新機能

累積更新プログラム 10 (CU10) は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU10 には、修正された問題は含まれていません。

> **注:**
>
> CU3 リリース以降、Linux VDA をインストールする前に、.NET Core Runtime 3.1 を前提条件としてインストールしてください。

累積更新プログラム 10 (CU10)

累積更新プログラム 10 (CU10)

リリース日: November 15, 2024

このリリースについて

累積更新プログラム 10 (CU10) は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU10 には修正された問題は含まれていません。

新機能

CPB/WCF のスケーラビリティの強化

Linux VDA CPB/WCF のスケーラビリティ強化が追加され、各 Citrix Cloud Connector は 3000 の Linux VDA をサポートできます。高可用性のために各サイトの場所に 2 つの Cloud Connector を展開でき、また各サイトの場所に最大 3000 の Linux VDA を展開できます。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 9 (CU9)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 8 (CU8)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 7 (CU7) ホットフィックス 1 (19.12.7001)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 7 (CU7)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 6 (CU6)

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 (初回リリース)(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/whats-new/1912-ltsr-initial-release.html)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

累積更新プログラム 9 (CU9) アップデート 1

リリース日: June 04, 2024

このリリースについて

累積更新プログラム 9 (CU9) アップデート 1 は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU9 には、1 の修正された問題が含まれています。

新機能

LDAPクエリのパフォーマンス最適化

新しいリリースでは、LDAPサーバーへの不要な通信が削除され、インデックス付きクエリキーを使用して、LDAPクエリのパフォーマンスを向上させ、コストを削減します。

累積更新プログラム 9 (CU9)

リリース日: April 30, 2024

このリリースについて

累積更新プログラム 9 (CU9) は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。CU9 には、修正された問題は含まれていません。

新機能

CPB/WCFのスケーラビリティの強化

Linux VDA CPB/WCFのスケーラビリティ強化が追加され、各Citrix Cloud Connectorは3000台のLinux VDAをサポートできます。高可用性を実現するために、各サイトの場所に2つのCloud Connectorを展開でき、各サイトの場所に最大3000台のLinux VDAを展開することもできます。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 8 (CU8)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 7 (CU7) ホットフィックス 1 (19.12.7001)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 7 (CU7)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 6 (CU6)

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 (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

1912 LTSR CU9 Update 1で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU9以降、以下の問題が修正されました。

- GPFファイルから大量のグループポリシーセットを処理する際、グループポリシーパーサーがCPUを100%消費する可能性があります。

累積更新プログラム8 (CU8)

リリース日: September 11, 2023

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は、全体的なパフォーマンス、セキュリティ、および安定性の向上に役立ついくつかの問題に対処しています。

Linux Virtual Delivery Agent 1912 LTSR累積更新プログラム7 (CU7) ホットフィックス1 (19.12.7001)

Linux Virtual Delivery Agent 1912 LTSR累積更新プログラム7 (CU7)

Linux Virtual Delivery Agent 1912 LTSR累積更新プログラム6 (CU6)

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 (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

累積更新プログラム 7 (CU7)

リリース日: March 15, 2023

このリリースについて

累積更新プログラム 7 (CU7) ホットフィックス 1 (19.12.7001) は、Linux Virtual Delivery Agent 1912 LTSR の最新リリースです。このホットフィックスは、1912 LTSR CU7 のリリース以降に報告された 1 つの問題を修正します。

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 7 (CU7)

Linux Virtual Delivery Agent 1912 LTSR 累積更新プログラム 6 (CU6)

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 (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

1912 LTSR CU7 Hotfix 1 (19.12.7001) で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU7 以降に修正された問題は次のとおりです。

- この修正はセキュリティ問題に対処します。詳細については、Knowledge Center の記事 CTX559370 を参照してください。

1912 LTSR CU7 で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU6 以降に修正された問題は次のとおりです。

- Microsoft Windows を実行しているラップトップから Linux VDA セッションに接続し、ラップトップの蓋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** に設定されている場合、ランダムなキーが VDA セッションに挿入されることがあります。
- 特定のラップトップを Linux VDA に接続し、**Fn** キーを押すと、**Delete** キーとして機能する場合があります。

累積更新プログラム 6 (CU6)

公開日: October 31, 2022

このリリースについて

CU6には、修正された問題は含まれていません。

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 (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

累積更新プログラム 5 (CU5)

リリース日: March 09, 2022

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）は、1912 LTSR CU4のリリース以降に報告された2つの問題を修正します。

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（初回リリース）

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

1912 LTSR CU5 で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU4 以降で、以下の問題が修正されました。

- 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。

- 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でのセッション起動が失敗する可能性があります。

累積更新プログラム 4 (CU4)

リリース日: November 03, 2021

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR Cumulative Update 4 (CU4) は、1912 LTSR CU3 のリリース以降に報告された 3 つの問題を修正します。

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 (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix 製品サブスクリプションアドバンテージの対象期間

1912 LTSR CU4 で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU3 以降で、以下の問題が修正されました。

- Citrix Workspace™ app for MacまたはLinuxを使用して接続されたLinux VDAで、Shift + Tabを押すと、**Shift**キーが2回押されたと認識される場合があります。
- Azure で Linux VM を作成するために Machine Creation Services™ が使用されている場合、RHEL で Linux VDA セッションを起動しようとすると失敗する可能性があります。
- SUSEまたはRHELでLinux VDAをアンインストールすると、/opt/Citrix/の場所にある空のフォルダーが削除されない場合があります。

累積更新プログラム 3 (CU3)

公開日: May 12, 2021

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR Cumulative Update 3 (CU3) は、1912 LTSR CU2のリリース以降に報告された9つの(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/whats-new/cumulative-update-3/fixed-issues.html)を修正しました。

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

非推奨と削除

Citrix 製品の Subscription Advantage 資格日

1912 LTSR CU3 で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU2 以降で、以下の問題が修正されました。

- 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場合があります。ユーザーには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表示されますが、その後メッセージが消えます。その結果、VDA上に複数の古いセッションが残ります。
- スキャンを実行した後、`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。
- Xiaomi Mi 10スマートフォンで、Picture Transfer Protocol (PTP) および Media Transfer Protocol (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。
- Citrix Desktop Viewer (CDViewer.exe) ウィンドウが消えた場合、Linux VDAへの再接続が失敗する場合があります。
- 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。
- Linux VDAセッションでWebカメラのUSBリダイレクトを使用している場合、`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。
- `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場合があります。
- スマートカードがLinux VDAに構成されている場合、動作しない場合があります。
- Linux VDAセッションに表示されるテキストが歪んだり、ぼやけたりする場合があります。

累積更新プログラム 2 (CU2)

リリース日: November 19, 2020

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR Cumulative Update 2 (CU2) は、1912 LTSR CU1 のリリース以降に報告された5つの問題を修正します。

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

Linux Virtual Delivery Agent 1912 LTSR (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix製品のサブスクリプション特典の対象日

1912 LTSR CU2 で修正された問題

Linux Virtual Delivery Agent 1912 LTSR CU1 以降で修正された問題は次のとおりです。

- LDAP署名が有効になっている場合、Delivery Controller™ へのLinux VDAの登録が失敗する可能性があります。
- VDAの起動を試みると、グレースクリーンになることがあります。この問題は、HDX™ユーザーポリシー検証のタイムアウトが原因です。多数のLDAPサーバーがあり、そのうちの1つ以上がVDAからアクセスできない場合に、タイムアウトが発生する可能性があります。
- 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。その他の規則は破棄されます。
- シームレスモードで公開アプリケーションを起動すると、公開アプリケーションは常に最前面に表示され、ローカルアプリケーションの上に重なります。公開アプリケーションを最小化した場合にのみ、ローカルアプリケーションを前面に表示できます。
- Machine Creation Services™ (MCS) で作成された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、期待どおりにホームフォルダーに自動的にコピーされない場合があります。
- 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変更を試みると、セッションが予期せず終了する可能性があります。

- 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可能性があります。

  SysStat.ReadUpTime: 要素 '29363.68' の変換中に NumberFormatException が発生しました。エラー: 入力文字列の形式が正しくありません。

累積更新プログラム 1 (CU1)

リリース日: May 7, 2020

このリリースについて

Linux Virtual Delivery Agent 1912 LTSR Cumulative Update 1 (CU1) は、1912 LTSR の初回リリース以降に報告された8つ以上の問題を修正します。

Linux Virtual Delivery Agent (初回リリース)

このリリースの既知の問題

非推奨と削除

Citrix製品のSubscription Advantage資格日

1912 LTSR CU1における修正された問題

Linux Virtual Delivery Agent 1912 LTSRの初回リリース以降、以下の問題が修正されました:

- Linux VDAが、ログオンしているユーザーのリストの表示に失敗する場合があります。
- リムーバブルUSBドライブをLinux VDAに汎用的にリダイレクトしようとすると、失敗する場合があります。この問題は、USBドライブがNTFS (New Technology File System) 形式でフォーマットされている場合に発生します。
- Linux VDAをバージョン1909またはバージョン1912に更新した後、初期化に時間がかかる場合があります。
- Quest Authentication Servicesを使用する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バージョン1909、1912 LTSR初回リリース、および2003を使用している場合に発生します。
- Linux VDAが、`Target frame rate` (FramesPerSecond) 設定で指定されているフレーム/秒に達しない場合があります。この問題は、Linux VDAにGPUがインストールされている場合に発生します。

- システムを再起動した後、`ctxjproxy`サービスがLDAPサーバーの検索に失敗する場合があります。
- 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と通信するポートが80ではない場合に発生します。
- Linux VDAをインストールする際、.NET Core Runtimeスクリプトが認証のために検証されない場合があります。

このリリースについて

新機能

1912 LTSR の新機能

Linux VDA バージョン 1912 には、以下の新機能と機能強化が含まれています。

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 プラットフォーム上に Linux VM を作成できます。詳しくは、「MCS を使用した Linux VM 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可能

MCS を使用して Linux VM 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、既存のすべての構成を継承できます。この実行中の VDA は、手動でインストールすることも、簡易インストールを使用して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 VM 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は、Linux VDA とクライアントデバイス間で 4 GB 以上のサイズのファイルを転送できるようになりました。この機能強化には、クライアントで Citrix Workspace™ app for Windows 1808 以降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 がバージョン 1.8.0 に更新されます。

1912 LTSR で修正された問題

Linux Virtual Delivery Agent 1909 以降、以下の問題が修正されました。

- 4Kモニターでは、キーストロークとリフレッシュレートに関連するGPUパフォーマンスの問題が発生する可能性があります。
- マウスとキーボードが同じウィンドウに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を変更できない場合、Linux VDIセッションが応答しなくなることがあります。
- IPv6アドレスのみを使用する仮想マシン (VM) を使用すると、Linux VDAの登録が失敗する場合があります。
- ポリシーを**Default**に設定すると、Linux VDAがデータベースを更新しない場合があります。この問題は、優先度の高いポリシーが**Default**に設定された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。
- ローカルキーボードレイアウト機能が有効になっている場合、クライアント側のハンガリー語環境では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。**DE**をローカル設定と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、ハンガリー語レイアウトでは機能しません。
- XClientとXServer間の通信を保護するために`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。
- Ubuntu 18.04では、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場合があります。

既知の問題

このリリースでは、以下の問題が確認されています。

- シームレスではない公開アプリケーションが起動直後に終了する場合があります。この問題は、Mutterをmutter-3.28.3-4より新しいバージョンにアップグレードした後に発生します。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。
- HDX 3D Proを有効にせずにNVIDIA GRID 3Dカードを使用すると、Linux VDAが期待どおりに動作しません。この問題は、RHEL 7.5以前、SUSE 12.3以前、およびUbuntu 16.04で発生します。その理由は、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では、複数のOpenGLライブラリが共存できないためです。
- ファイルのダウンロード中に予期しないウィンドウが表示されます。このウィンドウはファイルダウンロード機能には影響せず、しばらくすると自動的に消えます。
- PulseAudioのデフォルト設定により、サウンドサーバープログラムは20秒間アイドル状態が続くと終了します。PulseAudioが終了すると、オーディオは機能しません。この問題を回避するには、/etc/pulse/daemon.confファイルでexit-idle-time=-1を設定してください。
- SUSE 12.3の `libtcmalloc 4.3.0` により、プロセスが予期せず終了する場合があります。
- Ubuntu 16.04およびSUSE 12.3 VDAで、 `ctxhdx` サービスが予期せず終了する場合があります。この問題は、GNU Cライブラリ (`glibc`) バージョン2.22から2.24で発生します。この問題は `glibc 2.25` で修正されています。SUSE 12.3ディストリビューションを使用している場合は、SUSEが提供するパッチをインストールして問題を修正できます。Linux VDA 7.17のリリース時点では、Ubuntu 16.04用の修正はありません。
- SSL暗号化が有効でセッションの信頼性が無効になっている場合、Citrix Workspace™ app for Linuxでセッションを起動できません。
- `indicator-datetime-service` プロセスは$TZ環境変数を消費しません。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。
- Ubuntuグラフィック: HDX™ 3D Proでは、Desktop Viewerのサイズを変更した後、アプリケーションの周囲に黒いフレームが表示されたり、背景が黒く表示されたりする場合があります。
- Linux VDAの印刷リダイレクトによって作成されたプリンターは、セッションからログアウトした後も削除されない場合があります。

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

  この問題は、VDAサービスチケットの有効期限が切れて更新されると、最終的に自己解決する可能性があります。しかし、サービスチケットは有効期間が長いため、解決には時間がかかる場合があります。

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

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

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

- Windows向けCitrix Workspaceアプリでは、オーディオ録音中に音声の歪みが発生する可能性があります。

サードパーティ通知

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

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

廃止予定

この記事の発表は、段階的に廃止されるプラットフォーム、Citrix®製品、および機能について事前に通知し、お客様がタイムリーなビジネス上の意思決定を行えるようにすることを目的としています。Citrixは、それらがいつ廃止されるかを判断するために、顧客の使用状況とフィードバックを監視しています。発表は、その後のリリースで変更される可能性があり、すべての非推奨の機能や機能性を含まない場合があります。 製品ライフサイクルサポートの詳細については、(https://www.citrix.com/support/product-lifecycle/)の「製品ライフサイクルサポートポリシー」の記事を参照してください。

廃止予定と削除

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

項目	非推奨が発表された時期	削除時期
RHEL 6.9のサポート	1909	1909
RHEL 7.5、CentOS 7.5のサポート	1903	1903
RHEL 7.4、CentOS 7.4のサポート	1811	1811
RHEL 6.8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

Linuxディストリビューション

注:

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

CU3リリース以降、Linux VDAをインストールする前に、前提条件として.NET Core Runtime 3.1をインストールしてください。詳細については、`<https://docs.microsoft.com/ja-jp/dotnet/core/install/linux-package-managers>`を参照してください。

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

長期サービス (LTSR) 環境でのCurrent Release (CR) の使用方法やその他のFAQについては、Knowledge Centerの記事を参照してください。

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

重要:

OSベンダーからのサポートが終了した場合、Citrix®は問題の解決能力が制限される可能性があります。

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

- スースー・リナックス・エンタープライズ:
  - デスクトップ 12 Service Pack 3
  - サーバー 12 Service Pack 3
- レッドハット・エンタープライズ・リナックス
  - ワークステーション 7.7
  - ワークステーション 6.10
  - サーバー 7.7
  - サーバー 6.10
- CentOS リナックス
  - CentOS 7.7
  - CentOS 6.10
- Ubuntu リナックス
  - Ubuntu デスクトップ 18.04
  - Ubuntu サーバー 18.04
  - Ubuntu ライブサーバー 18.04
  - Ubuntu デスクトップ 16.04
  - Ubuntu サーバー 16.04
- Pardus リナックス
  - Pardus 17 (サポートされている機能範囲については、Knowledge Center の記事 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バージョンのマトリックスについては、以下の表を参照してください。詳細については、XorgModuleABIVersions を参照してください。

LINUXディストリビューション	XORGバージョン
RHEL 7.7、CentOS 7.7	1.20
RHEL 6.10, CentOS 6.10	1.17

LINUXディストリビューション	XORGバージョン
ウブンツ 18.04	1.19
ウブンツ 16.04	1.18
SUSE 12.3	1.18
パルドゥス 17	1.19

Ubuntu 16.04ではHWE Xorgサーバー1.19を使用しないでください。

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

注:

- CentOS 7.4にLinux VDA 1912 LTSR累積更新プログラム2（CU2）をインストールし、Citrix Virtual Apps and Desktopsサービスで使用する場合は、VDAの前に以下のコンポーネントをインストールしてください。
  - エックスオーグ 1.20.4
  - SELinux ポリシー 3.13.1-268
  - .NET Core ランタイム 2.1
  - GNOME 3.28.3 以降
- SUSE、RHEL、CentOS では Gnome および KDE デスクトップがサポートされています。Ubuntu 16.04 では Unity デスクトップがサポートされています。Ubuntu 18.04 では Gnome デスクトップがサポートされています。少なくとも 1 つのデスクトップがインストールされている必要があります。

シトリックス 仮想デスクトップ™

Linux VDA は、現在サポートされているすべてのバージョンの Citrix Virtual Desktops と互換性があります。Citrix Virtual Desktops の製品ライフサイクルに関する情報、および Citrix が特定のバージョンの製品のサポートを終了する時期については、Citrix 製品ライフサイクルマトリックス を参照してください。

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

サポートされるホストプラットフォームと仮想化環境

- ベアメタルサーバー
- シトリックス ハイパーバイザー™
- VMware ESX および ESXi
- マイクロソフト Hyper-V
- ニュータニックス AHV
- マイクロソフト Azure リソースマネージャー
- アマゾン ウェブ サービス (AWS)
- グーグル クラウド プラットフォーム (GCP)

ヒント:

サポートされているプラットフォームのリストについては、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 は、次の Active Directory 統合パッケージまたは製品をサポートしています。

- サンバ ウィンバインド
- クエスト認証サービス v4.1 以降
- セントリファイ ダイレクトコントロール
- SSSD
- PBIS (RHEL 7およびUbuntuと互換性があります)

ヒント:

サポートされているプラットフォームのリストについては、Active Directory統合パッケージのベンダーが提供するドキュメントを参照してください。

HDX™ 3D プロ

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

ハイパーバイザー

Linux VDAの場合、HDX 3D Proは、以下のハイパーバイザーによって提供されるGPUパススルーおよびGPU仮想化テクノロジーと互換性があります。

- シトリックス ハイパーバイザー
- VMware ESXおよびESXi
- ニュータニックス AHV

> **注:**
>
> これらの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

お使いのLinuxディストリビューションがサポートするNVIDIA GPUカードを確認するには、NVIDIA製品サポートマトリックスにアクセスし、**Hypervisor or Bare-Metal OS**、**Software Product Deployment**、**Hardware Supported**、および**Guest OS Support**の各列を確認してください。GPUカード用の最新のvGPUドライバーをインストールしてください。詳細については、NVIDIA Virtual GPU Software Supported GPUsを参照してください。

以下は、GPUパススルーおよびGPU仮想化のサポートについてテスト済みのGPUカードです。

GPUパススルー用にテスト済みのGPU:

- エヌビディア グリッド - テスラ T4
- エヌビディア GTX750Ti
- エヌビディア グリッド - テスラ M60
- NVIDIA GRID - K2
- エヌビディア グリッド - テスラ P40
- エヌビディア グリッド - テスラ P4
- エヌビディア グリッド - テスラ P100

vGPU用にテスト済みのGPU:

- エヌビディア グリッド - テスラ T4
- エヌビディア グリッド - テスラ V100
- エヌビディア グリッド - テスラ M60
- エヌビディア グリッド - テスラ M10
- エヌビディア グリッド - テスラ P40
- エヌビディア グリッド - テスラ P4
- エヌビディア グリッド - テスラ P100

インストール概要

このセクションでは、以下の手順について説明します。

- 簡単インストールを使用したクイックインストール（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を使用してLinux VMを作成する
- XenDesktop 7.6およびそれ以前のバージョン向けにDelivery Controllerを構成する

Easy Install を使用したクイックインストール (推奨)

> **重要:**

> 新規インストールの場合、迅速なインストールについてはこの記事を参照することをお勧めします。この記事では、Easy Install を使用して Linux VDA をインストールおよび構成する方法を順を追って説明します。Easy Install は、手動インストールよりも時間と労力を節約し、エラーが発生しにくいです。必要なパッケージをインストールし、構成ファイルを自動的にカスタマイズすることで、Linux VDA の実行環境をセットアップするのに役立ちます。

サポートされているディストリビューション

	ウィンバインド	SSSD	セントリファイ	PBIS
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
ウブンツ 18.04	はい	はい	はい	はい
ウブンツ 16.04	はい	はい	はい	はい

	ウィンバインド	SSSD	セントリファイ	PBIS
SUSE 12.3	はい	いいえ	はい	いいえ

Easy Install の使用

この機能を使用するには、次の手順を実行します。

1. 構成情報と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。

   (https://www.citrix.com/downloads/citrix-virtual-apps-and-desktops/)にアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開します。**コンポーネント**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。
3. Linux VDAのインストールを完了するために、ランタイム環境をセットアップします。

ステップ1：構成情報とLinuxマシンを準備する

簡単なインストールに必要な以下の構成情報を収集します。

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

重要:

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

考慮事項

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

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

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

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

  ログ:

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

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

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

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

  a. ドメインコントローラーで**管理ツール**を開きます。

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

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

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

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

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

    RHELの場合：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>`

    セントオーエス： 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>`

    スーゼ： 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の場合：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 サーバー上に /tmp/ctxinstall.conf ファイルが存在しない場合は、作成します。

    b. ファイルに「"centrifypkgpath=\<path name\>"」の行を追加します。

    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

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

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

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

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

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

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

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

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

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

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

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

- 簡易インストールは、インターネットからPBISパッケージを自動的にダウンロードするのに役立ちます。各ディストリビューションのURLは次のとおりです。

  RHEL 7 / CentOS 7 の場合: wget コマンド `<https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh>`

  Ubuntu の場合: wget コマンド `<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特定のバージョンのPBISパッケージを取得します。これを行うには、/opt/Citrix/VDA/sbin/ctxinstall.shファイル内の「"pbisDownloadPath"」行を変更して、PBISパッケージのURLを指定します。

  例については、次のスクリーンショットを参照してください。

手順2：ハイパーバイザーを準備する

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

Citrix Hypervisor™での時刻同期の修正

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

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、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値 1 を返します。

Microsoft Hyper-V での時刻同期の修正

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

管理オペレーティングシステムから:

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

> **注:**

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

ESX および ESXi での時刻同期の修正

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

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

1. vSphere Clientを開きます。
2. Linux VMの設定を編集します。
3. **Virtual Machine Properties**ダイアログで、**Options**タブを開きます。
4. **VMware Tools**を選択します。
5. **Advanced**ボックスで、**Synchronize guest time with host**のチェックを外します。

ステップ3：前提条件として.NET Core Runtimeをインストールする

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

- 1912 LTSR初期リリース、CU1、およびCU2の場合は、.NET Core Runtime 2.1をインストールします。
- CU3以降のリリースでは、.NET Core Runtime 3.1をインストールします。

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

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

ステップ4：Linux VDAパッケージをダウンロードする

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

ステップ 5: Linux VDAパッケージをインストールする

Linux VDAの環境をセットアップするには、次のコマンドを実行します。

RHELおよびCentOSディストリビューションの場合:

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

Ubuntuディストリビューションの場合:

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

sudo apt-get install -f
```

SUSEディストリビューションの場合:

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

ステップ 6: NVIDIA GRIDドライバーをインストールする

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

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

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX

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

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

ステップ 7: インストールを完了するためにランタイム環境をセットアップする

注:

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

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

注:

Easy Installは、27 MBを超える.NET Core Runtimeをダウンロードしている間、応答しないように見えることがあります。ダウンロードの進行状況については、/var/log/ctxinstall.logを確認してください。

手動で構成するには、次のコマンドを実行し、各プロンプトで関連するパラメーターを入力します。

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

サイレントモード:

サイレントモードでEasy Installを使用するには、ctxinstall.shを実行する前に、次の環境変数を設定します。

- **CTX_EASYINSTALL_HOSTNAME=host-name** – Linux VDAサーバーのホスト名を示します。
- **CTX_EASYINSTALL_DNS=ip-address-of-dns** – DNSのIPアドレス。
- **CTX_EASYINSTALL_NTPS=address-of-ntps** – NTPサーバーのIPアドレスまたは文字列名。
- **CTX_EASYINSTALL_DOMAIN=domain-name** – ドメインのNetBIOS名。

- **CTX_EASYINSTALL_REALM=realm-name** – Kerberosレルム名。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind \| sssd \| centrify \| pbis** – Active Directory統合方法を示します。
- **CTX_EASYINSTALL_USERNAME=domain-user-name** – ドメインユーザーの名前を示します。ドメインへの参加に使用されます。
- **CTX_EASYINSTALL_PASSWORD=password** – ドメインユーザーのパスワードを指定します。ドメインへの参加に使用されます。

ctxsetup.sh スクリプトは次の変数を使用します。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDA は、DNS CNAME レコードを使用して Delivery Controller 名を指定することをサポートしています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDA は、Delivery Controller に登録するために、スペースで区切られた Delivery Controller の完全修飾ドメイン名 (FQDN) のリストを必要とします。少なくとも 1 つの FQDN または CNAME を指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDA は、TCP/IP ポートを介して Delivery Controller と通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux VDA サービスは、マシンの起動後に開始されます。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux VDA サービスでは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux VDA のシステムファイアウォールで、必要なポート (デフォルトではポート 80 と 1494) を自動的に開くことができます。
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーションテクノロジーのセットである HDX 3D Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ (シングルセッション) モード (つまり、CTX_XDL_VDI_MODE=Y) 用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル (VDI) として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境の場合は、値を Y に設定します。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、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 ポートを含む LDAP FQDN のスペース区切りリストを提供できます。例: ad1.mycompany.com:389。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 ドメインのルートに設定された検索ベース (例: DC=mycompany,DC=com) を介して LDAP を照会します。検索パフォーマンスを向上させるには、検索ベース (例: OU=VDI,DC=mycompany,DC=com) を指定できます。不要な場合は、**\<none\>** に設定できます。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーテッド認証サービス (FAS) サーバーは、AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていませんが、代わりにセミコロンで区切られた FAS サーバーのリストを提供できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白をテキスト文字列 **\<none\>** で埋め、サーバーアドレスの順序を変更しないでください。
- **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 Runtime のインストールパス。デフォルトのパスは /usr/bin です。
- **CTX_XDL_START_SERVICE=Y \| N** – 構成完了時に Linux VDA サービスを開始するかどうか。

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

サイレントモードでは、環境変数を設定し、ctxinstall.shスクリプトを実行するために、以下のコマンド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
```

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \
```

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


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


CTX_XDL_START_SERVICE=Y|N \


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

ステップ8：XDPingを実行する

Linux VDA環境で一般的な構成の問題を確認するために、コマンドラインユーティリティであるLinux XDPingツールを提供しています。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コンピューターにインストールできます。XDPingは、Linux VDAパッケージがコンピューターにインストールされている必要はありません。このツールの詳細については、Knowledge Centerの記事CTX202015を参照してください。

ステップ9：Linux VDAを実行する

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo /sbin/service ctxhdx start


sudo /sbin/service ctxvda start
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo /sbin/service ctxvda stop
```

```
sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ 10: Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する

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

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

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

注:

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

ヒント:

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

ステップ 11: Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

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

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

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

> **重要:**
>
> アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、同じマシンへのデスクトップとアプリの配信をサポートしていません。

> マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 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使用して、この機能の使用によって発生する可能性のある問題をトラブルシューティングしてください。

SSSDを使用したドメインへの参加に失敗する

ドメインへの参加を試行すると、次のような出力でエラーが発生する可能性があります（画面出力についてはログを確認してください）。

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

/var/log/xdl/vda.log:

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

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

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

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

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

-  Kerberos is correctly configured on this machine.

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

Error Details:

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

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

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

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

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

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

/var/log/messages:

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

この問題を解決するには：

1. `rm -f /etc/krb5.keytab`コマンドを実行します。

2. `net ads leave $REALM -U $domain-administrator`コマンドを実行します。
3. Delivery Controller上のマシンカタログとデリバリーグループを削除します。
4. /opt/Citrix/VDA/sbin/ctxinstall.shを実行します。
5. Delivery Controller上にマシンカタログとデリバリーグループを作成します。

Ubuntuデスクトップセッションがグレースクリーンを表示する

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

この問題を解決するには：

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

   greeter-show-manual-login=true

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

/var/log/xdl/hdx.log:

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



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



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



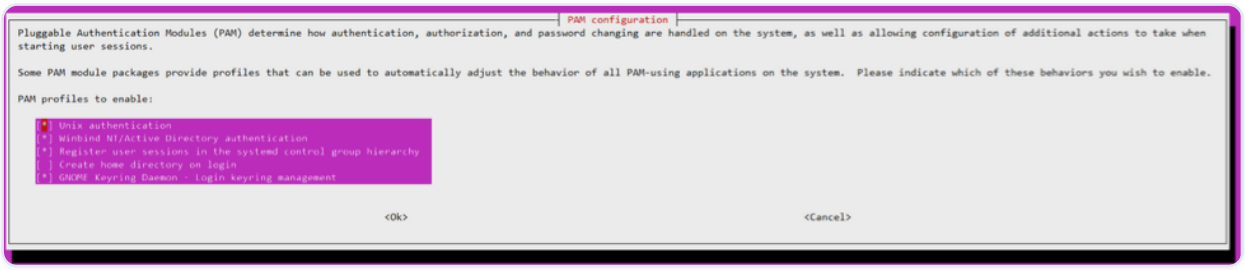
2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting normally.
```

> **ヒント:**

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

> この問題を解決するには：

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



セッションが起動しない、またはdbusエラーで短時間で終了する

/var/log/messages (RHELまたはCentOSの場合):

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



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



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



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

```
EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS)



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



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



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

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

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



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



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



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

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



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

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

SELinuxがSSHDのホームディレクトリへのアクセスを妨げる

ユーザーはセッションを起動できますが、ログオンできません。

/var/log/ctxinstall.log:

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



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



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



If you want to allow polyinstantiation to enabled



   Then you must tell SELinux about this by enabling the
```

```
'polyinstantiation_enabled' boolean.


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


    Do


       setsebool -P polyinstantiation_enabled 1


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


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


Then you should report this as a bug.


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


      Do


       allow this access for now by executing:


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

```
# semodule -i mypol.pp
```

この問題を解決するには：

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

   SELINUX=無効

2. VDA を再起動します。

RHEL/CentOS 用 Linux Virtual Delivery Agent を手動でインストールする

重要:

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

ステップ 1: VDA インストール用に RHEL 7/CentOS 7、RHEL 6/CentOS 6 を準備する

ステップ 1a: ネットワーク構成を確認する

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

ステップ 1b: ホスト名を設定する

マシンのホスト名が正しく報告されるように、**/etc/hostname** ファイル (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が含まれるようにします。

`hostname`

ステップ 1c: ホスト名にループバックアドレスを割り当てる

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

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

例:

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

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

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

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

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: 名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn


ping domain-controller-fqdn


nslookup delivery-controller-fqdn


ping delivery-controller-fqdn
```

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

ステップ1f: クロック同期の構成

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

RHEL 6.xおよびそれ以前のリリースでは、クロック同期にNTPデーモン (`ntpd`) を使用しますが、RHEL 7.xのデフォルト環境では、代わりに新しいChronyデーモン (`chronyd`) を使用します。これら2つのサービス間の構成および運用プロセスは似ています。

NTPサービスを構成する (RHEL 6/CentOS 6のみ)

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

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


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

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

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

変更を保存し、NTPデーモンを再起動します。

```
sudo /sbin/service ntpd restart
```

Chronyサービスを構成する (RHEL 7/CentOS 7のみ)

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

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



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

通常の展開では、時刻はローカルのドメインコントローラーから同期し、パブリックNTPプールサーバーから直接同期しないようにします。ドメイン内の各Active Directoryドメインコントローラーのサーバーエントリを追加します。

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

変更を保存し、Chronyデーモンを再起動します。

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

手順 1g: OpenJDK をインストールする

Linux VDA は OpenJDK に依存します。通常、ランタイム環境はオペレーティングシステムのインストールの一部としてインストールされます。

正しいバージョンを確認します。

```
sudo yum info java-1.8.0-openjdk
```

プリパッケージされた OpenJDK は以前のバージョンである可能性があります。必要に応じて最新バージョンに更新してください。

```
sudo yum -y update java-1.8.0-openjdk
```

新しいシェルを開き、Java のバージョンを確認します。

```
java -version
```

ヒント:

Delivery Controller との登録失敗を避けるため、OpenJDK 1.8.0 のみをインストールしていることを確認してください。システムから他のすべての Java バージョンを削除してください。

手順 1h: PostgreSQL をインストールする

Linux VDA には、RHEL 6 では PostgreSQL 8.4 以降、RHEL 7 では PostgreSQL 9.2 以降が必要です。

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
```

データベースを初期化し、マシン起動時にサービスが開始されるようにするには、次のインストール後の手順が必要です。このアクションにより、**/var/lib/pgsql/data** の下にデータベースファイルが作成されます。コマンドは PostgreSQL 8 と 9 で異なります。

- RHEL 7 のみ: PostgreSQL 9

```
sudo postgresql-setup initdb
```

- RHEL 6 のみ: PostgreSQL 8

```
sudo /sbin/service postgresql initdb
```

ステップ 1i: PostgreSQL を起動する

マシンの起動時にサービスを開始し、サービスをすぐに開始します。

- RHEL 7 のみ: PostgreSQL 9

```
sudo systemctl enable postgresql



sudo systemctl start postgresql
```

- RHEL 6 のみ: PostgreSQL 8

```
sudo /sbin/chkconfig postgresql on



sudo /sbin/service postgresql start
```

PostgreSQL のバージョンを確認するには、以下を使用します。

```
psql --version
```

データディレクトリが設定されていることを確認するには、**psql** コマンドラインユーティリティを使用します。

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

重要:

このリリースでは、`gperftools-libs` の新しい依存関係が追加されていますが、元のリポジトリには存在しません。`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てください。

ステップ 2: ハイパーバイザーを準備する

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

Citrix Hypervisor™ で時刻同期を修正する

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

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、Linux VM 内から Citrix Hypervisor の時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは 0 または 1 を返します。

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

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

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

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

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

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

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

管理オペレーティングシステムから:

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

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

ESXおよびESXiでの時刻同期の修正

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

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

1. vSphere Client を開きます。
2. Linux VM の設定を編集します。
3. **Virtual Machine Properties** ダイアログで、**Options** タブを開きます。
4. **VMware Tools** を選択します。
5. **Advanced** ボックスで、**Synchronize guest time with host** のチェックを外します。

ステップ 3: Linux 仮想マシン (VM) を Windows ドメインに追加する

Linux VDA は、Linux マシンを Active Directory (AD) ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ・ウィンバインド
- クエスト・オーセンティケーション・サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS (RHEL 7 のみ対応)

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDA のローカルアカウントと AD のアカウントで同じユーザー名が使用されている場合、セッションの起動に失敗する可能性があります。

サンバ・ウィンバインド

必要なパッケージをインストールまたは更新します。

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

マシンの起動時にWinbindデーモンが開始するように有効にする

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

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

Winbind認証を構成する

Winbindを使用してKerberos認証のため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
```

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

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

`--enablekrb5kdcdns --enablekrb5realmdns`

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

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

`kerberos method = secrets and keytab`

`winbind refresh tickets = true`

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

Windowsドメインに参加する

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

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

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

WinbindのPAMを構成する

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

`krb5_auth = yes`

`krb5_ccache_type = FILE` `mkhomedir = yes`

各設定の先頭にあるセミコロンが削除されていることを確認してください。これらの変更を適用するには、Winbindデーモンを再起動する必要があります。

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

ヒント:

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

/etc/krb5.conf を開き、\[libdefaults\] セクションの以下の設定を 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 は、すべての VDA マシン（Windows および Linux VDA）が Active Directory にコンピューターオブジェクトを持っていることを要求します。

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

```
sudo net ads testjoin
```

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

```
sudo net ads info
```

Kerberos 構成の確認

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

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

次のコマンドを使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username

id -u
```

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

```
klist
```

セッションを終了します。

```
exit
```

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

Quest 認証サービス

ドメインコントローラーでQuestを構成する

Active Directoryドメインコントローラーにクエストソフトウェアをインストールして構成済みであり、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されているものとします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには、次の手順を実行します。

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

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

Linux VDA で Quest を構成する

SELINUX ポリシーの適用を回避する

デフォルトの RHEL 環境では、SELinux が完全に適用されています。この適用により、Quest が使用する Unix ドメインソケット IPC メカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

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

`SELINUX=permissive`

この変更にはマシンの再起動が必要です。

```
reboot
```

重要:

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

VAS デーモンを構成する

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

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



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

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

PAM と NSS を構成する

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

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



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

Windows ドメインに参加する

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

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

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

ドメインメンバーシップを確認する

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

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

Quest が PAM を介してドメインユーザーを認証できることを確認するには、以前に使用したことのないドメインユーザーアカウントを使用して Linux VDA にログオンします。

```
ssh localhost -l domain\\username

id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

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

セントリファイ ダイレクトコントロール

Windows ドメインへの参加

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

```
su –

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

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

ドメインメンバーシップの確認

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

```
su –

adinfo
```

「Joined to domain」の値が有効であり、CentrifyDC モードが「connected」を返すことを確認します。モードが開始状態のままになっている場合、Centrify クライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all

adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の検証後、(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/installation-overview/redhat.html#step-6-install-the-linux-vda)に進みます。

SSSD

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

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

1. ドメインに参加し、ホストキータブを作成します
2. SSSDをセットアップします
3. NSS/PAMを構成します
4. Kerberos構成を検証します
5. ユーザー認証を検証します

必要なソフトウェア

Active Directoryプロバイダーは、SSSDバージョン1.9.0で初めて導入されました。それ以前のバージョンを使用している場合は、Active DirectoryでLDAPプロバイダーを構成するに記載されている手順に従ってください。

この記事に含まれる手順を使用する場合、以下の環境がテストおよび検証されています。

- RHEL 7.7以降
- CentOS 7.7 以降

ドメインへの参加とホストキーテーブルの作成

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

このセクションの情報は、`Samba` のアプローチのみを説明しています。`adcli` および `realmd` については、RHEL または CentOS のドキュメントを参照してください。これらの手順は、SSSD を構成する前に実行する必要があります。

必要なパッケージをインストールまたは更新します。

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

適切に構成されたファイルを持つ Linux クライアントで:

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

Samba および Kerberos 認証用にマシンを構成します。

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

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

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

`--enablekrb5kdcdns --enablekrb5realmdns`

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

`kerberos method = secrets and keytab`

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

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

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

SSSD のセットアップ

SSSD の設定は、次の手順で構成されます。

- 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（例: sssd.conf）に構成変更を加えます。
- **sssd** サービスを開始します。

sssd.conf 構成の例（必要に応じて追加オプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = ad.example.com

services = nss, pam



[domain/ad.example.com]

# Uncomment if you need offline logins

# cache_credentials = true
```

```
id_provider = ad

auth_provider = ad

access_provider = ad

ldap_id_mapping = true

ldap_schema = ad



# Should be specified as the lower-case version of the long version of the
Active Directory domain.

ad_domain = ad.example.com



# Kerberos settings

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U



# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



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

default_shell = /bin/bash

fallback_homedir = /home/%d/%u



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

```
available

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

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

sssd.conf のファイルの所有権とアクセス許可を設定します。

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

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

NSS/PAM の構成

RHEL/CentOS:

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

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



sudo service sssd start



sudo chkconfig sssd on
```

Kerberos 構成の確認

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

```
sudo klist -ke
```

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

```
sudo kinit –k MACHINE\$@REALM
```

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

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

```
sudo klist
```

ユーザー認証の確認

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

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

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

サポートされているログオン形式は次のとおりです。

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

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

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



id -u
```

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

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

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

```
klist
```

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

PBIS

必要な PBIS パッケージをダウンロードする

例:

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

PBIS インストールスクリプトを実行可能にする

例:

```
chmod +x pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

PBIS インストールスクリプトを実行する

例:

```
sh pbis-open-8.8.0.506.linux.x86_64.rpm.sh
```

Windowsドメインに参加する

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

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

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

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

ドメインメンバーシップを確認する

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

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。それ以外の場合は、ホスト名のみが表示されます。

ユーザー認証を確認する

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

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

手順4：前提条件として.NET Core Runtimeをインストールする

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

- 1912 LTSRの初期リリース、CU1、およびCU2の場合、.NET Core Runtime 2.1をインストールします。

- CU3以降のリリースでは、.NET Core Runtime 3.1をインストールします。

.NET Core Runtimeをインストールした後、`which dotnet`コマンドを実行してランタイムパスを見つけます。

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

ステップ5：Linux VDAパッケージをダウンロードする

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。Citrix Virtual Apps and Desktopsの適切なバージョンを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

ステップ6：Linux VDAをインストールする

新規インストールを行うことも、以前の2つのバージョンおよびLTSRリリースからの既存のインストールをアップグレードすることもできます。

新規インストールを行うには

1. (オプション)古いバージョンをアンインストールする

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

   a. Linux VDAサービスを停止します。

   ```
   sudo /sbin/service ctxvda stop



   sudo /sbin/service ctxhdx stop
   ```

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

   b. パッケージをアンインストールします。

```
sudo rpm -e XenDesktopVDA
```

注:

コマンドを実行するには、フルパスが必要です。または、**/opt/Citrix/VDA/sbin**と**/opt/Citrix/VDA/bin**をシステムパスに追加することもできます。

2. Linux VDAをインストールする

   - Linux VDA ソフトウェアを `Yum` を使用して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
     ```

     RHEL 6/CentOS 6 の場合:

     ```
     sudo yum install -y XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
     ```

   - RPM パッケージマネージャーを使用して Linux VDA ソフトウェアをインストールします。インストールする前に、以下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
     ```

     RHEL 6/CentOS 6 の場合:

     ```
     sudo rpm -i XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
     ```

     RHEL 7/CentOS 7 の RPM 依存関係リスト:

     ```
     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
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

RHEL 6/CentOS 6 の RPM 依存関係リスト:

```
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
```

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、セッションシャドウ機能を使用するために `python-websockify` と `x11vnc` を手動でインストールすることを目的としています。詳しくは、「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をアップグレードするには

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

- `Yum` を使用してソフトウェアを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
  ```

  RHEL 6/CentOS 6 の場合:

  ```
  sudo yum  install -y XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
  ```

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

  RHEL 7/CentOS 7の場合:

  ```
  sudo rpm -U XenDesktopVDA-19.12.0.50-1.el7_x.x86_64.rpm
  ```

  RHEL 6/CentOS 6の場合:

  ```
  sudo rpm -U XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
  ```

重要:

ソフトウェアのアップグレード後、Linux VDAマシンを再起動します。

手順7：NVIDIA GRIDドライバーをインストールする

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

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

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

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

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

```
yum install gcc



yum install "kernel-devel-$(uname -r)"



systemctl set-default multi-user.target
```

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

注:

GPUドライバーのインストール中に、各質問に対してデフォルト（「いいえ」）を選択します。

> **重要:**

> GPUパススルーが有効になった後、Linux VMはXenCenter経由でアクセスできなくなります。SSHを使用して接続してください。

> NVIDIA smiコードスニペット(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/linux-vda-nvidia-smi-code-snippet.png)

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

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

ステップ8：Linux VDAを構成する

パッケージをインストールした後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

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

プロンプトによる構成

プロンプトに従って手動で構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトで必要とされるオプションを環境変数で指定します。必要なすべての変数が存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerに登録するために使用する、スペース区切りのDelivery Controllerの完全修飾ドメイン名（FQDN）のリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - マシンの起動後にLinux Virtual Desktopサービスが開始されます。値はデフォルトでYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktopサービスでは、システムファイアウォールを介した受信ネットワーク接続が許可されている必要があります。Linux Virtual Desktop用に、必要なポート（デフォルトでポート80と1494）をシステムファイアウォールで自動的に開くことができます。デフォルトでYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \| 5** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ ウィンバインド
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーションテクノロジーのセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境では、この変数をYに設定します。この変数はデフォルトでNに設定されています。

- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに制限するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDAはDNSにクエリを実行してLDAPサーバーを検出します。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを含むLDAP FQDNのスペース区切りリストを提供できます。例：ad1.mycompany.com:389。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPにクエリを実行します。検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS)サーバーは、ADグループポリシーを介して構成されます。Linux VDAはADグループポリシーをサポートしていないため、代わりにFASサーバーのセミコロン区切りリストを提供できます。シーケンスはADグループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を**\<none\>**というテキスト文字列で埋め、サーバーアドレスのシーケンスを変更せずに維持してください。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス（`ctxvda`）をサポートするための.NET Core Runtime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 の構成が完了したときに Linux VDA サービスが開始されるかどうか。デフォルトでは Y に設定されています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_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
```

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST='list-ddc-fqdns' \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=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
```

構成の変更を削除する

シナリオによっては、Linux VDA パッケージをアンインストールせずに、**ctxsetup.sh** スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

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

構成の変更を削除するには：

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

重要:

> このスクリプトは、データベースからすべての構成データを削除し、Linux VDA を動作不能にします。

構成ログ

ctxsetup.sh および **ctxcleanup.sh** スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル **/tmp/xdl.configure.log** に書き込まれます。

変更を有効にするには、Linux VDA サービスを再起動します。

手順 9：XDPing を実行する

Linux VDA 環境における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` ツールを提供しています。サポートされている Linux ディストリビューションを実行している任意のコンピューターに `XDPing` パッケージをインストールできます。`XDPing` は、Linux VDA パッケージがコンピューターにインストールされている必要はありません。このツールの詳細については、Knowledge Center の記事 CTX202015 を参照してください。

手順 10：Linux VDA を実行する

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

Linux VDAの起動:

Linux VDAサービスを起動するには:

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDAの停止:

Linux VDAサービスを停止するには:

```
sudo /sbin/service ctxvda stop
```

```
sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDAの再起動:

Linux VDAサービスを再起動するには:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDAのステータスの確認:

Linux VDAサービスの実行ステータスを確認するには:

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ11：Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する

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

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

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

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

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

ステップ12：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

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

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

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

重要:

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

マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

SUSE 用 Linux Virtual Delivery Agent を手動でインストールする

重要:

新規インストールの場合、迅速なインストールにはeasy installを使用することをお勧めします。イージーインストールは、時間と労力を節約し、この記事で詳述されている手動インストールよりもエラーが発生しにくいです。

ステップ 1: インストールの準備

ステップ 1a: YaST ツールを起動する

SUSE Linux Enterprise YaST ツールは、オペレーティングシステムのあらゆる側面を設定するために使用されます。

テキストベースの YaST ツールを起動するには:

```
su -


yast
```

または、UI ベースの YaST ツールを起動するには:

```
su -



yast2 &
```

ステップ 1b: ネットワークを構成する

以下のセクションでは、Linux VDA で使用されるさまざまなネットワーク設定とサービスの構成に関する情報を提供します。ネットワークの構成は、Network Manager などの他の方法ではなく、YaST ツールを介して実行されます。これらの手順は、UI ベースの YaST ツールを使用することを前提としています。テキストベースの YaST ツールも使用できますが、ここでは文書化されていない異なるナビゲーション方法があります。

ホスト名と DNS を構成する

1. YaST 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、**ネットワーク設定方法**を**Wicked Service**に変更します。
3. **ホスト名/DNS**タブを開きます。
4. **DHCP 経由でホスト名を変更**のチェックを外します。
5. **ホスト名をループバック IP に割り当てる**をチェックします。
6. ネットワーク設定を反映するように、以下を編集します。
    - 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、これはADドメインコントローラーのIPアドレスです。
    - ドメイン検索リスト – DNSドメイン名を追加します。

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

> **ヒント:**

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

マルチキャストDNSを無効にする

SLEDのみ、デフォルト設定ではマルチキャストDNS（mDNS）が有効になっており、名前解決の結果が不安定になる可能性があります。SLESではmDNSはデフォルトで有効になっていないため、操作は不要です。

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

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

次のように変更します。

```
hosts: files dns
```

ホスト名を確認する

ホスト名が正しく設定されていることを確認します。

```
hostname
```

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

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドはマシンのFQDNを返します。

名前解決とサービス到達可能性の確認

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

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn
```

```
nslookup delivery-controller-fqdn
```

```
ping delivery-controller-fqdn
```

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

ステップ1c: NTPサービスの構成

VDA、Delivery Controller、およびドメインコントローラー間で正確なクロック同期を維持することが重要です。Linux VDAを仮想マシンとしてホストすると、クロックスキューの問題が発生する可能性があります。このため、リモートNTPサービスを使用して時刻を維持することが推奨されます。デフォルトのNTP設定にはいくつかの変更が必要になる場合があります。

1. YaST NTP設定を開き、**全般設定**タブを選択します。
2. NTPデーモンを開始セクションで、**今すぐ起動時**をチェックします。
3. 存在する場合は、**規律のないローカルクロック (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. **サーバータイプ**を選択し、**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せずにそのままにします。
8. **テスト**をクリックして、NTPサービスに到達可能であることを確認します。
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 ポリシーに関する既知の SUSE の問題により、NTP デーモンが起動に失敗する場合があります。詳細については、解決策 を参照してください。

手順 1d: Linux VDA 依存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、以下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 ランタイム環境 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

一部の必須パッケージは、すべての SUSE Linux Enterprise リポジトリで利用できるわけではありません。

- SLED 12: PostgreSQL は SLES 12 で利用可能ですが、SLED 12 では利用できません。ImageMagick は SLE 12 SDK ISO またはオンラインリポジトリ経由で利用できます。
- 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経由で利用できます。

この問題を解決するには、インストール元の 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入手します。つまり、SLED では SLES メディアから不足しているパッケージをインストールし、SLES では SLED メディアから不足しているパッケージをインストールします。次のアプローチでは、SLED と SLES の両方の ISO メディアファイルをマウントし、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
```

- 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
```

Kerberos クライアントをインストールする

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

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

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

OpenJDK をインストールする

Linux VDA は OpenJDK 1.8.0 に依存します。

> **ヒント:**

> Delivery Controller への登録失敗を避けるため、OpenJDK 1.8.0 のみをインストールしていることを確認してください。システムから他のすべてのバージョンの Java を削除してください。

- **SLED:**

1. SLED では、Java ランタイム環境は通常、オペレーティングシステムとともにインストールされます。インストールされているかどうかを確認します。

```
sudo zypper info java-1_8_0-openjdk
```

2. ステータスが期限切れとして報告された場合は、最新バージョンに更新します。

```
sudo zypper update java-1_8_0-openjdk
```

3. Java のバージョンを確認します。

```
java -version
```

- **SLES:**

1. SLESで、Javaランタイム環境をインストールします。

```
sudo zypper install java-1_8_0-openjdk
```

2. Javaのバージョンを確認します。

```
java -version
```

PostgreSQLをインストールする

SLED/SLES 12で、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
```

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

```
sudo systemctl enable postgresql



sudo systemctl restart postgresql
```

データベースファイルは/var/lib/pgsql/dataにあります。

リポジトリを削除する

依存パッケージがインストールされたら、以前に設定した代替エディションのリポジトリは削除してメディアをアンマウントできます。

- SLED 12で、パッケージを削除するコマンドを実行します。

```
sudo zypper rr sles



sudo umount /mnt/sles



sudo rmdir /mnt/sles
```

- SLED/SLES 12で、パッケージを削除するコマンドを実行します。

```
sudo zypper rr sdk



sudo umount /mnt/sdk



sudo rmdir /mnt/sdk
```

手順2：ハイパーバイザー用にLinux VMを準備する

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

Citrix Hypervisor™での時刻同期を修正する

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

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、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

```
xen.independent_wallclock = 1
```

これらの変更を確認するには、システムを再起動します。

```
reboot
```

再起動後、設定が正しいことを確認します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

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

管理オペレーティングシステムから：

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

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

ESXおよびESXiでの時刻同期の修正

VMwareの時刻同期機能が有効になっている場合、準仮想化された各Linux VM内で、NTPとハイパーバイザーの両方がシステムクロックを同期しようとする問題が発生します。他のサーバーとのクロックのずれを防ぐため、各Linuxゲスト内のシステムクロックはNTPと同期させる必要があります。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がインストールされた準仮想化Linuxカーネルを実行している場合：

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

ステップ3：Linux仮想マシン（VM）をWindowsドメインに追加する

Linux VDAは、LinuxマシンをActive Directory（AD）ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ ウィンバインド(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/installation-overview/suse.html#samba-winbind)
- クエスト認証サービス(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/installation-overview/suse.html#quest-authentication-services)
- セントリファイ ダイレクトコントロール

選択した方法に基づいて指示に従ってください。

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

サンバ Winbind

Windowsドメインへの参加

ドメインコントローラーに到達可能である必要があり、ドメインにマシンを追加する権限を持つActive Directoryユーザーアカウントが必要です。

1. YaST Windowsドメインメンバーシップを開きます。
2. 次の変更を行います。
   - **ドメインまたはワークグループ**をActive Directoryドメインの名前またはドメインコントローラーのIPアドレスに設定します。ドメイン名が大文字であることを確認してください。
   - **Linux認証にSMB情報も使用する**をチェックします。

- **ログイン時にホームディレクトリを作成する**をチェックします。
- **SSHのシングルサインオン**をチェックします。
- **オフライン認証**がチェックされていないことを確認します。このオプションはLinux VDAと互換性がありません。

3. **OK**をクリックします。一部のパッケージのインストールを求められた場合は、**インストール**をクリックします。
4. ドメインコントローラーが見つかった場合、ドメインに参加するかどうかを尋ねられます。**はい**をクリックします。
5. プロンプトが表示されたら、ドメインにコンピューターを追加する権限を持つドメインユーザー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. sambaおよびkrb5パッケージのインストールを求められた場合は、 をクリックします。

YaSTは、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を示している場合があります。マシンを再起動することをお勧めします。

```
su -



reboot
```

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では、デフォルトのKerberos資格情報キャッシュ名の指定が、通常の **FILE:/tmp/krb5cc_%{uid}** から **DIR:/run/user/%{uid}/krb5cc** に変更されました。この新しいDIRキャッシュ方法はLinux VDAと互換性がないため、手動で変更する必要があります。rootユーザーとして **/etc/krb5.conf** を編集し、設定されていない場合は **\[libdefaults\]** セクションに次の設定を追加します。

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

ドメインメンバーシップの確認

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

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
```

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

```
sudo net ads info
```

Kerberos構成の確認

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

```
sudo klist –ke
```

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

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

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

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

```
sudo klist
```

マシンアカウントの詳細を確認するには、次を使用します。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username

id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

```
klist
```

セッションを終了します。

```
exit
```

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

Quest認証サービス

ドメインコントローラーでのQuestの構成

Active Directoryドメインコントローラーにquestソフトウェアをインストールして構成し、Active Directoryでコンピューターオブジェクトを作成するための管理者権限が付与されていることを前提とします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

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

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

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

Linux VDAでのQuestの構成

VASデーモンの構成

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

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



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

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

PAMとNSSの構成

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

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



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

Windowsドメインへの参加

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

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

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

ドメインメンバーシップの確認

Delivery Controllerでは、すべてのVDAマシン（WindowsおよびLinux VDA）がActive Directoryにコンピューターオブジェクトを持っている必要があります。Questに参加しているLinuxマシンがドメイン上にあることを確認するには：

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

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

`ERROR: No domain could be found.`

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

ユーザー認証の確認

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username

id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

同様のテストは、GnomeまたはKDEコンソールに直接ログオンして実行できます。ドメイン参加の確認後、手順6：Linux VDAをインストールするに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインへの参加

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

```
su -

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

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

ドメインメンバーシップの確認

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

```
su -
```

```
adinfo
```

Joined to domain の値が有効であり、**CentrifyDC** の **mode** が **connected** を返すことを確認します。モードが開始状態のままになっている場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます。

```
adinfo --sysinfo all



adinfo –diag
```

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

```
adinfo --test
```

ドメイン参加の検証後、Step 6: Install the Linux VDAに進みます。

ステップ 4: 前提条件として .NET Core Runtime をインストールする

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

- 1912 LTSR 初期リリース、CU1、および CU2 の場合、.NET Core Runtime 2.1 をインストールします。
- CU3 以降のリリースの場合、.NET Core Runtime 3.1 をインストールします。

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

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

ステップ 5: Linux VDA パッケージをダウンロードする

Citrix Virtual Apps and Desktops ダウンロードページにアクセスします。Citrix Virtual Apps and Desktops の適切なバージョンを展開し、**Components** をクリックして、お使いの Linux ディストリビューションに一致する Linux VDA パッケージをダウンロードします。

ステップ 6: Linux VDA をインストールする

ステップ 6a: 古いバージョンをアンインストールする

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

1. Linux VDA サービスを停止する:

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

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

2. パッケージをアンインストールする:

```
sudo rpm -e XenDesktopVDA
```

> **重要:**
>
> 最新の2つのバージョンからのアップグレードがサポートされています。

> **注:**
>
> インストールコンポーネントは **/opt/Citrix/VDA/** にあります。

> コマンドを実行するには、フルパスが必要です。または、**/opt/Citrix/VDA/sbin** と **/opt/Citrix/VDA/bin** をシステムパスに追加することもできます。

ステップ6b：Linux VDAをインストールする

Zypperを使用してLinux VDAソフトウェアをインストールします。

SUSE 12の場合：

```
sudo zypper install XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
```

RPMパッケージマネージャーを使用してLinux VDAソフトウェアをインストールします。その前に、以下の依存関係を解決してください。

SUSE 12の場合：

```
sudo rpm -i XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
```

ステップ6c：Linux VDAをアップグレードする（オプション）

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

SUSE 12の場合：

```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
```

SUSE 12のRPM依存関係リスト：

```
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

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.8.1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

rpmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsLzma) <= 4.4.6-1

libtcmalloc4 >= 2.5
```

```
libcap-progs >= 2.22



xorg-x11-server >= 7.6_1.18.3-76.15



ibus >= 1.5
```

重要:

アップグレード後、Linux VDAマシンを再起動してください。

ステップ7：NVIDIA GRIDドライバーをインストールする

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

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

- シトリックス ハイパーバイザー
- ヴイエムウェア ESX

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

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

ステップ8：Linux VDAを構成する

パッケージをインストールした後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsetup.sh –help
```

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトで必要なオプションを環境変数で指定します。必要なすべての変数が存在する場合、スクリプトは情報を要求しません。

サポートされている環境変数には、次のものがあります。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用する、スペース区切りのDelivery Controllerの完全修飾ドメイン名（FQDN）のリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポートを介してDelivery Controllerと通信します。デフォルトではポート80です。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** - Linux Virtual Desktopサービスは、マシンの起動後に開始されます。この値はデフォルトでYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktopサービスは、システムファイアウォールを介して受信ネットワーク接続が許可される必要があります。Linux Virtual Desktopに必要なポート（デフォルトではポート80と1494）をシステムファイアウォールで自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4** – Linux VDAは、Delivery Controllerで認証するためにKerberos構成設定を必要とします。Kerberos構成は、システムにインストールおよび構成されているActive Directory統合ツールから決定されます。使用するサポートされているActive Directory統合方法を指定します。
  - 1 – サンバ Winbind

    - 2 – クエスト認証サービス
    - 3 – セントリファイ DirectControl
    - 4 – SSSD

- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDAは、リッチグラフィックアプリケーションの仮想化を最適化するために設計されたGPUアクセラレーション技術のセットであるHDX 3D Proをサポートしています。HDX 3D Proが選択されている場合、VDAはVDIデスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。
- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro環境では、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDAはDNSを介してLDAPサーバーを検出します。DNS検索結果をローカルサイトに限定するには、DNSサイト名を指定します。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDAは、LDAPサーバーを検出するためにDNSを照会します。DNSがLDAPサービスレコードを提供できない場合、LDAPポートを含むスペース区切りのLDAP FQDNのリストを提供できます。例：ad1.mycompany.com:389。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDAは、Active Directoryドメインのルートに設定された検索ベース（例：DC=mycompany,DC=com）を介してLDAPを照会します。検索パフォーマンスを向上させるために、検索ベース（例：OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで**\<none\>**に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – フェデレーション認証サービス (FAS) サーバーは、AD グループポリシーを通じて構成されます。Linux VDA は AD グループポリシーをサポートしないため、代わりにセミコロンで区切られた FAS サーバーのリストを指定できます。シーケンスは、AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスのシーケンスを変更せずに維持してください。
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** – 新しいブローカーエージェントサービス (`ctxvda`) をサポートするための .NET Core Runtime 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 の構成が完了したときに Linux VDA サービスを開始するかどうか。デフォルトでは Y に設定されています。

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
```

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \


CTX_XDL_AD_INTEGRATION=1|2|3|4 \
```

```
CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_START_SERVICE=Y|N \

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

構成の変更を削除する

シナリオによっては、Linux VDA パッケージをアンインストールせずに、**ctxsetup.sh** スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

```
sudo /usr/local/sbin/ctxcleanup.sh --help
```

構成の変更を削除するには：

```
sudo /usr/local/sbin/ctxcleanup.sh
```

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDA を動作不能にします。

構成ログ

ctxsetup.sh および **ctxcleanup.sh** スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイルに書き込まれます。

`/tmp/xdl.configure.log`

変更を有効にするには、Linux VDA サービスを再起動します。

ステップ 9：XDPing を実行する

Linux VDA環境における一般的な構成の問題を確認するために、コマンドラインユーティリティであるLinux `XDPing` ツールを提供しています。サポートされているLinuxディストリビューションを実行している任意のコンピューターに`XDPing` パッケージをインストールできます。`XDPing` は、Linux VDAパッケージがコンピューターにインストールされてい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。

ステップ10：Linux VDAを実行する

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

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo /sbin/service ctxhdx start



sudo /sbin/service ctxvda start
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx stop
```

注:

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

Linux VDAを再起動する：

Linux VDAサービスを再起動するには：

```
sudo /sbin/service ctxvda stop



sudo /sbin/service ctxhdx restart



sudo /sbin/service ctxvda start
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo /sbin/service ctxvda status



sudo /sbin/service ctxhdx status
```

ステップ11：Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する

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

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

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

注:

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

ヒント:

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

ステップ12：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

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

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

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

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

> マシンカタログとデリバリーグループの作成方法については、「Citrix Virtual Apps and Desktops 7 1912 LTSR」を参照してください。

Ubuntu 用 Linux Virtual Delivery Agent を手動でインストールする

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

ステップ 1: VDA インストール用に Ubuntu を準備する

ステップ 1a: ネットワーク構成を確認する

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

Ubuntu 18.04 Live Server を使用している場合は、ホスト名を設定する前に、**/etc/cloud/cloud.cfg** 構成ファイルで次の変更を行ってください。

```
preserve_hostname: true
```

ステップ 1b: ホスト名を設定する

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

```
hostname
```

ステップ 1c: ホスト名にループバックアドレスを割り当てる

マシンの DNS ドメイン名と完全修飾ドメイン名 (FQDN) が正しく報告されることを確認してください。そのためには、**/etc/hosts** ファイルの次の行を変更して、FQDN とホスト名を最初の 2 つのエントリとして含めます。

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

例:

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

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

> **注:**

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

> **ヒント:**

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

ステップ 1d: ホスト名を確認する

ホスト名が正しく設定されていることを確認してください。

```
hostname
```

このコマンドは、マシンのホスト名のみを返し、FQDNは返しません。

FQDNが正しく設定されていることを確認します。

```
hostname -f
```

このコマンドは、マシンのFQDNを返します。

ステップ1e: マルチキャストDNSを無効にする

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

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

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

変更後:

`hosts: files dns`

ステップ1f: 名前解決とサービス到達可能性を確認する

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

```
nslookup domain-controller-fqdn



ping domain-controller-fqdn
```

```
nslookup delivery-controller-fqdn
```

```
ping delivery-controller-fqdn
```

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

ステップ1g: クロック同期を構成する (chrony)

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

chronyをインストールします。

```
apt-get install chrony
```

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

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

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

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

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

変更を保存し、Chronyデーモンを再起動します。

```
sudo systemctl restart chrony
```

ステップ1h: OpenJDKをインストールする

Linux VDAはOpenJDKに依存しています。通常、ランタイム環境はオペレーティングシステムのインストールの一部としてインストールされます。

Ubuntu 16.04では、以下を使用してOpenJDKをインストールします。

```
sudo apt-get install -y default-jdk
```

Ubuntu 18.04では、以下を使用してOpenJDKをインストールします。

```
sudo apt-get install -y openjdk-8-jdk
```

ステップ1i: PostgreSQLをインストールする

Linux VDAはUbuntuでPostgreSQLバージョン9.xを必要とします。

```
sudo apt-get install -y postgresql



sudo apt-get install -y libpostgresql-jdbc-java
```

ステップ1j: Motifをインストールする

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

ステップ1k: その他のパッケージをインストールする

```
sudo apt-get install -y libsasl2-2



sudo apt-get install -y 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
```

ステップ1l: 次のパッケージをインストールする (Ubuntu 18.04のみ)

```
sudo apt-get install -y libgtk2.0-0
```

ステップ2: ハイパーバイザーを準備する

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

Citrix Hypervisor™での時刻同期を修正する

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

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、Linux VM内からCitrix Hypervisorの時刻同期機能が存在し、有効になっているかどうかを確認できます。

```
su -



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

このコマンドは0または1を返します。

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

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

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

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

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

`xen.independent_wallclock = 1`

これらの変更を確認するには、システムを再起動します。

```
su -



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

このコマンドは値1を返します。

Microsoft Hyper-Vでの時刻同期の修正

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

管理オペレーティングシステムから：

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

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

ESXおよびESXiでの時刻同期の修正

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

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

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

ステップ3：Linux仮想マシン（VM）をWindowsドメインに追加する

Linux VDAは、LinuxマシンをActive Directory (AD) ドメインに追加するためのいくつかの方法をサポートしています。

- サンバ・ウィンバインド
- クエスト・オーセンティケーション・サービス
- セントリファイ・ダイレクトコントロール
- SSSD
- PBIS(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/installation-overview/ubuntu.html#pbis)

選択した方法に基づいて指示に従ってください。

> **注:**
>
> Linux VDAのローカルアカウントとADのアカウントで同じユーザー名が使用されている場合、セッションの起動に失敗する可能性があります。

サンバ Winbind

必要なパッケージをインストールまたは更新する

```
sudo apt-get install winbind samba libnss-winbind libpam-winbind krb5-config krb5-locales krb5-user
```

マシンの起動時にWinbindデーモンが開始するように有効にする

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

```
sudo systemctl enable winbind
```

Kerberosを構成する

ルートユーザーとして **/etc/krb5.conf** を開き、次の設定を行います。

```
[libdefaults]



default_realm = REALM



dns_lookup_kdc = false









[realms]



REALM = {



admin_server = domain-controller-fqdn
```

```
kdc = domain-controller-fqdn

}




[domain_realm]

domain-dns-name = REALM

.domain-dns-name = REALM
```

このコンテキストでの **domain-dns-name** プロパティは、**example.com** のようなDNSドメイン名です。**REALM** は、**EXAMPLE.COM** のような大文字のKerberosレルム名です。

Winbind認証を構成する

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

/etc/samba/smb.conf を開き、次の設定を行います。

```
[global]

workgroup = WORKGROUP

security = ADS
```

```
realm = REALM

encrypt passwords = yes

idmap config *:range = 16777216-33554431

winbind trusted domains only = no

kerberos method = secrets and keytab

winbind refresh tickets = yes

template shell = /bin/bash
```

WORKGROUP は **REALM** の最初のフィールドであり、**REALM** は大文字のKerberosレルム名です。

`nsswitch` を構成する

/etc/nsswitch.conf を開き、次の行に `winbind` を追加します。

`passwd: compat winbind`

`group: compat winbind`

Windows ドメインへの参加

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

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

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

`winbind` の再起動

```
sudo systemctl restart winbind
```

Winbind 用の PAM の構成

次のコマンドを実行し、**Winbind NT/Active Directory authentication** および **Create home directory on login** オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ヒント:

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

ドメインメンバーシップの確認

Delivery Controller は、Windows または Linux のいずれであっても、すべての VDA マシンが Active Directory にコンピューターオブジェクトを持っていることを要求します。

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
```

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

```
sudo net ads info
```

Kerberos 構成の確認

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

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

以下を使用して、マシンのアカウントの詳細を調べます。

```
sudo net ads status
```

ユーザー認証の確認

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

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

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

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

```
ssh localhost -l domain\\username
```

```
id -u
```

注:

SSH コマンドを正常に実行するには、SSH が有効になっており、正しく機能していることを確認してください。

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

```
ls /tmp/krb5cc_uid
```

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

```
klist
```

セッションを終了します。

```
exit
```

同様のテストは、Gnome または KDE コンソールに直接ログオンして実行できます。ドメイン参加の確認後、(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/installation-overview/ubuntu.html#step-6-install-the-linux-vda)に進みます。

ヒント:

ユーザー認証に成功しても、ドメインアカウントでログオンしたときにデスクトップが表示されない場合は、マシンを再起動してからもう一度試してください。

Quest 認証サービス

ドメインコントローラーで Quest を構成する

Active Directory ドメインコントローラーに Quest ソフトウェアをインストールして構成し、Active Directory でコンピューターオブジェクトを作成するための管理者権限が付与されていることを前提とします。

ドメインユーザーがLinux VDAマシンにログオンできるようにする

ドメインユーザーがLinux VDAマシンでHDX™セッションを確立できるようにするには、次の手順を実行します。

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

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

Linux VDAでQuestを構成する

SELINUXポリシーの適用を回避する

デフォルトのRHEL環境では、SELinuxが完全に適用されています。この適用により、Questが使用するUnixドメインソケットIPCメカニズムが妨げられ、ドメインユーザーがログオンできなくなります。

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

`SELINUX=disabled`

この変更にはマシンの再起動が必要です。

```
reboot
```

重要:

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

VASデーモンを構成する

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

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



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

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

PAMとNSSを構成する

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

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



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

Windowsドメインに参加する

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

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

ユーザーは、コンピューターをActive Directoryドメインに参加させる権限を持つ任意のドメインユーザーです。domain-nameは、たとえばexample.comのようなドメインのDNS名です。

ドメインメンバーシップを確認する

Delivery Controllerは、WindowsまたはLinuxのいずれであっても、すべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持つことを要求します。Questに参加しているLinuxマシンがドメイン上にあることを確認するには：

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

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

`ERROR: No domain could be found.`

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

ユーザー認証を確認する

QuestがPAMを介してドメインユーザーを認証できることを確認するには、これまで使用されていないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
ssh localhost -l domain\\username



id -u
```

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

```
ls /tmp/krb5cc_uid
```

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

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

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

セントリファイ ダイレクトコントロール

Windowsドメインへの参加

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

```
su –

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

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

ドメインメンバーシップの確認

Delivery Controllerでは、WindowsまたはLinuxにかかわらず、すべてのVDAマシンがActive Directoryにコンピューターオブジェクトを持っている必要があります。Centrifyに参加しているLinuxマシンがドメイン上にあることを確認するには：

```
su –



adinfo
```

Joined to domainの値が有効であり、**CentrifyDC mode**が**connected**を返すことを確認します。モードが開始状態のままである場合、Centrifyクライアントはサーバー接続または認証の問題を抱えています。

より包括的なシステムおよび診断情報は、以下を使用して利用できます：

```
adinfo --sysinfo all
```

```
adinfo --diag
```

さまざまなActive DirectoryおよびKerberosサービスへの接続をテストするには：

```
adinfo --test
```

SSSD

Kerberosの構成

Kerberosをインストールするには、次のコマンドを実行します：

```
sudo apt-get install krb5-user
```

Kerberosを構成するには、ルートとして **/etc/krb5.conf** を開き、次の設定を行います：

```
[libdefaults]

 default_realm = REALM

 dns_lookup_kdc = false

[realms]

 REALM = {

  admin_server = domain-controller-fqdn

  kdc = domain-controller-fqdn
```

```
 }

[domain_realm]

 domain-dns-name = REALM

 .domain-dns-name = REALM
```

このコンテキストでの`domain-dns-name`プロパティは、`example.com`のようなDNSドメイン名です。`REALM`は、`EXAMPLE.COM`のような大文字のKerberosレルム名です。

ドメインに参加する

SSSDは、IDプロバイダーとしてActive Directoryを、認証にKerberosを使用するように構成する必要があります。しかし、SSSDはドメイン参加やシステムキータブファイルの管理のためのADクライアント機能を提供しません。代わりに、`adcli`、`realmd`、または`Samba`を使用できます。

> **注:**
>
> このセクションでは、`adcli`と`Samba`に関する情報のみを提供します。

- **`adcli`を使用してドメインに参加する場合は、次の手順を完了してください。**

1. `adcli`をインストールします。

   ```
   sudo apt-get install adcli
   ```

2. `adcli`を使用してドメインに参加します。

   古いシステムキータブファイルを削除し、以下を使用してドメインに参加します。

```
su -



rm -rf /etc/krb5.keytab



adcli join domain-dns-name -U user -H hostname-fqdn
```

userは、マシンをドメインに追加する権限を持つドメインユーザーです。**hostname-fqdn**は、マシンのFQDN形式のホスト名です。

-Hオプションは、Linux VDAが必要とするhost/hostname-fqdn@REALM形式でSPNを生成するために`adcli`に必要です。

3. システムキータブを確認します。

   `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。システムキータブファイルが作成されたことを確認するための最良の代替策は次のとおりです。

   ```
   sudo klist -ket
   ```

   各キーのタイムスタンプが、マシンがドメインに参加した時刻と一致することを確認します。

- **`Samba`を使用してドメインに参加する場合は、次の手順を完了してください。**

1. パッケージをインストールします。

   ```
   sudo apt-get install samba
   ```

2. `Samba`を構成します。

   /etc/samba/smb.conf を開き、以下の設定を行います。

   ```
   [global]



   workgroup = WORKGROUP
   ```

```
security = ADS

realm = REALM

client signing = yes

client use spnego = yes

kerberos method = secrets and keytab
```

WORKGROUP は **REALM** の最初のフィールドであり、REALM は大文字のKerberosレルム名です。

3. `Samba` を使用してドメインに参加します。

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

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

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

SSSDのセットアップ

必要なパッケージのインストールまたは更新:

必要なSSSDおよび構成パッケージがまだインストールされていない場合は、インストールします。

```
sudo apt-get install sssd
```

パッケージがすでにインストールされている場合は、更新が推奨されます。

```
sudo apt-get install --only-upgrade sssd
```

注:

デフォルトでは、Ubuntuのインストールプロセスは **nsswitch.conf** とPAMログインモジュールを自動的に構成します。

SSSDの構成

SSSDデーモンを開始する前に、SSSD構成の変更が必要です。SSSDの一部のバージョンでは、**/etc/sssd/sssd.conf** 構成ファイルがデフォルトでインストールされていないため、手動で作成する必要があります。rootとして、**/etc/sssd/sssd.conf** を作成または開き、以下の設定を行います。

```
[sssd]


services = nss, pam


config_file_version = 2


domains = domain-dns-name


[domain/domain-dns-name]


id_provider = ad


```

```
access_provider = ad

auth_provider = krb5

krb5_realm = REALM

# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer than
14 days

krb5_renewable_lifetime = 14d

# Set krb5_renew_interval to lower value if TGT ticket lifetime is shorter
than 2 hours

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true
```

```
override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

注:

ldap_id_mapping は **true** に設定されており、SSSD自体がWindows SIDをUnix UIDにマッピングする処理を行います。そうでない場合、Active DirectoryはPOSIX拡張機能を提供できる必要があります。PAMサービス `ctxhdx` は ad_gpo_map_remote_interactive に追加されます。

このコンテキストにおける `domain-dns-name` プロパティは、`example.com` のようなDNSドメイン名です。`REALM` は、`EXAMPLE.COM` のような大文字のKerberosレルム名です。NetBIOSドメイン名を構成する必要はありません。

ヒント:

これらの構成設定の詳細については、`sssd.conf` および `sssd-ad` のmanページを参照してください。

SSSDデーモンは、構成ファイルが所有者による読み取り権限のみを持つことを要求します。

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

SSSDデーモンを開始する

SSSDデーモンを今すぐ開始し、マシンの起動時にデーモンが開始されるようにするには、次のコマンドを実行します。

```
sudo systemctl start sssd



sudo systemctl enable sssd
```

PAM構成

次のコマンドを実行し、**SSS authentication**と**Create home directory on login**オプションが選択されていることを確認します。

```
sudo pam-auth-update
```

ドメインメンバーシップを確認する

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

- ドメインメンバーシップを確認するために`adcli`を使用する場合は、`sudo adcli info domain-dns-name`コマンドを実行してドメイン情報を表示します。
- ドメインメンバーシップを確認するために`Samba`を使用する場合は、`sudo net ads testjoin`コマンドを実行してマシンがドメインに参加していることを確認し、`sudo net ads info`コマンドを実行して追加のドメインおよびコンピューターオブジェクト情報を確認します。

Kerberos構成を確認する

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

```
sudo klist -ke
```

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

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

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

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

```
sudo klist
```

ユーザー認証を確認する

SSSD は、デーモンで認証を直接テストするためのコマンドラインツールを提供しておらず、PAM を介してのみ実行できます。

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

```
ssh localhost -l domain\\username



id -u



klist



exit
```

klist コマンドによって返された Kerberos チケットがそのユーザーに対して正しく、期限切れになっていないことを確認します。

ルートユーザーとして、以前の **id -u** コマンドによって返された uid に対応するチケットキャッシュファイルが作成されたことを確認します。

```
ls /tmp/krb5cc_uid
```

同様のテストは、KDE または Gnome Display Manager にログオンすることで実行できます。ドメイン参加の検証後、手順 6: Linux VDA をインストールする に進みます。

PBIS

必要な PBIS パッケージをダウンロードする

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
```

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
```

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
```

Windows ドメインに参加する

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

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

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

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

ドメインメンバーシップを確認する

Delivery Controller では、すべての VDA マシン (Windows および Linux VDA) が Active Directory にコンピューターオブジェクトを持っている必要があります。PBIS に参加している Linux マシンがドメイン上にあることを確認するには:

```
/opt/pbis/bin/domainjoin-cli query
```

マシンがドメインに参加している場合、このコマンドは現在参加しているADドメインとOUに関する情報を返します。そうでない場合は、ホスト名のみが表示されます。

ユーザー認証の確認

PBISがPAMを介してドメインユーザーを認証できることを確認するには、以前使用したことのないドメインユーザーアカウントを使用してLinux VDAにログオンします。

```
sudo ssh localhost -l domain\\user



id -u
```

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

```
ls /tmp/krb5cc_uid
```

セッションを終了します。

```
exit
```

ドメイン参加の検証後、手順6：Linux VDAのインストールに進みます。

手順4：前提条件として.NET Core Runtimeをインストールする

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

- 1912 LTSRの初回リリース、CU1、およびCU2の場合は、.NET Core Runtime 2.1をインストールします。

- CU3以降のリリースでは、.NET Core Runtime 3.1をインストールします。

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

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

手順5：Linux VDAパッケージをダウンロードする

Citrix Virtual Apps and Desktopsダウンロードページにアクセスします。適切なバージョンのCitrix Virtual Apps and Desktopsを展開し、**Components**をクリックして、お使いのLinuxディストリビューションに一致するLinux VDAパッケージをダウンロードします。

手順6：Linux VDAをインストールする

ステップ 6a: Linux VDA をインストールする

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

Ubuntu 18.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu18.04_amd64.deb
```

Ubuntu 16.04 の場合:

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu16.04_amd64.deb
```

Ubuntu 18.04 用 Debian 依存関係リスト:

```
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
```

Ubuntu 16.04 用 Debian 依存関係リスト:

```
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
```

注:

このバージョンの Linux VDA がサポートする Linux ディストリビューションと Xorg バージョンのマトリックスについては、「システム要件」を参照してください。

ステップ 6b: Linux VDA をアップグレードする (オプション)

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

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

ステップ 7: NVIDIA GRID ドライバーをインストールする

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

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

- シトリックス ハイパーバイザー
- ヴイエムウェア イーエスエックス

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

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

ステップ8：Linux VDAを構成する

パッケージをインストールした後、ctxsetup.shスクリプトを実行してLinux VDAを構成する必要があります。変更を加える前に、スクリプトは環境を検証し、すべての依存関係がインストールされていることを確認します。必要に応じて、いつでもスクリプトを再実行して設定を変更できます。

スクリプトは、プロンプトに従って手動で実行することも、事前設定された応答で自動的に実行することもできます。続行する前に、スクリプトに関するヘルプを確認してください。

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

プロンプトによる構成

プロンプトによる質問で手動構成を実行します。

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

自動構成

自動インストールの場合、セットアップスクリプトに必要なオプションは環境変数で提供できます。必要なすべての変数が存在する場合、スクリプトはユーザーに情報を要求しないため、スクリプトによるインストールプロセスが可能になります。

サポートされている環境変数には以下が含まれます。

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y \| N** – Linux VDAは、DNS CNAMEレコードを使用してDelivery Controller名を指定することをサポートしています。デフォルトではNに設定されています。
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'** – Linux VDAは、Delivery Controllerへの登録に使用するDelivery Controllerの完全修飾ドメイン名（FQDN）のスペース区切りリストを必要とします。少なくとも1つのFQDNまたはCNAMEエイリアスを指定する必要があります。
- **CTX_XDL_VDA_PORT=port-number** – Linux VDAは、TCP/IPポート（デフォルトではポート80）を介してDelivery Controllerと通信します。
- **CTX_XDL_REGISTER_SERVICE=Y \| N** – Linux Virtual Desktopサービスは、マシンの起動後に開始されます。デフォルトではYに設定されています。
- **CTX_XDL_ADD_FIREWALL_RULES=Y \| N** – Linux Virtual Desktop サービスは、システムファイアウォールを介した受信ネットワーク接続を許可する必要があります。Linux Virtual Desktop のシステムファイアウォールで、必要なポート（デフォルトでポート80と1494）を自動的に開くことができます。デフォルトではYに設定されています。
- **CTX_XDL_AD_INTEGRATION=1 \| 2 \| 3 \| 4 \|5** – Linux VDA は、Delivery Controller と認証するために Kerberos 構成設定を必要とします。Kerberos 構成は、システムにインストールおよび構成されている Active Directory 統合ツールから決定されます。使用するサポートされている Active Directory 統合方法を指定します。
  - 1 – サンバ・ウィンバインド
  - 2 – クエスト認証サービス
  - 3 – セントリファイ ダイレクトコントロール
  - 4 – SSSD
  - 5 – PBIS
- **CTX_XDL_HDX_3D_PRO=Y \| N** – Linux VDA は、リッチグラフィックアプリケーションの仮想化を最適化するために設計された GPU アクセラレーション技術のセットである HDX 3D

Pro をサポートしています。HDX 3D Pro が選択されている場合、VDA は VDI デスクトップ（シングルセッション）モード（つまり、CTX_XDL_VDI_MODE=Y）用に構成されます。

- **CTX_XDL_VDI_MODE=Y \| N** – マシンを専用デスクトップ配信モデル（VDI）として構成するか、ホスト型共有デスクトップ配信モデルとして構成するか。HDX 3D Pro 環境では、この変数をYに設定します。この変数はデフォルトでNに設定されています。
- **CTX_XDL_SITE_NAME=dns-name** – Linux VDA は DNS を介して LDAP サーバーを検出します。DNS 検索結果をローカルサイトに制限するには、DNS サイト名を指定します。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** – Linux VDA は DNS にクエリを実行して LDAP サーバーを検出します。DNS が LDAP サービスレコードを提供できない場合、LDAP ポートを含む LDAP FQDN のスペース区切りリストを提供できます。たとえば、ad1.mycompany.com:389 です。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_SEARCH_BASE=search-base-set** – Linux VDA は、Active Directory ドメインのルートに設定された検索ベース（例: DC=mycompany,DC=com）を介して LDAP にクエリを実行します。ただし、検索パフォーマンスを向上させるために、検索ベース（例: OU=VDI,DC=mycompany,DC=com）を指定できます。この変数はデフォルトで **\<none\>** に設定されています。
- **CTX_XDL_FAS_LIST='list-fas-servers'** – Federated Authentication Service (FAS) サーバーは AD グループポリシーを介して構成されます。Linux VDA は AD グループポリシーをサポートしていないため、代わりにセミコロン区切りの FAS サーバーのリストを提供できます。シーケンスは AD グループポリシーで構成されているものと同じである必要があります。サーバーアドレスが削除された場合は、その空白を **\<none\>** テキスト文字列で埋め、サーバーアドレスのシーケンスを変更せずに保持します。
- **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 Runtime 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 の構成が完了したときに Linux VDA サービスが開始されるかどうか。デフォルトではYに設定されています。

環境変数を設定し、構成スクリプトを実行します。

```
export CTX_XDL_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
```

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

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

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \


CTX_XDL_DDC_LIST='list-ddc-fqdns' \


CTX_XDL_VDA_PORT=port-number \


CTX_XDL_REGISTER_SERVICE=Y|N \


CTX_XDL_ADD_FIREWALL_RULES=Y|N \


CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \


CTX_XDL_HDX_3D_PRO=Y|N \


CTX_XDL_VDI_MODE=Y|N \
```

```
CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST='list-ldap-servers' \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST='list-fas-servers' \

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

CTX_XDL_START_SERVICE=Y|N \

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

構成の変更を削除する

シナリオによっては、Linux VDAパッケージをアンインストールせずに、**ctxsetup.sh**スクリプトによって行われた構成の変更を削除する必要がある場合があります。

続行する前に、このスクリプトに関するヘルプを確認してください。

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

構成の変更を削除するには：

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

重要:

このスクリプトは、データベースからすべての構成データを削除し、Linux VDAを動作不能にします。

構成ログ

ctxsetup.shおよび**ctxcleanup.sh**スクリプトは、コンソールにエラーを表示し、追加情報は構成ログファイル**/tmp/xdl.configure.log**に書き込まれます。

変更を有効にするには、Linux VDAサービスを再起動します。

Linux VDAソフトウェアをアンインストールする

Linux VDAがインストールされているかどうかを確認し、インストールされているパッケージのバージョンを表示するには：

```
dpkg -l xendesktopvda
```

詳細情報を表示するには：

```
apt-cache show xendesktopvda
```

Linux VDAソフトウェアをアンインストールするには：

```
dpkg -r xendesktopvda
```

注:

Linux VDAソフトウェアをアンインストールすると、関連するPostgreSQLおよびその他の構成データが削除されます。ただし、Linux VDAのインストール前にセットアップされたPostgreSQLパッケージおよびその他の依存パッケージは削除されません。

ヒント:

> このセクションの情報には、PostgreSQLを含む依存パッケージの削除は含まれていません。

ステップ9：XDPingを実行する

Linux VDA環境における一般的な構成の問題を確認するために、コマンドラインユーティリティであるLinux `XDPing` ツールを提供しています。サポートされているLinuxディストリビューションを実行している任意のコンピューターに `XDPing` パッケージをインストールできます。`XDPing` は、Linux VDAパッケージがコンピューターにインストールされている必要はありません。このツールの詳細については、Knowledge Centerの記事CTX202015を参照してください。

ステップ10：Linux VDAを実行する

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

Linux VDAを開始する：

Linux VDAサービスを開始するには：

```
sudo systemctl start ctxhdx



sudo systemctl start ctxvda
```

Linux VDAを停止する：

Linux VDAサービスを停止するには：

```
sudo systemctl stop ctxvda



sudo systemctl stop ctxhdx
```

> **注:**

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

> **Linux VDAを再起動する：**

Linux VDAサービスを再起動するには：

```
sudo systemctl stop ctxvda



sudo systemctl restart ctxhdx



sudo systemctl restart ctxvda
```

Linux VDAのステータスを確認する：

Linux VDAサービスの実行ステータスを確認するには：

```
sudo systemctl status ctxvda



sudo systemctl status ctxhdx
```

手順 11：Citrix Virtual AppsまたはCitrix Virtual Desktops™でマシンカタログを作成する

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

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

- オペレーティングシステムには、以下を選択します。
    - ホストされた共有デスクトップ配信モデルの場合は、**Multi-session OS**オプション。

    - VDI専用デスクトップ配信モデルの場合は、**Single-session OS**オプション。

- 同じマシンカタログ内でLinux VDAマシンとWindows VDAマシンを混在させないでください。

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

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

手順 12：Citrix Virtual Apps™またはCitrix Virtual Desktopsでデリバリーグループを作成する

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

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

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

マシンカタログとデリバリーグループの作成方法については、Citrix Virtual Apps and Desktops 7 1912 LTSR を参照してください。

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

7.18 リリース以降、MCS を使用して Linux VM を作成できます。

サポートされているハイパーバイザー

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

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

Citrix Hypervisor で MCS を使用して Linux VM を作成する

ステップ 1: マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれます。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。

ステップ 1a: Citrix VM Tools をインストールする

各 VM が xe CLI または XenCenter を使用できるようにするには、Citrix VM Tools をテンプレート VM にインストールする必要があります。ツールがインストールされていないと、VM のパフォーマンスが低下する可能性があります。ツールがないと、次のいずれも実行できません。

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

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

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

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

   RHEL/CentOS の場合:

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

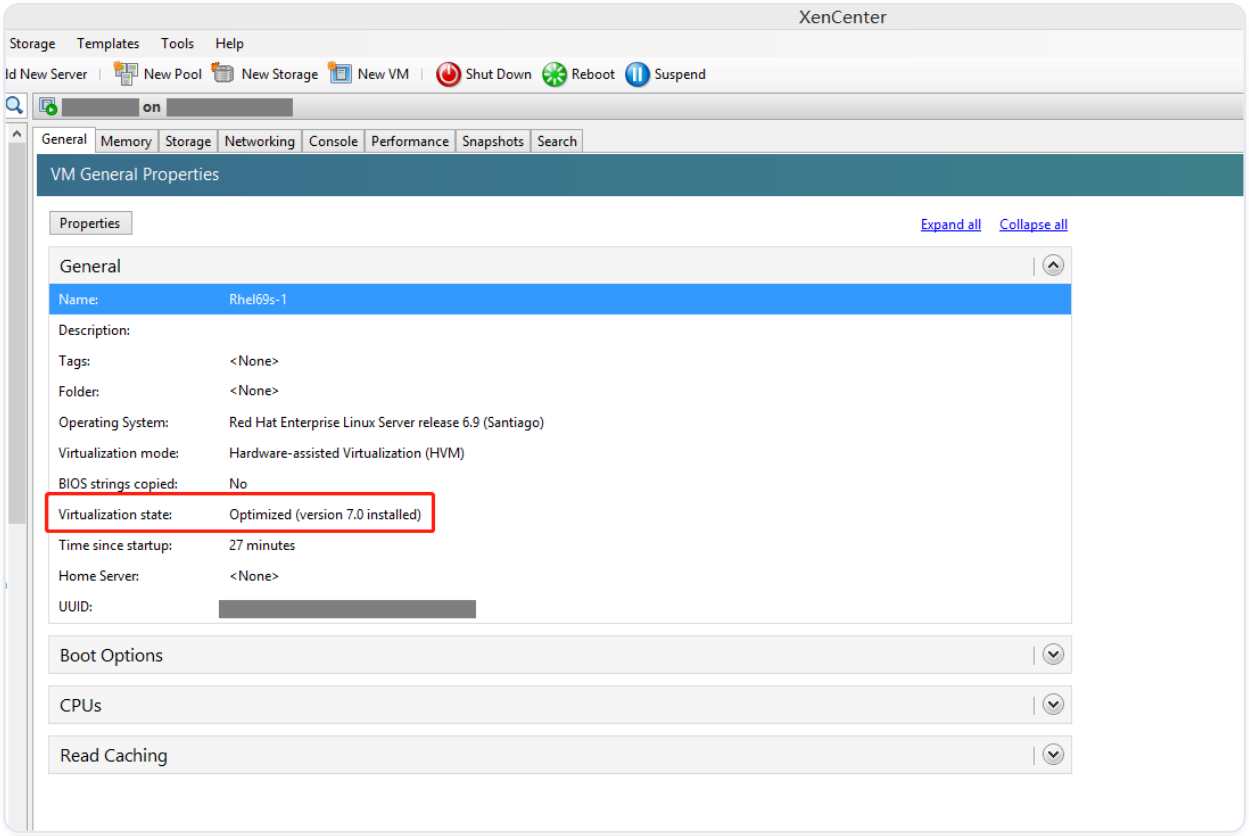
   Ubuntu の場合:

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

   SUSE 12 の場合:

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

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



ステップ 1b: テンプレート VM に Linux VDA パッケージをインストールする

> **注:**

現在実行中の VDA をテンプレート VM として使用する場合は、この手順を省略します。

テンプレート VM に Linux VDA パッケージをインストールする前に、.NET Core Runtime 3.1 をインストールします。詳しくは、「インストール概要」を参照してください。

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

RHEL/CentOS の場合:

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

Ubuntu の場合:

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



apt-get install -f
```

SUSE 12 の場合:

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

ステップ 1c: tdb-tools パッケージをインストールするためのリポジトリを有効にする

RHEL 7サーバーの場合:

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

RHEL 7ワークステーションの場合:

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

ステップ1d: ntfs-3gを含むEPELリポジトリをインストールする

RHEL 6/CentOS 6、RHEL 7/CentOS 7にEPELリポジトリをインストールします。これにより、後でdeploymcs.shを実行したときに、その中に含まれるntfs-3gパッケージがインストールされます。

ステップ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、コンパイルして、ntfs-3gを手動でインストールします。

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3gパッケージへのパスを入力します。

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

5. ntfs-3gをインストールします。

   ```
   ./configure

   make

   make install
   ```

ステップ1f: ランタイム環境をセットアップする

deploymcs.shを実行する前に、以下を実行します。

- **/etc/xdl/mcs/mcs.conf**内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、必要に応じて設定できる変数です。

- `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の同等のものと同じになります。ただし、`dns`および`AD_INTEGRATION`変数は引き続き構成する必要があります。デフォルト値はNで、これはMCSで作成されたマシン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 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`: ワークグループ名を設定します。これは、ADで構成されている場合はNetBIOS名（大文字と小文字を区別）です。それ以外の場合は、デフォルトでドメイン名になります。

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

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

  たとえば、レジストリ値をそれぞれ書き込んだり更新したりするには、次のコマンドラインを **/etc/xdl/mcs/mcs_local_setting.reg** ファイルに追加します。

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

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

ステップ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VMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する /etc/krb5.conf、/etc/samba/smb.conf、および /etc/sssd/sssd.conf ファイルをカスタマイズします。

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

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

注： テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$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VMをシャットダウンします。マスターイメージのスナップショットを作成して名前を付けます。

ステップ2：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。必要に応じて、その他の構成タスクを実行します。詳しくは、「Studioを使用してマシンカタログを作成する」を参照してください。

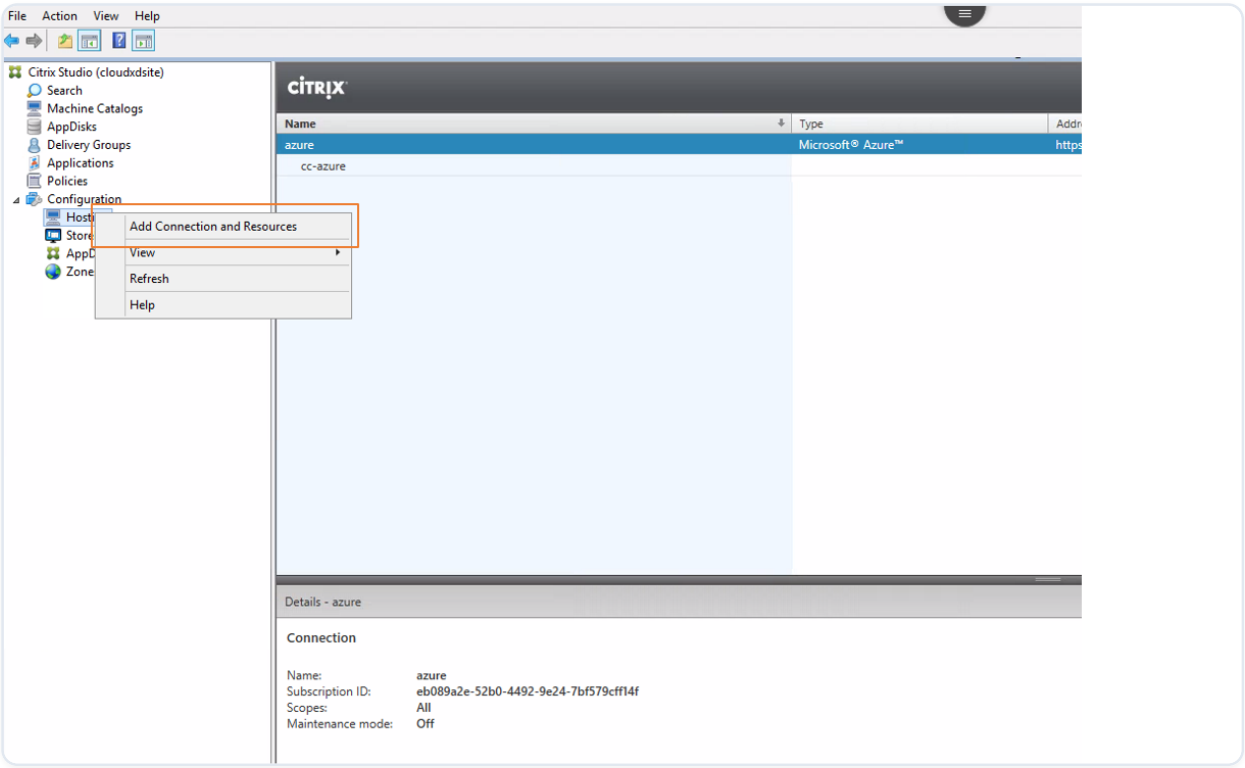
ステップ3: デリバリーグループを作成する

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

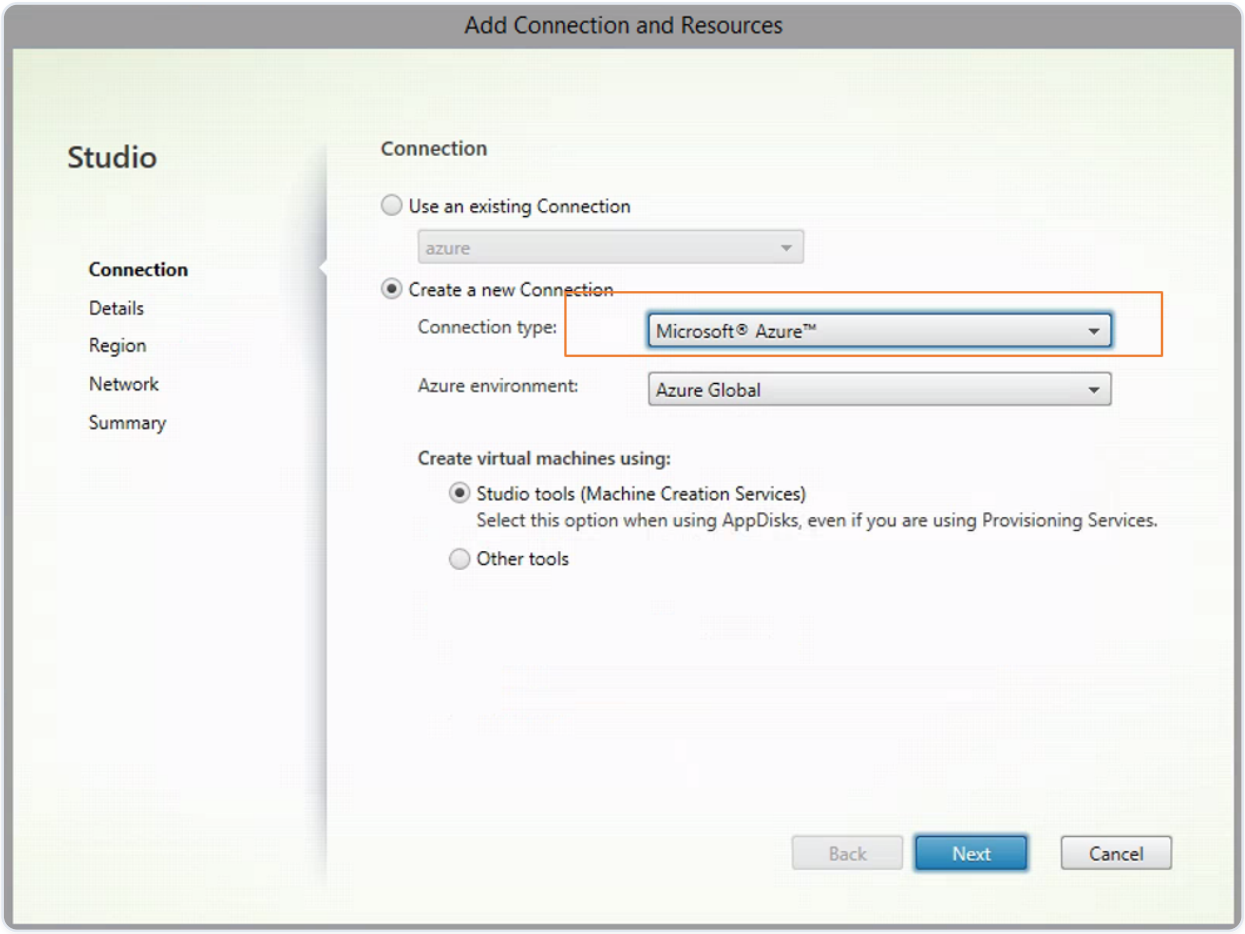
MCSを使用してAzure上にLinux VMを作成する

ステップ1：Citrix StudioでAzureへのホスティング接続を作成する

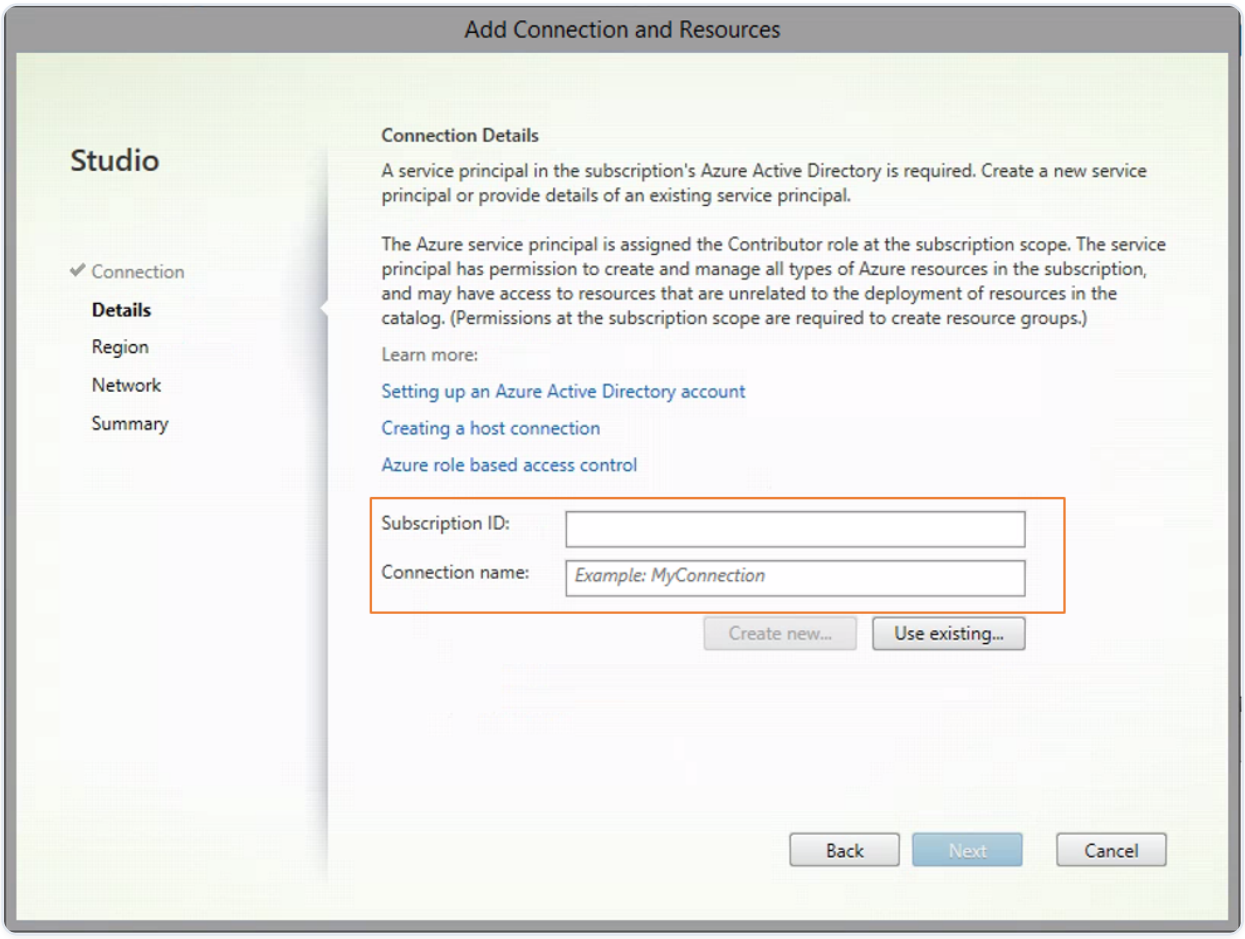
1. Citrix Studioで、**構成** > **ホスティング** > **接続とリソースの追加**を選択して、Azureへの接続を作成します。



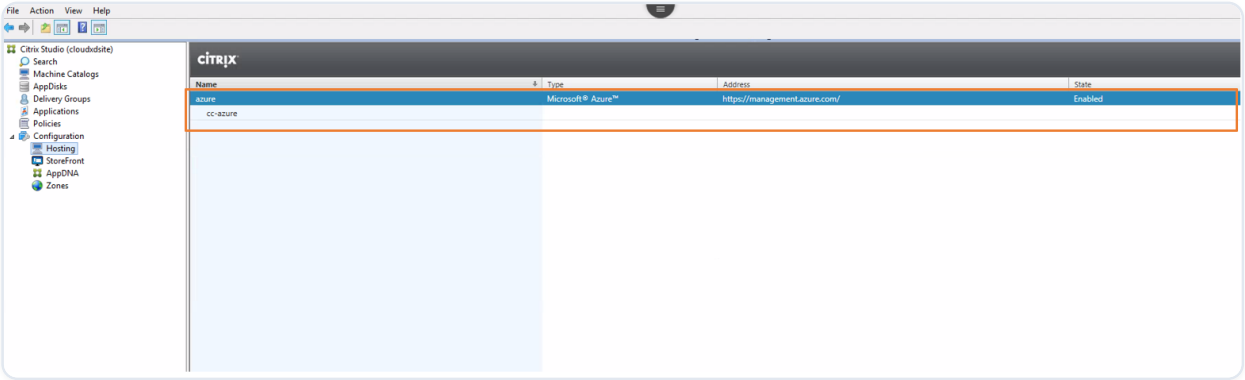
2. 接続の種類としてMicrosoft Azureを選択します。



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。



ホスティングペインに新しい接続が表示されます。



ステップ2：テンプレートVMでマスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。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。

ステップ2a：Ubuntu 18.04のcloud-initを構成する

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

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

/etc/cloud/cloud.cfg ファイルの **system_info** セクションに以下の行が存在することを確認してください。

```
system_info:

   network:

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

ステップ 2b: テンプレート VM に Linux VDA パッケージをインストールする

注:

現在実行中の VDA をテンプレート VM として使用する場合は、この手順を省略してください。

テンプレート VM に Linux VDA パッケージをインストールする前に、.NET Core Runtime 3.1 をインストールしてください。詳しくは、「インストール概要」を参照してください。

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

RHEL/CentOS の場合:

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

Ubuntu の場合:

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



apt-get install -f
```

SUSE 12 の場合:

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

ステップ 2c: tdb-tools パッケージをインストールするためのリポジトリを有効にする

RHEL 7 サーバーの場合:

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

RHEL 7 ワークステーションの場合:

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

ステップ 2d: ntfs-3g を含む EPEL リポジトリをインストールする

RHEL 6/CentOS 6、RHEL 7/CentOS 7 に EPEL リポジトリをインストールすると、後で deploymcs.sh を実行したときに、その中に含まれる ntfs-3g パッケージがインストールされます。

ステップ 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、コンパイルして、ntfs-3g を手動でインストールしてください。

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3g パッケージへのパスを入力します。

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

5. ntfs-3g をインストールします。

```
./configure

make

make install
```

ステップ 2f: ランタイム環境をセットアップする

deploymcs.sh を実行する前に、以下を実行します。

- **/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。以下にいくつかの変数を示しますが、そのうち `dns` と `AD_INTEGRATION` は設定する必要があります。

  注： 変数を複数の値で設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `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 に設定されている場合、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 上の同等のものと同じになります。ただし、`dns` と `AD_INTEGRATION` の変数は引き続き構成する必要があります。デフォルト値は N です。これは、MCS で作成されたマシン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 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`: ワークグループ名を設定します。これは、AD で構成されている場合は NetBIOS 名（大文字と小文字を区別）です。それ以外の場合は、デフォルトでドメイン名になります。

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

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

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

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

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

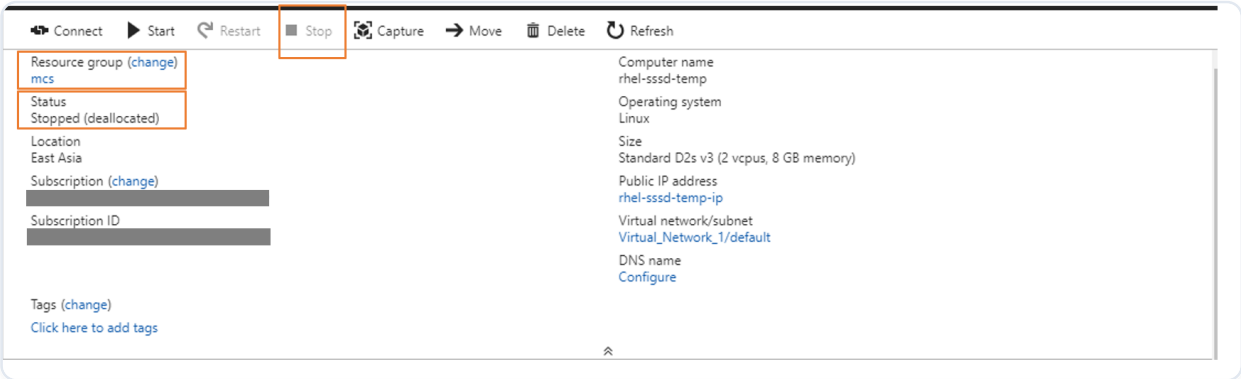
ステップ2g: マスターイメージを作成する

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VMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する /etc/krb5.conf、/etc/samba/smb.conf、および /etc/sssd/sssd.conf ファイルをカスタマイズします。

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

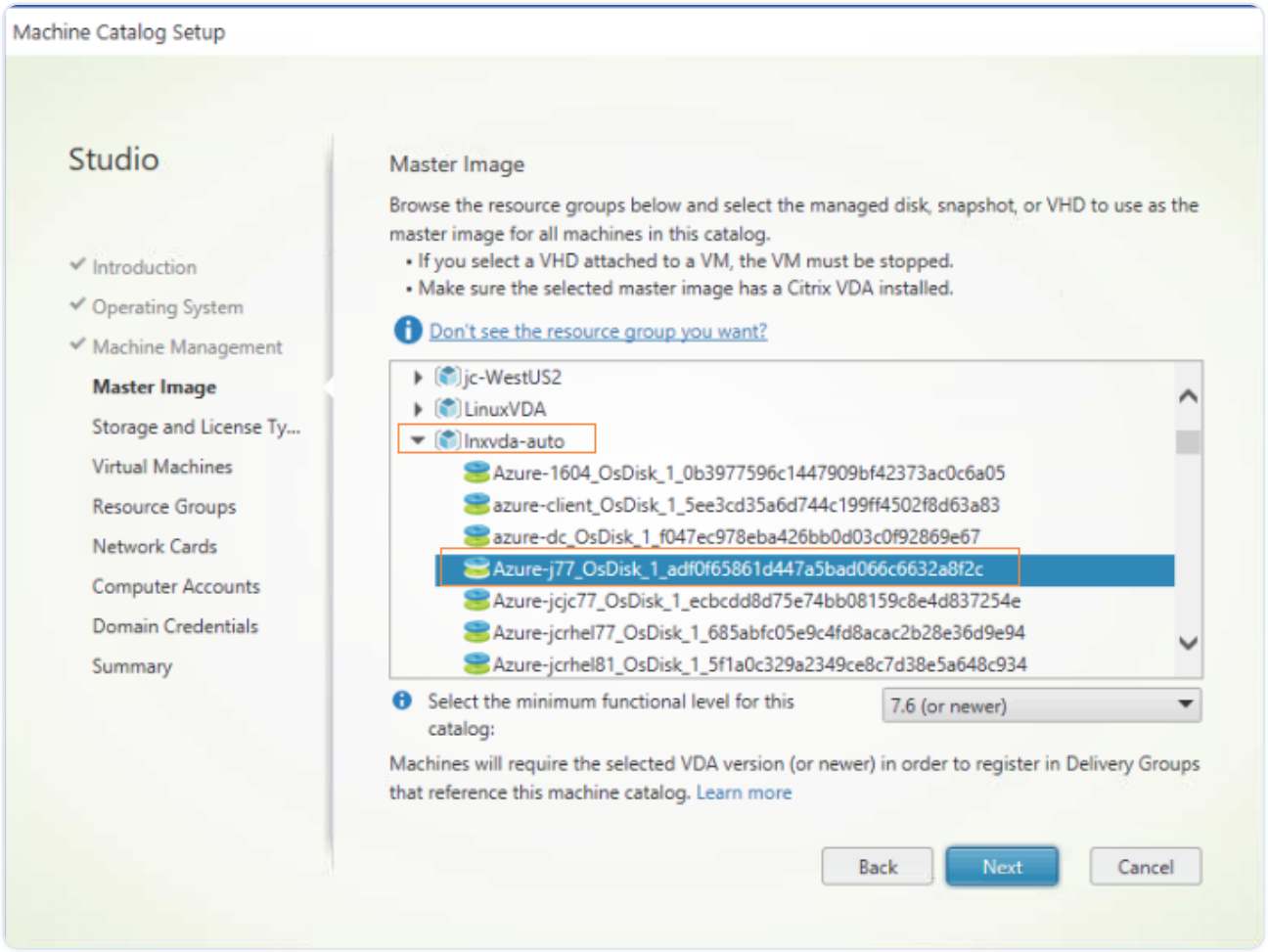
   SSSDユーザーの場合は、/etc/xdl/mcs/sssd.conf.tmpl、/etc/xdl/mcs/sssd_krb5.conf.tmpl、および /etc/xdl/mcs/sssd_smb.conf.tmpl テンプレートを更新します。

   注： テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用してください。
3. テンプレートVMにアプリケーションをインストールし、Azureポータルからテンプレートを VMをシャットダウンします。テンプレートVMの電源状態が **停止済み (割り当て解除済み)** であることを確認してください。ここでリソースグループの名前を覚えておいてください。Azureでマスターイメージを見つけるには、その名前が必要です。



ステップ3：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログで作成するVMの数を指定します。マシンカタログを作成する際、テンプレートVM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VMのVHDを見つけます。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、その他の構成タスクを実行します。詳細については、「Studioを使用してマシンカタログを作成する」を参照してください。

ステップ 4: デリバリーグループを作成する

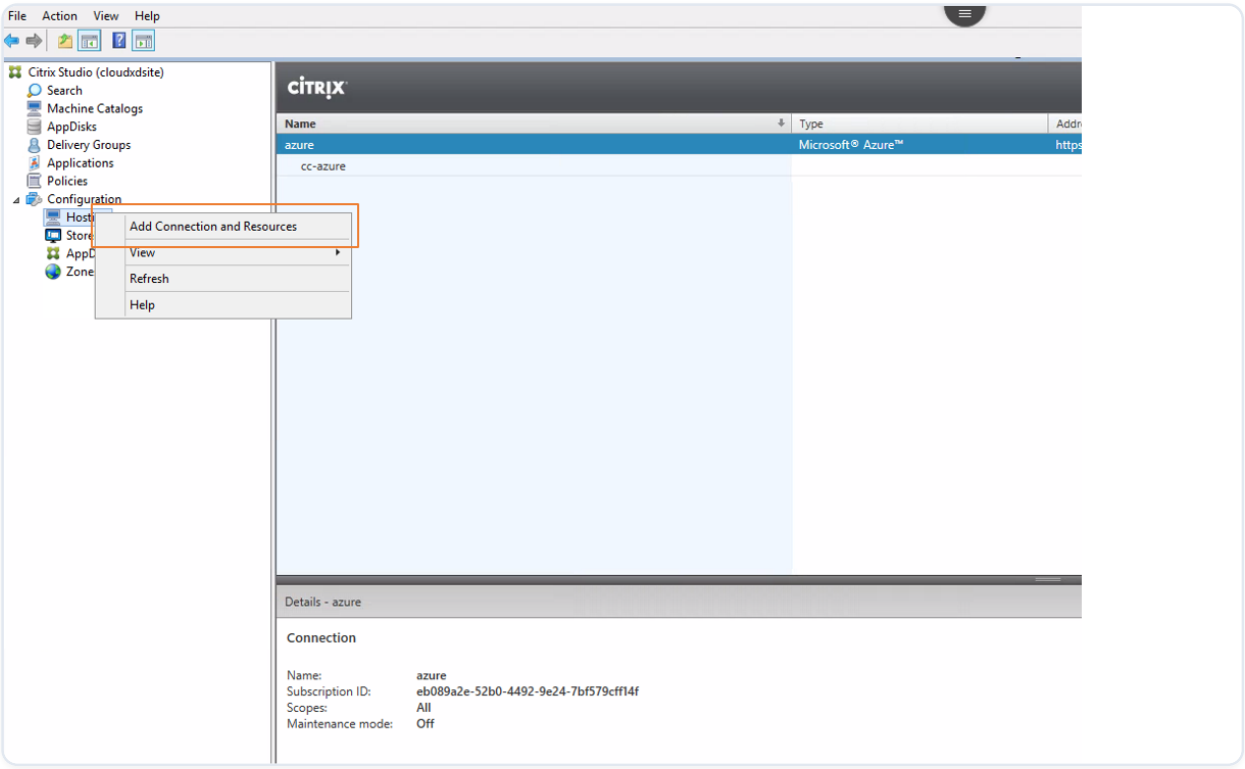
デリバリーグループは、1つ以上のマシンカタログから選択されたマシンの集合です。どのユーザーがそれらのマシンを使用できるか、およびそれらのユーザーが利用できるアプリケーションとデスクトップを指定します。詳細については、「デリバリーグループを作成する」を参照してください。

MCSを使用してVMware vSphere上にLinux VMを作成する

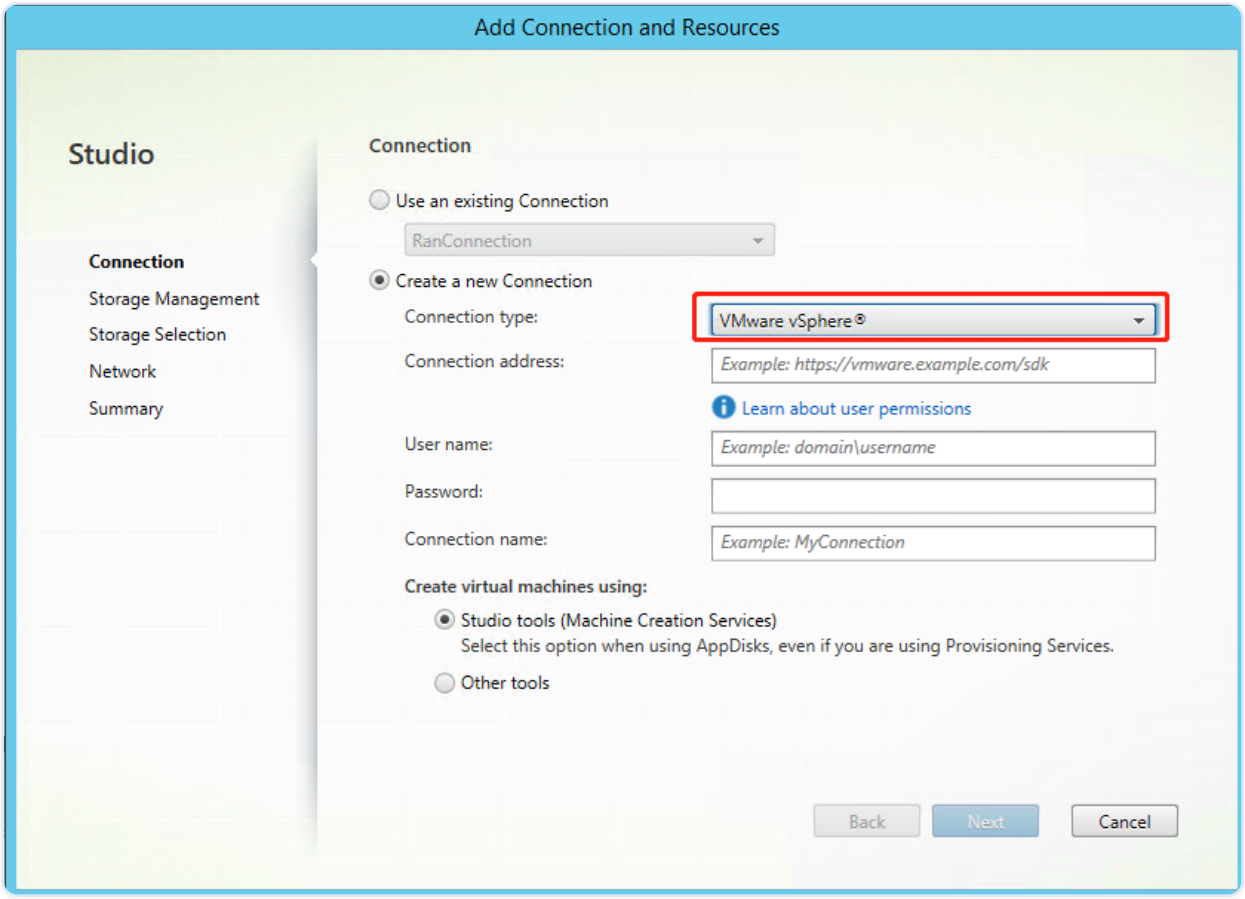
ステップ1: Citrix StudioでVMwareへのホスティング接続を作成する

1. vSphere環境にvCenter Serverをインストールします。詳細については、VMware vSphereを参照してください。

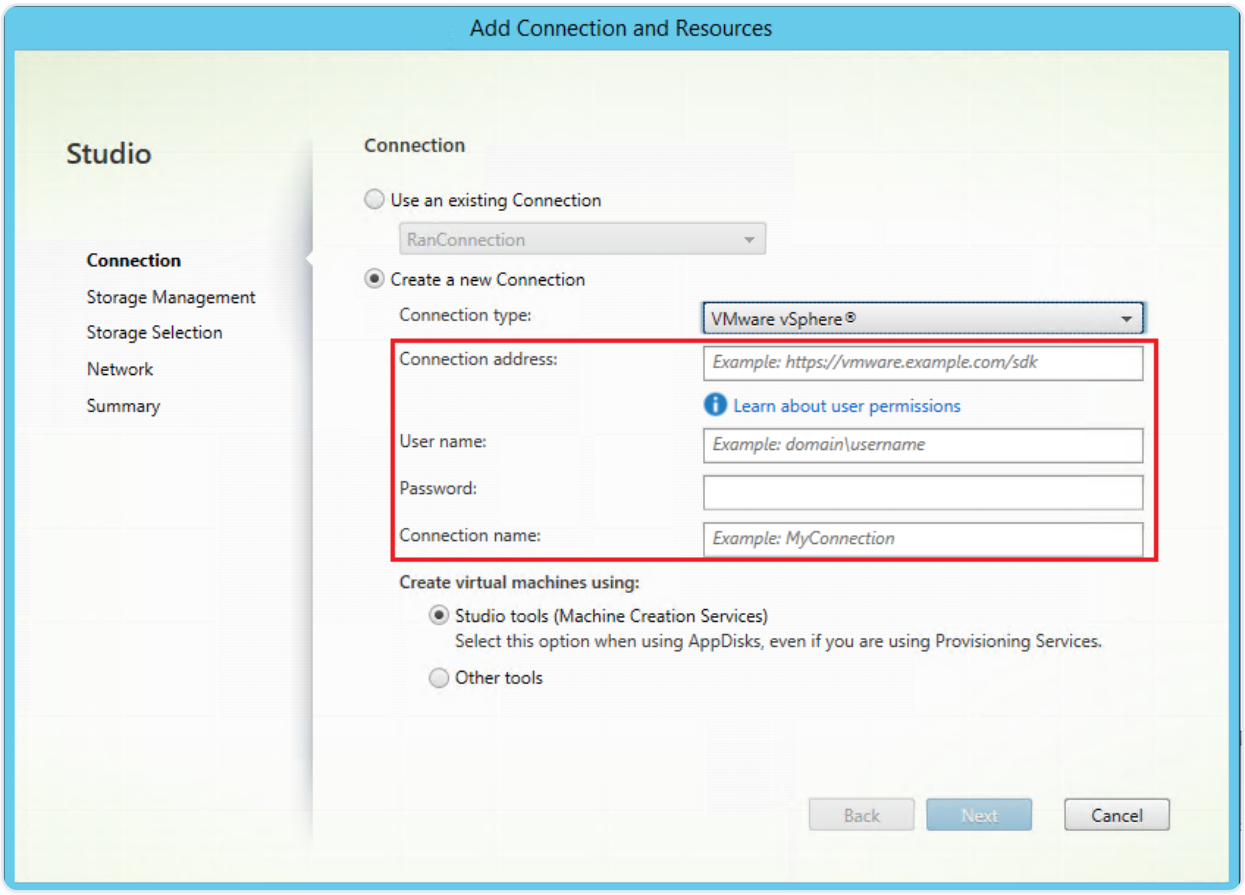
2. Citrix Studioで、**構成** > **ホスティング** > **接続およびリソースの追加**を選択して、VMware vSphereへの接続を作成します。



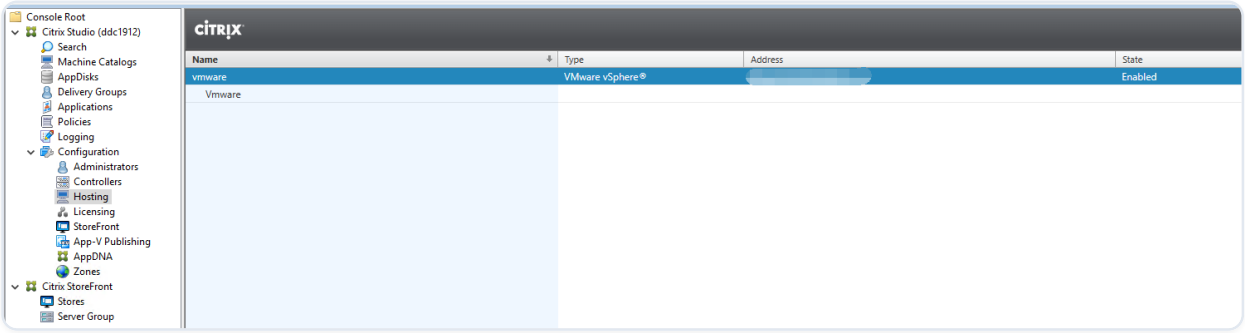
3. 接続の種類としてVMware vSphereを選択します。



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



ホスティングペインに新しい接続が表示されます。



ステップ2：マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。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。

ステップ2a：テンプレートVMにLinux VDAパッケージをインストールする

> **注:**

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

> テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Core Runtime 3.1をインストールしてください。詳細については、インストール概要を参照してください。

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

RHEL/CentOSの場合:

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

Ubuntuの場合:

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



apt-get install -f
```

SUSE 12の場合:

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

ステップ2b: tdb-toolsパッケージをインストールするためのリポジトリを有効にする

RHEL 7サーバーの場合:

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

RHEL 7ワークステーションの場合:

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

ステップ2c: ntfs-3gを含むEPELリポジトリをインストールする

RHEL 6/CentOS 6、RHEL 7/CentOS 7にEPELリポジトリをインストールします。これにより、後でdeploymcs.shを実行したときに、その中に含まれるntfs-3gパッケージがインストールされます。

ステップ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、コンパイルして、ntfs-3gを手動でインストールします。

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3gパッケージへのパスを入力します。

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

5. ntfs-3gをインストールします。

   ```
   ./configure

   make

   make install
   ```

Step 2e: ランタイム環境をセットアップする

deploymcs.sh を実行する前に、以下を実行します。

- **/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。以下にいくつかの変数を示しますが、そのうち `dns` と `AD_INTEGRATION` は設定する必要があります。

  注： 変数に複数の値を設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `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 の対応するファイルと同じになります。ただし、`dns` と `AD_INTEGRATION` の変数は引き続き構成する必要があります。デフォルト値は N で、これは MCS で作成されたマシン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 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`: ワークグループ名を設定します。これは、AD で構成されている場合は NetBIOS 名（大文字と小文字を区別）です。それ以外の場合は、デフォルトでドメイン名になります。

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

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

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

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

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

Step 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 VM で、構成テンプレートを更新して、作成されたすべての VM 上の関連する /etc/krb5.conf、/etc/samba/smb.conf、および /etc/sssd/sssd.conf ファイルをカスタマイズします。

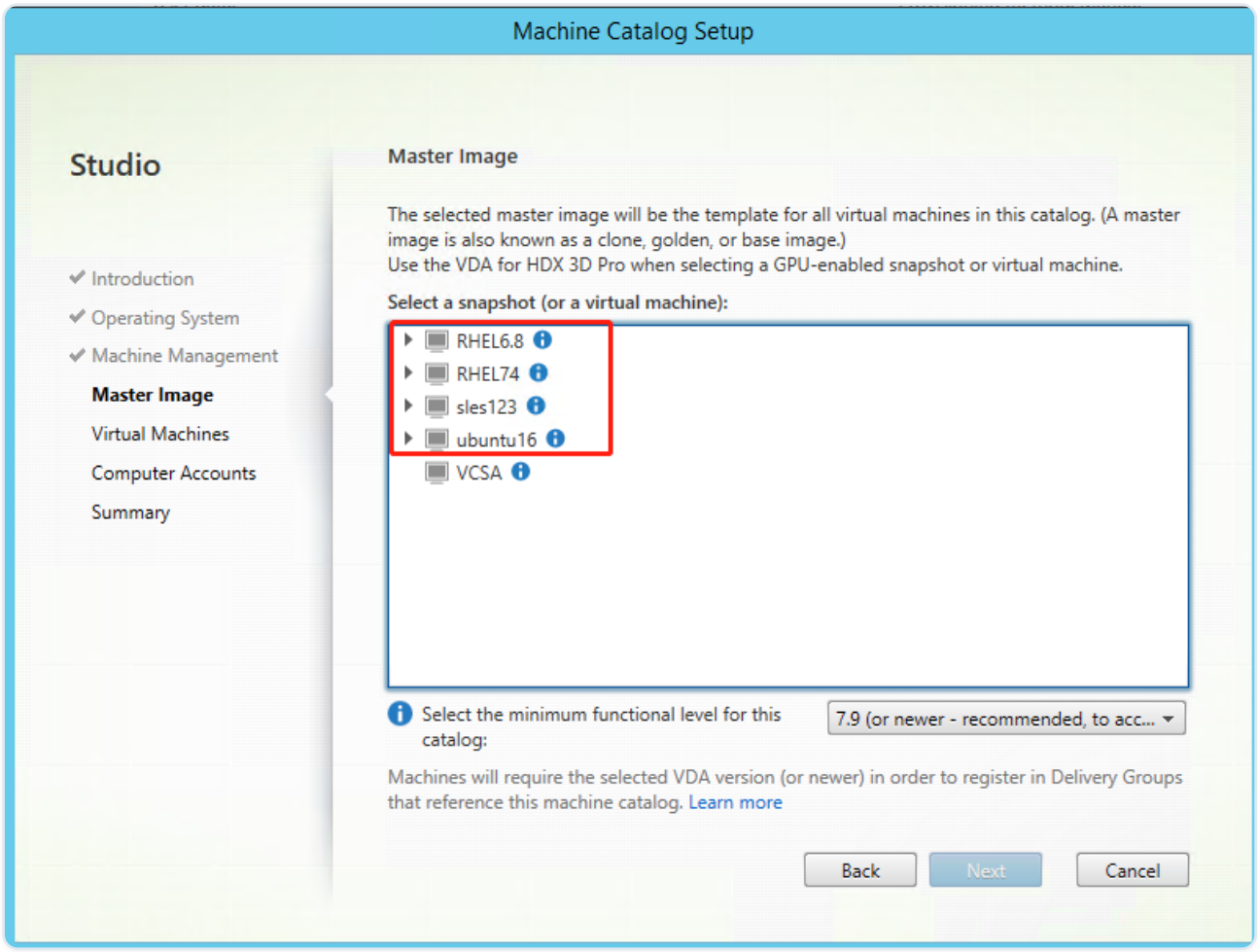
   Winbind ユーザーの場合は、/etc/xdl/mcs/winbind_krb5.conf.tmpl および /etc/xdl/mcs/winbind_smb.conf.tmpl テンプレートを更新します。

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

   注: テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用してください。
3. テンプレートVMへのアプリケーションのインストールが完了したら、VMwareからテンプレートVMをシャットダウンします。テンプレートVMのスナップショットを作成します。

ステップ3：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログに作成するVMの数を指定します。マシンカタログを作成する際に、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、その他の構成タスクを実行します。詳しくは、「Studioを使用してマシンカタログを作成する」を参照してください。

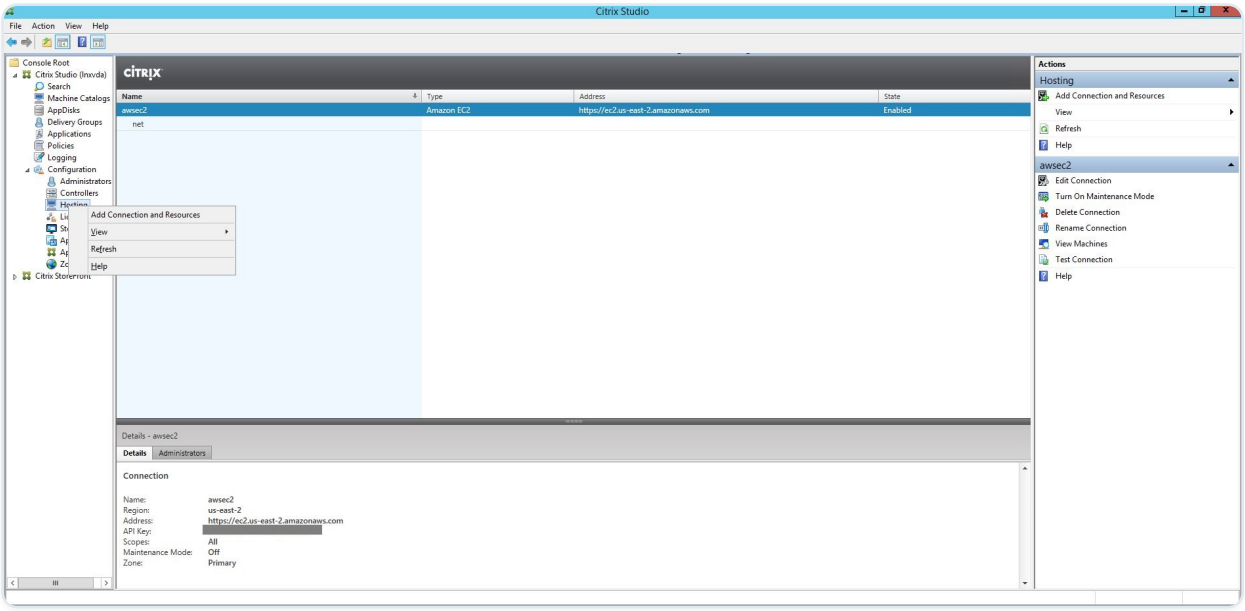
ステップ 4: デリバリーグループを作成する

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

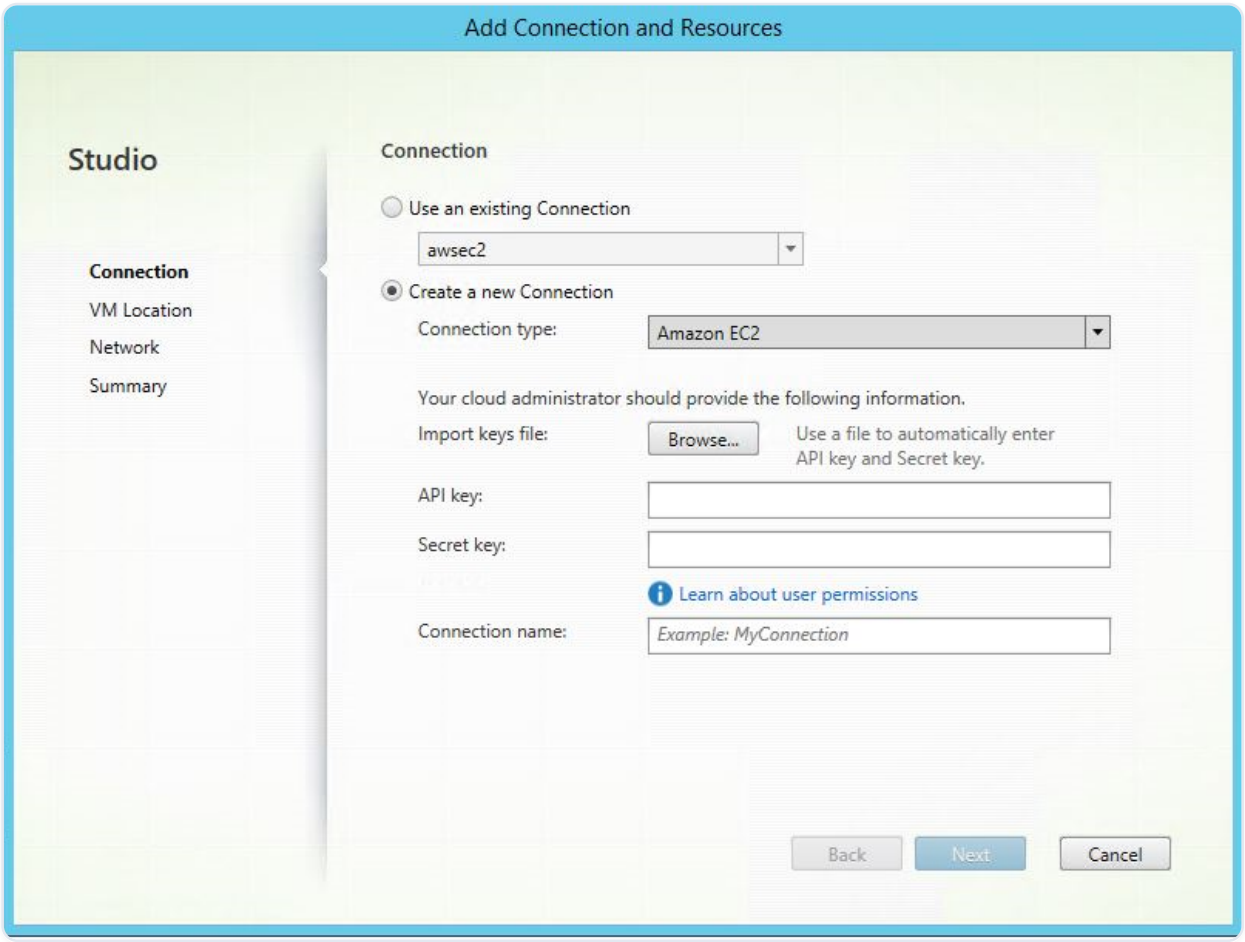
MCSを使用してAWS上にLinux VMを作成する

ステップ1：Citrix StudioでAWSへのホスティング接続を作成する

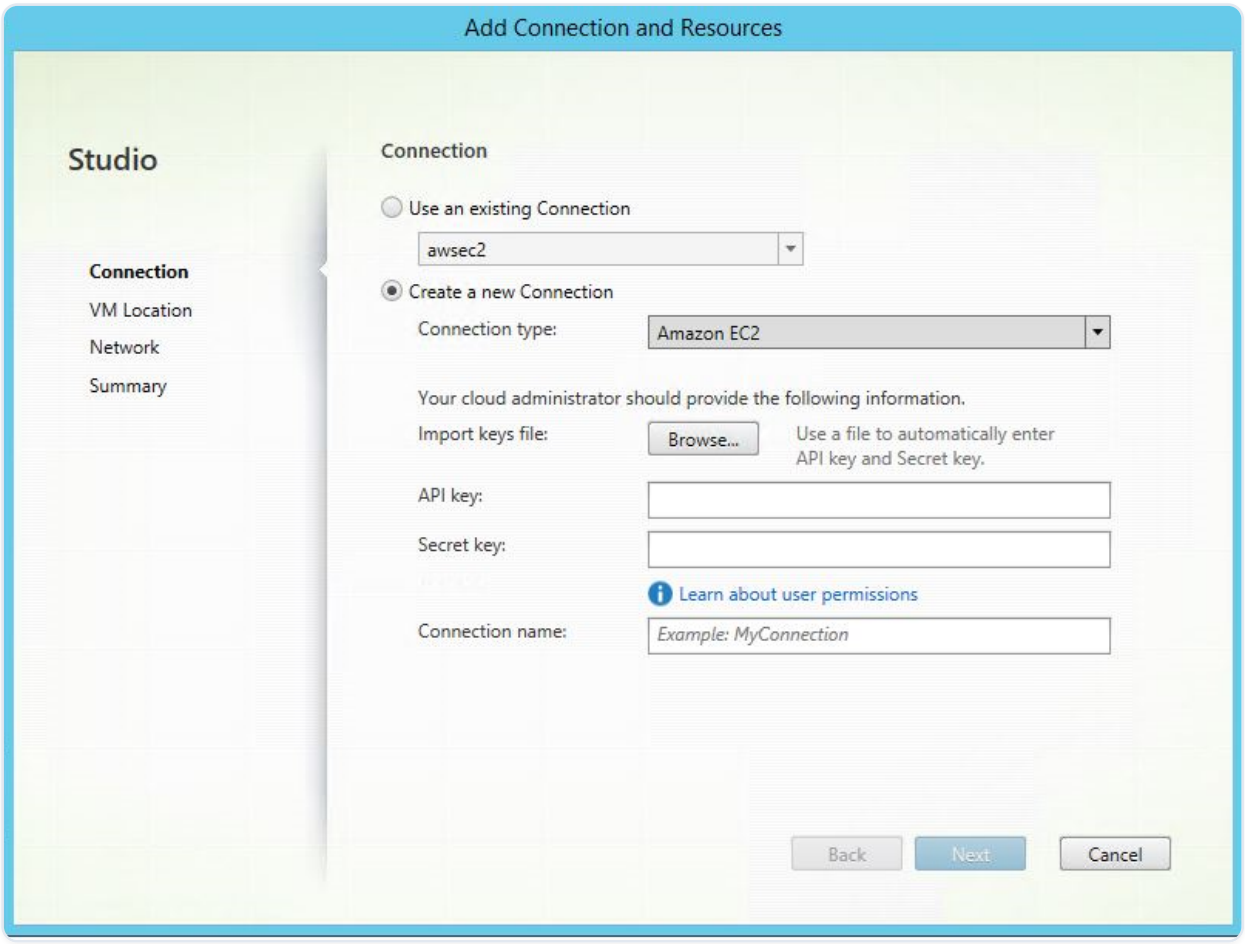
1. Citrix Studioで、**構成** > **ホスティング** > **接続とリソースの追加**を選択して、AWSへの接続を作成します。



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キーとシークレットキーを入力し、接続名を入力します。



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

a. AWSサービスにサインインします。

b. Identity and Access Management (IAM) コンソールに移動します。

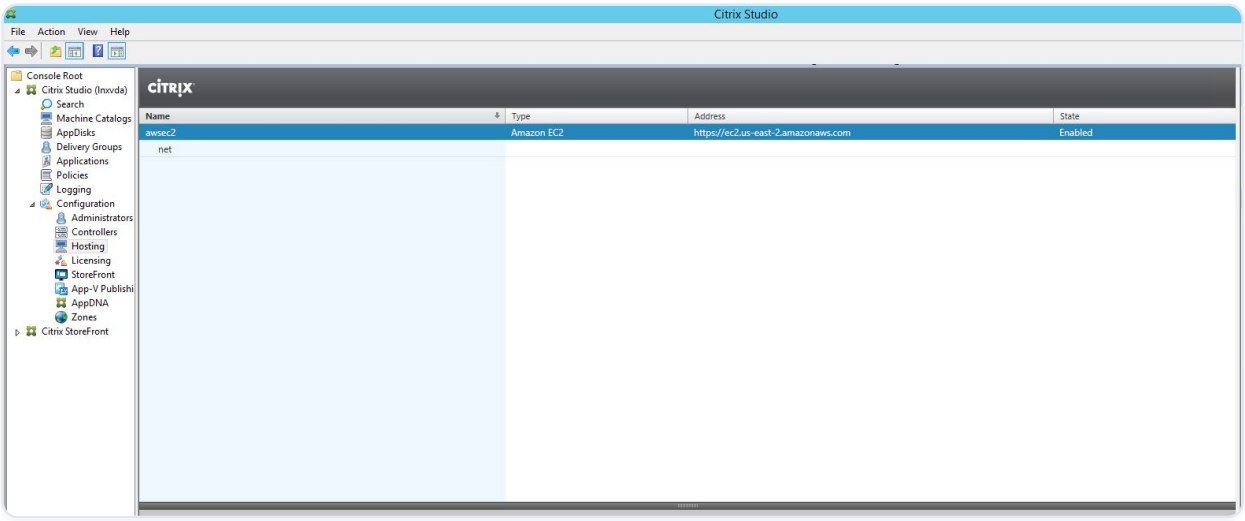
c. 左側のナビゲーションペインで、 を選択します。

d. 対象ユーザーを選択し、下にスクロールして タブを選択します。

e. 下にスクロールして をクリックします。新しいウィンドウが表示されます。

f. をクリックし、アクセスキーを安全な場所に保存します。

ホスティングペインに新しい接続が表示されます。



ステップ2：マスターイメージを準備する

マスターイメージには、オペレーティングシステム、非仮想化アプリケーション、VDA、およびその他のソフトウェアが含まれています。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。

ステップ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ホスト名が保持されるようにするには、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
   ```

   Ubuntu 18.04 をご利用の場合は、/etc/cloud/cloud.cfg ファイル内の system_info セクションに、以下の行が含まれていることを確認してください。

   ```
   system_info:

       network:

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

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

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

```
users:

- default
```

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

```
PasswordAuthentication yes
```

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

ステップ2b: テンプレートVMにLinux VDAパッケージをインストールする

注:

現在実行中のVDAをテンプレートVMとして使用する場合は、この手順を省略します。

テンプレートVMにLinux VDAパッケージをインストールする前に、.NET Core Runtime 3.1をインストールします。詳細については、「インストール概要」を参照してください。

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

RHEL/CentOSの場合:

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

Ubuntuの場合:

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

```
apt-get install -f
```

SUSE 12の場合:

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

ステップ2c: tdb-toolsパッケージをインストールするためのリポジトリを有効にする

RHEL 7サーバーの場合:

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

RHEL 7ワークステーションの場合:

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

ステップ2d: ntfs-3gを含むEPELリポジトリをインストールする

RHEL 6/CentOS 6、RHEL 7/CentOS 7にEPELリポジトリをインストールします。これにより、後でdeploymcs.shを実行したときに、そこに含まれるntfs-3gパッケージがインストールされます。

ステップ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、コンパイルして、ntfs-3gを手動でインストールします。

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

   ```
   sudo zypper install gcc

   sudo zypper install make
   ```

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

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

4. ntfs-3gパッケージへのパスを入力します。

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

5. ntfs-3gをインストールします。

```
./configure

make

make install
```

ステップ2f: ランタイム環境をセットアップする

deploymcs.shを実行する前に、以下を実行します。

- **/etc/xdl/mcs/mcs.conf**内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。以下にいくつかの変数を示しますが、そのうち`dns`と`AD_INTEGRATION`は設定する必要があります。

  注：変数を複数の値で設定できる場合は、値を単一引用符で囲み、スペースで区切ります。例：LDAP_LIST='aaa.lab:389 bbb.lab:389'。

  - `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の同等のものと同じになります。ただし、`dns`と`AD_INTEGRATION`の変数は引き続き構成する必要があります。デフォルト値はNで、これはMCSで作成されたマシン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 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`: ワークグループ名を設定します。これは、ADで構成されている場合はNetBIOS名（大文字と小文字を区別）です。そうでない場合は、デフォルトでドメイン名になります。

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

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

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

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

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

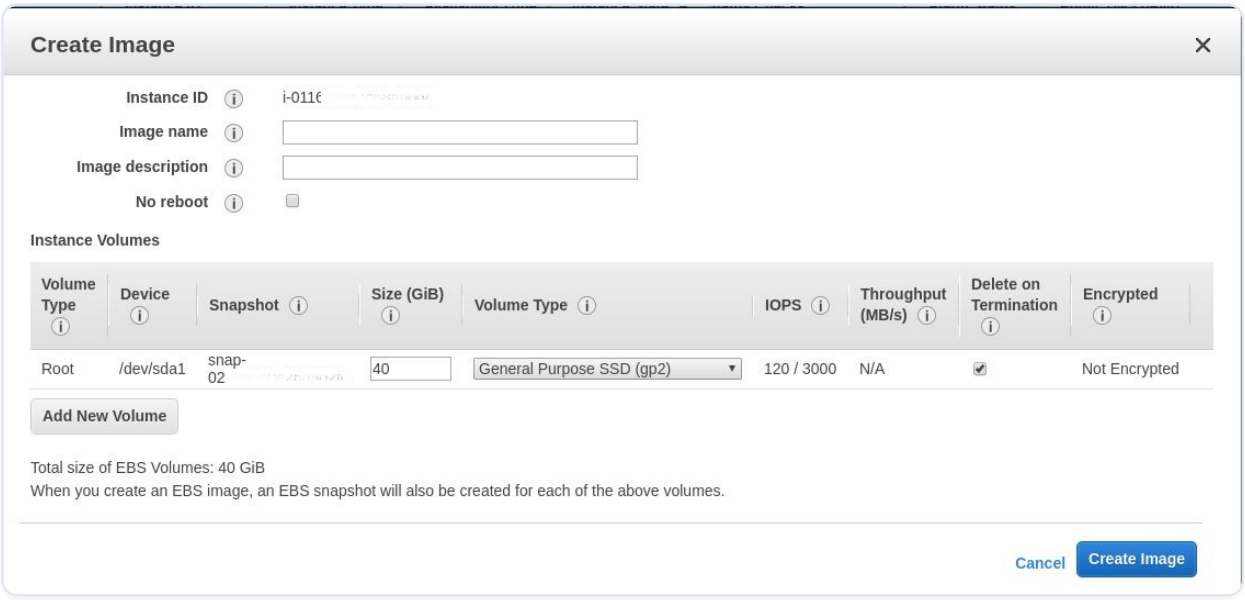
手順2g：マスターイメージを作成する

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VMで、構成テンプレートを更新して、作成されたすべてのVM上の関連する /etc/krb5.conf、/etc/samba/smb.conf、および /etc/sssd/sssd.conf ファイルをカスタマイズします。

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

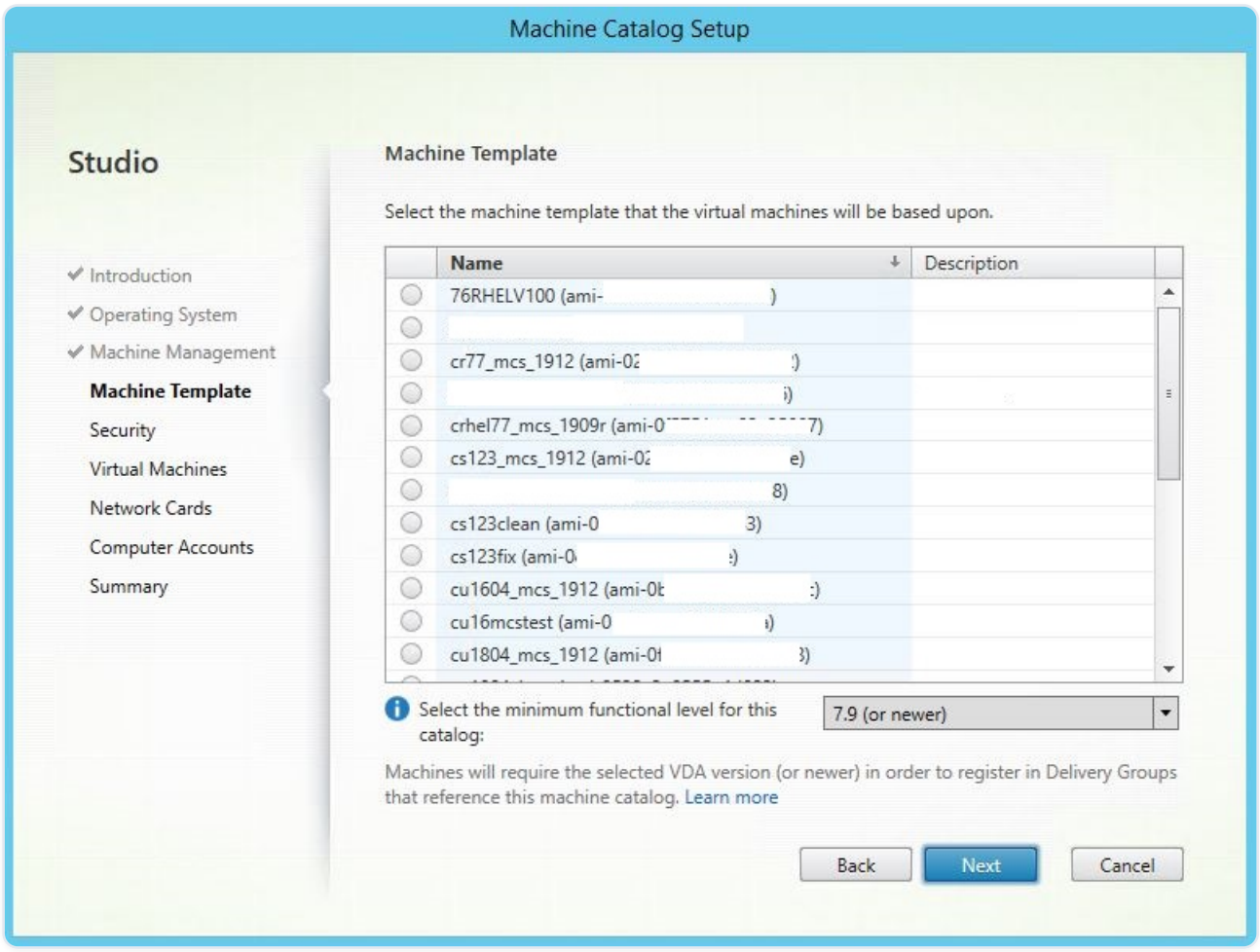
   SSSDユーザーの場合は、/etc/xdl/mcs/sssd.conf.tmpl、/etc/xdl/mcs/sssd_krb5.conf.tmpl、および /etc/xdl/mcs/sssd_smb.conf.tmpl テンプレートを更新します。

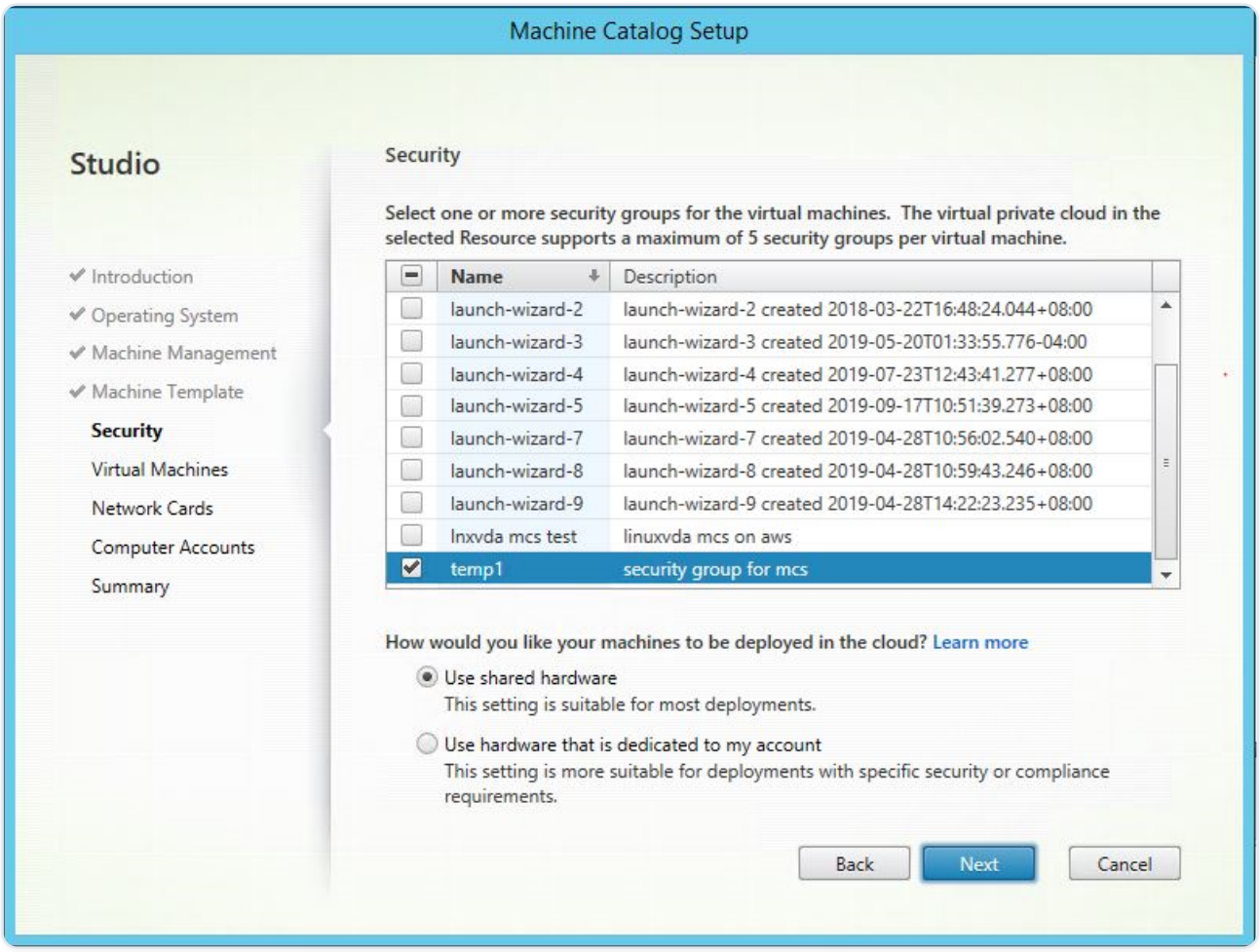
   注： テンプレートファイルで使用されている既存の形式を維持し、$WORKGROUP、$REALM、$realm、$AD_FQDNなどの変数を使用してください。
3. テンプレートVMにアプリケーションをインストールし、AWS EC2ポータルからテンプレートVMをシャットダウンします。テンプレートVMのインスタンス状態が **Stopped** であることを確認してください。
4. テンプレートVMを右クリックし、**Image** > **Create Image** を選択します。必要な情報を入力し、設定を行います。**Create Image** をクリックします。



ステップ3：マシンカタログを作成する

Citrix Studioで、マシンカタログを作成し、カタログで作成するVMの数を指定します。マシンカタログを作成する際に、マシンテンプレート（以前に作成したマスターイメージ）を選択し、1つ以上のセキュリティグループを選択します。





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

ステップ 4: デリバリーグループを作成する

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

MCS を使用して Linux VDA をアップグレードする

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

1. 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
   ```

   RHEL 6/CentOS 6 の場合：

```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.el6_x.x86_64.rpm
```

SUSE 12 の場合：

```
sudo rpm -U XenDesktopVDA-19.12.0.50-1.sle12_x.x86_64.rpm
```

Ubuntu 16.04 の場合：

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu16.04_amd64.deb
```

Ubuntu 18.04 の場合：

```
sudo dpkg -i xendesktopvda_19.12.0.50-1.ubuntu18.04_amd64.deb
```

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てマシンカタログを更新します。各マシンが再起動するまで待機します。マシンを手動で再起動しないでください。

デリバリーコントローラーの構成

XenDesktop 7.6 およびそれ以前のバージョンでは、Linux VDA をサポートするために変更が必要です。これらのバージョンでは、ホットフィックスまたは更新スクリプトが必要です。インストールおよび検証手順については、この記事で説明します。

Delivery Controller™ の構成を更新する

XenDesktop 7.6 SP2 の場合、Linux Virtual Desktops 用のブローカーを更新するためにホットフィックスアップデート 2 を適用します。ホットフィックスアップデート 2 はこちらから入手できます:

CTX142438: ホットフィックスアップデート 2 - Delivery Controller 7.6 (32ビット) 用 – 英語

XenDesktop 7.6 SP2 より前のバージョンの場合、**Update-BrokerServiceConfig.ps1** という名前の PowerShell スクリプトを使用して Broker Service の構成を更新できます。このスクリプトは、次のパッケージで入手できます:

- citrix-linuxvda-scripts.zip

ファーム内のすべての Delivery Controller で、次の手順を繰り返します:

1. **Update-BrokerServiceConfig.ps1** スクリプトを Delivery Controller マシンにコピーします。
2. ローカル管理者として Windows PowerShell コンソールを開きます。
3. **Update-BrokerServiceConfig.ps1** スクリプトが含まれているフォルダーに移動します。
4. **Update-BrokerServiceConfig.ps1** スクリプトを実行します:

```
.\Update-BrokerServiceConfig.ps1
```

ヒント:

デフォルトでは、PowerShell は PowerShell スクリプトの実行を防止するように構成されています。スクリプトの実行に失敗した場合は、再試行する前に PowerShell の実行ポリシーを変更してください:

```
Set-ExecutionPolicy Unrestricted
```

Update-BrokerServiceConfig.ps1 スクリプトは、Linux VDA で必要となる新しい WCF エンドポイントを使用して Broker Service の構成ファイルを更新し、Broker Service を再起動します。スクリプトは、Broker Service の構成ファイルの場所を自動的に特定します。元の構成ファイルのバックアップは、同じディレクトリにファイル名に **.prelinux** を付加して作成されます。

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ている Windows VDA のブローカー処理には影響しません。単一の Controller ファームで、Windows VDA と Linux VDA の両方のセッションをシームレスに管理およびブローカーできます。

Delivery Controllerの構成を確認する

必要な構成変更がDelivery Controllerに適用されると、**EndpointLinux**文字列が **%PROGRAMFILES%\\Citrix\\Broker\\Service\\BrokerService.exe.config**ファイルに5回表示されます。

Windowsコマンドプロンプトから、ローカル管理者としてログオンして以下を確認します。

```
cd "%PROGRAMFILES%"\Citrix\Broker\Service\

findstr EndpointLinux BrokerService.exe.config
```

Linux VDA の構成

このセクションでは、Linux VDA の機能について、機能の説明、構成、トラブルシューティングを含めて詳しく説明します。

ヒント:

ログ収集に使用される `xdlcollect` Bashスクリプトは、Linux VDAソフトウェアに統合されており、/opt/Citrix/VDA/binにあります。Linux VDAをインストールした後、`bash /opt/Citrix/VDA/bin/xdlcollect.sh` コマンドを実行してログを収集できます。

ログ収集が完了すると、スクリプトと同じフォルダーに圧縮ログファイルが生成されます。`xdlcollect` は、圧縮ログファイルをCitrix Insight Services (CIS)にアップロードするかどうかを尋ねることができます。同意した場合、アップロード完了後、`xdlcollect` はupload_IDを返します。アップロードしても、圧縮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のログファイルにアクセスできます。

NIS と Active Directory を統合する

この記事では、SSSD を使用して Linux VDA 上で NIS を Windows Active Directory (AD) と統合する方法について説明します。Linux VDA は Citrix Virtual Apps and Desktops のコンポーネントと見なされます。その結果、Windows AD 環境に密接に適合します。

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場合、アカウント情報（ユーザー名とパスワードの組み合わせ）が AD と NIS の両方で同じである必要があります。

注:

認証は引き続き AD サーバーによって実行されます。NIS+ はサポートされていません。NIS を UID および GID プロバイダーとして使用する場合、Windows サーバーからの POSIX 属性は使用されなくなります。

> ヒント:

> この方法は、Linux VDA を展開する非推奨の方法であり、特別なユースケースでのみ使用されます。RHEL/CentOS ディストリビューションの場合は、RHEL/CentOS 用 Linux Virtual Delivery Agent のインストール の手順に従ってください。Ubuntu ディストリビューションの場合は、Ubuntu 用 Linux Virtual Delivery Agent のインストール の手順に従ってください。

> SSSD とは？

SSSD はシステムデーモンです。その主な機能は、システムのキャッシュとオフラインサポートを提供できる共通のフレームワークを通じて、リモートリソースを識別および認証するためのアクセスを提供することです。PAM モジュールと NSS モジュールの両方を提供し、将来的には拡張ユーザー情報のための D-BUS ベースのインターフェイスをサポートできます。また、ローカルユーザーアカウントと拡張ユーザーデータを保存するためのより優れたデータベースも提供します。

必要なソフトウェア

AD プロバイダーは SSSD バージョン 1.9.0 で初めて導入されました。

この記事に含まれる手順を使用する場合、以下の環境がテストおよび検証されています。

- RHEL 7.7 以降
- CentOS 7.7 以降

NIS と AD を統合する

NIS を AD と統合するには、次の手順を実行します。

1. Linux VDA を NIS クライアントとして追加する
2. ドメインに参加し、Samba を使用してホストキータブを作成する(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/integrate-nis-with-active-directory.html#join-the-domain-and-create-a-host-keytab-using-samba)
3. SSSD をセットアップする(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/integrate-nis-with-active-directory.html#set-up-sssd)
4. NSS/PAM を構成する(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/integrate-nis-with-active-directory.html#configure-nsspam)

5. Kerberos 構成を確認する(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/integrate-nis-with-active-directory.html#verify-the-kerberos-configuration)
6. ユーザー認証を確認する(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/integrate-nis-with-active-directory.html#verify-user-authentication)

Linux VDA を NIS クライアントとして追加する

NIS クライアントを構成します。

```
yum –y install ypbind rpcbind oddjob-mkhomedir
```

NIS ドメインを設定します。

```
ypdomainname nis.domain

echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

NIS サーバーとクライアントの IP アドレスを **/etc/hosts** に追加します。

`{NIS server IP address}  server.nis.domain nis.domain`

`authconfig` を使用して NIS を構成します。

```
sudo authconfig --enablenis --nisdomain=nis.domain --
nisserver=server.nis.domain --enablemkhomedir --update
```

nis.domain は NIS サーバーのドメイン名を表します。**server.nis.domain** は NIS サーバーのホスト名であり、NIS サーバーの IP アドレスでもかまいません。

NIS サービスを構成します。

```
sudo systemctl start rpcbind ypbind



sudo systemctl enable rpcbind ypbind
```

NIS 構成が正しいことを確認します。

```
ypwhich
```

アカウント情報が NIS サーバーから利用可能であることを検証します。

```
getent passwd nisaccount
```

注:

nisaccount は NIS サーバー上の実際の NIS アカウントを表します。UID、GID、ホームディレクトリ、およびログインシェルが正しく構成されていることを確認してください。

Samba を使用してドメインに参加し、ホストキータブを作成する

SSSD は、ドメインへの参加やシステムキータブファイルの管理のための AD クライアント機能を提供しません。これらの機能を実現するには、いくつかの方法があります。

- アドクリ
- レルムディー
- ウィンバインド
- サンバ

このセクションの情報は、Samba のアプローチのみを説明しています。**realmd** については、RHEL または CentOS ベンダーのドキュメントを参照してください。これらの手順は、SSSD を構成する前に実行する必要があります。

Samba を使用してドメインに参加し、ホストキータブを作成する:

適切に構成されたファイルを持つ Linux クライアントで:

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

Samba および Kerberos 認証用にマシンを構成します:

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

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

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

`--enablekrb5kdcdns --enablekrb5realmdns`

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

`kerberos method = secrets and keytab`

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

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

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

SSSDのセットアップ

SSSDのセットアップは、次の手順で構成されます。

- Linuxクライアントマシンに **sssd-ad** および **sssd-proxy** パッケージをインストールします。
- さまざまなファイル（例：**sssd.conf**）に構成変更を加えます。
- **sssdサービス**を開始します。

/etc/sssd/sssd.conf

sssd.conf の構成例（必要に応じてオプションを追加できます）：

```
[sssd]

config_file_version = 2

domains = EXAMPLE

services = nss, pam



[domain/EXAMPLE]

# Uncomment if you need offline logins
```

```
# cache_credentials = true

re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?
P<domain>.+$))|(^(?P<name>[^@\\]+)$))

id_provider = proxy

proxy_lib_name = nis

auth_provider = ad

access_provider = ad



# Should be specified as the long version of the Active Directory domain.

ad_domain = EXAMPLE.COM



# Kerberos settings

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U



# Uncomment if service discovery is not working

# ad_server = server.ad.example.com



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

default_shell = /bin/bash

fallback_homedir = /home/%d/%u
```

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

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

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

sssd.conf のファイルの所有権とアクセス許可を設定します。

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

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

NSS/PAMの構成

RHEL/CentOS:

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

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



sudo systemctl start sssd



sudo systemctl enable sssd
```

ヒント:

Linux VDA設定を構成する際、SSSDの場合、Linux VDAクライアントに特別な設定がないことを考慮してください。**ctxsetup.sh** スクリプトの追加ソリューションについては、デフォルト値を使用してください。

Kerberos の構成を確認する

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

```
sudo klist -ke
```

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

```
sudo kinit –k MACHINE\$@REALM
```

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

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

```
sudo klist -ke
```

ユーザー認証を確認する

getent コマンドを使用して、ログオン形式がサポートされているか、および NSS が機能するかどうかを確認します。

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

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

サポートされているログオン形式は次のとおりです。

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

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

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



id -u
```

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

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

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

```
klist
```

アプリケーションの公開

Linux VDAバージョン7.13で、Citrixはサポートされているすべてのプラットフォームにシームレスアプリケーション機能を追加しました。この機能を使用するために、特定のインストール手順は必要ありません。

ヒント:

Linux VDAバージョン1.4で、Citrixは非シームレス公開アプリケーションとセッション共有のサポートを追加しました。

Citrix Studioを使用したアプリケーションの公開

デリバリーグループを作成するとき、または既存のデリバリーグループにアプリケーションを追加するときに、Linux VDAにインストールされているアプリケーションを公開できます。このプロセスは、Windows VDAにインストールされているアプリケーションを公開するのと同様です。詳し

くは、Citrix Virtual Apps and Desktopsのドキュメント（使用しているCitrix Virtual Apps and Desktopsのバージョンに基づく）を参照してください。

ヒント:

デリバリーグループを構成するときは、デリバリーの種類が**デスクトップとアプリケーション**または**アプリケーション**に設定されていることを確認してください。

重要:

アプリケーションの公開は、Linux VDAバージョン1.4以降でサポートされています。ただし、Linux VDAは、デスクトップとアプリを同じマシンに配信することをサポートしていません。この問題に対処するため、Citrixはアプリとデスクトップの配信用に個別のデリバリーグループを作成することを推奨します。

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。「TWIMode=Off」を設定してすでに無効にしている場合は、「TWIMode=On」に変更するのではなく、この設定を削除してください。そうしないと、公開されたデスクトップを起動できない可能性があります。

制限事項

Linux VDAは、単一ユーザーによる同じアプリケーションの複数の同時インスタンスの起動をサポートしていません。

既知の問題

アプリケーションの公開中に、以下の既知の問題が確認されています。

- 非矩形ウィンドウはサポートされていません。ウィンドウの角にサーバー側の背景が表示される場合があります。
- 公開されたアプリケーションのウィンドウコンテンツのプレビューはサポートされていません。

- 現在、シームレスモードは以下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ものに設定されていることを確認してください。
- 複数のLibreOfficeアプリケーションを実行すると、これらのアプリケーションはプロセスを共有するため、最初に起動されたものだけがCitrix Studioに表示されます。
- 「Dolphin」のような公開されたQt5ベースのアプリケーションでは、アイコンが表示されない場合があります。この問題を解決するには、`<https://wiki.archlinux.org/title/Qt>`の記事を参照してください。
- 同じICA®セッションで実行されている公開アプリケーションのすべてのタスクバーボタンは、同じグループにまとめられます。この問題を解決するには、タスクバーのプロパティでタスクバーボタンを結合しないように設定します。

リモートPCアクセス

概要

リモートPCアクセスは、Citrix Virtual Apps and Desktopsの拡張機能です。これにより、組織は従業員が物理的なオフィスPCへ安全にリモートアクセスすることを容易に許可できます。ユーザーがオフィスPCにアクセスできれば、仕事に必要なすべてのアプリケーション、データ、リソースにアクセスできます。

リモートPCアクセスは、仮想デスクトップとアプリケーションを提供するCitrix Virtual Apps™ and Desktopsと同じコンポーネントを使用します。リモートPCアクセスの展開と構成の要件およびプロセスは、仮想リソースを提供するための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。この一貫性により、一貫性のある統合された管理エクスペリエンスが提供されます。ユーザーは、Citrix HDXを使用してリモートオフィスPCセッションを配信することで、最高のユーザーエクスペリエンスを得られます。

詳細については、Citrix Virtual Apps and Desktopsドキュメントのリモートアクセスを参照してください。

構成

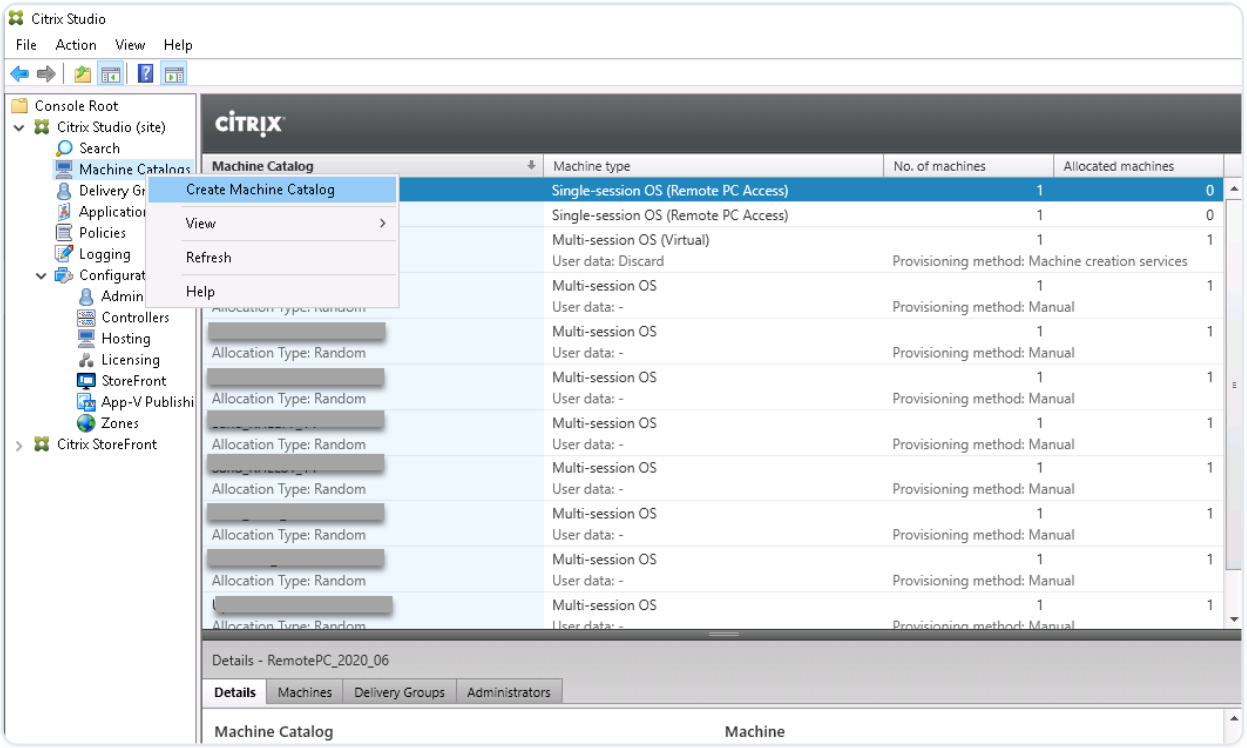
Linux PCセッションを配信するには、ターゲットPCにLinux VDAをインストールし、**リモートPCアクセス**タイプのマシンカタログを作成し、アクセスを要求するユーザーがマシンカタログ内のPCを利用できるようにデリバリーグループを作成します。次のセクションで手順を詳しく説明します。

ステップ1 - ターゲットPCにLinux VDAをインストールする

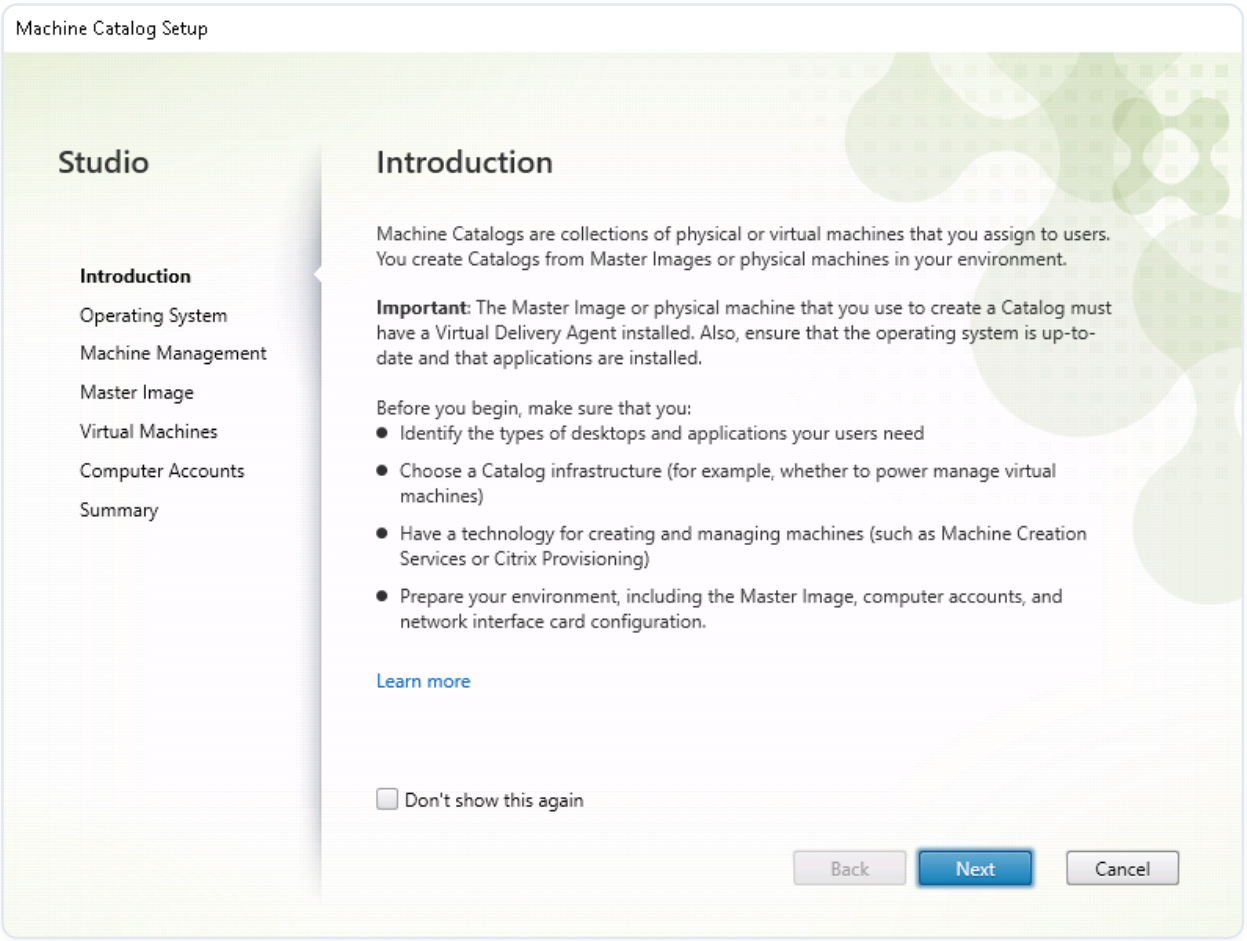
Linux VDAのインストールには、簡易インストールを使用することをお勧めします。インストール中に、`CTX_XDL_VDI_MODE`変数の値を`Y`に設定します。

ステップ2 - 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

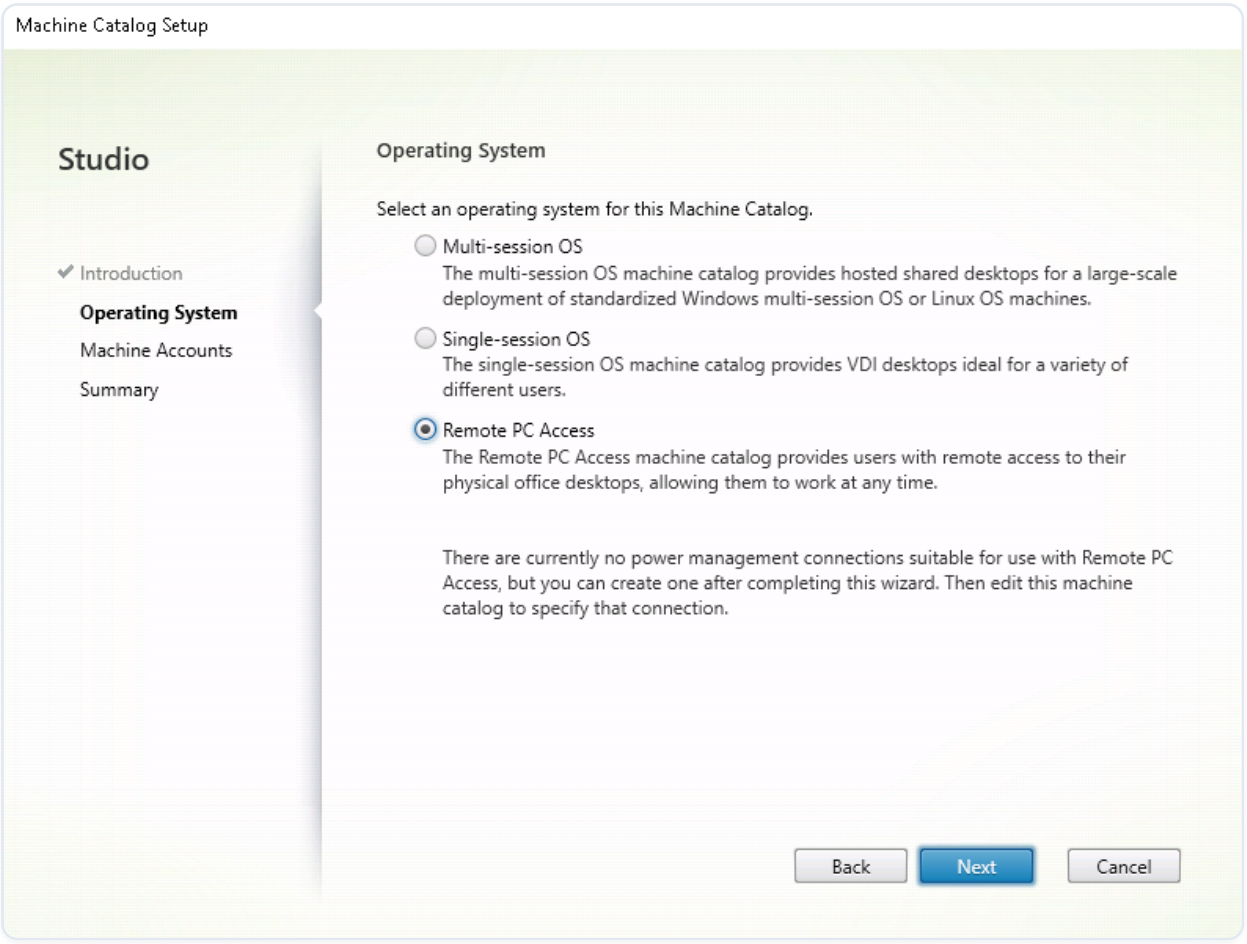
1. Citrix Studioで、**マシンカタログ**を右クリックし、ショートカットメニューから**マシンカタログの作成**を選択します。



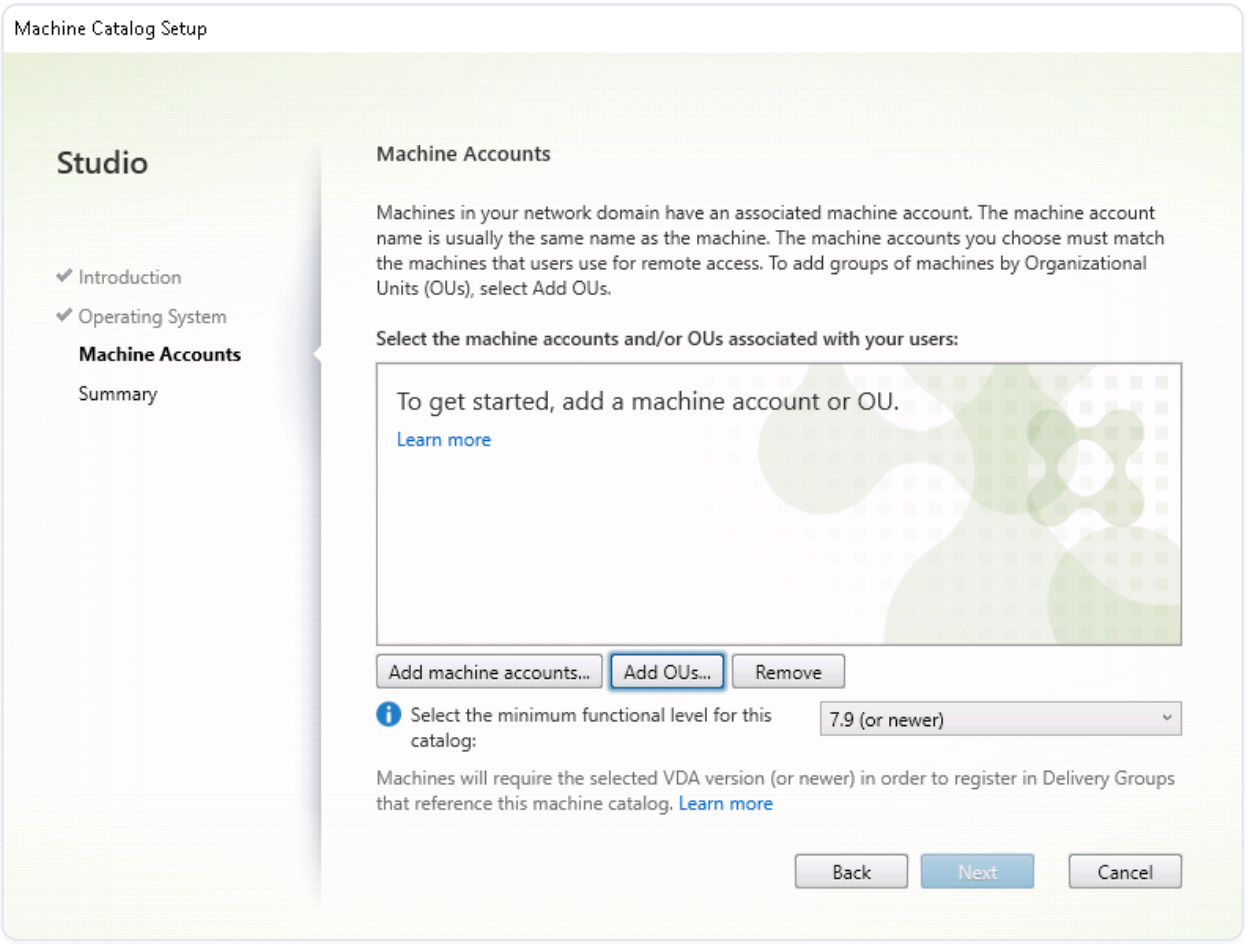
2. **はじめに**ページで**次へ**をクリックします。



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. ターゲットPCを含むOUを選択するには、**OUの追加**をクリックします。または、個々のマシンをマシンカタログに追加するには、**マシンアカウントの追加**をクリックします。



5. マシンカタログに名前を付けます。

   マシンカタログに名前を付けている画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-5-name-the-machine-catalog.png)

6. (オプション) マシンカタログを右クリックして、関連する操作を実行します。

   マシンカタログでの関連操作の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-6-edit-a-machine-catalog.png)

ステップ3 - アクセスを要求するユーザーがマシンカタログ内のPCを利用できるように、デリバリーグループを作成します

1. Citrix Studioで、 **［デリバリーグループ］** を右クリックし、ショートカットメニューから **［デリバリーグループの作成］** を選択します。

   デリバリーグループの作成を選択している画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-11-create-a-delivery-group.png)

2. 「**デリバリーグループの開始**」ページで **［次へ］** をクリックします。

   デリバリーグループの開始ページの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-22-getting-started-with-delivery-groups.png)

3. 手順2で作成したマシンカタログを選択し、デリバリーグループに関連付けます。

   マシンカタログをデリバリーグループに関連付けている画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-33-assocatiing-the-machine-catalog-with-the-delivery-group.png)

4. マシンカタログ内のPCにアクセスできるユーザーを追加します。追加したユーザーは、クライアントデバイス上のCitrix Workspace™アプリを使用して、PCにリモートでアクセスできます。

   マシンカタログ内のPCにアクセスできるユーザーを追加している画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/remote-pc-access-44-add-uses-to-access-pcs.png)

考慮事項

以下の考慮事項は、Linux VDAに固有のものです。

- Linux VDAは、物理マシンでは非3Dモードでのみ使用してください。NVIDIAドライバーの制限により、HDX™ 3Dモードが有効になっている場合、PCのローカル画面をブラックアウトできず、セッションのアクティビティが表示されます。この画面を表示することはセキュリティリスクです。
- 物理Linuxマシンには、シングルセッションOSタイプのマシンカタログを使用します。
- 統合されたWake on LAN機能は、Linuxマシンでは利用できません。
- 自動ユーザー割り当てはLinuxマシンでは利用できません。自動ユーザー割り当てを使用すると、ユーザーがPCにローカルでログオンしたときに、そのユーザーは自動的に自分のマシンに割り当てられます。このログオンは管理者の介入なしに行われます。クライアントデバイスで実行されているCitrix Workspaceアプリは、Remote PC Accessデスクトップセッション内で、ユーザーにオフィスPC上のアプリケーションとデータへのアクセスを提供します。
- ユーザーがすでにローカルでPCにログオンしている場合、StoreFront™からPCを起動しようとすると失敗します。
- 省電力オプションはLinuxマシンでは利用できません。

その他のリソース

以下は Remote PC Access に関するその他のリソースです。

- ソリューション設計ガイダンス: Remote PC Access Design Decisions。
- Remote PC Access アーキテクチャの例: Citrix Remote PC Access ソリューションのリファレンスアーキテクチャ。

印刷

この記事では、印刷のベストプラクティスに関する情報を提供します。

インストール

Linux VDAには、**cups**フィルターと**foomatic**フィルターの両方が必要です。これらのフィルターは、VDAのインストール時にインストールされます。ディストリビューションに基づいて、フィルターを手動でインストールすることもできます。例：

RHEL 7の場合：

```
sudo yum –y install cups



sudo yum -y install foomatic-filters
```

RHEL 6の場合：

```
sudo yum –y install cups



sudo yum -y install foomatic
```

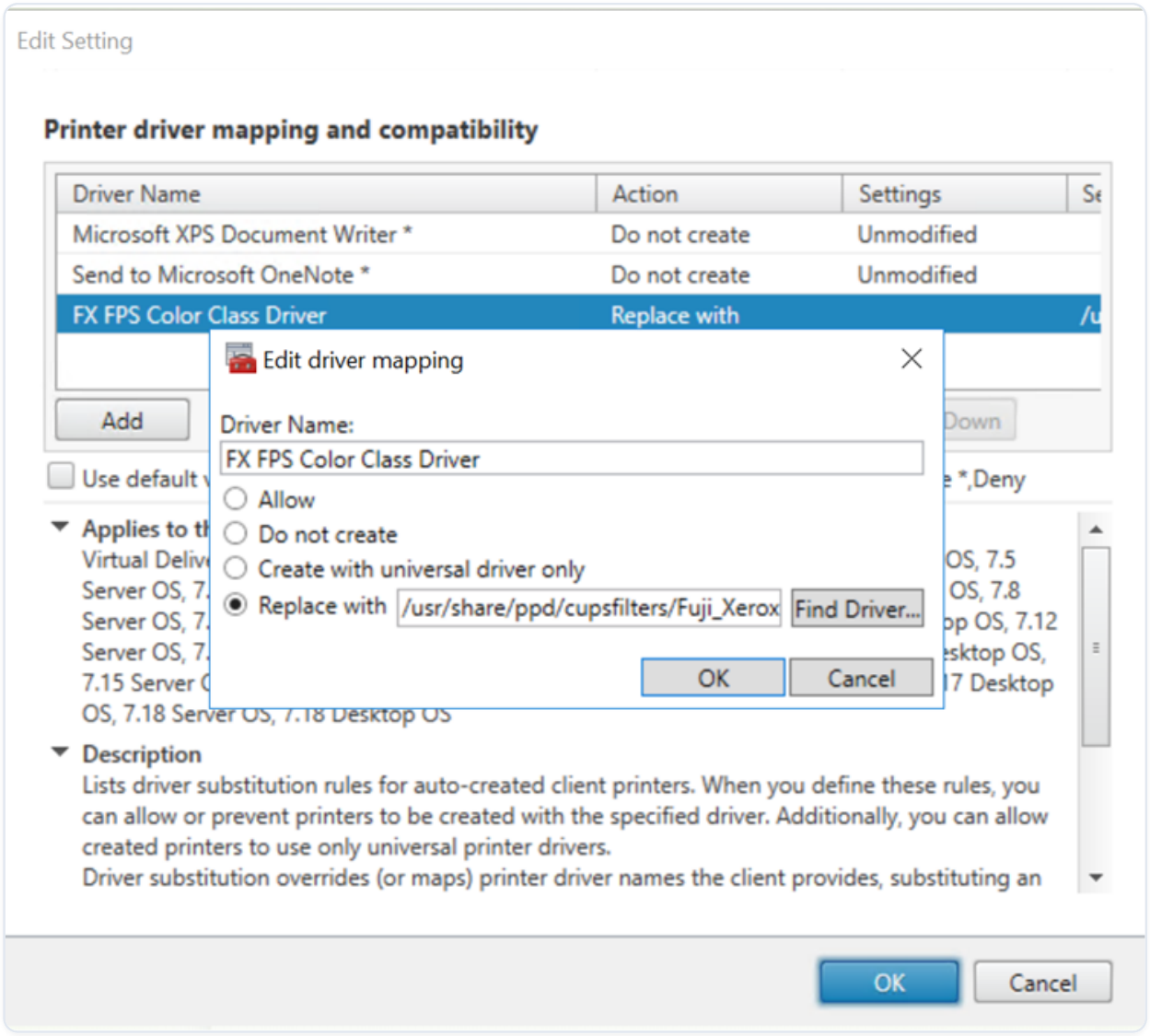
構成

Citrixが提供するユニバーサルプリンタードライバーには3種類（postscript、pcl5、pcl6）あります。ただし、ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合があります。この場合、以前のリリースでは、~/.CtxlpProfile$CLIENT_NAME構成ファイルを編集するしかありませんでした。バージョン1906以降では、代わりにCitrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成することを選択できます。

Citrix Studioで**プリンタードライバーのマッピングと互換性**ポリシーを構成するには：

1. **プリンタードライバーのマッピングと互換性**ポリシーを選択します。
2. 「**追加**」をクリックします。
3. **ドライバー名**にクライアントプリンターのドライバー名を入力します。Citrix Workspace™ app for Linuxを使用している場合は、代わりにプリンター名を入力します。

4\. **次で置換**を選択し、VDA上のドライバーファイルの絶対パスを入力します。



注:

- PPDドライバーファイルのみがサポートされています。
- **プリンタードライバーのマッピングと互換性**ポリシーのその他のオプションはサポートされていません。**次で置換**のみが有効になります。

使用方法

公開デスクトップと公開アプリケーションの両方から印刷できます。クライアント側のデフォルトプリンターのみがLinux VDAセッションにマップされます。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:**
>
> 同じユーザーが公開デスクトップと公開アプリケーションの両方を開いた場合、両方のプリンターがセッションで利用可能になります。公開アプリケーションセッションでデスクトッププリンターに印刷したり、公開デスクトップでアプリケーションプリンターに印刷したりすることはできません。

トラブルシューティング

印刷できません

印刷が正しく機能しない場合は、印刷デーモンである **ctxlpmngt** と CUPS フレームワークを確認してください。

印刷デーモンである **ctxlpmngt** はセッションごとのプロセスであり、セッションの期間中実行されている必要があります。次のコマンドを実行して、印刷デーモンが実行されていることを確認します。**ctxlpmngt** が実行されていない場合は、コマンドラインから手動で **ctxlpmngt** を開始します。

```
ps –ef | grep ctxlpmngt
```

印刷がまだ機能しない場合は、CUPS フレームワークを確認してください。**ctxcups** サービスはプリンター管理に使用され、Linux CUPS フレームワークと通信します。これはマシンごとに単一のプロセスであり、次のコマンドを実行して確認できます。

```
service ctxcups status
```

CUPS ログを収集するための追加手順

CUPS ログを収集するには、次のコマンドを実行して CUPS サービスファイルを構成します。そうしないと、CUPS ログは **hdx.log** に記録できません。

```
sudo service cups stop


sudo vi /etc/systemd/system/printer.target.wants/cups.service


PrivateTmp=false


sudo service cups start


sudo systemctl daemon-reload
```

注:

この構成は、問題が発生したときに完全な印刷ログを収集するためだけに行われます。通常、この構成はCUPSのセキュリティを破るため推奨されません。

印刷出力が文字化けする

互換性のないプリンタードライバーは、出力の文字化けを引き起こす可能性があります。ユーザーごとのドライバー構成が利用可能で、**~/.CtxlpProfile$CLIENT_NAME** 構成ファイルを編集することで構成できます。

```
[DEFAULT_PRINTER]


printername=


model=
```

```
ppdpath=



drivertype=
```

重要:

printername は、現在のクライアント側のデフォルトプリンターの名前を含むフィールドです。これは読み取り専用の値です。編集しないでください。

ppdpath、**model**、**drivertype** の各フィールドは、マップされたプリンターに対して1つしか有効にならないため、同時に設定することはできません。

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、**model=** オプションを使用してネイティブプリンタードライバーのモデルを構成します。**lpinfo** コマンドを使用して、プリンターの現在のモデル名を見つけることができます。

  ```
  lpinfo –m



  …



  xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0



  xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

  xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
  ```

  その後、プリンターに合わせてモデルを設定できます。

```
model=xerox/ph3115.ppd.gz
```

- ユニバーサルプリンタードライバーがクライアントプリンターと互換性がない場合は、ネイティブプリンタードライバーのPPDファイルパスを構成します。**ppdpath** の値は、ネイティブプリンタードライバーファイルの絶対パスです。

  例えば、/home/tester/NATIVE_PRINTER_DRIVER.ppd の下に **ppd driver** があります。

  ```
  ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
  ```

- Citrixが提供するユニバーサルプリンタードライバーには3種類あります（PostScript、PCL5、PCL6）。プリンターのプロパティに基づいてドライバーの種類を構成できます。

  例えば、クライアントのデフォルトプリンタードライバーの種類がPCL5の場合、**drivertype** を次のように設定します。

  ```
  drivertype=pcl5
  ```

出力サイズがゼロ

異なる種類のプリンターを試してください。また、CutePDFやPDFCreatorのような仮想プリンターを試して、この問題がプリンタードライバーに関連しているかどうかを確認してください。

印刷ジョブは、クライアントのデフォルトプリンターのプリンタードライバーに依存します。現在アクティブなドライバーの種類を特定することが重要です。クライアントプリンターがPCL5ドライバーを使用しているのに、Linux VDAがPostScriptドライバーを選択すると、問題が発生する可能性があります。

プリンタードライバーの種類が正しい場合は、次の手順を実行して問題を特定できます。

1. 公開デスクトップセッションにログオンします。
2. **vi ~/.CtxlpProfile$CLIENT_NAME** コマンドを実行します。
3. Linux VDAにスプールファイルを保存するために、次のフィールドを追加します。

   ```
   deletespoolfile=no
   ```

4. ログオフして再度ログオンし、構成の変更を読み込みます。

5. ドキュメントを印刷して問題を再現します。印刷後、スプールファイルは **/var/spool/cups-ctx/$logon_user/$spool_file** の下に保存されます。
6. スプールが空かどうかを確認します。スプールファイルがゼロの場合、問題を示しています。詳細なガイダンスについては、Citrixサポートに連絡してください（印刷ログを提供してください）。
7. スプールサイズがゼロでない場合は、ファイルをクライアントにコピーします。スプールファイルの内容は、クライアントのデフォルトプリンターのプリンタードライバーの種類に依存します。マップされたプリンター（ネイティブ）ドライバーがPostScriptの場合、スプールファイルはLinux OSで直接開くことができます。内容が正しいかどうかを確認します。

   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、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。以下の例では、PostScriptクライアントプリンターを例として使用しています。

   a. アクティブなセッションにログオンし、クライアントデスクトップでブラウザーを開きます。

   b. 印刷管理ポータルを開きます。

   ```
   localhost:631
   ```

   c. マップされたプリンター **CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID** を選択し、**プリンターの変更** をクリックします。この操作には管理者権限が必要です。

   d. cups-ctx接続を維持し、 をクリックしてプリンタードライバーを変更します。

   e. **製造元** および **モデル** フィールドで、Citrix UPDドライバーとは異なるプリンタードライバーを選択します。たとえば、CUPS-PDF仮想プリンターがインストールされている場合は、Generic CUPS-PDF Printerドライバーを選択します。変更を保存します。

   f. このプロセスが成功した場合、マップされたプリンターが新しく選択されたドライバーを使用できるように、**.CtxlpProfile$CLIENT_NAME** でドライバーのPPDファイルパスを構成します。

既知の問題

Linux VDAでの印刷中に、以下の問題が確認されています。

CTXPSドライバーは一部のPLCプリンターと互換性がありません

印刷出力の破損が発生した場合は、プリンタードライバーを製造元が提供するネイティブドライバーに設定してください。

大容量ドキュメントの印刷パフォーマンスが遅い

ローカルクライアントプリンターで大容量ドキュメントを印刷すると、ドキュメントはサーバー接続を介して転送されます。接続が遅い場合、転送に時間がかかることがあります。

他のセッションから見えるプリンターおよび印刷ジョブの通知

LinuxにはWindowsオペレーティングシステムと同じセッションの概念がありません。そのため、すべてのユーザーがシステム全体の通知を受け取ります。これらの通知は、CUPS構成ファイル ***/etc/cups/cupsd.conf*** を変更することで無効にできます。

ファイルに構成されている現在のポリシー名を特定します:

```
DefaultPolicy default
```

ポリシー名が *default* の場合、以下の行をデフォルトのポリシー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>
```

ファイル転送

Linux VDAとクライアントデバイス間で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サンドボックス属性をサポートするWebブラウザを実行している場合に利用できます。HTML5サンドボックス属性により、ユーザーはHTML5およびChrome用Citrix Workspace™アプリを使用して仮想アプリとデスクトップにアクセスできます。公開セッション内

では、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をアップロードおよびダウンロードできます。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上のファイルを選択して、そのファイルをLinux VDAにアップロードできます。

ツールバーのアップロードアイコンの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/upload-icon-on-toolbar.png)

> **注:**
>
> この機能は、RedHat7.7、CentOS7.6、SUSE12.3、Ubuntu16.04、および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が有効になっていることを確認してください。

ファイル転送ポリシー

Citrix Studioを使用して、ファイル転送ポリシーを設定できます。デフォルトでは、ファイル転送は有効になっています。

ファイル転送ポリシーの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/file-transfer-policies.png)

ポリシーの説明:

- **デスクトップとクライアント間のファイル転送を許可する。** ユーザーがCitrix Virtual Apps and Desktops™セッションとデバイス間でファイルを転送することを許可または禁止します。
- **デスクトップからファイルをダウンロードする。** ユーザーがCitrix Virtual Apps™ and Desktopsセッションからデバイスにファイルをダウンロードすることを許可または禁止します。
- **デスクトップにファイルをアップロードする。** ユーザーがデバイスからCitrix Virtual Apps and Desktopsセッションにファイルをアップロードすることを許可または禁止します。

> **注:**

> **デスクトップからファイルをダウンロードする**および**デスクトップにファイルをアップロードする**ポリシーが有効になるように、**デスクトップとクライアント間のファイル転送を許可する**ポリシーを**許可**に設定します。

使用方法

HTML5用Citrix Workspaceアプリを介してファイル転送機能を使用するには:

1. Citrix Studioで、**WebSockets接続**ポリシーを**許可**に設定します。

   WebSockets接続ポリシーの設定の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/websockets-connection.png)

2. Citrix Studioで、前述のファイル転送ポリシーを介してファイル転送を有効にします。
3. Citrix StoreFront管理コンソールで、**Stores**をクリックし、**Manage Receiver for Web Sites**ノードを選択して、**Always use Receiver for HTML5**オプションを選択してCitrix Receiver™ for HTML5を有効にします。

   「Always use Receiver for HTML5」オプションを選択している画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/always-use-receiver-for-html5.png)

4. 仮想デスクトップまたはWebブラウザアプリのセッションを起動します。Linux VDAとクライアントデバイス間でファイルをアップロードおよびダウンロードします。

Citrix Workspace app for Chromeを介してファイル転送機能を使用するには：

1. 前述のファイル転送ポリシーを介してファイル転送を有効にします。
2. ChromeウェブストアからCitrix Workspaceアプリを入手します。

   Citrix Workspace app for Chromeをすでにchromeアプリページに追加している場合は、この手順をスキップしてください。

   a. Google Chromeの検索ボックスに「**Citrix Workspace for Chrome**」と入力します。検索アイコンをクリックします。
   b. 検索結果の中から、Citrix Workspaceアプリが利用可能なChromeウェブストアへのURLをクリックします。

      Chrome版 シトリックス ワークスペース アプリ の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/workspace-app-for-chrome.png)

   c. **Add to Chrome**をクリックして、Citrix WorkspaceアプリをGoogle Chromeに追加します。

3. ChromeアプリページでCitrix Workspace app for Chromeをクリックします。
4. 接続するStoreFront™ストアのURLを入力します。

   以前にURLを入力したことがある場合は、この手順をスキップしてください。 ストアのURLを入力している画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/type-store-url.png)
5. 仮想デスクトップまたはWebブラウザアプリのセッションを起動します。Linux VDAとクライアントデバイス間でファイルをアップロードおよびダウンロードします。

PDF印刷

PDF印刷をサポートするバージョンのCitrix Workspace™アプリを使用すると、Linux VDAセッション内から変換されたPDFを印刷できます。セッションの印刷ジョブは、Citrix Workspaceアプリがインストールされているローカルマシンに送信されます。ローカルマシンでは、任意のPDFビューアを使用してPDFを開き、任意のプリンターで印刷できます。

Linux VDAは、以下のバージョンのCitrix WorkspaceアプリでPDF印刷をサポートしています。

- Citrix Receiver for HTML5 バージョン2.4から2.6.9、Citrix Workspaceアプリ1808 for HTML5以降
- Citrix Receiver for Chrome バージョン2.4から2.6.9、Citrix Workspaceアプリ1808 for Chrome以降
- Citrix Workspaceアプリ1905 for Windows以降

構成

PDF印刷をサポートするバージョンのCitrix Workspaceアプリを使用する以外に、Citrix Studioで以下のポリシーを有効にしてください。

- **クライアントプリンターリダイレクト** (デフォルトで有効)
- **PDFユニバーサルプリンターの自動作成** (デフォルトで無効)

これらのポリシーを有効にすると、起動したセッション内で**印刷**をクリックしたときに、プリンターを選択するための印刷プレビューがローカルマシンに表示されます。デフォルトプリンターの設定については、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

この記事では、Linux VDA のグラフィック構成と微調整に関するガイダンスを提供します。

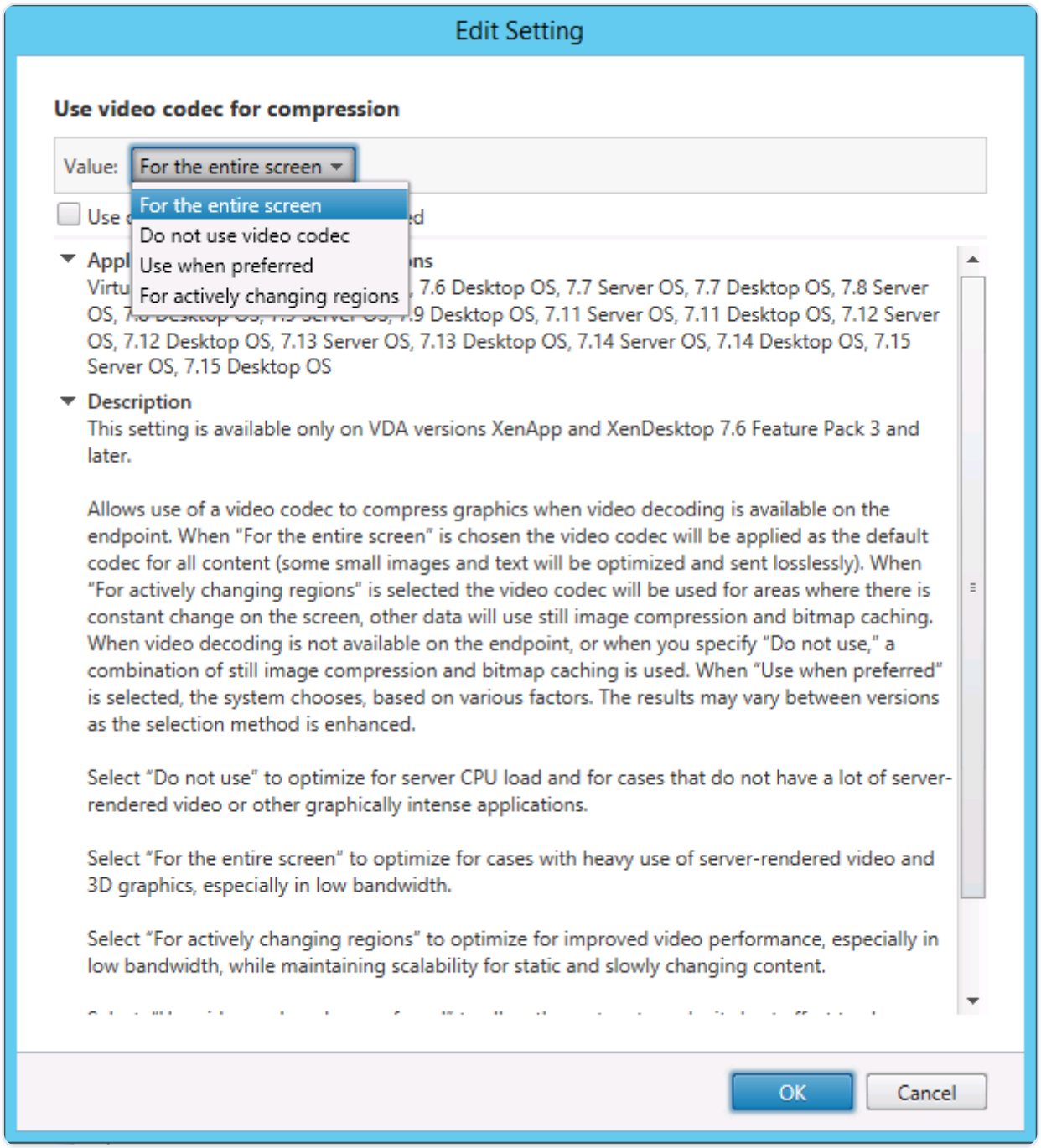
詳細については、システム要件とインストール概要のセクションを参照してください。

構成

Thinwire は、Linux VDA で使用されるディスプレイリモート処理テクノロジーです。このテクノロジーにより、あるマシンで生成されたグラフィックを、通常はネットワーク経由で別のマシンに送信して表示できます。

圧縮にビデオコーデックを使用グラフィックポリシーは、デフォルトのグラフィックモードを設定し、さまざまなユースケースに対して以下のオプションを提供します。

- **優先する場合にのみ使用**。この設定はデフォルトです。追加の構成は不要です。この設定を維持することで、すべてのCitrix®接続でThinwireが選択され、一般的なデスクトップワークロードのスケーラビリティ、帯域幅、および優れた画質のために最適化されます。
- **画面全体**。Thinwire をフルスクリーン H.264 または H.265 で配信し、特に 3D グラフィックを多用する場合のユーザーエクスペリエンスと帯域幅の向上を最適化します。
- **アクティブに変更される領域**。Thinwire の適応型ディスプレイテクノロジーは、動く画像（ビデオ、移動中の 3D）を識別し、画像が動いている画面の部分でのみ H.264 を使用します。**H.264 ビデオコーデックの選択的な使用**により、HDX Thinwire は、ビデオコンテンツなど、H.264 ビデオコーデックを使用して頻繁に更新される画面の一部を検出し、エンコードできます。静止画圧縮（JPEG、RLE）とビットマップキャッシュは、テキストや写真画像を含む画面の残りの部分で引き続き使用されます。ユーザーは、ビデオコンテンツの帯域幅の低減と品質の向上に加えて、ロスレステキストまたは他の場所での高品質画像の恩恵を受けます。この機能を有効にするには、ポリシー設定 **圧縮にビデオコーデックを使用** を **優先する場合にのみ使用**（デフォルト）または **アクティブに変更される領域** に変更します。詳細については、グラフィックポリシー設定を参照してください。



以下の視覚表示ポリシー設定を含む、その他のポリシー設定を使用して、ディスプレイリモート処理のパフォーマンスを微調整できます。

- **簡易グラフィックの優先色深度**
- **ターゲットフレームレート**
- **視覚的品質**

Thinwire の Build to Lossless に H.264 を使用

デフォルトでは、**視覚的品質**ポリシー設定の**Build to Lossless**の優先設定は、動画の場合、JPEG ではなく H.264 になりました。

H.264エンコーディングは、優れた画質を提供します。**Use video codec for compression**ポリシーは、その設定を制御し、デフォルトは**Use when preferred**です。**Build to Lossless**でJPEGを使用するように強制するには、**Use video codec for compression**ポリシーを**Do not use video codec**に設定します。クライアントがSelective H.264をサポートしていない場合、ポリシー設定に関係なく、**Build to Lossless**はJPEGにフォールバックします。Citrix Receiver for Windows 4.9〜4.12、Citrix Receiver for Linux 13.5〜13.10、Citrix Workspaceアプリ1808 for Windows以降、およびCitrix Workspaceアプリ1808 for Linux以降は、Selective H.264をサポートしています。**Visual quality**および**Use video codec for compression**ポリシー設定について詳しくは、「Visual display policy settings」および「Graphics policy settings」を参照してください。

H.265ビデオコーデックのサポート

7.18リリース以降、Linux VDAはリモートグラフィックとビデオのハードウェアアクセラレーションのためにH.265ビデオコーデックをサポートしています。この機能は、Citrix Receiver for Windows 4.10〜4.12、およびCitrix Workspaceアプリ1808 for Windows以降で使用できます。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する必要があります。クライアントのGPUがDXVAインターフェイスを使用したH.265デコードをサポートしていない場合、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フォールバックします。詳しくは、「H.265 video encoding」を参照してください。

VDAでH.265ハードウェアエンコーディングを有効にするには：

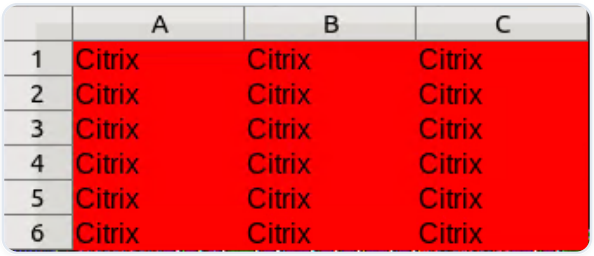
1. **Use hardware encoding for video codec**ポリシーを有効にします。
2. **Optimize for 3D graphics workload**ポリシーを有効にします
3. **Use video codec for compression**ポリシーがデフォルトであるか、**For the entire screen**に設定されていることを確認します。
4. **Visual quality**ポリシーが**Build to Lossless**または**Always Lossless**に**設定されていない**ことを確認します。

クライアントでH.265ハードウェアエンコーディングを有効にするには、「H.265 video encoding」を参照してください。

YUV444ソフトウェアエンコーディングのサポート

Linux VDAはYUV444ソフトウェアエンコーディングをサポートしています。YUVエンコーディングスキームは、各ピクセルに輝度値と色値の両方を割り当てます。YUVでは、「`Y`」が輝度または「ルミナンス」値を表し、「`UV`」が色または「クロミナンス」値を表します。このLinux VDAの機能は、Citrix Receiver for Windows 4.10〜4.12、およびCitrix Workspaceアプリ1808 for Windows以降で使用できます。

各Y、U、Vの一意の値は、8ビット、つまり1バイトのデータで構成されます。YUV444データ形式は、ピクセルあたり24ビットを送信します。YUV422データ形式は、2つのピクセル間でU値とV値を共有し、その結果、ピクセルあたり平均16ビットの伝送速度になります。次の表は、YUV444とYUV420の直感的な比較を示しています。

YUV444	YUV420
	yuv420の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/yuv420.png)

VDAでYUV444ソフトウェアエンコーディングを有効にするには：

1. **圧縮にビデオコーデックを使用**ポリシーが**画面全体**に設定されていることを確認します。
2. **視覚品質**ポリシーが**常にロスレス**または**ビルドトゥロスレス**に設定されていることを確認します。

帯域幅の見積もりに基づいて平均ビットレートを調整する

Citrixは、帯域幅の見積もりに基づいて平均ビットレートを調整することにより、HDX™ 3D Proハードウェアエンコーディングを強化します。

HDX 3D Proハードウェアエンコーディングが使用されている場合、VDAはネットワークの帯域幅を断続的に推定し、帯域幅の見積もり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。この新機能は、シャープネスと流暢さのバランスを取るメカニズムを提供します。

この機能はデフォルトで有効になっています。無効にするには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "DisableReconfigureEncoder" -d "0x00000001" --force
```

この機能を使用するだけでなく、次のコマンドを実行してシャープネスと流暢さのバランスを調整することもできます。**AverageBitRatePercent**および**MaxBitRatePercent**パラメーターは、帯域幅使用率のパーセンテージを設定します。設定値が高いほど、グラフィックはシャープになり、流暢さは低下します。推奨設定範囲は50～100です。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "AverageBitRatePercent" -d "90" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "MaxBitRatePercent" -d "100" --force
```

平均ビットレート調整では、画面が静止している場合、新しいフレームが送信されないため、最新のフレームは低品質の状態を維持します。シャープ化サポートは、最新のフレームを最高品質で再構成してすぐに送信することにより、この問題に対処できます。

Linux VDA Thinwireでサポートされているポリシーの完全なリストについては、「ポリシーサポートリスト」を参照してください。

Linux VDAでのマルチモニターサポートの構成については、「CTX220128」を参照してください。

トラブルシューティング

使用中のグラフィックモードを確認する

使用中のグラフィックモードを確認するには、次のコマンドを実行します（**0**はTW+、**1**は全画面ビデオコーデックを意味します）。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000000" --force`

H.264が使用中であるかを確認する

H.264が使用中であるかを確認するには、次のコマンドを実行します (**0** は使用中でないことを意味し、**1** は使用中であることを意味します)。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000000" --force
```

H.265が使用中であるかを確認する

全画面H.265が使用中であるかを確認するには、次のコマンドを実行します (**0** は使用中でないことを意味し、**1** は使用中であることを意味します)。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H265" -d "0x00000000" --force
```

どのYUVエンコードスキームが使用中であるかを確認する

どのYUVエンコードスキームが使用中であるかを確認するには、次のコマンドを実行します (**0** はYUV420を意味します。**1** はYUV422を意味します。**2** はYUV444を意味します)。

注： YUVFormat の値は、ビデオコーデックが使用されている場合にのみ意味があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000000" --force
```

YUV444ソフトウェアエンコードが使用中であるかを確認する

YUV444ソフトウェアエンコードが使用中であるかを確認するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
```

YUV444が使用中の場合、結果は次のようになります。

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "GraphicsMode" -d "0x00000001" --force
```

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "H264" -d "0x00000001" --force` `create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000000" --force`

`create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics" -t "REG_DWORD" -v "YUVFormat" -d "0x00000002" --force`

3D Proでハードウェアエンコードが使用中であるかを確認する

次のコマンドを実行します (**0** は使用中ではないことを意味し、**1** は使用中であることを意味します):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
```

結果は次のようになります。

`create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics" -t "REG_DWORD" -v "HardwareEncoding" -d "0x00000001" --force`

別の方法は、**nvidia-smi** コマンドを使用することです。ハードウェアエンコーディングが使用されている場合、出力は次のようになります。

```
Tue Apr 12 10:42:03 2016

+-----------------------------------------------------+

| NVIDIA-SMI 361.28     Driver Version: 361.28         |

|-------------------------------+----------------------
+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
==============================+======================+==================
===|

|   0  GRID K1             Off  | 0000:00:05.0     Off |
```

```
N/A |

| N/A   42C    P0    14W /  31W |    207MiB /  4095MiB |      8%
Default |

+-------------------------------+----------------------
+----------------------+




+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|

|    0      2164  C+G   /usr/local/bin/ctxgfx
106MiB |

|    0      2187    G   Xorg
85MiB |

+-----------------------------------------------------------------------------
----+
```

NVIDIA GRIDグラフィックドライバーが正しくインストールされていることを確認する

NVIDIA GRIDグラフィックドライバーが正しくインストールされていることを確認するには、**nvidia-smi** を実行します。結果は次のようになります。

```
+-----------------------------------------------------+

| NVIDIA-SMI 352.70     Driver Version: 352.70         |

|-------------------------------+----------------------
+----------------------+
```

```
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |

| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |

|
==============================+======================+==================
===|

|   0  Tesla M60           Off  | 0000:00:05.0     Off |
Off |

| N/A   20C    P0    37W / 150W |     19MiB /  8191MiB |      0%
Default |

+-------------------------------+----------------------
+----------------------+



+-----------------------------------------------------------------------------
----+

| Processes:                                                       GPU
Memory |

|  GPU       PID  Type  Process name
Usage      |

|
=============================================================================
===|

|  No running processes found
                                 |

+-----------------------------------------------------------------------------
----+
```

カードの正しい構成を設定します。

`etc/X11/ctx-nvidia.sh`

HDX 3D Proマルチモニターの再描画に関する問題

プライマリモニター以外の画面で再描画の問題が発生している場合は、NVIDIA GRIDライセンスが利用可能であることを確認してください。

Xorgエラーログを確認する

Xorgのログファイルは、**/var/log/** フォルダーに **Xorg.{DISPLAY}.log** のように命名されています。

既知の問題と制限事項

vGPUの場合、Citrix Hypervisor™ローカルコンソールにICAデスクトップセッション画面が表示される

回避策: 次のコマンドを実行して、VMのローカルVGAコンソールを無効にします。

Citrix Hypervisor 8.1以降の場合:

```
[root@xenserver® ~]# xe vgpu-param-set uuid=vgpu-uuid
extra_args=disable_vnc=1
```

Citrix Hypervisor 8.1より前の場合:

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
```

NVIDIA K2グラフィックカードは、パススルーモードでのYUV444ハードウェアエンコーディングをサポートしていません

ポリシー設定で**Build to Lossless**が有効になっている場合、NVIDIA K2グラフィックカードを使用してユーザーがアプリ/デスクトップセッションを起動すると、黒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ていないために発生します。詳細については、Video Encode and Decode GPU Support Matrixを参照してください。

ログオン時にGnome 3デスクトップのポップアップが遅くなる

これはGnome 3デスクトップセッション起動の制限です。

Citrix Workspaceアプリのウィンドウのサイズを変更すると、一部のOpenGL/WebGLアプリケーションが適切に描画されないことがあります

Citrix Workspaceアプリのウィンドウのサイズを変更すると、画面解像度が変更されます。NVIDIA独自のドライバーは一部の内部状態を変更し、アプリケーションがそれに応じて応答する必要がある場合があります。たとえば、WebGLライブラリ要素**lightgl.js**は、「`Rendering to this texture is not supported (incomplete frame buffer)`」というエラーを生成する可能性があります。

Thinwire プログレッシブ表示

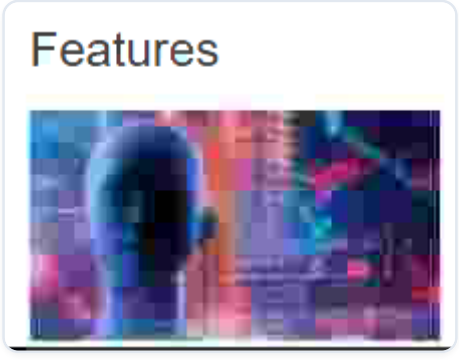
低帯域幅または高遅延の接続では、セッションの対話性が低下する可能性があります。たとえば、帯域幅が2 Mbps未満または遅延が200 msを超える接続では、Webページのスクロールが遅くなったり、応答しなくなったり、途切れたりすることがあります。キーボードとマウスの操作がグラフィックの更新に遅れることがあります。

バージョン7.17までは、ポリシー設定を使用して、セッションを低視覚品質に構成するか、より低い色深度（16ビットまたは8ビットグラフィック）を設定することで、帯域幅の消費を削減できました。ただし、ユーザーが弱い接続を使用していることを知っている必要がありました。HDX Thinwireは、ネットワークの状態に基づいて静止画の品質を動的に調整しませんでした。

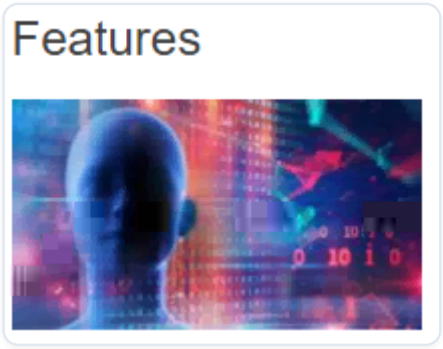
バージョン7.18以降、HDX Thinwireは、利用可能な帯域幅が2 Mbpsを下回るか、ネットワーク遅延が200 msを超えると、デフォルトでプログレッシブ更新モードに切り替わります。このモードでは、次のようになります。

- すべての静止画が高度に圧縮されます。
- テキスト品質が低下します。

たとえば、プログレッシブ更新モードがアクティブな以下のグラフィックでは、文字**F**と**e**に青いアーティファクトがあり、画像は高度に圧縮されています。このアプローチにより、帯域幅の消費が大幅に削減され、画像とテキストをより迅速に受信できるようになり、セッションの対話性が向上します。



セッションとの対話を停止すると、劣化した画像とテキストは徐々にシャープ化され、ロスレスになります。たとえば、以下のグラフィックでは、文字に青いアーティファクトが含まれなくなり、画像は元の品質で表示されます。



画像の場合、シャープ化にはランダムなブロック状の方法が使用されます。テキストの場合、個々の文字または単語の一部がシャープ化されます。シャープ化プロセスは複数のフレームにわたって行われます。このアプローチにより、単一の大きなシャープ化フレームによる遅延の発生が回避されます。

一時的な画像（ビデオ）は、引き続きアダプティブディスプレイまたはSelective H.264で管理されます。

プログレッシブモードの使用方法

デフォルトでは、プログレッシブモードは**視覚品質**ポリシー設定の**高**、**中**（デフォルト）、**低**に対してスタンバイ状態です。

プログレッシブモードは、次の場合に強制的にオフ（使用されない）になります。

- **視覚品質** = **常にロスレス** または **ロスレスに構築**
- **単純なグラフィックの優先色深度** = 8ビット
- **圧縮にビデオコーデックを使用** = **画面全体に対して** (全画面H.264が必要な場合)

プログレッシブモードがスタンバイ状態の場合、デフォルトでは、以下のいずれかの条件が発生すると有効になります。

- 利用可能な帯域幅が2 Mbpsを下回る
- ネットワーク遅延が200 msを超える

モード切り替えが発生した後、たとえネットワークの状態が悪くても一時的なものであっても、そのモードで最低10秒間費やされます。

プログレッシブモードの動作を変更する

次のコマンドを実行して、プログレッシブモードの動作は変更できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplay" -d "<value>" --force
```

ここで \<value\>:

0 = 常にオフ (いかなる状況でも使用しないでください)

1 = 自動 (ネットワークの状態に基づいて切り替え、デフォルト値)

2 = 常にオン

自動モード (1) の場合、次のいずれかのコマンドを実行して、プログレッシブモードが切り替わるしきい値を変更できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

ここで \<value\> は \<threshold in Kbps\> (デフォルト = 2,048)

例: 4096 = 帯域幅が4 Mbpsを下回った場合にプログレッシブモードをオンにする

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t
"REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

ここで、\<value\> は \<threshold in ms\> です（デフォルト = 200）

例：100 = ネットワーク遅延が100ミリ秒を下回った場合にプログレッシブモードを有効にする。

非GRID 3Dグラフィック

概要

この機能強化により、Linux VDAはNVIDIA GRID 3Dカードだけでなく、非GRID 3Dカードもサポートします。

インストール

非GRID 3Dグラフィック機能を使用するには、次の操作を行う必要があります。

- 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ます。
- Linux VDAのインストール時に、`CTX_XDL_HDX_3D_PRO`を`Y`に設定します。環境変数について詳しくは、「手順7：インストールを完了するためのランタイム環境のセットアップ」を参照してください。

構成

Xorg構成ファイル

3Dカードドライバーが NVIDIAの場合、構成ファイルは自動的にインストールおよび設定されます。

その他の種類の3Dカード

3Dカードドライバーが NVIDIAではない場合、`/etc/X11/`にインストールされている4つのテンプレート構成ファイルを変更する必要があります。

- ctx-driver_name-1.conf
- ctx-driver_name-2.conf
- ctx-driver_name-3.conf
- ctx-driver_name-4.conf

ctx-driver_name-1.confを例として、テンプレート構成ファイルを変更するには、次の手順を実行します。

1. 「***driver_name***」を実際のドライバー名に置き換えます。

   たとえば、ドライバー名が `intel` の場合、構成ファイル名を `ctx-intel-1`.conf に変更できます。

2. ビデオドライバー情報を追加します。

   各テンプレート構成ファイルには、「Device」という名前のセクションが含まれており、コメントアウトされています。このセクションでは、ビデオドライバー情報について説明しています。ビデオドライバー情報を追加する前に、このセクションを有効にしてください。このセクションを有効にするには：

a. 構成情報については、製造元が提供する3Dカードガイドを参照してください。ネイティブ構成ファイルを生成できます。Linux VDA ICA®セッションを使用していないときに、3Dカードがネイティブ構成ファイルを使用してローカル環境で動作することを確認してください。

b. ネイティブ構成ファイルの「Device」セクションを **ctx-driver_name-1.conf** にコピーします。

3. Linux VDAが手順1で設定された構成ファイル名を認識できるように、次のコマンドを実行してレジストリキーを設定します。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_SZ" -v "DriverName" -d "intel" --force
```

非GRID 3Dグラフィック機能を有効にする

非GRID 3Dグラフィック機能はデフォルトで無効になっています。XDamageEnabledを1に設定することで、次のコマンドを実行して有効にできます。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t
"REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

トラブルシューティング

グラフィック出力がない、または文字化けしている

3Dアプリケーションをローカルで実行でき、すべての構成が正しい場合、グラフィック出力がない、または文字化けしているのはバグが原因です。デバッグのためにトレース情報を収集するには、/opt/Citrix/VDA/bin/setlog を使用し、GFX_X11 を verbose に設定してください。

ハードウェアエンコードが機能しない

この機能はソフトウェアエンコードのみをサポートしています。

ポリシーを構成する

インストール

Linux VDAを準備するには、インストール記事に従ってください。

依存関係

Linux VDAパッケージをインストールする前に、これらの依存関係をインストールしていることを確認してください。

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
```

SLES/SELD:

```
sudo zypper install openldap2


sudo zypper install libxml2


sudo zypper install cyrus-sasl


sudo zypper install cyrus-sasl-gssapi
```

ウブントゥ:

```
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
```

構成

Citrix Studioでのポリシー設定

Citrix Studioでポリシーを設定するには、次の手順を実行します。

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. ポリシーサポートリストに従ってポリシーを設定します。

VDA上のLDAPサーバー設定

Linux VDA上のLDAPサーバー設定は、単一ドメイン環境ではオプションですが、複数ドメインおよび複数フォレスト環境では必須です。この設定は、ポリシーサービスがこれらの環境でLDAP検索を実行するために必要です。

Linux VDAパッケージのインストール後、次のコマンドを実行します。

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

推奨される形式ですべてのLDAPサーバーを入力します。LDAPポートを含むLDAP完全修飾ドメイン名（FQDN）のスペース区切りリスト（例：ad1.mycompany.com:389 ad2.mycomany.com:389）。

Linux VDA上のLDAPサーバー設定の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/ldap-server-setting-on-linux-vda-new.png)

この設定をレジストリに直接書き込むには、**ctxreg**コマンドを実行することもできます。

```
/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
```

ポリシーサポートリスト

Linux VDA ポリシーサポートリスト

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントのローカル時刻を使用する	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ラウンドトリップ計算	IcaRoundTripCheckEnabled	コンピューター	ICA\エンドユーザー監視	有効 (1)
ICAラウンドトリップ計算間隔	IcaRoundTripCheckPeriod	コンピューター	ICA\エンドユーザー監視	15
アイドル接続のICAラウンドトリップ計算	IcaRoundTripCheckWhenIdle	コンピューター	ICA\エンドユーザー監視	無効 (0)
セッション全体の帯域幅制限	LimitOverallBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限	LimitAudioBw	ユーザー	ICA\帯域幅	0
オーディオリダイレクトの帯域幅制限（パーセント）	LimitAudioBwPercent	ユーザー	ICA\帯域幅	0
クライアントUSBデバイスリダイレクトの帯域幅制限	LimitUSBBw	ユーザー	ICA\帯域幅	0

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントUSBデバイスリダイレクト帯域幅の割合	LimitUSBBwPercent	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限	LimitClipbdBW	ユーザー	ICA\帯域幅	0
クリップボードリダイレクトの帯域幅制限（パーセント）	LimitClipbdBWPercent	ユーザー	ICA\帯域幅	0
ファイルリダイレクト帯域幅制限	LimitCdmBw	ユーザー	ICA\帯域幅	0
ファイルリダイレクトの帯域幅制限（パーセント）	LimitCdmBwPercent	ユーザー	ICA\帯域幅	0
プリンターリダイレクト帯域幅制限	LimitPrinterBw	ユーザー	ICA\帯域幅	0
プリンターリダイレクトの帯域幅制限（パーセント）	LimitPrinterBwPercent	ユーザー	ICA\帯域幅	0
WebSockets接続	AcceptWebSocketsConnections	コンピューター	ICA\WebSockets	禁止
WebSocketsポート番号	WebSocketsPort	コンピューター	ICA\WebSockets	8008

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
WebSockets信頼済みオリジンサーバーリスト	WSTrustedOriginServerList	コンピューター	ICA\WebSockets	*
ICA® キープアライブ	SendICAKeepAlives	コンピューター	ICAキープアライブ	ICAキープアライブメッセージを送信しない (0)
ICAキープアライブタイムアウト	ICAKeepAliveTimeout	コンピューター	ICAキープアライブ	60秒
ICAリスナーポート番号	IcaListenerPortNumber	コンピューター	ICA	1494
HDX™ アダプティブトランスポート	HDXoverUDP	コンピューター	ICA	優先 (2)
セッションの信頼性接続	AcceptSessionReliabilityConnections	コンピューター	ICA\セッションの信頼性	許可 (1)
再接続UIの透過レベル	ReconnectionUiTransparencyLevel	コンピューター	ICA\自動クライアント再接続	80%
セッションの信頼性ポート番号	SessionReliabilityPort	コンピューター	ICA\セッションの信頼性	2598
セッションの信頼性タイムアウト	SessionReliabilityTimeout	コンピューター	ICA\セッションの信頼性	180s
クライアントの自動再接続	AllowAutoClientReconnect	ユーザー	ICA\自動クライアント再接続	許可 (1)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
クライアントオーディオリダイレクト	AllowAudioRedirection	ユーザー	オーディオ	許可 (1)
クライアントプリンターリダイレクト	AllowPrinterRedir	ユーザー	印刷	許可 (1)
PDFユニバーサルプリンター自動作成	PDFプリンターの自動作成	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	ドライバーマッピングリスト	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	クリップボードリダイレクトを許可	ユーザー	クリップボード	許可 (1)
クライアントUSBデバイスのリダイレクト	USBリダイレクトを許可	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	USBデバイス規則	ユーザー	USB	“\0”
動画圧縮	動画圧縮構成	ユーザー	シンワイヤー	有効 (1)
追加の色圧縮	追加の色圧縮	ユーザー	シンワイヤー	無効 (0)
目標最小フレームレート	目標最小フレーム/秒	ユーザー	シンワイヤー	10 フレーム/秒

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
ターゲットフレームレート	フレーム/秒	ユーザー	シンワイヤー	30 フレーム/秒
視覚的な品質	VisualQuality	ユーザー	シンワイヤー	中 (3)
圧縮にビデオコーデックを使用	VideoCodec	ユーザー	シンワイヤー	優先する場合に使用 (3)
ビデオコーデックにハードウェアエンコードを使用	UseHardwareEncodingForVideoCodec	ユーザー	シンワイヤー	有効 (1)
視覚的にロスレスな圧縮を許可する	AllowVisuallyLosslessCompression	ユーザー	シンワイヤー	無効 (0)
3Dグラフィックワークロード向けに最適化	OptimizeFor3dWorkload	ユーザー	シンワイヤー	無効 (0)
単純なグラフィックの優先色深度	PreferredColorDepth	ユーザー	シンワイヤー	24 ビット/ピクセル (1)
オーディオ品質	SoundQuality	ユーザー	オーディオ	高 – 高精細オーディオ (2)
クライアントマイクのリダイレクト	AllowMicrophoneRedir	ユーザー	オーディオ	許可 (1)
最大セッション数	MaximumNumberOfSessions	コンピューター	負荷管理	250
同時ログオン許容数	ConcurrentLogonsTolerance	コンピューター	負荷管理	2

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
コントローラーの自動更新を有効にする	EnableAutoUpdateOfControllers	コンピューター	仮想デリバリーエージェント設定	許可 (1)
クリップボード選択更新モード	ClipboardSelectionUpdateMode	ユーザー	クリップボード	3
プライマリ選択更新モード	PrimarySelectionUpdateMode	ユーザー	クリップボード	3
最大Speex品質	MaxSpeexQuality	ユーザー	オーディオ	5
クライアントドライブの自動接続	AutoConnectDrives	ユーザー	ファイルリダイレクト/CDM	有効 (1)
クライアント光学ドライブ	AllowCdromDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアント固定ドライブ	AllowFixedDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントフロッピードライブ	AllowFloppyDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントネットワークドライブ	AllowNetworkDrives	ユーザー	ファイルリダイレクト/CDM	許可 (1)
クライアントドライブのリダイレクト	AllowDriveRedir	ユーザー	ファイルリダイレクト/CDM	許可 (1)
読み取り専用クライアントドライブアクセス	ReadOnlyMappedDrive	ユーザー	ファイルリダイレクト/CDM	無効 (0)

STUDIO ポリシー	キー名	タイプ	モジュール	デフォルト値
自動キーボード表示	AllowAutoKeyboardPopUp	ユーザー	MRVC	無効 (0)
デスクトップとクライアント間のファイル転送を許可する	AllowFileTransfer	ユーザー	ファイル転送	許可
デスクトップからファイルをダウンロードする	AllowFileDownload	ユーザー	ファイル転送	許可
デスクトップにファイルをアップロードする	AllowFileUpload	ユーザー	ファイル転送	許可

以下のポリシーは、Citrix Studio Version 7.12以降で構成できます。

- MaxSpeexQuality

  値 (整数):

  デフォルト値: 5

  詳細:

  オーディオ品質が中または低の場合（ポリシー「オーディオ品質」を参照）、オーディオリダイレクトはSpeexコーデックでオーディオデータをエンコードします。Speexは非可逆コーデックであり、入力音声信号の忠実度を犠牲にして圧縮を実現します。他の音声コーデックとは異なり、品質とビットレートの間のトレードオフを制御することが可能です。Speexのエンコードプロセスは、ほとんどの場合、0から10までの範囲の品質パラメーターによって制御されます。品質が高いほど、ビットレートも高くなります。

  最大Speex品質は、オーディオ品質と帯域幅制限（ポリシー「オーディオリダイレクト帯域幅制限」を参照）に応じて、オーディオデータをエンコードするための最適なSpeex品質を選択します。オーディオ品質が中の場合、エンコーダーは広帯域モードになり、サンプリングレートが高くなります。オーディオ品質が低い場合、エンコーダーは狭帯域モードになり、サンプリングレートが低くなります。同じSpeex品質でも、モードによってビットレートが異なります。最適なSpeex品質は、最大値が以下の条件を満たす場合です。

- 最大Speex品質以下であること。
- そのビットレートが帯域幅制限以下であること。

関連設定: オーディオ品質、オーディオリダイレクト帯域幅制限

- PrimarySelectionUpdateMode

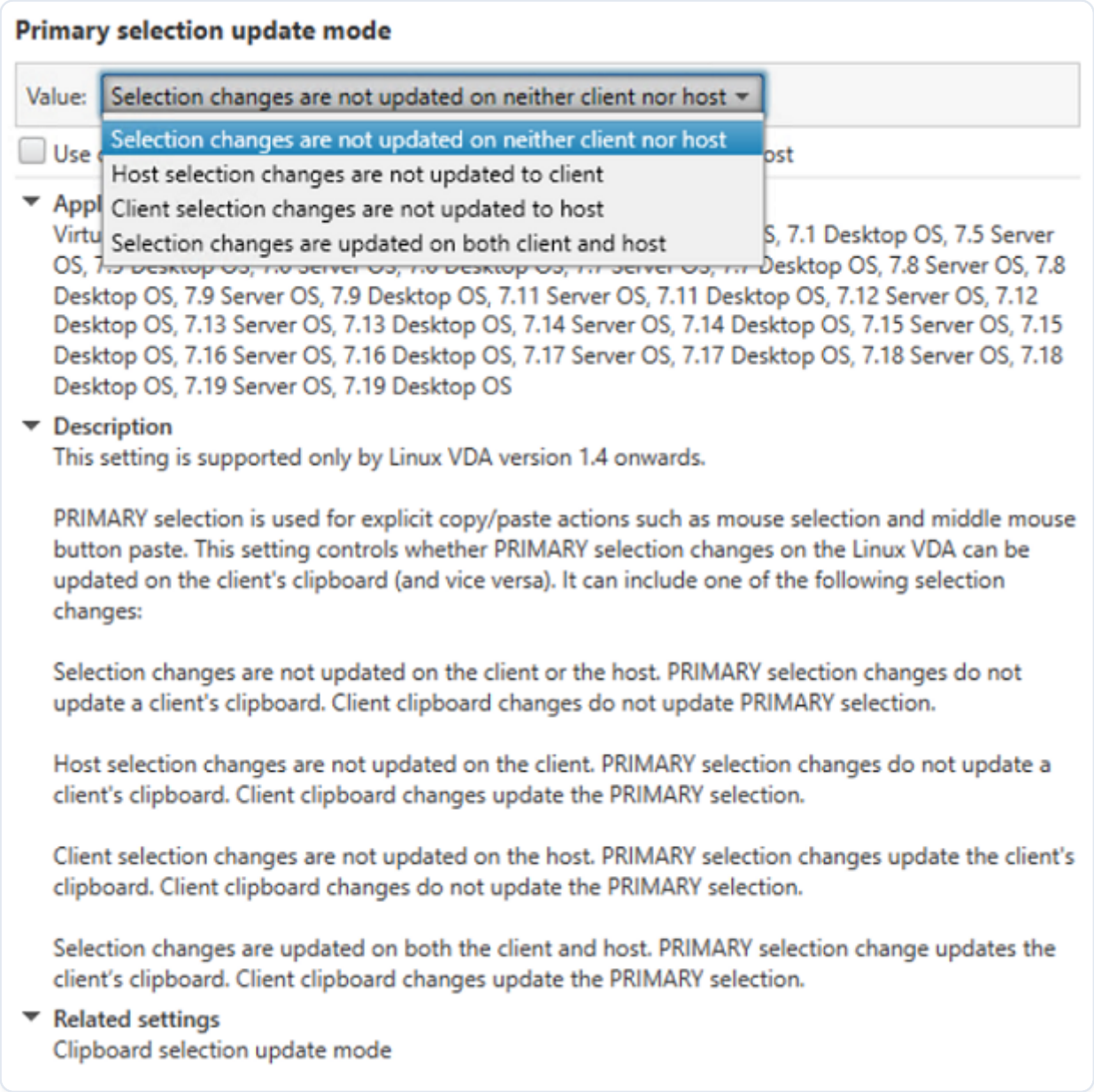
  値 (列挙型):

  デフォルト値: 3

  詳細:

  プライマリ選択は、データを選択し、マウスの中央ボタンを押して貼り付けるときに使用されます。

  このポリシーは、Linux VDAとクライアントでのプライマリ選択の変更が、互いのクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。



- **選択の変更はクライアントにもホストにも更新されない**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されない**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新しません。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されない**

Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新しません。

- **選択の変更はクライアントとホストの両方で更新される**

  Linux VDAでのプライマリ選択の変更は、クライアントのクリップボードを更新します。クライアントでのプライマリ選択の変更は、Linux VDAのクリップボードを更新します。このオプションがデフォルト値です。

関連設定: クリップボード選択更新モード

- ClipboardSelectionUpdateMode

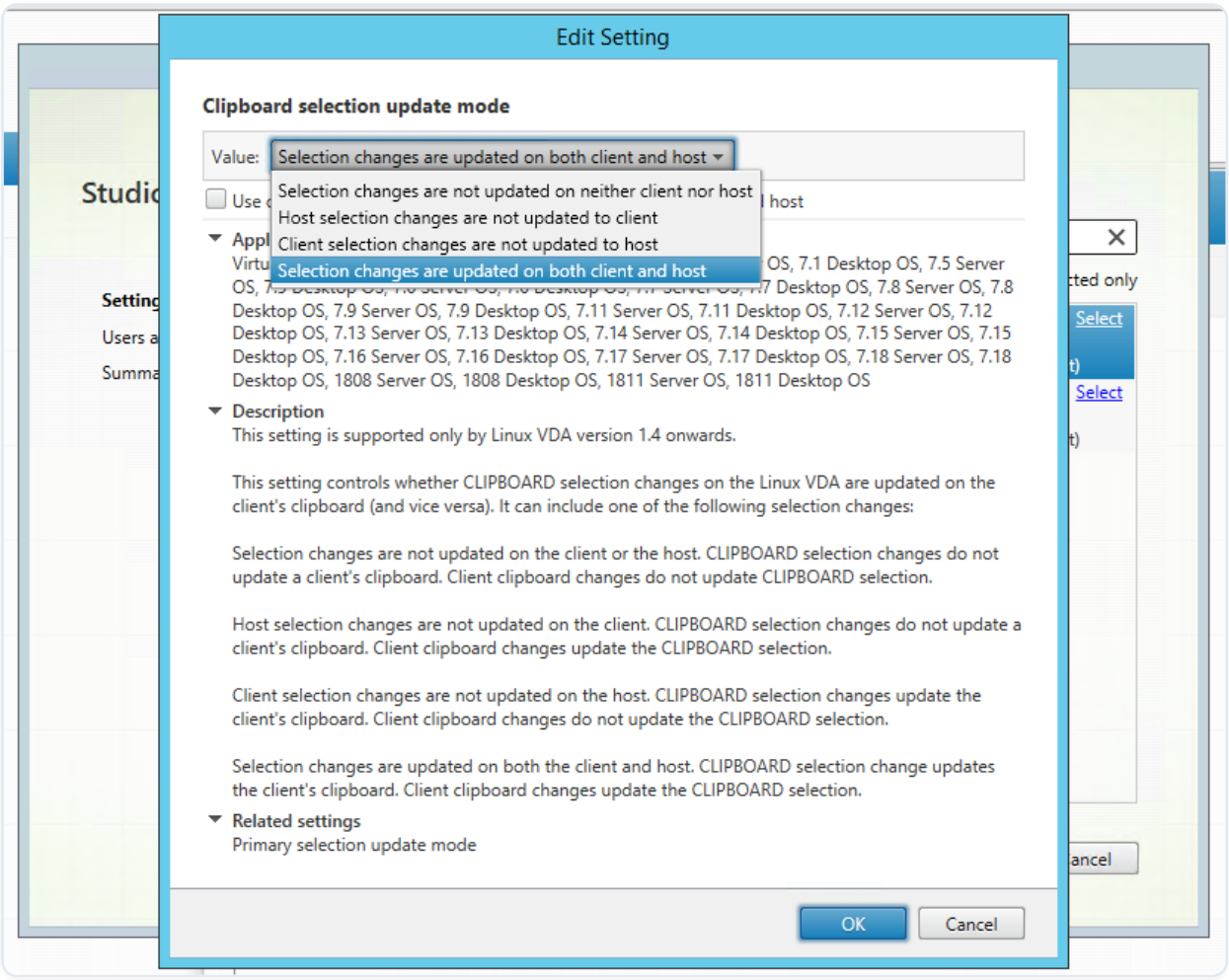
  値（列挙型）:

  デフォルト値: 3

  詳細:

  クリップボード選択は、データを選択し、ショートカットメニューから「コピー」を選択するなどして、明示的にクリップボードに「コピー」を要求するときに使用されます。クリップボード選択は主にMicrosoft Windowsのクリップボード操作に関連して使用されますが、プライマリ選択はLinuxに固有のものです。

  このポリシーは、Linux VDAとクライアントでのクリップボード選択の変更が、互いのクリップボードを更新できるかどうかを制御します。4つの値オプションがあります。



- **クライアントとホストのどちらにも選択の変更は更新されません**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **ホストの選択の変更はクライアントに更新されません**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新しません。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。

- **クライアントの選択の変更はホストに更新されません**

  Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新しません。

- **クライアントとホストの両方で選択の変更が更新されます**

Linux VDAでのクリップボード選択の変更は、クライアントのクリップボードを更新します。クライアントでのクリップボード選択の変更は、Linux VDAのクリップボードを更新します。このオプションがデフォルト値です。

関連設定: プライマリ選択更新モード

注:

Linux VDAは、クリップボード選択とプライマリ選択の両方をサポートしています。Linux VDAとクライアント間のコピー＆ペーストの動作を制御するには、Citrix®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推奨します。

IPv6の構成

Linux VDAは、Citrix Virtual Apps and Desktopsに合わせてIPv6をサポートしています。この機能を使用する際は、以下を考慮してください。

- デュアルスタック環境では、IPv6が明示的に有効になっていない限り、IPv4が使用されます。
- IPv4環境でIPv6が有効になっている場合、Linux VDAは機能しません。

重要:

- ネットワーク環境全体がIPv6である必要があり、Linux VDAだけではありません。
- Centrifyは純粋なIPv6をサポートしていません。

Linux VDAをインストールする際に、IPv6のための特別なセットアップタスクは必要ありません。

Linux VDAのIPv6を構成する

Linux VDAの構成を変更する前に、Linux仮想マシンが以前にIPv6ネットワークで動作していたことを確認してください。IPv6構成に関連するレジストリキーは2つあります。

```
“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“OnlyUseIPv6ControllerRegistration”

“HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD” -v
“ControllerRegistrationIPv6Netmask”
```

Linux VDAでIPv6を有効にするには、**OnlyUseIPv6ControllerRegistration**を1に設定する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v
"OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

Linux VDAに複数のネットワークインターフェイスがある場合、**ControllerRegistrationIPv6Netmask**を使用して、Linux VDAの登録に使用するインターフェイスを指定できます。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v
"ControllerRegistrationIPv6Netmask " -d "{IPv6 netmask}" --force
```

{IPv6 netmask}を実際のネットマスク（例：2000::/64）に置き換えます。

Citrix Virtual Apps and DesktopsにおけるIPv6展開の詳細については、IPv4/IPv6のサポートを参照してください。

トラブルシューティング

基本的なIPv6ネットワーク環境を確認し、ping6を使用してADおよびDelivery Controller™に到達可能かどうかを確認します。

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

CEIPに参加すると、Citrix製品の品質とパフォーマンスの向上に役立てるため、匿名の統計情報と使用状況情報がCitrixに送信されます。さらに、匿名のデータのコピーは、迅速かつ効率的な分析のためにGoogle Analytics (GA) に送信されます。

レジストリ設定

デフォルトでは、Linux VDAをインストールすると、CEIPに自動的に参加します。データの初回アップロードは、Linux VDAのインストール後約7日で行われます。このデフォルト設定はレジストリで変更できます。

- **CEIPスイッチ**

CEIPを有効または無効にするレジストリ設定 (デフォルト = 0):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: CEIPSwitch

値: 1 = 無効、0 = 有効

指定しない場合、CEIPは有効になります。

クライアントで次のコマンドを実行してCEIPを無効にできます:

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

- **GAスイッチ**

GAを有効または無効にするレジストリ設定 (デフォルト = 0):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: GASwitch

値: 1 = 無効, 0 = 有効

指定しない場合、GAは有効になります。

クライアントで次のコマンドを実行して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"
```

- **DataPersistPath**

データ永続化パスを制御するレジストリ設定 (デフォルト = /var/xdl/ceip):

場所: HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\CEIP

名前: DataPersistPath

値: 文字列

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"
```

構成されたパスが存在しないか、アクセスできない場合、データはデフォルトパスに保存されます。

Linux VDAから収集されたCEIPデータ

次の表は、収集される匿名情報の種類の例を示しています。このデータには、お客様を特定できるような詳細は含まれていません。

データポイント	キー名	説明
マシンGUID	machine_guid	データの発生元となるマシンを識別する
ADソリューション	ad_solution	マシンのドメイン参加方法を示すテキスト文字列
Linuxカーネルバージョン	kernel_version	マシンのカーネルバージョンを示すテキスト文字列
LVDAバージョン	vda_version	インストールされているLinux VDAのバージョンを示すテキスト文字列。
LVDAの更新または新規インストール	update_or_fresh_install	現在のLinux VDAパッケージが新規インストールされているか、更新されているかを示すテキスト文字列
LVDAのインストール方法	install_method	現在のLinux VDAパッケージがMCS、PVS、簡易インストール、または手動インストールを使用してインストー

データポイント	キー名	説明
		ルされていることを示すテキスト文字列。
HDX 3D Proが有効か無効か	hdx_3d_pro	マシンでHDX 3D Proが有効になっているかどうかを示すテキスト文字列
VDIモードが有効か無効か	vdi_mode	VDIモードが有効になっているかどうかを示すテキスト文字列
システムロケール	system_locale	このマシンのロケールを示すテキスト文字列
LVDAキーサービスの最終再起動時刻	シーティーエックスエイチディーエックス シーティーエックスブイディーエー	`ctxhdx`および`ctxvda`サービスの最終再起動時刻。形式はdd-hh:mm:ss（例：10-17:22:19）
GPUタイプ	gpu_type	マシンのGPUタイプを示す
CPUコア数	cpu_cores	マシンのCPUコア数を示す整数
CPU周波数	cpu_frequency	MHz単位のCPU周波数を示す浮動小数点数
物理メモリサイズ	memory_size	KB単位の物理メモリサイズを示す整数
起動されたセッション数	session_launch	このデータポイントを収集した時点でマシン上で起動された（ログオンまたは再接続された）セッション数を示す整数
Linux OS名とバージョン	os_name_version	マシンのLinux OS名とバージョンを示すテキスト文字列
セッションキー	session_key	データの発信元となるセッションを識別します

データポイント	キー名	説明
リソースの種類	resource_type	起動されたセッションのリソースの種類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を保存するために使用されます。セッションは切断/再接続できるため、1つのセッションに複数のアクティブ時間がある場合があります
セッション継続時間	session_duration_time	ログオンからログオフまでのセッションの継続時間を保存するために使用されます
Receiverクライアントの種類	receiver_type	セッションの起動に使用されたCitrix Workspaceアプリのタイプを示す整数
Receiverクライアントバージョン	receiver_version	セッションの起動に使用されたCitrix Workspaceアプリのバージョンを示すテキスト文字列
印刷回数	printing_count	セッションが印刷機能を使用した回数を示す整数
USBリダイレクト回数	usb_redirecting_count	セッションがUSBデバイスを使用した回数を示す整数
Gfxプロバイダータイプ	gfx_provider_type	セッションのグラフィックプロバイダータイプを示すテキスト文字列
シャドウイング回数	shadow_count	セッションがシャドウされた回数を示す整数
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構成された長い文字列

データポイント	キー名	説明
スマートカードのリダイレクト回数	scard_redirecting_count	セッションログオンおよびセッション内アプリのユーザー認証にスマートカードのリダイレクトが使用された回数を示す整数

USBリダイレクトを構成する

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、ユーザーはローカルに接続されているかのようにUSBデバイスを使用できます。

USBリダイレクトには、主に次の3つの機能領域があります。

- オープンソースプロジェクトの実装（VHCI）
- VHCIサービス
- USBサービス

オープンソースVHCI：

USBリダイレクト機能のこの部分は、IPネットワークを介した汎用USBデバイス共有システムを開発します。これは、Linuxカーネルドライバーと、カーネルドライバーと通信してすべてのUSBデータを取得できるいくつかのユーザーモードライブラリで構成されています。Linux VDAの実装では、CitrixはVHCIのカーネルドライバーを再利用します。ただし、Linux VDAとCitrix Workspaceアプリ間のすべてのUSBデータ転送は、Citrix ICA®プロトコルパッケージにカプセル化されます。

VHCIサービス：

VHCIサービスは、Citrixが提供するオープンソースサービスで、VHCIカーネルモジュールと通信します。このサービスは、VHCIとCitrix USBサービス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デバイス上のすべての仮想化とデータ転送を管理するCitrixモジュールを表します。

USBリダイレクトの仕組み

通常、USBデバイスがLinux VDAに正常にリダイレクトされると、システム/devパスに1つ以上のデバイスノードが作成されます。ただし、リダイレクトされたデバイスがアクティブなLinux VDAセッションで使用できない場合があります。USBデバイスは、適切に機能するためにドライバーに依存しており、一部のデバイスは特別なドライバーを必要とします。ドライバーが提供されていない場合、リダイレクトされたUSBデバイスはアクティブなLinux VDAセッションからアクセスできません。USBデバイスの接続を確保するには、ドライバーをインストールし、システムを適切に構成してください。

Linux VDAは、クライアントとの間で正常にリダイレクトされるUSBデバイスのリストをサポートしています。さらに、デバイス、特にUSBディスクは適切にマウントされ、ユーザーは追加の構成なしでディスクにアクセスできます。

サポートされているUSBデバイス

以下のデバイスは、このバージョンのLinux VDAをサポートすることが確認されています。その他のデバイスも自由に使用できますが、予期せぬ結果が生じる可能性があります。

注:

Linux VDAはUSB 2.0プロトコルのみをサポートします。

USBマスストレージデバイス	VID:PID	ファイルシステム
ネタック テクノロジー株式会社	0dd8:173c	FAT32
キングストン データトラベラー 101 II	0951:1625	FAT32
キングストン データトラベラー GT101 G2	1567:8902	FAT32
SanDisk SDCZ80フラッシュドライブ	0781:5580	FAT32
WD HDD	1058:10B8	FAT32

USB 3Dマウス	VID:PID
3Dコネクション スペースマウス プロ	046d: c62b

USBスキャナー	VID:PID
エプソン パーフェクション V330 フォト	04B8: 0142

USBリダイレクトの構成

Citrixポリシーは、USBデバイスリダイレクトが有効か無効かを制御します。さらに、デバイスの種類はDelivery Controller™ポリシーを使用して指定することもできます。Linux VDAのUSBリダイレクトを構成する場合は、次のポリシーとルールを構成します。

- クライアントUSBデバイスリダイレクトポリシー
- クライアントUSBデバイスリダイレクト規則

USBリダイレクトポリシーを有効にする

Citrix Studioで、クライアントとの間でUSBデバイスリダイレクトを有効（または無効）にします（ワークステーションホストのみ）。

設定の編集ダイアログで、次のようにします。

1. **許可**を選択します。
2. **OK**をクリックします。

USBリダイレクトを有効にするイメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/client-usb-redirection-policy.png)

USBリダイレクト規則を設定する

USBリダイレクトポリシーを有効にした後、Citrix Studioを使用して、Linux VDAで許可（または拒否）するデバイスを指定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で、次のようにします。

1. リダイレクト規則を追加するには、**新規**をクリックするか、既存の規則を確認するには**編集**をクリックします。
2. 規則を作成（または編集）した後、**OK**をクリックします。

クライアントUSBリダイレクト規則設定のイメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/client-usb-redirection-rule.png)

VHCIカーネルモジュールを構築する

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`）に依存します。これらのモジュールはLinux VDAディストリビューションの一部（RPMパッケージの一部）です。これらは公式のLinuxディストリビューションカーネルに基づいてコンパイルされ、次の表に示されています。

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-ジェネリック
ウブントゥ 16.04	4.4.0-142-generic

重要:

マシンのカーネルがLinux VDA用にビルドされたドライバーと互換性がない場合、USBサービスが起動に失敗する可能性があります。この場合、独自のVHCIカーネルモジュールをビルドした場合にのみ、USBリダイレクト機能を使用できます。

お使いのカーネルがCitrixによってビルドされたモジュールと互換性があるかを確認する

コマンドラインで、次のコマンドを実行してカーネルに一貫性があるかどうかを確認します。

```
insmod /opt/Citrix/VDA/lib64/usb-vhci-hcd.ko
```

コマンドが正常に実行された場合、カーネルモジュールは正常にロードされており、バージョンはCitrixによってインストールされたものと一致しています。

コマンドがエラーで実行された場合、カーネルはCitrixモジュールと互換性がなく、再ビルドする必要があります。

VHCIカーネルモジュールを再ビルドする

カーネルモジュールがCitrixバージョンと一致しない場合は、次の操作を行います。

1. Citrix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。「**Linux Virtual Delivery Agent (sources)**」セクションに含まれるファイルを選択します。
2. **citrix-linux-vda-sources.zip**ファイルを解凍します。**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ソースファイルを解凍します。
3. ヘッダーファイルと**Module.symvers**ファイルに基づいてカーネルモジュールをビルドします。適切なLinuxディストリビューションに基づいてカーネルヘッダーファイルをインストールし、**Module.symvers**を作成するには、次の手順を使用します。

   RHEL/CentOS:

   ```
   yum install kernel-devel
   ```

   SUSE 12:

   ```
   zypper install kernel-devel



   zypper install kernel-source
   ```

   ウブントゥ:

   ```
   apt-get install linux-headers
   ```

   ヒント:

   インストールが成功すると、次のような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フォルダーで、**Module.symvers**ファイルが存在することを確認します。このファイルがフォルダーにない場合は、カーネルをビルド（次のコマンドを順に実行: `make oldconfig`; `make prepare`; `make modules`; `make`）してこの

ファイルを取得するか、**/usr/src/kernels/3.10.0-327.10.1.el7.x86_64-obj/x86_64/defaults/module.***からコピーします。

5. 開発ツールをインストールするには、次のコマンドを実行します。

   RHEL/CentOS:

   ```
   yum groupinstall 'Development Tools'
   ```

   ウブントゥ 18.04:

   ```
   apt install build-essential

   apt install libelf-dev
   ```

   ウブントゥ 16.04:

   ```
   apt install build-essential
   ```

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
   ```

7. 「**vhci-hcd-1.15/**」フォルダーで、VHCIカーネルをビルドするために`make`を実行します。

   > **注:**
   >
   > ビルドが成功すると、「**vhci-hcd-1.15/**」フォルダーに`usb-vhci-hcd.ko`と`usb-vhci-iocifc.ko`が作成されます。

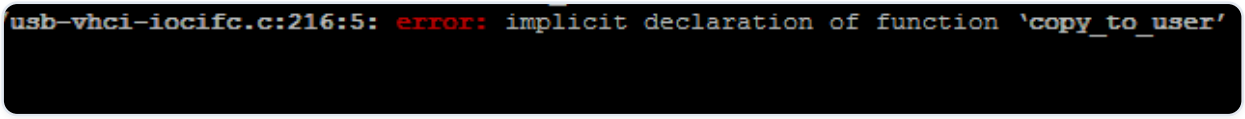
8. カーネルモジュールを新しくビルドしたものに置き換えます: **cp -f usb-vhci-*.ko /opt/Citrix/VDA/lib64/**
9. USBサービスを再起動します:

```
service ctxusbsd restart
```

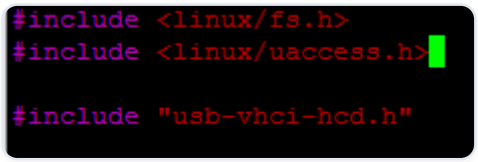
10. セッションからログオフし、再度ログオンします。USBリダイレクトが機能しているか確認します。

カーネルビルドの問題のトラブルシューティング

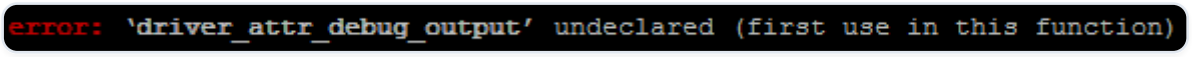
- **Ubuntu 16の特定のカーネルでカーネルビルドエラーが発生する場合があります。**エラーは `implicit declaration of function 'copy\_to\_user'` と表示されます。以下のスクリーンショットを参照してください。



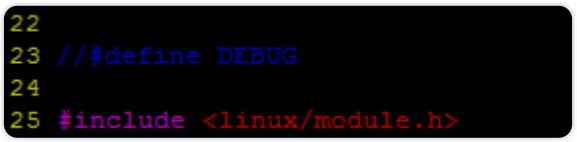
このエラーは、カーネルのヘッダーファイルの変更が原因で発生します。回避策として、**#include \<linux/uaccess.h\>** の行を `vhci-hcd-1.15/usb-vhci-iocifc.c` ファイルに追加します。



- **Ubuntu 16のカーネル4.15.0-29-genericでカーネルビルドエラーが発生する場合があります。**エラーは `'driver\_attr\_debug_output' undeclared` と表示されます。以下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、カーネルにシンボルが不足している場合に発生します。回避策として、`vhci-hcd-1.15/usb-vhci-iocifc.c` 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。

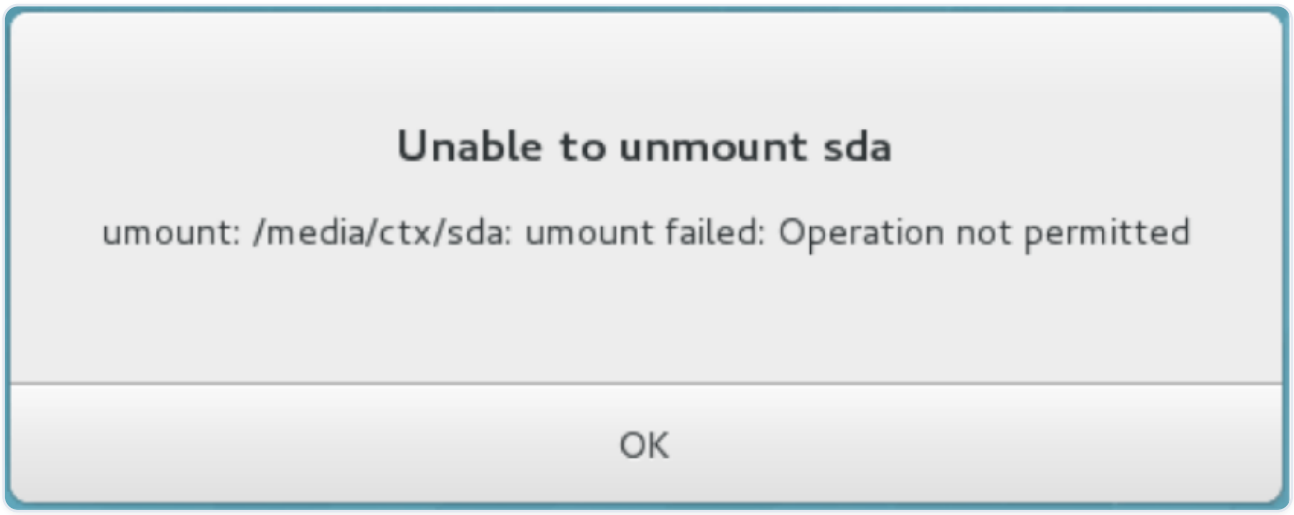


USBリダイレクトの問題のトラブルシューティング

このセクションの情報を使用して、Linux VDA の使用中に発生する可能性のあるさまざまな問題をトラブルシューティングしてください。

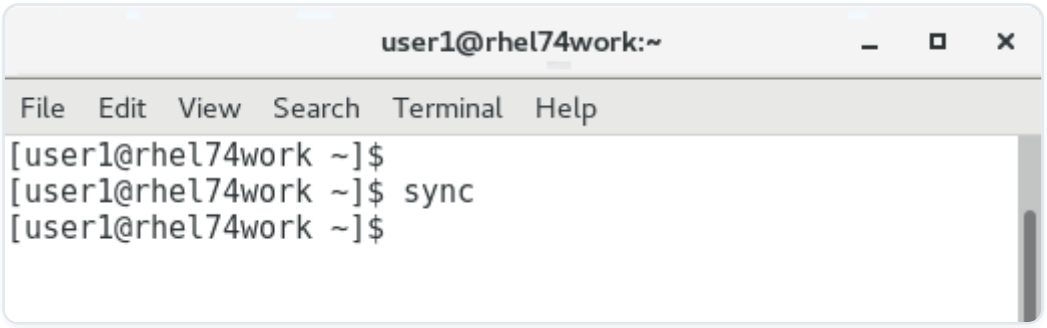
リダイレクトされた USB ディスクをアンマウントできない**

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制御のため、Linux VDA は、所有者のみがリダイレクトされたデバイスにアクセスできるように、これらすべてのデバイスを管理者権限で管理します。その結果、ユーザーは管理者権限なしではデバイスをアンマウントできません。



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

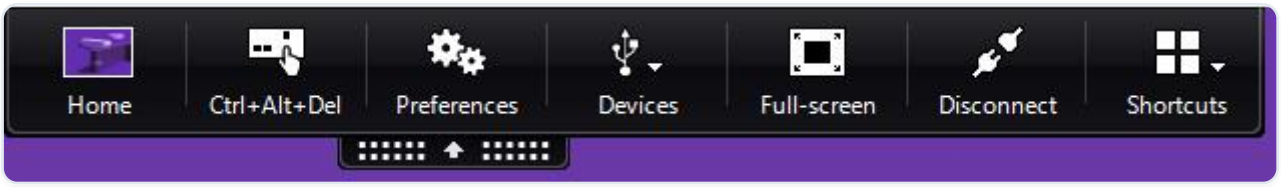
USB ディスクをセッションにリダイレクトし、それを変更しようとした場合（たとえば、ディスク上にいくつかのファイルを作成する）、その後すぐに Citrix Workspace アプリのツールバーを使用してリダイレクトを停止すると、変更または作成したファイルが失われる可能性があります。この問題は、ファイルシステムにデータを書き込む際に、システムがファイルシステム内のメモリキャッシュをマウントするため発生します。データはディスク自体には書き込まれません。Citrix Workspace アプリのツールバーを使用してリダイレクトを停止すると、データがディスクにフラッシュされる時間がなくなり、データが失われます。この問題を解決するには、USB リダイレクトを停止する前に、ターミナルで sync コマンドを使用してデータをディスクにフラッシュしてください。



Citrix Workspace アプリのツールバーにデバイスがない

Citrix Workspace アプリのツールバーにデバイスが表示されないことがあり、これは USB リダイレクトが行われていないことを示します。この問題が発生した場合は、以下を確認してください。

- ポリシーが USB リダイレクトを許可するように構成されている
- カーネルモジュールがご使用のカーネルと互換性がある



> **注:**
>
> **デバイス**タブは、Linux 用 Citrix Workspace アプリでは利用できません。

Citrix Workspace アプリのツールバーに USB デバイスが表示されるが、「*ポリシーによって制限されています*」と表示されてリダイレクトに失敗する

この問題が発生した場合は、次の操作を行います。

- Linux VDAポリシーを構成してリダイレクトを有効にします。
- Citrix Workspaceアプリのレジストリに追加のポリシー制限が構成されていないか確認します。レジストリパスの **DeviceRules** を確認し、この設定によってデバイスへのアクセスが拒否されていないことを確認します。

  HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Citrix\\ICA Client\\GenericUSB

USBデバイスは正常にリダイレクトされますが、セッションで使用できません

通常、サポートされているUSBデバイスのみがリダイレクトできます。他のデバイスもアクティブなLinux VDAセッションにリダイレクトされる場合があります。この場合、リダイレクトされたデバイスごとに、システム **/dev** パスにユーザーが所有するノードが作成されます。ただし、ユーザーがデバイスを正常に使用できるかどうかは、ドライバーと構成によって決まります。所有されている（接続されている）がアクセスできないデバイスが見つかった場合は、そのデバイスを無制限のポリシーに追加してください。

> **注:**
>
> USBドライブの場合、Linux VDAはディスクを構成してマウントします。ユーザー（およびそれをインストールした所有者のみ）は、追加の構成なしでディスクにアクセスできます。これは、サポートされているデバイスリストにないデバイスには当てはまらない場合があります。

セッションの信頼性を構成する

Citrix® は、サポートされているすべての Linux プラットフォームにセッションの信頼性機能を導入しました。セッションの信頼性はデフォルトで有効になっています。

セッションの信頼性により、ネットワークの中断時でも ICA セッションがシームレスに再接続されます。セッションの信頼性の詳細については、「クライアント自動再接続とセッションの信頼性」を参照してください。

注: セッションの信頼性接続を介して送信されるデータは、デフォルトでプレーンテキストです。セキュリティ上の理由から、TLS 暗号化を有効にすることをお勧めします。TLS 暗号化の詳細については、「TLS を使用したユーザーセッションの保護」を参照してください。

構成

Citrix Studio のポリシー設定

Citrix Studio でセッションの信頼性について、次のポリシーを設定できます。

- セッションの信頼性接続
- セッションの信頼性タイムアウト
- セッションの信頼性ポート番号
- 再接続 UI の透過レベル

詳細については、「セッションの信頼性ポリシー設定」および「クライアント自動再接続ポリシー設定」を参照してください。

注: **セッションの信頼性接続**または**セッションの信頼性ポート番号**ポリシーを設定した後、設定を有効にするには、VDA サービスと HDX™ サービスをこの順序で再起動してください。

Linux VDA の設定

- **セッションの信頼性 TCP リスナーの有効化/無効化**

デフォルトでは、セッションの信頼性 TCP リスナーは有効になっており、ポート 2598 でリッスンしています。リスナーを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "fEnableWinStation" -d "0x00000000"
```

注: 設定を有効にするには、HDX サービスを再起動してください。TCP リスナーを無効にしても、セッションの信頼性は無効になりません。**セッションの信頼性接続**ポリシーを通じて機能が有効になっている場合、セッションの信頼性は他のリスナー（SSL など）を通じて引き続き利用できます。

- **セッションの信頼性ポート番号**

次のコマンドを使用して、セッションの信頼性ポート番号を設定することもできます（例としてポート番号 2599 を使用）。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cg
p"   -v  "PortNumber" -d  "2599"
```

注: 設定を有効にするには、HDXサービスを再起動してください。ポート番号がCitrix Studioのポリシー設定で設定されている場合、Linux VDAでの設定は無視されます。VDA上のファイアウォールが、設定されたポートを介したネットワークトラフィックを禁止しないように構成されていることを確認してください。

- **サーバーからクライアントへのキープアライブ間隔**

セッションにアクティビティがない場合（たとえば、マウスの動きがない、画面の更新がない）、Linux VDAとICA®クライアントの間でセッション信頼性キープアライブメッセージが送信されます。キープアライブメッセージは、クライアントがまだ応答しているかどうかを検出するために使用されます。クライアントから応答がない場合、クライアントが再接続するまでセッションは中断されます。この設定は、連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpServerToClientKeepAlive"  -d  "10"  --force
```

- **クライアントからサーバーへのキープアライブ間隔**

この設定は、ICAクライアントからLinux VDAに送信される連続するキープアライブメッセージ間の秒数を指定します。デフォルトでは、この設定は構成されていません。これを構成するには、次のコマンドを実行します（例として10秒を使用）。

```
/opt/Citrix/VDA/bin/ctxreg create -k
"HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig"  -t "REG_DWORD"  -v
"CgpClientToServerKeepAlive"  -d  "10"  --force
```

トラブルシューティング

ポリシー設定でセッション信頼性を有効にした後、セッションを起動できない。

この問題を回避するには、次の手順を実行します。

1. Citrix Studioのポリシー設定でセッション信頼性を有効にした後、VDAサービスとHDXサービスがこの順序で再起動されていることを確認してください。
2. VDAで、次のコマンドを実行して、セッション信頼性TCPリスナーが実行されていることを確認します（例としてポート2598を使用）。

   ```
   netstat -an | grep 2598
   ```

   セッション信頼性ポートにTCPリスナーがない場合は、次のコマンドを実行してリスナーを有効にします。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations
   \cgp"   -v  "fEnableWinStation" -d "0x00000001"
   ```

ソフトキーボード

ソフトキーボード機能は、Linux仮想デスクトップまたはアプリケーションセッションで利用できます。ソフトキーボードは、入力フィールドに入ると自動的に表示され、入力フィールドを離れると非表示になります。

ソフトキーボードの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/soft-keyboard.png)

注:

この機能は、RHEL 7.7、CentOS 7.6、SUSE 12.3、Ubuntu 16.04、およびUbuntu 18.04で利用できます。Citrix Workspaceアプリ（iOSおよびAndroid用）でサポートされています。

機能の有効化と無効化

この機能はデフォルトで無効になっています。この機能を有効または無効にするには、**ctxreg** ユーティリティを使用します。特定のLinux VDAでの機能構成は、そのVDA上のすべてのセッションに適用されます。

この機能を有効にするには：

1. コマンドを実行します。

   ```
   /opt/Citrix/VDA/bin/ctxreg update -k
   "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
   "Enabled" -d "0x00000001"
   ```

2. Citrix Studioで、**自動キーボード表示**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、Intelligent Input Bus (IBus)をデフォルトのIMサービスとして構成するには、次のコマンドを実行します。

   ```
   echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
   ```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v
"Enabled" -d "0x00000000"
```

注:

上記の設定は、新しいセッションにログオンするか、現在のセッションからログオフして再度ログオンしたときに有効になります。

制限事項

- この機能は、Google Chrome、LibreOffice、およびその他のアプリでは期待どおりに動作しない場合があります。
- 手動で非表示にした後、ソフトキーボードを再度表示するには、入力フィールド以外の場所をクリックしてから、現在の入力フィールドを再度クリックします。

- Webブラウザーで1つの入力フィールドから別の入力フィールドにクリックしても、ソフトキーボードが表示されない場合があります。この問題を回避するには、入力フィールド以外の場所をクリックしてから、目的の入力フィールドをクリックします。
- この機能は、Unicode文字および2バイト文字（中国語、日本語、韓国語などの文字）をサポートしていません。
- ソフトキーボードはパスワード入力フィールドでは利用できません。
- ソフトキーボードが現在の入力フィールドと重なる場合があります。この場合、アプリのウィンドウを移動するか、画面をスクロールして、入力フィールドにアクセスしやすい位置に移動してください。
- Citrix Workspace™ アプリとHuaweiタブレット間の互換性の問題により、物理キーボードが接続されている場合でも、Huaweiタブレットではソフトキーボードが表示されます。

クライアント入力方式エディター (IME)

概要

中国語、日本語、韓国語などの2バイト文字は、IME を介して入力する必要があります。これらの文字は、Windows ネイティブの CJK IME など、クライアント側で Citrix Workspace™ アプリと互換性のある任意の IME を使用して入力します。

インストール

この機能は、Linux VDA をインストールすると自動的にインストールされます。

使用方法

通常どおり、Citrix Virtual Apps または Citrix Virtual Desktops™ セッションを開きます。

クライアント IME 機能の使用を開始するには、クライアント側で必要に応じて入力方式を変更します。

既知の問題

- Google スプレッドシートのセルに文字を入力するには、クライアント IME 機能を使用する前に、そのセルをダブルクリックする必要があります。
- クライアント IME 機能は、パスワードフィールドでは自動的に無効になりません。
- IME ユーザーインターフェイスは、入力領域でカーソルに追従しません。

複数の言語入力のサポート

Linux VDAバージョン1.4以降、Citrixは公開アプリケーションのサポートを追加しました。ユーザーはLinuxデスクトップ環境なしで目的のLinuxアプリケーションにアクセスできます。

しかし、Linux VDA上のネイティブ言語バーは、Linuxデスクトップ環境と高度に統合されているため、公開アプリケーションでは利用できませんでした。その結果、ユーザーは中国語、日本語、韓国語などのIMEを必要とする言語でテキストを入力できませんでした。さらに、アプリケーションセッション中にキーボードレイアウトを切り替えることもできませんでした。

これらの問題に対処するため、この機能はテキスト入力を受け入れる公開アプリケーションに言語バーを提供します。言語バーを使用すると、ユーザーはサーバー側のIMEを選択し、アプリケーションセッション中にキーボードレイアウトを切り替えることができます。

構成

この機能を有効または無効にするには、**ctxreg**ユーティリティを使用できます（デフォルトでは無効）。特定のLinux VDAサーバーでの機能構成は、そのVDAで公開されているすべてのアプリケーションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE \\SYSTEM\\CurrentControlSet\\Control\\Citrix\\LanguageBar」で、タイプはDWORDです。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE
\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d
"0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE
\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d
"0x00000000"
```

使用方法

使用方法は簡単です。

1. 機能を有効にします。

2. テキスト入力を受け入れる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、目的の言語（入力ソース）を追加します。

   地域と言語の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/select-region-language-dialog-box.png)
4. ドロップダウンメニューからIMEまたはキーボードレイアウトを選択します。
5. 選択したIMEまたはキーボードレイアウトを使用して言語を入力します。

> **注:**
>
> - VDA側の言語バーでキーボードレイアウトを変更する際は、クライアント側（Citrix Workspace™アプリを実行中）でも同じキーボードレイアウトが使用されていることを確認してください。
> - **Region & Language**ダイアログボックスで設定を行う前に、**accountsservice**パッケージをバージョン0.6.37以降にアップグレードする必要があります。

> IMEまたはキーボードレイアウトを選択する画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/region-language-dialog-box.png)

動的キーボードレイアウト同期

以前は、Linux VDAとクライアントデバイスのキーボードレイアウトは同じである必要がありました。例えば、クライアントデバイスでキーボードレイアウトが英語からフランス語に変更されたが、VDAでは変更されなかった場合、VDAもフランス語に変更されるまでキーマッピングの問題が発生し、継続する可能性がありました。

Citrix®は、VDAのキーボードレイアウトをクライアントデバイスのキーボードレイアウトと自動的に同期させることで、この問題を解決します。クライアントデバイスのキーボードレイアウトが変更されるたびに、VDAのレイアウトもそれに従います。

> **ヒント：**

> この機能は、Windows版Citrix Workspace™アプリでサポートされており、公開アプリケーション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無効になっています。この機能を有効または無効にするには、**ctxreg** ユーティリティを使用します。特定のLinux VDAでの機能構成は、そのVDA上のすべてのセッションに適用されます。

構成キーは「HKEY_LOCAL_MACHINE\\SYSTEM \\CurrentControlSet\\Control\\Citrix\\SyncKeyboardLayout」で、タイプはDWORDです。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "SyncKeyboardLayout" -d "0x00000001"
```

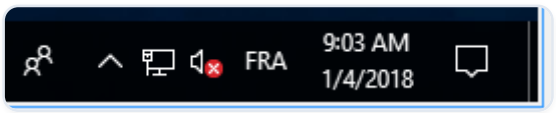
この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "SyncKeyboardLayout" -d "0x00000000"
```

使用方法

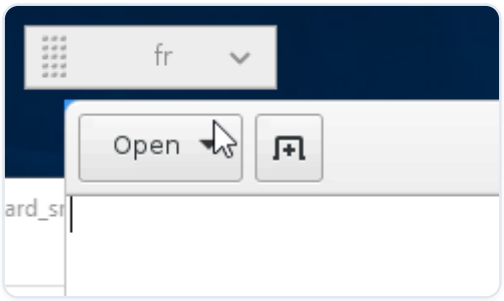
この機能を有効にすると、セッション中にクライアントデバイスのキーボードレイアウトが変更された場合、セッションのキーボードレイアウトもそれに応じて変更されます。

例えば、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。



すると、Linux VDAセッションのキーボードレイアウトも「fr」に変更されます。

アプリケーションセッションでは、言語バーを有効にしている場合、この自動変更を確認できます。



デスクトップセッションでは、タスクバーでこの自動変更を確認できます。



クライアントIMEユーザーインターフェイスの同期

概要

これまで、クライアントIMEユーザーインターフェイス（変換ウィンドウや候補ウィンドウを含む）は、画面の左上隅に配置されていました。カーソルに追従せず、テキスト入力領域のカーソルから離れた場所に表示されることがありました。

以前のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/composition-and-candidate-windows.png)

Citrix®は、クライアントIMEの使いやすさを向上させ、シームレスなエクスペリエンスをさらに改善しました。

現在のクライアントIMEユーザーインターフェイスの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/synced-composition-and-candidate-windows.png)

> **注:**

> この機能は、RHEL 7.x、CentOS 7.x、Ubuntu 16.04、Ubuntu 18.04、およびSUSE 12.xで利用できます。Citrix Workspaceアプリ for WindowsおよびMacで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、ユーザーインターフェイス言語をIME入力が必要な言語に設定するか、

> **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無効になっています。**ctxreg**ユーティリティを使用して、この機能を有効または無効にします。特定のLinux VDAでの機能構成は、そのVDA上のすべてのセッションに適用されます。

この機能を有効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "SyncClientIME" -d "0x00000001"
```

この機能を無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar"
-v "SyncClientIME" -d "0x00000000"
```

HDX™ インサイト

概要

HDX Insightは、Citrix Application Delivery Management (ADM) の一部であり、広く普及している業界標準のAppFlowに基づいています。これにより、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通過するCitrix ICAトラフィックに対する前例のないエンドツーエンドの可視性を提供することで、IT部門は優れたユーザーエクスペリエンスを提供できます。

Linux VDAは、HDX Insight機能を部分的にサポートしています。エンドユーザーエクスペリエンス管理 (EUEM) 機能が実装されていないため、時間に関連する一部のデータポイントは利用できません。

インストール

依存パッケージのインストールは不要です。

使用方法

HDX Insightは、Citrix Workspace™アプリとLinux VDAの間でCitrix ADCを介して渡されるICA®メッセージを分析します。

Linux VDAで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。既存の構成、設定、またはデータを失うことなく、NetScaler Insight Center展開を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

データポイントが表示されない

原因は2つ考えられます。

- HDX Insightが正しく構成されていません。

  たとえば、Citrix ADCでAppFlow®が有効になっていないか、Citrix ADMで誤ったCitrix ADCインスタンスが構成されています。

- Linux VDAでICA制御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、管理者に連絡してCitrixにバグを報告してください。

アプリケーションデータポイントが表示されない

シームレス仮想チャネルが有効になっており、シームレスアプリケーションがしばらく起動していることを確認してください。

アダプティブトランスポート

以前は実験的機能として利用可能でしたが、アダプティブトランスポートは本リリースで完全にサポートされる機能となりました。

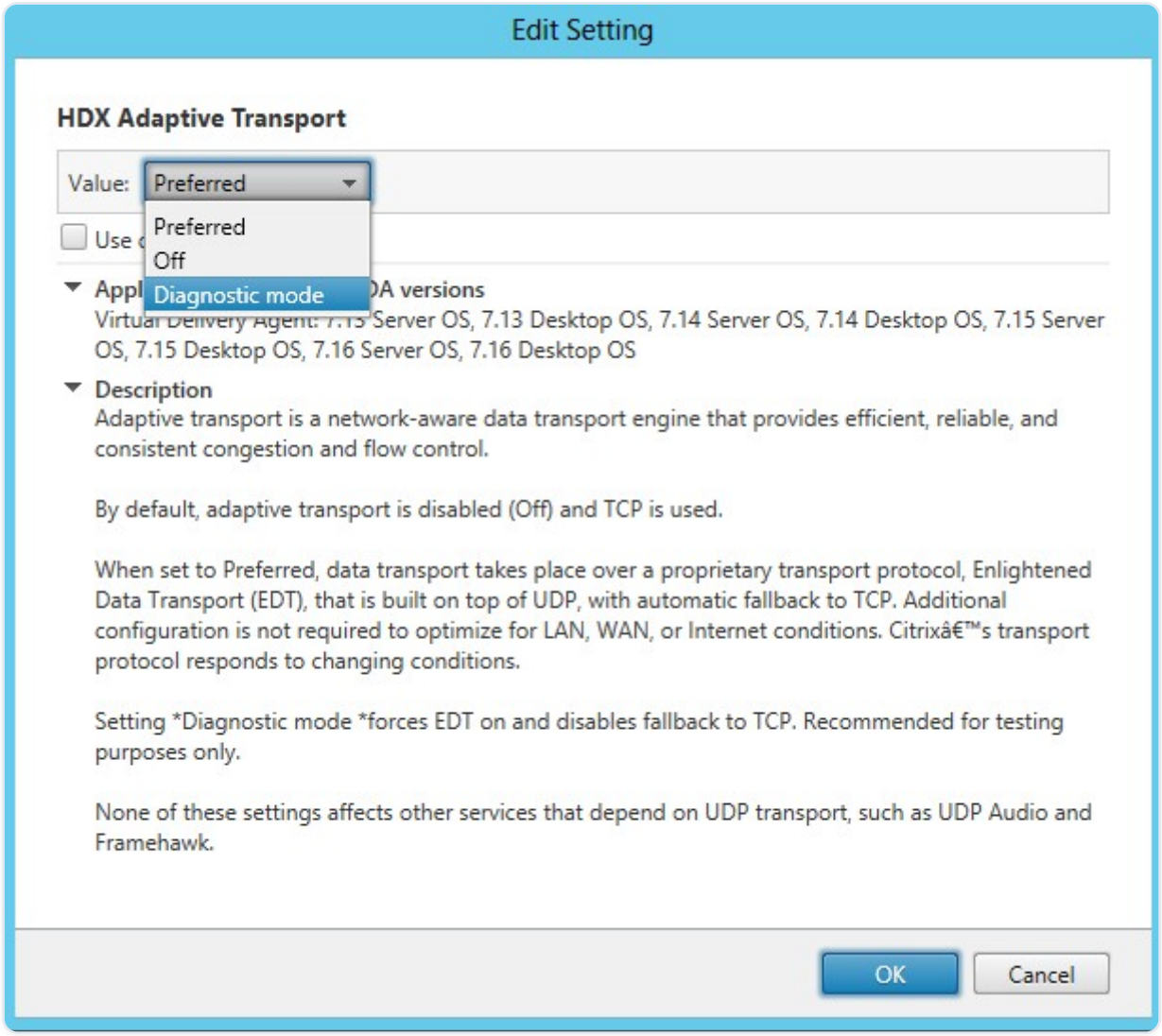
アダプティブトランスポートは、Citrix Virtual Apps and Desktops のデータ転送メカニズムです。より高速でスケーラブルになり、アプリケーションの対話性が向上し、困難な長距離WANおよび

インターネット接続での対話性も向上します。詳細については、「アダプティブトランスポート」を参照してください。

アダプティブトランスポートを有効にする

Citrix Studioで、**HDX™ Adaptive Transport**ポリシーが**Preferred**または**Diagnostic mode**に設定されていることを確認します。**Preferred**がデフォルトで選択されています。

- **Preferred**：可能な場合はEnlightened Data Transport（EDT）経由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。
- **Diagnostic mode**：EDTが強制的にオンになり、TCPへのフォールバックは無効になります。



アダプティブトランスポートを無効にする

アダプティブトランスポートを無効にするには、Citrix Studioで**HDX Adaptive Transport**ポリシーを**Off**に設定します。

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になっているか確認する

UDPリスナーが実行されているかを確認するには、次のコマンドを実行します。

```
netstat -an | grep "1494\|2598"
```

通常、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                 :::*
```

トレースの有効化

概要

ログの収集と問題の再現は、診断を遅らせ、ユーザーエクスペリエンスを低下させます。トレースの有効化機能は、このような作業を軽減します。Linux VDAでは、トレースはデフォルトで有効になっています。

構成

`ctxlogd`デーモンと`setlog`ユーティリティは、Linux VDAリリースパッケージに含まれるようになりました。デフォルトでは、Linux VDAをインストールして構成すると、`ctxlogd`デーモンが起動します。

ctxlogdデーモン

トレースされる他のすべてのサービスは、`ctxlogd`デーモンに依存します。Linux VDAのトレースを継続したくない場合は、`ctxlogd`デーモンを停止できます。

setlogユーティリティ

トレースの有効化は、**/opt/Citrix/VDA/bin/**パスにある`setlog`ユーティリティを使用して構成されます。rootユーザーのみが実行する権限を持っています。GUIを使用するか、コマンドを実行して構成を表示および変更できます。`setlog`ユーティリティに関するヘルプを表示するには、次のコマンドを実行します。

```
setlog help
```

値

デフォルトでは、**ログ出力パス**は**/var/log/xdl/hdx.log**に設定され、**最大ログサイズ**は200 MBに設定されており、**ログ出力パス**の下に最大2つの古いログファイルを保存できます。

現在の`setlog`値を表示します。

```
setlog values



log_path (Log Output Path) = /var/log/xdl/hdx.log



log_size (Max Log Size (MiB)) = 200



log_count (Max Old Log Files) = 2
```

単一の`setlog`値を表示または設定します。

```
setlog value <name> [<value>]
```

例:

```
setlog value log_size 100
```

レベル

デフォルトでは、ログレベルは**warning**（大文字と小文字を区別しない）に設定されています。

さまざまなコンポーネントに設定されているログレベルを表示するには、次のコマンドを実行します。

```
setlog levels
```

ログレベル（Disabled、Inherited、Verbose、Information、Warnings、Errors、Fatal Errorsを含む）を設定するには、次のコマンドを実行します。

```
setlog level <class> [<level>]
```

ログレベル	コマンドパラメータ (大文字と小文字を区別しない)
無効	なし
継承済み	継承
詳細	詳細
情報	情報
警告	警告
エラー	エラー
致命的なエラー	致命的

「`<class>`」変数は、Linux VDA の 1 つのコンポーネントを指定します。すべてのコンポーネントを対象とするには、「all」に設定します。例:

```
setlog level all error
```

フラグ

デフォルトでは、フラグは次のように設定されています。

```
setlog flags


DATE = true


TIME = true
```

```
NAME = true

PID = true

TID = false

SID = true

UID = false

GID = false

CLASS = false

LEVEL = false

FUNC = true

FILE = false
```

現在のフラグを表示:

```
setlog flags
```

単一のログフラグを表示または設定:

```
setlog flag <flag> [<state>]
```

既定値に戻す

すべてのレベル、フラグ、および値を既定の設定に戻します。

```
setlog default
```

重要:

`ctxlogd`サービスは、ルートユーザーのみが作成できる**/var/xdl/.ctxlog**ファイルを使用して構成されます。他のユーザーはこのファイルへの書き込み権限を持っていません。Citrixは、ルートユーザーが他のユーザーに書き込み権限を与えないことを推奨します。これに従わない場合、任意のまたは悪意のある構成が`ctxlogd`を引き起こし、サーバーのパフォーマンス、ひいてはユーザーエクスペリエンスに影響を与える可能性があります。

トラブルシューティング

/var/xdl/.ctxlogファイルが見つからない場合（例えば、誤って削除された場合）、`ctxlogd`デーモンが失敗し、`ctxlogd`サービスを再起動できません。

/var/log/messages:

```
Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
configuration file.



Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited,
code=exited, status=1/FAILURE



Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.
```

```
Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
```

この問題を解決するには、ルートユーザーとして`setlog`を実行して、**/var/xdl/.ctxlog**ファイルを再作成します。その後、他のサービスが依存している`ctxlogd`サービスを再起動します。

セッションのシャドウイング

セッションシャドウ機能を使用すると、ドメイン管理者はイントラネット内でユーザーのICAセッションを表示できます。この機能はnoVNCを使用してICAセッションに接続し、RHEL 7.xおよびUbuntu 16.04でのみサポートされています。

> **注:**
>
> セッションシャドウ機能を使用するには、Citrix Directorのバージョンが7.16以降である必要があります。

インストールと構成

依存関係

セッションシャドウイングには、`python-websockify`と`x11vnc`の2つの新しい依存関係が必要です。Ubuntu 16.04にLinux VDAをインストールすると、`python-websockify`と`x11vnc`の依存関係が自動的にインストールされます。RHEL 7.xでは、Linux VDAのインストール後に`python-websockify`と`x11vnc`を手動でインストールする必要があります。

RHEL 7.xで次のコマンドを実行して、`python-websockify`と`x11vnc` (`x11vnc`バージョン0.9.13以降) をインストールします。

```
sudo yum install -y python-websockify x11vnc
```

`python-websockify`と`x11vnc`を解決するには、RHEL 7.xで次のリポジトリを有効にします。

- Enterprise Linux 用追加パッケージ (EPEL)

  `python-websockify`と`x11vnc`の両方にEPELリポジトリが必要です。EPELリポジトリを有効にするには、次のコマンドを実行します。

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(rpm -E '%{rhel}').noarch.rpm
```

- オプションのRPM

  x11vncの一部の依存パッケージをインストールするために、次のいずれかのコマンドを実行してオプションのRPMリポジトリを有効にします。

  ワークステーションの場合:

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

  サーバーの場合:

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

ポート

セッションシャドウ機能は、Linux VDAからCitrix Directorへの接続を確立するために、6001～6099の範囲から利用可能なポートを自動的に選択します。そのため、同時にシャドウできるICA®セッションの数は99に制限されます。要件を満たすのに十分なポートが利用可能であることを確認してください。特にマルチセッションシャドウイングの場合は重要です。

レジストリ

次の表に関連するレジストリを示します:

レジストリ	説明	デフォルト値
EnableSessionShadowing	セッションシャドウイング機能を有効または無効にします	1 (有効)
ShadowingUseSSL	Linux VDAとCitrix Director間の接続を暗号化するかどうかを決定します	0 (無効)

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イングを無効にするには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent" -v "EnableSessionShadowing" -d
0x00000000
```

SSL

Linux VDAとCitrix Director間のnoVNC接続はWebSocketプロトコルを使用します。セッションシャドウイングの場合、`ws://`または`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定されます。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、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推奨します。Citrixは、`ws://`を使用するLinux VDAセッションシャドウイングに対するいかなるセキュリティ上の責任も負いません。

サーバーおよびルートSSL証明書の取得

証明書は信頼された認証局（CA）によって署名されている必要があります。

SSLを構成する各Linux VDAサーバーには、個別のサーバー証明書（キーを含む）が必要です。サーバー証明書は特定のコンピューターを識別するため、各サーバーの完全修飾ドメイン名（FQDN）を知っている必要があります。便宜上、ドメイン全体にワイルドカード証明書を使用することもできます。この場合、少なくともドメイン名を知っている必要があります。

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サーバーと通信する各Citrix Directorクライアントに、同じCAからの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、サーバー証明書を発行するのと同じCAから入手できます。サーバー証明書とクライアント証明書は、オペレーティングシステムにバンドルされているCA、エンタープライズCA（組織がアクセスを許可しているCA）、またはオペレーティングシステムにバンドルされていないCAからインストールできます。証明書を取得するために組織がどの方法を要求しているかについては、組織のセキュリティチームに相談してください。

重要:

- サーバー証明書の共通名は、Linux VDAサーバーの正確なFQDNであるか、少なくとも正しいワイルドカードとドメイン文字である必要があります。例: vda1.basedomain.com または *.basedomain.com。
- ハッシュアルゴリズム（SHA1やMD5を含む）は、一部のブラウザがサポートするデジタル証明書の署名には弱すぎます。そのため、SHA-256が最小標準として指定されています。

各Citrix Directorクライアントにルート証明書をインストールする

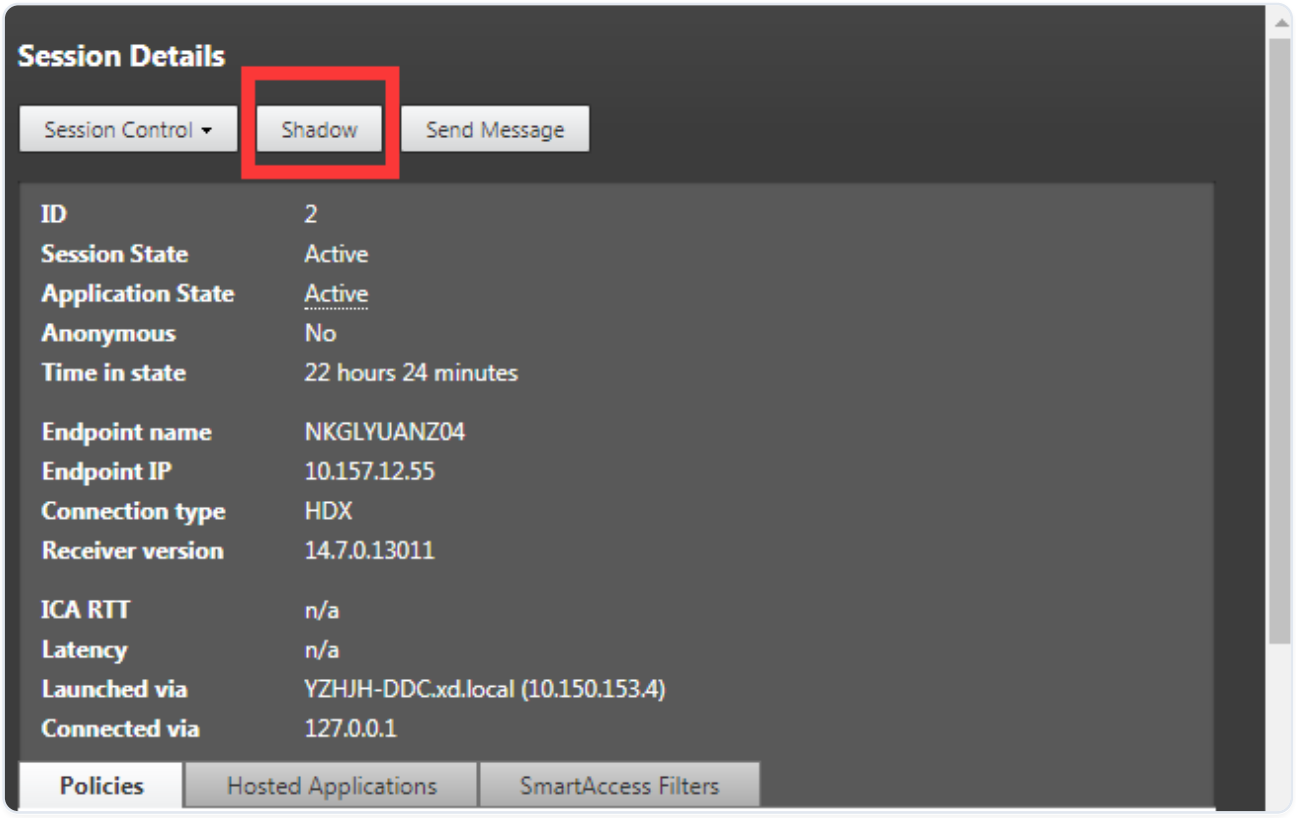
セッションシャドウイングはIISと同じレジストリベースの証明書ストアを使用するため、IISまたはMicrosoft Management Console (MMC) の証明書スナップインを使用してルート証明書をインストールできます。CAから証明書を受け取ったら、IISでWebサーバー証明書ウィザードを再起動すると、ウィザードが証明書をインストールします。または、MMCを使用してコンピューター上の証明書を表示およびインポートし、スタンドアロンのスナップインとして証明書を追加することもできます。Internet ExplorerとGoogle Chromeは、オペレーティングシステムにインストールされている証明書をデフォルトでインポートします。Mozilla Firefoxの場合、証明書マネージャーの**認証局**タブでルートSSL証明書をインポートする必要があります。

各Linux VDAサーバーにサーバー証明書とそのキーをインストールする

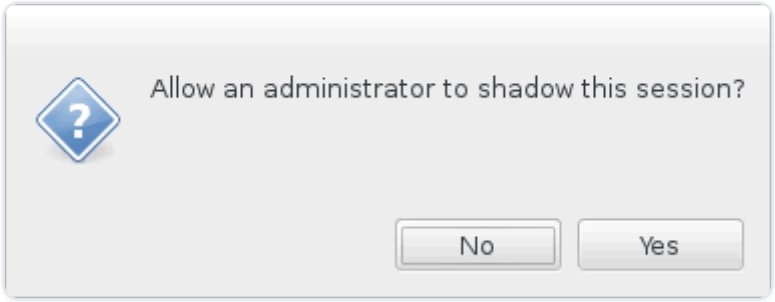
サーバー証明書には「shadowingcert.*」、キーファイルには「shadowingkey.*」という名前を付けます（*は形式を示すことができます。例：shadowingcert.csr、shadowingkey.key）。サーバー証明書とキーファイルをパス **/etc/xdl/shadowingssl** の下に置き、制限された権限で適切に保護してください。名前またはパスが正しくないと、Linux VDAは特定の証明書またはキーファイルを見つけることができず、Citrix Directorとの接続障害が発生します。

使用方法

Citrix Directorから、ターゲットセッションを見つけ、**セッションの詳細**ビューで**シャドウ**をクリックして、Linux VDAにシャドウイング要求を送信します。



接続が初期化されると、ICAセッションクライアント（Citrix Directorクライアントではない）に確認が表示され、ユーザーにセッションをシャドウする許可を求めます。



ユーザーが**はい**をクリックすると、Citrix Director側にウィンドウが表示され、ICAセッションがシャドウされていることを示します。

使用方法の詳細については、Citrix Directorドキュメントを参照してください。

制限事項

- セッションシャドウイングは、イントラネット内でのみ使用するように設計されています。Citrix Gateway経由で接続する場合でも、外部ネットワークでは機能しません。Citrixは、外部ネットワークでのLinux VDAセッションシャドウイングに関するいかなる責任も負いません。
- セッションシャドウイングが有効になっている場合、ドメイン管理者はICAセッションを表示することしかできず、書き込みや制御の権限はありません。
- 管理者がCitrix Directorから**シャドウ**をクリックすると、ユーザーにセッションをシャドウする許可を求める確認が表示されます。セッションは、セッションユーザーが許可を与えた場合にのみシャドウできます。
- 前述の確認には20秒のタイムアウト制限があります。時間が経過すると、シャドウイング要求は失敗します。
- 1つのICAセッションは、1つのCitrix Directorウィンドウで1人の管理者のみ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おり、その間に管理者Bがシャドウ要求を送信した場合、ユーザー権限取得の確認がユーザーデバイスに再表示されます。ユーザーが同意すると、管理者Aのシャドウ接続は停止し、管理者Bのために新しいシャドウ接続が確立されます。同じ管理者が同じICAセッションに対して別のシャドウ要求を送信した場合も同様です。
- セッションシャドウを使用するには、Citrix Director 7.16以降をインストールしてください。
- Citrix Directorクライアントは、ターゲットのLinux VDAサーバーに接続するために、IPアドレスではなくFQDNを使用します。したがって、Citrix DirectorクライアントはLinux VDAサーバーのFQDNを解決できる必要があります。

トラブルシューティング

セッションシャドウが失敗した場合は、Citrix Directorクライアントと Linux VDAの両方でデバッグを実行してください。

Citrix Directorクライアントで

ブラウザの開発者ツールを使用して、**コンソール**タブで出力ログを確認します。または、**ネットワーク**タブでShadowLinuxSession APIの応答を確認します。ユーザー権限取得の確認が表示されるにもかかわらず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://`接続に問題がある場合は、証明書を確認してください。

Linux VDA において

シャドウ要求に応じてユーザー権限取得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およびhdx.logファイルで手がかりを探します。vda.logファイルを取得するには、次の手順を実行します。

1. /etc/xdl/ctx-vda.confファイルを見つけます。vda.log設定を有効にするには、次の行のコメントを解除します。

   `Log4jConfig=”/etc/xdl/log4j.xml”`

2. /etc/xdl/log4j.xmlを開き、com.citrix.dmcの部分を見つけて、「info」を「trace」に次のように変更します。

   ```
   <!-- Broker Agent Plugin - Director VDA plugin Logger -->


    <logger name="com.citrix.dmc">


      <level value="trace"/>


    </logger>
   ```

3. `ctxvda`サービスを再起動するには、`service ctxvda restart`コマンドを実行します。

接続確立中にエラーが発生した場合：

1. セッションシャドウがポートを開くのを妨げるファイアウォールの制限がないか確認します。
2. SSLシナリオの場合、証明書とキーファイルが適切に命名され、正しいパスに配置されていることを確認します。
3. 新しいシャドウ要求のために、6001～6099の間に十分なポートが残っていることを確認します。

Citrix Workspace™ app for HTML5 のサポート

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、Citrix Workspace app for HTML5 を使用して Linux 仮想アプリおよびデスクトップに直接アクセスできます。Citrix Workspace app for HTML5 の詳細については、Citrix ドキュメントを参照してください。

この機能を有効にする

この機能はデフォルトで無効になっています。これを有効にするには、次の手順を実行します。

1. Citrix StoreFront™ で、Citrix Workspace app for HTML5 を有効にします。

   詳細な手順については、Knowledge Center の記事 CTX208163 の手順 1 を参照してください。

2. WebSocket 接続を有効にします。

   a. Citrix Studio で、**WebSockets connections** ポリシーを **Allowed** に設定します。

      他の WebSocket ポリシーも設定できます。WebSocket ポリシーの完全なリストについては、WebSockets policy settingsを参照してください。

   b. VDA で、設定を有効にするには、`ctxvda` サービスと `ctxhdx` サービスをこの順序で再起動します。

   c. VDA で、次のコマンドを実行して、WebSocket リスナーが実行されているかどうかを確認します。

      ```
      netstat -an | grep 8008
      ```

      WebSocket リスナーが実行されている場合、コマンド出力は次のようになります。

      ```
      tcp 0 0 :::8008 :::* LISTEN
      ```

   注: TLS 暗号化を有効にして、WebSocket 接続を保護することもできます。TLS 暗号化の有効化の詳細については、Secure user sessions using TLSを参照してください。

Citrix DirectorでLinuxセッションを監視する

Citrix Directorでは、Linuxセッションに対して以下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、**セッションの詳細**パネルを確認してください。

- ICA® RTT

  Linux VDAバージョン1903以降、ICA RTTメトリックが利用可能です。ICA RTT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ラウンドトリップ計算**および**ICAラウンドトリップ計算間隔**ポリシーを作成します。ポリシー作成の詳細については、「Studioを使用したポリシーの作成」を参照してください。

- プロトコル

  Linux VDAバージョン1909以降、プロトコル情報が利用可能です。Linuxセッションのトランスポートプロトコルは、**セッションの詳細**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

監視サービスデーモンは、定期的なスキャンを実行して主要なサービスを監視します。例外を検出すると、デーモンはサービスプロセスを再起動または停止し、リソースを解放するためにプロセスの残りをクリーンアップします。検出された例外は、**/var/log/xdl/ms.log** ファイルに記録されます。

構成

VDAを起動すると、監視サービスデーモンは自動的に起動します。

管理者権限で、**scanningpolicy.conf**、**rulesets.conf**、および **whitelist.conf** ファイルを通じてこの機能を構成できます。構成ファイルは **/opt/Citrix/VDA/sbin** にあります。

scanningpolicy.conf、**rulesets.conf**、および **whitelist.conf** ファイルでの変更を有効にするには、次のコマンドを実行して監視サービスデーモンを再起動します。

```
service ctxmonitorservice restart
```

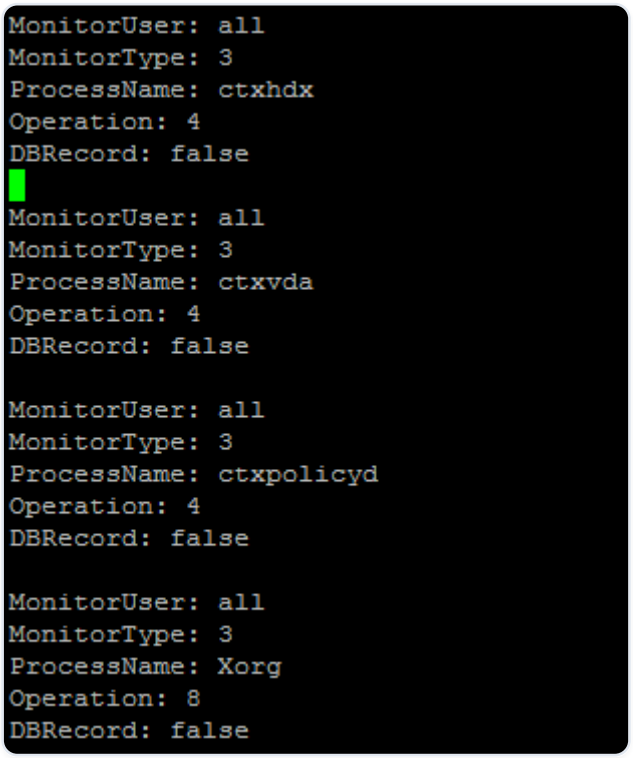
- **scanningpolicy.conf**

  この構成ファイルは、監視サービスデーモンを有効または無効にします。サービス検出間隔を設定し、検出された例外を修復するかどうかを指定します。

- MonitorEnable: true/false (デフォルトはtrue)
- DetectTime: 20 (単位：秒、デフォルト値：20、最小値：5)
- AutoRepair: true/false (デフォルトはtrue)
- MultBalance: 無効
- ReportAlarm: 無効

- **rulesets.conf**

  この構成ファイルは、監視対象サービスを指定します。次のスクリーンショットに示すように、デフォルトで4つのサービスが監視されます。



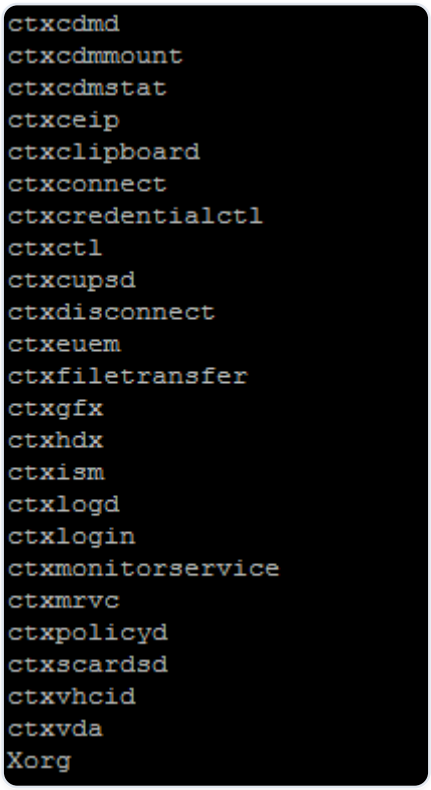
  監視する各サービスを設定するには、以下のフィールドを設定します。

  - MonitorUser: すべて
  - MonitorType: 3
  - ProcessName: <> (プロセス名は空白にできず、完全に一致する必要があります。)
  - Operation: 1/2/4/8 (1 = 例外が検出されたときにサービスを停止する。2 = 例外が検出されたときにサービスを強制終了する。4 = サービスを再起動する。8 = Xorgプロセスの残存物をクリーンアップする。)
  - DBRecord: 無効

- **whitelist.conf**

rulesets.confファイルで指定されたターゲットサービスは、**whitelist.conf**ファイルでも構成する必要があります。ホワイトリスト構成は、セキュリティのための二次フィルターです。

ホワイトリストを構成するには、**whitelist.conf**ファイルにプロセス名のみ（完全に一致する必要があります）を含めます。例については、以下のスクリーンショットを参照してください。



注:

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

TLS を使用してユーザーセッションを保護する

バージョン 7.16 以降、Linux VDA はセキュアなユーザーセッション向けに TLS 暗号化をサポートしています。TLS 暗号化はデフォルトで無効になっています。

TLS 暗号化を有効にする

セキュアなユーザーセッション向けにTLS暗号化を有効にするには、証明書を取得し、Linux VDAとDelivery Controller™（コントローラー）の両方でTLS暗号化を有効にします。

証明書を取得する

信頼された認証局 (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、次のセクションが含まれています。

- 証明書
- 暗号化されていない秘密キー
- 中間証明書 (オプション)

サーバー証明書の例：

サーバー証明書の例(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/example-server-certificate.png)

TLS 暗号化を有効にする

Linux VDA で TLS 暗号化を有効にする

Linux VDA で、**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。このツールで利用可能なオプションについては、**/opt/Citrix/VDA/sbin/enable_vdassl.sh -help** コマンドを実行してください。

ヒント: サーバー証明書は各 Linux VDA サーバーにインストールする必要があり、ルート証明書は各 Linux VDA サーバーとクライアントにインストールする必要があります。

コントローラーで TLS 暗号化を有効にする

> **注:**

> TLS暗号化は、デリバリーグループ全体に対してのみ有効にできます。特定のアプリケーションに対してTLS暗号化を有効にすることはできません。

> Controller上のPowerShellウィンドウで、ターゲットのデリバリーグループに対してTLS暗号化を有効にするには、以下のコマンドを順に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

> **注:**

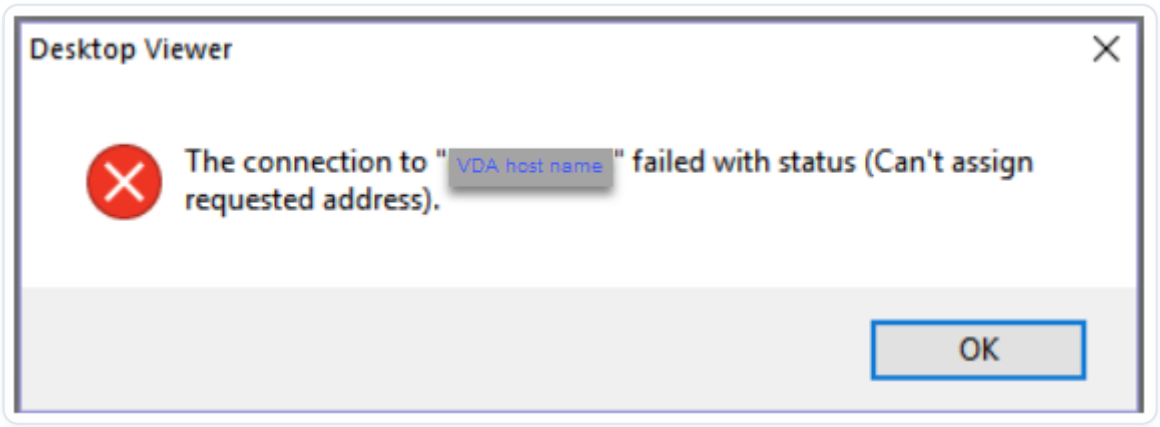
> ICA®セッションファイルにVDA FQDNのみが含まれるようにするには、**Set-BrokerSite –DnsResolutionEnabled $true**コマンドも実行できます。このコマンドはDNS解決を有効にします。DNS解決を無効にすると、ICAセッションファイルはVDA IPアドレスを開示し、SSLProxyHostやUDPDTLSPortなどのTLS関連項目に対してのみFQDNを提供します。

> ControllerでTLS暗号化を無効にするには、以下のコマンドを順に実行します。

1. `Add-PSSnapin citrix.*`.
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

トラブルシューティング

公開デスクトップセッションにアクセスしようとすると、Windows版Citrix Workspace™アプリで以下の「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可能性があります。



回避策として、**hosts**ファイルに次のようなエントリを追加します。

```
<IP address of the Linux VDA>      <FQDN of the Linux VDA>
```

Windowsマシンでは、**hosts**ファイルは通常`C:\Windows\System32\drivers\etc\hosts`にあります。

DTLS を使用したユーザーセッションの保護

DTLS 暗号化は、7.18 リリース以降で完全にサポートされている機能です。デフォルトでは、この機能は Linux VDA で有効になっています。詳細については、「Transport Layer Security」を参照してください。

DTLS 暗号化を有効にする

アダプティブトランスポートが有効になっていることを確認する

Citrix Studio で、**HDX™ Adaptive Transport** ポリシーが **Preferred** または **Diagnostic mode** に設定されていることを確認します。

Linux VDA で SSL 暗号化を有効にする

Linux VDA で、**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。このツールで利用可能なオプションについては、**/opt/Citrix/VDA/sbin/enable_vdassl.sh –h** コマンドを実行してください。

> **注:**
>
> 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 Citrix Workspace app 1808 for Windows 以降が必要です。クライアントが DTLS 1.0 のみをサポートしている場合（例：Citrix Receiver for Windows 4.11）、**enable_vdassl.sh** ツールを使用して、**SSLMinVersion** を **TLS_1.0** に、**SSLCipherSuite** を **COM** または **ALL** に設定します。

スマートカードのサポート

クライアントデバイスに接続されたスマートカードを使用して、Linux仮想デスクトップセッションへのログオン時に認証を行うことができます。この機能は、ICA®スマートカード仮想チャネルを介したスマートカードリダイレクトによって実装されます。セッション内でスマートカードを使用することもできます。使用例としては、ドキュメントへのデジタル署名の追加、メールの暗号化または復号化、スマートカード認証を必要とするWebサイトへの認証などがあります。

Linux VDAは、この機能に関してWindows VDAと同じ構成を使用します。詳細については、この記事の「スマートカード環境の構成」セクションを参照してください。

スマートカードを使用したパススルー認証の可用性は、以下の条件に依存します。

- Linux VDAがRHEL 7.7にインストールされていること。
- CoolKeyでサポートされているスマートカードが使用されていること。
- Citrix Workspace™ app for Windowsが使用されていること。

注:

Linux VDAセッション内でマッピングされたスマートカードを使用してCitrix Gatewayにサインオンすることは、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易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、公開された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を確認する

CoolKeyは、RHEL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の4種類のスマートカードをサポートしています。ただし、正式にサポートおよび検証されているカードの数はまだ限られています（Red Hat Enterprise Linuxでのスマートカードのサポートを参照）。

この記事では、構成を説明するための例としてYubiKey 4スマートカードを使用します。YubiKey 4は、Amazonやその他の小売業者から簡単に購入できるオールインワンのUSB CCID PIVデバイスです。CoolKeyドライバーはYubiKey 4をサポートしています。



組織でさらに高度なスマートカードが必要な場合は、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のインストール」(/ja-jp/linux-virtual-delivery-agent/1912-ltsr/configuration/support-for-smart-cards.html#install-the-smart-card-driver)を参照してください。スマートカードを挿入し、CoolKeyがスマートカードをサポートしていることを確認するために、次のコマンドを実行します。

```
pkcs11-tool --module libcoolkeypk11.so --list-slots
```

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、スロット情報が含まれる次の例のようになります。

```
[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 ~]#
```

s

構成

ルート証明書を準備する

ルート証明書は、スマートカード上の証明書を検証するために使用されます。ルート証明書をダウンロードしてインストールするには、次の手順を実行します。

1. 通常、CAサーバーからPEM形式のルート証明書を取得します。

   DERファイル（*.crt、*.cer、*.der）をPEMに変換するには、次のコマンドに似たコマンドを実行できます。次のコマンド例では、**certnew.cer**はDERファイルです。

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   ```

2. ルート証明書を`openssl`ディレクトリにインストールします。**certnew.pem**ファイルは例として使用されます。

   ```
   cp certnew.pem <path where you install the root certificate>
   ```

ルート証明書をインストールするためのパスを作成するには、`sudo mdkir -p <path where you install the root certificate>`を実行します。

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ためにNSSデータベースに依存しています。NSSデータベースを構成するには、次の手順を実行します。

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
   ```

2. 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するために、次のコマンドを実行します。

   ```
   certutil -L -d /etc/pki/nssdb
   ```

   ルート証明書が正常に追加された場合、コマンド出力は次の例のようになります。

   ```
   [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. NSS PKCS#11ライブラリにCoolKeyがインストールされているか確認します。

   ```
   modutil -list -dbdir /etc/pki/nssdb
   ```

   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。

   Coolkeyモジュールがインストールされている場合のコマンド出力(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/coolkeyinstalled.png)

   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モジュールを手動でインストールし、再度インストールを確認します。

   ```
   modutil -add "CoolKey PKCS #11 Module" -libfile libcoolkeypk11.so
   -dbdir /etc/pki/nssdb
   ```

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ためにローカル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。このパス内の各ルート証明書にはハッシュリンクがあります。準備されたルート証明書をインストールし、pam_pkcs11を構成するには、次のコマンドを実行します。

```
yum install pam_pkcs11



mkdir /etc/pam_pkcs11/cacerts/



cp certnew.pem /etc/pam_pkcs11/cacerts/



cacertdir_rehash /etc/pam_pkcs11/cacerts
```

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情報をデフォルトレルムに追加します。この設定は、**/etc/krb5.conf**構成ファイルで変更できます。
>
> スマートカードを初めて使用する前に、ctxsmartlogon.shスクリプトを実行してスマートカード環境を構成します。

> **ヒント:**
>
> ドメイン参加にSSSDを使用している場合は、ctxsmartlogon.shの実行後にSSSDサービスを再起動してください。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

ctxsmartlogon.shスクリプトを実行し、「はい」を選択する(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/smart-card-logon-script1.png)

ctxsmartlogon.shスクリプトを実行してスマートカードを無効にすることもできます。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります。

```
#*******************************************************************************
# ctxsmartlogon.sh 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いた構成については、Knowledge Centerの記事CTX206156を参照してください。

  次のステップに進む前に、すべてのコンポーネントが正しく構成され、秘密鍵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おり、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におけるPersonal Computer/Smart Card (PC/SC) 仕様の実装です。スマートカードおよびリーダーと通信するためのWindowsスマートカードインターフェイスを提供します。Linux VDAでのスマートカードリダイレクトは、PC/SCレベルで実装されています。

PC/SC Liteパッケージをインストールするには、次のコマンドを実行します。

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

スマートカードドライバーをインストールする

CoolKeyは、RHEL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。

```
yum install coolkey
```

スマートカード認証用のPAMモジュールをインストールする

pam_krb5およびkrb5-pkinitモジュールをインストールするには、次のコマンドを実行します。

```
yum install pam_krb5 krb5-pkinit
```

pam_krb5モジュールは、PAM対応アプリケーションがパスワードをチェックし、Key Distribution Center (KDC) からチケット保証チケットを取得するために使用できるプラガブル認証モジュールです。krb5-pkinitモジュールには、クライアントが秘密鍵と証明書を使用してKDCから初期資格情報を取得できるようにするPKINITプラグインが含まれています。

pam_krb5モジュールを構成する

pam_krb5モジュールは、スマートカード内の証明書を使用してKerberosチケットを取得するためにKDCと対話します。PAMでpam_krb5認証を有効にするには、次のコマンドを実行します。

```
authconfig --enablekrb5 --update
```

/etc/krb5.conf 設定ファイルに、実際のレルムに従ってPKINIT情報を追加します。

> **注:**
>
> **pkinit_cert_match** オプションは、PKINIT認証を試行する前にクライアント証明書が一致しなければならない照合ルールを指定します。照合ルールの構文は次のとおりです。
>
> `[relation-operator] component-rule …`

ここで `relation-operator` は、すべてのコンポーネントルールが一致する必要があることを意味する `&&`、または1つのコンポーネントルールのみが一致する必要があることを意味する \|\| のいずれかになります。

一般的なkrb5.confファイルの例を次に示します。

```
EXAMPLE.COM = {



    kdc = KDC. EXAMPLE.COM



    auth_to_local = RULE:[1:$1@$0]



    pkinit_anchors = FILE:<path where you install the root certificate>/
certnew.pem



    pkinit_kdc_hostname = KDC.EXAMPLE.COM



    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature



    pkinit_eku_checking = kpServerAuth



 }
```

PKINIT情報を追加すると、構成ファイルは次のようになります。

```
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認証の構成

PAM構成ファイルは、PAM認証に使用されるモジュールを示します。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を使用する場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_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を使用する場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_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を使用する場合、変更後の構成ファイルは次のようになります。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die]  pam_krb5.so preauth_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

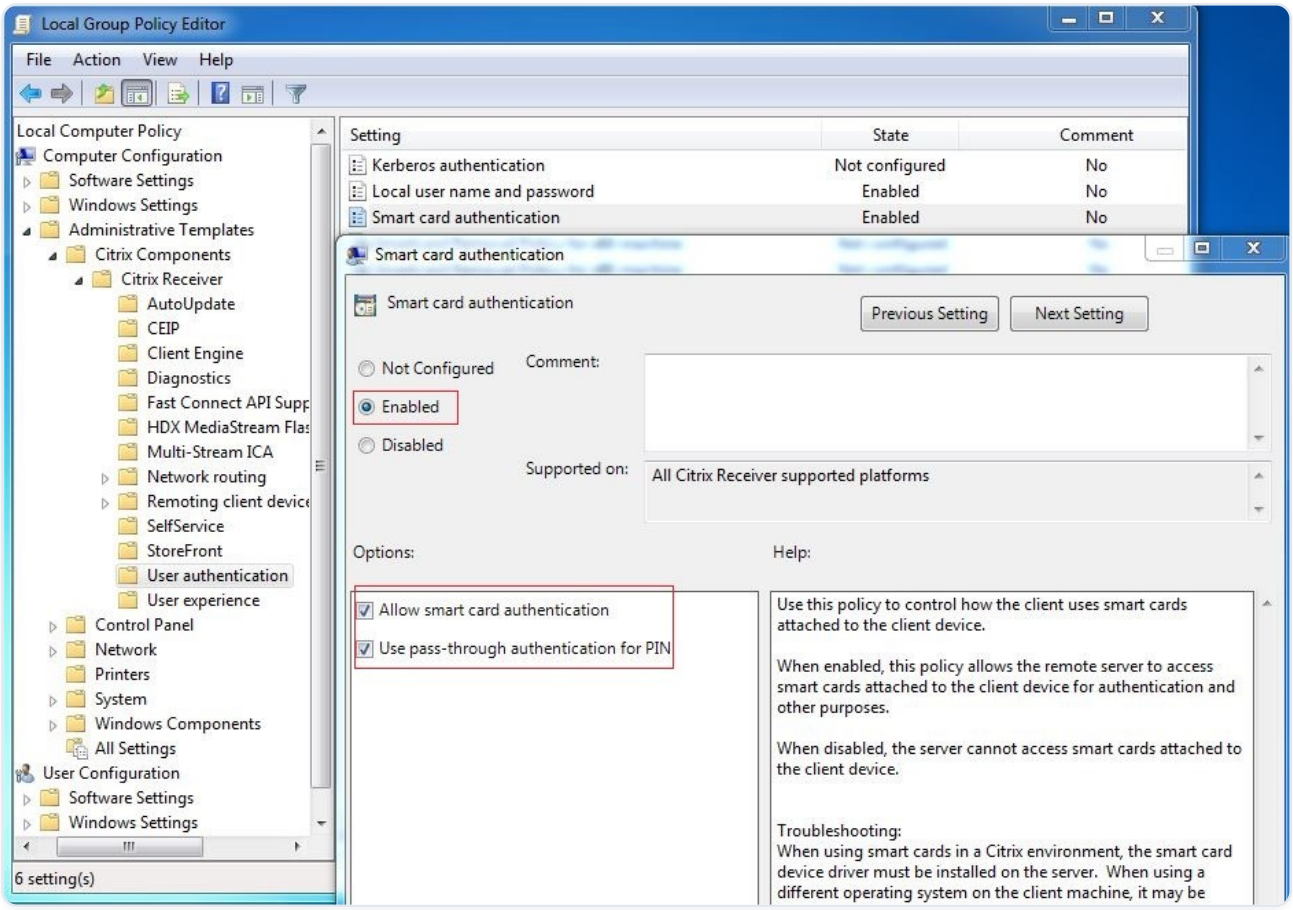
password    required      pam_pkcs11.so

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) は、仮想デスクトップおよびアプリケーションの起動でパススルー認証を実装するCitrixの機能です。この機能により、ユーザーがPINを入力する回数が減ります。Linux VDAでSSOを使用するには、Citrix Workspaceアプリを構成します。構成はWindows VDAと同じです。詳しくは、Knowledge Centerの記事「CTX133982」を参照してください。

Citrix Workspaceアプリでグループポリシーを構成する際に、次のようにスマートカード認証を有効にします。



高速スマートカードログオン

高速スマートカードは、既存のHDX PC/SCベースのスマートカードリダイレクトを改善したものです。これにより、高遅延のWAN環境でスマートカードを使用する際のパフォーマンスが向上します。詳しくは、「スマートカード」を参照してください。

Linux VDAは、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。

- シトリックス レシーバー Windows用 4.12
- Windows向けCitrix Workspaceアプリ1808以降

クライアントで高速スマートカードログオンを有効にする

高速スマートカードログオンは、VDAではデフォルトで有効になっており、クライアントではデフォルトで無効になっています。クライアントで高速スマートカードログオンを有効にするには、関連するStoreFrontサイトのdefault.icaファイルに次のパラメーターを含めます。

```
[WFClient]

SmartCardCryptographicRedirection=On
```

クライアントで高速スマートカードログオンを無効にする

クライアントで高速スマートカードログオンを無効にするには、関連するStoreFrontサイトのdefault.icaファイルから**SmartCardCryptographicRedirection**パラメーターを削除します。

使用方法

スマートカードを使用してLinux VDAにログオンする

SSOシナリオと非SSOシナリオの両方で、スマートカードを使用してLinux VDAにログオンできます。

- SSOシナリオでは、キャッシュされたスマートカード証明書とPINを使用してStoreFront™に自動的にログオンします。StoreFrontでLinux仮想デスクトップセッションを起動すると、PINがLinux VDAに渡され、スマートカード認証が行われます。
- 非SSOシナリオでは、証明書を選択してPINを入力し、StoreFrontにログオンするよう求められます。

  StoreFrontにログオンするためにPINを入力する(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/pin-new.png)

StoreFrontでLinux仮想デスクトップセッションを起動すると、Linux VDAへのログオンダイアログボックスが次のように表示されます。ユーザー名はスマートカードの証明書から抽出され、ログオン認証のためにPINを再度入力する必要があります。

この動作はWindows VDAと同じです。

XenDesktopログオンの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/xendesktop-login.png)

スマートカードを使用してセッションに再接続する

セッションに再接続するには、スマートカードがクライアントデバイスに接続されていることを確認してください。そうしないと、スマートカードが接続されていないため再認証が失敗し、Linux VDA側で灰色のキャッシュウィンドウが表示されてすぐに終了します。この場合、スマートカードを接続するように促す他のプロンプトは表示されません。

ただし、StoreFront側では、セッションに再接続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場合、StoreFront Webは次のようなアラートを表示する可能性があります。

スマートカードを挿入する画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/please-insert-smart-card.png)

制限事項

スマートカード取り外しポリシー

Linux VDAは、スマートカードの取り外しに対してデフォルトの動作のみを使用します。Linux VDAに正常にログオンした後でスマートカードを取り外しても、セッションは接続されたままであり、セッション画面はロックされません。

その他のスマートカードとPKCS#11ライブラリのサポート

サポートリストにはCoolKeyスマートカードのみが記載されていますが、Citrixは汎用スマートカードリダイレクトソリューションを提供しているため、他のスマートカードやPKCS#11ライブラリを試すことができます。特定のスマートカードまたはPKCS#11ライブラリに切り替えるには：

1. すべての`libcoolkeypk11.so`インスタンスをPKCS#11ライブラリに置き換えます。
2. PKCS#11ライブラリのパスをレジストリに設定するには、次のコマンドを実行します。

```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v
"PKCS11LibPath" -d "PATH"
```

ここで、**PATH**は、/usr/lib64/pkcs11/libcoolkeypk11.soのようなPKCS#11ライブラリを指します。

3. クライアントで高速スマートカードログオンを無効にします。

二重ホップシングルサインオン認証

この機能は、StoreFrontストアへのアクセスに入力されたユーザー資格情報を、Citrix Workspaceアプリ for LinuxおよびCitrix Receiver for Linux 13.10のAuthManagerモジュールに挿入します。挿入後、クライアントを使用して、Linux仮想デスクトップセッション内から仮想デスクトップとアプリケーションに、ユーザー資格情報を再度入力することなくアクセスできます。

注:

この機能は、Citrix Workspaceアプリ for LinuxおよびCitrix Receiver for Linux 13.10でサポートされています。

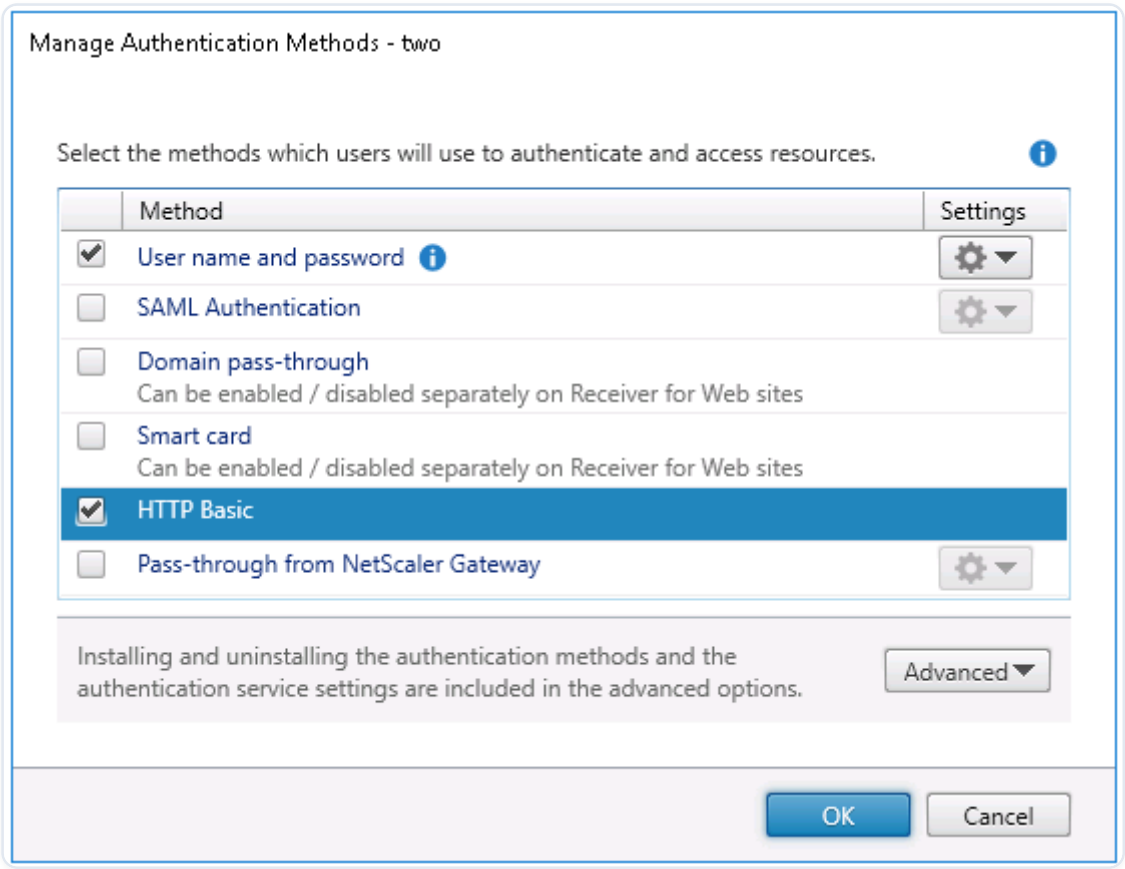
この機能を有効にするには：

1. Linux VDAに、Citrix Workspaceアプリ for LinuxまたはCitrix Receiver for Linux 13.10をインストールします。

   Citrix WorkspaceアプリまたはCitrix Receiver用のアプリを、Citrixダウンロードページからダウンロードします。

   デフォルトのインストールパスは/opt/Citrix/ICAClient/です。アプリを別のパスにインストールする場合は、ICAROOT環境変数を設定して実際のインストールパスを指すようにします。

2. Citrix StoreFront™管理コンソールで、ターゲットストアの**HTTP Basic**認証方法を追加します。



3. HTTP Basic認証を許可するために、AuthManager構成ファイル（$ICAROOT/config/AuthManConfig.xml）に次のキーを追加します。

```
<Protocols>

     <HTTPBasic>

         <Enabled>True</Enabled>

    </HTTPBasic>

</Protocols>
```

4. 指定されたディレクトリにルート証明書をインストールするには、次のコマンドを実行します。

```
cp rootcert.pem $ICAROOT/keystore/cacerts/

$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
```

5. 機能を有効にするには、次のコマンドを実行します。

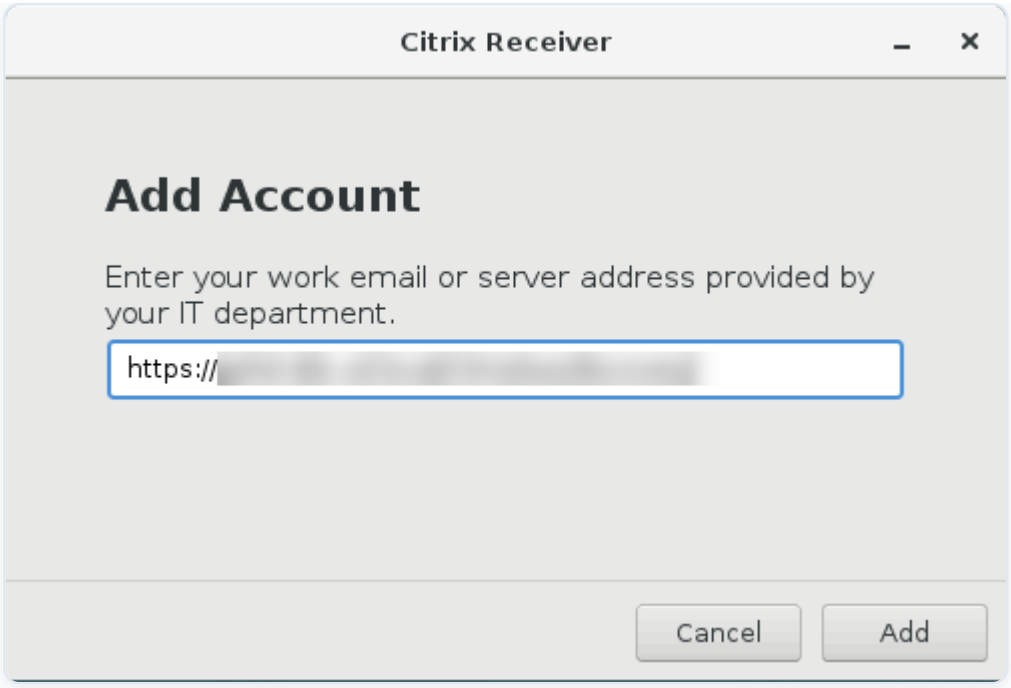
```
/opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\System\CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d
"0x00000001"
```

6. Linux仮想デスクトップセッションを起動し、そのセッション内でCitrix Workspaceアプリ for LinuxまたはCitrix Receiver for Linux 13.10を起動します。

   Linux仮想デスクトップセッション内でCitrix Workspaceアプリ for LinuxまたはCitrix Receiver for Linux 13.10を初めて起動するときに、ストアアカウントの入力を求められます。その後は、以前に指定したストアに自動的にログオンされます。

   注:

   ストアアカウントとしてHTTPS URLを入力します。

認証されていないセッションを構成する

この記事の情報を使用して、認証されていないセッションを構成します。この機能を使用するためにLinux VDAをインストールする際に、特別な設定は必要ありません。

注:

> 認証されていないセッションを構成する際は、セッションの事前起動はサポートされていないことに注意してください。セッションの事前起動は、Android版Citrix Workspace™アプリでもサポートされていません。

認証されていないストアを作成する

Linux VDAで認証されていないセッションをサポートするには、StoreFrontを使用して認証されていないストアを作成します。

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

認証されていないストアを作成したら、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して、認証されていない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のドキュメントの手順に従ってください。

認証されていないセッションのアイドル時間を設定する

認証されていないセッションのデフォルトのアイドルタイムアウトは10分です。この値は、レジストリ設定**AnonymousUserIdleTime**で構成されます。この値を変更するには、**ctxreg**ツールを使用します。たとえば、このレジストリ設定を5分に設定するには、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v AnonymousUserIdleTime
 -d 0x00000005
```

認証されていないユーザーの最大数を設定する

認証されていないユーザーの最大数を設定するには、レジストリキー**MaxAnonymousUserNumber**を使用します。この設定は、単一のLinux VDAで同時に実行される認証されていないセッションの数を制限します。このレジストリ設定を構成するには、**ctxreg**ツールを使用します。たとえば、値を32に設定するには、次のようにします。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
 "HKLM\System\CurrentControlSet\Control\Citrix"  -v MaxAnonymousUserNumber
 -d  0x00000020
```

> **重要:**

> 認証されていないセッションの数を制限してください。同時に起動されるセッションが多すぎると、利用可能なメモリが不足するなど、VDAで問題が発生する可能性があります。

トラブルシューティング

認証されていないセッションを構成する際は、次の点を考慮してください。

- **認証されていないセッションへのログオンに失敗しました。**

レジストリが次のように更新されていることを確認してください（0に設定）。

```
sudo /opt/Citrix/VDA/bin/ctxreg read –k
"HKLM\System\CurrentControlSet\Control\Citrix" –v MaxAnonymousUserNumber
```

ncsd サービスが実行されており、**passwd** キャッシュを有効にするように構成されていることを確認します。

```
ps  uax | grep nscd

cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

passwd キャッシュ変数が有効になっている場合は、**no** に設定し、**ncsd** サービスを再起動します。この構成を変更した後、Linux VDA を再インストールする必要がある場合があります。

- **KDE を使用した非認証セッションでロック画面ボタンが表示される。**

ロック画面ボタンとメニューは、非認証セッションではデフォルトで無効になっています。ただし、KDE では引き続き表示されることがあります。KDE で、特定のユーザーに対してロック画面ボタンとメニューを無効にするには、構成ファイル **$Home/.kde/share/config/kdeglobals** に次の行を追加します。例:

```
[KDE Action Restrictions]

action/lock_screen=false
```

ただし、**KDE Action Restrictions** パラメーターが、**/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals** のようなグローバルな **kdeglobals** ファイルで変更不可として構成されている場合、ユーザー構成は無効になります。

この問題を解決するには、システム全体の **kdeglobals** ファイルを修正して、**\[KDE Action Restrictions\]** セクションにある **\[$i\]** タグを削除するか、または、システム全体の構成を直接使用してロック画面ボタンとメニューを無効にしてみてください。KDE 構成の詳細については、KDE System Administration/Kiosk/Keys ページを参照してください。

LDAPS の構成

セキュア LDAP (LDAPS) を使用すると、Active Directory 管理対象ドメインでセキュア Lightweight Directory Access Protocol を有効にし、SSL (Secure Socket Layer)/TLS (Transport Layer Security) を介した通信を提供できます。

デフォルトでは、クライアントとサーバーアプリケーション間の LDAP 通信は暗号化されません。SSL/TLS を使用する LDAP (LDAPS) を使用すると、Linux VDA と LDAP サーバー間の LDAP クエリコンテンツを保護できます。

以下の Linux VDA コンポーネントは LDAPS に依存しています。

- ブローカーエージェント: Delivery Controller™ への Linux VDA 登録
- ポリシーサービス: ポリシー評価

LDAPS の構成には以下が含まれます。

- Active Directory (AD)/LDAP サーバーで LDAPS を有効にする
- クライアントで使用するためにルート CA をエクスポートする
- Linux VDA で LDAPS を有効/無効にする
- サードパーティプラットフォーム向けに LDAPS を構成する
- SSSD を構成する
- Winbind を構成する
- Centrify を構成する
- Quest を構成する

AD/LDAPサーバーでLDAPSを有効にする

Microsoft認証局 (CA) またはMicrosoft以外のCAのいずれかから、適切にフォーマットされた証明書をインストールすることで、LDAP over SSL (LDAPS) を有効にできます。

> **ヒント:**

> ドメインコントローラーにエンタープライズルートCAをインストールすると、LDAP over SSL/TLS (LDAPS) が自動的に有効になります。

> 証明書のインストール方法とLDAPS接続の確認方法の詳細については、Microsoftサポートサイトの「How to enable LDAP over SSL with a third-party certification authority」を参照してください。

多層 (2層または3層など) の認証局階層がある場合、ドメインコントローラー上のLDAPS認証に適した証明書が自動的に提供されるわけではありません。

多層認証局階層を使用してドメインコントローラーのLDAPSを有効にする方法については、Microsoft TechNetサイトの「LDAP over SSL (LDAPS) Certificate」の記事を参照してください。

クライアント使用のためにルート認証局を有効にする

クライアントは、LDAPサーバーが信頼するCAからの証明書を使用する必要があります。クライアントのLDAPS認証を有効にするには、ルートCA証明書を信頼されたキーストアにインポートします。

ルートCAのエクスポート方法の詳細については、MicrosoftサポートWebサイトの「How to export Root Certification Authority Certificate」を参照してください。

Linux VDAでLDAPSを有効または無効にする

Linux VDAでLDAPSを有効または無効にするには、次のスクリプトを（管理者としてログオンした状態で）実行します。

このコマンドの構文には、以下が含まれます。

- 提供されたルートCA証明書を使用してLDAP over SSL/TLSを有効にする:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
```

- SSL/TLSなしのLDAPにフォールバックする

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable
```

LDAPS専用のJavaキーストアは **/etc/xdl/.keystore** にあります。影響を受けるレジストリキーは次のとおりです。

```
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers


HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy


HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS


HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore
```

サードパーティプラットフォーム向けLDAPSの構成

Linux VDAコンポーネントの他に、VDAに準拠するいくつかのサードパーティソフトウェアコンポーネント（SSSD、Winbind、Centrify、Questなど）もセキュアなLDAPを必要とする場合があります。以下のセクションでは、LDAPS、STARTTLS、またはSASLサインアンドシールを使用してセキュアなLDAPを構成する方法について説明します。

> **ヒント:**
>
> これらのソフトウェアコンポーネントのすべてが、セキュアなLDAPを確保するためにSSLポート636を使用することを好むわけではありません。そしてほとんどの場合、LDAPS（ポート636上のSSL経由LDAP）は389上のSTARTTLSと共存できません。

SSSD

オプションに従って、ポート636または389でSSSDセキュアLDAPトラフィックを構成します。詳細については、SSSD LDAP Linux manページを参照してください。

ウィンバインド

Winbind LDAPクエリはADSメソッドを使用します。Winbindはポート389でのStartTLSメソッドのみをサポートしています。影響を受ける構成ファイルは、**/etc/samba/smb.conf** および **/etc/openldap/ldap.conf** (RHELの場合) または **/etc/ldap/ldap.conf** (Ubuntuの場合) です。ファイルを次のように変更します。

- smb.conf

  `ldap ssl = start tls`

  `ldap ssl ads = yes` `client ldap sasl wrapping = plain`

- ldap.conf

  `TLS_REQCERT never`

あるいは、セキュアなLDAPはSASL GSSAPIサインアンドシールによって構成できますが、TLS/SSLとは共存できません。SASL暗号化を使用するには、**smb.conf** の構成を変更します。

`ldap ssl = off`

`ldap ssl ads = no` `client ldap sasl wrapping = seal`

セントリファイ

Centrifyはポート636でのLDAPSをサポートしていません。ただし、ポート389でセキュアな暗号化を提供します。詳細については、Centrifyサイトを参照してください。

クエスト

Quest Authentication Serviceはポート636でのLDAPSをサポートしていませんが、異なる方法を使用してポート389でセキュアな暗号化を提供します。

トラブルシューティング

この機能を使用すると、以下の問題が発生する可能性があります。

- **LDAPS サービスの可用性**

  AD/LDAP サーバーで LDAPS 接続が利用可能であることを確認します。ポートはデフォルトで 636 です。

- **LDAPS が有効な場合に Linux VDA の登録が失敗しました**

  LDAP サーバーとポートが正しく構成されていることを確認します。まずルート CA 証明書を確認し、それが AD/LDAP サーバーと一致していることを確認します。

- **誤ってレジストリが変更されました**

  enable_ldaps.sh を使用せずに誤って LDAPS 関連のキーが更新された場合、LDAPS コンポーネントの依存関係が壊れる可能性があります。

- **Wireshark またはその他のネットワーク監視ツールから、LDAP トラフィックが SSL/TLS を介して暗号化されていない**

  デフォルトでは、LDAPS は無効になっています。**/opt/Citrix/VDA/sbin/enable_ldaps.sh** を実行して強制的に有効にします。

- **Wireshark またはその他のネットワーク監視ツールから LDAPS トラフィックがない**

  LDAP/LDAPS トラフィックは、Linux VDA の登録とグループポリシーの評価が行われるときに発生します。

- **AD サーバーで ldp connect を実行して LDAPS の可用性を確認できませんでした**

  IP アドレスの代わりに AD FQDN を使用してください。

- **/opt/Citrix/VDA/sbin/enable_ldaps.sh スクリプトを実行してルート CA 証明書をインポートできませんでした**

  CA 証明書のフルパスを指定し、ルート CA 証明書が正しいタイプであることを確認してください。一般的に、サポートされているほとんどの Java Keytool タイプと互換性があるはずです。サポートリストにない場合は、まずタイプを変換できます。証明書形式の問題が発生した場合は、Citrix® は base64 エンコードされた PEM 形式を推奨します。

- **Keytool -list でルート CA 証明書を表示できませんでした**

  /opt/Citrix/VDA/sbin/enable_ldaps.sh を実行して LDAPS を有効にすると、証明書が /etc/xdl/.keystore にインポートされ、キーストアを保護するためにパスワードが設定されます。パスワードを忘れた場合は、スクリプトを再実行してキーストアを作成できます。

Xauthority の設定

Linux VDA は、対話型リモート処理のために X11 ディスプレイ機能 (`xterm` および `gvim` を含む) を使用する環境をサポートします。この機能は、XClient と XServer 間の安全な通信を確保するために必要なセキュリティメカニズムを提供します。

この安全な通信の権限を保護するには、次の2つの方法があります。

- **Xhost**。デフォルトでは、Xhost はローカルホストの XClient のみが XServer と通信することを許可します。リモートの XClient が XServer にアクセスすることを許可する場合は、特定のコンピューターで Xhost コマンドを実行して権限を付与する必要があります。または、代わりに **xhost +** を使用して、任意の XClient が XServer に接続できるようにすることもできます。
- **Xauthority**。`.Xauthority` ファイルは、各ユーザーのホームディレクトリにあります。これは、XServer の認証のために xauth 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、その特定のディスプレイへの接続を認証するために Cookie が使用されます。

仕組み

Xorg が起動すると、`.Xauthority` ファイルが Xorg に渡されます。この `.Xauthority` ファイルには、次の要素が含まれています。

- ディスプレイ番号
- リモート要求プロトコル
- Cookie 番号

xauth コマンドを使用してこのファイルを参照できます。例:

```
# xauth -f ~/.Xauthority



# > list



# > us01msip06:107  MIT-MAGIC-COOKIE-1  fb228d1b695729242616c5908f11624b
```

XClient が Xorg にリモートで接続する場合、2つの前提条件を満たす必要があります。

- **DISPLAY** 環境変数をリモートの XServer に設定します。
- Xorg の Cookie 番号のいずれかを含む `.Xauthority` ファイルを取得します。

Xauthority の設定

リモート X11 ディスプレイのために Linux VDA で Xauthority を有効にするには、次の2つのレジストリキーを作成する必要があります。

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"XauthEnabled" -d "0x00000001" --force



sudo /opt/Citrix/VDA/bin/ctxreg create -k
```

```
"HKLM\System\CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v
"ListenTCP" -d "0x00000001" --force
```

Xauthority を有効にした後、`.Xauthority` ファイルを XClient に手動で渡すか、共有ホームディレクトリをマウントして渡します。

- `.Xauthority` ファイルを XClient に手動で渡す

  ICA® セッションを起動した後、Linux VDA は XClient 用の `.Xauthority` ファイルを生成し、ログオンユーザーのホームディレクトリに保存します。この `.Xauthority` ファイルをリモートの XClient マシンにコピーし、DISPLAY および XAUTHORITY 環境変数を設定できます。DISPLAY は `.Xauthority` ファイルに保存されているディスプレイ番号であり、XAUTHORITY は Xauthority のファイルパスです。例については、次のコマンドを参照してください。

  ```
  export DISPLAY={Display number stored in the Xauthority file}



  export XAUTHORITY={the file path of .Xauthority}
  ```

  注:

  XAUTHORITY 環境変数が設定されていない場合、デフォルトで `~/.Xauthority` ファイルが使用されます。

- 共有ホームディレクトリをマウントして `.Xauthority` ファイルを XClient に渡す

  便利な方法は、ログオンユーザー用に共有ホームディレクトリをマウントすることです。Linux VDA が ICA セッションを開始すると、ログオンユーザーのホームディレクトリの下に `.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 と共有されている場合、ユーザーはこの `.Xauthority` ファイルを XClient に手動で転送する必要はありません。DISPLAY および XAUTHORITY 環境変数が正しく設定されると、GUI は XServer デスクトップに自動的に表示されます。

トラブルシューティング

Xauthority が機能しない場合は、次のトラブルシューティング手順に従ってください。

1. root 権限を持つ管理者として、すべての Xorg クッキーを取得します。

   ```
   ps aux | grep -i xorg
   ```

   このコマンドは、Xorg プロセスと、起動時に Xorg に渡されたパラメーターを表示します。別のパラメーターは、どの `.Xauthority` ファイルが使用されているかを表示します。例:

   ```
   /var/xdl/xauth/.Xauthority110
   ```

   Xauth コマンドを使用してクッキーを表示します。

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   ```

2. **Xauth** コマンドを使用して、`~/.Xauthority` に含まれるクッキーを表示します。同じディスプレイ番号の場合、表示されるクッキーは Xorg と XClient の `.Xauthority` ファイルで同じである必要があります。
3. クッキーが同じである場合は、Linux VDA の IP アドレス（例: 10.158.11.11）と公開デスクトップのディスプレイ番号（例: 160）を使用して、リモートディスプレイポートのアクセシビリティを確認します。

   XClient マシンで次のコマンドを実行します。

   ```
   telnet 10.158.11.11 6160
   ```

   ポート番号は 6000 + \<display number\> の合計です。

   この telnet 操作が失敗した場合、ファイアウォールが要求をブロックしている可能性があります。

フェデレーション認証サービス

概要

Citrix 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た特権コンポーネントです。ユーザーに証明書を動的に発行し、スマートカードを持っているかのように Active Directory 環境にログオンできるようにします。この機能により、StoreFront™ は、Security Assertion Markup Language (SAML) アサーションなど、より広範な認

証オプションを使用できます。SAML は、インターネット上で従来の Windows ユーザーアカウントの代替として一般的に使用されます。

> **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 FAS サーバーとのデータ転送に短い接続を使用します。
>
> 次の図は、Microsoft 証明機関と統合され、StoreFront および VDA にサポートサービスを提供する FAS を示しています。

FAS アーキテクチャ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/fas-architecture.png)

ユーザーが Citrix 環境へのアクセスを要求すると、信頼された StoreFront サーバーは FAS に接続します。FAS は、そのセッションの証明書で単一の Citrix Virtual Apps または Citrix Virtual Desktops™ セッションが認証できるようにするチケットを付与します。VDA がユーザーを認証する必要がある場合、FAS に接続してチケットを引き換えます。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がある署名および復号化操作をすべて 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、他の 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 Server は、ドメインユーザーに証明書を自動的に発行するため、およびそれらのユーザー証明書と秘密キーにアクセスするために、登録機関証明書と秘密キーにアクセスできるように保護されている必要があります。

Citrix Virtual Apps™ または Citrix Virtual Desktops サイトの場合：

- Delivery Controller は、最低バージョン 7.9 である必要があります。
- StoreFront サーバーは、最低バージョン 3.6 (XenApp および XenDesktop 7.9 ISO に付属のバージョン) である必要があります。
- Linux VDA は、最低バージョン 7.18 である必要があります。通常の方法でマシンカタログを作成する前に、Federated Authentication Service グループポリシー構成が VDA に正しく適用

されていることを確認してください。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：

- 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>`

- 証明書ログオンのためにWindowsを構成する

  `<http://support.citrix.com/article/CTX206156>`

- フェデレーション認証サービスをインストールする

  フェデレーション認証サービス

証明書ログオンのためにWindowsを構成する

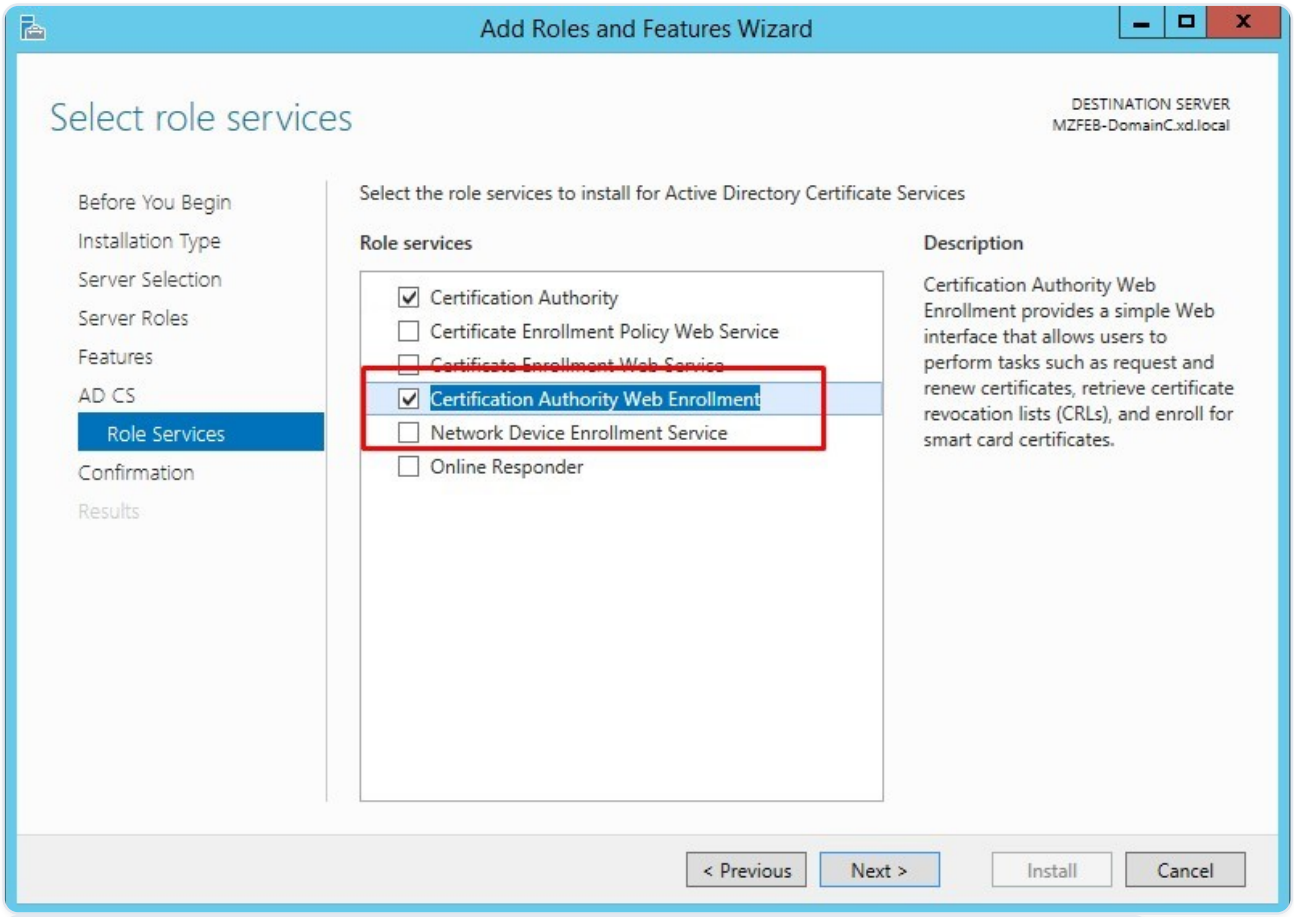
証明書ログオンのためにWindowsを構成する方法については、Knowledge Centerの記事CTX206156を開き、**Smart_card_config_Citrix_Env.pdf**ファイル（以下「PDFファイル」）をダウンロードして読んでください。PDFファイルに従って以下の手順を実行しますが、各手順で示されている相違点や補足事項に注意してください。操作対象のマシン（例：AD、Delivery Controller、StoreFront）に特に注意してください。

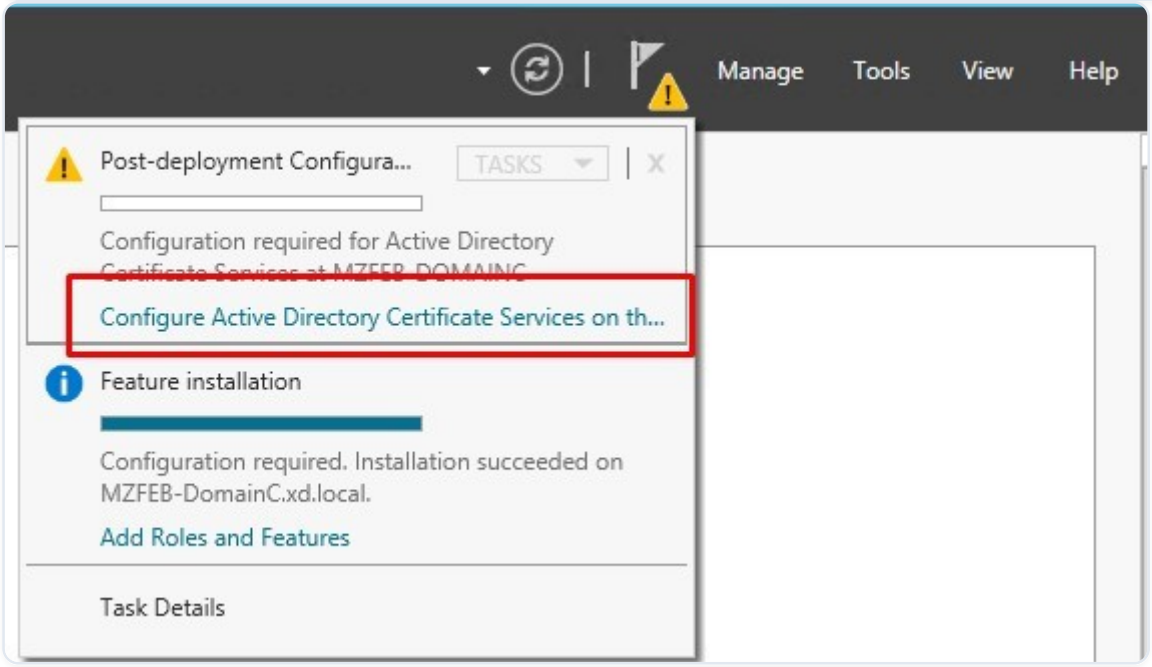
Windowsドメインをセットアップする（AD上）

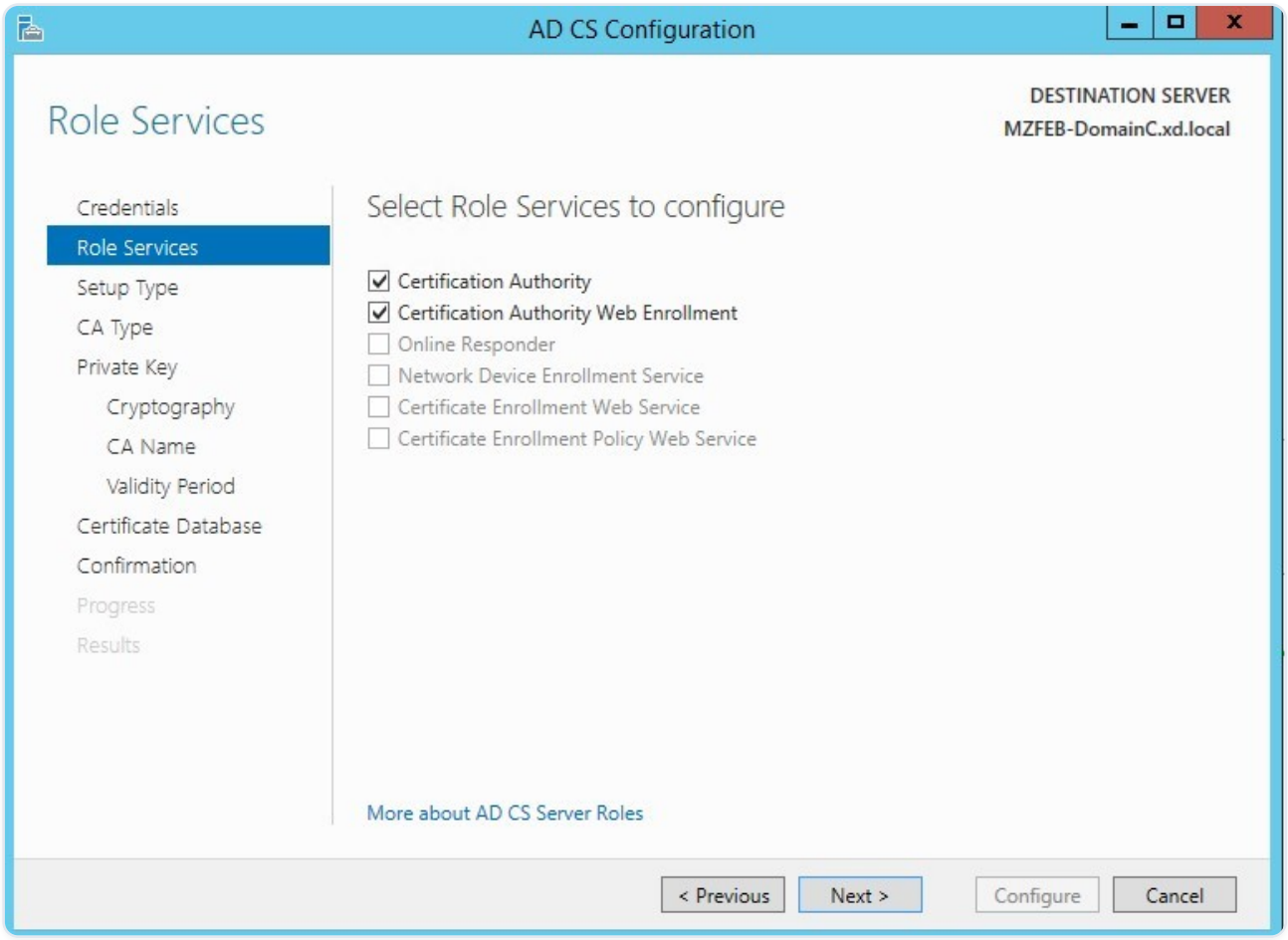
ドメインコントローラーの役割をインストールする

PDFファイルの「**Installing Domain Controller Roles**」セクションを参照してください。

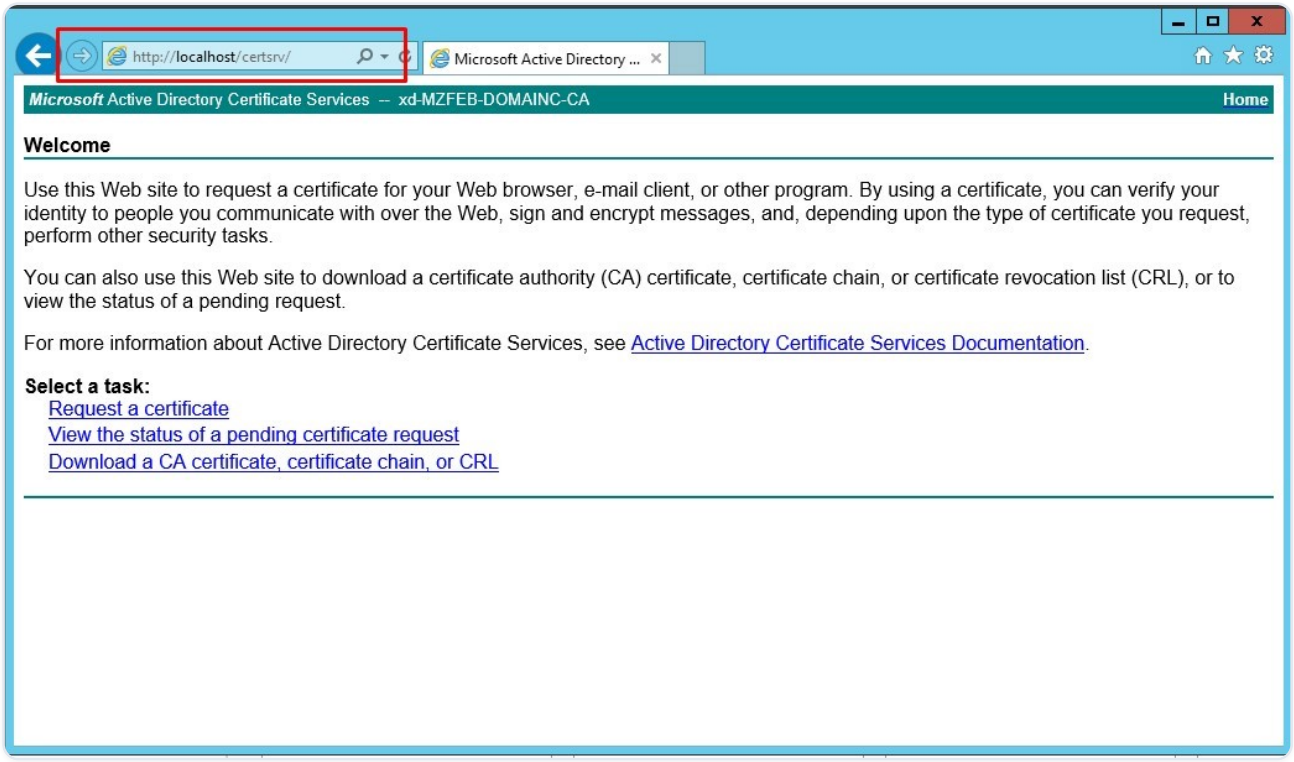
Active Directory証明書サービスをインストールする際は、以下のオプションが選択されていることを確認してください。


Add Roles and Features Wizard
Select role services
DESTINATION SERVER
MZFEB-DomainC.xd.local
Before You Begin
Installation Type
Server Selection
Server Roles
Features
AD CS
Role Services
Confirmation
Results
Select the role services to install for Active Directory Certificate Services
Role services
Certification Authority
Certificate Enrollment Policy Web Service
Certificate Enrollment Web Service
Certification Authority Web Enrollment
Network Device Enrollment Service
Online Responder
Description
Certification Authority Web Enrollment provides a simple Web interface that allows users to perform tasks such as request and renew certificates, retrieve certificate revocation lists (CRLs), and enroll for smart card certificates.
< Previous
Next >
Install
Cancel


Manage
Tools
View
Help
Post-deployment Configura...
TASKS
Configuration required for Active Directory Certificate Services at MZFEB-DOMAINC
Configure Active Directory Certificate Services on th...
Feature installation
Configuration required. Installation succeeded on MZFEB-DomainC.xd.local.
Add Roles and Features
Task Details



`http://localhost/certsrv/` を開き、以下のウェルカムページが表示されるか確認してください。表示されていれば、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。



スマートカード使用のために証明機関を準備する

補足はありません。PDFファイルの「**Preparing the Certificate Authority for Smart card usage**」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

補足はありません。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 上)

Microsoft IIS での HTTPS の構成

補足はありません。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 CA 証明書を取得する (AD 上)

補足はありません。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。

Windows に信頼された CA 証明書をインストールする

補足はありません。PDFファイルの「**Windows への信頼された CA 証明書のインストール**」セクションを参照してください。

Citrix StoreFront の構成 (StoreFront 上)

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構成後、共通ストアのベースURLは`http://`ではなく`https://`に強制的に設定されます。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の任意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またはSAMLを使用するように構成できますが、両方を同時に使用することはできません。SAMLが選択されている場合、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。

FASストアの作成イメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/create-store.png)

スマートカード認証方法の管理イメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/manage-authentication-methods.png)

作成されたFASストアのイメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/fas-store-created.png)

Internet Explorerを起動し、FASストアのURLを開きます（例: `https://mzgwy-ddc.xd.local/Citrix/FASWeb`）。

注: FASストアのURLには**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ます。

1. フェデレーション認証サービスをインストールする
2. StoreFrontサーバーでフェデレーション認証サービスプラグインを有効にする
3. グループポリシーを構成する
4. フェデレーション認証サービス管理コンソールを使用して、次の操作を行います。(a) 提供されたテンプレートを展開する、(b) 証明機関をセットアップする、(c) フェデレーション認証サービスが証明機関を使用することを承認する
5. ユーザー規則を構成する

各手順の詳細については、「フェデレーション認証サービス」を参照してください。各手順における次の相違点または補足事項に注意してください。操作対象のマシン（例: AD、Delivery Controller、StoreFront、FASサーバー）に特に注意してください。

フェデレーション認証サービスをインストールする（FASサーバー上）

セキュリティのため、FASはドメインコントローラーまたは証明機関と同様の方法で保護された専用サーバーにインストールしてください。

StoreFrontストアでFederated Authentication Serviceプラグインを有効にする（StoreFront上）

以下のコマンドで、StoreFrontの構成時に入力したFASストア名と同じものが使用されていることを確認してください。たとえば、この例ではFASがストア名です。

$StoreVirtualPath = "/Citrix/**FAS**"

デリバリーコントローラー™を構成する（デリバリーコントローラー上）

Federated Authentication Serviceを使用するには、Delivery Controllerが接続できるStoreFrontサーバーを信頼するように構成します。これには、**Set-BrokerSite -TrustRequestsSentToTheXmlServicePort $true** PowerShellコマンドレットを実行します。場合によっては、`Add-PSSnapin citrix.*`を先に実行する必要があるかもしれません。

グループポリシーを構成する（FASサーバーおよびAD上）

このセクションのステップ1～7を実行するには、管理者である必要があります。ステップ1はFASサーバーで実行し、ステップ2～7はADで実行する必要があります。

ステップ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てください。

FASポリシー設定の画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/set-fas-policy.png)

インセッション証明書サポートを有効にする

Linux VDAはインセッション証明書をサポートしていません。

Federated Authentication Service管理コンソールを使用する（FASサーバー上）

補足はありません。Federated Authentication Serviceの記事を参照してください。

証明書テンプレートを展開する（FASサーバー上）

補足はありません。Federated Authentication Serviceの記事を参照してください。

Active Directory Certificate Services (FASサーバー上) を設定する

補足はありません。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 (FASサーバー上) を承認する

補足はありません。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。

ユーザー規則 (FASサーバー上) を構成する

補足はありません。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。

詳細については、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**」セクションにある「**委任された登録エージェント**」および「**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 ADFS展開

フェデレーション認証サービス用のADFS IdPの展開方法については、フェデレーション認証サービス ADFS展開を参照してください。

Linux VDAを構成する

FASサーバーを設定する

新規のLinux VDAインストールの場合、FASを使用するには、ctxinstall.shまたはctxsetup.shの実行中にCTX_XDL_FAS_LISTを求められたときに、各FASサーバーのFQDNを入力します。Linux VDAはADグループポリシーをサポートしていないため、代わりにFASサーバーのセミコロン区切りリストを指定できます。サーバーアドレスが削除された場合は、その空白を**\<none\>**というテキスト文字列で埋め、サーバーアドレスの順序を変更せずに維持してください。

既存のLinux VDAインストールをアップグレードする場合は、ctxsetup.shを再実行してFASサーバーを設定できます。または、次のコマンドを実行してFASサーバーを設定し、`ctxvda`サービスを再起動して設定を有効にすることもできます。

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



service ctxvda restart
```

`ctxreg`を介してFASサーバーを更新するには、次のコマンドを実行します。

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k
"HKLM\Software\Citrix\VirtualDesktopAgent\Authentication\UserCredentialSer
vice" -v "Addresses" -d "<Your-FAS-Server-List>"



service ctxvda restart
```

ルートCA証明書をインストールする

ユーザーの証明書を検証するために、VDAにルートCA証明書をインストールします。前述の「**Microsoft CA (AD上) からCA証明書を取得する**」の手順からADルート証明書を取得するか、ルートCAサーバー`http://CA-SERVER/certsrv`からDER形式をダウンロードしてください。

> **注:**
>
> 以下のコマンドは、中間証明書の設定にも適用されます。
>
> DERファイル（.crt、.cer、.der）をPEMに変換するには、次のようなコマンドを実行します。

```
sudo openssl x509 -inform der -in root.cer -out root.pem
```

次に、ルートCA証明書をopensslディレクトリにインストールするには、次のようなコマンドを実行します。

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

> **注:**
>
> ルートCA証明書を**/root**パスの下に置かないでください。そうしないと、FASはルートCA証明書への読み取り権限を持ちません。

FASを構成する

FASを構成するには、次のスクリプトを実行します。

```
sudo /opt/Citrix/VDA/sbin/ctxfascfg.sh
```

> **注:**
>
> 上記のスクリプトは、単一のルートCA証明書を持つ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、/etc/krb5.confに中間パスを次のように追加します。

> EXAMPLE.COM = { …

> pkinit_anchors = FILE:/etc/pki/CA/certs/root.pem pkinit_pool = FILE:/etc/pki/CA/certs/intermediate.pem

> … }

> 2つの環境変数が追加され、`ctxfascfg.sh`をサイレントモードで実行できるようになります。

- **CTX_FAS_ADINTEGRATIONWAY=winbind \| sssd \| centrify** – Active Directory統合方法を示します。これは、`CTX_EASYINSTALL_ADINTEGRATIONWAY`が指定されている場合に`CTX_EASYINSTALL_ADINTEGRATIONWAY`と等しくなります。`CTX_EASYINSTALL_ADINTEGRATIONWAY`が指定されていない場合、`CTX_FAS_ADINTEGRATIONWAY`は独自の値を設定します。
- **CTX_FAS_ROOT_CA_PATH=\<root_CA_certificate\>** – ルートCA証明書のフルパスを指定します。

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（例: `/etc/pki/CA/certs/root.pem`）を入力します。

スクリプトはkrb5-pkinitおよびpam_krb5パッケージをインストールし、関連する構成ファイルを設定します。

制限事項

- FAS は限られたプラットフォームと AD 統合メソッドをサポートしています。以下のマトリックスを参照してください。

	ウィンバインド	SSSD	セントリファイ
RHEL 7.7 /CentOS 7.7	√	√	√
ウブントゥ 18.04	√	×	√

	ウィンバインド	SSSD	セントリファイ
ウブントゥ 16.04	√	×	√
SLES 12.3	√	×	√

- FAS はまだロック画面をサポートしていません。セッション内でロックボタンをクリックすると、FAS を使用してそのセッションに再度ログオンすることはできません。
- このリリースでは、Federated Authentication Service architectural overview 記事にまとめられている一般的な FAS 展開のみがサポートされ、**Windows 10 Azure AD Join** は含まれません。

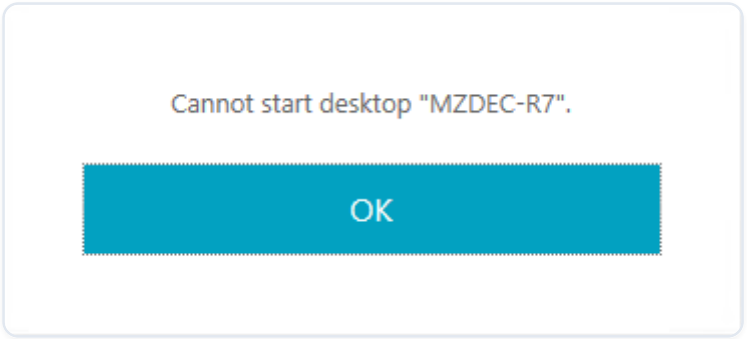
トラブルシューティング

FAS のトラブルシューティングを行う前に、Linux VDA が正しくインストールおよび構成されており、パスワード認証を使用して一般的なストアで非 FAS セッションを正常に起動できることを確認してください。

非 FAS セッションが正常に動作する場合は、**Login** クラスの HDX ログレベルを VERBOSE に、VDA ログレベルを TRACE に設定します。Linux VDA のトレースログを有効にする方法については、Knowledge Center の記事 CTX220130 を参照してください。

FAS サーバー構成エラー

FAS ストアからのセッション起動に失敗します。たとえば、次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見つけてください。

```
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
```

解決策

次のコマンドを実行して、Citrix レジストリ値「HKEY_LOCAL_MACHINE\\SOFTWARE\\Citrix\\VirtualDesktopAgent\\Authentication\\UserCredentialService」が \<Your-FAS-Server-List\> に設定されていることを確認します。

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

既存の設定が正しくない場合は、前述の Set FAS servers の手順に従って、再度設定してください。

ルートCA証明書の設定が正しくありません

FASストアからのセッション起動に失敗します。灰色のウィンドウが表示され、数秒後に消えます。

ルートCA証明書の設定が正しくないためにログオンが無効になったことを示す画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/incorrect-root-ca-cert-config.png)

/var/log/xdl/hdx.logを確認し、次のようなエラーログを見つけてください。

```
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
```

解決策

ルートCA証明書のフルパスが/etc/krb5.confに正しく設定されていることを確認してください。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

    ......

}
```

既存の設定が正しくない場合は、前のルートCA証明書をインストールするの手順に従って再度設定してください。

または、ルートCA証明書が有効であるかどうかを確認してください。

シャドウアカウントマッピングエラー

FASはSAML認証によって構成されています。ADFSユーザーがADFSログオンページでユーザー名とパスワードを入力した後、次のエラーが発生する可能性があります。

シャドウアカウントマッピングエラーの画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/shadow-account-mapping-error.png)

このエラーは、ADFSユーザーが正常に検証されたものの、ADにシャドウユーザーが構成されていないことを示しています。

解決策

ADでシャドウアカウントを設定します。

ADFSが構成されていません

FASストアへのログオン試行中に次のエラーが発生します。

ADFSが構成されていないことを示す画像(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/adfs-not-configured.png)

原因は、FASストアがSAML認証を使用するように構成されているにもかかわらず、ADFS展開が不足していることです。

解決策

フェデレーション認証サービス用にADFS IdPを展開します。詳細については、「フェデレーション認証サービス ADFS展開」(/ja-jp/federated-authentication-service/current-release/deployment-architectures/adfs.html)を参照してください。

関連情報

- 一般的なFAS展開については、「フェデレーション認証サービス アーキテクチャの概要」(/ja-jp/federated-authentication-service/1912-ltsr/deployment-architectures.html)の記事にまとめられています。
- 「ハウツー」記事は、「フェデレーション認証サービス詳細構成」(/ja-jp/federated-authentication-service/1912-ltsr/config-manage.html)の章で紹介されています。

既知の問題

FASを使用している場合、英語以外の文字を含む公開デスクトップまたはアプリセッションを起動しようとすると失敗する可能性があります。

英語以外の文字を含むセッションの起動に失敗するイメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/launching-nonenglish-session-fails.png)

回避策

CAツールで**テンプレートの管理**を右クリックし、**Citrix_SmartcardLogon**テンプレートを**この Active Directory情報から構築**から**要求で提供**に変更します。

要求で提供オプションのイメージ(/en-us/linux-virtual-delivery-agent/1912-ltsr/media/supply-in-the-request.png)

__index__

スマートカードを使用したパススルー認証

ユーザーは、Linux仮想デスクトップセッションにログオンするときに、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。ユーザーは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、このページの「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 Package Manager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
```

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。

s

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
   ```

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>
```

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
   ```

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
   ```

   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。

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
   ```

   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。

   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
   ```

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
```

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。

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります：



スマートカードを無効にするには:

```
sudo /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
```

結果は次のようになります：



- 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。

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

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。

```
yum install coolkey
```

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
```

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
```

/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



 }
```

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。

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。

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。

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。

（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
```

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

ユーザーは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、ユーザー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。ユーザーが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、ユーザーは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。



ユーザーが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、ユーザーは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、シトリックス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"
   ```

   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。

Linux Virtual Delivery Agentのドキュメント履歴

次の表に、Linux Virtual Delivery Agent 1912のドキュメントに対する重要な変更点を示します。 これらの更新に関する通知を受け取るには、RSSフィードを購読してください。

日付	変更	説明
15 Nov 2024	Citrix Virtual Apps and Desktops 1912 LTSR累積更新プログラム10のリリース	詳しくは、「新機能」を参照してください。
04 Jun 2024	Citrix Virtual Apps and Desktops 1912 LTSR累積更新プログラム9 Update 1のリリース	詳しくは、「新機能」を参照してください。
30 Apr 2024	Virtual Delivery Agent 1912 LTSR累積更新プログラム9のリリース	詳しくは、「新機能」を参照してください。
11 Sep 2023	Virtual Delivery Agent 1912 LTSR累積更新プログラム8のリリース	詳しくは、「新機能」を参照してください。
13 Jun 2023	Virtual Delivery Agent 1912 LTSR累積更新プログラム7 Hotfix 1（19.12.7001）のリリース	詳しくは、「新機能」を参照してください。
15 Mar 2023	Virtual Delivery Agent 1912 LTSR累積更新プログラム7のリリース	詳しくは、「新機能」を参照してください。
31 Oct 2022	Virtual Delivery Agent 1912 LTSR累積更新プログラム6のリリース	詳しくは、「新機能」を参照してください。

LINUX VIRTUAL DELIVERY AGENT 1912 LTSR

日付	変更	説明
09 Mar 2022	Virtual Delivery Agent 1912 LTSR累積更新プログラム5のリリース	詳しくは、「新機能」を参照してください。
03 Nov 2021	Virtual Delivery Agent 1912 LTSR累積更新プログラム4のリリース	詳しくは、「新機能」を参照してください。
12 May 2021	Virtual Delivery Agent 1912 LTSR累積更新プログラム3のリリース	詳しくは、「新機能」を参照してください。
19 Nov 2020	Virtual Delivery Agent 1912 LTSR累積更新プログラム2のリリース	詳しくは、「新機能」を参照してください。
07 May 2020	Virtual Delivery Agent 1912 LTSR累積更新プログラム1のリリース	詳しくは、「新機能」を参照してください。
18 Dec 2019	Linux Virtual Delivery Agent 1912 LTSRのリリース	Linux Virtual Delivery Agent 1912の新機能