートされます。

注：

インストールされているコンポーネントは、**/opt/Citrix/VDA/** で確認できます。

コマンドを実行するには、フルパスが必要です。代わりに、システムパスに**/opt/Citrix/VDA/sbin** および**/opt/Citrix/VDA/bin** を追加することもできます。

#### 手順 6b: Linux VDA のインストール

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

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

RPM Package Manager を使用して、Linux VDA ソフトウェアをインストールします：

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

**手順 6c: Linux VDA のアップグレード (オプション)**

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

**注:**

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

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

**RPM 依存関係一覧 (SUSE 15 の場合):**

```
1 java-11-openjdk >= 11
2
3 ImageMagick >= 7.0
4
5 dbus-1 >= 1.12.2
6
7 dbus-1-x11 >= 1.12.2
8
9 xorg-x11 >= 7.6_1
10
11 libXpm4 >= 3.5.12
12
13 libXrandr2 >= 1.5.1
14
15 libXtst6 >= 1.2.3
16
17 pam >= 1.3.0
18
19 bash >= 4.4
20
21 findutils >= 4.6
22
23 gawk >= 4.2
24
25 sed >= 4.4
26
27 cups >= 2.2
28
29 cups-filters >= 1.25
30
31 libxml2-2 >= 2.9
32
33 libmspack0 >= 0.6
34
```

```
35 ibus >= 1.5
36
37 libtcmalloc4 >= 2.5
38
39 libcap-progs >= 2.26
40
41 mozilla-nss-tools >= 3.53.1
42
43 libpython3_6m1_0 >= 3.6~
44
45 libQt5Widgets5 >= 5.12
46
47 libqrencode4 >= 4.0.0
48
49 libImlib2-1 >= 1.4.10
```

**重要:**

アップグレードした後、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、次のガイドを参照してください:

- [XenServer](#)
- [VMware ESX](#)
- [Nutanix 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、GPU を VM に割り当てます。
3. 仮想マシンを起動します。
4. ゲスト VM ドライバーを VM にインストールします。

**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、**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ます。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して **/etc/locale.gen** ファイルを編集してから、**sudo**

**locale-gen** コマンドを実行します。

パッケージのインストール後、**ctxsetup.sh**スクリプトを実行して、Linux VDA を構成する必要があります。このスクリプトは、変更を行う前に環境を確認し、すべての依存コンポーネントがインストールされていることが確認されます。必要に応じて、いつでもこのスクリプトを再実行して設定を変更できます。

このスクリプトは、手動で質問に回答しながら、または事前に構成した回答を使用して自動で実行できます。続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

#### 質問に回答する構成

次のようにして、質問に回答する手動構成を実行します：

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

#### 自動化された構成

インストールを自動化するために、環境変数を使用して、セットアップスクリプトで必要となるオプションを指定できます。必要な変数がすべて指定されていると、スクリプトによってユーザーに情報の入力を求めるメッセージが表示されることがなくなります。

サポートされる環境変数には次のようなものがあります：

- **CTX\_XDL\_NON\_DOMAIN\_JOINED=' y|n'**  
- マシンをドメインに参加させるかどうか。デフォルト値は ' n' です。ドメイン参加済みシナリオの場合は ' n' に設定します。
- **CTX\_XDL\_AD\_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。
- **CTX\_XDL\_DDC\_LIST=' <list-ddc-fqdns>'** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の、完全修飾ドメイン名 (FQDN) のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。
- **CTX\_XDL\_VDI\_MODE=' y|n'** - 専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を ' y' に設定します。
- **CTX\_XDL\_HDX\_3D\_PRO=' y|n'** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= ' y' となります)。

- **CTX\_XDL\_START\_SERVICE = 'y|n'** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE = 'y|n'** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。
- **CTX\_XDL\_ADD\_FIREWALL\_RULES = 'y|n'** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/mate/ <none>** '- セッションで使用する GNOME、GNOME クラシック、または MATE デスクトップ環境を指定します。' <none> 'に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。  
 コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。
- **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 をインストールするパス。デフォルトのパスは '**/usr/bin**' です。
- **CTX\_XDL\_VDA\_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX\_XDL\_SITE\_NAME =<dns-name>** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、' <none> 'に設定します。
- **CTX\_XDL\_LDAP\_LIST=' <list-ldap-servers>'** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。例: 「ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268」。Active Directory フォレストでより高速な LDAP クエリを有効にするには、ドメインコントローラーで [グローバルカタログ] を有効にし、関連する LDAP ポート番号で 3268 を指定します。この変数は、デフォルトでは ' <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使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。不要な場合は、' <none> 'に設定します。
- **CTX\_XDL\_SUPPORT\_DDC\_AS\_CNAME=' y|n'** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。

次のようにして、環境変数を設定し、構成スクリプトを実行します:

```
1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
```

```

4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

`sudo` コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべてのパラメーターを指定することができます：

```

1 sudo CTX_XDL_NON_DOMAIN_JOINED='n' \
2 CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \
3 CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \
4 CTX_XDL_VDI_MODE='y|n' \
5 CTX_XDL_HDX_3D_PRO='y|n' \
6 CTX_XDL_START_SERVICE='y|n' \
7 CTX_XDL_REGISTER_SERVICE='y|n' \
8 CTX_XDL_ADD_FIREWALL_RULES='y|n' \
9 CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>' \
10 CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \
11 CTX_XDL_VDA_PORT='<port-number>' \
12 CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \
13 CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \
14 CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \
15 CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \
16 /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

#### 構成変更の削除

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しないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

```
1 sudo /usr/local/sbin/ctxcleanup.sh --help
```

構成変更を削除するには：

```
1 sudo /usr/local/sbin/ctxcleanup.sh
```

**重要:**

このスクリプトにより、すべての構成データがデータベースから削除され、Linux VDA を操作できなくなります。

構成ログ

**ctxsetup.sh** および **ctxcleanup.sh** スクリプトでは、コンソールにエラーが表示され、次の構成ログファイルに追加情報が書き込まれます:

`/tmp/xdl.configure.log`

Linux VDA サービスを再起動し、変更を反映させます。

**手順 9: XDPing の実行**

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「**XDPing**」を参照してください。

**手順 10: Linux VDA の実行**

**ctxsetup.sh** スクリプトを使用して Linux VDA を構成したら、次のコマンドを使用して Linux VDA を制御します。

**Linux VDA の起動:**

Linux VDA サービスを起動するには、次のコマンドを実行します:

```
1 sudo systemctl start ctxhdx
2
3 sudo systemctl start ctxvda
```

**Linux VDA の停止:**

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

**注:**

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。



**Linux VDA の再起動:**

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl restart ctxhdx
4
5 sudo systemctl restart ctxvda
```

**Linux VDA の状態の確認:**

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda
2
3 sudo systemctl status ctxhdx
```

**手順 11: マシンカタログの作成**

マシンカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明について詳しくは、「[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。

次のように、Linux VDA マシンを含むマシンカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシンカタログの作成手順と異なる点があります:

- オペレーティングシステムには、次を選択します:
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。
- 同じマシンカタログで、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 の以前のバージョンは、「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** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **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暗黙的に選択されません。

**ヒント:**

マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

## 手順 12: デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明については、「[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は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで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、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよび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 2407](#)」を参照してください。

## Ubuntu への Linux VDA の手動インストール

August 20, 2024

### 重要:

新規インストールの場合は、[簡単インストール](#)を使用して簡易インストールを行うことをお勧めします。簡単インストールは時間と労力を節約するだけでなく、本記事に記載されている手動インストールよりもエラーを減らすことができます。

## 手順 1: 構成ファイル情報および Linux マシンの準備

### 手順 1a: ネットワーク構成の確認

ネットワークが正しく接続および構成されていることを確認してください。たとえば、DNS サーバーは Linux VDA で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` 構成ファイルに次の変更を加えます:

```
preserve_hostname: true
```

### 手順 **1b**: ホスト名の設定

マシンのホスト名が確実に正しく報告されるようにするには、**/etc/hostname** ファイルを変更してマシンのホスト名のみを記述します。

`hostname`

### 手順 **1c**: ホスト名へのループバックアドレスの割り当て

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ます。このためには、**/etc/hosts** ファイルの次の行の最初の 2 つのエントリに FQDN とホスト名が含まれるように編集します:

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、`hostname-fqdn` または `hostname` に対するそ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 文字以内である必要があります。

ヒント:

a~z、A~Z、0~9、およびハイフン (-) の文字のみ使用してください。アンダースコア (\_)、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了し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。

### 手順 **1d**: ホスト名の確認

次のコマンドで、ホスト名が正しく設定されていることを確認します:

```
1 hostname
```

このコマンドによって、そのマシンの完全修飾ドメイン名 (FQDN) ではなく、そのホスト名のみが返されます。

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します:

```
1 hostname -f
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

手順 **1e**: マルチキャスト **DNS** の無効化

デフォルトの設定でマルチキャスト DNS (**mDNS**) が有効であるため、名前解決の結果に不整合が発生する場合があります。

**mDNS** を無効にするには、**/etc/nsswitch.conf** を編集して、以下を含む行を変更します:

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

変更後:

```
hosts: files dns
```

手順 **1f**: 名前解決とサービス到達可能性の確認

次のコマンドで、完全修飾ドメイン名が解決できることと、ドメインコントローラーと Delivery Controller から ping に応答があることを確認します:

```
1 nslookup domain-controller-fqdn
2
3 ping domain-controller-fqdn
4
5 nslookup delivery-controller-fqdn
6
7 ping delivery-controller-fqdn
```

完全修飾ドメイン名を解決できない、またはこれらのマシンのいずれかから ping に応答がない場合は、手順を確認してから次に進んでください。

手順 **1g**: 時刻同期の構成 (**chrony**)

VDA、Delivery Controller、ドメインコントローラーの間で正確な時刻同期を維持することは重要です。仮想マシン (VM) として Linux VDA をホストすると、時刻が不正確になる問題が発生する可能性があります。したがって、リモートのタイムサービスを使用して時刻を維持することをお勧めします。

**chrony** のインストール:

```
1 apt-get install chrony
```

ルートユーザーとして **/etc/chrony/chrony.conf** を編集し、次のように各リモートタイムサーバーに対応する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
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応する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、パブリックサーバーの **\*.pool.ntp.org** エントリなど、一覧にあるその他のサーバーまたはプールエントリを削除します。

変更を保存してから、次のコマンドで Chrony デーモンを再起動します：

```
1 sudo systemctl restart chrony
```

#### 手順 1h: OpenJDK 11 のインストール

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 で 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：

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

#### 手順 1i: 使用するデータベースのインストールと指定

注：

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合は、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

このセクションでは、PostgreSQL と SQLite をインストールする方法と、どちらを使用するかを指定する方法について説明します。

**PostgreSQL** のインストール 次のコマンドを実行して、PostgreSQL をインストールします：

```
1 sudo apt-get install -y postgresql
2
3 sudo apt-get install -y libpostgresql-jdbc-java
```

マシンの起動時または即時で PostgreSQL を起動するには、それぞれ次のコマンドを実行します：

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl restart postgresql
```

**SQLite** をインストールする Ubuntu の場合、次のコマンドを実行して SQLite をインストールします:

```
1 sudo apt-get install -y sqlite3
```

使用するデータベースを指定する SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省きます。
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:

**/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** のインストール

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

手順 **1k**: 他のパッケージのインストール

Ubuntu 22.04 の場合:

```
1 sudo apt-get install -y libsasl2-2
2 sudo apt-get install -y libsasl2-modules-gssapi-mit
3 sudo apt-get install -y libldap-2.5-0
4 sudo apt-get install -y krb5-user
5 sudo apt-get install -y libgtk2.0-0
```

Ubuntu 20.04 の場合:

```
1 sudo apt-get install -y libsasl2-2
2 sudo apt-get install -y libsasl2-modules-gssapi-mit
3 sudo apt-get install -y libldap-2.4-2
4 sudo apt-get install -y krb5-user
5 sudo apt-get install -y libgtk2.0-0
```

手順 **2**: ハイパーバイザーの準備

サポートされるハイパーバイザー上で VM として Linux VDA を実行する場合、いくつかの変更が必要です。使用するハイパーバイザーのプラットフォームに合わせて、次の変更を行います。ベアメタルハードウェアで Linux マシンを実行する場合、変更は必要ありません。

## XenServer (旧称 Citrix Hypervisor) での時刻同期の修正

XenServer の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP と XenServer の問題が発生します。これは、NTP と Citrix Hypervisor の両方がシステムの時間を管理しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは、変更は必要ありません。

XenServer 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、XenServer の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます：

```
1 su -
2
3 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いるため、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います。こ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、以下の手順は必要ありません。

時刻同期機能が有効になっている場合は、ファイルに「1」と書き込んで無効にします：

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

この変更を永続化し、再起動後も保持するには、`/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ため、次のようにしてシステムを再起動します：

```
1 su -
2
3 cat /proc/sys/xen/independent_wallclock
```

このコマンドは 1 を返します。

## Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想マシンでは、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間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にします。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、[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:

この方法は VMware および XenServer (旧称 Citrix Hypervisor) の場合とは異なります。VMware および XenServer では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期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な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーで問題が発生します。これは、NTP とハイパーバイザーの両方がシステムの時間を同期しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools をインストールした状態で準仮想化 Linux カーネルを実行している場合は、次の操作を行います。

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. [仮想マシンのプロパティ] ダイアログボックスで、[オプション] タブをクリックします。
4. [VMware Tools] を選択します。
5. [詳細] ボックスで、[ホストとゲスト時刻を同期] チェックボックスをオフにします。

### 手順 3: Linux VM を Windows ドメインに追加

以下は、Linux マシンを Active Directory (AD) ドメインに追加する方法です:

- [Samba Winbind](#)
- [Quest Authentication Service](#)
- [Centrify DirectControl](#)
- [SSSD](#)
- [PBIS](#)

選択した方法の手順に従います。

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ことがあります。

### Samba Winbind

必要なパッケージのインストールまたは更新

```
1 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** デーモンを開始できるようにする 次のコマンドで、マシン起動時に Winbind デーモンが開始するように構成する必要があります。

```
1 sudo systemctl enable winbind
```

注:

winbindスクリプトが/etc/init.dにあることを確認します。

**Kerberos** を構成します ルートユーザーとして/etc/krb5.conf を開き、以下を設定します。

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
```

ここで **domain-dns-name** パラメーターは、DNS ドメイン名 (**example.com** など) です。 **REALM** は、大文字の Kerberos 領域名 (**EXAMPLE.COM** など) です。

**Winbind** 認証の構成 RHEL の **authconfig** や、SUSE の yast2 のようなツールが Ubuntu にないため、手動で Winbind を構成します。

**vim /etc/samba/smb.conf** コマンドを実行して/etc/samba/smb.conf を開いてから、以下を設定します:

```
[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す。**REALM** は大文字の Kerberos 領域名です。

**nsswitch** の構成 `/etc/nsswitch.conf` を開き、**winbind** を次の行に追加します：

```
passwd: compat winbind
group: compat winbind
```

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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：

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

ここで、**REALM** は大文字の Kerberos 領域名で、**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** の再起動

```
1 sudo systemctl restart winbind
```

**Winbind** 用の **PAM** の構成 次のコマンドを実行して、**[Winbind NT/Active Directory authentication]** オプションと **[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：

```
1 sudo pam-auth-update
```

ヒント：

マシンがドメインに参加済みの場合にのみ、**winbind** デーモンは実行を続けます。

ドメインメンバーシップの確認 Delivery Controller を使用するには、Windows または Linux に関係なく、すべての VDA マシンで Active Directory にコンピューターオブジェクトが必要です。

次のように、**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：

```
1 sudo net ads testjoin
```

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

```
1 sudo net ads info
```

**Kerberos** 構成の確認 Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドによって、システムの **keytab** ファイルが作成済みで keytab ファイルに有効なキーが含まれていることを確認します：

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します：

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します：

```
1 sudo klist
```

次のコマンドを使用して、マシンアカウントの詳細を調査します：

```
1 sudo net ads status
```

ユーザー認証の確認 次のように、**wbinfo** ツールを使用して、ドメインユーザーがドメインに対して認証できることを確認します：

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

ここで指定するドメインは AD ドメイン名で、Kerberos 領域名ではありません。bash シェルの場合、バックスラッシュ文字 (\) は、もう 1 つバックスラッシュ文字を指定してエスケープする必要があります。このコマンドにより、成功または失敗を示すメッセージが返されます。

Winbind PAM モジュールが正しく構成されていることを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\username
2
3 id -u
```

注：

SSH コマンドを正しく実行するには、SSH が有効で適切に機能していることを確認します。

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、ユーザーの Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

```
1 klist
```

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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を表示できない場合、マシンを再起動して再試行します。

## Quest Authentication Service

ドメインコントローラーでの **Quest** の構成 次の操作は、Quest ソフトウェアを Active Directory ドメインコントローラーにインストールし、構成していることと、管理者特権が付与され、**Active Directory** にコンピューターオブジェクトを作成できることを前提としています。

**Linux VDA** マシンにドメインユーザーがログオンできるようにする Linux VDA マシンで HDX セッションを確立する必要がある各ドメインユーザーに対して、次の操作を行います。

1. [Active Directory ユーザーとコンピューター] 管理コンソールで、目的のユーザーアカウントの Active Directory ユーザーの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] チェックボックスをオンにします。
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：

この手順は、ドメインユーザーがコンソール、RDP、SSH、またはその他のリモート処理プロトコルを使用してログオンできるように設定する場合も同じです。

## Linux VDA での Quest の構成

**SELinux** ポリシー適用の回避策 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 ドメインソケットの IPC のメカニズムに干渉し、ドメインユーザーのログオンを妨げます。

この問題を回避するための便利な方法は、SELinux の無効化です。ルートユーザーとして、`/etc/selinux/config` を編集し、**SELinux** 設定を次のとおりに変更します：

`SELINUX=disabled`

この変更にはマシンの再起動が必要です：

```
1 reboot
```

重要：

この設定は注意して使用してください。SELinux ポリシーの適用を無効にした後に再度有効にすると、ルートユーザーやその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てしまう可能性があります。

**VAS** デモンの構成 次のように Kerberos チケットの自動更新を有効にして、切断する必要があります。認証（オフラインログオン）は無効にする必要があります：

```
1 sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-  
   interval 32400  
2  
3 sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-  
   auth false
```

このコマンドにより、更新間隔が 9 時間（32,400 秒）に設定されます。すなわち、チケットのデフォルトの有効期間である 10 時間よりも 1 時間短くなります。チケットの有効期間がさらに短いシステムでは、より小さい値をこのパラメーターに設定します。

**PAM** および **NSS** の構成 HDX や、su、ssh、RDP などのそ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：

```
1 sudo /opt/quest/bin/vastool configure pam  
2  
3 sudo /opt/quest/bin/vastool configure nss
```

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

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

`user` は、コンピューターを Active Directory ドメインに追加す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（example.com など）です。

ドメインに追加後、Linux マシンを再起動します。

**ドメインメンバーシップの確認** Delivery Controller を使用するには、Windows または Linux に関係なく、すべての VDA マシンで Active Directory にコンピューターオブジェクトが必要です。Quest によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します：

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

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下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** PAM を使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\username  
2  
3 id -u
```

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

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

次のコマンドで、セッションを終了します。

```
1 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。

## Centrify DirectControl

**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、次のように Centrify の **adjoin** コマンドを使用して、Linux マシンを Active Directory ドメインに追加します：

```
1 su -  
2 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** ドメインに追加す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 マシンを追加す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 を使用するには、Windows または Linux に関係なく、すべての VDA マシンで **Active Directory** にコンピューターオブジェクトが必要です。Centrify により追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します：

```
1 su -
2
3 adinfo
```

**Joined to domain** 値が有効であることと、**CentrifyDC mode** で **connected** が返されることを確認します。CentrifyDC mode が starting のまま変化しない場合は、Centrify クライアントにサーバーとの接続の問題、または認証の問題が発生しています。

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
1 adinfo --sysinfo all
2
3 adinfo --diag
```

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

```
1 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** を構成します Kerberos をインストールするには、次のコマンドを実行します：

```
1 sudo apt-get install krb5-user
```

Kerberos を構成するには、**/etc/krb5.conf** をルートとして開き、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
```

ここで `domain-dns-name` パラメーターは、DNS ドメイン名 (example.com など) です。 `REALM` は、大文字の Kerberos 領域名 (EXAMPLE.COM など) です。

ドメインに参加する SSSD を構成して、Active Directory を ID プロバイダーおよび認証の Kerberos として使用します。ただし、SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する AD のクライアント機能が提供されていません。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** および **Samba** に関する情報のみを提供します。

- **adcli** を使用してドメインに参加する場合は、次の手順を実行します:

1. **adcli** をインストールします。

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

2. **adcli** でドメインに参加させます。

次を使用して古いシステム keytab ファイルを削除し、ドメインに参加させます。

```
1 su -
2
3 rm -rf /etc/krb5.keytab
4
5 adcli join domain-dns-name -U user -H hostname-fqdn
```

**user** は、ドメインにマシンを追加する権限があるドメインユーザーです。**hostname-fqdn** は、完全修飾ドメイン名形式のマシンのホスト名です。

**-H** オプションは、**adcli** が、Linux VDA で必要な `host/hostname-fqdn@REALM` という形式で SPN を生成するのに必要です。

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 マシンの場合は、`adcli testjoin` コマンドを実行して、マシンがドメインに参加しているかどうかをテストします。

- **Samba** を使用してドメインに参加する場合は、次の手順を実行します:

1. パッケージをインストールします。

```
1 sudo apt-get install samba krb5-user
```



## 2. Samba を構成します。

`/etc/samba/smb.conf` を開き、次を設定します。

```
[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す。**REALM** は大文字の Kerberos 領域名です。

## 3. Samba でドメインに参加させます。

ドメインコントローラーがアクセス可能で、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。

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

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

**SSSD** のセットアップ 必要なパッケージのインストールまたは更新:

必要な SSSD および構成パッケージがインストールされていない場合、インストールします。

```
1 sudo apt-get install sssd
```

パッケージが既にインストールされている場合、更新することをお勧めします。

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

注:

Ubuntu のインストールプロセスは、デフォルトで **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** の構成 SSSD デーモンを起動する前に、SSSD 構成の変更が必要です。SSSD の一部のバージョンでは、`/etc/sssdsmb.conf` 構成ファイルはデフォルトではインストールされないため、手動で作成する必要があります。root として `/etc/sssdsmb.conf` を作成するか開いて、次を設定します:

```
[sssdsmb]
```

```
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** に設定されるため、SSSD 自体が Windows SID を Unix UID にマッピングします。設定しない場合、Active Directory が POSIX 拡張を提供できるようにする必要があります。PAM サービス (ctxhdx) は、ad\_gpo\_map\_remote\_interactive に追加されます。

ここで **domain-dns-name** パラメーターは、DNS ドメイン名 (example.com など) です。**REALM** は、大文字の Kerberos 領域名 (EXAMPLE.COM など) です。NetBIOS ドメイン名を構成するための要件はありません。

構成設定について詳しくは、sssd.conf および sssd-ad に関する man ページを参照してください。

SSSD デーモンでは、構成ファイルに所有者読み取り権限のみが設定されている必要があります。

```
1 sudo chmod 0600 /etc/sss/sss.conf
```

**SSSD** デーモンの起動 次のコマンドを実行して、SSSD デーモンを起動し、マシンの起動時にもデーモンを起動できるようにします。

```
1 sudo systemctl start sssd
2
3 sudo systemctl enable sssd
```

**PAM** 構成 次のコマンドを実行して、[**SSS authentication**] オプションと [**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：

```
1 sudo pam-auth-update
```

ドメインメンバーシップの確認 Delivery Controller を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。

- **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 Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドによって、システムの keytab ファイルが作成済みで keytab ファイルに有効なキーが含まれていることを確認します：

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の `kinit` コマンドを実行し、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に対して認証します：

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT がキャッシュされたことを確認します：

```
1 sudo klist
```

ユーザー認証の確認 SSSD は、デーモンで直接認証をテストするコマンドラインツールを提供しません。PAM 経由でのみ完了できます。

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が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。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

ユーザーの **klist** コマンドで返される Kerberos チケットが正しく、期限切れではないことを確認します。

ルートユーザーとして、前述の **id -u** コマンドで返された 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：

```
1 ls /tmp/krb5cc_uid
```

KDE または Gnome Display Manager に直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「[手順 6: Linux VDA のインストール](#)」に進みます。

### PBIS

必要な **PBIS** パッケージをダウンロードする

```
1 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

```
1 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

```
1 sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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：

```
1 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 (example.com など) です。

注： Bash をデフォルトの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 を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。PBIS によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 /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の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

ユーザー認証の確認 PAM を使用した PBIS のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 sudo ssh localhost -l domain\user
2
3 id -u
```

次のコマンドで、**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:

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、セッションを終了します。

```
1 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 をインストールする

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

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

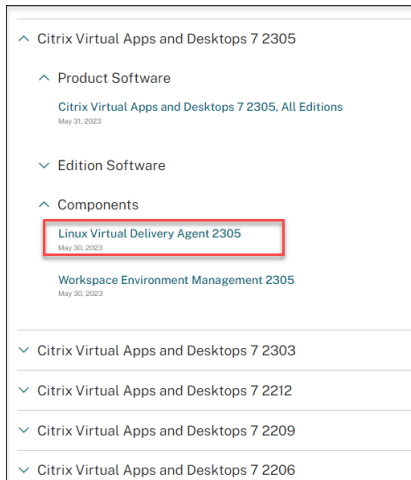
.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

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。

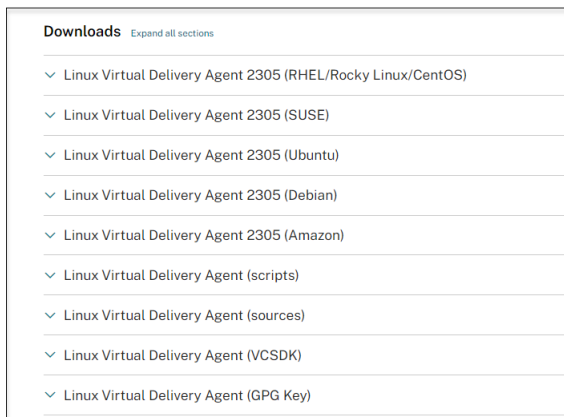
#### 手順 5: Linux VDA パッケージのダウンロード

1. [Citrix Virtual Apps and Desktops のダウンロードページ](#) にアクセスします。

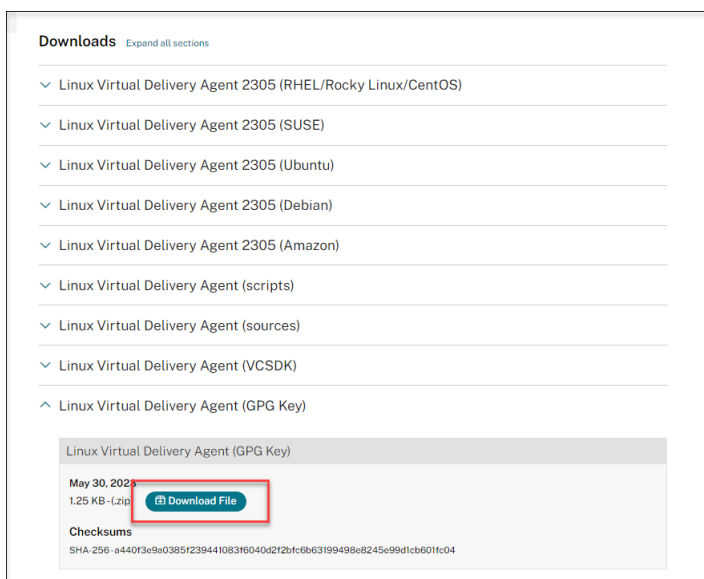
- 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
- Components** を展開して Linux VDA を見つけます。例:



- Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



- 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
- Linux VDA パッケージの整合性を検証するために使用できる GPG 公開キーをダウンロードします。例:



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

```
1 sudo apt-get install dpkg-sig
2 gpg --import <path to the public key>
3 dpkg-sig --verify <path to the Linux VDA package>
```

### 手順 6: Linux VDA のインストール

#### 手順 6a: Linux VDA のインストール

次のように、Debian Package Manager を使用して Linux VDA ソフトウェアをインストールします：

```
1 sudo dpkg -i <PATH>/<Linux VDA DEB>
2 apt-get install -f
```

注：

GCP 上の Ubuntu 20.04 の場合、RDNS を無効にします。これを行うには、`/etc/krb5.conf` の `[libdefaults]` に `rdns = false` 行を追加します。

#### Ubuntu 22.04 の Debian 依存関係一覧：

```
1 openjdk-11-jdk >= 11
2
3 imagemagick >= 8:6.9.11
4
5 libgtkmm-3.0-1v5 >= 3.24.5
6
7 ufw >= 0.36
8
```

```
9  ubuntu-desktop >= 1.481
10
11  libxrandr2 >= 2:1.5.2
12
13  libxtst6 >= 2:1.2.3
14
15  libxm4 >= 2.3.8
16
17  util-linux >= 2.37
18
19  gtk3-nocsd >= 3
20
21  bash >= 5.1
22
23  findutils >= 4.8.0
24
25  sed >= 4.8
26
27  cups >= 2.4
28
29  libmspack0 >= 0.10
30
31  ibus >= 1.5
32
33  libgoogle-perftools4 >= 2.9~
34
35  libpython3.10 >= 3.10~
36
37  libsasl2-modules-gssapi-mit >= 2.1.~
38
39  libnss3-tools >= 2:3.68
40
41  libqt5widgets5 >= 5.15~
42
43  libqrencode4 >= 4.1.1
44
45  libimlib2 >= 1.7.4
```

**Ubuntu 20.04 の Debian 依存関係一覧:**

```
1  openjdk-11-jdk >= 11
2
3  imagemagick >= 8:6.9.10
4
5  libgtkmm-3.0-1v5 >= 3.24.2
6
7  ufw >= 0.36
8
9  ubuntu-desktop >= 1.450
10
11  libxrandr2 >= 2:1.5.2
12
13  libxtst6 >= 2:1.2.3
```



```
14
15 libxm4 >= 2.3.8
16
17 util-linux >= 2.34
18
19 gtk3-nocsd >= 3
20
21 bash >= 5.0
22
23 findutils >= 4.7.0
24
25 sed >= 4.7
26
27 cups >= 2.3
28
29 libmspack0 >= 0.10
30
31 ibus >= 1.5
32
33 libgoogle-perftools4 >= 2.7~
34
35 libpython3.8 >= 3.8~
36
37 libsasl2-modules-gssapi-mit >= 2.1.~
38
39 libnss3-tools >= 2:3.49
40
41 libqt5widgets5 >= 5.7~
42
43 libqrencode4 >= 4.0.0
44
45 libimlib2 >= 1.6.1
```

注:

このバージョンの Linux VDA でサポートされている Linux ディストリビューションと Xorg のバージョンについては、「[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)

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

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2
3 sudo apt-get install -f
```

注:

- 既存のインストールをアップグレードすると、/etc/xdm にある構成ファイルが上書きされます。アップ

グレードを実行する前に、必ずファイルをバックアップしてください。

- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイル进行处理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。以下では、rpm と dpkg が構成ファイルの変更でどのように機能するかについて説明しています：

- **rpm**: by default keeps the local version and saves the new version from the package with a `.rpmnew` extension.
- **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as `.dpkg-new` or `.dpkg-dist`, use the following command:

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

### 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、次のガイドを参照してください：

- [XenServer](#)
- [VMware ESX](#)
- [Nutanix 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、GPU を VM に割り当てます。
3. 仮想マシンを起動します。
4. ゲスト VM ドライバーを VM にインストールします。

### 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、**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ます。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して `/etc/locale.gen` ファイルを編集してから、**sudo locale-gen** コマンドを実行します。

パッケージのインストール後、`ctxsetup.sh` スクリプトを実行して、Linux VDA を構成する必要があります。このスクリプトは、変更を行う前に環境を確認し、すべての依存コンポーネントがインストールされていることが確認されます。必要に応じて、いつでもこのスクリプトを再実行して設定を変更できます。

このスクリプトは、手動で質問に回答しながら、または事前に構成した回答を使用して自動で実行できます。続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

#### 質問に回答する構成

次のようにして、質問に回答する手動構成を実行します：

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

#### 自動化された構成

インストールを自動化するために、環境変数を使用して、セットアップスクリプトで必要となるオプションを指定できます。必要な変数がすべて指定されていると、スクリプトによってユーザーに情報の入力を求めるメッセージが表示されることがなくなり、インストール処理をスクリプト化できます。

サポートされる環境変数には次のようなものがあります：

- **CTX\_XDL\_NON\_DOMAIN\_JOINED=' y|n'**  
-マシンをドメインに参加させるかどうか。デフォルト値は ' n' です。ドメイン参加済みシナリオの場合は ' n' に設定します。
- **CTX\_XDL\_AD\_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** -Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。
- **CTX\_XDL\_DDC\_LIST=' <list-ddc-fqdns>'** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の、完全修飾ドメイン名 (FQDN) のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。
- **CTX\_XDL\_VDI\_MODE=' y|n'** -専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を 'y' に設定します。
- **CTX\_XDL\_HDX\_3D\_PRO=' y|n'** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= 'y' となります)。

- **CTX\_XDL\_START\_SERVICE = 'y|n'** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE = 'y|n'** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。
- **CTX\_XDL\_ADD\_FIREWALL\_RULES = 'y|n'** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/ <none>** '-セッションで使用する GNOME、GNOME クラシック、KDE、MATE、または **Xfce** デスクトップ環境を指定します。'**<none>**' に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。  
 コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。
- **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 をインストールするパス。デフォルトのパスは '**/usr/bin**' です。
- **CTX\_XDL\_VDA\_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX\_XDL\_SITE\_NAME =<dns-name>** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、'**<none>**' に設定します。
- **CTX\_XDL\_LDAP\_LIST=' <list-ldap-servers>'** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。例: 「ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268」。Active Directory フォレストでより高速な LDAP クエリを有効にするには、ドメインコントローラーで [グローバルカタログ] を有効にし、関連する LDAP ポート番号で 3268 を指定します。この変数は、デフォルトでは '**<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使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。不要な場合は、'**<none>**' に設定します。
- **CTX\_XDL\_SUPPORT\_DDC\_AS\_CNAME=' y|n'** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。

次のようにして、環境変数を設定し、構成スクリプトを実行します:

```
1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
```

```

4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<
  none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

sudo コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべてのパラメーターを指定することができます：

```

1 sudo CTX_XDL_NON_DOMAIN_JOINED='n' \
2 CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \
3 CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \
4 CTX_XDL_VDI_MODE='y|n' \
5 CTX_XDL_HDX_3D_PRO='y|n' \
6 CTX_XDL_START_SERVICE='y|n' \
7 CTX_XDL_REGISTER_SERVICE='y|n' \
8 CTX_XDL_ADD_FIREWALL_RULES='y|n' \
9 CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \
10 CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \
11 CTX_XDL_VDA_PORT='<port-number>' \
12 CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \
13 CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \
14 CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \
15 CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \
16 /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

#### 構成変更の削除

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しないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

構成変更を削除するには：

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

**重要:**

このスクリプトにより、すべての構成データがデータベースから削除され、Linux VDA を操作できなくなります。

構成ログ

**ctxsetup.sh** および **ctxcleanup.sh** スクリプトでは、コンソールにエラーが表示され、構成ログファイル **/tmp/xdl.configure.log** に追加情報が書き込まれます。

Linux VDA サービスを再起動し、変更を反映させます。

**Linux VDA** ソフトウェアのアンインストール

Linux VDA がインストールされているかどうかを確認したり、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、次のコマンドを実行します。

```
1 dpkg -l xendesktopvda
```

詳細を表示するには、次のコマンドを実行します。

```
1 apt-cache show xendesktopvda
```

Linux VDA ソフトウェアをアンインストールには、次のコマンドを実行します:

```
1 dpkg -r xendesktopvda
```

**注:**

Linux VDA ソフトウェアをアンインストールすると、関連付けられた 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 のインストールより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するパッケージは削除されません。

**ヒント:**

このセクションでは、PostgreSQL など、依存するパッケージの削除方法については説明していません。

**手順 9: XDPing の実行**

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「**XDPing**」を参照してください。

## 手順 10: Linux VDA の実行

**ctxsetup.sh** スクリプトを使用して Linux VDA を構成したら、次のコマンドを使用して Linux VDA を制御します。

### Linux VDA の起動:

Linux VDA サービスを起動するには、次のコマンドを実行します:

```
1 sudo systemctl start ctxhdx
2
3 sudo systemctl start ctxvda
```

### Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

#### 注:

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

### Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl restart ctxhdx
4
5 sudo systemctl restart ctxvda
```

### Linux VDA の状態の確認:

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda
2
3 sudo systemctl status ctxhdx
```

## 手順 11: マシンカタログの作成

マシンカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明については、「[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。

次のように、Linux VDA マシンを含むマシンカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシンカタログの作成手順と異なる点があります：

- オペレーティングシステムには、次を選択します：
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。
- 同じマシンカタログで、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 の以前のバージョンは、「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** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **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暗黙的に選択されません。

ヒント：

マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

### 手順 **12**：デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明については、「[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は、次の制約があります：

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

マシンカタログおよび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 2407](#)」を参照してください。

## Debian への Linux VDA の手動インストール

August 20, 2024



### 重要:

新規インストールの場合は、[簡単インストール](#)を使用して簡易インストールを行うことをお勧めします。簡単インストールは時間と労力を節約するだけでなく、本記事に記載されている手動インストールよりもエラーを減らすことができます。

### 手順 1: 構成ファイル情報および Linux マシンの準備

#### 手順 1a: ホスト名の設定

マシンのホスト名が確実に正しく報告されるようにするには、**/etc/hostname** ファイルを変更してマシンのホスト名のみを記述します。

#### hostname

#### 手順 1b: ホスト名へのループバックアドレスの割り当て

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ます。このためには、**/etc/hosts** ファイルの次の行の最初の 2 つのエントリに FQDN とホスト名が含まれるように編集します:

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、**hostname-fqdn**または**hostname**に対するその他の参照を削除します。

### 注:

Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。ホスト名は 15 文字以内である必要があります。

### ヒント:

a~z、A~Z、0~9、およびハイフン (-) の文字のみ使用してください。アンダースコア (\_)、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了し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。

#### 手順 1c: ホスト名の確認

マシンを再起動して、ホスト名が正しく設定されていることを確認します:

```
1 hostname
```

このコマンドによって、そのマシンの完全修飾ドメイン名 (FQDN) ではなく、そのホスト名のみが返されます。

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します：

```
1 hostname -f
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

手順 **1d**: マルチキャスト **DNS** の無効化

デフォルトの設定でマルチキャスト DNS (**mDNS**) が有効であるため、名前解決の結果に不整合が発生する場合があります。

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

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

変更後：

```
hosts: files dns
```

手順 **1e**: 名前解決とサービス到達可能性の確認

次のコマンドで、完全修飾ドメイン名が解決できることと、ドメインコントローラーと Delivery Controller から ping に応答があることを確認します：

```
1 nslookup domain-controller-fqdn
2
3 ping domain-controller-fqdn
4
5 nslookup delivery-controller-fqdn
6
7 ping delivery-controller-fqdn
```

完全修飾ドメイン名を解決できない、またはこれらのマシンのいずれかから ping に応答がない場合は、手順を確認してから次に進んでください。

手順 **1f**: 時刻同期の構成 (**chrony**)

VDA、Delivery Controller、ドメインコントローラーの間で正確な時刻同期を維持することは重要です。仮想マシン (VM) として Linux VDA をホストすると、時刻が不正確になる問題が発生する可能性があります。したがって、リモートのタイムサービスを使用して時刻を維持することをお勧めします。

chrony のインストール：

```
1 apt-get install chrony
```

ルートユーザーとして **/etc/chrony/chrony.conf** を編集し、次のように各リモートタイムサーバーに対応する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
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応する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、パブリックサーバーの **\*.pool.ntp.org** エントリなど、一覧にあるその他のサーバーまたはプールエントリを削除します。

変更を保存してから、次のコマンドで Chrony デーモンを再起動します:

```
1 sudo systemctl restart chrony
```

手順 **1g**: パッケージのインストール

```
1 sudo apt-get install -y libsasl2-2
2
3 sudo apt-get install -y libgtk2.0-0
```

手順 **1h**: リポジトリを追加して必要な依存関係をインストールする

Debian 11 の場合、**/etc/apt/sources.list** ファイルに **deb http://deb.debian.org/debian/bullseye main** 行を追加します。

手順 **1i**: 使用するデータベースのインストールと指定

注:

- VDI モードのみで SQLite を使用し、ホストされる共有デスクトップ配信モデルには 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 のために、SQLite と PostgreSQL は、それぞれを手動でインストールすることなく指定することができます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合は、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

このセクションでは、PostgreSQL と SQLite をインストールする方法と、どちらを使用するかを指定する方法について説明します。

**PostgreSQL** のインストール 次のコマンドを実行して、PostgreSQL をインストールします：

```
1 sudo apt-get update
2
3 sudo apt-get install -y postgresql
4
5 sudo apt-get install -y libpostgresql-jdbc-java
```

マシンの起動時または即時で PostgreSQL を起動するには、それぞれ次のコマンドを実行します：

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl start postgresql
```

**SQLite** をインストールする Debian の場合、次のコマンドを実行して SQLite をインストールします：

```
1 sudo apt-get install -y sqlite3
```

使用するデータベースを指定する SQLite と PostgreSQL の両方をインストールする場合、Linux VDA パッケージをインストールしてから **/etc/xdl/db.conf** を編集すると、どちらを使用するかを指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省きます。
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：

**/etc/xdl/db.conf** を使用して PostgreSQL のポート番号を構成することもできます。

## 手順 2：ハイパーバイザーの準備

サポートされるハイパーバイザー上で VM として Linux VDA を実行する場合、いくつかの変更が必要です。使用するハイパーバイザーのプラットフォームに合わせて、次の変更を行います。ベアメタルハードウェアで Linux マシンを実行する場合、変更は必要ありません。

## XenServer（旧称 Citrix Hypervisor）での時刻同期の修正

XenServer の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP と XenServer の問題が発生します。これは、NTP と Citrix Hypervisor の両方がシステムの時間を管理しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは、変更は必要ありません。

XenServer 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、XenServer の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます：

```
1 su -
2 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いるため、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います。こ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、以下の手順は必要ありません。

時刻同期機能が有効になっている場合は、ファイルに「1」と書き込んで無効にします：

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

この変更を永続化し、再起動後も保持するには、**`/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ため、次のようにしてシステムを再起動します：

```
1 su -
2 cat /proc/sys/xen/independent_wallclock
```

このコマンドは 1 を返します。

### Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想マシンでは、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間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にします。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、[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：

この方法は VMware および XenServer (旧称 Citrix Hypervisor) の場合とは異なります。VMware および XenServer では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期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な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーで問題が発生します。これは、NTP とハイパーバイザーの両方がシステムの時間を同期しようとするのが原因です。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

VMware Tools をインストールした状態で準仮想化 Linux カーネルを実行している場合は、次の操作を行います。

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. [仮想マシンのプロパティ] ダイアログボックスで、[オプション] タブをクリックします。
4. [VMware Tools] を選択します。
5. [詳細] ボックスで、[ホストとゲスト時刻を同期] チェックボックスをオフにします。

## 手順 3: Linux VM を Windows ドメインに追加

以下は、Linux マシンを Active Directory (AD) ドメインに追加する方法です:

- [Samba Winbind](#)
- Quest Authentication Service
- [Centrify DirectControl](#)
- [SSSD](#)
- [PBIS](#)

選択した方法の手順に従います。

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ことがあります。

## Samba Winbind

必要なパッケージのインストールまたは更新

```
1 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** デーモンを開始できるようにする 次のコマンドで、マシン起動時に Winbind デーモンが開始するように構成する必要があります。

```
1 sudo systemctl enable winbind
```

注:

`winbind` スクリプトが `/etc/init.d` にあることを確認します。

**Kerberos** を構成します ルートユーザーとして `/etc/krb5.conf` を開き、以下を設定します。

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
```

ここで **domain-dns-name** パラメーターは、DNS ドメイン名 (**example.com** など) です。 **REALM** は、大文字の Kerberos 領域名 (**EXAMPLE.COM** など) です。

**Winbind** 認証の構成 `vim /etc/samba/smb.conf` コマンドを実行して `/etc/samba/smb.conf` を開いてから、以下を設定します:

```
[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す。**REALM** は大文字の Kerberos 領域名です。

**nsswitch** の構成 `/etc/nsswitch.conf` を開き、`winbind` を次の行に追加します：

```
passwd: files systemd winbind
```

```
group: files systemd winbind
```

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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：

```
1 sudo net ads join <Kerberos realm name in uppercase> -U <domain user
  with permission to add computers to the domain>
```

#### Winbind の再起動

```
1 sudo systemctl restart winbind
```

**Winbind** 用の **PAM** の構成 次のコマンドを実行して、**[Winbind NT/Active Directory authentication]** オプションと **[Create home directory on login]** オプションが選択されているようにします：

```
1 sudo pam-auth-update
```

ヒント：

マシンがドメインに参加済みの場合にのみ、**winbind** デーモンは実行を続けます。

ドメインメンバーシップの確認 Delivery Controller を使用するには、Windows または Linux に関係なく、すべての VDA マシンで **Active Directory** にコンピューターオブジェクトが必要です。

次のように、**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：

```
1 sudo net ads testjoin
```

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

```
1 sudo net ads info
```

**Kerberos** 構成の確認 Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドによって、システムの **keytab** ファイルが作成済みで **keytab** ファイルに有効なキーが含まれていることを確認します：



```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します:

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します:

```
1 sudo klist
```

次のコマンドを使用して、マシンアカウントの詳細を調査します:

```
1 sudo net ads status
```

**ユーザー認証の確認** 次のように、**wbinfo** ツールを使用して、ドメインユーザーがドメインに対して認証できることを確認します:

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

ここで指定するドメインは AD ドメイン名で、Kerberos 領域名ではありません。bash シェルの場合、バックスラッシュ文字 (\) は、もう 1 つバックスラッシュ文字を指定してエスケープする必要があります。このコマンドにより、成功または失敗を示すメッセージが返されます。

Winbind PAM モジュールが正しく構成されていることを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\username
2
3 id -u
```

注:

SSH コマンドを正しく実行するには、SSH が有効で適切に機能していることを確認します。

次のコマンドで、**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:

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、ユーザーの Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します:

```
1 klist
```

次のコマンドで、セッションを終了します。

```
1 exit
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。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を表示できない場合、マシンを再起動して再試行します。

## Quest Authentication Service

ドメインコントローラーでの **Quest** の構成 次の操作は、Quest ソフトウェアを Active Directory ドメインコントローラーにインストールし、構成していることと、管理者特権が付与され、**Active Directory** にコンピューターオブジェクトを作成できることを前提としています。

**Linux VDA** マシンにドメインユーザーがログオンできるようにする Linux VDA マシンで HDX セッションを確立する必要がある各ドメインユーザーに対して、次の操作を行います。

1. [Active Directory ユーザーとコンピューター] 管理コンソールで、目的のユーザーアカウントの Active Directory ユーザーの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] チェックボックスをオンにします。
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:

この手順は、ドメインユーザーがコンソール、RDP、SSH、またはその他のリモート処理プロトコルを使用してログオンできるように設定する場合も同じです。

## Linux VDA での Quest の構成

**SELinux** ポリシー適用の回避策 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 ドメインソケットの IPC のメカニズムに干渉し、ドメインユーザーのログオンを妨げます。

この問題を回避するための便利な方法は、SELinux の無効化です。ルートユーザーとして、**/etc/selinux/config** を編集し、**SELinux** 設定を次のとおりに変更します:

## SELINUX=disabled

この変更にはマシンの再起動が必要です：

```
1 reboot
```

### 重要：

この設定は注意して使用してください。SELinux ポリシーの適用を無効にした後に再度有効にすると、ルートユーザーやその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てしまう可能性があります。

**VAS** デーモンの構成 次のように Kerberos チケットの自動更新を有効にして、切断する必要があります。認証（オフラインログイン）は無効にする必要があります：

```
1 sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-  
   interval 32400  
2  
3 sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-  
   auth false
```

このコマンドにより、更新間隔が9時間（32,400秒）に設定されます。すなわち、チケットのデフォルトの有効期間である10時間よりも1時間短くなります。チケットの有効期間がさらに短いシステムでは、より小さい値をこのパラメーターに設定します。

**PAM** および **NSS** の構成 HDX や、su、ssh、RDP などのその他のサービスを介したドメインユーザーのログイ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：

```
1 sudo /opt/quest/bin/vastool configure pam  
2  
3 sudo /opt/quest/bin/vastool configure nss
```

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

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

**user** は、コンピューターを Active Directory ドメインに追加す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（example.com など）です。

ドメインに追加後、Linux マシンを再起動します。

ドメインメンバーシップの確認 Delivery Controller を使用するには、Windows または Linux に関係なく、すべての VDA マシンで Active Directory にコンピューターオブジェクトが必要です。Quest によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します：

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

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下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** PAMを使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 ssh localhost -l domain\username  
2  
3 id -u
```

次のコマンドで、**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：

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します：

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

次のコマンドで、セッションを終了します。

```
1 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。

## Centrify DirectControl

**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、次のように Centrify の **adjoin** コマンドを使用して、Linux マシンを Active Directory ドメインに追加します：

```
1 su -  
2 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 ドメインに追加す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 マシンを追加す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 を使用するには、Windows または Linux に関係なく、すべての VDA マシンで Active Directory にコンピューターオブジェクトが必要です。Centrify により追加された

Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します：

```
1 su -
2
3 adinfo
```

**Joined to domain** 値が有効であることと、**CentrifyDC mode** で **connected** が返されることを確認します。CentrifyDC mode が starting のまま変化しない場合は、Centrify クライアントにサーバーとの接続の問題、または認証の問題が発生しています。

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
1 adinfo --sysinfo all
2
3 adinfo --diag
```

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

```
1 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** を構成します Kerberos をインストールするには、次のコマンドを実行します：

```
1 sudo apt-get install krb5-user
```

Kerberos を構成するには、**/etc/krb5.conf** をルートとして開き、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
rdns = false

[realms]
REALM = {
admin_server = domain-controller-fqdn
kdc = domain-controller-fqdn
}
```

[domain\_realm]

*domain-dns-name = REALM*

*.domain-dns-name = REALM*

ここで `domain-dns-name` パラメーターは、DNS ドメイン名 (example.com など) です。 *REALM* は、大文字の Kerberos 領域名 (EXAMPLE.COM など) です。

ドメインに参加する SSSD を構成して、Active Directory を ID プロバイダーおよび認証の Kerberos として使用します。ただし、SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する AD のクライアント機能が提供されていません。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** および **Samba** に関する情報のみを提供します。

- **adcli** を使用してドメインに参加する場合は、次の手順を実行します:

1. **adcli** をインストールします。

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

2. **adcli** でドメインに参加させます。

次を使用して古いシステム keytab ファイルを削除し、ドメインに参加させます。

```
1 su -
2
3 rm -rf /etc/krb5.keytab
4
5 adcli join domain-dns-name -U user -H hostname-fqdn
```

**user** は、ドメインにマシンを追加する権限があるドメインユーザーです。**hostname-fqdn** は、完全修飾ドメイン名形式のマシンのホスト名です。

**-H** オプションは、**adcli** が、Linux VDA で必要な `host/hostname-fqdn@REALM` という形式で SPN を生成するのに必要です。

3. システムの Keytab を確認します。

`sudo klist -ket` コマンドを実行して、システムの keytab ファイルが作成されていることを確認します。

各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致するかを検証します。

- **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。

```
1 sudo apt-get install samba krb5-user
```

## 2. Samba を構成します。

`/etc/samba/smb.conf` を開き、次を設定します。

```
[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す。`REALM` は大文字の Kerberos 領域名です。

## 3. Samba でドメインに参加させます。

ドメインコントローラーがアクセス可能で、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。

```
1 sudo net ads join <the Kerberos realm name in uppercase> -U <
  domain user with permission to add computers to the domain>
```

**SSSD のセットアップ** 必要なパッケージのインストールまたは更新:

必要な SSSD および構成パッケージがインストールされていない場合、インストールします。

```
1 sudo apt-get install sssd
```

パッケージが既にインストールされている場合、更新することをお勧めします。

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

**SSSD の構成** SSSD デーモンを起動する前に、SSSD 構成の変更が必要です。SSSD の一部のバージョンでは、`/etc/sss/sss.conf` 構成ファイルはデフォルトではインストールされないため、手動で作成する必要があります。root として `/etc/sss/sss.conf` を作成するか開いて、次を設定します:

```
[sss]
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** に設定されるため、SSSD 自体が Windows SID を Unix UID にマッピングします。設定しない場合、Active Directory が POSIX 拡張を提供できるようにする必要があります。PAM サービス (ctxhdx) は、ad\_gpo\_map\_remote\_interactive に追加されます。

ここで **domain-dns-name** パラメーターは、DNS ドメイン名 (example.com など) です。REALM は、大文字の Kerberos 領域名 (EXAMPLE.COM など) です。NetBIOS ドメイン名を構成するための要件はありません。

構成設定について詳しくは、sssd.conf および sssd-ad に関する man ページを参照してください。

SSSD デーモンでは、構成ファイルに所有者読み取り権限のみが設定されている必要があります。

```
1 sudo chmod 0600 /etc/sss/sss.conf
```

**SSSD** デーモンの起動 次のコマンドを実行して、SSSD デーモンを起動し、マシンの起動時にもデーモンを起動できるようにします。



```
1 sudo systemctl start sssd
2
3 sudo systemctl enable sssd
```

**PAM 構成** 次のコマンドを実行して、**[SSS authentication]** オプションと **[Create home directory on login]** オプションが選択されているようにします:

```
1 sudo pam-auth-update
```

**ドメインメンバーシップの確認** Delivery Controller を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。

- **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** Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドによって、システムの keytab ファイルが作成済みで keytab ファイルに有効なキーが含まれていることを確認します:

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の `kinit` コマンドを実行し、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に対して認証します:

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT がキャッシュされたことを確認します:

```
1 sudo klist
```

**ユーザー認証の確認** SSSD は、デーモンで直接認証をテストするコマンドラインツールを提供しません。PAM 経由でのみ完了できます。

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が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。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

ユーザーの **klist** コマンドで返される Kerberos チケットが正しく、期限切れではないことを確認します。

ルートユーザーとして、前述の **id -u** コマンドで返された 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：

```
1 ls /tmp/krb5cc_uid
```

KDE または Gnome Display Manager に直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「[手順 6: Linux VDA のインストール](#)」に進みます。

## PBIS

必要な **PBIS** パッケージをダウンロードする

```
1 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

```
1 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

```
1 sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

**Windows** ドメインへの参加 ドメインコントローラーがアクセス可能で、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：

```
1 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 (example.com など) です。

注： Bash をデフォルトの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 を使用するには、すべての VDA マシン (Windows と Linux VDA) で **Active Directory** にコンピューターオブジェクトが必要です。PBIS によって追加された Linux マシンがドメインに存在することを確認するには、次のコマンドを実行します:

```
1 /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の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

ユーザー認証の確認 PAM を使用した PBIS のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことが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。

```
1 sudo ssh localhost -l domain\user
2
3 id -u
```

次のコマンドで、**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:

```
1 ls /tmp/krb5cc_uid
```

次のコマンドで、セッションを終了します。

```
1 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 をインストールする

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

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

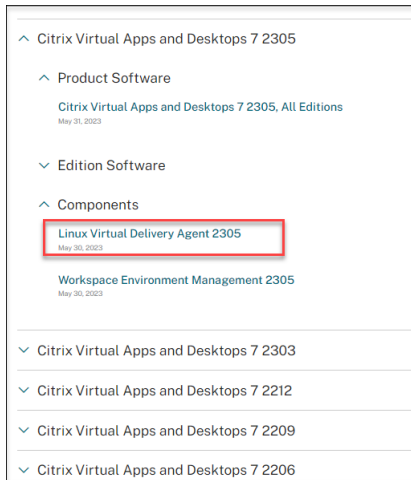
.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを特定します。

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。

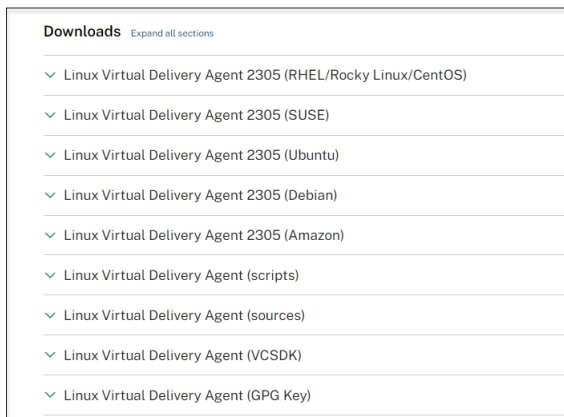
#### 手順 5: Linux VDA パッケージのダウンロード

1. [Citrix Virtual Apps and Desktops のダウンロードページ](#) にアクセスします。

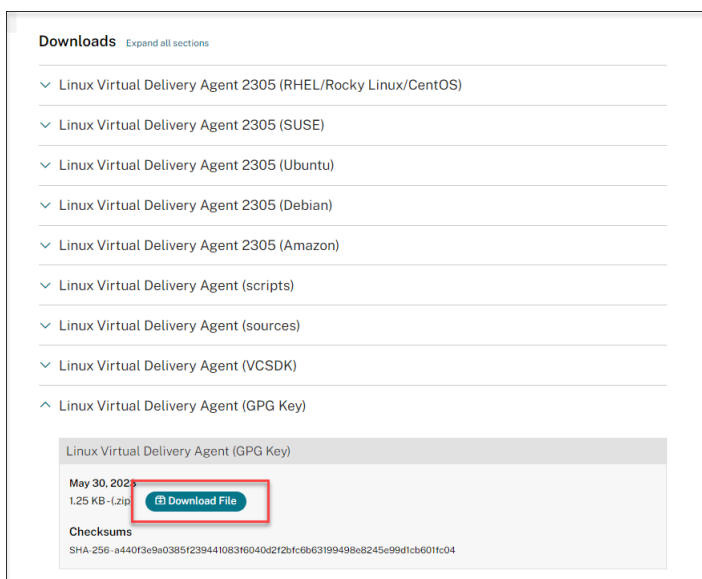
- 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
- Components** を展開して Linux VDA を見つけます。例：



- Linux VDA のリンクをクリックして、Linux VDA のダウンロードファイルにアクセスします。



- 使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
- Linux VDA パッケージの整合性を検証するために使用できる GPG 公開キーをダウンロードします。例：



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

```
1 sudo apt-get install dpkg-sig
2 gpg --import <path to the public key>
3 dpkg-sig --verify <path to the Linux VDA package>
```

### 手順 6: Linux VDA のインストール

#### 手順 6a: Linux VDA のインストール

次のように、Debian Package Manager を使用して Linux VDA ソフトウェアをインストールします:

**Debian 12.5** の場合:

```
1 sudo dpkg -i xendesktopvda_<version>.debian12_amd64.deb
```

**Debian 11** の場合:

```
1 sudo dpkg -i xendesktopvda_<version>.debian11_amd64.deb
```

**Debian 12.5** の依存関係一覧:

```
1 openjdk-17-jdk >= 17
2
3 imagemagick >= 8:6.9.11
4
5 ufw >= 0.36
6
7 desktop-base >= 12.0.6
8
9 libxrandr2 >= 2:1.5.2
```

```
10
11 libxtst6 >= 2:1.2.3
12
13 libxm4 >= 2.3.8
14
15 util-linux >= 2.38
16
17 gtk3-nocsd >= 3
18
19 bash >= 5.2
20
21 findutils >= 4.9.0
22
23 sed >= 4.9
24
25 cups >= 2.4
26
27 ghostscript >= 10.0.0~
28
29 libmspack0 >= 0.11
30
31 ibus >= 1.5
32
33 libgoogle-perftools4 >= 2.10~
34
35 libpython3.11 >= 3.11~
36
37 libsasl2-modules-gssapi-mit >= 2.1.~
38
39 libnss3-tools >= 2:3.87
40
41 libqt5widgets5 >= 5.15~
42
43 mutter >= 43.8
44
45 libqrencode4 >= 4.1.1
46
47 libimlib2 >= 1.10.0
48
49 libfuse2 >= 2.9.9
50
51 pulseaudio-utils >= 16.1
```

**Debian 11** の依存関係一覧:

```
1 libnss3-tools >= 2:3.61
2
3 libfuse2 >= 2.9
4
5 openjdk-11-jdk >= 11
6
7 imagemagick >= 8:6.9.10
8
```

```
9 ufw >= 0.36
10
11 desktop-base >= 10.0.2
12
13 libxrandr2 >= 2:1.5.1
14
15 libxtst6 >= 2:1.2.3
16
17 libxm4 >= 2.3.8
18
19 util-linux >= 2.33
20
21 gtk3-nocsd >= 3
22
23 bash >= 5.0
24
25 findutils >= 4.6.0
26
27 sed >= 4.7
28
29 cups >= 2.2
30
31 ghostscript >= 9.53~
32
33 libmspack0 >= 0.10
34
35 ibus >= 1.5
36
37 libgoogle-perftools4 >= 2.7~
38
39 libpython3.9 >= 3.9~
40
41 libsasl2-modules-gssapi-mit >= 2.1.~
42
43 libqt5widgets5 >= 5.5~
44
45 mutter >= 3.38.6~
46
47 libqrencode4 >= 4.0.0
48
49 libimlib2 >= 1.5.1
```

注:

このバージョンの Linux VDA でサポートされている Linux ディストリビューションと Xorg のバージョンについては、「[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)

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

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2
3 sudo apt-get install -f
```

注:

- 既存のインストールをアップグレードすると、`/etc/xdl`にある構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。
- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイルを処理するために、パッケージマネージャー **rpm** または **dpkg** に委任します。以下では、`rpm` と `dpkg` が構成ファイルの変更でどのように機能するかについて説明しています:

- **rpm**: by default keeps the local version and saves the new version from the package with a `.rpmnew` extension.
- **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as `.dpkg-new` or `.dpkg-dist`, use the following command:

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

## 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、次のガイドを参照してください:

- [XenServer](#)
- [VMware ESX](#)
- [Nutanix 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、GPU を VM に割り当てます。
3. 仮想マシンを起動します。
4. ゲスト VM ドライバーを VM にインストールします。

## 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、**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ます。OS にこのロケールがない場合は、**sudo locale-gen en\_US.UTF-8** コマンドを実行します。Debian の場合は、**# en\_US.UTF-8 UTF-8** 行のコメントを解除して **/etc/locale.gen** ファイルを編集してから、**sudo locale-gen** コマンドを実行します。

パッケージのインストール後、**ctxsetup.sh** スクリプトを実行して、Linux VDA を構成する必要があります。このスクリプトは、変更を行う前に環境を確認し、すべての依存コンポーネントがインストールされていることが確認されます。必要に応じて、いつでもこのスクリプトを再実行して設定を変更できます。

このスクリプトは、手動で質問に回答しながら、または事前に構成した回答を使用して自動で実行できます。続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します:

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

**質問に回答する構成**

次のようにして、質問に回答する手動構成を実行します:

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

**自動化された構成**

インストールを自動化するために、環境変数を使用して、セットアップスクリプトで必要となるオプションを指定できます。必要な変数がすべて指定されていると、スクリプトによってユーザーに情報の入力を求めるメッセージが表示されることがなくなり、インストール処理をスクリプト化できます。

サポートされる環境変数には次のようなものがあります:

- **CTX\_XDL\_NON\_DOMAIN\_JOINED=' y|n'**  
- マシンをドメインに参加させるかどうか。デフォルト値は ' n' です。ドメイン参加済みシナリオの場合は ' n' に設定します。
- **CTX\_XDL\_AD\_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。
- **CTX\_XDL\_DDC\_LIST=' <list-ddc-fqdns>'** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の、完全修飾ドメイン名 (FQDN) のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。
- **CTX\_XDL\_VDI\_MODE=' y|n'** - 専用デスクトップ配信モデル (VDI) またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を ' y' に設定します。

- **CTX\_XDL\_HDX\_3D\_PRO=' y|n'** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジーです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX\_XDL\_VDI\_MODE= 'y' となります)。
- **CTX\_XDL\_START\_SERVICE = 'y|n'** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。
- **CTX\_XDL\_REGISTER\_SERVICE = 'y|n'** - Linux Virtual Desktop サービスは、マシンの起動後に開始します。
- **CTX\_XDL\_ADD\_FIREWALL\_RULES = 'y|n'** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート (デフォルトではポート 80 およびポート 1494) を自動で開放できます。
- **CTX\_XDL\_DESKTOP\_ENVIRONMENT=gnome/gnome-classic/kde/mate/ <none>** '-セッションで使用使用する GNOME、GNOME クラシック、または MATE デスクトップ環境を指定します。' <none> 'に設定すると、VDA で構成されているデフォルトのデスクトップが使用されます。  
  
コマンドを実行したり、システムトレイを使用したりして、デスクトップ環境を切り替えることもできます。詳しくは、「[デスクトップ切り替えコマンド](#)」および「[システムトレイ](#)」を参照してください。
- **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 をインストールするパス。デフォルトのパスは '**/usr/bin**' です。
- **CTX\_XDL\_VDA\_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX\_XDL\_SITE\_NAME =<dns-name>** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、'**<none>**' に設定します。
- **CTX\_XDL\_LDAP\_LIST=' <list-ldap-servers>'** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。例: 「ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268」。Active Directory フォレストでより高速な LDAP クエリを有効にするには、ドメインコントローラーで [グローバルカタログ] を有効にし、関連する LDAP ポート番号で 3268 を指定します。この変数は、デフォルトでは '**<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使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます (例: OU=VDI,DC=mycompany,DC=com)。不要な場合は、'**<none>**' に設定します。
- **CTX\_XDL\_SUPPORT\_DDC\_AS\_CNAME=' y|n'** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。

次のようにして、環境変数を設定し、構成スクリプトを実行します：

```

1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

`sudo` コマンドに **-E** オプションを指定して実行し、作成する新しいシェルに既存の環境変数を渡します。最初の行として **#!/bin/bash** を記述し、前述のコマンドからなるシェルスクリプトファイルを作成することをお勧めします。

または、次のようにして、1つのコマンドですべてのパラメーターを指定することができます：

```

1 sudo CTX_XDL_NON_DOMAIN_JOINED='n' \
2 CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \
3 CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \
4 CTX_XDL_VDI_MODE='y|n' \
5 CTX_XDL_HDX_3D_PRO='y|n' \
6 CTX_XDL_START_SERVICE='y|n' \
7 CTX_XDL_REGISTER_SERVICE='y|n' \
8 CTX_XDL_ADD_FIREWALL_RULES='y|n' \
9 CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>' \
10 CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \
11 CTX_XDL_VDA_PORT='<port-number>' \
12 CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \
13 CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \
14 CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \
15 CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \
16 /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

#### 構成変更の削除

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しないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します：

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

構成変更を削除するには:

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

重要:

このスクリプトにより、すべての構成データがデータベースから削除され、Linux VDA を操作できなくなります。

構成ログ

**ctxsetup.sh** および **ctxcleanup.sh** スクリプトでは、コンソールにエラーが表示され、構成ログファイル **/tmp/xdl.configure.log** に追加情報が書き込まれます。

Linux VDA サービスを再起動し、変更を反映させます。

### Linux VDA ソフトウェアのアンインストール

Linux VDA がインストールされているかどうかを確認したり、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、次のコマンドを実行します。

```
1 dpkg -l xendesktopvda
```

詳細を表示するには、次のコマンドを実行します。

```
1 apt-cache show xendesktopvda
```

Linux VDA ソフトウェアをアンインストールには、次のコマンドを実行します:

```
1 dpkg -r xendesktopvda
```

注:

Linux VDA ソフトウェアをアンインストールすると、関連付けられた 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 のインストールより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するパッケージは削除されません。

ヒント:

このセクションでは、PostgreSQL など、依存するパッケージの削除方法については説明していません。

### 手順 9: XDPing の実行

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境での一般的な構成の問題を確認します。詳しくは、「**XDPing**」を参照してください。

## 手順 10: Linux VDA の実行

**ctxsetup.sh** スクリプトを使用して Linux VDA を構成したら、次のコマンドを使用して Linux VDA を制御します。

### Linux VDA の起動:

Linux VDA サービスを起動するには、次のコマンドを実行します:

```
1 sudo systemctl start ctxhdx
2
3 sudo systemctl start ctxvda
```

### Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl stop ctxhdx
```

#### 注:

**ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行して監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

### Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
1 sudo systemctl stop ctxvda
2
3 sudo systemctl restart ctxhdx
4
5 sudo systemctl restart ctxvda
```

### Linux VDA の状態の確認:

Linux VDA サービスの実行状態を確認するには、次のコマンドを実行します:

```
1 sudo systemctl status ctxvda
2
3 sudo systemctl status ctxhdx
```

## 手順 11: マシンカタログの作成

マシンカタログを作成し、Linux VDA マシンを追加する手順は、従来の Windows VDA での方法と似ています。このタスクを完了する方法の説明については、「[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。

次のように、Linux VDA マシンを含むマシンカタログの作成にはいくつかの制約があるため、Windows VDA マシンのマシンカタログの作成手順と異なる点があります：

- オペレーティングシステムには、次を選択します：
  - ホストされる共有デスクトップ配信モデルの場合、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、シングルセッション **OS** オプション。
- 同じマシンカタログで、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 の以前のバージョンは、「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** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **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暗黙的に選択されません。

ヒント：

マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

### 手順 12: デリバリーグループの作成

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手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「[デリバリーグループの作成](#)」を参照してください。

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は、次の制約があります：

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 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。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「[Citrix Virtual Apps and Desktops 7 2407](#)」を参照してください。

## 構成

July 16, 2024

このセクションでは、機能の説明、構成、トラブルシューティングなど、Linux VDA の機能について詳しく説明します。

## 管理

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [Linux VDA データ収集プログラム](#)
- [HDX Insight](#)
- [Citrix Telemetry Service との統合](#)
- [Citrix DaaS Standard for Azure の Linux VDA 自己更新](#)
- [Linux VM および Linux セッションのメトリック](#)
- [ログ収集](#)
- [セッションのシャドウ](#)
- [監視サービスデーモン](#)
- [トラブルシューティング](#)
- [その他](#)
  - [HTML5 向け Citrix Workspace アプリのサポート](#)
  - [Python 3 仮想環境の作成](#)
  - [NIS の Active Directory との統合](#)
  - [IPv6](#)
  - [LDAPS](#)
  - [Xauthority](#)

## Linux VDA データ収集プログラム

August 21, 2024

Linux VDA のインストールが完了すると、データ収集プログラムに自動的に参加することになります。データ収集プログラムは統計と使用状況データを収集し、そのデータを Citrix Analytics に送信して、Citrix 製品の品質とパフォーマンスの向上に役立てます。データのアップロードは、セッションの起動時に行われます。

## レジストリ設定

Linux VDA データ収集プログラムはデフォルトで有効に設定されています。次のレジストリ設定により、プログラムを有効または無効にできます：

場所: HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\CEIP

名前: **CEIPSwitch**

値: 1 = 無効、0 = 有効 (デフォルト)

指定しないと、Linux VDA データ収集プログラムは有効に設定されます。

次のコマンドを実行して、Linux VDA データ収集プログラムを無効にすることができます：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

## Linux VDA から収集されたデータ

次の表に、収集されるデータの種類の例を示します。

データポイント	キー名	説明
マシンのグローバル一意識別子	<b>machine_guid</b>	データの送信元のマシンを識別する GUID 文字列
Linux OS の名前およびバージョン	<b>os_name_version</b>	マシンの Linux OS の名前とバージョンを示す文字列
AD ソリューション	<b>ad_solution</b>	マシンのドメイン参加方式を示す文字列
Linux カーネルのバージョン	<b>kernel_version</b>	マシンのカーネルバージョンを示す文字列
GPU の種類	<b>gpu_type</b>	VDA マシンの GPU の種類
CPU コア	<b>cpu_cores</b>	マシンの CPU コア数を示す整数
CPU 周波数	<b>cpu_frequency</b>	CPU の周波数 (MHz) を示す浮動小数点数
物理メモリサイズ	<b>memory_size</b>	物理メモリのサイズ (KB) を示す整数
LVDA バージョン	<b>vda_version</b>	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 の更新または新規のインストール	<b>update_or_fresh_install</b>	現在の Linux VDA パッケージが新規インストールであるのか更新であるのかを示す文字列



データポイント	キー名	説明
VDI モードが有効かどうか	<b>vdi_mode</b>	VDI モードが有効かどうかを示す文字列
LVDA のインストール方法	<b>install_method</b>	現在の Linux VDA パッケージが MCS、PVS、簡単インストール、または手動インストールのいずれかでインストールされたかを示す文字列。
HDX 3D Pro が有効かどうか	<b>hdx_3d_pro</b>	マシンで HDX 3D Pro が有効かどうかを示す文字列
BCR が有効かどうか	<b>bcr</b>	ブラウザコンテンツのリダイレクト (BCR) がこのマシンで有効になっているかどうかを示す文字列
システムのロケール	<b>system_locale</b>	このマシンのロケールを示す文字列
ファーム ID	<b>farm_id</b>	ファーム ID を示す文字列。
VDA 仮想化の種類	<b>vda_virtualization</b>	VDA を作成したハイパーバイザーを示す文字列
セッションキー	<b>session_key</b>	データの発生元のセッションを識別
リソースの種類	<b>resource_type</b>	起動されたセッションのリソースの種類を示すテキスト文字列: デスクトップまたは<appname>
Receiver クライアントの種類	<b>receiver_type</b>	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	<b>receiver_version</b>	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	<b>printing_count</b>	セッションで印刷機能を使用した回数を示す整数
USB リダイレクト回数	<b>usb_redirecting_count</b>	セッションで USB デバイスを使用した回数を示す整数
<b>Gfx</b> プロバイダーの種類	<b>gfx_provider_type</b>	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
シャドウの回数	<b>shadow_count</b>	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	<b>ctxism_select</b>	ユーザーが選択したすべての言語が組み合わされた長い文字列

データポイント	キー名	説明
スマートカードリダイレクトカウン ト	<b>scard_redirecting_count</b>	スマートカードリダイレクトがセッ ションログオンおよびセッション中 アプリのユーザー認証に使用される 回数を示す整数
ビデオコーデックの種類	<b>graphic_video_codec_type</b>	どのビデオコーデックが Thinwire に使用されているかを示す文字列。
ログオン資格情報の種類	<b>credentials_type</b>	Linux VDA へのログオンに使用され る資格情報の種類を示す整数
ウォーターマーク	ウォーターマーク	セッションウォーターマークが有効 になっているかどうかを示す文字列
ウォーターマークの透明度	<b>watermark_transparency</b>	ウォーターマークの透明度を示す整 数
ウォーターマークのカスタムテキス トの長さ	<b>watermark_custom_text_len</b>	ウォーターマークのカスタムテキス トの長さを示す整数
MTU	<b>mtu</b>	このセッションで最大転送単位 (MTU) が使用されるかどうかを示 す文字列
MTU MSS	<b>mtu_mss</b>	最大セグメントサイズ (MSS) を示 す整数
ファイル転送	<b>filetrans</b>	ファイル転送ポリシー設定を示す整 数
ファイル転送のアップロード数	<b>filetrans_upload_count</b>	セッションで「アップロード」アイ コンがファイル転送のために使用さ れる回数を示す整数
ファイル転送のダウンロード数	<b>filetrans_download_count</b>	セッションで「ダウンロード」アイ コンがファイル転送のために使用さ れる回数を示す整数

## HDX Insight

August 20, 2024

### 概要

Linux VDA では、[HDX Insight](#)機能の一部をサポートしています。

### インストール

インストールする必要がある依存関係パッケージはありません。

### 使用状況

HDX Insight は、Citrix Workspace アプリと Linux VDA の間で Citrix ADC を介して渡される ICA メッセージを分析します。すべての HDX Insight データは、NSAP 仮想チャネルから圧縮されずに送信されます。NSAP 仮想チャネルはデフォルトでは有効になっています。

以下のコマンドで、それぞれ NSAP 仮想チャネルを無効、または有効にします：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP" -d "0x00000000" --force
```

```
1 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

データポイントがまったく表示されない

2 通りの原因が考えられます：

- HDX Insight が正しく構成されていません。

たとえば、Citrix ADC で AppFlow が有効になっていないか、Citrix ADM で不正な Citrix ADC インスタンスが構成されています。

- Linux VDA で ICA コントロール仮想チャネルが開始されていません。

```
ps aux | grep -i ctxctl
```

`ctxctl` が実行されていない場合は、Citrix にバグをレポートするよう管理者に連絡します。

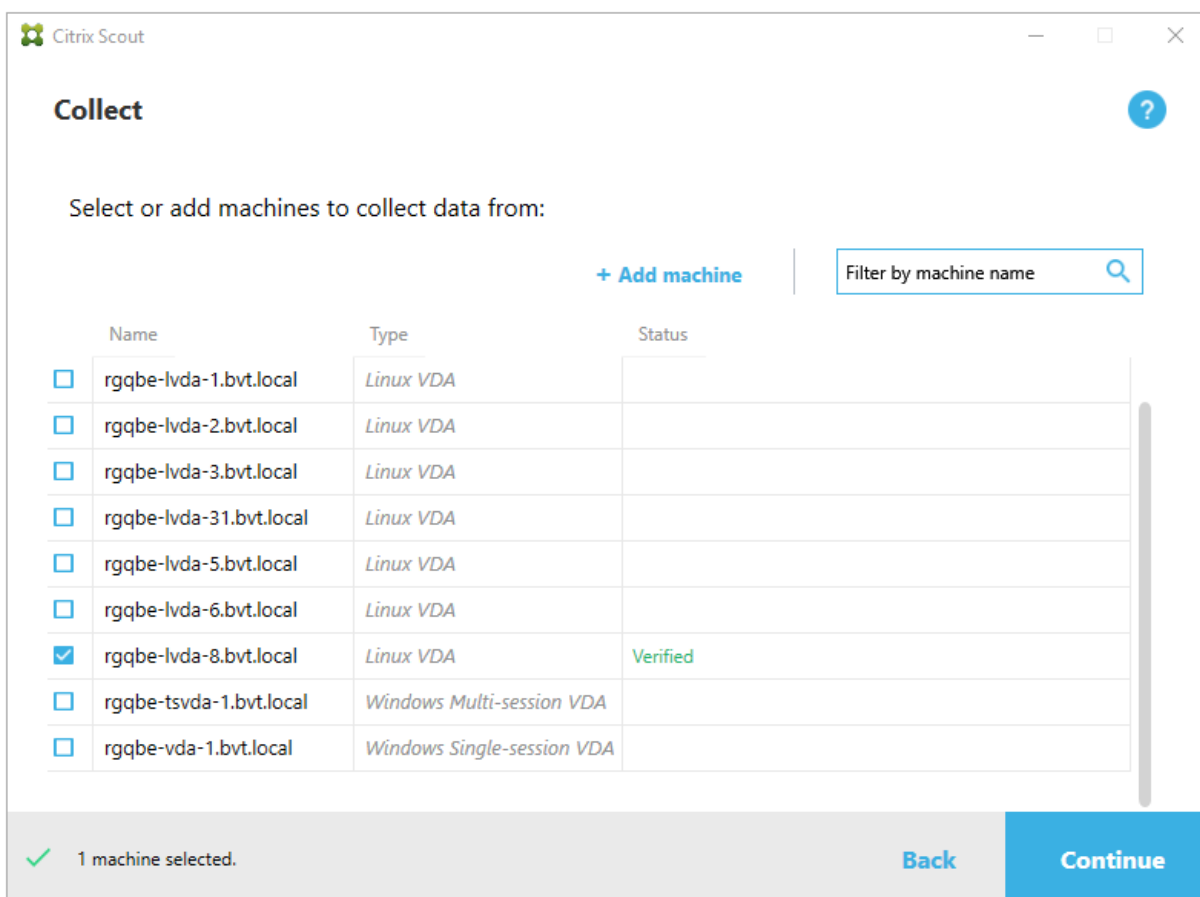
アプリケーションデータポイントがまったく表示されない

シームレス仮想チャネルが有効になっていることおよび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ます。

## Citrix Telemetry Service との統合

August 20, 2024

Linux VDA ソフトウェアに統合された Citrix Telemetry Service (`ctxtelemetry`) で Citrix Scout を実行し、`/opt/Citrix/VDA/bin/xdlcollect.sh` スクリプトを使用して Linux VDA のログを収集できます。



### Citrix Telemetry Service の有効化および無効化

- この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。
- この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、デフォルトでは TCP/IP ポート 7503 で Citrix Scout をリスンします。Delivery Controller で TCP/IP ポート 7502 を使用して、Citrix Scout と通信します。

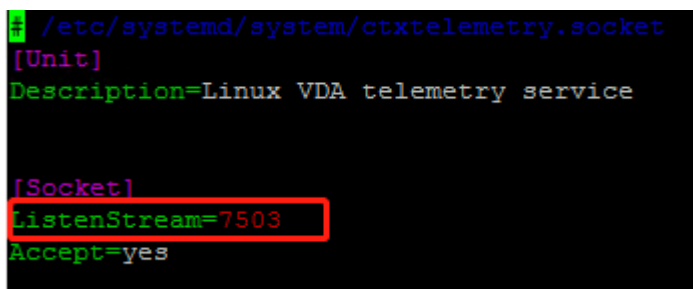
VDA のインストール後にポートを変更するには、以下を実行します：

1. Scout と通信するためのポートを変更するには、以下のコマンドを実行します。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t REG_DWORD -v "TelemetryServicePort" -d <port number> --force
```

2. Scout をリスンするためのソケットポートを変更するには、以下のコマンドを実行して `ctxtelemetry.socket` ファイルを開き、編集します。

```
1 sudo vi /etc/systemd/system/ctxtelemetry.socket
```



```
1 /etc/systemd/system/ctxtelemetry.socket
[Unit]
Description=Linux VDA telemetry service

[Socket]
ListenStream=7503
Accept=yes
```

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。

```
1 sudo systemctl daemon-reload
2 sudo systemctl stop ctxtelemetry.socket
3 sudo systemctl start ctxtelemetry.socket
```

4. ファイアウォールの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、**sudo ufw allow 7503** コマンドを実行してポート 7503 を有効にします。

注:

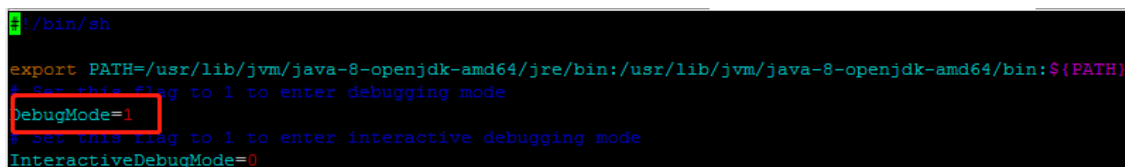
代わりに `ctxsetup.sh` を実行して、前述の手順 3 と 4 を自動化することもできます。

## デバッグモード

Citrix Telemetry Service が正常に機能していない場合、デバッグモードで原因を調査できます。

1. デバッグモードを有効にするには、以下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。

```
1 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      0 127.0.0.0:53            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/unic
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、以下のコマンドを実行して Citrix Telemetry Service を停止できます。

```
1 sudo netstat -ntlp
2 Kill -9 1958
```

3. Citrix Telemetry Service を再起動するには、Scout で Linux VDA を選択し、/var/log/xdl/で telemetry-debug.log を見つけます。

#### サービスの待機時間

ソケットポートを開く `systemd` デーモンは、デフォルトで起動し、ほとんどリソースを使用しません。Citrix Telemetry Service はデフォルトで停止し、Delivery Controller からログ収集要求があった場合のみ起動します。ログ収集の完了後、サービスは 15 分間新しい収集要求を待ち、要求がない場合は再度停止します。この待機時間は以下のコマンドで構成できます。最小値は 10 分です。10 分より少ない値を設定すると、最小値の 10 分が設定されます。待機時間の設定後、サービスを停止し再起動します。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
  VirtualDesktopAgent" -v "TelemetryServiceIdleTimeoutInMinutes" -d <
  number> -t REG_DWORD
```

#### 確認テスト

収集の開始前に、指定した各マシンについて自動で確認テストが実行されます。これらのテストで、要件が満たされているか確認されます。あるマシンでテストが失敗した場合、Scout には修正アクション案を含むメッセージが表示されます。確認テストについて詳しくは、Citrix Scout ドキュメントの「[確認テスト](#)」を参照してください。

## Azure を使用した Linux VDA の自動更新

August 20, 2024

この機能によって、Linux VDA ソフトウェアを即座に、またはスケジュールされた時間に自動的に更新することができます。これは、Citrix DaaS Standard for Azure (Citrix Virtual Apps and Desktops Standard for Azure の新名称) で Linux VDA を作成する場合に役立ちます。Azure の仮想マシンの管理者特権は必要ありません。詳しくは、「[Citrix DaaS Standard for Azure で Linux VDA を作成](#)」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：

手順 **1**：更新情報と新しい **VDA** パッケージを **Azure** コンテナにアップロードする

手順 **1a**：Azure ストレージアカウントでコンテナを作成し、コンテナアクセスレベルを [**BLOB (BLOB 専用の匿名読み取りアクセス)**] に設定します。

注：

Azure コンテナと BLOB は、お客様が独占的に保有および管理するものです。Citrix は、セキュリティ上の問題について責任を負いません。データのセキュリティとコスト効率を確保するには、自動更新が終わるたびにコンテナのアクセスレベルを [プライベート (匿名アクセスはありません)] に設定します。

手順 **1b**：VDA 更新情報を UpdateInfo.json という名前の JSON ファイルに組み込みます。ファイル形式の例については、次のブロックを参照してください：

```
1 {
2
3   "Version": "21.04.200.4",
4   "Distributions": [
5     {
6
7     "TargetOS": "RHEL7_9",
8     "PackageName": "",
9     "PackageHash": ""
10    }
11   ,
12   {
13
14   "TargetOS": "UBUNTU20_04",
15   "PackageName": "",
16   "PackageHash": ""
17   }
18 }
```

```
19 ]
20 }
```

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 つのアイテムが含まれています：

- “**TargetOS**”：“RHEL7\_9” (RHEL 7、CentOS 7、および Amazon Linux 2 の場合)、または “UBUNTU20\_04” のいずれかである必要があります。**ctxmonitord** は他のディストリビューションを認識しません。
- “**PackageName**”：指定されたバージョンの VDA パッケージのフルネーム。
- “**PackageHash**”：`shasum -a 256 <pkgname>` コマンドを使用して計算する SHA-256 値。

手順 1c: JSON ファイルと新しいバージョンの Linux VDA パッケージを Azure コンテナにアップロードします。

手順 2: マスターイメージまたは各 VDA で自動更新機能を有効にする

デフォルトでは、自動更新は無効になっています。Citrix DaaS Standard for Azure で Linux VDA を作成する場合、この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 でこの機能を直接有効にします。

自動更新を有効にするには、次のようなコマンドを実行して、`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\` でレジストリキーを編集します。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_DWORD" -v "fEnabled" -d "0
   x00000001" --force
2
3 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "ScheduledTime" -d "
   Immediately" --force
4
5 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "Url" -d "<Your-Azure-
   Container-Url>" --force
6
7 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "CaCertificate" -d "<Local
   -Certificate-Path-of-PortalAzureCom>" --force
```

次の表に、レジストリ設定を示します。

レジストリ設定	説明
fEnabled	この設定は必須です。デフォルト値は 0 です。これは、自動更新が無効になっていることを意味します。1 に設定すると、自動更新が有効になります。
Url	この設定は必須です。Azure コンテナの URL を設定して、更新情報と新しい VDA パッケージを取得します。



レジストリ設定	説明
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 パッケージをダウンロードした直後に更新を実行することを意味します。この選択肢は、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、パッケージをダウンロードするときにライブセッションがあると、ユーザーエクスペリエンスが損なわれる可能性があります。[NextStart] は、 <b>ctxmonitord</b> の次の開始時に更新を実行することを意味します。この選択肢は、ダウンロード速度が速くなく、更新が緊急でない場合に適しています。
CaCertificate	この設定はオプションです。Azure コンテナの URL を確認する PEM 証明書のフルパスを設定します。Azure BLOB の場合、Web ブラウザーから取得されて PEM に変換される portal.azure.com の証明書にすることができます。セキュリティのためには、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 または Debian ではサポートされていません。これは、Ubuntu または Debian で指定された証明書ファイルを使用してピアを検証するときに <code>curl</code> コマンドが正常に機能しないためです。証明書の最小特権が設定されているか確認してください。

**ctxmonitord** が再起動すると、最初に **Url** にクエリを実行して 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** は更新バージョンと現在のバージョンを比較します。現在のバージョンが以前のバージョンの場合、このサービスによって Azure から新しいバージョンの VDA パッケージがダウンロードされ、ローカルに保存されます。その後、[ScheduledTime] の設定に従って更新が実行されます。オンプレミス環境の場合、**ctxmonitord** を直接再起動して更新をトリガーできます。ただし、仮想マシンに対する管理者特権がない Citrix DaaS Standard for Azure では、**ctxmonitord** は VDA マシンを再起動した後で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。
- 環境内のすべての仮想マシン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。
- 提供された URL からダウンロードするには、ターゲット仮想マシンに `curl` をインストールする必要があります。

- 更新とロールバックの両方が失敗すると、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における Linux VDA のバージョンが異なる可能性があります。このケースは最適なものではありません。
- 通常、更新は完了するまでに数分かかります。Citrix Studio には状態インジケータはありません。

## Linux VM および Linux セッションのメトリック

August 20, 2024

次の表に、Linux VM および Linux セッションで使用できるメトリックを示します。

メトリック	最小必要な VDA バージョン	説明	注釈
ログオン期間	2402	合計ログオン時間は、Linux VDA 2109 以降で利用できます。これは、ユーザーが Citrix Workspace アプリから接続してからセッションを使用できるようになるまでのログオンプロセスの所要時間です。 <b>HDX</b> 接続、認証、 <b>GPO</b> などのログオンプロセスの各フェーズにかかる時間は、Linux VDA 2402 以降で利用できます。ログオン期間のメトリックを表示するには、[傾向] ビューの [ログオンパフォーマンス] タブを選択します。	Citrix Director と Monitor の両方で使用できます。
ICA 遅延	2311	ICA 遅延とは、ネットワークが低速であるかどうかを示すネットワーク遅延のことです。このメトリックにアクセスするには、[セッション詳細] ビューを開きます。	Citrix Director と Monitor の両方で使用できます。

メトリック	最小必要な VDA バージョ ン	説明	注釈
ポリシー	2311	現在のセッションで有効なすべてのポリシーは、[セッション詳細] ビューの [ポリシー] タブに表示されます。	Citrix Director と Monitor の両方で使用できます。
セッションの自動再接続回数	2109	セッションにおける自動再接続の数を表示するには、[傾向] 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回数] 列はセッション内で自動的に再接続を行う回数を表します。自動再接続は、[セッション画面の保持] ポリシーまたは [クライアントの自動再接続] ポリシーが有効な場合に有効になります。セッションの再接続と関連ポリシーについて詳しくは、次の記事を参照してください。 <a href="#">セッション</a> <a href="#">クライアントの自動再接続のポリシー設定</a> <a href="#">セッション画面の保持のポリシー設定</a>	Citrix Director と Monitor の両方で使用できます。
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 の両方で使用できます。

メトリック	最小必要な VDA バージョン	説明	注釈
Linux 仮想マシンのメトリック	2103	Linux VM のメトリック: CPU コアの数、メモリサイズ、ハードディスク容量、および現在および過去の CPU とメモリの使用率	Citrix Director と Monitor の両方で使用できます。
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、[セッション詳細] ビューに UDP または TCP として表示されます。	Citrix Director と Monitor の両方で使用できます。
ICA 往復時間	1903	ICA 往復時間 (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 のメトリックを取得するには、Citrix Studio で [ICA 往復測定] および [ICA 往復測定間隔] ポリシーを作成します。	Citrix Director と Monitor の両方で使用できます。

## ログ収集

August 20, 2024

### ログ記録メカニズムの概要

次の表は、Linux VDA のログ記録メカニズムの概要を示しています。

ログモジュール	ログファイル名の形式	ログファイル名の例	ログスコープ	単一ファイルの最大サイズ	ローテーションのしきい値 (デフォルト)	構成
HDX	hdx.log[.n]	hdx.log、 hdx.log.1	グラフィック、ログイン、オーディオ、キーボード、マウス	200MB	1 (current)、 2 (old)	setlog ユーティリティで構成可能
Jproxy	jproxy.log[.n.log]	jproxy.log、 jproxy.log.1.log	VDA 登録、ユーザー認証	20MB	1 (current)、 10 (old)	setlog ユーティリティまたは /etc/xdl/log4j2.xml で構成可能
VDA	vda.YYYY-MM-DD.hh.mm.ss.l0g	vda.2024-05-06.20.18.40.log	VDA 登録	50MiB	1 (current)、 1 (old)	setlog ユーティリティまたは /etc/xdl/brokeragent.c で構成可能

## 注:

- 最初のログファイルの名前には番号が含まれず、以降のファイルには「.n」または「n」という番号が付けられ、これがファイル番号を表します。たとえば、「hdx.log」は最初の HDX ログファイルであり、「hdx.log.1」は 2 番目の HDX ログファイルです。
- 単一ログファイルの最大サイズは、メガバイト (MiB) 単位で測定されます。
- 生成中であり、単一ファイルの最大サイズにまだ達していないログファイルは、「current」のログファイルと呼ばれます。「current」のログファイルが単一ファイルの最大サイズに達すると、ロールオーバーされて「old」ログファイルになります。
- ローテーションのしきい値は、保持できる「old」ログファイルの数を制限するように構成できます。この制限に達すると、最も古いログファイルが削除されます。

## ログ構成

このセクションでは、上記の表で概説した詳細を補足するために、ログ構成に関する追加情報を提供します。

**Linux VDA** ではデフォルトでログが有効

Linux VDA パッケージに、ctxlogd デーモンおよび setlog ユーティリティが追加されました。ctxlogd デーモンは、Linux VDA をインストールして構成すると、デフォルトで開始されます。トレースされた他のサービスはすべて

ctxlogd デーモンに依存しています。Linux VDA をトレースしない場合は、ctxlogd デーモンを停止できます。

### **/etc/xdl/brokeragent.conf** で VDA ログを構成する

注:

単一のログファイルの最大サイズなど、他のログパラメーターについて詳しく確認する必要がなく、VDA ログのログレベルのみを構成する場合は、この記事の後半で説明する setlog ユーティリティを使用できます。それ以外の場合は、/etc/xdl/brokeragent.conf を使用します。

VDA の/etc/xdl/brokeragent.conf ファイルは、VDA ログの構成に利用できます。例:

```
<!-- VDA trace settings -->
<add key="TraceProvider.Type" value="QueuedLogFileProvider"/>
<add key="TraceProvider.BaseFilename" value="/var/log/xdl/vda.log"/>
<add key="TraceProvider.AppendMode" value="true"/>
<add key="TraceProvider.SizeLimit" value="50"/>
<add key="TraceProvider.TrailCount" value="2"/>
<add key="TraceProvider.ProcessSafe" value="true"/>
<add key="TraceProvider.CategoryFilter" value="All" />
```

### **/etc/xdl/log4j2.xml** で Jproxy ログを構成する

注:

単一のログファイルの最大サイズなど、他のログパラメーターについて詳しく確認する必要がなく、Jproxy ログのログレベルのみを構成する場合は、この記事の後半で説明する setlog ユーティリティを使用できます。それ以外の場合は、/etc/xdl/log4j2.xml を使用します。

/etc/xdl/log4j2.xml で Jproxy ログを構成する場合の例です。**SizeBasedTriggeringPolicy** パラメーターは、単一の Jproxy ログファイルの最大サイズを指定し、**DefaultRollerStrategy** パラメーターは、保持できる Jproxy ログファイルの合計数を設定します。

```

<!-- Logging Properties -->
<Properties>
  <Property name="LOG_PATTERN">%d{yyyy-MM-dd HH:mm:ss.SSS}{GMT} [%-5p] [%tid] - %m%n</Property>
  <Property name="logfile">/var/log/xdl/jproxy.log</Property>
  <Property name="perfflogfile">/var/log/xdl/jproxyperf.log</Property>
</Properties>

<Appenders>

  <Console name="CONSOLE" target="SYSTEM_OUT">
    <PatternLayout pattern="${LOG_PATTERN}"/>
  </Console>

  <RollingFile name="logfile" fileName="${sys:logfile}"
    filePattern="${sys:logfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
      <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>

  <RollingFile name="perfflogfile" fileName="${sys:perfflogfile}"
    filePattern="${sys:perfflogfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
      <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>

```

### setlog ユーティリティで HDX ログを構成する

setlog ユーティリティは、/opt/Citrix/VDA/bin/パスにあります。このユーティリティを実行する権限があるのは、ルートユーザーのみです。GUI を使用するかコマンドを実行して、値（ログファイルのパス、単一ファイルのサイズ制限、ローテーションのしきい値）やログレベルなどの構成オプションを表示および変更できます。setlog ユーティリティのヘルプを表示するには、次のコマンドを実行します：

```
1 setlog help
```

値 デフォルトでは、HDX ログは/var/log/xdl/hdx.log に保存され、単一の HDX ログファイルのサイズ制限は 200MiB です。/var/log/xdl/hdx.log には、最大 2 つの「old」HDX ログファイルを保存できます。

現在のすべての setlog 値を表示するには、次のコマンドを実行します：

```

1 setlog values
2
3 log_path (Log Output Path) = /var/log/xdl/hdx.log
4
5 log_size (Max Log Size (MiB)) = 200
6
7 log_count (Max Old Log Files) = 2

```

単一の setlog 値を表示または設定するには、次のコマンドを実行します：

```
1 setlog value <name> [<value>]
```

例：

```
1 setlog value log_size 100
```

レベル デフォルトでは、ログレベルは **info**（大文字と小文字を区別しない）に設定されています。

トラブルシューティングが必要な問題が発生した場合、ほとんどのシナリオではログレベルを **verbose** に設定することをお勧めします。その後、問題を再現し、ログを収集します。

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

```
1 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	none
継承	inherit
トレース	trace
詳細	verbose
情報	info
警告	warning
エラー	error
致命的なエラー	fatal

**<class>**変数は、Linux VDA の 1 つのコンポーネントを指定します。すべてのコンポーネントをカバーするには、**all** に設定します。例：

```
1 setlog level all error
```

サポートされているすべてのクラスまたはコンポーネントを表示するには、次のコマンドを実行します：

```
1 setlog levels
```

デフォルトに戻す すべてのレベルおよび値をデフォルト設定に戻します：

```
1 setlog default
```

重要：

**ctxlogd** サービスは **/var/xdl/ctxlog** ファイルを使用して構成されます。このファイルは、ルートユーザーのみが作成できます。他のユーザーは、このファイルへの書き込み権限がありません。ルートユーザーは、他の



ユーザーに書き込み権限を許可しないことをお勧めします。許可すると、**ctxlogd** が恣意的に、または悪意をもって構成される危険性があります。これによってサーバーのパフォーマンスが影響を受け、ユーザーエクスペリエンスにも影響を与える可能性があります。

### ログ収集

**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。ログの収集に使用される **xdlcollect** Bash スクリプトは Linux VDA ソフトウェアに統合され、**/opt/Citrix/VDA/bin** に配置されます。

ログ収集が完了すると、VDA の **/tmp/xdlcollect** に ZIP ファイルが生成されます。

### トラブルシューティング

**/var/xdl.ctxlog** ファイルがない場合（過失による削除など）、**ctxlogd** デーモンが失敗し、**ctxlogd** サービスを再起動できません。

`/var/log/messages:`

```
1 Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
  configuration file.
2
3 Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited, code
  =exited, status=1/FAILURE
4
5 Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.
6
7 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** ファイルを再度作成します。次に、他のサービスが依存する **ctxlogd** サービスを再起動します。

### セッションのシャドウ

August 20, 2024

セッションのシャドウにより、ドメイン管理者はイントラネット内のユーザーの ICA セッションを閲覧できます。この機能では、noVNC を使用して ICA セッションに接続します。

注:

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

## インストールと構成

### 依存関係

セッションのシャドウには、`python-websockify`と`x11vnc`という、2つの新しい依存関係が必要です。Linux VDA をインストールした後、`python-websockify`と`x11vnc`を手動でインストールします。

#### 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:

```
1 sudo pip3 install websockify
2 sudo yum install x11vnc
```

#### RHEL 9.4/9.3/9.2/8.x および Rocky Linux 9.4/9.3/9.2/8.x 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。

```
1 sudo pip3 install websockify
2 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:

```
1 dnf install -y --nogpgcheck https://dl.fedoraproject.org/pub/epel/epel-
  release-latest-8.noarch.rpm
2
3 subscription-manager repos --enable "codeready-builder -for-rhel-8-
  x86_64-rpms"
```

#### Ubuntu の場合:

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

```
1 sudo pip3 install websockify
2 sudo apt-get install x11vnc
```

#### SUSE の場合:

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

```
1 sudo pip3 install websockify
2 sudo zypper install x11vnc
```

#### Debian 12.5 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:

```
1 apt install python3-websockify
2 sudo apt-get install x11vnc
```

**Debian 11** 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:

```
1 sudo pip3 install websockify
2 sudo apt-get install x11vnc
```

## ポート

セッションのシャドウ機能は、Linux VDA から Citrix Director への接続を構築するために、6001~6099 の範囲内で使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 に制限されています。要件を満たすために、特にマルチセッションのシャドウ用に十分なポートがあることを確認してください。

## レジストリ

次の表は、関連するレジストリの一覧です:

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションのシャドウ機能を有効または無効にします。	1 (有効)
ShadowingUseSSL	Linux VDA と Citrix Director 間の接続を暗号化するかどうかを決定します。	0 (無効)

Linux VDA で `ctxreg` コマンドを実行して、レジストリ値を変更します。たとえば、セッションシャドウを無効にするには、次のコマンドを実行します:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d 0x00000000
```

**SSL**

Linux VDA と Citrix Director 間の noVNC 接続では、WebSocket プロトコルが使用されます。セッションのシャドウの場合、`ws://`と`wss://`のどちらが選択されるかは、前述の「ShadowingUseSSL」レジストリによって決まります。デフォルトでは、`ws://`が選択されています。ただし、セキュリティ上の理由から、`wss://`を使用

して、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。  
`ws://`を使用した Linux VDA セッションのシャドウについては、Citrix はセキュリティ上のいかなる責任も負いません。

サーバー証明書とルート **SSL** 証明書を取得する 証明書には、信頼された証明機関 (CA) による署名が必要です。

Linux VDA サーバーで SSL を設定する場合は、サーバーごとに個別のサーバー証明書 (キーを含む) が必要です。また、サーバー証明書によって各コンピューターが識別されるため、各サーバーの完全修飾ドメイン名 (FQDN) を調べる必要があります。代わりにドメイン全体にワイルドカード証明書を使用できます。この場合、少なくともドメイン名を知っておく必要があります。

Linux VDA と通信する Citrix Director クライアントごとにルート証明書も必要です。ルート証明書は、サーバー証明書と同じ証明機関から入手できます。

次の CA からサーバー証明書とクライアント証明書をインストールできます：

- オペレーティングシステムにバンドルされている CA
- エンタープライズ CA (組織がアクセス可能にする 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手段を取るべきかについては、社内のセキュリティ担当部門に問い合わせてください。

**重要：**

- サーバー証明書の共通名は、Linux VDA の正確な FQDN、または少なくともワイルドカードとドメイン文字を正しく組み合わせたものである必要があります。たとえば、`vda1.basedomain.com` や `*.basedomain.com` などです。
- SHA1 や MD5 などのハッシュアルゴリズムは、一部のブラウザでサポートされるデジタル証明書の署名には弱すぎます。したがって、SHA-256 が最低基準として指定されています。

各 **Citrix Director** クライアントにルート証明書をインストールする セッションのシャドウと IIS で、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。証明機関から証明書を取得したら、IIS のサーバー証明書ウィザードを再び起動します。この操作により、自動的に証明書がインポートされます。または、Microsoft 管理コンソールの証明書スナップインで証明書を表示して、サーバーにインストールすることもできます。Internet Explorer と Google Chrome は、デフォルトで、オペレーティングシステムにインストールされている証明書をインポートします。Mozilla Firefox の場合、証明書マネージャーの [認証局証明書] タブでルート CA 証明書をインポートする必要があります。

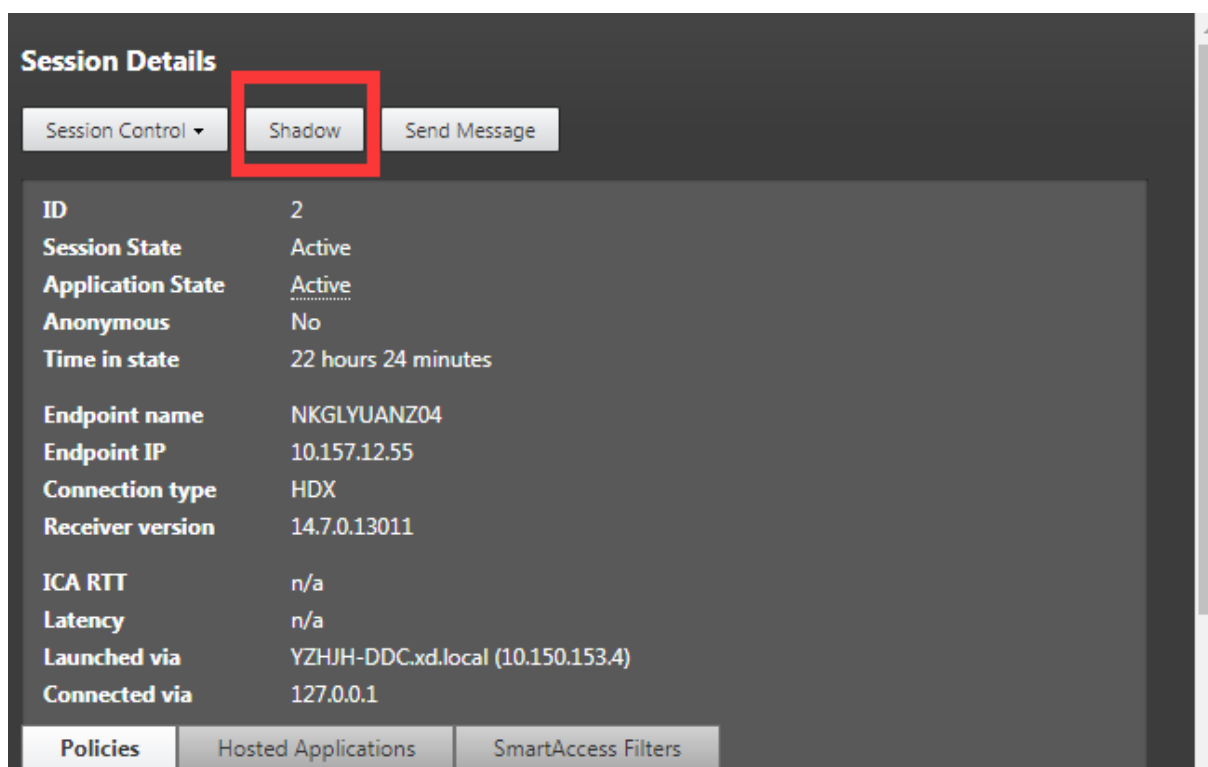
各 **Linux VDA** サーバーにサーバー証明書とそのキーをインストールする サーバー証明書に「`shadowingcert.*`」、キーファイルに「`shadowingkey.*`」と名前を指定します (\* は、`shadowingcert.pem` や `shadowingkey.key` のような形式となることを示す)。サーバー証明書とキーファイルを、パス `/etc/xdm/shadowingssl` の下に置き、制

限付きの権限で適切に保護し、ctxsrvr のみに読み取りアクセスを許可します。間違った名前やパスを使用すると、Linux VDA は特定の証明書やキーファイルを見つけることができなくなり、Citrix Directorとの接続に失敗することがあります。コマンドは次のとおりです：

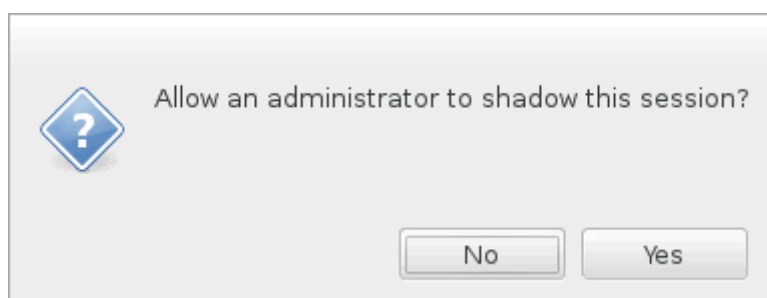
```
1 cp <vda's-public-key> /etc/xdl/shadowingssl/shadowingcert.pem
2 cp <vda's-server-private-key> /etc/xdl/shadowingssl/shadowingkey.key
3 sudo chown ctsxrvr:ctxadm /etc/xdl/shadowingssl/shadowingcert.pem
4 sudo chown ctsxrvr:ctxadm /etc/xdl/shadowingssl/shadowingkey.key
```

## 使用状況

Citrix Directorからターゲットのセッションを見つけ、[セッション詳細] ビューで [シャドウ] をクリックして、シャドウの要求を Linux VDA に送信します。



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) に確認メッセージが表示され、セッションをシャドウする許可がユーザーに求められます。



ユーザーが [はい] をクリックすると、ICA セッションがシャドウされていることを示すウィンドウが **Citrix Director** 側で開きます。

使用方法について詳しくは、[Citrix Director のドキュメント](#)を参照してください。

#### 制限事項

- セッションのシャドウは、イントラネットでのみ使用するよう設計されています。Citrix Gateway を介して接続する場合でも、外部ネットワークでは機能しません。外部ネットワークでの Linux VDA セッションのシャドウについては、Citrix はいかなる責任も負いません。
- セッションのシャドウを有効にすると、ドメイン管理者は ICA セッションのみを表示できますが、書き込みの権限や制御する権限はありません。
- 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メッセージが表示されます。セッションユーザーが許可を与えた場合にのみ、セッションをシャドウできます。
- 前述の確認メッセージには、20 秒のタイムアウト制限があります。タイムアウトになると、シャドウの要求は失敗します。
- 1 つのセッションは、1 人の管理者だけがシャドウできます。たとえば、セッション管理者 A がシャドウしている場合に、管理者 B がシャドウ要求を送信すると、ユーザーの許可を取得するための確認がユーザーデバイスに再度表示されます。ユーザーが同意すると、管理者 A のシャドウ接続は停止され、管理者 B に対して新しいシャドウ接続が構築されます。ある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すると、また新しいシャドウ接続を構築できます。
- セッションのシャドウを使用するには、**Citrix Director 7.16** 以降をインストールしてください。
- **Citrix Director** クライアントは、IP アドレスではなく FQDN を使用して、ターゲットの Linux VDA サーバーに接続します。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

セッションの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** クライアントの場合

Web ブラウザーの開発ツールを使用して、[コンソール] タブの出力ログを確認します。または、[ネットワーク] タブで ShadowLinuxSession API の応答を確認します。ユーザー権限を取得するための確認が表示されても接続が確立されない場合は、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://**接続で問題が発生した場合は、証明書を確認してください。

## Linux VDA の場合

シャドウ要求に応答して、ユーザーの許可を取得するための確認が表示されることを確認します。表示されない場合は、vda.log ファイルと hdx.log ファイルを調べてください。vda.log ファイルを取得するには、次の操作を実行します：

1. /etc/xdl/ctx-vda.conf ファイルを見つけます。vda.log の構成を有効にするには、次の行のコメントを外します：

```
Log4jConfig="/etc/xdl/log4j.xml"
```

2. /etc/xdl/log4j.xml を開き、com.citrix.dmc の部分を見つけ、次のように「info」を「trace」に変更します：

```
1  <!-- Broker Agent Plugin - Director VDA plugin Logger -->
2
3  <logger name="com.citrix.dmc">
4
5      <level value="trace"/>
6
7  </logger>
```

3. `systemctl restart ctxvda` コマンドを実行して、ctxvda サービスを再起動します。

接続確立中にエラーが発生した場合は、次の操作を実行してください：

1. セッションのシャドウがポートを開くのを止めるファイアウォール制限がないか確認します。
2. SSL シナリオの場合、証明書とキーファイルの名前が正しく指定され、正しいパスに置かれていることを確認します。
3. 新しいシャドウ要求で使用するための十分なポートが、6001~6099 の間に残っていることを確認します。

## 監視サービスデーモン

August 20, 2024

モニターサービス **ctxmonitord** は、Linux VDA のヘルス状態を監視するデーモンです。これは、3つのモジュールで構成されています：

- キー プロセスモニターモジュール：このモジュールは、予期しない障害が発生したときに主要なサービスを自動的に再起動し、対象プロセスの状態を定期的にスキャンして記録し、Xorg の残留物を適時にクリーンアップするように Systemd に委任します。ログは **/var/log/xdl/ms.log** に保存されます。
- XDPing 統合モジュール：このモジュールは、XDPing 分析およびバックアップタスクを定期的に行います。これらのタスクからの出力はすべて、**/var/log/xdl/msxdping.log** にあります。
- Linux VDA 自己更新モジュール：詳しくは、「[Azure を使用した Linux VDA の自動更新](#)」を参照してください。

## 構成

デフォルトでは、VDA を起動すると、監視サービス **ctxmonitord** が自動的に起動します。管理者権限があれば、`/etc/xdl/ctxmonitord.conf` および `/etc/xdl/whitelist.conf` でこれを構成できます。

### ctxmonitord.conf

この構成ファイルは、監視サービス `ctxmonitord` とそのモジュールの動作を指定します。デフォルトでは、次のように構成されています：

```
1      ; This is the configuration file for ctxmonitord service
2
3
4  ; Section 'service' configures the key process monitor module
5  [service]
6  MonitorEnable=true      ; true or false
7  DetectInterval=300      ; in seconds, minimum is 60
8
9  ; Section 'xdping' configures the XDPing integration module
10 [xdping]
11 XdpingEnable=true      ; true or false
12 AnalysisInterval=600    ; in seconds, minimum is 60, 0 means disable
13 BackupInterval=1       ; in days, 0 means disable
14
15 ; Section 'rules' configures the rules about how to monitor each key
16   process
17 ; Each rule should be named as rules.<ProcessName> where <ProcessName>
18   is the name of the process to be monitored
19 ; 'ProcessName' should be the name of the target process that is picked
20   up from whitelist.conf
21 ; 'MonitorType' should be 'HealthCheck' or 'ResidueCleanup'. All '
22   MonitorType' should be 'HealthCheck' except for 'Xorg'
23 [rules]
24 [rules.ctxhdx]
25 ProcessName=ctxhdx
26 MonitorType=HealthCheck
27
28 [rules.ctxvda]
29 ProcessName=ctxvda
30 MonitorType=HealthCheck
31
32 [rules.ctxjproxy]
33 ProcessName=ctxjproxy
34 MonitorType=HealthCheck
35
36 [rules.ctxpolicyd]
37 ProcessName=ctxpolicyd
38 MonitorType=HealthCheck
39
40 [rules.ctxlogd]
41 ProcessName=ctxlogd
```



```
38 MonitorType=HealthCheck
39
40 [rules.xorg]
41 ProcessName=Xorg
42 MonitorType=ResidueCleanup
```

## whitelist.conf

この構成ファイルでは、監視対象のターゲットプロセスの許可リストを指定します。ctxmonitord.conf ファイルで指定したターゲットサービスは、whitelist.conf ファイルにも表示されている必要があります。デフォルトでは、次のように構成されています:

```
1 ctxhdx
2 ctxvda
3 ctxjproxy
4 ctxlogd
5 ctxcdm
6 ctxcups
7 ctxpolicyd
8 ctxusbsd
9 ctxceip
10 ctxsdcd
11 ctxrunatboot
12 ctxgdt
13 ctxbcrd
14 ctxfidod
15 ctxpfwd
16 ctxwcamsd
17 ctxcertmgr
18 ctxscardsd # Except for SLES
19 ctxfedsd # Only for RHEL 8/9
20 Xorg
```

注:

- ctxmonitord.conf および whitelist.conf ファイルへの変更を適用するには、**systemctl restart ctxmonitord** コマンドを実行して監視サービスデーモンを再起動します。
- 予期しない障害が発生した場合にサービスを再起動する方法を構成するには、**/usr/lib/systemd/system/ctx\*.service** ファイルを編集します。たとえば、デフォルトの再起動オプションは次のとおりです:

```
1 Restart=on-failure
2 RestartSec=5
3 StartLimitInterval=60
4 StartLimitBurst=3
```

## トラブルシューティング

August 20, 2024

この記事では、**XDPing** を使用してトラブルシューティングを行う方法と、**ctxsdcutil** ユーティリティを使用してセッションメトリックを照会する方法について説明します。

### XDPing

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 環境での一般的な構成の問題をチェックする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。

このコマンドでは、**XDPing** に必要な **Python3** 仮想環境も作成されます。このコマンドで **Python 3** 仮想環境の作成に失敗した場合は、「[Python 3 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 ツールの使用時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** という名前の単一の実行可能ファイルが付属しています。

次の表では、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ます:

---

タスク	XDPing コマンド	注釈
コマンドラインオプションを表示するには	<code>sudo /opt/Citrix/VDA/bin/xdping -h</code>	-

---

タスク	XDPing コマンド	注釈
完全なテスト一式を実行するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping</b> (コマンドラインオプションなしで <b>XDPing</b> を実行する)	Linux <b>XDPing</b> 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。詳しくは、この記事の「個別のテスト」を参照してください。
VDA 登録状態の確認を実行するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -a</b>	詳しくは、この記事で後述する「登録状態の確認の範囲」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	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -b</b>	詳しくは、この記事で後述する「VDA データのバックアップと比較」を参照してください。
VDA のバックアップデータに関して最新の 2 つのコピーを比較するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -diff</b>	詳しくは、この記事で後述する「VDA データのバックアップと比較」を参照してください。
VDA のバックアップデータに関する特定の 2 つのコピーを比較するには	**sudo <b>/opt/Citrix/VDA/bin/xdping -diff=&lt; 特定のバックアップデータのディレクトリ &gt;:&lt; 別のバックアップデータのディレクトリ &gt; \* \*</b> 別のバックアップデータのディレクトリ > 特定のバックアップデータのディレクトリ > 別のバックアップデータのディレクトリ > 特定のバックアップデータのディレクトリ >	詳しくは、この記事で後述する「VDA データのバックアップと比較」を参照してください。
Linux VDA パッケージをインストールする前に環境を確認するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -preflight</b>	-
時刻テストや Kerberos やデータベーステストなど、特定のテストカテゴリのみを実行するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -T time,kerberos,database</b>	-
特定の Delivery Controller をプロンプトするには	**sudo <b>/opt/Citrix/VDA/bin/xdping -d \* \*</b>	-
VDA にインストールされている依存関係を表示するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -query-pkgs</b> or <b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -q</b>	-

タスク	XDPing コマンド	注釈
VDA にインストールされている依存関係を表示し、クエリ結果を特定のパスに保存するには	<pre> **sudo /opt/Citrix/VDA/bin/xdping -f &lt; クエリ結果を保存するための特定の パス &gt;\* \* または \* \* sudo /opt/Citrix/VDA/bin/xdping --query-pkgs-to &lt; クエリ結果を保 存するための特定のパス &gt;\* \* クエ リ結果を保存するための特定のパス &gt; クエリ結果を保存するための特定 のパス &gt; クエリ結果を保存するた めの特定のパス &gt; クエリ結果を保存す るための特定のパス &gt; </pre>	-

個別のテスト 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。これらのテストは、大きく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どうかを確認します。
- Linux ディストリビューションを含むマシン情報を識別して表示します。
- 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識別し、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使用可能なディスク容量を確認します。
- マシンのホストとドメイン名の構成を確認します。
- DNS 構成を確認し、参照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：
  - XenServer (旧称 Citrix Hypervisor)
  - Microsoft HyperV
  - VMware vSphere
- 時刻設定を確認し、ネットワークの時刻同期が機能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いて、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の LDAP 検索環境を確認します。
- Active Directory 統合が正しくセットアップされ、現在のマシンがドメインに参加しているかどうかを確認します。サポート対象：

- Samba Winbind
  - Dell Quest Authentication Services
  - Centrify DirectControl
  - SSSD
- Active Directory 内の Linux コンピューターオブジェクトの整合性を確認します。
  - Pluggable Authentication Module (PAM) 構成を確認します。
  - コアダンプの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を確認します。
  - Linux VDA パッケージを特定し、インストールの整合性を確認します。
  - PostgreSQL レジストリデータベースの整合性を確認します。
  - Linux VDA サービスが適切に構成され動作しているかを確認します。
  - VDA および HDX 構成の整合性を確認します。
  - 構成済みの各 Delivery Controller をプローブして、ブローカーサービスが到達可能、操作可能で、応答性があることをテストします。
  - マシンが Delivery Controller ファームに登録されているかを確認します。
  - アクティブまたは切断された各 HDX セッションの状態を確認します。
  - Linux VDA 関連のエラーと警告についてログファイルをスキャンします。
  - Xorg のバージョンが適切かを確認します。
  - 必要な依存関係がインストールされているかを確認します。

出力例 以下は、Kerberos テストを実行した場合の出力例です：

```
sudo xdping -T kerberos
```

```
Root User -----  
User:      root  
EUID:      0  
Verify user is root [Pass]  
  
Kerberos -----  
Kerberos version: 5  
Verify Kerberos available [Pass]  
Verify Kerberos version 5 [Pass]  
KRB5CCNAME: [Not set]  
             Distro default FILE:/tmp/krb5cc_%{uid}  
KRB5CCNAME type: [Supported]  
KRB5CCNAME format: [Default]  
Verify KRB5CCNAME cache type [Pass]  
Verify KRB5CCNAME format [Pass]  
Configuration file: /etc/krb5.conf [Exists]
```

```

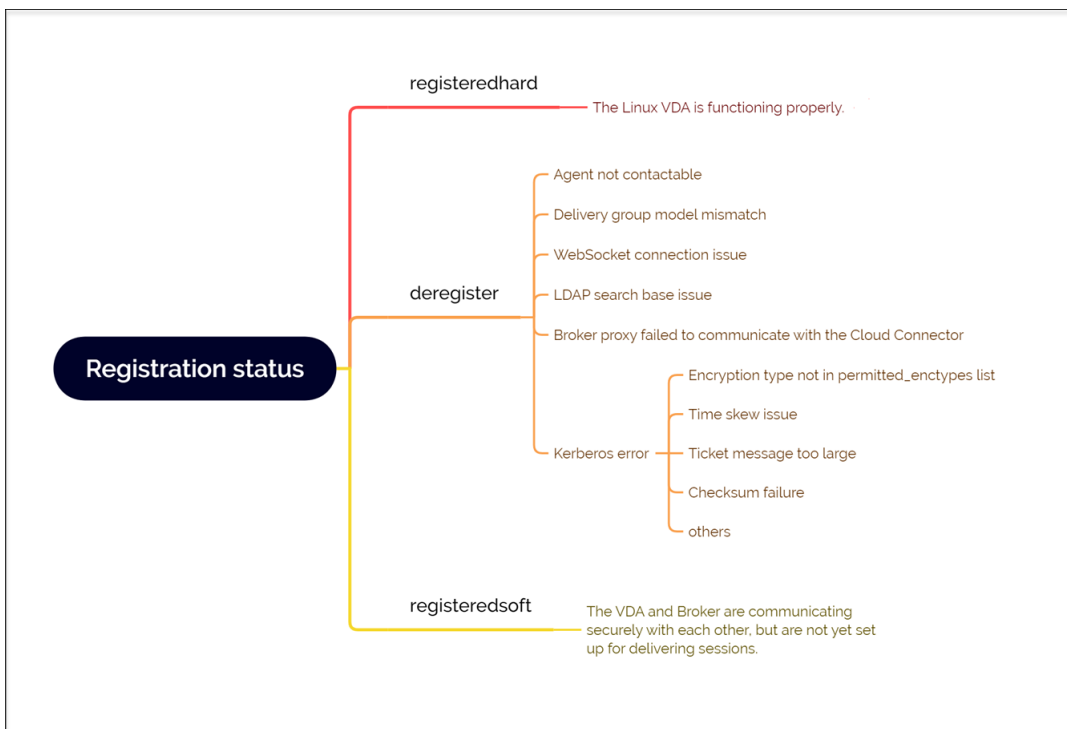
Verify Kerberos configuration file found [Pass]
Keytab file: /etc/krb5.keytab [Exists]
Default realm: XD2.LOCAL
Default realm KDCs: [NONE SPECIFIED]
Default realm domains: [NONE SPECIFIED]
DNS lookup realm: [Enabled]
DNS lookup KDC: [Enabled]
Weak crypto: [Disabled]
Clock skew limit: 300 s
  Verify system keytab file exists [Pass]
  Verify default realm set [Pass]
  Verify default realm in upper-case [Pass]
  Verify default realm not EXAMPLE.COM [Pass]
  Verify default realm domain mappings [Pass]
  Verify default realm master KDC configured [Pass]
  Verify Kerberos weak crypto disabled [Pass]
  Verify Kerberos clock skew setting [Pass]
Default ccache: [Not set]
      Distro default FILE:/tmp/krb5cc_%{uid}
Default ccache type: [Supported]
Default ccache format: [Default]
  Verify default credential cache cache type [Pass]
  Verify default credential cache format [Pass]
UPN system key [MYVDA1$@██████████]: [MISSING]
SPN system key [host/██████████@██████████]: [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$@██████████
      while getting initial credentials
  Verify KDC authentication [ERROR]
Failed to authenticate and obtain a Ticket Granting Ticket (TGT)
from the KDC authentication service for the machine account UPN
MYVDA1$@██████████. 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 登録状態の確認の範囲** Linux **XDPing** ツールには、VDA 登録状態を確認および分析するための役立つ分析モジュールも用意されています。登録状態の確認の範囲については、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** ツールには VDA バックアップモジュールが導入されています。このモジュールを使用すると、構成、データベース、バイナリの権限データなど、VDA の主要データをいつでもバックアップできます。VDA が正常に実行されている場合は、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、データの 2 つのコピーを比較して、トラブルシューティングを容易にします。次の表では、VDA データのバックアップと、対応する **XDPing** コマンドとの比較について説明します。

タスク	XDPing コマンド	注釈
VDA の主要データをバックアップするには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -b</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、 <b>/var/ctxbackup</b> のディレクトリに保存されます。バックアップデータのディレクトリには、現在の日付と時刻の名前が <b>yyyy-mm-dd-hh_mm_ss</b> 形式で付けられます。例： <b>2023-02-27-16_31_27</b> 。デフォルトでは、バックアップデータのディレクトリの最大数は 30 で、この数を超えると、 <b>XDPing</b> ツールは古いバックアップデータのディレクトリをローテーションまたは削除します。ディレクトリの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。 <code>sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Backup"-t "REG_DWORD"-v "MaxDirRotationCount"-d "0x00000005"--force</code>
VDA のバックアップデータに関して最新の 2 つのコピーを比較するには	<b>sudo</b> <b>/opt/Citrix/VDA/bin/xdping -diff</b>	-



タスク	XDPing コマンド	注釈
VDA のバックアップデータに関する特定の 2 つのコピーを比較するには	<pre>**sudo /opt/Citrix/VDA/bin/xdping -diff=&lt; 特定のバックアップデータのディレクトリ &gt;:&lt; 別のバックアップデータのディレクトリ &gt; \* \* 別のバックアップデータのディレクトリ &gt; 特定のバックアップデータのディレクトリ &gt; 別のバックアップデータのディレクトリ &gt; 特定のバックアップデータのディレクトリ &gt;</pre>	-

セッションメトリックのクエリユーティリティ

#### ctxsession

このユーティリティは、Windows ユーザーエクスペリエンスを提供します。このユーティリティを使用すると、次のスクリーンショットに示すようにセッションメトリックにアクセスできます：

```
root@u22:~# /opt/Citrix/VDA/bin/ctxsession -v

Session Id:      4
Protocols:      TCP-TLS-CGP-ICA
User Name:      st\wcx
Client Name:    BLR3-EWS-WPD030
Client Address: 10.109.143.143:58535
Local Address:  10.158.180.61:443
Remote Address: 10.109.143.143:58535
Security Protocol: TLSv1.2
Security Cipher: ECDHE-RSA-AES256-GCM-SHA384
Cipher Strength: 256 bits
ICA Encryption: Basic
Rendezvous Version: None
Reducer Version: 4

ICA Statistics:
ICA RTT: 0 ms
Sent Bandwidth: 112 bps
Received Bandwidth: 0 bps
```

## ctxqsession、ctxquser、ctxqfull、ctxquery

これらのユーティリティの使用方法については、help コマンドを実行してください。例:

```
root@ubuntu2204-33:~# /opt/Citrix/VDA/bin/ctxqsession --help
Usage: ctxqsession [-f short_format_options | -o long_format_options] [-m] [user user_name]
Display information about ICA connections to the local server.

Options:
  -f          Change the displayed information specified by short format options
  -o          Change the displayed information specified by long format options
  -m          Alter the column headers to remove spaces
  user       Display information about the specified user

Short format options:
  d  Client device name (client name)
  i  Session ID
  n  Session name
  p  Published application name
  P  Protocols
  S  Session state
  u  User name
  x  X display number

Long format options(comma separated):
  app  Published application name
  dev  Client device name (client name)
  id   Session ID
  proto Protocols
  sess Session name
  state Session state
  user  User name
  xdpi  X display number
```

ユーザーは自分のセッションメトリックのみを照会できます。他のユーザーのセッションメトリックを照会する権限を持つのは、ルートユーザーおよび **ctxadm** グループのユーザーのみです。

## ctxsdcutil

このユーティリティによって、VDA でホストされているすべてのセッションや特定のセッションについて以下のメトリックを照会できます。このためには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]` コマンドを実行します。[-c] 引数は、1 秒おきにメトリックを照会することを意味します。

- セッション入力帯域幅
- セッション出力帯域幅
- セッション出力速度
- 遅延 - 最新記録
- 往復時間
- **ThinWire** 出力帯域幅
- オーディオ出力帯域幅

- プリンター出力帯域幅
- ドライブ入力帯域幅
- ドライブ出力帯域幅

## その他

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [HTML5 向け Citrix Workspace アプリのサポート](#)
- [Python 3 仮想環境の作成](#)
- [NIS の Active Directory との統合](#)
- [IPv6](#)
- [LDAPS](#)
- [Xauthority](#)

## HTML5 向け Citrix Workspace アプリのサポート

July 16, 2024

HTML5 向け Citrix Workspace アプリを使用して、クライアントを Citrix Gateway に接続することなく Linux 仮想アプリおよびデスクトップに直接接続できます。HTML5 向け Citrix Workspace アプリについて詳しくは、[Citrix ドキュメント](#)を参照してください。

### この機能を有効にする

この機能はデフォルトでは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 HTML5 向け Citrix Workspace アプリを有効にします。  
詳細な手順については、Knowledge Center 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** 接続ポリシーを [許可] に設定します。

他の WebSocket ポリシーを設定することもできます。WebSocket ポリシーの完全な一覧については、「[WebSocket のポリシー設定](#)」を参照してください。

- b) VDA で `ctxvda` サービス、`ctxhdx` サービスの順に再起動して設定を有効にします。

- c) VDA で次のコマンドを実行して、WebSocket リスナーが動作しているかどうかを確認します。

```
netstat -an | grep 8008
```

WebSocket リスナーが動作している場合、コマンド出力は次のようになります：

```
tcp 0 0 :::8008 :::* LISTEN
```

注：セキュアな WebSocket 接続のために TLS 暗号化を有効に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、「[TLS によるユーザーセッションの保護](#)」を参照してください。

## Python 3 仮想環境の作成

August 20, 2024

ネットワークに接続している場合は、`sudo /opt/Citrix/VDA/bin/xdping` または `/opt/Citrix/VDA/sbin/enable_ldaps.sh` コマンドを実行して Python 3 仮想環境を作成できます。ただし、コマンドで Python 3 仮想環境を作成できない場合は、ネットワークに接続してなくても手動で作成できます。この記事では、ネットワークに接続せずに Python 3 仮想環境を作成するための前提条件と手順について詳しく説明します。

### 前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3` ディレクトリにアクセスするには、管理者権限が必要です。
- Python3 パッケージのホイールファイルが必要です。ホイールファイルは <https://pypi.org> からダウンロードできます。

### Python 3 仮想環境の作成

次の手順を実行して、Python 3 仮想環境を作成します：

1. Python 3 の依存関係をインストールします。

**Amazon Linux 2** の場合：

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

**RHEL** および **Rocky Linux** の場合:

```
1 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ためには、特定のリポジトリの有効化が必要な場合があります。RHEL 7 の場合、`subscription-manager repos --enable rhel-7-server-optional-rpms` コマンドを実行します。RHEL 8 の場合、`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:

```
1 apt-get -y install python3-dev python3-pip python3-venv libkrb5-dev
```

**SUSE** の場合:

```
1 zypper -n install lsb-release python3-devel python3-setuptools  
krb5-devel gcc libffi-devel libopenssl-devel
```

## 2. Python 3 仮想環境を作成します。

## 注:

pip ツールの使用時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
```

**Amazon Linux 2**、**Debian**、**RHEL**、**Rocky Linux**、**Ubuntu** の場合:

```
1 sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

**SUSE** の場合:

```
1 sudo ln -s /usr/lib/mit/bin/krb5-config /usr/bin/krb5-config  
2  
3 export PATH=$PATH:/usr/lib/mit/bin:/usr/lib/mit/sbin  
4  
5 sudo mkdir -p /usr/lib/mit/include/gssapi/  
6  
7 sudo ln -s /usr/include/gssapi/gssapi_ext.h/usr/lib/mit/include/  
gssapi/gssapi_ext.h  
8  
9 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。

```
1 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
  upgrade pip==21.3.1
2
3 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。

```
1 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
  upgrade pip==21.3.1
2
3 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
4
5 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /
  opt/Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

## NIS の Active Directory との統合

July 16, 2024

このトピックでは、SSSD を使用して、NIS を Linux VDA の Windows Active Directory (AD) と統合する方法について説明します。Linux VDA は、Citrix Virtual Apps and Desktops のコンポーネントと見なされます。そのため Linux VDA は、Windows AD 環境に密接に結びついています。

AD の代わりに NIS を UID および GID プロバイダーとして使用するには、AD と NIS でユーザー名とパスワードの組み合わせのアカウント情報を同一にする必要があります。

#### 注:

NIS を使用した場合も、認証は AD サーバーにより行われます。NIS+ はサポートされません。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 属性は使用されません。

#### ヒント:

これは、Linux VDA を展開する方法として廃止されているため、特定のユースケースでのみ使用してください。RHEL/CentOS ディストリビューションの場合は、「[Amazon Linux 2、CentOS、RHEL、および Rocky Linux への Linux VDA の手動インストール](#)」の手順に従います。Ubuntu ディストリビューションの場合は、「[Ubuntu への Linux VDA の手動インストール](#)」の手順に従います。

### SSSD とは?

SSSD はシステムデーモンです。SSSD 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する共通フレームワークを通じて、リモートリソースの識別および認証のアクセスを提供することです。PAM や NSS モジュールを提供しており、将来的には D-BUS ベースのインターフェイスもサポートして、拡張ユーザー情報に対応する予定です。また、ローカルユーザーアカウントと拡張ユーザー情報を保存するための優れたデータベースを提供します。

### NIS と AD の統合

NIS と AD を統合するには、次の手順を完了します:

手順 **1**: **Linux VDA** を **NIS** クライアントとして追加

NIS クライアントを構成します。

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

NIS ドメインを設定します。

```
1 ypdomainname nis.domain
2 echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

NIS サーバーとクライアントの IP アドレスを **/etc/hosts** に追加します:

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

**authconfig** で NIS を構成します:

```
1 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です。**server.nis.domain** は、NIS サーバーのホスト名であり、NIS サーバーの IP アドレスにもできます。

NIS のサービスを設定します:

```
1 sudo systemctl start rpcbind ypbind
2
3 sudo systemctl enable rpcbind ypbind
```

NIS の構成が正しいことを確認します:

```
1 ypwhich
```

NIS サーバーからアカウント情報が使用できることを確認します:

```
1 getent passwd nisaccount
```

注:

**nisaccount** は、NIS サーバーの実際の NIS アカウントです。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設定されていることを確認します。

手順 2: ドメインに参加し、**Samba** を使用してホストの **keytab** を作成

SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する AD のクライアント機能が提供されていません。この機能を取得するには次のような方法があります:

- `adcli`
- `realmd`
- `Winbind`
- `Samba`

このセクションでは、Samba によるアプローチについてのみ説明します。`realmd`については、RHEL または CentOS のベンダーのドキュメントを参照してください。SSSD を構成する前に、以下の手順に従う必要があります。

ドメインに参加し、**Samba** を使用してホストの **keytab** を作成する:

Linux クライアントで、適切に構成されたファイルを使用します:

- `/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:

```
1 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 領域名で、**domain** はドメインの NetBIOS 名です。

KDC サーバーおよび領域名を DNS ベースで参照する必要がある場合は、次の 2 つのオプションを前述のコマンドに追加します:

```
--enablekrb5kdcdns --enablekrb5realmdns
```

`/etc/samba/smb.conf` を開いて、**[Global]** セクションに次のエントリを追加します。ただし、追加するのは、**authconfig** ツールによって生成されたセクションの後です:

```
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できることと、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:

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



**REALM** は大文字の Kerberos 領域名で、**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。

### 手順 3: SSSD のセットアップ

#### 注

SSSD を Name Service Cache Daemon (NSCD) と併用すると、予期しない動作が発生する可能性があります。詳しくは、「[SSSD での NSCD の使用](#)」を参照してください。

SSSD のセットアップは、以下の手順で構成されています:

- Linux クライアントマシンに **sssd-ad** パッケージおよび **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に構成の変更を行う (**sssd.conf** など)。
- **sssd** サービスを開始します。

**/etc/sssd/sssd.conf** **sssd.conf** の設定の例 (必要に応じて追加の設定を行うことができます):

```
1 [sssd]
2 config_file_version = 2
3 domains = EXAMPLE
4 services = nss, pam
5
6 [domain/EXAMPLE]
7 # Uncomment if you need offline logins
8 # cache_credentials = true
9 re_expression = (((?P<domain>[^\]+)\(?P<name>.+))|((?P<name>[^\@]+)@(?P<
    <domain>.+))|(^(?P<name>[^\@]+)$))
10 id_provider = proxy
11 proxy_lib_name = nis
12 auth_provider = ad
13 access_provider = ad
14
15 # Should be specified as the long version of the Active Directory
    domain.
16 ad_domain = EXAMPLE.COM
17
18 # Kerberos settings
19 krb5_ccachedir = /tmp
20 krb5_ccname_template = FILE:%d/krb5cc_%U
21
22 # Uncomment if service discovery is not working
23 # ad_server = server.ad.example.com
24
25 # Comment out if the users have the shell and home dir set on the AD
    side
26 default_shell = /bin/bash
27 fallback_homedir = /home/%d/%u
28
```

```
29 # Uncomment and adjust if the default principal SHORTNAME$@REALM is not
    available
30 # ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

**ad.domain.com** と **server.ad.example.com** を対応する値で置き換えます。詳しくは、「[sssd-ad\(5\) - Linux man page](#)」を参照してください。

ファイルの所有権およびアクセス権を **sssd.conf** で設定します:

```
chown root:root /etc/sss/sss.conf
chmod 0600 /etc/sss/sss.conf
restorecon /etc/sss/sss.conf
```

#### 手順 4: NSS/PAM の構成

##### RHEL/CentOS:

**authconfig** を使用して SSSD を有効にします。**oddmkhomedir** をインストールして、このホームディレクトリの作成機能が SELinux に対応していることを確認します:

```
1 authconfig --enablesssd --enablesssdauth --enablemkhomedir --update
2
3 sudo systemctl start sssd
4
5 sudo systemctl enable sssd
```

##### ヒント:

Linux VDA の設定を行うときは、SSSD では Linux VDA クライアントの特別な設定がないことを考慮します。**ctxsetup.sh** スクリプトでのその他の解決方法としては、デフォルト値を使用します。

#### 手順 5: Kerberos 構成の確認

Linux VDA で使用できるように Kerberos が正しく構成されていることを確認するには、次のコマンドにより、システムの **keytab** ファイルが作成済みで **keytab** ファイルに有効なキーが含まれていることを確認します:

```
1 sudo klist -ke
```

このコマンドにより、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に対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、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に対して認証します:

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

マシン名と領域名は大文字で指定する必要があります。ドル記号 (\$) は、シェルによる置き換え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 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

次のコマンドを使用して、マシンアカウントの TGT チケットがキャッシュされたことを確認します:

```
1 sudo klist -ke
```

手順 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こと、および NSS が機能するかどうかを確認します:

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

**DOMAIN** パラメーターは短い形式のドメイン名です。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通りです:

- 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。以前はドメインユーザーアカウントは使用されていませんでした。

```
1 sudo ssh localhost -l DOMAIN\username
2
3 id -u
```

次の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:

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

次のコマンドで、ユーザーの Kerberos 資格情報キャッシュのチケットが有効で、期限切れではないことを確認します:

```
1 klist
```

## IPv6

July 16, 2024

Linux VDA では、Citrix Virtual Apps and Desktops に対応した IPv6 を使用できます。この機能を使用するときは、次の点に注意してください:

- デュアルスタック環境で、IPv6 が明示的に有効になっていない場合、IPv4 が使用されます。
- IPv4 環境で IPv6 を有効にすると、Linux VDA は機能しません。

**重要:**

- Linux VDA だけではなく、ネットワーク環境全体が IPv6 である必要があります。
- Centrifry ではピュア IPv6 をサポートしていません。

Linux VDA をインストールしている場合、IPv6 の特別なセットアップタスクは必要ありません。

## Linux VDA で IPv6 を構成する

Linux VDA の構成を変更する前に、以前 IPv6 ネットワークで Linux 仮想マシンが機能していたかを確認する必要があります。IPv6 の構成に関連する 2 つのレジストリキーがあります：

```
1 "HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
  OnlyUseIPv6ControllerRegistration"
2 "HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
  ControllerRegistrationIPv6Netmask"
```

**OnlyUseIPv6ControllerRegistration** を 1 に設定して、Linux VDA で IPv6 を有効にする必要があります：

```
1 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** で Linux VDA の登録に使用するネットワークインターフェイスを指定できます：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
  Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "
  ControllerRegistrationIPv6Netmask " -d "{
2   IPv6 netmask }
3 " --force
```

**{IPv6 netmask}** を実際のネットマスク（2000::/64 など）に置き換えます。

Citrix Virtual Apps and Desktops での IPv6 展開について詳しくは、「[IPv4/IPv6 サポート](#)」を参照してください。

## トラブルシューティング

基本の IPv6 ネットワーク環境をチェックしてから、ping6 を使用して AD および Delivery Controller に接続できるかを確認します。

## LDAPS

July 16, 2024

LDAPS は、LDAP 通信が TLS/SSL を使用して暗号化されるライトウェイトディレクトリアクセスプロトコル (LDAP) の安全なバージョンです。

デフォルトで、クライアントとサーバーアプリケーション間の LDAP 通信は暗号化されていません。LDAPS で、Linux VDA および LDAP サーバー間の LDAP クエリコンテンツを保護できます。

次の Linux VDA コンポーネントは、LDAPS との依存関係があります：

- ブローカーエージェント：Delivery Controller に Linux VDA を登録
- ポリシーサービス：ポリシー評価

以下は、LDAPS の構成に必要です：

- Active Directory (AD) /LDAP サーバーで LDAPS を有効化
- クライアントで使用するルート CA をエクスポート
- Linux VDA マシンで LDAPS を有効化または無効化
- サードパーティのプラットフォームで LDAPS の構成
- SSSD の構成
- Winbind の構成
- Centrify の構成
- Quest の構成

注：

次のコマンドを実行して、LDAP サーバーの監視サイクルを設定できます。デフォルト値は 15 分です。少なくとも 10 分に設定するようにしてください。

```
1 /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 の有効化

Microsoft 証明機関 (CA) または非 Microsoft CA のどちらかから適切な形式の証明書をインストールして、SSL 経由の LDAP (LDAPS) を有効にできます。

ヒント：

LDAPS は、ドメインコントローラーで会社のルート CA をインストールすると、自動的に有効になります。

証明書をインストールして、LDAPS 接続を確認する方法については、「[How to enable LDAP over SSL with a third-party certification authority](#)」を参照してください。

証明機関の階層に複数の層がある場合、ドメインコントローラーで LDAPS 認証の適切な証明書を自動的に取得できません。

複数の証明機関の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 \(LDAPS\) Certificate](#)」を参照してください。

### クライアントで使用するルート証明書 (CA) の有効化

クライアントは、LDAP サーバーが信頼する CA 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 証明書を信頼済みのキーストアにインポートします。

ルート CA をエクスポートする方法については詳しくは、Microsoft Support Web サイトで「[How to export Root Certification Authority Certificate](#)」を参照してください。

### Linux VDA マシンで LDAPS を有効化または無効化

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) **enable\_ldaps.sh** スクリプトを実行します。非対話モードで **enable\_ldaps.sh** スクリプトを実行するには、次の変数を利用している環境にエクスポートします:

```
1 #CTX_LDAPS_KEYSTORE_PASSWORD=  
2  
3 #CTX_LDAPS_LDAP_SERVERS=
```

- 指定されたルート CA 証明書で SSL/TLS 経由で LDAP を有効にするには:

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

- SSL/TLS 経由で LDAP チャンネルバインディングを有効にするには:

```
1 /opt/Citrix/VDA/sbin/enable_ldaps.sh -Enablecb pathToRootCA
```

注:

チャンネルバインディングのルート 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 Python 3 仮想環境が正常に作成されない場合は、「[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 ツールの使用時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 を使用せずに LDAP にフォールバックするには

```
““  
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable  
““
```

The Java keystore dedicated for LDAPS resides in **/etc/xdl/.keystore**. Affected registry keys include:

```
““  
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  
““
```

サードパーティのプラットフォームで **LDAPS** の構成

Linux VDA コンポーネントに加えて、VDA のさまざまなサードパーティのソフトウェアコンポーネントでは、SSSD、Winbind、Centrify、Quest などのセキュリティで保護された LDAP が必要な場合もあります。以下のセクションでは、LDAPS、STARTTLS または SASL（署名とシール）によるセキュリティで保護された LDAP を構成する方法について説明します。

ヒント:

これらすべてのソフトウェアコンポーネントで、SSL ポート 636 を使用し、セキュリティで保護された LDAP にすることが望ましいわけではありません。また、ほとんどの場合、LDAPS（ポート 636 での SSL 経由の LDAP）はポート 389 の STARTTLS と共存できません。

## SSSD

オプションごとに、ポート 636 またはポート 389 のセキュリティで保護された SSSD LDAP トラフィックを構成します。詳しくは、[SSSD LDAP Linux の man ページ](#)を参照してください。

## Winbind

Winbind LDAP クエリは、ADS メソッドを使用します。Winbind は、ポート 389 で StartTLS メソッドのみをサポートしています。影響を受ける構成ファイルは、**/etc/samba/smb.conf** と **/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、セキュリティで保護された LDAP は、SASL GSSAPI (署名およびシール) で構成できますが、TLS/SSL と共存することはできません。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
```

## Centrify

Centrify ではポート 636 の LDAPS をサポートしていません。一方、ポート 389 上のセキュリティで保護された暗号化は提供しています。詳しくは、[Centrify サイト](#)を参照してください。

## Quest

Quest 認証サービスはポート 636 の LDAPS をサポートしませんが、別の方法でポート 389 のセキュリティで保護された暗号化を提供します。

### トラブルシューティング

この機能を使用すると、以下の問題が発生することがあります:

- **LDAPS** サービスの可用性  
AD/LDAP サーバーで LDAPS 接続が使用可能であることを確認します。デフォルトでは、このポートは 636 です。
- **LDAPS** を有効にすると **Linux VDA** の登録が失敗する  
LDAP サーバーとポートが正しく構成されていることを確認します。最初にルート CA 証明書をチェックして、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

LDAPS 関連のキーを誤って **enable\_ldaps.sh** を使用せずに更新してしまうと、LDAPS コンポーネントの依存関係を損なう可能性があります。

- **LDAP** トラフィックは、**Wireshark** やその他のネットワーク監視ツールから **SSL/TLS** で暗号化されません  
デフォルトでは、LDAPS は無効になっています。それを強制するには、**/opt/Citrix/VDA/sbin/enable\_ldaps.sh** を実行します。

- **Wireshark** やその他のネットワーク監視ツールからの **LDAPS** トラフィックが存在しない

LDAP/LDAPS トラフィックは、Linux VDA の登録やグループポリシーの評価を行う際に発生します。

- **AD** サーバーで **LDP** 接続を実行して **LDAPS** の可用性を確認できない

IP アドレスの代わりに、AD FQDN を使用します。

- **/opt/Citrix/VDA/sbin/enable\_ldaps.sh** スクリプトを実行してルート **CA** 証明書をインポートできない  
CA 証明書のフルパスを指定して、ルート CA 証明書の種類が正しいことを確認します。サポートされている Java Keytool の種類の大半で対応しています。サポート一覧にない場合は、最初に種類を変更してください。証明書の形式の問題が発生した場合は、Base64 で暗号化された PEM 形式の使用をお勧めします。

- ルート **CA** 証明書を **Keytool** 一覧に表示できない

**/opt/Citrix/VDA/sbin/enable\_ldaps.sh** を実行して LDAPS を有効にすると、証明書が **/etc/xdm/.keystore** にインポートされ、キーストアを保護するパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再度実行して新しいキーストアを作成します。

## Xauthority

July 16, 2024

Linux VDA は、対話型のリモート制御で X11 ディスプレイ機能 (**xterm** と **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 間のセキュリティで保護された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。

このセキュリティで保護された通信の権限を保護するには、以下の 2 つの方法があります：

- **Xhost**。デフォルトでは、Xhost コマンドはローカルホスト XClient と XServer の通信のみを許可します。リモート XClient の XServer へのアクセスを許可すると、特定のマシンで権限を付与するために Xhost コマンドが実行される必要があります。あるいは、**xhost +** を使用して XClient に XServer への接続を許可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。このファイルは、XServer の認証の際に xauth が使用する Cookie に資格情報を保存するために使用されます。XServer

インスタンス (Xorg) が起動されるときに、特定のディスプレイへの接続を認証するためにこの Cookie が使われます。

## 機能

Xorg が起動されると、`.Xauthority` ファイルは Xorg に渡されます。この `.Xauthority` ファイルには次の要素が含まれます:

- 表示番号
- リモート要求プロトコル
- Cookie 番号

`xauth` コマンドを使用して、このファイルを参照できます。例:

```
1 # xauth -f ~/.Xauthority
2
3 # > list
4
5 # > us01msip06:107 MIT-MAGIC-COOKIE-1
    fb228d1b695729242616c5908f11624b
```

**XClient** がリモートで Xorg に接続する場合、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 で Cookie 番号の 1 つを含む `.Xauthority` 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 上で **Xauthority** を有効にするには、次の 2 個のレジストリキーを作成する必要があります:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
    CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "
    XauthEnabled" -d "0x00000001" --force
2
3 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
    CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "ListenTCP"
    -d "0x00000001" --force
```

**Xauthority** に有効にしてから、手動によるか共有ホームディレクトリをマウントすることで `.Xauthority` ファイルを **XClient** に渡します:

- `.Xauthority` ファイルを XClient に手動で渡す

ICA セッションを起動した後、Linux VDA は XClient 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ファイルを保存します。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`ファイルに保存した表示番号であり、**XAUTHORITY**は **Xauthority** のファイルパスです。たとえば、次のコマンドを表示します：

```
1 export DISPLAY={
2   Display number stored in the Xauthority file }
3
4
5 export XAUTHORITY={
6   the file path of .Xauthority }
```

注：

**XAUTHORITY** 環境変数が設定されていない場合、`~/Xauthority`ファイルがデフォルトで使用されます。

- 共有ホームディレクトリをマウントすることにより、`.Xauthority`ファイルを XClient に渡す

簡単な方法は、ログオンユーザーの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 セッションを起動すると、ログオンユーザーのホームディレクトリで `.Xauthority`ファイルが作成されます。このホームディレクトリが XClient と共有される場合、ユーザーがこの `.Xauthority`ファイルを手動で XClient に転送する必要はありません。**DISPLAY** および **XAUTHORITY** 環境変数を正しく設定した後、XServer で GUI が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、次のトラブルシューティング手順に従ってください：

1. root 特権を持つ管理者として、すべての Xorg Cookie を取得します：

```
1 ps aux | grep -i xorg
```

このコマンドは、起動中 Xorg に渡される Xorg プロセスとパラメーターを表示します。もう 1 つのパラメーターは、どの `.Xauthority`ファイルが使用されるかを表示します。例：

```
1 /var/xdl/xauth/.Xauthority110
```

**Xauth** コマンドを使用して、Cookie を表示します：

```
1 Xauth -f /var/xdl/xauth/.Xauthority110
```

2. **Xauth**コマンドを使用して、`~/Xauthority`に含まれる Cookie を表示します。同じ表示番号の場合、表示される Cookie は Xorg および XClient の `.Xauthority`ファイルで同じである必要があります。
3. Cookie が同じであれば、リモートディスプレイポートが Linux VDA の IP アドレスと公開デスクトップの表示番号を使用してアクセスできるかを確認します。

たとえば、XClient マシンで次のコマンドを実行します：

```
1 telnet 10.158.11.11 6160
```

ポート番号は、6000 + 表示番号の合計です。

telnet の操作が失敗すると、ファイアウォールが要求をブロックすることがあります。

## 認証

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [Azure Active Directory を使用した認証](#)
- [ダブルホップシングルサインオン認証](#)
- [フェデレーション認証サービス](#)
- [SSO 以外の認証](#)
- [スマートカード](#)
- [認証が不要なユーザー（匿名ユーザー）のアクセス](#)

## Azure Active Directory を使用した認証

August 20, 2024

注：

この機能は、Azure でホストされる VDA でのみ使用できます。

ニーズに基づいて、Azure に 2 種類の Linux VDA を展開できます：

- Azure AD DS 参加済み VM。VM が、Azure Active Directory (AAD) ドメインサービス (DS) の管理対象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。
- ドメイン非参加 VM。VM が、AAD ID サービスと統合されており、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 について詳しくは、こちらの [Microsoft 社の記事](#) を参照してください。

この記事では、ドメイン非参加 VDA で AAD ID サービスを有効化および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

- Ubuntu 22.04、20.04
- RHEL 8.10、8.9、8.8
- SUSE 15.5

詳しくは、[Microsoft 社の記事](#)を参照してください。

手順 **1: Azure Portal** でテンプレート仮想マシンを作成する

テンプレート仮想マシンを作成し、Azure CLI を VM にインストールします。

1. Azure Portal でテンプレート仮想マシンを作成します。[**Review + create**] をクリックする前に、[**Management**] タブで [**Login with Azure AD**] を選択してください。

The screenshot shows the 'Create a virtual machine' page in the Azure Portal, specifically the 'Management' tab. The page is titled 'Create a virtual machine' and has a breadcrumb 'Home > Create a resource >'. Below the title, there are tabs for 'Basics', 'Disks', 'Networking', 'Management', 'Advanced', 'Tags', and 'Review + create'. The 'Management' tab is selected. The page content includes sections for 'Azure Security Center', 'Monitoring', 'Identity', 'Azure AD', 'Auto-shutdown', 'Backup', and 'Guest OS updates'. In the 'Identity' section, the 'System assigned managed identity' checkbox is checked and highlighted with a red box. Below it, there is a note: 'System managed identity must be on to login with Azure AD credentials. Learn more'. In the 'Azure AD' section, the 'Login with Azure AD' checkbox is checked and highlighted with a red box. Below it, there is a note: 'RBAC role assignment of Virtual Machine Administrator Login or Virtual Machine User Login is required when using Azure AD login. Learn more'. At the bottom left, there is a blue button labeled 'Review + create'. At the bottom center, there are two buttons: '< Previous' and 'Next: Advanced >'. The 'Review + create' button is highlighted in blue.

2. テンプレート仮想マシンで Azure CLI を VM にインストールします。  
詳しくは、[Microsoft 社の記事](#)を参照してください。

## 手順 2: テンプレート仮想マシンでマスターイメージを準備

マスターイメージを準備するには、「[Azure での MCS を使用した Linux 仮想マシンの作成](#)」の「手順 3: マスターイメージを準備する」の手順に従ってください。

## 手順 3: テンプレート仮想マシン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、次の手順に従って 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。

```
1 Modify /var/xdl/mcs/mcs_util.sh
```

2. `function read_non_domain_joined_info()` を見つけて、`NonDomainJoined` の値を 2 に変更します。例として、次のコードブロックを参照してください。

```
1 function read_non_domain_joined_info()
2 {
3
4 log "Debug: Enter read_non_domain_joined_info"
5 # check if websocket enabled
6 TrustIdentity=`cat ${
7 id_disk_mnt_point }
8 ${
9 ad_info_file_path }
10 | grep '[TrustIdentity]' | sed 's/\s//g'`
11 if [ "$TrustIdentity" == "[TrustIdentity]" ]; then
12 NonDomainJoined=2
13 fi
14 ...
15 }
```

3. 変更を保存します。
4. テンプレート仮想マシンをシャットダウンします。

## 手順 4: テンプレート仮想マシンから Linux VM を作成する

ドメイン非参加テンプレート仮想マシンの準備ができたなら、次の手順に従って VM を作成します:

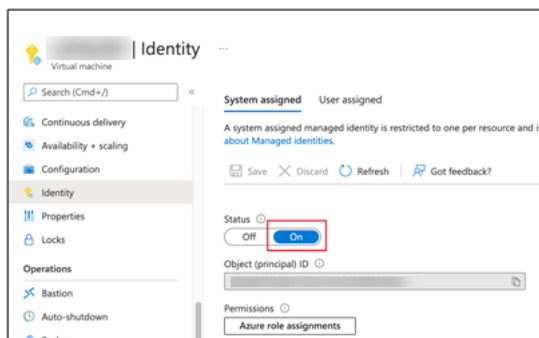
1. Citrix Cloud にサインインします。
2. Citrix DaaS をダブルクリックして、[完全な構成] 管理コンソールにアクセスします。
3. [マシンカタログ] で、Machine Creation Services を使用してテンプレート仮想マシンから Linux VM を作成することを選択します。詳しくは、Citrix DaaS ドキュメントの「[ドメイン非参加の VDA](#)」を参照してください。

## 手順 5: Linux VM に AAD ユーザーアカウントを割り当てる

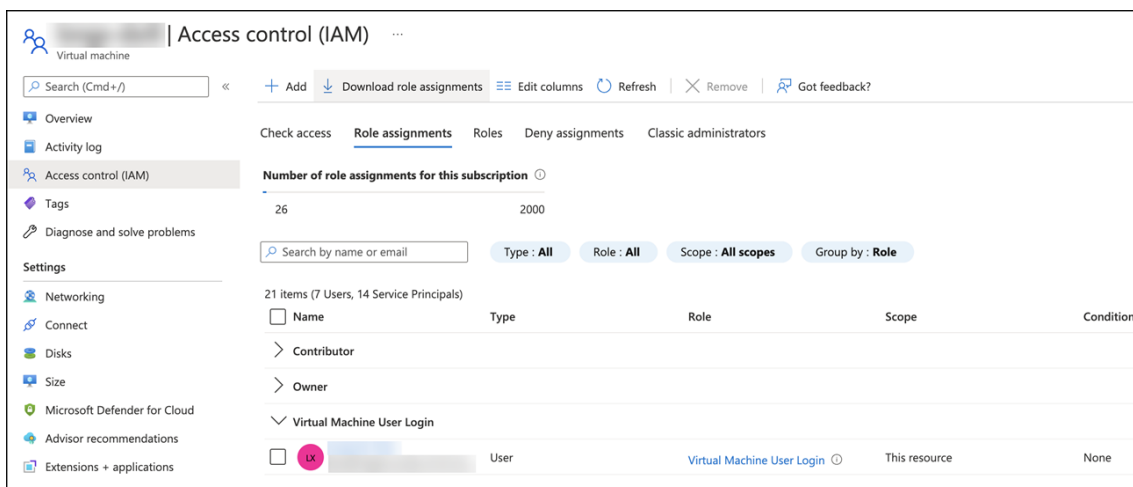
ドメイン非参加 VM を作成したら、それらに AAD ユーザーアカウントを割り当てます。

AAD ユーザーアカウントを VM に割り当てるには、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. **[Identify]** > **[System assigned]** タブで、**[System Identity]** を有効にします。



3. **[Access control (IAM)]** > **[Role assignments]** タブで、**[Virtual Machine User Login]** の欄を見つけ、必要に応じて AAD ユーザーアカウントを追加します。

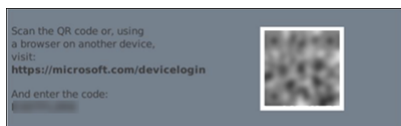


## ドメイン非参加 VDA へのログオン

組織内のエンドユーザーは、ドメイン非参加 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] ページが開きます。
2. ドメイン非参加デスクトップをダブルクリックします。[AAD LOGIN] ページが表示されます。

このページは、VDA で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。管理者は、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。手順について詳しくは、以降のセクションを参照してください。



3. 画面の指示に基づいて、次のいずれかの方法でデスクトップセッションにログオンします:

- QR コードをスキャンして、コードを入力します。
- AAD ユーザー名とパスワードを入力します。

### AAD アカウント/パスワードログインモードへの変更

デフォルトでは、Linux VDA はデバイスコードを使用して AAD ユーザーを認証します。詳しくは、[Microsoft 社の記事](#)を参照してください。ログインモードを [AAD アカウント/パスワード] に変更するには、次の手順を実行します:

VDA で次のコマンドを実行し、AADAcctPwdAuthEnable キーを見つけて、その値を 0x00000001 に変更します。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Services\CitrixBrokerAgent\WebSocket" -t "REG_DWORD" -v "AADAcctPwdAuthEnable" -d "0x00000001" --force
```

注:

この方法は、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使えません。

### ダブルホップシングルサインオン認証

July 16, 2024

StoreFront ストアにアクセスするためのユーザー資格情報を、Linux 向け Citrix Workspace アプリおよび Citrix Receiver for Linux 13.10 の AuthManager モジュールに入力できます。入力後、ユーザー資格情報を再度入力することなく、Linux 仮想デスクトップセッションから仮想デスクトップおよびアプリケーションに、クライアントを使用してアクセスできます。

注:

この機能は Linux 向け Citrix Workspace アプリおよび Citrix Receiver for Linux 13.10 でサポートされて



います。

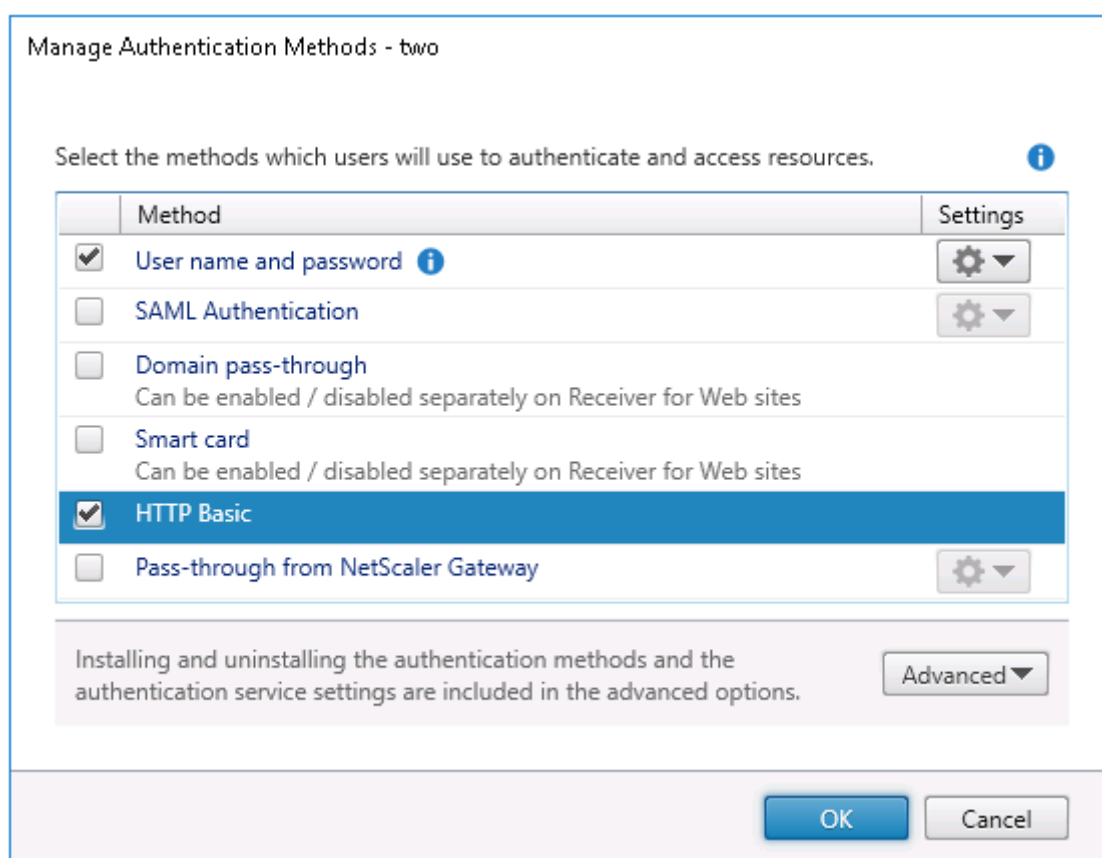
この機能を有効にするには:

1. Linux VDA に、Linux 向け Citrix Workspace アプリまたは 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。異なるパスにアプリをインストールする場合、`ICAROOT` 環境変数を実際のインストールパスを参照するよう設定します。

2. Citrix StoreFront 管理コンソールで、対象のストアに **HTTP** 基本認証方式を追加します。



3. HTTP 基本認証を許可するには、次のキーを AuthManager 構成ファイル (`$ICAROOT/config/AuthMan-Config.xml`) に追加します:

```

1 <Protocols>
2   <HTTPBasic>
3     <Enabled>True</Enabled>
4   </HTTPBasic>
5 </Protocols>

```

4. 次のコマンドを実行して、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します。

```
1 cp rootcert.pem $ICAROOT/keystore/cacerts/  
2 $ICAROOT/util/ctx_rehash $ICAROOT/keystore/cacerts/
```

5. 次のコマンドを実行して、この機能を有効にします:

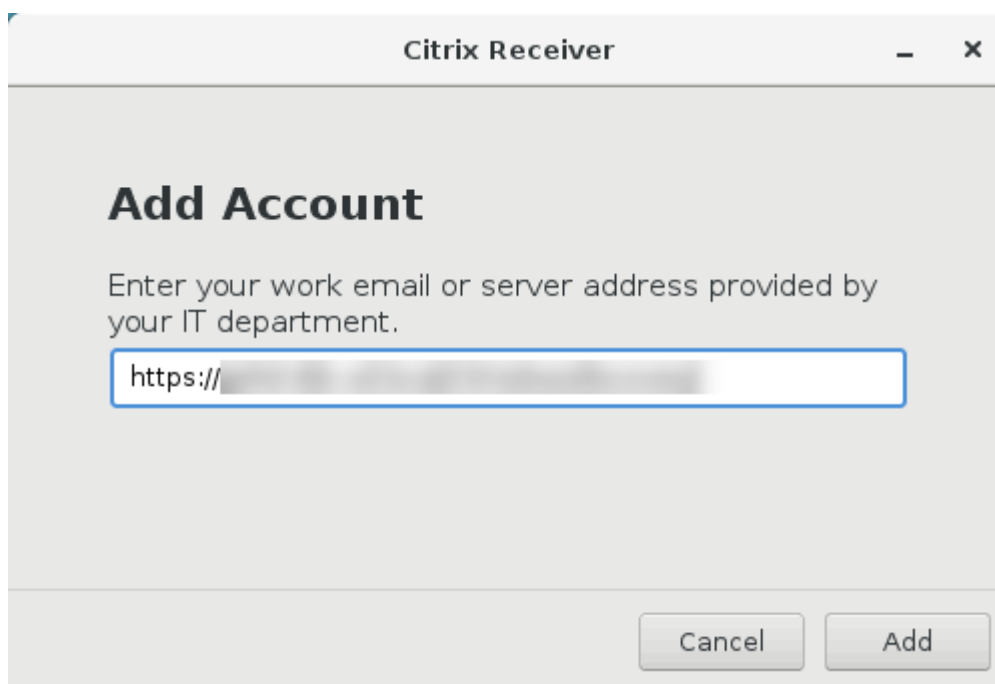
```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\  
CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d "0  
x00000001"
```

6. Linux 仮想デスクトップセッションを開始して、このセッションで Linux 向け Citrix Workspace アプリまたは Citrix Receiver for Linux 13.10 を起動します。

Citrix Workspace アプリを初めて起動したときに、ストアアカウントの入力を求められます。以降は、指定済みのストアに自動的にログオンします。

注:

ストアアカウントの HTTPS URL を入力します。



## フェデレーション認証サービス

August 20, 2024

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することができます。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 用の Windows 環

境の構成については、「[フェデレーション認証サービス](#)」を参照してください。この記事では、Linux VDA に固有の追加情報を提供します。

注:

- Linux VDA は、**In-session Behavior** ポリシーをサポートしていません。
- Linux VDA は、短い接続を使用して FAS サーバーとデータを送信します。

### サポートされているディストリビューション

FAS は、限定的な 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ます。次のマトリックスを参照してください:

	Winbind	SSSD	Centrify	PBIS	Quest
Amazon Linux 2	はい	はい	はい	はい	いいえ
Debian 12.5/11.9	はい	はい	はい	はい	いいえ
RHEL 9.4/9.3/9.2	はい	はい	いいえ	いいえ	はい
RHEL 8.10/8.9/8.8	はい	はい	はい	はい	はい
Rocky Linux 9.4/9.3/9.2	はい	はい	いいえ	いいえ	いいえ
Rocky Linux 8.10/8.9/8.8	はい	はい	いいえ	いいえ	いいえ
SUSE 15.5	はい	はい	はい	いいえ	いいえ
Ubuntu 22.04/20.04	はい	はい	はい	はい	はい

### Linux VDA での FAS の構成

#### 証明書のインストール

ユーザーの証明書を検証するには、ルート CA 証明書とすべての中間証明書を VDA にインストールします。たとえば、ルート CA 証明書をインストールするには、前述の「**Microsoft CA** からの **CA** 証明書の取得 (AD で)」の手順で AD ルート証明書を取得するか、またはルート CA サーバー (<http://CA-SERVER/certsrv>) から証明書をダウンロードします。

注:

次のコマンドは、中間証明書の構成にも適用されます。

たとえば、DER ファイル (.crt、.cer、.der) を PEM に変換するには、次のようなコマンドを実行します:

```
1 sudo openssl x509 -inform der -in root.cer -out root.pem
```

続いて、次のようなコマンドを実行して、ルート CA 証明書を `openssl` ディレクトリにインストールします:

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

注:

ルート CA 証明書を `/root` パス下に配置しないでください。配置すると、FAS にはルート CA 証明書への読み取り権限がなくなります。

### ctxfascfg.sh の実行

ctxfascfg.sh スクリプトを実行して FAS を構成します:

```
1 sudo /opt/Citrix/VDA/sbin/ctxfascfg.sh
```

ctxfascfg.sh をサイレントモードで実行できます。スクリプトをサイレントモードで実行する前に、次の環境変数を設定します:

- **CTX\_FAS\_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis | quest:** Active Directory の統合方式を指定。CTX\_EASYINSTALL\_ADINTEGRATIONWAY が指定されている場合、CTX\_EASYINSTALL\_ADINTEGRATIONWAY と同じ値です。CTX\_EASYINSTALL\_ADINTEGRATIONWAY が指定されていない場合、CTX\_FAS\_ADINTEGRATIONWAY は独自の値を使用します。
- **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フルパスを指定します。ここで、「certificate path」は証明書のパスです。
- **CTX\_FAS\_KDC\_HOSTNAME:** PBIS および Quest を選択するときに、キー配布センター (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。特に指定しない限り 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。詳しくは、Knowledge Center の [CTX322129](#) を参照してください。
- **CTX\_FAS\_SERVER\_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 グループポリシーにより構成されます。ドメイン GPO での FAS ポリシー設定について詳しくは、「[グループポリシーの構成](#)」を参照してください。Linux VDA は AD グループポリシーをサポートしていません。代わりに、セミコロンで区切られた FAS サーバーの一覧を使用できます。シーケンスは、AD グループポリシ

一で設定したものと同一である必要があります。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスの順番は変更しません。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'」。

既存の Linux VDA インストールをアップグレードするには、次のコマンドを実行して FAS サーバーを設定し、`ctxvda` サービスを再起動して設定を有効にすることができます。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
  VirtualDesktopAgent\Authentication\UserCredentialService" -t "
  REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
2
3 systemctl restart ctxjproxy
4
5 systemctl restart ctxvda
```

`ctxreg` を使用して FAS サーバーを更新するには、次のコマンドを実行します：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
  VirtualDesktopAgent\Authentication\UserCredentialService" -v "
  Addresses" -d "<Your-FAS-Server-List>"
2
3 systemctl restart ctxjproxy
4
5 systemctl restart ctxvda
```

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

次に、このスクリプトは `krb5-pkinit` パッケージと `pam_krb5` パッケージをインストールし、関連する構成ファイルを設定します。RHEL 8 以降の場合、`PAM_KRB5` が EPEL リポジトリに移動されると、スクリプトはそれらのディストリビューションで EPEL を有効にしようとします。

### FAS の無効化

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：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
  VirtualDesktopAgent\Authentication\UserCredentialService" -t "REG_SZ"
  -v "Addresses" -d "" --force
2
3 systemctl restart ctxjproxy
4
5 systemctl restart ctxvda
```

## 制限事項

- 現在、FAS は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、「[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](#)」の記事で説明している一般的な FAS 環境のみがサポートされており、**Windows 10 Azure AD Join** は含まれません。

## 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、次のことを確認してください：

- Linux VDA が正しくインストールされ、構成されています。
- FAS 以外のセッションは、パスワード認証を使用して共通ストアで正常に起動できます。

FAS 以外のセッションが適切に機能している場合は、**Login** クラスの HDX ログレベルを VERBOSE に設定し、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 [CTX220130](#) の記事を参照してください。

Linux **XDPing** ツールを使用して Linux VDA 環境に一般的な構成の問題がないか確認することもできます。

## FAS サーバー構成エラー

FAS ストアからセッションを起動すると失敗します。

**/var/log/xdl/hdx.log** を確認し、次のようなエラーログを探します：

```
1 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [
    Logon Type] Federated Authentication Logon.
2
3 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas:
    entry
4
5 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
    connect to server 0
6
7 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0:
    failed to connect: Connection refused.
8
9 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 configured in condb
10
11 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit
    , 43
12
13 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user:
    failed to validate fas credential
14
```

```
15 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\Your-FAS-Server-List」に設定されていることを確認します。

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

既存の設定が間違っている場合は、前述の「FAS サーバーの設定」の手順に従って再設定します。

間違った **CA** 証明書の構成

FAS ストアからセッションを起動すると失敗します。灰色のウィンドウが表示され、数秒後に消えます。



`/var/log/xdl/hdx.log`を確認し、次のようなエラーログを探します:

```
1 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
    get_logon_certificate: entry
2
3 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller:
    current process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]
4
5 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
    get_public_certificate: entry
6
7 2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
    for response...
8
9 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query
    to server success
10
11 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
    get_public_certificate: exit
12
13 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
    input size 1888
14
```

```
15 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
    output size 1415
16
17 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
    get_logon_certificate: get logon certificate success
18
19 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
    cache certificate success
20
21 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
    get_logon_certificate: exit, 0
22
23 2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
    pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策 /etc/krb5.confにルート CA 証明書とすべての中間証明書を格納するフルパスが正しく設定されていることを確認します。フルパスは次のようになります:

```
1  [realms]
2
3  EXAMPLE.COM = {
4
5
6      .....
7
8      pkinit_anchors = DIR:/etc/pki/CA/certs/
9
10     .....
11
12 }
```

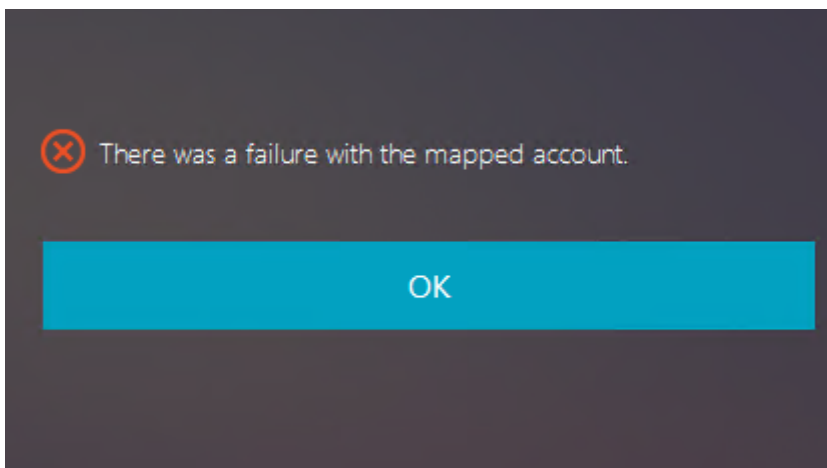
既存の設定が間違っている場合は、前述の「[証明書のインストール](#)」の手順に従って再設定します。

または、ルート CA 証明書が有効かどうかを確認します。

シャドウアカウントマッピングエラー

FAS は SAML 認証により構成されます。ADFS ユーザーが ADFS ログオンページでユーザー名とパスワードを入力すると、次のエラーが発生することがあります。





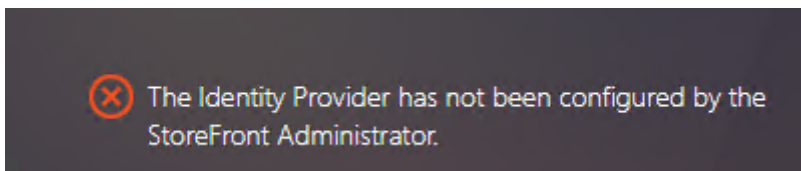
このエラーは次のことを意味します：

- ADFS ユーザーが正常に確認されたが、AD にシャドウユーザーが構成されていない。

解決策 AD にシャドウアカウントを設定します。

#### **ADFS** が構成されていない

FAS ストアへのログオン中に次のエラーが発生します：



この問題は、ADFS が展開されていない状態で、FAS ストアが SAML 認証を使用するよう構成した場合に発生します。

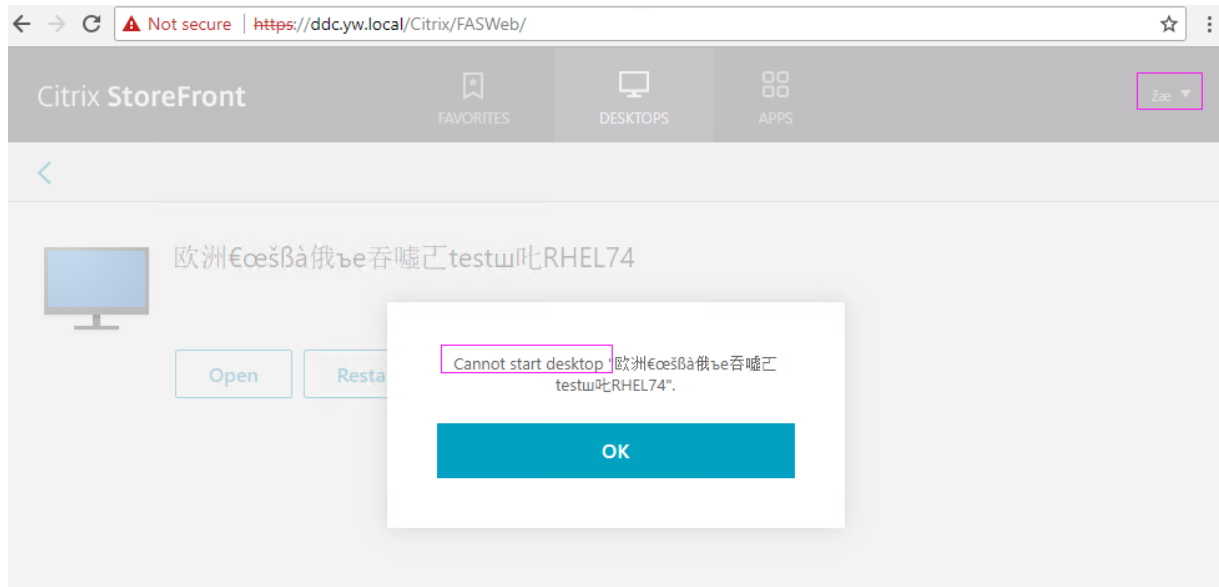
解決策 フェデレーション認証サービス用の ADFS IdP の展開詳しくは、「[フェデレーション認証サービスの ADFS の展開](#)」を参照してください。

#### 関連情報

- 一般的な FAS 環境については、「[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](#)」を参照してください。
- フェデレーション認証サービスの「[詳細な構成](#)」では「方法」の記事を紹介しています。

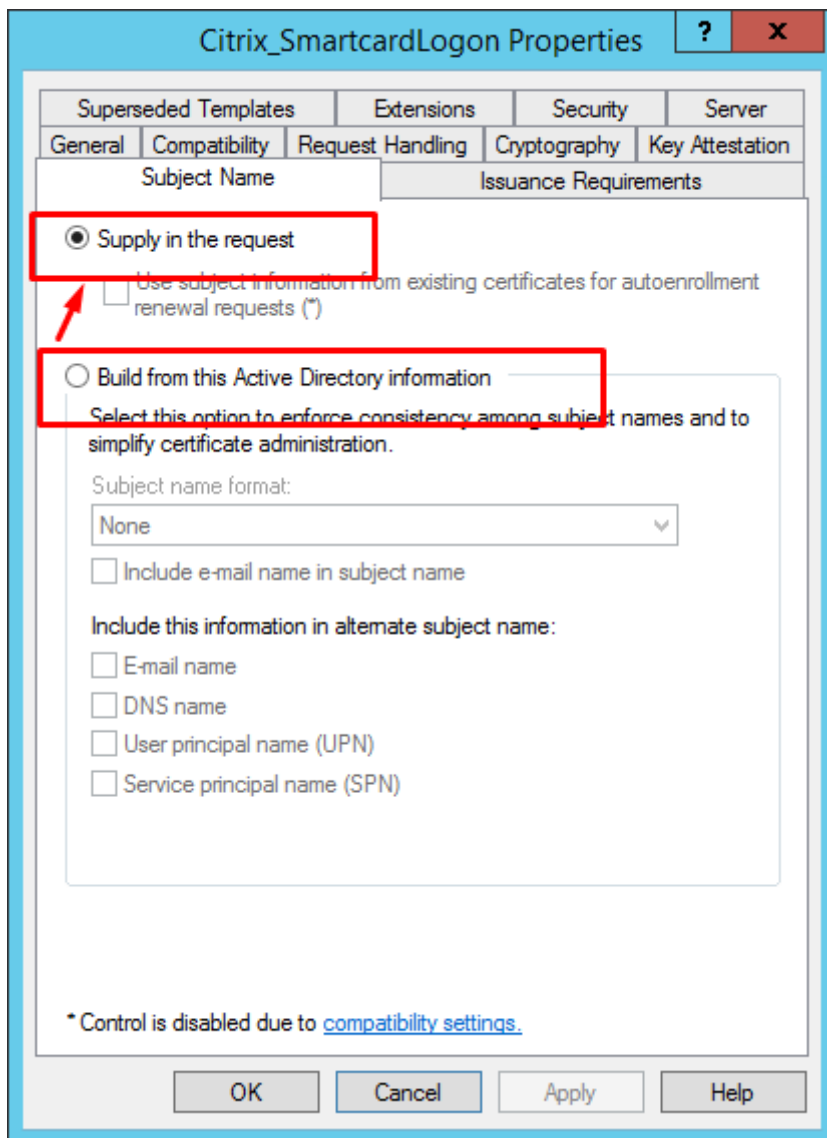
### 既知の問題

FAS が使用されている場合、英語以外の文字を使用して公開デスクトップまたはアプリセッションを開始しようとすると、失敗することがあります。



### 回避方法

CA ツールの [テンプレートの管理] を右クリックし、[Citrix\_SmartcardLogon] テンプレート上で [Active Directory の情報から構築する] を [要求に含まれる] に変更します:



## FIDO2 (プレビュー)

July 16, 2024

Linux VDA でホストされている Google Chrome を使用して Web サイトにアクセスするための FIDO2 認証を設定できます。

注:

この機能はプレビュー段階です。プレビュー機能は、一部が英語のままの場合があります。また、実稼働環境以外での使用をお勧めします。Citrix テクニカルサポートは、Technical Preview 機能で見つかった問題をサポートしません。

Linux VDA は、FIDO2 と Google Chrome の組み合わせのみをサポートします。

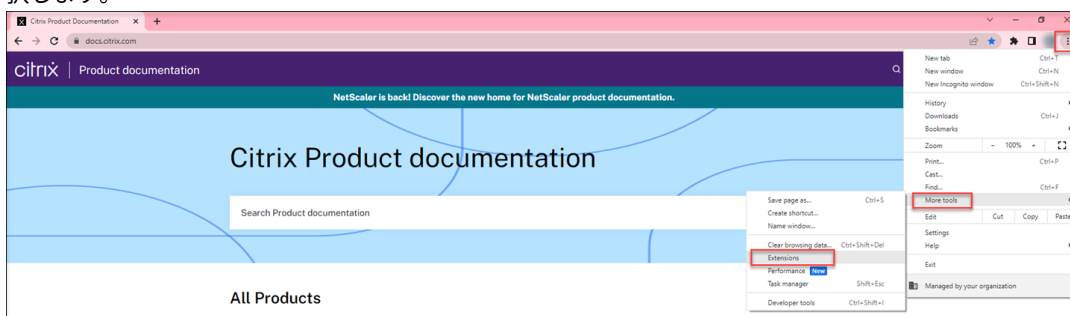
FIDO2 認証を設定するには、次の手順を実行します：

## 1. Citrix FIDO2 拡張パッケージをダウンロードします。

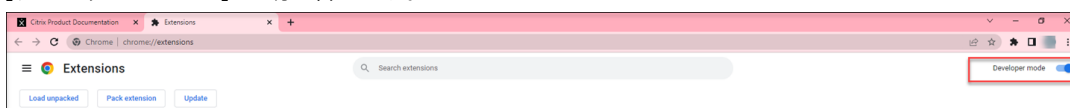
- a) [Citrix Virtual Apps and Desktops のダウンロードページ](#)にアクセスします。
- b) 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
- c) **Components** をクリックして Linux VDA を見つけます。
- d) Linux VDA をクリックしてダウンロードページを開きます。
- e) ソースパッケージをダウンロードします。
- f) ソースパッケージを解凍して、**FIDO2-JavaScript-Extensions.zip** を見つけます。
- g) FIDO2 拡張機能パッケージを解凍します。FIDO2 拡張機能ディレクトリは **[extensions] > [chrome] > [fido2]** にあります。

## 2. Google Chrome に Citrix FIDO2 拡張機能を追加します。

- a) Linux VDA でホストされている Google Chrome を開きます。
- b) アドレスバーの右側にある 3 つのドット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]** 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こむ]** をクリックし、**[extensions] > [chrome] > [fido2]** で拡張機能ディレクトリを選択します。

## 3. FIDO2 認証を使用する Web サイトで、FIDO2 認証を使用するための 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、認証方法として FIDO2 を追加します。

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をタッチするように求められます。

## SSO 以外の認証

July 16, 2024

この記事では、Linux VDA で SSO（シングルサインオン）ではない認証を有効にする方法について説明します。

### 概要

デフォルトでは、Linux VDA ではシングルサインオン（SSO）が有効になっています。ユーザーは、1 つの資格情報のセットを使用して Citrix Workspace アプリと VDA セッションにログオンします。

ユーザーが別の資格情報のセットを使用して VDA セッションにログオンできるようにするには、Linux VDA で SSO を無効にします。次の表は、SSO 以外の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 の無効化

Linux VDA で次のコマンドを実行します：

```
1 /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

August 20, 2024

Linux 仮想デスクトップセッションにログオンするときに、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上でのスマートカードのリダイレクトによ

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。次のようなユースケースがあります：

- ドキュメントへのデジタル署名の追加
- メールの暗号化または暗号化解除
- Web サイトへの認証

Linux VDA は、この機能に Windows VDA と同じ構成を使用します。詳しくは、この記事の「[スマートカード環境を構成する](#)」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

スマートカードによるパススルー認証を使用できるかは、次の条件により異なります：

- Linux VDA が、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る：
  - RHEL 9.4/ 9.3/9.2
  - RHEL 8.10/8.9/8.8
  - Rocky Linux 9.4/ 9.3/9.2
  - Rocky Linux 8.10/8.9/8.8
  - Ubuntu 22.04
  - Ubuntu 20.04
  - Debian 12.5/11.9

VDA のインストールが完了したら、VDA が Delivery Controller に登録でき、公開された Linux デスクトップセッションを Windows 資格情報を使用して起動できることを確認します。

- OpenSC がサポートするスマートカードが使用されている。詳しくは、「[OpenSC がスマートカードをサポートしていることの確認](#)」を参照してください。

### OpenSC がスマートカードをサポートしていることの確認

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。OpenSC は CoolKey と完全に互換性がある後継で、さまざまな種類のスマートカードをサポートします（「[Smart Card Support in Red Hat Enterprise Linux](#)」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 や他の小売業者から簡単に購入できる一体型の 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。

もっと高度なスマートカードが必要になった場合は、サポート対象の 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をインストールする](#)」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：

```
1 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、スマートカード内の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を完了します。

1. 通常は CA サーバーから、ルート証明書を PEM 形式で取得します。

次のようなコマンドを実行して、DER ファイル (\*.crt、\*.cer、\*.der) を PEM に変換できます。次のコマンド例では、**certnew.cer** は DER ファイルです。

```
1 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。例として **certnew.pem** ファイルを使用しています。

```
1 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 mkdir -p <path where you install the root certificate>`を実行します。

### スマートカード環境を構成する

`ctxsmartlogon.sh` スクリプトを使用してスマートカード環境を構成するか、手動で構成を完了することができます。

(オプション **1**) **ctxsmartlogon.sh** スクリプトを使用してスマートカード環境を構成する

注：

`ctxsmartlogon.sh` スクリプトは、PKINIT 情報をデフォルトの領域に追加します。この設定は、`/etc/krb5.conf` 構成ファイルを使用して変更できます。

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:

ドメインへの参加に SSSD を使用している場合は、ctxsmartlogon.sh の実行後に SSSD サービスを再起動してください。

RHEL 8 以降の場合、PAM\_KRB5 が EPEL リポジトリに移動されると、スクリプトはそれらのディストリビューションで EPEL を有効にしようとします。

```
1 sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります:

```
*****
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
*****
Step 1:Enable smart card logon.
Do you want enable smart card logon? (y/n)[y] y
Step 2:Select the Active Directory integration method.
Please select which Active Directory integration method to use:
  1: Winbind
  2: SSSD
  3: Centrify
Select one of the above options (1-3)[1] 1
Step 3:Install dependent packages.
Installing, please wait...
[krb5-pkinit][pam_krb5] already installed.
[pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][coolkey] already installed.
Packages installed. [Success]
Step 4:Configure krb5.conf.
Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
/etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/libcoolkeypk11.so):/usr/lib64/pkcs11/libcoolkeypk11.so
/etc/pam.d/ctxfsc configure successfully.
/etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

## 注:

ドメイン参加方法として Quest を使用する場合は、キー配布センター (KDC) のホスト名を指定する必要があります。例:



```
[root@x-rh79qu110 ~]# /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
*****
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
*****
Step 1:Enable smart card logon.
Do you want enable smart card logon? (y/n)[y]
Step 2:Select the Active Directory integration method.
Please select which Active Directory integration method to use:
  1: Winbind
  2: SSSD
  3: Centrify
  4: Quest
Select one of the above options (1-4)[1] 4
Step 3:Install dependent packages.
Installing, please wait...
[krb5-pkinit][pam_krb5] already installed.
[pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][opensc] already installed.
Packages installed. [Success]
Step 4:Configure krb5.conf.
Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
Specify the KDC hostname:kdc.test.local
/etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/opensc-pkcs11.so):usr/lib64/pkcs11/opensc-pkcs11.so
/etc/pam.d/ctxfsc configure successfully.
/etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

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

```
1 sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります:

```
*****
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
*****
Step 1:Enable smart card logon.
Do you want enable smart card logon? (y/n)[y] n
ctxsmartlogon.sh exit.
```

(オプション 2) スマートカード環境を手動で構成する Linux VDA は、Windows VDA と同じスマートカード環境を使用します。この環境では、ドメインコントローラー、Microsoft 証明機関 (CA)、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く構成について詳しくは、Knowledge Center の [CTX206156](#) の記事を参照してください。

次の手順に進む前に、以下の点について確認してください:

- すべてのコンポーネントが正しく構成されている。
- 秘密キーとユーザー証明書がスマートカードにダウンロードされている。
- スマートカードを使用して VDA に正常にログオンできる。

**PC/SC Lite** パッケージをインストールする PCSC Lite は、Linux でのパーソナルコンピューター/スマートカード (PC/SC) 仕様の実装です。スマートカードや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。

次のコマンドを実行して、PC/SC Lite パッケージをインストールします：

**RHEL 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：**

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

**Ubuntu 22.04、Ubuntu 20.04、Debian 12.5、Debian 11.9：**

```
1 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 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：**

```
1 yum install opensc
```

**Ubuntu 22.04、Ubuntu 20.04、Debian 12.5、Debian 11.9：**

```
1 apt-get install -y opensc
```

スマートカード認証用の **PAM** モジュールをインストールする 次のコマンドを実行して、pam\_krb5 および krb5-pkinit モジュールをインストールします。

**RHEL 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：**

```
1 yum install krb5-pkinit
```

**Ubuntu 22.04、Ubuntu 20.04：**

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

**Debian 12.5、Debian 11.9：**

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

pam\_krb5 モジュールは、プラグイン可能な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確認し、キー配布センター (KDC) のチケット配布チケットを取得できます。krb5-pkinit モジュールには PKINIT プラグインが含まれていて、クライアントが秘密キー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します。

**pam\_krb5** モジュールを構成する pam\_krb5 モジュールは KDC と対話して、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：

```
1 authconfig --enablekrb5 --update
```

**/etc/krb5.conf** 構成ファイルに、実際の領域に応じた PKINIT 情報を追加します。

注:

**pkinit\_cert\_match** オプションは、クライアント証明書が PKINIT 認証の試行に使用される前に一致する必要がある一致規則を指定します。一致規則の構文は次のとおりです:

**[relation-operator] component-rule ...**

**relation-operator** には、**&&** (すべてのコンポーネント規則が一致する必要がある) または **||** (1つのコンポーネント規則のみが一致する必要がある) のいずれかを使用できます。

汎用 **krb5.conf** ファイルの例を次に示します:

```
1 EXAMPLE.COM = {
2
3
4     kdc = KDC.EXAMPLE.COM
5
6     auth_to_local = RULE:[1:$1@$0]
7
8     pkinit_anchors = FILE:<path where you install the root certificate
9         >/certnew.pem
10
11     pkinit_kdc_hostname = KDC.EXAMPLE.COM
12
13     pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
14
15     pkinit_eku_checking = kpServerAuth
16 }
```

構成ファイルは、PKINIT 情報を追加した後、次のようになります。

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

**PAM** 認証を構成する PAM 構成ファイルは、どのモジュールを PAM 認証に使用しているかを示します。**pam\_krb5** を認証モジュールとして追加するには、**/etc/pam.d/smartcard-auth** ファイルに次の行を追加します:

```
auth [success=done ignore=ignore default=die] pam_krb5.so preauth_options  
=X509_user_identity=PKCS11:<path to the pkcs11 driver>/opensc-pkcs11.  
so
```

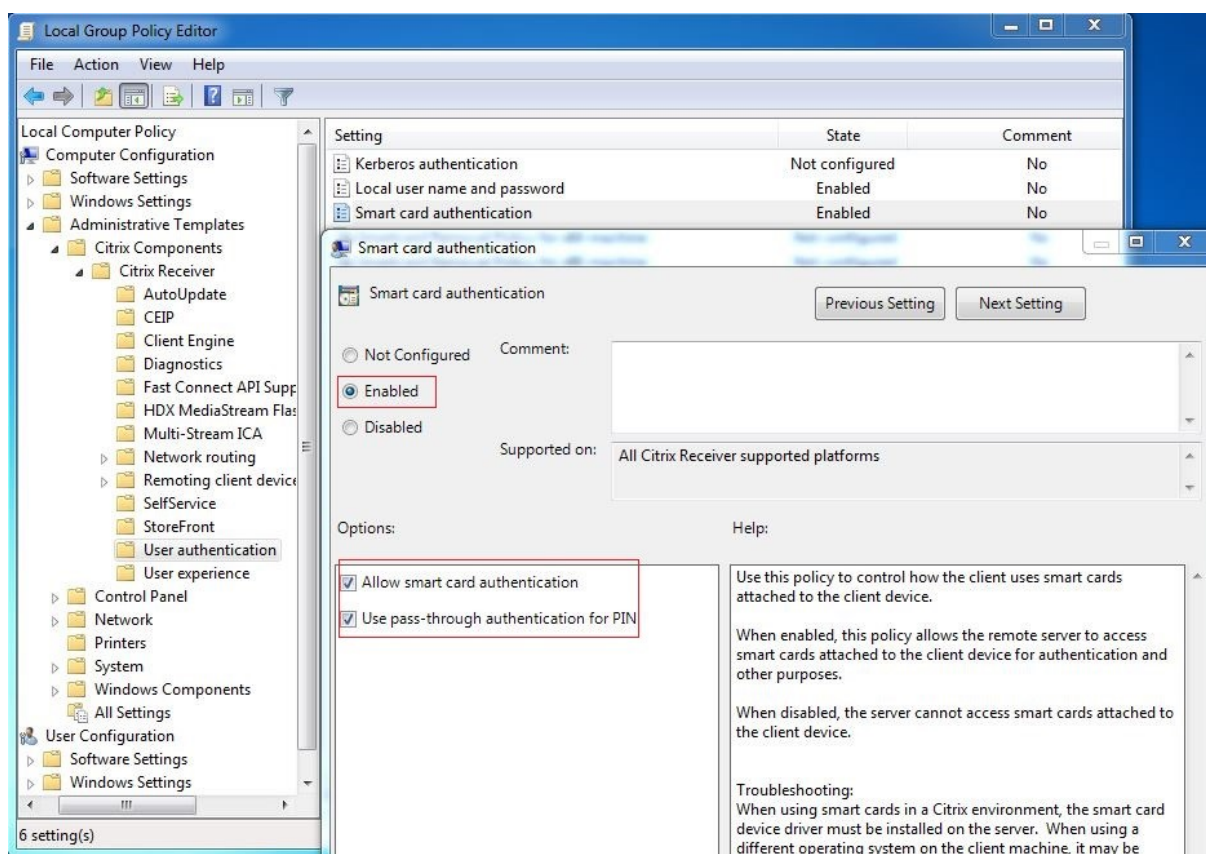
SSSD を使用した場合、変更後の構成ファイルは次のようになります。

```
##PAM-1.0  
# This file is auto-generated.  
# User changes will be destroyed the next time authconfig is run.  
auth      required      pam_env.so  
auth      [success=done ignore=ignore default=die] pam_krb5.so preauth_opt=X509_user_identity=PKCS11:/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so  
auth      sufficient    pam_permit.so  
auth      required      pam_deny.so  
  
account    required      pam_unix.so  
account    sufficient    pam_localuser.so  
account    sufficient    pam_succeed_if.so uid < 1000 quiet  
account    [default=bad success=ok user_unknown=ignore] pam_sss.so  
account    [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore] pam_krb5.so  
account    required      pam_permit.so  
  
session    optional      pam_keyinit.so revoke  
session    required      pam_limits.so  
-session   optional      pam_systemd.so  
#session   optional      pam_oddjob_mkhomedir.so umask=0077  
session    optional      pam_mkhomedir.so umask=0077  
session    [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid  
session    required      pam_unix.so  
session    optional      pam_sss.so  
session    optional      pam_krb5.so
```

(オプション) スマートカードを使用したシングルサインオン

シングルサインオン (SSO) とは、仮想デスクトップや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 の機能を指します。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。Windows VDA と同じ構成方法です。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ときは、次のようにスマートカード認証を有効にします。



### 高速スマートカードログオン

高速スマートカードは、既存の HDX PC/SC ベースのスマートカードリダイレクトの改良版です。遅延が大きい WAN 環境でスマートカードを使用する場合のパフォーマンスが向上しています。詳しくは、「[スマートカード](#)」を参照してください。

Linux VDA は、以下のバージョンの Citrix Workspace アプリで高速スマートカードをサポートしています：

- Citrix Receiver for Windows 4.12
- Windows 向け Citrix Workspace アプリ 1808 以降

クライアントで高速スマートカードログオンを有効にする 高速スマートカードログオンは、VDA ではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連する StoreFront サイトの default.ica ファイルに次のパラメーターを追加します：

```
1 [WFClient]
2 SmartCardCryptographicRedirection=0n
```

クライアントで高速スマートカードログオンを無効にする クライアントで高速スマートカードログオンを無効にするには、関連する StoreFront サイトの default.ica ファイルから **SmartCardCryptographicRedirection** パ

ラメーターを削除します。

## XDPing の実行

上記の手順の完了後、Linux **XDPing** ツールを使用して Linux VDA 環境に一般的な構成の問題がないか確認することもできます。

## 使用状況

スマートカードを使用して **Linux VDA** にログオンする

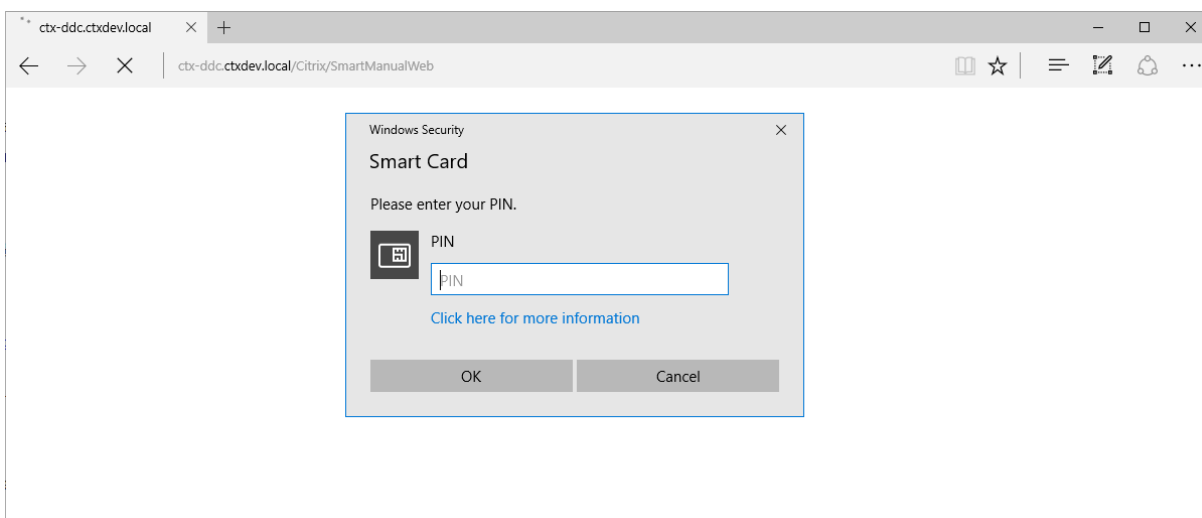
SSO シナリオと非 SSO シナリオの両方で、スマートカードを使用して Linux VDA にログオンできます。

- 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されます。StoreFront で Linux 仮想デスクトップセッションを開始すると、スマートカード認証のために PIN が Linux VDA に渡されます。
- 非 SSO シナリオでは、StoreFront にログオンするために証明書を選択して PIN を入力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開始すると、Linux VDA へのログオンの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 をもう一度入力する必要があります。

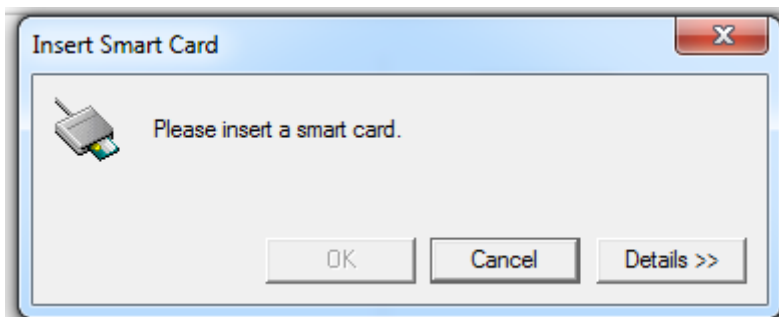
この動作は Windows VDA と同じです。



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ます。スマートカードが接続されていないと再認証は失敗するため、Linux VDA 側にグレーのキャッシュウインドウが表示されてすぐに終了します。この場合、スマートカードの接続を促すメッセージは表示されません。

ただし、StoreFront 側では、セッションに再接続したときにスマートカードが接続されていないと、StoreFront Web により次のような通知が表示されることがあります：



## 制限事項

### Linux ディストリビューションと AD の統合方法の限定的なサポート

- 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：

	Winbind	SSSD	Centrify	Quest
Debian 12.5/11.9	はい	はい	はい	いいえ

	Winbind	SSSD	Centrify	Quest
RHEL 9.4/9.3/9.2	はい	はい	いいえ	はい
RHEL 8.10/8.9/8.8	はい	はい	はい	はい
Rocky Linux 9.4/9.3/9.2	はい	はい	いいえ	いいえ
Rocky Linux 8.10/8.9/8.8	はい	はい	いいえ	いいえ
Ubuntu 22.04/20.04	はい	はい	はい	はい

#### スマートカードの取り出し設定

Linux VDA で次のレジストリキーを編集することで、セッション中にサインインしたユーザーのスマートカードがスマートカードリーダーから取り出されたときに何が起こるかを決定できます:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LocalPolicies\SecurityOptions

値の名前	種類	デフォルト	説明
SCardRemoveOption	REG_DWORD	0x00000000	この設定では、セッション中にサインインしたユーザーのスマートカードがスマートカードリーダーから取り出されたときに何が起こるかを決定できます。オプションは次のとおりです。 0 アクションなし  1 強制ログオフ: スマートカードを取り出すと、ユーザーは自動的にサインアウトされます。



値の名前	種類	デフォルト	説明
SCardRemoveActionDelaySecs	REG_DWORD	15	2 セッションを切断する: スマートカードを取り出すと、ユーザーがサインアウトされずにセッションが切断されます。ユーザーはスマートカードを再挿入して、セッションに後から再接続できます。 取り出しオプションが強制ログオフまたはセッションの切断に設定されている場合、アクションを実行する前にさらに数秒の遅延を指定できます。この設定は、スマートカードが取り出された後にユーザーセッションにメッセージボックスが表示され、指定された秒数ののち、セッションが強制的にログオフ/切断されることを示します。その秒数が経過する前にスマートカードを再挿入すると、セッションは中断されずに継続します

制限事項 Linux VDA は、一度に 1 つのスマートカードのみをサポートします。

他のスマートカードおよび **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のソリューションを提供します。サポート一覧に OpenSC スマートカードのみが表示されますが、他のスマートカードおよび PKCS#11 ライブラリの使用を試すこともできます。特定のスマートカードまたは PKCS#11 ライブラリに切り替えるには:

1. PKCS#11 ライブラリのすべての `opensc-pkcs11.so` インスタンスを置き換えます。
2. PKCS#11 ライブラリからレジストリへのパスを設定するには、次のコマンドを実行します:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v "
  PKCS11LibPath" -d "PATH"
```

**PATH** は PKCS#11 ライブラリ (/usr/lib64/pkcs11/opensc-pkcs11.so など) を参照します。

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。

## 認証が不要なユーザー（匿名ユーザー）のアクセス

July 16, 2024

StoreFront または Citrix Workspace アプリに資格情報を提示せずに、ユーザーが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 ストアがあり、デリバリーグループ内の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:

認証が不要なユーザーによるアクセスは、ドメイン参加済み VDA でのみサポートされます。

セッションの事前起動は、認証されていないユーザーに対してはサポートされていません。セッションの事前起動は、Android 向け Citrix Workspace アプリでもサポートされていません。

### 認証が不要な **StoreFront** ストアの作成

1. Citrix StoreFront 管理コンソールの左ペインで [ストア] ノードを選択して、[操作] ペインの [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て、[このストアへのアクセスを非認証（匿名）ユーザーにのみ許可する] を選択し、[次へ] をクリックします。

詳しくは、「[認証が不要なストアの作成](#)」を参照してください。

### デリバリーグループで認証が不要な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、いくつかのマシンカタログから選択したマシンをグループ化したものです。デリバリーグループ内のアプリケーションとデスクトップを使用できるユーザーを指定すると、認証されていないユーザーにアクセスを許可できます。詳しくは、「[デリバリーグループの作成](#)」を参照してください。

## 認証が不要な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

認証が不要なユーザーセッションにはデフォルトで 10 分のアイドルタイムアウトが設定され、セッションを切断すると自動的にログオフされます。レジストリ設定 **AnonymousUserIdleTime** でカスタムのアイドルタイムアウトを構成できます。たとえば、カスタムの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：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix" -v AnonymousUserIdleTime -d 0
  x00000005
```

## 認証が不要な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人数を設定するには、レジストリキー **MaxAnonymousUserNumber** を使用します。この設定により、単一の Linux VDA で同時に実行される認証が不要なユーザーセッション数が制限されます。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：

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

### 重要：

認証が不要なユーザーセッション数を制限します。同時に起動されるセッション数が非常に多い場合、VDA で使用できるメモリの不足などの問題を引き起こすことがあります。

## トラブルシューティング

認証が不要なユーザーセッションを構成するときは、次の点を考慮してください：

- 認証が不要なユーザーセッションにログオンできませんでした。

レジストリが次を含むように更新されたことを確認します (0 に設定)。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg read -k "HKLM\System\CurrentControlSet
  \Control\Citrix" -v MaxAnonymousUserNumber
```

**nscd** サービスが実行中で、**passwd** キャッシュを有効にするように設定されていることを確認します：

```
1 ps uax | grep nscd
2 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** に設定してから、**nscd** サービスを再起動します。設定の変更後に、Linux VDA の再インストールが必要となる場合があります。

- **KDE** でロック画面のボタンが認証不要のユーザーセッション中に表示されます。

デフォルトでは、ロック画面のボタンとメニューは、認証が不要なユーザーセッションでは無効になっています。ただし、KDE でなお表示されることがあります。KDE でロック画面のボタンとメニューを特定のユーザーに対して無効にするには、構成ファイル **\$Home/.kde/share/config/kdeglobals** に次の行を加えます。例：

```
1 [KDE Action Restrictions]
2 action/lock_screen=false
```

ただし、KDE Action Restrictionsパラメーターが、グローバルワイドなkdeglobalsファイル(/usr/share/kde-settings/kde-profile/default/share/config/kdeglobalsなど) で不変に設定されている場合、このユーザー設定は効果がありません。

この問題を解決するには、システムワイドな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、システムワイドな構成を直接使用して、ロック画面のボタンとメニューを無効にします。KDE 構成について詳しくは、「[KDE System Administration/Kiosk/Keys](#)」のページを参照してください。

## ファイル

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [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

July 16, 2024

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して貼り付けることができます。この機能には、Citrix Virtual Apps and Desktops 2006以降およびWindows向けCitrix Workspaceアプリ1903以降が必要です。

ファイルを正常にコピーして貼り付けるには、次のことを確認してください：

- ファイルの最大数が20を超えていない。
- 最大ファイルサイズが200MBを超えていない。
- Nautilusファイルマネージャーは、Linux VDAをインストールしたマシンで使用できます。
- ファイル名にはASCII文字のみが含まれ、特殊文字は含まれていない。

## サポートされている **Linux** ディストリビューション

ファイルのコピーと貼り付け機能は、Linux VDA がサポートするすべての Linux ディストリビューションで使用できます。

## 関連ポリシー

以下は、この機能の構成に関連したクリップボードポリシーです。クリップボードポリシーについて詳しくは、「[ポリシーサポート一覧](#)」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
- クリップボードの選択更新モード
- プライマリ選択更新モード

### 注:

ファイルのコピーと貼り付け機能を無効にするには、Citrix Studio で [クライアントクリップボードリダイレクト] ポリシーを [禁止] に設定します。

## 制限事項

- 切り取りはサポートされていません。ファイルの切り取り要求はコピー操作として扱われます。
- ドラッグアンドドロップはサポートされていません。
- ディレクトリのコピーはサポートされていません。
- ファイルのコピーと貼り付けは、順番に実行する必要があります。前のファイルが正常にコピーされて貼り付けられた後でのみ、次のファイルをコピーできます。

## ファイル転送

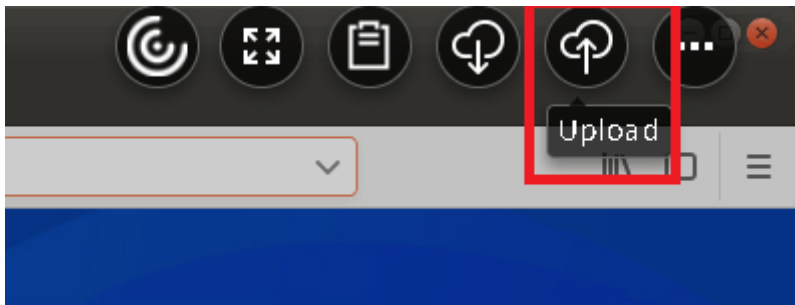
July 16, 2024

Linux VDA とクライアントデバイス間のファイル転送がサポートされています。この機能は、クライアントデバイスが HTML5 の sandbox 属性をサポートする Web ブラウザーを実行している場合に使用できます。HTML5 の sandbox 属性は、ユーザーが HTML5 向けまたは Chrome 向け Citrix Workspace アプリを使用して仮想デスクトップやアプリにアクセスできるようにします。

### 注:

ファイル転送機能は HTML5 向けおよび Chrome 向け Citrix Workspace アプリで使用できます。

公開アプリおよびデスクトップセッション内で、ファイル転送機能によって Linux VDA およびクライアントデバイス間のファイルのアップロードおよびダウンロードが可能になります。ファイルをクライアントデバイスから Linux VDA にアップロードするには、Citrix Workspace アプリのツールバーの [アップロード] アイコンをクリックして、ファイルダイアログから目的のファイルを選択します。ファイルを Linux VDA からクライアントデバイスにダウンロードするには、[ダウンロード] アイコンをクリックします。アップロードまたはダウンロード中にファイルを追加できます。一度に最大 10 個のファイルを転送できます。



注:

Linux VDA とクライアントデバイス間でファイルのアップロードおよびダウンロードを実行するには、Citrix Workspace アプリのツールバーを有効にしてください。

ファイルをドラッグアンドドロップできるバージョンの Citrix Workspace アプリを使用できます。

自動ダウンロードはファイル転送の拡張機能です。VDA の自分のデバイスに保存ディレクトリにファイルをダウンロードまたは移動すると、クライアントデバイスに自動的に転送されます。

注:

自動ダウンロードでは、**ChromeOS/HTML5** 向け **Citrix Workspace** アプリのファイル転送および **ChromeOS/HTML5** 向け **Citrix Workspace** アプリのファイルダウンロードポリシーを [許可] に設定する必要があります。

以下は自動ダウンロードの使用例です:

- ファイルを自分のデバイスに保存にダウンロードする場合

公開デスクトップと Web ブラウザーアプリのセッションで Web サイトからダウンロードしたファイルは、VDA の自分のデバイスに保存ディレクトリに保存して、クライアントデバイスに自動的に転送することができます。自動ダウンロードを機能させるには、Web ブラウザーのデフォルトのセッション内ダウンロードディレクトリを自分のデバイスに保存に設定し、HTML5 向けまたは Chrome 向け Citrix Workspace アプリを実行する Web ブラウザーでローカルのダウンロードディレクトリを設定します。

- ファイルを自分のデバイスに保存に移動またはコピーする場合

公開デスクトップセッションで目的のファイルを選択し、クライアントデバイスで使用するために自分のデバイスに保存ディレクトリに移動またはコピーします。

## ファイル転送のポリシー

デフォルトでは、ファイル転送は有効になっています。Citrix Studio を使って、[ユーザー設定] > [ICA\ファイルリダイレクト] にあるこれらのポリシーを変更します。ファイル転送のポリシーを使用する場合は、次の点を検討してください:

**ChromeOS/HTML5 向け Citrix Workspace** アプリのファイル転送 - Citrix Virtual Apps and Desktops セッションまたは Citrix DaaS セッションとエンドユーザーのデバイス間でのユーザーによるファイル転送を許可または拒否します。

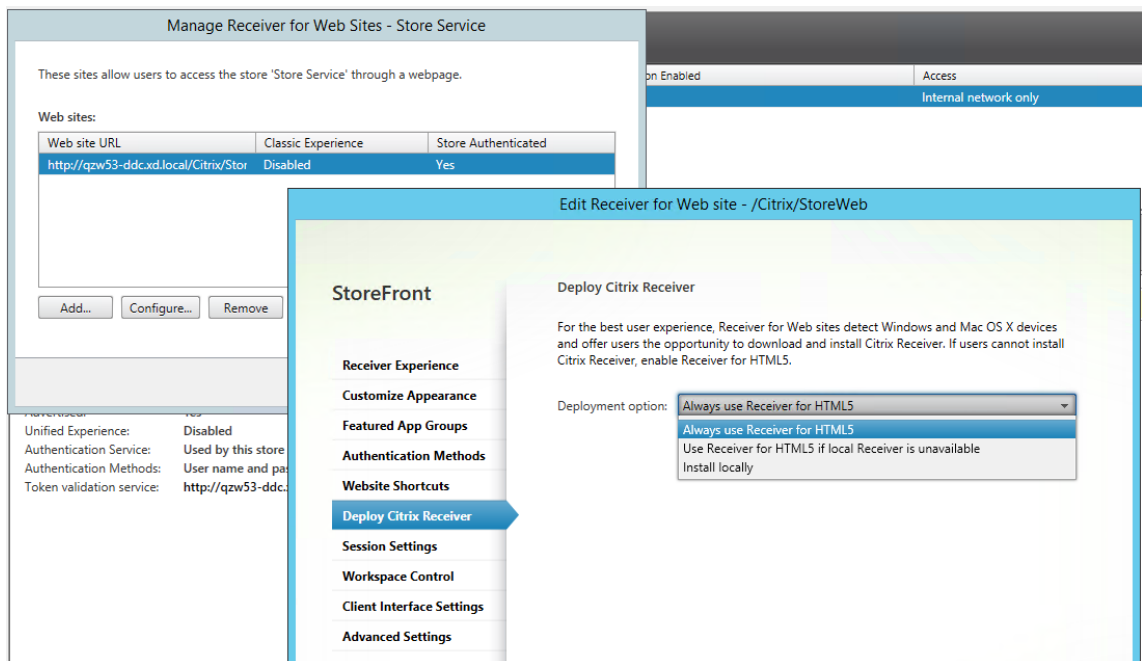
**ChromeOS/HTML5 向け Citrix Workspace** アプリのファイルアップロード - ユーザーによるファイルのアップロードを許可または拒否します。ユーザーのデバイスから Citrix Virtual Apps and Desktops セッションまたは Citrix DaaS セッションへのアップロードが対象になります。

**ChromeOS/HTML5 向け Citrix Workspace** アプリのファイルダウンロード - ユーザーによるファイルのダウンロードを許可または拒否します。Citrix Virtual Apps and Desktops セッションまたは Citrix DaaS セッションからユーザーのデバイスへのダウンロードが対象になります。

## 使用状況

**HTML5 向け Citrix Workspace** アプリで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。
2. Citrix Studio で前述のファイル転送ポリシーから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、[Receiver for Web サイトの管理] ノード、[常に **Receiver for HTML5** を使用] オプションを選択して、Citrix Receiver for HTML5 を有効にします。



4. 仮想デスクトップまたは Web ブラウザーアプリのセッションを開始します。Linux VDA とクライアントデバイス間で 1 つまたは複数のファイル転送を実行します。

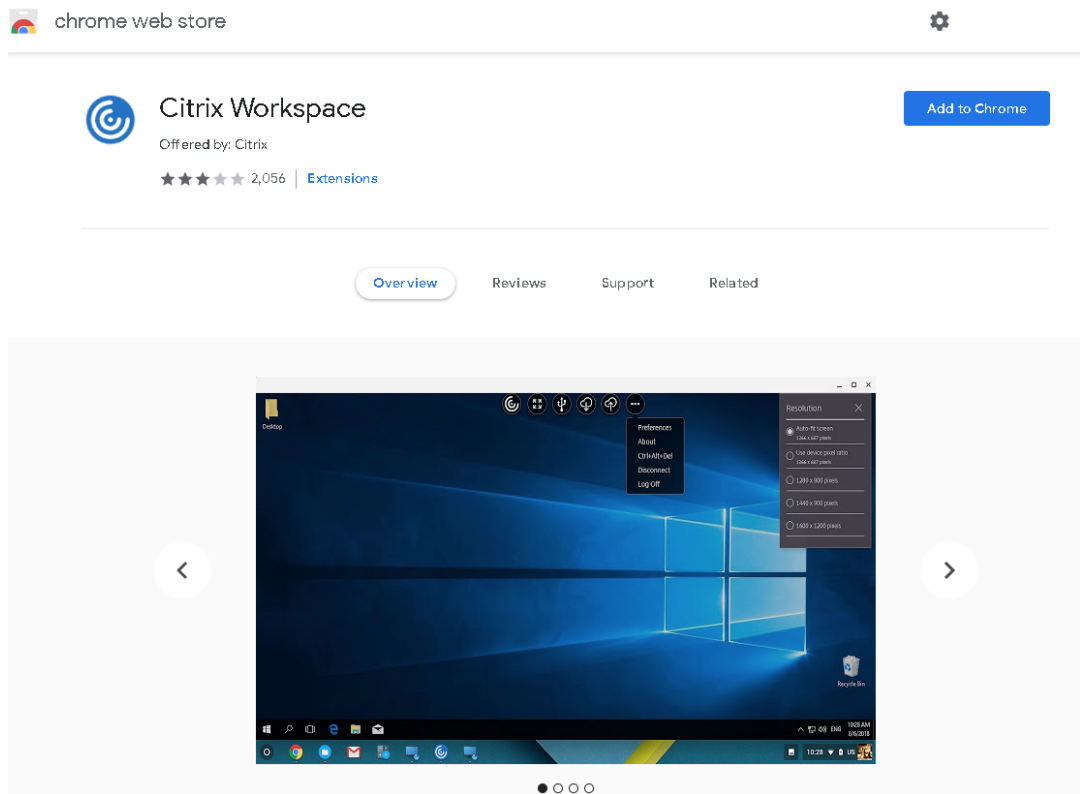
**Chrome** 向け **Citrix Workspace** アプリでファイル転送機能を使用するには:

1. 前述のファイル転送ポリシーから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。

Chrome アプリページから Chrome 向け Citrix Workspace アプリを追加済みの場合は、この手順を省略します。

- 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) 検索結果で Chrome ウェブストアへの URL をクリックすると、Citrix Workspace アプリを入手できます。





- c) [**Chrome** に追加] を選択して、Citrix Workspace アプリを Google Chrome に追加します。
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. StoreFront ストアの URL を入力して接続します。  
既に入力済みの場合はこの手順を省略します。
  5. 仮想デスクトップまたはアプリのセッションを開始します。Linux VDA とクライアントデバイス間で 1 つまたは複数のファイル転送を実行します。

## グラフィック

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [自動 DPI スケーリング](#)
- [クライアントのバッテリー状態の表示](#)
- [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 のプログレッシブ表示](#)

## 自動 DPI スケーリング

August 20, 2024

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。

この機能に関連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 の DPI マッチングを有効にする必要があります。Windows 向け Citrix Workspace アプリの場合は、[いいえ、ネイティブ解像度を使用します] オプションがオンになっていることを確認します。Windows 向け Citrix Workspace アプリの DPI スケーリングの構成について詳しくは、「[DPI スケーリング](#)」を参照してください。
- この機能をマルチモニターシナリオで機能させるには、各モニターを同じ DPI 設定で構成する必要があります。DPI が混在している状況はサポートされません。異なる DPI 設定で複数のモニターが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 クラシック、および KDE で有効になっています。KDE または MATE を使用する場合は、次の点を考慮してください。
  - KDE デスクトップ環境で実行されている Linux 仮想デスクトップの場合：
    - \* KDE Plasma 5 以降を使用することをお勧めします。
    - \* セッションの実行中にクライアント側で DPI 設定を変更するには、ユーザーがログオフしてから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は 1 と 2 のみがサポートされています。
    - \* セッションの実行中にクライアント側で DPI 設定を変更するには、ユーザーがログオフしてから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（100% や 200% など）。クライアント側で構成されたスケールファクターが小数型の場合、仮想セッション 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 値が 100% に変更されます。

### クライアント側のスケールファクタ

—	リモートセッション DPI	ディスプレイの拡大縮小
174% 以下	96 (1 x 96)	100%
175%~274%	192 (2 x 96)	200%
275%~399%	288 (3 x 96)	300%
400% 以上	384 (4 x 96)	400%

注:

Linux VDA は、MATE デスクトップの最大 200% の拡大をサポートします。

### クライアントのバッテリー状態の表示

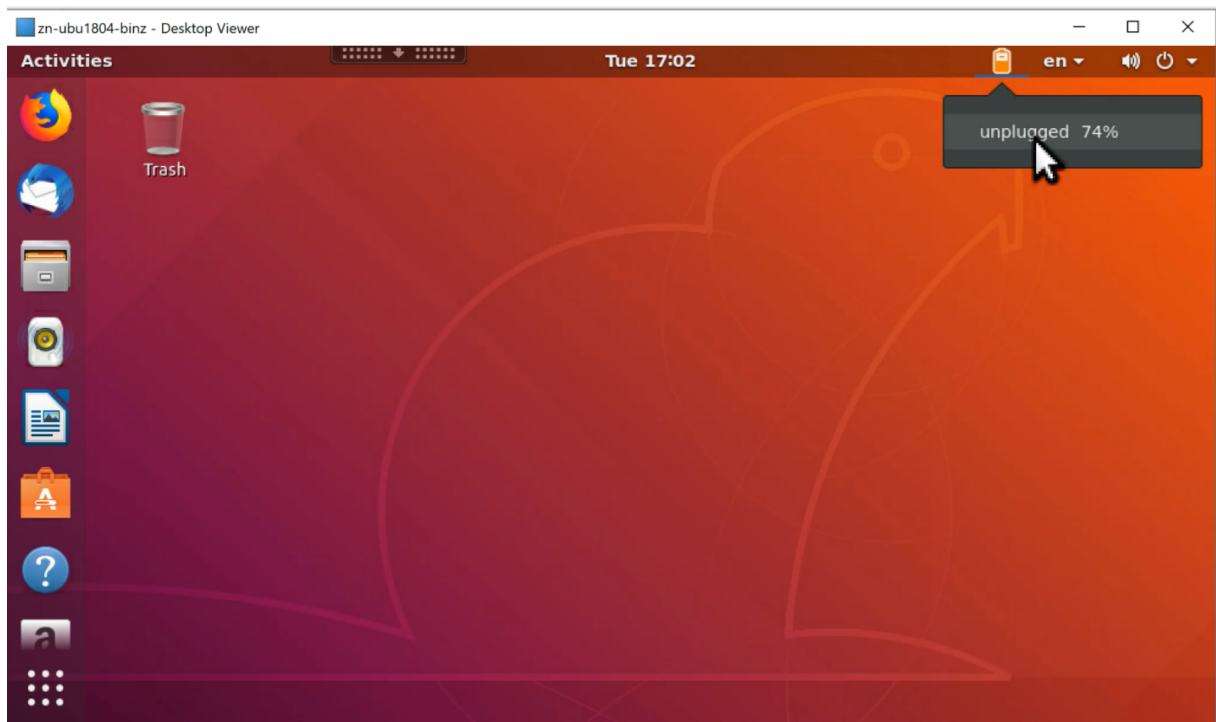
July 16, 2024

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ます。この機能はデフォルトで有効になっており、次のバージョンの Citrix Workspace アプリで使用できます:










- iOS 向け Citrix Workspace アプリ
- Linux 向け Citrix Workspace アプリ
- Mac 向け Citrix Workspace アプリ (バージョン 2204.1 はサポートされていません)
- Windows 向け Citrix Workspace アプリ (バージョン 2204.1 はサポートされていません)

### 概要

ユーザーが仮想デスクトップを開くと、Linux システムトレイにバッテリーアイコンが表示されます。バッテリーアイコンは、クライアントデバイスのバッテリー状態を示します。バッテリー残量のパーセンテージを確認するには、バッテリーアイコンをクリックします。例として、以下のスクリーンショットを参照してください:



異なるバッテリーアイコンは、異なるバッテリー状態を示します。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	充電状態	バッテリー残量のレベル	バッテリー残量のパーセンテージ
	充電中（「+」記号で表示）	高（緑色で表示）	 =80%
	充電中（「+」記号で表示）	中（黄色で表示）	20% 以上 80% 未満
	充電中（「+」記号で表示）	低（赤色で表示）	< 20%
	充電していない（「-」記号で表示）	高（緑色で表示）	 =80%
	充電していない（「-」記号で表示）	中（黄色で表示）	20% 以上 80% 未満
	充電していない（「-」記号で表示）	低（赤色で表示）	< 20%
	不明	不明	不明

---

バッテリーアイコン	充電状態	バッテリー残量のレベル	バッテリー残量のパーセンテージ
-----------	------	-------------	-----------------

---

## 構成

クライアントのバッテリー状態の表示は、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：

```
1 /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：

```
1 /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、Mobile Receiver Virtual Channel (MRVC) を共有してクライアントのバッテリー状態を表示する [ソフトキーボード機能](#) に影響を与えます。

使用するディストリビューションに応じて、機能を有効にするための追加手順を実行します。詳しくは、「[システムトレイの有効化](#)」セクションを参照してください。

## グラフィックの構成と微調整

August 20, 2024

ここでは、Linux VDA のグラフィックの構成と微調整について説明します。

詳しくは、「[システム要件](#)」および「[インストールの概要](#)」を参照してください。

## 構成

### 3D 画像ワークロードの最適化

この設定では、グラフィックを多用するワークロードに合わせて適切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のワークロードが大きいユーザーに対して、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。その他の設定がこのポリシーのデフォルト設定を明示的に上書きする場合、そちらが優先されます。

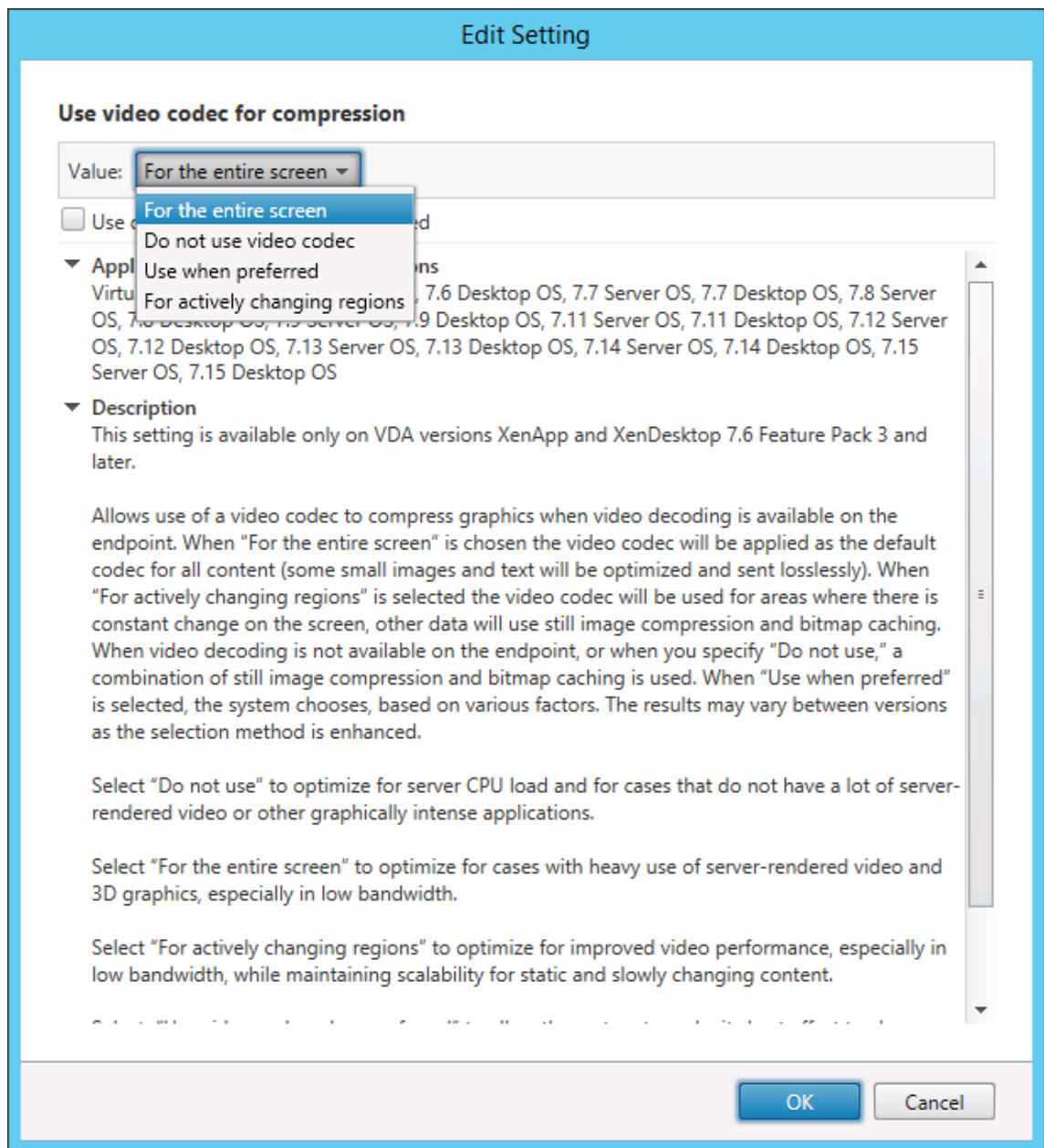
デフォルトでは、**[3D 画像ワークロードの最適化]** は無効になっています。

### ビデオコーデックでの圧縮

Thinwire は、Linux VDA で使用されているディスプレイリモートテクノロジーです。このテクノロジーを使用すると、あるマシンで生成されたグラフィックが、通常はネットワークを経由して、別のマシンに転送され、表示されます。

[\[圧縮にビデオコーデックを使用する\]](#) グラフィックポリシーでは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して次のオプションを提供します：

- [可能であれば使用]。この設定がデフォルトです。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、デスクトップの一般的なワークロードで、スケーラビリティ、帯域幅、および優れた画質の点で、確実に最適化されます。
- [画面全体に使用]。特に 3D グラフィックを多用する事例で、Thinwire を全画面 H.264 または H.265 を使用して配信して、ユーザーエクスペリエンスと帯域幅の改善を最適化します。[画面全体に使用] が選択されている場合、または [可能であれば使用] が選択され、[\[3D 画像ワークロードの最適化\]](#) が有効な場合、[セッションウォーターマーク](#)がサポートされます。
- [領域をアクティブに変更]。Thinwire のアダプティブ表示テクノロジーは、動画（ビデオ、3D インモーション）を識別します。画像が動く画面の部分でのみ AV1、H.265 または H.264 を使用します。グラフィックの圧縮に AV1、H.265 または H.264 ビデオコーデックを選択的に使用することにより、HDX Thinwire は頻繁に更新される画面の部分を検出してエンコードすることができます。静止画圧縮（JPEG、RLE）とビットマップキャッシングは、テキストや写真画像など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、帯域幅の消費が低い状態で、無損失テキストや高品質画像を組み合わせた品質の高いビデオコンテンツを視聴できます。[表示品質] ポリシーが [常に無損失] または [操作時は低品質] に設定されている場合、AV1 や H.265 を選択的に使用することはサポートされません。



次の視覚表示ポリシー設定など、いくつかの他のポリシー設定は、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使用できます。

- 簡素なグラフィックに対する優先的色の解像度
- ターゲットフレーム数
- 表示品質

## AV1/H.265/H.264 ハードウェアエンコーディング

[ビデオコーデックにハードウェアエンコーディングを使用します] ポリシーにより、GPU ハードウェアアクセラレーション (搭載している場合) を利用して、画面要素をビデオコーデックで圧縮できます。GPU ハードウェアアクセラレーションにより、ハードウェアリソースの使用率が最適化され、1 秒あたりのフレーム数/秒 (FPS) のパフォーマンスが大幅に向上します。

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する] ポリシーで設定されたすべてのグラフィックスモードをカバーします。

- 可能であれば使用
- 画面全体に使用
- 領域をアクティブに変更

ハードウェアビデオ圧縮を有効にするには、次の手順を実行します。

1. ビデオコーデックにハードウェアエンコーディングを使用しますポリシーを [有効] に設定します。
2. [圧縮にビデオコーデックを使用する] を [可能であれば使用]、[画面全体に使用]、または [領域をアクティブに変更] 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ます。

使用できるようにするには、AV1 または H.265 ビデオコーデックが、VDA および Citrix Workspace アプリの両方でサポートされ、かつ有効になっている必要があります。AV1 は、コーデックネゴシエーション時に H.265 および H.264 よりも優先されます。AV1 がサポートされていない場合は、H.265 がネゴシエートされます。AV1 と H.265 がともにサポートされていない場合、セッションは H.264 ビデオコーデックの使用に戻ります。GPU ハードウェアが利用可能でない場合、VDA はソフトウェアビデオコーデックを使用して、CPU ベースのエンコーディングにフォールバックします。

## AV1 ハードウェアエンコーディングの要件

### VDA

- VDA: 2311 以降
- GPU: **NVIDIA Ada Lovelace** 以降 (NVIDIA GPU がサポートするビデオコーデックのマトリックスについては、<https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new> の NVIDIA ドキュメントを参照してください。)
- NVIDIA グラフィックスドライバー 522.25 以降 (ビデオコーデック SDK v12.0)

### クライアント

- Windows 向け Citrix Workspace アプリ 2305 以降
- AV1 デコードをサポートするクライアント GPU:
  - NVIDIA Ampere 以降



- Intel 第 11 世代/Arc 以降
- AMD Radeon RX 6000 / Radeon Pro W6000 シリーズ (RDNA2) 以降

## H.265 ハードウェアエンコーディングの要件

### クライアント

- Citrix Receiver for Windows 4.10~4.12
- Windows 向け Citrix Workspace アプリ 1808 以降

クライアントで H.265 ハードウェアエンコーディングを有効にするには、「[H.265 ビデオエンコーディング](#)」を参照してください。

## H.265/H.264 無損失圧縮

NVIDIA GPU による HDX 3D PRO ハードウェアアクセラレーションでは、H.265/H.264 無損失圧縮が利用可能です。H.265 無損失圧縮には、Windows 向け Citrix Workspace アプリ 2305 以降が必要です。H.264 無損失圧縮には、次のクライアントが必要です。

- Windows 向け Citrix Workspace アプリ 2303 以降
- Apple M1 チップを使用している Mac 向け Citrix Workspace アプリ 2301 以降

H.265/H.264 無損失圧縮を有効にするには、次の手順を実行します。

1. ビデオコーデックにハードウェアエンコーディングを使用しますポリシーを [有効] に設定します。
2. [圧縮にビデオコーデックを使用する](#)ポリシーを [画面全体に使用] に設定します。
3. 表示品質ポリシーを [常に無損失] または [操作時は低品質] に設定します。

### 視覚的無損失の圧縮を使用する

視覚的無損失の圧縮を使用するポリシーにより、グラフィックに対して、真の無損失圧縮の代わりに視覚的に無損失の圧縮を使用できるようになります。視覚的無損失では、真の無損失よりもパフォーマンスは向上しますが、見た目にはわからない程度の軽微な損失が発生します。この設定により、[表示品質] 設定の値の使用方法が変更されます。

視覚的無損失の圧縮を使用するポリシーは、デフォルトで無効になっています。視覚的無損失圧縮を有効にするには、[視覚的無損失の圧縮を使用する] を [有効] に設定し、表示品質ポリシーを [操作時は低品質] に設定します。

圧縮にビデオコーデックを使用するポリシーが [ビデオコーデックを使用しない] に設定されている場合、視覚的無損失圧縮が静止画像エンコーディングに適用されます。圧縮にビデオコーデックを使用するポリシーが [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、視覚的無損失圧縮が H.264 エンコーディングに適用されます。

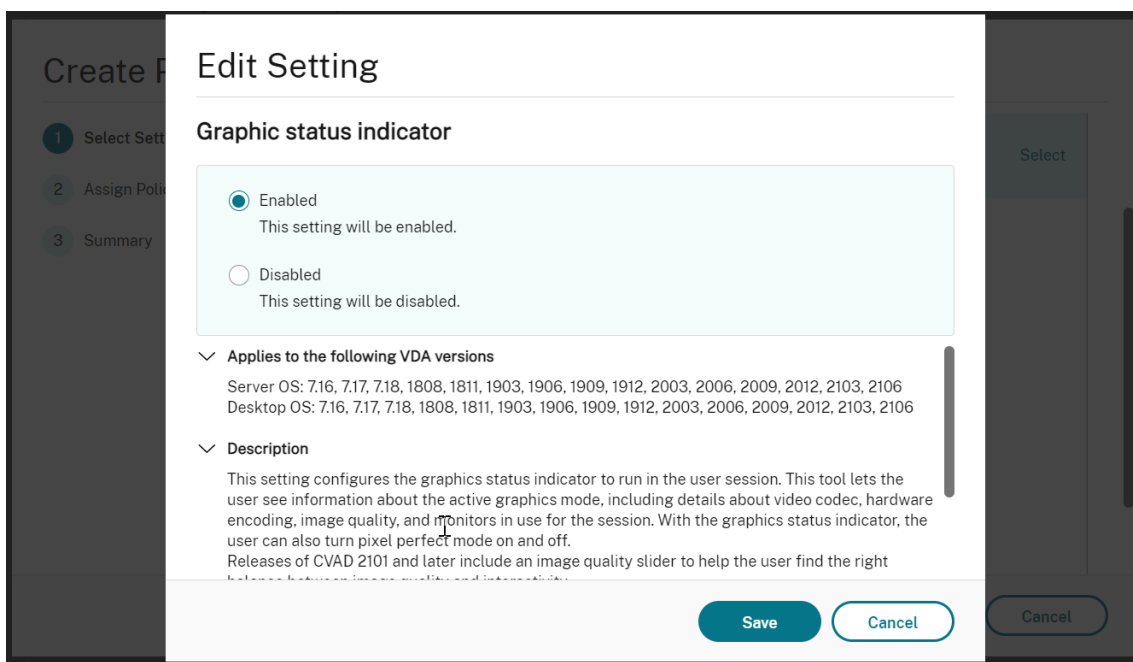
[表示品質] および [圧縮にビデオコーデックを使用する] のポリシー設定について詳しくは、「[視覚表示のポリシー設定](#)」と「[グラフィックのポリシー設定](#)」を参照してください。

### 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状態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。スライダーは、画質とインタラクティブ性のバランスを適切に調整するのに役立ちます。

スライダーを使用するには、次の手順を実行します：

1. Citrix Studio で [グラフィック状態インジケータ] ポリシーを有効にします。



2. 端末を開き、`ctxslider` コマンドを実行します。スライダーの UI が表示されます。

注：

- [表示品質] ポリシーを [常は無損失] に設定した場合、または [操作時は低品質] に設定した場合、スライダーの UI は表示されません。
- スライダー UI は、ターミナルとシステムトレイの両方から起動できます。



次の選択肢が利用可能になりました:

- 画質を変更するには、スライダーを動かします。スライダーは 0 から 9 まで動かすことができます。
- システム定義の設定を使用するには、[システムが決定する] を選択します。
- 無損失モードに切り替えるには、[完全に無損失] を選択します。

帯域幅推定に基づいて平均ビットレートを調整する

HDX 3D Pro ハードウェアエンコーディングが Citrix で拡張され、帯域幅推定に基づいて平均ビットレートを調整できます。

HDX 3D Pro ハードウェアエンコーディングを使用中の場合、VDA がネットワーク帯域幅を断続的に推定でき、エンコードされたフレームのビットレートを適宜調整できます。この新しい機能では、鮮明さと滑らかさのバランスを調整す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:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  DisableReconfigureEncoder" -d "0x00000001" --force
```

この機能だけでなく、以下のコマンドを実行することでも鮮明さと滑らかさのバランスを調整できます。

**AverageBitRatePercent** および **MaxBitRatePercent** パラメータは、帯域幅使用の割合を設定します。設定した値が大きいほど、グラフィックの鮮明さが向上し滑らかさが低下します。推奨設定範囲は 50~100 です。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  AverageBitRatePercent" -d "90" --force
2
3 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、画面が静止状態の場合、新しいフレームが送信されることがないため、最新のフレームは低品質状態のままです。鮮明さのサポートでは、最新のフレームを最高品質で再構成し、すぐに送信することでこの問題に対応します。

Linux VDA Thinwire でサポートされているポリシーをすべて示す一覧については、「[ポリシーサポート一覧](#)」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詳しくは、[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。全体的な CPU 消費量の負荷はわずかに大きくなります。この機能はデフォルトでは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:

```
1 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の確認

次のコマンドを実行して、使用されているグラフィックモードを確認します (**0** は TW+ を、**1** は全画面ビデオコーデックを意味します):

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

次の内容に類似した結果が出力されます:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "GraphicsMode"-d "0x00000000"--force
```

### AV1 が使用中であることの確認

注:

現在のセッションでどのビデオコーデックが使用されているかを確認するには、以下のコマンドを実行するか、システムトレイでグラフィック状態を確認します。

AV1 が使用中であることを確認するために、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、**1** は使用中であることを意味します):

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep AV1
```

たとえば、次の内容に類似した結果が出力されます:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "AV1"-d "0x00000000"--force
```

### H.265 が使用中であるかどうかの確認

全画面 H.265 が使用中であることを確認するために、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、**1** は使用中であることを意味します):

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

たとえば、次の内容に類似した結果が出力されます:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H265"-d "0x00000000"--force
```

**H.264** が使用中であるかどうかの確認

H.264 が使用中であることを確認するために、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、**1** は使用中であることを意味します):

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

たとえば、次の内容に類似した結果が出力されます:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"  
-v "H264"-d "0x00000000"--force
```

**YUV** エンコーディングスキームが使用中であるかどうかの確認

YUV エンコーディングスキームが使用中であることを確認するために、次のコマンドを実行します (**0** は YUV420、**1** は YUV422、**2** は YUV444 を意味します):

注:

ビデオコーデックが使用中の場合のみ、**YUVFormat** の値に意味があります。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

たとえば、次の内容に類似した結果が出力されます:

```
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中であることの確認

YUV444 ソフトウェアエンコーディングが使用中であることを確認するために、次のコマンドを実行します。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
```

YUV444 が使用中の場合、次の内容に類似した結果が出力されます:

```
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の確認

HDX 3D Pro が有効になっていることを確認するためには、次のコマンドを実行します：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep ProductEdition
2
3 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep StackSessionMode
4
5 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep 3DPro
```

HDX 3D Pro が有効になっている場合、次の内容に類似した結果が出力されます：

```
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 ライブラリが読み込ま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。次の内容に類似した結果が出力されます：

```
1 Tue Apr 12 10:42:03 2016
2 +-----+
3 | NVIDIA-SMI 361.28      Driver Version: 361.28      |
4 |-----+-----+-----+-----+-----+-----+-----+-----+
5 | GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile
6 | Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util
7 | Compute M. |
8 |=====+=====+=====+=====+=====+=====+=====+=====+
9 |    0  GRID K1              Off | 0000:00:05.0    Off |
10 | N/A   42C    P0           14W / 31W | 207MiB / 4095MiB |      8%
11 | Default |
12 +-----+-----+-----+-----+-----+-----+-----+-----+
13 | Processes:                                             GPU
14 |   Memory |
15 | GPU      PID  Type  Process name
16 | Usage    |
17 |=====+=====+=====+=====+=====+=====+=====+=====+
18 |    0      2164  C+G  /usr/local/bin/ctxgfx
19 | 106MiB |
```

```

17 |      0      2187      G      Xorg
      85MiB |
18 +-----+

```

### 3D Pro のハードウェアエンコーディングが使用中であることの確認

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、**1** は使用中であることを意味します）:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
```

3D Pro が使用中の場合、次の内容に類似した結果が出力されます:

```
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が正しくインストールされていることの確認

NVIDIA GRID グラフィックドライバ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。次の内容に類似した結果が出力されます:

```

1 +-----+
2 | NVIDIA-SMI 352.70      Driver Version: 352.70      |
3 |-----+-----+
4 | GPU Name          Persistence-M| Bus-Id        Disp.A | Volatile
5 | Fan Temp Perf     Pwr:Usage/Cap|               Memory-Usage | GPU-Util
6 | Compute M.       |               |
7 |====+====+====+====+====+====+====+====+====+====+====+====+====+====+
8 |  0  Tesla M60           Off | 0000:00:05.0   Off |
9 | N/A  20C   P0          37W / 150W |      19MiB /  8191MiB |      0%
10 | Default |
11 +-----+-----+
12 | Processes:                                             GPU
13 |   Memory |
14 | GPU      PID  Type  Process name
15 | Usage    |
16 |-----+-----+
17 | No running processes found

```





## HDX 画面共有

July 16, 2024

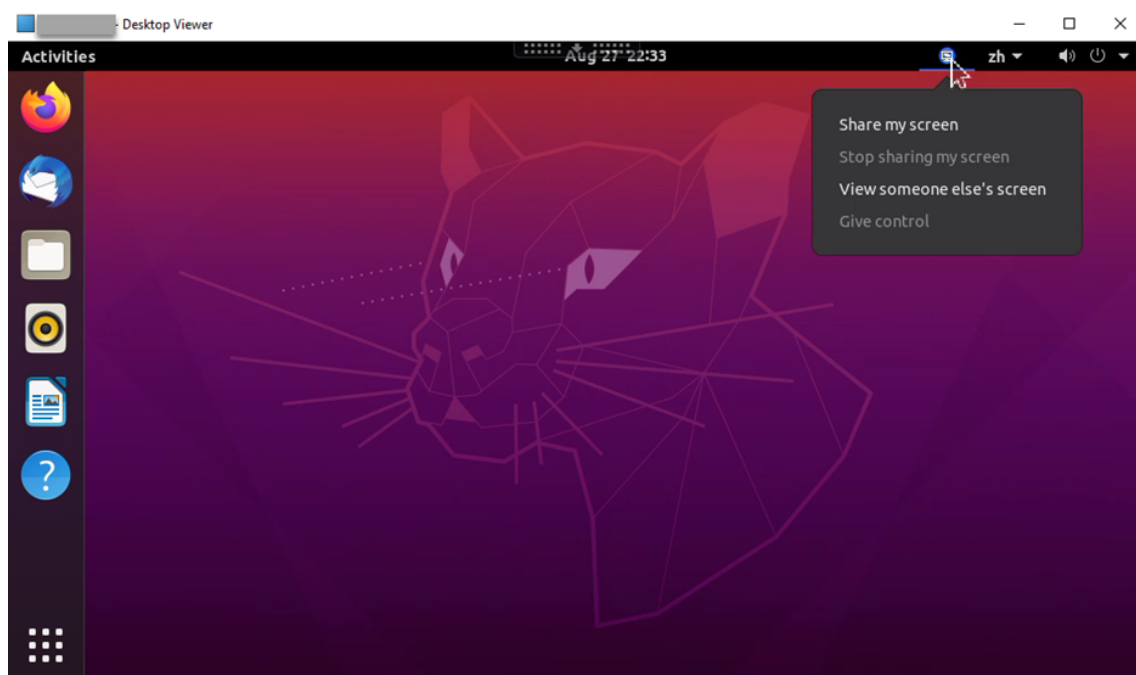
### 概要

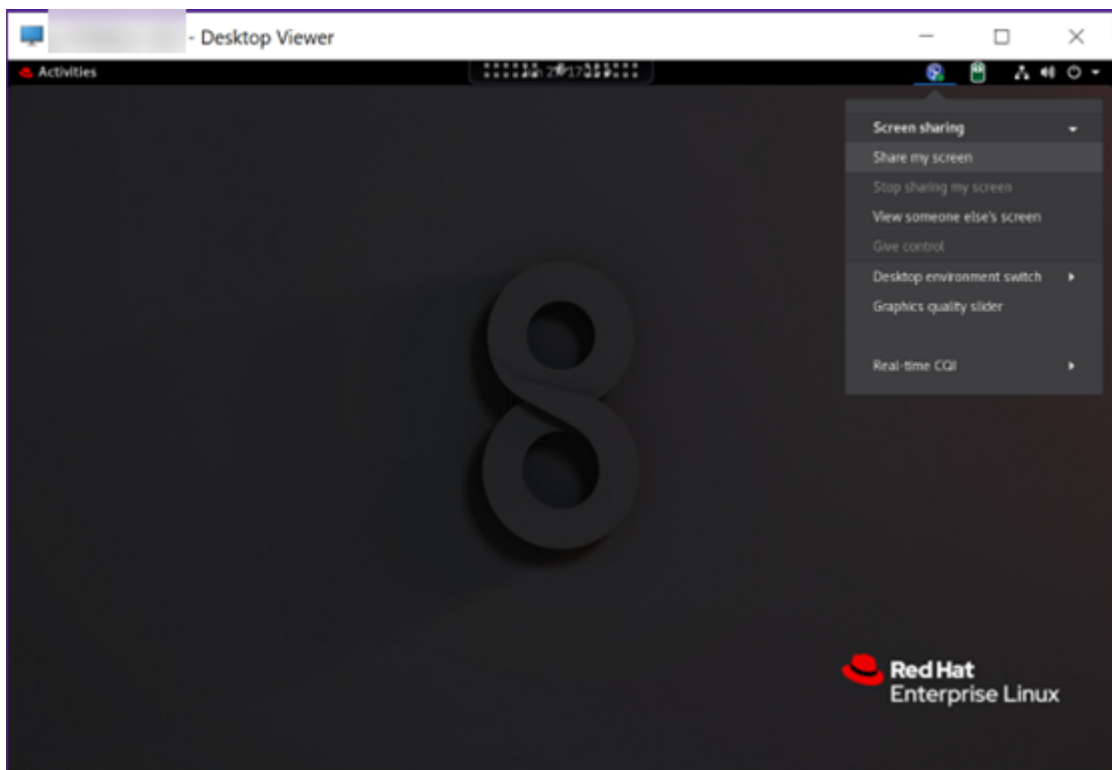
Linux VDA では、仮想デスクトップの画面をほかの仮想デスクトップのセッションユーザーと共有することができます。

次の例では、画面を共有して他の人の画面を表示する手順について説明します。

画面を共有する手順：

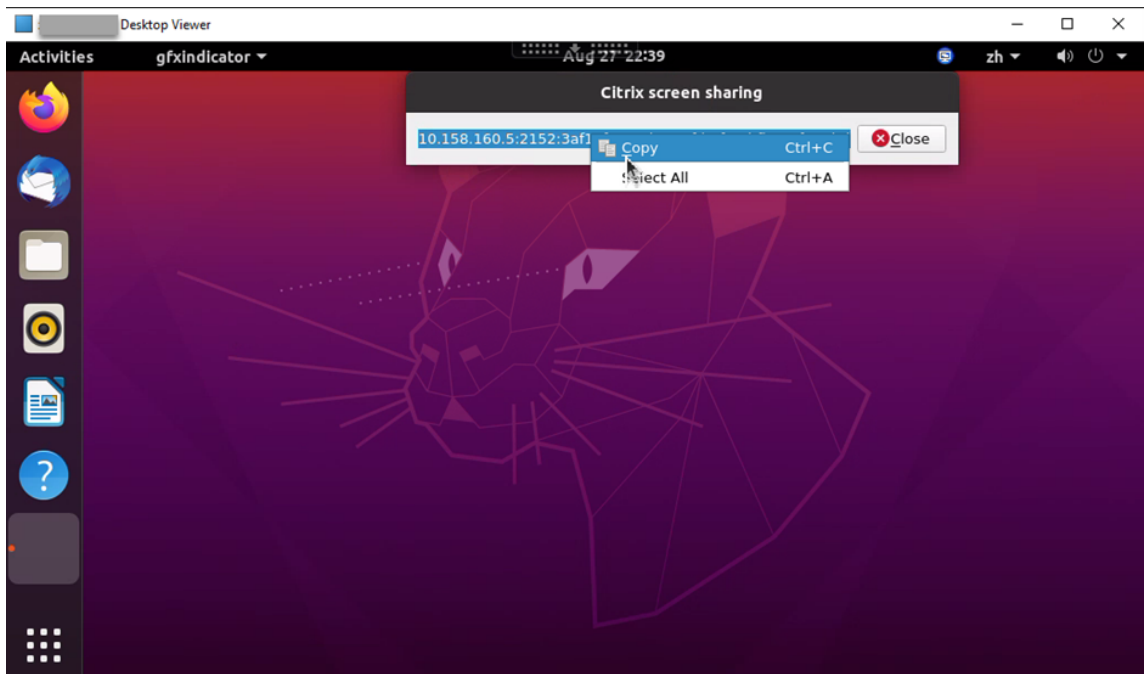
1. 仮想デスクトップの通知領域で、次のシステムトレイアイコンをクリックして [画面共有] > [自分の画面を共有] を選択します。





2. [コピーして閉じる] をクリックします。

画面の共有を停止して再開するまで、現在の画面共有コードは保持されます。

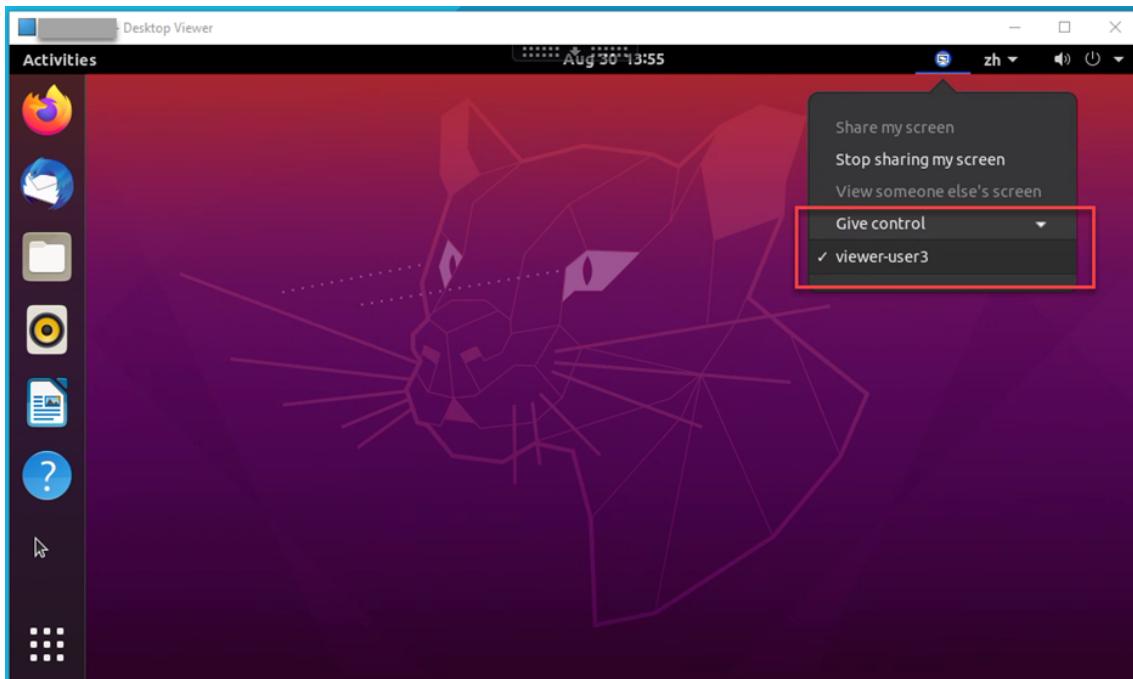


ヒント:

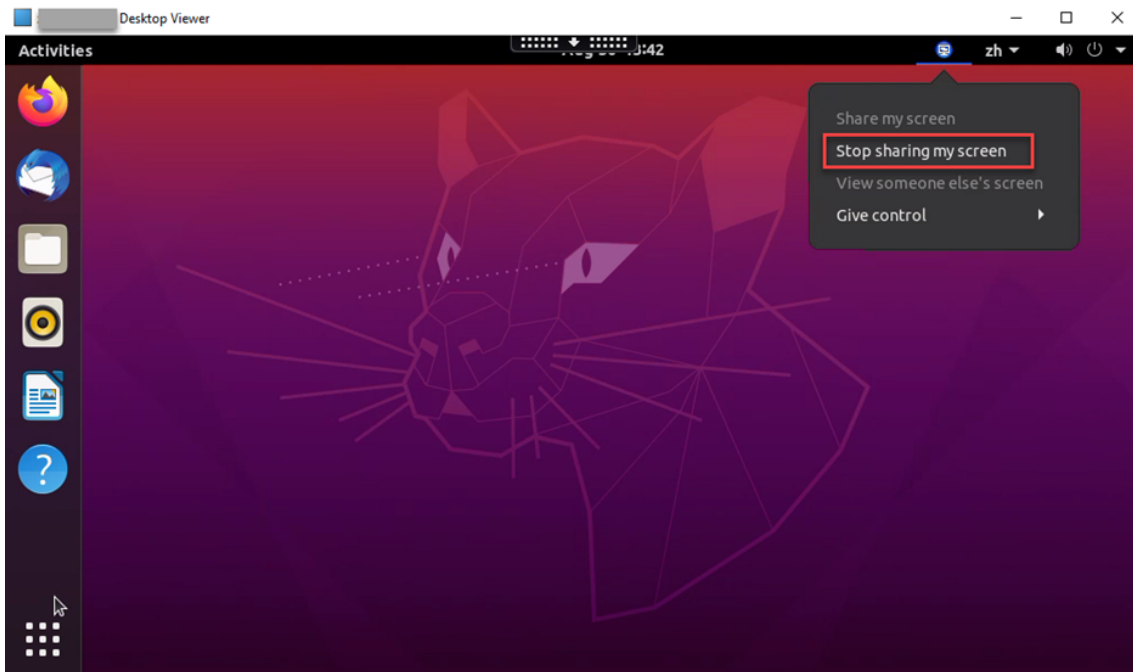
画面を共有している間は、画面の周囲に赤い境界線が表示されて、共有が進行中であることが示されま

す。

3. コピーしたコードを、画面を共有するほかの仮想デスクトップ上のセッションユーザーと共有します。
4. 閲覧者が画面を制御できるようにするには、[制御を渡す] を選択してから閲覧者の名前を選択します。制御を停止するには、閲覧者の名前をクリアします。

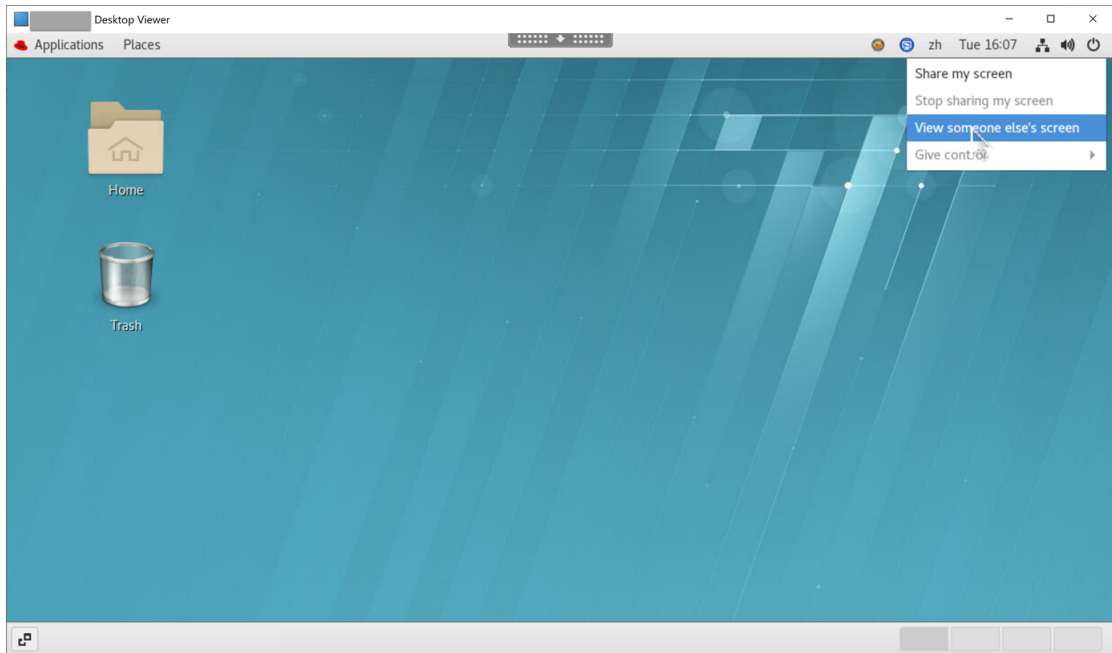


5. 画面の共有を停止するには、[画面の共有を停止] を選択します。

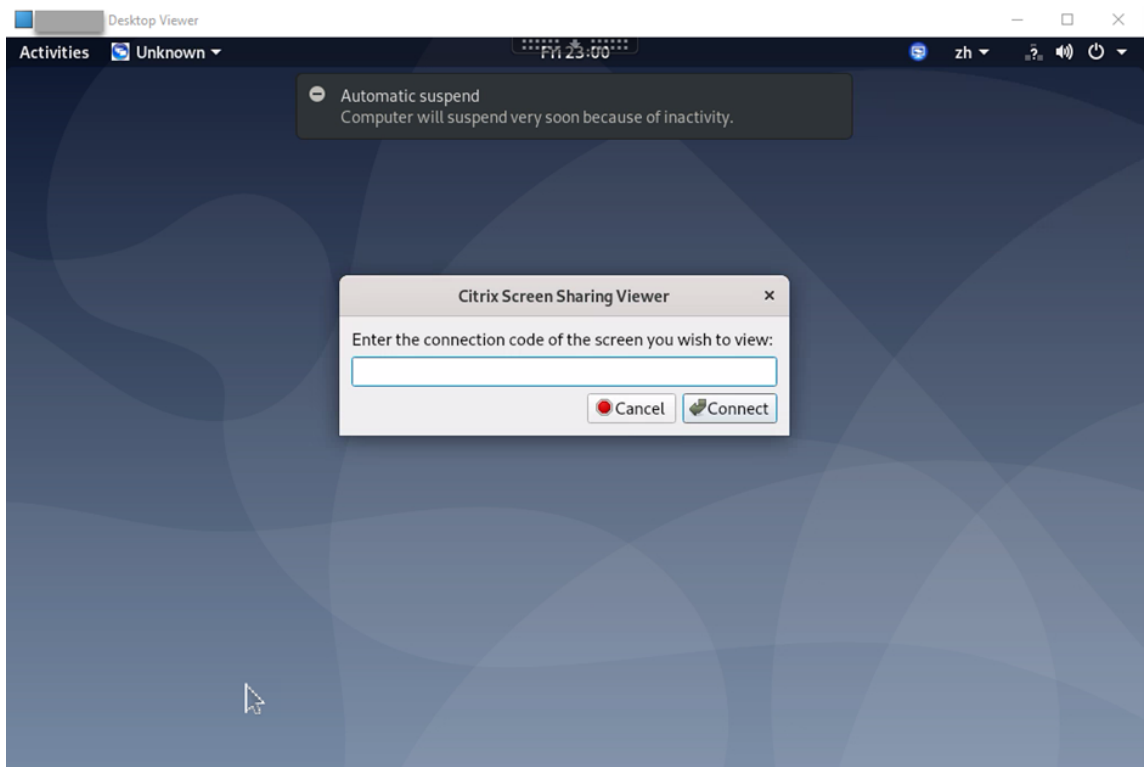


ほかのユーザーの画面を表示する手順:

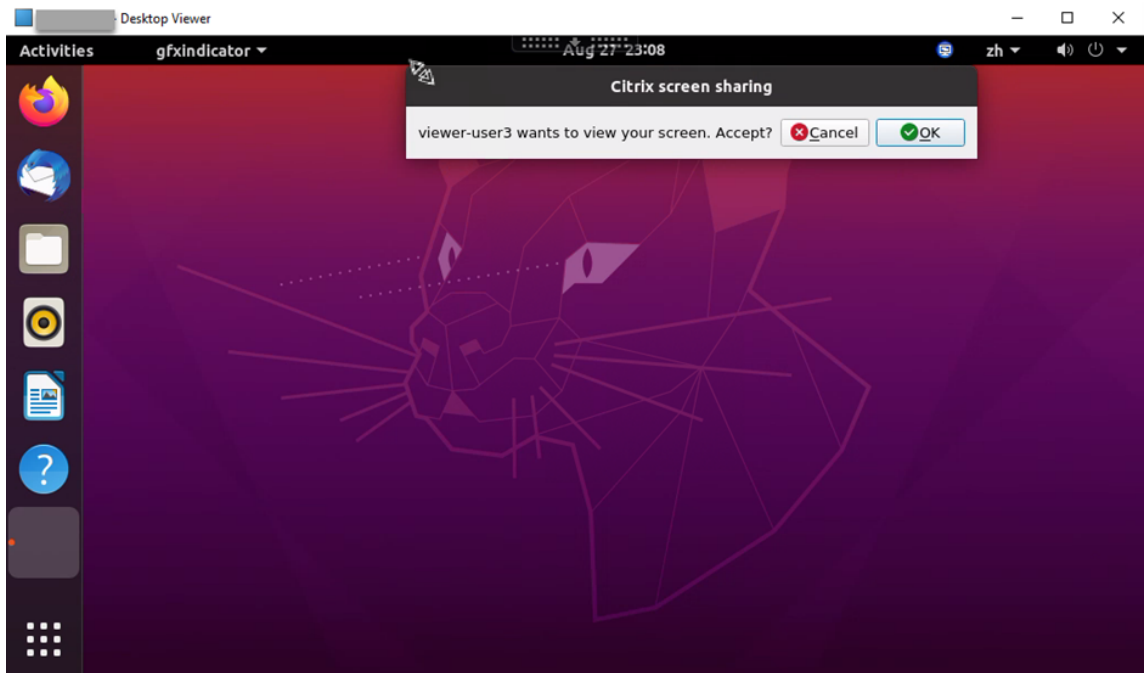
1. 仮想デスクトップの通知領域で、画面共有アイコンをクリックし、[他のユーザーの画面を表示] を選択します。



2. 表示する画面の接続コードを入力し、[接続] をクリックします。



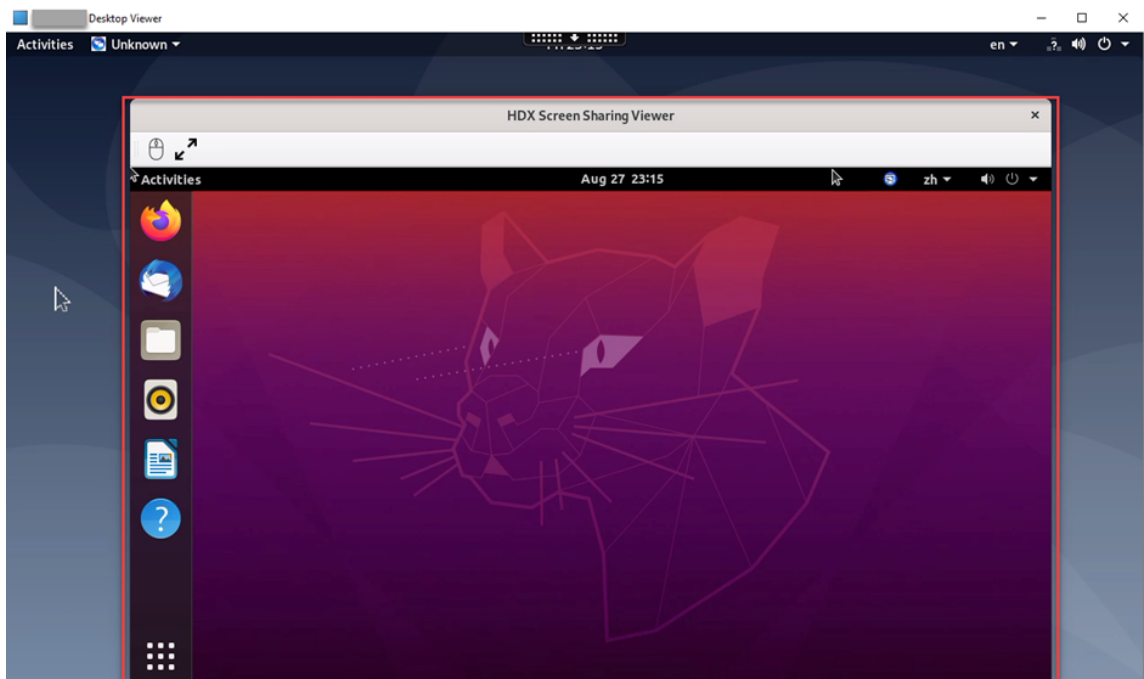
3. 画面共有者がリクエストを受け入れるのを待ちます。例:



ヒント:

- 共有者側では、Linux システムによってリクエストの通知が発行されます。
- 共有者が 30 秒以内にリクエストを受け入れない場合、リクエストは期限切れになり、プロンプトが表示されます。

4. 画面共有者が **[OK]** をクリックしてリクエストを受け入れると、共有画面が Desktop Viewer に表示されます。自分は、自動的に割り当てられたユーザー名で閲覧者として接続されます。

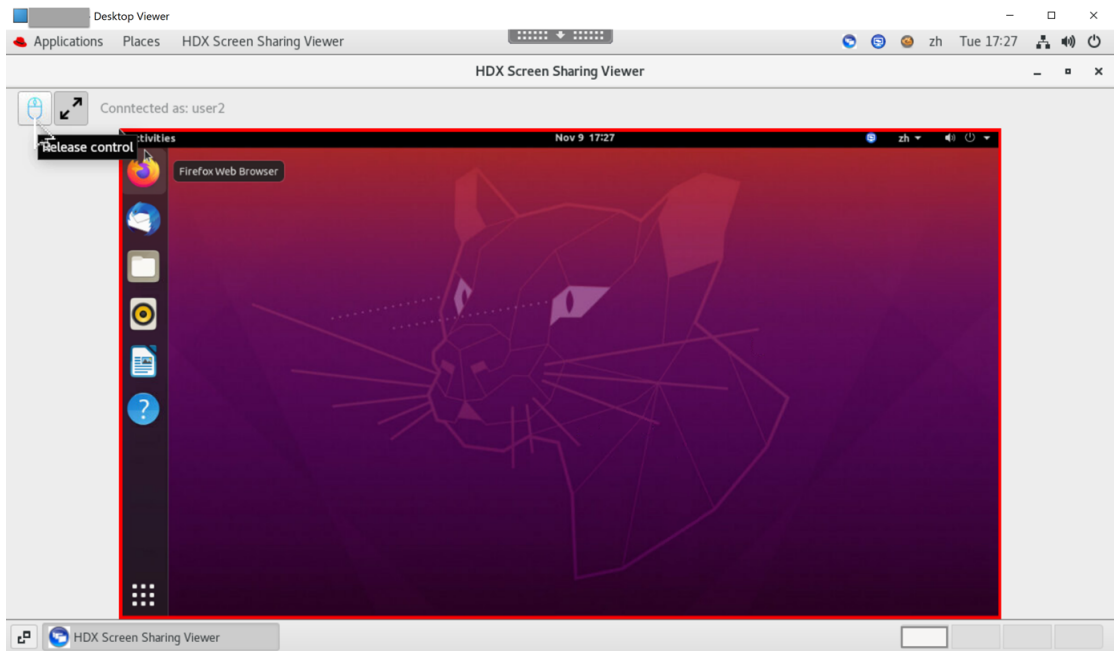


5. 共有画面の制御をリクエストするには、左上隅にあるマウスアイコンをクリックします。

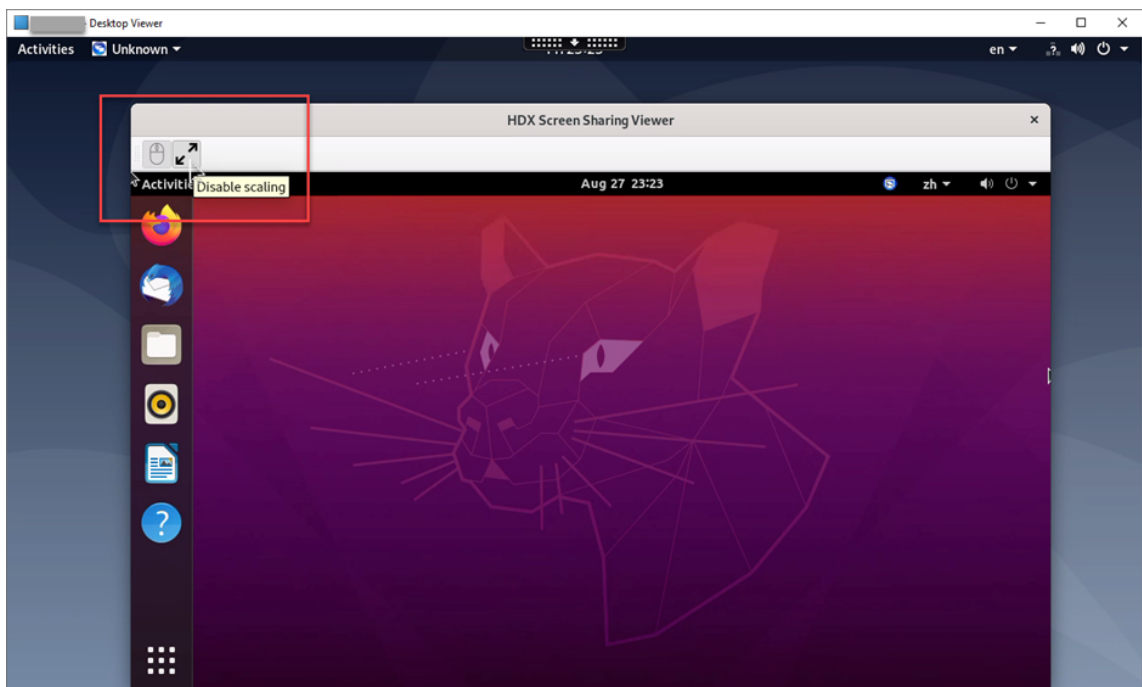
ヒント:

- 共有者が 30 秒以内にリクエストを受け入れない場合、リクエストは期限切れになります。
- 一度に 1 人の閲覧者のみが共有画面を制御できます。

共有画面の制御を解除するには、マウスアイコンをもう一度クリックします。



6. ディスプレイの拡大縮小を無効にしたり、ウィンドウサイズに拡大したりするには、マウスアイコンの横にあるアイコンをクリックします。



## 構成

デフォルトでは、画面共有機能は無効になっています。有効にするには、次の設定を完了します：

1. システムトレイを有効にします。
2. Citrix Virtual Apps and Desktops 2112 以降の場合は、Citrix Studio で [画面共有] ポリシーを有効にします。
3. (オプション) Citrix Virtual Apps and Desktops 2109 以前の場合は、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：

```
1 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。

## 注意事項

- 画面共有機能では、H.265 ビデオコーデックはサポートされていません。
- 画面共有機能は、アプリセッションでは使用できません。
- 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 人の閲覧者とセッション画面を共有できます。閲覧者の最大数は `ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire"-v "ScreenSharingViewerMaxNum"-d <hex_value>` で設定

できます。最大数に達した場合、ユーザーが追加の接続要求を受け入れようとするプロンプトが表示されま  
す。

## マルチモニターサポート

August 20, 2024

### 概要

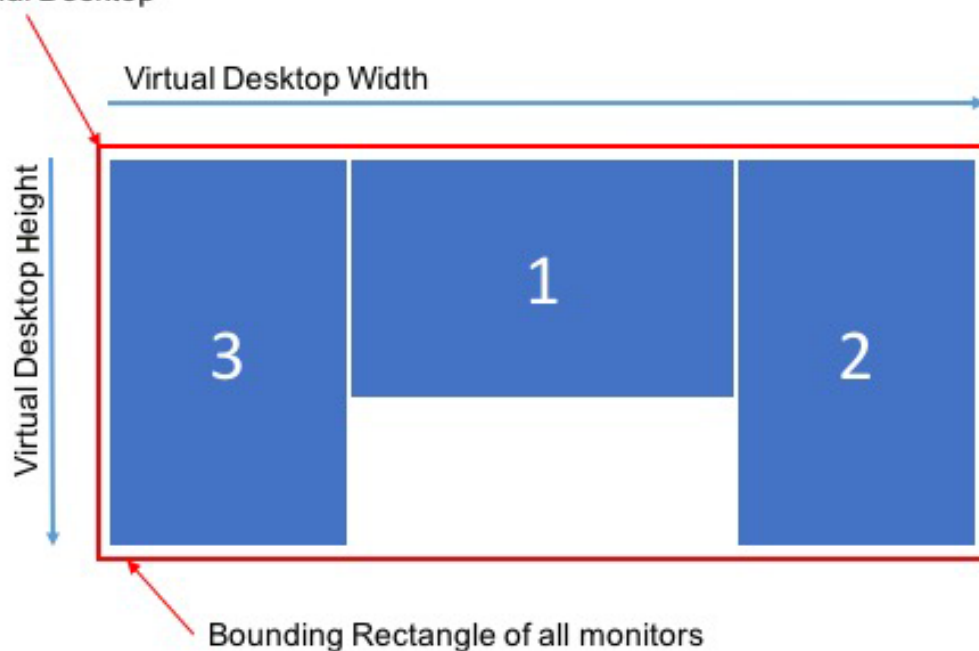
Linux VDA は、モニターごとのデフォルトの解像度が 2560×1600 である、そのまま使用できるマルチモニターをサ  
ポートしています。標準 VDA は最大 9 台のモニターをサポートし、HDX 3D Pro VDA は最大 4 台のモニターをサポ  
ートします。

ここでは、異なるモニター解像度とレイアウトに合わせて 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、モニター  
の実際のレイアウトではなく、すべてのモニターの外接四角形に基づきます。したがって、仮想デスクトップの領域  
は、理論上で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できることになります。

#### Origin of Virtual Desktop





## 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四角形の左上隅から計算されます。その点は  $X=0$ 、 $Y=0$  となり、 $X$  が水平軸、 $Y$  が垂直軸です。

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四角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の外接四角形の左下隅までの垂直距離 (ピクセル単位) です。

この計算は以下のために重要です。

- さまざまなクライアントモニターレイアウトを使用可能にする
- Linux VDA のメモリ使用量の把握

## さまざまなクライアントモニター構成を使用可能にする

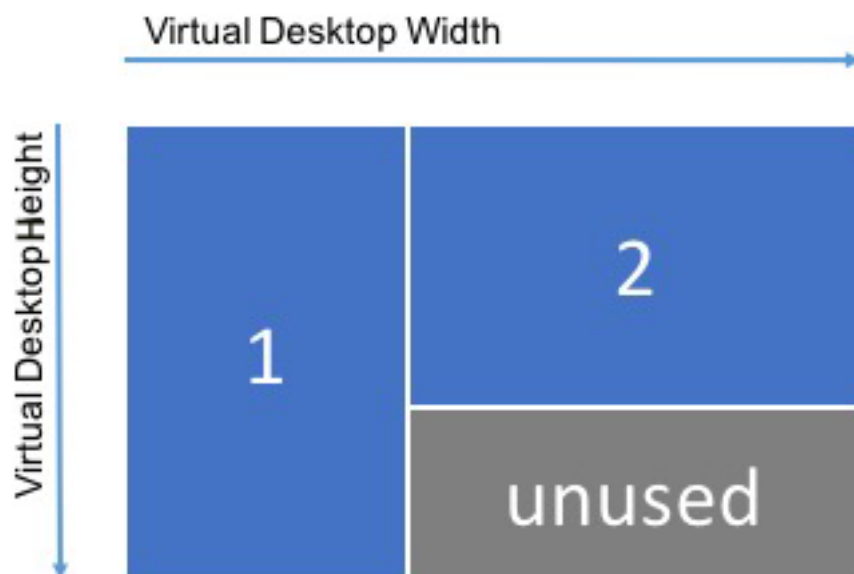
さまざまなクライアントモニター構成の最大仮想デスクトップサイズを把握することで、クライアントモニター構成に関して柔軟性があるように Linux VDA を構成できます。

次のようなクライアントモニター構成を検討します。



上の図は、解像度がそれぞれ  $2560 \times 1600$  の 2 台のモニターを備えた、そのまま使用できるマルチモニター構成を示しています。

ここで、次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。



上の図の各モニターの解像度が2560×1600の場合、そのまま使用できるマルチモニター構成のパラメーターでは不十分です。このモニターレイアウトでは、最大高さが小さすぎるため、仮想セッションデスクトップを収容できません。この例のクライアントモニター構成に対応するには、Linux VDA 仮想デスクトップのサイズを4160×2560に設定する必要があります。

マルチモニター構成で最大の柔軟性を得るには、サポートしようとしているすべてのモニターレイアウトの最小の外接四角形を見つけます。2560×1600のモニターを2台使用する構成では、以下のようなレイアウトが考えられます。

- 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、希望するすべてのレイアウトを収容できる最小の外接四角形になります。

すべてのユーザーが標準的な横長レイアウトでモニターを1台のみ使用している場合は、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合わせて設定します。



この例では、仮想デスクトップを 2560×1600 のサイズに設定する必要があります。デフォルトの構成は 5120×1600 および モニター 2 台であるため、単一モニター環境のメモリ使用量を最適化するためには構成の変更が必要です。

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 値 (インチあたりのドット数) の設定を調整します。詳細については、Knowledge Center の記事 [CTX230017](#) を参照してください。

## Linux VDA のメモリ使用量の把握

仮想デスクトップのサイズがわかれば、各 HDX セッションで使用されるメモリの量を計算できます。このメモリは、セッションの開始時にグラフィックスデータ用として各セッションに割り当てられるメモリです。セッションの存続中は変更されません。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、これはセッションごとのメモリ使用量を最も簡単に計算できる方法です。

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。

$$M = X \times Y \times Z。$$

各項目の意味は次のとおりです:

- **M** はセッションのグラフィックスに使用されるメモリの量です。
- **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色深度です。値はビット単位ではなくバイト単位であるため、32 ビットカラーの場合は 4 を使用します。

注:

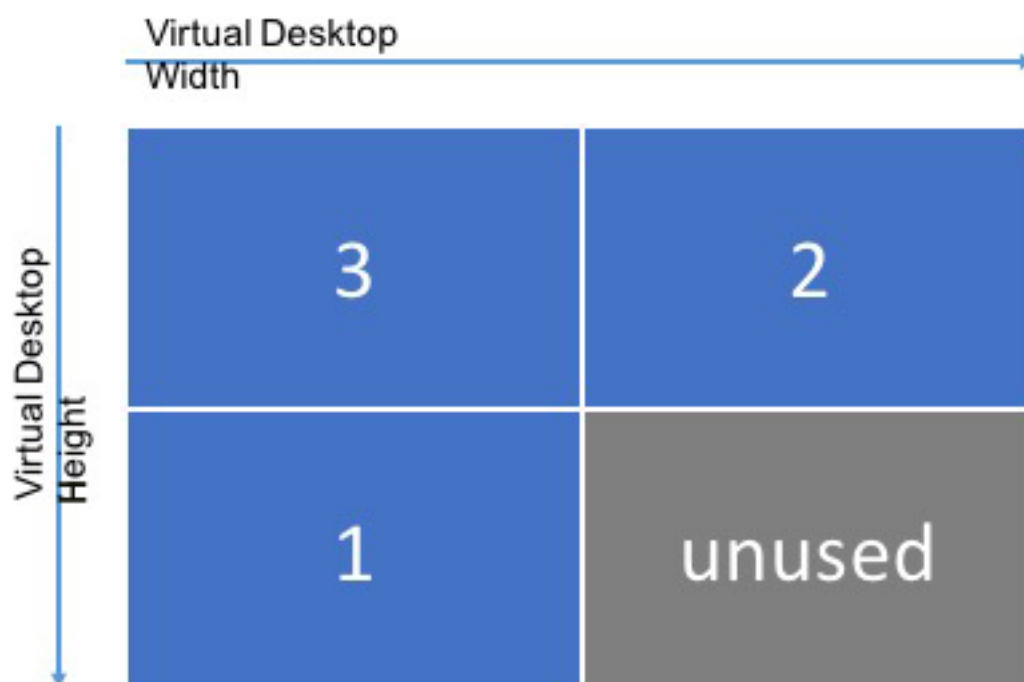
Xサーバーの色深度は、セッションの存続期間（ログインから切断/再接続を経てログオフするまで）に応じて開始し、変更することはできません。したがって、Linux VDA は仮想セッションデスクトップを常に 32 ビットとして割り当て、セッションに要求された色深度までのサンプルを割り当てます。

たとえば、1024×768 のセッションの場合、使用されるメモリは次のとおりです。

$$1024 \times 768 \times 4 / 2^{20} \text{ MB} = 3 \text{ MB}$$

各 Linux VDA のセッション密度を高めるには、メモリ使用量を理解することが重要です。

次のようなクライアントモニター構成を検討します。



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が 5120×3200 である必要があります。灰色の領域は未使用で、16,384,000 (2560 x 1600 x 4) バイトの無駄なメモリに相当することに注目してください。

### Citrix マルチモニター構成のパラメーター

次の構成パラメーターを使用して、Linux VDA のマルチモニター機能を制御できます。

- **MaxScreenNum**

パラメーター: HKEY\_LOCAL\_MACHINE/System/CurrentControlSet/Control/Citrix/Thinwire/MaxScreenNum

説明: サポートするモニターの数

種類: DWORD

デフォルト: 4

最大: 標準 VDA の場合は 9、HDX 3D Pro VDA の場合は 4

- **MaxFbWidth**

パラメータ: 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: HKEY\_LOCAL\_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFbHeight

説明: 仮想セッションデスクトップの最大高さ

種類: DWORD

デフォルト: 1,600

最大: 16,384 (8,192 x 2)

## Linux VDA マルチモニター構成の変更

次のセクションでは、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の概要を説明します。

以下を使用してモニターの最大数を設定します。

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxScreenNum" -d " NumMons" --force
```

ここでの **NumMons** は、標準 VDA の場合は 1~9、HDX 3D Pro VDA の場合は 1~4 の値になります。

以下を使用して、仮想セッションデスクトップの最大幅を設定します。

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxFbWidth" -d " MaxWidth" --force
```

ここでの **MaxWidth** は、**1,024~16,384** の値になります。

以下を使用して、仮想セッションデスクトップの最大高さを設定します。

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\
   Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxFbHeight" -d "
   MaxHeight" --force
```

ここでの**MaxHeight**は、**1,024~16,384**の値になります。

## 非仮想化 GPU

August 20, 2024

Linux VDA のドキュメントでは、非仮想化 GPU とは次を指します：

- リモート PC アクセスのシナリオで使用される GPU
- ハイパーバイザーから渡された GPU

この記事では、非仮想化 GPU のサポートに関する情報について説明します。

### **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。追加の構成は必要ありません。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がデフォルトで有効になります。

### **NVIDIA Capture SDK for Linux** をサポートしていない **NVIDIA GPU** および **AMD** や **Intel** などの他のメーカーの **GPU** と互換性がある

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。

**手順 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、「[手順 8: Runtime Environment をセットアップしてインストールを完了する](#)」を参照してください。

**手順 2:** **Xdamage** をインストールする

たとえば、**sudo apt-get install -y libxdamage1** を実行すると、XDamage を Ubuntu 20.04 にインストールできます。通常、XDamage は XServer の拡張機能として存在しています。

手順 3: 次のコマンドを実行して **XDamage** を有効にする

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
  CurrentControlSet\Control\Citrix\XDamage" -t "REG_DWORD" -v "
  XDamageEnabled" -d "0x00000001" --force
```

手順 4: **Xorg** 構成ファイルを変更する

次の 4 つのテンプレート構成ファイルは、**/etc/X11** にあります。接続されているモニターの数に基づいて、名前に対応する数字を含むテンプレート構成ファイルの 1 つを変更します。たとえば、モニターが 1 つだけ接続されている場合は、名前に数字 1 を含むテンプレート構成ファイル (ctx-driver\_name-1.conf) を変更します。2 つのモニターが接続されている場合は、名前に数字 2 を含むテンプレート構成ファイル (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** を例として使用しながら、以下の手順に従ってテンプレート構成ファイルを変更します:

1. **driver\_name** は、実際のドライバー名で置き換えてください。

たとえば、ドライバー名が **intel** の場合は、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あり、コメントアウトされています。このセクションでは、ビデオドライバー情報を記述します。ビデオドライバー情報を追加する前に、このセクションを有効にします。このセクションを有効にするには:

- a) GPU の製造元から提供されているガイドを参照して構成情報を確認します。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できることを確認します。
  - b) ネイティブ構成ファイルの [Device] セクションを **ctx-driver\_name-1.conf** にコピーします。
3. 次のコマンドを実行して、手順 1 で設定した構成ファイル名を Linux VDA が認識できるようにレジストリキーを設定します。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
  CurrentControlSet\Control\Citrix\XDamage" -t "REG_SZ" -v "
  DriverName" -d "intel" --force
```

## リモート PC アクセス VDA のモニターブランキング

Linux VDA は、非仮想化 GPU を使用するリモート PC アクセス 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 が含まれます。

この機能はデフォルトでは無効になっています。有効にするには、次の 2 つの手順を実行します：

1. 使用する Linux ディストリビューションに応じて、`evdi-dkms` パッケージをインストールします：

```
1 sudo apt install evdi-dkms
```

2. EVDI へのグラフィックディスプレイのオフロードを有効にします：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
EvdI" -d "0x00000001" --force
```

3. Intel の 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。無効にしない場合、ディスプレイマネージャーによって占有され、Citrix リモートセッションで Intel の GPU が使用できなくなります。

```
1 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か文字化けする

ローカルで 3D アプリケーションを実行でき、すべてを適切に構成しているのにグラフィックが出力されないまたはグラフィック出力がおかしい場合、原因はバグです。`/opt/Citrix/VDA/bin/setlog` を使用して `GFX_X11` を `verbose` に設定することでデバッグ用にトレース情報を収集します。

## セッションウォーターマーク

July 16, 2024

セッションウォーターマークは、データ盗難を防止し、追跡できるようにするために役立ちます。この情報は追跡可能であり、セッションデスクトップに表示されることで、データを盗むために写真や画面のキャプチャを実行するユーザーに対する抑止力になります。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止するものではありませんが、ある程度の抑止力とトレーサビリティを提供します。この機能の使用については、完全な情報のトレーサビリティを保証するものではありません。必要に応じてこの機能を他のセキュリティソリューションと組み合わせることをお勧めします。

セッションウォーターマークによって、データ盗難を追跡するための情報が伝えられます。最も重要なデータは、画面イメージが撮られたセッションの（ログオン資格情報で追跡される）ユーザー ID です。データ漏洩をより効果的に追跡するには、サーバーまたはクライアントのインターネットプロトコルアドレスや接続時間などのそ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のポリシー設定

#### セッションウォーターマークを有効化

この設定を有効にすると、セッション画面に、セッション固有の情報を示す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、これが有効になっているかどうかで異なります。

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** アドレスを含む

この設定を有効にすると、セッションで、現在のクライアント IP アドレスがウォーターマークとして表示されます。

デフォルトでは、[クライアント **IP** アドレスを含む] 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。表示される時間は、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む

この設定を有効にすると、セッションで、現在のログオンユーザー名がウォーターマークとして表示されます。表示形式は、USERNAME@DOMAINNAME です。ユーザー名は 20 文字までにすることをお勧めします。ユーザー名が

20 文字を超えている場合は、フォントサイズが小さくなるか、文字の一部が表示されず、ウォーターマークの効果が低下する可能性があります。

デフォルトでは、[ログオンユーザー名を含む] は有効になっています。

#### **VDA** ホスト名を含む

この設定を有効にすると、セッションで、現在の ICA セッションの VDA ホスト名がウォーターマークとして表示されます。

デフォルトでは、[**VDA** ホスト名を含む] は有効になっています。

#### **VDA** の IP アドレスを含む

この設定を有効にすると、セッションで、現在の ICA セッションの VDA IP アドレスがウォーターマークとして表示されます。

デフォルトでは、[**VDA** の IP アドレスを含む] は無効になっています。

#### セッションウォーターマークスタイル

この設定は、1 つ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で [複数] または [単一] を選択します。

その他のスタイルオプションについては、この記事の「ウォーターマークのカスタムテキスト」セクションを参照してください。

[複数] の場合は、セッションに 5 つのウォーターマークラベルが表示されます。中央に 1 つ、隅に 4 つです。

[単一] の場合は、セッションの中央にウォーターマークラベルが 1 つ表示されます。

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] になっています。

#### ウォーターマークの透明度

ウォーターマークの不透明度を 0~100 の範囲で指定できます。指定された値が大きいほど、ウォーターマークが不透明になります。

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。空ではない文字列を入力するか、構文を設定して文字列を形成するか、組み合わせを使用することにより、セッションウォーターマークに表示することができます。空ではない文字列は、1 行あたり最大 25 文字の Unicode 文字までがサポートされます。長い文字列は切り捨てられて 25 文字になります。

たとえば、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=0><newline><serverip><newline><
clientip><newline>Citrix Linux VDA<newline>Version 2207
```

すべての構文オプションの説明は、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	注釈
<style>	ウォーターマークのレイアウトスタイル	xstyle、single、tile、horizontal	xstyle	-
<position>	ウォーターマークの位置	center、topleft、topright、bottomleft、bottomright	center	レイアウトスタイルが [単一] に設定されている場合にのみ有効です。
<rotation>	特定の角度へのウォーターマークの回転	-180~180	0	-
<transparency>	ウォーターマークの不透明度	0~100	17	-
<font>	-	システムでサポートされているフォント	サンセリフ	-
<fontsize>	-	20~50	0（自動計算）	-
<fontzoom>	<fontsize>と<image>で設定したフォントと画像のサイズのパーセンテージ	0~	100	-

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	注釈
<image>	PNG ウォーターマーク	VDA 上の PNG 画像へのパス	-	この構文は、PNG ウォーターマークを構成します。アルファチャネル付きの PNG のみがサポートされています。PNG ウォーターマークを使用している場合は、<style>、<position>、<rotation>、<transparency>、および<fontzoom> 構文オプションのみが有効です。
<date>	セッション接続の日付 (YYYY/MM/DD) のプレースホルダー	-	-	-
<time>	セッション接続の時間 (HH:MM) のプレースホルダー	-	-	-
<domain>	ユーザーアカウントドメインのプレースホルダー	-	-	-
<username>	現在のログオンユーザー名のプレースホルダー (ユーザーアカウントドメインを除く)	-	-	-
<hostname>	VDA のホスト名のプレースホルダー	-	-	-
<clientip>	クライアントの IP アドレスのプレースホルダー	-	-	-

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	注釈
<serverip>	VDA の IP アドレスのプレースホルダー	-	-	-

## 注:

[ウォーターマークのカスタムテキスト] が少なくとも 1 つの有効な構文オプションで指定されている場合、他のすべてのセッションウォーターマークポリシー（[セッションウォーターマークを有効にする] 以外）は無視されます。

構文オプションを指定しないままにするか、サポートされていない値に設定すると、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、次のいずれかの場合にサポートされます：
  - [圧縮にビデオコーデックを使用する] が [画面全体に使用] に設定されている場合。
  - [圧縮にビデオコーデックを使用する] が [可能であれば使用] に設定され、[\[3D 画像ワークロードの最適化\]](#) 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、ブラウザーコンテンツのリダイレクトが使用されるセッションではサポートされていません。この機能を使用するには、ブラウザーコンテンツの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。
- 全画面ハードウェアアクセラレーションモード（全画面 H.264 または H.265 エンコーディング）でレガシー NVIDIA ドライバーを使用したセッションが実行されている場合は、セッションウォーターマークはサポートされておらず、表示されません。（この場合、レジストリで `NvCaptureType` が 2 に設定されています。）
- ウォーターマークは、セッションのシャドウでは表示されません。
- ユーザーが 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る画面にウォーターマークは含まれません。そのため、画像がコピーされるのを防ぐためにスクリーンショットへ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

July 16, 2024

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 の場合は、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ことができます。

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、以下の構成手順を実行します。

### 手順 1: **VirtualGL** のインストール

<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の場合は、**.deb** パッケージをダウンロードし、RHEL ベースの Linux ディストリビューションの場合は、**.rpm** パッケージをダウンロードします。

### 手順 2: **VirtualGL** の構成

1. Linux ディスプレイマネージャー (LightDM や GNOME Display Manager (GDM) など) 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。

```
1 #/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 (GLX+ EGL バックエンド) で使用するサーバーを構成する」には「1」を選択します
  - 「3D X サーバーのアクセスを **vglusers** グループに制限する」には「n」を選択します
  - 「フレームバッファデバイスのアクセスを **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。

### 手順 3: **GPU** アクセラレーションでの **OpenGL 3D** アプリケーションの実行

Linux VDA セッションで GPU アクセラレーションを使用して OpenGL 3D アプリケーションを実行するには、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 bash ターミナルを再起動します。このアプローチにより、bash ターミナルから起動されるすべての OpenGL 3D アプリケーションについて、共有 GPU アクセラレーションが有効になります。

```
1 #/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、指定したアプリケーションの名前で次のコマンドを実行します。

```
1 #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緊密に連動します。想定される機能とパフォーマンスを実現するには、共有 GPU アクセラレーションのディスプレイマネージャーとして LightDM を使用することをお勧めします。
- WebGL ハードウェアアクセラレーションは、Ubuntu および Debian 上の Firefox でのみサポートされません。

## スケーラビリティ

GPU を共有できる最大同時セッション数は、CPU とシステムメモリに依存します。これは、GPU の最大ビデオメモリにも大きく依存します。

例:

---

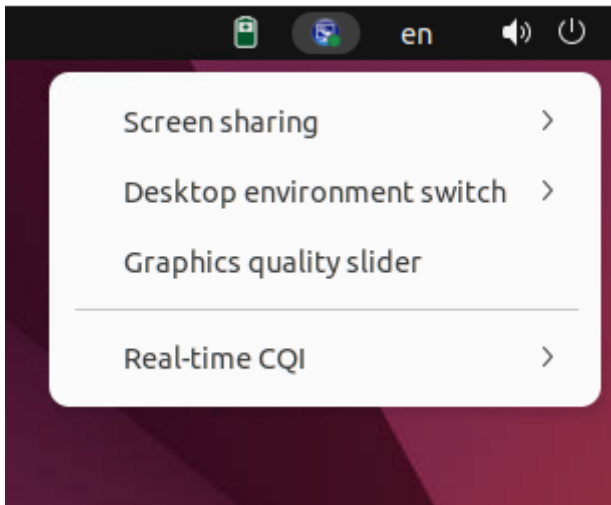
操作	結果
NVIDIA <b>M10-2B</b> vGPU には 2,048 MB のビデオメモリがあり、VariCAD Viewer などの OpenGL アプリケーションは各セッションのワークロードに 100 MB のビデオメモリを使用する場合、	理論上、サポートされる最大同時セッション数が 20 を超えることはありません。

---

## システムトレイ

August 20, 2024

セッションユーザーは、次のシステムトレイアイコンをクリックしてアクションを実行したり、インジケータを表示したりできます:



### システムトレイ内のアイテムの概要

各アイテムは、トグルのある機能に対応しています。アイテムに対応する機能が無効になっている場合、そのアイテムは非表示になり、表示されません。

- 画面共有

この機能の詳細については、「[HDX 画面共有](#)」を参照してください。

- デスクトップ環境の切り替え

このアイテムは、**ctxdesktopswitch.sh** の GUI です。詳しくは、「[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。

セッションユーザーによるデスクトップ環境のカスタマイズはデフォルトで有効になっています。無効にするには、次のコマンドを実行します：

```
1 /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

詳しくは、グラフィックの構成に関する記事の「[グラフィック品質スライダー](#)」セクションを参照してください。

- リアルタイム接続品質インジケータ

現在、ICA 往復時間 (RTT) と遅延データが表示されます。詳しくは、「[セッションデータの照会ユーティリティ](#)」を参照してください。

システムトレイアイコンは、リアルタイム接続品質インジケータの遅延に応じて異なる表示になります：





アイコンの表示変更の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:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  HighLatencyThreshold" -d "0x000000dc" --force
```

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  LowLatencyThreshold" -d "0x00000078" --force
```

実際の遅延が **LowLatencyThreshold** 以下の場合、アイコンのマークは緑色になります。実際の遅延が **HighLatencyThreshold** より大きい場合、アイコンのマークは赤色になります。それ以外の状況では、アイコンのマークは黄色になります。リアルタイム接続品質インジケータが無効になっている場合、アイコンには色のマークが付きません。

リアルタイム接続品質インジケータは、デフォルトでは有効で表示されます。これを無効にして非表示にし、トレイアイコンに色の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:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  EnableCqiShow" -d "0x00000000" --force
```

- グラフィック状態

このインジケータは、現在のセッションのグラフィック設定を示します。このオプションは、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:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  EnableGfxInfo" -d "0x00000000" --force
```

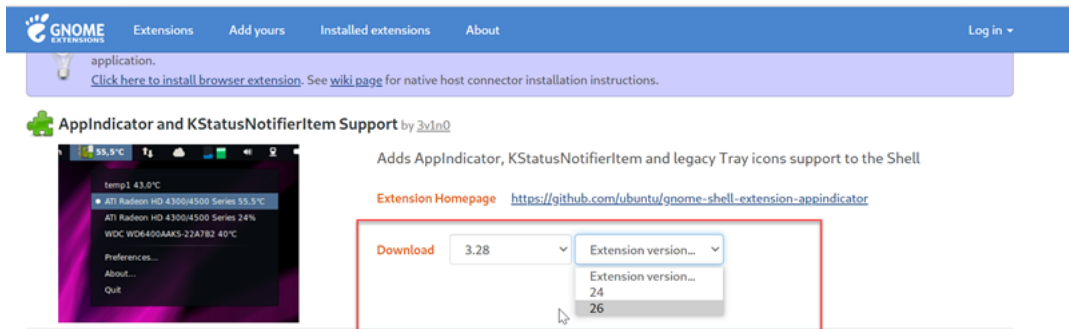
## システムトレイの有効化

システムトレイおよびクライアントのバッテリー状態の表示は、デフォルトで有効になっています。ただし、特定の状況においては、システムトレイおよびクライアントのバッテリー状態の表示を有効にするために追加の設定を行う必要があります。詳細は次のとおりです:

1. Citrix Studio で [グラフィック状態インジケータ] ポリシーを有効にします。
2. (この手順が必要になるのは、**GNOME** とともにインストールされた **RHEL 8.x/9.x**、**Rocky Linux 8.x/9.x**、**Debian 11**、または **SUSE 15.x** を使用している場合のみです。) GNOME シェルの互換性のある拡張機能

インストールして、AppIndicator サポートを有効にします。

- 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 シェルと互換性のある拡張機能をダウンロードします。たとえば、シェルのバージョンが 3.28 の場合、拡張機能のバージョンとして 24 または 26 を選択できます。



- c) ダウンロードしたパッケージを展開し、展開したディレクトリの名前を **appindicator-support@rgcjonas.gmail.com** に変更します。パッケージ内の **metadata.json** ファイルの「**uuid**」値が **appindicator-support@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、**appindicator-support@rgcjonas.gmail.com** のディレクトリを `/usr/share/gnome-shell/extensions/` 配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** ファイルをほか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、**appindicator-support@rgcjonas.gmail.com** ディレクトリの **metadata.json** ファイルはルートユーザーのみが読み取ることができます。画面共有をサポートするには、**metadata.json** ファイルをほかのユーザーも読み取れるようにします。

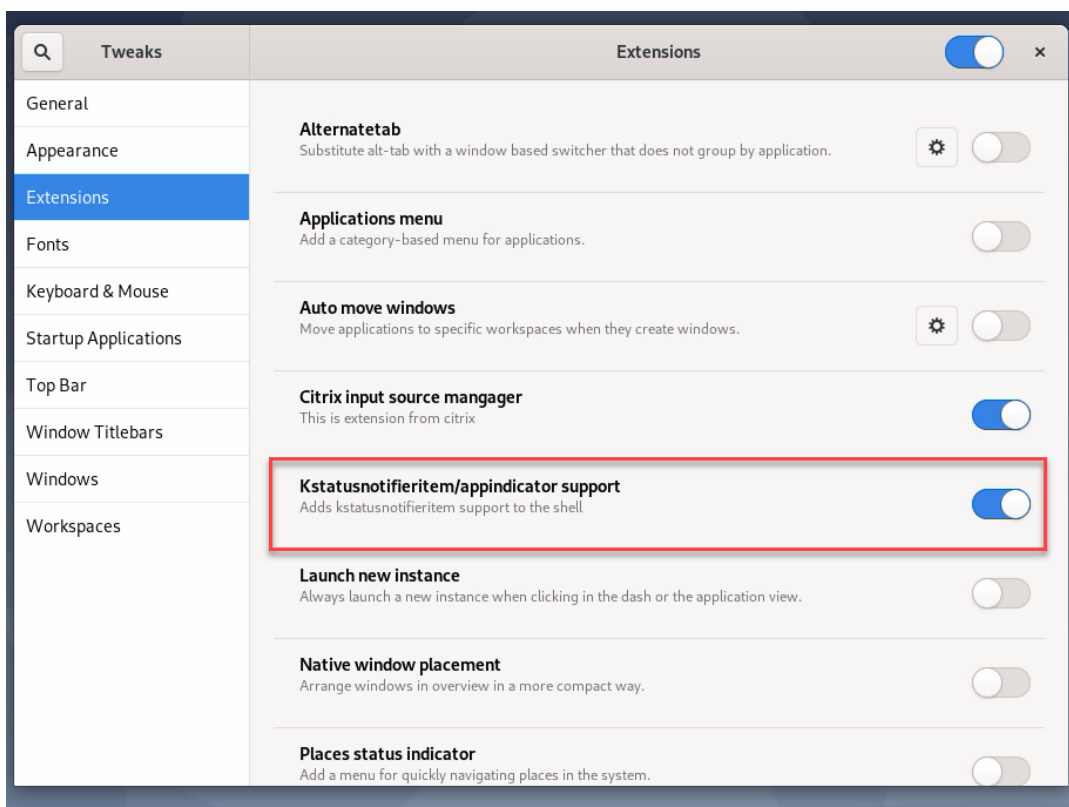
- f) GNOME Tweaks をインストールします。
- g) デスクトップ環境では、`Alt+F2`、`r`、`Enter` キーを順番に押すか、`killall -SIGQUIT gnome-shell` コマンドを実行して、GNOME シェルを再読み込みします。
- h) デスクトップ環境で、GNOME Tweaks を実行してから、Tweaks または Extensions ツールで **[KStatusNotifierItem/AppIndicator Support]** を有効にします。

注:

GNOME がインストールされた RHEL 9.x、Rocky Linux 9.x、および SUSE 15.x の場合、拡張機能の管理は Tweaks から移動されました。GNOME ソフトウェアアプリストアから GNOME 拡張機能をダウンロードし、GNOME 拡張機能で AppIndicator サポートを有効にすることをお勧めします。

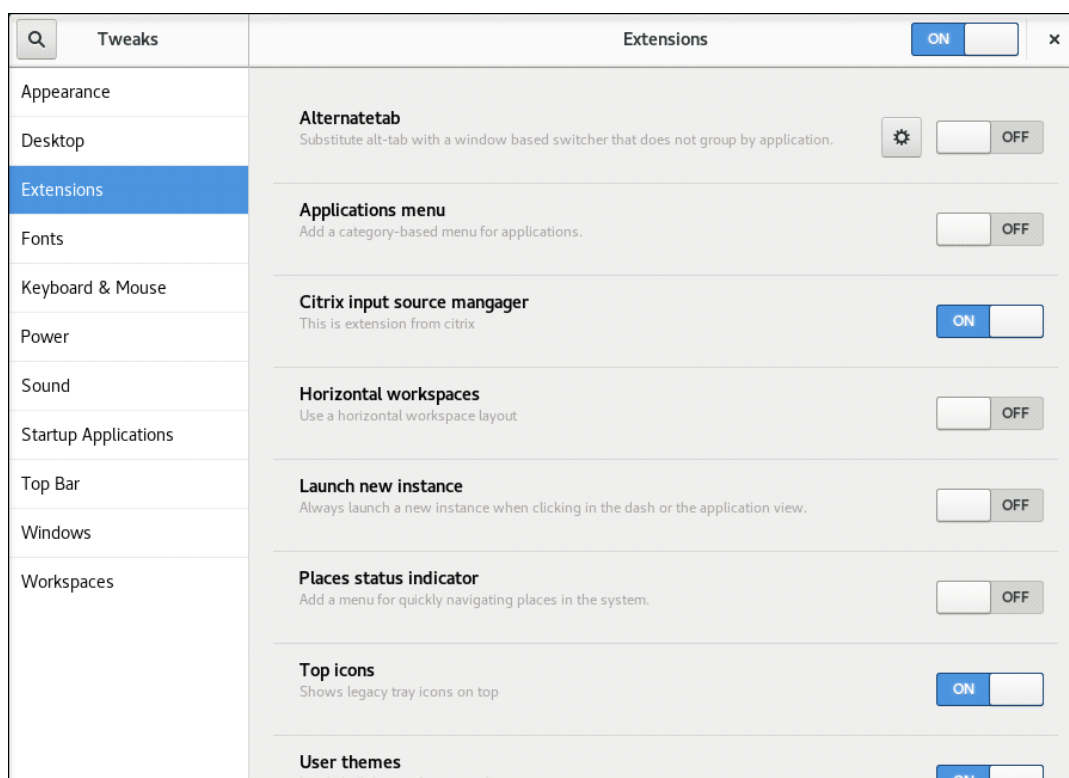
3. (この手順が必要になるのは、**GNOME** とともにインストールされた **Debian 11** を使用している場合のみです。) 次の手順を実行し、GNOME システムトレイアイコンをインストールして有効にします。

- a) `sudo apt install gnome-shell-extension-appindicator` コマンドを実行します。GNOME で拡張機能を表示するには、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) **[Activities]** 画面で **Tweaks** を検索します。
- c) Tweaks ツールで **[Extensions]** を選択します。
- d) **[Kstatusnotifieritem/appindicator support]** を有効にします。



4. (この手順が必要になるのは、**GNOME** とともにインストールされた **RHEL 7.9** を使用している場合のみです。) 次の手順を実行し、GNOME システムトレイアイコンをインストールして有効にします。

- a) **[Activities]** 画面で **Tweaks** を検索します。
- b) Tweaks ツールで **[Extensions]** を選択します。
- c) 上部のアイコンを有効にします。



d) セッションからログアウトし、再びセッションにログオンします。

## Thinwire のプログレッシブ表示

July 16, 2024

低帯域幅または高遅延の接続では、セッションのインタラクティブ性が低下する可能性があります。たとえば、Web ページのスクロールが遅くなったり、応答しなくなったり、途切れたりすることがあります。キーボードやマウスの操作がグラフィックの更新に追いつかないことがあります。

バージョン 7.17 までは、セッションを低表示品質に設定する、または色深度を低く（16 ビットまたは 8 ビットグラフィック）設定することで、ポリシー設定を使用して帯域幅消費を軽減できました。ただし、弱い接続状態であることをユーザーが知っている必要がありました。HDX Thinwire では、ネットワークの状態に基づいて静的な画像の品質を動的に調整することはありませんでした。

バージョン 7.18 以降、HDX Thinwire は、次のいずれかの場合に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：

- 使用可能な帯域幅が 2Mbps を下回っている。
- ネットワークの遅延が 200 ミリ秒を超えている。

このモードでは：

たとえば、プログレッシブ更新モードが有効な次のグラフィックでは、文字 **F** と **e** に青いアーティファクトがあり、イメージは大きく圧縮されています。このアプローチにより、帯域幅消費が大幅に軽減され、画像とテキストをより迅速に受信でき、セッションのインタラクティブ性が向上します。

## Features



セッションとの通信が停止すると、劣化した画像やテキストが徐々にシャープになり、劣化がなくなります。たとえば、次のグラフィックでは、文字に青のアーティファクトがなくなっており、画像が元の品質で表示されています。

## Features



画像の場合、ランダムにブロック単位でシャープ化します。テキストの場合、個々の文字や単語の一部がシャープ化します。シャープ化のプロセスは数フレームにわたって行われます。この方法により、単一の大きなシャープ化フレームによる遅延を回避します。

遷移画像（ビデオ）は、アダプティブ表示または Selective H.264 で管理されたままです。

### プログレッシブモードの動作

デフォルトでは、[表示品質] ポリシー設定が [高]、[中]（デフォルト）、または [低] の場合、プログレッシブモードはスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフ（使用されない）になります。

- [表示品質] が [常に無損失] または [操作時は低品質] である
- [単純なグラフィックスの優先色深度] が [8 ビット] である
- [圧縮にビデオコーデックを使用する] が [画面全体に使用]（全画面の H.264 が望ましい場合）である

プログレッシブモードがスタンバイ状態である場合、デフォルトでは次のいずれかの状況によって有効になります。

- 使用可能な帯域幅が 2 Mbps を下回っている

- ネットワーク遅延が 200 ミリ秒を上回っている

モードの切り替えが発生した後は、悪いネットワーク状況が瞬間的であっても、そのモードが最低 10 秒間継続されます。

## プログレッシブモードの動作の変更

プログレッシブモードの動作を変更するには、次のコマンドを実行します：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplay" -d "<value>" --force
```

value には以下を入力します：

0 = 常時オフ（いかなる場合でも使用しないでください）

1 = 自動（ネットワーク状態、デフォルト値に基づいてオンとオフを切り替える）

2 = 常時オン

自動モード（1）の場合、次のコマンドのいずれかを実行して、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。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

value には Kbps 単位のしきい値（デフォルト = 2,048）を入力します

例：帯域幅が 4Mbps を下回ると、プログレッシブモードがオンに切り替わります

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

value にはミリ秒単位のしきい値（デフォルト = 200）を入力します

例：ネットワーク遅延が 100 ミリ秒を下回ると、プログレッシブモードがオンに切り替わります。

## 一般コンテンツリダイレクト

July 16, 2024

## 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のリダイレクト

---

操作	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した場合。	クライアント側の全ボリュームが、ホームディレクトリの <b>ctxmnt</b> サブディレクトリの下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の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) でリダイレクトを構成した場合。	ユーザーが指定したローカルボリューム部分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、USB デバイスをローカルに接続されているかのように使用することができます。

## クライアントドライブマッピング

July 16, 2024

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のリダイレクトを使用して、ホスト側セッションでクライアント側ファイルにアクセスできるようにすることができます。クライアントドライブマッピングとクライアントフォルダーのリダイレクトの比較は次のとおりです：

---

操作	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した場合。	クライアント側の全ボリュームが、ホームディレクトリの <b>ctxmnt</b> サブディレクトリの下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の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) でリダイレクトを構成した場合。	ユーザーが指定したローカルボリューム部分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 でクライアントドライブリダイレクトポリシーを [許可] 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のポリシー設定](#)」を参照してください。

クライアントフォルダーの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の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：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Client
  Folder Redirection" -t "REG_DWORD" -v "CFROnlyModeAvailable" -d "0
  x00000001" --force
```

クライアントからホスト側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. 最新バージョンの Citrix Workspace アプリがインストールされていることを確認します。
2. Citrix Workspace アプリのインストール先ディレクトリで、**CtxCFRUI.exe** を実行します。
3. [カスタム] ラジオボタンをクリックし、フォルダーを追加、編集、または削除します。
4. セッションを切断してから再接続すると、変更が適用されます。

## USB デバイスリダイレクト

August 20, 2024

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、USB デバイスをローカルに接続されているかのように使用することができます。

ヒント：

ネットワーク遅延が 100 ミリ秒未満の場合は、USB デバイスリダイレクトを使用することをお勧めします。ネットワーク遅延が 200 ミリ秒を超える場合は、USB デバイスリダイレクトを使用しないでください。

USB デバイスリダイレクトの主な機能として、次の 3 つが挙げられます：

- 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 カーネルドライバおよびユーザーモードのライブラリで構成されており、ユーザーはカーネルドライバと通信してすべての USB データを取得することができます。

Linux VDA は、オープンソース 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 USB データ転送はすべて、Citrix ICA USB プロトコルに格納されます。

**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 アプリ間の通信の橋渡しとして機能します。

#### **Citrix USB** サービスモジュール:

Citrix USB サービスモジュールは、USB デバイスの接続や取り外しなど、USB デバイスの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、Linux VDA への USB デバイスのリダイレクトが正常に行われると、デバイスノードがシステムの/dev パスに作成されます。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 デバイスが正常に機能するかどうかはドライバーによって決まり、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 セッションにアクセスできません。USB デバイスの接続を確認するには、ドライバーをインストールしてシステムを正しく構成してください。

Linux VDA は、クライアントからリダイレクトが正常に行われた USB デバイスの一覧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。USB 3.0 デバイスをクライアントデバイスの USB 3.0 ポートに挿入できます。

次のデバイスは、Linux VDA のこのバージョンをサポートしていることが確認されています。ほかのデバイスを使用すると、予期せぬ結果が生じる場合があります。

---

USB 大容量記憶装置デバイス	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

---

USB 大容量記憶装置デバイス	VID:PID	ファイルシステム
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、最初に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おきます。

---

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

---

---

Yubico USB	VID:PID
Yubico YubiKey OTP+FIDO+CCID - Keyboard, HID	1050:0407

---

---

Web 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 の場合のみ) **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、デフォルトでは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 では使用できません。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するか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ます。

## USB デバイスリダイレクトポリシーの設定

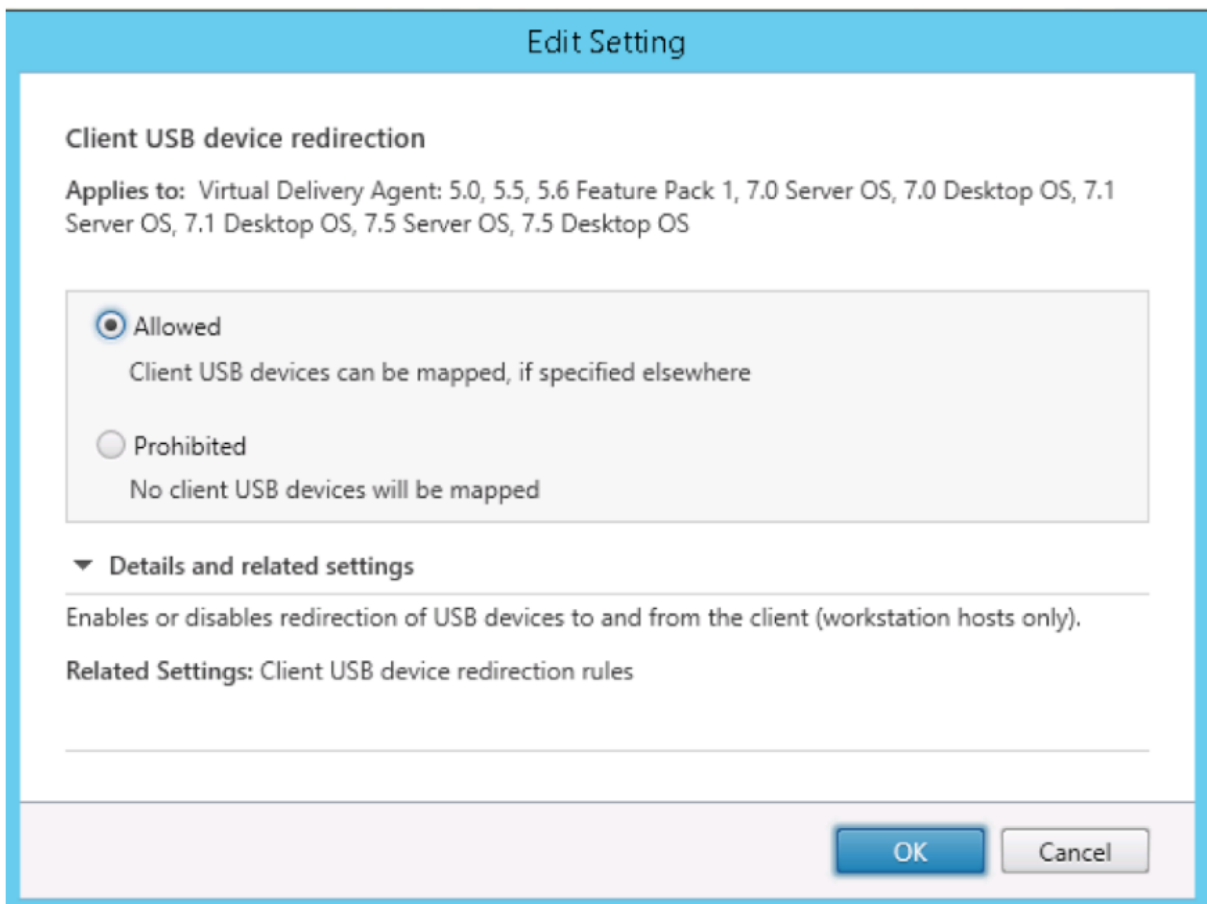
USB デバイスのリダイレクトの有効化および無効化は、Citrix ポリシーにより制御されます。Delivery Controller ポリシーを使用してデバイスの種類を指定することもできます。USB デバイスリダイレクトを Linux VDA に設定するには、次のポリシーと規則を設定します：

- クライアント USB デバイスリダイレクトポリシー
- クライアント USB デバイスリダイレクト規則

**USB デバイスリダイレクトの有効化** Citrix Studio で、クライアントからの USB デバイスリダイレクトを有効（または無効）にします（ワークステーションのホストの場合のみ）。

[設定の編集] ダイアログボックスで、以下の設定を行います：

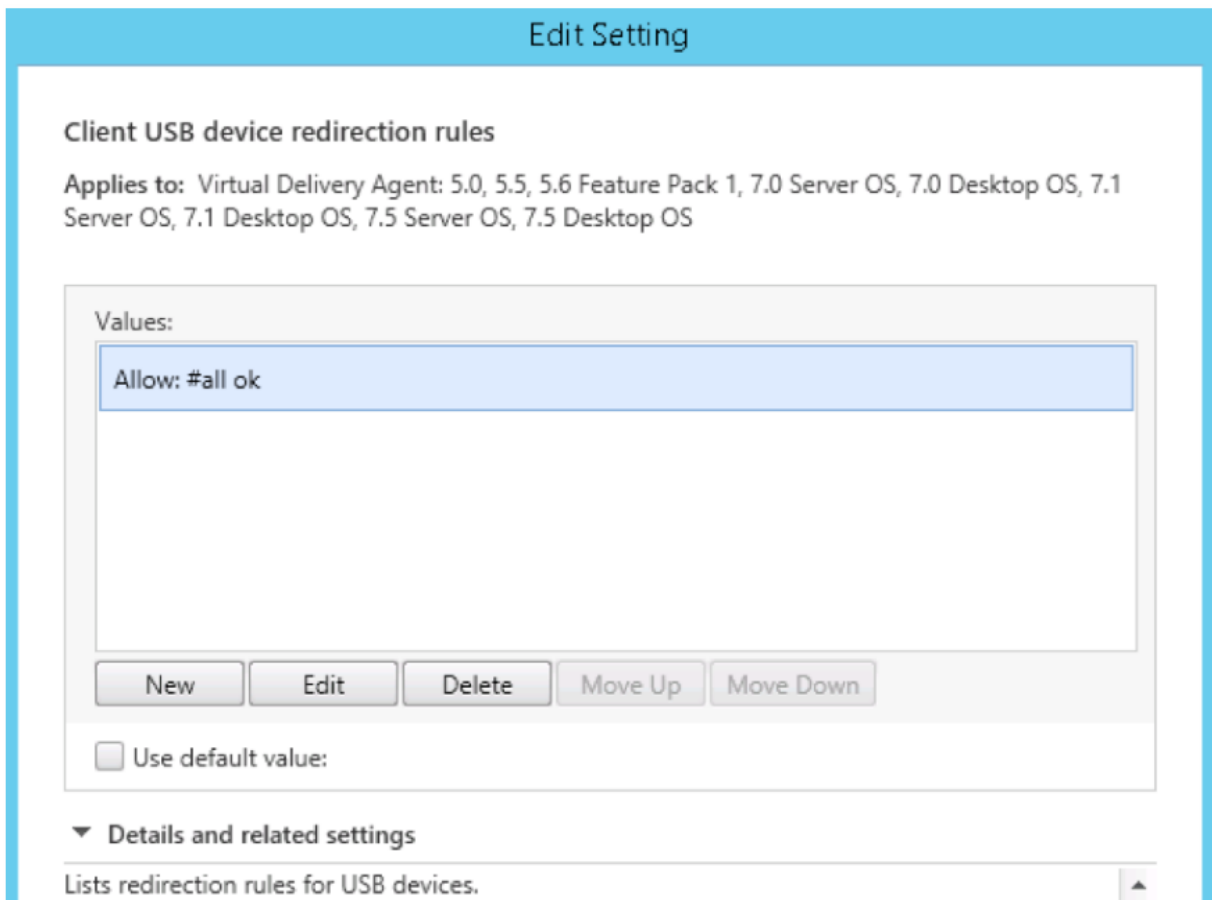
1. [許可] を選択します。
2. [OK] をクリックします。



**USB** デバイスリダイレクト規則の設定 USB リダイレクトポリシーを有効にしたら、Citrix Studio を使用して、Linux VDA での使用を許可または禁止するデバイスを指定して、リダイレクト規則を設定します。

[クライアント **USB** デバイスリダイレクト規則] ダイアログボックスで、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規則を追加するか、[編集] をクリックして既存の規則を確認します。
2. 規則の作成または編集後、[OK] をクリックします。



**Edit Setting**

**Client USB device redirection rules**

Applies to: Virtual Delivery Agent: 5.0, 5.5, 5.6 Feature Pack 1, 7.0 Server OS, 7.0 Desktop OS, 7.1 Server OS, 7.1 Desktop OS, 7.5 Server OS, 7.5 Desktop OS

Values:

- Allow: #all ok

New Edit Delete Move Up Move Down

Use default value:

▼ Details and related settings

Lists redirection rules for USB devices.

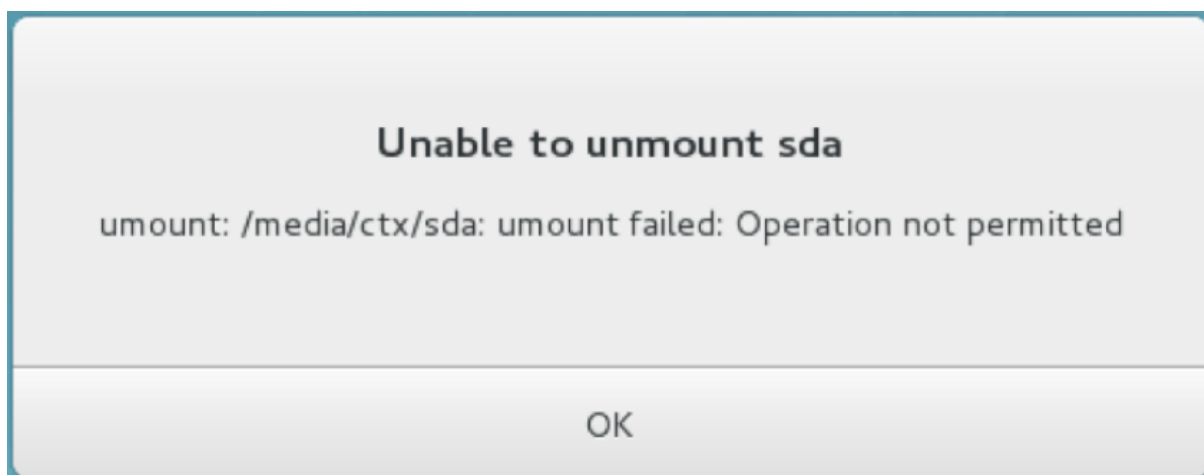
汎用 USB デバイスリダイレクトの設定について詳しくは、「[Citrix の汎用 USB リダイレクトの設定ガイド](#)」を参照してください。

## USB デバイスリダイレクト問題のトラブルシューティング

このセクションでは、Linux VDA の使用におけるさまざまな問題のトラブルシューティングについて説明します。

リダイレクトされた **USB** ディスクをマウント解除できない

Linux VDA では、Citrix Workspace アプリからリダイレクトされたすべての USB ディスクを管理者権限で管理し、所有者のみがリダイレクトされたデバイスにアクセスできるようにしています。そのため、管理者権限を持つユーザーだけがデバイスをマウント解除できます。



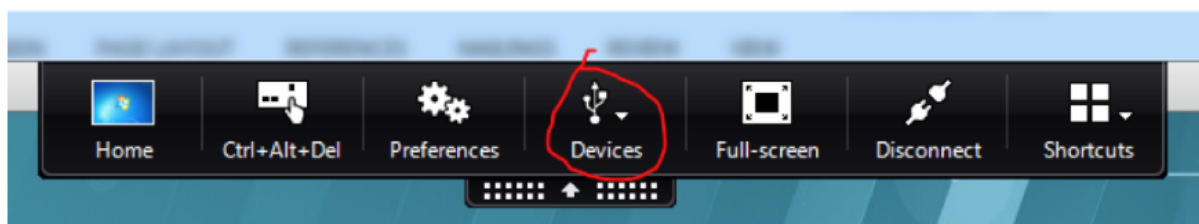
### USB ディスクのリダイレクトを停止するとファイルが失われる

Citrix Workspace アプリのツールバーを使用して USB ディスクのリダイレクトを直ちに停止すると、ディスク上で変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込むとメモリキャッシュがファイルシステムにマウントされることが原因で発生します。データはディスクそのものには書き込まれません。Citrix Workspace アプリのツールバーを使用してリダイレクトを停止した場合、データがディスクにフラッシュされる時間が残っていないため、データが失われます。

この問題を解決するには、ターミナルの **sync** コマンドを使用してデータをディスクにフラッシュしてから USB リダイレクトを停止します。

### Citrix Workspace アプリのツールバーにデバイスが見つからない場合

Citrix Workspace アプリのツールバーにデバイスが表示されなくなることがありますが、これは USB リダイレクトが行われていないことを示します。



問題が発生した場合は、次の点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設定になっている。
- 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、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：

```
1 modinfo usbip-core
```

2. 出力が次のように表示される場合は、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：

```
1 modinfo: ERROR: Module usbip-core not found.
```

- Amazon Linux 2、CentOS、RHEL、Rocky Linux の場合は、この記事の「[USB/IP カーネルモジュールのインストールまたはコンパイル](#)」セクションを参照してください。
- SUSE の場合は、<https://software.opensuse.org/package/usbip>から USB/IP パッケージをダウンロードしてインストールします。
- Ubuntu/Debian の場合は、次の手順を実行して USB/IP カーネルモジュールをコンパイルおよびインストールします：

- a) USB/IP カーネルモジュールのソースコードをダウンロードします。

<https://github.com/torvalds/linux/tree/master/drivers/usb/usbip>で 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、<https://github.com/torvalds/linux/tree/v4.15/drivers/usb/usbip>などのリンクを取得します。

[DownGit](#)に移動し、前述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用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：

```
1 unzip ${
2   USBIP_SRC }
3   .zip
4
5 cd usbip
```

- c) **Makefile** ファイルを次のように変更します：

```
1 # SPDX-License-Identifier: GPL-2.0
2
3 ccflags-$(CONFIG_USBIP_DEBUG) := -DDEBUG
4
5 obj-$(CONFIG_USBIP_CORE) += usbip-core.o
6
7 usbip-core-y := usbip_common.o usbip_event.o
8
9 obj-$(CONFIG_USBIP_VHCI_HCD) += vhci-hcd.o
10
```

```

11 vhci-hcd-y := vhci_sysfs.o vhci_tx.o vhci_rx.o vhci_hcd.o
12
13 #obj-$(CONFIG_USBIP_HOST) += usbip-host.o
14
15 #usbip-host-y := stub_dev.o stub_main.o stub_rx.o stub_tx.o
16
17 #obj-$(CONFIG_USBIP_VUDC) += usbip-vudc.o
18
19 #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:

```

1 apt-get install linux-headers-`uname -r`
2
3 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:

```

1 cp usbip-core.ko vhci-hcd.ko /opt/Citrix/VDA/lib64/

```

f) **ctxusbsd** サービスを再起動して、USB/IP カーネルモジュールを読み込みます:

```

1 systemctl restart ctxusbsd

```

**Citrix Workspace** アプリのツールバーに **USB** デバイスが表示されているのに「ポリシーの制限」という表記がある場合にリダイレクトが失敗する

問題が発生した場合は、次の手順を実行してください:

- Linux VDA ポリシーを、リダイレクトを有効にする設定にします。
- Citrix Workspace アプリのレジストリで追加のポリシー制限が構成されているかを確認します。レジストリパスで **DeviceRules** を確認し、この設定がデバイスのアクセスを拒否しないようにします:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA Client\GenericUSB
```

詳しくは、Knowledge Center

の記事「[USB デバイスの自動リダイレクトの設定方法](#)」を参照してください。

**USB** デバイスのリダイレクトは正常に行われるが、セッションでデバイスを使用できない

通常、リダイレクトできるのはサポートされている **USB デバイス** のみとなります。他のデバイスが Linux VDA のアクティブなセッションにリダイレクトできる場合もあります。リダイレクトしたデバイスごとに、ユーザーの所有するノードがシステムの **/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定されます。所有（プラグイン）しているもののアクセスできないデバイスを見つけた場合は、そのデバイスを制限されていないポリシーに追加します。



注:

USB ドライバーの場合は、Linux VDA がディスクの設定とマウントを行います。ユーザー（およびデバイスをインストールした所有者のみ）は追加の設定なしでディスクにアクセスできます。「サポートされているデバイス一覧」に掲載されていないデバイスについては、これが適用されないことがあります。

## クリップボードリダイレクト

July 16, 2024

クリップボードリダイレクトを使用すると、VDA セッションで実行中のアプリケーションとクライアントデバイスで実行中のアプリケーションとの間でデータをコピーして貼り付けることができます。

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可能な 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に関する **Citrix** ポリシー

#### クライアントクリップボードリダイレクト

この設定項目では、クライアントデバイスのクリップボードを VDA のクリップボードにマップ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のクリップボード間でデータを転送できなくするには、[禁止] を選択します。ただし、セッション内で動作するアプリケーション間でのクリップボードを介したデータ転送は無効になりません。

#### クリップボードリダイレクトの最大帯域幅 (Kbps)

この設定項目では、セッションとローカルのクリップボード間でのデータ転送で使用可能な最大帯域幅 (kbps) を指定します。

#### クリップボードリダイレクトの最大帯域幅 (%)

この設定項目では、セッションとローカルのクリップボード間でのデータ転送で使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

#### 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

この設定項目では、コピーして貼り付ける 1 回の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、[クリップボードのクライアントからセッションへの転送サイズを制限する] 設定を有効にします。次に、[サイズ制限] フィールドに、ローカルクリップボードとセッション間のデータ転送のサイズを定義する値をキロバイト単位で入力します。

デフォルトでは、この設定は無効になっており、クライアントからセッションへの転送に制限はありません。

#### 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

この設定項目では、コピーして貼り付ける 1 回の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、[クリップボードのセッションからクライアントへの転送サイズを制限する] 設定を有効にします。次に、[サイズ制限] フィールドに、セッションとローカルクリップボード間のデータ転送のサイズを定義する値をキロバイト単位で入力します。

デフォルトでは、この設定は無効になっており、セッションからクライアントへの転送に制限はありません。

#### クライアントクリップボードの書き込み制限とクライアントクリップボードに書き込みを許可する形式

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して貼り付けられるようになります (クライアントへの書き込み)。

以下のシステム定義のクリップボードデータ形式を追加できます。

- 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の書き込み制限とセッションクリップボードに書き込みを許可する形式

これら 2 つの設定を有効にすると、特定のデータ形式をクライアントからセッションにコピーして貼り付けられるようになります (クライアントへの書き込み)。

以下のシステム定義のクリップボードデータ形式を追加できます。

- 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

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [クライアント IME](#)
- [クライアント IME ユーザーインターフェイスの同期](#)
- [動的なキーボードレイアウトの同期](#)
- [ソフトキーボード](#)
- [多言語入力のサポート](#)

### X キーボード拡張 (XKB) の構成

August 20, 2024

2407 リリース以降、Linux VDA はキーボード構成のデフォルトの XKB 規則として evdev を指定します。この決定には、さまざまな正当な理由があります：

- 最近の Linux ディストリビューションのほとんどは、デフォルトで evdev XKB 規則を使用します。
- ほぼすべてのアプリケーションは、evdev XKB 規則のキーコードパターンに従っています。
- evdev XKB 規則は、基本的な英数字と一般的なファンクションキーに加えて、「Cancel」、「Redo」、「Undo」、「XF86Copy」、「XF86Open」、「XF86Paste」などの幅広いキーシボルの拡張サポートを提供します。
- Linux の evdev XKB 規則は、さまざまなキーボード関連の問題、特にキー入力に対する誤った応答に関する問題に対処します。

XFree86 規則に戻す場合は、次のレジストリ設定を行うことができます：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "XkbRuleName" -d "xorg"
```

このコマンドでは、大文字と小文字が区別される正確なテキスト「xorg」のみが「xfree86」規則セットとして認識され、その他の入力値は無効です。

## クライアント入力システム (IME)

July 16, 2024

### 概要

2 バイト文字（日本語、中国語、韓国語などの文字）は、IME から入力する必要があります。Windows ネイティブの CJK IME など、クライアント側で Citrix Workspace アプリと互換性がある任意の IME を使用して、これらの文字を入力します。

### インストール

この機能は、Linux VDA をインストールするときに自動でインストールされます。

### 使用状況

通常どおりに Citrix Virtual Apps または Citrix Virtual Desktops のセッションを開きます。

クライアント側 IME 機能の使用を開始するには、クライアント側での必要に応じて入力方式を変更します。

## 既知の問題

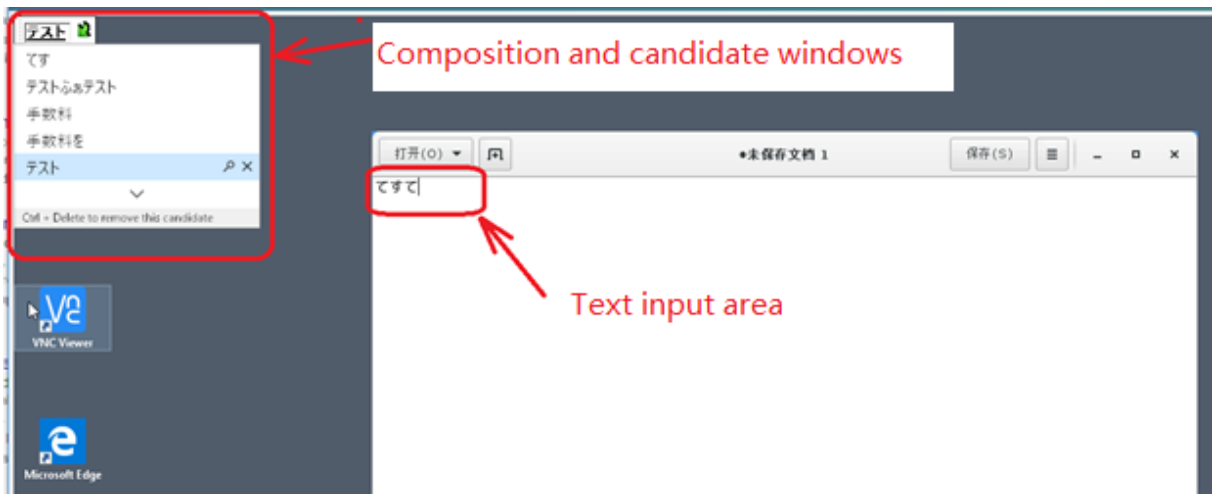
- クライアント側 IME 機能を使用して Google スプレッドシートのセルに文字を入力するには、スプレッドシート内のセルをダブルクリックする必要があります。
- クライアント側 IME 機能は [パスワード] フィールドで自動で無効になりません。
- IME ユーザーインターフェイスは、入力領域ではカーソルに追従しません。

## クライアント IME ユーザーインターフェイスの同期

July 16, 2024

### 概要

クライアント側 IME ユーザーインターフェイス（作成ウィンドウと候補ウィンドウを含む）は、これまで画面の左上隅に配置されていました。このインターフェイスはカーソルに追従せず、テキスト入力領域ではカーソルから離れて配置されることがありました：



Citrix ではユーザービリティが強化され、以下のように、クライアント側 IME でのユーザーエクスペリエンスがさらに改善されています：



### 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：
  - Ubuntu: <https://help.ubuntu.com/community/ibus>
  - CentOS、RHEL: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/7.0\\_release\\_notes/sect-red\\_hat\\_enterprise\\_linux-7.0\\_release\\_notes-internationalization-input\\_methods](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods)
  - Debian: <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

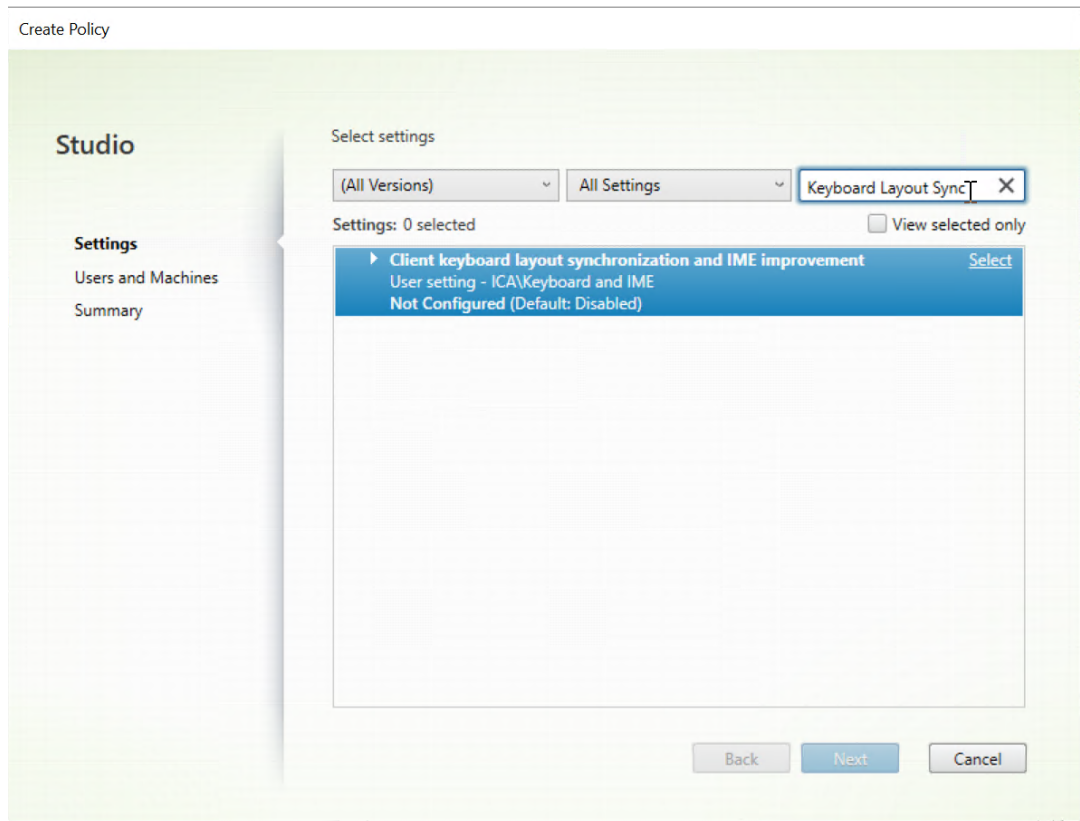
クライアント側 IME ユーザーインターフェースの同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、[クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを設定するか、`ctxreg`ユーティリティを使用してレジストリを編集します。

#### 注:

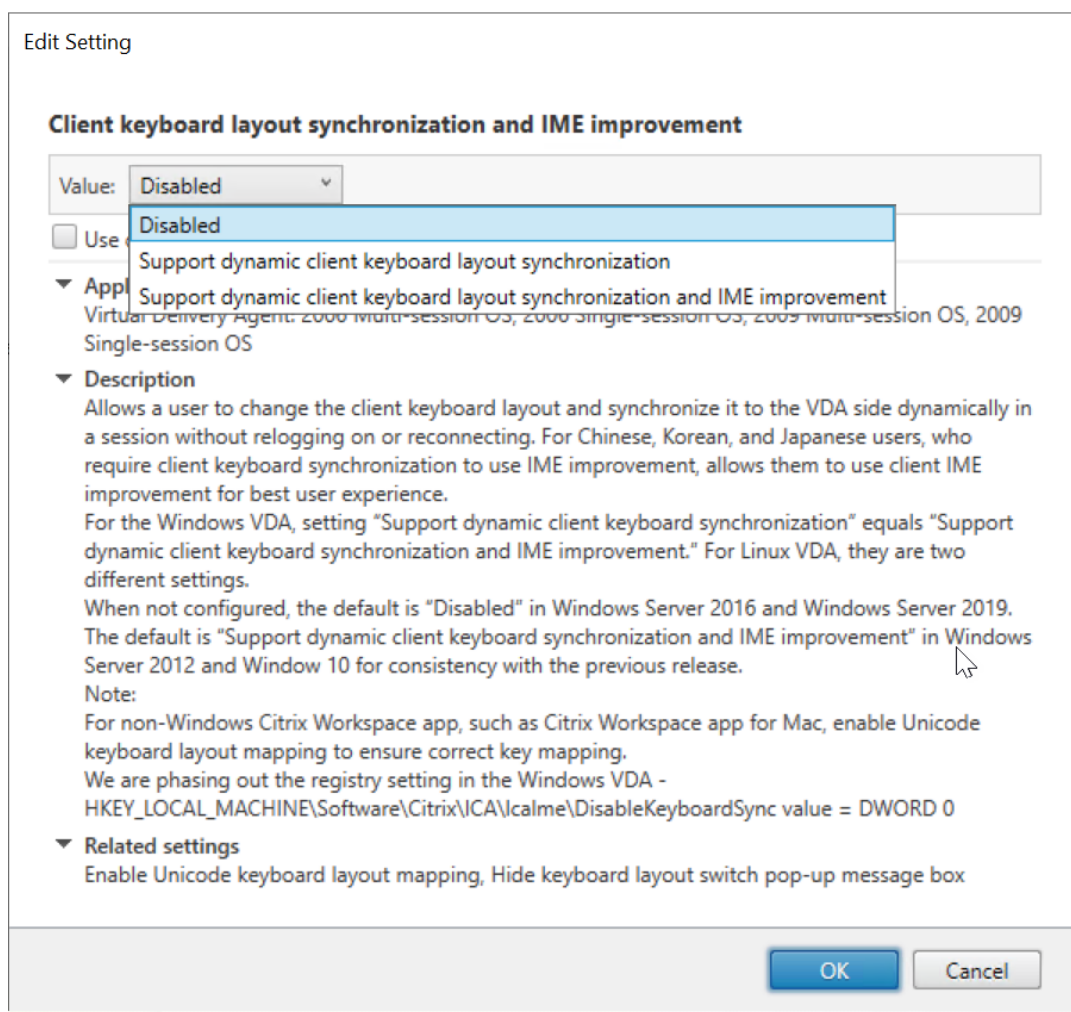
[クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトとマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定の Linux VDA のレジストリ設定は、その VDA のすべてのセッションに適用されます。

- [クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを設定して、クライアント IME ユーザーインターフェース同期機能を有効または無効にします:

1. 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。
2. [クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを検索します。



3. ポリシー名の横にある [選択] をクリックします。
4. ポリシーを設定します。



以下の3つのオプションが利用可能です：

- 無効：動的なキーボードレイアウトの同期とクライアント IME ユーザーインターフェイスの同期を無効にします。
  - 動的なクライアントキーボードレイアウトの同期のサポート： `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` の **SyncKeyboardLayout** レジストリキーの DWORD 値に関係なく、動的キーボードレイアウトの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の同期と **IME** の改善のサポート： `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` の **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の同期を有効にします。
- `ctxreg`ユーティリティを使用してレジストリを編集し、クライアント側 IME ユーザーインターフェイスの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。



```
1 /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。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncClientIME" -d "0x00000000"
```

## 動的なキーボードレイアウトの同期

July 16, 2024

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なければなりませんでしたが。キーマッピングの問題は、たとえばキーボードレイアウトがクライアントデバイスで英語からフランス語に変更され、VDA では変更されなかった場合などに発生し、VDA がフランス語に変更されるまでこの問題が存続することがありました。

Citrix では VDA のキーボードレイアウトとクライアントデバイスのキーボードレイアウトを自動的に同期させることで、この問題を解決しました。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たびに、VDA のレイアウトも変更されます。

注:

HTML5 向け Citrix Workspace アプリは、動的なキーボードレイアウトの同期機能をサポートしていません。

## 構成

動的なキーボードレイアウトの同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、[クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを設定するか、`ctxreg` ユーティリティを使用してレジストリを編集します。

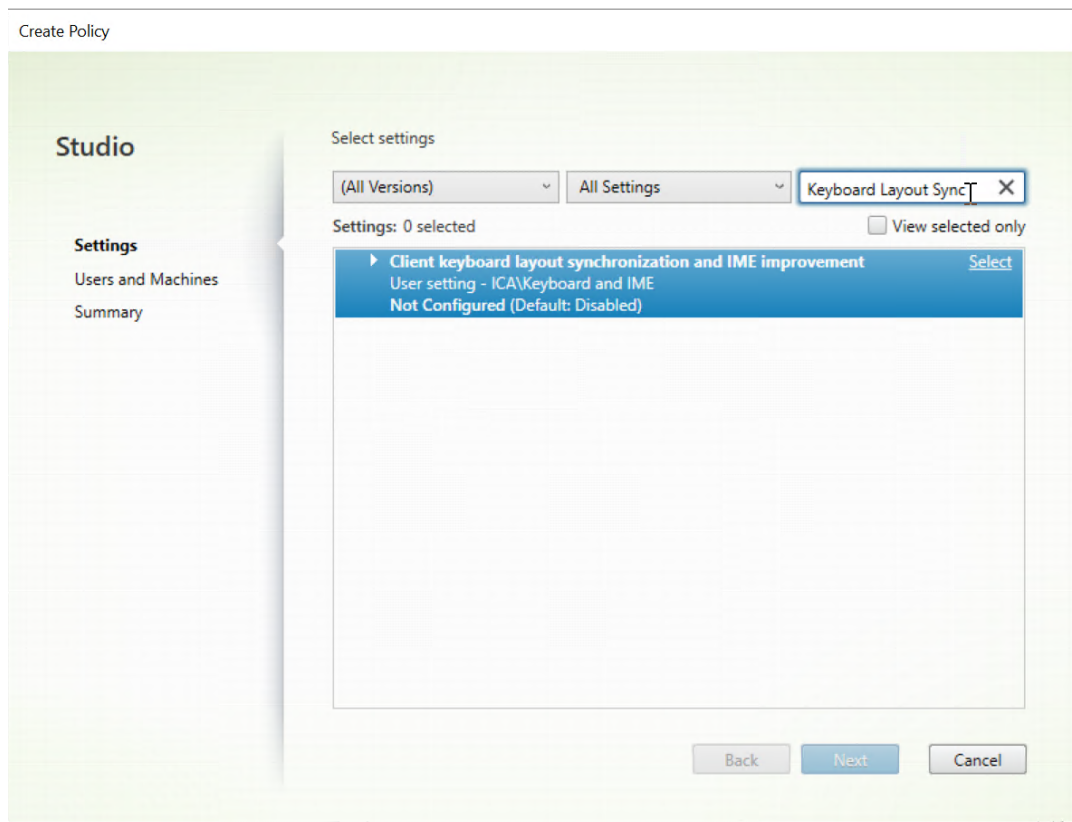
注:

[クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトとマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定の Linux VDA のレジストリ設定は、その VDA のすべてのセッションに適用されます。

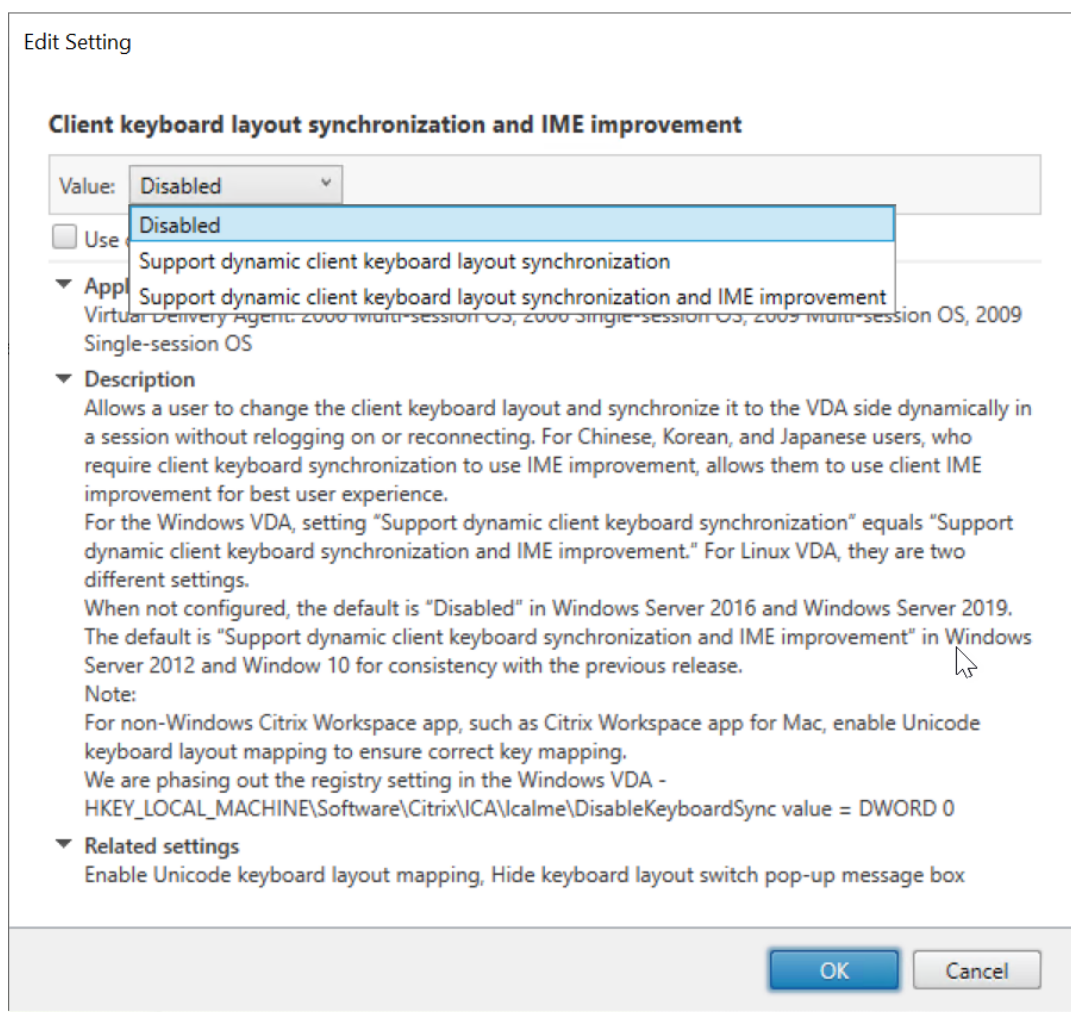
- [クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを設定して、動的なキーボードレイアウトの同期機能を有効または無効にします:

1. 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。

2. [クライアントキーボードレイアウトの同期と **IME** の改善] ポリシーを検索します。



3. ポリシー名の横にある [選択] をクリックします。
4. ポリシーを設定します。



以下の3つのオプションが利用可能です：

- 無効：動的なキーボードレイアウトの同期とクライアント IME ユーザーインターフェイスの同期を無効にします。
  - 動的なクライアントキーボードレイアウトの同期のサポート：HKEY\_LOCAL\_MACHINE \SYSTEM \CurrentControlSet \Control \Citrix \LanguageBar の **SyncKeyboardLayout** レジストリキーの DWORD 値に関係なく、動的キーボードレイアウトの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の同期と **IME** の改善のサポート：HKEY\_LOCAL\_MACHINE \SYSTEM \CurrentControlSet \Control \Citrix \LanguageBar の **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の同期を有効にします。
- **ctxreg**ユーティリティを使用してレジストリを編集し、動的なキーボードレイアウトの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：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncKeyboardLayout" -d "0x00000001"
```

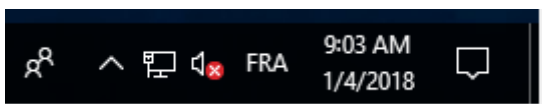
この機能を無効にするには、次のコマンドを実行します：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncKeyboardLayout" -d "0x00000000"
```

## 使用状況

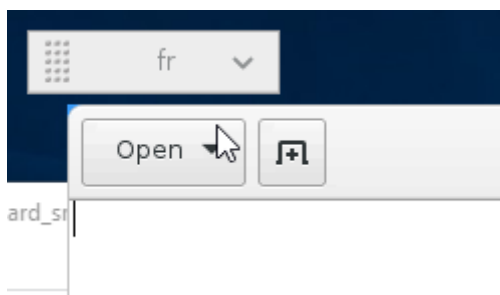
この機能を有効にすると、セッション中にクライアントデバイス上でキーボードレイアウトが変更された場合、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語 (FR) に変更すると、次のようになります。



Linux VDA セッションのキーボードレイアウトも「fr」に変わります。

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、この自動変更が表示されます：



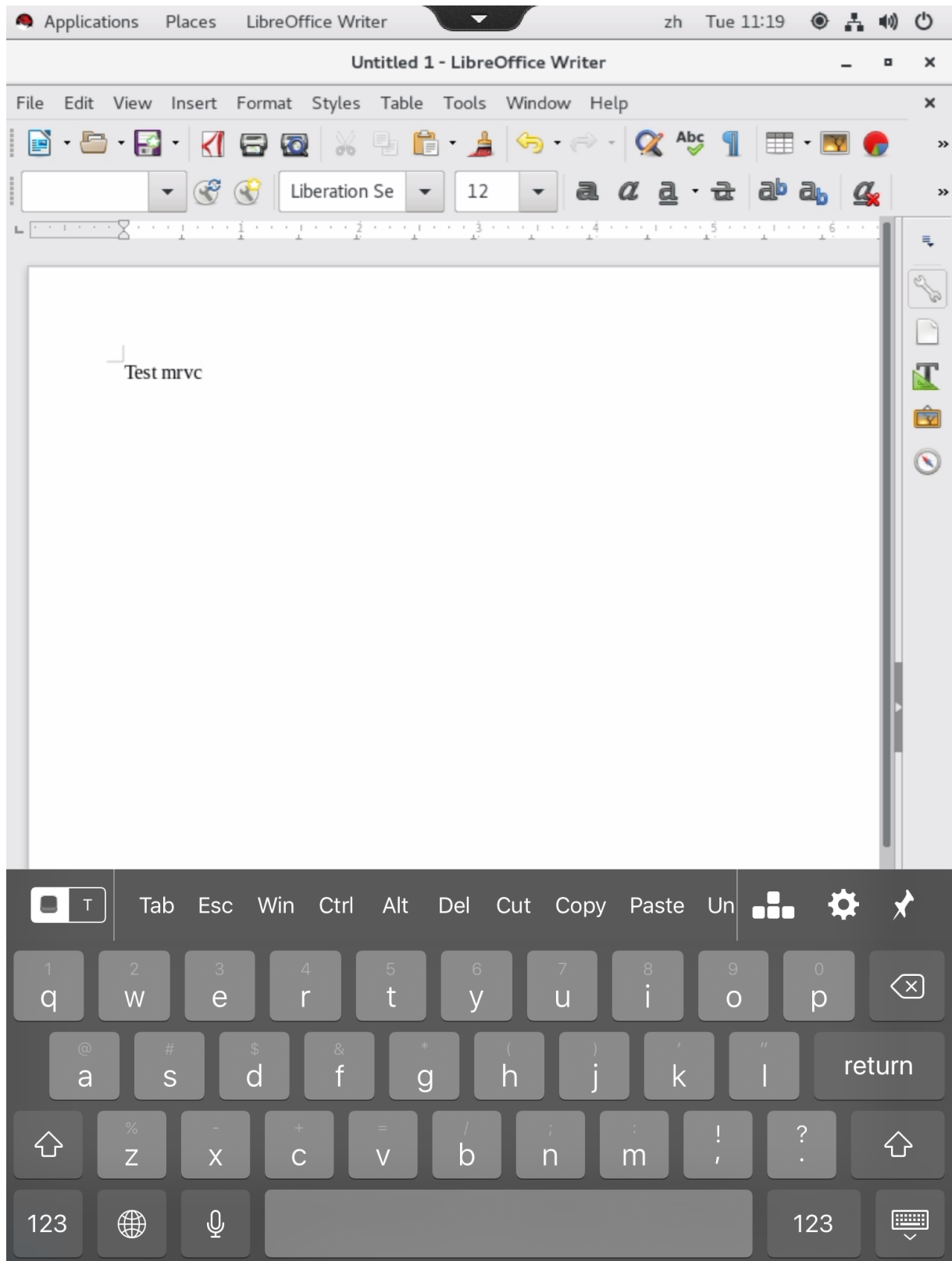
デスクトップセッションでは、この自動変更がタスクバーに表示されます：



## ソフトキーボード

July 16, 2024

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のセッションで利用できます。ソフトキーボードは、入力フィールドで入力を開始すると表示され、入力を終了すると非表示になります。



注:

この機能は、iOS 向け Citrix Workspace アプリおよび Android 向け 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は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 の機能構成は、その VDA 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:

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

2. Citrix Studio でキーボードの自動表示ポリシーを [許可] 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:

```
1 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。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc" -v "Enabled" -d "0x00000000"
```

注:

これらの設定は、新しいセッションにログオンする場合、またはログオフして現在のセッションに戻る場合に有効になります。

### 制限事項

- この機能は Google Chrome、LibreOffice、その他のアプリで機能しないことがあります。
- 手動でソフトキーボードを非表示にした後再表示するには、入力フィールド以外をクリックして、再度現在の入力フィールドをクリックします。
- Web ブラウザーで 1 つの入力フィールドをクリックしてから別のフィールドをクリックすると、ソフトキーボードが表示されないことがあります。この問題を回避するには、入力フィールド以外をクリックしてから対象の入力フィールドをクリックします。
- この機能は、Unicode 文字やダブルバイト文字（日本語、中国語、韓国語など）をサポートしません。

- ソフトキーボードは、パスワード入力フィールドでは利用できません。
- ソフトキーボードは、現在の入力フィールドと重なって表示されることがあります。この場合、アプリのウィンドウを移動するか、画面を上スクロールして入力フィールドをアクセスできる位置に移動します。
- Citrix Workspace アプリと Huawei タブレットとの互換性の問題によって、Huawei タブレットに物理キーボードが接続されている場合でもソフトキーボードが表示されます。

### 多言語入力のサポート

August 20, 2024

Linux VDA バージョン 1.4 以降では、Citrix で公開アプリケーションのサポートが追加されています。ユーザーは、Linux デスクトップ環境がなくても、必要な Linux アプリケーションにアクセスできます。

ただし、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、Linux VDA のネイティブ言語バーは公開アプリケーションでは使用できませんでした。その結果、中国語、日本語、韓国語など、IME が必要な言語でテキストを入力できませんでした。ユーザーが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に、この機能で、テキスト入力に対応した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、サーバー側の IME を選択したり、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することができます（デフォルトでは無効）。特定の Linux VDA サーバーの機能設定は、その VDA に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、種類は 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：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
  x00000001"
```

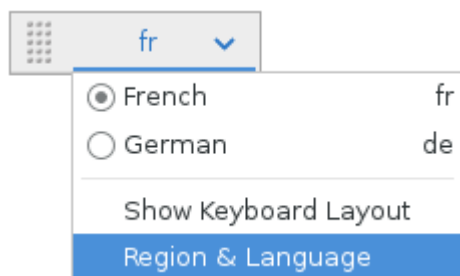
この機能を無効にするには、次のコマンドを実行します：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
  x00000000"
```

## 使用状況

使い方は簡単です。

1. 本機能を有効にします。
2. テキスト入力に対応できる公開アプリケーションにアクセスします。言語バーが、アプリケーションとともにセッションに表示されます。
3. ドロップダウンメニューから、[地域と言語] を選択して希望の言語（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、クライアント側（Citrix Workspace アプリが実行されている）でも同じキーボードレイアウトが使用されていることを確認してください。
- [地域と言語] ダイアログボックスで設定を行うには、**accountsservice** パッケージをバージョン 0.6.37 以降にアップグレードする必要があります。





## マルチメディア

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [オーディオ機能](#)
- [ブラウザー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

August 20, 2024

### 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がサポートされています：

- 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

一覧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従来の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は、ネットワーク状態に基づいてオーディオサンプリングのビットレートを動的に調整して、プレミアムなオーディオ環境を提供するため、VDA で [オーディオ品質ポリシー](#) を手動で構成する必要がなくなります。

次の表は、アダプティブオーディオと従来のオーディオとの比較を示しています：

アダプティブオーディオ	従来のオーディオ
最大オーディオサンプルレート：48kHz	最大オーディオサンプルレート：8kHz
ステレオチャンネル	モノチャンネル

## オーディオサービスの考慮事項

バージョン 2407 以降、Linux VDA は Debian 12.5、RHEL 9.x、Rocky Linux 9.x で PipeWire をサポートしており、PipeWire はデフォルトのオーディオサービスとなっています。他のディストリビューションの場合は、引き続き PulseAudio を使用してください。

### PulseAudio

- 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。

### PipeWire

Linux VDA は、Debian 12.5、RHEL 9.x、Rocky Linux 9.x でのみ PipeWire をサポートします。考慮すべき点は次のとおりです：

- Debian 12.5、RHEL 9.x、または Rocky Linux 9.x に Linux VDA バージョン 2407 以降がインストールされている場合は、PipeWire を使用しています。
- RHEL 9.x または Rocky Linux 9.x に Linux VDA の 2407 より前のバージョンがインストールされている場合は、PulseAudio を使用しています。PulseAudio を引き続き使用する場合は、何もしないでください。PipeWire に切り替えるには、次の手順を実行します：

1. PulseAudio パッケージを pipewire-pulseaudio に置き換えます：

```
1 dnf swap --allowerasing pulseaudio pipewire-pulseaudio
```

2. PipeWire 関連サービスをグローバルに有効にします：

```
1 systemctl --global --now enable pipewire pipewire-pulse  
pipewire-pulse.socket wireplumber
```

3. Linux VDA を再起動します。

### オーディオの損失耐性モード (**Technical Preview**)

損失耐性モードはオーディオをサポートします。この機能により、リアルタイムストリーミングのユーザーエクスペリエンスが向上し、ユーザーが遅延やパケット損失が大きいネットワーク経由で接続している場合に、EDT と比較して音質が向上します。

損失耐性モードと EDT について詳しくは、Citrix Virtual Apps and Desktops ドキュメントの「[追加情報](#)」を参照してください。

## オーディオの損失耐性モード機能の有効化

オーディオの損失耐性モード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. [HDX アダプティブトランスポートポリシー](#)を設定して、アダプティブトランスポートを有効にできます。アダプティブトランスポートはデフォルトで有効になっています。
2. [アダプティブオーディオポリシー](#)を設定して、アダプティブオーディオを有効にします。アダプティブオーディオはデフォルトで有効になっています。
3. 次のレジストリ設定を通じて、損失耐性モード（EDT の信頼性の低いトランスポート）を有効にします：

```
1 ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -t "REG_DWORD" -v "fEdtUnreliableAllowed" -d "0x00000001" --force
```

## クライアントの要件と設定

オーディオの損失耐性モードを使用するには、Citrix Workspace アプリがこの機能をサポートし、有効にするように構成されていることを確認してください。それ以外の場合、オーディオはデフォルトで EDT の信頼性の高いトランスポートに設定されます。

以下は、損失耐性モードをサポートする Citrix Workspace アプリの最小バージョンです：

- Windows 向け Citrix Workspace アプリバージョン 2309 以降
- Linux 向け Citrix Workspace アプリ：バージョン 2311 以降
- Mac 向け Citrix Workspace アプリ：バージョン 2311 以降

## オーディオパフォーマンスを向上させるパケット損失補間 (**Technical Preview**)

2407 リリース以降では、ジッターバッファメカニズムが向上し、アダプティブオーディオコーデックにパケット損失補間 (PLC) が追加されました。PLC は、失われたデータパケットの再構築に役立ちます。この機能強化により、パケット損失耐性とジッター耐性が向上し、損失耐性モード (EDT Lossy) でのオーディオパフォーマンスが向上します。

この機能はデフォルトでは無効になっています。この機能を有効にするには、次の手順を実行します：

1. [オーディオの損失耐性モード機能を有効にします](#)。
2. 複数のオーディオデバイスのサポートを許可する [オーディオリダイレクト機能を有効にします](#)。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -t "REG_DWORD" -v "fEnableAudioRedirectionV4" -d "0x00000001" --force
```

3. 損失耐性モードでオーディオパフォーマンスを向上させるには、オーディオで V2 の読み取り/書き込みを有効にします:

```
1 ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -t "REG_DWORD" -v "fEnableReadWriteV2" -d "0x00000001" --force
```

## 複数のオーディオデバイスのサポート (Technical Preview)

### 概要

バージョン 2311 以降、Linux VDA ではオーディオリダイレクト機能を導入しています。この機能により、Citrix Workspace アプリがインストールされているクライアントマシン上の複数のオーディオデバイスを、リモートの Linux VDA セッションにリダイレクトできます。

この機能を有効にすると、以下のように動作します:

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

### 構成

この機能を使用するには、Linux VDA で有効にして、サポートされている Citrix Workspace アプリを選択します。

**Linux VDA** での機能の有効化 複数のオーディオデバイスのサポートを許可するオーディオリダイレクト機能は、デフォルトでは無効になっています。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:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "fEnableAudioRedirectionV4" -t BIN -d "1"
```

この機能を無効にしたり、再び有効にしたりするには、それぞれ次のコマンドを実行します:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "fEnableAudioRedirectionV4" -d "0"
```

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "
  fEnableAudioRedirectionV4" -d "1"
```

クライアントの要件と設定 この機能は、次のクライアントでのみサポートされます：

- Windows 向け Citrix Workspace アプリ
- Linux 向け Citrix Workspace アプリ：バージョン 2212 以降
- HTML5 向け Citrix Workspace アプリ：バージョン 2306 以降
- Chrome 向け Citrix Workspace アプリ：バージョン 2306 以降
- Mac 向け Citrix Workspace アプリ：バージョン 2311 以降

機能を正常に動作させるには、Citrix Workspace アプリで適切な設定が必要です。詳しくは、[Citrix Workspace アプリ](#)のドキュメントを参照してください。

#### 既知の問題

PulseAudio の[問題](#)により、Ubuntu 22.04 セッションでオーディオデバイスを切り替えようとするとき失敗する可能性があります。この問題に対処するには、現在のセッション ユーザーの PulseAudio 設定を VDA から削除し、セッションを再度開きます。PulseAudio 構成を削除するには、`$ rm -r ~/.config/pulse` コマンドを実行します。

## ブラウザコンテンツリダイレクト

July 16, 2024

### 概要

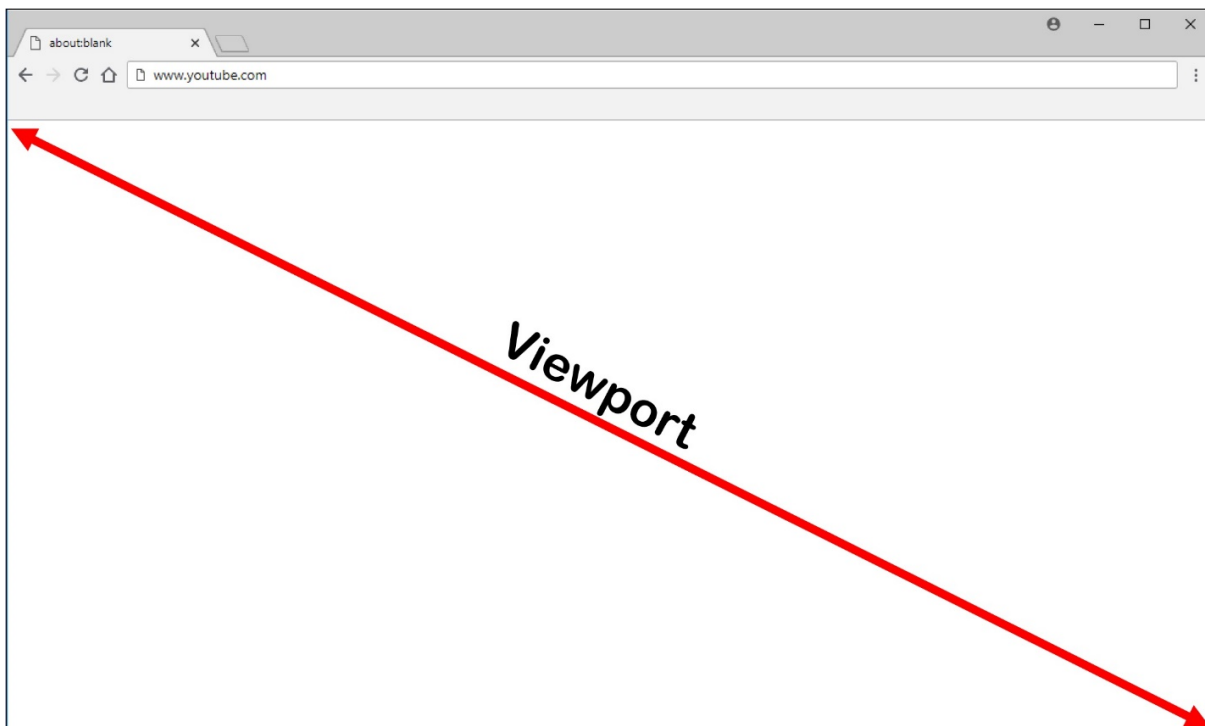
ブラウザコンテンツのリダイレクトでは、クライアント側の許可リストに登録された Web ページをレンダリングできます。この機能は、Citrix Workspace アプリを使用してクライアント側の対応するレンダリングエンジンをインスタンス化し、URL から HTTP および HTTPS コンテンツを取得します。

#### 注：

Linux VDA は、Google Chrome でのブラウザコンテンツの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。

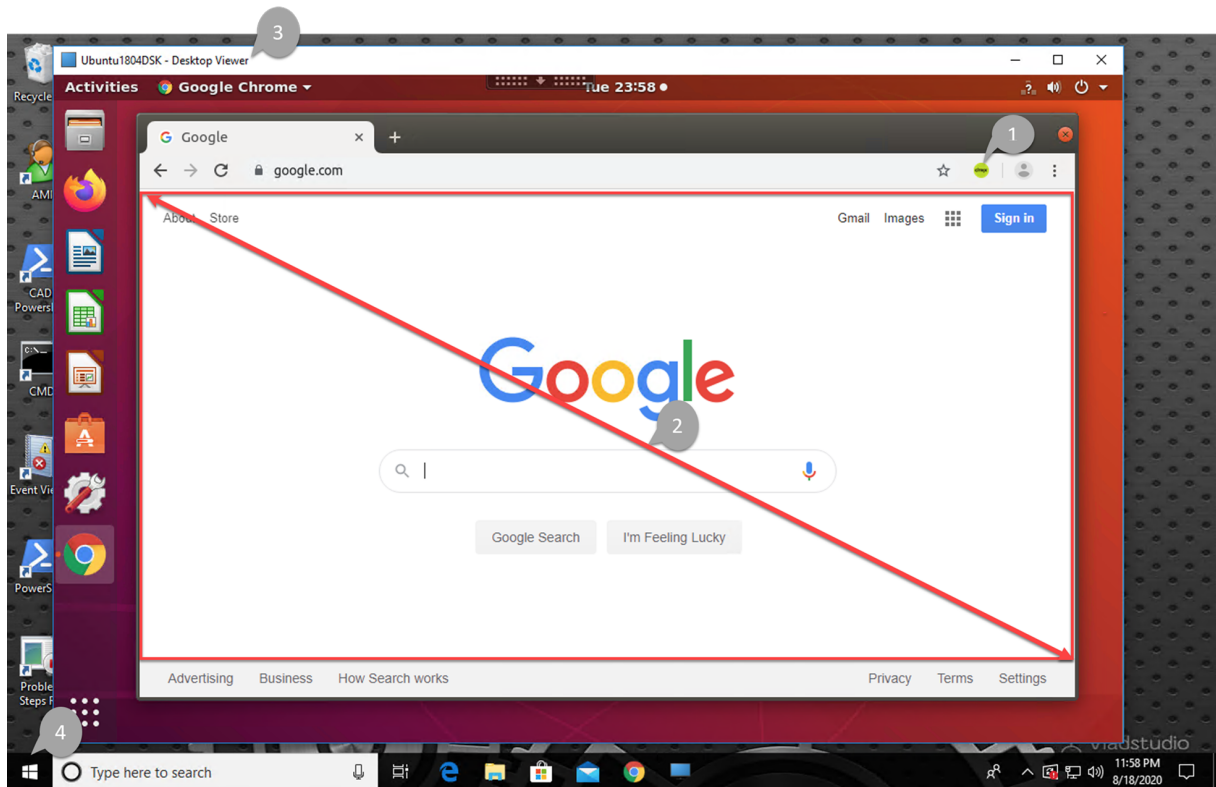
ブラウザのビューポートだけがリダイレクトされます。ビューポートは、コンテンツが表示されるブラウザ内の長方形の領域です。ビューポートには、アドレスバー、お気に入りバー、ステータスバーなどは含まれません。これらの項目は引き続き VDA の Web ブラウザーで実行されます。



リダイレクト用の許可リストに登録された URL を含む、アクセス制御リストを指定する Studio ポリシーを設定します。特定の URL についてリダイレクトを無効にする禁止リストを構成します。

許可リスト内に一致する URL があり、禁止リスト内にはない場合、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 をリレーします。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。

Citrix Workspace アプリは、Web サイトを仮想デスクトップブラウザのコンテンツ領域にシームレスにブレンドします。



1. Citrix ブラウザーコンテンツのリダイレクト拡張機能のアイコン

Chrome 拡張機能のアイコンの色は、ステータスを指定します。以下の3つの色のいずれかです：

- 緑：アクティブで接続されています
- グレー：現在のタブではアクティブではないかアイドル状態です
- 赤：壊れているか動作していません

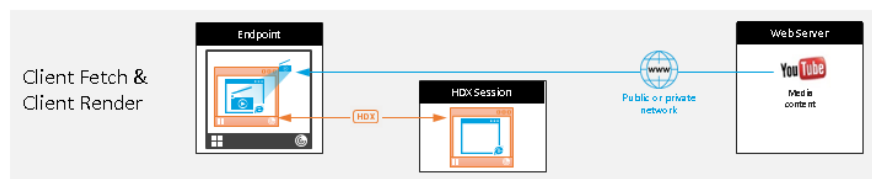
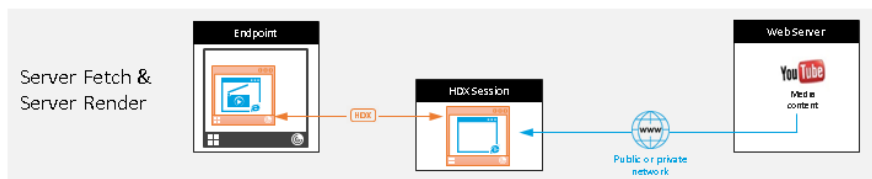
2. クライアントでレンダリングされた、または仮想デスクトップにブレンドされたビューポート

3. Linux VDA

4. Windows クライアント

Citrix Workspace アプリがコンテンツをどのようにフェッチするかシナリオを次に示します：

## Redirection scenarios



### Benefits:

- Better end user experience (Adaptive Bit Rate (ABR))
- Reduced VDA resource usage (CPU/RAM/IO)
- Reduced bandwidth consumption

- 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登録していないか、リダイレクトに失敗したため、リダイレクトはありません。VDA 上での Web ページのレンダリングに戻り、Thinwire を使用してグラフィックスを遠隔操作します。ポリシーを使用してフォールバックの動作を制御します。このシナリオでは、VDA での CPU、RAM、および帯域幅の消費量が多くなります。
- 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、Web サーバーから VDA を通じてコンテンツに接続し、フェッチします。このオプションは、クライアントが Web サーバーへアクセスできない場合 (シンクライアントなど) に便利です。VDA での CPU と RAM の消費量は少なくなりますが、ICA 仮想チャネルでは帯域幅が消費されます。

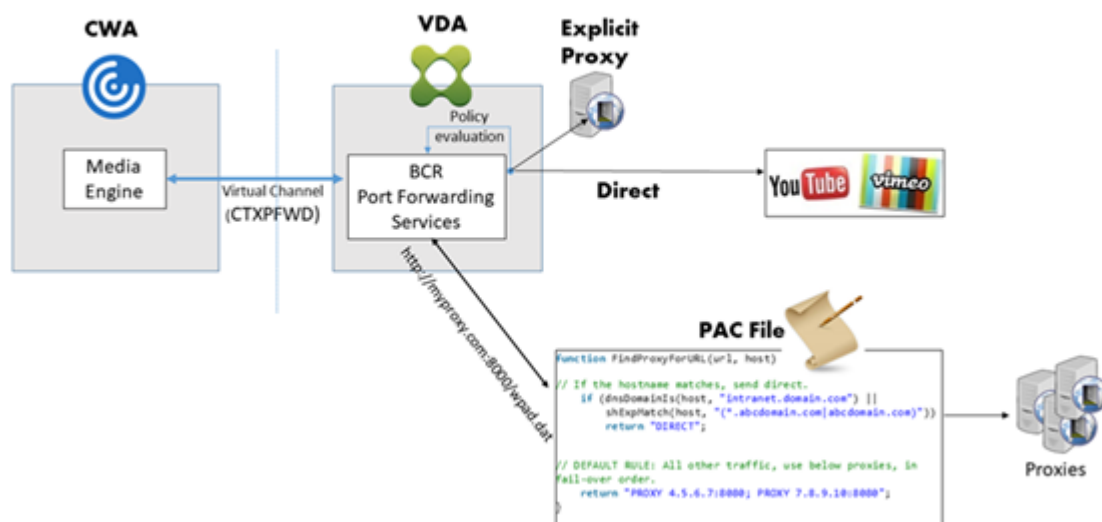
このシナリオには 3 つの動作モードがあります。CTXPFWD は、VDA が Web サーバーへアクセスできるようになるためにアクセスするプロキシデバイスに、データを転送します。

選択可能なポリシーオプション:

- Explicit Proxy - データセンターに単一の明示的なプロキシがある場合。
- Direct or Transparent - プロキシがない場合、または透過プロキシを使用している場合。
- PAC files - PAC ファイルに依存して、指定された URL のフェッチに VDA のブラウザーが適切なプロキシサーバーを自動で選択できる場合。

詳しくは、この記事で後述する「ブラウザーコンテンツリダイレクトのプロキシ構成」設定を参照してください。





- クライアントフェッチとクライアントレンダリング: Citrix Workspace アプリは Web サーバーに直接接続するため、インターネットにアクセスする必要があります。このシナリオでは、Citrix Virtual Apps and Desktops サイトからネットワーク、CPU、および RAM の使用量をすべてオフロードします。

長所:

- エンドユーザーエクスペリエンスの向上 (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)
- 消費帯域幅の削減

## システム要件

### Windows クライアント:

- Windows 向け Citrix Workspace アプリ 1809 以降

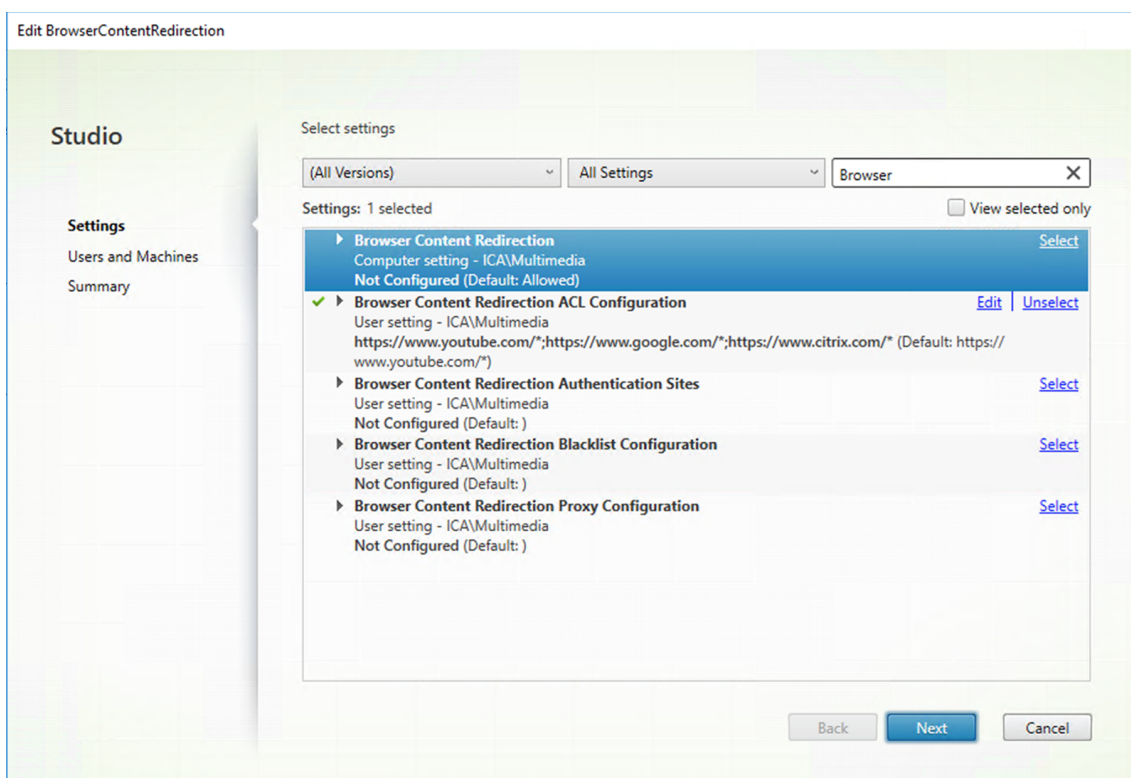
### Linux VDA:

- VDA のブラウザ: Citrix ブラウザーコンテンツの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のリダイレクトの構成

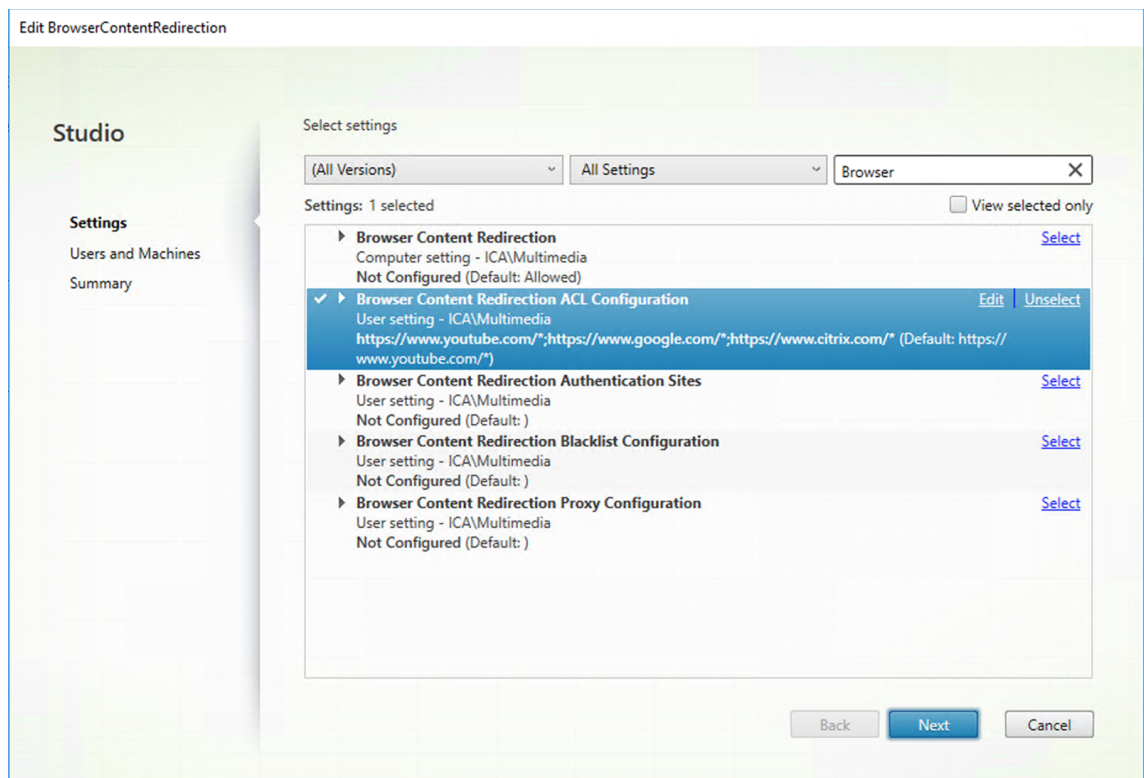
ブラウザコンテンツのリダイレクトを使用するには、関連するポリシーを構成し、Google Chrome にブラウザコンテンツのリダイレクト拡張機能をインストールします。このためには、次の手順を実行します:

1. Citrix Studio で、ブラウザーコンテンツのリダイレクトを有効にするには、[ブラウザーコンテンツのリダイレクト] を [許可] に設定します。



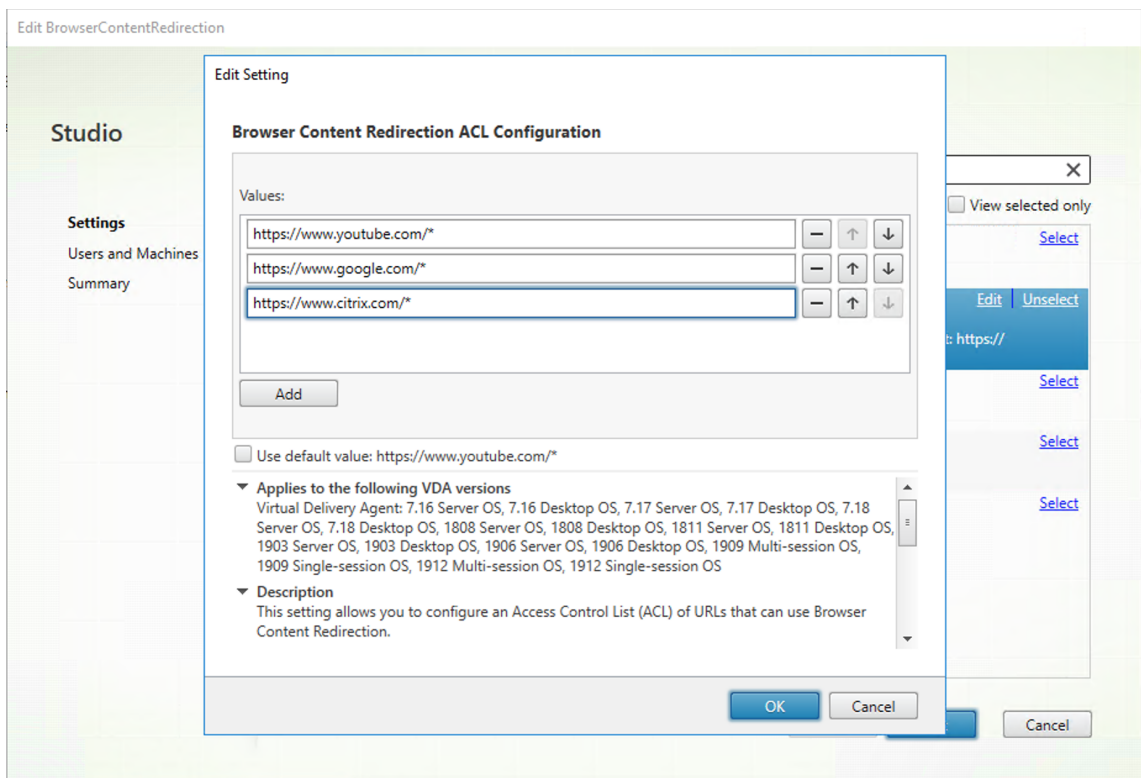
2. コンテンツをクライアントにリダイレクトできる URL の許可リストと、特定の URL のリダイレクトを無効にする禁止リストを指定します。禁止リストを構成することはオプションです。

[ブラウザーコンテンツリダイレクトの **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 を指定する場合、ワイルドカード\*を使用して、プロトコルを除くすべての URL コンポーネントを表すことができます。

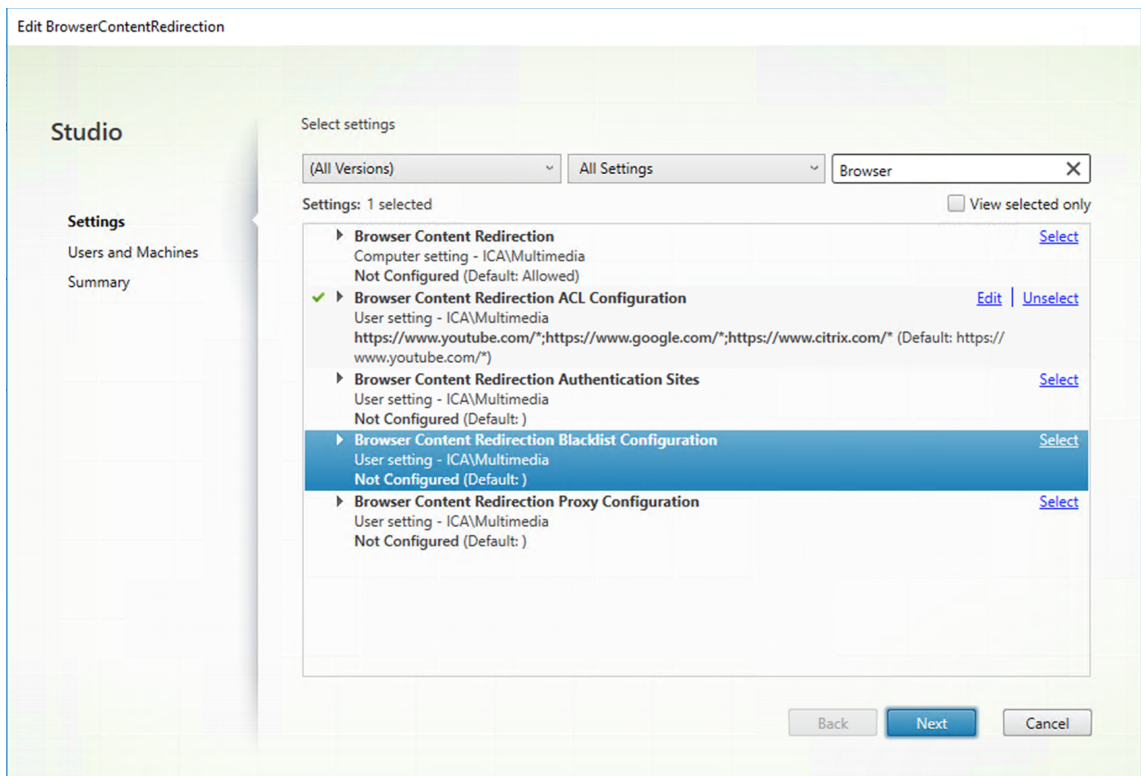


以下は許可されている例です：

- `http://www.xyz.com/index.html` (URL にパスを指定することにより、より細分化することができます。たとえば、`https://www.xyz.com/sports/index.html` を指定すると、`index.html` ページのみがリダイレクトされます。)
- `https://www.xyz.com/*`
- `http://www.xyz.com/*videos*`
- `http://*.xyz.com/`
- `http://*.*.com/`



[ブラウザコンテンツリダイレクトのブラックリスト構成] 設定は、特定の URL についてリダイレクトを無効にする禁止リストを指定します。



3. サーバ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、[ブラウザコンテンツリダイレクトのプロキシ構成] 設定を構成します。

この設定は、Web ブラウザーコンテンツリダイレクト用の VDA でのプロキシ設定のオプションです。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/透過型の設定を指定して有効にすると、Citrix Workspace アプリは常にサーバフェッチとクライアントレンダリングを最初に試行します。詳しくは、「フォールバックのメカニズム」を参照してください。

無効にした場合または構成していない場合にデフォルト値を使用すると、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。

明示的なプロキシで許可されたパターン:

`http://\<hostname/ip address>:\<port>`

例:

`http://proxy.example.citrix.com:80 http://10.10.10.10:8080`

**PAC/WPAD** ファイルで許可されたパターン:

`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`

直接または透過型のプロキシで許可されたパターン:

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\WebBrowserRedirectionProxyAd` . を編集してプロキシを設定することもできます。また、レジストリ値 `HKLM\Software\Citrix\HdxMediastream\AllowNonTlsPacUri` を使用すると、HTTP を介した 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、これは HTTP の使用が許可されていないことを意味します。

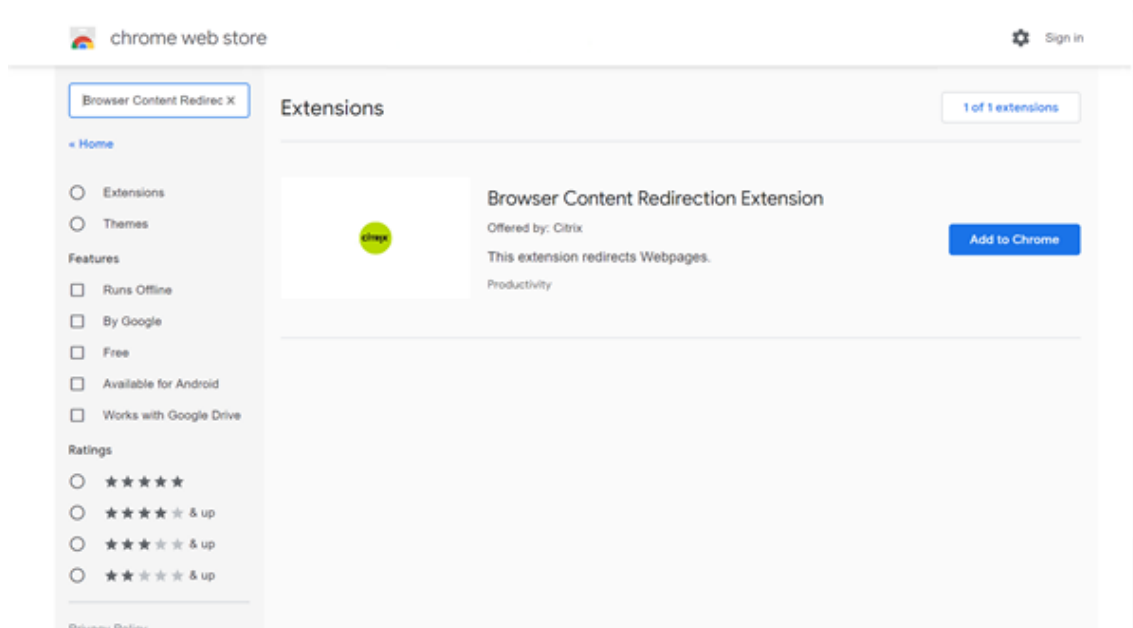
レジストリは、ポリシー設定のオプションを上書きします。関連するレジストリキーのリストについては、「Web ブラウザーコンテンツリダイレクトのレジストリキーの上書き」を参照してください。

4. VDA の [Chrome に追加] をクリックし、Chrome ウェブストアから Citrix ブラウザーコンテンツのリダイレクト拡張機能を追加します。これは、VDA 上のブラウザーが、(移動先の) URL が許可リストまたは禁止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:

この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 の拡張機能は、ユーザーごとにインストールします。拡張機能を追加または削除する場合に、ゴールデンイメージを更新する必要はありません。



### フォールバックのメカニズム

[ブラウザコンテンツリダイレクトのプロキシ構成] 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みます。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は、クライアントフェッチとクライアントレンダリングに戻ります。クライアントマシンが Web サーバーへアクセスできない場合、VDA 上のブラウザがサーバー上でページをリロードして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）。

### Web ブラウザーコンテンツリダイレクトのレジストリキーの上書き

#### 警告:

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、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場合もあります。レジストリエディターの誤用による障害に対して、Citrix は一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

HKLM\Software\Citrix\HdxMediastream

名前	種類	値
<b>WebBrowserRedirection</b>	DWORD	1= 許可、0= 禁止
<b>WebBrowserRedirectionAcl</b>	REG_MULTI_SZ	/
<b>WebBrowserRedirectionProxyAddress</b>	REG_SZ	次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。明示的なプロキシ - データセンターに単一の明示的なプロキシがある場合。直接または透過型 - プロキシがない場合、または透過プロキシを使用している場合。 <b>PAC</b> ファイル - PAC ファイルに依存して、指定された URL のフェッチに VDA のブラウザが適切なプロキシサーバーを自動で選択できる場合。
<b>WebBrowserRedirectionBlacklist</b>	REG_MULTI_SZ	/
<b>AllowNonTlsPacUri</b>	DWORD	HTTP を介した 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、これは HTTP の使用が許可されていないことを意味します。これを 1 に設定すると、 <b>HDXWebProxy.exe</b> は HTTP 経由（厳密には HTTPS 経由ではない）で PAC ファイルをダウンロードできます。

## HDX Web カメラビデオ圧縮

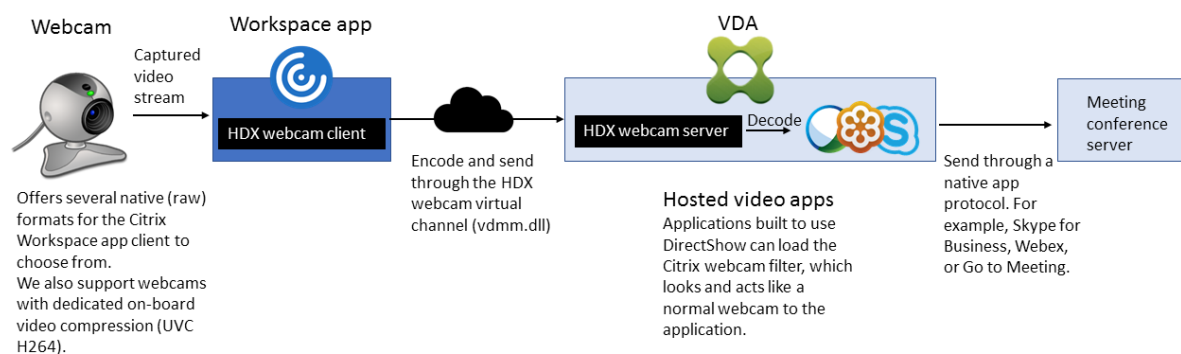
August 20, 2024

### 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 カメラを利用できるようになりました。この機能はデフォルトで有効にされています。可能であれば常に、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 Web カメラモードとも呼ばれます。このタイプの Web カメラビデオ圧縮では、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 H.264 ビデオを直接送信します。HDX Web カメラビデオ圧縮では、クライアントオペレーティングシステムに含まれるマルチメディアフレームワークテクノロジーにより、キャプチャデバイスのビデオをインターセプトし、トランスコードおよび圧縮します。各キャプチャデバイスの製造元から、OS カーネルのストリーミングアーキテクチャに組み込まれるドライバーが提供されています。

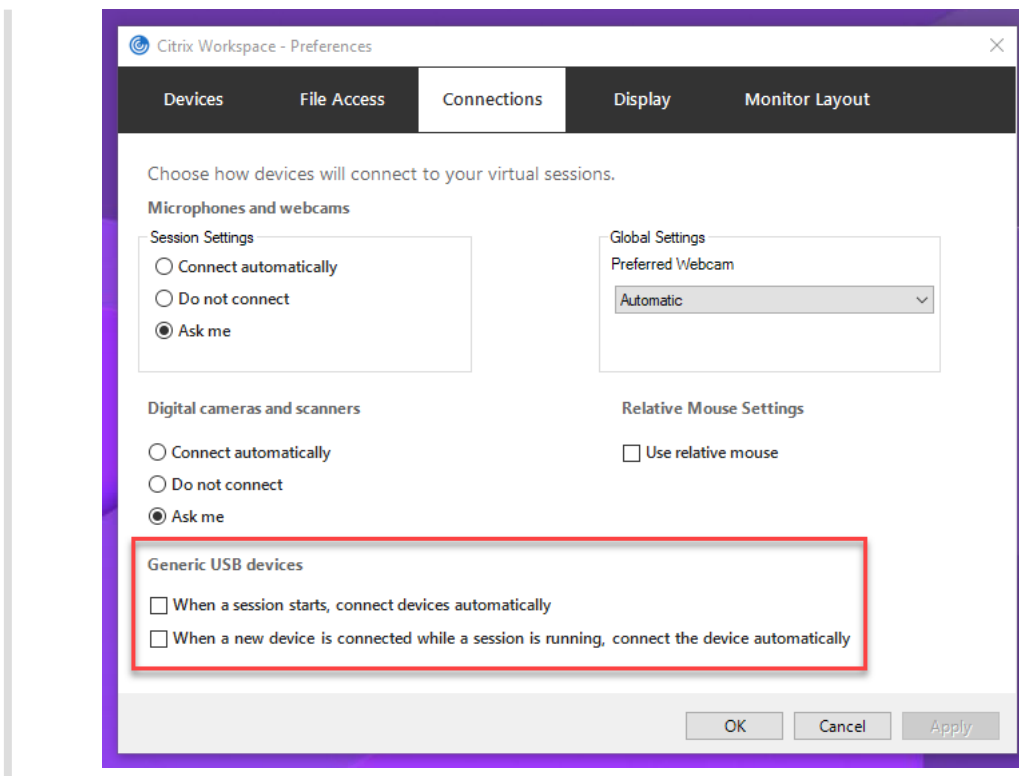
クライアントは、Web カメラとの通信を処理します。その後、サーバーで適切に表示できるビデオのみを、サーバーに送信します。サーバーが Web カメラと直接やり取りをするわけではありませんが、統合によりデスクトップでも同様のエクスペリエンスが得られます。Citrix Workspace アプリがビデオを圧縮するため、帯域幅が節約され、WAN シナリオでの回復性の向上します。



#### 注:

- この機能は、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 マシンでは使用できません。
- 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。
- サポートされている Web カメラの解像度は 48x32 から 1920x1080 の範囲です。
- Web カメラを使用している場合、Citrix Workspace アプリのツールバーの [汎用 **USB** デバイス] は選択しないでください。選択すると、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 Web カメラのビデオ圧縮は、次のバージョンの Citrix Workspace アプリをサポートします：

プラットフォーム

プロセッサ

Windows 向け Citrix Workspace アプリ

Windows 向け Citrix Workspace アプリは、XenApp および XenDesktop 7.17 以降上の 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ます。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した。

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、最適化された HDX Web カメラビデオ圧縮を使用できるのは 32 ビットアプリのみです。

完全にテスト済みの **Web** カメラ

Web カメラが異なれば、フレームレートや、明るさとコントラストのレベルも異なります。Citrix 製品では、初期の機能検証に次の Web カメラを使用します：

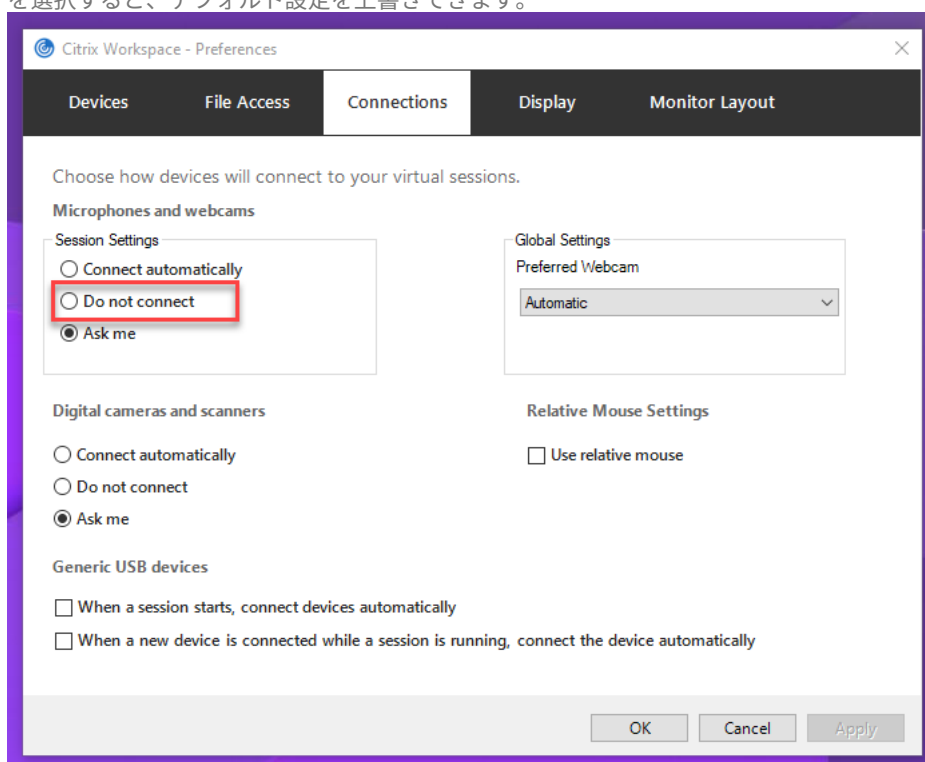
- Logitech HD Webcam C270
- Logitech Webcam C930e
- Microsoft-LifeCam-HD3000

## 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、次の検証と構成を完了します：

ヒント：

Citrix Workspace アプリのユーザーは、Desktop Viewer の [マイクと **Web** カメラ] 設定で [接続しない] を選択すると、デフォルト設定を上書きできます。



1. VDA のインストールが完了したら、VDA が Delivery Controller に登録でき、公開された Linux デスクトップセッションが Windows 資格情報を使用して正常に起動できることを確認します。
2. VDA がインターネットにアクセスできることを確認してから、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh` コマンドを実行して Web カメラの構成を完了します。VDA にインターネットアクセスがない場合は、手順 3 に進みます。

注：

`uname -r` とカーネルヘッダーの間でカーネルの不一致が発生することがあります。不一致があると、`ctxwcamcfg.sh` スクリプトが失敗します。HDX Web カメラの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

トを再実行します。

VDA が Debian に展開されている場合は、最新のカーネルバージョンで実行されていることを確認してください。それ以外の場合は、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ます：

```
1 sudo apt-get update
2 sudo apt-get dist-upgrade
3 sudo reboot
```

VDA が SUSE 15 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、再起動します：

```
1 zypper up kernel-default
2 reboot
```

ctxwcamcfg.sh スクリプトは、次のことに役立ちます：

a) `kernel-devel` および動的カーネルモジュールサポート (Dynamic Kernel Module Support: DKMS) プログラムを VDA に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：

上記のプログラムを RHEL および CentOS にインストールすると、`ctxwcamcfg.sh` スクリプトがインストールされ、VDA の次のリポジトリが有効になります。

- Extra Packages for Enterprise Linux (EPEL)
- RPM Fusion

b) <https://github.com/umlaeute/v4l2loopback> からオープンソースコード `v4l2loopback` をダウンロードし、DKMS を使用して `v4l2loopback` を管理します。

`v4l2loopback` は、V4L2 ループバックデバイスを作成できるカーネルモジュールです。

c) `sudo systemctl restart ctxwcamsd` コマンドを実行します。Linux VDA の Web カメラサービス、`ctxwcamsd` は、HDX Web カメラビデオ圧縮機能の `v4l2loopback` カーネルモジュールを再起動してロードします。

3. VDA にインターネットアクセスがない場合は、別のマシンで `v4l2loopback` カーネルモジュールをビルドしてから、VDA にコピーします。

- a) インターネットにアクセスでき、かつ VDA と同じカーネルバージョンのマシンを準備します。`uname -r` コマンドは、カーネルのバージョンを見つけるのに役立ちます。
- b) マシンで `sudo mkdir -p /var/xdl` コマンドを実行します。
- c) `/var/xdl/configure_*` を、VDA から `/var/xdl/` のマシンにコピーします。

- d) 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/`パスに `v4l2loopback.ko` ファイルが作成されます。`ctxwcamcfg.sh` スクリプトを実行するときに発生する可能性のあるエラーは無視します。
- e) `v4l2loopback.ko` をマシンから VDA にコピーして、`/opt/Citrix/VDA/lib64/`に配置します。
- f) VDA で、`sudo systemctl restart ctxwcamd` コマンドを実行して Web カメラサービスを再起動し、`v4l2loopback` カーネルモジュールをロードします。

## ドメイン非参加の Linux VDA

July 16, 2024

### 概要

ドメイン非参加の VDA を使用すると、VDA およびユーザー認証用の Active Directory ドメインに VDA を参加させる必要がなくなります。ドメイン非参加の VDA を作成するときは、VDA をクラウドコントロールプレーンに登録するための公開キーと秘密キーのペアを生成します。これにより、Active Directory ドメインへの参加は不要になりました。ユーザーがドメイン非参加の VDA からセッションを開始すると、VDA は、Citrix Workspace アプリへのログオンのためにユーザーが使用するユーザー名を使ってローカルマッピングアカウントを作成します。VDA は、SSO とセッションの再接続のためにローカルにマップされるアカウントが使用するランダムパスワードを割り当てます。ランダムパスワードを変更すると、SSO とセッションの再接続に失敗します。SSO を無効にするには、「[SSO 以外の認証](#)」を参照してください。

#### 重要:

- ドメイン非参加 VDA は、Citrix DaaS でサポートされます。
  - コントロールプレーンは、Citrix DaaS 経由で展開する必要があります。
  - ドメイン非参加 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せん。ドメイン非参加 VDA は、Citrix DaaS のコントロールプレーンによって管理されます。
  - ドメイン非参加 VDA を作成するには、[Rendezvous V2](#) を有効にします。Cloud Connector: オンプレミスハイパーバイザーでマシンをプロビジョニングする予定の場合、または Workspace で Active Directory を ID プロバイダーとして使用する場合にのみ必要です。
- ドメイン非参加の VDA を作成する場合、Machine Creation Services (MCS) と簡単インストールの両方を使用できます。詳しくは、「[MCS を使用したドメイン非参加 Linux VDA の作成](#)」および「[簡単インストールを使用したドメイン非参加 Linux VDA の作成](#)」を参照してください。

- MCS はベアメタルサーバーをサポートしていません。

## ドメイン非参加の **Linux VDA** で使用できる機能

ドメイン非参加の **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

ドメイン非参加の 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。また、ユーザーのユーザー識別子 (UID)、グループ識別子 (GID)、ホームディレクトリ、ログインシェルなどのユーザー属性を指定することもできます。この機能を使用するには、次の手順を実行します：

1. 次のコマンドを実行して、この機能を有効にします：

```
1 /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	必須かオプションか	説明
<code>uid</code>	必須	ユーザー識別子 (UID) は、Linux によってシステム上の各ユーザーに割り当てられる番号です。ユーザーがアクセスできるシステムリソースを決定します。
<code>gid</code>	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。
<code>homedir</code>	オプション	Linux ホームディレクトリは、特定のユーザー用のディレクトリです。
<code>shell</code>	オプション	ログインシェルは、ユーザーアカウントへのログイン時にユーザーに提供されるシェルです。

次に、`getuidgid.sh` スクリプトの例を示します：

注：

スクリプトで指定する属性が有効であることを確認してください。

```
1 #!/bin/bash
2
```

```
3 #####
4 #
5 # Citrix Virtual Apps & Desktops For Linux Script: Get uid and gid
  # for the user
6 #
7 # Copyright (c) Citrix Systems, Inc. All Rights Reserved.
8 #
9
10 export LC_ALL="en_US.UTF-8"
11
12 function get_uid_gid_for_user()
13 {
14
15 echo "uid:12345"
16 echo "gid:1003"
17 echo "homedir:/home/$1"
18 echo "shell:/bin/sh"
19 }
20
21
22 get_uid_gid_for_user $1
```

### SSO 以外の認証

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、1 つの資格情報のセットを使用して Citrix Workspace アプリと VDA セッションにログオンします。ユーザーが別の資格情報のセットを使用して VDA セッションにログオンできるようにするには、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非参加 VDA は、AAD の ID サービスと統合され、ユーザー認証を提供します。詳しくは、「[Azure Active Directory での認証](#)」を参照してください。

### Rendezvous V2

ドメイン非参加 VDA は、Rendezvous V2 を使用した Citrix Cloud Connector のバイパス用として、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。

### ポリシーサポート一覧

August 20, 2024

**Linux VDA** ポリシーサポート一覧

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クライアントのローカルタイムゾーンを使用する	UseLocalTimeOfClient	ブール値	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 往復測定	IcaRoundTripCheckEnabled	ブール値	ICA\ユーティリティ	有効 (1)
ICA 往復測定間隔	IcaRoundTripCheckPeriod	整数	ICA\ユーティリティ	15
アイドル接続の ICA 往復測定	IcaRoundTripCheckWhenIdle	ブール値	ICA\ユーティリティ	無効 (0)
セッション全体の最大帯域幅	LimitOverallBw	整数	ICA\帯域幅	0

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

オーディオダイレクタの最大帯域幅 <b>(Kbps)</b>	LimitAudioBw	ワザ	ICA\帯域幅	0
-----------------------------------	--------------	----	---------	---

オーディオダイレクタの最大帯域幅 <b>(%)</b>	LimitAudioBwPercent	ワザ	ICA\帯域幅	0
--------------------------------	---------------------	----	---------	---

<b>USB</b> デバイスリダイレクトの最大帯域幅 <b>(Kbps)</b>	LimitUSBDevBw	ワザ	ICA\帯域幅	0
---	---------------	----	---------	---

<b>USB</b> デバイスリダイレクトの帯域幅 <b>(%)</b>	LimitUSBDevBwPercent	ワザ	ICA\帯域幅	0
--	----------------------	----	---------	---



Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

ファイ ルリダ イレク トの最 大帯域 幅	LimitCdmBwザ	—	ICA\帯 域幅	0
--------------------------------------	-------------	---	-------------	---

(Kbps)

ファイ ルリダ イレク トの最 大帯域 幅 (%)	LimitCdmBwPer	—	ICA\帯 域幅	0
--	---------------	---	-------------	---

プリン ターリ ダイレ クトの 最大帯 域幅	LimitPrinterBw	—	ICA\帯 域幅	0
---------------------------------------	----------------	---	-------------	---

(Kbps)

プリン ターリ ダイレ クトの 最大帯 域幅 (%)	LimitPrinterBwPer	—	ICA\帯 域幅	0
--	-------------------	---	-------------	---

**WebSockets** ConceptWebSocketsICAWebSockets

接続	ユーザ
	—

**WebSockets** WebSocketsPort ICA\WebSockets

ポート 番号	ユーザ
	—

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

**WebSockets** **TrustedOriginSendAllWebSockets**

信頼される接続元サーバー一覧		ユーター		
		—		

**ICA** **SendICAKeepAliveICA** **ICA** **ICA**

<b>Keep-Alive</b>		ユーター	<b>Keep-Alive</b>	Keep-Alive メッセージ (0)を送信しない
		—		

**ICA** **ICAKeepAliveTimeoutICA** **ICA** **ICA**

<b>Keep-Alive</b>		ユーター	<b>Keep-Alive</b>	60 秒
		—		

**ICA** **IcaListenerPortNumberICA** **ICA** **ICA**

<b>ICA</b>		ユーター		1494
		—		

**HDX** **HDXoverUDPICA** **ICA** **ICA**

<b>HDX</b>		ユーター		優先 (2)
		—		

**RendezvousProtocolICA** **ICA** **ICA**

<b>RendezvousProtocol</b>		ユーター		禁止
		—		

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
セッション画面の保持	AcceptSessionReliabilityConnections	ブール値	セッション画面の保持	許可 (1)
再接続の透過レベル	ReconnectionDiscrepancyLevel	ブール値	クライアントの自動接続	30%
セッション画面の保持のポート番号	SessionReliabilityPort	整数	セッション画面の保持	2598
セッション画面の保持のタイムアウト	SessionReliabilityTimeout	整数	セッション画面の保持	180 秒
クライアントの自動再接続	AllowAutoClientReconnect	ブール値	クライアントの自動接続	許可 (1)
クライアントオーディオダイレクト	AllowAudioRedirection	ブール値	オーディオ	許可 (1)

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クライアントプリンターダイレクト	AllowPrinterRedir	印刷	印刷	許可 (1)
PDFユーティリティを自動作成する	AutoCreatePDFPrinter	印刷	印刷	無効 (0)
プリンターライバリのマッピングと互換性	DriverMappingList	印刷	印刷	" Microsoft XPS  Document  Writer *, Deny ; Send to Microsoft  OneNote *, Deny "

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クライアントクリップボードリダイレクト	AllowClipboardRedirection	ブール	クリップボード	許可 (1)
クライアントからのクリップボードの転送サイズを制限する	LimitClipboardTransfer	整数	C2H	無効 (0)
クライアントからのクリップボードのセッションからのクライアントへの転送サイズを制限する	LimitClipboardTransfer	整数	H2C	無効 (0)

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クリップボードの最大帯域幅 (Kbps)	LimitClipboardBWP	帯域幅	ICA\帯域幅	0
クリップボードの最大帯域幅 (%)	LimitClipboardPercent	帯域幅	ICA\帯域幅	0
クライアントクリップボードの書き込み制限	RestrictClipboardWrite	無効	プロトコルド	(0)
クライアントクリップボードに書き込みを許可する形式	ClientClipboardWriteAllowedFormats	プロトコルド	プロトコルド	

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
セッションクリップボードの書き込み制限	RestrictSessionClipboardWrite	—	プロパティ	無効 (0)
セッションクリップボードに書き込みを許可する形式	SessionClipboardWriteAllowedFormats	—	プロパティ	
クライアント USB デバイスリダイレクト	AllowUSBRedirect	—	USB	禁止 (0)
クライアント USB デバイスリダイレクト規則	USBDeviceRules	—	USB	” “
動画圧縮	MovingImageCompression	—	プロパティ	有効 (1)
エクストラ色圧縮	ExtraColorCompression	—	プロパティ	無効 (0)

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
保持する最低フレーム数	TargetedMinimumFramesPerSecond	—	Thinwire	30
ターゲットットフレーム数	FramesPerSecond	—	Thinwire	30fps
表示品質	VisualQuality	—	Thinwire	中 (3)
圧縮にビデオコードックを使用する	VideoCodec	—	Thinwire	選択された場合使用する (3)
ビデオコードックにハードウェアエンコーディングを使用します	UseHardwareEncoding	—	Thinwire	有効 (1)
視覚的無損失の圧縮を使用する	AllowVisuallyLosslessCompression	—	Thinwire	無効 (0)



Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

3D 画像ワークロードの最適化	OptimizeFor3DWorkload	ブール	Thinwire	無効 (0)
-----------------	-----------------------	-----	----------	--------

簡素なグラフィックに対する優先的色の解像度	PreferredColorDepth	ブール	Thinwire	24 ビット/ピクセル (1)
-----------------------	---------------------	-----	----------	-----------------

グラフィック状態インジケータ	DisplayLossesIndicators	ブール	Thinwire	無効 (0)
----------------	-------------------------	-----	----------	--------

音質	SoundQuality	ブール	オーディオ	高 - 高品位オーディオ (2)
----	--------------	-----	-------	------------------

クライアントマイクリダイレクト	AllowMicrophoneRedirection	ブール	オーディオ	許可 (1)
-----------------	----------------------------	-----	-------	--------

最大セッション数	MaximumNumberOfSessions	整数	管理	250
----------	-------------------------	----	----	-----

同時ログオンストレンス	ConcurrentLogonsPerUser	整数	管理	2
-------------	-------------------------	----	----	---

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
コンローラの自動更新を有効にする	EnableAutoUpdate	ブール値	OfficeControl	許可 (1)
クリップボードの選択更新モード	ClipboardSelectionUpdateMode	ブール値	Clipboard	ブロード
プライマリ選択更新モード	PrimarySelectionUpdateMode	ブール値	Clipboard	ブロード
スピーク最大品質	MaxSpeechQuality	整数	オーディオ	5
クライアントドライブに自動接続する	AutoConnectDrives	ブール値	ファイリダイレクト\CDM	有効 (1)
クライアント側光学式ドライブ	AllowCDROMDrives	ブール値	ファイリダイレクト\CDM	許可 (1)
クライアント側固定ドライブ	AllowFixedDrives	ブール値	ファイリダイレクト\CDM	許可 (1)

## Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クライアント側フロッピードライブ	AllowFloppyDrives	ファイ	ルリダイレクタ\CDM	許可 (1)
クライアント側ネットワークドライブ	AllowNetworkDrives	ファイ	ルリダイレクタ\CDM	許可 (1)
クライアント側リムーバブルドライブ	AllowRemovableDrives	ファイ	ルリダイレクタ\CDM	許可 (1)
クライアント側ドライブリダイレクト	AllowDriveRedirect	ファイ	ルリダイレクタ\CDM	許可 (1)
クライアント側の読み取り専用アクセス	ReadOnlyMappedDrive	ファイ	ルリダイレクタ\CDM	無効 (0)

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
キーボードの自動表示	AllowAutoKeyboardPopUp	—	KeyboardPopUp	無効 (0)
デスクトップとクライアント間のファイル転送を許可する	AllowFileTransfer	—	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	—	ファイ ル転送	許可
デスクトップにファイルをアップロード	AllowFileUpload	—	ファイ ル転送	許可
セッションアイドルタイム	EnableSessionIdleTimer	—	セッションタイマー	有効 (1)

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
セッションアイドルタイムの間の隔	SessionIdleTimerInterval	—	セッションタイマー	1,440 分
切断セッションタイマー	EnableSessionDisconnectTimer	—	セッションタイマー	無効 (0)
切断セッションタイマーの間隔	SessionDisconnectTimerPeriod	—	セッションタイマー	1440 分
ブラウザのコンテンツリダイレクト	WebBrowserRedirectACL	ユーティリティ	ルチメディア	許可
ブラウザのコンテンツリダイレクトのACL構成	WebBrowserRedirectACL	—	ルチメディア	https://www.youtube.com/*

**ACL** 構成

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

**Web** WebBrowserRedirectorAuthenticationSites

ブラウザーコンテンツツリダイレクト認証サイト	—		ルチメディア	
------------------------	---	--	--------	--

**Web** WebBrowserRedirectorBlacklist

ブラウザーコンテンツツリダイレクトの禁止リスト構成	—		ルチメディア	
---------------------------	---	--	--------	--

**Web** WebBrowserRedirectorProxy

ブラウザーコンテンツツリダイレクトのプロキシ構成	—		ルチメディア	
--------------------------	---	--	--------	--

**Web** EnableSessionWatermark 無効

セッションウォーターマークの有効化	—		セッションウォーターマークの内容	
-------------------	---	--	------------------	--

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
クライアント IP アドレスを含む	IncludeClientIPAddresses	ブール値	セッションウォーカー マーク の内容	無効
接続時間を含める	IncludeConnectTime	ブール値	セッションウォーカー マーク の内容	無効
ログオンユーザー名を含む	IncludeLogonUserName	ブール値	セッションウォーカー マーク の内容	有効
VDA ホスト名を含む	IncludeVDAHostName	ブール値	セッションウォーカー マーク の内容	有効





Studio

ポリシー	キー名	種類	モジュール	デフォルト値
ウォーターマークのカスタムテキスト	WatermarkCustomText	Text	セッションウォーターマークの内容	有効なカスタムテキストの構成は次のようになります。 <tagname=fontname=Arialfontfamily=UnicodeMSfontstyle=domainfontsize=47username=newlinefontzoom=120>Citrix</tagname=newline>style=tile>Najing</tagname></td>

---

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
セッションメトリックの収集	UsageDataCollectionThroughClient	ユータ	有効	(1)
		—		

Studio

ポリシー	キー名	種類	モジュール	デフォルト値
------	-----	----	-------	--------

動的ウィンドウプレビュー	EnableDynamicPreview	ブール値	VDI	この設定では、シームレスウィンドウの表示を有効または無効にします。この設定を有効にすると、公開アプリのウィンドウで画面上のコンテンツをプレビューできます。たとえば、Windows向け Citrix Workspace アプリを使用して Google Chrome
--------------	----------------------	------	-----	---

注:

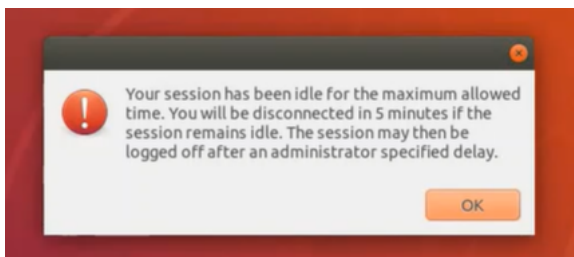
Windows Virtual Delivery Agent (VDA) のみが、User Datagram Protocol (UDP) でのオーディオ転送をサポートしています。Linux VDA ではサポートされていません。詳しくは、「[User Datagram Protocol \(UDP\) でのオーディオリアルタイムトランスポート](#)」を参照してください。

Citrix ポリシー設定を使用して、Citrix Studio のセッション接続タイマーを構成できます:

- セッションアイドルタイマー: アイドル状態のセッションに時間制限を適用するかどうかを決定します。
- セッションアイドルタイマーの間隔: アイドル状態のセッションの時間制限を設定します。セッションアイドルタイマーが [有効] になっていて、アクティブなセッションが設定された時間内にユーザー入力を受信しなかった場合、セッションは切断されます。
- 切断セッションタイマー: 切断されたセッションに時間制限を適用するかどうかを決定します。
- 切断セッションタイマーの間隔: 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。

このポリシー設定のいずれかを変更する場合は、環境全体で設定が一貫していることを確認してください。

アイドル状態のセッションの制限時間が経過すると、警告メッセージが表示されます。例として、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、アイドルタイマーをリセットするためのユーザー入力が必要です。



次のポリシーは、Citrix Studio バージョン 7.12 以降で構成できます。

- MaxSpeexQuality

値 (整数): [0-10]

デフォルト値: 5

詳細:

オーディオダイレクトで、音質が中または低の場合、オーディオデータを Speex でエンコードします (音質のポリシーを参照)。Speex は劣化を伴うコーデックであり、入力音声信号の品質を犠牲にして圧縮します。その他の音声コーデックと違い、品質とビットレートのバランスを制御できます。Speex のエンコーディングプロセスは、ほとんどの場合、0 から 10 の範囲の品質パラメーターで制御します。品質が高いほど、ビットレートも高くなります。

Speex 最大品質は、最高の Speex 品質を選択して音声品質と帯域幅制限に従ってオーディオデータをエンコードします (オーディオダイレクトおよび帯域幅制限のポリシー参照)。音声品質が中の場合、エンコーダー

は広帯域モードの、より高いサンプルレートになります。音声品質が低の場合、エンコーダーは狭帯域モードで、より低いサンプルレートになります。同じ Speex 品質でも、モードとビットレートは異なります。最高の Speex 品質は、以下の条件を満たす最大の値です。

- 品質が Speex 最大品質以下
- ビットレートが帯域幅制限以下

関連設定：音質、オーディオリダイレクトの最大帯域幅

- **PrimarySelectionUpdateMode**

値（列挙）：[0, 1, 2, 3]

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場合に使用されます。

この設定は、Linux VDA でのプライマリ選択の変更がクライアントのクリップボードで更新されるかどうかを制御します（逆の場合も同様）。値には、次の 4 つのオプションがあります：

### Primary selection update mode

Value: Selection changes are not updated on neither client nor host

Use Selection changes are not updated on neither client nor host

Host selection changes are not updated to client

Client selection changes are not updated to host

Selection changes are updated on both client and host

OS, 7.1 Desktop OS, 7.5 Server OS, 7.1 Desktop OS, 7.8 Server OS, 7.8 Desktop OS, 7.9 Server OS, 7.9 Desktop OS, 7.11 Server OS, 7.11 Desktop OS, 7.12 Server OS, 7.12 Desktop OS, 7.13 Server OS, 7.13 Desktop OS, 7.14 Server OS, 7.14 Desktop OS, 7.15 Server OS, 7.15 Desktop OS, 7.16 Server OS, 7.16 Desktop OS, 7.17 Server OS, 7.17 Desktop OS, 7.18 Server OS, 7.18 Desktop OS, 7.19 Server OS, 7.19 Desktop OS

▼ Description

This setting is supported only by Linux VDA version 1.4 onwards.

PRIMARY selection is used for explicit copy/paste actions such as mouse selection and middle mouse button paste. This setting controls whether PRIMARY selection changes on the Linux VDA can be updated on the client's clipboard (and vice versa). It can include one of the following selection changes:

Selection changes are not updated on the client or the host. PRIMARY selection changes do not update a client's clipboard. Client clipboard changes do not update PRIMARY selection.

Host selection changes are not updated on the client. PRIMARY selection changes do not update a client's clipboard. Client clipboard changes update the PRIMARY selection.

Client selection changes are not updated on the host. PRIMARY selection changes update the client's clipboard. Client clipboard changes do not update the PRIMARY selection.

Selection changes are updated on both the client and host. PRIMARY selection change updates the client's clipboard. Client clipboard changes update the PRIMARY selection.

▼ Related settings

Clipboard selection update mode

- 選択の変更はクライアントでもホストでも更新されません  
Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されることはありません。
- ホスト選択の変更はクライアントで更新されません  
Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されます。
- クライアント選択の変更はホストで更新されません  
Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されることはありません。
- 選択の変更は、クライアントとホストの両方で更新されます

Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されます。このオプションがデフォルト値です。

関連設定: クリップボード選択更新モード

- ClipboardSelectionUpdateMode

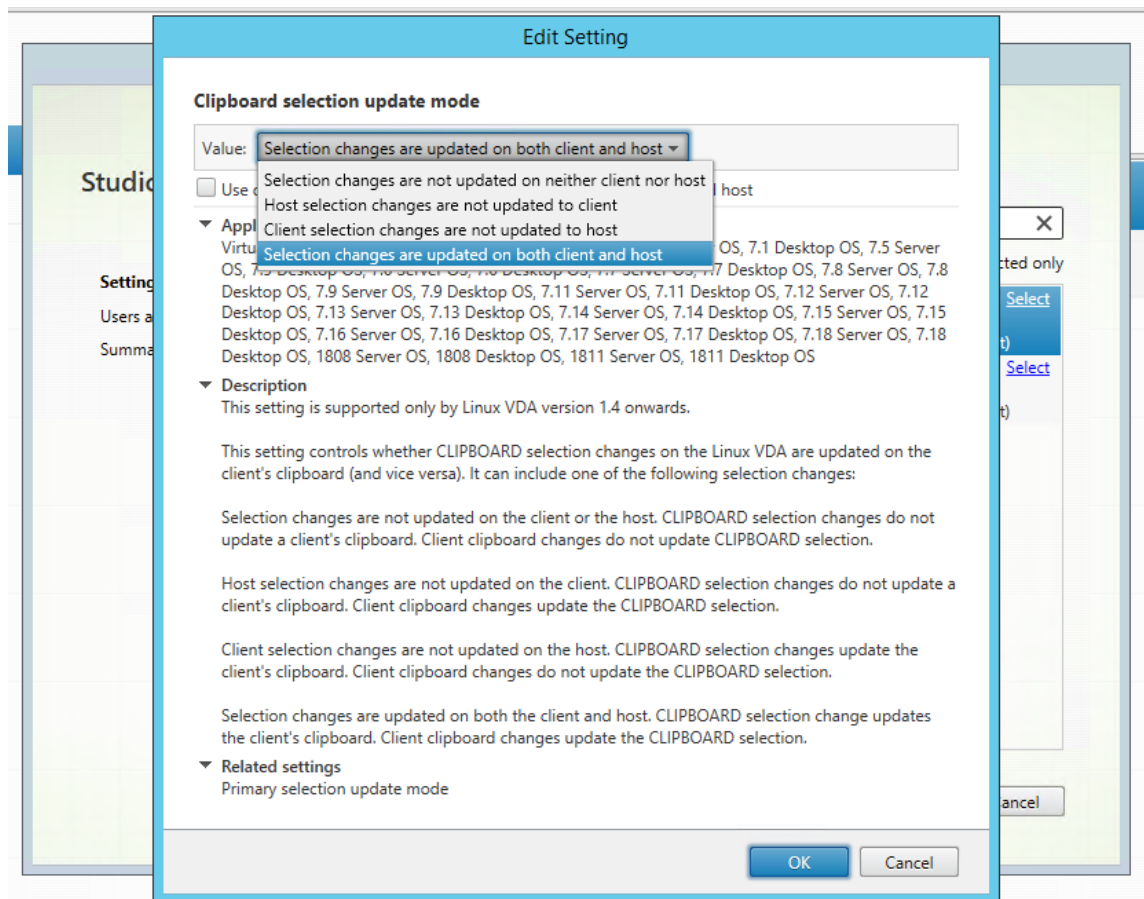
値 (列挙): [0, 1, 2, 3]

デフォルト値: 3

詳細:

クリップボード選択は、いくつかのデータを選択し、ショートカットメニューの「コピー」を選択するなど、クリップボードにコピーされることを明示的に要求する場合に使用します。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 特有の操作です。

このポリシーは、Linux VDA でのクリップボード選択の変更がクライアントのクリップボードで更新されるかどうかを制御します (逆の場合も同様)。値には、次の 4 つのオプションがあります:



- 選択の変更はクライアントでもホストでも更新されません

Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードは更新されませ

ん。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されることはありません。

- ホスト選択の変更はクライアントで更新されません

Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されます。

- クライアント選択の変更は、ホストで更新されません

Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されることはありません。

- 選択の変更は、クライアントとホストの両方で更新されます

Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されます。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 では、クリップボード選択とプライマリ選択の両方がサポートされています。Linux VDA とクライアント間のコピーおよび貼り付けの動作を制御するには、クリップボード選択更新モードとプライマリ選択更新モードの両方を同じ値に設定することをお勧めします。

## 印刷

July 16, 2024

このセクションでは、以下のトピックについて説明します:

- [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

August 20, 2024

ここでは、印刷のベストプラクティスについて説明します。



## インストール

Linux VDA では、**cups** フィルターと **foomatic** フィルターの両方が必要です。フィルターは VDA とともにインストールされます。フィルターは、ディストリビューションに基づいて手動でインストールすることもできます。

## 印刷のポリシー設定

### クライアントプリンターリダイレクト

この設定項目は、クライアントプリンターを VDA セッションにマップするかどうかを決定できるようにします。デフォルトでは許可されます。

### クライアントプリンターを自動作成する

この設定項目では、VDA セッションにマップできるクライアントプリンターを指定します。デフォルトでは、[すべてのクライアントプリンターを自動作成する] に設定されています。これは、すべてのクライアントプリンターが VDA セッションにマップされることを意味します。この設定について詳しくは、Citrix Virtual Apps and Desktops ドキュメントの「[クライアントプリンターを自動作成する](#)」を参照してください。

### PDF ユニバーサルプリンターを自動作成する

**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可能性があります。この場合、以前のリリースでの唯一のオプションは、`~/.CtxlpProfile$CLIENT_NAME` 構成ファイルを編集することでした。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オプションが追加されています。

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。Linux 向け Citrix Workspace アプリを使用している場合は、代わりにプリンター名を入力します。
4. 必要に応じて次のアクションを実行します:

- クライアントプリンターを VDA セッションにリダイレクトし、ユニバーサルプリンタードライバーのみを使用するには、[許可] または [ユニバーサルドライバーのみで作成する] を選択します。
- 指定されたドライバー名を持つクライアントプリンターが VDA セッションにリダイレクトされないようにするには、[作成しない] を選択します。
- サーバーと同じドライバーを持つもののドライバー名が異なるクライアントプリンターに、サーバーアプリケーションがアクセスできるようにするには、[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。

## Edit Setting

## Printer driver mapping and compatibility

Driver Name	Action	Settings	Se
Microsoft XPS Document Writer *	Do not create	Unmodified	
Send to Microsoft OneNote *	Do not create	Unmodified	
FX FPS Color Class Driver	Replace with	/u	

**Edit driver mapping** [Close]

Driver Name:

Allow  
 Do not create  
 Create with universal driver only  
 Replace with

Use default v

▼ Applies to the  
 Virtual Deliv  
 Server OS, 7.  
 Server OS, 7.  
 Server OS, 7.  
 7.15 Server C  
 OS, 7.18 Server OS, 7.18 Desktop OS

▼ Description  
 Lists driver substitution rules for auto-created client printers. When you define these rules, you can allow or prevent printers to be created with the specified driver. Additionally, you can allow created printers to use only universal printer drivers.  
 Driver substitution overrides (or maps) printer driver names the client provides, substituting an

注:

PPD ドライバーファイルのみがサポートされています。

## 使用状況

公開デスクトップおよび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 セッションにマッピングできます。プリンター名はデスクトップとアプリケーションとで異なります。

- 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:**

同一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は、どちらのプリンターもセッションで使用できます。公開アプリケーションセッション内でのデスクトッププリンターを使用した印刷、または公開デスクトップでのアプリケーションプリンターを使用した印刷は失敗します。

## 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、印刷デーモン **ctxlpmngt** と **CUPS** フレームワークを確認します。

印刷デーモン **ctxlpmngt** はセッションごとのプロセスで、セッション期間を通して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中であることを確認します。**ctxlpmngt** が実行中でない場合は、コマンドラインから手動で **ctxlpmngt** を起動します。

```
1 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ます。**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。これはマシンごとの単一プロセスであり、以下のコマンドを実行して確認できます：

```
1 systemctl status ctxcups
```

### **CUPS** ログを収集するための追加手順

CUPS ログを収集するには、以下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。構成しないと、CUPS ログが **hdx.log** で記録されません：

```
1 sudo systemctl stop cups
2
3 sudo vi /etc/systemd/system/printer.target.wants/cups.service
4
5 PrivateTmp=false
6
7 sudo systemctl start cups
8
9 sudo systemctl daemon-reload
```

## 注:

この構成は、問題が発生した場合に完全な印刷ログを収集することのみを目的としています。この構成により CUPS のセキュリティが破られるため、通常の状態ではこの構成はお勧めしません。

## 印刷出力が文字化けする

対応していないプリンタードライバーを使用していることが、出力の文字化けの原因になっている可能性があります。ユーザーごとのドライバー構成を使用できるため、`~/.CtulpProfile$CLIENT_NAME` 構成ファイルを編集して構成できます:

```
1 [DEFAULT_PRINTER]
2
3 printername=
4
5 model=
6
7 ppdpath=
8
9 drivertype=
```

## 重要:

**printername** は、現在のクライアント側の通常使うプリンターの名前が指定されているフィールドです。これは読み取り専用の値です。編集しないでください。

**ppdpath**、**model**、**drivertype** の各フィールドは、マップされたプリンターに対していずれか 1 つのフィールドしか有効にならないため、同時には設定できません。

- ユニバーサルプリンタードライバーがクライアントプリンターに対応していない場合、**model=** オプションを使用してネイティブプリンタードライバーのモデルを構成します。プリンターの現在のモデル名は、**lpinfo** コマンドを使用して表示できます:

```
1 lpinfo - m
2
3 ...
4
5 xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
6
7 xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
8 xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
```

次のようにして、プリンターに一致するようにモデルを設定できます。

```
1 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に対応していない場合、ネイティブプリンタードライバーの PPD ファイルのパスを構成します。**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、**ppd** ドライバーが/home/tester/NATIVE\_PRINTER\_DRIVER.ppd にある場合は、次のようになります：

```
1  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は 3 種類（Postscript、pcl5、pcl6）です。プリンターのプロパティに基づいてドライバーの種類を構成できます。

たとえば、クライアントが通常使うプリンターのドライバーの種類が PCL5 である場合は、**drivertype** を次のように指定します：

```
1  drivertype=pcl5
```

#### 出力サイズがゼロ

別の種類のプリンターを試します。また、CutePDF や PDFCreator などの仮想プリンターを使用して、この問題がプリンタードライバーに関連するものかどうかを確認します。

印刷ジョブは、クライアントが通常使用するプリンターのドライバーによって異なります。現在適用されているドライバーの種類を特定することが重要です。クライアントのプリンターが PCL5 ドライバーを使用している一方で、Linux VDA が PostScript ドライバーを選択していると、問題が発生する場合があります。

プリンタードライバーの種類が正しい場合は、次の手順に従って問題を特定します。

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. 次のフィールドを追加して、スプールファイルを Linux VDA に保存します：

```
1  deletespoolfile=no
```

4. いったんログオフしてからログオンし直して、構成の変更を読み込みます。
5. ドキュメントを印刷して問題を再現します。印刷が完了すると、スプールファイルが/var/spool/cups-ctx/\$logon\_user/\$spool\_fileに保存されます。
6. スプールファイルが空であるかどうかを確認します。スプールファイルのサイズが 0 の場合は、これが問題になります。Citrix サポートに印刷ログを提供して、ガイダンスに従ってください。
7. スプールファイルのサイズが 0 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、クライアントが通常使用する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の（ネイティブ）ドライバーが PostScript である場合、スプールファイルは Linux OS で直接開くことができます。内容が正しいかを確認します。

スプールファイルが PCL の場合、またはクライアント OS が Windows の場合は、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が別のプリンタードライバーを使用するように変更します。以下では、PostScript クライアントプリンターを例として使用します：

- a) アクティブセッションにログオンして、クライアントデスクトップでブラウザを開きます。
- b) 印刷管理ポータルを開きます：

```
1 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** と **CTX** 間の接続を保持したまま [続行] をクリックし、プリンタードライバーを変更します。
- e) **[Make]** フィールドと **[Model]** フィールドで、Citrix UPD ドライバーではなく別の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、[汎用 CUPS-PDF プリンター] ドライバーを選択します。変更を保存します。
- f) このプロセスが正常に完了した場合は、ドライバーの PPD ファイルパスを `.CtxlpProfile$CLIENT_NAME` で設定し、マップされたプリンターが新たに選択し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 での印刷について、次の問題が確認されています。

### CTXPS ドライバーが一部の PLC プリンターに対応しない

印刷出力が適切でない場合は、プリンタードライバーを、製造元から提供されたネイティブプリンタードライバーに設定してください。

### サイズの大きな文書の印刷が遅い

ローカルのクライアントプリンターでサイズの大きなドキュメントを印刷すると、そのドキュメントはサーバーとの接続を介して転送されます。遅い接続では、この転送に時間がかかることがあります。

### 別のセッションからプリンター通知と印刷ジョブ通知が表示される

Linux でのセッションの考え方は、Windows オペレーティングシステムとは異なります。したがって、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。次の CUPS 構成ファイルを変更して、これらの通知を無効にできます：  
**/etc/cups/cupsd.conf**。

次のように、構成されている現在のポリシー名がこのファイルに記述されています。

## DefaultPolicy **default**

ポリシー名が *default* である場合は、次の行をデフォルトポリシーの XML ブロックに追加します：

```
1 <Policy default>
2
3     # Job/subscription privacy...
4
5     JobPrivateAccess default
6
7     JobPrivateValues default
8
9     SubscriptionPrivateAccess default
10
11    SubscriptionPrivateValues default
12
13    ... ..
14
15    <Limit Create-Printer-Subscription>
16
17        Require user @OWNER
18
19        Order deny,allow
20
21    </Limit>
22
23    <Limit All>
24
25        Order deny,allow
26
27    </Limit>
28
29 </Policy>
```

## PDF 印刷

July 16, 2024

PDF 印刷に対応したバージョンの Citrix Workspace アプリを使用すると、Linux VDA セッションから変換された PDF を印刷できます。セッション印刷ジョブは、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、選択した PDF ビューアーを使用して PDF を開き、選択したプリンターで印刷することができます。

Linux VDA は以下のバージョンの Citrix Workspace アプリで PDF 印刷をサポートします：

- Citrix Receiver for HTML5 バージョン 2.4~2.6.9、HTML5 向け Citrix Workspace アプリ 1808 以降
- Citrix Receiver for Chrome バージョン 2.4~2.6.9、Chrome 向け Citrix Workspace アプリ 1808 以降
- Windows 向け Citrix Workspace アプリ 1905 以降

## 構成

PDF 印刷機能に対応した Citrix Workspace アプリを使用し、Citrix Studio で以下のポリシーを設定します：

- [クライアントプリンターリダイレクト] を [許可] に設定します（デフォルトは [許可]）
- [PDF ユニバーサルプリンターを自動作成する] を [有効] に設定します（デフォルトは [無効]）
- [クライアントプリンターを自動作成する] を [すべてのクライアントプリンターを自動作成する] に設定します。

これらのポリシーが有効になっている場合、起動されたセッションで [印刷] をクリックすると、ローカルマシンの印刷プレビューに表示され、プリンターを選択できます。デフォルトプリンターの設定について詳しくは、[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

August 21, 2024

### 概要

リモート PC アクセスは、Citrix Virtual Apps and Desktops の拡張機能です。これにより、組織は従業員が物理的なオフィス PC に安全な方法でリモートアクセスできるようにします。ユーザーが社内 PC にアクセスできる場合、作業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。

リモート PC アクセスでは、仮想デスクトップとアプリケーションを配信するのと同じ Citrix Virtual Apps and Desktops コンポーネントが使用されます。リモート PC アクセス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、一貫性のある統一された管理エクスペリエンスが実現されます。ユーザーは、Citrix HDX を使用してリモートオフィス PC セッションを提供することで、最高のユーザーエクスペリエンスを実現できます。

詳しくは、Citrix Virtual Apps and Desktops ドキュメントの「[リモート PC アクセス](#)」を参照してください。

### 注意事項

次の考慮事項は、Linux VDA に固有のものです：

- 物理マシンの場合、Linux VDA は非 3D モードでのみ使用します。NVIDIA のドライバーの制限により、HDX 3D モードが有効になっている場合、PC のローカル画面はブラックアウトしません。この画面の表示は、セキュリティ上のリスクの可能性があります。
- 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使用できません。自動ユーザー割り当てを使用すると、ユーザーは PC にローカルでログオンしたときに、自分のマシンに自動的に割り当てられます。このログオンには、管理者による介入は必要ありません。クライアント側で動作する Citrix Workspace アプリにより、リモート PC アクセスセッションで社内の PC 上のアプリケーションやデータにアクセスできます。
- ユーザーが既にローカルで 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。
- Linux マシンでは、省電力オプションは使用できません。

## 構成

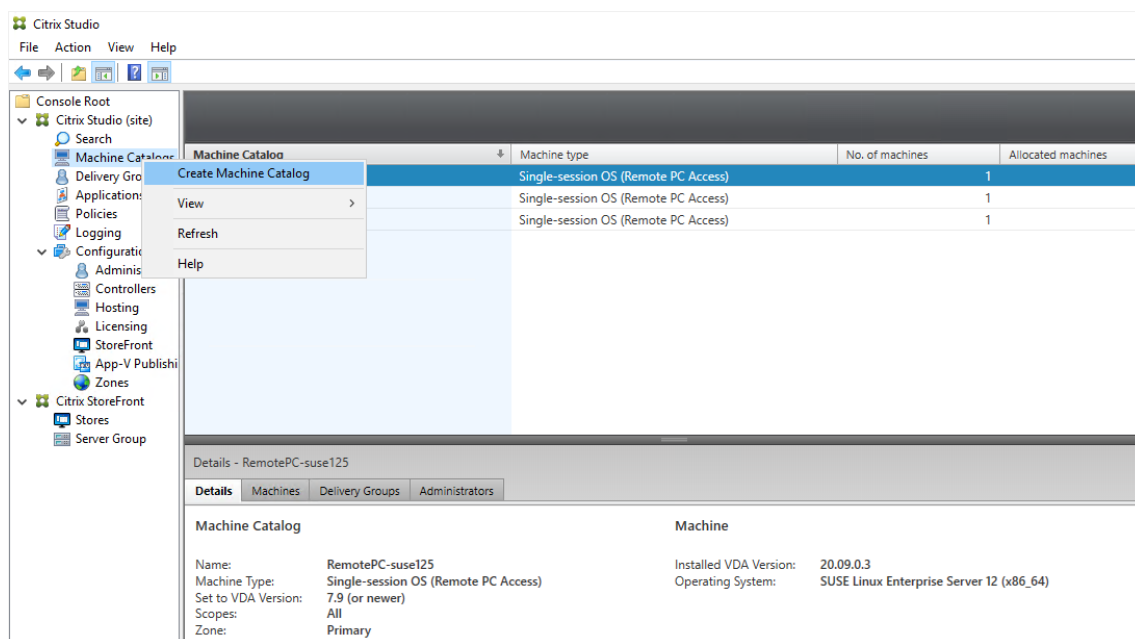
Linux PC セッションを配信するには、対象の PC に Linux VDA をインストールし、リモート **PC** アクセスタイプのマシンカタログを作成し、デリバリーグループを作成して、アクセスを要求するユーザーがマシンカタログ内の PC を利用できるようにします。次のセクションでは、手順について詳しく説明します：

### 手順 1 - 対象の **PC** に **Linux VDA** をインストールする

[簡単インストール](#)を使用して Linux VDA をインストールすることをお勧めします。インストール中、CTX\_XDL\_VDI\_MODE変数の値をYに設定します。

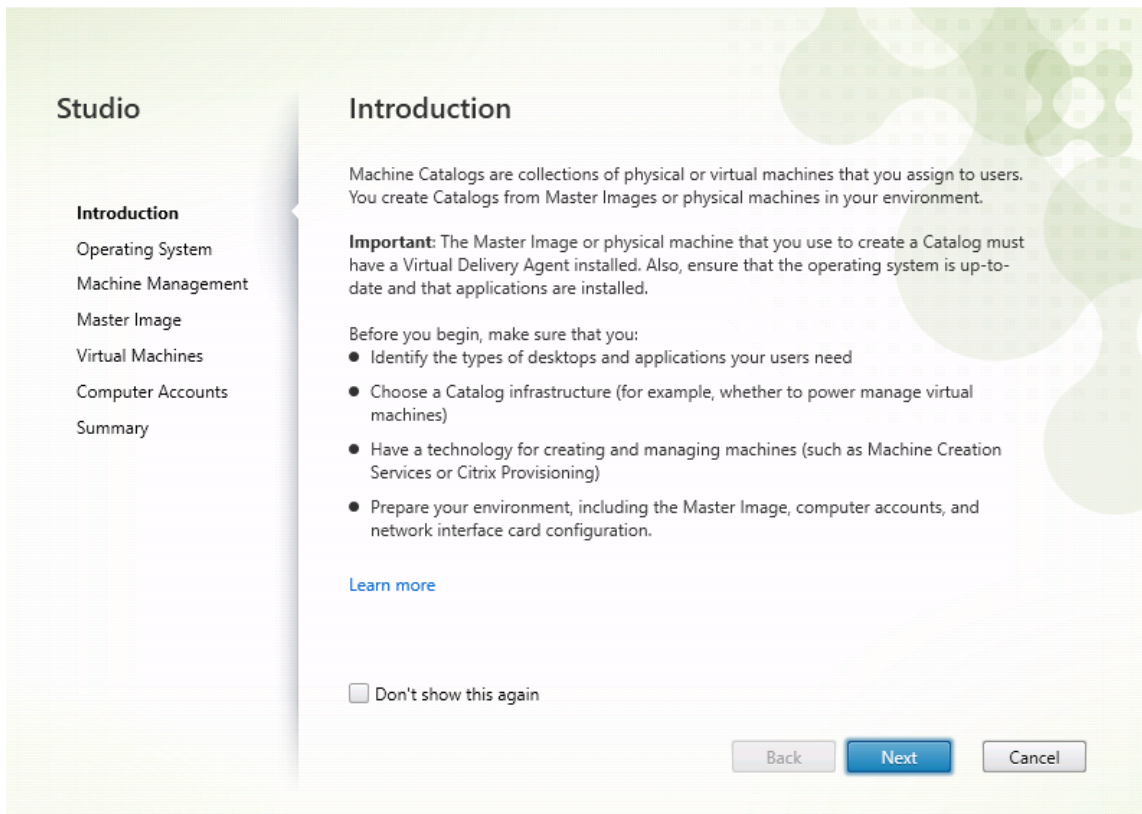
### 手順 2 - リモート **PC** アクセスタイプのマシンカタログを作成する

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。



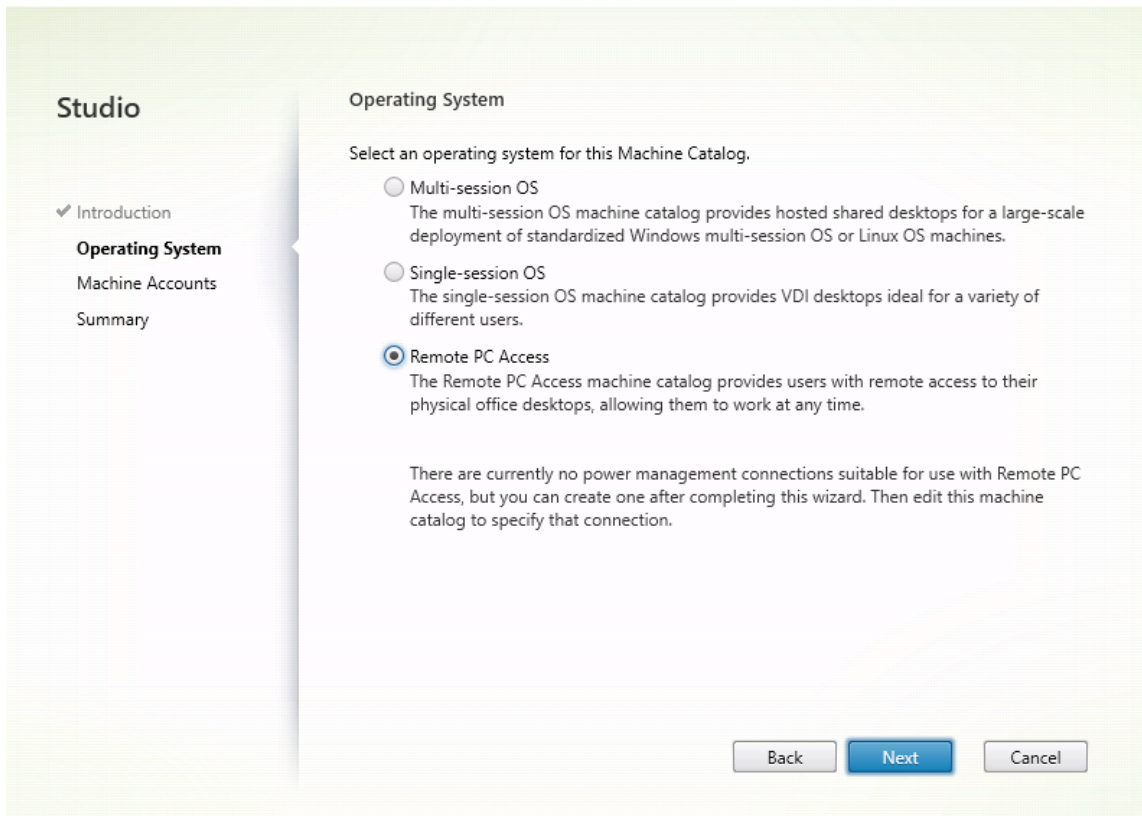
2. [はじめに] ページで [次へ] をクリックします。

Machine Catalog Setup



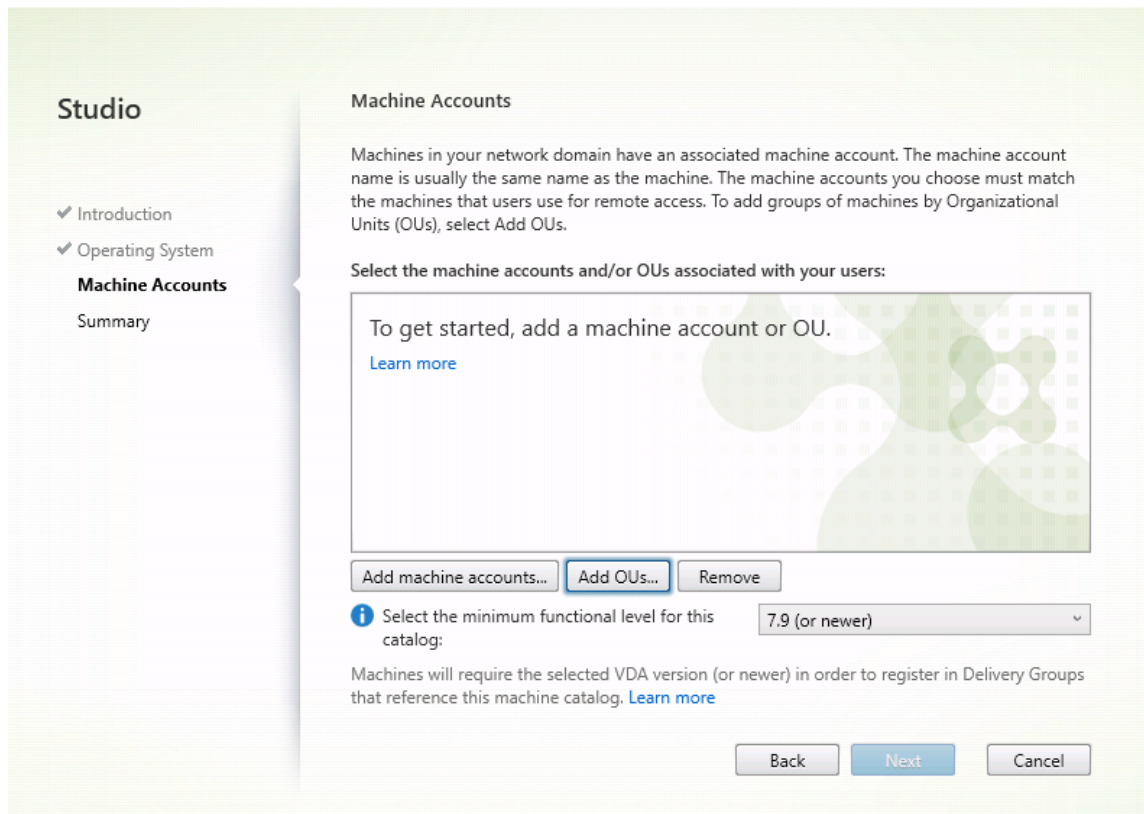
3. [オペレーティングシステム] ページで [リモート PC アクセス] を選択します。

Machine Catalog Setup



4. [OU の追加] をクリックして対象の PC を含む OU を選択するか、[マシンアカウントの追加] をクリックして個別のマシンをマシンカタログに追加します。

Machine Catalog Setup



5. マシンカタログに名前を付けます。

## Machine Catalog Setup

**Studio**

- ✓ Introduction
- ✓ Operating System
- ✓ Machine Accounts
- Summary**

**Summary**

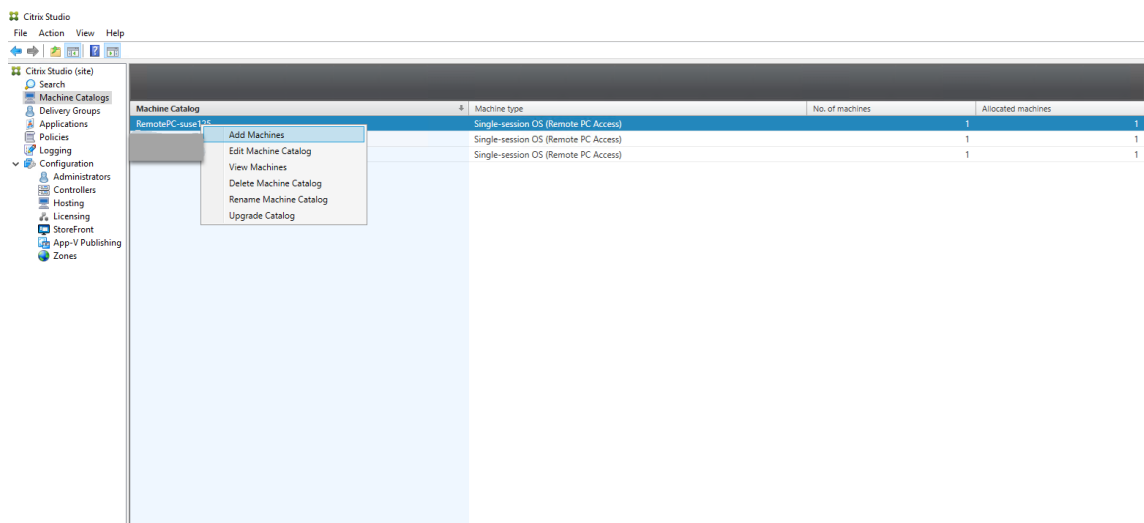
Machine type:	Remote PC Access
Machines added:	1 organizational unit (OU)
VDA version:	7.9 (or newer)
Scopes:	-
Zone:	Primary

Machine Catalog name:

Machine Catalog description for administrators: (Optional)

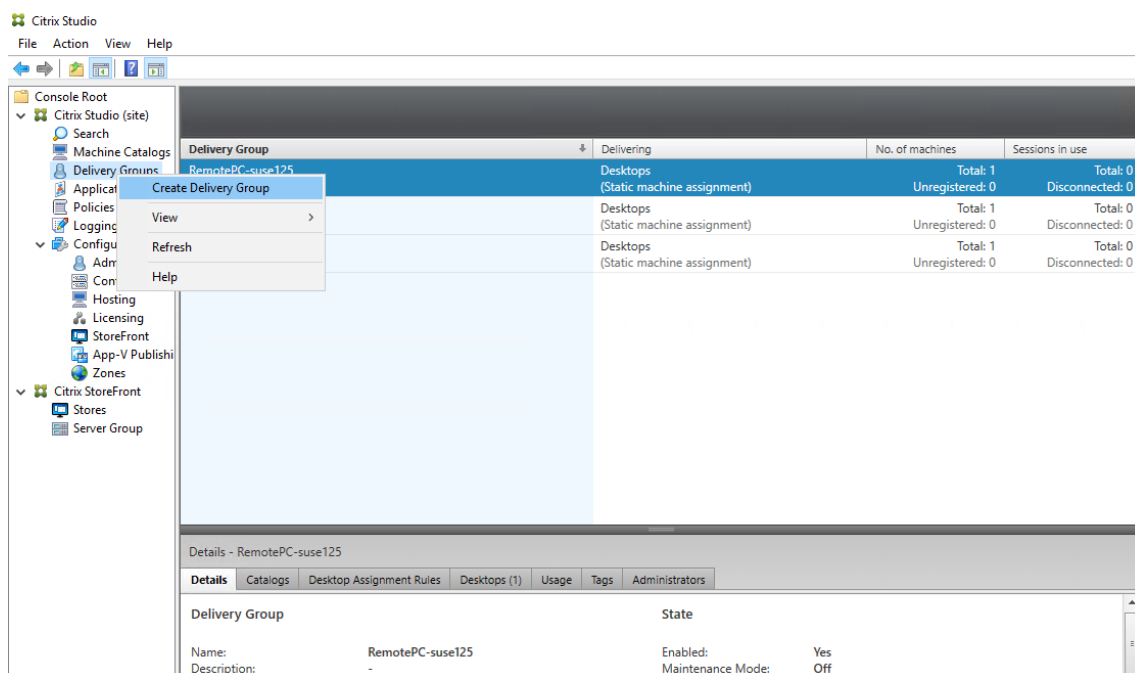
To complete the deployment, assign this Machine Catalog to a Delivery Group by selecting Delivery Groups and then Create or Edit a Delivery Group.

6. (オプション) マシンカタログを右クリックして、必要な操作を実行します。

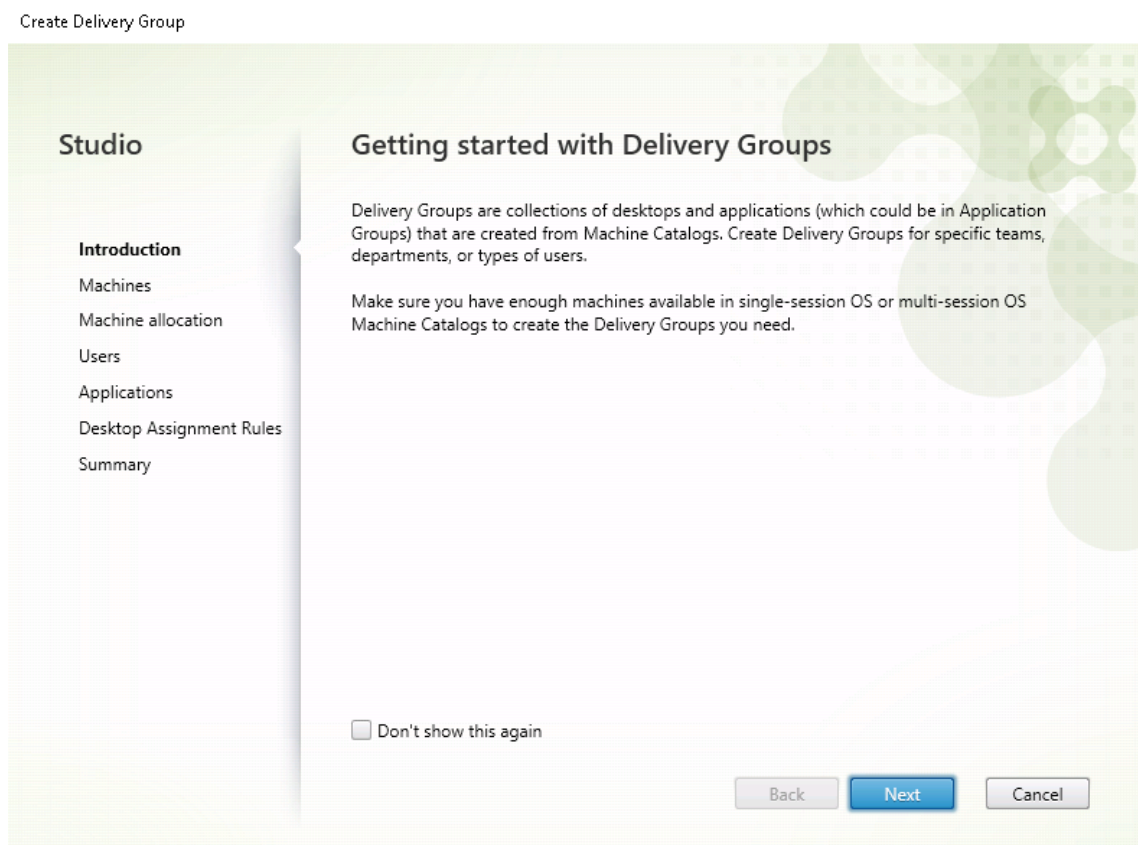


手順 **3** - デリバリーグループを作成してアクセスを要求したユーザーがマシンカタログで **PC** を利用できるようにする

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で [デリバリーグループの作成] を選択します。

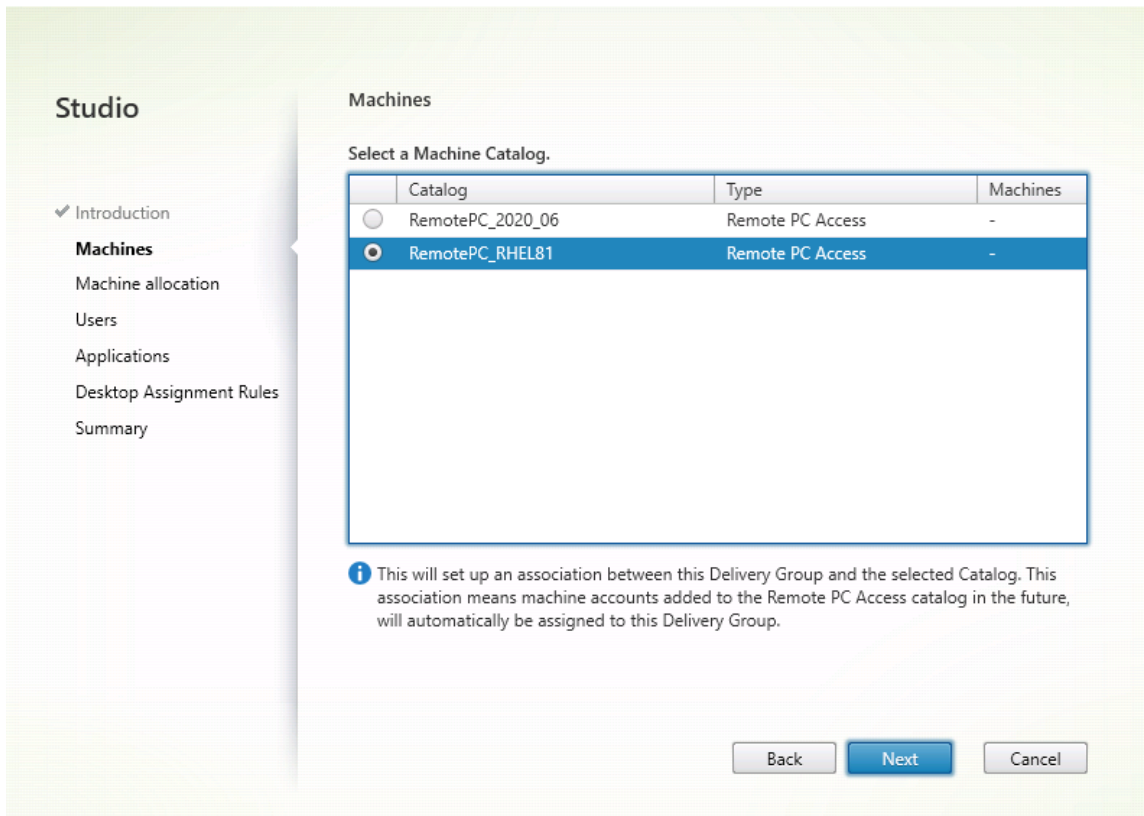


2. [デリバリーグループの作成] ページで [次へ] をクリックします。

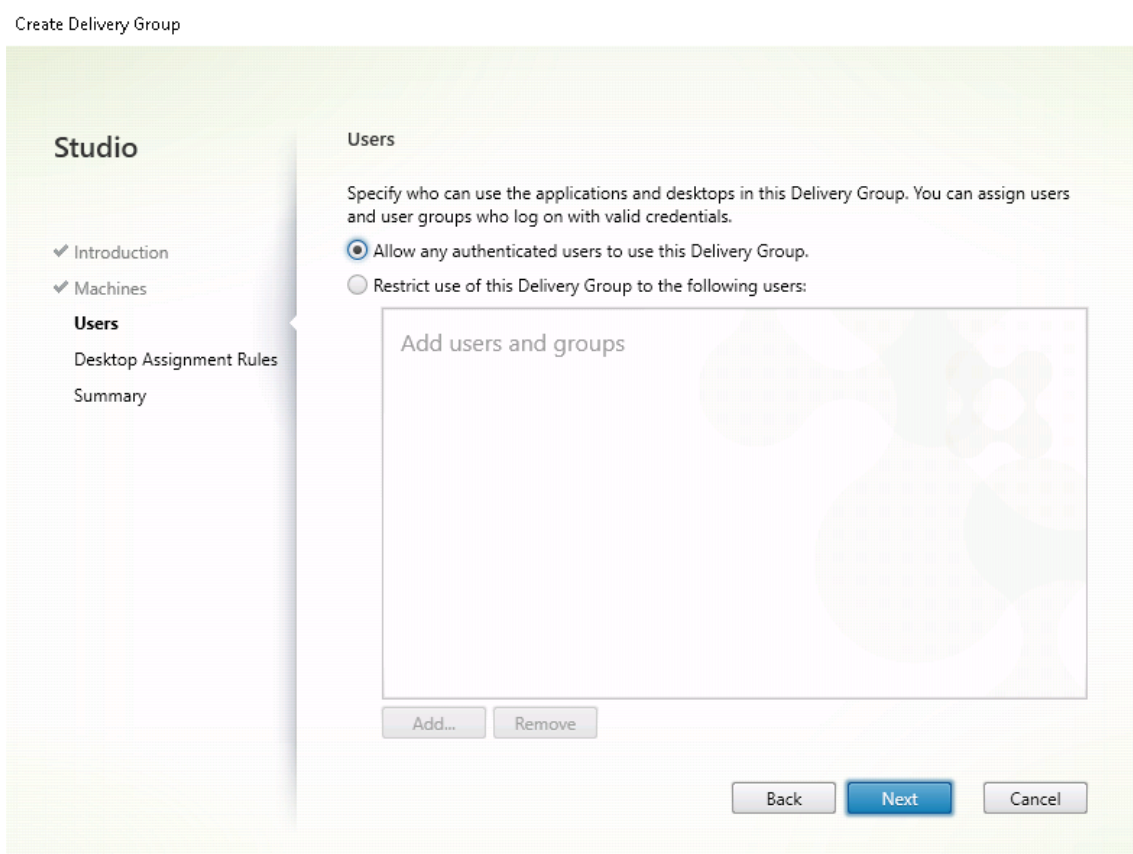


3. 手順2 で作成したマシンカタログを選択して、デリバリーグループに関連付けます。

Create Delivery Group



4. PC にアクセスできるユーザーをマシンカタログに追加します。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

リモート PC アクセスでは Wake on LAN がサポートされ、物理 PC をリモートから起動できます。この機能により、ユーザーが退社時に PC の電源をオフにできるようになるため、消費電力を節約できます。また、電源が突然オフになった PC にもリモートアクセスできるようになります。

Wake on LAN 機能を使用すると、Delivery Controller の指示に従って、PC 上で実行中の VDA から 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。これによって、マジックパケットを配信するために追加のインフラストラクチャコンポーネントまたはサードパーティ製ソリューションに依存する必要がなくなります。

Wake on LAN 機能は、従来の 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 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

以下は、Wake on LAN 機能を使用するためのシステム要件です：

- コントロールプレーン：
  - Citrix DaaS (旧称：Citrix Virtual Apps and Desktops サービス)



- Citrix Virtual Apps and Desktops 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

### Wake on LAN の構成

現在、統合された Wake on LAN の構成は、PowerShell の使用のみがサポートされています。

Wake on LAN を構成するには:

1. リモート PC アクセスマシンカタログをまだ作成していない場合は作成します。
2. Wake on LAN ホスト接続をまだ作成してい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 ホスト接続の一意的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:

```
1 # Load Citrix SnapIns
2 Add-PSSnapIn -Name "*citrix*"
3
4 # Provide the name of the Wake on LAN host connection
5 [string]$connectionName = "Remote PC Access Wake on LAN"
6
7 # Create the hypervisor connection
8 $hypHc = New-Item -Path xdhyp:\Connections `
9             -Name $connectionName `
10            -HypervisorAddress "N/A" `
11            -UserName "woluser" `
12            -Password "wolpwd" `
13            -ConnectionType Custom `
14            -PluginId VdaWOLMachineManagerFactory `
15            -CustomProperties "<CustomProperties></
16                               CustomProperties>" `
17            -Persist
18 $bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
19             $hypHc.HypervisorConnectionUid
20 # Wait for the connection to be ready before trying to use it
21 while (-not $bhc.IsReady)
22 {
```

```

23
24     Start-Sleep -s 5
25     $bhc = Get-BrokerHypervisorConnection -
           HypHypervisorConnectionUid $hypHc.HypervisorConnectionUid
26 }

```

ホスト接続の準備ができたなら、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：

```

1 $bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"
2 $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、その接続をリモート PC アクセスマシンカタログに関連付けます：

```

1 Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog -
  RemotePCHypervisorConnectionUid $hypUId

```

## 5. マシンカタログ内の各仮想マシンの BIOS および NIC で Wake on LAN を有効にします。

注： Wake on LAN を有効にする方法は、マシン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 を表示し、Wake on LAN 機能を有効にします。  
BIOS にアクセスする方法は、マザーボードの製造元と製造元が選択した BIOS ベンダーによって異なります。
  - b) 設定を保存して、マシンを再起動します。
- NIC で Wake on LAN を有効にするには：
  - a) `sudo ethtool <NIC>` コマンドを実行して、NIC がマジックパケットをサポートしているかどうかを確認します。  
<NIC> は NIC のデバイス名です (例: `eth0`)。 `sudo ethtool <NIC>` コマンドによって、NIC の機能に関する出力を生成することができます：
    - 出力に `Supports Wake-on: <letters>` のような行が含まれ、<letters> に文字 `g` が含まれている場合、NIC は Wake on LAN マジックパケット方式をサポートしていません。
    - 出力に `Wake-on: <letters>` のような行が含まれ、<letters> に文字 `g` が含まれ、文字 `d` が含まれていない場合、Wake on LAN マジックパケット方式が有効になっています。ただし、<letters> に `d` 文字が含まれている場合は、Wake on LAN 機能が無効になっていることを示しています。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、毎回起動後に `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、利用しているディストリビューションに基づいて次の手順を実行します：

**Ubuntu:**

インターフェイス構成ファイル/etc/network/interfacesにup ethtool -s <NIC> wol g行を追加します。例:

```
1 # ifupdown has been replaced by netplan(5) on this system.  
   See  
2 # /etc/netplan for current configuration.  
3 # To re-enable ifupdown on this system, you can run:  
4 # sudo apt install ifupdown  
5 auto eth0  
6 iface eth0 inet static  
7     address 10.0.0.1  
8     netmask 255.255.240.0  
9     gateway 10.0.0.1  
10    up ethtool -s eth0 wol g
```

**RHEL/SUSE:**

次のETHTOOL\_OPTSパラメーターをインターフェイス構成ファイル/etc/sysconfig/network-scripts/ifcfg-<NIC>に追加します:

```
1 ETHTOOL_OPTS="-s ${  
2   DEVICE }  
3   wol g"
```

## 設計上の考慮事項

リモート PC アクセスで Wake on LAN を使用する場合は、次の点を考慮してください:

- 複数のマシンカタログでは同じ Wake on LAN ホスト接続を使用できます。
- PC が別の PC をウェイクアップするには、両方の PC が同じサブネット内にあり、同じ Wake on LAN ホスト接続を使用する必要があります。PC が同じマシンカタログにあるか、別のマシンカタログ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。環境に複数のゾーンがある場合は、各ゾーンに Wake on LAN ホスト接続が必要です。同じことが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配信されます。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ウェイクアップできるようにするには、サブネット内で (Wake on LAN 接続ごとに) 少なくとも 1 台の PC がオンになっている必要があります。

## 運用上の考慮事項

以下は、Wake on LAN 機能を使用する場合の考慮事項です:

- 統合された Wake on LAN 機能を使用して PC をウェイクアップするには、VDA を少なくとも 1 回登録する必要があります。

- Wake on LAN は、PC のウェイクアップにのみ使用できます。再起動やシャットダウンなど、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 内でのプロパティ編集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 PC へのセッションを開始しようとしたときに、VDA が登録解除されている場合
  - 管理者が Studio または PowerShell から電源オンのコマンドを手動で送信する場合
- Delivery Controller は PC の電源の状態を認識しないため、Studio では電源の状態のところに [サポートされていません] と表示されます。Delivery Controller 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

リモート PC アクセスのその他のリソースは次のとおりです：

- ソリューション設計ガイダンス：「[リモート PC アクセス設計の決定](#)」。
- リモート PC アクセスアーキテクチャの例：「[Citrix のリモート PC アクセス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

July 16, 2024

このセクションでは、以下のトピックについて説明します：

- [アダプティブトランスポート](#)
- [一時的なホームディレクトリを使用したログオン](#)
- [アプリケーションの公開](#)
- [セッション画面の保持](#)
- [Rendezvous V1](#)
- [Rendezvous V2](#)
- [TLS によるユーザーセッションの保護](#)
- [DTLS による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

July 16, 2024

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 が使用できない場合、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、ユーザーデータグラムプロトコル (UDP) 上に構築された Citrix 独自のトランスポートプロトコルです。サーバーのスケーラビリティを維持しながら、要求の厳しい長距離接続で優れたユーザーエクスペリエンスを提供します。EDT は、信頼性の低いネットワーク上のすべての ICA 仮想チャネルのデータスループットを向上させ、より優れた、より一貫性のあるユーザーエクスペリエンスを提供します。

詳しくは、Citrix Virtual Apps and Desktops ドキュメントの「[アダプティブトランスポート](#)」を参照してください。

### アダプティブトランスポートの有効化または無効化

アダプティブトランスポートはデフォルトで有効になっています。[HDX アダプティブトランスポート] ポリシー設定を使用して、以下のオプションを構成できます。

**Edit Setting**

**HDX adaptive transport**

Value: Preferred

Use default value: Preferred

▼ Applies to the following VDA versions

Virtual Delivery Agent: 7.13 Server OS, 7.13 Desktop OS, 7.14 Server OS, 7.14 Desktop OS, 7.15 Server OS, 7.15 Desktop OS, 7.16 Server OS, 7.16 Desktop OS, 7.17 Server OS, 7.17 Desktop OS, 7.18 Server OS, 7.18 Desktop OS, 1808 Multi-session OS, 1808 Single-session OS, 1811 Multi-session OS, 1811 Single-session OS, 1903 Multi-session OS, 1903 Single-session OS, 1906 Multi-session OS, 1906 Single-session OS, 1909 Multi-session OS, 1909 Single-session OS, 1912 Multi-session OS, 1912 Single-session OS, 2003 Multi-session OS, 2003 Single-session OS, 2006 Multi-session OS, 2006 Single-session OS, 2009 Multi-session OS, 2009 Single-session OS, 2012 Multi-session OS, 2012 Single-session OS, 2103 Multi-session OS, 2103 Single-session OS, 2106 Multi-session OS, 2106 Single-session OS, 2109 Multi-session OS, 2109 Single-session OS, 2112 Multi-session OS, 2112 Single-session OS, 2203 Multi-session OS, 2203 Single-session OS, 2206 Multi-session OS, 2206 Single-session OS, 2209 Multi-session OS, 2209 Single-session OS, 2212 Multi-session OS, 2212 Single-session OS, 2303 Multi-session OS, 2303 Single-session OS, 2305 Multi-session OS, 2305 Single-session OS, 2308 Multi-session OS, 2308 Single-session OS, 2311 Multi-session OS, 2311 Single-session OS, 2402 Multi-session OS, 2402 Single-session OS

▼ Description

Adaptive transport is a network-aware data transport engine that provides efficient, reliable, and consistent congestion and flow control.

By default, adaptive transport is set to Preferred, data transport takes place over a proprietary transport protocol, Enlightened Data Transport (EDT), that is built on top of UDP, with automatic fallback to TCP. Additional configuration is not required to optimize for LAN, WAN, or Internet conditions. Citrix's transport protocol responds to changing conditions.

When set to Off, adaptive transport is disabled and TCP is used. Recommended when using SD-WAN WAN optimization, which provides cross-session tokenized compression, since WAN optimization has its own congestion and flow control.

Setting Diagnostic mode forces EDT on and disables fallback to TCP. Recommended for testing purposes only.

None of these settings affects other services that depend on UDP transport, such as UDP Audio and Framehawk.

OK Cancel

- 優先: アダプティブトランスポートが有効になっており、TCP へのフォールバックが有効な状態で、EDT (Enlightened Data Transport) を優先トランスポートプロトコルとして使用します。
- 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っています。この設定は、テストとトラブルシューティングにのみ使用することをお勧めします。
- オフ: アダプティブトランスポートは無効になっており、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中かどうかを確認する

EDT が現在のセッションのトランスポートプロトコルとして使用中かどうかを確認するには、次のコマンドを実行します。

```
1 /opt/Citrix/VDA/bin/ctxquery -f iP
```

EDT が使用中の場合、表示されるトランスポートプロトコルには次の例のように UDP が含まれます。

```
jll@jll-ubuntu:~/Desktop$ ctxquery -f iP
SESSION:ID      TRANSPORT PROTOCOLS  RENDEZVOUS
jll-u20:0       -                    -
jll-u20:1       -                    -
jll-u20:2       -                    -
jll-u20:12      UDP-CGP-ICA         NONE
jll@jll-ubuntu:~/Desktop$
```

## EDT MTU 検出

MTU Discovery により、セッション確立時に EDT が最大伝送単位 (MTU) を自動的に決定できるようにします。これにより、パフォーマンスの低下やセッションの確立失敗となる可能性のある、EDT パケットのフラグメンテーションが防止されます。

システム要件:

- Linux VDA 最小バージョン 2012
- Citrix Workspace アプリ:
  - Windows: 1911 以降
- Citrix ADC:
  - 13.0.52.24 以降
  - 12.1.56.22 以降
- セッション画面の保持を有効にする必要があります

クライアントプラットフォームまたはこの機能をサポートしていないバージョンを使用している場合、環境に適したカスタムの EDT MTU の構成については、[CTX231821](#)を参照してください。

## VDA での EDT MTU Discovery の制御

EDT MTU 検出はデフォルトで VDA で有効になっています。これを有効または無効にするには、次のように `MtuDiscovery` レジストリキーを設定します:

- EDT MTU 検出を有効にするには、次のコマンドを使用して `MtuDiscovery` レジストリキーを設定し、VDA を再起動して、VDA が登録されるのを待ちます：

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

- EDT MTU 検出を無効にする場合は、`MtuDiscovery` レジストリ値を削除します。

**警告：**

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、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場合もあります。レジストリエディターの誤用による障害に対して、Citrix では一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

#### クライアントで **EDT MTU** 検出を制御する

ICA ファイルに `MtuDiscovery` パラメーターを追加することで、クライアント上で EDT MTU 検出を選択的に制御できます。この機能を無効にする場合は、`Application` セクションで次のように設定します：

`MtuDiscovery=0ff`

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を機能させるには、VDA で EDT MTU 検出を有効にします。VDA で EDT MTU 検出が有効になっていない場合、ICA ファイルパラメーターは機能しません。

#### 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ています。この実装により、EDT はより高いスループットを実現し、待ち時間を短縮して、ユーザーエクスペリエンスを向上させることができます。

この機能はデフォルトで有効になっています。これを無効または有効にするには、次のコマンドを実行してから `ctxhdx` サービスを再起動します：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
  Control\Terminal Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "
  REG_DWORD" -v "edtBBR" -d "0x00000000" --force
```

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
  Control\Terminal Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "
  REG_DWORD" -v "edtBBR" -d "0x00000001" --force
```

## HDX アダプティブスループット

August 20, 2024

HDX アダプティブスループットは、出力バッファを調整することで、ICA セッションのピークスループットをインテリジェントに微調整します。出力バッファの数は、最初は大きい値に設定されます。値を大きくすることで、特に高遅延のネットワークで、データをより迅速かつ効率的にクライアントに送信できます。この機能により、ユーザーエクスペリエンスが向上します。高い双方向性、高速なファイル転送、スムーズなビデオ再生、および高いフレームレートと解像度を提供します。セッションの双方向性を常に測定して、ICA セッション内のデータストリームが双方向性に悪影響を及ぼしているかどうかを判別します。悪影響を及ぼしている場合、スループットを低下させて、大規模データストリームがセッションに与える影響を減らし、双方向性を回復できるようにします。

### 重要:

HDX アダプティブスループットでは、メカニズムをクライアントから VDA に移行することにより、出力バッファの設定方法を変更しています。手動による構成は必要ありません。

この機能には、VDA バージョン 2311 以降が必要です。このオプションは、デフォルトで有効になっています。無効になっている場合は、VDA で次のコマンドを実行して有効にすることができます:

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

この機能が適用されるのは、機能が有効になった後に起動したセッションのみです。

## セッションログオン画面のカスタム背景とバナーメッセージ

July 16, 2024

カスタムの背景またはバナーメッセージをセッションログオン画面に追加する

### ヒント:

SUSE 15.5 でこの機能を使用するには、<http://download.opensuse.org/distribution/leap/15.3/repodata/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、カスタムの背景またはバナーメッセージをセッションログオン画面に追加できます。背景とバナーメッセージの両方をセッションログオン画面に追加するために、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。セッションを開いた後、最初にバナーメッセージページが表示され、次に認証ダイアログが表示されます。



カスタムバナーメッセージのタイトルを設定するには、次を実行します：

```
1 /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。

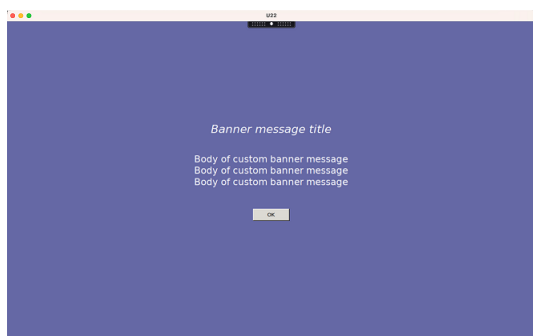
カスタムバナーメッセージの本文テキストを設定するには、次を実行します：

```
1 /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：



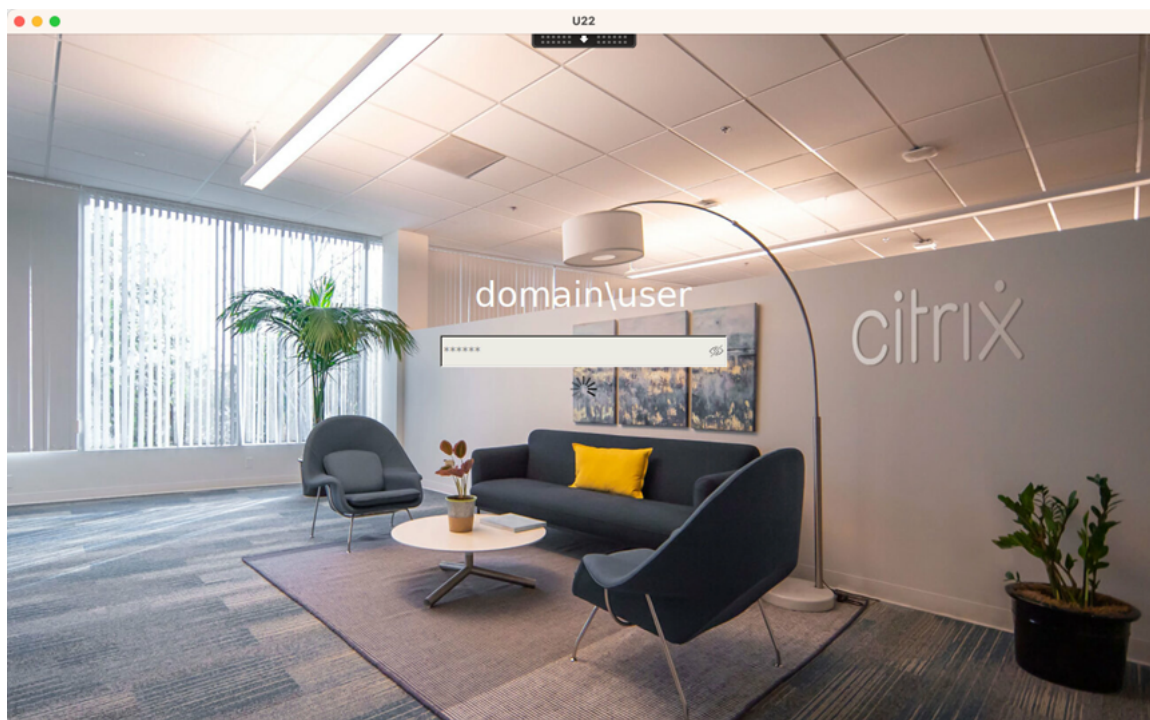
カスタムバナーメッセージが正常に表示されるためには、エスケープ文字の使用が必要な場合があります。たとえば、HTML で **&** を表す場合に **&** を使用し、文字 **!** の前にバックラッシュ (`**\**`) を追加して、Bash コマンドでエスケープします。

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：

```
1 /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が背景画像のパスにアクセスする必要があります。

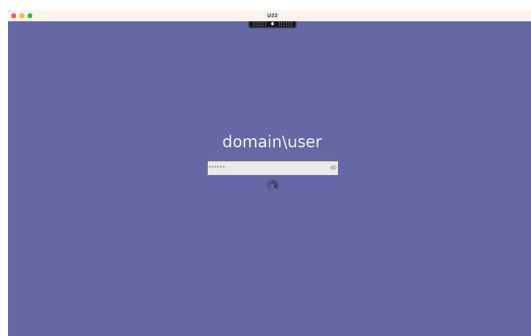
例：



### 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：

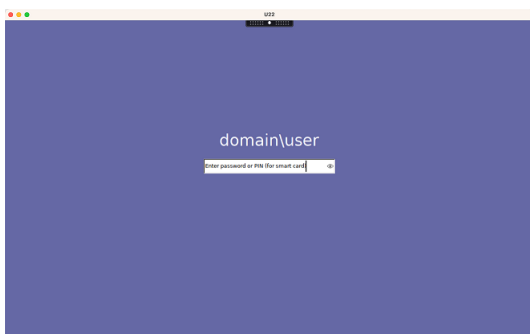
- シングルサインオン（SSO）シナリオでのセッションログオン：



でのセッション ログオン

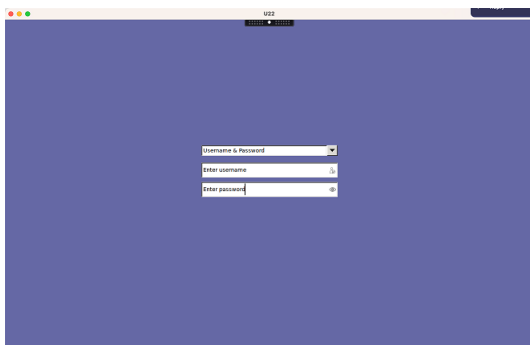
ログオンプロセスが表示されます。

- 一般的な SSO 以外のシナリオでのセッションログオン：

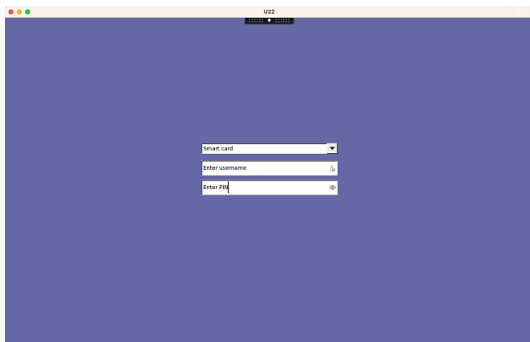


- SSO 以外のシナリオでは、パスワードまたは PIN コードが必要です。
  - ユーザーはパスワードと PIN コードの表示を切り替えることができるため、間違っただけの入力を見つけやすくなります。
- ユーザーが Citrix Workspace アプリへのログオンに使用されたものとは異なる資格情報を使用する場合の、SSO 以外のシナリオでのセッションログオン:

セッションログオンに使用されたユーザー名とパスワード:



セッションログオンに使用されるスマートカード:



SSO 以外の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については、「[SSO 以外の認証](#)」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

August 20, 2024

**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。この機能をセッションユーザーが使用できるようにするには、事前に VDA にデスクトップ環境をインストールする必要があります。

次の表は、セッションユーザーによるカスタムデスクトップ環境をサポートする Linux ディストリビューションとデスクトップ環境のマトリックスを示しています。

Linux ディストリビューション	サポートされるデスクトップ
Debian 12.5	MATE、GNOME、GNOME-Classic、KDE、 <b>Xfce</b>
Debian 11.9	MATE、GNOME、GNOME-Classic、KDE
RHEL 8.10/8.9/8.8	MATE、GNOME、GNOME-Classic、 <b>Xfce</b>
Rocky Linux 8.10/8.9/8.8	MATE、GNOME、GNOME-Classic、KDE、 <b>Xfce</b>
SUSE 15.5	MATE、GNOME、GNOME クラシック
Ubuntu 22.04/20.04	MATE、GNOME、GNOME-Classic、KDE、 <b>Xfce</b>

## デスクトップ切り替えコマンド

注:

端末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。

端末からターゲット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:

対象のデスクトップ環境が次の場合:	次のコマンドを実行します:
GNOME	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh GNOME</code>
GNOME クラシック	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh GNOME-CLASSIC</code>
MATE	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh MATE</code>
KDE	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh KDE</code>

**KDE** のヒント

- Magnus は、KDE の起動時に読み込ま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 の起動時に発生する QT 警告を無効にするには、次のエントリを追加して `/usr/share/qt5/qtlogging.ini` を root ユーザーとして編集します。

```
1 qt.qpa.xcb.xcberror.error=false
2 qt.qpa.xcb.warning=false
3 qt.qpa.xcb.error=false
```

- KDE で画面のロック解除に失敗する場合があります。回避策として、デスクトップの自動ロック機能を無効にすることをお勧めします。

## 一時的なホームディレクトリを使用したログオン

July 16, 2024

Linux VDA のマウントポイントに障害が発生した場合に備えて、一時的なホームディレクトリを指定できます。一時的なホームディレクトリを指定すると、セッションログオン中、マウントポイントに障害が発生したとき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的なホームディレクトリに保存されます。

次の表に、ホームディレクトリの設定に役立つレジストリキーを示します。

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、「はい」を意味します。値が 0 に設定されている場合、ホームディレクトリなしのセッションログオンは無効になります。	<code>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "LogNoHome"-d "0x00000001"--force</code>

レジストリキー	説明	コマンド
HomeMountPoint	Linux VDA にローカルマウントポイントを設定します。たとえば、 <code>/mnt/home</code> 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 <code>/mnt/home/domain/&lt;user_name&gt;</code> です。マウントポイントが環境内のユーザーの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、 <code>CheckUserHomeMountPoint</code> が0に設定されている場合にのみ有効です。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_SZ"-v "HomeMountPoint"-d "&lt;A directory where the NFS share is to be mounted&gt;"--force</pre>
CheckUserHomeMountPoint	ユーザー固有のホームディレクトリをLinux VDAのマウントポイントとして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 <b>1</b> に設定します。デフォルト値は <b>0</b> です。	<pre>ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "CheckUserHomeMountPoint"-d "0x00000001"--force</pre>
TempHomeDirectoryPath	マウントポイントに障害が発生した場合に備えて、Linux VDA に一時的なホームディレクトリを設定します。デフォルトの値は <code>/tmp</code> です。一時的なホームディレクトリの設定は、 <code>HomeMountPoint</code> と <code>CheckUserHomeMountPoint</code> で決定されるマウントポイントが使用できない場合にのみ有効になります。ユーザーの一時的なホームディレクトリは <code>/tmp/CTXSmf_user_id</code> です。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_SZ"-v "TempHomeDirectoryPath"-d "&lt;/tmp by default&gt;"--force</pre>

レジストリキー	説明	コマンド
CheckMountPointRetryTime	マウントが成功したかどうかを何秒に1回の頻度でチェックするかの値を設定します。デフォルト値は5です。	<pre>ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "CheckMountPointRetryTime"-d "0x000000010"--force</pre>
RemoveHomeOnLogoff	ユーザーのログオフ時に一時的なホームディレクトリを削除するかどうかを制御します。1は「はい」、0は「いいえ」を意味します。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "RemoveHomeOnLogoff"-d "0x00000000"--force</pre>

## アプリケーションの公開

July 16, 2024

Linux VDA バージョン 7.13 では、Citrix でシームレスアプリケーション機能がサポート対象のすべての Linux プラットフォームに追加されました。この機能を使用するのに特別なインストール手順は不要です。

ヒント:

Linux VDA Version 1.4 では、非シームレスな公開アプリケーションとセッションの共有のサポートが Citrix で追加されました。

### Citrix Studio を使ってアプリケーションを公開する

デリバリーグループを作成したり、既存のデリバリーグループにアプリケーションを追加したりすると、Linux VDA にインストールしたアプリケーションを公開することができます。このプロセスは、Windows VDA にインストールしたアプリケーションを公開する場合と同様です。詳しくは、[Citrix Virtual Apps and Desktops ドキュメント](#) (使用中の Citrix Virtual Apps and Desktops のバージョン) を参照してください。

注:

- デリバリーグループの構成では、デリバリーの種類を [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] に設定します。
- アプリおよびデスクトップの配信用に個別の VDA とデリバリーグループを作成することをお勧めします。
- シームレスアプリケーションを使用するには、StoreFront でシームレスモードを無効にしないでください。シームレスモードは、デフォルトで有効になっています。既に「TWIMode=Off」を設定して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。削除しない場合は、公開デスクトップを起動できないことがあります。

### 制限事項

Linux VDA では、1 人のユーザーが同じアプリケーションの複数の同時インスタンスを起動することはできません。

アプリセッションでは、アプリに固有のショートカットのみが正常に機能します。

### 既知の問題

アプリケーション公開時の既知の問題は次のとおりです:

- 非矩形のウィンドウはサポートされません。ウィンドウの隅にサーバー側の背景が表示されることがあります。
- ウィンドウの内容を公開アプリケーションからプレビューすることはサポートされていません。
- 複数の LibreOffice アプリケーションによってプロセスが共有されるため、Citrix Studio には最初に起動したもののみが表示されます。
- 「Dolphin」などの公開された Qt5 ベースのアプリケーションについてはアイコンが表示されないことがあります。この問題を解決するには、<https://wiki.archlinux.org/title/Qt>の記事を参照してください。
- Linux アプリケーションには多くの場合、使用中のアプリケーションに関する情報を含む [About] ウィンドウがあり、これらのウィンドウに詳細情報への Web リンクが表示されます。[About] ウィンドウの Web リンクをクリックすると、**calc**、**gedit**、**calendar**、**LibreOffice Suite** などの公開アプリケーション内からブラウザを起動できます。ブラウザが意図せず起動されると、アプリケーションの分離がバイパスされることになり、セキュリティ上の危険が発生する可能性があります。この問題に対応するには、次の手順を実行してデフォルトのブラウザを変更します:

1. カスタムの場所に `none.sh` ファイルを作成します。次に例を示します:

```
1 sudo mkdir /home/none
2
3 sudo touch /home/none/none.sh
4
5 sudo chmod +x /home/none/none.sh
```



2. 次の行を `none.sh` ファイルに追加します:

```
1 #!/bin/bash
2
3 echo "NONE"
```

3. `/etc/xdg/mimeapps.list` ファイルを `sudo` 権限で作成して、次の行を `mimeapps.list` ファイルに追加します:

```
1 [Default Applications]
2
3 text/html=none.desktop
4
5 x-scheme-handler/http=none.desktop
6
7 x-scheme-handler/https=none.desktop
8
9 x-scheme-handler/about=none.desktop
10
11 x-scheme-handler/unknown=none.desktop
```

4. `/usr/share/applications/none.desktop` ファイルを `sudo` 権限で作成し、次の行を `none.desktop` ファイルに追加します:

```
1 [Desktop Entry]
2
3 Encoding=UTF-8
4
5 Version=1.0
6
7 Type=Application
8
9 Terminal=false
10
11 Exec=/home/none/none.sh
12
13 Name=None
14
15 Icon=/home/none/none.sh
```

`none.sh` ファイルをカスタムの場所に配置できるため、`none.desktop` が `none.sh` ファイルを適切に参照できることを確認してください。

## Rendezvous V1

July 16, 2024

Citrix Gateway サービスを使用する場合、Rendezvous プロトコルにより、トラフィックが Citrix Cloud Connector をバイパスして、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つのタイプがあります：1) VDA 登録とセッション仲介のための制御用トラフィック、2) HDX セッショントラフィック。

Rendezvous V1 では、HDX セッショントラフィックが Cloud Connector をバイパスできますが、それでも、Cloud Connector が VDA 登録とセッション仲介のためのすべての制御用トラフィックにプロキシを使用する必要があります。

## 要件

- Citrix Workspace と Citrix Gateway サービスを使用した環境へのアクセス。
- コントロールプレーン：Citrix DaaS (旧称 Citrix Virtual Apps and Desktops サービス)。
- Linux VDA バージョン 2112 以降。
  - 非透過 HTTP プロキシには、バージョン 2112 以降が必要です。
  - 透過プロキシおよび SOCKS5 プロキシには、バージョン 2204 以降が必要です。
- Citrix ポリシーで Rendezvous プロトコルを有効にします。詳しくは、「[Rendezvous プロトコルポリシー設定](#)」を参照してください。
- VDA は、すべてのサブドメインを含む [https://\\*.nssvc.net](https://*.nssvc.net) にアクセスできる必要があります。この方法ですべてのサブドメインを許可リストに登録できない場合、代わりに [https://\\*.c.nssvc.net](https://*.c.nssvc.net) および [https://\\*.g.nssvc.net](https://*.g.nssvc.net) を使用します。詳しくは、Citrix Cloud のドキュメント (Virtual Apps and Desktops サービス内) の「[インターネット接続の要件](#)」セクションおよび Knowledge Center の記事 [CTX270584](#) を参照してください。
- Cloud Connector は、セッションを仲介する場合、VDA の FQDN を取得する必要があります。このタスクを完了するには、次の2つの方法があります：
  - サイトの **DNS** 解決を有効にします。[完全な構成] > [設定] に移動し、[DNS 解決を有効にする] 設定をオンにします。または、Citrix Virtual Apps and Desktops Remote PowerShell SDK を使用して、コマンド `Set-BrokerSite -DnsResolutionEnabled $true` を実行します。Citrix Virtual Apps and Desktops Remote PowerShell SDK について詳しくは、「[SDK および API](#)」を参照してください。
  - **VDA** の **PTR** レコードを含む **DNS** 逆引き参照ゾーン。このオプションを選択した場合は、常に PTR レコードの登録を試行するように VDA を構成することをお勧めします。これを行うには、グループポリシーエディターまたはグループポリシーオブジェクトを使用して、[コンピューターの構成] > [管理用テンプレート] > [ネットワーク] > [DNS クライアント] に移動し、[PTR レコードを登録する] を [有効] および [登録] に設定します。接続の DNS サフィックスがドメインの DNS サフィックスと一致しない場合は、マシンが PTR レコードを正常に登録できるように、[接続固有の **DNS** サフィックス] 設定も構成する必要があります。

注:

DNS 解決オプションを使用する場合、Cloud Connector で VDA マシンの完全修飾ドメイン名 (FQDN) を解決できなければなりません。内部ユーザーが VDA マシンに直接接続する場合、クライアントデバイスも VDA マシンの FQDN を解決できる必要があります。

DNS 逆引き参照ゾーンを使用する場合、PTR レコードの FQDN は VDA マシンの FQDN と一致する必要があります。PTR レコードに別の FQDN が含まれている場合、Rendezvous 接続は失敗します。たとえば、マシンの FQDN が `vda01.domain.net` の場合、PTR レコードには `vda01.domain.net` が含まれている必要があります。 `vda01.sub.domain.net` などの別の FQDN だと機能しません。

## プロキシ構成

VDA では、HTTP プロキシおよび SOCKS5 プロキシを介した Rendezvous 接続の確立がサポートされています。

## 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 HTTP プロキシおよび SOCKS5 プロキシがサポートされています。
- パケットの暗号化解除と検査はサポートされていません。VDA と Gateway サービスの間の ICA トラフィックが傍受、暗号化解除、または検査されないように、例外を構成します。例外を構成しないと、接続が切断されます。
- HTTP プロキシでは、Negotiate および Kerberos 認証プロトコルを使用して、マシンベースの認証がサポートされています。プロキシサーバーに接続するとき、**Negotiate** 認証スキームによって Kerberos プロトコルが自動的に選択されます。Kerberos は、Linux VDA でサポートされてい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、セッションの確立時に `HTTP/<proxyURL>` 形式の SPN を生成します。この場合、プロキシ URL は **Rendezvous** プロキシの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 プロキシによる認証は、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、要件で指定されている Gateway サービスアドレス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。
- EDT を介したデータ転送をサポートしているのは、SOCKS5 プロキシのみです。HTTP プロキシの場合、ICA のトランスポートプロトコルとして TCP を使用します。

## 透過プロキシ

透過 HTTP プロキシは Rendezvous でサポートされています。ネットワークで透過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。

## 非透過プロキシ

ネットワークで非透過プロキシを使用している場合は、[Rendezvous プロキシの構成](#)の設定を行います。この設定が有効になっている場合、VDA が使用するプロキシを認識できるように、HTTP または SOCKS5 プロキシアドレスを指定します。例：

- プロキシアドレス: `http://<URL or IP>:<port>` または `socks5://<URL or IP>:<port>`

## Rendezvous の検証

すべての要件を満たしている場合は、次の手順に従って、Rendezvous が使用されているかを検証します：

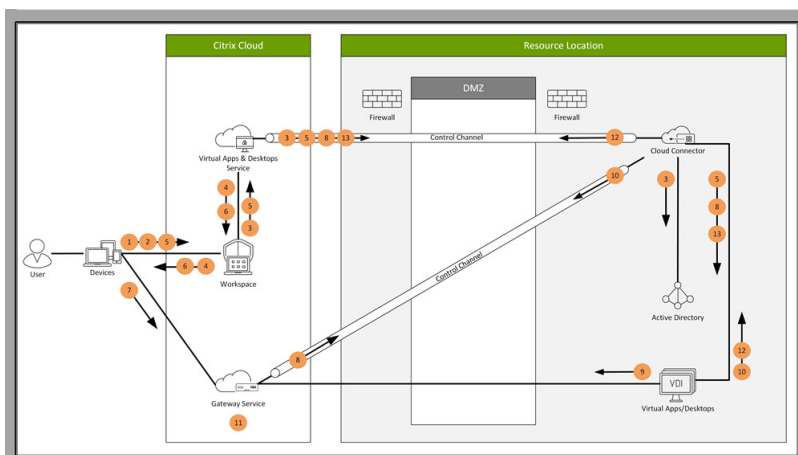
1. VDA でターミナルを起動します。
2. `/opt/Citrix/VDA/bin/ctxquery -f iP`を実行します。
3. [トランスポートプロトコル] は接続の種類を示します：
  - TCP Rendezvous: TCP - TLS - CGP - ICA
  - EDT Rendezvous: UDP - DTLS - CGP - ICA
  - Cloud Connector を介したプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

### ヒント：

Rendezvous が有効で VDA が Citrix Gateway サービスに直接到達できない場合、VDA はフォールバックし Cloud Connector を介して HDX セッションにプロキシ接続します。

## Rendezvous のしくみ

この図は、Rendezvous 接続フローの概要です。



フローを理解するためには、この手順を実行してください。

1. Citrix Workspace に移動します。
2. Citrix Workspace で資格情報を入力します。
3. オンプレミス Active Directory を使用する場合、Citrix DaaS は Cloud Connector チャンネルを使用して Active Directory で資格情報を認証します。
4. Citrix Workspace に、Citrix DaaS から列挙されたリソースが表示されます。
5. Citrix Workspace でリソースを選択します。Citrix DaaS は、VDA にメッセージを送信して、受信セッションの準備をします。
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 に接続するためにこのチケットを提供し、Citrix Cloud は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、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 経由で接続にプロキシを設定します。
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 でライセンスを検証します。
13. Citrix DaaS は、Cloud Connector 経由で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。

## Rendezvous V2

July 16, 2024

Citrix Gateway サービスを使用する場合、Rendezvous プロトコルにより、トラフィックが Citrix Cloud Connector をバイパスして、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 つのタイプがあります: 1) VDA 登録とセッション仲介のための制御用トラフィック、2) HDX セッショントラフィック。

Rendezvous V1 では、HDX セッショントラフィックが Cloud Connector をバイパスできますが、それでも、Cloud Connector が VDA 登録とセッション仲介のためのすべての制御用トラフィックにプロキシを使用する必要があります。

シングルセッションおよびマルチセッションの Linux VDA で Rendezvous V2 を使用するために、標準の AD ドメイン参加マシンと非ドメイン参加マシンがサポートされています。ドメイン非参加マシンでは、Rendezvous V2 は、HDX トラフィックと制御用トラフィックの両方が Cloud Connector をバイパスできるようにします。

## 要件

Rendezvous V2 を使用するための要件は次のとおりです:

- Citrix Workspace と Citrix Gateway サービスを使用した環境へのアクセス。
- コントロールプレーン: Citrix DaaS (旧称 Citrix Virtual Apps and Desktops サービス)。
- VDA バージョン 2201 以降。
  - HTTP プロキシおよび SOCKS5 プロキシには、バージョン 2204 以降が必要です。
- Citrix ポリシーで Rendezvous プロトコルを有効にします。詳しくは、「[Rendezvous プロトコルポリシー設定](#)」を参照してください。
- VDA は以下にアクセスする必要があります:
  - TCP 443での[https://\\*.xendesktop.net](#)。この方法ですべてのサブドメインを許可できない場合は、[https://<customer\\_ID>.xendesktop.net](#)を使用できます。<customer\_ID>は、Citrix Cloud 管理者ポータルに表示される Citrix Cloud 顧客 ID です。
  - [https://\\*.nssvc.net](#) (すべてのサブドメインを含みます)。この方法ですべてのサブドメインをホワイトリストに登録できない場合、代わりに[https://\\*.c.nssvc.net](#)および[https://\\*.g.nssvc.net](#)を使用します。詳しくは、Citrix Cloud のドキュメント (Virtual Apps and Desktops サービス内) の「[インターネット接続の要件](#)」セクションおよび Knowledge Center の記事[CTX270584](#)を参照してください。
- VDA は、前述のアドレスに接続する必要があります:
  - TCP 443 では、TCP Rendezvous 用。
  - UDP 443 では、EDT Rendezvous 用。

## プロキシ構成

VDA は、Rendezvous を使用する場合、制御用トラフィックと HDX セッショントラフィックの両方のプロキシを介した接続をサポートします。どちらのタイプのトラフィックも要件と考慮事項が異なるため、慎重に確認してください。

### 制御用トラフィックプロキシの考慮事項

- HTTP プロキシのみがサポートされています。
- パケットの暗号化解除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用トラフィックが傍受、暗号化解除、または検査されないように、例外を構成します。信頼済みの証明書が見つからない場合は、失敗します。
- プロキシ認証はサポートされていません。
- 制御用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：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\  
VirtualDesktopAgent" -t "REG_SZ" -v "ProxySettings" -d "http  
://<URL or IP>:<port>" --force
```

### HDX トラフィックプロキシの考慮事項

- 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 トラフィックに対してプロキシを構成するには、[\[Rendezvous プロキシの構成\]](#) ポリシー設定を使用します。
- パケットの暗号化解除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、暗号化解除、または検査されないように、例外を構成します。信頼済みの証明書が見つからない場合は、失敗します。
- HTTP プロキシでは、Negotiate および Kerberos 認証プロトコルを使用して、マシンベースの認証がサポートされています。プロキシサーバーに接続するとき、**Negotiate** 認証スキームによって Kerberos プロトコルが自動的に選択されます。Kerberos は、Linux VDA でサポートされてい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、セッションの確立時に HTTP/<proxyURL>形式の SPN を生成します。この場合、プロキシ URL は **Rendezvous** プロキシの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 プロキシによる認証は、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、要件で指定されている Gateway サービスアドレス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。
- EDT を介したデータ転送をサポートしているのは、SOCKS5 プロキシのみです。HTTP プロキシの場合、ICA のトランスポートプロトコルとして TCP を使用します。

### 透過プロキシ

透過 HTTP プロキシは Rendezvous でサポートされています。ネットワークで透過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

以下は、ご使用の環境で Rendezvous を構成するための手順です：

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 をインストールした後、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：

```
1 /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 [許可] に設定します。Rendezvous プロトコルはデフォルトでは無効になっています。Rendezvous プロトコルが有効（許可）の場合、Rendezvous V1 ではなく V2 が有効になります。
  - Citrix ポリシーフィルターが正しく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。
  - 別のポリシーを上書きしないように、Citrix ポリシーの優先度が正しいことを確認してください。

## Rendezvous の検証

セッションが Rendezvous プロトコルを使用しているかどうかを確認するには、ターミナルで `/opt/Citrix/VDA/bin/ctxquery -f iP` コマンドを実行します。

表示されるトランスポートプロトコルは、接続の種類を示しています：

- TCP Rendezvous: TCP - TLS - CGP - ICA
- EDT Rendezvous: 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

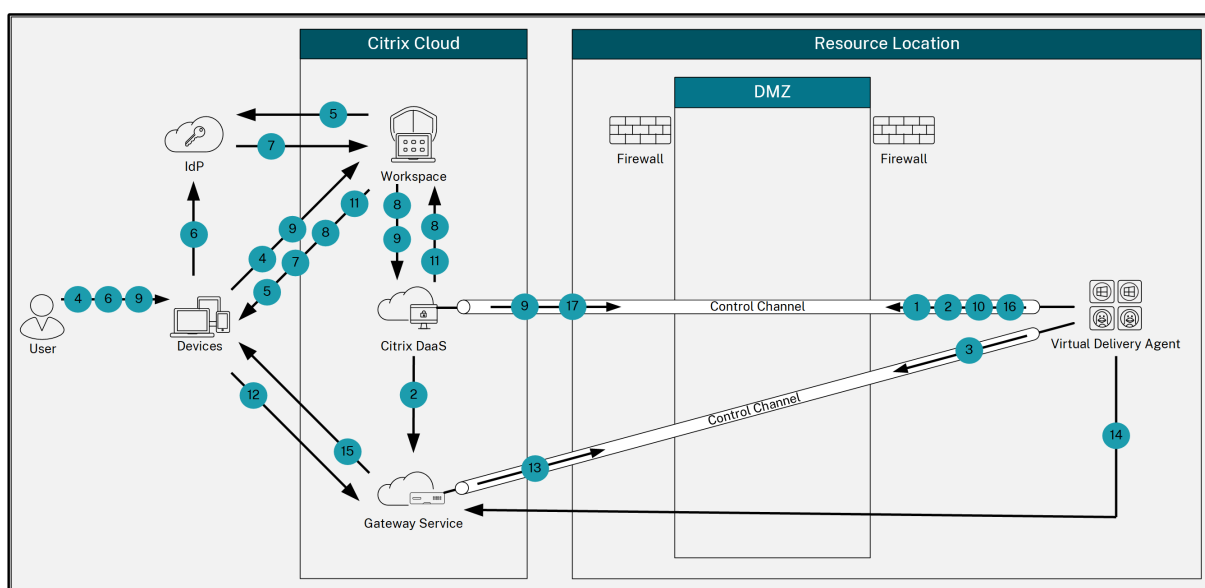
Rendezvous V2 が使用されている場合、プロトコルのバージョンは 2.0 を表示します。

ヒント:

Rendezvous が有効で VDA が Citrix Gateway サービスに直接到達できない場合、VDA はフォールバックし Cloud Connector を介して HDX セッションにプロキシ接続します。

## Rendezvous トラフィックフロー

次の図は、Rendezvous のトラフィックフローに関する一連の手順を示しています。



1. VDA は、Citrix Cloud との WebSocket 接続を確立し、登録します。
2. VDA は Citrix Gateway サービスに登録し、専用のトークンを取得します。
3. VDA は、Gateway サービスとの永続的な制御接続を確立します。
4. ユーザーは Citrix Workspace に移動します。
5. Workspace は認証構成を評価し、認証のためにユーザーを適切な ID プロバイダーにリダイレクトします。
6. ユーザーは自分の資格情報を入力します。
7. ユーザーの資格情報が正常に検証された後、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て表示します。
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 は要求を Citrix DaaS に送信し、Citrix DaaS は接続を仲介し、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 経由で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 サービスに接続し、接続するリソースを認証および識別するための起動チケットを提供します。

13. Gateway サービスは、接続情報を VDA に送信します。
14. VDA は、Gateway サービスへの直接接続を確立します。
15. Gateway サービスは、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、適用するポリシーを VDA に送信します。

## Secure HDX (Technical Preview)

August 20, 2024

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

エンドツーエンド暗号化 (E2EE) 機能により、Citrix Gateway などの中間ネットワーク要素が ICA トラフィックを暗号化解除することができなくなります。これにより、環境のセキュリティ態勢が向上し、構成と管理が簡単になります。

### システム要件

- Linux VDA 2311 (最小バージョン)
- Delivery Controller 2308 (最小バージョン)
- StoreFront 2308 (最小バージョン)
- Windows 向け Citrix Workspace アプリ 2308 (最小バージョン)

### 構成

#### エンドツーエンド暗号化の有効化

エンドツーエンド暗号化 (E2EE) 機能はデフォルトでは無効になっています。これを有効にするには、Citrix Studio で **Secure HDX** ポリシーを [有効] に設定します。

#### 証明書の書き換えのスケジュール

エンドツーエンド暗号化 (E2EE) 機能には、Linux VDA 上の **ctxcertmgr** サービスが管理する自己署名証明書とその秘密キーが必要です。

**ctxcertmgr** サービスが開始または再起動すると、新しい自己署名証明書が作成されます。デフォルトでは、**ctxcertmgr** サービスは証明書 (秘密キーを含む) を 7 日ごとに、午前 2 時に更新します。次のようなレジストリ設定を使用して証明書の書き換えをスケジュール設定することもできます：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -  
  t "REG_SZ" -v "CaRotationStartDate" -d "2023-10-19" --force  
2  
3 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -  
  t "REG_SZ" -v "CaRotationTime" -d "00:45:30" --force  
4  
5 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -  
  t "REG_DWORD" -v "CaRotationPeriod" -d "0x00000005" --force
```

上記の例では、初回の証明書の書き換えの時刻は、2023年10月19日の0時45分30秒に設定されています。その後、**ctxcermgr** サービスは5日ごとに、0時45分30秒に証明書を書き換えます。スケジュールされた日時は、Linux VDA 上の日時です。

## DTLS によるユーザーセッションの保護

July 16, 2024

DTLS 暗号化機能は、7.18 リリースから完全にサポートされます。この機能は Linux VDA ではデフォルトで有効になっています。詳しくは、「[Transport Layer Security](#)」を参照してください。

### DTLS 暗号化の有効化

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、[**HDX** アダプティブトランスポート] ポリシーが [優先] または [診断モード] に設定されていることを確認します。

### Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**/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。このツールで使用できる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、DTLS 1.2、DTLS 1.3 をサポートし、デフォルトで DTLS 1.2 を使用します。Citrix Workspace アプリで使用されている DTLS のバージョン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 が使用されている必要があります。Citrix Workspace アプリが DTLS 1.0 のみをサポートしている場合（Citrix Receiver for Windows 4.11 の場合など）は、**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 によるユーザーセッションの保護

July 16, 2024

バージョン 7.16 以降、Linux VDA は、ユーザーセッションのセキュリティ保護のために TLS 暗号化をサポートします。TLS 暗号化はデフォルトでは無効になっています。

### TLS 暗号化を有効にする

ユーザーセッションを保護するために TLS 暗号化を有効にするには、Linux VDA と Delivery Controller (Controller) の両方で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、次のセクションがあります。

- 証明書
- 暗号化されていない秘密キー
- 中間証明書 (必須ではありません)

サーバー証明書の例:



## TLS 暗号化を有効にする

**Linux VDA** で **TLS** 暗号化を有効にする Linux VDA では、**/opt/Citrix/VDA/sbin** ディレクトリの **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。このスクリプトで使用できる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。

```
root@xui804:~# /opt/Citrix/VDA/sbin/enable_vdassl.sh
==Enable/Disable SSL on Linux VDA==
To enable SSL, a certificate file must be specified, otherwise the local certificate file under
/etc/xdm/.sslkeystore/ is used. If the local certificate file does not exist, the command
fails. You can specify the SSL port number, version and cipher suite, otherwise, their default
values are used!

Usage: enable_vdassl.sh -Disable
       Disable Linux VDA SSL.

Usage: enable_vdassl.sh -Enable [-Certificate <CERT-FILE>] [-SSLPort <SSL-PORT-NUMBER>]
       [-SSLMinVersion <SSL-MIN-VERSION>] [-SSLCipherSuite <SSL-CIPHER-SUITE>]
       Enable Linux VDA SSL.

Options:
-Certificate <CERT-FILE>
  Specify a certificate file, where <CERT-FILE> must include the full file path. Only one format
  is currently supported, that is PEM.

-RootCertificate <ROOT-CERT-FILE>
  Specify a root certificate file, where <ROOT-CERT-FILE> must include the full file path. The root certificate will be put in the local keystore(under /etc/xdm/.sslkeystore/cacerts).

-SSLPort <SSL-PORT-NUMBER>
  Specify an SSL port number. Unless otherwise specified, the default port 443 used.

-SSLMinVersion <TLS_1.0|TLS_1.1|TLS_1.2|TLS_1.3>
  Specify SSL version. Unless otherwise specified, the default value TLS_1.2 is used.

-SSLCipherSuite <GOV|COM|ALL>
  Specify an SSL Cipher suite. Unless otherwise specified, the default value GOV is used.

Examples:
enable_vdassl.sh -Enable -Certificate "/home/cert001.pem"
Enable Linux VDA SSL using Certificate cert001.pem.

enable_vdassl.sh -Enable -RootCertificate "/home/rootCR.cer"
Enable Linux VDA SSL using Root Certificate rootCR.cer with local certificate(under /etc/xdm/.sslkeystore).

enable_vdassl.sh -Enable -SSLPort 445
Enable Linux VDA SSL on port 445 using local certificate(under /etc/xdm/.sslkeystore).

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445
Enable Linux VDA SSL using Certificate cert001.pem on port 445, with default SSLMinVersion and SSLCipherSuite.

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445 -SSLMinVersion "TLS_1.2"
Enable Linux VDA SSL using Certificate cert001.pem on port 445 with SSLMinVersion TLS_1.2 and default SSLCipherSuite..

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445 -SSLMinVersion "TLS_1.2" -SSLCipherSuite "GOV"
Enable Linux VDA SSL using Certificate cert001.pem on port 445 with SSLMinVersion TLS_1.2 and SSLCipherSuite GOV.
```

ヒント: 各 Linux VDA サーバーにサーバー証明書をインストールし、各 Linux VDA サーバーとクライアントにルート証明書をインストールする必要があります。

## Controller で TLS 暗号化を有効にする

注:

TLS 暗号化は、デリバリーグループ全体に対してのみ有効にすることができます。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、次のコマンドを順番に実行して、ターゲットのデリバリーグループの TLS 暗号化を有効にします。

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule -HdxSslEnabled $true`

注:

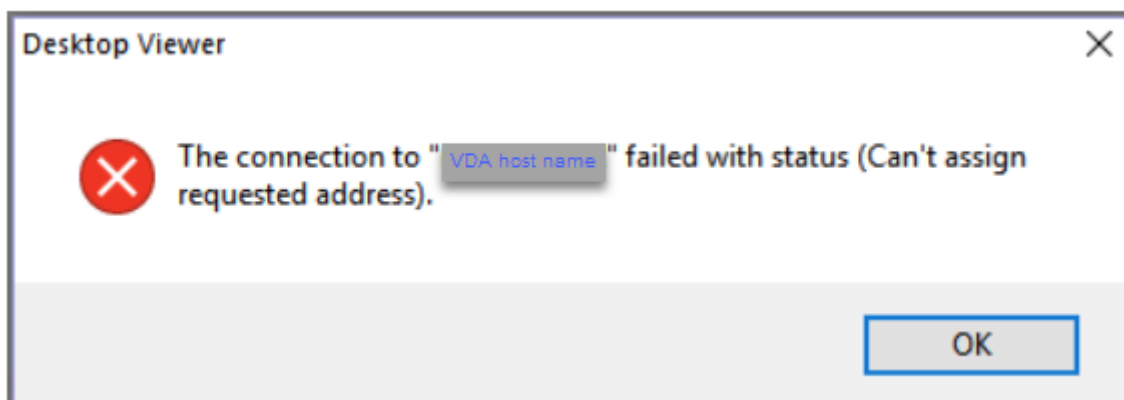
VDA FQDN のみが ICA セッションファイルに含まれるように、`Set-BrokerSite -DnsResolutionEnabled $true` コマンドを実行することもできます。このコマンドは、DNS 解決を有効にします。DNS 解決を無効にすると、ICA セッションファイルは VDA の IP アドレスを開示し、`SSLProxyHost` や `UDPDTLSPort` などの TLS 関連項目に対してのみ FQDN を提供します。

Controller で TLS 暗号化を無効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule -HdxSslEnabled $false`
3. `Set-BrokerSite -DnsResolutionEnabled $false`

### トラブルシューティング

公開されたデスクトップセッションにアクセスしようとする、Windows 向け Citrix Workspace アプリで、次の「要求されたアドレスを割り当てることができません」というエラーが発生することがあります:



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。

### セッション画面の保持

August 20, 2024

Citrix でサポートされているすべての Linux プラットフォームには、セッション画面の保持機能が導入されています。セッション画面の保持は、デフォルトで有効になっています。

セッション画面の保持によって ICA セッションは、ネットワークの中断を挟んでもシームレスに再接続されます。セッション画面の保持について詳しくは、「[クライアントの自動再接続とセッション画面の保持](#)」を参照してください。

注:

セッション画面の保持の接続を介して送信されるデータは、デフォルトではプレーンテキストです。セキュリティを確保するため、TLS 暗号化を有効にすることをお勧めします。TLS 暗号化について詳しくは、「[TLS によるユーザーセッションの保護](#)」を参照してください。

### 構成

#### Citrix Studio のポリシー設定

Citrix Studio で、セッション画面の保持に関する次のポリシーを設定できます。

- セッション画面の保持
- セッション画面の保持のタイムアウト
- セッション画面の保持のポート番号
- 再接続 UI の透過レベル

詳しくは、「[セッション画面の保持のポリシー設定](#)」および「[クライアントの自動再接続のポリシー設定](#)」を参照してください。

注:

セッション画面の保持の接続またはセッション画面の保持のポート番号ポリシーを設定したら、VDA サービスと HDX サービスをこの順序で再起動して設定を有効にします。

#### Linux VDA の設定

- セッション画面の保持の **TCP** リスナーを有効または無効にする

デフォルトでは、セッション画面の保持の TCP リスナーは有効になっており、ポート 2598 でリッスンします。リスナーを無効にするには、次のコマンドを実行します。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cgp" -v "fEnableWinStation" -d "0x00000000"
```



## 注:

設定を有効にするには、HDX サービスを再起動してください。TCP リスナーを無効にしても、セッション画面の保持は無効になりません。セッション画面の保持の接続ポリシーによって機能が有効になっている場合、セッション画面の保持は他のリスナー（SSL など）を介して引き続き利用できます。

- セッション画面の保持のポート番号

次のコマンドを使用して、セッション画面の保持の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）。

```
1 /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 の設定は無視されます。VDA 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設定されていることを確認します。

- サーバーからクライアントへの **Keep-Alive** の間隔

Keep-Alive メッセージは、セッション中にアクティビティがない場合（例：マウスが移動しない、画面の更新がない）、Linux VDA とクライアント間で送信されます。Keep-Alive 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、セッションは、クライアントが再接続するまで中断されます。この設定では、Keep-Alive メッセージの送信間隔を秒単位で指定します。デフォルトでは、この設定は構成されていません。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）。

```
1 /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 **Keep-Alive** の間隔

この設定では、ICA クライアントから Linux VDA に送信される Keep-Alive メッセージの送信間隔を秒単位で指定します。デフォルトでは、この設定は構成されていません。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）。

```
1 /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

ポリシーの設定によってセッション画面の保持を有効にした後に、セッションを起動できない。

この問題を解決するには、以下の手順に従います。

1. Citrix Studio のポリシー設定でセッション画面の保持を有効にした後、VDA サービスと HDX サービスがこの順序で再起動されることを確認します。
2. VDA で、次のコマンドを使用してセッション画面の保持の 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)。

```
1 netstat -an | grep 2598
```

セッション画面の保持のポートに 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。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cgp" -v "fEnableWinStation" -d "0x00000001"
```

## セッションの録画 (プレビュー)

July 16, 2024

Linux VDA でホストされているセッションの録画と再生ができます。

注:

この機能はプレビュー段階です。プレビュー機能は、一部が英語のままの場合があります。また、実稼働環境以外での使用をお勧めします。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。

## セッションの録画の有効化と無効化

Linux VDA のセッションの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:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "SmAudAllowed" -d "0x00000001" --force
```

```
1 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 でセッションの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ユーザーに通知されます。

## 録画のファイルサイズの指定

録画ファイルのサイズが大きくなるにつれて、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再生箇所を変更するときに反応が遅くなります。ファイルサイズを制御するにはファイルのしきい値を指定します。録画ファイルがこの限界に達すると、現在のファイルが閉じられ、録画を続行するために追加のファイルが作成されます。この操作をロールオーバーと呼びます。

次のコマンドを使用すると、1つのロールオーバーに2つのしきい値を指定できます：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
   CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
   RolloverFileSizeInMB" -d "0x00000032" --force
2
3 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
   CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
   RolloverTimeInHours" -d "0x0000000c" --force
```

- **RolloverFileSizeInMB**。このサイズに達すると現在のファイルが閉じ、新しいファイルが開きます。デフォルトでは、ロールオーバーはサイズが50MBを超えると発生します。サポートされる値は、10~300です。
- **RolloverTimeInHours**。この時間に達すると、現在のファイルが閉じ、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。サポートされる値は、1~24です。

ロールオーバーは、上記の2つの条件の最初の1つが満たされたときに発生します。たとえば、ファイルサイズとして17MB、時間として6時間を指定したとします。録画ファイルが3時間で17MBに達すると、セッションの録画によりファイルが閉じられ、新しいファイルが開きます。

多くの小さなファイルが作成されないように、ファイルサイズに指定された値にかかわらず、少なくとも1時間が経過するまでロールオーバーは起こりません。この規則の例外は、ファイルサイズが300MBを超えた場合です。

## 録画の保存場所の指定

デフォルトでは、録画ファイルは/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：

```
1 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
```

録画ファイルは、ローカルドライブに、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への適切なアクセス権限を構成し、ユーザーctxsrvrに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、次の手順を実行して、Session Recording プレーヤーまたは Session Recording Web プレーヤーをインストールします：

1. Citrix アカウント資格情報を使用して、[Citrix Virtual Apps and Desktops のダウンロードページ](#)にアクセスして、製品ファイルをダウンロードします。ファイルを解凍します。
2. SessionRecordingPlayer.msi および SessionRecordingWebPlayer.msi をダブルクリックし、手順に従ってインストールを完了します。

ヒント：

Session Recording Web プレーヤーを使用するには、Session Recording サーバーのみにインストールし、Session Recording サーバーで録画が利用できることを確認します。詳しくは、[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、録画通知が中央に配置されない場合があります。

## 仮想チャネル SDK（プレビュー）

July 16, 2024

Linux VDA 用の仮想チャネルソフトウェア開発キット（SDK）を使用すると、VDA 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 向け Citrix 仮想チャネル SDK のドキュメント](#)を参照してください。

Linux VDA 向け Citrix 仮想チャネル 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。Citrix Virtual Apps and Desktops の適切なバージョンを展開し、[コンポーネント] をクリックして 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段階です。プレビュー機能は、一部が英語のままの場合があります。また、実稼働環境以外での使用をお勧めします。Citrix テクニカルサポートは、Technical Preview 機能で見つかった問題をサポートしません。

## Wayland (プレビュー)

August 20, 2024

Linux VDA は RHEL 9.4/9.3/9.2、Rocky Linux 9.4/9.3/9.2、Ubuntu 22.04 の GNOME で Wayland をサポートします。次の機能は、Wayland で完全にテスト済みです：

- 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段階です。プレビュー機能は、一部が英語のままの場合があります。また、実稼働環境以外での使用をお勧めします。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。
- 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。

```
1 /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** に設定されています。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どうかを確認する

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 の使用による次の制限事項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がVDAのキーボードレイアウトと同期しません。

## ベストプラクティス

July 16, 2024

このセクションでは、次の手順について説明します：

- [Google Cloud Platform \(GCP\)](#) で [Machine Creation Services \(MCS\)](#) を使用した [Linux VDA](#) の作成

## Google Cloud Platform (GCP) で Machine Creation Services (MCS) を使用した Linux VDA の作成

August 20, 2024

MCS を使用して GCP 上に Linux VDA を作成するには、次の手順を実行します：

手順 1: [GCP 上に Linux 仮想マシン \(VM\) を作成する](#)

手順 2: [GCP サービスアカウントを作成する](#)

手順 3: [Citrix Studio](#) で GCP へのホスト接続を作成する

手順 4: [Linux VDA マスターイメージを準備する](#)

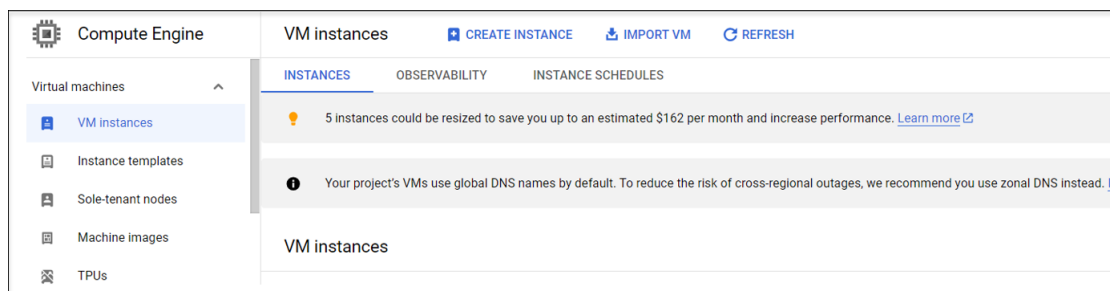
手順 5: [マシンカタログを作成する](#)

手順 6: [デリバリーグループを作成する](#)

### 手順 1: GCP 上に Linux VM を作成する

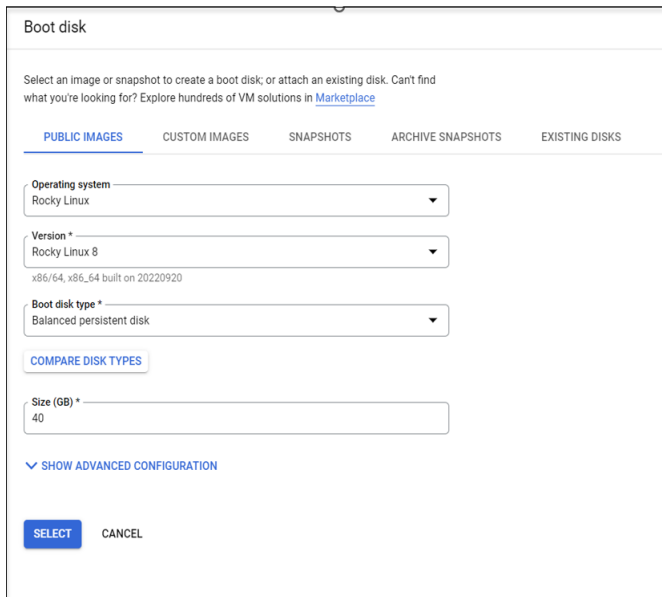
手順 1a: Google Cloud コンソールで、**[Compute Engine]** > **[VM instances]** に移動します。

手順 1b: [VM instances] ページで、**[CREATE INSTANCE]** をクリックして VM インスタンスを作成します。



手順 **1c**: 次の構成を作成し、他の構成はデフォルトのままにします:

- VM インスタンスの名前を入力します。
- VM をホストするリージョンとゾーンを選択します。
- (オプション) VM に GPU を追加します。詳しくは、この記事の「手順 **4c**」を参照してください。
- **[Boot disk]** セクションで、VM のオペレーティングシステムとディスクサイズを選択します。例:



Boot disk

Select an image or snapshot to create a boot disk or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

**PUBLIC IMAGES** CUSTOM IMAGES SNAPSHOTS ARCHIVE SNAPSHOTS EXISTING DISKS

Operating system  
Rocky Linux

Version \*  
Rocky Linux 8  
x86\_64, x86\_64 built on 20220920

Boot disk type \*  
Balanced persistent disk

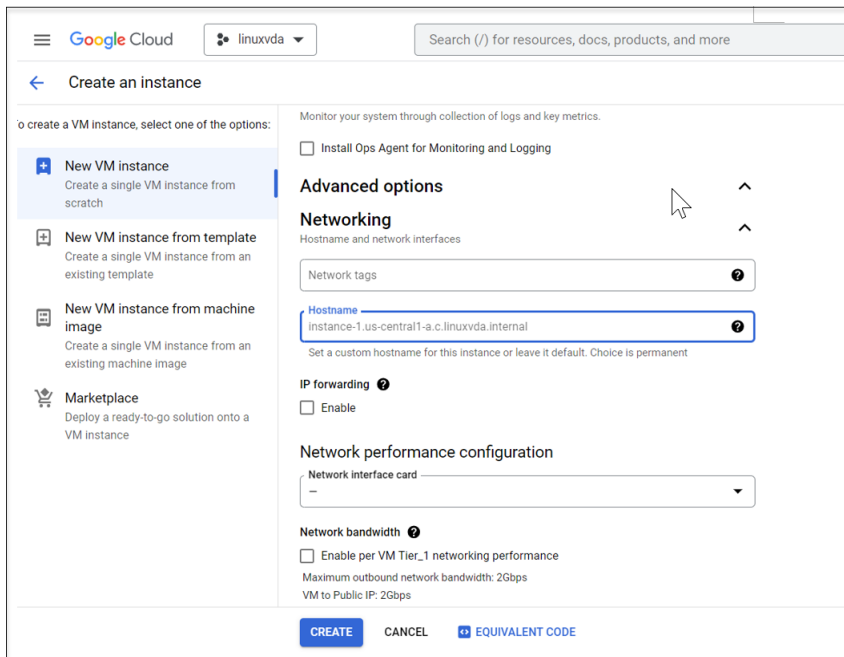
[COMPARE DISK TYPES](#)

Size (GB) \*  
40

[SHOW ADVANCED CONFIGURATION](#)

**SELECT** CANCEL

- **[Advanced options]** > **[Networking]** に移動して、**[Hostname]** フィールドに完全修飾ドメイン名を設定します。



Google Cloud linuxvda Search (/) for resources, docs, products, and more

← Create an instance

to create a VM instance, select one of the options:

- New VM instance**  
Create a single VM instance from scratch
- New VM instance from template  
Create a single VM instance from an existing template
- New VM instance from machine image  
Create a single VM instance from an existing machine image
- Marketplace  
Deploy a ready-to-go solution onto a VM instance

Monitor your system through collection of logs and key metrics.

Install Ops Agent for Monitoring and Logging

**Advanced options**

**Networking**  
Hostname and network interfaces

Network tags

**Hostname**  
instance-1.us-central1-a.c.linuxvda.internal  
Set a custom hostname for this instance or leave it default. Choice is permanent

**IP forwarding**

Enable

**Network performance configuration**

Network interface card

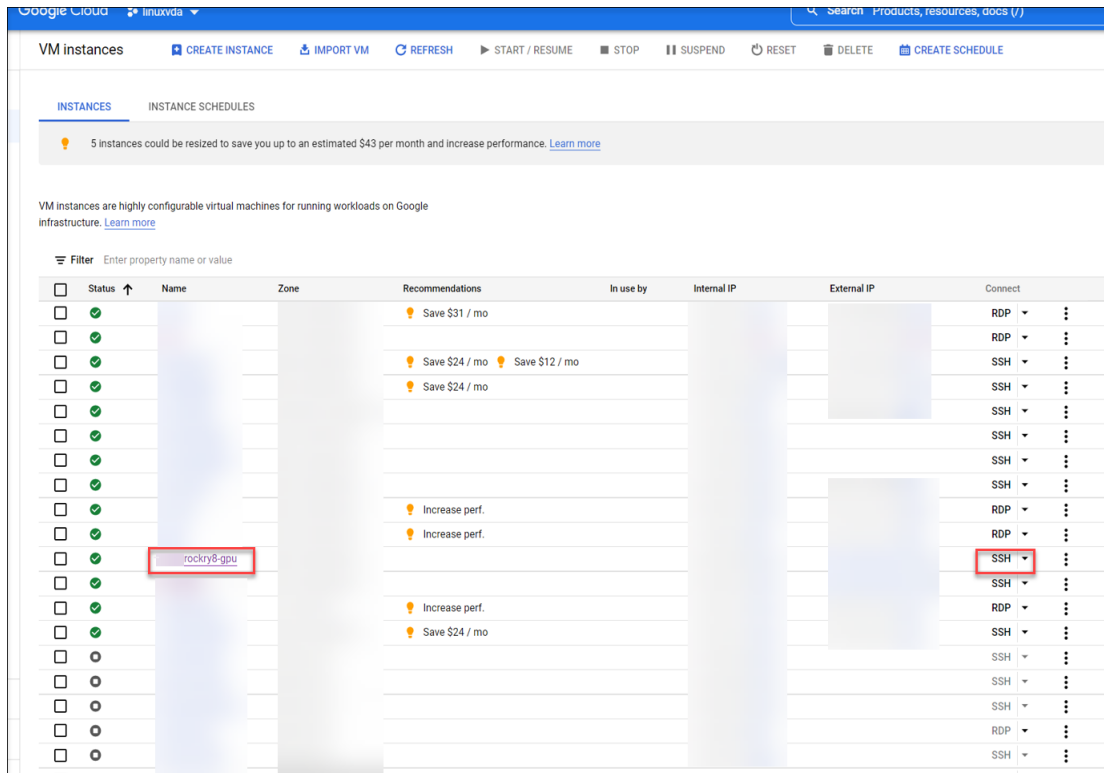
**Network bandwidth**

Enable per VM Tier\_1 networking performance  
Maximum outbound network bandwidth: 2Gbps  
VM to Public IP: 2Gbps

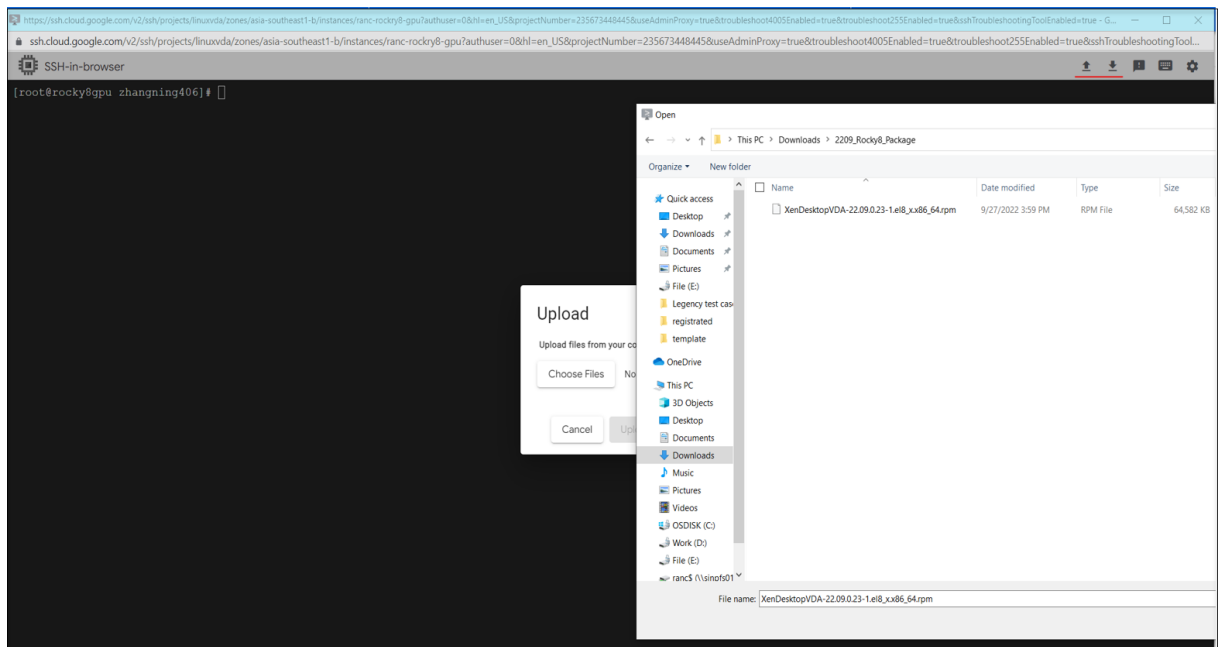
**CREATE** CANCEL [EQUIVALENT CODE](#)

手順 **1d**: **[Create]** をクリックして VM インスタンスを作成します。

手順 **1e**: VM が作成されたら、**Compute Engine** ダッシュボードに戻り、一覧から VM インスタンスを見つけて、SSH ボタンをクリックし、VM に接続します。



手順 **1f**: Web ベースの SSH クライアントを介して Linux VDA パッケージを VM にアップロードします。



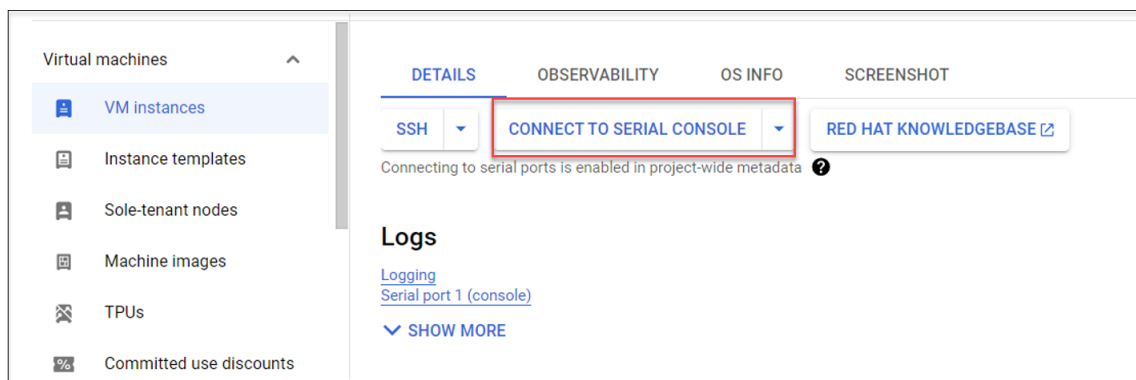
手順 **1g**: SSH を使用した VM へのアクセスの失敗を回避します。

再起動後に VM 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、仮想マシンに初めてログオン



するときにルートパスワードを設定し、ルートとして仮想マシンにログオンできることを確認します。次に、仮想マシンを再起動した後、コンソールで次のコマンドを実行します：

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```



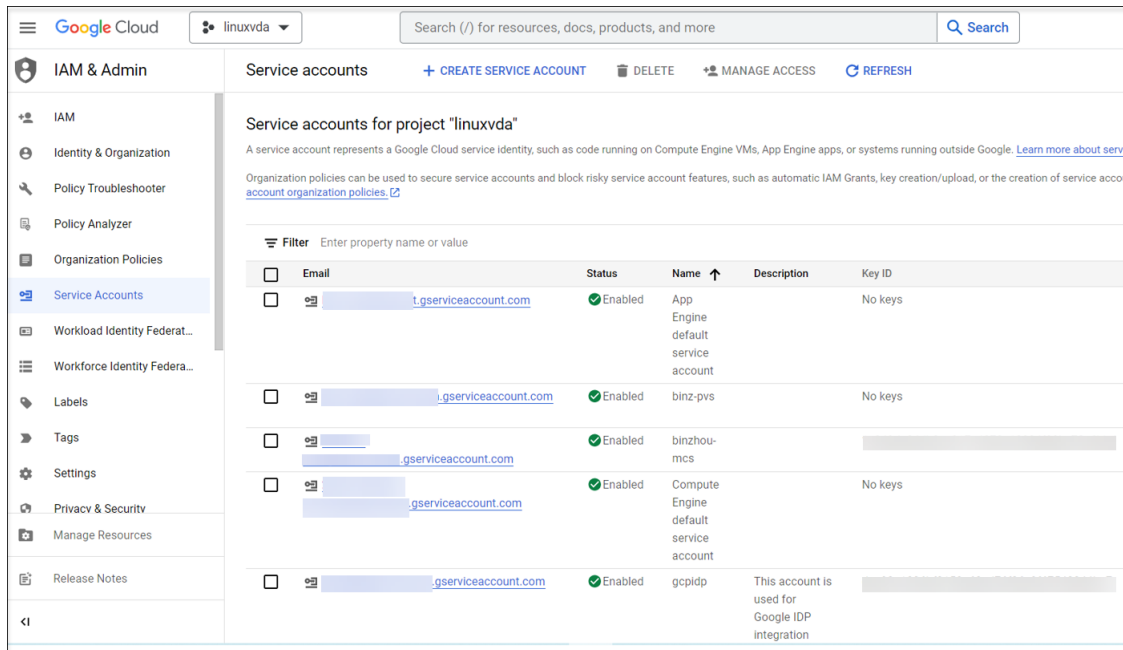
### 手順 2: GCP サービスアカウントを作成する

このセクションでは、サービスアカウントキーの作成やサービスアカウントへの必要な役割の付与など、GCP サービスアカウントの作成方法について説明します。

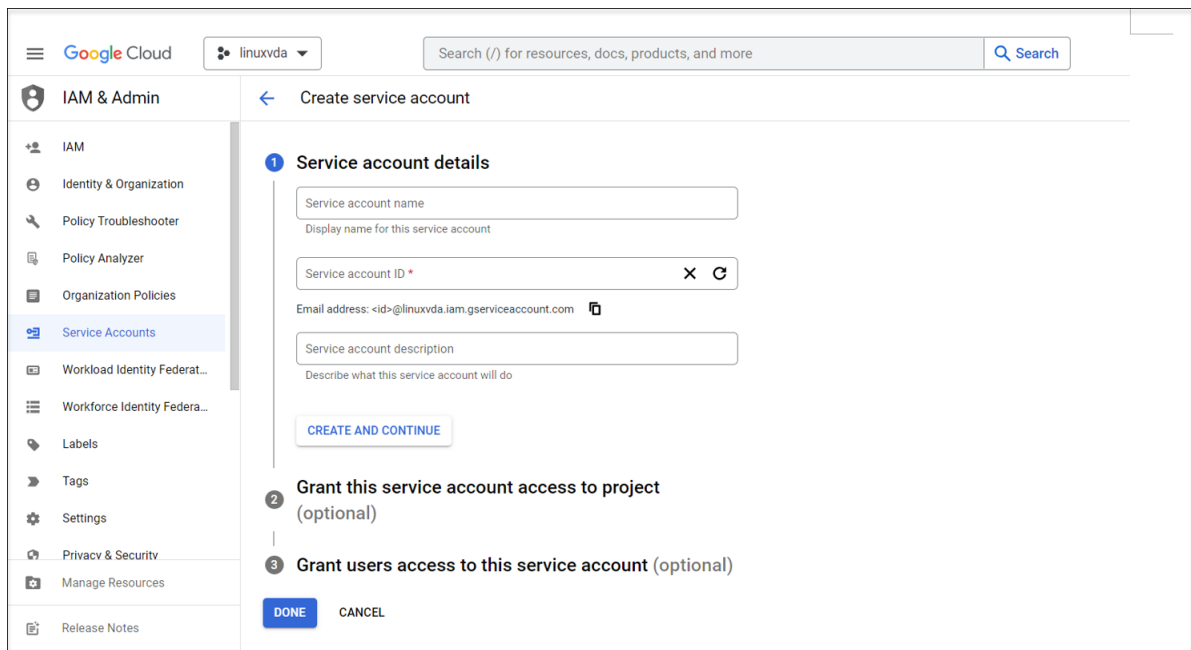
注：

GCP サービスアカウントを作成する管理者には、[Service Account Admin](#)(roles/iam.serviceAccountAdmin) の IAM 役割が付与されていることを確認します。

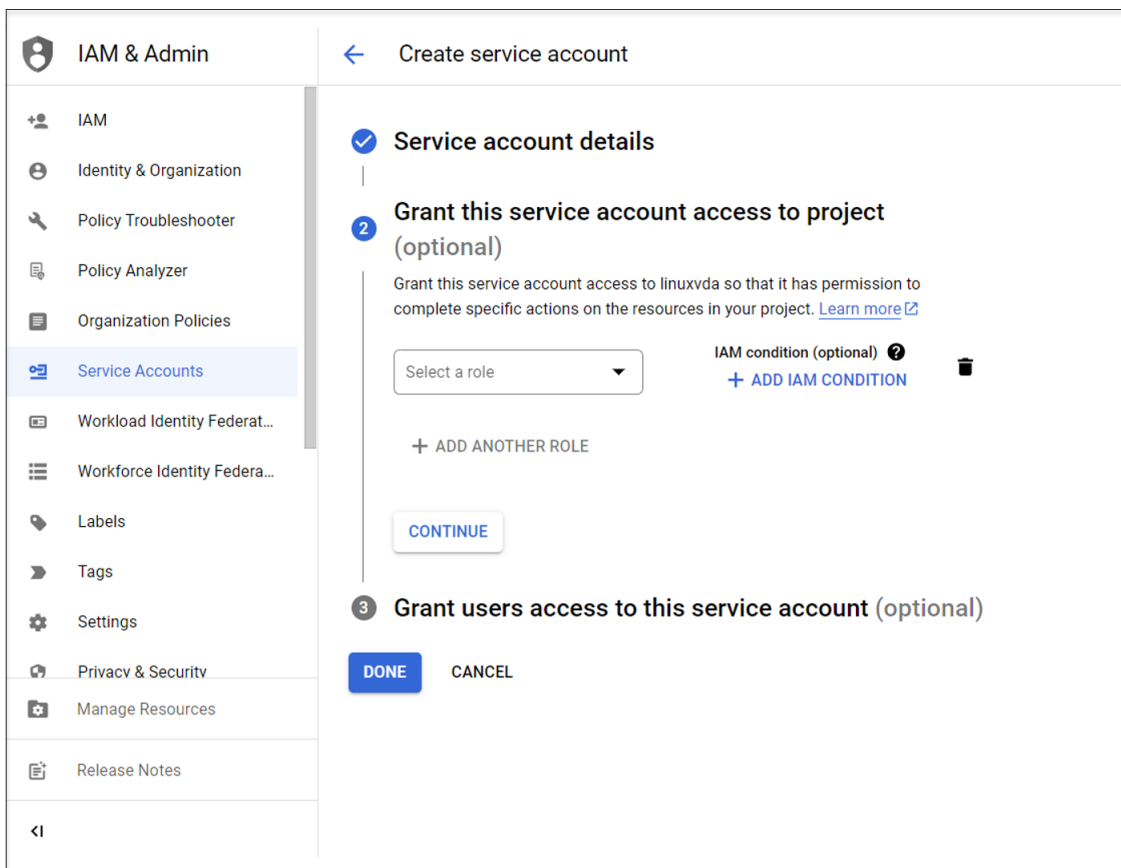
手順 2a: Google Cloud コンソールで **[IAM & Admin]** > **[Service Accounts]** に移動して、**[Create Service Account]** タブをクリックします。



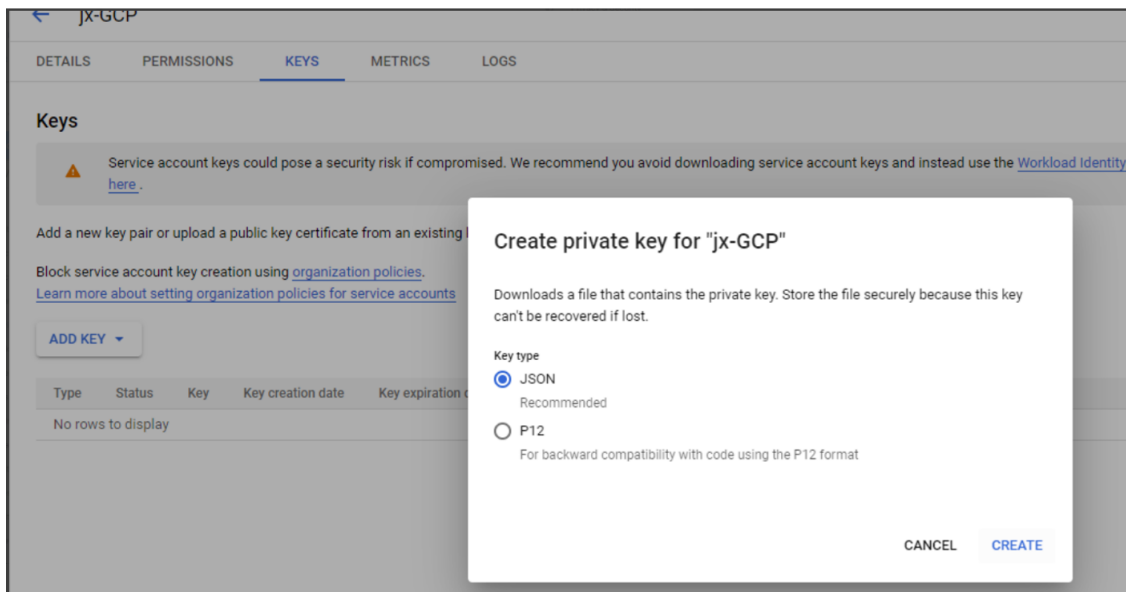
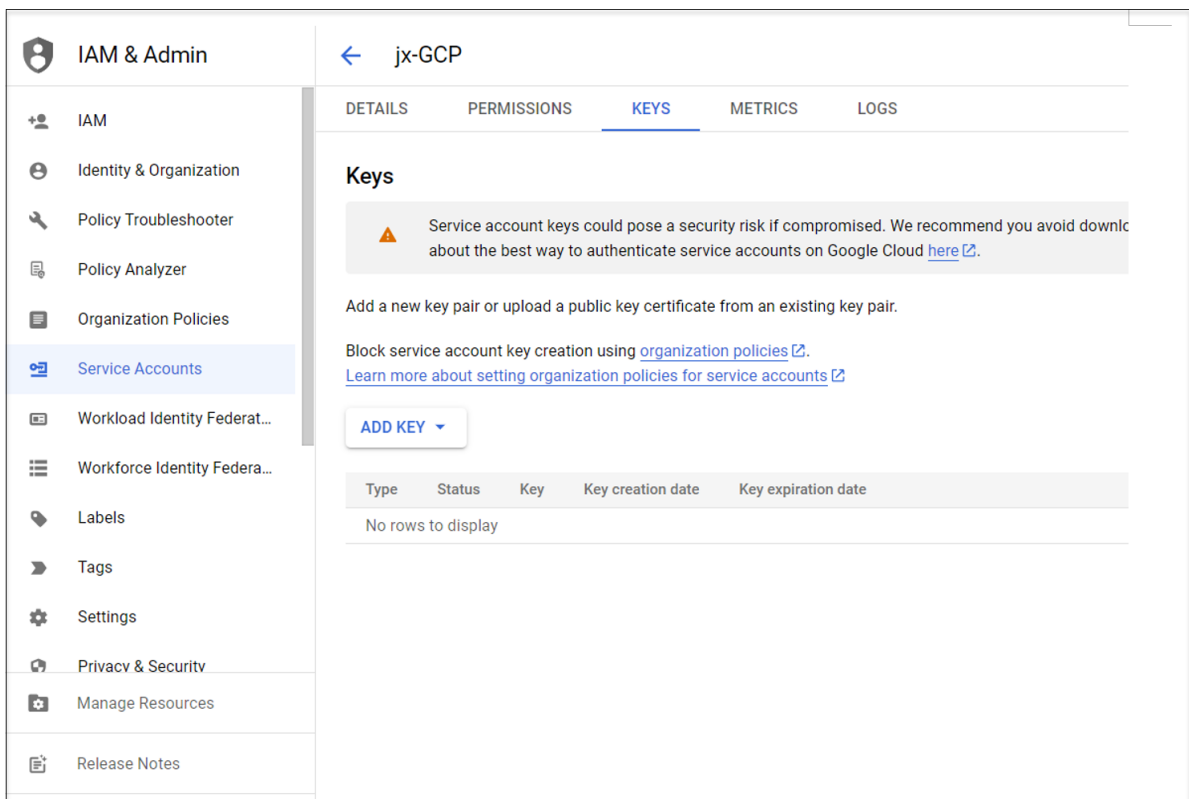
手順 **2b**: [Service account details] の手順でフィールドに値を入力し、[Create and continue] をクリックします。



手順 **2c**: オプションの手順をスキップし、下部にある [Done] をクリックします。



手順 **2d**: 再度 **[IAM & Admin]** > **[Service Accounts]** に移動して、**[Create Service Account]** タブをクリックします。新しく作成したサービスアカウントを見つけて、**[Keys]** タブに移動し、**[Add key]** > **[Create new key]** > **[JSON]** > **[Create]** をクリックします。



注:

キーファイルをダウンロードすると、再度ダウンロードすることはできません。

手順 **2e**: Google Cloud コンソールで、[IAM & Admin] > [IAM] の順に移動して、[Add] をクリックします。**New members** フィールドで新しく作成したサービスアカウントを検索して選択し、リソースへのアクセスを許可するサービスアカウントの役割を選択します。[Add another role] をクリックして役割の付与を続け、新しく作

成したサービスアカウントに次のすべての役割を確実に付与します。

- コンピューティング管理者
- ストレージ管理者
- **Cloud Build** エディター
- サービスアカウントユーザー
- クラウドデータストアユーザー
- コンピューティングインスタンス管理者（ベータ）
- 所有者

例:



Role	Excess Permissions
Cloud Build Editor	4/7 excess permissions
Cloud Datastore User	15/16 excess permissions
Compute Admin	524/576 excess permissions
Compute Instance Admin (beta)	121/162 excess permissions
Owner	4804/4871 excess permissions
Service Account User	3/5 excess permissions
Storage Admin	13/24 excess permissions

### 手順 3: Citrix Studio で GCP へのホスト接続を作成する

Google Cloud Platform 仮想化環境に合わせて GCP 環境をセットアップしてから、次の手順を実行して GCP へのホスト接続を作成します。

1. オンプレミスの Delivery Controller の場合は、オンプレミスの Citrix Studio で [構成] > [ホスト] > [接続およびリソースの追加] の順に選択してホスト接続を作成します。クラウドの Delivery Controller の場合は、Citrix Cloud の Web ベースの Studio コンソールで [管理] > [ホスト] > [接続およびリソースの追加] の順に選択し、ホスト接続を作成します。
2. 接続およびリソースの追加ウィザードで、接続の種類として **Google Cloud Platform** を選択します。  
たとえば、Citrix Cloud の Web ベースの Studio コンソールでは次のようになります:

**Add Connection and Resources**

1 Connection  
2 Region  
3 Network  
4 Scopes  
5 Summary

**Connection**

Use an existing connection  
BingTest

Create a new connection

Zone name:  
[Redacted]

Connection type:  
Google Cloud Platform

Service account key:  
Import key...

Service account ID:  
[Redacted]

Connection name:  
[Redacted]

Create virtual machines using:  
 Citrix provisioning tools (Machine Creation Services or Citrix Provisioning)  
 Other tools

Next Cancel 7 ↻

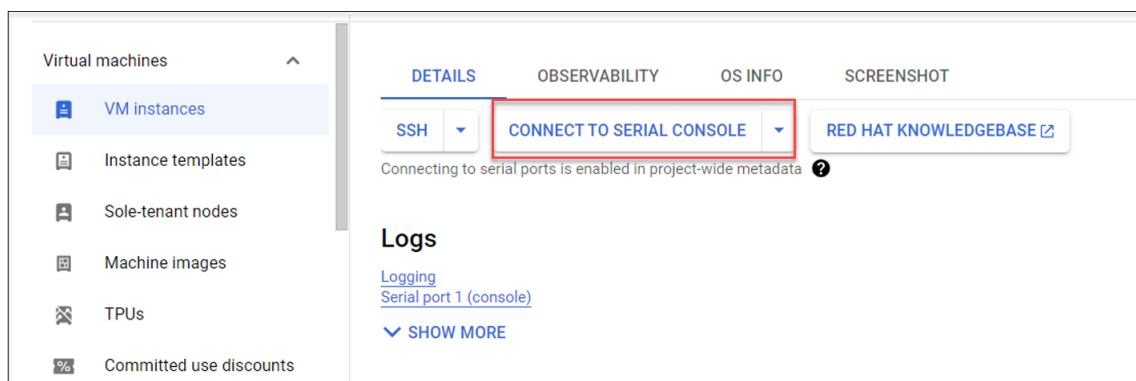
3. GCP アカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードの指示に従って、各ページの操作を行います。特定のページの内容は、選択した接続の種類によって異なります。各ページの操作を終えたら、[概要] ページに到達するまで [次へ] を選択します。詳しくは、「[MCS を使用したドメイン非参加の Linux VDA の作成](#)」の記事にある「[手順 2: ホスト接続の作成](#)」を参照してください。

#### 手順 4: Linux VDA マスターイメージを準備する

手順 4a: (RHEL 8.x/9.x および Rocky Linux 8.x/9.x の場合) イーサネット接続を構成します。

GCP でホストされている RHEL 8.x/9.x および Rocky Linux 8.x/9.x に Linux VDA をインストールすると、イーサネット接続が失われ、仮想マシンの再起動後に Linux VDA にアクセスできなくなることがあります。この問題を回避するには、仮想マシンに初めてログオンするときにルートパスワードを設定し、ルートとして仮想マシンにログオンできることを確認します。次に、仮想マシンを再起動した後、コンソールで次のコマンドを実行します:

```
1 nmcli dev connect eth0
2 systemctl restart NetworkManager
```



手順 **4b**: テンプレート仮想マシンに Linux VDA パッケージをインストールします。

テンプレート VM で次の手順を実行して、Linux VDA パッケージをインストールします:

1. .NET をインストールします。

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

Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みフィードからインストールします。それ以外の場合は、Microsoft パッケージフィードから .NET をインストールします。詳しくは、<https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers> を参照してください。

2. Linux VDA パッケージをインストールします:

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

3. EPEL リポジトリを有効にします:

```
1 sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

手順 **4c**: GCP 上のテンプレート VM にグラフィック処理装置 (GPU) を追加します (オプション)。

1. Google Cloud コンソールで、1 つ以上の GPU をテンプレート VM に追加します。GPU と GCP の追加と削除については、「<https://cloud.google.com/compute/docs/gpus/add-remove-gpus>」を参照してください。

← Create an instance

To create a VM instance, select one of the options:

- New VM instance**  
Create a single VM instance from scratch
- New VM instance from template  
Create a single VM instance from an existing template
- New VM instance from machine image  
Create a single VM instance from an existing machine image
- Marketplace  
Deploy a ready-to-go solution onto a VM instance

Name \*  
instance-1

MANAGE TAGS AND LABELS

Region \*  
us-central1 (Iowa)  
Region is permanent

Zone \*  
us-central1-a  
Zone is permanent

Machine configuration

General purpose Compute optimized Memory optimized **GPUs**

Graphics processing units (GPUs) accelerate specific workloads on your instances such as machine learning and data processing. [Learn More](#)

GPU type  
NVIDIA T4

Number of GPUs  
1

Enable Virtual Workstation (NVIDIA GRID)

Series	Description	vCPUs	Memory	Platform
<b>N1</b>	Balanced price & performance	1 - 96	1.8 - 624 GB	Intel Skylake

Machine type  
Choose a machine type with preset amounts of vCPUs and memory that suit most workloads.

2. 適切な GPU ドライバーをテンプレート VM にインストールします。詳しくは、<https://cloud.google.com/compute/docs/gpus/install-drivers-gpu>を参照してください。

必要な **NVIDIA** ドライバーのバージョン:

Compute Engine 上で実行される NVIDIA GPU は、NVIDIA ドライバーの次のバージョンを使用する必要があります:

- L4 GPU の場合:
  - Linux: 525.60.13 以降
- A100 GPU の場合:
  - Linux: 450.80.02 以降
- T4、P4、P100、および V100 GPU の場合:
  - Linux: 410.79 以降
- K80 GPU の場合 (製品終了):
  - Linux: 410.79 - 最新の R470 バージョン

K80 GPU について NVIDIA は、[R470 ドライバークラッシュ](#)が、デバッグサポートを受けることができる最後のドライバーバージョンであると発表しました。この更新を確認するには、[NVIDIA ソフトウェアサポートマトリックス](#)を参照してください。

インストールスクリプト:

次のスクリプトを使用して、インストールプロセスを自動化できます:

```
1 https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install\_gpu\_driver.py --output install_gpu_driver.py
```



以下のオペレーティングシステムがサポートされています：

インストールスクリプトは、次の Linux ディストリビューションでテスト済みです：

- Debian 11
- Red Hat Enterprise Linux (RHEL) 8
- Rocky Linux 8
- Ubuntu 20/22

このスクリプトを他の Linux ディストリビューションで使用すると、インストールは失敗します。Linux VM の場合、このスクリプトは NVIDIA ドライバーのみをインストールします。

a) インストールスクリプトをダウンロードします。

```
1 curl https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

b) スクリプトへの完全なアクセスを許可します。

```
1 chmod 777 install_gpu_driver.py
```

c) インストールスクリプトを実行します。

```
1 python3 install_gpu_driver.py
```

d) gdm3 で Wayland を無効にします。

- 次のいずれかの場所でディストリビューションの Wayland 構成ファイルを見つけます：
  - /etc/gdm3/custom.conf (Ubuntu)
  - /etc/gdm/custom.conf (CentOS、RHEL、Rocky Linux)
- sudo/root 権限でファイルを開きます。
- 行の最初の # を削除して、**WaylandEnable=false** のコメントを解除します。
- 仮想マシンを再起動します。

e) NVIDIA 510 以降のドライバーをインストールした場合は、GSP ファームウェアを無効にします。

GSP ファームウェアが有効になっている場合は、NVIDIA モジュールパラメーター **NVreg\_EnableGpuFirmware** を 0 に設定して無効にします。

次のエントリを/etc/modprobe.d/nvidia.conf ファイルに追加して、このパラメーターを設定します：

- options nvidia NVreg\_EnableGpuFirmware=0
- /etc/modprobe.d/nvidia.conf ファイルが存在しない場合は、作成します。

この手順を完了するときは、次の点に注意してください：

- sudo を使用してコマンドを実行し、構成ファイルを作成および更新します。

- VM を再起動するには、Linux ターミナルで **sudo reboot** を使用するか、VM を停止して起動します。

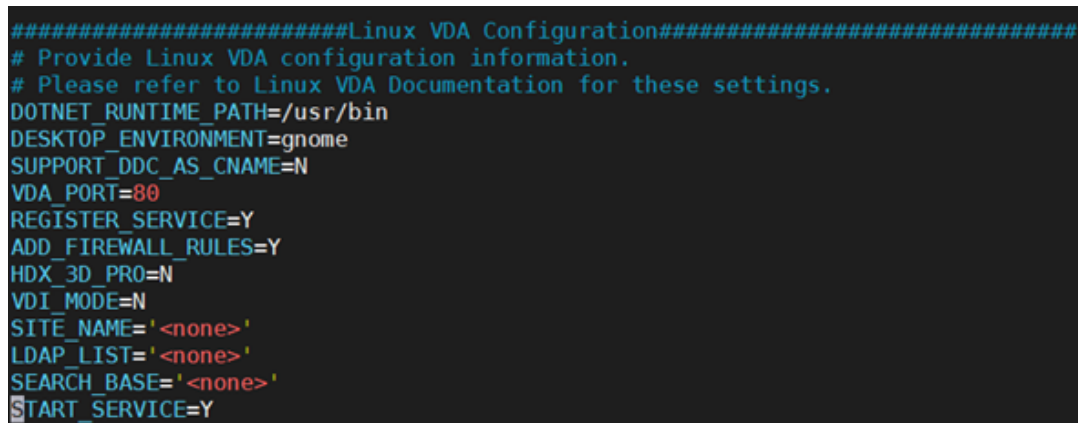
手順 **4d**: MCS 変数を構成します。

/etc/xdl/mcs/mcs.conf ファイルを編集して MCS 変数を構成します。以下は、ドメイン非参加シナリオとドメイン参加済みシナリオで構成できる MCS 変数です:

- ドメイン非参加シナリオの場合

デフォルトの変数値を使用するか、必要に応じて変数をカスタマイズできます (オプション):

```
DOTNET_RUNTIME_PATH=**path-to-install-dotnet-runtime \**
DESKTOP_ENVIRONMENT= **gnome | mate \**
REGISTER_SERVICE=Y | N
ADD_FIREWALL_RULES=Y | N
VDI_MODE=Y | N
START_SERVICE=Y | N
```



```
#####Linux VDA Configuration#####
# Provide Linux VDA configuration information.
# Please refer to Linux VDA Documentation for these settings.
DOTNET_RUNTIME_PATH=/usr/bin
DESKTOP_ENVIRONMENT=gnome
SUPPORT_DDC_AS_CNAME=N
VDA_PORT=80
REGISTER_SERVICE=Y
ADD_FIREWALL_RULES=Y
HDX_3D_PRO=N
VDI_MODE=N
SITE_NAME=' <none>'
LDAP_LIST=' <none>'
SEARCH_BASE=' <none>'
START_SERVICE=Y
```

- ドメイン参加済みシナリオの場合

- **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/sss.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 の構成ファイルと同じファイルになります。ただし、**dns**変数と**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、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。現在実行中の VDA をテンプレート仮想マシン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、値を 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。特に指定のない限り、これはドメインコントローラーの IP アドレスです。
- **WORKGROUP**: ワークグループ名を、AD で構成した NetBIOS 名 (大文字と小文字を区別) に設定します。設定しなかった場合、MCS はマシンのホスト名の直後に続くドメイン名の部分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、ワークグループ名として **citrix** が正しい選択であるにもかかわらず、MCS は **lvda** を使用することになります。ワークグループ名を正しく設定するようにしてください。

- **AD\_INTEGRATION**: Winbind、SSSD、PBIS、または Centrify を設定します。Linux ディストリビューションのマトリックスと MSC がサポートするドメイン参加方法については、「[サポートされているディストリビューション](#)」を参照してください。
- **CENTRIFY\_DOWNLOAD\_PATH**: Server Suite Free (旧称 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の更新を自動化](#)」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

```

手順 **4e**: マスターイメージを作成します

1. (SSSD + RHEL 8.x/9.x または Rocky Linux 8.x/9.x のみの場合) `update-crypto-policies --set DEFAULT:AD-SUPPORT` コマンドを実行してテンプレート仮想マシンを再起動します。
2. `/opt/Citrix/VDA/sbin/deploymcs.sh` を実行します。

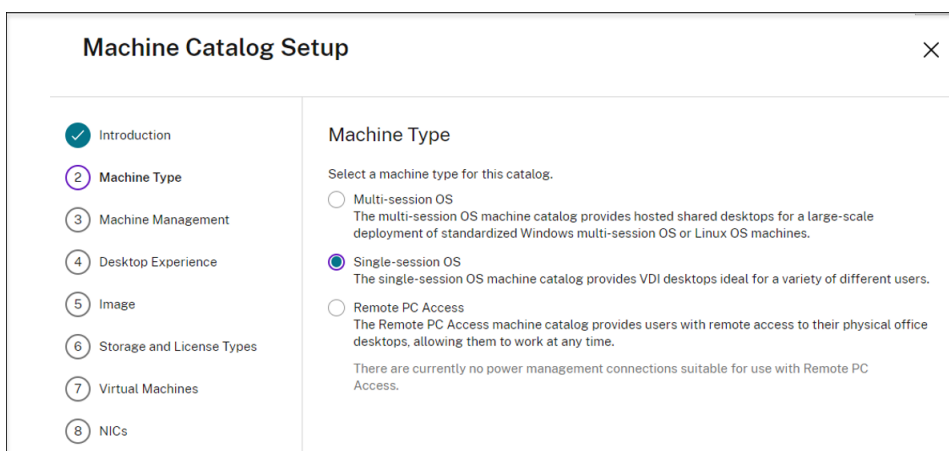
ドメイン非参加のシナリオの場合、次のエラーは正常であり、プロセスの続行が妨げられることはありません。

```
[root@localhost ~]#
[root@localhost ~]# /opt/Citrix/VDA/sbin/deploymcs.sh
Installing Linux VDA dependency packages ...
Installing package redhat-lsb-core
Installing package nautilus
Installing package totem-nautilus
Installing package brasero-nautilus
Installing package pulseaudio
Installing package pulseaudio-module-x11
Installing package pulseaudio-gdm-hooks
Installing package pulseaudio-module-bluetooth
Installing package alsa-plugins-pulseaudio
Installing package pciutils
Installing package openssh
Installing package openssh-clients
Installing package java-11-openjdk
Installing package chrony
Installing package krb5-workstation
Installing package oddjob-mkhomedir
Starting PostgreSQL database ...
Installing package tdb-tools
Installing package ntfs-3g
/opt/Citrix/VDA/lib64/mcs ~
installing mcs systemd unit file: ad_join.service ...
~
/opt/Citrix/VDA/lib64/mcs ~
~
Installing winbind dependency packages ...
Installing package samba-winbind
Installing package samba-winbind-clients
ERROR: Exit funtion conf_dns, dns not configured, please do it manually.
[root@localhost ~]#
```

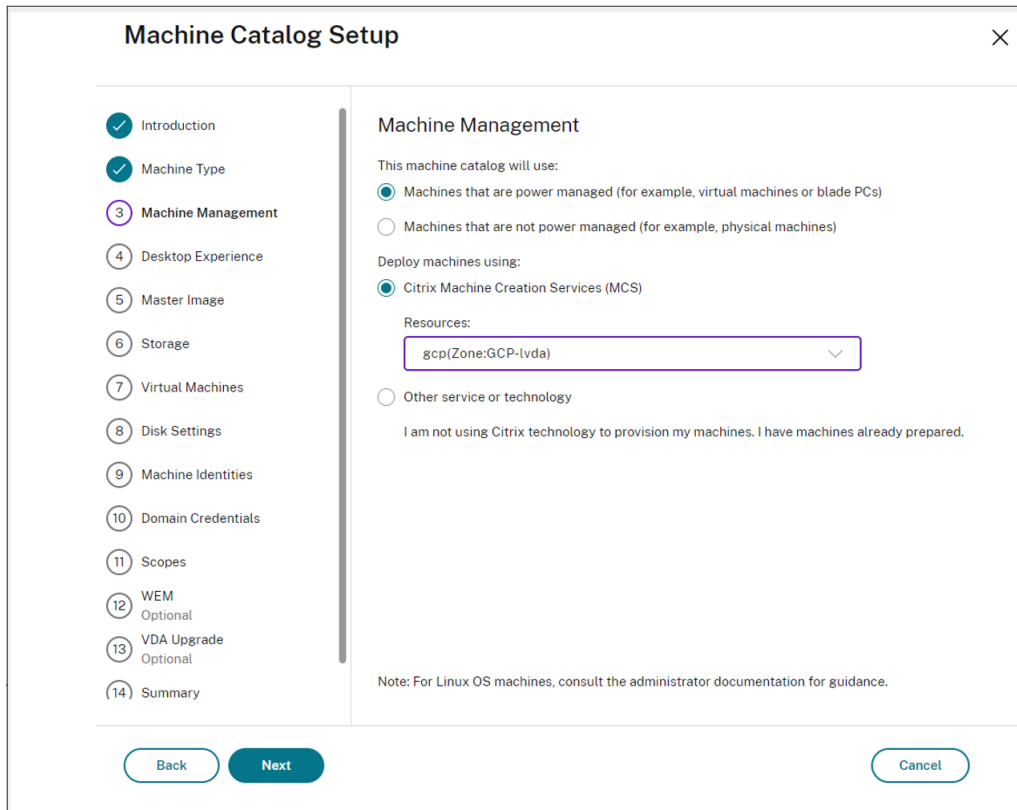
3. テンプレート仮想マシンにアプリケーションをインストールし、テンプレート仮想マシンをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。

#### 手順 5: マシンカタログを作成する

1. Citrix Cloudにサインインします。左上のメニューで、[マイサービス] > [DaaS] を選択します。
2. [管理] > [完全な構成] の左側ペインで [マシンカタログ] を選択します。
3. 操作バーで [マシンカタログの作成] を選択します。
4. [マシンの種類] ページで、[マルチセッション OS] または [シングルセッション OS] を選択し、[次へ] を選択します。



5. [マシン管理] ページで、[電源管理されているマシン] および [Citrix Machine Creation Services] オプションを選択してから [次へ] を選択します。複数のリソースがある場合は、メニューから 1 つ選択してください。



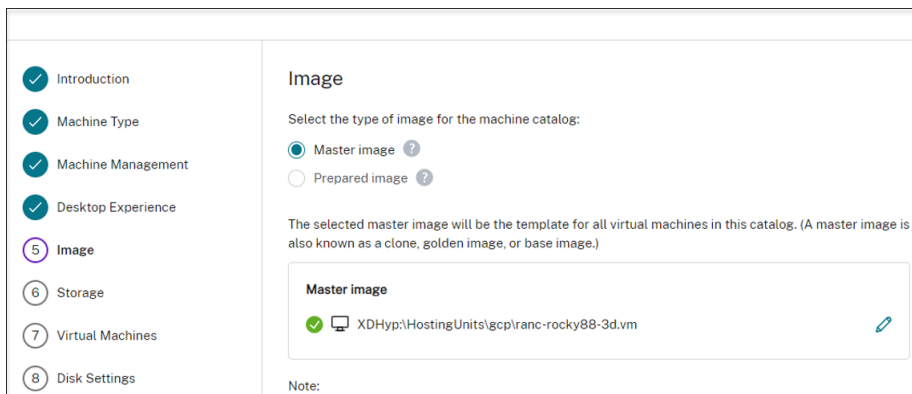
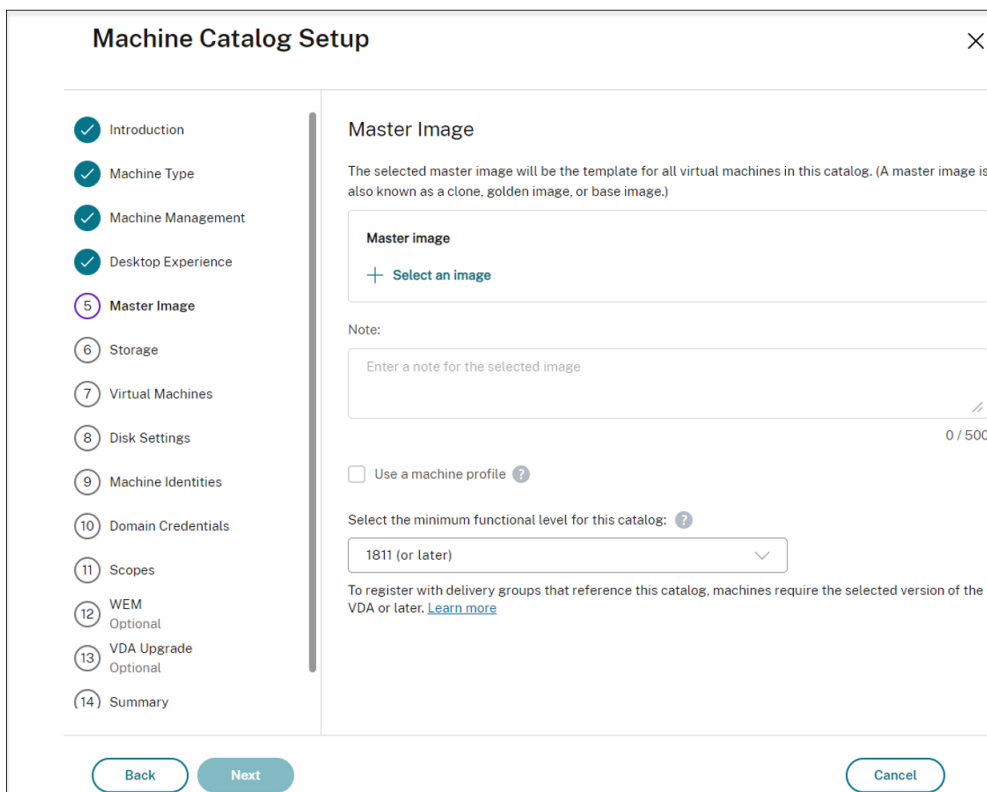
The screenshot shows the 'Machine Catalog Setup' dialog box. On the left is a vertical list of steps: Introduction, Machine Type, Machine Management (highlighted with a purple circle), Desktop Experience, Master Image, Storage, Virtual Machines, Disk Settings, Machine Identities, Domain Credentials, Scopes, WEM Optional, VDA Upgrade Optional, and Summary. The main area is titled 'Machine Management' and contains the following options:

- This machine catalog will use:
  - Machines that are power managed (for example, virtual machines or blade PCs)
  - Machines that are not power managed (for example, physical machines)
- Deploy machines using:
  - Citrix Machine Creation Services (MCS)
  - Other service or technology
- Resources:
  - A dropdown menu with the value 'gcp(Zone:GCP-lvda)' selected.
  - Other service or technology
- I am not using Citrix technology to provision my machines. I have machines already prepared.

Note: For Linux OS machines, consult the administrator documentation for guidance.

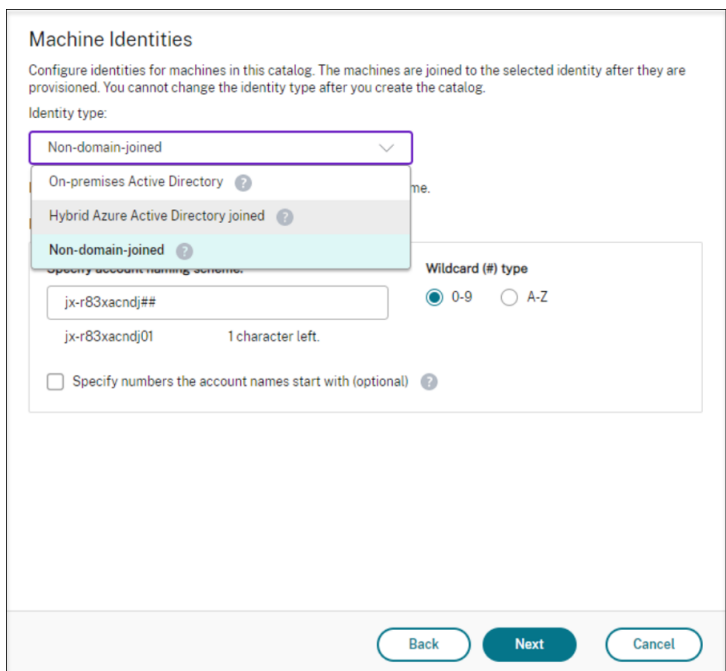
At the bottom, there are three buttons: 'Back', 'Next' (highlighted in dark blue), and 'Cancel'.

6. [マスターイメージ] ページで、上記で作成したマスターイメージを選択します。

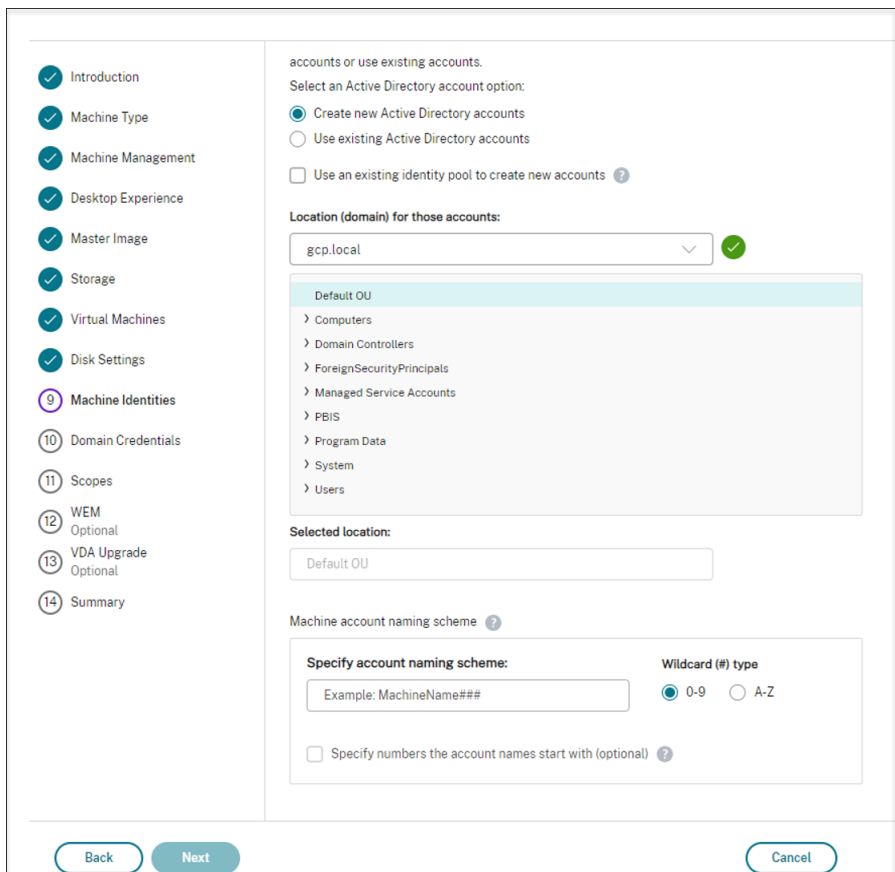


7. [マシン ID] ページで、マスターイメージがドメインに参加していない場合は [ドメイン非参加] を選択します。マスターイメージをドメインに参加させている場合は Active Directory アカウントを選択します。

ドメイン非参加シナリオの場合:



ドメイン参加済みシナリオの場合：



- [新しい **Active Directory** アカウントを作成する] を選択する場合、ドメインを選択してから Active Directory で作成されたプロビジョニング済みの VM コンピューターアカウントで名前付けスキームに

対応した文字列を入力します。アカウント名前付けスキームに指定できる文字数は 1~64 文字であり、空白スペース、非 ASCII 文字、および特殊文字を含めることはできません。

- [既存の **Active Directory** アカウントを使用する] を選択した場合、[参照] を選択し、選択したマシンの既存の Active Directory コンピューターアカウントに移動します。
- [ドメイン資格情報] ページで、[資格情報の入力] を選択し、ユーザー名とパスワードを入力し、[保存] を選択してから [次へ] を選択します。入力する資格情報には、Active Directory アカウント操作を実行する権限が必要です。

8. 他のページで追加の設定を構成します。詳しくは、「[Google Cloud Platform カタログの作成](#)」を参照してください。

9. [概要] ページで、情報を確認し、カタログの名前を指定してから、[完了] を選択します。

**Machine Catalog Setup**

Introduction  
Machine Type  
Machine Management  
Desktop Experience  
Master Image  
Storage  
Virtual Machines  
Disk Settings  
Machine Identities  
Domain Credentials  
Scopes  
WEM  
Optional  
VDA Upgrade  
Optional  
14 Summary

**Summary**

Machine type:	Single-session OS
Machine management:	Virtual
Provisioning method:	Machine creation services (MCS)
Desktop experience:	Users connect to the same desktop each time they log on
	Save changes on the local disk
Resources:	gcp
Master image:	rocky88-3d
Storage:	Standard persistent disk
VDA version:	1811 (or later)
Number of VMs to create:	1
Virtual CPUs:	2
Memory (MB):	7680
Hard disk (GB):	40
Available zones:	asia-southeast1-a, asia-southeast1-b, asia-southeast1-c
Identity type:	On-premises AD
Computer accounts:	Create new accounts
New accounts location:	gcp.local (Domain)

Machine catalog name:  
Example: Windows 7 SP1 Sales -2GB

Machine catalog description for administrators (optional):  
Example: Windows 7 SP1 desktops for the London Sales office

To complete the deployment, assign this machine catalog to a delivery group by selecting delivery groups and then create or edit a delivery group.

マシンカタログの作成が完了するまでに時間がかかる場合があります。完了すると、カタログが一覧表示されます。Google Cloud コンソールで、ターゲットノードグループにマシンが作成されていることを確認できます。

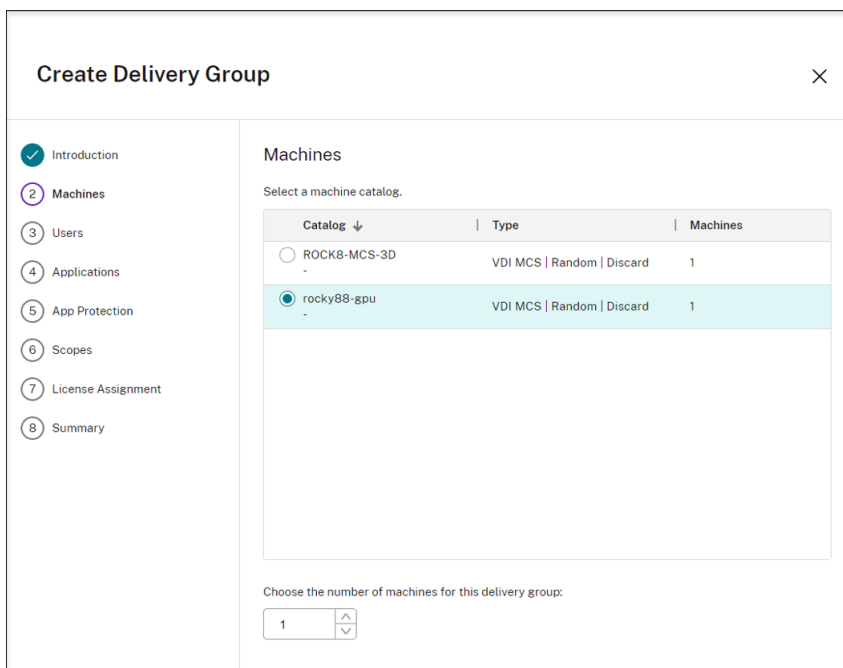
## 手順 6: デリバリーグループを作成する

デリバリーグループは、いくつかのマシンカタログから選択したマシンをグループ化したものです。それらのマシンを使用できるユーザーと、そのユーザーに提供するアプリケーションおよびデスクトップを指定します。

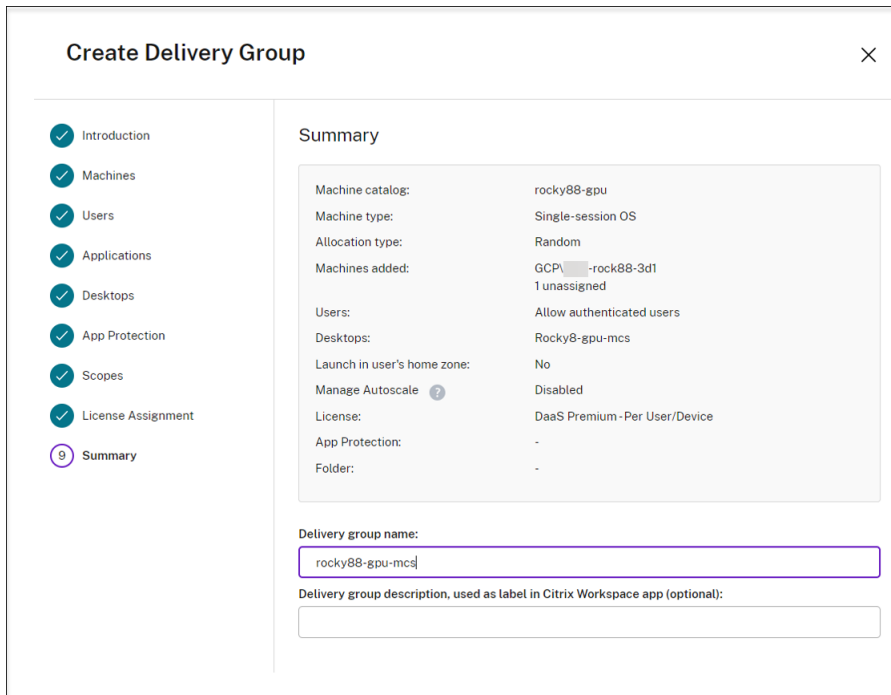
詳しくは、[Citrix DaaS](#)ドキュメントの「デリバリーグループの作成」を参照してください。



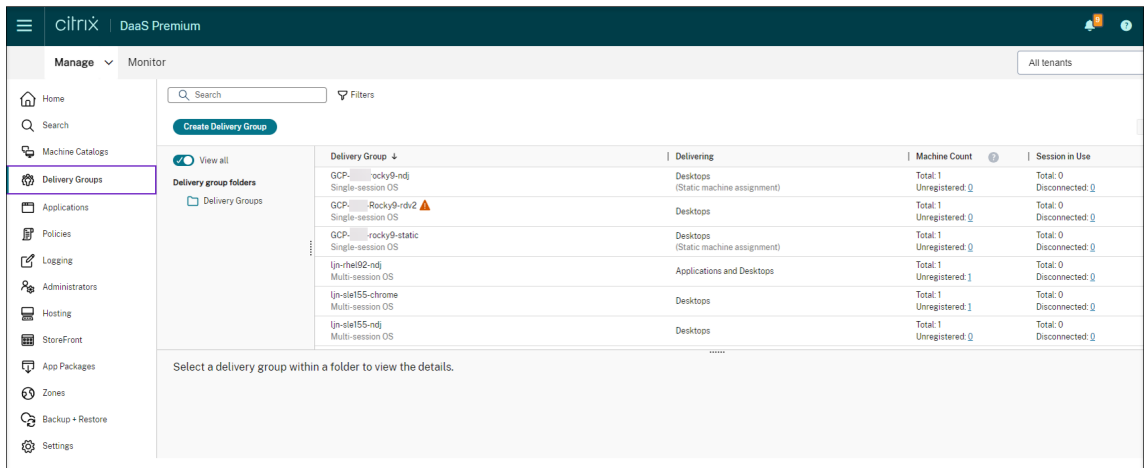
1. Citrix Cloudにサインインします。左上のメニューで、[マイサービス] > [DaaS] を選択します。
2. [管理] > [完全な構成] の左側ペインで [デリバリーグループ] を選択します。
3. 操作バーで [デリバリーグループの作成] を選択します。デリバリーグループ作成ウィザードが開きます。  
選択内容によっては、異なるウィザードページが表示されることがあります。
4. [マシン] ページでマシンカタログを選択して、そのカタログから使用するマシンの番号を選択します。



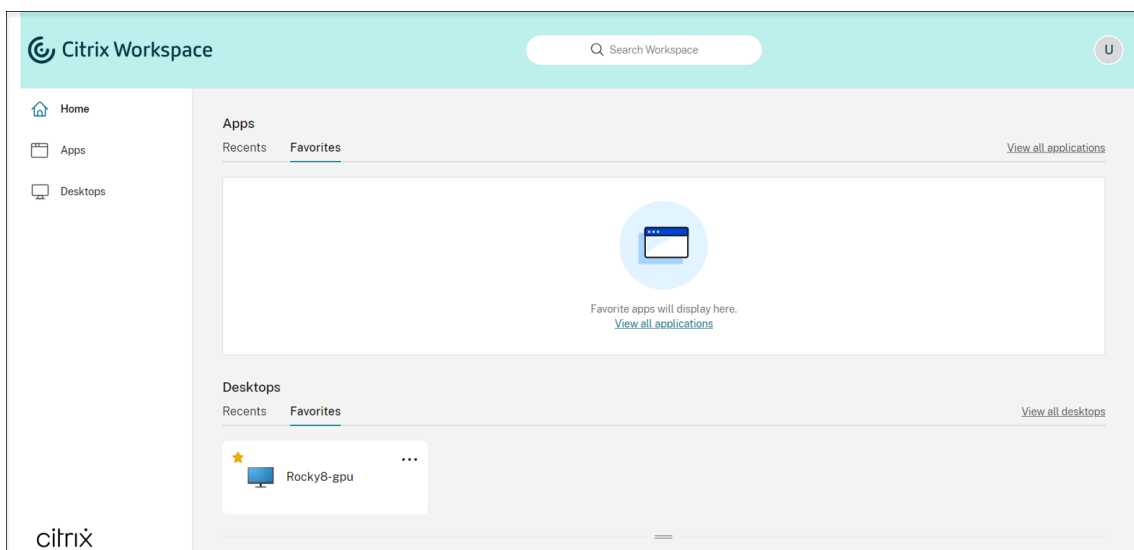
5. 他のページで追加の設定を構成します。詳しくは、「[デリバリーグループの作成](#)」を参照してください。
6. [概要] ページでデリバリーグループの名前を入力します。オプションで、Citrix Workspace アプリと [完全な構成] 管理インターフェイスに表示される説明を入力することもできます。例：  
デリバリーグループの名前を入力します：



[完全な構成] 管理インターフェイスでデリバリーグループの一覧を表示します:



Citrix Workspace アプリで提供されたマシン



## Ansible を使用した展開の管理

August 20, 2024

Ansible は、展開全体にわたってアプリケーション、構成、および更新を展開するプロセスを自動化するのに役立ちます。この記事では、Ansible を使用して展開を効率的に管理する手順を段階的に説明します。

### 手順 1: 何を展開するかを決定する

開始する前に、アプリケーション、サービス、構成、環境変数など、何を展開する必要があるかを特定します。

### 手順 2: Ansible プロジェクトをセットアップする

Ansible プロジェクトのディレクトリ構造を作成します。プレイブックのコンテンツを整理する重要な方法の 1 つは、Ansible の「roles」機能です。詳しくは、Ansible のドキュメントで「Roles」を参照してください。

以下に、参考までにディレクトリ構造の例を 2 つ示します：

#### ディレクトリ構造の例 #1

```
1 production # inventory file for production servers
2 staging # inventory file for staging environment
3
4 group_vars/
```

```
5     group1.yml           # here we assign variables to particular
      groups
6     group2.yml
7  host_vars/
8     hostname1.yml       # here we assign variables to particular
      systems
9     hostname2.yml
10
11  library/               # if any custom modules, put them here (
      optional)
12  module_utils/         # if any custom module_utils to support
      modules, put them here (optional)
13  filter_plugins/       # if any custom filter plugins, put them here
      (optional)
14
15  site.yml               # master playbook
16  webservers.yml        # playbook for webserver tier
17  dbservers.yml         # playbook for dbserver tier
18
19  roles/
20     common/             # this hierarchy represents a "role"
21         tasks/         #
22             main.yml   # <-- tasks file can include smaller files
                        # if warranted
23         handlers/     #
24             main.yml   # <-- handlers file
25         templates/    # <-- files for use with the template
            resource
26             ntp.conf.j2 # <----- templates end in .j2
27         files/        #
28             bar.txt    # <-- files for use with the copy resource
29             foo.sh     # <-- script files for use with the script
                        resource
30         vars/          #
31             main.yml   # <-- variables associated with this role
32         defaults/     #
33             main.yml   # <-- default lower priority variables for
                        this role
34         meta/         #
35             main.yml   # <-- role dependencies
36         library/       # roles can also include custom modules
37         module_utils/  # roles can also include custom module_utils
38         lookup_plugins/ # or other types of plugins, like lookup in
                        this case
39
40     webtier/           # same kind of structure as "common" was
                        above, done for the webtier role
41     monitoring/       # ""
42     fooapp/           # ""
```

## ディレクトリ構造の例 #2

```
1 inventories/
2   production/
3     hosts           # inventory file for production servers
4     group_vars/
5       group1.yml   # here we assign variables to particular
6         groups
7       group2.yml
8     host_vars/
9       hostname1.yml # here we assign variables to particular
10        systems
11      hostname2.yml
12
13 staging/
14   hosts           # inventory file for staging environment
15   group_vars/
16     group1.yml   # here we assign variables to particular
17       groups
18     group2.yml
19   host_vars/
20     stagehost1.yml # here we assign variables to particular
21       systems
22     stagehost2.yml
23
24 library/
25 module_utils/
26 filter_plugins/
27
28 site.yml
29 webservers.yml
30 dbservers.yml
31
32 roles/
33   common/
34   webtier/
35   monitoring/
36   fooapp/
```

### 手順 3: インベントリを構成する

インベントリファイル (inventory.ini) を定義します。インベントリファイルには、通常、Ansible を使用して管理するホストと、ホスト名、IP アドレス、グループメンバーシップなどの必要な詳細が一覧表示されます。例:

```
1 # Hostname and ip address
2 [UBUNTU2004]
3 <ip address>
4 [UBUNTU2204]
5 <ip address>
6 [RHEL8]
7 <ip address>
8 [RHEL9]
```

```
9 <ip address>
10 [DEBIAN11]
11 <ip address>
12 [DEBIAN12]
13 <ip address>
14 [SUSE15]
15 <ip address>
16
17 [all:children]
18 UBUNTU2004
19 UBUNTU2204
20 RHEL8
21 RHEL9
22 DEBIAN11
23 DEBIAN12
24 SUSE15
25
26 [all:vars]
27 ansible_user=<ansible execute user e.g root>
28 ansible_password=<>
29 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

#### 手順 4: Ansible プレイブックを作成する

プレイブック (.yaml ファイル) を作成して、展開タスクを自動化します。このセクションでは、さまざまな展開タスクを自動化するためのプレイブック例を示します。

#### Linux ディストリビューションにパッチを適用するためのプレイブック例

Ansible を使用してさまざまな Linux ディストリビューションにパッチを適用するには、`patch-for-different-distribution.yaml` という名前の YAML プレイブックファイルを作成し、次のようなタスクをファイルに入力します。hosts ディレクティブは、プレイブックタスクを実行する対象となるインベントリ（このコンテキストでは `inventory.ini`）からのターゲットホストを定義します。

```
1 - name: Upgrade and Reboot RHEL & Debian family Linux distros
2   hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
3   vars:
4     reboot_connect_timeout: 5
5     reboot_post_reboot_delay: 15
6     reboot_timeout: 600
7   tasks:
8     # Upgrade RHEL family OS packages
9     - name: Upgrade RHEL Family OS packages
10      ansible.builtin.yum:
11        name: '*'
12        state: latest
13      when:
```

```
14     - ansible_facts['distribution'] == "RedHat"
15     - ansible_facts['distribution_major_version'] == "7"
16
17     # Upgrade RHEL family OS packages
18     - name: Upgrade RHEL Family OS packages
19       ansible.builtin.yum:
20         name: '*'
21         state: latest
22     when:
23       - ansible_facts['distribution'] == "RedHat"
24       - ansible_facts['distribution_major_version'] == "8"
25
26     # Upgrade RHEL family OS packages
27     - name: Upgrade RHEL Family OS packages
28       ansible.builtin.yum:
29         name: '*'
30         state: latest
31     when:
32       - ansible_facts['distribution'] == "RedHat"
33       - ansible_facts['distribution_major_version'] == "9"
34
35     # Ubuntu Family upgrade
36     - name: Update repositories cache
37       apt:
38         update_cache: yes
39     when:
40       - ansible_facts['distribution'] == "Ubuntu"
41       - ansible_facts['distribution_major_version'] == "20"
42
43     - name: Update all packages to their latest version
44       apt:
45         name: "*"
46         state: latest
47     when:
48       - ansible_facts['distribution'] == "Ubuntu"
49       - ansible_facts['distribution_major_version'] == "22"
50
51     # Debian Family upgrade
52     - name: Upgrade the OS (apt-get dist-upgrade)
53       apt:
54         upgrade: dist
55     when:
56       - ansible_facts['distribution'] == "Debian"
57       - ansible_facts['distribution_major_version'] == "11"
58
59     - name: Upgrade the OS (apt-get dist-upgrade)
60       apt:
61         upgrade: dist
62     when:
63       - ansible_facts['distribution'] == "Debian"
64       - ansible_facts['distribution_major_version'] == "12"
65
66     # Reboot after upgrade
```

```
67     - name: Reboot host
68       reboot:
69         connect_timeout: ""
70         post_reboot_delay: ""
71         reboot_timeout: ""
```

### .Net 環境をインストールするためのプレイブック例

次のプレイブック例では、特定の Linux ディストリビューションに異なるバージョンの .Net 環境をインストールします。

```
1  - name: Install dotnet runtime environment on Linux distros
2  hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
3  tasks:
4  # Install dotnet runtime environment on RHEL7
5  - name: Enable the rhel-7-server-dotnet-rpms repository
6    command: subscription-manager repos --enable=rhel-7-server-dotnet
   -rpms
7    when:
8      - ansible_facts['distribution'] == "RedHat"
9      - ansible_facts['distribution_major_version'] == "7"
10
11 - name: Install dotnet runtime environment on RHEL7
12   ansible.builtin.yum:
13     name: rh-dotnet60-aspnetcore-runtime-6.0
14     state: present
15   when:
16     - ansible_facts['distribution'] == "RedHat"
17     - ansible_facts['distribution_major_version'] == "7"
18
19 - name: Remove /usr/bin/dotnet if it exists
20   file:
21     path: /usr/bin/dotnet
22     state: absent
23
24 - name: Create a symbolic link
25   file:
26     src: /opt/rh/rh-dotnet60/root/usr/lib64/dotnet/dotnet
27     dest: /usr/bin/dotnet
28     state: link
29
30 # RHEL8 linux vda install dotnet runtime environment
31 - name: Install dotnet-runtime-8.0
32   ansible.builtin.dnf:
33     name: dotnet-runtime-8.0
34     state: present
35   when:
36     - ansible_facts['distribution'] == "RedHat"
37     - ansible_facts['distribution_major_version'] == "8"
38
```



```
39 - name: Install aspnetcore-runtime-8.0
40   ansible.builtin.dnf:
41     name: aspnetcore-runtime-8.0
42     state: present
43   when:
44     - ansible_facts['distribution'] == "RedHat"
45     - ansible_facts['distribution_major_version'] == "8"
46
47 # RHEL9 linux vda install dotnet runtime environment
48 - name: Install dotnet-runtime-8.0
49   ansible.builtin.dnf:
50     name: dotnet-runtime-8.0
51     state: present
52   when:
53     - ansible_facts['distribution'] == "RedHat"
54     - ansible_facts['distribution_major_version'] == "9"
55
56 - name: Install aspnetcore-runtime-8.0
57   ansible.builtin.dnf:
58     name: aspnetcore-runtime-8.0
59     state: present
60   when:
61     - ansible_facts['distribution'] == "RedHat"
62     - ansible_facts['distribution_major_version'] == "9"
63
64 # Ubuntu20.04 linux vda install dotnet runtime environment
65 - name: Register Microsoft key and feed
66   shell: |
67     wget https://packages.microsoft.com/config/ubuntu/20.04/
68           packages-microsoft-prod.deb -O packages-microsoft-prod.deb
69     dpkg -i packages-microsoft-prod.deb
70     rm packages-microsoft-prod.deb
71   when:
72     - ansible_facts['distribution'] == "Ubuntu"
73     - ansible_facts['distribution_major_version'] == "20"
74
75 - name: Install dotnet-runtime-8.0
76   ansible.builtin.apt:
77     name: dotnet-runtime-8.0
78     state: present
79     update_cache: yes
80   when:
81     - ansible_facts['distribution'] == "Ubuntu"
82     - ansible_facts['distribution_major_version'] == "20"
83
84 - name: Install aspnetcore-runtime-8.0
85   ansible.builtin.apt:
86     name: aspnetcore-runtime-8.0
87     state: present
88     update_cache: yes
89   when:
90     - ansible_facts['distribution'] == "Ubuntu"
91     - ansible_facts['distribution_major_version'] == "20"
```

```
91
92 # Ubuntu22.04 linux vda install dotnet runtime environment
93 - name: Install dotnet-runtime-8.0
94   ansible.builtin.apt:
95     name: dotnet-runtime-8.0
96     state: present
97     update_cache: yes
98   when:
99     - ansible_facts['distribution'] == "Ubuntu"
100     - ansible_facts['distribution_major_version'] == "22"
101
102 - name: Install aspnetcore-runtime-8.0
103   ansible.builtin.apt:
104     name: aspnetcore-runtime-8.0
105     state: present
106     update_cache: yes
107   when:
108     - ansible_facts['distribution'] == "Ubuntu"
109     - ansible_facts['distribution_major_version'] == "22"
110
111 # Debian11 linux vda install dotnet runtime environment
112 - name: Register Microsoft key and feed
113   shell: |
114     wget https://packages.microsoft.com/config/debian/11/packages-
115           microsoft-prod.deb -O packages-microsoft-prod.deb
116     dpkg -i packages-microsoft-prod.deb
117     rm packages-microsoft-prod.deb
118   when:
119     - ansible_facts['distribution'] == "Debian"
120     - ansible_facts['distribution_major_version'] == "11"
121
122 - name: Install dotnet-runtime-8.0
123   ansible.builtin.apt:
124     name: dotnet-runtime-8.0
125     state: present
126     update_cache: yes
127   when:
128     - ansible_facts['distribution'] == "Debian"
129     - ansible_facts['distribution_major_version'] == "11"
130
131 - name: Install aspnetcore-runtime-8.0
132   ansible.builtin.apt:
133     name: aspnetcore-runtime-8.0
134     state: present
135     update_cache: yes
136   when:
137     - ansible_facts['distribution'] == "Debian"
138     - ansible_facts['distribution_major_version'] == "11"
139
140 # Debian12 linux vda install dotnet runtime environment
141 - name: Register Microsoft key and feed
142   shell: |
143     wget https://packages.microsoft.com/config/debian/12/packages-
```

```
        microsoft-prod.deb -0 packages-microsoft-prod.deb
143     dpkg -i packages-microsoft-prod.deb
144     rm packages-microsoft-prod.deb
145     when:
146     - ansible_facts['distribution'] == "Debian"
147     - ansible_facts['distribution_major_version'] == "12"
148
149 - name: Install dotnet-runtime-8.0
150   ansible.builtin.apt:
151     name: dotnet-runtime-8.0
152     state: present
153     update_cache: yes
154   when:
155   - ansible_facts['distribution'] == "Debian"
156   - ansible_facts['distribution_major_version'] == "12"
157
158 - name: Install aspnetcore-runtime-8.0
159   ansible.builtin.apt:
160     name: aspnetcore-runtime-8.0
161     state: present
162     update_cache: yes
163   when:
164   - ansible_facts['distribution'] == "Debian"
165   - ansible_facts['distribution_major_version'] == "12"
166
167 # Sles15 linux vda install dotnet runtime environment
168 - name: Register Microsoft key and feed
169   shell: |
170     sudo rpm -Uvh https://packages.microsoft.com/config/sles/15/
171     sudo ln -s /etc/yum.repos.d/microsoft-prod.repo /etc/zypp/repos
172     .d/microsoft-prod.repo
173   when:
174   - ansible_facts['distribution'] == "SLES"
175   - ansible_facts['distribution_major_version'] == "15"
176
177 - name: Install dotnet-runtime-8.0
178   community.general.zypper:
179     name: dotnet-runtime-8.0
180     state: present
181     update_cache: yes
182   when:
183   - ansible_facts['distribution'] == "SLES"
184   - ansible_facts['distribution_major_version'] == "15"
185
186 - name: Install aspnetcore-runtime-8.0
187   community.general.zypper:
188     name: aspnetcore-runtime-8.0
189     state: present
190     update_cache: yes
191   when:
192   - ansible_facts['distribution'] == "SLES"
193   - ansible_facts['distribution_major_version'] == "15"
```

```
193
194 # Amazon2 linux vda install dotnet runtime environment
195 - name: Install dotnet-runtime-8.0
196   ansible.builtin.yum:
197     name: dotnet-runtime-8.0
198     state: present
199   when:
200     - ansible_facts['distribution'] == "Amazon"
201     - ansible_facts['distribution_major_version'] == "2"
202
203 - name: Install aspnetcore-runtime-8.0
204   ansible.builtin.yum:
205     name: aspnetcore-runtime-8.0
206     state: present
207   when:
208     - ansible_facts['distribution'] == "Amazon"
209     - ansible_facts['distribution_major_version'] == "2"
```

### Linux VDA をアップグレードするためのプレイブック例

Ansible を使用して Linux VDA のアップグレードを自動化するには、2つの個別のプレイブックを作成します。1つのプレイブック、たとえば `get_the_build.yml` は、Linux VDA パッケージをターゲットマシン（ホスト）にダウンロードして転送するため専用です。他のプレイブック、たとえば `linux_upgrade.yml` には、以前にダウンロードしたパッケージを使用して、ターゲットマシン上の Linux VDA をアップグレードするように設計されたタスクが含まれています。

#### プレイブック例 `get_the_build.yml`

```
1 - hosts: localhost
2   name: Get the latest release build to local
3   vars:
4     build_url: <linux vda download link> # replace with your actual
5     local_tmp: "/tmp/" # replace with your actual value.
6     remote_tmp: "/tmp/" # replace with your actual value.
7     linuxvda_file_name : "linux vda rpm/deb file name" # replace with
8     your actual value.
9   tasks:
10  - name: Download the file
11    get_url:
12      url: ""
13      dest: ""
14    tags:
15      - get
16
17 - hosts: <host1,host2,host3> # replace with your actual hosts in the
18   name: Copy a file to remote location
19   tasks:
```

```
20 - name: Copy vda to the remote machine
21   ansible.builtin.copy:
22     src: ""
23     dest: ""
24     remote_src: no
25   tags:
26     - copy
```

#### プレイブック例 **linux\_upgrade.yml**

```
1 - name: Upgrade Linux VDA and Reboot RHEL & Debian Linux distros
2   hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
3   vars:
4     remote_tmp: "/path/to/remote/tmp" # replace with your actual path
5     rhel7_file_name: "rhel7_file.rpm" # replace with your actual file
   name
6     rhel8_file_name: "rhel8_file.rpm" # replace with your actual file
   name
7     rhel9_file_name: "rhel9_file.rpm" # replace with your actual file
   name
8     ubuntu2004_file_name: "ubuntu2004_file.deb" # replace with your
   actual file name
9     ubuntu2204_file_name: "ubuntu2204_file.deb" # replace with your
   actual file name
10    debian11_file_name: "debian11_file.deb" # replace with your actual
   file name
11    debian12_file_name: "debian12_file.deb" # replace with your actual
   file name
12    suse15_file_name: "suse15_file.deb" # replace with your actual file
   name
13    amazon2_file_name: "amazon2_file.rpm" # replace with your actual
   file name
14   tasks:
15     # Upgrade RHEL linux vda packages
16     - name: Upgrade RHEL7 linux vda packages
17       ansible.builtin.yum:
18         name: ""
19         state: present
20       when:
21         - ansible_facts['distribution'] == "RedHat"
22         - ansible_facts['distribution_major_version'] == "7"
23
24     # Upgrade RHEL linux vda packages
25     - name: Upgrade RHEL8 linux vda packages
26       ansible.builtin.yum:
27         name: ""
28         state: present
29       when:
30         - ansible_facts['distribution'] == "RedHat"
31         - ansible_facts['distribution_major_version'] == "8"
32
33     # Upgrade RHEL linux vda packages
34     - name: Upgrade RHEL9 linux vda packages
```

```
35     ansible.builtin.yum:
36         name: ""
37         state: present
38     when:
39         - ansible_facts['distribution'] == "RedHat"
40         - ansible_facts['distribution_major_version'] == "9"
41
42     # Ubuntu20.04 linux vda upgrade
43     - name: Ubuntu20.04 linux vda upgrade
44       ansible.builtin.apt:
45         deb: ""
46     when:
47         - ansible_facts['distribution'] == "Ubuntu"
48         - ansible_facts['distribution_major_version'] == "20"
49
50     - name: Ubuntu22.04 linux vda upgrade
51       ansible.builtin.apt:
52         deb: ""
53     when:
54         - ansible_facts['distribution'] == "Ubuntu"
55         - ansible_facts['distribution_major_version'] == "22"
56
57     # Debian Linux VDA upgrade
58     - name: Debian11 Linux VDA upgrade
59       ansible.builtin.apt:
60         deb: ""
61     when:
62         - ansible_facts['distribution'] == "Debian"
63         - ansible_facts['distribution_major_version'] == "11"
64
65     - name: Debian12 Linux VDA upgrade
66       ansible.builtin.apt:
67         deb: ""
68     when:
69         - ansible_facts['distribution'] == "Debian"
70         - ansible_facts['distribution_major_version'] == "12"
71
72     # Sles15 Linux VDA upgrade
73     - name: Sles15 Linux VDA upgrade
74       community.general.zypper:
75         name: ""
76         state: present
77     when:
78         - ansible_facts['distribution'] == "SLES"
79         - ansible_facts['distribution_major_version'] == "15"
80
81     # Amazon2 Linux VDA upgrade
82     - name: Amazon2 Linux VDA upgrade
83       ansible.builtin.yum:
84         name: ""
85     when:
86         - ansible_facts['distribution'] == "Amazon"
87         - ansible_facts['distribution_major_version'] == "2"
```

```
88     # Reboot after upgrade
89     - name: Reboot host
90       reboot:
91         connect_timeout: ""
92         post_reboot_delay: ""
93         reboot_timeout: ""
```

ネットワークファイルシステム（**NFS**）サーバーをホームディレクトリとしてマウントするためのプレイブック例

次のプレイブック例では、NFS サーバーをターゲットホストのホームディレクトリとしてマウントします：

```
1  - hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
2  vars:
3    nfs_server = <nfsserver ip address> # replace with your actual
   values
4    mount_points = /home/<domain realm>/user1,/home/<domain realm>user2
   # replace with your actual values
5    nfs_shares = user1,user2 # replace with your actual values
6    owners = user1,user2 # replace with your actual values
7    groups = group1,group2 # replace with your actual values
8  tasks:
9    - name: Enable NFS as home directory
10     ansible.builtin.command:
11       cmd: "/opt/Citrix/VDA/bin/ctxreg create -k 'HKLM\System\
   CurrentControlSet\Control\Citrix' -t 'REG_DWORD' -v '
   CheckUserHomeMountPoint' -d '0x00000001' --force"
12     register: result
13     failed_when: result.rc != 0
14     check_mode: no
15
16    - name: Mount NFS shares
17     ansible.builtin.mount:
18       path: ""
19       src: ":"
20       fstype: nfs
21       opts: rw,noexec
22       state: mounted
23       loop: ""
24
25    - name: Set owner, group and mode for NFS client paths
26     ansible.builtin.file:
27       path: ""
28       owner: ""
29       group: ""
30       mode: ""
31     loop: ""
```

リモートコマンド実行のプレイブック例

#### レジストリ設定を変更するためのプレイブック例

```
1 - hosts: <host1,host2,host3> # replace with your actual hosts in the
  inventory file.
2 vars:
3   registry_key: "your_registry_key" # E.g. registry_key = HKLM\
  System\CurrentControlSet\Control\Terminal Server\Wds\icawd
4   registry_type: "your_registry_type" # E.g. registry_type =
  REG_DWORD
5   registry_value: "your_registry_value" # E.g. registry_value =
  AdaptiveScalingEnabled
6   registry_data: "your_registry_data" # E.g. registry_data = 0
  x00000000
7 tasks:
8 - name: Execute AdaptiveScaling redirection script
  ansible.builtin.command:
9   cmd: "/opt/Citrix/VDA/bin/ctxreg create -k "" -t "" -v "" -d ""
  --force"
10  register: result
11  failed_when: result.rc != 0
12  check_mode: no
13
```

#### RHEL マイナーバージョンをロックするためのプレイブック例

```
1 - hosts: <host1,host2,host3> # replace with your actual hosts in the
  inventory file.
2 vars:
3   rhel_minor_version: "9.3" # replace with your actual minor version
  such as 9.3, 8.8
4 tasks:
5 - name: Lock system to a specific minor version
  ansible.builtin.command:
6   cmd: "subscription-manager release --set="
7   register: result
8   failed_when: "'Error' in result.stderr"
9
```





© 2024 Cloud Software Group, Inc. All rights reserved. Cloud Software Group, the Cloud Software Group logo, and other marks appearing herein are property of Cloud Software Group, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner(s).