Linux仮想配信エージェント 2411

重要:

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

Linux Virtual Delivery Agent (VDA) は、Citrix Workspace™アプリがインストールされている任意のデバイスから、どこからでもLinux仮想アプリとデスクトップへのアクセスを可能にします。

サポートされているLinuxディストリビューションに基づいて、仮想アプリとデスクトップを提供できます。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を構成した後、Citrix Studioを使用してアプリとデスクトップをユーザーに提供します。

新機能

2411の新機能

セッションの安定性の向上

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

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

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

Ubuntu 24.04、Debian 12.7、Debian 11.11、および SUSE 15.6 のサポート

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

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

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

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

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

PostgreSQL バージョンサポートの強化

Linux VDA は、Linux ディストリビューションによって提供されるバージョンに関係なく、カスタム PostgreSQL バージョンの使用をサポートするようになりました。Linux ディストリビューションによって提供されるバージョンと少なくとも同程度の最新の PostgreSQL バージョンを使用することをお勧めします。カスタムバージョンの PostgreSQL を効果的に使用するには、新しいバージョンに合わせて `/etc/xdl/db.conf` が適切に構成されていることを確認してください。詳細については、インストール記事のデータベース指定手順（例: 「手順 7: 使用するデータベースを指定する」および「手順 1g: 使用するデータベースをインストールして指定する」）を参照してください。

ルートレス Xorg

Linux VDA は現在、非ルートユーザー権限で Xorg を実行することをサポートしています。これは「rootless」Xorg とも呼ばれます。Rootless Xorg は、root として実行する場合と比較して、セキュリティが大幅に向上しています。詳細については、「Rootless Xorg」を参照してください。

サービスアカウント「ctxsrvr」が新しい場所に移動されました

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

ファイルのコピーと貼り付け機能の強化

このリリースには、ファイル名における特殊文字のサポートを目的とした、ファイルのコピーと貼り付け機能の強化が含まれています。詳細については、「ファイルのコピーと貼り付け」を参照してください。

クロスプラットフォームキーボード機能の改善

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

Webカメラ機能の改善

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

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

Ubuntu 24.04でのPipeWireのサポート

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

複数のオーディオデバイスとオーディオのロス許容モードのサポートが一般提供開始されました

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

アダプティブオーディオ用オーディオ品質エンハンサー（プレビュー）

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

その他の自動クライアント再接続ポリシーのサポート

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

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

詳細なポリシーの説明については、「自動クライアント再接続ポリシー設定」を参照してください。

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

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

以前のリリースでの新機能

1912 **LTSR**以降2407 **CR**までのリリースに含まれる新機能については、「新機能の履歴」を参照してください。

修正された問題

Linux Virtual Delivery Agent 2411には、以下の修正された問題が含まれています。

- Linux VDA上の切断されたセッションに再接続すると、グレースクリーンになることがあり、ユーザーは回復のためにシステムを再起動するか、手動でログオフする必要があります。
- 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。
- GPFファイルから大量のグループポリシーセットを処理する際に、グループポリシーパーサーがCPUを100%消費する可能性があります。
- 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™インサイト情報を表示できなくなります。
- テンプレートマシンでのad_join.shの誤った実行により、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る可能性があります。この修正により、deploymcs.shスクリプトにチェックが導入され、フラグが検出された場合にそれを特定して削除します。
- 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アクセスをブロックします。この問題は、デフォルトでほとんどのLinuxディストリビューションが、各プロセスで開くことができる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を使い果たすと、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。

  スマートカードログオンが有効なVDAにこの問題が発生します。症状としては、**/var/log/xdl/hdx.log**に「**Failed to accept new connection: Too many open files**」というエラーが多数記録され、**/proc/\${pid}/fd/にファイル記述子が蓄積されます。ここで、\${pid}はctxscardsd**のプロセスIDを示します。PIDを特定するには、**systemctl status ctxscardsd\|grep PID**コマンドを実行します。

この問題を軽減するには、スマートカードサービスの最大オープンファイル制限を増やすか、スマートカードサービスを再起動することができます。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:

```
systemctl restart ctxscardsd
```

- 現在のサービスの最大オープンファイルを照会するには:

```
cat /proc/${PID}/limits
```

- スマートカードサービスの最大オープンファイルを設定するには:

  a. 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:

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

  b. 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:

  ```
  LimitNOFILE=65536
  ```

  c. **systemd**デーモンをリロードし、**ctxscardsd**サービスを再起動します:

  ```
  systemctl daemon-reload

  systemctl restart ctxscardsd
  ```

  d. 新しい制限を確認してください:

  ```
  cat /proc/${PID}/limits
  ```

注:

> 最大オープンファイル数を増やすことで、ファイルディスクリプタ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場合があります。

- オーディオのプラグアンドプレイはサポートされていません。そのため、ユーザーは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する必要があります。オーディオ録音アプリケーションの起動後にキャプチャデバイスを接続すると、アプリケーションが応答しなくなり、再起動が必要になる場合があります。同様に、録音中にキャプチャデバイスを抜くと、同じ問題が発生する可能性があります。この問題は、バージョン2411以降でデフォルトで有効になっている複数オーディオデバイスサポート機能の導入により解決されます。

既知の問題

このリリースでは、以下の問題が確認されています。

- 複数のDelivery Controllerが構成された環境で、オンプレミスハイパーバイザー上にドメインに参加していないVirtual Delivery Agent (VDA) をプロビジョニングするためにMachine Creation Services (MCS) を使用すると、`ctxsetup`プロセスが失敗する可能性があります。エラーは、ターゲットVDAの/var/log/ad_join.logファイルで確認できます。

  この問題を回避するには、以下の手順を実行します。

  a. マスターイメージで、`/var/xdl/mcs/mcs_util.sh`を見つけて編集します。

     a. マスターイメージでターミナルまたはコマンドプロンプトを開きます。
     b. `/var/xdl/mcs/`ディレクトリに移動します。
     c. テキストエディターを使用して`mcs_util.sh`ファイルを開きます。
     d. `read_websocket_ddc_info()`関数を見つけます。
     e. 297行目付近にある`log_debug "Exit read_websocket_ddc_info"`という行を見つけます。この行の前に`DDCS=$(echo "${DDCS}" | tr ',' ' ')`という行を挿入します。

  b. 新しいスナップショットを作成し、その新しいスナップショットでVDAをプロビジョニングします。

- Mesaグラフィックスライブラリ（特にバージョン24.1.0以降）は、Ubuntu 24.04仮想デスクトップで特定のシステムアプリケーション（ファイルマネージャー、設定、電卓など）の表示に関する問題を引き起こしています。以下の2つの回避策が提供されています。

回避策1：Rootless Xorgを無効にする

以下のコマンドを使用して、一時的な回避策としてRootless Xorgを無効にできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "0" --force



sudo systemctl restart ctxhdx
```

注:

Red Hatは、Xorgサーバーに影響を与えるCVE-2024-31083セキュリティ脆弱性に対処しました。

回避策2：Rootless Xorgの機能を継続するには、cap_ipc_owner機能を付与する

Rootless Xorg機能を維持し、ICA®セッションでMIT-SHMが機能するようにするには、Rootless Xorgを有効にする前に、ラッパースクリプトではなくXorg実行可能ファイルに直接`cap_ipc_owner`機能を適用します。以下のディストリビューション固有のコマンドを使用してください。

- Ubuntu/Debian: `sudo setcap 'cap_ipc_owner=+ep' /usr/lib/xorg/Xorg`
- RHEL/Rocky の問題: `sudo setcap 'cap_ipc_owner=+ep' /usr/libexec/Xorg`
- SUSE: `sudo setcap 'cap_ipc_owner=+ep' /usr/libexec/Xorg /usr/bin/Xorg`

- 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にアップグレードした後、Linux VDAが期待どおりに動作しません。詳細については、`<https://issues.redhat.com/browse/RHEL-17122>`を参照してください。
- PostgreSQLで設定された最大接続数が同時セッションを処理するのに不十分な場合、セッション起動の失敗が発生します。この問題を回避するには、**postgresql.conf**ファイルで**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例外がスローされるため、VDA登録が失敗する可能性があります。

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

この問題を回避するには、次の手順を実行します。

- 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秒に変更します。

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

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

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

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

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

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

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

  a. グループポリシー管理コンソールを開きます。

  b. ターゲットドメインを見つけて、**Default Domain Policy** を選択します。

c. **Default Domain Policy** を右クリックし、**Edit** を選択します。グループポリシー管理エディターが開きます。

d. **コンピューターの構成 > ポリシー > Windows の設定 > セキュリティの設定 > ローカル ポリシー > セキュリティ オプション** を選択します。

e. **Network security: Configure encryption types allowed for Kerberos** をダブルクリックします。

f. **DES_CBC_CRC**、**DES_CBC_MD5**、および **RC4_HMAC_MD5** のチェックボックスをオフにし、**AES128_HMAC_SHA1**、**AES256_HMAC_SHA1**、および **Future encryption types** を選択します。

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

> **注:**
>
> 弱い暗号化の種類を使用すると、展開が攻撃に対して脆弱になります。

a. Linux VDA で /etc/krb5.conf ファイルを開きます。

b. セクションの下に次のエントリを追加します。

```
allow_weak_crypto= TRUE
```

- Linux VDA は暗号化に SecureICA 1.0 をサポートしていません。Linux VDA で SecureICA 1.0 を有効にすると、セッションの起動に失敗します。
- Ubuntu グラフィック: HDX™ 3D Pro では、Desktop Viewer のサイズを変更した後、アプリケーションの周囲に黒いフレームが表示されたり、背景が黒く表示されたりする場合があります。
- Linux VDA の印刷リダイレクトによって作成されたプリンターは、セッションからログアウトした後も削除されない場合があります。
- ディレクトリに多数のファイルとサブディレクトリが含まれている場合、CDM ファイルがありません。この問題は、クライアント側にファイルまたはディレクトリが多すぎる場合に発生する可能性があります。
- 英語以外の言語では、UTF-8 エンコーディングのみがサポートされています。
- Citrix Workspace™ app for Android の CAPS LOCK の状態は、セッションローミング中に反転する場合があります。既存の接続を Citrix Workspace app for Android にローミングすると、CAPS LOCK の状態が失われる可能性があります。回避策として、拡張キーボードの Shift キーを使用して、大文字と小文字を切り替えます。

- Mac版Citrix Workspaceアプリを使用してLinux VDAに接続すると、ALTキーを含むショートカットキーが常に機能するとは限りません。Mac版Citrix Workspaceアプリは、デフォルトで左右のOptions/Altキーの両方にAltGrを送信します。この動作はCitrix Workspaceアプリの設定内で変更できますが、結果はアプリケーションによって異なります。
- Linux VDAがドメインに再参加すると、登録が失敗します。再参加により、新しいKerberosキーセットが生成されます。しかし、Brokerは以前のKerberosキーセットに基づいた、キャッシュされた古いVDAサービスチケットを使用する可能性があります。VDAがBrokerへの接続を試みると、BrokerはVDAへの戻りのセキュリティコンテキストを確立できない場合があります。一般的な症状は、VDA登録の失敗です。

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

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

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

  このコマンドは、Citrix Broker Serviceが実行されているNetwork Serviceプリンシパルによって保持されているLSAキャッシュ内のすべてのサービスチケットをパージします。これにより、他のVDAや、場合によっては他のサービスのサービスチケットも削除されます。ただし、これは無害です。これらのサービスチケットは、必要に応じてKDCから再取得できます。
- Windows版Citrix Workspaceアプリでは、オーディオ録音中に音声の歪みが発生する場合があります。

サードパーティ通知

Linux Virtual Delivery Agent バージョン 2411(/ja-jp/linux-virtual-delivery-agent/2411/downloads/lvda-v2411-third-party-notices.pdf) (PDFダウンロード)

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

廃止予定

この記事のお知らせは、段階的に廃止されるプラットフォーム、Citrix®製品、および機能について事前に通知するものであり、お客様がタイムリーなビジネス上の意思決定を行えるようにするためのものです。Citrixは、お客様の使用状況とフィードバックを監視して、それらがいつ廃止さ

れるかを判断します。お知らせは、その後のリリースで変更される可能性があり、すべての非推奨の機能や機能性を含まない場合があります。 製品ライフサイクルサポートの詳細については、製品ライフサイクルサポートポリシーの記事を参照してください。

廃止予定と削除

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

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

項目	非推奨が発表された時期	削除時期
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
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

項目	非推奨が発表された時期	削除時期
RHEL 6.8、CentOS 6.8 のサポート	1811	1811
RHEL 7.3、CentOS 7.3 のサポート	7.18	7.18
RHEL 6.6、CentOS 6.6 のサポート	7.16	7.16
SUSE 11.4	7.16	7.16

新機能の履歴

この記事では、1912 LTSRから2407 CRまでにリリースされた新機能に関する情報を提供します。

2407 の新機能

RHEL 9.4 / 8.10、Rocky Linux 9.4 / 8.10、およびDebian 12.5 / 11.9のサポート

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

- RHEL 9.4
- ロッキー Linux 9.4
- RHEL 8.10
- ロッキー Linux 8.10
- デビアン 12.5
- デビアン 11.9

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

新しい.NETランタイムが必要

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

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

オンプレミス環境へのトークンベースの登録の拡張（プレビュー）

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

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

詳細については、「簡易インストールを使用したドメイン非参加Linux VDAの作成（プレビュー）」を参照してください。

SSSDがデフォルトのドメイン参加方法として指定されました

SSSDは、Winbindに代わって、ドメイン参加のデフォルトの方法になりました。この変更は、ctxsetup.shまたは簡易インストールスクリプト (ctxinstall.sh) を対話モードで実行する際の構成動作に影響します。詳細については、「簡易インストールを使用したドメイン参加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を処理するパッケージマネージャーへの委任

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

スマートカード取り外し設定のサポート

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

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

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

オーディオのロス許容モード（プレビュー）

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

オーディオパフォーマンスを向上させるために実装されたパケット損失隠蔽（プレビュー）

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

この機能を有効にするには、オーディオの損失許容モード (プレビュー)機能も有効にする必要があります。詳細については、オーディオ機能の記事のオーディオパフォーマンスを向上させるためのパケット損失隠蔽 (プレビュー)を参照してください。

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

Linux VDAは、Debian 12.5、RHEL 9.x、およびRocky Linux 9.xでPipeWireをサポートするようになりました。これらのOSではPipeWireがデフォルトのオーディオサービスです。詳細については、オーディオ機能を参照してください。

グラフィックパフォーマンスの向上

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

ログインUIに表示されるログインプロセスの詳細

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

- **Citrix®グループポリシー設定の評価**
- **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デスクトップ環境でのキーボードレイアウト同期のサポート

このリリースでは、動的キーボードレイアウト同期機能がXFCEおよびKDE Plasma 5デスクトップ環境に拡張されました。詳細については、動的キーボードレイアウト同期を参照してください。

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

より応答性が高く予測可能なキーボード動作でユーザーエクスペリエンスを向上させるため、Linux VDAはキーボード設定のデフォルトXKBルールとしてevdevを指定するようになりました。

詳細については、X Keyboard Extension (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、インデックス付きクエリキーを使用してLDAPクエリのパフォーマンスを向上させ、コストを削減します。

Cloud ConnectorのVDAスケーラビリティの向上

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

HDXアダプティブスループットがデフォルトで有効になりました

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

> **注:**
>
> デプロイ管理にAnsibleの使用に関心のあるお客様向けに、包括的なベストプラクティス記事を提供しています。詳細については、「Ansibleを使用したデプロイの管理」を参照してください。

2402 LTSRの新機能

RHEL 8.9、Rocky Linux 8.9、RHEL 9.3、およびRocky Linux 9.3のサポート

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

- RHEL 8.9
- ロッキー・リナックス 8.9
- RHEL 9.3
- ロッキー・リナックス 9.3

詳細については、「システム要件」を参照してください。

新しい.NETバージョンが必要

このリリースから、Linux VDAをインストールする前に、RHEL 7.9とAmazon Linux 2を除くすべてのサポート対象Linuxディストリビューションに.NET Runtime 8.0をインストールする必要があります。RHEL 7.9とAmazon Linux 2については、引き続き.NET Runtime 6.0をインストールしてください。

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

RHEL、Rocky Linux、UbuntuにおけるXfceのサポート

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

詳しくは、「システム要件」を参照してください。

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

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

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

インストールされている依存関係を確認する機能

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

詳しくは、「**XDPing**で実行できるタスク」を参照してください。

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングを以下のディストリビューションに拡張しました。

- SUSE 15.5

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

ログオン時間の内訳

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

公開アプリケーション向けの動的ウィンドウプレビュー

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

この機能は、Citrix Studioで設定できる**動的ウィンドウプレビュー**ポリシーに依存します。詳しくは、「動的ウィンドウプレビュー」および「ポリシーサポートリスト」を参照してください。

新規および最適化されたセッションメトリッククエリユーティリティ

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

このリリースでは、ヘルプ情報も最適化され、**ctxqsession**、**ctxquser**、**ctxqfull**、**ctxquery**などのレガシーセッションメトリッククエリユーティリティにアクセス制御が追加されました。以前はアクセス制御がなかったため、Linux VDAのログオンユーザーは、いずれかのツールを介してすべ

てのセッションのメトリックを照会できました。アクセス制御が追加されたことで、ユーザーは自分のセッションメトリックのみを照会でき、rootユーザーと**ctxadm**グループのユーザーのみが他のユーザーのセッションメトリックを照会する権限を持ちます。

詳しくは、「セッションメトリッククエリユーティリティ」を参照してください。

2311 の新機能

SUSE 15.5およびDebian 11.7のサポート

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

HDX アダプティブスループット

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

AOMedia Video 1 (AV1) のサポート

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

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

システムトレイでグラフィックステータスを利用可能

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

詳細については、「システムトレイ」を参照してください。

複数のオーディオデバイスのサポート（プレビュー）

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

この機能を有効にすると、次のようになります。

- クライアントマシン上のすべてのローカルオーディオデバイスがセッションに表示されます。CitrixAudioSink（オーディオ出力）またはCitrixAudioSource（オーディオ入力）の代わりに、オーディオデバイスはそれぞれのデバイス名で表示されます。セッション内のアプリでオーディオデバイスを選択するか、セッション中にデフォルトのオーディオデバイスを使用できます。これは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ります。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を行うと動的に更新されます。

詳細については、「複数のオーディオデバイスのサポート」を参照してください。

トークンベースの登録（プレビュー）

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

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

- Cloud Connectorのインストールとメンテナンスの必要がなくなります。
- ユーザーおよびマシンの認証におけるADへの依存関係を排除し、ドメインに参加していないマシンの認証を可能にします。

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

簡単インストール機能の強化

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

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

Linux VDAデータ収集プログラム

Linux VDAのインストール後、データ収集プログラムに自動的に参加するようになりました。データ収集プログラムは、統計情報と使用状況データを収集し、Citrix 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。プログラムを無効にして詳細を表示するには、「Linux VDAデータ収集プログラム」を参照してください。

Linuxセッションで利用可能な新しいメトリック

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

- **ICA®遅延**

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

- **ポリシー**

  現在のセッションに適用されているすべてのポリシーは、**セッションの詳細**ビューの［ポリシー］タブに表示されます。

詳しくは、「Linux VMおよびLinuxセッションのメトリック」を参照してください。

強化されたEDT輻輳制御が一般提供開始になりました

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

Rendezvous V2 がデフォルトで有効

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

EDT MTU 検出が VDA でデフォルトで有効になりました

MTU 検出により、EDT はセッション確立時に最大転送単位 (MTU) を自動的に決定できます。これにより、パフォーマンスの低下やセッション確立の失敗につながる可能性のある EDT パケットの断片化を防ぎます。詳しくは、「アダプティブトランスポート」を参照してください。

Secure HDX (プレビュー)

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

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

2308の新機能

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

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

Linux VDA は、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 8.8
- ロッキー Linux 8.8
- RHEL 9.2
- ロッキー Linux 9.2

詳細については、システム要件を参照してください。

マルチセッションLinux VDAでの共有GPUアクセラレーション

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

特定のデータ形式をセッションとクライアント間でコピー＆ペースト可能

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

Questサポートの拡張

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

ブラウザコンテンツリダイレクトのサーバーフェッチとクライアントレンダリング

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

EDT輻輳制御の強化（プレビュー）

Enlightened Data Transport (EDT) プロトコルを最適化するために、新しい輻輳制御アルゴリズムが導入されました。この実装により、EDTはより高いスループットを実現し、遅延を削減して、ユーザーエクスペリエンスを向上させることができます。この機能はデフォルトで無効になっています。詳細については、アダプティブトランスポートを参照してください。

システムトレイのメニュー項目の追加

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

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

詳細については、「システムトレイ」を参照してください。

H.265ハードウェアエンコードの強化

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

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

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

Amazon Linux 2、RHEL 7.9、CentOS 7.9でRealmを徹底的にテスト

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

詳細については、「Amazon Linux 2、CentOS、RHEL、Rocky LinuxにLinux VDAを手動でインストールする」を参照してください。

2305 の新機能

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

SQLiteのサポート

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

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

セッションログオンの機能強化

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

- ユーザーがパスワードとPINコードの表示/非表示を切り替えられるようにし、ユーザーが誤った入力を簡単に見つけられるようにします。
- セッションログオンバナーメッセージと認証ダイアログを別々の画面に表示します。
- セッションログオンプロセスを表示し、非SSOシナリオでのセッション再接続にパスワードまたはPINコードを要求します。
- ユーザーがCitrix Workspaceアプリへのログオンに使用した資格情報とは異なる資格情報でVDAセッションにログオンした場合、認証ダイアログは別々に表示されます。ユーザーは、資格情報が誤って入力された時期を簡単に特定できます。
- 非SSOシナリオで、以下のユーザー認証方法の組み合わせをサポートします。

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

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

一時ホームディレクトリ設定に利用できるレジストリキーの追加

このリリースでは、一時ホームディレクトリ設定に役立つ2つのレジストリキーが追加されました。

- CheckUserHomeMountPoint
- CheckMountPointRetryTime

詳しくは、「一時ホームディレクトリでのログオン」を参照してください。

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

Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするためのFIDO2認証を設定できるようになりました。詳細については、「FIDO2 (プレビュー)」を参照してください。

HDX 3D PROのH.264ロスレス圧縮サポート

NVIDIA GPUによるHDX 3D PROハードウェアアクセラレーションで、H.264ロスレス圧縮サポートが利用可能になりました。詳細については、「H.264ロスレス圧縮」を参照してください。

リダイレクトするURLを指定するためのワイルドカードサポートの拡張

コンテンツをクライアントにリダイレクトできるURLを指定する場合、プロトコルを除くすべてのURLコンポーネントを表すために`*`ワイルドカードを使用できるようになりました。詳細については、「ブラウザーコンテンツリダイレクト」を参照してください。

XDPingを使用したVDAデータバックアップと比較のサポート

トラブルシューティングのために、**XDPing**ツールにVDAバックアップモジュールを追加しました。このモジュールを使用すると、構成、データベース、バイナリ権限データなど、VDAの主要データをいつでもバックアップできます。VDAが正常に動作しているときに、VDAの主要データをバックアップできます。後でVDAに障害が発生した場合に備えて、データの別のコピーをバックアップし、2つのデータコピーを比較してトラブルシューティングを容易にすることができます。詳細については、「**XDPing**」を参照してください。

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

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

> **注:**
>
> SSSDを使用してドメインに接続されている、現在実行中のRHEL 8.x/9.xまたはRocky Linux 8.x/9.x VDAをMCSのテンプレートVMとして使用するには、以下を確認してください。
>
> - VDAは、簡易インストールではなく手動でインストールされます。簡易インストールではRHEL 8.x/9.xおよびRocky Linux 8.x/9.x**Adcli**を使用しますが、SSSDと**Adcli**の組み合わせはMCSでサポートされていません。
> - SambaサーバーがAD認証にSSSDを使用するように構成されています。詳細については、`<https://access.redhat.com/solutions/3802321>`のRed Hatの記事を参照してください。
>
> 詳細については、「MCSを使用したLinux VDAの作成」を参照してください。

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングは以下のディストリビューションに拡張されました。

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

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

2303の新機能

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

RHEL 9.1 および Rocky Linux 9.1 のサポート

サポート対象のディストリビューションとして、RHEL 9.1 および Rocky Linux 9.1 を追加しました。詳細については、システム要件を参照してください。

新しい Linux ストリーミングターゲットデバイスのサポート

Linux ストリーミングを次のディストリビューションに拡張しました。

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

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

簡単なインストール機能強化

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

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

簡易インストールの使用について詳しくは、「簡易インストールを使用してドメイン参加済みVDAを作成する」を参照してください。

XDPingの機能強化

XDPingツールに、VDA登録ステータスを確認および分析する分析モジュールを追加しました。VDA登録ステータスチェックを実行するには、**sudo /opt/Citrix/VDA/bin/xdping -a**コマンドを実行します。

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

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

2301の新機能

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

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

サポート対象のディストリビューションとして、RHEL 8.7、Rocky Linux 8.7、およびSUSE 15.4を追加しました。詳しくは、「システム要件」を参照してください。

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

Waylandのサポート（プレビュー）

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

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

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

動的クライアントドライブマッピングとクライアントフォルダーリダイレクト

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

2212 の新機能

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

RHEL 9.0およびRocky Linux 9.0のサポート

サポート対象のディストリビューションとしてRHEL 9.0とRocky Linux 9.0を追加しました。詳細については、「システム要件」を参照してください。

USBデバイスリダイレクトの機能強化

USBデバイスリダイレクトに関して、以下の機能強化を行いました。

- USBデバイスリダイレクトの新しい仮想ホストコントローラーとしてUSB/IPを使用するようになりました。ほとんどの場合、USB/IPカーネルモジュールはLinuxカーネルバージョン3.17以降でリリースされているため、デフォルトでカーネルモジュールをビルドする必要はありません。
- USB 3.0ポートのサポートを追加しました。クライアントデバイスのUSB 3.0ポートにUSB 3.0デバイスを挿入できるようになりました。
- バルク転送を最適化し、転送速度を向上させました。この向上は、高遅延環境でより顕著です。
- USBデバイスリダイレクトを、**TD-RDF5A Transcend USB device**や複合USBデバイス**Yubico YubiKey OTP+FIDO+CCID**などのより多くのUSBデバイスに拡張しました。

詳細については、「USBデバイスリダイレクト」を参照してください。

NVIDIA製特定非仮想化GPUのハードウェアアクセラレーション

Linux VDAドキュメントでは、非仮想化GPUは以下を指します。

- リモートPCアクセスシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

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

詳しくは、「非仮想化GPU」を参照してください。

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

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

MCS構成を含めるための簡易インストールGUIの拡張

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

Linuxセッションの記録のサポート（プレビュー）

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

データベースオプションが利用可能に（プレビュー）

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

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

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

- MCSを使用してLinux VDAを作成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`

注:

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

詳細については、インストール記事を参照してください。

2210の新機能

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

HDX 3D Pro向けGPUハードウェアアクセラレーションの強化

GPUとLinuxシステムメモリ間のデータ転送効率を向上させました。また、3Dグラフィックレンダリングとハードウェアエンコーディングの遅延も削減しました。これらの機能強化により、ハードウェアリソースの利用が最適化され、1秒あたりのフレーム数（FPS）のパフォーマンスが大幅に向上します。詳細については、H.264ハードウェアエンコーディングを参照してください。

クリップボードを介したデータ転送のサイズ制限のサポート

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

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

ポリシー設定の詳細については、ICA policy settingsのClient clipboard redirectionセクションを参照してください。

Linux VDAがサポートするポリシー設定については、Policy support listを参照してください。

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングを以下のディストリビューションに拡張しました。

- RHEL 8.6
- ロッキーLinux 8.6
- ウブントゥ 22.04

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

シェルスクリプトの機能強化

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

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

Easy install も、特定のドメイン参加方法に関連する構成テンプレートを使用します。

2209 の新機能

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

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

サポート対象のディストリビューションとして、RHEL 8.6、Rocky Linux 8.6、および Ubuntu 22.04 を追加しました。

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

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

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

- SSSD を使用してドメインに参加している SUSE VDA。

これらの VDA にログオンするユーザーの認証に FAS を使用できるようになりました。詳しくは、「フェデレーション認証サービス」を参照してください。

セッションユーザーによるカスタムデスクトップ環境

セッションユーザーは、デスクトップ環境をカスタマイズできるようになりました。この機能を有効にするには、VDA にデスクトップ環境を事前にインストールする必要があります。詳しくは、「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。

GPG 署名

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

公開キーを取得するには：

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

公開キーを使用してLinux VDAパッケージの整合性を検証するには：

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

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

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

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

  ```
  sudo apt-get install dpkg-sig

  gpg --import <path to the public key>

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

2207の新機能

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

非SSOシナリオでのより多くのユーザー認証方法のサポート

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

セッションウォーターマークを構成するための追加オプション

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

KVMハイパーバイザーのサポート

Kernel-based Virtual Machine（KVM）は、Linux向けの主要なオープンソース仮想化テクノロジーです。Linuxの一部であり、Linuxマシンをハイパーバイザーに変えて、複数の分離された仮想マシン（VM）をホストできるようにします。Linux VDAはKVMハイパーバイザーで適切に動作します。ただし、KVMハイパーバイザーは、MCSを使用して仮想マシンを作成する場合にはサポートされていません。

すべてのサポート対象Linuxディストリビューションでファイルコピーと貼り付けが利用可能

以前は、**ファイルコピーと貼り付け機能**はDebian 10、RHEL 7.9、およびUbuntu 18.04でのみ利用可能でした。このリリースから、この機能はLinux VDAがサポートするすべてのLinuxディストリビューションに拡張されました。詳しくは、「ファイルコピーと貼り付け」を参照してください。

2206の新機能

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

Debian 11.3のサポート

サポート対象ディストリビューションとしてDebian 11.3を追加しました。

Azure AD資格情報を使用したユーザー認証のサポート

Linux VDAは、Azure Active Directory (AD) と統合してユーザー認証を提供できるようになりました。この機能により、ユーザーはAzure AD資格情報を使用して、ドメインに参加していないLinuxデスクトップにログオンできます。詳しくは、「Azure Active Directoryでの認証」を参照してください。

クライアントのバッテリー状態表示

Linux VDAは、仮想デスクトップ内のクライアントデバイスのバッテリー状態をリダイレクトして表示できるようになりました。この機能はデフォルトで有効になっています。詳しくは、「クライアントのバッテリー状態表示」を参照してください。

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

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

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

詳しくは、「非vGPUグラフィックカード」を参照してください。

セッションログオン画面でのカスタム背景とバナーメッセージ

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

自動DPIスケーリング

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

LDAPクエリの高速化

ポーリングモードで各LDAPサーバーをクエリするだけでなく、Linux VDAはマルチドメインフォレスト内のLDAPサーバーを同時にクエリできるようになりました。LDAPクエリを高速化するに

は、ドメインコントローラーで**グローバルカタログ**を有効にし、関連するLDAPポート番号を3268として指定します。LDAPサーバーとLDAPポートの指定方法については、お使いのディストリビューションに基づいたLinux VDAインストール記事のCTX_XDL_LDAP_LIST変数に関する説明を参照してください。

カスタムFASポート

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

2204の新機能

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

ハードウェアエンコードが選択的コーデックに拡張されました

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

一時ホームディレクトリでのログオンのサポート

このリリースから、Linux VDA上のマウントポイントが失敗した場合に備えて、一時ホームディレクトリを指定できます。一時ホームディレクトリが指定されている場合、マウントポイントが失敗すると、セッションログオン中にプロンプトが表示されます。ユーザーデータは一時ホームディレクトリに保存されます。詳細については、一時ホームディレクトリでのログオンを参照してください。

Rendezvous HDXトラフィックのSOCKS5プロキシサポート

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

Rendezvous の透過型プロキシのサポート

Rendezvous で透過型 HTTP プロキシがサポートされるようになりました。ネットワークで透過型プロキシを使用している場合、VDA で追加の構成は必要ありません。

GNOME Classic デスクトップのサポート

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

アプリのタスクバーボタンのグループ化

以前は、同じセッションで実行されている公開アプリケーションのすべてのタスクバーボタンが単一のグループにまとめられていました。このリリース以降、アプリの複数のウィンドウが開いている場合でも、各アプリは単一のタスクバーボタンとして表示されます。

2203の新機能

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

Rendezvous V2 の完全サポート

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

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

Rendezvous V1 では HDX トラフィックが Cloud Connector をバイパスできますが、VDA 登録とセッションブローカーのためのすべての制御トラフィックをプロキシするには、Cloud Connector が引き続き必要です。

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

詳細については、「Rendezvous V2」を参照してください。

簡単インストール GUI が利用可能になりました

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

- システム環境を確認する
- 依存関係のインストール

- VDA を指定されたドメインに参加させる
- ランタイム環境を構成する

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

HDXグラフィックの改善

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

2201の新機能

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

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

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

セッションシャドウイング機能がすべてのサポート対象ディストリビューションに拡張されました

セッションシャドウイング機能を使用すると、ドメイン管理者はイントラネット内でユーザーの ICA セッションを表示できます。以前は RHEL 7.x および Ubuntu 16.04 でのみ利用可能でしたが、この機能は Linux VDA がサポートするすべてのディストリビューションに拡張されました。詳細については、「セッションのシャドウイング」を参照してください。

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

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

Transport Layer Security (TLS) 1.3 のサポート

Linux VDA は TLS 1.3 をサポートするようになりました。/opt/Citrix/VDA/sbin ディレクトリにある `enable_vdassl.sh` スクリプトを使用すると、TLS を有効にし、最小 TLS バージョンを設定できます。詳細については、「Secure user sessions using TLS」を参照してください。

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

2112 の新機能

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

Amazon Linux 2 のサポート (プレビュー)

サポート対象のディストリビューションとして Amazon Linux 2 を追加しました。詳細については、「System requirements」を参照してください。

HDX 画面共有の完全サポート

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

HTTPプロキシを介したRendezvous接続

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

Rendezvousプロキシ認証

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

アダプティブオーディオのサポート

アダプティブオーディオはデフォルトで有効になっています。アダプティブオーディオを使用すると、VDA で audio quality policies を手動で構成する必要がありません。アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、プレ

ミアムなオーディオ体験を提供します。詳細については、「Adaptive audio」を参照してください。

新しいLinuxストリーミングターゲットデバイスのサポート

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

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

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

キーボードレイアウト同期の機能強化

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

キーボード入力の強化

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

2110の新機能

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

SUSE 15.3およびSUSE 15.2のサポート

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

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

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

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

PBISを使用してWindowsドメインに参加しているMCSで作成されたLinux VDAにログオンするユーザーを、FASを使用して認証できるようになりました。詳細については、フェデレーション認証サービスを構成するおよびMCSで作成されたVMでFASを有効にするを参照してください。

2つの環境変数がctxfascfg.shに追加されました

ctxfascfg.sh に以下の変数を追加しました。

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

詳しくは、「ctxfascfg.sh の実行」(/ja-jp/linux-virtual-delivery-agent/2110/configuration/federated-authentication-service.html#run-ctxfascfgsh)を参照してください。

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

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

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

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

- RHEL 8.4 / CentOS 8 (2105)
- RHEL 7.9 / CentOS 7.9
- ウブントゥ 16.04
- デビアン 10.9
- SUSE 15.3
- SUSE 15.2

2109 の新機能

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

HDX画面共有のサポート (プレビュー)

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

Linux VDA用 シトリックス バーチャルチャネル ソフトウェア開発キット (SDK) (プレビュー)

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

Linux VDA用Citrix Virtual Channel SDKは、Citrix Virtual Apps and Desktopsダウンロードページからダウンロードできます。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**「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。

PBISを使用するVDAのFASサポート

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

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

Citrix ProvisioningのLinuxストリーミング機能を使用すると、Citrix Virtual Apps and Desktops環境にLinux仮想デスクトップを直接プロビジョニングできます。詳しくは、「Citrix Provisioningを使用してLinux VMを作成する」を参照してください。

MATEデスクトップのキーボードレイアウト同期サポート

キーボードレイアウト同期のサポートをMATEデスクトップに拡張しました。クライアントデバイスのキーボードレイアウトが変更されるたびに、MATEデスクトップ環境をインストールしたVDAのレイアウトもそれに従います。詳しくは、「動的キーボードレイアウト同期」を参照してください。

セッションデータクエリユーティリティ

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

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

新しいLinuxセッションメトリックが利用可能になりました

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

- **ログオン期間**

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

- **セッション自動再接続数**

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

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

2107の新機能

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

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

MCSで作成されたRHEL 7/CentOS 7およびRHEL 8/CentOS 8マシンをWindowsドメインに参加させるためのCentrifyサポートを追加しました。詳しくは、「Linux VMを作成するためのMachine Creation Services（MCS）の使用」を参照してください。

USBリダイレクトのためのカーネルモジュール再構築の簡素化

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

RHEL 8.4およびCentOS 8（2105）のサポート

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

Debianでのスマートカードサポート

Linux仮想デスクトップセッションにログオンする際、クライアントデバイスに接続されたスマートカードを認証に使用できます。このリリースでは、Debian 10.7セッションでスマートカードを使用することもできます。詳しくは、「スマートカードのサポート」を参照してください。

2106の新機能

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

Linux VDAのセルフアップデート

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

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

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

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

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

HDX Web カメラビデオ圧縮のサポート

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

グラフィック品質スライダー

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

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

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

- クリップボード
  - クライアントクリップボードリダイレクト
  - クリップボード選択更新モード
  - プライマリ選択更新モード

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

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

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

2104の新機能

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

Citrix DaaS™におけるドメイン非参加Linux VDAのサポート

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

OpenJDK 11が必須に

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

XDPingの変更点

`ctxsetup.sh`を実行しても`XDPing`はインストールされなくなりました。`sudo /opt/Citrix/VDA/bin/xdping`を実行して`XDPing`をインストールできます。このコマンドは、`XDPing`に必要な**Python3**仮想環境も作成します。詳しくは、「**XDPing**」を参照してください。

Ubuntuのスマートカードサポート

ユーザーは、Linux仮想デスクトップセッションにログオンする際に、クライアントデバイスに接続されたスマートカードを認証に使用できます。このリリースでは、Ubuntu 20.04、Ubuntu 18.04、およびUbuntu 16.04セッションでスマートカードパススルー認証を使用することもできます。詳しくは、「スマートカードを使用したパススルー認証」を参照してください。

MATEデスクトップのサポート

CentOS、RHEL、Ubuntu、Debianで軽量MATEデスクトップのサポートを追加しました。ctxinstall.sh、ctxsetup.sh、およびdeploymcs.shスクリプトで利用可能な新しい変数を通じて、MATEまたはGNOMEデスクトップを指定できるようになりました。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、この変数の値をmateに設定する必要があります。詳しくは、「簡単なインストール」および「RHEL/CentOS用Linux Virtual Delivery Agentのインストール」などの手動インストール記事を参照してください。

RHEL 8、CentOS 8、SUSE 12.5のPBISサポート

RHEL 8、CentOS 8、SUSE 12.5マシンをWindowsドメインに参加させるためのPBISサポートを追加しました。

2103の新機能

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

Debian 10.7およびCentOS 8.3のサポート

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

SUSEマシンをWindowsドメインに参加させるためのSSSDサポート

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

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

Citrix Virtual Apps and Desktops Standard for Azureで、ドメイン参加済みおよび非ドメイン参加済みの両方のLinux VDAを作成するための単一イメージを提供するようになりました。この機能により、イメージの準備とメンテナンスが簡素化されます。詳しくは、「Citrix Virtual Apps and Desktops Standard for AzureでLinux VDAを作成する」を参照してください。

ファイル転送の機能強化

プログレスバーの再設計、ダウンロード速度の高速化、いくつかのバグの解決により、ファイル転送機能が強化されました。また、一度に転送できるファイルの数を10個から100個に増やしました。詳しくは、「ファイル転送」を参照してください。

Citrix Studioでのセッション接続タイマー構成のサポート

以前は、アイドル状態および切断されたセッションの制限時間は、各VDAで個別に`/opt/Citrix/VDA/bin/ctxcfg`を介してのみ構成できました。このリリースでは、Citrix Studioでセッション接続タイマーを構成するための以下のポリシーが追加されました。

- **セッションアイドルタイマー**：アイドルセッションに時間制限を適用するかどうかを決定します。
- **セッションアイドルタイマー間隔**：アイドルセッションの時間制限を設定します。**セッションアイドルタイマー**が**有効**に設定されており、アクティブなセッションが設定された時間内にユーザー入力を受信しなかった場合、セッションは切断されます。
- **切断されたセッションタイマー**：切断されたセッションに時間制限を適用するかどうかを決定します。
- **切断されたセッションのタイマー間隔**: 切断されたセッションがログオフされるまでの間隔を設定します。

ポリシー設定を更新する場合は、展開全体で一貫性があることを確認してください。ポリシーの詳細については、「ポリシーサポートリスト」を参照してください。

アイドルセッションの制限時間が経過すると、警告メッセージが表示されます。例については、次のスクリーンショットを参照してください。**OK**を押すと警告メッセージは閉じますが、セッションをアクティブに保つことはできません。セッションをアクティブに保つには、アイドルタイマーをリセットするためにユーザー入力を提供してください。

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

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

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

各Linux VMの新しいメトリック:

- CPUコア数

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

各Linuxセッションの新しいメトリック:

- アイドル時間

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

Linux VDAのFAS機能強化

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

RHEL 8.3およびUbuntu 18.04.5のLinuxストリーミングサポート（プレビュー）

Citrix ProvisioningでLinuxストリーミング機能を使用すると、Linux仮想デスクトップをCitrix Virtual Apps and Desktops環境に直接プロビジョニングできます。詳しくは、「Linuxストリーミング」を参照してください。

2012年の新機能

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

エンライトンド データ トランスポート (EDT) MTU 検出

EDTは、セッション確立時に最大転送単位（MTU）を自動的に決定します。これにより、パフォーマンスの低下やセッションの確立の失敗につながる可能性のあるEDTパケットの断片化を防ぎます。詳しくは、「EDT MTU検出」を参照してください。

LDAPチャネルバインディングのサポート

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

Linux VDAに統合されたXDPing

Linux **XDPing**ツールは、Linux VDA環境で一般的な構成の問題をチェックするプロセスを自動化するコマンドラインベースのアプリケーションです。以前は、CTX202015ページから**XDPing**をダウンロードし、Linux VDAとは別にツールをインストールする必要がありました。このリリース以降、`XDPing`はLinux VDAに統合されており、Linux VDAのインストール時に自動的にインストールできます。

RHEL 7.9、RHEL 8.3のサポート

サポートされるディストリビューションとしてRHEL 7.9およびRHEL 8.3を追加しました。

Ubuntu 20.04のサポート

このリリースでは、Ubuntu 20.04が完全にサポートされています。

Rendezvousプロトコルのサポート

Linux VDAがRendezvousプロトコルをサポートするようになりました。Citrix Gatewayサービスを使用する環境では、Rendezvousプロトコルにより、HDXセッションはCitrix Cloud Connector™をバイパスし、Citrix Gatewayサービスに直接かつ安全に接続できます。

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

MCSを使用してGCP上にLinux VMを作成できます。

LinuxマシンでWake on LANが利用可能に

統合されたWake on LAN機能がLinuxマシンで利用可能になりました。

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

2009の新機能

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

RHEL 8.2のサポート

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

Ubuntu 20.04のサポート（プレビュー）

Ubuntu 20.04でLinux VDAを配布できます。

ブラウザコンテンツリダイレクトのサポート

このリリース以降、Linux VDAはGoogle Chromeでのブラウザコンテンツリダイレクトをサポートします。ブラウザコンテンツリダイレクトは、許可リストにあるWebページをクライアント側でレンダリングする機能を提供します。詳細については、「ブラウザコンテンツリダイレクト」を参照してください。

セッションウォーターマークのサポート

Linux VDAはセッションウォーターマークをサポートするようになりました。元のドキュメントの内容を変更することなく、セッション画面全体に表示するテキストのレイヤーであるウォーターマークを指定できます。テキストベースのセッションウォーターマークは、データ盗難を抑止し、追跡を可能にするのに役立ちます。この追跡可能な情報は、写真やスクリーンキャプチャを使用してデータを盗む人々への抑止力として、セッションデスクトップに表示されます。詳細については、「セッションウォーターマーク」を参照してください。

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

Microsoft Azureからあらゆるデバイスに仮想アプリとデスクトップを配信するために、Citrix Virtual Apps and Desktops Standard for Azureでドメインに参加していないLinux VDAを作成できるようになりました。詳細については、「Citrix DaaS Standard for AzureでLinux VDAを作成する」および「Citrix Virtual Apps and Desktops Standard for Azure」を参照してください。

動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期を設定するためのポリシーが利用可能

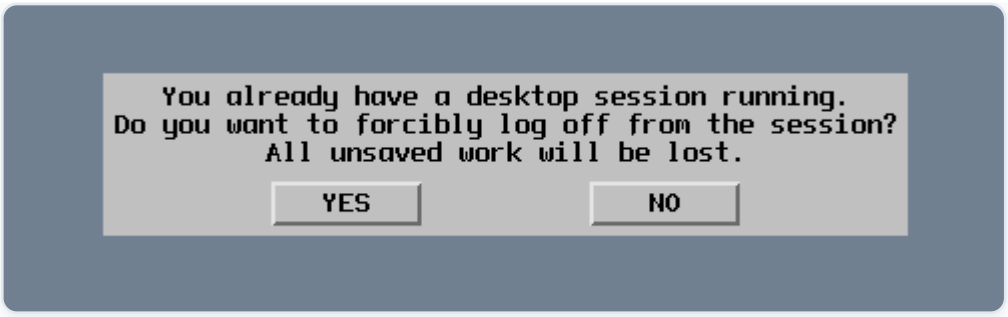
以前は、動的キーボードレイアウト同期およびクライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、各ターゲットVDAのレジストリを編集する必要がありました。このリリース以降、**Client Keyboard Layout Sync and IME Improvement**ポリシーを使用してこれらの機能を設定できます。このポリシーはレジストリ設定よりも優先され、指定したユーザーおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。詳細については、「動的キーボードレイアウト同期」および「クライアントIMEユーザーインターフェイス同期」を参照してください。

VDAから追加のデスクトップセッションを開くときに表示されるプロンプト

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

注:

- gdm3を介してVDAにログインする際にプロンプトで**YES**をクリックすると、コンソールデスクトップの画面が黒くなり、応答しないように見えます。Ctrl+Alt+F1を使用してVDAログインウィンドウに戻ることができます。
- プロンプトは30秒間表示されます。その後、回答は自動的に**NO**に設定されます。



2006年の新機能

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

言語サポート

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

SUSE 12 SP5のサポート

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

クライアントドライブマッピングの機能強化

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

セッションとクライアント間でのファイルコピーと貼り付けのサポート

Citrix Virtual Apps and Desktops 2006およびWindows用Citrix Workspaceアプリ1903以降を使用すると、ユーザーはセッションとローカルクライアント間でファイルをコピー＆ペーストできるようになりました。コピー＆ペーストは、右クリックメニューまたはキーボードショートカットを使用して機能します。詳しくは、「ファイルコピーと貼り付け」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/configure/file/file-copy-and-paste.html)を参照してください。

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

2003の新機能

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

RHEL 8のサポート

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

Citrix Telemetry Serviceとの統合

Citrix Telemetry Service (**ctxtelemetry**) は、Linux VDAソフトウェアと統合されています。これにより、Delivery ControllerからCitrix Scoutを実行して、Linux VDAに関するログを収集できます。詳しくは、「Citrix Scout」(/ja-jp/citrix-virtual-apps-desktops/manage-deployment/cis/scout.html)を参照してください。

軽量NSAP仮想チャネルのサポート

Linux VDAはNSAP仮想チャネルのサポートを追加します。すべてのHDX Insightデータは、NSAP仮想チャネルからのみ取得され、非圧縮で送信されるため、スケーラビリティとパフォーマンスが向上します。詳しくは、「HDX Insight」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/configure/administration/hdx-insight.html)を参照してください。

自動ダウンロード

Citrixは、ファイル転送の機能強化として自動ダウンロードを追加しました。これは、HTML5用Citrix WorkspaceアプリおよびChrome用Citrix Workspaceアプリで利用できます。この機能強化により、VDA上に**Save to My Device**ディレクトリが提供されます。**Save to My Device**ディレクトリにダウンロードまたは移動したファイルは、クライアントデバイスに自動的に転送されます。詳しくは、「ファイル転送」(/ja-jp/linux-virtual-delivery-agent/2402-ltsr/configure/file/file-transfer.html)を参照してください。

NVIDIA Tesla T4のサポート

NVIDIA Tesla T4がサポートされています。

テクニカルプレビューの機能

テクニカルプレビューの機能は、非本番環境または限定的な本番環境で使用でき、お客様がフィードバックを共有する機会を提供します。Citrix®はテクニカルプレビューの機能に対するサポートケースを受け付けていませんが、改善のためのフィードバックを歓迎します。Citrixは、フィードバックの重大度、重要度、および優先度に基づいて対応する場合があります。

テクニカルプレビューの機能一覧

以下の表に、テクニカルプレビューの機能、その説明、および最初に利用可能になったバージョンを示します。

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
オンプレミス環境に拡張されたトークンベースの登録	簡易インストールを使用してドメインに参加していないVDAを作成する場合、クラウドとオンプレミスの両方のデリバリーコントローラーのマシンカタログにドメインに参加していないVDAを登録できます。Citrix Web Studioを使用して、空のマシンカタログを作成し、登録トークンを生成できます。詳細については、「簡易インストールを使用したドメインに参加していないLinux VDAの作成（プレビュー）」	2407	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	を参照してください。		
オーディオのロス許容モード	オーディオは、Enlightened Data Transport (EDT) の損失許容プロトコルでサポートされています。この機能により、ユーザーが高遅延およびパケット損失のあるネットワークを介して接続している場合のリアルタイムストリーミングのユーザーエクスペリエンスが向上します。この機能を有効にすると、Citrix Virtual Apps and Desktopsの適応型トランスポートは、より良いオーディオエクスペリエンスのためにEDT損失許容トランスポートプロトコルを使用します。この機能はデフォルトで無効になっています。詳細については、「オーディオ機能」の記事の「オーディオの損失許容モード」を参照してください。	2407	2411
適応型オーディオのオーディオ品質エンハンサー	バージョン2411以降、適応型オーディオのオーディオ品質エンハンサーをプレ	2411	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	ビュー機能として導入しました。この機能強化により、以前のサンプルからオーディオをインテリジェントに再構築することで、短期間のパケット損失や中断を効果的に管理し、品質の著しい劣化を防ぎます。さらに、必要な場合にのみ失われたオーディオパケットを適応的に回復します。オーディオ品質エンハンサーは、パケット損失の持続的な変化に基づいて自動的に有効/無効になり、良好なネットワーク条件と不良なネットワーク条件の両方でオーディオ再生と録音の品質を最適化します。詳細については、「オーディオ機能」を参照してください。		
複数のオーディオデバイスのサポート	この機能により、Citrix Workspaceアプリがインストールされているクライアントマシン上の複数のオーディオデバイスをリモートLinux VDAセッションにリダイレクトできます。この機能を有効	2311	2411

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	にすると、クライアントマシン上のすべてのローカルオーディオデバイスがセッションに表示されます。CitrixAudioSink（オーディオ出力）またはCitrixAudioSource（オーディオ入力）の代わりに、オーディオデバイスはそれぞれのデバイス名で表示されます。セッション内のアプリでオーディオデバイスを選択するか、セッション中にデフォルトのオーディオデバイス（クライアントマシンのデフォルトオーディオデバイスでもある）を使用できます。必要に応じて、クライアントマシンのシステム設定からデフォルトのオーディオデバイスを変更できます。クライアントマシンのデフォルトオーディオデバイスが更新されると、新しいデバイスがセッションのデフォルトオーディオデバイスとして表示されます。また、オーディオデバイスは、プラ		

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	グインまたは取り外し時にセッション内で動的に更新されます。詳細については、「複数のオーディオデバイスのサポート」を参照してください。		
トークンベースの登録	この機能により、ドメインに参加していないVDAをマシンカタログに登録し、トークンファイルを使用してこれらのVDAをCitrix Cloudコントロールプレーンに認証できます。トークンベースの登録は、Citrix以外のプロビジョニングテクノロジーを使用してマシン（物理または仮想）を独自に準備するユースケースに最適です。これにより、Cloud Connectorのインストールとメンテナンスの必要がなくなり、ユーザーおよびマシン認証のAD依存関係が解消され、ドメインに参加していないマシンの認証が可能になります。詳細については、「簡易インストールを使用したドメインに参加していないLinux	2311	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	VDAの作成（プレビュー）」を参照してください。		
セキュアな HDX	Citrix Workspaceアプリ（クライアント）とVDA（セッションホスト）の間でICAセッションをエンドツーエンドで暗号化できます。エンドツーエンド暗号化（E2EE）機能により、Citrix Gatewayを含む中間ネットワーク要素がICAトラフィックを復号化することを防ぎます。これにより、環境のセキュリティ体制が向上し、構成と管理が容易になります。詳細については、「Secure HDX（プレビュー）」を参照してください。	2311	該当なし
強化されたEDT輻輳制御	Enlightened Data Transport (EDT) プロトコルを最適化するために、新しい輻輳制御アルゴリズムが導入されました。この実装により、EDTはより高いスループットを達成し、遅延を削減して、ユーザーエクスペリエンスを向上させることができま	2308	2311

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	す。この機能はデフォルトで無効になっています。詳細については、「適応型トランスポート」を参照してください。		
Fast Identity Online (FIDO2) 認証のサポート	Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするためのFIDO2認証を設定できるようになりました。詳細については、FIDO2 (プレビュー)を参照してください。	2305	該当なし
Waylandのサポート	Linux VDAは、RHEL 9.4/9.3/9.2、Rocky Linux 9.4/9.3/9.2、およびUbuntu 22.04のGNOMEでWaylandをサポートします。詳細については、Wayland (プレビュー)を参照してください。	2301	該当なし
Linuxセッションの録画のサポート	Linux VDAでホストされているセッションを録画および再生できます。詳細については、セッション録画 (プレビュー)を参照してください。	2212	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
データベースオプション	PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後、`/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。詳細については、インストール記事を参照してください。	2212	2305
Amazon Linux 2のサポート	Amazon Linux 2がサポート対象のディストリビューションとして追加されました。詳細については、システム要件を参照してください。	2112	2203
HDX画面共有	Linux VDAを使用すると、仮想デスクトップの画面を他の仮想デスクトップのセッションユーザーと共有できます。画面共有機能はデフォルトで無効になっています。詳細については、HDX画面共有を参照してください。	2109	2112
Linux VDA 用 Citrix Virtual Channel ソフトウェア開発キット (SDK)	Linux VDA用Citrix Virtual Channel SDKは、Citrix Virtual Apps and Desktops ダウンロードページからダウンロードで	2109	該当なし

テクニカルプレビュー機能	説明	提供バージョン	一般提供 (GA) バージョン
	きます。適切なバージョンのCitrix Virtual Apps and Desktopsを展開し、**Components**をクリックしてLinux VDAのダウンロードを選択します。Virtual Channel SDKを使用すると、VDAで実行するサーバー側アプリケーションを作成できます。詳細については、Linux VDA用Citrix Virtual Channel SDKドキュメントを参照してください。		
Ubuntu 20.04のサポート	Ubuntu 20.04にLinux VDAをインストールできます。	2009	2012

システム要件

Linux Virtual Delivery Agent (VDA) の最新リリースは、Citrix Virtual Apps and Desktops と連携しています。また、ライフサイクルの終了に達していない以前のバージョンのCitrix Virtual Apps and Desktops とも下位互換性があります。Citrix製品のライフサイクルに関する情報、およびCitrixが特定のバージョンの製品のサポートを終了する時期を確認するには、Citrix製品ライフサイクルマトリックスを参照してください。

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

ここで説明されていないコンポーネント（Citrix Workspace™アプリなど）のシステム要件は、それぞれのドキュメントセットに記載されています。

Long Term Service (LTSR) 環境でのCurrent Release (CR) の使用に関する情報、およびその他のFAQについては、Knowledge Centerの記事を参照してください。

サポートされているLinuxディストリビューション、Xorg バージョン、およびデスクトップ環境

このバージョンのLinux VDAがサポートするLinuxディストリビューション、Xorgバージョン、およびデスクトップ環境の表については、次の表を参照してください。詳細については、XorgModuleABIVersionsを参照してください。

LINUXディストリビューション	XORGバージョン	サポートされているデスクトップ
アマゾン Linux 2	1.20	MATE
デビアン 12.7/12.5	1.20	グノーム、グノーム クラシック、ケーディーイー、マテ、エックスエフシーイー
デビアン 11.11	1.20	グノーム、グノーム クラシック、ケーディーイー、マテ

LINUXディストリビューション	XORGバージョン	サポートされているデスクトップ
RHEL 9.4/9.2	1.20	グノーム、エックスエフシーイー
RHEL 8.10/8.8	1.20	グノーム、グノーム クラシック、マテ、エックスエフシーイー
ロッキー リナックス 9.4/9.2	1.20	グノーム、エックスエフシーイー
ロッキーリナックス 8.10/8.8	1.20	グノーム、グノームクラシック、ケイディーイー、マテ、エックスエフシーイー
SUSE 15.6	1.20	グノーム、グノームクラシック、マテ
ウブントゥ 22.04/20.04	1.21	グノーム、グノームクラシック、ケーディーイー、マテ、エックスエフシーイー
ウブントゥ 24.04	1.21	GNOME (Mutter 46.2-1 以降が必要), GNOME Classic (Mutter 46.2-1 以降が必要), KDE, MATE, Xfce

重要:

Mesaグラフィックスライブラリ（特にバージョン24.1.0より高いもの）は、Ubuntu 24.04の仮想デスクトップで、特定のシステムアプリケーション（ファイルマネージャー、設定、電卓など）に表示上の問題を引き起こしています。Mesaをすでに更新しているかどうかに応じて、2つの回避策があります。詳細については、「既知の問題」(/ja-jp/linux-virtual-delivery-agent/2411/whats-new/known-issues.html)を参照してください。

注:

- OSベンダーからのサポートが終了すると、Citrixは問題の修復能力が制限される可能性があります。非推奨または削除されたプラットフォームについては、「非推奨」(/ja-jp/linux-virtual-delivery-agent/2411/whats-new/removed-features.html)を参照してください。
- 少なくとも1つのデスクトップがインストールされている必要があります。セッションで使用するデスクトップ環境は、ctxinstall.shまたはctxsetup.shスクリプトを使用して指定できます。
- Red Hat Enterprise Linuxのドキュメントによると、RHEL 9で利用可能なデスクトップ環境はGNOMEのみです。ただし、EPELリポジトリがインストールされている場合は、RHEL 9およびRocky Linux 9でXfceデスクトップ環境も使用できます。
- Ubuntuでは `HWE kernel` または `HWE Xorg` を使用しないでください。
- カーネルに対するサードパーティの変更により、Linux VDAに必要なモジュールが不足する場合があります。この場合、必要なカーネルモジュールをビルドしてインストールする必要があります。新しいカーネルモジュールをインストールする際は、セキュアブートが無効になっていることを確認してください。
- ユーザー名の形式は、現在のディスプレイマネージャーの `systemd` 構文規則に準拠している必要があります。`systemd` ユーザー名構文の詳細については、ユーザー/グループ名構文を参照してください。

.Net の要件

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

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

ネットワークポート

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

サポートされているホストプラットフォームと仮想化環境

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

> **注:**

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

> 2203リリース以降、Citrix Virtual Apps and Desktops™およびCitrix DaaS向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal Hybrid Multi-Cloud (HMC) ライセンスが必要です。

Active Directory統合パッケージ

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

	WINBIND (ウィンバインド)	SSSD	セントリファイ	PBIS	クエスト
アマゾン リナックス 2	はい	はい	はい	はい	いいえ
デビアン 12.7/12.5	はい	はい	はい	はい	はい

	WINBIND (ウィンバインド)	SSSD	セントリファイ	PBIS	クエスト
デビアン 11.11	はい	はい	はい	はい	いいえ
RHEL 9.4/9.2、ロッキー リナックス 9.4/9.2/8.10/8.8	はい	はい	はい	いいえ	はい (Quest v4.1 以降)
RHEL 8.10/8.8	はい	はい	はい	はい	はい (Quest v4.1 以降)
SUSE 15.6	はい	はい	はい	はい	いいえ
ウブントゥ 24.04	はい	はい	いいえ	いいえ	いいえ
ウブントゥ 22.04/20.04	はい	はい	はい	はい	はい (Quest v4.1 以降)

Cloud Connector のサイズとスケールの考慮事項

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

- 各 Citrix Cloud Connector™ (4 vCPU、10 GB メモリ) は、6,000 台の Linux VDA をサポートできます。
- 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は最大 6,000 台の Linux VDA を展開します。

データベースの考慮事項

- VDI モードでは SQLite を使用し、ホスト型共有デスクトップ配信モデルでは PostgreSQL を使用することをお勧めします。

- 簡単なインストールと MCS の場合、手動でインストールすることなく、SQLite または PostgreSQL を使用するように指定できます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。
- 手動インストールの場合、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は、Linux VDA パッケージのインストール後に **/etc/xdl/db.conf** を編集して、どちらか一方を使用するように指定できます。

HDX™ 3D プロ

Citrix Virtual Apps™ and Desktops の HDX 3D Pro は、GPU アクセラレーション対応のデスクトップとアプリケーションを可能にします。優れた 3D グラフィックエクスペリエンスを確保するために、ネットワーク帯域幅、遅延、およびその他の関連するインフラストラクチャ条件を慎重に考慮することをお勧めします。たとえば、可能な限り VDA と Citrix Workspace アプリを同じリージョンに展開します。

ハイパーバイザー

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

- XenServer® (旧 Citrix ハイパーバイザー)
- VMware vSphere ハイパーバイザー
- ニュータニックス AHV
- マイクロソフト アジュール
- アマゾン ウェブ サービス (AWS)
- Google Cloud プラットフォーム (GCP)

> **注:**
>
> 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は以下の種類のGPUをサポートしています。

NVIDIA vGPU

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

お使いのGPUカードに最新のvGPUドライバーをインストールしてください。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は以下を指します。

- リモートPCアクセスシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

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

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

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

インストール概要

このセクションでは、以下の手順について説明します。

- インストール準備
- 簡易インストールを使用してドメイン参加済みVDAを作成
- MCSを使用してドメイン非参加のLinux VDAを作成
- 簡易インストールを使用してドメイン非参加のLinux VDAを作成（プレビュー）
- MCSを使用してLinux VDAを作成
- MCSを使用してFASが有効なドメイン参加済みLinux VDAを作成
- Citrix Provisioningを使用してLinux VDAを作成
- シトリックス ダース スタンダード フォー アジュール でリナックス ブイディーエー を作成
- Linux VDAを手動でインストール
  - Amazon Linux 2、CentOS、RHEL、およびRocky LinuxにLinux VDAを手動でインストール
  - SUSEにLinux VDAを手動でインストール
  - UbuntuにLinux VDAを手動でインストール
  - DebianにLinux VDAを手動でインストール

インストール準備

この記事では、特定の状況に基づいて適切なインストール方法を選択する方法を提示し、関連するインストールガイドに誘導します。Citrix Virtual Apps and DesktopsおよびCitrix DaaSの展開計画については、インストール準備と開始: 展開の計画と構築をそれぞれ参照してください。

VDAのインストールを開始する前に、以下に注意してください。

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

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

注:

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

簡易インストールを使用してドメイン参加済みVDAを作成する

重要:

- 新規インストールの場合、迅速なインストールについてはこの記事を参照することをお勧めします。この記事では、簡易インストールを使用してLinux VDAをインストールおよび構成する方法を順を追って説明します。簡易インストールは、時間と労力を節約し、手動インストールよりもエラーが発生しにくくなります。必要なパッケージをインストールし、構成ファイルを自動的にカスタマイズすることで、Linux VDAの実行環境をセットアップするのに役立ちます。
- ドメインに参加していないVDAを作成するには、Machine Creation Services (MCS) と簡易インストールの両方を使用できます。詳細については、「MCSを使用してドメインに参加していないLinux VDAを作成する」および「簡易インストールを使用してドメインに参加していないLinux VDAを作成する（プレビュー）」を参照してください。
- ドメインに参加していないVDAで利用可能な機能については、「ドメインに参加していないVDA」を参照してください。

ステップ1：構成情報とLinuxマシンを準備する

簡易インストールに必要な以下の構成情報を収集します。

- ホスト名 – Linux VDAをインストールするマシンのホスト名。
- ドメインネームサーバーのIPアドレス。
- NTPサーバーのIPアドレスまたは文字列名。
- ドメイン名 – ドメインのNetBIOS名。
- レルム名 – Kerberosレルム名。
- ドメインの完全修飾ドメイン名（FQDN）。
- Active Directory (AD) 統合方法 - 現在、簡易インストールはSSSD、Winbind、Centrify、PBIS、およびQuestをサポートしています。簡易インストールは、RHELおよびRocky Linuxの場合にのみQuest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。

重要:

- Linux VDA をインストールするには、リポジトリが Linux マシンに正しく追加されていることを確認します。
- セッションを起動するには、X Window System とデスクトップ環境がインストールされていることを確認します。
- セキュリティのため、簡易インストールではドメイン参加パスワードは保存されません。対話モードで簡易インストールスクリプト (ctxinstall.sh) を実行するたびに、ドメイン参加パスワードを手動で入力する必要があります。サイレントモードでは、ドメイン参加パスワードを **/Citrix/VDA/sbin/ctxinstall.conf** で設定するか、パスワードをエクスポートする必要があります。ドメイン参加に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の権限を委任してください。これを行うには、**Delegation of Control Wizard** を使用してドメインコントローラーで制御を委任します。

手順 2：ハイパーバイザーを準備する

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

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

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

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

```
su -
```

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

このコマンドは 0 または 1 を返します。

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

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

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

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

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

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

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

管理オペレーティングシステムから：

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**が選択されていることを確認します。

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

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとハイパーバイザーに関する問題が発生します。両方ともシステムクロックを同期しようとします。他のサーバーとのクロックのずれを防ぐため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

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

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。
4. 「**VMware Tools**」を選択します。
5. 「**Advanced**」ボックスで、「**Synchronize guest time with host**」のチェックを外します。

ステップ3：.NETのインストール

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

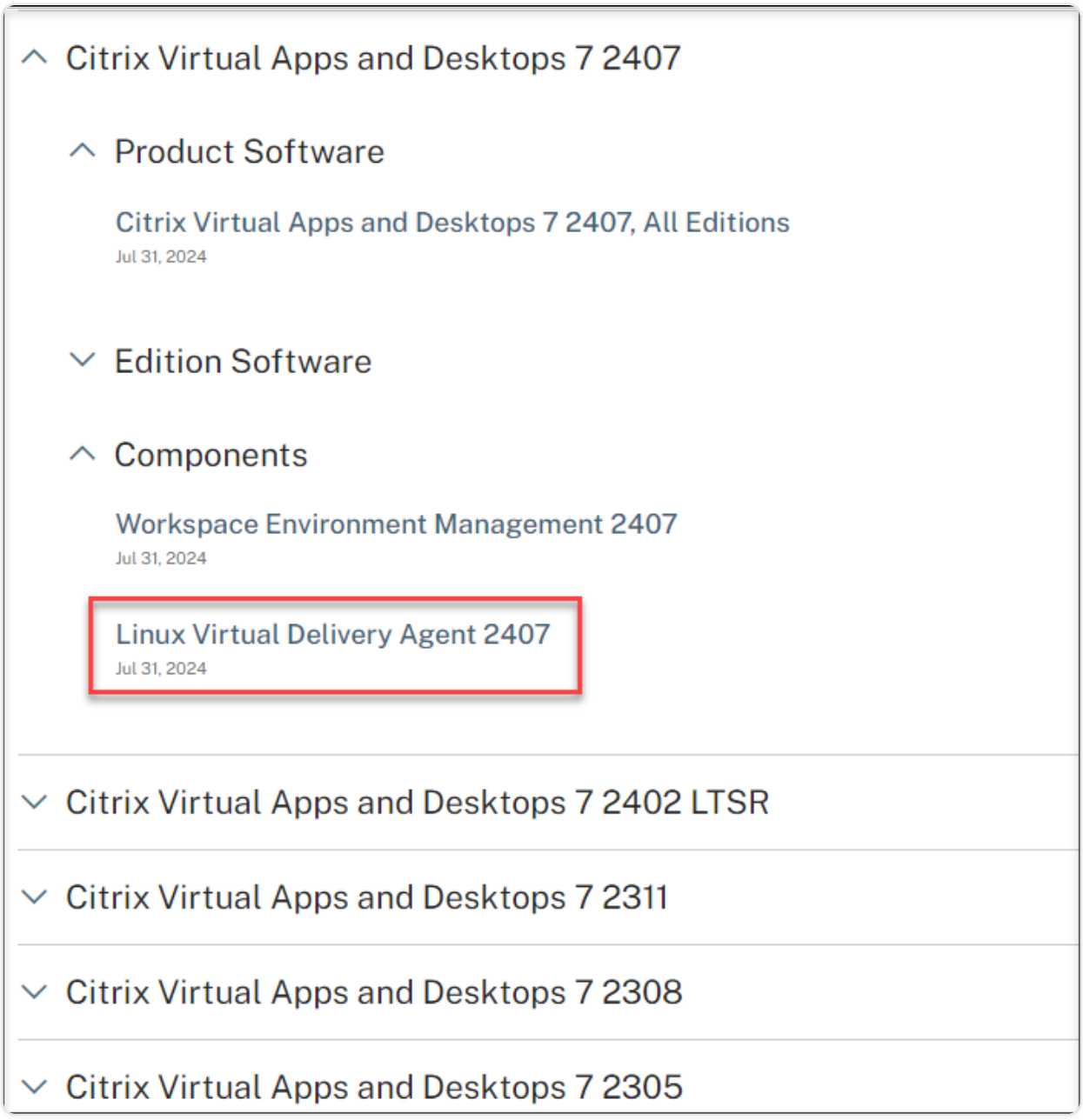
必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

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

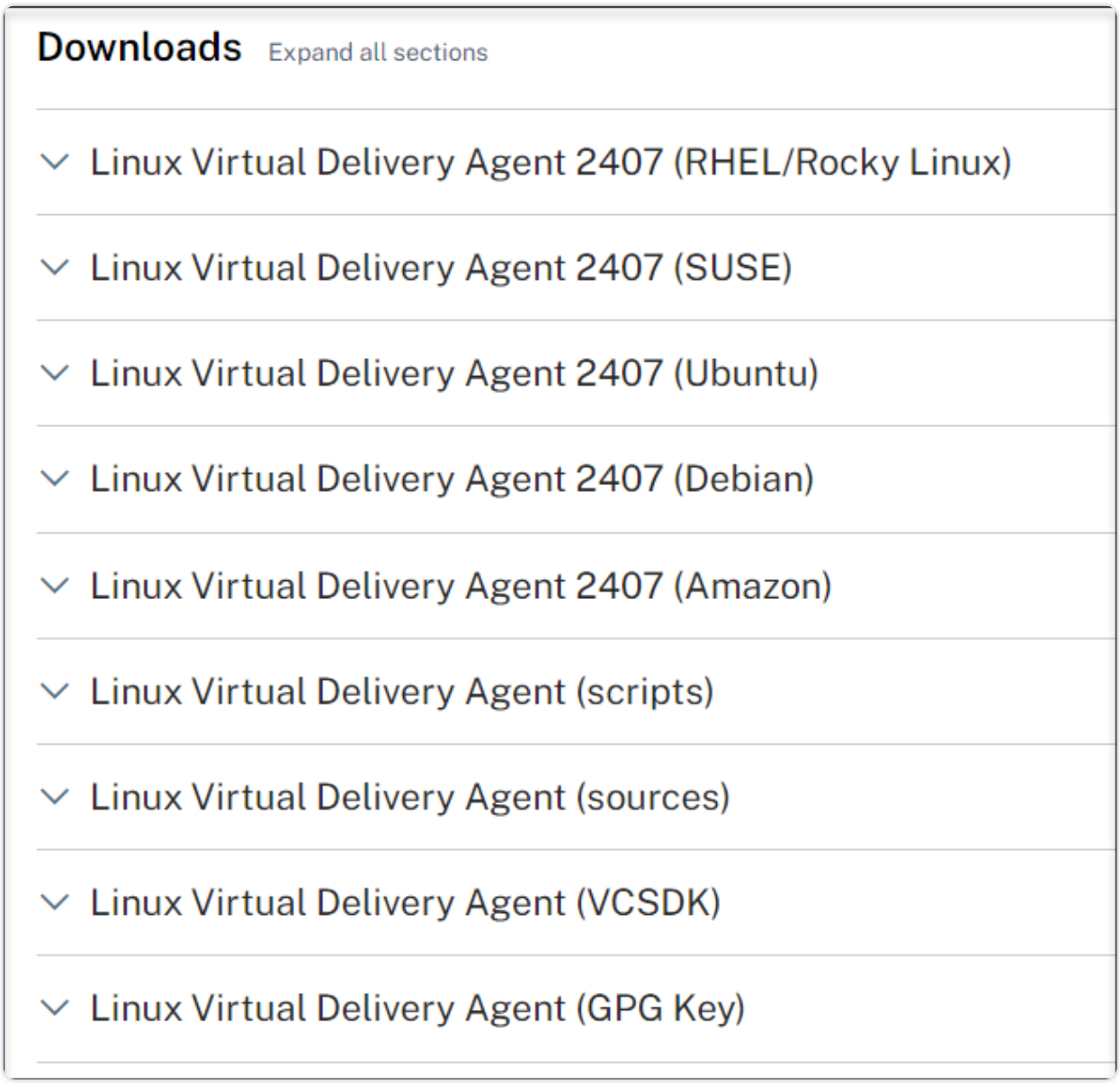
コマンドの出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンドの出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

ステップ4：Linux VDAパッケージのダウンロード

1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 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のダウンロードにアクセスします。



5. お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：

   GPG公開鍵 (/en-us/linux-virtual-delivery-agent/2411/media/gpg-key.png)

   公開鍵を使用してLinux VDAパッケージの整合性を検証するには：

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

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

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

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

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

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

ステップ5：Linux VDAパッケージをインストールする

Linux VDAの環境をセットアップするには、次のコマンドを実行します。

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

注:

- RHELおよびCentOSの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>` の手順を参照してください。
- RHEL 9.4/9.2およびRocky Linux 9.4/9.2にLinux VDAをインストールする前に、**libsepol** パッケージをバージョン3.4以降に更新してください。

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

注:

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAにアクセスできなくなる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワー

ドを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

Ubuntu/Debianディストリビューションの場合：

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

sudo apt-get install -f
```

注:

- Debian 11ディストリビューションに必要な依存関係をインストールするには、`deb http://deb.debian.org/debian/ bullseye main` の行を/etc/apt/sources.listファイルに追加します。
- GCP上のUbuntu 24.04、22.04、20.04の場合、RDNSを無効にします。そのためには、/etc/krb5.confの**の下にrdns = false**の行を追加します。

SUSEディストリビューションの場合：

1. AWS、Azure、GCP 上の SUSE 15.6 の場合、以下を確認してください。
   - **libstdc++6** バージョン 12 以降を使用していること。
   - **/etc/sysconfig/windowmanager** 内の **Default_WM** パラメーターが **"gnome"** に設定されていること。
2. Linux VDA をインストールするには、次のコマンドを実行します。

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

ステップ 6: NVIDIA GRID ドライバーをインストールする

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

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

- XenServer
- ヴイエムウェア イーエスエックス
- ニュータニックス AHV

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

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

ステップ 7: 使用するデータベースを指定する

Linux VDA パッケージのインストール後、SQLite と PostgreSQL を切り替えることができます。これを行うには、次の手順を完了します。

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

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

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

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432
```

```
## PostgreSQL customized
# only the following value means true, otherwise false:
#     true
#     yes
#     y
#     YES
#     Y
# default is false
DbCustomizePostgreSQL=false

## PostgreSQL service name
# specify the service name of PostgreSQL for Linux VDA
# default is "postgresql"
DbPostgreSQLServiceName="postgresql"
```

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

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

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

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

注:

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

ctxinstall.sh

ctxinstall.shは、いくつかの事前構成を行い、VDA実行環境変数を設定するためのEasy Installスクリプトです。

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

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

注:

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

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

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は 'n' です。ドメイン参加シナリオの場合は、'n' に設定します。
- **CTX_XDL_AD_INTEGRATION='sssd\|winbind\|centrify\|pbis\|quest'** – Linux VDA は、Delivery Controller で認証するために Kerberos 構成設定を必要とします。Kerberos 構成は、システムにインストールおよび構成されている Active Directory 統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDA は、Delivery Controller に登録するために、Delivery Controller の完全修飾ドメイン名 (FQDN) のスペース区切りリストを必要とします。少なくとも 1 つの FQDN または CNAME を指定する必要があります。
- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル (VDI) またはホスト型共有デスクトップ配信モデルとして構成するかどうか。HDX 3D Pro 環境の場合は、値を **'y'** に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ (シングルセッション) モード (つまり、CTX_XDL_VDI_MODE=**'y'**) 用に構成されます。
- **CTX_XDL_START_SERVICE='y\|n'** – 構成完了時に Linux VDA サービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシン起動後に Linux Virtual Desktop サービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA サービスは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux Virtual Desktop のシステムファイアウォールで、必要なポート (デフォルトではポート 80 および 1494) を自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または**Xfce**デスクトップ環境を指定します。**'<none\>'**に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（**ctxvda**）をサポートするために.NETをインストールするパス。デフォルトのパスは**'/usr/bin'**です。

- **CTX_XDL_VDA_PORT=port-number** – Linux VDAはTCP/IPポートを介してDelivery Controllerと通信します。
- **CTX_XDL_SITE_NAME=<dns-name\>** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。不要な場合は、**'<none\>'**に設定します。
- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDAはLDAPサーバーを検出するためにDNSをクエリします。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを含むLDAP FQDNのスペース区切りリストを提供できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数はデフォルトで**'<none\>'**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPをクエリします。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**'<none\>'**に設定します。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y\|n'** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。
- **CTX_EASYINSTALL_DNS='<ip-address-of-dns\>'** – DNSのIPアドレス。
- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDAサーバーのホスト名。
- **CTX_EASYINSTALL_NTPS=address-of-ntps** – NTPサーバーのIPアドレスまたは文字列名。
- **CTX_EASYINSTALL_REALM=realm-name** – Kerberosレルム名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_USERNAME=domain-user-name** – マシンをドメインに参加させるユーザーの名前。
- **CTX_EASYINSTALL_PASSWORD=password** – マシンをドメインに参加させるユーザーのパスワード。

> **注:**
>
> ドメイン参加に管理者アカウントを使用しないことをお勧めします。代わりに、管理者アカウント以外のActive Directoryユーザーにドメイン参加の権限を委任してください。これを行

うには、**Delegation of Control Wizard**を使用してドメインコントローラーで制御を委任します。

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

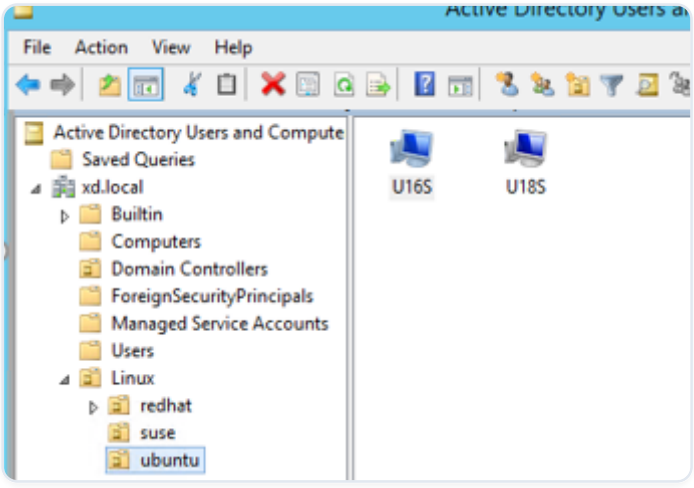
- **CTX_EASYINSTALL_NETBIOS_DOMAIN=netbios-domain-name** – NetBIOSドメイン名は通常、ドット（.）で区切られたDNSドメイン名の最初のコンポーネントです。そうでない場合は、別のNetBIOSドメイン名をカスタマイズしてください。この変数はオプションです。
- **CTX_EASYINSTALL_OU=ou-value** – OUの値は、さまざまな**AD**統合方法によって異なります。OU値の例については、この記事の考慮事項セクションの表を参照してください。この変数はオプションです。
- **CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=centrify-local-path** – Easy installは、インターネットからCentrifyパッケージをダウンロードするのに役立ちます。ただし、Centrifyがすでにインストールされている場合は、この変数で定義されたローカルディレクトリからCentrifyパッケージを取得できます。この変数はオプションです。
- **CTX_EASYINSTALL_PBIS_LOCAL_PATH= pbis-local-path** – Easy installは、インターネットからPBISパッケージをダウンロードするのに役立ちます。ただし、PBISがすでにインストールされている場合は、この変数で定義されたローカルディレクトリからPBISパッケージを取得できます。この変数はオプションです。

考慮事項

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

  a. 特定のOUがドメインコントローラーに存在することを確認します。

     OUの例については、次のスクリーンショットを参照してください。



b. **/opt/Citrix/VDA/sbin/ctxinstall.conf**で環境変数**CTX_EASYINSTALL_OU**を設定します。

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

OS	ウィンバインド	SSSD	セントリファイ	PBIS
アマゾン リナックス 2	`"Linux/amazon"`	`"Linux/amazon"`	`"XD.LOCAL/Linux/amazon"`	`"Linux/amazon"`
デビアン	`"Linux/debian"`	`"Linux/debian"`	`"XD.LOCAL/Linux/debian"`	`"Linux/debian"`
RHEL 9.4/9.2、ロッキー リナックス 9.4/9.2	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	該当なし	該当なし
RHEL 8.x	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	`"XD.LOCAL/Linux/redhat"`	`"Linux/redhat"`
ロッキー リナックス 8.x	`"OU=redhat,OU=Linux"`	`"OU=redhat,OU=Linux"`	該当なし	該当なし
SUSE	`"Linux/suse"`	`"Linux/suse"`	`"XD.LOCAL/Linux/suse"`	`"Linux/suse"`

OS	ウィンバインド	SSSD	セントリファイ	PBIS
ウブントゥ	`"Linux/ubuntu"`	`"Linux/ubuntu"`	`"XD.LOCAL/Linux/ubuntu"`	`"Linux/ubuntu"`

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

  ログ:

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

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

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

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

  a. ドメインコントローラーで**管理ツール**を開きます。

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

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

  - マシンが純粋な**IPv6**ネットワーク経由で必要なパッケージをダウンロードできるように、Linuxリポジトリを構成する必要があります。
  - Centrifyは純粋な**IPv6**ネットワークではサポートされていません。

  注:

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

- セッションで使用するデスクトップ環境は、前述のCTX_XDL_DESKTOP_ENVIRONMENT変数を使用して指定できます。コマンドを実行するか、システムトレイを使用してデスクトップ環境を切り替えることもできます。詳細については、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- ドメインに参加する方法としてCentrifyを選択した場合、ctxinstall.shスクリプトにはCentrifyパッケージが必要です。ctxinstall.shがCentrifyパッケージを取得する方法は次のとおりです。
  - Easy Installは、Centrifyパッケージをインターネットから自動的にダウンロードするのに役立ちます。各ディストリビューションのURLは次のとおりです。

    Amazon Linux 2/RHEL: wget `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-rhel6-x86_64.tgz>` を実行します。

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

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

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

```
ls  -ls  /home/mydir

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

      Amazon Linux 2、RHEL 8、SUSE 15.6 の場合: wget `<https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh>`

      Debian、Ubuntuの場合: wget `<https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.deb.sh>`

- インターネットからPBISパッケージの特定のバージョンを取得します。これを行うには、/opt/Citrix/VDA/sbin/ctxinstall.shファイル内の「pbisDownloadRelease」および「pbisDownloadExpectedSHA256」の行を変更します。

  例については、次のスクリーンショットを参照してください。



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

対話モード

対話モードで**ctxinstall.sh**スクリプトを実行するには、**-S**オプションなしで**sudo /opt/Citrix/VDA/sbin/ctxinstall.sh**コマンドを使用します。コマンドラインインターフェイスの各プロンプトで、関連する変数値を入力します。変数がすでに設定されている場合、ctxinstall.shは変更したい場合に確認を求めます。

サイレントモード

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

各変数のエクスポートされた値は、設定されていない限り、**/Citrix/VDA/sbin/ctxinstall.conf**の値よりも優先されます。ドメイン参加パスワードを除き、更新されたすべての値は**/Citrix/VDA/sbin/ctxinstall.conf**に保存されます。したがって、サイレントモードでは、/Citrix/VDA/sbin/ctxinstall.conf**でドメイン参加パスワードを設定するか、パスワードをエクスポートする必要があります。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

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

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

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'
```

```
export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

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

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

export CTX_XDL_VDA_PORT='<port-number>'

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

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

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

export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'

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

export CTX_EASYINSTALL_HOSTNAME='<host-name>'

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

export CTX_EASYINSTALL_REALM='<realm-name>'

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

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

export CTX_EASYINSTALL_PASSWORD='<password>'

export CTX_EASYINSTALL_NETBIOS_DOMAIN='<netbios-domain>'

export CTX_EASYINSTALL_OU='<organization-unit>'

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

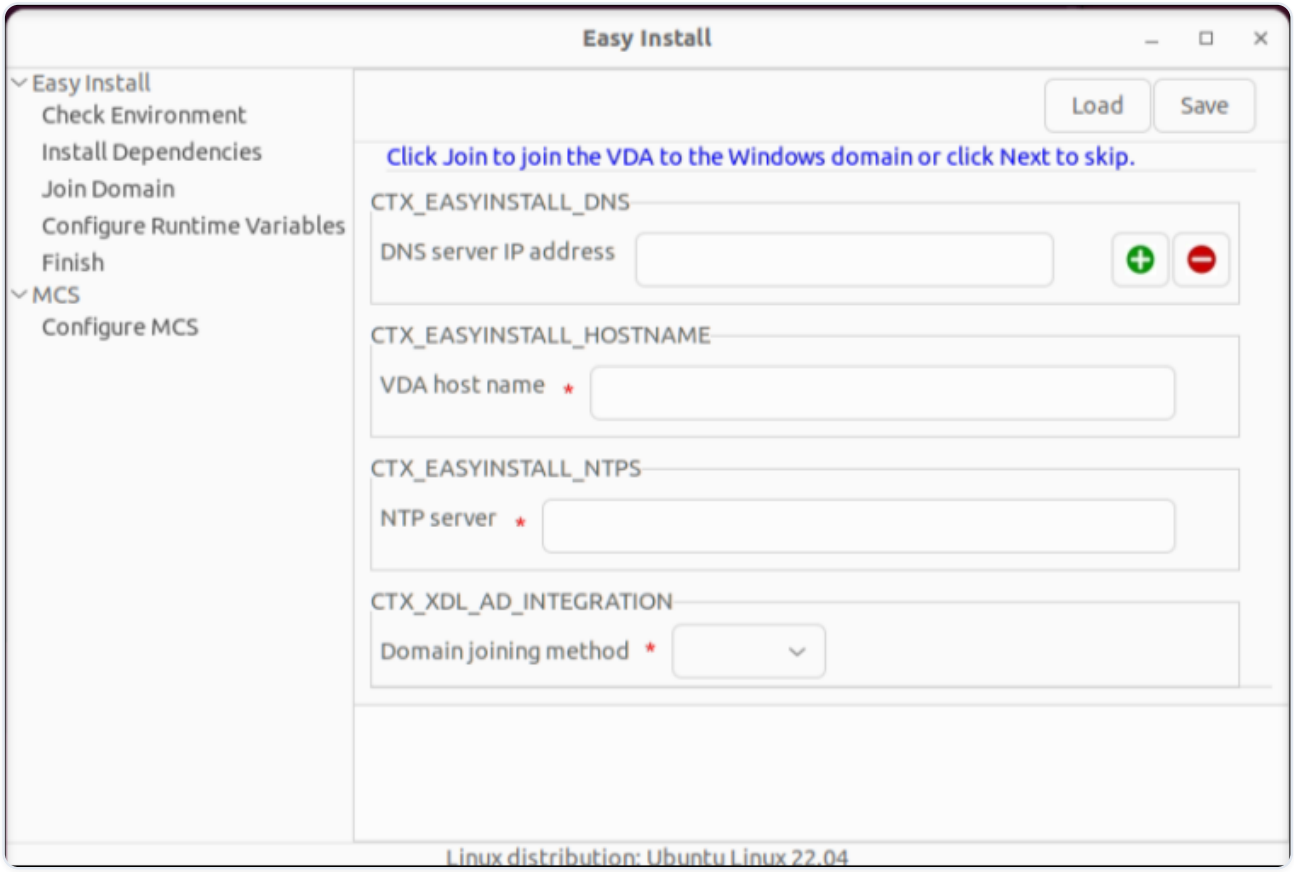
sudoコマンドを実行するときは、**-E**オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。上記のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべての変数を指定することもできます。

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

GUI

GUIを介してEasy Installを使用できます。VDAのデスクトップ環境で**/opt/Citrix/VDA/bin/easyinstall**コマンドを実行し、Easy Install GUIの指示に従います。



Easy Install GUIは、次の操作をガイドします。

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

> **ヒント:**

「**保存**」をクリックして、指定したパスのローカルファイルに変数設定を保存します。「**読み込み**」をクリックして、指定したファイルから変数設定を読み込みます。MCS変数の構成については、「ステップ3：マスターイメージの準備」を参照してください。

ステップ9：XDPingを実行する

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

ステップ10：Linux VDAを実行する

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx.service



sudo systemctl start ctxvda.service
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda.service



sudo systemctl stop ctxhdx.service
```

注:

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

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda.service



sudo systemctl restart ctxhdx.service



sudo systemctl start ctxvda.service
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda.service



sudo systemctl status ctxhdx.service
```

ステップ11：マシンカタログを作成する

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

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

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

注:

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

ヒント:

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

ステップ12：デリバリーグループを作成する

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

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

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

重要:

アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2407」を参照してください。

ステップ 13: Linux VDA をアップグレードする (オプション)

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

注:

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

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

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

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

注:

RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 で Linux VDA をアップグレードする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

SUSE ディストリビューションの場合:

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

Ubuntu/Debian ディストリビューションの場合:

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



sudo apt-get install -f
```

トラブルシューティング

このセクションの情報を使用して、簡易インストール機能の使用によって発生する可能性のある問題をトラブルシューティングします。

SSSD を使用したドメイン参加に失敗する

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

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

/var/log/xdl/vda.log:

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

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

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

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

-  The Active Directory provider (e.g. winbind daemon) service is running
```

```
and correctly configured.

-  Kerberos is correctly configured on this machine.

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

Error Details:

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

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

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

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

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

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

/var/log/messages:

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

この問題を解決するには:

1. `rm -f /etc/krb5.keytab`コマンドを実行します。
2. `net ads leave $REALM -U $domain-administrator`コマンドを実行します。
3. Delivery Controllerでマシンカタログとデリバリーグループを削除します。

4. /opt/Citrix/VDA/sbin/ctxinstall.shを実行します。
5. Delivery Controllerでマシンカタログとデリバリーグループを作成します。

Ubuntuデスクトップセッションでグレースクリーンが表示される

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

この問題を解決するには:

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

   `greeter-show-manual-login=true`

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

/var/log/xdl/hdx.log:

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



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



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



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

ヒント:

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

この問題を解決するには：

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

Ubuntuでセッションを起動できません(/en-us/linux-virtual-delivery-agent/2411/media/ubuntu-cannot-launch-session.png)

セッションが起動しない、またはdbusエラーでセッションがすぐに終了する

/var/log/messages（RHELまたはCentOSの場合）：

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



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



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



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



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

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


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

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

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


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


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


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


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

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

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

SELinuxによりSSHDがホームディレクトリにアクセスできない

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/xdl/ctxinstall.log：

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



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



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



If you want to allow polyinstantiation to enabled



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



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

```
    Do


       setsebool -P polyinstantiation_enabled 1


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


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


Then you should report this as a bug.


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


      Do


       allow this access for now by executing:


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


# semodule -i mypol.pp
```

この問題を解決するには：

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

   SELINUX=disabled (無効)

2. VDA を再起動します。

MCS を使用してドメイン非参加型 Linux VDA を作成する

この記事では、Machine Creation Services™ (MCS) を使用して、Citrix DaaS でドメイン非参加型 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型 VDA の作成は、Citrix DaaS™ のみでサポートされています。Citrix DaaS のお客様の場合:
  - コントロールプレーンは Citrix DaaS 上に展開する必要があります。
  - ドメイン非参加型 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。ドメイン非参加型 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。
  - ドメイン非参加型 VDA を作成するには、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 は、オンプレミスのハイパーバイザーにマシンをプロビジョニングする場合、または Workspace で Active Directory を ID プロバイダーとして使用する場合にのみ必要です。
- ドメイン非参加型 VDA を作成するには、MCS と簡易インストールの両方を使用できます。詳細については、「MCS を使用してドメイン非参加型 Linux VDA を作成する」および「簡易インストールを使用してドメイン非参加型 Linux VDA を作成する (プレビュー)」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。
- ドメイン非参加型 Linux VDA で利用できる機能は次のとおりです。
  - ドメイン非参加型 VDA で指定された属性を持つローカルユーザーを作成する
  - 非 SSO 認証
  - Azure Active Directory を使用した認証

- ランデブー V2

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

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

ステップ 1a: オンプレミスDelivery Controller用の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、Delivery Controllerに**XD MCS AHV Plugin**を選択してインストールします。

オンプレミスDelivery Controller用のNutanix AHVプラグイン(/en-us/linux-virtual-delivery-agent/2411/media/nutanix-plug-in-for-cvad.png)

ステップ 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のないリソースの場所を提供する場合でも、Citrix Cloud Connectorを登録する必要があります。

ステップ 1c: プラグインのインストール後に以下の手順を完了します

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

> **ヒント:**

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

ステップ 2: ホスト接続を作成する

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

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

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS**を選択します。
3. **Manage > Full Configuration**から、左側のペインで**Hosting**を選択します。
4. アクションバーで を選択します。
5. ウィザードは以下のページを案内します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、**Summary**ページに到達するまで**Next**を選択します。

ステップ 2a: 接続

接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2411/media/add-conn-connection.png)

「**接続**」ページで：

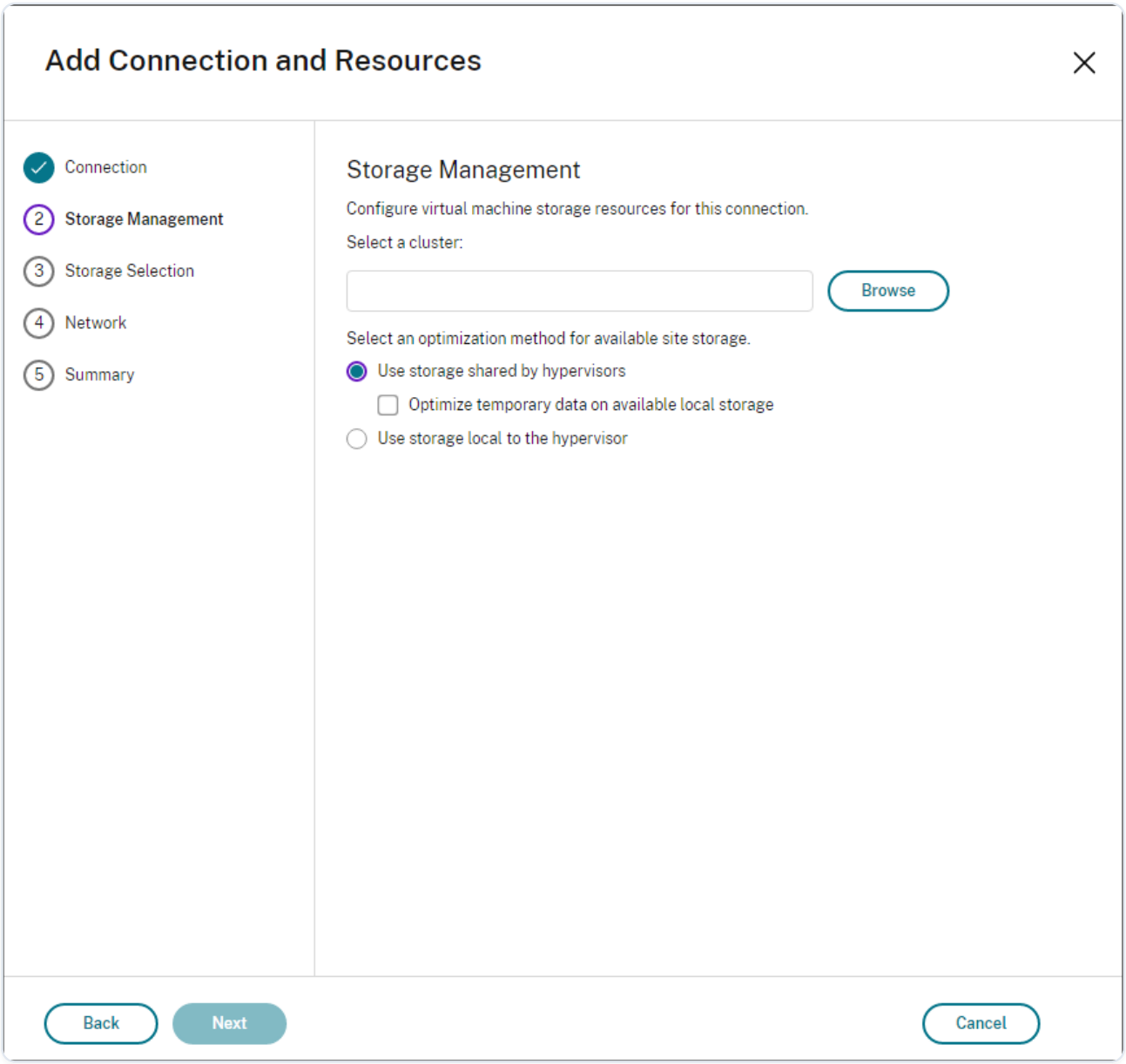
- 接続を作成するには、 を選択します。既存の接続と同じホスト構成に基づいて接続を作成するには、 を選択し、関連する接続を選択します。
- 「**ゾーン名**」フィールドでゾーンを選択します。オプションは、構成したすべてのリソースの場所です。
- 「**接続の種類**」フィールドでハイパーバイザーまたはクラウドサービスを選択します。オプションは、ゾーンにプラグインが適切にインストールされているハイパーバイザーとクラウド

サービスです。 または、PowerShellコマンド `Get-HypHypervisorPlugin -ZoneUid` を使用して、選択したゾーンで利用可能なハイパーバイザープラグインのリストを取得できます。

- 接続名を入力します。この名前は、**管理**表示に表示されます。
- 仮想マシンを作成するツールとして、Machine Creation Services または Citrix Provisioning を選択します。

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

ステップ 2b: ストレージ管理



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

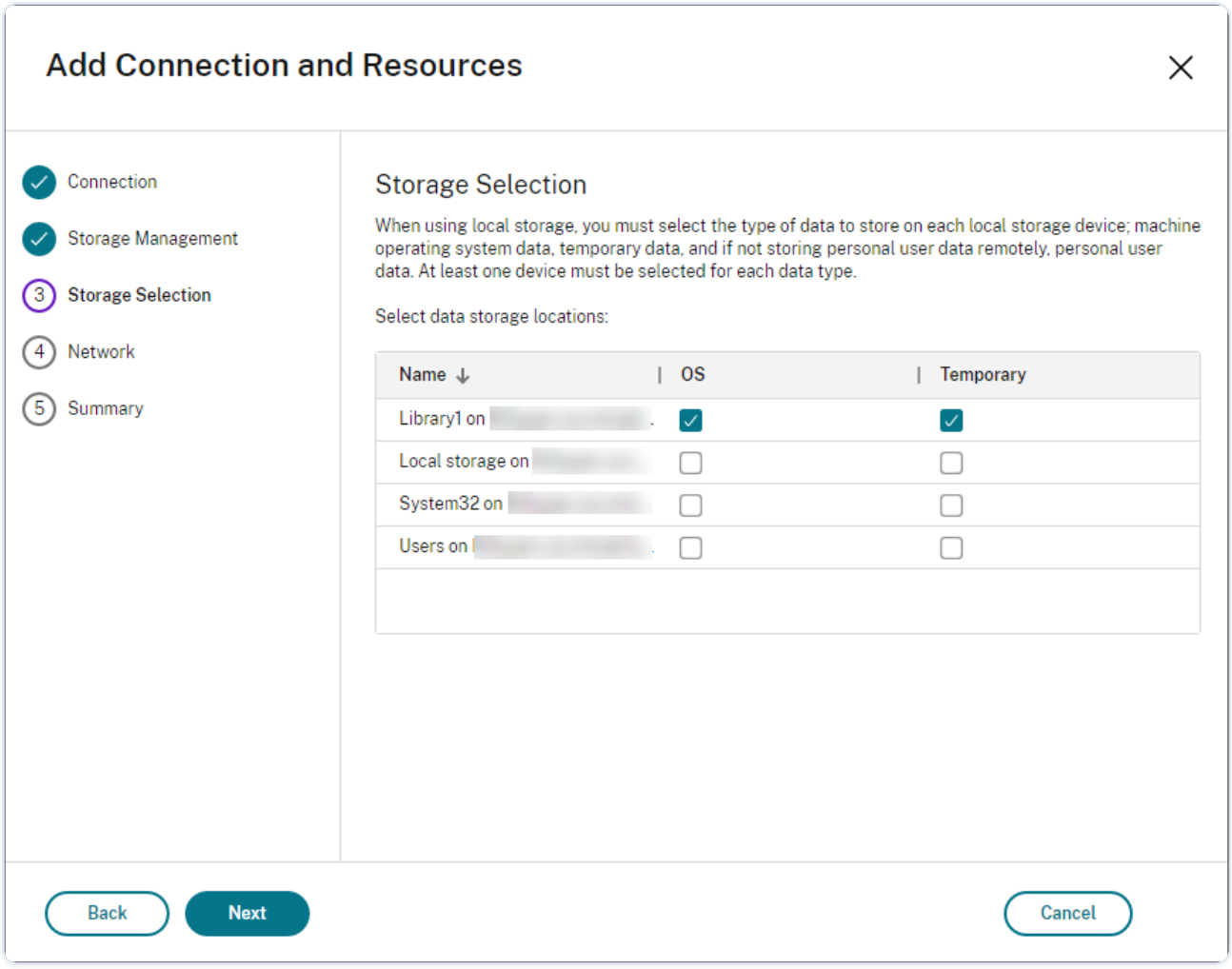
Hyper-V または VMware ホストへの接続を構成している場合は、クラスター名を参照して選択します。その他の接続タイプでは、クラスター名は要求されません。

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

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

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

ステップ 2c: ストレージの選択



ストレージの選択の詳細については、「ホストストレージ」を参照してください。

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

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

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

1. 使用するストレージデバイスを変更するには、**「選択」**をクリックします。
2. **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、**OK**を選択します。

ステップ 2d: リージョン

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

ステップ 2e: ネットワーク

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

一部の接続タイプ（Azure Resource Managerなど）では、VMが使用するサブネットも表示されます。1つまたは複数のサブネットを選択します。

ステップ2f: 概要

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

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

> **注:**

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

詳細については、「接続の作成と管理」を参照してください。

ステップ3: マスターイメージの準備

ヒント:

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

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

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

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

1. 使用中のハイパーバイザーのバージョンに基づいて、XenServerダウンロードページまたはCitrix Hypervisorダウンロードページからlinux用XenServer VM Toolsファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz`ファイルをLinux VMまたはLinux VMがアクセスできる共有ドライブにコピーします。
3. tar ファイルの内容を抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. Linux ディストリビューションに基づいて、`xe-guest-utilities` パッケージをインストールするには、次のコマンドを実行します。

RHEL/CentOS/Rocky Linux/SUSE の場合:

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

Ubuntu/Debian の場合:

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

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

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

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

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

- .NET Runtime に加えて、Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリビューションに .ASP.NET Core Runtime をインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。その他のディストリビューションにはバージョン 8 が必要です。
- 必要な .NET バージョンが Linux ディストリビューションに含まれている場合は、組み込みフィードからインストールします。そうでない場合は、Microsoft パッケージフィードから .NET をインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` を参照してください。

.NET のインストール後、Linux ディストリビューションに基づいて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:

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

注:

GCP でホストされている RHEL 8.x/9.x および Rocky Linux 8.x/9.x に Linux VDA をインストールした後、VM の再起動後にイーサネット接続が失われ、Linux VDA に到達できなくなることがあります。この問題を回避するには、VM を再起動する前に次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

Ubuntu/Debian の場合:

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



apt-get install -f
```

SUSE の場合:

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

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

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

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

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

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

```
sudo zypper install gcc

sudo zypper install make
```

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

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

4. ntfs-3g パッケージへのパスを入力します。

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

5. ntfs-3g をインストールします。

```
./configure

make

make install
```

ステップ 3e: (Ubuntu のみ) /etc/network/interfaces ファイルを編集する

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

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

ステップ 3f: (Ubuntu のみ) /etc/resolv.conf をポイントする

`/run/systemd/resolve/stub-resolv.conf` をポイントする代わりに、`/etc/resolv.conf` を `/run/systemd/resolve/resolv.conf` にポイントします。

```
unlink /etc/resolv.conf



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

Step 3g: 使用するデータベースを指定する

Linux VDAパッケージのインストール後、SQLiteとPostgreSQLを切り替えることができます。これを行うには、次の手順を完了します。

注:

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

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

```
# database configuration file for Linux VDA



## database choice

# possible choices are:
```

```
#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false
```

```
DbCustomizePostgreSQL=false



## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

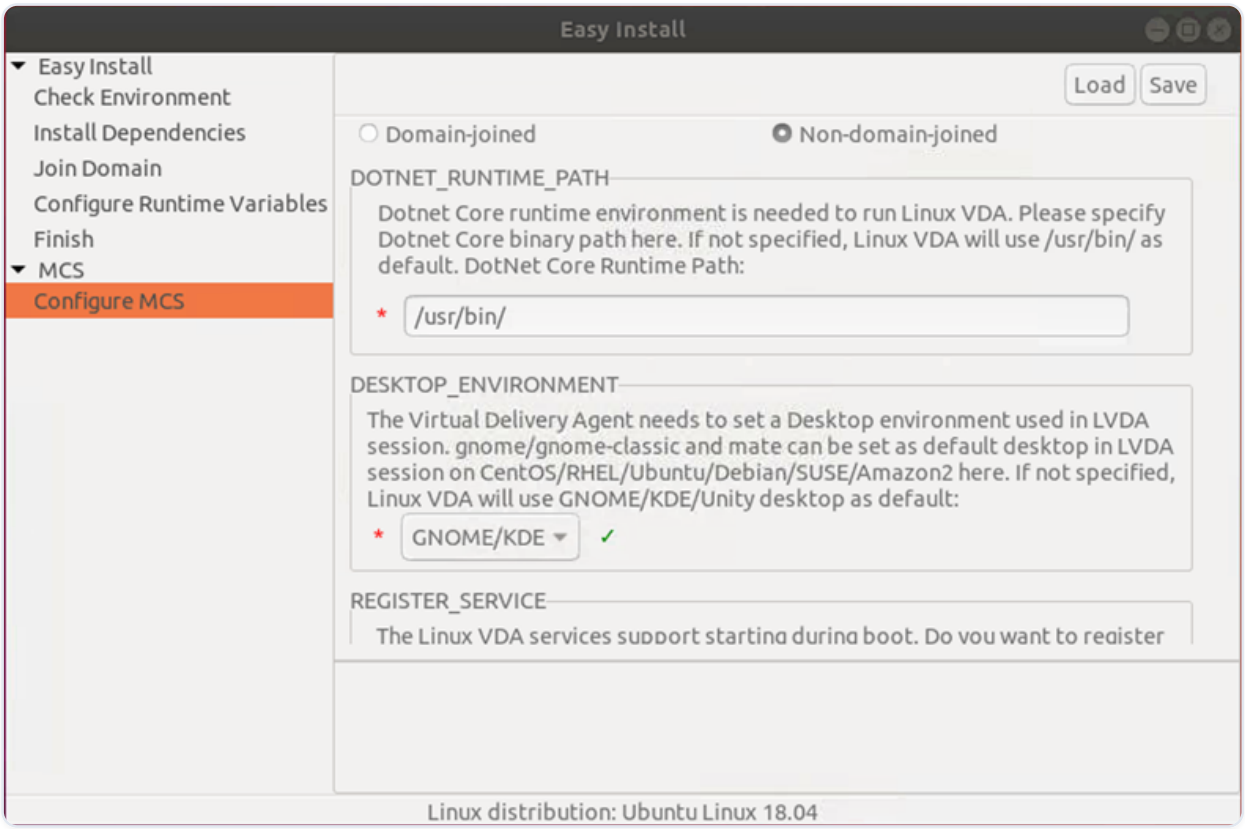
DbPostgreSQLServiceName="postgresql"
```

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

Step 3h: MCS変数を構成する

MCS変数を構成する方法は2つあります。

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



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

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

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

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

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

`START_SERVICE`=はい | いいえ

(オプション) ステップ 3i: MCS のレジストリ値を書き込むか更新する

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

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

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

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

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

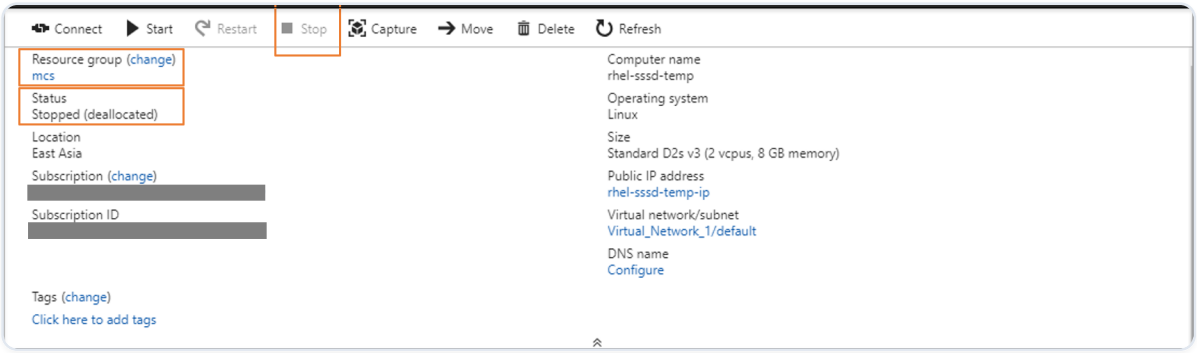
ステップ 3j: マスターイメージを作成する

1. `/etc/xdl/mcs/mcs.conf`を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、**Deploy**をク

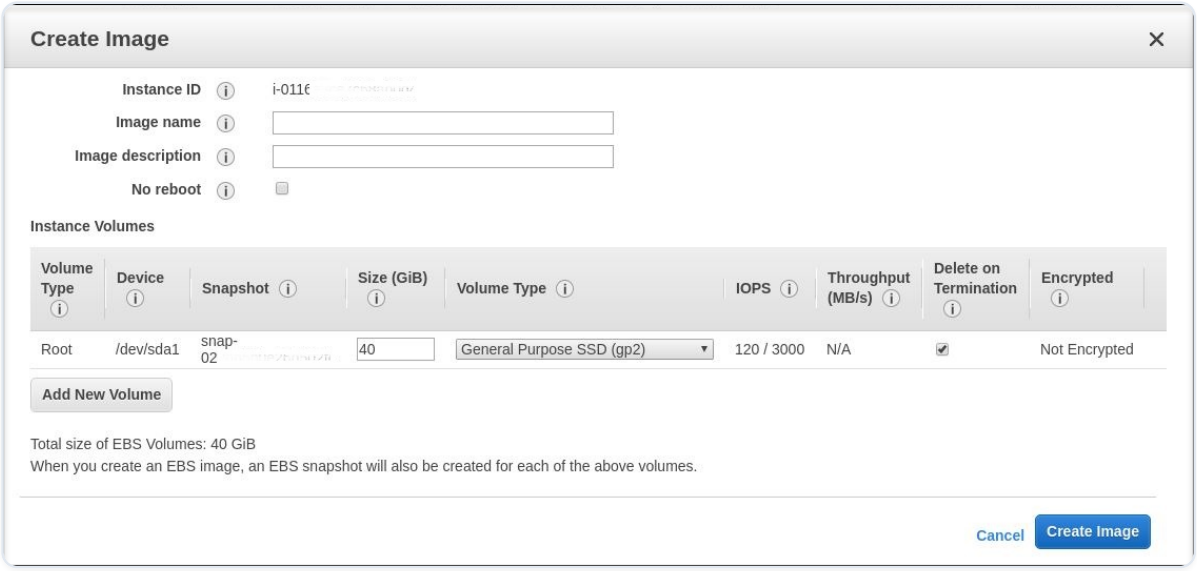
リックします。GUIで**Deploy**をクリックすると、GUIで設定した変数が`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。

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

   - **(XenServer (旧Citrix Hypervisor)、GCP、およびVMware vSphereの場合)** テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。
   - **(Azureの場合)** テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを**Stopped (deallocated)**と表示されていることを確認します。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるにはその名前が必要です。



   - **(AWSの場合)** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンスステータスが**Stopped**と表示されていることを確認します。テンプレートVMを右クリックし、**Image > Create Image**を選択します。必要な情報を入力し、設定を行います。**Create Image**をクリックします。



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

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

Step 4: マシンカタログを作成する

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

   Nutanix 固有の **Container** ページで、以前にテンプレートVMに指定したコンテナを選択します。

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

   Virtual Machines ページで、仮想CPUの数とvCPUあたりのコア数を確認します。マシン展開方法としてMCSを選択し、カタログで作成するマシンのIDとして**Non-domain-joined**を選択します。

   必要に応じてその他の構成タスクを実行します。詳細については、「マシンカタログの作成」を参照してください。

> **注:**
>
> Delivery Controller™ でのマシンカタログ作成プロセスにかなりの時間がかかる場合は、Nutanix Prism に移動し、**Preparation** というプレフィックスが付いたマシンを手動で起動します。この方法により、作成プロセスを続行できます。

Step 5: デリバリーグループを作成する

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

ドメインに参加していないLinux VDAを簡易インストールで作成する（プレビュー）

この記事では、簡易インストール方法を使用して、Citrix DaaSまたはCitrix Virtual Apps and Desktops™ 2407以降のバージョンで、ドメインに参加していないLinux VDAをマシンカタログに作成および登録する手順を説明します。

重要:

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

- 「Azure Active Directoryによる認証」
- ランデブー V2

ステップ1：空のマシンカタログを作成する

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

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

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

- オペレーティングシステムについては、以下を選択します。
  - ホスト型共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。
- 同じマシンカタログ内でドメイン参加マシンと非ドメイン参加マシンを混在させないでください。
- Remote PC Accessマシンカタログは、ドメイン参加マシンのみがサポートされており、非ドメイン参加マシンはサポートされていません。

注:

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

ステップ2：登録トークンを作成する

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

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

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

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

ステップ3：.NETのインストール

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

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

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

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

ステップ4：自己署名CA証明書の追加

自己署名CA証明書の設定の詳細については、「WebSocket用の自己署名証明書を構成する」を参照してください。このセクションでは、Linux VDAへのCA証明書の配置と更新について説明します。

- RHEL、Rocky Linux、およびAmazon Linux 2の場合：

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

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

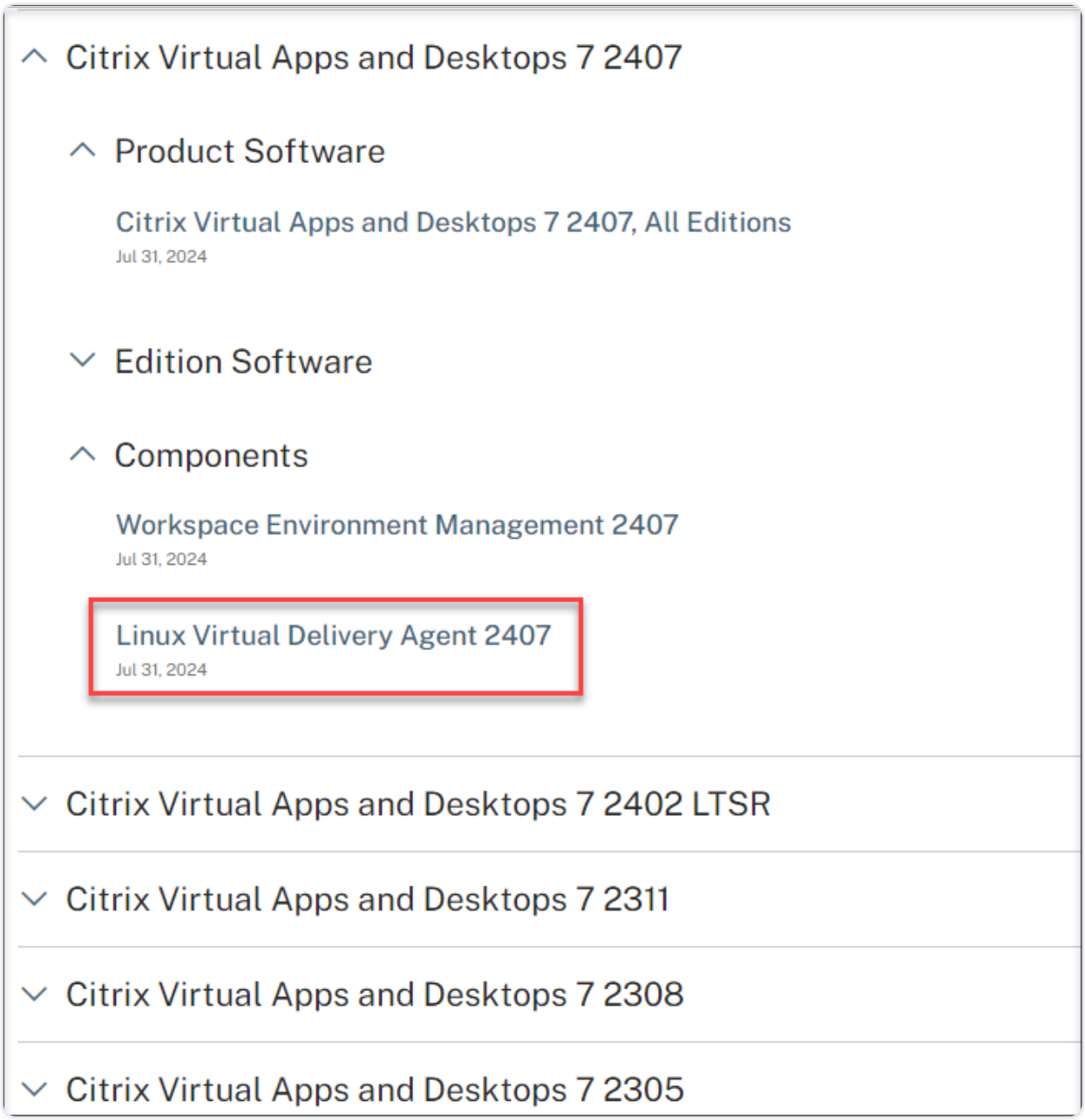
- SUSE、Ubuntu、およびDebianの場合：

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

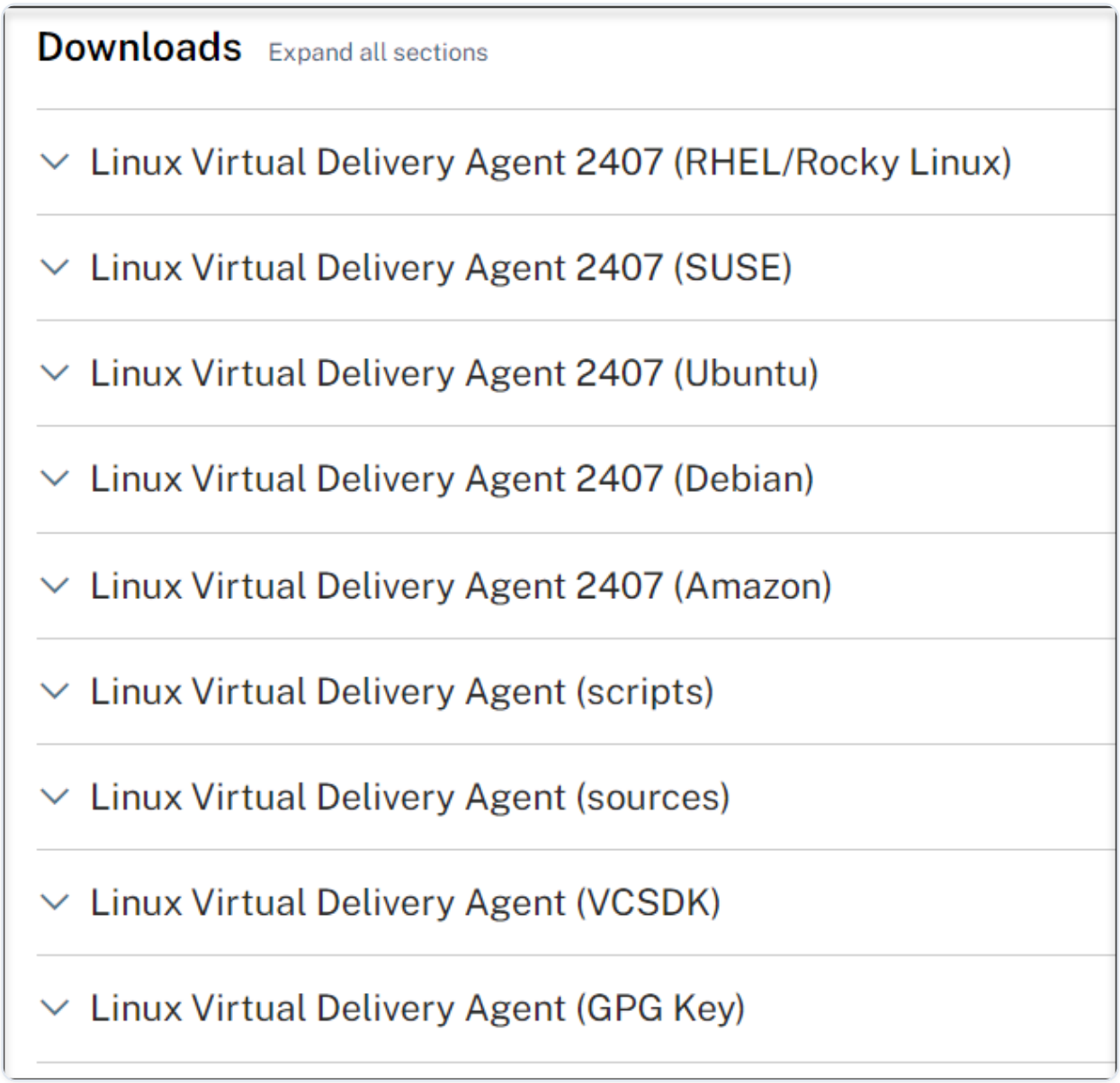
  ```
  sudo update-ca-certificates
  ```

ステップ5：Linux VDAパッケージのダウンロード

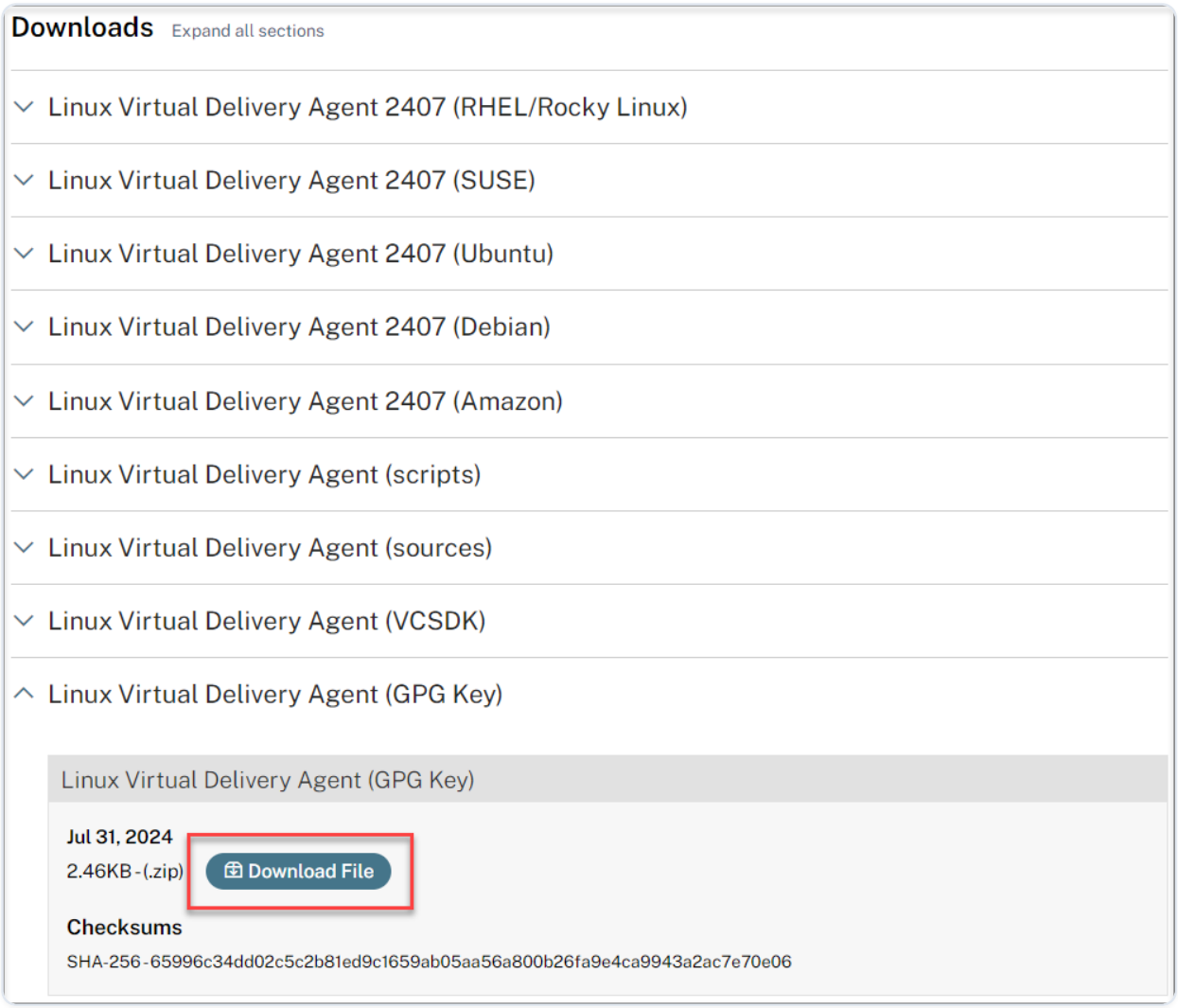
1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. Citrix Virtual Apps™ and Desktopsの適切なバージョンを展開します。
3. **Components**を展開してLinux VDAを見つけます。例：



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



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



公開鍵を使用してLinux VDAパッケージの整合性を検証するには：

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

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

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

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

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

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

手順6：Linux VDAパッケージをインストールする

Linux VDAの環境をセットアップするには、次のコマンドを実行します。

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

注:

- RHELおよびCentOSの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>`の指示を参照してください。
- RHEL 9.4/9.2およびRocky Linux 9.4/9.2にLinux VDAをインストールする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

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

注:

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAに到達できなくなることがあります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

Ubuntu/Debianディストリビューションの場合:

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

sudo apt-get install -f
```

> **注:**
>
> - Debian 11ディストリビューションに必要な依存関係をインストールするには、/etc/apt/sources.listファイルに`deb http://deb.debian.org/debian/ bullseye main`行を追加します。
> - GCP上のUbuntu 24.04、22.04、20.04では、RDNSを無効にします。そのためには、/etc/krb5.confの**の下にrdns = false**行を追加します。

> SUSEディストリビューションの場合:

1. AWS、Azure、GCP上のSUSE 15.6では、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こと。
   - **/etc/sysconfig/windowmanager**の**Default_WM**パラメーターが**"gnome"**に設定されていること。
2. Linux VDAをインストールするには、次のコマンドを実行します:

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

ステップ7: NVIDIA GRIDドライバーをインストールする

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

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

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

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

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

4. VMにゲストVMドライバー（クラウドベンダーまたはNVIDIAから提供されるもの）をインストールします。

ステップ8：使用するデータベースを指定する

Linux VDAパッケージのインストール後、**/etc/xdl/db.conf** を編集することで、使用するSQLiteまたはPostgreSQLを指定できます。

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

> **注:**
>
> - VDIモードでのみSQLiteを使用することをお勧めします。
> - 簡易インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用できます。**/etc/xdl/db.conf** で特に指定がない限り、Linux VDAはデフォルトでPostgreSQLを使用します。
> - また、**/etc/xdl/db.conf** を使用してPostgreSQLのポート番号を構成することもできます。

ステップ9：簡易インストールスクリプトを実行して環境とVDAを構成し、インストールを完了する

Linux VDAパッケージのインストール後、ctxinstall.shスクリプト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てください。OSに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、**/etc/locale.gen** ファイルを編集して **# en_US.UTF-8 UTF-8** の行のコメントを解除し、**sudo locale-gen** コマンドを実行します。

ctxinstall.sh

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

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

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

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

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

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

- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル (VDI) またはホスト型共有デスクトップ配信モデルとして構成するかどうか。HDX 3D Pro 環境の場合は、値を 'y' に設定してください。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDA は HDX 3D Pro をサポートしています。これは、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットです。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ (シングルセッション) モード用に構成されます (つまり、CTX_XDL_VDI_MODE='y' です)。
- **CTX_XDL_START_SERVICE='y\|n'** – 構成完了時に Linux VDA サービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシンの起動後に Linux Virtual Desktop サービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDAサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktopのシステムファイアウォールで、必要なポート（デフォルトではポート80および1494）を自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。この変数を指定しない場合、VDAで構成されているデフォルトのデスクトップが使用されます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、またはXfceデスクトップ環境を指定します。'<none\>'に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。CTX_XDL_DESKTOP_ENVIRONMENT変数を使用して、セッションで使用するデスクトップ環境を指定できます。コマンドを実行するか、システムトレイを使用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に.NETをインストールするパス。デフォルトのパスは「/usr/bin」です。
- **CTX_XDL_VDA_PORT='<port-number\>'** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。

対話モード

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

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

サイレントモード

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

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

```
export CTX_XDL_NON_DOMAIN_JOINED='y'

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

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

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

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

export CTX_XDL_VDA_PORT='<port-number>'

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

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

または、単一のコマンドを使用してすべての変数を指定することもできます。

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

ステップ10：XDPingを実行する

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

ステップ 11: Linux VDA を実行する

Linux VDA を起動する:

Linux VDA サービスを起動するには:

```
sudo systemctl start ctxhdx.service



sudo systemctl start ctxvda.service
```

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo systemctl stop ctxvda.service



sudo systemctl stop ctxhdx.service
```

> **注:**

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

> **Linux VDA を再起動する:**

Linux VDA サービスを再起動するには:

```
sudo systemctl stop ctxvda.service


sudo systemctl restart ctxhdx.service


sudo systemctl start ctxvda.service
```

Linux VDA のステータスを確認する:

Linux VDA サービスの実行ステータスを確認するには:

```
sudo systemctl status ctxvda.service


sudo systemctl status ctxhdx.service
```

ステップ 12: デリバリーグループを作成する

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

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

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

重要:

アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。

> マシンカタログとデリバリーグループの作成方法については、Citrix Virtual Apps and Desktops 7 2407を参照してください。

ステップ13：ローカルアカウントマッピングを有効にする

ローカルアカウントマッピングが有効になっていない場合

ドメインに参加していないVDAでホストされているセッションは、ログオン時に失敗し、パスワードプロンプトが表示されますが、正しいパスワードを受け入れません。次のようなエラーが`hdx.log`で見つかります。

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

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

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

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

ローカルアカウントマッピングを有効にする

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

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

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

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

MCSを使用して、ドメイン参加済みおよびドメイン非参加のVDAを作成できます。Citrix DaaSでドメイン非参加のLinux VDAを作成する場合は、専用の記事「ドメイン非参加のLinux VDAをMCSで作成する」(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-non-domain-joined-linux-vdas-using-mcs.html)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GUIの**AD_INTEGRATION**変数には、デフォルト値がなくなりました。必要に応じて値を設定する必要があります。詳細については、この記事の「手順3i: MCS変数を構成する」(#step-3i-configure-mcs-variables)セクションを参照してください。
- /etc/xdl/mcs/mcs.confの**UPDATE_MACHINE_PW**エントリの有効な値は、**enabled**または**disabled**ではなく、**Y**または**N**になりました。詳細については、この記事の「マシンアカウントパスワードの自動更新」(#automate-machine-account-password-updates)セクションを参照してください。

サポートされているディストリビューション

	ウィンバインド	SSSD	セントリファイ	PBIS
デビアン 12.7/12.5/11.11	はい	はい	いいえ	はい
RHEL 9.4/9.2	はい	はい	はい	いいえ
RHEL 8.10/8.8	はい	はい	はい	はい
ロッキーリナックス 9.4/9.2	はい	はい	はい	いいえ
ロッキーリナックス 8.10/8.8	はい	はい	はい	いいえ

	ウィンバインド	SSSD	セントリファイ	PBIS
SUSE 15.6	はい	はい	いいえ	はい
ウブントゥ 24.04	はい	はい	いいえ	はい
ウブントゥ 22.04/20.04	はい	はい	いいえ	はい

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

LINUXディストリビューション	CENTRIFYバージョン
RHEL 7/8	5.8.0
SUSE	5.7.1
デビアン、ウブントゥ	5.6.1

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

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

  - テンプレートマシンで、`/etc/xdl/mcs/mcs.conf`ファイルに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、そのすべての下位のMCSで作成されたマシンに対して書き込み権限とパスワードリセット権限を持つ組織単位 (OU) を作成します。
  - `/opt/Citrix/VDA/sbin/deploymcs.sh` の実行が完了した後、MCSで作成されたマシンを再起動する前に、デプロイメントに応じてDelivery ControllerまたはCitrix Cloud Connectorで `klist -li 0x3e4 purge` を実行します。

- SSSDを使用してドメインに接続されている、現在実行中のRHEL 8.x/9.xまたはRocky Linux 8.x/9.x VDAをMCSのテンプレートVMとして使用するには、以下を確認してください。

- VDAは手動でインストールされており、簡易インストールは使用されていません。簡易インストールはRHEL 8.x/9.xおよびRocky Linux 8.x/9.xで **Adcli** を使用し、MCSはSSSDと **Adcli** の組み合わせをサポートしていません。
- SambaサーバーがAD認証にSSSDを使用するように構成されていること。詳細については、`<https://access.redhat.com/solutions/3802321>` のRed Hatの記事を参照してください。

サポートされているハイパーバイザー

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

サポートされてい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、予期しない結果が発生する可能性があります。

MCSを使用してLinux VMを作成する

考慮事項

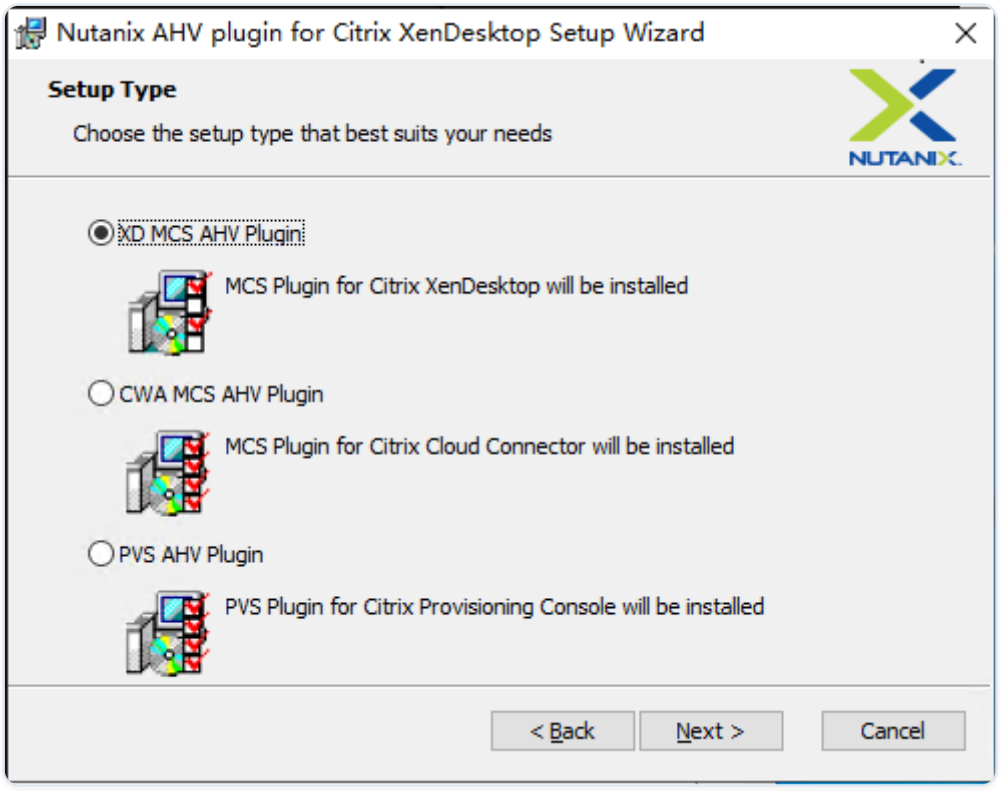
- 2203リリース以降、Citrix Virtual Apps and Desktops™およびCitrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal Hybrid Multi-Cloud (HMC) ライセンスが必要です。
- ベアメタルサーバーは、MCSを使用して仮想マシンを作成する目的ではサポートさ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

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

手順1a：オンプレミスのDelivery Controller用Nutanix AHVプラグインのインストールと登録

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



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

Citrix Cloud™ Connectors 用の **CWA MCS AHV Plugin** を選択してインストールします。Citrix Cloud テナントに登録されているすべての Citrix Cloud Connectors にプラグインをインストールします。AHV を持たないリソースロケーションを提供する場合でも、Citrix Cloud Connectors を登録する必要があります。

ステップ 1c: プラグインのインストール後に次の手順を完了します

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

> ヒント:

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

ステップ 2: ホスト接続を作成する

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

> **注:**
>
> オンプレミスの Delivery Controllers の場合、オンプレミスの Citrix Studio で **> >** を選択してホスト接続を作成します。クラウド Delivery Controllers の場合、Citrix Cloud の Web ベースの Studio コンソールで **> >** を選択してホスト接続を作成します。
>
> 詳細については、Citrix Virtual Apps and Desktops ドキュメントの 接続とリソースの作成と管理 および Citrix DaaS ドキュメントの 接続の作成と管理 を参照してください。

- Citrix Studio で Azure へのホスト接続を作成する
- Citrix Studio で AWS へのホスト接続を作成する
- Citrix Studio で XenServer へのホスト接続を作成する
- Citrix Studio で GCP へのホスト接続を作成する
- Citrix Studio で Nutanix へのホスト接続を作成する
- Citrix StudioでVMwareへのホスト接続を作成する(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-vmware-in-citrix-studio)

Citrix StudioでAzureへのホスト接続を作成する

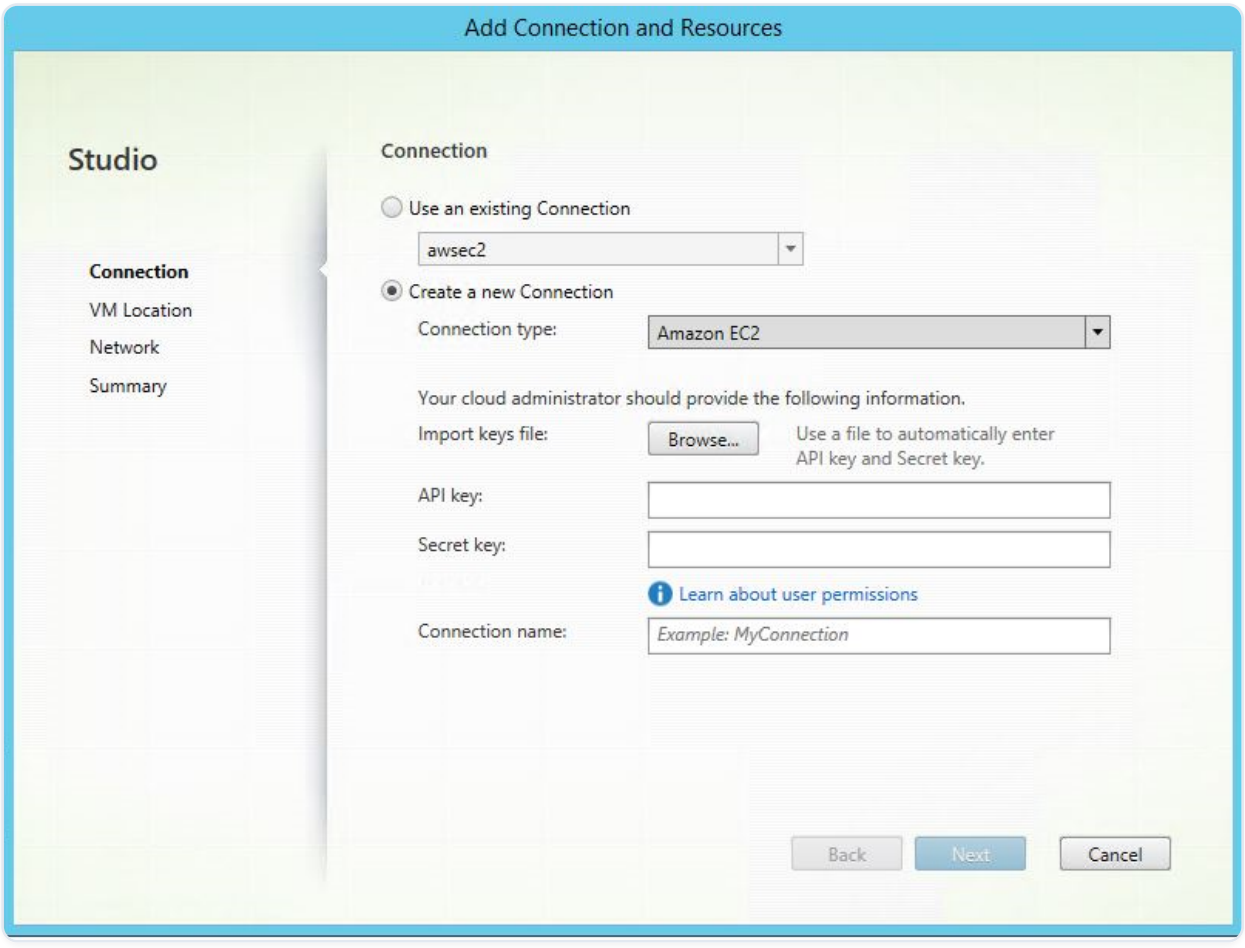
1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**［構成］>［ホスティング］>［接続とリソースの追加］を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**［管理］>［ホスティング］>［接続とリソースの追加］**を選択してホスト接続を作成します。
2. **［接続とリソースの追加］**ウィザードで、接続の種類としてMicrosoft Azureを選択します。
3. 接続の種類としてMicrosoft Azureを選択します。

4. ウィザードの指示に従ってページを進めます。特定のページの内容は、選択した接続の種類によって異なります。各ページの完了後、**［次へ］**を選択して**［概要］**ページに進みます。詳しくは、記事「MCSを使用してドメインに参加していないLinux VDAを作成する」の「手順2：ホスト接続を作成する」**を参照してください。

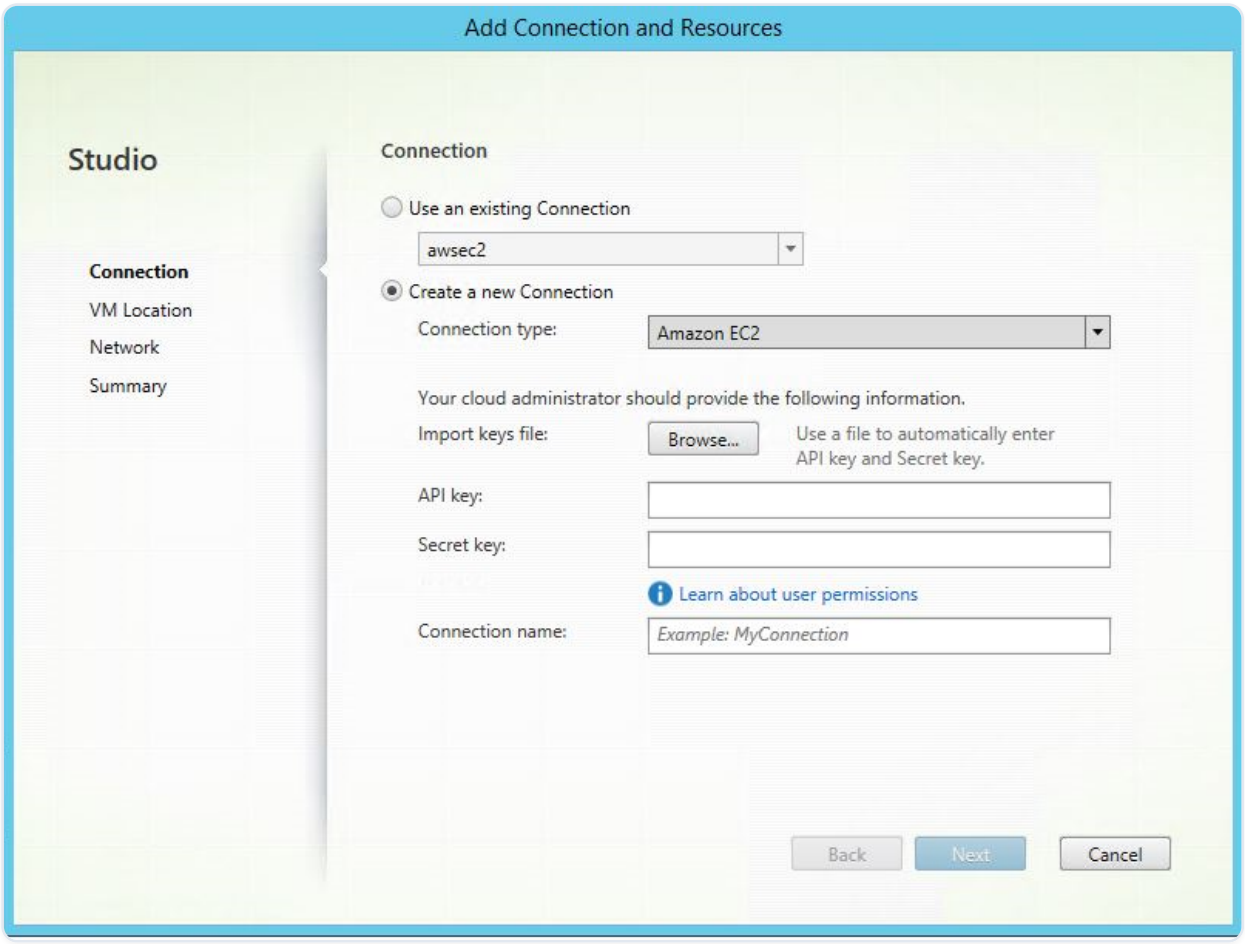
Citrix StudioでAWSへのホスト接続を作成する

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



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



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

a. AWSサービスにサインインします。

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

c. 対象ユーザーを選択し、下にスクロールして タブを選択します。

d. 下にスクロールして をクリックします。新しいウィンドウが表示されます。

e. をクリックし、アクセスキーを安全な場所に保存します。

4. ウィザードがページを案内します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、 を選択して ページに到達します。

Citrix StudioでXenServerへのホスト接続を作成する

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

2. ウィザードで、 フィールドでXenServer（旧Citrix Hypervisor）を選択します。
3. 接続アドレス（XenServer URL）と資格情報を入力します。
4. 接続名を入力します。

Citrix StudioでGCPへのホスト接続を作成する

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

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

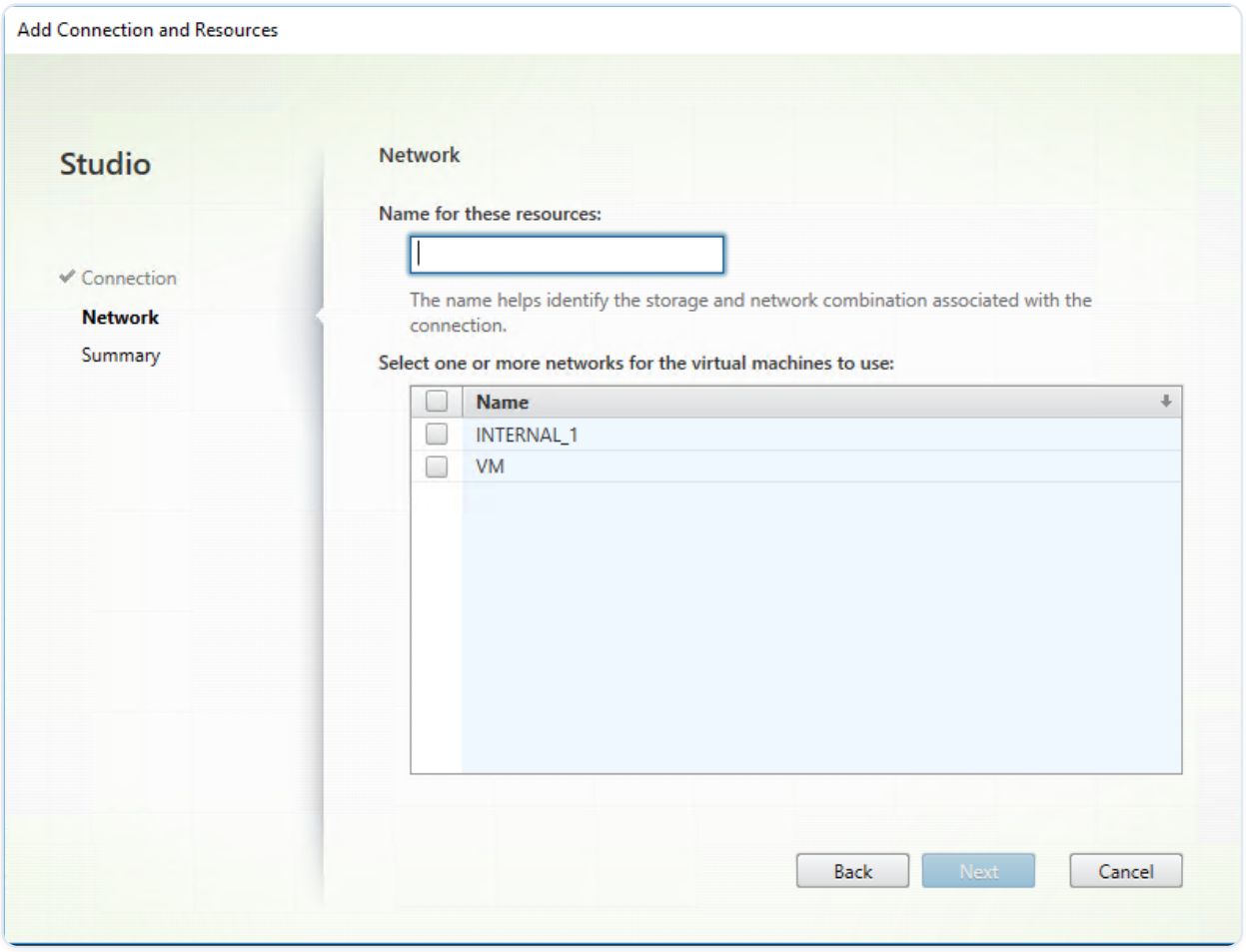
   例：Citrix CloudのWebベースのStudioコンソールで：

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2411/media/add-conn-connection.png)
3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードがページを案内します。特定のページコンテンツは、選択した接続タイプによって異なります。各ページを完了したら、**Summary**ページに到達するまで**Next**を選択します。詳細については、Create non-domain-joined Linux VDAs using MCSの記事の「**手順2：ホスト接続を作成する**」を参照してください。

Citrix StudioでNutanixへのホスト接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
2. **Add Connection and Resources**ウィザードの**Connection**ページで、接続タイプとしてNutanix AHVを選択し、ハイパーバイザーアドレス、資格情報、および接続名を指定します。**Network**ページで、ユニットのネットワークを選択します。

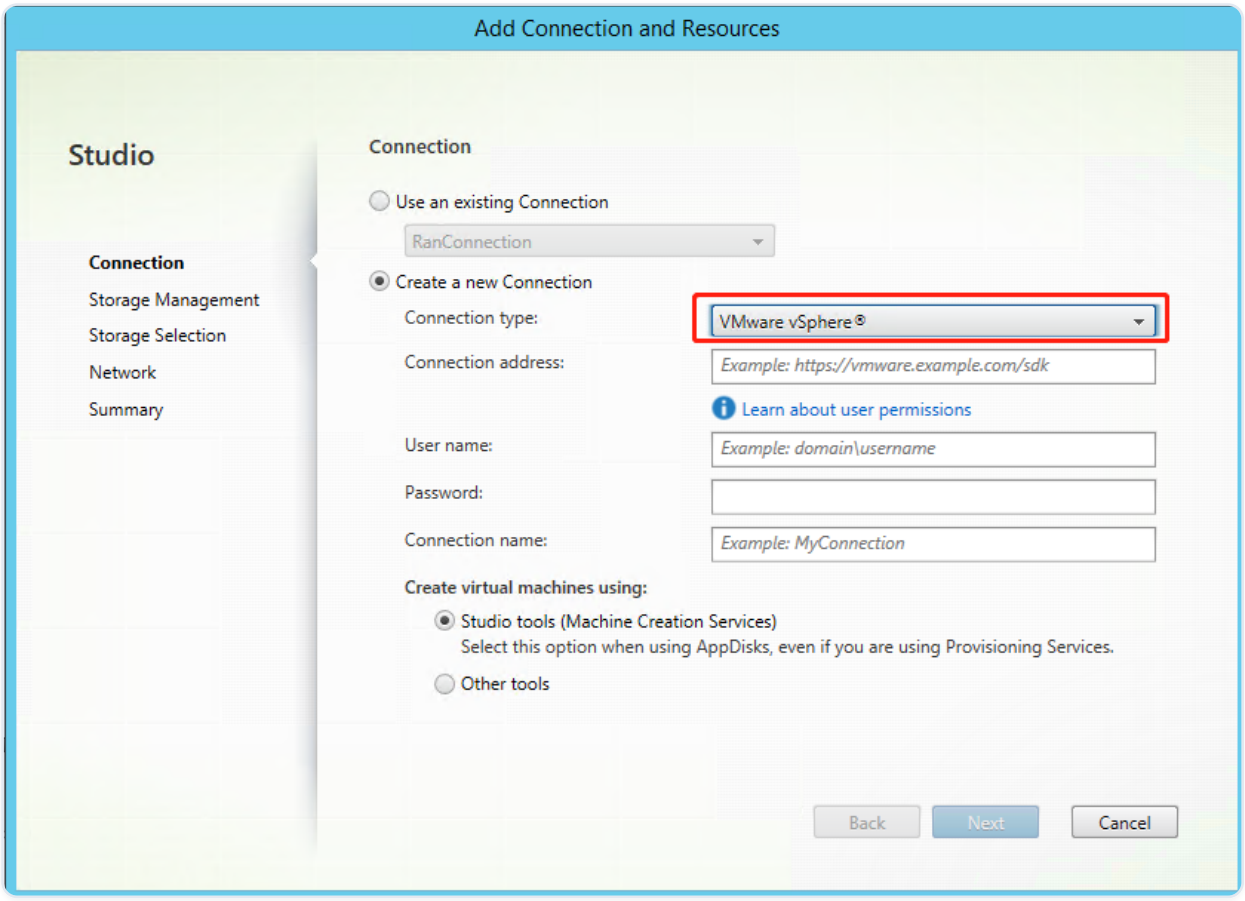
   例：オンプレミスのCitrix Studioの場合：



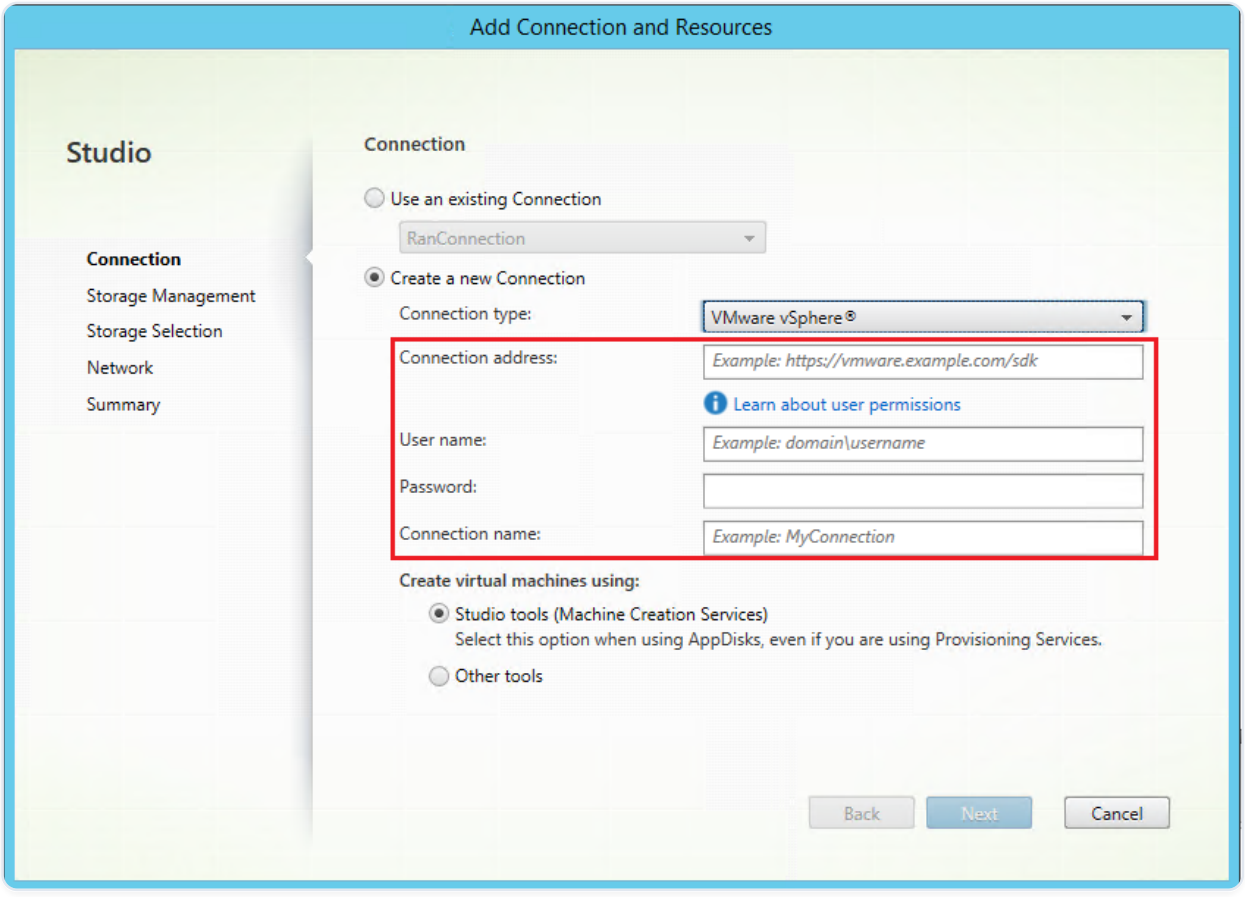
Citrix StudioでVMwareへのホスト接続を作成する

1. vSphere環境にvCenter Serverをインストールします。詳細については、VMware vSphereを参照してください。
2. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**Configuration > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**Manage > Hosting > Add Connection and Resources**を選択してホスト接続を作成します。
3. 接続タイプとしてVMware vSphereを選択します。

   例：オンプレミスのCitrix Studioの場合：



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



手順3：マスターイメージを準備する

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

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

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

1. 使用中のハイパーバイザーのバージョンに基づいて、XenServer Downloads pageまたはCitrix Hypervisor Downloads pageからLinux用XenServer VM Toolsファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz`ファイルをLinux VM、またはLinux VMがアクセスできる共有ドライブにコピーします。
3. tarファイルの内容を抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. お使いのLinuxディストリビューションに基づいて、`xe-guest-utilities`パッケージをインストールするには、次のコマンドを実行します。

   RHEL/CentOS/Rocky Linux/SUSEの場合:

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

   Ubuntu/Debianの場合:

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

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

ステップ 3b: AWS、Azure、GCP上のSUSE 15.6の構成を確認する

AWS、Azure、GCP 上の SUSE 15.6 の場合、以下を確認してください。

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

ステップ 3c: GCP 上の Ubuntu 20.04 で RDNS を無効にする

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

ステップ 3d: テンプレート VM に .NET をインストールする

> **注:**
>
> 現在実行中の VDA をテンプレート VM として使用する場合は、この手順をスキップしてください。SSSD を使用してドメインに接続されている現在実行中の RHEL 8.x/9.x または Rocky Linux 8.x/9.x VDA をテンプレート VM として使用する場合は、以下を確認してください。
>
> - VDA は手動でインストールされており、簡易インストールを使用していないこと。簡易インストールは RHEL 8.x/9.x および Rocky Linux 8.x/9.x に **Adcli** を使用しますが、SSSD と **Adcli** の組み合わせは MCS でサポートされていません。
> - Samba サーバーが AD 認証に SSSD を使用するように構成されていること。詳細については、`<https://access.redhat.com/solutions/3802321>` の Red Hat の記事を参照してください。

> Linux VDA パッケージをインストールする前に、テンプレート VM に .NET をインストールし、以下の点に注意してください。

- .NET Runtime に加えて、Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリビューションに .ASP.NET Core Runtime をインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。その他のディストリビューションにはバージョン 8 が必要です。
- お使いの Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みのフィードからインストールしてください。そうでない場合は、Microsoft パッケージフィードから .NET をインストールしてください。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` を参照してください。

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

.NET のインストール後、お使いの Linux ディストリビューションに基づいて以下のコマンドを実行し、Linux VDA をインストールします。

RHEL/CentOS/Rocky Linux の場合:

> **注:**
>
> RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 に Linux VDA をインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

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

Ubuntu/Debian の場合:

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



apt-get install -f
```

SUSE の場合:

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

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

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

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

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

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

```
sudo zypper install gcc

sudo zypper install make
```

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

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

4. ntfs-3g パッケージへのパスを入力します。

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

5. ntfs-3g をインストールします。

```
./configure

make

make install
```

ステップ 3h: 使用するデータベースを指定する

Linux VDA パッケージのインストール後に、SQLite と PostgreSQL を切り替えることができます。これを行うには、次の手順を完了します。

注:

- VDI モードでは SQLite を使用し、ホスト型共有デスクトップ配信モデルでは PostgreSQL を使用することをお勧めします。
- 簡易インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で特に指定がない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションで提供されているバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように`/etc/xdl/db.conf`を編集し、簡易インストールスクリプト（`ctxinstall.sh`）または

> MCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。
>
> - **/etc/xdl/db.conf**を使用して、PostgreSQLのポート番号を構成することもできます。

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

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432
```

```
## PostgreSQL customized
# only the following value means true, otherwise false:
#     true
#     yes
#     y
#     YES
#     Y
# default is false
DbCustomizePostgreSQL=false

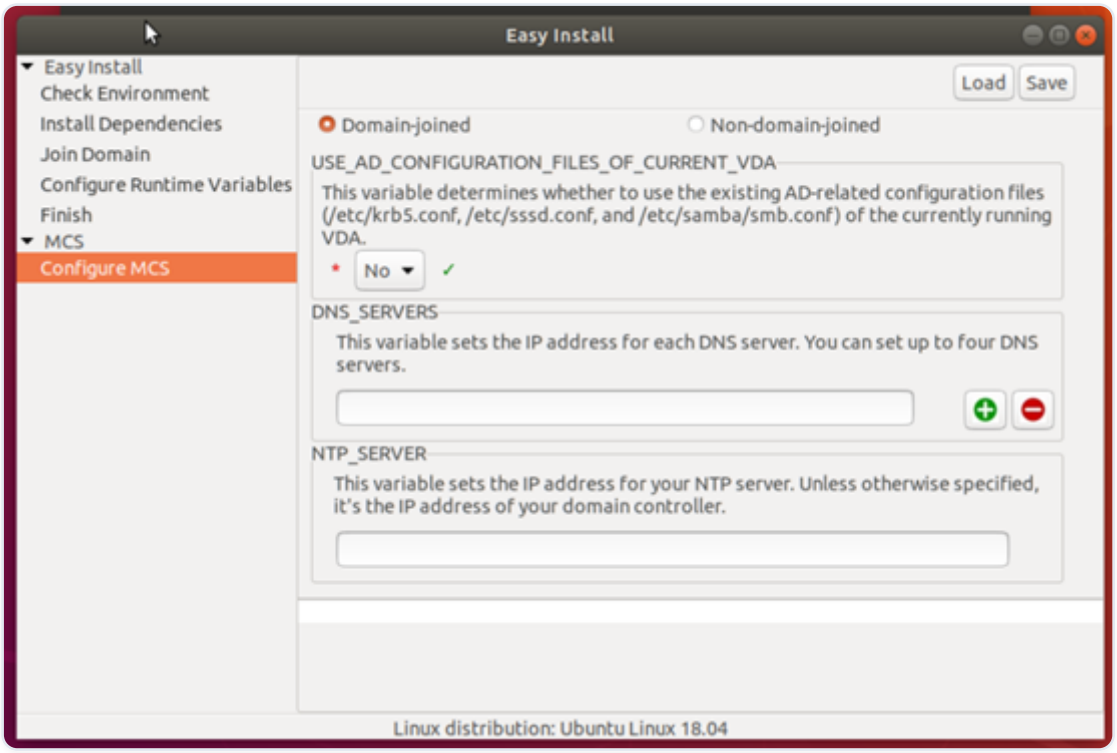
## PostgreSQL service name
# specify the service name of PostgreSQL for Linux VDA
# default is "postgresql"
DbPostgreSQLServiceName="postgresql"
```

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

手順3i：MCS変数を構成する

MCS変数を構成する方法は2つあります。

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



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

以下は、ドメイン非参加シナリオおよびドメイン参加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`=.NETランタイムのインストールパス

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

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

  `START_SERVICE`=Y | N

- **ドメイン参加シナリオの場合**

  - `Use_AD_Configuration_Files_Of_Current_VDA`: 現在実行中のVDAの既存のAD関連構成ファイル（/etc/krb5.conf、/etc/sssd.conf、および/etc/samba/smb.conf）を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実

行中のVDAの対応するファイルと同じになります。ただし、`dns`および`AD_INTEGRATION`変数を構成する必要があります。デフォルト値はNです。これは、マスターイメージ上の構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。現在実行中のVDAをテンプレートVMとして使用するには、値をYに設定します。それ以外の場合は、Nに設定します。

- `dns`: 各DNSサーバーのIPアドレスを設定します。最大4つのDNSサーバーを設定できます。
- `NTP_SERVER`: NTPサーバーのIPアドレスを設定します。特に指定がない限り、ドメインコントローラーのIPアドレスです。
- `WORKGROUP`: ADで構成したNetBIOS名（大文字と小文字を区別）にワークグループ名を設定します。設定しない場合、MCSはマシンホスト名の直後にあるドメイン名の一部をワークグループ名として使用します。たとえば、マシンアカウントが**user1.lvda.citrix.com**の場合、MCSはワークグループ名として**lvda**を使用しますが、**citrix**が正しい選択です。ワークグループ名が正しく設定されていることを確認してください。
- `AD_INTEGRATION`: SSSD、Winbind、PBIS、またはCentrifyを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされるディストリビューションを参照してください。
- `CENTRIFY_DOWNLOAD_PATH`: Server Suite Free（旧Centrify Express）パッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。
- `CENTRIFY_SAMBA_DOWNLOAD_PATH`: Centrify Sambaパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。
- `PBIS_DOWNLOAD_PATH`: PBISパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をPBISに設定した場合にのみ有効になります。
- `UPDATE_MACHINE_PW`: マシンアカウントのパスワード更新の自動化を有効または無効にします。詳細については、マシンアカウントのパスワード更新を自動化するを参照してください。
- Linux VDA構成変数:

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

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

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

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

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

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

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

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

(オプション) ステップ3j: MCSのレジストリ値を書き込むか更新する

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

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

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

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

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

注

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

ステップ 3k: マスターイメージを作成する

1. （SSSD + RHEL 8.x/9.x または Rocky Linux 8.x/9.x のみ）`update-crypto-policies --set DEFAULT:AD-SUPPORT`コマンドを実行し、テンプレートVMを再起動します。

2. `/etc/xdl/mcs/mcs.conf`を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、**Deploy**をクリックします。GUIで**Deploy**をクリックすると、GUIで設定した変数が`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。
3. （現在実行中のVDAをテンプレートVMとして使用している場合、またはドメインに参加していないシナリオの場合は、この手順をスキップしてください。）テンプレートVMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する`/etc/krb5.conf`、`/etc/samba/smb.conf`、および`/etc/sssd/sssd.conf`ファイルをカスタマイズします。

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

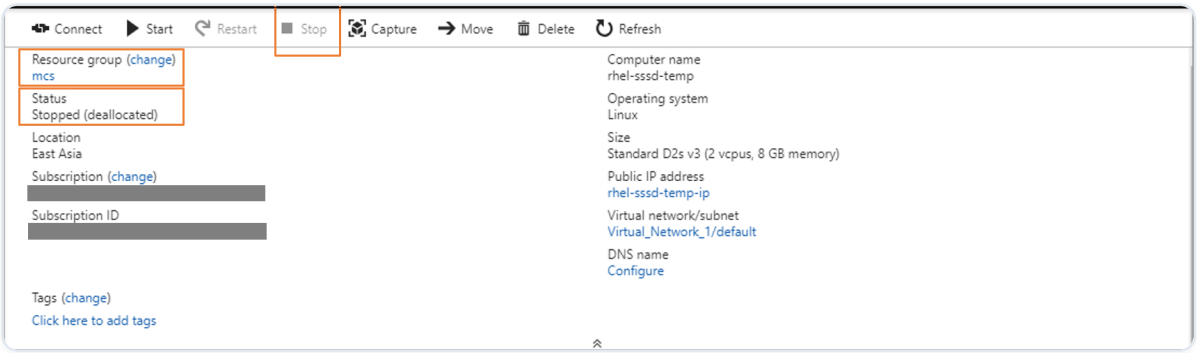
   SSSDユーザーの場合は、`/etc/xdl/ad_join/sssd.conf.tmpl`、`/etc/xdl/ad_join/sssd_krb5.conf.tmpl`、および`/etc/xdl/ad_join/sssd_smb.conf.tmpl`テンプレートを更新します。

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

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

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

   - **（XenServer、GCP、およびVMware vSphereの場合）** テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。
   - **（Azureの場合）** テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを VMをシャットダウンします。テンプレートVMの電源状態が**停止済み（割り当て解除済み）**であることを確認します。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるには、この名前が必要です。



- **（AWSの場合）** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**であることを確認します。テンプレートVMを右クリックし、**Image > Create Image**を選択します。必要な情報を入力し、設定を行います。**Create Image**をクリックします。

  EBSイメージの作成(/en-us/linux-virtual-delivery-agent/2411/media/create-ebs-image.png)

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

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

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

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAに到達できなくなる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認してください。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```

ステップ4: マシンカタログを作成する

Citrix StudioまたはWeb Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成するときは、マスターイメージを選択し、次の点を考慮してください。

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

  デスクトップエクスペリエンス(/en-us/linux-virtual-delivery-agent/2411/media/machine-catalog-setup.png)

  デスクトップエクスペリエンスページで、次のいずれかを選択します。

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

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

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

  - ユーザーの変更をローカルディスク上のデスクトップに保存します（永続的）。
  - ユーザーがログオフしたときにユーザーの変更を破棄し、仮想デスクトップをクリアします（非永続的）。ユーザーパーソナライゼーションレイヤーを使用している場合は、このオプションを選択します。
- 永続マシンを含むMCSカタログのマスターイメージを更新する場合、カタログに追加された新しいマシンは、更新されたイメージを使用します。既存のマシンは、元のマスターイメージを引き続き使用します。

詳細については、Citrix Virtual Apps and Desktops ドキュメントおよび Citrix DaaS ドキュメントの「マシンカタログの作成」を参照してください。

> **注:**

Nutanix環境では、Delivery Controller™でのマシンカタログ作成プロセスにかなりの時間がかかる場合、Nutanix Prismに移動し、**Preparation**というプレフィックスが付いたマシンを手動で起動してください。この方法により、作成プロセスを続行できます。

ステップ5：デリバリーグループを作成する

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

詳細については、Citrix Virtual Apps and Desktops ドキュメントおよび Citrix DaaS ドキュメントの「デリバリーグループの作成」を参照してください。

注:

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

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

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

                   join_domain_samba

                   restart_service winbind /usr/bin/systemctl
   ```

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

MCSを使用してLinux VDAをアップグレードする

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

1. Linux VDAを現在のリリースにアップグレードする前に、.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を除くすべてのサポートされている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、引き続き .NET Runtime 6.0 をインストールします。

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

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:**
>
> - 自動ソフトウェア更新をスケジュールするには、Azure を介した Linux VDA の自己更新 機能を使用することもできます。この目的を達成するには、テンプレートマシンの etc/xdl/mcs/mcs_local_setting.reg ファイルにコマンドラインを追加します。たとえば、次のコマンドラインを追加できます。
>
>   ```
>   create -k
>   "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t
>   "REG_DWORD" -v "fEnabled" -d "0x00000001" --force
>
>
>
>   create -k
>   "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t
>   "REG_SZ" -v "ScheduledTime" -d "Immediately" --force
>
>
>
>   create -k
>   "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t
>   "REG_SZ" -v "Url" -d "<Your-Azure-Container-Url>" –force
>   ```

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

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

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

RHEL および Rocky Linux ディストリビューションの場合:

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

注:

RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 で Linux VDA をアップグレードする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

SUSE ディストリビューションの場合:

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

Ubuntu/Debian ディストリビューションの場合:

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



sudo apt-get install -f
```

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

マシンアカウントのパスワード更新を自動化する

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

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

   `UPDATE_MACHINE_PW="Y"`

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

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

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

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

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

重要:

2212リリース以降の重要な変更点は次のとおりです。

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

サポートされているディストリビューション

	ウィンバインド	SSSD	PBIS
デビアン 12.7/12.5/11.11	はい	はい	はい
RHEL 9.4/9.2	はい	いいえ	いいえ
RHEL 8.x	はい	いいえ	はい
ロッキー Linux 9.4/9.2	はい	いいえ	いいえ
ロッキー Linux 8.x	はい	いいえ	いいえ
ウブントゥ 24.04/22.04/20.04	はい	いいえ	いいえ
SUSE 15.6	はい	いいえ	いいえ

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

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

サポートされているハイパーバイザー

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

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

MCSを使用してFASが有効なLinux VDAを作成する

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

考慮事項

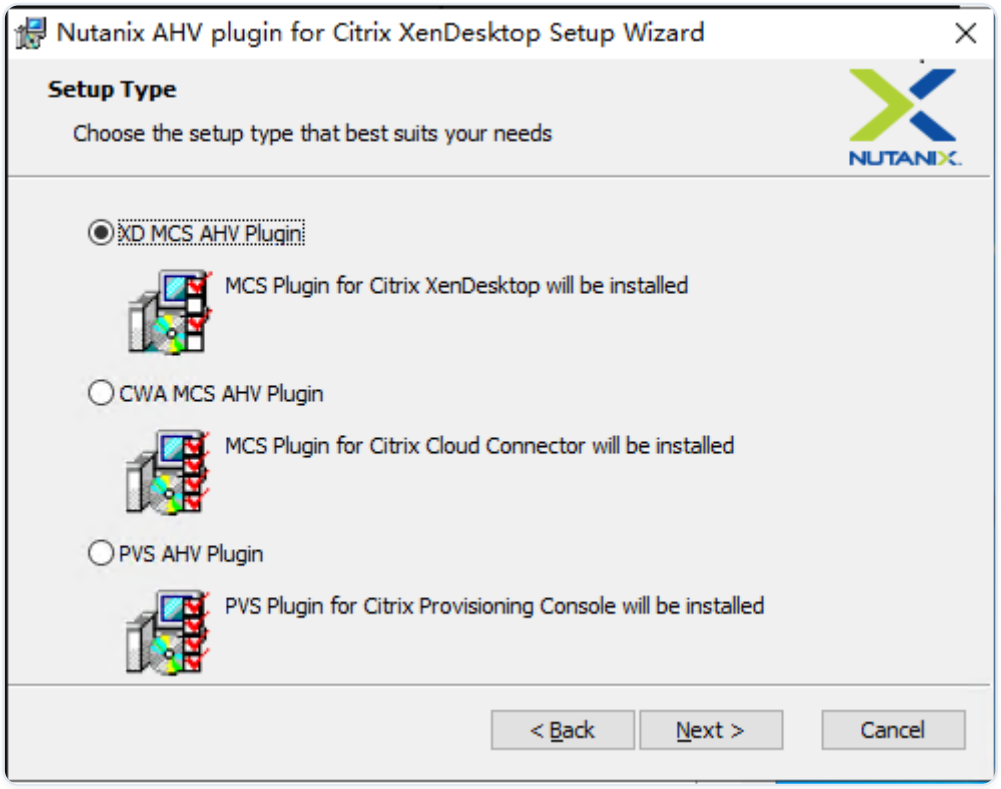
- 2203リリース以降、Citrix Virtual Apps and Desktops™およびCitrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）向けに、Microsoft Azure、AWS、GCPでLinux VDAをホストできます。これらのパブリッククラウドホスト接続をCitrix Virtual Apps and Desktops展開に追加するには、Citrix Universal Hybrid Multi-Cloud（HMC）ライセンスが必要です。
- ベアメタルサーバーは、MCSを使用して仮想マシンを作成する目的ではサポートされて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

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

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

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

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



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

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

ステップ 1c: プラグインのインストール後に次の手順を完了します

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`にNutanix Acropolisフォルダーが作成されていることを確認します。

- `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"`コマンドを実行します。
- オンプレミスのDelivery ControllerでCitrix Host、Citrix Broker、およびCitrix Machine Creation Servicesを再起動するか、Citrix Cloud ConnectorでCitrix RemoteHCLServerサービスを再起動します。

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

ステップ 2: ホスト接続を作成する

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

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

> 詳細については、Citrix Virtual Apps and Desktopsのドキュメントの［接続とリソースの作成および管理］(/ja-jp/citrix-virtual-apps-desktops/install-configure/connections.html)、およびCitrix DaaSのドキュメントの［接続の作成と管理］(/ja-jp/citrix-daas/install-configure/connections.html)を参照してください。

- ［Citrix StudioでAzureへのホスト接続を作成する］(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-azure-in-citrix-studio)
- ［Citrix StudioでAWSへのホスト接続を作成する］(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-aws-in-citrix-studio)

- ［Citrix StudioでXenServerへのホスト接続を作成する］(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-xenserver-in-citrix-studio)
- ［Citrix StudioでGCPへのホスト接続を作成する］(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#create-a-host-connection-to-gcp-in-citrix-studio)
- Citrix StudioでNutanixへのホスト接続を作成する
- Citrix StudioでVMwareへのホスト接続を作成する

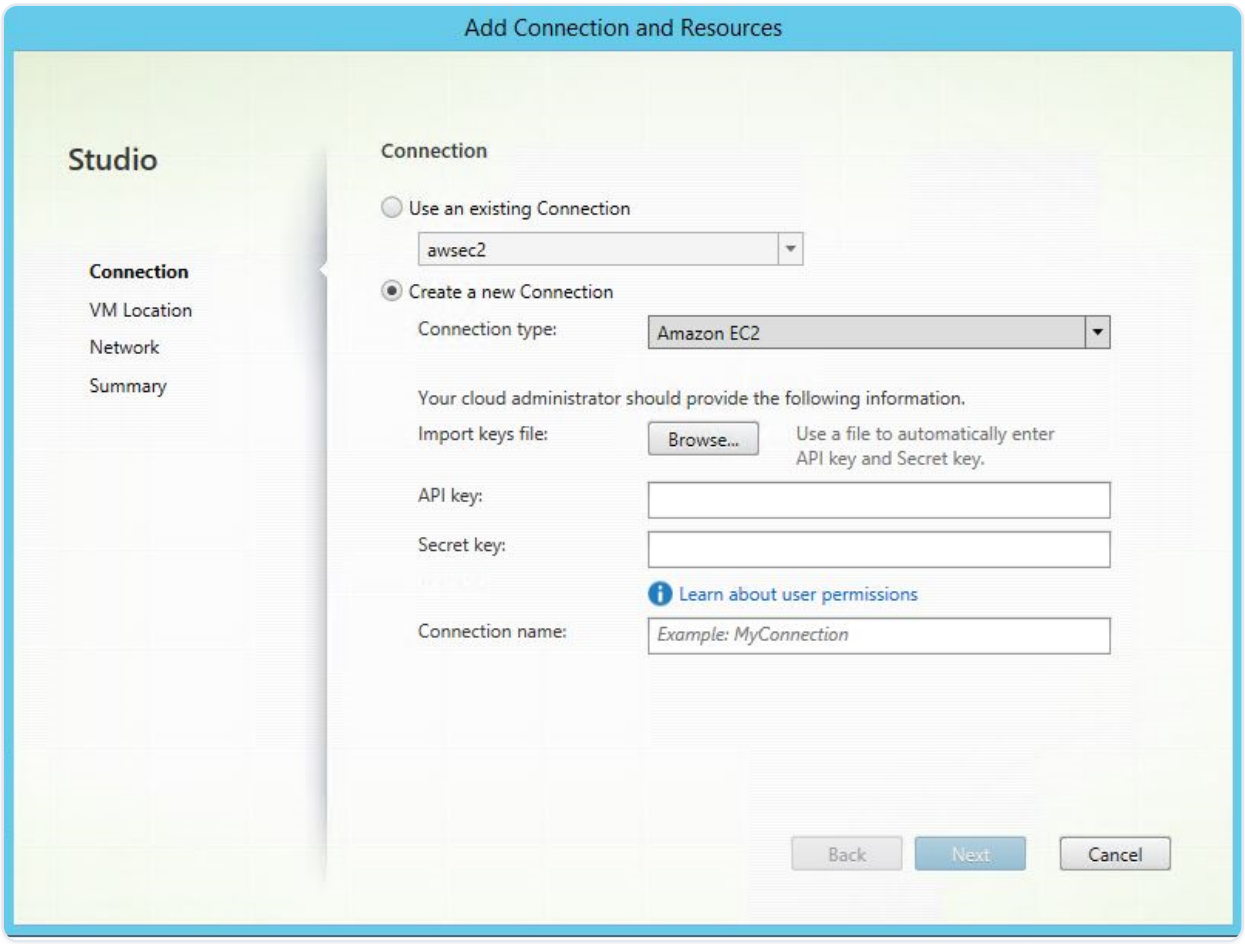
Citrix StudioでAzureへのホスト接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続およびリソースの追加**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続およびリソースの追加**を選択して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ページを進めます。各ページの内容は、選択した接続の種類によって異なります。各ページを完了したら、**次へ**を選択して**概要**ページに進みます。詳しくは、MCSを使用してドメインに参加していないLinux VDAを作成するの記事の「**手順2：ホスト接続を作成する**」を参照してください。

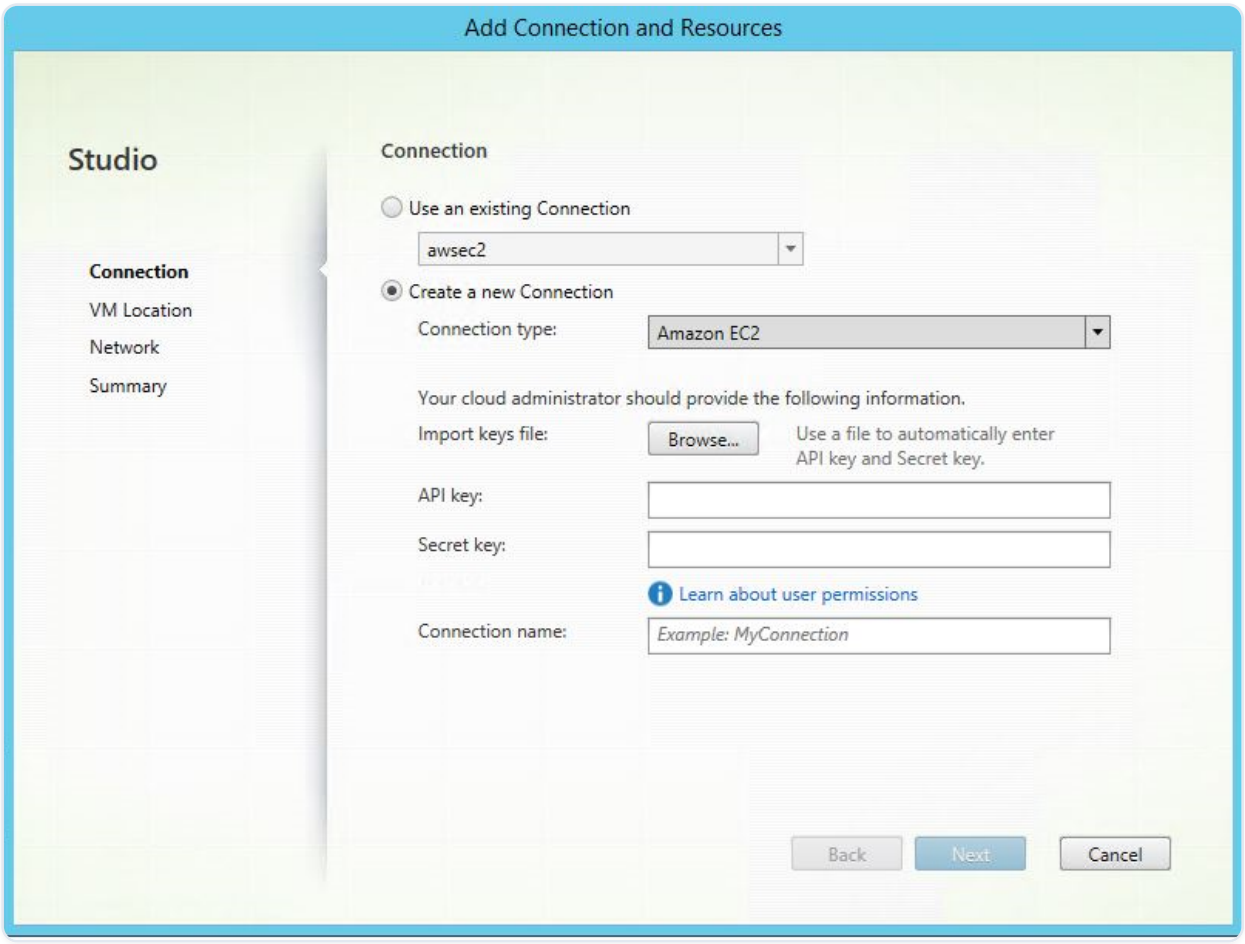
Citrix StudioでAWSへのホスト接続を作成する

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続およびリソースの追加**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続およびリソースの追加**を選択して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、次の手順を実行します。



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



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

a. AWS サービスにサインインします。

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

c. ターゲットユーザーを選択し、下にスクロールして タブを選択します。

d. 下にスクロールして をクリックします。新しいウィンドウが表示されます。

e. をクリックし、アクセスキーを安全な場所に保存します。

4. ウィザードがページを案内します。特定のページコンテンツは、選択した接続タイプによって異なります。各ページを完了したら、 を選択して ページに到達します。

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

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

2. ウィザードで、 フィールドで XenServer (旧 Citrix Hypervisor) を選択します。
3. 接続アドレス (XenServer の URL) と資格情報を入力します。
4. 接続名を入力します。

Citrix Studio で GCP へのホスト接続を作成する

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

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

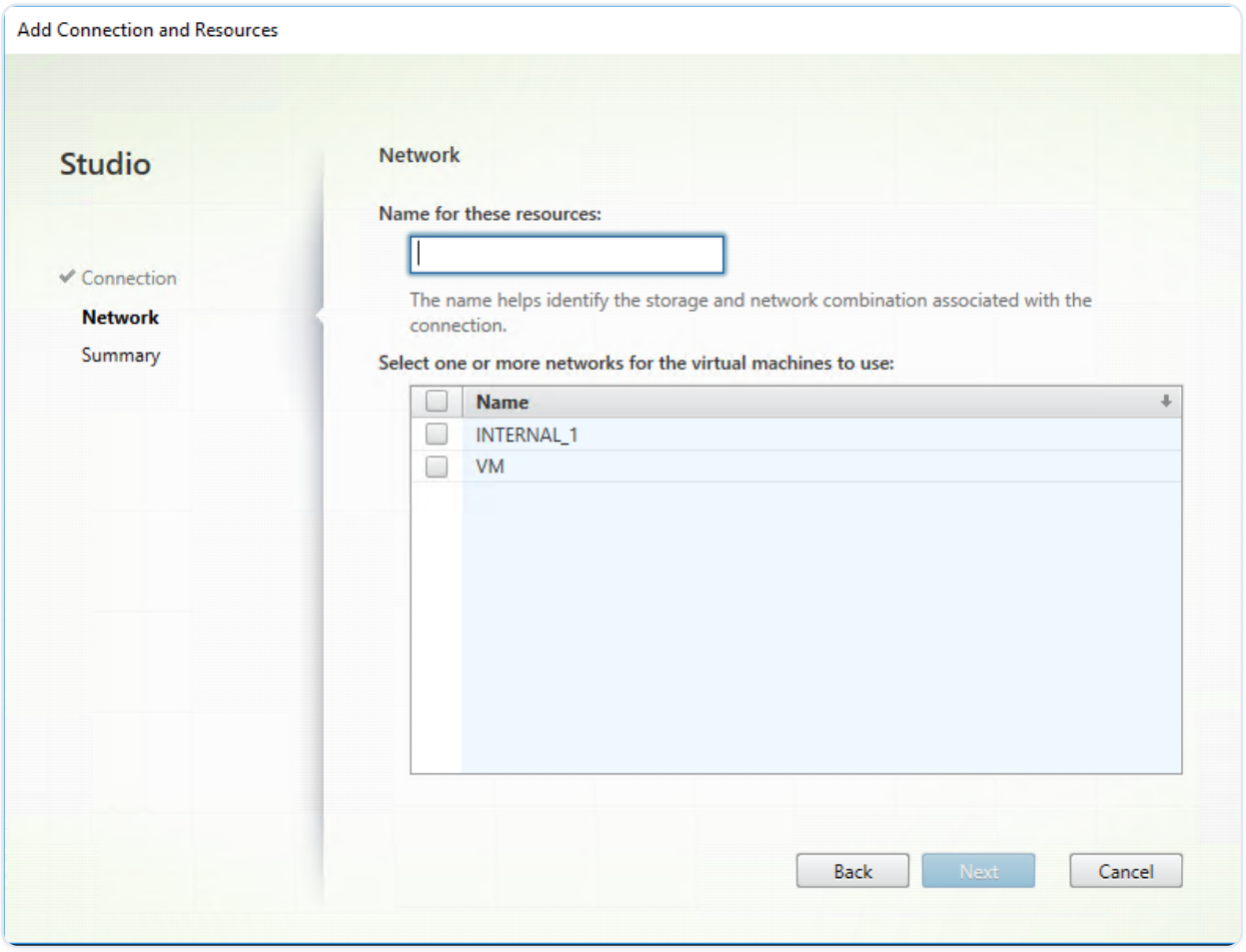
   例: Citrix Cloud の Web ベースの Studio コンソールで:

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2411/media/add-conn-connection.png)
3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードの指示に従ってページを進めます。特定のページの内容は、選択した接続の種類によって異なります。各ページの完了後、**次へ**を選択して**概要**ページに到達するまで進みます。詳細については、MCSを使用してドメインに参加していないLinux VDAを作成するの記事の「**ステップ2：ホスト接続を作成する**」を参照してください。

Citrix StudioでNutanixへのホスト接続を作成する

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

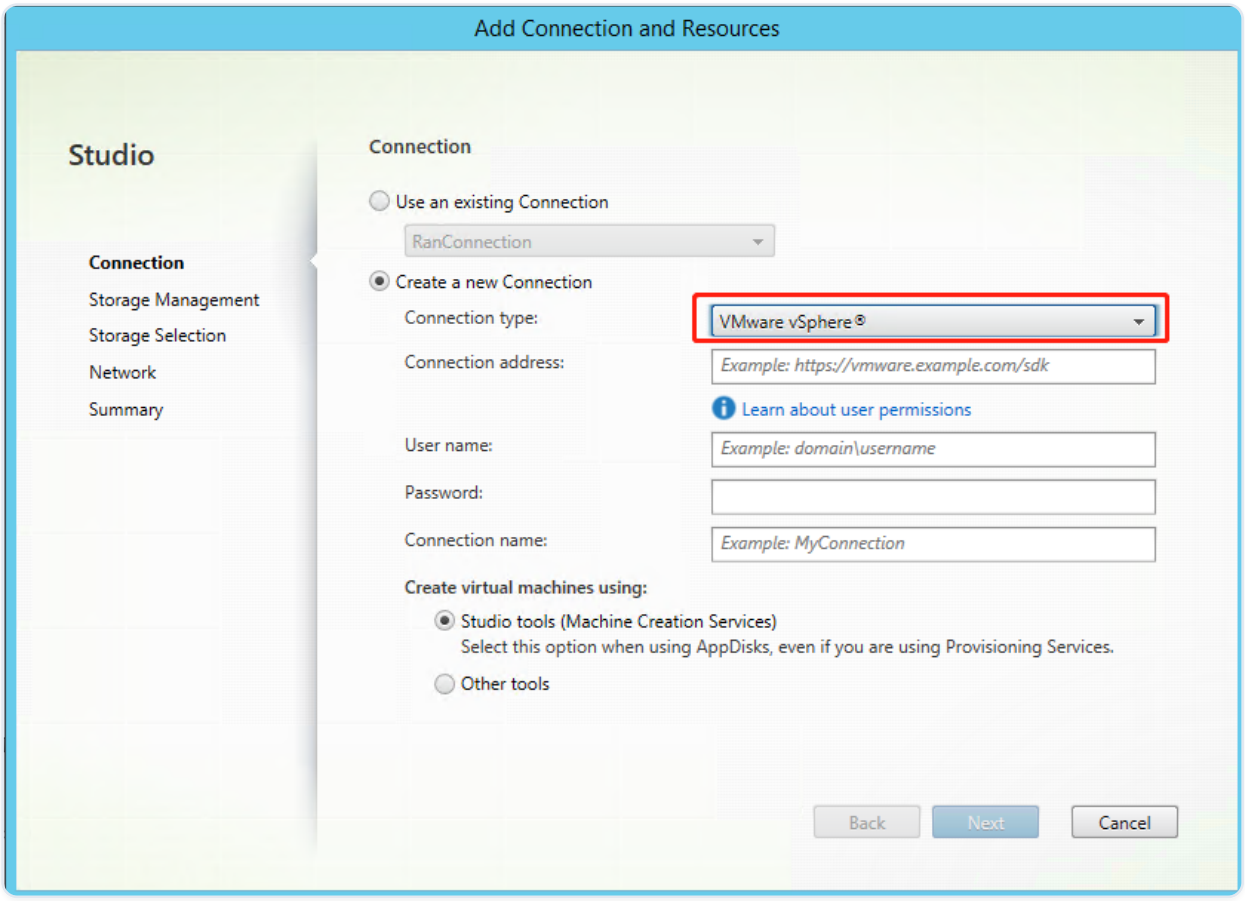
   たとえば、オンプレミスのCitrix Studioの場合：



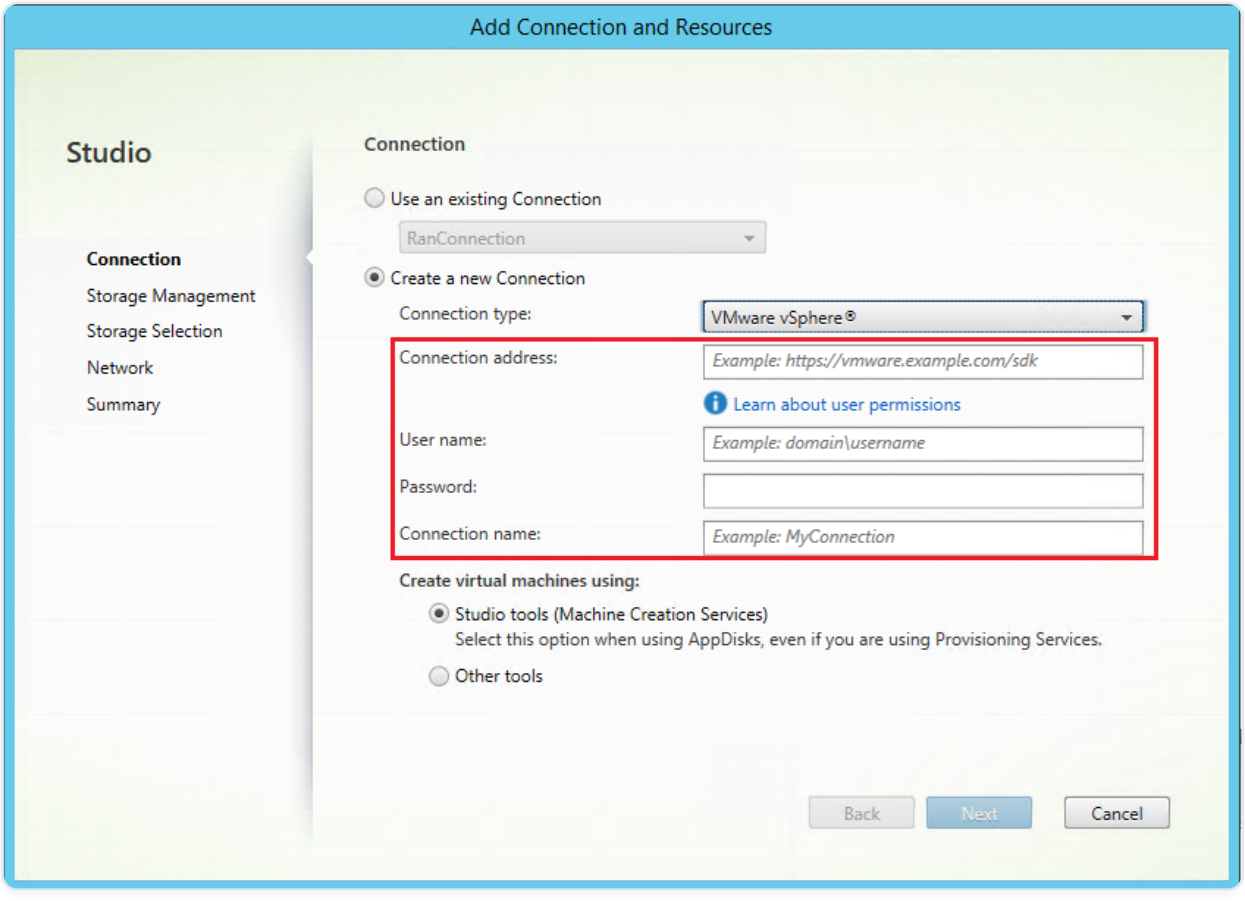
Citrix StudioでVMwareへのホスト接続を作成する

1. vSphere環境にvCenter Serverをインストールします。詳細については、VMware vSphereを参照してください。
2. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで**構成 > ホスティング > 接続およびリソースの追加**を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで**管理 > ホスティング > 接続およびリソースの追加**を選択して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の場合：



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



ステップ3: マスターイメージを準備する

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

xe CLIまたはXenCenterを使用する各VMのテンプレートVMにXenServer VM Toolsをインストールします。ツールをインストールしないと、VMのパフォーマンスが低下する可能性があります。ツールがないと、次の操作を実行できません。

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

1. 使用中のハイパーバイザーのバージョンに基づいて、XenServerダウンロードページまたはCitrix Hypervisorダウンロードページからを XenServer VM Tools for Linuxファイルをダウンロードします。
2. `LinuxGuestTools-xxx.tar.gz`ファイルをLinux VM、またはLinux VMがアクセスできる共有ドライブにコピーします。
3. tarファイルの内容を抽出します: `tar -xzf LinuxGuestTools-xxx.tar.gz`
4. お使いのLinuxディストリビューションに基づいて、次のコマンドを実行して`xe-guest-utilities`パッケージをインストールします。

   RHEL/CentOS/Rocky Linux/SUSEの場合:

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

   Ubuntu/Debianの場合:

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

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

ステップ 3b: AWS、Azure、および GCP 上の SUSE 15.5 の構成を確認する

AWS、Azure、および GCP 上の SUSE 15.5 の場合、以下を確認してください。

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

ステップ 3c: GCP 上の Ubuntu 20.04 の RDNS を無効にする

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

ステップ 3d: テンプレート VM に .NET をインストールする

Linux VDA パッケージをインストールする前に、テンプレート VM に .NET をインストールし、次の点に注意してください。

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

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

.NET のインストール後、Linux ディストリビューションに基づいて次のコマンドを実行して、Linux VDA をインストールします。

RHEL/CentOS/Rocky Linux の場合:

> **注:**

> RHEL 9.4/9.3/9.2 および Rocky Linux 9.4/9.3/9.2 に Linux VDA をインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

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

Ubuntu/Debian の場合:

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



apt-get install -f
```

SUSEの場合:

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

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

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

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

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

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3g パッケージへのパスを入力します。

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

5. ntfs-3g をインストールします。

```
./configure

make

make install
```

ステップ 3h: VM を Windows ドメインに参加させる

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

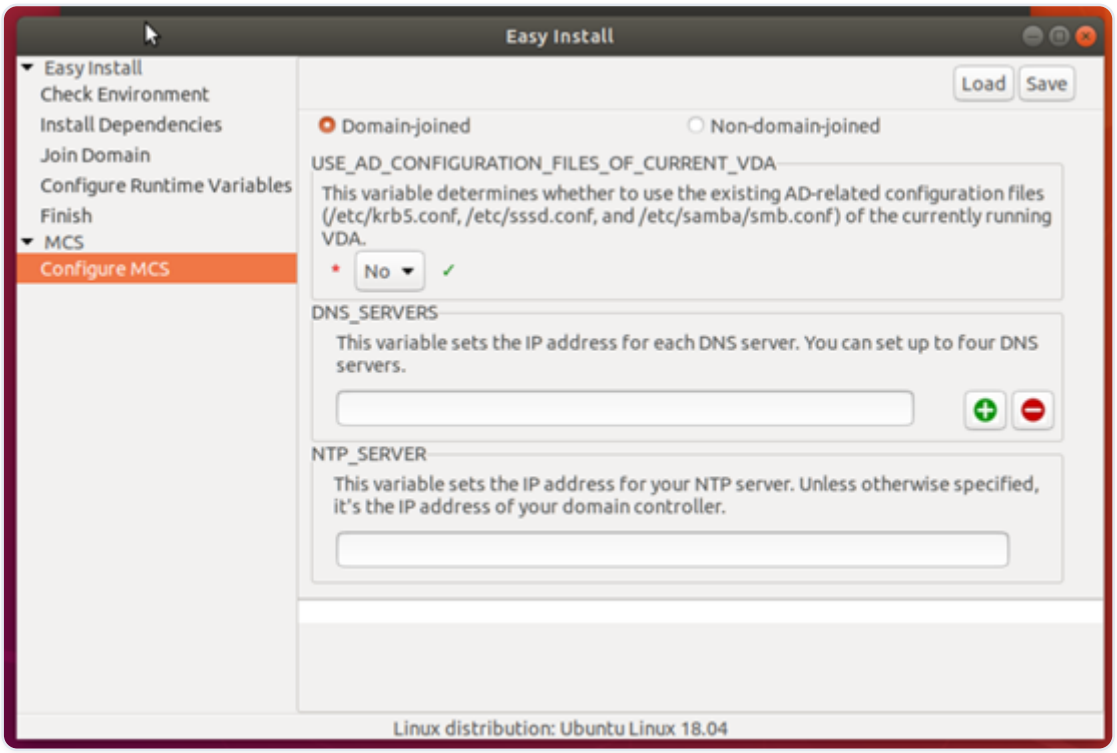
ステップ 3i: VM で FAS を構成する

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

ステップ 3j: MCS 変数を構成する

MCS変数を構成する方法は2つあります。

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



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

FASが有効なシナリオで構成できるMCS変数は次のとおりです。

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



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



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



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



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



-  `PBIS_DOWNLOAD_PATH`: Sets the path for downloading the PBIS package.
```

The value takes effect only when you set the `AD_INTEGRATION` variable to PBIS.

- `UPDATE_MACHINE_PW`: Enables or disables automating machine account password updates. For more information, see [Automate machine account password updates](https://docs.citrix.com/us/en/linux-virtual-delivery-agent/2411/installation-overview/create-linux-vdas-using-mcs.html#マシンアカウントのパスワード更新を自動化する).

- Linux VDA configuration variables:

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

  `DESKTOP_ENVIRONMENT`=gnome | mate

  `SUPPORT_DDC_AS_CNAME`=Y | N

  `VDA_PORT`=port-number

  `REGISTER_SERVICE`=Y | N

  `ADD_FIREWALL_RULES`=Y | N

  `HDX_3D_PRO`=Y | N

  `VDI_MODE`=Y | N

  `SITE_NAME`=dns-site-name | '<none\>'

  `LDAP_LIST`='list-ldap-servers' | '<none\>'

  `SEARCH_BASE`=search-base-set | '<none\>'

  `START_SERVICE`=Y | N

  `TELEMETRY_SOCKET_PORT`=port-number

```
`TELEMETRY_PORT`=port-number
```

ステップ3k：MCSのレジストリ値を書き込むか更新する

FASサーバーアドレスを設定するには、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。

```
create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
```

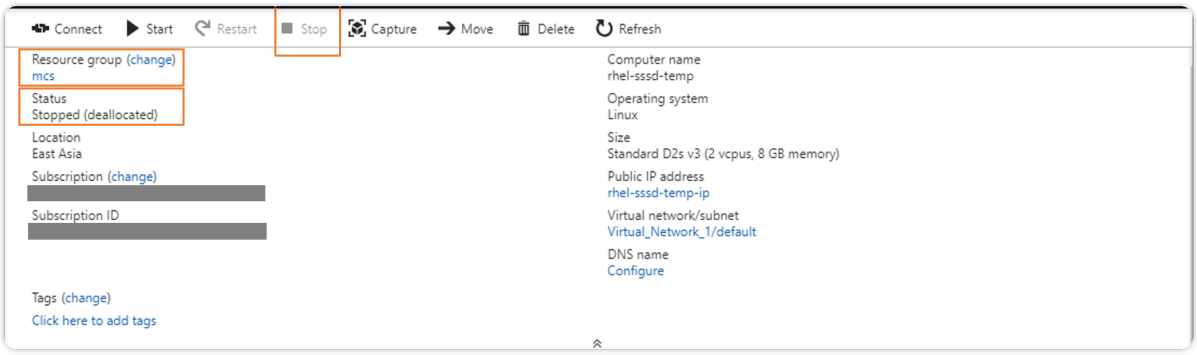
注

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

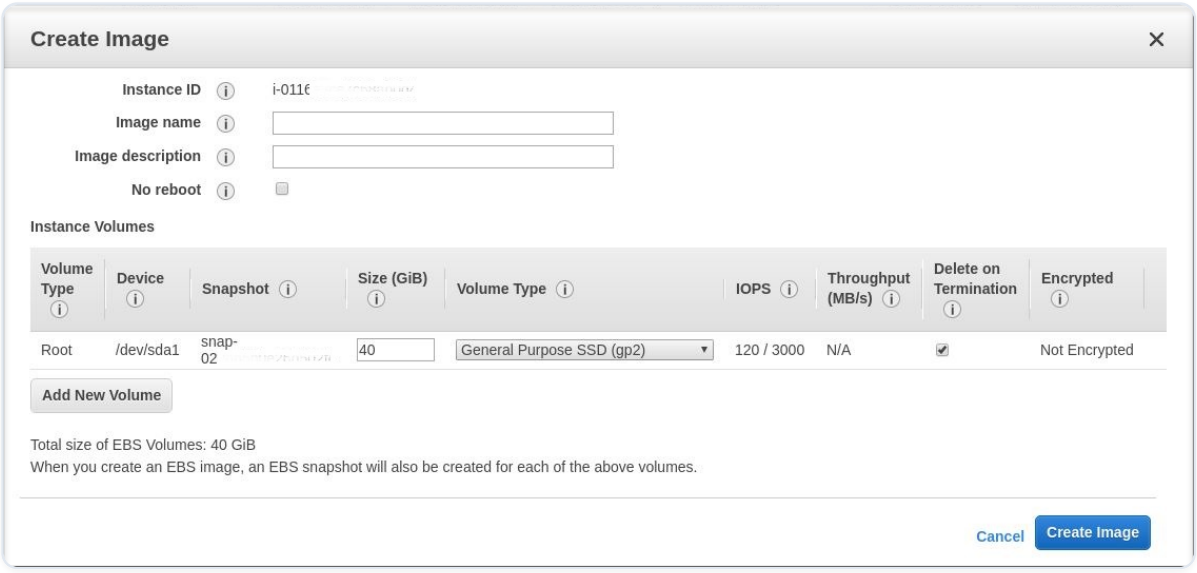
ステップ3l：マスターイメージを作成する

1. （SSSD + RHEL 8.x/9.xまたはRocky Linux 8.x/9.xの場合のみ）`update-crypto-policies --set DEFAULT:AD-SUPPORT`コマンドを実行し、テンプレートVMを再起動します。
2. `/etc/xdl/mcs/mcs.conf`を編集してMCS変数を構成する場合は、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、**展開**をクリックします。GUIで**展開**をクリックすると、GUIで設定した変数が`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。
3. 使用するパブリッククラウドに基づいて、マスターイメージのスナップショットを作成し、名前を付けます。

   - **（XenServer、GCP、およびVMware vSphereの場合）** テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成し、名前を付けます。
   - **（Azureの場合）** テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを VMをシャットダウンします。テンプレートVMの電源ステータスが**停止済み（割り当て解除済み）**であることを確認します。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるには、この名前が必要です。



- **(AWSの場合)** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**であることを確認します。テンプレートVMを右クリックし、**イメージ > イメージの作成**を選択します。必要に応じて情報を入力し、設定を行います。**イメージの作成**をクリックします。



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

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

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

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAに到達できなくなる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認します。その後、VMを再起動してからコンソールで次のコマンドを実行します。

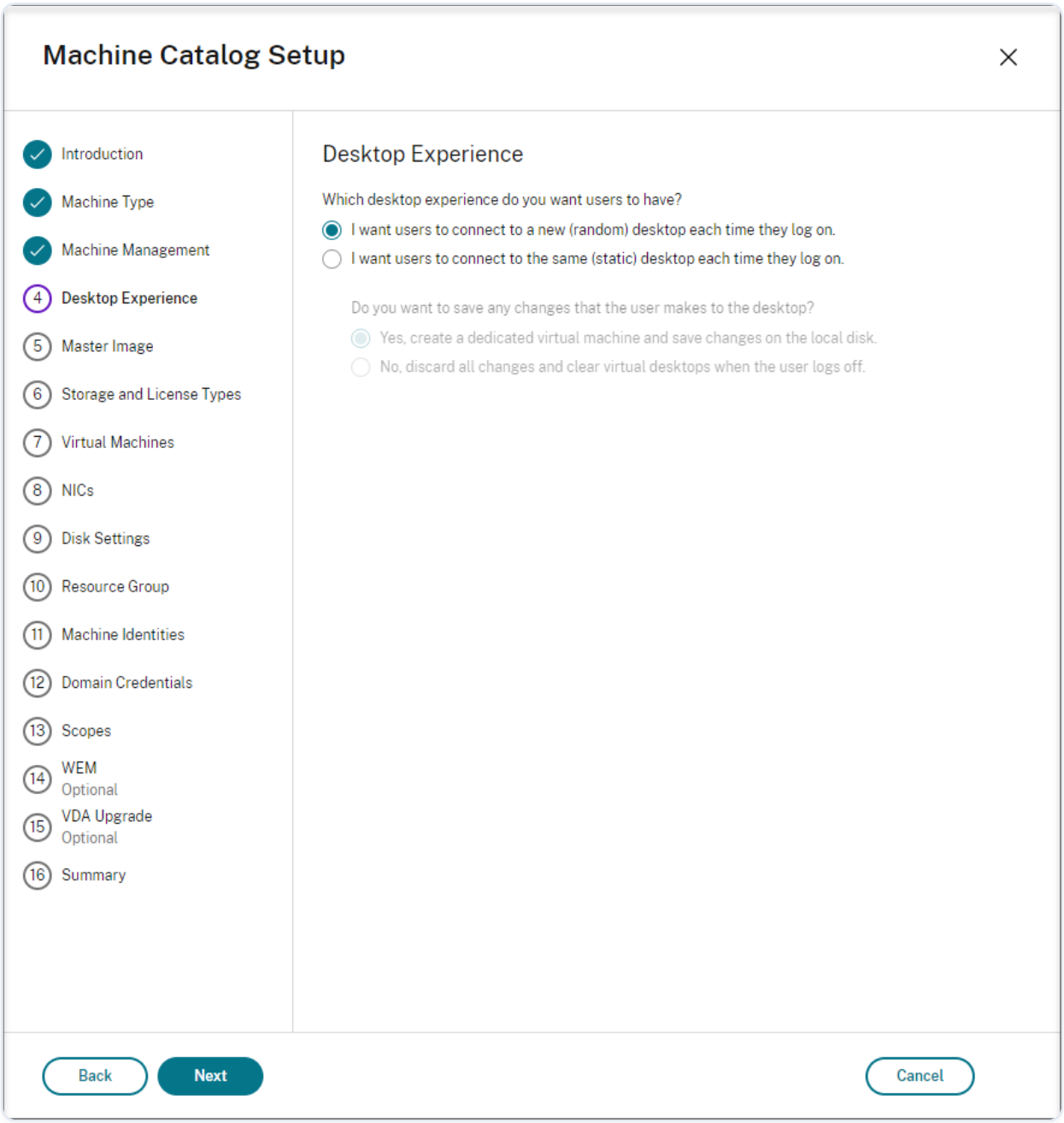
```
nmcli dev connect eth0

systemctl restart NetworkManager
```

ステップ4: マシンカタログを作成する

Citrix StudioまたはWeb Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成するときは、マスターイメージを選択し、次の点を考慮してください。

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



デスクトップエクスペリエンスページで、次のいずれかを選択します。

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

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

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

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

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

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

詳細については、Citrix Virtual Apps and Desktops ドキュメントおよび Citrix DaaS ドキュメントの「マシンカタログの作成」を参照してください。

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

ステップ5：デリバリーグループを作成する

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

詳細については、Citrix Virtual Apps and Desktops ドキュメントおよび Citrix DaaS ドキュメントの「デリバリーグループの作成」を参照してください。

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

1. ADSI Editコンソールに移動し、未登録のVMを選択して、そのマシンアカウントの**msDS-SupportedEncryptionTypes**属性を編集します。

2. VMで**ctxjproxy**サービスと**ctxvda**サービスを再起動します。VMのステータスが**Registered**に変わった場合は、手順3から5に進みます。
3. テンプレートVMで**/var/xdl/mcs/ad_join.sh**ファイルを開きます。
4. **/var/xdl/mcs/ad_join.sh** ファイル内の次の行の後に、**net ads enctypes set $NEW_HOSTNAME\$ <Decimal value of encryption type attribute, for example, 28\> -U $NEW_HOSTNAME\$ -P password** の行を追記します。

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

                join_domain_samba

                restart_service winbind /usr/bin/systemctl
```

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

マシンアカウントのパスワード更新を自動化する

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

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

   `UPDATE_MACHINE_PW="Y"`

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

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

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

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

MCSで作成されたVMでFASを有効にする

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

MCSで作成されたVMでFASを有効にするには、次の手順を実行します。

1. /etc/xdl/mcs/mcs.conf で変数を設定します。

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

   a. `Use_AD_Configuration_Files_Of_Current_VDA` の値を Y に設定します。

   b. `VDI_MODE` など、必要に応じて他の変数を設定します。

2. FASサーバーアドレスを設定するには、**/etc/xdl/mcs/mcs_local_setting.reg** ファイルに次のコマンドラインを追加します。

   ```
   create -k
   "HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
   Service" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
   ```

3. ルートCA証明書をインポートします。

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

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

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

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

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

以下のLinuxディストリビューションがサポートされています。

- ウブントゥ 22.04/20.04
- RHEL 9.4/9.2/8.10/8.8
- ロッキーリナックス 9.4/9.2/8.10/8.8

重要:

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

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

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

Citrix DaaS Standard for Azure (旧称 Citrix Virtual Apps and Desktops Standard for Azure) では、ドメイン参加済みと非ドメイン参加済みの両方の Linux VDA を作成して、Microsoft Azure から任意のデバイスに仮想アプリと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

この機能をサポートする Linux ディストリビューションは次のとおりです。

- RHEL 9.4
- RHEL 9.2
- RHEL 8.10
- RHEL 8.8

- ロッキー リナックス 9.4
- ロッキー リナックス 9.2
- ロッキー リナックス 8.10
- ロッキー リナックス 8.8
- SUSE 15.6
- ウブントゥ 24.04
- ウブントゥ 22.04
- ウブントゥ 20.04

ステップ1：Azureでマスターイメージを準備する

注:

自動ソフトウェアアップデートをスケジュールするために、Linux VDAセルフアップデート機能を使用することもできます。この目的を達成するには、マスターイメージ上のetc/xdl/mcs/mcs_local_setting.regファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。

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



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



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



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

1. Azureで、サポートされているディストリビューションのLinux VMを作成します。
2. 必要に応じて、Linux VMにデスクトップ環境をインストールします。
3. VMに、Linuxディストリビューションに基づいて.NETをインストールします。

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

   必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。詳しくは、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。
4. （Ubuntuのみ）`/etc/network/interfaces`ファイルに`source /etc/network/interfaces.d/*`行を追加します。
5. （Ubuntuのみ）`/etc/resolv.conf`を`/run/systemd/resolve/stub-resolv.conf`ではなく`/run/systemd/resolve/resolv.conf`にポイントします。

   ```
   unlink /etc/resolv.conf



   ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
   ```
6. Linux VDAパッケージをインストールします。
7. 使用するデータベースを指定します。

   PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後に、SQLiteとPostgreSQLを切り替えることもできます。そのためには、次の手順を実行します。

   a. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールである場合は、この手順を省略します。

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

      注:

- VDIモードでのみSQLiteを使用することをお勧めします。
- 簡単インストールとMCSの場合、SQLiteとPostgreSQLを手動でインストールすることなく切り替えることができます。`/etc/xdl/db.conf`で特に指定がない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡単インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。**db.conf**ファイルの例については、ステップ7：使用するデータベースを指定するを参照してください。
- `/etc/xdl/db.conf`を使用してPostgreSQLのポート番号を構成することもできます。

8. MCS変数を変更します。

   MCS変数を構成する方法は2つあります。

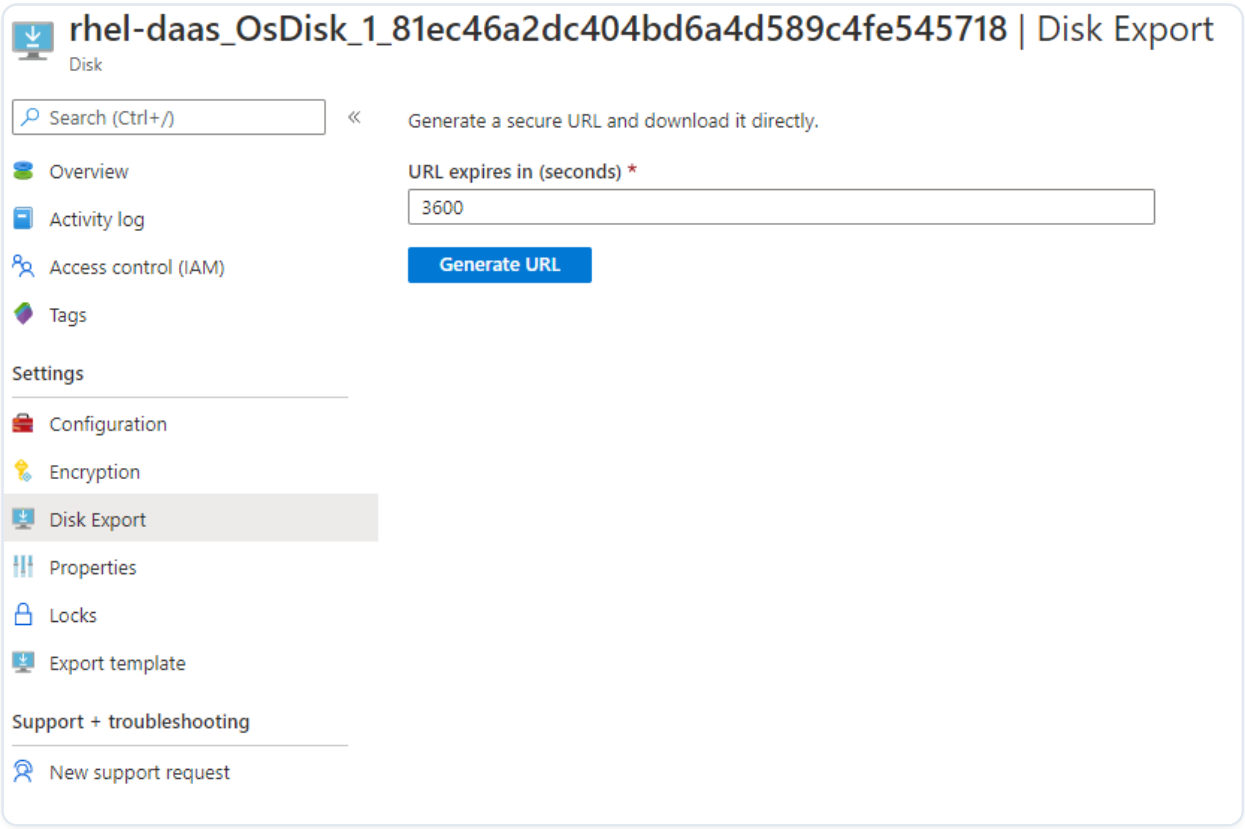
   - `/etc/xdl/mcs/mcs.conf`ファイルを編集します。
   - 簡単インストールGUIを使用します。簡単インストールGUIを開くには、Linux VDAのデスクトップ環境で`/opt/Citrix/VDA/bin/easyinstall`コマンドを実行します。

   注:

   `dns`変数を指定しないままにします。 マシンカタログを作成する際に**Static**または**Random**タイプを選択した場合は、`VDI_MODE=Y`を設定します。

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

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



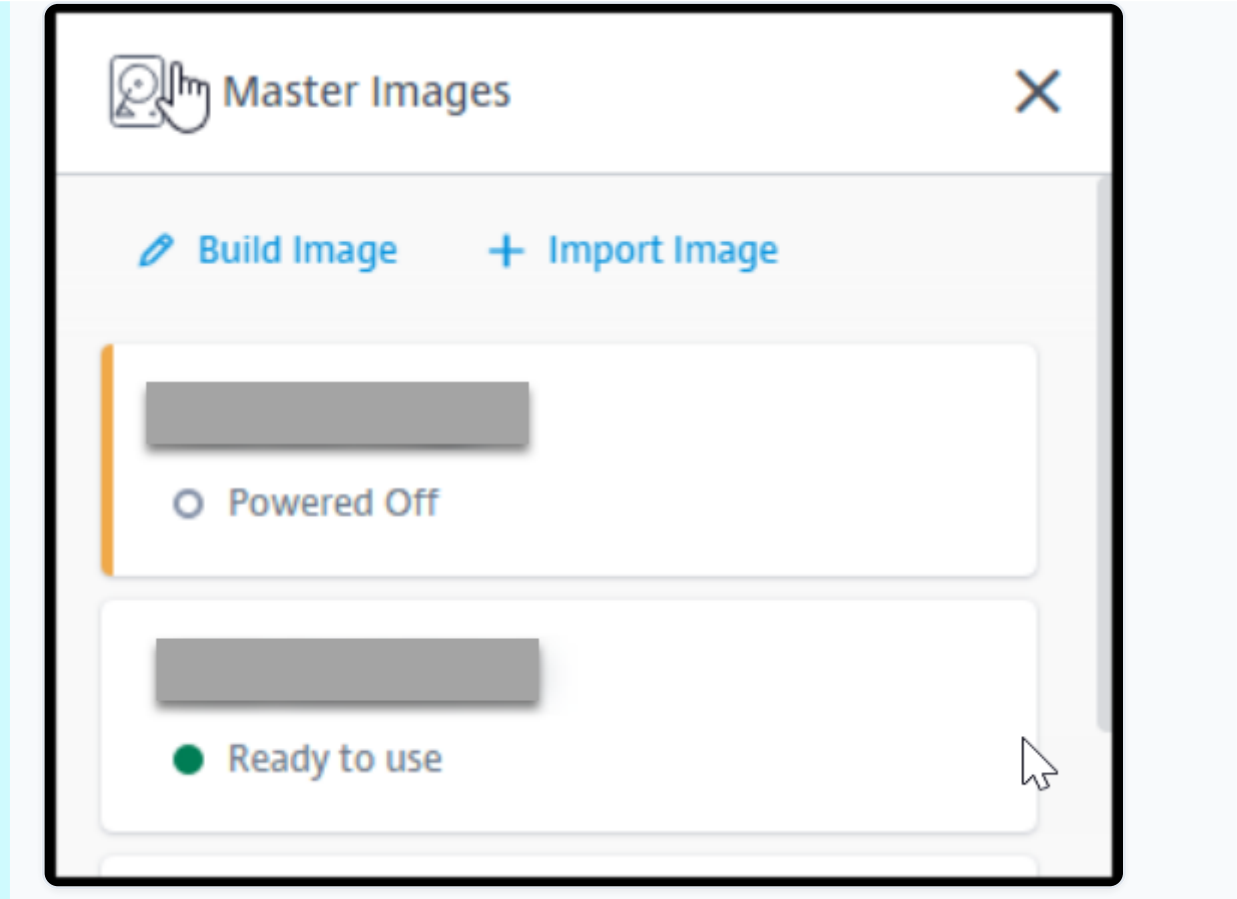
10. （オプション）マスターイメージでグループポリシー設定を行います。`ctxreg`ツールを使用してグループポリシー設定を行うことができます。たとえば、次のコマンドはPDF印刷用の**Auto-create PDF Universal Printer**ポリシーを有効にします。

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

ステップ2：Azureからマスターイメージをインポートする

1. Manageダッシュボードから、右側の**マスターイメージ**を展開します。表示には、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:**
>
> このサービスのほとんどの管理者アクティビティは、**Manage**および**Monitor**ダッシュボードを通じて管理されます。最初のカタログを作成すると、Citrix Cloud™にサインインして**Managed Desktops**サービスを選択した後、**Manage**ダッシュボードが自動的に起動します。



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

   Azure からイメージをインポートする(/en-us/linux-virtual-delivery-agent/2411/media/import-an-image-from-azure.png)
4. ウィザードの指示に従って、マスターイメージのインポートを完了します。

ステップ3：マシンカタログを作成する

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

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

Linux VDA を手動でインストールする

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

- アマゾン Linux 2、CentOS、RHEL、およびRocky Linux
- SUSE
- ウブントゥ
- デビアン

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

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

ステップ1：構成情報とLinuxマシンの準備

ステップ1a：ネットワーク構成の確認

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

ステップ1b：ホスト名の設定

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

`hostname`

ステップ1c：ホスト名にループバックアドレスを割り当てる

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

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

例:

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

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

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

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

ステップ1d：ホスト名の確認

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e：名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

ステップ1f：クロック同期の構成

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

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

CHRONYサービスの構成

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

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


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

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

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

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chronyd
```

ステップ1g：使用するデータベースのインストールと指定

注:

- VDIモードではSQLiteを、ホスト型共有デスクトップ配信モデルではPostgreSQLを使用することをお勧めします。
- 簡単インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で別途指定されていない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡単インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、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。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

POSTGRESQLのインストール

このセクションでは、Linuxディストリビューションが提供するPostgreSQLのバージョンをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

PostgreSQLをインストールするには、次のコマンドを実行します。

```
sudo yum -y install postgresql-server
```

```
sudo yum -y install postgresql-jdbc
```

RHEL 8.xおよびRHEL 9.4/9.2の場合、PostgreSQL用に`libpq`をインストールするには、次のコマンドを実行します。

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

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

```
sudo postgresql-setup initdb
```

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

PostgreSQLのバージョンを確認するには、次を使用します。

```
psql --version
```

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

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

SQLITEのインストール

SQLiteをインストールするには、次のコマンドを実行します。

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

使用するデータベースの指定

SQLiteとPostgreSQLの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh**を実行します。クリーンインストールである場合は、この手順を省略します。
2. 使用するデータベースを指定するために、**/etc/xdl/db.conf** を編集します。以下は **db.conf** ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432
```

```
## PostgreSQL customized
# only the following value means true, otherwise false:
#     true
#     yes
#     y
#     YES
#     Y
# default is false
DbCustomizePostgreSQL=false

## PostgreSQL service name
# specify the service name of PostgreSQL for Linux VDA
# default is "postgresql"
DbPostgreSQLServiceName="postgresql"
```

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

3. **ctxsetup.sh** を実行します。

注:

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

ステップ2：ハイパーバイザーを準備する

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

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

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

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

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

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

管理オペレーティングシステムから:

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**が選択されていることを確認します。

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

ESXおよびESXiでの時刻同期の修正

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

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

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。
4. 「**VMware Tools**」を選択します。

5. 「**Advanced**」ボックスで、「**Synchronize guest time with host**」のチェックを外します。

ステップ3：Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory (AD) ドメインに追加するには、以下の方法が利用可能です。

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

選択した方法に基づいて、指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名が使用されている場合、セッションの起動に失敗する可能性があります。

サンバ Winbind

RHEL 9.4/9.2およびRocky Linux 9.4/9.2の場合、**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、次のコマンドを実行します。

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

必要なパッケージをインストールまたは更新します。

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

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

Amazon Linux 2の場合：

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

マシン起動時にWINBINDデーモンが起動するように有効にする

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

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

WINBIND認証を構成する

Winbindを使用してKerberos認証用にマシンを構成します。

1. 次のコマンドを実行します。

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

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

   Amazon Linux 2の場合：

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

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

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

   `--enablekrb5kdcdns --enablekrb5realmdns`

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

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

   `kerberos method = secrets and keytab`

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

3. （RHEL 9.4/9.2/8.xおよびRocky Linux 9.4/9.2/8.xのみ）**/etc/krb5.conf**を開き、`[libdefaults]`、`[realms]`、および`[domain_realm]`セクションの下にエントリを追加します。

   `[libdefaults]`セクションの下：

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

   `default_realm = REALM` `dns_lookup_kdc = true`

   `[realms]`セクションの下：

   `REALM = {`

   `kdc = fqdn-of-domain-controller` `}`

   `[domain_realm]`セクションの下：

   `realm = REALM`

   `.realm = REALM`

Linux VDA は、システムキータブファイル /etc/krb5.keytab を使用して、Delivery Controller で認証および登録する必要があります。以前の Kerberos メソッド設定により、マシンが最初にドメインに参加したときに Winbind がシステムキータブファイルを作成するよう強制されます。

WINDOWS ドメインへの参加

ドメインコントローラーに到達可能である必要があり、ドメインにコンピューターを追加する権限を持つ Active Directory ユーザーアカウントが必要です。

Linux VM を Windows ドメインに追加するには、次のコマンドを実行します。

```
sudo realm join -U user --client-software=winbind REALM
```

ヒント:

Amazon Linux 2 で実行されている Linux VM の場合、次のコマンドを使用して Windows ドメインに追加することもできます。

```
sudo net ads join REALM -U user
```

> **REALM** は大文字の Kerberos レルム名で、**user** はドメインにコンピューターを追加する権限を持つドメインユーザーです。

WINBIND の PAM を構成する

デフォルトでは、Winbind PAM モジュール (pam_winbind) の構成では、Kerberos チケットのキャッシュとホームディレクトリの作成は有効になっていません。**/etc/security/pam_winbind.conf** を開き、\[Global\] セクションの下に次のエントリを追加または変更します。

`krb5_auth = yes`

`krb5_ccache_type = FILE` `mkhomedir = yes`

各設定の先頭にあるセミコロンが削除されていることを確認してください。これらの変更には、Winbind デーモンの再起動が必要です。

```
sudo systemctl restart winbind
```

> **ヒント:**

> マシンがドメインに参加している場合にのみ、`winbind` デーモンは実行され続けます。

> **/etc/krb5.conf** を開き、\[libdefaults\] セクションの下にある次の設定を KEYRING から FILE タイプに変更します。

`default_ccache_name = FILE:/tmp/krb5cc_%{uid}`

RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 の場合、Winbind の SELinux の問題を解決するには、次のコマンドを実行します。

```
ausearch -c 'winbindd' --raw | audit2allow -M my-winbindd -p /etc/selinux/
targeted/policy/policy.*



semodule -X 300 -i my-winbindd.pp
```

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が **Active Directory** にコンピューターオブジェクトを持っていることを要求します。

マシンがドメインに参加していることを確認するには、**Samba** の **net ads** コマンドを実行します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

KERBEROS構成の確認

KerberosがLinux VDAで使用するために正しく構成されていることを確認するには、システムkeytabファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

マシンのアカウントの詳細を確認するには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

wbinfoツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

Kerberos資格情報キャッシュ内のチケットが有効で期限切れでないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、手順6：Linux VDAのインストール に進みます。

Quest 認証サービス

ドメインコントローラーでQUESTを構成する

Active Directoryドメインコントローラーにソフトウェアをインストールして構成済みであり、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。

2. 「**Unixアカウント**」タブを選択します。
3. 「**Unixを有効にする**」をオンにします。
4. 「**プライマリGID番号**」を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDAでQUESTを構成する

SELinuxポリシーの適用を回避する
デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨害され、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinuxを無効にすることです。rootユーザーとして、***/etc/selinux/config*** を編集し、**SELinux** の設定を変更します。

```
SELINUX=permissive
```

この変更にはマシンの再起動が必要です。

```
reboot
```

> **重要:**
>
> この設定は慎重に使用してください。無効にした後にSELinuxポリシーの適用を再度有効にすると、rootユーザーや他のローカルユーザーであっても、完全にロックアウトされる可能性があります。

VASデーモンを構成する

Kerberosチケットの自動更新は有効にして切断する必要があります。認証 (オフラインログオン) は無効にする必要があります。

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間 (32,400秒) に設定します。これは、デフォルトの10時間のチケット有効期間より1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。

PAMとNSSを構成する

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、PAMとNSSを手動で構成するために次のコマンドを実行します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

WINDOWSドメインに参加する

Questの**vastool**コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。**domain-name**はドメインのDNS名です (例: example.com)。

ドメイン参加後、Linuxマシンを再起動します。

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン (WindowsおよびLinux VDA) が`Active Directory`にコンピューターオブジェクトを持つことを要求します。Questに参加しているLinuxマシンがドメイン上にあることを確認するには:

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます。

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証を確認する

QuestがPAMを介してドメインユーザーを認証できることを確認するには、以前に使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -uコマンドによって返されるUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos 資格情報キャッシュ内のチケットが有効で期限切れでないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/manual-installation-overview/redhat.html#step-6-install-the-linux-vda)に進みます。

セントリファイ ダイレクトコントロール

WINDOWS ドメインに参加する

Centrify DirectControl Agent がインストールされている場合、Centrify `adjoin` コマンドを使用して Linux マシンを Active Directory ドメインに参加させます。

```
su –

adjoin -w -V -u user domain-name
```

user パラメーターは、コンピューターを Active Directory ドメインに参加させる権限を持つ Active Directory ドメインユーザーです。**domain-name** は、Linux マシンを参加させるドメインの名前です。

ドメインメンバーシップを確認する

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が Active Directory にコンピューターオブジェクトを持っていることを要求します。Centrify に参加している Linux マシンがドメイン上にあることを確認するには:

```
su –

adinfo
```

「Joined to domain」の値が有効であり、CentrifyDC モードが「connected」を返すことを確認します。モードが開始状態のままになっている場合、Centrify クライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all

adinfo –diag
```

さまざまな Active Directory および Kerberos サービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/manual-installation-overview/redhat.html#step-6-install-the-linux-vda)に進みます。

SSSD

SSSD を使用している場合は、このセクションの指示に従ってください。このセクションには、Linux VDA マシンを Windows ドメインに参加させるための手順と、Kerberos 認証を構成するためのガイダンスが含まれています。

RHELでSSSDをセットアップするには、以下を実行します。

1. ドメインに参加し、ホストキー配布ファイルを作成する
2. SSSDをセットアップする
3. SSSDを有効にする
4. Kerberos構成を確認する
5. ユーザー認証を確認する

ドメインに参加し、ホストキー配布ファイルを作成する

SSSDは、ドメインへの参加やシステムキー配布ファイルの管理のためのActive Directoryクライアント機能を提供しません。代わりに、**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向けの**Samba**アプローチと、RHEL 8.x/9.xおよびRocky Linux 8.x/9.x向けの`adcli`アプローチについて説明します。**realmd**については、RHELのドキュメントを参照してください。これらの手順は、SSSDを構成する前に実行する必要があります。

- **サンバ (Amazon Linux 2):**

  必要なパッケージをインストールまたは更新します。

  ```
  sudo yum -y install krb5-workstation authconfig oddjob-mkhomedir samba-common-tools
  ```

  適切に構成されたファイルを持つLinuxクライアントで：

  - /etc/krb5.conf
  - /etc/samba/smb.conf:

  SambaおよびKerberos認証用にマシンを構成します。

  ```
  sudo authconfig --smbsecurity=ads --smbworkgroup=domain --smbrealm=REALM --krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --update
  ```

  ここで、**REALM** は大文字の Kerberos レルム名、**domain** は Active Directory ドメインの短い NetBIOS 名です。

  > **注:**

> この記事の設定は、単一ドメイン、単一フォレストモデルを対象としています。
> Kerberos は AD インフラストラクチャに基づいて構成してください。

> KDC サーバーとレルム名の DNS ベースのルックアップが必要な場合は、上記のコマンドに次の 2 つのオプションを追加します。

`--enablekrb5kdcdns --enablekrb5realmdns`

/etc/samba/smb.conf を開き、**\[Global\]** セクションの下に、ただし **authconfig** ツールによって生成されたセクションの後に、次のエントリを追加します。

`kerberos method = secrets and keytab`

`winbind offline logon = no`

Windows ドメインに参加します。ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントがあることを確認してください。

```
sudo net ads join REALM -U user
```

REALM は大文字の Kerberos レルム名、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

- **Adcli (RHEL 9.4/9.2/8.x および Rocky Linux 9.4/9.2/8.x):**

  必要なパッケージをインストールまたは更新します。

  ```
  sudo yum -y install samba-common samba-common-tools krb5-workstation
  authconfig oddjob-mkhomedir realmd oddjob authselect
  ```

  Samba および Kerberos 認証用にマシンを構成します。

  ```
  sudo authselect select sssd with-mkhomedir --force
  ```

  /etc/krb5.conf を開き、 および セクションの下にエントリを追加します。

  セクションの下に:

  `REALM = {`

`kdc = fqdn-of-domain-controller` `}`

セクションの下に:

`realm = REALM` `.realm = REALM`

Windows ドメインに参加します。ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントがあることを確認してください。

```
sudo realm join REALM -U user
```

REALM は大文字の Kerberos レルム名、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD をセットアップする

SSSD の設定は、次の手順で構成されます。

- `sudo yum -y install sssd` コマンドを実行して、Linux VDA に **sssd-ad** パッケージをインストールします。
- さまざまなファイル（例: sssd.conf）に構成変更を加えます。
- **sssd** サービスを開始します。

(RHEL 9.4/9.2/8.x および Rocky Linux 9.4/9.2/8.x のみ) **/etc/sssd/sssd.conf** を開き、 セクションの下に次のエントリを追加します。

`ad_gpo_access_control = permissive`

`full_name_format = %2$s\%1$s` `fallback_homedir = /home/%d/%u`

`# Kerberos settings` `krb5_ccachedir = /tmp`

`krb5_ccname_template = FILE:%d/krb5cc_%U`

ad.example.com、**server.ad.example.com** を対応する値に置き換えます。詳細については、sssd-ad(5) - Linux man page を参照してください。

sssd.conf のファイルの所有権と権限を設定します。

`chown root:root /etc/sssd/sssd.conf`

`chmod 0600 /etc/sssd/sssd.conf` `restorecon /etc/sssd/sssd.conf`

SSSD を有効にする

RHEL 9.4/9.2/8.x および Rocky Linux 9.4/9.2/8.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。

```
sudo systemctl restart sssd

sudo systemctl enable sssd.service

sudo chkconfig sssd on
```

Amazon Linux 2 の場合:

authconfig を使用して SSSD を有効にします。ホームディレクトリの作成が SELinux と互換性があることを確認するために、**oddjob-mkhomedir** をインストールします。

```
authconfig --enablesssd --enablesssdauth --enablemkhomedir --update



sudo systemctl start sssd



sudo chkconfig sssd on
```

KERBEROS 構成を確認する

システムの **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用して、Kerberos **kinit** コマンドを実行し、ドメインコントローラーでマシンを認証します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は、大文字で指定する必要があります。ドル記号 (**$**) は、シェルによる置換を防ぐためにバックスラッシュ (****) でエスケープする必要があります。一部の環境では、

DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントの TGT チケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist
```

ユーザー認証の確認

getent コマンドを使用して、ログオン形式がサポートされており、NSS が機能することを確認します。

```
sudo getent passwd DOMAIN\\username
```

DOMAIN パラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOS サフィックス形式: `username@DOMAIN`

SSSD PAM モジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
sudo ssh localhost –l DOMAIN\\username



id -u
```

コマンドによって返された **uid** に対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_{uid}
```

ユーザーの Kerberos 資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
klist
```

ドメイン参加の確認後、手順 6: Linux VDA のインストール に進みます。

PBIS

必要なPBISパッケージをダウンロードする

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/
pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行可能にする

```
chmod +x pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行する

```
sh pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

WINDOWSドメインに参加する

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
/opt/pbis/bin/domainjoin-cli join domain-name user
```

userは、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name**は、example.comなどのドメインのDNS名です。

注： Bashをデフォルトシェルとして設定するには、**/opt/pbis/bin/config LoginShellTemplate/bin/bash**コマンドを実行します。

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。PBISに参加したLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。そうでない場合は、ホスト名のみが表示されます。

ユーザー認証を確認する

PBISがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

id -uコマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順6：Linux VDAをインストールするに進みます。

ステップ4：.NETのインストール

.NETランタイムに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションに.ASP.NET Core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。その他のディストリビューションにはバージョン8が必要です。

必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

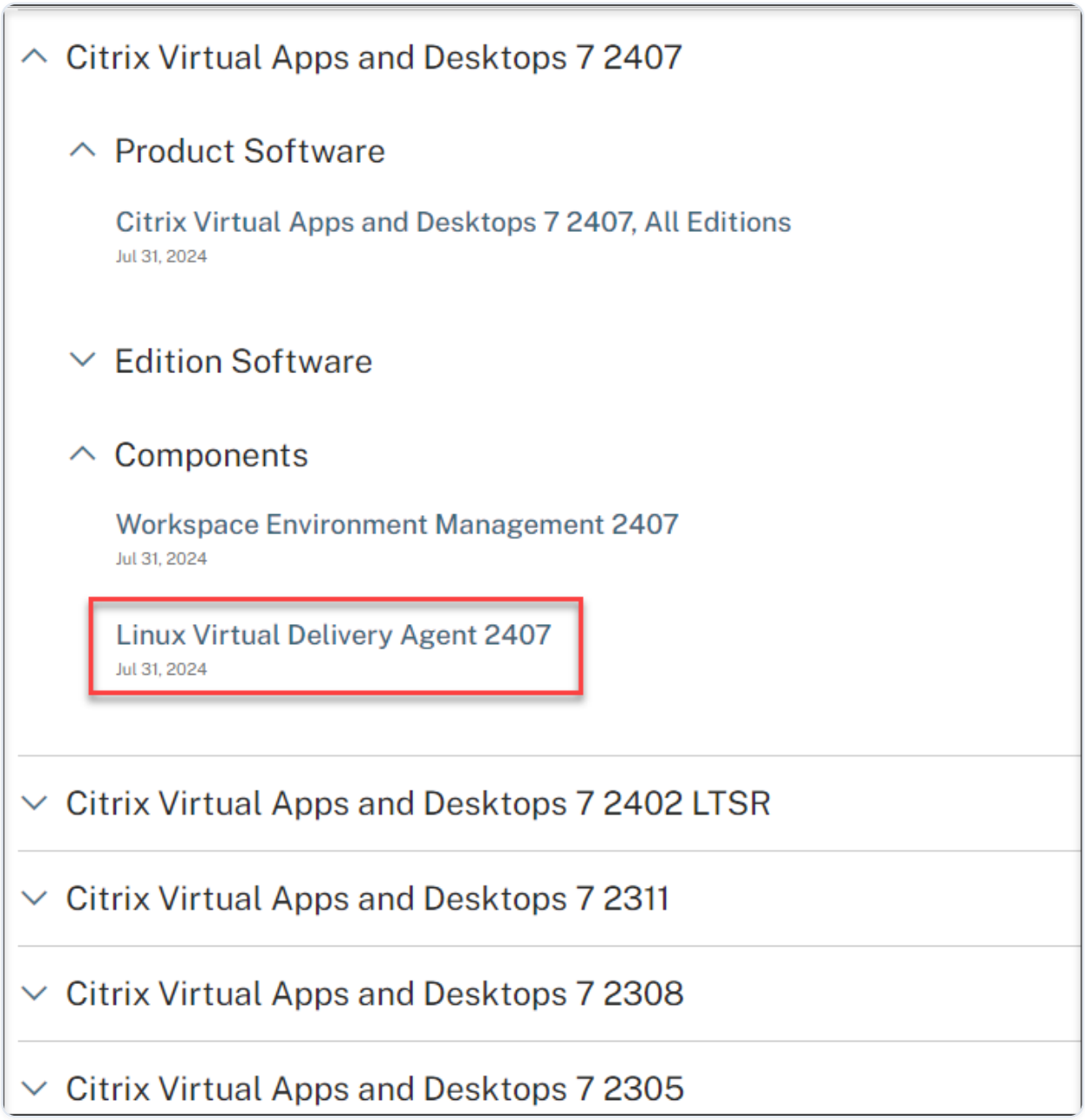
.NETのインストール後、**which dotnet**コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

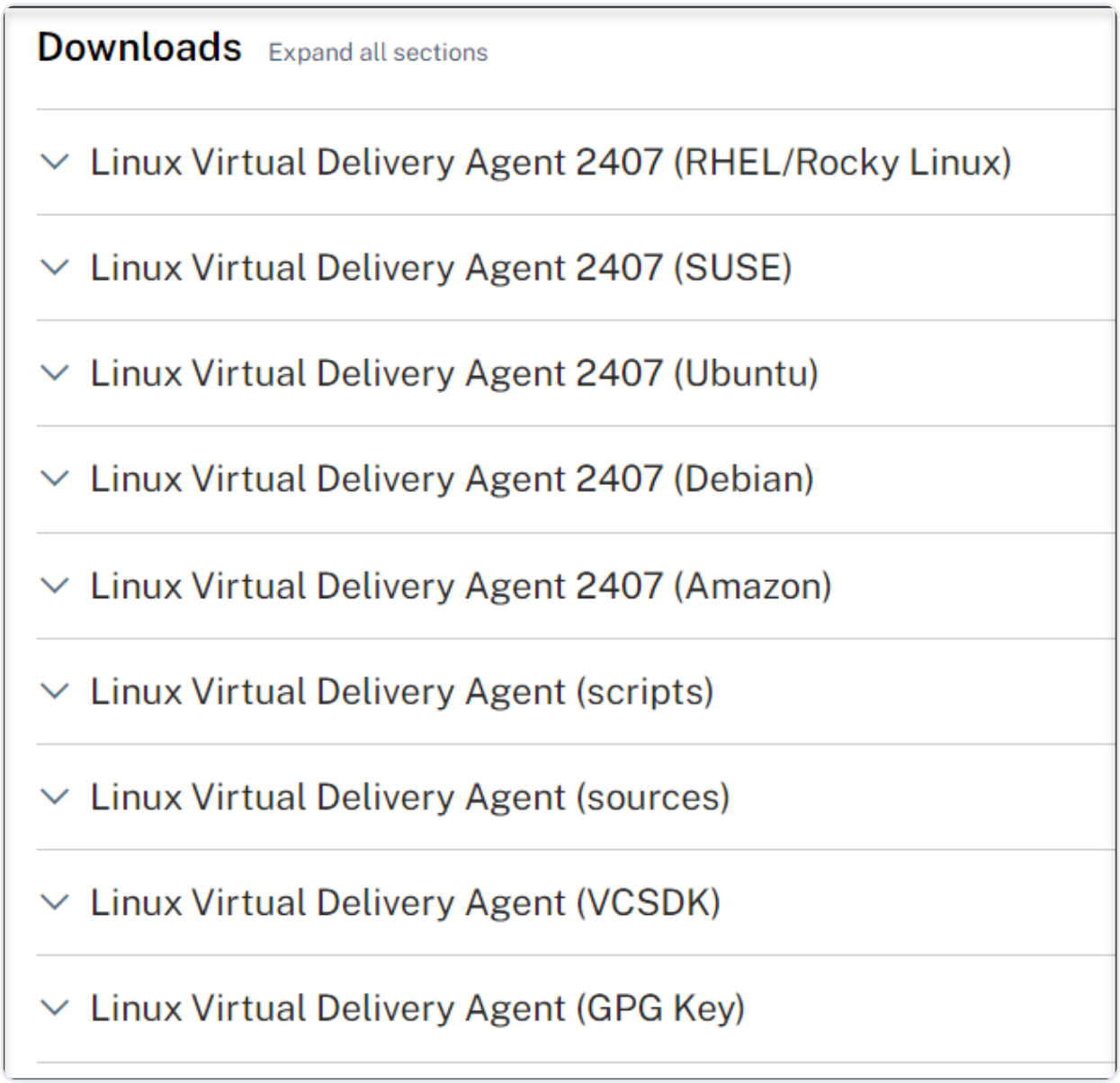
ステップ5：Linux VDAパッケージのダウンロード

1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。

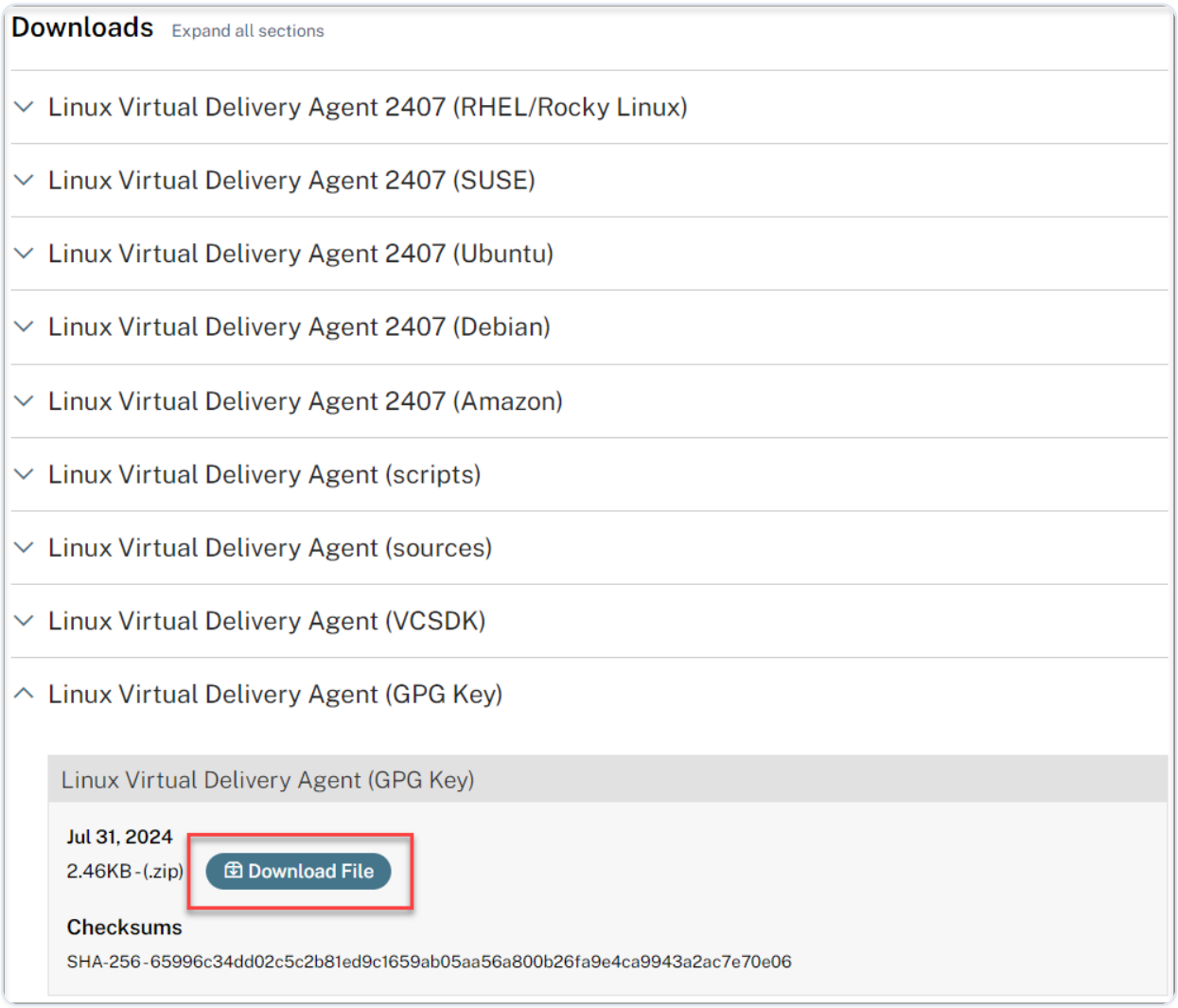
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. **Components**を展開してLinux VDAを見つけます。例：



4. Linux VDAリンクをクリックして、Linux VDAダウンロードにアクセスします。



5. お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：



Linux VDAパッケージの整合性を検証するには、以下のコマンドを実行して公開鍵をRPMデータベースにインポートし、パッケージの整合性を確認します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

ステップ6：Linux VDAをインストールする

新規インストールまたは既存のインストールのアップグレードが可能です。Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311へ、および1912 LTSRから2203 LTSRへアップグレードできます。

ステップ6a：新規インストールを実行する

1. （オプション）古いバージョンをアンインストールします。

   以前の2つのバージョンおよびLTSRリリース以外の古いバージョンをインストールしている場合は、新しいバージョンをインストールする前にアンインストールしてください。

a. Linux VDAサービスを停止します。

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

> **注:**
>
> **ctxvda**および**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

b. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

> **注:**
>
> コマンドを実行するには、フルパスが必要です。または、**/opt/Citrix/VDA/sbin**と**/opt/Citrix/VDA/bin**をシステムパスに追加することもできます。

2. Linux VDAパッケージをダウンロードします。

   Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

3. `Yum`を使用してLinux VDAソフトウェアをインストールします。

> **注:**

- RHELおよびRocky Linuxの場合、Linux VDAを正常にインストールする前にEPELリポジトリをインストールしてください。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>`の指示を参照してください。
- RHEL 9.4/9.2およびRocky Linux 9.4/9.2にLinux VDAをインストールする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

Amazon Linux 2 の場合:

```
sudo yum install -y XenDesktopVDA-<version>.amzn2.x86_64.rpm
```

RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 の場合:

```
sudo yum install -y XenDesktopVDA-<version>.el9_x.x86_64.rpm
```

RHEL 8.x および Rocky Linux 8.x の場合:

```
sudo yum install -y XenDesktopVDA-<version>.el8_x.x86_64.rpm
```

RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 の RPM 依存関係リスト:

```
gtk2 >= 2.24.33

java-17-openjdk >= 17

tzdata-java >= 2022

ImageMagick >= 6.9

firewalld >= 0.6.3

policycoreutils-python >= 2.8.9

policycoreutils-python-utils >= 2.8

python3-policycoreutils >= 2.8

dbus >= 1.12.8
```

```
dbus-common >= 1.12.8

dbus-daemon >= 1.12.8

dbus-tools >= 1.12.8

dbus-x11 >= 1.12.8

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12

libXrandr >= 1.5.1

libXtst >= 1.2.3

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6

gawk >= 4.2

sed >= 4.5

cups >= 1.6.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

ibus >= 1.5
```

```
nss-tools >= 3.44.0

cyrus-sasl-gssapi >= 2.1

python3 >= 3.6~

qt5-qtbase >= 5.5~

qt5-qtbase-gui >= 5.5~

qrencode-libs >= 3.4.4

imlib2 >= 1.4.9

fuse-libs >= 2.9

pulseaudio-utils >= 15.0
```

RHEL 8.x および Rocky Linux 8.x の RPM 依存関係リスト:

```
java-17-openjdk >= 17

ImageMagick >= 6.9

firewalld >= 0.6.3

policycoreutils-python >= 2.8.9

policycoreutils-python-utils >= 2.8

python3-policycoreutils >= 2.8

dbus >= 1.12.8

dbus-common >= 1.12.8

dbus-daemon >= 1.12.8

dbus-tools >= 1.12.8

dbus-x11 >= 1.12.8
```

```
xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12

libXrandr >= 1.5.1

libXtst >= 1.2.3

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6

gawk >= 4.2

depends_on sed >= 4.5

pulseaudio >= 14.0

pulseaudio-module-x11 >= 14.0

pulseaudio-module-bluetooth >= 14.0

alsa-plugins-pulseaudio >= 1.1.9

cups >= 1.6.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

ibus >= 1.5
```

```
nss-tools >= 3.44.0

gperftools-libs >= 2.4

cyrus-sasl-gssapi >= 2.1

python3 >= 3.6~

qt5-qtbase >= 5.5~

qt5-qtbase-gui >= 5.5~

qrencode-libs >= 3.4.4

imlib2 >= 1.4.9

fuse-libs >= 2.9
```

Amazon Linux 2 の RPM 依存関係リスト:

```
java-17-openjdk >= 17

ImageMagick >= 6.7.8.9

firewalld >= 0.3.9

policycoreutils-python >= 2.0.83

dbus >= 1.6.12

dbus-x11 >= 1.6.12

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

xorg-x11-server-Xorg >= 1.20.4

libXpm >= 3.5.10

libXrandr >= 1.4.1
```

```
libXtst >= 1.2.2

pam >= 1.1.8

util-linux >= 2.23.2

xorg-x11-utils >= 7.5

bash >= 4.2

findutils >= 4.5

gawk >= 4.0

sed >= 4.2

pulseaudio >= 10.0

pulseaudio-module-x11 >= 10.0

pulseaudio-module-bluetooth >= 10.0

pulseaudio-gdm-hooks >= 10.0

alsa-plugins-pulseaudio >= 1.1.1

cups >= 1.6.0

foomatic-filters >= 4.0.9

libxml2 >= 2.9

libmspack >= 0.5

ibus >= 1.5

cyrus-sasl-gssapi >= 2.1

gperftools-libs >= 2.4

nss-tools >= 3.44.0

qt5-qtbase >= 5.5~
```

```
qrencode-libs >= 3.4.1

imlib2 >= 1.4.5

fuse-libs >= 2.9
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

手順 6b: 既存のインストールをアップグレードする (オプション)

Linux VDA は、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDA を 2308 から 2311 に、1912 LTSR から 2203 LTSR にアップグレードできます。

Amazon Linux 2、RHEL、および Rocky Linux ディストリビューションの場合:

```
sudo yum -y localinstall <PATH>/<Linux VDA RPM>
```

注:

- 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、必ずファイルをバックアップしてください。
- RHEL 9.4/9.2 および Rocky Linux 9.4/9.2 で Linux VDA をアップグレードする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。
- 2407 リリース以降、Linux VDA は、アップグレード中の構成ファイルの処理をパッケージマネージャーの **rpm** または **dpkg** に委任します。以下に、rpm と dpkg が構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを `.rpmnew` 拡張子で保存します。

    - **dpkg**: 続行方法の選択肢を対話形式で求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを **.dpkg-new** または **.dpkg-dist** として保存しながら、Linux VDA をサイレントアップグレードするには、次のコマンドを使用します。

      ```
      dpkg --force-confold -i package.deb  # Always keep your version, then save new package's version as *.dpkg-new or *.dpkg-dist
      ```

- ソフトウェアのアップグレード後にLinux VDAマシンを再起動します。

ステップ7：NVIDIA GRIDドライバーをインストールする

HDX 3D Proを有効にするには、ハイパーバイザーとVDAマシンにNVIDIA GRIDドライバーをインストールする必要があります。

注:

Amazon Linux 2でHDX 3D Proを使用するには、NVIDIAドライバー470をインストールすることをお勧めします。詳細については、「システム要件」を参照してください。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、以下のガイドを参照してください。

- XenServer
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、次の手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. XenCenter®で、VMにGPUを割り当てます。
3. VMを起動します。
4. NVIDIA GRIDドライバー用にVMを準備します。

```
yum install gcc
```

```
yum install "kernel-devel-$(uname -r)"



systemctl set-default multi-user.target
```

5. NVIDIA GRIDドライバーをインストールするには、Red Hat Enterprise Linuxドキュメントの手順に従ってください。

注:

GPUドライバーのインストール中に、各質問に対してデフォルト（「いいえ」）を選択します。

重要:

GPUパススルーが有効になると、Linux VMはXenCenter経由でアクセスできなくなります。SSHを使用して接続してください。

```
nvidia-smi

+-----------------------------------------------------+
| NVIDIA-SMI 352.70     Driver Version: 352.70        |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla M60           Off  | 0000:00:05.0     Off |                  Off |
| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

高解像度とマルチモニター機能を利用するには、有効なNVIDIAライセンスが必要です。ライセンスを申請するには、「GRID Licensing Guide.pdf - DU-07757-001 September 2015」の製品ドキュメントに従ってください。

手順8：Linux VDAを構成する

> **注:**
>
> ランタイム環境をセットアップする前に、**en_US.UTF-8**ロケールがOSにインストールされていることを確認してください。ロケールがOSで利用できない場合は、**sudo locale-gen en_US.UTF-8**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集し、**# en_US.UTF-8 UTF-8**行のコメントを解除してから、**sudo locale-gen**コマンドを実行します。

> パッケージのインストール後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションを環境変数で指定します。必要な変数がすべて存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンを ドメインに参加させるかどうか。デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。

- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、値を**'y'**に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーション技術のセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=**'y'**）に構成されます。
- **CTX_XDL_START_SERVICE='y\|n'** – 構成が完了したときにLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – マシンの起動後にLinux Virtual Desktopサービスが開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDAサービスは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux Virtual Desktopの場合、必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または**Xfce**デスクトップ環境を指定します。**'<none\>'**に設定すると、VDAに構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用してデスクトップ環境を切り替えることもできます。詳細については、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（ctxvda）をサポートするために.NETをインストールするパス。デフォルトのパスは**'/usr/bin'**です。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。

- **CTX_XDL_SITE_NAME=<dns-name\>** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。不要な場合は、**'<none\>'**に設定します。
- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDAはDNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合は、LDAPポートを含むLDAP FQDNのスペース区切りリストを提供できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数はデフォルトで**'<none\>'**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPにクエリを実行します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**'<none\>'**に設定します。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y\|n'** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
```

```
export CTX_XDL_VDA_PORT='<port-number>'

export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'

export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'

export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'

export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

sudoコマンドを実行するときは、**-E**オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。上記のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \
```

```
CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \

CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

/opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

構成変更の削除

特定のシナリオでは、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトはコンソールにエラーを表示し、追加情報は構成ログファイル **/tmp/xdl.configure.log** に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

手順9：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、**XDPing**を参照してください。

手順10：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成した後、次のコマンドを実行してLinux VDAを制御できます。

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

> **注:**
>
> **ctxvda**サービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

> **Linux VDAを再起動する：**

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx
```

```
sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda


sudo systemctl status ctxhdx
```

ステップ11：マシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、マシンカタログを作成するおよびマシンカタログを管理するを参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**Multi-session OS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**Single-session OS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホストされた共有デスクトップ配信モデルが暗示されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに単一ユーザーの配信モデルが暗示されます。

ヒント:

削除されたマシンをActive Directoryドメインに再参加させる場合は、そのマシンをマシンカタログから削除し、再度追加してください。

ステップ 12: デリバリーグループを作成する

Linux VDAマシンを含むマシンカタログを作成し、デリバリーグループに追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループを作成する」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

重要:

アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2411」を参照してください。

SUSE に Linux VDA を手動でインストールする

重要:

新規インストールの場合、迅速なインストールのために簡易インストールを使用することをお勧めします。簡易インストールは、時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: YaST ツールを起動する

SUSE Linux Enterprise YaST ツールは、オペレーティングシステムのあらゆる側面を構成するために使用されます。

テキストベースの YaST ツールを起動するには:

```
su -



yast
```

UI ベースの YaST ツールを起動するには:

```
su -



yast2 &
```

ステップ 1b: ネットワークを構成する

以下のセクションでは、Linux VDA で使用されるさまざまなネットワーク設定とサービスを構成する方法について説明します。ネットワークの構成は、Network Manager などの他の方法ではなく、YaST ツールを介して実行されます。これらの手順は、UI ベースの YaST ツールを使用することを前提としています。テキストベースの YaST ツールも使用できますが、ここでは説明されていない異なるナビゲーション方法があります。

ホスト名とドメインネームシステム (DNS) を構成する

1. UI ベースの YaST ツールを起動します。
2. **System** を選択し、次に **Network Settings** を選択します。
3. **Hostname/DNS** タブを開きます。
4. **Set Hostname via DHCP** の **no** オプションを選択します。
5. **Modify DNS Configuration** の **Use Custom Policy** オプションを選択します。
6. ネットワーク設定を反映するように、以下を編集します。

    - **静的ホスト名** – マシンのDNSホスト名を追加します。

- **ネームサーバー** – DNSサーバーのIPアドレスを追加します。通常、これはADドメインコントローラーのIPアドレスです。
- **ドメイン検索リスト** – DNSドメイン名を追加します。

7. `/etc/hosts` ファイルの次の行を変更して、FQDNとホスト名を最初の2つのエントリとして含めます。

   ```
   127.0.0.1 <FQDN of the VDA> <hostname of the VDA> localhost
   ```

> **注:**
>
> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。したがって、ホスト名は15文字を超えてはなりません。 **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダースコア (_)、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは、デリバリーコントローラーのホスト名にも適用されます。

ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、完全修飾ドメイン名 (FQDN) は返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

名前解決とサービス到達性を確認する

FQDNを解決し、ドメインコントローラーとデリバリーコントローラー™にpingできることを確認します。

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

FQDNを解決できない場合、またはこれらのマシンのいずれかにpingできない場合は、続行する前に手順を確認してください。

ステップ 1c: NTP サービスを構成する

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することが重要です。Linux VDA を仮想マシン (VM) としてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモート NTP サービスを使用して時刻を維持することが推奨されます。デフォルトの NTP 設定にいくつかの変更が必要になる場合があります。

SUSE 15.6 の場合:

1. UI ベースの YaST ツールを起動します。
2. **［ネットワークサービス］** を選択し、次に **［NTP 設定］** を選択します。
3. **［NTP デーモンの開始］** セクションで、**［今すぐ起動してブート時にも起動］** を選択します。
4. **［構成ソース］** に **［動的］** を選択します。
5. 必要に応じて NTP サーバーを追加します。NTP サービスは通常、Active Directory ドメインコントローラーでホストされます。
6. /etc/chrony.conf に以下の行が存在する場合は、削除またはコメントアウトします。

   `include /etc/chrony.d/*.conf`

   chrony.conf を編集した後、`chronyd` サービスを再起動します。

   ```
   sudo systemctl restart chronyd.service
   ```

ステップ 1d: Linux VDA の依存パッケージをインストールする

SUSE Linux Enterprise 用の Linux VDA ソフトウェアは、以下のパッケージに依存します。

- Open Motif ランタイム環境 2.3.1 以降
- Cups 1.6.0 以降
- ImageMagick 6.8 以降

リポジトリを追加する

ほとんどの必須パッケージ（ImageMagickを除く）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クライアントをインストールする

Linux VDAとDelivery Controller間の相互認証のためにKerberosクライアントをインストールします。

```
sudo zypper install krb5-client
```

Kerberosクライアントの設定は、使用するActive Directory統合アプローチによって異なります。以下の説明を参照してください。

使用するデータベースをインストールして指定する

注:

- VDIモードでのみSQLiteを使用し、ホスト型共有デスクトップ配信モデルにはPostgreSQLを使用することをお勧めします。
- 簡単インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用するよう指定できます。**/etc/xdl/db.conf**で別途指定されていない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、`/etc/xdl/db.conf`を編集して新しいバージョンを反映させ、簡単インストールスクリプト（`ctxinstall.sh`）またはMCSスクリプト（`deploymcs.sh`）を実行する前にPostgreSQLサービスを開始する必要があります。

- 手動インストールの場合、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。Linuxディストリビューションが提供するバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

PostgreSQLをインストールする
このセクションでは、Linuxディストリビューションが提供するPostgreSQLのバージョンをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

`Postgresql`をインストールするには、次のコマンドを実行します。

```
sudo zypper install postgresql-server



sudo zypper install postgresql-jdbc
```

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLiteをインストールする
SUSEの場合、SQLiteをインストールするには、次のコマンドを実行します。

```
sudo zypper install sqlite3
```

使用するデータベースを指定します
SQLiteとPostgreSQLの両方をインストールした場合、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh**を実行します。新規インストールの場合、この手順は省略してください。

2. 使用するデータベースを指定するために**/etc/xdl/db.conf**を編集します。以下は**db.conf**ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:
```

```
#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL**をtrueに設定します。

3. **ctxsetup.sh**を実行します。

注:

/etc/xdl/db.confを使用して、PostgreSQLのポート番号を構成することもできます。

ステップ2：ハイパーバイザーを準備する

サポートされているハイパーバイザー上でLinux VDAをVMとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに基づいて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

XenServer（旧Citrix Hypervisor™）での時刻同期を修正する

XenServer®の時刻同期機能が有効になっている場合、準仮想化された各Linux VM内でNTPとXenServerの両方がシステムクロックを管理しようとするため、問題が発生します。他のサーバーとのクロックのずれを防ぐには、各Linuxゲスト内のシステムクロックをNTPと同期させます。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

XenServer VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からXenServerの時刻同期機能が存在し、有効になっているかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能が無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合は、ファイルに **1** を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を再起動後も永続的にするには、**/etc/sysctl.conf** ファイルを編集し、次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
reboot
```

再起動後、設定が正しいことを確認します。

```
su -
```

```
cat /proc/sys/xen/independent_wallclock
```

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux Integration Services がインストールされている Linux VM は、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、この機能を NTP サービスと同時に有効にしてください。

管理オペレーティングシステムから：

1. Hyper-V マネージャーコンソールを開きます。
2. Linux VM の設定で、**統合サービス** を選択します。
3. **時刻同期** が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTP との競合を避けるためにホストの時刻同期が無効になっている VMware や XenServer (旧 Citrix Hypervisor) とは異なります。Hyper-V の時刻同期は、NTP の時刻同期と共存し、補完することができます。

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP とハイパーバイザーとの間で問題が発生します。両方ともシステムクロックを同期しようとします。他のサーバーとのクロックのずれを避けるために、各 Linux ゲスト内のシステムクロックを NTP と同期させます。この場合、ホストの時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合：

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **Virtual Machine Properties**ダイアログで、**Options**タブを開きます。
4. **VMware Tools**を選択します。
5. **Advanced**ボックスで、**Synchronize guest time with host**のチェックを外します。

ステップ3：Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory（AD）ドメインに追加するには、次の方法があります。

- Samba Winbind (サンバ・ウィンバインド)
- セントリファイ ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名を使用すると、セッションの起動が失敗する可能性があります。

Samba Winbind (サンバ・ウィンバインド)

WINDOWSドメインに参加

ドメインコントローラーに到達可能であり、マシンをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

1. YaSTを起動し、**ネットワークサービス**、次に**Windowsドメインメンバーシップ**を選択します。
2. 次の変更を行います。
   - **ドメインまたはワークグループ**をActive Directoryドメインの名前またはドメインコントローラーのIPアドレスに設定します。ドメイン名が大文字であることを確認してください。
   - **Linux認証にSMB情報を使用**をオンにします。
     - **ログイン時にホームディレクトリを作成**をオンにします。
     - **SSHのシングルサインオン**をオンにします。
     - **オフライン認証**がオフになっていることを確認します。このオプションはLinux VDAと互換性がありません。
3. **OK**をクリックします。一部のパッケージのインストールを求められた場合は、**インストール**をクリックします。

4. ドメインコントローラーが見つかった場合、ドメインに参加するかどうかを尋ねられます。**はい**をクリックします。
5. プロンプトが表示されたら、マシンをドメインに追加する権限を持つドメインユーザーの資格情報を入力し、**OK**をクリックします。
6. サービスを手動で再起動するか、マシンを再起動します。マシンを再起動することをお勧めします。

```
su -

reboot
```

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っていることを要求します。

マシンがドメインに参加していることを確認するには、**Samba**の**net ads**コマンドを実行します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

KERBEROS構成の確認

システムkeytabファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist –ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェル置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNS ドメイン名が Kerberos レルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

次を使用してマシンアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

ドメインユーザーがドメインで認証できることを確認するには、**wbinfo**ツールを使用します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字を別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認します。これを行うには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -uコマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効で期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

Gnome または KDE コンソールに直接ログオンすることで、同様のテストを実行できます。ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/manual-installation-overview/suse.html#step-6-install-the-linux-vda)に進んでください。

Quest 認証サービス

ドメインコントローラーでのQUESTの構成

ドメインコントローラーにQuestソフトウェアをインストールして構成済みであり、`Active Directory`でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unixを有効にする**をチェックします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDAでのQUESTの構成

VASデーモンの構成
Kerberosチケットの自動更新を有効にし、切断する必要があります。認証（オフラインログオン）は無効にする必要があります：

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間のチケット有効期間より1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。
PAM と NSS を構成する
HDX および su、ssh、RDP などの他のサービスを介したドメインユーザーログオンを有効にするには、PAM と NSS を手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

WINDOWS ドメインに参加する

Quest `vastool` コマンドを使用して、Linux マシンを Active Directory ドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

user は、マシンを Active Directory ドメインに参加させる権限を持つドメインユーザーです。**domain-name** は、ドメインの DNS 名です（例: example.com）。

ドメイン参加後、Linux マシンを再起動します。

ドメインメンバーシップを確認する

Delivery Controller は、すべての VDA マシン（Windows および Linux VDA）が `Active Directory` にコンピューターオブジェクトを持つことを要求します。Quest に参加している Linux マシンがドメイン上にあることを確認するには、次の手順を実行します。

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます。

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証を確認する

Quest が PAM を介してドメインユーザーを認証できることを確認します。これを行うには、以前使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username

id -u
```

id -u コマンドによって返された uid に対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos 資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンすることでも実行できます。ドメイン参加の検証後、手順 6: Linux VDA をインストールする に進みます。

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

Centrify DirectControl Agentがインストールされている場合、Centrifyの**adjoin**コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
sudo adjoin -w -V -u user domain-name
```

userは、マシンをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name**は、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持つことを要求します。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
sudo adinfo
```

Joined to domainの値が有効であることを確認し、**CentrifyDC**の**mode**が**connected**を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

SSSD

SUSEでSSSDを使用している場合は、このセクションの指示に従ってください。このセクションには、Linux VDAマシンをWindowsドメインに参加させるための手順が含まれており、Kerberos認証を構成するためのガイダンスを提供します。

SUSEでSSSDをセットアップするには、次の手順を完了します。

1. ドメインに参加し、ホストキータブを作成します
2. SSSD用にPAMを構成します

3. SSSD をセットアップする
4. SSSD を有効にする
5. ドメインメンバーシップを確認する
6. Kerberos の構成を確認する
7. ユーザー認証を確認する

ドメインに参加し、ホストキータブを作成する

SSSD は、ドメインへの参加やシステムキータブファイルの管理のための Active Directory クライアント機能を提供しません。代わりに、**Samba** のアプローチを使用できます。SSSD を構成する前に、以下の手順を完了してください。

1. Name Service Cache Daemon (NSCD) デーモンを停止して無効にします。

```
sudo systemctl stop nscd

sudo systemctl disable nscd
```

2. ホスト名と Chrony の時刻同期を確認します。

```
hostname

hostname -f

chronyc traking
```

3. 必要なパッケージをインストールまたは更新します。

```
sudo zypper install samba-client sssd-ad
```

4. ルートユーザーとして `/etc/krb5.conf` ファイルを編集し、**kinit** ユーティリティがターゲットドメインと通信できるようにします。、、および セクションの下に、次のエントリを追加します。

> **注:**
>
> AD インフラストラクチャに基づいて Kerberos を構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]

    dns_canonicalize_hostname = false

    rdns = false

    default_realm = REALM

    forwardable = true

[realms]

    REALM = {

        kdc = fqdn-of-domain-controller

        default_domain = realm

        admin_server = fqdn-of-domain-controller
    }
[domain_realm]
```

```
    .realm = REALM
```

realm は Kerberos レルム名（例：example.com）です。**REALM** は大文字の Kerberos レルム名（例：EXAMPLE.COM）です。

5. ルートユーザーとして `/etc/samba/smb.conf` を編集し、**net** ユーティリティがターゲットドメインと通信できるようにします。 セクションの下に、次のエントリを追加します。

```
[global]

    workgroup = domain



    client signing = yes



    client use spnego = yes



    kerberos method = secrets and keytab



    realm = REALM



    security = ADS
```

domain は Active Directory ドメインの短い NetBIOS 名（例：EXAMPLE）です。

6. ユーザーとグループを解決する際にSSSDを参照するように、**/etc/nsswitch.conf** ファイルの **passwd** および **group** エントリを変更します。

```
passwd: compat sss
```

```
group: compat sss
```

7. 構成済みのKerberosクライアントを使用して、ターゲットドメインに管理者として認証します。

```
kinit administrator
```

8. **net** ユーティリティを使用して、システムをドメインに参加させ、システムキータブファイルを生成します。

```
net ads join osname="SUSE Linux Enterprise Server" osVersion=15 -U
administrator
```

SSSD 用の PAM を構成する

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。

```
sudo zypper install sssd sssd-ad
```

SSSD を介したユーザー認証のために PAM モジュールを構成し、ユーザーログオン用のホームディレクトリを作成します。

```
sudo pam-config --add  --sss

sudo pam-config --add --mkhomedir
```

SSSD をセットアップする

1. SSSD デーモンがターゲットドメインと通信できるように、ルートユーザーとして `/etc/sssd/sssd.conf` を編集します。`sssd.conf` の構成例（必要に応じて追加オプションを追加できます）：

```
[sssd]

    config_file_version = 2

    services = nss,pam

    domains = domain-dns-name
```

```
[domain/domain-dns-name]
    id_provider = ad
    auth_provider = ad
    access_provider = ad
    ad_domain = domain-dns-name
    ad_server = fqdn-of-domain-controller
    ldap_id_mapping = true
    ldap_schema = ad

# Kerberos settings
    krb5_ccachedir = /tmp
    krb5_ccname_template = FILE:%d/krb5cc_%U

# Comment out if the users have the shell and home dir set on the AD
side

    fallback_homedir = /home/%d/%u
    default_shell = /bin/bash

# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available
```

```
# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM



    ad_gpo_access_control = permissive
```

domain-dns-name は、example.com などの DNS ドメイン名です。

> **注:**
>
> **ldap_id_mapping** は true に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングします。そうでない場合、Active Directory は POSIX 拡張機能を提供できる必要があります。Linux セッションでの無効なログオンエラーを防ぐために、**ad_gpo_access_control** は **permissive** に設定されています。`sssd.conf` および `sssd-ad` の man ページを参照してください。

2. `sssd.conf` のファイルの所有権とアクセス許可を設定します。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSD を有効にする

システム起動時に SSSD デーモンを有効にして開始するには、次のコマンドを実行します。

```
sudo systemctl enable sssd

sudo systemctl start sssd
```

ドメインメンバーシップを確認する

1. マシンがドメインに参加していることを確認するには、**Samba** の `net ads` コマンドを実行します。

```
sudo net ads testjoin
```

2. 追加のドメインおよびコンピューターオブジェクト情報を検証するには、次のコマンドを実行します。

```
sudo net ads info
```

KERBEROS構成を検証する

システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。

これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberosの**kinit**コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。シェル置換を防ぐために、ドル記号（**$**）はバックスラッシュ（****）でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを、次を使用して確認します。

```
sudo klist
```

ユーザー認証を検証する

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

SSSD PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

```
klist



exit
```

`klist`コマンドによって返されたKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

rootユーザーとして、以前の`id -u`コマンドによって返されたuidに対応するチケットキャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、手順6：Linux VDAをインストールするに進みます。

PBIS

必要なPBISパッケージをダウンロードする

例:

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/
pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行可能にする

例:

```
chmod +x pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBISインストールスクリプトを実行する

例:

```
sh pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

WINDOWSドメインに参加する

ドメインコントローラーに到達可能であり、マシンをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
/opt/pbis/bin/domainjoin-cli join domain-name user
```

userは、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名です（例: example.com）。

注：Bashをデフォルトシェルとして設定するには、/opt/pbis/bin/config LoginShellTemplate/bin/bashコマンドを実行します。

ドメインメンバーシップを確認する

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持つことを要求します。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。そうでない場合は、ホスト名のみが表示されます。

ユーザー認証を確認する

PBISがPAMを介してドメインユーザーを認証できることを確認します。これを行うには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

id -uコマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順6：Linux VDAをインストールするに進みます。

手順4：.NETをインストールする

.NET Runtime に加えて、Linux VDA をインストールまたはアップグレードする前に、サポートされているすべての Linux ディストリビューションに .ASP.NET Core Runtime をインストールする必要があります。Amazon Linux 2 にはバージョン 6 が必要です。その他のディストリビューションにはバージョン 8 が必要です。

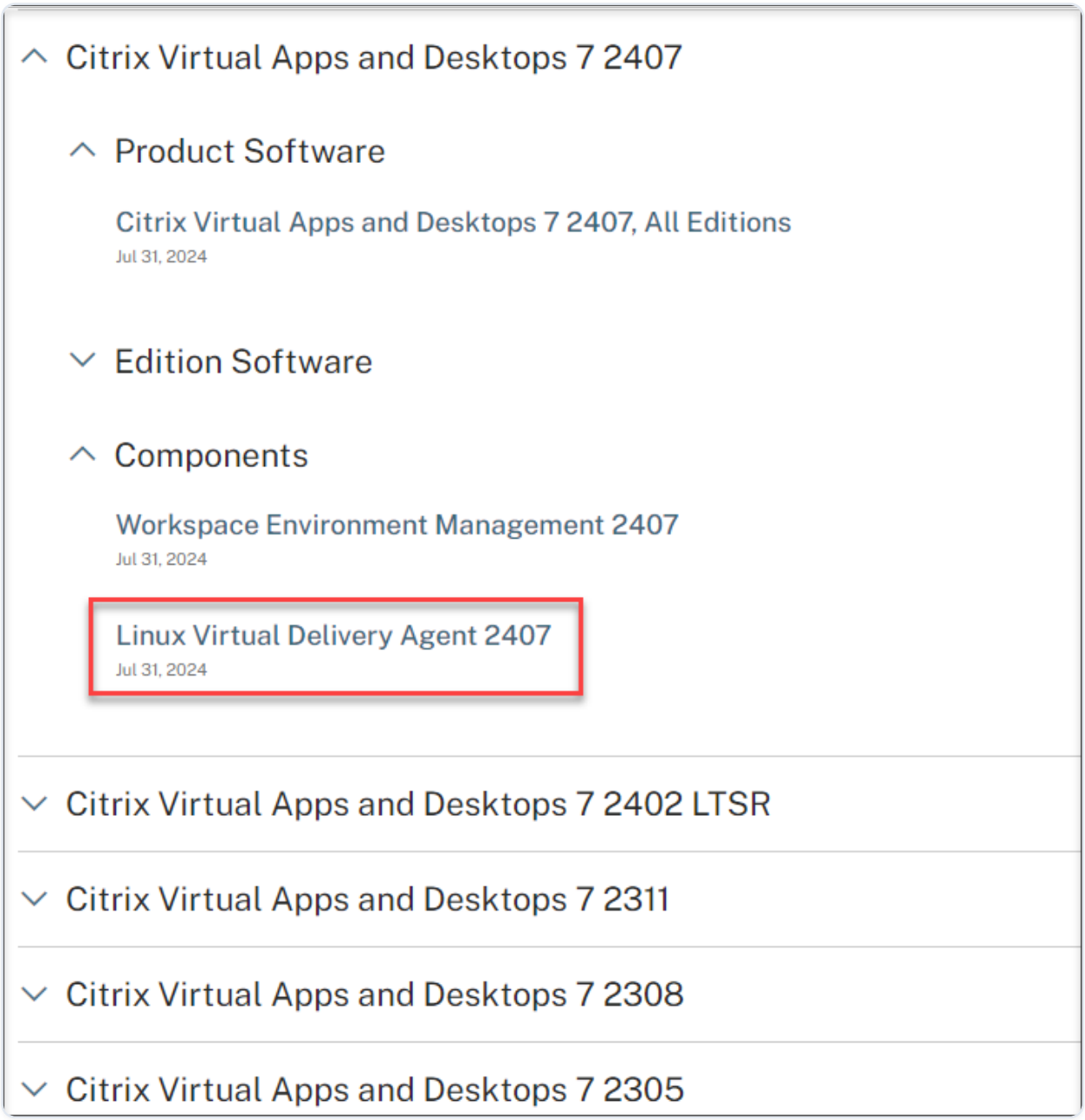
お使いの Linux ディストリビューションに必要な .NET バージョンが含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoft パッケージフィードから .NET をインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを見つけます。

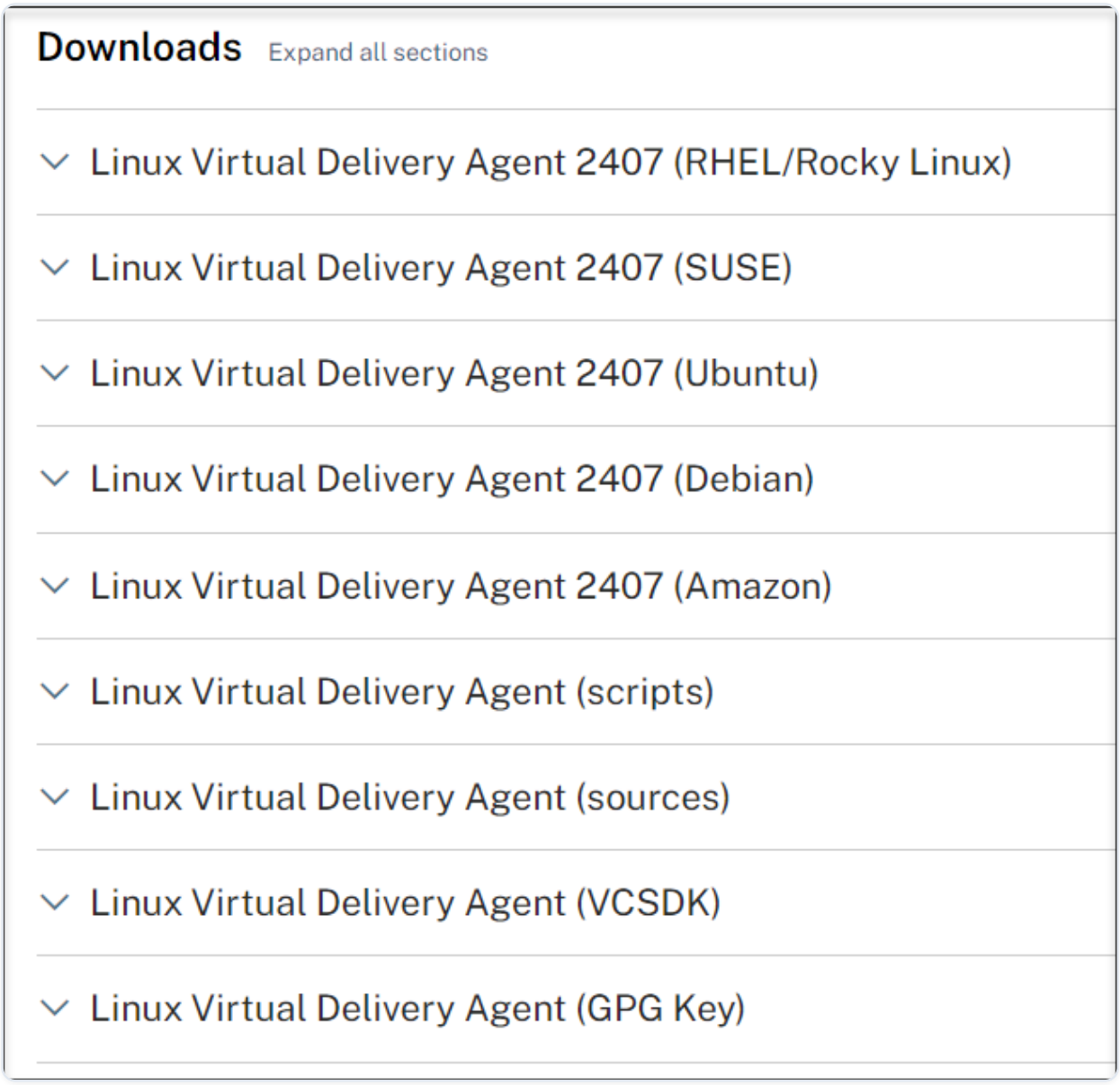
コマンド出力に基づいて、.NET ランタイムバイナリパスを設定します。たとえば、コマンド出力が /aa/bb/dotnet の場合、/aa/bb を .NET バイナリパスとして使用します。

手順 5：Linux VDA パッケージをダウンロードする

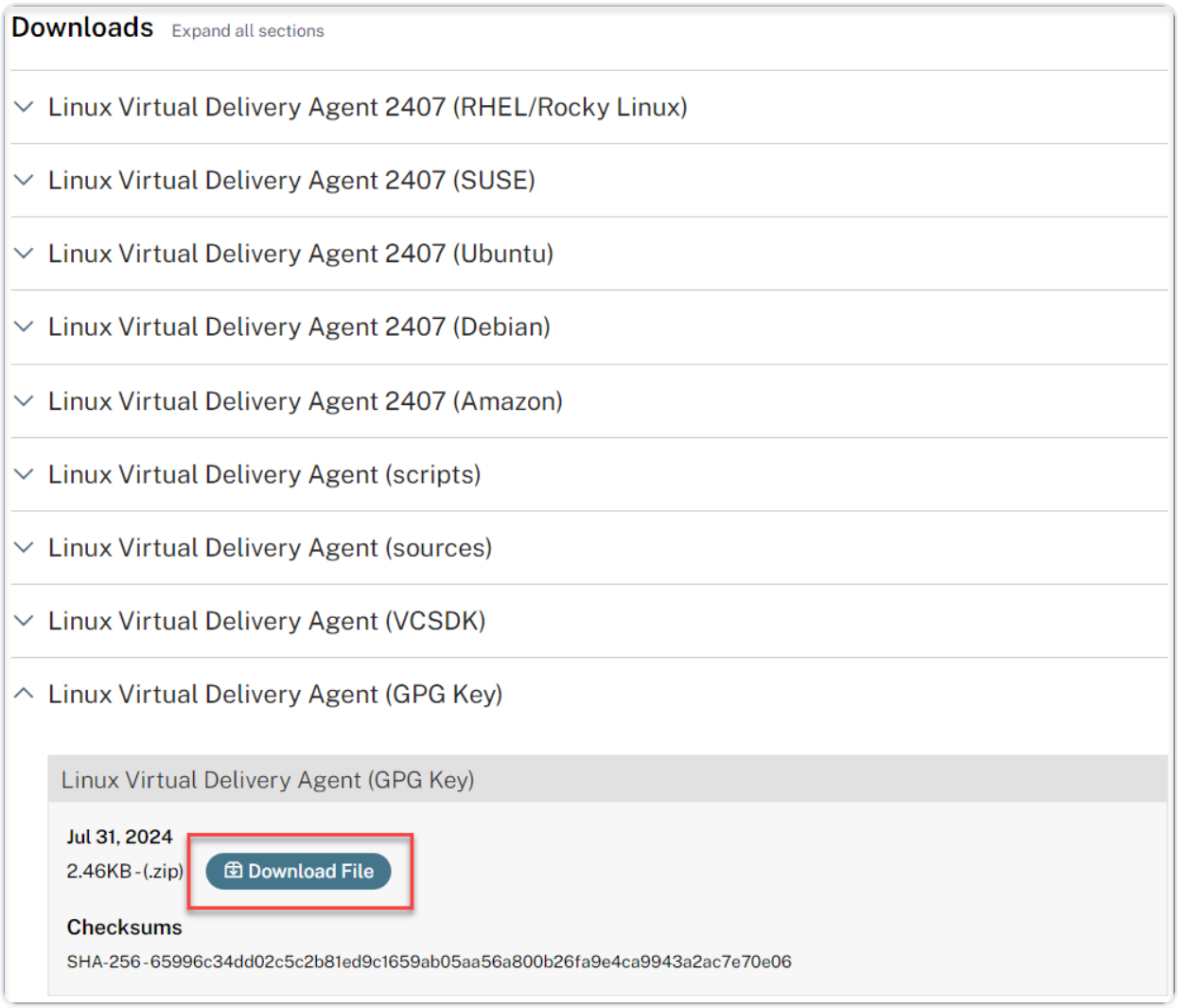
1. Citrix Virtual Apps and Desktops ダウンロードページにアクセスします。
2. Citrix Virtual Apps and Desktops の適切なバージョンを展開します。
3. **Components** を展開して Linux VDA を見つけます。例：



4. Linux VDA リンクをクリックして、Linux VDA のダウンロードにアクセスします。



5. お使いの Linux ディストリビューションに一致する Linux VDA パッケージをダウンロードします。
6. Linux VDA パッケージの整合性を検証するために使用できる GPG 公開鍵をダウンロードします。例：



公開鍵を使用して Linux VDA パッケージの整合性を検証するには、次のコマンドを実行して公開鍵を RPM データベースにインポートし、パッケージの整合性を確認します。

```
rpmkeys --import <path to the public key>

rpm --checksig --verbose <path to the Linux VDA package>
```

ステップ6: Linux VDAをインストールする

ステップ6a: 古いバージョンをアンインストールする

以前の2つのバージョンおよびLTSRリリース以外の古いバージョンをインストールしている場合は、新しいバージョンをインストールする前にアンインストールしてください。

1. Linux VDAサービスを停止します:

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

注:

ctxvdaサービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

2. パッケージをアンインストールします:

```
sudo rpm -e XenDesktopVDA
```

重要:

最新の2つのバージョンからのアップグレードはサポートされています。

注:

インストールされているコンポーネントは **/opt/Citrix/VDA/** にあります。

コマンドを実行するにはフルパスが必要です。または、**/opt/Citrix/VDA/sbin**と**/opt/Citrix/VDA/bin**をシステムパスに追加することもできます。

ステップ6b: Linux VDAをインストールする

Zypperを使用してLinux VDAソフトウェアをインストールします:

```
sudo zypper install XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

ステップ6c: Linux VDAをアップグレードする (オプション)

Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311へ、および1912 LTSRから2203 LTSRへアップグレードできます。

> **注:**
>
> 既存のインストールをアップグレードすると、/etc/xdlの下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、ファイルをバックアップしてください。

```
sudo zypper -i install <PATH>/<Linux VDA RPM>
```

SUSE 15のRPM依存関係リスト:

```
java-17-openjdk >= 17

ImageMagick >= 7.0

dbus-1 >= 1.12.2

dbus-1-x11 >= 1.12.2

xorg-x11 >= 7.6_1

libXpm4 >= 3.5.12

libXrandr2 >= 1.5.1

libXtst6 >= 1.2.3

pam >= 1.3.0

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.4

cups >= 2.2
```

```
cups-filters >= 1.25

libxml2-2 >= 2.9

libmspack0 >= 0.6

ibus >= 1.5

libQt5DBus5 >= 5.12

libtcmalloc4 >= 2.5

libcap-progs >= 2.26

mozilla-nss-tools >= 3.53.1

libpython3_6m1_0 >= 3.6~

libQt5Widgets5 >= 5.12

libqrencode4 >= 4.0.0

libImlib2-1 >= 1.4.10

libgtk-2_0-0 >= 2.24

libgthread-2_0-0 >= 2.48

pulseaudio-utils >= 5.0

lsb-release >= 2.0

pkexec >= 121

cyrus-sasl-gssapi >= 2.1

libfuse2 >= 2.9
```

重要:

アップグレード後にLinux VDAマシンを再起動します。

ステップ7：NVIDIA GRIDドライバーをインストールする

HDX 3D Proを有効にするには、ハイパーバイザーとVDAマシンにNVIDIA GRIDドライバーをインストールする必要があります。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、以下のガイドを参照してください。

- XenServer
- ヴイエムウェア イーエスエックス
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、以下の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバー（クラウドベンダーまたはNVIDIAから提供）をインストールします。

ステップ8：Linux VDAを構成する

> **注:**
>
> ランタイム環境をセットアップする前に、OSに**en_US.UTF-8**ロケールがインストールされていることを確認してください。OSでロケールが利用できない場合は、**sudo locale-gen en_US.UTF-8**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集して**# en_US.UTF-8 UTF-8**行のコメントを解除し、**sudo locale-gen**コマンドを実行します。

> パッケージをインストールした後、`ctxsetup.sh`スクリプトを実行してLinux VDAを構成する必要があります。スクリプトが変更を加える前に、環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh –help
```

プロンプトによる構成

プロンプトが表示される手動構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションを環境変数で指定します。必要なすべての変数が存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンをドメインに結合するかどうか。 デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合は、値を**'y'**に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=**'y'**）用に構成されます。
- **CTX_XDL_START_SERVICE='y\|n'** – 構成完了時にLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – Linux Virtual Desktopサービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDAサービスは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux Virtual Desktop

の場合、必要なポート（デフォルトではポート80および1494）をシステムファイアウォールで自動的に開くことができます。

- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate/'<none\>'** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。**'<none\>'**に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。

- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（ctxvda）をサポートするための.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 ポートを介して Delivery Controller と通信します。

- **CTX_XDL_SITE_NAME=<dns-name\>** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに限定するには、DNS サイト名を指定します。不要な場合は、**'<none\>'** に設定します。

- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDA は DNS にクエリを実行して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directory フォレスト内で LDAP クエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連する LDAP ポート番号を 3268 として指定します。この変数は、デフォルトで **'<none\>'** に設定されています。

- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介して LDAP にクエリを実行します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**'<none\>'** に設定します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y\|n'** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'

export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'

export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'

export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

sudo コマンドを実行するときは、既存の環境変数を新しく作成するシェルに渡すために **-E** オプションを入力します。上記のコマンドから、最初の行に **#\!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=winbind|centrify|sssd|pbis|quest \
```

```
CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \

CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \

CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

/opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

構成の変更を削除する

シナリオによっては、Linux VDA パッケージをアンインストールせずに、**ctxsetup.sh** スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /usr/local/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /usr/local/sbin/ctxcleanup.sh
```

> **重要:**
>
> このスクリプトは、データベースからすべての構成データを削除し、Linux VDA を動作不能にします。

構成ログ

ctxsetup.sh および **ctxcleanup.sh** スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイルに書き込まれます。

`/tmp/xdl.configure.log`

変更を有効にするには、Linux VDAサービスを再起動します。

ステップ9: XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、**XDPing**を参照してください。

ステップ10: Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成した後、以下のコマンドを実行してLinux VDAを制御できます。

Linux VDAを開始する:

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDAを停止する:

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda
```

```
sudo systemctl stop ctxhdx
```

注:

ctxvdaサービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDAを再起動する:

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する:

Linux VDAサービスの実行中のステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ11: マシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成する場合とは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**Multi-session OS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**Single-session OS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

> **注:**
>
> Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホストされた共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに1人のユーザーという配信モデルが意味されます。

> **ヒント:**
>
> マシンをActive Directoryドメインから削除して再参加させる場合は、マシンをマシンカタログから削除して再度追加する必要があります。

手順12：デリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンするように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

> **重要:**
>
> アプリケーションの公開はLinux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは同じマシンへのデスクトップとアプリの配信をサポートしていません。

> マシンカタログとデリバリーグループの作成方法については、Citrix Virtual Apps and Desktops 7 2411を参照してください。

Ubuntu に Linux VDA を手動でインストールする

> **重要:**
>
> 新規インストールの場合、迅速なインストールには easy install を使用することをお勧めします。Easy install は時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: ネットワーク構成を確認する

ネットワークが正しく接続され、構成されていることを確認してください。たとえば、Linux VDA で DNS サーバーを構成する必要があります。

Ubuntu Live Server を使用している場合は、ホスト名を設定する前に、**/etc/cloud/cloud.cfg** 構成ファイルで次の変更を行ってください。

```
preserve_hostname: true
```

ステップ 1b: ホスト名を設定する

マシンのホスト名が正しく報告されるように、**/etc/hostname** ファイルをマシンのホスト名のみを含むように変更します。

```
hostname
```

ステップ 1c: ホスト名にループバックアドレスを割り当てる

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されることを確認してください。その方法は、**/etc/hosts** ファイルの次の行を変更して、FQDN とホスト名を最初の 2 つのエントリとして含めることです。

`127.0.0.1 hostname-fqdn hostname localhost`

例:

`127.0.0.1 vda01.example.com vda01 localhost`

ファイル内の他のエントリから `hostname-fqdn` または `hostname` への他の参照をすべて削除します。

> **注:**
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダーバー (_)、スペース、およびその他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは Delivery Controller のホスト名にも適用されます。

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認してください。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: マルチキャストDNSを無効にする

デフォルト設定ではマルチキャストDNS (**mDNS**) が有効になっており、名前解決の結果が不安定になる可能性があります。

mDNSを無効にするには、**/etc/nsswitch.conf**を編集し、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

変更後:

`hosts: files dns`

ステップ1f: 名前解決とサービスの到達可能性を確認する

FQDNを解決し、ドメインコントローラーとDelivery Controller™にpingできることを確認します。

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn



nslookup delivery-controller-fqdn



ping delivery-controller-fqdn
```

FQDNを解決できない場合、またはこれらのマシンのいずれかにpingできない場合は、続行する前に手順を確認してください。

ステップ1g: クロック同期を構成する (chrony)

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することは非常に重要です。Linux VDAを仮想マシン (VM) としてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモート時刻サービスとの時刻同期が推奨されます。

chronyをインストールします。

```
apt-get install chrony
```

rootユーザーとして、**/etc/chrony/chrony.conf**を編集し、各リモートタイムサーバーのサーバーエントリを追加します。

`server peer1-fqdn-or-ip-address iburst`

`server peer2-fqdn-or-ip-address iburst`

通常の展開では、パブリックNTPプールサーバーから直接ではなく、ローカルのドメインコントローラーから時刻を同期します。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

ループバックIPアドレス、localhost、およびパブリックサーバーの***.pool.ntp.org**エントリを含む、リストされている他の**サーバー**または**プール**エントリをすべて削除します。

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h：使用するデータベースをインストールして指定する

注:

- VDIモードではSQLiteを、ホストされた共有デスクトップ配信モデルではPostgreSQLを使用することをお勧めします。
- 簡単インストールおよびMCSの場合、SQLiteまたはPostgreSQLを手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf**で別途指定されていない限り、Linux VDAはデフォルトでPostgreSQLを使用します。Linuxディストリビューションで提供されているバージョンではなく、カスタムバージョンのPostgreSQLが必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように`/etc/xdl/db.conf`を編集し、簡単インストールスクリプト (`ctxinstall.sh`) またはMCSスクリプト (`deploymcs.sh`) を実行する前にPostgreSQLサービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。Linuxディストリビューションで提供されているバージョンではなく、カスタムバージョンのPostgreSQLを使用できます。SQLiteとPostgreSQLの両方をインストールする場合は、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

POSTGRESQLをインストールする

このセクションでは、Linuxディストリビューションで提供されているPostgreSQLのバージョンをインストールする方法について説明します。カスタムバージョンのPostgreSQLが必要な場合は、特定の要件に基づいてインストールできます。

PostgreSQLをインストールするには、次のコマンドを実行します。

```
sudo apt-get install -y postgresql



sudo apt-get install -y libpostgresql-jdbc-java
```

マシンの起動時または即座にPostgreSQLを開始するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLITEをインストールする

Ubuntuの場合、SQLiteをインストールするには、次のコマンドを実行します。

```
sudo apt-get install -y sqlite3
```

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は、Linux VDAパッケージのインストール後に**/etc/xdl/db.conf**を編集して、どちらか一方を使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh**を実行します。新規インストールの場合、この手順は省略してください。
2. **/etc/xdl/db.conf** を編集して、使用するデータベースを指定します。以下は **db.conf** ファイルの例です。

   ```
   # database configuration file for Linux VDA



   ```

```
## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL

DbType="PostgreSQL"




## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432




## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES
```

```
#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name

# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

PostgreSQL のカスタムバージョンを使用するには、**DbCustomizePostgreSQL** を true に設定します。

3. **ctxsetup.sh** を実行します。

注:

PostgreSQL のポート番号を設定するには、**/etc/xdl/db.conf** を使用することもできます。

ステップ 1i: Motif をインストールする

```
sudo apt-get install -y libxm4
```

ステップ 1j: その他のパッケージをインストールする

Ubuntu 24.04 の場合:

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap2

sudo apt-get install -y krb5-user
```

```
sudo apt-get install -y libgtk2.0-0
```

Ubuntu 22.04 の場合:

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.5-0

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

Ubuntu 20.04 の場合:

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.4-2

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

ステップ 2: ハイパーバイザーを準備する

サポートされているハイパーバイザー上で Linux VDA を VM として実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに基づいて、以下の変更を行ってください。Linux マシンをベアメタルハードウェアで実行している場合、変更は不要です。

XenServer (旧 Citrix Hypervisor™) での時刻同期の修正

XenServer® の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP と XenServer の両方で問題が発生します。両方ともシステムクロックを管理しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVM モードでは変更は不要です。

XenServer VM Tools がインストールされた準仮想化 Linux カーネルを実行している場合、Linux VM 内から XenServer の時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは 0 または 1 を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合は、ファイルに 1 を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を再起動後も永続的に適用するには、**/etc/sysctl.conf file** を編集し、次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



cat /proc/sys/xen/independent_wallclock
```

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux Integration Services がインストールされている Linux VM は、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、NTP サービスと並行してこの機能を有効にしてください。

管理オペレーティングシステムから：

1. Hyper-V マネージャーコンソールを開きます。
2. Linux VM の設定で、**Integration Services** を選択します。
3. **Time synchronization** が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTP との競合を避けるためにホストの時刻同期が無効になっている VMware や XenServer (旧 Citrix Hypervisor) とは異なります。Hyper-V の時刻同期は、NTP の時刻同期と共存し、補完することができます。

ESX および ESXi での時刻同期の修正

VMware Time Synchronization機能が有効になっている場合、各準仮想化Linux VM内で、NTPとハイパーバイザーに関する問題が発生します。どちらもシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホスト時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合:

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **仮想マシンプロパティ**ダイアログで、**オプション**タブを開きます。
4. **VMware Tools**を選択します。
5. **詳細設定**ボックスで、**ゲスト時刻をホストと同期**のチェックを外します。

ステップ3: Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory (AD) ドメインに追加するには、以下の方法があります:

- サンバ・ウィンバインド
- クエスト認証サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ Winbind

必要なパッケージをインストールまたは更新する

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-config krb5-locales krb5-user
```

マシンの起動時にWINBINDデーモンが開始するように有効にする

Winbindデーモンは、マシンの起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind` スクリプトが `/etc/init.d` の下にあることを確認してください。

KERBEROSを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]
```

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM = {`

`admin_server = domain-controller-fqdn`

`kdc = domain-controller-fqdn`

`}`

`[domain_realm]`

`domain-dns-name = REALM`

`.domain-dns-name = REALM`

このコンテキストにおける**domain-dns-name**パラメーターは、**example.com**のようなDNSドメイン名です。**REALM**は、**EXAMPLE.COM**のような大文字のKerberosレルム名です。

WINBIND認証の構成

UbuntuにはRHELの**authconfig**やSUSEのyast2のようなツールがないため、Winbindを手動で構成します。

vim /etc/samba/smb.confコマンドを実行して**/etc/samba/smb.conf**を開き、次の設定を行います。

`[global]`

`workgroup = WORKGROUP`

`security = ADS`

`realm = REALM`

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUPは**REALM**の最初のフィールドであり、**REALM**は大文字のKerberosレルム名です。

NSSWITCHの構成

/etc/nsswitch.confを開き、次の行に**winbind**を追加します。

`passwd: compat winbind`

`group: compat winbind`

WINDOWSドメインへの参加

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
sudo net ads join REALM -U user
```

ここで、**REALM**は大文字のKerberosレルム名であり、**user**はコンピューターをドメインに追加する権限を持つドメインユーザーです。

`WINBIND`の再起動

```
sudo systemctl restart winbind
```

WINBINDのPAMの構成

次のコマンドを実行し、**Winbind NT/Active Directory authentication**と**Create home directory on login**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ヒント:

winbind デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

ドメインメンバーシップの確認

Delivery Controllerは、WindowsかLinuxかにかかわらず、すべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを必要とします。

マシンがドメインに参加していることを確認するには、**Samba** の **net ads** コマンドを実行します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

KERBEROS構成の確認

Linux VDAで使用するためにKerberosが正しく構成されていることを確認するには、システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos **kinit** コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。シェル置換を防ぐために、ドル記号 ($) はバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

マシンのアカウント詳細を確認するには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

ドメインユーザーがドメインで認証できることを確認するには、**wbinfo** ツールを使用します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、以前に使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっており、適切に動作していることを確認してください。

id -u コマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効であり、期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンすることで実行できます。ドメイン参加の検証後、手順6: Linux VDAをインストールするに進みます。

ヒント:

ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップが表示されない場合は、マシンを再起動してからもう一度試してください。

Quest認証サービス

ドメインコントローラーでQUESTを構成する

Active Directoryドメインコントローラーにソフトウェアをインストールして構成済みであり、`Active Directory`でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unix対応**をオンにします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDA で QUEST を構成する

SELinux ポリシーの適用を回避する
デフォルトの RHEL 環境では、SELinux が完全に適用されています。この適用により、Quest が使用する Unix ドメインソケット IPC メカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinux を無効にすることです。root ユーザーとして、**/etc/selinux/config** を編集し、**SELinux** の設定を変更します。

`SELINUX=disabled`

この変更にはマシンの再起動が必要です。

```
reboot
```

> **重要:**

> この設定は慎重に使用してください。無効にした後に SELinux ポリシーの適用を再度有効にすると、root ユーザーや他のローカルユーザーであっても、完全にロックアウトされる可能性があります。

VAS デーモンを構成する
Kerberos チケットの自動更新は有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を 9 時間（32,400 秒）に設定します。これは、デフォルトの 10 時間のチケット有効期間よりも 1 時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより低い値に設定してください。
PAM と NSS を構成する
HDX および su、ssh、RDP などの他のサービスを介したドメインユーザーのログオンを有効にするには、次のコマンドを実行して PAM と NSS を手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

WINDOWS ドメインに参加する

Quest の **vastool** コマンドを使用して、Linux マシンを Active Directory ドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターを Active Directory ドメインに参加させる権限を持つ任意のドメインユーザーです。**domain-name** はドメインの DNS 名であり、たとえば example.com のようになります。

ドメイン参加後、Linuxマシンを再起動します。

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsかLinuxかにかかわらず）がActive Directoryにコンピューターオブジェクトを持つことを必要とします。Questに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます。

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、以前に使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

id -uコマンドによって返されるUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で期限切れではないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/manual-installation-overview/ubuntu.html#step-6-install-the-linux-vda)に進みます。

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

Centrify DirectControlエージェントがインストールされている状態で、Centrifyの**adjoin**コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su –

adjoin -w -V -u user domain-name
```

userパラメーターは、コンピューターを**Active Directory**ドメインに参加させる権限を持つ任意のActive Directoryドメインユーザーです。**domain-name**パラメーターは、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

デリバリーコントローラーは、WindowsまたはLinuxにかかわらず、すべてのVDAマシンが**Active Directory**にコンピューターオブジェクトを持つことを要求します。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
su –



adinfo
```

ドメインに参加済みの値が有効であり、**CentrifyDCモード**が**接続済み**を返すことを確認します。モードが開始状態のまま動かない場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo --diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

SSSD

KERBEROSの構成

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、ルートとして **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。次の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]
```

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストにおける`domain-dns-name`パラメーターは、example.comのようなDNSドメイン名です。*REALM*は、EXAMPLE.COMのような大文字のKerberosレルム名です。

ドメインに参加する

SSSDは、IDプロバイダーとしてActive Directoryを、認証にKerberosを使用するように構成する必要があります。ただし、SSSDはドメイン参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。代わりに、**adcli**、**realmd**、または**Samba**を使用できます。

> **注:**
>
> このセクションでは、**adcli**と**Samba**に関する情報のみを提供します。

- **adcliを使用してドメインに参加する場合は、次の手順を実行します。**

1. **adcli**をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. **adcli**を使用してドメインに参加します。

   古いシステムキータブファイルを削除し、以下を使用してドメインに参加します。

   ```
   su -


   rm -rf /etc/krb5.keytab


   adcli join domain-dns-name -U user -H hostname-fqdn
   ```

   userは、マシンをドメインに追加する権限を持つドメインユーザーです。**hostname-fqdn**は、マシンのFQDN形式のホスト名です。

   -Hオプションは、Linux VDAが必要とするhost/hostname-fqdn@REALM形式でSPNを生成するために**adcli**に必要です。

3. 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マシンでは、`adcli testjoin`コマンドを実行して、マシンがドメインに参加しているかどうかをテストします。

- **Sambaを使用してドメインに参加する場合は、次の手順を実行します。**

1. パッケージをインストールします。

```
sudo apt-get install samba krb5-user
```

2. **Samba** を構成します。

   /etc/samba/smb.conf を開き、次の設定を行います。

   `[global]`

   `workgroup =` *`WORKGROUP`*

   `security = ADS`

   `realm =` *`REALM`*

   `client signing = yes`

   `client use spnego = yes`

   `kerberos method = secrets and keytab`

   WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字の Kerberos レルム名です。

3. **Samba** を使用してドメインに参加します。

   ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つ Windows アカウントが必要です。

```
sudo net ads join REALM -U user
```

   ここで、**REALM** は大文字の Kerberos レルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

SSSD をセットアップする

必要なパッケージをインストールまたは更新します:

必要な SSSD および構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新をお勧めします。

```
sudo apt-get install --only-upgrade sssd
```

> **注:**
>
> デフォルトでは、Ubuntu のインストールプロセスは **nsswitch.conf** と PAM ログインモジュールを自動的に構成します。

SSSD を構成する

SSSD デーモンを開始する前に、SSSD の構成変更が必要です。一部のバージョンの SSSD では、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトでインストールされておらず、手動で作成する必要があります。root として、**/etc/sssd/sssd.conf** を作成または開き、次の設定を行います。

`[sssd]`

`services = nss, pam`

`config_file_version = 2`

`domains =` *`domain-dns-name`*

`[domain/` *`domain-dns-name`* `]`

`id_provider = ad`

`access_provider = ad`

`auth_provider = krb5`

`krb5_realm =` *`REALM`*

`# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14 days`

```
krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

> **注:**
>
> ldap_id_mapping は **true** に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングする処理を行います。そうでない場合、Active Directory は POSIX 拡張機能を提供できる必要があります。PAM サービス `ctxhdx` は ad_gpo_map_remote_interactive に追加されます。

> このコンテキストでの **domain-dns-name** パラメーターは、example.com のような DNS ドメイン名です。**REALM** は、EXAMPLE.COM のような大文字の Kerberos レルム名です。NetBIOS ドメイン名を設定する必要はありません。

> 構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

> SSSD デーモンは、構成ファイルが所有者による読み取り専用のアクセス許可を持っている必要があります。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSD デーモンの開始

SSSD デーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM の構成

次のコマンドを実行し、**SSS 認証**と**ログイン時にホームディレクトリを作成**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が `Active Directory` にコンピューターオブジェクトを持っていることを要求します。

- ドメインメンバーシップを確認するために **adcli** を使用する場合は、`sudo adcli info domain-dns-name` コマンドを実行してドメイン情報を表示します。
- ドメインメンバーシップを確認するために **Samba** を使用する場合は、`sudo net ads testjoin` コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info` コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

KERBEROS構成の確認

Linux VDAでKerberosが正しく構成されていることを確認するには、システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos `kinit` コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。シェル置換を防ぐために、ドル記号 ($) はバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

SSSD PAMモジュールが正しく構成されていることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返されるKerberosチケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

ルートユーザーとして、以前の **id -u** コマンドによって返されたuidに対応するチケットキャッシュファイルが作成されていることを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDEまたはGnome Display Managerにログオンすることで実行できます。ドメイン参加の検証後、手順6：Linux VDAのインストール に進みます。

PBIS

必要なPBISパッケージをダウンロードする

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
9.1.0/pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行可能にする

```
sudo chmod +x pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

PBISインストールスクリプトを実行する

```
sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

WINDOWSドメインへの参加

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
sudo /opt/pbis/bin/domainjoin-cli join domain-name user
```

user は、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

注: Bashをデフォルトシェルとして設定するには、**sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** コマンドを実行します。

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（Windows VDAおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持っている必要があります。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには、次の手順を実行します。

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順6: Linux VDAのインストールに進みます。

手順4: .NETのインストール

.NET Runtimeに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションに.ASP.NET Core Runtimeをインストールする必要があります。Amazon Linux 2にはバージョン6が必要です。その他のディストリビューションにはバージョン8が必要です。

お使いのLinuxディストリビューションに必要な.NETバージョンが含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。詳しくは、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

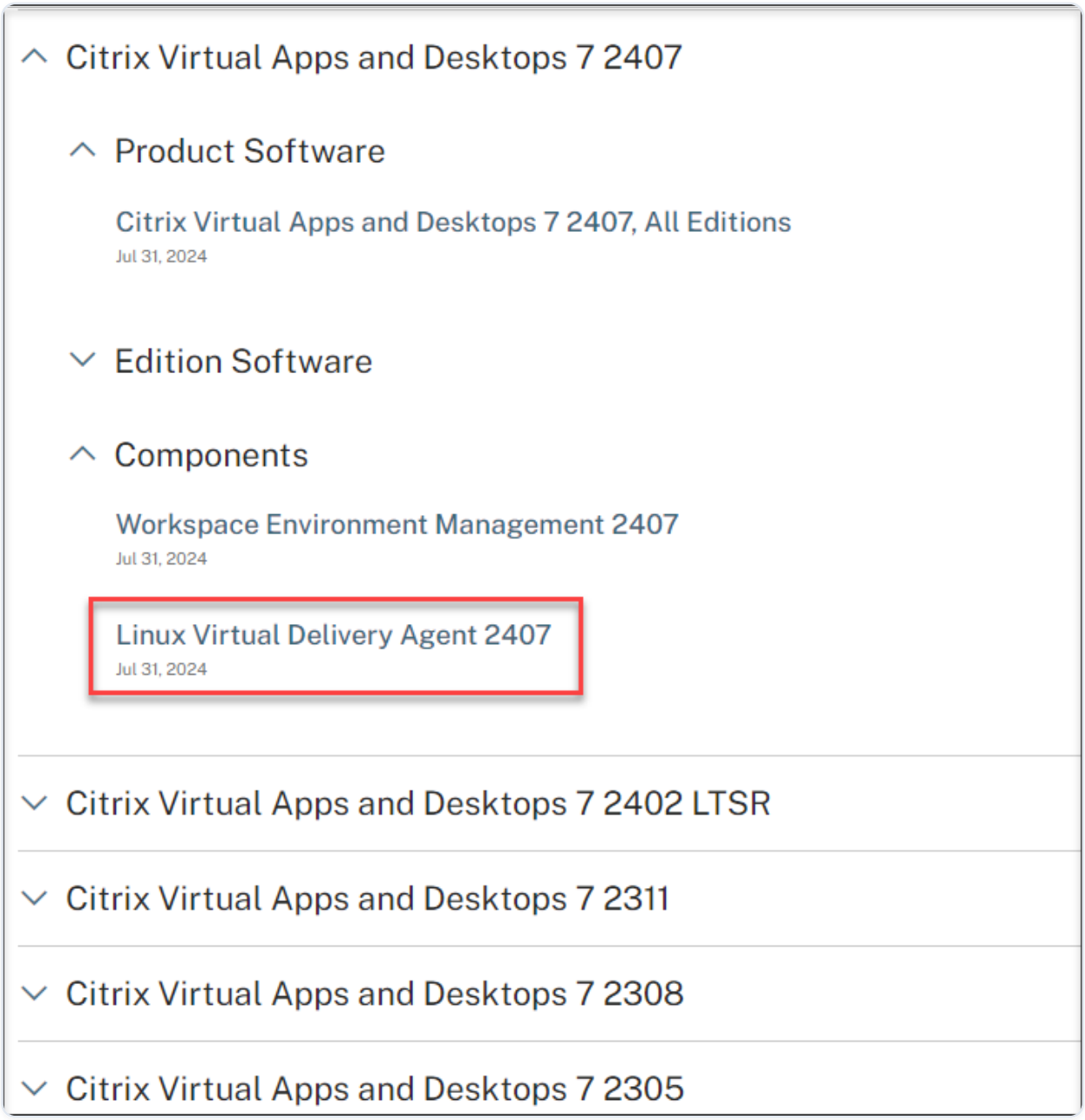
.NET のインストール後、**which dotnet** コマンドを実行してランタイムパスを見つけます。

コマンドの出力に基づいて、.NET ランタイムバイナリパスを設定します。たとえば、コマンドの出力が /aa/bb/dotnet の場合、/aa/bb を .NET バイナリパスとして使用します。

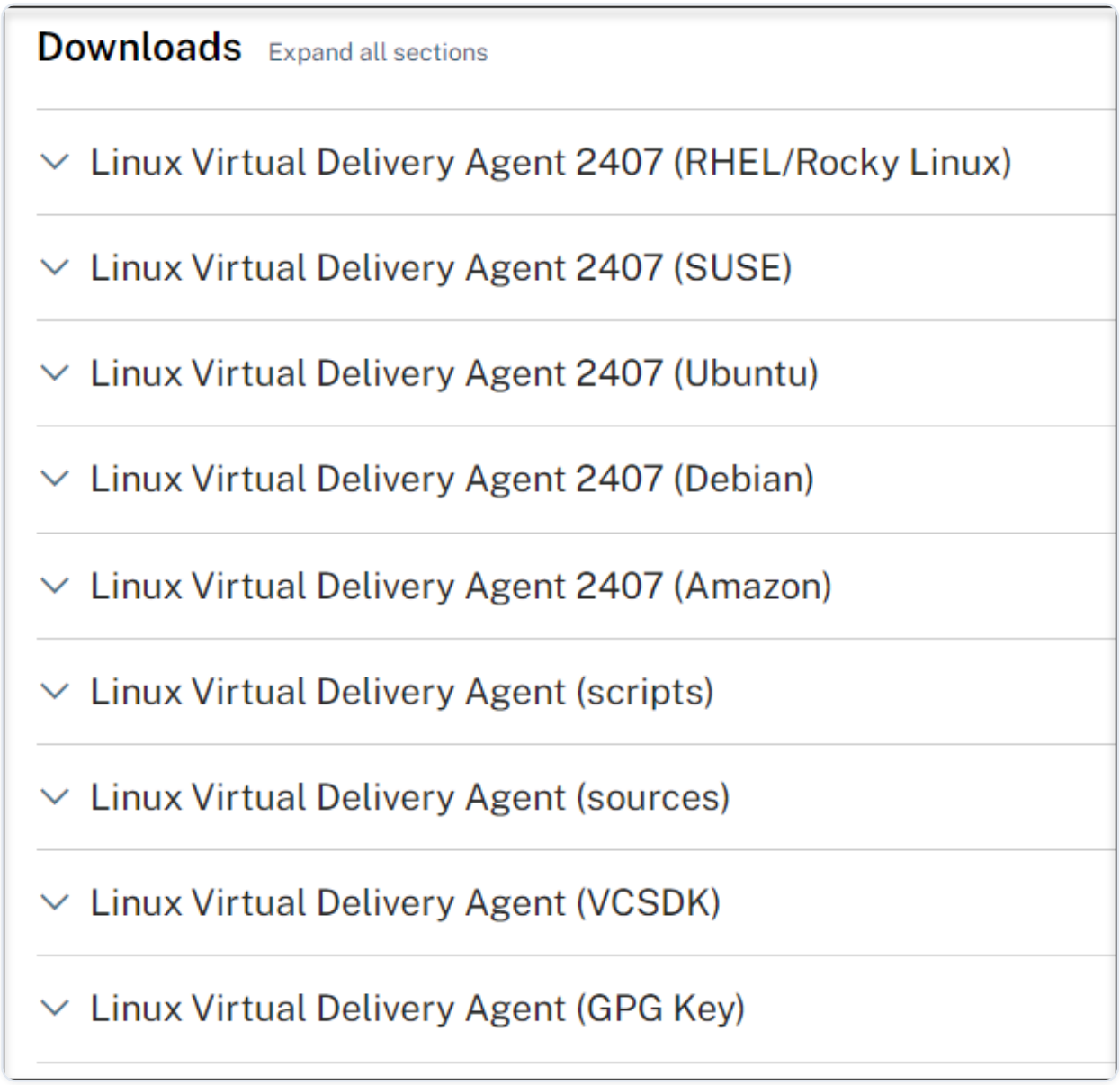
ステップ 5: Linux VDA パッケージをダウンロードする

1. Citrix Virtual Apps and Desktops ダウンロードページにアクセスします。

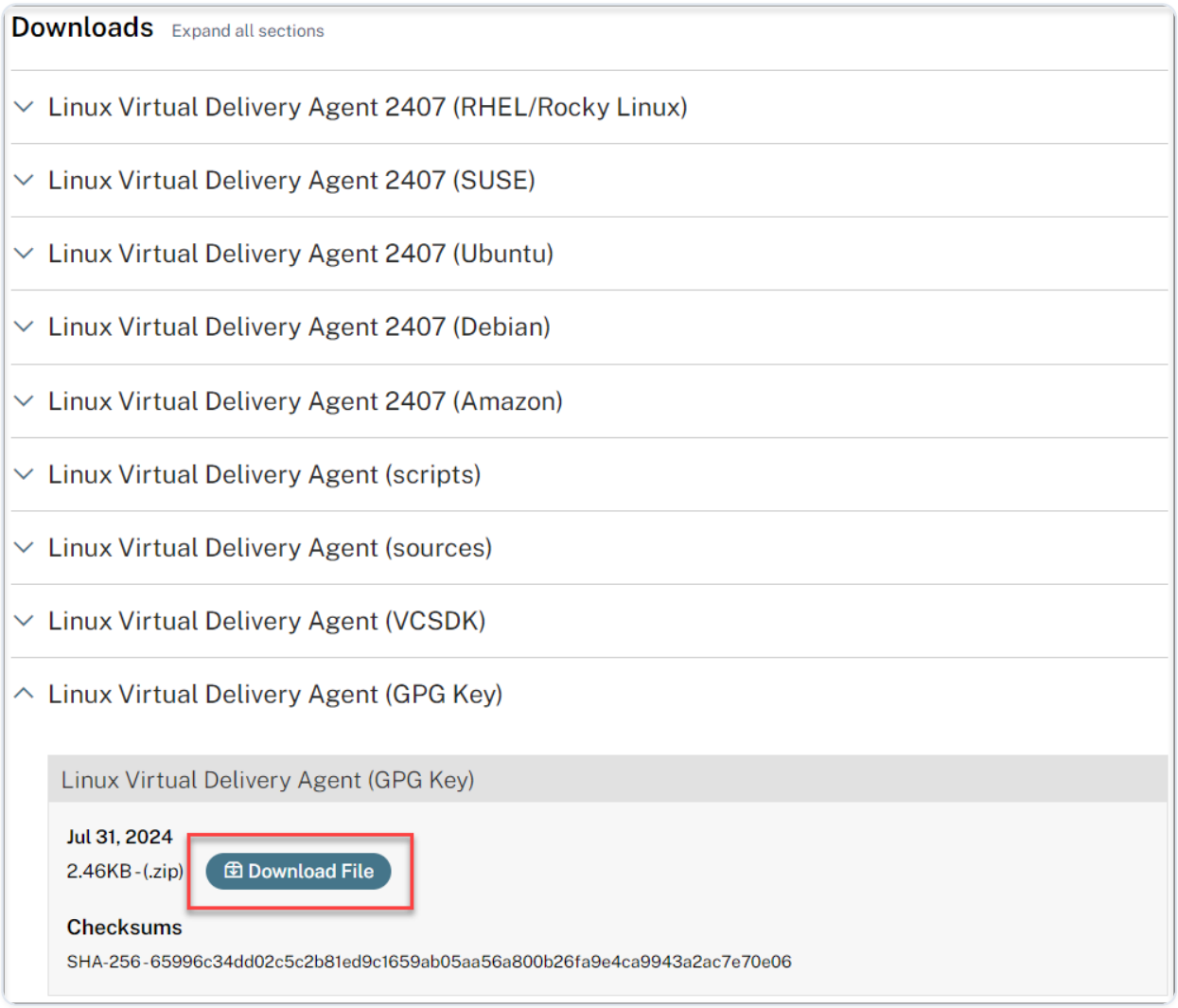
2. Citrix Virtual Apps and Desktops の適切なバージョンを展開します。
3. **Components** を展開して Linux VDA を見つけます。例:



4. Linux VDA リンクをクリックして、Linux VDA のダウンロードにアクセスします。



5. お使いの Linux ディストリビューションに一致する Linux VDA パッケージをダウンロードします。
6. Linux VDA パッケージの整合性を検証するために使用できる GPG 公開鍵をダウンロードします。例:



Linux VDA パッケージの整合性を検証するには、次のコマンドを実行して公開鍵を DEB データベースにインポートし、パッケージの整合性を確認します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

ステップ 6: Linux VDA をインストールする

ステップ 6a: Linux VDA をインストールする

Debian パッケージマネージャーを使用して Linux VDA ソフトウェアをインストールします。

```
sudo dpkg –i <PATH>/<Linux VDA DEB>

apt-get install -f
```

注:

GCP 上の Ubuntu 24.04、Ubuntu 22.04、および 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 の下に **rdns = false** の行を追加します。

Ubuntu 24.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.12

libgtkmm-3.0-1t64 >= 3.24.9

ufw >= 0.36

ubuntu-desktop >= 1.539

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.39

gtk3-nocsd >= 3

bash >= 5.2

findutils >= 4.9.0

sed >= 4.9

cups >= 2.4

libmspack0t64 >= 0.11

curl >= 8.5.0

libbsd-dev >= 0.12.1-1build1
```

```
ibus >= 1.5

libqt5dbus5 >= 5.15~

libgoogle-perftools4 >= 2.15

libpython3.12 >= 3.12~

libsasl2-modules-gssapi-mit >= 2.1.~

libnss3-tools >= 2:3.98

libqt5widgets5 >= 5.15~

libqrencode4 >= 4.1.1

libimlib2 >= 1.12.1

libfuse2 >= 2.9

mutter >= 46.0

pulseaudio-utils >= 16.1
```

Ubuntu 22.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.11

libgtkmm-3.0-1v5 >= 3.24.5

ufw >= 0.36

ubuntu-desktop >= 1.481

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8
```

```
util-linux >= 2.37

gtk3-nocsd >= 3

bash >= 5.1

findutils >= 4.8.0

sed >= 4.8

cups >= 2.4

libmspack0 >= 0.10

ibus >= 1.5

libqt5dbus5 >= 5.15~

libgoogle-perftools4    2.9~

libpython3.10 >= 3.10~

libsasl2-modules-gssapi-mit >= 2.1.~

libnss3-tools >= 2:3.68

libqt5widgets5 >= 5.15~

libqrencode4 >= 4.1.1

libimlib2 >= 1.7.4

libfuse2 >= 2.9

mutter >= 42.5
```

Ubuntu 20.04 の Debian 依存関係リスト:

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.10
```

```
libgtkmm-3.0-1v5 >= 3.24.2

ufw >= 0.36

ubuntu-desktop >= 1.450

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.34

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.7.0

sed >= 4.7

cups >= 2.3

libmspack0 >= 0.10

ibus >= 1.5

libqt5dbus5 >= 5.7~

libgoogle-perftools4 >= 2.7~

libpython3.8 >= 3.8~

libsasl2-modules-gssapi-mit >= 2.1.~

libnss3-tools >= 2:3.49

libqt5widgets5 >= 5.7~

libqrencode4 >= 4.0.0

libimlib2 >= 1.6.1
```

```
libfuse2 >= 2.9
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

ステップ 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、1912 LTSR から 2203 LTSR にアップグレードできます。

```
sudo dpkg -i <PATH>/<Linux VDA deb>



sudo apt-get install -f
```

注:

- 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、ファイルをバックアップしてください。
- 2407 リリース以降、Linux VDA は、アップグレード中に構成ファイルを処理するためにパッケージマネージャーの **rpm** または **dpkg** に委任します。以下に、rpm と dpkg が構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを .rpmnew 拡張子で保存します。
  - **dpkg**: 続行方法の選択肢を対話形式で求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを **.dpkg-new** または **.dpkg-dist** として保存しながら、Linux VDA をサイレントアップグレードするには、次のコマンドを使用します。

```
dpkg --force-confold -i package.deb  # Always keep your version,
then save new package's version as *.dpkg-new or *.dpkg-dist
```

ステップ 7: NVIDIA GRID ドライバーのインストール

HDX 3D Pro を有効にするには、ハイパーバイザーと VDA マシンに NVIDIA GRID ドライバーをインストールする必要があります。

特定のハイパーバイザーに NVIDIA GRID Virtual GPU Manager (ホストドライバー) をインストールおよび構成するには、次のガイドを参照してください。

- XenServer
- ヴイエムウェア イーエスエックス
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、次の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバー（クラウドベンダーまたはNVIDIAから）をインストールします。

ステップ8：Linux VDAを構成する

注:

ランタイム環境をセットアップする前に、OSに**en_US.UTF-8**ロケールがインストールされていることを確認してください。ロケールがOSで利用できない場合は、**sudo locale-gen en_US.UTF-8**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集し、**# en_US.UTF-8 UTF-8**行のコメントを解除してから、**sudo locale-gen**コマンドを実行します。

パッケージをインストールした後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がイン

> ストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトに従って手動で構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションは環境変数で提供できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を求めず、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンをドメインに 参加させるかどうか。デフォルト値は「n」です。ドメイン参加シナリオの場合、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – Linux VDAは、Delivery Controllerとの認証のためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDAは、Delivery Controllerに登録するために、スペースで区切られたDelivery Controllerの完全修飾ドメイン名（FQDN）のリストを必要とします。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、値を**「y」**に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーション技術のセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=**「y」**）用に構成されます。

- **CTX_XDL_START_SERVICE='y\|n'** – 構成が完了したときにLinux VDAサービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – Linux Virtual Desktopサービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDAサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktopの場合、必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/'<none\>'** – セッションで使用するGNOME、GNOME Classic、KDE、MATE、または**Xfce**デスクトップ環境を指定します。**'<none\>'**に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳細については、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（ctxvda）をサポートするために.NETをインストールするパス。デフォルトのパスは**「/usr/bin」**です。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。
- **CTX_XDL_SITE_NAME=<dns-name\>** – Linux VDAは、DNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。不要な場合は、**'<none\>'**に設定します。
- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDAは、LDAPサーバーを検出するためにDNSを照会します。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを含むスペース区切りのLDAP FQDNのリストを提供できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数はデフォルトで**'<none\>'**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを介してLDAPを照会します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、**'<none\>'**に設定します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|
xfce|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'

export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'

export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'

export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

sudo コマンドを実行するときは、既存の環境変数を新しく作成されるシェルに渡すために、**-E** オプションを入力します。前述のコマンドから、最初の行に **#\!/bin/bash** を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメータを指定することもできます。

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \

CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \

CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

/opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

構成の変更を削除する

シナリオによっては、Linux VDA パッケージをアンインストールせずに、**ctxsetup.sh** スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDA を動作不能にします。

構成ログ

ctxsetup.sh および **ctxcleanup.sh** スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル **/tmp/xdl.configure.log** に書き込まれます。

変更を有効にするには、Linux VDA サービスを再起動します。

Linux VDA ソフトウェアをアンインストールする

Linux VDA がインストールされているかどうかを確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

注:

Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前にセットアップされたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

ヒント:

> このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

ステップ9：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、**XDPing**を参照してください。

ステップ10：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成したら、以下のコマンドを使用してLinux VDAを制御します。

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

> **注:**
>
> **ctxvda**サービスと**ctxhdx**サービスを停止する前に、**systemctl stop ctxmonitord**コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの稼働状況を確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ11：マシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと同様です。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します。
  - ホスト型共有デスクトップ配信モデルの場合は、**Multi-session OS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**Single-session OS**オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

> Citrix Studioの初期バージョンでは、「Linux OS」という概念をサポートしていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト型共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに単一ユーザーの配信モデルが意味されます。

> **ヒント:**

> マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除し、再度追加する必要があります。

ステップ12：デリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンは許可しないでください。
- デリバリーグループと、Windowsマシンを含むマシンカタログを混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2411」を参照してください。

DebianにLinux VDAを手動でインストールする

> **重要:**

> 新規インストールの場合、迅速なインストールにはeasy installの使用をお勧めします。Easy install は時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ1：構成情報とLinuxマシンの準備

ステップ1a：ホスト名の設定

マシンのホスト名が正しく報告されるように、**/etc/hostname**ファイルを変更して、マシンのホスト名のみが含まれるようにします。

`hostname`

ステップ1b：ホスト名にループバックアドレスを割り当てる

マシンのDNSドメイン名と完全修飾ドメイン名（FQDN）が正しく報告されることを確認してください。そのためには、**/etc/hosts**ファイルの次の行を変更して、FQDNとホスト名を最初の2つのエントリとして含めます。

`127.0.0.1 hostname-fqdn hostname localhost`

例:

`127.0.0.1 vda01.example.com vda01 localhost`

ファイル内の他のエントリから、`hostname-fqdn`または`hostname`への他の参照をすべて削除します。

> **注:**
>
> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。ホスト名は15文字を超えてはなりません。

> **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン（-）文字のみを使用してください。アンダーバー（_）、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールは、Delivery Controllerのホスト名にも適用されます。

ステップ1c：ホスト名の確認

マシンを再起動し、ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ 1d: マルチキャスト DNS を無効にする

デフォルト設定ではマルチキャスト DNS (**mDNS**) が有効になっており、名前解決の結果が一貫しなくなる可能性があります。

mDNS を無効にするには、**/etc/nsswitch.conf** を編集して、次の行を変更します。

`hosts: files mdns_minimal [NOTFOUND=return] dns`

変更後:

`hosts: files dns`

ステップ 1e: 名前解決とサービス到達可能性を確認する

FQDN を解決し、ドメインコントローラーと Delivery Controller™ に ping を実行できることを確認します。

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn



nslookup delivery-controller-fqdn



ping delivery-controller-fqdn
```

FQDN を解決できない場合、またはこれらのマシンのいずれかに ping を実行できない場合は、続行する前に手順を確認してください。

ステップ 1f: クロック同期を構成する (chrony)

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することは非常に重要です。Linux VDA を仮想マシン (VM) としてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモート時刻サービスとの時刻同期が推奨されます。

chrony をインストールします。

```
apt-get install chrony
```

ルートユーザーとして、**/etc/chrony/chrony.conf** を編集し、各リモートタイムサーバーのサーバーエントリを追加します。

`server peer1-fqdn-or-ip-address iburst`

`server peer2-fqdn-or-ip-address iburst`

通常の展開では、パブリック NTP プールサーバーから直接ではなく、ローカルのドメインコントローラーから時刻を同期します。ドメイン内の各 Active Directory ドメインコントローラーのサーバーエントリを追加します。

ループバック IP アドレス、localhost、およびパブリックサーバーの ***.pool.ntp.org** エントリを含む、リストされている他のすべての **server** または **pool** エントリを削除します。

変更を保存し、Chrony デーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ 1g: パッケージをインストールする

```
sudo apt-get install -y libsasl2-2



sudo apt-get install -y libgtk2.0-0
```

ステップ 1h: 必要な依存関係をインストールするためのリポジトリを追加する

Debian 11 の場合、`deb http://deb.debian.org/debian/ bullseye main` 行を /etc/apt/sources.list ファイルに追加します。

ステップ 1i: 使用するデータベースをインストールして指定する

注:

- VDI モードでのみ SQLite を使用し、ホストされた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。
- 簡易インストールおよび MCS の場合、SQLite または PostgreSQL を手動でインストールすることなく使用するように指定できます。**/etc/xdl/db.conf** で特に指定しない限り、Linux VDA はデフォルトで PostgreSQL を使用します。Linux ディストリビューションによって提供されるバージョンではなく、カスタムバージョンの PostgreSQL が必要な場合は、指定されたバージョンを手動でインストールし、新しいバージョンを反映するように `/etc/xdl/db.conf` を編集し、簡易インストールスクリプト (`ctxinstall.sh`) または MCS スクリプト (`deploymcs.sh`) を実行する前に PostgreSQL サービスを開始する必要があります。
- 手動インストールの場合、SQLite、PostgreSQL、またはその両方を手動でインストールする必要があります。Linux ディストリビューションによって提供されるバージョンではなく、カスタムバージョンの PostgreSQL を使用できます。SQLite と PostgreSQL の両方をインストールする場合は、Linux VDA パッケージのインストール後に **/etc/xdl/db.conf** を編集して、どちらか一方を使用するように指定できます。

POSTGRESQL をインストールする

このセクションでは、Linux ディストリビューションによって提供されるバージョンの PostgreSQL をインストールする方法について説明します。カスタムバージョンの PostgreSQL が必要な場合は、特定の要件に基づいてインストールできます。

PostgreSQL をインストールするには、次のコマンドを実行します。

```
sudo apt-get update



sudo apt-get install -y postgresql
```

```
sudo apt-get install -y libpostgresql-jdbc-java
```

マシンの起動時またはすぐに PostgreSQL を開始するには、次のコマンドをそれぞれ実行します。

```
sudo systemctl enable postgresql


sudo systemctl start postgresql
```

SQLITE をインストールする

Debian の場合、SQLite をインストールするには、次のコマンドを実行します。

```
sudo apt-get install -y sqlite3
```

使用するデータベースを指定する

SQLite と PostgreSQL の両方をインストールする場合は、Linux VDA パッケージのインストール後に **/etc/xdl/db.conf** を編集して、どちらか一方を使用するように指定できます。

1. **/opt/Citrix/VDA/sbin/ctxcleanup.sh** を実行します。新規インストールの場合、この手順は省略します。
2. 使用するデータベースを指定するために **/etc/xdl/db.conf** を編集します。以下は **db.conf** ファイルの例です。

```
# database configuration file for Linux VDA



## database choice

# possible choices are:

#     SQLite

#     PostgreSQL

# default choice is PostgreSQL
```

```
DbType="PostgreSQL"



## database port

# specify database port for the database.

# if not specified, default port will be used:

# SQLite: N/A

# PostgreSQL: 5432

DbPort=5432



## PostgreSQL customized

# only the following value means true, otherwise false:

#     true

#     yes

#     y

#     YES

#     Y

# default is false

DbCustomizePostgreSQL=false



## PostgreSQL service name
```

```
# specify the service name of PostgreSQL for Linux VDA

# default is "postgresql"

DbPostgreSQLServiceName="postgresql"
```

カスタムバージョンのPostgreSQLを使用するには、**DbCustomizePostgreSQL** を true に設定します。

3. **ctxsetup.sh** を実行します。

注:

PostgreSQLのポート番号を設定するために、**/etc/xdl/db.conf** を使用することもできます。

ステップ2：ハイパーバイザーの準備

サポートされているハイパーバイザー上でLinux VDAをVMとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに基づいて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

XenServer (旧称 Citrix Hypervisor™) での時刻同期の修正

XenServer®の時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとXenServerの両方がシステムクロックを管理しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

XenServer VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からXenServerの時刻同期機能が存在し、有効になっているかを確認できます。

```
su -

cat /proc/sys/xen/independent_wallclock
```

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能は無効になっており、それ以上の操作は不要です。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は不要です。

有効になっている場合、ファイルに1を書き込むことで時刻同期機能を無効にします。

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
```

この変更を永続化し、再起動後も維持するには、**/etc/sysctl.conf ファイル** を編集して次の行を追加します。

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -

cat /proc/sys/xen/independent_wallclock
```

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux統合サービスがインストールされているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、NTPサービスと並行してこの機能を有効にしてください。

管理オペレーティングシステムから:

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTPとの競合を避けるためにホストの時刻同期が無効になっているVMwareやXenServer（旧Citrix Hypervisor）とは異なります。Hyper-Vの時刻同期は、NTPの時刻同期と共存し、補完することができます。

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとハイパーバイザーとの間で問題が発生します。両方ともシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるために、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合:

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。
4. 「**VMware Tools**」を選択します。
5. 「**Advanced**」ボックスで、「**Synchronize guest time with host**」のチェックを外します。

ステップ 3: Linux VM を Windows ドメインに追加する

Linux マシンを Active Directory (AD) ドメインに追加するには、次の方法が利用可能です。

- サンバ・ワインバインド
- クエスト認証サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて手順に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名が使用されている場合、セッションの起動が失敗する可能性があります。

サンバ・ワインバインド

必要なパッケージをインストールまたは更新する

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-config krb5-locales krb5-user
```

WINBIND デーモンがマシンの起動時に開始するように有効にする

Winbind デーモンは、マシンの起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind` スクリプトが `/etc/init.d` の下にあることを確認してください。

KERBEROSの構成

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。次の設定は、単一ドメイン、単一フォレストモデルを対象としています。

`[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

WINBIND認証の構成

vim /etc/samba/smb.conf コマンドを実行して **/etc/samba/smb.conf** を開き、次の設定を行います。

`[global]`

`workgroup =` *`WORKGROUP`*

`security = ADS`

`realm =` *`REALM`*

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字の Kerberos レルム名です。

NSSWITCH の設定

/etc/nsswitch.conf を開き、次の行に `winbind` を追加します。

`passwd: files systemd winbind`

`group: files systemd winbind`

WINDOWS ドメインへの参加

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

```
sudo net ads join <Kerberos realm name in uppercase> -U <domain user with
permission to add computers to the domain>
```

WINBIND の再起動

```
sudo systemctl restart winbind
```

WINBIND 用の PAM の設定

次のコマンドを実行し、**Winbind NT/Active Directory 認証**と**ログイン時にホームディレクトリを作成**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

> **ヒント:**
>
> **winbind** デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

ドメインメンバーシップの確認

Delivery Controller は、Windows または Linux のすべての VDA マシンが `Active Directory` にコンピューターオブジェクトを持つことを要求します。

マシンがドメインに参加していることを確認するために、**Samba** の **net ads** コマンドを実行します。

```
sudo net ads testjoin
```

追加のドメインおよびコンピューターオブジェクト情報を確認するには、次のコマンドを実行します。

```
sudo net ads info
```

KERBEROS 設定の確認

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用して、Kerberosの**kinit**コマンドを実行し、ドメインコントローラーでマシンを認証します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist
```

以下を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

wbinfoツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインはADドメイン名であり、Kerberosレルム名ではありません。bashシェルでは、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAMモジュールが正しく構成されていることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

> **注:**
>
> SSHコマンドを正常に実行するには、SSHが有効になっており、正しく機能していることを確認してください。

> **id -u**コマンドによって返されたuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンして実行できます。ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

> **ヒント:**
>
> ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップが表示されない場合は、マシンを再起動してもう一度試してください。

Quest認証サービス

ドメインコントローラーでQUESTを構成する

Active Directoryドメインコントローラーにクエストソフトウェアをインストールして構成済みであり、`Active Directory`でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには：

1. Active Directoryユーザーとコンピューター管理コンソールで、そのユーザーアカウントのActive Directoryユーザープロパティを開きます。
2. **Unixアカウント**タブを選択します。
3. **Unix対応**をオンにします。
4. **プライマリGID番号**を実際のドメインユーザーグループのグループIDに設定します。

> **注:**
>
> これらの手順は、コンソール、RDP、SSH、またはその他のリモートプロトコルを使用してログオンするドメインユーザーを設定する場合にも同様に適用されます。

LINUX VDAでQUESTを構成する

SELinuxポリシーの適用を回避する
デフォルトのRHEL環境ではSELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

この問題を回避する便利な方法は、SELinuxを無効にすることです。rootユーザーとして、***/etc/selinux/config***を編集し、**SELinux**設定を変更します。

```
SELINUX=disabled
```

この変更にはマシンの再起動が必要です。

```
reboot
```

> **重要:**
>
> この設定は慎重に使用してください。無効にした後にSELinuxポリシーの適用を再度有効にすると、rootユーザーや他のローカルユーザーであっても完全にロックアウトされる可能性があります。

VAS デーモンを構成する
Kerberos チケットの自動更新を有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

```
sudo /opt/quest/bin/vastool configure vas vasd auto-ticket-renew-interval
32400



sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-auth
false
```

このコマンドは、更新間隔を9時間（32,400秒）に設定します。これは、デフォルトの10時間のチケット有効期間より1時間短いです。チケット有効期間が短いシステムでは、このパラメーターをより小さい値に設定してください。

PAM と NSS を構成する

HDX および su、ssh、RDP などの他のサービスを介したドメインユーザーログオンを有効にするには、次のコマンドを実行して PAM と NSS を手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam



sudo /opt/quest/bin/vastool configure nss
```

WINDOWS ドメインに参加する

Quest の **vastool** コマンドを使用して、Linux マシンを Active Directory ドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターを Active Directory ドメインに参加させる権限を持つ任意のドメインユーザーです。**domain-name** はドメインの DNS 名です（例: example.com）。

ドメイン参加後、Linux マシンを再起動します。

ドメインメンバーシップを確認する

Delivery Controller は、Windows または Linux のすべての VDA マシンが Active Directory にコンピューターオブジェクトを持つことを要求します。Quest に参加している Linux マシンがドメイン上にあることを確認するには、次のようにします。

```
sudo /opt/quest/bin/vastool info domain
```

マシンがドメインに参加している場合、このコマンドはドメイン名を返します。マシンがどのドメインにも参加していない場合、次のエラーが表示されます。

`ERROR: No domain could be found.`

`ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm` `default_realm not configured in vas.conf. Computer may not be joined to domain`

ユーザー認証を確認する

Quest が PAM を介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username



id -u
```

id -u コマンドによって返された UID に対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

Kerberos資格情報キャッシュ内のチケットが有効で期限切れでないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順6: Linux VDAのインストールに進みます。

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

Centrify DirectControl Agentがインストールされている状態で、Centrify `adjoin`コマンドを使用してLinuxマシンをActive Directoryドメインに参加させます。

```
su –

adjoin -w -V -u user domain-name
```

userパラメーターは、コンピューターをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name**パラメーターは、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxを問わず、すべてのVDAマシンがActive Directory内にコンピューターオブジェクトを持つことを要求します。Centrifyで参加したLinuxマシンがドメイン上にあることを確認するには：

```
su –


adinfo
```

Joined to domainの値が有効であり、**CentrifyDC mode**が**connected**を返すことを確認します。モードが開始状態のまま動かない場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all


adinfo --diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、手順6: Linux VDAのインストールに進みます。

SSSD

KERBEROSの構成

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを設定するには、rootとして **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを設定します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

> `[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`rdns = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

このコンテキストでの `domain-dns-name` パラメーターは、example.comのようなDNSドメイン名です。*REALM* は、EXAMPLE.COMのような大文字のKerberosレルム名です。

ドメインへの参加

SSSDは、Active DirectoryをIDプロバイダーとして、Kerberosを認証に使用するように構成する必要があります。ただし、SSSDはドメインへの参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。代わりに、**adcli**、**realmd**、または**Samba**を使用できます。

> **注:**
>
> このセクションでは、**adcli**と**Samba**に関する情報のみを提供します。

- **adcliを使用してドメインに参加する場合は、次の手順を完了します。**

1. **adcli**をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. **adcli** を使用してドメインに参加します。

   古いシステムキーファイル (keytab file) を削除し、以下を使用してドメインに参加します。

   ```
   su -



   rm -rf /etc/krb5.keytab



   adcli join domain-dns-name -U user -H hostname-fqdn
   ```

   user は、マシンをドメインに追加する権限を持つドメインユーザーです。**hostname-fqdn** は、マシンの FQDN 形式のホスト名です。

   -H オプションは、Linux VDA が必要とする host/hostname-fqdn@REALM 形式の SPN を **adcli** が生成するために必要です。

3. システムキーファイル (keytab) を確認します。

   システムキーファイルが作成されていることを確認するために、`sudo klist -ket` コマンドを実行します。

各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **Samba を使用してドメインに参加する場合は、次の手順を実行します。**

1. パッケージをインストールします。

```
sudo apt-get install samba krb5-user
```

2. **Samba** を構成します。

   /etc/samba/smb.conf を開き、次の設定を行います。

   `[global]`

   `workgroup =` *`WORKGROUP`*

   `security = ADS`

   `realm =` *`REALM`*

   `client signing = yes`

   `client use spnego = yes`

   `kerberos method = secrets and keytab`

   WORKGROUP は *REALM* の最初のフィールドであり、*REALM* は大文字の Kerberos レルム名です。

3. **Samba** を使用してドメインに参加します。

   ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つWindowsアカウントが必要です。

```
sudo net ads join <the Kerberos realm name in uppercase> -U <domain
user with permission to add computers to the domain>
```

SSSD のセットアップ

必要なパッケージのインストールまたは更新:

必要なSSSDおよび構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新が推奨されます。

```
sudo apt-get install --only-upgrade sssd
```

SSSD の構成

SSSD デーモンを開始する前に、SSSD 構成の変更が必要です。SSSD の一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトでインストールされていないため、手動で作成する必要があります。root として、**/etc/sssd/sssd.conf** を作成または開き、次の設定を行います。

`[sssd]`

`services = nss, pam`

`config_file_version = 2`

`domains =` *`domain-dns-name`*

`[domain/` *`domain-dns-name`* `]`

`id_provider = ad`

`access_provider = ad`

`auth_provider = krb5`

`krb5_realm =` *`REALM`*

`# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14 days`

`krb5_renewable_lifetime = 14d`

`# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than 2 hours`

`krb5_renew_interval = 1h`

`krb5_ccachedir = /tmp`

`krb5_ccname_template = FILE:%d/krb5cc_%U`

`# This ldap_id_mapping setting is also the default value`

`ldap_id_mapping = true`

`override_homedir = /home/%d/%u`

`default_shell = /bin/bash`

`ad_gpo_map_remote_interactive = +ctxhdx`

> **注:**
>
> ldap_id_mapping は **true** に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングします。そうでない場合、`Active Directory` は POSIX 拡張機能を提供できる必要があります。PAM サービス `ctxhdx` は ad_gpo_map_remote_interactive に追加されます。

> このコンテキストでの **domain-dns-name** パラメーターは、example.com のような DNS ドメイン名です。*REALM* は、EXAMPLE.COM のような大文字の Kerberos レルム名です。NetBIOS ドメイン名を構成する必要はありません。

> 構成設定の詳細については、sssd.conf および `sssd-ad` の man ページを参照してください。

> SSSD デーモンは、構成ファイルが所有者読み取り権限のみを持つことを要求します。

```
sudo chmod 0600 /etc/sssd/sssd.conf
```

SSSD デーモンの開始

SSSDデーモンを今すぐ起動し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd
```

```
sudo systemctl enable sssd
```

PAM構成

次のコマンドを実行し、**SSS認証**と**ログイン時にホームディレクトリを作成**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ドメインメンバーシップの確認

Delivery Controllerは、すべてのVDAマシン（WindowsおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持っていることを要求します。

- ドメインメンバーシップを確認するために**adcli**を使用する場合は、`sudo adcli info domain-dns-name`コマンドを実行してドメイン情報を表示します。
- ドメインメンバーシップを確認するために**Samba**を使用する場合は、`sudo net ads testjoin`コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info`コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

KERBEROS構成の確認

KerberosがLinux VDAで使用するために正しく構成されていることを確認するには、システムキータブファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos `kinit`コマンドを実行します。

```
sudo kinit -k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェル置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTがキャッシュされていることを確認するには、次を使用します。

```
sudo klist
```

ユーザー認証の確認

SSSDは、デーモンと直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

SSSD PAMモジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返された Kerberos チケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

root ユーザーとして、以前の **id -u** コマンドによって返された uid に対応するチケットキャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDE または Gnome Display Manager にログオンすることで実行できます。ドメイン参加の検証後、手順 6: Linux VDA をインストールするに進みます。

PBIS

必要な PBIS パッケージをダウンロードする

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download/
9.1.0/pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

PBIS インストールスクリプトを実行可能にする

```
sudo chmod +x pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

PBIS インストールスクリプトを実行する

```
sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

WINDOWS ドメインに参加する

ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

```
sudo /opt/pbis/bin/domainjoin-cli join domain-name user
```

user は、コンピューターを Active Directory ドメインに追加する権限を持つドメインユーザーです。**domain-name** は、ドメインの DNS 名です（例: example.com）。

注: Bash をデフォルトシェルとして設定するには、**sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** コマンドを実行します。

ドメインメンバーシップを確認する

Delivery Controller は、すべての VDA マシン（Windows および Linux VDA）が `Active Directory` にコンピューターオブジェクトを持つことを要求します。PBIS に参加している Linux マシンがドメイン上にあることを確認するには:

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加している AD ドメインと OU に関する情報を返します。そうでない場合は、ホスト名のみが表示されます。

ユーザー認証を確認する

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview/manual-installation-overview/debian.html#step-6-install-the-linux-vda)に進みます。

ステップ4：.NETのインストール

.NET Runtimeに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションに.ASP.NET Core Runtimeをインストールする必要があります。Amazon Linux 2にはバージョン6が必要です。その他のディストリビューションにはバージョン8が必要です。

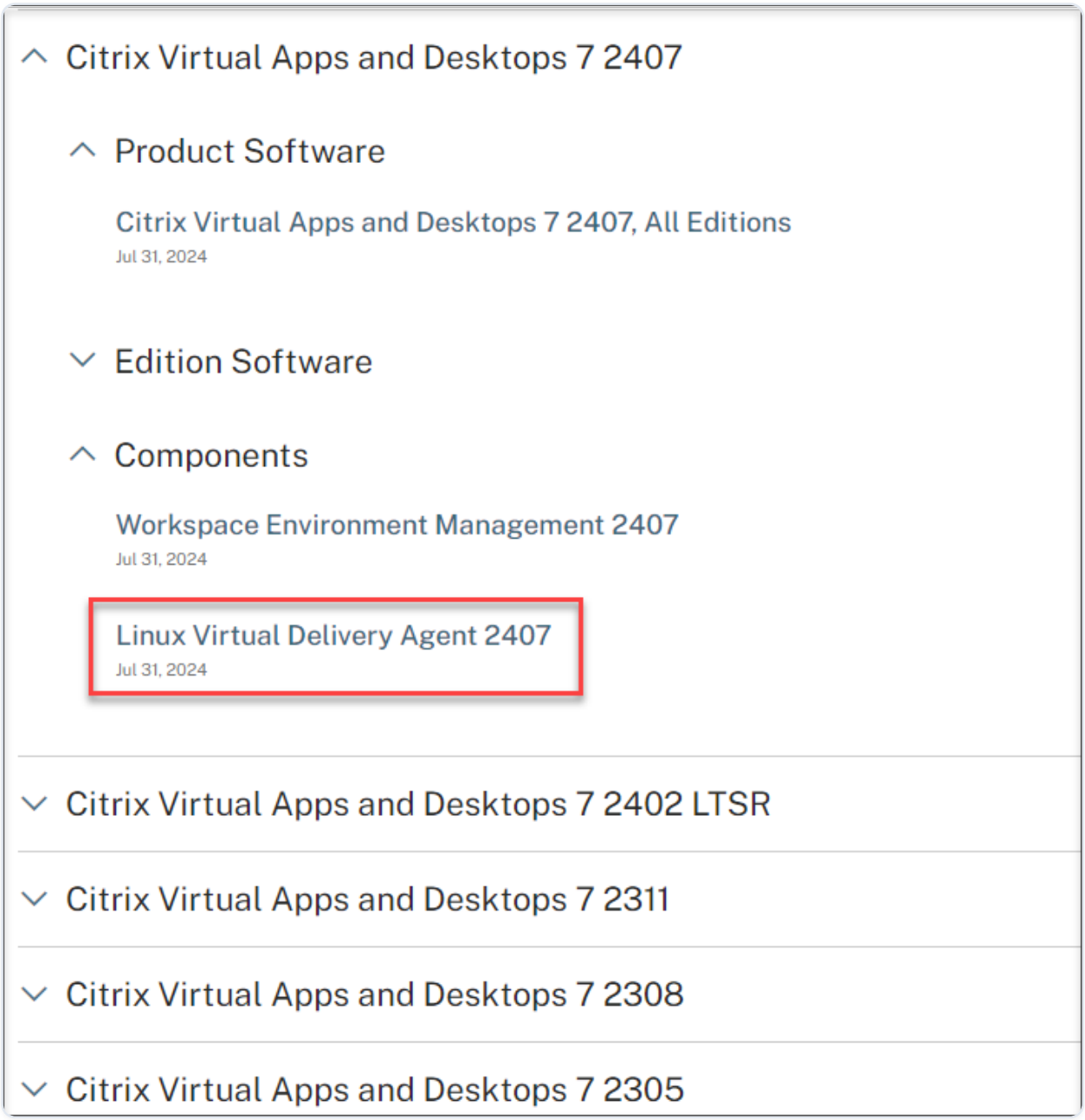
必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。そうでない場合は、Microsoftパッケージフィードから.NETをインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

.NETのインストール後、**which dotnet** コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

ステップ5：Linux VDAパッケージのダウンロード

1. Citrix Virtual Apps and Desktops ダウンロードページにアクセスします。
2. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
3. Linux VDAを見つけるには、**Components** を展開します。例：



4. Linux VDAのダウンロードにアクセスするには、Linux VDAリンクをクリックします。

   「Linux VDA のダウンロード」(/en-us/linux-virtual-delivery-agent/2411/media/linux-vda-downloads.png)

5. お使いのLinuxディストリビューションに合ったLinux VDAパッケージをダウンロードします。
6. Linux VDAパッケージの整合性を検証するために使用できるGPG公開鍵をダウンロードします。例：

   「GPG公開鍵」(/en-us/linux-virtual-delivery-agent/2411/media/gpg-key.png)

   Linux VDAパッケージの整合性を検証するには、次のコマンドを実行して公開鍵をDEBデータベースにインポートし、パッケージの整合性を確認します。

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

dpkg-sig --verify <path to the Linux VDA package>
```

ステップ6：Linux VDAをインストールする

ステップ6a：Linux VDAをインストールする

Debianパッケージマネージャーを使用してLinux VDAソフトウェアをインストールします。

Debian 12の場合：

```
sudo dpkg -i xendesktopvda_<version>.debian12_amd64.deb
```

Debian 11の場合：

```
sudo dpkg -i xendesktopvda_<version>.debian11_amd64.deb
```

Debian 12の依存関係リスト：

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.11

ufw >= 0.36

desktop-base >= 12.0.6

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.38

gtk3-nocsd >= 3

bash >= 5.2
```

```
findutils >= 4.9.0

sed >= 4.9

cups >= 2.4

ghostscript >= 10.0.0~

libmspack0 >= 0.11

ibus >= 1.5

libgoogle-perftools4 >= 2.10~

libpython3.11 >= 3.11~

libsasl2-modules-gssapi-mit >= 2.1.~

libnss3-tools >= 2:3.87

libqt5widgets5 >= 5.15~

mutter >= 43.8

libqrencode4 >= 4.1.1

libimlib2 >= 1.10.0

libfuse2 >= 2.9.9

pulseaudio-utils >= 16.1
```

Debian 11の依存関係リスト：

```
libnss3-tools >= 2:3.61



libfuse2 >= 2.9
```

```
openjdk-17-jdk >= 17

imagemagick >= 8:6.9.10

ufw >= 0.36

desktop-base >= 10.0.2

libxrandr2 >= 2:1.5.1

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.33

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.6.0
```

```
sed >= 4.7

cups >= 2.2

ghostscript >= 9.53~

libmspack0 >= 0.10

ibus >= 1.5

libgoogle-perftools4 >= 2.7~

libpython3.9 >= 3.9~

libsasl2-modules-gssapi-mit >= 2.1.~

libqt5widgets5 >= 5.5~

mutter >= 3.38.6~

libqrencode4 >= 4.0.0
```

```
libimlib2 >= 1.5.1
```

注:

このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、システム要件を参照してください。

ステップ6b：Linux VDAをアップグレードする（オプション）

Linux VDAは、最新バージョンからのアップグレードをサポートしています。たとえば、Linux VDAを2308から2311に、1912 LTSRから2203 LTSRにアップグレードできます。

```
sudo dpkg -i <PATH>/<Linux VDA deb>



sudo apt-get install -f
```

注:

- 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを行う前に、ファイルをバックアップしてください。
- 2407リリース以降、Linux VDAはアップグレード中の構成ファイルの処理をパッケージマネージャーの**rpm**または**dpkg**に委任します。以下に、**rpm**と**dpkg**が構成ファイルの変更とどのように相互作用するかを説明します。
  - **rpm**: デフォルトではローカルバージョンを保持し、パッケージからの新しいバージョンを**.rpmnew**拡張子で保存します。
  - **dpkg**: 続行方法の選択肢を対話形式で求めます。ローカル構成ファイルを保持し、新しいパッケージバージョンを**.dpkg-new**または**.dpkg-dist**として保存しながら、Linux VDAをサイレントアップグレードするには、次のコマンドを使用します。

```
dpkg --force-confold -i package.deb  # Always keep your version,
then save new package's version as *.dpkg-new or *.dpkg-dist
```

ステップ7：NVIDIA GRIDドライバーをインストールする

HDX 3D Proを有効にするには、ハイパーバイザーとVDAマシンにNVIDIA GRIDドライバーをインストールする必要があります。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールおよび構成するには、以下のガイドを参照してください。

- XenServer
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールおよび構成するには、次の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。
3. VMを起動します。
4. VMにゲストVMドライバー（クラウドベンダーまたはNVIDIAから）をインストールします。

ステップ8：Linux VDAを構成する

注:

ランタイム環境をセットアップする前に、OSに**en_US.UTF-8**ロケールがインストールされていることを確認してください。OSでロケールが利用できない場合は、**sudo locale-gen en_US.UTF-8**コマンドを実行します。Debianの場合、**/etc/locale.gen**ファイルを編集して**# en_US.UTF-8 UTF-8**行のコメントを解除し、**sudo locale-gen**コマンドを実行します。

パッケージをインストールした後、ctxsetup.sh スクリプトを実行して Linux VDA を構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がイ

> ンストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションは、環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求せず、スクリプトによるインストールプロセスを可能にします。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_NON_DOMAIN_JOINED='y\|n'** – マシンをドメインに結合するかどうか。 ドメイン。デフォルト値は「n」です。ドメイン参加シナリオの場合は、「n」に設定します。
- **CTX_XDL_AD_INTEGRATION='winbind\|sssd\|centrify\|pbis\|quest'** – Linux VDA は、Delivery Controller で認証するために Kerberos 構成設定を必要とします。Kerberos 構成は、システムにインストールおよび構成されている Active Directory 統合ツールから決定されます。
- **CTX_XDL_DDC_LIST='<list-ddc-fqdns\>'** – Linux VDA は、Delivery Controller に登録するために、Delivery Controller の完全修飾ドメイン名 (FQDN) のスペース区切りリストを必要とします。少なくとも1つの FQDN または CNAME を指定する必要があります。
- **CTX_XDL_VDI_MODE='y\|n'** – マシンを専用デスクトップ配信モデル (VDI) として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境の場合は、値を **'y'** に設定します。
- **CTX_XDL_HDX_3D_PRO='y\|n'** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ (シングルセッション) モード (つまり、CTX_XDL_VDI_MODE=**'y'**) 用に構成されます。

- **CTX_XDL_START_SERVICE='y\|n'** – 構成が完了したときに Linux VDA サービスを開始するかどうかを決定します。
- **CTX_XDL_REGISTER_SERVICE='y\|n'** – Linux Virtual Desktop サービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES='y\|n'** – Linux VDA サービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktop のシステムファイアウォールで、必要なポート (デフォルトではポート 80 と 1494) を自動的に開くことができます。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/'<none\>'** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。これを「**'<none\>'**」に設定すると、VDAで構成されているデフォルトのデスクトップが使用されます。

  コマンドを実行するか、システムトレイを使用することで、デスクトップ環境を切り替えることもできます。詳しくは、「デスクトップ切り替えコマンド」および「システムトレイ」を参照してください。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（ctxvda）をサポートするために.NETをインストールするパス。デフォルトのパスは「**'/usr/bin'**」です。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。
- **CTX_XDL_SITE_NAME=<dns-name\>** – Linux VDAは、DNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。不要な場合は、「**'<none\>'**」に設定します。
- **CTX_XDL_LDAP_LIST='<list-ldap-servers\>'** – Linux VDAは、DNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合は、LDAPポートを持つLDAP FQDNのスペース区切りリストを提供できます。例：ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。Active Directoryフォレスト内でLDAPクエリを高速化するには、ドメインコントローラーでグローバルカタログを有効にし、関連するLDAPポート番号を3268として指定します。この変数は、デフォルトで「**'<none\>'**」に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPにクエリを実行します。検索パフォーマンスを向上させるには、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。不要な場合は、「**'<none\>'**」に設定します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_NON_DOMAIN_JOINED='n'

export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest

export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'

export CTX_XDL_VDI_MODE='y|n'

export CTX_XDL_HDX_3D_PRO='y|n'

export CTX_XDL_START_SERVICE='y|n'

export CTX_XDL_REGISTER_SERVICE='y|n'

export CTX_XDL_ADD_FIREWALL_RULES='y|n'

export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>'

export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

export CTX_XDL_VDA_PORT='<port-number>'

export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'

export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'

export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'

export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

sudoコマンドを実行するときは、**-E**オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。前のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_NON_DOMAIN_JOINED='n' \

CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \

CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

CTX_XDL_VDI_MODE='y|n' \

CTX_XDL_HDX_3D_PRO='y|n' \

CTX_XDL_START_SERVICE='y|n' \

CTX_XDL_REGISTER_SERVICE='y|n' \

CTX_XDL_ADD_FIREWALL_RULES='y|n' \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|'<none>' \

CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>' \

CTX_XDL_VDA_PORT='<port-number>' \

CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>' \

CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>' \

CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>' \

CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n' \

/opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル **/tmp/xdl.configure.log** に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動してください。

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を確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

注:

Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前に設定されたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

ヒント:

> このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

ステップ9：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。詳細については、**XDPing**を参照してください。

ステップ10：Linux VDAを実行する

ctxsetup.shスクリプトを使用してLinux VDAを構成したら、以下のコマンドを使用してLinux VDAを制御します。

Linux VDAを起動する：

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

> **注:**
>
> **ctxvda** および **ctxhdx** サービスを停止する前に、**systemctl stop ctxmonitord** コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ11：マシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログを作成する」および「マシンカタログを管理する」を参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します。
  - ホストされた共有デスクトップ配信モデルの場合は、**マルチセッションOS** オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS** オプション。
- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

注:

> Citrix Studioの初期バージョンでは、「Linux OS」という概念はサポートされていませんでした。ただし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト型共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンあたり単一ユーザーの配信モデルが意味されます。

> **ヒント:**
>
> マシンをActive Directoryドメインから削除して再参加させる場合、マシンをマシンカタログから削除し、再度追加する必要があります。

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プロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループとWindowsマシンを含むマシンカタログを混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2411」を参照してください。

設定

このセクションでは、Linux VDA の機能について、機能の説明、構成、トラブルシューティングを含めて詳しく説明します。

管理

このセクションには、次のトピックが含まれています。

- Linux VDAデータ収集プログラム
- HDX インサイト
- Citrix Telemetry Serviceとの統合
- Citrix DaaS Standard for Azure向けLinux VDAの自己更新
- Linux VMおよびLinuxセッションメトリック
- ログ収集
- セッションシャドウイング
- モニターサービスデーモン
- トラブルシューティング
- その他
    - Citrix Workspaceアプリ for HTML5のサポート
    - **Python3**仮想環境の作成
    - NISとActive Directoryの統合
    - **IPv6**
    - LDAPS
    - **Xオーソリティ**

Linux VDA データ収集プログラム

Linux VDA のインストール後、データ収集プログラムに自動的に参加します。データ収集プログラムは、統計情報と使用状況データを収集し、Citrix 製品の品質とパフォーマンスの向上に役立てる

ため、そのデータを Citrix Analytics に送信します。データはセッション起動時にアップロードされます。

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なっています。次のレジストリ設定を使用して、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）

指定されていない場合、Linux VDA データ収集プログラムは有効になります。

Linux VDA データ収集プログラムを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

Linux VDA から収集されるデータ

次の表は、収集されるデータの種類の例を示しています。

データポイント	キー名	説明
マシン GUID	**machine_guid**	データの発信元となるマシンを識別するGUID文字列
Linux OS名とバージョン	**os_name_version**	マシンのLinux OS名とバージョンを示す文字列
ADソリューション	**ad_solution**	マシンのドメイン参加方法を示す文字列
Linuxカーネルバージョン	**kernel_version**	マシンのカーネルバージョンを示す文字列
GPUタイプ	**gpu_type**	VDAマシンのGPUタイプを示す
CPUコア数	**cpu_cores**	マシンのCPUコア数を示す整数

データポイント	キー名	説明
CPU周波数	**cpu_frequency**	CPU周波数をMHzで示す浮動小数点数
物理メモリサイズ	**memory_size**	物理メモリサイズをKBで示す整数
LVDAバージョン	**vda_version**	インストールされているLinux VDAのバージョンを示す文字列。
LVDAの更新または新規インストール	**update_or_fresh_install**	現在のLinux VDAパッケージが新規インストールされているか、更新されているかを示す文字列
VDIモードが有効かどうか	**vdi_mode**	VDIモードが有効になっているかどうかを示す文字列
LVDAのインストール方法	**install_method**	現在のLinux VDAパッケージがMCS、PVS、簡易インストール、または手動インストールを使用してインストールされていることを示す文字列。
HDX™ 3D Proが有効かどうか	**hdx_3d_pro**	マシンでHDX 3D Proが有効になっているかどうかを示す文字列
BCRが有効かどうか	**ビーシーアール**	このマシンでブラウザコンテンツリダイレクト (BCR) が有効になっているかどうかを示す文字列
システムロケール	**system_locale**	このマシンのロケールを示す文字列
ファームID	**farm_id**	ファームIDを示す文字列。
VDA仮想化の種類	**vda_virtualization**	VDAを作成したハイパーバイザーを示す文字列

データポイント	キー名	説明
セッションキー	**session_key**	データの発信元となるセッションを識別する
リソースの種類	**resource_type**	起動されたセッションのリソースタイプを示す文字列: デスクトップまたは `<appname>`
Receiverクライアントの種類	**receiver_type**	セッションの起動に使用されたCitrix Workspace™アプリの種類を示す整数
Receiverクライアントのバージョン	**receiver_version**	セッションの起動に使用されたCitrix Workspaceアプリのバージョンを示す文字列
印刷回数	**printing_count**	セッションが印刷機能を使用した回数を示す整数
USBリダイレクト回数	**usb_redirecting_count**	セッションがUSBデバイスを使用した回数を示す整数
Gfxプロバイダーの種類	**gfx_provider_type**	セッションのグラフィックプロバイダーの種類を示す文字列
シャドウイング回数	**shadow_count**	セッションがシャドウイングされた回数を示す整数
ユーザーが選択した言語	**ctxism_select**	ユーザーが選択したすべての言語を含む複合的な長い文字列
スマートカードリダイレクト回数	**scard_redirecting_count**	セッションログオンおよびセッション内アプリのユーザー認証にスマートカードリダイレクトが使用された回数を示す整数。

データポイント	キー名	説明
ビデオコーデックの種類	**grahpic_video_codec_type**	Thinwireにどのビデオコーデックが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	**credentials_type**	Linux VDAにログオンするために使用された資格情報の種類を示す整数。
透かし	**透かし**	セッションの透かしが有効になっているかどうかを示す文字列。
透かしの透明度	**watermark_transparency**	透かしの透明度を示す整数。
透かしカスタムテキスト長	**watermark_custom_text_len**	ウォーターマークのカスタムテキストの長さを表す整数
MTU	**MTU**	このセッションで最大伝送単位 (MTU) が使用されているかどうかを示す文字列
MTU MSS	**mtu_mss**	最大セグメントサイズ (MSS) を表す整数
ファイル転送	**ファイル転送**	ファイル転送ポリシー設定を表す整数
ファイル転送のアップロード回数	**filetrans_upload_count**	セッション中にファイル転送で「アップロード」アイコンが使用された回数を表す整数
ファイル転送のダウンロード回数	**filetrans_download_count**	セッション内でファイル転送のために「ダウンロード」アイコンが使用された回数を示す整数

HDX™ インサイト

概要

Linux VDA は、HDX Insight機能を部分的にサポートしています。

インストール

依存パッケージをインストールする必要はありません。

使用方法

HDX Insight は、Citrix Workspace™ アプリと Linux VDA の間で Citrix ADC を介して渡される ICA® メッセージを分析します。すべての HDX Insight データは NSAP 仮想チャネルから取得され、非圧縮で送信されます。NSAP 仮想チャネルはデフォルトで有効になっています。

次のコマンドは、NSAP 仮想チャネルの無効化と有効化をそれぞれ行います。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000000" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000001" --force
```

トラブルシューティング

データポイントが表示されない

考えられる原因は2つあります。

- HDX Insight が正しく構成されていません。

  たとえば、Citrix ADC で AppFlow® が有効になっていない、または Citrix ADM で誤った Citrix ADC インスタンスが構成されている場合などです。

- ICA Control Virtual Channel が Linux VDA で開始されていません。

  `ps aux | grep -i ctxctl`

  `ctxctl`が実行されていない場合は、管理者に連絡してCitrixにバグを報告してください。

アプリケーションデータポイントが表示されない

シームレス仮想チャネルが有効になっており、シームレスアプリケーションが実行されていることを確認してください。

Citrix Telemetry Serviceとの統合

Linux VDAソフトウェアにCitrix Telemetry Service (`ctxtelemetry`) を統合することで、Citrix Scoutを実行できます。Citrix Scoutは、/opt/Citrix/VDA/bin/xdlcollect.shスクリプトを使用してLinux VDAに関するログを収集します。

Citrix Scoutの画像(/en-us/linux-virtual-delivery-agent/2411/media/citrix-scout.png)

Citrix Telemetry Serviceを有効化および無効化する

- サービスを有効にするには、**sudo systemctl enable ctxtelemetry.socket**コマンドを実行します。
- サービスを無効にするには、**sudo systemctl disable ctxtelemetry.socket**を実行します。

ポート

Citrix Telemetry Service (`ctxtelemetry`) は、デフォルトでTCP/IPポート7503を使用してCitrix Scoutをリッスンします。Delivery ControllerではTCP/IPポート7502を使用してCitrix Scoutと通信します。

VDAのインストール後にポートを変更するには、次の手順を実行します。

1. Scoutとの通信に使用するポートを変更するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg create -k
   "HKLM\Software\Citrix\VirtualDesktopAgent" -t REG_DWORD -v
   "TelemetryServicePort" -d <port number> --force
   ```

2. Scoutをリッスンするためのソケットポートを変更するには、次のコマンドを実行してctxtelemetry.socketファイルを開き、編集します。

   ```
   sudo vi /etc/systemd/system/ctxtelemetry.socket
   ```

   ソケットポートの画像(/en-us/linux-virtual-delivery-agent/2411/media/socket-port.png)

3. ソケットポートを再起動するには、次のコマンドを実行します。

   ```
   sudo systemctl daemon-reload

   sudo systemctl stop ctxtelemetry.socket
   ```

```
sudo systemctl start ctxtelemetry.socket
```

4. ファイアウォール構成で新しいポートを有効にします。

   たとえば、Ubuntuを使用している場合は、**sudo ufw allow 7503**コマンドを実行してポート7503を有効にします。

注:

代わりにctxsetup.shを実行して、前述の手順3と4を自動化することもできます。

デバッグモード

Citrix Telemetry Serviceが期待どおりに動作しない場合は、デバッグモードを有効にして原因を特定できます。

1. デバッグモードを有効にするには、次のコマンドを実行して`ctxtelemetry`ファイルを開き、DebugModeの値を1に変更します。

   ```
   sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
   ```

   デバッグモードを有効にする(/en-us/linux-virtual-delivery-agent/2411/media/enable-debug-mode.png)

2. Citrix Telemetry Serviceを手動で停止するか、サービスが自動的に停止するまで15分間待ちます。

   テレメトリを停止(/en-us/linux-virtual-delivery-agent/2411/media/stop-telemetry.png)

   この例では、次のコマンドを実行してCitrix Telemetry Serviceを停止できます。

   ```
   sudo netstat -ntlp

   Kill -9 1958
   ```

3. Citrix Telemetry Serviceを再起動するには、ScoutでLinux VDAを選択し、/var/log/xdl/にあるtelemetry-debug.logを見つけます。

サービス待機時間

ソケットポートを開く`systemd`デーモンはデフォルトで起動し、リソースをほとんど使用しません。Citrix Telemetry Serviceはデフォルトで停止しており、Delivery Controllerからのログ収集要求があった場合にのみ起動します。ログ収集が完了すると、サービスは15分間新しい収集要求を待ち、要求がない場合は再度停止します。待機時間は次のコマンドで構成できます。最小値は10分です。10分未満の値を設定した場合、最小値である10分が適用されます。待機時間を設定したら、サービスを停止して再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"TelemetryServiceIdleTimeoutInMinutes" -d <number> -t REG_DWORD
```

検証テスト

収集が開始される前に、選択された各マシンに対して検証テストが自動的に実行されます。これらのテストは、要件が満たされていることを確認します。マシンのテストが失敗した場合、Scoutは推奨される修正措置を含むメッセージを表示します。検証テストの詳細については、Citrix Scoutドキュメントの検証テストセクションを参照してください。

Linux VDA の Azure を介したセルフアップデート

この機能は、Linux VDA ソフトウェアを即座に、またはスケジュールされた時間に自動的に更新するのに役立ちます。これは、Citrix DaaS Standard for Azure (旧称 Citrix Virtual Apps and Desktops Standard for Azure) で Linux VDA を作成する場合に役立ちます。Azure の VM の管理者権限は必要ありません。詳細については、「Citrix DaaS Standard for Azure で Linux VDA を作成する」を参照してください。

構成

この機能を使用するには、次の手順を完了します。

ステップ 1: 更新情報と新しい VDA パッケージを Azure コンテナにアップロードする

ステップ 1a: Azure ストレージアカウントの下にコンテナを作成し、コンテナのアクセスレベルを **Blob (匿名読み取りアクセスのみ)** に設定します。

> **注:**
>
> Azure コンテナと BLOB は、お客様が排他的に保持および管理します。Citrix® は、それらに関するセキュリティ問題について責任を負いません。データセキュリティとコスト効率を

確保するため、**セルフアップデート**のたびにコンテナのアクセスレベルを **Private (匿名アクセスなし)** に設定してください。

ステップ 1b: VDA 更新情報を UpdateInfo.json という名前の JSON ファイルに組み込みます。ファイル形式の例については、次のブロックを参照してください。

```
{

 "Version": "21.04.200.4",

"Distributions":[

{

"TargetOS": "RHEL7_9",

"PackageName": "",

"PackageHash": ""

},

{

"TargetOS": "UBUNTU20_04",

"PackageName": "",

"PackageHash": ""

}

]

}
```

ここで、**「Version」** は新しい VDA バージョンを示し、**「Distributions」** は更新オブジェクトの配列です。各オブジェクトには次の 3 つの項目が含まれます。

- 「**TargetOS**」：「RHEL7_9」（RHEL 7、CentOS 7、Amazon Linux 2 の場合）または「UBUNTU20_04」である必要があります。**ctxmonitord**` は他のディストリビューションを認識しません。
- 「**PackageName**」: 指定されたバージョンの VDA パッケージの完全な名前。
- 「**PackageHash**」：`shasum -a 256 <pkgname>` コマンドを使用して計算する SHA-256 値。

ステップ 1c: JSON ファイルと新しいバージョンの Linux VDA パッケージを Azure コンテナにアップロードします。

ステップ 2: マスターイメージまたは各 VDA でセルフアップデート機能を有効にする

デフォルトでは、**セルフアップデート**は無効になっています。Citrix DaaS Standard for Azure で Linux VDA を作成する場合、機能の有効化はマスターイメージで実行する必要があります。それ以外の場合は、各ターゲット VDA で直接機能を有効にします。

セルフアップデートを有効にするには、HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\SelfUpdate のレジストリキーを編集するために、次のようなコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_DWORD"
-v "fEnabled" -d "0x00000001" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"ScheduledTime" -d "Immediately" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"Url" -d "<Your-Azure-Container-Url>" --force



/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t "REG_SZ" -v
"CaCertificate" -d "<Local-Certificate-Path-of-PortalAzureCom>" --force
```

以下の表は、レジストリ設定について説明しています。

レジストリ設定	説明
fEnabled	この設定は必須です。デフォルトでは値は0で、**セルフアップデート**が無効であることを意味します。1に設定すると、**セルフアップデート**を有効にできます。
Url	この設定は必須です。更新情報と新しいVDAパッケージを取得するためのAzureコンテナのURLを設定します。
ScheduledTime	この設定は必須です。**Immediately**または**NextStart**に設定できます。**Immediately**は、VDAパッケージのダウンロード直後にアップデートを実行することを意味します。このオプションは、ダウンロード速度が速く、アップデートが緊急である場合に適しています。ただし、パッケージのダウンロード中にライブセッションがある場合、ユーザーエクスペリエンスを妨げる可能性があります。**NextStart**は、**ctxmonitord**の次回の起動時にアップデートを実行することを意味します。このオプションは、ダウンロード速度があまり速くなく、アップデートが緊急ではない場合に適しています。
CaCertificate	この設定はオプションです。Azureコンテナの URLを検証するためのPEM証明書のフルパスを設定します。Azure BLOBの場合、portal.azure.comの証明書をブラウザから取得し、PEMに変換したものを使用できます。セキュリティのため、このレジストリ設定を追加することをお勧めしますが、Ubuntu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際に、`curl`コマンドが期待どおりに機能しないためです。証明書の権限を最小限に設定してください。

ctxmonitordが再起動すると、まず**Url**を照会してUpdateInfo.jsonファイルを取得し、JSONファイルから更新バージョンを取得します。次に、**ctxmonitord**は更新バージョンを現在のバージョンと比較します。現在のバージョンが古い場合、サービスはAzureからVDAパッケージの新しいバージョンをダウンロードし、ローカルに保存します。その後、**ScheduledTime**の設定に従ってアップデートを実行します。オンプレミス展開の場合、**ctxmonitord**を直接再起動してアップデートをトリガーできます。ただし、VMに対する管理者権限がないCitrix DaaS Standard for Azureでは、**ctxmonitord**はVDAマシンが再起動された後にのみ再起動できます。アップデートが失敗した場合、VDAは既存のバージョンにロールバックされます。

注:

- マスターイメージで構成したレジストリ設定は変更できません。
- 環境内のすべてのVMが同時にパッケージをダウンロードすると、ローカルネットワークが混雑する可能性があります。
- 指定されたURLからダウンロードするには、ターゲットVMに`curl`をインストールする必要があります。
- アップデートとロールバックの両方が失敗した場合、ユーザーデータは失われます。
- アップデートが失敗し、ロールバックが成功した場合、同じネットワーク上のユーザーは異なるバージョンのLinux VDAを使用している可能性があります。この状況は最適ではありません。
- アップデートの完了には通常数分かかります。Citrix Studioにはステータスインジケーターがありません。

Linux VMとLinuxセッションのメトリック

次の表に、Linux VMおよびLinuxセッションで利用可能なメトリックを示します。

メトリック	必要な最小VDAバージョン	説明	備考
ログオン時間	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	Citrix DirectorおよびMonitorで利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
		セッションが使用可能になるまでのログオンプロセスを測定したものです。**HDX™接続**、**認証**、**GPO**など、ログオンプロセスの各フェーズにかかる時間は、Linux VDA 2402以降で利用できます。ログオン時間のメトリックを表示するには、**トレンド**ビューで**ログオンパフォーマンス**タブを選択します。	
ICA®遅延	2311	ICA遅延は、ネットワークが遅いかどうかを示すネットワーク遅延です。このメトリックにアクセスするには、**セッションの詳細**ビューを開きます。	Citrix DirectorおよびMonitorで利用可能です。
ポリシー	2311	現在のセッションに適用されているすべてのポリシーは、**セッションの詳細**ビューの**ポリシー**タブに表示されます。	Citrix DirectorおよびMonitorで利用可能です。
セッションの自動再接続回数	2109	セッション内の自動再接続の数を表示するには、**トレンド**ビューにアクセスします。条件を設定し、**適用**をクリックして検索結果を絞り	Citrix DirectorおよびMonitorで利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
		込みます。**セッションの自動再接続回数**列には、セッション内の自動再接続の数が表示されます。**自動再接続**は、**セッションの信頼性**または**クライアントの自動再接続**ポリシーが有効な場合に有効になります。セッションの再接続と関連ポリシーの詳細については、次の記事を参照してください。 セッション クライアント自動再接続ポリシー設定 セッションの信頼性ポリシー設定	
アイドル時間	2103	このメトリックにアクセスするには、「**フィルター > セッション > すべてのセッション**」を選択して「**すべてのセッション**」ページを開きます。	Citrix DirectorおよびMonitorで利用可能です。
Linux VMのメトリック	2103	Linux VMのメトリックは、CPUコア数、メモリサイズ、ハードディスク容量、および現在と過去のCPUとメモリの使用率です。	Citrix DirectorおよびMonitorで利用可能です。

メトリック	必要な最小VDAバージョン	説明	備考
プロトコル	1909	Linuxセッションのトランスポートプロトコルは、「**セッションの詳細**」ビューにUDPまたはTCPとして表示されます。	Citrix DirectorおよびMonitorで利用可能です。
ICA RTT	1903	ICAラウンドトリップ時間（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メトリックを取得するには、Citrix Studioで「**ICAラウンドトリップ計算**」および「**ICAラウンドトリップ計算間隔**」ポリシーを作成します。	Citrix DirectorおよびMonitorで利用可能です。

ログ収集

ログメカニズムの概要

以下の表は、Linux VDAのログメカニズムの概要を示します。

ロギングモジュール	ログファイル名の形式	ログファイル名の例	ロギングスコープ	単一ファイルの最大サイズ	ローテーションしきい値（デフォルト）	構成
HDX	hdx.log[.n]	hdx.log、hdx.log.1	グラフィック、ログイン、オーディオ、キー	200 MiB	現在1つ、古い2つ	setlog ユーティリティを介して構成可能

ロギングモジュール	ログファイル名の形式	ログファイル名の例	ロギングスコープ	単一ファイルの最大サイズ	ローテーションしきい値（デフォルト）	構成
			ボード、マウス			
ジェイプロキシ	jproxy.log[.n.log]	jproxy.log, jproxy.log.1.log	VDA登録、ユーザー認証	20 MiB	現在のファイル1つ、古いファイル10個	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.log	vda.2024-05-06.20.18.40.log	VDA登録	50 MiB	現在のファイル1つ、古いファイル1つ	setlogユーティリティまたは/etc/xdl/brokeragent.confで構成可能

注:

- 最初のログファイルには名前に番号がなく、後続のファイルには「.n」という番号が付けられます。ここで「n」はファイル番号を表します。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番目のファイルです。
- 単一のログファイルの最大サイズはメビバイト (MiB) で測定されます。
- 生成中で、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、「現在の」ログファイルと呼ばれます。「現在の」ログファイルが単一ファイルの最大サイズに達すると、ロールオーバーされて「古い」ログファイルになります。
- 保持できる「古い」ログファイルの数を制限するために、ローテーションしきい値を構成できます。制限に達すると、最も古いログファイルが削除されます。

ロギング構成

このセクションでは、上記の表に記載されている詳細を補足する、ロギング構成に関する追加情報を提供します。

Linux VDAでデフォルトで有効になっているロギング

ctxlogdデーモンとsetlogユーティリティは、Linux VDAリリースパッケージに含まれています。デフォルトでは、ctxlogdデーモンはLinux VDAのインストールと構成後に起動します。トレースされる他のすべてのサービスはctxlogdデーモンに依存します。Linux VDAのトレースを維持したくない場合は、ctxlogdデーモンを停止できます。

/etc/xdl/brokeragent.conf を介したVDAロギングの構成

> **注:**

> 単一のログファイルの最大サイズなど、他のロギングパラメーターを詳しく調べずにVDAロギングのログレベルのみを構成したい場合は、この記事の後半で説明するsetlogユーティリティを使用できます。それ以外の場合は、/etc/xdl/brokeragent.confを使用してください。

> VDA上の/etc/xdl/brokeragent.confファイルは、VDAロギングの構成に利用できます。例：

```
<!-- VDA trace settings -->
<add key="TraceProvider.Type" value="QueuedLogFileProvider"/>
<add key="TraceProvider.BaseFilename" value="/var/log/xdl/vda.log"/>
<add key="TraceProvider.AppendMode" value="true"/>
<add key="TraceProvider.SizeLimit" value="50"/>
<add key="TraceProvider.TrailCount" value="2"/>
<add key="TraceProvider.ProcessSafe" value="true"/>
<add key="TraceProvider.CategoryFilter" value="All" />
```

/etc/xdl/log4j2.xml を介したJproxyロギングの構成

> **注:**

> 単一のログファイルの最大サイズなど、他のロギングパラメーターを詳しく調べずにJproxyロギングのログレベルのみを構成したい場合は、この記事の後半で説明するsetlogユーティリティを使用できます。それ以外の場合は、/etc/xdl/log4j2.xmlを使用してください。

以下は、/etc/xdl/log4j2.xml を介してJproxyロギングを構成する例です。この例では、**SizeBasedTriggeringPolicy** パラメーターが単一のJproxyログファイルの最大サイズを指定し、**DefaultRollerStrategy** パラメーターが保持できるJproxyログファイルの総数を設定します。

```
<!-- Logging Properties -->
<Properties>
    <Property name="LOG_PATTERN">%d{yyyy-MM-dd HH:mm:ss.SSS}{GMT} [%-5p] [%tid] - %m%n</Property>
    <Property name="logfile">/var/log/xdl/jproxy.log</Property>
    <Property name="perflogfile">/var/log/xdl/jproxyperf.log</Property>
</Properties>

<Appenders>

  <Console name="CONSOLE" target="SYSTEM_OUT">
    <PatternLayout pattern="${LOG_PATTERN}"/>
  </Console>

  <RollingFile name="logfile" fileName="${sys:logfile}"
    filePattern="${sys:logfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
        <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>

  <RollingFile name="perflogfile" fileName="${sys:perflogfile}"
    filePattern="${sys:perflogfile}.%i.log">
    <PatternLayout pattern="${LOG_PATTERN}"/>
    <Policies>
        <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>
```

setlogユーティリティを介したHDX™ロギングの構成

setlogユーティリティは/opt/Citrix/VDA/bin/パスにあります。rootユーザーのみが実行する権限を持っています。GUIを使用するか、コマンドを実行して、値（ログファイルパス、単一ファイルサイズ制限、ローテーションしきい値）やログレベルを含む構成オプションを表示および変更できます。setlogユーティリティのヘルプについては、次のコマンドを実行してください。

```
setlog help
```

値

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200 MiBで、最大2つの「古い」HDXログファイルを**/var/log/xdl/hdx.log**に保存できます。

現在のすべてのsetlog値を表示するには、次のコマンドを実行します。

```
setlog values
```

```
log_path (Log Output Path) = /var/log/xdl/hdx.log


log_size (Max Log Size (MiB)) = 200


log_count (Max Old Log Files) = 2
```

単一のsetlog値を表示または設定するには、次のコマンドを実行します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

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は、ほとんどのシナリオでログレベルを**verbose**に設定することをお勧めします。その後、問題を再現してログを収集してください。

ログレベル（Disabled、Inherited、Trace、Verbose、Information、Warnings、Errors、Fatal Errorsを含む）を設定するには、次のコマンドを実行します。

```
setlog level <class> [<level>]
```

ログレベル	コマンドパラメーター（大文字と小文字を区別しない）
無効	なし
継承	継承
トレース	トレース
詳細	詳細

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）
情報	情報
警告	警告
エラー	エラー
致命的なエラー	致命的

`<class>` 変数は、Linux VDA のコンポーネントを指定します。すべてのコンポーネントを対象とするには、それを all に設定します。例:

```
setlog level all error
```

サポートされているすべてのクラスまたはコンポーネントを表示するには、次のコマンドを実行します:

```
setlog levels
```

デフォルトに戻す

すべてのレベルと値をデフォルト設定に戻します:

```
setlog default
```

重要:

ctxlogd サービスは、ルートユーザーのみが作成できる **/var/xdl/.ctxlog** ファイルを使用して構成されます。他のユーザーはこのファイルへの書き込み権限を持っていません。ルートユーザーは他のユーザーに書き込み権限を与えないことをお勧めします。これに従わない場合、**ctxlogd** への任意の、または悪意のある構成が発生し、サーバーのパフォーマンス、ひいてはユーザーエクスペリエンスに影響を与える可能性があります。

ログ収集

ログを収集するには、**bash /opt/Citrix/VDA/bin/xdlcollect.sh** コマンドを実行します。ログの収集に使用される **xdlcollect** Bash スクリプトは、Linux VDA ソフトウェアに統合されており、**/opt/Citrix/VDA/bin** にあります。

ログ収集が完了すると、VDA 上の **/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 ファイルが見つからない場合（誤って削除された場合など）、**ctxlogd** デーモンは失敗し、**ctxlogd** サービスを再起動できません。

/var/log/messages:

```
Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
configuration file.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited,
code=exited, status=1/FAILURE



Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
```

この問題を解決するには、ルートユーザーとして setlog を実行して **/var/xdl/.ctxlog** ファイルを再作成します。その後、他のサービスが依存している **ctxlogd** サービスを再起動します。

セッションシャドウイング

セッションシャドウイングにより、ドメイン管理者はイントラネット内でユーザーのICA®セッションを表示できます。この機能は、noVNCを使用してICAセッションに接続します。

> **注:**
>
> この機能を使用するには、`Citrix Director` 7.16以降を使用してください。

インストールと構成

依存関係

セッションシャドウイングには、`python-websockify`と`x11vnc`の2つの新しい依存関係が必要です。Linux VDAのインストール後に、`python-websockify`と`x11vnc`を手動でインストールしてください。

Amazon Linux2の場合:

`python-websockify`と`x11vnc`をインストールするには、次のコマンドを実行します(`x11vnc`バージョン0.9.13以降):

```
sudo pip3 install websockify

sudo yum install x11vnc
```

RHEL 9.x/8.x および Rocky Linux 9.x/8.x の場合:

`python-websockify`と`x11vnc`をインストールするには、次のコマンドを実行します(`x11vnc`バージョン0.9.13以降)。

```
sudo pip3 install websockify

sudo yum install x11vnc
```

`x11vnc`を解決するには、EPELおよびCodeReady Linux Builderリポジトリを有効にします。

```
dnf install -y --nogpgcheck https://dl.fedoraproject.org/pub/epel/epel-
release-latest-8.noarch.rpm



subscription-manager repos --enable "codeready-builder  -for-rhel-8-
x86_64-rpms"
```

Ubuntuの場合:

`python-websockify`と`x11vnc`をインストールするには、次のコマンドを実行します(`x11vnc`バージョン0.9.13以降):

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

SUSEの場合:

まず、YaSTまたは次の**SUSEConnect**コマンドのいずれかを使用して、「SUSE Linux Enterprise Workstation Extension 15 SP6」モジュールを有効にします。

```
suseconnect -p sle-we/15.6/x86_64 -r <regcode>
```

詳細については、SUSEのドキュメントを参照してください: `<https://documentation.suse.com/ja-jp/sles/15-SP6/html/SLES-all/article-modules.html>`。

次に、以下のコマンドを実行して `python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールします。

```
sudo pip3 install websockify

sudo zypper install x11vnc
```

Debian 12 の場合:

以下のコマンドを実行して `python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールします。

```
apt install python3-websockify

sudo apt-get install x11vnc
```

Debian 11 の場合:

以下のコマンドを実行して `python-websockify` と `x11vnc` (`x11vnc` バージョン 0.9.13 以降) をインストールします。

```
sudo pip3 install websockify

sudo apt-get install x11vnc
```

ポート

セッションシャドウイング機能は、Linux VDA から `Citrix Director` への接続を確立するために、6001～6099 の範囲から利用可能なポートを自動的に選択します。したがって、同時にシャドウイングできる ICA セッションの数は 99 に制限されます。特にマルチセッションシャドウイングの場合、要件を満たすのに十分なポートが利用可能であることを確認してください。

レジストリ

次の表に、関連するレジストリを示します。

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションシャドウイング機能を有効または無効にします	1 (有効)
ShadowingUseSSL	Linux VDA と Citrix Director 間の接続を暗号化するかどうかを決定します	0 (無効)

レジストリ値を変更するには、Linux VDA で `ctxreg` コマンドを実行します。たとえば、セッションシャドウイングを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d
"0x00000000"
```

SSL

Linux VDAとCitrix Director間のnoVNC接続はWebSocketプロトコルを使用します。セッションシャドウイングの場合、`ws://`と`wss://`のどちらが選択されるかは、前述の「ShadowingUseSSL」レジストリによって異なります。デフォルトでは、`ws://`が選択されます。ただし、セキュリティ上の理由から、`wss://`を使用し、各Citrix Directorクライアントと各Linux VDAサーバーに証明書をインストールすることをお勧めします。Citrixは、`ws://`を使用してLinux VDAセッションシャドウイングを行う場合のセキュリティ上の責任を一切負いません。

SSLを有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "ShadowingUseSSL" -d
"0x00000001"
```

サーバーおよびルートSSL証明書を取得する

証明書は、信頼された認証局（CA）によって署名されている必要があります。

SSLを構成する各Linux VDAサーバーには、個別のサーバー証明書（キーを含む）が必要です。サーバー証明書は特定のコンピューターを識別するため、各サーバーの完全修飾ドメイン名（FQDN）を知っている必要があります。便宜上、ドメイン全体にワイルドカード証明書を使用することを検討してください。

Linux VDAと通信する各Citrix Directorクライアントにもルート証明書が必要です。ルート証明書は、サーバー証明書を発行するのと同じCAから入手できます。

以下のCAからサーバーおよびクライアント証明書をインストールできます。

- オペレーティングシステムにバンドルされているCA
- エンタープライズCA（組織がアクセスを許可しているCA）
- オペレーティングシステムにバンドルされていないCA

証明書を取得するためにどの方法が必要かについては、組織のセキュリティチームに相談してください。

> **重要:**
>
> - サーバー証明書のコモンネームは、Linux VDAの正確なFQDNであるか、少なくとも正しいワイルドカードとドメイン文字である必要があります。例：vda1.basedomain.comまたは*.basedomain.com。
> - SHA1やMD5を含むハッシュアルゴリズムは、一部のブラウザがサポートするデジタル証明書の署名には弱すぎます。そのため、SHA-256が最小標準として指定されています。
> - Chromeは、自己署名SSL証明書を安全でないと見なし、受け入れなくなりました。`NET::ERR_CERT_COMMON_NAME_INVALID`エラーは、生成された証明書にSAN（subjectAltName）フィールドがないために発生します。この問題を解決するには、SANフィールドを含む拡張属性（X509 v3拡張）を持つ証明書を提供してください。

各CITRIX DIRECTORクライアントにルート証明書をインストールする

セッションシャドウイングはIISと同じレジストリベースの証明書ストアを使用するため、IISまたはMicrosoft Management Console (MMC) の証明書スナップインを使用してルート証明書をインストールできます。CAから証明書を受け取ったら、IISでWebサーバー証明書ウィザードを再起動すると、ウィザードが証明書をインストールします。または、MMCを使用してコンピューター上の

証明書を表示およびインポートし、スタンドアロンスナップインとして証明書を追加することもできます。Internet ExplorerとGoogle Chromeは、デフォルトでオペレーティングシステムにインストールされている証明書をインポートします。Mozilla Firefoxの場合は、Certificate Managerの**Authorities**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.*」という名前を付けます（*は形式を示します。例：shadowingcert.pemおよびshadowingkey.key）。サーバー証明書とキーファイルをパス **/etc/xdl/shadowingssl** の下に置き、**ctxsrvr** のみが読み取りアクセス権を持つように、制限されたアクセス許可で適切に保護します。名前またはパスが正しくないと、Linux VDAは特定の証明書またはキーファイルを見つけることができず、`Citrix Director` との接続失敗の原因となります。コマンドは次のとおりです。

```
cp <vda's-public-key> /etc/xdl/shadowingssl/shadowingcert.pem

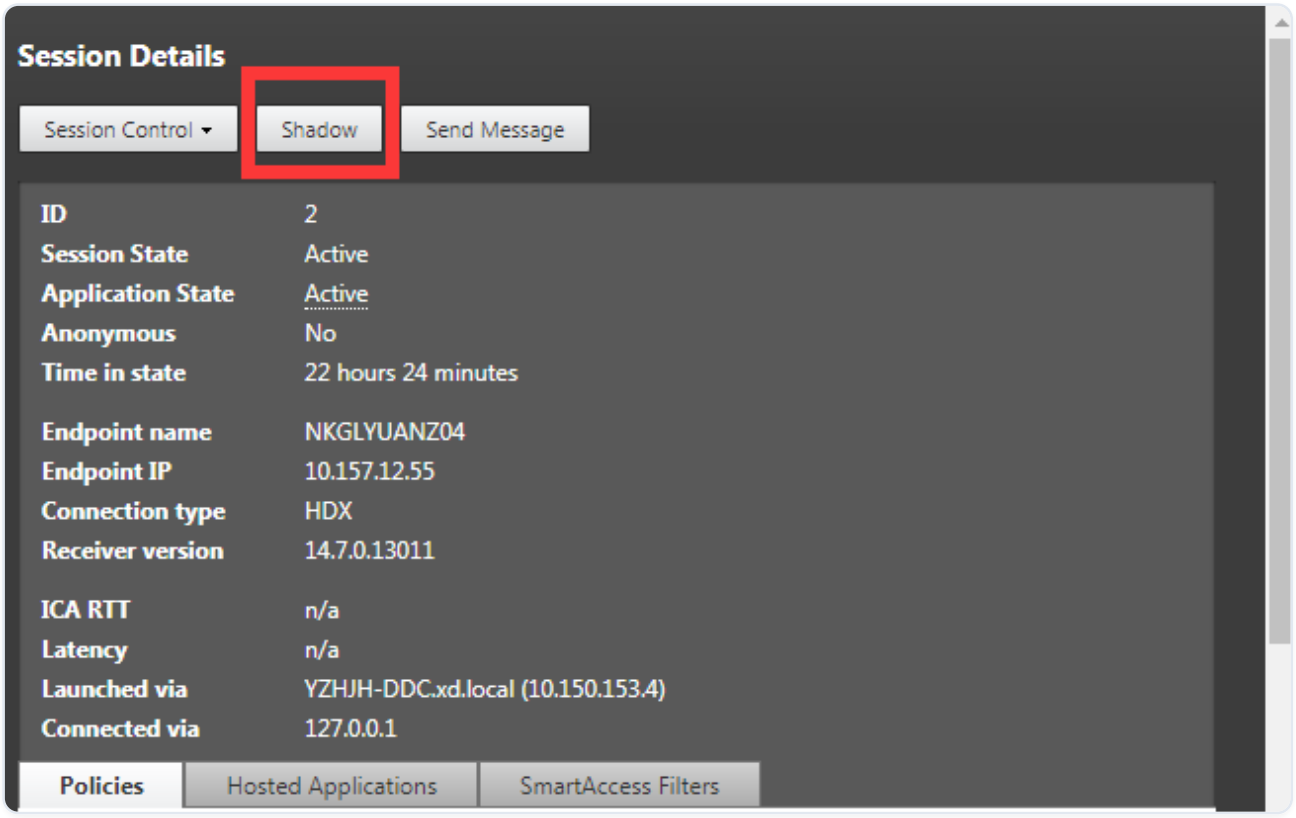
cp <vda's-server-private-key> /etc/xdl/shadowingssl/shadowingkey.key

sudo chown ctxsrvr:ctxadm  /etc/xdl/shadowingssl/shadowingcert.pem

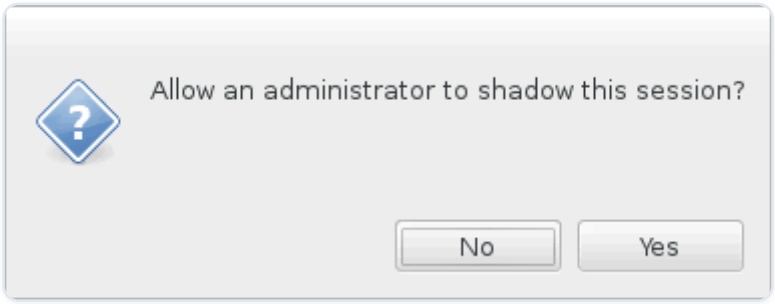
sudo chown ctxsrvr:ctxadm  /etc/xdl/shadowingssl/shadowingkey.key
```

使用方法

`Citrix Director` から、ターゲットセッションを見つけ、**Session Details**ビューで **Shadow** をクリックして、Linux VDAにシャドウ要求を送信します。



接続が初期化されると、ICAセッションクライアント（`Citrix Director` クライアントではない）に確認が表示され、ユーザーにセッションをシャドウする許可を求めます。



ユーザーが **Yes** をクリックすると、`Citrix Director` 側にウィンドウが表示され、ICAセッションがシャドウされていることを示します。

使用方法の詳細については、Citrix Directorドキュメント を参照してください。

制限事項

- VDAがドメインに参加しており、認証にAzure Active Directory (AAD) を使用してMicrosoft Azureでホストされている場合、セッションシャドウ機能は機能しません。
- セッションシャドウイングは、イントラネット内でのみ使用するように設計されています。Citrix Gateway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。Citrixは、外部ネットワークでのLinux VDAセッションシャドウイングに対する一切の責任を負いません。
- セッションシャドウイングが有効になっている場合、ドメイン管理者はICAセッションを表示することしかできず、書き込みまたは制御する権限はありません。

- 管理者が`Citrix Director`から**Shadow**をクリックすると、確認が表示され、ユーザーにセッションをシャドウする許可を求めます。セッションは、セッションユーザーが許可を与えた場合にのみシャドウできます。
- 前述の確認には20秒のタイムアウト制限があります。時間が経過すると、シャドウ要求は失敗します。
- 1つのセッションは1人の管理者のみがシャドウできます。たとえば、管理者Bが管理者Aがシャドウしているセッションに対してシャドウ要求を送信した場合、ユーザーの許可を得るための確認がユーザーデバイスに再表示されます。ユーザーが同意した場合、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が構築されます。管理者が同じセッションに対して別のシャドウ要求を送信した場合も、新しいシャドウ接続を構築できます。
- セッションシャドウイングを使用するには、`Citrix Director`7.16以降をインストールします。
- `Citrix Director`クライアントは、ターゲットのLinux VDAサーバーに接続するために、IPアドレスではなくFQDNを使用します。したがって、`Citrix Director`クライアントはLinux VDAサーバーのFQDNを解決できる必要があります。

トラブルシューティング

セッションシャドウイングが失敗した場合は、`Citrix Director`クライアントとLinux VDAの両方でデバッグを実行します。

Citrix Directorクライアントで

ブラウザーの開発者ツールを使用して、**コンソール**タブで出力ログを確認します。または、**ネットワーク**タブでShadowLinuxSession APIの応答を確認します。ユーザー権限取得の確認が表示されても接続の確立に失敗する場合は、VDAのFQDNを手動でpingして、`Citrix Director`がFQDNを解決できることを確認します。`wss://`接続に問題がある場合は、証明書を確認してください。

リナックス VDA で

1. `/var/log/xdl/vda.log`ファイルで手がかりを確認します。
2. `/var/xdl/sessionshadowing.sh`ファイルを編集し、「logFile」変数を変更して、Directorからのセッションシャドウイング中に手がかりを追跡できるログファイルを指定します。
3. また、noVNC接続で証明書が正しく機能するかどうかを手動で確認することもできます。
    a. **ps aux \| grep xorg**を実行して、現在のセッションのXorgディスプレイ番号**$display-num**（例: **:3**）を見つけます。
    b. 次のコマンドを実行してx11vncサーバーを起動し、受信接続を待ちます。

> **注:**
>
> 次のコマンドを実行する前に、**$passwd**、**$port**、**$display-num**の各変数を設定します。

```
runuser -l "ctxsrvr" -s /bin/bash -c "websockify <port> -v --cert  /
etc/xdl/shadowingssl/shadowingcert.pem --key  /etc/xdl/shadowingssl/
shadowingkey.key -- x11vnc  -viewonly  -shared -passwd $passwd
-rfbport $port -display $display-num -many -o /var/log/xdl/
x11vnc.log"
```

c. 次のようにnoVNCを使用して接続し、SSLモードを確認します。VDAのFQDNとポート番号を入力します。この例では、ポート番号は6009です。

noVNCを使用して接続する(/en-us/linux-virtual-delivery-agent/2411/media/connect-using-novnc.png)

d. VDA上のWebsockifyによって出力されたエラー、またはクライアント上のブラウザーによって報告されたエラーを解決します。

接続確立中の主要な確認事項:

i. セッションシャドウイングがポートを開くのを妨げるファイアウォールの制限がないか確認します。

ii. SSLシナリオの場合、証明書とキーファイルが適切に命名され、正しいパスに配置されていることを確認します。

iii. 新しいシャドウイング要求のために、6001～6099の間で十分なポートが残っていることを確認します。

iv. `openssl x509 -in shadowingcert.pem -text -noout`を実行して、証明書が正しく構成されていることを確認します。特にCNフィールドとSANフィールドに注意してください。

v. RHEL 8では、`rebind.so`が見つからないという問題が発生する可能性があります。この問題を解決するには、次のコマンドを実行します。

```
ln -s /usr/bin/rebind.so /usr/local/bin/rebind.so
```

モニターサービスデーモン

モニターサービスである **ctxmonitord** は、Linux VDA のヘルスステータスを監視するデーモンです。これは次の 3 つのモジュールで構成されています。

- 主要プロセス監視モジュール: このモジュールは、予期せぬ障害発生時に Systemd に主要サービスの自動再起動を委任し、ターゲットプロセスのステータスを定期的にスキャンして記録し、Xorg の残存ファイルをタイムリーにクリーンアップします。ログは **/var/log/xdl/ms.log** に保存されます。
- XDPing 統合モジュール: このモジュールは、XDPing 分析タスクとバックアップタスクを定期的に実行します。これらのタスクからのすべての出力は **/var/log/xdl/msxdping.log** で確認できます。
- Linux VDA 自己更新モジュール: 詳しくは、「Azure を介した Linux VDA の自己更新」を参照してください。

構成

デフォルトでは、VDA を起動するとモニターサービス **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 を介して構成できます。

ctxmonitord.conf

この構成ファイルは、モニターサービス ctxmonitord とそのモジュールの動作を指定します。デフォルトでは、次のように構成されています。

```
     ; This is the configuration file for ctxmonitord service




; Section 'service' configures the key process monitor module

[service]

MonitorEnable=true     ; true or false

DetectInterval=300      ; in seconds, minimum is 60



; Section 'xdping' configures the XDPing integration module
```

```
[xdping]

XdpingEnable=true      ; true or false

AnalysisInterval=600    ; in seconds, minimum is 60, 0 means disable

BackupInterval=1        ; in days, 0 means disable



; Section 'rules' configures the rules about how to monitor each key
process

; Each rule should be named as rules.<ProcessName> where <ProcessName> is
the name of the process to be monitored

; 'ProcessName' should be the name of the target process that is picked up
from whitelist.conf

; 'MonitorType' should be 'HealthCheck' or 'ResidueCleanup'. All
'MonitorType' should be 'HealthCheck' except for 'Xorg'

[rules]

[rules.ctxhdx]

ProcessName=ctxhdx

MonitorType=HealthCheck



[rules.ctxvda]

ProcessName=ctxvda

MonitorType=HealthCheck



[rules.ctxjproxy]

ProcessName=ctxjproxy
```

```
MonitorType=HealthCheck



[rules.ctxpolicyd]

ProcessName=ctxjproxy

MonitorType=HealthCheck



[rules.ctxlogd]

ProcessName=ctxlogd

MonitorType=HealthCheck



[rules.xorg]

ProcessName=Xorg

MonitorType=ResidueCleanup
```

whitelist.conf

この構成ファイルは、監視対象のターゲットプロセスのホワイトリストを指定します。ctxmonitord.conf ファイルで指定されたターゲットサービスは、whitelist.conf ファイルにもリストされている必要があります。デフォルトでは、次のように構成されています。

```
ctxhdx

ctxvda

ctxjproxy

ctxlogd

ctxcdm
```

```
ctxcups

ctxpolicyd

ctxusbsd

ctxceip

ctxsdcd

ctxrunatboot

ctxgdtd

ctxbcrd

ctxfidod

ctxpfwd

ctxwcamsd

ctxcertmgr

ctxscardsd  # Except for SLES

ctxfedsd    # Only for RHEL 8/9

Xorg
```

注:

- ctxmonitord.conf および whitelist.conf ファイルでの変更を有効にするには、**systemctl restart ctxmonitord** コマンドを実行して、モニターサービスデーモンを再起動します。
- 予期せぬ障害発生時にサービスが再起動する方法を構成するには、**/usr/lib/systemd/system/ctx*.service** ファイルを編集します。たとえば、以下はデフォルトの再起動オプションです。

```
Restart=on-failure

RestartSec=5

StartLimitInterval=60

StartLimitBurst=3
```

トラブルシューティング

この記事では、**XDPing** を使用したトラブルシューティングと、**ctxsdcutil** ユーティリティを使用したセッションメトリックのクエリ方法について説明します。

エックスディーピング

Linux **XDPing** ツールはコマンドラインアプリケーションです。Linux VDA環境における一般的な構成の問題を自動的にチェックします。

Linux XDPing ツールのインストール

ctxsetup.sh を実行しても **XDPing** はインストールされません。**XDPing** をインストールするには、`sudo /opt/Citrix/VDA/bin/xdping` を実行します。 このコマンドは、**XDPing** に必要な **Python3** 仮想環境も作成します。このコマンドで **Python3** 仮想環境の作成に失敗した場合は、「Create a **Python3** virtual environment」の手順に従って手動で作成してください。

pip ツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼できるホストを /etc/pip.conf ファイルに追加することを検討してください。

`[global]`

`trusted-host =` `pypi.org`

`files.pythonhosted.org`

XDPing で実行できるタスク

XDPing には、コマンドシェルから実行される **xdping** という単一の実行可能ファイルが付属しています。

次の表は、対応する **XDPing** コマンドで実行できるさまざまなタスクについて説明しています。

タスク	XDPING コマンド	備考
コマンドラインオプションを表示するには	**sudo /opt/Citrix/VDA/bin/xdping -h**	該当なし
テストスイート全体を実行するには	**sudo /opt/Citrix/VDA/bin/xdping**（コマンドラインオプションなしで **XDPing** を実行）	Linux **XDPing** ツールは、システム上で150を超える個別のテストを実行します。詳細については、この記事の後半にある個別のテストを参照してください。
VDA登録ステータスチェックを実行するには	**sudo /opt/Citrix/VDA/bin/xdping -a**	詳細については、この記事の後半にある登録ステータスチェックの範囲を参照してください。
VDAの主要データをバックアップするには	**sudo /opt/Citrix/VDA/bin/xdping -b** コマンド	詳細については、この記事の後半にあるVDAデータのバックアップと比較を参照してください。
VDAバックアップデータの最新の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff** コマンド	詳細については、この記事の後半にあるVDAデータのバックアップと比較を参照してください。
VDAバックアップデータの特定の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff=<one backup data directory>:<the other backup data directory>**	詳細については、この記事の後半にあるVDAデータのバックアップと比較を参照してください。
Linux VDAパッケージをインストールする前に環境を確認するには	**sudo /opt/Citrix/VDA/bin/xdping --preflight**	該当なし
特定のテストカテゴリのみを実行する場合、たとえば、時間、Kerberos、およびデータベースのテスト	**sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos,database**	該当なし

タスク	XDPING コマンド	備考
特定のDelivery Controllerをプローブする場合	**sudo /opt/Citrix/VDA/bin/xdping -d <FQDN of the Delivery Controller>**	該当なし
VDAにインストールされている依存関係を表示する場合	**sudo /opt/Citrix/VDA/bin/xdping --query-pkgs** または **sudo /opt/Citrix/VDA/bin/xdping -q**	該当なし
VDAにインストールされている依存関係を表示し、クエリ結果を特定のパスに保存する場合	**sudo /opt/Citrix/VDA/bin/xdping -f <クエリ結果を保存する特定のパス>** または **sudo /opt/Citrix/VDA/bin/xdping --query-pkgs-to <クエリ結果を保存する特定のパス>**	該当なし

個別テスト

Linux **XDPing**ツールは、システム上で150を超える個別テストを実行します。これらは、大まかに次のように分類されます。

- Linux VDAのシステム要件が満たされているかを確認します。
- Linuxディストリビューションを含むマシン情報を特定して表示します。
- Linuxカーネルの互換性を確認します。
- Linux VDAの動作に影響を与える可能性のある既知のLinuxディストリビューションの問題を確認します。
- Security-Enhanced Linux (SELinux) のモードと互換性を確認します。
- ネットワークインターフェイスを特定し、ネットワーク設定を確認します。
- ストレージのパーティション分割と利用可能なディスク領域を確認します。
- マシンのホスト名とドメイン名の構成を確認します。
- DNS 構成を確認し、ルックアップテストを実行します。
- 基盤となるハイパーバイザーを特定し、仮想マシンの構成を確認します。サポート対象:
  - ゼノサーバー (旧 シトリックス ハイパーバイザー™)
  - マイクロソフト Hyper-V

    - ヴイエムウェア ブイスフィア

- 時刻設定を確認し、ネットワーク時刻同期が動作しているか確認します。
- PostgreSQL サービスが適切に構成され、動作しているか確認します。
- SQLite が適切に構成され、動作しているか確認します。
- ファイアウォールが有効になっており、必要なポートが開いているか確認します。
- Kerberos 構成を確認し、認証テストを実行します。
- グループポリシーサービスエンジン用の LDAP 検索環境を確認します。
- Active Directory 統合が適切に設定されており、現在のマシンがドメインに参加しているか確認します。サポート対象:
    - サンバ Winbind
    - デル クエスト 認証サービス
    - セントリファイ ダイレクトコントロール
    - SSSD
- Active Directory内のLinuxコンピューターオブジェクトの整合性を確認します。
- Pluggable Authentication Module (PAM) の構成を確認します。
- コアダンプパターンを確認します。
- Linux VDAに必要なパッケージがインストールされているか確認します。
- Linux VDAパッケージを特定し、インストールの整合性を確認します。
- PostgreSQLレジストリデータベースの整合性を確認します。
- Linux VDAサービスが適切に構成され、動作しているか確認します。
- VDAおよびHDX™構成の整合性を確認します。
- 各構成済みDelivery Controller™をプローブして、Broker Serviceが到達可能で、動作しており、応答性があることをテストします。
- マシンがDelivery Controllerファームに登録されているか確認します。
- 各アクティブまたは切断されたHDXセッションの状態を確認します。
- Linux VDA関連のエラーと警告についてログファイルをスキャンします。
- Xorgのバージョンが適切かどうかを確認します。
- 必要な依存関係がインストールされているかどうかを確認します。

出力例

以下は、Kerberosテストを実行した際の出力例です。

Kerberosテストの出力例の最初の部分(/en-us/linux-virtual-delivery-agent/2411/media/kerberos-test-sample-output-1.png)

Kerberosテストの出力例の2番目の部分(/en-us/linux-virtual-delivery-agent/2411/media/kerberos-test-sample-output-2.png)

Kerberosテストの出力例の3番目の部分(/en-us/linux-virtual-delivery-agent/2411/media/kerberos-test-sample-output-3.png)

VDA登録ステータスチェックの範囲

Linuxの**XDPing**ツールには、VDA登録ステータスを確認および分析するのに役立つ分析モジュールも用意されています。登録ステータスチェックの範囲については、次のスクリーンショットを参照してください。

登録ステータスチェック(/en-us/linux-virtual-delivery-agent/2411/media/registration-status-checks.png)

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ました。このモジュールを使用すると、構成、データベース、バイナリ権限データなど、VDAの主要データをいつでもバックアップできます。VDAが正常に動作しているときに、VDAの主要データをバックアップできます。後でVDAに障害が発生した場合に備えて、データの別のコピーをバックアップし、2つのデータコピーを比較してトラブルシューティングを容易にすることができます。次の表に、対応する**XDPing**コマンドを使用したVDAデータのバックアップと比較について説明します。

タスク	XDPINGコマンド	備考
VDAの主要データをバックアップするには	**sudo /opt/Citrix/VDA/bin/xdping -b**	バックアップコマンドを実行するたびに、バックアップデータのコピーが生成され、**/var/ctxbackup** の下のディレクトリに保存されます。バックアップデータディレクトリは、**yyyy-mm-dd-hh_mm_ss** 形式の現在の日時で命名されます（例: **2023-02-27-16_31_27**）。デフォルトでは、バックアップデータディレクトリの最大数

タスク	XDPINGコマンド	備考
		は30であり、この数を超えると**XDPing**ツールは古いバックアップデータディレクトリをローテーションまたは削除します。ディレクトリローテーションの数をカスタマイズするには、次のコマンドを実行します: `sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Backup" -t "REG_DWORD" -v "MaxDirRotationCount" -d "0x0000005" --force`
VDAバックアップデータの最新の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff**	該当なし
VDAバックアップデータの特定の2つのコピーを比較するには	**sudo /opt/Citrix/VDA/bin/xdping -diff=<one backup data directory>:<the other backup data directory>**	該当なし

セッションメトリック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
```

ctxqセッション, ctxqユーザー, ctxqフル, ctxqクエリ

これらのユーティリティの使用方法については、ヘルプコマンドを実行してください（例:

```
root@ubuntu2204-33:~# /opt/Citrix/VDA/bin/ctxqsession --help
Usage: ctxqsession [-f short_format_options | -o long_format_options] [-m] [user user_name]
Display information about ICA connections to the local server.

Options:
    -f          Change the displayed information specified by short format options
    -o          Change the displayed information specified by long format options
    -m          Alter the column headers to remove spaces
    user        Display information about the specified user

Short format options:
    d    Client device name (client name)
    i    Session ID
    n    Session name
    p    Published application name
    P    Protocols
    S    Session state
    u    User name
    x    X display number

Long format options(comma separated):
    app      Published application name
    dev      Client device name (client name)
    id       Session ID
    proto    Protocols
    sess     Session name
    state    Session state
    user     User name
    xdpy     X display number
```

ユーザーは自分のセッションメトリックのみをクエリできます。他のユーザーのセッションメトリックをクエリする権限を持つのは、rootユーザーと**ctxadm**グループのユーザーのみです。

ctxsdcユーティリティ

このユーティリティは、VDAでホストされているすべてのセッションまたは特定のセッションの以下のメトリックを照会するのに役立ちます。そのためには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]` コマンドを実行します。`[-c]` 引数は、1秒ごとにメトリックを照会することを意味します。

- **入力セッション帯域幅**
- **出力セッション帯域幅**
- **出力セッション回線速度**
- **遅延 - 最終記録**
- **ラウンドトリップタイム**
- **出力ThinWire帯域幅**
- **出力オーディオ帯域幅**
- **出力プリンター帯域幅**
- **入力ドライブ帯域幅**
- **出力ドライブ帯域幅**

その他

このセクションには、次のトピックが含まれています。

- HTML5 向け Citrix Workspace アプリのサポート
- Python3 仮想環境の作成
- NIS と Active Directory の統合
- IPv6
- LDAPS
- **Xオーソリティ**

Citrix Workspace™ app for HTML5 のサポート

Citrix Workspace app for HTML5 を使用すると、クライアントを Citrix Gateway に接続せずに、Linux 仮想アプリとデスクトップに直接アクセスできます。Citrix Workspace app for HTML5 の詳細については、Citrix ドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトで無効になっています。有効にするには、次の手順を実行します。

1. Citrix StoreFront™ で、Citrix Workspace app for HTML5 を有効にします。

   詳細な手順については、Knowledge Center の記事 CTX208163のステップ1を参照してください。

2. WebSocket 接続を有効にします。

   a. Citrix Studio で、**WebSockets connections** ポリシーを **Allowed** に設定します。

      他の WebSocket ポリシーも設定できます。WebSocket ポリシーの完全なリストについては、WebSockets ポリシー設定を参照してください。

   b. 設定を有効にするには、VDA で `ctxvda` サービスと `ctxhdx` サービスをこの順序で再起動します。

   c. VDA で、WebSocket リスナーが実行されているかどうかを確認するには、次のコマンドを実行します。

      ```
      netstat -an | grep 8008
      ```

      WebSocket リスナーが実行されている場合、コマンド出力は次のようになります。

      ```
      tcp 0 0 :::8008 :::* LISTEN
      ```

   注： WebSocket 接続を保護するために TLS 暗号化を有効にすることもできます。TLS 暗号化の有効化については、TLS を使用したユーザーセッションの保護を参照してください。

Python3仮想環境を作成する

ネットワークに接続している場合、`sudo /opt/Citrix/VDA/bin/xdping`コマンドを実行するとPython3仮想環境を作成できます。ただし、コマンドでPython3仮想環境を作成できない場合は、ネットワーク接続がなくても手動で作成できます。この記事では、ネットワーク接続なしでPython3仮想環境を作成するための前提条件と手順について詳しく説明します。

前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3`ディレクトリにアクセスするには、管理者権限が必要です。
- `Python3`パッケージのwheelファイルが配置されています。wheelファイルは`<https://pypi.org/>`からダウンロードできます。

Python3仮想環境を作成する

Python3仮想環境を作成するには、次の手順を実行します。

1. Python3の依存関係をインストールします。

   Amazon Linux 2の場合：

   ```
   yum -y install python3 python3-devel krb5-devel gcc
   ```

   RHELおよびRocky Linuxの場合：

   ```
   yum -y install python3-devel krb5-devel gcc
   ```

   > **注:**
   >
   > 一部の依存関係をインストールするには、特定のレポジトリを有効にする必要がある場合があります。RHEL 7の場合は`subscription-manager repos --enable rhel-7-server-optional-rpms`コマンドを実行します。RHEL 8の場合は`subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms`コマンドを実行します。

   Debian、Ubuntuの場合：

   ```
   apt-get -y install python3-dev python3-pip python3-venv libkrb5-dev
   libffi-dev
   ```

   SUSEの場合：

   ```
   zypper -n install lsb-release python3-devel python3-setuptools krb5-
   devel gcc libffi-devel libopenssl-devel
   ```

2. Python3仮想環境を作成します。

   > **注:**

pipツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼できるホストを/etc/pip.confファイルに追加することを検討してください。

\> `[global]`

`trusted-host =`

\> `pypi.org`

`files.pythonhosted.org`

Amazon Linux 2、Debian、RHEL、Rocky Linux、Ubuntu をご利用の場合：

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

SUSEの場合：

```
sudo ln -s /usr/lib/mit/bin/krb5-config /usr/bin/krb5-config


export PATH=$PATH:/usr/lib/mit/bin:/usr/lib/mit/sbin


sudo mkdir -p /usr/lib/mit/include/gssapi/


sudo ln -s /usr/include/gssapi/gssapi_ext.h/usr/lib/mit/include/gssapi/
gssapi_ext.h


sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

3. **XDPing**の依存関係をインストールします。

Debian 12の場合：

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
asn1crypto==1.5.1 cffi==1.17.1 cryptography==43.0.1 decorator==5.1.1
gssapi==1.8.3 ldap3==2.9.1 netifaces==0.11.0 packaging==24.1
pg8000==1.31.2 prettytable==3.11.0 psutil==6.0.0 pyasn1==0.6.1
pyparsing==3.1.4 python-dateutil==2.9.0.post0 scramp==1.4.5 six==1.16.0
termcolor==2.4.0 wcwidth==0.2.13



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

その他のLinuxディストリビューションの場合：

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
asn1crypto==1.5.1 cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1
gssapi==1.7.3 ldap3==2.9.1 netifaces==0.11.0 packaging==21.3
pg8000==1.26.0 psutil==5.9.0 pyasn1==0.4.8 pycparser==2.21
pyparsing==3.0.8 scramp==1.4.1 six==1.16.0 termcolor==1.1.0
prettytable==2.5.0



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

NIS と Active Directory の統合

この記事では、SSSD を使用して Linux VDA 上で NIS を Windows Active Directory (AD) と統合する方法について説明します。Linux VDA は Citrix Virtual Apps and Desktops のコンポーネントと見なされます。そのため、Windows AD 環境に緊密に適合します。

UID および GID プロバイダーとして AD の代わりに NIS を使用するには、アカウント情報（ユーザー名とパスワードの組み合わせ）が AD と NIS で同じである必要があります。

> **注:**
>
> 認証は引き続き AD サーバーによって実行されます。NIS+ はサポートされていません。NIS を UID および GID プロバイダーとして使用する場合、Windows サーバーからの POSIX 属性は使用されなくなります。

> **ヒント:**
>
> この方法は、Linux VDA を展開する非推奨の方法であり、特別なユースケースでのみ使用されます。RHEL/CentOS ディストリビューションの場合は、Amazon Linux 2、CentOS、RHEL、および Rocky Linux に Linux VDA を手動でインストールする の手順に従ってください。Ubuntu ディストリビューションの場合は、Ubuntu に Linux VDA を手動でインストールする の手順に従ってください。

> SSSD とは？

SSSD はシステムデーモンです。その主な機能は、システムにキャッシュとオフラインサポートを提供できる共通のフレームワークを通じて、リモートリソースの識別と認証へのアクセスを提供することです。PAM モジュールと NSS モジュールの両方を提供し、将来的には拡張ユーザー情報のための D-BUS ベースのインターフェイスをサポートできます。また、ローカルユーザーアカウントと拡張ユーザーデータを保存するためのより優れたデータベースも提供します。

NIS と AD の統合

NIS を AD と統合するには、次の手順を実行します。

ステップ 1：LINUX VDA を NIS クライアントとして追加する

NIS クライアントを構成します。

```
yum –y install ypbind rpcbind oddjob-mkhomedir
```

NIS ドメインを設定します。

```
ypdomainname nis.domain

echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

NIS サーバーとクライアントの IP アドレスを **/etc/hosts** に追加します。

`{NIS server IP address}  server.nis.domain nis.domain`

authconfig で NIS を構成します。

```
sudo authconfig --enablenis --nisdomain=nis.domain --
nisserver=server.nis.domain --enablemkhomedir --update
```

nis.domain は NIS サーバーのドメイン名を表します。**server.nis.domain** は NIS サーバーのホスト名であり、NIS サーバーの IP アドレスでもかまいません。

NISサービスを構成します。

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS設定が正しいことを確認します。

```
ypwhich
```

アカウント情報がNISサーバーから利用可能であることを検証します。

```
getent passwd nisaccount
```

注:

> The **nisaccount** は、NISサーバー上の実際のNISアカウントを表します。UID、GID、ホームディレクトリ、およびログインシェルが正しく構成されていることを確認してください。

ステップ2：SAMBAを使用してドメインに参加し、ホストキータブを作成する

SSSDは、ドメインへの参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。これらの機能を実現するには、いくつかの方法があります。

- `adcli`
- `realmd`
- `Winbind`
- `Samba`

このセクションの情報は、Sambaのアプローチのみを説明しています。`realmd`については、RHELまたはCentOSベンダーのドキュメントを参照してください。これらの手順は、SSSDを構成する前に実行する必要があります。

Sambaを使用してドメインに参加し、ホストキータブを作成する：

適切に構成されたファイルを持つLinuxクライアントで：

- /etc/krb5.conf
- /etc/samba/smb.conf:

SambaおよびKerberos認証用にマシンを構成します。

```
sudo authconfig --smbsecurity=ads --smbworkgroup=domain --smbrealm=REALM
--krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --update
```

ここで、**REALM** は大文字のKerberosレルム名であり、**domain** はドメインのNetBIOS名です。

KDCサーバーとレルム名のDNSベースのルックアップが必要な場合は、上記のコマンドに次の2つのオプションを追加します。

`--enablekrb5kdcdns --enablekrb5realmdns`

/etc/samba/smb.conf を開き、**authconfig** ツールによって生成されたセクションの後に、**\[Global\]** セクションの下に次のエントリを追加します。

`kerberos method = secrets and keytab`

`winbind offline logon = no`

Windowsドメインに参加するには、ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つADユーザーアカウントが必要です。

```
sudo net ads join REALM -U user
```

REALM は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

ステップ3：SSSDをセットアップする

> **注**
>
> SSSDをName Service Cache Daemon (NSCD) と一緒に使用すると、予期しない動作が発生する可能性があります。詳細については、「Using NSCD with SSSD」を参照してください。

> SSSDのセットアップは、次の手順で構成されます。

- Linuxクライアントマシンに **sssd-ad** および **sssd-proxy** パッケージをインストールします。
- さまざまなファイル（例：**sssd.conf**）に構成変更を加えます。
- **sssdサービス**を開始します。

/etc/sssd/sssd.conf
sssd.conf の構成例（必要に応じてオプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = EXAMPLE

services = nss, pam



[domain/EXAMPLE]

# Uncomment if you need offline logins
```

```
# cache_credentials = true

re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?
P<domain>.+$))|(^(?P<name>[^@\\]+)$))

id_provider = proxy

proxy_lib_name = nis

auth_provider = ad

access_provider = ad



# Should be specified as the long version of the Active Directory domain.

ad_domain = EXAMPLE.COM



# Kerberos settings

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U



# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



# Comment out if the users have the shell and home dir set on the AD side

default_shell = /bin/bash

fallback_homedir = /home/%d/%u
```

```
# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available

# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

ad.domain.com、**server.ad.example.com** を対応する値に置き換えます。詳細については、「**sssd-ad**(5) - Linux man page」を参照してください。

sssd.conf のファイル所有権と権限を設定します。

`chown root:root /etc/sssd/sssd.conf`

`chmod 0600 /etc/sssd/sssd.conf` `restorecon /etc/sssd/sssd.conf`

ステップ4：NSS/PAMを構成する

RHEL/CentOS:

authconfig を使用してSSSDを有効にします。ホームディレクトリの作成がSELinuxと互換性があることを確認するために **oddjob-mkhomedir** をインストールします。

```
authconfig --enablesssd --enablesssdauth --enablemkhomedir --update



sudo systemctl start sssd



sudo systemctl enable sssd
```

> **ヒント:**
>
> Linux VDA設定を構成する際、SSSDについてはLinux VDAクライアントに特別な設定がないことを考慮してください。**ctxsetup.sh**スクリプトの追加ソリューションには、デフォルト値を使用してください。

ステップ5：KERBEROS構成の確認

KerberosがLinux VDAで使用できるように正しく構成されていることを確認するには、システム**keytab**ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberos **kinit**コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐためにバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを確認するには、次を使用します。

```
sudo klist -ke
```

ステップ6：ユーザー認証の確認

getentコマンドを使用して、ログオン形式がサポートされているか、およびNSSが機能するかどうかを確認します。

```
sudo getent passwd DOMAIN\\username
```

DOMAINパラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず**getent**コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOSサフィックス形式: `username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、ドメインユーザーアカウントを使用してLinux VDAにログオンします。このドメインユーザーアカウントは、以前に使用されたことがないものとします。

```
sudo ssh localhost -l DOMAIN\\username



id -u
```

コマンドによって返された**uid**に対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_{uid}
```

ユーザーのKerberos認証情報キャッシュ内のチケットが有効で期限切れでないことを確認します。

```
klist
```

IPv6

Linux VDAは、Citrix Virtual Apps and Desktopsと連携するためにIPv6をサポートしています。この機能を使用する際は、次の点を考慮してください。

- デュアルスタック環境では、IPv6が明示的に有効になっていない限り、IPv4が使用されます。
- IPv4環境でIPv6が有効になっている場合、Linux VDAは機能しません。

重要:

- ネットワーク環境全体がIPv6である必要があり、Linux VDAだけでなく全体がIPv6である必要があります。
- Centrifyは純粋なIPv6をサポートしていません。

Linux VDAをインストールする際に、IPv6に関する特別なセットアップタスクは必要ありません。

Linux VDAのIPv6を構成する

Linux VDAの構成を変更する前に、Linux仮想マシンが以前にIPv6ネットワークで動作していたことを確認してください。IPv6構成に関連するレジストリキーは2つあります。

```
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration"

"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"ControllerRegistrationIPv6Netmask"
```

Linux VDAでIPv6を有効にするには、**OnlyUseIPv6ControllerRegistration**を1に設定する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

Linux VDAに複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask**を使用して、Linux VDAの登録に使用するインターフェイスを指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ControllerRegistrationIPv6Netmask " -d "{IPv6 netmask}" --force
```

{IPv6 netmask} を実際のネットマスク（例えば、2000::/64 のように）に置き換えます。

Citrix Virtual Apps and DesktopsでのIPv6展開の詳細については、IPv4/IPv6のサポートを参照してください。

トラブルシューティング

基本的なIPv6ネットワーク環境を確認し、ping6を使用してADおよびDelivery Controller™に到達可能かどうかを確認します。

LDAPS

LDAPSは、LDAP通信がTLS/SSLを使用して暗号化される、Lightweight Directory Access Protocol (LDAP) のセキュアなバージョンです。

デフォルトでは、クライアントとサーバーアプリケーション間のLDAP通信は暗号化されません。LDAPSを使用すると、Linux VDAとLDAPサーバー間のLDAPクエリコンテンツを保護できます。

以下のLinux VDAコンポーネントはLDAPSに依存しています。

- ブローカーエージェント：Delivery Controller™へのLinux VDA登録

- ポリシーサービス：ポリシー評価

LDAPSの構成には以下が含まれます。

- アクティブディレクトリ (AD)/LDAPサーバーでLDAPSを有効にする
- クライアントで使用するためにルートCAをエクスポートする
- Linux VDAでLDAPSを有効または無効にします。
- サードパーティプラットフォーム向けにLDAPSを構成する
- SSSDを構成する
- Winbindを構成する
- Centrifyを構成する
- Questを構成する

注:

LDAPサーバーの監視サイクルを設定するには、次のコマンドを実行します。デフォルト値は15分です。少なくとも10分に設定してください。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --
force
```

AD/LDAPサーバーでLDAPSを有効にする

Microsoft 認証局 (CA) または Microsoft 以外の CA のいずれかから、適切な形式の証明書をインストールすることで、LDAP over SSL (LDAPS) を有効にできます。

ヒント:

ドメインコントローラーにエンタープライズルート CA をインストールすると、LDAPS は自動的に有効になります。

証明書のインストール方法と LDAPS 接続の確認方法の詳細については、サードパーティの認証局で LDAP over SSL を有効にする方法を参照してください。

多層認証局階層を使用している場合、ドメインコントローラーで LDAPS 認証用の適切な証明書が自動的に利用できるわけではありません。

多層認証局階層を使用してドメインコントローラーで LDAPS を有効にする方法については、LDAP over SSL (LDAPS) 証明書の記事を参照してください。

クライアントが使用できるようにルート認証局を有効にする

クライアントは、LDAP サーバーが信頼する CA の証明書を使用する必要があります。クライアントの LDAPS 認証を有効にするには、ルート CA 証明書を信頼されたキーストアにインポートします。

ルート CA のエクスポート方法の詳細については、Microsoft サポート Web サイトのルート認証局証明書をエクスポートする方法を参照してください。

Linux VDA で LDAPS を有効または無効にする

Linux VDA で LDAPS を有効または無効にするには、(管理者としてログオンした状態で) **enable_ldaps.sh** スクリプトを実行します。**enable_ldaps.sh** スクリプトを非対話モードで実行するには、次の変数を環境にエクスポートします。

```
#CTX_LDAPS_KEYSTORE_PASSWORD=



#CTX_LDAPS_LDAP_SERVERS=
```

- 提供されたルート CA 証明書を使用して LDAP over SSL/TLS を有効にするには (LDAP チャネルバインディングがサポートされています):

  ```
  /opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
  ```

- SSL/TLS なしの LDAP にフォールバックする

  ```
  /opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
  ```

LDAPS 専用の Java キーストアは **/etc/xdl/.keystore** にあります。影響を受けるレジストリキーは次のとおりです。

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers


HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy


HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS


HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore


HKLM\Software\Citrix\VirtualDesktopAgent\EnableChannelBinding
```

サードパーティプラットフォーム向けの LDAPS を構成する

Linux VDAコンポーネントのほかに、VDAに準拠するサードパーティ製ソフトウェアコンポーネント（SSSD、Winbind、Centrify、Questなど）もセキュアLDAPを必要とする場合があります。以下のセクションでは、LDAPS、STARTTLS、またはSASL署名およびシーリングを使用してセキュアLDAPを構成する方法について説明します。

ヒント:

これらのソフトウェアコンポーネントのすべてが、セキュアLDAPを確保するためにSSLポート636を使用することを推奨している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でのSTARTTLSと共存できません。

SSSD

オプションに従って、ポート636またはポート389でSSSDセキュアLDAPトラフィックを構成します。詳しくは、SSSD LDAP Linux manページを参照してください。

ウィンバインド

Winbind LDAPクエリはADSメソッドを使用します。Winbindはポート389でのStartTLSメソッドのみをサポートします。影響を受ける構成ファイルは、**/etc/samba/smb.conf** および **/etc/**

openldap/ldap.conf (Amazon Linux 2、RHEL、Rocky Linux、CentOS、SUSEの場合)、または **/etc/ldap/ldap.conf** (DebianおよびUbuntuの場合) です。ファイルを次のように変更します。

- smb.conf

  `ldap ssl = start tls`

  `ldap ssl ads = yes` `client ldap sasl wrapping = plain`

- ldap.conf

  `TLS_REQCERT never`

または、SASL GSSAPI署名およびシーリングによってセキュアLDAPを構成することもできますが、TLS/SSLとは共存できません。SASL暗号化を使用するには、**smb.conf**構成を変更します。

`ldap ssl = off`

`ldap ssl ads = no` `client ldap sasl wrapping = seal`

セントリファイ

Centrifyはポート636でのLDAPSをサポートしていません。ただし、ポート389でセキュアな暗号化を提供します。詳しくは、Centrifyサイトを参照してください。

クエスト

Quest Authentication Serviceはポート636でのLDAPSをサポートしていませんが、異なるメソッドを使用してポート389でセキュアな暗号化を提供します。

トラブルシューティング

この機能を使用すると、次の問題が発生する可能性があります。

- **LDAPSサービスの可用性**

  AD/LDAPサーバーでLDAPS接続が利用可能であることを確認します。ポートはデフォルトで636です。

- **LDAPSが有効な場合にLinux VDAの登録が失敗しました**

  LDAPサーバーとポートが正しく構成されていることを確認します。まずルートCA証明書を確認し、AD/LDAPサーバーと一致していることを確認します。

- **誤ってレジストリを変更してしまった場合**

  enable_ldaps.shを使用せずに誤ってLDAPS関連のキーを更新した場合、LDAPSコンポーネントの依存関係が壊れる可能性があります。

- **Wiresharkまたはその他のネットワーク監視ツールからSSL/TLSで暗号化されていないLDAPトラフィック**

  デフォルトでは、LDAPSは無効になっています。強制的に有効にするには、**/opt/Citrix/VDA/sbin/enable_ldaps.sh**を実行します。

- **Wiresharkまたはその他のネットワーク監視ツールからLDAPSトラフィックがない**

  LDAP/LDAPSトラフィックは、Linux VDAの登録とグループポリシーの評価が行われるときに発生します。

- **ADサーバーでldp connectを実行してLDAPSの可用性を確認できませんでした**

  IPアドレスの代わりにAD FQDNを使用してください。

- **/opt/Citrix/VDA/sbin/enable_ldaps.shスクリプトを実行してルートCA証明書をインポートできませんでした**

  CA証明書のフルパスを指定し、ルートCA証明書が正しいタイプであることを確認します。これは、Java Keytoolでサポートされているほとんどのタイプと互換性があるはずです。サポートリストにない場合は、まずタイプを変換できます。証明書形式の問題が発生した場合は、base64エンコードされたPEM形式をお勧めします。

- **Keytool -listでルートCA証明書を表示できませんでした**

  `/opt/Citrix/VDA/sbin/enable_ldaps.sh`を実行してLDAPSを有効にすると、証明書は/etc/xdl/.keystoreにインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

Xオーソリティ

Linux VDAは、インタラクティブなリモート処理のためにX11ディスプレイ機能（`xterm`や`gvim`を含む）を使用する環境をサポートしています。この機能は、XClientとXServer間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信の許可を保護する方法は2つあります。

- **Xhost**。デフォルトでは、XhostはlocalhostのXClientのみがXServerと通信することを許可します。リモートのXClientがXServerにアクセスすることを許可する場合、特定のマシンでXhostコマンドを実行して許可を与える必要があります。または、**xhost +** を使用して、任意のXClientがXServerに接続できるようにすることもできます。
- **Xauthority**。`.Xauthority`ファイルは各ユーザーのホームディレクトリにあります。これは、XServerの認証のためにxauthが使用するCookieに資格情報を保存するために使用されま

す。XServerインスタンス（Xorg）が起動すると、その特定のディスプレイへの接続を認証するためにCookieが使用されます。

仕組み

Xorgが起動すると、`.Xauthority`ファイルがXorgに渡されます。この`.Xauthority`ファイルには次の要素が含まれています。

- ディスプレイ番号
- リモート要求プロトコル
- Cookie番号

`xauth`コマンドを使用してこのファイルを参照できます。例：

```
# xauth -f ~/.Xauthority



# > list



# > us01msip06:107  MIT-MAGIC-COOKIE-1  fb228d1b695729242616c5908f11624b
```

XClientがXorgにリモートで接続する場合、2つの前提条件を満たす必要があります。

- **DISPLAY**環境変数をリモートXServerに設定します。
- Xorg内のCookie番号の1つを含む`.Xauthority`ファイルを取得します。

Xauthorityの構成

リモートX11ディスプレイ用にLinux VDAで**Xauthority**を有効にするには、次の2つのレジストリキーを作成する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"XauthEnabled" -d "0x00000001" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
```

```
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"ListenTCP" -d "0x00000001" --force
```

Xauthority を有効にした後、`.Xauthority` ファイルを **XClient** に手動で渡すか、共有ホームディレクトリをマウントして渡します。

- `.Xauthority` ファイルを XClient に手動で渡す

  ICA® セッションの起動後、Linux VDA は XClient 用の `.Xauthority` ファイルを生成し、ログオンユーザーのホームディレクトリに保存します。この `.Xauthority` ファイルをリモートの XClient マシンにコピーし、**DISPLAY** および **XAUTHORITY** 環境変数を設定できます。`DISPLAY` は `.Xauthority` ファイルに保存されているディスプレイ番号で、`XAUTHORITY` は **Xauthority** のファイルパスです。例については、次のコマンドを参照してください。

  ```
  export DISPLAY={Display number stored in the Xauthority file}



  export XAUTHORITY={the file path of .Xauthority}
  ```

  注:

  XAUTHORITY 環境変数が設定されていない場合、デフォルトで `~/.Xauthority` ファイルが使用されます。

- 共有ホームディレクトリをマウントして `.Xauthority` ファイルを XClient に渡す

  便利な方法は、ログオンユーザー用に共有ホームディレクトリをマウントすることです。Linux VDA が ICA セッションを開始すると、`.Xauthority` ファイルがログオンユーザーのホームディレクトリに作成されます。このホームディレクトリが XClient と共有されている場合、ユーザーはこの `.Xauthority` ファイルを XClient に手動で転送する必要はありません。**DISPLAY** および **XAUTHORITY** 環境変数が正しく設定されると、GUI は XServer デスクトップに自動的に表示されます。

トラブルシューティング

Xauthority が機能しない場合は、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 権限を持つ管理者として、すべての Xorg クッキーを取得します。

```
ps aux | grep -i xorg
```

このコマンドは、Xorg プロセスと、起動時に Xorg に渡されたパラメーターを表示します。別のパラメーターには、どの `.Xauthority` ファイルが使用されているかが表示されます。
例：

```
/var/xdl/xauth/.Xauthority110
```

Xauth コマンドを使用してクッキーを表示します。

```
Xauth -f /var/xdl/xauth/.Xauthority110
```

2. `Xauth` コマンドを使用して、`~/.Xauthority` に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーは Xorg と XClient の `.Xauthority` ファイルで同じである必要があります。
3. クッキーが同じ場合は、Linux VDA の IP アドレスと公開デスクトップのディスプレイ番号を使用して、リモートディスプレイポートのアクセシビリティを確認します。

   たとえば、XClient マシンで次のコマンドを実行します。

   ```
   telnet 10.158.11.11 6160
   ```

   ポート番号は 6000 + \<ディスプレイ番号\> の合計です。

   この telnet 操作が失敗した場合、ファイアウォールが要求をブロックしている可能性があります。

認証

このセクションには、次のトピックが含まれています。

- Azure Active Directory を使用した認証(/ja-jp/linux-virtual-delivery-agent/2411/configure/authentication/aad-authentication.html)
- ダブルホップシングルサインオン認証(/ja-jp/linux-virtual-delivery-agent/2411/configure/authentication/single-sign-on.html)
- フェデレーション認証サービス

- 非SSO認証(/ja-jp/linux-virtual-delivery-agent/2411/configure/authentication/non-sso-authentication.html)
- スマートカード(/ja-jp/linux-virtual-delivery-agent/2411/configure/authentication/support-for-smart-cards.html)
- 未認証（匿名）ユーザーによるアクセス(/ja-jp/linux-virtual-delivery-agent/2411/configure/authentication/access-by-unauthenticated-users.html)

Azure Active Directory を使用した認証

> **注:**
>
> この機能は、Azure でホストされている VDA でのみ利用できます。

> 必要に応じて、Azure に 2 種類の Linux VDA を展開できます。

- Azure AD DS に参加している VM。VM は Azure Active Directory (AAD) Domain Services (DS) マネージド ドメインに参加しています。ユーザーはドメイン資格情報を使用して VM にログオンします。
- ドメインに参加していない VM。VM は AAD ID サービスと統合され、ユーザー認証を提供します。ユーザーは AAD 資格情報を使用して VM にログオンします。

AAD DS と AAD の詳細については、この Microsoft の記事 を参照してください。

この記事では、ドメインに参加していない VDA で AAD ID サービスを有効にして構成する方法について説明します。

サポートされているディストリビューション

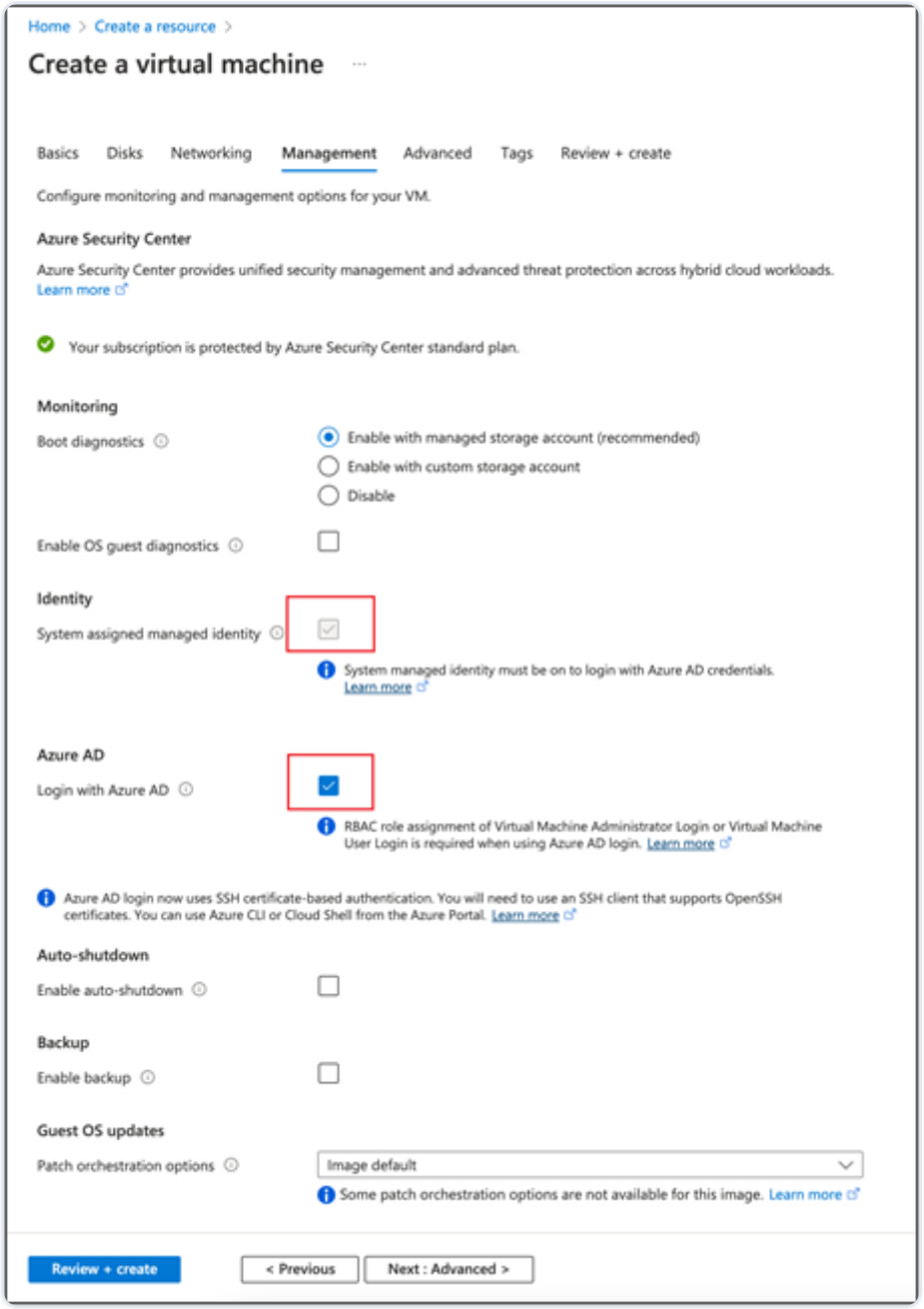
- ウブントゥ 24.04, 22.04, 20.04
- RHEL 8.10, 8.8
- SUSE 15.6

詳細については、このMicrosoftの記事を参照してください。

ステップ 1: Azure ポータルでテンプレート VM を作成する

テンプレート VM を作成し、VM に Azure CLI をインストールします。

1. Azure ポータルでテンプレート VM を作成します。「**Review + create**」をクリックする前に、「**Management**」タブで「**Login with Azure AD**」を必ず選択してください。



2. テンプレートVMにAzure CLIをインストールします。 詳細については、このMicrosoftの記事を参照してください。

ステップ2：テンプレートVMでマスターイメージを準備する

マスターイメージを準備するには、MCSを使用してAzureにLinux VMを作成するの**ステップ3：マスターイメージを準備する**に従ってください。

ステップ3：テンプレートVMをドメイン非参加モードに設定する

マスターイメージを作成したら、以下の手順に従ってVMをドメイン非参加モードに設定します。

1. コマンドプロンプトから次のスクリプトを実行します。

```
Modify /var/xdl/mcs/mcs_util.sh
```

2. `function read_non_domain_joined_info()`を見つけて、`NonDomainJoined`の値を`2`に変更します。例については、以下のコードブロックを参照してください。

```
function read_non_domain_joined_info()

{

log "Debug: Enter read_non_domain_joined_info"

# check if websocket enabled

TrustIdentity=`cat ${id_disk_mnt_point}${ad_info_file_path} | grep '\
[TrustIdentity\]' | sed 's/\s//g'`

if [ "$TrustIdentity" == "[TrustIdentity]" ]; then

NonDomainJoined=2

fi

...

}
```

3. 変更を保存します。
4. テンプレートVMをシャットダウンします。

ステップ4：テンプレートVMからLinux VMを作成する

ドメイン非参加のテンプレートVMの準備ができたら、以下の手順に従ってVMを作成します。

1. Citrix Cloudにサインインします。
2. Citrix DaaS™をダブルクリックし、Full Configuration管理コンソールにアクセスします。
3. **マシンカタログ**で、Machine Creation Servicesを使用してテンプレートVMからLinux VMを作成することを選択します。詳細については、Citrix DaaSの記事ドメイン非参加を参照してください。

ステップ5：AADユーザーアカウントをLinux VMに割り当てる

ドメイン非参加のVMを作成したら、AADユーザーアカウントをそれらに割り当てます。

VMにAADユーザーアカウントを割り当てるには、次の手順に従います。

1. 管理者アカウントを使用してVMにアクセスします。
2. **>** タブで、 を有効にします。

   AADでログインを有効にする設定(/en-us/linux-virtual-delivery-agent/2411/media/aad-auth-identity-setting.png)
3. **>** タブで、 領域を見つけ、必要に応じてAADユーザーアカウントを追加します。

   AADでのログインを有効にする(/en-us/linux-virtual-delivery-agent/2411/media/aad-auth-assign-aad-account.png)

ドメインに参加していないVDAへのログオン

組織のエンドユーザーは、ドメインに参加していないVDAに2つの方法でログオンできます。詳細な手順は次のとおりです。

1. Workspaceアプリを起動し、AADユーザー名とパスワードを入力してワークスペースにログオンします。Workspaceページが表示されます。
2. ドメインに参加していないデスクトップをダブルクリックします。AADログインページが表示されます。

   ページは、VDA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。管理者として、必要に応じてログインモードをAADアカウント/パスワードに変更できます。詳細な手順については、次のセクションを参照してください。

   AADでのログインを有効にする(/en-us/linux-virtual-delivery-agent/2411/media/aad-auth-device-code.png)
3. 画面の指示に従って、次のいずれかの方法でデスクトップセッションにログオンします。

   - QRコードをスキャンしてコードを入力します。
   - AADユーザー名とパスワードを入力します。

AAD アカウント/パスワード ログインモードに変更する

デフォルトでは、Linux VDA はデバイスコードを使用して AAD ユーザーを認証します。詳細については、このMicrosoft の記事を参照してください。ログインモードを *AAD アカウント/パスワード* に変更するには、次の手順に従います。

VDA で次のコマンドを実行し、`AADAcctPwdAuthEnable`キーを見つけ、その値を`0x00000001`に変更します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Services\CitrixBrokerAgent\WebSocket" -t
"REG_DWORD" -v "AADAcctPwdAuthEnable" -d "0x00000001" --force
```

注:

このアプローチは、Microsoft アカウントや二要素認証が有効になっているアカウントでは機能しません。

二重ホップシングルサインオン認証

StoreFrontストアにアクセスするためのユーザー資格情報は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10のAuthManagerモジュールに挿入できます。挿入後、Linux仮想デスクトップセッション内からクライアントを使用して仮想デスクトップおよびアプリケーションにアクセスでき、ユーザー資格情報を再度入力する必要はありません。

注:

この機能は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10でサポートされています。

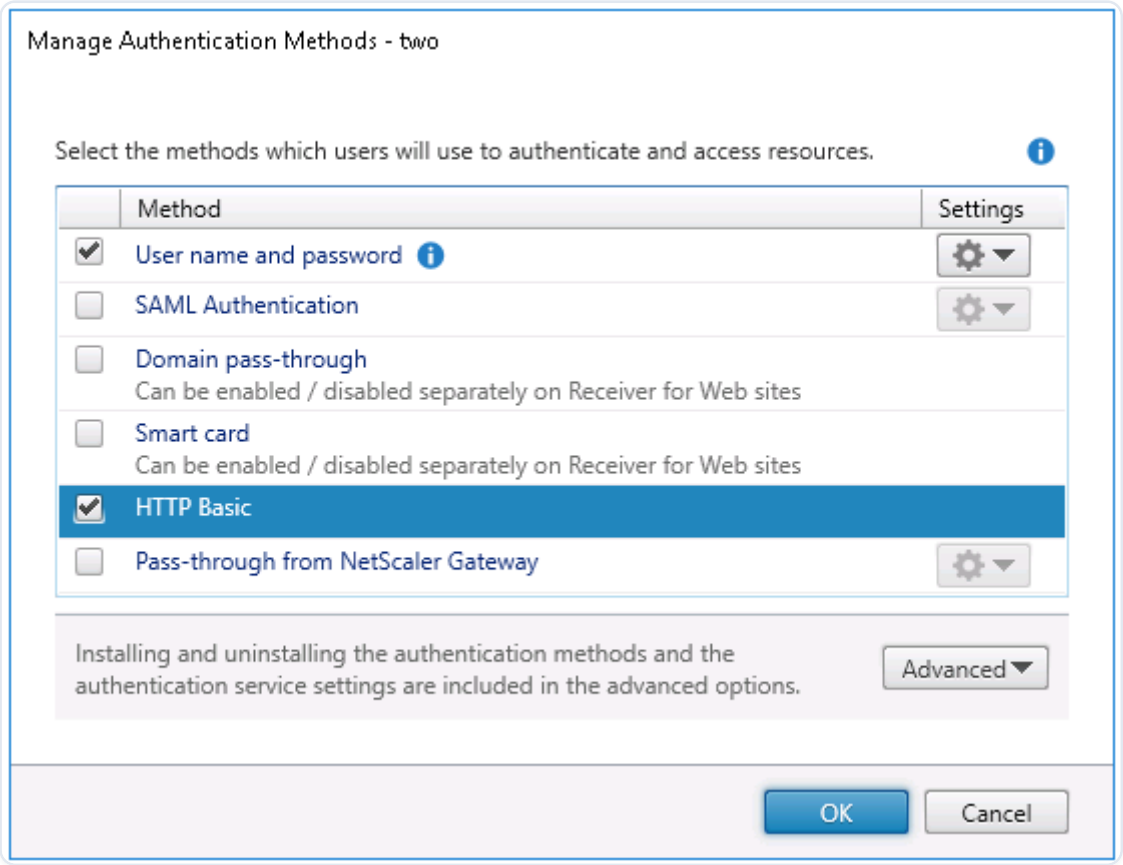
この機能を有効にするには：

1. Linux VDAに、Linux向けCitrix WorkspaceアプリまたはLinux向けCitrix Receiver 13.10をインストールします。

   Citrix WorkspaceアプリまたはCitrix ReceiverのCitrixダウンロードページからアプリをダウンロードします。

デフォルトのインストールパスは/opt/Citrix/ICAClient/です。アプリを別のパスにインストールする場合は、ICAROOT環境変数を実際のインストールパスを指すように設定します。

2. Citrix StoreFront™管理コンソールで、ターゲットストアに**HTTP Basic**認証方法を追加します。



3. HTTP Basic認証を許可するために、AuthManager構成ファイル（$ICAROOT/config/AuthManConfig.xml）に次のキーを追加します。

```
<Protocols>

     <HTTPBasic>

         <Enabled>True</Enabled>

    </HTTPBasic>

</Protocols>
```

4. 指定されたディレクトリにルート証明書をインストールするには、次のコマンドを実行します。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. この機能を有効にするには、次のコマンドを実行します。

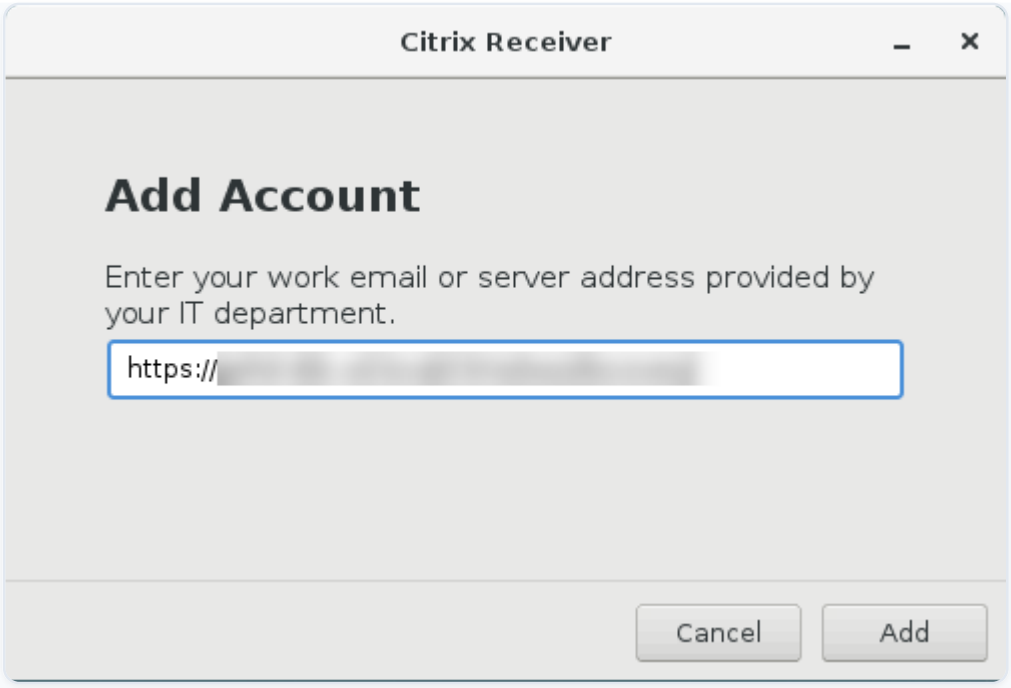
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

6. Linux仮想デスクトップセッションを起動し、そのセッション内でLinux向けCitrix Workspaceアプリまたは Linux向けCitrix Receiver 13.10を起動します。

   Citrix Workspace™アプリを初めて起動すると、ストアアカウントの入力を求められます。その後は、以前に指定したストアに自動的にログオンされます。

注:

ストアアカウントとしてHTTPS URLを入力します。



フェデレーション認証サービス

Federated Authentication Service (FAS) を使用して、Linux VDA にログオンするユーザーを認証できます。Linux VDA は、FAS ログオン機能に Windows VDA と同じ Windows 環境を使用します。FAS の Windows 環境の構成については、「Federated Authentication Service」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、**インセッション動作**ポリシーをサポートしていません。
- Linux VDA は、FAS サーバーとのデータ送信に短い接続を使用します。

サポートされているディストリビューション

FAS は、限られた Linux ディストリビューションとドメイン参加方法をサポートしています。以下のマトリックスを参照してください。

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
アマゾン リナックス 2	はい	はい	はい	はい	いいえ
デビアン 12.7/12.5/11.11	はい	はい	はい	はい	いいえ
RHEL 9.4/9.2	はい	はい	いいえ	いいえ	はい
RHEL 8.10/8.8	はい	はい	はい	はい	はい
ロッキー Linux 9.4/9.2	はい	はい	いいえ	いいえ	いいえ
ロッキー Linux 8.10/8.8	はい	はい	いいえ	いいえ	いいえ

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
SUSE 15.6	はい	はい	はい	いいえ	いいえ
ウブンツ 24.04	はい	はい	いいえ	いいえ	いいえ
ウブンツ 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ために、ルートCA証明書とすべての中間証明書をVDAにインストールします。たとえば、ルートCA証明書をインストールするには、前の「**Microsoft CA (AD上) からCA証明書を取得する**」手順からADルート証明書を取得するか、ルートCAサーバー `http://CA-SERVER/certsrv` からダウンロードします。

> **注:**
>
> 次のコマンドは、中間証明書の設定にも適用されます。
>
> たとえば、DERファイル (.crt、.cer、.der) をPEMに変換するには、次のようなコマンドを実行します。

```
sudo openssl x509 -inform der -in root.cer -out root.pem
```

次に、次のようなコマンドを実行して、ルートCA証明書を `openssl` ディレクトリにインストールします。

```
sudo cp root.pem /etc/pki/CA/certs/
```

> **注:**

> ルートCA証明書を **/root** パスに配置しないでください。そうしないと、FASはルートCA証明書に対する **読み取り** 権限を持ちません。

ctxfascfg.shの実行

FASを構成するには、ctxfascfg.shスクリプトを実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxfascfg.sh
```

`ctxfascfg.sh` をサイレントモードで実行できます。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の統合方法を示します。`CTX_EASYINSTALL_ADINTEGRATIONWAY` が指定されている場合、これは `CTX_EASYINSTALL_ADINTEGRATIONWAY` と等しくなります。`CTX_EASYINSTALL_ADINTEGRATIONWAY` が指定されていない場合、`CTX_FAS_ADINTEGRATIONWAY` は独自の値を設定します。
- **CTX_FAS_CERT_PATH \=\<certificate path\>**: ルート証明書とすべての中間証明書を格納する完全なパスを指定します。
- **CTX_FAS_KDC_HOSTNAME**: PBISとQuestを選択した場合のKey Distribution Center（KDC）のホスト名を指定します。
- **CTX_FAS_PKINIT_KDC_HOSTNAME**: PKINIT KDCホスト名を指定します。特に指定がない限り、CTX_FAS_KDC_HOSTNAMEと同じです。複数のDelivery Controllerがある場合は、ドメインのすべてのKDCのホスト名を **/etc/krb5.conf** ファイルのpkinit_kdc_hostnameに追加します。詳しくは、Knowledge Centerの記事CTX322129を参照してください。
- **CTX_FAS_SERVER_LIST='list-fas-servers'** – Federated Authentication Service（FAS）サーバーは、ADグループポリシーを通じて構成されます。ドメインGPOでのFASポリシー設定については、「グループポリシーの構成」を参照してください。Linux VDAはADグループポリシーをサポートしていませんが、代わりにFASサーバーのセミコロン区切りのリストを提供できます。シーケンスは、ADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白をテキスト文字列 **'\<none\>'** で埋め、サーバーアドレスの順序を変更しないでください。FASサーバーと適切に通信するには、FASサーバーで指定されたポート番号と一致するポート番号を追加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。

既存のLinux VDAインストールをアップグレードするには、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します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
Service" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force



systemctl restart ctxjproxy



systemctl restart ctxvda
```

`ctxreg`を介してFASサーバーを更新するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredential
Service" -v "Addresses" -d "<Your-FAS-Server-List>"



systemctl restart ctxjproxy



systemctl restart ctxvda
```

正しいActive Directory統合方法を選択し、証明書の正しいパス（例：`/etc/pki/CA/certs/`）を入力します。

スクリプトはkrb5-pkinitおよびpam_krb5パッケージをインストールし、関連する構成ファイルを設定します。RHEL 8以降では、PAM_KRB5がEPELリポジトリに移動されたため、スクリプトはこれらのディストリビューションでEPELを有効にしようとします。

FASを無効にする

Linux VDAでFASを無効にするには、次のコマンドを使用してConfDBからすべてのFASサーバーを削除します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "" --force



systemctl restart ctxjproxy



systemctl restart ctxvda
```

制限事項

- FASはまだロック画面をサポートしていません。セッションでロックボタンをクリックすると、FASを使用してセッションに再度ログオンすることはできません。
- このリリースは、Federated Authentication Serviceのアーキテクチャ概要の記事にまとめられている一般的なFAS展開のみをサポートしており、**Windows 10 Azure AD Join**は含まれていません。

トラブルシューティング

FASのトラブルシューティングを行う前に、以下を確認してください。

- Linux VDAが正しくインストールおよび構成されていること。
- パスワード認証を使用して、共通ストアで非FASセッションが正常に起動できること。

FAS以外のセッションが正常に動作する場合は、**Login**クラスのHDXログレベルをVERBOSEに、VDAログレベルをTRACEに設定します。Linux VDAのトレースログを有効にする方法については、Knowledge Centerの記事CTX220130を参照してください。

Linux **XDPing**ツールを使用して、Linux VDA環境に存在する可能性のある一般的な構成の問題を確認することもできます。

FASサーバー構成エラー

FASストアからのセッション起動に失敗します。

/var/log/xdl/hdx.logを確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [Logon
Type] Federated Authentication Logon.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: entry



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
connect to server 0



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0: failed
to connect: Connection refused.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: failed
to connect to server [0], please confirm if fas service list is well
configurated in condb



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit,
43



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: failed
to validate fas credential



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate:
failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER
```

解決策

次のコマンドを実行して、Citrixレジストリ値「HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\VirtualDesktopAgent\\Authentication\\UserCredentialService」が\<Your-FAS-Server-List\>に設定されていることを確認します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

既存の設定が正しくない場合は、前のFASサーバーの設定の手順に従って再度設定してください。

CA証明書の構成が正しくありません

FASストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。



/var/log/xdl/hdx.logを確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: entry



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller: current
process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_public_certificate: entry



2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
for response...
```

```
2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query to
server success


2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
get_public_certificate: exit


2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
input size 1888


2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
output size 1415


2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: get logon certificate success


2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
cache certificate success


2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: exit, 0


2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策

`/etc/krb5.conf` で、ルートCA証明書とすべての中間証明書が保存されてい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。

```
 [realms]

EXAMPLE.COM = {

    ......

    pkinit_anchors = DIR:/etc/pki/CA/certs/

    ......

}
```

既存の設定が正しくない場合は、前の証明書のインストールの手順に従って再度設定してください。

または、ルートCA証明書が有効であるかどうかを確認します。

シャドウアカウントマッピングエラー

FASはSAML認証によって構成されます。ADFSユーザーがADFSログオンページでユーザー名とパスワードを入力した後、次のエラーが発生する可能性があります。

シャドウアカウントマッピングエラー(/en-us/linux-virtual-delivery-agent/2411/media/shadow-account-mapping-error.png)

このエラーは、次のことを示しています。

- ADFSユーザーは正常に検証されましたが、ADにシャドウユーザーが構成されていません。

解決策

ADでシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に次のエラーが発生します。

ADFSが構成されていません(/en-us/linux-virtual-delivery-agent/2411/media/adfs-not-configured.png)

FASストアをSAML認証を使用するように構成しているにもかかわらず、ADFS展開が不足している場合にこの問題が発生します。

解決策

フェデレーション認証サービス用にADFS IdPを展開します。詳しくは、「フェデレーション認証サービスのADFS展開」を参照してください。

関連情報

- 一般的なFAS展開については、「フェデレーション認証サービスのアーキテクチャの概要」の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービスの高度な構成」の章で紹介されています。

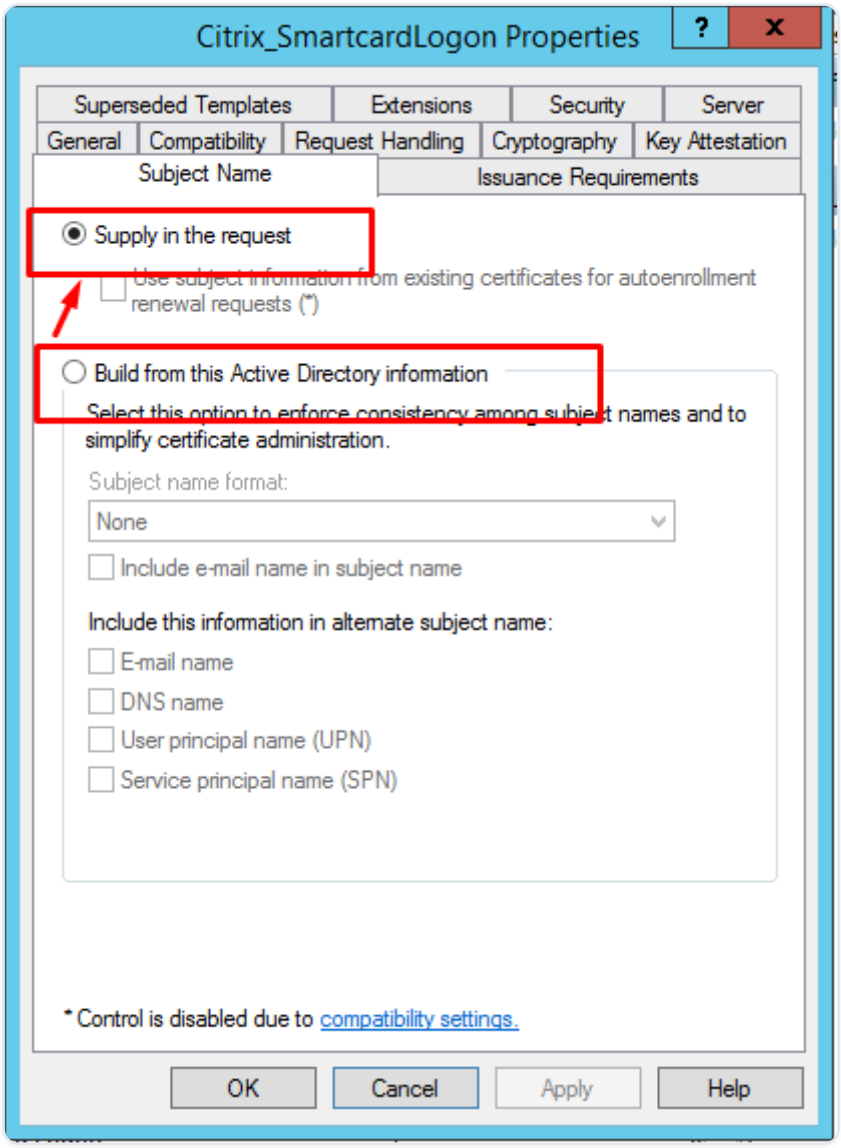
既知の問題

FASを使用している場合、英語以外の文字を含む公開デスクトップまたはアプリセッションを起動しようとすると、失敗する可能性があります。

英語以外の文字を含むセッションの起動に失敗する(/en-us/linux-virtual-delivery-agent/2411/media/launching-nonenglish-session-fails.png)

回避策

CAツールで**Manage Templates**を右クリックし、**Citrix_SmartcardLogon**テンプレートを**Build from this Active Directory information**から**Supply in the request**に変更します。



FIDO2 (プレビュー)

Linux VDAでホストされているGoogle Chromeを使用してWebサイトにアクセスするために、FIDO2認証を設定できます。

> **注:**
>
> Linux VDAは、FIDO2とGoogle Chromeの組み合わせのみをサポートします。

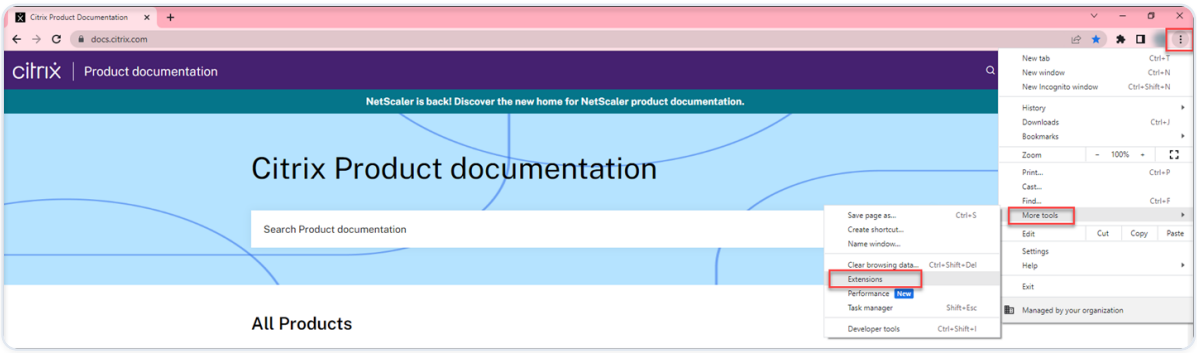
FIDO2認証を設定するには、次の手順を実行します。

1. Citrix FIDO2拡張機能パッケージをダウンロードします。
   a. Citrix Virtual Apps and Desktopsダウンロードページに移動します。

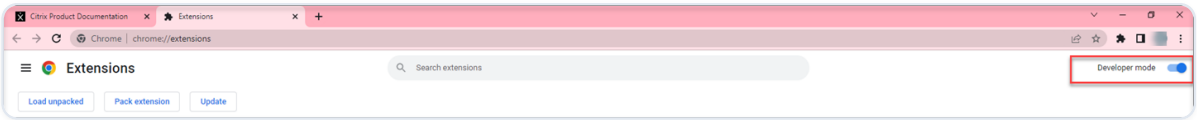
b. Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。
c. Linux VDAを見つけるには、**Components**をクリックします。
d. Linux VDAをクリックして、そのダウンロードページを開きます。
e. ソースパッケージをダウンロードします。
f. ソースパッケージを解凍して、**FIDO2-JavaScript-Extensions.zip**を見つけます。
g. FIDO2拡張機能パッケージを解凍します。FIDO2拡張機能ディレクトリは、**extensions > chrome > fido2**にあります。

2. Google ChromeでCitrix FIDO2拡張機能を追加します。

a. Linux VDAでホストされているGoogle Chromeを開きます。
b. アドレスバーの右側にある3点メニューをクリックし、次に、**その他のツール > 拡張機能**を選択します。



c. 「**開発者モード**」をオンに切り替えます。



d. 「**パッケージ化されていない拡張機能を読み込む**」をクリックし、「**extensions > chrome > fido2**」にある拡張機能ディレクトリを選択します。

3. FIDO2認証を使用したいWebサイトで、FIDO2認証を使用するための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、FIDO2を認証方法として追加します。

FIDO2認証が設定されると、Webサイトに正常にアクセスするためにセキュリティキーに触れるよう求められます。

非SSO認証

この記事では、Linux VDAで非SSO認証を有効にする方法について説明します。

概要

デフォルトでは、Linux VDAではシングルサインオン（SSO）が有効になっています。ユーザーは、Citrix Workspace™アプリとVDAセッションに、1組の資格情報を使用してログオンします。

ユーザーが異なる資格情報を使用してVDAセッションにログオンするようにするには、Linux VDAでSSOを無効にします。次の表に、非SSOシナリオでサポートされるユーザー認証方法の組み合わせを示します。

CITRIX WORKSPACEアプリ	VDAセッション
ユーザー名	ユーザー名
スマートカード	ユーザー名
ユーザー名	スマートカード
FAS	ユーザー名
FAS	スマートカード

SSOを無効にする

Linux VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t
"REG_DWORD" -v "fPromptForDifferentUser" -d "0x00000001" --force
```

制限事項

非SSOシナリオでは、Linux VDAセッション内のアプリケーションウィンドウが「常に手前に表示」に設定されている場合、切断後に再接続した際に、ユーザーが資格情報を入力して認証することを妨げる可能性があります。

スマートカード

クライアントデバイスに接続されたスマートカードを使用して、Linux仮想デスクトップセッションへのログオン時に認証を行うことができます。この機能は、ICA®スマートカード仮想チャネル

を介したスマートカードリダイレクトによって実装されます。セッション内でスマートカードを使用することもできます。使用例は次のとおりです。

- ドキュメントへのデジタル署名の追加
- メールの暗号化または復号化
- Webサイトへの認証

Linux VDAは、この機能についてWindows VDAと同じ構成を使用します。詳しくは、この記事の「スマートカード環境の構成」セクションを参照してください。

> **注:**
>
> Linux VDAセッション内でマッピングされたスマートカードを使用してCitrix Gatewayにサインオンすることはサポートされていません。

前提条件

スマートカードパススルー認証の可用性は、以下の条件に左右されます。

- Linux VDAがDelivery Controller™に登録でき、Windows資格情報を使用して公開されたLinuxデスクトップセッションを開くことができること。
- OpenSCでサポートされているスマートカードが使用されていること。詳しくは、「OpenSCがスマートカードをサポートしていることを確認する」を参照してください。

OpenSCがスマートカードをサポートしていることを確認する

OpenSCは、RHEL 7.4以降で広く使用されているスマートカードドライバーです。CoolKeyと完全に互換性のある代替品として、OpenSCは多くの種類のスマートカードをサポートしています（「Red Hat Enterprise Linuxでのスマートカードのサポート」を参照）。

この記事では、YubiKeyスマートカードを構成例として使用して説明します。YubiKeyは、Amazonなどの小売業者から簡単に購入できるオールインワンのUSB CCID PIVデバイスです。OpenSCドライバーはYubiKeyをサポートしています。

組織でより高度なスマートカードが必要な場合は、サポートされているLinuxディストリビューションとOpenSCパッケージがインストールされた物理マシンを準備してください。OpenSCのインストールについては、「スマートカードドライバーのインストール」を参照してください。スマートカードを挿入し、次のコマンドを実行して、OpenSCがスマートカードをサポートしていることを確認します。

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を完了します。

1. ルート証明書をPEM形式で取得します。通常、CAサーバーから取得します。

   DERファイル（*.crt、*.cer、*.der）をPEMに変換するには、次のようなコマンドを実行します。次のコマンド例では、**certnew.cer**はDERファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   ```

2. ルート証明書を`openssl`ディレクトリにインストールします。**certnew.pem**ファイルは例として使用されます。

   ```
   cp certnew.pem <path where you install the root certificate>
   ```

   ルート証明書をインストールするためのパスを作成するには、`sudo mdkir -p <path where you install the root certificate>`を実行します。

スマートカード環境を構成する

ctxsmartlogon.shスクリプトを使用してスマートカード環境を構成するか、手動で構成を完了できます。

（オプション1）CTXSMARTLOGON.SHスクリプトを使用してスマートカード環境を構成する

> **注:**
>
> ctxsmartlogon.shスクリプトは、PKINIT情報をデフォルトレルムに追加します。この設定は、**/etc/krb5.conf**構成ファイルを通じて変更できます。

> スマートカードを初めて使用する前に、ctxsmartlogon.shスクリプトを実行してスマートカード環境を構成します。

ヒント:

ドメイン参加にSSSDを使用している場合は、ctxsmartlogon.shを実行した後、SSSDサービスを再起動してください。 RHEL 8以降では、PAM_KRB5がEPELリポジトリに移動されたため、スクリプトはこれらのディストリビューションでEPELを有効にしようとします。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

```
#*********************************************************************
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*********************************************************************
Step 1:Enable smart card logon.
  Do you want enable smart card logon? (y/n)[y] y
Step 2:Select the Active Directory integration method.
  Please select which Active Directory integration method to use:
     1: Winbind
     2: SSSD
     3: Centrify
  Select one of the above options (1-3)[1] 1
Step 3:Install dependent packages.
  Installing, please wait...
    [krb5-pkinit][pam_krb5] already installed.
    [pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][coolkey] already installed.
  Packages installed. [Success]
Step 4:Configure krb5.conf.
  Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
  /etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
  Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/libcoolkeypk11.so):/usr/lib64/pkcs11/libcoolkeypk11.so
  /etc/pam.d/ctxfsc configure successfully.
  /etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

注:

ドメイン参加方法としてQuestを使用する場合、Key Distribution Center (KDC) のホスト名を指定する必要があります。例：

](/en-us/linux-virtual-delivery-agent/2411/media/kdc-host-name-being-required.png)

ctxsmartlogon.shスクリプトを実行してスマートカードを無効にすることもできます。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

```
#*******************************************************************************
# ctxsmartlogon.sh 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いた構成については、Knowledge Centerの記事CTX206156を参照してください。

次のステップに進む前に、以下を確認してください。

- すべてのコンポーネントが正しく構成されていること。
- 秘密キーとユーザー証明書をスマートカードにダウンロードしていること。
- スマートカードを使用してVDAに正常にログオンできること。

PC/SC Liteパッケージをインストールする
PCSC Liteは、LinuxにおけるPersonal Computer/Smart Card (PC/SC) 仕様の実装です。スマートカードおよびリーダーと通信するためのWindowsスマートカードインターフェイスを提供します。Linux VDAでのスマートカードリダイレクトは、PC/SCレベルで実装されます。

PC/SC Liteパッケージをインストールするには、次のコマンドを実行します。

RHEL 9.4/9.2/8.x、Rocky Linux 9.4/9.2/8.x の場合：

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

Ubuntu 24.04、Ubuntu 22.04、Ubuntu 20.04、Debian 12.7、Debian 12.5、Debian 11.11:

```
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 9.4/9.2/8.x、Rocky Linux 9.4/9.2/8.x の場合：

```
yum install opensc
```

Ubuntu 24.04、Ubuntu 22.04、Ubuntu 20.04、Debian 12.7、Debian 12.5、Debian 11.11：

```
apt-get install -y opensc
```

スマートカード認証用のPAMモジュールをインストールする
pam_krb5およびkrb5-pkinitモジュールをインストールするには、次のコマンドを実行します。

RHEL 9.4/9.2/8.x バージョン、Rocky Linux 9.4/9.2/8.x バージョン:

```
yum install krb5-pkinit
```

Ubuntu 24.04 バージョン、Ubuntu 22.04 バージョン、Ubuntu 20.04 バージョン:

```
apt-get install libpam-krb5 krb5-pkinit
```

Debian 12.7、Debian 12.5、Debian 11.11 の各バージョン:

```
apt-get install -y libpam-krb5 krb5-pkinit
```

pam_krb5モジュールは、プラガブル認証モジュールです。PAM対応アプリケーションは、pam_krb5を使用してパスワードをチェックし、キー配布センター (KDC) からチケット許可チケットを取得できます。krb5-pkinitモジュールには、クライアントが秘密鍵と証明書を使用してKDCから初期資格情報を取得できるようにするPKINITプラグインが含まれています。
pam_krb5モジュールを構成する
pam_krb5モジュールは、スマートカード内の証明書を使用してKerberosチケットを取得するためにKDCと対話します。PAMでpam_krb5認証を有効にするには、次のコマンドを実行します。

```
authconfig --enablekrb5 --update
```

/etc/krb5.conf構成ファイルで、実際のレルムに従ってPKINIT情報を追加します。

注:

pkinit_cert_matchオプションは、PKINIT認証を試行する前にクライアント証明書が一致する必要がある一致ルールを指定します。一致ルールの構文は次のとおりです。

component-rule の設定...

ここで、**relation-operator**は、すべてのコンポーネントルールが一致する必要があることを意味する**&&、または1つのコンポーネントルールのみが一致する必要があることを意味する\|\|**のいずれかです。

汎用krb5.confファイルの例を次に示します。

```
EXAMPLE.COM = {

    kdc = KDC.EXAMPLE.COM

    auth_to_local = RULE:[1:$1@$0]

    pkinit_anchors = FILE:<path where you install the root certificate>/
certnew.pem

    pkinit_kdc_hostname = KDC.EXAMPLE.COM

    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature

    pkinit_eku_checking = kpServerAuth
```

```
}
```

PKINIT情報を追加すると、構成ファイルは次のようになります。

```
CTXDEV.LOCAL = {
    kdc = [redacted]
    auth_to_local = RULE:[1:$1@$0]
    pkinit_kdc_hostname = ctx-ad.ctxdev.local
    pkinit_anchors = FILE:/etc/pki/CA/certs/certnew.pem
    pkinit_eku_checking = kpServerAuth
    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
}
```

PAM認証を構成する

PAM構成ファイルは、PAM認証に使用されるモジュールを示します。pam_krb5を認証モジュールとして追加するには、**/etc/pam.d/smartcard-auth** ファイルに次の行を追加します。

`auth [success=done ignore=ignore default=die] pam_krb5.so preauth_options=X509_user_identity=PKCS11:<path to the pkcs11 driver>/opensc-pkcs11.so`

SSSDを使用する場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_opt=X509_user_identity=PKCS11:/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so
auth        sufficient    pam_permit.so
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore]  pam_krb5.so
account     required      pam_permit.so

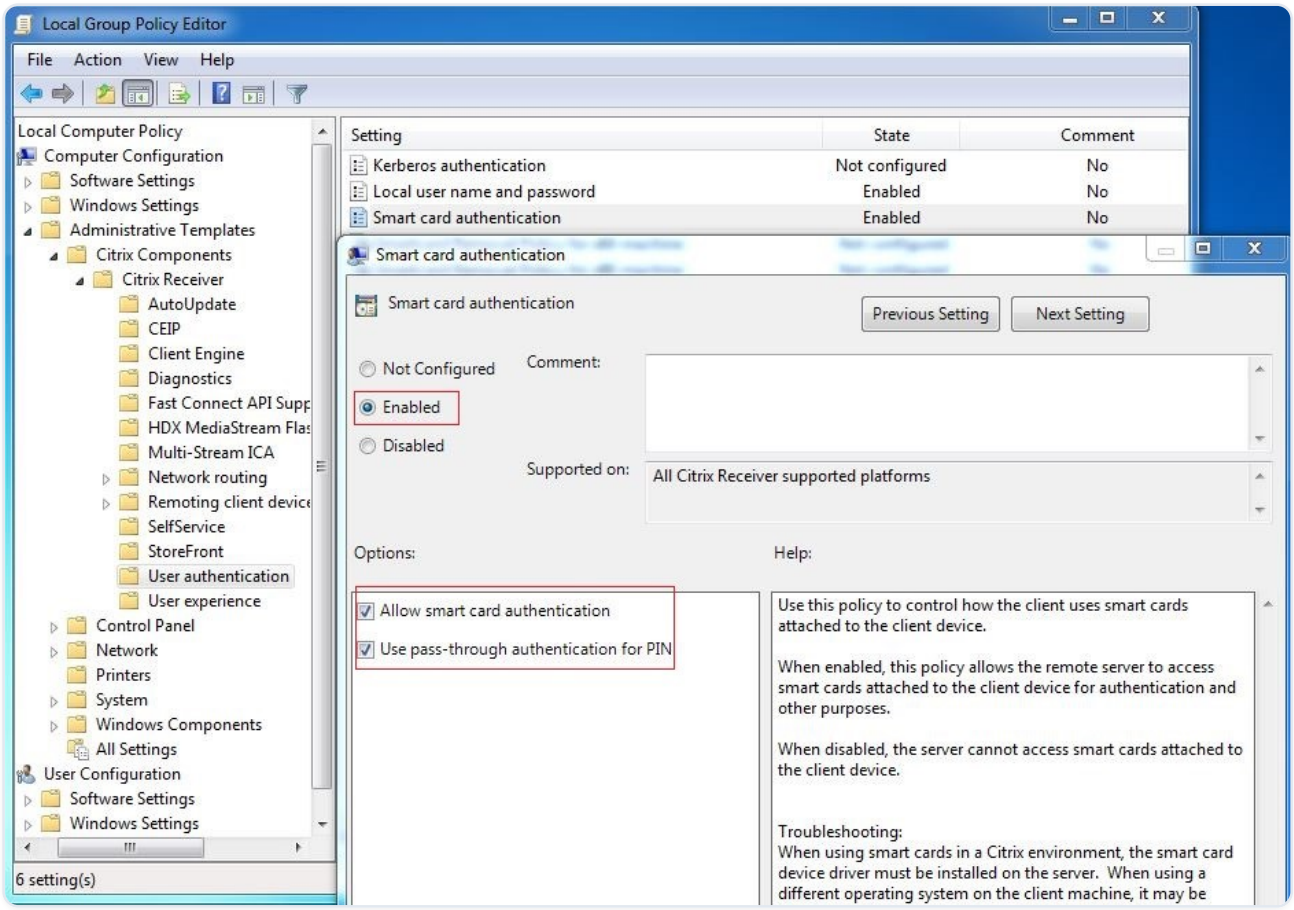
session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
#session     optional      pam_oddjob_mkhomedir.so umask=0077
session     optional      pam_mkhomedir.so umask=0077
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_sss.so
session     optional      pam_krb5.so
```

(オプション) スマートカードを使用したシングルサインオン

シングルサインオン (SSO) は、仮想デスクトップおよびアプリケーションの起動でパススルー認証を実装するCitrixの機能です。この機能により、ユーザーがPINを入力する回数が減ります。

Linux VDAでSSOを使用するには、Citrix Workspaceアプリを構成します。構成はWindows VDAと同じです。詳細については、Knowledge Centerの記事 CTX133982を参照してください。

Citrix Workspace™アプリでグループポリシーを構成する際に、スマートカード認証を次のように有効にします。



Fast Smart Cardログオン

Fast Smart Cardは、既存のHDX PC/SCベースのスマートカードリダイレクトを改善したものです。これにより、高遅延WAN環境でスマートカードを使用する際のパフォーマンスが向上します。詳細については、スマートカードを参照してください。

Linux VDAは、Citrix Workspaceアプリの次のバージョンでFast Smart Cardをサポートしています。

- シトリックス レシーバー Windows用 4.12
- Windows向けCitrix Workspaceアプリ1808以降

クライアントでFAST SMART CARDログオンを有効にする

高速スマートカードログオンは、VDAではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連するStoreFrontサイトのdefault.icaファイルに次のパラメーターを含めます。

```
[WFClient]

SmartCardCryptographicRedirection=On
```

クライアントで高速スマートカードログオンを無効にする

クライアントで高速スマートカードログオンを無効にするには、関連するStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

XDPingを実行する

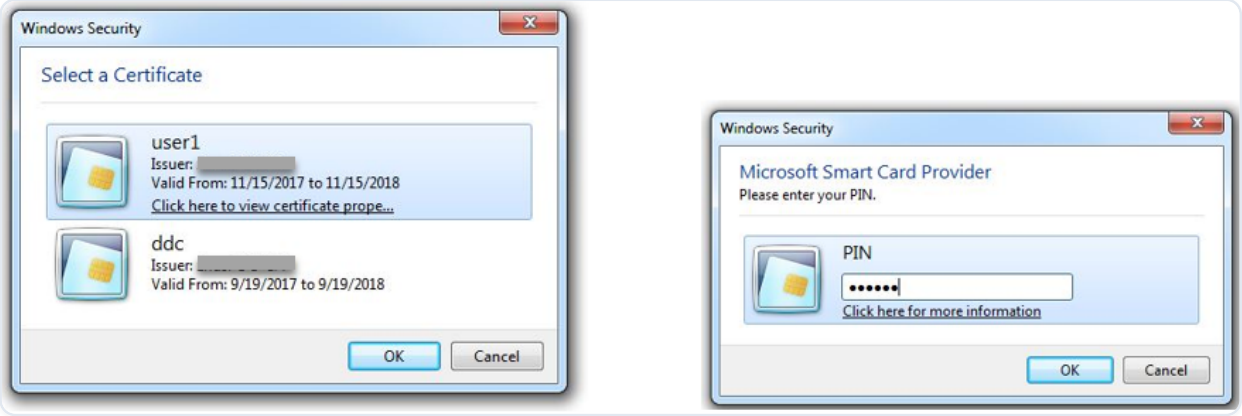
上記の手順を完了した後、Linux **XDPing**ツールを使用して、Linux VDA環境に存在する可能性のある一般的な構成の問題を確認できます。

使用方法

スマートカードを使用してLinux VDAにログオンする

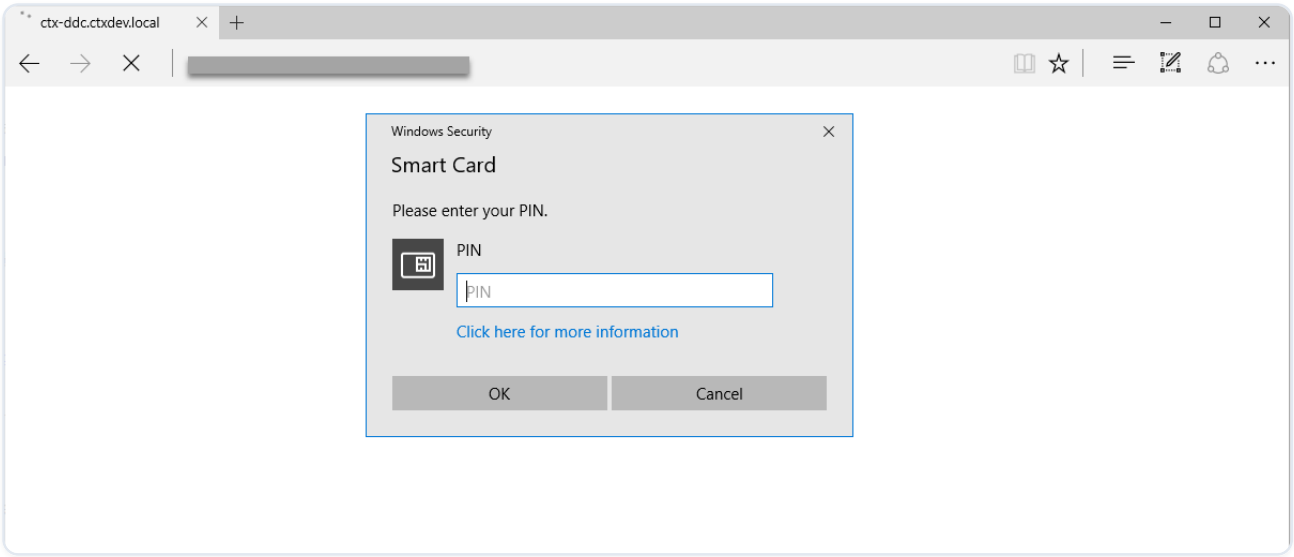
スマートカードを使用して、SSOシナリオと非SSOシナリオの両方でLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用してStoreFront™に自動的にログオンします。StoreFrontでLinux仮想デスクトップセッションを起動すると、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、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起動すると、次のようにLinux VDAへのログオンダイアログボックスが表示されます。ユーザー名はスマートカードの証明書から抽出され、ログオン認証のためにPINを再度入力する必要があります。

この動作はWindows VDAと同じです。



スマートカードを使用してセッションに再接続する

セッションに再接続するには、スマートカードがクライアントデバイスに接続されていることを確認してください。そうしないと、Linux VDA側で灰色のキャッシュウィンドウが表示され、スマートカードが接続されていないため再認証が失敗し、すぐに終了します。この場合、スマートカードを接続するよう促す他のプロンプトは表示されません。

ただし、StoreFront側では、セッションに再接続するときにスマートカードが接続されていない場合、StoreFront Webは次のようなアラートを表示する可能性があります。

スマートカードの挿入(/en-us/linux-virtual-delivery-agent/2411/media/please-insert-smart-card.png)

制限事項

限られたLinuxディストリビューションとAD統合方法のサポート

- スマートカードパススルー認証は、限られたLinuxディストリビューションとAD統合方法をサポートしています。以下のマトリックスを参照してください。

	ウィンバインド	SSSD	セントリファイ	クエスト
デビアン 12.7/12.5/11.11	はい	はい	はい	いいえ
RHEL 9.4/9.2	はい	はい	いいえ	はい
RHEL 8.10/8.8	はい	はい	はい	はい

	ウィンバインド	SSSD	セントリファイ	クエスト
ロッキー Linux 9.4/9.2	はい	はい	いいえ	いいえ
ロッキー Linux 8.10/8.8	はい	はい	いいえ	いいえ
ウブントゥ 24.04	はい	はい	いいえ	いいえ
ウブントゥ 22.04/20.04	はい	はい	はい	はい

スマートカード取り外し設定

サインインしているユーザーのスマートカードがセッション中にスマートカードリーダーから取り外されたときの動作を指定するには、Linux VDAで次のレジストリキーを編集します。

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LocalPolicies\SecurityOptions

値の名前	タイプ	デフォルト	説明
SCardRemoveOption	REG_DWORD	0x00000000	この設定は、サインインしているユーザーのスマートカードがセッション中にスマートカードリーダーから取り外されたときに何が起こるかを決定します。オプションは次のとおりです。
\^	\^	\^	0 **アクションなし**
\^	\^	\^	1 **強制ログオフ**: スマートカードが取り外されると、ユーザーは自動的にサインアウトされます。

値の名前	タイプ	デフォルト	説明
\^	\^	\^	2 **セッション切断**: スマートカードが取り外されると、ユーザーをサインアウトせずにセッションが切断されます。ユーザーは後でスマートカードを再挿入してセッションに再接続できます。
SCardRemoveActionDelaySeconds	REG_DWORD	15	取り外しオプションが**強制ログオフ**または**セッション切断**に設定されている場合、アクションが実行されるまでの遅延を数秒間指定できます。この設定により、スマートカードが取り外された後、ユーザーセッションにメッセージボックスが表示され、指定された秒数後にセッションが強制ログオフまたは切断されることを示します。その時間内にスマートカードを再挿入すると、セッションは中断されずに続行されます。

制限事項

Linux VDAは、一度に1つのスマートカードリーダーのみをサポートします。

その他のスマートカードとPKCS#11ライブラリのサポート

Citrixは汎用スマートカードリダイレクトソリューションを提供します。サポートリストにはOpenSCスマートカードのみが記載されていますが、他のスマートカードやPKCS#11ライブラリの使用を試すことができます。特定のスマートカードまたはPKCS#11ライブラリに切り替えるには：

1. すべての`opensc-pkcs11.so`インスタンスを、お使いのPKCS#11ライブラリに置き換えます。
2. PKCS#11ライブラリのパスをレジストリに設定するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v
   "PKCS11LibPath" -d "PATH"
   ```

   ここで、**PATH** は `/usr/lib64/pkcs11/opensc-pkcs11.so` のようなPKCS#11ライブラリを指します。
3. クライアントで高速スマートカードログオンを無効にします。

認証されていない（匿名）ユーザーによるアクセス

ユーザーは、StoreFront™またはCitrix Workspace™アプリに資格情報を提示することなく、アプリケーションやデスクトップにアクセスできます。認証されていないユーザーにアクセスを許可するには、認証されていないStoreFrontストアを用意し、デリバリーグループで認証されていないユーザーのアクセスを有効にする必要があります。

> **注:**
>
> 認証されていないユーザーによるアクセスは、ドメイン参加済みのVDAでのみサポートされます。

> セッションの事前起動は、認証されていないユーザーではサポートされていません。セッションの事前起動は、Android版Citrix Workspaceアプリでもサポートされていません。

認証されていないStoreFrontストアを作成する

1. Citrix StoreFront管理コンソールの左ペインで**ストア**ノードを選択し、**操作**ペインで**ストアの作成**をクリックします。

2. **ストア名**ページで、ストアの名前を指定し、**認証されていないユーザーのみがこのストアにアクセスすることを許可する**を選択して、**次へ**をクリックします。

詳しくは、「ストアの作成」を参照してください。

デリバリーグループで認証されていないユーザーのアクセスを有効にする

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループ内のアプリケーションやデスクトップを使用できるユーザーを指定する際に、認証されていないユーザーにアクセスを許可できます。詳しくは、 デリバリーグループの作成を参照してください。

認証されていないユーザーセッションのアイドルタイムアウトを設定する

認証されていないユーザーセッションには、デフォルトで10分間のアイドルタイムアウトがあり、クライアントが切断されると自動的にログオフされます。レジストリ設定**AnonymousUserIdleTime**を通じてカスタムアイドルタイムアウトを構成できます。たとえば、5分間のカスタムアイドルタイムアウトを設定するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v AnonymousUserIdleTime
 -d 0x00000005
```

認証されていないユーザーの最大数を設定する

認証されていないユーザーの最大数を設定するには、レジストリキー**MaxAnonymousUserNumber**を使用します。この設定は、単一のLinux VDAで同時に実行される認証されていないユーザーセッションの数を制限します。このレジストリ設定を構成するには、**ctxreg**ツールを使用します。たとえば、値を32に設定するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v MaxAnonymousUserNumber
 -d  0x00000020
```

重要:

認証されていないユーザーセッションの数を制限してください。同時に起動されるセッションが多すぎると、利用可能なメモリ不足など、VDAで問題が発生する可能性があります。

トラブルシューティング

認証されていないユーザーセッションを設定する際に、以下を考慮してください。

- **認証されていないユーザーセッションへのログオンに失敗しました。**

レジストリが以下を含むように更新されていることを確認してください (0に設定)。

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsd サービスが実行されており、**passwd** キャッシュを有効にするように構成されていることを確認してください。

```
ps  uax | grep nscd

cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

passwd キャッシュ変数が有効になっている場合は **no** に設定し、**ncsd** サービスを再起動します。この構成を変更した後、Linux VDA を再インストールする必要がある場合があります。

- **KDE を使用する認証されていないユーザーセッションでロック画面ボタンが表示される。**

ロック画面ボタンとメニューは、認証されていないユーザーセッションではデフォルトで無効になっています。ただし、KDE では引き続き表示されることがあります。KDE で特定のユーザーのロック画面ボタンとメニューを無効にするには、構成ファイル **$Home/.kde/share/config/kdeglobals** に次の行を追加します。例：

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、`/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals` のようなグローバルな `kdeglobals` ファイルで `KDE Action Restrictions` パラメーターが不変として構成されている場合、ユーザー構成は効果がありません。

この問題を解決するには、システム全体の `kdeglobals` ファイルを変更して、`[KDE Action Restrictions]` セクションの `[$i]` タグを削除するか、システム全体の構成を直接使用してロック画面ボタンとメニューを無効にします。KDE の構成の詳細については、KDE System Administration/Kiosk/Keys ページ を参照してください。

ファイル

このセクションには、以下のトピックが含まれています。

- ファイルのコピーと貼り付け
- ファイル転送

ファイルのコピーと貼り付け

ユーザーは、右クリックメニューまたはキーボードショートカットを使用して、セッションとローカルクライアント間でファイルをコピー＆ペーストできます。この機能には、Citrix Virtual Apps and Desktops™ 2006以降、およびWindows用Citrix Workspace™アプリ1903以降が必要です。

ファイルを正常にコピー＆ペーストするには、以下を確認してください。

- ファイルの最大数が20を超えないこと。
- 最大ファイルサイズが200 MBを超えないこと。
- Linux VDAをインストールしたマシンでNautilusファイルマネージャーが利用可能であること。
- ファイル名にASCII文字のみが含まれ、特殊文字が含まれていないこと。

サポートされているLinuxディストリビューション

ファイルのコピー＆ペースト機能は、Linux VDAがサポートするすべてのLinuxディストリビューションで利用できます。

関連するポリシー

以下のクリップボードポリシーは、この機能の構成に関連しています。クリップボードポリシーの詳細については、「ポリシーサポートリスト」を参照してください。

- クライアントクリップボードリダイレクト
- クリップボード選択更新モード
- プライマリ選択更新モード

> **注:**
>
> ファイルのコピー＆ペースト機能を無効にするには、Citrix Studioで**クライアントクリップボードリダイレクト**ポリシーを**禁止**に設定します。 **クリップボード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ポリシーと**クリップボードのセッションからクライアントへの転送サイズを制限する**ポリシーは、転送されるファイルのサイズではなく、クリップボードのバッファサイズを制御します。サポートされているポリシーの詳細については、「ポリシーサポートリスト」を参照してください。

制限事項

- 切り取りはサポートされていません。ファイルを切り取る要求は、コピー操作として扱われます。
- ドラッグアンドドロップはサポートされていません。
- ディレクトリのコピーはサポートされていません。
- ファイルのコピーと貼り付けは順次実行する必要があります。前のファイルが正常にコピーされ、貼り付けられた後にのみ、次のファイルをコピーできます。

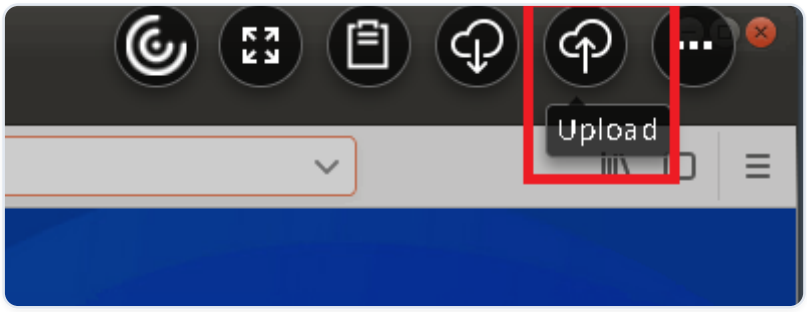
ファイル転送

ファイル転送は、Linux VDAとクライアントデバイス間でサポートされています。この機能は、クライアントデバイスがHTML5サンドボックス属性をサポートするWebブラウザを実行している場合に利用できます。HTML5サンドボックス属性を使用すると、ユーザーはCitrix Workspace™ app for HTML5およびfor Chromeを使用して仮想デスクトップとアプリにアクセスできます。

注:

ファイル転送は、Citrix Workspace app for HTML5およびfor Chromeで利用できます。

公開アプリケーションおよびデスクトップセッション内では、ファイル転送により、Linux VDAとクライアントデバイス間でファイルのアップロードとダウンロードが可能です。クライアントデバイスからLinux VDAにファイルをアップロードするには、Citrix Workspaceアプリのツールバーにある**アップロード**アイコンをクリックし、ファイルダイアログから目的のファイルを選択します。Linux VDAからクライアントデバイスにファイルをダウンロードするには、**ダウンロード**アイコンをクリックします。アップロードまたはダウンロード中にファイルを追加できます。一度に最大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をアップロードおよびダウンロードするには、Citrix Workspaceアプリのツールバーを有効にします。 ファイルをドラッグアンドドロップできるバージョンのCitrix Workspaceアプリを使用できます。

自動ダウンロードは、ファイル転送の機能強化です。VDA上の**マイデバイスに保存**ディレクトリにダウンロードまたは移動したファイルは、クライアントデバイスに自動的に転送されます。

注:

自動ダウンロードには、**Citrix Workspace app for ChromeOS/HTML5のファイル転送**ポリシーと**Citrix Workspace app for ChromeOS/HTML5のファイルダウンロード**ポリシーが**許可**に設定されている必要があります。

自動ダウンロードのユースケースをいくつか示します。

- **マイデバイスに保存**にファイルをダウンロードする

  公開デスクトップおよびWebブラウザアプリケーションセッション内では、Webサイトからダウンロードしたファイルを**VDA上のマイデバイスに保存**ディレクトリに保存して、クライアントデバイスに自動転送できます。自動ダウンロードを実現するには、セッション内Webブラウザのデフォルトダウンロードディレクトリを**マイデバイスに保存**に設定し、Citrix Workspace app for HTML5またはfor Chromeを実行するWebブラウザでローカルダウンロードディレクトリを設定します。

- **マイデバイスに保存**にファイルを移動またはコピーする

公開デスクトップセッション内では、対象ファイルを選択し、それらを**マイデバイスに保存**ディレクトリに移動またはコピーして、クライアントデバイスで利用できるようにします。

ファイル転送ポリシー

デフォルトでは、ファイル転送は有効になっています。これらのポリシーは、**ユーザー設定 > ICA\ファイルリダイレクト**の下にあります。Citrix Studioを使用して変更してください。ファイル転送ポリシーを使用する際は、以下を考慮してください。

Citrix Workspace app for ChromeOS/HTML5のファイル転送: エンドユーザーがCitrix Virtual Apps and DesktopsセッションまたはCitrix DaaSセッションとエンドユーザーのデバイス間でファイルを転送することを許可または禁止します。

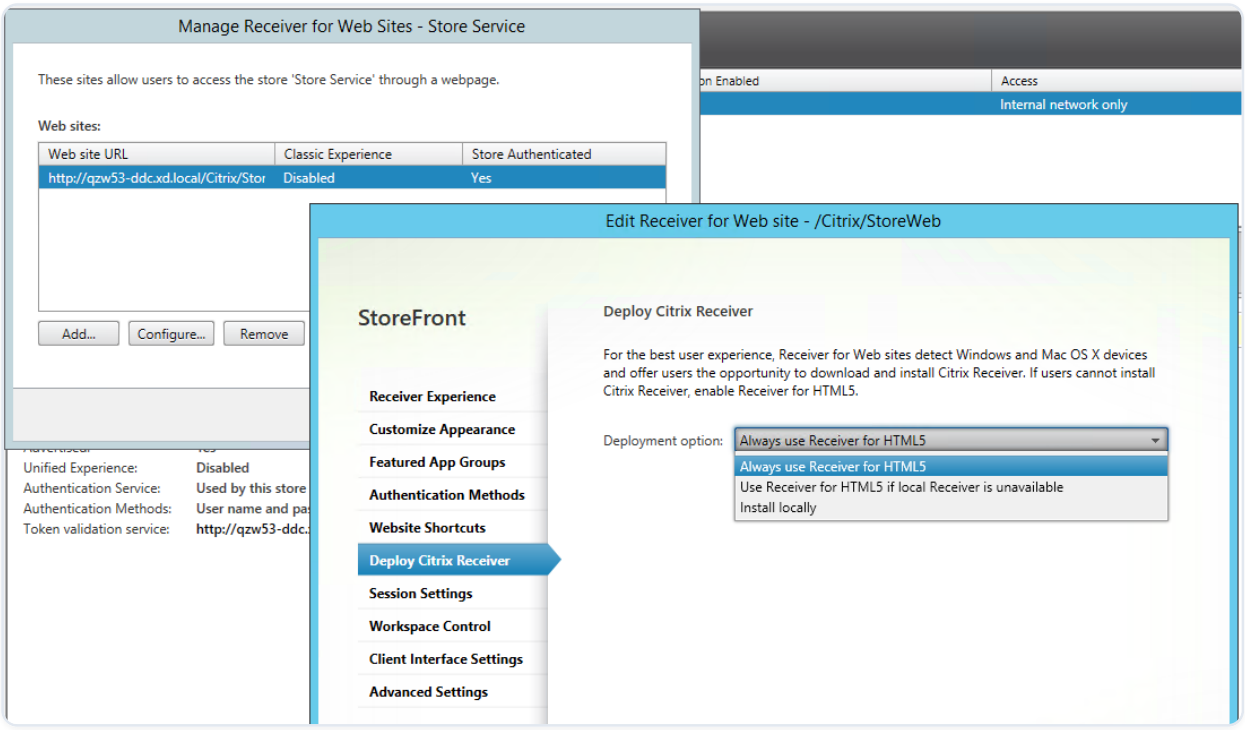
Citrix Workspace app for ChromeOS/HTML5のファイルアップロード: ユーザーがファイルをアップロードすることを許可または禁止します。ユーザーのデバイスからCitrix Virtual Apps and DesktopsセッションまたはCitrix DaaSセッションへのアップロードが可能です。

Citrix Workspace app for ChromeOS/HTML5のファイルダウンロード: ユーザーがファイルをダウンロードすることを許可または禁止します。Citrix Virtual Apps and Desktopsセッションまたは Citrix DaaSセッションからユーザーのデバイスへのダウンロードが可能です。

使用方法

Citrix Workspace app for HTML5 を介してファイル転送機能を使用するには:

1. Citrix Studio で、**WebSockets connections** ポリシーを **Allowed** に設定します。
2. Citrix Studio で、前述のファイル転送ポリシーを介してファイル転送を有効にします。
3. Citrix StoreFront 管理コンソールで、**Stores** をクリックし、**Manage Receiver for Web Sites** ノードを選択して、**Always use Receiver for HTML5** オプションを選択して Citrix Receiver™ for HTML5 を有効にします。



4. 仮想デスクトップまたはWebブラウザアプリセッションを起動します。Linux VDAとクライアントデバイス間で1つ以上のファイル転送を実行します。

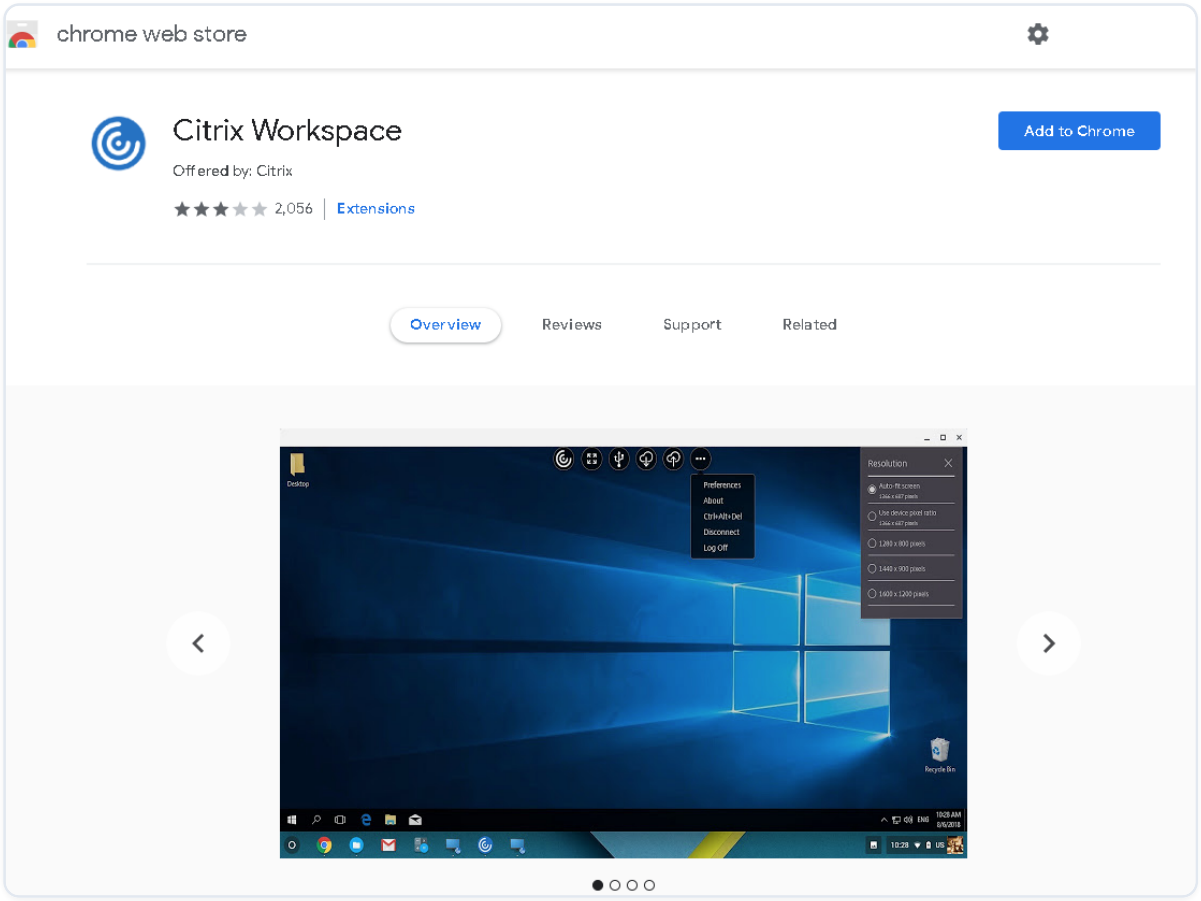
Citrix Workspace app for Chrome を介してファイル転送機能を使用するには:

1. 前述のファイル転送ポリシーを介してファイル転送を有効にします。
2. Chrome Web Store から Citrix Workspace app を入手します。

   Citrix Workspace app for Chrome を Chrome アプリページにすでに追加している場合は、この手順をスキップします。

   a. Google Chrome の検索ボックスに「**Citrix Workspace for Chrome**」と入力します。検索アイコンをクリックします。

   b. 検索結果の中から、Citrix Workspace app が利用可能な Chrome Web Store へのURLをクリックします。



c. Citrix Workspace app を Google Chrome に追加するには、**Add to Chrome** をクリックします。

3. Chrome アプリページで Citrix Workspace app for Chrome をクリックします。
4. 接続するStoreFront™ストアのURLを入力します。

   以前にURLを入力したことがある場合は、この手順をスキップします。
5. 仮想デスクトップまたはアプリセッションを起動します。Linux VDAとクライアントデバイスの間で1つ以上のファイル転送を実行します。

グラフィックス

このセクションには、以下のトピックが含まれています。

- 自動DPIスケーリング(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/automatic-dpi-scaling.html)
- クライアントのバッテリーステータス表示(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/client-battery-status-display.html)
- グラフィックの構成と微調整(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/configuring-graphics.html)

- HDX画面共有(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/hdx-screen-sharing.html)
- 非仮想化GPU(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/non-virtualized-gpus.html)
- セッションウォーターマーク(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/session-watermark.html)
- Thinwireプログレッシブディスプレイ(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/thinwire-progressive-display.html)

自動DPIスケーリング

Linux VDA は自動 DPI スケーリングをサポートしています。ユーザーが仮想デスクトップまたはアプリケーションセッションを開くと、セッション内の DPI 値は、クライアント側の DPI 設定に合わせて自動的に変更されます。

この機能に関する考慮事項は次のとおりです。

- この機能を使用するには、Citrix Workspace で DPI マッチングを有効にする必要があります。Citrix Workspace アプリ for Windows の場合、**いいえ、ネイティブ解像度を使用します** オプションが選択されていることを確認してください。Citrix Workspace アプリ for Windows の DPI スケーリングの構成の詳細については、「DPI スケーリング」を参照してください。
- マルチモニター環境でこの機能を使用するには、各モニターが同じ DPI 設定で構成されている必要があります。混在 DPI シナリオはサポートされていません。モニターが異なる DPI 設定で構成されている場合、Linux VDA はすべての画面に最小の DPI 値を適用します。
- この機能は、MATE、GNOME、GNOME Classic、および KDE で有効になっています。KDE または MATE を使用する場合は、次の点を考慮してください。
  - KDE デスクトップ環境で実行されている Linux 仮想デスクトップの場合：
    - KDE Plasma 5 以降を使用することをお勧めします。
    - セッションの実行中にクライアント側で DPI 設定を変更すると、ユーザーはログオフして再度ログオンする必要があります。
  - MATE デスクトップ環境で実行されている Linux 仮想デスクトップの場合：
    - スケールファクター 1 および 2 のみがサポートされています。
    - セッションの実行中にクライアント側で DPI 設定を変更すると、ユーザーはログオフして再度ログオンする必要があります。

- 仮想セッションの DPI 値は、クライアント側の DPI 設定に従って自動的に変更されます。現在、この機能は、100% や 200% など、整数型のスケールファクターのみをサポートしています。クライアント側で構成されているスケールファクターが小数型の場合、仮想セッションの DPI は次の表に従って整数スケールファクターに変更されます。例：スケールファクターが 125% の場合、DPI 値は 100% に変更されます。

クライアント側のスケールファクター	リモートセッションの DPI	ディスプレイのスケーリング
174%以下	96 (1 x 96)	100%
175%～274%	192 (2 x 96)	200%
275%～399%	288 (3 x 96)	300%
400%以上	384 (4 x 96)	400%

> **注:**
>
> Linux VDAは、MATEデスクトップで最大200%のスケーリングをサポートします。

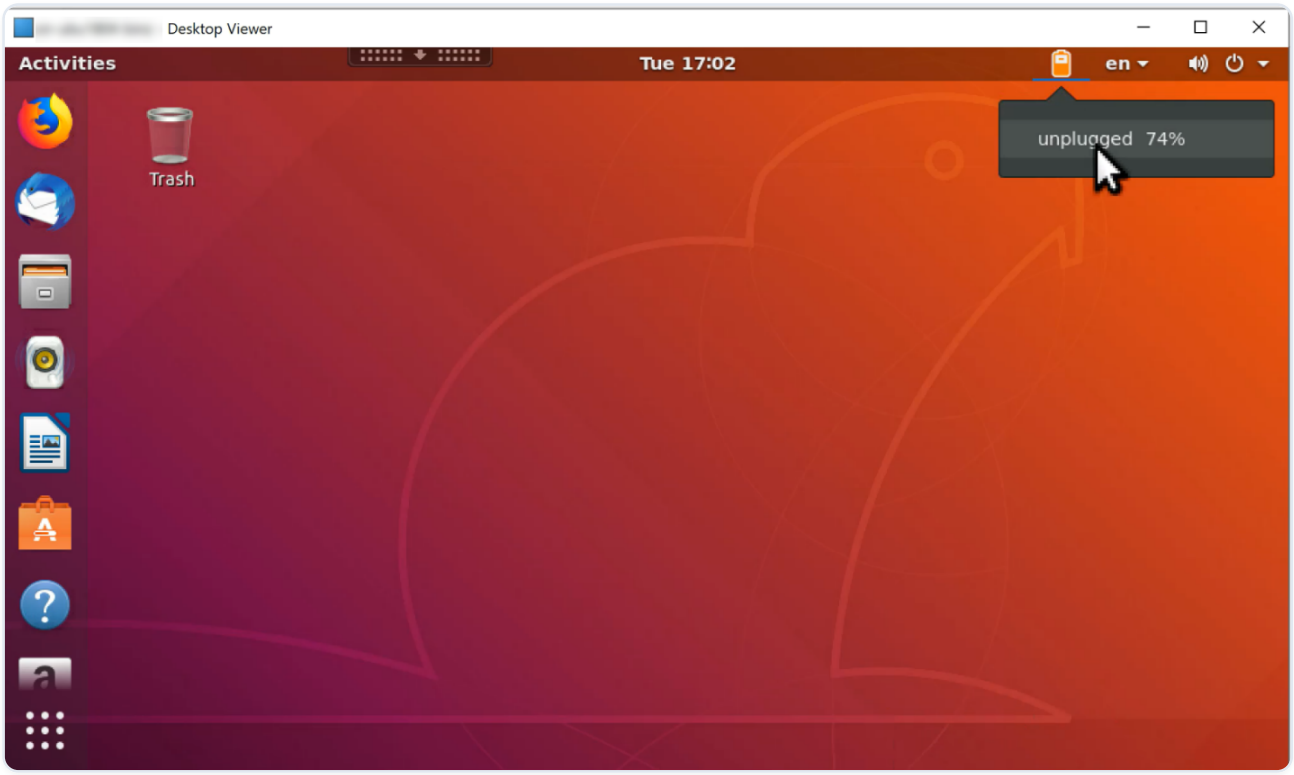
クライアントのバッテリー状態表示

Linux VDAは、仮想デスクトップ内のクライアントデバイスのバッテリー状態をリダイレクトして表示できます。この機能はデフォルトで有効になっており、以下のバージョンのCitrix Workspace™アプリで利用可能です。

- iOS向けCitrix Workspaceアプリ
- Linux向けCitrix Workspaceアプリ
- Mac版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版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。バッテリー残量の割合を確認するには、バッテリーアイコンをクリックします。例えば、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、以下の表を参照してください。

バッテリーアイコン	充電状態	バッテリー残量レベル	バッテリー残量割合
	充電中（「+」記号で表示）	高（緑色で表示）	>=80%
	充電中、「+」記号で示される	中、琥珀色で示される	20%以上80%未満
	充電中、「+」記号で示される	低、赤色で示される	< 20%
	充電中ではない、「-」記号で示される	高、緑色で示される	>=80%
	充電中ではない、「-」記号で示される	中、琥珀色で示される	>=20% かつ <80%
	充電中ではない、「-」記号で示される	低い、赤色で示される	< 20%

バッテリーアイコン	充電状態	バッテリー残量レベル	バッテリー残量割合
	不明	不明	不明

構成

クライアントバッテリーステータス表示は、デフォルトで有効になっています。

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000001"
```

注:

上記のコマンドは、クライアントバッテリーステータス表示とMobile Receiver Virtual Channel (MRVC)を共有するソフトキーボード機能に影響を与えます。

お使いのディストリビューションに基づいて、この機能を有効にするための追加手順を完了してください。詳細については、「システムトレイを有効にする」セクションを参照してください。

グラフィックの構成と微調整

この記事では、Linux VDAのグラフィック構成と微調整について説明します。

詳しくは、(/ja-jp/linux-virtual-delivery-agent/2411/system-requirements.html)と(/ja-jp/linux-virtual-delivery-agent/2411/installation-overview.html)セクションを参照してください。

構成

3Dグラフィックワークロード向けに最適化

この設定は、グラフィックを多用するワークロードに最適なデフォルト値を構成します。グラフィックを多用するアプリケーションに重点を置くユーザーに対して、この設定を有効にします。このポリシーは、セッションでGPUが利用可能な場合にのみ適用してください。このポリシーによって設定されたデフォルト設定を明示的に上書きする他の設定は、優先されます。

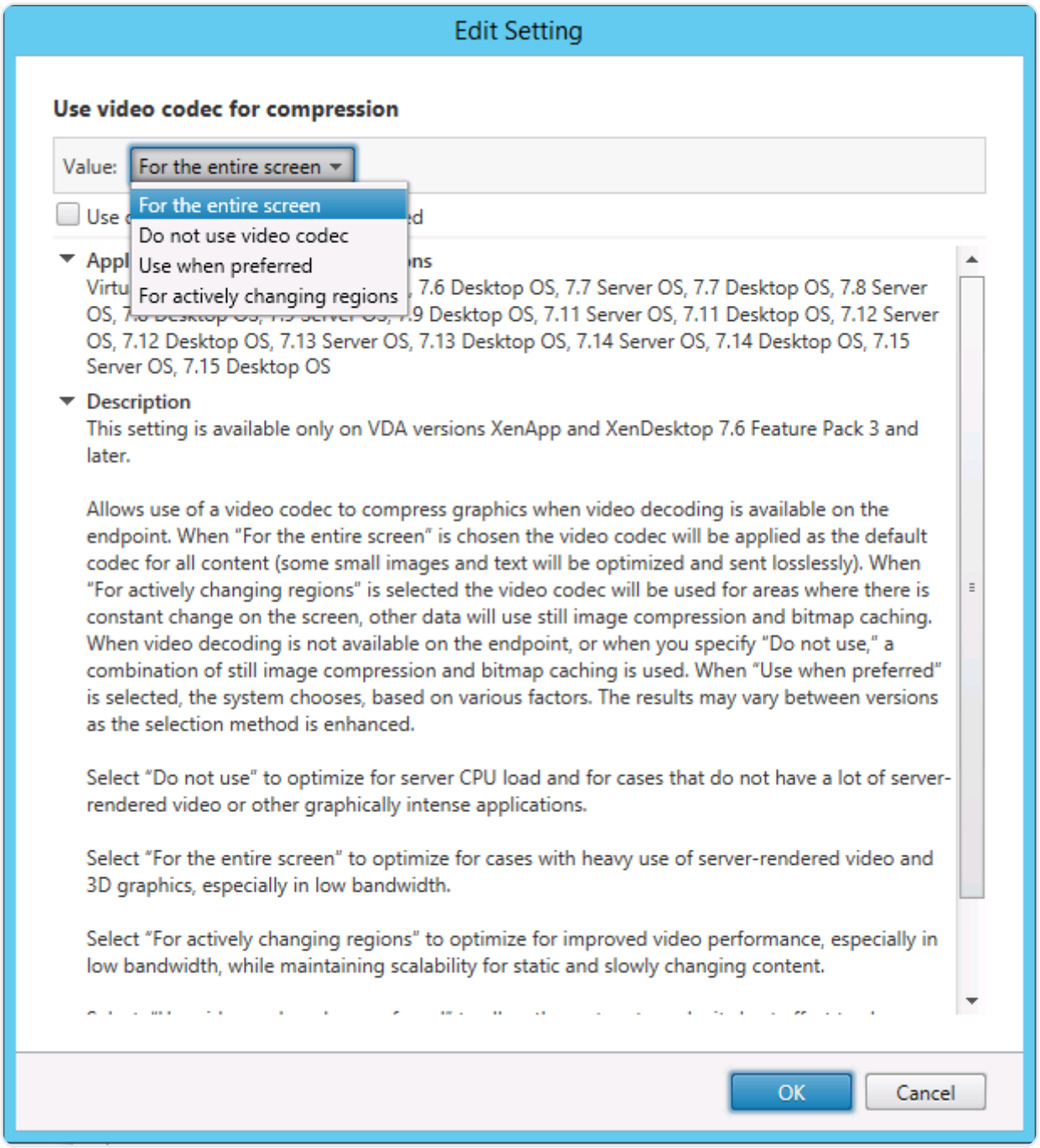
デフォルトでは、**3Dグラフィックワークロード向けに最適化**は無効になっています。

圧縮用ビデオコーデック

Thinwireは、Linux VDAで使用されるディスプレイリモートテクノロジーです。このテクノロジーにより、あるマシンで生成されたグラフィックを、通常はネットワーク経由で別のマシンに送信して表示できます。

(/ja-jp/citrix-virtual-apps-desktops/policies/reference/ica-policy-settings/graphics-policy-settings.html)グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して以下のオプションを提供します。

- **優先する場合にのみ使用する**。この設定はデフォルトです。追加の構成は不要です。これにより、すべてのCitrix®接続でThinwireが選択され、一般的なデスクトップワークロードのスケーラビリティ、帯域幅、および優れた画質のために最適化されます。
- **画面全体**。画面全体のH.264またはH.265でThinwireを提供し、特に3Dグラフィックを多用する場合に、ユーザーエクスペリエンスと帯域幅の向上を最適化します。(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/session-watermark.html)は、**画面全体**が選択されている場合、または**優先する場合にのみ使用する**が選択され、(/ja-jp/linux-virtual-delivery-agent/2411/configure/graphics/configuring-graphics.html#)が有効になっている場合にサポートされます。
- **アクティブに変更される領域**。Thinwireのアダプティブディスプレイテクノロジーは、動く画像（ビデオ、移動中の3D）を識別します。画像が動いている画面の一部でのみ、AV1、H.265、またはH.264を使用します。AV1、H.265、またはH.264ビデオコーデックを選択的に使用することで、HDX Thinwireは頻繁に更新される画面の一部を検出し、エンコードできます。静止画圧縮（JPEG、RLE）とビットマップキャッシュは、テキストや写真画像を含む画面の残りの部分で引き続き使用されます。ユーザーは、ビデオコンテンツの帯域幅消費量の削減と品質の向上、およびロスレステキストまたは高品質の画像を他の場所で組み合わせるという利点を得られます。**視覚品質**ポリシーが**常にロスレス**または**ロスレスにビルド**に設定されている場合、AV1およびH.265の選択的な使用はサポートされません。



ディスプレイリモート処理のパフォーマンスを微調整するために、以下の視覚表示ポリシー設定を含む他のポリシー設定を使用できます。

- シンプルなグラフィックの優先色深度
- ターゲットフレームレート
- 視覚品質

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する**」ポリシーは、利用可能な場合、GPUハードウェアアクセラレーションを使用して、ビデオコーデックで画面要素を圧縮する

ことを可能にします。GPUハードウェアアクセラレーションは、ハードウェアリソースの利用を最適化し、1秒あたりのフレーム数 (FPS) のパフォーマンスを大幅に向上させます。

GPUハードウェアアクセラレーションは、**圧縮にビデオコーデックを使用する**ポリシーによって設定されたすべてのグラフィックモードをカバーします。

- **「優先する場合にのみ使用」**
- **「画面全体」**
- **「アクティブに変更される領域」**

ハードウェアビデオ圧縮を有効にするには、次の手順を実行します。

1. **「ビデオコーデックにハードウェアエンコーディングを使用する」**ポリシーを**「有効」**に設定します。
2. **「圧縮にビデオコーデックを使用する」**を**「優先する場合にのみ使用」**、**「画面全体」**、または**「アクティブに変更される領域」**に設定します。**「ビデオコーデックを使用しない」**に設定されていないことを確認してください。

使用可能にするには、AV1またはH.265ビデオコーデックがVDAとCitrix Workspaceアプリの両方でサポートされ、有効になっている必要があります。コーデックネゴシエーション中、AV1はH.265およびH.264よりも優先されます。AV1がサポートされていない場合、H.265がネゴシエートされます。AV1とH.265の両方がサポートされていない場合、セッションはH.264ビデオコーデックの使用にフォールバックします。GPUハードウェアが利用できない場合、VDAはソフトウェアビデオコーデックを使用したCPUベースのエンコーディングにフォールバックします。

AV1ハードウェアエンコーディングの要件

VDA

- VDA: 2311以降
- GPU: **NVIDIA Ada Lovelace以降** (NVIDIA GPUがサポートするビデオコーデックのマトリックスについては、`<https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new>`のNVIDIAドキュメントを参照してください。)
- NVIDIAグラフィックドライバー 522.25以降 (Video Codec SDK v12.0)

クライアント

- Windows版 Citrix Workspace™ アプリ 2305 以降
- AV1デコードをサポートするクライアントGPU:
    - NVIDIA Ampere 以降

- Intel 第11世代 / Arc 以降
- AMD Radeon RX 6000 / Radeon Pro W6000シリーズ (RDNA2) 以降

H.265ハードウェアエンコードの要件

クライアント

- Windows版 シトリックス レシーバー 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ドを有効にするには、H.265ビデオエンコードを参照してください。

H.265/H.264ロスレス圧縮

H.265/H.264ロスレス圧縮は、NVIDIA GPUによるHDX 3D PROハードウェアアクセラレーションで利用できます。H.265ロスレス圧縮には、Citrix Workspace app 2305 for Windows以降が必要です。H.264ロスレス圧縮には、以下のクライアントが必要です。

- Citrix Workspace アプリ 2303 Windows 用以降
- Apple M1チップを搭載したCitrix Workspace app 2301 for Mac 以降

H.265/H.264ロスレス圧縮を有効にするには、次の手順を実行します。

1. **ビデオコーデックにハードウェアエンコードを使用**ポリシーを**有効**に設定します。
2. **圧縮にビデオコーデックを使用**ポリシーを**画面全体**に設定します。
3. 「**視覚品質**」ポリシーを「**常に損失**」または「**ロスレスに構築**」に設定します。

視覚的にロスレスな圧縮を許可する

「**視覚的にロスレスな圧縮を許可**」ポリシーは、グラフィックの真のロスレス圧縮の代わりに、**視覚的に**ロスレスな圧縮を使用することを許可します。視覚的にロスレスは、真のロスレスよりもパフォーマンスを向上させますが、目視では気づかない程度のわずかな損失があります。この設定は、「**視覚品質**」設定の値の使用方法を変更します。

「**視覚的にロスレスな圧縮を許可**」ポリシーは、デフォルトで無効になっています。**視覚的に**ロスレスな圧縮を有効にするには、「**視覚的にロスレスな圧縮を許可**」を「**有効**」に設定し、「**視覚品質**」ポリシーを「**ロスレスに構築**」に設定します。

「**圧縮にビデオコーデックを使用**」ポリシーが「**ビデオコーデックを使用しない**」に設定されている場合、**視覚的に**ロスレスな圧縮は静止画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**」ポリシーが「**ビデオコーデックを使用しない**」以外のグラフィックモードに設定されている場合、**視覚的に**ロスレスな圧縮はH.264エンコーディングに適用されます。

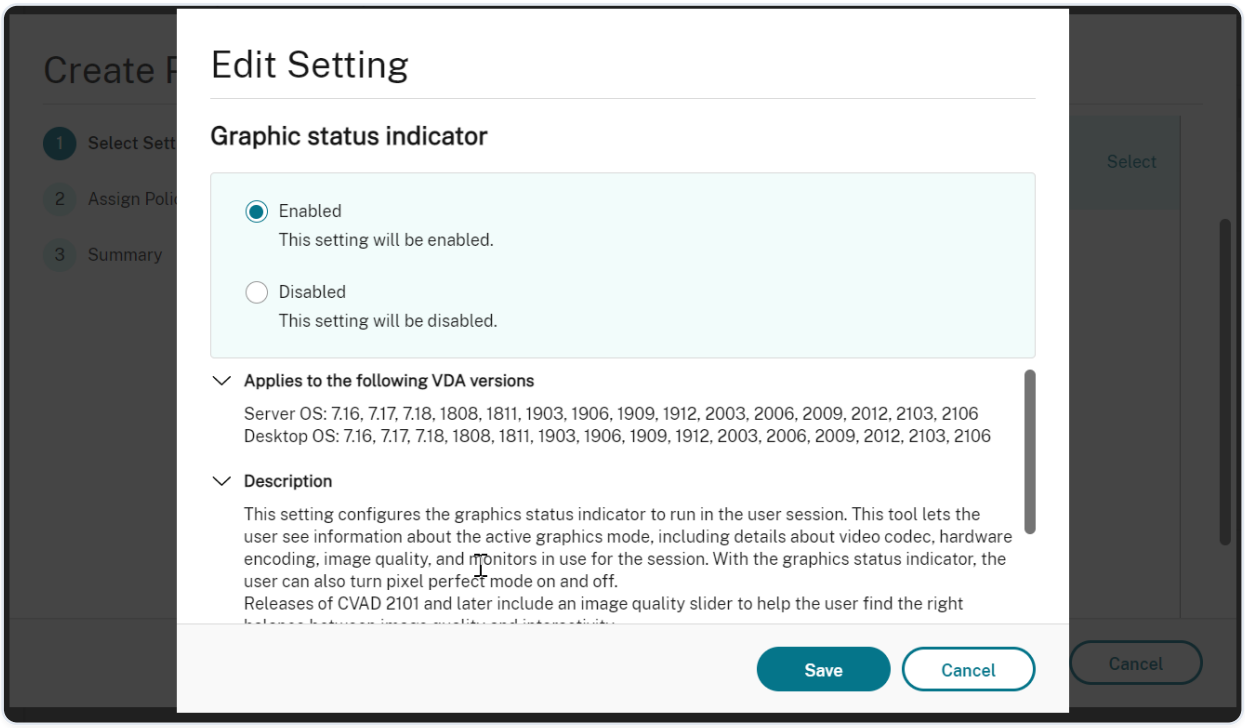
「**視覚品質**」および「**圧縮にビデオコーデックを使用**」ポリシー設定の詳細については、視覚表示ポリシー設定およびグラフィックポリシー設定を参照してください。

グラフィック品質スライダー

仮想Linuxセッションで実行されるグラフィックステータスインジケーターツールに、グラフィック品質スライダーを含めました。このスライダーは、画質とインタラクティブ性の適切なバランスを見つけるのに役立ちます。

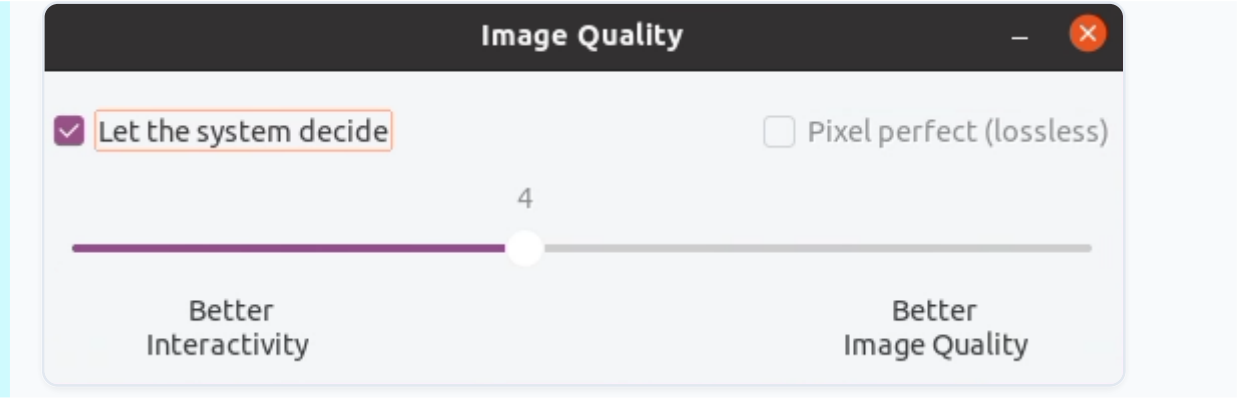
スライダーを使用するには、次の手順を完了します。

1. Citrix Studioで「**グラフィックステータスインジケーター**」ポリシーを有効にします。



2. ターミナルを開き、`ctxslider`コマンドを実行します。スライダーUIが表示されます。

> **注:**
>
> - 「**視覚品質**」ポリシーを「**常にロスレス**」または「**ロスレスに構築**」に設定している場合、スライダーUIは表示されません。
> - ターミナルとシステムトレイの両方からスライダーUIを起動できます。



次の選択肢が利用可能です。

- 画像品質を変更するには、スライダーを動かします。スライダーは0～9の範囲をサポートしています。
- システム定義の設定を使用するには、**システムに任せる**を選択します。
- ロスレスモードに切り替えるには、**ピクセルパーフェクト**を選択します。

帯域幅の見積もりに基づいて平均ビットレートを調整する

Citrixは、帯域幅の見積もりに基づいて平均ビットレートを調整することにより、HDX™ 3D Pro ハードウェアエンコーディングを強化します。

HDX 3D Proハードウェアエンコーディングが使用されている場合、VDAはネットワークの帯域幅を断続的に推定し、それに応じてエンコードされたフレームのビットレートを調整できます。この新機能は、シャープネスと流暢さのバランスを取るメカニズムを提供します。

この機能はデフォルトで有効になっています。無効にするには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "DisableReconfigureEncoder" -d "0x00000001" --force
```

この機能を使用するだけでなく、シャープネスと流暢さの間で調整するために次のコマンドを実行することもできます。**AverageBitRatePercent**と**MaxBitRatePercent**パラメーターは、帯域幅使用率のパーセンテージを設定します。設定値が高いほど、グラフィックはシャープになりますが、流暢さは低下します。推奨される設定範囲は50～100です。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "AverageBitRatePercent" -d "90" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "MaxBitRatePercent" -d "100" --force
```

平均ビットレート調整では、画面が静止している場合、新しいフレームが送信されないため、最新のフレームは低品質の状態のままになります。シャープ化サポートは、最新のフレームを最高品質で再構成してすぐに送信することにより、この問題に対処できます。

Linux VDA Thinwireでサポートされているポリシーの完全なリストについては、ポリシーサポートリストを参照してください。

Linux VDAでのマルチモニターサポートの構成については、CTX220128を参照してください。

並列処理

Thinwireは、特定のタスクを並列化することで、1秒あたりのフレーム数（FPS）を向上させることができますが、全体的なCPU消費量がわずかに増加するというオーバーヘッドがあります。この機能はデフォルトで無効になっています。この機能を有効にするには、VDAで次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"ParallelProcessing" -d "0x00000001" --force
```

トラブルシューティング

使用中のグラフィックモードを確認する

使用中のグラフィックモードを確認するには、次のコマンドを実行します（**0**はTW+、**1**は全画面ビデオコーデックを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000000" --force`

AV1が使用されていることを確認する

> **注:**
>
> 現在のセッションでどのビデオコーデックが使用されているかを確認するには、以下のコマンドを実行するか、システムトレイからグラフィックステータスを確認します。

> AV1が使用されていることを確認するには、次のコマンドを実行します（**0**は使用されていないことを意味します。**1**は使用されていることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep AV1
```

例えば、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "AV1" -d "0x00000000" --force
```

H.265が使用されていることを確認する

全画面H.265が使用されていることを確認するには、次のコマンドを実行します（**0**は使用されていないことを意味します。**1**は使用されていることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

例えば、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H265" -d "0x00000000" --force
```

H.264が使用されていることを確認する

H.264が使用されていることを確認するには、次のコマンドを実行します（**0**は使用されていないことを意味します。**1**は使用されていることを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

例えば、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000000" --force
```

どのYUVエンコーディングスキームが使用されているかを確認する

どのYUVエンコーディングスキームが使用されているかを確認するには、次のコマンドを実行します（**0**はYUV420を意味します。**1**はYUV422を意味します。**2**はYUV444を意味します）。

> **注:**
>
> **YUVFormat**の値は、ビデオコーデックが使用されている場合にのみ意味があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

例えば、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000000" --force
```

YUV444 ソフトウェアエンコーディングが使用されていることを確認する

YUV444 ソフトウェアエンコーディングが使用されていることを確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
```

YUV444 が使用されている場合、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000001" --force
```

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000001" --force
```

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000000" --force
```

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000002" --force
```

HDX 3D Pro が有効になっていることを確認する

HDX 3D Pro が有効になっていることを確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep ProductEdition



sudo /opt/Citrix/VDA/bin/ctxreg dump | grep StackSessionMode



sudo /opt/Citrix/VDA/bin/ctxreg dump | grep 3DPro
```

HDX 3D Pro が有効になっている場合、結果は次のようになります。

`create -k "HKLM\Software\Citrix\VirtualDesktopAgent\State" -t "REG_SZ" -v "ProductEdition" -d "<PLT or ENT>" --force`

`create -k "HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t "REG_DWORD" -v "StackSessionMode" -d "0x00000000" --force`

`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "3DPro" -d "0x00000000" --force`

HDX 3D Pro に必要な NVIDIA ライブラリがロードされていることを確認するには、Linux VDA で **nvidia-smi** コマンドを実行します。結果は次のようになります。

```
Tue Apr 12 10:42:03 2016

+------------------------------------------------------+

| NVIDIA-SMI 361.28     Driver Version: 361.28         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
==============================+======================+==================
===|
```

```
|   0  GRID K1             Off  | 0000:00:05.0     Off |
N/A |

| N/A   42C    P0    14W /  31W |    207MiB /  4095MiB |      8%
Default |

+-------------------------------+----------------------
+----------------------+



+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|

|    0      2164  C+G   /usr/local/bin/ctxgfx
106MiB |

|    0      2187    G   Xorg
85MiB |

+-----------------------------------------------------------------------------
----+
```

3D Pro でハードウェアエンコーディングが使用されていることを確認する

次のコマンドを実行します (**0** は使用されていないことを意味します。**1** は使用中であることを意味します):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
```

3D Pro が使用されている場合、結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000001" --force`

NVIDIA GRID グラフィックドライバーが正しくインストールされていることを確認する

NVIDIA GRID グラフィックドライバーが正しくインストールされていることを確認するには、**nvidia-smi** を実行します。結果は次のようになります。

```
+-----------------------------------------------------+

| NVIDIA-SMI 352.70     Driver Version: 352.70         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
===============================+======================+==================
===|

|   0  Tesla M60           Off  | 0000:00:05.0     Off |
Off |

| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%
Default |

+-------------------------------+----------------------
+----------------------+



+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |
```

```
|
==============================================================================
===|

|  No running processes found
                                |

+------------------------------------------------------------------------------
----+
```

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

HDX 3D Pro マルチモニターの再描画の問題

プライマリモニター以外の画面で再描画の問題が発生している場合は、NVIDIA GRID ライセンスが利用可能であることを確認してください。

Xorg エラーログを確認する

Xorg のログファイルは、**/var/log/** フォルダー内に **Xorg.{DISPLAY}.log** のような名前で保存されています。

既知の問題と制限事項

vGPU の場合、XenServer (旧 Citrix Hypervisor™) のローカルコンソールに ICA デスクトップセッション画面が表示されます

回避策: 次のコマンドを実行して、VM のローカル VGA コンソールを無効にします。

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

Gnome 3 デスクトップのポップアップがログオン時に遅い

これは Gnome 3 デスクトップセッションの起動における制限です。

Citrix Workspace アプリのウィンドウサイズを変更すると、一部の OpenGL/WebGL アプリケーションが適切にレンダリングされない

Citrix Workspace アプリのウィンドウサイズを変更すると、画面解像度が変更されます。NVIDIA 独自のドライバーは一部の内部状態を変更し、アプリケーションがそれに応じて応答する必要がある場合があります。たとえば、WebGL ライブラリ要素の **lightgl.js** は、「**このテクスチャへの**

レンダリングはサポートされていません (不完全なフレームバッファー)」というエラーを生成する可能性があります。

HDX™ 画面共有

概要

Linux VDAを使用すると、仮想デスクトップの画面を他の仮想デスクトップのセッションユーザーと共有できます。

注:

- 複数のLinux VDA間でセッション共有を行う場合、キーボード入力の問題を回避するために、同じVDAバージョンにアップグレードすることをお勧めします。
- ビューアーのセッションと共有者のセッションで異なるキーボードレイアウトが存在する場合、共有者のキーボードレイアウトが有効になります。
- Unicodeモードを使用すると制限が多いため、ビューアーのCitrix Workspace™アプリではスキャンコードモードを使用することをお勧めします。
- Citrix WorkspaceアプリクライアントのUnicodeモードでは、一部の文字が認識されない可能性があります。
- Unicodeモードを使用する場合は、混在した出力にならないように、共有者とビューアーが同じキーボードレイアウトを使用することをお勧めします。
- ショートカットキーの組み合わせがビューアーと共有者の両方でアクティブな場合、それらはビューアーでのみ有効になります。
- ショートカットキーにSuperキーが含まれている場合、それはビューアーで有効になり、共有者にも送信されます。

次の例では、画面を共有し、他のユーザーの画面を表示する手順を説明します。

画面を共有するには：

1. 仮想デスクトップの通知領域で、次のシステムトレイアイコンをクリックし、**画面共有 > 自分の画面を共有**を選択します。

   自分の画面を共有メニューを選択します(/en-us/linux-virtual-delivery-agent/2411/media/select-the-share-my-screen-menu.png)

自分の画面を共有メニューを選択します(/en-us/linux-virtual-delivery-agent/2411/media/select-the-share-my-screen-menu-2.png)

2. **コピーして閉じる**をクリックします。

   現在の画面共有コードは、画面共有を停止して再開するまで保持されます。

   接続コードをコピーします。(/en-us/linux-virtual-delivery-agent/2411/media/copy-the-connection-code.png)

   > **ヒント:**
   >
   > 画面を共有している間は、画面の周囲に赤い枠が表示され、共有中であることを示します。

3. 画面を共有したい他の仮想デスクトップのセッションユーザーと、コピーしたコードを共有します。
4. 閲覧者に画面を制御させるには、**Give control**を選択し、閲覧者の名前を選択します。制御を停止するには、閲覧者の名前の選択を解除します。

   制御を許可します。(/en-us/linux-virtual-delivery-agent/2411/media/give-control.png)

5. 画面共有を停止するには、**Stop sharing my screen**を選択します。

   メニューを選択します。(/en-us/linux-virtual-delivery-agent/2411/media/select-the-stop-sharing-my-screen-menu.png)

他のユーザーの画面を表示するには：

1. 仮想デスクトップの通知領域で、**screen sharing**アイコンをクリックし、**View someone else's screen**を選択します。

   メニューを選択します。(/en-us/linux-virtual-delivery-agent/2411/media/select-the-view-someone-else-s-screen-menu.png)

2. 表示したい画面の接続コードを入力し、**Connect**をクリックします。

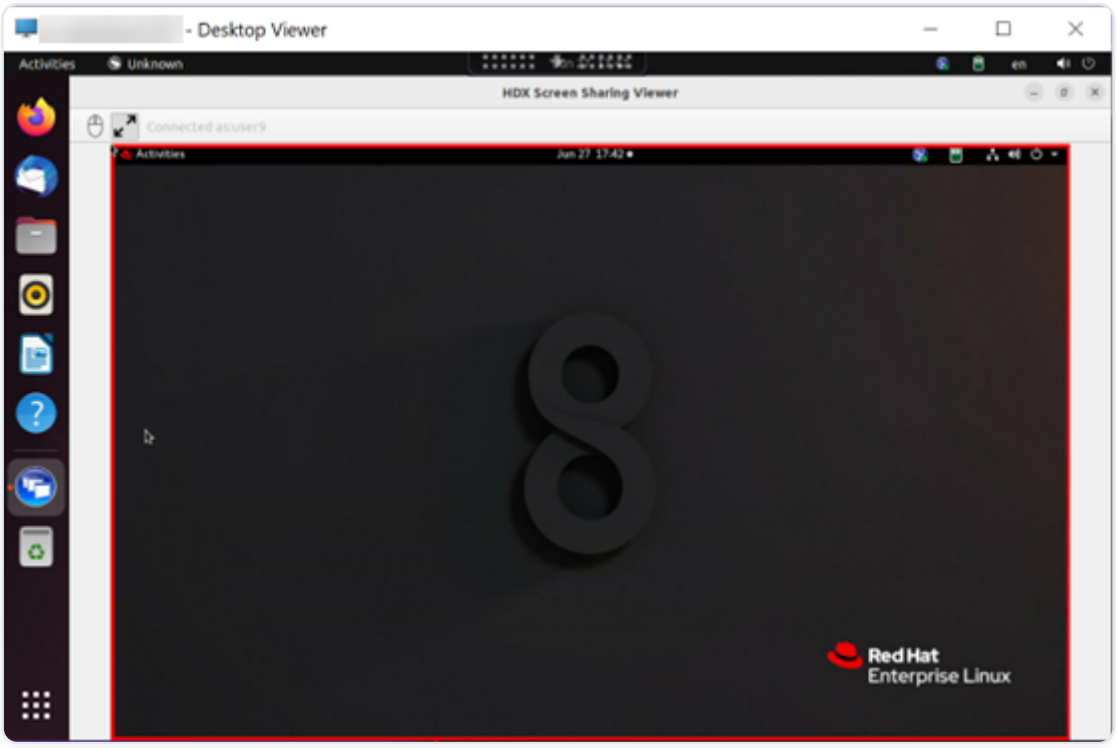
   接続コードを入力します。(/en-us/linux-virtual-delivery-agent/2411/media/enter-the-connection-code.png)

3. 画面共有者がリクエストを承認するまで待ちます。例：

画面共有を承認または拒否します。(/en-us/linux-virtual-delivery-agent/2411/media/accept-or-deny-screen-sharing.png)

> **ヒント:**
>
> - 共有者側では、Linuxシステムがリクエストの通知を発行します。
> - 共有者が30秒以内にリクエストを承認しない場合、リクエストは期限切れになり、プロンプトが表示されます。

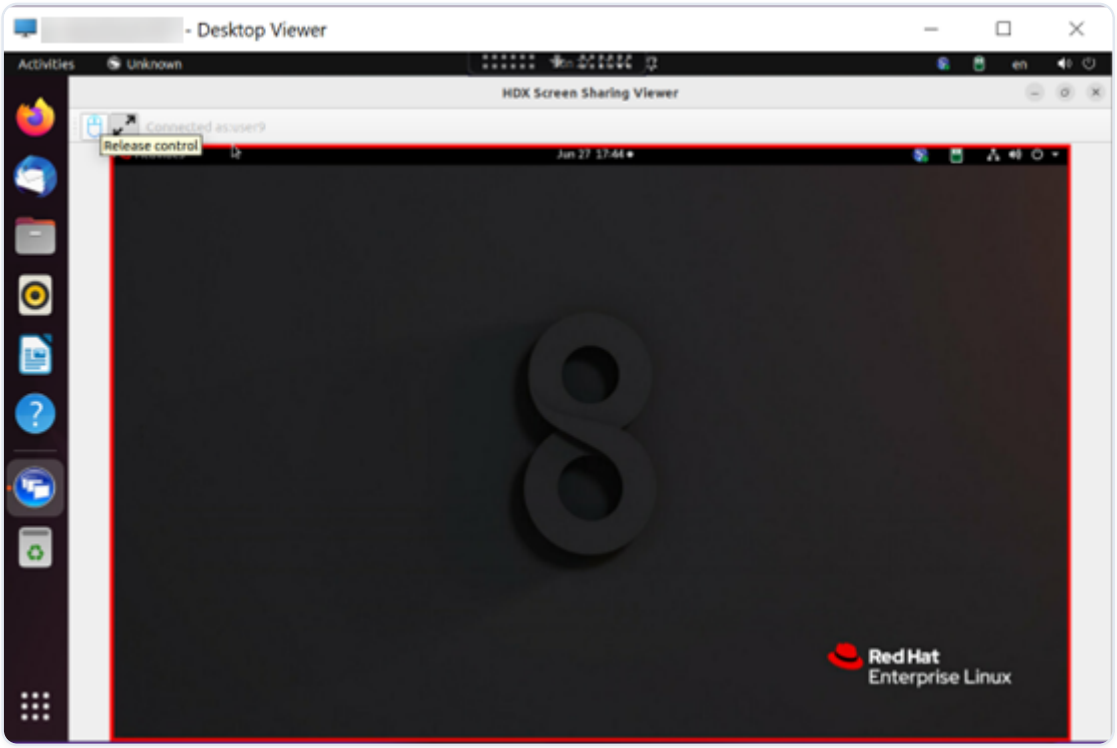
4. 画面共有者が**OK**をクリックしてリクエストを承認すると、共有画面がDesktop Viewerに表示されます。自動的に割り当てられたユーザー名でビューアとして接続されます。



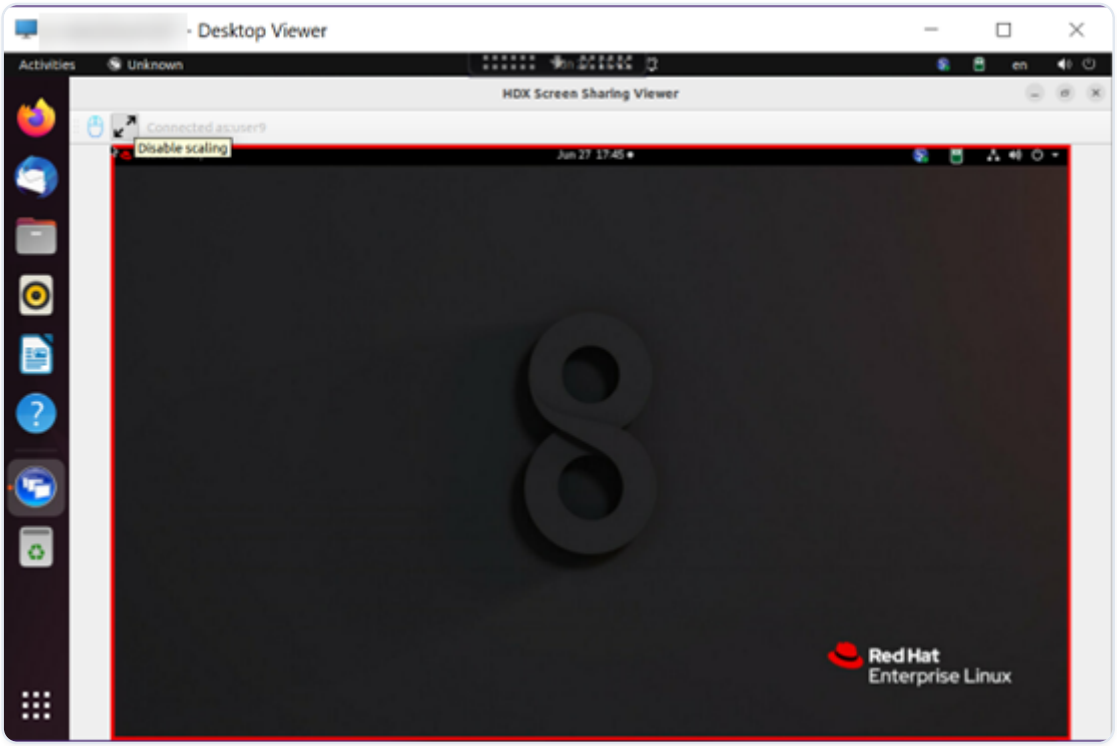
5. 共有画面の制御を要求するには、左上隅にあるマウスアイコンをクリックします。

> **ヒント:**
>
> - 共有者が30秒以内にリクエストを承認しない場合、リクエストは期限切れになります。
> - 一度に共有画面を制御できるビューアは1人だけです。

> 共有画面の制御を解除するには、マウスアイコンをもう一度クリックします。



6. 表示スケーリングを無効にするか、ウィンドウサイズに合わせてスケーリングするには、マウスアイコンの隣にあるアイコンをクリックします。



構成

画面共有機能はデフォルトで無効になっています。有効にするには、次の設定を完了してください。

1. システムトレイを有効にする。
2. Citrix Virtual Apps and Desktops™ 2112以降の場合、Citrix Studioで**ScreenSharing**ポリシーを有効にします。
3. (オプション) Citrix Virtual Apps™ and Desktops 2109以前の場合、次のコマンドを実行してLinux VDAで画面共有を有効にします。

   ```
   sudo /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v
   "EnableScreenSharing" -d "0x00000001"
   ```

4. ファイアウォールでポート52525～52625を許可します。

考慮事項

- 画面共有機能はH.265ビデオコーデックをサポートしていません。
- 画面共有機能はアプリセッションでは利用できません。
- デスクトップセッションのユーザーは、デフォルトで最大10人の視聴者とセッション画面を共有できます。最大視聴者数は`ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v "ScreenSharingViewerMaxNum" -d <hex_value>`を介して構成可能です。最大数に達すると、ユーザーが追加の接続要求を受け入れようとしたときにプロンプトが表示されます。

マルチモニターのサポート

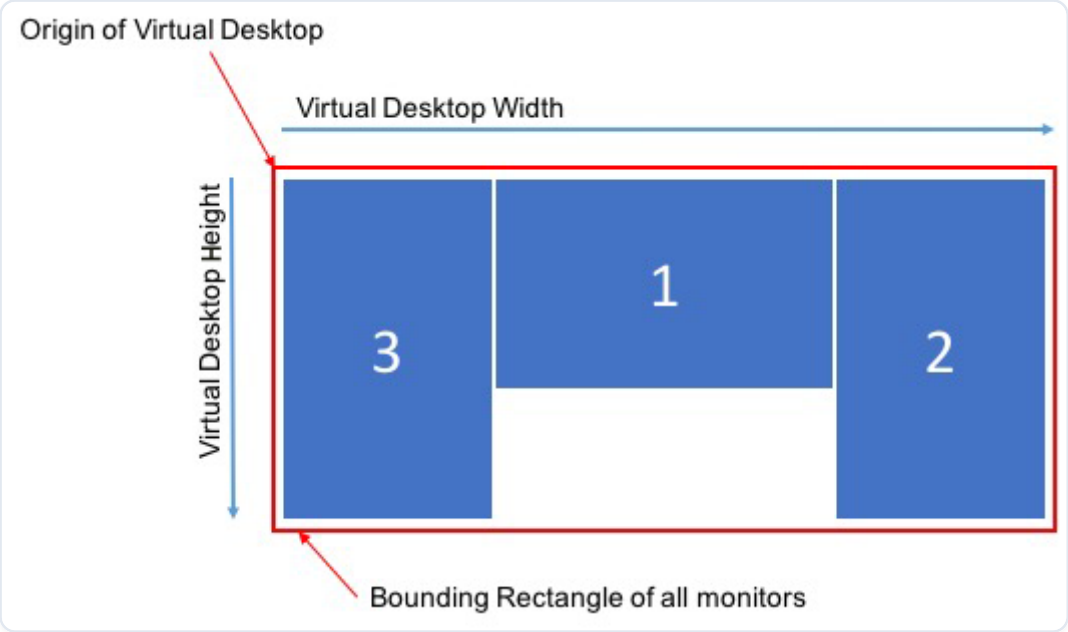
概要

Linux VDAは、モニターあたり2560×1600のデフォルト解像度で、標準でマルチモニターをサポートします。標準VDAは最大9台のモニターをサポートし、HDX™ 3D Pro VDAは最大4台のモニターをサポートします。

この記事では、さまざまなモニター解像度とレイアウトに対応するようにLinux VDAを構成する方法について説明します。

仮想セッションデスクトップ

Windows VDAと同様に、Linux VDAにはマルチモニター仮想デスクトップの概念があります。これは、すべてのモニターの境界矩形に基づいており、モニターの実際のレイアウトに基づいているわけではありません。したがって、仮想デスクトップの領域は、理論的にはクライアントのモニターがカバーする領域よりも大きくなる可能性があります。



仮想セッションデスクトップのサイズ

仮想セッションデスクトップの原点は、すべてのモニターの境界矩形の左上隅から計算されます。その点はX = 0、Y = 0に位置し、XとYはそれぞれ水平軸と垂直軸です。

仮想セッションデスクトップの幅は、原点からすべてのモニターの境界矩形の右上隅までの水平距離（ピクセル単位）です。

同様に、仮想セッションデスクトップの高さは、原点からすべてのモニターの境界矩形の左下隅までの垂直距離（ピクセル単位）です。

この計算は、次の理由から重要です。

- さまざまなクライアントモニターレイアウトに対応するため
- Linux VDAでのメモリ使用量を理解するため

さまざまなクライアントモニター構成に対応する

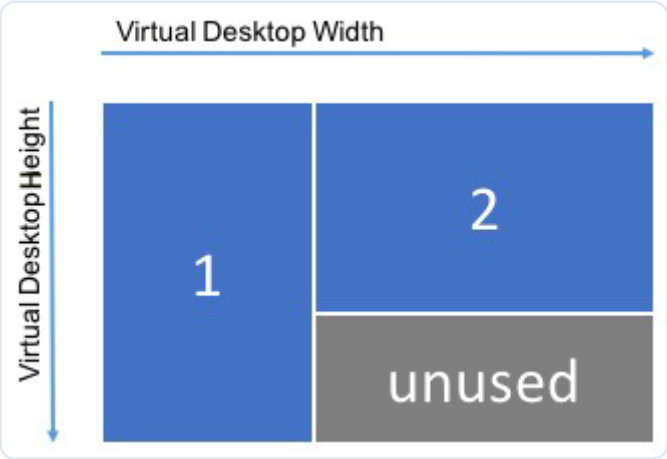
さまざまなクライアントモニター構成における最大仮想デスクトップサイズを知ることで、クライアントモニター構成に関して柔軟なLinux VDAを構成できます。

次のクライアントモニター構成を検討してください。



上の図は、それぞれ2560×1600の解像度を持つ2つのモニターを備えた、既製のマルチモニター構成を示しています。

次に、以下のクライアントモニター構成で同じLinux VDAに接続することを検討します。



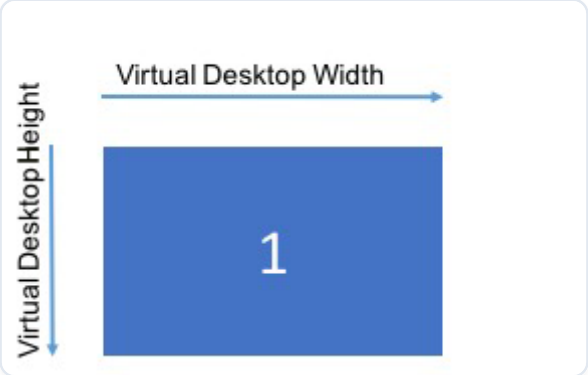
上の図の各モニターの解像度が2560×1600の場合、既製のマルチモニター構成パラメーターでは不十分です。このモニターレイアウトの仮想セッションデスクトップを収容するには、最大高さが小さすぎます。この例のクライアントモニター構成に対応するには、Linux VDA仮想デスクトップのサイズを4160×2560に設定する必要があります。

マルチモニター構成で最大の柔軟性を得るには、サポートしたいすべてのモニターレイアウトの最小外接矩形を見つけます。2つの2560×1600モニターを持つ構成の場合、可能なレイアウトは次のとおりです。

- **Monitor1** 2560×1600 および **Monitor2** 2560×1600
- **Monitor1** 1600×2560 および **Monitor2** 2560×1600
- **Monitor1** 2560×1600 および **Monitor2** 1600×2560
- **Monitor1** 1600×2560 および **Monitor2** 1600×2560

上記のすべてのレイアウトに対応するには、5120×2560の仮想セッションデスクトップが必要です。これは、目的のすべてのレイアウトを収容できる最小の外接矩形です。

すべてのユーザーが一般的な横向きレイアウトで1つのモニターしか持っていない場合は、最大仮想デスクトップサイズをモニターの最高解像度に設定します。



この例では、仮想デスクトップのサイズを2560×1600に設定する必要があります。デフォルト構成は5120×1600でモニター2台であるため、シングルモニター展開のメモリ使用量を最適化するには構成変更が必要です。

> **注:**
>
> マルチモニター設定でデスクトップが不適切な解像度で表示される場合は、Citrix WorkspaceアプリでDPI（Dots Per Inch）設定を調整してください。詳細については、Knowledge Centerの記事CTX230017を参照してください。

Linux VDAでのメモリ使用量の理解

仮想デスクトップのサイズを知ることで、各HDXセッションで使用されるメモリ量を計算できます。このメモリは、セッション開始時にグラフィックデータのために各セッションに割り当てられるメモリです。セッションの存続期間中、このメモリは変化しません。このメモリはセッションで使用されるメモリの総量ではありませんが、セッションごとのメモリ使用量を計算する最も簡単な方法です。

各HDXセッションに割り当てられるメモリ量を計算するには、次の式を使用します。

M = X × Y × Z であり、,

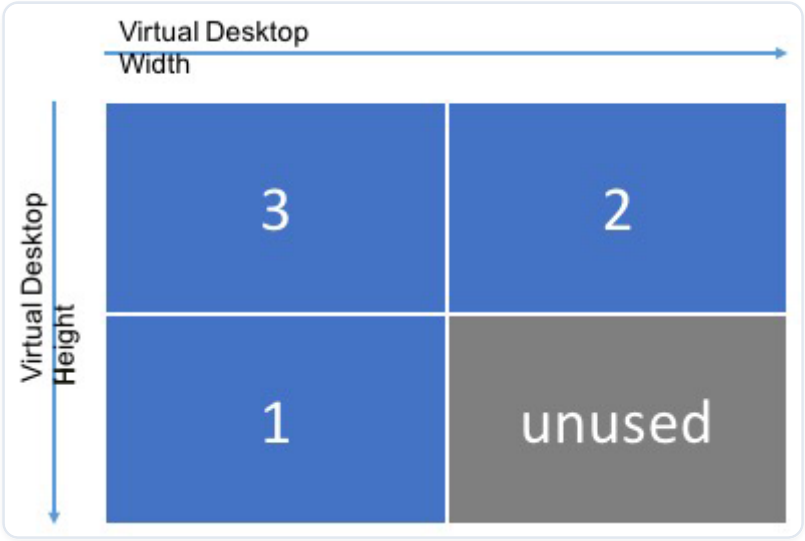
ここで：

- **M** はセッショングラフィックに使用されるメモリ量です。
- **X** は仮想セッションデスクトップの幅です。
- **Y** は仮想セッションデスクトップの高さです。

- **Z** はHDXセッションウィンドウのカラー深度です。値はビットではなくバイト単位なので、32ビットカラーの場合は4を使用します。

> **注：**
>
> Xサーバーのカラー深度はセッションの存続期間中（**ログインから切断/再接続を経てログオフまで**）変更できません。したがって、Linux VDAは常に仮想セッションデスクトップを32ビットとして割り当て、セッションで要求されたカラー深度にダウンサンプリングします。
>
> たとえば、1024×768セッションの場合、使用されるメモリは次のとおりです。
>
> 1024 × 768 × 4 / 2\^20 メガバイト = 3 メガバイト
>
> メモリ使用量を理解することは、各Linux VDAでのセッション密度を高める上で重要です。

次のクライアントモニター構成を検討してください。



各モニターの解像度が2560×1600の場合、このクライアントモニター構成に対応するには、仮想セッションデスクトップのサイズは5120×3200である必要があります。灰色の領域は未使用であり、16,384,000バイト（つまり、2560 x 1600 x 4）の無駄なメモリに相当することに注意してください。

Citrix® マルチモニター構成パラメーター

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`

  パラメーター: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFbWidth

  説明: 仮想セッションデスクトップの最大幅

  種類: DWORD

  デフォルト: 5,120

  最大: 16,384 (8,192 x 2)

- `MaxFbHeight`

  パラメーター: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFbHeight

  説明: 仮想セッションデスクトップの最大高さ

  種類: DWORD

  デフォルト: 1,600

  最大値: 16,384 (8,192 x 2)

Linux VDAのマルチモニター構成を変更する

以下のセクションでは、Linux VDAでマルチモニター機能を有効化、構成、および無効化する方法について説明します。

以下の方法でモニターの最大数を設定します:

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxScreenNum" -d "NumMons" --force
```

`NumMons` は、標準VDAの場合は1～9、HDX 3D Pro VDAの場合は1～4の間の値です。

以下の方法で仮想セッションデスクトップの最大幅を設定します:

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxFbWidth" -d " MaxWidth" --force
```

`MaxWidth` は、`1,024` と `16,384` の間の値です。

以下の方法で仮想セッションデスクトップの最大高さを設定します:

```
sudo ctxreg create -k " HKEY_LOCAL_MACHINE
\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"MaxFbHeight" -d " MaxHeight" --force
```

`MaxHeight` は、`1,024` と `16,384` の間の値です。

非仮想化GPU

Linux VDAドキュメントでは、非仮想化GPUは以下を指します。

- リモートPCアクセスシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

この記事では、非仮想化GPUのサポートに関する情報を提供します。

NVIDIA Capture SDK for Linuxをサポートするnvidia GPUでHDX™ 3D Proを有効にする

NVIDIA Capture SDK for Linuxをサポートするnvidia GPUの場合、Linux VDAのインストール時に**CTX_XDL_HDX_3D_PRO**を**Y**に設定するだけでHDX 3D Proを有効にできます。追加の設定は不要です。HDX 3D Proを有効にすると、ハードウェアアクセラレーションはデフォルトで有効になります。

NVIDIA Capture SDK for Linuxをサポートしないnvidia GPU、およびAMDやIntelなどの他のメーカーのGPUとの互換性

> **注:**
>
> このシナリオでは、ソフトウェアエンコーディングのみがサポートされます。

ステップ1：Linux VDAのインストール時にCTX_XDL_HDX_3D_PROをYに設定する

環境変数については、「ステップ8：環境とVDAを構成してインストールを完了するための簡易インストールを実行する」を参照してください。

ステップ2：Xdamageをインストールする

たとえば、Ubuntu 20.04にXDamageをインストールするには、**sudo apt-get install -y libxdamage1**を実行します。通常、XDamageはXServerの拡張機能として存在します。

ステップ3：次のコマンドを実行してXDamageを有効にする

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

ステップ4：Xorg構成ファイルを変更する

/etc/X11 の下に、次の4つのテンプレート構成ファイルがあります。接続されているモニターの数に基づいて、名前に対応する番号が含まれるテンプレート構成ファイルのいずれかを変更します。たとえば、モニターが1台しか接続されていない場合は、名前に1が含まれるテンプレート構成ファイル、つまり ctx-driver_name-1.conf を変更します。モニターが2台接続されている場合は、名前に2が含まれるテンプレート構成ファイル、つまり ctx-driver_name-2.conf を変更します。

- ctx-driver_name-1.conf
- ctx-driver_name-2.conf
- ctx-driver_name-3.conf
- ctx-driver_name-4.conf

ctx-driver_name-1.conf を例として、テンプレート構成ファイルを変更するには、次の手順を実行します。

1. ***driver_name*** を実際のドライバー名に置き換えます。

たとえば、ドライバー名が `intel` の場合、構成ファイル名を `ctx-intel-1.conf` に変更できます。

2. ビデオドライバー情報を追加します。

   各テンプレート構成ファイルには、「Device」という名前のセクションが含まれており、コメントアウトされています。このセクションでは、ビデオドライバー情報について説明しています。ビデオドライバー情報を追加する前に、このセクションを有効にします。このセクションを有効にするには、次の手順を実行します。

   a. 構成情報については、GPUメーカーが提供するガイドを参照してください。ネイティブ構成ファイルを生成できます。ネイティブ構成ファイルを使用して、GPUがローカル環境で動作することを確認してください。

   b. ネイティブ構成ファイルの「Device」セクションを **ctx-driver_name-1.conf** にコピーします。

3. 次のコマンドを実行してレジストリキーを設定し、Linux VDAが手順1で設定した構成ファイル名を認識できるようにします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_SZ" -v "DriverName" -d "intel" --force
```

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とDebian 11が含まれます。

この機能はデフォルトで無効になっています。有効にするには、次の2つの手順を完了します。

1. お使いのLinuxディストリビューションに基づいて、`evdi-dkms`パッケージをインストールします。

```
sudo apt install evdi-dkms
```

2. グラフィックディスプレイのオフロードをEVDIに有効にします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "Evdi" -d "0x00000001" --force
```

3. Intel GPUを使用している場合は、ディスプレイマネージャーを無効にします。そうしないと、ディスプレイマネージャーがIntel GPUを占有し、Citrix®リモートセッションで利用できなくなります。

```
sudo systemctl disable --now gdm
```

トラブルシューティング

グラフィック出力がない、または文字化けしている

3Dアプリケーションをローカルで実行でき、すべての構成が正しい場合、グラフィック出力の欠落または文字化けはバグの結果です。デバッグ用のトレース情報を収集するには、/opt/Citrix/VDA/bin/setlogを使用し、GFX_X11をverboseに設定します。

ルートレス Xorg

Linux VDA は、非ルートユーザー権限で Xorg を実行することをサポートしています。これは「ルートレス」Xorg とも呼ばれます。ルートレス Xorg は、root として実行する場合と比較して、セキュリティが大幅に向上します。

注:

- Xorg は、Amazon Linux 2 または RHEL 7 ではルートレス操作をサポートしていません。
- ルートレス Xorg は、GPU ドライバーからのサポートも必要とします。Remote PC Access VDA を使用している場合は、GPU ドライバーがルートレス Xorg をサポートしていることをテストして確認してください。

ルートレス Xorg は、GPU を搭載していない VDA および NVIDIA GPU を使用する VDA でデフォルトで有効になっています。この機能は、以下のコマンドを使用して手動で有効または無効にできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "1" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"Rootless" -d "0" --force
```

セッションウォーターマーク

セッションウォーターマークは、データ窃盗の抑止と追跡を可能にします。データ窃盗のために写真やスクリーンキャプチャを使用するユーザーへの抑止力として、追跡可能な情報がセッションデスクトップに表示されます。ウォーターマークは、テキストレイヤーまたはアルファチャネル付きのPNG画像として指定できます。ウォーターマークは、元のドキュメントの内容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。データ窃盗を完全に防ぐものではありませんが、ある程度の抑止力と追跡可能性を提供します。この機能を使用した場合の完全な情報追跡可能性は保証しません。代わりに、この機能を必要に応じて他のセキュリティソリューションと組み合わせて使用することをお勧めします。

セッションウォーターマークには、データ窃盗を追跡するための情報が含まれています。最も重要なデータは、スクリーンイメージが取得されたセッションのユーザーのIDであり、これはログオン資格情報によって追跡されます。データ漏洩をより効果的に追跡するには、サーバーまたはクライアントのインターネットプロトコルアドレスや接続時間などの他の情報を含めます。

ユーザーエクスペリエンスを調整するには、画面上のウォーターマークの配置と表示を構成するために、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ます。

セッションウォーターマークポリシー設定

セッションウォーターマークを有効にする

この設定を有効にすると、セッション表示には、セッション固有の情報を表示する不透明なウォーターマークが表示されます。他のウォーターマーク設定は、この設定が有効になっているかどうかに依存します。

デフォルトでは、セッションウォーターマークは無効になっています。

クライアントIPアドレスを含める

この設定を有効にすると、セッションは現在のクライアントIPアドレスをウォーターマークとして表示します。

デフォルトでは、**クライアントIPアドレスを含める**は無効になっています。

接続時間を含める

この設定を有効にすると、セッションウォーターマークに接続時間が表示されます。形式はyyyy/mm/dd hh:mmです。表示される時刻は、システムクロックとタイムゾーンに基づいています。

デフォルトでは、**接続時間を含める**は無効になっています。

ログオンユーザー名を含める

この設定を有効にすると、セッションは現在のログオンユーザー名をウォーターマークとして表示します。表示形式はUSERNAME@DOMAINNAMEです。ユーザー名は最大20文字にすることをお勧めします。ユーザー名が20文字を超えると、フォントサイズが小さくなったり、切り捨てられたりする可能性があり、ウォーターマークの効果が低下します。

デフォルトでは、**ログオンユーザー名を含める**は有効になっています。

VDAホスト名を含める

この設定を有効にすると、セッションは現在のICA®セッションのVDAホスト名をウォーターマークとして表示します。

デフォルトでは、**VDAホスト名を含める**は有効になっています。

VDA IPアドレスを含める

この設定を有効にすると、セッションは現在のICAセッションのVDA IPアドレスをウォーターマークとして表示します。

デフォルトでは、**VDA IPアドレスを含める**は無効になっています。

セッションウォーターマークのスタイル

この設定は、単一のウォーターマークテキストラベルを表示するか、複数のラベルを表示するかを制御します。**値**ドロップダウンメニューから**複数**または**単一**を選択します。 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**」は、セッションに5つのウォーターマークラベルを表示します。中央に1つ、四隅に4つです。

「**単一**」は、セッションの中央に単一のウォーターマークラベルを表示します。

デフォルトでは、**セッションウォーターマークスタイル**は**複数**です。

透かしの透明度

ウォーターマークの不透明度を0から100まで指定できます。指定する値が大きいほど、ウォーターマークは不透明になります。

デフォルトでは、値は17です。

ウォーターマークのカスタムテキスト

デフォルトでは値は空です。空ではない文字列を入力するか、文字列を形成する構文を設定するか、またはその組み合わせを使用してセッションウォーターマークに表示できます。空ではない文字列は、1行あたり最大25個のUnicode文字をサポートします。25文字を超える文字列は切り捨てられます。

例えば、ポリシーを次の値に設定できます。

```
<date>
<time><newline><username><style=single><fontsize=40><font=Ubuntu><position=center><rotation
Linux VDA<newline>Version 2207
```

すべての構文オプションの説明については、次の表を参照してください。

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<style>`	ウォーターマークのレイアウトスタイル	`xstyle`、`single`、`tile`、`horizontal`	`xstyle`	-

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<position>`	ウォーターマークの位置	`center`, `topleft`, `topright`, `bottomleft`, `bottomright`	`center`	レイアウトスタイルが**single**に設定されている場合にのみ有効です。
`<rotation>`	ウォーターマークの特定の角度への回転	-180–180	0	-
`<transparency>`	ウォーターマークの不透明度	0–100	17	-
`<font>`	-	システムでサポートされているフォント	サン	-
`<fontsize>`	-	20〜50	0 (自動計算)	-
`<fontzoom>`	`<fontsize>` および `<image>` で設定したフォントサイズと画像サイズの割合	0 –	100	-
`<image>`	PNGウォーターマーク	VDA上のPNG画像へのパス	該当なし	この構文はPNGウォーターマークを構成します。アルファチャネルを持つPNGのみがサポートされています。PNGウォーターマークを使用している場合、`<style>`、`<position>`

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
				、`<rotation>`、`<transparency>`、および`<fontzoom>`の構文オプションのみが有効になります。
`<date>`	セッション接続日 (YYYY/MM/DD) のプレースホルダー	該当なし	該当なし	-
`<time>`	セッション接続時刻 (HH:MM) のプレースホルダー	該当なし	該当なし	-
`<domain>`	ユーザーアカウントドメインのプレースホルダー	該当なし	該当なし	-
`<username>`	現在のログオンユーザー名 (ユーザーアカウントドメインを除く) のプレースホルダー	該当なし	該当なし	-
`<hostname>`	VDAのホスト名のプレースホルダー	該当なし	該当なし	-
`<clientip>`	クライアントのIPアドレスのプレースホルダー	該当なし	該当なし	-

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<serverip>`	VDAのIPアドレスのプレースホルダー	該当なし	該当なし	-

注:

ウォーターマークカスタムテキストが少なくとも1つの有効な構文オプションで指定されている場合、**セッションウォーターマークを有効にする**を除く他のすべてのセッションウォーターマークポリシーは無視されます。

構文オプションを指定しないか、サポートされていない値に設定した場合、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、以下のいずれかのケースでサポートされます。
  - **ビデオコーデックを圧縮に使用**が**画面全体**に設定されている場合。
  - **ビデオコーデックを圧縮に使用**が**優先する場合に使用**に設定されており、3Dグラフィックワークロード向けに最適化が有効になっている場合。
- ブラウザコンテンツリダイレクトが使用されているセッションでは、セッションウォーターマークはサポートされません。セッションウォーターマーク機能を使用するには、ブラウザコンテンツリダイレクトが無効になっていることを確認してください。
- レガシーNVIDIAドライバーを使用し、セッションが全画面ハードウェアアクセラレーションH.264またはH.265エンコードモードで実行されている場合、セッションウォーターマークはサポートされず、表示されません。(この場合、レジストリでNvCaptureTypeは2に設定されています。)
- セッションシャドウイングではウォーターマークは表示されません。
- Print Screenキーを押して画面をキャプチャした場合、VDA側でキャプチャされた画面にはウォーターマークは含まれません。画面キャプチャがコピーされないように対策を講じることをお勧めします。

マルチセッションLinux VDAでの共有GPUアクセラレーション

HDX™ 3D PROは、VDIデスクトップ（シングルセッションモード）用に構成されたLinux VDAのみをサポートします。マルチセッションLinux VDAの場合、OpenGL 3Dアプリケーションを高速化するために、共有GPUアクセラレーションを有効にできます。

> **注:**
>
> Waylandディスプレイサーバーは、共有GPUアクセラレーションではサポートされていません。

構成

マルチセッションLinux VDAで共有GPUアクセラレーションを有効にしてOpenGL 3Dアプリケーションを高速化するには、以下の構成手順を完了してください。

ステップ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には**.deb**パッケージを、RHELベースのLinuxディストリビューションには**.rpm**パッケージをダウンロードしてください。

ステップ2：VirtualGLの構成

1. Linuxディスプレイマネージャー（例：LightDMまたはGNOME Display Manager (GDM)）を停止します。
2. 次のコマンドを実行して、VirtualGL構成スクリプトを実行します。

   ```
   #/opt/VirtualGL/bin/vglserver_config
   ```

   スクリプトの実行中に、以下の選択を行うことをお勧めします。

   - 「VirtualGLで使用するサーバーを構成する (GLX + EGLバックエンド)」には「1」を選択します。
   - 「3D Xサーバーへのアクセスを**vglusers**グループに制限する」には「n」を選択します。
   - 「フレームバッファデバイスへのアクセスを**vglusers**グループに制限する」には「n」を選択します。
   - 「XTEST拡張を無効にする」には「n」を選択します。

3. 構成スクリプトを終了し、Linuxディスプレイマネージャーを再起動します。

ステップ 3: GPU アクセラレーションを使用して OpenGL 3D アプリケーションを実行する

Linux VDA セッションで GPU アクセラレーションを使用して OpenGL 3D アプリケーションを実行するには、2 つの方法があります。

- **方法 1:** すべての OpenGL 3D アプリケーションで共有 GPU アクセラレーションを有効にする

  これを行うには、Linux VDA で bash ターミナルを開き、次のコマンドを実行してから、bash ターミナルを再起動します。このアプローチにより、bash ターミナルから起動されたすべての OpenGL 3D アプリケーションで共有 GPU アクセラレーションが有効になります。

  ```
  #/opt/Citrix/VDA/sbin/ctxgpushare.sh enable
  ```

- **方法 2:** 特定の OpenGL 3D アプリケーションで共有 GPU アクセラレーションを有効にする:

  これを行うには、Linux VDA でターミナルを開き、アプリケーション名を指定して次のコマンドを実行します。

  ```
  #vglrun <AppName>
  ```

制限事項

- 共有 GPU アクセラレーションは、Linux VDA 上のディスプレイマネージャーと密接に連携します。期待される機能とパフォーマンスを実現するには、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ています。

スケーラビリティ

GPU を共有できる同時セッションの最大サポート数は、CPU とシステムメモリによって異なります。また、GPU の最大ビデオメモリにも大きく依存します。

例:

もし	ならば
NVIDIA **M10-2B** vGPU が 2,048 MB のビデオメモリを持ち、VariCAD ビューアなどの OpenGL アプリケーションが各セッションで	理論上、同時セッションの最大サポート数は 20 を超えることはできません。

もし	ならば
そのワークロードに 100 MB のビデオメモリを使用する場合、,	

システムトレイ

セッションユーザーは、次のシステムトレイアイコンをクリックして、アクションを実行したり、インジケーターを表示したりできます。

システムトレイアイコン(/en-us/linux-virtual-delivery-agent/2411/media/system-tray-icon.png)

システムトレイの項目について

各項目は、トグル付きの機能に対応しています。項目に対応する機能が無効になっている場合、その項目は非表示になり、表示されません。

- **画面共有**

  この機能の詳細については、「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**

  この項目は、**ctxdesktopswitch.sh** の GUI です。詳細については、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

  セッションユーザーによるデスクトップ環境のカスタマイズは、デフォルトで有効になっています。無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k
  "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
  -v "EnableDesktopSwitch" -d "0x00000000" --force
  ```

- **グラフィック品質スライダー**

  詳細については、グラフィック構成の記事の「グラフィック品質スライダー」セクションを参照してください。

- **リアルタイム CQI**

  現在、ICA ラウンドトリップタイム (RTT) と遅延データが表示されます。詳細については、「セッションメトリッククエリユーティリティ」を参照してください。

  システムトレイアイコンは、リアルタイム CQI の遅延に応じて異なる表示になります。



アイコンの表示が変化するタイミングを制御するしきい値があります。デフォルトでは、次のように設定されています。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "HighLatencyThreshold" -d "0x000000dc" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "LowLatencyThreshold" -d "0x00000078" --force
```

実際の遅延が**LowLatencyThreshold**以下の場合、アイコンは緑色で表示されます。実際の遅延が**HighLatencyThreshold**より大きい場合、アイコンは赤色で表示されます。その他の状況では、アイコンは黄色で表示されます。リアルタイムCQIが無効になっている場合、アイコンには色マークがありません。

リアルタイムCQIはデフォルトで有効になっており、表示されます。これを無効にして非表示にし、トレイアイコンに色マークを表示しないようにするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "EnableCqiShow" -d "0x00000000" --force
```

- **グラフィックステータス**

  このインジケーターは、現在のセッションのグラフィック設定を表示します。デフォルトで有効になっています。これを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "EnableGfxInfo" -d "0x00000000" --force
```

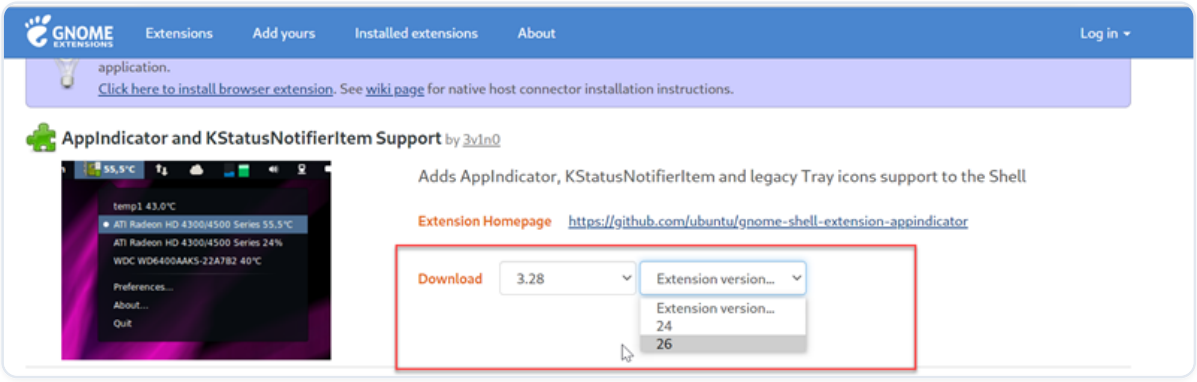
システムトレイを有効にする

システムトレイとクライアントのバッテリーステータス表示は、デフォルトで有効になっています。ただし、特定の条件下では、システムトレイとクライアントのバッテリーステータス表示を有効にするために追加の設定を行う必要があります。詳細を以下に示します。

1. Citrix Studioでグラフィックステータスインジケーターポリシーを有効にします。
2. (**この手順は、GNOMEが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にのみ必要です。**) AppIndicatorのサポートを有効にするには、GNOMEシェル用の互換性のある拡張機能をインストールします。

   a. `gnome-shell --version`コマンドを実行して、GNOMEシェルのバージョンを確認します。

   b. `<https://extensions.gnome.org/extension/615/appindicator-support>`からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能のバージョンとして24または26を選択できます。



   c. ダウンロードしたパッケージを解凍し、解凍したディレクトリの名前を`appindicatorsupport@rgcjonas.gmail.com`に変更します。パッケージ内の**metadata.json**ファイルの**「uuid」**値が`appindicatorsupport@rgcjonas.gmail.com`に設定されていることを確認します。

   d. `mv`コマンドを実行して、`appindicatorsupport@rgcjonas.gmail.com`ディレクトリを`/usr/share/gnome-shell/extensions/`の下の場所に移動します。

   e. `chmod a+r metadata.json`コマンドを実行して、**metadata.json**ファイルを他のユーザーが読み取れるようにします。

> **ヒント:**
>
> デフォルトでは、`appindicatorsupport@rgcjonas.gmail.com`ディレクトリ内の**metadata.json**ファイルはルートユーザーのみが読み取ることができます。画面共有

> をサポートするには、**metadata.json** ファイルを他のユーザーも読み取れるようにします。

f. GNOME Tweaks をインストールします。

g. デスクトップ環境で、`Alt+F2`、`r`、`Enter` の各キーを順番に押すか、`killall -SIGQUIT gnome-shell` コマンドを実行して GNOME シェルをリロードします。

h. デスクトップ環境で GNOME Tweaks を実行し、Tweaks または Extensions ツールで **KStatusNotifierItem/AppIndicator Support** を有効にします。

> **注:**
>
> GNOME がインストールされている RHEL 9.x、Rocky Linux 9.x、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 Software アプリストアから GNOME Extensions をダウンロードし、GNOME Extensions で AppIndicator サポートを有効にすることをお勧めします。

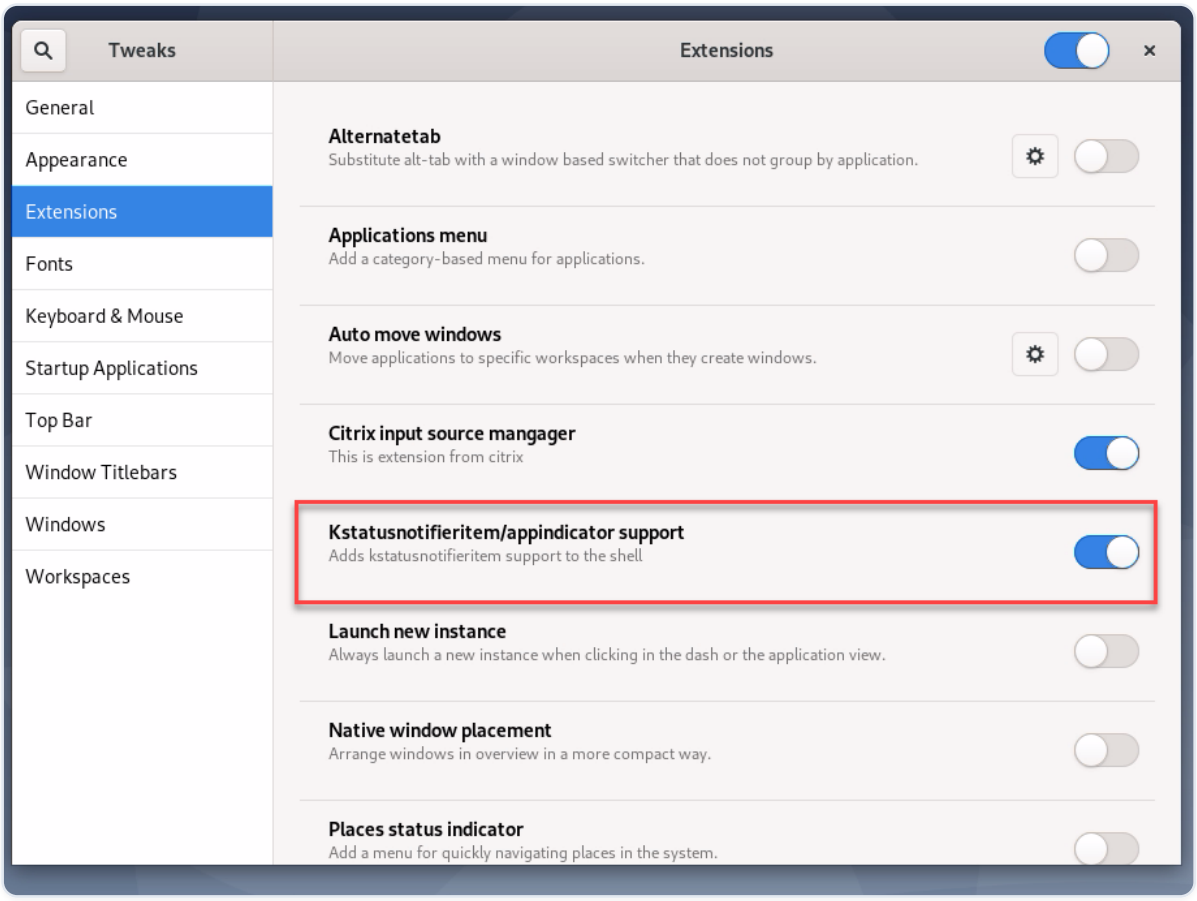
3. (**この手順は、GNOME がインストールされた Debian 11 を使用している場合にのみ必要です。**) GNOME システムトレイアイコンをインストールして有効にするには、次の手順を完了します。

a. `sudo apt install gnome-shell-extension-appindicator` コマンドを実行します。GNOME が拡張機能を認識するには、一度ログアウトしてから再度ログインする必要がある場合があります。

b. **アクティビティ**画面で Tweaks を検索します。

c. Tweaks ツールで **Extensions** を選択します。

d. **Kstatusnotifieritem/appindicator support** を有効にします。



4. (**この手順は、GNOME がインストールされた RHEL 7.9 を使用している場合に必要です。**) GNOME システムトレイアイコンをインストールして有効にするには、次の手順を完了します。

   a. **アクティビティ**画面で Tweaks を検索します。

   b. Tweaks ツールで **Extensions** を選択します。

   c. 「**トップアイコン**」を有効にします。

      トップアイコン(/en-us/linux-virtual-delivery-agent/2411/media/top-icons.png)

   d. セッションからログアウトし、再度ログインします。

Thinwire プログレッシブ表示

低帯域幅または高遅延の接続では、セッションの対話性が低下する可能性があります。たとえば、Webページのスクロールが遅くなったり、応答しなくなったり、途切れたりすることがあります。キーボードとマウスの操作がグラフィックの更新に遅れることがあります。

バージョン7.17までは、ポリシー設定を使用して、セッションの視覚品質を**低**に設定するか、より低い色深度（16ビットまたは8ビットグラフィック）を設定することで、帯域幅の消費を削減でき

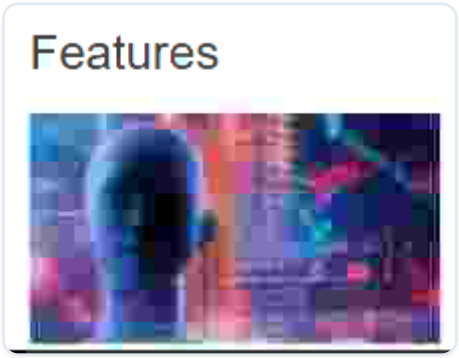
ました。ただし、ユーザーが弱い接続を使用していることを知っている必要がありました。HDX Thinwireは、ネットワークの状態に基づいて静止画の品質を動的に調整しませんでした。

バージョン7.18以降、HDX Thinwireは、以下のいずれかのケースで、デフォルトでプログレッシブ更新モードに切り替わります。

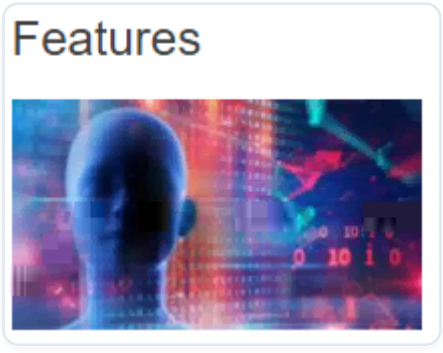
- 利用可能な帯域幅が2 Mbpsを下回る場合。
- ネットワーク遅延が200 msを超える場合。

このモードでは：

たとえば、プログレッシブ更新モードがアクティブな以下のグラフィックでは、文字の**F**と**e**に青いアーティファクトがあり、画像は高度に圧縮されています。このアプローチにより、帯域幅の消費が大幅に削減され、画像やテキストをより迅速に受信できるようになり、セッションの対話性が向上します。



セッションとの対話を停止すると、劣化した画像とテキストは徐々にシャープ化され、ロスレスになります。たとえば、以下のグラフィックでは、文字に青いアーティファクトがなくなり、画像が元の品質で表示されます。



画像の場合、シャープ化にはランダムなブロック状の方法が使用されます。テキストの場合、個々の文字または単語の一部がシャープ化されます。シャープ化プロセスは複数のフレームにわたって行われます。このアプローチにより、単一の大きなシャープ化フレームによる遅延の発生が回避されます。

一時的な画像（ビデオ）は、引き続きアダプティブディスプレイまたはSelective H.264で管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは**視覚品質**ポリシー設定の**高**、**中**（デフォルト）、**低**でスタンバイ状態になっています。

プログレッシブモードは、次の場合に強制的にオフになります（使用されません）。

- **視覚品質** = **常にロスレス** または **ロスレスにビルド**
- **単純なグラフィックの優先色深度** = 8ビット
- **圧縮にビデオコーデックを使用** = **画面全体** (全画面H.264が必要な場合)

プログレッシブモードがスタンバイ状態の場合、デフォルトでは以下のいずれかの条件が発生すると有効になります。

- 利用可能な帯域幅が2 Mbpsを下回る
- ネットワーク遅延が200 msを超える

モード切り替えが発生した後、たとえネットワークの状態が悪くても一時的なものであっても、そのモードで最低10秒間は維持されます。

プログレッシブモードの動作を変更する

次のコマンドを実行して、プログレッシブモードの動作を変更できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplay" -d "<value>" --force
```

ここで、\<value\> は次のとおりです。

0 = 常にオフ (いかなる状況でも使用しない)

1 = 自動 (ネットワークの状態に基づいて切り替え、デフォルト値)

2 = 常にオン

自動モード (1) の場合、プログレッシブモードが切り替わるしきい値を変更するには、次のいずれかのコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

ここで、\<value\> は \<Kbps単位のしきい値\> です (デフォルト = 2,048)。

例: 4096 = 帯域幅が4 Mbpsを下回る場合、プログレッシブモードをオンにする

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

\<値\> は \<しきい値 (ミリ秒)\> (デフォルト = 200) です

例: 100 = ネットワーク遅延が100ミリ秒を下回る場合、プログレッシブモードをオンにする。

一般的なコンテンツリダイレクト

クライアントドライブマッピングとクライアントフォルダーリダイレクト

条件	結果
ホスト (VDA) でクライアントドライブマッピングのみを有効にする場合、,	クライアント側のフルボリュームは、ホームディレクトリの **ctxmnt** サブディレクトリの下のセッションに自動的にマッピングされます。
ホスト (VDA) でクライアントフォルダーリダイレクトを有効にし、ユーザーがユーザーデバイス (クライアント) でそれを構成する場合、,	ユーザーが指定したローカルボリュームの一部がリダイレクトされます。

USBデバイスリダイレクト

USB デバイスは、Citrix Workspace™ app と Linux VDA デスクトップ間で共有されます。USB デバイスがデスクトップにリダイレクトされると、ローカルに接続されているかのように USB デバイスを使用できます。

クライアントドライブマッピング

クライアントドライブマッピングとクライアントフォルダーリダイレクトを使用して、クライアント側のファイルをホスト側のセッションでアクセスできるようにすることができます。クライアントドライブマッピングとクライアントフォルダーリダイレクトの比較は次のとおりです。

条件	結果
ホスト（VDA）でクライアントドライブマッピングのみを有効にする場合、,	クライアント側のフルボリュームは、ホームディレクトリの**ctxmnt**サブディレクトリの下のセッションに自動的にマッピングされます。
ホスト（VDA）でクライアントフォルダーリダイレクトを有効にし、ユーザーがユーザーデバイス（クライアント）でそれを構成する場合、,	ユーザーが指定したローカルボリュームの一部がリダイレクトされます。

クライアントドライブマッピングを有効にする

クライアントドライブマッピングを有効にするには、Citrix Studioで**Client drive redirection**ポリシーを**Allowed**に設定します。ポリシーの詳細については、「ファイルリダイレクトポリシー設定」を参照してください。

クライアントフォルダーリダイレクトを有効にし、リダイレクトするフォルダーを指定する

クライアントフォルダーリダイレクトを有効にするには、VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Client Folder
Redirection" -t "REG_DWORD" -v "CFROnlyModeAvailable" -d "0x00000001" --
force
```

クライアントからホスト側のセッションにリダイレクトするフォルダーを指定するには、ユーザーデバイスで次の手順を実行します。

1. Citrix Workspace™アプリの最新バージョンがインストールされていることを確認してください。
2. Citrix Workspaceアプリのインストールディレクトリから、**CtxCFRUI.exe**を起動します。
3. **Custom** ラジオボタンを選択し、フォルダーを追加、編集、または削除します。
4. 設定を有効にするには、セッションを切断して再接続します。

USBデバイスリダイレクト

USBデバイスはCitrix Workspace™アプリとLinux VDAデスクトップ間で共有されます。USBデバイスがデスクトップにリダイレクトされると、ローカルに接続されているかのようにUSBデバイスを使用できます。

ヒント:

ネットワーク遅延が100ミリ秒未満の場合は、USBデバイスリダイレクトを使用することをお勧めします。ネットワーク遅延が200ミリ秒を超える場合は、USBデバイスリダイレクトを使用しないでください。

USBデバイスリダイレクトには、主に3つの機能領域があります。

- オープンソースUSB/IPプロジェクト
- Citrix USBセッションモジュール
- Citrix USBサービスモジュール

オープンソースUSB/IPプロジェクト:

USB/IPプロジェクトは、Linuxカーネルドライバーと、カーネルドライバーと通信してすべてのUSBデータを取得できるユーザーモードライブラリで構成されています。

Linux VDAは、オープンソースのUSB/IPプロジェクトに基づいてUSBデバイスリダイレクトを実装し、USB/IPのカーネルドライバーとユーザーモードライブラリを再利用します。ただし、Linux VDAとCitrix Workspaceアプリ間のすべてのUSBデータ転送は、Citrix ICA USBプロトコルによってカプセル化されます。

Citrix USBセッションモジュール:

Citrix USBセッションモジュールは、USB/IPカーネルモジュールとCitrix Workspaceアプリ間の通信ブリッジとして機能します。

Citrix USBサービスモジュール:

Citrix USBサービスモジュールは、USBデバイスの接続や切断など、USBデバイスに対するすべての操作を管理します。

USBデバイスリダイレクトの仕組み

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム /dev パスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで利用できない場合があります。USBデバイスは正常に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスでき

ません。USBデバイスの接続性を確保するには、ドライバーをインストールし、システムを適切に構成してください。

Linux VDA は、クライアントから正常にリダイレクトされる USB デバイスのリストをサポートしています。

サポートされている USB デバイス

> **ヒント:**
>
> USB 3.0 ポートのサポートを追加しました。クライアントデバイスの USB 3.0 ポートに USB 3.0 デバイスを挿入できます。

以下のデバイスは、このバージョンの Linux VDA をサポートすることが確認されています。その他のデバイスも自由に使用できますが、予期しない結果が生じる可能性があります。

USB マスストレージデバイス	VID:PID	ファイルシステム
`Netac Technology Co., Ltd`	0dd8:173c	FAT32, NTFS
`Kingston Datatraveler 101 II`	0951:1625	FAT32, NTFS
`Kingston Datatraveler GT101 G2`	1567:8902	FAT32, NTFS
`SanDisk SDCZ80 flash drive`	0781:5580	FAT32, NTFS
`WD HDD`	1058:10B8	FAT32, NTFS
`Toshiba Kingston DataTraveler 3.0 USB device`	0930:6545	FAT32, NTFS
`Taiwan OEM – OBSOLETE VendorCo ProductCode Disk 2.0`	FFFF:5678	FAT32, NTFS

USB マスストレージデバイス	VID:PID	ファイルシステム
`TD-RDF5A Transcend USB device`	8564:4000	FAT32, NTFS

注:

Amazon Linux 2、RHEL、Rocky Linux、SUSEでNTFSを使用するには、まずこれらのディストリビューションでNTFSサポートを有効にしてください。

USB 3Dマウス	VID:PID
`3DConnexion SpaceMouse Pro`	046d: c62b

USBスキャナー	VID:PID
`Epson Perfection V330 photo`	04B8: 0142

ユビコ USB	VID:PID
`Yubico YubiKey OTP+FIDO+CCID -Keyboard, HID`	1050:0407

ウェブカメラ USB	VID:PID
`Logitech composite USB device – WebCam, Audio`	0460:0825

USBデバイスリダイレクトを構成する

(RHELおよびRocky Linuxのみ) USB/IPカーネルモジュールをインストールまたはコンパイルする

Linux VDA は、USB デバイスリダイレクトの仮想ホストコントローラーとして USB/IP を使用します。ほとんどの場合、USB/IP カーネルモジュールは Linux カーネルバージョン 3.17 以降でリリースされているため、デフォルトでカーネルモジュールをビルドする必要はありません。ただし、RHEL および Rocky Linux では USB/IP カーネルモジュールは利用できません。これらの

Linux ディストリビューションで USB デバイスリダイレクトを使用するには、USB/IP カーネルモジュールをインストールまたはコンパイルする必要があります。お使いの Linux ディストリビューションに基づいて、`<https://pkgs.org/download/kmod-usbip>` から USB/IP をダウンロードしてインストールしてください。

USBデバイスリダイレクトポリシーを設定する

Citrixポリシーは、USBデバイスリダイレクトが有効か無効かを制御します。デバイスの種類は、Delivery Controller™ポリシーを使用して指定することもできます。Linux VDAのUSBデバイスリダイレクトを構成する際は、以下のポリシーを構成してください。

- クライアントUSBデバイスリダイレクトポリシー
- クライアントUSBデバイスリダイレクト規則

USBデバイスリダイレクトを有効にする

Citrix Studioで、クライアントからのUSBデバイスリダイレクトを有効（または無効）にします（ワークステーションホストのみ）。

設定の編集ダイアログで:

1. **許可**を選択します。
2. **OK**をクリックします。

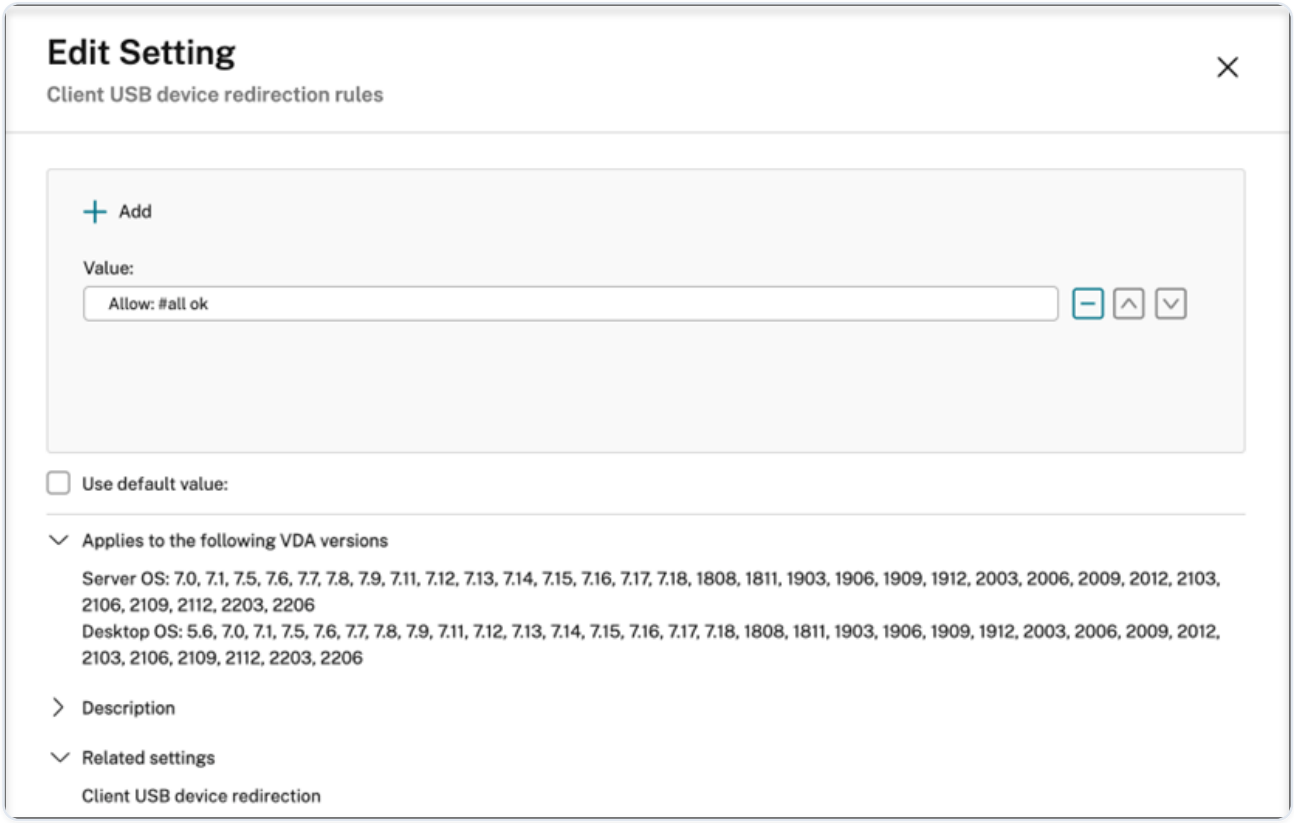


USBデバイスリダイレクト規則を設定する

USBリダイレクトポリシーを有効にした後、Citrix Studioを使用して、Linux VDAで許可（または拒否）するデバイスを指定することで、リダイレクト規則を設定します。

［クライアントUSBデバイスリダイレクト規則］ダイアログで、次の操作を行います。

1. リダイレクト規則を追加するには、**［新規］をクリックします。既存の規則を確認するには、［編集］**をクリックします。
2. 規則を作成（または編集）したら、**［OK］**をクリックします。



注:

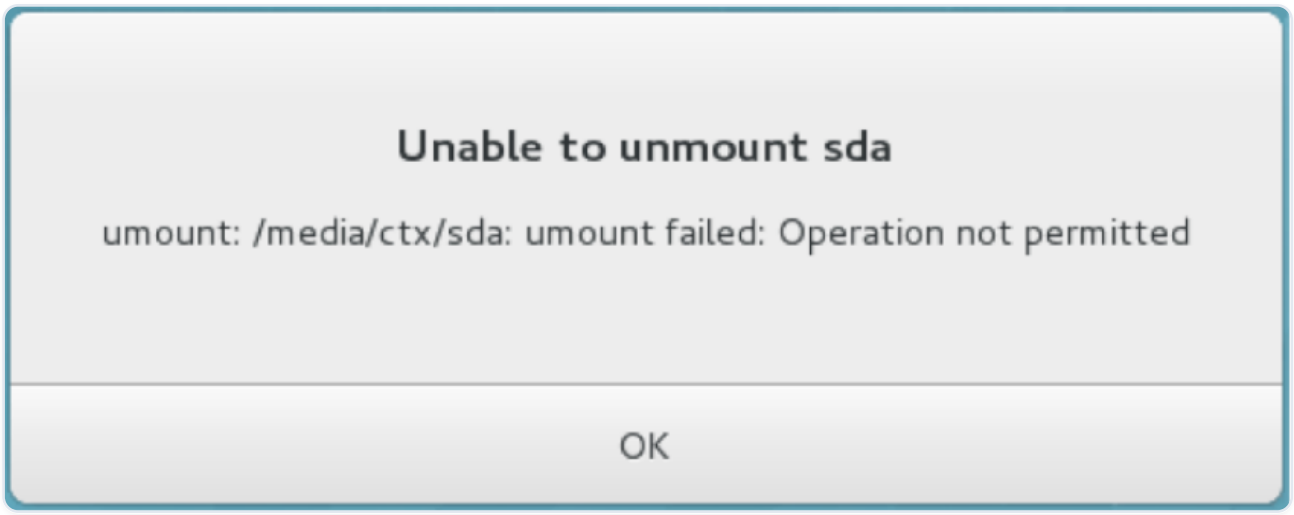
USBデバイスリダイレクトを構成する際は、**［クライアントUSBデバイスリダイレクト］オプションを［許可］に設定し、［クライアントUSBデバイスリダイレクト規則］を［Allow:#all ok］**として構成してください。両方の設定を構成しないと、セッション内でクリックしたときに外部マウスが消える可能性があります。

USBデバイスリダイレクトの問題をトラブルシューティングする

このセクションの情報を使用して、Linux VDAの使用時に発生する可能性のあるさまざまな問題をトラブルシューティングしてください。

リダイレクトされたUSBディスクをアンマウントできない

Linux VDAは、Citrix Workspaceアプリからリダイレクトされたすべてのディスクを管理者権限で管理し、所有者のみがリダイレクトされたデバイスにアクセスできるようにします。そのため、デバイスをアンマウントできるのは管理者権限を持つユーザーのみです。



USBディスクのリダイレクトを停止するとファイルが失われる

Citrix Workspaceアプリのツールバーを使用してUSBディスクのリダイレクトをすぐに停止すると、ディスク上で変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込むときに、システムがファイルシステム内のメモリキャッシュをマウントするため、データがディスク自体に書き込まれないために発生します。Citrix Workspaceアプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。

この問題を解決するには、USBリダイレクトを停止する前に、ターミナルで **sync** コマンドを使用してデータをディスクにフラッシュします。

Citrix Workspaceアプリのツールバーにデバイスがない

Citrix Workspaceアプリのツールバーにデバイスが表示されない場合があります。これは、USBリダイレクトが行われていないことを示しています。

ワークスペースアプリのツールバーにあるデバイスタブ(/en-us/linux-virtual-delivery-agent/2411/media/client-usb-redirection-troubleshoot.png)

この問題が発生した場合は、以下を確認してください。

- ポリシーがUSBデバイスリダイレクトを許可するように構成されていること。
- Citrix USBサービスモジュールが実行されていること。

ポリシーが正しく設定されていない場合は、この記事の「USBデバイスリダイレクトポリシーの設定」セクションを参照して修正してください。

Citrix USBサービスモジュールが実行されていない場合は、次の手順を実行してください。

1. 次のコマンドを使用して、お使いのLinuxディストリビューションでUSB/IPカーネルモジュールが利用可能かどうかを確認します。

   ```
   modinfo usbip-core
   ```

2. 出力が次のように表示される場合は、お使いのLinuxディストリビューションに基づいてUSB/IPカーネルモジュールをインストールまたはコンパイルします。

   ```
   modinfo: ERROR: Module usbip-core not found.
   ```

   - RHELおよびRocky Linuxについては、この記事の「USB/IPカーネルモジュールのインストールまたはコンパイル」セクションを参照してください。
   - SUSEの場合は、`<https://software.opensuse.org/package/usbip>`からUSB/IPパッケージをダウンロードしてインストールします。
   - Ubuntu/Debianの場合は、USB/IPカーネルモジュールをコンパイルしてインストールするために、次の手順を実行します。

     a. USB/IPカーネルモジュールのソースコードをダウンロードします。

        `<https://github.com/torvalds/linux/tree/master/drivers/usb/usbip>`にあるLinuxカーネルリポジトリにアクセスし、対象のLinuxカーネルバージョン (v4.15以降) タグを選択して、`<https://github.com/torvalds/linux/tree/v4.15/drivers/usb/usbip>`のようなリンクを取得します。

        DownGitにアクセスし、前のリンクを入力してUSB/IPソースコードをダウンロードするためのダウンロードリンクを作成します。

     b. 以下のコマンドを使用してソースファイルを解凍します。

        ```
        unzip ${USBIP_SRC}.zip



        cd usbip
        ```

c. **Makefile**ファイルを次のように変更します。

```
# SPDX-License-Identifier: GPL-2.0


ccflags-$(CONFIG_USBIP_DEBUG) := -DDEBUG


obj-$(CONFIG_USBIP_CORE) += usbip-core.o


usbip-core-y := usbip_common.o usbip_event.o


obj-$(CONFIG_USBIP_VHCI_HCD) += vhci-hcd.o


vhci-hcd-y := vhci_sysfs.o vhci_tx.o vhci_rx.o vhci_hcd.o


#obj-$(CONFIG_USBIP_HOST) += usbip-host.o


#usbip-host-y := stub_dev.o stub_main.o stub_rx.o stub_tx.o


#obj-$(CONFIG_USBIP_VUDC) += usbip-vudc.o


#usbip-vudc-y := vudc_dev.o vudc_sysfs.o vudc_tx.o vudc_rx.o
vudc_transfer.o vudc_main.o
```

d. ソースコードをコンパイルします。

```
apt-get install linux-headers-`uname -r`



make -C /lib/modules/`uname -r`/build M=$PWD
```

e. USB/IPカーネルモジュールをインストールします。

```
cp usbip-core.ko vhci-hcd.ko /opt/Citrix/VDA/lib64/
```

f. **ctxusbsd**サービスを再起動して、USB/IPカーネルモジュールをロードします。

```
systemctl restart ctxusbsd
```

USBデバイスがCitrix Workspaceアプリのツールバーに表示されているにもかかわらず、「ポリシーによって制限されています」と表示される場合のリダイレクトの失敗

問題が発生した場合は、次の操作を行います。

- リダイレクトを有効にするようにLinux VDAポリシーを構成します。
- Citrix Workspaceアプリのレジストリに追加のポリシー制限が構成されていないか確認します。レジストリパスの**DeviceRules**を確認し、この設定によってデバイスへのアクセスが拒否されていないことを確認します。

  HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Citrix\\ICA Client\\GenericUSB

USBデバイスは正常にリダイレクトされますが、セッションで使用できません

通常、サポートされているUSBデバイスのみがリダイレクトできます。他のデバイスもアクティブなLinux VDAセッションにリダイレクトされる場合があります。リダイレクトされたすべてのデバイスについて、ユーザーが所有するノードがシステム**/dev**パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている（接続されている）がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限ポリシーに追加します。

> **注:**
>
> USBドライブの場合、Linux VDAはディスクを構成してマウントします。ユーザー（およびそれをインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。

> これは、サポートされているデバイスリストにないデバイスには当てはまらない場合があります。

クリップボードリダイレクト

クリップボードリダイレクトを使用すると、VDAセッションで実行されているアプリケーションとクライアントデバイスで実行されているアプリケーションの間でデータをコピー＆ペーストできます。

この記事では、クリップボードリダイレクトを実現するために利用できるCitrix®ポリシーについて説明します。

クリップボードリダイレクトのCitrixポリシー

クライアントクリップボードリダイレクト

この設定は、クライアントデバイスのクリップボードがVDAのクリップボードにマップされることを許可または禁止します。

デフォルトでは、クリップボードリダイレクトは**許可**に設定されています。

セッションとローカルクリップボード間のコピー＆ペーストデータ転送を禁止するには、**禁止**を選択します。ユーザーは、セッションで実行されているアプリケーション間では引き続きデータをコピー＆ペーストできます。

クリップボードリダイレクトの帯域幅制限

この設定は、セッションとローカルクリップボード間のデータ転送に許可される最大帯域幅（kbps）を指定します。

クリップボードリダイレクトの帯域幅制限（パーセント）

この設定は、セッションとローカルクリップボード間のデータ転送に許可される最大帯域幅を、総セッション帯域幅の割合として指定します。

クリップボードのクライアントからセッションへの転送サイズを制限する

この設定は、単一のコピー＆ペースト操作でクライアントデバイスから仮想セッションに転送できるクリップボードデータの最大サイズを指定します。

クリップボード転送サイズを制限するには、**クライアントからセッションへのクリップボード転送サイズを制限**設定を有効にします。次に、**サイズ制限**フィールドにキロバイト単位の値を入力して、ローカルクリップボードとセッション間のデータ転送サイズを定義します。

デフォルトでは、この設定は無効になっています。

クリップボードのセッションからクライアントへの転送サイズを制限する

この設定は、単一のコピー＆ペースト操作で仮想セッションからクライアントデバイスに転送できるクリップボードデータの最大サイズを指定します。

クリップボード転送サイズを制限するには、**セッションからクライアントへのクリップボード転送サイズを制限**設定を有効にします。次に、**サイズ制限**フィールドにキロバイト単位の値を入力して、セッションとローカルクリップボード間のデータ転送サイズを定義します。

デフォルトでは、この設定は無効になっています。

クライアントクリップボードへの書き込みを制限 および クライアントクリップボードへの書き込み許可形式

この2つの設定を有効にすると、特定のデータ形式をセッションからクライアントにコピー＆ペースト（クライアントへの書き込み）できるようになります。

次のクリップボード形式はシステム定義です。

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

セッションクリップボード書き込みの制限 および セッションクリップボード書き込み許可形式

これら2つの設定を有効にすると、クライアントからセッションへ（セッションへの書き込み）特定のデータ形式をコピー＆ペーストできるようになります。

次のクリップボード形式はシステム定義です。

- CF_TEXT
- CF_BITMAP

- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

キーボード

このセクションには、次のトピックが含まれています。

- クライアントIME(/ja-jp/linux-virtual-delivery-agent/2411/configure/keyboard/client-ime.html)
- クライアントIMEユーザーインターフェイスの同期(/ja-jp/linux-virtual-delivery-agent/2411/configure/keyboard/client-ime-user-interface-synchronization.html)
- 動的キーボードレイアウトの同期(/ja-jp/linux-virtual-delivery-agent/2411/configure/keyboard/dynamic-keyboard-layout-synchronization.html)
- ソフトキーボード(/ja-jp/linux-virtual-delivery-agent/2411/configure/keyboard/soft-keyboard.html)
- 複数言語入力のサポート(/ja-jp/linux-virtual-delivery-agent/2411/configure/keyboard/configure-server-ime.html)

Xキーボード拡張機能 (XKB) の設定

2407リリース以降、Linux VDAはキーボード設定のデフォルトのXKBルールとしてevdevを指定しています。この決定にはいくつかの理由があります。

- ほとんどの最新のLinuxディストリビューションは、デフォルトでevdev XKBルールを使用しています。
- ほとんどすべてのアプリケーションが、evdev XKBルールのキーコードパターンに従っています。
- evdev XKBルールは、基本的な英数字や一般的なファンクションキーを超えて、「Cancel」、「Redo」、「Undo」、「XF86Copy」、「XF86Open」、「XF86Paste」など、幅広いキーシンボルに対する拡張サポートを提供します。

- Linuxのevdev XKBルールは、様々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ルールに戻したい場合は、以下のレジストリ設定を行うことができます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "XkbRuleName" -d "xorg"
```

このコマンドでは、正確な大文字と小文字を区別するテキスト「xorg」のみが「xfree86」ルールセットとして認識されます。その他の入力値は無効です。

クライアント入力メソッドエディター (IME)

概要

中国語、日本語、韓国語などの2バイト文字は、IMEを介して入力する必要があります。これらの文字は、クライアント側でCitrix Workspace™アプリと互換性のある任意のIME（Windowsネイティブの CJK IMEなど）を使用して入力します。

インストール

この機能は、Linux VDAをインストールすると自動的にインストールされます。

使用方法

通常どおりCitrix Virtual AppsまたはCitrix Virtual Desktops™セッションを開きます。

クライアントIME機能の使用を開始するには、クライアント側で必要に応じて入力方法を変更します。

既知の問題

- Googleスプレッドシートのセルに文字を入力するには、クライアントIME機能を使用する前にそのセルをダブルクリックする必要があります。
- クライアントIME機能は、パスワードフィールドでは自動的に無効になりません。
- IMEユーザーインターフェイスは、入力領域でカーソルに追従しません。

クライアントIMEユーザーインターフェイスの同期

概要

これまで、クライアントIMEユーザーインターフェイス（変換ウィンドウと候補ウィンドウを含む）は画面の左上隅に配置されていました。カーソルに追従せず、テキスト入力領域のカーソルから離れた場所に表示されることがありました。

以前のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/2411/media/composition-and-candidate-windows.png)

Citrix®は、クライアントIMEの使いやすさを向上させ、ユーザーエクスペリエンスをさらに改善します。

現在のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/2411/media/synced-composition-and-candidate-windows.png)

この機能を使用するための前提条件

1. Linux VDAでIntelligent Input Bus (`IBus`) を有効にします。Linux OSで `IBus` を有効にする方法については、OSベンダーのドキュメントを参照してください。例：
   - ウブントゥ：`<https://help.ubuntu.com/community/ibus>`
   - CentOS、RHEL: `<https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods>`
   - デビアン：`<https://wiki.debian.org/I18n/ibus>`
   - SUSE: `<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-gnome-settings.html#sec-gnome-settings-lang>`
2. この機能は自動的にインストールされますが、使用する前に有効にする必要があります。

機能の有効化と無効化

クライアントIMEユーザーインターフェイス同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、`ctxreg` ユーティリティを介してレジストリを編集します。

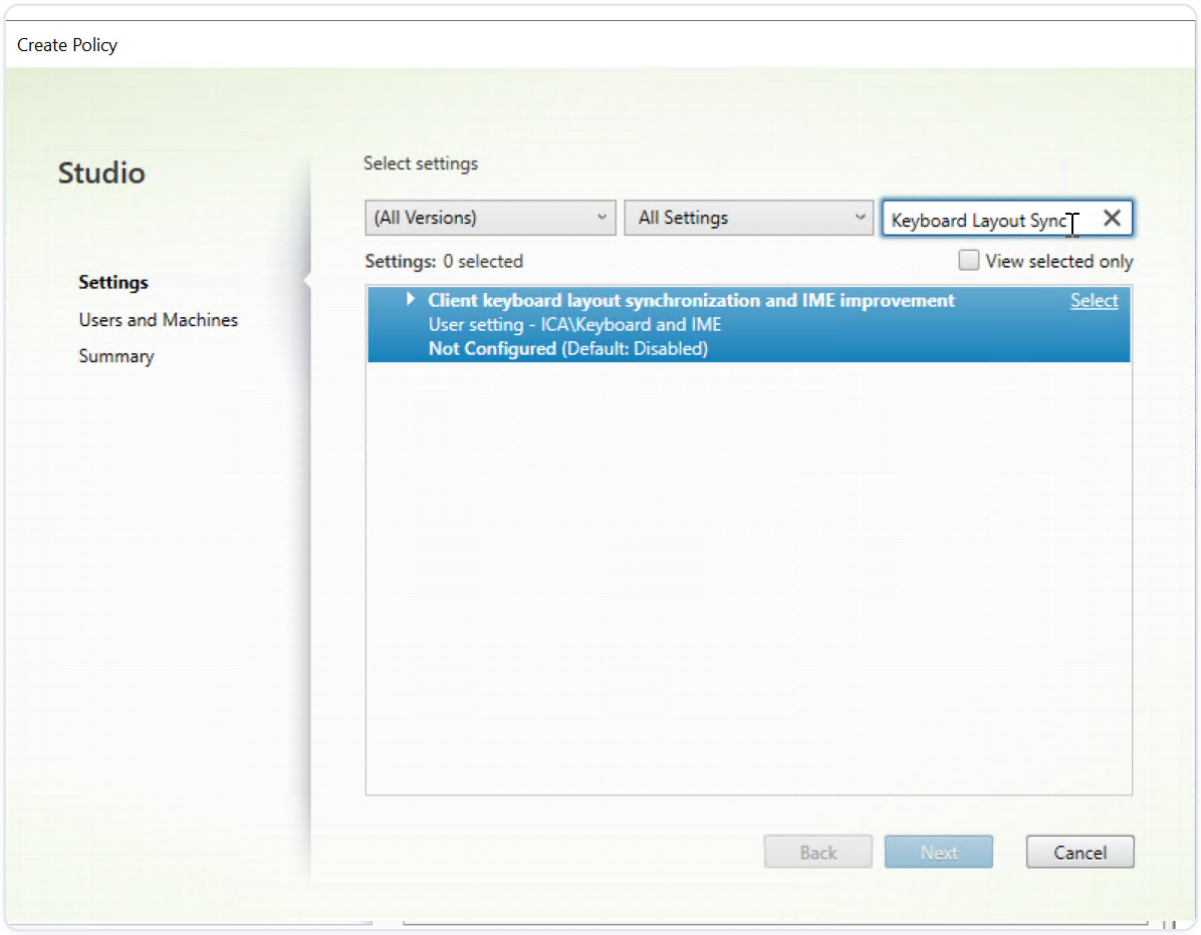
> **注:**

Client Keyboard Layout Sync and IME Improvementポリシーは、レジストリ設定よりも優先され、指定したユーザーおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- 「**クライアントキーボードレイアウト同期およびIME改善**」ポリシーを設定して、クライアントIMEユーザーインターフェース同期機能を有効または無効にします。

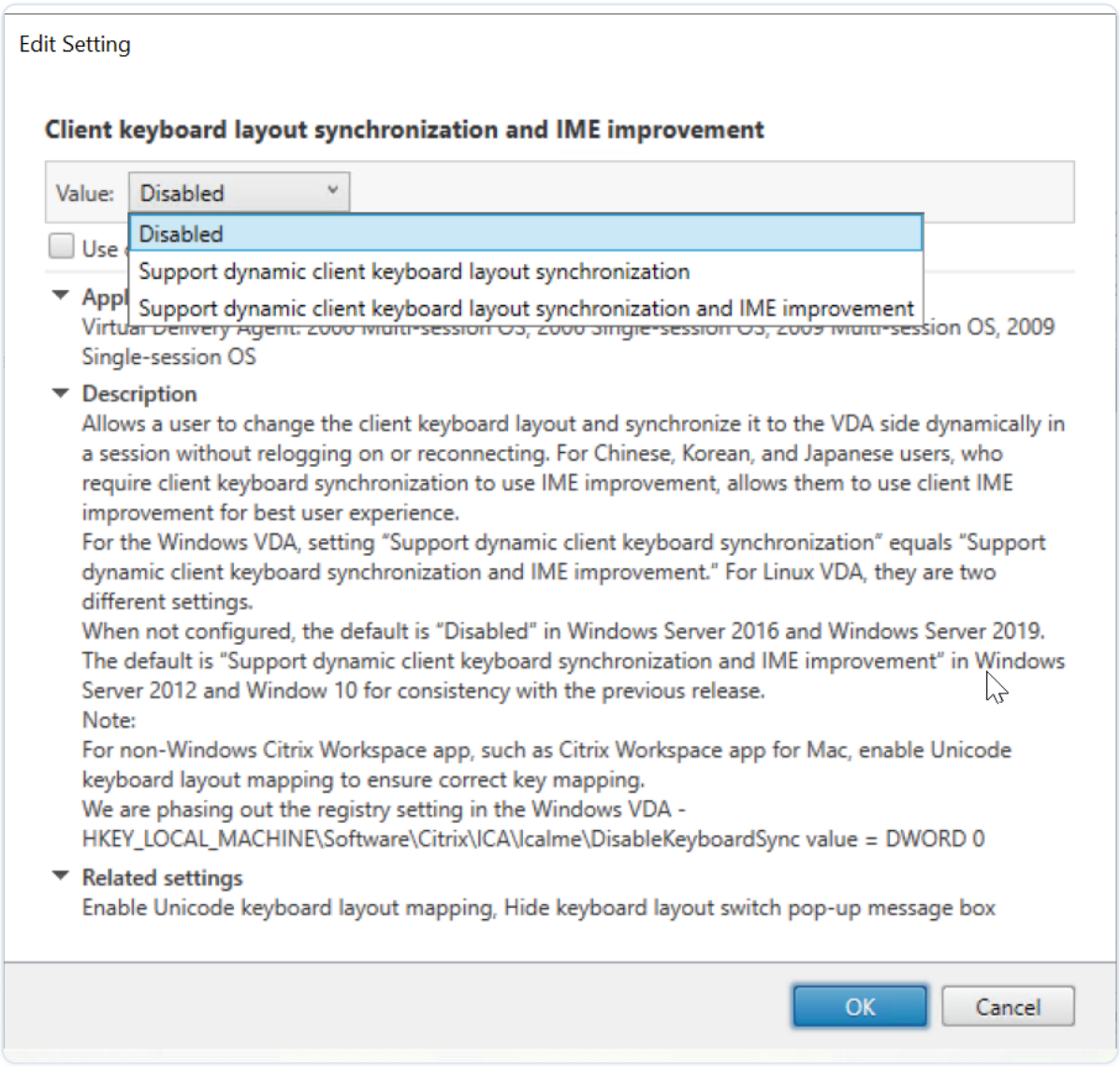
  a. Studioで、**［ポリシー］**を右クリックし、［ポリシーの作成］**を選択します。

  b. 「**クライアントキーボードレイアウト同期およびIME改善**」ポリシーを検索します。



  c. ポリシー名の横にある**［選択］**をクリックします。

  d. ポリシーを設定します。



次の3つのオプションがあります。

- **無効**：動的キーボードレイアウト同期とクライアントIMEユーザーインターフェース同期を無効にします。
- **動的クライアントキーボードレイアウト同期をサポート**：`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期を有効にします。
- **動的クライアントキーボードレイアウト同期とIME改善をサポート**：`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期とクライアントIMEユーザーインターフェース同期の両方を有効にします。

- `ctxreg`ユーティリティを介してレジストリを編集し、クライアントIMEユーザーインターフェース同期機能を有効または無効にします。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v  "SyncClientIME" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncClientIME" -d "0x00000000"
  ```

動的キーボードレイアウト同期

以前は、Linux VDA とクライアントデバイスのキーボードレイアウトは同じである必要がありました。たとえば、クライアントデバイスでキーボードレイアウトが英語からフランス語に変更されても、VDA では変更されない場合、キーマッピングの問題が発生する可能性がありました。

Citrix® は、VDA のキーボードレイアウトをクライアントデバイスのキーボードレイアウトと自動的に同期することで、この問題に対処します。クライアントデバイスのキーボードレイアウトが変更されるたびに、VDA のレイアウトもそれに追従します。

> **注:**
>
> HTML5 用 Citrix Workspace™ アプリは、動的キーボードレイアウト同期機能をサポートしていません。

構成

動的キーボードレイアウト同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**クライアントキーボードレイアウト同期とIMEの改善**ポリシーを設定するか、`ctxreg` ユーティリティを介してレジストリを編集します。

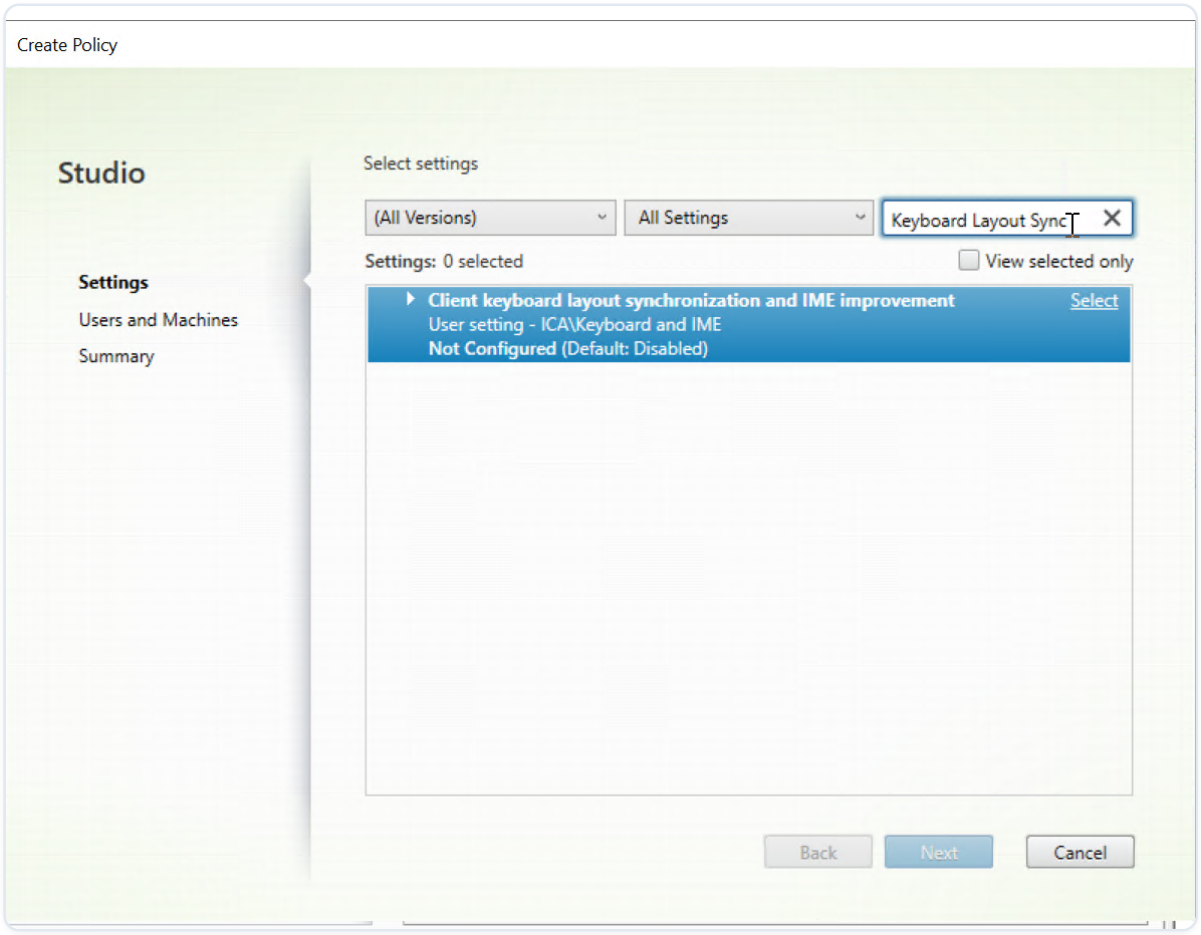
> **注:**

クライアントキーボードレイアウト同期とIMEの改善ポリシーは、レジストリ設定よりも優先され、指定したユーザーおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDAのレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- 動的キーボードレイアウト同期機能を有効または無効にするには、**クライアントキーボードレイアウト同期とIMEの改善**ポリシーを設定します。

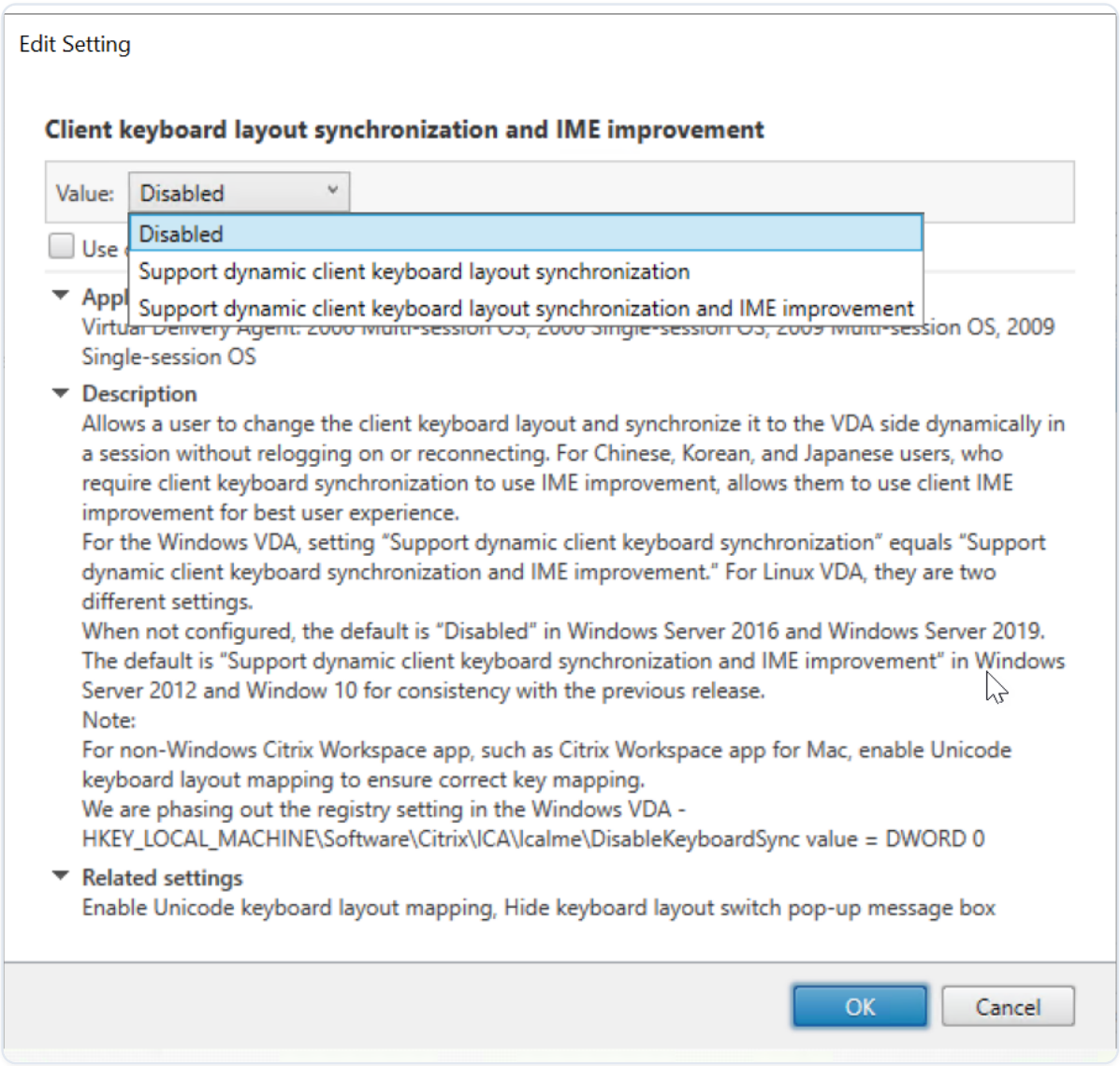
  a. Studio で、**Policies** を右クリックし、**Create Policy** を選択します。

  b. **クライアントキーボードレイアウト同期とIMEの改善**ポリシーを検索します。



  c. ポリシー名の横にある **Select** をクリックします。

  d. ポリシーを設定します。



3つのオプションがあります。

- **Disabled**: 動的キーボードレイアウト同期とクライアント IME ユーザーインターフェイス同期を無効にします。
- **Support dynamic client keyboard layout synchronization**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` の **SyncKeyboardLayout** レジストリキーの DWORD 値に関係なく、動的クライアントキーボードレイアウト同期を有効にします。
- **動的クライアントキーボードレイアウト同期とIMEの改善のサポート**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` にある**SyncKeyboardLayout** および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期の両方を有効にします。

- `ctxreg`ユーティリティを介してレジストリを編集し、動的キーボードレイアウト同期機能を有効または無効にします。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000000"
  ```

使用方法

この機能が有効になっている場合、セッション中にクライアントデバイスでキーボードレイアウトが変更されると、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語（FR）に変更すると、次のようになります。

クライアントデバイスでキーボードをフランス語に変更している画像(/en-us/linux-virtual-delivery-agent/2411/media/keyboard-changed-to-french.png)

すると、Linux VDAセッションのキーボードレイアウトも「fr.」に変更されます。

アプリケーションセッションでは、言語バーを有効にしている場合、この自動変更を確認できます。

セッションでキーボードがフランス語に変更されている画像(/en-us/linux-virtual-delivery-agent/2411/media/changed-to-fr.png)

デスクトップセッションでは、タスクバーでこの自動変更を確認できます。

タスクバーのフランス語の画像(/en-us/linux-virtual-delivery-agent/2411/media/fr-in-the-task-bar.png)

ソフトキーボード

ソフトキーボード機能は、Linux仮想デスクトップまたはアプリケーションセッションで利用できます。ソフトキーボードは、入力フィールドに入ると自動的に表示され、入力フィールドを離れると非表示になります。

ソフトキーボード(/en-us/linux-virtual-delivery-agent/2411/media/soft-keyboard.png)

> **注:**
>
> この機能は、iOSおよびAndroid版のCitrix Workspace™アプリでサポートされています。

機能の有効化と無効化

この機能はデフォルトで無効になっています。**ctxreg**ユーティリティを使用して、この機能を有効または無効にします。特定のLinux VDAでの機能構成は、そのVDA上のすべてのセッションに適用されます。

この機能を有効にするには：

1. コマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
   "Enabled" -d "0x00000001"
   ```

2. Citrix Studioで、**Automatic keyboard display**ポリシーを**Allowed**に設定します。
3. (オプション) RHEL 7およびCentOS 7の場合、次のコマンドを実行して、Intelligent Input Bus (`IBus`) をデフォルトのIMサービスとして構成します。

   ```
   echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
   ```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

> **注:**
>
> 前述の設定は、新しいセッションにログオンするか、現在のセッションからログオフして再度ログオンしたときに有効になります。

制限事項

- この機能は、Google Chrome、LibreOffice、およびその他のアプリで期待どおりに動作しない場合があります。
- ソフトキーボードを手動で非表示にした後、再度表示するには、入力フィールド以外の場所をクリックしてから、現在の入力フィールドを再度クリックします。
- Webブラウザで1つの入力フィールドから別の入力フィールドにクリックしても、ソフトキーボードが表示されない場合があります。この問題を回避するには、入力フィールド以外の場所をクリックしてから、目的の入力フィールドをクリックします。
- この機能は、Unicode文字および2バイト文字（中国語、日本語、韓国語の文字など）をサポートしていません。
- ソフトキーボードは、パスワード入力フィールドでは利用できません。
- ソフトキーボードが現在の入力フィールドと重なることがあります。この場合、アプリウィンドウを移動するか、画面をスクロールして入力フィールドをアクセス可能な位置に移動してください。
- Citrix Workspace app と Huawei タブレット間の互換性の問題により、物理キーボードが接続されていても、Huawei タブレットにソフトキーボードが表示されます。

複数の言語入力のサポート

Linux VDAバージョン1.4以降、Citrixは公開アプリケーションのサポートを追加しました。ユーザーはLinuxデスクトップ環境なしで、目的のLinuxアプリケーションにアクセスできます。

しかし、Linux VDAのネイティブ言語バーは、Linuxデスクトップ環境と高度に統合されているため、公開アプリケーションでは利用できませんでした。その結果、ユーザーは中国語、日本語、韓国語などのIMEを必要とする言語でテキストを入力できませんでした。また、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け入れる公開アプリケーション用の言語バーを提供します。言語バーを使用すると、ユーザーはサーバー側のIMEを選択し、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

この機能を有効または無効にするには、**ctxreg**ユーティリティを使用できます（デフォルトでは無効）。特定のLinux VDAサーバーでの機能構成は、そのVDAで公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \\SYSTEM\\CurrentControlSet\\Control\\Citrix\\LanguageBar」で、タイプはDWORDです。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000000"
```

使用方法

使用方法は簡単です。

1. 機能を有効にします。
2. テキスト入力を受け入れることができる公開アプリケーションにアクセスします。セッション中に、アプリケーションの横に言語バーが表示されます。
3. ドロップダウンメニューから、**地域と言語**を選択して、目的の言語（入力ソース）を追加します。

   地域と言語の画像(/en-us/linux-virtual-delivery-agent/2411/media/select-region-language-dialog-box.png)
4. ドロップダウンメニューからIMEまたはキーボードレイアウトを選択します。
5. 選択したIMEまたはキーボードレイアウトを使用して言語を入力します。

> **注:**

- VDA側の言語バーでキーボードレイアウトを変更する場合は、クライアント側（Citrix Workspace™アプリを実行している）で同じキーボードレイアウトが使用されていることを確認してください。
- **地域と言語**ダイアログボックスで設定を行う前に、**accountsservice**パッケージをバージョン0.6.37以降にアップグレードする必要があります。

IMEまたはキーボードレイアウトを選択(/en-us/linux-virtual-delivery-agent/2411/media/region-language-dialog-box.png)

マルチメディア

このセクションには、次のトピックが含まれています。

- オーディオ機能
- ブラウザコンテンツリダイレクト
- HDX Webカメラビデオ圧縮

オーディオ機能

アダプティブオーディオ

アダプティブオーディオはデフォルトで有効になっています。次のCitrix Workspace™アプリクライアントをサポートしています。

- Windows向けCitrix Workspaceアプリ – 2109以降のバージョン
- Linux向けCitrix Workspaceアプリ – 2109以降のバージョン
- Mac向けCitrix Workspaceアプリ – 2109以降のバージョン

リストに含まれていないクライアントを使用する場合、アダプティブオーディオはレガシーオーディオにフォールバックします。

アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、オーディオエクスペリエンスを向上させます。これにより、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レガシーオーディオの比較を示しています。

アダプティブオーディオ	レガシーオーディオ
最大オーディオサンプルレート: 48 kHz	最大オーディオサンプルレート: 8 kHz
ステレオチャネル	モノラルチャネル

オーディオサービスの考慮事項

Linux VDAは、Debian 12.5/12.7、RHEL 9.x、Rocky Linux 9.x、およびUbuntu 24.04でPipeWireをサポートしています。これらのOSではPipeWireがデフォルトのオーディオサービスです。その他のディストリビューションでは、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では、PulseAudio 13.99以降を使用してください。

PipeWire

Linux VDAは、Debian 12.5/12.7、RHEL 9.x、Rocky Linux 9.x、およびUbuntu 24.04でのみPipeWireをサポートします。以下にいくつかの考慮事項を示します。

- Debian 12.5/12.7、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、PipeWireを使用しています。
- Ubuntu 24.04にLinux VDAバージョン2411以降がインストールされている場合、PipeWireを使用しています。
- RHEL 9.xまたはRocky Linux 9.xにLinux VDAバージョン2407より前のバージョンがインストールされている場合、PulseAudioを使用しています。PulseAudioの使用を継続するには、何もしないでください。PipeWireに切り替えるには、以下の手順を実行してください。

  a. PulseAudioパッケージをpipewire-pulseaudioに置き換えます。

  ```
  dnf swap --allowerasing pulseaudio pipewire-pulseaudio
  ```

  b. PipeWire関連サービスをグローバルに有効にします。

  ```
  systemctl --global --now enable pipewire pipewire-pulse pipewire-
  pulse.socket wireplumber
  ```

  c. Linux VDAを再起動します。

オーディオのロス許容モード

ロス許容モードはオーディオをサポートします。この機能は、リアルタイムストリーミングのユーザーエクスペリエンスを向上させ、ユーザーが高遅延およびパケット損失のあるネットワークを介して接続している場合に、EDTを介したオーディオ品質を改善します。

損失許容モードとEDTの詳細については、Citrix Virtual Apps and Desktopsドキュメントの「追加情報」を参照してください。

オーディオ機能のロス許容モードを有効にする

オーディオのロス許容モードはデフォルトで有効になっています。無効になっている場合は、以下の手順を実行して再度有効にしてください。

1. HDX adaptive transportポリシーを設定して、アダプティブトランスポートを有効にします。アダプティブトランスポートはデフォルトで有効になっています。
2. Adaptive audio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. Loss tolerant mode for audioポリシーを設定して、ロス許容モード（EDT信頼性の低いトランスポート）を有効にします。オーディオのロス許容モードはデフォルトで無効になっています。
4. (オプション) リモート接続の場合、Citrix Gateway Serviceが必要です。

クライアントの要件と設定

オーディオのロス許容モードを使用するには、Citrix Workspaceアプリがこの機能をサポートし、有効にするように構成されていることを確認してください。そうでない場合、オーディオはデフォルトでEDT Reliable transportになります。

損失許容モードをサポートするCitrix Workspaceアプリの最小バージョンは次のとおりです。

- Windows版Citrix Workspaceアプリ、最小バージョン2309
- Linux版Citrix Workspaceアプリ、最小バージョン2311
- Mac版Citrix Workspaceアプリ、最小バージョン2311

アダプティブオーディオ用オーディオ品質エンハンサー（プレビュー）

バージョン2411以降、アダプティブオーディオにオーディオ品質エンハンサー（V2）が追加されました。

オーディオ品質エンハンサーは、以前のサンプルからオーディオをインテリジェントに再構築することで、短期間のパケット損失や中断を効果的に管理し、品質の著しい劣化を防ぎます。さらに、

必要な場合にのみ、失われたオーディオパケットをインテリジェントかつ適応的に回復します。パケット損失が持続的に増減すると、適応的に有効化および無効化されます。オーディオ品質エンハンサーは、良好なネットワーク状態と不良なネットワーク状態の両方で、オーディオ再生と録音の品質を最適化します。

オーディオ品質エンハンサーを有効にするには、VDAで次の手順を実行します。

1. アダプティブオーディオを有効にします。
2. オーディオの損失許容モードを有効にします。
3. 機能を有効にします。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v
"fEnablePacketLossConcealment" -t REG_DWORD -d 1 --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v
"fEnableNetStat" -t REG_DWORD -d 1 --force
```

4. 機能を完全に有効にするには、クライアント側のオーディオ品質エンハンサーも有効にする必要があります。つまり、オーディオ再生と録音でエンドツーエンドで機能させるには、VDA側とCitrix Workspaceアプリ側の両方でオーディオ品質エンハンサーを有効にする必要があります。

オーディオ品質エンハンサーを無効にするには、VDAで次の手順を実行します。

1. 機能を無効にします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v
"fEnablePacketLossConcealment" -d 0
```

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v
"fEnableNetStat" -d 0
```

2. 機能を完全に無効にするには、クライアント側のオーディオ品質エンハンサーも無効にする必要があります。

複数のオーディオデバイスのサポート

概要

バージョン2311以降、Linux VDAにオーディオリダイレクト機能が導入されました。この機能により、Citrix Workspaceアプリがインストールされているクライアントマシン上の複数のオーディオデバイスを、リモートのLinux VDAセッションにリダイレクトできます。

この機能が有効になっている場合：

- クライアントマシン上のすべてのローカルオーディオデバイスがセッションに表示されます。CitrixAudioSink（オーディオ出力）またはCitrixAudioSource（オーディオ入力）の代わりに、オーディオデバイスはそれぞれのデバイス名で表示されます。
- セッション内のオーディオデバイスは、接続または取り外しを行うと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、サポートされているCitrix Workspaceアプリを選択します。

LINUX VDAで機能を有効にする

複数のオーディオデバイスのサポート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か、再度有効にするには、それぞれ次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\\System\
\CurrentControlSet\\Control\\Citrix\\VirtualChannels\\Audio" -v
"fEnableAudioRedirectionV4" -d "0"
```

```
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、以下のクライアントでのみサポートされています。

- Windows版 Citrix Workspaceアプリ
- Citrix Workspaceアプリ for Linux 最小バージョン 2212
- Citrix Workspaceアプリ for HTML5 最小バージョン 2306
- Citrix Workspaceアプリ for Chrome 最小バージョン 2306
- Citrix Workspaceアプリ for Mac 最小バージョン 2311

- Citrix Workspaceアプリ for Android 最小バージョン 2405

この機能が期待どおりに動作するようにするには、Citrix Workspaceアプリで適切な設定が必要です。詳細については、Citrix Workspaceアプリのドキュメントを参照してください。

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間の切り替えが失敗する可能性があります。この問題に対処するには、VDAから現在のセッションユーザーのPulseAudio設定を削除し、セッションを再開します。PulseAudio設定を削除するには、`$ rm -r ~/.config/pulse`コマンドを実行します。

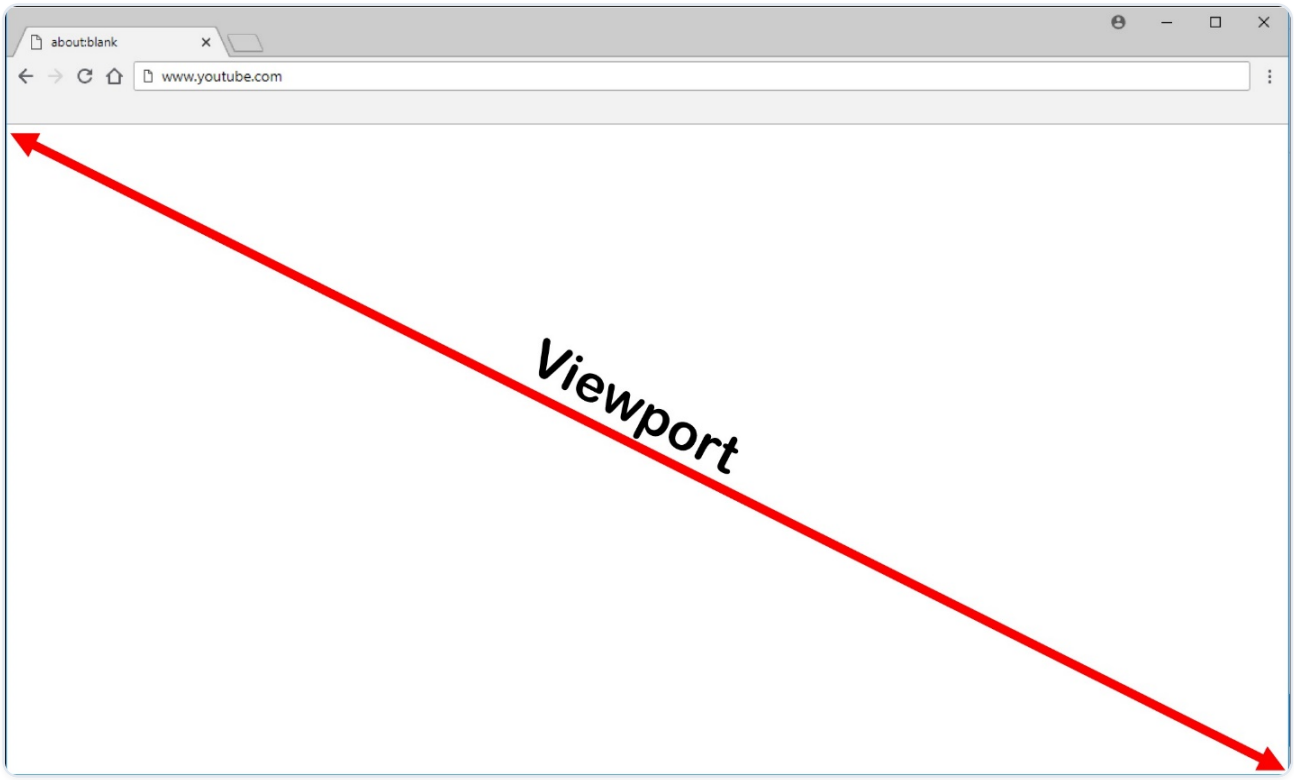
ブラウザコンテンツリダイレクト

概要

ブラウザコンテンツリダイレクトは、クライアント側で許可リスト内のWebページをレンダリングする機能を提供します。この機能は、Citrix Workspace™アプリを使用して、クライアント側で対応するレンダリングエンジンをインスタンス化し、URLからHTTPおよびHTTPSコンテンツを取得します。

> **注:**
>
> Linux VDAは、Google Chromeでのブラウザコンテンツリダイレクトをサポートしています。
>
> このオーバーレイWebレイアウトエンジンは、VDA上ではなくクライアント上で実行され、クライアントのCPU、GPU、RAM、およびネットワークを使用します。

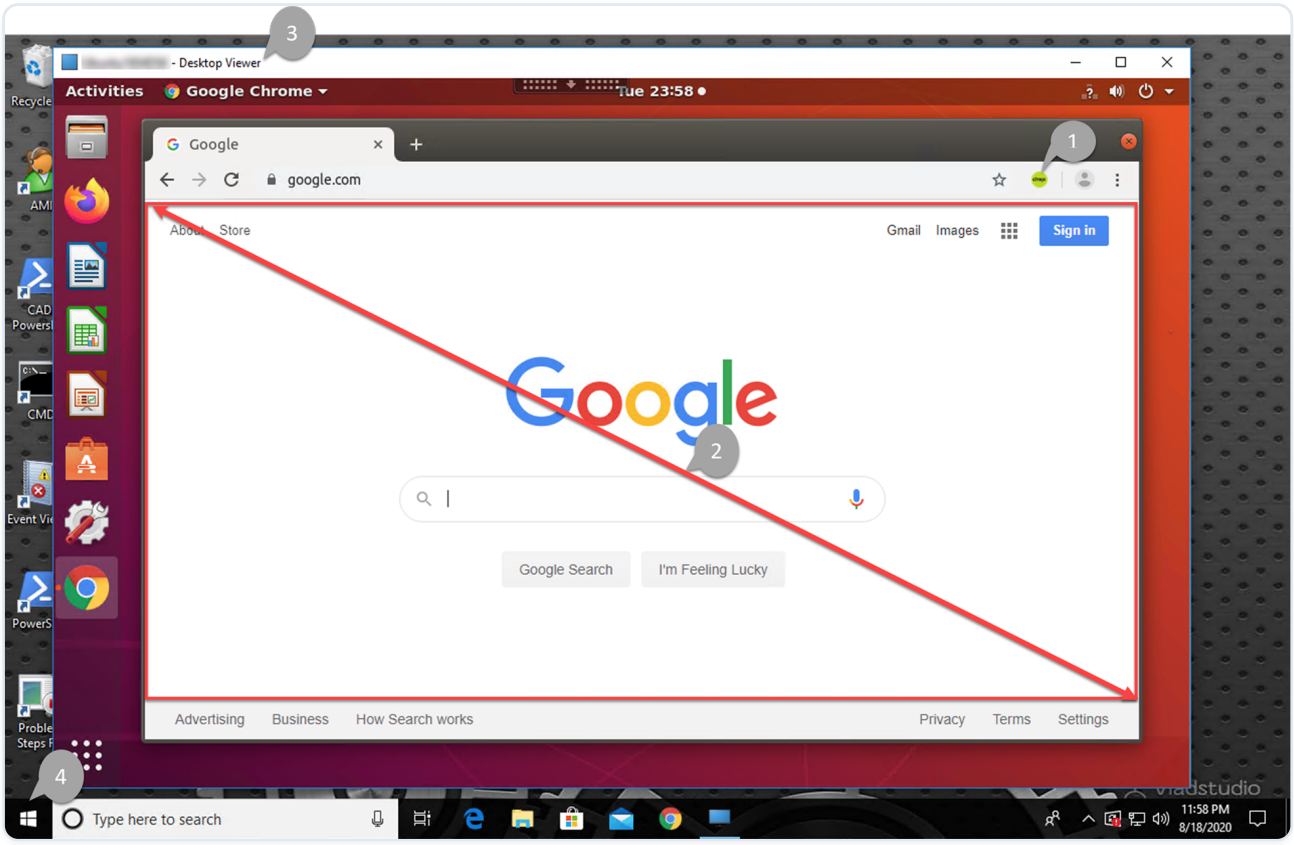
ブラウザのビューポートのみがリダイレクトされます。ビューポートとは、ブラウザ内でコンテンツが表示される長方形の領域です。ビューポートには、アドレスバー、お気に入りバー、ステータスバーなどの項目は含まれません。これらの項目は、VDA上のブラウザで引き続き実行されます。



リダイレクト用の許可リスト内のURLを含むアクセス制御リストを指定するStudioポリシーを構成します。特定のURLのリダイレクトを無効にするブロックリストを構成します。

URLが許可リスト内で見つかり、ブロックリスト内では見つからない場合、仮想チャネル (CTXCSB) は、Citrix Workspaceアプリにリダイレクトが必要であることを指示し、URLを中継します。その後、Citrix Workspaceアプリはローカルレンダリングエンジンをインスタンス化し、Webサイトを表示します。

その後、Citrix Workspaceアプリはウェブサイトを仮想デスクトップのブラウザコンテンツ領域にシームレスにブレンドし直します。



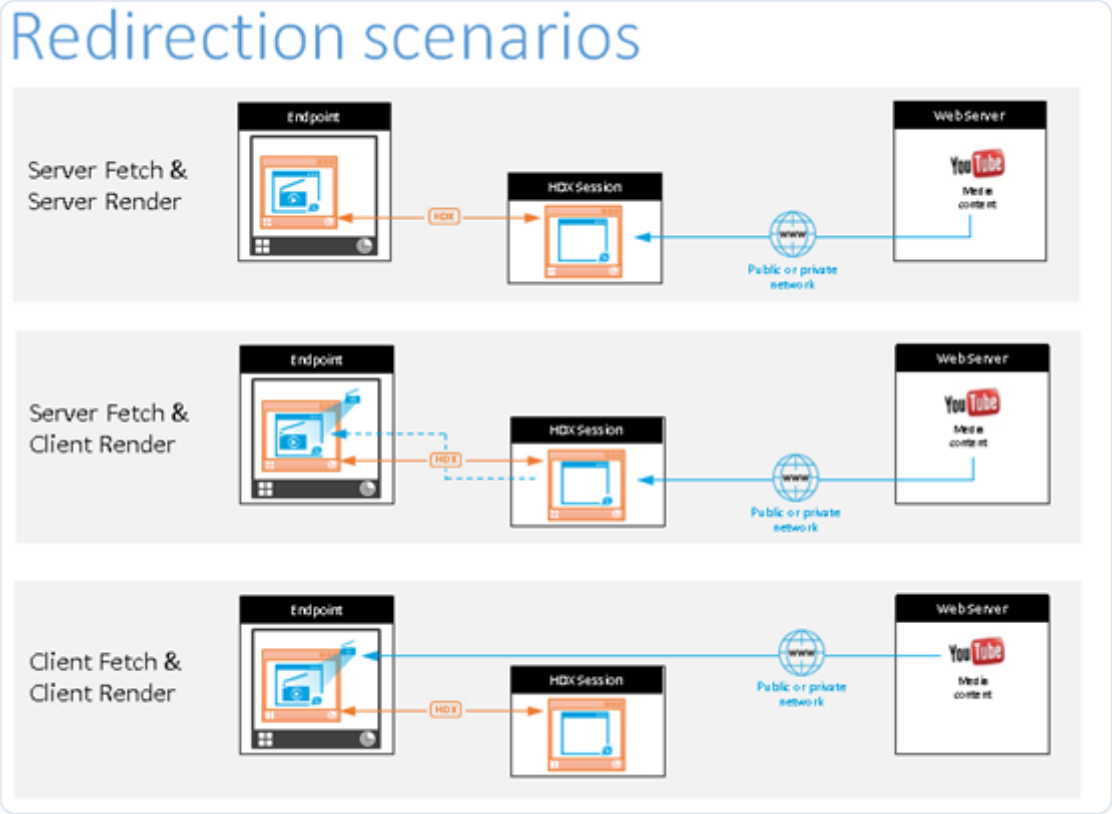
1. Citrix®ブラウザコンテンツリダイレクト拡張機能のアイコン

   拡張機能アイコンの色は、Chrome拡張機能のステータスを示します。以下の3色のいずれかです。

   - 緑：アクティブで接続済み
   - 灰色：現在のタブで非アクティブ/アイドル状態
   - 赤：破損/動作していません

2. クライアントでレンダリングされたビューポート、または仮想デスクトップにブレンドバックされたビューポート
3. リナックス ブイディーエー
4. Windowsクライアント

Citrix Workspaceアプリがコンテンツをフェッチするシナリオを次に示します。



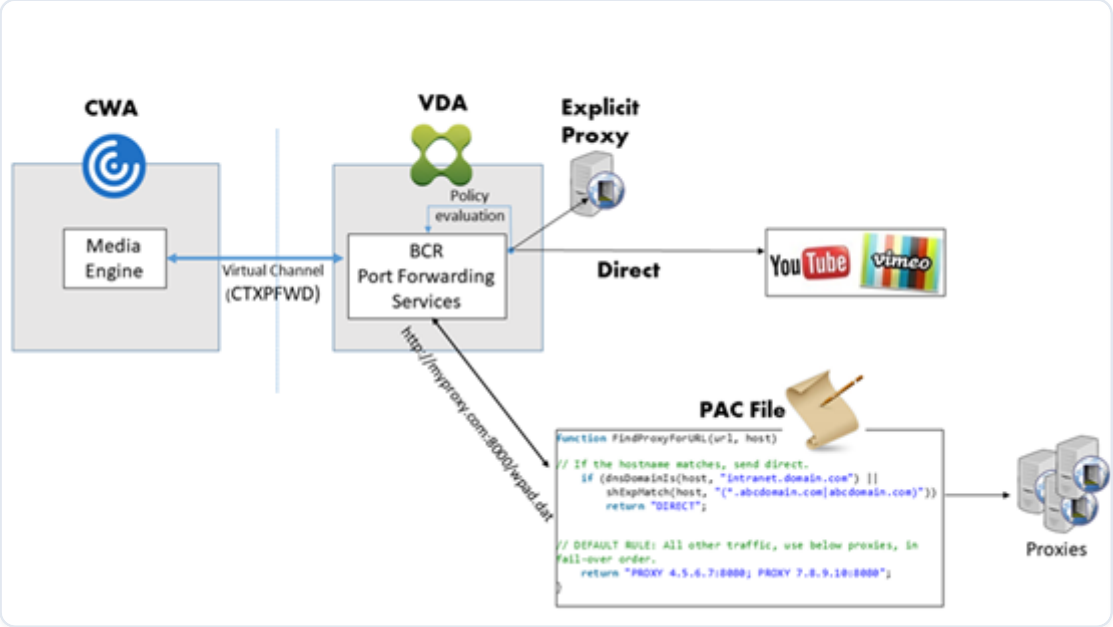
- **サーバーフェッチとサーバーレンダリング:** サイトを許可リストに追加しなかったか、リダイレクトが失敗したため、リダイレクトは行われません。VDAでWebページをレンダリングし、Thinwireを使用してグラフィックをリモート処理するようにフォールバックします。ポリシーを使用してフォールバック動作を制御します。このシナリオでは、VDAでCPU、RAM、および帯域幅の消費量が高くなります。
- **サーバーフェッチとクライアントレンダリング**: Citrix Workspace appは、仮想チャネル (CTXPFWD) を使用してVDAを介してWebサーバーに接続し、コンテンツをフェッチします。このオプションは、クライアントがWebサーバーにアクセスできない場合 (シンクライアントなど) に役立ちます。これにより、VDAでのCPUとRAMの消費は抑えられますが、ICA®仮想チャネルで帯域幅が消費されます。

  このシナリオには3つの動作モードがあります。CTXPFWDは、VDAがWebサーバーにアクセスするために使用するプロキシデバイスにデータを転送します。

  選択するポリシーオプション：

  - 明示的プロキシ - データセンターに単一の明示的プロキシがある場合。
  - 直接または透過的 - プロキシがない場合、または透過的プロキシを使用している場合。
  - PACファイル - 指定されたURLをフェッチするために、VDA内のブラウザが適切なプロキシサーバーを自動的に選択できるようにPACファイルに依存している場合。

詳しくは、この記事の後半にある**Browser Content Redirection Proxy Configuration**設定を参照してください。



- **クライアントフェッチとクライアントレンダリング:** Citrix Workspace appはWebサーバーに直接接続するため、インターネットアクセスが必要です。このシナリオでは、Citrix Virtual Apps and Desktops™サイトからすべてのネットワーク、CPU、およびRAMの使用量をオフロードします。

  利点:

  - より良いエンドユーザーエクスペリエンス（アダプティブビットレート（ABR））
  - VDAリソース使用量の削減（CPU/RAM/IO）
  - 帯域幅消費量の削減

システム要件

Windowsクライアント:

- Citrix Workspaceアプリ 1809 for Windows以降

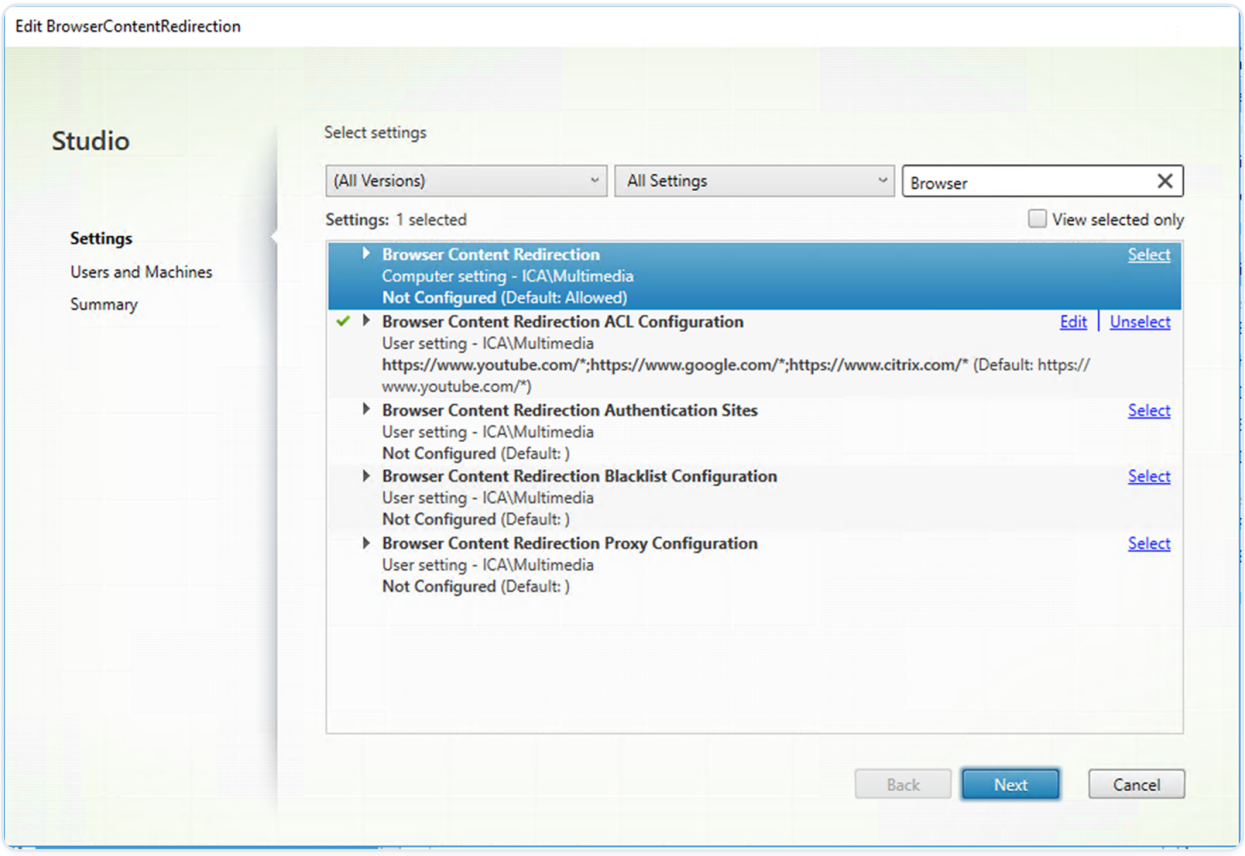
リナックス VDA:

- VDA上のブラウザー: Google Chrome v66 以降でCitrix browser content redirection拡張機能が追加されているもの

ブラウザーコンテンツリダイレクトの構成

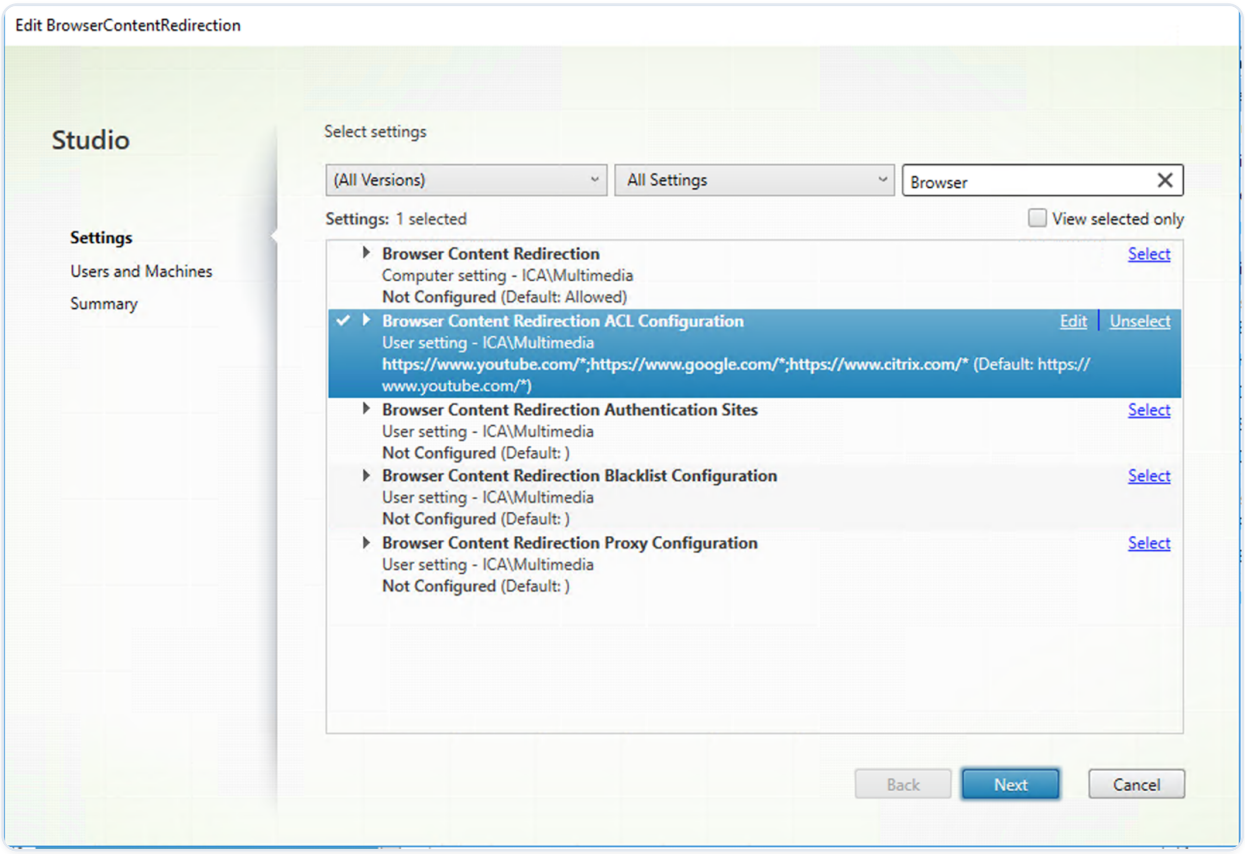
ブラウザーコンテンツリダイレクトを使用するには、関連するポリシーを構成し、Google Chromeにブラウザーコンテンツリダイレクト拡張機能をインストールします。これを行うには、次の手順を完了します。

1. Citrix Studioで、ブラウザーコンテンツリダイレクトを有効にするには、**Browser Content Redirection**を**Allowed**に設定します。



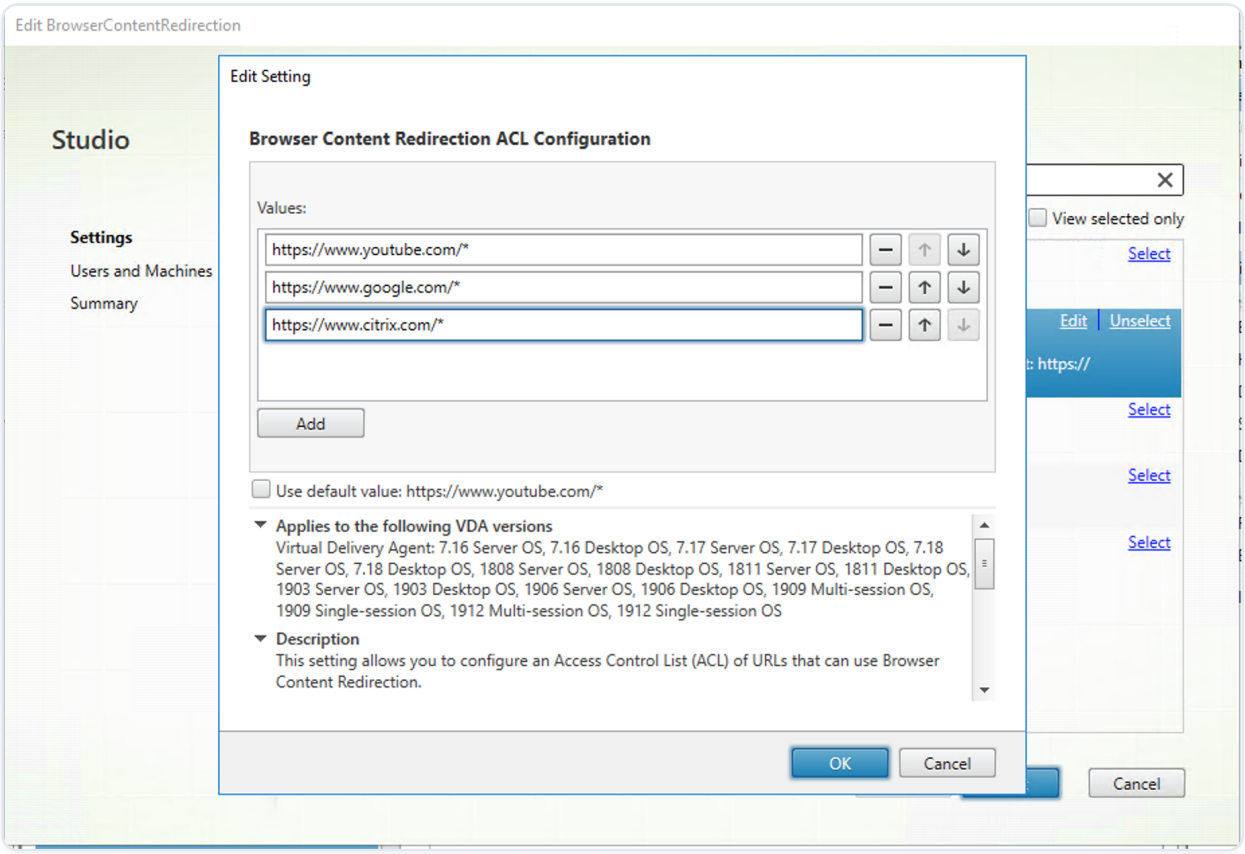
2. コンテンツをクライアントにリダイレクトできるURLの許可リストと、特定のURLのリダイレクトを無効にするブロックリストを指定します。ブロックリストの構成はオプションです。

   「**ブラウザコンテンツリダイレクトACL構成**」設定は、コンテンツをクライアントにリダイレクトできるURLの許可リストを指定します。URLを指定する際、プロトコルを除くすべてのURLコンポーネントを表すために`*`ワイルドカードを使用できます。

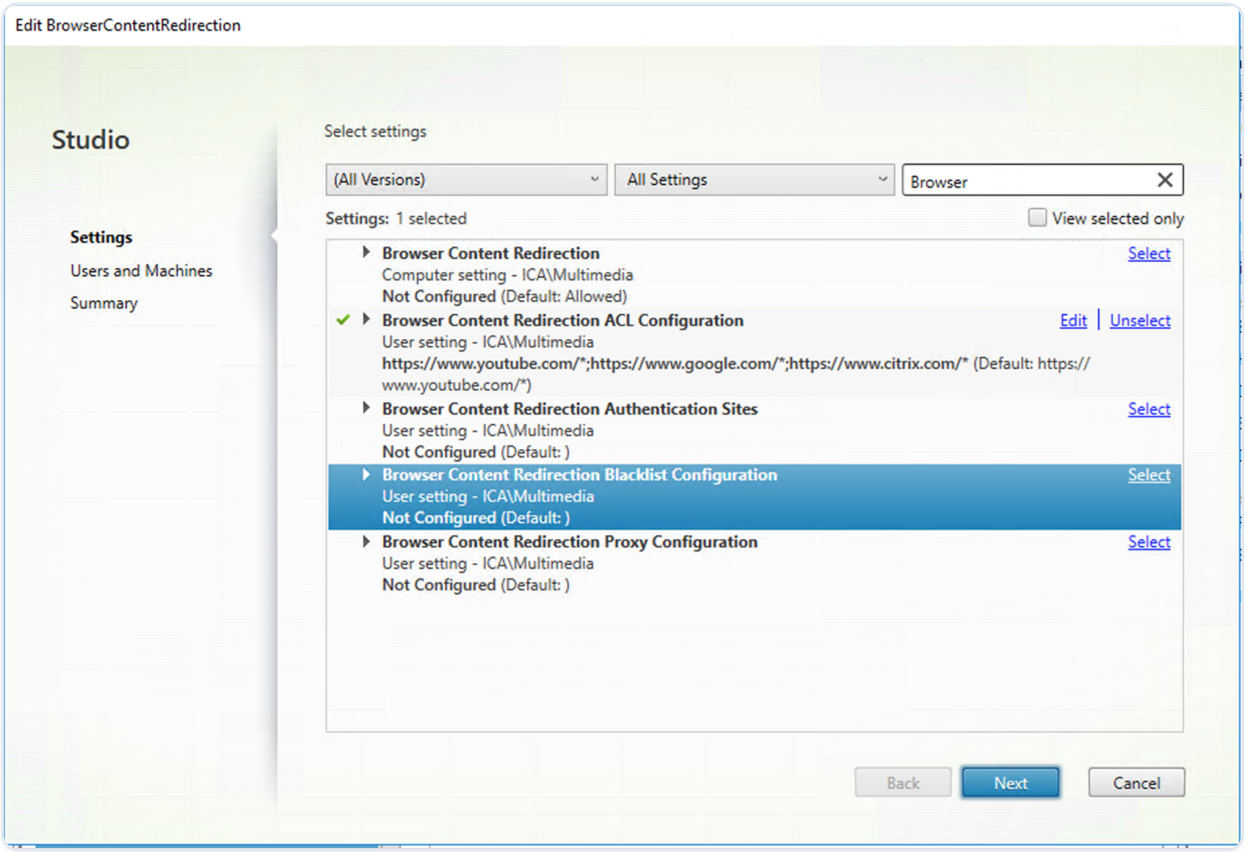


以下は許可される例です。

- `http://www.xyz.com/index.html` (URLにパスを指定することで、より詳細な粒度を実現できます。たとえば、`https://www.xyz.com/sports/index.html`を指定した場合、`index.html`ページのみがリダイレクトされます。)
- `https://www.xyz.com/*`
- `http://www.xyz.com/*videos*`
- `http://*.xyz.com/`
- `http://*.*.com/`



「**ブラウザコンテンツリダイレクトブラックリスト構成**」設定は、特定のURLのリダイレクトを無効にするブロックリストを指定します。



3. サーバーフェッチとクライアントレンダリングを有効にするには、「**ブラウザコンテンツリダイレクトプロキシ構成**」設定を構成します。

   この設定は、ブラウザコンテンツリダイレクト用のVDA上のプロキシ設定の構成オプションを提供します。有効なプロキシアドレスとポート番号、PAC/WPAD URL、またはDirect/Transparent設定で有効にされている場合、Citrix Workspaceアプリは常に最初にサーバーフェッチとクライアントレンダリングを試行します。詳細については、フォールバックメカニズムを参照してください。

   無効になっているか、構成されておらず、デフォルト値を使用している場合、Citrix Workspaceアプリはクライアントフェッチとクライアントレンダリングを試行します。

   デフォルトでは、この設定は**禁止**されています。

   明示的なプロキシの許可パターン:

   `http://<hostname/ip address\>:\<port\>`

   例:

   `http://proxy.example.citrix.com:80 http://10.10.10.10:8080`

   PAC/WPADファイルの許可パターン:

   `http://<hostname/ip address>:<port>/<path>/<Proxy.pac>`

   例: `http://wpad.myproxy.com:30/configuration/pac/Proxy.pac`

   `https://<hostname/ip address>:<port>/<path>/<wpad.dat>`

   例: `http://10.10.10.10/configuration/pac/wpad.dat`

   直接または透過プロキシの許可パターン:

   ポリシーテキストボックスに**DIRECT**と入力します。

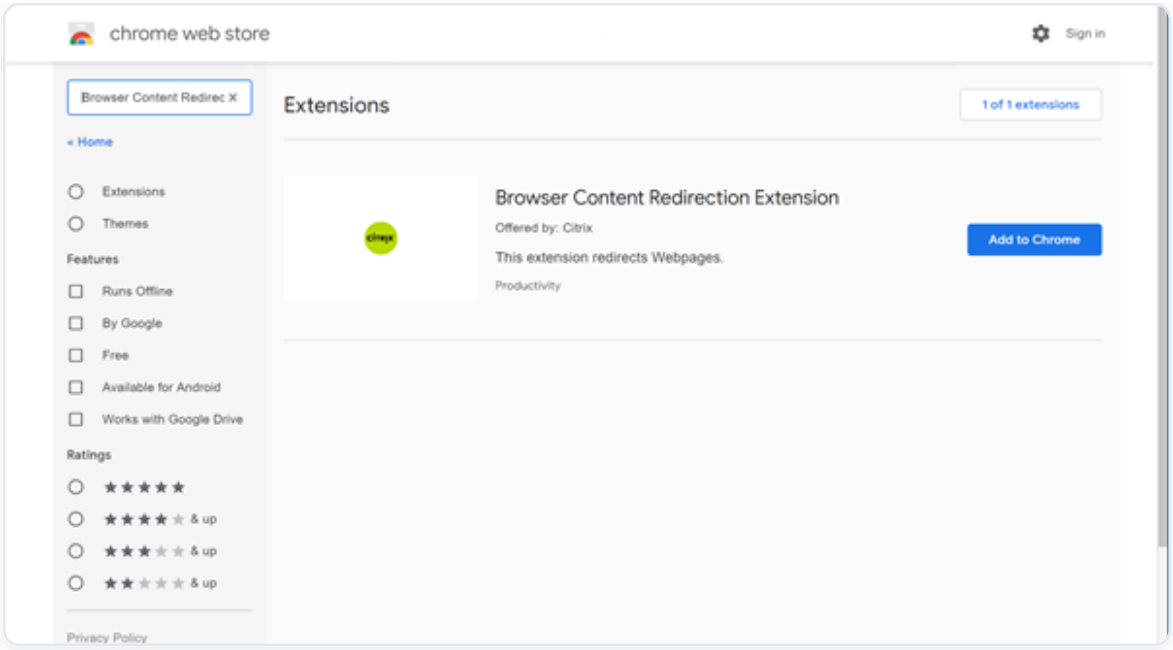
   > **注:**
   >
   > レジストリ値 `HKLM\Software\Citrix\HdxMediastream\WebBrowserRedirectionProxyAddress.` を編集してプロキシを設定することもできます。さらに、`HKLM\Software\Citrix\HdxMediastream\AllowNonTlsPacUri` レジストリ値を使用す

> ると、HTTP経由でのPACファイルのダウンロードを許可するかどうかを決定できます。デフォルト値は0で、HTTPは許可されません。

> レジストリはポリシー設定のオプションを上書きします。関連するレジストリキーのリストについては、ブラウザコンテンツリダイレクトレジストリキーのオーバーライドを参照してください。

4. VDAで**Chromeに追加**をクリックして、ChromeウェブストアからCitrixブラウザコンテンツリダイレクト拡張機能を追加します。これにより、VDA上のブラウザが、ナビゲートされているURLが許可リストまたはブロックリストと一致するかどうかを検出するのに役立ちます。

> **重要:**
>
> クライアントには拡張機能は不要です。VDAにのみ追加してください。
>
> \>
>
> Chrome拡張機能はユーザーごとにインストールされます。拡張機能を追加または削除するためにゴールデンイメージを更新する必要はありません。



フォールバックメカニズム

ブラウザコンテンツリダイレクトプロキシ構成ポリシーを有効にすると、Citrix Workspaceアプリはサーバーフェッチとクライアントレンダリングを試行します。サーバーフェッチとクライアントレンダリングが失敗した場合、クライアントフェッチとクライアントレンダリングにフォールバックします。クライアントマシンがWebサーバーにアクセスできない場合、VDA上のブラウザはページを再読み込みしてサーバー上でレンダリングできます（サーバーフェッチとサーバーレンダリング）。

ブラウザコンテンツリダイレクトのレジストリキーのオーバーライド

> **警告:**
>
> レジストリを誤って編集すると、オペレーティングシステムの再インストールが必要になるような深刻な問題が発生する可能性があります。Citrixは、レジストリエディターの誤った使用によって生じる問題が解決されることを保証できません。レジストリエディターの使用は、お客様ご自身の責任において行ってください。編集する前に必ずレジストリをバックアップしてください。

`HKLM\Software\Citrix\HdxMediastream`

名前	タイプ	値
WebBrowserRedirection	DWORD	1=許可、0=禁止
WebBrowserRedirectionAcl	REG_MULTI_SZ	/
WebBrowserRedirectionProxyAddress	REG_SZ	以下のいずれかのモードに設定すると、サーバーフェッチクライアントレンダリングが有効になります: **明示的プロキシ** - データセンターに単一の明示的プロキシがある場合。**直接または透過的** - プロキシがない場合、または透過的プロキシを使用している場合。**PACファイル** - PACファイルに依存している場合、VDA内のブラウザが指定されたURLのフェッチに適したプ

名前	タイプ	値
		ロキシサーバーを自動的に選択できます。
WebBrowserRedirectionBlacklist	REG_MULTI_SZ	/
AllowNonTlsPacUri	DWORD	HTTP経由でのPACファイルのダウンロードを許可するかどうかを決定します。デフォルト値は0で、HTTPは許可されません。1に設定すると、**HDXWebProxy.exe**はHTTP経由でPACファイルをダウンロードできます（厳密にはHTTPS経由ではありません）。

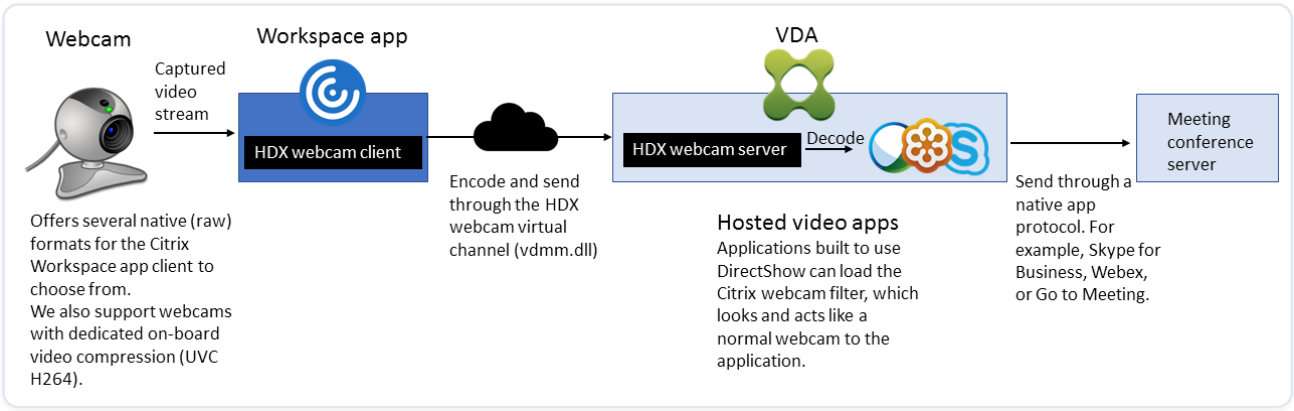
HDX™ウェブカメラビデオ圧縮

概要

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDXウェブカメラビデオ圧縮を使用してウェブカメラを使用できるようになりました。この機能はデフォルトで有効になっています。可能な場合は常にHDXウェブカメラビデオ圧縮を使用することをお勧めします。

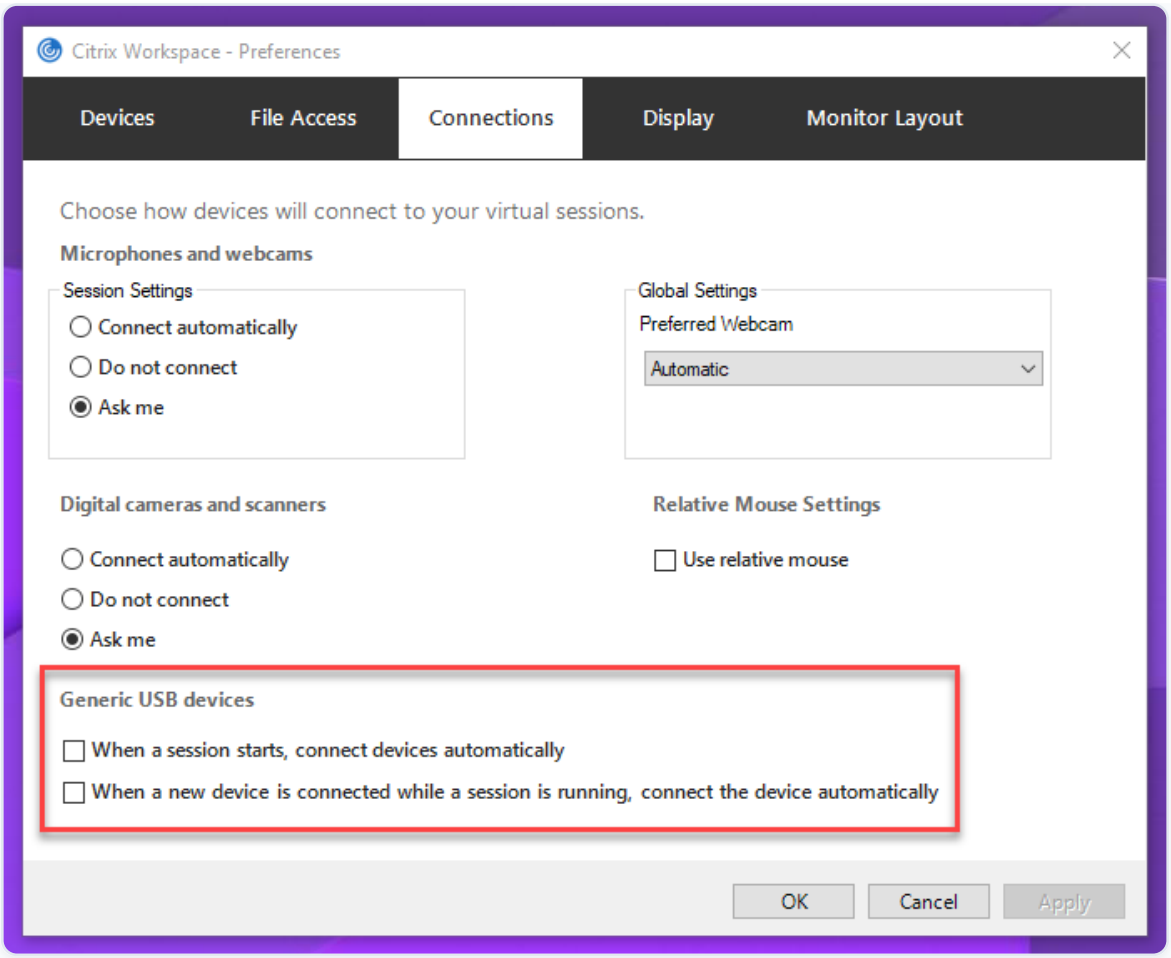
HDXウェブカメラビデオ圧縮は、**最適化された**ウェブカメラモードとも呼ばれます。この種類のウェブカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDXウェブカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオをインターセプトし、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントがウェブカメラとの通信を処理します。その後、クライアントはビデオを適切に表示できるサーバーにのみ送信します。サーバーはウェブカメラを直接処理しませんが、その統合により、デスクトップで同じエクスペリエンスが得られます。Workspaceアプリは、帯域幅を節約し、WANシナリオでの回復性を向上させるためにビデオを圧縮します。



注:

- この機能はAzureマシンでは利用できません。この機能が依存する**videodev**カーネルモジュールがAzureマシンにないためです。
- この機能は、Citrix WorkspaceアプリクライアントからのH.264ビデオのみをサポートします。
- サポートされるウェブカメラの解像度は48x32から1920x1080の範囲です。
- ウェブカメラを使用しているときに、Citrix Workspace™アプリのツールバーから**Generic USB devices**を選択しないでください。そうしないと、予期しない問題が発生する可能性があります。



サポートされているCitrix Workspaceアプリ

HDXウェブカメラビデオ圧縮は、Citrix Workspaceアプリの次のバージョンをサポートしています。

プラットフォーム	プロセッサ
Windows版 Citrix Workspaceアプリ	Windows向けCitrix Workspaceアプリは、XenAppおよびXenDesktop 7.17以降で32ビットおよび64ビットアプリのWebカメラビデオ圧縮をサポートしています。それ以前のバージョンでは、Windows向けCitrix Workspaceアプリは32ビットアプリのみをサポートしています。
Chrome向けCitrix Workspaceアプリ	一部のARM ChromebookはH.264エンコーディングをサポートしていないため、32ビットアプリのみが最適化されたHDX Webカメラビデオ圧縮を使用できます。

完全にテストされたウェブカメラ

ウェブカメラによってフレームレート、明るさ、コントラストのレベルが異なります。Citrix®は、初期機能検証のために以下のウェブカメラを使用しています。

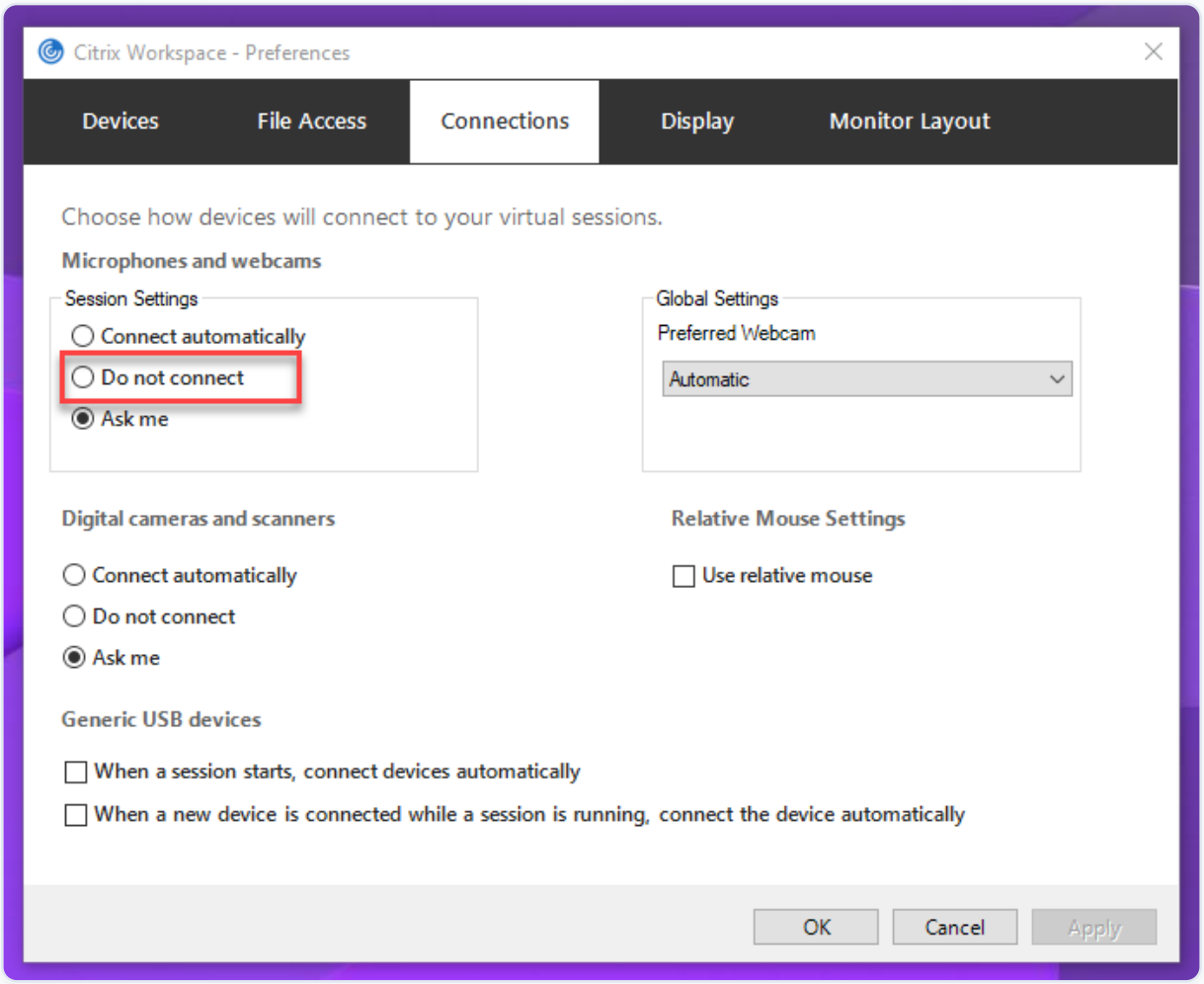
- ロジクール HD ウェブカメラ C270
- ロジクール ウェブカメラ C930e
- マイクロソフト LifeCam-HD3000

構成

この機能はデフォルトで有効になっています。これを使用するには、以下の検証と構成を完了してください。

ヒント:

Citrix Workspaceアプリのユーザーは、Desktop Viewerの「**マイクとWebカメラ**」設定で「**接続しない**」を選択することで、デフォルト設定を上書きできます。



1. VDAのインストールが完了したら、VDAがDelivery Controller™に登録できること、および公開されたLinuxデスクトップセッションがWindows資格情報を使用して正常に起動できることを確認します。
2. VDAがインターネットにアクセスできることを確認し、その後、Webカメラの設定を完了するために`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行します。VDAがインターネットにアクセスできない場合は、手順3に進みます。

   > **注:**
   >
   > `uname -r`とカーネルヘッダーの間でカーネルの不一致が発生する可能性があります。この不一致により、ctxwcamcfg.shスクリプトが失敗します。HDX Webカメラビデオ圧縮を適切に使用するには、**sudo apt-get dist-upgrade**を実行し、VDAを再起動してから、ctxwcamcfg.shスクリプトを再実行します。

   VDAがDebianに展開されている場合、最新のカーネルバージョンで実行されていることを確認してください。そうでない場合は、最新のカーネルバージョンに更新するために次のコマンドを実行します。

   ```
   sudo apt-get update

   sudo apt-get dist-upgrade

   sudo reboot
   ```

   VDAがSUSE 15に展開されている場合、最新のカーネルバージョンに更新し、再起動するために次のコマンドを実行します。

   ```
   zypper up kernel-default

   reboot
   ```

   ctxwcamcfg.shスクリプトは、次のことに役立ちます。

   a. VDAに`kernel-devel`とDynamic Kernel Module Support (DKMS) プログラムをインストールします。

- `kernel-devel`は、対応するバージョンの仮想Webカメラカーネルモジュールを構築するために使用されます。
- DKMSは、仮想Webカメラカーネルモジュールを動的に管理するために使用されます。

> **注:**
>
> RHEL、Rocky Linux、およびCentOSに上記のプログラムをインストールする場合、`ctxwcamcfg.sh`スクリプトは、VDAに次のリポジトリをインストールして有効にします。
>
> - エンタープライズLinux用追加パッケージ (EPEL)
> - RPMフュージョン

b. `<https://github.com/umlaeute/v4l2loopback>`から`v4l2loopback`のオープンソースコードをダウンロードし、DKMSを使用して`v4l2loopback`を管理します。

`v4l2loopback`は、V4L2ループバックデバイスを作成できるカーネルモジュールです。

c. `sudo systemctl restart ctxwcamsd`コマンドを実行します。Linux VDAのWebカメラサービスである`ctxwcamsd`が再起動し、HDX Webカメラビデオ圧縮機能のために`v4l2loopback`カーネルモジュールをロードします。

3. VDAにインターネットアクセスがない場合は、別のマシンでv4l2loopbackカーネルモジュールをビルドし、それをVDAにコピーします。

a. インターネットアクセスがあり、VDAと同じカーネルバージョンを持つマシンを準備します。`uname -r`コマンドはカーネルバージョンを見つけるのに役立ちます。

b. そのマシンで、`sudo mkdir -p /var/xdl`コマンドを実行します。

c. VDAから`/var/xdl/configure_*`を、`/var/xdl/`配下のマシンにコピーします。

d. そのマシンで、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行してカーネルモジュールをビルドします。コマンドが正常に実行されると、`/var/lib/dkms/v4l2loopback/1.81b8df79107d1fbf392fdcbaa051bd227a9c94c1/$(uname -r)/x86_64/module/`パスの下にv4l2loopback.koファイルが作成されます。ctxwcamcfg.shスクリプトの実行時に発生する可能性のあるエラーは無視してください。

e. マシンからv4l2loopback.koをVDAにコピーし、`/opt/Citrix/VDA/lib64/`の下に配置します。

f. VDAで、`sudo systemctl restart ctxwcamsd`コマンドを実行してWebカメラサービスを再起動し、v4l2loopbackカーネルモジュールをロードします。

ドメインに参加していないLinux VDA

概要

ドメインに参加していないVDAは、VDAおよびユーザー認証のためにVDAをActive Directoryドメインに参加させる必要性をなくします。ドメインに参加していないVDAを作成する際、VDAをクラウドコントロールプレーンに登録するために公開鍵と秘密鍵のペアを生成します。したがって、Active Directoryドメインへの参加は不要になります。ユーザーがドメインに参加していないVDAからセッションを起動すると、VDAはユーザーがCitrix Workspaceアプリにログオンするために使用するユーザー名を使用してローカルマッピングアカウントを作成します。VDAは、ローカルマッピングアカウントがSSOおよびセッション再接続に使用するランダムなパスワードを割り当てます。ランダムなパスワードを変更すると、SSOおよびセッション再接続は失敗します。SSOを無効にするには、「非SSO認証」を参照してください。

重要:

- ドメインに参加していないVDAはCitrix DaaSでサポートされています。
  - コントロールプレーンはCitrix DaaS上に展開する必要があります。
  - ドメインに参加していないVDAは、パブリッククラウドまたはオンプレミスのデータセンターに展開できます。Citrix DaaSのコントロールプレーンが、ドメインに参加していないVDAを管理します。
  - ドメインに参加していないVDAを作成するには、Rendezvous V2を有効にする必要があります。Cloud Connectorは、オンプレミスのハイパーバイザーにマシンをプロビジョニングする場合、またはWorkspaceでActive DirectoryをIDプロバイダーとして使用する場合にのみ必要です。
- ドメインに参加していないVDAを作成するには、Machine Creation Services (MCS) と簡易インストールを両方使用できます。詳しくは、「MCSを使用してドメインに参加していないLinux VDAを作成する」および「簡易インストールを使用してドメインに参加していないLinux VDAを作成する」を参照してください。
  - MCSはベアメタルサーバーをサポートしていません。

ドメインに参加していないLinux VDAで利用可能な機能

ドメインに参加していないVDAで指定された属性を持つローカルユーザーを作成する

ドメインに参加していないVDAでホストされているセッションを開くと、VDAはデフォルト属性を持つローカルユーザーを自動的に作成します。VDAは、Citrix Workspace™アプリにログオンするために使用したユーザー名に基づいてローカルユーザーを作成します。ユーザーのユーザー識別子（UID）、グループID（GID）、ホームディレクトリ、ログインシェルなどのユーザー属性を指定することもできます。この機能を使用するには、次の手順を実行します。

1. 機能を有効にするには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg create -k
   "HKLM\Software\Citrix\VirtualDesktopAgent\LocalMappedAccount" -t
   "REG_DWORD" -v "CreateWithUidGid" -d "0x00000001" --force
   ```

2. VDAのインストールパスの下にある`/var/xdl/getuidgid.sh`スクリプトで、次の属性を指定します。

属性	必須またはオプション	説明
`uid`	必須	ユーザー識別子 (UID) は、Linux がシステム上の各ユーザーに割り当てる番号です。これにより、ユーザーがアクセスできるシステムリソースが決まります。
`gid`	必須	グループ識別子 (GID) は、特定のグループを表すために使用される番号です。
`homedir`	オプション	Linux のホームディレクトリは、特定のユーザーのためのディレクトリです。
`shell`	オプション	ログインシェルは、ユーザーがユーザーアカウントにログインしたときに与えられるシェルです。

   以下は、`getuidgid.sh`スクリプトの例です。

注:

スクリプトで指定された属性が有効であることを確認してください。

```
#!/bin/bash



################################################################################
########

#

# Citrix Virtual Apps™ & Desktops For Linux Script: Get uid and gid for
the user

#

# Copyright (c) Citrix Systems, Inc. All Rights Reserved.

#



export LC_ALL="en_US.UTF-8"



function get_uid_gid_for_user()

{

echo "uid:12345"

echo "gid:1003"

echo "homedir:/home/$1"

echo "shell:/bin/sh"

}
```

```
get_uid_gid_for_user $1
```

非SSO認証

デフォルトでは、Linux VDA はシングルサインオン (SSO) が有効になっています。ユーザーは、Citrix Workspace アプリと VDA セッションに、1組の資格情報を使用してログオンします。 ユーザーが異なる資格情報を使用して VDA セッションにログオンするようにするには、Linux VDA で SSO を無効にします。詳細については、「非SSO認証」を参照してください。

Azure Active Directory を使用した認証

Azure に展開するドメイン非参加型 VDA は、AAD ID サービスと統合してユーザー認証を提供します。詳細については、「Azure Active Directory を使用した認証」を参照してください。

ランデブー V2

ドメイン非参加型 VDA は、Citrix Cloud Connector をバイパスするために Rendezvous V2 を使用することがサポートされています。詳細については、「Rendezvous V2」を参照してください。

ポリシーサポートリスト

Linux VDAポリシーサポートリスト

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントのローカル時間を使用	UseLocalTimeOfClient	ユーザー	ICA\タイムゾーン制御	**サーバーのタイムゾーンを使用**
ICAラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICAラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15
アイドル接続のICAラウンドトリップ計算	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザーモニタリング	無効 (0)

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
セッション全体の帯域幅制限	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限（パーセント）	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクトの帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクト帯域幅の割合	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅制限	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅制限の割合	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限の割合	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
WebSockets接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocketsポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008
WebSockets信頼済みオリジンサーバーリスト	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICA®キープアライブ	SendICAKeepAlives	コンピューター	**ICAキープアライブ**	ICAキープアライブメッセージを送信しない (0)
ICAキープアライブタイムアウト	ICAKeepAliveTimeout	コンピューター	**ICAキープアライブ**	60秒
ICAリスナーポート番号	IcaListenerPortNumber	コンピューター	ICA	1494
HDX™アダプティブトランスポート	HDXoverUDP	コンピューター	ICA	優先(2)
ランデブープロトコル	RendezvousProtocol	コンピューター	ICA	禁止
セッションの信頼性接続	AcceptSessionReliabilityConnections	コンピューター	ICA\セッションの信頼性	許可(1)
再接続UIの透明度レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\自動クライアント再接続	80%
セッションの信頼性ポート番号	SessionReliabilityPort	コンピューター	ICA\セッションの信頼性	2598

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
セッションの信頼性タイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッションの信頼性	180 秒
自動クライアント再接続	AllowAutoClientReconnect	ユーザー	ICA\自動クライアント再接続	許可 (1)
クライアント自動再接続認証	AllowAutoClientReconnectAuthentication	コンピューター	ICA\自動クライアント再接続	無効 (0)
クライアント自動再接続ログ	AllowAutoClientReconnectLogging	コンピューター	ICA\自動クライアント再接続	無効 (0)
クライアント自動再接続タイムアウト	AllowAutoClientReconnectTimeout	コンピューター	ICA\自動クライアント再接続	120秒
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDFユニバーサルプリンターの自動作成	AutoCreatePDFPrinter	ユーザー	印刷	無効 (0)
プリンタードライバーのマッピングと互換性	DriverMappingList	ユーザー	印刷	`"Microsoft XPS Document Writer *, Deny;Send to Microsoft OneNote *, Deny"`

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントクリップボードリダイレクト	AllowClipboardRedir	ユーザー	クリップボード	許可 (1)
クリップボードのクライアントからセッションへの転送サイズを制限	LimitClipboardTransferC2H	ユーザー	ICA	無効 (0)
クリップボードセッションからクライアントへの転送サイズを制限	LimitClipboardTransferH2C	ユーザー	ICA	無効 (0)
クリップボードリダイレクトの帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限の割合	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0
クライアントのクリップボード書き込みを制限	RestrictClientClipboardWrite	ユーザー	クリップボード	無効 (0)
クライアントのクリップボード書き込みで許可される形式	ClientClipboardWriteAllowedFormats	ユーザー	クリップボード	該当なし
セッションクリップボードへの書き込みを制限する	RestrictSessionClipboardWrite	ユーザー	クリップボード	無効 (0)

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
セッションクリップボードへの書き込み許可形式	SessionClipboardWriteAllowedFormats	ユーザー	クリップボード	該当なし
クライアントUSBデバイスリダイレクト	AllowUSBRedir	ユーザー	USB	禁止 (0)
クライアントUSBデバイスリダイレクト規則	USBDeviceRules	ユーザー	USB	"\0"
動画圧縮	MovingImageCompressionConfiguration	ユーザー	シンワイヤー	有効 (1)
追加の色圧縮	ExtraColorCompression	ユーザー	シンワイヤー	無効 (0)
ターゲットの最小フレームレート	TargetedMinimumFramesPerSecond	ユーザー	シンワイヤー	10 フレーム/秒
ターゲットフレームレート	FramesPerSecond	ユーザー	シンワイヤー	30 フレーム/秒
視覚的品質	VisualQuality	ユーザー	シンワイヤー	中 (3)
圧縮にビデオコーデックを使用	VideoCodec	ユーザー	シンワイヤー	優先する場合に使用 (3)
ビデオコーデックにハードウェアエンコードを使用	UseHardwareEncodingForVideoCodec	ユーザー	シンワイヤー	有効 (1)

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
視覚的にロスレスな圧縮を許可する	AllowVisuallyLosslessCompression	ユーザー	シンワイヤー	無効 (0)
3Dグラフィックワークロード向けに最適化する	OptimizeFor3dWorkload	ユーザー	シンワイヤー	無効 (0)
シンプルなグラフィックの優先色深度	PreferredColorDepth	ユーザー	シンワイヤー	24ビット/ピクセル (1)
グラフィックスステータスインジケーター	DisplayLosslessIndicator	ユーザー	シンワイヤー	無効 (0)
オーディオ品質	SoundQuality	ユーザー	オーディオ	高 – 高品位オーディオ (2)
クライアントマイクのリダイレクト	AllowMicrophoneRedir	ユーザー	オーディオ	許可 (1)
最大セッション数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオン許容数	ConcurrentLogonsTolerance	コンピューター	負荷管理	2
コントローラーの自動更新を有効にする	EnableAutoUpdateOfControllers	コンピューター	仮想配信エージェント設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
最大Speex品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブの自動接続	AutoConnectDrives	ユーザー	ファイルリダイレクト\CDM	有効 (1)
クライアント光学ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアント固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアントフロッピーディスクドライブ	AllowFloppyDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアントネットワークドライブ	AllowNetworkDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアントリムーバブルドライブ	AllowRemoveableDrives	ユーザー	ファイルリダイレクト\CDM	許可 (1)
クライアントドライブリダイレクト	AllowDriveRedir	ユーザー	ファイルリダイレクト\CDM	許可 (1)
読み取り専用クライアントドライブアクセス	ReadOnlyMappedDrive	ユーザー	ファイルリダイレクト\CDM	無効 (0)
キーボードの自動表示	AllowAutoKeyboardPopUp	ユーザー	MRVC	無効 (0)
デスクトップとクライアント間のファイル転送を許可する	AllowFileTransfer	ユーザー	ファイル転送	許可

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
デスクトップからファイルをダウンロードする	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロードする	AllowFileUpload	ユーザー	ファイル転送	許可
セッションアイドルタイマー	EnableSessionIdleTimer	ユーザー	セッションタイマー	有効 (1)
セッションアイドルタイマー間隔	SessionIdleTimerInterval	ユーザー	セッションタイマー	1440分
切断されたセッションタイマー	EnableSessionDisconnectTimer	ユーザー	セッションタイマー	無効 (0)
切断されたセッションタイマー間隔	SessionDisconnectTimerPeriod	ユーザー	セッションタイマー	1440分
ブラウザコンテンツリダイレクト	WebBrowserRedirection	コンピューター	ICA\マルチメディア	許可
ブラウザコンテンツリダイレクトACL構成	WebBrowserRedirectionAcl	ユーザー	ICA\マルチメディア	`https://www.youtube.com/*`
ブラウザコンテンツリダイレクト認証サイト	WebBrowserRedirectionAuthenticationSites	ユーザー	ICA\マルチメディア	ヌル
ブラウザコンテンツリダイレクトブラックリスト構成	WebBrowserRedirectionBlacklist	ユーザー	ICA\マルチメディア	ヌル

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
ブラウザコンテンツリダイレクトプロキシ構成	WebBrowserRedirectionProxy	ユーザー	ICA\マルチメディア	ヌル
セッションウォーターマークを有効にする	EnableSessionWatermark	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	無効
クライアントIPアドレスを含める	IncludeClientIPAddress	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	無効
接続時間を含める	IncludeConnectTime	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	無効
ログオンユーザー名を含める	IncludeLogonUsername	ユーザー	ICA\セッション透かし\透かしコンテンツ	有効
VDA ホスト名を含める	IncludeVDAHostname	ユーザー	ICA\セッション透かし\透かしコンテンツ	有効
VDA IPアドレスを含める	IncludeVDAIPAddress	ユーザー	ICA\セッション透かし\透かしコンテンツ	無効
セッションウォーターマークのスタイル	WatermarkStyle	ユーザー	ICA\セッション透かし\透かしスタイル	複数

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
ウォーターマークの透明度	WatermarkTransparency	ユーザー	ICA\セッション透かし\透かしスタイル	17
ウォーターマークカスタムテキスト	WatermarkCustomText	ユーザー	ICA\セッションウォーターマーク\ウォーターマークコンテンツ	有効なカスタムテキスト構成は、`<font=Arial Unicode MS><domain><newline><fontsize=47><username><newline><fontzoom=120>Citrix<newline><style=tile>Najing <clientip><newline><rotation=45><serverip><position=bottomleft>` のようになります。詳細については、「セッションウォーターマーク」を参照してください。
セッションメトリックの収集	UsageDataCollectionThroughClient	コンピューター	ICA	有効 (1)
動的ウィンドウプレビュー	EnableDynamicPreview	コンピューター	ICA\グラフィックス	この設定は、シームレスウィンドウの表示を有効または無効にします。この設定を有効にす

STUDIOポリシー	キー名	タイプ	モジュール	デフォルト値
				ると、公開アプリケーションのウィンドウで画面上のコンテンツをプレビューできます。たとえば、Windows 用 Citrix Workspace アプリを使用して Google Chrome セッションの複数のウィンドウを開き、タスクバーの Chrome アイコンにカーソルを合わせると、コンテンツをプレビューできます。詳細については、「動的ウィンドウプレビュー」を参照してください。デフォルトでは、この設定は**有効 (1)** です。

注:

Windows Virtual Delivery Agent (VDA) のみが、User Datagram Protocol (UDP) 経由のオーディオをサポートしています。Linux VDA はサポートしていません。詳細については、「User Datagram Protocol (UDP) 経由のオーディオリアルタイム転送」を参照してください。

> Citrix Studio でセッション接続タイマーを構成するには、以下の Citrix ポリシー設定を使用できます。

- **セッションアイドルタイマー**: アイドルセッションに時間制限を適用するかどうかを決定します。
- **セッションアイドルタイマー間隔**: アイドル状態のセッションに時間制限を設定します。**セッションアイドルタイマー**が**有効**に設定されており、設定された時間内にアクティブなセッションがユーザー入力を受信しなかった場合、セッションは切断されます。
- **切断されたセッションのタイマー**: 切断されたセッションに時間制限を適用するかどうかを決定します。
- **切断されたセッションのタイマー間隔**: 切断されたセッションがログオフされるまでの間隔を設定します。

ポリシー設定を更新する際は、展開全体でそれらが一貫していることを確認してください。

アイドルセッションの時間制限が期限切れになると、警告メッセージが表示されます。例については、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にユーザー入力を提供してください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2411/media/warning-message-idle-session-timeout.png)

以下のポリシーは、Citrix Studioバージョン7.12以降で構成できます。

- MaxSpeexQuality

  値 (整数):

  デフォルト値: 5

  詳細:

  オーディオ品質が中または低の場合（ポリシー「オーディオ品質」を参照）、オーディオリダイレクトはSpeexコーデックでオーディオデータをエンコードします。Speexは非可逆コーデックであり、入力音声信号の忠実度を犠牲にして圧縮を実現します。他の音声コーデックとは異なり、品質とビットレートの間のトレードオフを制御することが可能です。Speexのエンコードプロセスは、ほとんどの場合、0から10の範囲の品質パラメーターによって制御されます。品質が高いほど、ビットレートも高くなります。

最大Speex品質は、オーディオ品質と帯域幅制限（ポリシー「オーディオリダイレクト帯域幅制限」を参照）に従って、オーディオデータをエンコードするための最適なSpeex品質を選択します。オーディオ品質が中の場合、エンコーダーは広帯域モードになり、サンプリングレートが高くなります。オーディオ品質が低い場合、エンコーダーは狭帯域モードになり、サンプリングレートが低くなります。同じSpeex品質でも、異なるモードでは異なるビットレートになります。最適なSpeex品質とは、最大値が以下の条件を満たす場合です。

- 最大Speex品質以下であること。
- そのビットレートが帯域幅制限以下であること。

関連設定: オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

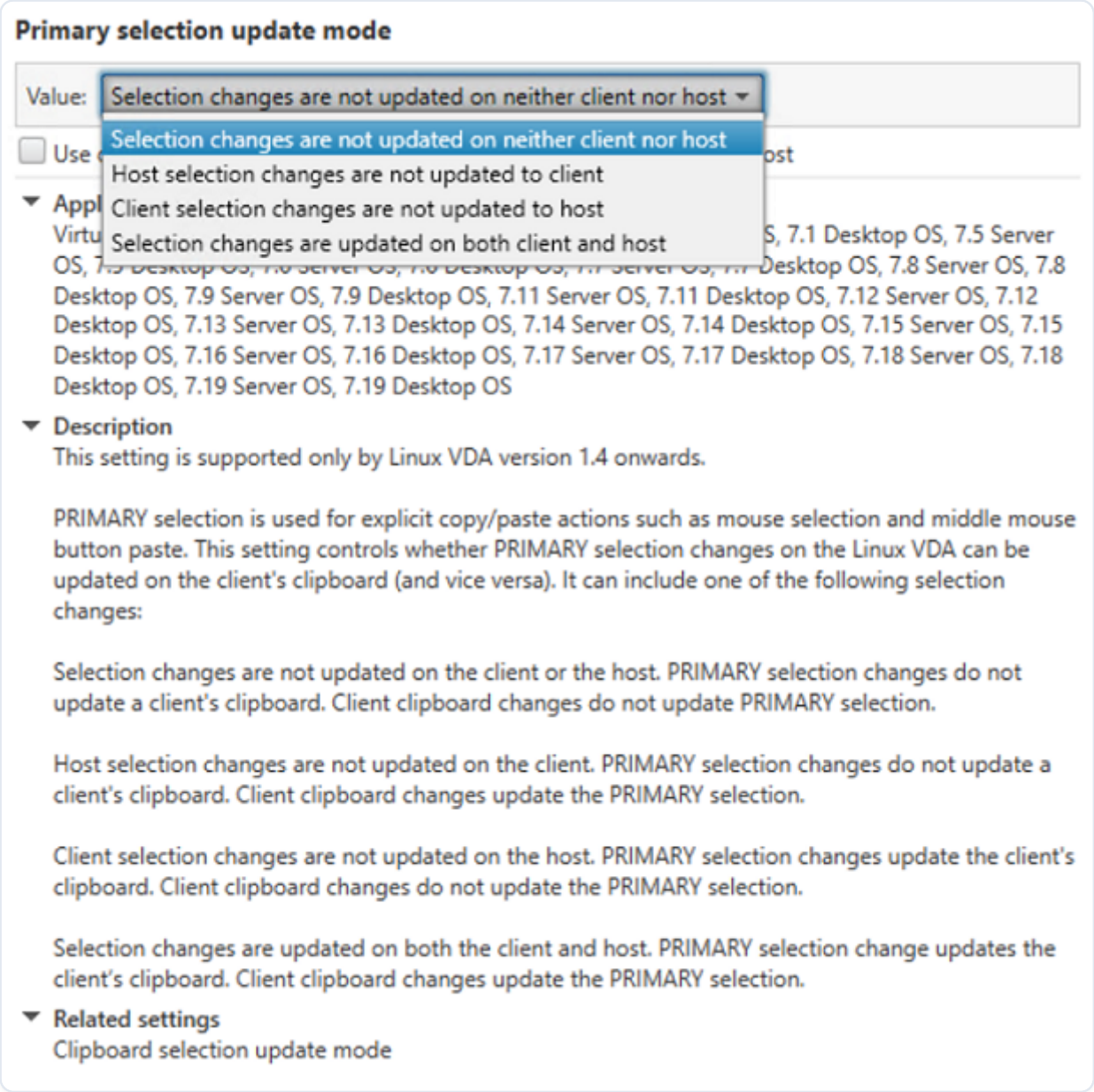
値 (列挙型):

既定値: 3

詳細:

プライマリ選択は、データを選択し、マウスの中央ボタンを押して貼り付けるときに使用されます。

このポリシーは、Linux VDA とクライアント上でのプライマリ選択の変更が相互にクリップボードを更新できるかどうかを制御します。次の4つの値オプションがあります。



- **クライアントとホストのどちらでも選択の変更は更新されない**

  Linux VDA 上でのプライマリ選択の変更は、クライアント上のクリップボードを更新しません。クライアント上でのプライマリ選択の変更は、Linux VDA 上のクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されない**

  Linux VDA 上でのプライマリ選択の変更は、クライアント上のクリップボードを更新しません。クライアント上でのプライマリ選択の変更は、Linux VDA 上のクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されない**

Linux VDA 上でのプライマリ選択の変更は、クライアント上のクリップボードを更新します。クライアント上でのプライマリ選択の変更は、Linux VDA 上のクリップボードを更新しません。

- **クライアントとホストの両方で選択の変更が更新される**

  Linux VDA 上でのプライマリ選択の変更は、クライアント上のクリップボードを更新します。クライアント上でのプライマリ選択の変更は、Linux VDA 上のクリップボードを更新します。このオプションが既定値です。

関連設定: クリップボード選択更新モード

- ClipboardSelectionUpdateMode

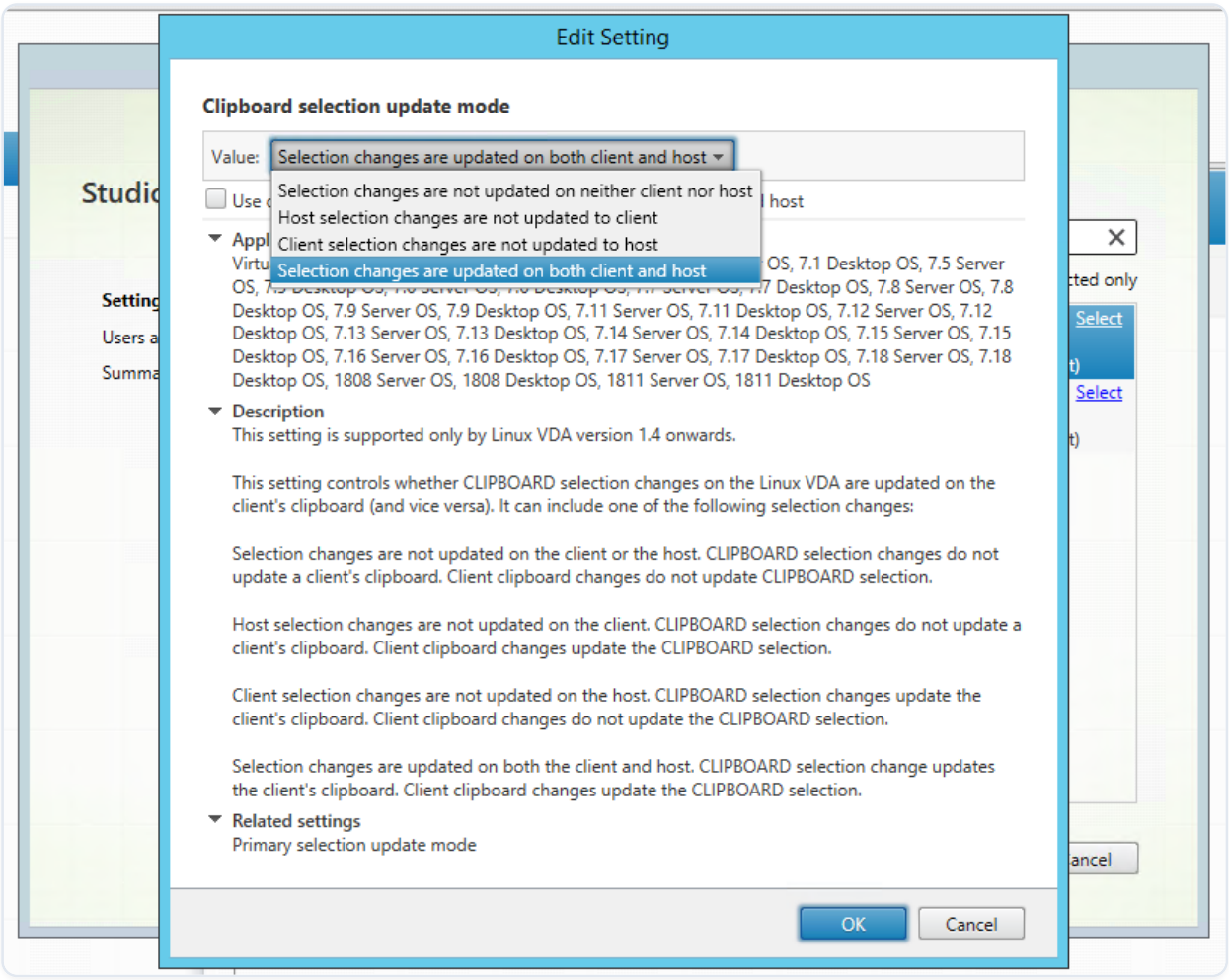
  値 (列挙型):

  既定値: 3

  詳細:

  クリップボード選択は、データをいくつか選択し、ショートカットメニューから「コピー」を選択するなどして、明示的にクリップボードへのコピーを要求する場合に使用されます。クリップボード選択は主にMicrosoft Windowsのクリップボード操作に関連していますが、プライマリ選択はLinux固有のものです。

  このポリシーは、Linux VDAとクライアント上でのクリップボード選択の変更が、互いのクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。



- **クライアントとホストのどちらにも選択の変更は更新されません**

  Linux VDA上でのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアント上でのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されません**

  Linux VDA上でのクリップボード選択の変更は、クライアント上のクリップボードを更新しません。クライアント上でのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されません**

  Linux VDA上でのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアント上でのクリップボード選択の変更は、Linux VDA上のクリップボードを更新しません。

- **クライアントとホストの両方で選択の変更が更新されます**

Linux VDA上でのクリップボード選択の変更は、クライアント上のクリップボードを更新します。クライアント上でのクリップボード選択の変更は、Linux VDA上のクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

> **注:**
>
> Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピー＆ペーストの動作を制御するには、クリップボード選択更新モードとプライマリ選択更新モードの両方を同じ値に設定することをお勧めします。

印刷

このセクションには、次のトピックが含まれています。

- 印刷のベストプラクティス
- PDF印刷

印刷のベストプラクティス

この記事では、印刷のベストプラクティスに関する情報を提供します。

インストール

Linux VDAには、**cups**フィルターと**foomatic**フィルターの両方が必要です。これらのフィルターは、VDAのインストール時にインストールされます。ディストリビューションに基づいて、フィルターを手動でインストールすることもできます。

印刷ポリシー設定

クライアントプリンターリダイレクト

この設定により、クライアントプリンターをVDAセッションにマップするかどうかを決定できます。デフォルトでは、クライアントプリンターのマッピングは許可されています。

クライアントプリンターを自動作成

この設定では、VDAセッションにマップできるクライアントプリンターを指定します。デフォルトでは、**すべてのクライアントプリンターを自動作成**に設定されており、すべてのクライアントプ

リンターがVDAセッションにマップされます。この設定の詳細については、Citrix Virtual Apps and Desktopsのドキュメントにあるクライアントプリンターの自動作成を参照してください。

PDFユニバーサルプリンターの自動作成

PDF印刷機能を使用するには、このポリシーを**有効**に設定します。

プリンタードライバーのマッピングと互換性

Citrixは3種類のユニバーサルプリンタードライバー（postscript、pcl5、pcl6）を提供しています。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合があります。この場合、以前のリリースでは、~/.CtxlpProfile$CLIENT_NAME構成ファイルを編集するしかありませんでした。バージョン1906以降では、代わりにCitrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成することを選択できます。

Citrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成するには：

1. **プリンタードライバーのマッピングと互換性**ポリシーを選択します。
2. 「**追加**」をクリックします。
3. **ドライバー名**にクライアントプリンターのドライバー名を入力します。Citrix Workspace™アプリ for Linuxを使用している場合は、代わりにプリンター名を入力します。
4. 必要に応じて、次のアクションを実行します。

    - クライアントプリンターをVDAセッションにリダイレクトし、ユニバーサルプリントドライバーのみを使用できるようにするには、**許可**または**ユニバーサルドライバーのみで作成**を選択します。
    - 指定されたドライバー名のクライアントプリンターがVDAセッションにリダイレクトされないようにするには、**作成しない**を選択します。
    - サーバーアプリケーションが、サーバーと同じドライバーを持ちながらドライバー名が異なるクライアントプリンターにアクセスできるようにするには、**Replace with** を選択し、VDA上のドライバーファイルの絶対パスを入力します。

    プリンタードライバーのマッピングと互換性ポリシー(/en-us/linux-virtual-delivery-agent/2411/media/printer-driver-mapping-and-compatibili-policy1.png)

> **注:**
>
> PPDドライバーファイルのみがサポートされています。

使用方法

公開デスクトップと公開アプリケーションの両方から印刷できます。すべてのクライアントプリンターをVDAセッションにマッピングできます。デスクトップとアプリケーションではプリンター名が異なります。

- 公開デスクトップの場合:

  `<client printer name>:$CLIENT_NAME:dsk$SESSION_ID`

- 公開アプリケーションの場合:

  `<client printer name>:$CLIENT_NAME:app$SESSION_ID`

> **注:**
>
> 同じユーザーが公開デスクトップと公開アプリケーションの両方を開いた場合、両方のプリンターがセッションで利用可能になります。公開アプリケーションセッションでデスクトッププリンターに印刷したり、公開デスクトップでアプリケーションプリンターに印刷したりすることはできません。

トラブルシューティング

印刷できません

印刷が正しく機能しない場合は、印刷デーモン **ctxlpmngt** と **CUPs framework** を確認してください。

印刷デーモン **ctxlpmngt** はセッションごとのプロセスであり、セッション中は実行されている必要があります。印刷デーモンが実行されていることを確認するには、次のコマンドを実行します。**ctxlpmngt** が実行されていない場合は、コマンドラインから手動で **ctxlpmngt** を開始します。

```
ps –ef | grep ctxlpmngt
```

印刷がまだ機能しない場合は、**CUPS** フレームワークを確認してください。**ctxcups** サービスはプリンター管理に使用され、Linux CUPS フレームワークと通信します。これはマシンごとに1つのプロセスであり、次のコマンドを実行して確認できます。

```
systemctl status ctxcups
```

CUPSログを収集するための追加手順

CUPSログを収集するには、次のコマンドを実行してCUPSサービスファイルを構成します。そうしないと、CUPSログは **hdx.log** に記録されません。

```
sudo systemctl stop cups



sudo vi /etc/systemd/system/printer.target.wants/cups.service



PrivateTmp=false



sudo systemctl start cups



sudo systemctl daemon-reload
```

注:

この構成は、問題が発生したときに完全な印刷ログを収集するためにのみ行われます。通常の状況では、この構成はCUPSのセキュリティを破るため推奨されません。

印刷出力が文字化けする

互換性のないプリンタードライバーは、文字化けした出力を引き起こす可能性があります。ユーザーごとのドライバー構成が利用可能で、**~/.CtxlpProfile$CLIENT_NAME** 構成ファイルを編集することで構成できます。

```
[DEFAULT_PRINTER]



printername=
```

```
model=


ppdpath=


drivertype=
```

重要:

printername は、現在のクライアント側のデフォルトプリンターの名前を含むフィールドです。これは読み取り専用の値です。編集しないでください。

ppdpath、**model**、および **drivertype** の各フィールドは同時に設定できません。**マップされたプリンター**に対しては1つしか有効にならないためです。

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合、**model=** オプションを使用して、ネイティブプリンタードライバーのモデルを構成します。**lpinfo** コマンドを使用して、プリンターの現在のモデル名を見つけることができます。

```
lpinfo –m


…


xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0


xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
```

```
```

その後、プリンターに合わせてモデルを設定できます。

```
model=xerox/ph3115.ppd.gz
```

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合、ネイティブプリンタードライバーのPPDファイルパスを構成します。**ppdpath** の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  たとえば、/home/tester/NATIVE_PRINTER_DRIVER.ppd の下に **ppd driver** があります。

  ```
  ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
  ```

- Citrix が提供するユニバーサルプリンタードライバーには3種類あります（postscript、pcl5、pcl6）。プリンターのプロパティに基づいてドライバーの種類を構成できます。

  たとえば、クライアントのデフォルトプリンタードライバーの種類がPCL5の場合、**drivertype** を次のように設定します。

  ```
  drivertype=pcl5
  ```

出力サイズがゼロ

異なる種類のプリンターを試してください。また、CutePDFやPDFCreatorのような仮想プリンターを試して、この問題がプリンタードライバーに関連しているかどうかを確認してください。

印刷ジョブは、クライアントのデフォルトプリンターのプリンタードライバーに依存します。現在アクティブなドライバーの種類を特定することが重要です。クライアントプリンターがPCL5ドライバーを使用しているのに、Linux VDAがPostscriptドライバーを選択した場合、問題が発生する可能性があります。

プリンタードライバーの種類が正しい場合、次の手順を実行して問題を特定できます。

1. 公開されたデスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME** コマンドを実行します。
3. Linux VDAにスプールファイルを保存するには、次のフィールドを追加します。

   ```
   deletespoolfile=no
   ```

4. ログオフして再度ログオンし、構成の変更を読み込みます。
5. ドキュメントを印刷して問題を再現します。印刷後、スプールファイルは`/var/spool/cups-ctx/$logon_user/$spool_file`に保存されます。
6. スプールが空かどうかを確認します。スプールファイルがゼロの場合、問題が発生しています。詳細については、Citrixサポートに問い合わせてください（印刷ログを提供してください）。
7. スプールサイズがゼロでない場合は、ファイルをクライアントにコピーします。スプールファイルの内容は、クライアントのデフォルトプリンターのプリンタードライバーの種類によって異なります。マップされたプリンター（ネイティブ）ドライバーがPostScriptの場合、スプールファイルはLinux OSで直接開くことができます。内容が正しいかどうかを確認してください。

   スプールファイルがPCLの場合、またはクライアントOSがWindowsの場合は、スプールファイルをクライアントにコピーし、別のプリンタードライバーを使用してクライアント側のプリンターで印刷します。
8. **マップされたプリンター**を別のプリンタードライバーを使用するように変更します。次の例では、PostScriptクライアントプリンターを例として使用しています。

   a. アクティブなセッションにログオンし、クライアントデスクトップでブラウザーを開きます。

   b. 印刷管理ポータルを開きます。

   ```
   localhost:631
   ```

   c. **マップされたプリンター** `CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID`を選択し、**プリンターの変更**を選択します。この操作には管理者権限が必要です。

   d. **cups-ctx**接続を保持し、**続行**をクリックしてプリンタードライバーを変更します。

   e. **製造元**および**モデル**フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。

   f. このプロセスが成功した場合は、`.CtxlpProfile$CLIENT_NAME`でドライバーのPPDファイルパスを構成して、マップされたプリンターが新しく選択されたドライバーを使用できるようにします。

既知の問題

Linux VDAでの印刷中に、次の問題が特定されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブなものに設定してください。

大容量ドキュメントの印刷パフォーマンスの低下

ローカルクライアントプリンターで大容量ドキュメントを印刷すると、ドキュメントはサーバー接続を介して転送されます。低速な接続では、転送に時間がかかる場合があります。

他のセッションから見えるプリンターおよび印刷ジョブの通知

Linuxには、Windowsオペレーティングシステムと同じセッションの概念がありません。そのため、すべてのユーザーがシステム全体の通知を受け取ります。これらの通知は、CUPS構成ファイル: **/etc/cups/cupsd.conf** を変更することで無効にできます。

ファイルに構成されている現在のポリシー名を見つけます。

`DefaultPolicy default`

ポリシー名が *default* の場合は、以下の行をデフォルトポリシーのXMLブロックに追加します。

```
<Policy default>

     # Job/subscription privacy...

     JobPrivateAccess default

     JobPrivateValues default

     SubscriptionPrivateAccess default
```

```
    SubscriptionPrivateValues default

    … …

    <Limit Create-Printer-Subscription>

        Require user @OWNER

        Order deny,allow

    </Limit>

    <Limit All>

        Order deny,allow

    </Limit>

</Policy>
```

PDF印刷

PDF印刷をサポートするCitrix Workspace™アプリのバージョンを使用すると、Linux VDAセッション内から変換されたPDFを印刷できます。セッションの印刷ジョブは、Citrix Workspaceアプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、任意のPDFビューアを使用してPDFを開き、任意のプリンターで印刷できます。

Linux VDAは、Citrix Workspaceアプリの以下のバージョンでPDF印刷をサポートしています。

- Citrix Receiver for HTML5バージョン2.4～2.6.9、Citrix Workspaceアプリ1808 for HTML5以降
- Citrix Receiver for Chromeバージョン2.4～2.6.9、Citrix Workspaceアプリ1808 for Chrome以降
- Citrix Workspaceアプリ1905 for Windows以降

構成

PDF印刷をサポートするCitrix Workspaceアプリのバージョンを使用する以外に、Citrix Studioで以下のポリシーを設定します。

- **Client Printer Redirection**を**Allowed**に設定します（デフォルトは**Allowed**）。
- **Auto-create PDF Universal Printer**を**Enabled**に設定します（デフォルトは**Disabled**）。
- **クライアントプリンターの自動作成** を **すべてのクライアントプリンターを自動作成** に設定します。

これらのポリシーを有効にすると、起動したセッション内で**Print**をクリックしたときに、プリンターを選択するための印刷プレビューがローカルマシンに表示されます。デフォルトプリンターの設定については、Citrix Workspaceアプリのドキュメントを参照してください。

リモートPCアクセス

概要

Remote PC Accessは、Citrix Virtual Apps and Desktopsの拡張機能です。これにより、組織は従業員が物理的なオフィスPCに安全にリモートアクセスできるように簡単に設定できます。ユーザーがオフィスPCにアクセスできれば、仕事に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

Remote PC Accessは、仮想デスクトップとアプリケーションを提供するCitrix Virtual Apps™ and Desktopsと同じコンポーネントを使用します。Remote PC Accessの展開と構成の要件とプロセスは、Citrix Virtual Apps and Desktopsの展開に必要な要件とプロセスと同じです。この一貫性により、一貫した統合された管理エクスペリエンスが提供されます。ユーザーは、Citrix HDXを使用し

てリモートオフィスPCセッションを配信することで、最高のユーザーエクスペリエンスを得られます。

詳細については、Citrix Virtual Apps and DesktopsドキュメントのRemote PC Accessを参照してください。

考慮事項

以下の考慮事項は、Linux VDAに固有のものです。

- 物理マシンでは、Linux VDAは非3Dモードでのみ使用してください。NVIDIAドライバーの制限により、HDX™ 3Dモードが有効になっている場合、PCのローカル画面をブラックアウトできません。この画面が表示されることは、潜在的なセキュリティリスクです。
- 物理Linuxマシンには、シングルセッションOSタイプのマシンカタログを使用します。
- Linuxマシンでは、自動ユーザー割り当ては利用できません。自動ユーザー割り当てでは、ユーザーがPCにローカルでログオンすると、自動的にマシンに割り当てられます。このログオンは、管理者の介入なしに行われます。クライアント上のCitrix Workspace™アプリは、Remote PC Accessデスクトップセッション内で、ユーザーがオフィスPC上のアプリケーションとデータにアクセスするのに役立ちます。
- ユーザーがすでにローカルでPCにログオンしている場合、StoreFront™からPCを起動しようとすると失敗します。
- 省電力オプションはLinuxマシンでは利用できません。

構成

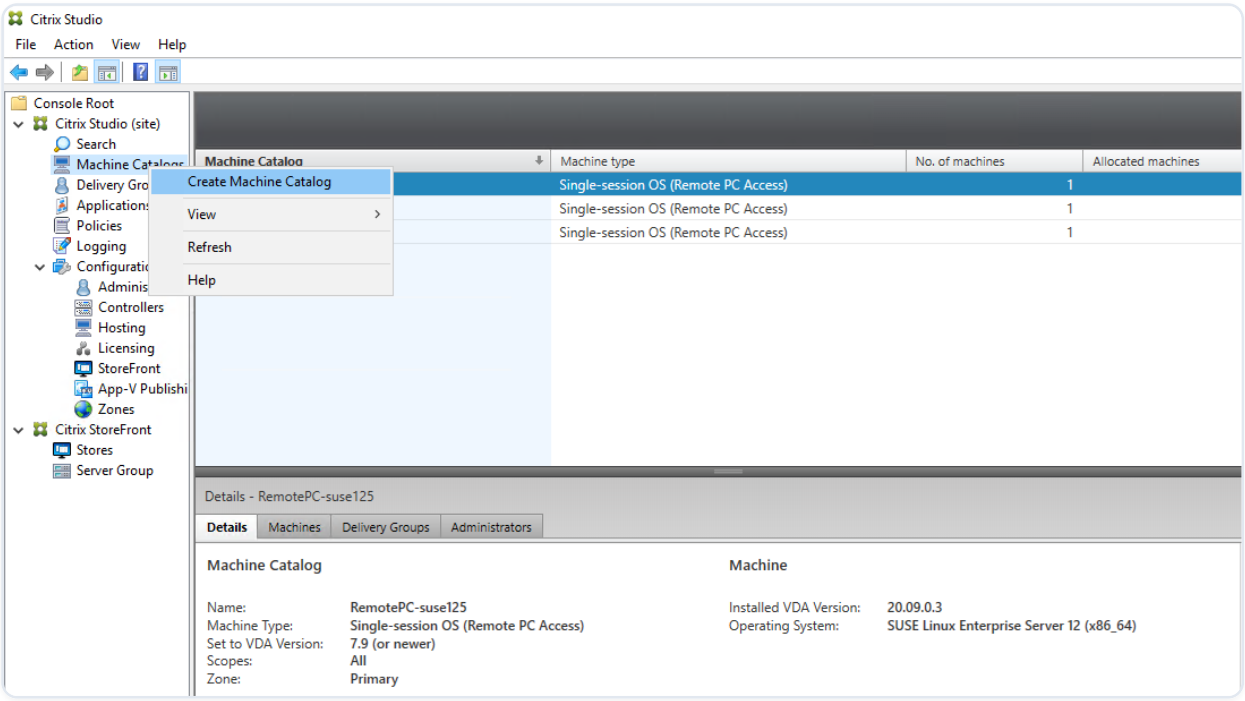
Linux PCセッションを配信するには、ターゲットPCにLinux VDAをインストールし、**Remote PC Access**タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成します。以下のセクションで手順を詳しく説明します。

ステップ1 - ターゲットPCにLinux VDAをインストールする

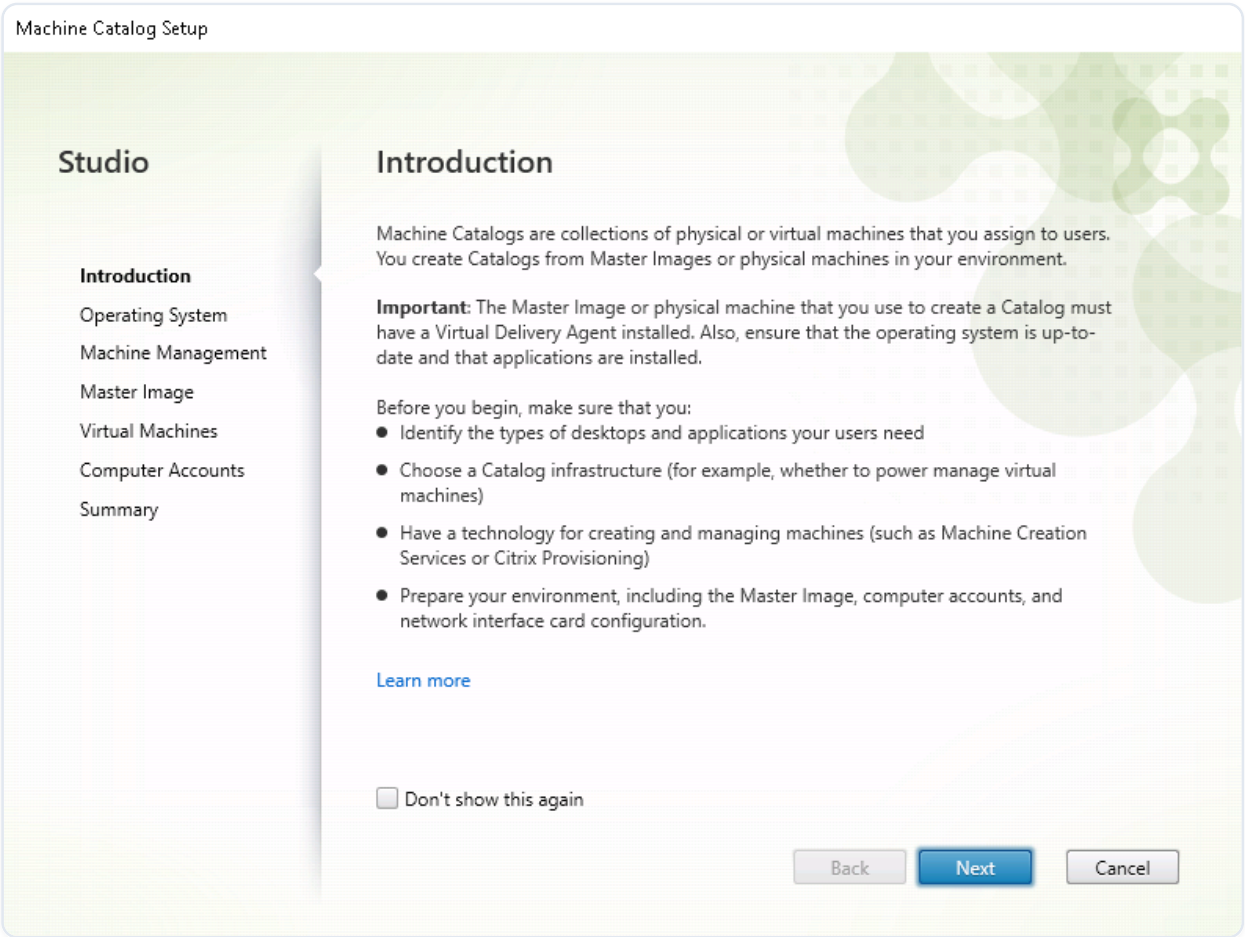
Linux VDAのインストールには、easy installを使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE`変数の値を`Y`に設定します。

ステップ2 - Remote PC Accessタイプのマシンカタログを作成する

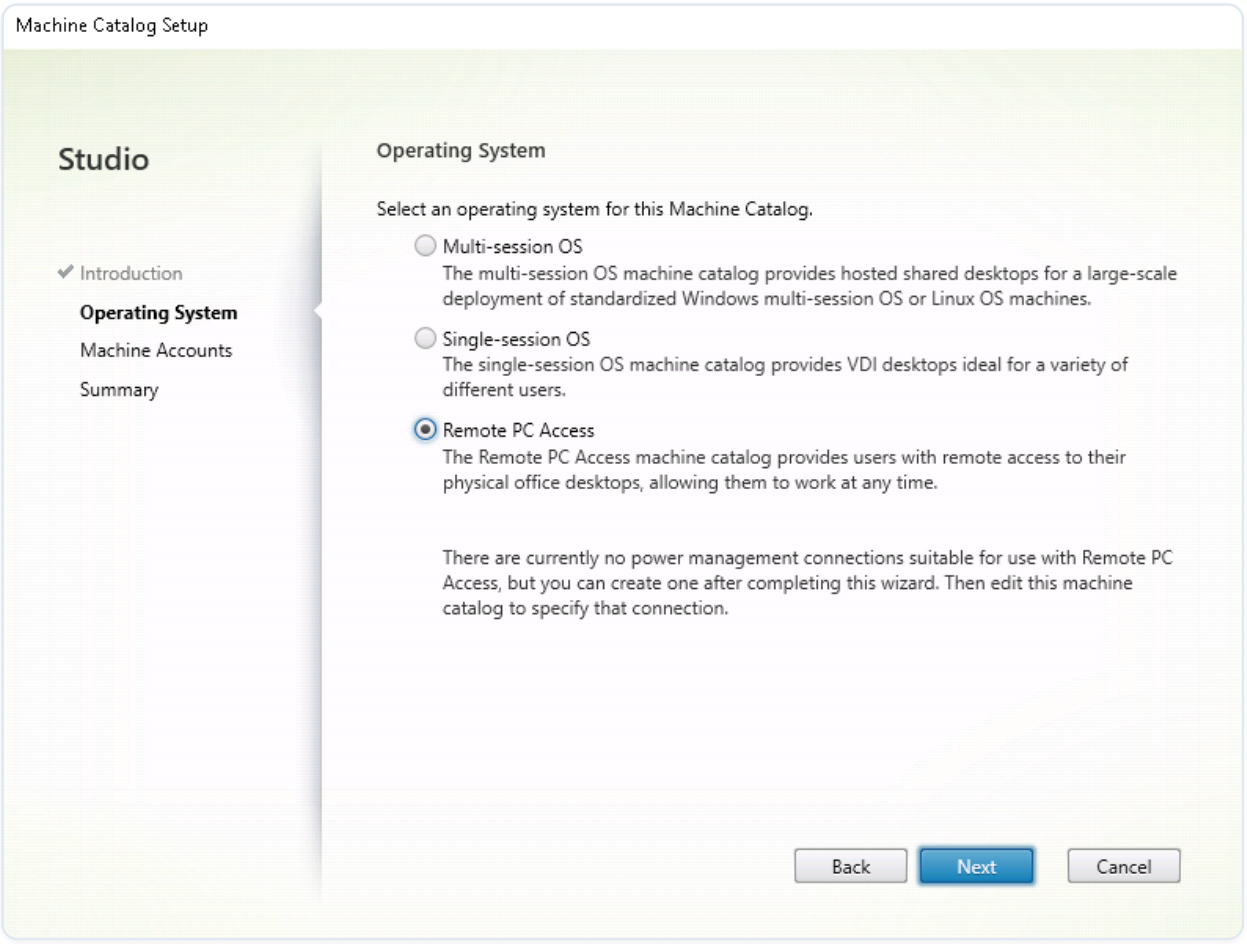
1. Citrix Studioで、**Machine Catalogs**を右クリックし、ショートカットメニューから**Create Machine Catalog**を選択します。



2. **Introduction**ページで**Next**をクリックします。



3. **Operating System**ページで**Remote PC Access**を選択します。



4. ターゲットPCを含むOUを選択するには「**OUの追加**」をクリックするか、マシンカタログに個々のマシンを追加するには「**マシンアカウントの追加**」をクリックします。

   OUまたはマシンアカウントを追加するイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-4-add-ous-or-add-machine-accounts.png)

5. マシンカタログに名前を付けます。

   マシンカタログに名前を付けるイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-5-name-the-machine-catalog.png)

6. (オプション) マシンカタログを右クリックし、関連する操作を実行します。

   マシンカタログでの関連操作のイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-6-edit-a-machine-catalog.png)

ステップ3 - アクセスを要求するユーザーがマシンカタログ内のPCを利用できるように、デリバリーグループを作成します

1. Citrix Studioで、「**デリバリーグループ**」を右クリックし、ショートカットメニューから「**デリバリーグループの作成**」を選択します。

デリバリーグループの作成を選択するイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-11-create-a-delivery-group.png)

2. 「**デリバリーグループの開始**」ページで「**次へ**」をクリックします。

   デリバリーグループの開始ページのイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-22-getting-started-with-delivery-groups.png)

3. ステップ2で作成したマシンカタログを選択し、デリバリーグループに関連付けます。

   マシンカタログをデリバリーグループに関連付けるイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-33-assocatiing-the-machine-catalog-with-the-delivery-group.png)

4. マシンカタログ内のPCにアクセスできるユーザーを追加します。追加したユーザーは、クライアントデバイス上のCitrix Workspaceアプリを使用して、PCにリモートでアクセスできます。

   マシンカタログ内のPCにアクセスできるユーザーを追加するイメージ(/en-us/linux-virtual-delivery-agent/2411/media/remote-pc-access-44-add-uses-to-access-pcs.png)

Wake on LAN (ウェイク・オン・ラン)

Remote PC AccessはWake on LANをサポートしており、ユーザーは物理PCをリモートで起動できます。この機能により、ユーザーは使用していないときにオフィスPCの電源を切ったままにして、エネルギーコストを節約できます。また、誤ってマシンの電源が切れてしまった場合でも、リモートアクセスを可能にします。

Wake on LAN機能を使用すると、デリバリーコントローラーからの指示があった場合、PC上で実行されているVDAから、そのPCが存在するサブネットにマジックパケットが直接送信されます。これにより、追加のインフラストラクチャコンポーネントやマジックパケット配信のためのサードパーティソリューションに依存することなく、この機能が動作します。

Wake on LAN機能は、レガシーなSCCMベースのWake on LAN機能とは異なります。SCCMベースのWake on LANの詳細については、「Wake on LAN – SCCM統合」を参照してください。

システム要件

Wake on LAN機能を使用するためのシステム要件は次のとおりです。

- コントロールプレーン:
  - Citrix DaaS™ （以前の名称：Citrix Virtual Apps and Desktops サービス）
  - Citrix Virtual Apps and Desktops 2012 以降のバージョン
- 物理PC:

- VDAバージョン2012以降
- BIOSおよびNICでWake on LANが有効になっていること

Wake on LANの構成

現在、統合されたWake on LANの構成はPowerShellを使用してのみサポートされています。

Wake on LANを構成するには:

1. まだ Remote PC Access マシンカタログがない場合は、作成します。
2. まだ Wake on LAN ホスト接続がない場合は、作成します。

> **注:**
>
> Wake on LAN機能を使用するには、「Microsoft Configuration Manager Wake on LAN」タイプのホスト接続がある場合、ホスト接続を作成します。

3. Wake on LAN ホスト接続の一意の識別子を取得します。
4. Wake on LAN ホスト接続をマシンカタログに関連付けます。

   Wake on LAN ホスト接続を作成するには：

```
# Load Citrix SnapIns

Add-PSSnapIn -Name "*citrix*"



# Provide the name of the Wake on LAN host connection

[string]$connectionName = "Remote PC Access Wake on LAN"



# Create the hypervisor connection

$hypHc = New-Item -Path xdhyp:\Connections `

             -Name $connectionName `

             -HypervisorAddress "N/A" `
```

```
            -UserName "woluser" `

            -Password "wolpwd" `

            -ConnectionType Custom `

            -PluginId VdaWOLMachineManagerFactory `

            -CustomProperties "<CustomProperties></CustomProperties>" `

            -Persist



$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid



# Wait for the connection to be ready before trying to use it

while (-not $bhc.IsReady)

{

    Start-Sleep -s 5

    $bhc = Get-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid

}
```

ホスト接続の準備ができたら、次のコマンドを実行してホスト接続の一意の識別子を取得します。

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"

$hypUid = $bhc.Uid
```

接続の一意の識別子を取得したら、次のコマンドを実行して、その接続を Remote PC Access マシンカタログに関連付けます。

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog
-RemotePCHypervisorConnectionUid $hypUid
```

5. マシンカタログ内の各VMのBIOSおよびNICでWake on LANを有効にします。

   注： Wake on LANを有効にする方法は、マシンの構成によって異なります。

   - BIOSでWake on LANを有効にするには：

     a. BIOSに入り、Wake on LAN機能を有効にします。

        BIOSにアクセスする方法は、マザーボードの製造元と、製造元が選択したBIOSベンダーによって異なります。

     b. 設定を保存し、マシンを再起動します。

   - NICでWake on LANを有効にするには：

     a. `sudo ethtool <NIC>`コマンドを実行して、NICがマジックパケットをサポートしているかどうかを確認します。

        `<NIC>`はNICのデバイス名です（例：`eth0`）。`sudo ethtool <NIC>`コマンドは、NICの機能に関する出力を提供します。

        - 出力に`Supports Wake-on: <letters>`のような行が含まれており、`<letters>`に文字`g`が含まれている場合、お使いのNICはWake on LANマジックパケット方式をサポートしています。
        - 出力に`Wake-on: <letters>`のような行が含まれており、`<letters>`に文字`g`が含まれていて、文字`d`が含まれていない場合、Wake on LANマジックパケット方式が有効になっています。ただし、`<letters>`に文字`d`が含まれている場合、Wake on LAN機能が無効になっていることを示します。この場合、`sudo ethtool -s <NIC> wol g`コマンドを実行してWake on LANを有効にしてください。

     b. ほとんどのディストリビューションでは、起動ごとに`sudo ethtool -s <NIC> wol g`コマンドが必要です。このオプションを永続的に設定するには、お使いのディストリビューションに基づいて以下の手順を実行してください。

        Ubuntu:

        インターフェース構成ファイル`/etc/network/interfaces`に`up ethtool -s <NIC> wol g`の行を追加します。例:

```
# ifupdown has been replaced by netplan(5) on this system. See

# /etc/netplan for current configuration.

# To re-enable ifupdown on this system, you can run:

# sudo apt install ifupdown

auto eth0

iface eth0 inet static

            address 10.0.0.1

            netmask 255.255.240.0

            gateway 10.0.0.1

            up ethtool -s eth0 wol g
```

RHEL/SUSE:

インターフェース構成ファイル `/etc/sysconfig/network-scripts/ifcfg-<NIC>` に次の `ETHTOOL_OPTS` パラメーターを追加します。

```
ETHTOOL_OPTS="-s ${DEVICE} wol g"
```

設計上の考慮事項

Remote PC Access で Wake on LAN を使用する予定がある場合は、次の点を考慮してください。

- 複数のマシンカタログで同じ Wake on LAN ホスト接続を使用できます。
- ある PC が別の PC を起動するには、両方の PC が同じサブネット内にあり、同じ Wake on LAN ホスト接続を使用している必要があります。PC が同じマシンカタログ内にあるか、異なるマシンカタログ内にあるかは関係ありません。
- ホスト接続は特定のゾーンに割り当てられます。展開に複数のゾーンが含まれている場合は、各ゾーンに Wake on LAN ホスト接続が必要です。これはマシンカタログにも当てはまります。
- マジックパケットはグローバルブロードキャストアドレス255.255.255.255を使用してブロードキャストされます。このアドレスがブロックされていないことを確認してください。

- そのサブネット内のマシンを起動できるようにするには、Wake on LAN接続ごとに、サブネット内に少なくとも1台のPCが電源オンになっている必要があります。

運用上の考慮事項

Wake on LAN機能を使用する際の考慮事項は次のとおりです。

- 統合されたWake on LAN機能を使用してPCを起動できるようにするには、VDAが少なくとも1回登録されている必要があります。
- Wake on LANはPCを起動するためにのみ使用できます。再起動やシャットダウンなどの他の電源操作はサポートしていません。
- Wake on LAN接続が作成されると、Studioに表示されます。ただし、Studio内でそのプロパティを編集することはサポートされていません。
- マジックパケットは次の2つの方法のいずれかで送信されます。
  - ユーザーが自分のPCへのセッションを起動しようとし、VDAが未登録の場合
  - 管理者がStudioまたはPowerShellから手動で電源オンコマンドを送信した場合
- デリバリーコントローラーはPCの電源状態を認識しないため、Studioでは電源状態の下に**サポートされていません**と表示されます。デリバリーコントローラーはVDA登録状態を使用して、PCがオンかオフかを判断します。

その他のリソース

以下は Remote PC Access に関するその他のリソースです。

- ソリューション設計ガイダンス: Remote PC Access Design Decisions。
- Remote PC Access アーキテクチャの例: Citrix Remote PC Access ソリューションのリファレンスアーキテクチャ。

セッション

このセクションには、次のトピックが含まれています。

- アダプティブトランスポート
- 一時ホームディレクトリでのログオン
- アプリケーションの公開
- セッションの信頼性
- ランデブー V1

- ランデブー V2
- TLS を使用したユーザーセッションの保護
- DTLS を使用したユーザーセッションの保護

アダプティブトランスポート

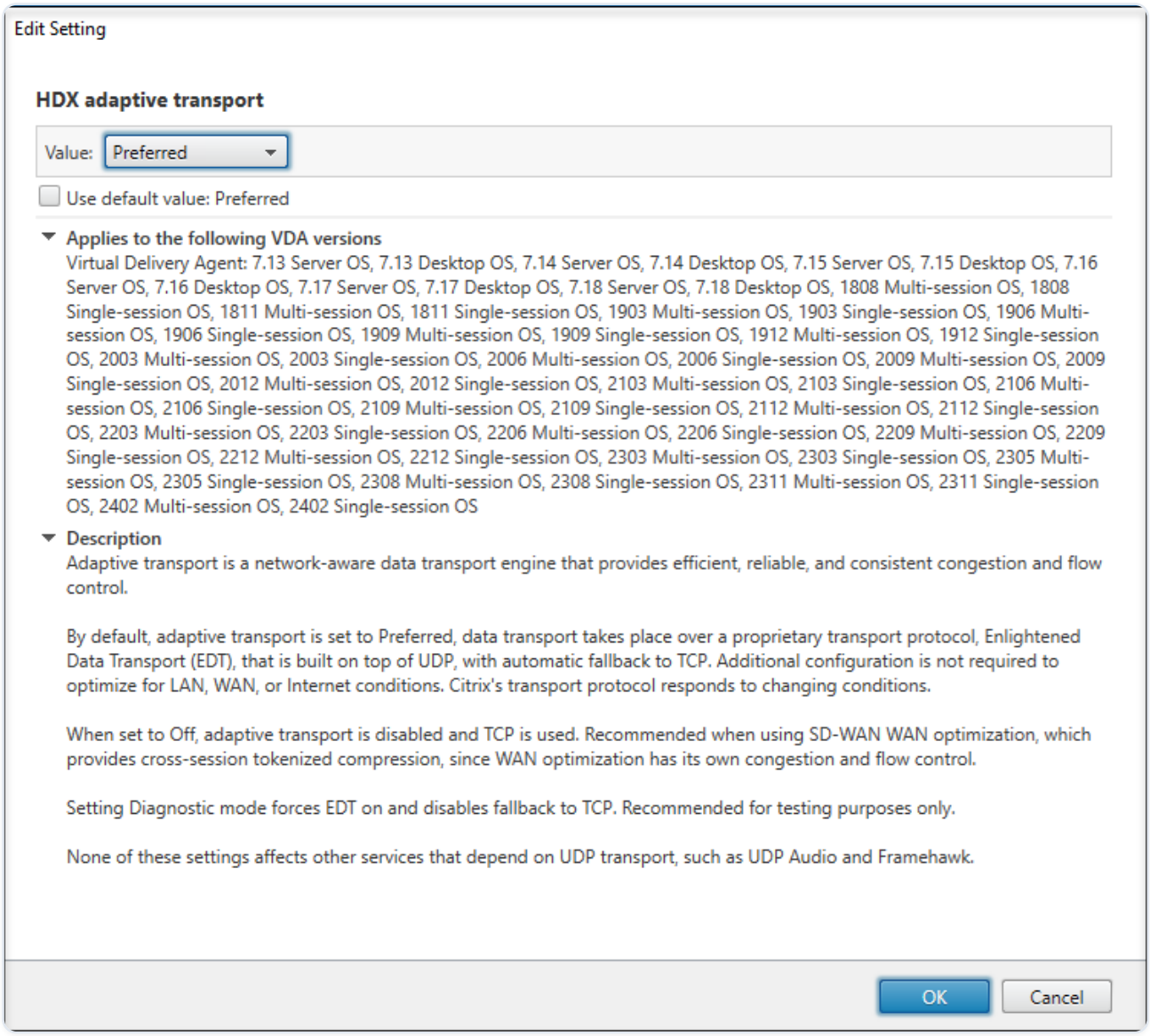
アダプティブトランスポートは、Citrix Virtual Apps and Desktops™のメカニズムであり、ICA接続のトランスポートプロトコルとしてEnlightened Data Transport (EDT)を使用できます。EDTが利用できない場合、アダプティブトランスポートはTCPに切り替わります。

EDTは、User Datagram Protocol (UDP)上に構築されたCitrix独自のトランスポートプロトコルです。サーバーのスケーラビリティを維持しながら、困難な長距離接続において優れたユーザーエクスペリエンスを提供します。EDTは、信頼性の低いネットワーク上のすべてのICA®仮想チャネルのデータスループットを向上させ、より良く、より一貫したユーザーエクスペリエンスを提供します。

詳細については、Citrix Virtual Apps and Desktopsのドキュメントのアダプティブトランスポートを参照してください。

アダプティブトランスポートを有効または無効にする

アダプティブトランスポートはデフォルトで有効になっています。**HDX™ Adaptive Transport**ポリシー設定を使用して、次のオプションを構成できます。



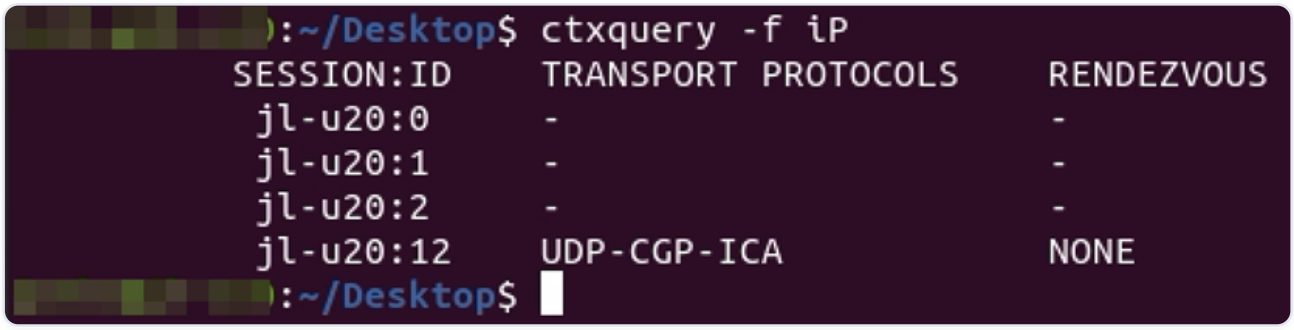
- **優先**: アダプティブトランスポートが有効になり、Enlightened Data Transport (EDT)を優先トランスポートプロトコルとして使用し、TCPへのフォールバックを伴います。
- **診断モード**: アダプティブトランスポートが有効になり、EDTの使用を強制します。TCPへのフォールバックは無効になります。この設定は、テストとトラブルシューティングのみに推奨されます。
- **オフ**: アダプティブトランスポートが無効になり、トランスポートにはTCPのみが使用されます。

アダプティブトランスポートが使用されているかどうかを確認する

現在のセッションでEDTがトランスポートプロトコルとして使用されているかどうかを確認するには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxquery -f iP
```

EDTが使用されている場合、表示されるトランスポートプロトコルにはUDPが含まれます。例:



EDT MTU検出

MTU検出により、EDTはセッション確立時に最大伝送単位 (MTU) を自動的に決定できます。これにより、パフォーマンスの低下やセッション確立の失敗につながる可能性のあるEDTパケットの断片化が防止されます。

システム要件:

- Linux VDA バージョン 2012 以降
- Citrix Workspace™ アプリ:
  - Windows: 1911 以降
- シトリックス ADC:
  - 13.0.52.24 以降
  - 12.1.56.22 以降
- セッションの信頼性を有効にする必要があります

この機能をサポートしないクライアントプラットフォームまたはバージョンを使用している場合は、環境に適したカスタムEDT MTUの構成の詳細について、「CTX231821」を参照してください。

VDAでのEDT MTU検出の制御

EDT MTU検出は、デフォルトでVDAで有効になっています。`MtuDiscovery`レジストリキーを次のように設定することで、有効または無効にできます。

- EDT MTU検出を有効にするには、次のコマンドを使用して`MtuDiscovery`レジストリキーを設定し、VDAを再起動して、VDAが登録されるのを待ちます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t
"REG_DWORD" -v "MtuDiscovery" -d "0x00000001" --force
```

- EDT MTU検出を無効にするには、`MtuDiscovery`レジストリ値を削除します。

> **警告:**
>
> レジストリを誤って編集すると、オペレーティングシステムの再インストールが必要になるような深刻な問題が発生する可能性があります。Citrixは、**レジストリエディター**の誤った使用によって生じる問題が解決できることを保証できません。**レジストリエディター**は自己責任で使用してください。編集する前に必ずレジストリをバックアップしてください。

クライアントでのEDT MTU検出の制御

ICAファイルに`MtuDiscovery`パラメーターを追加することで、クライアントでEDT MTU検出を選択的に制御できます。この機能を無効にするには、`Application`セクションで次のように設定します。

`MtuDiscovery=Off`

この機能を再度有効にするには、ICAファイルから`MtuDiscovery`パラメーターを削除します。

> **重要:**
>
> このICAファイルパラメーターが機能するには、VDAでEDT MTU検出を有効にする必要があります。VDAでEDT MTU検出が有効になっていない場合、ICAファイルパラメーターは効果がありません。

強化されたEDT輻輳制御

EDTプロトコルを最適化するために、輻輳制御アルゴリズムが導入されました。この実装により、EDTはより高いスループットを達成し、遅延を削減することで、強化されたユーザーエクスペリエンスを実現します。

この機能はデフォルトで有効になっています。これを無効にして再度有効にするには、それぞれ以下のコマンドを実行し、その後、`ctxhdx`サービスを再起動します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal
Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d
"0x00000000" --force
```

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal
Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d
"0x00000001" --force
```

HDX™アダプティブスループット

HDXアダプティブスループットは、出力バッファを調整することで、ICA®セッションのピークスループットをインテリジェントに微調整します。出力バッファの数は、初期値として高い値に設定されます。この高い値により、特に高遅延ネットワークにおいて、データがより迅速かつ効率的にクライアントに送信されるようになります。この機能は、ユーザーエクスペリエンスを向上させます。これにより、インタラクティブ性が向上し、ファイル転送が高速化され、ビデオ再生がよりスムーズになり、フレームレートと解像度が高くなります。ICAセッション内のデータストリームがインタラクティブ性に悪影響を与えていないか、セッションのインタラクティブ性が常に測定されます。そのような場合、大規模なデータストリームがセッションに与える影響を軽減し、インタラクティブ性を回復させるために、スループットが低下されます。

重要:

HDXアダプティブスループットは、メカニズムをクライアントからVDAに移動することで、出力バッファの設定方法を変更します。手動での設定は不要です。

この機能には、VDAの最小バージョン2311が必要です。これ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できます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t
"REG_DWORD" -v "AdaptiveScalingEnabled" -d "0x00000001" --force
```

この機能は、有効化後に起動されたセッションにのみ適用されます。

セッションのログオン画面におけるカスタム背景およびバナーメッセージ

セッションのログオン画面にカスタム背景またはバナーメッセージを追加する

> **ヒント:**

> SUSE 15.6でこの機能を使用するには、`<http://download.opensuse.org/distribution/leap/15.3/repo/oss/>`から`imlib2`をインストールします。

> 次のコマンドを使用して、セッションの**ログオン**画面にカスタム背景またはバナーメッセージを追加できます。セッションの**ログオン**画面に背景とバナーメッセージの両方を追加するには、バナーメッセージを背景画像に埋め込むことができます。セッションを開くと、まずバナーメッセージページが表示され、その後、認証ダイアログが表示されます。

カスタムバナーメッセージのタイトルを設定するには、次を実行します:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"LogonDisplayStringTitle" -d "<Banner message title>" --force
```

バナーメッセージのタイトルの最大長は64バイトです。

カスタムバナーメッセージの本文を設定するには、次を実行します:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"LogonDisplayString" -d "Body of custom banner message\nBody of custom
banner message\nBody of custom banner message\n" --force
```

バナーメッセージの本文の最大長は1,024バイトです。

> **ヒント:**

> `\n`要素は改行を作成します。この例では、バナーメッセージ画面は次のようになります:



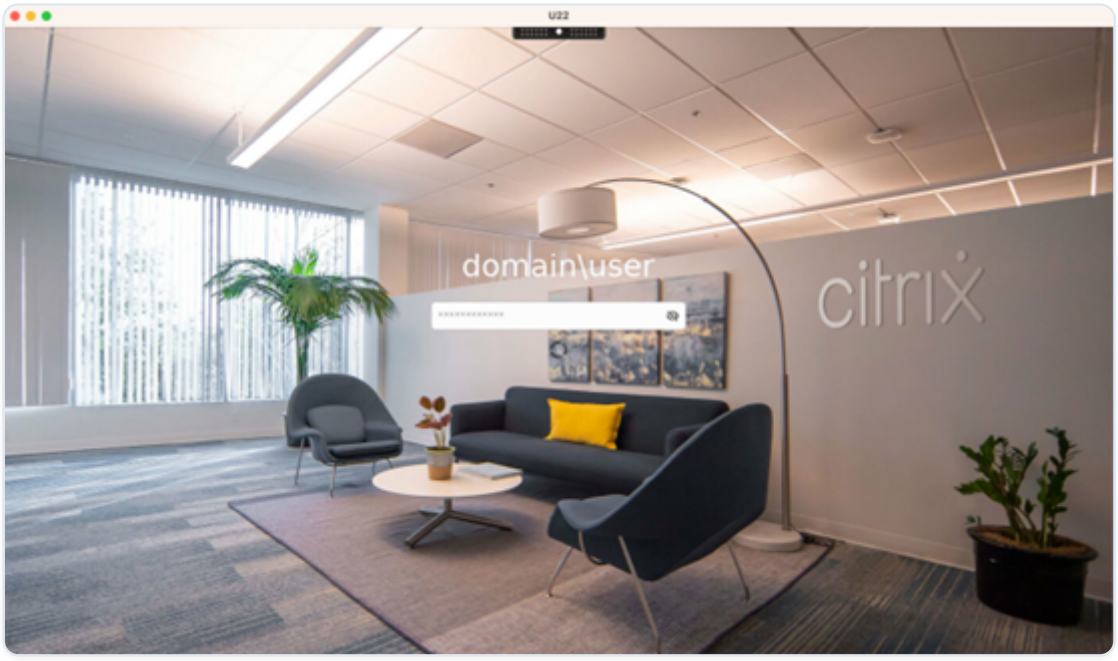
カスタムバナーメッセージが期待どおりに表示されるようにするには、エスケープ文字を使用する必要がある場合があります。たとえば、HTMLで**&**を表すには**&amp\;**を使用し、**Bash**コマンドで**\!**文字をエスケープするには、その前にバックスラッシュ **(****) を付けます。

セッションの**ログオン**画面にカスタム背景を追加するには、次を実行します:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v
"BackgroundImagePath" -d "<path to the background image>" --force
```

カスタム背景を表示するには、セッションユーザーが背景画像のパスにアクセスできる必要があります。

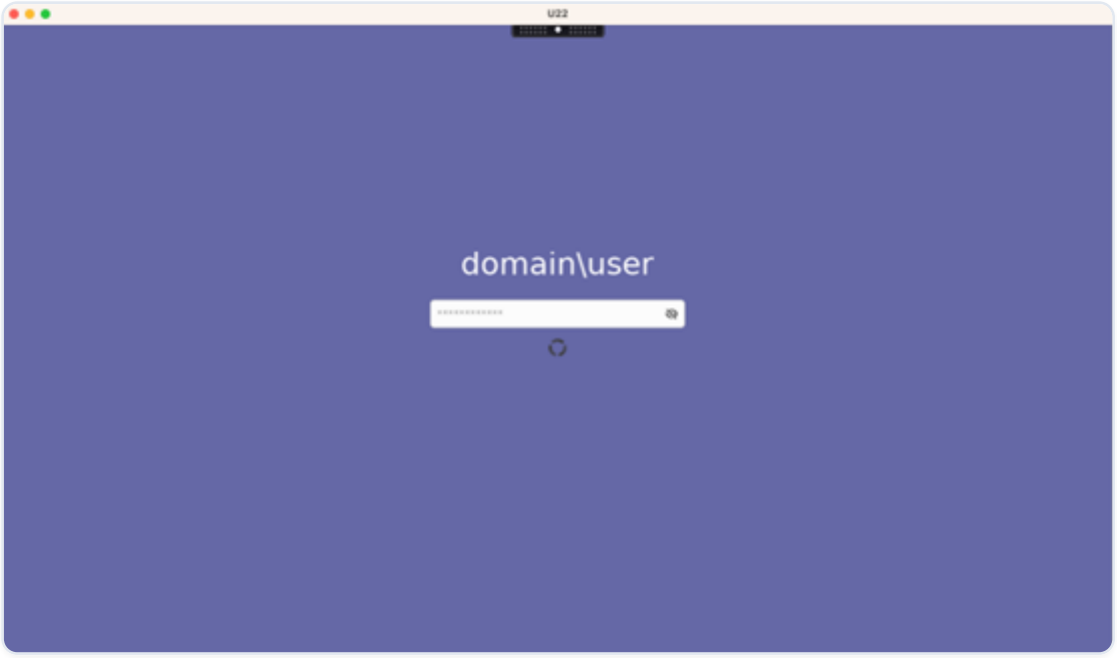
例:



セッションのログオン画面の例

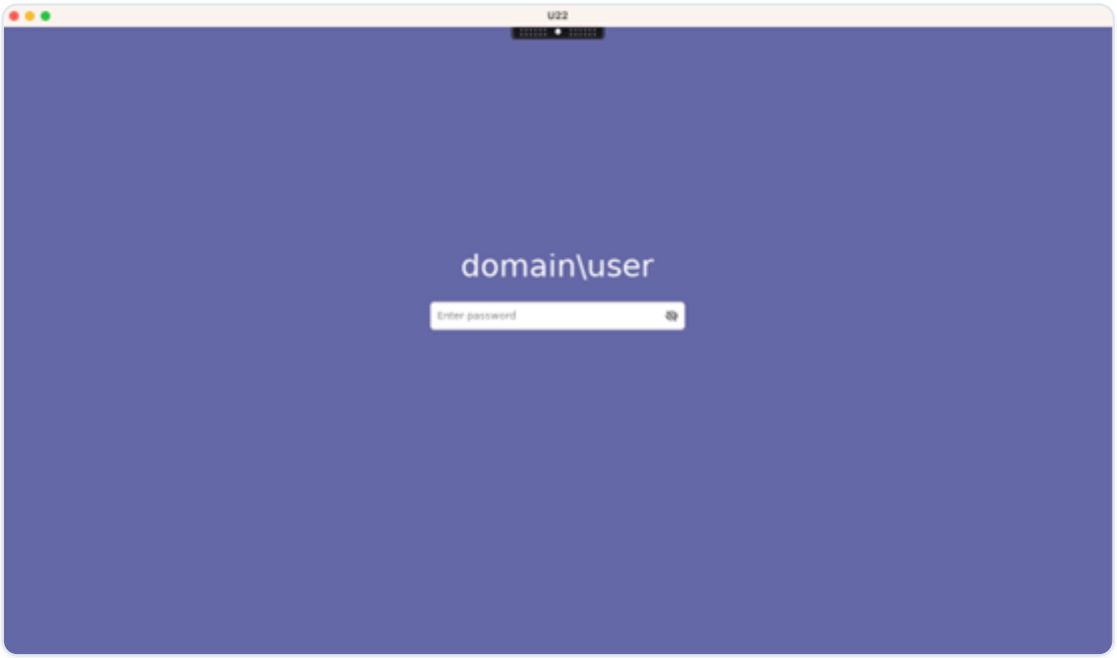
さまざまなシナリオにおけるセッションログオン画面の例を次に示します。

- 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**:

  ログオンプロセスが表示されます。

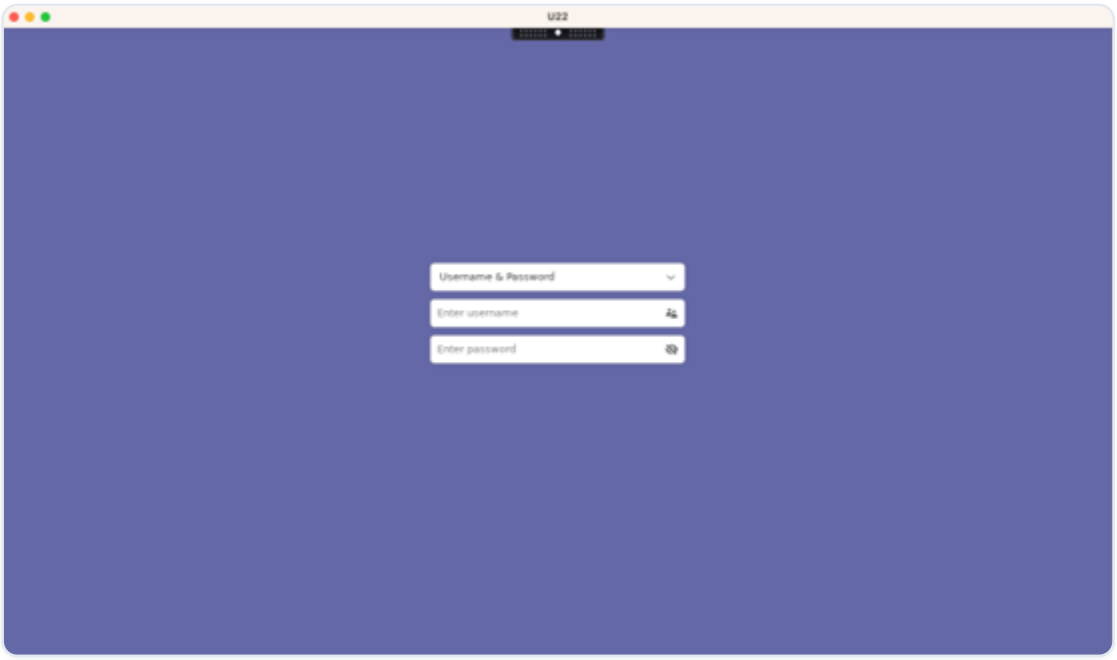
- 一般的な非SSOシナリオでのセッション**ログオン**:



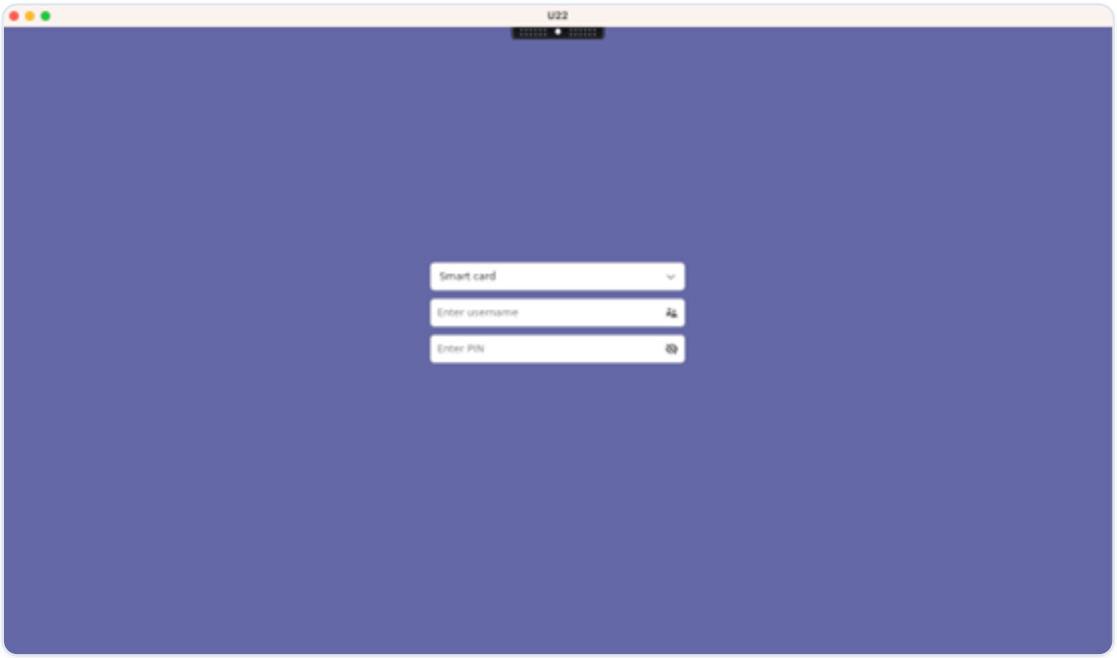
  - 非SSOシナリオでは、パスワードまたはPINコードが必要です。
  - ユーザーはパスワードとPINコードの表示を切り替えることができ、誤った入力を簡単に特定できます。

- ユーザーがCitrix Workspace™アプリへのログオンに使用した資格情報とは異なる資格情報でVDAセッションにログオンする場合の、非SSOシナリオでのセッションログオン:

  セッションログオンに使用されるユーザー名とパスワード:



  セッションログオンに使用されるスマートカード:



非SSOシナリオでサポートされているユーザー認証方法の組み合わせについては、「非SSO認証」を参照してください。

セッションユーザーによるカスタムデスクトップ環境

CTX_XDL_DESKTOP_ENVIRONMENT変数を使用して、セッションユーザーのデスクトップ環境を指定できます。2209リリース以降、セッションユーザーは独自のデスクトップ環境をカスタマイズできます。セッションユーザーがこの機能を使用できるようにするには、VDAにデスクトップ環境を事前にインストールする必要があります。

次の表は、セッションユーザーによるカスタムデスクトップ環境をサポートするLinuxディストリビューションとデスクトップ環境の対応表を示しています。

LINUXディストリビューション	サポートされているデスクトップ
デビアン 12.7/12.5	メイト、グノーム、グノームクラシック、ケーディーイー、エックスエフシーイー
デビアン 11.11	メイト、グノーム、グノームクラシック、ケーディーイー
RHEL 9.4/9.2	グノーム、エックスエフシーイー
RHEL 8.10/8.8	メイト、グノーム、グノームクラシック、エックスエフシーイー
ロッキーリナックス 9.4/9.2	グノーム、エックスエフシーイー

LINUXディストリビューション	サポートされているデスクトップ
ロッキーリナックス 8.10/8.8	メイト、グノーム、グノームクラシック、ケーディーイー、エックスエフシーイー
SUSE 15.6	メイト、グノーム、グノームクラシック
ウブントゥ 22.04/20.04	メイト、グノーム、グノームクラシック、ケーディーイー、エックスエフシーイー
ウブントゥ 24.04	メイト、ケーディーイー、エックスエフシーイー

デスクトップ切り替えコマンド

> **注:**
>
> ターミナルとシステムトレイの両方からデスクトップ環境を切り替えることができます。

ターミナルから目的のデスクトップ環境に切り替えるには、セッション内で対応するコマンドを実行します。

目的のデスクトップ環境が次の場合は、	次のコマンドを実行します。
GNOME	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME`
グノームクラシック	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh GNOME-CLASSIC`
MATE	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh MATE`
KDE	`/opt/Citrix/VDA/bin/ctxdesktopswitch.sh KDE`

KDE のヒント

- KDE の起動時に Magnus がロードされることがあります。回避策として、`sudo apt remove magnus` を実行して Magnus パッケージを削除できます。

- KDE の起動中に発生する QT 警告を無効にするには、ルートユーザーとして `/usr/share/qt5/qtlogging.ini` を編集し、次のエントリを追加します。

```
qt.qpa.xcb.xcberror.error=false

qt.qpa.xcb.warning=false

qt.qpa.xcb.error=false
```

- KDE で画面のロック解除に失敗する場合があります。回避策として、デスクトップの自動ロック機能を無効にすることをお勧めします。

一時ホームディレクトリでのログオン

Linux VDA上のマウントポイントが失敗する場合に備えて、一時ホームディレクトリを指定できます。一時ホームディレクトリが指定されている場合、マウントポイントが失敗すると、セッションログオン中にプロンプトが表示されます。ユーザーデータは一時ホームディレクトリに保存されます。

次の表は、ホームディレクトリ設定に役立つレジストリキーについて説明しています。

レジストリキー	説明	コマンド
`LogNoHome`	ユーザーがホームディレクトリなしでセッションにログオンできるかどうかを制御します。デフォルト値は1で、「はい」を意味します。値が0に設定されている場合、ホームディレクトリなしでのセッションログオンは無効になります。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "LogNoHome" -d "0x00000001" --force`
`HomeMountPoint`	Linux VDA上にローカルマウントポイントを設定します。たとえば、`/mnt/home`がマウントポイントの場合、ユーザーのホームディレクトリは`/mnt/home/domain/<user_name>`です。マウントポイントが環境内のユーザーホームディレクトリと同	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "HomeMountPoint" -d "<A directory where the NFS share is to be mounted>" --force`

レジストリキー	説明	コマンド
	じであることを確認してください。この設定は、`CheckUserHomeMountPoint`が0に設定されている場合にのみ有効になります。	
`CheckUserHomeMountPoint`	Linux VDA上のマウントポイントとしてユーザー固有のホームディレクトリを設定するかどうかを制御します。ユーザー固有のホームディレクトリをマウントポイントとして設定する場合は、値を**1**に設定します。デフォルト値は**0**です。	`ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "CheckUserHomeMountPoint" -d "0x00000001" --force`
`TempHomeDirectoryPath`	マウントポイントが失敗した場合に備えて、Linux VDA上に一時ホームディレクトリを設定します。デフォルト値は`/tmp`です。一時ホームディレクトリの設定は、`HomeMountPoint`と`CheckUserHomeMountPoint`によって決定されるマウントポイントが利用できない場合にのみ有効になります。ユーザーの一時ホームディレクトリは`/tmp/CTXSmf_user_id`です。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "TempHomeDirectoryPath" -d "</tmp by default>" --force`
`CheckMountPointRetryTime`	マウントが成功したかどうかを、1秒に1回の頻度でチェックする回数を設定します。デフォルト値は5です。	`ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v "CheckMountPointRetryTime" -d "0x000000010" --force`
`RemoveHomeOnLogoff`	ユーザーログオフ時に一時ホームディレクトリを削除す	`create -k "HKLM\System\CurrentCont`

レジストリキー	説明	コマンド
	るかどうかを制御します。1は「はい」を意味します。0は「いいえ」を意味します。	`rolSet\Control\Citrix" -t "REG_DWORD" -v "RemoveHomeOnLogoff" -d "0x00000000" --force`

アプリケーションの公開

Linux VDAバージョン7.13で、Citrixはサポートされているすべてのシームレスアプリケーション機能を追加しました。この機能を使用するために、特定のインストール手順は必要ありません。

ヒント:

Linux VDAバージョン1.4で、Citrixは非シームレス公開アプリケーションとセッション共有のサポートを追加しました。

Citrix Studioを使用したアプリケーションの公開

デリバリーグループを作成するとき、または既存のデリバリーグループにアプリケーションを追加するときに、Linux VDAにインストールされているアプリケーションを公開できます。このプロセスは、Windows VDAにインストールされているアプリケーションを公開するのと同様です。詳しくは、Citrix Virtual Apps and Desktopsのドキュメント（使用しているCitrix Virtual Apps and Desktopsのバージョンに基づく）を参照してください。

注:

- デリバリーグループを構成するときは、デリバリーの種類が**デスクトップとアプリケーション**または**アプリケーション**に設定されていることを確認してください。
- アプリケーションとデスクトップのデリバリーには、個別のVDAとデリバリーグループを作成することをお勧めします。
- シームレスアプリケーションを使用するには、StoreFrontでシームレスモードを無効にしないでください。シームレスモードはデフォルトで有効になっています。「TWIMode=Off」を設定してすでに無効にしている場合は、「TWIMode=On」に変更す

> るのではなく、この設定を削除してください。そうしないと、公開されたデスクトップを起動できない可能性があります。

制限事項

Linux VDAは、単一ユーザーによる同じアプリケーションの複数の同時インスタンスの起動をサポートしていません。

アプリセッションでは、アプリケーションに固有のショートカットのみが期待どおりに機能します。

既知の問題

アプリケーションの公開中に、以下の既知の問題が確認されています。

- 非矩形ウィンドウはサポートされていません。ウィンドウの角にサーバー側の背景が表示される場合があります。
- 公開されたアプリケーションのウィンドウコンテンツのプレビューはサポートされていません。
- 複数のLibreOfficeアプリケーションを実行すると、これらのアプリケーションはプロセスを共有するため、最初に起動されたものだけがCitrix Studioに表示されます。
- 「Dolphin」のような公開されたQt5ベースのアプリケーションでは、アイコンが表示されない場合があります。この問題を解決するには、`<https://wiki.archlinux.org/title/Qt>`の記事を参照してください。
- Linuxアプリケーションには、使用中のアプリケーションに関する情報を含む「バージョン情報」ウィンドウがよくあり、詳細情報へのWebリンクがそれらのウィンドウによく見られます。「バージョン情報」ウィンドウのWebリンクをクリックすると、**calc**、**gedit**、**calendar**、**LibreOffice Suite**などの公開アプリケーション内からWebブラウザが起動する可能性があります。意図しないWebブラウザの起動は、アプリケーションの分離を迂回し、セキュリティを侵害する可能性があります。この問題を解決するには、以下の手順を実行してデフォルトのWebブラウザを変更します。

  a. カスタムの場所にnone.shファイルを作成します。例:

```
sudo mkdir /home/none



sudo touch /home/none/none.sh
```

```
sudo chmod +x /home/none/none.sh
```

b. none.shファイルに次の行を追加します。

```
#!/bin/bash

echo "NONE"
```

c. sudoを使用して/etc/xdg/mimeapps.listファイルを作成し、その後、mimeapps.listファイルに次の行を追加します。

```
[Default Applications]

text/html=none.desktop

x-scheme-handler/http=none.desktop

x-scheme-handler/https=none.desktop

x-scheme-handler/about=none.desktop

x-scheme-handler/unknown=none.desktop
```

d. sudoを使用して/usr/share/applications/none.desktopファイルを作成し、その後、none.desktopファイルに次の行を追加します。

```
[Desktop Entry]


Encoding=UTF-8


Version=1.0


Type=Application


Terminal=false


Exec=/home/none/none.sh


Name=None


Icon=/home/none/none.sh
```

none.shファイルをカスタムの場所に配置できるため、none.desktopがnone.shファイルを正しく参照できることを確認してください。

ランデブー V1

Citrix Gatewayサービスを使用する場合、Rendezvousプロトコルにより、トラフィックはCitrix Cloud™ Connectorsをバイパスし、Citrix Cloudコントロールプレーンに直接かつ安全に接続できます。

考慮すべきトラフィックには2つの種類があります。1) VDA登録およびセッションブローカリングのための制御トラフィック、2) HDX™セッションのトラフィックです。

Rendezvous V1では、HDXセッションのトラフィックがCloud Connectorをバイパスできますが、VDA登録およびセッションブローカリングのためのすべての制御トラフィックをCloud Connectorがプロキシする必要があります。

要件

- Citrix Workspace™およびCitrix Gatewayサービスを使用した環境へのアクセス。
- コントロールプレーン：Citrix DaaS（旧Citrix Virtual Apps and Desktops™サービス）。
- Linux VDAバージョン2112以降。
  - バージョン2112は、非透過型HTTPプロキシに必要な最小バージョンです。
  - バージョン2204は、透過型およびSOCKS5プロキシに必要な最小バージョンです。
- CitrixポリシーでRendezvousプロトコルを有効にします。詳細については、「Rendezvousプロトコルポリシー設定」を参照してください。
- VDAは、すべてのサブドメインを含む`https://*.nssvc.net`にアクセスできる必要があります。その方法ですべてのサブドメインを許可リストに追加できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳細については、Citrix Cloudドキュメントの「インターネット接続要件」セクション（Virtual Apps and Desktopサービスの下）およびKnowledge Centerの記事「CTX270584」を参照してください。
- Cloud Connectorは、セッションをブローカリングする際にVDAのFQDNを取得する必要があります。このタスクは、次の2つの方法のいずれかで実行します。
  - **サイトのDNS解決を有効にする。** **Full Configuration > Settings**に移動し、**Enable DNS resolution**設定をオンにします。または、Citrix Virtual Apps and Desktops Remote PowerShell SDKを使用してコマンド`Set-BrokerSite -DnsResolutionEnabled $true`を実行します。Citrix Virtual Apps and Desktops Remote PowerShell SDKの詳細については、「SDKとAPI」を参照してください。
  - **VDAのPTRレコードを含むDNS逆引き参照ゾーン。** このオプションを選択する場合は、VDAが常にPTRレコードの登録を試みるように構成することをお勧めします。これを行うには、グループポリシーエディターまたはグループポリシーオブジェクトを使用し、**Computer Configuration > Administrative Templates > Network > DNS Client**に移動して、**Register PTR Records**を**Enabled and Register**に設定します。接続のDNSサフィックスがドメインのDNSサフィックスと一致しない場合は、マシンがPTRレコードを正常に登録できるように、**Connection-specific DNS suffix**設定も構成する必要があります。

> **注:**

> DNS解決オプションを使用する場合、Cloud ConnectorはVDAマシンの完全修飾ドメイン名（FQDN）を解決できる必要があります。内部ユーザーがVDAマシンに直接接続する場合、クライアントデバイスもVDAマシンのFQDNを解決できる必要があります。

> DNS逆引き参照ゾーンを使用する場合、PTRレコード内のFQDNはVDAマシンのFQDNと一致する必要があります。PTRレコードに異なるFQDNが含まれている場合、Rendezvous接続は失敗します。たとえば、マシンのFQDNが`vda01.domain.net`の場合、PTRレコードには`vda01.domain.net`が含まれている必要があります。`vda01.sub.domain.net`のような異なるFQDNは機能しません。

プロキシ構成

VDAは、HTTPおよびSOCKS5プロキシを介したRendezvous接続の確立をサポートしています。

プロキシに関する考慮事項

Rendezvousでプロキシを使用する際は、次の点を考慮してください。

- 非透過型HTTPプロキシとSOCKS5プロキシがサポートされています。
- パケットの復号化と検査はサポートされていません。VDAとGateway Service間のICA®トラフィックが傍受、復号化、または検査されないように例外を構成してください。そうしないと、接続が切断されます。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、**Negotiate authentication**スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

> **注:**
>
> Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名（SPN）を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAは、セッションを確立する際に`HTTP/<proxyURL>`の形式でSPNを生成します。このとき、プロキシURLは**Rendezvous proxy**ポリシー設定から取得されます。SPNを作成しない場合、認証は失敗します。

- SOCKS5プロキシでの認証は現在サポートされていません。SOCKS5プロキシを使用する場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように例外を構成する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

透過型プロキシ

Rendezvousでは透過型HTTPプロキシがサポートされています。ネットワークで透過型プロキシを使用している場合、VDAに追加の構成は必要ありません。

非透過型プロキシ

ネットワークで非透過型プロキシを使用する場合は、Rendezvous proxy configuration設定を構成します。この設定が有効になっている場合、VDAが使用するプロキシを認識できるように、HTTPまたはSOCKS5プロキシアドレスを指定します。例：

- プロキシアドレス: `http://<URL or IP>:<port>` または `socks5://<URL or IP>:<port>`

Rendezvousの検証

すべての要件を満たしている場合は、Rendezvousが使用されているかどうかを検証するために、次の手順に従ってください。

1. VDA上でターミナルを起動します。
2. `/opt/Citrix/VDA/bin/ctxquery -f iP` を実行します。
3. トランスポートプロトコルは接続の種類を示します:
    - TCP ランデブー: TCP - TLS - CGP - ICA
    - EDT ランデブー: UDP - DTLS - CGP - ICA
    - Cloud Connector経由のプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

> **ヒント:**
>
> Rendezvousが有効な状態でVDAがCitrix Gatewayサービスに直接到達できない場合、VDAはCloud Connectorを介してHDXセッションをプロキシするようフォールバックします。

Rendezvousの仕組み

この図は、Rendezvous接続フローの概要です。

Rendezvousプロトコルの概要(/en-us/linux-virtual-delivery-agent/2411/media/rendezvous-overview.png)

以下の手順に従ってフローを理解してください。

1. Citrix Workspaceに移動します。
2. Citrix Workspaceで資格情報を入力します。
3. オンプレミスのActive Directoryを使用している場合、Citrix DaaS™はCloud Connectorチャネルを介してActive Directoryを使用して資格情報を認証します。
4. Citrix WorkspaceはCitrix DaaSから列挙されたリソースを表示します。
5. Citrix Workspaceからリソースを選択します。Citrix DaaSは、受信セッションの準備をするようVDAにメッセージを送信します。
6. Citrix Workspaceは、Citrix Cloudによって生成されたSTAチケットを含むICAファイルをエンドポイントに送信します。
7. エンドポイントはCitrix Gatewayサービスに接続し、VDAに接続するためのチケットを提供し、Citrix Cloudがそのチケットを検証します。
8. Citrix Gatewayサービスは、接続情報をCloud Connectorに送信します。Cloud Connectorは、その接続がRendezvous接続であるかどうかを判断し、その情報をVDAに送信します。
9. VDAはCitrix Gatewayサービスへの直接接続を確立します。
10. VDAとCitrix Gatewayサービス間の直接接続が不可能な場合、VDAはCloud Connectorを介して接続をプロキシします。
11. Citrix Gatewayサービスは、エンドポイントデバイスとVDA間の接続を確立します。
12. VDAは、Cloud Connectorを介してCitrix DaaSとライセンスを検証します。
13. Citrix DaaSは、Cloud Connectorを介してセッションポリシーをVDAに送信します。これらのポリシーが適用されます。

ランデブー V2

Citrix Gatewayサービスを使用する場合、Rendezvousプロトコルにより、トラフィックはCitrix Cloud™ Connectorをバイパスし、Citrix Cloudコントロールプレーンに直接かつ安全に接続できます。

考慮すべきトラフィックは2種類あります。1) VDA登録とセッションブローカリングのための制御トラフィック、2) HDX™セッショントラフィックです。

Rendezvous V1では、HDXセッショントラフィックがCloud Connectorをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックをCloud Connectorがプロキシする必要があります。

標準のADドメイン参加済みマシンとドメイン非参加マシンは、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を使用するためにサポートされています。ドメイン非参加マシンでは、Rendezvous V2によりHDXトラフィックと制御トラフィックの両方がCloud Connectorをバイパスできます。

要件

Rendezvous V2を使用するための要件は次のとおりです。

- Citrix Workspace™およびCitrix Gatewayサービスを使用した環境へのアクセス。
- コントロールプレーン: Citrix DaaS（旧Citrix Virtual Apps and Desktops™サービス）。
- VDAバージョン2201以降。
    - HTTPおよびSOCKS5プロキシには、バージョン2204が最低限必要です。
- CitrixポリシーでRendezvousプロトコルを有効にします。詳しくは、「Rendezvousプロトコルポリシー設定」を参照してください。
- VDAは以下にアクセスできる必要があります。
    - `TCP 443` の `https://*.xendesktop.net`。その方法ですべてのサブドメインを許可できない場合は、`https://<customer_ID>.xendesktop.net` を使用できます。ここで、`<customer_ID>` はCitrix Cloud管理者ポータルに表示されているCitrix Cloudの顧客IDです。
    - `https://*.nssvc.net`（すべてのサブドメインを含む）。その方法ですべてのサブドメインをホワイトリストに登録できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳しくは、Citrix Cloudドキュメントのインターネット接続要件セクション（Virtual Apps and Desktopサービスの下）およびKnowledge Centerの記事CTX270584を参照してください。
- VDAは、前述のアドレスに接続できる必要があります。
    - TCP 443で、TCP Rendezvous用。
    - UDP 443で、EDT Rendezvous用。

プロキシ構成

Rendezvousを使用する場合、VDAは制御トラフィックとHDXセッションの両方のトラフィックに対してプロキシ経由での接続をサポートします。両方のトラフィックタイプで要件と考慮事項が異なるため、注意深く確認してください。

制御トラフィックのプロキシに関する考慮事項

- HTTPプロキシのみがサポートされています。
- パケットの復号化と検査はサポートされていません。VDAとCitrix Cloudコントロールプレーン間の制御トラフィックが傍受、復号化、または検査されないように例外を設定してください。そうしないと、接続が失敗します。
- プロキシ認証はサポートされていません。
- 制御トラフィックのプロキシを構成するには、次のようにレジストリを編集します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ProxySettings" -d "http://<URL or IP>:<port>" --force
```

HDXトラフィックのプロキシに関する考慮事項

- HTTPおよびSOCKS5プロキシがサポートされています。
- EDTはSOCKS5プロキシでのみ使用できます。
- HDXトラフィックのプロキシを構成するには、Rendezvous proxy configurationポリシー設定を使用します。
- パケットの復号化と検査はサポートされていません。VDAとCitrix Cloudコントロールプレーン間のHDXトラフィックが傍受、復号化、または検査されないように例外を設定してください。そうしないと、接続が失敗します。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、**Negotiate authentication scheme**がKerberosプロトコルを自動的に選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

> **注:**
>
> Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名 (SPN) を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAは、セッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。このとき、プ

> ロキシURLは**Rendezvous proxy**ポリシー設定から取得されます。SPNを作成しない場合、認証は失敗します。

- SOCKS5プロキシを使用した認証は現在サポートされていません。SOCKS5プロキシを使用している場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように、例外を構成する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

トランスペアレントプロキシ

Rendezvousでは、トランスペアレントHTTPプロキシがサポートされています。ネットワークでトランスペアレントプロキシを使用している場合、VDAに追加の構成は必要ありません。

Rendezvous V2の構成方法

環境でRendezvousを構成する手順は次のとおりです。

1. すべての要件が満たされていることを確認してください。
2. VDAのインストール後、必要なレジストリキーを設定するために次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"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**Allowed**に設定します。Rendezvousプロトコルはデフォルトで無効になっています。Rendezvousプロトコルが有効（**Allowed**）になっている場合、V1ではなくRendezvous V2が有効になります。
   - Citrixポリシーフィルターが適切に設定されていることを確認します。このポリシーは、Rendezvousを有効にする必要があるマシンに適用されます。
   - Citrixポリシーが他のポリシーを上書きしないように、正しい優先順位が設定されていることを確認します。

Rendezvousの検証

セッションがRendezvousプロトコルを使用しているかどうかを確認するには、ターミナルで`/opt/Citrix/VDA/bin/ctxquery -f iP`コマンドを実行します。

表示されるトランスポートプロトコルは、接続の種類を示します。

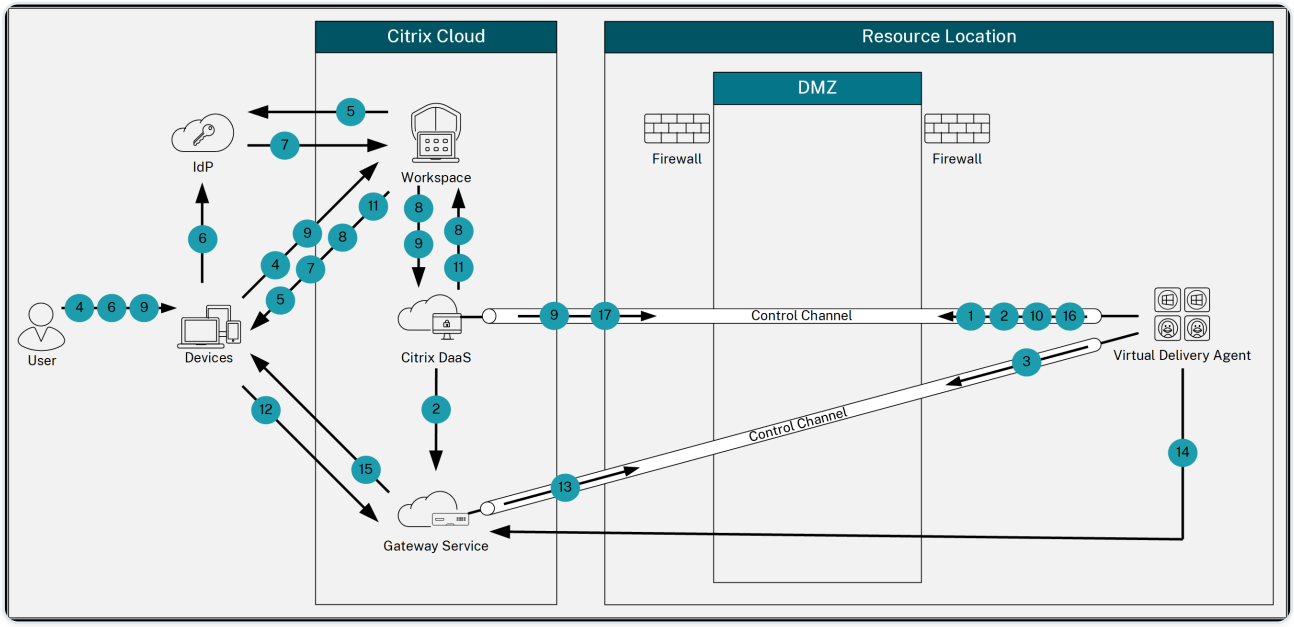
- TCPランデブー: TCP - TLS - CGP - ICA
- EDTランデブー: UDP - DTLS - CGP - ICA
- Cloud Connector経由のプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

Rendezvous V2が使用されている場合、プロトコルバージョンは2.0と表示されます。

> **ヒント:**
>
> Rendezvousが有効になっている状態でVDAがCitrix Gatewayサービスに直接到達できない場合、VDAはHDXセッションをCloud Connector経由でプロキシするようにフォールバックします。

Rendezvousトラフィックフロー

次の図は、Rendezvousトラフィックフローに関する一連の手順を示しています。



1. VDAはCitrix CloudとのWebSocket接続を確立し、登録します。
2. VDAはCitrix Gateway Serviceに登録し、専用のトークンを取得します。
3. VDAはGateway Serviceとの永続的な制御接続を確立します。
4. ユーザーはCitrix Workspaceに移動します。
5. Workspaceは認証構成を評価し、認証のためにユーザーを適切なIdPにリダイレクトします。

6. ユーザーは資格情報を入力します。
7. ユーザー資格情報の検証に成功すると、ユーザーはWorkspaceにリダイレクトされます。
8. Workspaceはユーザーのリソースをカウントし、それらを表示します。
9. ユーザーはWorkspaceからデスクトップまたはアプリケーションを選択します。WorkspaceはCitrix DaaS™にリクエストを送信し、Citrix DaaS™は接続を仲介し、VDAにセッションの準備をするよう指示します。
10. VDAはRendezvous機能とそのIDで応答します。
11. Citrix DaaSは起動チケットを生成し、Workspaceを介してユーザーデバイスに送信します。
12. ユーザーのエンドポイントはGateway Serviceに接続し、起動チケットを提供して、接続するリソースを認証および識別します。
13. Gateway Serviceは接続情報をVDAに送信します。
14. VDAはGateway Serviceとのセッションのために直接接続を確立します。
15. Gateway ServiceはエンドポイントとVDA間の接続を完了します。
16. VDAはセッションのライセンスを検証します。
17. Citrix DaaSは適用可能なポリシーをVDAに送信します。

セキュアHDX™ (プレビュー)

Citrix Workspace™アプリ (クライアント) とVDA (セッションホスト) の間で、ICA®セッションをエンドツーエンドで暗号化できます。

エンドツーエンド暗号化 (E2EE) 機能により、Citrix Gatewayを含む中間ネットワーク要素がICAトラフィックを復号化することはできません。これにより、環境のセキュリティ体制が向上し、設定と管理が簡単になります。

システム要件

- Linux VDAの最小バージョン2311
- Delivery Controller™の最小バージョン2308
- StoreFront™の最小バージョン2308
- Windows向けCitrix Workspaceアプリの最小バージョン2308

構成

エンドツーエンド暗号化を有効にする

エンドツーエンド暗号化 (E2EE) 機能は、デフォルトで無効になっています。これを有効にするには、Citrix Studioで**Secure HDX**ポリシーを**有効**に設定します。

証明書の更新をスケジュールする

エンドツーエンド暗号化 (E2EE) 機能には、Linux VDA上の**ctxcertmgr**サービスが管理する自己署名証明書とその秘密キーが必要です。

ctxcertmgrサービスが起動または再起動すると、新しい自己署名証明書が作成されます。デフォルトでは、**ctxcertmgr**サービスは2年ごとに午前2時に証明書 (秘密キーを含む) を更新します。次のようなレジストリ設定で証明書の更新をスケジュールすることもできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_SZ" -v "CaRotationStartDate" -d "2023-10-19" --force



/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_SZ" -v "CaRotationTime" -d "00:45:30" --force



/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -t
"REG_DWORD" -v "CaRotationPeriod" -d "0x00000005" --force
```

上記の例では、最初の証明書更新時刻は2023-10-19の00:45:30に設定されています。その後、**ctxcermgr**サービスは5日ごとに00:45:30に証明書を更新します。スケジュールされた日付と時刻は、Linux VDA上の日付と時刻です。

DTLS を使用してユーザーセッションを保護する

DTLS 暗号化は、7.18 リリース以降で完全にサポートされている機能です。デフォルトでは、この機能は Linux VDA で有効になっています。詳細については、「Transport Layer Security」を参照してください。

DTLS 暗号化を有効にする

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、**HDX™ Adaptive Transport** ポリシーが **Preferred** または **Diagnostic mode** に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**/opt/Citrix/VDA/sbin** にある **enable_vdassl.sh** ツールを使用して SSL 暗号化を有効 (または無効) にします。このツールで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -h` コマンドを実行してください。

> **注:**
>
> Citrix Workspace™ アプリでどのバージョンの DTLS が使用されているかを確認します。Linux VDA と Citrix Workspace アプリの両方で同じバージョンの DTLS が使用されていることを確認してください。

TLS を使用してユーザーセッションを保護する

バージョン 7.16 以降、Linux VDA はセキュアなユーザーセッションのための TLS 暗号化をサポートしています。TLS 暗号化はデフォルトで無効になっています。

TLS 暗号化を有効にする

セキュアなユーザーセッションのために TLS 暗号化を有効にするには、Linux VDA と Delivery Controller™ (Controller) の両方に証明書をインストールし、TLS 暗号化を有効にします。

Linux VDA に証明書をインストールする

PEM 形式のサーバー証明書と CRT 形式のルート証明書を取得します。サーバー証明書には以下のセクションが含まれます。

- 証明書
- 暗号化されていない秘密鍵
- 中間証明書 (オプション)

サーバー証明書の例:

サーバー証明書の例(/en-us/linux-virtual-delivery-agent/2411/media/example-server-certificate.png)

TLS 暗号化を有効にする

LINUX VDA で TLS 暗号化を有効にする

Linux VDA では、**/opt/Citrix/VDA/sbin** ディレクトリにある `enable_vdassl.sh` スクリプトを使用して TLS 暗号化を有効 (または無効) にします。スクリプトで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -help` コマンドを実行してください。

SSL を有効にするスクリプト(/en-us/linux-virtual-delivery-agent/2411/media/script-for-enabling-ssl.png)

ヒント: 各 Linux VDA サーバーにはサーバー証明書を、各 Linux VDA サーバーおよびクライアントにはルート証明書をインストールする必要があります。

コントローラーでTLS暗号化を有効にする

> **注:**
>
> TLS暗号化は、デリバリーグループ全体に対してのみ有効にできます。特定のアプリケーションに対してTLS暗号化を有効にすることはできません。

コントローラーのPowerShellウィンドウで、ターゲットデリバリーグループのTLS暗号化を有効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

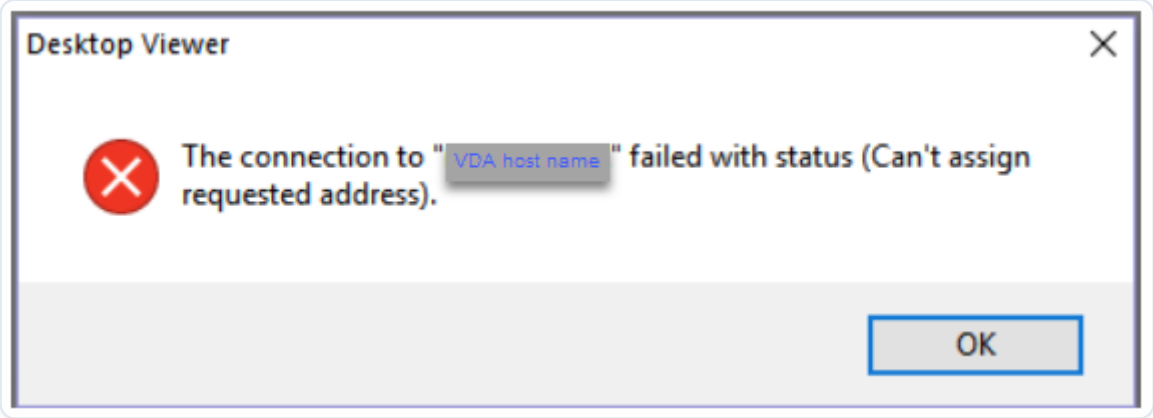
> **注:**
>
> ICA®セッションファイルにVDA FQDNのみが含まれるようにするには、`Set-BrokerSite –DnsResolutionEnabled $true` コマンドを実行することもできます。このコマンドはDNS解決を有効にします。DNS解決を無効にすると、ICAセッションファイルはVDAのIPアドレスを開示し、`SSLProxyHost` や `UDPDTLSPort` などのTLS関連項目にのみFQDNを提供します。

コントローラーでTLS暗号化を無効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

トラブルシューティング

公開デスクトップセッションにアクセスしようとすると、Windows版Citrix Workspace™アプリで次の「Can't assign requested address」エラーが発生する可能性があります。



回避策として、**hosts**ファイルに次のようなエントリを追加します。

```
<IP address of the Linux VDA>      <FQDN of the Linux VDA>
```

Windowsマシンでは、**hosts**ファイルは通常`C:\Windows\System32\drivers\etc\hosts`にあります。

セッションの信頼性

Citrix® は、サポートされているすべての Linux プラットフォームにセッションの信頼性機能を導入しています。セッションの信頼性はデフォルトで有効になっています。

セッションの信頼性により、ネットワークの中断時でも ICA セッションがシームレスに再接続されます。セッションの信頼性の詳細については、「自動クライアント再接続とセッションの信頼性」を参照してください。

> **注**:
>
> セッションの信頼性接続を介して送信されるデータは、デフォルトでプレーンテキストです。セキュリティ上の理由から、TLS 暗号化を有効にすることをお勧めします。TLS 暗号化の詳細については、「TLS を使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studio のポリシー設定

Citrix Studio でセッションの信頼性について、以下のポリシーを設定できます。

- セッションの信頼性接続
- セッションの信頼性タイムアウト
- セッションの信頼性ポート番号
- 再接続 UI の透過レベル

詳細については、「セッションの信頼性ポリシー設定」および「自動クライアント再接続ポリシー設定」を参照してください。

> **注:**
>
> **セッションの信頼性接続**または**セッションの信頼性ポート番号**ポリシーを設定した後、設定を有効にするには、VDA サービスと HDX™ サービスをこの順序で再起動します。

Linux VDA の設定

- **セッションの信頼性 TCP リスナーの有効化/無効化**

デフォルトでは、セッションの信頼性 TCP リスナーは有効になっており、ポート 2598 でリッスンしています。リスナーを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\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ます。TCP リスナーを無効にしても、セッションの信頼性は無効になりません。**セッションの信頼性接続**ポリシーを通じて機能が有効になっている場合、セッションの信頼性は他のリスナー（SSL など）を通じて引き続き利用できます。

- **セッションの信頼性ポート番号**

次のコマンドを使用して、セッションの信頼性ポート番号を設定することもできます（例としてポート番号 2599 を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cgp" -v "PortNumber" -d "2599"
```

注:

設定を有効にするには、HDX サービスを再起動します。ポート番号が **Citrix Studio** のポリシー設定を通じて設定されている場合、Linux VDA での設定は無視されます。VDA のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように構成されていることを確認してください。

- **サーバーからクライアントへのキープアライブ間隔**

セッションでアクティビティ（たとえば、マウスの動きや画面の更新がない場合）がない場合、Linux VDAとクライアント間でキープアライブメッセージが送信されます。キープアライブメッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントから応答がない場合、クライアントが再接続するまでセッションは中断されます。この設定は、連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpServerToClientKeepAlive"  -d  "10"  --force
```

- **クライアントからサーバーへのキープアライブ間隔**

この設定は、ICA®クライアントからLinux VDAに送信される連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpClientToServerKeepAlive"  -d  "10"  --force
```

トラブルシューティング

ポリシー設定を通じてセッションの信頼性を有効にした後、セッションを起動できません。

この問題を回避するには、次の手順を実行します。

1. Citrix Studioでポリシー設定を通じてセッションの信頼性を有効にした後、VDAサービスとHDXサービスがこの順序で再起動されていることを確認します。
2. VDAで、セッションの信頼性TCPリスナーが実行されていることを確認するために、次のコマンドを実行します（例としてポート2598を使用）。

   ```
   netstat -an | grep 2598
   ```

   セッションの信頼性ポートにTCPリスナーがない場合、次のコマンドを実行してリスナーを有効にします。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations
   \cgp"   -v  "fEnableWinStation" -d "0x00000001"
   ```

セッションレコーディング (プレビュー)

Linux VDAでホストされているセッションを記録および再生できます。

セッションレコーディングを有効または無効にする

Linux VDAのセッションレコーディングを有効または無効にするには、**SmAudAllowed**をそれぞれ**1**または**0**に設定します。次のコマンドを使用できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000001" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000000" --force
```

注:

> Linux VDAでセッションレコーディングを有効にすると、ユーザーはセッションにログオンしたときに、セッションが記録されていることを通知されます。

記録のファイルサイズを指定する

記録のサイズが大きくなると、記録ファイルのダウンロードに時間がかかり、再生中にシークスライダーを使用して移動する際の応答が遅くなります。ファイルサイズを制御するには、ファイルのしきい値制限を指定します。記録がこの制限に達すると、現在のファイルは閉じられ、記録を続行するために追加のファイルが作成されます。このアクションはロールオーバーと呼ばれます。

次のコマンドを使用して、ロールオーバーの2つのしきい値を指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverFileSizeInMB" -d "0x00000032" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverTimeInHours" -d "0x0000000c" --force
```

- **RolloverFileSizeInMB**。現在のファイルは、そのサイズに達すると閉じられ、新しいファイルが開きます。デフォルトでは、サイズが50 MBを超えるとロールオーバーが発生します。サポートされる値: 10～300。
- **RolloverTimeInHours**。期間に達すると、現在のファイルは閉じられ、新しいファイルが開きます。デフォルトでは、セッションが12時間記録されるとロールオーバーが発生します。サポートされる値: 1～24。

ロールオーバーは、上記の2つの条件のうち最初に満たされたときに発生します。たとえば、ファイルサイズに17 MB、期間に6時間を指定したとします。記録が3時間で17 MBに達すると、セッションレコーディングはファイルを閉じ、新しいファイルを開きます。

多数の小さなファイルが作成されるのを防ぐため、ファイルサイズに指定された値に関係なく、少なくとも1時間が経過するまでロールオーバーは発生しません。この規則の例外は、ファイルサイズが300 MBを超える場合です。

記録の保存場所を指定する

記録ファイルは、デフォルトで`/var/xdl/session_recordings`に保存されます。別のパスを指定するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_SZ" -v
"Path" -d "<your custom storage path>" --force
```

記録は、ローカルドライブまたはネットワークパスを指すマウントポイントに保存できます。設定したストレージパスに適切なアクセス許可を構成し、ユーザー`ctxsrvr`にそのパスへの書き込み権限を付与します。

録画の表示

録画を表示するには、Session Recording PlayerまたはSession Recording Web Playerをインストールするため、次の手順を実行します。

1. Citrixアカウントの資格情報を使用してCitrix Virtual Apps and Desktopsダウンロードページにアクセスし、製品ファイルをダウンロードします。ファイルを解凍します。
2. SessionRecordingPlayer.msiとSessionRecordingWebPlayer.msiをダブルクリックし、指示に従ってインストールを完了します。

ヒント:

Session Recording Web Playerを使用するには、Session Recordingサーバーにのみインストールし、録画がSession Recordingサーバーで利用可能であることを確認してください。詳細については、Citrix Session Recording ドキュメントを参照してください。

制限事項

- 仮想アプリセッションの場合、録画通知が中央に表示されない場合があります。

仮想チャネルSDK (プレビュー)

Linux VDA用の仮想チャネルソフトウェア開発キット (SDK) を使用すると、VDA上で実行するサーバー側アプリケーションを作成できます。詳細については、Citrix Virtual Channel SDK for the Linux VDAドキュメントを参照してください。

Linux VDA用Citrix Virtual Channel SDKは、Citrix Virtual Apps and Desktopsダウンロードページからダウンロードできます。適切なバージョンのCitrix Virtual Apps and Desktopsを展開し、**Components**をクリックしてLinux VDAダウンロードを選択します。

Wayland (プレビュー)

Linux VDA は、RHEL 9.4/9.2、Rocky Linux 9.4/9.2、および Ubuntu 22.04 の GNOME で Wayland をサポートしています。Wayland では、以下の機能が完全にテストされています。

- オーディオ
- クリップボード
- クライアントドライブマッピング (CDM)
- 印刷
- USBデバイスリダイレクト

> **注:**
>
> - HDX™ 3D Pro はサポートされていません。
> - Linux仮想アプリセッションはサポートされていません。

Waylandを有効にする

Waylandを使用するには、次のコマンドを実行して、レジストリキー **EnableWayland** を **1** に設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Wayland" -t "REG_DWORD" -v
"EnableWayland" -d "0x00000001" --force
```

デフォルトでは、レジストリキー **EnableWayland** は **0** に設定されており、これはX11が使用されることを意味します。

Waylandが使用中であるかを確認する

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が使用中の場合、出力に **'wayland'** が表示されます。

制限事項

Waylandを使用している場合、以下の制限が確認されています。

- クライアントデバイスのキーボードレイアウトは、VDAのキーボードレイアウトと同期されません。

ベストプラクティス

このセクションでは、次の手順について説明します。

- WebSocket 用の自己署名証明書を構成する(/ja-jp/linux-virtual-delivery-agent/2411/best-practices/configure-self-signed-certificates-for-websocket.html)
- Machine Creation Services (MCS) を使用して Google Cloud Platform (GCP) 上に Linux VDA を作成する(/ja-jp/linux-virtual-delivery-agent/2411/best-practices/create-linux-vda-on-gcp-using-mcs.html)
- Ansible を使用して展開を管理する(/ja-jp/linux-virtual-delivery-agent/2411/best-practices/manage-your-deployment-using-ansible.html)

WebSocket 用の自己署名証明書を構成する

2402 LTSR の初回リリース以降、Citrix Virtual Apps and Desktops では、Citrix Brokering Protocol (CBP) を介した WebSocket テクノロジーを使用して、VDA と Delivery Controller 間の通信を容易にすることができます。この機能では、VDA から Delivery Controller への通信に TLS ポート 443 のみが必要です。詳細については、Citrix Virtual Apps and Desktops ドキュメントの「VDA と Delivery Controller 間の WebSocket 通信」を参照してください。

WebSocket は、クライアントとサーバー間のリアルタイムで双方向の通信を可能にする強力なテクノロジーです。ただし、特に **wss://** を使用する場合に安全な接続を確保するには、特に開発環境やテスト環境では、自己署名証明書の設定が必要になることがよくあります。この記事では、WebSocket 用の自己署名証明書を構成するためのベストプラクティスについて説明します。

ステップ 1: (ドメインに参加していない VDA のみ) DNS サーバーを構成する

- **Ubuntu および Debian VDA の場合:**

1. **/etc/systemd/resolved.conf** を次のように編集して DNS 設定を変更します。

```
[Resolve]

# Some examples of DNS servers which may be used for DNS= and
FallbackDNS=:
```

```
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com
2606:4700:4700::1111#cloudflare-dns.com
2606:4700:4700::1001#cloudflare-dns.com

# Google:     8.8.8.8#dns.google 8.8.4.4#dns.google
2001:4860:4860::8888#dns.google 2001:4860:4860::8844#dns.google

# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net
2620:fe::fe#dns.quad9.net 2620:fe::9#dns.quad9.net

DNS=<DNS IP address>

#FallbackDNS=

#Domains=

#DNSSEC=no

#DNSOverTLS=no

#MulticastDNS=no

#LLMNR=no

#Cache=no-negative

#CacheFromLocalhost=no

#DNSStubListener=yes

#DNSStubListenerExtra=

#ReadEtcHosts=yes

#ResolveUnicastSingleLabel=no
```

2. systemd-resolved サービスを再起動します。

```
sudo service systemd-resolved restart
```

詳細については、`<https://notes.enovision.net/linux/changing-dns-with-resolve>` を参照してください。

- **Amazon Linux 2、RHEL、Rocky Linux、および SUSE VDA の場合:**

1. 次の **nmcli** コマンドを実行して、接続名の一覧を取得します。

```
sudo nmcli connection
```

2. 別の **nmcli** コマンドを実行して、DNS IP アドレスを設定します。

```
sudo nmcli con mod {connectionNameHere} ipv4.dns "<dns ip address>"
```

たとえば、次のコマンドを使用して DNS IP アドレスを 192.168.2.254 に設定できます。

```
sudo nmcli con mod eth0 ipv4.dns "192.168.2.254"
```

3. 次のいずれかのコマンドを実行して、新しい DNS 設定をリロードします。

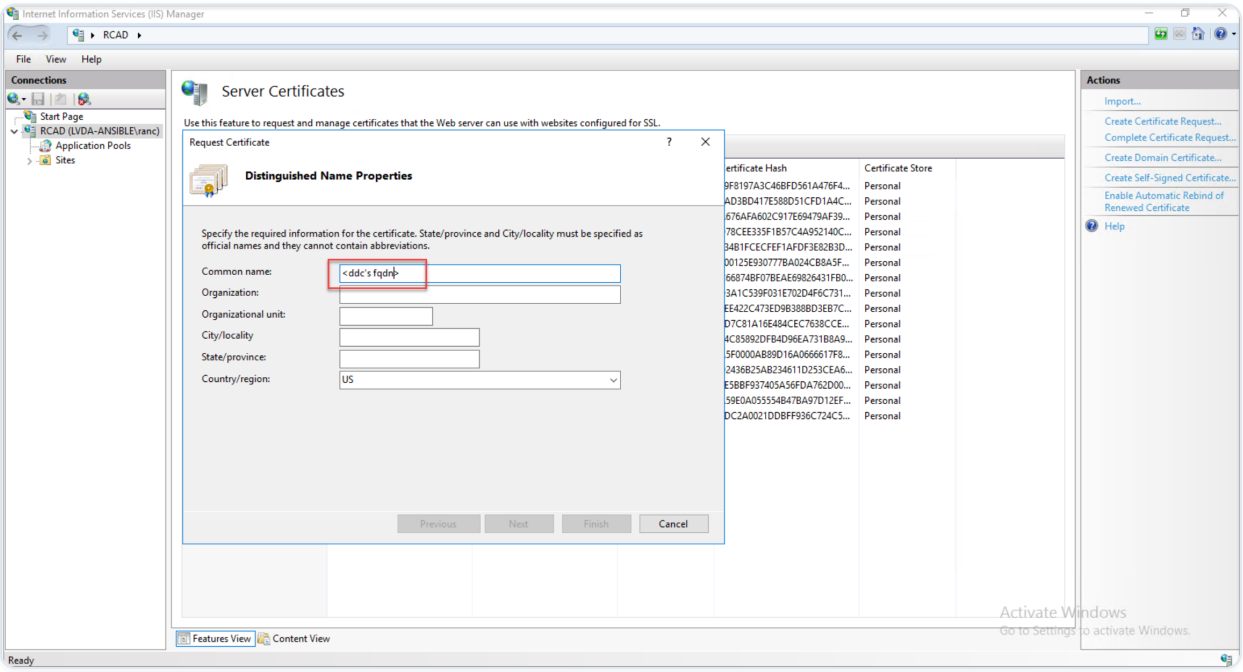
```
sudo systemctl restart NetworkManager.service
```

```
sudo nmcli connection reload
```

詳細については、`<https://www.cyberciti.biz/faq/change-dns-ip-address-rhel-redhat-linux/>` を参照してください。

ステップ 2: 認証局 (CA) から証明書を要求する

1. 証明書要求を開始します。証明書要求を開始するときは、Delivery Controller の完全修飾ドメイン名 (FQDN) を入力します。



2. 証明書の堅牢なセキュリティを確保するために、2048ビット以上のビット長を選択します。

   ビット長を選択(/en-us/linux-virtual-delivery-agent/2411/media/select-a-bit-length.png)

3. 簡単に識別できるように、証明書要求ファイルにわかりやすい名前を割り当てます。

   証明書要求ファイルに名前を付ける(/en-us/linux-virtual-delivery-agent/2411/media/name-the-certificate-request-file.png)

4. 生成された証明書要求ファイルをメモ帳などのテキストエディターで開き、その中のすべてのコンテンツを選択します。

   生成された証明書要求ファイルを開く(/en-us/linux-virtual-delivery-agent/2411/media/open-the-generated-certificate-request-file.png)

5. Web証明書サーバーにログインし、証明書を要求する手順に進みます。

   Web証明書サーバーにログインする(/en-us/linux-virtual-delivery-agent/2411/media/log-in-to-your-web-certificate-server.png)

   高度な証明書要求(/en-us/linux-virtual-delivery-agent/2411/media/advanced-certificate-request.png)

6. コピーした要求ファイルの内容をWebサーバーの適切なフィールドに貼り付け、**Webサーバー**証明書テンプレートを選択します。

   証明書要求を送信する(/en-us/linux-virtual-delivery-agent/2411/media/certificate-template.png)

7. Base 64エンコード形式で証明書をダウンロードします。

証明書をダウンロードする(/en-us/linux-virtual-delivery-agent/2411/media/download-the-certificate.png)

8. 証明書要求を完了します。ダウンロードが完了すると、証明書要求プロセスは完了です。

   証明書要求を完了する(/en-us/linux-virtual-delivery-agent/2411/media/complete-the-certificate-request.png)

ステップ 3: Delivery Controller FQDN 証明書をバインドする

1. Delivery Controller FQDN 証明書をPFXファイルとしてエクスポートします。

   Delivery Controller FQDN 証明書をエクスポートする(/en-us/linux-virtual-delivery-agent/2411/media/export-the-delivery-controller-fqdn-certificate.png)

2. エクスポートしたPFX証明書を、Delivery Controllerサーバーのインターネットインフォメーションサービス (IIS) にインポートします。

   エクスポートしたPFX証明書をインポートする(/en-us/linux-virtual-delivery-agent/2411/media/import-the-exported-pex-certificate.png)

3. インポートした証明書をIIS内のデフォルトのWebサイトにバインドします。

   インポートした証明書をバインドする(/en-us/linux-virtual-delivery-agent/2411/media/bind-the-imported-certificate.png)

4. バインドプロセス中に、インポートした特定の証明書を選択していることを確認してください。

   バインド中にインポートした証明書を選択する(/en-us/linux-virtual-delivery-agent/2411/media/select-the-imported-certificate-during-binding.png)

ステップ 4: Linux VDAでCA証明書を保存および更新する

1. CA証明書をダウンロードします。例:

   CA証明書をダウンロードする(/en-us/linux-virtual-delivery-agent/2411/media/download-ca-certificate.png)

2. CA証明書を配置して更新します。

   - **RHEL、Rocky Linux、およびAmazon Linux 2の場合:**

     trust anchor <path/CA certificate> コマンドを使用してCA証明書を追加します。**/etc/pki/ca-trust/source/anchors** ディレクトリに証明書が手動で配置されていないことを確認して

ください。読み取り専用フィールドに関連するエラーが発生した場合は、そのディレクトリに存在する証明書をすべて削除してください。

- **SUSE、Ubuntu、Debian の場合:**

  ルートCA証明書を **/usr/local/share/ca-certificates** ディレクトリに配置します。証明書に **.crt** 拡張子がない場合は、それに応じて名前を変更します。その後、**update-ca-certificate** コマンドを実行します。

Machine Creation Services™ (MCS) を使用して Google Cloud Platform (GCP) に Linux VDA を作成する

MCS を使用して GCP に Linux VDA を作成するには、次の手順を実行します。

ステップ 1: GCP に Linux 仮想マシン (VM) を作成する

ステップ 2: GCP サービスアカウントを作成する

ステップ 3: Citrix Studio で GCP へのホスト接続を作成する

ステップ 4: Linux VDA マスターイメージを準備する

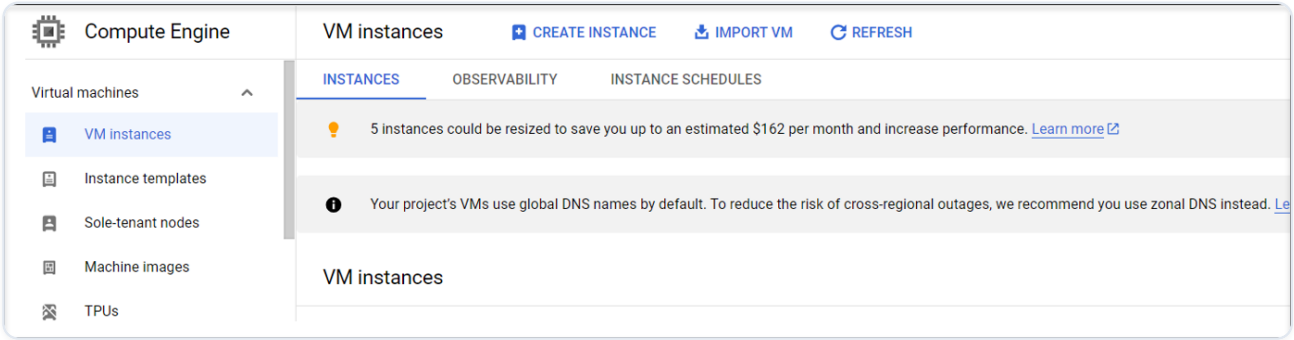
ステップ 5: マシンカタログを作成する

ステップ 6: デリバリーグループを作成する

ステップ 1: GCP に Linux VM を作成する

ステップ 1a: Google Cloud コンソールで、**Compute Engine > VM instances** に移動します。

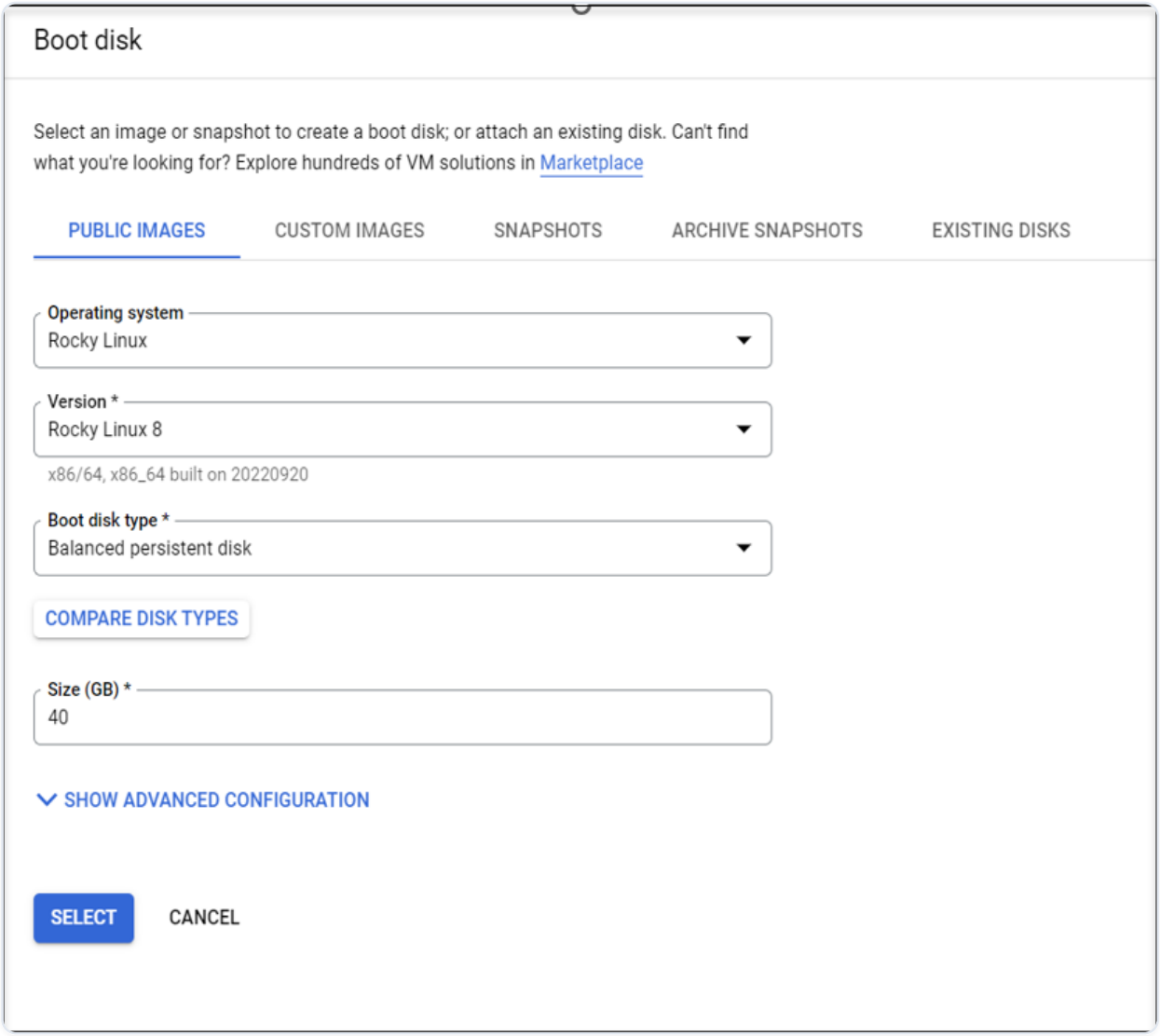
ステップ 1b: VM instances ページで、**CREATE INSTANCE** をクリックして VM インスタンスを作成します。



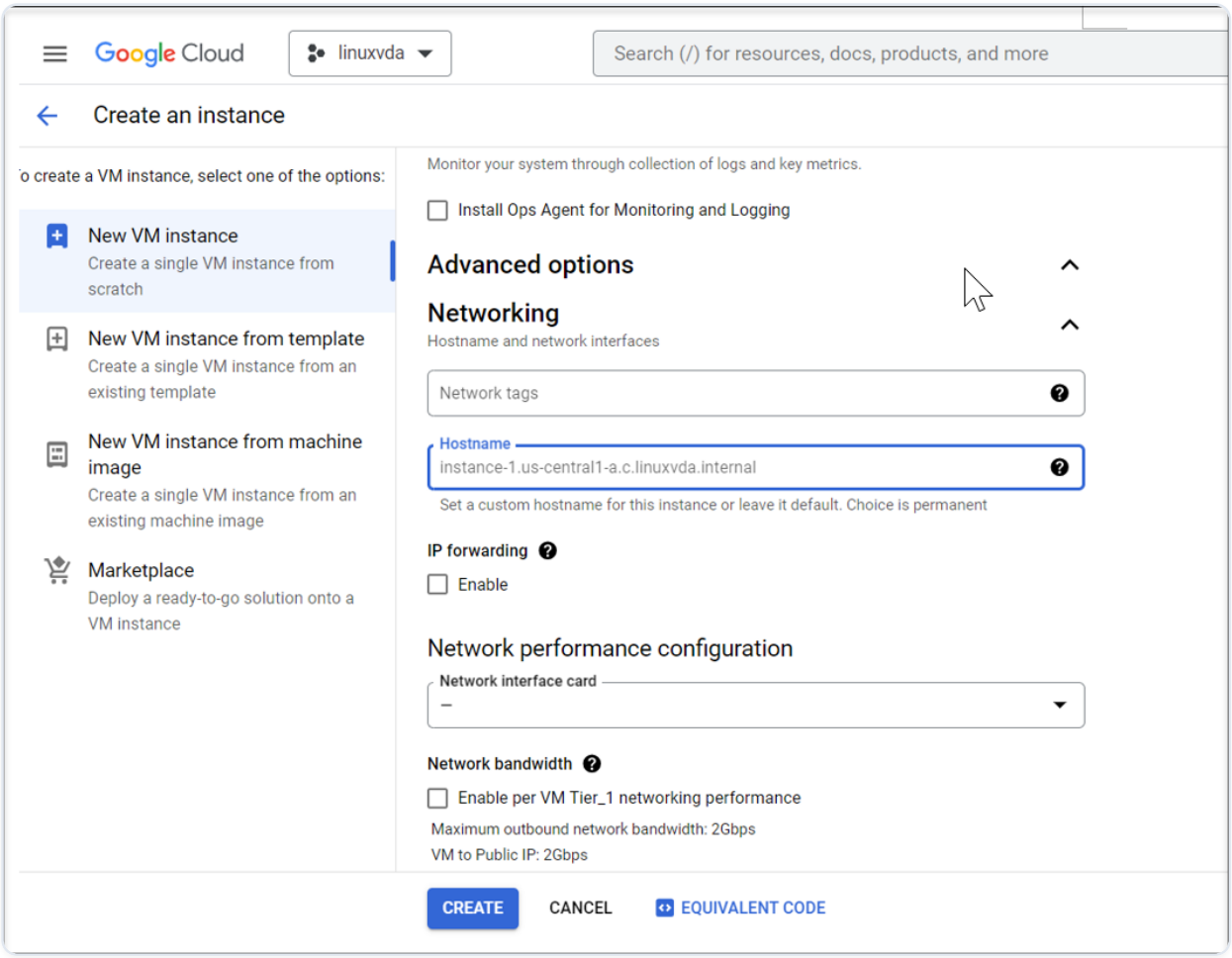
ステップ 1c: 次の設定を行い、その他の設定はデフォルトのままにします。

- VM インスタンスの名前を入力します。

- VM をホストするリージョンとゾーンを選択します。
- (オプション) VMにGPUを追加します。詳細については、この記事の後半にある**Step 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:

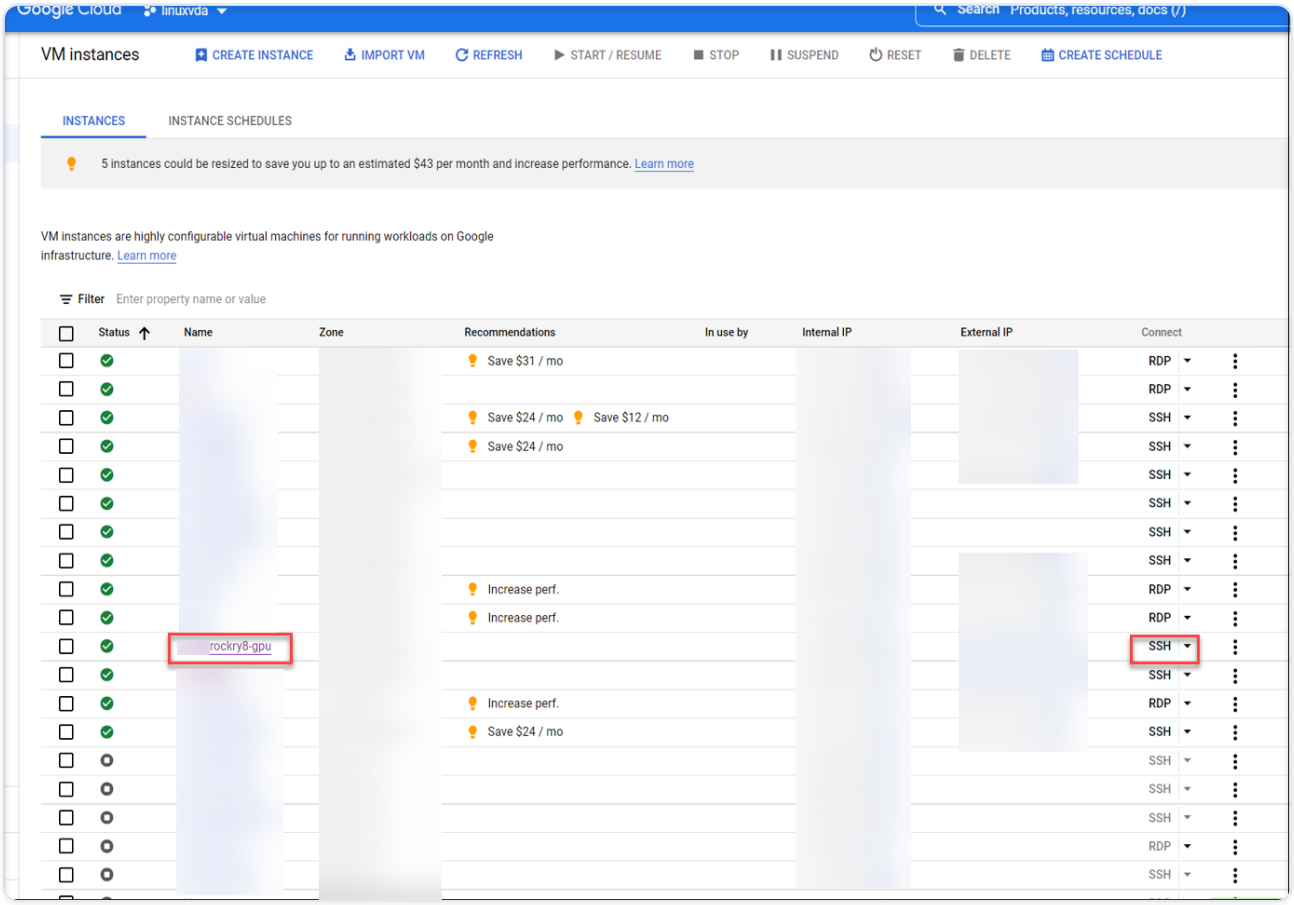


- **Advanced options > Networking**に移動し、**Hostname**フィールドをFQDNに設定します。

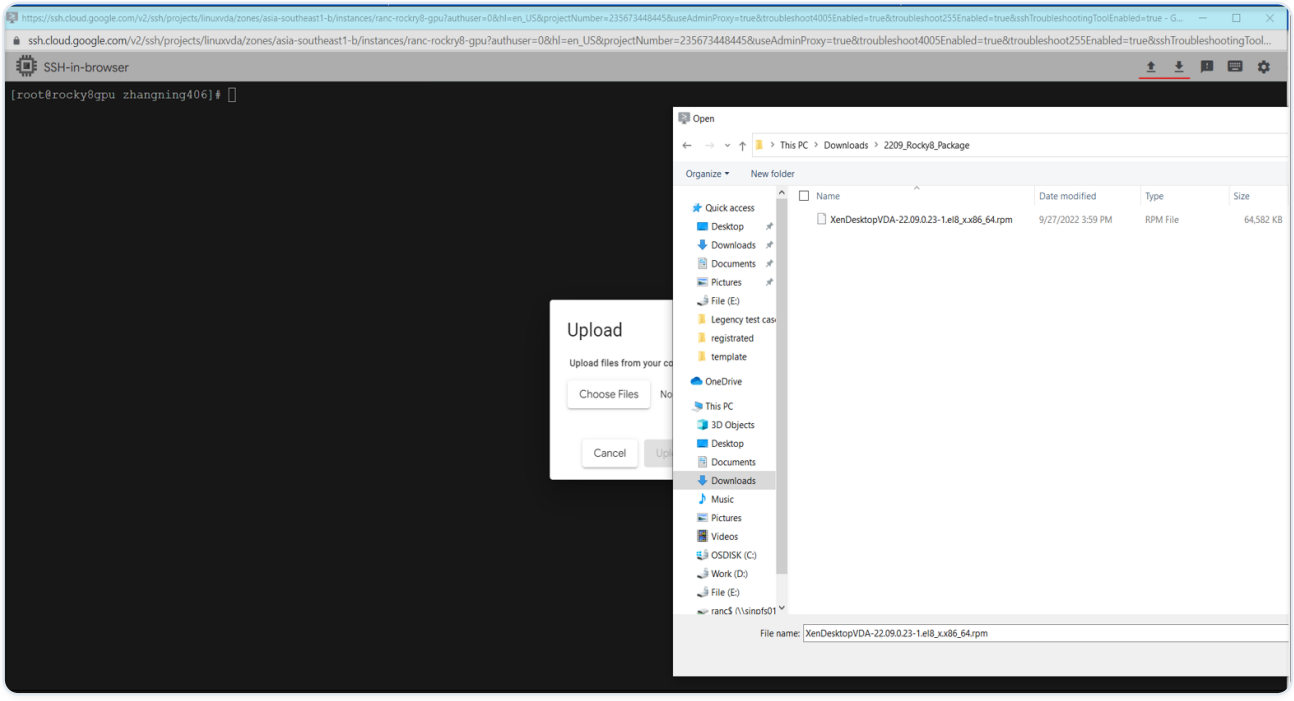


Step 1d: CreateをクリックしてVMインスタンスを作成します。

Step 1e: VMが作成されたら、**Compute Engine**ダッシュボードに戻り、リストからVMインスタンスを見つけ、SSHボタンをクリックしてVMに接続します。



Step 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。

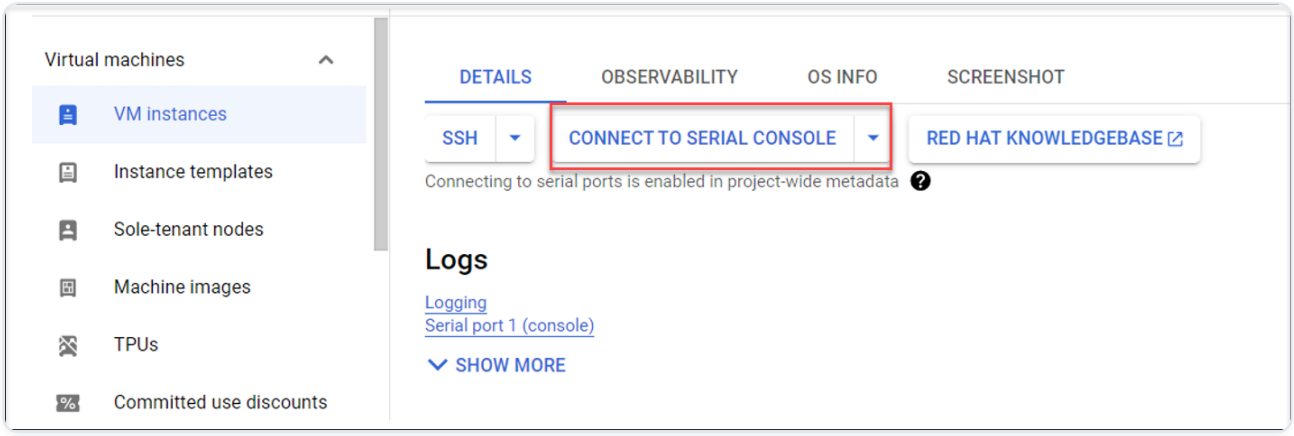


Step 1g: SSHを使用したVMへのアクセス障害を防ぎます。

VMは再起動後に到達不能になる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認します。その後、VMを再起動した後、コンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```



Step 2: GCPサービスアカウントを作成する

このセクションでは、GCPサービスアカウントの作成について説明します。これには、サービスアカウントキーの作成と、サービスアカウントに必要なロールの付与が含まれます。

> **注:**

> GCPサービスアカウントを作成するには、Service Account Admin (roles/iam.serviceAccountAdmin) IAMロールが付与されていることを確認してください。

> **ステップ2a:** Google Cloudコンソールで、**IAMと管理 > サービスアカウント**に移動し、**サービスアカウントを作成**タブをクリックします。

サービスアカウントを作成タブ(/en-us/linux-virtual-delivery-agent/2411/media/create-service-account-tab.png)

ステップ2b: サービスアカウントの詳細ステップで、フィールドに値を入力し、**作成して続行**をクリックします。

サービスアカウントの詳細(/en-us/linux-virtual-delivery-agent/2411/media/service-account-details-on-gcp.png)

ステップ2c: オプションの手順をスキップし、下部にある**完了**をクリックします。

GCPサービスアカウントの作成が完了しました(/en-us/linux-virtual-delivery-agent/2411/media/done-with-creating-a-gcp-service-account.png)

ステップ2d: 再び**IAMと管理 > サービスアカウント**に移動し、**サービスアカウント**タブをクリックします。新しく作成したサービスアカウントを見つけ、**キー**タブに移動し、**キーを追加 > 新しいキーを作成 > JSON > 作成**をクリックします。

GCPサービスアカウントキーを追加(/en-us/linux-virtual-delivery-agent/2411/media/add-a-gcp-service-account-key-new.png)

キーファイルをダウンロード(/en-us/linux-virtual-delivery-agent/2411/media/download-key-file.png)

> **注:**
>
> キーファイルをダウンロードした後、再度ダウンロードすることはできません。

> **ステップ2e:** Google Cloudコンソールで、**IAMと管理 > IAM**に移動し、**追加**をクリックします。**新しいメンバー**フィールドで新しく作成したサービスアカウントを検索して選択し、そのサービスアカウントにリソースへのアクセスを許可するためのロールを選択します。**別のロールを追加**をクリックしてロールの付与を続け、新しく作成したサービスアカウントに以下のすべてのロールが付与されていることを確認します。

- **Compute 管理者**
- **Storage 管理者**
- **Cloud Build 編集者**
- **サービスアカウントユーザー**
- **クラウド データストア ユーザー**
- **Compute Instance 管理者 (ベータ版)**
- **所有者**

例:



ステップ3：Citrix StudioでGCPへのホスト接続を作成する

Google Cloud Platform仮想化環境に従ってGCP環境をセットアップし、以下の手順を実行してGCPへのホスト接続を作成します。

1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで **構成 > ホスティング > 接続とリソースの追加** を選択してホスト接続を作成します。クラウドのDelivery Controllerの場合、Citrix Cloud™上のWebベースのStudioコンソールで **管理 > ホスティング > 接続とリソースの追加** を選択してホスト接続を作成します。
2. **接続とリソースの追加** ウィザードで、接続の種類として **Google Cloud Platform** を選択します。

   例：Citrix Cloud上のWebベースのStudioコンソールで、:

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2411/media/add-conn-connection.png)
3. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
4. ウィザードの指示に従ってページを進めます。特定のページの内容は、選択した接続の種類によって異なります。各ページを完了したら、**次へ** を選択して **概要** ページに到達します。詳細については、MCSを使用してドメインに参加していないLinux VDAを作成するの記事のステップ2：ホスト接続を作成するを参照してください。

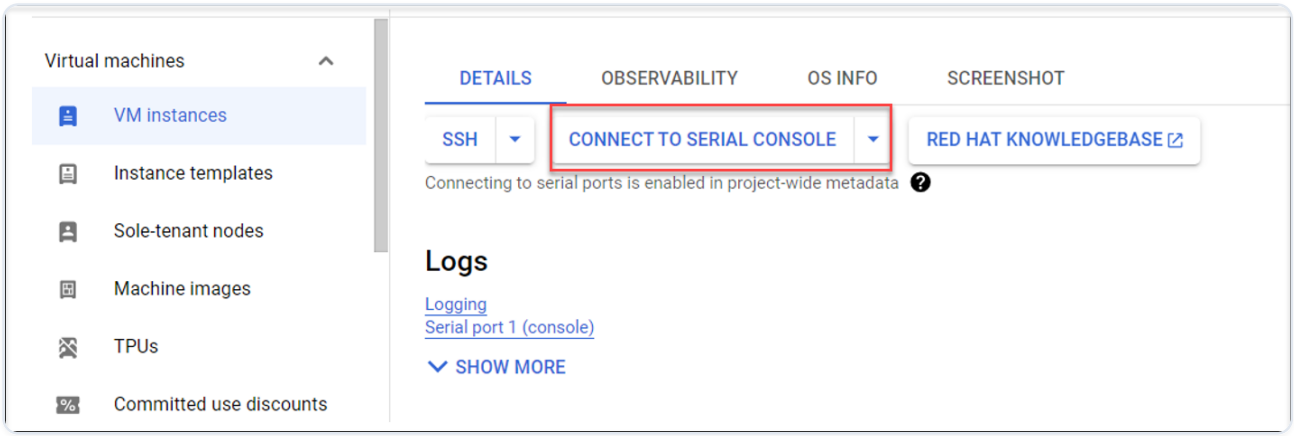
ステップ4：Linux VDAマスターイメージを準備する

ステップ4a： (RHEL 8.x/9.xおよびRocky Linux 8.x/9.xの場合) イーサネット接続を構成します。

GCPでホストされているRHEL 8.x/9.xおよびRocky Linux 8.x/9.xにLinux VDAをインストールした後、VMの再起動後にイーサネット接続が失われ、Linux VDAに到達できなくなる可能性があります。この問題を回避するには、VMに初めてログオンするときにrootパスワードを設定し、rootとしてVMにログオンできることを確認します。その後、VMを再起動してからコンソールで次のコマンドを実行します。

```
nmcli dev connect eth0

systemctl restart NetworkManager
```



手順4b： テンプレートVMにLinux VDAパッケージをインストールします。

テンプレートVMで、Linux VDAパッケージをインストールするには、次の手順を実行します。

1. .NETをインストールします。

   .NET Runtimeに加えて、Linux VDAをインストールまたはアップグレードする前に、サポートされているすべてのLinuxディストリビューションに.ASP.NET Core Runtimeをインストールする必要があります。Amazon Linux 2にはバージョン6が必要です。その他のディストリビューションにはバージョン8が必要です。

   必要な.NETバージョンがLinuxディストリビューションに含まれている場合は、組み込みのフィードからインストールします。含まれていない場合は、Microsoftパッケージフィードから.NETをインストールします。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

2. Linux VDAパッケージをインストールします。

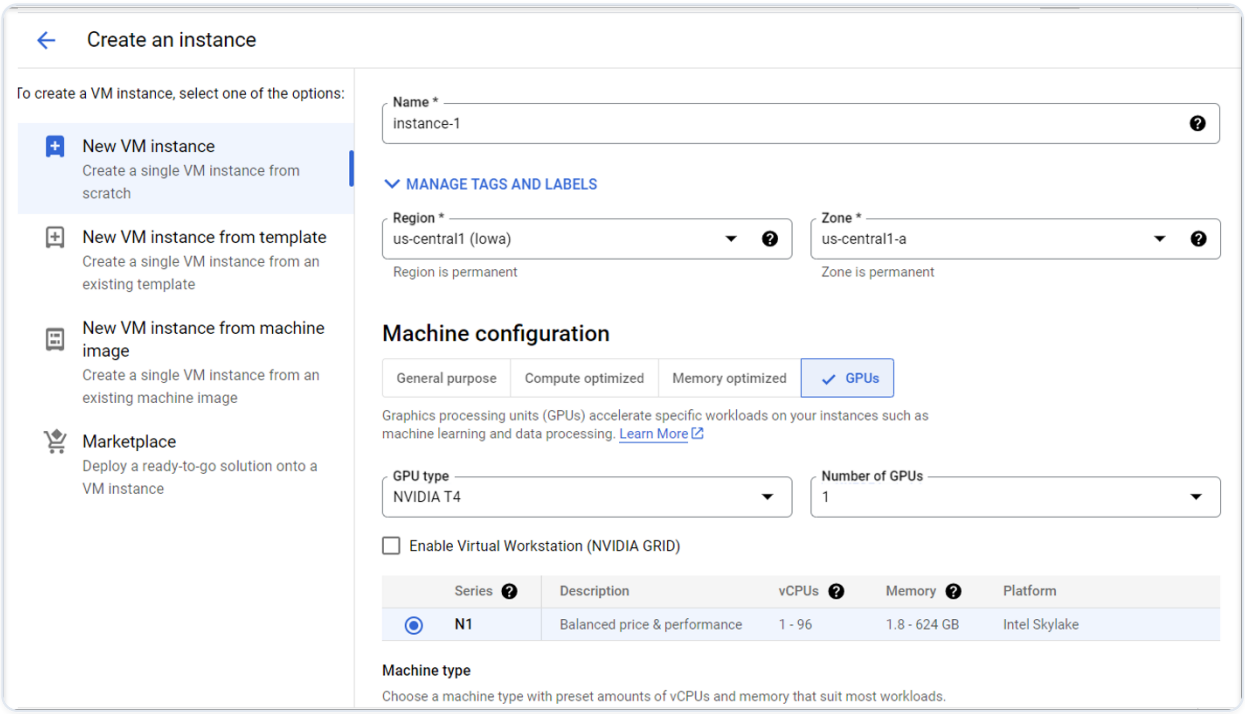
   ```
   sudo yum –y localinstall <PATH>/<Linux VDA RPM>
   ```

3. EPELリポジトリを有効にします。

   ```
   sudo rpm -ivh  https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   ```

手順4c： GCP上のテンプレートVMにグラフィックス処理ユニット（GPU）を追加します（オプション）。

1. Google Cloudコンソールで、テンプレートVMに1つ以上のGPUを追加します。GCPでのGPUの追加と削除については、`<https://cloud.google.com/compute/docs/gpus/add-remove-gpus>`を参照してください。



2. テンプレートVMに適切なGPUドライバーをインストールします。詳細については、`<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で実行されているNVIDIA GPUは、次のNVIDIAドライバーバージョンを使用する必要があります。

   - L4 GPUの場合：
     - Linux : 525.60.13 以降
   - A100 GPUの場合:
     - Linux : 450.80.02 以降
   - T4、P4、P100、およびV100 GPUの場合:
     - Linux : 410.79 以降
   - K80 GPUの場合 (サポート終了):
     - Linux : 410.79 - 最新のR470バージョン

   K80 GPUの場合、NVIDIAは、R470 driverブランチがデバッグサポートを受け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 Software Support Matrixを参照してください。

インストールスクリプト:

以下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。

```
https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

サポートされているオペレーティングシステム:

インストールスクリプトは、以下のLinuxディストリビューションでテストされました:

- デビアン 11
- レッドハットエンタープライズリナックス (RHEL) 8
- ロッキーリナックス 8
- ウブントゥ 20/22

他のLinuxディストリビューションでこのスクリプトを使用すると、インストールは失敗します。Linux VMの場合、このスクリプトはNVIDIAドライバーのみをインストールします。

a. インストールスクリプトをダウンロードします。

```
curl https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

b. スクリプトに完全なアクセス権を付与します。

```
chmod 777 install_gpu_driver.py
```

c. インストールスクリプトを実行します。

```
python3 install_gpu_driver.py
```

d. gdm3でWaylandを無効にします。

- お使いのディストリビューションのWayland構成ファイルを次のいずれかの場所で見つけます。
  - /etc/gdm3/custom.conf (ウブントゥ)

    - /etc/gdm/custom.conf (CentOS、RHEL、Rocky Linux)

  - sudo/root権限でファイルを開きます。
  - 行頭の # を削除して、**WaylandEnable=false** のコメントを解除します。
  - VMを再起動します。

e. NVIDIA 510以降のドライバーをインストールした場合は、GSPファームウェアを無効にします。

   GSPファームウェアが有効になっている場合は、NVIDIAモジュールパラメーター **NVreg_EnableGpuFirmware** を0に設定して無効にします。

   /etc/modprobe.d/nvidia.conf ファイルに次のエントリを追加して、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際には、次の点に注意してください。

   - 設定ファイルを作成および更新するコマンドを実行するには、sudoを使用します。
   - 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。

ステップ4d: MCS変数を構成します。

/etc/xdl/mcs/mcs.confファイルを編集して、MCS変数を構成します。以下は、ドメインに参加していないシナリオとドメインに参加しているシナリオで構成できるMCS変数です。

- **ドメインに参加していないシナリオの場合**

  デフォルトの変数値を使用するか、必要に応じて変数をカスタマイズできます（オプション）：

  `DOTNET_RUNTIME_PATH` =*path-to-install-dotnet-runtime **

  `DESKTOP_ENVIRONMENT` = *gnome | mate ** `REGISTER_SERVICE` =**Y | N**

  `ADD_FIREWALL_RULES` =**Y | N** `VDI_MODE` =**Y | N**

  `START_SERVICE` =**Y | N**

ドメインに参加していないシナリオのMCS変数(/en-us/linux-virtual-delivery-agent/2411/media/mcs-variables-for-non-domain-joined-scenarios.png)

- **ドメインに参加しているシナリオの場合**
  - `Use_AD_Configuration_Files_Of_Current_VDA`: 現在実行中のVDAの既存のAD関連構成ファイル（/etc/krb5.conf、/etc/sssd.conf、および/etc/samba/smb.conf）を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDAの同等のファイルと同じになります。ただし、`dns`および`AD_INTEGRATION`変数を構成する必要があります。デフォルト値はNです。これは、マスターイメージの構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。現在実行中のVDAをテンプレートVMとして使用するには、値をYに設定します。それ以外の場合は、Nに設定します。
  - `dns`: 各DNSサーバーのIPアドレスを設定します。最大4つのDNSサーバーを設定できます。
  - `NTP_SERVER`: NTPサーバーのIPアドレスを設定します。特に指定がない限り、ドメインコントローラーのIPアドレスです。
  - `WORKGROUP`: ワークグループ名を、ADで構成したNetBIOS名（大文字と小文字を区別）に設定します。設定しない場合、MCSはマシンホスト名の直後にあるドメイン名の一部をワークグループ名として使用します。たとえば、マシンアカウントが**user1.lvda.citrix.com**の場合、MCSは**lvda**をワークグループ名として使用しますが、正しい選択は**citrix**です。ワークグループ名が正しく設定されていることを確認してください。
  - `AD_INTEGRATION`: Winbind、SSSD、PBIS、またはCentrifyを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、サポートされているディストリビューションを参照してください。
  - `CENTRIFY_DOWNLOAD_PATH`: Server Suite Free（旧Centrify Express）パッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。
  - `CENTRIFY_SAMBA_DOWNLOAD_PATH`: Centrify Sambaパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をCentrifyに設定した場合にのみ有効になります。
  - `PBIS_DOWNLOAD_PATH`: PBISパッケージをダウンロードするためのパスを設定します。この値は、`AD_INTEGRATION`変数をPBISに設定した場合にのみ有効になります。
  - `UPDATE_MACHINE_PW`: マシンアカウントのパスワード更新の自動化を有効または無効にします。詳細については、マシンアカウントのパスワード更新を自動化するを参照してください。

- Linux VDA構成変数:

  `DOTNET_RUNTIME_PATH` =*path-to-install-dotnet-runtime **

  `DESKTOP_ENVIRONMENT` = *gnome | mate ** `SUPPORT_DDC_AS_CNAME` **=Y | N**

  `VDA_PORT` **=port-number** `REGISTER_SERVICE` **=Y | N**

  `ADD_FIREWALL_RULES` **=Y | N** `HDX_3D_PRO` **=Y | N**

  `VDI_MODE` **=Y | N** `SITE_NAME` **=dns-site-name | '<none\>'**

  `LDAP_LIST` **='list-ldap-servers' | '<none\>'** `SEARCH_BASE` **=search-base-set | '<none\>'**

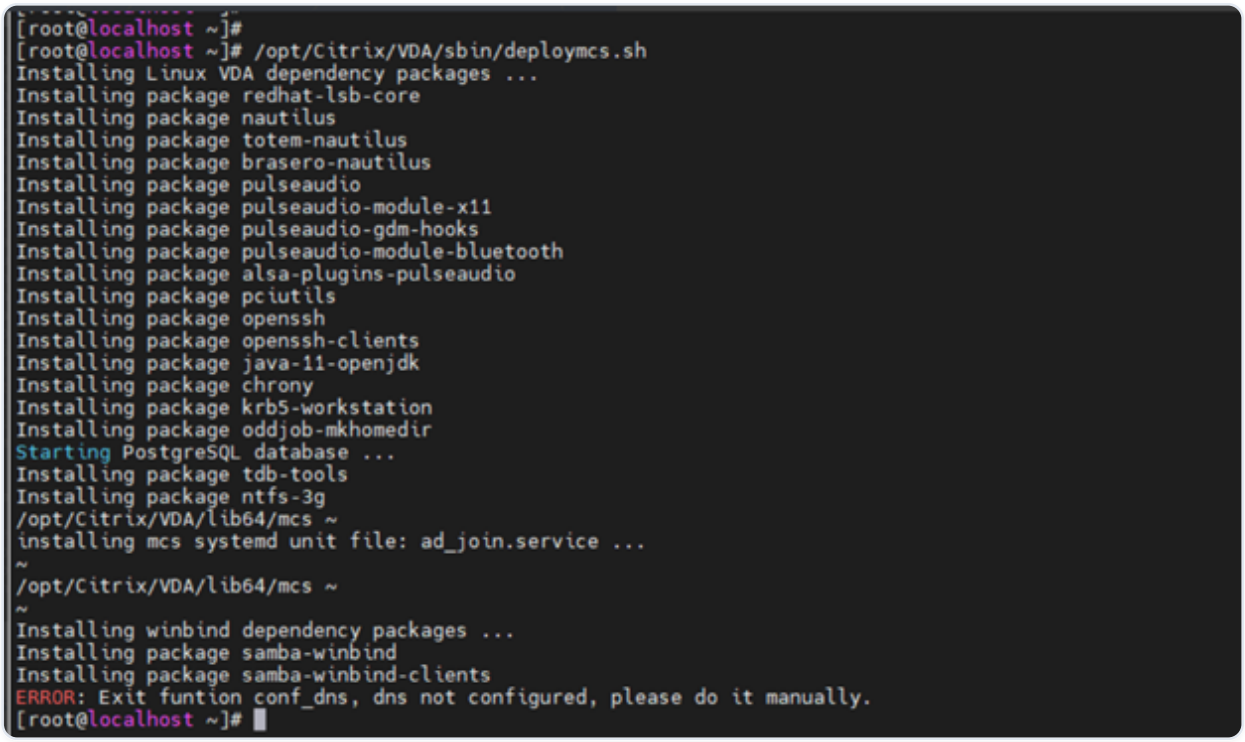
  `FAS_LIST` **='list-fas-servers' | '<none\>'** `START_SERVICE` **=Y | N**

  `TELEMETRY_SOCKET_PORT` **=port-number** `TELEMETRY_PORT` **=port-number**

ステップ 4e: マスターイメージを作成する

1. （SSSD + RHEL 8.x/9.xまたはRocky Linux 8.x/9.xのみ） `update-crypto-policies --set DEFAULT:AD-SUPPORT` コマンドを実行し、テンプレートVMを再起動します。
2. `/opt/Citrix/VDA/sbin/deploymcs.sh` を実行します。

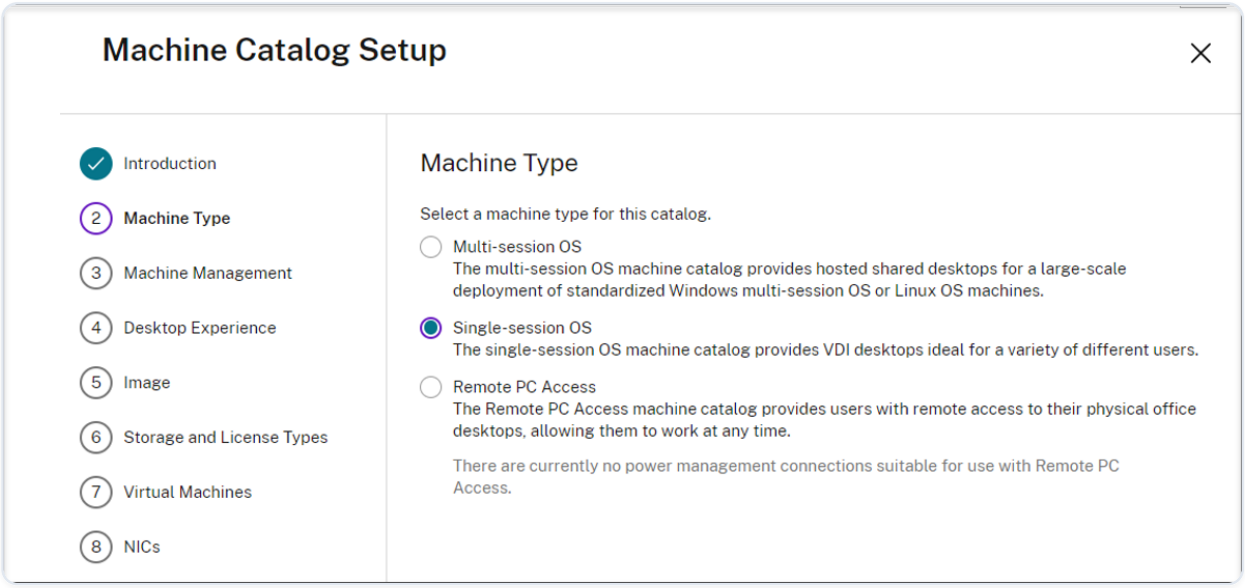
   ドメインに参加していないシナリオでは、以下のエラーは正常であり、続行を妨げるものではありません。



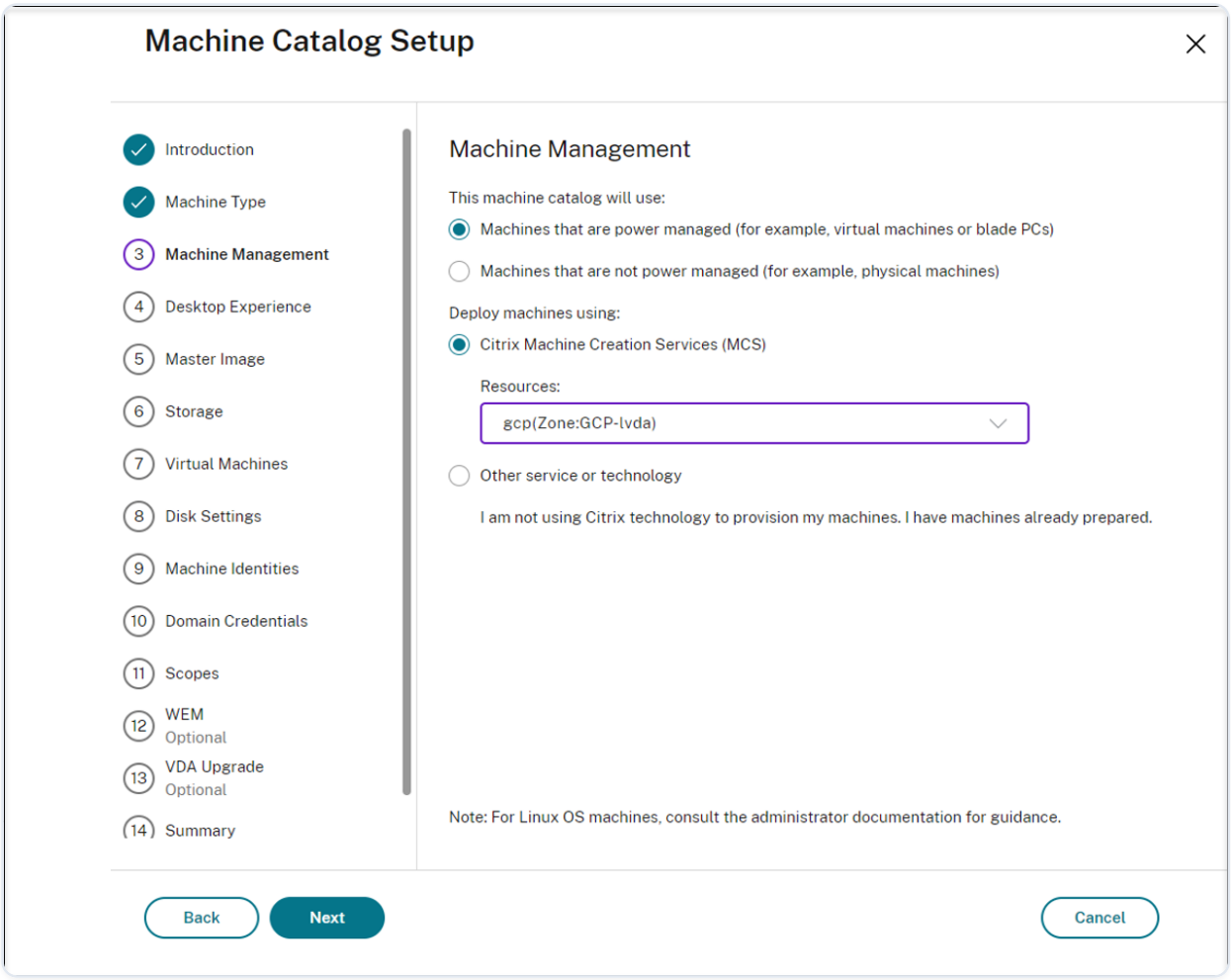
3. テンプレートVMにアプリケーションをインストールし、テンプレートVMをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。

ステップ 5: マシンカタログを作成する

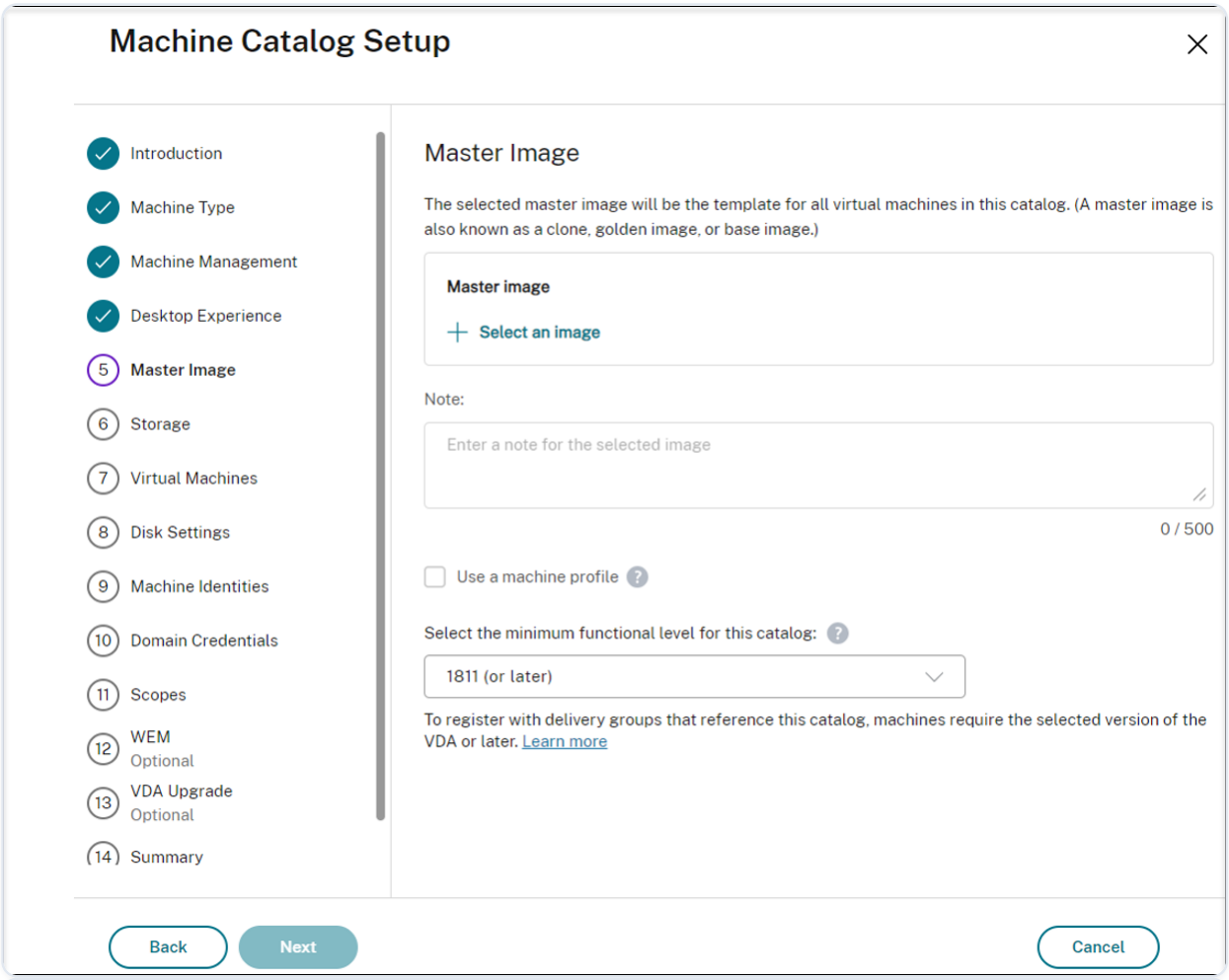
1. Citrix Cloudにサインインします。左上のメニューで、**My Services > DaaS**を選択します。
2. 「**管理 > 完全な構成**」から、左ペインで「**マシンカタログ**」を選択します。
3. アクションバーで を選択します。
4. 「**マシンタイプ**」ページで、「**マルチセッションOS**」または「**シングルセッションOS**」を選択し、「**次へ**」を選択します。

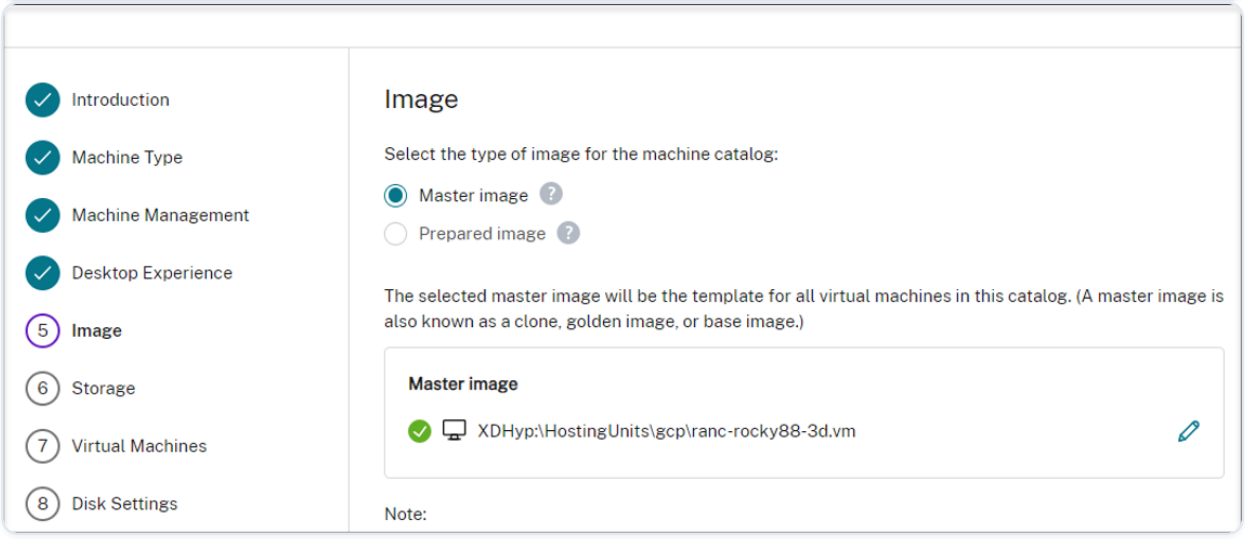


5. 「**マシン管理**」ページで、「**電源管理されているマシン**」と「**Citrix Machine Creation Services**」オプションを選択し、「**次へ**」を選択します。複数のリソースがある場合は、メニューから1つ選択します。



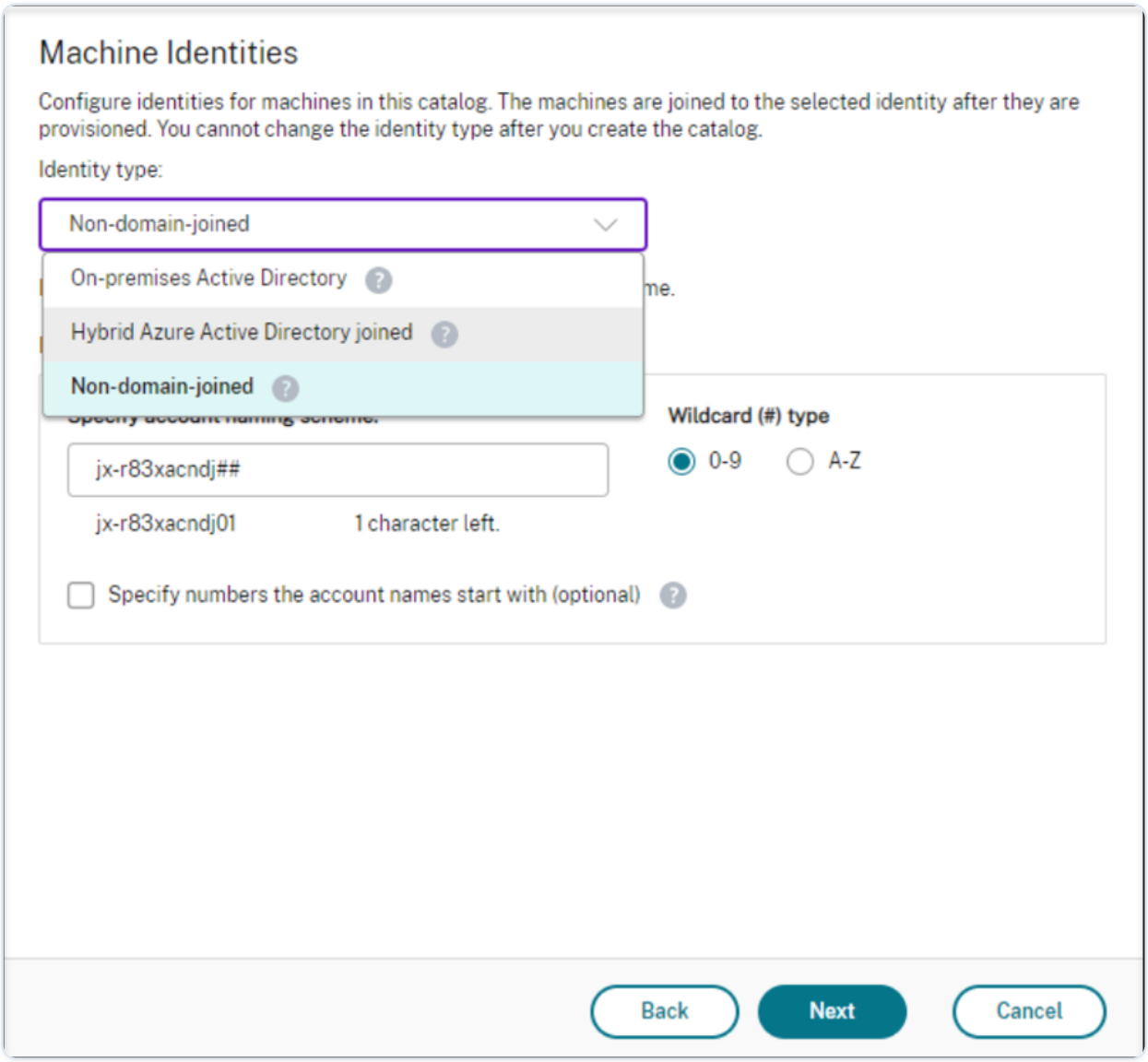
6. 「**マスターイメージ**」ページで、以前に作成したマスターイメージを選択します。



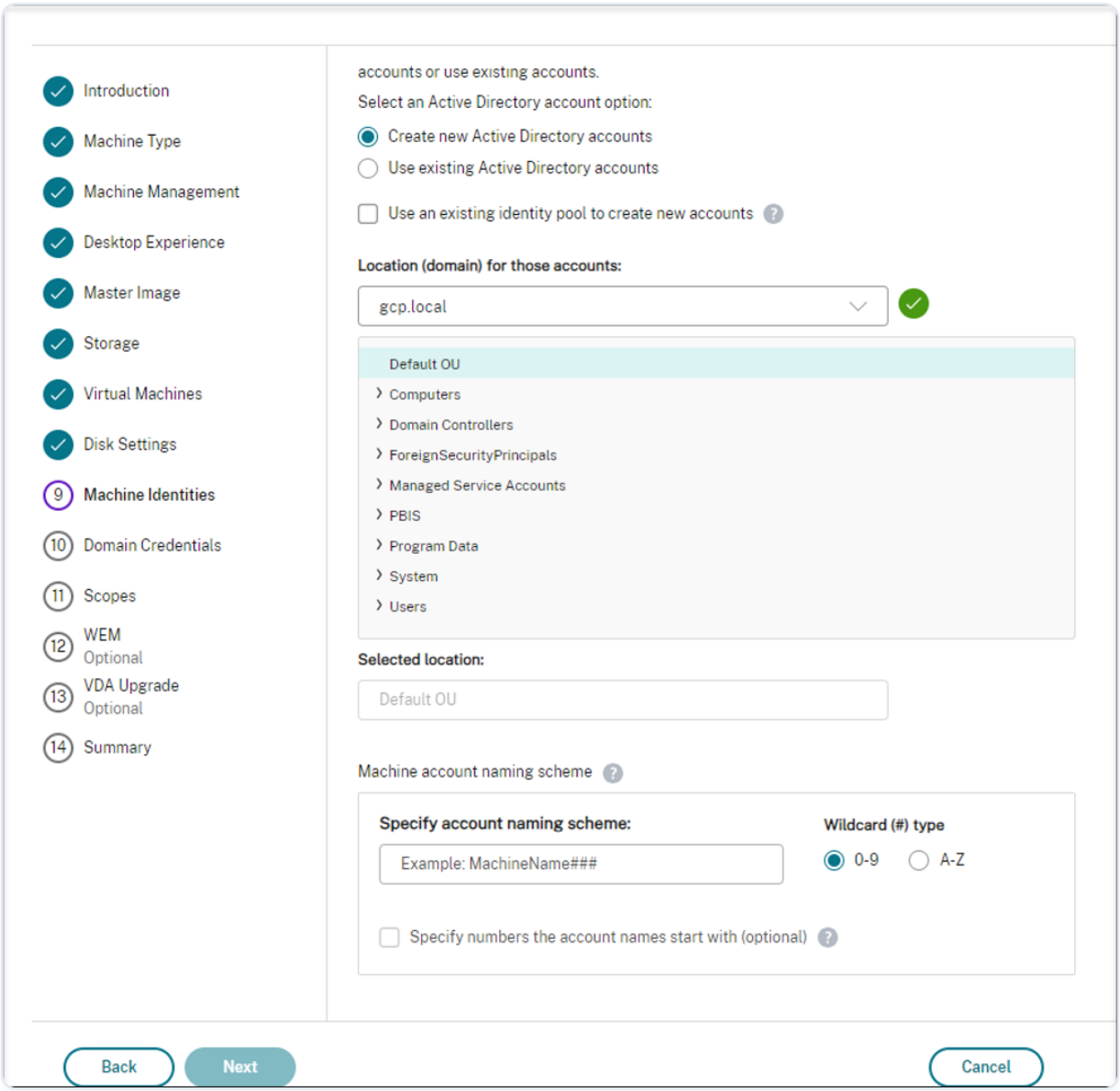


7. 「**マシンID**」ページで、マスターイメージがどのドメインにも参加していなかった場合は「**ドメインに参加していない**」を選択し、マスターイメージをドメインに参加させた場合はActive Directoryアカウントを選択します。

   ドメインに参加していないシナリオの場合:

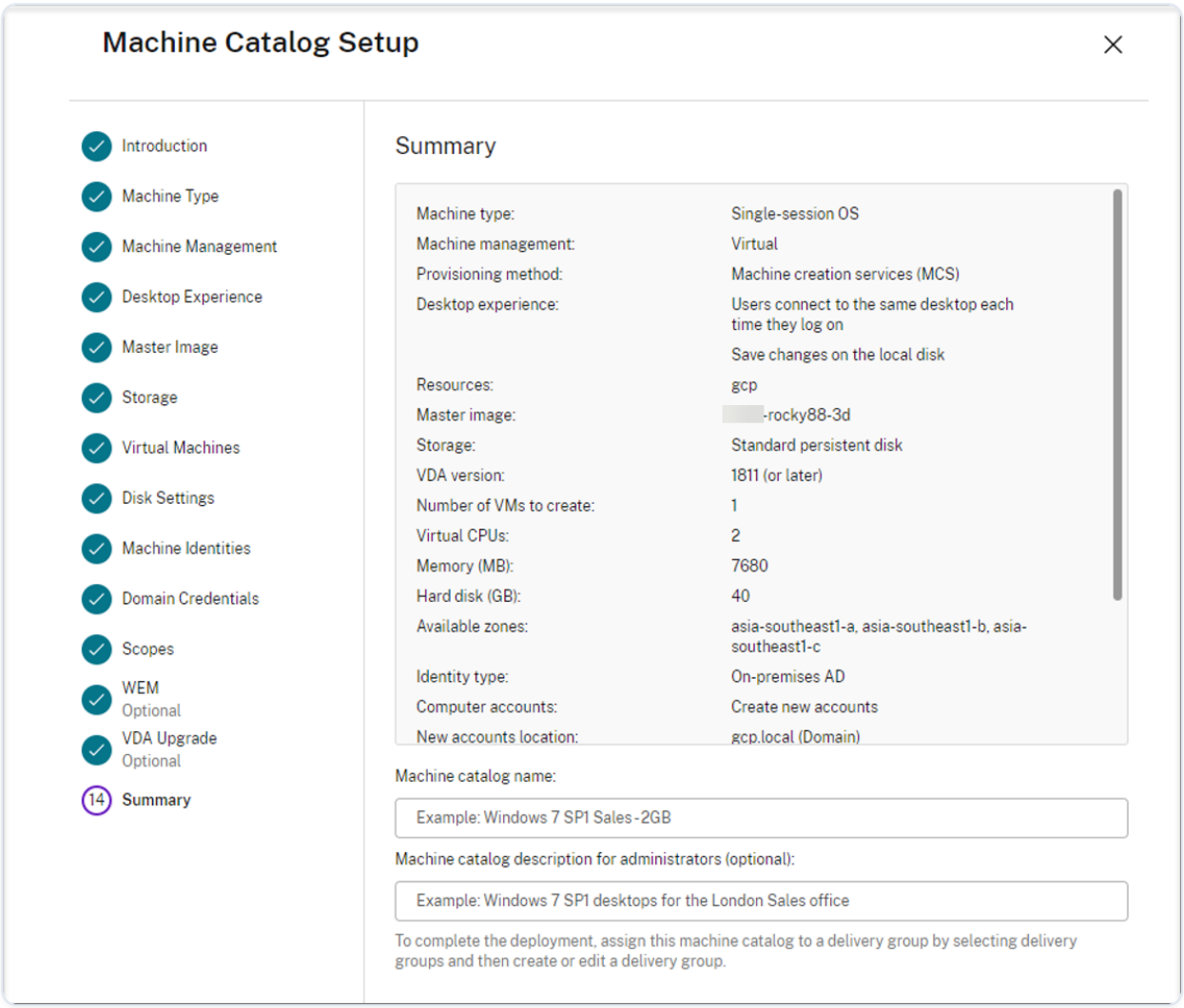


ドメインに参加しているシナリオの場合:



- **新しいActive Directoryアカウントを作成**を選択した場合は、ドメインを選択し、Active Directoryで作成されるプロビジョニングされたVMコンピューターアカウントの命名スキームを表す文字のシーケンスを入力します。アカウントの命名スキームは1～64文字で、空白スペース、非ASCII文字、または特殊文字を含めることはできません。
- **既存のActive Directoryアカウントを使用**を選択した場合は、**参照**を選択して、選択したマシン用の既存のActive Directoryコンピューターアカウントに移動します。
- 「**ドメイン資格情報**」ページで、「**資格情報を入力**」を選択し、ユーザー名とパスワードを入力して「**保存**」を選択し、「**次へ**」を選択します。入力する資格情報には、Active Directoryアカウント操作を実行する権限が必要です。

8. 他のページで追加設定を構成します。詳細については、Google Cloud Platformカタログの作成を参照してください。
9. **概要**ページで、情報を確認し、カタログの名前を指定して、**完了**を選択します。



マシンカタログの作成には時間がかかる場合があります。完了すると、カタログが一覧表示されます。Google Cloudコンソールで、マシンがターゲットノードグループに作成されていることを確認できます。

ステップ6：デリバリーグループを作成する

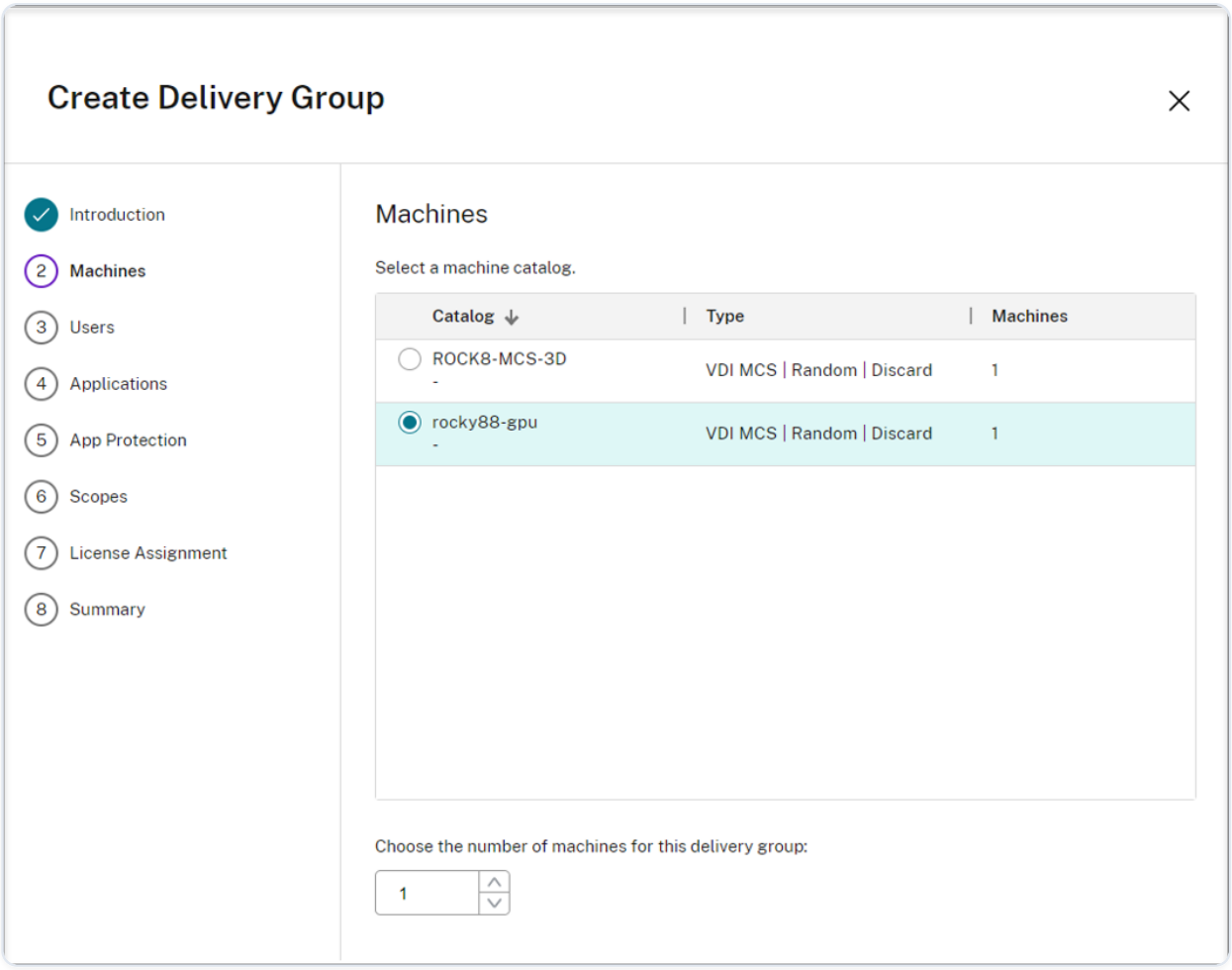
デリバリーグループは、1つ以上のマシンカタログから選択されたマシンのコレクションです。どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。

詳しくは、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］**の順に選択します。
2. **［管理］>［完全な構成］から、左側のペインで［デリバリーグループ］**を選択します。
3. アクションバーで**［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。

表示されるウィザードページは、選択内容によって異なる場合があります。

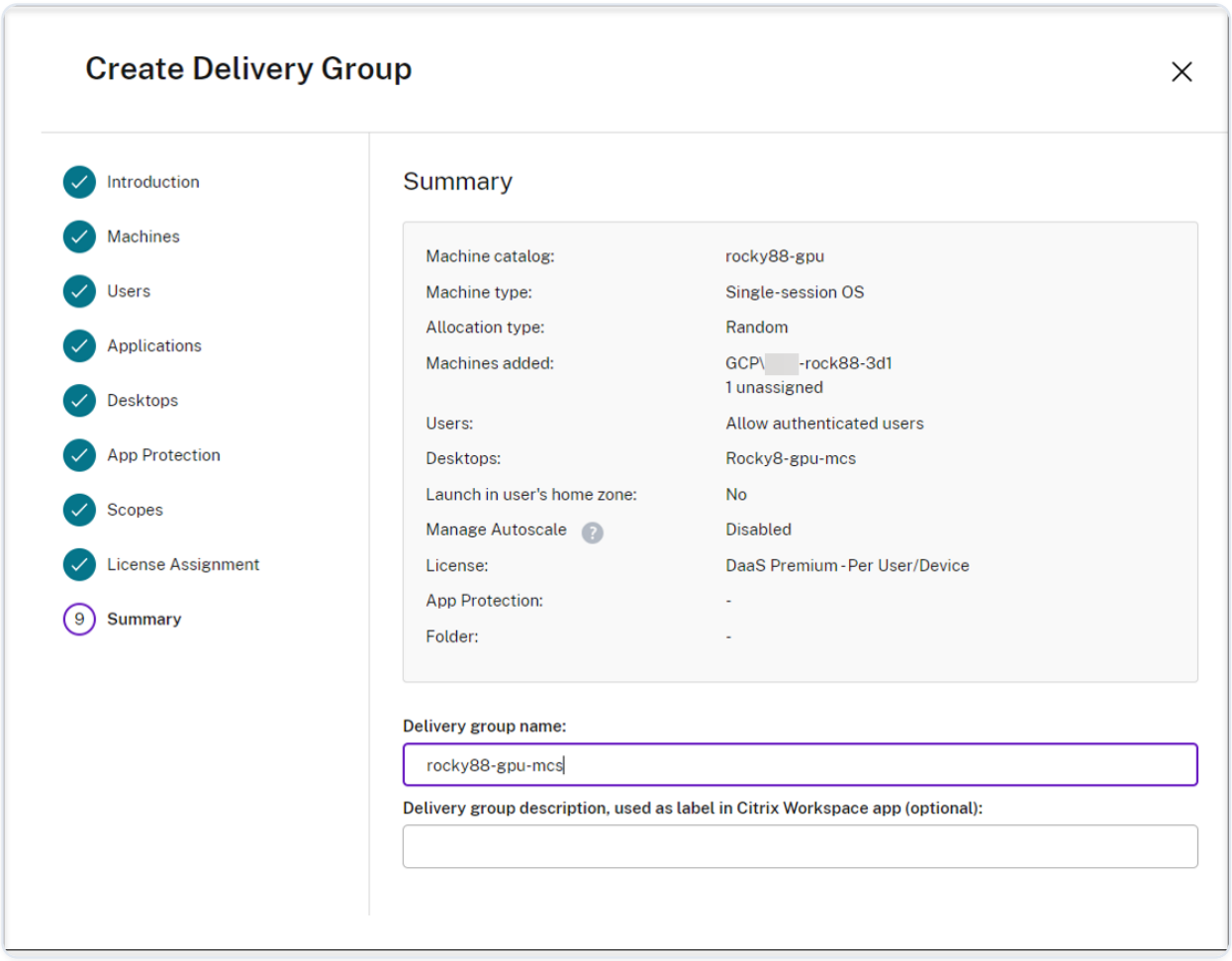
4. **［マシン］**ページで、マシンカタログを選択し、そのカタログから使用するマシンの数を選択します。



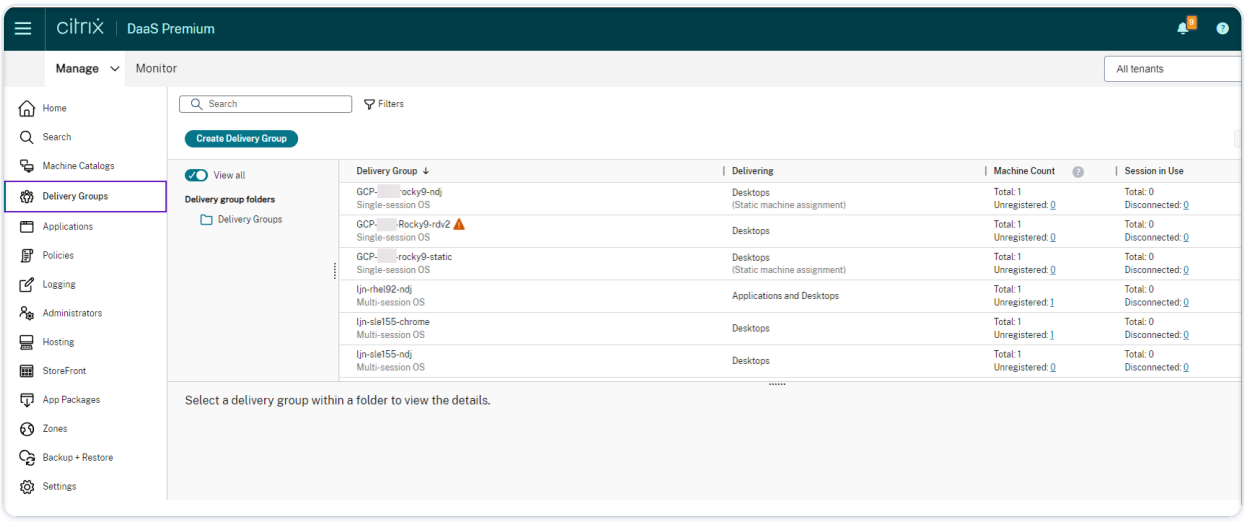
5. その他のページで追加設定を構成します。詳しくは、「デリバリーグループの作成」を参照してください。

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。また、Citrix Workspace™アプリおよび［完全な構成］**管理インターフェイスに表示される説明を（オプションで）入力することもできます。例：

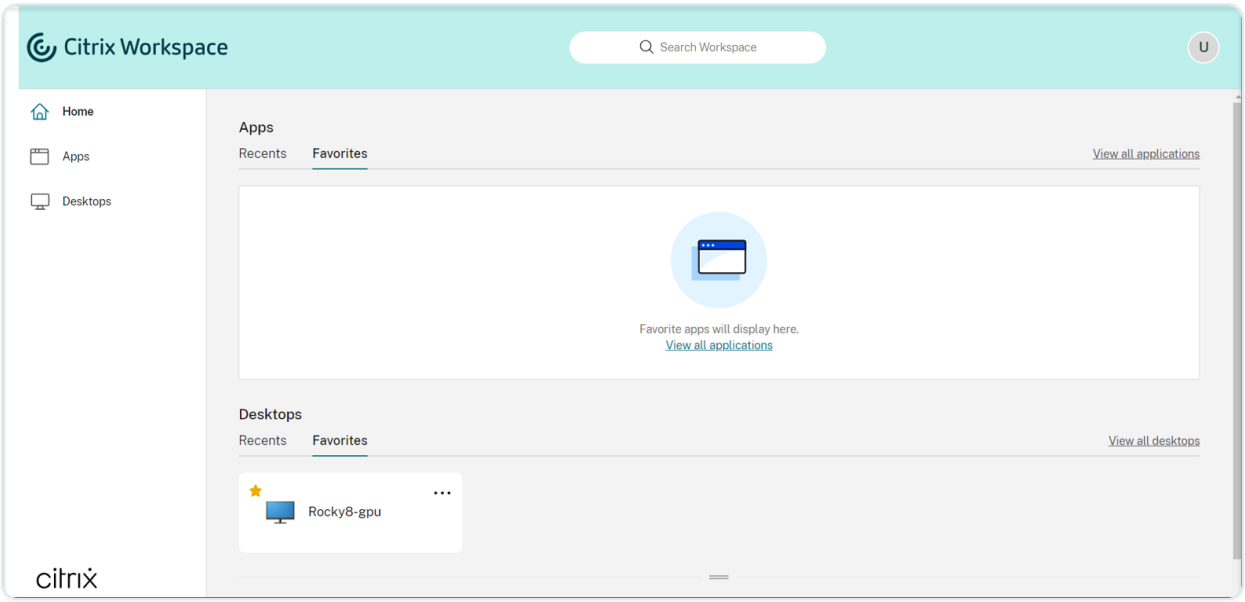
   デリバリーグループの名前を入力します。



** ［完全な構成］**管理インターフェイスのデリバリーグループのリスト：



Citrix Workspaceアプリで配信されたマシン



Ansible を使用した展開の管理

Ansible は、展開全体にわたるアプリケーション、構成、および更新の展開プロセスを自動化するのに役立ちます。この記事では、Ansible を使用して展開を効率的に管理するための段階的な手順を説明します。

ステップ 1: 展開するものを決定する

開始する前に、アプリケーション、サービス、構成、環境変数など、展開する必要があるものを特定します。

ステップ 2: Ansible プロジェクトをセットアップする

Ansible プロジェクトのディレクトリ構造を作成します。プレイブックのコンテンツを整理する重要な方法の 1 つは、Ansible の「ロール」機能です。詳細については、Ansible ドキュメントのロール を参照してください。

以下に、参考として 2 つのディレクトリ構造の例を示します。

ディレクトリ構造の例 #1

```
production                # inventory file for production servers

staging                   # inventory file for staging environment



group_vars/
```

```
   group1.yml             # here we assign variables to particular groups

   group2.yml

host_vars/

   hostname1.yml          # here we assign variables to particular systems

   hostname2.yml



library/                  # if any custom modules, put them here
(optional)

module_utils/             # if any custom module_utils to support modules,
put them here (optional)

filter_plugins/           # if any custom filter plugins, put them here
(optional)



site.yml                  # master playbook

webservers.yml            # playbook for webserver tier

dbservers.yml             # playbook for dbserver tier



roles/

    common/               # this hierarchy represents a "role"

        tasks/            #

            main.yml      #  <-- tasks file can include smaller files if
warranted

        handlers/         #

            main.yml      #  <-- handlers file
```

```
        templates/        #  <-- files for use with the template resource
            ntp.conf.j2   #  <------- templates end in .j2
        files/            #
            bar.txt       #  <-- files for use with the copy resource
            foo.sh        #  <-- script files for use with the script
resource
        vars/             #
            main.yml      #  <-- variables associated with this role
        defaults/         #
            main.yml      #  <-- default lower priority variables for this
role
        meta/             #
            main.yml      #  <-- role dependencies
        library/          # roles can also include custom modules
        module_utils/     # roles can also include custom module_utils
        lookup_plugins/   # or other types of plugins, like lookup in this
case


    webtier/              # same kind of structure as "common" was above,
done for the webtier role
    monitoring/           # ""
    fooapp/               # ""
```

ディレクトリ構造の例 #2

```
inventories/
   production/
      hosts               # inventory file for production servers
      group_vars/
         group1.yml       # here we assign variables to particular groups
         group2.yml
      host_vars/
         hostname1.yml    # here we assign variables to particular systems
         hostname2.yml

   staging/
      hosts               # inventory file for staging environment
      group_vars/
         group1.yml       # here we assign variables to particular groups
         group2.yml
      host_vars/
         stagehost1.yml   # here we assign variables to particular systems
         stagehost2.yml

library/
module_utils/
```

```
filter_plugins/



site.yml

webservers.yml

dbservers.yml



roles/

    common/

    webtier/

    monitoring/

    fooapp/
```

ステップ 3: インベントリを構成する

インベントリファイル (inventory.ini) を定義します。インベントリファイルには通常、Ansible を使用して管理するホストが、ホスト名、IP アドレス、グループメンバーシップなどの必要な詳細とともにリストされます。例:

```
# Hostname and ip address

[UBUNTU2004]

<ip address>

[UBUNTU2204]

<ip address>

[RHEL8]

<ip address>
```

```
[RHEL9]

<ip address>

[DEBIAN11]

<ip address>

[DEBIAN12]

<ip address>

[SUSE15]

<ip address>



[all:children]

UBUNTU2004

UBUNTU2204

RHEL8

RHEL9

DEBIAN11

DEBIAN12

SUSE15



[all:vars]

ansible_user=<ansible execute user e.g root>

ansible_password=<>
```

```
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

ステップ 4: Ansible プレイブックを作成する

展開タスクを自動化するためのプレイブック (.yml ファイル) を作成します。このセクションでは、さまざまな展開タスクを自動化するためのプレイブックの例を示します。

Linux ディストリビューションをパッチするためのプレイブックの例

Ansible を使用してさまざまな Linux ディストリビューションにパッチを適用するには、patch-for-different-distribution.yml という名前の YAML プレイブックファイルを作成し、次のタスクと同様の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。

```
- name: Upgrade and Reboot RHEL & Debian family Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    reboot_connect_timeout: 5

    reboot_post_reboot_delay: 15

    reboot_timeout: 600

  tasks:

    # Upgrade RHEL family OS packages

    - name: Upgrade RHEL Family OS packages

      ansible.builtin.yum:

        name: '*'

        state: latest

      when:

        - ansible_facts['distribution'] == "RedHat"
```

```
    - ansible_facts['distribution_major_version'] == "7"


# Upgrade RHEL family OS packages
- name: Upgrade RHEL Family OS packages
  ansible.builtin.yum:
    name: '*'
    state: latest
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "8"


# Upgrade RHEL family OS packages
- name: Upgrade RHEL Family OS packages
  ansible.builtin.yum:
    name: '*'
    state: latest
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "9"


# Ubuntu Family upgrade
- name: Update repositories cache
```

```
    apt:
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "20"


  - name: Update all packages to their latest version
    apt:
      name: "*"
      state: latest
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "22"


  # Debian Family upgrade
  - name: Upgrade the OS (apt-get dist-upgrade)
    apt:
      upgrade: dist
    when:
      - ansible_facts['distribution'] == "Debian"
      - ansible_facts['distribution_major_version'] == "11"
```

```
    - name: Upgrade the OS (apt-get dist-upgrade)
      apt:
        upgrade: dist
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "12"


    # Reboot after upgrade
    - name: Reboot host
      reboot:
        connect_timeout: "{{ reboot_connect_timeout }}"
        post_reboot_delay: "{{ reboot_post_reboot_delay }}"
        reboot_timeout: "{{ reboot_timeout }}"
```

.Net 環境をインストールするためのプレイブックの例

以下のプレイブックの例は、特定の Linux ディストリビューションに異なるバージョンの .Net 環境をインストールします。

```
- name: Install dotnet runtime environment on Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  tasks:

    # Install dotnet runtime environment on RHEL7

    - name: Enable the rhel-7-server-dotnet-rpms repository
```

```
      command: subscription-manager repos --enable=rhel-7-server-dotnet-
rpms

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"



    - name: Install dotnet runtime environment on RHEL7

      ansible.builtin.yum:

        name: rh-dotnet60-aspnetcore-runtime-6.0

        state: present

      when:

        - ansible_facts['distribution'] == "RedHat"

        - ansible_facts['distribution_major_version'] == "7"



    - name: Remove /usr/bin/dotnet if it exists

      file:

        path: /usr/bin/dotnet

        state: absent



    - name: Create a symbolic link

      file:

        src: /opt/rh/rh-dotnet60/root/usr/lib64/dotnet/dotnet
```

```
    dest: /usr/bin/dotnet
    state: link


# RHEL8 linux vda install dotnet runtime environment
- name: Install dotnet-runtime-8.0
  ansible.builtin.dnf:
    name: dotnet-runtime-8.0
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "8"


- name: Install aspnetcore-runtime-8.0
  ansible.builtin.dnf:
    name: aspnetcore-runtime-8.0
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "8"


# RHEL9 linux vda install dotnet runtime environment
- name: Install dotnet-runtime-8.0
```

```
      ansible.builtin.dnf:
        name: dotnet-runtime-8.0
        state: present
      when:
        - ansible_facts['distribution'] == "RedHat"
        - ansible_facts['distribution_major_version'] == "9"


    - name: Install aspnetcore-runtime-8.0
      ansible.builtin.dnf:
        name: aspnetcore-runtime-8.0
        state: present
      when:
        - ansible_facts['distribution'] == "RedHat"
        - ansible_facts['distribution_major_version'] == "9"


    # Ubuntu20.04 linux vda install dotnet runtime environment
    - name: Register Microsoft key and feed
      shell: |
        wget https://packages.microsoft.com/config/ubuntu/20.04/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb
        dpkg -i packages-microsoft-prod.deb
        rm packages-microsoft-prod.deb
```

```
    when:
      - ansible_facts['distribution'] == "Ubuntu"
      - ansible_facts['distribution_major_version'] == "20"

- name: Install dotnet-runtime-8.0
  ansible.builtin.apt:
    name: dotnet-runtime-8.0
    state: present
    update_cache: yes
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "20"

- name: Install aspnetcore-runtime-8.0
  ansible.builtin.apt:
    name: aspnetcore-runtime-8.0
    state: present
    update_cache: yes
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "20"
```

```
# Ubuntu22.04 linux vda install dotnet runtime environment
- name: Install dotnet-runtime-8.0
  ansible.builtin.apt:
    name: dotnet-runtime-8.0
    state: present
    update_cache: yes
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "22"


- name: Install aspnetcore-runtime-8.0
  ansible.builtin.apt:
    name: aspnetcore-runtime-8.0
    state: present
    update_cache: yes
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "22"


# Debian11 linux vda install dotnet runtime environment
- name: Register Microsoft key and feed
  shell: |
```

```
        wget https://packages.microsoft.com/config/debian/11/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb

        dpkg -i packages-microsoft-prod.deb

        rm packages-microsoft-prod.deb

      when:

        - ansible_facts['distribution'] == "Debian"

        - ansible_facts['distribution_major_version'] == "11"



    - name: Install dotnet-runtime-8.0

      ansible.builtin.apt:

        name: dotnet-runtime-8.0

        state: present

        update_cache: yes

      when:

        - ansible_facts['distribution'] == "Debian"

        - ansible_facts['distribution_major_version'] == "11"



    - name: Install aspnetcore-runtime-8.0

      ansible.builtin.apt:

        name: aspnetcore-runtime-8.0

        state: present

        update_cache: yes
```

```
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "11"


    # Debian12 linux vda install dotnet runtime environment
    - name: Register Microsoft key and feed
      shell: |
        wget https://packages.microsoft.com/config/debian/12/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb
        dpkg -i packages-microsoft-prod.deb
        rm packages-microsoft-prod.deb
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "12"


    - name: Install dotnet-runtime-8.0
      ansible.builtin.apt:
        name: dotnet-runtime-8.0
        state: present
        update_cache: yes
      when:
        - ansible_facts['distribution'] == "Debian"
```

```
        - ansible_facts['distribution_major_version'] == "12"


    - name: Install aspnetcore-runtime-8.0
      ansible.builtin.apt:
        name: aspnetcore-runtime-8.0
        state: present
        update_cache: yes
      when:
        - ansible_facts['distribution'] == "Debian"
        - ansible_facts['distribution_major_version'] == "12"


    # Sles15 linux vda install dotnet runtime environment
    - name: Register Microsoft key and feed
      shell: |
        sudo rpm -Uvh https://packages.microsoft.com/config/sles/15/
packages-microsoft-prod.rpm
        sudo ln -s /etc/yum.repos.d/microsoft-prod.repo /etc/zypp/repos.d/
microsoft-prod.repo
      when:
        - ansible_facts['distribution'] == "SLES"
        - ansible_facts['distribution_major_version'] == "15"


    - name: Install dotnet-runtime-8.0
```

```
    community.general.zypper:
      name: dotnet-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "SLES"
      - ansible_facts['distribution_major_version'] == "15"


  - name: Install aspnetcore-runtime-8.0
    community.general.zypper:
      name: aspnetcore-runtime-8.0
      state: present
      update_cache: yes
    when:
      - ansible_facts['distribution'] == "SLES"
      - ansible_facts['distribution_major_version'] == "15"


  # Amazon2 linux vda install dotnet runtime environment
  - name: Install dotnet-runtime-8.0
    ansible.builtin.yum:
      name: dotnet-runtime-8.0
      state: present
```

```
      when:
        - ansible_facts['distribution'] == "Amazon"
        - ansible_facts['distribution_major_version'] == "2"


    - name: Install aspnetcore-runtime-8.0
      ansible.builtin.yum:
        name: aspnetcore-runtime-8.0
        state: present
      when:
        - ansible_facts['distribution'] == "Amazon"
        - ansible_facts['distribution_major_version'] == "2"
```

Linux VDA をアップグレードするためのプレイブックの例

Ansible を使用して Linux VDA のアップグレードを自動化するには、2 つの個別のプレイブックを作成できます。get_the_build.yml のような 1 つのプレイブックは、Linux VDA パッケージをターゲットマシン (ホスト) にダウンロードして転送することに特化しています。もう 1 つのプレイブック (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。

プレイブックの例 get_the_build.yml

```
- hosts: localhost
  name: Get the latest release build to local
  vars:
    build_url: <linux vda download link>  # replace with your actual
value.
    local_tmp: "/tmp/"  # replace with your actual value.
```

```
    remote_tmp: "/tmp/"  # replace with your actual value.

    linuxvda_file_name : "linux vda rpm/deb file name" # replace with your
actual value.

  tasks:

  - name: Download the file

    get_url:

      url: "{{ build_url }}"

      dest: "{{ local_tmp }}"

    tags:

      - get




- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  name: Copy a file to remote location

  tasks:

  - name: Copy vda to the remote machine

    ansible.builtin.copy:

      src: "{{ local_tmp }}{{ linuxvda_file_name }}"

      dest: "{{ remote_tmp }}"

      remote_src: no

    tags:
```

```
      - copy
```

プレイブックの例 linux_upgrade.yml

```
- name: Upgrade Linux VDA and Reboot RHEL & Debian Linux distros

  hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    remote_tmp: "/path/to/remote/tmp"  # replace with your actual path

    rhel7_file_name: "rhel7_file.rpm"  # replace with your actual file
name

    rhel8_file_name: "rhel8_file.rpm"  # replace with your actual file
name

    rhel9_file_name: "rhel9_file.rpm"  # replace with your actual file
name

    ubuntu2004_file_name: "ubuntu2004_file.deb"  # replace with your
actual file name

    ubuntu2204_file_name: "ubuntu2204_file.deb"  # replace with your
actual file name

    debian11_file_name: "debian11_file.deb" # replace with your actual
file name

    debian12_file_name: "debian12_file.deb" # replace with your actual
file name

    suse15_file_name: "suse15_file.deb" # replace with your actual file
name

    amazon2_file_name: "amazon2_file.rpm" # replace with your actual file
name

  tasks:
```

```
# Upgrade RHEL linux vda packages
- name: Upgrade RHEL7 linux vda packages
  ansible.builtin.yum:
    name: "{{ remote_tmp }}{{ rhel7_file_name }}"
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "7"


# Upgrade RHEL linux vda packages
- name: Upgrade RHEL8 linux vda packages
  ansible.builtin.yum:
    name: "{{ remote_tmp }}{{ rhel8_file_name }}"
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "8"


# Upgrade RHEL linux vda packages
- name: Upgrade RHEL9 linux vda packages
  ansible.builtin.yum:
    name: "{{ remote_tmp }}{{ rhel9_file_name }}"
```

```
    state: present
  when:
    - ansible_facts['distribution'] == "RedHat"
    - ansible_facts['distribution_major_version'] == "9"


# Ubuntu20.04 linux vda upgrade
- name: Ubuntu20.04 linux vda upgrade
  ansible.builtin.apt:
    deb: "{{ remote_tmp }}{{ ubuntu2004_file_name }}"
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "20"


- name: Ubuntu22.04 linux vda upgrade
  ansible.builtin.apt:
    deb: "{{ remote_tmp }}{{ ubuntu2204_file_name }}"
  when:
    - ansible_facts['distribution'] == "Ubuntu"
    - ansible_facts['distribution_major_version'] == "22"


# Debian Linux VDA upgrade
- name: Debian11 Linux VDA upgrade
```

```
    ansible.builtin.apt:
      deb: "{{ remote_tmp }}{{ debian11_file_name }}"
    when:
      - ansible_facts['distribution'] == "Debian"
      - ansible_facts['distribution_major_version'] == "11"


  - name: Debian12 Linux VDA upgrade
    ansible.builtin.apt:
      deb: "{{ remote_tmp }}{{ debian12_file_name }}"
    when:
      - ansible_facts['distribution'] == "Debian"
      - ansible_facts['distribution_major_version'] == "12"


  # Sles15 Linux VDA upgrade
  - name: Sles15 Linux VDA upgrade
    community.general.zypper:
      name: "{{ remote_tmp }}{{ suse15_file_name }}"
      state: present
    when:
      - ansible_facts['distribution'] == "SLES"
      - ansible_facts['distribution_major_version'] == "15"
```

```
    # Amazon2 Linux VDA upgrade

    - name: Amazon2 Linux VDA upgrade

      ansible.builtin.yum:

        name: "{{ remote_tmp }}{{ amazon2_file_name }}"

      when:

        - ansible_facts['distribution'] == "Amazon"

        - ansible_facts['distribution_major_version'] == "2"

    # Reboot after upgrade

    - name: Reboot host

      reboot:

        connect_timeout: "{{ reboot_connect_timeout }}"

        post_reboot_delay: "{{ reboot_post_reboot_delay }}"

        reboot_timeout: "{{ reboot_timeout }}"
```

ネットワークファイルシステム (NFS) サーバーをホームディレクトリとしてマウントするためのプレイブックの例

以下のプレイブックの例は、ターゲットホスト上の NFS サーバーをホームディレクトリとしてマウントします。

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    nfs_server = <nfsserver ip address> # replace with your actual values

    mount_points = /home/<domain realm>/user1,/home/<domain realm>user2  #
replace with your actual values
```

```
    nfs_shares = user1,user2  # replace with your actual values

    owners = user1,user2   # replace with your actual values

    groups = group1,group2  # replace with your actual values

  tasks:

    - name: Enable NFS as home directory

      ansible.builtin.command:

        cmd: "/opt/Citrix/VDA/bin/ctxreg create -k 'HKLM\\System\
\CurrentControlSet\\Control\\Citrix' -t 'REG_DWORD' -v
'CheckUserHomeMountPoint' -d '0x00000001' --force"

      register: result

      failed_when: result.rc != 0

      check_mode: no



    - name: Mount NFS shares

      ansible.builtin.mount:

        path: "{{ item.0 }}"

        src: "{{ nfs_server }}:{{ item.1 }}"

        fstype: nfs

        opts: rw,nolock

        state: mounted

      loop: "{{ mount_points | zip(nfs_shares) | list }}"



    - name: Set owner, group and mode for NFS client paths
```

```
      ansible.builtin.file:

        path: "{{ item.0 }}"

        owner: "{{ item.1 }}"

        group: "{{ item.2 }}"

        mode: "{{ mode }}"

      loop: "{{ mount_points | zip(owners, groups) | list }}"
```

リモートコマンド実行のためのプレイブックの例

レジストリ設定を変更するためのプレイブックの例

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    registry_key: "your_registry_key"      #  E.g. registry_key =
HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd

    registry_type: "your_registry_type"    #  E.g. registry_type =
REG_DWORD

    registry_value: "your_registry_value"  #  E.g. registry_value =
AdaptiveScalingEnabled

    registry_data: "your_registry_data"    #  E.g. registry_data =
0x00000000

  tasks:

  - name: Execute AdaptiveScaling redirection script

    ansible.builtin.command:

      cmd: "/opt/Citrix/VDA/bin/ctxreg create -k \"{{ registry_key }}\" -t
\"{{ registry_type }}\" -v \"{{ registry_value }}\" -d
\"{{ registry_data }}\" --force"
```

```
    register: result

    failed_when: result.rc != 0

    check_mode: no
```

RHEL マイナーバージョンをロックするためのプレイブックの例

```
- hosts: <host1,host2,host3>  # replace with your actual hosts in the
inventory file.

  vars:

    rhel_minor_version: "9.3"  # replace with your actual minor version
such as 9.3, 8.8

  tasks:

    - name: Lock system to a specific minor version

      ansible.builtin.command:

        cmd: "subscription-manager release --set={{ rhel_minor_version }}"

      register: result

      failed_when: "'Error' in result.stderr"
```

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 2411のドキュメントに対する重要な変更点を示します。 これらの更新に関する通知を受け取るには、RSSフィードを購読してください。

日付	変更	説明
18 Feb 2025	既知の問題を追加	詳しくは、「解決された問題」を参照してください。
27 Dec 2024	記事を追加	詳しくは、「WebSocketの自己署名証明書の構成」を参照してください。
03 Dec 2024	Linux Virtual Delivery Agent 2411の初期リリース	詳しくは、「新機能」を参照してください。