Linux バーチャル デリバリー エージェント 2301

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あらゆるデバイスからどこからでも、Linux仮想アプリとデスクトップへのアクセスを可能にします。

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

新機能

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（実験的）

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

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

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

動的クライアントドライブマッピングとクライアントフォルダーリダイレクト

以前は、セッション開始後にクライアントに接続されたドライブはセッションにマップされませんでした。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。詳しくは、「クライアントドライブマッピング」を参照してください。

以前のリリースでの新機能

1912 LTSR 以降 2212 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 2212以降、以下の問題が修正されました。

- Linux VDAセッションを開始しようとすると、グレースクリーンになり、最終的にセッションが切断されることがあります。
- マシンアカウントのパスワードリセットが失敗すると、VDAがドメイン接続を失うことがあります。
- VDAが登録状態であっても、Linux VDAセッションを開始しようとすると失敗することがあります。
- ユーザーセッションからログオフするのに長い時間 (最大60秒) がかかることがあります。

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。

- HDX™ 3D Pro を有効にすると、拡張モニター上のセッションがブラックアウトし、プライマリモニターのみがセッションを適切に表示します。この問題を解決するには、VDA でターミナルを開き、必要に応じて次のコマンドを実行します。
  - デュアルモニターの場合は、次を実行します。

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

  - トリプルモニターの場合は、次を実行します。

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

  - クアッドモニターの場合は、次を実行します。

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

- **/var/log/xdl/jproxy.log** でスローされた以下の LDAP 例外により、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、次の操作を行います。

  - LDAP タイムアウト値を変更します。たとえば、次のコマンドを使用して LDAP タイムアウト値を 60 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、Knowledge Center の記事 CTX474888 を参照してください。
- Kerberos で **RC4_HMAC_MD5** 暗号化タイプが許可されている場合、Linux VDA は Controller への登録に失敗し、次のエラーメッセージが表示されることがあります。

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

この問題に対処するには、Active Directory ドメイン全体で (**RC4_HMAC_MD5** を無効にするか、*または特定の OU で*)、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 ドメイン全体で **RC4_HMAC_MD5** を無効にするには、次の手順を実行します。

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

Linux VDAで脆弱な暗号化タイプを許可するには、次の手順を実行します。

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

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

```
allow_weak_crypto= TRUE
```

- Linux VDAは、暗号化にSecureICA 1.0をサポートしていません。Linux VDAでSecureICA 1.0を有効にすると、セッション起動が失敗します。

- シームレスではない公開アプリケーションが起動直後に終了する場合があります。この問題は、Mutterがmutter-3.28.3-4より新しいバージョンにアップグレードされた後に発生します。この問題を回避するには、mutter-3.28.3-4以前を使用してください。
- ファイルダウンロード中に予期しないウィンドウが表示されます。このウィンドウはファイルダウンロード機能には影響せず、しばらくすると自動的に消えます。
- PulseAudioのデフォルト設定では、20秒間アイドル状態が続くとサウンドサーバープログラムが終了します。PulseAudioが終了すると、オーディオが機能しなくなります。この問題を回避するには、/etc/pulse/daemon.confファイルでexit-idle-time=-1を設定します。
- SSL暗号化が有効でセッションの信頼性が無効になっている場合、Linux版Citrix Workspace™アプリでセッションを起動できません。
- Ubuntuグラフィック：HDX 3D Proで、Desktop Viewerのサイズ変更後にアプリケーションの周囲に黒いフレームが表示されたり、背景が黒く表示されたりする場合があります。
- Linux VDAの印刷リダイレクトによって作成されたプリンターは、セッションからログアウトした後も削除されない場合があります。
- CDMファイルは、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に不足します。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 状態は、セッションローミング中に反転する可能性があります。既存の接続を Citrix Workspace app for Android にローミングすると、CAPS LOCK 状態が失われることがあります。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。
- Citrix Workspace app for Mac を使用して Linux VDA に接続する場合、ALT を含むショートカットキーが常に機能するとは限りません。Citrix Workspace app for Mac は、デフォルトで左右の Options/Alt キーの両方に対して AltGr を送信します。この動作は Citrix Workspace app の設定内で変更できますが、結果はアプリケーションによって異なります。
- Linux VDA がドメインに再参加すると、登録が失敗します。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に接続しようとすると、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、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 および潜在的に他のサービスのサービスチケットが削除されます。ただし、これは無害です。これらのサービスチケットは、必要に応じて KDC から再取得できます。

- オーディオのプラグアンドプレイはサポートされていません。ICA®セッションでオーディオの録音を開始する前に、オーディオキャプチャデバイスをクライアントマシンに接続できます。オーディオ録音アプリケーションの開始後にキャプチャデバイスが接続された場合、アプリケーションが応答しなくなり、再起動する必要があります。録音中にキャプチャデバイスが取り外された場合、同様の問題が発生する可能性があります。
- Citrix Workspace app for Windows では、オーディオ録音中に音声の歪みが発生する可能性があります。

サードパーティ通知

Linux Virtual Delivery Agent バージョン 2301 (PDFダウンロード)

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

廃止予定

この記事のお知らせは、段階的に廃止されるプラットフォーム、Citrix®製品、および機能について事前に通知するものであり、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します。Citrixは、お客様の使用状況とフィードバックを監視し、いつそれらが廃止されるかを決定します。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、すぐに削除されるわけではありません。Citrixは、このリリースでは引き続きサ

ポートしますが、将来のCurrent Releaseで削除されます。 **削除済み**の項目は、Linux VDAで削除されたか、またはサポートされなくなりました。

項目	非推奨が発表されたバージョン	削除時期
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バージョン	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2212 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。

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

RHEL 9.0およびRocky Linux 9.0をサポート対象のディストリビューションとして追加しました。詳しくは、「システム要件」を参照してください。

USBデバイスリダイレクトの機能強化

USBデバイスリダイレクトに関して、以下の機能強化を行いました。

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

詳しくは、「USBデバイスリダイレクト」を参照してください。

NVIDIA製特定の非仮想化GPUのハードウェアアクセラレーション

Linux VDA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

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

詳しくは、「非仮想化GPU」を参照してください。

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

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

MCS構成を含むように簡易インストールGUIを拡張

以前は、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変数を構成できるようになりました。GUIで**Deploy**をクリックすると、GUIで設定した変数が`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。詳しくは、「手順3：マスターイメージの準備」を参照してください。

Linuxセッションの記録のサポート（実験的）

実験的な機能として、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、「セッション記録（実験的）」を参照してください。

データベースオプションが利用可能になりました（実験的）

実験的な機能として、PostgreSQLに加えてSQLiteを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、SQLiteとPostgreSQLを切り替えることもできます。これを行うには、次の手順を完了します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略してください。

2. 使用するデータベースを指定するために`/etc/xdl/db.conf`を編集します。
3. インストール方法に基づいて、次のいずれかのスクリプトを実行します。

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

注:

- VDIモードでのみSQLiteを使用することをお勧めします。
- 手動インストールの場合、SQLiteとPostgreSQLを切り替える前に、手動でインストールする必要があります。簡易インストールおよびMCSの場合、手動でインストールすることなくSQLiteとPostgreSQLを切り替えることができます。
- `/etc/xdl/db.conf`を使用してPostgreSQLのポート番号を構成することもできます。

詳しくは、インストールに関する記事を参照してください。

2210の新機能

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

HDX™ 3D Pro向けGPUハードウェアアクセラレーションの強化

GPUとLinuxシステムメモリ間のデータ転送効率を向上させました。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ドにおける遅延を削減しました。これらの機能強化により、ハードウェアリソースの利用が最適化され、フレーム/秒（FPS）のパフォーマンスが大幅に向上します。詳しくは、H.264ハードウェアエンコードを参照してください。

クリップボードを介したデータ転送のサイズ制限のサポート

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

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

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

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

新しいLinuxストリーミングターゲットデバイスのサポート

Linuxストリーミングを以下のディストリビューションに拡張しました。

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

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

シェルスクリプトの機能強化

シェルスクリプトをメンテナンスしやすいように機能強化し、次の構成テンプレートを`/etc/xdl/mcs/`から`/etc/xdl/ad_join/`に移動しました。

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

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

2209 の新機能

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

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

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

SSSD および PBIS 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

Linux VDA パッケージは GPG を使用して署名されているため、公開キーを使用して Linux VDA パッケージの整合性を検証できます。

公開キーを取得するには：

1. Citrix Virtual Apps and Desktops ダウンロードページ にアクセスします。
2. 適切なバージョンの Citrix Virtual Apps and Desktops を展開します。
3. ［**コンポーネント**］ をクリックして、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、以下の新機能と機能強化が含まれています。

非SSOシナリオでのより多くのユーザー認証方法のサポート

以前は、シングルサインオン（SSO）が無効になっている場合、ユーザーは異なるユーザー名を使用してCitrix Workspaceアプリおよび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。詳細については、「非SSO認証」を参照してください。

セッションウォーターマークを構成するための追加オプション

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

KVMハイパーバイザーのサポート

Kernel-based Virtual Machine (KVM) は、Linux向けの主要なオープンソース仮想化テクノロジーです。これはLinuxの一部であり、Linuxマシンをハイパーバイザーに変えて、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ハイパーバイザー上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。

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

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

2206の新機能

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

Debian 11.3のサポート

Debian 11.3 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。

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。詳細については、「Azure Active Directory を使用した認証」を参照してください。

クライアントのバッテリー状態表示

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

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

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

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

詳細については、「非vGPUグラフィックカード」を参照してください。

セッションログオン画面のカスタム背景とバナーメッセージ

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

自動DPIスケーリング

Linux VDA は自動DPIスケーリングをサポートするようになりました。ユーザーが仮想デスクトップまたはアプリケーションセッションを開くと、セッション内のDPI値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。このリリース以降、ctxsetup.shのCTX_XDL_FAS_LISTを介して、Linux VDA側でFASポートをカスタマイズできます。詳しくは、お使いのディストリビューションに基づいたLinux VDAインストール記事を参照してください。

2204の新機能

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

ハードウェアエンコードが選択的コーデックに拡張

以前は画面全体でのみ利用可能だったH.264ハードウェアエンコードは、アクティブに変化する領域での選択的な使用をサポートするようになりました。この機能により、CPUのビデオ圧縮消費がハードウェアにオフロードされ、画質と1秒あたりのフレーム数（FPS）が向上します。この機能を有効にする方法については、「グラフィックの構成と微調整」を参照してください。

一時ホームディレクトリでのログオンのサポート

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

Rendezvous HDXトラフィックのSOCKS5プロキシサポート

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

Rendezvousの透過型プロキシサポート

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

GNOME Classicデスクトップのサポート

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

アプリのタスクバーボタンのグループ化

以前は、同じセッションで実行されている公開アプリケーションのすべてのタスクバーボタンが1つのグループにまとめられていました。このリリースから、アプリの複数のウィンドウが開いている場合でも、各アプリは単一のタスクバーボタンとして表示されます。

2203の新機能

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

Rendezvous V2の完全サポート

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

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

Rendezvous V1ではHDXトラフィックがCloud Connectorをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックをプロキシするには、依然としてCloud Connectorが必要です。

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

詳しくは、「Rendezvous V2」を参照してください。

簡単インストールGUIが利用可能になりました

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

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

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

HDXグラフィックの改善

Thinwireは、特定のタスクを並列化することで、フレーム/秒 (FPS) の数を向上させることができますが、その分、CPU全体の消費量がわずかに増加します。この機能はデフォルトで無効になっています。詳しくは、グラフィックの構成の記事の並列処理セクションを参照してください。

2201の新機能

Linux VDA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。デスクトップ環境について詳しくは、インストール記事を参照してください。

セッションシャドウイング機能がサポートされているすべてのディストリビューションに拡張

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

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アプリへのログオンに使用したユーザー名に基づいてローカルユーザーを作成します。ユーザーのユーザー識別子

(UID)、グループID (GID)、ホームディレクトリ、ログインシェルなどのユーザー属性を指定できるようになりました。詳しくは、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。詳しくは、TLSを使用したユーザーセッションの保護を参照してください。

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

2112の新機能

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

Amazon Linux 2のサポート（実験的）

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

HDX画面共有の完全サポート

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

HTTPプロキシ経由の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プロキシ経由でのRendezvous接続の確立をサポートするようになりました。詳しくは、「Rendezvousプロトコル」を参照してください。

Rendezvousプロキシ認証

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

アダプティブオーディオのサポート

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

新しいLinuxストリーミングターゲットデバイスのサポート

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

- SUSE リナックス エンタープライズ サーバー 15 SP2 (SUSE 15.2)
- SUSE リナックス エンタープライズ サーバー 15 SP3 (SUSE 15.3)

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

キーボードレイアウト同期の機能強化

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

キーボード入力の機能強化

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

2110の新機能

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

SUSE 15.3およびSUSE 15.2のサポート

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

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

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

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

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

ctxfascfg.shに2つの環境変数が追加されました

ctxfascfg.shに以下の変数を追加しました。

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

詳しくは、「ctxfascfg.shを実行する」を参照してください。

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

Citrix Workspaceアプリにログオンした後、異なる資格情報を使用してCitrix Virtual Apps™ 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、セッション起動時に資格情報の入力を求められます。この機能は、次のディストリビューションでサポートされています。

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

2109の新機能

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

HDX画面共有のサポート（実験的）

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

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

Linux VDA用Citrix Virtual Channel SDKが利用可能になりました。Virtual Channel SDKを使用すると、VDA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 用の Citrix Virtual Channel SDK は、Citrix Virtual Apps and Desktops ダウンロードページ からダウンロードできます。適切なバージョンの Citrix Virtual Apps and Desktops を展開し、**Components** をクリックして Linux VDA のダウンロードを選択します。

PBIS を使用する VDA の FAS サポート

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

MATE デスクトップのキーボードレイアウト同期サポート

キーボードレイアウト同期サポートを MATE デスクトップに拡張しました。クライアントデバイスのキーボードレイアウトが変更されるたびに、MATE デスクトップ環境がインストールされている VDA のレイアウトもそれに合わせて変更されます。詳しくは、「動的キーボードレイアウト同期」を参照してください。

セッションデータクエリユーティリティ

各 Linux VDA のセッションデータをクエリするために使用できるユーティリティ (`ctxsdcutil`) を提供するようになりました。VDA でホストされているすべてのセッションまたは特定のセッ

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

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

新しいLinuxセッションメトリックが利用可能になりました

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

- ログオン期間

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

- セッションの自動再接続回数

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

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

2107の新機能

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

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

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

USBリダイレクトのカーネルモジュール再構築の簡素化

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

RHEL 8.4およびCentOS 8 (2105) のサポート

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

Debianでのスマートカードのサポート

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

2106の新機能

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

Linux VDAのセルフアップデート

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

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

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

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

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

HDX Webカメラビデオ圧縮のサポート

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

グラフィック品質スライダー

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

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

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

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

    - クライアントプリンターリダイレクト
    - PDFユニバーサルプリンターの自動作成
- ファイル転送
    - デスクトップとクライアント間のファイル転送を許可する
    - デスクトップから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

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

2104の新機能

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

Citrix DaaS™におけるドメインに参加していないLinux VDAのサポート

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

OpenJDK 11 が必須に

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

XDPing の変更点

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

Ubuntu のスマートカードサポート

ユーザーは、Linux 仮想デスクトップセッションにログオンする際に、クライアントデバイスに接続されたスマートカードを認証に使用できます。このリリースでは、Ubuntu 20.04、Ubuntu

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

MATE デスクトップのサポート

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

RHEL 8、CentOS 8、SUSE 12.5 の PBIS サポート

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

2103 の新機能

Linux VDA のバージョン 2103 には、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

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

SUSEマシンをWindowsドメインに参加させるためのSSSDサポート

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

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

Citrix Virtual Apps and Desktops Standard for Azureで、ドメイン参加済みと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、ファイル転送機能を強化しました。また、一度に転送できるファイルの数を10個から100個に増やしました。詳しくは、「ファイル転送」を参照してください。

Citrix Studioでのセッション接続タイマー構成のサポート

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

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

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

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

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

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

このリリースでは、Citrix Director の Linux VM および Linux セッションに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

詳しくは、「Citrix Director での Linux VM および Linux セッションの監視」(/ja-jp/linux-virtual-delivery-agent/2103/configuration/monitor-linux-sessions-in-director.html)を参照してください。

Linux VDA の FAS 強化

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

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

Citrix Provisioning の Linux ストリーミング機能を使用すると、Citrix Virtual Apps and Desktops 環境に Linux 仮想デスクトップを直接プロビジョニングできます。詳しくは、「Linux ストリーミング」(/ja-jp/linux-virtual-delivery-agent/2103/configuration/linux-streaming.html)を参照してください。

2012 の新機能

Linux VDA のバージョン 2012 には、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伝送単位 (MTU) を自動的に決定します。これにより、パフォーマンスの低下やセッション確立の失敗につながる可能性のある EDT パケットの断片化を防ぎます。詳しくは、「EDT MTU 検出」(/ja-jp/linux-virtual-delivery-agent/2301/configure/session/adaptive-transport.html#edt-mtu-discovery)を参照してください。

LDAPチャネルバインディングのサポート

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

Linux VDAに統合されたXDPing

Linux **XDPing**ツールは、Linux VDA環境における一般的な構成の問題をチェックするプロセスを自動化するコマンドラインベースのアプリケーションです。以前は、CTX202015ページから**XDPing**をダウンロードし、Linux VDAとは別にツールをインストールする必要がありました。このリリースから、`XDPing`はLinux VDAに統合され、Linux VDAのインストール時に自動的にインストールできます。

RHEL 7.9、RHEL 8.3のサポート

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

Ubuntu 20.04のサポート

以前は実験的な機能として提供されていましたが、このリリースではUbuntu 20.04が完全にサポートされます。

Rendezvousプロトコルのサポート

Linux VDAはRendezvousプロトコルをサポートするようになりました。Citrix Gatewayサービスを使用する環境では、Rendezvousプロトコルにより、HDXセッションが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。

グーグル クラウド プラットフォーム (GCP) 上でのマシン作成サービス (MCS) のサポート

MCSを使用してGCP上にLinux VMを作成できるようになりました。詳しくは、「Machine Creation Services（MCS）を使用したLinux VDAの作成」を参照してください。

LinuxマシンでWake on LANが利用可能に

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

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

2009年の新機能

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

(実験的) Ubuntu 20.04のサポート

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

ブラウザーコンテンツリダイレクトのサポート

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

セッションウォーターマークのサポート

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

Citrix Virtual Apps and Desktops Standard for Azureにおけるドメイン非参加Linux VDAのサポート

Citrix Virtual Apps and Desktops Standard for Azureでドメイン非参加Linux VDAを作成し、Microsoft Azureから任意のデバイスに仮想アプリとデスクトップを配信できるようになりました。詳しくは、「Citrix DaaS Standard for AzureでLinux VDAを作成する」および「Citrix Virtual Apps and Desktops Standard for Azure」を参照してください。

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

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

VDAから追加のデスクトップセッションを開くときに表示されるプロンプト

VDAからCitrix HDX、RDP、および直接コンソール接続を介して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、次のプロンプトで現在のデスクトップセッションから強制的にログオフすることで、新しいデスクトップセッションを続行できるようになりました。このプロンプトは、VDAがインストールされているLinuxプラットフォームに関係なく、VDAから追加のデスクトップセッションを開くたびに表示されます。

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

> プロンプトの画像(/en-us/linux-virtual-delivery-agent/2301/media/prompt.png)

2006の新機能

Linux VDAバージョン2006には、次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。詳しくは、「Citrix製品の言語サポート」を参照してください。

SUSE 12 SP5のサポート

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

クライアントドライブマッピングの機能強化

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

セッションとクライアント間のファイルコピー＆ペーストのサポート

Citrix Virtual Apps and Desktops 2006およびWindows用Citrix Workspace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、「ファイルのコピーとペースト」を参照してください。

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

2003の新機能

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

RHEL 8のサポート

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

Citrix Telemetry Serviceとの統合

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

軽量NSAP仮想チャネルのサポート

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

自動ダウンロード

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

NVIDIA Tesla T4のサポート

NVIDIA Tesla T4がサポートされています。

システム要件

Linux VDAのCurrent Release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、およびCitrixが特定のバージョンの製品のサポートを終了する時期については、Citrix製品ライフサイクルマトリックスを参照してください。

Linux VDAの構成プロセスは、Windows VDAとは若干異なります。任意のDelivery Controller™ファームは、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Current Release (CR) を使用する方法、およびその他のFAQについては、Knowledge Centerの記事を参照してください。

Linuxディストリビューション

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

> **重要:**

> OSベンダーからのサポートが終了すると、Citrix®は問題の修正能力が制限される可能性があります。

> 非推奨または削除されたプラットフォームについては、非推奨を参照してください。

- アマゾン Linux
  - アマゾン Linux 2
- セントオーエス リナックス
  - CentOS 7.9
- デビアン リナックス

    - デビアン 11.3
- レッドハット エンタープライズ リナックス
    - ワークステーション 9.0
    - ワークステーション 8.7
    - ワークステーション 8.6
    - ワークステーション 8.4
    - ワークステーション 7.9
    - サーバー 9.0
    - サーバー 8.7
    - サーバー 8.6
    - サーバー 8.4
    - サーバー 7.9
- ロッキー Linux 9.0
- ロッキー Linux 8.7
- ロッキー Linux 8.6
- SUSE リナックス エンタープライズ：
    - サーバー 15 サービスパック 4
- Ubuntu リナックス
    - Ubuntu デスクトップ 22.04
    - Ubuntu サーバー 22.04
    - Ubuntu デスクトップ 20.04
    - Ubuntu サーバー 20.04
    - Ubuntu デスクトップ 18.04
    - Ubuntu サーバー 18.04
    - Ubuntu ライブ サーバー 18.04

> **注:**

CentOSプロジェクトはCentOS Streamに焦点を移します。RHEL 8の再構築版であるCentOS Linux 8は、2021年末に終了します。CentOS Streamはその後も継続し、Red Hat Enterprise Linuxのアップストリーム（開発）ブランチとして機能します。詳細については、`<https://www.redhat.com/en/blog/centos-stream-building-innovative-future-enterprise-linux>`を参照してください。

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、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	サポートされるデスクトップ
アマゾン リナックス 2	1.20	マテ、GNOME、GNOME Classic
デビアン 11.3	1.20	メイト、グノーム、グノームクラシック、KDE
RHEL 9.0、ロッキー リナックス 9.0	1.20	GNOME
RHEL 8.7/8.6/8.4	1.20	メイト、グノーム、グノームクラシック
RHEL 7.9、CentOS 7.9	1.20	メイト、グノーム、グノームクラシック、ケーディーイー
Rocky リナックス 8.7/8.6	1.20	メイテ、グノーム、グノームクラシック、ケーディーイー
SUSE 15.4	1.20	メイト、グノーム、グノームクラシック
ウブントゥ 22.04	1.21	メイト、グノーム、グノームクラシック、ケーディーイー
ウブントゥ 20.04	1.20	メイト、グノーム、グノームクラシック、ケーディーイー
ウブントゥ 18.04	1.19	メイト、グノーム、グノームクラシック、ケーディーイー

ヒント:

Ubuntuでは`HWE kernel`または`HWE Xorg`を使用しないでください。

少なくとも1つのデスクトップがインストールされている必要があります。セッションで使用するGNOMEまたはMATEデスクトップ環境は、ctxinstall.shまたはctxsetup.shスクリプトを介して指定できます。 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、User/Group Name Syntax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および仮想化環境

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

注:

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

Citrix Virtual Apps and Desktops 7 2003から2112までは、Microsoft Azure、AWS、GCPでのLinux VDAのホスティングは、Citrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）でのみサポートされていました。2203リリース以降、これらのパブリッククラウドでCitrix DaaSとCitrix Virtual Apps and Desktopsの両方に対して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、

Hybrid Rights Licenseが必要です。**Hybrid Rights License**の詳細については、「Transition and Trade-Up (TTU) with Hybrid Rights」を参照してください。

Active Directory統合パッケージ

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

	ウィンバインド	SSSD	セントリファイ	PBIS	クエスト
アマゾンLinux 2	はい	はい	はい	はい	いいえ
デビアン11.3	はい	はい	はい	はい	いいえ
RHEL 9.0	はい	はい	いいえ	いいえ	いいえ
RHEL 8.7/8.6/8.4	はい	はい	はい	はい	いいえ
RHEL 7.9、CentOS 7.9	はい	はい	はい	はい	はい (Quest v4.1以降)
ロッキーLinux 9.0	はい	はい	いいえ	いいえ	いいえ
ロッキーLinux 8.7/8.6	はい	はい	いいえ	いいえ	いいえ
SUSE 15.4	はい	はい	はい	はい	いいえ
ウブント22.04/20.04/18.04	はい	はい	はい	はい	はい (Quest v4.1以降)

HDX™ 3D プロ

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

ハイパーバイザー

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

- シトリックス ハイパーバイザー
- ヴイエムウェア vSphere ハイパーバイザー
- ニュータニックス AHV
- マイクロソフト アジュール
- アマゾン ウェブ サービス (AWS)
- グーグル クラウド プラットフォーム (GCP)

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

GPU

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

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

インストール概要

このセクションでは、以下の手順について説明します。

- 簡易インストールを使用してドメイン参加済みVDAを作成する
- MCSを使用してドメイン非参加のLinux VDAを作成する
- MCSを使用してLinux VDAを作成する
- Citrix Provisioningを使用してLinux VDAを作成する
- Citrix DaaS Standard for AzureでLinux VDAを作成する
- Linux VDAを手動でインストールする
  - Amazon Linux 2、CentOS、RHEL、およびRocky LinuxにLinux VDAを手動でインストールする
  - SUSEにLinux VDAを手動でインストールする
  - UbuntuにLinux VDAを手動でインストールする
  - DebianにLinux VDAを手動でインストールする

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、手動インストールよりも時間と労力を節約し、エラーが発生しにくいです。必要なパッケージをインストールし、構成ファイルを自動的にカスタマイズすることで、Linux VDAの実行環境をセットアップするのに役立ちます。

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

ステップ1：構成情報とLinuxマシンを準備する

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)

重要:

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

考慮事項

- ワークグループ名は、デフォルトでドメイン名です。環境内のワークグループをカスタマイズするには、次の手順を実行します。

  a. Linux VDAマシン上に/tmp/ctxinstall.confファイルを作成します。

  b. ファイルにworkgroup=\<your workgroup\>の行を追加し、変更を保存します。

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

  ログ:

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

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

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

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

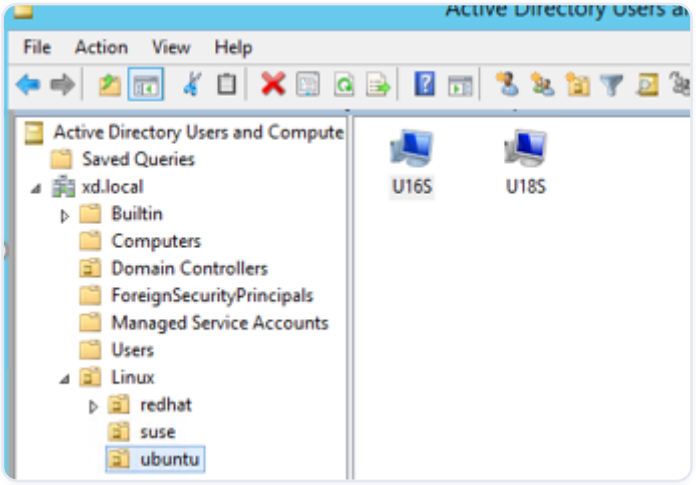
a. ドメインコントローラーで**管理ツール**を開きます。

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

- VDAを特定のOUに参加させるには、次の手順を実行します。

  a. 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. VDA上に/tmp/ctxinstall.confファイルを作成します。

  c. /tmp/ctxinstall.confファイルにou=\<your ou\>の行を追加します。

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

OS	ウィンバインド	SSSD	セントリファイ	PBIS
アマゾン Linux 2	`ou="Linux/amazon"`	`ou="Linux/amazon"`	`ou="XD.LOCAL/Linux/amazon"`	`ou="Linux/amazon"`
デビアン	`ou="Linux/debian"`	`ou="Linux/debian"`	`ou="XD.LOCAL/Linux/debian"`	`ou="Linux/debian"`
RHEL 9.0、Rocky Linux 9.0	`ou="OU=redhat,OU=Linux"`	`ou="OU=redhat,OU=Linux"`	該当なし	該当なし
RHEL 8.x および Rocky Linux 8.x	`ou="OU=redhat,OU=Linux"`	`ou="OU=redhat,OU=Linux"`	`ou="XD.LOCAL/Linux/redhat"`	`ou="Linux/redhat"`
RHEL 7	`ou="Linux/redhat"`	`ou="Linux/redhat"`	`ou="XD.LOCAL/Linux/redhat"`	`ou="Linux/redhat"`
SUSE	`ou="Linux/suse"`	`ou="Linux/suse"`	`ou="XD.LOCAL/Linux/suse"`	`ou="Linux/suse"`
ウブントゥ	`ou="Linux/ubuntu"`	`ou="Linux/ubuntu"`	`ou="XD.LOCAL/Linux/ubuntu"`	`ou="Linux/ubuntu"`

- Easy Installは、Linux VDA 7.16以降、純粋なIPv6をサポートします。以下の前提条件と制限が適用されます。
  - マシンが純粋なIPv6ネットワーク経由で必要なパッケージをダウンロードできるように、Linuxリポジトリが設定されている必要があります。
  - Centrifyは、純粋なIPv6ネットワークではサポートされていません。

> **注:**

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

- ドメインに参加する方法としてCentrifyを選択した場合、ctxinstall.shスクリプトにはCentrifyパッケージが必要です。ctxinstall.shがCentrifyパッケージを取得する方法は2つあります。
  - Easy Installは、インターネットからCentrifyパッケージを自動的にダウンロードするのに役立ちます。各ディストリビューションのURLは次のとおりです。

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

    CentOSでは、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 コマンドで `<https://downloads.centrify.com/products/server-suite/2022/component-update-1/delinea-server-suite-2022-deb9-x86_64.tgz>` を実行します。
  - 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、次の手順を実行します。

    a. Linux VDA サーバー上に /tmp/ctxinstall.conf ファイルが存在しない場合は、そのファイルを作成してください。

    b. そのファイルに「centrifypkgpath=\<path name\>」という行を追加してください。

    例:

    ```
    cat /tmp/ctxinstall.conf

    set "centrifypkgpath=/home/mydir"

    ls  -ls  /home/mydir

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- ドメインに参加する方法としてPBISを選択した場合、ctxinstall.shスクリプトにはPBISパッケージが必要です。ctxinstall.shがPBISパッケージを取得する方法は2つあります。

- 簡単インストールでは、インターネットからPBISパッケージを自動的にダウンロードできます。たとえば、各ディストリビューションのURLは次のとおりです。

  Amazon Linux 2、CentOS 7、RHEL 8、RHEL 7、SUSE 15.4: 詳細については`wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh`を参照してください。

  デビアン、ウブントゥ: `wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.deb.sh`

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

  例については、次のスクリーンショットを参照してください。



ステップ2：ハイパーバイザーの準備

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

Citrix Hypervisor™での時刻同期の修正

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

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

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

この変更を再起動後も永続的に適用するには、**/etc/sysctl.conf**ファイルを編集し、次の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration ServicesがインストールされているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの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. **時刻の同期**が選択されていることを確認します。

> **注:**

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

ESXおよびESXiでの時刻同期の修正

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

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

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

ステップ3：.NET Runtime 6.0のインストール

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

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

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

ステップ4：Linux VDAパッケージのダウンロード

1. Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。
2. 「Citrix Virtual Apps and Desktops」の適切なバージョンを展開します。
3. 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パッケージと、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。

   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、次のコマンドを実行します：

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

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

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

```
sudo apt-get install dpkg-sig

gpg --import <path to the public key>

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

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 9.0およびRocky Linux 9.0にLinux VDAをインストールする前に、**libsepol** パッケージをバージョン3.4以降に更新してください。

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

注:

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

```
nmcli dev connect eth0

service NetworkManager restart
```

Ubuntu/Debianディストリビューションの場合：

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

sudo apt-get install -f
```

注:

- Debian 11.3ディストリビューションに必要な依存関係をインストールするには、/etc/apt/sources.listファイルに `deb http://deb.debian.org/debian/ bullseye main` の行を追加します。
- GCP上のUbuntu 20.04の場合、RDNSを無効にします。そのためには、/etc/krb5.confの の下に **rdns = false** 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 15.4 の場合、以下を確認してください。
   - **libstdc++6** バージョン 12 以降を使用していること。
   - **/etc/sysconfig/windowmanager** 内の **Default_WM** パラメーターが **"gnome"** に設定されていること。
2. Linux VDA をインストールするには、次のコマンドを実行します。

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

ステップ 6: NVIDIA GRID ドライバーをインストールする

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

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

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

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

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

ステップ 7: 使用するデータベースを指定する

実験的な機能として、PostgreSQL に加えて SQLite を使用できます。Linux VDA パッケージのインストール後に`/etc/xdl/db.conf`を編集することで、SQLite と PostgreSQL を切り替えることもできます。

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

> **注:**
>
> - VDIモードではSQLiteのみを使用することをお勧めします。
> - Easy Install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`で特に指定しない限り、Linux VDAはデフォルトでPostgreSQLを使用します。
> - PostgreSQLのポート番号を設定するには、`/etc/xdl/db.conf`を使用することもできます。

ステップ8：インストールを完了するためのランタイム環境のセットアップ

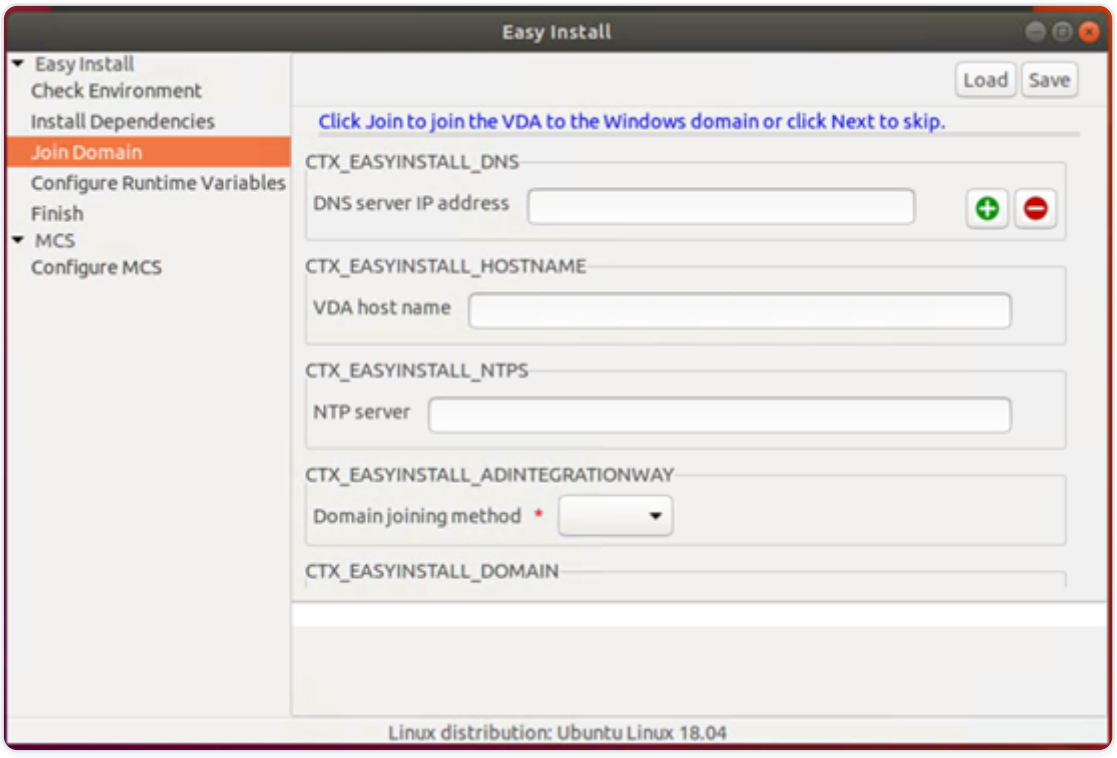
Linux VDAパッケージをインストールした後、ctxinstall.shスクリプトを使用して実行環境を構成します。このスクリプトは、対話モードまたはサイレントモードで実行できます。

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

> **対話モード：**

対話モードでEasy Installを使用する方法は2つあります。

- `sudo /opt/Citrix/VDA/sbin/ctxinstall.sh`コマンドを実行し、コマンドラインインターフェイスの各プロンプトで関連するパラメーターを入力します。
- VDAのデスクトップ環境で`/opt/Citrix/VDA/bin/easyinstall`コマンドを実行し、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
- ランタイム環境を構成する

> **ヒント:**
>
> 「**保存**」をクリックして、指定したパスのローカルファイルに変数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。MCS変数の構成については、「手順3：マスターイメージの準備」を参照してください。

「**サイレントモード：**」:

サイレントモードで簡易インストールを使用するには、ctxinstall.shを実行する前に、次の環境変数を設定します。

- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDAサーバーのホスト名を示します。
- **CTX_EASYINSTALL_DNS=ip-address-of-dns** – DNSのIPアドレス。
- **CTX_EASYINSTALL_NTPS=address-of-ntps** – NTPサーバーのIPアドレスまたは文字列名。
- **CTX_EASYINSTALL_DOMAIN=domain-name** – ドメインのNetBIOS名。
- **CTX_EASYINSTALL_REALM=realm-name** – Kerberosレルム名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis** – Active Directory統合方法を示します。
- **CTX_EASYINSTALL_USERNAME=domain-user-name** – ドメインユーザーの名前を示します。ドメインへの参加に使用されます。
- **CTX_EASYINSTALL_PASSWORD=password** – ドメインユーザーのパスワードを指定します。ドメインへの参加に使用されます。

ctxsetup.shスクリプトは、次の変数を使用します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAがDelivery Controllerに登録するために使用する、スペース区切りのDelivery Controllerの完全修飾ドメイン名（FQDN）のリストが必要です。少なくとも1つのFQDNまたはCNAMEを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktopサービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDAサービスでは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktopの場合、必要なポート（デフォルトではポート80および1494）をシステムファイアウォールで自動的に開くことができます。
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合は、値をYに設定します。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、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。LDAPポート番号を389と指定すると、Linux VDAは指定されたドメイン内の各LDAPサーバーにポーリングモードでクエリを実行します。ポリシーの数がx、LDAPサーバーの数がyの場合、Linux VDAは合計XかけるYのクエリを実行します。ポーリング時間がしきい値を超えると、セッションログオンが失敗する可能性があります。より高速なLDAPクエリを有効にするには、ドメインコントローラーで**グローバルカタログ**を有効にし、関連するLDAPポート番号を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_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス（FAS）サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていませんが、代わりにセミコロン区切りのFASサーバーのリストを指定できます。シーケンスは、ADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を**\<none\>**テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。FASサーバーと適切に通信するには、FASサーバーで指定されているものと一致するポート番号を付加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Runtime 6.0をインストールするパス。デフォルトのパスは/usr/binです。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEの場合は、変数値を**mate**に設定する必要があります。

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

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

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

  - **MATE デスクトップの場合**

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

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


fi
```

- **GNOME Classic デスクトップの場合**

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


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


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

- **GNOME デスクトップの場合**

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


if [ -n "$GSESSION" ]; then
exec gnome-session


fi
```

b. ターゲットセッションユーザーと700ファイル権限を共有します。

バージョン2209以降、セッションユーザーはデスクトップ環境をカスタマイズできます。この機能を有効にするには、VDAに切り替え可能なデスクトップ環境を事前にインストールする必要があります。詳細については、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

- **CTX_XDL_START_SERVICE=Y \| N** – 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_TELEMETRY_SOCKET_PORT** – Citrix Scoutのリッスン用ソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutとの通信用ポート。デフォルトポートは7502です。

いずれかのパラメーターが設定されていない場合、インストールは対話モードに戻り、ユーザー入力が求められます。すべてのパラメーターが環境変数を通じてすでに設定されている場合でも、ctxinstall.shスクリプトは.NET Runtime 6.0のインストールパスについてユーザー入力を求めます。

サイレントモードでは、環境変数を設定するために次のコマンドを実行し、その後ctxinstall.shスクリプトを実行する必要があります。

```
export CTX_EASYINSTALL_HOSTNAME=host-name


export CTX_EASYINSTALL_DNS=ip-address-of-dns


export CTX_EASYINSTALL_NTPS=address-of-ntps


export CTX_EASYINSTALL_DOMAIN=domain-name


export CTX_EASYINSTALL_REALM=realm-name


export CTX_EASYINSTALL_FQDN=ad-fqdn-name
```

```
export CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis

export CTX_EASYINSTALL_USERNAME=domain-user-name

export CTX_EASYINSTALL_PASSWORD=password

export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N

export CTX_XDL_DDC_LIST='list-ddc-fqdns'

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y | N

export CTX_XDL_ADD_FIREWALL_RULES=Y | N

export CTX_XDL_HDX_3D_PRO=Y | N

export CTX_XDL_VDI_MODE=Y | N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y | N

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

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \
```

```
CTX_XDL_FAS_LIST='list-fas-servers' \


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


CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \


CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \


CTX_XDL_TELEMETRY_PORT=port-number \


CTX_XDL_START_SERVICE=Y|N \


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

ステップ9: XDPingを実行する

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

ステップ10: Linux VDAを実行する

Linux VDA を開始する:

Linux VDA サービスを開始するには:

```
sudo /sbin/service ctxhdx start
```

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

Linux VDA を停止する:

Linux VDA サービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDA を再起動する:

Linux VDA サービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDA のステータスを確認する:

Linux VDA サービスの稼働状況を確認するには:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ 11: マシンカタログを作成する

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

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

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

注:

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

ヒント:

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

ステップ12：デリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループを作成する」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 2212」を参照してください。

ステップ13：Linux VDAをアップグレードする（オプション）

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

RHEL 7およびCentOS 7の場合：

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

RHEL 8およびRocky Linux 8の場合：

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

RHEL 9.0およびRocky Linux 9.0の場合:

> **注:**
>
> RHEL 9.0およびRocky Linux 9.0でLinux VDAをアップグレードする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

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

SUSEの場合:

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

Ubuntu 18.04の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu18.04_amd64.deb
```

Ubuntu 20.04の場合:

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

Ubuntu 22.04の場合:

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

トラブルシューティング

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

SSSDを使用したドメインへの参加が失敗する

ドメインに参加しようとするとエラーが発生し、次のような出力が表示されることがあります (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る場合に発生します。さらに、ローカルユーザーアカウントを使用してログオンすると、マシンのコンソールにもグレースクリーンが表示されます。

この問題を解決するには：

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

   `greeter-show-manual-login=true`

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

/var/log/xdl/hdx.log:

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



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



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



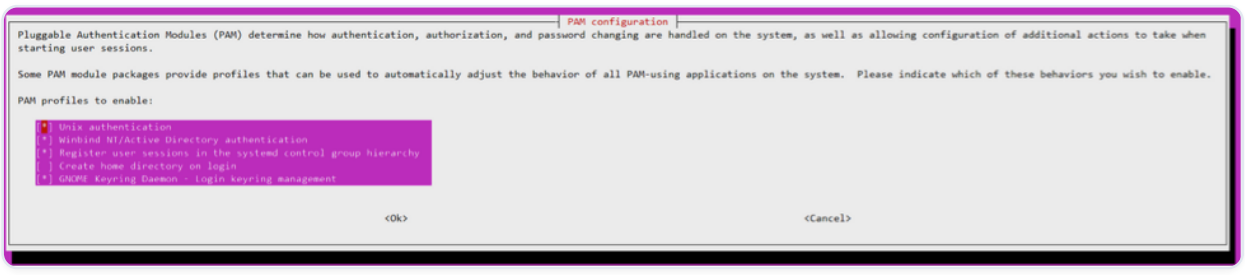
2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting normally.
```

ヒント:

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

この問題を解決するには：

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



セッションが起動しない、またはdbusエラーですぐに終了する

/var/log/messages (RHELまたはCentOSの場合):

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



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



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



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



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



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



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

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

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

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



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



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



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



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

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

SELinuxがSSHDのホームディレクトリへのアクセスを妨げる

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/ctxinstall.log:

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



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



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



If you want to allow polyinstantiation to enabled



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



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



    Do



       setsebool -P polyinstantiation_enabled 1



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



If you believe that sshd should be allowed setattr access on the root
```

```
directory by default.


Then you should report this as a bug.


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


      Do


       allow this access for now by executing:


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


# semodule -i mypol.pp
```

この問題を解決するには:

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

   SELINUX=無効

2. VDA を再起動します。

ドメインに参加していないLinux VDAを作成する

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

重要:

- ドメインに参加していないVDAは、Citrix DaaSでサポートされています。
  - コントロールプレーンはCitrix DaaS上に展開する必要があります。
  - ドメインに参加していないVDAは、パブリッククラウドまたはオンプレミスのデータセンターに展開できます。ドメインに参加していないVDAは、Citrix DaaSのコントロールプレーンによって管理されます。
  - Citrix Cloud ConnectorsをバイパスするようにRendezvous V2を構成できます。そうしないと、VDAをコントロールプレーンに接続するためにCloud Connectorsをインストールする必要があります。
- ドメインに参加していないVDAを作成するには、MCSを使用する必要があります。
  - ベアメタルサーバーはMCSではサポートされていません。
- ドメインに参加していないLinux VDAでは、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(/ja-jp/linux-virtual-delivery-agent/2301/configure/session/rendezvous-v2.html)

(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 Support Portalで入手できるNutanix Acropolis MCSプラグインインストールガイドを参照してください。

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

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

オンプレミスのDelivery Controller用Nutanix AHVプラグイン(/en-us/linux-virtual-delivery-agent/2301/media/nutanix-plug-in-for-cvad.png)

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

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

ステップ 1c: プラグインのインストール後に次の手順を完了します

- 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 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

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

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

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

ステップ 2a: 接続

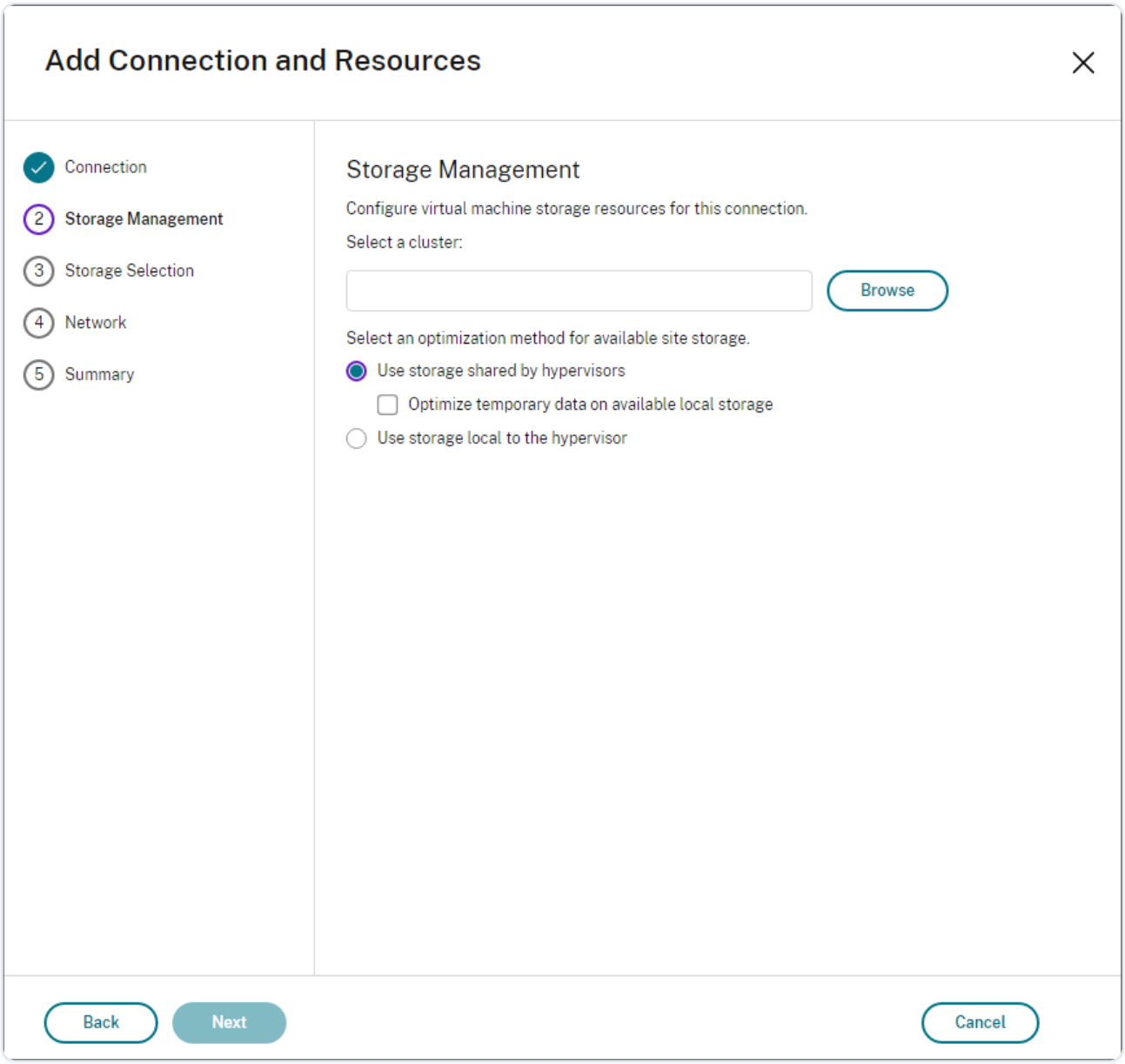
接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2301/media/add-conn-connection.png)

「**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を選択します: Machine Creation ServicesまたはCitrix Provisioning。

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を使用している場合、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。

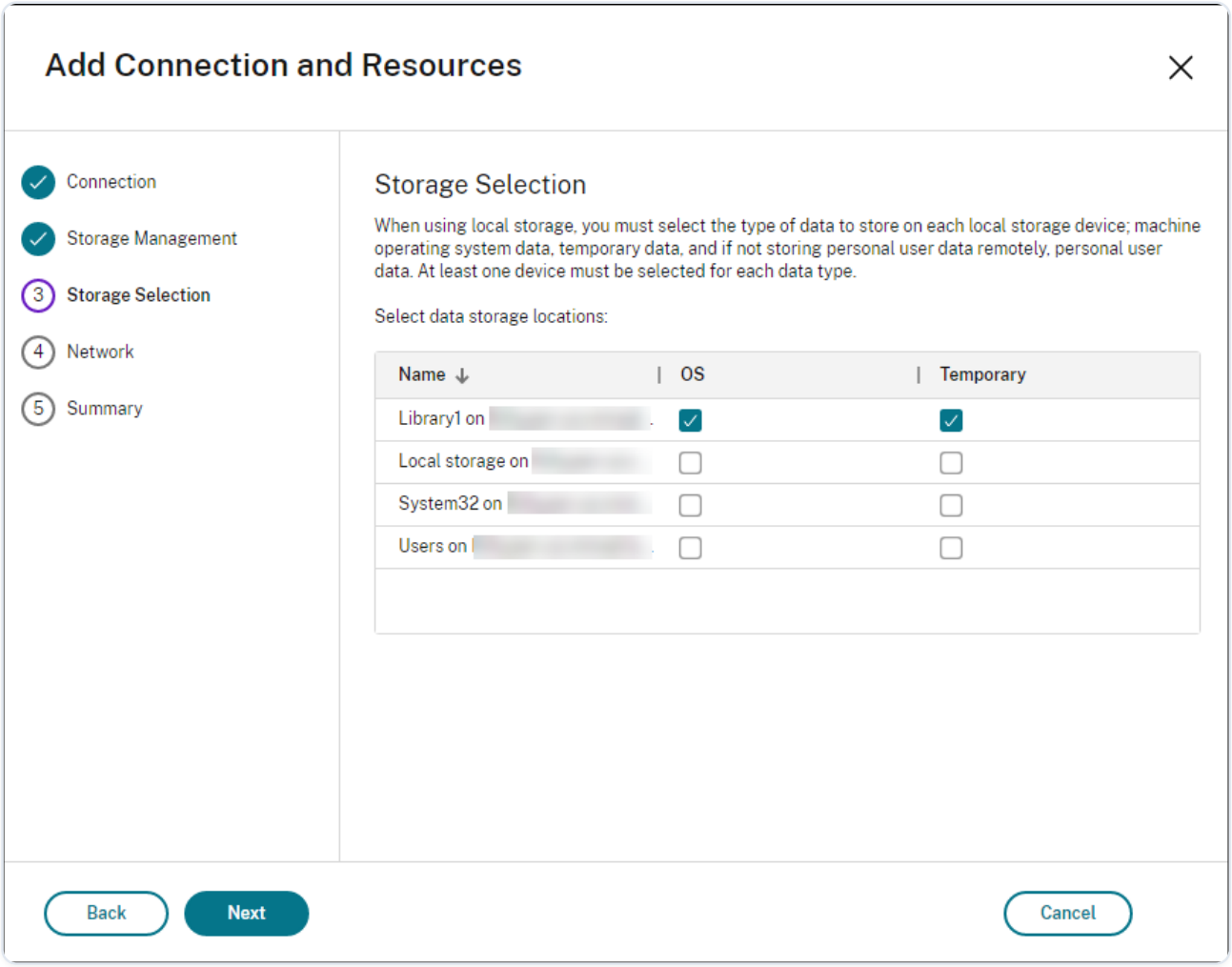
Hyper-V または VMware ホストへの接続を構成している場合は、クラスター名を参照して選択します。その他の接続タイプでは、クラスター名は要求されません。

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。

- ハイパーバイザー間で共有されるストレージを選択した場合、利用可能なローカルストレージに一時データを保持するかどうかを指定します。（この接続を使用するマシンカタログで、デフォルト以外のサイズの一時ストレージを指定できます。）**例外:** クラスター共有ボリューム（CSV）を使用する場合、Microsoft System Center Virtual Machine Managerでは、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の負荷を軽減するためにIntelliCacheを使用するかどうかを指定します。Citrix Hypervisor仮想化環境を参照してください。

ステップ 2c: ストレージの選択



ストレージの選択の詳細については、「ホストストレージ」を参照してください。

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

ストレージの選択ページの下部には、ハイパーバイザー間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**を有効にした場合に、さらに多くの構成オプションが含まれます。一時データに使用するローカルストレージデバイス（同じハイパーバイザープール内）を選択できます。

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

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

ステップ 2d: リージョン

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

ステップ 2e: ネットワーク

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

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

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ってください。確認が完了したら、**「完了」**を選択します。

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

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

ステップ3：マスターイメージの準備

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

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

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

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

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

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

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

   RHEL/CentOS/Rocky Linux の場合:

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

   Ubuntu/Debian の場合:

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

   SUSE の場合:

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

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

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

1. VMが再起動または停止されたときにVDAホスト名が永続化されるようにするには、次のコマンドを実行します:

   ```
   echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
   ```

   /etc/cloud/cloud.cfgファイル内の**system_info**セクションの下に次の行が存在することを確認します:

   ```
   system_info:

      network:

          renderers: ['netplan', 'eni', 'sysconfig']
   ```

2. AWS上のMCSで作成されたVMにSSHでリモートアクセスするには、それらのVMにはキー名がアタッチされていないため、パスワード認証を有効にします。必要に応じて、次の操作を行います。

   - `cloud-init`構成ファイル、/etc/cloud/cloud.cfgを編集します。**ssh_pwauth: true**の行が存在することを確認します。**set-password**の行と、それに続く行が存在する場合は、削除またはコメントアウトします。

     ```
     users:

     - default
     ```

   - `cloud-init`によって作成されたデフォルトユーザー`ec2-user`または`ubuntu`を使用する予定がある場合、`passwd`コマンドを使用してユーザーパスワードを変更できます。新しいパスワードは、後でMCSで作成されたVMにログインするために覚えておいてください。

   - `/etc/ssh/sshd_config`ファイルを編集して、次の行が存在することを確認します:

     ```
     PasswordAuthentication yes
     ```

ファイルを保存し、`sudo service sshd restart`コマンドを実行します。

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

注:

現在実行中のVDAをテンプレートVMとして使用する場合は、この手順をスキップします。

テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Runtime 6.0をインストールします。

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

RHEL/CentOS/Rocky Linuxの場合：

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

注:

RHELおよびCentOSの場合、Linux VDAをインストールして`deploymcs.sh`を正常に実行する前に、EPELリポジトリをインストールしてください。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>`の指示を参照してください。

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

  ```
  nmcli dev connect eth0

  service NetworkManager restart
  ```

Ubuntu/Debianの場合：

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



apt-get install -f
```

SUSEの場合：

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

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

RHEL 7サーバーの場合：

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

RHEL 7ワークステーションの場合：

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

ステップ3e: （SUSEの場合）ntfs-3gを手動でインストールする

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

1. GNU 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 3f: 使用するデータベースを指定する

実験的な機能として、PostgreSQL に加えて SQLite を使用できます。Linux VDA パッケージのインストール後、SQLite と PostgreSQL を切り替えることもできます。これを行うには、次の手順を完了します:

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

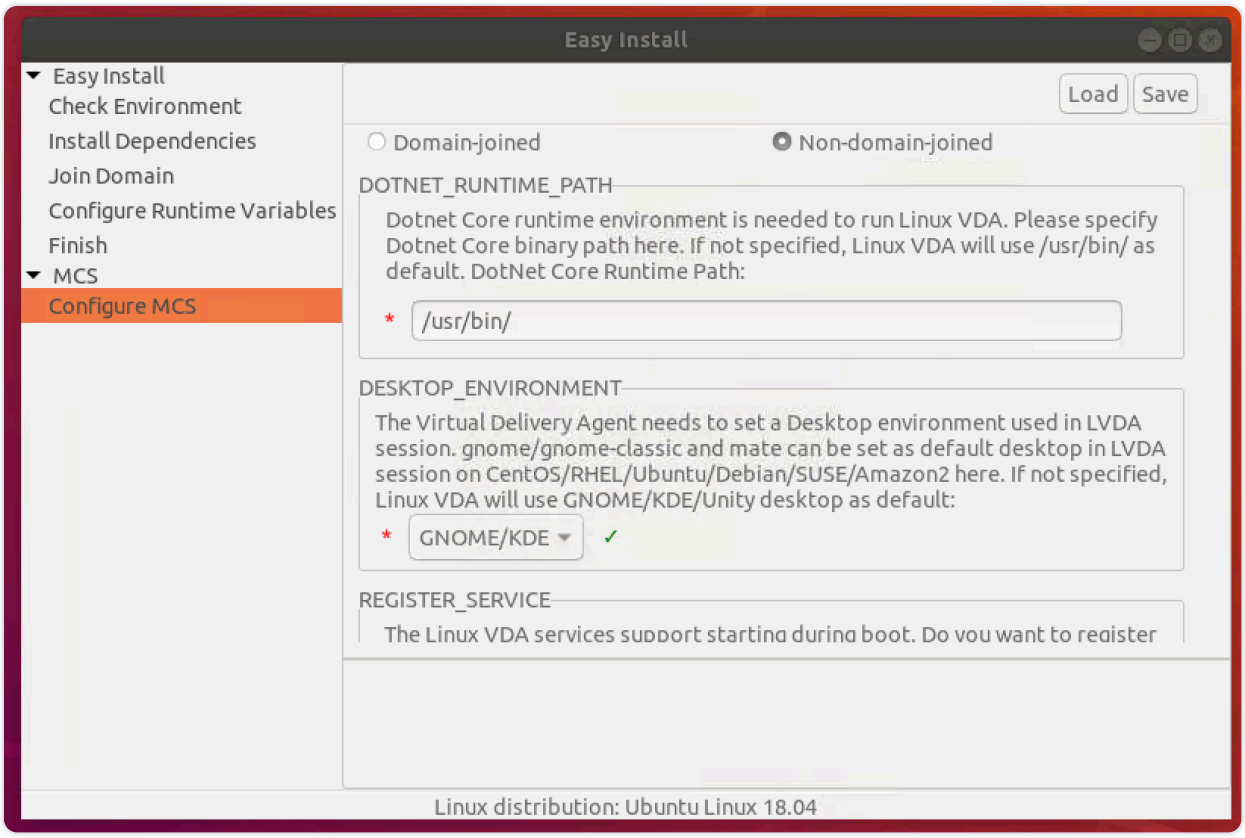
注:

- VDI モードでのみ SQLite を使用することをお勧めします。
- 簡単なインストールと MCS の場合、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。
- `/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 3g: MCS 変数を構成する

MCS 変数を構成するには、次の2つの方法があります:

- `/etc/xdl/mcs/mcs.conf` ファイルを編集します。
- 簡単インストール 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:
>
> **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、ドメインに参加していないシナリオ用に構成できる MCS 変数です。デフォルトの変数値を使用するか、必要に応じて変数をカスタマイズできます (オプション):

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

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

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

`START_SERVICE` =**Y | N**

ステップ3h: MCSのレジストリ値を書き込むか更新する（オプション）

テンプレートマシンで、必要に応じてレジストリ値を書き込むか更新するために、`/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。この操作により、MCS でプロビジョニングされたマシンが再起動するたびにデータと設定が失われるのを防ぎます。

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

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

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

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

ステップ3i: 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、`/opt/Citrix/VDA/sbin/deploymcs.sh`を実行します。GUIを使用してMCS変数を構成する場合は、**Deploy**をクリックします。

   簡易インストールGUIのDeployボタン(/en-us/linux-virtual-delivery-agent/2301/media/deploy-button-on-easy-install-gui.png)

   GUIで**Deploy**をクリックすると、GUIで設定した変数が、`/etc/xdl/mcs/mcs.conf`ファイルで設定した変数を上書きします。
2. 使用するパブリッククラウドに基づいて、マスターイメージのスナップショットを作成し、名前を付けます。

   - **(Citrix Hypervisor、GCP、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、その名前が必要です。

     テンプレートVMの停止済み電源ステータス(/en-us/linux-virtual-delivery-agent/2301/media/stopped-deallocated2.png)

- **(AWSの場合)** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンスステータスが**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。必要に応じて情報を入力し、設定を行います。**Create Image**をクリックします。

  EBSイメージの作成(/en-us/linux-virtual-delivery-agent/2301/media/create-ebs-image.png)

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

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

ステップ4：マシンカタログを作成する

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

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

   マスターイメージページで、イメージスナップショットを選択します。

   仮想マシンページで、仮想CPUの数とvCPUあたりのコア数を確認します。マシン展開方法としてMCSを選択し、カタログに作成するマシンのIDとして**ドメイン非参加**を選択します。

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

> **注:**

Delivery Controller™でのマシンカタログ作成プロセスにかなりの時間がかかる場合は、Nutanix Prismに移動し、**Preparation**というプレフィックスが付いたマシンを手動でパワーオンします。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、「デリバリーグループの作成」を参照してください。

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

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

重要:

2212 リリース以降の重要な変更点は次のとおりです。

- /etc/xdl/mcs/mcs.conf ファイル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 3h: MCS 変数の構成」セクションを参照してください。
- /etc/xdl/mcs/mcs.conf の **UPDATE_MACHINE_PW** エントリの有効な値は、**enabled** または **disabled** ではなく、**Y** または **N** になりました。詳細については、この記事の「マシンアカウントパスワードの自動更新」セクションを参照してください。

サポートされているディストリビューション

	ウィンバインド	SSSD	セントリファイ	PBIS
デビアン 11.3	はい	はい	いいえ	はい
RHEL 9.0	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
RHEL 8.7/8.6/8.4	はい	いいえ	はい	はい
ロッキー Linux 9.0	はい	いいえ	いいえ	いいえ
ロッキー Linux 8.7/8.6	はい	いいえ	いいえ	いいえ
RHEL 7.9、CentOS 7.9	はい	はい	はい	はい
SUSE 15.4	はい	はい	いいえ	はい
ウブントゥ ２２．０４、ウブントゥ ２０．０４、ウブントゥ １８．０４	はい	はい	いいえ	はい

サポートされているハイパーバイザー

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

サポートされているハイパーバイザー以外のハイパーバイザーでマスターイメージを準備しようとすると、予期しない結果が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

- Citrix Virtual Apps and Desktops 7 2003からCitrix Virtual Apps and Desktops 7 2112まで、Microsoft Azure、AWS、GCPでのLinux VDAのホスティングは、Citrix DaaS（旧Citrix Virtual

Apps and Desktopsサービス）でのみサポートされていました。2203リリース以降、Citrix DaaSとCitrix Virtual Apps and Desktopsの両方で、これらのパブリッククラウドに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、**ハイブリッド権限ライセンス**が必要です。**ハイブリッド権限ライセンス**の詳細については、「Transition and Trade-Up (TTU) with Hybrid Rights」を参照してください。

- 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。
- 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。

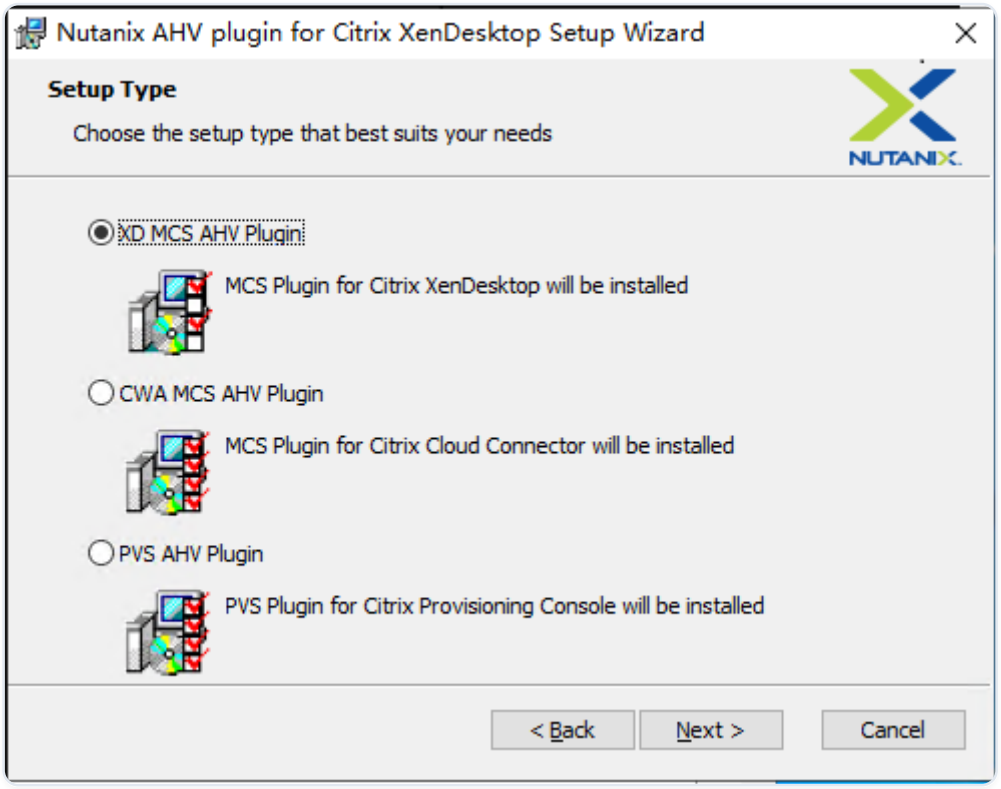
- MCSで作成されたマシンをWindowsドメインに参加させるためにPBISまたはCentrifyを使用している場合は、次のタスクを完了してください。
  - テンプレートマシンで、`/etc/xdl/mcs/mcs.conf`ファイルにPBISまたはCentrifyパッケージのダウンロードパスを構成するか、PBISまたはCentrifyパッケージを直接インストールします。
  - `/opt/Citrix/VDA/sbin/deploymcs.sh`を実行する前に、配下のMCSで作成されたすべてのマシンに対して書き込み権限とパスワードリセット権限を持つ組織単位（OU）を作成します。
  - `/opt/Citrix/VDA/sbin/deploymcs.sh`の実行が完了した後、MCSで作成されたマシンを再起動する前に、展開に応じてDelivery ControllerまたはCitrix Cloud Connectorで`klist -li 0x3e4 purge`を実行します。

(Nutanixのみ) 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 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向けNutanix AHVプラグインをインストールして登録する

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



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

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

手順1c：プラグインのインストール後に次の手順を完了する

- `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0`にNutanix Acropolisフォルダーが作成されていることを確認します。

- 「`"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe" -PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"`」コマンドを実行します。
- オンプレミスのDelivery ControllerでCitrix Host、Citrix Broker、Citrix Machine Creation Servicesを再起動するか、Citrix Cloud 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。

手順2：ホスト接続を作成する

このセクションでは、Azure、AWS、GCP、Nutanix AHV、およびVMware vSphereへのホスト接続の作成について説明します。

- 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GCPへのホスト接続を作成する
- Citrix StudioでNutanixへのホスト接続を作成する
- Citrix StudioでVMwareへのホスト接続を作成する

Citrix StudioでAzureへのホスト接続を作成する

1. Citrix Cloudにサインインします。
2. 左上のメニューで、**［My Services］>［DaaS］**の順に選択します。
3. **［管理］>［完全な構成］から、左ペインで［ホスティング］**を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。

   接続の追加イメージ(/en-us/linux-virtual-delivery-agent/2301/media/add-conn-connection.png)
5. 接続の種類としてMicrosoft Azureを選択します。
6. ウィザードのページに従って操作します。特定のページコンテンツは、選択した接続の種類によって異なります。各ページを完了したら、**［次へ］を選択して［概要］**ページに進みま

す。詳しくは、非ドメイン参加Linux VDAの作成の記事の手順2：ホスト接続を作成するを参照してください。

Citrix StudioでAWSへのホスト接続を作成する

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

   Amazon EC2の選択(/en-us/linux-virtual-delivery-agent/2210/media/choose-amazon-ec2.png)
3. AWSアカウントのAPIキーとシークレットキーを入力し、接続名を入力します。

   アクセスキーペア(/en-us/linux-virtual-delivery-agent/2210/media/access-key-pair.png)

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

   a. AWSサービスにサインインします。
   b. Identity and Access Management (IAM) コンソールに移動します。
   c. 左側のナビゲーションペインで、**ユーザー**を選択します。
   d. 対象ユーザーを選択し、下にスクロールして**セキュリティ認証情報**タブを選択します。
   e. 下にスクロールして**アクセスキーの作成**をクリックします。新しいウィンドウが表示されます。
   f. **.csvファイルのダウンロード**をクリックし、アクセスキーを安全な場所に保存します。
4. ウィザードはページを案内します。特定のページの内容は、選択した接続の種類によって異なります。各ページを完了したら、**概要**ページに到達するまで**次へ**を選択します。

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、以下の手順でGCPへのホスト接続を作成します。

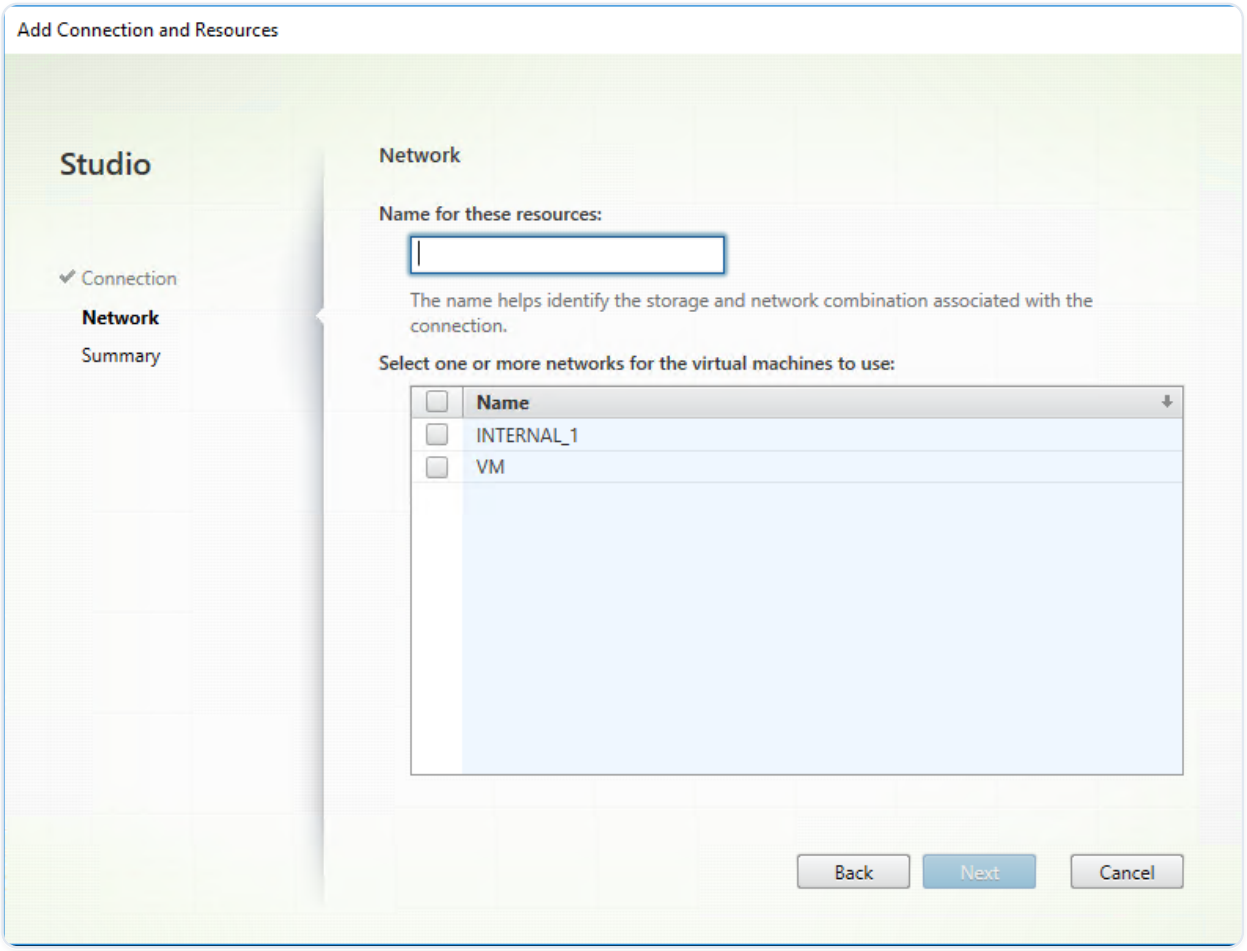
1. Citrix Cloudにサインインします。
2. 左上のメニューで、**My Services > DaaS** を選択します。
3. **Manage > Full Configuration** から、左ペインで **Hosting** を選択します。
4. アクションバーで**「接続とリソースの追加」**を選択します。
5. 接続の種類として **Google Cloud Platform** を選択します。

6. GCPアカウントのサービスアカウントキーをインポートし、接続名を入力します。
7. ウィザードがページを案内します。特定のページの内容は、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、**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、非ドメイン参加Linux VDAの作成 の記事にある 手順2：ホスト接続の作成 を参照してください。

Citrix StudioでNutanixへのホスト接続を作成する

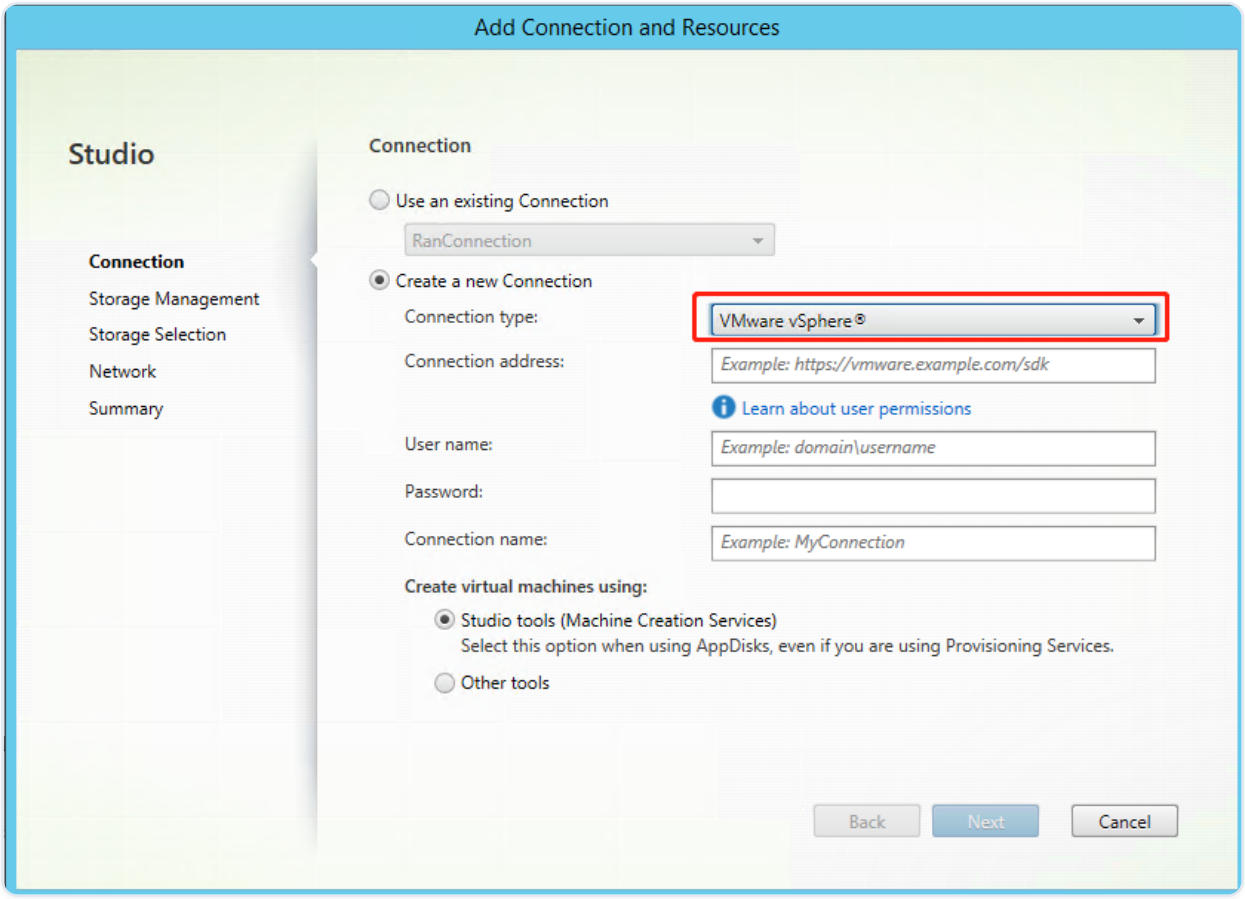
1. オンプレミスのDelivery Controllerの場合、オンプレミスのCitrix Studioで **Configuration > Hosting > Add Connection and Resources** を選択します。クラウドのDelivery Controllerの場合、Citrix CloudのWebベースのStudioコンソールで **Manage > Hosting > Add Connection and Resources** を選択して、Nutanixハイパーバイザーへの接続を作成します。
2. **Add Connection and Resources** ウィザードの **Connection** ページで、接続の種類として Nutanix AHVを選択し、ハイパーバイザーのアドレス、資格情報、および接続名を指定します。**Network** ページで、ユニットのネットワークを選択します。

   たとえば、オンプレミスのCitrix Studioの場合：

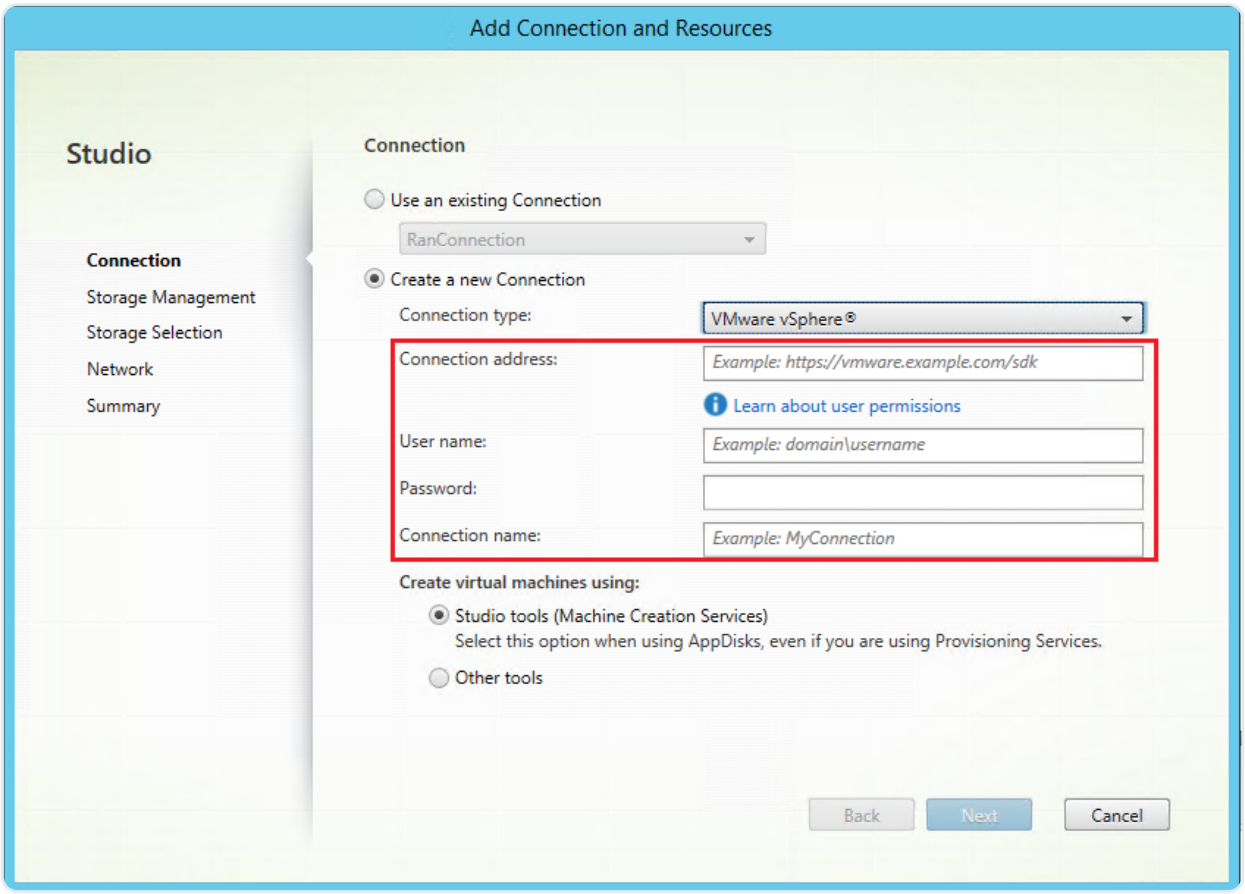


Citrix StudioでVMwareへのホスト接続を作成する

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. Citrix Studioで、**構成 > ホスティング > 接続とリソースの追加** を選択します。
3. 接続の種類としてVMware vSphereを選択します。



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



ステップ 3: マスターイメージを準備する

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

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

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

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

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

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

   RHEL/CentOS/Rocky Linux の場合:

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

   Ubuntu/Debian の場合:

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

   SUSEの場合:

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

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

手順3b：AWS、Azure、およびGCP上のSUSE 15.4の構成を確認する

AWS、Azure、およびGCP上のSUSE 15.4の場合、以下を確認してください。

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

手順3c：Azure、AWS、およびGCP上のUbuntu 18.04向けにcloud-initを構成する

1. VMが再起動または停止されたときにVDAホスト名が保持されるようにするには、次のコマンドを実行します。

   ```
   echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
   ```

   /etc/cloud/cloud.cfgファイルの**system_info**セクションに次の行が存在することを確認します。

   ```
   system_info:

      network:
   ```

```
        renderers: ['netplan', 'eni', 'sysconfig']
```

2. AWSでMCSによって作成されたVMにリモートでアクセスするためにSSHを使用するには、それらのVMにキー名がアタッチされていないため、パスワード認証を有効にします。必要に応じて、次の手順を実行します。

   - `cloud-init`構成ファイル /etc/cloud/cloud.cfg を編集します。**ssh_pwauth: true**の行が存在することを確認します。**set-password**の行とそれに続く行が存在する場合は、削除またはコメントアウトします。

     ```
     users:

     - default
     ```

   - `cloud-init`によって作成されたデフォルトユーザー`ec2-user`または`ubuntu`を使用する予定がある場合は、`passwd`コマンドを使用してユーザーパスワードを変更できます。新しいパスワードは、後でMCSによって作成されたVMにログインするために覚えておいてください。
   - `/etc/ssh/sshd_config`ファイルを編集して、次の行が存在することを確認します。

     ```
     PasswordAuthentication yes
     ```

     ファイルを保存し、`sudo service sshd restart`コマンドを実行します。

手順3d：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

> **注:**

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

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

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

RHEL/CentOS/Rocky Linuxの場合：

注:

- RHELおよびCentOSの場合、Linux VDAをインストールして`deploymcs.sh`を正常に実行する前に、EPELリポジトリをインストールしてください。EPELのインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>`の指示を参照してください。
- RHEL 9.0およびRocky Linux 9.0にLinux VDAをインストールする前に、**libsepol** パッケージをバージョン3.4以降に更新してください。

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

Ubuntu/Debianの場合：

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



apt-get install -f
```

SUSEの場合：

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

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

RHEL 7サーバーの場合：

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

RHEL 7ワークステーションの場合：

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

ステップ3g：（SUSEの場合）ntfs-3gを手動でインストールする

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

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3gパッケージへのパスを入力します。

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

5. ntfs-3gをインストールします。

   ```
   ./configure

   make

   make install
   ```

ステップ3h: 使用するデータベースを指定する

実験的機能として、PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後も、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。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールである場合は、この手順を省略します。

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

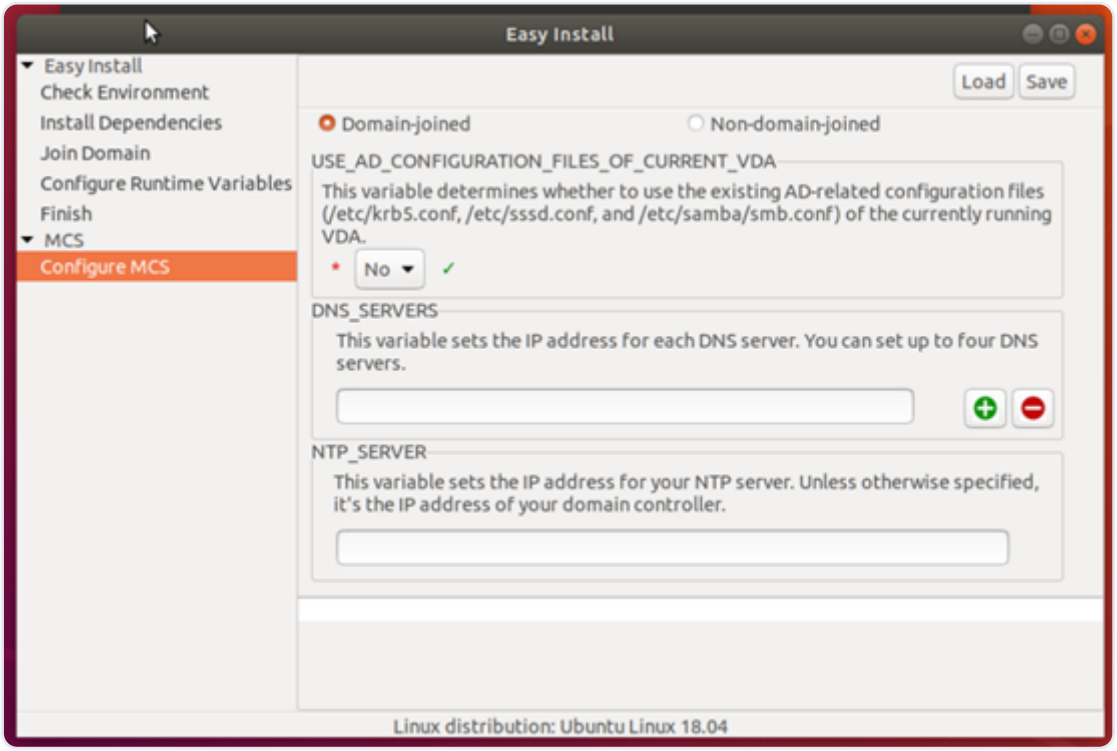
注:

- VDIモードでのみSQLiteを使用することをお勧めします。
- 簡単なインストールとMCSの場合、手動でインストールすることなくSQLiteとPostgreSQLを切り替えることができます。`/etc/xdl/db.conf`で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。
- 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:

> をクリックすると、指定したパスのローカルファイルに変数の設定が保存されます。 をクリックすると、指定したファイルから変数の設定が読み込まれます。

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**

  デフォルトの変数を使用するか、必要に応じて変数をカスタマイズできます（オプション）：

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

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

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

  `START_SERVICE`=**Y | N**

- **ドメインに参加しているシナリオの場合**

  - `Use_AD_Configuration_Files_Of_Current_VDA`: 現在実行中のVDAの既存のAD関連構成ファイル（/etc/krb5.conf、/etc/sssd.conf、および/etc/samba/smb.conf）を使用するかどうかを決定します。Yに設定すると、MCSで作成されたマシンの構成ファイルは、現在実行中のVDAの対応するファイルと同じになります。ただし、`dns`変数と`AD_INTEGRATION`変数は引き続き構成する必要があります。デフォルト値はNです。これは、マスターイメージの構成テンプレートがMCSで作成されたマシンの構成ファイルを決定することを意味します。
  - `dns`: 各DNSサーバーのIPアドレスを設定します。最大4つのDNSサーバーを設定できます。
  - `NTP_SERVER`: NTPサーバーのIPアドレスを設定します。特に指定がない限り、ドメインコントローラーのIPアドレスです。
  - `WORKGROUP`: ADで構成したNetBIOS名（大文字と小文字を区別）にワークグループ名を設定します。それ以外の場合、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`: Winbind、SSSD、PBIS、またはCentrifyを設定します。MSCがサポートするLinuxディストリビューションとドメイン参加方法の対応表については、この記事のサポートされ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パスを設定します。この値は、`AD_INTEGRATION`変数をPBISに設定した場合にのみ有効になります。
- `UPDATE_MACHINE_PW`: マシンアカウントのパスワードの自動更新を有効または無効にします。詳しくは、「マシンアカウントのパスワードの自動更新」を参照してください。
- Linux VDA構成変数:

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

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

  `VDA_PORT`=**port-number** `REGISTER_SERVICE`=**Y | N**

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

  `VDI_MODE`=**Y | N** `SITE_NAME`=**dns-site-name | '<none\>'**

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

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

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

手順3j: MCSのレジストリ値を書き込むまたは更新する

テンプレートマシンで、必要に応じてレジストリ値を書き込むまたは更新するためのコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加します。この操作により、MCS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。

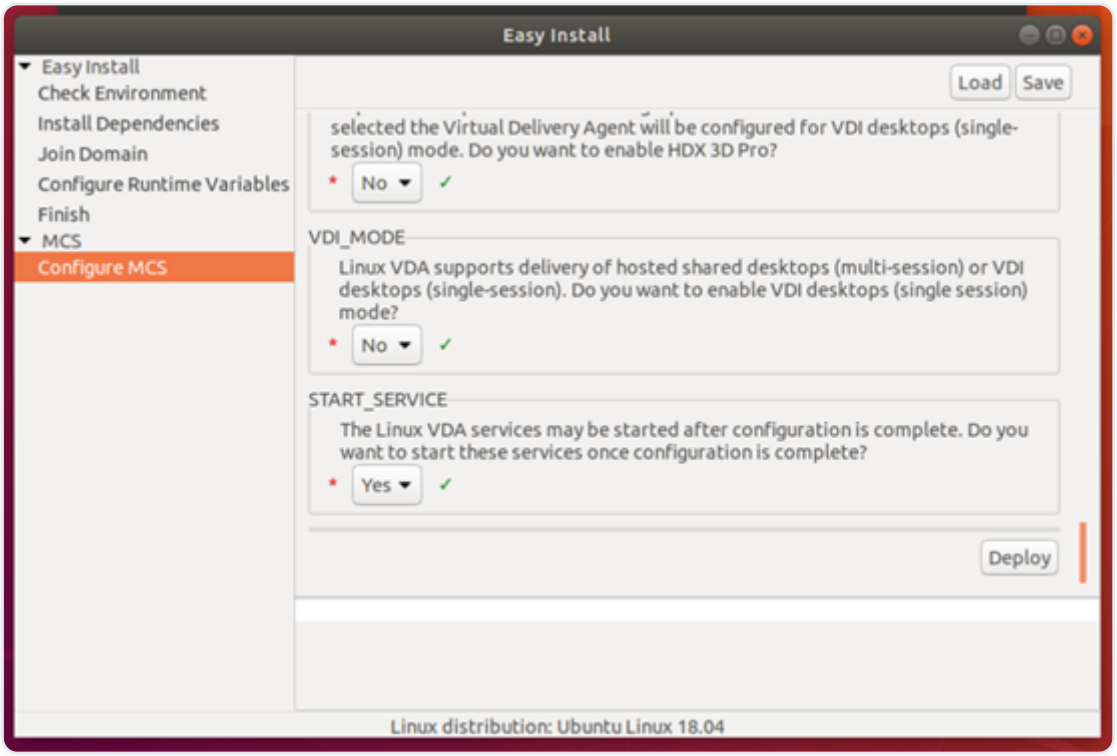
たとえば、レジストリ値を書き込むまたは更新するために、次のコマンドラインを`/etc/xdl/mcs/mcs_local_setting.reg`ファイルに追加できます。

```
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"
```

手順3k: 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. （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、`/etc/xdl/ad_join/centrify_krb5.conf.tmpl`と`/etc/xdl/ad_join/centrify_smb.conf.tmpl`テンプレートを更新します。

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

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

   - **(Citrix Hypervisor、GCP、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てください。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるには、この名前が必要です。

     テンプレートVMの停止済み電源ステータス(/en-us/linux-virtual-delivery-agent/2301/media/stopped-deallocated2.png)
   - **(AWSの場合)** テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が**停止済み**であることを確認してください。テンプレートVMを右クリックし、**イメージ > イメージの作成**を選択します。必要な情報を入力し、設定を行います。**イメージの作成**をクリックします。

     EBSイメージの作成中(/en-us/linux-virtual-delivery-agent/2301/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を再起動する前に次のコマンドを実行します。

```
nmcli dev connect eth0

service NetworkManager restart
```

ステップ4: マシンカタログの作成

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成する際に、マスターイメージを選択します。以下に例を示します。

テンプレートVMのVHDの検索(/en-us/linux-virtual-delivery-agent/2301/media/find-vhd-of-template-vm2.png)

マスターイメージの選択(/en-us/linux-virtual-delivery-agent/2301/media/choose-master-image-from-the-list.png)

Nutanix固有の**コンテナ**ページで、以前にテンプレートVMに指定したコンテナを選択します。**マスターイメージ**ページで、イメージスナップショットを選択します。**仮想マシン**ページで、仮想CPUの数とvCPUあたりのコア数を確認します。

> **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パワーオンします。この方法により、作成プロセスを続行できます。

必要に応じて、その他の構成タスクを実行します。詳しくは、「Studioを使用してマシンカタログを作成する(/ja-jp/citrix-virtual-apps-desktops/install-configure/machine-catalogs-create.html#create-a-machine-catalog-using-web-studio)」を参照してください。

ステップ5：デリバリーグループを作成する

デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。デリバリーグループは、どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳しくは、「デリバリーグループを作成する(/ja-jp/citrix-virtual-apps-desktops/install-configure/delivery-groups-create.html)」を参照してください。

MCSを使用してLinux VDAをアップグレードする

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

1. Linux VDAを現在のリリースにアップグレードする前に、.NET Runtime 6.0がインストールされていることを確認してください。
2. テンプレートマシンでLinux VDAをアップグレードします。

注:

自動ソフトウェア更新をスケジュールするには、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、テンプレートマシンのetc/xdl/mcs/mcs_local_setting.regファイルにコマンドラインを追加します。 たとえば、次のコマンドラインを追加できます。

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



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



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

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

RHEL 7およびCentOS 7の場合：

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

RHEL 8.xおよびRocky Linux 8.xの場合：

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

RHEL 9.0およびRocky Linux 9.0の場合：

注:

RHEL 9.0およびRocky Linux 9.0でLinux VDAをアップグレードする前に、**libsepol**パッケージをバージョン3.4以降に更新してください。

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

SUSEの場合:

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

Ubuntu 18.04の場合:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu18.04_amd64.deb
```

Ubuntu 20.04の場合:

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

Ubuntu 22.04の場合:

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

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

マシンアカウントパスワードの更新を自動化する

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

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

   `UPDATE_MACHINE_PW="Y"`

2. /opt/Citrix/VDA/sbin/deploymcs.shの実行後、/etc/cron.d/mcs_update_password_cronjobを開いて更新時間と頻度を設定します。デフォルト設定では、毎週日曜日の午前2時30分にマシンアカウントのパスワードが更新されます。

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)`

このエラーを解消するには、チケットキャッシュを定期的にクリアします。すべてのDelivery Controllerまたはドメインコントローラーでキャッシュクリーンアップタスクをスケジュールできます。

MCSで作成されたVMでFASを有効にする

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

	ウィンバインド	SSSD	セントリファイ	PBIS
RHEL 9.0	はい	いいえ	いいえ	いいえ
RHEL 8.x	はい	いいえ	いいえ	はい

	ウィンバインド	SSSD	セントリファイ	PBIS
ロッキー Linux 9.0	はい	いいえ	いいえ	いいえ
ロッキー Linux 8.x	はい	いいえ	いいえ	いいえ
RHEL 7、CentOS 7	はい	はい	いいえ	はい
ウブントゥ 22.04、ウブントゥ 20.04、ウブントゥ 18.04	はい	いいえ	いいえ	いいえ
デビアン 11.3	はい	いいえ	いいえ	いいえ
SUSE 15.4	はい	いいえ	いいえ	いいえ

テンプレートVMでマスターイメージを準備する際にFASを有効にする

1. ルートCA証明書をインポートします。

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

2. ctxfascfg.shを実行します。詳細については、「ctxfascfg.shの実行」を参照してください。
3. `/etc/xdl/mcs/mcs.conf`で変数を設定します。

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

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

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

MCSで作成されたVMでFASを有効にする

前述のようにテンプレートマシンでFASが有効になっていない場合、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. `FAS_LIST` 変数をFASサーバーアドレスに設定します。
   c. `VDI_MODE` など、必要に応じて他の変数を設定します。

2. ルートCA証明書をインポートします。

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

3. `/opt/Citrix/VDA/sbin/ctxfascfg.sh` スクリプトを実行します。詳細については、ctxfascfg.shの実行を参照してください。

Citrix Provisioning™ を使用した Linux VDA の作成

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™ 環境で Linux 仮想デスクトップを直接プロビジョニングできます。

以下の Linux ディストリビューションがサポートされています:

- ウブントゥ 22.04
- ウブントゥ 20.04

- ウブントゥ 18.04
- RHEL 8.6
- ロッキー リナックス 8.6
- RHEL 8.4
- RHEL 7.9

重要:

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

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

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

Citrix DaaS Standard for Azure (旧称 Citrix Virtual Apps and Desktops Standard for Azure) で、ドメイン参加済みおよび非ドメイン参加済みの両方の Linux VDA を作成し、Microsoft Azure から任意のデバイスに仮想アプリとデスクトップを配信できます。詳しくは、「Citrix DaaS Standard for Azure」(/ja-jp/citrix-daas-azure.html#domain-joined-and-non-domain-joined)を参照してください。

サポートされているLinuxディストリビューション

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

- RHEL 9.0
- RHEL 8.7

- RHEL 8.6
- RHEL 8.4
- ロッキーリナックス 9.0
- ロッキーリナックス 8.7
- ロッキーリナックス 8.6
- SUSE 15.4
- ウブントゥ 22.04
- ウブントゥ 20.04
- ウブントゥ 18.04

手順 1：Azure でマスターイメージを準備する

注:

自動ソフトウェアアップデートをスケジュールするために、Linux VDA セルフアップデート機能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。 例えば、次のコマンドラインを追加できます。

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



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



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



create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate" -t
```

```
"REG_SZ" -v "CaCertificate" -d "&lt;Local-Certificate-Path-of-
PortalAzureCom>" --force
```

1. Azureで、サポートされているディストリビューションのLinux VMを作成します。
2. 必要に応じて、Linux VMにデスクトップ環境をインストールします。
3. VMに、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>` の指示に従って.NET Runtime 6.0をインストールします。
4. (Ubuntuのみ) `source /etc/network/interfaces.d/*` 行を `/etc/network/interfaces` ファイルに追加します。
5. (Ubuntuのみ) `/run/systemd/resolve/stub-resolv.conf` ではなく、`/etc/resolv.conf` を `/run/systemd/resolve/resolv.conf` に向けます。

   ```
   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。

   実験的な機能として、PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後に、SQLiteとPostgreSQLを切り替えることもできます。これを行うには、次の手順を完了します。

   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。

> - 簡単なインストールとMCSの場合、SQLiteとPostgreSQLを手動でインストールすることなく切り替えることができます。`/etc/xdl/db.conf` を介して別途指定されない限り、Linux VDAはデフォルトでPostgreSQLを使用します。
> - `/etc/xdl/db.conf` を使用して、PostgreSQLのポート番号を構成することもできます。

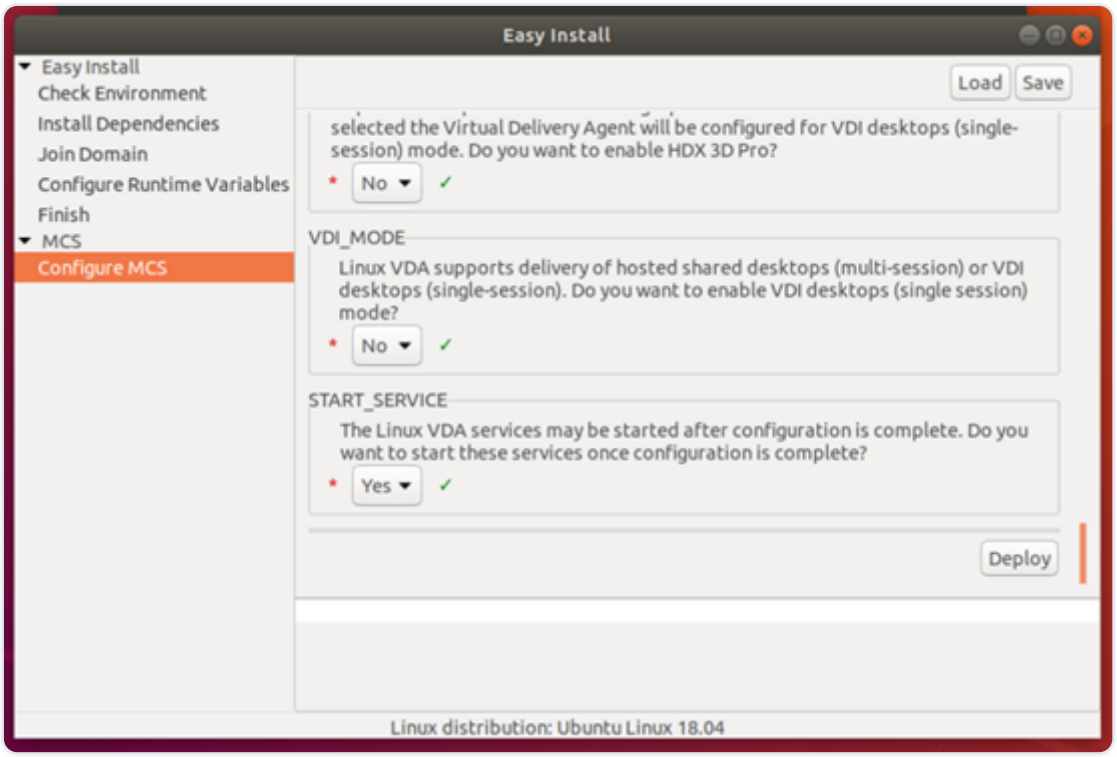
8. MCS変数を変更します。

   MCS変数を構成する方法は2つあります。

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

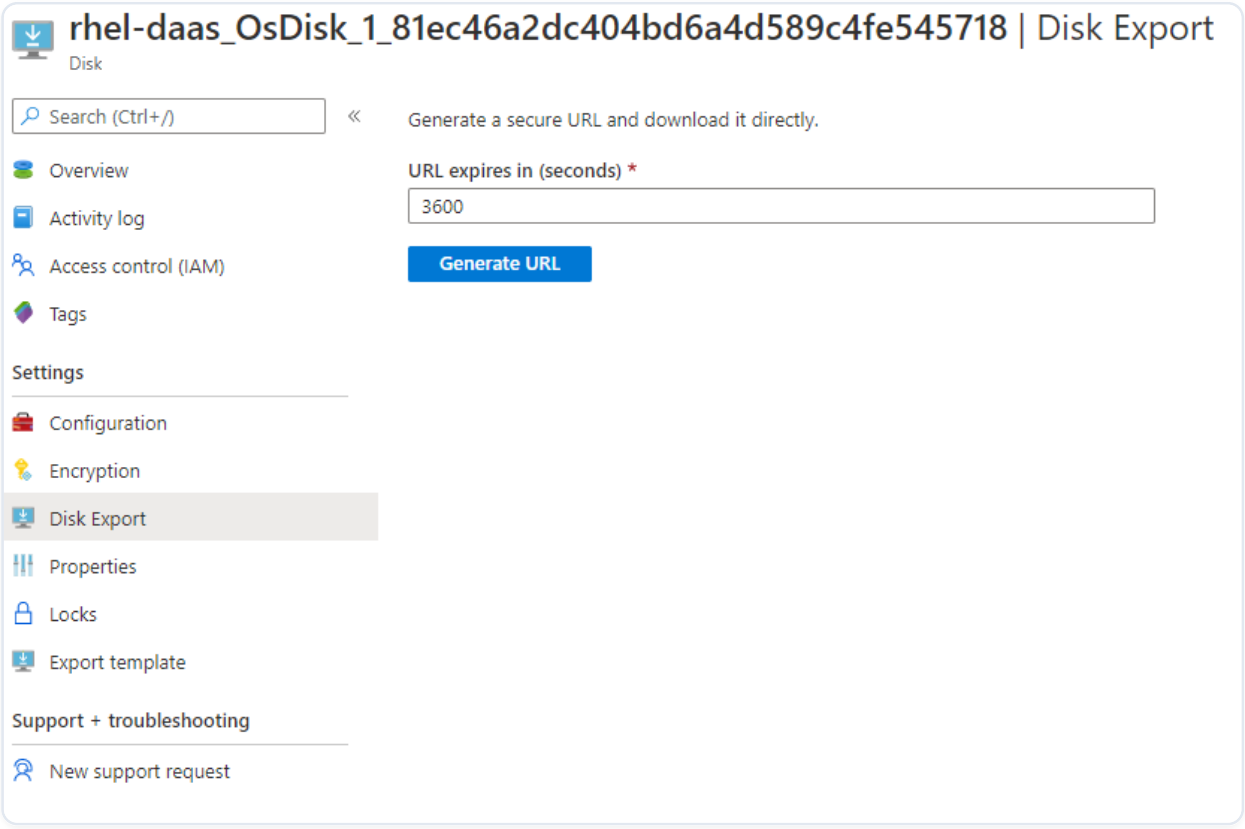
   > **注:**
   >
   > `dns` 変数は指定しないでください。 マシンカタログの作成時に**Static**または**Random**タイプを選択する場合は、`VDI_MODE=Y` を設定します。

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



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

9. Azureで、VMを停止（または割り当て解除）します。**Disk Export**をクリックして、他のVMを作成するためのマスターイメージとして使用できる仮想ハードディスク（VHD）ファイルのSAS URLを生成します。



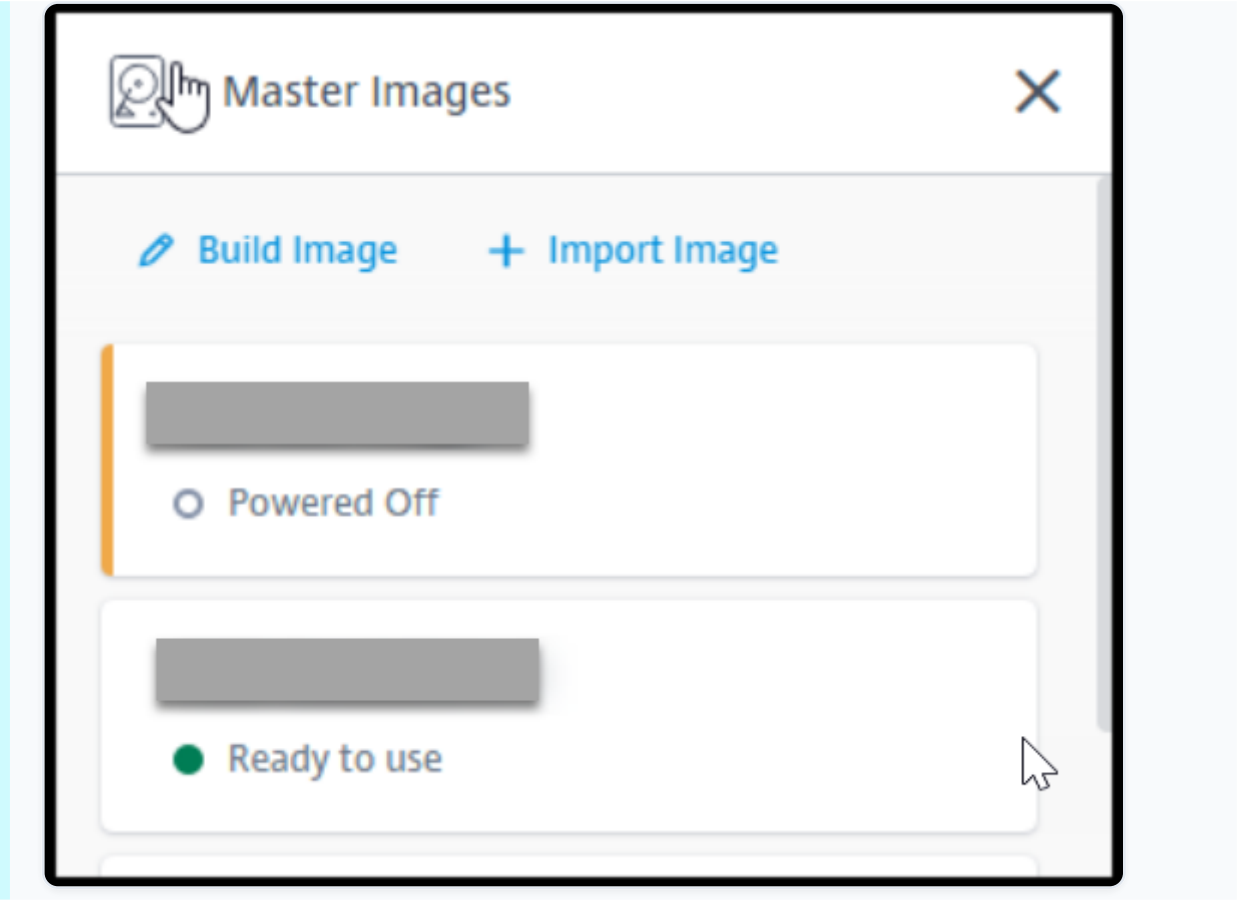
10. （オプション）マスターイメージでグループポリシー設定を行います。`ctxreg`ツールを使用してグループポリシー設定を行うことができます。たとえば、次のコマンドはPDF印刷の**Auto-create PDF Universal Printer**ポリシーを有効にします。

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

ステップ2：Azureからマスターイメージをインポートする

1. Manageダッシュボードから、右側の**Master Images**を展開します。Citrixが提供するマスターイメージと、作成およびインポートしたイメージが表示されます。

> **ヒント:**
>
> このサービスの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した後、Citrix Cloud™にサインインして**Managed Desktops**サービスを選択すると、**Manage**ダッシュボードが自動的に起動します。



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

   Azureからイメージをインポートする画像(/en-us/linux-virtual-delivery-agent/2301/media/import-an-image-from-azure.png)

4. ウィザードの指示に従って、マスターイメージのインポートを完了します。

ステップ3：マシンカタログを作成する

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

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

Linux VDA を手動でインストールする

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

- アマゾン Linux 2、CentOS、RHEL、およびロッキー Linux
- SUSE](/ja-jp/linux-virtual-delivery-agent/2301/installation-overview/manual-installation-overview/suse.html)
- ウブントゥ
- デビアン

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

> **重要:**

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

ステップ1：構成情報とLinuxマシンを準備する

ステップ1a：ネットワーク構成を確認する

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

ステップ1b：ホスト名を設定する

マシンのホスト名が正しく報告されるように、**/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、FQDNとホスト名を最初の2つのエントリとして含むように変更します。

`127.0.0.1 hostname-fqdn hostname localhost localhost.localdomain localhost4 localhost4.localdomain4`

例:

`127.0.0.1 vda01.example.com vda01 localhost localhost.localdomain localhost4 localhost4.localdomain4`

ファイル内の他のエントリから、**hostname-fqdn** または **hostname** への他の参照をすべて削除します。

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

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

ステップ1d：ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: 名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

ステップ1f: クロック同期の構成

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

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

CHRONYサービスの構成

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

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


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

一般的な展開では、パブリックNTPプールサーバーから直接ではなく、ローカルのドメインコントローラーから時刻を同期します。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

ループバックIPアドレス、localhost、およびパブリックサーバーの***.pool.ntp.org**エントリを含む、リストされている他の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 /sbin/service chronyd restart
```

ステップ1g: PulseAudioのインストール (RHEL 9.0およびRocky Linux 9.0のみ)

pulseaudioをインストールするには、次のコマンドを実行します。

```
sudo yum -y install pulseaudio --allowerasing
```

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

```
autospawn = yes
```

ステップ 1h: OpenJDK 11 をインストールする

Linux VDA には OpenJDK 11 が必要です。

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

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

正しいバージョンを確認します。

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

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

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

ステップ 1i: 使用するデータベースをインストールして指定する

実験的な機能として、PostgreSQL に加えて SQLite を使用できます。Linux VDA パッケージのインストール後に `/etc/xdl/db.conf` を編集することで、SQLite と PostgreSQL を切り替えることもできます。手動インストールの場合、それらを切り替える前に SQLite と PostgreSQL を手動でインストールする必要があります。

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

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

POSTGRESQL をインストールする

Linux VDA には PostgreSQL が必要です。

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

PostgreSQLをインストールするには、次のコマンドを実行します。

```
sudo yum -y install postgresql-server



sudo yum -y install postgresql-jdbc
```

RHEL 8.xおよびRHEL 9.0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、次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

次のコマンドを使用してPostgreSQLのバージョンを確認します。

```
psql --version
```

（RHEL 7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をインストールする

SQLiteをインストールするには、次のコマンドを実行します。

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

使用するデータベースを指定する

SQLite、PostgreSQL、またはその両方をインストールした後、Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集して、使用するデータベースを指定できます。これを行うには、次の手順を実行します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略してください。
2. 使用するデータベースを指定するために`/etc/xdl/db.conf`を編集します。
3. `ctxsetup.sh`を実行します。

注:

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

ステップ2：ハイパーバイザーを準備する

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。

Citrix Hypervisor™での時刻同期の修正

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

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

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration Servicesがインストールされ ているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックが正確に保たれるようにするには、この機能をNTPサービスと併せて有効にする必要があります。

管理オペレーティングシステムから:

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

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

ESX および ESXi で時刻同期を修正する

VMware の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP およびハイパーバイザーとの問題が発生します。どちらもシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

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

1. vSphere Client を開きます。
2. Linux VM の設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。

4. **VMware Tools**を選択します。
5. **詳細設定**ボックスで、**ゲストの時刻をホストと同期**のチェックを外します。

ステップ 3：Linux VM を Windows ドメインに追加する

Linux マシンを Active Directory (AD) ドメインに追加するには、次の方法があります：

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

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ ウィンバインド

必要なパッケージをインストールまたは更新します。

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

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

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

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

マシンの起動時にWINBINDデーモンを開始するように有効にする

Winbindデーモンは、マシンの起動時に開始するように構成する必要があります。

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

WINBIND認証を構成する

Winbindを使用して、Kerberos認証用にマシンを構成します。

1. 次のコマンドを実行します。

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

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

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

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

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

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

   `--enablekrb5kdcdns --enablekrb5realmdns`

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

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

   `kerberos method = secrets and keytab`

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

3. (RHEL 8.x/9.0 および Rocky Linux 8.x/9.0 のみ) **/etc/krb5.conf** を開き、`[libdefaults]`、`[realms]`、および `[domain_realm]` セクションの下にエントリを追加します。

   `[libdefaults]` セクションの下:

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

   `default_realm = REALM` `dns_lookup_kdc = true`

`[realms]` セクションの下:

`REALM = {`

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

`[domain_realm]` セクションの下:

`realm = REALM`

`.realm = REALM`

Linux VDA は、Delivery Controller で認証および登録するためにシステムキータブファイル /etc/krb5.keytab を必要とします。以前の Kerberos メソッド設定により、マシンが最初にドメインに参加したときに Winbind がシステムキータブファイルを作成するよう強制されます。

WINDOWS ドメインへの参加

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

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

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

Amazon Linux 2 および RHEL 7 の場合:

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

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

WINBIND 用に PAM を構成する

デフォルトでは、Winbind PAM モジュール (pam_winbind) の構成では、Kerberos チケットのキャッシュとホームディレクトリの作成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。これらの変更を適用するには、Winbind デーモンを再起動する必要があります。

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

ヒント:

`winbind` デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

/etc/krb5.conf を開き、\[libdefaults\] セクションの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 9.0 および Rocky Linux 9.0 の場合、次のコマンドを実行して、`pam_winbind` がルートディレクトリの所有権を変更するのを防ぎ、Winbind の SELinux の問題を解決します。

```
usermod -d /nonexistent nobody



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。

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

```
sudo net ads testjoin
```

次のコマンドを実行して、追加のドメインおよびコンピューターオブジェクト情報を確認します。

```
sudo net ads info
```

KERBEROS 構成を確認する

Linux VDA で使用するために Kerberos が正しく構成されていることを確認するには、システムの keytab ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

次のコマンドを使用して、マシンのアカウント詳細を調べてください。

```
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、期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

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

クエスト認証サービス

ドメインコントローラーでのQUESTの構成

Active Directoryドメインコントローラーに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-enabled**」をチェックします。
4. **Primary GID Number** を実際のドメインユーザーグループのグループIDに設定します。

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

LINUX VDAでのQUESTの構成

SELinuxポリシーの適用を回避する
デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用は、Questが使用するUnixドメインソケットIPCメカニズムと干渉し、ドメインユーザーのログオンを妨げます。

この問題を回避する便利な方法は、SELinuxを無効にすることです。rootユーザーとして、**/etc/selinux/config** を編集し、**SELinux** の設定を変更します。

`SELINUX=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、完全なロックアウトを引き起こす可能性があります。

VASデーモンの構成

Kerberosチケットの自動更新は有効にし、切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

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



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

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

PAMとNSSの構成

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

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



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

WINDOWSドメインへの参加

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

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

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

ドメインメンバーシップの確認

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

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用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。

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**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、次の手順を実行します。

```
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動かない場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all

adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/2301/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、このセクションの手順に従ってください。このセクションには、Linux VDAマシンをWindowsドメインに参加させる手順と、Kerberos認証を構成するためのガイダンスが含まれています。

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

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

ドメインに参加し、ホストキータブを作成する

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

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

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

  必要なパッケージをインストールまたは更新します。

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

  適切に構成されたファイルを持つ Linux クライアントで:

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

  Samba および Kerberos 認証用にマシンを構成します。

  ```
  sudo authconfig --smbsecurity=ads --smbworkgroup=domain --smbrealm=REALM --krb5realm=REALM --krb5kdc=fqdn-of-domain-controller --update
  ```

  ここで、**REALM** は大文字の Kerberos レルム名であり、**domain** は Active Directory ドメインの短い NetBIOS 名です。

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

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

  `--enablekrb5kdcdns --enablekrb5realmdns`

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

  `kerberos method = secrets and keytab`

`winbind offline logon = no`

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

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

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

- **Adcli (RHEL 8.x/9.0 および Rocky Linux 8.x/9.0):**

必要なパッケージをインストールまたは更新します。

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

SambaおよびKerberos認証用にマシンを構成します。

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

/etc/krb5.confを開き、セクションとセクションの下にエントリを追加します。

セクションの下:

`REALM = {`

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

セクションの下:

`realm = REALM` `.realm = REALM`

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

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

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

SSSDをセットアップする

SSSDのセットアップは、次の手順で構成されます。

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

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

```
[sssd]
config_file_version = 2
domains = ad.example.com
services = nss, pam

[domain/ad.example.com]
# Uncomment if you need offline logins
# cache_credentials = true

id_provider = ad
auth_provider = ad
access_provider = ad
ldap_id_mapping = true
ldap_schema = ad

# Should be specified as the lower-case version of the long version of the Active Directory domain.
ad_domain = ad.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.example.com、**server.ad.example.com**を対応する値に置き換えてください。詳細については、「sssd-ad(5) - Linux man page」を参照してください。

(RHEL 8.x/9.0 および Rocky Linux 8.x/9.0 のみ) **/etc/sssd/sssd.conf** を開き、 セクションに次のエントリを追加します。

```
ad_gpo_access_control = permissive
```

`full_name_format = %2$s\%1$s` `fallback_homedir = /home/%d/%u`

`# Kerberos settings` `krb5_ccachedir = /tmp`

`krb5_ccname_template = FILE:%d/krb5cc_%U`

sssd.conf のファイルの所有権とアクセス許可を設定します。

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

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

SSSD を有効にする

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

SSSD を有効にするには、次のコマンドを実行します。

```
sudo systemctl restart sssd

sudo systemctl enable sssd.service

sudo chkconfig sssd on
```

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

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

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



sudo service sssd start



sudo chkconfig sssd on
```

KERBEROS 構成を確認する

システム **keytab** ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

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

```
sudo kinit –k MACHINE\$@REALM
```

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

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

```
sudo klist
```

ユーザー認証を確認する

getent コマンドを使用して、ログオン形式がサポートされており、NSS が機能していることを確認します。

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

DOMAIN パラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

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

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

```
sudo ssh localhost –l DOMAIN\\username
```

```
id -u
```

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

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

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

```
klist
```

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

PBIS

必要な PBIS パッケージをダウンロードする

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/
pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBIS インストールスクリプトを実行可能にする

```
chmod +x pbis-open-9.1.0.551.linux.x86_64.rpm.sh
```

PBIS インストールスクリプトを実行する

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

WINDOWS ドメインに参加する

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
```

ユーザーは、コンピューターをActive Directoryドメインに追加する権限を持つドメインユーザーです。**ドメイン名**は、たとえばexample.comのようなドメインのDNS名です。

注： Bashをデフォルトシェルとして設定するには、**/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およびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

手順4：.NET Runtime 6.0のインストール

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

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

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

手順5：Linux VDAパッケージのダウンロード

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

   Linux VDAパッケージの整合性を検証するには、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>
   ```

   ```
   ```

ステップ6：Linux VDAをインストールする

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

ステップ6a：新規インストールを実行する

1. （オプション）古いバージョンをアンインストールする

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

   a. Linux VDAサービスを停止します。

   ```
   sudo /sbin/service ctxvda stop
   ```

```
sudo /sbin/service ctxhdx stop
```

注:

`ctxvda`および`ctxhdx`サービスを停止する前に、**service ctxmonitorservice stop**コマンドを実行してモニターサービスデーモンを停止します。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。

b. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

注:

コマンドを実行するには、フルパスが必要です。または、**/opt/Citrix/VDA/sbin**と**/opt/Citrix/VDA/bin**をシステムパスに追加することもできます。

2. Linux VDAパッケージをダウンロードする

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

3. Linux VDA をインストールする

注:

- CentOS、RHEL、および Rocky Linux の場合、Linux VDA を正常にインストールする前に EPEL リポジトリをインストールしてください。EPEL のインストール方法については、`<https://docs.fedoraproject.org/ja-jp/epel/>` の手順を参照してください。
- RHEL 9.0 および Rocky Linux 9.0 に Linux VDA をインストールする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

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

  Amazon Linux 2 の場合:

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

  RHEL 9.0 および Rocky Linux 9.0 の場合:

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

  RHEL 8.x および Rocky Linux 8.x の場合:

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

  CentOS 7 および RHEL 7 の場合:

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

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

  Amazon Linux 2 の場合:

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

  RHEL 9.0 および Rocky Linux 9.0:

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

  RHEL 8.x および Rocky Linux 8.x の場合:

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

  CentOS 7 および RHEL 7 の場合:

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

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

```
java-11-openjdk >= 11

 icoutils >= 0.32

 firewalld >= 0.6.3

 policycoreutils-python >= 2.8.9

 policycoreutils-python-utils >= 2.8

 python3-policycoreutils >= 2.8

 dbus >= 1.12.8

 dbus-common >= 1.12.8

 dbus-daemon >= 1.12.8

 dbus-tools >= 1.12.8

 dbus-x11 >= 1.12.8
```

```
xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12

libXrandr >= 1.5.1

libXtst >= 1.2.3

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6
```

```
gawk >= 4.2

sed >= 4.5

cups >= 1.6.0

foomatic-filters >= 4.0.9

cups-filters >= 1.20.0

ghostscript >= 9.25

libxml2 >= 2.9

libmspack >= 0.7

krb5-workstation >= 1.13

ibus >= 1.5

nss-tools >= 3.44.0
```

```
gperftools-libs >= 2.4

cyrus-sasl-gssapi >= 2.1

python3 >= 3.6~

qt5-qtbase >= 5.5~

qt5-qtbase-gui >= 5.5~

qrencode-libs >= 3.4.4

imlib2 >= 1.4.9
```

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

```
java-11-openjdk >= 11

icoutils >= 0.32

firewalld >= 0.6.3
```

```
policycoreutils-python >= 2.8.9

policycoreutils-python-utils >= 2.8

python3-policycoreutils >= 2.8

dbus >= 1.12.8

dbus-common >= 1.12.8

dbus-daemon >= 1.12.8

dbus-tools >= 1.12.8

dbus-x11 >= 1.12.8

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12
```

```
libXrandr >= 1.5.1

libXtst >= 1.2.3

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user >= 2.32.1

xorg-x11-utils >= 7.5

bash >= 4.3

findutils >= 4.6

gawk >= 4.2

sed >= 4.5

cups >= 1.6.0
```

```
foomatic-filters >= 4.0.9


cups-filters >= 1.20.0


ghostscript >= 9.25


libxml2 >= 2.9


libmspack >= 0.7


krb5-workstation >= 1.13


ibus >= 1.5


nss-tools >= 3.44.0


gperftools-libs >= 2.4


cyrus-sasl-gssapi >= 2.1


python3 >= 3.6~
```

```
qt5-qtbase >= 5.5~


qt5-qtbase-gui >= 5.5~


qrencode-libs >= 3.4.4


imlib2 >= 1.4.9
```

CentOS 7 および RHEL 7 の RPM 依存関係リスト:

```
java-11-openjdk >= 11


ImageMagick >= 6.7.8.9


firewalld >= 0.3.9


policycoreutils-python >= 2.0.83


dbus >= 1.6.12


dbus-x11 >= 1.6.12


xorg-x11-server-utils >= 7.7
```

```
xorg-x11-xinit >= 1.3.2

xorg-x11-server-Xorg >= 1.20.4

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

pam >= 1.1.8

util-linux >= 2.23.2

bash >= 4.2

findutils >= 4.5

gawk >= 4.0

sed >= 4.2
```

```
cups >= 1.6.0

foomatic-filters >= 4.0.9

libxml2 >= 2.9

libmspack >= 0.5

ibus >= 1.5

cyrus-sasl-gssapi >= 2.1

python3 >= 3.6~

gperftools-libs >= 2.4

nss-tools >= 3.44.0

qt5-qtbase >= 5.5~

qt5-qtbase >= 5.5~
```

```
imlib2 >= 1.4.5
```

Amazon Linux 2 の RPM 依存関係リスト:

```
java-11-openjdk >= 11

ImageMagick >= 6.7.8.9

firewalld >= 0.3.9

policycoreutils-python >= 2.0.83

dbus >= 1.6.12

dbus-x11 >= 1.6.12

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

xorg-x11-server-Xorg >= 1.20.4

libXpm >= 3.5.10
```

```
libXrandr >= 1.4.1

libXtst >= 1.2.2

pam >= 1.1.8

util-linux >= 2.23.2

bash >= 4.2

findutils >= 4.5

gawk >= 4.0

sed >= 4.2

cups >= 1.6.0

foomatic-filters >= 4.0.9

libxml2 >= 2.9
```

```
libmspack >= 0.5

ibus >= 1.5

cyrus-sasl-gssapi >= 2.1

gperftools-libs >= 2.4

nss-tools >= 3.44.0

qt5-qtbase >= 5.5~

qrencode-libs >= 3.4.1

imlib2 >= 1.4.5
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

RHEL 7.x に Linux VDA をインストールした後、`sudo yum install -y python-websockify x11vnc` コマンドを実行します。これは、セッションシャドウ機能を使用

するために、`python-websockify` と `x11vnc` を手動でインストールすることを目的としています。詳細については、「セッションのシャドウ」を参照してください。

手順 6b: 既存のインストールをアップグレードする（オプション）

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

注:

- 既存のインストールをアップグレードすると、/etc/xdl の下の構成ファイルが上書きされます。アップグレードを実行する前に、ファイルをバックアップしてください。
- RHEL 9.0 および Rocky Linux 9.0 で Linux VDA をアップグレードする前に、**libsepol** パッケージをバージョン 3.4 以降に更新してください。

- `Yum` を使用してソフトウェアをアップグレードするには：

  Amazon Linux 2 の場合:

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

  RHEL 9.0 および Rocky Linux 9.0 の場合:

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

  RHEL 8.x および Rocky Linux 8.x の場合:

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

  CentOS 7 および RHEL 7 の場合:

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

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

  Amazon Linux 2 の場合:

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

RHEL 9.0 および Rocky Linux 9.0 の場合:

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

RHEL 8.x および Rocky Linux 8.x の場合:

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

CentOS 7 および RHEL 7 の場合:

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

注:

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

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

重要:

ソフトウェアのアップグレード後、Linux VDA マシンを再起動します。

手順 7: NVIDIA GRID ドライバーをインストールする

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

注:

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

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

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

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

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

```
yum install gcc



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



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

5. NVIDIA GRIDドライバーをインストールするには、(https://access.redhat.com/solutions/1155663)のの手順に従ってください。

注:

> GPUドライバーのインストール中、各質問に対してデフォルト（「いいえ」）を選択してください。

> **重要:**

> GPUパススルーが有効になると、Linux VMはXenCenter経由でアクセスできなくなります。SSHを使用して接続してください。

> NVIDIA smiコードスニペット(/en-us/linux-virtual-delivery-agent/2301/media/linux-vda-nvidia-smi-code-snippet.png)

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

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

ステップ8：Linux VDAを構成する

パッケージのインストール後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

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

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションを環境変数で指定します。必要なすべての変数が存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerに登録するために、スペースで区切られたDelivery Controllerの完全修飾ドメイン名 (FQDN) のリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。デフォルトではポート80です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux VDAサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDAサービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux仮想デスクトップの場合、必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \| 5** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ Winbind
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、グラフィックを多用するアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーション技術のセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するかを決定します。HDX 3D Pro環境では、この変数をYに設定します。この変数はデフォルトでNに設定されています。

- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。例: ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。LDAP ポート番号を 389 と指定すると、Linux VDA は指定されたドメイン内の各 LDAP サーバーをポーリングモードで照会します。ポリシーの数が x で LDAP サーバーの数が y の場合、Linux VDA は合計で X かける Y のクエリを実行します。ポーリング時間がしきい値を超えると、セッションログオンが失敗する可能性があります。LDAP クエリを高速化するには、ドメインコントローラーで **Global Catalog** を有効にし、関連する LDAP ポート番号を 3268 と指定します。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例: DC=mycompany,DC=com）を介して LDAP を照会します。検索パフォーマンスを向上させるには、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。この変数は、デフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりに FAS サーバーのセミコロン区切りリストを提供できます。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'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするために .NET Runtime 6.0 をインストールするパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用する GNOME、GNOME Classic、または MATE デスクトップ環境を指定します。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

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

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

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

- **MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **GNOME Classic デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

b. ターゲットセッションユーザーと 700 ファイル権限を共有します。

バージョン 2209 以降、セッションユーザーはデスクトップ環境をカスタマイズできます。この機能を有効にするには、VDA に切り替え可能なデスクトップ環境を事前にインストールする必要があります。詳しくは、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

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

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N



export CTX_XDL_DDC_LIST='list-ddc-fqdns'



export CTX_XDL_VDA_PORT=port-number



export CTX_XDL_REGISTER_SERVICE=Y|N
```

```
export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

```
export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number


export CTX_XDL_TELEMETRY_PORT=port-number


export CTX_XDL_START_SERVICE=Y|N


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

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \


CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \
```

```
CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \
```

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

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。

手順 9：XDPing を実行する

`sudo /opt/Citrix/VDA/bin/xdping` を実行して、Linux VDA 環境における一般的な構成の問題を確認します。詳しくは、「XDPing」を参照してください。

手順 10：Linux VDA を実行する

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

Linux VDAの起動:

Linux VDAサービスを起動するには:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDAの停止:

Linux VDAサービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

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

> **Linux VDAの再起動:**

Linux VDAサービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDAのステータスの確認:

Linux VDAサービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ11: マシンカタログを作成する

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

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

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

注:

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

ヒント:

削除したマシンをActive Directoryドメインに再参加させる場合は、そのマシンをマシンカタログから削除し、再度追加してください。

ステップ12：デリバリーグループを作成する

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

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

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

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

> マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2212」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 の 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** のオプションで **no** を選択します。
5. **Modify DNS Configuration** のオプションで **Use Custom Policy** を選択します。
6. ネットワーク設定を反映するように、以下を編集します。

    - **静的ホスト名** – マシンのDNSホスト名を追加します。
    - **ネームサーバー** – DNSサーバーのIPアドレスを追加します。通常、これはADドメインコントローラーのIPアドレスです。
    - **ドメイン検索リスト** – DNSドメイン名を追加します。

7. `/etc/hosts` ファイルの次の行を、FQDNとホスト名を最初の2つのエントリとして含むように変更します。

   ```
   127.0.0.1 <FQDN of the VDA> <hostname of the VDA> localhost
   ```

> **注:**
>
> Linux VDAは現在、NetBIOS名の切り捨てをサポートしていません。そのため、ホスト名は15文字を超えてはなりません。 **ヒント:**
>
> a～z、A～Z、0～9、およびハイフン (-) の文字のみを使用してください。アンダースコア (_)、スペース、その他の記号は避けてください。ホスト名を数字で始めたり、ハイフンで終わらせたりしないでください。このルールはDelivery Controllerのホスト名にも適用されます。

ホスト名の確認

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

名前解決とサービスの到達可能性の確認

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1c: NTPサービスを構成する

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することが重要です。Linux VDAを仮想マシン (VM) としてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモートNTPサービスを使用して時刻を維持することが推奨されます。デフォルトのNTP設定にいくつかの変更が必要になる場合があります。

SUSE 15.4の場合:

1. UIベースのYaSTツールを起動します。
2. **ネットワークサービス**を選択し、次に**NTP設定**を選択します。
3. **NTPデーモンの開始**セクションで、**今すぐ起動してブート時にも起動**を選択します。
4. **構成ソース**に**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Linux VDAソフトウェアは、以下のパッケージに依存しています。

- OpenJDK 11
- Open Motif ランタイム環境 2.3.1 以降
- Cups 1.6.0以降

- ImageMagick 6.8以降

リポジトリの追加

ImageMagickを除くほとんどの必要なパッケージは、公式リポジトリから入手できます。ImageMagickパッケージを入手するには、YaSTまたは次のコマンドを使用して`sle-module-desktop-applications`リポジトリを有効にします。

`SUSEConnect -p sle-module-desktop-applications/<version number>/x86_64`

KERBEROSクライアントのインストール

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

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

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

OPENJDK 11のインストール

Linux VDAにはOpenJDK 11が必要です。

OpenJDK 11をインストールするには、次のコマンドを実行します。

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

使用するデータベースのインストールと指定

実験的な機能として、PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集することで、SQLiteとPostgreSQLを切り替えることもできます。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。

このセクションでは、PostgreSQLおよびSQLiteデータベースをインストールする方法と、使用するデータベースを指定する方法について説明します。

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

PostgreSQLのインストール

`Postgresql`をインストールするには、次のコマンドを実行します。

```
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、次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLiteをインストールする
SUSEの場合、SQLiteをインストールするには次のコマンドを実行します。

```
sudo zypper install sqlite3
```

使用するデータベースを指定する
SQLite、PostgreSQL、またはその両方をインストールした後、Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集して、使用するデータベースを指定できます。これを行うには、次の手順を完了します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールの場合、この手順は省略してください。
2. `/etc/xdl/db.conf`を編集して、使用するデータベースを指定します。
3. `ctxsetup.sh`を実行します。

> **注:**
>
> PostgreSQLのポート番号を設定するには、`/etc/xdl/db.conf`を使用することもできます。

ステップ2：ハイパーバイザーを準備する

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

Citrix Hypervisor™での時刻同期を修正する

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

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

```
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** 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、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 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で問題が発生します。両方ともシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックをNTPと同期させます。この場合、ホスト時刻同期を無効にする必要があります。

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

1. vSphere Clientを開きます。

2. Linux VMの設定を編集します。
3. **Virtual Machine Properties**ダイアログで、**Options**タブを開きます。
4. **VMware Tools**を選択します。
5. **Advanced**ボックスで、**Synchronize guest time with host**のチェックを外します。

ステップ3：Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory (AD) ドメインに追加するには、次の方法が利用可能です：

- サンバ ウィンバインド
- 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 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、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ると、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。マシンの再起動をお勧めします。

```
su -

reboot
```

ドメインメンバーシップの確認

Delivery Controller は、すべての VDA マシン (Windows および Linux VDA) が Active Directory にコンピューターオブジェクトを持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

システムのキー配布ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist –ke
```

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

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

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

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

```
sudo klist
```

マシンアカウントの詳細を調べるには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username

id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

```
klist
```

セッションを終了します。

```
exit
```

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

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`にコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

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

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

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

LINUX VDAでのQUESTの構成

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

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



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

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

PAMとNSSを構成する

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザーログオンを有効にするには、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）。

ドメインメンバーシップを確認する

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

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

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

`ERROR: No domain could be found.`

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

ユーザー認証を確認する

QuestがPAMを介してドメインユーザーを認証できることを確認します。これを行うには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

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

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

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

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

userは、マシンをActive Directoryドメインに参加させる権限を持つActive Directoryドメインユーザーです。**domain-name**は、Linuxマシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（Windows VDAおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。Centrifyで参加したLinuxマシンがドメイン上にあることを確認するには：

```
sudo adinfo
```

Joined to domainの値が有効であり、**CentrifyDC**の**mode**が**connected**を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

SSSD

SUSEでSSSDを使用している場合は、このセクションの指示に従ってください。このセクションには、Linux VDAマシンをWindowsドメインに参加させる手順が含まれており、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、ホストキー配布ファイルを作成する

SSSD は、ドメインへの参加やシステムキー配布ファイルの管理のための Active Directory クライアント機能を提供しません。代わりに **Samba** アプローチを使用できます。SSSD を構成する前に、以下の手順を完了してください。

1. ネームサービスキャッシュデーモン (NSCD) デーモンを停止して無効にします。

```
sudo systemctl stop nscd

sudo systemctl disable nscd
```

2. ホスト名と Chrony 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 セクションの下に以下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。 セクションの下に以下のエントリを追加します。

```
[global]

    workgroup = domain



    client signing = yes



    client use spnego = yes



    kerberos method = secrets and keytab



    realm = REALM



    security = ADS
```

domain は 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. 設定済みのKerberosクライアントを使用して、管理者としてターゲットドメインに認証します。

```
kinit administrator
```

8. **net** ユーティリティを使用して、システムをドメインに参加させ、システムキータブファイルを生成します。

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

SSSD 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 のような DNS ドメイン名です。

> **注:**
>
> **ldap_id_mapping** は true に設定されており、SSSD 自体が Windows SID を Unix UID にマッピングします。そうでない場合、Active Directory は POSIX 拡張機能を提供できる必要があります。**ad_gpo_access_control** は、Linux セッションでの無効なログオンエラーを防ぐために **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権限を設定します。

```
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 **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、次の方法で確認します。

```
sudo klist
```

ユーザー認証を検証する

SSSDは、デーモンで直接認証をテストするためのコマンドラインツールを提供しておらず、PAMを介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u
```

```
klist



exit
```

`klist` コマンドによって返された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、Active Directoryドメインにマシンを追加する権限を持つドメインユーザーです。**domain-name** は、ドメインのDNS名です（例: example.com）。

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

ドメインメンバーシップを確認する

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

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、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\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

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

手順4：.NET Runtime 6.0をインストールする

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

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

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

ステップ 5: Linux VDA パッケージをダウンロードする

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

   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、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>
   ```

ステップ 6: Linux VDA をインストールする

ステップ 6a: 古いバージョンをアンインストールする

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

1. Linux VDA サービスを停止します。

   ```
   sudo /sbin/service ctxvda stop



   sudo /sbin/service ctxhdx stop
   ```

> **注:**
>
> `ctxvda` および `ctxhdx` サービスを停止する前に、`service ctxmonitorservice stop` コマンドを実行してモニターサービスデーモンを停止します。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。

2. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

> **重要:**
>
> 最新の2つのバージョンからのアップグレードがサポートされています。
>
> **注:**
>
> インストールされたコンポーネントは **/opt/Citrix/VDA/** にあります。
>
> コマンドを実行するには、完全なパスが必要です。または、**/opt/Citrix/VDA/sbin** と **/opt/Citrix/VDA/bin** をシステムパスに追加することもできます。

ステップ 6b: Linux VDA をインストールする

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

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

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

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

ステップ 6c: Linux VDA をアップグレードする（オプション）

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

注:

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

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

SUSE 15 の RPM 依存関係リスト:

```
java-11-openjdk >= 11


ImageMagick >= 7.0


dbus-1 >= 1.12.2


dbus-1-x11 >= 1.12.2


xorg-x11 >= 7.6_1


libXpm4 >= 3.5.12


libXrandr2 >= 1.5.1


libXtst6 >= 1.2.3
```

```
pam >= 1.3.0

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.4

cups >= 2.2

cups-filters >= 1.25

libxml2-2 >= 2.9

libmspack0 >= 0.6

ibus >= 1.5

libtcmalloc4 >= 2.5
```

```
libcap-progs >= 2.26


mozilla-nss-tools >= 3.53.1


libpython3_6m1_0 >= 3.6~


libQt5Widgets5 >= 5.12


libqrencode4 >= 4.0.0


libImlib2-1 >= 1.4.10
```

重要:

アップグレード後、Linux VDA マシンを再起動します。

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、次のガイドを参照してください。

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

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

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

手順8：Linux VDAを構成する

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。スクリプトは変更を加える前に、環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh –help
```

プロンプトによる構成

プロンプトで質問に答える手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプト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_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用する、スペース区切りのDelivery Controllerの完全修飾ドメイン名（FQDN）のリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。

- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux VDAサービスはマシンの起動後に開始されます。この値はデフォルトでYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDAサービスでは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux VDAに必要なポート（デフォルトでポート80および1494）をシステムファイアウォールで自動的に開くことができます。デフォルトでYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – Samba Winbind (サンバ・ウィンバインド)
  - 2 – Quest Authentication Service (クエスト認証サービス)
  - 3 – Centrify DirectControl (セントリファイ・ダイレクトコントロール)
  - 4 – SSSD
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合は、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDAはDNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合は、LDAPポートを持つLDAP FQDNのスペース区切りリストを指定できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。LDAPポート番号を389として指定すると、Linux VDAは指定されたドメイン内の各LDAPサーバーをポーリングモードでクエリします。ポリシーの数がx、LDAPサーバーの数がyの場合、Linux VDAはXにYを乗じた合計クエリを実行します。ポーリング時間がしきい値を超えると、セッションログオンが失敗する可能性があります。より高速なLDAPクエリを有効にするには、ドメイン

コントローラーで**Global Catalog**を有効にし、関連するLDAPポート番号を3268として指定します。この変数はデフォルトで**\<none\>**に設定されています。

- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを介してLDAPにクエリを実行します。検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service（FAS）サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていませんが、代わりにセミコロン区切りのFASサーバーのリストを提供できます。シーケンスはADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を**\<none\>**テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。FASサーバーと適切に通信するには、FASサーバーで指定されているポート番号と一致するポート番号を付加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするために.NET Runtime 6.0をインストールするパス。デフォルトのパスは/usr/binです。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合は、変数値を**mate**に設定する必要があります。

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

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

  b. デスクトップ環境を指定するために、`.xsession` ファイルを編集します。

  - **SUSE 15 の MATE デスクトップの場合**

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

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


fi
```

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

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


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


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

- **SUSE 15 の GNOME デスクトップの場合**

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


if [ -n "$GSESSION" ]; then
exec gnome-session


fi
```

c. ターゲットセッションユーザーと 700 ファイル権限を共有します。

バージョン 2209 以降、セッションユーザーはデスクトップ環境をカスタマイズできます。この機能を有効にするには、VDA に切り替え可能なデスクトップ環境を事前にインストールする必要があります。詳しくは、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

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

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N



export CTX_XDL_DDC_LIST='list-ddc-fqdns'



export CTX_XDL_VDA_PORT=port-number



export CTX_XDL_REGISTER_SERVICE=Y|N



export CTX_XDL_ADD_FIREWALL_RULES=Y|N



export CTX_XDL_AD_INTEGRATION=1|2|3|4



export CTX_XDL_HDX_3D_PRO=Y|N
```

```
export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N
```

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

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \
```

```
CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

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

構成の変更を削除する

一部のシナリオでは、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成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てください。

ステップ9: XDPingを実行する

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

ステップ10: Linux VDAを実行する

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

Linux VDAを開始する:

Linux VDAサービスを開始するには:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDAを停止する:

Linux VDAサービスを停止するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDAを再起動する:

Linux VDAサービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDAのステータスを確認する:

Linux VDAサービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status
```

```
sudo /sbin/service ctxhdx status
```

ステップ11: マシンカタログを作成する

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

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

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

注:

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

ヒント:

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

ステップ12: デリバリーグループを作成する

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

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

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

> **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 2212」を参照してください。

Ubuntu に Linux VDA を手動でインストールする

> **重要:**

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

ステップ 1: 構成情報と Linux マシンを準備する

ステップ 1a: ネットワーク構成を確認する

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

Ubuntu 18.04 Live Server を使用している場合は、ホスト名を設定する前に **/etc/cloud/cloud.cfg** 構成ファイルで次の変更を行ってください。

```
preserve_hostname: true
```

ステップ 1b: ホスト名を設定する

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

`hostname`

ステップ 1c: ホスト名にループバックアドレスを割り当てる

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されることを確認してください。そのためには、**/etc/hosts** ファイルの次の行を変更して、FQDN とホスト名を最初の2つのエントリとして含めます。

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、`hostname-fqdn` または `hostname` への他の参照をすべて削除します。

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

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

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e：マルチキャストDNSを無効にする

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

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

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

に。

`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`

一般的な展開では、パブリックNTPプールサーバーから直接ではなく、ローカルのドメインコントローラーから時刻を同期します。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

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

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h: OpenJDK 11のインストール

Linux VDAにはOpenJDK 11が必要です。

Ubuntu 20.04およびUbuntu 18.04では、次のコマンドを使用してOpenJDK 11をインストールします。

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

ステップ1i: 使用するデータベースのインストールと指定

実験的な機能として、PostgreSQLに加えてSQLiteを使用できます。Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集することで、SQLiteとPostgreSQLを切り替えることもできます。手動インストールの場合、それらを切り替える前に、SQLiteとPostgreSQLを手動でインストールする必要があります。

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

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

POSTGRESQLのインストール

次のコマンドを実行してPostgreSQLをインストールします。

```
sudo apt-get install -y postgresql



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

次のコマンドを実行して、それぞれ、マシンの起動時または即座にPostgreSQLを開始します。

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

SQLITEのインストール

Ubuntuの場合、次のコマンドを実行してSQLiteをインストールします。

```
sudo apt-get install -y sqlite3
```

使用するデータベースの指定

SQLite、PostgreSQL、またはその両方をインストールした後、Linux VDAパッケージのインストール後に`/etc/xdl/db.conf`を編集して、使用するデータベースを指定できます。これを行うには、次の手順を完了します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh`を実行します。新規インストールである場合は、この手順を省略してください。
2. 使用するデータベースを指定するために`/etc/xdl/db.conf`を編集します。

3. `ctxsetup.sh`を実行します。

> **注:**
>
> PostgreSQLのポート番号を構成するには、**/etc/xdl/db.conf**を使用することもできます。

手順1j: Motifをインストールする

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

手順1k: その他のパッケージをインストールする

Ubuntu 22.04の場合:

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.5-0

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

Ubuntu 20.04、Ubuntu 18.04の場合:

```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.4-2

sudo apt-get install -y krb5-user

sudo apt-get install -y libgtk2.0-0
```

手順2: ハイパーバイザーを準備する

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

Citrix Hypervisor™での時刻同期を修正する

Citrix Hypervisorの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとCitrix Hypervisorの両方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。

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

```
su -



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

このコマンドは0または1を返します:

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

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

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

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

この変更を再起動後も永続的に適用するには、**/etc/sysctl.conf ファイル**を編集し、次の行を追加します。

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

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

管理オペレーティングシステムから：

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

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

ESX および ESXi での時刻同期の修正

VMware の時刻同期機能が有効になっている場合、各準仮想化 Linux VM 内で NTP とハイパーバイザーとの間で問題が発生します。どちらもシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるため、各 Linux ゲスト内のシステムクロックが NTP と同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。

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

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **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、次の方法が利用可能です。

- サンバ・ウィンバインド
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新する

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-config krb5-locales krb5-user
```

マシン起動時にWINBINDデーモンが開始するように有効にする

Winbindデーモンは、マシン起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind` スクリプトが `/etc/init.d` の下にあることを確認してください。

KERBEROSを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

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

> `[libdefaults]`

`default_realm =` *`REALM`*

`dns_lookup_kdc = false`

`[realms]`

`REALM` `= {`

`admin_server =` *`domain-controller-fqdn`*

`kdc =` *`domain-controller-fqdn`*

`}`

`[domain_realm]`

`domain-dns-name` `=` *`REALM`*

`.domain-dns-name` `=` *`REALM`*

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

WINBIND認証を構成する

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

/etc/samba/smb.confを開き、以下の設定を行います。

`[global]`

`workgroup =` *`WORKGROUP`*

`security = ADS`

`realm =` *`REALM`*

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`winbind trusted domains only = no`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUPは**REALM**の最初のフィールドであり、**REALM**はKerberosレルム名の大文字表記です。

NSSWITCHを構成する

/etc/nsswitch.confを開き、以下の行に**winbind**を追加します。

`passwd: compat winbind`

`group: compat winbind`

WINDOWSドメインに参加する

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

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

ここで、**REALM**はKerberosレルム名の大文字表記であり、**user**はコンピューターをドメインに追加する権限を持つドメインユーザーです。

`WINBIND`を再起動する

```
sudo systemctl restart winbind
```

WINBIND用にPAMを構成する

以下のコマンドを実行し、**Winbind NT/Active Directory authentication**と**Create home directory on login**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

> **ヒント:**
>
> **winbind**デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

ドメインメンバーシップを確認する

デリバリーコントローラーは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持っていることを要求します。

マシンがドメインに参加していることを確認するには、**Samba**の**net ads**コマンドを実行します。

```
sudo net ads testjoin
```

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

```
sudo net ads info
```

KERBEROS構成の確認

Linux VDAで使用するためにKerberosが正しく構成されていることを確認するには、システムの**keytab**ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してドメインコントローラーでマシンを認証するには、Kerberos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。シェルによる置換を防ぐため、ドル記号 ($) はバックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

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

```
sudo klist
```

マシンのアカウントの詳細を調べるには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username



id -u
```

注:

SSHコマンドを正常に実行するには、SSHが有効になっていて、正しく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るuidに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーのKerberos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
```

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/2301/installation-overview/manual-installation-overview/ubuntu.html#step-6-install-the-linux-vda)に進みます。

ヒント:

ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップが表示されない場合は、マシンを再起動してからもう一度試してください。

Quest認証サービス

ドメインコントローラーでのQUESTの構成

Active Directoryドメインコントローラーにインストールして構成済みであり、`Active Directory`でコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLINUX VDAマシンにログオンできるようにする

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

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

注:

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

LINUX VDAでのQUESTの構成

SELinuxポリシーの適用を回避する
デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Quest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デーモンの構成
Kerberosチケットの自動更新は有効にして切断する必要があります。認証（オフラインログオン）は無効にする必要があります。

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



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

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

HDXおよびsu、ssh、RDPなどの他のサービスを介したドメインユーザー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ドメインへの参加

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

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

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

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxのすべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持っていることを要求します。Questに参加しているLinuxマシンがドメイン上にあることを確認するには：

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username
```

```
id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

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

セントリファイ ダイレクトコントロール

WINDOWSドメインへの参加

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

```
su –

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

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

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxのいずれであっても、すべてのVDAマシンが**Active Directory**にコンピューターオブジェクトを持つことを要求します。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには、次の手順を実行します。

```
su -



adinfo
```

Joined to domain の値が有効であり、**CentrifyDC mode** が **connected** を返すことを確認します。モードが開始状態のままになっている場合、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/2301/installation-overview/manual-installation-overview/ubuntu.html#step-6-install-the-linux-vda)に進みます。

SSSD

KERBEROSを構成する

Kerberosをインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、rootとして **/etc/krb5.conf** を開き、パラメーターを設定します。

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

> `[libdefaults]`

`default_realm = REALM`

`dns_lookup_kdc = false`

`[realms]`

`REALM = {`

`admin_server = domain-controller-fqdn`

`kdc = domain-controller-fqdn`

`}`

`[domain_realm]`

`domain-dns-name = REALM`

`.domain-dns-name = REALM`

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

ドメインに参加する

SSSD は、ID プロバイダーとして Active Directory を、認証に Kerberos を使用するように構成する必要があります。ただし、SSSD はドメイン参加やシステムキーファイル管理のための AD クライアント機能を提供しません。代わりに、**adcli**、**realmd**、または **Samba** を使用できます。

> **注:**
>
> このセクションでは、**adcli** と **Samba** に関する情報のみを提供します。

- **adcli を使用してドメインに参加する場合は、次の手順を実行します。**

1. **adcli** をインストールします。

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

2. **adcli** を使用してドメインに参加します。

古いシステムキーファイルを削除し、以下を使用してドメインに参加します。

```
su -



rm -rf /etc/krb5.keytab



adcli join domain-dns-name -U user -H hostname-fqdn
```

user は、マシンをドメインに追加する権限を持つドメインユーザーです。**hostname-fqdn** は、マシンの FQDN 形式のホスト名です。

-H オプションは、Linux VDA が必要とする host/hostname-fqdn@REALM 形式で SPN を生成するために **adcli** に必要です。

3. ドメインメンバーシップを確認します。

   Ubuntu 22.04 および Ubuntu 20.04 マシンの場合、マシンがドメインに参加しているかどうかをテストするには、`adcli testjoin` コマンドを実行します。

   Ubuntu 18.04 マシンの場合、`sudo klist -ket` コマンドを実行します。**adcli** ツールの機能は限られています。このツールは、マシンがドメインに参加しているかどうかをテストする方法を提供しません。最善の代替策は、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ことです。各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認してください。

- **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 構成の変更が必要です。SSSD の一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトでインストールされておらず、手動で作成する必要があります。root として、**/etc/sssd/sssd.conf** を作成または開いて、次の設定を行います。

```
[sssd]

services = nss, pam

config_file_version = 2

domains = domain-dns-name

[domain/ domain-dns-name ]

id_provider = ad

access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than 14
days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter than
2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash
```

`ad_gpo_map_remote_interactive = +ctxhdx`

注:

ldap_id_mapping は **true** に設定されており、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。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読み取り権限のみを持つことを要求します。

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

SSSD デーモンの開始

SSSD デーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM 構成

次のコマンドを実行し、**SSS authentication** および **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

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。

- もし **adcli** を使用してドメインメンバーシップを確認する場合は、`sudo adcli info domain-dns-name` コマンドを実行してドメイン情報を表示します。
- もし **Samba** を使用してドメインメンバーシップを確認する場合は、`sudo net ads testjoin` コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info` コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

KERBEROS 構成の確認

Kerberos が Linux VDA で使用するために正しく構成されていることを確認するには、システムの keytab ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

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

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

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、デーモンで認証を直接テストするためのコマンドラインツールを提供しておらず、PAM を介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返された 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の確認

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持っている必要があります。PBISに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の確認後、手順6：Linux VDAのインストールに進みます。

手順4：.NET Runtime 6.0のインストール

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

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

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

手順5：Linux VDAパッケージのダウンロード

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、お使いのLinuxディストリビューションに一致するLinux VDAパッケージと、Linux VDAパッケージの整合性検証に使用できるGPG公開キーを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、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>

   ```
   ```

ステップ6：Linux VDAをインストールする

ステップ6a：Linux VDAをインストールする

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

Ubuntu 22.04の場合：

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

Ubuntu 20.04の場合：

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

注:

GCP上のUbuntu 20.04では、RDNSを無効にします。そのためには、/etc/krb5.confの**の下にrdns = false**の行を追加します。

Ubuntu 18.04の場合：

```
sudo dpkg -i xendesktopvda_<version>.ubuntu18.04_amd64.deb
```

Ubuntu 22.04のDebian依存関係リスト：

```
openjdk-11-jdk >= 11


imagemagick >= 8:6.9.11


libgtkmm-3.0-1v5 >= 3.24.5


ufw >= 0.36


ubuntu-desktop >= 1.481
```

```
libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.37

gtk3-nocsd >= 3

bash >= 5.1

findutils >= 4.8.0

sed >= 4.8

cups >= 2.4

libmspack0 >= 0.10

ibus >= 1.5
```

```
libgoogle-perftools4 >= 2.9~

libpython3.10 >= 3.10~

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

libnss3-tools >= 2:3.68

libqt5widgets5 >= 5.15~

libqrencode4 >= 4.1.1

libimlib2 >= 1.7.4
```

Ubuntu 20.04のDebian依存関係リスト：

```
openjdk-11-jdk >= 11

imagemagick >= 8:6.9.10

libgtkmm-3.0-1v5 >= 3.24.2

ufw >= 0.36
```

```
ubuntu-desktop >= 1.450

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.34

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.7.0

sed >= 4.7

cups >= 2.3

libmspack0 >= 0.10
```

```
ibus >= 1.5

libgoogle-perftools4 >= 2.7~

libpython3.8 >= 3.8~

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

libnss3-tools >= 2:3.49

libqt5widgets5 >= 5.7~

libqrencode4 >= 4.0.0

libimlib2 >= 1.6.1
```

Ubuntu 18.04のDebian依存関係リスト：

```
openjdk-11-jdk >= 11

imagemagick >= 8:6.8.9.9

ufw >= 0.35
```

```
libgtkmm-3.0-1v5 >= 3.22.2

ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

gtk3-nocsd >= 3

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1
```

```
libmspack0 >= 0.6

ibus >= 1.5

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

libgoogle-perftools4 >= 2.4~

libpython3.6 >= 3.6~

libnss3-tools >= 2:3.35

libqt5widgets5 >= 5.7~

libqrencode3 >= 3.4.4

libimlib2 >= 1.4.10
```

注:

> このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、「システム要件」(/ja-jp/linux-virtual-delivery-agent/2301/system-requirements.html)を参照してください。

ステップ6b：Linux VDAをアップグレードする（オプション）

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

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

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

ステップ7：NVIDIA GRIDドライバーをインストールする

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

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

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

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

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

ステップ8：Linux VDAを構成する

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

プロンプトで質問に答える手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求せず、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数は次のとおりです。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux VDAサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDAサービスは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux VDAに必要なポート（デフォルトではポート80および1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。

- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \|5** – Linux VDAは、Delivery Controllerとの認証のためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ ウィンバインド
  - 2 – Quest 認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するかを指定します。HDX 3D Pro 環境では、この変数を Y に設定します。この変数はデフォルトで N に設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに限定するには、DNS サイト名を指定します。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合、LDAP ポートを含む LDAP FQDN のスペース区切りリストを指定できます。たとえば、ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268 のように指定します。LDAP ポート番号を 389 と指定した場合、Linux VDA は指定されたドメイン内の各 LDAP サーバーをポーリングモードで照会します。ポリシーの数が x で LDAP サーバーの数が y の場合、Linux VDA は合計で X かける Y のクエリを実行します。ポーリング時間がしきい値を超えると、セッションログオンが失敗する可能性があります。より高速な LDAP クエリを有効にするには、ドメインコントローラーで **Global Catalog** を有効にし、関連する LDAP ポート番号を 3268 と指定します。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。ただし、検索パフォーマンスを向上させるために、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで **\<none\>** に設定されています。

- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりに FAS サーバーのセミコロン区切りリストを提供できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。FAS サーバーと適切に通信するには、FAS サーバーで指定されているものと一致するポート番号を付加してください。例: CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするための .NET Runtime 6.0 のインストールパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用する GNOME、GNOME Classic、または MATE デスクトップ環境を指定します。この変数を指定しない場合、VDA に現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップが MATE の場合は、変数の値を **mate** に設定する必要があります。

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

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

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

  - **MATE デスクトップの場合**

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



if [ -n "$MSESSION" ]; then

  exec mate-session



fi
```

- **GNOME Classic デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **GNOME デスクトップの場合**

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



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

c. ターゲットセッションユーザーと700ファイル権限を共有します。

バージョン2209以降、セッションユーザーはデスクトップ環境をカスタマイズできます。この機能を有効にするには、事前にVDAに切り替え可能なデスクトップ環境をインストールする必要があります。詳細については、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成が完了したときにLinux VDAサービスを開始するかどうか。デフォルトではYに設定されています。
- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutがリッスンするソケットポート。デフォルトポートは7503です。

- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N


export CTX_XDL_DDC_LIST='list-ddc-fqdns'


export CTX_XDL_VDA_PORT=port-number


export CTX_XDL_REGISTER_SERVICE=Y|N


export CTX_XDL_ADD_FIREWALL_RULES=Y|N


export CTX_XDL_AD_INTEGRATION=1|2|3|4|5


export CTX_XDL_HDX_3D_PRO=Y|N


export CTX_XDL_VDI_MODE=Y|N


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


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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

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。最初の行に**#\!/bin/bash**を指定して、上記のコマンドからシェルスクリプトファイルを作成することをお勧めします。

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \
```

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

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \
```

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


CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \


CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \


CTX_XDL_TELEMETRY_PORT=port-number \


CTX_XDL_START_SERVICE=Y|N \


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

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

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

構成の変更を削除するには：

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

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル**/tmp/xdl.configure.log**に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

Linux VDAソフトウェアをアンインストールする

Linux VDAがインストールされているかを確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

注:

Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前に設定されたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

ヒント:

このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

ステップ9：XDPingを実行する

Linux VDA環境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`と`ctxhdx`サービスを停止する前に、`service ctxmonitorservice stop`コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

> **Linux VDAの再起動:**

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のステータスの確認:

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: マシンカタログの作成

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

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

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

注:

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

ヒント:

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

ステップ12: デリバリーグループの作成

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 2212を参照してください。

Debian に Linux VDA を手動でインストールする

> **重要:**
>
> 新規インストールの場合、迅速なインストールには簡単インストールを使用することをお勧めします。簡単インストールは、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、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
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、`hostname-fqdn`または`hostname`への他の参照を削除します。

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

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

ステップ1d：ホスト名の確認

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、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: マルチキャストDNSを無効にする

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

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

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

に変更します。

`hosts: files dns`

ステップ1f: 名前解決とサービスの到達可能性を確認する

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

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn



nslookup delivery-controller-fqdn



ping delivery-controller-fqdn
```

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

ステップ1g: クロック同期を設定する (chrony)

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

chronyをインストールします。

```
apt-get install chrony
```

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、時刻はローカルのドメインコントローラーから同期し、パブリックNTPプールサーバーから直接同期することはありません。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

ループバックIPアドレス、localhost、およびパブリックサーバーの ***.pool.ntp.org** エントリを含む、リストされている他の **server** または **pool** エントリをすべて削除します。

変更を保存し、Chrony デーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ 1h: パッケージのインストール

```
sudo apt-get install -y libsasl2-2



sudo apt-get install -y libgtk2.0-0
```

ステップ 1i: 必要な依存関係をインストールするためのリポジトリの追加

Debian 11.3 の場合、/etc/apt/sources.list ファイルに `deb http://deb.debian.org/debian/ bullseye main` の行を追加します。

ステップ 1j: 使用するデータベースのインストールと指定

実験的な機能として、PostgreSQL に加えて SQLite を使用できます。Linux VDA パッケージのインストール後に `/etc/xdl/db.conf` を編集することで、SQLite と PostgreSQL を切り替えることもできます。手動インストールの場合、切り替える前に SQLite と PostgreSQL を手動でインストールする必要があります。

このセクションでは、PostgreSQL および SQLite データベースのインストール方法と、使用するデータベースの指定方法について説明します。

> **注:**
>
> SQLite は VDI モードでのみ使用することをお勧めします。

POSTGRESQL のインストール

PostgreSQL をインストールするには、次のコマンドを実行します。

```
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 を起動するには、それぞれ次のコマンドを実行します。

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

SQLITE のインストール

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の指定

SQLite、PostgreSQL、またはその両方をインストールした後、Linux VDA パッケージのインストール後に `/etc/xdl/db.conf` を編集して、使用するデータベースを指定できます。これを行うには、次の手順を実行します。

1. `/opt/Citrix/VDA/sbin/ctxcleanup.sh` を実行します。新規インストールの場合、この手順は省略してください。

2. 使用するデータベースを指定するために`/etc/xdl/db.conf`を編集します。
3. `ctxsetup.sh`を実行します。

> **注:**
>
> PostgreSQLのポート番号を構成するには、**/etc/xdl/db.conf**を使用することもできます。

ステップ2：ハイパーバイザーの準備

サポートされているハイパーバイザー上でLinux VDAをVMとして実行する場合、いくつかの変更が必要です。使用しているハイパーバイザープラットフォームに基づいて、以下の変更を行ってください。Linuxマシンをベアメタルハードウェアで実行している場合、変更は不要です。

Citrix Hypervisor™での時刻同期の修正

Citrix Hypervisorの時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとCitrix Hypervisorの両方がシステムクロックを管理しようとするため、問題が発生します。クロックが他のサーバーと同期しなくなるのを避けるため、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホストの時刻同期を無効にする必要があります。HVMモードでは変更は不要です。

Citrix VM Toolsがインストールされた準仮想化Linuxカーネルを実行している場合、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかを確認できます。

```
su -

cat /proc/sys/xen/independent_wallclock
```

このコマンドは0または1を返します。

- 0 - 時刻同期機能が有効になっており、無効にする必要があります。
- 1 - 時刻同期機能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
```

この変更を再起動後も永続的に適用するには、**/etc/sysctl.confファイル**を編集し、以下の行を追加します。

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -

cat /proc/sys/xen/independent_wallclock
```

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

Hyper-V Linux Integration ServicesがインストールされているLinux VMは、Hyper-Vの時刻同期機能を使用してホストオペレーティングシステムの時刻を使用できます。システムクロックの精度を維持するために、NTPサービスと並行してこの機能を有効にしてください。

管理オペレーティングシステムから:

1. Hyper-Vマネージャーコンソールを開きます。
2. Linux VMの設定で、**統合サービス**を選択します。
3. **時刻同期**が選択されていることを確認します。

> **注:**
>
> このアプローチは、NTPとの競合を避けるためにホスト時刻同期が無効になっているVMwareおよびCitrix Hypervisorとは異なります。Hyper-Vの時刻同期は、NTP時刻同期と共存し、補完することができます。

ESXおよびESXiでの時刻同期の修正

VMware時刻同期機能が有効になっている場合、各準仮想化Linux VM内でNTPとハイパーバイザーとの間で問題が発生します。両方ともシステムクロックを同期しようとします。クロックが他のサーバーと同期しなくなるのを避けるために、各Linuxゲスト内のシステムクロックがNTPと同期していることを確認してください。この場合、ホスト時刻同期を無効にする必要があります。

VMware Toolsがインストールされた準仮想化Linuxカーネルを実行している場合:

1. vSphere Clientを開きます。

2. Linux VMの設定を編集します。
3. **仮想マシンのプロパティ**ダイアログで、**オプション**タブを開きます。
4. **VMware Tools**を選択します。
5. 「**詳細設定**」ボックスで、「**ゲストの時刻をホストと同期**」のチェックを外します。

ステップ3：Linux VMをWindowsドメインに追加する

LinuxマシンをActive Directory (AD) ドメインに追加するには、以下の方法があります。

- サンバ・ウィンバインド
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名を使用すると、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新する

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-config krb5-locales krb5-user
```

マシン起動時にWINBINDデーモンが開始するように有効にする

Winbindデーモンは、マシン起動時に開始するように構成する必要があります。

```
sudo systemctl enable winbind
```

> **注:**
>
> `winbind`スクリプトが`/etc/init.d`の下にあることを確認してください。

KERBEROSを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、単一ドメイン、単一フォレストモデルを対象としています。

```
[libdefaults]
default_realm = REALM
dns_lookup_kdc = false
[realms]
REALM = {
admin_server = domain-controller-fqdn
kdc = domain-controller-fqdn
}
[domain_realm]
domain-dns-name = REALM
.domain-dns-name = REALM
```

このコンテキストでの **domain-dns-name** パラメーターは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

WINBIND認証を構成する

/etc/samba/smb.conf を開き、次の設定を行います。

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
```

`encrypt passwords = yes`

`idmap config *:range = 16777216-33554431`

`winbind trusted domains only = no`

`kerberos method = secrets and keytab`

`winbind refresh tickets = yes`

`template shell = /bin/bash`

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字のKerberosレルム名です。

`NSSWITCH` を構成する

/etc/nsswitch.conf を開き、次の行に `winbind` を追加します。

`passwd: systemd winbind`

`group: systemd winbind`

WINDOWSドメインへの参加

ドメインコントローラーに到達可能である必要があり、コンピューターをドメインに追加する権限を持つActive Directoryユーザーアカウントが必要です。

```
sudo net ads join REALM -U user
```

REALM は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

WINBINDの再起動

```
sudo systemctl restart winbind
```

WINBINDのPAMを構成する

次のコマンドを実行し、**Winbind NT/Active Directory authentication** および **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

> **ヒント:**
>
> `winbind` デーモンは、マシンがドメインに参加している場合にのみ実行され続けます。

ドメインメンバーシップの確認

Delivery Controllerは、WindowsまたはLinuxのいずれであっても、すべてのVDAマシンが `Active Directory` にコンピューターオブジェクトを持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します。

```
sudo klist
```

マシンアカウントの詳細を調べてください。

```
sudo net ads status
```

ユーザー認証の確認

wbinfo ツールを使用して、ドメインユーザーがドメインで認証できることを確認します。

```
wbinfo --krb5auth=domain\\username%password
```

ここで指定するドメインは、Kerberos レルム名ではなく、AD ドメイン名です。bash シェルの場合、バックスラッシュ (\\) 文字は別のバックスラッシュでエスケープする必要があります。このコマンドは、成功または失敗を示すメッセージを返します。

Winbind PAM モジュールが正しく構成されていることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username



id -u
```

注:

SSH コマンドを正常に実行するには、SSH が有効になっており、正しく機能していることを確認してください。

id -u コマンドによって返された uid に対応する Kerberos 資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

ユーザーの Kerberos 資格情報キャッシュ内のチケットが有効で、期限切れになっていないことを確認します。

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンすることで実行できます。ドメイン参加の確認後、手順 6: Linux VDA 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、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 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 ユーザーのプロパティを開きます。
2. **Unix アカウント**タブを選択します。
3. **Unix-enabled**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=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、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、次のコマンドを実行してPAMとNSSを手動で構成します。

```
sudo /opt/quest/bin/vastool configure pam

sudo /opt/quest/bin/vastool configure nss
```

WINDOWSドメインに参加する

Quest `vastool` コマンドを使用して、LinuxマシンをActive Directoryドメインに参加させます。

```
sudo /opt/quest/bin/vastool -u user join domain-name
```

ユーザーは、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。`domain-name` はドメインのDNS名です（例: example.com）。

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinuxにかかわらず、すべてのVDAマシンが `Active Directory` にコンピューターオブジェクトを持つことを要求します。Questで参加したLinuxマシンがドメイン上にあることを確認するには、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になっていないことを確認します。

```
/opt/quest/bin/vastool klist
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順 6: Linux VDA のインストールに進みます。

セントリファイ ダイレクトコントロール

WINDOWS ドメインへの参加

Centrify DirectControl Agent がインストールされている場合、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** パラメーターは、Linux マシンを参加させるドメインの名前です。

ドメインメンバーシップの確認

Delivery Controller は、Windows または Linux のいずれであっても、すべての VDA マシンが Active Directory にコンピューターオブジェクトを持つことを要求します。Centrify に参加している Linux マシンがドメイン上にあることを確認するには：

```
su -



adinfo
```

Joined to domain の値が有効であり、**CentrifyDC mode** が **connected** を返すことを確認します。モードが開始状態のままになっている場合、Centrify クライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo --diag
```

さまざまな Active Directory および Kerberos サービスへの接続をテストします。

```
adinfo --test
```

ドメイン参加の検証後、手順 6: Linux VDA のインストールに進みます。

SSSD

KERBEROS の構成

Kerberos をインストールするには、次のコマンドを実行します。

```
sudo apt-get install krb5-user
```

Kerberos を構成するには、root として **/etc/krb5.conf** を開き、パラメーターを設定します。

> **注:**
>
> ADインフラストラクチャに基づいてKerberosを構成します。以下の設定は、シングルドメイン、シングルフォレストモデルを対象としています。

```
[libdefaults]
```

`default_realm = REALM`

`dns_lookup_kdc = false`

`[realms]`

`REALM = {`

`admin_server = domain-controller-fqdn`

`kdc = domain-controller-fqdn`

`}`

`[domain_realm]`

`domain-dns-name = REALM`

`.domain-dns-name = REALM`

このコンテキストでの`domain-dns-name`パラメーターは、example.comのようなDNSドメイン名です。*REALM*は、EXAMPLE.COMのような大文字のKerberosレルム名です。

ドメインに参加する

SSSDは、Active DirectoryをIDプロバイダーとして、Kerberosを認証に使用するように構成する必要があります。ただし、SSSDはドメイン参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。代わりに、**adcli**、**realmd**、または**Samba**を使用できます。

> **注:**
>
> このセクションでは、**adcli**と**Samba**に関する情報のみを提供します。

- **adcliを使用してドメインに参加する場合は、次の手順を完了してください。**

1. **adcli**をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. **adcli**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 形式の SPN を **adcli** が生成するために必要です。

3. システムキータブを確認します。

   `sudo klist -ket` コマンドを実行して、システムキータブファイルが作成されていることを確認します。

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **Samba** を使用してドメインに参加する場合は、次の手順を完了します。

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する:

必要な SSSD および構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新をお勧めします。

```
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 構成の変更が必要です。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 デーモンを開始する

SSSDデーモンを今すぐ起動し、マシンの起動時にデーモンが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では、すべてのVDAマシン（WindowsおよびLinux VDA）が`Active Directory`にコンピューターオブジェクトを持っている必要があります。

- 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
```

KDE または Gnome Display Manager にログオンして、同様のテストを実行できます。ドメイン参加の検証後、手順 6: Linux VDA 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、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

id -u コマンドによって返されたUIDに対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

手順4：.NET Runtime 6.0のインストール

Linux VDAをインストールする前に、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`の指示に従って.NET Runtime 6.0をインストールします。

.NET Runtime 6.0のインストール後、**which dotnet** コマンドを実行してランタイムパスを見つけます。

コマンド出力に基づいて、.NETランタイムバイナリパスを設定します。たとえば、コマンド出力が/aa/bb/dotnetの場合、/aa/bbを.NETバイナリパスとして使用します。

手順5：Linux VDAパッケージのダウンロード

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、お使いのLinuxディストリビューションに一致するLinux VDAパッケージと、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。

   Linux VDAパッケージの整合性を検証するには、公開鍵をDEBデータベースにインポートし、次のコマンドを実行します。

   ```
   ```

   ```
   sudo apt-get install dpkg-sig

   gpg --import <path to the public key>

   dpkg-sig --verify <path to the Linux VDA package>
   ```

   ```
   ```

手順6：Linux VDAのインストール

手順6a：Linux VDAのインストール

Debianパッケージマネージャーを使用してLinux VDAソフトウェアをインストールします。

```
sudo dpkg -i xendesktopvda_<version>.debian10_amd64.deb
```

Debian 11.3の依存関係リスト:

```
openjdk-11-jdk >= 11

imagemagick >= 8:6.9.10

ufw >= 0.36

desktop-base >= 10.0.2

libxrandr2 >= 2:1.5.1

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.33

gtk3-nocsd >= 3

bash >= 5.0

findutils >= 4.6.0
```

```
sed >= 4.7

cups >= 2.2

ghostscript >= 9.53~

libmspack0 >= 0.10

ibus >= 1.5

libgoogle-perftools4 >= 2.7~

libpython3.9 >= 3.9~

libsasl2-modules-gssapi-mit >= 2.1.~

libqt5widgets5 >= 5.5~

mutter >= 3.38.6~

libqrencode4 >= 4.0.0
```

```
libimlib2 >= 1.5.1
```

注:

このバージョンのLinux VDAがサポートするLinuxディストリビューションとXorgバージョンのマトリックスについては、システム要件を参照してください。

ステップ6b: Linux VDAをアップグレードする（オプション）

既存のインストールを、以前の2つのバージョンおよびLTSRリリースからアップグレードできます。

```
sudo dpkg -i <PATH>/<Linux VDA deb>
```

注:

既存のインストールをアップグレードすると、/etc/xdlの下にある構成ファイルが上書きされます。アップグレードを実行する前に、ファイルをバックアップしてください。

ステップ7: NVIDIA GRIDドライバーをインストールする

HDX 3D Proを有効にするには、ハイパーバイザーとVDAマシンにNVIDIA GRIDドライバーをインストールする必要があります。

特定のハイパーバイザーにNVIDIA GRID Virtual GPU Manager（ホストドライバー）をインストールして構成するには、次のガイドを参照してください。

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX
- ニュータニックス AHV

NVIDIA GRIDゲストVMドライバーをインストールして構成するには、次の一般的な手順を実行します。

1. ゲストVMがシャットダウンされていることを確認します。
2. ハイパーバイザーのコントロールパネルで、VMにGPUを割り当てます。

3. VMを起動します。
4. VMにゲストVMドライバーをインストールします。

手順8：Linux VDAを構成する

パッケージのインストール後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
```

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
```

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションは環境変数で指定できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求せず、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して デリバリーコントローラー 名を指定することをサポートしています。デフォルトでは N に設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller に登録するために使用する、スペース区切りの Delivery Controller の完全修飾ドメイン名 (FQDN) のリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。デフォルトではポート 80 です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux VDAサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDAサービスは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux VDAに必要な

ポート（デフォルトではポート80および1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。

- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \|5** – Linux VDA は、デリバリーコントローラーと認証するために Kerberos 構成設定を必要とします。Kerberos 構成は、システムにインストールおよび構成されている Active Directory 統合ツールから決定されます。使用するサポートされている Active Directory 統合方法を指定します。
  - 1 – サンバ Winbind
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計された一連のGPUアクセラレーションテクノロジーである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_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境の場合、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAは、DNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDAは、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。LDAPポート番号を389と指定した場合、Linux VDAは指定されたドメイン内の各LDAPサーバーをポーリングモードで照会します。ポリシーがx個、LDAPサーバーがy個ある場合、Linux VDAは合計でXかけるYのクエリを実行します。ポーリング時間がしきい値を超えると、セッションログオンが失敗する可能性があります。より高速なLDAPクエリを有効にするには、ドメインコントローラーで**グローバルカタログ**を有効にし、関連するLDAPポート番号を3268と指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPを照会します。ただし、検索パフォーマンスを向上させるために、検索ベース（例：

OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。

- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていませんが、代わりにセミコロン区切りのFASサーバーのリストを提供できます。シーケンスは、ADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を**\<none\>**というテキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。FASサーバーと適切に通信するには、FASサーバーで指定されているものと一致するポート番号を付加してください。例：CTX_XDL_FAS_LIST='fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Runtime 6.0のインストールパス。デフォルトのパスは/usr/binです。
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/gnome-classic/mate** – セッションで使用するGNOME、GNOME Classic、またはMATEデスクトップ環境を指定します。変数を指定しない場合、VDAに現在インストールされているデスクトップが使用されます。ただし、現在インストールされているデスクトップがMATEである場合、変数の値を**mate**に設定する必要があります。

  ターゲットセッションユーザーのデスクトップ環境は、次の手順で変更することもできます。

  a. VDA上の**$HOME/<username\>**ディレクトリの下に`.xsession`ファイルを作成します。

  b. `.xsession`ファイルを編集して、ディストリビューションに基づいてデスクトップ環境を指定します。

  - **MATEデスクトップの場合**

```
MSESSION="$(type -p mate-session)"



if [ -n "$MSESSION" ]; then

  exec mate-session
```

```
fi
```

- **GNOME Classicデスクトップの場合**

```
GSESSION="$(type -p gnome-session)"



if [ -n "$GSESSION" ]; then

export GNOME_SHELL_SESSION_MODE=classic



exec gnome-session --session=gnome-classic

fi
```

- **GNOMEデスクトップの場合**

```
GSESSION="$(type -p gnome-session)"



if [ -n "$GSESSION" ]; then

exec gnome-session



fi
```

c. ターゲットセッションユーザーと700ファイル権限を共有します。

バージョン2209以降、セッションユーザーはデスクトップ環境をカスタマイズできます。この機能を有効にするには、VDAに切り替え可能なデスクトップ環境を事前にインストールする必要があります。詳しくは、「セッションユーザーによるカスタムデスクトップ環境」を参照してください。

- **CTX_XDL_START_SERVICE=Y \| N** – Linux VDAの構成が完了したときにLinux VDAサービスを開始するかどうか。デフォルトではYに設定されています。

- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scoutをリッスンするためのソケットポート。デフォルトポートは7503です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scoutと通信するためのポート。デフォルトポートは7502です。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N


export CTX_XDL_DDC_LIST='list-ddc-fqdns'


export CTX_XDL_VDA_PORT=port-number


export CTX_XDL_REGISTER_SERVICE=Y|N


export CTX_XDL_ADD_FIREWALL_RULES=Y|N


export CTX_XDL_AD_INTEGRATION=1|2|3|4|5


export CTX_XDL_HDX_3D_PRO=Y|N


export CTX_XDL_VDI_MODE=Y|N


export CTX_XDL_SITE_NAME=dns-site-name | '<none>'
```

```
export CTX_XDL_LDAP_LIST='list-ldap-servers' | '<none>'


export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'


export CTX_XDL_FAS_LIST='list-fas-servers' | '<none>'


export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime


export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | gnome-classic | mate |
'<none>'


export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number


export CTX_XDL_TELEMETRY_PORT=port-number


export CTX_XDL_START_SERVICE=Y|N


sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
```

sudoコマンドを実行するときは、**-E**オプションを入力して、既存の環境変数を新しく作成されるシェルに渡します。上記のコマンドから、最初の行に**#\!/bin/bash**を含むシェルスクリプトファイルを作成することをお勧めします。

または、単一のコマンドを使用してすべてのパラメーターを指定することもできます。

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \
```

```
CTX_XDL_FAS_LIST='list-fas-servers' \

CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \

CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
```

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セットアップされた PostgreSQL パッケージおよびその他の依存パッケージは削除されません。

> **ヒント:**

> このセクションの情報には、PostgreSQL を含む依存パッケージの削除は含まれていません。

ステップ 9：XDPing を実行する

Linux VDA 環境で一般的な構成の問題を確認するには、`sudo /opt/Citrix/VDA/bin/xdping` を実行します。詳細については、「XDPing」を参照してください。

ステップ 10：Linux VDA を実行する

ctxsetup.sh スクリプトを使用して Linux VDA を構成したら、次のコマンドを使用して Linux VDA を制御します。

Linux VDA を起動する：

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、`service ctxmonitorservice stop` コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンが停止したサービスを再起動します。

> **Linux VDAを再起動する：**

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

ステップ11：マシンカタログを作成する

マシンカタログを作成し、Linux VDAマシンを追加するプロセスは、従来のWindows VDAのアプローチと似ています。これらのタスクを完了する方法の詳細については、「マシンカタログの作成」および「マシンカタログの管理」を参照してください。

Linux VDAマシンを含むマシンカタログを作成する場合、Windows VDAマシン用のマシンカタログを作成するプロセスとは異なるいくつかの制限があります。

- オペレーティングシステムには、以下を選択します。
  - ホスト型共有デスクトップ配信モデルの場合は、**マルチセッションOS**オプション。
  - VDI専用デスクトップ配信モデルの場合は、**シングルセッションOS**オプション。
- 同じマシンカタログにLinux VDAマシンとWindows VDAマシンを混在させないでください。

> **注:**
>
> Citrix Studioの初期バージョンでは、「Linux OS」という概念をサポートしていませんでした。しかし、**Windows Server OS**または**Server OS**オプションを選択すると、同等のホスト

> 型共有デスクトップ配信モデルが意味されます。**Windows Desktop OS**または**Desktop OS**オプションを選択すると、マシンごとに単一ユーザーの配信モデルが意味されます。

> **ヒント:**
>
> マシンをActive Directoryドメインから削除して再参加させる場合、そのマシンをマシンカタログから削除して再度追加する必要があります。

ステップ12：デリバリーグループを作成する

デリバリーグループを作成し、Linux VDAマシンを含むマシンカタログを追加するプロセスは、Windows VDAマシンとほぼ同じです。これらのタスクを完了する方法の詳細については、「デリバリーグループの作成」を参照してください。

Linux VDAマシンカタログを含むデリバリーグループを作成する場合、以下の制限が適用されます。

- 選択したADユーザーとグループが、Linux VDAマシンにログオンできるように適切に構成されていることを確認してください。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- デリバリーグループをWindowsマシンを含むマシンカタログと混在させないでください。

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 2212」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述します。

管理

このセクションには、次のトピックが含まれています。

- CEIP
- HDXインサイト
- Citrix Telemetry Service との統合
- Azure 向け Citrix DaaS Standard の Linux VDA 自己更新
- Linux VM と Linux セッションのメトリック
- ログ収集
- セッションシャドウイング
- モニターサービスデーモン
- ツールとユーティリティ
- その他
    - HTML5 向け Citrix Workspace アプリのサポート
    - Python3 仮想環境の作成
    - NIS と Active Directory の統合
    - IPv6
    - LDAPS
    - **Xオーソリティ**

Citrixカスタマーエクスペリエンス向上プログラム (CEIP)

CEIPに参加すると、匿名統計情報と使用状況情報がCitrixに送信され、Citrix製品の品質とパフォーマンス向上に役立てられます。さらに、匿名データのコピーは、迅速かつ効率的な分析のためにGoogle Analytics (GA) に送信されます。GAはデフォルトで無効になっています。

レジストリ設定

デフォルトでは、Linux VDAをインストールすると、CEIPに自動的に参加します。データの初回アップロードは、Linux VDAのインストール後約7日で行われます。このデフォルト設定はレジストリで変更できます。

- **CEIPスイッチ**

CEIPを有効または無効にするレジストリ設定（デフォルト = 0）：

場所：HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前：**CEIPSwitch**

値：1 = 無効、0 = 有効

指定しない場合、CEIPは有効になります。

クライアントで次のコマンドを実行してCEIPを無効にできます：

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

- **GAスイッチ**

GAを有効または無効にするレジストリ設定（デフォルト = 1）：

場所：HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前：**GASwitch**

値: 1 = 無効, 0 = 有効

指定しない場合、GA は無効になります。

クライアントで次のコマンドを実行して GA を有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "GASwitch" -d "0"
```

- **DataPersistPath**

データ永続化パスを制御するレジストリ設定 (デフォルト = /var/xdl/ceip):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: DataPersistPath

値: 文字列

このパスを設定するには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "DataPersistPath" -d
"your_path"
```

構成されたパスが存在しないかアクセスできない場合、データはデフォルトパスに保存されます。

Linux VDA から収集された CEIP データ

次の表は、収集される匿名情報の種類の例を示しています。このデータには、お客様を特定するような詳細は含まれていません。

データポイント	キー名	説明
マシン GUID	**machine_guid**	データの発信元となるマシンを識別する
ADソリューション	**ad_solution**	マシンのドメイン参加方法を示すテキスト文字列
Linuxカーネルバージョン	**kernel_version**	マシンのカーネルバージョンを示すテキスト文字列
LVDAバージョン	**vda_version**	インストールされているLinux VDAのバージョンを示すテキスト文字列。
LVDAの更新または新規インストール	**update_or_fresh_install**	現在のLinux VDAパッケージが新規インストールされているか、更新されているかを示すテキスト文字列
LVDAのインストール方法	install_method	現在のLinux VDAパッケージがMCS、PVS、簡易インストール、または手動インストールを使用してインストー

データポイント	キー名	説明
		ルされていることを示すテキスト文字列。
HDX™ 3D Proが有効かどうか	**hdx_3d_pro**	HDX 3D Proがマシンで有効になっているかどうかを示すテキスト文字列。
VDIモードが有効かどうか	**vdi_mode**	VDIモードが有効になっているかどうかを示すテキスト文字列。
システムロケール	**system_locale**	このマシンのロケールを示すテキスト文字列。
LVDAキーサービスの最終再起動時刻	**シーティーエックスエイチディーエックス シーティーエックスブイディーエー**	`ctxhdx`および`ctxvda`サービスの最終再起動時刻。形式はdd-hh:mm:ssで、例: 10-17:22:19
GPUの種類	**gpu_type**	マシンのGPUタイプを示します
CPUコア	**cpu_cores**	マシンのCPUコア数を示す整数
CPU周波数	**cpu_frequency**	MHz単位のCPU周波数を示す浮動小数点数
物理メモリサイズ	**memory_size**	KB単位の物理メモリサイズを示す整数
起動されたセッション数	**session_launch**	このデータポイントを収集した時点で、マシン上で起動された（ログオンまたは再接続された）セッション数を示す整数
Linux OS名とバージョン	**os_name_version**	マシンのLinux OS名とバージョンを示すテキスト文字列
セッションキー	**session_key**	データが生成されたセッションを識別する

データポイント	キー名	説明
リソースの種類	**resource_type**	起動されたセッションのリソースの種類（デスクトップまたは `<appname>`）を示すテキスト文字列
アクティブセッション時間	**active_session_time**	セッションのアクティブ時間を保存するために使用されます。セッションは切断/再接続できるため、1つのセッションは複数のアクティブ時間を持つことができます。
セッション継続時間	**session_duration_time**	ログオンからログオフまでのセッションの継続時間を保存するために使用されます。
Receiverクライアントの種類	**receiver_type**	セッションの起動に使用されたCitrix Workspace™アプリのタイプを示す整数
Receiverクライアントバージョン	**receiver_version**	セッションの起動に使用されたCitrix Workspaceアプリのバージョンを示すテキスト文字列
印刷回数	**printing_count**	セッションが印刷機能を使用する回数を示す整数
USBリダイレクト回数	**usb_redirecting_count**	セッションがUSBデバイスを使用する回数を示す整数
Gfxプロバイダータイプ	**gfx_provider_type**	セッションのグラフィックプロバイダータイプを示すテキスト文字列
シャドウイング回数	**shadow_count**	セッションがシャドウされた回数を表す整数
ユーザーが選択した言語	**ctxism_select**	ユーザーが選択したすべての言語を含む構成された長い文字列

データポイント	キー名	説明
スマートカードのリダイレクト回数	**scard_redirecting_count**	セッションログオンおよびセッション内アプリのユーザー認証にスマートカードのリダイレクトが使用された回数を表す整数

HDX™ インサイト

概要

Linux VDA は、HDX Insight 機能を部分的にサポートしています。

インストール

依存パッケージをインストールする必要はありません。

使用方法

HDX Insight は、Citrix Workspace™ アプリと Linux VDA の間で Citrix ADC を介して渡される ICA® メッセージを分析します。すべての HDX Insight データは NSAP 仮想チャネルから取得され、非圧縮で送信されます。NSAP 仮想チャネルはデフォルトで有効になっています。

次のコマンドは、NSAP 仮想チャネルをそれぞれ無効化および有効化します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000000" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP"
-d "0x00000001" --force
```

トラブルシューティング

データポイントが表示されない

2つの原因が考えられます。

- HDX Insight が正しく構成されていません。

たとえば、Citrix ADC で AppFlow® が有効になっていないか、Citrix ADM で誤った Citrix ADC インスタンスが構成されています。

- Linux VDA で ICA Control Virtual Channel が開始されていません。

  `ps aux | grep -i ctxctl`

  `ctxctl` が実行されていない場合は、管理者に連絡して Citrix にバグを報告してください。

アプリケーションデータポイントが表示されない

シームレス仮想チャネルが有効になっており、シームレスアプリケーションが実行されていることを確認してください。

Citrix Telemetry Service との統合

Linux VDA ソフトウェアに Citrix Telemetry Service (`ctxtelemetry`) が統合されているため、Citrix Scout を実行できます。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/2301/media/citrix-scout.png)

> **注:**
>
> Linux VDA 1912 以前のバージョンからアップグレードした後、Citrix Telemetry Service (`ctxtelemetry`) の変数を構成するには、/opt/Citrix/VDA/sbin/ctxsetup.sh を再実行する必要があります。変数について詳しくは、「簡単な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。

Citrix Telemetry Service の有効化と無効化

- サービスを有効にするには、**sudo systemctl enable ctxtelemetry.socket** コマンドを実行します。
- サービスを無効にするには、**sudo systemctl disable ctxtelemetry.socket** を実行します。

ポート

Citrix Telemetry Service (`ctxtelemetry`) は、デフォルトで TCP/IP ポート 7503 を使用して Citrix Scout をリッスンします。Delivery Controller 上では TCP/IP ポート 7502 を使用して Citrix Scout と通信します。

Linux VDA のインストール時に、デフォルトのポートを使用するか、以下の変数を通じてポートを変更できます。

- **CTX_XDL_TELEMETRY_SOCKET_PORT** – Citrix Scout をリッスンするためのソケットポート。デフォルトポートは 7503 です。
- **CTX_XDL_TELEMETRY_PORT** – Citrix Scout と通信するためのポート。デフォルトポートは 7502 です。

VDA のインストール後にポートを変更するには、次の手順を実行します。

1. Scout と通信するためのポートを変更するには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\Software\Citrix\VirtualDesktopAgent" -v "TelemetryServicePort" -d
   <port number> -t REG_DWORD
   ```

2. Scout をリッスンするためのソケットポートを変更するには、次のコマンドを実行して ctxtelemetry.socket ファイルを開き、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/2301/media/socket-port.png)

3. ソケットポートを再起動するには、次のコマンドを実行します。

   ```
   sudo systemctl daemon-reload

   sudo systemctl stop ctxtelemetry.socket

   sudo systemctl start ctxtelemetry.socket
   ```

4. ファイアウォール構成で新しいポートを有効にします。

   たとえば、Ubuntu を使用している場合は、ポート 7503 を有効にするために **sudo ufw allow 7503** コマンドを実行します。

デバッグモード

Citrix Telemetry Service が期待どおりに動作しない場合は、デバッグモードを有効にして原因を特定できます。

1. デバッグモードを有効にするには、次のコマンドを実行して `ctxtelemetry` ファイルを開き、DebugMode の値を 1 に変更します。

   ```
   sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
   ```

   デバッグモードを有効にするイメージ (/en-us/linux-virtual-delivery-agent/2301/media/enable-debug-mode.png)

2. Citrix Telemetry Service を手動で停止するか、サービスが自動的に停止するまで 15 分間待ちます。

   テレメトリを停止するイメージ (/en-us/linux-virtual-delivery-agent/2301/media/stop-telemetry.png)

   この例では、Citrix Telemetry Service を停止するために次のコマンドを実行できます。

   ```
   sudo netstat -ntlp

   Kill -9 1958
   ```

3. Citrix Telemetry Service を再起動するには、Scout で Linux VDA を選択し、/var/log/xdl/ にある telemetry-debug.log を見つけます。

サービス待機時間

ソケットポートを開く `systemd` デーモンはデフォルトで起動し、リソースをほとんど使用しません。Citrix Telemetry Service はデフォルトで停止しており、Delivery Controller から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とともにメッセージを表示します。検証テストの詳細については、Citrix Scout ドキュメントの「検証テスト」セクションを参照してください。

Linux VDA の Azure を介したセルフアップデート

この機能は、Linux VDA ソフトウェアを即時またはスケジュールされた時間に自動的に更新するのに役立ちます。Citrix DaaS Standard for Azure (旧称 Citrix Virtual Apps and Desktops Standard for Azure) で Linux VDA を作成する場合に便利です。Azure の VM の管理者権限は必要ありません。詳細については、「Citrix DaaS Standard for Azure で Linux VDA を作成する」を参照してください。

構成

この機能を使用するには、次の手順を完了します。

手順 1: 更新情報と新しい VDA パッケージを Azure コンテナーにアップロードする

手順 1a: Azure ストレージアカウントの下にコンテナーを作成し、コンテナーのアクセスレベルを **Blob (匿名読み取りアクセスのみ)** に設定します。

> **注:**

> Azure コンテナーと BLOB は、お客様が排他的に保持および管理します。Citrix® は、それらに関するセキュリティ問題について責任を負いません。データセキュリティとコスト効率を確保するため、各 **セルフアップデート** の後でコンテナーのアクセスレベルを **Private (匿名アクセスなし)** に設定してください。

> 手順 1b: VDA 更新情報を UpdateInfo.json という名前の JSON ファイルに組み込みます。ファイル形式の例については、次のブロックを参照してください。

```
{

 "Version": "21.04.200.4",

"Distributions":[

{

"TargetOS": "RHEL7_9",

"PackageName": "",
```

```
"PackageHash": ""

},

{

"TargetOS": "UBUNTU18_04",

"PackageName": "xendesktopvda_21.04.200.4-1.ubuntu18.04_amd64.deb",

"PackageHash":
"4148cc3f25d3717e3cbc19bd953b42c72bd38ee3fcd7f7034c2cd6f2b15b3c5a"

},

{

"TargetOS": "UBUNTU20_04",

"PackageName": "",

"PackageHash": ""

}

]

}
```

ここで、**"Version"** は新しい VDA バージョンを示し、**"Distributions"** は更新オブジェクトの配列です。各オブジェクトには次の 3 つの項目が含まれます。

- **"TargetOS"**: 「RHEL7_9」（RHEL 7、CentOS 7、Amazon Linux 2 の場合）、「UBUNTU18_04」、または「UBUNTU20_04」である必要があります。`ctxmonitorservice` は他のディストリビューションを認識しません。
- **"PackageName"**: 指定されたバージョンの VDA パッケージの完全な名前。
- **"PackageHash"**: `shasum -a 256 <pkgname>` コマンドを使用して計算する SHA-256 値。

手順 1c: JSON ファイルと新しいバージョンの Linux VDA パッケージを Azure コンテナーにアップロードします。

手順 2: マスターイメージまたは各 VDA で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。

レジストリ設定	説明
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、`ctxmonitorservice`の次回の起動時にアップデートを実行することを意味します。このオプションは、ダウンロード速度が速くなく、アップデートが緊急ではない場合に適しています。
CaCertificate	この設定はオプションです。AzureコンテナのURLを検証するためのPEM証明書のフルパスを設定します。Azure BLOBの場合、ブラウザから取得しPEMに変換されたportal.azure.comの証明書を使用できます。セキュリティのため、このレジストリ設定を追加することをお勧めしますが、Ubuntuでのみサポートされています。RHELでは、`curl`コマンド用の一部のNSSライブラリのリンクが不足しています。証明書の最小限の権限を設定するようにしてください。

`ctxmonitorservice`が再起動すると、まず**Url**をクエリしてUpdateInfo.jsonファイルを取得し、JSONファイルから更新バージョンを取得します。次に、`ctxmonitorservice`は更新バージョンを現在のバージョンと比較します。現在のバージョンが以前のものである場合、サービスはAzureから新しいバージョンのVDAパッケージをダウンロードし、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、`ctxmonitorservice`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、`ctxmonitorservice`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。
- 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てもロールバックが成功した場合、同じネットワーク上のユーザーは異なるバージョンのLinux VDAを使用する可能性があります。このケースは最適ではありません。
- 更新には通常、数分かかります。Citrix Studio にはステータスインジケーターはありません。

Linux VMとLinuxセッションのメトリック

次の表に、Linux VMおよびLinuxセッションで利用可能なメトリックをいくつか示します。

メトリック	最小VDAバージョン要件	説明	備考
ログオン期間	2109	これは、ユーザーがCitrix Workspaceアプリから接続してからセッションが使用可能になるまでのログオンプロセスを測定したものです。セッションのメトリックを表示するには、Citrix DaaS（旧Citrix Virtual Apps and Desktopsサービス）のMonitorタブに移動します。Monitorは、Citrix Virtual Apps and Desktops	Monitorでのみ利用可能です。

メトリック	最小VDAバージョン要件	説明	備考
		のCurrent ReleaseおよびLTSR展開を監視およびトラブルシューティングするためのDirectorコンソールとして利用できます。Monitorタブで、**Average Logon Duration**セクションの**View Historical Trend**をクリックします。**Logon Performance**ページで、フィルター条件を設定し、**Apply**をクリックしてメトリックを視覚化します。	
セッションの自動再接続回数	2109	セッション内の自動再接続の数を表示するには、**Trends**ビューにアクセスします。条件を設定し、**Apply**をクリックして検索結果を絞り込みます。**Session Auto Reconnect Count**列には、セッション内の自動再接続の数が表示されます。自動再接続は、**Session Reliability**または**Auto Client Reconnect**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、	Citrix DirectorとMonitorの両方で利用可能です。

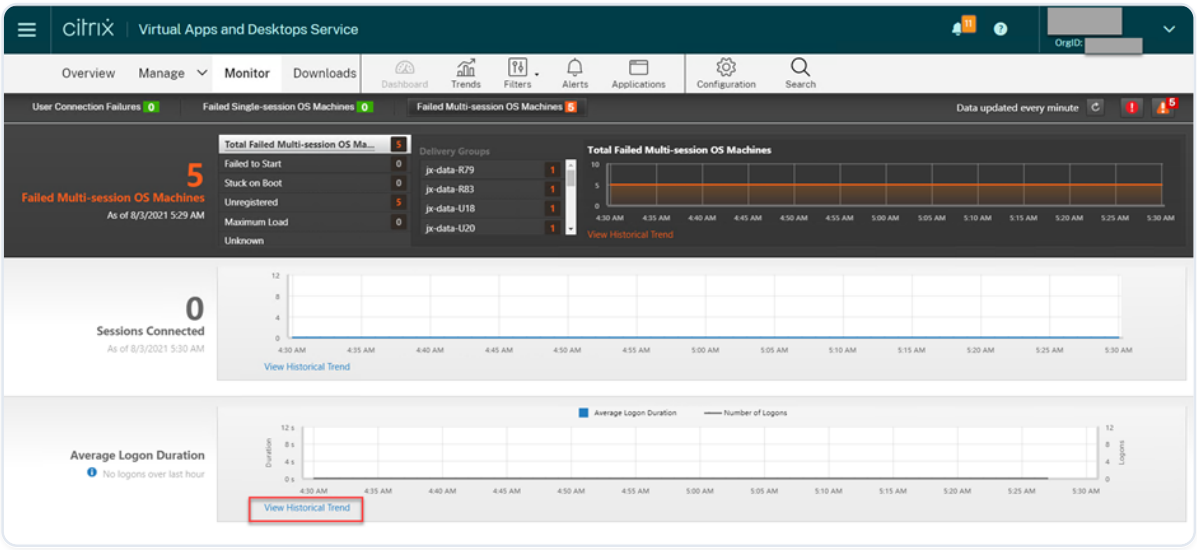
メトリック	最小VDAバージョン要件	説明	備考
		Sessionsを参照してください。ポリシーの詳細については、Auto client reconnect policy settingsおよびSession reliability policy settingsを参照してください。	
アイドル時間	2103	このメトリックにアクセスするには、**Filters > Sessions > All Sessions**を選択して**All Sessions**ページを開きます。	Citrix DirectorとMonitorの両方で利用可能です。
Linux VMのメトリック	2103	Linux VMでは、CPUコア数、メモリサイズ、ハードディスク容量、および現在と過去のCPUおよびメモリ使用率などのメトリックが利用可能です。	Citrix DirectorとMonitorの両方で利用可能です。
プロトコル	1909	Linuxセッションのトランスポートプロトコルは、**セッションの詳細**パネルにUDPまたはTCPとして表示されます。	Citrix DirectorとMonitorの両方で利用可能です。
ICA® RTT	1903	ICAラウンドトリップ時間（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**	Citrix DirectorとMonitorの両方で利用可能です。

メトリック	最小VDAバージョン要件	説明	備考
		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。	

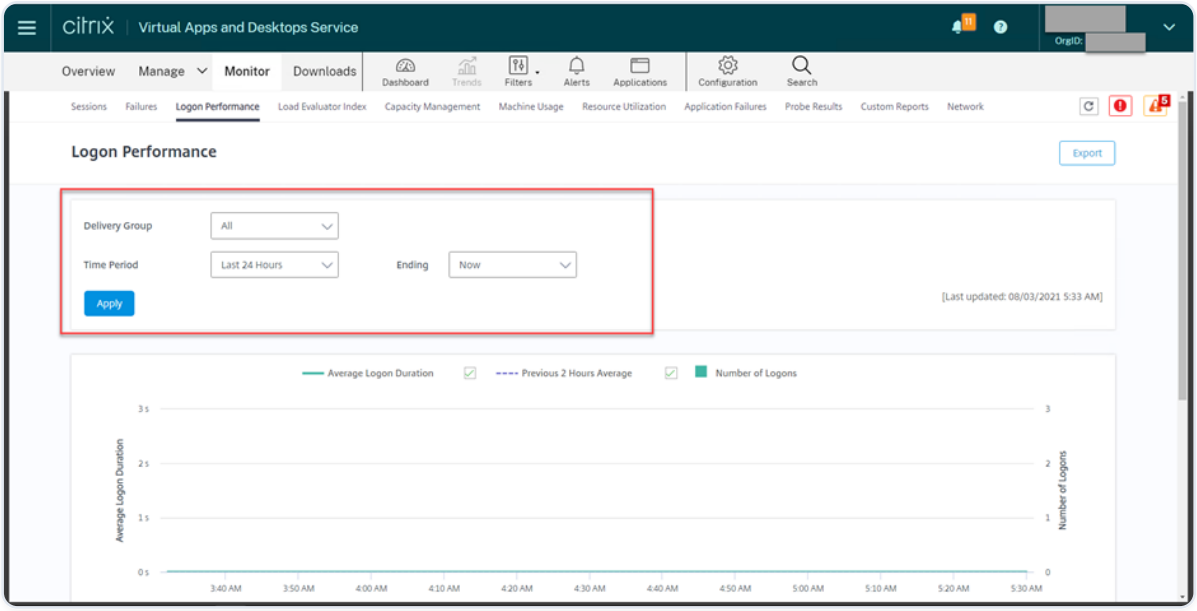
Citrix DirectorおよびMonitorでさまざまなメトリックにアクセスする方法の例

- ログオン期間

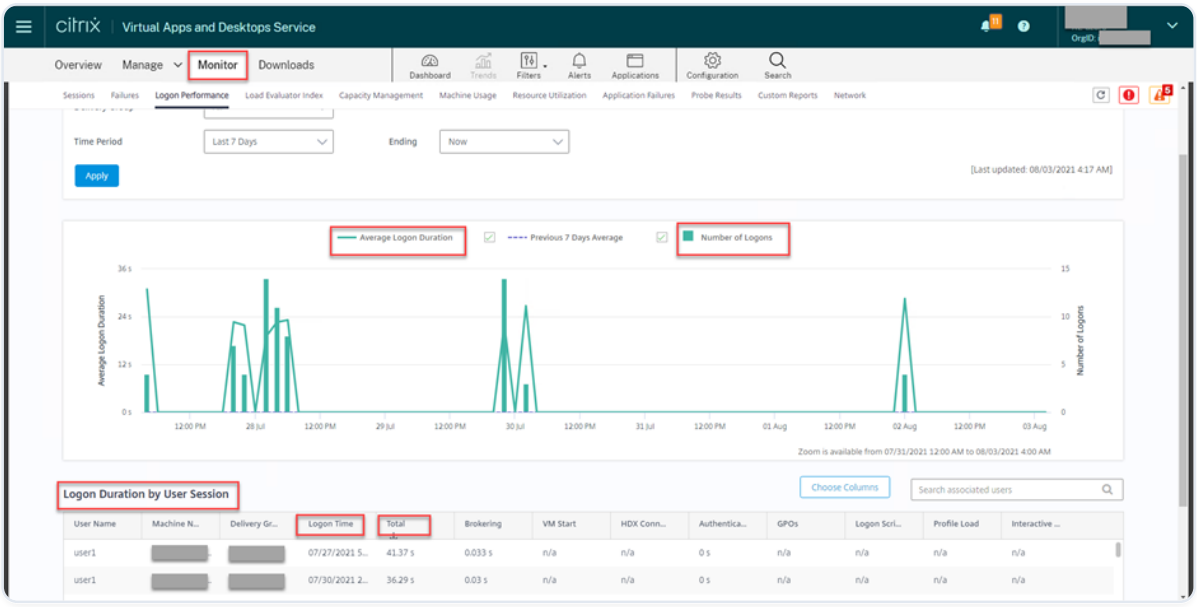
a. Citrix DaaSのMonitorタブで、**平均ログオン期間**セクションの**履歴トレンドの表示**をクリックします。



b. **ログオンパフォーマンス**ページで、フィルター条件を設定します。

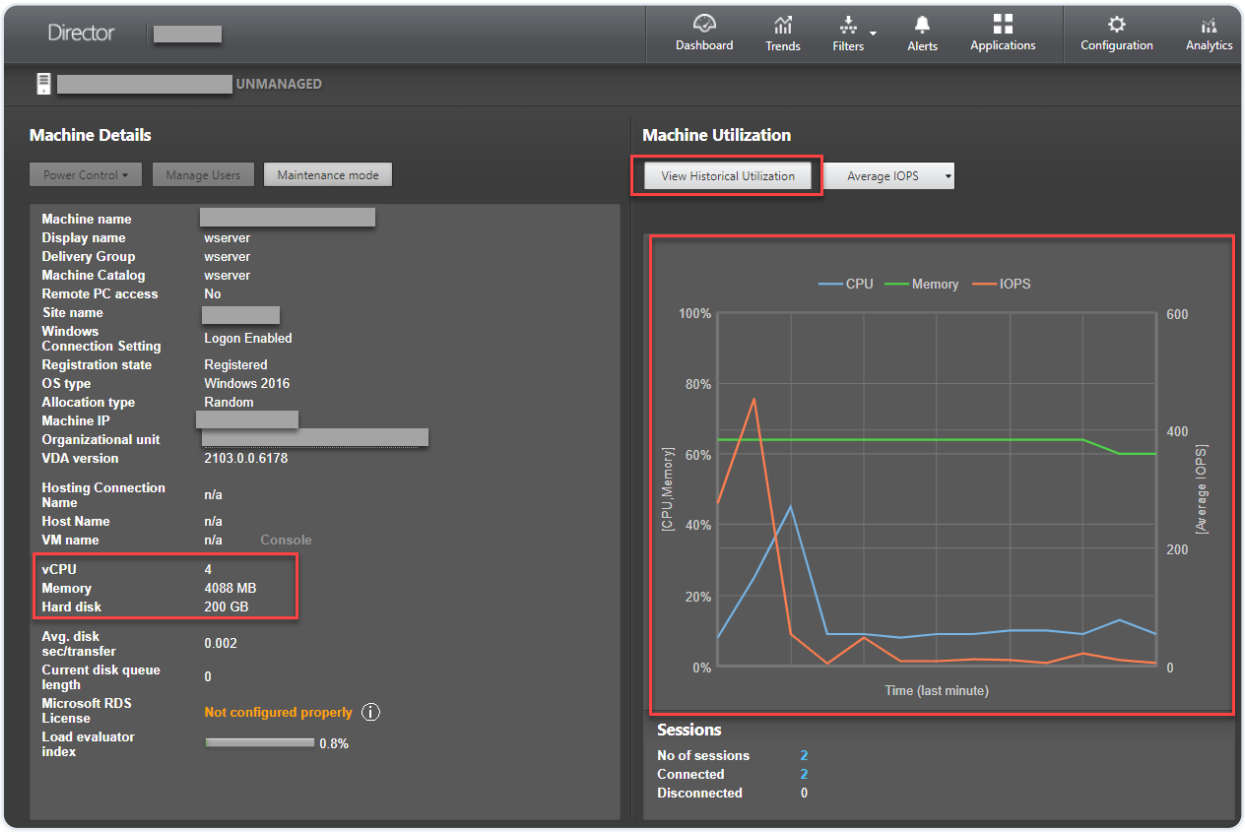


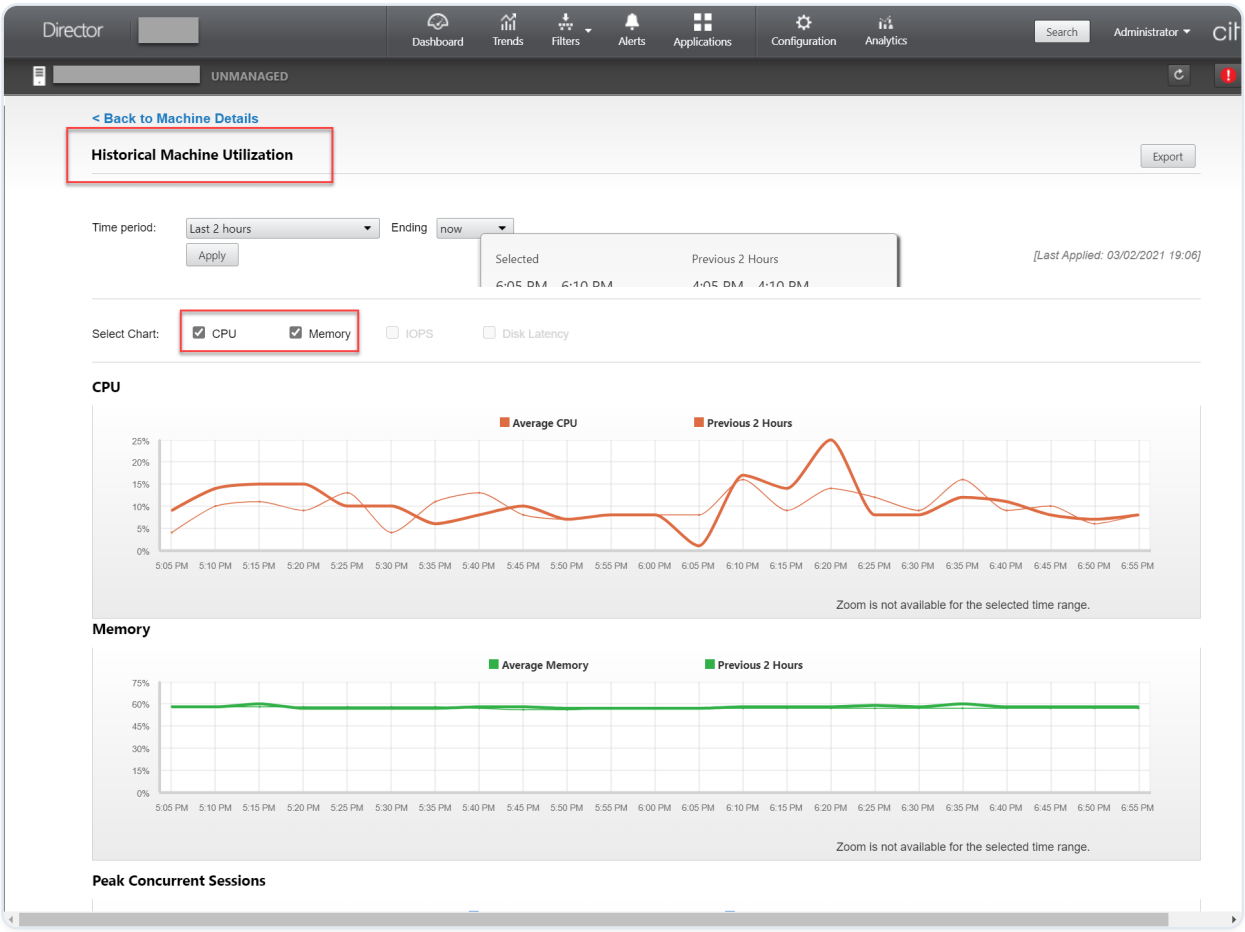
c. ログオン期間メトリックを視覚化するには、**適用**をクリックします。



- Linux VMのCPUコア数、メモリサイズ、ハードディスク容量、および現在と過去のCPUとメモリの使用率

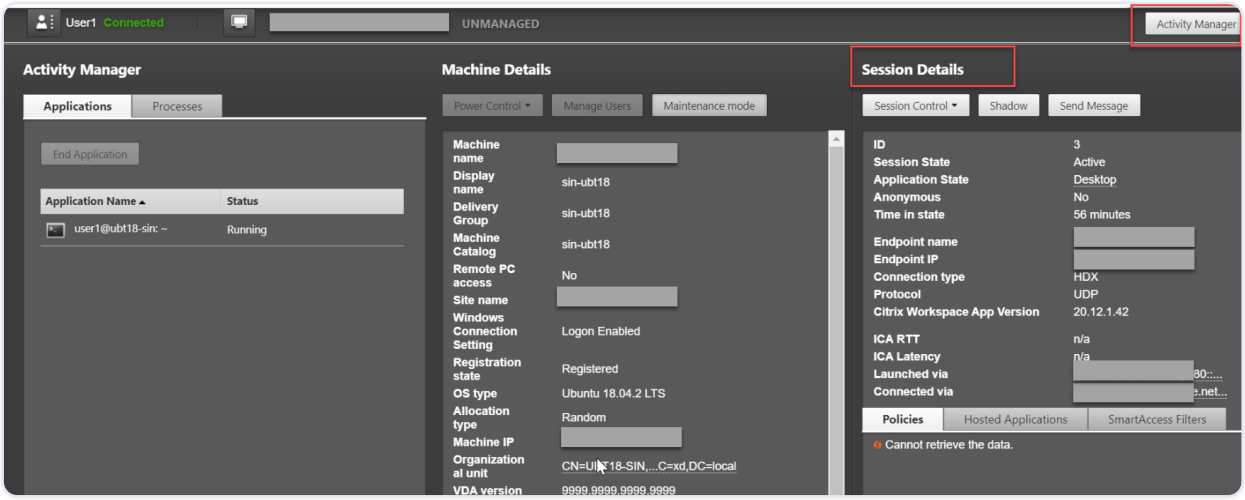
  Linux VMのこれらのメトリックにアクセスするには、Citrix DirectorまたはMonitorでVMを見つけ、**マシンの詳細**パネルを確認します。例：





- ICA RTT、プロトコル

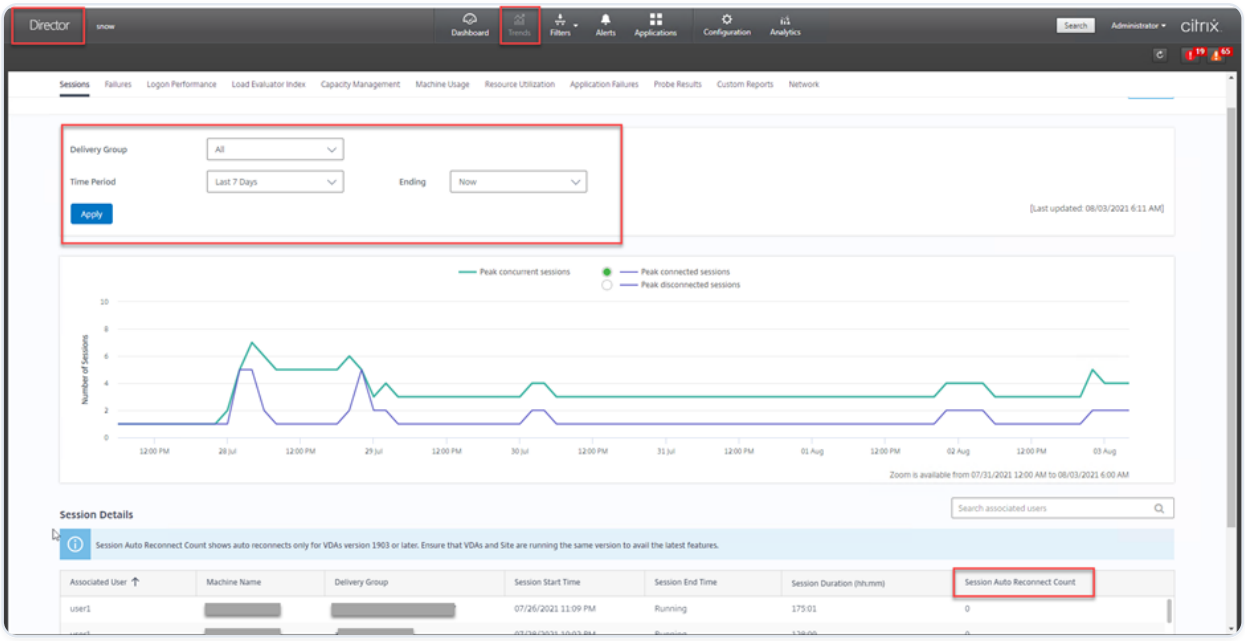
  Linuxセッションのメトリックを表示するには、**フィルター > セッション > すべてのセッション**を選択して**すべてのセッション**ページを開くか、**セッションの詳細**パネルにアクセスします。**セッションの詳細**パネルにアクセスするには、**すべてのセッション**ページを開き、目的のセッションをクリックして、その**アクティビティマネージャー**ビューにアクセスします。例：



- セッションの自動再接続回数

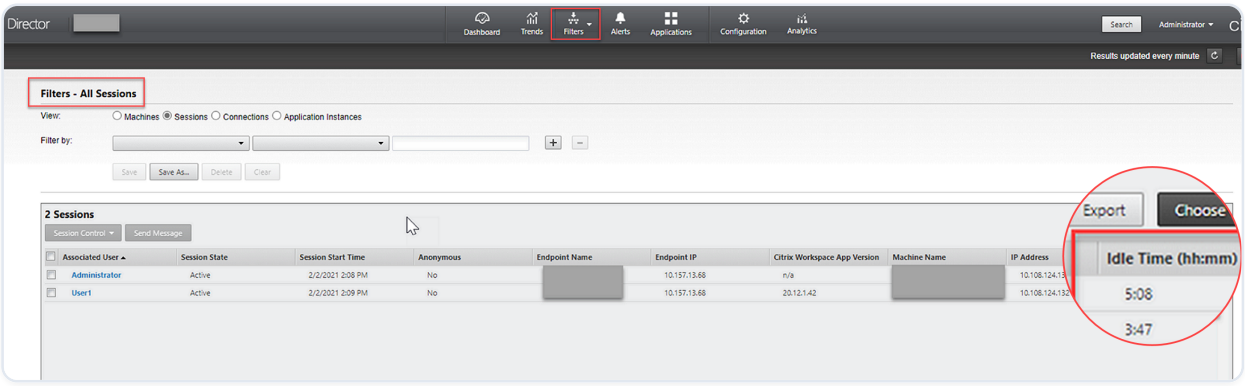
セッションでの自動再接続の回数を表示するには、**トレンド**ビューにアクセスします。条件を設定し、**適用**をクリックして検索結果を絞り込みます。

セッションの自動再接続回数列には、セッションでの自動再接続の回数が表示されます。例：



- アイドル時間

  例:



ログ収集

概要

ログ収集は、デフォルトでLinux VDAに対して有効になっています。

構成

`ctxlogd`デーモンと`setlog`ユーティリティは、Linux VDAリリースパッケージに含まれています。デフォルトでは、Linux VDAのインストールと構成後に`ctxlogd`デーモンが起動します。

ctxlogdデーモン

トレースされる他のすべてのサービスは、`ctxlogd`デーモンに依存します。Linux VDAをトレースしたくない場合は、`ctxlogd`デーモンを停止できます。

setlogユーティリティ

ログ収集は、**/opt/Citrix/VDA/bin/**パスにある`setlog`ユーティリティを使用して構成されます。rootユーザーのみが実行する権限を持っています。GUIを使用するか、コマンドを実行して構成を表示および変更できます。`setlog`ユーティリティに関するヘルプについては、次のコマンドを実行してください。

```
setlog help
```

値

デフォルトでは、**ログ出力パス**は**/var/log/xdl/hdx.log**に設定され、**最大ログサイズ**は200 MBに設定されており、**ログ出力パス**の下に最大2つの古いログファイルを保存できます。

現在の`setlog`値を表示します。

```
setlog values


log_path (Log Output Path) = /var/log/xdl/hdx.log


log_size (Max Log Size (MiB)) = 200


log_count (Max Old Log Files) = 2
```

単一の`setlog`値を表示または設定します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

デフォルトでは、ログレベルは**warning**（大文字と小文字を区別しない）に設定されています。

異なるコンポーネントに設定されているログレベルを表示するには、次のコマンドを実行します。

```
setlog levels
```

ログレベル（Disabled、Inherited、Verbose、Information、Warnings、Errors、Fatal Errorsを含む）を設定するには、次のコマンドを実行します。

```
setlog level <class> [<level>]
```

ログレベル	コマンドパラメーター（大文字と小文字を区別しない）
無効	なし
継承済み	継承
詳細	詳細
情報	情報
警告	警告
エラー	エラー
致命的なエラー	致命的

`<class>` 変数は、Linux VDA の 1 つのコンポーネントを指定します。すべてのコンポーネントを対象とするには、all に設定します。例:

```
setlog level all error
```

フラグ

デフォルトでは、フラグは次のように設定されています:

```
setlog flags


DATE = true


TIME = true


NAME = true


PID = true


TID = false


SID = true


UID = false


GID = false


CLASS = false


LEVEL = false
```

```
FUNC = true



FILE = false
```

現在のフラグを表示:

```
setlog flags
```

単一のログフラグを表示または設定:

```
setlog flag <flag> [<state>]
```

デフォルトに戻す

すべてのレベル、フラグ、および値をデフォルト設定に戻します。

```
setlog default
```

> **重要:**
>
> `ctxlogd` サービスは、ルートユーザーのみが作成できる**/var/xdl/.ctxlog**ファイルを使用して構成されます。他のユーザーはこのファイルへの書き込み権限がありません。ルートユーザーは他のユーザーに書き込み権限を与えないことをお勧めします。これに従わない場合、任意のまたは悪意のある構成が `ctxlogd` になる可能性があり、その結果、サーバーのパフォーマンス、ひいてはユーザーエクスペリエンスに影響を与える可能性があります。

トラブルシューティング

/var/xdl/.ctxlogファイルが見つからない場合（たとえば、誤って削除された場合）、`ctxlogd` デーモンが失敗し、`ctxlogd` サービスを再起動できません。

/var/log/messages:

```
Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
configuration file.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited,
code=exited, status=1/FAILURE



Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
```

この問題を解決するには、ルートユーザーとして`setlog`を実行して**/var/xdl/.ctxlog**ファイルを再作成します。その後、他のサービスが依存している`ctxlogd`サービスを再起動します。

セッションのシャドウイング

セッションのシャドウイングにより、ドメイン管理者はイントラネットでユーザーのICA®セッションを表示できます。この機能は、noVNCを使用してICAセッションに接続します。

> **注:**
>
> この機能を使用するには、`Citrix Director` 7.16以降を使用してください。

インストールと構成

依存関係

セッションのシャドウイングには、`python-websockify`と`x11vnc`の2つの新しい依存関係が必要です。Linux VDAのインストール後に、`python-websockify`と`x11vnc`を手動でインストールしてください。

RHEL 7.xおよびAmazon Linux2の場合:

`python-websockify`と`x11vnc`（`x11vnc`バージョン0.9.13以降）をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo yum install x11vnc
```

(RHEL 7.xの場合) Extra Packages for Enterprise Linux (EPEL) およびオプションのRPMリポジトリを有効にして、`python-websockify`と`x11vnc`を解決します。

- EPEL

  `x11vnc`にはEPELリポジトリが必要です。EPELリポジトリを有効にするには、次のコマンドを実行します。

  ```
  yum install https://dl.fedoraproject.org/pub/epel/epel-release-
  latest-7.noarch.rpm
  ```

- オプションのRPM

  `x11vnc`の一部の依存パッケージをインストールするために、オプションの`RPMs`リポジトリを有効にするには、次のコマンドを実行します。

  ```
  subscription-manager repos --enable rhel-7-server-optional-rpms --
  enable rhel-7-server-extras-rpms
  ```

RHEL 8.x/9.0 および Rocky Linux 8.x/9.0 の場合:

`python-websockify`と`x11vnc`（`x11vnc`バージョン0.9.13以降）をインストールするには、次のコマンドを実行します。

```
sudo pip3 install websockify

sudo yum install x11vnc
```

EPELおよびCodeReady Linux Builderリポジトリを有効にして、`x11vnc`を解決します。

```
dnf install -y --nogpgcheck https://dl.fedoraproject.org/pub/epel/epel-
release-latest-8.noarch.rpm
```

```
subscription-manager repos --enable "codeready-builder  -for-rhel-8-
x86_64-rpms"
```

Ubuntuの場合:

`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:

`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 zypper install x11vnc
```

Debian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
```

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の範囲内で利用可能なポートを自動的に選択します。そのため、同時にシャドウできるICAセッションの数は99に制限されます。特にマルチセッションシャドウイングの場合、要件を満たすのに十分なポートが利用可能であることを確認してください。

レジストリ

次の表は、関連するレジストリを示しています:

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では、`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://`を使用するLinux VDAセッションシャドウイングに対するいかなるセキュリティ責任も負いません。

SSLを有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "ShadowingUseSSL" -d
"0x00000001"
```

サーバーおよびルートSSL証明書の取得

証明書は、信頼された認証局（CA）によって署名されている必要があります。

SSLを構成する各Linux VDAサーバーには、個別のサーバー証明書（キーを含む）が必要です。サーバー証明書は特定のコンピューターを識別するため、各サーバーの完全修飾ドメイン名（FQDN）を知っている必要があります。便宜上、ドメイン全体にワイルドカード証明書を使用することを検討してください。

Linux VDAと通信する各Citrix Directorクライアントにもルート証明書が必要です。ルート証明書は、サーバー証明書を発行するのと同じCAから入手できます。

サーバー証明書とクライアント証明書は、次のCAからインストールできます。

- オペレーティングシステムにバンドルされているCA
- エンタープライズCA（組織がアクセスを許可しているCA）
- オペレーティングシステムにバンドルされていないCA

証明書を取得するためにどの方法が必要かについては、組織のセキュリティチームに相談してください。

重要:

- サーバー証明書の共通名（Common Name）は、Linux VDAの正確なFQDNであるか、少なくとも正しいワイルドカードとドメイン文字である必要があります。例: vda1.basedomain.com または *.basedomain.com。
- SHA1やMD5を含むハッシュアルゴリズムは、一部のブラウザーがサポートするにはデジタル証明書の署名として弱すぎます。そのため、SHA-256が最小標準として指定されています。
- Chrome は自己署名 SSL 証明書を安全でないとみなし、受け入れなくなりました。`NET::ERR_CERT_COMMON_NAME_INVALID` エラーは、生成された証明書に SAN (subjectAltName) フィールドがないために発生します。この問題を解決するには、SAN フィールドを含む拡張属性 (X509 v3 拡張) を持つ証明書を提供してください。

各 CITRIX DIRECTOR クライアントにルート証明書をインストールする

セッションシャドウイングは、IIS と同じレジストリベースの証明書ストアを使用するため、IIS または Microsoft 管理コンソール (MMC) の証明書スナップインを使用してルート証明書をインストールできます。CA から証明書を受け取ったら、IIS で Web サーバー証明書ウィザードを再起動すると、ウィザードが証明書をインストールします。または、MMC を使用してコンピューター上の証明書を表示およびインポートし、スタンドアロンの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、オペレーティングシステムにインストールされている証明書をデフォルトでインポートします。Mozilla Firefox の場合は、証明書マネージャーの **［認証局］** タブでルート SSL 証明書をインポートする必要があります。

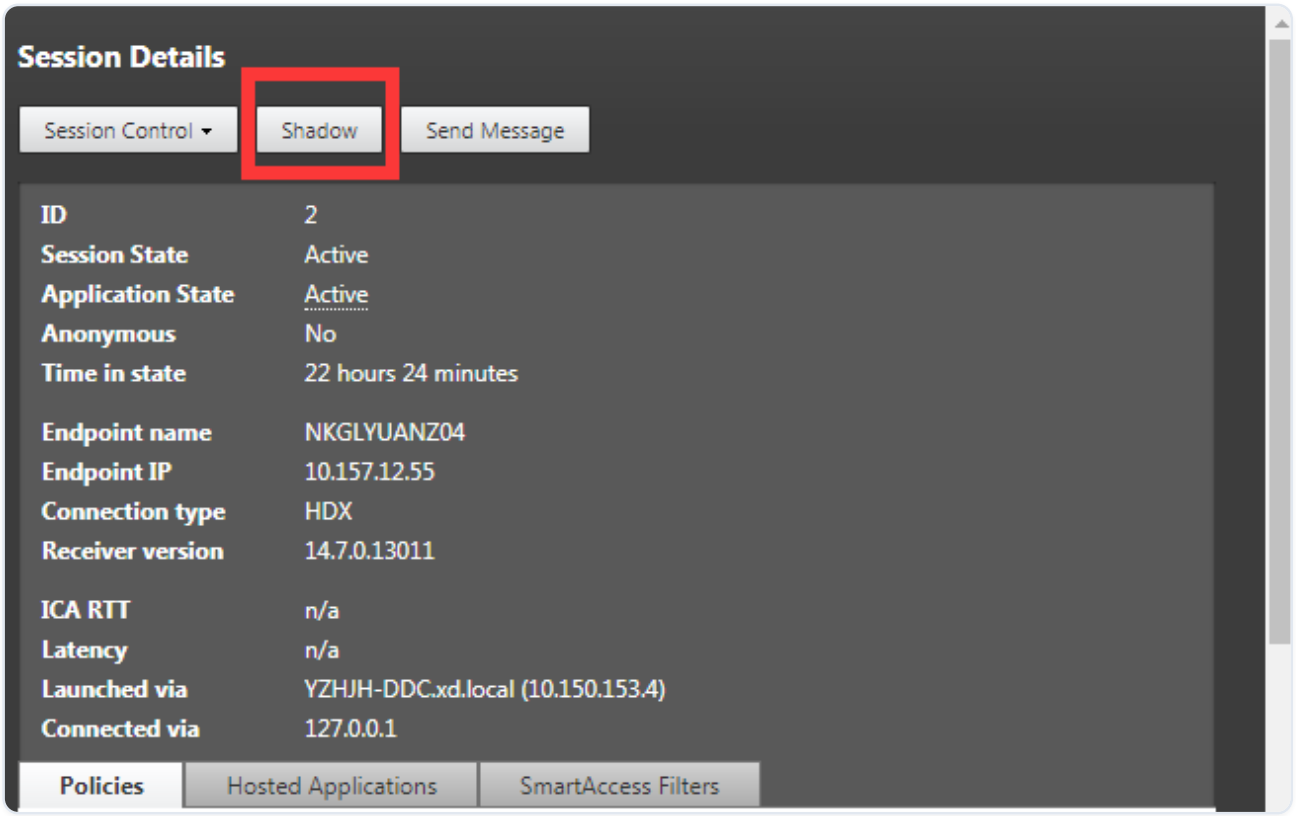
各 LINUX VDA サーバーにサーバー証明書とそのキーをインストールする

サーバー証明書には「shadowingcert.*」、キーファイルには「shadowingkey.*」という名前を付けます（* は形式を示します。例：shadowingcert.pem 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、制限された権限で適切に保護して

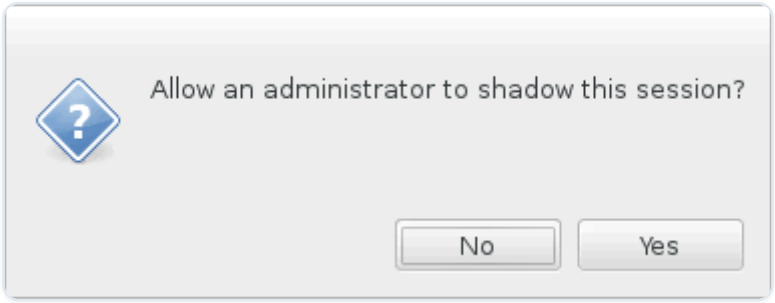
ください。名前またはパスが正しくないと、Linux VDA は特定の証明書またはキーファイルを見つけることができず、`Citrix Director` との接続障害が発生します。

使用方法

`Citrix Director` から、ターゲットセッションを見つけ、**［セッションの詳細］** ビューで **［シャドウ］** をクリックして、Linux VDA にシャドウイング要求を送信します。



接続が初期化されると、ICA セッションクライアント（`Citrix Director` クライアントではない）に確認が表示され、ユーザーにセッションをシャドウする許可を求めます。



ユーザーが **［はい］** をクリックすると、`Citrix Director` 側にウィンドウが表示され、ICA セッションがシャドウされていることを示します。

使用方法の詳細については、「Citrix Director Documentation」を参照してください。

制限事項

- セッションシャドウイングは、イントラネット内でのみ使用するように設計されています。Citrix Gateway 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関するいかなる責任も負いません。
- セッションシャドウイングが有効になっている場合、ドメイン管理者は ICA セッションを表示することしかできず、書き込みや制御の権限はありません。
- 管理者が `Citrix Director` から **［シャドウ］** をクリックすると、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。
- 前述の確認にはタイムアウト制限があり、20 秒です。時間が経過すると、シャドウイング要求は失敗します。
- 1つのセッションは1人の管理者のみがシャドウできます。たとえば、管理者Bが管理者Aがシャドウしているセッションに対してシャドウ要求を送信すると、ユーザーデバイスにユーザー権限取得の確認が再表示されます。ユーザーが同意した場合、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が構築されます。管理者が同じセッションに対して別のシャドウ要求を送信した場合でも、新しいシャドウ接続を構築できます。
- セッションシャドウを使用するには、`Citrix Director` 7.16以降をインストールします。
- `Citrix Director` クライアントは、ターゲットのLinux VDAサーバーに接続するために、IPアドレスではなくFQDNを使用します。したがって、`Citrix Director` クライアントはLinux VDAサーバーのFQDNを解決できる必要があります。

トラブルシューティング

セッションシャドウが失敗した場合は、`Citrix Director` クライアントとLinux VDAの両方でデバッグを行います。

Citrix Directorクライアントで

ブラウザの開発者ツールを使用して、**Console**タブで出力ログを確認します。または、**Network**タブでShadowLinuxSession APIの応答を確認します。ユーザー権限取得の確認が表示されるものの、接続の確立に失敗する場合は、VDAのFQDNを手動でpingして、`Citrix Director` がFQDNを解決できることを確認します。`wss://` 接続に問題がある場合は、証明書を確認してください。

Linux VDA で

1. `/var/log/xdl/vda.log` ファイルで手がかりを確認します。

2. `/var/xdl/sessionshadowing.sh`ファイルを編集し、「logFile」変数を変更して、Directorからのセッションシャドウ中に手がかりを追跡できるログファイルを指定します。
3. また、noVNC接続で証明書が正しく機能するかどうかを手動で確認することもできます。

   a. **ps aux \| grep xorg**を実行して、現在のセッションのXorgディスプレイ番号`<display-num>`（例: `:3`）を見つけます。

   b. 次のコマンドを実行してx11vncサーバーを起動し、受信接続を待ちます。

   ```
   runuser -l "ctxsrvr" -s /bin/bash -c "websockify <port> -v --cert  /
   etc/xdl/shadowingssl/shadowingcert.pem --key  /etc/xdl/shadowingssl/
   shadowingkey.key -- x11vnc  -viewonly  -shared -passwd <passwd>
   -rfbport <port> -display <display-num> -many -o /var/log/xdl/
   x11vnc.log"
   ```

   c. 次のようにnoVNCを使用して接続し、SSLモードを確認します。VDAのFQDNとポート番号を入力します。この例では、ポート番号は6009です。

   noVNCを使用して接続(/en-us/linux-virtual-delivery-agent/current-release/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によって報告されたエラーを解決します。

   接続確立時の主要なチェックポイント:

      i. セッションシャドウイングがポートを開くのを妨げるファイアウォール制限がないか確認します。

      ii. SSLシナリオの場合、証明書とキーファイルが適切に命名され、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の間に十分なポートが残っていることを確認します。

      iv. `/var/xdl/sessionshadowing.sh`で必要となるため、「netstat」がインストールされていることを確認します。

      v. 証明書が正しく構成されていることを確認するために`openssl x509 -in shadowingcert.pem -text -noout`を実行し、特にCNフィールドとSANフィールドに注意してください。

      vi. RHEL 8では、`rebind.so`が見つからないという問題が発生する可能性があります。この問題を解決するには、次のコマンドを実行します。

```
ln -s /usr/bin/rebind.so /usr/local/bin/rebind.so
```

監視サービスデーモン

監視サービスデーモンは、定期的なスキャンを実行して主要なサービスを監視します。例外を検出すると、デーモンはサービスプロセスを再起動または停止し、リソースを解放するためにプロセスの残りをクリーンアップします。検出された例外は、**/var/log/xdl/ms.log** ファイルに記録されます。

構成

監視サービスデーモンは、VDAを起動すると自動的に開始されます。

管理者権限で、**/opt/Citrix/VDA/sbin** の下にある **scanningpolicy.conf**、**rulesets.conf**、および **whitelist.conf** ファイルを介してこの機能を構成できます。

scanningpolicy.conf、**rulesets.conf**、および **whitelist.conf** ファイルでの変更を有効にするには、次のコマンドを実行して監視サービスデーモンを再起動します。

```
service ctxmonitorservice restart
```

- **scanningpolicy.conf**

  この構成ファイルは、監視サービスデーモンを有効または無効にします。サービス検出間隔を設定し、検出された例外を修復するかどうかを指定します。

  - MonitorEnable: true/false (デフォルトはtrue)
  - DetectTime: 20 (単位: 秒、デフォルト値: 20、最小値: 5)
  - AutoRepair: true/false (デフォルトはtrue)
  - MultBalance: フォールス
  - ReportAlarm: フォールス

- **rulesets.conf**

  この構成ファイルは、監視するターゲットサービスを指定します。次のスクリーンショットに示すように、デフォルトで4つの監視対象サービスがあります。

  デフォルトで監視される4つのサービス (/en-us/linux-virtual-delivery-agent/2301/media/four-defaut-monitored-services.png)

監視する各サービスを設定するには、次のフィールドを設定します。

- **MonitorUser**: すべて
- **MonitorType**: 3
- **ProcessName**: <> (プロセス名は空白にできず、正確に一致する必要があります。)
- **Operation**: 1/2/4/8 (1 = 例外が検出されたときにサービスを停止する。2 = 例外が検出されたときにサービスを強制終了する。4 = サービスを再起動する。8 = Xorg プロセス残存物をクリアする。)
- **DBRecord**: 無効

- **whitelist.conf**

  rulesets.conf ファイルで指定されたターゲットサービスは、**whitelist.conf** ファイルでも構成する必要があります。ホワイトリスト構成は、セキュリティのための二次フィルターです。

  ホワイトリストを構成するには、**whitelist.conf** ファイルにプロセス名のみを含めます（プロセス名は正確に一致する必要があります）。例については、次のスクリーンショットを参照してください。

```
ctxcdmd
ctxcdmmount
ctxcdmstat
ctxceip
ctxclipboard
ctxconnect
ctxcredentialctl
ctxctl
ctxcupsd
ctxdisconnect
ctxeuem
ctxfiletransfer
ctxgfx
ctxhdx
ctxism
ctxlogd
ctxlogin
ctxmonitorservice
ctxmrvc
ctxpolicyd
ctxscardsd
ctxvhcid
ctxvda
Xorg
```

> **注:**

> `ctxvda`、`ctxhdx`、および `ctxpolicyd` サービスを停止する前に、`service ctxmonitorservice stop` コマンドを実行してモニターサービスデーモンを停止してください。そうしないと、モニターサービスデーモンは停止したサービスを再起動します。

ツールとユーティリティ

セッションデータクエリユーティリティ

各Linux VDA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`) を提供しています。VDAでホストされているすべてのセッションまたは特定のセッションの以下のデータを照会するには、`/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]` コマンドを実行します。`[-c]` 引数は、1秒ごとにデータを照会することを意味します。

- **入力セッション帯域幅**
- **出力セッション帯域幅**
- **出力セッション回線速度**
- **遅延 - 最終記録**
- **ラウンドトリップ時間**
- **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**

xdlcollect Bashスクリプト

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。Linux VDAをインストールした後、`bash /opt/Citrix/VDA/bin/xdlcollect.sh` コマンドを実行してログを収集できます。ログの収集が完了すると、スクリプトと同じフォルダーに圧縮ログファイルが生成されます。`xdlcollect` Bashスクリプトは、圧縮ログファイルをCitrix Insight Services (CIS) にアップロードするかどうかを尋ねることができます。同意すると、アップロード完了後に `xdlcollect` がupload_IDを返します。アップロードによって、ローカルマシンから圧縮ログファイルが削除されることはありません。他のユーザーは、upload_IDを使用してCISのログファイルにアクセスできます。

エックスディーピング

Linux **XDPing** ツールはコマンドラインアプリケーションです。Linux VDA 環境における一般的な構成の問題を自動的にチェックします。

Linux **XDPing** ツールは、システム上で150を超える個別のテストを実行し、それらは大まかに次のように分類されます。

- Linux VDA のシステム要件が満たされているかを確認する
- 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する
- Linux カーネルの互換性を確認する
- Linux VDA の動作に影響を与える可能性のある既知の Linux ディストリビューションの問題がないか確認する
- Security-Enhanced Linux (SELinux) のモードと互換性を確認する
- ネットワークインターフェイスを特定し、ネットワーク設定を確認する
- ストレージのパーティション分割と利用可能なディスク容量を確認する
- マシンのホスト名とドメイン名の構成を確認する
- DNS 構成を確認し、ルックアップテストを実行する
- 基盤となるハイパーバイザーを特定し、仮想マシンの構成を確認する。サポート対象:
  - シトリックス ハイパーバイザー™
  - マイクロソフト ハイパーブイ
  - ヴイエムウェア ブイスフィア
- 時刻設定を確認し、ネットワーク時刻同期が動作しているかを確認する
- PostgreSQL サービスが適切に構成され、動作しているかを確認します。
- ファイアウォールが有効になっており、必要なポートが開いているかを確認します。
- Kerberos の構成を確認し、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を確認します。サポート対象:
  - サンバ・ウィンバインド
  - Dell Quest 認証サービス
  - セントリファイ ダイレクトコントロール

    - SSSD

- 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を確認します。
- Linux VDA パッケージを特定し、インストールの整合性を確認します。
- PostgreSQL レジストリデータベースの整合性を確認します。
- Linux VDAサービスが適切に構成され、動作しているかを確認します。
- VDAおよびHDX™構成の整合性を確認します。
- 各構成済みDelivery Controller™をプローブして、Broker Serviceが到達可能で、動作しており、応答性があることをテストします。
- マシンがDelivery Controllerファームに登録されているかを確認します。
- アクティブまたは切断された各HDXセッションの状態を確認します。
- Linux VDA関連のエラーと警告についてログファイルをスキャンします。
- Xorgのバージョンが適切であるかを確認します。

Linux XDPingツールを使用する

> **注:**
>
> ctxsetup.shを実行しても**XDPing**はインストールされません。**XDPing**をインストールするには、`sudo /opt/Citrix/VDA/bin/xdping`を実行します。 このコマンドは、**XDPing**に必要なPython3仮想環境も作成します。このコマンドがPython3仮想環境の作成に失敗した場合は、Create a Python3 virtual environmentの指示に従って手動で作成してください。

> pipツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼済みホストを/etc/pip.confファイルに追加することを検討してください。

```
[global] trusted-host =
```

```
pypi.org files.pythonhosted.org
```

> **XDPing**には、コマンドシェルから実行される`xdping`という単一の実行可能ファイルが付属しています。

コマンドラインオプションを表示するには、`-h`オプションを使用します。

```
sudo /opt/Citrix/VDA/bin/xdping -h
```

全てのテストスイートを実行するには、コマンドラインオプションなしで`xdping`を実行します。

```
sudo /opt/Citrix/VDA/bin/xdping
```

Linux VDAパッケージをインストールする前に環境を確認するには、`pre-flight`テストを実行します。

```
sudo /opt/Citrix/VDA/bin/xdping --preflight
```

特定のテストカテゴリのみ（例：時間テストとKerberosテスト）を実行するには、`-T`オプションを使用します。

```
sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos
```

特定のXenDesktop® Controllerを調査するには：

```
sudo /opt/Citrix/VDA/bin/xdping -d myddc.domain.net
```

サンプル出力

以下は、Kerberosテストを実行した際のサンプル出力です。

Kerberosテストのサンプル出力の最初の部分(/en-us/linux-virtual-delivery-agent/2301/media/kerberos-test-sample-output-1.png)

Kerberosテストのサンプル出力の2番目の部分(/en-us/linux-virtual-delivery-agent/2301/media/kerberos-test-sample-output-2.png)

Kerberosテストのサンプル出力の3番目の部分(/en-us/linux-virtual-delivery-agent/2301/media/kerberos-test-sample-output-3.png)

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 の 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™ アプリ (HTML5向け) のサポート

Citrix Workspace app for HTML5を使用すると、クライアントをCitrix Gatewayに接続せずに、Linux仮想アプリおよびデスクトップに直接アクセスできます。Citrix Workspace app for HTML5について詳しくは、Citrixドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトで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。

      その他のWebSocketポリシーも設定できます。WebSocketポリシーの完全なリストについては、「WebSocketsポリシー設定」を参照してください。

   b. 設定を有効にするには、VDAで`ctxvda`サービスと`ctxhdx`サービスをこの順序で再起動します。

   c. VDAで、次のコマンドを実行してWebSocketリスナーが実行されているかどうかを確認します。

      ```
      netstat -an | grep 8008
      ```

      WebSocketリスナーが実行されている場合、コマンド出力は次のようになります。

      ```
      tcp 0 0 :::8008 :::* LISTEN
      ```

注： TLS暗号化を有効にして、WebSocket接続を保護することもできます。TLS暗号化の有効化について詳しくは、「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` または `/opt/Citrix/VDA/sbin/enable_ldaps.sh` コマンドを実行すると、Python3 仮想環境を作成できます。ただし、コマンドで Python3 仮想環境を作成できない場合は、ネットワーク接続がなくても手動で作成できます。この記事では、ネットワーク接続なしで Python3 仮想環境を作成するための前提条件と手順について詳しく説明します。

前提条件

- `/opt/Citrix/VDA/sbin/ctxpython3` ディレクトリにアクセスするには、管理者権限が必要です。
- `Python3` パッケージの wheel ファイルが配置されていること。wheel ファイルは `<https://pypi.org/>` からダウンロードできます。

Python3 仮想環境を作成する

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
```

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 環境をご利用の場合:

```
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. LDAPS の依存関係をインストールします。

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip==21.3.1



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1 gssapi==1.7.3
ldap3==2.9.1 pyasn1==0.4.8 pycparser==2.21 six==1.16.0
```

4. **XDPing** の依存関係をインストールします。

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
upgrade pip==21.3.1



sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
asn1crypto==1.5.1 cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1
gssapi==1.7.3 ldap3==2.9.1 netifaces==0.11.0 packaging==21.3
pg8000==1.26.0 psutil==5.9.0 pyasn1==0.4.8 pycparser==2.21
pyparsing==3.0.8 scramp==1.4.1 six==1.16.0 termcolor==1.1.0
```

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /opt/
Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

NISとActive Directoryの統合

この記事では、SSSDを使用してLinux VDAでNISとWindows Active Directory (AD) を統合する方法について説明します。Linux VDAはCitrix Virtual Apps and Desktopsのコンポーネントと見なされます。その結果、Windows AD環境にしっかりと適合します。

UIDおよびGIDプロバイダーとしてADの代わりにNISを使用する場合、アカウント情報（ユーザー名とパスワードの組み合わせ）がADとNISで同じである必要があります。

注:

認証は引き続きADサーバーによって実行されます。NIS+はサポートされていません。NISをUIDおよびGIDプロバイダーとして使用する場合、WindowsサーバーからのPOSIX属性は使用されなくなります。

ヒント:

この方法は、Linux VDAを展開する非推奨の方法であり、特別なユースケースでのみ使用されます。RHEL/CentOSディストリビューションの場合は、Install the Linux VDA on Amazon Linux 2, CentOS, RHEL, and Rocky Linux manuallyの指示に従ってください。Ubuntuディストリビューションの場合は、Install the Linux VDA on Ubuntu manuallyの指示に従ってください。

SSSDとは？

SSSDはシステムデーモンです。その主な機能は、システムにキャッシュとオフラインサポートを提供できる共通のフレームワークを通じて、リモートリソースを識別および認証するためのアクセスを提供することです。PAMモジュールとNSSモジュールの両方を提供し、将来的には拡張ユーザー情報のためにD-BUSベースのインターフェースをサポートできます。また、ローカルユーザーアカウントと拡張ユーザーデータを保存するためのより優れたデータベースも提供します。

NISとADの統合

NISをADと統合するには、次の手順を完了します。

ステップ1：LINUX VDAをNISクライアントとして追加する

NISクライアントを構成します。

```
yum –y install ypbind rpcbind oddjob-mkhomedir
```

NISドメインを設定します。

```
ypdomainname nis.domain

echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

NISサーバーとクライアントのIPアドレスを **/etc/hosts** に追加します。

`{NIS server IP address}  server.nis.domain nis.domain`

`authconfig` でNISを構成します。

```
sudo authconfig --enablenis --nisdomain=nis.domain --
nisserver=server.nis.domain --enablemkhomedir --update
```

nis.domain はNISサーバーのドメイン名を表します。**server.nis.domain** はNISサーバーのホスト名であり、NISサーバーのIPアドレスでもかまいません。

NISサービスを構成します:

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS構成が正しいことを確認します:

```
ypwhich
```

アカウント情報がNISサーバーから利用可能であることを検証します:

```
getent passwd nisaccount
```

> **注:**
>
> 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: ドメインに参加し、SAMBAを使用してホストキータブを作成する

SSSDは、ドメインへの参加やシステムキータブファイルの管理のための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、前のコマンドに次の2つのオプションを追加します:

`--enablekrb5kdcdns --enablekrb5realmdns`

/etc/samba/smb.conf を開き、**authconfig** ツールによって生成されたセクションの後に、**\[Global\]** セクションの下に次のエントリを追加します:

`kerberos method = secrets and keytab`

`winbind offline logon = no`

Windowsドメインに参加するには、ドメインコントローラーに到達可能であり、コンピューターをドメインに追加する権限を持つADユーザーアカウントが必要です。

```
sudo net ads join REALM -U user
```

REALM は大文字のKerberosレルム名であり、**user** はコンピューターをドメインに追加する権限を持つドメインユーザーです。

ステップ3：SSSDをセットアップする

SSSDのセットアップは、次の手順で構成されます。

- Linuxクライアントマシンに **sssd-ad** および **sssd-proxy** パッケージをインストールします。
- さまざまなファイル（例：**sssd.conf**）に構成変更を加えます。
- **sssdサービス** を開始します。

/etc/sssd/sssd.conf
sssd.conf 構成の例（必要に応じてオプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = EXAMPLE

services = nss, pam



[domain/EXAMPLE]

# Uncomment if you need offline logins

# cache_credentials = true

re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?
P<domain>.+$))|(^(?P<name>[^@\\]+)$))

id_provider = proxy
```

```
proxy_lib_name = nis

auth_provider = ad

access_provider = ad



# Should be specified as the long version of the Active Directory domain.

ad_domain = EXAMPLE.COM



# Kerberos settings

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U



# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



# Comment out if the users have the shell and home dir set on the AD side

default_shell = /bin/bash

fallback_homedir = /home/%d/%u



# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
available

# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

ad.domain.com、**server.ad.example.com** を対応する値に置き換えます。詳細については、**sssd-ad**(5) - Linux man page を参照してください。

sssd.conf のファイルの所有権とアクセス許可を設定します。

```
chown root:root /etc/sssd/sssd.conf
```

```
chmod 0600 /etc/sssd/sssd.conf
```

```
restorecon /etc/sssd/sssd.conf
```

ステップ4：NSS/PAMを構成する

RHEL/CentOS:

authconfig を使用してSSSDを有効にします。ホームディレクトリの作成がSELinuxと互換性があることを確認するために **oddjob-mkhomedir** をインストールします。

```
authconfig --enablesssd --enablesssdauth --enablemkhomedir --update



sudo systemctl start sssd



sudo systemctl enable sssd
```

> **ヒント:**

> Linux VDA設定を構成する際、SSSDにはLinux VDAクライアント用の特別な設定がないことを考慮してください。**ctxsetup.sh** スクリプトの追加ソリューションについては、デフォルト値を使用してください。

ステップ 5: KERBEROS構成の確認

Linux VDAでKerberosが正しく構成されていることを確認するには、システム**keytab**ファイルが作成され、有効なキーが含まれていることを確認します。

```
sudo klist -ke
```

このコマンドは、プリンシパル名と暗号スイートのさまざまな組み合わせで利用可能なキーのリストを表示します。これらのキーを使用してマシンをドメインコントローラーで認証するには、Kerberosの**kinit**コマンドを実行します。

```
sudo kinit –k MACHINE\$@REALM
```

マシン名とレルム名は大文字で指定する必要があります。ドル記号 ($) は、シェルによる置換を防ぐために、バックスラッシュ (\\) でエスケープする必要があります。一部の環境では、DNSドメイン名がKerberosレルム名と異なる場合があります。レルム名が使用されていることを確認してください。このコマンドが成功した場合、出力は表示されません。

マシンアカウントのTGTチケットがキャッシュされていることを、以下を使用して確認します。

```
sudo klist -ke
```

ステップ 6: ユーザー認証の確認

getentコマンドを使用して、ログオン形式がサポートされているか、およびNSSが機能するかどうかを確認します。

```
sudo getent passwd DOMAIN\\username
```

DOMAINパラメーターは、短いバージョンのドメイン名を示します。別のログオン形式が必要な場合は、まず**getent**コマンドを使用して確認してください。

サポートされているログオン形式は次のとおりです。

- ダウンレベルログオン名: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOSサフィックス形式: `username@DOMAIN`

SSSD PAMモジュールが正しく構成されていることを確認するには、ドメインユーザーアカウントを使用してLinux VDAにログオンします。このドメインユーザーアカウントは、以前に使用されたことがないものです。

```
sudo ssh localhost –l DOMAIN\\username
```

```
id -u
```

コマンドによって返された**uid**に対応するKerberos資格情報キャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_{uid}
```

ユーザーのKerberos資格情報キャッシュ内のチケットが有効で期限切れになっていないことを確認します。

```
klist
```

IPv6

Linux VDAは、Citrix Virtual Apps and Desktopsとの整合性を保つためにIPv6をサポートしています。この機能を使用する際は、次の点を考慮してください。

- デュアルスタック環境では、IPv6が明示的に有効になっていない限り、IPv4が使用されます。
- IPv4環境でIPv6が有効になっている場合、Linux VDAは機能しません。

重要:

- ネットワーク環境全体がIPv6である必要があり、Linux VDAだけではありません。
- 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作業は必要ありません。

Linux VDAのIPv6を構成する

Linux VDAの構成を変更する前に、Linux仮想マシンが以前にIPv6ネットワークで動作していたことを確認してください。IPv6構成に関連するレジストリキーは2つあります。

```
“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“OnlyUseIPv6ControllerRegistration”
```

```
“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“ControllerRegistrationIPv6Netmask”
```

OnlyUseIPv6ControllerRegistration を1に設定して、Linux VDAでIPv6を有効にする必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

Linux VDAに複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask** を使用して、Linux VDAの登録に使用するインターフェイスを指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ControllerRegistrationIPv6Netmask " -d "{IPv6 netmask}" --force
```

{IPv6 netmask} を実際のネットマスク（例: 2000::/64）に置き換えます。

Citrix Virtual Apps and DesktopsでのIPv6展開の詳細については、IPv4/IPv6 supportを参照してください。

トラブルシューティング

基本的なIPv6ネットワーク環境を確認し、ping6を使用してADとDelivery Controller™に到達可能かどうかを確認します。

LDAPS

LDAPS は、LDAP 通信が TLS/SSL を使用して暗号化される、Lightweight Directory Access Protocol (LDAP) のセキュアバージョンです。

デフォルトでは、クライアントとサーバーアプリケーション間の LDAP 通信は暗号化されません。LDAPS を使用すると、Linux VDA と LDAP サーバー間の LDAP クエリコンテンツを保護できます。

以下の Linux VDA コンポーネントは LDAPS に依存しています。

- ブローカーエージェント: Delivery Controller™ への Linux VDA 登録
- ポリシーサービス: ポリシー評価

LDAPS の構成には以下が含まれます。

- Active Directory (AD)/LDAP サーバーで LDAPS を有効にする
- クライアントで使用するためにルート CA をエクスポートする
- Linux VDA で LDAPS を有効/無効にする
- サードパーティプラットフォーム向けに LDAPS を構成する
- SSSD を構成する
- Winbind を構成する
- Centrify を構成する
- Quest を構成する

注:

LDAP サーバーの監視サイクルを設定するには、次のコマンドを実行します。デフォルト値は 15 分です。少なくとも 10 分に設定してください。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v
"ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --
force
```

AD/LDAPサーバーでLDAPSを有効にする

Microsoft 認証局 (CA) または Microsoft 以外の CA 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の LDAPS 認証のために適切な証明書が自動的に提供されません。

多層認証局階層を使用してドメインコントローラーで LDAPS を有効にする方法については、LDAP over SSL (LDAPS) 証明書 の記事を参照してください。

クライアントの使用のためにルート認証局を有効にする

クライアントは、LDAP サーバーが信頼する CA からの証明書を使用する必要があります。クライアントの LDAPS 認証を有効にするには、ルート CA 証明書を信頼されたキーストアにインポートします。

ルート CA のエクスポート方法の詳細については、Microsoft サポート Web サイトのルート認証局証明書をエクスポートする方法 を参照してください。

Linux VDA で LDAPS を有効または無効にする

Linux VDA で LDAPS を有効または無効にするには、次のスクリプトを実行します (管理者としてログオンしている間):

このコマンドの構文には以下が含まれます。

- 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:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
```

- チャネルバインディングを使用して LDAP over SSL/TLS を有効にする:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enablecb pathToRootCA
```

> **注:**
>
> チャネルバインディング用のルート CA 証明書は PEM 形式である必要があります。
> LDAPS を有効にしても Python3 仮想環境が正常に作成されない場合は、Python3 仮想環境を作成する の手順に従って手動で作成してください。
>
> pipツールを使用する際に発生する可能性のあるSSL接続エラーに対処するには、次の信頼できるホストを/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
```

- SSL/TLSなしでLDAPにフォールバックする

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
```

LDAPS専用のJavaキーストアは **/etc/xdl/.keystore** にあります。影響を受けるレジストリキーは次のとおりです。

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers

HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy

HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS

HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore

HKLM\Software\Citrix\VirtualDesktopAgent\EnableChannelBinding
```

サードパーティプラットフォーム向けLDAPSの構成

Linux VDAコンポーネントの他に、VDAに準拠するSSSD、Winbind、Centrify、Questなどのいくつかのサードパーティソフトウェアコンポーネントもセキュア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署名と封印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を使用することを好むわけではありません。そしてほとんどの場合、LDAPS（ポート636上のSSL経由のLDAP）はポート389上のSTARTTLSと共存できません。

SSSD

オプションに従って、ポート636またはポート389でSSSDセキュアLDAPトラフィックを構成します。詳細については、SSSD LDAP Linux manページを参照してください。

ウィンバインド

Winbind LDAPクエリはADSメソッドを使用します。Winbindはポート389でのStartTLSメソッドのみをサポートします。影響を受ける構成ファイルは **/etc/samba/smb.conf** および **/etc/openldap/ldap.conf** (RHELの場合) または **/etc/ldap/ldap.conf** (Ubuntuの場合) です。ファイルを次のように変更します。

- smb.conf

  `ldap ssl = start tls`

  `ldap ssl ads = yes` `client ldap sasl wrapping = plain`

- ldap.conf

  `TLS_REQCERT never`

あるいは、SASL GSSAPI署名と封印によってセキュア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** の構成を変更します。

`ldap ssl = off`

`ldap ssl ads = no` `client ldap sasl wrapping = seal`

セントリファイ

Centrifyはポート636でのLDAPSをサポートしていません。ただし、ポート389でセキュアな暗号化を提供します。詳細については、Centrifyサイトを参照してください。

クエスト

Quest Authentication Serviceはポート636でのLDAPSをサポートしていませんが、別の方法を使用してポート389で安全な暗号化を提供します。

トラブルシューティング

この機能を使用すると、次の問題が発生する可能性があります。

- **LDAPSサービスの可用性**

  AD/LDAPサーバーでLDAPS接続が利用可能であることを確認します。ポートはデフォルトで636です。

- **LDAPSが有効な場合に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を介してLDAPトラフィックが暗号化されていない**

  デフォルトでは、LDAPSは無効になっています。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証明書のインポートに失敗しました**

  CA証明書のフルパスを指定し、ルートCA証明書が正しいタイプであることを確認してください。ほとんどのサポートされているJava Keytoolタイプと互換性があるはずです。サポートリストにない場合は、まずタイプを変換できます。証明書形式の問題が発生した場合は、base64エンコードされたPEM形式をお勧めします。

- **Keytool -list を使用してルートCA証明書の表示に失敗しました**

`/opt/Citrix/VDA/sbin/enable_ldaps.sh`を実行してLDAPSを有効にすると、証明書は/etc/xdl/.keystoreにインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

エックスオーソリティ

Linux VDA は、対話型リモート処理のために X11 ディスプレイ機能 (`xterm` および `gvim` を含む) を使用する環境をサポートします。この機能は、XClient と XServer 間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信の権限を保護するには、2つの方法があります。

- **Xhost**。デフォルトでは、Xhost は localhost の XClient のみが XServer と通信することを許可します。リモートの XClient が XServer にアクセスすることを許可する場合、特定のコンピューターで権限を付与するために Xhost コマンドを実行する必要があります。または、代わりに **xhost +** を使用して、任意の XClient が XServer に接続することを許可できます。
- **Xauthority**。`.Xauthority` ファイルは、各ユーザーのホームディレクトリにあります。これは、XServer の認証のために xauth によって使用される Cookie に資格情報を保存するために使用されます。XServer インスタンス (Xorg) が起動されると、Cookie はその特定のディスプレイへの接続を認証するために使用されます。

仕組み

Xorg が起動すると、`.Xauthority` ファイルが Xorg に渡されます。この `.Xauthority` ファイルには、次の要素が含まれています。

- ディスプレイ番号
- リモート要求プロトコル
- Cookie 番号

`xauth` コマンドを使用してこのファイルを参照できます。例:

```
# xauth -f ~/.Xauthority



# > list
```

```
# > us01msip06:107  MIT-MAGIC-COOKIE-1  fb228d1b695729242616c5908f11624b
```

XClient が Xorg にリモートで接続する場合、2つの前提条件を満たす必要があります。

- **DISPLAY** 環境変数をリモートの XServer に設定します。
- Xorg の Cookie 番号のいずれかを含む `.Xauthority` ファイルを取得します。

Xauthority の構成

リモート X11 ディスプレイのために Linux VDA で **Xauthority** を有効にするには、次の2つのレジストリキーを作成する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"XauthEnabled" -d "0x00000001" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"ListenTCP" -d "0x00000001" --force
```

Xauthority を有効にした後、`.Xauthority` ファイルを **XClient** に手動で、または共有ホームディレクトリをマウントして渡します。

- `.Xauthority` ファイルを XClient に手動で渡す

  ICA® セッションを起動すると、Linux VDA は XClient 用の `.Xauthority` ファイルを生成し、ログオンユーザーのホームディレクトリに保存します。この `.Xauthority` ファイルをリモートの 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` ファイルはログオンユーザーのホームディレクトリに作成されます。このホームディレクトリが XClient と共有されている場合、ユーザーはこの `.Xauthority` ファイルを XClient に手動で転送する必要はありません。**DISPLAY** および **XAUTHORITY** 環境変数が正しく設定されると、GUI は XServer デスクトップに自動的に表示されます。

トラブルシューティング

Xauthority が機能しない場合、以下のトラブルシューティング手順に従ってください。

1. root権限を持つ管理者として、すべての Xorg クッキーを取得します。

   ```
   ps aux | grep -i xorg
   ```

   このコマンドは、Xorg プロセスと、起動時に Xorg に渡されたパラメーターを表示します。別のパラメーターは、どの `.Xauthority` ファイルが使用されているかを表示します。例：

   ```
   /var/xdl/xauth/.Xauthority110
   ```

   Xauth コマンドを使用してクッキーを表示します。

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   ```

2. `Xauth` コマンドを使用して、`~/.Xauthority` に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーは Xorg と XClient の `.Xauthority` ファイルで同じである必要があります。
3. クッキーが同じである場合、Linux VDA の IP アドレスと公開デスクトップのディスプレイ番号を使用して、リモートディスプレイポートのアクセシビリティを確認します。

   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
- ダブルホップシングルサインオン認証
- フェデレーション認証サービス
- SSO 以外の認証
- スマートカード
- 匿名ユーザーによる非認証セッション

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 サービスと統合してユーザー認証を提供します。ユーザーは AAD 資格情報を使用して VM にログオンします。

AAD DS 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 22.04、20.04、18.04
- RHEL 8.7, 8.6, 8.4, 7.9
- SUSE 15.4

詳細については、このMicrosoftの記事を参照してください。

既知の問題と回避策

RHEL 7.9 では、AAD ユーザー認証後に PAM (Pluggable Authentication Module) `pam_loginuid.so` が `loginuid` の設定に失敗します。この問題により、AAD ユーザーは VDA セッションにアクセスできません。

この問題を回避するには、`/etc/pam.d/remote` で行 `Session required pam_loginuid.so` をコメントアウトします。例については、次のスクリーンショットを参照してください。

```
#%PAM-1.0
auth       substack     password-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
#session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      password-auth
session    include      postlogin
```

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、その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。「**Management**」タブで「**Login with Azure AD**」を選択してから「**Review + create**」をクリックしてください。

   AADでのログインを有効にする設定(/en-us/linux-virtual-delivery-agent/2301/media/aad-auth-create-vm.png)

2. テンプレートVMにAzure CLIをインストールします。 詳細については、このMicrosoft記事を参照してください。

ステップ2：テンプレートVMでマスターイメージを準備する

マスターイメージを準備するには、Machine Creation Services (MCS) を使用したLinux VDAの作成の「**ステップ3：マスターイメージを準備する**」に従ってください。

ステップ3：テンプレートVMを非ドメイン参加モードに設定する

マスターイメージを作成したら、以下の手順に従ってVMを非ドメイン参加モードに設定します。

1. コマンドプロンプトから次のスクリプトを実行します。

```
Modify /var/xdl/mcs/mcs_util.sh
```

2. `function read_non_domain_joined_info()`を見つけて、`NonDomainJoined`の値を`2`に変更します。例については、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

非ドメイン参加のテンプレートVMの準備ができたら、以下の手順に従ってVMを作成します。

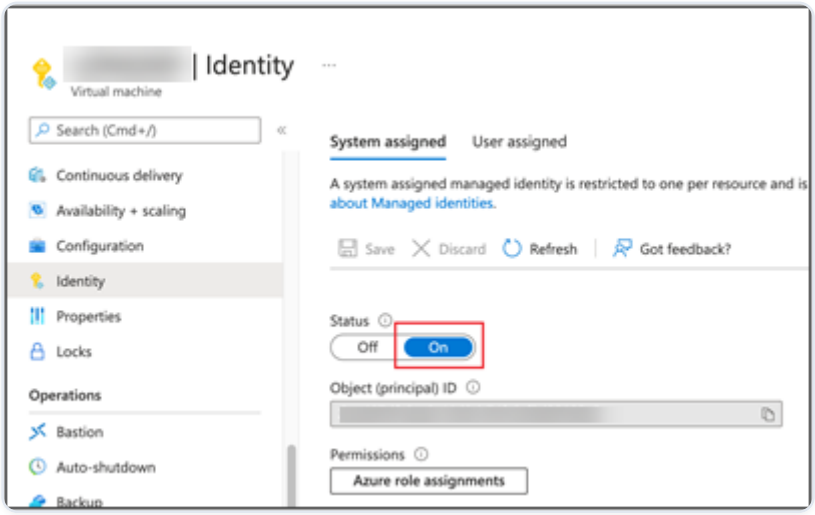
1. Citrix Cloudにサインインします。
2. Citrix DaaS™をダブルクリックし、Full Configuration管理コンソールにアクセスします。
3. 「**マシンカタログ**」で、テンプレートVMからLinux VMを作成するためにMachine Creation Servicesを使用することを選択します。詳細については、Citrix DaaSの記事「ドメインに参加していない」を参照してください。

ステップ5：Linux VMにAADユーザーアカウントを割り当てる

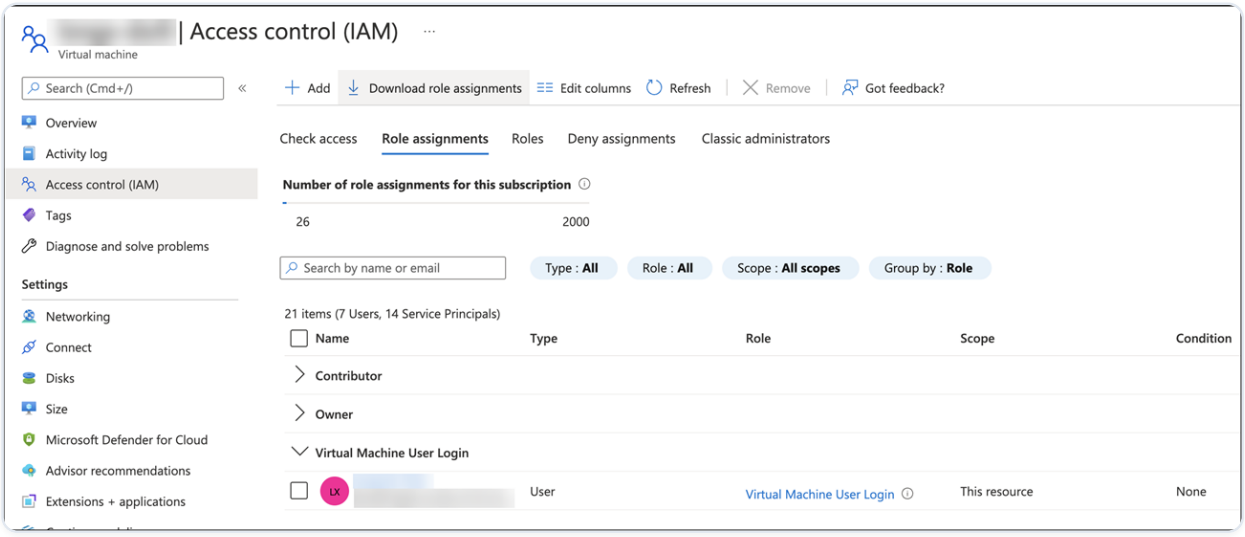
ドメインに参加していないVMを作成したら、それらにAADユーザーアカウントを割り当てます。

VMにAADユーザーアカウントを割り当てるには、次の手順に従います。

1. 管理者アカウントを使用してVMにアクセスします。
2. 「**ID > システム割り当て済み**」タブで、「**システムID**」を有効にします。



3. 「**アクセス制御 (IAM) > ロールの割り当て**」タブで、「**仮想マシンユーザーログイン**」領域を見つけ、必要に応じてAADユーザーアカウントを追加します。



ドメインに参加していないVDAにログオンする

組織のエンドユーザーは、2つの方法でドメインに参加していないVDAにログオンできます。詳細な手順は次のとおりです。

1. Workspaceアプリを起動し、AADユーザー名とパスワードを入力してワークスペースにログオンします。Workspaceページが表示されます。
2. ドメインに参加していないデスクトップをダブルクリックします。AADログインページが表示されます。

   ページは、VDAに設定されているログインモード（デバイスコードまたはAADアカウント/パスワード）によって異なります。デフォルトでは、Linux VDAは次のようにデバイスコードログインモードを使用してAADユーザーを認証します。管理者として、必要に応じてログインモードをAADアカウント/パスワードに変更できます。詳細な手順については、次のセクションを参照してください。

3. 画面の指示に従って、次のいずれかの方法でデスクトップセッションにログオンします。
   - QRコードをスキャンしてコードを入力します。
   - AADユーザー名とパスワードを入力します。

AADアカウント/パスワードログインモードへの変更

デフォルトでは、Linux VDAはデバイスコードを使用してAADユーザーを認証します。詳細については、このMicrosoftの記事を参照してください。ログインモードを*AADアカウント/パスワード*に変更するには、次の手順に従います。

VDAで次のコマンドを実行し、キー`AADAcctPwdAuthEnable`を見つけて、その値を`0x00000001`に変更します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Services\CitrixBrokerAgent\WebSocket" -t
"REG_DWORD" -v "AADAcctPwdAuthEnable" -d "0x00000001" --force
```

注:

このアプローチは、Microsoftアカウントや2要素認証が有効になっているアカウントでは機能しません。

ダブルホップシングルサインオン認証

StoreFrontストアにアクセスするためのユーザー資格情報は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10のAuthManagerモジュールに挿入できます。挿入後、クライアントを使用して、Linux仮想デスクトップセッション内から仮想デスクトップ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:

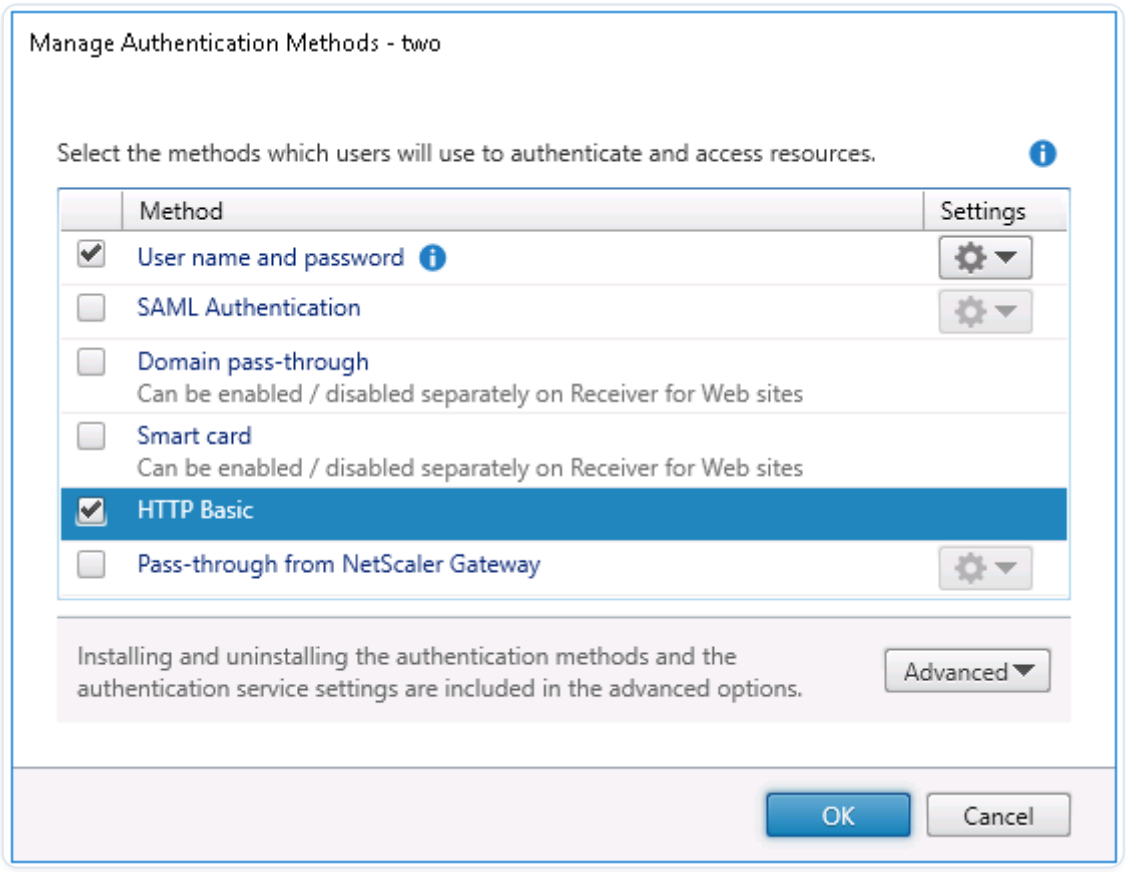
この機能は、Linux向けCitrix WorkspaceアプリおよびLinux向けCitrix Receiver 13.10でサポートされています。

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. 次のコマンドを実行して、指定されたディレクトリにルート証明書をインストールします。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. 次のコマンドを実行して、機能を有効にします。

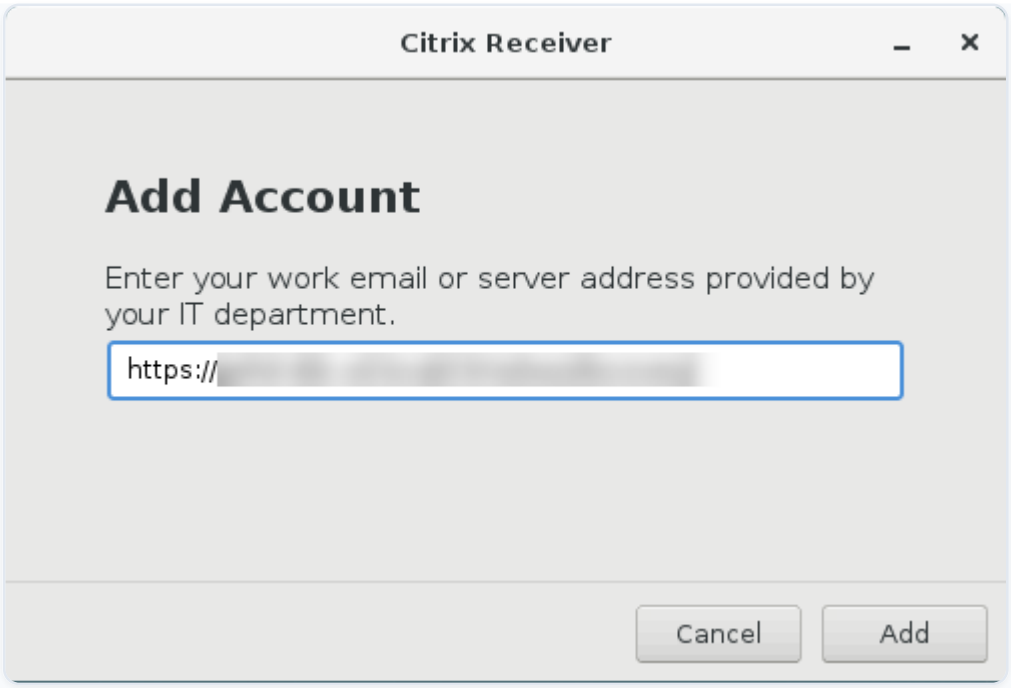
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

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きに、ストアアカウントの入力を求められます。その後、以前に指定したストアに自動的にログオンされます。

   注:

   ストアアカウントとしてHTTPS URLを入力します。



フェデレーション認証サービス

フェデレーション認証サービス (FAS) を使用して、Linux VDAにログオンするユーザーを認証できます。Linux VDAは、FASログオン機能に関してWindows VDAと同じWindows環境を使用します。FASのWindows環境の構成については、「フェデレーション認証サービス」を参照してください。この記事では、Linux VDAに特化した追加情報を提供します。

注:

- Linux VDAは、**セッション内動作**ポリシーをサポートしていません。
- Linux VDAは、FASサーバーとのデータ転送に短い接続を使用します。
- 2206リリース以降、ctxsetup.shのCTX_XDL_FAS_LISTを介して、Linux VDA側でFASポートをカスタマイズできます。詳細については、お使いのディストリビューションに基づいたLinux VDAインストール記事を参照してください。

Linux VDAでのFASの構成

RHEL 8.xおよびRocky Linux 8.xでのFASのサポート

FASは、RHEL 8.xおよびRocky Linux 8.xで非推奨となっているpam_krb5モジュールに依存します。マルチセッションOSモードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでFASを使用する場合は、以下の手順が必要です。シングルセッションOS（VDI）モードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでのFASについては、以下の手順をスキップできます。

1. 次のWebサイトからpam_krb5-2.4.8-6ソースコードをダウンロードします。

   `<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>`。

2. RHEL 8.xおよびRocky Linux 8.xにpam_krb5モジュールをビルドしてインストールします。

   ```
   yum install make gcc krb5-devel pam-devel autoconf libtool

   rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio -div

   tar xvzf pam_krb5-2.4.8.tar.gz

   cd pam_krb5-2.4.8

   ./configure --prefix=/usr

   make

   make install
   ```

3. /usr/lib64/security/の下にpam_krb5.soが存在することを確認します。

   ```
   ls -l /usr/lib64/security | grep pam_krb5
   ```

FASサーバーの設定

新規のLinux VDAインストールでFASを使用するには、ctxinstall.shまたはctxsetup.shを実行するときに、各FASサーバーのFQDNを入力します。Linux VDAはADグループポリシーをサポートしていないため、代わりにセミコロン区切りのFASサーバーリストを指定できます。サーバーアドレスが削除された場合は、その空白を**\<none\>**というテキスト文字列で埋め、サーバーアドレスの順序を変更しないでください。

既存のLinux VDAインストールをアップグレードするには、`ctxsetup.sh`を再実行してFASサーバーを設定できます。または、次のコマンドを実行してFASサーバーを設定し、`ctxvda`サービスを再起動して設定を有効にすることもできます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force



service ctxjproxy restart



service ctxvda restart
```

`ctxreg`を介してFASサーバーを更新するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -v "Addresses" -d "<Your-FAS-Server-List>"



service ctxjproxy restart



service ctxvda restart
```

証明書のインストール

ユーザーの証明書を検証するには、ルートCA証明書とすべての中間証明書をVDAにインストールします。たとえば、ルートCA証明書をインストールするには、前の「**Microsoft CAからCA証明書を取得する (AD上)**」の手順からADルート証明書を取得するか、ルートCAサーバーからそのDER形式をダウンロードします`http://CA-SERVER/certsrv`。

> **注:**
>
> 次のコマンドは、中間証明書の設定にも適用されます。

> 次のようなコマンドを実行して、DERファイル（.crt、.cer、.der）をPEMに変換します。

```
sudo openssl x509 -inform der -in root.cer -out root.pem
```

次に、次のようなコマンドを実行して、ルートCA証明書を`openssl`ディレクトリにインストールします。

```
sudo cp root.pem /etc/pki/CA/certs/
```

> **注:**
>
> ルートCA証明書を**/root**パスの下に置かないでください。そうしないと、FASはルートCA証明書に対する読み取り権限を持ちません。

ctxfascfg.shを実行する

FASを構成するには、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るように、環境変数が追加されます。

- **CTX_FAS_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis**: Active Directory統合方法を示します。これは、`CTX_EASYINSTALL_ADINTEGRATIONWAY`が指定されている場合、`CTX_EASYINSTALL_ADINTEGRATIONWAY`と等しくなります。

`CTX_EASYINSTALL_ADINTEGRATIONWAY`が指定されていない場合、`CTX_FAS_ADINTEGRATIONWAY`は独自の値を設定します。

- **CTX_FAS_CERT_PATH \=\<certificate path\>**: ルート証明書とすべての中間証明書が保存されている完全なパスを指定します。
- **CTX_FAS_KDC_HOSTNAME**: PBISを選択した場合のキー配布センター（KDC）のホスト名を指定します。
- **CTX_FAS_PKINIT_KDC_HOSTNAME**: PKINIT KDCホスト名を指定します。これは、特に指定がない限りCTX_FAS_KDC_HOSTNAMEと同じです。複数のデリバリーコントローラーがある場合は、ドメインのすべてのKDCのホスト名を**/etc/krb5.conf**ファイルのpkinit_kdc_hostnameに追加します。詳細については、Knowledge Centerの記事CTX322129を参照してください。

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。

その後、スクリプトはkrb5-pkinitおよびpam_krb5パッケージをインストールし、関連する構成ファイルを設定します。

FASを無効にする

Linux VDA で FAS を無効にするには、次のコマンドを使用して ConfDB からすべての FAS サーバーを削除します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -t "REG_SZ" -v "Addresses" -d "" --force



service ctxjproxy restart



service ctxvda restart
```

制限事項

- FAS は、限られた Linux プラットフォームと AD 統合方法をサポートしています。次のマトリックスを参照してください。

	ウィンバインド	SSSD	セントリファイ	PBIS
アマゾン Linux 2	はい	はい	はい	はい
デビアン 11.3	はい	はい	はい	はい
RHEL 9.0	はい	はい	いいえ	いいえ
RHEL 8.7/8.6/8.4	はい	はい	はい	はい
RHEL 7.9、CentOS 7.9	はい	はい	はい	はい
ロッキー Linux 9.0	はい	はい	いいえ	いいえ
ロッキー Linux 8.7/8.6	はい	はい	いいえ	いいえ
SUSE 15.4	はい	はい	はい	いいえ
ウブントゥ 22.04/20.04/18.04	はい	はい	はい	はい

- FAS はまだロック画面をサポートしていません。セッション内でロックボタンをクリックすると、FAS を使用してそのセッションに再度ログオンすることはできません。
- このリリースでは、(/ja-jp/federated-authentication-service/2212/deployment-architectures.html) 記事の「Federated Authentication Service のアーキテクチャの概要」にまとめられている一般的な FAS 展開のみがサポートされており、**Windows 10 Azure AD Join** は含まれていません。

トラブルシューティング

FAS のトラブルシューティングを行う前に、Linux VDA が正しくインストールおよび構成されていること、およびパスワード認証を使用して共通ストアで非 FAS セッションが正常に起動できることを確認してください。

非 FAS セッションが正常に動作する場合は、**Login** クラスの HDX ログレベルを VERBOSE に、VDA ログレベルを TRACE に設定します。Linux VDA のトレースログを有効にする方法については、Knowledge Center の記事 CTX220130 を参照してください。

FAS サーバー構成エラー

FAS ストアからのセッション起動に失敗します。

/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [Logon
Type] Federated Authentication Logon.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: entry



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
connect to server 0



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0: failed
to connect: Connection refused.



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: failed
to connect to server [0], please confirm if fas service list is well
configurated in condb



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit,
43



2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: failed
to validate fas credential
```

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate:
failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER
```

解決策

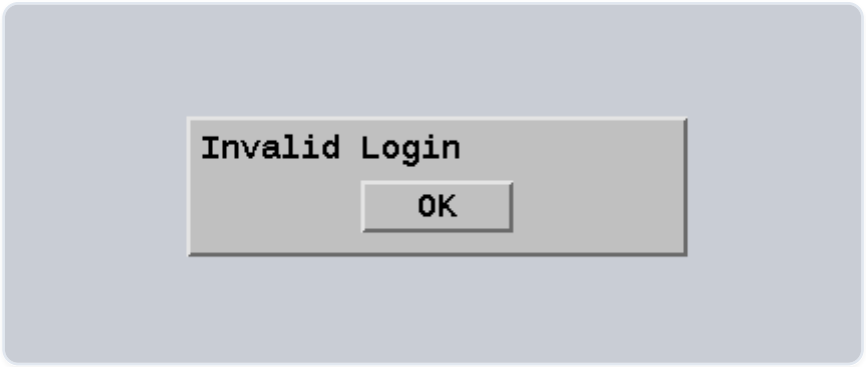
Citrix レジストリ値「HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\VirtualDesktopAgent\\Authentication\\UserCredentialService」が \<Your-FAS-Server-List\> に設定されていることを確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

既存の設定が正しくない場合は、前述の (/ja-jp/linux-virtual-delivery-agent/2301/configure/authentication/federated-authentication-service.html) 手順に従って再度設定してください。

CA 証明書構成が正しくありません

FAS ストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。



/var/log/xdl/hdx.log を確認し、次のようなエラーログを見つけてください。

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: entry



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller: current
process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]



2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
get_public_certificate: entry
```

```
2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
for response...

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query to
server success

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
get_public_certificate: exit

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
input size 1888

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
output size 1415

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: get logon certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
cache certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
get_logon_certificate: exit, 0
```

```
2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解決策

`/etc/krb5.conf` で、ルート CA 証明書とすべての中間証明書を格納する完全なパスが正しく設定されていることを確認してください。完全なパスは次のようになります。

```
 [realms]

EXAMPLE.COM = {

    ......

    pkinit_anchors = DIR:/etc/pki/CA/certs/

    ......

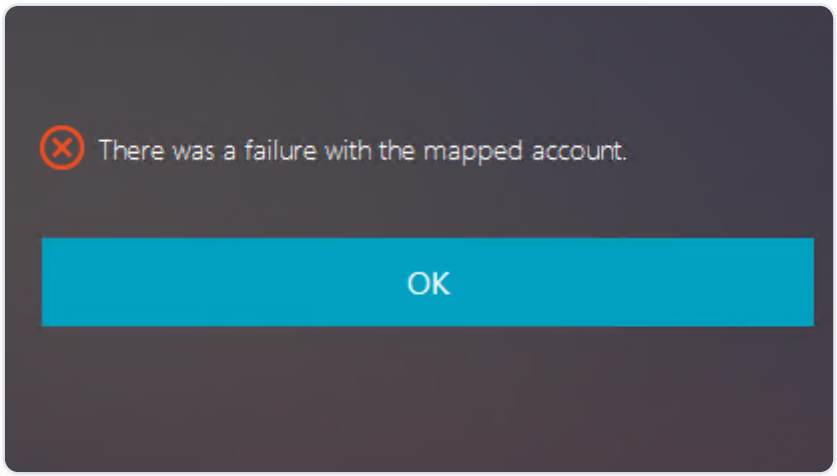
}
```

既存の設定が正しくない場合は、前述の (/ja-jp/linux-virtual-delivery-agent/2301/configure/authentication/federated-authentication-service.html#install-certificates) 手順に従って再度設定してください。

あるいは、ルートCA証明書が有効であるか確認してください。

シャドウアカウントマッピングエラー

FASはSAML認証によって構成されます。ADFSユーザーがADFSログオンページでユーザー名とパスワードを入力した後、次のエラーが発生する可能性があります。



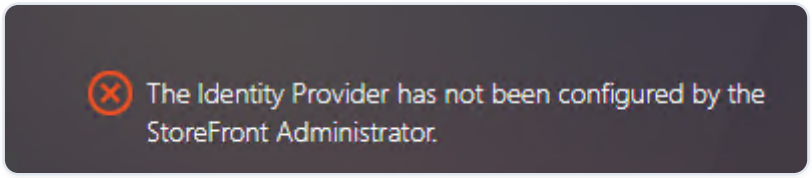
このエラーは、ADFSユーザーが正常に検証されたものの、AD上にシャドウユーザーが構成されていないことを示しています。

解決策

ADでシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に次のエラーが発生します。



この問題は、FASストアがSAML認証を使用するように構成されているにもかかわらず、ADFSの展開が不足している場合に発生します。

解決策

フェデレーション認証サービス用にADFS IdPを展開します。詳細については、「フェデレーション認証サービスADFSの展開」を参照してください。

関連情報

- 一般的なFAS展開については、「フェデレーション認証サービスアーキテクチャの概要」の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービス詳細構成」の章で紹介されています。

既知の問題

FASを使用している場合、非英語文字を含む公開デスクトップまたはアプリセッションを起動しようとすると失敗することがあります。

非英語文字を含むセッションの起動に失敗しました(/en-us/linux-virtual-delivery-agent/2301/media/launching-nonenglish-session-fails.png)

回避策

CAツールで「**テンプレートの管理**」を右クリックし、「**Citrix_SmartcardLogon**」テンプレートを「**このActive Directory情報から構築**」から「**要求で提供**」に変更します。

「要求で提供」オプション(/en-us/linux-virtual-delivery-agent/2301/media/supply-in-the-request.png)

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

SSOを無効にする

Linux VDAで次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\WinStations\tcp" -t
"REG_DWORD" -v "fPromptForDifferentUser" -d "0x00000001" --force
```

スマートカード

クライアントデバイスに接続されたスマートカードを使用して、Linux仮想デスクトップセッションにログオンする際の認証を行うことができます。この機能は、ICA®スマートカード仮想チャネルを介したスマートカードリダイレクトによって実装されます。セッション内でスマートカードを使用することもできます。使用例は次のとおりです。

- ドキュメントにデジタル署名を追加する
- メールを暗号化または復号化する
- Webサイトへの認証

この機能では、Linux VDAはWindows VDAと同じ構成を使用します。詳細については、この記事の「スマートカード環境の構成」セクションを参照してください。

> **注:**
>
> Linux VDAセッション内でマップされたスマートカードを使用してCitrix Gatewayにサインオンすることはサポートされていません。

前提条件

スマートカードパススルー認証の可用性は、以下の条件に依存します。

- お使いのLinux VDAは、以下のいずれかのディストリビューションにインストールされています。
  - RHEL 9.0
  - RHEL 8.7/8.6/8.4
  - RHEL 7、CentOS 7
  - ロッキーリナックス 9.0
  - ロッキーリナックス 8.7/8.6
  - ウブントゥ 22.04
  - ウブントゥ 20.04
  - ウブントゥ 18.04
  - デビアン 11.3

VDAのインストールが完了したら、VDAがDelivery Controller™に登録できること、およびWindows資格情報を使用して公開されたLinuxデスクトップセッションを開けることを確認します。

- OpenSCでサポートされているスマートカードが使用されます。詳細については、「OpenSCがスマートカードをサポートしていることを確認する」を参照してください。
- Citrix Workspace™ app for Windowsが使用されます。

OpenSCがスマートカードをサポートしていることを確認する

OpenSCは、RHEL 7.4以降で広く使用されているスマートカードドライバーです。CoolKeyと完全に互換性のある代替品として、OpenSCは多くの種類のスマートカードをサポートしています（「Red Hat Enterprise Linuxでのスマートカードのサポート」を参照）。

この記事では、構成を説明するための例としてYubiKeyスマートカードを使用します。YubiKeyは、Amazonやその他の小売業者から簡単に購入できるオールインワンのUSB CCID PIVデバイスです。OpenSCドライバーはYubiKeyをサポートしています。

組織でさらに高度なスマートカードが必要な場合は、サポートされているLinuxディストリビューションとOpenSCパッケージがインストールされた物理マシンを準備します。OpenSCのインストールについては、「スマートカードドライバーをインストールする」を参照してください。スマートカードを挿入し、次のコマンドを実行して、OpenSCがスマートカードをサポートしていることを確認します。

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. 通常、CAサーバーからPEM形式のルート証明書を取得します。

   次のコマンドと同様のコマンドを実行して、DERファイル（*.crt、*.cer、*.der）をPEMに変換できます。次のコマンド例では、**certnew.cer**はDERファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   ```

2. ルート証明書を`openssl`ディレクトリにインストールします。**certnew.pem**ファイルは例として使用されます。

```
cp certnew.pem <path where you install the root certificate>
```

ルート証明書をインストールするためのパスを作成するには、`sudo mdkir -p <path where you install the root certificate>`を実行します。

RHEL 8.xおよびRocky Linux 8.xでpam_krb5モジュールをビルドする

スマートカード認証はpam_krb5モジュールに依存しますが、これはRHEL 8.xおよびRocky Linux 8.xでは非推奨です。マルチセッションOSモードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでスマートカード認証を使用する場合は、以下の手順が必要です。シングルセッションOS (VDI) モードで提供されるRHEL 8.xおよびRocky Linux 8.xマシンでのスマートカード認証については、以下の手順をスキップできます。

1. `<https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html>`からpam_krb5-2.4.8-6ソースコードをダウンロードします。
2. RHEL 8.xおよびRocky Linux 8.xでpam_krb5モジュールをビルドしてインストールします。

```
yum install -y opensc pcsc-lite pcsc-lite-libs pcsc-lite-ccid nss-tools

yum install gcc krb5-devel pam-devel autoconf libtool

rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio –div

tar xvzf pam_krb5-2.4.8.tar.gz

cd pam_krb5-2.4.8

./configure --prefix=/usr

make

make install
```

3. /usr/lib64/security/にpam_krb5.soが存在することを確認します。

```
ls -l /usr/lib64/security | grep pam_krb5
```

スマートカード環境を構成する

ctxsmartlogon.shスクリプトを使用してスマートカード環境を構成するか、手動で構成を完了できます。

(オプション1) CTXSMARTLOGON.SHスクリプトを使用してスマートカード環境を構成する

> **注:**
>
> ctxsmartlogon.shスクリプトは、PKINIT情報をデフォルトレルムに追加します。この設定は、**/etc/krb5.conf**構成ファイルを通じて変更できます。
>
> スマートカードを初めて使用する前に、ctxsmartlogon.shスクリプトを実行してスマートカード環境を構成します。
>
> **ヒント:**
>
> ドメイン参加にSSSDを使用している場合は、ctxsmartlogon.shを実行した後、SSSDサービスを再起動してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

```
#*******************************************************************************
# ctxsmartlogon.sh 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.
```

ctxsmartlogon.shスクリプトを実行してスマートカードを無効にすることもできます。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行し、「いいえ」を選択する(/en-us/linux-virtual-delivery-agent/2301/media/smart-card-logon-script2.png)

（オプション2）スマートカード環境を手動で構成する

Linux VDAは、Windows VDAと同じスマートカード環境を使用します。この環境では、ドメインコントローラー、Microsoft証明機関（CA）、インターネットインフォメーションサービス、Citrix StoreFront、Citrix Workspaceアプリなど、複数のコンポーネントを構成する必要があります。YubiKeyスマートカードに基づいた構成については、Knowledge Centerの記事CTX206156を参照してください。

次の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。

対象: RHEL 9.0/8.x, Rocky Linux 9.0/8.x, RHEL 7/CentOS 7:

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

Ubuntu 22.04、Ubuntu 20.04、Ubuntu 18.04、Debian 11.3：

```
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 9.0/8.x、Rocky Linux 9.0/8.x、RHEL 7/CentOS 7：

```
yum install opensc
```

以下のオペレーティングシステム: Ubuntu 22.04, Ubuntu 20.04, Ubuntu 18.04, Debian 11.3：

```
apt-get install -y opensc
```

スマートカード認証用のPAMモジュールをインストールする
pam_krb5およびkrb5-pkinitモジュールをインストールするには、次のコマンドを実行します。

RHEL 7/CentOS 7:

```
yum install pam_krb5 krb5-pkinit
```

RHEL 9.0/8.x および Rocky Linux 9.0/8.x の場合:

```
yum install krb5-pkinit
```

Ubuntu 22.04、Ubuntu 20.04、および Ubuntu 18.04 の場合:

```
apt-get install libpam-krb5 krb5-pkinit
```

デビアン 11.3:

```
apt-get install -y libpam-krb5 krb5-pkinit
```

pam_krb5モジュールはプラガブル認証モジュールです。PAM対応アプリケーションは、pam_krb5を使用してパスワードをチェックし、Key Distribution Center (KDC) からチケット保証チケットを取得できます。krb5-pkinitモジュールには、クライアントが秘密鍵と証明書を使用してKDCから初期資格情報を取得できるようにするPKINITプラグインが含まれています。
pam_krb5モジュールを構成する
pam_krb5モジュールは、スマートカード内の証明書を使用してKerberosチケットを取得するためにKDCと対話します。PAMでpam_krb5認証を有効にするには、次のコマンドを実行します。

```
authconfig --enablekrb5 --update
```

/etc/krb5.conf構成ファイルで、実際のレルムに従ってPKINIT情報を追加します。

注:

pkinit_cert_matchオプションは、PKINIT認証を試行する前にクライアント証明書が一致する必要がある照合ルールを指定します。照合ルールの構文は次のとおりです。

コンポーネントルール ...

ここで、**relation-operator**は**&&（すべてのコンポーネントルールが一致する必要があることを意味します）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します）のいずれかです。

一般的なkrb5.confファイルの例を次に示します。

```
EXAMPLE.COM = {


    kdc = KDC.EXAMPLE.COM


    auth_to_local = RULE:[1:$1@$0]


    pkinit_anchors = FILE:<path where you install the root certificate>/
certnew.pem


    pkinit_kdc_hostname = KDC.EXAMPLE.COM


    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
```

```
    pkinit_eku_checking = kpServerAuth



}
```

PKINIT情報を追加すると、構成ファイルは次のようになります。

```
CTXDEV.LOCAL = {
    kdc = [redacted]
    auth_to_local = RULE:[1:$1@$0]
    pkinit_kdc_hostname = ctx-ad.ctxdev.local
    pkinit_anchors = FILE:/etc/pki/CA/certs/certnew.pem
    pkinit_eku_checking = kpServerAuth
    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
}
```

PAM認証を構成する

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

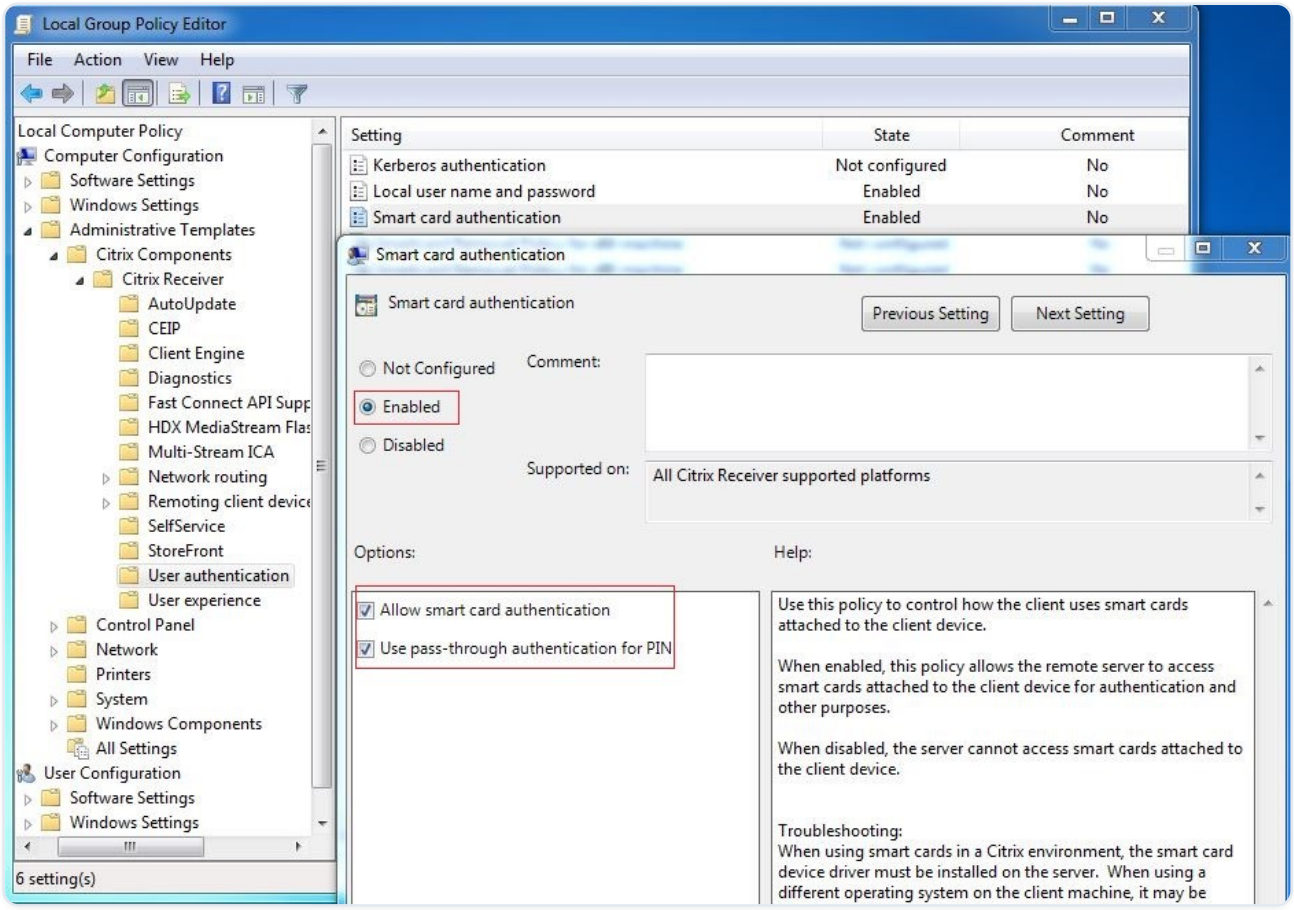
session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
#session     optional      pam_oddjob_mkhomedir.so umask=0077
session     optional      pam_mkhomedir.so umask=0077
session     [success=1 default=ignore] pam_succeed_if.so 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。詳しくは、Knowledge Centerの記事CTX133982を参照してください。

Citrix Workspaceアプリでグループポリシーを構成する際に、次のようにスマートカード認証を有効にします。



高速スマートカードログオン

高速スマートカードは、既存のHDX PC/SCベースのスマートカードリダイレクトを改善したものです。これにより、待機時間の長いWAN環境でスマートカードを使用する場合のパフォーマンスが向上します。詳しくは、スマートカードを参照してください。

Linux VDAは、Citrix Workspaceアプリの次のバージョンで高速スマートカードをサポートしています。

- シトリックス レシーバー (Windows版) 4.12
- Windows向けCitrix Workspaceアプリ1808以降

クライアントで高速スマートカードログオンを有効にする

高速スマートカードログオンは、VDAではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連するStoreFrontサイトのdefault.icaファイルに次のパラメーターを含めます。

```
[WFClient]

SmartCardCryptographicRedirection=On
```

クライアントで高速スマートカードログオンを無効にする

クライアントで高速スマートカードログオンを無効にするには、関連するStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

使用方法

スマートカードを使用してLinux VDAにログオンする

SSOシナリオと非SSOシナリオの両方で、スマートカードを使用してLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用して、StoreFront™に自動的にログオンします。StoreFrontでLinux仮想デスクトップセッションを起動すると、PINがスマートカード認証のためにLinux VDAに渡されます。
- 非SSOシナリオでは、証明書を選択し、PINを入力してStoreFrontにログオンするよう求められます。

  StoreFrontにログオンするためにPINを入力する(/en-us/linux-virtual-delivery-agent/2301/media/pin-new.png)

StoreFrontでLinux仮想デスクトップセッションを起動すると、Linux VDAへのログオンダイアログボックスが次のように表示されます。ユーザー名はスマートカードの証明書から抽出され、ログオン認証のためにPINを再度入力する必要があります。

この動作はWindows VDAと同じです。

ログオン認証(/en-us/linux-virtual-delivery-agent/2301/media/logon-authentication.png)

スマートカードを使用してセッションに再接続する

セッションに再接続するには、スマートカードがクライアントデバイスに接続されていることを確認してください。そうしないと、スマートカードが接続されていないため再認証が失敗し、Linux VDA側で灰色のキャッシュウィンドウが表示されてすぐに終了します。この場合、スマートカードを接続するよう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。

スマートカードを挿入する画像(/en-us/linux-virtual-delivery-agent/2301/media/please-insert-smart-card.png)

制限事項

制限されたLinuxディストリビューションとAD統合方法のサポート

- スマートカードパススルー認証は、制限されたLinuxディストリビューションとAD統合方法をサポートしています。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
デビアン 11.3	はい	はい	はい
RHEL 9.0	はい	はい	いいえ
RHEL 8.7/8.6/8.4	はい	はい	はい
RHEL 7.9、CentOS 7.9	はい	はい	はい
ロッキー リナックス 9.0	はい	はい	いいえ
ロッキー リナックス 8.7/8.6	はい	はい	いいえ
ウブントゥ 22.04/20.04/18.04	はい	はい	はい

スマートカード取り外しポリシー

現在、Linux VDAはスマートカードの取り外しに対してデフォルトの動作のみを使用します。Linux VDAに正常にログオンした後でスマートカードを取り外しても、セッションは接続されたままで、セッション画面はロックされません。

その他のスマートカードとPKCS#11ライブラリのサポート

Citrixは汎用スマートカードリダイレクトソリューションを提供しています。サポートリストにはOpenSCスマートカードのみが記載されていますが、他のスマートカードやPKCS#11ライブラリ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. `opensc-pkcs11.so` のすべてのインスタンスを、ご使用のPKCS#11ライブラリに置き換えてください。
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ライブラリのパスを指します。
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。

匿名ユーザーによる非認証セッション

この記事の情報を使用して、非認証セッションを構成します。この機能を使用するためにLinux VDAをインストールする際に、特別な設定は必要ありません。

> **注:**
>
> 非認証セッションを構成する際は、セッションの事前起動はサポートされていないことに注意してください。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。

非認証ストアを作成する

Linux VDAで非認証セッションをサポートするには、StoreFrontを使用して非認証ストアを作成します。

デリバリーグループで非認証ユーザーを有効にする

非認証ストアを作成したら、デリバリーグループで非認証ユーザーを有効にして、非認証セッションをサポートします。デリバリーグループで非認証ユーザーを有効にするには、Citrix Virtual Apps and Desktopsのドキュメントの手順に従ってください。

非認証セッションのアイドル時間を設定する

非認証セッションのデフォルトのアイドルタイムアウトは10分です。この値は、レジストリ設定 **AnonymousUserIdleTime** を介して構成されます。この値を変更するには、**ctxreg** ツールを使用します。たとえば、このレジストリ設定を5分に設定するには、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v AnonymousUserIdleTime
 -d 0x00000005
```

非認証ユーザーの最大数を設定する

非認証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非認証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、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v MaxAnonymousUserNumber
 -d  0x00000020
```

重要:

非認証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、利用可能なメモリの不足など、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

非認証セッションを構成する際は、次の点を考慮してください。

- **非認証セッションへのログオンに失敗しました。**

レジストリが次のように更新されていることを確認してください（0に設定）。

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsd サービスが実行中で、**passwd** キャッシュを有効にするように構成されていることを確認してください。

```
ps  uax | grep nscd

cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

passwd キャッシュ変数が有効になっている場合は、**no** に設定し、**ncsd** サービスを再起動します。この構成を変更した後、Linux VDA を再インストールする必要がある場合があります。

- **KDE を使用した認証されていないセッションでロック画面ボタンが表示されます。**

認証されていないセッションでは、ロック画面ボタンとメニューはデフォルトで無効になっています。ただし、KDE では引き続き表示される場合があります。KDE で、特定のユーザーのロック画面ボタンとメニューを無効にするには、構成ファイル **$Home/.kde/share/config/kdeglobals** に次の行を追加します。例：

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、`/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` ファイルを修正して `[KDE Action Restrictions]` セクションの `[$i]` タグを削除するか、またはシステム全体の構成を直接使用してロック画面ボタンとメニューを無効にします。KDE 構成の詳細については、KDE System Administration/Kiosk/Keys ページ を参照してください。

ファイル

このセクションには、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™ app 1903以降が必要です。

ファイルを正常にコピー＆ペーストするには、以下を確認してください。

- ファイルの最大数が20を超えないこと。
- ファイルの最大サイズが200 MBを超えないこと。

- Linux VDAをインストールしたマシンでNautilusファイルマネージャーが利用可能であること。

サポートされているLinuxディストリビューション

ファイルのコピー＆ペースト機能は、Linux VDAがサポートするすべてのLinuxディストリビューションで利用できます。

関連ポリシー

以下のクリップボードポリシーは、この機能の設定に関連しています。クリップボードポリシーの詳細については、ポリシーサポートリストを参照してください。

- クライアントクリップボードリダイレクト
- クリップボード選択更新モード
- プライマリ選択更新モード

> **注:**
>
> ファイルのコピー＆ペースト機能を無効にするには、Citrix Studioで**Client clipboard redirection**ポリシーを**Prohibited**に設定します。

制限事項

- 切り取りはサポートされていません。ファイルを切り取る要求はコピーとして扱われます。
- ドラッグ＆ドロップはサポートされていません。
- ディレクトリのコピーはサポートされていません。
- ファイルのコピーと貼り付けは、順次実行する必要があります。前のファイルが正常にコピーおよび貼り付けされた後にのみ、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により、ユーザーはCitrix Workspace™ app for HTML5 and 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 and for Chromeで利用できます。

公開されたアプリおよびデスクトップセッション内で、ファイル転送によりLinux VDAとクライアントデバイス間でのファイルのアップロードとダウンロードが可能になります。クライアントデバイスからLinux VDAにファイルをアップロードするには、Citrix Workspace appのツールバーにある**アップロード**アイコンをクリックし、ファイルダイアログから目的のファイルを選択します。Linux VDAからクライアントデバイスにファイルをダウンロードするには、**ダウンロード**アイコンをクリックします。アップロードまたはダウンロード中にファイルを追加できます。一度に最大100個のファイルを転送できます。

ツールバーのアップロードアイコンの画像(/en-us/linux-virtual-delivery-agent/2301/media/upload-icon-on-toolbar.png)

注:

Linux VDAとクライアントデバイス間でファイルをアップロードおよびダウンロードするには、Citrix Workspace app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 appを使用できます。

Autodownloadはファイル転送の機能強化です。VDA上の**Save to My Device**ディレクトリにダウンロードまたは移動したファイルは、自動的にクライアントデバイスに転送されます。

注:

Autodownloadには、**Allow file transfer between desktop and client**および**Download file from desktop**ポリシーが**Allowed**に設定されている必要があります。

自動ダウンロードのユースケースをいくつか示します。

- ファイルを**Save to My Device**にダウンロードする

  公開されたデスクトップおよびWebブラウザアプリセッション内で、Webサイトからダウンロードしたファイルは、VDA上の**Save to My Device**ディレクトリに保存され、クライアント

デバイスに自動的に転送されます。自動ダウンロードを実現するには、セッション内Webブラウザのデフォルトダウンロードディレクトリを**Save to My Device**に設定し、Citrix Workspace app for HTML5またはfor Chromeを実行するWebブラウザでローカルダウンロードディレクトリを設定します。

- ファイルを**Save to My Device**に移動またはコピーする

  公開されたデスクトップセッション内で、対象ファイルを選択し、クライアントデバイスで利用できるように**Save to My Device**ディレクトリに移動またはコピーします。

ファイル転送ポリシー

Citrix Studioを使用してファイル転送ポリシーを設定できます。デフォルトでは、ファイル転送は有効になっています。

ファイル転送ポリシーの画像(/en-us/linux-virtual-delivery-agent/2301/media/file-transfer-policies.png)

ポリシーの説明:

- **デスクトップとクライアント間のファイル転送を許可する。** ユーザーがCitrix Virtual Apps and Desktops™セッションと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。
- **デスクトップからファイルをダウンロードする。** ユーザーがCitrix Virtual Apps™ and Desktopsセッションからデバイスに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。
- **デスクトップにファイルをアップロードする。** ユーザーがデバイスからCitrix Virtual Apps and Desktopsセッションに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。

> **注:**
>
> 「**デスクトップからファイルをダウンロードする**」および「**デスクトップにファイルをアップロードする**」ポリシーを有効にするには、「**デスクトップとクライアント間のファイル転送を許可する**」ポリシーを「**許可**」に設定します。

使用方法

Citrix Workspaceアプリ for HTML5を介してファイル転送機能を使用するには：

1. Citrix Studioで、**WebSockets接続**ポリシーを**許可**に設定します。

WebSockets接続ポリシーの設定画像(/en-us/linux-virtual-delivery-agent/2301/media/websockets-connection.png)

2. Citrix Studioで、前述のファイル転送ポリシーを介してファイル転送を有効にします。
3. Citrix StoreFront管理コンソールで、**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。

   Always use Receiver for HTML5オプションを選択する画像(/en-us/linux-virtual-delivery-agent/2301/media/always-use-receiver-for-html5.png)
4. 仮想デスクトップまたはWebブラウザアプリセッションを起動します。Linux VDAとクライアントデバイス間で1つ以上のファイル転送を実行します。

Citrix Workspaceアプリ for Chromeを介してファイル転送機能を使用するには：

1. 前述のファイル転送ポリシーを介してファイル転送を有効にします。
2. ChromeウェブストアからCitrix Workspaceアプリを入手します。

   Citrix Workspaceアプリ for ChromeをChromeアプリページにすでに追加している場合は、この手順をスキップしてください。

   a. Google Chrome の検索ボックスに **Citrix Workspace for Chrome** と入力します。検索アイコンをクリックします。
   b. 検索結果の中から、Citrix Workspace アプリが利用可能な Chrome ウェブストアへの URL をクリックします。

      Citrix Workspaceアプリ for Chrome の画像(/en-us/linux-virtual-delivery-agent/2301/media/workspace-app-for-chrome.png)
   c. Citrix Workspace アプリを Google Chrome に追加するには、**Chrome に追加**をクリックします。
3. Chrome アプリページで Citrix Workspace app for Chrome をクリックします。
4. 接続する StoreFront™ ストアの URL を入力します。

   以前に URL を入力した場合は、この手順をスキップします。
5. 仮想デスクトップまたはアプリセッションを起動します。Linux VDA とクライアントデバイス間でファイルを1回以上転送します。

グラフィックス

このセクションには、次のトピックが含まれています。

- 自動DPIスケーリング
- クライアントのバッテリー状態表示
- グラフィックの構成と微調整
- HDX画面共有
- 非仮想化GPU
- セッションウォーターマーク
- Thinwireプログレッシブ表示

自動DPIスケーリング

Linux VDAは自動DPIスケーリングをサポートしています。ユーザーが仮想デスクトップまたはアプリケーションセッションを開くと、セッション内のDPI値はクライアント側のDPI設定に合わせて自動的に変更されます。

この機能に関する考慮事項は次のとおりです。

- この機能を使用するには、Citrix WorkspaceでDPIマッチングを有効にする必要があります。Citrix Workspaceアプリ for Windowsの場合、**いいえ、ネイティブ解像度を使用します**オプションが選択されていることを確認してください。Citrix Workspaceアプリ for WindowsのDPIスケーリングの構成について詳しくは、DPIスケーリングを参照してください。
- マルチモニター環境でこの機能を使用するには、各モニターが同じDPI設定で構成されている必要があります。混合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。
        - セッション実行中にクライアント側でDPI設定を変更した場合、ユーザーはログオフして再度ログオンする必要があります。
    - MATEデスクトップ環境で実行されているLinux仮想デスクトップの場合：
        - スケールファクター1および2のみがサポートされています。

    - セッション実行中にクライアント側でDPI設定を変更した場合、ユーザーはログオフして再度ログオンする必要があります。

- 仮想セッションのDPI値は、クライアント側のDPI設定に従って自動的に変更されます。現在、この機能は、100%や200%などの整数型のスケールファクターのみをサポートしています。クライアント側で構成されているスケールファクターが小数型の場合、仮想セッションのDPIは次の表に従って整数型のスケールファクターに変更されます。例：スケールファクターが125%の場合、DPI値は100%に変更されます。

クライアント側のスケールファクター	リモートセッションDPI
174%以下	96 (1 x 96)
175%～274%	192 (2 x 96)
275%～399%	288 (3 x 96)
400%以上	384 (4 x 96)

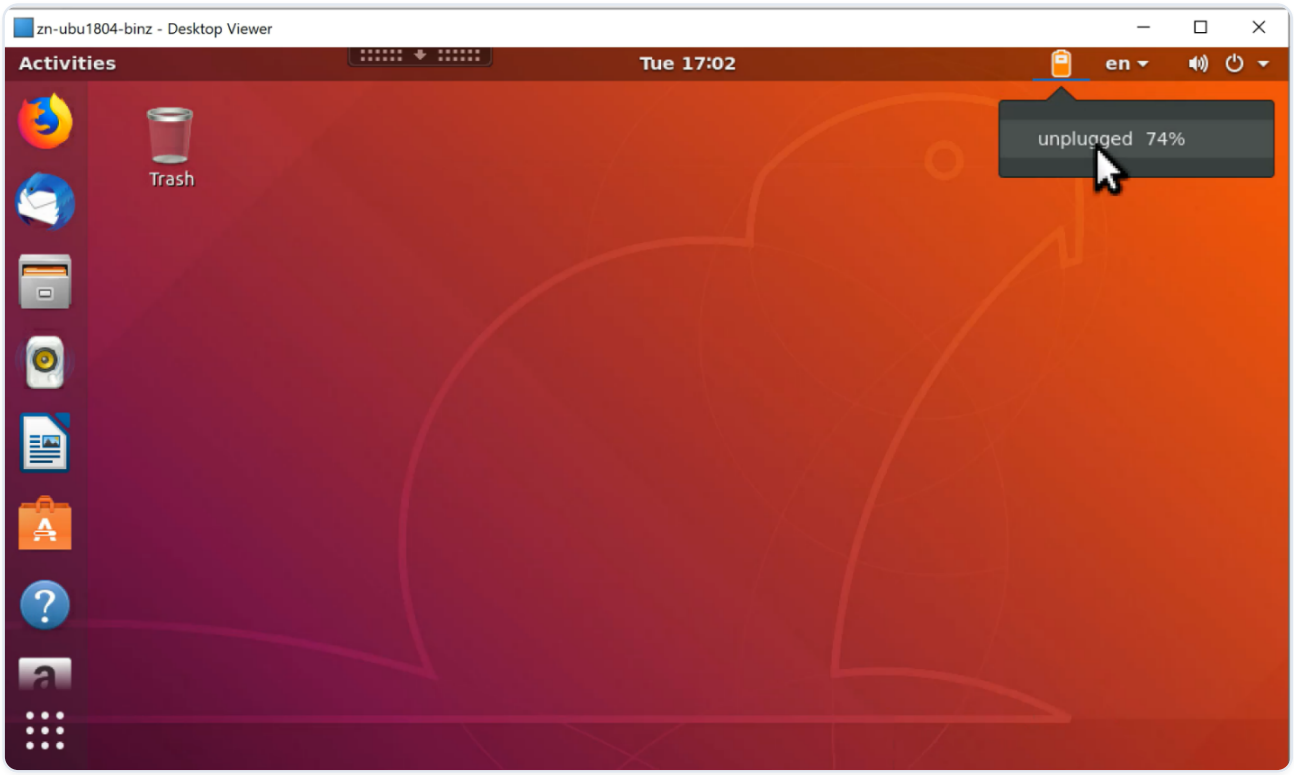
クライアントのバッテリー状態表示

Linux VDAは、仮想デスクトップでクライアントデバイスのバッテリー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™ appで利用できます。

- iOS向けCitrix Workspaceアプリ
- Linux向けCitrix Workspaceアプリ
- Mac向けCitrix Workspaceアプリ（バージョン2204.1はサポートされていません）
- Windows向けCitrix Workspaceアプリ（バージョン2204.1はサポートされていません）

概要

ユーザーが仮想デスクトップを開くと、Linuxシステムトレイにバッテリーアイコンが表示されます。バッテリーアイコンは、クライアントデバイスのバッテリー状態を示します。バッテリー残量の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。例として、次のスクリーンショットを参照してください。



異なるバッテリーアイコンは異なるバッテリー状態を示します。概要については、次の表を参照してください。

バッテリーアイコン	充電状態	バッテリー残量レベル	バッテリー残量の割合
	充電中（「+」記号で表示）	高（緑色で表示）	>=80%
充電中でバッテリー残量が中程度のバッテリー(/en-us/linux-virtual-delivery-agent/2301/media/battery-charging-medium.png)	充電中、「+」記号で示されます	中程度、アンバー色で示されます	20%以上80%未満
充電中でバッテリー残量が少ないバッテリー(/en-us/linux-virtual-delivery-agent/2301/media/battery-charging-low.png)	充電中、「+」記号で示されます	低い、赤色で示されます	< 20%

バッテリーアイコン	充電状態	バッテリー残量レベル	バッテリー残量の割合
充電中でなくバッテリー残量が多いバッテリー(/en-us/linux-virtual-delivery-agent/2301/media/battery-not-charging-high.png)	充電中ではない、「-」記号で示されます	高い、緑色で示されます	>=80%
充電中でなくバッテリー残量が中程度のバッテリー(/en-us/linux-virtual-delivery-agent/2301/media/battery-not-charging-medium.png)	充電中ではない、「-」記号で示されます	中程度、アンバー色で示されます	20%以上80%未満
	充電中ではないことを「-」記号で表示	低残量、赤色で表示	20%未満
	不明	不明	不明

構成

クライアントのバッテリー状態表示は、デフォルトで有効になっています。

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000001"
```

注:

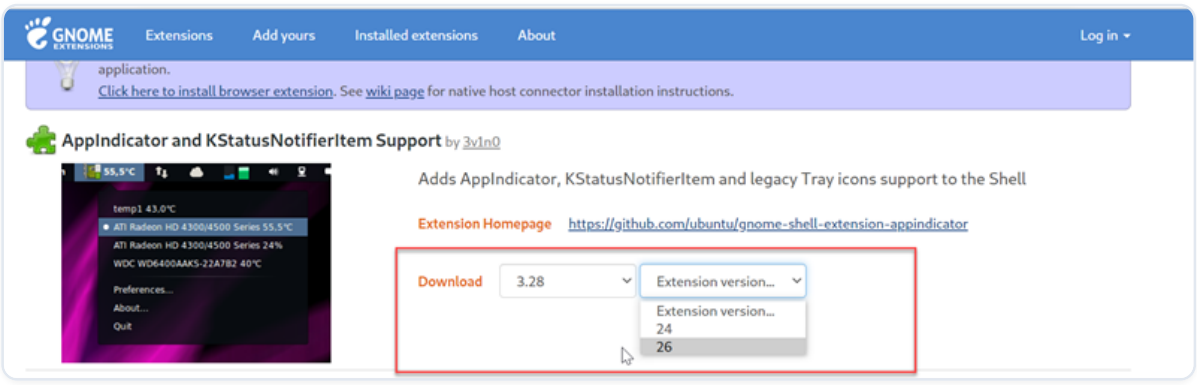
上記のコマンドは、クライアントのバッテリー状態表示とMobile Receiver Virtual Channel (MRVC) を共有するソフトキーボード機能に影響を与えます。

お使いのディストリビューションに基づいて、以下の追加手順を完了してください。

1. GNOMEがインストールされているRHEL 8.xまたはSUSE 15.xを使用している場合は、AppIndicatorのサポートを有効にするために、GNOMEシェル用の互換性のある拡張機能をインストールしてください。

   a. GNOMEシェルのバージョンを確認するには、`gnome-shell --version`コマンドを実行します。

   b. `<https://extensions.gnome.org/extension/615/appindicator-support>`からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能のバージョンとして24または26を選択できます。



   c. ダウンロードしたパッケージを解凍します。パッケージ内の **metadata.json** ファイルにある **"uuid"** の値が appindicatorsupport@rgcjonas.gmail.com に設定されていることを確認します。

   d. `/usr/share/gnome-shell/extensions/` の下の場所に appindicatorsupport@rgcjonas.gmail.com ディレクトリを移動するには、`mv` コマンドを実行します。

   e. **metadata.json** ファイルを他のユーザーが読み取り可能にするには、`chmod a+r metadata.json` コマンドを実行します。

   ヒント:

> デフォルトでは、appindicatorsupport@rgcjonas.gmail.com ディレクトリ内の **metadata.json** ファイルは、root ユーザーのみが読み取り可能です。画面共有をサポートするには、**metadata.json** ファイルを他のユーザーも読み取り可能にします。

f. GNOME Tweaks をインストールします。

g. デスクトップ環境で、`Alt+F2`、`r`、`Enter` キーを順番に押すか、`killall -SIGQUIT gnome-shell` コマンドを実行して GNOME shell をリロードします。

h. デスクトップ環境で GNOME Tweaks を実行し、Tweaks ツールで **KStatusNotifierItem/AppIndicator Support** を有効にします。

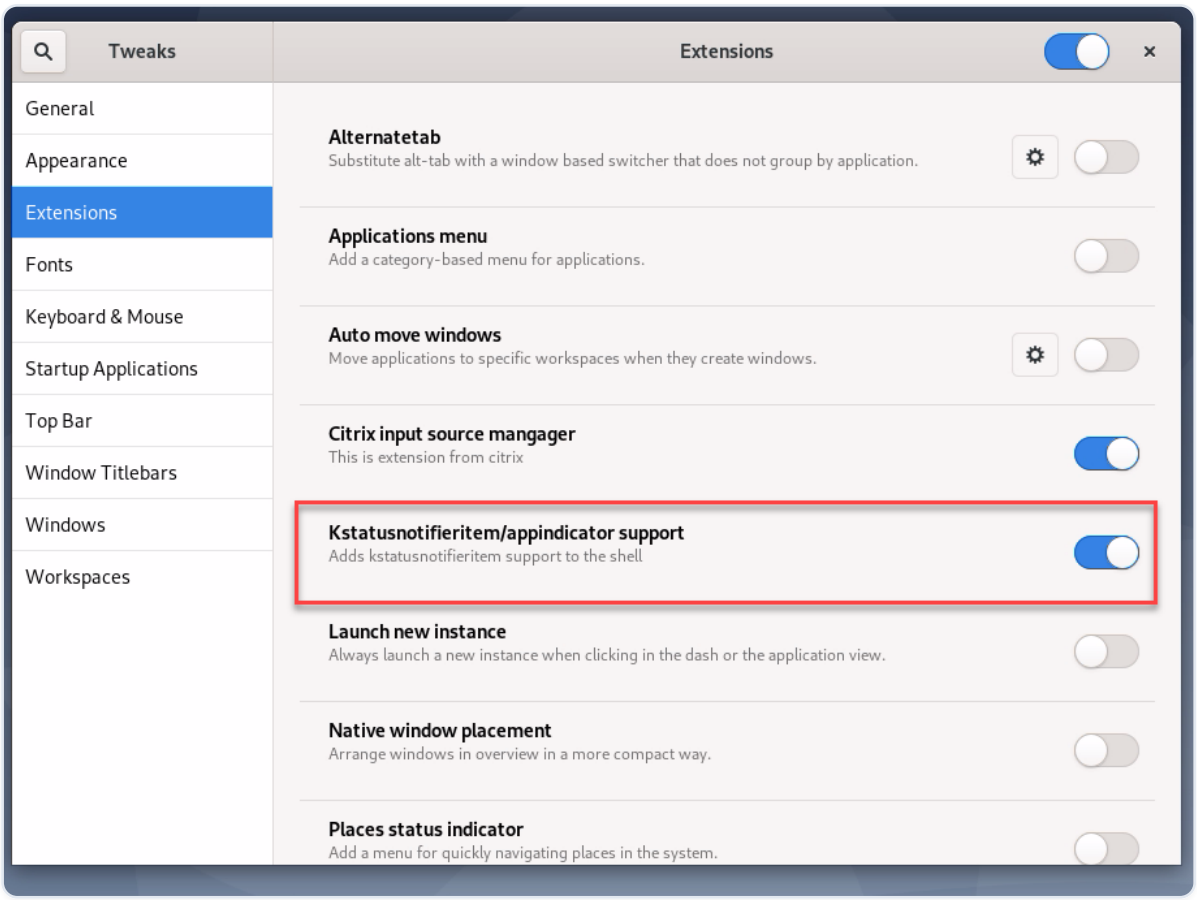
2. GNOME をインストールした Debian 11.3 を使用している場合は、GNOME システムトレイアイコンをインストールして有効にするために、次の手順を実行します。

a. `sudo apt install gnome-shell-extension-appindicator` コマンドを実行します。GNOME が拡張機能を認識するには、一度ログアウトしてから再度ログインする必要がある場合があります。

b. **Activities** 画面で Tweaks を検索します。

c. Tweaks ツールで **Extensions** を選択します。

d. **Kstatusnotifieritem/appindicator support** を有効にします。



グラフィック構成と微調整

この記事では、Linux VDAのグラフィック構成と微調整に関するガイダンスを提供します。

詳細については、「システム要件」および「インストール概要」セクションを参照してください。

構成

3Dグラフィックワークロード向けに最適化

この設定は、グラフィックを多用するワークロードに最適なデフォルト値を構成します。グラフィックを多用するアプリケーションに重点を置くユーザーに対して、この設定を有効にします。このポリシーは、セッションでGPUが利用可能な場合にのみ適用してください。このポリシーによって設定されたデフォルト設定を明示的に上書きする他の設定は、優先されます。

デフォルトでは、**3Dグラフィックワークロードの最適化**は無効になっています。

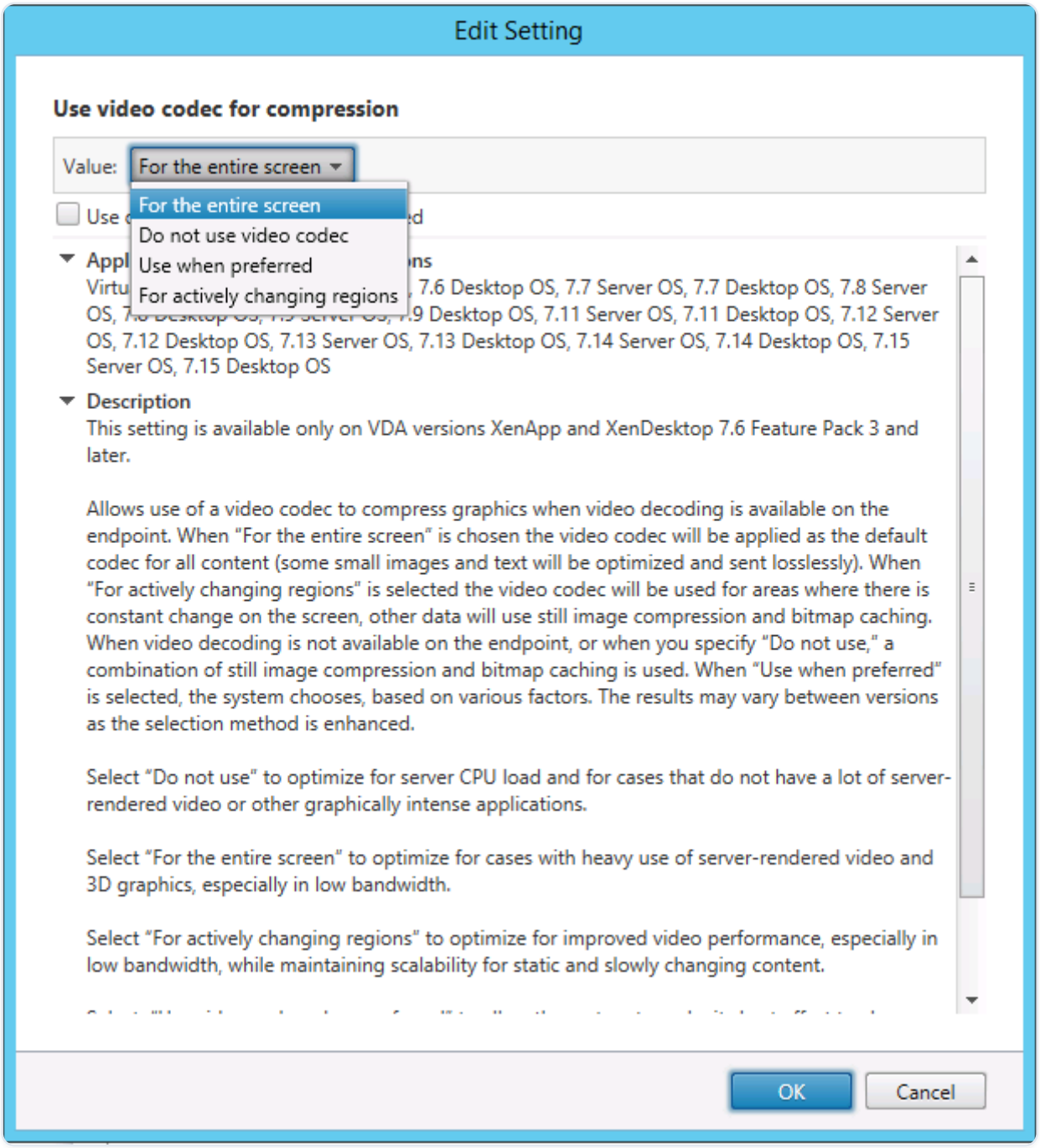
圧縮用ビデオコーデック

Thinwireは、Linux VDAで使用されるディスプレイリモート処理テクノロジーです。このテクノロジーにより、あるマシンで生成されたグラフィックを、通常はネットワーク経由で別のマシンに送信して表示できます。

「圧縮にビデオコーデックを使用」グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して次のオプションを提供します。

- **優先時に使用**。この設定はデフォルトです。追加の構成は不要です。これにより、すべてのCitrix®接続でThinwireが選択され、一般的なデスクトップワークロードのスケーラビリティ、帯域幅、および優れた画質のために最適化されます。
- **画面全体の場合**。ThinwireをフルスクリーンH.264またはH.265で配信し、特に3Dグラフィックを多用する場合に、ユーザーエクスペリエンスと帯域幅の向上を最適化します。「**画面全体の場合**」が選択されている場合、または「**優先時に使用**」が選択され、「3Dグラフィックワークロードの最適化」が有効になっている場合に、セッションウォーターマークがサポートされます。
- **アクティブに変更される領域の場合**。Thinwireのアダプティブディスプレイテクノロジーは、動く画像（ビデオ、移動中の3D）を識別します。画像が動いている画面の部分でのみH.264を使用します。H.264ビデオコーデックの選択的な使用により、HDX Thinwireは、H.264ビデオコーデックを使用して頻繁に更新される画面の一部を検出し、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、および他の場所でのロスレステキストまたは高品質画像の恩恵を受けます。この機能を有

効にするには、**圧縮にビデオコーデックを使用**ポリシーを**優先時に使用**（デフォルト）または**アクティブに変更される領域の場合**に設定します。詳細については、「グラフィックポリシー設定」を参照してください。



ディスプレイリモート処理のパフォーマンスを微調整するために、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**

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許可します。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。

GPUハードウェアアクセラレーションは、ハードウェアリソースの利用を最適化し、フレーム/秒 (FPS) のパフォーマンスを大幅に向上させます。

バージョン2210以降、GPUハードウェアアクセラレーションは以下のグラフィックモードをサポートします。

- 優先する場合に使用
- 画面全体向け
- 頻繁に変化する領域向け

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する」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。視覚的にロスレスな圧縮は、真のロスレス圧縮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**設定の値が使用される方法を変更します。

「視覚的にロスレスな圧縮を許可する」ポリシーはデフォルトで無効になっています。視覚的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する」**を**「有効」**に設定し、**「視覚品質」**ポリシーを**「ロスレスにビルド」**に設定します。

「圧縮にビデオコーデックを使用する」ポリシーが**「ビデオコーデックを使用しない」**に設定されている場合、視覚的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する」**ポリシーが**「ビデオコーデックを使用しない」**以外のグラフィックモードに設定されている場合、視覚的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。

以下のクライアントはSelective H.264をサポートしています。

- Windows 向け Citrix Receiver のバージョン 4.9 から 4.12 まで
- Citrix Receiver for Linux 13.5 から 13.10 まで
- Windows 向け Citrix Workspace™ app のバージョン 1808 以降
- シトリックス ワークスペース アプリ 1808 Linux 用 以降

視覚品質および**圧縮にビデオコーデックを使用する**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。

H.265ビデオコーデックのサポート

7.18リリース以降、Linux VDAはリモートグラフィックとビデオのハードウェアアクセラレーションのためにH.265ビデオコーデックをサポートしています。

この機能は以下で使用できます。

- シトリックス レシーバー for Windows 4.10〜4.12
- Windows向けCitrix Workspaceアプリ1808以降

この機能を利用するには、Linux VDAとクライアントの両方で有効にします。クライアントのGPUがDXVAインターフェイスを使用したH.265デコードをサポートしていない場合、**H.265 decoding for graphics**ポリシー設定は無視され、セッションはH.264ビデオコーデックの使用にフォールバックします。詳しくは、「H.265ビデオエンコーディング」を参照してください。

VDAでH.265ハードウェアエンコーディングを有効にするには：

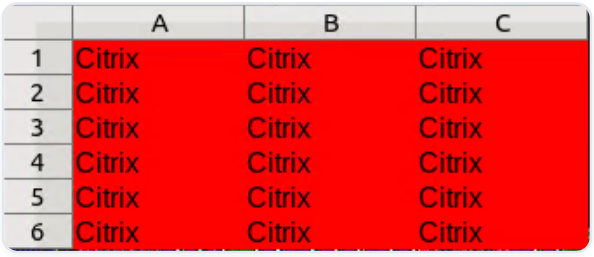
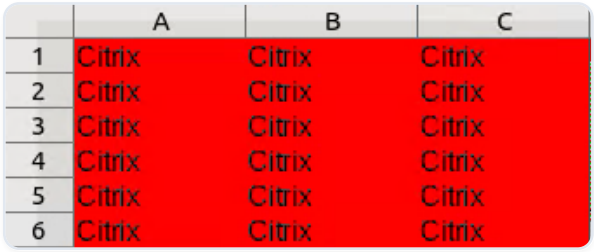
1. **Use hardware encoding for video codec**ポリシーを有効にします。
2. **Optimize for 3D graphics workload**ポリシーを有効にします。
3. **Use video codec for compression**ポリシーがデフォルトであるか、**For the entire screen**に設定されていることを確認します。
4. **Visual quality**ポリシーが**Build to Lossless**または**Always Lossless**に**設定されていない**ことを確認します。

クライアントでH.265ハードウェアエンコーディングを有効にするには、「H.265ビデオエンコーディング」を参照してください。

YUV444ソフトウェアエンコーディングのサポート

Linux VDAはYUV444ソフトウェアエンコーディングをサポートしています。YUVエンコーディングスキームは、各ピクセルに輝度値と色値の両方を割り当てます。YUVでは、**Y**は輝度またはルマ値を表し、**UV**は色またはクロマ値を表します。この機能は、Citrix Receiver for Windows 4.10〜4.12、およびCitrix Workspaceアプリ1808 for Windows以降で使用できます。

各Y、U、またはVの一意の値は、8ビット（1バイト）のデータで構成されます。YUV444データ形式は、1ピクセルあたり24ビットを送信します。YUV422データ形式は、2つのピクセル間でU値とV値を共有するため、1ピクセルあたりの平均伝送速度は16ビットになります。次の表は、YUV444とYUV420の直感的な比較を示しています。

YUV444	YUV420
	

VDAでYUV444ソフトウェアエンコーディングを有効にするには：

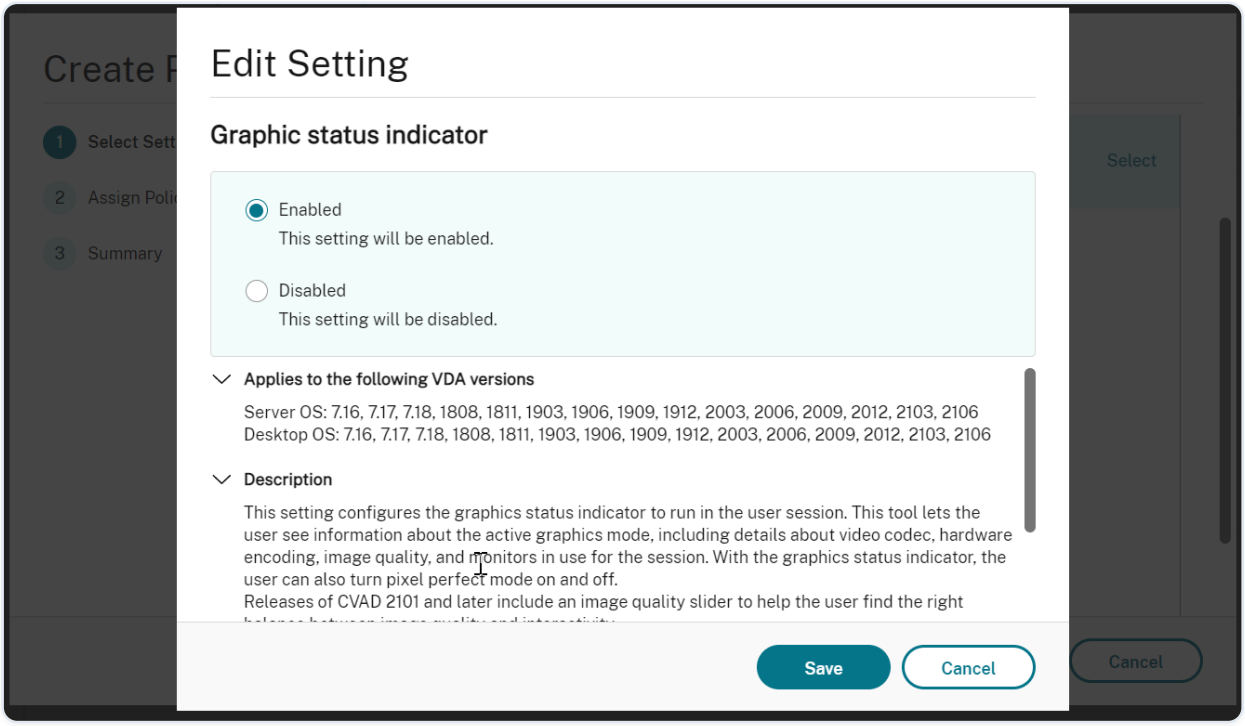
1. **圧縮にビデオコーデックを使用**ポリシーが**画面全体**に設定されていることを確認します。
2. **視覚的品質**ポリシーが**常にロスレス**または**ロスレスにビルド**に設定されていることを確認します。

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組み込みました。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。

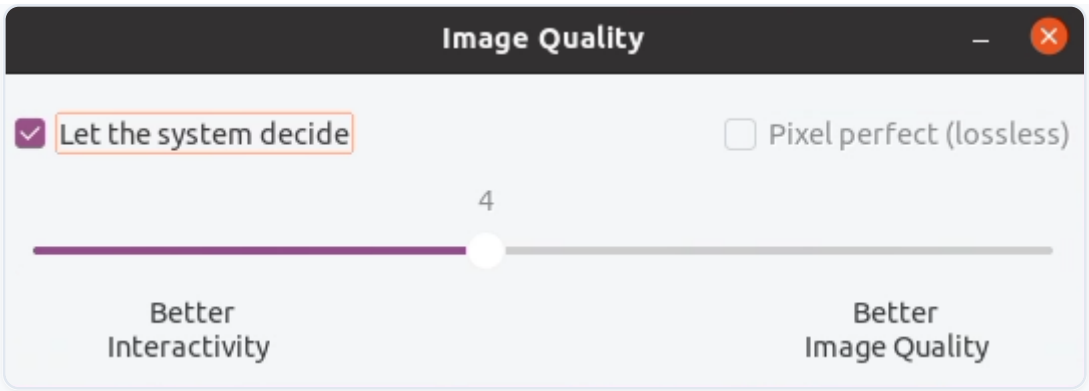
スライダーを使用するには、次の手順を実行します。

1. Citrix Studioで**グラフィックステータスインジケーター**ポリシーを有効にします。



2. ターミナルを開き、`ctxslider`コマンドを実行します。スライダーUIが表示されます。

> **注:**
>
> **視覚的品質**ポリシーを**常にロスレス**または**ロスレスにビルド**に設定している場合、スライダーUIは表示されません。



次の選択肢が利用可能です：

- 画像品質を変更するには、スライダーを動かします。スライダーは0～9の範囲をサポートしています。
- システム定義の設定を使用するには、**システムに決定させる**を選択します。
- ロスレスモードに切り替えるには、**ピクセルパーフェクト**を選択します。

帯域幅の見積もりに基づいて平均ビットレートを調整する

Citrixは、帯域幅の見積もりに基づいて平均ビットレートを調整することで、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、次のコマンドを実行してシャープネスと流暢さのバランスを調整することもできます。**AverageBitRatePercent**および**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。推奨設定範囲は50～100です。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "AverageBitRatePercent" -d "90" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "MaxBitRatePercent" -d "100" --force
```

平均ビットレート調整では、画面が静止している場合、新しいフレームが送信されないため、最新のフレームは低品質の状態のままになります。シャープ化サポートは、最新のフレームを最高品質で再構成してすぐに送信することで、この問題に対処できます。

Linux VDA Thinwireでサポートされているポリシーの完全なリストについては、「ポリシーサポートリスト」を参照してください。

Linux VDAでのマルチモニターサポートの構成については、「CTX220128」を参照してください。

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します。この機能はデフォルトで無効になっています。この機能を有効にするには、VDAで次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v
"ParallelProcessing" -d "0x00000001" --force
```

トラブルシューティング

使用中のグラフィックモードを確認する

使用中のグラフィックモードを確認するには、次のコマンドを実行します（**0**はTW+、**1**は全画面ビデオコーデックを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000000" --force`

H.264が使用されていることを確認する

H.264が使用されていることを確認するには、次のコマンドを実行します（**0**は使用されていないことを意味します。**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
```

たとえば、結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000000" --force`

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
```

たとえば、結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H265" -d "0x00000000" --force`

どのYUVエンコードスキームが使用されているかを確認する

どのYUVエンコード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
```

たとえば、結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000000" --force`

YUV444ソフトウェアエンコード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ド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 が使用中の場合、結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000001" --force`

NVIDIA GRID グラフィックドライバーが正しくインストールされていることを確認する

NVIDIA GRID グラフィックドライバーが正しくインストールされていることを確認するには、**nvidia-smi** を実行します。結果は次のようになります。

```
+-----------------------------------------------------+

| NVIDIA-SMI 352.70     Driver Version: 352.70         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
```

```
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
==============================+======================+==================
===|

|   0  Tesla M60           Off  | 0000:00:05.0     Off |
Off |

| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%
Default |

+-------------------------------+----------------------
+----------------------+




+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|

|  No running processes found
                                 |

+-----------------------------------------------------------------------------
----+
```

カードの正しい構成を設定します:

`etc/X11/ctx-nvidia.sh`

HDX 3D Pro マルチモニターの再描画に関する問題

プライマリモニター以外の画面で再描画の問題が発生している場合は、NVIDIA GRID ライセンスが利用可能であることを確認してください。

Xorg エラーログを確認する

Xorg のログファイルは、**/var/log/** フォルダーに **Xorg.{DISPLAY}.log** のように命名されています。

既知の問題と制限事項

vGPU の場合、Citrix Hypervisor™ のローカルコンソールに ICA デスクトップセッション画面が表示されます

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:

Citrix Hypervisor 8.1 以降の場合:

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

Citrix Hypervisor 8.1 より前のバージョンについて:

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
```

ログオン時の Gnome 3 デスクトップのポップアップの遅延

これは Gnome 3 デスクトップセッションの起動に関する制限事項です。

Citrix Workspace アプリのウィンドウのサイズを変更すると、一部の OpenGL/WebGL アプリケーションが適切にレンダリングされない

Citrix Workspace アプリのウィンドウのサイズを変更すると、画面解像度が変更されます。NVIDIA 独自のドライバーは一部の内部状態を変更し、アプリケーションがそれに応じて応答する必要がある場合があります。たとえば、WebGL ライブラリ要素の **lightgl.js** は、「**このテクスチャへのレンダリングはサポートされていません (不完全なフレームバッファ)**」というエラーを生成する可能性があります。

HDX™ 画面共有

概要

Linux VDA を使用すると、仮想デスクトップの画面を他の仮想デスクトップのセッションユーザーと共有できます。

次の例では、画面を共有し、他のユーザーの画面を表示する手順について説明します。

画面を共有するには:

1. 仮想デスクトップの通知領域で、**画面共有**アイコンをクリックし、**画面を共有**を選択します。

   画面共有メニューを選択します(/en-us/linux-virtual-delivery-agent/2301/media/select-the-share-my-screen-menu.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/2301/media/copy-the-connection-code.png)

   > **ヒント:**
   >
   > 画面を共有している間は、画面の周囲に赤い枠が表示され、共有が進行中であることを示します。
3. コピーしたコードを、画面を共有したい他の仮想デスクトップのセッションユーザーと共有します。
4. 閲覧者に画面を制御させるには、**制御を許可**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(/en-us/linux-virtual-delivery-agent/2301/media/give-control.png)
5. 画面共有を停止するには、**画面共有を停止**を選択します。

   画面共有停止メニューを選択します(/en-us/linux-virtual-delivery-agent/2301/media/select-the-stop-sharing-my-screen-menu.png)

他のユーザーの画面を表示するには：

1. 仮想デスクトップの通知領域で、**画面共有**アイコンをクリックし、**他のユーザーの画面を表示**を選択します。

   「他のユーザーの画面を表示」メニューを選択(/en-us/linux-virtual-delivery-agent/2301/media/select-the-view-someone-else-s-screen-menu.png)

2. 表示したい画面の接続コードを入力し、**接続**をクリックします。

   接続コードを入力(/en-us/linux-virtual-delivery-agent/2301/media/enter-the-connection-code.png)

3. 画面共有者が要求を承認するまで待ちます。例：

   画面共有を承認または拒否(/en-us/linux-virtual-delivery-agent/2301/media/accept-or-deny-screen-sharing.png)

   > **ヒント:**
   >
   > - 共有者側では、Linuxシステムが要求の通知を発行します。
   > - 共有者が30秒以内に要求を承認しない場合、要求は期限切れとなり、プロンプトが表示されます。

4. 画面共有者が**OK**をクリックして要求を承認すると、共有画面がDesktop Viewerに表示されます。自動的に割り当てられたユーザー名でビューアーとして接続されます。

   共有画面を表示(/en-us/linux-virtual-delivery-agent/2301/media/view-a-shared-screen.png)

5. 共有画面の制御を要求するには、左上隅にあるマウスアイコンをクリックします。

   > **ヒント:**
   >
   > - 共有者が30秒以内に要求を承認しない場合、要求は期限切れとなります。
   > - 共有画面を一度に制御できるビューアーは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。

   共有画面の制御を解除する(/en-us/linux-virtual-delivery-agent/2301/media/release-control-over-shared-screen.png)

6. ディスプレイのスケーリングを無効にするか、ウィンドウサイズに合わせてスケーリングするには、マウスアイコンの横にあるアイコンをクリックします。

   ウィンドウのスケーリング(/en-us/linux-virtual-delivery-agent/2301/media/window-scaling.png)

構成

画面共有機能はデフォルトで無効になっています。これを有効にするには、以下の設定を完了してください。

1. Citrix Studioでグラフィックステータスインジケーターポリシーを有効にします。
2. Citrix Virtual Apps and Desktops™ 2112以降の場合、Citrix Studioで**ScreenSharing**ポリシーを有効にします。
3. (オプション) Citrix Virtual Apps™ and Desktops 2109以前の場合、次のコマンドを実行してLinux VDAで画面共有を有効にします。

   ```
   sudo /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -v
   "EnableScreenSharing" -d "0x00000001"
   ```

4. ファイアウォールでポート52525～52625を許可します。
5. (オプション) RHEL 8.x、Debian 11、またはGNOMEがインストールされたSUSE 15.xを使用している場合は、AppIndicatorのサポートを有効にするために、GNOMEシェル用の互換性のある拡張機能をインストールします。

   a. `gnome-shell --version`コマンドを実行して、GNOMEシェルのバージョンを確認します。

   b. `<https://extensions.gnome.org/extension/615/appindicator-support>`からGNOMEシェル用の互換性のある拡張機能をダウンロードします。たとえば、シェルバージョンが3.28の場合、拡張機能のバージョンとして24または26を選択できます。

      GNOMEシェル拡張機能(/en-us/linux-virtual-delivery-agent/2301/media/gnome-extension.png)

   c. ダウンロードしたパッケージを解凍します。パッケージ内の**metadata.json**ファイルにある**"uuid"**の値が**appindicatorsupport@rgcjonas.gmail.com**に設定されていることを確認します。

   d. `mv`コマンドを実行して、appindicatorsupport@rgcjonas.gmail.comディレクトリを`/usr/share/gnome-shell/extensions/`の下の場所に移動します。

e. `chmod a+r metadata.json`コマンドを実行して、**metadata.json**ファイルを他のユーザーが読み取り可能にします。

> **ヒント:**
>
> デフォルトでは、appindicatorsupport@rgcjonas.gmail.com ディレクトリ内の **metadata.json** ファイルは、rootユーザーのみが読み取り可能です。画面共有をサポートするには、**metadata.json** ファイルを他のユーザーも読み取れるようにします。

f. GNOME Tweaksをインストールします。

g. デスクトップ環境で、`Alt+F2`、`r`、`Enter` キーを順番に押すか、`killall -SIGQUIT gnome-shell` コマンドを実行してGNOMEシェルをリロードします。

h. デスクトップ環境でGNOME Tweaksを実行し、Tweaksツールで **KStatusNotifierItem/AppIndicator Support** を有効にします。

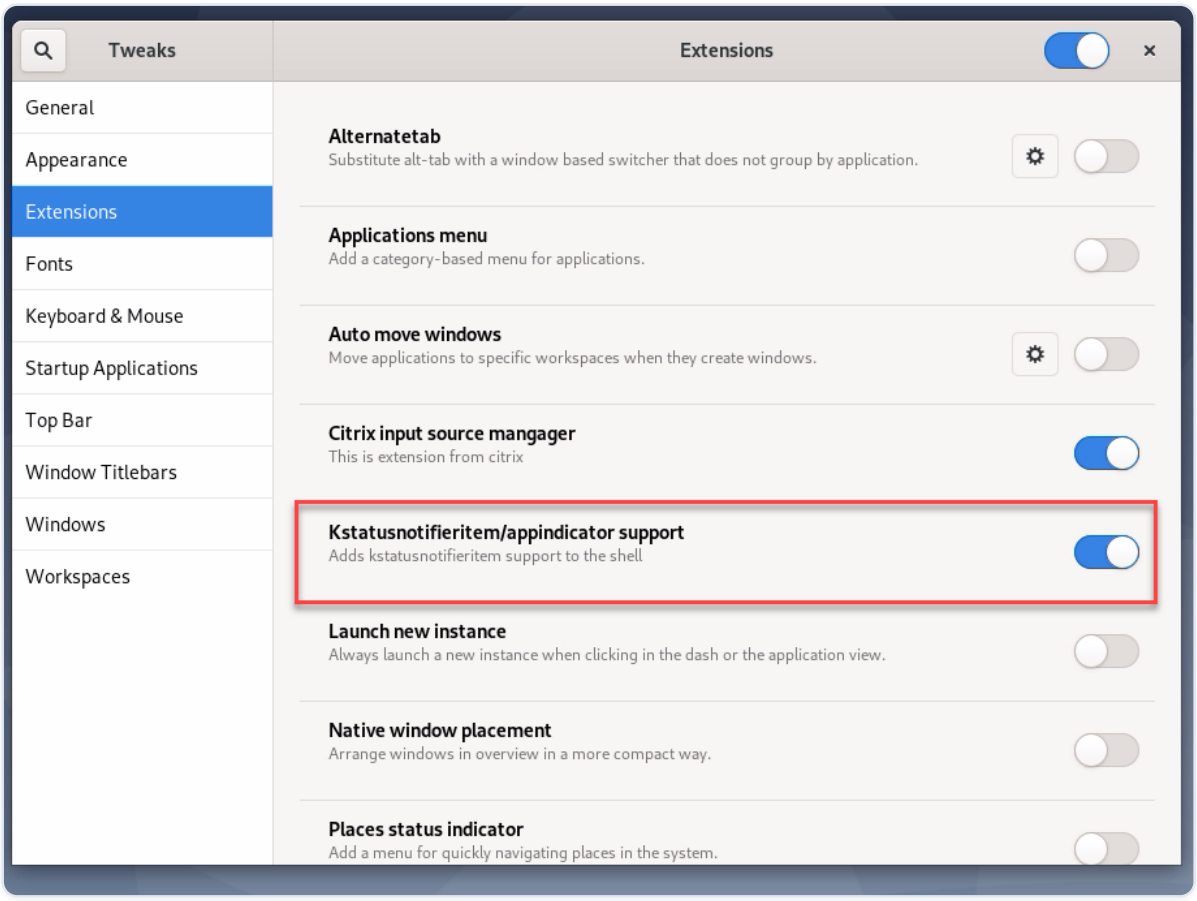
6. (オプション) GNOMEがインストールされたDebian 11.3を使用している場合は、GNOMEシステムトレイアイコンをインストールして有効にするために、次の手順を完了します。

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. **Activities** 画面でTweaksを検索します。

c. Tweaksツールで **Extensions** を選択します。

d. **Kstatusnotifieritem/appindicator support** を有効にします。



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。

非仮想化GPU

Linux VDAのドキュメントでは、非仮想化GPUは以下を指します。

- リモートPCアクセスシナリオで使用されるGPU
- ハイパーバイザーからパススルーされたGPU

この記事では、非仮想化GPUでHDX™ 3D Proを有効にする方法について説明します。

前提条件

- HDX 3D Proを有効にします。これを行うには、Linux VDAのインストール時に `CTX_XDL_HDX_3D_PRO` を `Y` に設定します。環境変数については、「ステップ8：インストールを完了するためのランタイム環境のセットアップ」を参照してください。
- NVIDIA Linux Capture SDKがサポートするGPUの場合、HDX 3D Proを有効にすると、ハードウェアアクセラレーションがデフォルトで有効になります。追加の設定は不要です。
- NVIDIA Linux Capture SDKがサポートしないGPUの場合、`XDamage` をインストールします。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。

構成

Xorg構成ファイルを変更する

NVIDIA非仮想化GPUの場合

構成ファイルは自動的にインストールおよび設定されます。

その他のGPUの場合

`/etc/X11/` にインストールされている4つのテンプレート構成ファイルを変更する必要があります。

- ctx-driver_name-1.conf
- ctx-driver_name-2.conf
- ctx-driver_name-3.conf
- ctx-driver_name-4.conf

ctx-driver_name-1.conf を例として、テンプレート構成ファイルを変更するには、次の手順を実行します。

1. ***driver_name*** を実際のドライバー名に置き換えます。

   たとえば、ドライバー名が `intel` の場合、構成ファイル名を `ctx-intel-1.conf` に変更できます。
2. ビデオドライバー情報を追加します。

   各テンプレート構成ファイルには、「Device」という名前のセクションが含まれており、コメントアウトされています。このセクションにはビデオドライバー情報が記述されています。ビ

デオドライバー情報を追加する前に、このセクションを有効にしてください。このセクションを有効にするには:

a. 構成情報については、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、ローカル環境でGPUが動作することを確認してください。

b. ネイティブ構成ファイルの「Device」セクションを **ctx-driver_name-1.conf** にコピーします。

3. 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
```

XDamage を有効にする

NVIDIA Linux Capture SDK のリリースノートのサポートされているハードウェアセクションに記載されていない GPU を使用している場合は、次を使用して `XDamage` を有効にします。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

リモート PC アクセス VDA のモニターブランキング

Linux VDA は、非仮想化 GPU を使用するリモート PC アクセス VDA の物理モニターブランキングをサポートしています。

この機能をサポートする完全にテスト済みの Linux ディストリビューションには、Ubuntu 20.04 および Debian 11.3 が含まれます。

この機能はデフォルトで無効になっています。有効にするには、次の2つの手順を完了してください。

1. お使いのLinuxディストリビューションに基づいて、`evdi-dkms` パッケージをインストールします:

```
sudo apt install evdi-dkms
```

2. EVDIへのグラフィックディスプレイのオフロードを有効にします:

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD"
-v "Evdi" -d "0x00000001" --force
```

3. Intel GPUを使用している場合は、ディスプレイマネージャーを無効にしてください。そうしないと、Intel GPUがディスプレイマネージャーによって占有され、Citrix®リモートセッションで利用できなくなります。

```
sudo systemctl disable --now gdm
```

トラブルシューティング

グラフィック出力がない、または文字化けしている

3Dアプリケーションをローカルで実行でき、すべての構成が正しい場合、グラフィック出力がない、または文字化けしているのはバグが原因です。デバッグ用のトレース情報を収集するには、/opt/Citrix/VDA/bin/setlogを使用し、GFX_X11をverboseに設定してください。

ハードウェアエンコードが機能しない

`Xdamage` を使用する場合、ソフトウェアエンコードのみがサポートされます。

セッションウォーターマーク

セッションウォーターマークは、データ窃盗を抑止し、追跡を可能にするのに役立ちます。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、テキストのレイヤーまたはアルファチャネル付きのPNG画像として指定できます。ウォーターマークは、元のドキュメントの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。この機能を使用しても、完全な情報追跡可能性を保証するものではありません。代わりに、この機能を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、スクリーン画像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ために、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てください。

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、セッションディスプレイにはセッション固有の情報を表示する不透明なウォーターマークが表示されます。他のウォーターマーク設定は、この設定が有効になっているかどうかに依存します。

デフォルトでは、セッションウォーターマークは無効になっています。

クライアントIPアドレスを含める

この設定を有効にすると、セッションは現在のクライアントIPアドレスをウォーターマークとして表示します。

デフォルトでは、**クライアントIPアドレスを含める**は無効になっています。

接続時間を含める

この設定を有効にすると、セッションウォーターマークに接続時間が表示されます。形式はyyyy/mm/dd hh:mmです。表示される時刻は、システムクロックとタイムゾーンに基づいています。

デフォルトでは、**接続時間を含める**は無効になっています。

ログオンユーザー名を含める

この設定を有効にすると、セッションは現在のログオンユーザー名をウォーターマークとして表示します。表示形式はUSERNAME@DOMAINNAMEです。ユーザー名は最大20文字にすることをお勧めします。ユーザー名が20文字を超えると、フォントサイズが小さくなったり、切り捨てられたりする可能性があり、ウォーターマークの効果が低下します。

デフォルトでは、**ログオンユーザー名を含める**は有効になっています。

VDAホスト名を含める

この設定を有効にすると、セッションは現在のICA®セッションのVDAホスト名をウォーターマークとして表示します。

デフォルトでは、**VDAホスト名を含める**は有効になっています。

VDA IPアドレスを含める

この設定を有効にすると、セッションは現在のICAセッションのVDA IPアドレスをウォーターマークとして表示します。

デフォルトでは、**VDA IPアドレスを含める**は無効になっています。

セッションウォーターマークのスタイル

この設定は、単一のウォーターマークテキストラベルを表示するか、複数のラベルを表示するかを制御します。**値**ドロップダウンメニューから**複数**または**単一**を選択します。 その他のスタイルオプションについては、この記事の**ウォーターマークカスタムテキスト**セクションを参照してください。

「**複数**」は、セッションに5つのウォーターマークラベルを表示します。中央に1つ、四隅に4つです。

「**単一**」は、セッションの中央に単一のウォーターマークラベルを表示します。

デフォルトでは、**セッションウォーターマークスタイル**は**複数**です。

透かしの透明度

ウォーターマークの不透明度を0から100まで指定できます。指定する値が大きいほど、ウォーターマークは不透明になります。

デフォルトでは、値は17です。

ウォーターマークのカスタムテキスト

値はデフォルトで空です。空で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`	-
`<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設定します。アルファチャネルを持つPNGのみがサポートされています。PNG

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
				ウォーターマークを使用している場合、`<style>`、`<position>`、`<rotation>`、`<transparency>`、および`<fontzoom>`の構文オプションのみが有効になります。
`<date>`	セッション接続日 (YYYY/MM/DD) のプレースホルダー	該当なし	該当なし	-
`<time>`	セッション接続時刻 (HH:MM) のプレースホルダー	該当なし	該当なし	-
`<domain>`	ユーザーアカウントドメインのプレースホルダー	該当なし	該当なし	-
`<username>`	現在のログオンユーザー名 (ユーザーアカウントドメインを除く) のプレースホルダー	該当なし	該当なし	-
`<hostname>`	VDAのホスト名のプレースホルダー	該当なし	該当なし	-

構文オプション	説明	有効な設定（大文字と小文字を区別）	デフォルト値	備考
`<clientip>`	クライアントのIPアドレスのプレースホルダー	該当なし	該当なし	-
`<serverip>`	VDAのIPアドレスのプレースホルダー	該当なし	該当なし	-

注:

ウォーターマークのカスタムテキスト が有効な構文設定で指定されている場合、**セッション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、既定値が使用されます。

制限事項

- セッションウォーターマークは、次のいずれかのケースでサポートされます。
  - が に設定されている場合。
  - が に設定され、3Dグラフィックワークロード向けに最適化 が有効になっている場合。
- セッションウォーターマークは、ブラウザーコンテンツリダイレクトが使用されているセッションでは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、ブラウザーコンテンツリダイレクトが無効になっていることを確認してください。
- レガシーNVIDIAドライバーで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が2に設定されます。）
- 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。

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、セッションのインタラクティブ性が低下する可能性があります。例えば、Webページのスクロールが遅くなったり、応答しなくなったり、途切れたりすることがあります。キーボードとマウスの操作は、グラフィックの更新に遅れが生じる可能性があります。

バージョン7.17までは、ポリシー設定を使用して、セッションの視覚品質を**低**に構成するか、より低い色深度（16ビットまたは8ビットグラフィック）を設定することで、帯域幅の消費を削減できました。しかし、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、次のいずれかのケースでデフォルトでプログレッシブ更新モードに切り替わります。

- 利用可能な帯域幅が2 Mbpsを下回る。
- ネットワーク遅延が200ミリ秒を超える。

このモードでは：

例えば、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**F**と**e**に青いアーティファクトがあり、画像は高度に圧縮されています。このアプローチにより帯域幅の消費が大幅に削減され、画像とテキストをより迅速に受信できるようになり、セッションのインタラクティブ性が向上します。

Thinwire プログレッシブ表示 1 の画像(/en-us/linux-virtual-delivery-agent/2301/media/thinwireprogressivedisplay.png)

セッションとの対話を停止すると、劣化した画像とテキストは、徐々にロスレスにシャープ化されます。例えば、以下のグラフィックでは、文字に青いアーティファクトは含まれなくなり、画像はソース品質で表示されます。

Thinwire プログレッシブ表示 2 の画像(/en-us/linux-virtual-delivery-agent/2301/media/thinwireprogressivedisplay2.png)

画像の場合、シャープ化にはランダムなブロック状の方法が使用されます。テキストの場合、個々の文字または単語の一部がシャープ化されます。シャープ化プロセスは複数のフレームにわたって

行われます。このアプローチにより、単一の大きなシャープ化フレームによる遅延の発生が回避されます。

一時的な画像（ビデオ）は、アダプティブディスプレイまたはSelective H.264で引き続き管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは、**画質**ポリシー設定の**高**、**中**（デフォルト）、および**低**に対してスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフになります（使用されません）。

- **視覚品質** = **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** = **画面全体に対して**（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化しても、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\>：

0 = 常にオフ（いかなる状況でも使用しない）

1 = 自動（ネットワークの状態に基づいて切り替え、デフォルト値）

2 = 常にオン

自動モード（1）の場合、次のいずれかのコマンドを実行して、プログレッシブモードが切り替わるしきい値を変更できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

ここで、\<value\> は \<Kbps単位のしきい値\> です（デフォルト = 2,048）。

例: 4096 = 帯域幅が4 Mbpsを下回った場合、プログレッシブモードをオンにする

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

ここで、\<value\> は \<ミリ秒単位のしきい値\> です (デフォルト = 200)

例: 100 = ネットワーク遅延が100ミリ秒を下回った場合、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) 上でクライアントドライブマッピングのみを有効にした場合、,	クライアント側のフルボリュームは、ホームディレクトリ内の **ctxmnt** サブディレクトリの下にあるセッションに自動的にマッピングされます。
ホスト (VDA) 上でクライアントフォルダーリダイレクトを有効にし、ユーザーがユーザーデバイス (クライアント) 上でそれを構成した場合、,	ユーザーによって指定されたローカルボリュームの一部がリダイレクトされます。

USBデバイスリダイレクト

USBデバイスはCitrix Workspace™ アプリとLinux VDAデスクトップ間で共有されます。USBデバイスがデスクトップにリダイレクトされると、ローカルに接続されているかのようにUSBデバイスを使用できます。

クライアントドライブマッピング

クライアントドライブマッピングとクライアントフォルダーリダイレクトを使用して、クライアント側のファイルをホスト側のセッションでアクセス可能にすることができます。クライアントドライブマッピングとクライアントフォルダーリダイレクトの比較は次のとおりです。

条件	結果
ホスト (VDA) でクライアントドライブマッピングのみを有効にする場合、,	クライアント側のフルボリュームは、ホームディレクトリ内の **ctxmnt** サブディレクトリの下のセッションに自動的にマップされます。
ホスト (VDA) 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、主に次の3つの機能領域があります。

- オープンソースUSB/IPプロジェクト
- Citrix USBセッションモジュール
- Citrix USBサービスモジュール

オープンソースUSB/IPプロジェクト:

USB/IPプロジェクトは、Linuxカーネルドライバーと、カーネルドライバーと通信してすべてのUSBデータを取得できるいくつかのユーザーモードライブラリで構成されています。

Linux VDAは、オープンソースのUSB/IPプロジェクトに基づいてUSBデバイスリダイレクトを実装し、USB/IPのカーネルドライバーとユーザーモードライブラリを再利用します。ただし、Linux VDAとCitrix Workspaceアプリ間のすべてのUSBデータ転送は、Citrix ICA USBプロトコルによってカプセル化されます。

Citrix USBセッションモジュール:

Citrix USBセッションモジュールは、USB/IPカーネルモジュールとCitrix Workspaceアプリ間の通信ブリッジとして機能します。

Citrix USBサービスモジュール:

Citrix USBサービスモジュールは、USBデバイスの接続や切断など、USBデバイスに対するすべての操作を管理します。

USBデバイスリダイレクトの仕組み

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム /dev パスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで使用できない場合があります。USBデバイスは正常に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスできません。USBデバイスの接続性を確保するには、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。その他のデバイスは自由に使用できますが、予期しない結果が生じる可能性があります。

USB マスストレージデバイス	VID:PID	ファイルシステム
`Netac Technology Co., Ltd`	0dd8:173c	FAT32, NTFS
`Kingston Datatraveler 101 II`	0951:1625	FAT32, NTFS
`Kingston Datatraveler GT101 G2`	1567:8902	FAT32, NTFS
`SanDisk SDCZ80 flash drive`	0781:5580	FAT32, NTFS
`WD HDD`	1058:10B8	FAT32、NTFS

USB マスストレージデバイス	VID:PID	ファイルシステム
`Toshiba Kingston DataTraveler 3.0 USB device`	0930:6545	FAT32、NTFS
`Taiwan OEM – OBSOLETE VendorCo ProductCode Disk 2.0`	FFFF:5678	FAT32、NTFS
`TD-RDF5A Transcend USB device`	8564:4000	FAT32、NTFS

注:

Amazon Linux 2、CentOS、RHEL、Rocky Linux、SUSEでNTFSを使用するには、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

USB WEBカメラ	VID:PID
`Logitech composite USB device – WebCam, Audio`	0460:0825

USBデバイスリダイレクトの設定

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 (CentOS、RHEL、Rocky Linuxのみ)

Linux VDA は、USB デバイスリダイレクトの仮想ホストコントローラーとして USB/IP を使用します。ほとんどの場合、USB/IP カーネルモジュールは Linux カーネルバージョン 3.17 以降でリリースされているため、デフォルトでカーネルモジュールをビルドする必要はありません。ただし、USB/IP カーネルモジュールは CentOS、RHEL、Rocky Linux では利用できません。これらの Linux ディストリビューションで USB デバイスリダイレクトを使用するには、USB/IP カーネルモジュールをインストールまたはコンパイルする必要があります。お使いの Linux ディストリビューションに基づいて、`<https://pkgs.org/download/kmod-usbip>` から USB/IP をダウンロードしてインストールしてください。

USBデバイスリダイレクトポリシーの設定

Citrixポリシーは、USBデバイスリダイレクトが有効か無効かを制御します。デバイスの種類は、Delivery Controller™ポリシーを使用して指定することもできます。Linux VDAのUSBデバイスリダイレクトを構成する場合は、次のポリシーとルールを構成します。

- クライアントUSBデバイスリダイレクトポリシー
- クライアントUSBデバイスリダイレクト規則

USBデバイスリダイレクトを有効にする

Citrix Studioで、クライアントからのUSBデバイスリダイレクトを有効（または無効）にします（ワークステーションホストのみ）。

設定の編集ダイアログで、:

1. **許可**を選択します。
2. **OK**をクリックします。

USBデバイスリダイレクトを有効にする(/en-us/linux-virtual-delivery-agent/2301/media/client-usb-redirection-policy.png)

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）され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、:

1. リダイレクト規則を追加するには**新規**をクリックし、既存の規則を確認するには**編集**をクリックします。

2. 規則を作成（または編集）した後、**OK**をクリックします。

クライアントUSBデバイスリダイレクト規則の設定(/en-us/linux-virtual-delivery-agent/2301/media/client-usb-redirection-rule.png)

USBデバイスリダイレクトの問題をトラブルシューティングする

このセクションの情報を使用して、Linux VDAの使用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ます。

リダイレクトされたUSBディスクをアンマウントできない

Linux VDAは、Citrix Workspaceアプリからリダイレクトされたすべてのユーザーに対して、USBディスクを管理者権限で管理し、所有者のみがリダイレクトされたデバイスにアクセスできるようにします。そのため、デバイスをアンマウントするには管理者権限が必要です。

デバイスをアンマウントできない(/en-us/linux-virtual-delivery-agent/2301/media/unable-to-mount-disk.png)

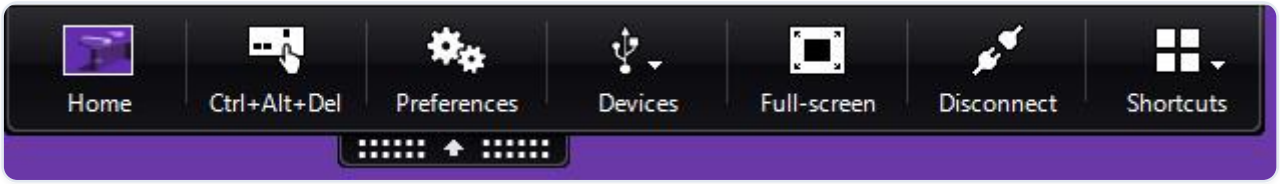
USBディスクのリダイレクトを停止するとファイルが失われる

Citrix Workspaceアプリのツールバーを使用してUSBディスクのリダイレクトをすぐに停止すると、ディスク上で変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込む際に、システムがファイルシステム内のメモリキャッシュをマウントするため発生します。データはディスク自体には書き込まれません。Citrix Workspaceアプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。

この問題を解決するには、USBリダイレクトを停止する前に、ターミナルで**sync**コマンドを使用してデータをディスクにフラッシュします。

Citrix Workspaceアプリのツールバーにデバイスが表示されない

Citrix Workspaceアプリのツールバーにデバイスが一覧表示されない場合があります。これは、USBリダイレクトが行われていないことを示しています。



この問題が発生した場合は、以下を確認してください。

- ポリシーがUSBデバイスリダイレクトを許可するように構成されている。
- Citrix USBサービスモジュールが実行されている。

ポリシーが正しく設定されていない場合は、この記事の「USBデバイスリダイレクトポリシーの設定」セクションを参照して修正してください。

Citrix USBサービスモジュールが実行されていない場合は、次の手順を実行します。

1. 次のコマンドを使用して、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、Linuxディストリビューションに基づいてUSB/IPカーネルモジュールをインストールまたはコンパイルします。

   ```
   modinfo: ERROR: Module usbip-core not found.
   ```

   - Amazon Linux 2、CentOS、RHEL、Rocky Linuxについては、この記事の「USB/IPカーネルモジュールのインストールまたはコンパイル」セクションを参照してください。
   - SUSEの場合は、`<https://software.opensuse.org/package/usbip>`からUSB/IPパッケージをダウンロードしてインストールします。
   - Ubuntu/Debianの場合は、次の手順を実行してUSB/IPカーネルモジュールをコンパイルおよびインストールします。

     a. USB/IPカーネルモジュールのソースコードをダウンロードします。

     `<https://github.com/torvalds/linux/tree/master/drivers/usb/usbip>`にあるLinuxカーネルリポジトリに移動し、ターゲットのLinuxカーネルバージョン（v4.15以降）タグを選択して、`<https://github.com/torvalds/linux/tree/v4.15/drivers/usb/usbip>`のようなリンクを取得します。

     DownGitにアクセスし、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てください。

     b. 次のコマンドを使用してソースファイルを解凍します:

     ```
     unzip ${USBIP_SRC}.zip



     cd usbip
     ```

c. **Makefile**ファイルを次のように変更します:

```
# SPDX-License-Identifier: GPL-2.0


ccflags-$(CONFIG_USBIP_DEBUG) := -DDEBUG


obj-$(CONFIG_USBIP_CORE) += usbip-core.o


usbip-core-y := usbip_common.o usbip_event.o


obj-$(CONFIG_USBIP_VHCI_HCD) += vhci-hcd.o


vhci-hcd-y := vhci_sysfs.o vhci_tx.o vhci_rx.o vhci_hcd.o


#obj-$(CONFIG_USBIP_HOST) += usbip-host.o


#usbip-host-y := stub_dev.o stub_main.o stub_rx.o stub_tx.o


#obj-$(CONFIG_USBIP_VUDC) += usbip-vudc.o


#usbip-vudc-y := vudc_dev.o vudc_sysfs.o vudc_tx.o vudc_rx.o
vudc_transfer.o vudc_main.o
```

d. ソースコードをコンパイルします:

```
apt-get install linux-headers-`uname -r`



make -C /lib/modules/`uname -r`/build M=$PWD
```

e. USB/IPカーネルモジュールをインストールします:

```
cp usbip-core.ko vhci-hcd.ko /opt/Citrix/VDA/lib64/
```

f. **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:

```
service ctxusbsd restart
```

USBデバイスがCitrix Workspaceアプリのツールバーに表示されるが、*ポリシーによって制限されていますと*表示される場合のリダイレクトの失敗

この問題が発生した場合、次の操作を実行します:

- Linux VDAポリシーを構成して、リダイレクトを有効にします。
- Citrix Workspaceアプリのレジストリに追加のポリシー制限が構成されていないか確認します。レジストリパスの**DeviceRules**を確認して、この設定によってデバイスへのアクセスが拒否されていないことを確認します:

  HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Citrix\\ICA Client\\GenericUSB

USBデバイスは正常にリダイレクトされますが、セッションで使用できません

通常、サポートされているUSBデバイスのみがリダイレクトできます。他のデバイスもアクティブなLinux VDAセッションにリダイレクトされる場合があります。リダイレクトされたすべてのデバイスについて、ユーザーが所有するノードがシステム**/dev**パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている（接続されている）がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限のポリシーに追加します。

> **注:**

> USBドライブの場合、Linux VDAはディスクを構成してマウントします。ユーザー（およびそれをインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。サポートされているデバイスリストにないデバイスでは、そうではない場合があります。

キーボード

このセクションには、次のトピックが含まれています。

- クライアントIME
- クライアントIMEユーザーインターフェイスの同期
- 動的キーボードレイアウトの同期
- ソフトキーボード
- 複数言語入力のサポート

クライアント入力メソッドエディター (IME)

概要

中国語、日本語、韓国語などの2バイト文字は、IMEを介して入力する必要があります。そのような文字は、クライアント側の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™セッションを起動します。

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式を変更します。

既知の問題

- クライアントIME機能を使用してGoogleスプレッドシートのセルに文字を入力する前に、そのセルをダブルクリックする必要があります。
- クライアントIME機能は、パスワードフィールドでは自動的に無効になりません。
- IMEユーザーインターフェイスは、入力領域でカーソルに追従しません。

クライアントIMEユーザーインターフェースの同期

概要

これまで、クライアントIMEのユーザーインターフェース（変換ウィンドウや候補ウィンドウを含む）は、画面の左上隅に配置されていました。カーソルに追従せず、テキスト入力領域のカーソルから離れた場所に表示されることがありました。

以前のクライアントIMEユーザーインターフェースの画像(/en-us/linux-virtual-delivery-agent/2301/media/composition-and-candidate-windows.png)

Citrix®は、クライアントIMEの使いやすさを向上させ、ユーザーエクスペリエンスを次のようにさらに改善します。

現在のクライアントIMEユーザーインターフェースの画像(/en-us/linux-virtual-delivery-agent/2301/media/synced-composition-and-candidate-windows.png)

この機能を使用するための前提条件

1. Linux VDAでIntelligent Input Bus (`IBus`) を有効にします。Linux OSで `IBus` を有効にする方法については、OSベンダーのドキュメントを参照してください。例：
   - ウブントゥ： `<https://help.ubuntu.com/community/ibus>`
   - CentOS、RHEL: `<https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods>`
   - デビアン： `<https://wiki.debian.org/I18n/ibus>`
   - SUSE: `<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-gnome-settings.html#sec-gnome-settings-lang>`
2. この機能は自動的にインストールされますが、使用する前に有効にする必要があります。

機能の有効化と無効化

クライアントIMEユーザーインターフェース同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、 `ctxreg` ユーティリティを介してレジストリを編集します。

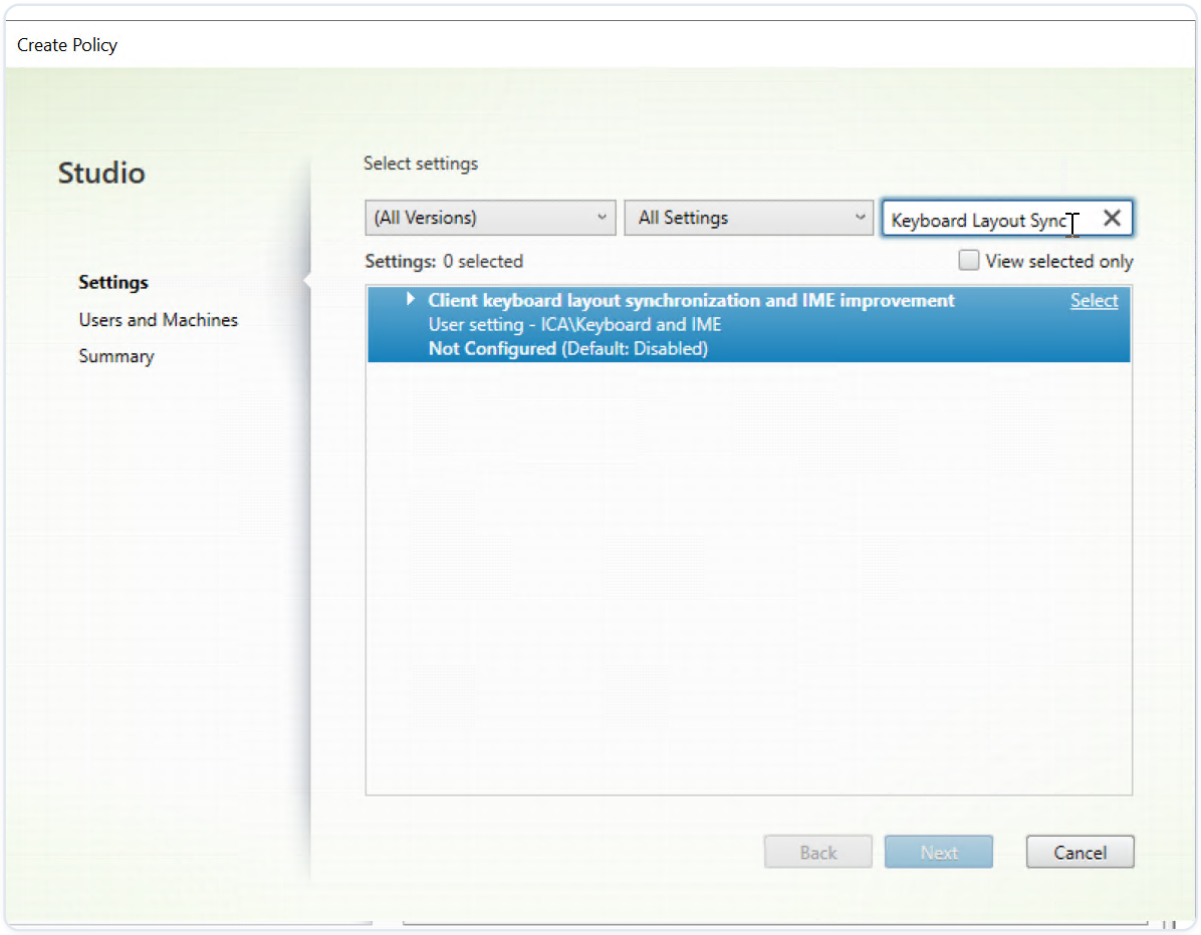
> **注:**

> **Client Keyboard Layout Sync and IME Improvement**ポリシーは、レジストリ設定よりも優先され、指定したユーザーオブジェクトおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- クライアントIMEユーザーインターフェイス同期機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定します。

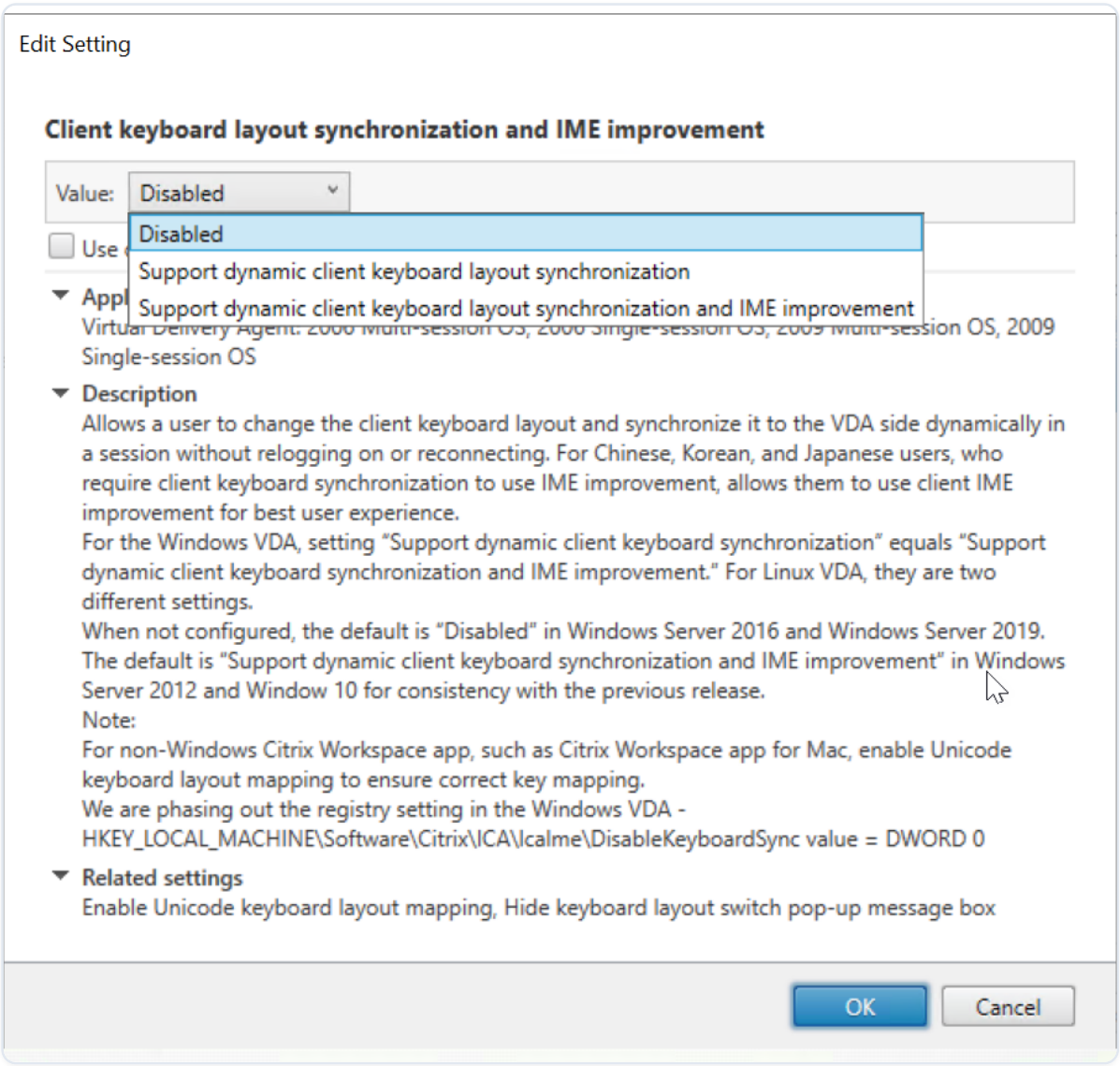
  a. Studioで、**ポリシー**を右クリックし、**ポリシーの作成**を選択します。

  b. **Client Keyboard Layout Sync and IME Improvement**ポリシーを検索します。



  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。



3つのオプションがあります。

- **無効**: 動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期を無効にします。
- **動的クライアントキーボードレイアウト同期をサポート**:
  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期を有効にします。
- **動的クライアントキーボードレイアウト同期とIMEの改善をサポート**:
  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期の両方を有効にします。

- `ctxreg`ユーティリティを介してレジストリを編集し、クライアントIMEユーザーインターフェイス同期機能を有効または無効にします。

  この機能を有効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v  "SyncClientIME" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します。

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncClientIME" -d "0x00000000"
  ```

動的キーボードレイアウト同期

以前は、Linux VDAとクライアントデバイスのキーボードレイアウトは同じである必要がありました。たとえば、クライアントデバイスでキーボードレイアウトが英語からフランス語に変更されたにもかかわらず、VDAでは変更されなかった場合、キーマッピングの問題が発生する可能性がありました。

Citrix®は、VDAのキーボードレイアウトをクライアントデバイスのキーボードレイアウトと自動的に同期させることで、この問題に対処します。クライアントデバイスのキーボードレイアウトが変更されるたびに、VDAのレイアウトもそれに追従します。

注:

HTML5向けCitrix Workspace™アプリは、動的キーボードレイアウト同期機能をサポートしていません。

構成

動的キーボードレイアウト同期機能は、デフォルトで無効になっています。この機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定するか、`ctxreg`ユーティリティを介してレジストリを編集します。

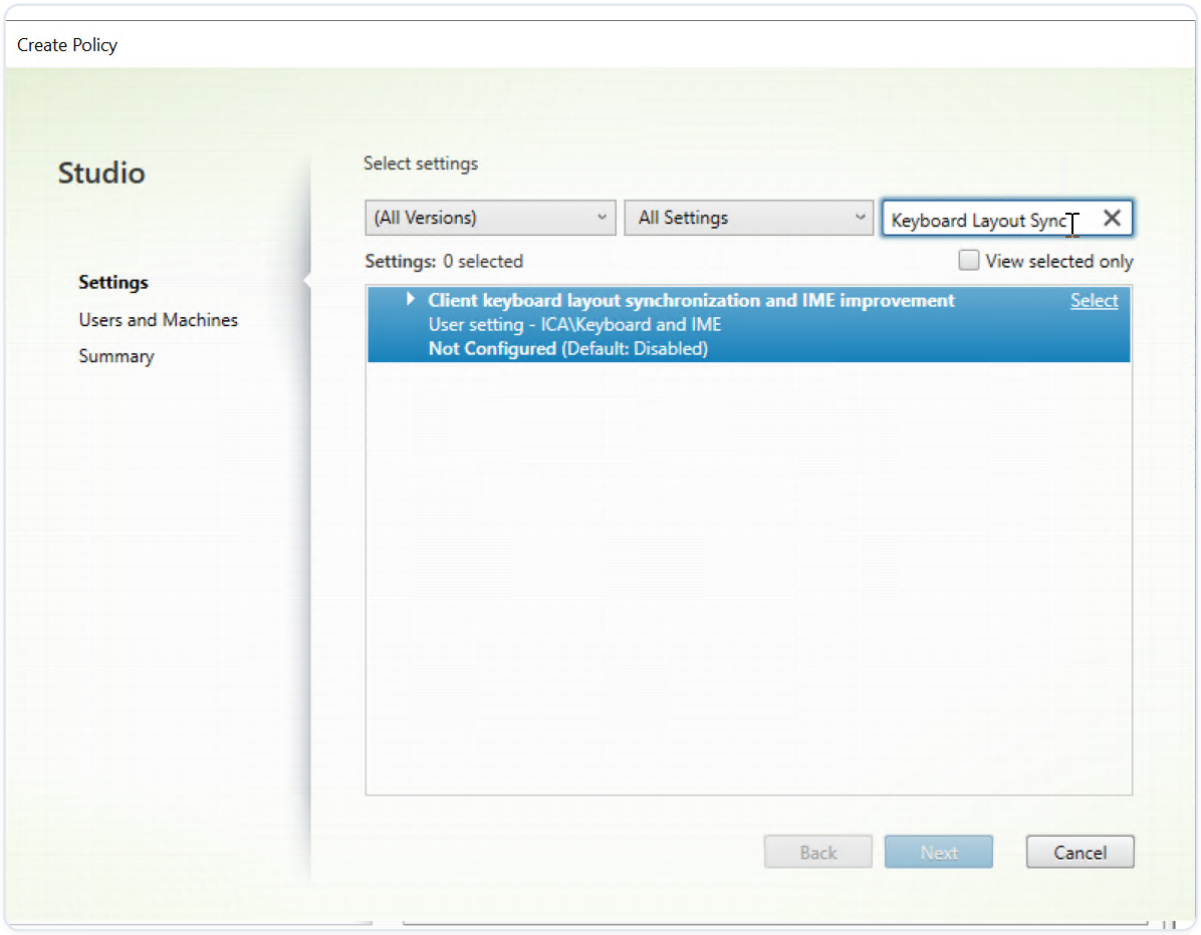
注:

> **Client Keyboard Layout Sync and IME Improvement**ポリシーは、レジストリ設定よりも優先され、指定したユーザーおよびマシンオブジェクト、またはサイト内のすべてのオブジェクトに適用できます。特定のLinux VDA上のレジストリ設定は、そのVDA上のすべてのセッションに適用されます。

- 動的キーボードレイアウト同期機能を有効または無効にするには、**Client Keyboard Layout Sync and IME Improvement**ポリシーを設定します。

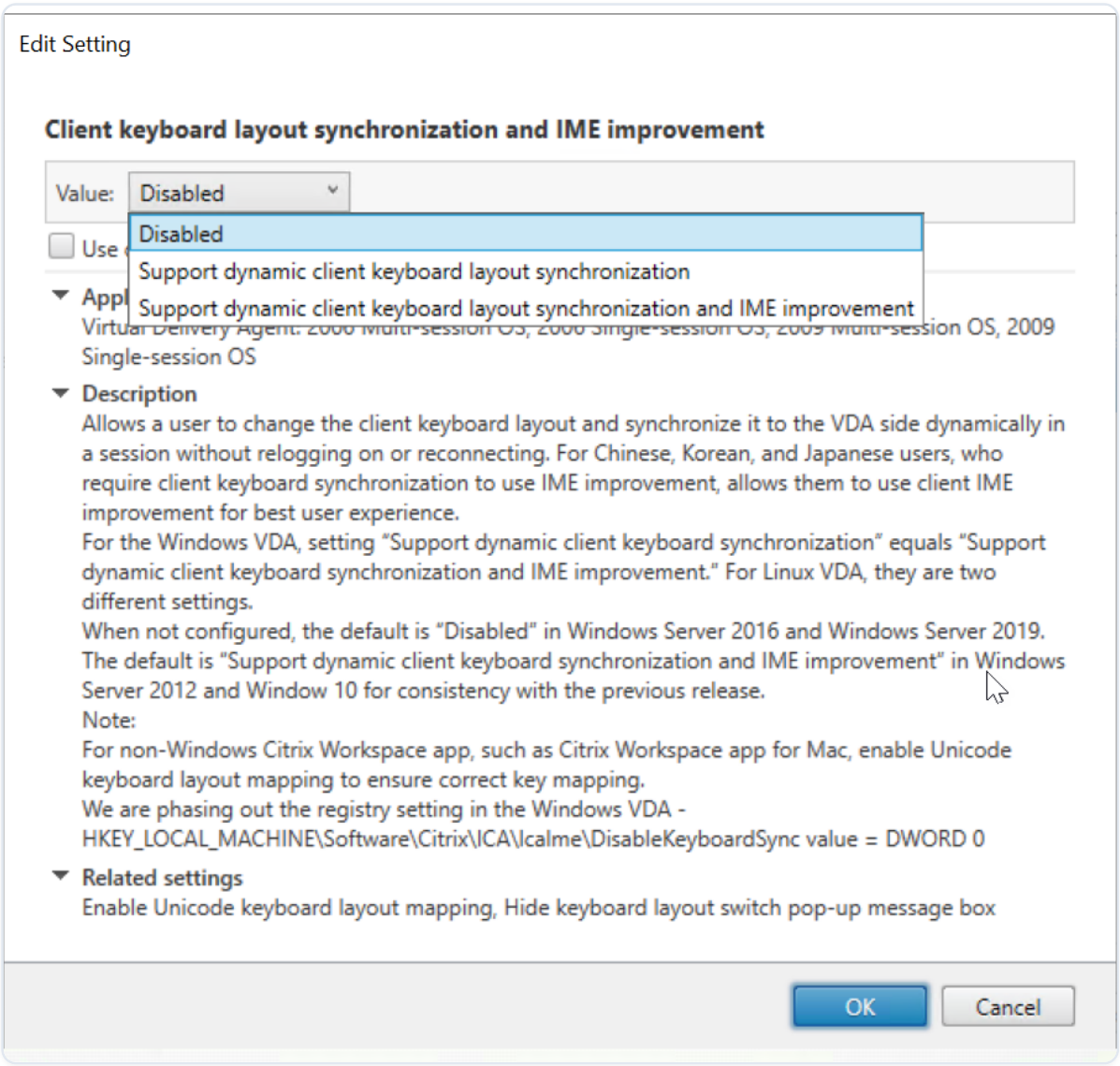
  a. Studioで、**Policies**を右クリックし、**Create Policy**を選択します。

  b. **Client Keyboard Layout Sync and IME Improvement**ポリシーを検索します。



  c. ポリシー名の横にある**Select**をクリックします。

  d. ポリシーを設定します。



3つのオプションがあります。

- **Disabled**：動的キーボードレイアウト同期とクライアントIMEユーザーインターフェイス同期を無効にします。
- **Support dynamic client keyboard layout synchronization**：`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**レジストリキーのDWORD値に関係なく、動的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。
- **動的なクライアントキーボードレイアウトの同期とIMEの改善をサポート**: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar`にある**SyncKeyboardLayout**および**SyncClientIME**レジストリキーのDWORD値に関係なく、動的なキーボードレイアウトの同期とクライアントIMEユーザーインターフェイスの同期の両方を有効にします。

- `ctxreg`ユーティリティを介してレジストリを編集し、動的なキーボードレイアウト同期機能を有効または無効にします:

  この機能を有効にするには、次のコマンドを実行します:

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000001"
  ```

  この機能を無効にするには、次のコマンドを実行します:

  ```
  /opt/Citrix/VDA/bin/ctxreg update -k
  "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar
  " -v "SyncKeyboardLayout" -d "0x00000000"
  ```

使用方法

この機能が有効になっている場合、セッション中にクライアントデバイス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/2301/media/keyboard-changed-to-french.png)

その後、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/2301/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/2301/media/fr-in-the-task-bar.png)

ソフトキーボード

ソフトキーボード機能は、Linux仮想デスクトップまたはアプリケーションセッションで利用できます。ソフトキーボードは、入力フィールドに入ると自動的に表示され、入力フィールドを離れると自動的に非表示になります。

ソフトキーボード(/en-us/linux-virtual-delivery-agent/2301/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:**
>
> 上記の設定は、新しいセッションにログオンしたとき、または現在のセッションからログオフして再度ログオンしたときに有効になります。

制限事項

- この機能は、Google Chrome、LibreOffice、およびその他のアプリでは期待どおりに動作しない場合があります。
- ソフトキーボードを手動で非表示にした後、再度表示するには、入力フィールド以外の場所をクリックしてから、現在の入力フィールドを再度クリックします。
- Webブラウザーで1つの入力フィールドから別の入力フィールドにクリックしても、ソフトキーボードが表示されない場合があります。この問題を回避するには、入力フィールド以外の場所をクリックしてから、目的の入力フィールドをクリックします。
- この機能は、Unicode文字および2バイト文字（中国語、日本語、韓国語などの文字）をサポートしていません。
- ソフトウェアキーボードは、パスワード入力フィールドでは利用できません。
- ソフトウェアキーボードが現在の入力フィールドと重なる場合があります。この場合、アプリのウィンドウを移動するか、画面をスクロールして入力フィールドをアクセス可能な位置に移動してください。
- Citrix WorkspaceアプリとHuaweiタブレット間の互換性の問題により、物理キーボードが接続されている場合でも、Huaweiタブレットではソフトウェアキーボードが表示されます。

複数言語入力のサポート

Linux VDA バージョン 1.4 以降、Citrix は公開アプリケーションのサポートを追加しました。ユーザーは、Linux デスクトップ環境なしで、目的の Linux アプリケーションにアクセスできます。

しかし、Linux VDA のネイティブ言語バーは、公開アプリケーションでは利用できませんでした。これは、言語バーが Linux デスクトップ環境と密接に統合されているためです。その結果、ユーザーは中国語、日本語、韓国語など、IME を必要とする言語でテキストを入力できませんでした。また、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け付ける公開アプリケーション用の言語バーを提供します。言語バーを使用すると、ユーザーはサーバー側の IME を選択し、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

ctxreg ユーティリティを使用して、この機能を有効または無効にできます（デフォルトでは無効）。特定の Linux VDA サーバーでの機能構成は、その VDA で公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \\SYSTEM\\CurrentControlSet\\Control\\Citrix\\LanguageBar」で、種類は DWORD です。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar"
-v "Enabled" -d "0x00000000"
```

使用方法

使用方法は簡単です。

1. 機能を有効にします。
2. テキスト入力を受け付ける公開アプリケーションにアクセスします。セッションに、アプリケーションの横に言語バーが表示されます。
3. ドロップダウンメニューから、**地域と言語**を選択して、目的の言語（入力ソース）を追加します。

   地域と言語の画像(/en-us/linux-virtual-delivery-agent/2301/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™アプリを実行中）でも同じキーボードレイアウトが使用されていることを確認してください。
- **Region & Language**ダイアログボックスで設定を行う前に、**accountsservice**パッケージをバージョン0.6.37以降にアップグレードする必要があります。

IMEまたはキーボードレイアウトを選択するイメージ(/en-us/linux-virtual-delivery-agent/2301/media/region-language-dialog-box.png)

マルチメディア

このセクションには、次のトピックが含まれています。

- オーディオ機能
- ブラウザコンテンツリダイレクト
- HDX Webカメラビデオ圧縮

オーディオ機能

アダプティブオーディオ

アダプティブオーディオはデフォルトで有効になっています。次のCitrix Workspace™アプリクライアントをサポートしています。

- Windows向けCitrix Workspaceアプリ – 2109以降のバージョン
- Linux向けCitrix Workspaceアプリ – 2109以降のバージョン
- Mac向けCitrix Workspaceアプリ – 2109以降のバージョン

リストにないクライアントを使用すると、アダプティブオーディオはレガシーオーディオにフォールバックします。

アダプティブオーディオを使用すると、VDAでオーディオ品質ポリシーを手動で構成する必要はありません。アダプティブオーディオは、ネットワークの状態に基づいてオーディオサンプリングビットレートを動的に調整し、プレミアムなオーディオエクスペリエンスを提供します。

次の表は、アダプティブオーディオとレガシーオーディオの比較を示しています。

アダプティブオーディオ	レガシーオーディオ
最大オーディオサンプルレート: 48 kHz	最大オーディオサンプルレート: 8 kHz
ステレオチャンネル	モノラルチャンネル

ヒント:

RHEL 8.xでPulseAudio 13.99以降を使用してください。

ブラウザコンテンツリダイレクト

概要

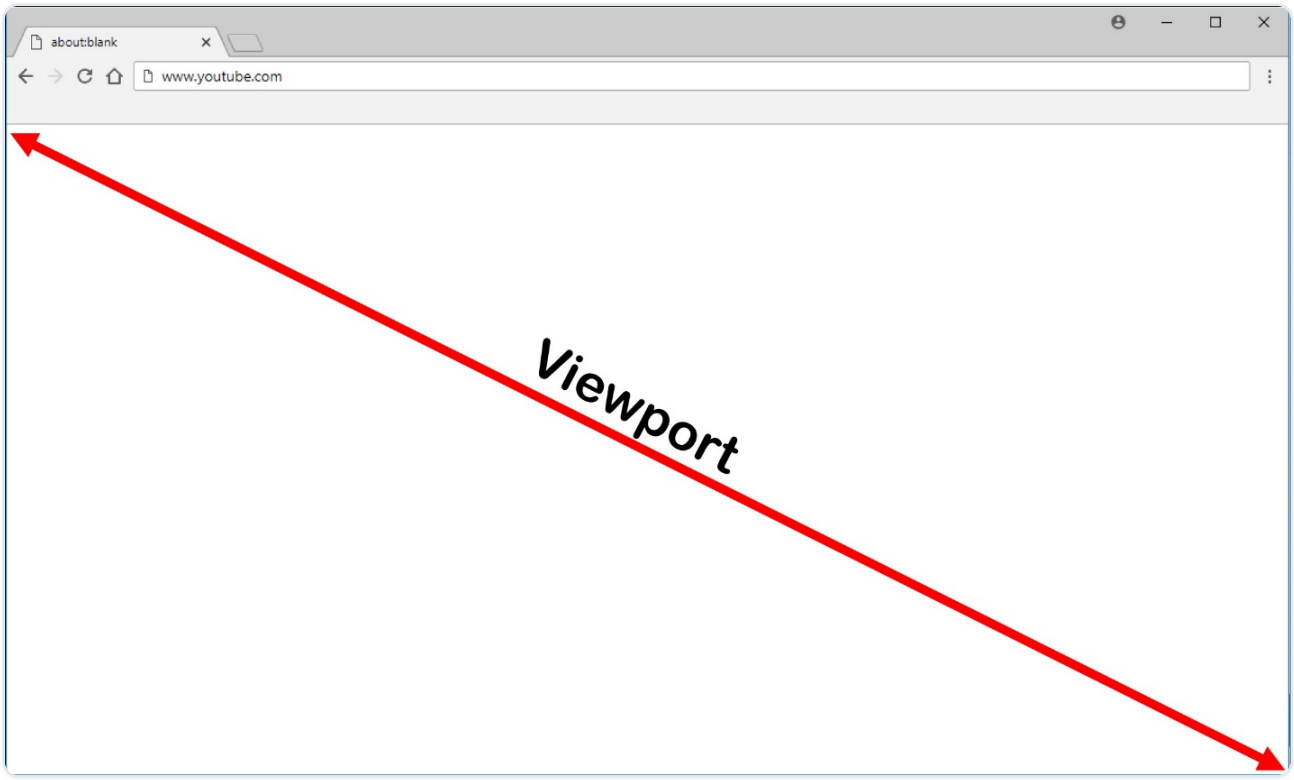
Linux VDAは、Google Chromeでのブラウザコンテンツリダイレクトをサポートしています。ブラウザコンテンツリダイレクトは、許可リストにあるWebページをクライアント側でレンダリングする機能を提供します。この機能は、Citrix Workspace™アプリを使用してクライアント側で対応するレンダリングエンジンをインスタンス化し、URLからHTTPおよびHTTPSコンテンツをフェッチします。

注:

許可リストを使用すると、クライアント側にリダイレクトされるWebページを指定できます。逆に、ブロックリストを使用すると、クライアント側にリダイレクトされないWebページを指定できます。

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。



システム要件

Windowsクライアント:

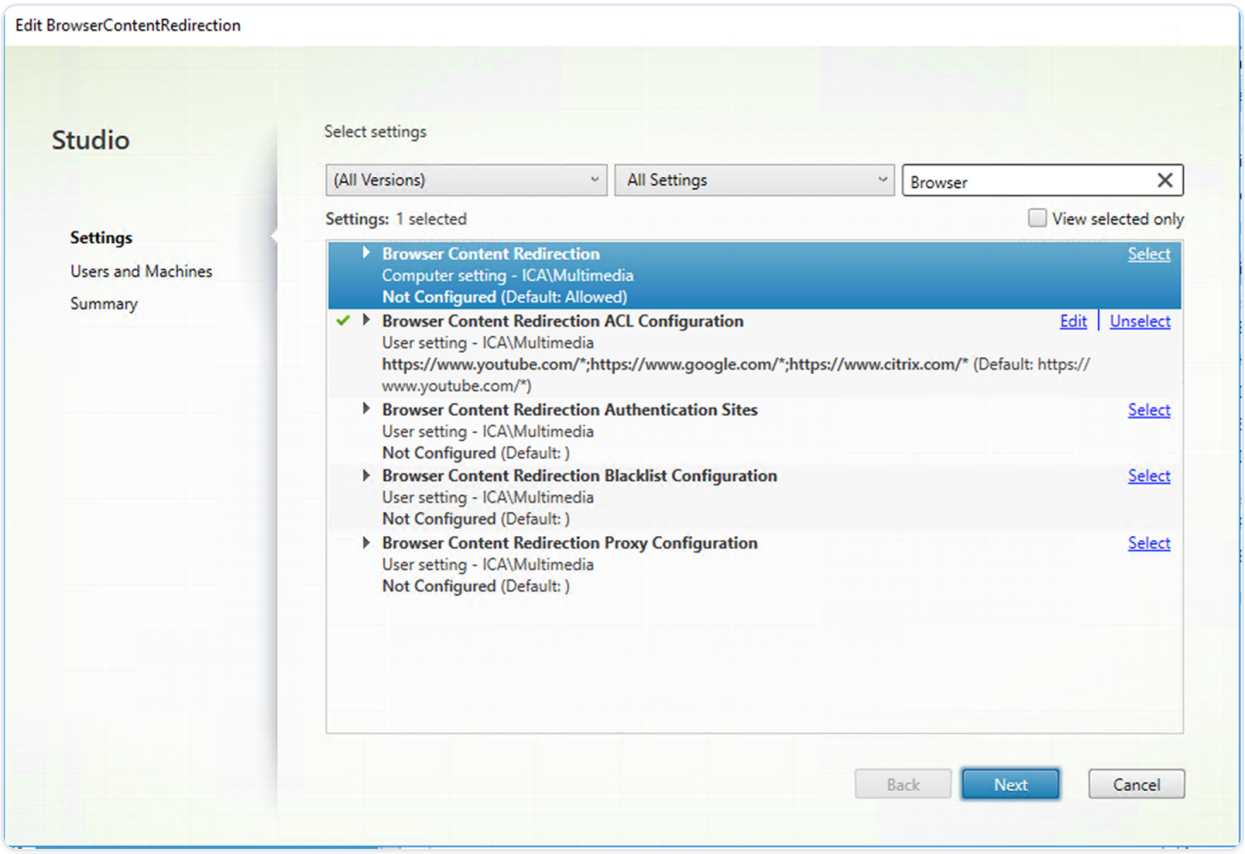
- Citrix Workspaceアプリ 1809 for Windows以降

リナックス VDA:

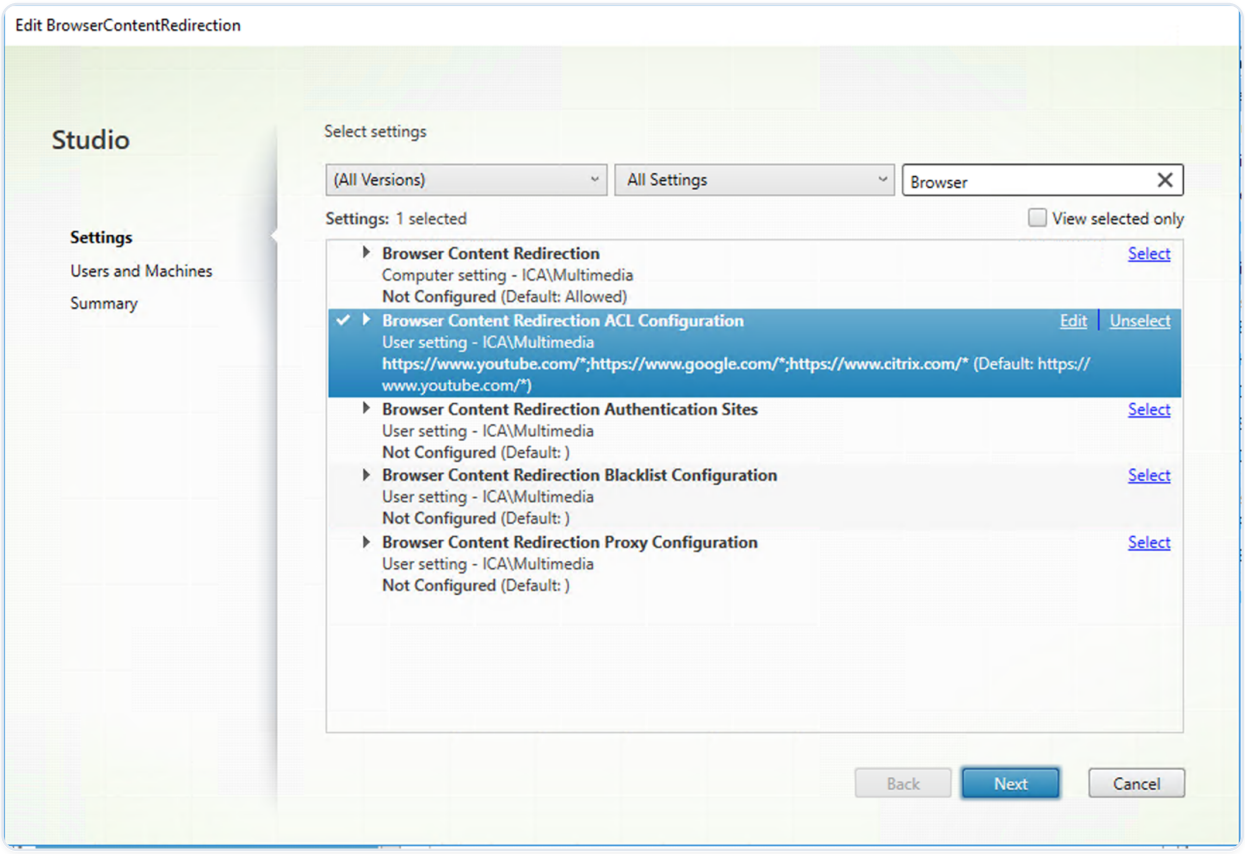
- VDA上のブラウザ: Citrix®ブラウザコンテンツリダイレクト拡張機能が追加されたGoogle Chrome v66以降

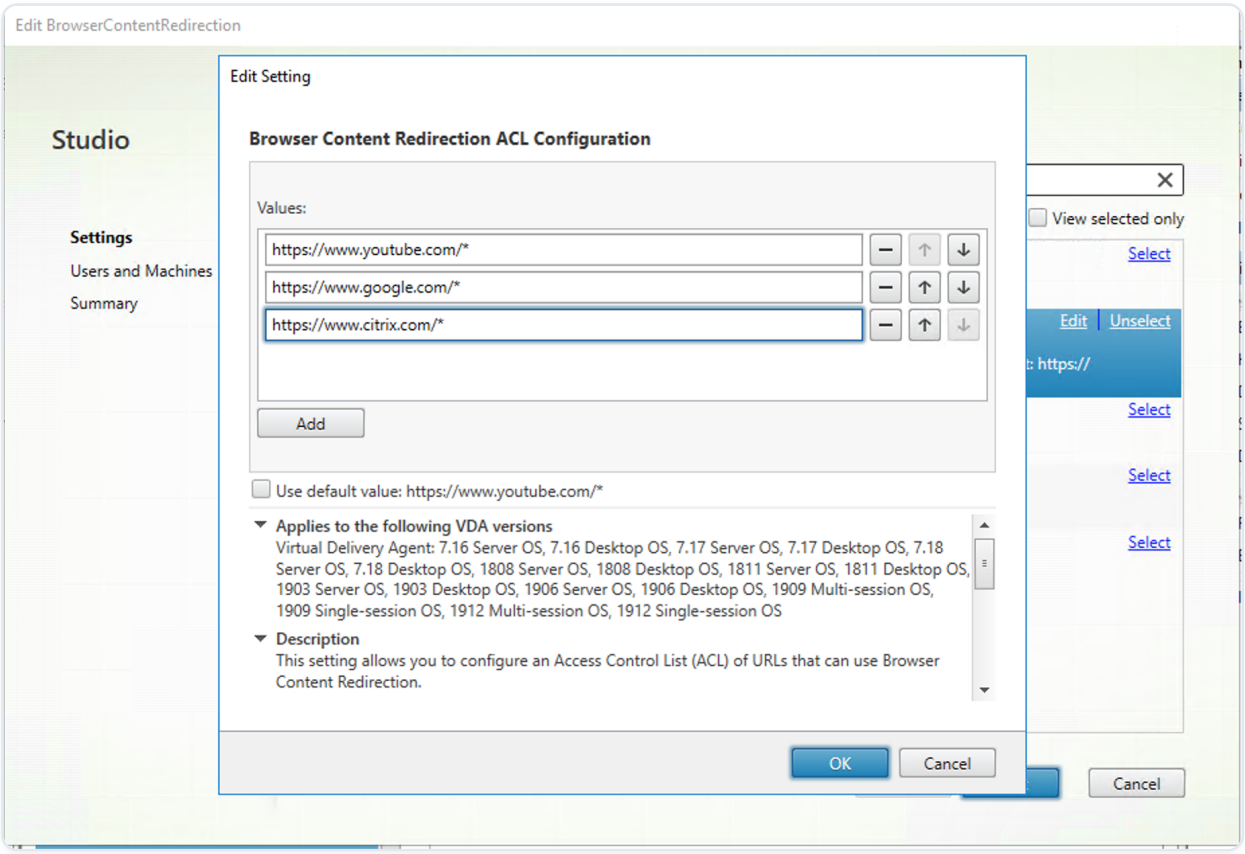
ブラウザコンテンツリダイレクトの構成

1. Citrix Studioで、ブラウザコンテンツリダイレクト用のURLの許可リストとブロックリストを指定するポリシーを構成します。ブラウザコンテンツリダイレクトは、デフォルトで**許可**に設定されています。



ブラウザコンテンツリダイレクトACL構成設定は、ブラウザコンテンツリダイレクトを使用できるURLの許可リストを指定します。





ブラウザコンテンツリダイレクトブラックリスト構成設定は、ブラウザコンテンツリダイレクトを使用できないURLのブロックリストを指定します。

ブロックリストを設定するイメージ(/en-us/linux-virtual-delivery-agent/2301/media/block-list.png)

> **注:**
>
> Linux VDA は現在、**ブラウザコンテンツリダイレクトプロキシ構成**設定をサポートしていません。

2. VDA で **Chrome に追加**をクリックして、Chrome ウェブストアから Citrix ブラウザコンテンツリダイレクト拡張機能を追加します。これにより、VDA 上のブラウザが、アクセス中の URL が許可リストまたはブロックリストと一致するかどうかを検出できるようになります。

> **重要:**
>
> この拡張機能はクライアントには不要です。VDA にのみ追加してください。

> \>

> Chrome 拡張機能はユーザーごとにインストールされます。拡張機能を追加または削除するためにゴールデンイメージを更新する必要はありません。

> Chrome ウェブストアから Citrix ブラウザコンテンツリダイレクト拡張機能を追加するイメージ(/en-us/linux-virtual-delivery-agent/2301/media/add-to-chrome.png)

URL が許可リスト（例: `https://www.mycompany.com/`）で見つかり、ブロックリストには見つからない場合、仮想チャネル (CTXCSB) は Citrix Workspace アプリにリダイレクトが必要であることを指示し、URL を中継します。その後、Citrix Workspace アプリはローカルレンダリングエンジンをインスタンス化し、ウェブサイトを表示します。

その後、Citrix Workspaceアプリはウェブサイトを仮想デスクトップのブラウザコンテンツ領域にシームレスにブレンドし直します。

ビューポートとクライアントのイメージ(/en-us/linux-virtual-delivery-agent/2301/media/viewport-desktop-viewer.png)

1. Citrix ブラウザコンテンツリダイレクト拡張機能のアイコン

   拡張機能アイコンの色は、Chrome 拡張機能のステータスを示します。色は次の3つのいずれかです。

   - 緑: アクティブで接続済み
   - グレー: 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るビューポート、または仮想デスクトップにブレンドされるビューポート
3. リナックス ブイディーエー
4. Windows クライアント

リダイレクトのシナリオ

Citrix Workspaceアプリがコンテンツをフェッチするシナリオを次に示します。

リダイレクトのシナリオの画像(/en-us/linux-virtual-delivery-agent/2301/media/browser-viewport-redirection-scenarios.png)

- **サーバーフェッチとサーバーレンダリング:** サイトを許可リストに追加しなかったか、リダイレクトが失敗したため、リダイレクトは行われません。VDAでWebページをレンダリングするフォールバックが行われ、Thinwireを使用してグラフィックがリモート処理されます。フォールバック動作を制御するには、ポリシーを使用します。このシナリオでは、VDAでCPU、RAM、帯域幅の消費量が高くなります。
- **クライアントフェッチとクライアントレンダリング:** Citrix WorkspaceアプリがWebサーバーに直接アクセスするため、インターネットアクセスが必要です。このシナリオでは、Citrix Virtual Apps and Desktops™サイトからすべてのネットワーク、CPU、RAMの使用量がオフロードされます。

フォールバックメカニズム

クライアントのリダイレクトが失敗する場合があります。たとえば、クライアントマシンに直接インターネットアクセスがない場合、エラー応答がVDAに戻されることがあります。そのような場合、VDA上のブラウザーがページを再読み込みし、サーバー上でレンダリングできます。

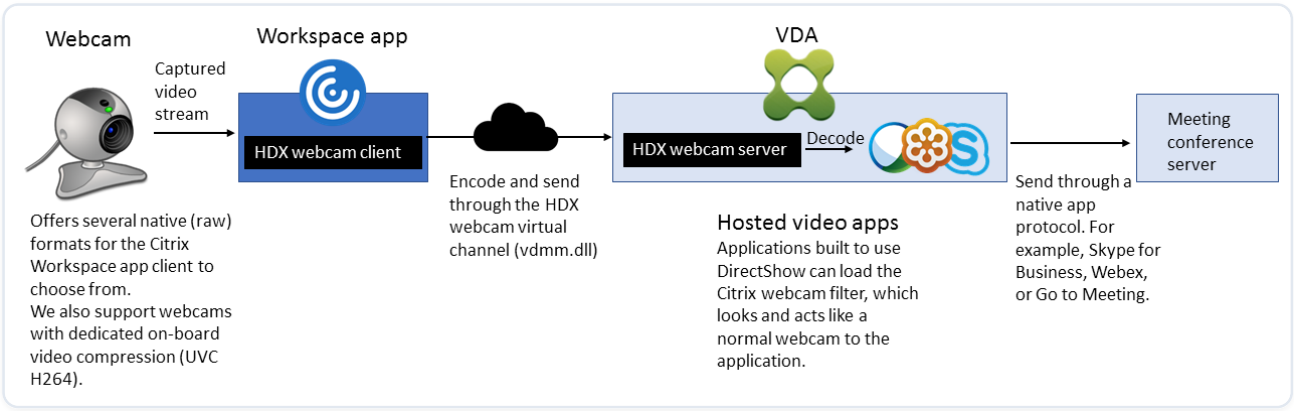
HDX™ Webカメラビデオ圧縮

概要

Linux VDAセッションで実行されているビデオ会議アプリケーションのユーザーは、HDX Webカメラビデオ圧縮を使用してWebカメラを使用できるようになりました。この機能はデフォルトで有効になっています。可能な場合は常にHDX Webカメラビデオ圧縮を使用することをお勧めします。

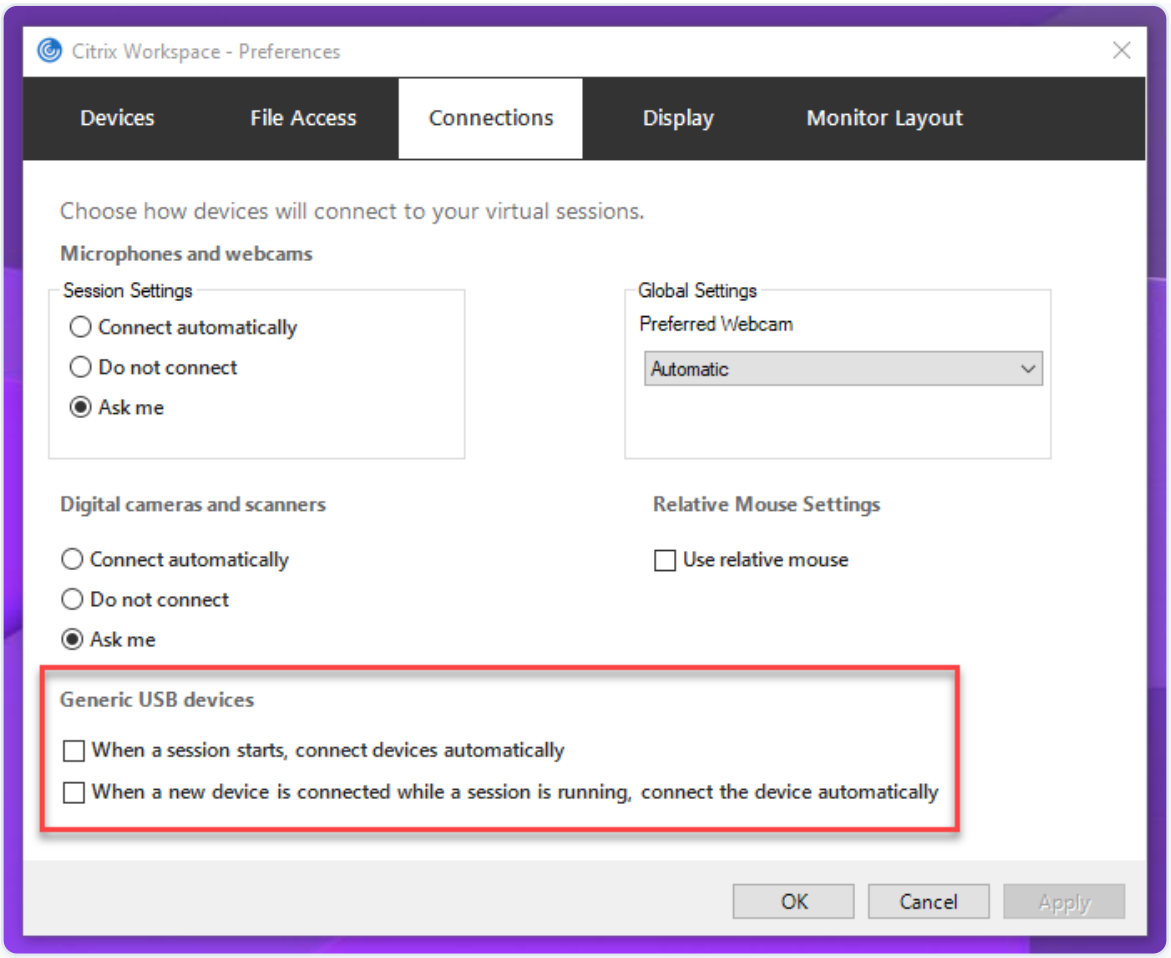
HDX Webカメラビデオ圧縮は、**最適化された**Webカメラモードとも呼ばれます。この種類のWebカメラビデオ圧縮は、H.264ビデオを仮想セッションで実行されているビデオ会議アプリケーションに直接送信します。HDX Webカメラビデオ圧縮は、クライアントオペレーティングシステムの一部であるマルチメディアフレームワークテクノロジーを使用して、キャプチャデバイスからのビデオを傍受し、トランスコードして圧縮します。キャプチャデバイスの製造元は、OSカーネルストリーミングアーキテクチャに接続するドライバーを提供します。

クライアントはWebカメラとの通信を処理します。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Webカメラを直接処理しませんが、その統合により、デスクトップで同じエクスペリエンスが得られます。Workspaceアプリは、帯域幅を節約し、WANシナリオでの回復性を向上させるためにビデオを圧縮します。



注:

- 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、この機能はAzureマシンでは利用できません。
- この機能は、Citrix WorkspaceアプリクライアントからのH.264ビデオのみをサポートします。
- サポートされているWebカメラの解像度は48x32から1920x1080の範囲です。
- Webカメラを使用しているときに、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Webカメラ

Webカメラによってフレームレートが異なり、明るさやコントラストのレベルも異なります。Citrix®は、初期機能検証のために次のWebカメラを使用しています。

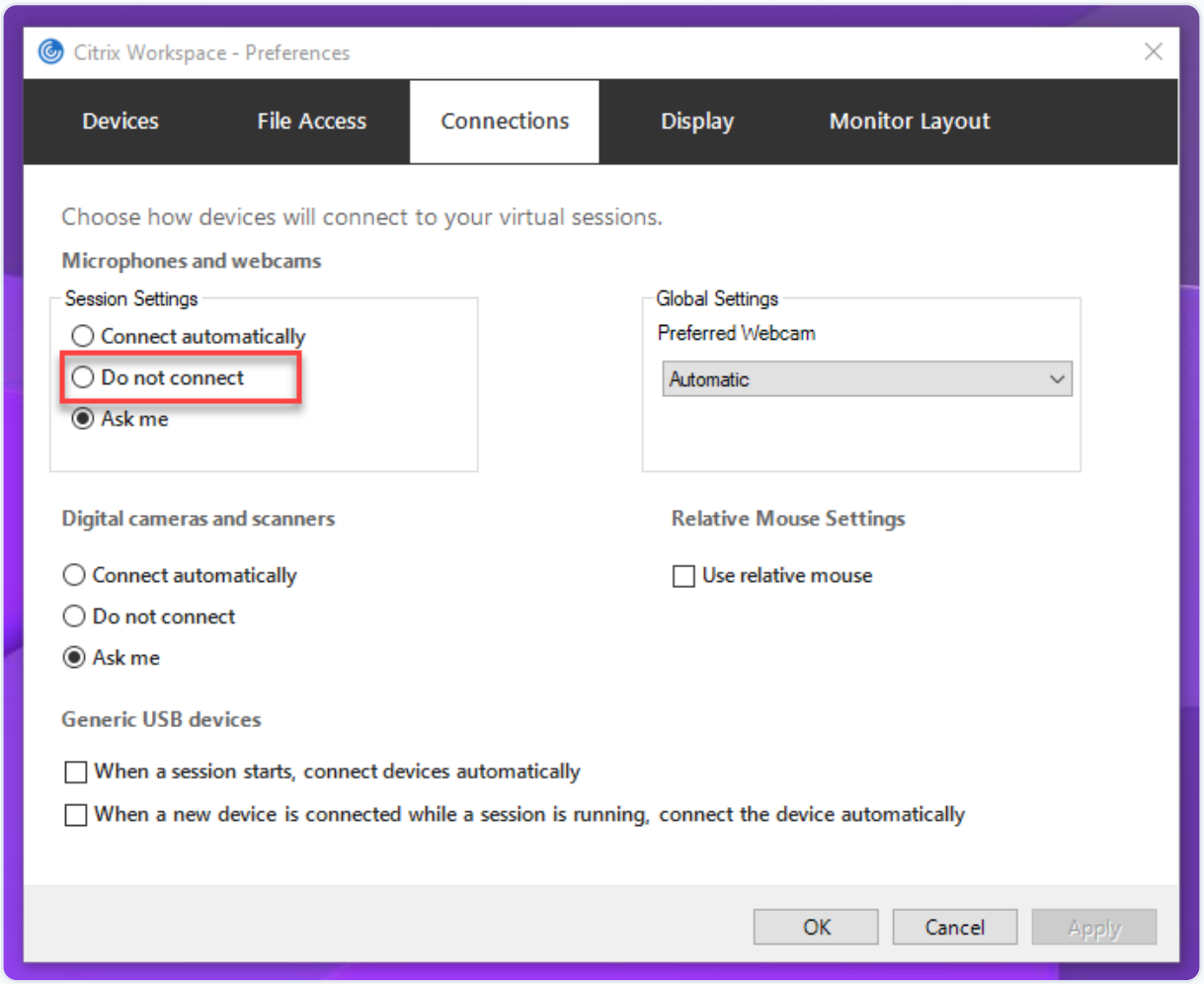
- Logitech HD ウェブカメラ C270
- Logitech ウェブカメラ C930e
- マイクロソフト ライフカム HD3000

構成

この機能はデフォルトで有効になっています。これを使用するには、次の検証と構成を完了してください。

ヒント:

Citrix Workspace app のユーザーは、Desktop Viewer の **マイクとWebカメラ** 設定で **接続しない** を選択することにより、デフォルト設定を上書きできます。



1. VDAのインストールが完了したら、VDAがDelivery Controller™に登録でき、公開されたLinuxデスクトップセッションがWindows資格情報を使用して正常に起動できることを確認します。
2. VDAにインターネットアクセスがあることを確認してから、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にインターネットアクセスがない場合は、手順3に進みます。

   > **注:**
   >
   > `uname -r`とカーネルヘッダーの間でカーネルの不一致が発生する可能性があります。この不一致により、ctxwcamcfg.shスクリプトが失敗します。HDX Webカメラビデオ圧縮を適切に使用するには、**sudo apt-get dist-upgrade**を実行し、VDAを再起動してから、ctxwcamcfg.shスクリプトを再実行します。

   VDAがDebianに展開されている場合は、最新のカーネルバージョンで実行されていることを確認してください。そうでない場合は、次のコマンドを実行して最新のカーネルバージョンに更新してください。

   ```
   sudo apt-get update

   sudo apt-get dist-upgrade

   sudo reboot
   ```

   VDAがSUSE 15.3、SUSE 15.2、またはSUSE 12.5に展開されている場合は、次のコマンドを実行して最新のカーネルバージョンに更新し、再起動してください。

   ```
   zypper up kernel-default

   reboot
   ```

   ctxwcamcfg.shスクリプトは、次のことに役立ちます。

   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。
>
> - Enterprise Linux 用追加パッケージ (EPEL)
> - RPM フュージョン

b. `<https://github.com/umlaeute/v4l2loopback>`から`v4l2loopback`のオープンソースコードをダウンロードし、DKMSを使用して`v4l2loopback`を管理します。

`v4l2loopback`は、V4L2ループバックデバイスを作成できるカーネルモジュールです。

c. `sudo service ctxwcamsd restart`コマンドを実行します。Linux VDAの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、VDAにコピーします。

a. インターネットアクセスがあり、VDAと同じカーネルバージョンを持つビルドマシンを準備します。`uname -r`コマンドはカーネルバージョンを見つけるのに役立ちます。

b. ビルドマシンで、`sudo mkdir -p /var/xdl`コマンドを実行します。

c. VDAから`/var/xdl/configure_*`をコピーし、ビルドマシンの`/var/xdl/`の下に配置します。

d. ビルドマシンで、`sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh`コマンドを実行してカーネルモジュールをビルドします。コマンドが正常に実行されると、`/var/lib/dkms/v4l2loopback/1.81b8df79107d1fbf392fdcbaa051bd227a9c94c1/$(uname -r)/x86_64/module/`パスの下にv4l2loopback.koファイルが作成されます。ctxwcamcfg.shスクリプトの実行時に発生する可能性のあるエラーは無視してください。

e. v4l2loopback.koをビルドマシンからVDAにコピーし、`/opt/Citrix/VDA/lib64/`の下に配置します。

f. VDAで、`sudo service ctxwcamsd restart`コマンドを実行して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がなくなります。ドメインに参加していないVDAを作成するときは、VDAをクラウドコントロールプレーンに登録するための公開鍵と秘密鍵のペアを生成します。したがって、Active Directoryドメインへの参加は不要になります。ユーザーがドメインに参加していないVDAからセッションを起動すると、VDAは、ユーザーがCitrix Workspaceアプリにログオンするために使用するユーザー名を使用して、ローカルマッピングアカウントを作成します。VDAは、ローカルマッピングアカウントがSSOとセッション再接続に使用するランダムなパスワードを割り当てます。ランダムなパスワードを変更すると、SSOとセッション再接続は失敗します。SSOを無効にするには、「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。
  - Rendezvous V2を構成してCitrix Cloud Connectorをバイパスできます。そうしないと、VDAをコントロールプレーンに接続するためにCloud Connectorをインストールする必要があります。
- ドメインに参加していないVDAを作成するには、Machine Creation Services™ (MCS)を使用する必要があります。
  - MCSはベアメタルサーバーをサポートしていません。

ドメインに参加していないLinux VDAで利用可能な機能

ドメインに参加していないVDAで指定された属性を持つローカルユーザーを作成する

ドメインに参加していないVDAでホストされているセッションを開くと、VDAはデフォルト属性を持つローカルユーザーを自動的に作成します。VDAは、Citrix Workspace™アプリにログオンするために使用したユーザー名に基づいてローカルユーザーを作成します。ユーザーのユーザー識別子（UID）、グループID（GID）、ホームディレクトリ、ログインシェルなどのユーザー属性を指定することもできます。この機能を使用するには、次の手順を実行します。

1. この機能を有効にするには、次のコマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg create -k
   "HKLM\Software\Citrix\VirtualDesktopAgent\LocalMappedAccount" -t
   "REG_DWORD" -v "CreateWithUidGid" -d "0x00000001" --force
   ```

2. VDAのインストールパスの下にある`/var/xdl/getuidgid.sh`スクリプトで、次の属性を指定します。

属性	必須またはオプション	説明
`uid`	必須	ユーザー識別子 (UID) は、Linux がシステム上の各ユーザーに割り当てる番号です。これにより、ユーザー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。詳しくは、「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。詳しくは、「Azure Active Directory を使用した認証」を参照してください。

ランデブー V2

ドメイン非参加型 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。詳しくは、「Rendezvous V2」を参照してください。

ドメイン非参加型 Linux VDA の作成

MCSを使用して、Citrix DaaSでドメインに参加していないLinux VDAを作成します。詳しくは、「ドメインに参加していないLinux VDAの作成」を参照してください。

ポリシーサポートリスト

Linux VDA ポリシーサポートリスト

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ク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)
クリップボードのセッションからクライアント	LimitClipboardTransferH2C	ユーザー	ICA	無効 (0)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
への転送サイズを制限				
クライアントのローカル時間を使用	UseLocalTimeOfClient	ユーザー	ICA\タイムゾーン制御	サーバーのタイムゾーンを使用する
ICAラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICAラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15
アイドル接続のICAラウンドトリップ計算	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザー監視	無効 (0)
セッション全体の帯域幅制限	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクト帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクト帯域幅制限の割合	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクト帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクト帯域幅の割合	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
クリップボードリダイレクト帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クリップボードリダイレクト帯域幅制限の割合	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0
ファイルリダイレクトの帯域幅制限	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクトの帯域幅制限率	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの帯域幅制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの帯域幅制限率	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0
WebSockets接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocketsポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008
WebSockets信頼済みオリジンサーバーリスト	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICAキープアライブ	SendICAKeepAlives	コンピューター	**ICAキープアライブ**	ICAキープアライブメッセージを送信しない (0)
ICAキープアライブタイムアウト	ICAKeepAliveTimeout	コンピューター	**ICAキープアライブ**	60秒

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ICAリスナーポート番号	IcaListenerPortNumber	コンピューター	ICA	1494
HDX™アダプティブトランスポート	HDXoverUDP	コンピューター	ICA	優先(2)
セッションの信頼性接続	AcceptSessionReliabilityConnections	コンピューター	ICA\セッションの信頼性	許可(1)
再接続UIの透過レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\自動クライアント再接続	80%
セッションの信頼性ポート番号	SessionReliabilityPort	コンピューター	ICA\セッションの信頼性	2598
セッションの信頼性タイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッションの信頼性	180 秒
クライアント自動再接続	AllowAutoClientReconnect	ユーザー	ICA\クライアント自動再接続	許可 (1)
クライアントオーディオのリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターのリダイレクト	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`

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
				`Microsoft OneNote *, Deny"`
クライアントクリップボードのリダイレクト	AllowClipboardRedir	ユーザー	クリップボード	許可 (1)
クライアントUSBデバイスのリダイレクト	AllowUSBRedir	ユーザー	USB	禁止 (0)
クライアントUSBデバイスのリダイレクト規則	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する	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	AllowVisuallyLosslessCompression	ユーザー	シンワイヤー	無効 (0)
3Dグラフィックワークロード向けに最適化	OptimizeFor3dWorkload	ユーザー	シンワイヤー	無効 (0)
シンプルなグラフィックの優先色深度	PreferredColorDepth	ユーザー	シンワイヤー	24ビット/ピクセル (1)
オーディオ品質	SoundQuality	ユーザー	オーディオ	高 – 高精細オーディオ (2)
クライアントマイクのリダイレクト	AllowMicrophoneRedir	ユーザー	オーディオ	許可 (1)
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	バーチャル デリバリー エージェント 設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3
最大Speex品質	MaxSpeexQuality	ユーザー	オーディオ	5

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントドライブの自動接続	AutoConnectDrives	ユーザー	ファイルリダイレクト/CDM	有効 (1)
クライアント光学ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントフロッピーディスクドライブ	AllowFloppyDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントネットワークドライブ	AllowNetworkDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントドライブリダイレクト	AllowDriveRedir	ユーザー	ファイルリダイレクト/CDM	許可 (1)
読み取り専用クライアントドライブアクセス	ReadOnlyMappedDrive	ユーザー	ファイルリダイレクト/CDM	無効 (0)
キーボードの自動表示	AllowAutoKeyboardPopUp	ユーザー	MRVC	無効 (0)
デスクトップとクライアント間のファイル転送を許可	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロード	AllowFileDownload	ユーザー	ファイル転送	許可

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
デスクトップにファイルをアップロード	AllowFileUpload	ユーザー	ファイル転送	許可
セッションアイドルタイマー	EnableSessionIdleTimer	ユーザー	セッションタイマー	有効 (1)
セッションアイドルタイマー間隔	SessionIdleTimerInterval	ユーザー	セッションタイマー	1440分
切断されたセッションタイマー	EnableSessionDisconnectTimer	ユーザー	セッションタイマー	無効 (0)
切断されたセッションタイマー間隔	SessionDisconnectTimerPeriod	ユーザー	セッションタイマー	1440分

注:

Windows Virtual Delivery Agent (VDA) のみが、User Datagram Protocol (UDP) 経由のオーディオをサポートしています。Linux VDA はサポートしていません。詳細については、「User Datagram Protocol (UDP) 経由のオーディオリアルタイムトランスポート」を参照してください。

Citrix Studioでセッション接続タイマーを構成するには、以下のCitrixポリシー設定を使用できます。

- **セッションアイドルタイマー**: アイドル状態のセッションに時間制限を適用するかどうかを決定します。
- **セッションアイドルタイマー間隔**: アイドル状態のセッションに時間制限を設定します。**セッションアイドルタイマー**が**有効**で、設定された時間内にアクティブなセッションがユーザー入力を受信しなかった場合、セッションは切断されます。
- **切断されたセッションタイマー**: 切断されたセッションに時間制限を適用するかどうかを決定します。

- **切断されたセッションタイマー間隔**: 切断されたセッションがログオフされるまでの間隔を設定します。

いずれかのポリシー設定を更新する際は、展開全体でそれらが一貫していることを確認してください。

アイドルセッションの時間制限が期限切れになると、警告メッセージが表示されます。例については、次のスクリーンショットを参照してください。**OK**を押すと警告メッセージは閉じますが、セッションをアクティブに保つことはできません。セッションをアクティブに保つには、ユーザー入力を提供してアイドルタイマーをリセットしてください。

警告メッセージ(/en-us/linux-virtual-delivery-agent/2301/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):

  既定値: 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できます。Speexのエンコードプロセスは、ほとんどの場合、0から10までの範囲の品質パラメーターによって制御されます。品質が高いほど、ビットレートも高くなります。

  最大Speex品質は、オーディオ品質と帯域幅制限（ポリシー「オーディオリダイレクト帯域幅制限」を参照）に基づいて、オーディオデータをエンコードするための最適なSpeex品質を選択します。オーディオ品質が中の場合、エンコーダーは広帯域モードになり、サンプリングレートが高くなります。オーディオ品質が低い場合、エンコーダーは狭帯域モードになり、サンプリングレートが低くなります。同じSpeex品質でも、モードによってビットレートが異なります。最適なSpeex品質とは、最大値が次の条件を満たす場合です。

  - 最大Speex品質以下であること。
  - そのビットレートが帯域幅制限以下であること。

  関連設定: オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

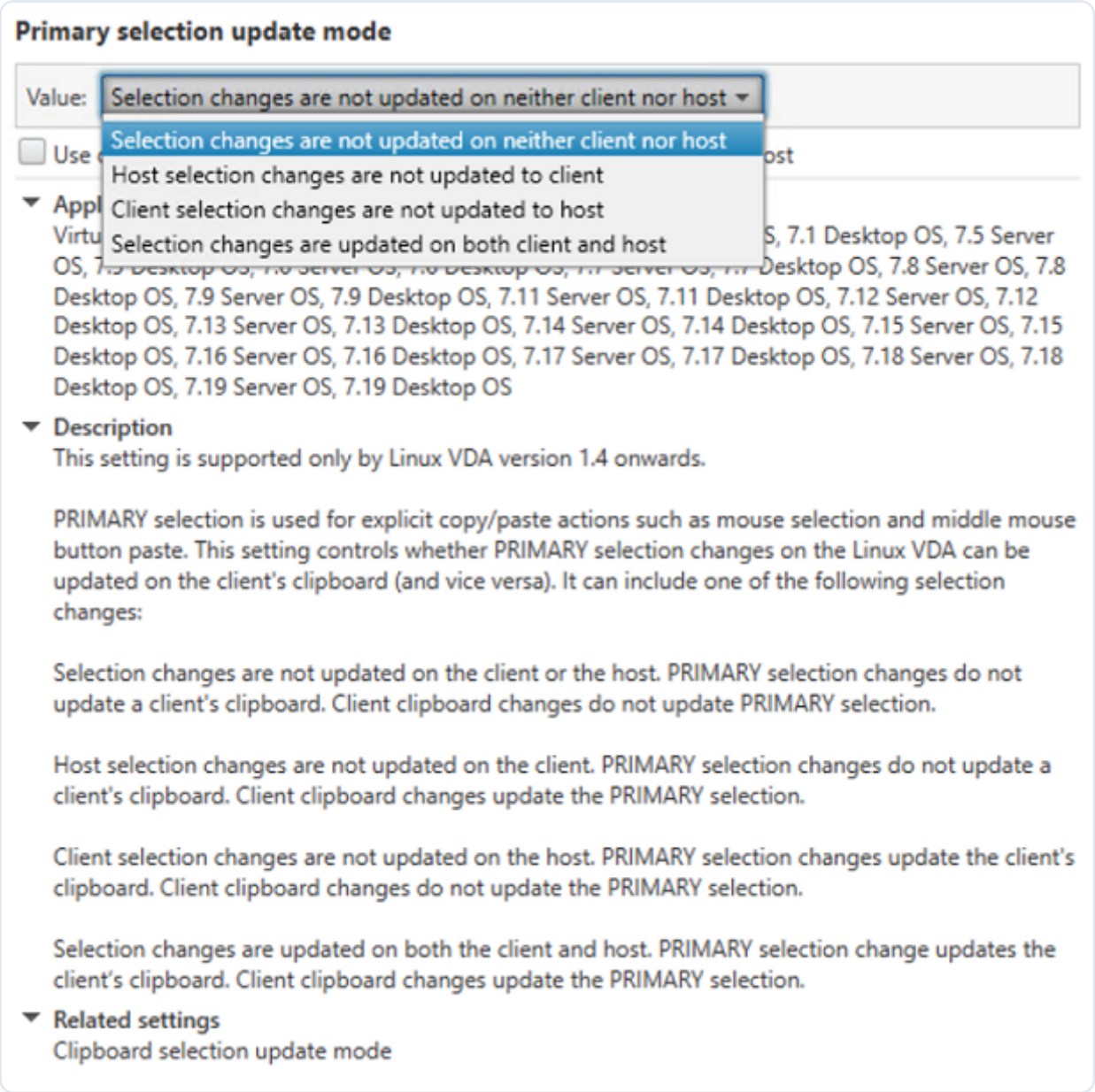
  値 (列挙型):

  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。4つの値オプションがあります。



    - **クライアントとホストのどちらでも選択の変更は更新されません**

Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新しません。

- **ホストの選択変更はクライアントに更新されない**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新します。

- **クライアントの選択変更はホストに更新されない**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新しません。

- **クライアントとホストの両方で選択変更が更新される**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新します。このオプションがデフォルト値です。

関連設定: クリップボード選択更新モード

- ClipboardSelectionUpdateMode

  値 (enum):

  デフォルト値: 3

  詳細:

  クリップボード選択は、データをいくつか選択し、ショートカットメニューから「コピー」を選択するなどして、明示的にクリップボードに「コピー」するよう要求する場合に使用されます。クリップボード選択は主にMicrosoft Windowsのクリップボード操作に関連して使用されますが、プライマリ選択はLinuxに固有のものです。

  このポリシーは、Linux VDAとクライアントでのクリップボード選択の変更が相互にクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。

  クリップボード選択更新モードの画像(/en-us/linux-virtual-delivery-agent/7-15-ltsr/media/clipboard-selection-update-mode.png)

  - **クライアントとホストのどちらにも選択変更が更新されない**

Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **ホストの選択変更はクライアントに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。

- **クライアントの選択変更はホストに更新されない**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **クライアントとホストの両方で選択変更が更新される**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

注:

Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピー＆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(/ja-jp/linux-virtual-delivery-agent/2301/configure/printing/printing-best-practice.html)
- PDF印刷(/ja-jp/linux-virtual-delivery-agent/2301/configure/printing/pdf-printing.html)

印刷のベストプラクティス

この記事では、印刷のベストプラクティスに関する情報を提供します。

インストール

Linux VDAには、**cups**フィルターと**foomatic**フィルターの両方が必要です。これらのフィルターは、VDAのインストール時にインストールされます。ディストリビューションに基づいて、フィルターを手動でインストールすることもできます。例：

RHEL 7の場合：

```
sudo yum –y install cups



sudo yum -y install foomatic-filters
```

印刷ポリシー設定

クライアントプリンターリダイレクト

この設定は、クライアントプリンターがVDAセッションにマップされるかどうかを制御します。デフォルトでは、クライアントプリンターのマッピングは許可されています。

クライアントプリンターを自動作成

この設定は、VDAセッションにマップできるクライアントプリンターを指定します。デフォルトでは、**すべてのクライアントプリンターを自動作成**に設定されており、これはすべてのクライアントプリンターがVDAセッションにマップされることを意味します。この設定の詳細については、Citrix Virtual Apps and Desktopsのドキュメントのクライアントプリンターの自動作成を参照してください。

PDFユニバーサルプリンターの自動作成

PDF印刷機能を使用するには、このポリシーを**有効**に設定します。

プリンタードライバーのマッピングと互換性

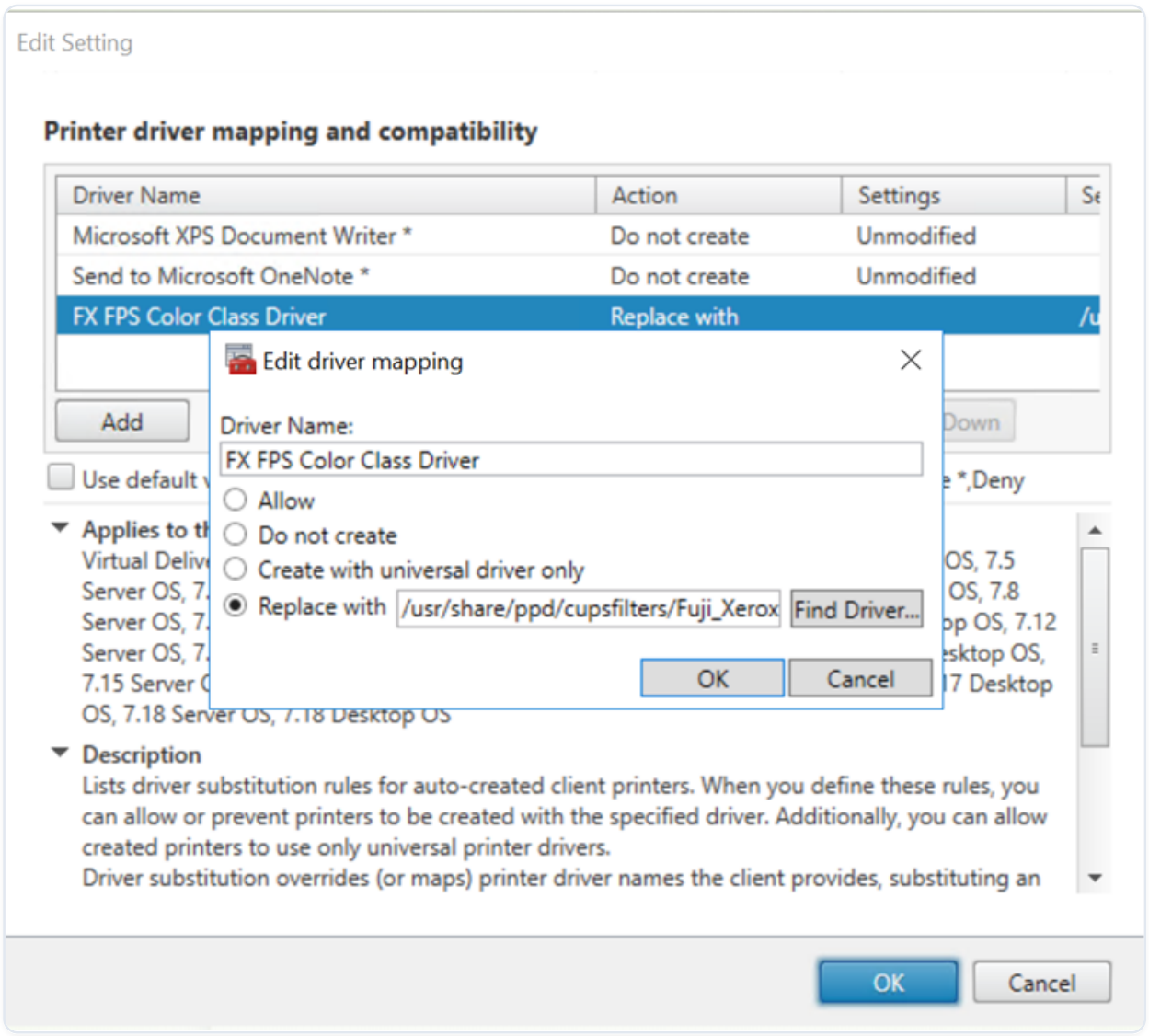
Citrixは、3種類のユニバーサルプリンタードライバー（postscript、pcl5、pcl6）を提供しています。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合があります。この場合、以前のリリースでは、~/.CtxlpProfile$CLIENT_NAME構成ファイルを編集するしかありませんでした。バージョン1906以降では、代わりにCitrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成することを選択できます。

Citrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成するには：

1. **プリンタードライバーのマッピングと互換性**ポリシーを選択します。

2. 「**追加**」をクリックします。
3. **ドライバー名**にクライアントプリンターのドライバー名を入力します。Citrix Workspace™ app for Linuxを使用している場合は、代わりにプリンター名を入力します。

   4\. **次で置換**を選択し、VDA上のドライバーファイルの絶対パスを入力します。



注:

- PPDドライバーファイルのみがサポートされています。
- **プリンタードライバーのマッピングと互換性**ポリシーのその他のオプションはサポートされていません。**次で置き換える**のみが有効になります。

使用方法

公開デスクトップと公開アプリケーションの両方から印刷できます。すべてのクライアントプリンターをVDAセッションにマッピングできます。プリンター名はデスクトップとアプリケーションで異なります。

- 公開デスクトップの場合:

  `<client printer name>:$CLIENT_NAME:dsk$SESSION_ID`

- 公開アプリケーションの場合:

  `<client printer name>:$CLIENT_NAME:app$SESSION_ID`

> **注:**
>
> 同じユーザーが公開デスクトップと公開アプリケーションの両方を開いた場合、両方のプリンターがセッションで利用可能になります。公開アプリケーションセッションでデスクトッププリンターに印刷したり、公開デスクトップでアプリケーションプリンターに印刷したりすることはできません。

トラブルシューティング

印刷できない

印刷が正しく機能しない場合は、印刷デーモン**ctxlpmngt**とCUPSフレームワークを確認してください。

印刷デーモン**ctxlpmngt**はセッションごとのプロセスであり、セッションの期間中実行されている必要があります。次のコマンドを実行して、印刷デーモンが実行されていることを確認します。**ctxlpmngt**が実行されていない場合は、コマンドラインから手動で**ctxlpmngt**を開始します。

```
ps –ef | grep ctxlpmngt
```

印刷がまだ機能しない場合は、**CUPS**フレームワークを確認してください。**ctxcups**サービスはプリンター管理に使用され、Linux CUPSフレームワークと通信します。これはマシンごとに1つのプロセスであり、次のコマンドを実行して確認できます。

```
service ctxcups status
```

CUPSログを収集するための追加手順

CUPSログを収集するには、次のコマンドを実行してCUPSサービスファイルを構成します。そうしないと、CUPSログは**hdx.log**に記録されません。

```
sudo service cups stop


sudo vi /etc/systemd/system/printer.target.wants/cups.service


PrivateTmp=false


sudo service cups start


sudo systemctl daemon-reload
```

注:

この構成は、問題が発生したときに完全な印刷ログを収集するためだけに行われます。通常の状況では、CUPSのセキュリティを破るため、この構成は推奨されません。

印刷出力が文字化けする

互換性のないプリンタードライバーは、出力の文字化けを引き起こす可能性があります。ユーザーごとのドライバー構成が利用可能で、**~/.CtxlpProfile$CLIENT_NAME**構成ファイルを編集することで構成できます。

```
[DEFAULT_PRINTER]


printername=
```

```
model=


ppdpath=


drivertype=
```

重要:

「**printername**」は、現在のクライアント側のデフォルトプリンターの名前を含むフィールドです。これは読み取り専用の値です。編集しないでください。

ppdpath、**model**、**drivertype** の各フィールドは、マップされたプリンターに対して1つしか有効にならないため、同時に設定することはできません。

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、**model**= オプションを使用してネイティブプリンタードライバーのモデルを構成します。プリンターの現在のモデル名は、**lpinfo** コマンドを使用して見つけることができます。

  ```
  lpinfo –m


  …


  xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0


  xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

  xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
  ```

```
```

その後、プリンターに合わせてモデルを設定できます。

```
model=xerox/ph3115.ppd.gz
```

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、ネイティブプリンタードライバーの PPD ファイルパスを構成します。**ppdpath** の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  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
  ```

出力サイズがゼロになる

異なる種類のプリンターを試してください。また、CutePDF や PDFCreator のような仮想プリンターを試して、この問題がプリンタードライバーに関連しているかどうかを確認してください。

印刷ジョブは、クライアントのデフォルトプリンターのプリンタードライバーに依存します。現在アクティブなドライバーの種類を特定することが重要です。クライアントプリンターが PCL5 ドライバーを使用しているのに、Linux VDA が Postscript ドライバーを選択すると、問題が発生する可能性があります。

プリンタードライバーの種類が正しい場合は、次の手順を実行して問題を特定できます。

1. 公開デスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME** コマンドを実行します。
3. Linux VDA にスプールファイルを保存するために、次のフィールドを追加します。

   ```
   deletespoolfile=no
   ```

4. 構成の変更を読み込むには、ログオフして再度ログオンします。
5. 問題を再現するためにドキュメントを印刷します。印刷後、スプールファイルは **/var/spool/cups-ctx/$logon_user/$spool_file** に保存されます。
6. スプールが空かどうかを確認します。スプールファイルがゼロの場合、それは問題を示しています。詳細なガイダンスについては、Citrixサポートに連絡してください（印刷ログを提供してください）。
7. スプールサイズがゼロでない場合は、ファイルをクライアントにコピーします。スプールファイルの内容は、クライアントのデフォルトプリンターのプリンタードライバーの種類によって異なります。マップされたプリンター（ネイティブ）ドライバーがPostScriptの場合、スプールファイルはLinux OSで直接開くことができます。内容が正しいか確認してください。

   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。以下の例では、PostScriptクライアントプリンターを例として使用します。

   a. アクティブなセッションにログオンし、クライアントデスクトップでブラウザーを開きます。
   b. 印刷管理ポータルを開きます。

      ```
      localhost:631
      ```

   c. マップされたプリンター **CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID** と **Modify Printer** 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、**Continue** をクリックしてプリンタードライバーを変更します。
   e. **Make** および **Model** フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。
   f. このプロセスが成功した場合、マップされたプリンターが新しく選択されたドライバーを使用できるように、**.CtxlpProfile$CLIENT_NAME** でドライバーのPPDファイルパスを構成します。

既知の問題

Linux VDAでの印刷中に、以下の問題が確認されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブのものに設定してください。

大容量ドキュメントの印刷パフォーマンスの低下

ローカルクライアントプリンターで大容量ドキュメントを印刷すると、ドキュメントはサーバー接続を介して転送されます。低速な接続では、転送に時間がかかる場合があります。

他のセッションから表示されるプリンターおよび印刷ジョブの通知

LinuxにはWindowsオペレーティングシステムと同じセッションの概念がありません。そのため、すべてのユーザーがシステム全体の通知を受け取ります。これらの通知は、CUPS構成ファイルである **/etc/cups/cupsd.conf** を変更することで無効にできます。

ファイルに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、次の行をデフォルトポリシーのXMLブロックに追加します。

```
<Policy default>

     # Job/subscription privacy...

     JobPrivateAccess default

     JobPrivateValues default

     SubscriptionPrivateAccess default

     SubscriptionPrivateValues default
```

```
    … …

    <Limit Create-Printer-Subscription>

        Require user @OWNER

        Order deny,allow

    </Limit>

    <Limit All>

        Order deny,allow

    </Limit>

</Policy>
```

PDF印刷

PDF印刷をサポートするバージョンのCitrix Workspace™アプリを使用すると、Linux VDAセッション内から変換されたPDFを印刷できます。セッションの印刷ジョブは、Citrix Workspaceアプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、選択したPDFビューアを使用してPDFを開き、選択した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。

- **クライアントプリンターリダイレクト**を**許可**に設定します（デフォルトで**許可**）。
- **PDFユニバーサルプリンターの自動作成**を**有効**に設定します（デフォルトで**無効**）。
- **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**印刷**をクリックしたときに、プリンターを選択するための印刷プレビューがローカルマシンに表示されます。デフォルトプリンターの設定については、(/ja-jp/citrix-workspace-app.html)のを参照してください。

リモートPCアクセス

概要

リモートPCアクセスは、Citrix Virtual Apps and Desktopsの拡張機能です。これにより、組織は従業員が物理的なオフィスPCに安全な方法でリモートで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、業務に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

リモートPCアクセスは、仮想デスクトップとアプリケーションを提供するのと同じCitrix Virtual Apps™ and Desktopsコンポーネントを使用します。リモートPCアクセスの展開と構成の要件およびプロセスは、Citrix Virtual Apps and Desktopsの展開に必要な要件およびプロセスと同じです。この統一性により、一貫性のある統合された管理エクスペリエンスが提供されます。ユーザーは、Citrix HDXを使用してリモートオフィスPCセッションを配信することで、最高のユーザーエクスペリエンスを得られます。

詳細については、Citrix Virtual Apps and Desktopsのドキュメントでリモートアクセス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。この画面を表示することは、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、ユーザーがリモートPCアクセスデスクトップセッション内でオフィスPC上のアプリケーションとデータにアクセスするのに役立ちます。
- ユーザーがすでにローカルでPCにログオンしている場合、StoreFront™からPCを起動しようとすると失敗します。
- 省電力オプションはLinuxマシンでは利用できません。

構成

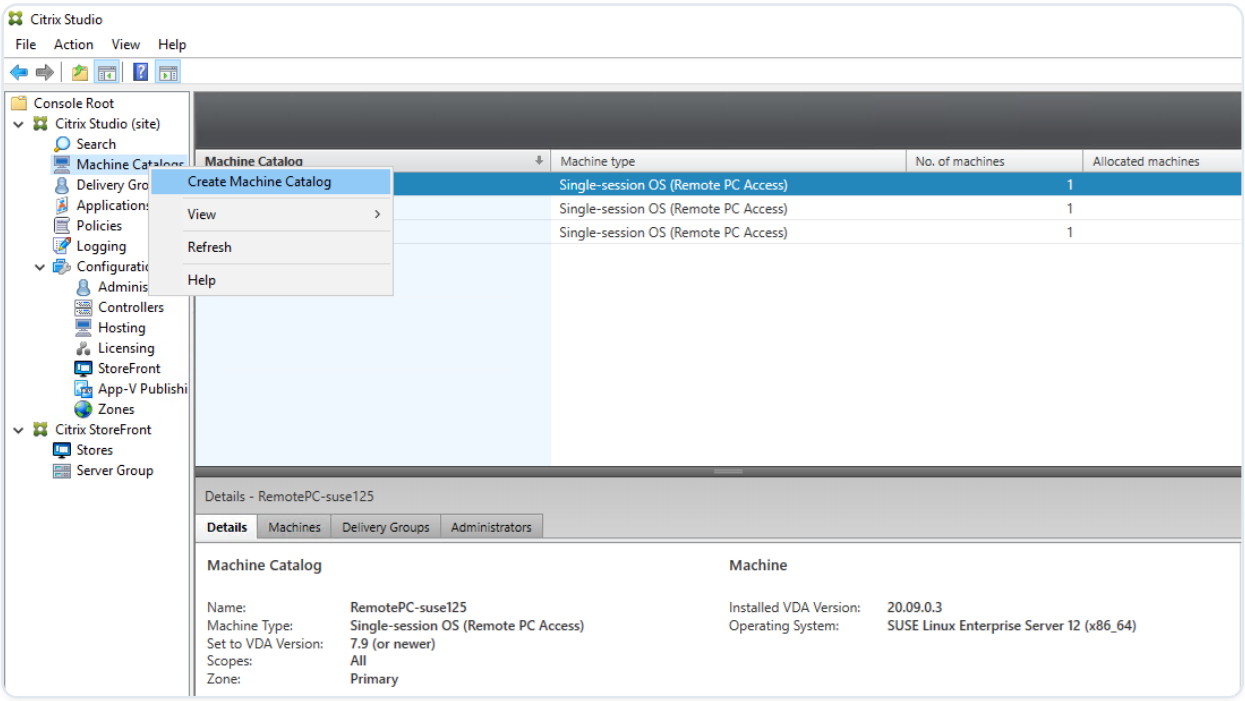
Linux PCセッションを配信するには、ターゲットPCにLinux VDAをインストールし、**リモートPCアクセス**タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成します。次のセクションで手順を詳しく説明します。

ステップ1 - ターゲットPCにLinux VDAをインストールする

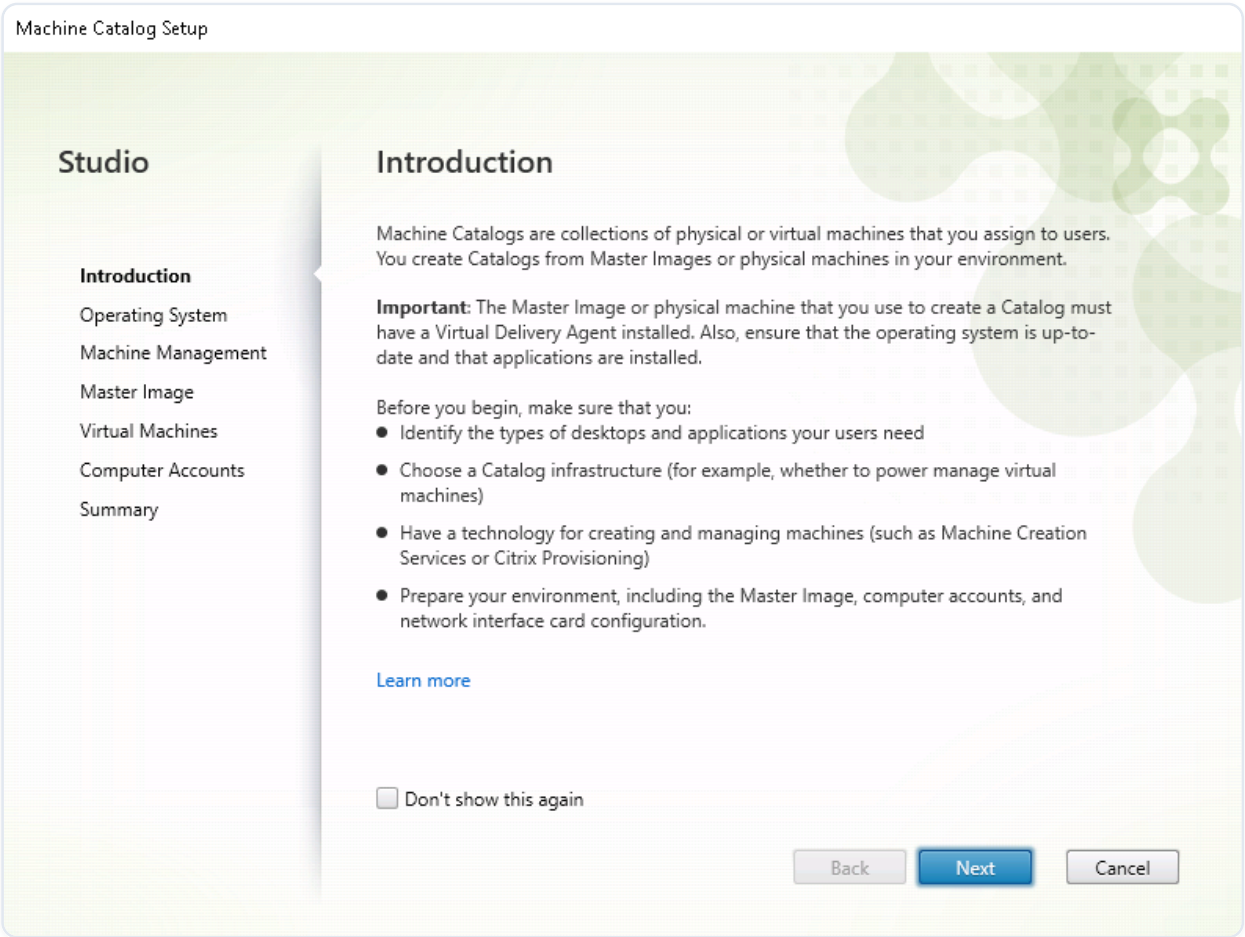
Linux VDAをインストールするには、簡易インストールを使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE`変数の値を`Y`に設定します。

ステップ2 - リモートPCアクセスタイプのマシンカタログを作成する

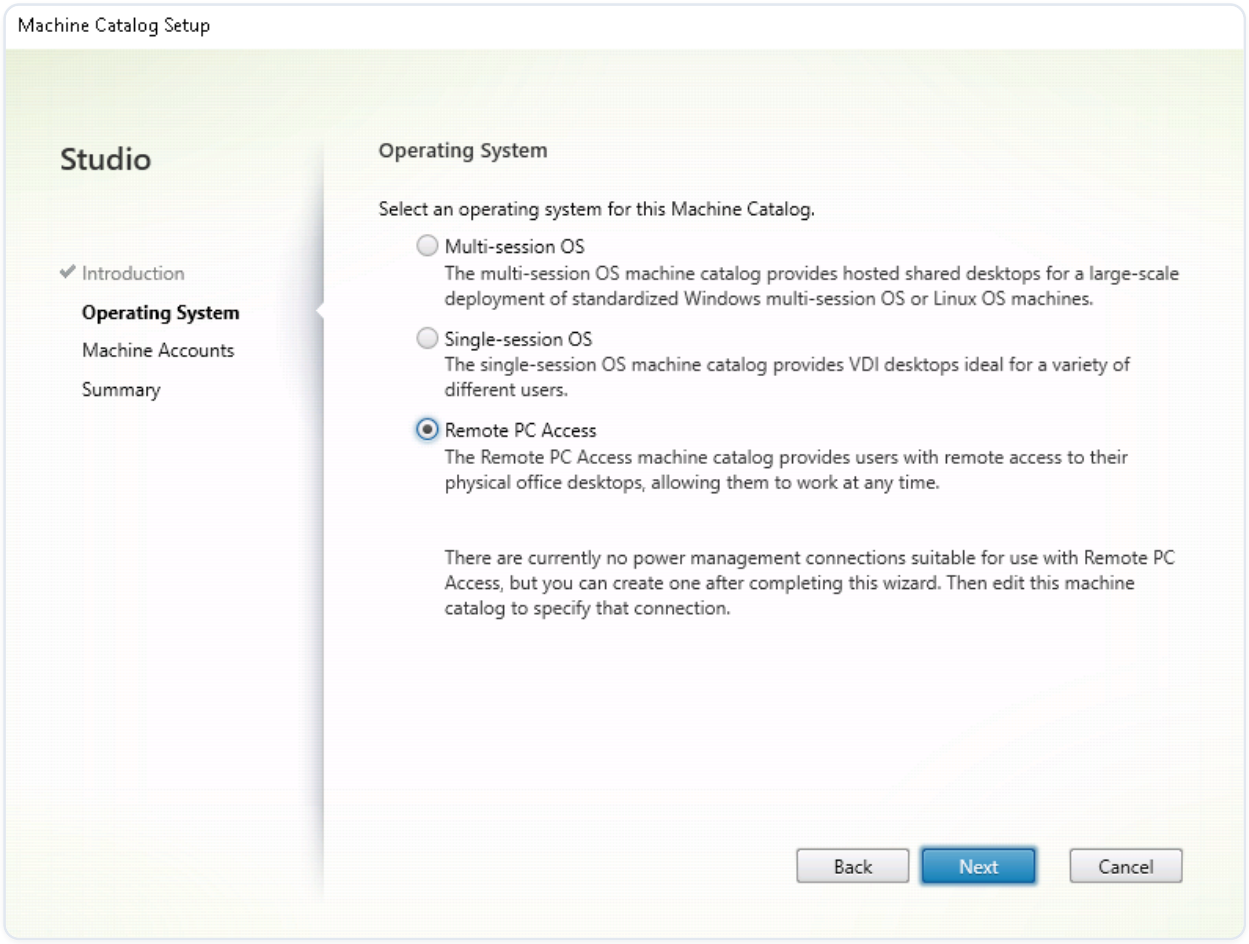
1. Citrix Studioで、**マシンカタログ**を右クリックし、ショートカットメニューから**マシンカタログの作成**を選択します。



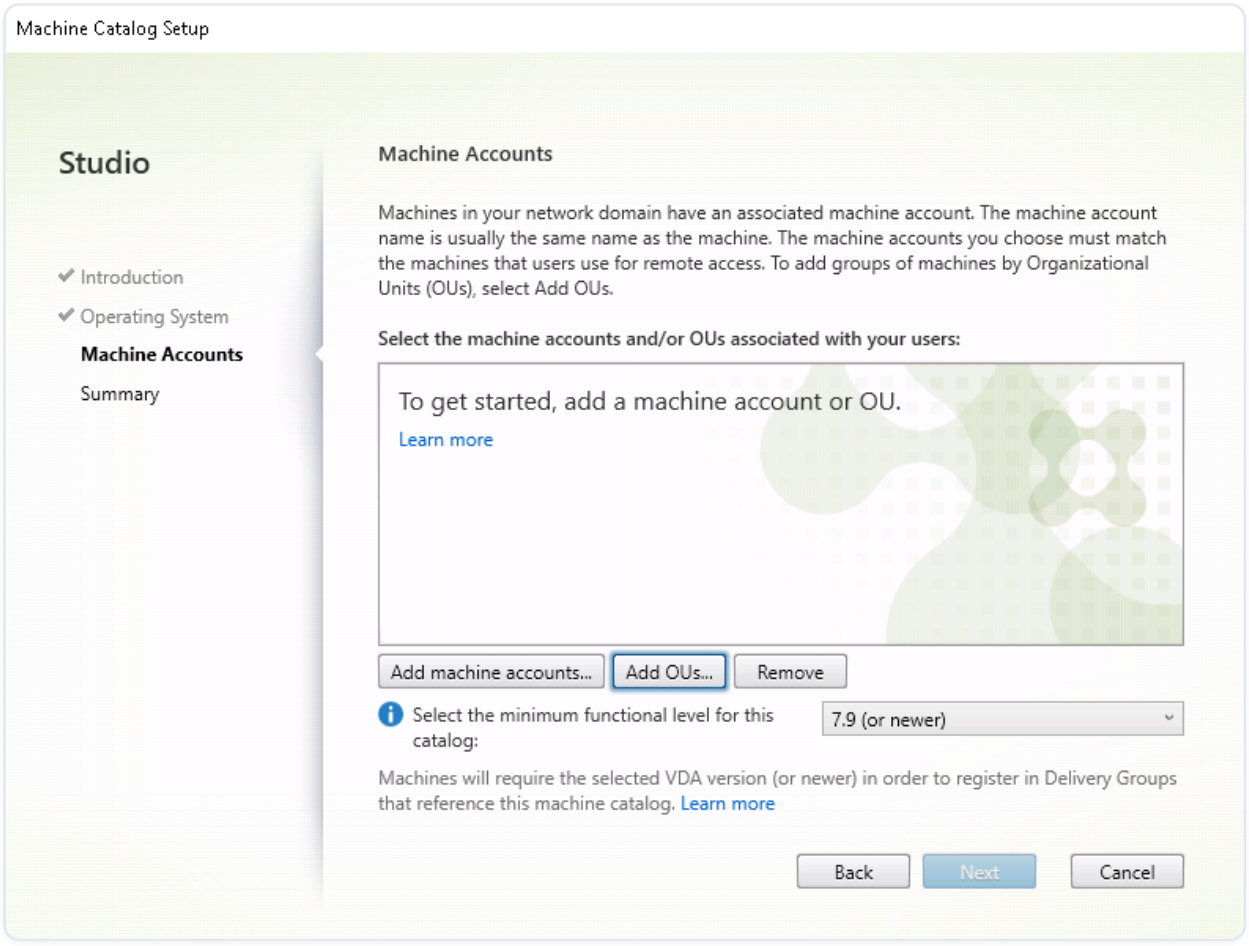
2. **はじめに**ページで**次へ**をクリックします。



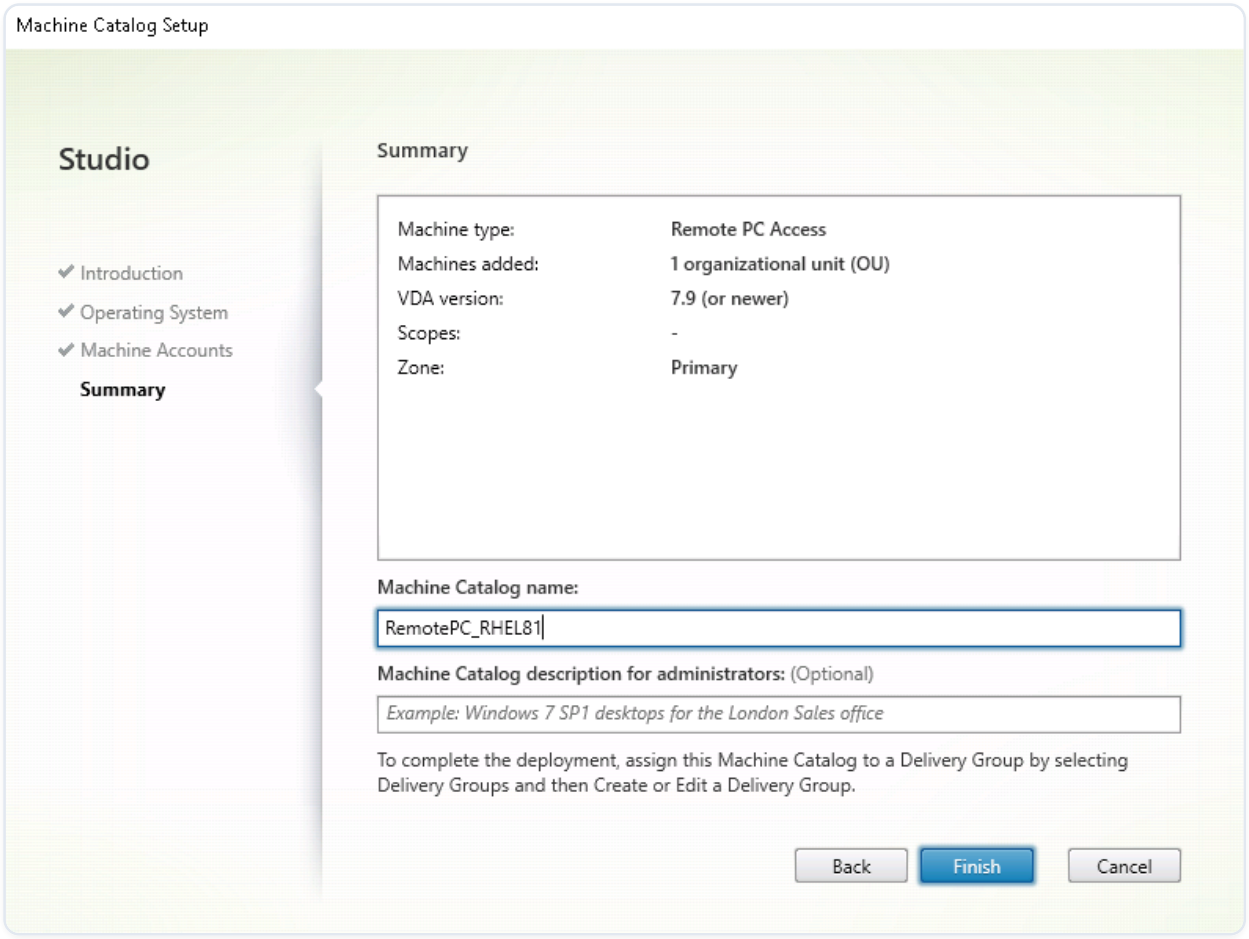
3. **オペレーティングシステム**ページで**リモートPCアクセス**を選択します。



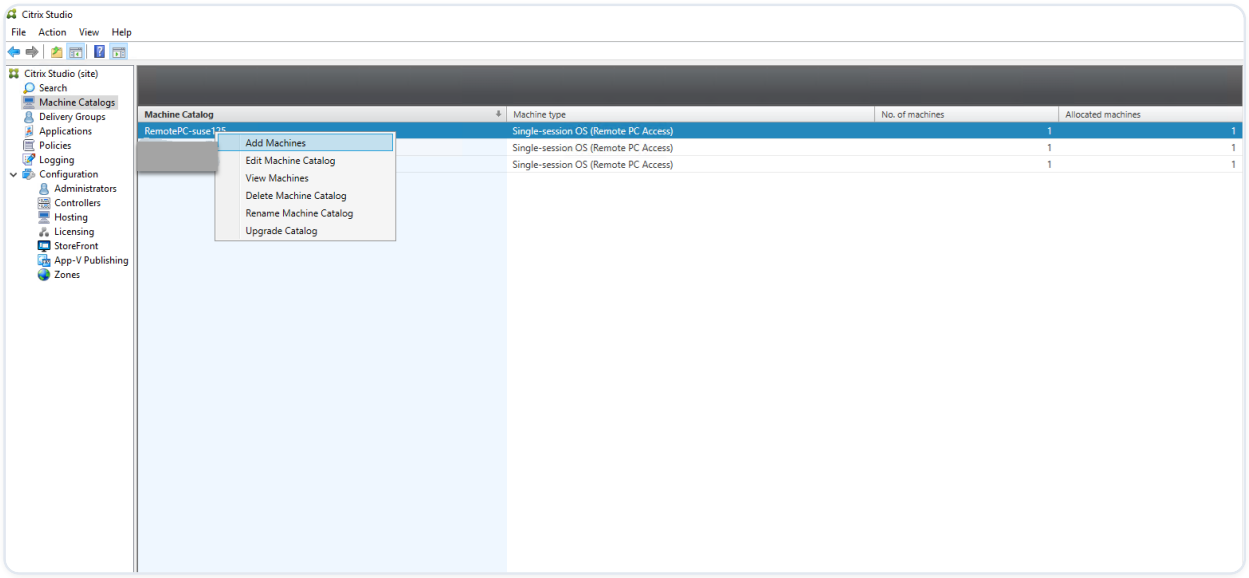
4. 「**OUの追加**」をクリックしてターゲットPCを含むOUを選択するか、「**マシンアカウントの追加**」をクリックして個々のマシンをマシンカタログに追加します。



5. マシンカタログに名前を付けます。

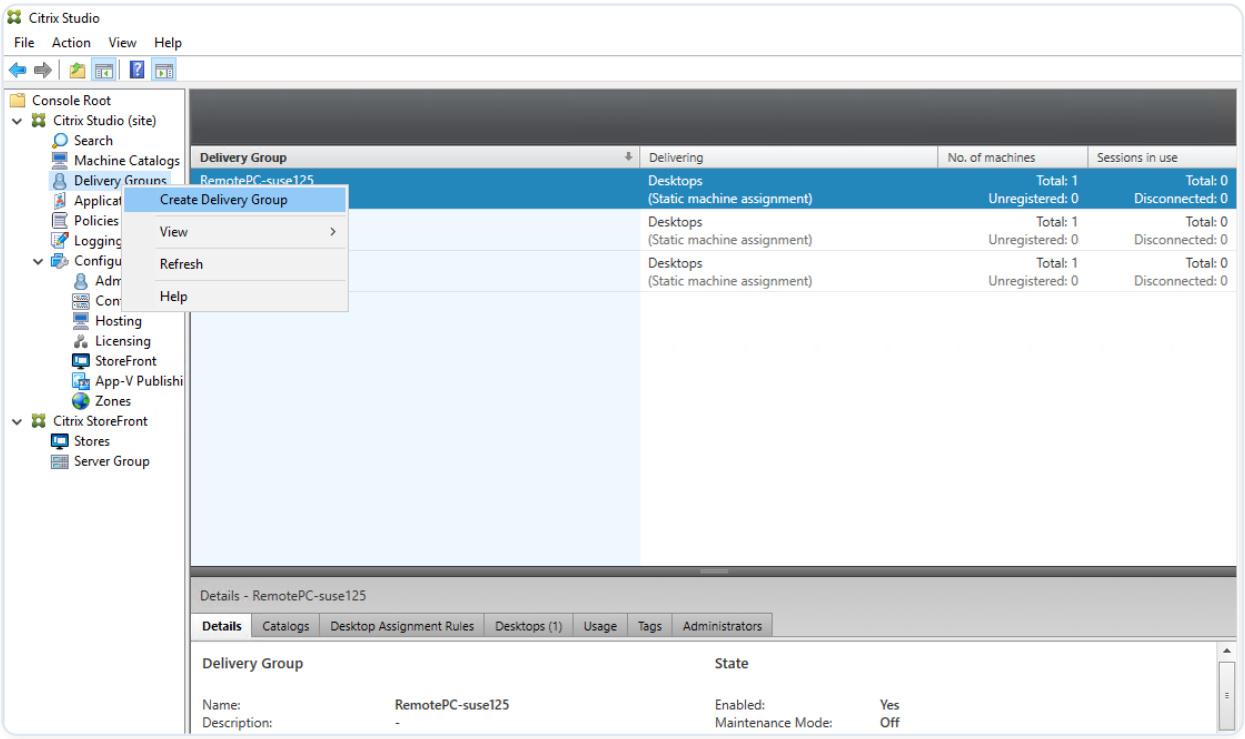


6. （オプション）マシンカタログを右クリックして、関連する操作を実行します。

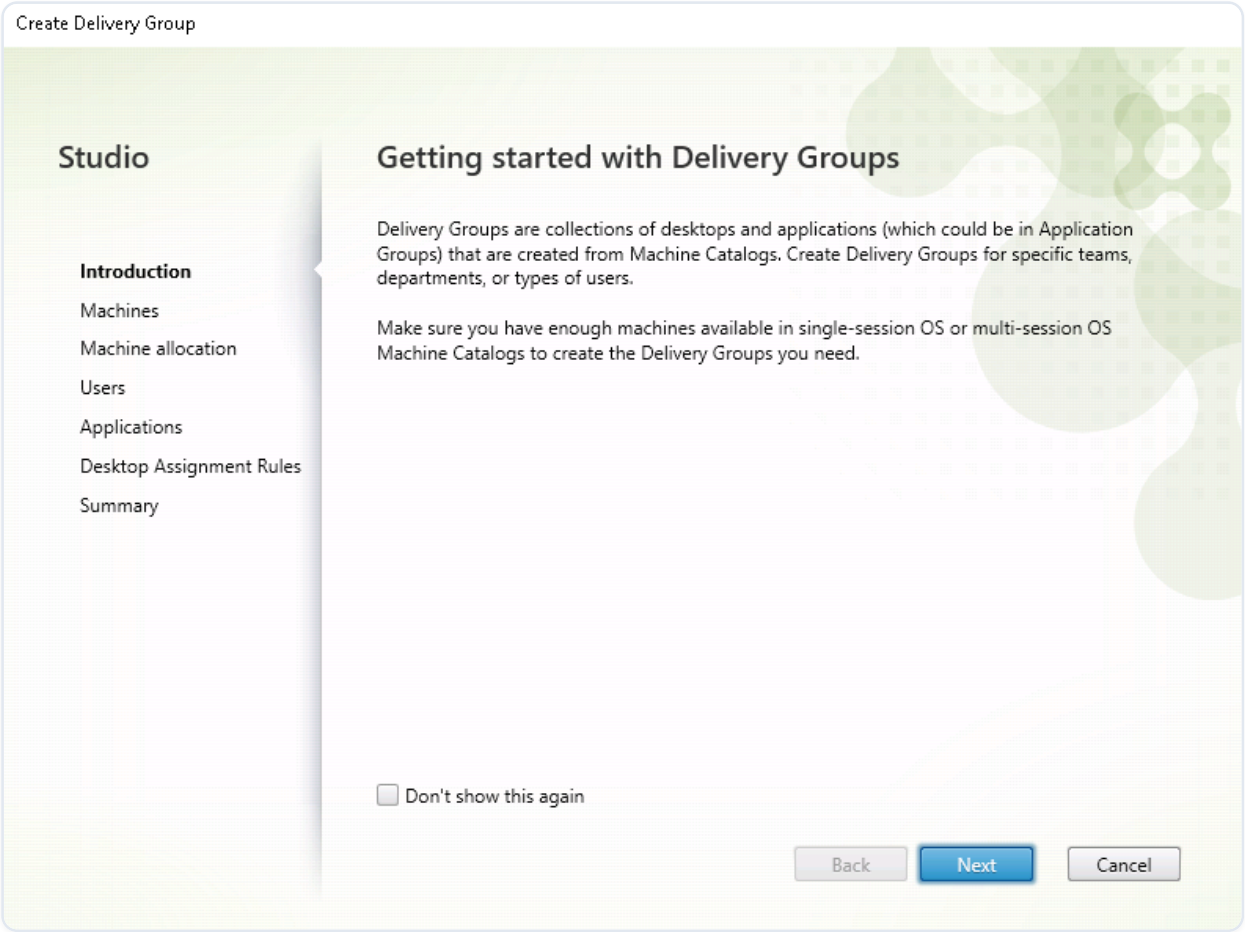


ステップ3 - デリバリーグループを作成して、マシンカタログ内のPCを、アクセスを要求するユーザーが利用できるようにします

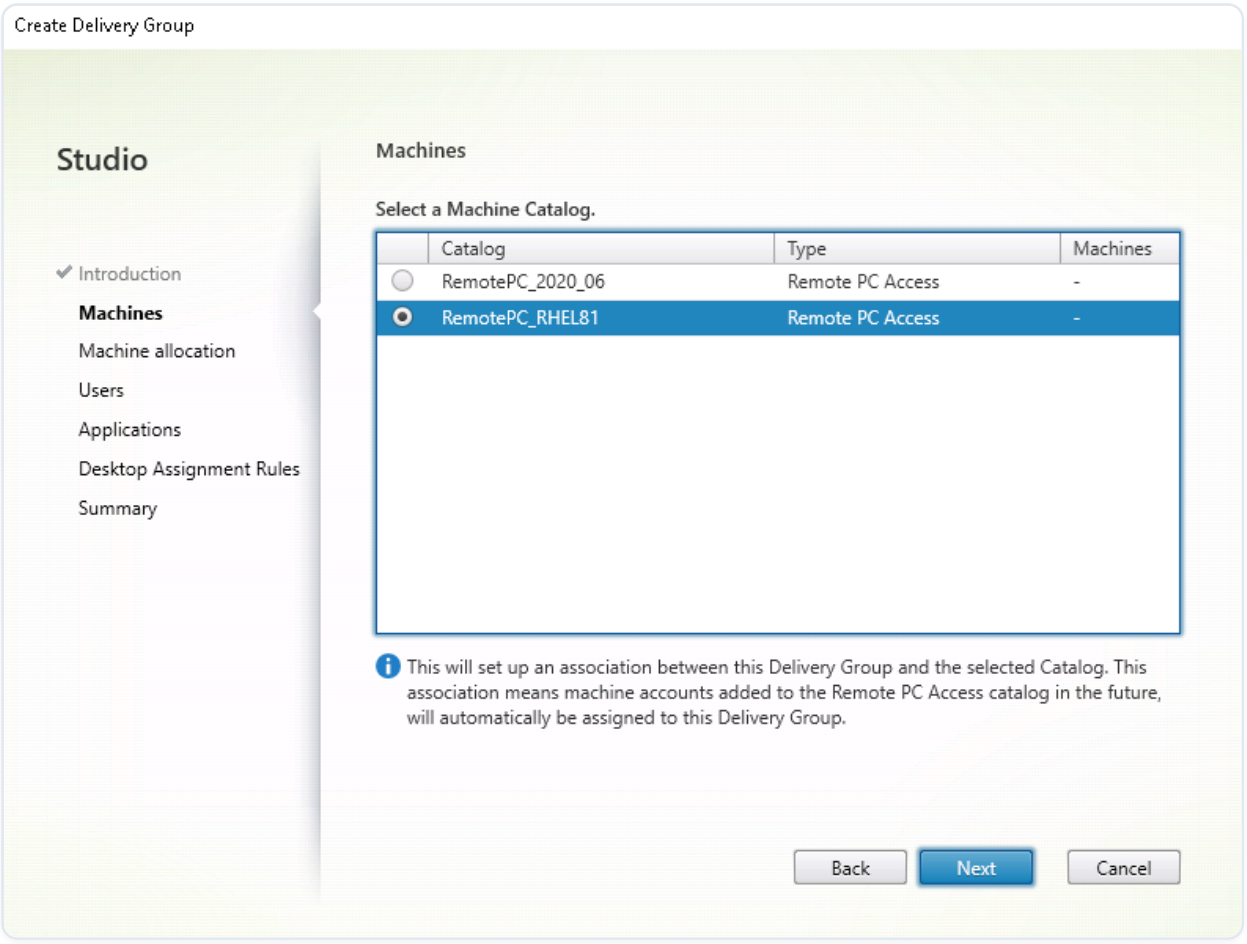
1. Citrix Studioで、「**デリバリーグループ**」を右クリックし、ショートカットメニューから「**デリバリーグループの作成**」を選択します。



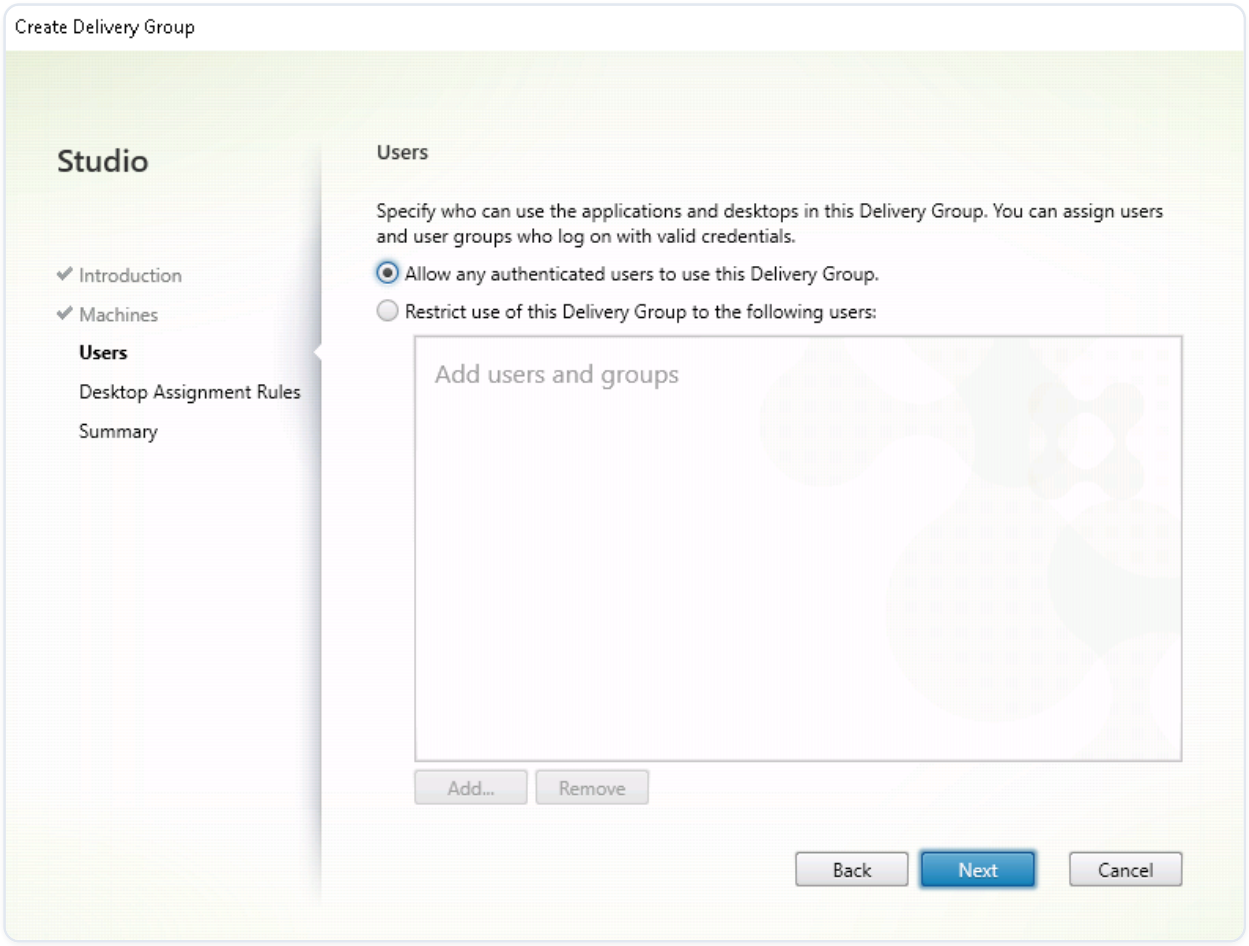
2. 「**デリバリーグループの開始**」ページで「**次へ**」をクリックします。



3. ステップ2で作成したマシンカタログを選択し、デリバリーグループに関連付けます。



4. マシンカタログ内のPCにアクセスできるユーザーを追加します。追加したユーザーは、クライアントデバイス上のCitrix Workspaceアプリを使用して、PCにリモートでアクセスできます。



Wake on LAN (ウェイク・オン・ラン)

Remote PC AccessはWake on LANをサポートしており、ユーザーは物理PCをリモートで起動できます。この機能により、ユーザーは使用していないときにオフィスPCの電源を切ったままにして、エネルギーコストを節約できます。また、誤ってマシンの電源が切れてしまった場合でも、リモートアクセスを可能にします。

Wake on LAN機能を使用すると、デリバリーコントローラーからの指示があった場合、PC上で実行されているVDAから、そのPCが存在するサブネットにマジックパケットが直接送信されます。これにより、追加のインフラストラクチャコンポーネントやマジックパケット配信のためのサードパーティソリューションに依存することなく、この機能が動作します。

Wake on LAN機能は、レガシーなSCCMベースのWake on LAN機能とは異なります。SCCMベースのWake on LANの詳細については、「Wake on LAN – SCCM統合」を参照してください。

システム要件

Wake on LAN機能を使用するためのシステム要件は次のとおりです。

- コントロールプレーン:

  - シトリックス DaaS™ (旧称 シトリックス バーチャル アプリケーションズ アンド デスクトップ サービス)
  - Citrix Virtual Apps and Desktops のバージョン 2012 以降

- 物理PC:
  - VDAバージョン2012以降
  - BIOSおよびNICでWake on LANが有効になっていること

Wake on LANの構成

現在、統合されたWake on LANの構成は、PowerShellを使用してのみサポートされています。

Wake on LANを構成するには:

1. まだ Remote PC Access マシンカタログがない場合は、作成します。
2. まだ Wake on LAN ホスト接続がない場合は、作成します。

> **注:**
>
> Wake on LAN機能を使用するには、「Microsoft Configuration Manager Wake on LAN」タイプのホスト接続がある場合、ホスト接続を作成します。

3. Wake on LAN ホスト接続の一意の識別子を取得します。
4. Wake on LAN ホスト接続をマシンカタログに関連付けます。

   Wake on LAN ホスト接続を作成するには：

```
# Load Citrix SnapIns

Add-PSSnapIn -Name "*citrix*"



# Provide the name of the Wake on LAN host connection

[string]$connectionName = "Remote PC Access Wake on LAN"
```

```
# Create the hypervisor connection

$hypHc = New-Item -Path xdhyp:\Connections `

             -Name $connectionName `

             -HypervisorAddress "N/A" `

             -UserName "woluser" `

             -Password "wolpwd" `

             -ConnectionType Custom `

             -PluginId VdaWOLMachineManagerFactory `

             -CustomProperties "<CustomProperties></CustomProperties>" `

             -Persist



$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid



# Wait for the connection to be ready before trying to use it

while (-not $bhc.IsReady)

{

    Start-Sleep -s 5

    $bhc = Get-BrokerHypervisorConnection -HypHypervisorConnectionUid
$hypHc.HypervisorConnectionUid

}
```

ホスト接続の準備ができたら、次のコマンドを実行してホスト接続の一意の識別子を取得します。

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"

$hypUid = $bhc.Uid
```

接続の一意の識別子を取得したら、次のコマンドを実行して、その接続を Remote PC Access マシンカタログに関連付けます。

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog
-RemotePCHypervisorConnectionUid $hypUid
```

5. マシンカタログ内の各VMで、BIOSおよびNICのWake on LANを有効にします。

   注： Wake on LANを有効にする方法は、マシンの構成によって異なります。

   - BIOSでWake on LANを有効にするには：

     a. BIOSに入り、Wake on LAN機能を有効にします。

        BIOSにアクセスする方法は、マザーボードの製造元と、製造元が選択したBIOSベンダーによって異なります。

     b. 設定を保存して、マシンを再起動します。

   - NICでWake on LANを有効にするには：

     a. `sudo ethtool <NIC>` コマンドを実行して、NICがマジックパケットをサポートしているかどうかを確認します。

        `<NIC>` はNICのデバイス名です（例：`eth0`）。`sudo ethtool <NIC>` コマンドは、NICの機能に関する出力を提供します：

        - 出力に `Supports Wake-on: <letters>` のような行があり、`<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>` のような行があり、`<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は有効になっています。ただし、`<letters>` に文字 `d` が含まれている場合、Wake on LAN機能が無効になっていることを示します。この場合、`sudo ethtool -s <NIC> wol g` コマンドを実行してWake on LANを有効にしてください。

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:

インターフェース設定ファイル`/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:

インターフェース設定ファイル`/etc/sysconfig/network-scripts/ifcfg-<NIC>`に以下の`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で表示されます。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する場合
- 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

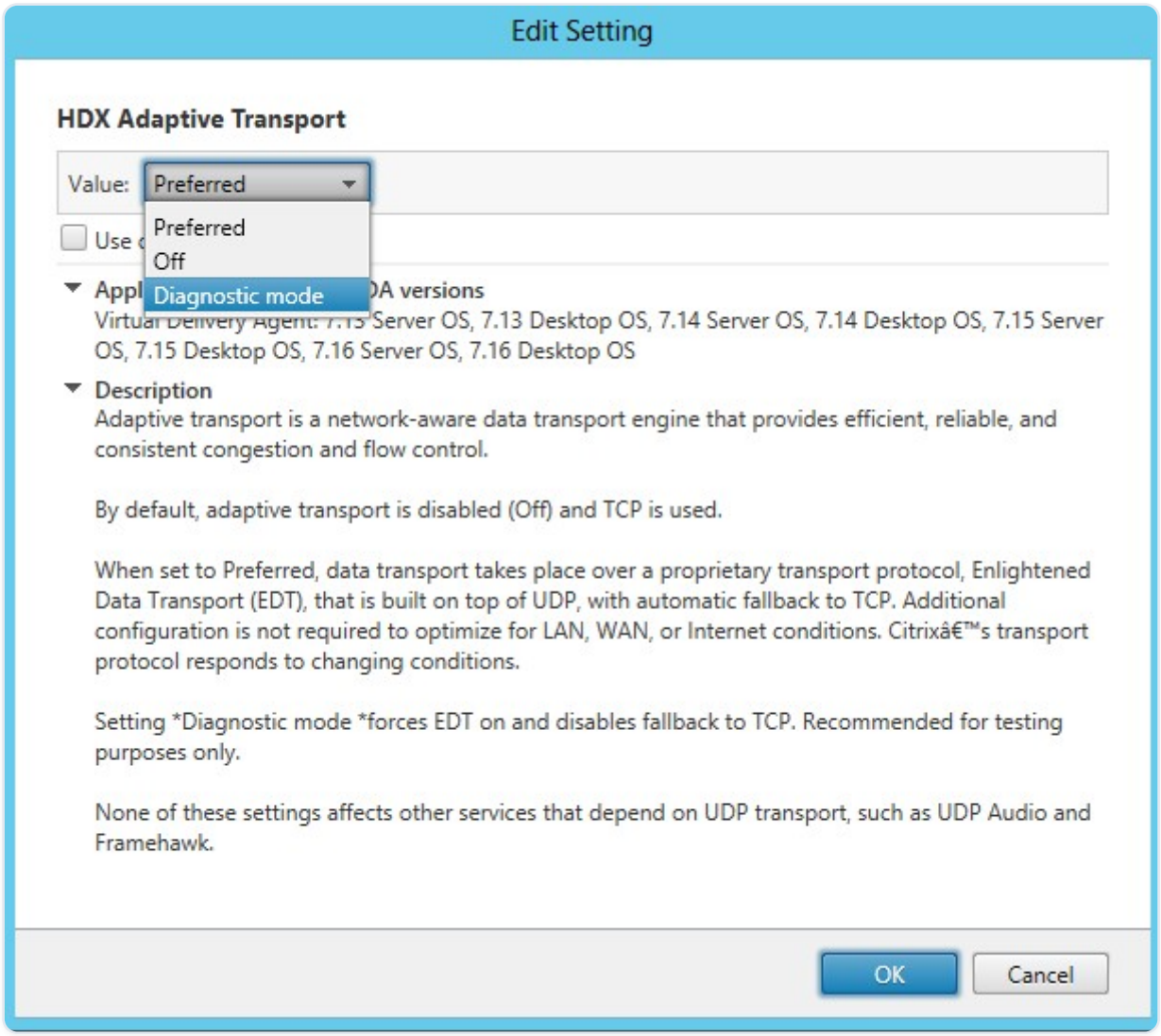
アダプティブトランスポート

アダプティブトランスポートは、Citrix Virtual Apps and Desktopsのデータ転送メカニズムです。これは、より高速で、拡張性が高く、アプリケーションの対話性を向上させ、困難な長距離WANおよびインターネット接続において、より対話的になります。アダプティブトランスポートの詳細については、アダプティブトランスポートを参照してください。

アダプティブトランスポートを有効にする

Citrix Studioで、**HDX™ Adaptive Transport**ポリシーが**Preferred**または**Diagnostic mode**に設定されていることを確認します。**Preferred**がデフォルトで選択されています。

- **Preferred**: 可能であればEnlightened Data Transport (EDT) を介したアダプティブトランスポートが使用され、TCPへのフォールバックが行われます。
- **Diagnostic mode**: EDTが強制的にオンになり、TCPへのフォールバックは無効になります。



アダプティブトランスポートを無効にする

アダプティブトランスポートを無効にするには、Citrix Studioで**HDX Adaptive Transport**ポリシーを**Off**に設定します。

アダプティブトランスポートが有効になっているか確認する

UDPリスナーが実行されているかどうかを確認するには、次のコマンドを実行します。

```
netstat -an | grep "1494\|2598"
```

通常の状況では、出力は次のようになります。

```
udp        0      0 0.0.0.0:2598            0.0.0.0:*
```

```
udp        0      0 :::1494                 :::*
```

EDT MTU検出

EDTは、セッション確立時に最大転送単位 (MTU) を自動的に決定します。これにより、パフォーマンスの低下やセッション確立の失敗につながる可能性のあるEDTパケットの断片化を防ぎます。

最小要件:

- リナックス VDA 2012
- Citrix Workspace™ アプリ 1911 Windows版
- シトリックス ADC:
  - 13.0.52.24
  - 12.1.56.22
- セッションの信頼性を有効にする必要があります

この機能をサポートし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、環境に適したカスタムEDT MTUを構成できます。詳細については、Knowledge Centerの記事CTX231821を参照してください。

> **警告:**
>
> レジストリを誤って編集すると、オペレーティングシステムの再インストールが必要になるような深刻な問題が発生する可能性があります。Citrixは、**レジストリエディター**の誤った使用によって生じる問題が解決できることを保証できません。**レジストリエディター**の使用は、お客様ご自身の責任において行ってください。編集する前に必ずレジストリをバックアップしてください。

VDAでEDT MTU検出を有効または無効にする

EDT MTU検出はデフォルトで無効になっています。

- EDT MTU検出を有効にするには、次のコマンドを使用して`MtuDiscovery`レジストリキーを設定し、VDAを再起動して、VDAが登録されるのを待ちます。

`/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t "REG_DWORD" -v "MtuDiscovery" -d "0x00000001" --force`

- EDT MTU検出を無効にするには、`MtuDiscovery`レジストリ値を削除します。

クライアントでEDT MTU検出を制御する

ICA®ファイルに`MtuDiscovery`パラメーターを追加することで、クライアント上でEDT MTU検出を選択的に制御できます。この機能を無効にするには、`Application`セクションの下に以下を設定します。

`MtuDiscovery=Off`

この機能を再度有効にするには、ICAファイルから`MtuDiscovery`パラメーターを削除します。

> **重要:**
>
> このICAファイルパラメーターが機能するには、VDAでEDT MTU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。

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

以下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。

```
create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "LogonDisplayString" -d "<text of custom logon banner message>" --force
```

```
create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "BackgroundImagePath" -d "<path to your custom logon screen background image>" --force
```

SUSE 15.4でこの機能を使用するには、`<http://download.opensuse.org/distribution/leap/15.3/repo/oss/>`から`imlib2`をインストールしてください。

> **ヒント:**

> LogonDisplayStringを使用してカスタムバナーメッセージを追加すると、ログオン画面の背景はデフォルトで青色になります。

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。この機能を有効にするには、事前にVDAにデスクトップ環境をインストールする必要があります。

次の表は、Linuxディストリビューションと、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デスクトップ環境の対応表を示しています。

LINUXディストリビューション	サポートされているデスクトップ
Debian11.3	メイト、グノーム、グノームクラシック、ケイディーイー
RHEL 8.6, RHEL 8.4	MATE (メイト), GNOME (グノーム), GNOME-Classic (グノームクラシック)
RHEL 7.9	メイト、グノーム、グノームクラシック、ケイディーイー
Rocky Linux (ロッキーリナックス) 8.6	メイト、グノーム、グノームクラシック、ケーディーイー
SUSE 15.3	MATE (メイト), GNOME (グノーム), GNOME-Classic (グノームクラシック)
Ubuntu (ウブントゥ) 22.04, Ubuntu (ウブントゥ) 20.04, Ubuntu (ウブントゥ) 18.04	メイト、グノーム、グノーム クラシック、KDE

デスクトップ切り替えコマンド

ターゲット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。

ターゲット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では、Magnusが起動時にロードされることがあります。回避策として、`sudo apt remove magnus`を実行してMagnusパッケージを削除できます。
- KDEの起動中に発生するQT警告を無効にするには、ルートユーザーとして`/usr/share/qt5/qtlogging.ini`を編集し、次のエントリを追加します:

  ```
  qt.qpa.xcb.xcberror.error=false

  qt.qpa.xcb.warning=false

  qt.qpa.xcb.error=false
  ```

- KDEでは画面のロック解除に失敗することがあります。回避策として、デスクトップの自動ロック機能を無効にすることをお勧めします。

一時ホームディレクトリでのログオン

Linux VDA上のマウントポイントが失敗した場合に備えて、一時ホームディレクトリを指定できます。一時ホームディレクトリが指定されている場合、マウントポイントが失敗すると、セッションログオン中にプロンプトが表示されます。ユーザーデータは一時ホームディレクトリに保存されます。

次の表は、ホームディレクトリ設定に役立つレジストリキーについて説明します。

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>`になります。マウントポイントが環境内のユーザーホームディレクトリと同じであることを確認してください。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "HomeMountPoint" -d "<A directory where the NFS share is to be mounted>" --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。このレジストリキーは`HomeMountPoint`に依存します。システムがマウントポイントが利用できないことを検出した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`
`RemoveHomeOnLogoff`	ユーザーログオフ時に一時ホームディレクトリを削除するかどうかを制御します。1は「はい」、0は「いいえ」を意味します。	`create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_DWORD" -v`

レジストリキー	説明	コマンド
		`"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。
> - アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。この問題に対処するため、アプリとデスクトップの配信には個別のデリバリーグループを作成することをお勧めします。
> - シームレスアプリケーションを使用するには、StoreFrontでシームレスモードを無効にしないでください。シームレスモードはデフォルトで有効になっています。「TWIMode=Off」を設定してすでに無効にしている場合は、「TWIMode=On」に変更す

> るのではなく、この設定を削除してください。そうしないと、公開されたデスクトップを起動できない可能性があります。

制限事項

Linux VDAは、単一ユーザーによる同じアプリケーションの複数の同時インスタンスの起動をサポートしていません。

アプリセッションでは、アプリケーションに固有のショートカットのみが期待どおりに機能します。

既知の問題

アプリケーションの公開中に、以下の既知の問題が確認されています。

- 非矩形ウィンドウはサポートされていません。ウィンドウの角にサーバー側の背景が表示される場合があります。
- 公開されたアプリケーションのウィンドウコンテンツのプレビューはサポートされていません。
- 複数のLibreOfficeアプリケーションを実行すると、これらのアプリケーションはプロセスを共有するため、最初に起動されたものだけがCitrix Studioに表示されます。
- 「Dolphin」のような公開されたQt5ベースのアプリケーションでは、アイコンが表示されない場合があります。この問題を解決するには、`<https://wiki.archlinux.org/title/Qt>`の記事を参照してください。

ランデブー V1

Citrix Gatewayサービスを使用する場合、Rendezvousプロトコルにより、トラフィックはCitrix Cloud™ Connectorsをバイパスし、Citrix Cloudコントロールプレーンに直接かつ安全に接続できます。

考慮すべきトラフィックには2つの種類があります。1) VDA登録とセッションブローカリングのための制御トラフィック。2) HDX™セッションのトラフィック。

Rendezvous V1では、HDXセッションのトラフィックがCloud Connectorsをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックをCloud Connectorsがプロキシする必要があります。

要件

- Citrix Workspace™およびCitrix Gatewayサービスを使用した環境へのアクセス。

- コントロールプレーン：Citrix DaaS（旧Citrix Virtual Apps and Desktops™サービス）。
- Linux VDAバージョン2112以降。
  - バージョン2112は、非透過型HTTPプロキシに必要な最小バージョンです。
  - バージョン2204は、透過型およびSOCKS5プロキシに必要な最小バージョンです。
- CitrixポリシーでRendezvousプロトコルを有効にします。詳細については、「Rendezvousプロトコルポリシー設定」を参照してください。
- VDAは、すべてのサブドメインを含め、`https://*.nssvc.net`にアクセスできる必要があります。その方法ですべてのサブドメインをホワイトリストに登録できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳細については、Citrix Cloudドキュメントの「インターネット接続要件」セクション（Virtual Apps and Desktopサービスの下）およびKnowledge Centerの記事「CTX270584」を参照してください。
- Cloud Connectorsは、セッションをブローカリングする際にVDAのFQDNを取得する必要があります。この目標を達成するには、サイトのDNS解決を有効にします。Citrix DaaS Remote PowerShell SDKを使用して、コマンド`Set-BrokerSite -DnsResolutionEnabled $true`を実行します。Citrix DaaS Remote PowerShell SDKの詳細については、「SDKとAPI」を参照してください。

プロキシ構成

VDAは、HTTPおよびSOCKS5プロキシを介したRendezvous接続の確立をサポートしています。

プロキシに関する考慮事項

Rendezvousでプロキシを使用する場合は、次の点を考慮してください。

- 非透過型HTTPプロキシとSOCKS5プロキシがサポートされています。
- パケットの復号化と検査はサポートされていません。VDAとGateway Service間のICA®トラフィックが傍受、復号化、または検査されないように例外を設定してください。そうしないと、接続が切断されます。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用したマシンベースの認証をサポートしています。プロキシサーバーに接続すると、Negotiate認証スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

> **注:**

> Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名（SPN）を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAはセッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。このプロキシURLは、**Rendezvous proxy**ポリシー設定から取得されます。SPNを作成しない場合、認証は失敗します。

- SOCKS5プロキシでの認証は現在サポートされていません。SOCKS5プロキシを使用する場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように例外を設定する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

透過型プロキシ

Rendezvousでは透過型HTTPプロキシがサポートされています。ネットワークで透過型プロキシを使用している場合、VDAに追加の設定は不要です。

非透過型プロキシ

ネットワークで非透過型プロキシを使用する場合は、Rendezvous proxy configuration設定を構成します。この設定を有効にすると、VDAが使用するプロキシを認識できるように、HTTPまたはSOCKS5プロキシアドレスを指定します。例:

- プロキシアドレス: `http://<URL or IP>:<port>` または `socks5://<URL or IP>:<port>`

Rendezvousの検証

すべての要件を満たしている場合は、Rendezvousが使用されているかどうかを検証するために、次の手順に従ってください。

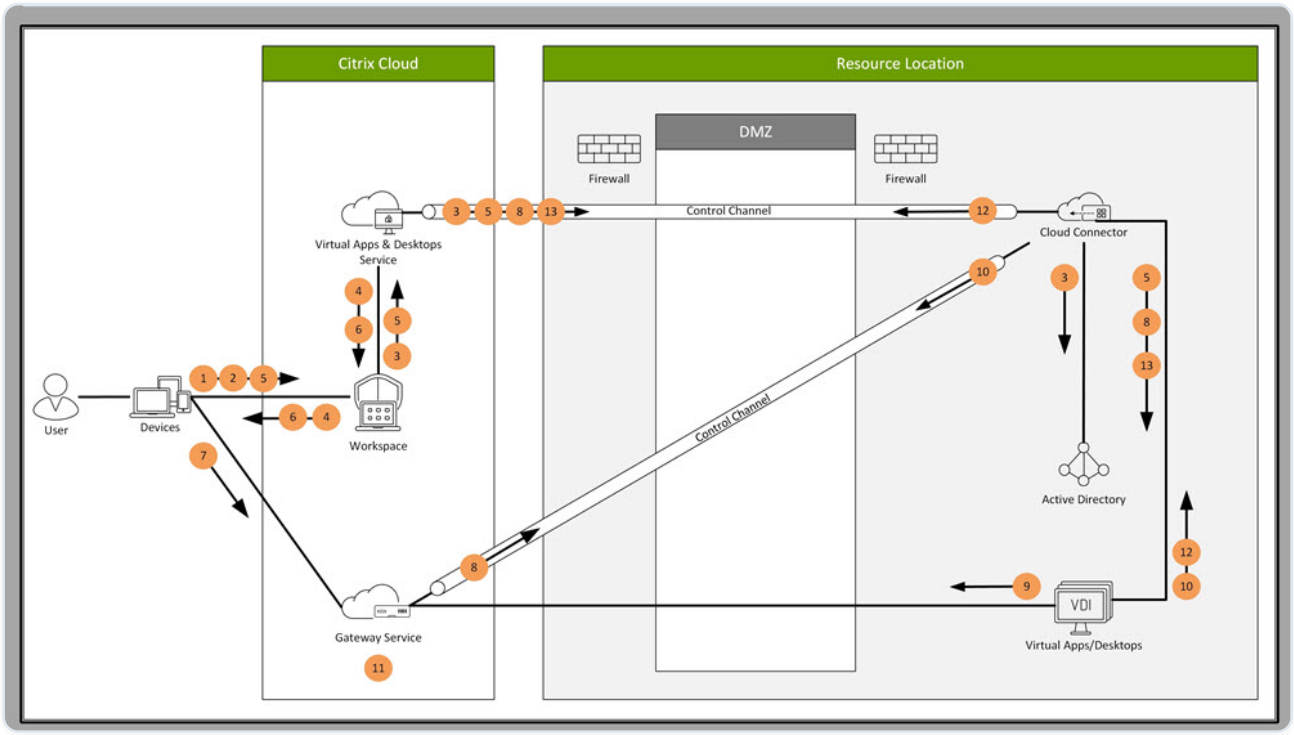
1. VDAでターミナルを起動します。
2. `/opt/Citrix/VDA/bin/ctxquery -f iP`を実行します。
3. TRANSPORT PROTOCOLSは接続の種類を示します。
    - TCP ランデブー: TCP - TLS - CGP - ICA
    - EDT ランデブー: UDP - DTLS - CGP - ICA
    - Cloud Connectorを介したプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

> **ヒント:**

> Rendezvousが有効な状態で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接続フローの概要を示しています。



フローを理解するには、以下の手順に従ってください。

1. Citrix Workspaceに移動します。
2. Citrix Workspaceで資格情報を入力します。
3. オンプレミスのActive Directoryを使用している場合、Citrix DaaS™は、Cloud Connectorチャネルを使用してActive Directoryで資格情報を認証します。
4. Citrix Workspaceは、Citrix DaaSから列挙されたリソースを表示します。
5. Citrix Workspaceからリソースを選択します。Citrix DaaSは、受信セッションの準備をするようVDAにメッセージを送信します。
6. Citrix Workspaceは、Citrix Cloudによって生成されたSTAチケットを含むICAファイルをエンドポイントに送信します。
7. エンドポイントはCitrix Gatewayサービスに接続し、VDAに接続するためのチケットを提供し、Citrix Cloudが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経由で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経由でCitrix DaaSとライセンスを検証します。
13. Citrix DaaSは、Cloud Connector経由でVDAにセッションポリシーを送信します。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には、1) VDA登録およびセッションブローカリング用の制御トラフィックと、2) HDX™セッションのトラフィックの2種類があります。

Rendezvous V1では、HDXセッションのトラフィックがCloud Connectorをバイパスできますが、VDA登録とセッションブローカリングのためのすべての制御トラフィックは、引き続きCloud Connectorがプロキシする必要があります。

標準のADドメイン参加マシンおよび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を使用する場合にサポートされます。ドメイン非参加マシンでは、Rendezvous V2により、HDXトラフィックと制御トラフィックの両方がCloud 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」(/ja-jp/citrix-virtual-apps-desktops/policies/reference/ica-policy-settings.html#rendezvous-protocol)を参照してください。
- VDAは、すべてのサブドメインを含め、`https://*.nssvc.net`にアクセスできる必要があります。その方法ですべてのサブドメインをホワイトリストに登録できない場合は、代わりに`https://*.c.nssvc.net`と`https://*.g.nssvc.net`を使用してください。詳しくは、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の場合。
  - UDP 443で、EDT Rendezvousの場合。

プロキシ構成

VDAは、Rendezvousを使用する際に、制御トラフィックとHDXセッショントラフィックの両方でプロキシ経由の接続をサポートします。両方のトラフィックタイプで要件と考慮事項が異なるため、注意深く確認してください。

制御トラフィックのプロキシに関する考慮事項

- HTTPプロキシのみがサポートされています。
- パケットの復号化と検査はサポートされていません。VDAとCitrix Cloudコントロールプレーン間の制御トラフィックが傍受、復号化、または検査されないように例外を構成してください。そうしないと、接続が失敗します。
- プロキシ認証はサポートされていません。
- 制御トラフィックのプロキシを構成するには、次のようにレジストリを編集します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ProxySettings" -d "http://<URL or IP>:<port>" --force
```

HDXトラフィックのプロキシに関する考慮事項

- HTTPおよびSOCKS5プロキシがサポートされています。
- EDTはSOCKS5プロキシでのみ使用できます。
- HDXトラフィックのプロキシを構成するには、Rendezvousプロキシ構成ポリシー設定を使用します。

- パケットの復号化と検査はサポートされていません。VDAとCitrix Cloudコントロールプレーン間のHDXトラフィックが傍受、復号化、または検査されないように例外を構成してください。そうしないと、接続が失敗します。
- HTTPプロキシは、NegotiateおよびKerberos認証プロトコルを使用してマシンベースの認証をサポートします。プロキシサーバーに接続すると、Negotiate認証スキームが自動的にKerberosプロトコルを選択します。Kerberosは、Linux VDAがサポートする唯一のスキームです。

> **注:**
>
> Kerberosを使用するには、プロキシサーバーのサービスプリンシパル名 (SPN) を作成し、それをプロキシのActive Directoryアカウントに関連付ける必要があります。VDAは、セッション確立時に`HTTP/<proxyURL>`の形式でSPNを生成します。このとき、プロキシURLは**Rendezvousプロキシ**ポリシー設定から取得されます。SPNを作成しない場合、認証は失敗します。

- SOCKS5プロキシでの認証は現在サポートされていません。SOCKS5プロキシを使用する場合は、Gateway Serviceアドレス（要件で指定）宛てのトラフィックが認証をバイパスできるように例外を構成する必要があります。
- SOCKS5プロキシのみがEDTを介したデータ転送をサポートします。HTTPプロキシの場合は、ICAのトランスポートプロトコルとしてTCPを使用してください。

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に追加の設定は必要ありません。

Rendezvous V2の構成方法

環境でRendezvousを構成する手順は次のとおりです。

1. すべての要件が満たされていることを確認します。
2. VDAのインストール後、次のコマンドを実行して必要なレジストリキーを設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"GctRegistration" -d "0x00000001" --force
```

3. VDAマシンを再起動します。

4. Citrixポリシーを作成するか、既存のポリシーを編集します。
    - Rendezvousプロトコル設定を**許可**に設定します。
    - Citrixポリシーフィルターが適切に設定されていることを確認します。このポリシーは、Rendezvousを有効にする必要があるマシンに適用されます。
    - Citrixポリシーが、他のポリシーを上書きしないように正しい優先順位を持っていることを確認します。

Rendezvousの検証

セッションがRendezvousプロトコルを使用しているかどうかを確認するには、ターミナルで`/opt/Citrix/VDA/bin/ctxquery -f iP`コマンドを実行します。

表示されるトランスポートプロトコルは、接続の種類を示します。

- TCPランデブー：TCP - TLS - CGP - ICA
- EDT ランデブー：UDP - DTLS - CGP - ICA
- Cloud Connector を介したプロキシ: TCP - PROXY - SSL - CGP - ICA または UDP - PROXY - DTLS - CGP - ICA

Rendezvous V2 が使用されている場合、プロトコルバージョンは 2.0 と表示されます。

> **ヒント:**
>
> Rendezvous が有効になっている状態で VDA が Citrix Gateway サービスに直接到達できない場合、VDA は HDX セッションを Cloud Connector を介してプロキシするようにフォールバックします。

DTLS を使用したユーザーセッションの保護

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、「トランスポート層セキュリティ」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™ アダプティブトランスポート**ポリシーが**優先**または**診断モード**に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**/opt/Citrix/VDA/sbin** にある **enable_vdassl.sh** ツールを使用して SSL 暗号化を有効（または無効）にします。このツールで利用可能なオプションについては、`/opt/Citrix/VDA/sbin/enable_vdassl.sh -h` コマンドを実行してください。

> **注:**
>
> Linux VDA は DTLS 1.0 と DTLS 1.2 の両方をサポートしており、デフォルトでは DTLS 1.2 を使用します。Citrix Workspace アプリでどのバージョンの DTLS が使用されているかを確認してください。Linux VDA と Citrix Workspace アプリの両方で同じバージョンの DTLS が使用されていることを確認してください。Citrix Workspace アプリが DTLS 1.0 のみをサポートしている場合（例: Citrix Receiver for Windows 4.11）、**enable_vdassl.sh** ツールを使用して、**SSLMinVersion** を **TLS_1.0** に、**SSLCipherSuite** を **COM** または **ALL** に設定します。

TLSを使用してユーザーセッションを保護する

バージョン7.16以降、Linux VDAはセキュアなユーザーセッションのための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キー
- 中間証明書（オプション）

サーバー証明書の例：

サーバー証明書の例(/en-us/linux-virtual-delivery-agent/2301/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/2301/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。

CONTROLLER で TLS 暗号化を有効にする

> **注:**

> TLS 暗号化は、デリバリーグループ全体に対してのみ有効にできます。特定のアプリケーションに対して TLS 暗号化を有効にすることはできません。

> Controller の PowerShell ウィンドウで、ターゲットデリバリーグループの TLS 暗号化を有効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

> **注:**

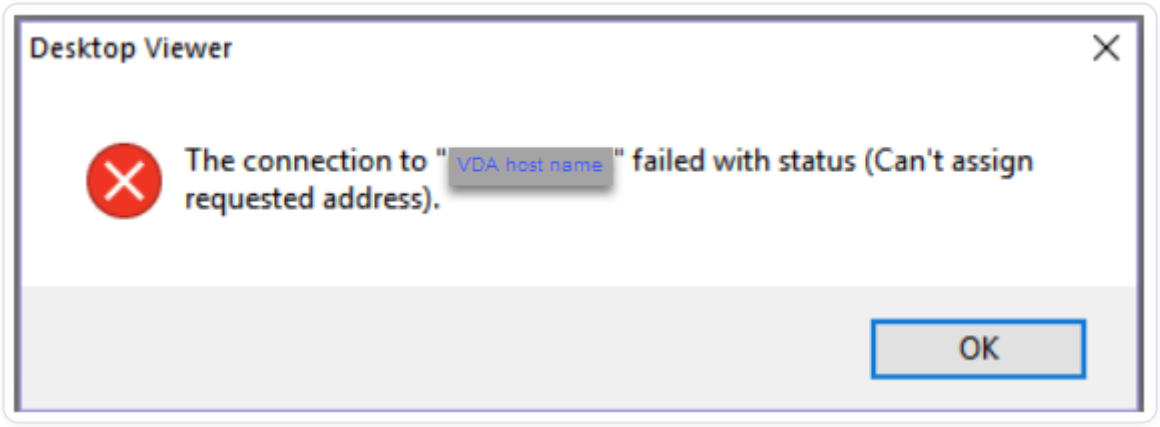
> ICA® セッションファイルに VDA FQDN のみが含まれるようにするには、`Set-BrokerSite –DnsResolutionEnabled $true` コマンドも実行できます。このコマンドは DNS 解決を有効にします。DNS 解決を無効にすると、ICA セッションファイルは VDA IP アドレスを開示し、`SSLProxyHost` や `UDPDTLSPort` などの TLS 関連項目にのみ FQDN を提供します。

> Controller で TLS 暗号化を無効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

トラブルシューティング

公開されたデスクトップセッションにアクセスしようとすると、Windows 用 Citrix Workspace™ アプリで次の「"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にICAセッションをシームレスに再接続します。セッションの信頼性について詳しくは、「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暗号化について詳しくは、「TLSを使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studioでのポリシー設定

Citrix Studioでセッションの信頼性について次のポリシーを設定できます。

- セッションの信頼性接続
- セッションの信頼性タイムアウト
- セッションの信頼性ポート番号
- 再接続UIの透過性レベル

詳しくは、「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、この順序でVDAサービスとHDX™サービスを再起動してください。

Linux VDAでの設定

- **セッションの信頼性TCPリスナーの有効化/無効化**

デフォルトでは、セッションの信頼性TCPリスナーは有効になっており、ポート2598でリッスンしています。リスナーを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "fEnableWinStation" -d "0x00000000"
```

注: 設定を有効にするには、HDXサービスを再起動してください。TCPリスナーを無効にしても、セッションの信頼性は無効になりません。**セッションの信頼性接続**ポリシーを通じて機能が有効になっている場合、セッションの信頼性は他のリスナー（SSLなど）を介して引き続き利用できます。

- **セッションの信頼性ポート番号**

次のコマンドを使用して、セッションの信頼性ポート番号を設定することもできます（例としてポート番号2599を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "PortNumber" -d  "2599"
```

注: 設定を有効にするには、HDXサービスを再起動してください。ポート番号が**Citrix Studio**のポリシー設定で設定されている場合、Linux VDAでの設定は無視されます。VDA上のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように構成されていることを確認してください。

- **サーバーからクライアントへのキープアライブ間隔**

セッションでアクティビティがない場合（たとえば、マウスの動きや画面の更新がない場合）、Linux VDAとクライアントの間でキープアライブメッセージが送信されます。キープアライブメッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントからの応答がない場合、クライアントが再接続するまでセッションは中断されます。この設定は、連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpServerToClientKeepAlive"  -d  "10"  --force
```

- **クライアントからサーバーへのキープアライブ間隔**

この設定は、ICA®クライアントからLinux VDAに送信される連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpClientToServerKeepAlive"  -d  "10"  --force
```

トラブルシューティング

ポリシー設定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ない。

この問題を回避するには、次の手順を実行します。

1. Citrix Studioでポリシー設定を通じてセッションの信頼性を有効にした後、VDAサービスとHDXサービスがこの順序で再起動されていることを確認してください。
2. VDAで、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は、次のコマンドを実行してリスナーを有効にします。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations
\cgp"   -v  "fEnableWinStation" -d "0x00000001"
```

セッション録画（実験的機能）

実験的機能として、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。

セッション録画を有効または無効にする

Linux VDAのセッション録画を有効または無効にするには、**SmAudAllowed**をそれぞれ**1**または**0**に設定します。次のコマンドを使用できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000001" --force
```

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "SmAudAllowed" -d "0x00000000" --force
```

注:

Linux VDAでセッション録画を有効にすると、ユーザーがセッションにログオンしたときに、セッションが録画されていることが通知されます。

録画ファイルのサイズを指定する

録画のサイズが大きくなると、録画ファイルのダウンロードに時間がかかり、再生中にシークスライダーを使用して移動する際の応答が遅くなります。ファイルサイズを制御するには、ファイルのしきい値制限を指定します。録画がこの制限に達すると、現在のファイルは閉じられ、録画を続行するために追加のファイルが作成されます。このアクションは「ロールオーバー」と呼ばれます。

次のコマンドを使用して、ロールオーバーの2つのしきい値を指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverFileSizeInMB" -d "0x00000032" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD"
-v "RolloverTimeInHours" -d "0x0000000c" --force
```

- **RolloverFileSizeInMB**。現在のファイルがこ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**。この期間に達すると、現在のファイルが閉じられ、新しいファイルが開きます。デフォルトでは、セッションが12時間録画されるとロールオーバーが発生します。サポートされる値：1～24。

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。録画が3時間で17 MBに達すると、セッション録画はファイルを閉じ、新しいファイルを開きます。

多く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。

録画の保存場所を指定する

録画ファイルはデフォルトで`/var/xdl/session_recordings`に保存されます。別のパスを指定するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_SZ" -v
"Path" -d "<your custom storage path>" --force
```

録画は、ローカルドライブまたはネットワークパスを指すマウントポイントに保存できます。設定したストレージパスに適切なアクセス許可を構成し、ユーザー`ctxsrvr`にそのパスへの書き込み許可を付与します。

録画を表示

録画を表示するには、以下の手順に従ってSession Recording PlayerまたはSession Recording Web Playerをインストールします。

1. Citrixアカウントの資格情報を使用して、Citrix Virtual Apps and Desktopsダウンロードページにアクセスし、製品ファイルをダウンロードします。ファイルを解凍します。
2. SessionRecordingPlayer.msi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ことがあります。

仮想チャネルSDK (実験的)

Linux VDA向け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上で動作するサーバー側アプリケーションを作成できます。詳細については、Citrix Virtual Channel SDK for the Linux VDAドキュメントを参照してください。

Citrix Virtual Channel SDK for the Linux VDAは、Citrix Virtual Apps and Desktopsダウンロードページからダウンロードできます。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックしてLinux VDAのダウンロードを選択します。

Wayland (実験的)

実験的な機能として、Linux VDAはRHEL 9.0、Rocky Linux 9.0、およびUbuntu 22.04上の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が使用されているか確認する

1. Linuxでターミナルウィンドウを開きます。
2. **echo $XDG_SESSION_TYPE** コマンドを実行します。

Waylandが使用されている場合、出力に **'wayland'** が表示されます。

制限事項

Waylandを使用している場合、以下の制限が確認されています。

- クライアントデバイスのキーボードレイアウトは、VDAのキーボードレイアウトと同期されません。
- RHEL 9.0、Rocky Linux 9.0では、セッションからログオフするのに約20秒かかります。

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 2301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
30 Jan 2023	Linux Virtual Delivery Agent 2301の初期リリース	「新機能」を参照してください。