Linux Virtual Delivery Agent 1912 LTSR

Contents

新機能

September 25, 2023

累積更新プログラム 8（CU8）は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU8 ではさまざまな問題に対応しているため、パフォーマンス、セキュリティ、および安定性が総合的に向上しています。

> 注:
>
> CU3 リリース以降、前提条件として Linux VDA をインストールする前に.NET Core ランタイム 3.1 をインストールします。

累積更新プログラム **8**（**CU8**）

September 25, 2023

リリース日：2023 年 9 月 11 日

このリリースについて

累積更新プログラム 8（CU8）は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU8 ではさまざまな問題に対応しているため、パフォーマンス、セキュリティ、および安定性が総合的に向上しています。

累積更新プログラム **7**（**CU7**）

July 19, 2023

リリース日：2023 年 3 月 15 日

このリリースについて

累積更新プログラム 7（CU7）Hotfix 1（19.12.7001）は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。この Hotfix では、1912 LTSR CU7 のリリース以降に報告された 1 個の問題が修正されています。

1912 LTSR CU7 Hotfix 1（**19.12.7001**）で解決された問題

July 19, 2023

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU7 以降で解決されています：

- この修正により、セキュリティ上の問題が 1 件解決されます。詳しくは、Knowledge Center のCTX559370を参照してください。

1912 LTSR CU7 で解決された問題

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU6 以降で解決されています:

- Microsoft Windows を実行しているノート PC から Linux VDA セッションに接続しており、ノート PC のふたを閉じるアクションが［何もしない］、［休止状態］、または［スリープ状態］に設定されている場合は、ランダムキーが VDA セッションに挿入されることがあります。[CVADHELP-18438]
- 特定のノートブックを Linux VDA に接続して **Fn** キーを押すと、**Delete** キーとして機能する場合があります。[CVADHELP-21630]

累積更新プログラム **6**（**CU6**）

November 7, 2022

リリース日：2022 年 10 月 31 日

このリリースについて

CU6 で解決された問題はありません。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 5（CU5）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 4（CU4）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 3（CU3）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 2（CU2）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 1（CU1）

Linux Virtual Delivery Agent 1912 LTSR（初期リリース）

このリリースの既知の問題について

廃止と削除

Citrix Product Subscription Advantage の有効期限

累積更新プログラム **5**（**CU5**）

March 11, 2022

リリース日：2022 年 3 月 9 日

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 5（CU5）では、1912 LTSR CU4 のリリース以降に報告された 2 個の問題が修正されています。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 4（CU4）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 3（CU3）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 2（CU2）

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 1（CU1）

Linux Virtual Delivery Agent 1912 LTSR（初期リリース）

このリリースの既知の問題について

廃止と削除

Citrix Product Subscription Advantage の有効期限

1912 LTSR CU5 で解決された問題

March 11, 2022

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU4 以降で解決されています：

- Linux VDA デスクトップがキーボードとマウスの入力に応答しない場合があります。[CVADHELP-18498]
- 複数のドメインコントローラーがある環境で 1 つのドメインコントローラーをシャットダウンすると、Linux VDA でのセッションの起動が失敗する可能性があります。[CVADHELP-18900]

累積更新プログラム 4（CU4）

January 28, 2022

リリース日：2021 年 11 月 3 日

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 4（CU4）では、1912 LTSR CU3 のリリース以降に報告された 3 個の問題が修正されています。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 3（CU3）

1912 LTSR CU4 で解決された問題

January 28, 2022

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU3 以降で解決されています:

- Mac 用または Linux 用の Citrix Workspace アプリを使用して接続された Linux VDA では、Shift + Tab キーを押すと、**Shift** キーの 2 度押しとして登録されることがあります。[CVADHELP-16831]
- Azure での Linux 仮想マシンの作成に Machine Creation Services が使用されている場合は、RHEL で Linux VDA セッションを開始しようとすると失敗する可能性があります。[CVADHELP-17244]
- SUSE または RHEL で Linux VDA をアンインストールしても、/opt/Citrix/にある空のフォルダーが削除されない場合があります。[CVADHELP-18241]

累積更新プログラム 3（CU3）

January 28, 2022

リリース日: 2021 年 5 月 12 日

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 3（CU3）では、1912 LTSR CU2 のリリース以降に報告された 9 個の問題が修正されています。

このリリースの既知の問題について

廃止と削除

Citrix Product Subscription Advantage の有効期限

1912 LTSR CU3 で解決された問題

January 28, 2022

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU2 以降で解決されています:

- 64 ビット Linux VDA をアップグレードした後、ユーザーがアプリケーションを起動しようとすると失敗する可能性があります。アプリケーションが起動中であることを示すメッセージがユーザーに表示された後、メッセージは消えます。その結果、複数の古いセッションが VDA に残ります。[CVADHELP-15899]
- スキャンの実行後、`ctxmonitorservice`プロセスが SIGABRT エラーで予期せず終了する場合があります。[CVADHELP-15969]
- Xiaomi Mi 10 スマートフォンで、画像転送プロトコル（PTP）およびメディア転送プロトコル（MTP）USB オプションを選択しようとすると失敗する場合があります。[CVADHELP-16188]
- Citrix Desktop Viewer（CDViewer.exe）ウィンドウが消えると、Linux VDA への再接続が失敗する場合があります。[CVADHELP-16239]
- Linux VDA では、一部のアプリケーションが［デバイス］オプションに表示される Web カメラデバイスを認識しない場合があります。[CVADHELP-16247]
- Linux VDA セッションで Web カメラに USB リダイレクトを使用すると、`ctxusbsd`プロセスが`segfault`エラーで予期せず終了する場合があります。[CVADHELP-16366]
- `ctxcdmd`プロセスが正しい SELinux セキュリティコンテキストで実行されない可能性があります。[CVADHELP-16381]
- スマートカードが Linux VDA に構成されている場合、スマートカードが機能しない可能性があります。[CVADHELP-16488]
- Linux VDA セッションに表示されるテキストが、歪んでぼやけている可能性があります。[CVADHELP-17199]

累積更新プログラム 2（CU2）

January 28, 2022

リリース日：2020 年 11 月 19 日

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 2（CU2）では、1912 LTSR CU1 のリリース以降に報告された 5 個の問題が修正されています。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 1（CU1）

Linux Virtual Delivery Agent 1912 LTSR（初期リリース）

このリリースの既知の問題について

廃止と削除

Citrix Product Subscription Advantage の有効期限

1912 LTSR CU2 で解決された問題

January 28, 2022

次の問題は、Linux Virtual Delivery Agent 1912 LTSR CU1 以降で解決されています:

- LDAP 署名を有効にすると、Linux VDA を Delivery Controller に登録する操作に失敗する場合があります。[CVADHELP-14481]
- VDA を起動しようとすると、灰色の画面になる場合があります。この問題は、HDX ユーザーポリシー検証のタイムアウトの結果です。タイムアウトは、LDAP サーバーが多数あり、1 つ以上のサーバーが VDA にアクセスできない場合に発生する可能性があります。[CVADHELP-14746]
- Linux VDA は、HDX ポリシー設定の最上位の規則である ［クライアント **USB** デバイスリダイレクト規則］のみを認識する場合があります。他の規則は破棄されます。[CVADHELP-14971]
- 公開アプリケーションをシームレスモードで起動すると、公開アプリケーションが常に一番上に表示され、ローカルアプリケーションの上に表示されます。公開アプリケーションを最小化しない限り、ローカルアプリケーションを前面に表示することはできません。[CVADHELP-15134]
- Machine Creation Services（MCS）によって作成された Ubuntu 仮想マシンにログオンすると、.bashrc や.profile などの特定のファイルが正常にホームフォルダーに自動的にコピーされないことがあります。[CVADHELP-15306]
- Ubuntu を実行している Linux VDA に NVIDIA GRID グラフィックカードがインストールされている場合、セッションのサイズを変更しようとすると、セッションが予期せず終了することがあります。[CVADHELP-15664]
- ロケールを英語以外の言語に変更すると、パフォーマンスカウンターが文字列値を数値に変換できなくなり、VDA ログに次のエラーが生成されることがあります。

[PerfCounter] [エラー**] SysStat.ReadUpTime**：要素「**29363.68**」を変換しようとして **NumberFormatException** が発生しました。エラー：入力文字列が正しい形式ではありませんでした。

[CVADHELP-15767]

累積更新プログラム **1**（**CU1**）

August 10, 2021

リリース日：2020 年 5 月 7 日

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 1（CU1）は、1912 LTSR の初期リリース以降に報告された 9 個以上の問題を修正します。

Linux Virtual Delivery Agent（初期リリース）

このリリースの既知の問題について

廃止と削除

Citrix Product Subscription Advantage の有効期限

1912 LTSR CU1 で解決された問題

January 28, 2022

次の問題は、Linux Virtual Delivery Agent 1912 LTSR（初期リリース）以降で解決されています：

- Linux VDA がログオンユーザーの一覧を表示できないことがあります。[CVADHELP-13659]
- リムーバブル USB ドライブを Linux VDA に汎用リダイレクトしようとすると失敗する可能性があります。この問題は、USB ドライブが NTFS（New Technology File System）でフォーマットされている場合に発生します。[CVADHELP-13675]
- Linux VDA をバージョン 1909 またはバージョン 1912 に更新後、初期化に時間がかかることがあります。[CVADHELP-13802]
- Quest Authentication Service を使用する Linux VDA は、Delivery Controller への登録に失敗する場合があります。この問題は、Linux VDA バージョン 1909、1912 LTSR 初期リリース、および 2003 を使用している場合に発生します。[CVADHELP-14027]

- Linux VDA は、`Target frame rate`（FramesPerSecond）設定で指定されたフレーム数/秒を達成できない場合があります。この問題は、GPU が Linux VDA にインストールされている場合に発生します。[CVADHELP-14267]
- システムを再起動した後、`ctxjproxy`サービスが LDAP サーバーを見つけられない場合があります。[CVADHELP-14269]
- Linux VDA が Delivery Controller への登録に失敗する場合があります。この問題は、Linux VDA が Delivery Controller と通信するためのポートが 80 でない場合に発生します。[CVADHELP-14270]
- Linux VDA をインストールすると、.NET Core ランタイムスクリプトの認証が検証されない場合があります。[CVADHELP-14424]

このリリースについて

August 10, 2021

新機能

1912 LTSR の新機能

Linux VDA のバージョン 1912 には、次の新機能と強化された機能があります。

AWS プラットフォームでの **MCS** のサポート

Machine Creation Services（MCS）を使用して、AWS プラットフォームで Linux 仮想マシンを作成できます。詳しくは、「MCS を使用した Linux 仮想マシンの作成」を参照してください。

現在実行中の **VDA** をテンプレートとして使用

MCS を使用して Linux 仮想マシンを作成する場合、現在実行中の VDA をテンプレートとして使用し、既存の構成をすべて継承できます。この実行中の VDA は、手動でインストールすることも、簡易インストールでインストールすることもできます。詳しくは、「MCS を使用した Linux 仮想マシンの作成」を参照してください。

クライアントドライブマッピング：大容量ファイル転送のサポート

クライアントドライブマッピングでは、Linux VDA とクライアントデバイス間で、4GB 以上のサイズのファイル転送がサポートされるようになりました。この拡張機能では、クライアントで 1808 以降の Windows 向け Citrix Workspace アプリが実行されている必要があります。

> 注:
>
> このリリースでは、サポートされているすべてのディストリビューションで OpenJDK をバージョン 1.8.0 に更新します。

1912 LTSR で解決された問題

January 28, 2022

次の問題は、Linux Virtual Delivery Agent 1909 以降で解決されています:

- 4K モニターでは、キーストロークと更新頻度に関連した GPU パフォーマンスの問題が発生する可能性があります。[CVADHELP-12661]
- マウスやキーボードが同じウィンドウでフォーカスされていない場合、またはマウスがフォーカスの変更に失敗すると、Linux VDI セッションが応答しなくなることがあります。[CVADHELP-12768]
- IPv6 アドレスのみを使用する仮想マシン（VM）では、Linux VDA の登録に失敗することがあります。[CVADHELP-13103]
- ポリシーをデフォルトに設定すると、Linux VDA でデータベースが更新されないことがあります。この問題は、優先度の高いポリシーがデフォルトに設定されている優先度の低いポリシーをリセットできないために発生します。[CVADHELP-13107]
- ローカルキーボードレイアウト機能を有効にすると、クライアント側のハンガリー語の環境でキーボードレイアウトの同期が機能しないことがあります。ローカル設定を **DE** にしてアプリケーションを起動すると、言語は同期されますが、ハンガリー語のレイアウトでは機能しません。[CVADHELP-13199]
- XClient と XServer 間の通信を保護するように`Xauthority`を構成すると、IPv4 アドレスのみが追加されます。IPv6 アドレスは追加されません。[CVADHELP-13255]
- Ubuntu 18.04 で Machine Creation Services（MCS）を使用してマシンカタログを作成または更新しようとすると、失敗することがあります。[CVADHELP-13178]

既知の問題

July 19, 2023

このリリースでは、次の問題が確認されています:

- 非シームレスな公開アプリケーションは、起動直後に終了する場合があります。この問題は、mutter-3.28.3-4 より新しいバージョンに Mutter が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 以前のバージョンを使用してください。[LNXVDA-6967]
- HDX 3D Pro を有効にせずに NVIDIA GRID 3D カードを使用すると、Linux VDA が正常に動作しません。この問題は、RHEL 7.5 以前、SUSE 12.3 以前、Ubuntu 16.04 で発生します。その理由は、複数の OpenGL ライブラリがこれらの Linux ディストリビューションのグラフィックシステム環境内で共存できないためです。
- ファイルのダウンロード中、予期しないウィンドウが表示されます。このウィドウはファイルのダウンロード機能に影響を及ぼすことはなく、しばらくしてから自動的に消えます。[LNXVDA-5646]
- PulseAudio のデフォルト設定によって、サウンドサーバープログラムが 20 秒間非アクティブ状態になった後、終了します。PulseAudio が終了すると、オーディオは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。[LNXVDA-5464]
- SUSE 12.3 の`libtcmalloc 4.3.0`によって、プロセスが予期せず終了することがあります。
- `ctxhdx`サービスが、Ubuntu 16.04 VDA および SUSE 12.3 VDA で予期せず終了することがあります。この問題は、GNU C ライブラリ (`glibc`) のバージョン 2.22～2.24 で発生します。この問題は`glibc 2.25`で解決されています。SUSE 12.3 ディストリビューションを使用している場合、SUSE が提供するパッチをインストールして問題を解決できます。Linux VDA 7.17 がリリースされた時点での Ubuntu 16.04 の修正はありません。[LNXVDA-4481]
- SSL 暗号化が有効でセッション画面の保持が無効になっている場合、Linux 向け Citrix Workspace アプリでセッションを開始できません。[RFLNX-1557]
- `indicator-datetime-service`プロセスで $TZ 環境変数が使用されません。クライアントとセッションが異なるタイムゾーンにある場合、Ubuntu 16.04 Unity Desktop の Unity パネルにはクライアントの時刻が表示されません。[LNXVDA-2128]
- Ubuntu のグラフィック: HDX 3D Pro で、Desktop Viewer をサイズ変更した後、アプリケーションの周囲に黒い枠が表示されたり、まれに背景が黒く表示される場合があります。
- Linux VDA 印刷リダイレクトで作成されたプリンターは、セッションからログアウト後、削除されることがあります。
- ディレクトリにファイルやサブディレクトリが多数含まれているときに、CDM ファイルが欠落します。クライアント側のファイルやディレクトリが非常に多い場合、この問題が生じることがあります。
- このリリースでは、英語以外の言語では UTF-8 エンコードのみがサポートされます。
- セッションのローミング時、Android 向け Citrix Workspace アプリで CapsLock が通常とは反対の状態になる場合があります。Android 向け Citrix Workspace アプリへの既存の接続をローミングすると、CapsLock 状態が失われる場合があります。回避策として、拡張キーボードの Shift キーを使用して大文字と小文字を切り替えます。
- Mac 向け Citrix Workspace アプリを使用して Linux VDA に接続している場合、Alt キーを使用するショートカットキーが機能しないことがあります。Mac 向け Citrix Workspace アプリでは、左右どちらの

option/alt キーを押しても、デフォルトでは AltGr が送信されます。Citrix Workspace アプリの設定でこの動作を変更することはできますが、結果はアプリケーションによって異なります。

- Linux VDA をドメインに再度追加すると、登録できません。再度追加することにより、Kerberos キーの新しいセットが生成されます。しかし、ブローカーは、Kerberos キーの以前のセットに基づいた、キャッシュに存在する期限切れの VDA サービスチケットを使用する可能性があります。VDA がブローカーに接続しようとするときに、ブローカーは VDA に返すセキュリティコンテキストを確立できないことがあります。通常見られる現象は、VDA 登録の失敗です。

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

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

  ```
  klist -li 0x3e4 purge
  <!--NeedCopy-->
  ```

  このコマンドにより、Citrix Broker Service を実行する Network Service プリンシパルが LSA キャッシュに保持するサービスチケットはすべて削除されます。これにより、ほかの VDA のサービスチケットが削除されます。また、その他のサービスのサービスチケットも削除される可能性があります。ただし、この処理は悪影響を及ぼしません。これらのサービスチケットは、再度必要になった時に KDC から再取得できます。

- オーディオのプラグアンドプレイがサポートされません。ICA セッションでオーディオの録音を開始する前に、オーディオキャプチャデバイスをクライアントマシンに接続できます。オーディオ録音アプリケーションの開始後にキャプチャデバイスを接続した場合は、アプリケーションが応答しなくなって再起動する必要が生じる可能性があります。録音中にキャプチャデバイスが取り外されると、同様の問題が発生する可能性があります。
- Windows 向け Citrix Workspace アプリでオーディオ録音中にオーディオの歪みが生じることがあります。

サードパーティ製品についての通知

August 15, 2022

Linux Virtual Delivery Agent 1912 LTSR（PDF のダウンロード）

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

廃止

November 11, 2021

この記事の告知は、お客様が適宜ビジネス上の決定を下せるように、段階的に廃止されるプラットフォーム、Citrix 製品、機能について前もってお知らせするためのものです。シトリックスではお客様の使用状況とフィードバックをチェックして、各プラットフォーム、Citrix 製品、機能を撤廃するかどうかを判断しています。お知らせする内容は以降のリリースで変わることがあり、廃止される機能がすべて含まれるわけではありません。
製品ライフサイクルサポートについて詳しくは、「製品ライフサイクルサポートポリシー」の文書を参照してください。

廃止と削除

廃止または削除されるプラットフォーム、Citrix 製品、機能を以下の表に示します。
廃止されたアイテムはすぐには削除されません。このリリースでは Citrix が引き続きサポートしていますが、今後のリリースでは削除される予定です。
削除されたアイテムは Linux VDA で削除されたか、サポートされなくなりました。

アイテム	廃止が発表されたバージョン	削除されたバージョン
RHEL 6.9 のサポート	1909	1909
RHEL7.5、CentOS 7.5 のサポート	1903	1903
RHEL7.4、CentOS 7.4 のサポート	1811	1811
RHEL 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 のサポート	7.16	7.16
SUSE 11.4	7.16	7.16

システム要件

January 28, 2022

Linux ディストリビューション

> 注:
>
> このトピックで説明されていないシステム要件コンポーネント（Citrix Workspace アプリなど）については、各コンポーネントのドキュメントを参照してください。
>
> CU3 リリース以降、前提条件として Linux VDA をインストールする前に.NET Core ランタイム 3.1 をインス

> トールします。詳しくは、https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managersを参照してください。
>
> Linux VDA では、暗号化で SecureICA はサポートされていません。Linux VDA で SecureICA を有効にすると、セッションの起動に失敗します。
>
> 長期サービスリリース（LTSR）環境での最新リリース（CR）の使用について、およびその他のよくある質問については、Knowledge Center の記事を参照してください。

Linux VDA では、次の Linux ディストリビューションがサポートされています:

> 重要:
>
> ご利用の OS ベンダーのサポートが期限切れになると、問題の修正において Citrix の機能が制限される場合があります。
> 廃止された、または削除されたプラットフォームについては、「廃止」を参照してください。

- SUSE Linux Enterprise
  - Desktop 12 Service Pack 3
  - Server 12 Service Pack 3
- Red Hat Enterprise Linux
  - Workstation 7.7
  - Workstation 6.10
  - Server 7.7
  - Server 6.10
- CentOS Linux
  - CentOS 7.7
  - CentOS 6.10
- Ubuntu Linux
  - Ubuntu Desktop 18.04
  - Ubuntu Server 18.04
  - Ubuntu Live Server 18.04
  - Ubuntu Desktop 16.04
  - Ubuntu Server 16.04
- Pardus Linux
  - Pardus 17（サポートされる機能範囲について詳しくは、Knowledge Center 記事CTX238492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 のバージョンについては、次の表を参照してください。詳しくは、「XorgModuleABIVersions」を参照してください。

Linux ディストリビューション	Xorg バージョン
RHEL 7.7、CentOS 7.7	1.20
RHEL 6.10、CentOS 6.10	1.17
Ubuntu 18.04	1.19
Ubuntu 16.04	1.18
SUSE 12.3	1.18
Pardus 17	1.19

Ubuntu 16.04 で HWE Xorg server 1.19 を使用しないでください。

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

> 注:
>
> - Linux VDA 1912 LTSR 累積更新プログラム 2（CU2）を CentOS 7.4 にインストールし、Citrix Virtual Apps and Desktops サービスで使用する場合は、VDA の前に次のコンポーネントをインストールしてください:
> - Xorg 1.20.4
> - SELinux policy 3.13.1-268
> - .NET Core Runtime 2.1
> - GNOME 3.28.3 or later
> - Gnome および KDE デスクトップは、SUSE、RHEL、CentOS でサポートされています。Unity デスクトップは、Ubuntu 16.04 でサポートされます。Gnome デスクトップは、Ubuntu 18.04 でサポートされます。1 つまたは複数のデスクトップをインストールする必要があります。

Citrix Virtual Desktops

Linux VDA は、現在サポートされているすべての Citrix Virtual Desktops のバージョンと互換性があります。Citrix Virtual Desktops 製品のライフサイクル、および製品のバージョンごとのサポートが停止される時期について詳しくは、Citrix 製品マトリックスを参照してください。

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

サポートされるホストプラットフォームおよび仮想化環境

- ベアメタルサーバー

- Citrix Hypervisor
- VMware ESX および ESXi
- Microsoft Hyper-V
- Nutanix AHV
- Microsoft Azure Resource Manager
- Amazon Web Services（AWS）
- Google Cloud Platform（GCP）

> ヒント：
>
> サポートされるプラットフォームの一覧については、ベンダーのドキュメントを参照してください。

Active Directory 統合パッケージ

Linux VDA では、以下の Active Directory 統合パッケージまたは製品がサポートされています：

- Samba Winbind
- Quest Authentication Services v4.1 以降
- Centrify DirectControl
- SSSD
- PBIS（RHEL 7 および Ubuntu と互換性があります）

> ヒント：
>
> サポート対象プラットフォームの一覧については、Active Directory 統合パッケージのベンダーが提供しているドキュメントを参照してください。

HDX 3D Pro

Citrix Virtual Apps and Desktops の HDX 3D Pro 機能を使用すると、グラフィック処理装置（GPU）によるハードウェアアクセラレーションで最高の性能を発揮するデスクトップとアプリケーションを配信できます。

ハイパーバイザー

Linux VDA の場合、次のハイパーバイザーが提供する GPU パススルーや GPU 仮想化技術と互換性があります：

- Citrix Hypervisor
- VMware ESX および ESXi
- Nutanix AHV

> 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。

GPU

Linux ディストリビューションがサポートする NVIDIA GPU カードを確認するには、NVIDIA 製品サポートマトリックスに移動し、ハイパーバイザーまたはベアメタル **OS**、ソフトウェア製品の展開、ハードウェアサポート、およびゲスト **OS** サポートの列を確認してください。GPU カード用の最新の vGPU ドライバーをインストールしていることを確認してください。詳しくは、「NVIDIA Virtual GPU Software Supported GPUs」を参照してください。

以下は、GPU パススルーと GPU 仮想化のサポートについてテストした GPU カードです。

GPU パススルーについてテストした GPU：

- NVIDIA GRID - Tesla T4
- NVIDIA GTX750Ti
- NVIDIA GRID - Tesla M60
- NVIDIA GRID - K2
- NVIDIA GRID - Tesla P40
- NVIDIA GRID - Tesla P4
- NVIDIA GRID - Tesla P100

vGPU 用にテストした GPU：

- NVIDIA GRID - Tesla T4
- NVIDIA GRID - Tesla V100
- NVIDIA GRID - Tesla M60
- NVIDIA GRID - Tesla M10
- NVIDIA GRID - Tesla P40
- NVIDIA GRID - Tesla P4
- NVIDIA GRID - Tesla P100

インストールの概要

October 9, 2021

このセクションでは、次の手順について説明します：

- 簡単インストールによる簡易インストール（新規インストールに推奨）
- さまざまな Linux ディストリビューションに基づく手動インストール
- MCS を使用した Linux 仮想マシンの作成
- XenDesktop 7.6 以前のバージョンを対象とした Delivery Controller の構成

簡単インストールによる簡易インストール（推奨）

March 9, 2023

> 重要:
>
> 新規インストールの場合、簡易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、簡単インストールを使用して Linux VDA をインストールおよび構成する方法について説明します。簡単インストールは時間と労力を節約するだけでなく、手動のインストールよりもエラーを減らすことができます。必要なパッケージをインストールして、構成ファイルを自動的にカスタマイズすることで、Linux VDA の実行環境をセットアップできます。

サポートされているディストリビューション

	Winbind	SSSD	Centrify	PBIS
RHEL 7.7	はい	はい	はい	はい
RHEL 6.10	はい	はい	はい	いいえ
CentOS 7.7	はい	はい	はい	はい
CentOS 6.10	はい	はい	はい	いいえ
Ubuntu 18.04	はい	はい	はい	はい
Ubuntu 16.04	はい	はい	はい	はい
SUSE 12.3	はい	いいえ	はい	いいえ

簡単インストールの使用

この機能を使用するには、以下の手順に従ってください:

1. 構成ファイル情報および Linux マシンを準備します。
2. Linux VDA パッケージをインストールします。
   Citrix Virtual Apps and Desktops のダウンロードページにアクセスします。適切なバージョンの Citrix Virtual Apps and Desktops を展開します。［**Components**］をクリックして、使用中の Linux ディストリビューションに対応した Linux VDA パッケージをダウンロードします。
3. Linux VDA のインストールを完了するには Runtime Environment をセットアップします。

手順 **1**：構成ファイル情報および **Linux** マシンの準備

簡単インストールに必要な以下の構成情報を収集します。

- ホスト名 - Linux VDA がインストールされるマシンのホスト名
- ドメインネームサーバーの IP アドレス
- NTP サーバーの IP アドレスまたは文字列名
- ドメイン名 - ドメインの NetBIOS 名
- 領域名 - Kerberos 領域名
- ドメインの完全修飾ドメイン名（FQDN）

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

注意事項

- ワークグループ名はデフォルトではドメイン名です。ご使用の環境内のワークグループをカスタマイズするには、以下の手順に従ってください。

  a. Linux VDA マシンで、/tmp/ctxinstall.conf ファイルを作成します。
  b.「workgroup=<your workgroup>」という行をこのファイルに追加して、変更を保存します。ここで、「your workgroup」はワークグループ名です。

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

  ログ：

  ```
  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.
  <!--NeedCopy-->
  ```

  この問題は、Centrify およびその構成に特有のものです。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の ［管理ツール］を開きます。
  b.［**Active Directory** のサイトとサービス］を選択します。
  c.［サブネット］の適切なサブネットアドレスを追加します。

- Linux VDA 7.16 では、簡単なインストールは、ピュア IPv6 をサポートしています。以下のような前提条件と制限事項があります：

- お使いのマシンがピュア IPv6 ネットワーク経由で必要なパッケージをダウンロードできるように、Linux リポジトリを設定する必要があります。
- Centrify は、ピュア IPv6 ネットワークではサポートされていません。

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

- ドメインに参加させる方式として Centrify を選択する場合、ctxinstall.sh スクリプトでは Centrify パッケージが必要です。ctxinstall.sh で Centrify パッケージを取得する方法は 2 通りあります。
  - 簡単インストールは、インターネットから Centrify パッケージを自動でダウンロードするために役立ちます。ディストリビューションごとの URL は次のとおりです:

    RHEL: wget http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-rhel4-x86_64.tgz?_ga=1.178323680.558673738.1478847956

    CentOS: wget http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-rhel4-x86_64.tgz?_ga=1.186648044.558673738.1478847956

    SUSE: wget http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-suse10-x86_64.tgz?_ga=1.10831088.558673738.1478847956

    Ubuntu: wget http://edge.centrify.com/products/centrify-suite/2016-update-1/installers/centrify-suite-2016.1-deb7-x86_64.tgz?_ga=1.178323680.558673738.1478847956
  - Centrify パッケージをローカルディレクトリから取得します。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 サーバーで/tmp/ctxinstall.conf ファイルが存在していない場合は作成します。
    b.「centrifypkgpath=<path name>」という行をこのファイルに追加します。ここで、「path name」はパス名です。

    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
```



```
 8      756 -r--r--r--. 1 root root   770991 May 13  2016
       centrifydc-ldapproxy-5.3.1-rhel4-x86_64.rpm
 9      268 -r--r--r--. 1 root root   271296 May 13  2016
       centrifydc-nis-5.3.1-rhel4-x86_64.rpm
10      1888 -r--r--r--. 1 root root  1930084 Apr 12  2016
       centrifydc-openssh-7.2p2-5.3.1-rhel4-x86_64.rpm
11      124 -rw-rw-r--. 1 root root   124543 Apr 19  2016
       centrify-suite.cfg
12      0 lrwxrwxrwx. 1 root root       10 Jul  9  2012 install-
       express.sh -> install.sh
13      332 -r-xr-xr--. 1 root root   338292 Apr 10  2016 install
       .sh
14      12 -r--r--r--. 1 root root    11166 Apr  9  2015 release-
       notes-agent-rhel4-x86_64.txt
15      4 -r--r--r--. 1 root root     3732 Aug 24  2015 release-
       notes-da-rhel4-x86_64.txt
16      4 -r--r--r--. 1 root root     2749 Apr  7  2015 release-
       notes-nis-rhel4-x86_64.txt
17      12 -r--r--r--. 1 root root     9133 Mar 21  2016 release-
       notes-openssh-rhel4-x86_64.txt
18  <!--NeedCopy-->
```

- ドメインに参加させる方式として PBIS を選択する場合、ctxinstall.sh スクリプトでは PBIS パッケージが必要です。ctxinstall.sh で PBIS パッケージを取得する方法は 2 通りあります:
  - 簡単インストールは、インターネットから PBIS パッケージを自動でダウンロードするために役立ちます。ディストリビューションごとの URL は次のとおりです:

    RHEL 7/CentOS 7: wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh

    Ubuntu: wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh
  - インターネットから PBIS パッケージの特定のバージョンを取得します。このためには、/opt/Citrix/VDA/sbin/ctxinstall.sh ファイルの「pbisDownloadPath」行を変更して PBIS パッケージの URL を指定します。

    例として、以下のスクリーンショットを参照してください:

手順 **2**: ハイパーバイザーの準備

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

Citrix Hypervisor での時刻同期の修正

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

一部の Linux ディストリビューションでは、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、Citrix Hypervisor の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 0 または 1 を返します:

- 0 - 時刻同期機能が有効になっているため、無効にする必要があります。
- 1 - 時刻同期機能が無効になっています。これ以上の操作は必要ありません。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は必要ありません。

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

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するため、次のようにしてシステムを再起動します:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想ホストでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にする必要があります。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、［統合サービス］を選択します。
3. ［時刻の同期］が選択されていることを確認します。

> 注:
>
> この方法は VMware および Citrix Hypervisor の場合とは異なります。VMware および Citrix Hypervisor では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期を補完することができます。

ESX および **ESXi** での時刻同期の修正

VMware の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーの両方がシステムの時間を同期しようとすることが原因となり問題が発生します。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

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

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

手順 **3**: 前提条件として**.NET Core** ランタイムをインストール

Linux VDA のインストール前に、https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managersの手順に従って.NET Core ランタイムをインストールします。

- 1912 LTSR の初期リリースである CU1 および CU2 の場合は、.NET Core ランタイム 2.1 をインストールします。
- CU3 以降のリリースの場合は、.NET Core ランタイム 3.1 をインストールします。

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

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

手順 **4**: **Linux VDA** パッケージのダウンロード

Citrix Virtual Apps and Desktops のダウンロードページにアクセスします。適切なバージョンの Citrix Virtual Apps and Desktops を展開し、**Components** をクリックして、使用中の Linux ディストリビューションに対応

する Linux VDA パッケージをダウンロードします。

手順 5：Linux VDA パッケージのインストール

Linux VDA の環境をセットアップするには、次のコマンドを実行します。

RHEL および CentOS ディストリビューション

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

Ubuntu ディストリビューション

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

SUSE ディストリビューションの場合：

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

手順 6：NVIDIA GRID ドライバーのインストール

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

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

- Citrix Hypervisor
- VMware ESX

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

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

手順 7：Runtime Environment をセットアップしてインストールを完了する

> 注：

> ランタイム環境をセットアップする前に、`en_US.UTF-8`ロケールがインストールされていることを確認します。OS にこのロケールがない場合は、`sudo locale-gen en_US.UTF-8`コマンドを実行します。

Linux VDA パッケージのインストール後、ctxinstall.sh スクリプトを使用して、実行環境を構成します。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:
>
> サイズが 27MB を超える.NET Core ランタイムをダウンロード中、簡易インストールが応答しないように見える場合があります。ダウンロードの進行状況については、/var/log/ctxinstall.log で確認します。

手動構成を実行するには、次のコマンドを実行し、プロンプトごとに関連パラメーターを入力します。

```
sudo /opt/Citrix/VDA/sbin/ctxinstall.sh
<!--NeedCopy-->
```

サイレントモード:

サイレントモードで簡単インストールを使用するには、ctxinstall.sh を実行する前に以下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指定します。
- **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 領域名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis** - Active Directory の統合方式を指定します。
- **CTX_EASYINSTALL_USERNAME=domain-user-name** - ドメインに参加させるために使用されるドメインユーザーの名前を指定します。
- **CTX_EASYINSTALL_PASSWORD=password** - ドメインに参加させるために使用されるドメインユーザーのパスワードを指定します。

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 では、DNS CNAME レコードを使用して、Delivery Controller 名を指定することができます。
- **CTX_XDL_DDC_LIST=’list-ddc-fqdns’** - Linux VDA には、Delivery Controller の登録に使用する Delivery Controller の完全修飾ドメイン名（FQDN）のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME を指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** - Linux VDA は、TCP/IP ポート経由で Delivery Controller と通信します。
- **CTX_XDL_REGISTER_SERVICE=Y | N** - Linux VDA サービスは、マシンの起動後に開始します。

- **CTX_XDL_ADD_FIREWALL_RULES = Y | N** - Linux VDA サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux VDA 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。
- **CTX_XDL_HDX_3D_PRO=Y | N** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連のグラフィックアクセラレーションテクノロジです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ（シングルセッション）モード用に構成されます（つまり、CTX_XDL_VDI_MODE=Y となります）。
- **CTX_XDL_VDI_MODE = Y | N** - 専用デスクトップ配信モデル（VDI）またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境の場合は、値を Y に設定します。
- **CTX_XDL_SITE_NAME = dns-name** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。不要な場合は、**<none>** に設定します。
- **CTX_XDL_LDAP_LIST=’list-ldap-servers’** - Linux VDA は、DNS を照会して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合は、LDAP の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。たとえば、ad1.mycompany.com:389 となります。不要な場合は、**<none>** に設定します。
- **CTX_XDL_SEARCH_BASE = search-base-set** - Linux VDA は、Active Directory ドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます（例：OU=VDI,DC=mycompany,DC=com）。不要な場合は、**<none>** に設定できます。
- **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 グループポリシーにより構成されます。Linux VDA は AD グループポリシーをサポートしていません。代わりに、セミコロンで区切られた FAS サーバーの一覧を使用できます。シーケンスは、AD グループポリシーで設定したものと同じである必要があります。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスの順番は変更しません。
- **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 Core ランタイム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_START_SERVICE= Y | N** - 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。

設定されていないパラメーターがあるとインストールは対話モードにロールバックし、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使用して既に設定されている場合、ctxinstall.sh スクリプトは、.NET Core ランタイムをインストールするためのパスの入力を要求します。

サイレントモードでは、次のコマンドを実行して環境変数を設定してから 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_START_SERVICE=Y | N

sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh
<!--NeedCopy-->
```

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

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

```
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_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

手順 8：XDPing の実行

Linux VDA 環境での一般的な構成の問題を確認するために、コマンドラインユーティリティである Linux XDPing ツールが提供されています。XDPing パッケージは、サポート対象の Linux ディストリビューションを実行している任意のマシンにインストールできます。XDPing では、Linux VDA パッケージをマシンにインストールする必要はありません。このツールについて詳しくは、Knowledge Center の記事CTX202015を参照してください。

手順 9：Linux VDA の実行

Linux VDA の起動：

Linux VDA サービスを起動するには、次のコマンドを実行します：

```
sudo /sbin/service ctxhdx start

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の停止：

Linux VDA サービスを停止するには、次のコマンドを実行します：


```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx stop
<!--NeedCopy-->
```

> 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx restart

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の状態の確認:

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

```
sudo /sbin/service ctxvda status

sudo /sbin/service ctxhdx status
<!--NeedCopy-->
```

手順 **10**: **Citrix Virtual Apps** または **Citrix Virtual Desktops** でのマシンカタログの作成

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

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

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

> 注:
>
> Citrix Studio の以前のバージョンは、「Linux OS」という概念をサポートしていませんでした。ただし、[**Windows** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **OS**] オプションを選択すると、マシンごとに単一ユーザーの配信モデルが暗黙的に選択されます。
>
> ヒント:
>
> マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

手順 **11**: **Citrix Virtual Apps** または **Citrix Virtual Desktops** でのデリバリーグループの作成

デリバリーグループを作成し、Linux VDA マシンを含むマシンカタログを追加する手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「デリバリーグループの作成」を参照してください。

Linux VDA マシンカタログを含むデリバリーグループを作成する場合は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

> 重要:
>
> アプリケーションの公開は、Linux VDA バージョン 1.4 以降でサポートされています。ただし、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

トラブルシューティング

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

SSSD を使用してドメインに参加できない

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


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

/var/log/xdl/vda.log:

```
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.
<!--NeedCopy-->
```

/var/log/messages:

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

この問題を解決するには、次の手順に従います:

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

Ubuntu のデスクトップセッションで灰色の画面が表示される

セッションを起動すると、空のデスクトップでブロックされる問題が発生します。また、マシンのコンソールでも、ローカルユーザーアカウントを使用してログオンすると灰色の画面が表示されます。

この問題を解決するには、次の手順に従います:

1. `sudo apt-get update`コマンドを実行します。
2. `sudo apt-get install unity lightdm`コマンドを実行します。
3. 次の行を/etc/lightdm/lightdm.conf に追加します:
   greeter-show-manual-login=true

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

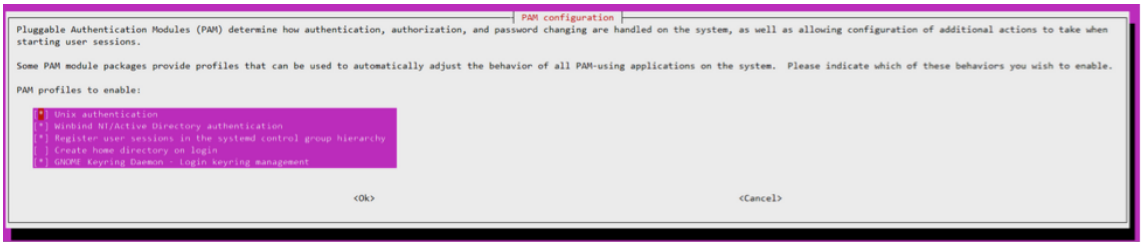
/var/log/xdl/hdx.log:

```
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.
<!--NeedCopy-->
```

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

この問題を解決するには、次の手順に従います:

1. コマンドラインで、**pam-auth-update** を入力します。
2. 表示されたダイアログで、［ログイン時にホームディレクトリを作成する］が選択されていることを確認します。



dbus エラーによりセッションを起動または終了できない

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

```
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.
<!--NeedCopy-->
```

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

```
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
<!--NeedCopy-->
```

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

SELinux で **SSHD** がホームディレクトリにアクセスできない

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/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
<!--NeedCopy-->
```

この問題を解決するには、次の手順に従います:

1. /etc/selinux/config に次の変更を加えることで、SELinux を無効にします。

   SELINUX=disabled

2. VDA を再起動します。

Linux Virtual Delivery Agent for RHEL/CentOS の手動インストール

December 13, 2022

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

手順 **1**: **VDA** をインストールする **RHEL 7/CentOS 7**、**RHEL 6/CentOS 6** の準備

手順 **1a**: ネットワーク構成の確認

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

手順 **1b**: ホスト名の設定

マシンのホスト名が確実に正しく報告されるようにするには、**/etc/hostname** ファイル（RHEL 7 および CentOS 7 の場合）または**/etc/sysconfig/network** ファイル（RHEL 6 および CentOS 6 の場合）を変更してマシンのホ

スト名のみを記述します。

```
hostname
```

手順 **1c**：ホスト名へのループバックアドレスの割り当て

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

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

例：

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

ファイル内の他のエントリから、**hostname-fqdn** または **hostname** に対するその他の参照を削除します。

> 注：
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字以内である必要があります。
>
> ヒント：
>
> a～z、A～Z、0～9、およびハイフン（-）の文字のみ使用してください。アンダースコア（_）、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了したりしないでください。このルールは、Delivery Controller のホスト名にも適用されます。

手順 **1d**：ホスト名の確認

次のコマンドで、ホスト名が正しく設定されていることを確認します：

```
hostname
<!--NeedCopy-->
```

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

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します：

```
hostname -f
<!--NeedCopy-->
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

手順 **1e**：名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

ping delivery-controller-fqdn
<!--NeedCopy-->
```

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

手順 **1f**：時刻同期の構成

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

RHEL 6.x 以前のリリースでは、時刻同期に NTP デーモン（`ntpd`）を使用しています。一方、RHEL 7.x のデフォルト環境では、新しい Chrony デーモン（`chronyd`）を代わりに使用しています。この 2 つのサービスの構成と操作手順は類似しています。

NTP サービスの構成（**RHEL 6/CentOS 6** のみ）　ルートユーザーとして **/etc/ntp.conf** を編集し、次のように各リモートタイムサーバーに対応するサーバーエントリを追加します：

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

server peer2-fqdn-or-ip-address iburst
<!--NeedCopy-->
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応するサーバーエントリを追加します。

ループバック IP アドレス、localhost、パブリックサーバーの ***.pool.ntp.org** エントリなど、一覧にあるその他の **server** エントリを削除します。

変更を保存してから、次のコマンドで NTP デーモンを再起動します：

```
sudo /sbin/service ntpd restart
<!--NeedCopy-->
```


Chrony サービスの構成（**RHEL 7/CentOS 7** のみ）　ルートユーザーとして **/etc/chrony.conf** を編集し、次のように各リモートタイムサーバーに対応するサーバーエントリを追加します：

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

server peer2-fqdn-or-ip-address iburst
<!--NeedCopy-->
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応するサーバーエントリを追加します。

ループバック IP アドレス、localhost、パブリックサーバーの ***.pool.ntp.org** エントリなど、一覧にあるその他の server エントリを削除します。

変更を保存してから、次のコマンドで Chrony デーモンを再起動します：

```
sudo /sbin/service chronyd restart
<!--NeedCopy-->
```

手順 **1g**：**OpenJDK** のインストール

Linux VDA は OpenJDK に依存しています。通常、Runtime Environment は、オペレーティングシステムの一部としてインストールされています。

正しいバージョンを確認します。

```
sudo yum info java-1.8.0-openjdk
<!--NeedCopy-->
```

事前にパッケージされた OpenJDK は、以前のバージョンである可能性があります。必要に応じて、次のコマンドで最新バージョンに更新します：

```
sudo yum -y update java-1.8.0-openjdk
<!--NeedCopy-->
```

新しいシェルを開き、次のコマンドで Java のバージョンを確認します：

```
java -version
<!--NeedCopy-->
```

> ヒント：
>
> Delivery Controller の登録で失敗しないために、OpenJDK 1.8.0 のみをインストールするようにしてください。その他のバージョンの Java は、システムからすべて削除します。

手順 **1h**：**PostgreSQL** のインストール

Linux VDA には、PostgreSQL 8.4 以降（RHEL 6 の場合）または PostgreSQL 9.2 以降（RHEL 7 の場合）のいずれかが必要です。

次のパッケージをインストールします：

```
sudo yum -y install postgresql-server

sudo yum -y install postgresql-jdbc
<!--NeedCopy-->
```

データベースを初期化し、マシンの起動時にサービスが確実に開始されるようにするには、次に示すインストール後の手順が必要です。この操作により、**/var/lib/pgsql/data** にデータベースファイルが作成されます。このコマンドは、PostgreSQL 8 と 9 では異なります：

- RHEL 7 のみ：PostgreSQL 9

```
sudo postgresql-setup initdb
<!--NeedCopy-->
```

- RHEL 6 のみ：PostgreSQL 8

```
sudo /sbin/service postgresql initdb
<!--NeedCopy-->
```

手順 **1i**：**PostgreSQL** の起動

マシンの起動時にサービスを開始し、直ちにサービスを開始します：

- RHEL 7 のみ：PostgreSQL 9

```
sudo systemctl enable postgresql

sudo systemctl start postgresql
<!--NeedCopy-->
```

- RHEL 6 のみ：PostgreSQL 8

```
sudo /sbin/chkconfig postgresql on

sudo /sbin/service postgresql start
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```


次のように **psql** コマンドラインユーティリティを使用して、データディレクトリが設定されていることを確認します:

```
sudo -u postgres psql -c 'show data_directory'
<!--NeedCopy-->
```

> 重要:
>
> このリリースでは、`gperftools-libs`に新しい依存関係が追加されていますが、この依存関係は元のリポジトリには存在していません。`sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm`コマンドを使用して、リポジトリを追加します。
> RHEL 6/CentOS 6 のみが影響を受けます。Linux VDA パッケージをインストールする前に、このコマンドを実行します。

手順 **2**: ハイパーバイザーの準備

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

Citrix Hypervisor での時刻同期の修正

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

一部の Linux ディストリビューションでは、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、Citrix Hypervisor の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 0 または 1 を返します:

- 0 - 時刻同期機能が有効になっているため、無効にする必要があります。
- 1 - 時刻同期機能が無効になっています。これ以上の操作は必要ありません。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は必要ありません。

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


```
sudo echo 1 > /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するため、次のようにしてシステムを再起動します:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想ホストでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にする必要があります。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、［統合サービス］を選択します。
3. ［時刻の同期］が選択されていることを確認します。

> 注:
>
> この方法は VMware および Citrix Hypervisor の場合とは異なります。VMware および Citrix Hypervisor では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期を補完することができます。

ESX および **ESXi** での時刻同期の修正

VMware の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーの両方がシステムの時間を同期しようとすることが原因となり問題が発生します。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

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

1. vSphere Client を開きます。
2. Linux 仮想マシンの設定を編集します。
3. ［仮想マシンのプロパティ］ダイアログボックスで、［オプション］タブをクリックします。

4. ［**VMware Tools**］を選択します。
5. ［詳細］ボックスで、［ホストとゲスト時刻を同期］チェックボックスをオフにします。

手順 3：Linux 仮想マシン（VM）を Windows ドメインに追加

Linux VDA は、Linux マシンを Active Directory（AD）ドメインに追加するさまざまな方法をサポートします。

- Samba Winbind
- Quest Authentication Services
- Centrify DirectControl
- SSSD
- PBIS（RHEL 7 とのみ互換性があります）

選択した方法の手順に従います。

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

Samba Winbind

次のようにして、必要なパッケージをインストールまたは更新します：

```
sudo yum -y install samba-winbind samba-winbind-clients krb5-workstation authconfig oddjob-mkhomedir
<!--NeedCopy-->
```

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

```
sudo /sbin/chkconfig winbind on
<!--NeedCopy-->
```

Winbind 認証の構成　次のようにして、Winbind を使用した Kerberos 認証用にマシンを構成します：

```
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
<!--NeedCopy-->
```


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

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

`--enablekrb5kdcdns --enablekrb5realmdns`

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

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

```
kerberos method = secrets and keytab
winbind refresh tickets = true
```

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

Windows ドメインへの参加　ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です:

```
1 sudo net ads join REALM -U user
2 <!--NeedCopy-->
```

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

Winbind 用の **PAM** の構成　デフォルトでは、Winbind PAM モジュール（pam_winbind）の構成で、Kerberos チケットキャッシュとホームディレクトリの作成が有効になっていません。**/etc/security/pam_winbind.conf** を開いて、[Global] セクションで次のとおりにエントリを追加または変更します:

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

各設定の先頭のセミコロンは必ず削除します。これらを変更するには、次のようにして Winbind デーモンを再起動する必要があります:

```
1 sudo /sbin/service winbind restart
2 <!--NeedCopy-->
```


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

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

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

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

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

```
sudo net ads testjoin
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

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

```
sudo klist -ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の`kinit`コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します：

```
sudo kinit -k MACHINE$@REALM
<!--NeedCopy-->
```

マシン名と領域名は大文字で指定する必要があります。ドル記号（$）は、シェルによる置き換えを防ぐためにバックスラッシュ（\）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist
<!--NeedCopy-->
```

次のコマンドを使用して、マシンアカウントの詳細を調査します：

```
sudo net ads status
<!--NeedCopy-->
```


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

```
wbinfo --krb5auth=domain\username%password
<!--NeedCopy-->
```

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

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

```
ssh localhost -l domain\username
id -u
<!--NeedCopy-->
```

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

```
klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6: Linux VDA のインストール」に進みます。

Quest Authentication Services

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

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

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

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

Linux VDA での **Quest** の構成

SELinux ポリシー適用の回避策　デフォルトの RHEL 環境では、SELinux が完全に適用されています。この適用により、Quest が使用する Unix ドメインソケットの IPC のメカニズムに干渉し、ドメインユーザーのログオンを妨げます。

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

```
SELINUX=permissive
```

この変更にはマシンの再起動が必要です:

```
reboot
<!--NeedCopy-->
```

> 重要:
>
> この設定は注意して使用してください。SELinux ポリシーの適用を無効にした後に再度有効にすると、ルートユーザーやその他のローカルユーザーであっても、完全にロックアウトされてしまう可能性があります。

VAS デーモンの構成　次のように Kerberos チケットの自動更新を有効にして、切断する必要があります。認証（オフラインログオン）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
<!--NeedCopy-->
```

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

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

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

```



```
sudo /opt/quest/bin/vastool configure nss
<!--NeedCopy-->
```

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

```
sudo /opt/quest/bin/vastool -u user join domain-name
<!--NeedCopy-->
```

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

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

```
sudo /opt/quest/bin/vastool info domain
<!--NeedCopy-->
```

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下のエラーが表示されます:

```
ERROR: No domain could be found.
ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm
default_realm not configured in vas.conf. Computer may not be joined
to domain
```

ユーザー認証の確認　PAM を使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことがないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\username
id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

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

```
/opt/quest/bin/vastool klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。


```
exit
<!--NeedCopy-->
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6: Linux VDA のインストール」に進みます。

Centrify DirectControl

Windows ドメインへの参加　Centrify DirectControl Agent がインストールされている場合、次のように Centrify の`adjoin`コマンドを使用して、Linux マシンを Active Directory ドメインに追加します:

```
su -
adjoin -w -V -u user domain-name
<!--NeedCopy-->
```

user パラメーターは、コンピューターを Active Directory ドメインに追加する権限を持つ任意の Active Directory ドメインユーザーです。**domain-name** は、Linux マシンを追加するドメインの名前です。

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

```
su -
adinfo
<!--NeedCopy-->
```

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

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
adinfo --sysinfo all
adinfo -diag
<!--NeedCopy-->
```

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

```
adinfo --test
<!--NeedCopy-->
```

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

SSSD

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

SSSD を RHEL および CentOS でセットアップするには、次の作業を行います。

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

必要なソフトウェア　Active Directory プロバイダーは、SSSD Version 1.9.0 で初めて導入されました。古いバージョンを使用している場合は、Configuring the LDAP provider with Active Directoryの指示に従ってください。

次の環境については、このドキュメントに記載した指示を使用したテストおよび検証を行っています:

- RHEL 7.7 以降
- CentOS 7.7 以降

ドメインに参加してホストの **keytab** を作成　SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する Active Directory のクライアント機能が提供されていません。代わりに`adcli`、`realmd`または`Samba`を使用できます。

このセクションでは、`Samba`によるアプローチについてのみ説明します。`adcli`および`realmd`に関しては、RHEL または CentOS のドキュメントを参照してください。SSSD を構成する前に、以下の手順に従う必要があります。

次のようにして、必要なパッケージをインストールまたは更新します:

```
sudo yum -y install krb5-workstation authconfig oddjob-mkhomedir samba-common-tools
<!--NeedCopy-->
```

Linux クライアントで、適切に構成されたファイルを使用します。

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

Samba および Kerberos 認証用にマシンを構成します:

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


ここで、**REALM** は大文字の Kerberos 領域名で、**domain** は Active Directory ドメインの短い NetBIOS 名です。

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

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

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

```
kerberos method = secrets and keytab
```

Windows ドメインに参加します。ドメインコントローラーに到達できることと、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントがあることを確認します:

```
sudo net ads join REALM -U user
<!--NeedCopy-->
```

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

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

- Linux VDA に **sssd-ad** パッケージをインストールします。
- さまざまなファイルに設定の変更を行います（sssd.conf など）。
- **sssd** サービスを開始します。

sssd.conf の設定の例（必要に応じて追加の設定を行うことができます）:

```
[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
<!--NeedCopy-->
```

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

ファイルの所有権およびアクセス権限を sssd.conf で設定します:

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

NSS/PAM の構成　RHEL/CentOS:

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

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

sudo service sssd start

sudo chkconfig sssd on
<!--NeedCopy-->
```

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

```
sudo klist -ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します:

```
sudo kinit - k MACHINE$@REALM
<!--NeedCopy-->
```


マシン名と領域名は大文字で指定する必要があります。ドル記号（**$**）は、シェルによる置き換えを防ぐためにバックスラッシュ（****）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist
<!--NeedCopy-->
```

ユーザー認証の確認　**getent** コマンドを使用して、ログオン形式がサポートされていること、および NSS が機能するかを確認します：

```
sudo getent passwd DOMAIN\username
<!--NeedCopy-->
```

DOMAIN パラメーターは短い形式のドメイン名です。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認します。

サポートされているログオン形式は次の通りです：

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

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

```
sudo ssh localhost  - l DOMAIN\username

id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_{
 uid }

<!--NeedCopy-->
```

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

```
klist
<!--NeedCopy-->
```

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

PBIS

必要な **PBIS** パッケージをダウンロードする　例:

```
wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/
    pbis-open-8.8.0.506.linux.x86_64.rpm.sh
<!--NeedCopy-->
```

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-8.8.0.506.linux.x86_64.rpm.sh
<!--NeedCopy-->
```

PBIS インストールスクリプトを実行する　例:

```
sh pbis-open-8.8.0.506.linux.x86_64.rpm.sh
<!--NeedCopy-->
```

Windows ドメインへの参加　ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です:

```
/opt/pbis/bin/domainjoin-cli join domain-name user
<!--NeedCopy-->
```

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

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

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

```
/opt/pbis/bin/domainjoin-cli query
<!--NeedCopy-->
```

マシンがドメインに参加している場合、このコマンドは現在の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

ユーザー認証の確認　PAM を使用した PBIS のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことがないドメインユーザーアカウントを使用して Linux VDA にログオンします。


```
ssh localhost -l domain\user

id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

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**: 前提条件として**.NET Core** ランタイムをインストール

Linux VDA のインストール前に、https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managersの手順に従って.NET Core ランタイムをインストールします。

- 1912 LTSR の初期リリースである CU1 および CU2 の場合は、.NET Core ランタイム 2.1 をインストールします。
- CU3 以降のリリースの場合は、.NET Core ランタイム 3.1 をインストールします。

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

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

手順 **5**: **Linux VDA** パッケージのダウンロード

Citrix Virtual Apps and Desktops サービスのダウンロードページに移動します。適切なバージョンの Citrix Virtual Apps and Desktops を展開し、**Components** をクリックして、使用中の Linux ディストリビューションに対応する Linux VDA パッケージをダウンロードします。

手順 **6**: **Linux VDA** のインストール

新規にインストールするか、最新の 2 バージョンと 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。

新規インストール手順

1. （オプション）古いバージョンのアンインストール

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

   a) 次のコマンドで、Linux VDA サービスを停止します：

   ```
   sudo /sbin/service ctxvda stop

   sudo /sbin/service ctxhdx stop
   <!--NeedCopy-->
   ```

   > 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

   b) 次のコマンドで、パッケージをアンインストールします：

   ```
   sudo rpm -e XenDesktopVDA
   <!--NeedCopy-->
   ```

   > 注：
   >
   > コマンドを実行するには、フルパスが必要です。代わりに、システムパスに**/opt/Citrix/VDA/sbin** および**/opt/Citrix/VDA/bin** を追加することもできます。

2. Linux VDA のインストール

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

     RHEL 7/CentOS 7 の場合：

     ```
     sudo yum install -y XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
     <!--NeedCopy-->
     ```

     RHEL 6/CentOS 6 の場合：

     ```
     sudo yum install -y XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
     <!--NeedCopy-->
     ```

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

     RHEL 7/CentOS 7 の場合：


```
sudo rpm -i XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
<!--NeedCopy-->
```

RHEL 6/CentOS 6 の場合：

```
sudo rpm -i XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
<!--NeedCopy-->
```

RPM 依存関係一覧（**RHEL 7/CentOS 7** の場合）：

```
postgresql-server >= 9.2

postgresql-jdbc >= 9.2

java-1.8.0-openjdk >= 1.8.0

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

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

motif >= 2.3.4

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
```



```

openldap >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.6.0

gperftools-libs >= 2.4

rpmlib(FileDigests) <= 4.6.0-1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

pmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsXz) <= 5.2-1
<!--NeedCopy-->
```

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

RPM 依存関係一覧（**RHEL 6/CentOS 6** の場合）:

```
postgresql-jdbc >= 8.4

postgresql-server >= 8.4

java-1.8.0-openjdk >= 1.8.0

ImageMagick >= 6.5.4.7

GConf2 >= 2.28.0

system-config-firewall-base >= 1.2.27

policycoreutils-python >= 2.0.83

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.0.9

ConsoleKit >= 0.4.1

dbus >= 1.2.24

dbus-x11 >= 1.2.24

```



```
libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

openmotif >= 2.3.3

pam >= 1.1.1

util-linux-ng >= 2.17.2

bash >= 4.1

findutils >= 4.4

gawk >= 3.1

sed >= 4.2

cups >= 1.4.0

foomatic >= 4.0.0

openldap >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.7

python-requests >= 2.6.0

gperftools-libs >= 2.0

rpmlib(FileDigests) <= 4.6.0-1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

rpmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsXz) <= 5.2-1
<!--NeedCopy-->
```

> 注:
>
> 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、セッションの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が目的です。詳しくは、「セッションのシャドウ」を参照してください。

既存のインストールのアップグレード手順

最新の 2 バージョンと 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。

- Yumを使用してアップグレードするには:

  RHEL 7/CentOS 7 の場合:

  ```
  sudo yum install -y XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
  <!--NeedCopy-->
  ```

  RHEL 6/CentOS 6 の場合:

  ```
  sudo yum  install -y XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
  <!--NeedCopy-->
  ```

- RPM Package Manager を使用してアップグレードするには:

  RHEL 7/CentOS 7 の場合:

  ```
  sudo rpm -U XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
  <!--NeedCopy-->
  ```

  RHEL 6/CentOS 6 の場合:

  ```
  sudo rpm -U XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
  <!--NeedCopy-->
  ```

> 重要:
>
> ソフトウェアをアップグレードした後、Linux VDA マシンを再起動してください。

手順 7: NVIDIA GRID ドライバーのインストール

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

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

- Citrix Hypervisor
- VMware ESX

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

1. ゲスト VM がシャットダウンされていることを確認します。
2. XenCenter で、GPU を VM に割り当てます。
3. 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
<!--NeedCopy-->
```

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

> 注:
>
> GPU ドライバーのインストール時は、すべての質問でデフォルト（「いいえ」）を選択してください。
>
> 重要:
>
> GPU パススルーを有効にすると、XenCenter を利用して Linux 仮想マシンにアクセスできなくなります。SSH を使用して接続します。

nvidia-smi

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

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

次のコマンドで、カードに適切な構成を設定します:

etc/X11/ctx-nvidia.sh

高い解像度やマルチモニター機能を利用するには、有効な NVIDIA ライセンスが必要です。このライセンスを申請するには、『GRID Licensing Guide.pdf - DU-07757-001 September 2015』の製品ドキュメントの指示に従ってください。

手順 8：Linux VDA の構成

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

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

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
<!--NeedCopy-->
```

質問に回答する構成

次のようにして、質問に回答する手動構成を実行します：

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

自動化された構成

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

サポートされる環境変数には次のようなものがあります：

- **CTX_XDL_SUPPORT_DDC_AS_CNAME = Y | N** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定できます。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 には、Delivery Controller の登録に使用する Delivery Controller の完全修飾ドメイン名（FQDN）のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME 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）を使用して、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します。デフォルトでは値は Y に設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES = Y | N** - Linux Virtual Desktop サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。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 には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの

Active Directory 統合ツールから指定します。次に示す、サポートされている Active Directory 統合方法のうち、使用するものを指定します：

  - 1 - Samba Winbind
  - 2 - Quest Authentication Service
  - 3 - Centrify DirectControl
  - 4 - SSSD
  - 5 - PBIS

- **CTX_XDL_HDX_3D_PRO=Y | N** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ（シングルセッション）モード用に構成されます（つまり、CTX_XDL_VDI_MODE=Y となります）。
- **CTX_XDL_VDI_MODE = Y | N** - 専用デスクトップ配信モデル（VDI）またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境では、この変数を Y に設定します。デフォルトでは N に設定されています。
- **CTX_XDL_SITE_NAME = dns-name** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数は、デフォルトでは **<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 の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。たとえば、ad1.mycompany.com:389 となります。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_SEARCH_BASE = search-base-set** - Linux VDA は、Active Directory ドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます（例：OU=VDI,DC=mycompany,DC=com）。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_FAS_LIST=‘list-fas-servers’** - フェデレーション認証サービス（FAS）サーバーは、AD グループポリシーにより構成されます。Linux VDA は AD グループポリシーをサポートしていないため、代わりにセミコロンで区切られた FAS サーバーの一覧を使用できます。シーケンスは、AD グループポリシーで設定したものと同じである必要があります。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスのシーケンスを変更せずに維持します。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** - 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Core ランタイム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_START_SERVICE = Y | N** - Linux VDA 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。デフォルトでは Y に設定されています。

次のようにして、環境変数を設定し、構成スクリプトを実行します：

```
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_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

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

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

```
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_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

構成変更の削除

シナリオによっては、Linux VDA パッケージをアンインストールしないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します:

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
<!--NeedCopy-->
```

構成変更を削除するには:

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
<!--NeedCopy-->
```

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

構成ログ

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

Linux VDA サービスを再起動し、変更を反映させます。

手順 9: XDPing の実行

Linux VDA 環境での一般的な構成の問題を確認するために、コマンドラインユーティリティである Linux `XDPing` ツールが提供されています。`XDPing`パッケージは、サポートされている Linux ディストリビューションを実行している任意のマシンにインストールできます。`XDPing`は、Linux VDA パッケージをマシンにインストールする必要はありません。このツールについて詳しくは、Knowledge Center の記事CTX202015を参照してください。

手順 **10**：**Linux VDA** の実行

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

Linux VDA の起動：

Linux VDA サービスを起動するには、次のコマンドを実行します：

```
sudo /sbin/service ctxhdx start

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の停止：

Linux VDA サービスを停止するには、次のコマンドを実行します：

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx stop
<!--NeedCopy-->
```

> 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

Linux VDA の再起動：

Linux VDA サービスを再起動するには、次のコマンドを実行します：

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx restart

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の状態の確認：

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

```
sudo /sbin/service ctxvda status

sudo /sbin/service ctxhdx status
<!--NeedCopy-->
```


手順 **11**：**Citrix Virtual Apps** または **Citrix Virtual Desktops** でのマシンカタログの作成

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

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

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

> 注：
>
> Citrix Studio の以前のバージョンは、「Linux OS」という概念をサポートしていませんでした。ただし、［**Windows** サーバー **OS**］オプションまたは ［サーバー **OS**］オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。［**Windows** デスクトップ **OS**］オプションまたは［デスクトップ **OS**］オプションを選択すると、マシンごとに単一ユーザーの配信モデルが暗黙的に選択されます。
>
> ヒント：
>
> マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

手順 **12**：**Citrix Virtual Apps** または **Citrix Virtual Desktops** でのデリバリーグループの作成

デリバリーグループを作成し、Linux VDA マシンを含むマシンカタログを追加する手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「デリバリーグループの作成」を参照してください。

Linux VDA マシンカタログを含むデリバリーグループを作成する場合は、次の制約があります：

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

> 重要：
>
> アプリケーションの公開は、Linux VDA バージョン 1.4 以降でサポートされています。ただし、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

Linux Virtual Delivery Agent for SUSE の手動インストール

December 13, 2022

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

手順 **1**: インストールの準備

手順 **1a**: **YaST** ツールの起動

SUSE Linux Enterprise YaST ツールを使用して、オペレーティングシステムのすべての要素を構成します。

テキストベースの YaST ツールを起動する方法

```
su -

yast
<!--NeedCopy-->
```

代わりに、UI ベースの YaST ツールを起動する方法

```
su -

yast2 &
<!--NeedCopy-->
```

手順 **1b**: ネットワークの構成

以降のセクションでは、Linux VDA で使用するさまざまなネットワーク設定およびサービスの構成方法に関する情報について説明します。ネットワークの構成は、Network Manager などの他の方法ではなく、YaST ツールで実行する必要があります。次の手順は、UI ベースの YaST ツールを使用することが前提となっています。テキストベースの YaST ツールも使用できますが、ナビゲーション方法が異なり、ここでは説明していません。

ホスト名と **DNS** の構成

1. YaST の［ネットワーク設定］を開きます。
2. SLED 12 のみ:［グローバルオプション］タブで、［ネットワークのセットアップ方法］を ［**Wicked** サービス］に変更します。
3. ［ホスト名/**DNS**］タブを開きます。

4. ［**DHCP** でホスト名を変更する］チェックボックスをオフにします。
5. ［ホスト名をループバック **IP** に割り当てる］チェックボックスをオンにします。
6. 以下を編集してネットワーク設定に反映させます。

   - ホスト名–マシンの DNS ホスト名を追加します。
   - ドメイン名–マシンの DNS ドメイン名を追加します。
   - ネームサーバー–DNS サーバーの IP アドレスを追加します。通常は AD ドメインコントローラーの IP アドレスです。
   - ［ドメイン検索］一覧–DNS ドメイン名を追加します。

> 注:
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字以内である必要があります。
>
> ヒント:
>
> a～z、A～Z、0～9、およびハイフン（-）の文字のみ使用してください。アンダースコア（_）、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了したりしないでください。このルールは、Delivery Controller のホスト名にも適用されます。

マルチキャスト **DNS** の無効化　SLED でのみ、デフォルトの設定でマルチキャスト DNS（mDNS）が有効であるため、名前解決の結果に不整合が発生する場合があります。SLES の場合、mDNS はデフォルトでは有効化されていないため、特に操作を行う必要はありません。

mDNS を無効にするには、**/etc/nsswitch.conf** を編集して、以下を含む行を変更します:

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

変更後:

```
hosts: files dns
```

ホスト名の確認　次のコマンドで、ホスト名が正しく設定されていることを確認します:

```
hostname
<!--NeedCopy-->
```

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

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します:

```
hostname -f
<!--NeedCopy-->
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

名前解決とサービス到達可能性の確認　次のコマンドで、完全修飾ドメイン名が解決できることと、ドメインコントローラーと Delivery Controller から ping に応答があることを確認します:

```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

ping delivery-controller-fqdn
<!--NeedCopy-->
```

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

手順 **1c**: **NTP** サービスの構成

VDA、Delivery Controller、ドメインコントローラーの間で正確な時刻同期を維持することが重要です。仮想マシンとして Linux VDA をホストすると、時刻が不正確になる問題が発生する可能性があります。したがって、リモート NTP サービスを使用して時刻を維持することをお勧めします。次のように、デフォルト NTP 設定にいくつかの変更が必要な場合があります。

1. YaST の［NTP 環境設定］を開いて、［一般的な設定］タブをクリックします。
2. ［NTP デーモンを起動する］セクションで、［今すぐ開始し、システム起動時に開始するよう設定］をクリックします。
3. 表示されている場合は、［規律に従わないローカル時計（**LOCAL**）］項目を選択し、［削除］をクリックします。
4. ［追加］をクリックして、NTP サーバーのエントリを追加します。
5. ［サーバーの種類］を選択して、［次へ］をクリックします。
6. ［アドレス］フィールドに、NTP サーバーの DNS 名を入力します。このサービスは、通常 Active Directory ドメインコントローラーでホストされます。
7. ［オプション］フィールドは変更しません。
8. ［テスト］をクリックして、NTP サービスに到達できるかどうかを確認します。
9. 一連のウィンドウで ［**OK**］をクリックして、変更を保存します。

> 注:
>
> SLES 12 の実装では、AppArmor ポリシーに関する SUSE の既知の問題が原因で、NTP デーモンが起動に失敗することがあります。詳しくは、解決方法に従ってください。

手順 **1d**: **Linux VDA** に依存するパッケージのインストール

SUSE Linux Enterprise 用の Linux VDA ソフトウェアは、次のパッケージに依存しています。

- PostgreSQL

  - SLED/SLES 12：バージョン 9.3 以降

- OpenJDK 1.8.0
- Open Motif Runtime Environment 2.3.1 以降
- CUPS
  - SLED/SLES 12：バージョン 1.6.0 以降
- Foomatic フィルター
  - SLED/SLES 12：バージョン 1.0.0 以降
- ImageMagick
  - SLED/SLES 12：バージョン 6.8 以降

リポジトリの追加　次のように、必要なパッケージの中には、一部の SUSE Linux Enterprise リポジトリでは入手できないものがあります。

- SLED 12：PostgreSQL は、SLES 12 では入手できますが、SLED 12 では入手できません。ImageMagick は、SLE 12 SDK ISO またはオンラインリポジトリから入手できます。
- SLES 12：問題はありません。すべてのパッケージが利用可能です。ImageMagick は、SLE 12 SDK ISO またはオンラインリポジトリから入手できます。

この問題を解決するには、インストール元となる SLE の代替エディションのメディアから、不足しているパッケージを取得します。すなわち、SLED で不足しているパッケージを SLES メディアからインストールし、SLES で不足しているパッケージを SLED メディアからインストールします。次の方法では、SLED および SLES の ISO メディアファイルを両方ともマウントして、リポジトリを追加します。

- SLED 12 で、次のコマンドを実行します：

```
sudo mkdir -p /mnt/sles

sudo mount -t iso9660 path-to-iso/SLES-12-SP3-DVD-x86_64-GM-DVD1.iso /
    mnt/sles

sudo zypper ar -f /mnt/sles sles
<!--NeedCopy-->
```

- SLED/SLES 12 で、次のコマンドを実行します：

```
sudo mkdir -p /mnt/sdk

sudo mount -t iso9660 path-to-iso/SLE-12-SP3-SDK-DVD-x86_64-GM-DVD1.iso
     /mnt/sdk

sudo zypper ar -f /mnt/sdk sdk
<!--NeedCopy-->
```


Kerberos クライアントのインストール　次のコマンドで、Linux VDA と Delivery Controller 間の相互認証用に Kerberos クライアントをインストールします。

```
sudo zypper install krb5-client
<!--NeedCopy-->
```

Kerberos クライアントの構成は、使用する Active Directory 統合の方法によって異なります。以下の説明を参照してください。

OpenJDK のインストール　OpenJDK 1.8.0 に依存する Linux VDA

> ヒント：
>
> Delivery Controller の登録で失敗しないために、OpenJDK 1.8.0 のみをインストールするようにしてください。その他のバージョンの Java は、システムからすべて削除します。

- **SLED**:

1. SLED では、Java Runtime Environment は通常オペレーティングシステムとともにインストールされています。次のコマンドで、インストールされているか確認してください。

```
sudo zypper info java-1_8_0-openjdk
<!--NeedCopy-->
```

2. ステータスが out-of-date であると報告された場合は、次のようにして最新バージョンに更新します：

```
sudo zypper update java-1_8_0-openjdk
<!--NeedCopy-->
```

3. 次のコマンドで、Java のバージョンを確認します。

```
java -version
<!--NeedCopy-->
```

- **SLES**:

1. SLES では、次のようにして Java Runtime Environment をインストールします。

```
sudo zypper install java-1_8_0-openjdk
<!--NeedCopy-->
```

2. 次のコマンドで、Java のバージョンを確認します。

```
java -version
<!--NeedCopy-->
```

PostgreSQL のインストール　SLED/SLES 12 で、次のようにしてパッケージをインストールします：


```
sudo zypper install postgresql-init

sudo zypper install postgresql-server

sudo zypper install postgresql-jdbc
<!--NeedCopy-->
```

データベースサービスを初期化し、マシンの起動時に PostgreSQL が確実に開始されるようにするには、次に示すインストール後の手順が必要です。

```
sudo systemctl enable postgresql

sudo systemctl restart postgresql
<!--NeedCopy-->
```

データベースファイルは/var/lib/pgsql/data にあります。

リポジトリの削除　依存するパッケージがインストールされると、以前にセットアップした代替エディションのリポジトリを削除し、メディアをマウント解除することができます。

- SLED 12 で、次のコマンドを実行してパッケージを削除します:

```
sudo zypper rr sles

sudo umount /mnt/sles

sudo rmdir /mnt/sles
<!--NeedCopy-->
```

- SLED/SLES 12 で、次のコマンドを実行してパッケージを削除します:

```
sudo zypper rr sdk

sudo umount /mnt/sdk

sudo rmdir /mnt/sdk
<!--NeedCopy-->
```

手順 **2**: ハイパーバイザー用 **Linux** 仮想マシンの準備

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

Citrix Hypervisor での時刻同期の修正

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

一部の Linux ディストリビューションでは、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、Citrix Hypervisor の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 0 または 1 を返します:

- 0 - 時刻同期機能が有効になっているため、無効にする必要があります。
- 1 - 時刻同期機能が無効になっています。これ以上の操作は必要ありません。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は必要ありません。

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

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するため、次のようにしてシステムを再起動します:

```
reboot
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

このコマンドは 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想マシンでは、Hyper-V の時刻同期機能を適用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービス

とともにこの機能を有効にします。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、［統合サービス］を選択します。
3. ［時刻の同期］が選択されていることを確認します。

> 注:
>
> この方法は VMware および Citrix Hypervisor の場合とは異なります。VMware および Citrix Hypervisor では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期を補完することができます。

ESX および **ESXi** での時刻同期の修正

VMware の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーの両方がシステムの時間を同期しようとすることが原因で問題が発生します。システムの時刻と他のサーバーの時刻との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時刻が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

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

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

手順 **3**：**Linux** 仮想マシン（**VM**）を **Windows** ドメインに追加

Linux VDA は、Linux マシンを Active Directory（AD）ドメインに追加するさまざまな方法をサポートします。

- Samba Winbind
- Quest Authentication Services
- Centrify DirectControl

選択した方法の手順に従います。

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

Samba Winbind

Windows ドメインへの参加　ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です。

1. YaST の［Windows ドメインメンバーシップ］を開きます。
2. 以下の変更を行います。
   - ［ドメイン/ワークグループ］に Active Directory ドメインの名前またはドメインコントローラーの IP アドレスを設定します。ドメイン名は必ず大文字にします。
   - ［**Linux** の認証にも **SMB** の情報を使用する］チェックボックスをオンにします。
   - ［**Create Home Directory on Login**］チェックボックスをオンにします。
   - ［**SSH** 向けのシングルサインオン］チェックボックスをオンにします。
   - ［オフライン認証］チェックボックスがオフになっていることを確認します。Linux VDA は、このオプションに対応していません。
3. ［**OK**］をクリックします。いくつかのパッケージのインストールを促すメッセージが表示された場合は、［インストール］をクリックします。
4. ドメインコントローラーが見つかると、ドメインに参加するかどうかを確認するメッセージが表示されます。［**Yes**］をクリックします。
5. メッセージが表示されたら、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の資格情報を入力し、［**OK**］をクリックします。
6. 成功を示すメッセージが表示されます。
7. いくつかの Samba および krb5 パッケージのインストールを促すメッセージが表示されたら、［インストール］をクリックします。

YaST により、これらの変更には一部のサービスまたはマシンの再起動が必要であることが示される場合があります。マシンを再起動することをお勧めします:

```
su -

reboot
<!--NeedCopy-->
```

SLED/SLES 12 のみ:**Kerberos** 資格情報キャッシュ名のパッチ適用　SLED/SLES 12 は、デフォルトの Kerberos 資格情報キャッシュ名指定が通常の **FILE:/tmp/krb5cc_%{uid}** から **DIR:/run/user/%{uid}/krb5cc** に変更されました。Linux VDA はこの新しい DIR によるキャッシュ方法に対応していないため、手動で変更する必要があります。次の設定がない場合は、ルートユーザーとして **/etc/krb5.conf** を編集して、［**libdefaults**］セクションに追加します:

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


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

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

```
sudo net ads testjoin
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

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

```
sudo klist  - ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の`kinit`コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します：

```
sudo kinit -k MACHINE$@REALM
<!--NeedCopy-->
```

マシン名と領域名は大文字で指定する必要があります。ドル記号（$）は、シェルによる置き換えを防ぐためにバックスラッシュ（\）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist
<!--NeedCopy-->
```

次のコマンドを使用して、マシンアカウントの詳細を調査します。

```
sudo net ads status
<!--NeedCopy-->
```

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

```
wbinfo --krb5auth=domain\username%password
<!--NeedCopy-->
```


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

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

```
ssh localhost -l domain\username
id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

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

```
klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6：Linux VDA のインストール」に進みます。

Quest Authentication Services

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

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

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

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

Linux VDA での **Quest** の構成

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

```
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
<!--NeedCopy-->
```

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

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

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

sudo /opt/quest/bin/vastool configure nss
<!--NeedCopy-->
```

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

```
sudo /opt/quest/bin/vastool -u user join domain-name
<!--NeedCopy-->
```

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

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

```
sudo /opt/quest/bin/vastool info domain
<!--NeedCopy-->
```


マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下のエラーが表示されます:

```
ERROR: No domain could be found.
ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm
default_realm not configured in vas.conf. Computer may not be joined
to domain
```

ユーザー認証の確認　PAM を使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことがないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\username
id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

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

```
/opt/quest/bin/vastool klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6: Linux VDA のインストール」に進みます。

Centrify DirectControl

Windows ドメインへの参加　Centrify DirectControl Agent がインストールされている場合、次のように Centrify の **adjoin** コマンドを使用して、Linux マシンを Active Directory ドメインに追加します:

```
su -
adjoin -w -V -u user domain-name
<!--NeedCopy-->
```

user は、コンピューターを Active Directory ドメインに追加する権限を持つ任意の Active Directory ドメインユーザーです。**domain-name** は、Linux マシンを追加するドメインの名前です。

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

```
su -

adinfo
<!--NeedCopy-->
```

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

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
adinfo --sysinfo all

adinfo - diag
<!--NeedCopy-->
```

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

```
adinfo --test
<!--NeedCopy-->
```

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**：前提条件として**.NET Core** ランタイムをインストール

Linux VDA のインストール前に、https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managersの手順に従って.NET Core ランタイムをインストールします。

- 1912 LTSR の初期リリースである CU1 および CU2 の場合は、.NET Core ランタイム 2.1 をインストールします。
- CU3 以降のリリースの場合は、.NET Core ランタイム 3.1 をインストールします。

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

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

手順 **5**：**Linux VDA** パッケージのダウンロード

Citrix Virtual Apps and Desktops サービスのダウンロードページに移動します。適切なバージョンの Citrix Virtual Apps and Desktops を展開し、**Components** をクリックして、使用中の Linux ディストリビューションに対応する Linux VDA パッケージをダウンロードします。

手順 **6**：**Linux VDA** のインストール

手順 **6a**：古いバージョンのアンインストール

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

1. 次のコマンドで、Linux VDA サービスを停止します：

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx stop
<!--NeedCopy-->
```

> 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

2. 次のコマンドで、パッケージをアンインストールします：

```
sudo rpm -e XenDesktopVDA
<!--NeedCopy-->
```

> 重要：
>
> 最新の 2 バージョンからのアップグレードがサポートされます。
>
> 注：
>
> インストールコンポーネントは**/opt/Citrix/VDA/**にあります。
>
> コマンドを実行するには、フルパスが必要です。代わりに、システムパスに**/opt/Citrix/VDA/sbin** および**/opt/Citrix/VDA/bin** を追加することもできます。

手順 **6b**：**Linux VDA** のインストール

Zypper を使用して Linux VDA ソフトウェアをインストールします：

SUSE 12 の場合：

```
sudo zypper install XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
<!--NeedCopy-->
```

RPM Package Manager を使用して、Linux VDA ソフトウェアをインストールします。その前に、次の依存関係を解決します。

SUSE 12 の場合：


```
sudo rpm -i XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
<!--NeedCopy-->
```

手順 **6c**：**Linux VDA** のアップグレード（オプション）

最新の 2 バージョンと 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。

SUSE 12 の場合：

```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
<!--NeedCopy-->
```

RPM 依存関係一覧（**SUSE 12** の場合）：

```
postgresql-server >= 9.3

postgresql-jdbc >= 9.2

java-1.8.0-openjdk >= 1.8.0

ImageMagick >= 6.8

dbus-1 >= 1.8.8

dbus-1-x11 >= 1.8.8

libXpm4 >= 3.5.11

libXrandr2 >= 1.4.2

libXtst6 >= 1.2.2

motif >= 2.3

pam >= 1.1.8

bash >= 4.2

findutils >= 4.5

gawk >= 4.1

sed >= 4.2

cups >= 1.6.0

cups-filters-foomatic-rip >= 1.0.0

openldap2 >= 2.4

cyrus-sasl >= 2.1
```



```
38
39 cyrus-sasl-gssapi >= 2.1
40
41 libxml2 >= 2.9
42
43 python-requests >= 2.8.1
44
45 rpmlib(PayloadFilesHavePrefix) <= 4.0-1
46
47 rpmlib(CompressedFileNames) <= 3.0.4-1
48
49 rpmlib(PayloadIsLzma) <= 4.4.6-1
50
51 libtcmalloc4 >= 2.5
52
53 libcap-progs >= 2.22
54
55 xorg-x11-server >= 7.6_1.18.3-76.15
56
57 ibus >= 1.5
58 <!--NeedCopy-->
```

> 重要:
>
> アップグレードした後、Linux VDA マシンを再起動してください。

手順 7: NVIDIA GRID ドライバーのインストール

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

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

- Citrix Hypervisor
- VMware ESX

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

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

手順 8: Linux VDA の構成

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

ます。必要に応じて、いつでもこのスクリプトを再実行して設定を変更できます。

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

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh - help
<!--NeedCopy-->
```

質問に回答する構成

次のようにして、質問に回答する手動構成を実行します:

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

自動化された構成

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

サポートされる環境変数には次のようなものがあります:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME = Y | N** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定できます。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 には、Delivery Controller の登録に使用する Delivery Controller の完全修飾ドメイン名（FQDN）のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME 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）を使用して、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します。デフォルトでは値は Y に設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES = Y | N** - Linux Virtual Desktop サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。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** - Linux VDA には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。次に示す、サポートされている Active Directory 統合方法のうち、使用するものを指定します:
  - 1 - Samba Winbind
  - 2 - Quest Authentication Service

  - 3 - Centrify DirectControl
  - 4 - SSSD

- **CTX_XDL_HDX_3D_PRO=Y | N** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ（シングルセッション）モード用に構成されます（つまり、CTX_XDL_VDI_MODE=Y となります）。
- **CTX_XDL_VDI_MODE = Y | N** - 専用デスクトップ配信モデル（VDI）またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境では、この変数を Y に設定します。デフォルトでは N に設定されています。
- **CTX_XDL_SITE_NAME = dns-name** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数は、デフォルトでは **<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 の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。たとえば、ad1.mycompany.com:389 となります。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_SEARCH_BASE = search-base-set** - Linux VDA は、Active Directory ドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを使用して LDAP を照会します。検索のパフォーマンスを改善するために、検索ベースを指定できます（例：OU=VDI,DC=mycompany,DC=com）。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_FAS_LIST= ‘list-fas-servers’** - フェデレーション認証サービス（FAS）サーバーは、AD グループポリシーにより構成されます。Linux VDA は AD グループポリシーをサポートしていないため、代わりにセミコロンで区切られた FAS サーバーの一覧を使用できます。シーケンスは、AD グループポリシーで設定したものと同じである必要があります。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスのシーケンスを変更せずに維持します。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** - 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Core ランタイム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_START_SERVICE = Y | N** - Linux VDA 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。デフォルトでは Y に設定されています。

次のようにして、環境変数を設定し、構成スクリプトを実行します：

```
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_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

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

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

```
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_START_SERVICE=Y|N \
```



```

/opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

構成変更の削除

シナリオによっては、Linux VDA パッケージをアンインストールしないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します:

```
sudo /usr/local/sbin/ctxcleanup.sh --help
<!--NeedCopy-->
```

構成変更を削除するには:

```
sudo /usr/local/sbin/ctxcleanup.sh
<!--NeedCopy-->
```

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

構成ログ

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

`/tmp/xdl.configure.log`

Linux VDA サービスを再起動し、変更を反映させます。

手順 **9**: **XDPing** の実行

Linux VDA 環境での一般的な構成の問題を確認するために、コマンドラインユーティリティである Linux `XDPing` ツールが提供されています。`XDPing`パッケージは、サポートされている Linux ディストリビューションを実行している任意のマシンにインストールできます。`XDPing`は、Linux VDA パッケージをマシンにインストールする必要はありません。このツールについて詳しくは、Knowledge Center の記事CTX202015を参照してください。

手順 **10**: **Linux VDA** の実行

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

Linux VDA の起動:

Linux VDA サービスを起動するには、次のコマンドを実行します:

```
sudo /sbin/service ctxhdx start

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の停止:

Linux VDA サービスを停止するには、次のコマンドを実行します:

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx stop
<!--NeedCopy-->
```

> 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

Linux VDA の再起動:

Linux VDA サービスを再起動するには、次のコマンドを実行します:

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx restart

sudo /sbin/service ctxvda start
<!--NeedCopy-->
```

Linux VDA の状態の確認:

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

```
sudo /sbin/service ctxvda status

sudo /sbin/service ctxhdx status
<!--NeedCopy-->
```

手順 **11**: **Citrix Virtual Apps** または **Citrix Virtual Desktops** でのマシンカタログの作成

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

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

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

> 注:
>
> Citrix Studio の以前のバージョンは、「Linux OS」という概念をサポートしていませんでした。ただし、[**Windows** サーバー **OS**] オプションまたは [サーバー **OS**] オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。[**Windows** デスクトップ **OS**] オプションまたは [デスクトップ **OS**] オプションを選択すると、マシンごとに単一ユーザーの配信モデルが暗黙的に選択されます。
>
> ヒント:
>
> マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

手順 **12**: **Citrix Virtual Apps** または **Citrix Virtual Desktops** でのデリバリーグループの作成

デリバリーグループを作成し、Linux VDA マシンを含むマシンカタログを追加する手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「デリバリーグループの作成」を参照してください。

Linux VDA マシンカタログを含むデリバリーグループを作成する場合は、次の制約があります:

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

> 重要:
>
> アプリケーションの公開は、Linux VDA バージョン 1.4 以降でサポートされています。ただし、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

Linux Virtual Delivery Agent for Ubuntu の手動インストール

December 13, 2022

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

手順 **1**: **Ubuntu for VDA** をインストールする準備

手順 **1a**: ネットワーク構成の確認

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

Ubuntu 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** 構成ファイルに次の変更を加えます:

```
preserve_hostname: true
```

手順 **1b**: ホスト名の設定

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

```
hostname
```

手順 **1c**: ホスト名へのループバックアドレスの割り当て

マシンの DNS ドメイン名と完全修飾ドメイン名（FQDN）が正しく報告されることを確認します。このためには、**/etc/hosts** ファイルの次の行の最初の 2 つのエントリに FQDN とホスト名が含まれるように編集します:

```
127.0.0.1 hostname-fqdn hostname localhost
```

例:

```
127.0.0.1 vda01.example.com vda01 localhost
```

ファイル内の他のエントリから、**hostname-fqdn** または **hostname** に対するその他の参照を削除します。

> 注:
>
> Linux VDA は現在、NetBIOS 名の切り捨てをサポートしていません。したがって、ホスト名は 15 文字以内である必要があります。
>
> ヒント:
>
> a～z、A～Z、0～9、およびハイフン（-）の文字のみ使用してください。アンダースコア（_）、スペース、およびその他の記号は使用しないでください。ホスト名を数字で開始したり、ハイフンで終了したりしないでください。このルールは、Delivery Controller のホスト名にも適用されます。

手順 **1d**: ホスト名の確認

次のコマンドで、ホスト名が正しく設定されていることを確認します:

```
hostname
<!--NeedCopy-->
```

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

次のコマンドで、完全修飾ドメイン名が正しく設定されていることを確認します:

```
hostname -f
<!--NeedCopy-->
```

このコマンドにより、そのマシンの完全修飾ドメイン名が返されます。

手順 **1e**: マルチキャスト **DNS** の無効化

デフォルトの設定でマルチキャスト DNS（**mDNS**）が有効であるため、名前解決の結果に不整合が発生する場合があります。

mDNS を無効にするには、**/etc/nsswitch.conf** を編集して、以下を含む行を変更します:

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

変更後:

```
hosts: files dns
```

手順 **1f**: 名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn

```



```
ping domain-controller-fqdn

nslookup delivery-controller-fqdn

ping delivery-controller-fqdn
<!--NeedCopy-->
```

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

手順 **1g**：時刻同期の構成（**chrony**）

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

chrony のインストール：

```
apt-get install chrony
<!--NeedCopy-->
```

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

```
server peer1-fqdn-or-ip-address iburst
server peer2-fqdn-or-ip-address iburst
```

一般的な環境では、時間はローカルドメインコントローラーから同期します。公開 NTP プールサーバーから直接は同期しません。ドメインの各 Active Directory ドメインコントローラーに対応するサーバーエントリを追加します。

ループバック IP アドレス、localhost、パブリックサーバーの ***.pool.ntp.org** エントリなど、一覧にあるその他のサーバーまたはプールエントリを削除します。

変更を保存してから、次のコマンドで Chrony デーモンを再起動します：

```
sudo systemctl restart chrony
<!--NeedCopy-->
```

手順 **1h**：**OpenJDK** のインストール

Linux VDA は OpenJDK に依存しています。通常、Runtime Environment は、オペレーティングシステムの一部としてインストールされています。

Ubuntu 16.04 では、以下を実行して OpenJDK をインストールします：

```
sudo apt-get install -y default-jdk
<!--NeedCopy-->
```


Ubuntu 18.04 では、以下を実行して OpenJDK をインストールします:

```
sudo apt-get install -y openjdk-8-jdk
<!--NeedCopy-->
```

手順 **1i**: **PostgreSQL** のインストール

Linux VDA を使用するには、Ubuntu 上に PostgreSQL バージョン 9.x が必要です:

```
sudo apt-get install -y postgresql

sudo apt-get install -y libpostgresql-jdbc-java
<!--NeedCopy-->
```

手順 **1j**: **Motif** のインストール

```
sudo apt-get install -y libxm4
<!--NeedCopy-->
```

手順 **1k**: 他のパッケージのインストール

```
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 cups
<!--NeedCopy-->
```

手順 **1l**: 次のパッケージのインストール（**Ubuntu 18.04** のみ）

```
sudo apt-get install -y libgtk2.0-0
<!--NeedCopy-->
```

手順 **2**: ハイパーバイザーの準備

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

Citrix Hypervisor での時刻同期の修正

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

一部の Linux ディストリビューションでは、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、Citrix Hypervisor の時刻同期機能が存在するかどうかと、Linux 仮想マシン内で有効になっているかどうかを確認できます:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 0 または 1 を返します:

- 0 - 時刻同期機能が有効になっているため、無効にする必要があります。
- 1 - 時刻同期機能が無効になっています。これ以上の操作は必要ありません。

/proc/sys/xen/independent_wallclock ファイルが存在しない場合、以下の手順は必要ありません。

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

```
sudo echo 1 > /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するため、次のようにしてシステムを再起動します:

```
su -

cat /proc/sys/xen/independent_wallclock
<!--NeedCopy-->
```

このコマンドは 1 を返します。

Microsoft Hyper-V での時刻同期の修正

Hyper-V Linux 統合サービスがインストールされた Linux 仮想マシンでは、Hyper-V の時刻同期機能を使用してホストオペレーティングシステムの時間を利用できます。システムの時間を正確な状態で維持するには、NTP サービスとともにこの機能を有効にする必要があります。

管理オペレーティングシステムで、次の操作を行います。

1. Hyper-V マネージャーを開きます。
2. Linux 仮想マシンの設定で、［統合サービス］を選択します。
3. ［時刻の同期］が選択されていることを確認します。

> 注:
>
> この方法は VMware および Citrix Hypervisor の場合とは異なります。VMware および Citrix Hypervisor では、NTP との競合を避けるためにホストの時刻同期を無効にします。Hyper-V の時刻同期は、NTP と共存し、NTP の時刻同期を補完することができます。

ESX および **ESXi** での時刻同期の修正

VMware の時刻同期機能が有効な場合、それぞれの準仮想化 Linux 仮想マシンで、NTP とハイパーバイザーの両方がシステムの時間を同期しようとすることが原因となり問題が発生します。システムの時間と他のサーバーの時間との同期が失われるのを防ぐには、各 Linux ゲストのシステムの時間が NTP と同期する必要があります。この場合、ホストの時刻同期を無効にする必要があります。

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

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

手順 **3**：**Linux** 仮想マシン（**VM**）を **Windows** ドメインに追加

Linux VDA は、Linux マシンを Active Directory（AD）ドメインに追加するさまざまな方法をサポートします。

- Samba Winbind
- Quest Authentication Services
- Centrify DirectControl
- SSSD
- PBIS

選択した方法の手順に従います。

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

Samba Winbind

必要なパッケージのインストールまたは更新

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-
    config krb5-locales krb5-user
<!--NeedCopy-->
```

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

```
sudo systemctl enable winbind
<!--NeedCopy-->
```

Kerberos の構成　ルートユーザーとして**/etc/krb5.conf** を開き、以下を設定します。

```
[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
<!--NeedCopy-->
```

ここで **domain-dns-name** プロパティは、DNS ドメイン名（**example.com** など）です。**REALM** は、大文字の Kerberos 領域名（**EXAMPLE.COM** など）です。

Winbind 認証の構成　RHEL の`authconfig`や、SUSE の yast2 のようなツールが Ubuntu にないため、手動で Winbind を構成します。

/etc/samba/smb.conf を開き、次を設定します。

```
[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
<!--NeedCopy-->
```

WORKGROUP は、**REALM** の最初のフィールドです。**REALM** は大文字の Kerberos 領域名です。

nsswitch の構成　**/etc/nsswitch.conf** を開き、winbindを次の行に追加します:

```
passwd: compat winbind
group: compat winbind
```

Windows ドメインへの参加　ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です:

```
sudo net ads join REALM -U user
<!--NeedCopy-->
```

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

winbind の再起動

```
sudo systemctl restart winbind
<!--NeedCopy-->
```

Winbind 用の **PAM** の構成　次のコマンドを実行して、[**Winbind NT/Active Directory authentication**] オプションと [**Create home directory on login**] オプションが選択されているようにします:

```
sudo pam-auth-update
<!--NeedCopy-->
```


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

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

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

```
sudo net ads testjoin
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

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

```
sudo klist -ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の`kinit`コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します：

```
sudo kinit -k MACHINE$@REALM
<!--NeedCopy-->
```

マシン名と領域名は大文字で指定する必要があります。ドル記号（$）は、シェルによる置き換えを防ぐためにバックスラッシュ（\）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist
<!--NeedCopy-->
```

次のコマンドを使用して、マシンアカウントの詳細を調査します：

```
sudo net ads status
<!--NeedCopy-->
```

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


```
wbinfo --krb5auth=domain\username%password
<!--NeedCopy-->
```

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

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

```
ssh localhost -l domain\username

id -u
<!--NeedCopy-->
```

> 注:
>
> SSH コマンドを正しく実行するには、SSH が有効で適切に機能していることを確認します。

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

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

```
klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

Gnome コンソールまたは KDE コンソールに直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6: Linux VDA のインストール」に進みます。

> ヒント:
>
> ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップを表示できない場合、マシンを再起動して再試行します。

Quest Authentication Services

ドメインコントローラーでの **Quest** の構成　次の操作は、Quest ソフトウェアを Active Directory ドメインコントローラーにインストールし、構成していることと、管理者特権が付与され、Active Directory にコンピューターオ

ブジェクトを作成できることを前提としています。

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

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

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

Linux VDA での **Quest** の構成

SELinux ポリシー適用の回避策　デフォルトの RHEL 環境では、SELinux が完全に適用されています。この適用により、Quest が使用する Unix ドメインソケットの IPC のメカニズムに干渉し、ドメインユーザーのログオンを妨げます。

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

```
SELINUX=disabled
```

この変更にはマシンの再起動が必要です:

```
reboot
<!--NeedCopy-->
```

> 重要:
>
> この設定は注意して使用してください。SELinux ポリシーの適用を無効にした後に再度有効にすると、ルートユーザーやその他のローカルユーザーであっても、完全にロックアウトされてしまう可能性があります。

VAS デーモンの構成　次のように Kerberos チケットの自動更新を有効にして、切断する必要があります。認証（オフラインログオン）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
<!--NeedCopy-->
```


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

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

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

sudo /opt/quest/bin/vastool configure nss
<!--NeedCopy-->
```

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

```
sudo /opt/quest/bin/vastool -u user join domain-name
<!--NeedCopy-->
```

user は、コンピューターを Active Directory ドメインに追加する権限を持つ任意のドメインユーザーです。domain-name は、ドメインの DNS 名（example.com など）です。

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

```
sudo /opt/quest/bin/vastool info domain
<!--NeedCopy-->
```

マシンがドメインに参加している場合は、ドメイン名が返されます。マシンがドメインに追加していない場合、以下のエラーが表示されます:

```
ERROR: No domain could be found.
ERROR: VAS_ERR_CONFIG: at ctx.c:414 in _ctx_init_default_realm
default_realm not configured in vas.conf. Computer may not be joined
to domain
```

ユーザー認証の確認　PAM を使用した Quest のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことがないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\username

id -u
<!--NeedCopy-->
```


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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

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

```
/opt/quest/bin/vastool klist
<!--NeedCopy-->
```

次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

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

Centrify DirectControl

Windows ドメインへの参加　Centrify DirectControl Agent がインストールされている場合、次のように Centrify の`adjoin`コマンドを使用して、Linux マシンを Active Directory ドメインに追加します:

```
su -
adjoin -w -V -u user domain-name
<!--NeedCopy-->
```

user パラメーターは、コンピューターを Active Directory ドメインに追加する権限を持つ任意の Active Directory ドメインユーザーです。**domain-name** パラメーターは、Linux マシンを追加するドメインの名前です。

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

```
su -

adinfo
<!--NeedCopy-->
```

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

次を使用すると、より包括的なシステム情報と診断情報を取得できます。

```
adinfo --sysinfo all

adinfo --diag
```



```
<!--NeedCopy-->
```

さまざまな Active Directory および Kerberos サービスとの接続をテストするには、次のコマンドを実行します:

```
adinfo --test
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

Kerberos を構成するには、**/etc/krb5.conf** をルートとして開き、次を設定します:

```
[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
<!--NeedCopy-->
```

ここで`domain-dns-name`プロパティは、DNS ドメイン名（`example.com`など）です。`REALM`は大文字の Kerberos 領域名で、`EXAMPLE.COM`などです。

ドメインに参加する　SSSD を構成して、Active Directory を ID プロバイダーおよび認証の Kerberos として使用します。ただし、SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する AD のクライアント機能が提供されていません。代わりに`adcli`、`realmd`または`Samba`を使用できます。

> 注:
>
> このセクションでは、adcli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
<!--NeedCopy-->
```

2. adcliでドメインに参加します。

   次を使用して古いシステム keytab ファイルを削除し、ドメインに参加させます。

```
su -

rm -rf /etc/krb5.keytab

adcli join domain-dns-name -U user -H hostname-fqdn
<!--NeedCopy-->
```

   user は、ドメインにマシンを追加する権限があるドメインユーザーです。**hostname-fqdn** は、完全修飾ドメイン名形式のマシンのホスト名です。

   -H オプションは、adcliが、Linux VDA で必要な host/hostname-fqdn@REALM という形式で SPN を生成するのに必要です。

3. システムの Keytab を確認します。

   adcliツールの機能は限られており、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は提供されていません。システムの keytab ファイルが作成されていることを確認するための最良の代替方法:

```
sudo klist -ket
<!--NeedCopy-->
```

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致するかを検証します。

- **Samba**を使用してドメインに参加する場合は、次の手順を実行します:

1. パッケージをインストールします。

```
sudo apt-get install samba
<!--NeedCopy-->
```

2. Sambaを構成します。

   /etc/samba/smb.conf を開き、次を設定します。

```
[global]

workgroup = WORKGROUP
```



```
security = ADS

realm = REALM

client signing = yes

client use spnego = yes

kerberos method = secrets and keytab
<!--NeedCopy-->
```

WORKGROUP は、**REALM** の最初のフィールドです。REALM は大文字の Kerberos 領域名です。

3. Sambaでドメインに参加します。

   ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Windows アカウントが必要です。

```
sudo net ads join REALM -U user
<!--NeedCopy-->
```

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

SSSD のセットアップ　必要なパッケージのインストールまたは更新:

必要な SSSD および構成パッケージがインストールされていない場合、インストールします。

```
sudo apt-get install sssd
<!--NeedCopy-->
```

パッケージが既にインストールされている場合、更新することをお勧めします。

```
sudo apt-get install --only-upgrade sssd
<!--NeedCopy-->
```

> 注:
>
> Ubuntu のインストールプロセスは、デフォルトで自動的に **nsswitch.conf** および PAM ログインモジュールを構成します。

SSSD の構成　SSSD デーモンを起動する前に、SSSD 構成の変更が必要です。SSSD の一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルはデフォルトではインストールされないため、手動で作成する必要があります。root として **/etc/sssd/sssd.conf** を作成するか開いて、次を設定します:

```
[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
<!--NeedCopy-->
```

> 注:
>
> ldap_id_mapping は **true** に設定されるため、SSSD 自体が Windows SID を Unix UID にマッピングします。設定しない場合、Active Directory が POSIX 拡張を提供できるようにする必要があります。PAM サービス（`ctxhdx`）は、ad_gpo_map_remote_interactive に追加されます。

ここで`domain-dns-name`プロパティは、DNS ドメイン名（`example.com`など）です。`REALM`は大文字の Kerberos 領域名で、`EXAMPLE.COM`などです。NetBIOS ドメイン名を構成するための要件はありません。

> ヒント：
>
> この構成設定について詳しくは、`sssd.conf`および`sssd-ad`に関する man ページを参照してください。

SSSD デーモンでは、構成ファイルに所有者読み取り権限のみが設定されている必要があります。

```
sudo chmod 0600 /etc/sssd/sssd.conf
<!--NeedCopy-->
```

SSSD デーモンの起動　次のコマンドを実行して、SSSD デーモンを起動し、マシンの起動時にもデーモンを起動できるようにします。

```
sudo systemctl start sssd

sudo systemctl enable sssd
<!--NeedCopy-->
```

PAM 構成　次のコマンドを実行して、［**SSS authentication**］オプションと ［**Create home directory on login**］オプションが選択されているようにします：

```
sudo pam-auth-update
<!--NeedCopy-->
```

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

- `adcli`を使用してドメインメンバーシップを確認する場合は、`sudo adcli info domain-dns-name`コマンドを実行してドメイン情報を表示します。
- `Samba`を使用してドメインメンバーシップを確認する場合は、`sudo net ads testjoin`コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info`コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

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

```
sudo klist -ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の`kinit`コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します：


```
sudo kinit -k MACHINE$@REALM
<!--NeedCopy-->
```

マシン名と領域名は大文字で指定する必要があります。ドル記号（$）は、シェルによる置き換えを防ぐためにバックスラッシュ（\）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist
<!--NeedCopy-->
```

ユーザー認証の確認　SSSD は、デーモンで直接認証をテストするコマンドラインツールを提供しません。PAM 経由でのみ完了できます。

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

```
ssh localhost -l domain\username

id -u

klist

exit
<!--NeedCopy-->
```

ユーザーの **klist** コマンドで返される Kerberos チケットが正しく、期限切れではないことを確認します。

ルートユーザーとして、前述の **id -u** コマンドで返された UID に対応するチケットキャッシュファイルが作成されたことを確認します:

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```

KDE または Gnome Display Manager に直接ログオンすると、同様のテストを実行できます。ドメイン参加の確認後、「手順 6: Linux VDA のインストール」に進みます。

PBIS

必要な **PBIS** パッケージをダウンロードする　例:

```
sudo wget https://github.com/BeyondTrust/pbis-open/releases/download
    /8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh
<!--NeedCopy-->
```


PBIS インストールスクリプトを実行可能にする　例:

```
sudo chmod +x pbis-open-8.8.0.506.linux.x86_64.deb.sh
<!--NeedCopy-->
```

PBIS インストールスクリプトを実行する　例:

```
sudo sh pbis-open-8.8.0.506.linux.x86_64.deb.sh
<!--NeedCopy-->
```

Windows ドメインへの参加　ドメインコントローラーがアクセス可能で、コンピューターをドメインに追加する権限を持つ Active Directory ユーザーアカウントが必要です:

```
sudo /opt/pbis/bin/domainjoin-cli join domain-name user
<!--NeedCopy-->
```

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

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

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

```
/opt/pbis/bin/domainjoin-cli query
<!--NeedCopy-->
```

マシンがドメインに参加している場合、このコマンドは現在の AD ドメインと OU の情報を返します。参加していない場合は、ホスト名だけが表示されます。

ユーザー認証の確認　PAM を使用した PBIS のドメインユーザーの認証が可能かどうかを確認するには、以前に使用したことがないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
sudo ssh localhost -l domain\user

id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_uid
<!--NeedCopy-->
```


次のコマンドで、セッションを終了します。

```
exit
<!--NeedCopy-->
```

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: 前提条件として.NET Core ランタイムをインストール

Linux VDA のインストール前に、https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managersの手順に従って.NET Core ランタイムをインストールします。

- 1912 LTSR の初期リリースである CU1 および CU2 の場合は、.NET Core ランタイム 2.1 をインストールします。
- CU3 以降のリリースの場合は、.NET Core ランタイム 3.1 をインストールします。

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

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

手順 5: Linux VDA パッケージのダウンロード

Citrix Virtual Apps and Desktops サービスのダウンロードページに移動します。適切なバージョンの Citrix Virtual Apps and Desktops を展開し、**Components** をクリックして、使用中の Linux ディストリビューションに対応する Linux VDA パッケージをダウンロードします。

手順 6: Linux VDA のインストール

手順 6a: Linux VDA のインストール

次のように、Debian Package Manager を使用して Linux VDA ソフトウェアをインストールします。

Ubuntu 18.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu18.04_amd64.deb
<!--NeedCopy-->
```

Ubuntu 16.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu16.04_amd64.deb
<!--NeedCopy-->
```

Ubuntu 18.04 の Debian 依存関係一覧:


```
postgresql >= 9.5

libpostgresql-jdbc-java >= 9.2

openjdk-8-jdk >= 1.8.0

gtk3-nocsd >=3

imagemagick >= 8:6.8.9.9

ufw >= 0.35

ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1

libldap-2.4-2 >= 2.4.42

libsasl2-modules-gssapi-mit >= 2.1.~

python-requests >= 2.9.1

libgoogle-perftools4 >= 2.4~

xserver-xorg-core >= 2:1.18

xserver-xorg-core << 2:1.19

x11vnc>=0.9.13

python-websockify >= 0.6.1
<!--NeedCopy-->
```

Ubuntu 16.04 の Debian 依存関係一覧:

```
postgresql >= 9.5

libpostgresql-jdbc-java >= 9.2

```



```
default-jdk >= 2:1.8

imagemagick >= 8:6.8.9.9

ufw >= 0.35

ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1

libldap-2.4-2 >= 2.4.42

libsasl2-modules-gssapi-mit >= 2.1.~

python-requests >= 2.9.1

libgoogle-perftools4 >= 2.4~

xserver-xorg-core >= 2:1.18

xserver-xorg-core << 2:1.19

x11vnc>=0.9.13

python-websockify >= 0.6.1
<!--NeedCopy-->
```

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

手順 **6b**: **Linux VDA** のアップグレード（オプション）

最新の 2 バージョンと 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。

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



```
<!--NeedCopy-->
```

手順 7：NVIDIA GRID ドライバーのインストール

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

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

- Citrix Hypervisor
- VMware ESX

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

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

手順 8：Linux VDA の構成

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

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

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh --help
<!--NeedCopy-->
```

質問に回答する構成

次のようにして、質問に回答する手動構成を実行します：

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```


自動化された構成

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

サポートされる環境変数には次のようなものがあります:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME = Y | N** - Linux VDA では、DNS CNAME レコードを使用して、Delivery Controller 名を指定できます。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 には、Delivery Controller の登録に使用する Delivery Controller の完全修飾ドメイン名（FQDN）のスペース区切りの一覧が必要です。1 つまたは複数の完全修飾ドメイン名または CNAME 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）を使用して、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します。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 Virtual Desktop サービスでは、ネットワーク受信接続がシステムのファイアウォールの通過を許可されている必要があります。Linux Virtual Desktop 用に、システムのファイアウォールの必要なポート（デフォルトではポート 80 およびポート 1494）を自動で開放できます。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 には、Delivery Controller に対して認証するために Kerberos 構成設定が必要です。Kerberos 構成は、システムにインストールおよび構成済みの Active Directory 統合ツールから指定します。次に示す、サポートされている Active Directory 統合方法のうち、使用するものを指定します:
  - 1 - Samba Winbind
  - 2 - Quest Authentication Service
  - 3 - Centrify DirectControl
  - 4 - SSSD
  - 5 - PBIS
- **CTX_XDL_HDX_3D_PRO=Y | N** - Linux VDA では、HDX 3D Pro がサポートされます。これは、強力なグラフィックアプリケーションの仮想化を最適にするための一連の GPU アクセラレーションテクノロジです。HDX 3D Pro を選択した場合、VDA は VDI デスクトップ（シングルセッション）モード用に構成されます（つまり、CTX_XDL_VDI_MODE=Y となります）。
- **CTX_XDL_VDI_MODE = Y | N** - 専用デスクトップ配信モデル（VDI）またはホストされる共有デスクトップ配信モデルのどちらとしてマシンを構成するかを決定します。HDX 3D Pro 環境では、この変数を Y に設定します。デフォルトでは N に設定されています。
- **CTX_XDL_SITE_NAME = dns-name** - Linux VDA は、DNS を使用して LDAP サーバーを検出します。DNS の検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数は、デフォルトでは **<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 の FQDN および LDAP ポートのスペース区切りの一覧を指定できます。たとえば、ad1.mycompany.com:389 となります。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_SEARCH_BASE = search-base-set** - Linux VDA は、Active Directory ドメインのルート（例：DC=mycompany,DC=com）に設定された検索ベースを使用して LDAP を照会します。ただし、検索のパフォーマンスを改善するために、検索ベースを指定できます（例：OU=VDI,DC=mycompany,DC=com）。この変数は、デフォルトでは **<none>** に設定されています。
- **CTX_XDL_FAS_LIST=’list-fas-servers’** - フェデレーション認証サービス（FAS）サーバーは、AD グループポリシーにより構成されます。Linux VDA は AD グループポリシーをサポートしていないため、代わりにセミコロンで区切られた FAS サーバーの一覧を使用できます。シーケンスは、AD グループポリシーで設定したものと同じである必要があります。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスのシーケンスを変更せずに維持します。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** - 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Core ランタイム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_START_SERVICE = Y | N** - Linux VDA 構成の完了時に Linux VDA サービスが開始されるようにするかどうかを指定します。デフォルトでは Y に設定されています。

次のようにして、環境変数を設定し、構成スクリプトを実行します：

```
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_START_SERVICE=Y|N

sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

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

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

```
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_START_SERVICE=Y|N \

/opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

構成変更の削除

シナリオによっては、Linux VDA パッケージをアンインストールしないで、**ctxsetup.sh** スクリプトによって行われた構成変更を削除することが必要となる場合があります。

続行する前に、次のコマンドを使用してこのスクリプトのヘルプを確認します:

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh --help
<!--NeedCopy-->
```


構成変更を削除するには:

```
sudo /opt/Citrix/VDA/sbin/ctxcleanup.sh
<!--NeedCopy-->
```

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

構成ログ

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

Linux VDA サービスを再起動し、変更を反映させます。

Linux VDA ソフトウェアのアンインストール

Linux VDA がインストールされているかどうかを確認したり、インストールされているパッケージのバージョンを表示するには、次のコマンドを実行します。

```
dpkg -l xendesktopvda
<!--NeedCopy-->
```

詳細を表示するには、次のコマンドを実行します。

```
apt-cache show xendesktopvda
<!--NeedCopy-->
```

Linux VDA ソフトウェアをアンインストールには、次のコマンドを実行します:

```
dpkg -r xendesktopvda
<!--NeedCopy-->
```

> 注:
>
> Linux VDA ソフトウェアをアンインストールすると、関連付けられた PostgreSQL およびその他の構成データが削除されます。ただし、Linux VDA のインストールより前にセットアップされた、PostgreSQL パッケージおよびその他の依存するパッケージは削除されません。
>
> ヒント:
>
> このセクションでは、PostgreSQL など、依存するパッケージの削除方法については説明していません。

手順 9：XDPing の実行

Linux VDA 環境での一般的な構成の問題を確認するために、コマンドラインユーティリティである Linux `XDPing` ツールが提供されています。`XDPing`パッケージは、サポートされている Linux ディストリビューションを実行している任意のマシンにインストールできます。`XDPing`は、Linux VDA パッケージをマシンにインストールする必要はありません。このツールについて詳しくは、Knowledge Center の記事CTX202015を参照してください。

手順 10：Linux VDA の実行

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

Linux VDA の起動：

Linux VDA サービスを起動するには、次のコマンドを実行します：

```
sudo systemctl start ctxhdx

sudo systemctl start ctxvda
<!--NeedCopy-->
```

Linux VDA の停止：

Linux VDA サービスを停止するには、次のコマンドを実行します：

```
sudo systemctl stop ctxvda

sudo systemctl stop ctxhdx
<!--NeedCopy-->
```

> 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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

Linux VDA の再起動：

Linux VDA サービスを再起動するには、次のコマンドを実行します：

```
sudo systemctl stop ctxvda

sudo systemctl restart ctxhdx

sudo systemctl restart ctxvda
<!--NeedCopy-->
```

Linux VDA の状態の確認：

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


```
sudo systemctl status ctxvda

sudo systemctl status ctxhdx
<!--NeedCopy-->
```

手順 **11**：**Citrix Virtual Apps** または **Citrix Virtual Desktops** でのマシンカタログの作成

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

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

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

> 注：
>
> Citrix Studio の以前のバージョンは、「Linux OS」という概念をサポートしていませんでした。ただし、［**Windows** サーバー **OS**］オプションまたは ［サーバー **OS**］オプションを選択すると、同等のホストされる共有デスクトップ配信モデルが暗黙的に選択されます。［**Windows** デスクトップ **OS**］オプションまたは［デスクトップ **OS**］オプションを選択すると、マシンごとに単一ユーザーの配信モデルが暗黙的に選択されます。
>
> ヒント：
>
> マシンが Active Directory ドメインから削除された後に再度追加された場合は、そのマシンをマシンカタログから削除してから再度追加する必要があります。

手順 **12**：**Citrix Virtual Apps** または **Citrix Virtual Desktops** でのデリバリーグループの作成

デリバリーグループを作成し、Linux VDA マシンを含むマシンカタログを追加する手順は、Windows VDA マシンの場合とほとんど同じです。このタスクを完了する方法の説明について詳しくは、「デリバリーグループの作成」を参照してください。

Linux VDA マシンカタログを含むデリバリーグループを作成する場合は、次の制約があります：

- 選択する AD ユーザーおよびグループを、Linux VDA マシンにログオンするように適切に構成しておきます。
- 認証されていない（匿名）ユーザーのログオンを許可しないでください。
- Windows マシンを含むマシンカタログをデリバリーグループで混在させないでください。

マシンカタログおよびデリバリーグループの作成方法について詳しくは、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

Machine Creation Services（MCS）を使用した Linux 仮想マシンの作成

May 15, 2023

7.18 リリース以降、MCS を使用して Linux 仮想マシンを作成できます。

サポートされるハイパーバイザー

- AWS
- Citrix Hypervisor
- Microsoft Azure
- VMware vSphere

サポート対象ではないハイパーバイザーでマスターイメージを準備しようとすると、予期しない問題が発生することがあります。

Citrix Hypervisor での MCS を使用した Linux 仮想マシンの作成

手順 **1**：マスターイメージの準備

マスターイメージには、オペレーティングシステム、仮想化しないアプリケーション、VDA などのソフトウェアをインストールしておきます。マスターイメージを準備するには、次の手順を実行します。

手順 **1a**：**Citrix VM Tools** のインストール　xe CLI または XenCenter を使用できるようにするには、仮想マシンごとにテンプレート仮想マシンに Citrix VM Tools をインストールする必要があります。このツールがインストールされていないと、仮想マシンのパフォーマンスが低下する可能性があります。ツールがなければ、次のいずれも実行できません。

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

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


```
sudo mount /dev/cdrom /mnt
<!--NeedCopy-->
```

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

   RHEL/CentOS の場合:

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

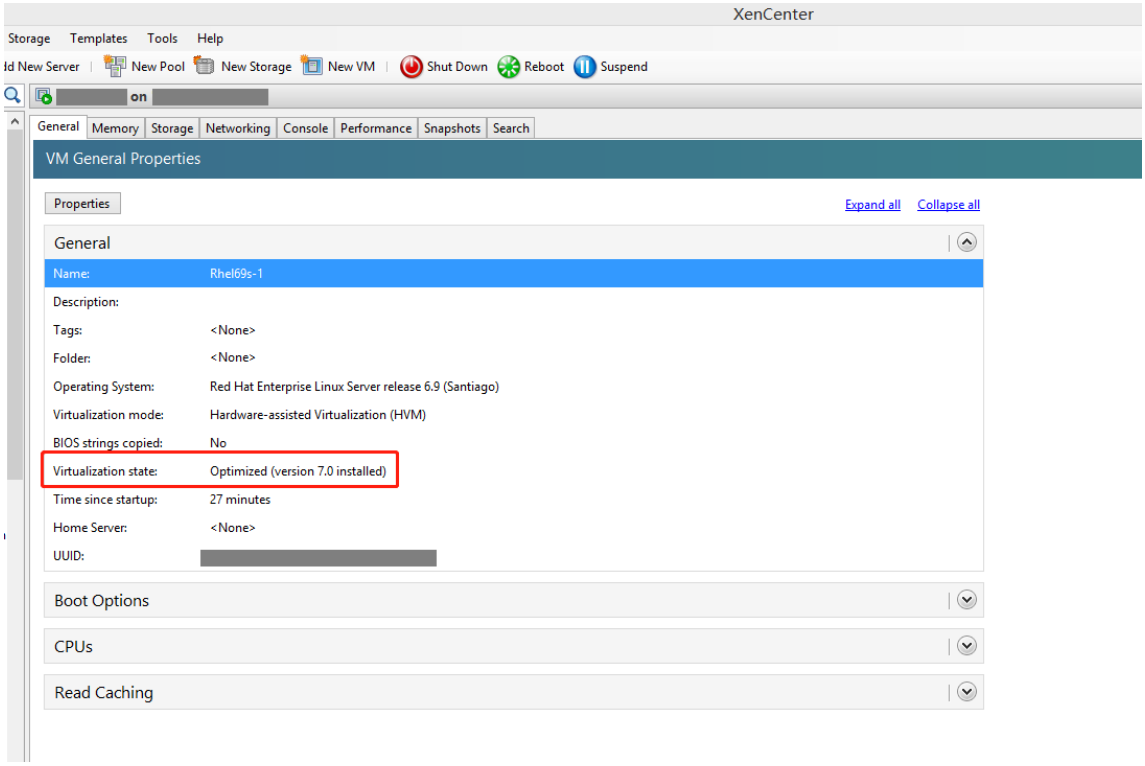
   Ubuntu の場合:

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

   SUSE 12 の場合:

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

3. XenCenter の ［全般］タブで、テンプレート仮想マシンの仮想化状態を確認します。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 ［最適化済み］となります:



手順 **1b**：テンプレート仮想マシンに **Linux VDA** パッケージをインストールする

> 注：
>
> 現在実行中の VDA をテンプレート仮想マシンとして使用するには、この手順を省略します。
>
> テンプレート仮想マシンに Linux VDA パッケージをインストールする前に、.NET Core ランタイム 3.1 をインストールします。詳しくは、「インストールの概要」を参照してください。

使用している Linux ディストリビューションごとに、次のコマンドを実行して、Linux VDA の環境をセットアップします。

RHEL/CentOS の場合：

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

Ubuntu の場合：

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

apt-get install -f
<!--NeedCopy-->
```

SUSE 12 の場合：

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


手順 **1c**：リポジトリを有効にして **tdb-tools** パッケージをインストールする　**RHEL 7** サーバーの場合：

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
<!--NeedCopy-->
```

RHEL 7 ワークステーションの場合：

```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
<!--NeedCopy-->
```

手順 **1d**：**ntfs-3g** が含まれる **EPEL** リポジトリをインストールする　EPEL リポジトリを RHEL 6/CentOS 6、RHEL 7/CentOS 7 にインストールし、後から deploymcs.sh を実行すると ntfs-3g パッケージがインストールされるようにします。

手順 **1e**：**SUSE 12** に **ntfs-3g** を手動でインストールする　SUSE 12 プラットフォームには、ntfs-3g を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、ntfs-3g を手動でインストールします：

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

   ```
   sudo zypper install gcc
   sudo zypper install make
   <!--NeedCopy-->
   ```

2. ntfs-3g パッケージをダウンロードします。

3. ntfs-3g パッケージを展開します。

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

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>
   <!--NeedCopy-->
   ```

5. ntfs-3g をインストールします：

   ```
   ./configure
   make
   make install
   <!--NeedCopy-->
   ```

手順 **1f**：ランタイム環境のセットアップ　**deploymcs.sh** の実行前に、次の操作を行います：

- **/etc/xdl/mcs/mcs.conf** の変数を変更します。**mcs.conf** 構成ファイルには、MCS と Linux VDA を設定するための変数が含まれています。必要に応じて設定できる変数は次のとおりです：

  - `Use_Existing_Configurations_Of_Current_VDA`：現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在実行中の VDA の構成ファイルと同じファイルになります。ただし、`dns`変数と`AD_INTEGRATION`変数を構成する必要があります。デフォルト値は N です。これは、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。
  - `dns`：DNS の IP アドレスを設定します。
  - `AD_INTEGRATION`：Winbind または SSSD を設定します。
  - `WORKGROUP`：AD で構成されている場合、そのワークグループ名を設定します。これは NetBIOS 名です（大文字と小文字を区別）。それ以外の場合は、デフォルトではドメイン名です。

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

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

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

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

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

手順 **1g**：マスターイメージを作成する

1. **/opt/Citrix/VDA/sbin/deploymcs.sh** を実行します。
2. （オプション）テンプレート仮想マシン上で構成テンプレートを更新して、作成されたすべての仮想マシン上の/etc/krb5.conf、/etc/samba/smb.conf、および/etc/sssd/sssd.conf ファイルを適宜カスタマイズします。

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

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

   注：テンプレートファイルで使用されている既存の形式を保持し、$WORKGROUP、$REALM、$realm、および $AD_FQDN などの変数を使用してください。

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

手順 **2**：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仮想マシンの数を指定します。必要に応じて他の構成タスクを実行します。詳しくは、「Studio でのマシンカタログの作成」を参照してください。

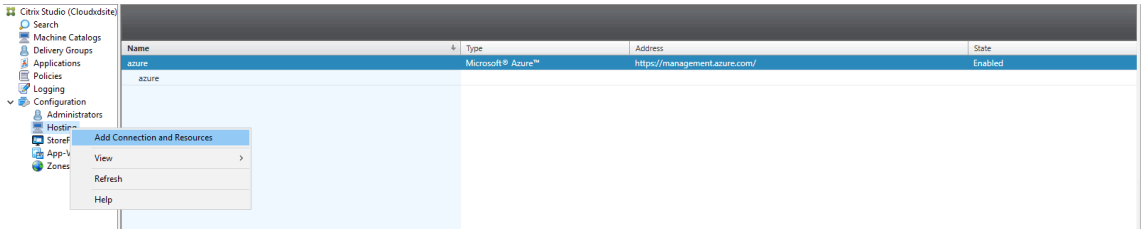
手順 **3**：デリバリーグループの作成

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

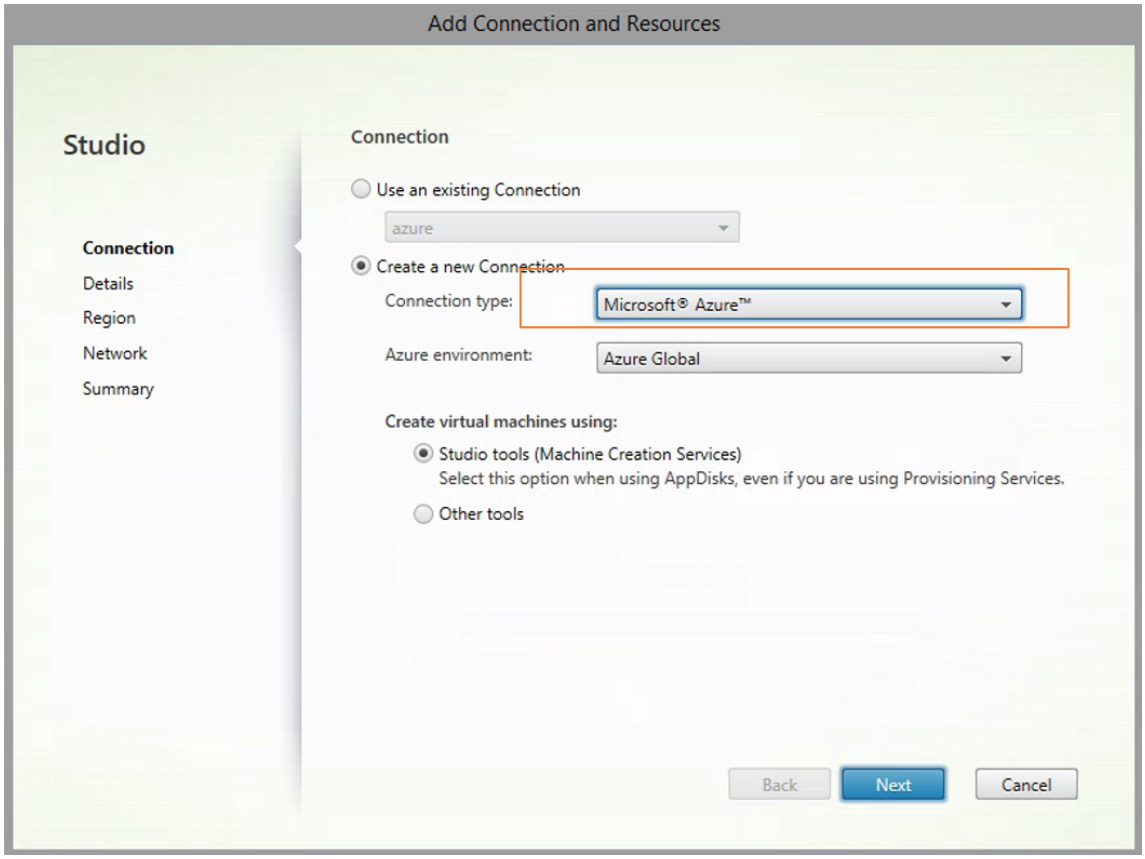
Azure での **MCS** を使用した **Linux** 仮想マシンの作成

手順 **1**：**Citrix Studio** での **Azure** へのホスティング接続の作成

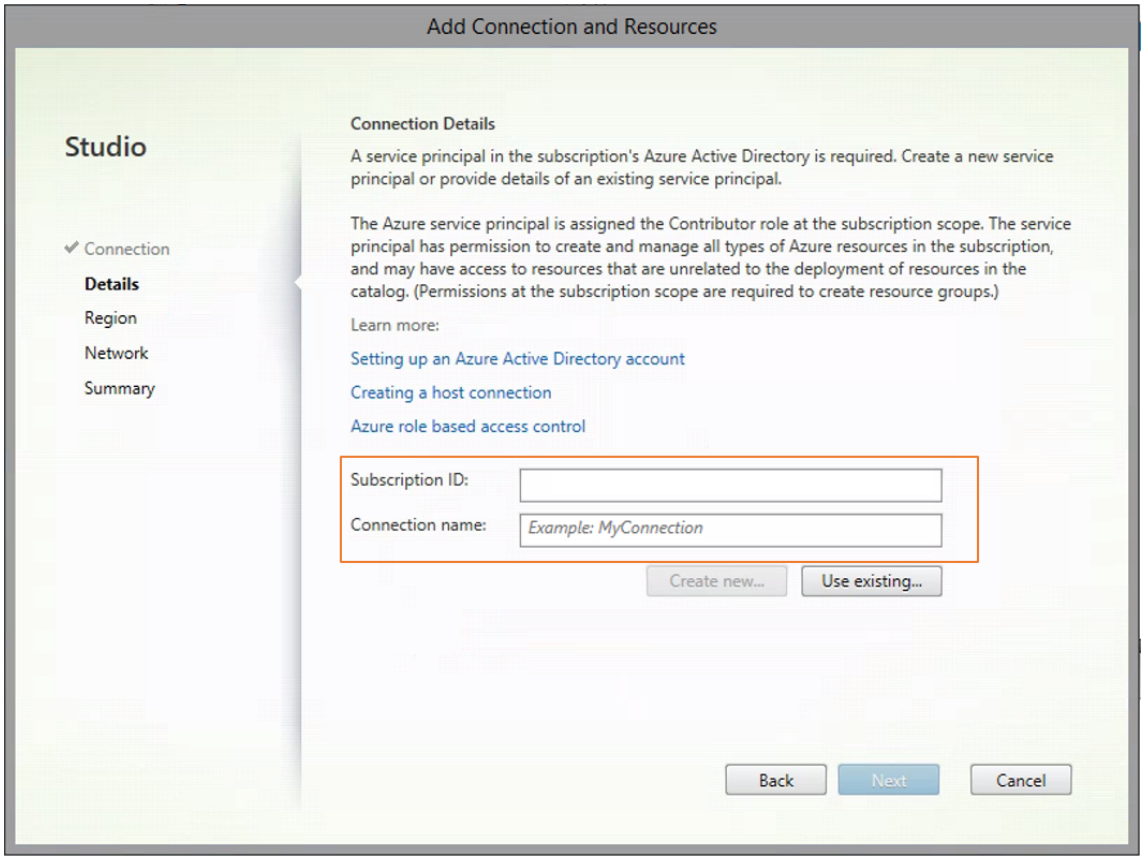
1. Citrix Studio で、［構成］>［ホスト］>［接続およびリソースの追加］の順に選択して、Azure への接続を作成します。



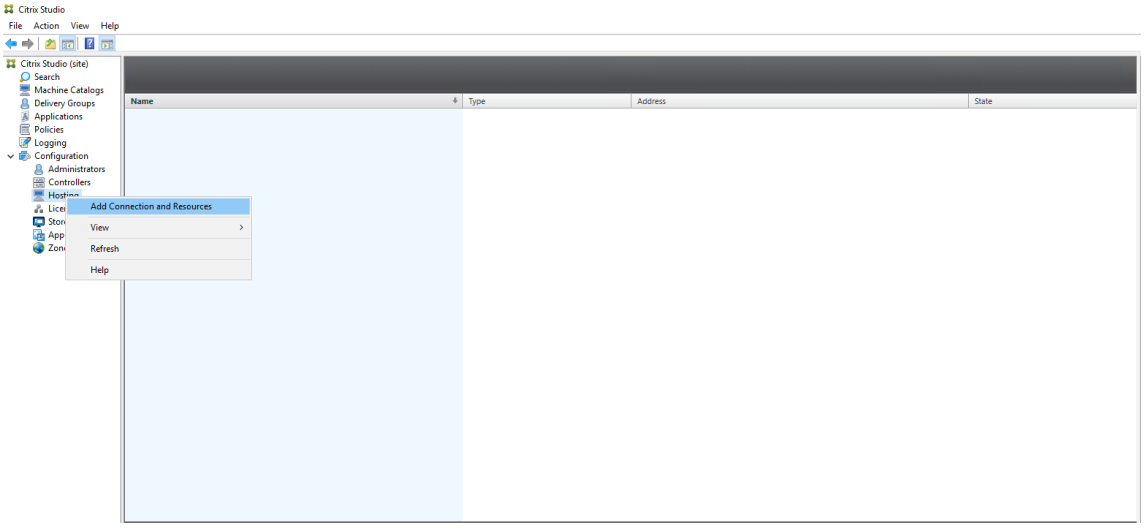
2. 接続の種類として［Microsoft Azure］を選択します。



3. Azure アカウントのサブスクリプション ID と接続名を入力します。



新しい接続がホストペインに表示されます。



手順 **2**：テンプレート仮想マシンでマスターイメージを準備

マスターイメージには、オペレーティングシステム、仮想化しないアプリケーション、VDA などのソフトウェアをインストールしておきます。マスターイメージを準備するには、次の手順を実行します。

手順 **2a**：**Ubuntu 18.04** 用に **cloud-init** を構成する　仮想マシンの再起動または停止時に VDA ホスト名を維持するには、次のコマンドを実行します。

```
echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
<!--NeedCopy-->
```

/etc/cloud/cloud.cfg ファイルの **system_info** セクションの下に次の行があることを確認します。

```
system_info:
   network:
       renderers: ['netplan', 'eni', 'sysconfig']
<!--NeedCopy-->
```

手順 **2b**：テンプレート仮想マシンに **Linux VDA** パッケージをインストールする

> 注：
>
> 現在実行中の VDA をテンプレート仮想マシンとして使用するには、この手順を省略します。
>
> テンプレート仮想マシンに Linux VDA パッケージをインストールする前に、.NET Core ランタイム 3.1 をインストールします。詳しくは、「インストールの概要」を参照してください。

使用している Linux ディストリビューションごとに、次のコマンドを実行して、Linux VDA の環境をセットアップします。

RHEL/CentOS の場合：

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

Ubuntu の場合：

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

apt-get install -f
<!--NeedCopy-->
```

SUSE 12 の場合：

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

手順 **2c**：リポジトリを有効にして **tdb-tools** パッケージをインストールする　**RHEL 7** サーバーの場合：

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
<!--NeedCopy-->
```

RHEL 7 ワークステーションの場合：


```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
<!--NeedCopy-->
```

手順 **2d**：**ntfs-3g** が含まれる **EPEL** リポジトリをインストールする　EPEL リポジトリを RHEL 6/CentOS 6、RHEL 7/CentOS 7 にインストールし、後から deploymcs.sh を実行すると ntfs-3g パッケージがインストールされるようにします。

手順 **2e**：**SUSE 12** に **ntfs-3g** を手動でインストールする　SUSE 12 プラットフォームには、ntfs-3g を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、ntfs-3g を手動でインストールします：

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

   ```
   sudo zypper install gcc
   sudo zypper install make
   <!--NeedCopy-->
   ```

2. ntfs-3g パッケージをダウンロードします。

3. ntfs-3g パッケージを展開します。

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

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>
   <!--NeedCopy-->
   ```

5. ntfs-3g をインストールします：

   ```
   ./configure
   make
   make install
   <!--NeedCopy-->
   ```

手順 **2f**：ランタイム環境をセットアップする　**deploymcs.sh** の実行前に、次の操作を行います：

- **/etc/xdl/mcs/mcs.conf** の変数を変更します。**mcs.conf** 構成ファイルには、MCS と Linux VDA を設定するための変数が含まれています。以下は、`dns`と`AD_INTEGRATION`を設定する必要がある変数の一部です：

  注：変数が複数の値で設定できる場合は、値を一重引用符で囲み、スペースで区切ります。たとえば、LDAP_LIST=’aaa.lab:389 bbb.lab:389’のように表示されます。

  - `Use_Existing_Configurations_Of_Current_VDA`：現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在

実行中の VDA の構成ファイルと同じファイルになります。ただし、dns変数とAD_INTEGRATION変数を構成する必要があります。デフォルト値は N です。これは、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。

- dns：DNS の IP アドレスを設定します。
- AD_INTEGRATION：Winbind または SSSD を設定します（SSSD は SUSE ではサポートされていません）。
- WORKGROUP：AD で構成されている場合、そのワークグループ名を設定します。これは NetBIOS 名です（大文字と小文字を区別）。それ以外の場合は、デフォルトではドメイン名です。

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

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

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

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

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

手順 **2g**：マスターイメージを作成する

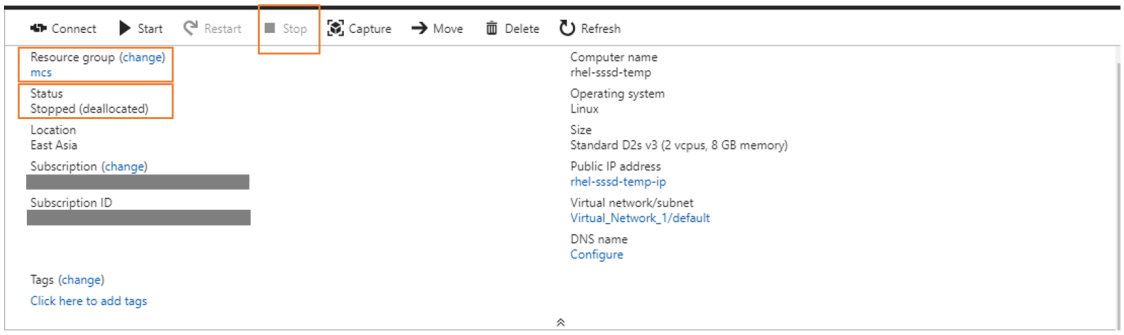
1. **/opt/Citrix/VDA/sbin/deploymcs.sh** を実行します。
2. （オプション）テンプレート仮想マシン上で構成テンプレートを更新して、作成されたすべての仮想マシン上の/etc/krb5.conf、/etc/samba/smb.conf、および/etc/sssd/sssd.conf ファイルを適宜カスタマイズします。

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

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

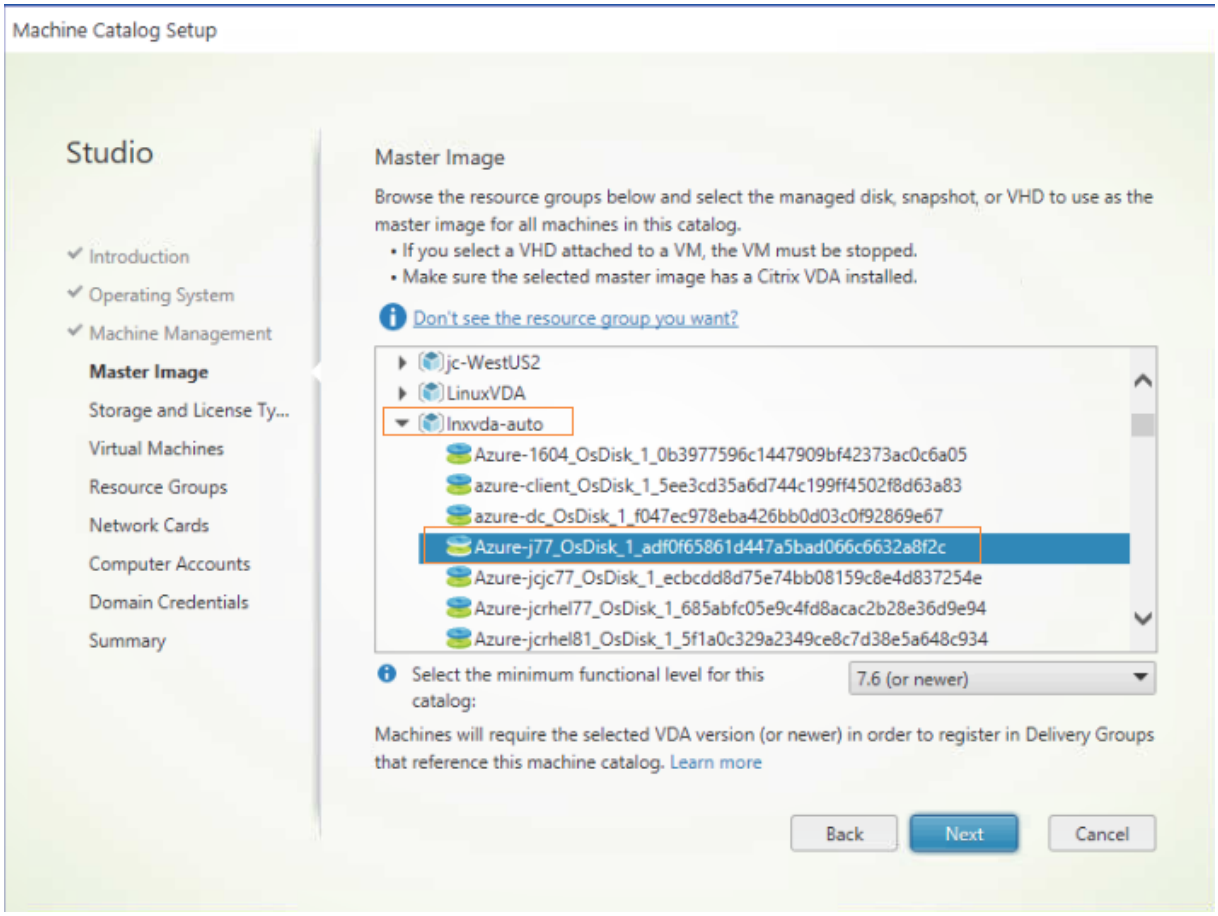
   注：テンプレートファイルで使用されている既存の形式を保持し、$WORKGROUP、$REALM、$realm、および $AD_FQDN などの変数を使用してください。

3. テンプレート仮想マシンにアプリケーションをインストールし、Azure Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンの電源状態が、[**Stopped**（**deallocated**)] になっていることを確認します。ここでリソースグループの名前を覚えておいてください。Azure でマスターイメージを検索する際に名前が必要です。



手順 **3**：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仮想マシンの数を指定します。マシンカタログを作成するときは、テンプレート仮想マシンが属するリソースグループからマスターイメージを選択し、テンプレート仮想マシンの VHD を探します。以下のスクリーンショットを参照してください。



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

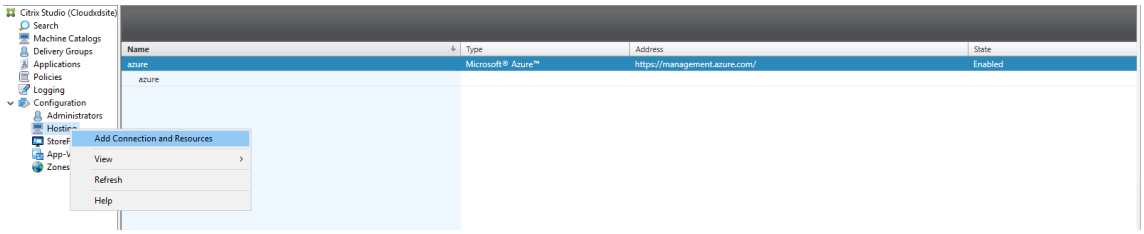
手順 **4**：デリバリーグループの作成

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

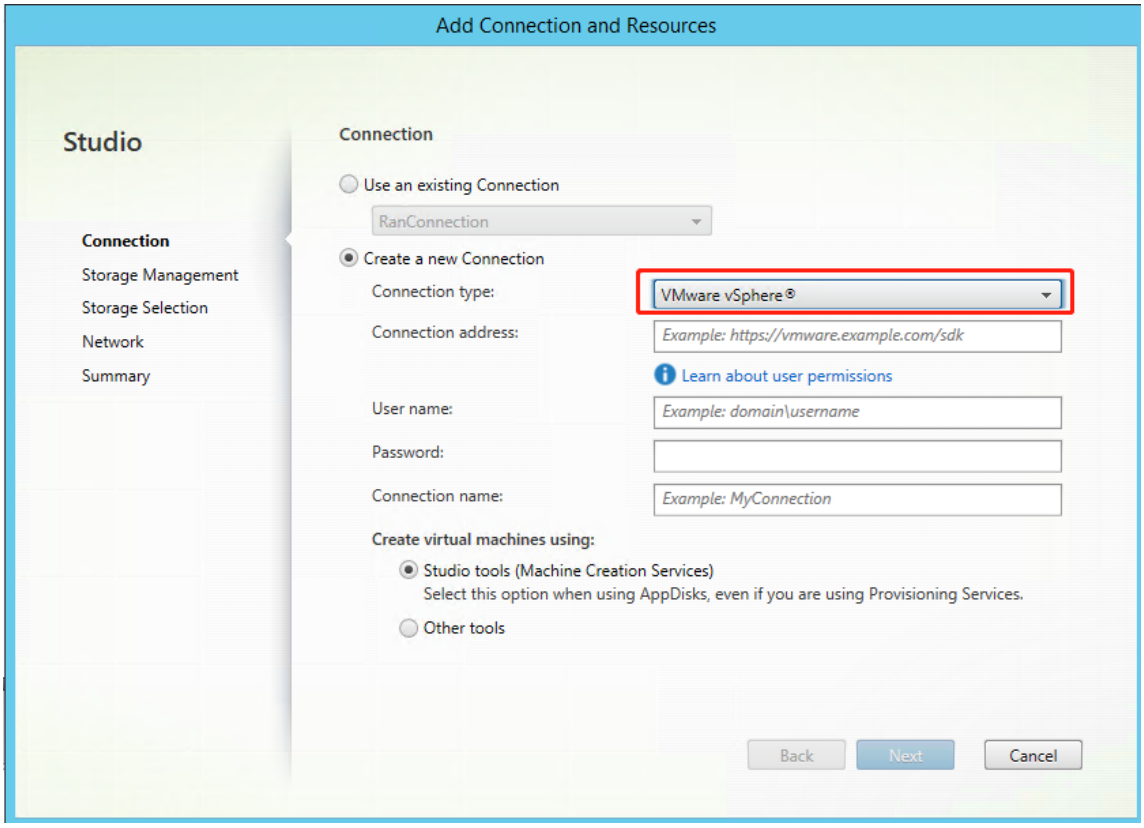
VMware vSphere での **MCS** を使用した **Linux** 仮想マシンの作成

手順 **1**：**Citrix Studio** での **VMware** へのホスティング接続の作成

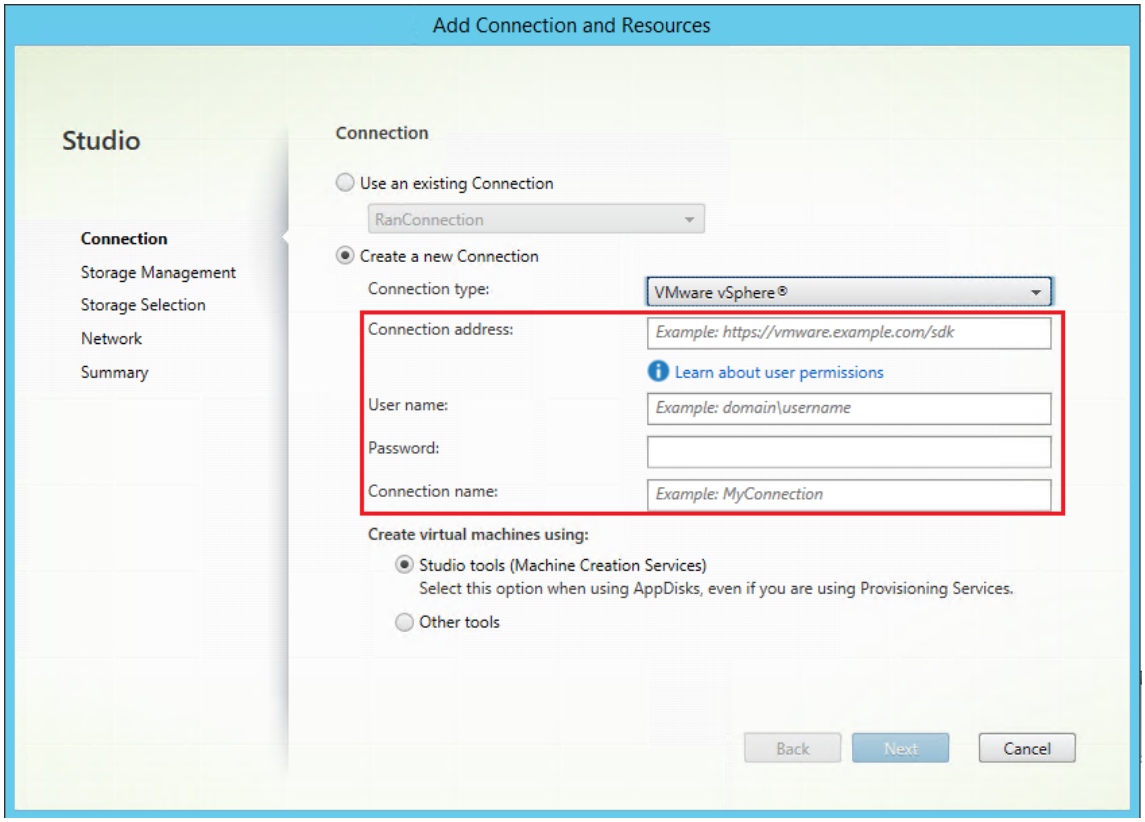
1. vSphere 環境に vCenter Server をインストールします。詳しくは、「VMware vSphere」を参照してください。
2. Citrix Studio で、［構成］>［ホスト］>［接続およびリソースの追加］の順に選択して、VMware vSphere への接続を作成します。



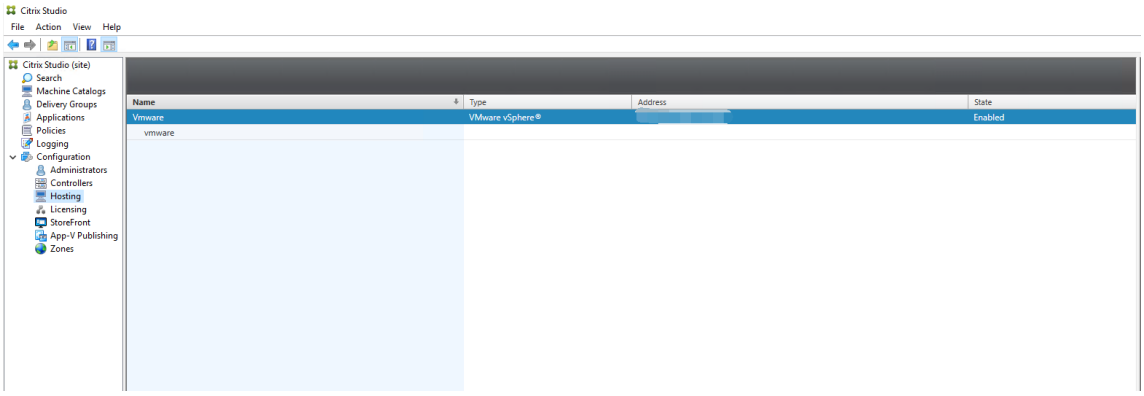
3. 接続の種類として［VMware vSphere］を選択します。



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



新しい接続がホストペインに表示されます。



手順 **2**：マスターイメージの準備

マスターイメージには、オペレーティングシステム、仮想化しないアプリケーション、VDA などのソフトウェアをインストールしておきます。マスターイメージを準備するには、次の手順を実行します。

手順 **2a**：テンプレート仮想マシンで **Linux VDA** パッケージをインストールする

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

> テンプレート仮想マシンに Linux VDA パッケージをインストールする前に、.NET Core ランタイム 3.1 をインストールします。詳しくは、「インストールの概要」を参照してください。

使用している Linux ディストリビューションごとに、次のコマンドを実行して、Linux VDA の環境をセットアップします。

RHEL/CentOS の場合:

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

Ubuntu の場合:

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

apt-get install -f
<!--NeedCopy-->
```

SUSE 12 の場合:

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

手順 **2b**: リポジトリを有効にして **tdb-tools** パッケージをインストールする **RHEL 7** サーバーの場合:

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
<!--NeedCopy-->
```

RHEL 7 ワークステーションの場合:

```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
<!--NeedCopy-->
```

手順 **2c**: **ntfs-3g** が含まれる **EPEL** リポジトリをインストールする EPEL リポジトリを RHEL 6/CentOS 6、RHEL 7/CentOS 7 にインストールし、後から deploymcs.sh を実行すると ntfs-3g パッケージがインストールされるようにします。

手順 **2d**: **SUSE 12** に **ntfs-3g** を手動でインストールする SUSE 12 プラットフォームには、ntfs-3g を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、ntfs-3g を手動でインストールします:

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

   ```
   sudo zypper install gcc
   sudo zypper install make
   <!--NeedCopy-->
   ```

2. ntfs-3g パッケージをダウンロードします。

3. ntfs-3g パッケージを展開します。

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

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>
<!--NeedCopy-->
```

5. ntfs-3g をインストールします:

```
./configure
make
make install
<!--NeedCopy-->
```

手順 **2e**: ランタイム環境をセットアップする　**deploymcs.sh** の実行前に、次の操作を行います:

- **/etc/xdl/mcs/mcs.conf** の変数を変更します。**mcs.conf** 構成ファイルには、MCS と Linux VDA を設定するための変数が含まれています。以下は、`dns`と`AD_INTEGRATION`を設定する必要がある変数の一部です:

  注: 変数が複数の値で設定できる場合は、値を一重引用符で囲み、スペースで区切ります。たとえば、LDAP_LIST='aaa.lab:389 bbb.lab:389' のように表示されます。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在実行中の VDA の構成ファイルと同じファイルになります。ただし、`dns`変数と`AD_INTEGRATION`変数を構成する必要があります。デフォルト値は N です。これは、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。
  - `dns`: DNS の IP アドレスを設定します。
  - `AD_INTEGRATION`: Winbind または SSSD を設定します（SSSD は SUSE ではサポートされていません）。
  - `WORKGROUP`: AD で構成されている場合、そのワークグループ名を設定します。これは NetBIOS 名です（大文字と小文字を区別）。それ以外の場合は、デフォルトではドメイン名です。

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

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

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


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

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

手順 **2f**：マスターイメージを作成する

1. **/opt/Citrix/VDA/sbin/deploymcs.sh** を実行します。
2. （オプション）テンプレート仮想マシン上で構成テンプレートを更新して、作成されたすべての仮想マシン上の/etc/krb5.conf、/etc/samba/smb.conf、および/etc/sssd/sssd.conf ファイルを適宜カスタマイズします。

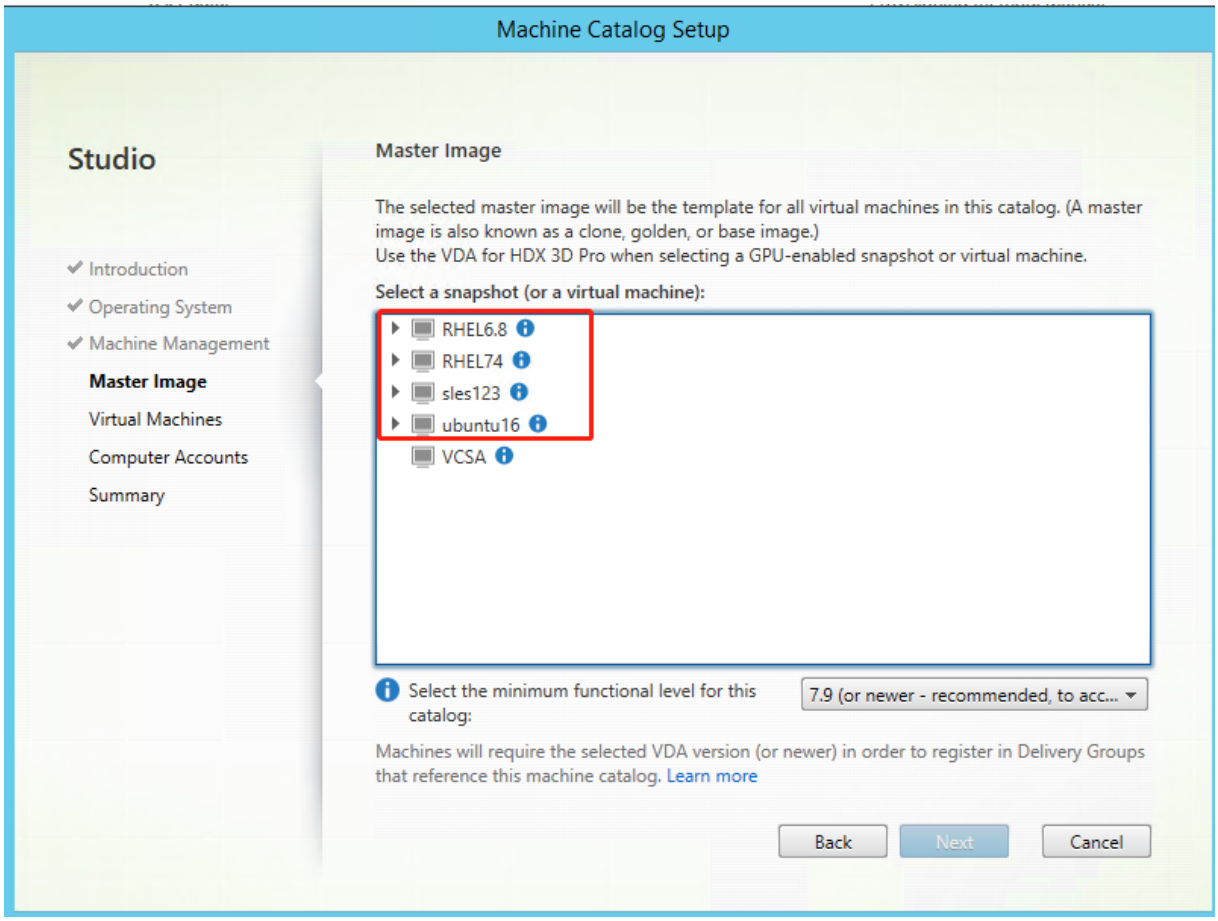
   Winbind ユーザーの場合、/etc/xdl/mcs/winbind_krb5.conf.tmpl および/etc/xdl/mcs/winbind_smb.conf.tmpl の各テンプレートを更新します。

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

   注：テンプレートファイルで使用されている既存の形式を保持し、$WORKGROUP、$REALM、$realm、および $AD_FQDN などの変数を使用してください。
3. テンプレート仮想マシンにアプリケーションをインストールしたら、VMware でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンのスナップショットを作成します。

手順 **3**：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仮想マシンの数を指定します。マシンカタログを作成するときは、スナップショットリストからマスターイメージを選択します。



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

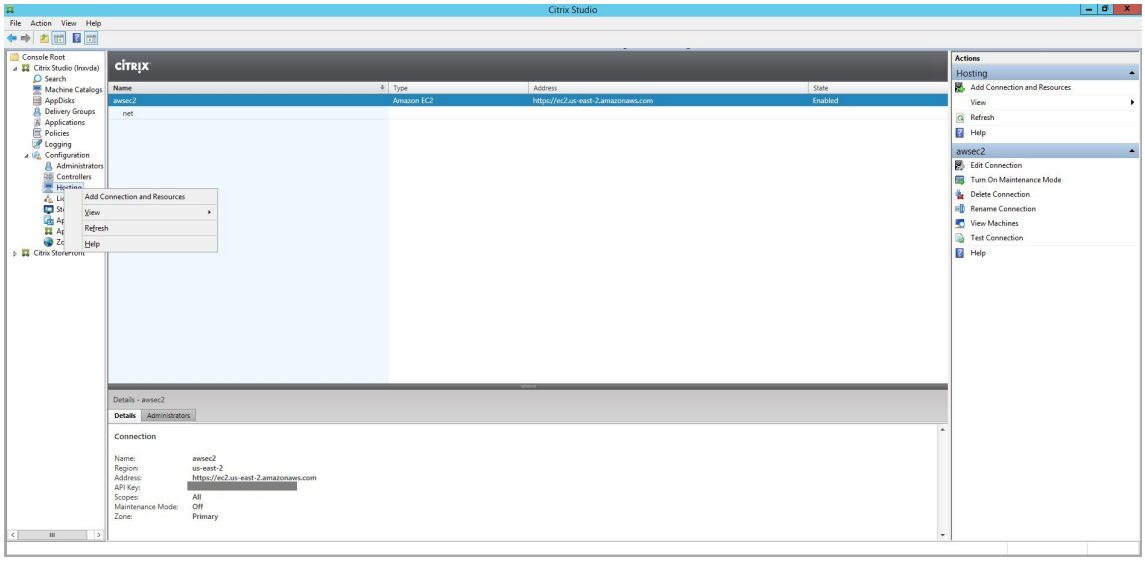
手順 **4**: デリバリーグループの作成

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

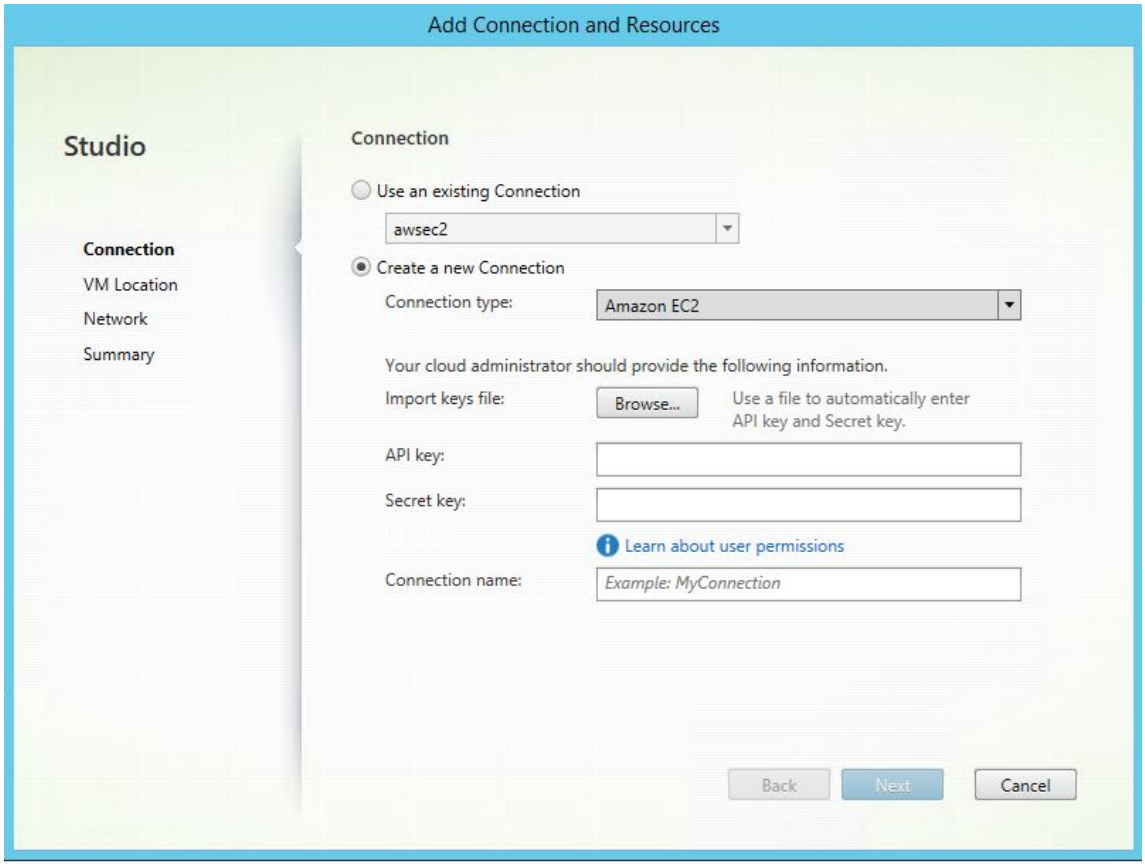
AWS での **MCS** を使用した **Linux** 仮想マシンの作成

手順 **1**: **Citrix Studio** での **AWS** へのホスティング接続の作成

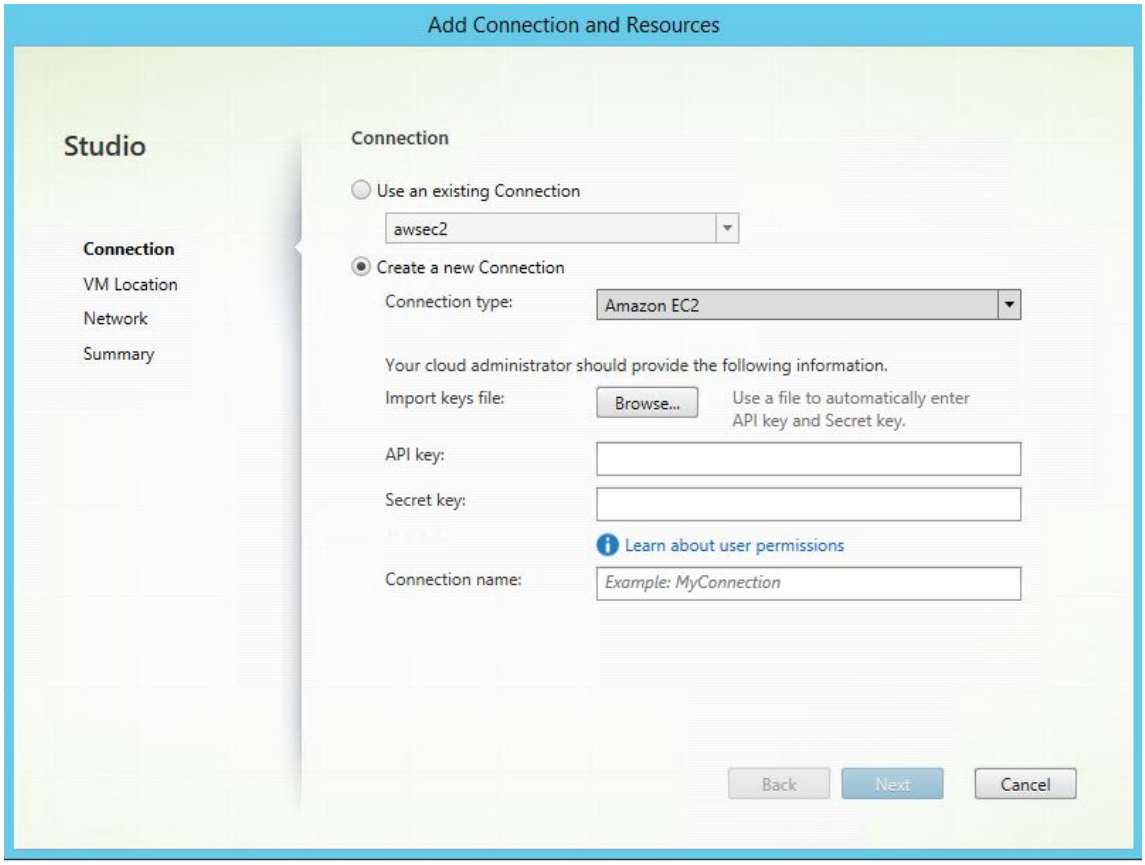
1. Citrix Studio で、[構成] > [ホスト] > [接続およびリソースの追加] の順に選択して、AWS への接続を作成します。



2. 接続の種類として ［**Amazon EC2**］を選択します。



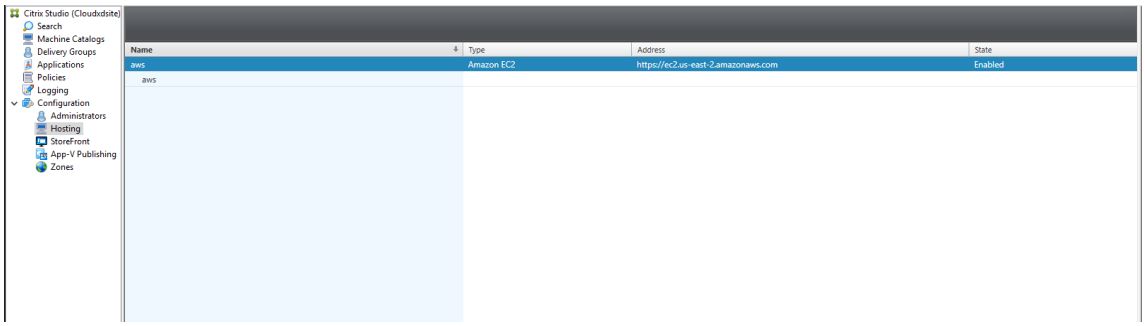
3. AWS アカウントの API キーと秘密キーを入力し、接続名を入力します。



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

a) AWS サービスにサインインします。
b) ID およびアクセス管理（IAM）コンソールに移動します。
c) 左側のナビゲーションペインで、［**Users**］を選択します。
d) 対象ユーザーを選択して下にスクロールして、［**Security credentials**］タブを選択します。
e) 下にスクロールして、［**Create access key**］をクリックします。新しいウィンドウが開きます。
f) ［**Download .csv file**］をクリックし、アクセスキーを安全な場所に保存します。

新しい接続がホストペインに表示されます。



手順 **2**：マスターイメージの準備

マスターイメージには、オペレーティングシステム、仮想化しないアプリケーション、VDA などのソフトウェアをインストールしておきます。マスターイメージを準備するには、次の手順を実行します。

手順 **2a**：**cloud-init** を構成する

1. EC2 インスタンスの再起動または停止時に VDA ホスト名を保持するには、次のコマンドを実行して VDA ホスト名を保持します。

```
echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
<!--NeedCopy-->
```

   Ubuntu 18.04 の場合、/etc/cloud/cloud.cfg ファイルの system_info セクションの下に次の行があることを確認します。

```
system_info:
    network:
        renderers: ['netplan', 'eni', 'sysconfig']
<!--NeedCopy-->
```

2. AWS で MCS が作成した仮想マシンに SSH を使用してリモートアクセスする場合、これらの仮想マシン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。必要に応じて次の操作を実行します。

   - cloud-init 構成ファイル/etc/cloud/cloud.cfg を編集します。**ssh_pwauth: true** 行が存在することを確認します。**set-password** 行と次の行が存在する場合は、その行を削除するか、コメントを追加します。

```
users:
- default
<!--NeedCopy-->
```

   - 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。新しいパスワードを記録して、MCS が作成した仮想マシンにログインするときに使用できるようにします。

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

```
PasswordAuthentication yes
<!--NeedCopy-->
```

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

手順 **2b**：テンプレート仮想マシンに **Linux VDA** パッケージをインストールする

> 注:
>
> 現在実行中の VDA をテンプレート仮想マシンとして使用するには、この手順を省略します。
>
> テンプレート仮想マシンに Linux VDA パッケージをインストールする前に、.NET Core ランタイム 3.1 をインストールします。詳しくは、「インストールの概要」を参照してください。

使用している Linux ディストリビューションごとに、次のコマンドを実行して、Linux VDA の環境をセットアップします。

RHEL/CentOS の場合:

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

Ubuntu の場合:

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

apt-get install -f
<!--NeedCopy-->
```

SUSE 12 の場合:

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

手順 **2c**: リポジトリを有効にして **tdb-tools** パッケージをインストールする **RHEL 7** サーバーの場合:

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
<!--NeedCopy-->
```

RHEL 7 ワークステーションの場合:

```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
<!--NeedCopy-->
```

手順 **2d**: **ntfs-3g** が含まれる **EPEL** リポジトリをインストールする EPEL リポジトリを RHEL 6/CentOS 6、RHEL 7/CentOS 7 にインストールし、後から deploymcs.sh を実行すると ntfs-3g パッケージがインストールされるようにします。

手順 **2e**: **SUSE 12** に **ntfs-3g** を手動でインストールする SUSE 12 プラットフォームには、ntfs-3g を提供するリポジトリがありません。ソースコードをダウンロードし、コンパイルし、ntfs-3g を手動でインストールします:

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


```
sudo zypper install gcc
sudo zypper install make
<!--NeedCopy-->
```

2. ntfs-3g パッケージをダウンロードします。
3. ntfs-3g パッケージを展開します。

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

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>
<!--NeedCopy-->
```

5. ntfs-3g をインストールします:

```
./configure
make
make install
<!--NeedCopy-->
```

手順 **2f**: ランタイム環境をセットアップする　**deploymcs.sh** の実行前に、次の操作を行います:

- **/etc/xdl/mcs/mcs.conf** の変数を変更します。**mcs.conf** 構成ファイルには、MCS と Linux VDA を設定するための変数が含まれています。以下は、`dns`と`AD_INTEGRATION`を設定する必要がある変数の一部です:

  注: 変数が複数の値で設定できる場合は、値を一重引用符で囲み、スペースで区切ります。たとえば、LDAP_LIST=’aaa.lab:389 bbb.lab:389’のように表示されます。

  - `Use_Existing_Configurations_Of_Current_VDA`: 現在実行中の VDA の既存の構成を使用するかどうかを決定します。Y に設定すると、MCS で作成されたマシンの構成ファイルは、現在実行中の VDA の構成ファイルと同じファイルになります。ただし、`dns`変数と`AD_INTEGRATION`変数を構成する必要があります。デフォルト値は N です。これは、MCS が作成したマシン上の構成ファイルがマスターイメージ上の構成テンプレートによって決定されることを意味します。
  - `dns`: DNS の IP アドレスを設定します。
  - `AD_INTEGRATION`: Winbind または SSSD を設定します（SSSD は SUSE ではサポートされていません）。
  - `WORKGROUP`: AD で構成されている場合、そのワークグループ名を設定します。これは NetBIOS 名です（大文字と小文字を区別）。それ以外の場合は、デフォルトではドメイン名です。

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

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

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

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

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

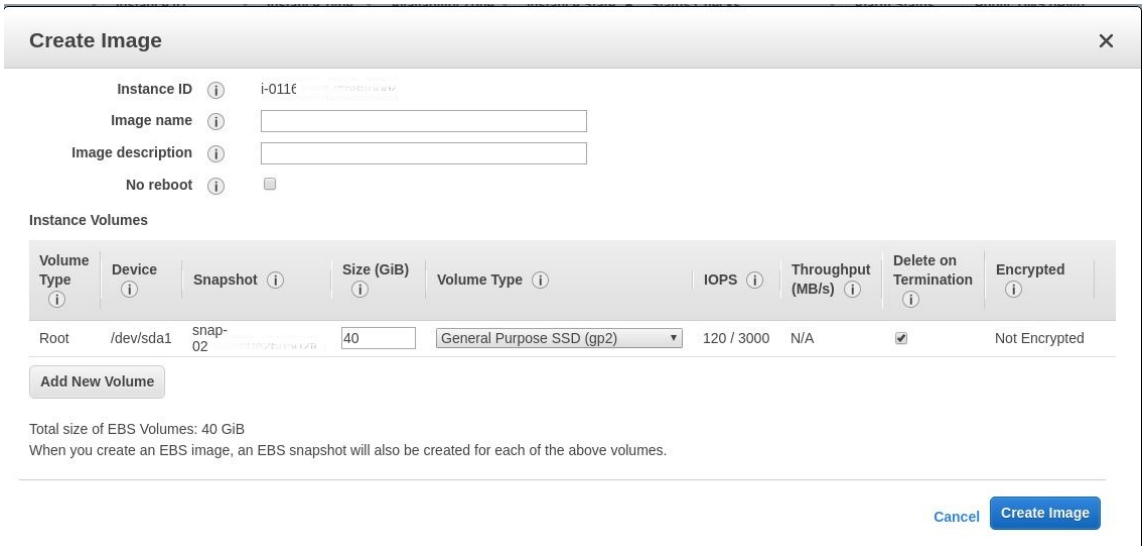
手順 **2g**: マスターイメージを作成する

1. **/opt/Citrix/VDA/sbin/deploymcs.sh** を実行します。
2. (オプション) テンプレート仮想マシン上で構成テンプレートを更新して、作成されたすべての仮想マシン上の/etc/krb5.conf、/etc/samba/smb.conf、および/etc/sssd/sssd.conf ファイルを適宜カスタマイズします。

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

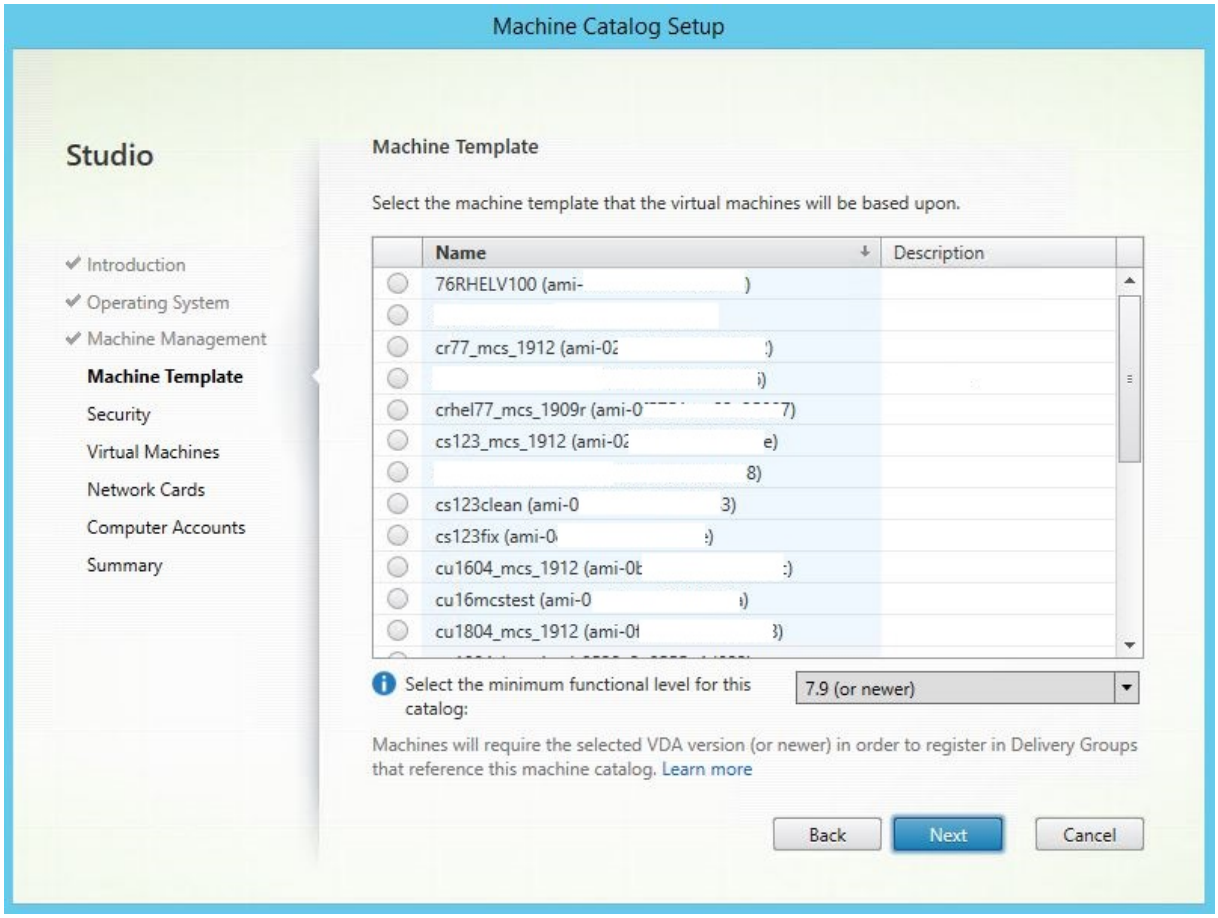
   SSSD ユーザーの場合、/etc/xdl/mcs/sssd.conf.tmpl、/etc/xdl/mcs/sssd_krb5.conf.tmpl、/etc/xdl/mcs/sssd_smb.conf.tmpl の各テンプレートを更新します。

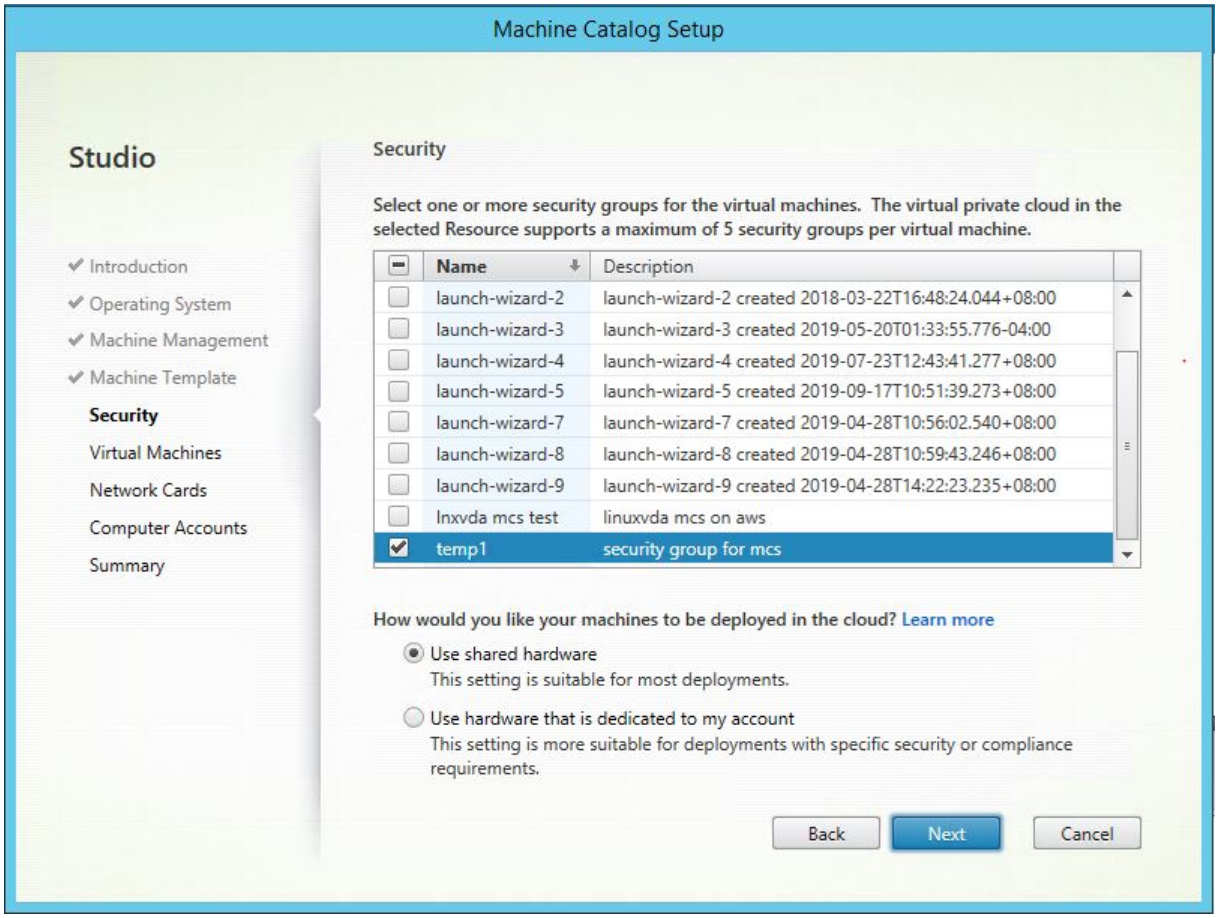
   注: テンプレートファイルで使用されている既存の形式を保持し、$WORKGROUP、$REALM、$realm、および $AD_FQDN などの変数を使用してください。
3. テンプレート仮想マシンにアプリケーションをインストールし、AWS EC2 Portal でテンプレート仮想マシンをシャットダウンします。テンプレート仮想マシンのインスタンス状態が、[**Stopped**] になっていることを確認します。
4. テンプレート仮想マシンを右クリックし、[**Image**] > [**Create Image**] を選択します。必要に応じて情報を入力し、設定を行います。[**Create Image**] をクリックします。



手順 **3**：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仮想マシンの数を指定します。マシンカタログの作成時に、マシンテンプレート（上記で作成したマスターイメージ）を選択し、1 つまたは複数のセキュリティグループを選択します。





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

手順 **4**: デリバリーグループの作成

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

MCS を使用した **Linux VDA** のアップグレード

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

1. テンプレートマシンで Linux VDA をアップグレードします:

   RHEL 7/CentOS 7 の場合:

   ```
   sudo rpm -U XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
   <!--NeedCopy-->
   ```

   RHEL 6/CentOS 6 の場合:


```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
<!--NeedCopy-->
```

SUSE 12 の場合:

```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
<!--NeedCopy-->
```

Ubuntu 16.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu16.04_amd64.deb
<!--NeedCopy-->
```

Ubuntu 18.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu18.04_amd64.deb
<!--NeedCopy-->
```

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

Delivery Controller の構成

May 13, 2020

XenDesktop 7.6 以前のバージョンで Linux VDA をサポートするには、変更を加える必要があります。そのため、これらのバージョンでは、Hotfix またはアップデートスクリプトが必要です。これらのインストールと確認については、このセクションで説明しています。

Delivery Controller 構成の更新

XenDesktop 7.6 SP2 の場合、Hotfix Update 2 を適用して、Linux Virtual Desktop 用のブローカーを更新します。Hotfix Update 2 は、以下から入手できます。

- CTX142438: Hotfix Update 2 - Delivery Controller 7.6（32 ビット）用 - 英語
- CTX142439: Hotfix Update 2 - Delivery Controller 7.6（64 ビット）用 - 英語

XenDesktop 7.6 SP2 より前のバージョンでは、**Update-BrokerServiceConfig.ps1** という名前の PowerShell スクリプトを使用してブローカーサービスの構成を更新できます。このスクリプトは次のパッケージから入手できます。

- citrix-linuxvda-scripts.zip

次の手順をサーバーファーム内の各 Delivery Controller で繰り返します:

1. **Update-BrokerServiceConfig.ps1** スクリプトを Delivery Controller マシンにコピーします。
2. ローカル管理者のコンテキストで Windows PowerShell コンソールを開きます。
3. **Update-BrokerServiceConfig.ps1** スクリプトを含むフォルダーを参照します。
4. **Update-BrokerServiceConfig.ps1** スクリプトを実行します:

```
.\Update-BrokerServiceConfig.ps1
<!--NeedCopy-->
```

> ヒント:
>
> デフォルトでは、PowerShell は PowerShell スクリプトを実行できないように構成されています。スクリプトの実行に失敗する場合は、再試行する前に PowerShell 実行ポリシーを変更します。
>
> ```
> Set-ExecutionPolicy Unrestricted
> <!--NeedCopy-->
> ```

Update-BrokerServiceConfig.ps1 スクリプトを実行すると、Linux VDA に必要とされる新しい WCF エンドポイントを使用してブローカーサービス構成ファイルが更新され、ブローカーサービスが再起動します。このスクリプトでは、自動的にブローカーサービス構成ファイルの場所が特定されます。元の構成ファイルのバックアップが、**.prelinux** という拡張子のファイル名で同じディレクトリに作成されます。

これらの変更は、同じ Delivery Controller ファームを使用するように構成された Windows VDA の仲介には影響しません。単一の Controller ファームは、Windows VDA と Linux VDA の両方とのセッションをシームレスに管理し、仲介できます。

> 注:
>
> Linux VDA は、暗号化で Secure ICA をサポートしません。Linux VDA で Secure ICA を有効にすると、セッションの起動に失敗します。

Delivery Controller 構成の確認

必要な構成変更が Delivery Controller に適用されているかどうかを確認するには、**%PROGRAMFILES%\Citrix\Broker\Service\BrokerService.exe.config** ファイル中に **EndpointLinux** ストリングが 5 回出現していることを確認します。

Windows コマンドプロンプトで、ローカル管理者としてログオンし、以下を確認します。

```
cd "%PROGRAMFILES%"\Citrix\Broker\Service\
findstr EndpointLinux BrokerService.exe.config
<!--NeedCopy-->
```


Linux VDA の構成

November 21, 2020

このセクションでは、機能の説明、構成、トラブルシューティングなど、Linux VDA の機能について詳しく説明します。

> ヒント:
>
> 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、/opt/Citrix/VDA/bin に配置されます。Linux VDA をインストールした後、`bash /opt/Citrix/VDA/bin/xdlcollect.sh`コマンドを実行してログを収集できます。
>
> ログ収集の完了後、圧縮されたログファイルが同じフォルダーにスクリプトとして生成されます。`xdlcollect`が、圧縮されたログファイルを Citrix Insight Services（CIS）にアップロードするかを確認します。同意した場合、`xdlcollect`はアップロードが完了した後に upload_ID を返します。アップロードしても、圧縮されたログファイルはローカルマシンから削除されません。他のユーザーは、upload_ID を使用して CIS にあるログファイルにアクセスできます。

NIS の Active Directory との統合

November 11, 2021

このトピックでは、SSSD を使用して、NIS を Linux VDA の Windows Active Directory（AD）と統合する方法について説明します。Linux VDA は、Citrix Virtual Apps and Desktops のコンポーネントと見なされます。そのため Linux VDA は、Windows AD 環境に密接に結びついています。

AD の代わりに NIS を UID および GID プロバイダーとして使用するには、AD と NIS でユーザー名とパスワードの組み合わせのアカウント情報を同一にする必要があります。

> 注:
>
> NIS を使用した場合も、認証は AD サーバーにより行われます。NIS+ はサポートされません。NIS を UID および GID プロバイダーとして使用する場合、Windows サーバーからの POSIX 属性は使用されません。
>
> ヒント:
>
> これは、Linux VDA を展開する方法として廃止済みであるため、特定のユースケースでのみ使用してください。RHEL/CentOS ディストリビューションの場合は、「Linux Virtual Delivery Agent for RHEL/CentOS のインストール」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、「Linux Virtual Delivery Agent for Ubuntu のインストール」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。SSSD の主な機能は、システムにキャッシュとオフラインサポートを提供する共通フレームワークを通じて、リモートリソースの識別および認証のアクセスを提供することです。PAM や NSS モジュールを提供しており、将来的には D-BUS ベースのインターフェイスもサポートして、拡張ユーザー情報に対応する予定です。また、ローカルユーザーアカウントと拡張ユーザー情報を保存するための優れたデータベースを提供します。

必要なソフトウェア

AD プロバイダーは、SSSD Version 1.9.0 で初めて導入されました。

次の環境については、このドキュメントに記載した指示を使用したテストおよび検証を行っています:

- RHEL 7.7 以降
- CentOS 7.7 以降

NIS と **AD** の統合

NIS と AD を統合するには、次の手順を実行します:

1. Linux VDA を NIS クライアントとして追加
2. ドメインに参加し、Samba を使用してホストの keytab を作成
3. SSSD のセットアップ
4. NSS/PAM の構成
5. Kerberos 構成の確認
6. ユーザー認証の確認

Linux VDA を **NIS** クライアントとして追加

NIS クライアントを構成します。

```
yum -y install ypbind rpcbind oddjob-mkhomedir
<!--NeedCopy-->
```

NIS ドメインを設定します。

```
ypdomainname nis.domain
echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

nis.domain は、NIS サーバーのドメイン名です。**server.nis.domain** は、NIS サーバーのホスト名であり、NIS サーバーの IP アドレスにもできます。

NIS のサービスを設定します。

```
sudo systemctl start rpcbind ypbind

sudo systemctl enable rpcbind ypbind
<!--NeedCopy-->
```

NIS の構成が正しいことを確認します。

```
ypwhich
<!--NeedCopy-->
```

NIS サーバーからアカウント情報が使用できることを確認します。

```
getent passwd nisaccount
<!--NeedCopy-->
```

> 注:
>
> **nisaccount** は、NIS サーバーの実際の NIS アカウントです。UID、GID、ホームディレクトリ、およびログインシェルが正しく設定されていることを確認します。

ドメインに参加し、**Samba** を使用してホストの **keytab** を作成

SSSD では、ドメイン参加とシステムの keytab ファイルの管理に関する AD のクライアント機能が提供されていません。この機能を取得するには次のような方法があります:

- adcli
- realmd
- Winbind
- Samba

このセクションでは、Samba によるアプローチについてのみ説明します。**realmd** については、RHEL または CentOS のベンダーのドキュメントを参照してください。SSSD を構成する前に、以下の手順に従う必要があります。

ドメインに参加し、**Samba** を使用してホストの **keytab** を作成する:

Linux クライアントで、適切に構成されたファイルを使用します。

- /etc/krb5.conf

- /etc/samba/smb.conf:

Samba および Kerberos 認証用にマシンを構成します:

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

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

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

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

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

```
kerberos method = secrets and keytab
```

Windows ドメインに参加するには、ドメインコントローラーに到達できることと、コンピューターをドメインに追加する権限を持つ AD ユーザーアカウントが必要です。

```
sudo net ads join REALM -U user
<!--NeedCopy-->
```

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

SSSD のセットアップ

SSSD のセットアップは、以下の手順で構成されています:

- Linux クライアントマシンに **sssd-ad** パッケージおよび **sssd-proxy** パッケージをインストールします。
- さまざまなファイルに設定の変更を行います（**sssd.conf** など）。
- **sssd** サービスを開始します。

/etc/sssd/sssd.conf　**sssd.conf** の設定の例（必要に応じて追加の設定を行うことができます）:

```
[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
<!--NeedCopy-->
```

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

ファイルの所有権およびアクセス権を **sssd.conf** で設定します:

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

NSS/PAM の構成

RHEL/CentOS:

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

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

sudo systemctl start sssd

sudo systemctl enable sssd
<!--NeedCopy-->
```


> ヒント：
>
> Linux VDA の設定を行うときは、SSSD では Linux VDA クライアントの特別な設定がないことを考慮します。**ctxsetup.sh** スクリプトでのその他の解決方法としては、デフォルト値を使用します。

Kerberos 構成の確認

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

```
sudo klist -ke
<!--NeedCopy-->
```

このコマンドにより、プリンシパル名と暗号スイートのさまざまな組み合わせに対して使用できるキーの一覧が表示されます。Kerberos の **kinit** コマンドを実行し、これらのキーを使用して、マシンをドメインコントローラーに対して認証します：

```
sudo kinit  - k MACHINE$@REALM
<!--NeedCopy-->
```

マシン名と領域名は大文字で指定する必要があります。ドル記号（$）は、シェルによる置き換えを防ぐためにバックスラッシュ（\）でエスケープする必要があります。環境によっては、DNS ドメイン名が Kerberos 領域名と異なります。したがって、必ず領域名を使用します。このコマンドが成功すると、出力は表示されません。

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

```
sudo klist -ke
<!--NeedCopy-->
```

ユーザー認証の確認

getent コマンドを使用して、ログオン形式がサポートされていること、および NSS が機能するかどうかを確認します：

```
sudo getent passwd DOMAIN\username
<!--NeedCopy-->
```

DOMAIN パラメーターは短い形式のドメイン名です。別のログオン形式が必要な場合は、まず **getent** コマンドを使用して確認します。

サポートされているログオン形式は次の通りです：

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

SSSD PAM モジュールが正しく構成されていることを確認するには、ドメインユーザーアカウントを使用して Linux VDA にログオンします。以前はドメインユーザーアカウントは使用されていませんでした。

```
sudo ssh localhost  - l DOMAIN\username

id -u
<!--NeedCopy-->
```

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

```
ls /tmp/krb5cc_{
 uid }

<!--NeedCopy-->
```

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

```
klist
<!--NeedCopy-->
```

公開アプリケーション

July 8, 2022

Linux VDA バージョン 7.13 では、Citrix でシームレスアプリケーション機能がサポート対象のすべての Linux プラットフォームに追加されました。この機能を使用するのに特別なインストール手順は不要です。

> ヒント:
>
> Linux VDA Version 1.4 では、非シームレスな公開アプリケーションとセッションの共有のサポートが Citrix で追加されました。

Citrix Studio を使ってアプリケーションを公開する

デリバリーグループを作成したり、既存のデリバリーグループにアプリケーションを追加したりすると、Linux VDA にインストールしたアプリケーションを公開することができます。このプロセスは、Windows VDA にインストールしたアプリケーションを公開する場合と同様です。詳しくは、Citrix Virtual Apps and Desktops ドキュメント（使用中の Citrix Virtual Apps and Desktops のバージョン）を参照してください。

> ヒント:
>
> デリバリーグループの構成では、デリバリーの種類を ［デスクトップとアプリケーション］ または ［アプリケ

> ーション］に設定します。
>
> 重要:
>
> アプリケーションの公開は、Linux VDA バージョン 1.4 以降でサポートされています。ただし、同一マシンへのデスクトップおよびアプリの配信は、Linux VDA でサポートされていません。この問題に対処するには、アプリおよびデスクトップの配信で個別のデリバリーグループを作成することを Citrix ではお勧めします。
>
> 注:
>
> シームレスアプリケーションを使用するには、StoreFront でシームレスモードを無効にしないでください。シームレスモードは、デフォルトで有効になっています。既に「TWIMode=Off」を設定して無効にしている場合は、「TWIMode=On」に変更するのではなく、この設定を削除してください。削除しない場合は、公開デスクトップを起動できないことがあります。

制限事項

Linux VDA では、1 人のユーザーが同じアプリケーションの複数の同時インスタンスを起動することはできません。

既知の問題

アプリケーション公開時の既知の問題は次のとおりです:

- 非矩形のウィンドウはサポートされません。ウィンドウの隅にサーバー側の背景が表示されることがあります。
- ウィンドウの内容を公開アプリケーションからプレビューすることはサポートされていません。
- 現在、シームレスモードでは次のウィンドウマネージャーをサポートしています。Mutter、Metacity、および Compiz（Ubuntu 16.04）。Kwin およびその他のウィンドウマネージャーはサポートされていません。ウィンドウマネージャーが、サポートされているモードに設定されていることを確認してください。
- 複数の LibreOffice アプリケーションによってプロセスが共有されるため、Citrix Studio には最初に起動したもののみが表示されます。
- 「Dolphin」などの公開された Qt5 ベースのアプリケーションについてはアイコンが表示されないことがあります。この問題を解決するには、https://wiki.archlinux.org/title/Qtの記事を参照してください。
- 同じ ICA セッション内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同じグループに結合されます。この問題を解決するには、タスクバーボタンを結合しないようにタスクバーのプロパティを設定します。

リモート **PC** アクセス

November 11, 2021

概要

リモート PC アクセスは、Citrix Virtual Apps and Desktops の拡張機能です。これにより、組織は従業員が物理的なオフィス PC に安全な方法でリモートアクセスできるようにします。ユーザーが社内 PC にアクセスできる場合、作業に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

リモート PC アクセスでは、仮想デスクトップとアプリケーションを配信するのと同じ Citrix Virtual Apps and Desktops コンポーネントが使用されます。リモート PC アクセスの展開と構成の要件およびプロセスは、仮想リソース配信のための Citrix Virtual Apps and Desktops の展開に必要な要件およびプロセスと同じです。この統一性により、一貫性のある統一された管理エクスペリエンスが実現されます。ユーザーは、Citrix HDX を使用してリモートオフィス PC セッションを提供することで、最高のユーザーエクスペリエンスを実現できます。

詳しくは、Citrix Virtual Apps and Desktops ドキュメントの「リモート PC アクセス」を参照してください。

構成

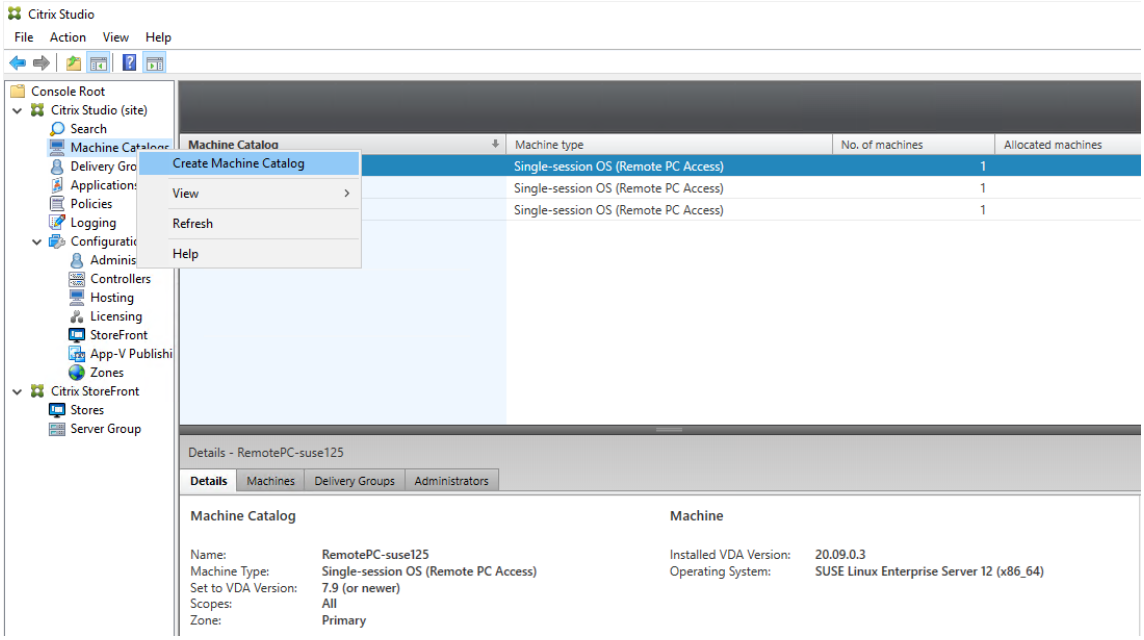
Linux PC セッションを配信するには、対象の PC に Linux VDA をインストールし、リモート **PC** アクセスタイプのマシンカタログを作成し、配信グループを作成して、アクセスを要求するユーザーがマシンカタログ内の PC を利用できるようにします。次のセクションでは、手順について詳しく説明します:

手順 **1 -** 対象の **PC** に **Linux VDA** をインストールする

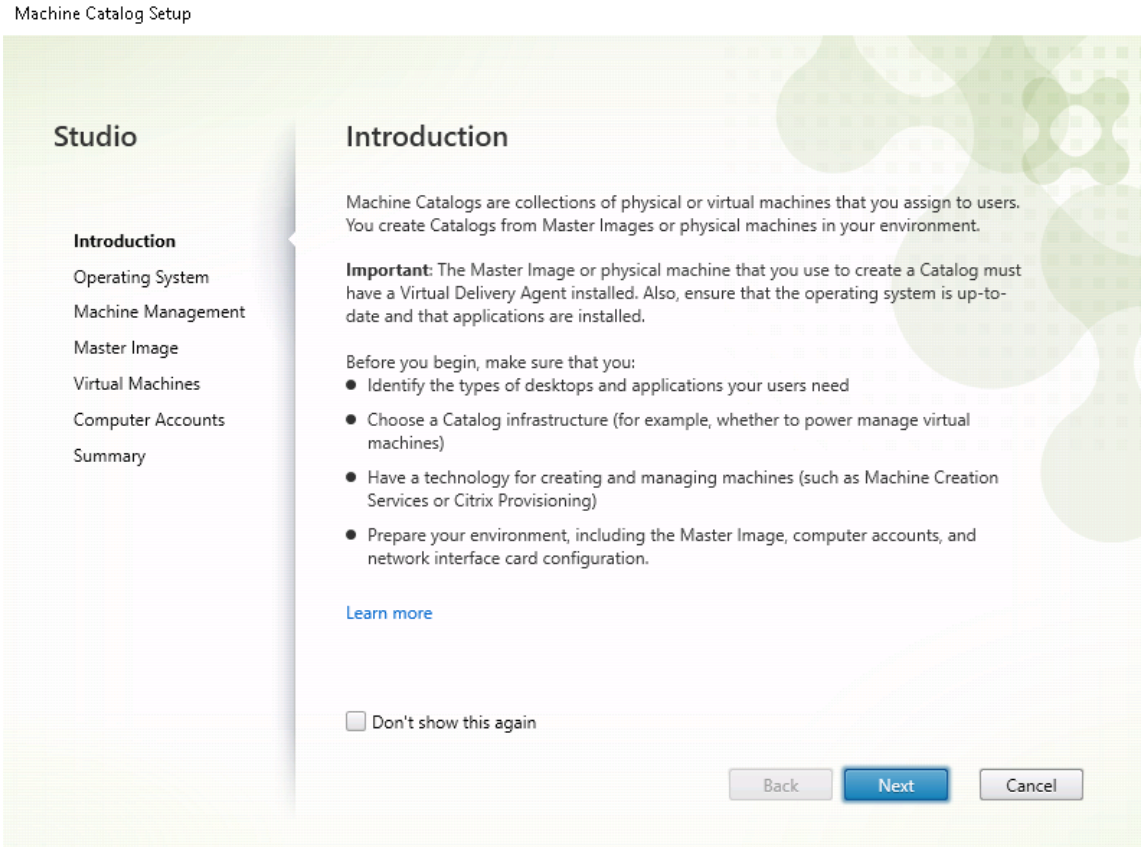
簡単インストールを使用して Linux VDA をインストールすることをお勧めします。インストール中、`CTX_XDL_VDI_MODE`変数の値を`Y`に設定します。

手順 **2 -** リモート **PC** アクセスタイプのマシンカタログを作成する

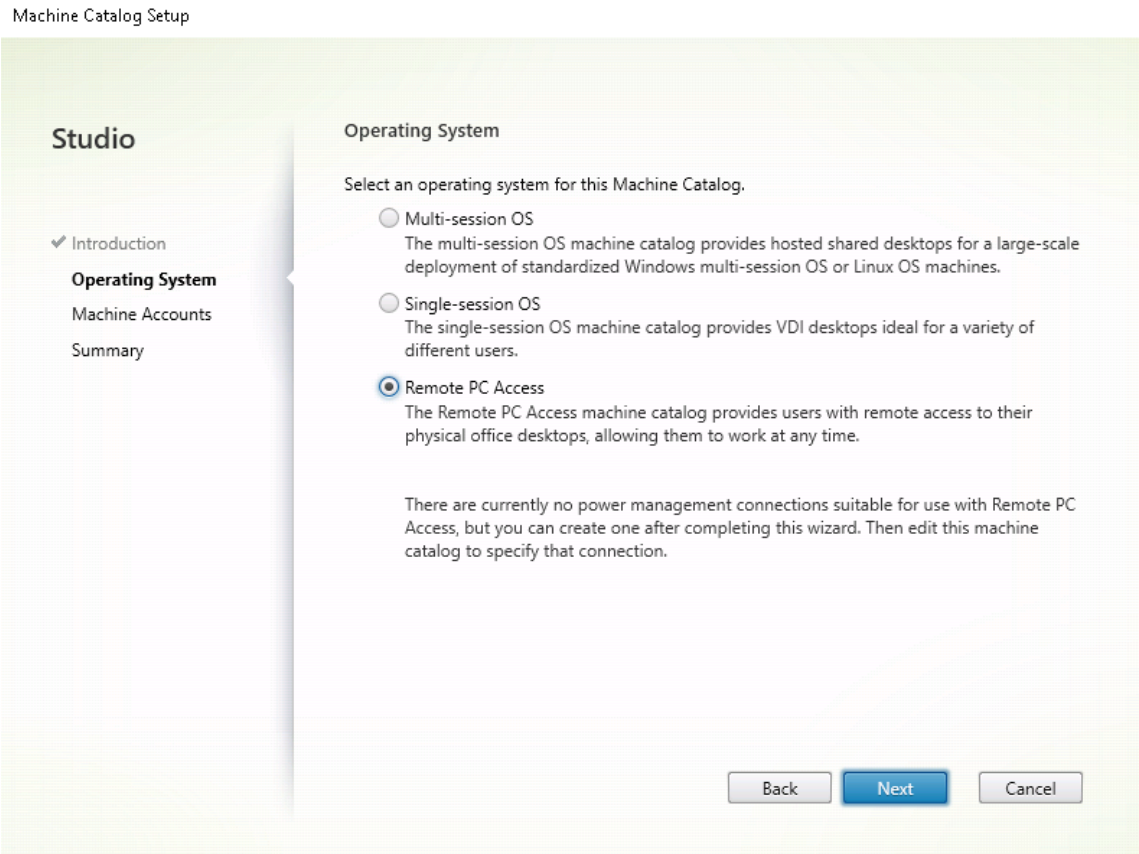
1. Citrix Studio で ［マシンカタログ］を右クリックし、ショートカットメニューから ［マシンカタログの作成］を選択します。



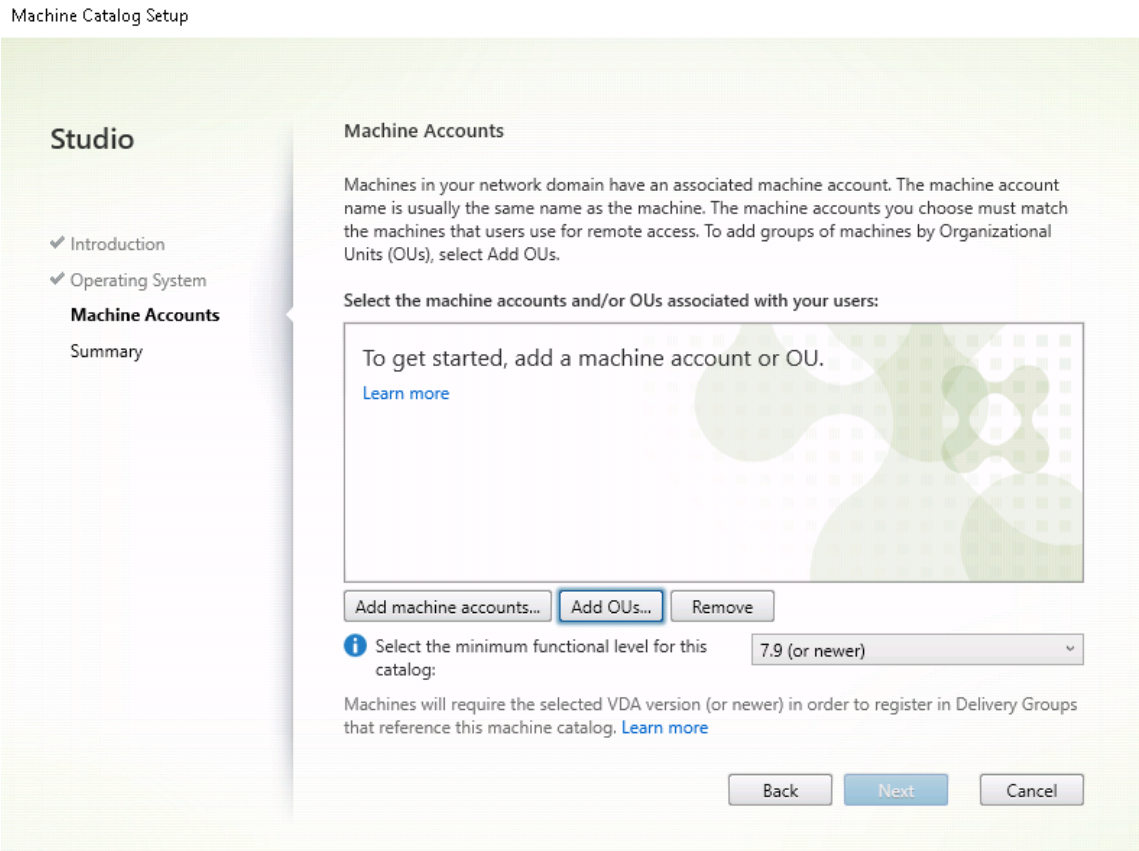
2. ［はじめに］ページで ［次へ］をクリックします。



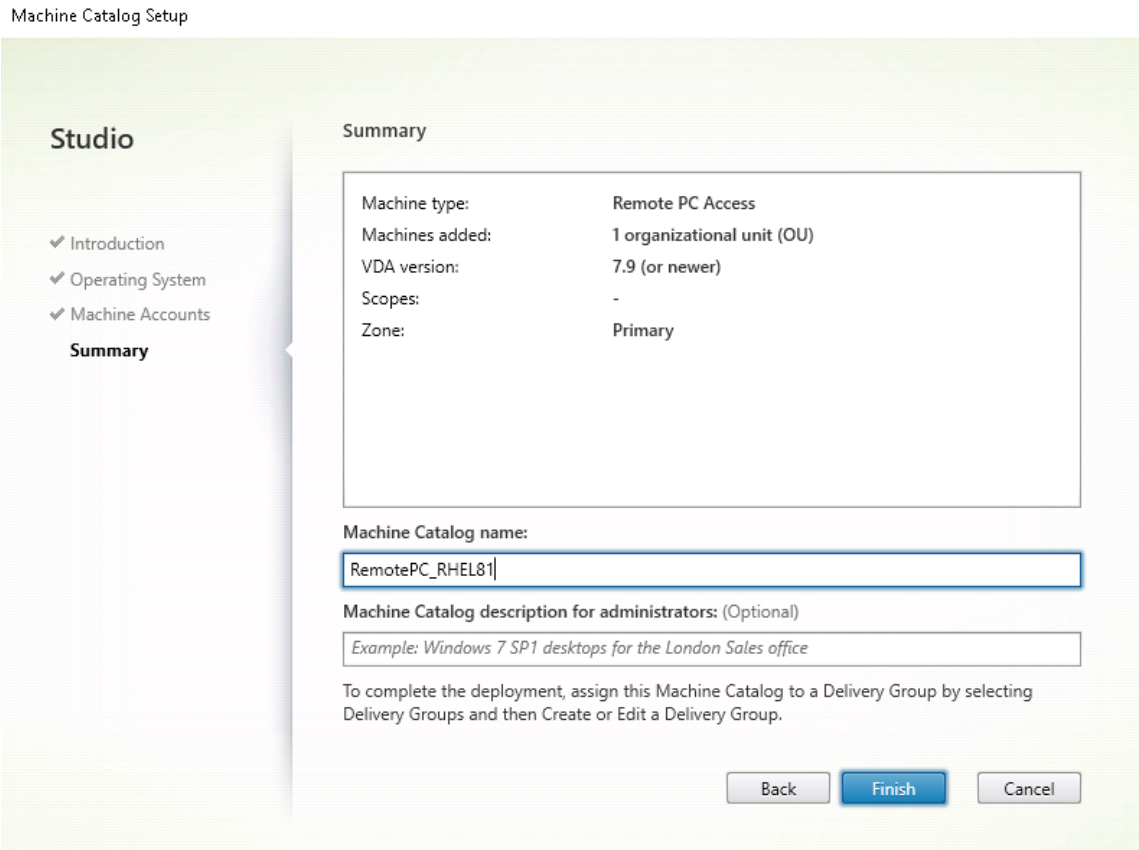
3. ［オペレーティングシステム］ページで ［リモート **PC** アクセス］を選択します。



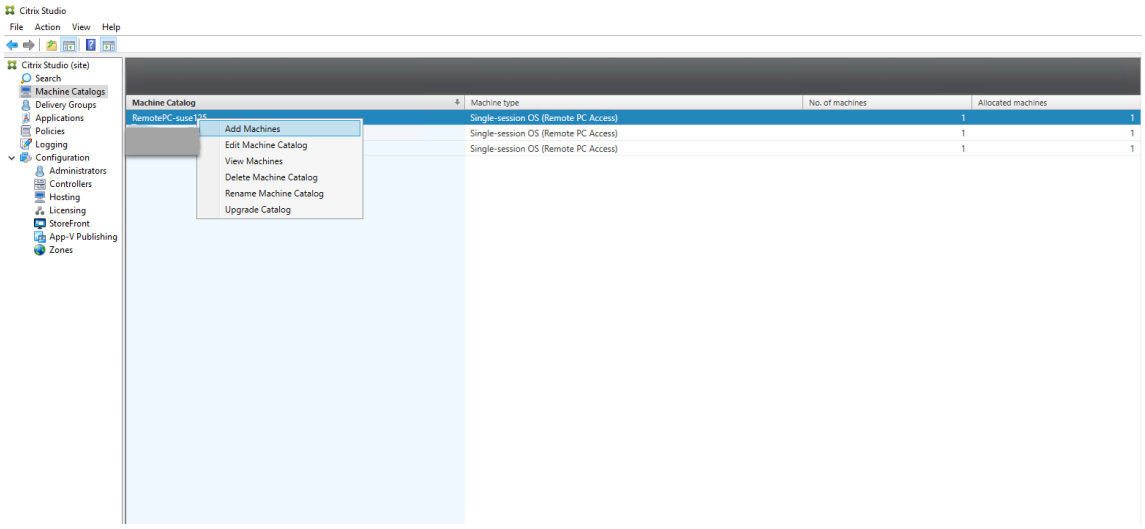
4. ［**OU** の追加］をクリックして対象の PC を含む OU を選択するか、［マシンアカウントの追加］をクリックして個別のマシンをマシンカタログに追加します。



5. マシンカタログに名前を付けます。

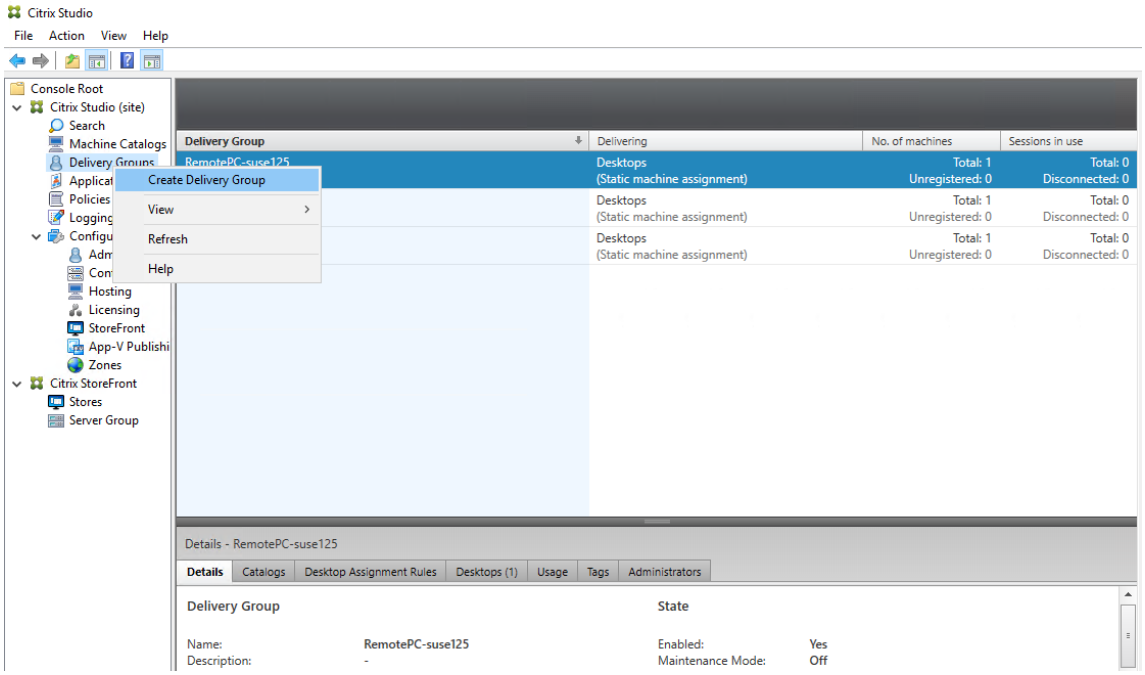


6. （オプション）マシンカタログを右クリックして、必要な操作を実行します。

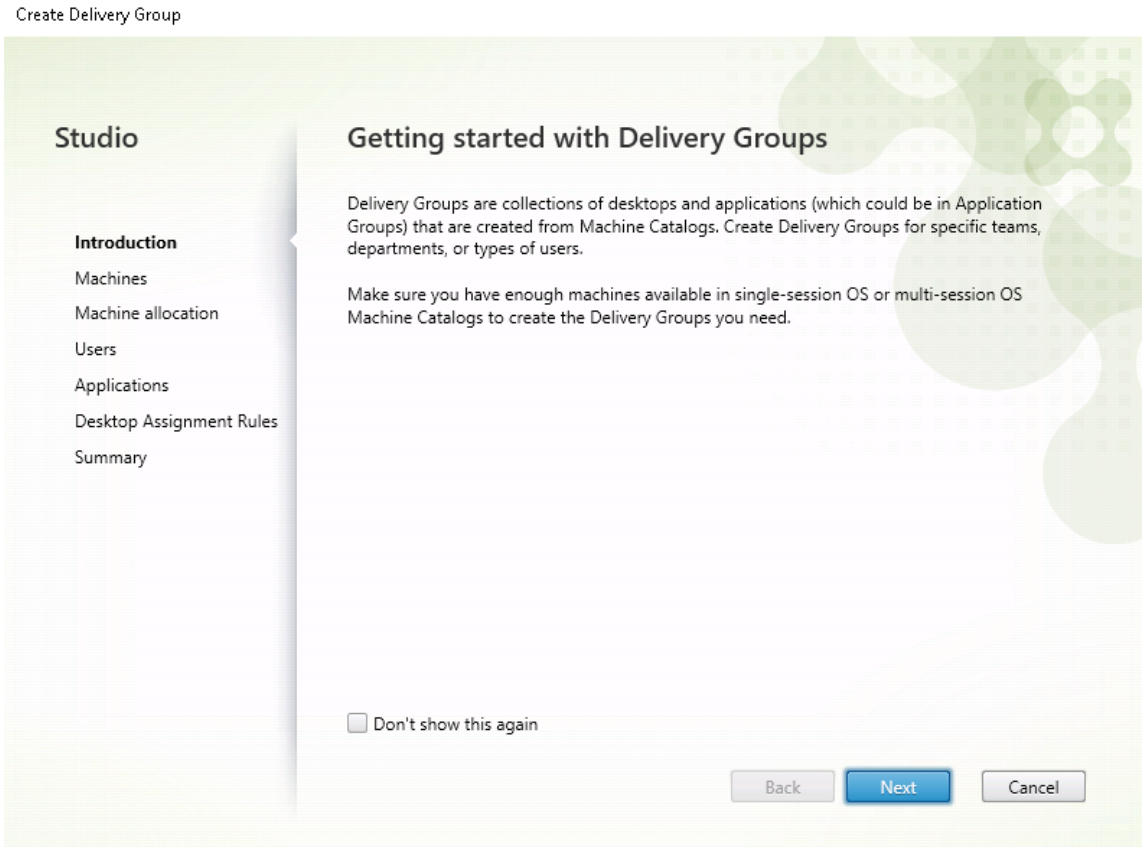


手順 **3 -** デリバリーグループを作成してアクセスを要求したユーザーがマシンカタログで **PC** を利用できるようにする

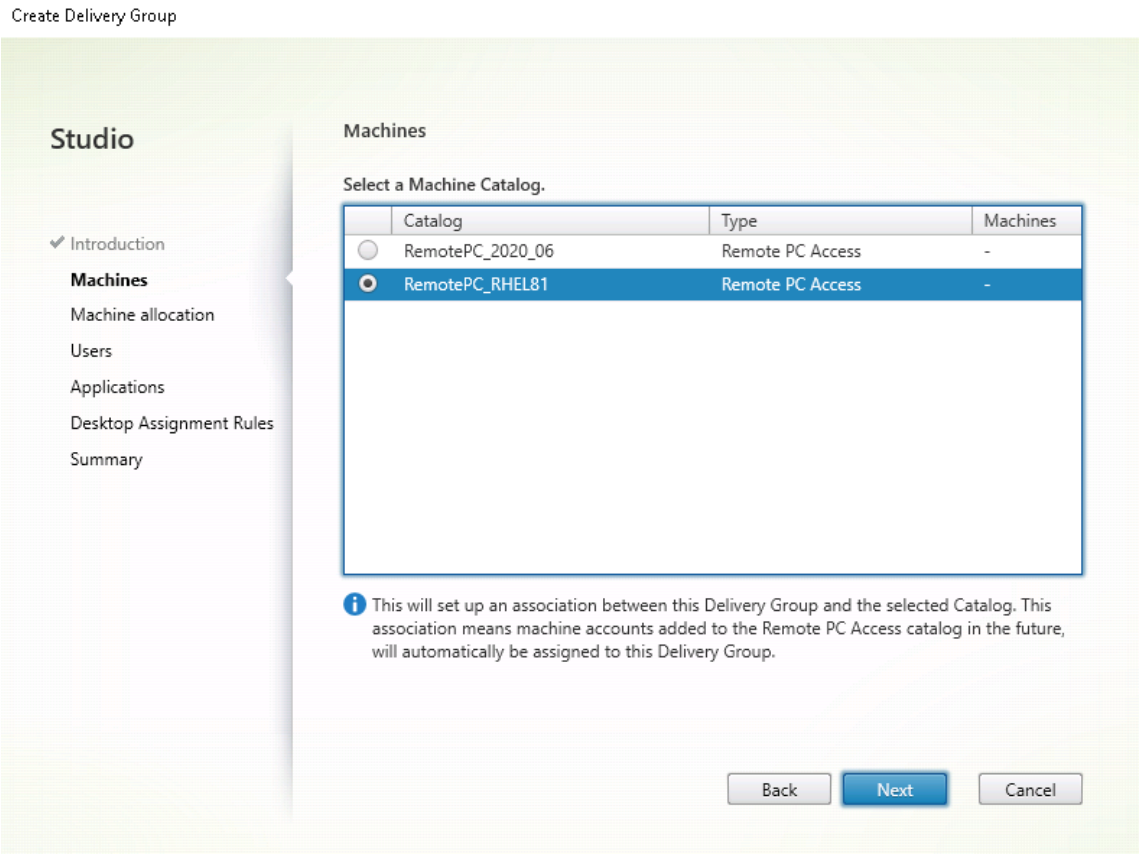
1. Citrix Studio で ［デリバリーグループ］を右クリックし、ショートカットメニューで ［デリバリーグループの作成］を選択します。



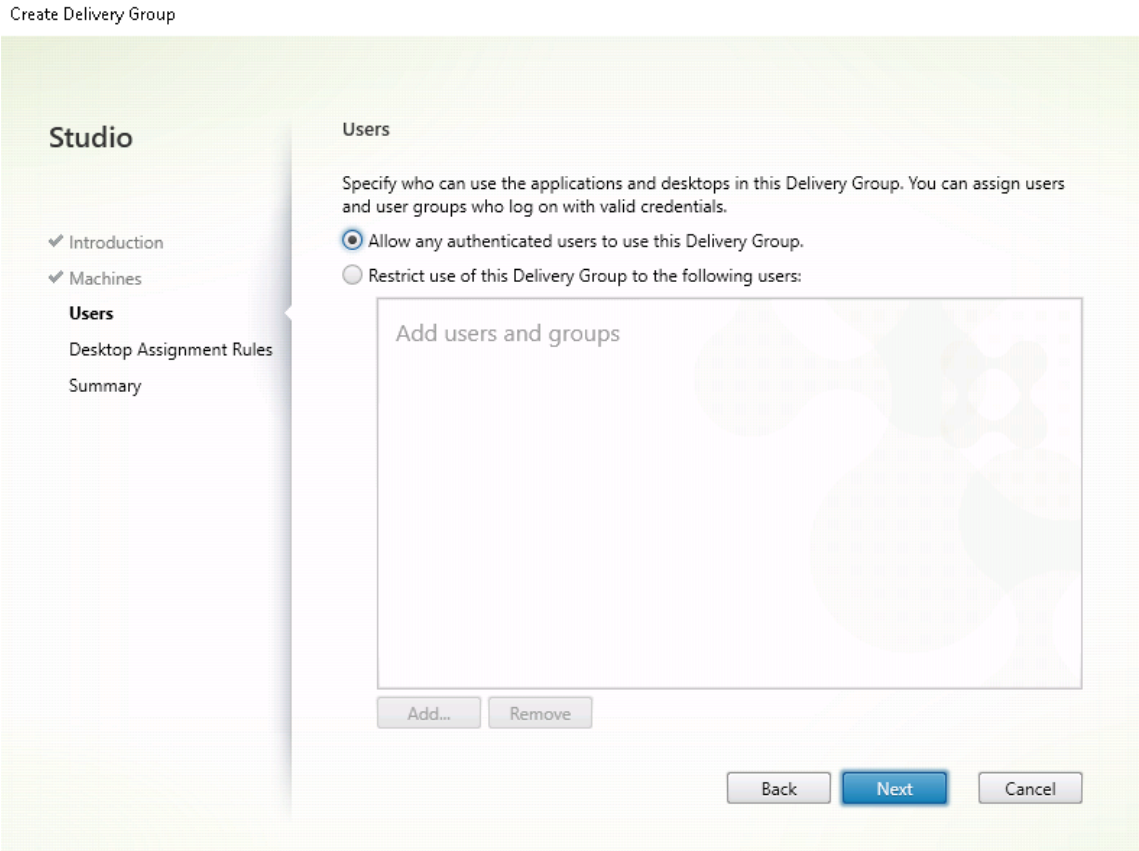
2. ［デリバリーグループの作成］ページで ［次へ］をクリックします。



3. 手順 2 で作成したマシンカタログを選択して、デリバリーグループに関連付けます。



4. PC にアクセスできるユーザーをマシンカタログに追加します。追加したユーザーは、クライアントデバイス上の Citrix Workspace アプリを使用して、PC にリモートでアクセスできます。



注意事項

次の考慮事項は、Linux VDA に固有のものです:

- Linux VDA は、非 3D モードの物理マシンでのみ使用します。NVIDIA のドライバーの制限により、HDX 3D モードが有効になっている場合、PC のローカル画面はブラックアウトせず、画面にはセッションのアクティビティが表示されます。この画面の表示は、セキュリティ上のリスクです。
- 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。
- 統合された Wake on LAN 機能は、Linux マシンでは使用できません。
- Linux マシンでは、自動ユーザー割り当ては使用できません。自動ユーザー割り当てを使用すると、ユーザーは PC にローカルでログオンしたときに、自分のマシンに自動的に割り当てられます。このログオンには、管理者による介在は必要ありません。クライアントデバイス上で動作する Citrix Workspace アプリにより、リモート PC アクセスセッションで社内の PC 上のアプリケーションやデータにアクセスできます。
- ユーザーが既にローカルで PC にログオンしている場合、StoreFront から PC を起動しようとすると失敗します。
- Linux マシンでは、省電力オプションは使用できません。

その他のリソース

リモート PC アクセスのその他のリソースは次のとおりです:

- ソリューション設計ガイダンス:「リモート PC アクセス設計の決定」。
- リモート PC アクセスアーキテクチャの例:「Citrix のリモート PC アクセスソリューションのリファレンスアーキテクチャ」。

印刷

November 11, 2021

ここでは、印刷のベストプラクティスについて説明します。

インストール

Linux VDA では、**cups** フィルターと **foomatic** フィルターの両方が必要です。フィルターは VDA とともにインストールされます。フィルターは、ディストリビューションに基づいて手動でインストールすることもできます。例:

RHEL 7 の場合:

```
sudo yum  - y install cups

sudo yum -y install foomatic-filters
<!--NeedCopy-->
```

RHEL 6 の場合:

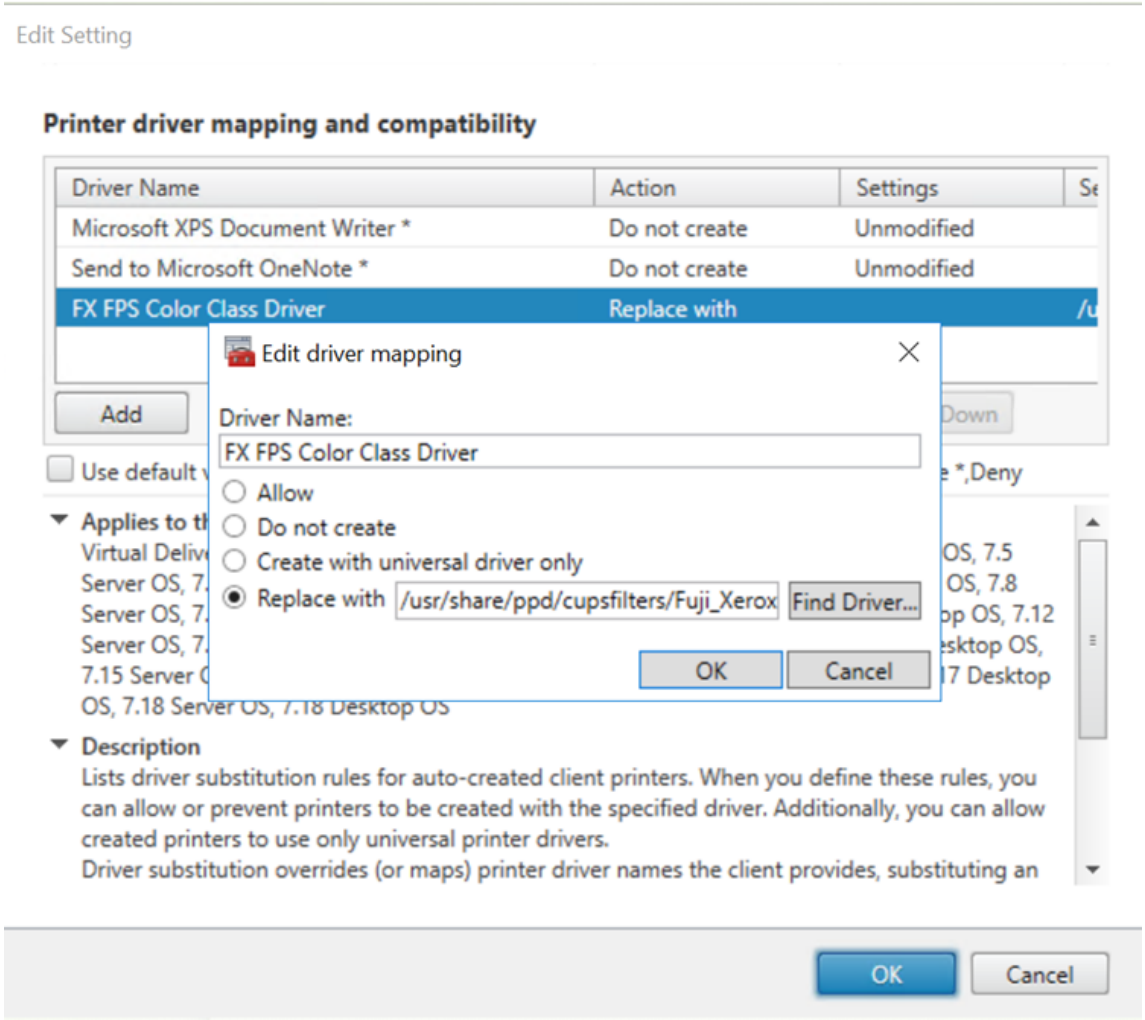
```
sudo yum  - y install cups

sudo yum -y install foomatic
<!--NeedCopy-->
```

構成

Citrix が提供するユニバーサルプリンタードライバーは 3 種類（Postscript、pcl5、pcl6）です。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない可能性があります。この場合、以前のリリースでの唯一のオプションは、~/.CtxlpProfile$CLIENT_NAME 構成ファイルを編集することでした。バージョン 1906 以降では、代わりに Citrix Studio で ［プリンタードライバーのマッピングと互換性］ポリシーを構成するオプションが追加されています。

Citrix Studio で ［プリンタードライバーのマッピングと互換性］ポリシーを構成するには:

1. ［プリンタードライバーのマッピングと互換性］ポリシーを選択します。
2. ［追加］をクリックします。
3. ［ドライバー名］にクライアントプリンターのドライバー名を入力します。Linux 向け Citrix Workspace アプリを使用している場合は、代わりにプリンター名を入力します。
4. ［置換］を選択し、VDA のドライバーファイルへ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。［置換］のみが選択可能になります。

用途

公開デスクトップおよび公開アプリケーションの両方から印刷できます。クライアント側のデフォルトプリンターのみが、Linux VDA セッションに割り当てられます。プリンター名はデスクトップとアプリケーションとで異なりま

す。

- 公開デスクトップの場合
  `CitrixUniversalPrinter:$CLIENT_NAME:dsk$SESSION_ID`
- 公開アプリケーションの場合
  `CitrixUniversalPrinter:$CLIENT_NAME:app$SESSION_ID`

> 注:
>
> 同一ユーザーが公開デスクトップと公開アプリケーションの両方を開いた場合は、どちらのプリンターもセッションで使用できます。公開アプリケーションセッション内でのデスクトッププリンターを使用した印刷、または公開デスクトップでのアプリケーションプリンターを使用した印刷は失敗します。

トラブルシューティング

印刷できない

印刷が正しく機能しない場合、印刷デーモン **ctxlpmngt** と CUPS フレームワークを確認します。

印刷デーモン **ctxlpmngt** はセッションごとのプロセスで、セッション期間を通して実行されている必要があります。次のコマンドを実行して、印刷デーモンが実行中であることを確認します。**ctxlpmngt** が実行中でない場合は、コマンドラインから手動で **ctxlpmngt** を起動します。

```
ps  - ef | grep ctxlpmngt
<!--NeedCopy-->
```

それでも印刷が機能しない場合は、CUPS フレームワークを確認します。**ctxcups** サービスはプリンター管理に使用され、Linux CUPS フレームワークと通信します。これはマシンごとの単一プロセスであり、以下のコマンドを実行して確認できます:

```
service ctxcups status
<!--NeedCopy-->
```

CUPS ログを収集するための追加手順

CUPS ログを収集するには、以下のコマンドを実行して CUPS サービスファイルを構成します。構成しないと、CUPS ログが **hdx.log** で記録されません:

```
sudo service cups stop

sudo vi /etc/systemd/system/printer.target.wants/cups.service

PrivateTmp=false

sudo service cups start
```



```

sudo systemctl daemon-reload
<!--NeedCopy-->
```

> 注:
>
> この構成は、問題が発生した場合に完全な印刷ログを収集することのみを目的としています。この構成により CUPS のセキュリティが破られるため、通常の状況ではこの構成はお勧めしません。

印刷出力が文字化けする

対応していないプリンタードライバーを使用していることが、出力の文字化けの原因になっている可能性があります。ユーザーごとのドライバー構成を使用できるため、**~/.CtxlpProfile$CLIENT_NAME** 構成ファイルを編集して構成できます:

```
[DEFAULT_PRINTER]

printername=

model=

ppdpath=

drivertype=
<!--NeedCopy-->
```

> 重要:
>
> **printername** は、現在のクライアント側の通常使うプリンターの名前が指定されているフィールドです。これは読み取り専用の値です。編集しないでください。
>
> **ppdpath**、**model**、**drivertype** の各フィールドは、マップされたプリンターに対していずれか 1 つのフィールドしか有効にならないため、同時には設定できません。

- ユニバーサルプリンタードライバーがクライアントプリンターに対応していない場合、**model**= オプションを使用してネイティブプリンタードライバーのモデルを構成します。プリンターの現在のモデル名は、**lpinfo** コマンドを使用して表示できます:

  ```
  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

  <!--NeedCopy-->
  ```

  次のようにして、プリンターに一致するようにモデルを設定できます。


```
model=xerox/ph3115.ppd.gz
<!--NeedCopy-->
```

- ユニバーサルプリンタードライバーがクライアントプリンターに対応していない場合、ネイティブプリンタードライバーの PPD ファイルのパスを構成します。**ppdpath** の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  たとえば、**ppd** ドライバーが/home/tester/NATIVE_PRINTER_DRIVER.ppd にある場合は、次のようになります:

  ```
  ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
  <!--NeedCopy-->
  ```

- Citrix が提供するユニバーサルプリンタードライバーは 3 種類（Postscript、pcl5、pcl6）です。プリンターのプロパティに基づいてドライバーの種類を構成できます。

  たとえば、クライアントが通常使うプリンターのドライバーの種類が PCL5 である場合は、**drivertype** を次のように指定します:

  ```
  drivertype=pcl5
  <!--NeedCopy-->
  ```

出力サイズがゼロ

別の種類のプリンターを試します。また、CutePDF や PDFCreator などの仮想プリンターを使用してみて、この問題がプリンタードライバーに関連するものかどうかを確認します。

印刷ジョブは、クライアントが通常使用するプリンターのドライバーによって異なります。現在適用されているドライバーの種類を特定することが重要です。クライアントのプリンターが PCL5 ドライバーを使用している一方で、Linux VDA が PostScript ドライバーを選択していると、問題が発生する場合があります。

プリンタードライバーの種類が正しい場合は、次の手順に従って問題を特定します。

1. 公開デスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME** コマンドを実行します。
3. 次のフィールドを追加して、スプールファイルを Linux VDA に保存します:

   ```
   deletespoolfile=no
   <!--NeedCopy-->
   ```

4. いったんログオフしてからログオンし直して、構成の変更を読み込みます。
5. ドキュメントを印刷して問題を再現します。印刷が完了すると、**/var/spool/cups-ctx/$logon_user/$spool_file** にスプールファイルが保存されます。
6. スプールファイルが空であるかどうかを確認します。スプールファイルのサイズが 0 の場合は、これが問題になります。Citrix サポートに印刷ログを提供して、ガイダンスに従ってください。

7. スプールファイルのサイズが 0 でない場合は、ファイルをクライアントにコピーします。スプールファイルの内容は、クライアントが通常使用するプリンタードライバーの種類によって異なります。マップされたプリンターの（ネイティブ）ドライバーが PostScript である場合、スプールファイルは Linux OS で直接開くことができます。内容が正しいかを確認します。

   スプールファイルが PCL の場合、またはクライアント OS が Windows の場合は、スプールファイルをクライアントにコピーし、別のプリンタードライバーを使用してクライアント側のプリンターで印刷します。

8. マップされたプリンターが別のプリンタードライバーを使用するように変更します。以下では、PostScript クライアントプリンターを例として使用します：

   a) アクティブセッションにログオンして、クライアントデスクトップでブラウザーを開きます。

   b) 印刷管理ポータルを開きます：

   ```
   localhost:631
   <!--NeedCopy-->
   ```

   c) マップされたプリンター ［**CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID**］を選択し、［プリンターの変更］をクリックします。この操作には管理者権限が必要です。

   d) CUPS と CTX 間の接続を保持したまま［続行］をクリックし、プリンタードライバーを変更します。

   e) ［**Make**］フィールドと ［**Model**］フィールドで、Citrix UPD ドライバーではなく別のドライバーを選択します。たとえば、CUPS-PDF 仮想プリンターがインストールされている場合は、［汎用 CUPS-PDF プリンター］ドライバーを選択します。変更を保存します。

   f) このプロセスが正常に完了した場合は、ドライバーの PPD ファイルパスを**.CtxlpProfile$CLIENT_NAME** で設定し、マップされたプリンターが新たに選択したドライバーを使用できるようにします。

既知の問題

Linux VDA での印刷について、次の問題が確認されています。

CTXPS ドライバーが一部の **PLC** プリンターに対応しない

印刷出力が適切でない場合は、プリンタードライバーを、製造元から提供されたネイティブプリンタードライバーに設定してください。

サイズの大きな文書の印刷が遅い

ローカルのクライアントプリンターでサイズの大きなドキュメントを印刷すると、そのドキュメントはサーバーとの接続を介して転送されます。遅い接続では、この転送に時間がかかることがあります。

別のセッションからプリンター通知と印刷ジョブ通知が表示される

Linux でのセッションの考え方は、Windows オペレーティングシステムとは異なります。したがって、すべてのユーザーがシステム全体の通知を受け取ります。次の CUPS 構成ファイルを変更して、これらの通知を無効にできます: **/etc/cups/cupsd.conf**。

次のように、構成されている現在のポリシー名がこのファイルに記述されています。

```
DefaultPolicy default
```

ポリシー名が *default* である場合は、次の行をデフォルトポリシーの XML ブロックに追加します:

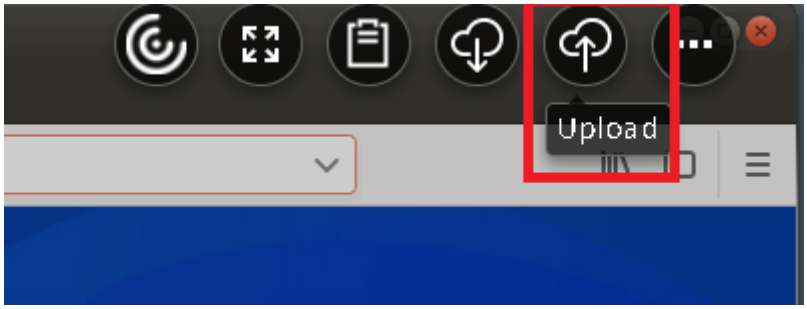
```
<Policy default>

     # 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>
<!--NeedCopy-->
```

ファイル転送

November 11, 2021

Linux VDA とクライアントデバイス間のファイル転送がサポートされています。この機能は、クライアントデバイスが HTML5 の sandbox 属性をサポートする Web ブラウザーを実行している場合に使用できます。HTML5

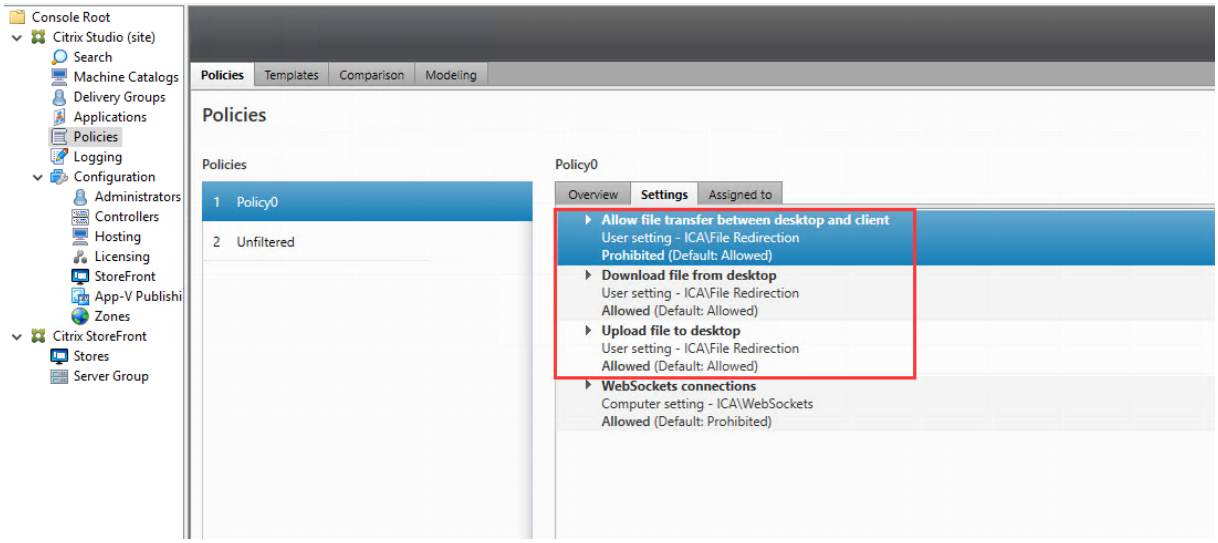
の sandbox 属性は、ユーザーが HTML5 向けまたは Chrome 向け Citrix Workspace アプリを使用して Virtual Apps and Desktops にアクセスできるようにします。公開セッションで、Citrix Workspace アプリのツールバーを使用して、Linux VDA とクライアントデバイス間でファイルのアップロードおよびダウンロードを実行できます。たとえば、ツールバーの ［アップロード］ アイコンをクリックし、クライアントデバイスでファイルを選択して、Linux VDA にファイルをアップロードできます。



> 注:
>
> この機能は、RedHat7.7、CentOS7.6、SUSE12.3、Ubuntu16.04、Ubuntu18.04 で利用できます。
> この機能を使用するには、Citrix Workspace アプリのツールバーを有効にしてください。

ファイル転送のポリシー

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。



ポリシーの説明:

- デスクトップとクライアント間のファイル転送を許可する。Citrix Virtual Apps and Desktops セッションとユーザーデバイス間でのユーザーによるファイル転送を許可または拒否します。
- デスクトップからのファイルのダウンロード。Citrix Virtual Apps and Desktops セッションからユーザーデバイスへのユーザーによるファイルのダウンロードを許可または拒否します。

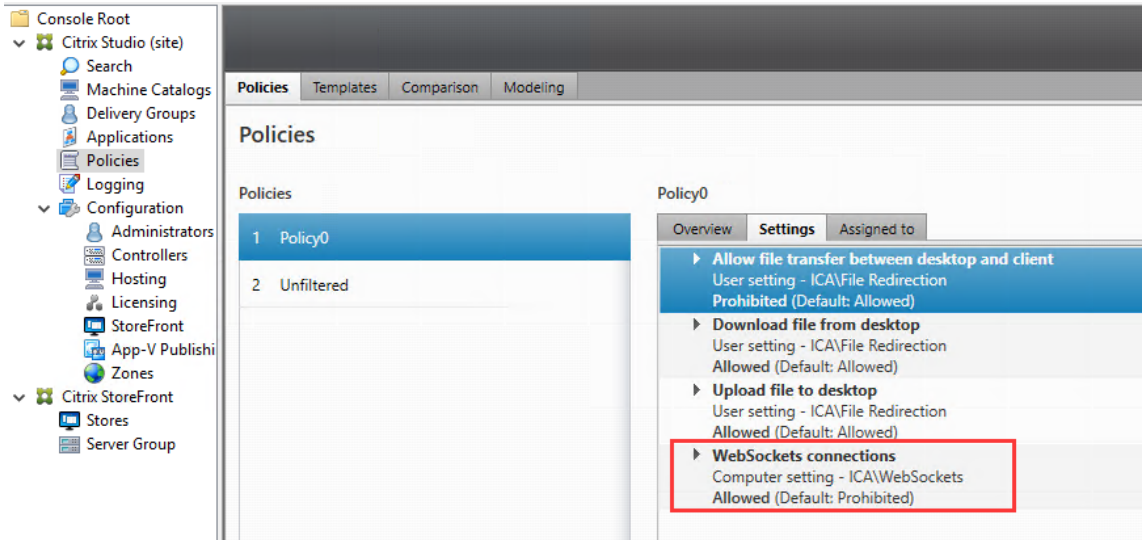
- デスクトップへのファイルのアップロード。ユーザーデバイスから Citrix Virtual Apps and Desktops セッションへのユーザーによるファイルのアップロードを許可または拒否します。

> 注:
>
> デスクトップからファイルをダウンロードおよびデスクトップにファイルをアップロード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。

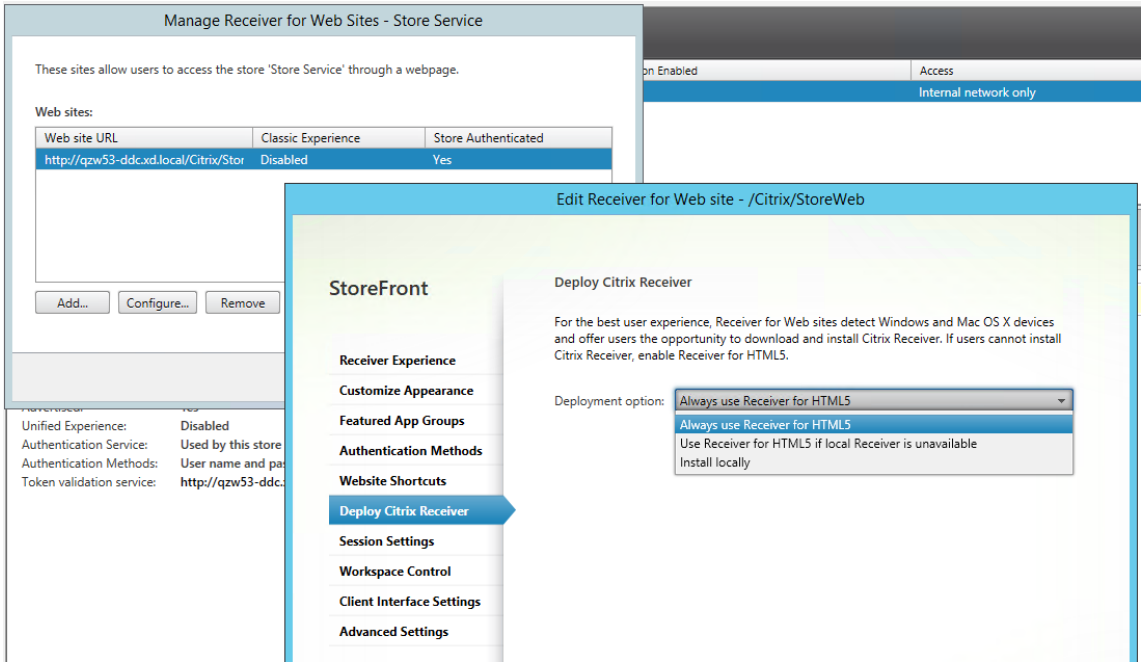
用途

HTML5 向け Citrix Workspace アプリでファイル転送機能を使用するには:

1. Citrix Studio で、**WebSockets** 接続ポリシーを ［許可］ に設定します。



2. Citrix Studio で前述のファイル転送ポリシーからファイル転送を有効にします。
3. Citrix StoreFront 管理コンソールで ［ストア］ をクリックし、［**Receiver for Web** サイトの管理］ ノード、［常に **Receiver for HTML5** を使用］ オプションを選択して、Citrix Receiver for HTML5 を有効にします。



4. 仮想デスクトップまたは Web ブラウザーアプリのセッションを開始します。Linux VDA とクライアントデバイス間でファイルをアップロードおよびダウンロードします。

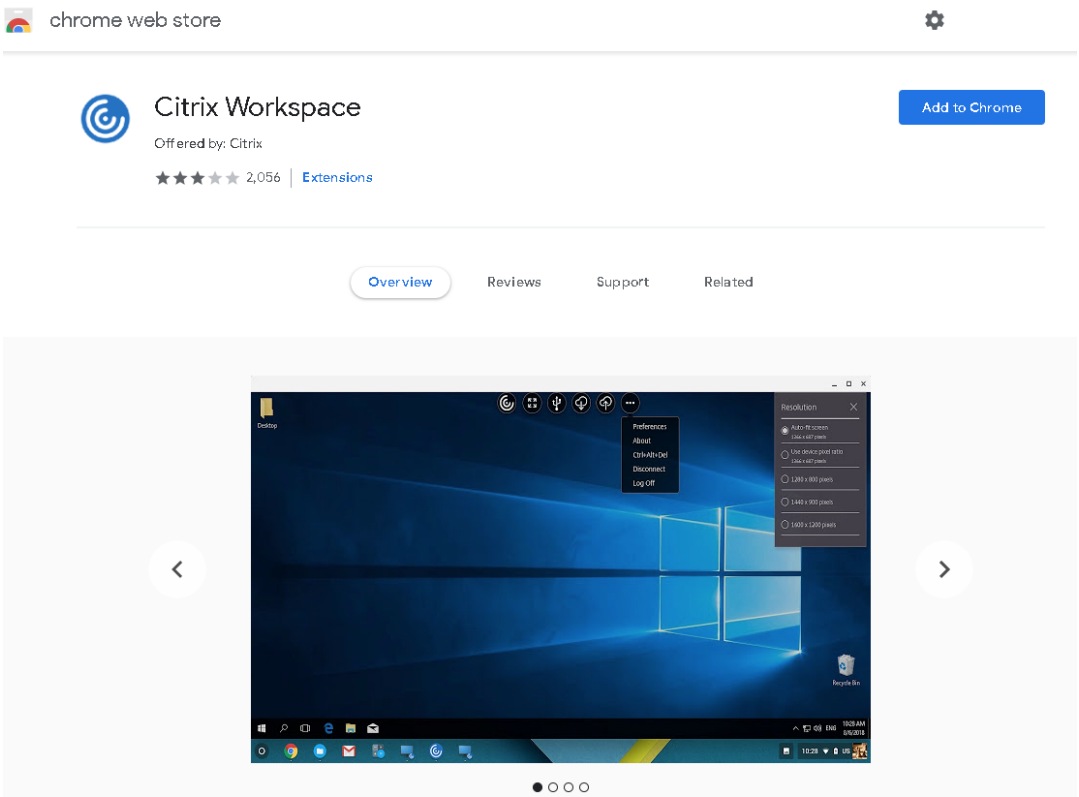
Chrome 向け Citrix Workspace アプリでファイル転送機能を使用するには:

1. 前述のファイル転送ポリシーからファイル転送を有効にします。
2. Chrome ウェブストアから Citrix Workspace アプリを入手します。

   Chrome アプリページから Chrome 向け Citrix Workspace アプリを追加済みの場合は、この手順を省略します。

   a) Google Chrome の検索ボックスに「**Citrix Workspace for Chrome**」と入力します。検索アイコンをクリックします。

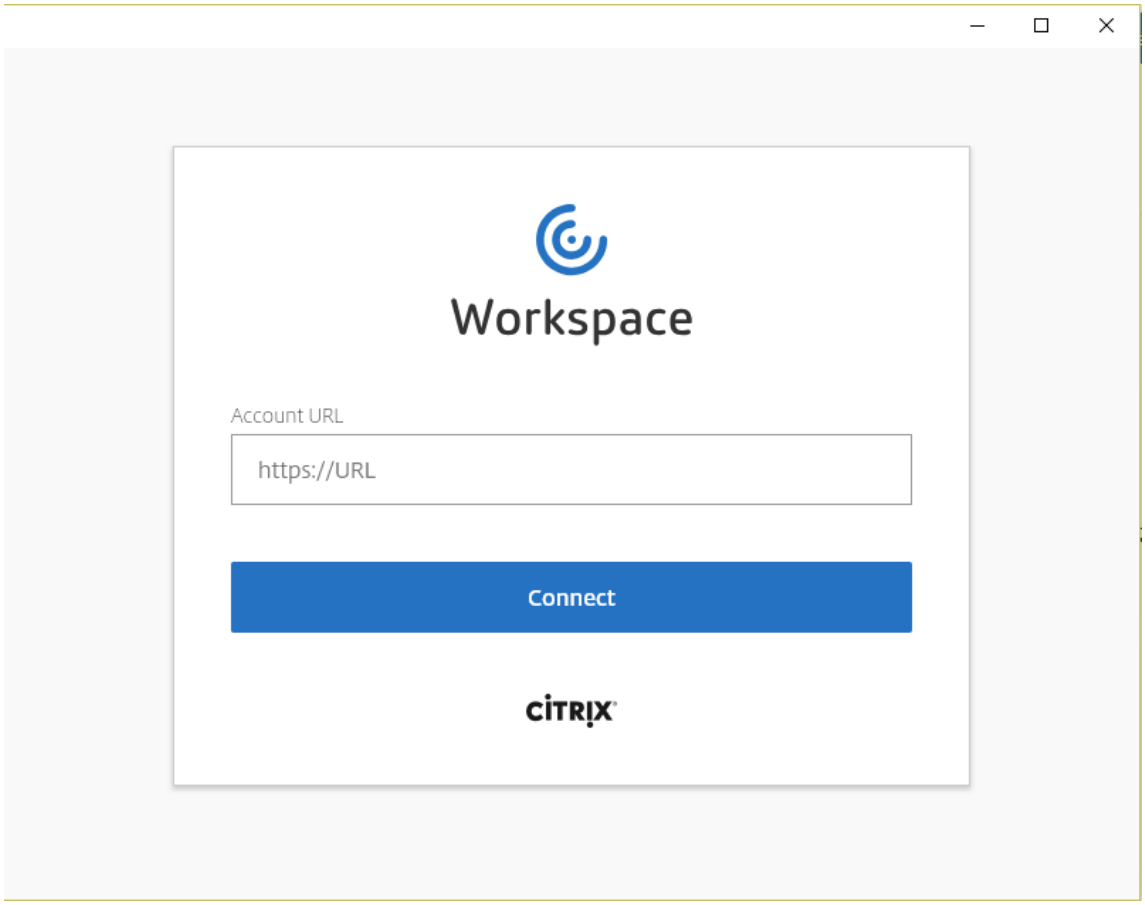
   b) 検索結果で Chrome ウェブストアへの URL をクリックすると、Citrix Workspace アプリを入手できます。



c) ［**Chrome** に追加］を選択して、Citrix Workspace アプリを Google Chrome に追加します。

3. Chrome アプリページで Chrome 向け Citrix Workspace アプリをクリックします。
4. StoreFront ストアの URL を入力して接続します。

   既に入力済みの場合はこの手順を省略します。



5. 仮想デスクトップまたは Web ブラウザーアプリのセッションを開始します。Linux VDA とクライアントデバイス間でファイルをアップロードおよびダウンロードします。

PDF 印刷

August 10, 2021

PDF 印刷に対応したバージョンの Citrix Workspace アプリを使用すると、Linux VDA セッションから変換された PDF を印刷できます。セッション印刷ジョブは、Citrix Workspace アプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、選択した PDF ビューアーを使用して PDF を開き、選択したプリンターで印刷することができます。

Linux VDA は以下のバージョンの Citrix Workspace アプリで PDF 印刷をサポートします:

- Citrix Receiver for HTML5 バージョン 2.4～2.6.9、HTML5 向け Citrix Workspace アプリ 1808 以降
- Citrix Receiver for Chrome バージョン 2.4～2.6.9、Chrome 向け Citrix Workspace アプリ 1808 以降
- Windows 向け Citrix Workspace アプリ 1905 以降

構成

PDF 印刷機能に対応した Citrix Workspace アプリを使用し、Citrix Studio で以下のポリシーを有効にします:

- クライアントプリンターのリダイレクト（デフォルトで有効）
- **PDF** ユニバーサルプリンターを自動作成する（デフォルトで無効）

これらのポリシーが有効になっている場合、起動されたセッションで ［印刷］ をクリックすると、ローカルマシンの印刷プレビューに表示され、プリンターを選択できます。デフォルトプリンターの設定について詳しくは、Citrix Workspace アプリのドキュメントを参照してください。

グラフィックの構成

November 27, 2023

ここでは、Linux VDA のグラフィックの構成と微調整について説明します。

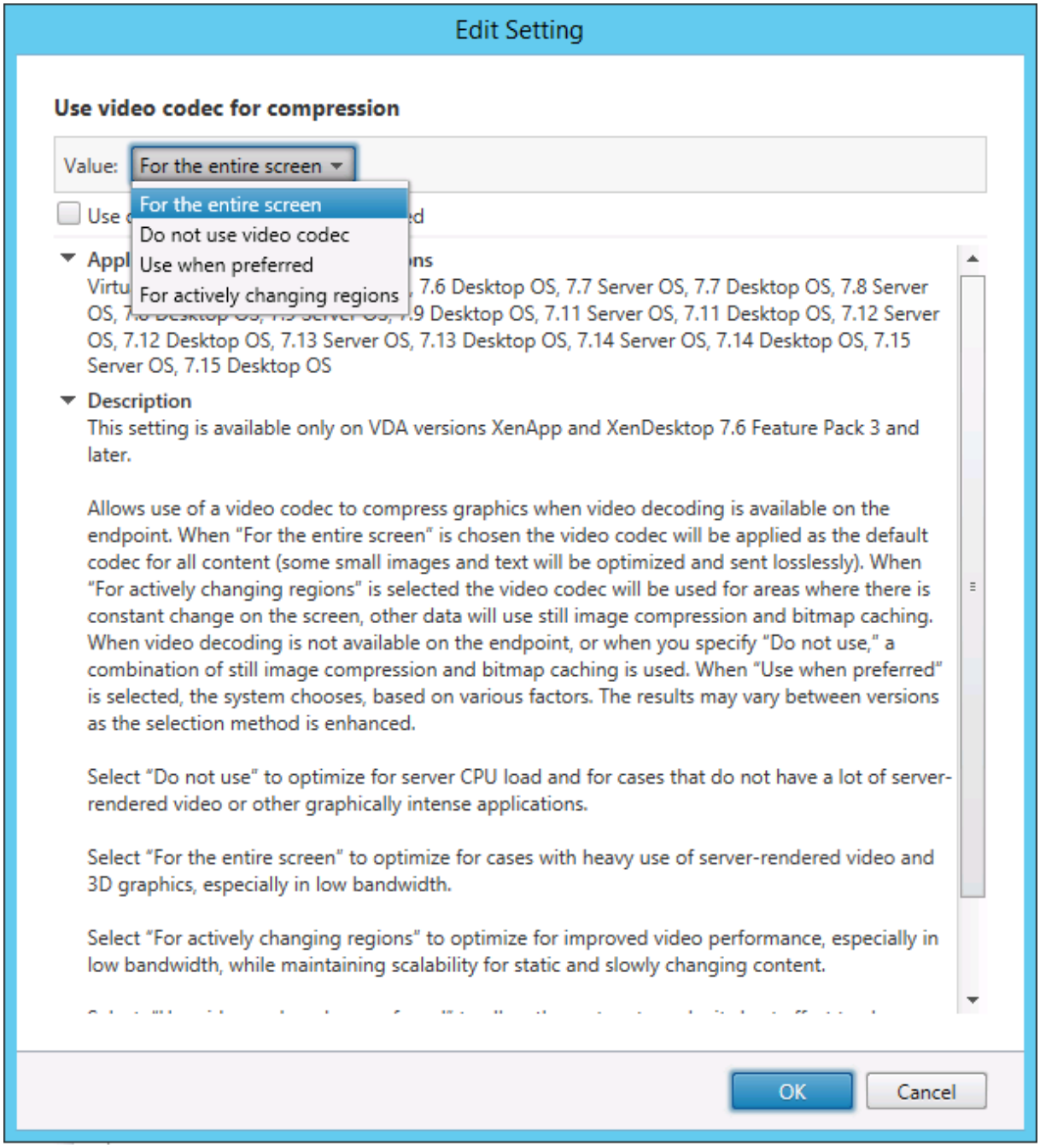
詳しくは、「システム要件」および「インストールの概要」を参照してください。

構成

Thinwire は、Linux VDA で使用されているディスプレイリモートテクノロジです。このテクノロジを使用すると、あるマシンで生成されたグラフィックが、通常はネットワークを経由して、別のマシンに転送され、表示されます。

［圧縮にビデオコーデックを使用する］ グラフィックポリシーでは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して次のオプションを提供します:

- ［可能であれば使用］。この設定がデフォルトです。追加の構成は必要ありません。この設定を保持することにより、すべての Citrix 接続で Thinwire が選択され、デスクトップの一般的なワークロードで、スケーラビリティ、帯域幅、および優れた画質の点で、確実に最適化されます。
- ［画面全体に使用］。特に 3D グラフィックを多用する事例で、Thinwire を全画面 H.264 または H.265 を使用して配信して、ユーザーエクスペリエンスと帯域幅の改善を最適化します。
- ［領域をアクティブに変更］。Thinwire のアダプティブ表示テクノロジは、動画（ビデオ、3D インモーション）を識別し、画像が動く画面の部分でのみ H.264 を使用します。グラフィックの圧縮に **H.264** ビデオコーデックを選択的に使用 することにより、HDX Thinwire はビデオコンテンツなどの 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部分を検出してエンコードすることができます。静止画圧縮（JPEG、RLE）とビットマップキャッシングは、テキストや写真画像などを含む画面の残りの部分で引き続き使用されます。ユーザーは、低帯域幅でありながら、無損失テキストや高品質画像を組み合わせた品質の高いビデオコンテンツを視聴できます。この機能を有効にするには、ポリシー設定の ［圧縮にビデオコーデックを使用する］ を、［可能であれば使用］（デフォルト）または ［アクティブに変化する領域］ に設定します。詳しくは、「グラフィックのポリシー設定」を参照してください。



次の視覚表示ポリシー設定など、いくつかの他のポリシー設定は、ディスプレイリモートのパフォーマンスを微調整するために使用できます。

- 単純なグラフィックスの優先色深度
- ターゲットフレーム数
- 表示品質

Thinwire で［操作時は低品質］に **H.264** を使用

デフォルトでは、［表示品質］ポリシー設定の ［操作時は低品質］設定が、動画に対しては JPEG ではなく H.264 になりました。

H.264 エンコーディングでは優れた画質が提供されます。［圧縮にビデオコーデックを使用する］ポリシーにより、優先設定（デフォルトは ［可能であれば使用］）が制御されます。［操作時は低品質］で JPEG が使用されるよう強制するには、［圧縮にビデオコーデックを使用する］ポリシーを ［ビデオコーデックを使用しない］に設定します。クライアントで Selective H.264 がサポートされていない場合、［操作時は低品質］はポリシー設定に関係なく JPEG に戻ります。Citrix Receiver for Windows 4.9～4.12、Citrix Receiver for Linux 13.5～13.10、Windows 向け Citrix Workspace アプリ 1808 以降、Linux 向け Citrix Workspace アプリ 1808 以降で Selective H.264 がサポートされます。［表示品質］および ［圧縮にビデオコーデックを使用する］のポリシー設定について詳しくは、「視覚表示のポリシー設定」と「グラフィックのポリシー設定」を参照してください。

H.265 ビデオコーデックのサポート

7.18 リリースから、Linux VDA は、リモートグラフィックやビデオのハードウェアアクセラレーションで 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、Citrix Receiver for Windows 4.10～4.12 および Windows 向け Citrix Workspace アプリ 1808 以降で使用できます。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する H.265 デコードをサポートしていない場合、グラフィックポリシー設定の H.265 デコードは無視され、セッションは H.264 ビデオコーデックの使用に戻ります。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ドを有効にするには:

1. ［ビデオコーデックにハードウェアエンコーディングを使用します］ポリシーを有効にします。
2. ［**3D** 画像ワークロードの最適化］ポリシーを有効にします。
3. ［圧縮にビデオコーデックを使用する］ポリシーがデフォルトであること、または ［画面全体に使用］に設定されていることを確認します。
4. ［表示品質］ポリシーが ［操作時は低品質］または ［常に無損失］に設定されていないことを確認します。

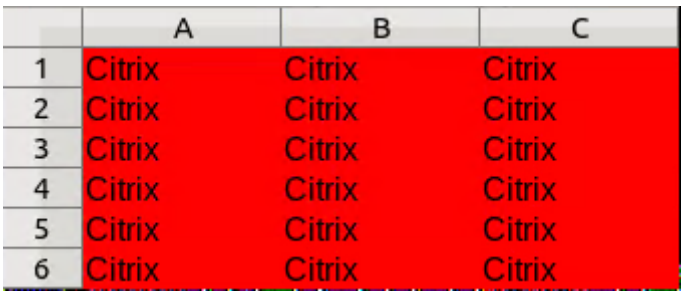
クライアントで H.265 ハードウェアエンコーディングを有効にするには、「H.265 ビデオエンコーディング」を参照してください。

YUV444 ソフトウェアエンコーディングのサポート

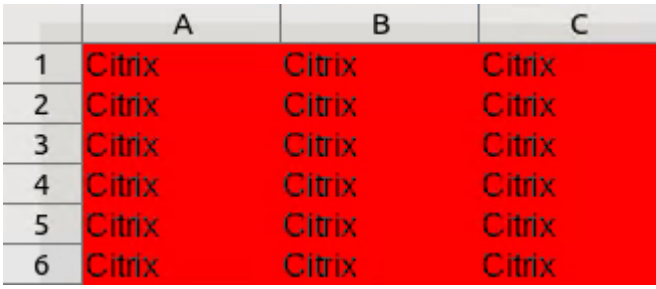
Linux VDA は YUV444 ソフトウェアエンコーディングをサポートします。YUV エンコーディングスキームは、明るさと色の両方の値を各ピクセルに割り当てます。YUV では、「Y」は明るさまたは「輝度」値、「UV」は色または「彩度」値を示します。Linux VDA のこの機能は、Citrix Receiver for Windows 4.10～4.12 および Windows 向け Citrix Workspace アプリ 1808 以降で使用できます。

各固有の Y、U、V 値は 8 ビットまたは 1 バイトのデータで構成されています。YUV444 データ形式は 1 ピクセルあたり 24 ビットを転送します。YUV422 データ形式は 2 ピクセル間で U 値と V 値を共有し、平均転送速度は 16 ビット/ピクセルになります。以下の表は、YUV444 と YUV420 の直観的な比較です。

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する］ポリシーが ［画面全体に使用］に設定されていることを確認します。
2. ［表示品質］ポリシーが ［常に無損失］または ［操作時は低品質］に設定されていないことを確認します。

帯域幅推定に基づいて平均ビットレートを調整する

HDX 3D Pro ハードウェアエンコーディングが Citrix で拡張され、帯域幅推定に基づいて平均ビットレートを調整できます。

HDX 3D Pro ハードウェアエンコーディングを使用中の場合、VDA がネットワーク帯域幅を断続的に推定でき、帯域幅の推定値に基づい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。この新しい機能では、鮮明さと滑らかさのバランスを調整するメカニズムを提供します。

この機能はデフォルトで有効になっています。無効にするには、次のコマンドを実行します:

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    DisableReconfigureEncoder" -d "0x00000001" --force
<!--NeedCopy-->
```

この機能だけでなく、以下のコマンドを実行することでも鮮明さと滑らかさのバランスを調整できます。**AverageBitRatePercent** および **MaxBitRatePercent** パラメーターは、帯域幅使用の割合を設定します。設定した値が大きいほど、グラフィックの鮮明さが向上し滑らかさが低下します。推奨設定範囲は 50～100 です。

```
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
```



```
<!--NeedCopy-->
```

平均ビットレート調整で、画面が静止状態の場合、新しいフレームが送信されることがないため、最新のフレームは低品質状態のままです。鮮明さのサポートでは、最新のフレームを最高品質で再構成し、すぐに送信することでこの問題に対応します。

Linux VDA Thinwire でサポートされているポリシーをすべて示す一覧については、「ポリシーサポート一覧」を参照してください。

Linux VDA でのマルチモニターサポートの構成について詳しくは、CTX220128を参照してください。

トラブルシューティング

使用中のグラフィックモードの確認

次のコマンドを実行して、使用されているグラフィックモードを確認します（**0** は TW+ を、**1** は全画面ビデオコーデックを意味します）：

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
<!--NeedCopy-->
```

次の内容に類似した結果が出力されます：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "GraphicsMode"-d "0x00000000"--force
```

使用中の **H.264** の確認

H.264 が使用中であることを確認するために、次のコマンドを実行します（**0** は使用されていないことを、**1** は使用中であることを意味します）：

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
<!--NeedCopy-->
```

次の内容に類似した結果が出力されます：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H264"-d "0x00000000"--force
```

使用中の **H.265** の確認

全画面 H.265 が使用中であることを確認するために、次のコマンドを実行します（**0** は使用されていないことを、**1** は使用中であることを意味します）：


```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
<!--NeedCopy-->
```

次の内容に類似した結果が出力されます：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H265"-d "0x00000000"--force
```

YUV エンコーディングスキームが使用中であるかどうかの確認

YUV エンコーディングスキームが使用中であることを確認するために、次のコマンドを実行します（**0** は YUV420、**1** は YUV422、**2** は YUV444 を意味します）：

注：ビデオコーデックが使用中の場合のみ、YUVFormat の値に意味があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
<!--NeedCopy-->
```

次の内容に類似した結果が出力されます：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "YUVFormat"-d "0x00000000"--force
```

使用中の **YUV444** ソフトウェアエンコーディングの確認

YUV444 ソフトウェアエンコーディングが使用中であることを確認するために、次のコマンドを実行します：

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
<!--NeedCopy-->
```

YUV444 が使用中の場合、次の内容に類似した結果が出力されます：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "GraphicsMode"-d "0x00000001"--force
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H264"-d "0x00000001"--force
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "HardwareEncoding"-d "0x00000000"--force
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "YUVFormat"-d "0x00000002"--force
```

3D Pro のハードウェアエンコーディングが使用中であるかどうかの確認

次のコマンドを実行します（**0** は使用されていないことを、**1** は使用中であることを意味します）：


```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
<!--NeedCopy-->
```

次の内容に類似した結果が出力されます。

```
create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics"-t "REG_DWORD"
-v "HardwareEncoding"-d "0x00000001"--force
```

これ以外にも **nvidia-smi** コマンドを使用する方法があります。ハードウェアエンコーディングが使用中の場合は、次の内容に類似した結果が出力されます:

```
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 |
+-----------------------------------------------------------------------------+

<!--NeedCopy-->
```

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の確認

NVIDIA GRID グラフィックス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。次の内容に類似した結果が出力されます。


```
+-----------------------------------------------------+
| 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                                                 |
+-----------------------------------------------------------------------------+
<!--NeedCopy-->
```

次のコマンドで、カードに適切な構成を設定します:

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

HDX 3D Pro マルチモニターでの再描画の問題

プライマリモニター以外の画面で再描画の問題が発生している場合は、NVIDIA GRID ライセンスが利用可能であることを確認してください。

Xorg のエラーログを確認する

Xorg のログファイルは、**Xorg.{DISPLAY}.log** に類似した名前で **/var/log/** フォルダー内にあります。

既知の問題と制限事項

vGPU で、**Citrix Hypervisor** のローカルコンソールに **ICA** デスクトップのセッション画面が表示される

回避策：次のコマンドを実行して、仮想マシンのローカル 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
<!--NeedCopy-->
```

8.1 より前の Citrix Hypervisor の場合：

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
<!--NeedCopy-->
```

NVIDIA K2 グラフィックカードは、パススルーモードで **YUV444** ハードウェアエンコーディングをサポートしない

ポリシー設定で ［操作時は低品質］を有効にすると、ユーザーが NVIDIA K2 グラフィックカードを使用してアプリケーションまたはデスクトップのセッションを開始したときに、黒色または灰色の画面が表示されます。この問題は、NVIDIA K2 グラフィックカードがパススルーモードで YUV444 ハードウェアエンコーディングをサポートしないことが原因で発生します。詳しくは、「ビデオエンコードおよびデコードの GPU サポートマトリックス」を参照してください。

Gnome 3 デスクトップのポップアップがログオン時に遅くなる

これは Gnome 3 デスクトップのセッション開始時の機能的制限です。

一部の **OpenGL** および **WebGL** アプリケーションが、**Citrix Workspace** アプリウィンドウのサイズ変更時に適切に表示されない

Citrix Workspace アプリのウィンドウサイズを変更すると、画面の解像度も変更されます。NVIDIA の独自ドライバーにより内部状態が一部変更されるため、それに応じた対応がアプリケーションに求められる場合があります。たとえば、WebGL ライブラリ要素の **lightgl.js** によって「`Rendering to this texture is not supported (incomplete frame buffer)`」というエラーメッセージが生成されることがあります。

Thinwire のプログレッシブ表示

November 11, 2021

低帯域幅または高遅延の接続では、セッションのインタラクティブ性が低下する可能性があります。たとえば、2Mbps 未満の帯域幅の接続または 200 ミリ秒を超える遅延が発生する接続では、Web ページのスクロールが遅くなったり、応答しなかったり、不安定になったりすることがあります。キーボードやマウスの操作がグラフィックの更新に追い付かないことがあります。

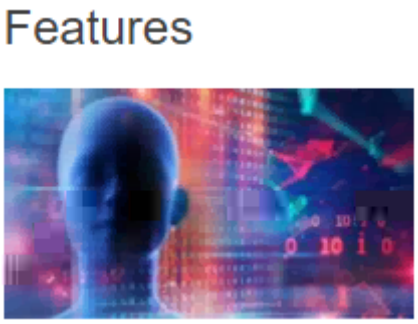
バージョン 7.17 までは、セッションを低表示品質に設定する、または色深度を低く（16 ビットまたは 8 ビットグラフィック）設定することで、ポリシー設定を使用して帯域幅消費を軽減できました。ただし、弱い接続状態であることをユーザーが知っている必要がありました。HDX Thinwire では、ネットワークの状態に基づいて静的な画像の品質を動的に調整することはありませんでした。

バージョン 7.18 以降では、使用可能な帯域幅が 2Mbps を下回る、またはネットワーク遅延が 200 ミリ秒を超えると、HDX Thinwire は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。このモードでは：

- すべての静止画像は大幅に圧縮されます。
- テキストの品質が低下します。

たとえば、プログレッシブ更新モードが有効な次のグラフィックでは、文字 **F** と **e** に青いアーティファクトがあり、イメージは大きく圧縮されています。このアプローチにより、帯域幅消費が大幅に軽減され、画像とテキストをより迅速に受信でき、セッションのインタラクティブ性が向上します。

セッションとの通信が停止すると、劣化した画像やテキストが徐々にシャープになり、劣化がなくなります。たとえば、次のグラフィックでは、文字に青のアーティファクトがなくなっており、画像が元の品質で表示されています。



画像の場合、ランダムにブロック単位でシャープ化します。テキストの場合、個々の文字や単語の一部がシャープ化します。シャープ化のプロセスは数フレームにわたって行われます。この方法により、単一の大きなシャープ化フレームによる遅延を回避します。

遷移画像（ビデオ）は、アダプティブ表示または Selective H.264 で管理されたままです。

プログレッシブモードの動作

デフォルトでは、［表示品質］ポリシー設定が ［高］、［中］（デフォルト）、または ［低］の場合、プログレッシブモードはスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフ（使用されない）になります。

- ［表示品質］が ［常に無損失］ または ［操作時は低品質］である
- ［単純なグラフィックスの優先色深度］が［8 ビット］である
- ［圧縮にビデオコーデックを使用する］が ［画面全体に使用］（全画面の H.264 が望ましい場合）である

プログレッシブモードがスタンバイ状態である場合、デフォルトでは次のいずれかの状況によって有効になります。

- 使用可能な帯域幅が 2 Mbps を下回っている
- ネットワーク遅延が 200 ミリ秒を上回っている

モードの切り替えが発生した後は、悪いネットワーク状況が瞬間的であっても、そのモードが最低 10 秒間継続されます。

プログレッシブモードの動作の変更

プログレッシブモードの動作を変更するには、次のコマンドを実行します：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplay" -d "<value>" --force
2 <!--NeedCopy-->
```

value には次の値を入力します：

0 = 常時オフ（いかなる場合でも使用しないでください）

1 = 自動（ネットワーク状態、デフォルト値に基づいてオンとオフを切り替える）

2 = 常時オン

自動モード（1）の場合、次のコマンドのいずれかを実行して、プログレッシブモードが切り替わるしきい値を変更できます。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg  create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
2 <!--NeedCopy-->
```

value には Kbps 単位のしきい値（デフォルト = 2,048）を入力します

例：帯域幅が 4Mbps を下回ると、プログレッシブモードがオンに切り替わります


```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k "HKEY_LOCAL_MACHINE\SOFTWARE
    \CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplayLatencyThreshold" -d "<value>" --force
<!--NeedCopy-->
```

value にはミリ秒単位のしきい値（デフォルト = 200）を入力します

例：ネットワーク遅延が 100 ミリ秒を下回ると、プログレッシブモードがオンに切り替わります。

GRID 以外の 3D グラフィック

March 11, 2024

概要

Linux VDA で NVIDIA GRID 3D カードだけでなく、GRID 以外の 3D カードもサポートするように機能が拡張されました。

インストール

GRID 以外の 3D グラフィックスの機能を使用するには、以下を実行する必要があります：

- 前提条件として XDamage をインストールします。通常、XDamage は XServer の拡張機能として存在しています。
- Linux VDA をインストールする場合は、`CTX_XDL_HDX_3D_PRO`を`Y`に設定します。環境変数については、「手順 7：Runtime Environment をセットアップしてインストールを完了する」を参照してください。

構成

Xorg の構成ファイル

ご使用の 3D カードドライバーが NVIDIA の場合、構成ファイルは自動でインストールおよび設定されます。

他の種類の **3D** カード

ご使用の 3D カードドライバーが NVIDIA 以外の場合は、`/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 を例として使用しながら、以下の手順に従ってテンプレート構成ファイルを変更します:

1. ***driver_name*** は、実際のドライバー名で置き換えてください。

   たとえば、ドライバー名が`intel`の場合は、構成ファイル名を`ctx-intel-1`に変更できます。

2. ビデオドライバー情報を追加します。

   各テンプレート構成ファイルには、「Device」という名前のセクションがあり、コメントアウトされています。このセクションでは、ビデオドライバー情報を記述します。ビデオドライバー情報を追加する前に、このセクションを有効にします。このセクションを有効にするには:

   a) 製造元から提供されている 3D カードガイドを参照して構成情報を確認します。ネイティブ構成ファイルを生成できます。Linux VDA ICA セッションを使用していないときに、ネイティブ構成ファイルを使用して、ローカル環境で 3D カードが動作可能であることを確認します。

   b) ネイティブ構成ファイルの［Device］セクションを **ctx-driver_name-1.conf** にコピーします。

3. 次のコマンドを実行して、手順 1 で設定した構成ファイル名を Linux VDA が認識できるようにレジストリキーを設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
    CurrentControlSet\Control\Citrix\XDamage" -t "REG_SZ" -v "
    DriverName" -d "intel" --force
<!--NeedCopy-->
```

GRID 以外の **3D** グラフィック機能を有効にする

GRID 以外の 3D グラフィック機能はデフォルトで無効です。次のコマンドを実行して XDamageEnabled を 1 に設定することで有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
    CurrentControlSet\Control\Citrix\XDamage" -t "REG_DWORD" -v "
    XDamageEnabled" -d "0x00000001" --force
<!--NeedCopy-->
```

トラブルシューティング

グラフィック出力がないか文字化けする

ローカルで 3D アプリケーションを実行でき、すべてを適切に構成しているのにグラフィック出力がないまたは不明瞭であるとすると、原因はバグです。/opt/Citrix/VDA/bin/setlog を使用して GFX_X11 を verbose に設定する

ことでデバッグ用にトレース情報を収集します。

ハードウェアエンコーディングが機能しない

この機能ではソフトウェアエンコーディングのみをサポートしています。

ポリシーの設定

March 11, 2022

インストール

Linux VDA の準備についてはインストールのトピックを参照してください。

依存関係

Linux VDA パッケージのインストール前に、次の依存関係をインストールします。

RHEL/CentOS:

```
sudo yum -y install openldap

sudo yum -y install libxml2

sudo yum -y install cyrus-sasl

sudo yum -y install cyrus-sasl-gssapi
<!--NeedCopy-->
```

SLES/SELD:

```
sudo zypper install openldap2

sudo zypper install libxml2

sudo zypper install cyrus-sasl

sudo zypper install cyrus-sasl-gssapi
<!--NeedCopy-->
```

Ubuntu:

```
sudo apt-get install -y libldap-2.4-2

```



```
sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit
<!--NeedCopy-->
```

構成

Citrix Studio のポリシー設定

Citrix Studio のポリシー設定は、次の操作を行います。

1. **Citrix Studio** を開きます。
2. ［ポリシー］パネルを選択します。
3. ［ポリシーの作成］をクリックします。
4. 「ポリシーサポート一覧」に沿ってポリシーを設定します。

VDA での **LDAP** サーバーの設定

Linux VDA での LDAP サーバーの設定は、単一ドメインの環境では必須ではありませんが、複数ドメインおよび複数フォレストの環境では必須です。これらの環境で LDAP 検索を実行するには、ポリシーサービスに LDAP サーバーの設定が必要です。

Linux VDA パッケージのインストール後に、次のコマンドを実行します。

```
/opt/Citrix/VDA/sbin/ctxsetup.sh
<!--NeedCopy-->
```

すべての LDAP サーバーを、推奨される形式である LDAP の完全修飾ドメイン名（FQDN）および LDAP ポートのスペース区切りの一覧（例：ad1.mycompany.com:389 ad2.mycomany.com:389）で入力します。

```
Checking CTX_XDL_LDAP_LIST... value not set.
The Virtual Delivery Agent by default queries DNS to discover LDAP servers, however if DNS is unable to provide
LDAP service records, you may provide a space-separated list of LDAP Fully Qualified Domain Names (FQDNs) with
LDAP port (e.g. ad1.mycompany.com:389).
If required, please provide the FQDN:port of at least one LDAP server. [<none>]:
```

また、**ctxreg** コマンドを実行して、この設定をレジストリに直接書き込むこともできます：

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
    VirtualDesktopAgent" -t "REG_SZ" -v "ListOfLDAPServers" -d "ad1.
    mycompany.com:389 ad2.mycomany.com:389" --force
<!--NeedCopy-->
```


ポリシーサポート一覧

November 11, 2021

Linux VDA ポリシーサポート一覧

Studio ポリシー	キー名	種類	モジュール	デフォルト値
クライアントのローカルタイムゾーンを使用する	UseLocalTimeOfClient	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 往復測定	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザーモニタリング	有効 (1)
ICA 往復測定間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザーモニタリング	15
アイドル接続の ICA 往復測定	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザーモニタリング	無効 (0)
セッション全体の最大帯域幅	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの最大帯域幅（Kbps）	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの最大帯域幅（%）	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
USB デバイスリダイレクトの最大帯域幅	LimitUSBBw	ユーザー	ICA\帯域幅	0
USB デバイスリダイレクトの帯域幅（%）	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの最大帯域幅（Kbps）	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの最大帯域幅（%）	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0

Studio ポリシー	キー名	種類	モジュール	デフォルト値
ファイルリダイレクトの最大帯域幅（Kbps）	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクトの最大帯域幅（%）	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの最大帯域幅（Kbps）	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの最大帯域幅（%）	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0
WebSocket 接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocket ポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008
WebSocket 信頼される接続元サーバー一覧	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICA Keep-Alive	SendICAKeepAlives	コンピューター	ICA Keep-Alive	ICA Keep-Alive メッセージ (0) を送信しない
ICA Keep-Alive タイムアウト	ICAKeepAliveTimeout	コンピューター	ICA Keep-Alive	60 秒
ICA リスナーポートの番号	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）
セッション画面の保持	AcceptSessionReliabilityConnections	コンピューター	ICA\セッション画面の保持	許可 (1)
再接続 UI の透過レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\クライアントの自動接続	80%
セッション画面の保持のポート番号	SessionReliabilityPort	コンピューター	ICA\セッション画面の保持	2598
セッション画面の保持のタイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッション画面の保持	180 秒
クライアントの自動再接続	AllowAutoClientReconnect	コンピューター	ICA\クライアントの自動接続	許可 (1)
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)

Studio ポリシー	キー名	種類	モジュール	デフォルト値
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDF ユニバーサルプリンターを自動作成する	AutoCreatePDFPrinter	ユーザー	印刷	無効 (0)
プリンタードライバーのマッピングと互換性	DriverMappingList	ユーザー	印刷	`"Microsoft XPS Document Writer *, Deny;Send to Microsoft OneNote *, Deny"`
クライアントクリップボードリダイレクト	AllowClipboardRedir	ユーザー	クリップボード	許可 (1)
クライアント USB デバイスリダイレクト	AllowUSBRedir	ユーザー	USB	禁止 (0)
クライアント USB デバイスリダイレクト規則	USBDeviceRules	ユーザー	USB	“\0”
動画圧縮	MovingImageCompressionConfiguration	ユーザー	Thinwire	有効 (1)
エクストラ色圧縮	ExtraColorCompression	ユーザー	Thinwire	無効 (0)
保持する最低フレーム数	TargetedMinimumFramesPerSecond	ユーザー	Thinwire	10fps
ターゲットフレーム数	FramesPerSecond	ユーザー	Thinwire	30fps
表示品質	VisualQuality	ユーザー	Thinwire	中 (3)
圧縮にビデオコーデックを使用する	VideoCodec	ユーザー	Thinwire	選択された場合使用する (3)
ビデオ コーデックにハードウェアエンコーディングを使用します	UseHardwareEncodingForVideoCodec	ユーザー	Thinwire	有効 (1)
視覚的無損失の圧縮を使用する	AllowVisuallyLosslessCompression	ユーザー	Thinwire	無効 (0)

Studio ポリシー	キー名	種類	モジュール	デフォルト値
3D 画像ワークロードの最適化	OptimizeFor3dWorkload	ユーザー	Thinwire	無効 (0)
単純なグラフィックスの優先色深度	PreferredColorDepth	ユーザー	Thinwire	24 ビット/ピクセル (1)
音質	SoundQuality	ユーザー	オーディオ	高 - 高品位オーディオ (2)
クライアントマイクリダイレクト	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
同時ログオントレランス	ConcurrentLogonsTolerance	コンピューター	負荷管理	2
Controller の自動更新を有効にする	EnableAutoUpdateOfControllers	コントローラー	Virtual Delivery Agent 設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3
Speex 最大品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブに自動接続する	AutoConnectDrives	ユーザー	ファイルリダイレクト/CDM	有効 (1)
クライアント側光学式ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント側固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント側フロッピードライブ	AllowFloppyDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント側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)
クライアント側ドライブへの読み取り専用アクセス	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)

Studio ポリシー	キー名	種類	モジュール	デフォルト値
デスクトップとクライアント間のファイル転送を許可する	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロード	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロード	AllowFileUpload	ユーザー	ファイル転送	許可

次のポリシーは、Citrix Studio バージョン 7.12 以降で構成できます。

- MaxSpeexQuality

  値（整数）：[0〜10]

  デフォルト値：5

  詳細：

  オーディオリダイレクトで、音質が中または低の場合、オーディオデータを Speex でエンコードします（音質のポリシーを参照）。Speex は劣化を伴うコーデックであり、入力音声信号の品質を犠牲にして圧縮します。その他の音声コーデックと違い、品質とビットレートのバランスを制御できます。Speex のエンコーディングプロセスは、ほとんどの場合、0 から 10 の範囲の品質パラメーターで制御します。品質が高いほど、ビットレートも高くなります。

  Speex 最大品質は、最高の Speex 品質を選択して音声品質と帯域幅制限に従ってオーディオデータをエンコードします（オーディオリダイレクトおよび帯域幅制限のポリシー参照）。音声品質が中の場合、エンコーダーは広帯域モードの、より高いサンプルレートになります。音声品質が低の場合、エンコーダーは狭帯域モードで、より低いサンプルレートになります。同じ Speex 品質でも、モードとビットレートは異なります。最高の Speex 品質は、以下の条件を満たす最大の値です。

  - 品質が Speex 最大品質以下
  - ビットレートが帯域幅制限以下

  関連設定：音質、オーディオリダイレクトの最大帯域幅

- PrimarySelectionUpdateMode

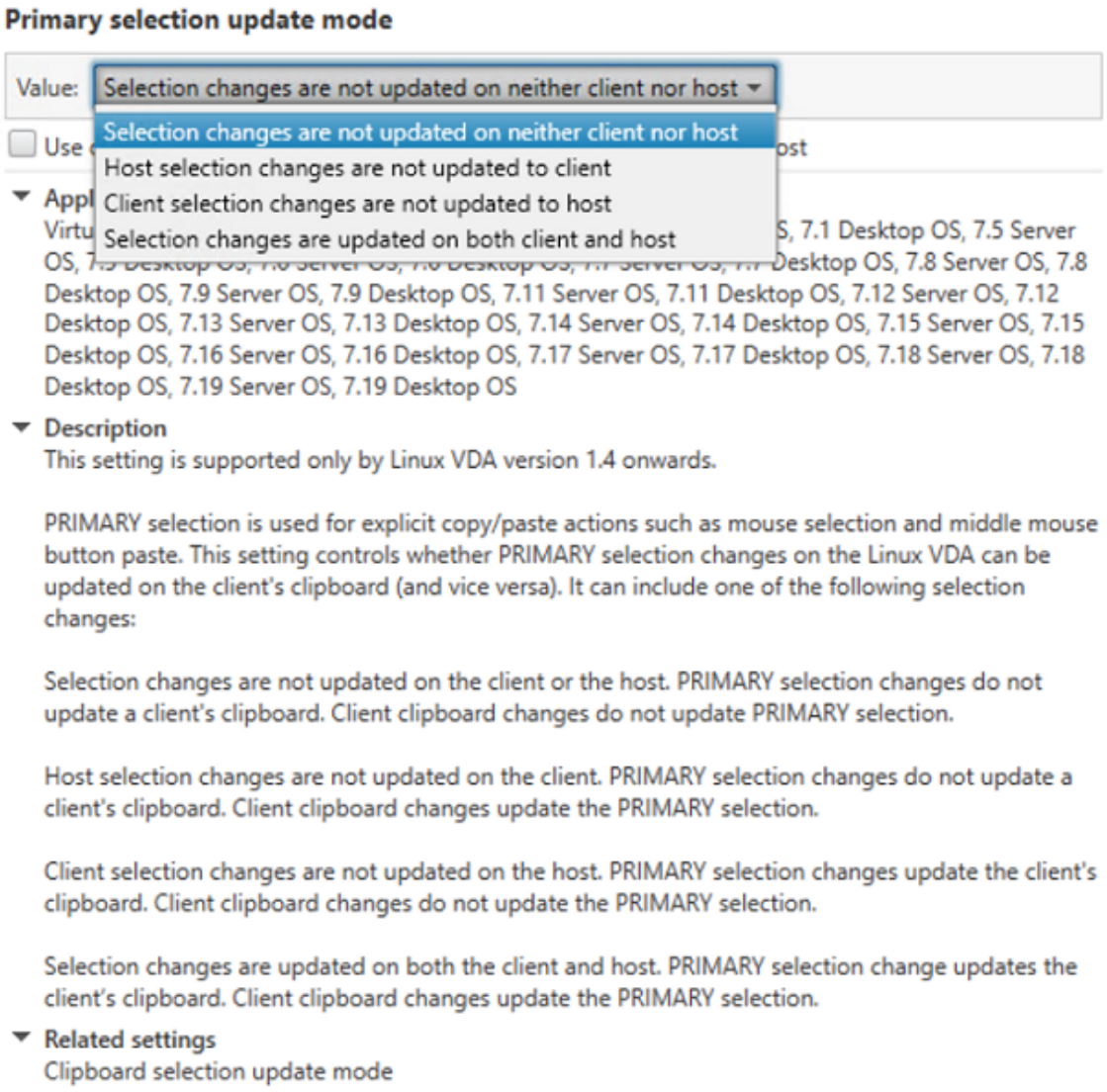
  値（列挙）：[[0, 1, 2, 3]]

  デフォルト値：3

  詳細：

  プライマリ選択は、データを選択し、マウスの中央ボタンを押して貼り付ける場合に使用されます。

この設定は、Linux VDA でのプライマリ選択の変更がクライアントのクリップボードで更新されるかどうかを制御します（逆の場合も同様）。値には、次の 4 つのオプションがあります:



- 選択の変更はクライアントでもホストでも更新されません
  Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されることはありません。
- ホスト選択の変更はクライアントで更新されません
  Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されます。
- クライアント選択の変更はホストで更新されません
  Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されることはありませ

ん。

- 選択の変更は、クライアントとホストの両方で更新されます
  Linux VDA でのプライマリ選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのプライマリ選択の変更により、Linux VDA のクリップボードが更新されます。このオプションがデフォルト値です。

関連設定：クリップボード選択更新モード

- ClipboardSelectionUpdateMode

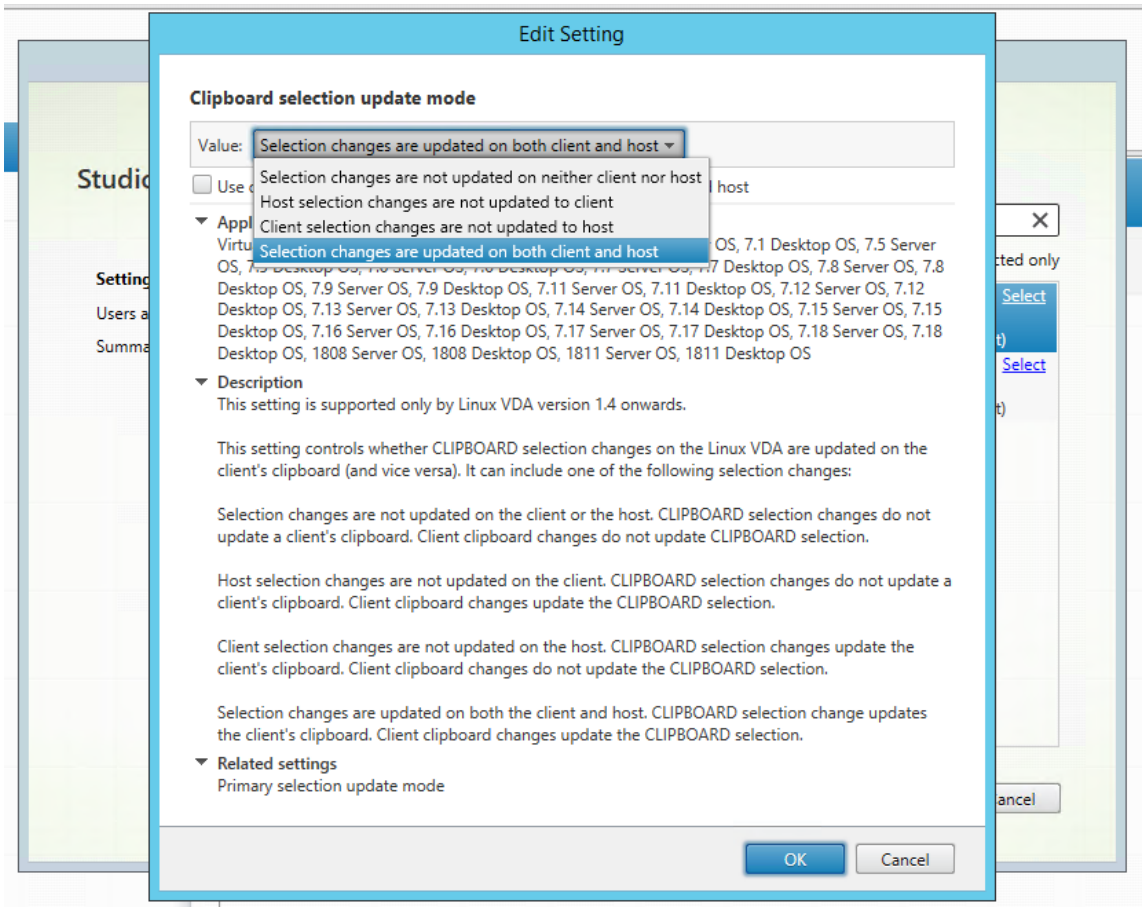
値（列挙）：[[0, 1, 2, 3]]

デフォルト値：3

詳細：

クリップボード選択は、いくつかのデータを選択し、ショートカットメニューの「コピー」を選択するなど、クリップボードに「コピー」することを明示的に要求する場合に使用します。クリップボード選択は、主に Microsoft Windows のクリップボード操作に関連して使用され、プライマリ選択は Linux 特有の操作です。

このポリシーは、Linux VDA でのクリップボード選択の変更がクライアントのクリップボードで更新されるかどうかを制御します（逆の場合も同様）。値には、次の 4 つのオプションがあります：



- 選択の変更はクライアントでもホストでも更新されません
  Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されることはありません。
- ホスト選択の変更はクライアントで更新されません
  Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードは更新されません。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されます。
- クライアント選択の変更は、ホストで更新されません
  Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されることはありません。
- 選択の変更は、クライアントとホストの両方で更新されます
  Linux VDA でのクリップボード選択の変更によって、クライアント上のクリップボードが更新されます。クライアントのクリップボード選択の変更により、Linux VDA のクリップボードが更新されます。このオプションがデフォルト値です。

関連設定：プライマリ選択更新モード

> 注：
>
> Linux VDA では、クリップボード選択とプライマリ選択の両方がサポートされています。Linux VDA とクライアント間のコピーおよび貼り付けの動作を制御するには、Citrix で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。

IPv6 の構成

November 11, 2021

Linux VDA では、Citrix Virtual Apps and Desktops に IPv6 を使用できます。この機能を使用するときは、次の点に注意してください。

- デュアルスタック環境で、IPv6 が明示的に有効になっていない場合、IPv4 が使用されます。
- IPv4 環境で IPv6 を有効にすると、Linux VDA は機能しません。

> 重要：
>
> - Linux VDA だけではなく、ネットワーク環境全体が IPv6 である必要があります。
> - Centrify ではピュア IPv6 をサポートしていません。

Linux VDA をインストールしている場合、IPv6 の特別なセットアップタスクは必要ありません。

Linux VDA で IPv6 を構成する

Linux VDA の構成を変更する前に、以前 IPv6 ネットワークで Linux 仮想マシンが機能していたかを確認する必要があります。IPv6 の構成に関連する 2 つのレジストリキーがあります。

```
1  “HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD”
     -v “OnlyUseIPv6ControllerRegistration”
2  “HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD”
     -v “ControllerRegistrationIPv6Netmask”
3  <!--NeedCopy-->
```

OnlyUseIPv6ControllerRegistration を 1 に設定して、Linux VDA で IPv6 を有効にします:

```
1  sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
     Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
     OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
2  <!--NeedCopy-->
```

Linux VDA に複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask** で Linux VDA の登録に使用するネットワークインターフェイスを指定できます:

```
1  sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
     Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "
     ControllerRegistrationIPv6Netmask " -d "{
2   IPv6 netmask }
3   " --force
4  <!--NeedCopy-->
```

{IPv6 netmask} を実際のネットマスク（2000::/64 など）に置き換えます。

Citrix Virtual Apps and Desktops での IPv6 展開について詳しくは、「IPv4/IPv6 support」を参照してください。

トラブルシューティング

基本の IPv6 ネットワーク環境をチェックしてから、ping6 を使用して AD および Delivery Controller に接続できるかを確認します。

Citrix カスタマーエクスペリエンス向上プログラム（CEIP）の構成

February 11, 2021

CEIP に参加すると、匿名の統計および使用状況情報が、Citrix 製品の品質およびパフォーマンスを向上させる目的で送信されます。この匿名データのコピーは、より迅速かつ効率的に分析するために Google Analytics（GA）にも送信されます。

レジストリ設定

デフォルトでは、ユーザーは Linux VDA のインストール時に CEIP に自動で参加します。Linux VDA のインストールからおよそ 7 日後に、初回データアップロードが行われます。このデフォルト設定はレジストリで変更できます。

- **CEIPSwitch**

CEIP を有効または無効にするレジストリ設定（デフォルトは 0）:

場所：HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

名前：CEIPSwitch

値のデータ：1 = 無効、0 = 有効

未指定の場合、CEIP は有効です。

クライアント上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"
<!--NeedCopy-->
```

- **GASwitch**

GA を有効または無効にするレジストリ設定（デフォルトは 0）:

場所：HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

名前：GASwitch

値のデータ：1 = 無効、0 = 有効

未指定の場合、GA は有効です。

クライアント上で次のコマンドを実行して GA を無効にできます:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    Citrix\CEIP" -v "GASwitch" -d "1"
<!--NeedCopy-->
```

- **DataPersistPath**

データ永続パス（デフォルトは/var/xdl/ceip）を制御するレジストリ設定:

場所：HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

名前：DataPersistPath

値のデータ：文字列

次のコマンドを実行してこのパスを設定できます。


```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    Citrix\CEIP" -v "DataPersistPath" -d "your_path"
<!--NeedCopy-->
```

構成し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

次の表では、収集される匿名の情報の種類の例を紹介します。データでは、お客様を特定するすべての詳細は含まれません。

データポイント	キー名	説明
マシンのグローバル一意識別子	machine_guid	データの発生元のマシンを識別
AD ソリューション	ad_solution	マシンのドメイン参加方式を示すテキスト文字列
Linux カーネルのバージョン	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 の更新または新規のインストール	update_or_fresh_install	現在の Linux VDA パッケージが新規インストールであるのか更新であるの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、簡単インストール、または手動インストールのいずれかでインストールされたかを示すテキスト文字列
HDX 3D Pro が有効かどうか	hdx_3d_pro	マシンで HDX 3D Pro が有効かどうかを示すテキスト文字列
VDI モードが有効化かどうか	vdi_mode	VDI モードが有効かどうかを示すテキスト文字列
システムのロケール	system_locale	このマシンのロケールを示すテキスト文字列
LVDA キーサービスの前回再起動時間	ctxhdx ctxvda	dd-hh:mm:ss 形式（例: 10-17:22:19）による`ctxhdx`および`ctxvda`サービスの前回再起動時間
GPU の種類	gpu_type	マシンの GPU の種類
CPU コア	cpu_cores	マシンの CPU コア数を示す整数

データポイント	キー名	説明
CPU 周波数	cpu_frequency	CPU の周波数（MHz）を示す浮動小数点数
物理メモリサイズ	memory_size	物理メモリのサイズ（KB）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にあった起動された（ログオン済みまたは接続済み）セッションの数を示す整数
Linux OS の名前およびバージョン	os_name_version	マシンの Linux OS の名前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	データの発生元のセッションを識別
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を示すテキスト文字列：デスクトップまたは<appname>
アクティブセッション時間	active_session_time	セッションのアクティブ時間の保存に使用。セッションは切断や再接続がありえるため、単一のセッションのアクティブ時間が複数になることがあります
セッション継続時間	session_duration_time	ログオンからログオフまでのセッションの継続時間の保存に使用
Receiver クライアントの種類	receiver_type	セッションの起動に使用された Citrix Workspace アプリの種類を示す整数
Receiver クライアントのバージョン	receiver_version	セッションの起動に使用された Citrix Workspace アプリのバージョンを示すテキスト文字列
印刷回数	printing_count	セッションで印刷機能を使用した回数を示す整数
USB リダイレクト回数	usb_redirecting_count	セッションで USB デバイスを使用した回数を示す整数
Gfx プロバイダーの種類	gfx_provider_type	セッションのグラフィックプロバイダーの種類を示すテキスト文字列
シャドウの回数	shadow_count	セッションがシャドウされた回数を示す整数
ユーザーが選択した言語	ctxism_select	ユーザーが選択したすべての言語を含む、合成された長い文字列

データポイント	キー名	説明
スマートカードリダイレクトカウント	scard_redirecting_count	スマートカードリダイレクトがセッションログオンおよびセッション中アプリのユーザー認証に使用される回数を示す整数

USB リダイレクトの設定

November 15, 2021

USB デバイスは、Citrix Workspace アプリと Linux VDA デスクトップ間で共有されます。USB デバイスがデスクトップにリダイレクトされると、USB デバイスをローカルに接続されているかのように使用することができます。

USB リダイレクトの主な機能として、次の 3 つが挙げられます。

- オープンソースプロジェクトの導入（VHCI）
- VHCI サービス
- USB サービス

オープンソース **VHCI**:

USB リダイレクトのこの機能により、IP ネットワーク上でシステムを共有する汎用 USB デバイスを開発します。この機能は Linux カーネルドライバーおよびユーザーモードのライブラリで構成されており、ユーザーはカーネルドライバーと通信してすべての USB データを取得することができます。Linux VDA の導入では、VHCI のカーネルドライバーが Citrix で再利用されます。ただし、Linux VDA と Citrix Workspace アプリ間の USB データ転送はすべて Citrix ICA プロトコルパッケージに格納されます。

VHCI サービス:

VHCI サービスは、Citrix が提供する、VHCI カーネルモジュールとの通信のためのオープンソースサービスです。このサービスは VHCI と Citrix USB サービスの間のゲートウェイとして機能します。

USB サービス:

USB サービスは、USB デバイスでの仮想化およびデータ転送をすべて管理する Citrix モジュールです。

USB リダイレクトのしくみ

通常、Linux VDA への USB デバイスのリダイレクトが正常に行われると、デバイスノードがシステムの/dev パスに作成されます。ただし、リダイレクトされたデバイスがアクティブな Linux VDA セッションで使用できない場合があります。USB デバイスが正常に機能するかどうかはドライバーによって決まり、一部のデバイスは特別なドライバ

ーを必要とします。ドライバーが提供されていない場合、リダイレクトされた USB デバイスはアクティブな Linux VDA セッションにアクセスできません。確実に USB デバイスを接続するには、正しくドライバーをインストールしてシステムを設定してください。

Linux VDA は、クライアントとの間でリダイレクトが正常に行われた USB デバイスの一覧をサポートしています。また、デバイス、特に USB ディスクが適切にマウントされるため、ユーザーは追加の設定なしでディスクにアクセスできます。

サポートされている **USB** デバイス

次のデバイスは、Linux VDA のこのバージョンをサポートしていることが確認されています。他のデバイスを使用すると、予期せぬ結果が生じる場合があります。

> 注:
>
> Linux VDA では、USB 2.0 プロトコルのみがサポートされます。

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
Kingston DataTraveler 101 II	0951:1625	FAT32
Kingston DataTraveler GT101 G2	1567:8902	FAT32
SanDisk SDCZ80 Flash Drive	0781:5580	FAT32
WD HDD	1058:10B8	FAT32

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

USB リダイレクトの設定

USB デバイスのリダイレクトの有効化および無効化は、Citrix ポリシーにより制御されます。また、Delivery Controller ポリシーを使用してデバイスの種類を指定することもできます。USB リダイレクトを Linux VDA に設定するには、次のポリシーと規則を設定します。

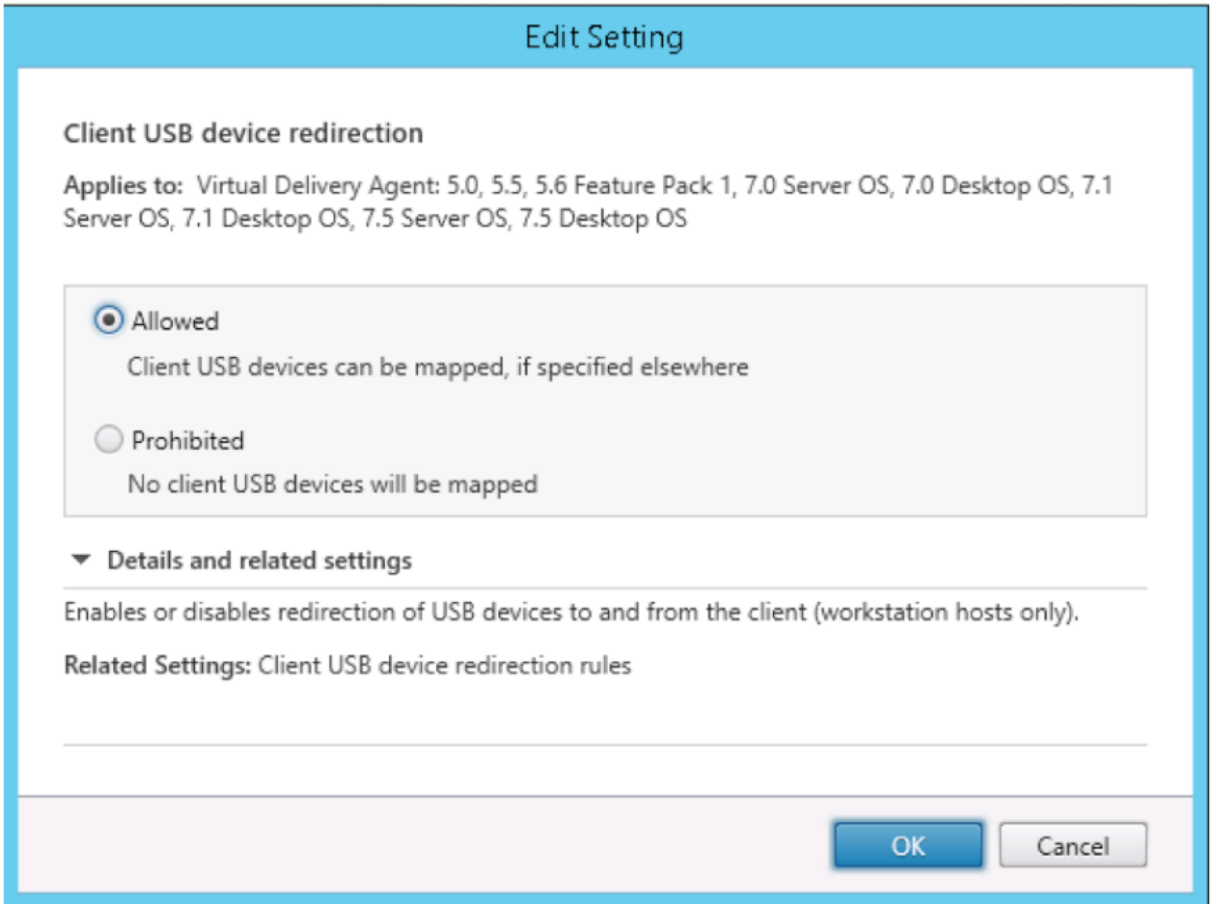
- クライアント USB デバイスリダイレクトポリシー
- クライアント USB デバイスリダイレクト規則

USB リダイレクトポリシーを有効にする

Citrix Studio で、クライアントと USB デバイス間のリダイレクトを有効または無効にします（ワークステーションのホストの場合のみ）。

［設定の編集］ダイアログボックスで、以下の設定を行います。

1. ［許可］を選択します。
2. ［**OK**］をクリックします。

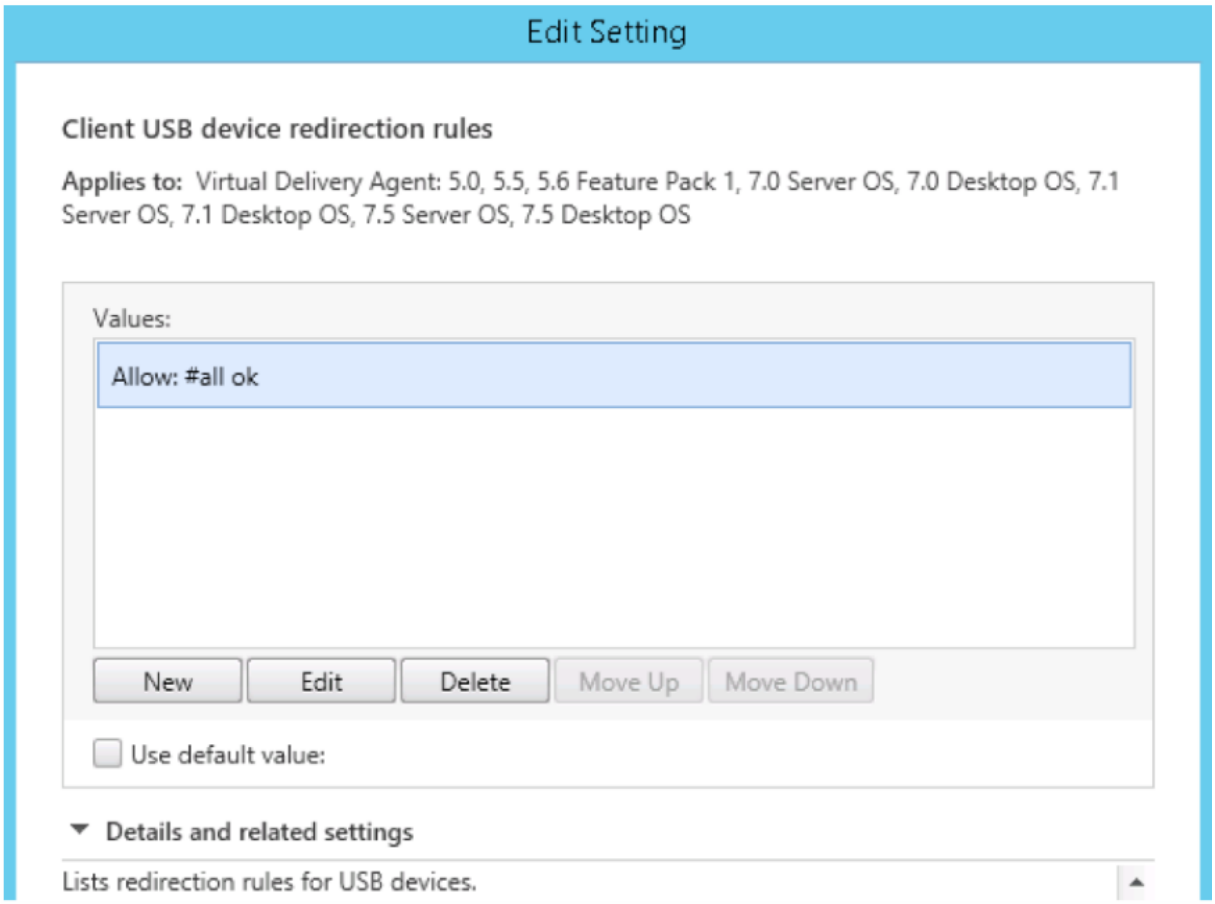


USB リダイレクト規則の設定

USB リダイレクトポリシーを有効にしたら、Citrix Studio を使用して、Linux VDA での使用を許可または禁止するデバイスを指定して、リダイレクト規則を設定します。

［クライアント USB デバイスリダイレクト規則］ダイアログボックスで、

1. ［新規］をクリックしてリダイレクト規則を追加するか、［編集］をクリックして既存の規則を確認します。
2. 規則の作成または編集後、［**OK**］をクリックします。



汎用 USB リダイレクトの設定について詳しくは、「Citrix の汎用 USB リダイレクトの設定ガイド」を参照してください。

VHCI カーネルモジュールを構築します

USB リダイレクトは VHCI カーネルモジュール（`usb-vhci-hcd.ko`および`usb-vhci-iocif.ko`）によって異なります。これらのモジュールは、RPM パッケージの一部として Linux VDA ディストリビューションに含まれます。これらは、Linux 公式ディストリビューションのカーネルをベースにコンパイルされたもので、次の表にまとめられています。

サポートされている Linux ディストリビューション	カーネルバージョン
RHEL 7.7、CentOS 7.7	3.10.0-1062
SUSE 12.3	4.4.73-5-default
Ubuntu 18.04	4.15.0-45-generic

サポートされている Linux ディストリビューション	カーネルバージョン
Ubuntu 16.04	4.4.0-142-generic

> 重要:
>
> 使用するマシンのカーネルが、Linux VDA 用のドライバーに対応していない場合は、USB サービスの起動が失敗することがあります。この場合は、VHCI カーネルモジュールを構築している場合のみ、USB リダイレクト機能を使用できます。

使用するカーネルが **Citrix** の構築したモジュールに対応しているかを確認する

コマンドラインで次のコマンドを実行し、カーネルが対応しているかを確認します:

```
insmod /opt/Citrix/VDA/lib64/usb-vhci-hcd.ko
<!--NeedCopy-->
```

コマンドが正常に実行される場合は、カーネルモジュールのロードに成功し、バージョンが Citrix によりインストールされたものに対応しています。

コマンドの実行でエラーが生じた場合、カーネルは Citrix のモジュールに対応していないため、再構築の必要があります。

VHCI カーネルモジュールの再構築

カーネルモジュールが Citrix のバージョンに対応していない場合は、次の手順に従います。

1. Citrix のダウンロードサイトから、LVDA ソースコードをダウンロードします。セクション ［**Linux Virtual Delivery Agent**（ソース)］に含まれるファイルを選択します。
2. **citrix-linux-vda-sources.zip** ファイルを展開します。**linux-vda-souces/vhci-hcd-1.15.tar.bz2** に移動し、**tar xvf vhci-hcd-1.15.tar.bz2** を使用して VHCI ソースファイルを展開します。
3. ヘッダーファイルおよび **Module.symvers** ファイルに基づいて、カーネルモジュールを構築します。適切な Linux ディストリビューションに基づき、次の手順に従って、カーネルヘッダーファイルをインストールして **Module.symvers** を作成します:

   RHEL/CentOS:

   ```
   yum install kernel-devel
   <!--NeedCopy-->
   ```

   SUSE 12:


```
zypper install kernel-devel

zypper install kernel-source
<!--NeedCopy-->
```

Ubuntu:

```
apt-get install linux-headers
<!--NeedCopy-->
```

> ヒント:
>
> インストールが正常に完了すると、以下に類似したカーネルフォルダーが作成されます:
>
> /usr/src/kernels/3.10.0-327.10.1.el7.x86_64

4. /usr/src/kernels/3.10.0-327.10.1.el7.x86_64 フォルダー内に **Module.symvers** ファイルがあることを確認します。フォルダーにこのファイルがない場合は、カーネルを構築してこのファイルを取得するか（次のコマンドを順番に実行: `make oldconfig`; `make prepare`; `make modules`; `make`）、**/usr/src/kernels/3.10.0-327.10.1.el7.x86_64-obj/x86_64/defaults/module.*** からファイルをコピーします。

5. 次のコマンドを実行して、開発ツールをインストールします。

   RHEL/CentOS:

```
yum groupinstall 'Development Tools'
<!--NeedCopy-->
```

   Ubuntu 18.04:

```
apt install build-essential
apt install libelf-dev
<!--NeedCopy-->
```

   Ubuntu 16.04:

```
apt install build-essential
<!--NeedCopy-->
```

6. **vhci-hcd-1.15/Makefile** ファイルで、VCHI の Makefile を変更し、KDIR をカーネルディレクトリに設定します:

```
#KDIR = $(BUILD_PREFIX)/lib/modules/$(KVERSION)/build

KDIR = /usr/src/kernels/3.10.0-327.10.1.el7.x86_64
<!--NeedCopy-->
```

7. **vhci-hcd-1.15/**フォルダーで、`make`コマンドを実行して VHCI カーネルを構築します。

> 注:
>
> 構築に成功すると、usb-vhci-hcd.koおよびusb-vhci-iocifc.koが **vhci-hcd-1.15/**フォルダーに作成されます。

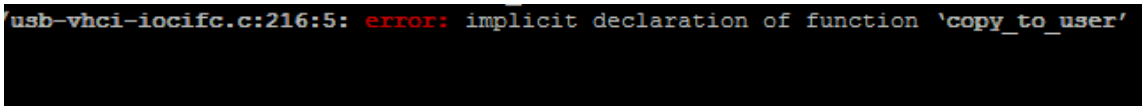
8. カーネルモジュールを新しく構築したモジュールに置き換えます: **cp -f usb-vhci-*.ko /opt/Citrix/VDA/lib64/**
9. USB サービスを再起動します:

```
service ctxusbsd restart
<!--NeedCopy-->
```

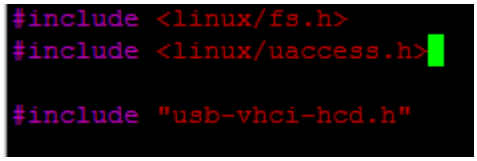
10. ログオフして再びセッションに戻ります。USB リダイレクトが機能しているかを確認します。

カーネル構築の問題のトラブルシューティング

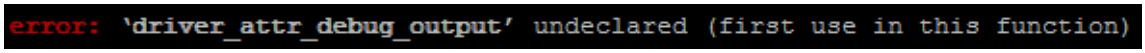
- **Ubuntu 16** の特定のカーネルで、カーネル構築エラーが発生することがあります。このエラーは、implicit declaration of function ‘copy_to_user’ と表示されます。次のスクリーンショットを参照してください。



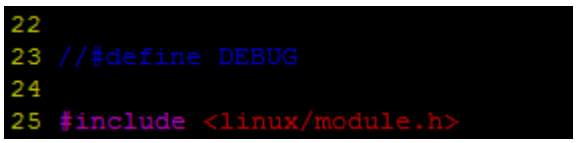
このエラーは、カーネルのヘッダーファイルの変更が原因で発生します。この問題を回避するには、vhci-hcd-1.15/usb-vhci-iocifc.cファイルに **#include <linux/uaccess.h>** という行を追加します。



- **Ubuntu 16** のカーネル **4.15.0-29-generic** で、カーネル構築エラーが発生することがあります。このエラーは、‘driver_attr_debug_output’ undeclaredと表示されます。次のスクリーンショットを参照してください。



このエラーは、カーネルのシンボルが欠落している場合に発生します。この問題を回避するには、vhci-hcd-1.15/usb-vhci-iocifc.cファイルとvhci-hcd-1.15/usb-vhci-hcd.cファイルで DEBUG のマクロ定義を無効にします。

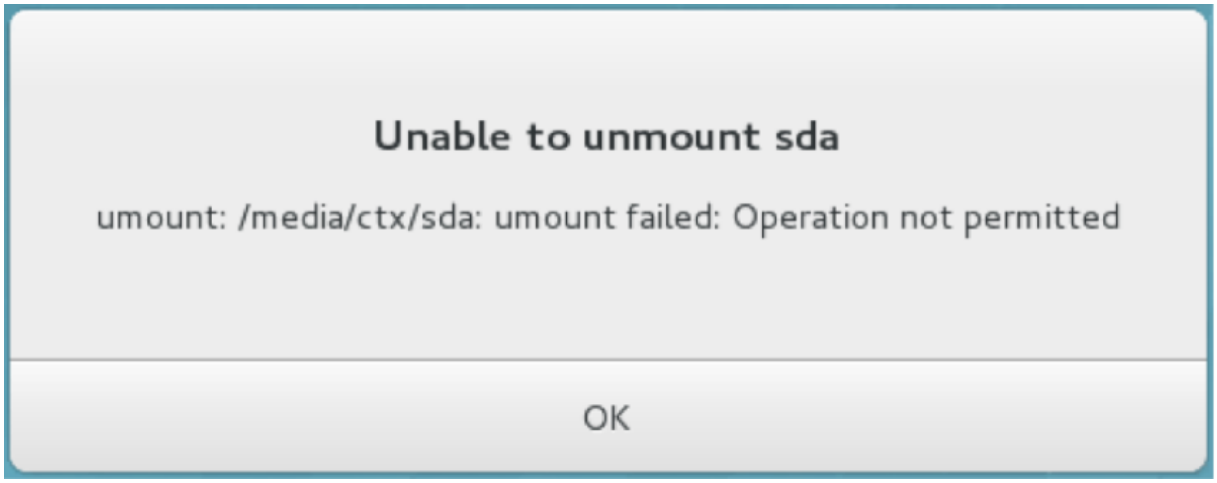


USB リダイレクト問題のトラブルシューティング

このセクションでは、Linux VDA の使用におけるさまざまな問題のトラブルシューティングについて説明します。

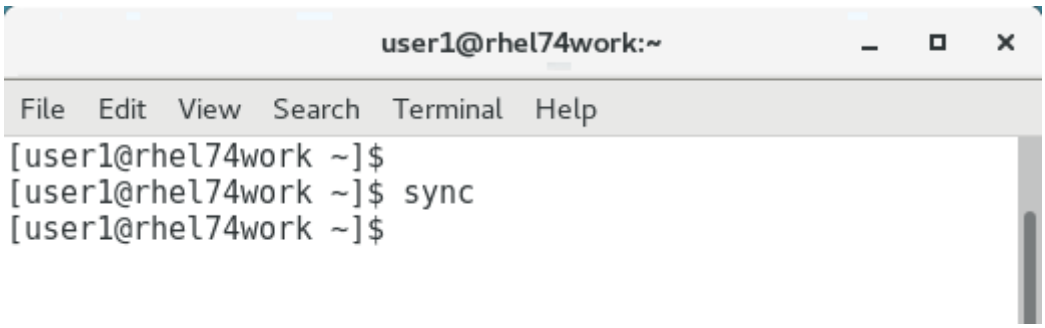
リダイレクトされた USB ディスクをマウント解除できない **

Linux VDA では、Citrix Workspace アプリからリダイレクトされたすべての USB ディスクのアクセスを制御するため、これらのデバイスをすべて管理者権限で管理し、リダイレクトされたデバイスに所有者のみがアクセスできるようにしています。そのため、管理者権限を持たないユーザーはデバイスをマウント解除できません。



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

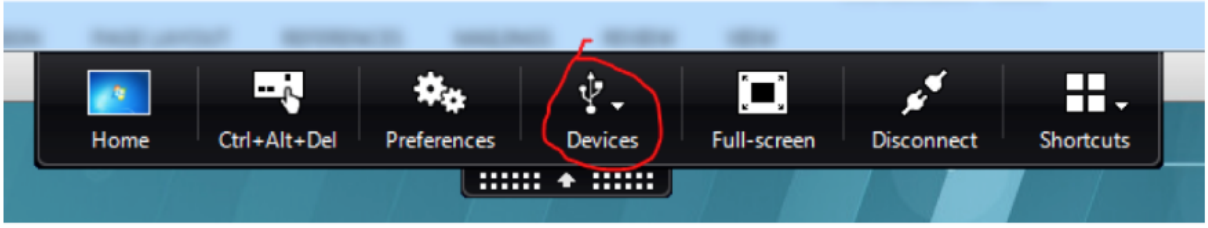
USB ディスクをセッションにリダイレクトして、ディスク上でのファイルの作成などでディスクを変更した後に、Citrix Workspace アプリのツールバーを使用して直ちにリダイレクトを停止すると、変更または作成したファイルが失われることがあります。この問題は、ファイルシステムにデータを書き込むとメモリキャッシュがファイルシステムにマウントされることが原因で発生します。データはディスクそのものには書き込まれません。Citrix Workspace アプリのツールバーを使用してリダイレクトを停止した場合、データがディスクにフラッシュされる時間が残っていないため、データが失われます。この問題を解決するには、ターミナルの sync コマンドを使用してデータをディスクにフラッシュしてから USB リダイレクトを停止します。



Citrix Workspace アプリのツールバーにデバイスが見つからない場合

Citrix Workspace アプリのツールバーにデバイスが表示されなくなることがありますが、これは USB リダイレクトが行われていないことを示します。問題が発生した場合は、次の点を確認してください。

- ポリシーが、USB リダイレクトを許可する設定になっている
- カーネルモジュールが、使用するカーネルに対応している



> 注:
>
> ［デバイス］タブは Linux 向け Citrix Workspace アプリで使用できません。

Citrix Workspace アプリのツールバーに **USB** デバイスが表示されるが ［ポリシーの制限］と表記されリダイレクトが失敗する

問題が発生した場合は、次の手順を実行してください:

- Linux VDA ポリシーを、リダイレクトを有効にする設定にします。
- Citrix Workspace アプリのレジストリで追加のポリシー制限が構成されているかを確認します。レジストリパスで **DeviceRules** を確認し、この設定がデバイスのアクセスを拒否していないことを確認します:

  HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA Client\GenericUSB

詳しくは、Knowledge Center の記事「USB デバイスの自動リダイレクトの設定方法」を参照してください。

USB デバイスのリダイレクトは正常に行われるが、セッションでデバイスを使用できない

通常、リダイレクトできるのはサポートされている USB デバイスのみとなります。他のデバイスが Linux VDA のアクティブなセッションにリダイレクトできる場合もあります。この場合、リダイレクトしたデバイスごとに、ユーザーの所有するノードがシステムの**/dev** パスに作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかはドライバーと構成によって決定されます。所有（プラグイン）しているもののアクセスできないデバイスを見つけた場合は、そのデバイスを制限されていないポリシーに追加します。

> 注:
>
> USB ドライバーの場合は、Linux VDA がディスクの設定とマウントを行います。ユーザー（およびデバイスを

> インストールした所有者のみ）は追加の設定なしでディスクにアクセスできます。「サポートされているデバイス一覧」に掲載されていないデバイスについては、これが適用されないことがあります。

セッション画面の保持の構成

December 13, 2022

Citrix でサポートされているすべての Linux プラットフォームには、セッション画面の保持機能が導入されています。セッション画面の保持は、デフォルトで有効になっています。

セッション画面の保持によって ICA セッションは、ネットワークの中断を挟んでもシームレスに再接続されます。セッション画面の保持について詳しくは、「クライアントの自動再接続とセッション画面の保持」を参照してください。

注：セッション画面の保持の接続を介して送信されるデータは、デフォルトではプレーンテキストです。セキュリティを確保するため、TLS 暗号化を有効にすることをお勧めします。TLS 暗号化について詳しくは、「TLS によるユーザーセッションの保護」を参照してください。

構成

Citrix Studio のポリシー設定

Citrix Studio で、セッション画面の保持に関する次のポリシーを設定できます。

- セッション画面の保持
- セッション画面の保持のタイムアウト
- セッション画面の保持のポート番号
- 再接続 UI の透過レベル

詳しくは、「セッション画面の保持のポリシー設定」および「クライアントの自動再接続のポリシー設定」を参照してください。

注：セッション画面の保持の接続またはセッション画面の保持のポート番号ポリシーを設定したら、VDA サービスと HDX サービスをこの順序で再起動して設定を有効にします。

Linux VDA の設定

- セッション画面の保持の **TCP** リスナーを有効または無効にする

デフォルトでは、セッション画面の保持の TCP リスナーは有効になっており、ポート 2598 でリッスンします。リスナーを無効にするには、次のコマンドを実行します。


```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "
    fEnableWinStation" -d "0x00000000"
<!--NeedCopy-->
```

注：設定を有効にするには、HDX サービスを再起動してください。TCP リスナーを無効にしても、セッション画面の保持は無効になりません。セッション画面の保持の接続ポリシーによって機能が有効になっている場合、セッション画面の保持は他のリスナー（SSL など）を介して引き続き利用できます。

- セッション画面の保持のポート番号

次のコマンドを使用して、セッション画面の保持のポート番号を設定することもできます（例としてポート番号 2599 を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k  "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "PortNumber"
     -d  "2599"
<!--NeedCopy-->
```

注：設定を有効にするには、HDX サービスを再起動してください。Citrix Studio のポリシー設定でポート番号が設定されている場合、Linux VDA の設定は無視されます。VDA のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように設定されていることを確認します。

- サーバーからクライアントへの **Keep-Alive** の間隔

セッション画面の保持の Keep-Alive メッセージは、セッション中にアクティビティがない場合（例：マウスが移動しない、画面が変更しない）、Linux VDA と ICA クライアント間で送信されます。Keep-Alive メッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントからの応答がない場合、セッションは、クライアントが再接続するまで中断されます。この設定では、Keep-Alive メッセージの送信間隔を秒単位で指定します。デフォルトでは、この設定は構成されていません。構成するには、次のコマンドを実行します（例として 10 秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    Citrix\XTEConfig"  -t "REG_DWORD"  -v  "CgpServerToClientKeepAlive"
     -d  "10"  --force
```

- クライアントからサーバーへの **Keep-Alive** の間隔

この設定では、ICA クライアントから Linux VDA に送信される Keep-Alive メッセージの送信間隔を秒単位で指定します。デフォルトでは、この設定は構成されていません。構成するには、次のコマンドを実行します（例として 10 秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    Citrix\XTEConfig"  -t "REG_DWORD"  -v  "CgpClientToServerKeepAlive"
     -d  "10"  --force
<!--NeedCopy-->
```


トラブルシューティング

ポリシーの設定によってセッション画面の保持を有効にした後に、セッションを起動できない。

この問題を解決するには、以下の手順に従います。

1. Citrix Studio のポリシー設定でセッション画面の保持を有効にした後、VDA サービスと HDX サービスがこの順序で再起動されることを確認します。
2. VDA で、次のコマンドを使用してセッション画面の保持の TCP リスナーが実行されていることを確認します（例としてポート 2598 を使用）。

```
netstat -an | grep 2598
<!--NeedCopy-->
```

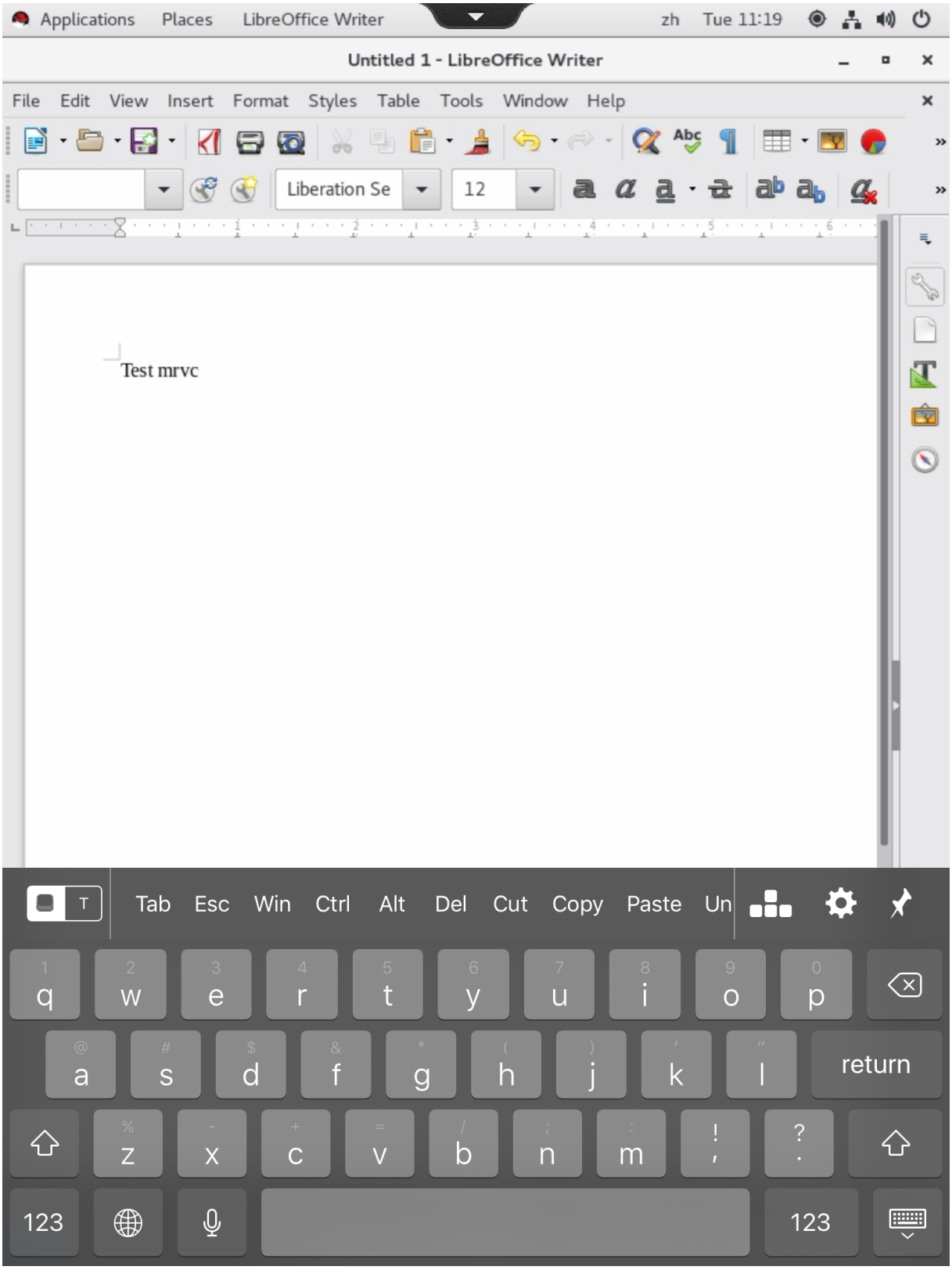
セッション画面の保持のポートに TCP リスナーがない場合は、次のコマンドを実行してリスナーを有効にします。

```
/opt/Citrix/VDA/bin/ctxreg update -k  "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "
    fEnableWinStation" -d "0x00000001"
<!--NeedCopy-->
```

ソフトキーボード

November 11, 2021

ソフトキーボード機能は、Linux 仮想デスクトップまたはアプリケーションのセッションで利用できます。ソフトキーボードは、入力フィールドで入力を開始すると表示され、入力を終了すると非表示になります。



> 注:
>
> この機能は RHEL 7.7、CentOS 7.6、SUSE 12.3、Ubuntu 16.04、Ubuntu 18.04 で利用できます。iOS 向け Citrix Workspace アプリおよび Android 向け Citrix Workspace アプリでサポートされています。

機能の有効化と無効化

この機能はデフォルトでは無効になっています。**ctxreg** ユーティリティを使用して、この機能を有効または無効にします。特定の Linux VDA の機能構成は、その VDA のすべてのセッションに適用されます。

この機能を有効にするには:

1. 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"
<!--NeedCopy-->
```

2. Citrix Studio でキーボードの自動表示ポリシーを ［許可］に設定します。
3. （オプション）RHEL 7 および CentOS 7 の場合、次のコマンドを実行して Intelligent Input Bus（IBus）をデフォルトの IM サービスとして構成します:

```
echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
<!--NeedCopy-->
```

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"
<!--NeedCopy-->
```

> 注:
>
> これらの設定は、新しいセッションにログオンする場合、またはログオフして現在のセッションに戻る場合に有効になります。

制限事項

- この機能は Google Chrome、LibreOffice、その他のアプリで機能しないことがあります。
- 手動でソフトキーボードを非表示にした後再表示するには、入力フィールド以外をクリックして、再度現在の入力フィールドをクリックします。
- Web ブラウザーで 1 つの入力フィールドをクリックしてから別のフィールドをクリックすると、ソフトキーボードが表示されないことがあります。この問題を回避するには、入力フィールド以外をクリックしてから対象の入力フィールドをクリックします。

- この機能は、Unicode 文字やダブルバイト文字（日本語、中国語、韓国語など）をサポートしません。
- ソフトキーボードは、パスワード入力フィールドでは利用できません。
- ソフトキーボードは、現在の入力フィールドと重なって表示されることがあります。この場合、アプリのウィンドウを移動するか、画面を上にスクロールして入力フィールドをアクセスできる位置に移動します。
- Citrix Workspace アプリと Huawei タブレットとの互換性の問題によって、Huawei タブレットに物理キーボードが接続されている場合でもソフトキーボードが表示されます。

クライアント入力システム（**IME**）

November 21, 2020

概要

2 バイト文字（日本語、中国語、韓国語などの文字）は、IME から入力する必要があります。Windows ネイティブの CJK IME など、クライアント側で Citrix Workspace アプリと互換性がある任意の IME を使用して、これらの文字を入力します。

インストール

この機能は、Linux VDA をインストールするときに自動でインストールされます。

用途

通常どおりに Citrix Virtual Apps または Citrix Virtual Desktops のセッションを開きます。

クライアント側 IME 機能の使用を開始するには、クライアント側での必要に応じて入力方式を変更します。

既知の問題

- クライアント側 IME 機能を使用して Google スプレッドシートのセルに文字を入力するには、スプレッドシート内のセルをダブルクリックする必要があります。
- クライアント側 IME 機能は［パスワード］フィールドで自動で無効になりません。
- IME ユーザーインターフェイスは、入力領域ではカーソルに追従しません。

多言語入力のサポート

March 9, 2022

Linux VDA バージョン 1.4 以降では、Citrix で公開アプリケーションのサポートが追加されています。ユーザーは、Linux デスクトップ環境がなくても、必要な Linux アプリケーションにアクセスできます。

ただし、言語バーは Linux デスクトップ環境と高度に統合されているため、Linux VDA のネイティブ言語バーは公開アプリケーションでは使用できませんでした。その結果、中国語、日本語、韓国語など、IME が必要な言語でテキストを入力できませんでした。さらに、ユーザーがアプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するために、この機能で、テキスト入力に対応した公開アプリケーション用の言語バーを提供します。言語バーを使用すると、サーバー側の IME を選択したり、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

ctxreg ユーティリティを使用して、この機能を有効または無効にすることができます（デフォルトでは無効）。特定の Linux VDA サーバーの機能設定は、その VDA に公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar」で、種類は DWORD です。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE \SYSTEM\
    CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
    x00000001"
<!--NeedCopy-->
```

この機能を無効にするには、次のコマンドを実行します:

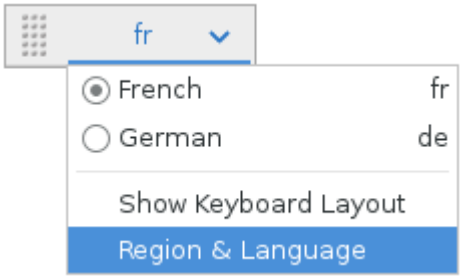
```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE \SYSTEM\
    CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
    x00000000"
<!--NeedCopy-->
```

用途

使い方は簡単です。

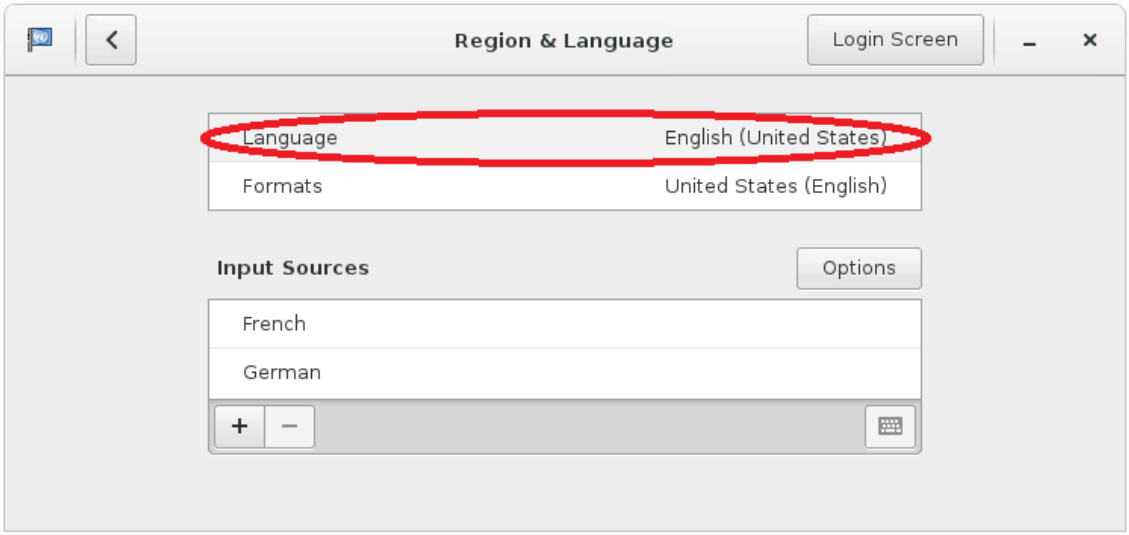
1. 本機能を有効にします。

2. テキスト入力に対応できる公開アプリケーションにアクセスします。言語バーが、アプリケーションとともにセッションに表示されます。
3. ドロップダウンメニューから、[地域と言語] を選択して希望の言語（入力ソース）を追加します。



4. ドロップダウンメニューから IME またはキーボードレイアウトを選択します。
5. 選択した IME またはキーボードレイアウトを使用して言語を入力します。

> 注:
>
> - VDA 側の言語バーでキーボードレイアウトを変更する場合、クライアント側（Citrix Workspace アプリが実行されている）でも同じキーボードレイアウトが使用されていることを確認してください。
> - [地域と言語] ダイアログボックスで設定を行うには、**accountsservice** パッケージをバージョン 0.6.37 以降にアップグレードする必要があります。



動的なキーボードレイアウトの同期

November 11, 2021

以前は、Linux VDA とクライアントデバイスのキーボードレイアウトは同じでなければなりませんでした。たとえば、キーボードレイアウトがクライアントデバイスで英語からフランス語に変更され、VDA では変更されなかった場合、キーマッピングの問題が発生し、VDA がフランス語に変更されるまで問題が存続することがありました。

この問題は、Citrix では VDA のキーボードレイアウトとクライアントデバイスのキーボードレイアウトを自動的に同期させることで対処しました。クライアントデバイスのキーボードレイアウトが変更されるたびに、VDA のレイアウトも変更されます。

> ヒント：
>
> この機能は Windows 向け Citrix Workspace アプリでサポートされており、公開されたアプリとデスクトップの両方に対応しています。

構成

この機能はデフォルトでは無効になっています。**ctxreg** ユーティリティを使用して、この機能を有効または無効にします。特定の Linux VDA の機能構成は、その VDA のすべてのセッションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control\Citrix\SyncKeyboardLayout」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 "SyncKeyboardLayout
    " -d "0x00000001"
<!--NeedCopy-->
```

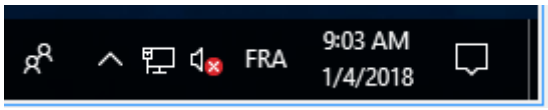
この機能を無効にするには、次のコマンドを実行します：

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\
    CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncKeyboardLayout
    " -d "0x00000000"
<!--NeedCopy-->
```

用途

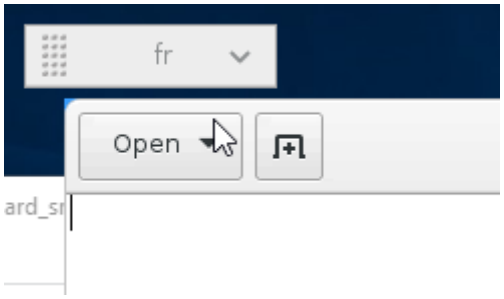
この機能を有効にすると、セッション中にクライアントデバイス上でキーボードレイアウトが変更された場合、セッションのキーボードレイアウトもそれに応じて変更されます。

たとえば、クライアントデバイスのキーボードレイアウトをフランス語（FR）に変更すると、次のようになります。

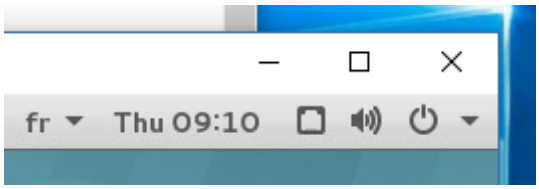


Linux VDA セッションのキーボードレイアウトも「fr」に変わります。

アプリケーションセッションでは、言語バーを有効にしている場合、この自動変更が表示されます。



デスクトップセッションでは、この自動変更がタスクバーに表示されます:

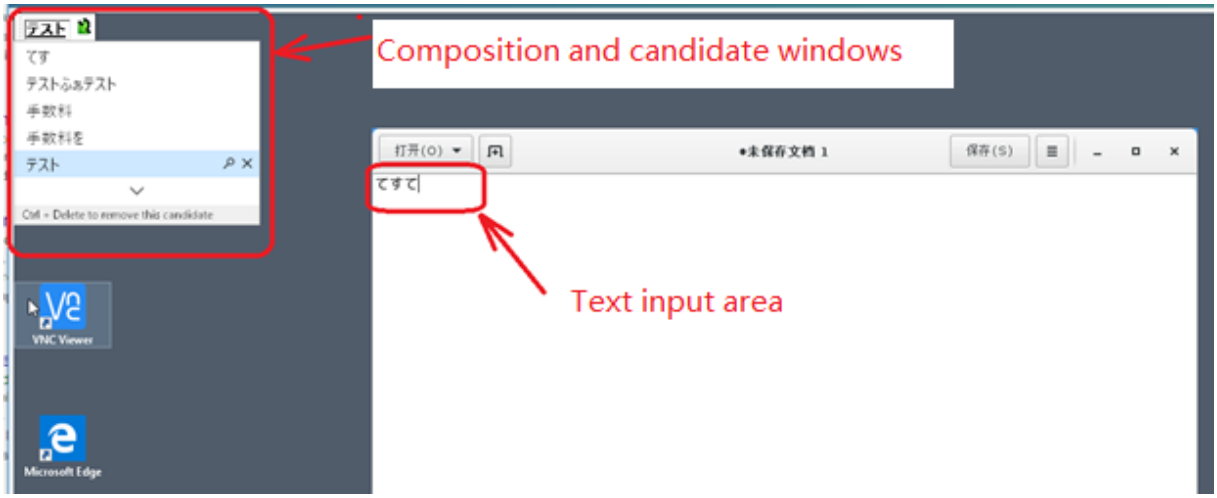


クライアント側 IME ユーザーインターフェイスの同期

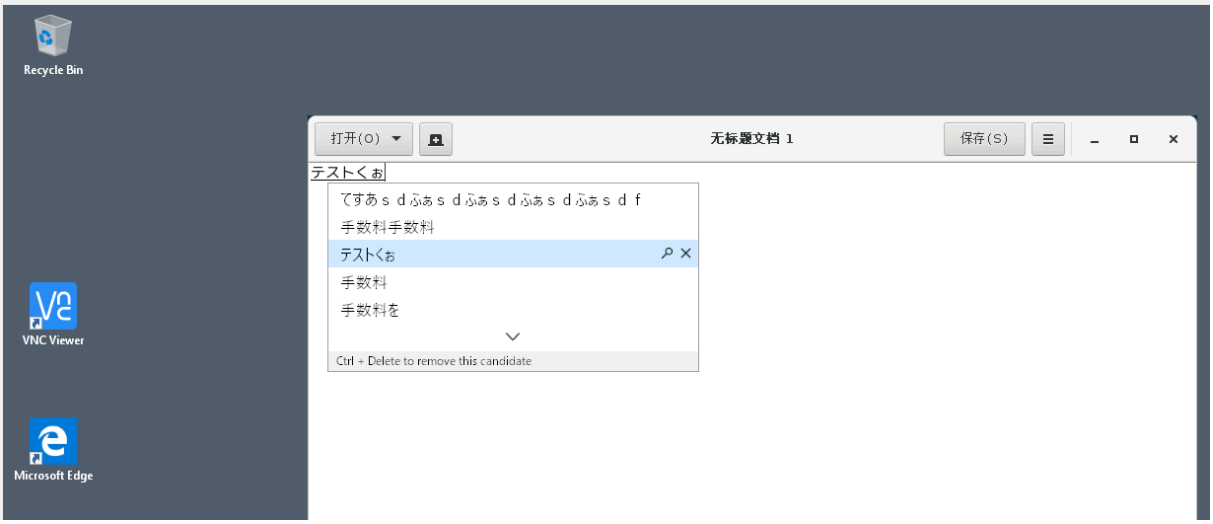
March 11, 2022

概要

クライアント側 IME ユーザーインターフェイス（作成ウィンドウと候補ウィンドウを含む）は、これまで画面の左上隅に配置されていました。このインターフェイスはカーソルに追従せず、テキスト入力領域ではカーソルから離れて配置されることがありました:



Citrix ではユーザービリティが強化され、以下のように、クライアント側 IME でのシームレスな操作がさらに改善されています:



> 注:
>
> この機能は、RHEL 7.x、CentOS 7.x、Ubuntu 16.04、Ubuntu 18.04、SUSE 12.x。Windows 向けおよび Mac 向け Citrix Workspace アプリでは、エコーキャンセルがサポートされています。
>
> RHEL 7.x デスクトップセッションでこの機能を使用するには、IBus を有効にする必要があります。たとえば、入力に IME が必要なユーザーインターフェイス言語を設定するか、**GTK_IM_MODULE=ibus** を **${HOME}/.config/imsettings/xinputrc** ファイルに追加します。
>
> 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

この機能はデフォルトでは無効になっています。**ctxreg** ユーティリティを使用して、この機能を有効または無効にします。特定の Linux VDA の機能構成は、その VDA のすべてのセッションに適用されます。

この機能を有効にするには、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"
<!--NeedCopy-->
```

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"
<!--NeedCopy-->
```


HDX Insight

February 9, 2024

概要

HDX Insight は、Citrix Application Delivery Management（ADM）の一部であり、一般的な業界標準の AppFlow をベースにしています。この機能は Citrix ADC または Citrix SD-WAN アプリケーションネットワークファブリックを経由する Citrix ICA トラフィックに対して、エンドツーエンドの優れた可視性を実現し、IT を通じて優れたユーザーエクスペリエンスを提供できるようにします。

Linux VDA では、HDX Insight 機能の一部をサポートしています。EUEM（End User Experience Monitoring: エンドユーザー状況監視）機能は実装されていないため、期間に関連するデータポイントの一部を使用できません。

インストール

インストールする必要のある依存関係パッケージはありません。

使用状況

HDX Insight は、Citrix Workspace アプリと Linux VDA の間で Citrix ADC を介して渡される ICA メッセージを分析します。

Linux VDA を含む NetScaler Insight Center 展開をセットアップし、HDX Insight 機能を有効にする必要があります。NetScaler Insight Center の展開を、既存の構成や設定、データを失うことなく、Citrix ADM に移行できます。詳しくは、「NetScaler Insight Center から Citrix ADM への移行」を参照してください。

トラブルシューティング

データポイントがまったく表示されない

2 通りの原因が考えられます。

- HDX Insight が正しく構成されていません。

  たとえば、Citrix ADC で AppFlow が有効になっていないか、Citrix ADM で不正な Citrix ADC インスタンスが構成されています。

- Linux VDA で ICA コントロール仮想チャネルが開始されていません。

  ```
  ps aux | grep -i ctxctl
  ```

  `ctxctl`が実行されていない場合は、Citrix にバグをレポートするよう管理者に連絡します。

アプリケーションデータポイントがまったく表示されない

シームレス仮想チャネルが有効になっていることおよびシームレスアプリケーションが起動されてしばらく経過していることを確認します。

アダプティブトランスポート

November 11, 2021

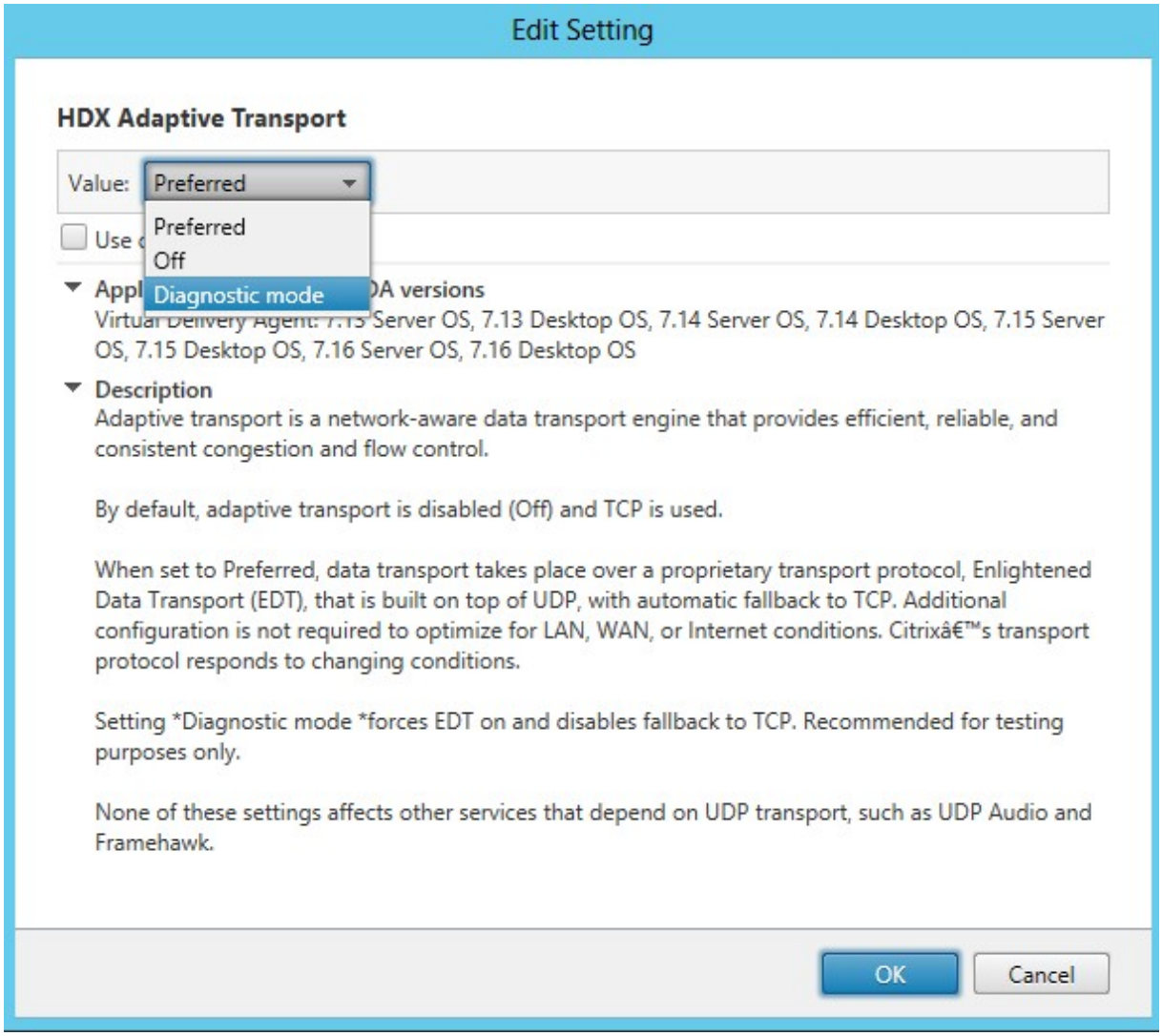
これまで実験的に導入されていたアダプティブトランスポート機能は、このリリースで本格的に導入されます。

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。高速で拡張性が高く、アプリケーションの対話機能が向上し、厳しい長距離の WAN とインターネット接続でのインタラクティブ性を高めます。詳しくは、「アダプティブトランスポート」を参照してください。

アダプティブトランスポートを有効にする

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。デフォルトでは、［優先］が選択されています。

- 優先：可能な場合、Enlightened Data Transport（EDT）でのアダプティブトランスポートが使用され、TCP にフォールバックします。
- 診断 モード：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** アダプティブトランスポート］ポリシーを［オフ］に設定します。

トラブルシューティング

アダプティブトランスポートが有効かどうかを確認する

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"
<!--NeedCopy-->
```


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                 :::*
<!--NeedCopy-->
```

トレースオン

October 9, 2021

概要

ログの収集および問題の再現によって、診断速度やユーザーエクスペリエンスが低下します。トレースオン機能は、こうした事態に対応します。トレースは、Linux VDA でデフォルトで有効になっています。

構成

Linux VDA パッケージに、`ctxlogd`デーモンおよび`setlog`ユーティリティが追加されました。`ctxlogd`デーモンは、Linux VDA をインストールして構成すると、デフォルトで開始されます。

ctxlogd デーモン

トレースされた他のサービスはすべて`ctxlogd`デーモンに依存しています。Linux VDA をトレースしない場合は、`ctxlogd`デーモンを停止できます。

setlog ユーティリティ

トレースオン機能は、`setlog`ユーティリティ（パス：**/opt/Citrix/VDA/bin/**）で構成されます。このユーティリティを実行する権限があるのは、ルートユーザーのみです。GUI を使用するかコマンドを実行して、構成を表示したり変更したりできます。`setlog`ユーティリティのヘルプを表示するには、次のコマンドを実行します：

```
setlog help
<!--NeedCopy-->
```

値　デフォルトでは、［ログ出力パス］は **/var/log/xdl/hdx.log**、［最大ログサイズ］は 200MB に設定されています。［ログ出力パス］には、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
<!--NeedCopy-->
```

単一の`setlog`値を表示または設定します:

```
setlog value <name> [<value>]
<!--NeedCopy-->
```

例:

```
setlog value log_size 100
<!--NeedCopy-->
```

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。

さまざまなコンポーネントに設定された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
<!--NeedCopy-->
```

ログレベル（Disabled、Inherited、Verbose、Information、Warnings、Errors、Fatal Errors）を設定するには、次のコマンドを実行します:

```
setlog level <class> [<level>]
<!--NeedCopy-->
```

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）
Disabled	none
Inherited	inherit
Verbose	verbose
Information	info
Warnings	warning
Errors	error
Fatal Errors	fatal

`<class>`変数は、Linux VDA の 1 つのコンポーネントを指定します。すべてのコンポーネントをカバーするには、all に設定します。例:


```
setlog level all error
<!--NeedCopy-->
```

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
<!--NeedCopy-->
```

現在のフラグを表示します:

```
setlog flags
<!--NeedCopy-->
```

1 つのログフラグを表示または設定します:

```
setlog flag <flag> [<state>]
<!--NeedCopy-->
```

デフォルトに戻す　すべてのレベル、フラグ、値をデフォルト設定に戻します:

```
setlog default
<!--NeedCopy-->
```


> 重要:
>
> ctxlogdサービスは**/var/xdl/.ctxlog** ファイルを使用して構成されます。このファイルは、ルートユーザーのみが作成できます。他のユーザーは、このファイルへの書き込み権限がありません。ルートユーザーが他のユーザーに書き込み権限を許可しないことを Citrix ではお勧めします。許可すると、ctxlogdが恣意的に、または悪意をもって構成される危険性があります。これによってサーバーのパフォーマンスが影響を受け、ユーザーエクスペリエンスにも影響を与える可能性があります。

トラブルシューティング

/var/xdl/.ctxlog ファイルがない場合（過失による削除など）、ctxlogdデーモンが失敗し、ctxlogdサービスを再起動できません。

/var/log/messages:

```
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.
<!--NeedCopy-->
```

この問題を解決するには、ルートユーザーとしてsetlogを実行して、**/var/xdl/.ctxlog** ファイルを再度作成します。次に、他のサービスが依存するctxlogdサービスを再起動します。

セッションのシャドウ

November 11, 2021

セッションのシャドウ機能により、ドメイン管理者はイントラネット内のユーザーの ICA セッションを閲覧できます。この機能は、noVNC を使用して ICA セッションに接続し、RHEL 7.x と Ubuntu 16.04 でのみサポートされています。

> 注:
>
> セッションのシャドウ機能を使用するには、Citrix Director のバージョンを 7.16 以降にする必要があります。

インストールと構成

依存関係

セッションのシャドウには、python-websockifyとx11vncという、2 つの新しい依存関係が必要です。python-websockifyとx11vncの依存関係は、Ubuntu 16.04 に 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。RHEL 7.x では、Linux VDA をインストールした後に、python-websockifyとx11vncを手動でインストールする必要があります。

RHEL 7.x で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 yum install -y python-websockify x11vnc
<!--NeedCopy-->
```

python-websockifyとx11vncを解決するには、RHEL 7.x 上で次のリポジトリを有効にします：

- Extra Packages for Enterprise Linux (EPEL)

  python-websockifyとx11vncの両方に EPEL リポジトリが必要です。次のコマンドを実行して、EPEL リポジトリを有効にします：

  ```
  sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(rpm -E '%{
  rhel }
  ').noarch.rpm
  <!--NeedCopy-->
  ```

- オプションの RPM

  x11vnc 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 RPM リポジトリを有効にするには、次のいずれかのコマンドを実行します。

  ワークステーションの場合：

  ```
  subscription-manager repos --enable=rhel-7-workstation-optional-rpms
  <!--NeedCopy-->
  ```

  サーバーの場合：

  ```
  subscription-manager repos --enable=rhel-7-server-optional-rpms
  <!--NeedCopy-->
  ```

ポート

セッションのシャドウ機能は、Linux VDA から Citrix Director への接続を構築するために、6001～6099 の範囲内で使用可能なポートを自動的に選択します。したがって、同時にシャドウできる ICA セッションの数は 99 に制限さ

れています。要件を満たすために、特にマルチセッションのシャドウ用に十分なポートがあることを確認してください。

レジストリ

次の表は、関連するレジストリの一覧です:

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションのシャドウ機能を有効または無効にします。	1（有効）
ShadowingUseSSL	Linux VDA と Citrix Director 間の接続を暗号化するかどうかを決定します。	0（無効）

Linux VDA で`ctxreg`コマンドを実行して、レジストリ値を変更します。たとえば、セッションシャドウを無効にするには、次のコマンドを実行します:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
    VirtualDesktopAgent" -v "EnableSessionShadowing" -d 0x00000000
```

SSL

Linux VDA と Citrix Director 間の noVNC 接続では、WebSocket プロトコルが使用されます。セッションのシャドウの場合、`ws://`と`wss://`のどちらが選択されるかは、前述の「ShadowingUseSSL」レジストリによって決まります。デフォルトでは、`ws://`が選択されています。ただし、セキュリティ上の理由から、Citrix は、`wss://`を使用して各 Citrix Director クライアントと各 Linux VDA サーバーに証明書をインストールすることをお勧めします。`ws://`を使用した Linux VDA セッションのシャドウについては、シトリックスはセキュリティ上のいかなる責任も負いません。

サーバー証明書とルート **SSL** 証明書を取得する　証明書には、信頼された証明機関（CA）による署名が必要です。

Linux VDA サーバーで SSL を設定する場合は、サーバーごとに個別のサーバー証明書（キーを含む）が必要です。また、サーバー証明書によって各コンピューターが識別されるため、各サーバーの完全修飾ドメイン名（FQDN）を調べる必要があります。便宜上、代わりにドメイン全体にワイルドカード証明書を使用できます。この場合、少なくともドメイン名を知っておく必要があります。

各サーバーにサーバー証明書をインストールするだけでなく、Linux VDA サーバーを使用して通信を行う各 Citrix Director クライアントに、同じ証明機関が発行するルート証明書をインストールする必要があります。ルート証明書は、サーバー証明書と同じ証明機関から入手できます。サーバー証明書とクライアント証明書は、オペレーティングシステムに組み込まれている証明機関、社内証明機関（社内で構築する証明機関）、またはオペレーティングシステム

に組み込まれていない証明機関のものをインストールできます。証明書を取得するためにどの手段を取るべきかについては、社内のセキュリティ担当部門に問い合わせてください。

重要:

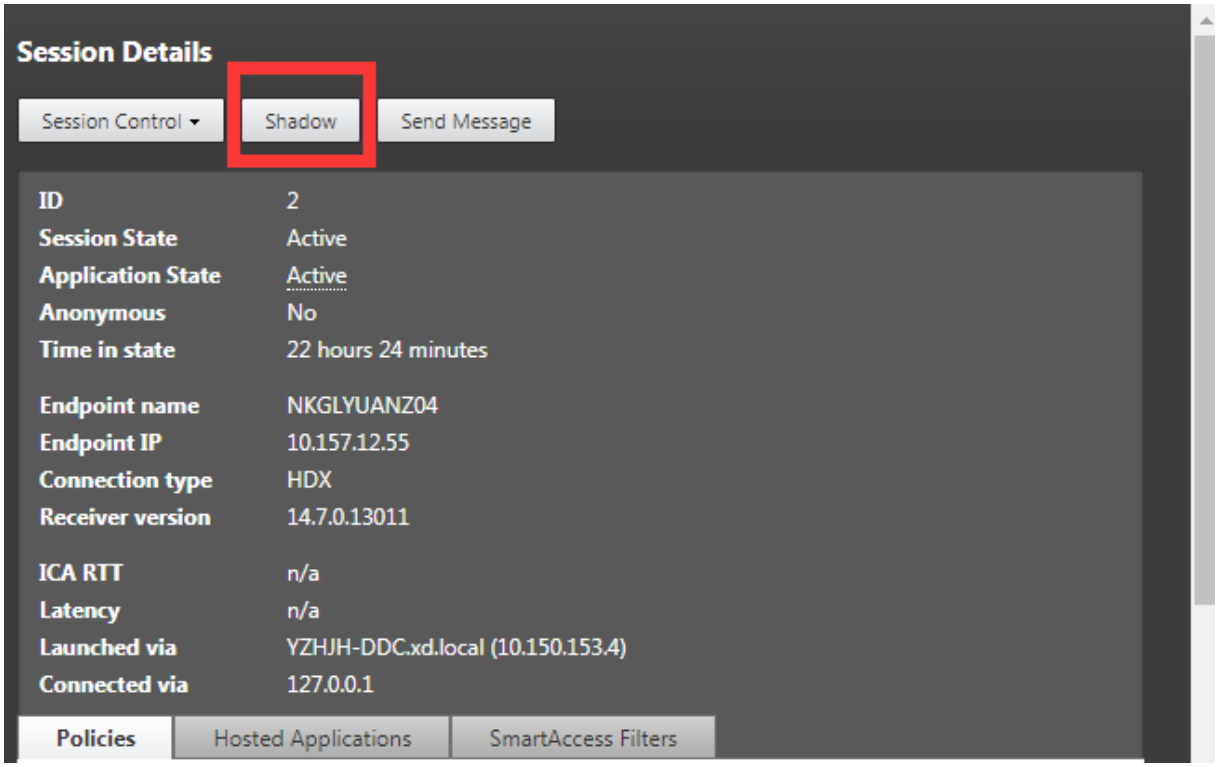
- サーバー証明書の共通名は、Linux VDA サーバーの正確な FQDN、または少なくともワイルドカードとドメイン文字を正しく組み合わせたものである必要があります。たとえば、vda1.basedomain.com や *.basedomain.com などです。
- SHA1 や MD5 などのハッシュアルゴリズムは、一部のブラウザーでサポートされるデジタル証明書の署名には弱すぎます。したがって、SHA-256 が最低基準として指定されています。

各 **Citrix Director** クライアントにルート証明書をインストールする　セッションのシャドウと IIS で、同じレジストリベースの証明書ストアを使用するため、IIS または Microsoft 管理コンソール（MMC）の証明書スナップインを使用してルート証明書をインストールできます。証明機関から証明書を取得したら、IIS のサーバー証明書ウィザードを再び起動します。この操作により、自動的に証明書がインポートされます。または、Microsoft 管理コンソールの証明書スナップインで証明書を表示して、サーバーにインストールすることもできます。Internet Explorer と Google Chrome は、デフォルトで、オペレーティングシステムにインストールされている証明書をインポートします。Mozilla Firefox の場合、証明書マネージャーの ［認証局証明書］タブでルート SSL 証明書をインポートする必要があります。

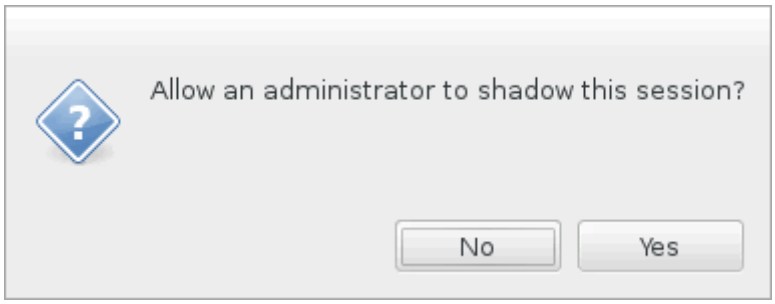
各 **Linux VDA** サーバーにサーバー証明書とそのキーをインストールする　サーバー証明書に「shadowingcert.*」、キーファイルに「shadowingkey.*」と名前を指定します（* は、shadowingcert.csr や shadowingkey.key のように、形式を示すことができます）。サーバー証明書とキーファイルを、パス **/etc/xdl/shadowingssl** の下に置き、制限付きの権限で適切に保護します。間違った名前やパスを使用すると、Linux VDA は特定の証明書やキーファイルを見つけることができなくなり、Citrix Director との接続に失敗することがあります。

用途

Citrix Director からターゲットのセッションを見つけ、［セッション詳細］ビューで ［シャドウ］をクリックして、シャドウの要求を Linux VDA に送信します。



接続が初期化されると、ICA セッションクライアント（Citrix Director クライアントではない）に確認メッセージが表示され、セッションをシャドウする許可がユーザーに求められます。



ユーザーが ［はい］ をクリックすると、ICA セッションがシャドウされていることを示すウィンドウが Citrix Director 側で開きます。

使用方法について詳しくは、Citrix Director のドキュメントを参照してください。

制限事項

- セッションのシャドウは、イントラネットでのみ使用するよう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。外部ネットワークでの Linux VDA セッションのシャドウについて、シトリックスではいかなる責任も負いません。
- セッションのシャドウを有効にすると、ドメイン管理者は ICA セッションのみを表示できますが、書き込みの権限や制御する権限はありません。

- 管理者が Citrix Director から ［シャドウ］をクリックすると、セッションをシャドウする許可をユーザーに求める確認メッセージが表示されます。セッションユーザーが許可を与えた場合にのみ、セッションをシャドウできます。
- 前述の確認メッセージには、20 秒のタイムアウト制限があります。タイムアウトになると、シャドウの要求は失敗します。
- 各 ICA セッションは、1 つの Citrix Director ウィンドウで、1 人の管理者だけがシャドウできます。ICA セッションが管理者 A によってシャドウされていて、その間に管理者 B がシャドウ要求を送信した場合、ユーザーの許可を取得するための確認がユーザーデバイスに再度表示されます。ユーザーが同意すると、管理者 A のシャドウ接続は停止され、管理者 B に対して新しいシャドウ接続が構築されます。同じ管理者によって同じ ICA セッションの別のシャドウ要求が送信された場合も同じです。
- セッションのシャドウを使用するには、Citrix Director 7.16 以降をインストールしてください。
- Citrix Director クライアントは、IP アドレスではなく FQDN を使用して、ターゲットの Linux VDA サーバーに接続します。したがって、Citrix Director クライアントは、Linux VDA サーバーの FQDN を解決できる必要があります。

トラブルシューティング

セッションのシャドウが失敗した場合は、Citrix Director クライアントと Linux VDA の両方でデバッグを実行します。

Citrix Director クライアントの場合

Web ブラウザーの開発ツールを使用して、［コンソール］タブの出力ログを確認します。または、［ネットワーク］タブで ShadowLinuxSession API の応答を確認します。ユーザーの許可を取得するための確認が表示されても接続が確立されない場合は、Linux VDA の FQDN を手動で ping して、Citrix Director が FQDN を解決できることを確認します。`wss://`接続で問題が発生した場合は、証明書を確認してください。

Linux VDA の場合

シャドウ要求に応答して、ユーザーの許可を取得するための確認が表示されることを確認します。表示されない場合は、vda.log ファイルと hdx.log ファイルを調べてください。vda.log ファイルを取得するには、次の操作を実行します。

1. /etc/xdl/ctx-vda.conf ファイルを見つけます。vda.log の構成を有効にするには、次の行のコメントを外します：

   ```
   Log4jConfig=" /etc/xdl/log4j.xml"
   ```

2. /etc/xdl/log4j.xml を開き、com.citrix.dmc の部分を見つけ、次のように「info」を「trace」に変更します：


```
<!-- Broker Agent Plugin - Director VDA plugin Logger -->

 <logger name="com.citrix.dmc">

   <level value="trace"/>

 </logger>
<!--NeedCopy-->
```

3. `service ctxvda restart`コマンドを実行して、`ctxvda`サービスを再起動します。

接続確立中にエラーが発生した場合は、次の操作を実行してください。

1. セッションのシャドウがポートを開くのを止めるファイアウォール制限がないか確認します。
2. SSL シナリオの場合、証明書とキーファイルの名前が正しく指定され、正しいパスに置かれていることを確認します。
3. 新しいシャドウ要求で使用するための十分なポートが、6001～6099 の間に残っていることを確認します。

HTML5 向け Citrix Workspace アプリのサポート

November 11, 2021

HTML5 向け Citrix Workspace アプリのこのリリース以降、クライアントを Citrix Gateway に接続することなく Linux 仮想アプリおよびデスクトップに直接接続できます。HTML5 向け Citrix Workspace アプリについて詳しくは、Citrix ドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトでは無効になっています。有効にするには、次の手順を実行します:

1. Citrix StoreFront で HTML5 向け Citrix Workspace アプリを有効にします。

   詳細な手順については、Knowledge Center 記事CTX208163の手順 1 を参照してください。

2. WebSocket 接続を有効にします。

   a) Citrix Studio で、**WebSockets** 接続ポリシーを ［許可］に設定します。

      他の WebSocket ポリシーを設定することもできます。WebSocket ポリシーの完全な一覧については、「WebSocket のポリシー設定」を参照してください。

   b) VDA で`ctxvda`サービス、`ctxhdx`サービスの順に再起動して設定を有効にします。

c) VDA で次のコマンドを実行して、WebSocket リスナーが動作しているかどうかを確認します。

```
netstat -an | grep 8008
```

WebSocket リスナーが動作している場合、コマンド出力は次のようになります:

```
tcp 0 0 :::8008 :::* LISTEN
```

注: セキュアな WebSocket 接続のために TLS 暗号化を有効にすることもできます。TLS 暗号化を有効にする方法については、「TLS によるユーザーセッションの保護」を参照してください。

Citrix Director を使用した Linux セッションの監視

June 22, 2023

Citrix Director で、Linux セッションの次のメトリックが利用可能です。メトリックを表示するには、Citrix Director で対象セッションを見つけて ［セッションの詳細］パネルを確認します。

- ICA 往復時間

  Linux VDA バージョン 1903 以降、ICA 往復時間のメトリックが利用可能になりました。ICA 往復時間のメトリックを表示するには、Citrix Director 1903 以降を使用し、Citrix Studio で ［**ICA** 往復測定］および［**ICA** 往復測定間隔］ポリシーを作成します。ポリシーの作成については、「Studio でポリシーを作成する」を参照してください。

- Protocol

  Linux VDA バージョン 1909 以降、プロトコル情報を利用できます。Linux セッションのトランスポートプロトコルは、［セッション詳細］パネルに **UDP** または **TCP** として表示されます。

監視サービスデーモン

November 11, 2021

監視サービスデーモンは、定期的に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

VDA を起動すると、監視サービスデーモンが自動的に起動します。

この機能は、管理者権限を使用して **scanningpolicy.conf**、**rulesets.conf**、**whitelist.conf** ファイルで構成することができます。構成ファイルは、**/opt/Citrix/VDA/sbin** にあります。

scanningpolicy.conf、**rulesets.conf**、**whitelist.conf** ファイルへの変更を適用するには、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
<!--NeedCopy-->
```

- **scanningpolicy.conf**

  この構成ファイルでは、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（デフォルト値は true）
  - DetectTime：20（単位：秒、デフォルト値：20、最小値：5）
  - AutoRepair：true/false（デフォルト値は true）
  - MultBalance：false
  - ReportAlarm：false

- **rulesets.conf**

  この構成ファイルでは、監視対象のサービスを指定します。次のスクリーンショットが示すように、デフォルトでは 4 つの監視対象サービスがあります。

```
MonitorUser: all
MonitorType: 3
ProcessName: ctxhdx
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: ctxvda
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: ctxpolicyd
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: Xorg
Operation: 8
DBRecord: false
```


各監視サービスを構成するには、以下のフィールドを指定します。

- MonitorUser：all
- MonitorType：3
- ProcessName：<>（プロセス名は空白にすることはできません。また、完全に一致する必要があります。）
- Operation：1/2/4/8（1 = 例外が検出されるとサービスを停止します。2 = 例外が検出されるとサービスを強制終了します。4 = サービスを再起動します。8 = Xorg プロセスの残りをクリーンアップします。）
- DBRecord：false

- **whitelist.conf**

  rulesets.conf ファイルで指定した監視対象サービスは、**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、セキュリティ上のセカンダリフィルターとなります。

  ホワイトリストを構成するには、**whitelist.conf** ファイルにプロセス名のみを含めます（完全に一致する必要があります）。例として、以下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監視サービスデーモンを停止します。これを実行しない場合、監視サービスデーモンは停止したサービスを再起動します。

TLS によるユーザーセッションの保護

March 1, 2022

バージョン 7.16 では、Linux VDA は、ユーザーセッションのセキュリティ保護のために TLS 暗号化をサポートしています。TLS 暗号化はデフォルトでは無効になっています。

TLS 暗号化を有効にする

ユーザーセッションを保護するために TLS 暗号化を有効にするには、Linux VDA と Delivery Controller（Controller）の両方で証明書を取得し、TLS 暗号化を有効にします。

証明書を取得する

信頼できる認証機関（CA）から PEM 形式のサーバー証明書と CRT 形式のルート証明書を取得します。サーバー証明書には、次のセクションがあります。

- 証明書
- 暗号化されていない秘密キー
- 中間証明書（必須ではありません）

サーバー証明書の例：


```
-----BEGIN CERTIFICATE-----
MIIDTTCCAragAwIBAgIJALluncpiqGXCMA0GCSqGSIb3DQEBBQUAMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
MB4XDTA4MDkzMDEwNTk1M1oXDTI4MDkyNTEwNTk1M1owgYoxCzAJBgNVBAYTAlVL
MRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEUMBIGA1UE
ChMLQ2l0cml4IFRlc3QxGzAZBgNVBAsTElNlcnZlciBDZXJ0aWZpY2F0ZTEgMB4G
A1UEAxMXY2EwMDEtc2MwMDEuY2l0cml0ZS5uZXQwgZ8wDQYJKoZIhvcNAQEBBQAD
gYOAMIGJAoGBALCTTOdxcivbIOLOF66xgO5gkNeIGKVP+37pSKV8B66lWCVzr6p9
t72Fa+9oCcf2x/ue274NXFcg4fqGRDsrEwl3YxM6COyBf7L6psrsCDNnBP1q8TJH
4xoPIXUeaW4MVk/3PVyfhHKs4fz8yy1I4VDnXVHhw+OFQ2Bq3NhwsRhnAgMBAAGj
gdwwgdkwCQYDVR0TBAIwADAdBgNVHQ4EFgQUrLidzYot+CUXSh9xMfp1M+/O8yOw
gZkGA1UdIwSBkTCBjoAU85kN1EPJOcVhcOss1slseDQwGsKha6RpMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
ggkAy8nC8dcB32EwEQYJYIZIAYb4QgEBBAQDAgVgMA0GCSqGSIb3DQEBBQUAA4GB
AD5axBYHwIxJCJzNt2zdXnbp2O0yUToWElBwQe/9cGaP6CpjoxJ7FJa2/8IpaT68
Vel8u1SEYY1GKCGCw93pc7sPKqb8pGBRI5/dygb+geFk1Q7KyVbuOIjOtr3pkxAe
b6CFJtNLudHUrwF6lOrB72zbyz3PiIx+HEwt1jOj8Z4K
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCwkOzncXIr2yNC9BeusYDuYJDXiBilT/t+6UilfAeupVglc6+q
fbe9hWvvaAnH9sf7ntu+DVxXIOH6hkQ7KxMNd2MTOgjsgX+y+qbK7AgzZwT9avEy
R+MaDyF1HmluDFZP9z1cn4RyrOH8/MstSOFQ511R4cPtBUNgatzYcLEYZwIDAQAB
AoGBAKwBgZu/bkl8edgB8YPyU7diiBX89IOs4b/aPjM+JDmjxb8N96RsPO24p9Ea
FtUc9+iL8mEroLUbSicCXjsJFc+cxg9vVaNa6EEkkBj735oCUERqSxOYb/lAdck/
FXzUOtqytUe/KHgcSgjtjrSeqLJqMm+yxzBAatVRTTzGdwAhAkEA3l1KRZjIN5uz
Enmi2RTI3ngBhBP/S3GEbvJfKsD5n2Ri9O+OoEPxclvvp5ne8QOzUpshbjFEPbOC
ykZ6UassFwJBAMtI5yPnV9ewPzJoaNjZIJcMtNXDchSlxXiJiyzv+Qmr8RuQz9Pv
fIenmTrfZ+Wo4DaKg+8ar2OvOnKFOHFAmDECQQDEWRlH6cE3WyCfN1u942M9XkhR
GvSpR7+b///vL6Nwwv3CwPV9n8DTpL+wuDkJZ9nCvRteil9MlaMTYjs3alNvAkEA
qy5JzZcbBnrYzMbVO32jju7ZPISnhTGO1xDjzMSLLpTGpNLN34b0k3sTclr8L42E
uQjtTqRm+wdsrVF3lFazkQJANudmsUVv3gZKhMGaV2hzIdXIfHyOIYv+3leZhQY6
h5eEmxSZS5OTvyNGt2e6m2ZgaZmjTagH59TCBHvR5nof2g==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIDGTCCAoKgAwIBAgIJAMvJwvHXAd9hMA0GCSqGSIb3DQEBBQUAMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
MB4XDTA4MDkzMDEwNDExMVoXDTI4MDkyNTEwNDExMVowZzELMAkGA1UEBhMCVUsx
EjAQBgNVBAgTCUNhbWJyaWRnZTE5MBAGA1UEBxMJQ2FtYm9lcmSlMRQwEgYDVQQK
EwtDaXRyaXggVGVzdDEaMBgGA1UEAxMRY2EwMDEuY2l0cml0ZS5uZXQwgZ8wDQYJ
KoZIhvcNAQEBBQADgYOAMIGJAoGBAKVZmF7Uj7uOnvO3Qwdfi0nr3QkNH2DXpWrZ
Zh8cI9Vv+UFRUiC6oB7izLtBMFn3fOUP7i2CfkHN3ZGJ17p89pdyjket1MslVeJw
acOqrYvD+fNNSvJjunTbaCywVtALjmFSfMHeZJXVSckrpEhnk0nkMS16tcrya/K/
osSlzvI3AgMBAAGjgcwwgckwDAYDVR0TBAUwAwEB/zAdBgNVHQ4EFgQU85kN1EPJ
OcVhcOss1slseDQwGsIwgZkGA1UdIwSBkTCBjoAU85kN1EPJOcVhcOss1slseDQw
GsKha6RpMGcxCzAJBgNVBAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNV
BAcTCUNhbWJvdXJuZTEUMBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNh
MDAxLmNpdHJpdGUubmV0ggkAy8nC8dcB32EwDQYJKoZIhvcNAQEFBQADgYEAIZ4Z
gXLLXf12RNqh/awtSbd41Ugv8BIKAsg5zhNAiTiXbzz8Cl3ec53Fb6nigMWc5Tli
UNCLXwnxRUiD4O0tESLX9ACUNH3I94yxOgujkSOSBni21jjZTvfBB32Rmr5DByJg
UmKORn/hdqMlcqpe5wO6as6+HN4wUOi+hEtUMME=
-----END CERTIFICATE-----
```


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、**enable_vdassl.sh** ツールを使用して、TLS 暗号化を有効または無効にします。このツールは、**/opt/Citrix/VDA/sbin** ディレクトリにあります。このツールで使用できるオプションについては、**/opt/Citrix/VDA/sbin/enable_vdassl.sh -help** コマンドを実行してください。

ヒント：各 Linux VDA サーバーにサーバー証明書をインストールし、各 Linux VDA サーバーとクライアントにルート証明書をインストールする必要があります。

Controller で **TLS** 暗号化を有効にする

> 注：
>
> TLS 暗号化は、デリバリーグループ全体に対してのみ有効にすることができます。特定のアプリケーションに対して TLS 暗号化を有効にすることはできません。

Controller の PowerShell ウィンドウで、次のコマンドを順番に実行して、ターゲットのデリバリーグループの TLS 暗号化を有効にします。

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME'| Set-BrokerAccessPolicyRule -HdxSslEnabled $true`

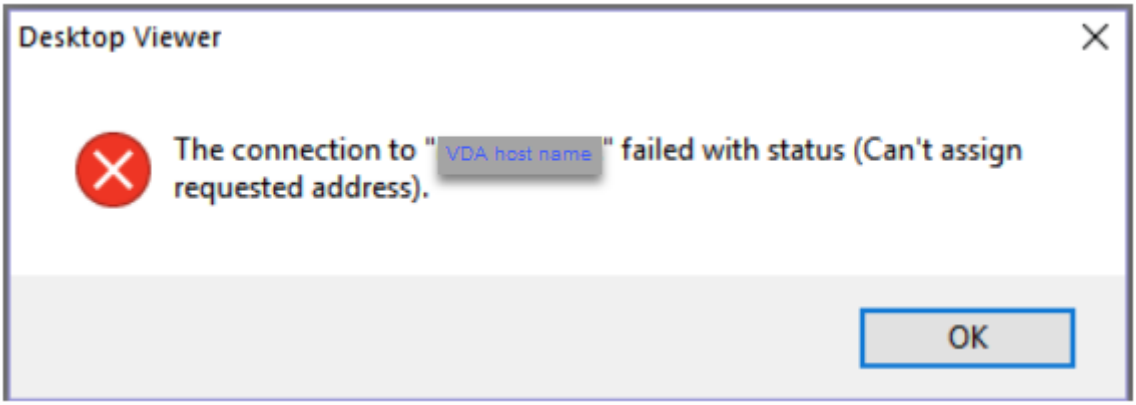
> 注：VDA FQDN のみが ICA セッションファイルに含まれるように、**Set-BrokerSite -DnsResolutionEnabled $true** コマンドを実行することもできます。
>
> このコマンドは、DNS 解決を有効にします。DNS 解決を無効にすると、ICA セッションファイルは VDA の IP アドレスを開示し、SSLProxyHost や UDPDTLSPort などの TLS 関連項目に対してのみ FQDN を提供します。

Controller で TLS 暗号化を無効にするには、次のコマンドを順番に実行します。

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME'| Set-BrokerAccessPolicyRule -HdxSslEnabled $false`
3. `Set-BrokerSite -DnsResolutionEnabled $false`

トラブルシューティング

公開されたデスクトップセッションにアクセスしようとすると、Windows 向け Citrix Workspace アプリで、次の「要求されたアドレスを割り当てることができません」というエラーが発生することがあります：



回避策として、**hosts** ファイルに次のようなエントリを追加します:

```
<IP address of the Linux VDA>      <FQDN of the Linux VDA>
```

Windows マシンでは、**hosts** ファイルは通常、`C:\Windows\System32\drivers\etc\hosts`にあります。

DTLS によるユーザーセッションの保護

August 10, 2021

DTLS 暗号化機能は、7.18 リリースから完全にサポートされます。この機能は Linux VDA ではデフォルトで有効になっています。詳しくは、「Transport Layer Security」を参照してください。

DTLS 暗号化の有効化

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、[**HDX** アダプティブトランスポート] ポリシーが [優先] または [診断モード] に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**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。このツールは **/opt/Citrix/VDA/sbin** にあります。このツールで使用できるオプションについては、**/opt/Citrix/VDA/sbin/enable_vdassl.sh -h** コマンドを実行してください。

> 注:
>
> 現在、Linux VDA は DTLS 1.0 と DTLS 1.2 の両方をサポートしています。DTLS 1.2 には Citrix Receiver

> for Windows 4.12 または Windows 向け Citrix Workspace アプリ 1808 以降が必要です。使用しているクライアントが DTLS 1.0（Citrix Receiver for Windows 4.11 など）のみをサポートしている場合は、**enable_vdassl.sh** ツールを使用して、**SSLMinVersion** を **TLS_1.0** に **SSLCipherSuite** を **COM** または **ALL** に設定します。

スマートカードのサポート

March 11, 2022

Linux 仮想デスクトップセッションにログオンするときに、クライアントデバイスに接続されたスマートカードを認証に使うことができます。この機能は、ICA スマートカード仮想チャネル上でのスマートカードのリダイレクトによって実装されます。セッション内でスマートカードを使用することもできます。使用例としては、ドキュメントにデジタル署名を追加する、電子メールを暗号化または復号化する、スマートカード認証が必要な Web サイトを認証するなどがあります。

Linux VDA は、この機能に Windows VDA と同じ構成を使用します。詳しくは、この記事の「スマートカード環境を構成する」セクションを参照してください。

スマートカードを使用したパススルー認証を行えるかどうかは、次の条件により異なります:

- Linux VDA が RHEL 7.7 にインストールされている。
- CoolKey がサポートするスマートカードが使用されている。
- Windows 向け Citrix Workspace アプリが使用されている。

> 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、公式にはサポートされていません。

RHEL 7.7 に **Linux VDA** ソフトウェアをインストールする

RPM パッケージマネージャーまたは簡単インストールを使用して Linux VDA ソフトウェアをインストールします。「インストールの概要」セクションを参照してください。

VDA のインストールが完了したら、VDA が Delivery Controller に登録でき、公開されている Linux デスクトップセッションがパスワード認証を使用して正常に起動できることを確認します。

CoolKey がスマートカードをサポートしていることの確認

CoolKey は、RHEL 上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 は、CoolKey カード、CAC、PIV、PKCS # 15 という 4 種類のスマートカードをサポートしています。しかし、公式にサポートされ検証されて

いるカードの数はまだ限られています（「Smart Card Support in Red Hat Enterprise Linux」を参照してください）。

この記事では、構成を説明するための例として、YubiKey 4 スマートカードを使用します。YubiKey 4 は、Amazon や他の小売業者から簡単に購入できる一体型の USB CCID PIV デバイスです。CoolKey ドライバーは、YubiKey 4 をサポートしています。

もっと高度なスマートカードが必要になった場合は、RHEL 7.7 と CoolKey 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。CoolKey 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、CoolKey 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 libcoolkeypk11.so --list-slots
<!--NeedCopy-->
```

CoolKey がスマートカードをサポートしている場合、コマンド出力は次のようになり、スロット情報が含まれています。

```
[root@rhphy ~]# pkcs11-tool --module libcoolkeypk11.so --list-slots
Available slots:
Slot 0 (0x1): Yubico Yubikey 4 CCID 00 00
  token label        : user1
  token manufacturer :
  token model        :
  token flags        : login required, token initialized, PIN initialized, readonly
  hardware version   : 0.0
  firmware version   : 0.0
  serial num         :
[root@rhphy ~]#
```


構成

ルート証明書を準備する

ルート証明書は、スマートカード内の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. 通常は CA サーバーから、ルート証明書を PEM 形式で取得します。

   次のようなコマンドを実行して、DER ファイル（*.crt、*.cer、*.der）を PEM に変換できます。次のコマンド例では、**certnew.cer** は DER ファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   <!--NeedCopy-->
   ```

2. ルート証明書を`openssl`ディレクトリにインストールします。例として **certnew.pem** ファイルを使用しています。

   ```
   cp certnew.pem <path where you install the root certificate>
   <!--NeedCopy-->
   ```

   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。

NSS データベースを構成する

Linux VDA ログオンモジュールは、スマートカードと証明書にアクセスするときに NSS データベースに依存しています。次の手順を実行して、NSS データベースを構成します。

1. 前述のルート証明書を NSS データベースに追加します。

   ```
   certutil -A -n "My Corp Root" -t "CT,C,C" -a -d /etc/pki/nssdb -i
       /etc/pki/CA/certs/certnew.pem
   <!--NeedCopy-->
   ```

2. 次のコマンドを実行して、ルート証明書が NSS データベースに正常に追加されたことを確認します。

   ```
   certutil -L -d /etc/pki/nssdb
   <!--NeedCopy-->
   ```

   ルート証明書が正常に追加されている場合、コマンド出力は次のようになります。

   ```
   [root@rh73ws LVDA]# certutil -L -d /etc/pki/nssdb

   Certificate Nickname                                         Trust Attributes
                                                                SSL,S/MIME,JAR/XPI

   My Corp Root                                                 CT,C,C
   ```

3. CoolKey が NSS PKCS#11 ライブラリにインストールされているかどうかを確認します。


```
modutil -list -dbdir /etc/pki/nssdb
<!--NeedCopy-->
```

CoolKey モジュールがインストールされている場合、コマンド出力は次のようになります。

```
[root@rh73demo ~]# modutil -list -dbdir /etc/pki/nssdb

Listing of PKCS #11 Modules
-----------------------------------------------------------
  1. NSS Internal PKCS #11 Module
         slots: 2 slots attached
        status: loaded

         slot: NSS Internal Cryptographic Services
        token: NSS Generic Crypto Services

         slot: NSS User Private Key and Certificate Services
        token: NSS Certificate DB

  2. CoolKey PKCS #11 Module
        library name: libcoolkeypk11.so
         slots: There are no slots attached to this module
        status: loaded
-----------------------------------------------------------
```

CoolKey モジュール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手動でインストールし、再度インストールを確認してください。

```
modutil -add "CoolKey PKCS #11 Module" -libfile libcoolkeypk11.so
    -dbdir /etc/pki/nssdb
<!--NeedCopy-->
```

4. pam_pkcs11 モジュールを構成します。

   pam_pkcs11 モジュールは、ユーザー証明書を検証するときにローカルの VDA 構成に依存します。pam_pkcs11 で使用されるデフォルトのルート証明書は、**/etc/pam_pkcs11/cacerts/**にあります。このパスの各ルート証明書にはハッシュリンクがあります。次のコマンドを実行して、準備されたルート証明書をインストールし、pam_pkcs11 を構成します。

```
yum install pam_pkcs11

mkdir /etc/pam_pkcs11/cacerts/

cp certnew.pem /etc/pam_pkcs11/cacerts/

cacertdir_rehash /etc/pam_pkcs11/cacerts
<!--NeedCopy-->
```


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、手動で構成することができます。

- ctxsmartlogon.sh スクリプトを使用してスマートカード環境を構成する

  > 注:
  >
  > ctxsmartlogon.sh スクリプトは、PKINIT 情報をデフォルトの領域に追加します。この設定は、**/etc/krb5.conf** 構成ファイルを使用して変更できます。

  スマートカードを初めて使用する前に、ctxsmartlogon.sh スクリプトを実行してスマートカード環境を構成します。

  > ヒント:
  >
  > ドメインへの参加に SSSD を使用している場合は、ctxsmartlogon.sh の実行後に SSSD サービスを再起動してください。

```
1  sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
2  <!--NeedCopy-->
```

  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:

```
1  sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
2  <!--NeedCopy-->
```

  結果は次のようになります:


```
#*****************************************************************************
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*****************************************************************************
Step 1:Enable smart card logon.
  Do you want enable smart card logon? (y/n)[y] n
ctxsmartlogon.sh exit.
```

- 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 4 スマートカードに基づく構成について詳しくは、Knowledge Center のCTX206156の記事を参照してください。

  次の手順に進む前に、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がスマートカードにダウンロードされていること、スマートカードを使用して Windows VDA に正常にログオンできることを確認してください。

PC/SC Lite パッケージをインストールする

PCSC Lite は、Linux でのパーソナルコンピューター/スマートカード（PC/SC）仕様の実装です。スマートカードやリーダーと通信するための Windows スマートカードインターフェイスを提供します。Linux VDA でのスマートカードリダイレクトは、PC/SC レベルで実装されています。

次のコマンドを実行して、PC/SC Lite パッケージをインストールします。

```
1 yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
2 <!--NeedCopy-->
```

スマートカードドライバーをインストールする

CoolKey は、RHEL 上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 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。

```
1 yum install coolkey
2 <!--NeedCopy-->
```

スマートカード認証用の **PAM** モジュールをインストールする

次のコマンドを実行して、pam_krb5 および krb5-pkinit モジュールをインストールします。


```
yum install pam_krb5 krb5-pkinit
<!--NeedCopy-->
```

pam_krb5 モジュールもプラグイン可能な認証モジュールであり、PAM 対応アプリケーションがパスワードを確認したり、キー配布センター（KDC）のチケット配布チケットを取得したりするために、このモジュールを使用できます。krb5-pkinit モジュールには PKINIT プラグインが含まれていて、クライアントが秘密キーと証明書を使用して KDC から初期資格情報を取得できるようにします。

pam_krb5 モジュールを構成する

pam_krb5 モジュールは KDC と対話して、スマートカード内の証明書を使用して Kerberos チケットを取得します。PAM で pam_krb5 認証を有効にするには、次のコマンドを実行します:

```
authconfig --enablekrb5 --update
<!--NeedCopy-->
```

/etc/krb5.conf 構成ファイルに、実際の領域に応じた PKINIT 情報を追加します。

> 注:
>
> **pkinit_cert_match** オプションは、クライアント証明書が PKINIT 認証の試行に使用される前に一致する必要がある一致規則を指定します。一致規則の構文は次のとおりです:
>
> `[relation-operator] component-rule …`
>
> `relation-operator`は&&（すべてのコンポーネント規則が一致する必要がある）または||（1 つのコンポーネント規則のみが一致する必要がある）のいずれかを使用できます。

汎用 krb5.conf ファイルの例を次に示します:

```
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

  }

<!--NeedCopy-->
```


構成ファイルは、PKINIT 情報を追加した後、次のようになります。

```
XD.LOCAL = {
 kdc = [redacted]
 auth_to_local = RULE:[1:$1@$0]
 pkinit_anchors = FILE:/etc/pki/CA/certs/certnew.pem
 pkinit_kdc_hostname = SZCXC-DOMAINC.XD.LOCAL
 pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
 pkinit_eku_checking = kpServerAuth
}
```

PAM 認証を構成する

PAM 構成ファイルは、どのモジュールを PAM 認証に使用しているかを示します。pam_krb5 を認証モジュールとして追加するには、**/etc/pam.d/smartcard-auth** ファイルに次の行を追加します:

```
auth [success=done ignore=ignore default=die] pam_krb5.so preauth_options
=X509_user_identity=PKCS11:/usr/lib64/pkcs11/libcoolkeypk11.so
```

SSSD を使用した場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_options=X509_user_identity=PKCS11:/usr/lib64/pkcs11/libcoolkeypk11.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

password    required      pam_pkcs11.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
```

Winbind を使用した場合、変更後の構成ファイルは次のようになります。


```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_options=X509_user_identity=PKCS11:/usr/lib64/pkcs11/libcoolkeypk11.so
auth        sufficient    pam_permit.so
auth        required      pam_deny.so

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_winbind.so
account     [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore]  pam_krb5.so
account     required      pam_permit.so

password    required      pam_pkcs11.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_winbind.so
session     optional      pam_krb5.so
```

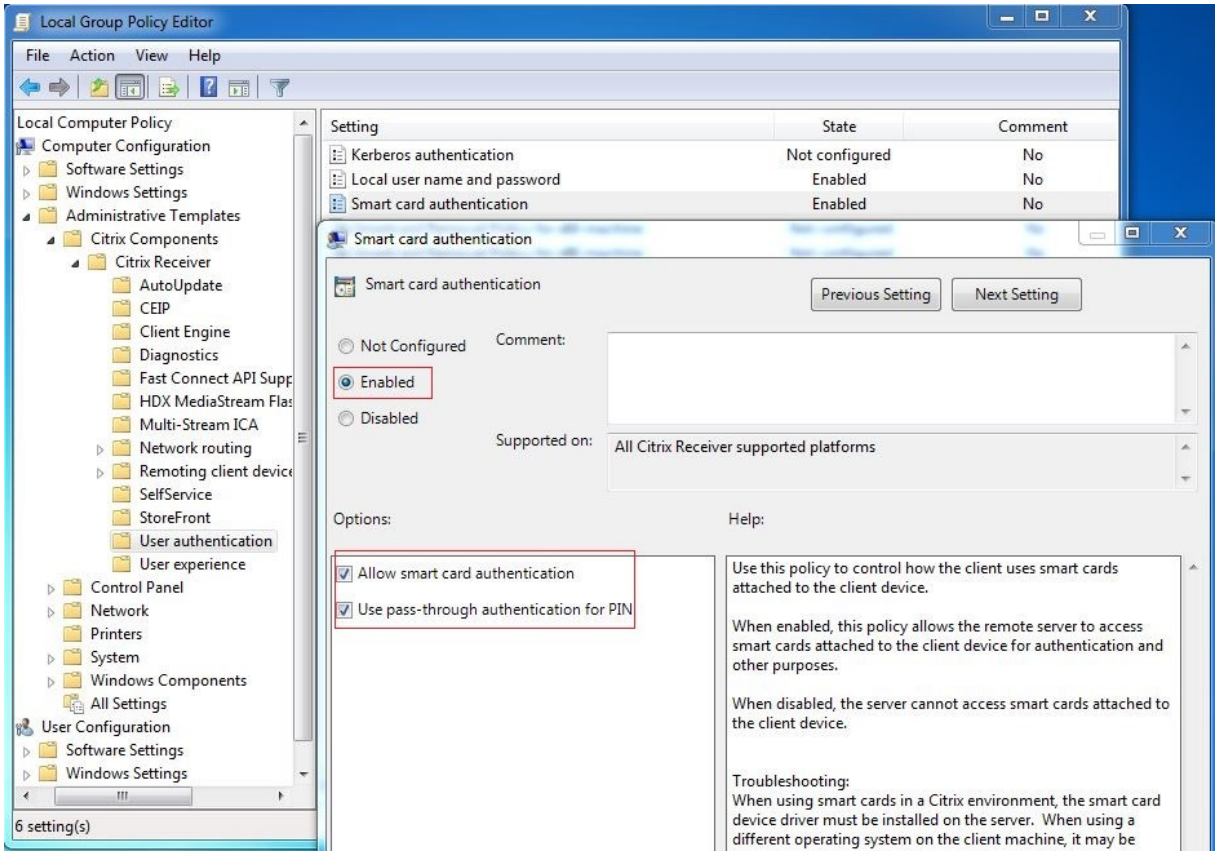
Centrify を使用した場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_options=X509_user_identity=PKCS11:/usr/lib64/pkcs11/libcoolkeypk11.so
auth        sufficient    pam_permit.so
auth        required      pam_deny.so

account     required      pam_nologin.so
account     required      pam_krb5.so
account     required      pam_permit.so

password    required      pam_pkcs11.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
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_krb5.so
```

（オプション）スマートカードを使用したシングルサインオン

シングルサインオン（SSO）とは、仮想デスクトップやアプリケーションの起動時にパススルー認証を実装する Citrix の機能を指します。この機能により、ユーザーが PIN を入力する回数が減ります。Linux VDA で SSO を使用するには、Citrix Workspace アプリを構成します。Windows VDA と同じ構成方法です。詳しくは、Knowledge Center の記事CTX133982を参照してください。

Citrix Workspace アプリでグループポリシーを構成するときは、次のようにスマートカード認証を有効にします。



高速スマートカードログオン

高速スマートカードは、既存の HDX PC/SC ベースのスマートカードリダイレクトの改良版です。遅延が大きい WAN 環境でスマートカードを使用する場合のパフォーマンスが向上しています。詳しくは、「スマートカード」を参照してください。

Linux VDA は、以下のバージョンの Citrix Workspace アプリで高速スマートカードをサポートしています:

- Citrix Receiver for Windows 4.12
- Windows 向け Citrix Workspace アプリ 1808 以降

クライアントで高速スマートカードログオンを有効にする　高速スマートカードログオンは、VDA ではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連する StoreFront サイトの default.ica ファイルに次のパラメーターを追加します:

```
[WFClient]
SmartCardCryptographicRedirection=On
<!--NeedCopy-->
```

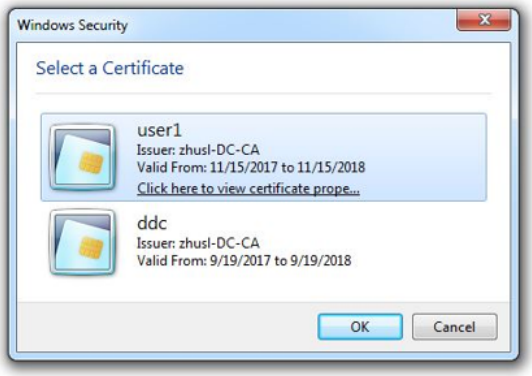

クライアントで高速スマートカードログオンを無効にする　クライアントで高速スマートカードログオンを無効にするには、関連する StoreFront サイトの default.ica ファイルから **SmartCardCryptographicRedirection** パラメーターを削除します。

用途

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、自動的に StoreFront にログオンされます。StoreFront で Linux 仮想デスクトップセッションを開始すると、スマートカード認証のために PIN が Linux VDA に渡されます。
- 非 SSO シナリオでは、StoreFront にログオンするために証明書を選択して PIN を入力するよう求められます。



StoreFront で Linux 仮想デスクトップセッションを開始すると、Linux VDA へのログオンのダイアログボックスが次のように表示されます。ユーザー名はスマートカードの証明書から抽出され、ログオン認証のために PIN をもう一度入力する必要があります。

この動作は Windows VDA と同じです。

スマートカードを使用してセッションに再接続する

セッションに再接続するには、スマートカードがクライアントデバイスに接続されていることを確認します。スマートカードが接続されていないと再認証は失敗するため、Linux VDA 側にグレーのキャッシュウィンドウが表示されてすぐに終了します。この場合、スマートカードの接続を促すメッセージは表示されません。

ただし、StoreFront 側では、セッションに再接続しようとしたときにスマートカードが接続されていないと、StoreFront Web により次のような通知が表示されることがあります。

制限事項

スマートカード取り出し時の動作ポリシー

現在、Linux VDA はスマートカードの削除にデフォルトの動作のみを使用しています。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になり、セッション画面はロックされません。

他のスマートカードおよび **PKCS#11** ライブラリのサポート

サポート一覧に CoolKey スマートカードのみが表示されますが、Citrix は汎用スマートカードリダイレクトによる方法を提供しているため、他のスマートカードおよび PKCS#11 ライブラリの使用を試すこともできます。特定のスマートカードまたは PKCS#11 ライブラリに切り替えるには:

1. PKCS#11 ライブラリのすべての`libcoolkeypk11.so`インスタンスを置き換えます。
2. PKCS#11 ライブラリからレジストリへのパスを設定するには、次のコマンドを実行します:

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
      CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v "
      PKCS11LibPath" -d "PATH"
   <!--NeedCopy-->
   ```

   PATH は PKCS#11 ライブラリ（/usr/lib64/pkcs11/libcoolkeypk11.so など）を参照します。
3. クライアントで高速スマートカードログオンを無効にします。

ダブルホップシングルサインオン認証

November 11, 2021

この機能によって、ユーザー資格情報が入力され、Linux 向け Citrix Workspace アプリおよび Citrix Receiver for Linux 13.10 の AuthManager モジュールの StoreFront ストアにアクセスできます。入力後、ユーザー資格情報

を再度入力することなく、Linux 仮想デスクトップセッションから仮想デスクトップおよびアプリケーションに、クライアントを使用してアクセスできます。

> 注:
>
> この機能は Linux 向け Citrix Workspace アプリおよび Citrix Receiver for Linux 13.10 でサポートされています。

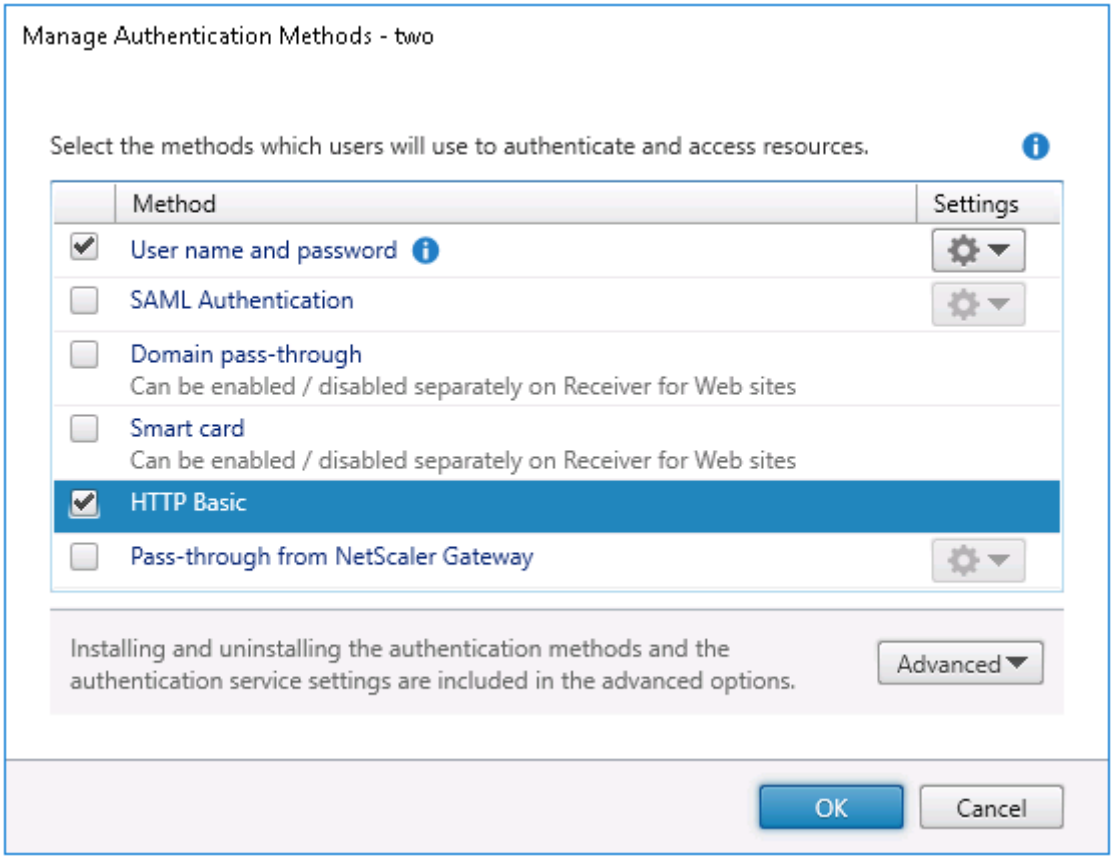
この機能を有効にするには:

1. Linux VDA に、Linux 向け Citrix Workspace アプリまたは Citrix Receiver for Linux 13.10 をインストールします。

   Citrix Workspace アプリまたは Citrix Receiver のCitrix ダウンロードページからアプリをダウンロードします。

   デフォルトのインストールパスは、/opt/Citrix/ICAClient/です。異なるパスにアプリをインストールする場合、ICAROOT 環境変数を実際のインストールパスを参照するよう設定します。

2. Citrix StoreFront 管理コンソールで、対象のストアに **HTTP** 基本認証方式を追加します。



3. HTTP 基本認証を許可するには、次のキーを AuthManager 構成ファイル（$ICAROOT/config/AuthManConfig.xml）に追加します:


```
<Protocols>
     <HTTPBasic>
         <Enabled>True</Enabled>
    </HTTPBasic>
</Protocols>
<!--NeedCopy-->
```

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/
<!--NeedCopy-->
```

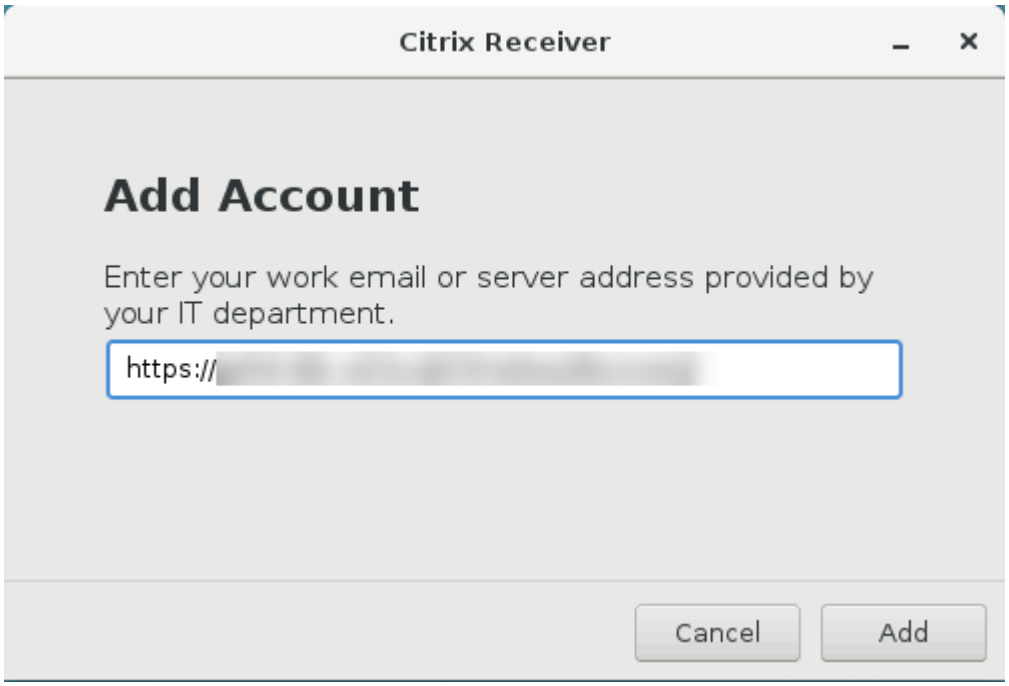
5. 次のコマンドを実行して、この機能を有効にします:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
   CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d "0
   x00000001"
<!--NeedCopy-->
```

6. Linux 仮想デスクトップセッションを開始して、このセッションで Linux 向け Citrix Workspace アプリまたは Citrix Receiver for Linux 13.10 を起動します。

   Linux 仮想デスクトップで Linux 向け Citrix Workspace アプリまたは Citrix Receiver for Linux 13.10 を初めて起動する場合、ストアアカウントの入力を求められます。以降は、指定済みのストアに自動的にログオンします。

   > 注:
   >
   > ストアアカウントの HTTPS URL を入力します。



認証が不要なセッションの構成

September 25, 2023

この記事の情報を使用して、認証が不要なセッションを構成します。Linux VDA をインストールしてこの機能を使用するために特別な設定は一切必要ありません。

> 注:
>
> 認証が不要なセッションを構成する場合は、セッションの事前起動がサポートされないことを考慮してください。セッションの事前起動は、Android 向け Citrix Workspace アプリでもサポートされていません。

認証が不要なストアの作成

Linux VDA で認証が不要なセッションをサポートするには、StoreFront を使用して認証が不要なストアを作成します。

デリバリーグループで認証が不要なユーザーのアクセスを有効にする

認証が不要なストアを作成したら、デリバリーグループで認証が不要なユーザーのアクセスを有効にして、認証が不要なセッションをサポートします。デリバリーグループで認証されていないユーザーを有効にするには、Citrix Virtual Apps and Desktops のドキュメントの指示に従います。

認証が不要なセッションのアイドル時間を設定する

認証が不要なセッションのアイドル状態のタイムアウト値は、デフォルトで 10 分です。この値の設定は、レジストリ設定 **AnonymousUserIdleTime** で行います。**ctxreg** ツールを使ってこの値を変更します。たとえば、このレジストリ設定を 5 分にするには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k  "HKLM\System\
    CurrentControlSet\Control\Citrix"  -v AnonymousUserIdleTime  -d 0
    x00000005
<!--NeedCopy-->
```

認証が不要なユーザーの最大数を設定する

認証されていないユーザーの最大人数を設定するには、レジストリキー **MaxAnonymousUserNumber** を使用します。この設定により、単一の Linux VDA で同時に実行される認証が不要なセッション数が制限されます。このレジストリ設定を構成するには、**ctxreg** ツールを使用します。たとえば、値を 32 に設定するには、次のコマンドを実行します。


```
sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix" -v MaxAnonymousUserNumber -d 0x00000020
<!--NeedCopy-->
```

> 重要:
>
> 認証が不要なセッション数を制限します。同時に起動されるセッション数が非常に多い場合、VDA で使用できるメモリの不足などの問題を引き起こすことがあります。

トラブルシューティング

認証が不要なセッションを構成するときは、次の点を考慮してください。

- 認証が不要なセッションにログオンできませんでした。

レジストリが次を含むように更新されたことを確認します（0 に設定）。

```
sudo /opt/Citrix/VDA/bin/ctxreg read -k "HKLM\System\CurrentControlSet\Control\Citrix" -v MaxAnonymousUserNumber
<!--NeedCopy-->
```

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設定されていることを確認します:

```
ps uax | grep nscd
cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
<!--NeedCopy-->
```

passwd キャッシュ変数が有効になっている場合は、**no** に設定してから、**ncsd** サービスを再起動します。設定の変更後に、Linux VDA の再インストールが必要となる場合があります。

- **KDE** でロック画面のボタンが認証不要のセッション中に表示されます。

デフォルトでは、ロック画面のボタンとメニューは、認証が不要なセッションでは無効になっています。ただし、KDE でなお表示されることがあります。KDE でロック画面のボタンとメニューを特定のユーザーに対して無効にするには、構成ファイル **$Home/.kde/share/config/kdeglobals** に次の行を加えます。例:

```
[KDE Action Restrictions]
action/lock_screen=false
<!--NeedCopy-->
```

ただし、**KDE** アクション制限事項パラメーターがグローバルワイドな **kdeglobals** ファイル（**/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals** など）で不変に設定されている場合、ユーザー設定は効果がありません。

この問題を解決するには、システムワイドな **kdeglobals** ファイルを変更して ［**KDE** アクション制限事項］セクションの **\$i]** タグを削除するか、システムワイドな構成を直接使用して、ロック画面のボタンとメニューを無効にします。KDE 構成について詳しくは、「KDE System Administration/Kiosk/Keys」ページを参照してください。

LDAPS の構成

November 11, 2021

セキュリティで保護された LDAP（LDAPS）によって、Active Directory 管理対象ドメインに SSL（Secure Socket Layer）/TLS（Transport Layer Security）経由のセキュリティ保護された Lightweight Directory Access Protocol の通信を提供できます。

デフォルトで、クライアントとサーバーアプリケーション間の LDAP 通信は暗号化されていません。SSL/TLS を使用した LDAP（LDAPS）で、Linux VDA および LDAP サーバー間の LDAP クエリコンテンツを保護できます。

次の Linux VDA コンポーネントは、LDAPS との依存関係があります。

- ブローカーエージェント：Delivery Controller に Linux VDA を登録
- ポリシーサービス：ポリシー評価

以下は、LDAPS の構成に必要です。

- Active Directory（AD）/LDAP サーバーで LDAPS を有効化
- クライアントで使用するルート CA をエクスポート
- Linux VDA で LDAPS を有効化または無効化
- サードパーティのプラットフォームで LDAPS の構成
- SSSD の構成
- Winbind の構成
- Centrify の構成
- Quest の構成

AD/LDAP サーバーで LDAPS の有効化

Microsoft 証明機関（CA）または非 Microsoft CA のどちらかから適切な形式の証明書をインストールして、SSL 経由の LDAP（LDAPS）を有効にできます。

> ヒント：
>
> SSL/TLS 経由の LDAP（LDAPS）は、ドメインコントローラーで会社のルート CA をインストールすると、自動的に有効になります。

証明書をインストールして、LDAPS 接続を確認する方法について詳しくは、Microsoft Knowledgebase のサポート技術情報で「How to enable LDAP over SSL with a third-party certification authority」を参照してください。

証明機関の階層に複数の層（2 層または 3 層）がある場合、ドメインコントローラーで LDAPS 認証の適切な証明書を自動的に取得できません。

複数の証明機関の階層を使用してドメインコントローラーで LDAPS を有効にする方法について詳しくは、Microsoft TechNet Web サイトで「LDAP over SSL (LDAPS) Certificate」を参照してください。

クライアントで使用するルート証明書（**CA**）の有効化

クライアントは、LDAP サーバーが信頼する CA の証明書を使用する必要があります。クライアントの LDAPS 認証を有効にするには、ルート CA 証明書を信頼済みのキーストアにインポートします。

ルート CA をエクスポートする方法について詳しくは、Microsoft Support Web サイトで「How to export Root Certification Authority Certificate」を参照してください。

Linux VDA マシンで **LDAPS** を有効化または無効化

Linux VDA で LDAPS を有効または無効にするには、（管理者としてログオンして）次のスクリプトを実行します。

このコマンドの構文には次が含まれます。

- 指定されたルート CA 証明書で SSL/TLS 経由で LDAP を有効にします。

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
<!--NeedCopy-->
```

- SSL/TLS を使用せずに LDAP にフォールバックします。

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
<!--NeedCopy-->
```

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には、次が含まれます。

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers

HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy

HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS

HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore
<!--NeedCopy-->
```

サードパーティのプラットフォームで **LDAPS** の構成

Linux VDA コンポーネントに加えて、VDA のさまざまなサードパーティのソフトウェアコンポーネントでは、SSSD、Winbind、Centrify、Quest などのセキュリティで保護された LDAP が必要な場合もあります。以下のセクションでは、LDAPS、STARTTLS または SASL（署名とシール）によるセキュリティで保護された LDAP を構成する方法について説明します。

> ヒント：
>
> これらすべてのソフトウェアコンポーネントで、SSL ポート 636 を使用し、セキュリティで保護された LDAP にすることが望ましいわけではありません。また、ほとんどの場合、LDAPS（ポート 636 での SSL 経由の LDAP）はポート 389 の STARTTLS と共存できません。

SSSD

オプションごとに、ポート 636 またはポート 389 のセキュリティで保護された SSSD LDAP トラフィックを構成します。詳しくは、SSSD LDAP Linux の man ページを参照してください。

Winbind

Winbind LDAP クエリは、ADS メソッドを使用します。Winbind は、ポート 389 で StartTLS メソッドのみをサポートしています。影響を受ける構成ファイルは、**/etc/samba/smb.conf** と **/etc/openldap/ldap.conf**（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
  ```

また、セキュリティで保護された LDAP は、SASL GSSAPI（署名およびシール）で構成されますが、TLS/SSL と共存することはできません。SASL 暗号化を使用するには、**smb.conf** 構成を変更します。

```
ldap ssl = off
ldap ssl ads = no
client ldap sasl wrapping = seal
```

Centrify

Centrify ではポート 636 の LDAPS をサポートしていません。一方、ポート 389 上のセキュリティで保護された暗号化は提供しています。詳しくは、Centrify サイトを参照してください。

Quest

Quest 認証サービスはポート 636 の LDAPS をサポートしませんが、別の方法でポート 389 のセキュリティで保護された暗号化を提供します。

トラブルシューティング

この機能を使用すると、以下の問題が発生することがあります。

- **LDAPS** サービスの可用性

  AD/LDAP サーバーで LDAPS 接続が使用可能であることを確認します。デフォルトでは、このポートは 636 です。

- **LDAPS** を有効にすると **Linux VDA** の登録が失敗する

  LDAP サーバーとポートが正しく構成されていることを確認します。最初にルート CA 証明書をチェックして、AD/LDAP サーバーと一致することを確認します。

- 誤ったレジストリ変更

  LDAPS 関連のキーが誤って **enable_ldaps.sh** を使用せずに更新されると、LDAPS コンポーネントの依存関係を損なう可能性があります。

- **LDAP** トラフィックは、**Wireshark** やその他のネットワーク監視ツールから **SSL/TLS** で暗号化されません

  デフォルトでは、LDAPS は無効になっています。それを強制するには、**/opt/Citrix/VDA/sbin/enable_ldaps.sh** を実行します。

- **Wireshark** やその他のネットワーク監視ツールからの **LDAPS** トラフィックが存在しない

  LDAP/LDAPS トラフィックは、Linux VDA の登録やグループポリシーの評価を行う際に発生します。

- **AD** サーバーで **LDP** 接続を実行して **LDAPS** の可用性を確認できない

  IP アドレスの代わりに、AD FQDN を使用します。

- **/opt/Citrix/VDA/sbin/enable_ldaps.sh** スクリプトを実行してルート **CA** 証明書をインポートできない

  CA 証明書のフルパスを指定して、ルート CA 証明書の種類が正しいことを確認します。通常は、サポートされている Java Keytool の種類の大半で対応しています。サポート一覧にない場合は、最初に種類を変更してください。証明書の形式の問題が発生した場合は、Citrix では Base64 で暗号化された PEM 形式の使用をお勧めします。

- ルート **CA** 証明書を **Keytool** 一覧に表示できない

  /opt/Citrix/VDA/sbin/enable_ldaps.sh を実行して LDAPS を有効にすると、証明書が/etc/xdl/.keystore にインポートされ、キーストアを保護するパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再度実行して新しいキーストアを作成します。

Xauthority の構成

November 11, 2021

Linux VDA は、対話型のリモート制御で X11 ディスプレイ機能（`xterm`と`gvim`を含む）を使用する環境をサポートしています。この機能は、XClient と XServer 間のセキュリティで保護された通信を確保するために必要なセキュリティメカニズムを提供します。

このセキュリティで保護された通信の権限を保護するには、以下の 2 つの方法があります。

- **Xhost**。デフォルトでは、Xhost コマンドはローカルホスト XClient と XServer の通信のみを許可します。リモート XClient の XServer へのアクセスを許可すると、特定のマシンで権限を付与するために Xhost コマンドが実行される必要があります。あるいは、**xhost +** を使用して XClient に XServer への接続を許可することもできます。
- **Xauthority**。`.Xauthority`ファイルは、各ユーザーのホームディレクトリにあります。このファイルは、XServer の認証の際に xauth が使用する Cookie に資格情報を保存するために使用されます。XServer インスタンス（Xorg）が起動されるときに、特定のディスプレイへの接続を認証するためにこの Cookie が使用されます。

機能

Xorg が起動されると、`.Xauthority`ファイルは Xorg に渡されます。この`.Xauthority`ファイルには次の要素が含まれます:

- 表示番号
- リモート要求プロトコル
- Cookie 番号

xauth コマンドを使用して、このファイルを参照できます。例:

```
# xauth -f ~/.Xauthority

# > list

# > us01msip06:107  MIT-MAGIC-COOKIE-1
      fb228d1b695729242616c5908f11624b
<!--NeedCopy-->
```

XClient がリモートで Xorg に接続する場合、2 つの前提条件を満たす必要があります:

- **DISPLAY** 環境変数をリモート XServer に設定します。
- Xorg で Cookie 番号の 1 つを含む`.Xauthority`を取得します。

Xauthority の構成

リモート X11 ディスプレイ用に Linux VDA 上で Xauthority を有効にするには、次の 2 個のレジストリキーを作成する必要があります:


```
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
<!--NeedCopy-->
```

Xauthority に有効にしてから、手動によるか共有ホームディレクトリをマウントすることで `.Xauthority` ファイルを XClient に渡します。

- `.Xauthority` ファイルを XClient に手動で渡す

  ICA セッションを起動した後、Linux VDA は XClient の `.Xauthority` ファイルを生成し、ログオンユーザーのホームディレクトリにファイルを保存します。この `.Xauthority` ファイルをリモート XClient マシンにコピーし、DISPLAY および XAUTHORITY 環境変数を設定できます。DISPLAY は `.Xauthority` ファイルに保存した表示番号で、XAUTHORITY は Xauthority のファイルパスです。たとえば、次のコマンドを表示します:

  ```
  export DISPLAY={
  Display number stored in the Xauthority file }


  export XAUTHORITY={
  the file path of .Xauthority }

  <!--NeedCopy-->
  ```

  > 注:
  >
  > XAUTHORITY 環境変数が設定されていない場合、`~/.Xauthority` ファイルがデフォルトで使用されます。

- 共有ホームディレクトリをマウントすることにより `.Xauthority` ファイルを XClient に渡す

  簡単な方法は、ログオンユーザーの共有ホームディレクトリをマウントすることです。Linux VDA が ICA セッションを起動すると、ログオンユーザーのホームディレクトリで `.Xauthority` ファイルが作成されます。このホームディレクトリが XClient と共有される場合、ユーザーがこの `.Xauthority` ファイルを手動で XClient に転送する必要はありません。DISPLAY および XAUTHORITY 環境変数を正しく設定した後、XServer で GUI が自動的に表示されます。

トラブルシューティング

Xauthority が機能しない場合は、次のトラブルシューティング手順に従ってください:

1. root 特権を持つ管理者として、すべての Xorg Cookie を取得します:


```
ps aux | grep -i xorg
<!--NeedCopy-->
```

このコマンドは、起動中 Xorg に渡される Xorg プロセスとパラメーターを表示します。もう 1 つのパラメーターは、どの `.Xauthority` ファイルが使用されるかを表示します。例:

```
/var/xdl/xauth/.Xauthority110
<!--NeedCopy-->
```

Xauth コマンドを使用して、Cookie を表示します:

```
Xauth -f /var/xdl/xauth/.Xauthority110
<!--NeedCopy-->
```

2. **Xauth** コマンドを使用して、`~/.Xauthority` に含まれる Cookie を表示します。同じ表示番号の場合、表示される Cookie は Xorg および XClient の `.Xauthority` ファイルで同じである必要があります。
3. Cookie が同じあれば、リモートディスプレイポートが Linux VDA の IP アドレス（例：10.158.11.11）と公開デスクトップの表示番号（例：160）を使用してアクセスできるかを確認します。

   XClient マシンで次のコマンドを実行します。

```
telnet 10.158.11.11 6160
<!--NeedCopy-->
```

   ポート番号は、6000 + 表示番号の合計です。

   telnet の操作が失敗すると、ファイアウォールが要求をブロックすることがあります。

フェデレーション認証サービス

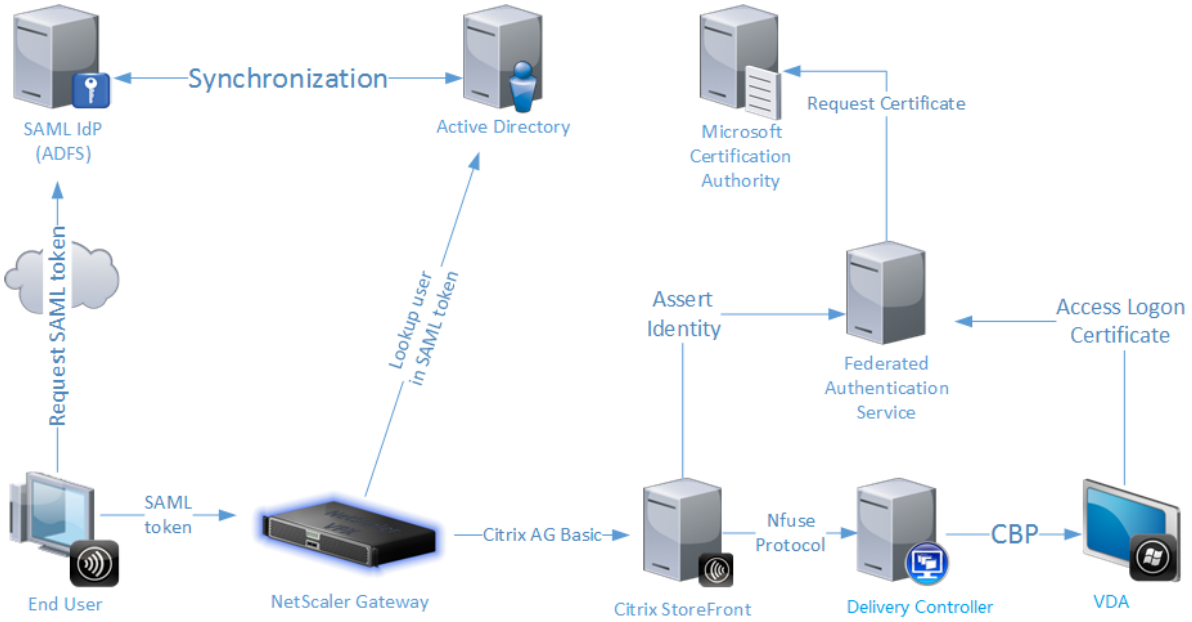
May 15, 2023

概要

Citrix フェデレーション認証サービス（Federated Authentication Service：FAS）は、Active Directory 証明書サービスと統合するように設計されている、権限が付与されたコンポーネントです。Citrix Federated Authentication Service ではユーザー向けの証明書が動的に発行され、ユーザーはスマートカードを持っている場合と同じように Active Directory 環境にログオンできます。この機能により、セキュリティアサーションマークアップランゲージ（Security Assertion Markup Language：SAML）アサーションなどの広範な認証オプションを StoreFront で使用できます。SAML は、インターネット上で従来の Windows ユーザーアカウントに代わるものとして一般的に使用されています。

> 注:
>
> SAML 認証を使用するには、VDA で FAS を適切に構成します。
>
> CU3 以降、Linux VDA は短い接続を使用して FAS サーバーとデータを送信します。

以下の図に、Microsoft 証明機関と統合された FAS による、StoreFront と VDA へのサポートサービスの提供について示します。



ユーザーが Citrix 環境へのアクセスを要求すると、信頼済みの StoreFront サーバーが FAS にアクセスします。FAS は、単一の Citrix Virtual Apps または Citrix Virtual Desktops セッションがそのセッションの証明書で認証できるようにするチケットを付与します。VDA でユーザーを認証する必要がある場合、VDA は FAS にアクセスしてチケットを使用します。FAS のみがユーザー証明書の秘密キーにアクセスできます。VDA は、証明書を使用して実行する必要のあるすべての署名処理および暗号化解除処理を、FAS に送信する必要があります。

要件

FAS は、Windows Server 2008 R2 以降でサポートされています。

- FAS は、ほかの Citrix コンポーネントを含まないサーバーにインストールすることをお勧めします。
- Windows サーバーは、ドメインユーザーに対して自動的に証明書を発行するために登録機関の証明書および秘密キーにセキュアにアクセスする必要があります。また、これらのユーザー証明書および秘密キーにセキュアにアクセスする必要があります。

Citrix Virtual Apps サイトまたは Citrix Virtual Desktops サイトの要件:

- Delivery Controller のバージョンは 7.9 以上である必要があります。

- StoreFront サーバーのバージョンは 3.6（XenApp および XenDesktop 7.9 ISO で提供されるバージョン）以上である必要があります。
- Linux VDA のバージョンは 7.18 以上である必要があります。マシンカタログを通常どおりに作成する前に、フェデレーション認証サービスのグループポリシー構成が適切に VDA に適用されていることを検証してください。詳しくは、この記事の「グループポリシーの構成」セクションを参照してください。

参照先ドキュメント：

- Active Directory 証明書サービス
  https://social.technet.microsoft.com/wiki/contents/articles/1137.active-directory-certificate-services-ad-cs-introduction.aspx
- 証明書ログオン用の Windows の構成
  http://support.citrix.com/article/CTX206156
- フェデレーション認証サービスのインストール
  フェデレーション認証サービス

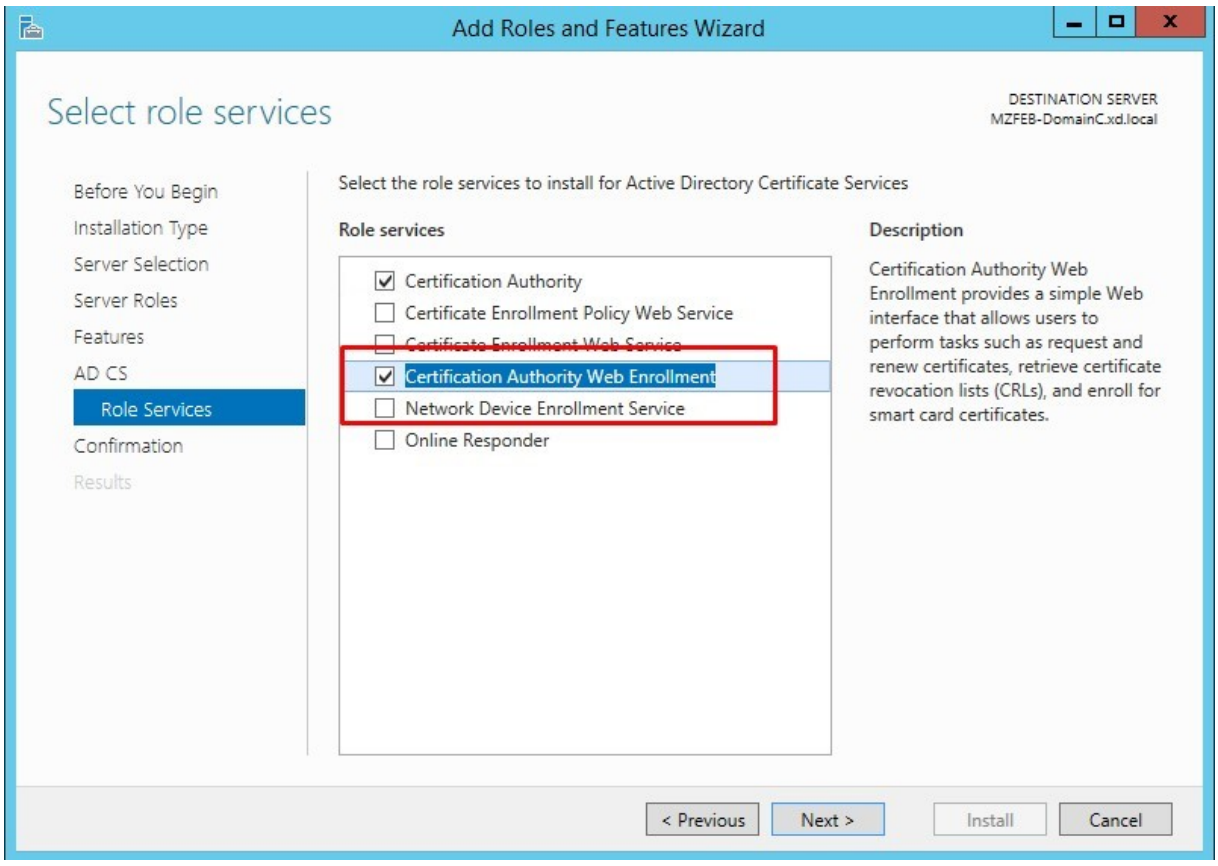
証明書ログオン用の **Windows** の構成

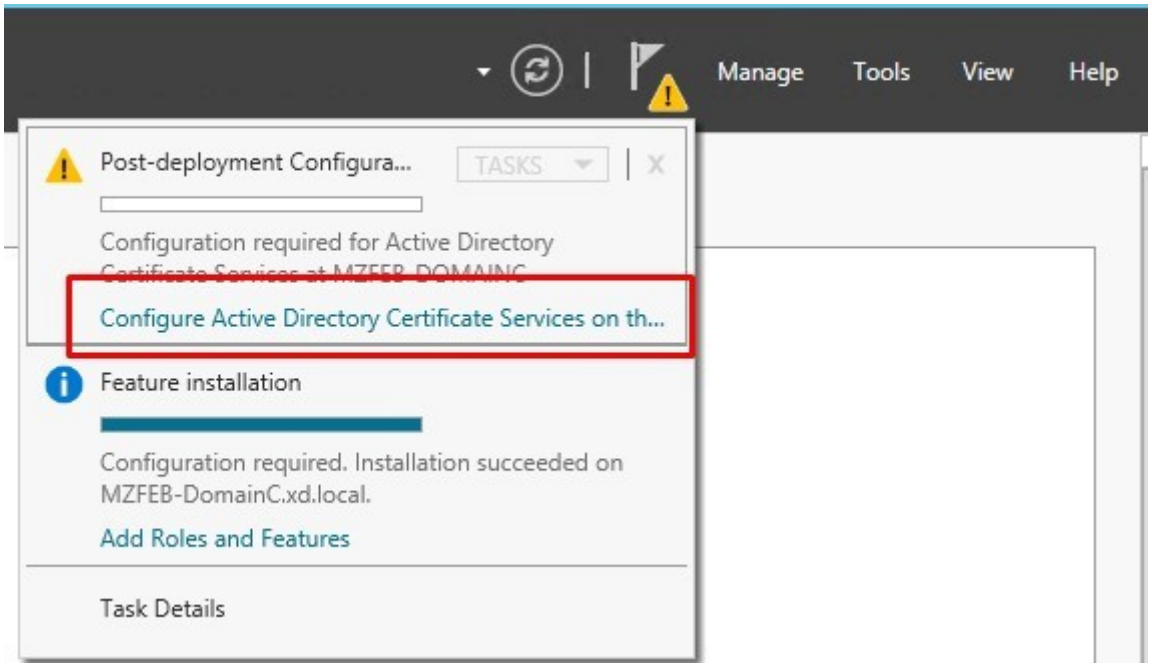
証明書ログオン用に Windows を構成する方法については、Knowledge Center のCTX206156の記事を開き、**Smart_card_config_Citrix_Env.pdf** ファイル（以下「PDF ファイル」）をダウンロードして参照してください。PDF ファイルに従って、各手順で示されている相違点や補足を確認しながら、以下の手順を実行します。操作しているターゲットマシン（AD、Delivery Controller、StoreFront など）には特に注意してください。

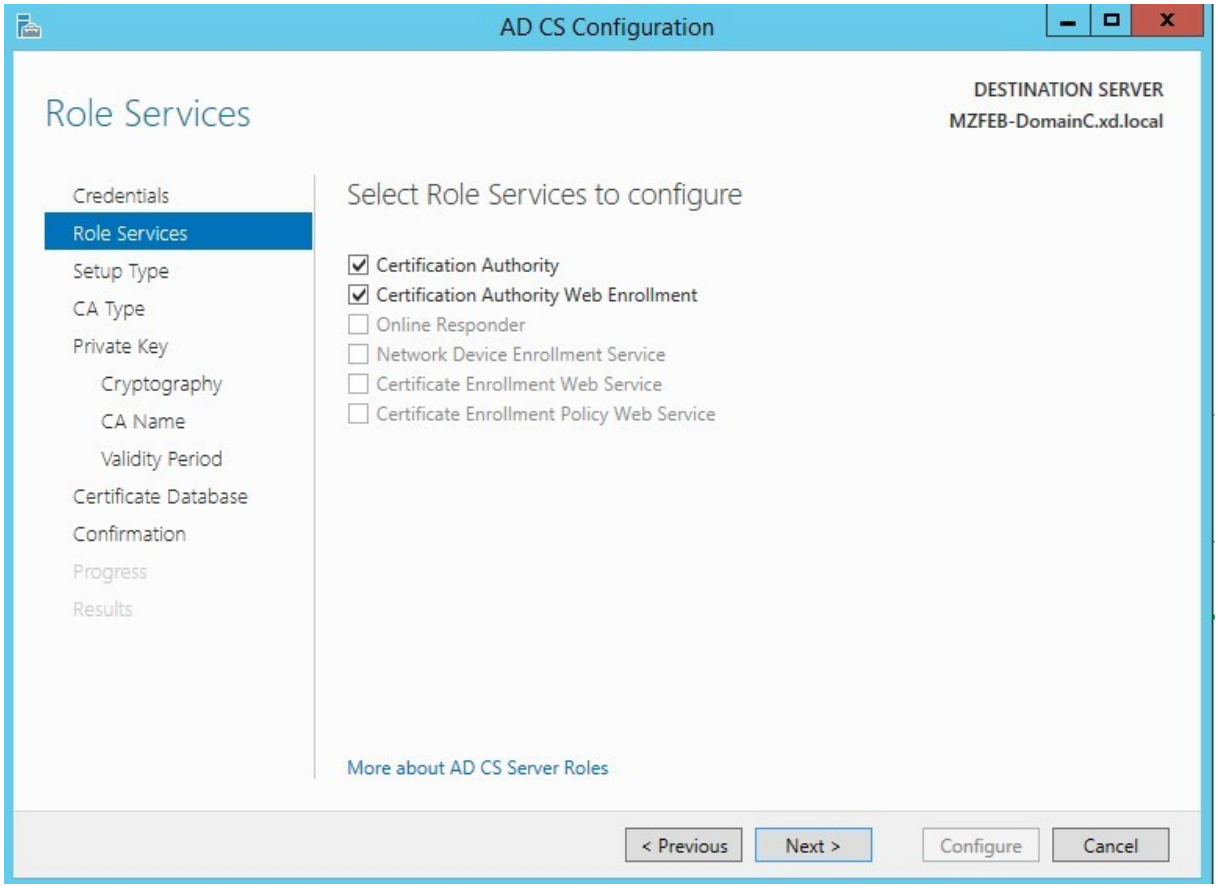
Windows ドメインのセットアップ（**AD** で）

ドメインコントローラーの役割のインストール　PDF ファイルの「ドメインコントローラーの役割のインストール」セクションを参照してください。

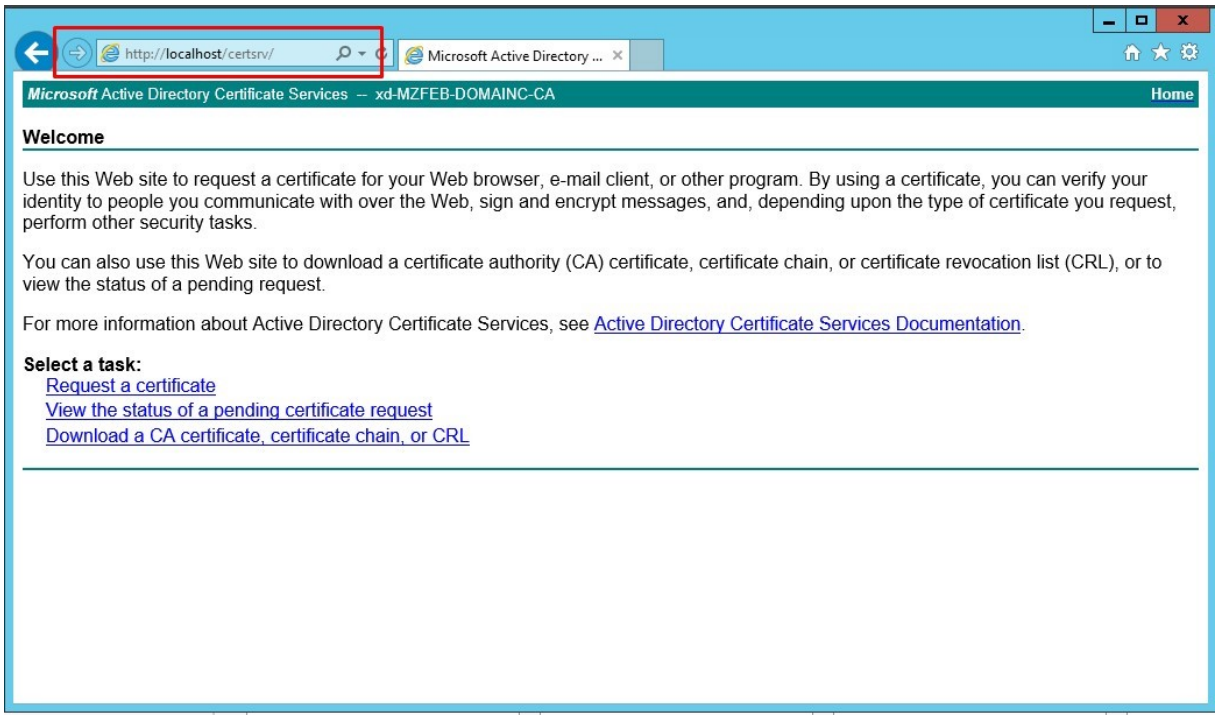
Active Directory 証明書サービスのインストール中に、以下のオプションが選択されていることを確認します：







`http://localhost/certsrv/`を開いて、次のような Welcome ページが表示されているかどうかを確認します。表示されている場合は、Active Directory 証明書サービスは正常にインストールされています。



スマートカードを使用するための認証機関の準備　補足なし。PDF ファイルの「スマートカードを使用するための認証機関の準備」セクションを参照してください。

ドメインコントローラー証明書の発行　補足なし。PDF ファイルの「ドメインコントローラー証明書の発行」セクションを参照してください。

HTTPS 用の **Microsoft IIS** の構成（**StoreFront** で）

Microsoft IIS での **HTTPS** の構成　補足なし。PDF ファイルの「**Microsoft IIS** での **HTTPS** の構成」セクションを参照してください。

ドメインに参加していないコンピューター

PDF ファイルの「ドメインに参加していないコンピューター」セクションを参照してください。

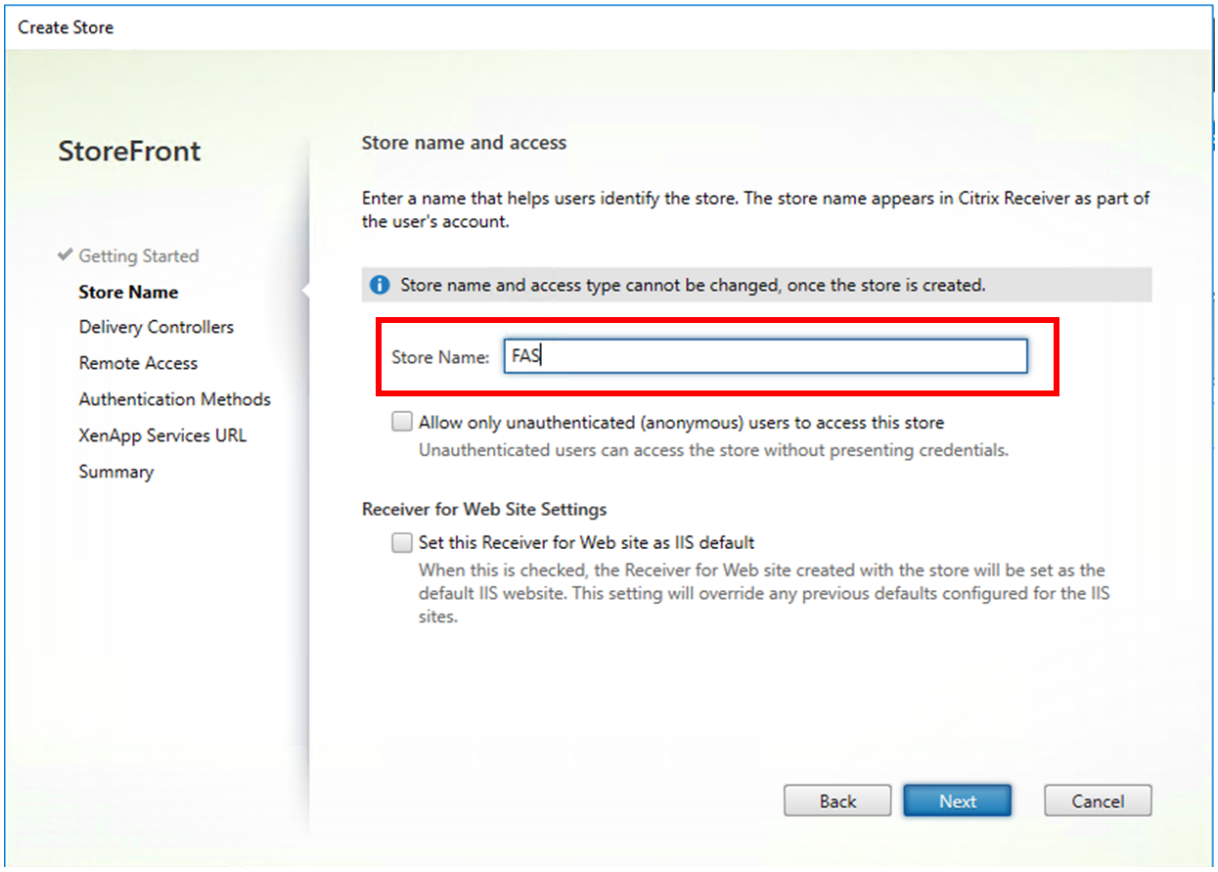
Microsoft CA からの **CA** 証明書の取得（**AD** で）　補足なし。PDF ファイルの「**Microsoft CA** からの **CA** 証明書の取得」セクションを参照してください。

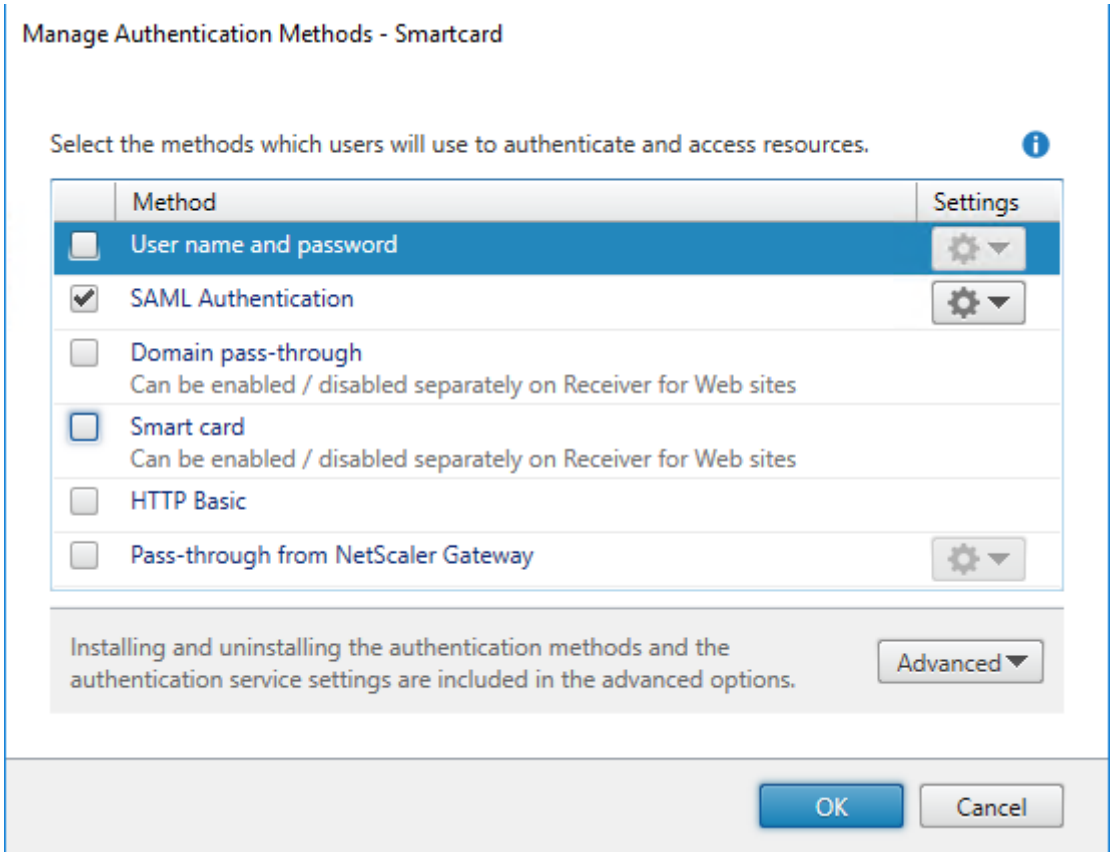
信頼できる **CA** 証明書の **Windows** へのインストール　補足なし。PDF ファイルの「信頼できる **CA** 証明書の **Windows** へのインストール」セクションを参照してください。

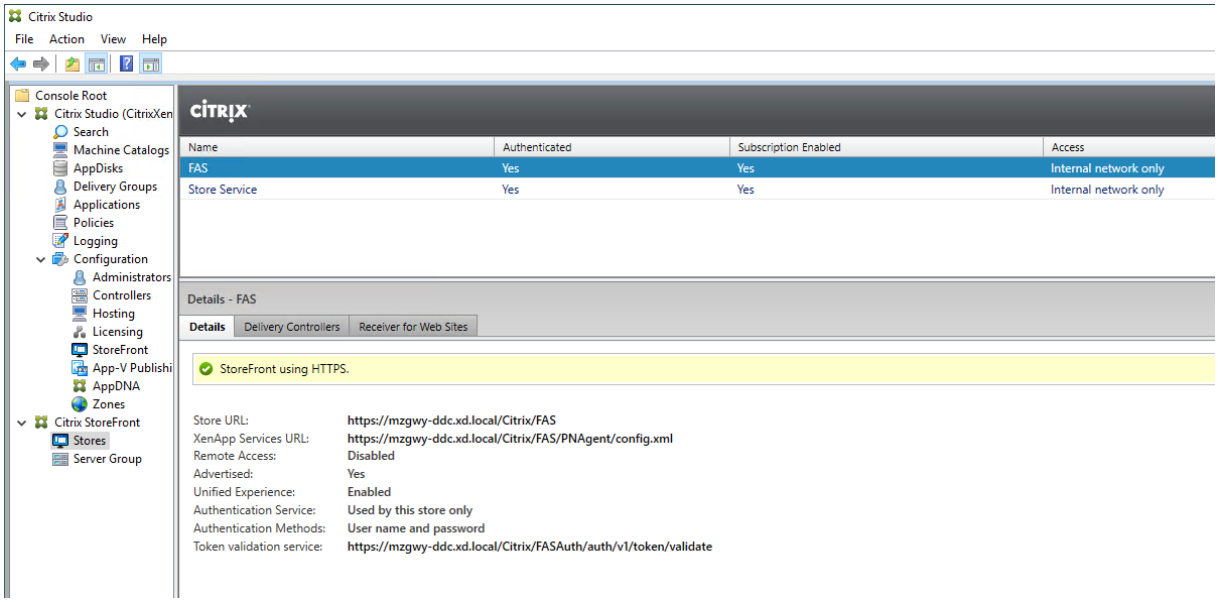
Citrix StoreFront の構成（**StoreFront** で）

ストアの作成　PDF ファイルの「ストアの作成」セクションを参照してください。

前述の IIS 構成の後、共通ストアのベース URL は`http://`ではなく強制的に`https://`に設定されます。FAS はスマートカードとストアを共有しないため、FAS には新しいストアが必要です。Linux VDA FAS は、StoreFront のすべての認証方法と互換性があります。たとえば、FAS ストアはパスワードや SAML を使用するように構成できますが、その両方を同時に使用することはできません。SAML を選択すると、StoreFront の URL は自動的に IdP にリダイレクトされ、パスワード認証方法は無視されます。







Internet Explorer を起動し、FAS ストアの URL を開きます（たとえば、`https://mzgwy-ddc.xd.local/Citrix/FASWeb`）。

注：FAS ストアの URL に **Web** が付加されている必要があります。

FAS のインストールとセットアップ

インストールとセットアップのプロセスは、次の手順で構成されています。

1. フェデレーション認証サービスのインストール
2. StoreFront サーバーでのフェデレーション認証サービスプラグインの有効化
3. グループポリシーの構成
4. フェデレーション認証サービスの管理コンソールを使用した作業：（a）提供されたテンプレートの展開、（b）証明機関のセットアップ、（c）フェデレーション認証サービスへの証明機関の使用権限の付与
5. ユーザー規則の構成

各手順については、「フェデレーション認証サービス」を参照してください。各手順で、次の相違点または補足を確認してください。操作しているターゲットマシン（AD、Delivery Controller、StoreFront、FAS サーバーなど）には特に注意してください。

フェデレーション認証サービスのインストール（**FAS** サーバーで）

セキュリティ上の理由により、FAS は、ドメインコントローラーや証明機関と同様にセキュリティ保護されている専用サーバーにインストールします。

StoreFront ストアでのフェデレーション認証サービスプラグインの有効化（**StoreFront** で）

以下のコマンドでは、StoreFront を構成するときに入力したのと同じ FAS ストア名を使用してください。たとえば、この例では FAS がストア名です。

$StoreVirtualPath = “/Citrix/**FAS**”

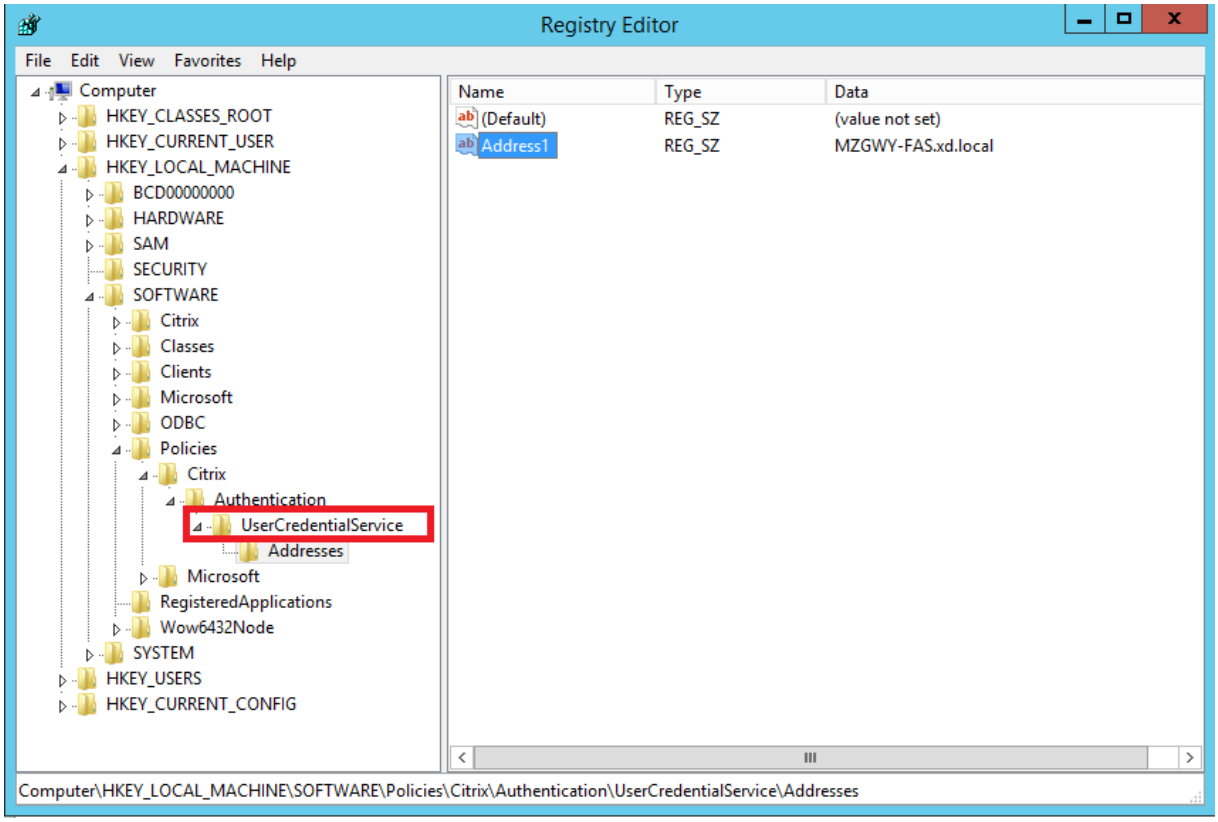
Delivery Controller の構成（**Delivery Controller** で）

フェデレーション認証サービスを使用するには、それに接続可能な StoreFront サーバーを信頼するように Delivery Controller を構成します。PowerShell コマンドレット **Set-BrokerSite -TrustRequestsSentToTheXmlServicePort $true** を実行します。場合によっては、最初に`Add-PSSnapin citrix.*`を実行する必要があります。

グループポリシーの構成（**FAS** サーバーと **AD** で）

このセクションの手順 1〜7 は、管理者のみが実行できます。手順 1 は FAS サーバー上で実行する必要があり、手順 2〜7 は AD 上で実行する必要があります。

手順 1〜7 を完了したら、FAS サーバーのレジストリエディターで、FAS ポリシーが設定されていることを確認します。



セッション中の証明書サポートの有効化　Linux VDA は、セッション内証明書をサポートしていません。

フェデレーション認証サービス管理コンソールの使用（**FAS** サーバーで）

補足なし。「フェデレーション認証サービス」の記事を参照してください。

証明書テンプレートの展開（**FAS** サーバーで）

補足なし。「フェデレーション認証サービス」の記事を参照してください。

Active Directory 証明書サービスのセットアップ（**FAS** サーバーで）

補足なし。「フェデレーション認証サービス」の記事を参照してください。

フェデレーション認証サービスへの権限付与（**FAS** サーバーで）

補足なし。「フェデレーション認証サービス」の記事を参照してください。

ユーザールールの構成（**FAS** サーバーで）

補足なし。「フェデレーション認証サービス」の記事を参照してください。

詳しくは、「フェデレーション認証サービス」の記事の「セキュリティに関する考慮事項」セクションに記載されている「委任された登録エージェント」と「アクセス制御リストの構成」も参照してください。

フェデレーション認証サービスの **ADFS** の展開

フェデレーション認証サービス用に ADFS IdP を展開する方法については、「フェデレーション認証サービスの ADFS の展開」を参照してください。

Linux VDA の構成

FAS サーバーの設定

Linux VDA を新規にインストールする場合、FAS を使用するには、ctxinstall.sh または ctxsetup.sh の実行中に CTX_XDL_FAS_LIST を求められた際に、各 FAS サーバーの完全修飾ドメイン名を入力します。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で区切られた FAS サーバーの一覧を使用できます。いずれかのサーバーアドレスが削除されている場合は、その空白を **<none>** という文字列で埋めて、サーバーアドレスのシーケンスを変更せずに維持します。

インストール済みの Linux VDA をアップグレードする場合は、ctxsetup.sh を再実行することで 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ができます。

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

service ctxvda restart
<!--NeedCopy-->
```

`ctxreg`を使用して FAS サーバーを更新するには、次のコマンドを実行します:

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

service ctxvda restart
<!--NeedCopy-->
```


ルート **CA** 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 証明書を VDA にインストールします。前述の「**Microsoft CA** からの **CA** 証明書の取得（**AD** で）」の手順で AD ルート証明書を取得するか、またはルート CA サーバー（`http://CA-SERVER/certsrv`）から証明書の DER 形式をダウンロードします。

> 注:
>
> 次のコマンドは、中間証明書の構成にも適用されます。

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
<!--NeedCopy-->
```

続いて、次のようなコマンドを実行して、ルート CA 証明書を openssl ディレクトリにインストールします:

```
sudo cp root.pem /etc/pki/CA/certs/
<!--NeedCopy-->
```

> 注:
>
> ルート CA 証明書を **/root** パス下に置かないでください。置いてしまうと、FAS はルート CA 証明書の読み取り権限を持ちません。

FAS の構成

次のスクリプトを実行して FAS パラメーターを設定します:

```
sudo /opt/Citrix/VDA/sbin/ctxfascfg.sh
<!--NeedCopy-->
```

> 注:
>
> 上記のスクリプトは、単一のルート CA 証明書を使用するシナリオのみを処理します。
>
> 環境に中間証明書がある場合は、次のように/etc/krb5.conf に中間パスを追加します:
>
> [realms]
> EXAMPLE.COM = {
> ...
> pkinit_anchors = FILE:/etc/pki/CA/certs/root.pem
> pkinit_pool = FILE:/etc/pki/CA/certs/intermediate.pem
> ...
> }

`ctxfascfg.sh`をサイレントモードで実行できるように、2 つの環境変数が追加されます:

- **CTX_FAS_ADINTEGRATIONWAY=winbind | sssd | centrify** –Active Directory の統合方式を指定。CTX_EASYINSTALL_ADINTEGRATIONWAYが指定されている場合、CTX_EASYINSTALL_ADINTEGRATIO と同じ値です。CTX_EASYINSTALL_ADINTEGRATIONWAYが指定されていない場合、CTX_FAS_ADINTEGRATIONWAYは独自の値を使用します。
- **CTX_FAS_ROOT_CA_PATH=<root_CA_certificate>** –ルート CA 証明書のフルパスを指定。ここで、root_CA_certificate はルート CA 証明書名です。

正しい Active Directory 統合方法を選択し、ルート CA 証明書の正しいパスを入力します（例: /etc/pki/CA/certs/root.pem）。

次に、このスクリプトは krb5-pkinit パッケージと pam_krb5 パッケージをインストールし、関連する構成ファイルを設定します。

制限事項

- FAS でサポートされているプラットフォームと AD の統合方法は限られています。次のマトリックスを参照してください:

	Winbind	SSSD	Centrify
RHEL 7.7/CentOS 7.7	√	√	√
Ubuntu 18.04	√	×	√
Ubuntu 16.04	√	×	√
SLES 12.3	√	×	√

- 現在、FAS はロック画面をサポートしていません。セッションでロックボタンをクリックすると、FAS を使用してセッションに再度ログオンすることはできません。
- このリリースでは、「フェデレーション認証サービスのアーキテクチャの概要」の記事で説明している一般的な FAS 環境のみがサポートされており、**Windows 10 Azure AD Join** は含まれません。

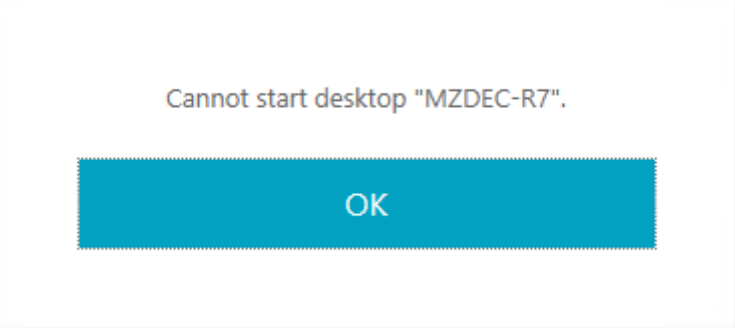
トラブルシューティング

パスワード認証を使用して FAS 以外のセッションを共通ストアで正常に起動できるよう、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され、構成されていることを確認してください。

FAS 以外のセッションが適切に機能している場合は、**Login** クラスの HDX ログレベルを VERBOSE に設定し、VDA ログレベルを TRACE に設定します。Linux VDA のトレースログを有効にする方法については、Knowledge Center のCTX220130の記事を参照してください。

FAS サーバー構成エラー

FAS ストアからセッションを起動すると失敗します。例として、以下のスクリーンショットを参照してください:



/var/log/xdl/hdx.log を確認し、次のようなエラーログを探します:

```
2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: query2fas: failed to retrieve data: No such file or directory.

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: sayhello2fas_internal: Failed to query.

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: sayhello2fas_convertcredential: exit.

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: LoginFasValidate: Failed to start FAS.

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: receive_data: LoginFASValidate - parameters check error.

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: receive_data: Exit FAILURE

2018-03-27 10:17:56.722 <P10122:S2> citrix-ctxlogin: main: EXITING login process..., FAILURE
<!--NeedCopy-->
```

解決策　次のコマンドを実行して、Citrix レジストリ値「HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent\A
が <Your-FAS-Server-List> に設定されていることを確認します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
<!--NeedCopy-->
```

既存の設定が間違っている場合は、前述の「FAS サーバーの設定」の手順に従って再設定します。

間違っているルート **CA** 証明書の構成

FAS ストアからセッションを起動すると失敗します。灰色のウィンドウが表示され、数秒後に消えます。

/var/log/xdl/hdx.log を確認し、次のようなエラーログを探します:

```
2018-03-27 10:15:52.227 <P9099:S3> citrix-ctxlogin: validate_user: pam_authenticate err,can retry for user user1@CTXFAS.LAB

2018-03-27 10:15:52.228 <P9099:S3> citrix-ctxlogin: logout_user: closing session and pam transaction

2018-03-27 10:15:52.228 <P9099:S3> citrix-ctxlogin: validate_user: Exit (user=user1@CTXFAS.LAB)=INVALID_PASSWORD

2018-03-27 10:15:52.228 <P9099:S3> citrix-ctxlogin: LoginBoxValidate: failed validation of user 'user1@CTXFAS.LAB', INVALID_PASSWORD

2018-03-27 10:15:52.228 <P9099:S3> citrix-ctxlogin: Audit_login_failure : Not yet implemented
<!--NeedCopy-->
```

解決策　/etc/krb5.conf にルート CA 証明書のフルパスが正しく設定されていることを確認します。フルパスは次のようになります:

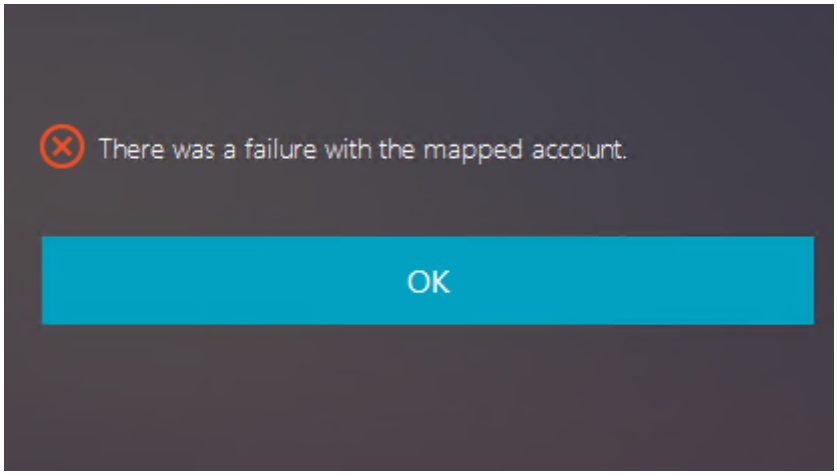
```
 [realms]

EXAMPLE.COM = {


    ......

    pkinit_anchors = FILE:/etc/pki/CA/certs/root.pem

    ......

 }

<!--NeedCopy-->
```

既存の設定が間違っている場合は、前述の「ルート CA 証明書のインストール」の手順に従って再設定します。

または、ルート CA 証明書が有効かどうかを確認します。

シャドウアカウントマッピングエラー

FAS は SAML 認証により構成されます。ADFS ユーザーが ADFS ログオンページでユーザー名とパスワードを入力すると、次のエラーが発生することがあります。

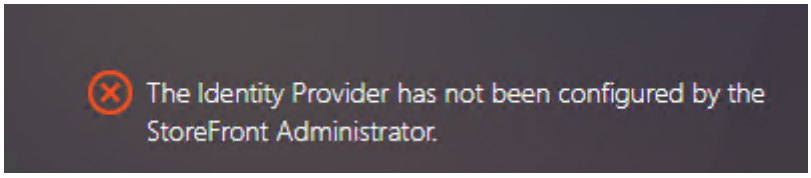


このエラーは、ADFS ユーザーが正常に確認されたが、AD にシャドウユーザーが構成されていないことを示しています。

解決策　AD にシャドウアカウントを設定します。

ADFS が構成されていない

FAS ストアへのログオン中に次のエラーが発生します：



ADFS が展開されていない状態で、FAS ストアが SAML 認証を使用するように構成されていることが原因です。

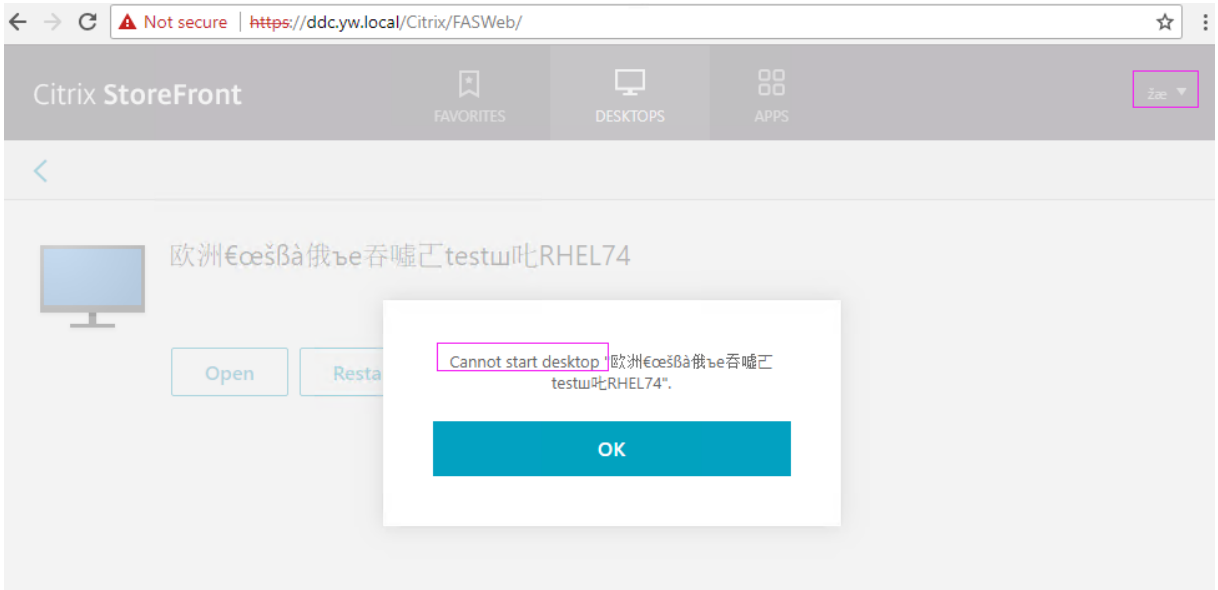
解決策　フェデレーション認証サービス用の ADFS IdP の展開詳しくは、「フェデレーション認証サービスの ADFS の展開」を参照してください。

関連情報

- 一般的な FAS 環境については、「フェデレーション認証サービスのアーキテクチャの概要」を参照してください。
- フェデレーション認証サービスの「詳細な構成」では「方法」の記事を紹介しています。

既知の問題

FAS が使用されている場合、英語以外の文字を使用して公開デスクトップまたはアプリセッションを開始しようとすると、失敗することがあります。



回避方法

CA ツールの ［テンプレートの管理］を右クリックし、［**Citrix_SmartcardLogon**］テンプレート上で ［**Active Directory** の情報から構築する］を ［要求に含まれる］に変更します: