



Linux Virtual Delivery Agent 2407

Contents

Linux Virtual Delivery Agent 2407	6
新增功能	6
已修复的问题	9
已知问题	10
第三方声明	13
弃用	13
系统要求	15
安装概述	19
准备安装	20
使用轻松安装创建已加入域的 VDA	21
使用 MCS 创建未加入域的 Linux VDA	42
使用轻松安装创建未加入域的 Linux VDA (预览版)	54
使用 Machine Creation Services (MCS) 创建 Linux VDA	63
使用 Citrix Provisioning 创建 Linux VDA	84
在 Citrix DaaS Standard for Azure 中创建 Linux VDA	85
手动安装 Linux VDA	90
在 Amazon Linux 2 、 RHEL 和 Rocky Linux 上手动安装 Linux VDA	90
在 SUSE 上手动安装 Linux VDA	121
在 Ubuntu 上手动安装 Linux VDA	145
在 Debian 上手动安装 Linux VDA	173
配置	201
管理	201
Linux VDA 数据收集计划	202

HDX Insight	204
与 Citrix Telemetry Service 集成	205
通过 Azure 进行 Linux VDA 自助更新	208
Linux VM 和 Linux 会话指标	211
日志收集	213
会话影子处理	218
监视服务守护程序	223
故障排除	225
其他	234
适用于 HTML5 的 Citrix Workspace 应用程序支持	234
创建 Python3 虚拟环境	235
将 NIS 与 Active Directory 集成	237
IPv6	242
LDAPS	243
Xauthority	247
身份验证	249
使用 Azure Active Directory 进行身份验证	250
双跃点单点登录身份验证	254
联合身份验证服务	256
FIDO2 (预览版)	264
非 SSO 身份验证	265
智能卡	266
未经身份验证的 (匿名) 用户进行访问	276
File	278

文件复制和粘贴	279
文件传输	280
图形	283
自动 DPI 缩放	284
客户端电池状态显示	285
图形配置和微调	287
HDX 屏幕共享	297
多显示器支持	304
未虚拟化的 GPU	310
会话水印	312
多会话 Linux VDA 上的共享 GPU 加速	316
系统托盘	317
Thinwire 渐进式显示	322
常规内容重定向	324
客户端驱动器映射	325
USB 设备重定向	326
剪贴板重定向	333
键盘	335
X 键盘扩展 (XKB) 配置	336
客户端输入法编辑器 (IME)	336
客户端 IME 用户界面同步	337
动态键盘布局同步	341
软键盘	344
支持多语言输入	347

多媒体	348
音频功能	349
浏览器内容重定向	352
HDX 网络摄像机视频压缩	362
未加入域的 Linux VDA	366
策略支持列表	369
打印	387
打印最佳做法	387
PDF 打印	394
Remote PC Access	394
会话	406
自适应传输	406
HDX 自适应吞吐量	409
会话登录屏幕上的自定义背景和横幅消息	409
按会话用户划分的自定义桌面环境	412
使用临时主目录登录	414
发布应用程序	416
Rendezvous V1	418
Rendezvous V2	421
Secure HDX (技术预览版)	425
使用 DTLS 保护用户会话安全	426
使用 TLS 保护用户会话安全	426
会话可靠性	430
会话录制 (预览版)	432

虚拟通道 SDK (预览版)	434
Wayland (预览版)	435
最佳做法	436
使用 Machine Creation Services (MCS) 在 Google 云端平台 (GCP) 上创建 Linux VDA	436
使用 Ansible 管理您的部署	455

Linux Virtual Delivery Agent 2407

August 20, 2024

重要：

[生命周期里程碑](#)中介绍了当前版本 (CR) 和长期服务版本 (LTSR) 的产品生命周期策略。

Linux Virtual Delivery Agent (VDA) 支持在任何位置从安装了 Citrix Workspace 应用程序的任何设备访问 Linux 虚拟应用程序和桌面。

您可以根据[受支持的 Linux 发行版](#)交付虚拟应用程序和桌面。在您的 Linux 虚拟机 (VM) 上安装 VDA 软件，配置 Delivery Controller，然后使用 Citrix Studio 向用户提供这些应用程序和桌面。

新增功能

August 20, 2024

2407 中的新增功能

支持 **RHEL 9.4/8.10**、**Rocky Linux 9.4/8.10** 和 **Debian 12.5/11.9**

Linux VDA 现在支持以下 Linux 发行版：

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

有关支持的 Linux 发行版的详细信息，请参阅[系统要求](#)。

需要新的 .NET 运行时

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

基于令牌的注册已扩展到本地环境（技术预览版）

此版本将基于令牌的注册功能扩展到了本地环境。使用轻松安装创建未加入域的 VDA 时，您现在可以：

- 使用包含云和本地 Delivery Controller 的计算机目录注册未加入域的 VDA。
- 使用 Citrix Web Studio 创建一个空的计算机目录并生成注册令牌。

有关详细信息，请参阅[使用轻松安装创建未加入域的 Linux VDA（预览版）](#)。

SSSD 被指定为默认域加入方法

SSSD 现在是加入域的默认方法，取代了 Winbind。此更改会影响您在交互模式下运行 `ctxsetup.sh` 或轻松安装脚本 (`ctxinstall.sh`) 时的配置行为。有关详细信息，请参阅[使用轻松安装创建已加入域的 VDA](#)。

在升级期间委派包管理器来处理配置文件

此版本在升级期间委派包管理器 **rpm** 或 **dpkg** 来处理配置文件。此外，此版本解决了由 rpmLint 标记的一些问题，并重新构建了 RPM 包以更好地符合 RPMLint 的要求。有关详细信息，请参阅[步骤 13: 升级 Linux VDA（可选）](#)。

支持智能卡移除设置

现在，您可以通过编辑 Linux VDA 上的以下注册表项，决定在会话期间将已登录用户的智能卡从智能卡读卡器中移除时将发生什么：有关详细信息，请参阅[智能卡移除设置](#)。

能够阻止特定的客户端打印机重定向到 HDX 会话

现在，您可以通过配置 Citrix 策略设置打印机驱动程序映射和兼容性来阻止特定的客户端打印机重定向到 HDX 会话。有关详细信息，请参阅[打印机驱动程序映射和兼容性](#)。

音频的容损模式（技术预览版）

现在，支持通过 Enlightened Data Transport (EDT) 容损协议来传输音频。当用户通过延迟和数据包丢失率均较高的网络进行连接时，此功能改进了实时流技术推送的用户体验。启用此功能后，Citrix Virtual Apps and Desktops 中的“自适应传输”将使用“EDT 容损”协议来实现更出色的音频体验。默认情况下，此功能处于禁用状态。有关详细信息，请参阅[音频功能](#)一文中的[音频的容损模式](#)。

实施了用于改善音频性能的数据包丢失隐藏（技术预览版）

自 2407 版本起，抖动缓冲机制得到了改进，并为自适应音频编解码器添加了数据包丢失隐藏 (PLC) 功能。PLC 有助于重新构建丢失的数据包。此增强功能有助于改进数据包容损和抖动容限，因而提高了音频的容损模式（EDT 有损）的音频性能。

要启用此功能，还需要启用 [音频的容损模式](#) 功能。有关详细信息，请参阅 [音频功能](#) 一文中的 [用于改善音频性能的数据包丢失隐藏（技术预览版）](#)。

在 **Debian 12.5**、**RHEL 9.x** 和 **Rocky Linux 9.x** 中支持 **PipeWire**

在 Debian 12.5、RHEL 9.x 和 Rocky Linux 9.x 中，Linux VDA 现在支持 PipeWire，在这些版本中，它是默认的音频服务。有关详细信息，请参阅 [音频功能](#)。

增强了图形性能

此版本通过在 Linux VDA 上将 JPEG Turbo 升级到版本 3 (V3) 增强了图形性能。

登录界面上显示登录过程的详细信息

现在，会话用户在打开 Linux VDA 会话时可以在登录界面上看到以下登录过程详细信息：

- 评估 **Citrix** 组策略设置
- 验证用户访问权限
- 启动桌面环境（如果是虚拟桌面会话）或启动应用程序窗口（如果是虚拟应用程序会话）

在 **KDE** 和 **XFCE** 桌面环境中支持键盘布局同步

此版本将动态键盘布局同步功能扩展到了 XFCE 和 KDE Plasma 5 桌面环境。有关详细信息，请参阅 [动态键盘布局同步](#)。

将 **evdev** 设为默认 **X** 键盘扩展 (**XKB**) 规则

为了通过更快的响应和可预测的键盘行为来实现更好的用户体验，Linux VDA 现在将 evdev 指定为键盘配置的默认 XKB 规则。

有关详细信息，请参阅 [X 键盘扩展 \(XKB\) 配置](#)。

LDAP 查询性能优化

在新版本中，删除了与 LDAP 服务器的不必要通信，并使用索引查询密钥来实现更出色的 LDAP 查询性能并降低成本。

改进了 Cloud Connector 的 VDA 可扩展性

此版本对 VDA 可扩展性进行了改进，使每个 Citrix Cloud Connector（4 个 vCPU，10 GB 内存）现在可以支持 6,000 个 Linux VDA。有关详细信息，请参阅 [Cloud Connector 的大小和扩展注意事项](#)。

HDX 自适应吞吐量现在默认处于启用状态

有关详细信息，请参阅 [HDX 自适应吞吐量](#)。

注意：

从本版本开始，我们为有兴趣使用 Ansible 进行部署管理的客户提供了一篇全面的最佳做法文章。有关详细信息，请参阅 [使用 Ansible 管理您的部署](#)。

早期版本中的新增功能

有关在 1912 LTSR 之后发布的版本（直到 2402 LTSR）中包含的新增功能，请参阅 [新增功能历史记录](#)。

已修复的问题

August 20, 2024

Linux Virtual Delivery Agent 2407 包含下列已修复的问题：

- Root 用户无法将只读文件复制到使用 CDM 映射的客户端设备。[CVADHELP-25124]
- 处理来自 GPF 文件的大型组策略集时，组策略解析器可能会消耗 100% 的 CPU。[CVADHELP-25323]
- 当会话断开连接时，ctxnsap 进程意外退出，导致 Citrix Application Delivery Management (ADM) 无法显示 HDX Insight 信息。[CVADHELP-25493]
- 在模板计算机上不正确执行 ad_join.sh 会生成一个名为 mcs_setting_flag 的标志，从而阻止 MCS 新创建的计算机加入域。此修复程序在 deploymcs.sh 脚本中引入了检查，以识别该标志并在检测到它时将其删除。[CVADHELP-25117]

已知问题

August 20, 2024

在本版本中确定了以下问题：

- 智能卡服务在智能卡身份验证期间泄露文件描述符，导致新的智能卡访问受阻。出现此问题的原因是，默认情况下，大多数 Linux 发行版将每个进程的最大打开文件数限制为 1,024 个。当智能卡服务用尽此限制时，它将无法再建立新的连接，从而有效地阻止了后续的智能卡访问。

此问题会影响启用了智能卡登录的 VDA。症状包括 `grep PID**`。
`/var/log/xdl/hdx.log` 中出现大量的无法接受新连接：
打开到文件太多错误，并且 `/proc/${pid}/fd/` 中会累积
文件描述符，其中 **`${pid}`** 表示 **`ctxscardsd`** 的进程 ID。
要确定 PID，请使用命令 `**systemctl status
ctxscardsd`

要缓解此问题，您可以增大智能卡服务的最大打开文件数上限，或者重新启动智能卡服务。在尝试重新启动该服务之前，请确保不存在活动会话。使用以下命令增大上限或重新启动该服务：

- 要重新启动智能卡服务，请执行以下操作：

```
1 systemctl restart ctxscardsd
```

- 要查询当前该服务的最大打开文件数，请执行以下操作：

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

- 要设置智能卡服务的最大打开文件数，请执行以下操作：

1. 以只读模式打开 **`ctxscardsd.service`** 文件以检查当前设置：

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

2. 将以下行添加到 **`ctxscardsd.service`** 的 **Service** 部分以增大上限：

```
1 LimitNOFILE=65536
```

3. 重新加载 **`systemd`** 守护程序并重新启动 **`ctxscardsd`** 服务：

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

4. 验证新上限：

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

注意：

增大最大打开文件数可以延长用完文件描述符之前的时间，但最终可能仍需要重新启动 **ctxscardsd**。

[LNXVDA-17784]

- 如果不指定 `CTX_XDL_DESKTOP_ENVIRONMENT` 变量，则无法知道在后续进程中要使用的桌面。因此，未配置与桌面相关的环境变量，并且依赖这些变量的应用程序或插件可能无法按预期运行。[LNXVDA-16212]
- 由于 GNOME 存在问题，在 RHEL 8.X、Rocky Linux 8.x、RHEL 9.x 和 Rocky Linux 9.x 上将 **samba-winbind** 升级到 4.18.6 版本后，Linux VDA 无法按预期运行。有关详细信息，请参阅 <https://issues.redhat.com/browse/RHEL-17122>。
- 在 PostgreSQL 中设置的最大连接数不足，无法处理并发会话时，会发生会话启动失败。要解决此问题，请通过修改 **postgresql.conf** 文件中的 **max_connections** 设置来增加最大连接数。
- 由于 **/var/log/xdl/jproxy.log** 中引发以下 LDAP 异常，VDA 注册可能会失败：

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

要解决此问题，请执行以下操作：

- 更改 LDAP 超时值。例如，请使用以下命令将 LDAP 超时值更改为 60 秒：

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

- 通过设置搜索库加快 LDAP 查询速度。可以使用 `ctxsetup.sh` 中的 `CTX_XDL_SEARCH_BASE` 变量或者使用以下命令来设置搜索库：

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

[CVADHELP-20895]

- Microsoft 在 2022 年 11 月发布了适用于 Windows 10 的累积更新 KB5019966 和 KB5019964。更新导致域加入和注册失败。要解决此问题，请参阅知识中心文章 [CTX474888](https://support.microsoft.com/CTX474888)。
- 在使用 Kerberos 允许的 **RC4_HMAC_MD5** 加密类型的情况下，Linux VDA 可能无法向控制器注册，并出现以下错误消息：

Error: Failure unspecified at GSS-API level (Mechanism level: Encryption type RC4 with HMAC is not supported/enabled) (错误: 在 GSS-API 级别未指定故障 (机制级别: 不支持/未启用 HMAC 加密类型 RC4))

要解决此问题,请在您的 Active Directory 域中(或者,具体来说在 *OU* 上)禁用 **RC4_HMAC_MD5**,或者允许在 Linux VDA 上使用弱加密类型。之后,请使用 **klist -li 0x3e4 purge** 命令清除 Controller 和 Citrix Cloud Connector 上缓存的 Kerberos 票据,然后重新启动 Linux VDA。

要在您的 Active Directory 域中全局禁用 **RC4_HMAC_MD5**,请完成以下步骤:

1. 打开组策略管理控制台。
2. 找到目标域,然后选择 **Default Domain Policy** (默认域策略)。
3. 右键单击 **Default Domain Policy** (默认域策略) 并选择 **Edit** (编辑)。组策略管理编辑器将打开。
4. 选择计算机配置 > 策略 > **Windows** 设置 > 安全设置 > 本地策略 > 安全选项。
5. 双击网络安全: 配置 **Kerberos** 允许的加密类型。
6. 取消选中 **DES_CBC_CRC**、**DES_CBC_MD5** 和 **RC4_HMAC_MD5** 复选框,然后选择 **AES128_HMAC_SHA1**、**AES256_HMAC_SHA1** 和 将来的加密类型。

要允许在 Linux VDA 上使用弱加密类型,请完成以下步骤:

注意:

弱加密类型使您的部署容易受到攻击。

1. 在 Linux VDA 上打开 `/etc/krb5.conf` 文件。
2. 在 **[libdefaults]** 部分下添加以下条目:

```
allow_weak_crypto= TRUE
```

- Linux VDA 不支持使用 SecureICA 进行加密。在 Linux VDA 上启用 SecureICA 会导致会话启动失败。
- 在 GNOME 桌面会话中,尝试更改键盘布局可能会失败。[CVADHELP-15639]
- Ubuntu 图形: 在 HDX 3D Pro 中,调整 Desktop Viewer 的大小后,应用程序周围可能会显示一个黑框,或者有时背景会显示为黑色。
- 注销会话后,可能不会删除 Linux VDA 打印重定向创建的打印机。
- 目录中包含大量文件和子目录时 CDM 文件会缺失。如果客户端有太多文件或目录,则可能会出现此问题。
- 在本版本中,仅支持对非英语语言使用 UTF-8 编码。
- 适用于 Android 的 Citrix Workspace 应用程序的 CapsLock 键状态可能会在会话漫游期间反转。漫游与适用于 Android 的 Citrix Workspace 应用程序的现有连接时,Caps Lock 键状态可能会丢失。解决方法:使用扩展键盘上的 Shift 键在大写与小写之间切换。
- 使用适用于 Mac 的 Citrix Workspace 应用程序连接至 Linux VDA 时,含 Alt 的快捷键有时不起作用。默认情况下,对于左侧和右侧 Options/Alt 键,适用于 Mac 的 Citrix Workspace 应用程序都会发送 AltGr。您可以在 Citrix Workspace 应用程序设置中修改此行为,但是结果因不同的应用程序而异。

- Linux VDA 重新加入域时注册失败。重新加入将生成一组全新的 Kerberos 密钥。但是，Broker 可能会根据先前的一组 Kerberos 密钥使用缓存的过时 VDA 服务票据。VDA 尝试连接到 Broker 时，Broker 可能无法与 VDA 建立返回安全上下文。常见症状是 VDA 注册失败。

VDA 服务票据过期并续订后，此问题最终会自行解决。但是，由于服务票据的有效期很长，因此可能需要很长时间。

解决方法：清除 Broker 的票据缓存。重新启动 Broker 或在 Broker 上以管理员身份从命令提示窗口运行以下命令：

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

此命令会清除 Citrix Broker Service 运行所在的网络服务主体持有的 LSA 缓存中的所有服务票据。此命令也会删除其他 VDA 的服务票据，因而可能会影响其他服务。但是，此操作不会造成负面影响，因为这些服务可在需要时从 KDC 重新获取这些服务票据。

- 不支持音频即插即用。您可以先将音频捕获设备连接到客户端计算机，然后开始在 ICA 会话中录制音频。如果在启动了音频录制应用程序后连接捕获设备，应用程序可能会无响应，因此您必须将其重新启动。如果在录制期间拔下捕获设备，可能会出现类似的问题。
- 适用于 Windows 的 Citrix Workspace 应用程序可能会在音频录制期间遇到音频失真问题。

第三方声明

August 20, 2024

[Linux Virtual Delivery Agent 2407](#) (PDF 下载)

此 Linux VDA 版本可能包含根据该文档中定义的条款许可使用的第三方软件。

弃用

August 20, 2024

本文中的声明提前通知您正在逐渐淘汰的平台、Citrix 产品和功能，以便您能够及时制定业务决策。Citrix 将监视客户使用情况和反馈以确定其退出时间。在后续版本中声明可能会有更改，可能不会包括每个弃用的特性或功能。

有关产品生命周期支持的详细信息，请参阅 [Product Lifecycle Support Policy](#) (产品生命周期支持策略) 一文。

弃用和删除

下表显示了已弃用或删除的平台、Citrix 产品和功能。

已弃用的项目不会立即删除。Citrix 在此版本中继续支持这些项目，但将在未来的当前版本中将其删除。在 Linux VDA

中，
已删除的项目已被删除或不再受支持。

项目	已在其中宣布弃用的版本	已在其中删除的版本
支持 RHEL 9.3、RHEL 8.9	2407	2411
支持 Rocky Linux 9.3、Rocky Linux 8.9	2407	2411
支持 Ubuntu 20.04	2407	2502
支持 SUSE 15.5	2407	2411
支持 Debian 11.7/11.3	2407	2407
支持 RHEL 9.0、RHEL 8.6	2402	2407
支持 Rocky Linux 9.0、Rocky Linux 8.6	2402	2407
支持 RHEL 7.9、CentOS 7.9	2402	2407
支持 SUSE 15.4	2308	2311
支持 Rocky Linux 9.1、Rocky Linux 8.7	2305	2308
支持 RHEL 9.1、RHEL 8.7	2305	2308
支持 RHEL 8.4	2303	2308
支持 Ubuntu 18.04	2212	2305
支持 SUSE 15.3	2210	2301
支持 Debian 10.9	2206	2210
支持 SUSE 15.2	2206	2209
支持 RHEL 8.2	2206	2209
支持 RHEL 8.1、RHEL 8.3	2203	2206
支持 RHEL 7.8、CentOS 7.8	2203	2204
支持 CentOS 8.x	2110	2201
支持 SUSE 12.5	2109	2204
支持 Ubuntu 16.04	2109	2203
支持 RHEL 7.7、CentOS 7.7	2006	2009
支持 SUSE 12.3	2006	2006
支持 RHEL 6.10、CentOS 6.10	2003	2003

项目	已在其中宣布弃用的版本	已在其中删除的版本
支持 RHEL 6.9、CentOS 6.9	1909	1909
支持 RHEL 7.5、CentOS 7.5	1903	1903
支持 RHEL 7.4、CentOS 7.4	1811	1811
支持 RHEL 6.8、CentOS 6.8	1811	1811
支持 RHEL 7.3、CentOS 7.3	7.18	7.18
支持 RHEL 6.6、CentOS 6.6	7.16	7.16
SUSE 11.4	7.16	7.16

系统要求

August 20, 2024

Linux VDA 的当前版本与 Citrix Virtual Apps and Desktops 保持一致。它还向后兼容尚未结束其生命周期的 Citrix Virtual Apps and Desktops 的早期版本。要获取有关 Citrix 产品生命周期的信息，以及了解何时 Citrix 会停止支持特定的产品版本，请参阅 [Citrix 产品生命周期表](#)。

Linux VDA 与 Windows VDA 的配置过程略有差别。所有 Delivery Controller 场都能为 Windows 和 Linux 桌面提供代理服务。

本文档中未涉及的组件（例如 Citrix Workspace 应用程序）的系统要求在其各自的文档集中进行说明。

有关在长期服务版本 (LTSR) 环境中使用当前版本 (CR) 以及其他常见问题解答的信息，请参阅 [知识中心文章](#)。

支持的 Linux 发行版、Xorg 版本和桌面环境

有关此版本的 Linux VDA 支持的 Linux 发行版、Xorg 版本和桌面环境的列表，请参见下表。有关详细信息，请参阅 [XorgModuleABIVersions](#)。

Linux 发行版	Xorg 版本	支持的桌面
Amazon Linux 2	1.20	MATE
Debian 12.5	1.20	GNOME、GNOME Classic、KDE、MATE、 Xfce*
Debian 11.9	1.20	GNOME、GNOME Classic、KDE、MATE

Linux 发行版	Xorg 版本	支持的桌面
RHEL 9.4/9.3/9.2	1.20	GNOME、 Xfce
RHEL 8.10/8.9/8.8	1.20	GNOME、GNOME Classic、MATE、 Xfce
Rocky Linux 9.4/9.3/9.2	1.20	GNOME、 Xfce *
Rocky Linux 8.10/8.9/8.8	1.20	GNOME、GNOME Classic、KDE、MATE、 Xfce
SUSE 15.5	1.20	GNOME、GNOME Classic、MATE
Ubuntu 22.04	1.21	GNOME、GNOME Classic、KDE、MATE、 Xfce
Ubuntu 20.04	1.20	GNOME、GNOME Classic、KDE、MATE、 Xfce

注意：

- 当操作系统供应商提供的支持过期时，Citrix 解决问题的能力可能会受到限制。对于已弃用或已删除的平台，请参阅[弃用](#)。
- 必须至少安装一个桌面。可以通过 `ctxinstall.sh` 或 `ctxsetup.sh` 脚本指定要在会话中使用的 GNOME、GNOME Classic 或 MATE 桌面环境。
- 根据 [Red Hat Enterprise Linux 文档](#)，GNOME 是 RHEL 9 中唯一可用的桌面环境。但是，如果您安装了 EPEL 存储库，也可以在 RHEL 9 和 Rocky Linux 9 中使用 **Xfce** 桌面环境。
- 请勿在 Ubuntu 上使用 [HWE kernel](#) 或 [HWE Xorg](#)。
- 您的用户名格式必须符合当前显示管理器的 `systemd` 语法规则。有关 `systemd` 用户名语法的详细信息，请参阅[用户/组名称语法](#)。

.Net 要求

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

支持的主机平台和虚拟化环境

- 裸机服务器
- Amazon Web Services (AWS)

- XenServer (以前称为 Citrix Hypervisor)
- Google 云端平台 (GCP)
- 基于内核的虚拟机 (KVM)
- Microsoft Azure
- Microsoft Hyper-V
- VMware vSphere Hypervisor
- Nutanix AHV

注意：

在所有情况下，受支持的处理器架构均为 x86-64。

自 2203 版本起，您可以在 Microsoft Azure、AWS 和 GCP 上为 Citrix Virtual Apps and Desktops 以及 Citrix DaaS (以前称为 Citrix Virtual Apps and Desktops 服务) 托管 Linux VDA。要将这些公有云主机连接添加到您的 Citrix Virtual Apps and Desktops 部署中，您需要 Citrix Universal 混合多云 (HMC, Citrix Universal Hybrid Multi-Cloud) 许可证。

Active Directory 集成软件包

Linux VDA 支持以下 Active Directory 集成软件包或产品：

	Winbind	SSSD	Centrify	PBIS	Quest
Amazon Linux 2	是	是	是	是	否
Debian 12.5/11.9	是	是	是	是	是
RHEL 9.4/9.3/9.2、Rocky Linux 9.4/9.3/9.2/8.10/8.9/8.8	是	是	否	否	是 (Quest v4.1 及更高版本)
RHEL 8.10/8.9/8.8	是	是	是	是	是 (Quest v4.1 及更高版本)
SUSE 15.5	是	是	是	是	是
Ubuntu 22.04/20.04	是	是	是	是	是 (Quest v4.1 及更高版本)

Cloud Connector 的大小和扩展注意事项

使用 Citrix Cloud Connector 将 Linux VDA 连接到控制平面时，请注意以下事项，具体取决于 Citrix 内部测试：

- 每个 Citrix Cloud Connector (4 个 vCPU, 10 GB 内存) 可以支持 6,000 个 Linux VDA。
- 在每个资源位置中部署两个 Cloud Connector 以实现高可用性, 并在每个资源位置中最多部署 6,000 个 Linux VDA。

数据库注意事项

- 我们建议您仅将 SQLite 用于 VDI 模式, 将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS, 您可以指定使用 SQLite 或 PostgreSQL, 而不必手动安装。除非通过 **/etc/xdl/db.conf** 另行指定, 否则 Linux VDA 默认使用 PostgreSQL。
- 对于手动安装, 您必须手动安装 SQLite、PostgreSQL 或者两者。如果您同时安装了 SQLite 和 PostgreSQL, 则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

HDX 3D Pro

借助 Citrix Virtual Apps and Desktops 的 HDX 3D Pro, 您可以交付使用图形处理器 (GPU) 进行硬件加速实现最佳性能的桌面和应用程序。

虚拟机管理程序

对于 Linux VDA, HDX 3D Pro 与以下虚拟机管理程序兼容:

- XenServer (以前称为 Citrix Hypervisor)
- VMware vSphere Hypervisor
- Nutanix AHV
- Microsoft Azure
- Amazon Web Services (AWS)
- Google 云端平台 (GCP)

注意:

虚拟机管理程序与某些 Linux 发行版兼容。

要对 Amazon Linux 2 使用 HDX 3D Pro, 我们建议您安装 NVIDIA 驱动程序 470。

GPU

对于 Linux VDA, HDX 3D Pro 支持以下类型的 GPU:

NVIDIA vGPU 要了解您的 Linux 发行版支持的 NVIDIA GPU 卡，请转到 [NVIDIA 产品支持列表](#)，然后查看虚拟机管理程序或裸机操作系统、软件产品部署、硬件支持和来宾操作系统支持列。

确保为 GPU 卡安装了最新的 vGPU 驱动程序。目前，Linux VDA 最高支持 vGPU 17。有关详细信息，请参阅 [NVIDIA 虚拟 GPU 软件支持的 GPU](#)。

未虚拟化的 **GPU** 在 Linux VDA 文档中，未虚拟化的 GPU 是指：

- Remote PC Access 场景中使用的 GPU
- 从虚拟机管理程序传递的 GPU

支持适用于 **Linux** 的 **NVIDIA Capture SDK** 的 **NVIDIA GPU** 对于支持 [适用于 Linux 的 NVIDIA Capture SDK](#) 的 NVIDIA GPU，请通过在安装 Linux VDA 时将 **CTX_XDL_HDX_3D_PRO** 设置为 **Y** 来启用 HDX 3D Pro。无需执行其他配置。启用 HDX 3D Pro 后，硬件加速默认处于启用状态。

提示：

要将 HDX 3D Pro 与 NVIDIA GPU 配合使用，必须安装支持 NVIDIA Capture SDK 版本 8 的 NVIDIA 显卡驱动程序版本。有关详细信息，请参阅 [NVIDIA Capture SDK 文档](#)。

安装概述

July 15, 2024

此部分将指导您完成以下过程：

- [准备安装](#)
- [使用轻松安装创建已加入域的 VDA](#)
- [使用 MCS 创建未加入域的 Linux VDA](#)
- [使用轻松安装创建未加入域的 Linux VDA（预览版）](#)
- [使用 MCS 创建 Linux VDA](#)
- [使用 Citrix Provisioning 创建 Linux VDA](#)
- [在 Citrix DaaS Standard for Azure 中创建 Linux VDA](#)
- [手动安装 Linux VDA](#)
 - [在 Amazon Linux 2、CentOS、RHEL 和 Rocky Linux 上手动安装 Linux VDA](#)
 - [在 SUSE 上手动安装 Linux VDA](#)

- 在 [Ubuntu](#) 上手动安装 Linux VDA
- 在 [Debian](#) 上手动安装 Linux VDA

准备安装

July 16, 2024

本文介绍了一种根据您的具体情况选择合适的安装方法的方法，并引导您阅读相关安装指南。有关规划 Citrix Virtual Apps and Desktops 和 Citrix DaaS 部署的信息，请分别参阅[准备安装](#)和[入门：规划和构建部署](#)。

在开始安装 VDA 之前，请注意以下事项：

- 通过检查 [Linux VDA 的系统要求](#) 来验证您的目标部署是否得到官方支持。
- 决定是否将您的 VDA 连接到域。
- 根据您的预配方法和域加入决定，从下表中选出合适的链接：

	轻松安装脚本	Machine Creation Services (MCS)	Citrix Provisioning	纯手动	注意
已加入域	使用轻松安装创建已加入域的 VDA	使用 Machine Creation Services (MCS) 创建 Linux VDA	使用 Citrix Provisioning 创建 Linux VDA	手动安装 Linux VDA	不适用
未加入域	使用轻松安装创建未加入域的 Linux VDA (预览版)	使用 MCS 创建未加入域的 Linux VDA	不支持	不支持	仅限 DaaS

注意：

- Remote PC Access 用例不支持 MCS。
- 要进行批量预配，您可以自行使用 MCS、Citrix Provisioning 或第三方自动化。对于第三方自动化，您可以选择在脚本中加入轻松安装脚本（推荐）或自动执行纯手动安装步骤。
- 要在单个 VM 上快速进行概念证明 (POC)，我们建议您使用轻松安装。

- 您还可以运行轻松安装脚本 (ctxinstall.sh -s) 来快速更新环境变量，例如 CTX_XDL_DDC_LIST 和 CTX_XDL_LDAP_LIST。

使用轻松安装创建已加入域的 VDA

August 20, 2024

重要:

- 对于全新安装，建议您参阅本文以进行快速安装。本文将分步介绍如何使用轻松安装来安装和配置 Linux VDA。轻松安装省时又省力，与手动安装相比，更不易于出错。它可以通过自动安装必需的软件包并自定义配置文件来帮助您设置 Linux VDA 的运行环境。
- 要创建未加入域的 VDA，您可以同时使用 Machine Creation Services (MCS) 和轻松安装。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)和[使用轻松安装创建未加入域的 Linux VDA \(预览版\)](#)。
- 要了解适用于未加入域的 VDA 的功能，请转到[未加入域的 VDA](#)。

步骤 1: 准备配置信息和 Linux 计算机

收集轻松安装所需的以下配置信息:

- 主机名 - 要安装 Linux VDA 的计算机的主机名。
- 域名服务器的 IP 地址。
- NTP 服务器的 IP 地址或字符串名称。
- 域名 - 域的 NetBIOS 名称。
- 领域名称 - Kerberos 领域名称。
- 域的完全限定域名 (FQDN)。
- Active Directory (AD) 集成方法 - 目前，轻松安装支持 SSSD、Winbind、Centrify、PBIS 和 Quest。对于 RHEL 和 Rocky Linux，轻松安装仅支持 Quest。
- 用户名 - 将计算机加入域的用户名称。
- 密码 - 将计算机加入域的用户密码。
- OU - 组织单位。可选。

重要:

- 要安装 Linux VDA，请确认是否在 Linux 计算机上正确添加了存储库。
- 要启动会话，请确认是否安装了 X Windows 系统和桌面环境。
- 出于安全考虑，轻松安装不会保存域加入密码。每次在交互模式下运行轻松安装脚本 (ctxinstall.sh) 时，都必须手动输入域加入密码。在无提示模式下，必须在 `/Citrix/VDA/sbin/ctxinstall.conf` 中设置域加入密码或导出密码。我们建议您不要使用管理员帐户加入域。相反，请将加入域的权限委派给除管理员帐户

之外的 Active Directory 用户。为此，请使用委派控制向导在域控制器上委托控制权。

步骤 2：准备虚拟机管理程序

在支持的虚拟机管理程序上将 Linux VDA 当作虚拟机运行时，需要作出一些更改。根据正在使用的虚拟机管理程序平台做出以下更改。如果正在裸机硬件上运行 Linux 计算机，则无需作出任何更改。

修复 XenServer（以前称为 Citrix Hypervisor）上的时间同步问题

启用了 XenServer 时间同步功能时，在每个半虚拟化 Linux VM 中，您会遇到 NTP 和 XenServer 出现的问题。两者都试图管理系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。无需在 HVM 模式下进行任何更改。

如果您正在运行半虚拟化 Linux 内核，并且安装了 XenServer Tools，则可以检查 XenServer 时间同步功能是否存在以及是否已从 Linux VM 中启用：

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

此命令返回 0 或 1：

- 0 - 时间同步功能已启用，且必须禁用。
- 1 - 时间同步功能已禁用，无需采取任何操作。

如果 `/proc/sys/xen/independent_wallclock` 文件不存在，则不需要执行以下步骤。

如果已启用，请通过向该文件写入 1 来禁用时间同步功能：

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

要使此更改成为永久更改，并在重新启动后仍然有效，请编辑 `/etc/sysctl.conf` 文件并添加以下行：

```
xen.independent_wallclock = 1
```

要验证这些更改，请重新启动系统：

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

此命令返回值 1。

修复 Microsoft Hyper-V 上的时间同步问题

安装了 Hyper-V Linux 集成服务的 Linux VM 可应用 Hyper-V 时间同步功能来使用主机操作系统的时间。为确保系统时钟始终精确可靠，必须一同启用此功能与 NTP 服务。

从管理操作系统中：

1. 打开 Hyper-V 管理器控制台。
2. 对于 Linux VM 的设置，请选择 **Integration Services**（集成服务）。
3. 确保已选择 **Time synchronization**（时间同步）。

注意：

此方法与 VMware 和 XenServer（以前称为 Citrix Hypervisor）不同，这两款产品会禁用主机时间同步功能以避免与 NTP 发生冲突。Hyper-V 时间同步可以与 NTP 时间同步共存并互补。

修复 ESX 和 ESXi 上的时间同步问题

启用 VMware 时间同步功能后，在每个半虚拟化的 Linux VM 中，您都会遇到 NTP 和虚拟机管理程序的问题。两者都试图同步系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。

如果正在运行安装了 VMware Tools 的半虚拟化 Linux 内核：

1. 打开 vSphere Client。
2. 编辑 Linux VM 设置。
3. 在 **Virtual Machine Properties**（虚拟机属性）对话框中，打开 **Options**（选项）选项卡。
4. 选择 **VMware Tools**。
5. 在 **Advanced**（高级）框中，取消选中 **Synchronize guest time with host**（与主机同步客户机时间）。

步骤 3：安装 .NET

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

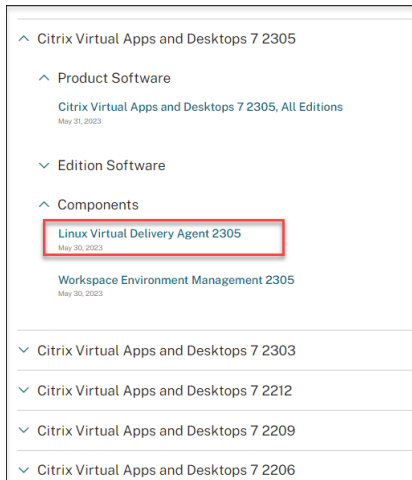
安装 .NET 后，请运行 **which dotnet** 命令查找您的运行时路径。

根据命令输出，设置 .NET Runtime 二进制文件路径。例如，如果命令输出为 /aa/bb/dotnet，请使用 /aa/bb 作为 .NET 二进制文件路径。

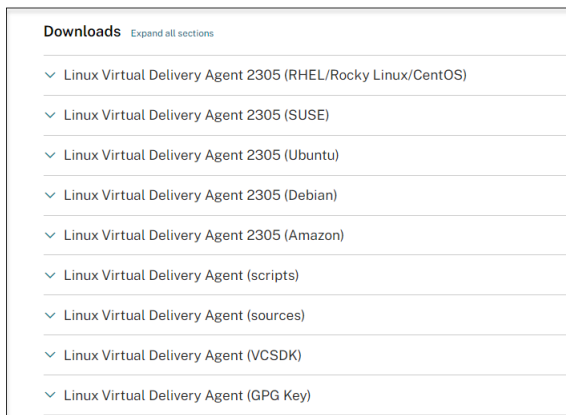
步骤 4：下载 Linux VDA 软件包

1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。

3. 展开组件以查找 Linux VDA。例如：

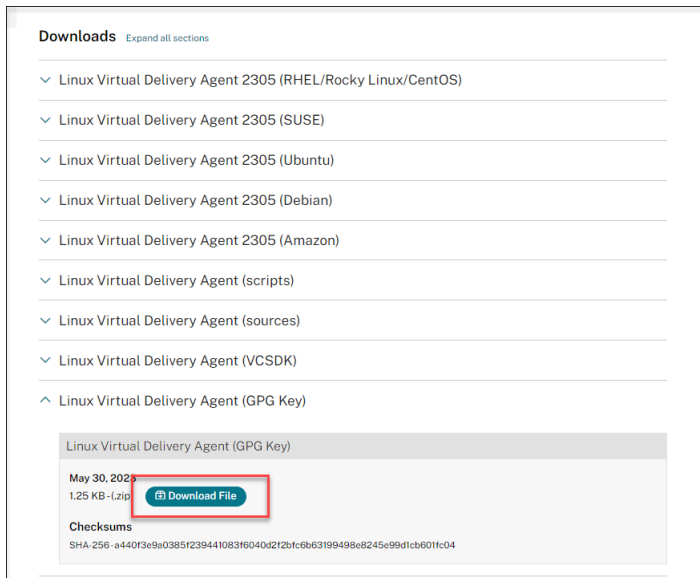


4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。

6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要使用公钥验证 Linux VDA 软件包的完整性，请执行以下操作：

- 对于 RPM 软件包，请运行以下命令以将公钥导入到 RPM 数据库中以及检查软件包的完整性：

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

- 对于 DEB 软件包，请运行以下命令以将公钥导入到 DEB 数据库中以及检查软件包的完整性：

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

步骤 5：安装 Linux VDA 软件包

要为 Linux VDA 设置环境，请运行以下命令。

对于 Amazon Linux 2、CentOS、RHEL 和 Rocky Linux 发行版：

注意：

- 对于 RHEL 和 CentOS，请先安装 EPEL 存储库，然后才能成功安装 Linux VDA。有关如何安装 EPEL 的信息，请参阅 <https://docs.fedoraproject.org/en-US/epel/> 上提供的说明。
- 在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上安装 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

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

注意：

在 GCP 上托管的 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 上安装 Linux VDA 后，以太网连接可能会断开，VM 重新启动后可能无法访问 Linux VDA。要解决此问题，请在首次登录 VM 时设置 root 用户密码，并确保您能够以 root 用户身份登录 VM。然后，在重新启动 VM 后，在控制台中运行以下命令：

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

对于 Ubuntu/Debian 发行版：

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

注意：

- 要为 Debian 11 发行版安装必要的依赖项，请将 `deb http://deb.debian.org/debian/bullseye main` 行添加到 `/etc/apt/sources.list` 文件。
- 对于 GCP 上的 Ubuntu 20.04，请禁用 RDNS。为此，请在 `/etc/krb5.conf` 的 `[libdefaults]` 下添加 `rdns = false` 行。

对于 SUSE 发行版：

1. 对于 AWS、Azure 和 GCP 上的 SUSE 15.5，请确保：
 - 您使用的是 **libstdc++6** 版本 12 或更高版本。
 - `/etc/sysconfig/windowmanager` 中的 **Default_WM** 参数设置为 “**gnome**”。
2. 请运行以下命令来安装 Linux VDA：

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

步骤 6：安装 NVIDIA GRID 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器（主机驱动程序），请参阅以下指南：

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

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序，请执行下面的常规步骤：

1. 确保来宾 VM 已关闭。
2. 在虚拟机管理程序控制面板中，为 VM 分配一个 GPU。

3. 启动 VM。
4. 在 VM 上安装来宾 VM 驱动程序。

步骤 7: 指定要使用的数据库

安装 Linux VDA 软件包后,您可以在 SQLite 与 PostgreSQL 之间切换。为此,请完成以下步骤:

注意:

- 我们建议您仅将 SQLite 用于 VDI 模式,将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS,您可以指定使用 SQLite 或 PostgreSQL,而不必手动安装。除非通过 `/etc/xdl/db.conf` 另行指定,否则 Linux VDA 默认使用 PostgreSQL。
- 也可以使用 `/etc/xdl/db.conf` 配置 PostgreSQL 的端口号。

1. 编辑 `/etc/xdl/db.conf` 以指定要使用的数据库。
2. 运行 `sudo /opt/Citrix/VDA/sbin/ctxinstall.sh` 或 `/opt/Citrix/VDA/bin/easyinstall`。

步骤 8: 运行轻松安装来配置环境和 VDA 以完成安装

安装 Linux VDA 软件包后,使用 `ctxinstall.sh` 脚本或 GUI 来配置运行环境。

注意:

在设置运行时环境之前,请确保已在您的操作系统中安装了 `en_US.UTF-8` 区域设置。如果该区域设置在您的操作系统中不可用,请运行 `sudo locale-gen en_US.UTF-8` 命令。对于 Debian,通过取消注释 `# en_US.UTF-8 UTF-8` 行来编辑 `/etc/locale.gen` 文件,然后运行 `sudo locale-gen` 命令。

ctxinstall.sh

`ctxinstall.sh` 为轻松安装脚本,用于进行一些预配置和设置 VDA 运行环境变量。

- 只有 root 用户可以运行此脚本。
- 轻松安装使用 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 作为其配置文件来设置、保存和同步使用的所有环境变量的值。我们建议您仔细阅读模板 (`ctxinstall.conf.tmpl`),然后自定义自己的 `ctxinstall.conf`。首次创建配置文件时,请使用以下任一方法:
 - 通过复制 `/opt/Citrix/VDA/sbin/ctxinstall.conf.tmpl` 模板文件并将其另存为 `/opt/Citrix/VDA/sbin/ctxinstall.conf`。
 - 通过运行 `ctxinstall.sh`。每次运行 `ctxinstall.sh` 时,您的输入都会保存在 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 中。
- 轻松安装功能支持模块化运行。模块包括预检查、安装、域配置、设置和验证。

- 可以在 `/var/log/xdl/ctxinstall.log` 中找到此脚本的调试详细信息。

有关详细信息，请使用帮助命令 **ctxinstall.sh -h**。

注意：

- 遵循最小权限原则，确保只有 root 用户才能读取 `/opt/Citrix/VDA/sbin/ctxinstall.conf`，因为文件中可能会设置域加入密码。
- 卸载 Linux VDA 会删除 `/opt/Citrix/VDA` 下的文件。我们建议您在卸载 VDA 之前备份 `/opt/Citrix/VDA/sbin/ctxinstall.conf`。

可以在交互模式或无提示模式下运行 `ctxinstall.sh`。在运行该脚本之前，请设置以下环境变量：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于已加入域的场景，请将其设置为 'n'。
- **CTX_XDL_AD_INTEGRATION=' sssd|winbind|centrify|pbis|quest'** - Linux VDA 要求使用 Kerberos 配置设置向 Delivery Controller 进行身份验证。Kerberos 配置根据系统上已安装和已配置的 Active Directory 集成工具确定。
- **CTX_XDL_DDC_LIST=' <list-ddc-fqdns>'** - Linux VDA 要求提供由空格分隔的 Delivery Controller 完全限定域名 (FQDN) 列表以用于向 Delivery Controller 进行注册。必须至少指定一个 FQDN 或 CNAME。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境，将该值设置为 'y'。
- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro，这是一组 GPU 加速技术，旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro，则要为 VDI 桌面（单会话）模式配置 VDA（即 `CTX_XDL_VDI_MODE=' y'`）。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口（默认端口 80 和 1494）。
- **CTX_XDL_DESKTOP_ENVIRONMENT='gnome/gnome-classic/kde/mate/xfce/' <none> '-** 指定要在会话中使用的 GNOME、GNOME Classic、KDE、MATE 或 **Xfce** 桌面环境。如果将其设置为 '**<none>**'，则将使用在 VDA 上配置的默认桌面。
- **CTX_XDL_DOTNET_RUNTIME_PATH='path-to-install-dotnet-runtime>'** - 安装 .NET 以支持新的 Broker 代理服务 (`ctxvda`) 的路径。默认路径为 '`/usr/bin`'。
- **CTX_XDL_VDA_PORT='port-number'** - Linux VDA 将通过 TCP/IP 端口与 Delivery Controller 通信。
- **CTX_XDL_SITE_NAME='<dns-name>'** - Linux VDA 通过 DNS 发现 LDAP 服务器。要将 DNS 搜索结果限制为本地站点，应指定 DNS 站点名称。如果不需要，请将其设置为 '**<none>**'。

- **CTX_XDL_LDAP_LIST=' <list-ldap-servers>'** - Linux VDA 查询 DNS 来发现 LDAP 服务器。如果 DNS 无法提供 LDAP 服务记录，您可以提供以空格分隔的 LDAP FQDN（带有 LDAP 端口）列表。例如，ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。要在 Active Directory 林中启用更快的 LDAP 查询，请在域控制器上启用全局目录，并将相关的 LDAP 端口号指定为 3268。默认情况下，此变量设置为 '**<none>**'。
- **CTX_XDL_SEARCH_BASE=search-base-set** - Linux VDA 通过设置为 Active Directory 域根的搜索库来查询 LDAP（例如，DC=mycompany,DC=com）。为提高搜索性能，可以指定搜索基础（例如 OU=VDI,DC=mycompany,DC=com）。如果不需要，请将其设置为 '**<none>**'。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME=' y|n'** - Linux VDA 支持使用 DNS CNAME 记录指定 Delivery Controller 名称。
- **CTX_EASYINSTALL_DNS=' <ip-address-of-dns>'** - DNS 的 IP 地址。
- **CTX_EASYINSTALL_HOSTNAME=host-name** - Linux VDA 服务器的主机名。
- **CTX_EASYINSTALL_NTPS=address-of-ntp** - NTP 服务器的 IP 地址或字符串名称。
- **CTX_EASYINSTALL_REALM=realm-name** - Kerberos 领域名称。
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_USERNAME=domain-user-name** - 将计算机加入域的用户名称。
- **CTX_EASYINSTALL_PASSWORD=password** - 将计算机加入域的用户密码。

注意：

我们建议您不要使用管理员帐户加入域。相反，请将加入域的权限委派给除管理员帐户之外的 Active Directory 用户。为此，请使用委派控制向导在域控制器上委托控制权。

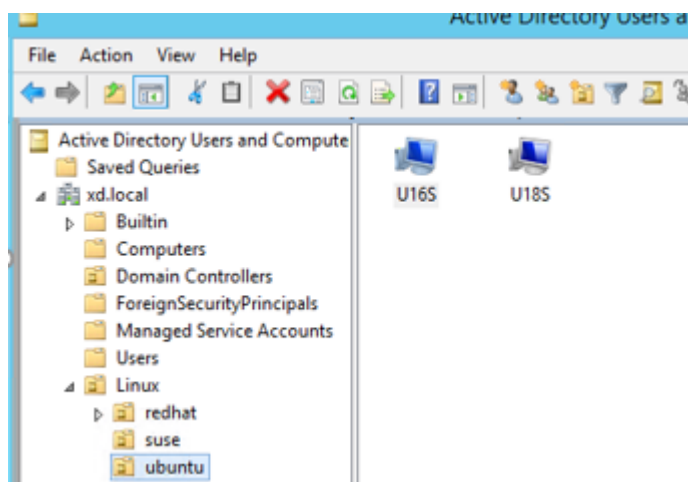
以下四个变量是可选变量。即使未设置，ctxinstall.sh 也不会无提示模式下中止，也不会提示您在交互模式下输入用户。您只能通过导出其值或编辑 **/Citrix/VDA/sbin/ctxinstall.conf** 对其进行设置。

- **CTX_EASYINSTALL_NETBIOS_DOMAIN=netbios-domain-name** - NetBIOS 域名通常是由点 (.) 分隔的 DNS 域名的第一个组成部分。否则，请自定义不同的 NetBIOS 域名。此变量是可选的。
- **CTX_EASYINSTALL_OU=ou-value** - OU 值因不同的 AD 集成方法而异。有关 OU 值的示例，请参阅本文注意事项部分中的表格。此变量是可选的。
- **CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=centrify-local-path** - 轻松安装有助于从 Internet 上下载 Centrify 软件包。但是，如果已经安装了 Centrify，则可以从此变量定义的本地目录中获取 Centrify 软件包。此变量是可选的。
- **CTX_EASYINSTALL_PBSIS_LOCAL_PATH=pbis-local-path** - 轻松安装有助于从 Internet 上下载 PBSIS 软件包。但是，如果已经安装了 PBSIS，则可以从此变量定义的本地目录中获取 PBSIS 软件包。此变量是可选的。

注意事项

- NetBIOS 域名通常是由点 (.) 分隔的 DNS 域名的第一个组成部分。要在您的环境中自定义不同的 NetBIOS 域名,请在 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 中设置环境变量 `CTX_EASYINSTALL_NETBIOS_DOMAIN`。
- 要将 VDA 连接到特定 OU, 请执行以下操作:
 1. 确保域控制器上存在特定的 OU。

有关示例 OU, 请参阅下面的屏幕截图:



2. 在 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 中设置环境变量 `CTX_EASYINSTALL_OU`。

OU 值因 AD 方法的不同而异。下表反映了前面屏幕截图中的示例 OU 名称。可以在贵组织中使用任何其他 OU 名称。

操作系统	Winbind	SSSD	Centrify	PBIS
Amazon Linux 2	"Linux/ amazon"	"Linux/ amazon"	"XD.LOCAL/ Linux/amazon "	"Linux/ amazon"
Debian	"Linux/ debian"	"Linux/ debian"	"XD.LOCAL/ Linux/debian "	"Linux/ debian"
RHEL 9.4/9.3/9.2、 Rocky Linux 9.4/9.3/9.2	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	不适用	不适用
RHEL 8.x	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	"XD.LOCAL/ Linux/redhat "	"Linux/ redhat"

操作系统	Winbind	SSSD	Centrify	PBIS
Rocky Linux 8.x	"OU=redhat, OU=Linux"	"OU=redhat, OU=Linux"	不适用	不适用
SUSE	"Linux/suse"	"Linux/suse"	"XD.LOCAL/ Linux/suse"	"Linux/suse"
Ubuntu	"Linux/ ubuntu"	"Linux/ ubuntu"	"XD.LOCAL/ Linux/ubuntu "	"Linux/ ubuntu"

- Centrify 不支持纯 **IPv6** DNS 配置。/etc/resolv.conf 中至少需要有一个使用 **IPv4** 的 DNS 服务器，**adclient** 才能正确查找 AD 服务。

日志：

```

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

```

此问题是 Centrify 及其配置独有的。要解决此问题，请执行以下操作：

- 在域控制器上打开管理工具。
 - 选择 **Active Directory** 站点和服务。
 - 为子网添加正确的子网地址。
- 自 Linux VDA 7.16 起，轻松安装支持纯 **IPv6**。以下先决条件和限制适用：
 - 必须配置您的 Linux 存储库，以确保您的计算机可以通过纯 **IPv6** 网络下载所需软件包。
 - 在纯 **IPv6** 网络中不支持 Centrify。

注意：

如果您的网络是纯 **IPv6**，并且所有输入均采用恰当的 **IPv6** 格式，VDA 将通过 **IPv6** 向 Delivery Controller 注册。如果您的网络具有混合 **IPv4** 和 **IPv6** 配置，第一个 DNS IP 地址的类型将决定是使用 **IPv4** 还是使用 **IPv6** 来注册。

- 如前所述，您可以使用 CTX_XDL_DESKTOP_ENVIRONMENT 变量来指定要在会话中使用的桌面环境。您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息，请参阅[桌面切换命令](#)和[系统托盘](#)。
- 如果选择 Centrify 作为加入域的方法，ctxinstall.sh 脚本需要 Centrify 软件包。ctxinstall.sh 获取 Centrify 软件包的方法如下：
 - 轻松安装可帮助自动从 Internet 下载 Centrify 软件包。下面是每个发行版的 URL：

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

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

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

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

- 如果已安装 Centrify，则从本地目录中获取 Centrify 软件包。要指定 Centrify 软件包的目录，请在 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 中设置 `CTX_EASYINSTALL_CENTRIFY_LOCAL_PATH=/home/mydir` 例如：

```

1  ls -ls /home/mydir
2      9548 -r-xr-xr-x. 1 root root  9776688 May 13  2016
      adcheck-rhel4-x86_64
3      4140 -r--r--r--. 1 root root  4236714 Apr 21  2016
      centrifyda-3.3.1-rhel4-x86_64.rpm
4      33492 -r--r--r--. 1 root root 34292673 May 13  2016
      centrifydc-5.3.1-rhel4-x86_64.rpm
5      4 -rw-rw-r--. 1 root root    1168 Dec  1  2015
      centrifydc-install.cfg
6      756 -r--r--r--. 1 root root    770991 May 13  2016
      centrifydc-ldaproxy-5.3.1-rhel4-x86_64.rpm
7      268 -r--r--r--. 1 root root    271296 May 13  2016
      centrifydc-nis-5.3.1-rhel4-x86_64.rpm
8      1888 -r--r--r--. 1 root root 1930084 Apr 12  2016
      centrifydc-openssh-7.2p2-5.3.1-rhel4-x86_64.rpm
9      124 -rw-rw-r--. 1 root root   124543 Apr 19  2016
      centrify-suite.cfg
10     0 lrwxrwxrwx. 1 root root         10 Jul  9  2012 install-
      express.sh -> install.sh
11     332 -r-xr-xr--. 1 root root   338292 Apr 10  2016 install
      .sh
12     12 -r--r--r--. 1 root root    11166 Apr  9  2015 release-
      notes-agent-rhel4-x86_64.txt
13     4 -r--r--r--. 1 root root     3732 Aug 24  2015 release-
      notes-da-rhel4-x86_64.txt
14     4 -r--r--r--. 1 root root     2749 Apr  7  2015 release-
      notes-nis-rhel4-x86_64.txt
15     12 -r--r--r--. 1 root root     9133 Mar 21  2016 release-
      notes-openssh-rhel4-x86_64.txt

```

- 如果选择 PBIS 作为加入域的方法，`ctxinstall.sh` 脚本需要 PBIS 软件包。`ctxinstall.sh` 获取 PBIS 软件包的方法：
 - 轻松安装可帮助自动从 Internet 下载 PBIS 软件包。例如，下面是每个发行版的 URL：

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

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

- 从 Internet 获取 PBIS 软件包的特定版本。为此，请更改 `/opt/Citrix/VDA/sbin/ctxinstall.sh` 文件中的 `pbisDownloadRelease` 和 `pbisDownloadExpectedSHA256` 行。

有关示例，请参阅以下屏幕截图：

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

- 如果已经安装了 PBIS，则从本地目录获取 PBIS 软件包。要指定 PBIS 软件包的目录，请在 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 中设置环境变量 `CTX_EASYINSTALL_PBIS_LOCAL_PATH`。

交互式模式 要在交互模式下运行 `ctxinstall.sh` 脚本，请使用不带 `-S` 选项的 `sudo /opt/Citrix/VDA/sbin/ctxinstall.sh` 命令。在命令行界面的每个提示符处键入相关的变量值。如果已经设置了变量，`ctxinstall.sh` 会要求您确认，以防您想要对其进行更改。

无提示模式 在无提示模式下，必须使用 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 或导出命令设置前面的变量。之后，请运行 `ctxinstall.sh -S`（请注意，此处字母 **S** 为大写字母）。如果未设置所有必需变量或者某些值无效，则除非有默认值，否则 `ctxinstall.sh` 将中止执行。

除非未设置，否则每个变量的导出值都会覆盖 `/Citrix/VDA/sbin/ctxinstall.conf` 中的值。除域加入密码外，所有更新后的值都保存在 `/Citrix/VDA/sbin/ctxinstall.conf` 中。因此，在无提示模式下，您必须在 `/Citrix/VDA/sbin/ctxinstall.conf` 中设置域加入密码或者导出密码。

```
1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT=gnome|gnome-classic|kde|mate|xfce|'<none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'
11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>'|'<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>'|'<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>'|'<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 export CTX_EASYINSTALL_DNS='<ip-address-of-dns>'
17 export CTX_EASYINSTALL_HOSTNAME='<host-name>'
18 export CTX_EASYINSTALL_NTFS='<address-of-ntfs>'
19 export CTX_EASYINSTALL_REALM='<realm-name>'
```

```

20 export CTX_EASYINSTALL_FQDN='<ad-fqdn-name>'
21 export CTX_EASYINSTALL_USERNAME='<domain-user-name>'
22 export CTX_EASYINSTALL_PASSWORD='<password>'
23 export CTX_EASYINSTALL_NETBIOS_DOMAIN='<netbios-domain>'
24 export CTX_EASYINSTALL_OU='<organization-unit>'
25 sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh -S

```

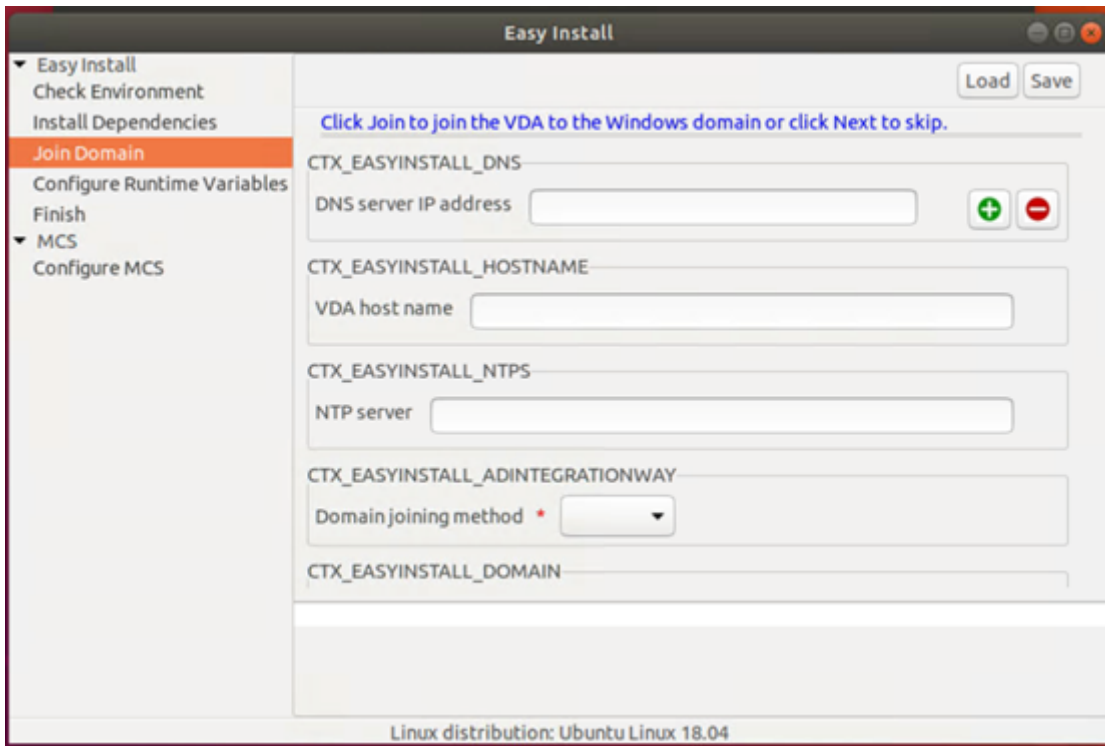
运行 `sudo` 命令时，键入 **-E** 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 **#!/bin/bash** 作为第一行来创建 shell 脚本文件。

或者，您可以使用单个命令指定所有变量。

要设置 VDA 运行环境变量（以 **CTX_XDL_** 开头的变量），您可以运行 **ctxinstall.sh -s**（请注意，此处字母 **s** 为小写字母）。

GUI

可以通过 GUI 进行轻松安装。在 VDA 的桌面环境中运行 `/opt/Citrix/VDA/bin/easyinstall` 命令，然后按照轻松安装 GUI 上的说明进行操作。



轻松安装 GUI 将引导您完成以下操作：

- 检查系统环境
- 安装依赖项
- 将 VDA 加入指定域
- 配置运行时环境

提示:

单击保存将变量设置保存到指定路径下的本地文件中。单击加载从您指定的文件加载变量设置。有关配置 MCS 变量的信息，请参阅[步骤 3: 准备主映像](#)。

步骤 9: 运行 XDPing

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅[XDPing](#)。

步骤 10: 运行 Linux VDA

启动 Linux VDA:

启动 Linux VDA 服务:

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

停止 Linux VDA:

要停止 Linux VDA 服务，请执行以下操作:

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

注意:

在停止 `ctxvda` 和 `ctxhdx` 服务之前，请运行 `systemctl stop ctxmonitord` 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

重新启动 Linux VDA:

重新启动 Linux VDA 服务:

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

检查 Linux VDA 的状态:

要检查 Linux VDA 服务的运行状态，请执行以下操作:

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

步骤 11: 创建计算机目录

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明, 请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制, 使得该过程不同于为 Windows VDA 计算机创建计算机目录:

- 对于操作系统, 请选择:
 - 多会话操作系统选项 (对于托管共享桌面交付模型)。
 - 单会话操作系统选项 (适用于 VDI 专用桌面交付模型)。
- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。

注意:

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是, 选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

提示:

如果删除计算机后将其重新加入 Active Directory 域, 则必须删除计算机, 然后将其重新添加到计算机目录。

步骤 12: 创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明, 请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制:

- 确保所选 AD 用户和组已正确配置, 能够登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的 (匿名) 用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

重要:

Linux VDA 1.4 及更高版本支持发布应用程序。但是, Linux VDA 不支持将桌面和应用程序交付给相同的计算机。

有关如何创建计算机目录和交付组的信息, 请参阅 [Citrix Virtual Apps and Desktops 7 2407](#)。

步骤 13: 升级 Linux VDA (可选)

Linux VDA 支持从最新版本进行升级。例如,您可以将 Linux VDA 从 2308 升级到 2311,从 1912 LTSR 升级到 2203 LTSR。

注意:

- 升级现有安装将覆盖 /etc/xdm 下的配置文件。在进行升级之前,请务必备份这些文件。
- 从 2407 版本开始, Linux VDA 委派包管理器 **rpm** 或 **dpkg** 在升级期间处理配置文件。下面介绍了 rpm 和 dpkg 如何与配置文件更改交互:
 - **rpm**: by default keeps the local version and saves the new version from the package with a .rpmnew extension.
 - **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as **.dpkg-new** or **.dpkg-dist**, use the following command:

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

对于 **Amazon Linux 2**、**RHEL** 和 **Rocky Linux** 发行版:

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

注意:

在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上升级 Linux VDA 之前,请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

对于 **SUSE** 发行版:

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

对于 **Ubuntu/Debian** 发行版:

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

故障排除

请利用此部分中的信息对可能因使用轻松安装功能而引发的问题进行故障排除。

使用 **SSSD** 加入域失败

尝试加入域时可能会出现错误，输出类似如下（要进行屏幕打印，请验证日志）：

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

/var/log/xdl/vda.log:

```
1 2016-11-04 02:11:52.317 [INFO ] - The Citrix Desktop Service
  successfully obtained the following list of 1 delivery controller(s)
  with which to register: 'CTXDDC.citrixlab.local (10.158.139.214)'.
2 2016-11-04 02:11:52.362 [ERROR] - RegistrationManager.
  AttemptRegistrationWithSingleDdc: Failed to register with http://
  CTXDDC.citrixlab.local:80/Citrix/CdsController/IRegistrar. Error:
  General security error (An error occurred in trying to obtain a TGT:
  Client not found in Kerberos database (6))
3 2016-11-04 02:11:52.362 [ERROR] - The Citrix Desktop Service cannot
  connect to the delivery controller 'http://CTXDDC.citrixlab.local
  :80/Citrix/CdsController/IRegistrar' (IP Address '10.158.139.214')
4 Check the following:- The system clock is in sync between this machine
  and the delivery controller.
5 - The Active Directory provider (e.g. winbind daemon) service is
  running and correctly configured.
6 - Kerberos is correctly configured on this machine.
7 If the problem persists, please refer to Citrix Knowledge Base article
  CTX117248 for further information.
8 Error Details:
9 Exception 'General security error (An error occurred in trying to
  obtain a TGT: Client not found in Kerberos database (6))' of type '
  class javax.xml.ws.soap.SOAPFaultException'.
10 2016-11-04 02:11:52.362 [INFO ] - RegistrationManager.
  AttemptRegistrationWithSingleDdc: The current time for this VDA is
  Fri Nov 04 02:11:52 EDT 2016.
11 Ensure that the system clock is in sync between this machine and the
  delivery controller.
12 Verify the NTP daemon is running on this machine and is correctly
  configured.
13 2016-11-04 02:11:52.364 [ERROR] - Could not register with any
  controllers. Waiting to try again in 120000 ms. Multi-forest - false
14 2016-11-04 02:11:52.365 [INFO ] - The Citrix Desktop Service failed to
  register with any controllers in the last 470 minutes.
```

/var/log/messages:

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

in Kerberos database

要解决此问题，请执行以下操作：

1. 运行 `rm -f /etc/krb5.keytab` 命令。
2. 运行 `net ads leave $REALM -U $domain-administrator` 命令。
3. 在 Delivery Controller 上删除计算机目录和交付组。
4. 运行 `/opt/Citrix/VDA/sbin/ctxinstall.sh`。
5. 在 Delivery Controller 上创建计算机目录和交付组。

Ubuntu 桌面会话显示灰屏

启动会话时会出现此问题，随后将在空桌面中阻止启动会话功能。此外，使用本地用户帐户登录时，计算机的控制台也显示灰屏。

要解决此问题，请执行以下操作：

1. 运行 `sudo apt-get update` 命令。
2. 运行 `sudo apt-get install unity lightdm` 命令。
3. 向 `/etc/lightdm/lightdm.conf` 中添加以下行：
`greeter-show-manual-login=true`

由于缺少主目录，尝试启动 **Ubuntu** 桌面会话失败

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

```
1 2016-11-02 13:21:19.015 <P22492:S1> citrix-ctxlogin: StartUserSession:
   failed to change to directory(/home/CITRIXLAB/ctxadmin) errno(2)
2
3 2016-11-02 13:21:19.017 <P22227> citrix-ctxhdx: logSessionEvent:
   Session started for user ctxadmin.
4
5 2016-11-02 13:21:19.023 <P22492:S1> citrix-ctxlogin: ChildPipeCallback:
   Login Process died: normal.
6
7 2016-11-02 13:21:59.217 <P22449:S1> citrix-ctxgfx: main: Exiting
   normally.
```

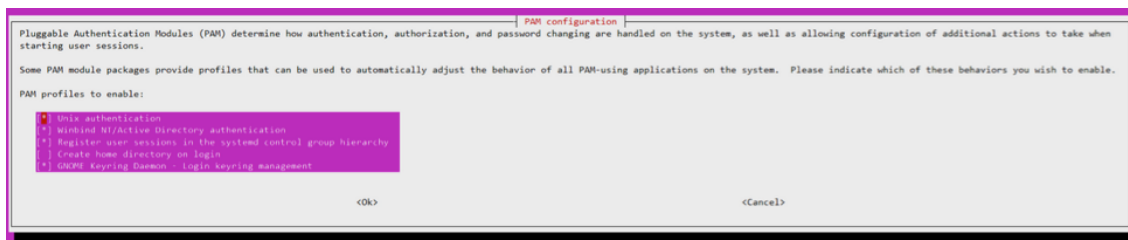
提示：

此问题的根本原因是没有为域管理员创建主目录。

要解决此问题，请执行以下操作：

1. 在命令行中，键入 **pam-auth-update**。

2. 在生成的对话框中，确认是否已选中 **Create home directory login**（创建主目录登录信息）。



会话不启动，也不快速结束并显示 **dbus** 错误

/var/log/messages（适用于 RHEL 或 CentOS）：

```

1 Oct 27 04:17:16 CentOS7 citrix-ctxhdx[8978]: Session started for user
  CITRIXLAB\ctxadmin.
2
3 Oct 27 04:17:18 CentOS7 kernel: traps: gnome-session[19146] trap int3
  ip:7f89b3bde8d3 sp:7fff8c3409d0 error:0
4
5 Oct 27 04:17:18 CentOS7 gnome-session[19146]: ERROR: Failed to connect
  to system bus: Exhausted all available authentication mechanisms (
  tried: EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS) (available: EXTERNAL,
  DBUS_COOKIE_SHA1, ANONYMOUS)#012aborting...
6
7 Oct 27 04:17:18 CentOS7 gnome-session: gnome-session[19146]: ERROR:
  Failed to connect to system bus: Exhausted all available
  authentication mechanisms (tried: EXTERNAL, DBUS_COOKIE_SHA1,
  ANONYMOUS) (available: EXTERNAL, DBUS_COOKIE_SHA1, ANONYMOUS)
8
9 Oct 27 04:17:18 CentOS7 gnome-session: aborting...
10
11 Oct 27 04:17:18 CentOS7 citrix-ctxgfx[18981]: Exiting normally.
12
13 Oct 27 04:17:18 CentOS7 citrix-ctxhdx[8978]: Session stopped for user
  CITRIXLAB\ctxadmin.

```

或者，对于 Ubuntu 发行版，请使用日志 /var/log/syslog：

```

1 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] pid.c:
  Stale PID file, overwriting.
2
3 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] bluez5-
  util.c: Failed to get D-Bus connection: Did not receive a reply.
  Possible causes include: the remote application did not send a reply
  , the message bus security policy blocked the reply, the reply
  timeout expired, or the network connection was broken.
4
5 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25326]: [pulseaudio] hashmap
  .c: Assertion 'h' failed at pulsecore/hashmap.c:116, function
  pa_hashmap_free(). Aborting.
6

```

```

7 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: [pulseaudio] core-
  util.c: Failed to connect to system bus: Did not receive a reply.
  Possible causes include: the remote application did not send a reply
  , the message bus security policy blocked the reply, the reply
  timeout expired, or the network connection was broken.
8
9 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: message repeated 10
  times: [ [pulseaudio] core-util.c: Failed to connect to system bus:
  Did not receive a reply. Possible causes include: the remote
  application did not send a reply, the message bus security policy
  blocked the reply, the reply timeout expired, or the network
  connection was broken.]
10
11 Nov  3 11:03:52 user01-HVM-domU pulseaudio[25352]: [pulseaudio] pid.c:
  Daemon already running.Nov  3 11:03:58 user01-HVM-domU citrix-ctxgfx
  [24693]: Exiting normally

```

某些组或模块在重新启动后才会生效。如果日志中出现 **dbus** 错误消息，我们建议您重新启动系统并重试。

SELinux 可以防止 **SSHD** 访问主目录

用户可以启动会话，但不能登录。

/var/log/xdl/ctxinstall.log:

```

1 Jan 25 23:30:31 yz-rhel72-1 setroubleshoot[3945]: SELinux is preventing
  /usr/sbin/sshd from setattr access on the directory /root. For
  complete SELinux messages. run sealert -l 32f52c1f-8ff9-4566-a698
  -963a79f16b81
2
3 Jan 25 23:30:31 yz-rhel72-1 python[3945]: SELinux is preventing /usr/
  sbin/sshd from setattr access on the directory /root.
4
5 ***** Plugin catchall_boolean (89.3 confidence) suggests
  *****
6
7 If you want to allow polyinstantiation to enabled
8
9 Then you must tell SELinux about this by enabling the '
  polyinstantiation_enabled' boolean.
10
11 You can read 'None' man page for more details.
12
13 Do
14
15     setsebool -P polyinstantiation_enabled 1
16
17 ***** Plugin catchall (11.6 confidence) suggests
  *****
18
19 If you believe that sshd should be allowed setattr access on the root
  directory by default.

```

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

要解决此问题，请执行以下操作：

1. 通过对 /etc/selinux/config 进行以下更改来禁用 SELinux。

```
SELINUX=disabled
```

2. 重新启动 VDA。

使用 **MCS** 创建未加入域的 **Linux VDA**

August 20, 2024

本文将引导您使用 Machine Creation Services (MCS) 在 Citrix DaaS 中创建未加入域的 Linux VDA。

重要：

- 只有 Citrix DaaS 支持使用 MCS 创建未加入域的 VDA。对于 Citrix DaaS 客户：
 - 您的控制平面必须在 Citrix DaaS 上部署。
 - 可以在公有云或本地数据中心部署未加入域的 VDA。未加入域的 VDA 由 Citrix DaaS 中的控制平面进行管理。
 - 要创建未加入域的 VDA，使用 Citrix Gateway 服务的客户必须确保启用 [Rendezvous V2](#)。仅当您计划在本地虚拟机管理程序上预配计算机或者想要在 Workspace 中使用 Active Directory 作为身份提供程序时，才需要使用 Cloud Connector。
- 要创建未加入域的 VDA，您可以同时使用 MCS 和轻松安装。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)和[使用轻松安装创建未加入域的 Linux VDA（预览版）](#)。
- MCS 不支持裸机服务器。
- 以下功能适用于未加入域的 Linux VDA：
 - [在未加入域的 VDA 上创建具有指定属性的本地用户](#)
 - [非 SSO 身份验证](#)
 - [使用 Azure Active Directory 进行身份验证](#)

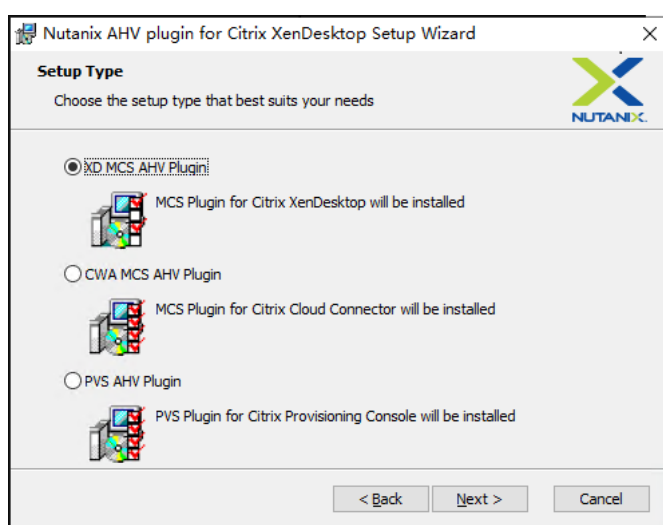
- Rendezvous V2

(仅限 **Nutanix**) 步骤 **1**: 安装并注册 **Nutanix AHV** 插件

从 Nutanix 中获取 Nutanix AHV 插件包。在您的 Citrix Virtual Apps and Desktops 环境中安装并注册该插件。有关详细信息，请参阅 [Nutanix 支持门户](#) 中提供的《Nutanix Acropolis MCS Plugin Installation Guide》(Nutanix Acropolis MCS 插件安装指南)。

步骤 **1a**: 为本地 **Delivery Controller** 安装并注册 **Nutanix AHV** 插件

安装 Citrix Virtual Apps and Desktops 后，选择并在 Delivery Controller 上安装 **XD MCS AHV** 插件。



步骤 **1b**: 为云端 **Delivery Controller** 安装并注册 **Nutanix AHV** 插件

选择并安装适用于 Citrix Cloud Connector 的 **CWA MCS AHV** 插件。在向 Citrix Cloud 租户注册的所有 Citrix Cloud Connector 上安装插件。即使 Citrix Cloud Connector 在没有 AHV 的情况下提供资源位置，也必须注册 Citrix Cloud Connector。

步骤 **1c**: 安装插件后完成以下步骤

- 验证是否在 `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0` 中创建了 Nutanix Acropolis 文件夹。
- 运行 `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe"-PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\v1.0.0.0"` 命令。

- 在本地 Delivery Controller 上重新启动 Citrix Host Service、Citrix Broker Service 和 Citrix Machine Creation Service，或者在 Citrix Cloud Connector 上重新启动 Citrix RemoteHCLServer Service。

提示：

我们建议您在安装或更新 Nutanix AHV 插件时停止并重新启动 Citrix Host Service、Citrix Broker Service 和 Machine Creation Services。

步骤 2：创建主机连接

主机是在您的资源位置中使用的虚拟机管理程序或云服务。此步骤允许您指定 DaaS 用于与主机上的 VM 通信的信息。详细信息包括资源位置、主机类型、访问凭据、要使用的存储方法以及主机上的 VM 可以使用哪些网络。

重要：

创建连接之前，您的资源位置中的主机资源（存储和网络）必须可用。

1. 登录 Citrix Cloud。
2. 在左上角的菜单中，选择我的服务 > **DaaS**。
3. 在管理 > 完整配置中，选择左侧窗格中的托管。
4. 在操作栏中选择添加连接和资源。
5. 向导将引导您完成以下页面。特定页面内容取决于所选连接类型。完成每一页之后，请选择下一步，直到到达摘要页为止。

步骤 2a: 连接

Add Connection and Resources [X]

1 Connection
2 Region
3 Network
4 Scopes
5 Summary

Connection

Use an existing connection
BingTest

Create a new connection
Zone name: []

Connection type: [Google Cloud Platform]

Service account key: [Import key...]

Service account ID: []

Connection name: []

Create virtual machines using:
 Citrix provisioning tools (Machine Creation Services or Citrix Provisioning)
 Other tools

Next [Cancel 7 ↻]

在连接页面上:

- 要创建一个连接，请选择创建新连接。要基于与现有连接相同的主机配置创建一个连接，请选择使用现有连接，然后选择相关连接。
- 在区域名称字段中选择一个区域。选项是您配置的所有资源位置。
- 在连接类型字段中选择虚拟机管理程序或云服务。选项是已在区域中正确安装插件的虚拟机管理程序和云服务。或者，您可以使用 PowerShell 命令 `Get-HypervisorPlugin -ZoneUid` 获取所选区域中可用的虚拟机管理程序插件列表。
- 输入连接名称。此名称将显示在管理显示屏中。
- 选择用于创建虚拟机的工具：Machine Creation Services 或 Citrix Provisioning。

连接页面上的信息因您使用的主机（连接类型）而异。例如，使用 Azure Resource Manager 时，可以使用现有的服务主体或者创建一个主体。

步骤 2b: 管理存储

Add Connection and Resources [X]

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

Storage Management
Configure virtual machine storage resources for this connection.

Select a cluster:
[Text Input] [Browse]

Select an optimization method for available site storage.

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

[Back] [Next] [Cancel]

有关存储管理类型和方法的信息，请参阅[主机存储](#)。

如果要配置与 Hyper-V 或 VMware 主机的连接，请浏览到群集名称并选择一个名称。其他连接类型不需要群集名称。

选择存储管理方法：虚拟机管理程序共享的存储或虚拟机管理程序的本地存储。

- 如果选择虚拟机管理程序共享的存储，请指出是否要在可用的本地存储中保存临时数据。（可以在使用此连接的计算机目录中指定非默认临时存储大小。）例外：使用群集存储卷 (CSV) 时，Microsoft System Center Virtual Machine Manager 不允许在本地存储中创建的临时数据缓存磁盘。尝试在管理控制台中配置该存储管理设置失败。

如果使用 Citrix Hypervisor 池中的共享存储，请指出是否要使用 IntelliCache 来降低共享存储设备上的负载。请参阅 [Citrix Hypervisor 虚拟化环境](#)。

步骤 2c: 存储选择

Add Connection and Resources [X]

Connection
 Storage Management
 Storage Selection
 Network
 Summary

Storage Selection

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

Select data storage locations:

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

有关存储选择的详细信息，请参阅[主机存储](#)。

请至少为每种可用的数据类型选择一个主机存储设备。在上一页面中选择的存储管理方法将影响此页面上可供选择的数据类型。必须至少为每种受支持的数据类型选择一个存储设备，才能继续进入向导中的下一页面。

如果已选择由虚拟机管理程序共享的存储并启用了 **Optimize temporary data on available local storage**（优化可用本地存储中的临时数据），则选择存储页面的下半部分将包含更多配置选项。您可以选择要用于存储临时数据的本地存储设备（在同一虚拟机管理程序池中）。

系统会显示当前选择的存储设备数量（在图中，显示“已选择 1 个存储设备”）。将鼠标悬停在该条目上时，将显示所选设备的名称（除非未配置任何设备）。

1. 选择选择更改要使用的存储设备。
2. 在选择存储对话框中，选中或取消选中存储设备复选框，然后选择确定。

步骤 2d: 区域

（仅对某些主机类型显示。）区域选择指示 VM 的部署位置。实际上，请选择一个靠近用户在其中访问其应用程序的区域。

步骤 2e: 网络

输入资源的名称。此名称在管理控制台中显示，以表示与连接关联的存储和网络组合。

选择 VM 使用的一个或多个网络。

某些连接类型（例如 Azure Resource Manager）还列出 VM 要使用的子网。请选择一个或多个子网。

步骤 2f: 摘要

检查所做的选择；如果要做出更改，请使用返回到上一个向导页面。完成检查后，请选择完成。

谨记：如果在本地存储临时数据，则可以在创建包含使用此连接的计算机的目录时为临时数据配置非默认值。

注意：

不显示完全访问权限管理员的作用域。有关详细信息，请参阅[管理员、角色和作用域](#)。

有关详细信息，请参阅[创建和管理连接](#)。

步骤 3: 准备主映像

提示：

可以使用单个映像来创建已加入域和未加入域的 VDA。

【仅适用于 XenServer（以前称为 Citrix Hypervisor）】步骤 3a: 安装 XenServer VM Tools

在模板 VM 上安装 XenServer VM Tools，以便每个 VM 都使用 xe CLI 或 XenCenter。除非安装这些工具，否则 VM 的性能可能会很慢。如果没有这些工具，无法执行以下任何操作：

- 彻底关闭、重新启动或挂起 VM。
 - 在 XenCenter 中查看 VM 性能数据。
 - 迁移正在运行的 VM（通过 [XenMotion](#)）。
 - 创建快照或带有内存（检查点）的快照，以及还原到快照。
 - 在正在运行的 Linux VM 上调整 vCPU 数。
1. 根据正在使用的虚拟机管理程序版本，从 [XenServer 下载页面](#) 或 [Citrix Hypervisor 下载页面](#) 下载适用于 Linux 的 XenServer VM Tools 文件。
 2. 将 `LinuxGuestTools-xxx.tar.gz` 文件复制到 Linux VM 或 Linux VM 可以访问的共享驱动器。
 3. 解压 tar 文件的内容：`tar -xzf LinuxGuestTools-xxx.tar.gz`
 4. 根据您的 Linux 发行版，运行以下命令安装 `xe-guest-utilities` 软件包。

对于 **RHEL/CentOS/Rocky Linux/SUSE**：

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

对于 **Ubuntu/Debian**：

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

5. 在 XenCenter 中的常规选项卡上检查模板 VM 的虚拟化状态。如果正确安装了 XenServer VM Tools，则虚拟化状态显示已优化。

步骤 **3b**: 在模板 **VM** 上安装 **.NET** 和 **Linux VDA** 软件包

注意:

要使用当前正在运行的 VDA 作为模板 VM，请跳过此步骤。

在安装 Linux VDA 软件包之前，请在模板 VM 上安装 .NET 并注意以下事项:

- 除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。
- 如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装 .NET 后，根据您的 Linux 发行版运行以下命令以安装 Linux VDA:

对于 **RHEL/CentOS/Rocky Linux**:

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

注意:

在 GCP 上托管的 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 上安装 Linux VDA 后，以太网连接可能会断开，VM 重新启动后可能无法访问 Linux VDA。要解决此问题，请在重新启动 VM 之前运行以下命令:

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

对于 **Ubuntu/Debian**:

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

对于 **SUSE**:

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

步骤 **3c**: (仅限 **RHEL**) 安装可以提供 **ntfs-3g** 的 **EPEL** 存储库 在 RHEL 8 上安装 EPEL 存储库。有关如何安装 EPEL 的信息，请参阅 <https://docs.fedoraproject.org/en-US/epel/> 上提供的说明。

步骤 3e: (仅适用于 **SUSE**) 手动安装 **ntfs-3g**

在 SUSE 平台上, 所有存储库都不提供 **ntfs-3g**。下载源代码, 编译并手动安装 **ntfs-3g**:

1. 安装 GNU Compiler Collection (GCC) 编译器系统并将软件包设置为:

```
1 sudo zypper install gcc
2 sudo zypper install make
```

2. 下载 **ntfs-3g** 软件包。

3. 解压缩 **ntfs-3g** 软件包:

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

4. 输入 **ntfs-3g** 软件包的路径:

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

5. 安装 **ntfs-3g**:

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

步骤 3f: (仅适用于 **Ubuntu**) 编辑 **/etc/network/interfaces** 文件

将 `source /etc/network/interfaces.d/*` 行添加到 `/etc/network/interfaces` 文件中。

提示:

`/etc/network/interfaces` 文件可能在您的 Ubuntu 计算机上不提供。如果该文件不存在, 您需要先安装 `net-tools` 和 `ifupdown` 软件包。

步骤 3f: (仅适用于 **Ubuntu**) 指向 **/etc/resolv.conf**

请将 `/etc/resolv.conf` 指向 `/run/systemd/resolve/resolv.conf`, 而非将其指向 `/run/systemd/resolve/stub-resolv.conf`:

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

步骤 3h: 指定要使用的数据库

安装 Linux VDA 软件包后, 您可以在 SQLite 与 PostgreSQL 之间切换。为此, 请完成以下步骤:

注意：

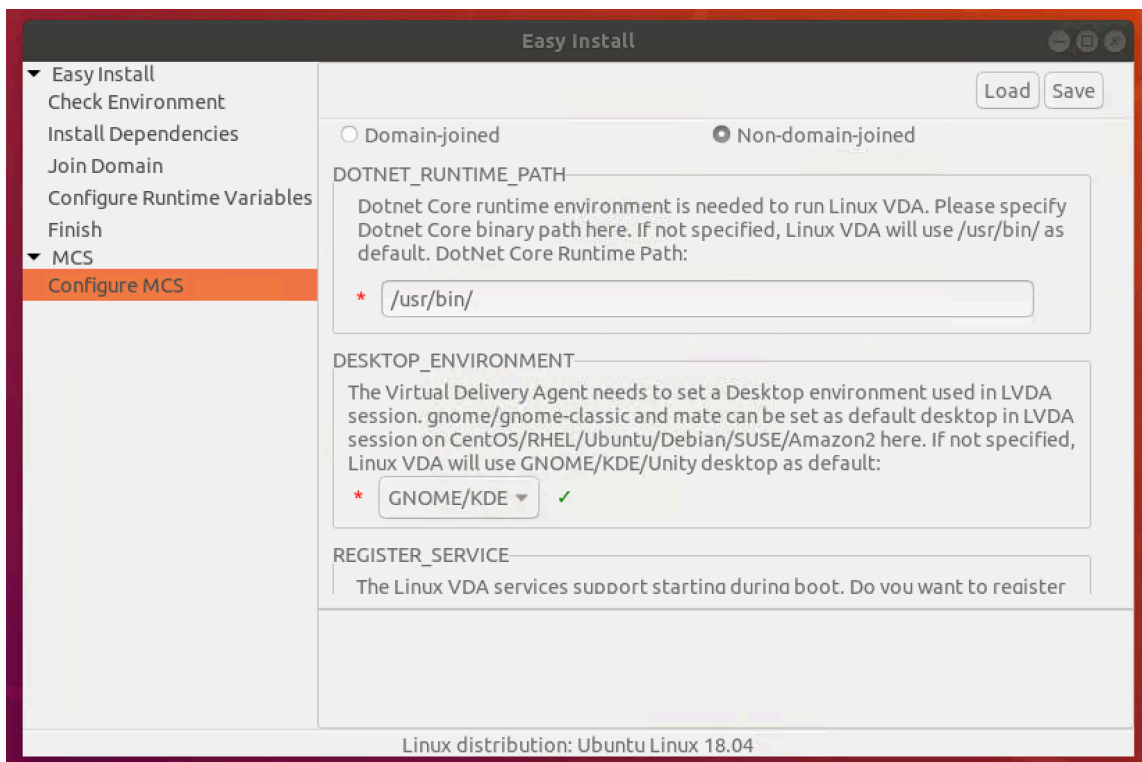
- 我们建议您仅将 SQLite 用于 VDI 模式，将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS，您可以指定使用 SQLite 或 PostgreSQL，而不必手动安装。除非通过 `/etc/xdl/db.conf` 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 也可以使用 `/etc/xdl/db.conf` 配置 PostgreSQL 的端口号。

1. 运行 `/opt/Citrix/VDA/sbin/ctxcleanup.sh`。如果是全新安装，则省略此步骤。
2. 请在运行 `deploymcs.sh` 前编辑 `/etc/xdl/db.conf`。

步骤 3i: 配置 MCS 变量

配置 MCS 变量有两种方法：

- 编辑 `/etc/xdl/mcs/mcs.conf` 文件。
- 使用轻松安装 GUI。要打开轻松安装 GUI，请在您的 Linux VDA 的桌面环境中运行 `/opt/Citrix/VDA/bin/easyinstall` 命令。



提示：

单击保存将变量设置保存到指定路径下的本地文件中。单击加载从您指定的文件加载变量设置。

下面是您可以为未加入域的场景配置的 MCS 变量。可以使用默认变量值或者根据需要自定义变量（可选）：

```
DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime
DESKTOP_ENVIRONMENT=gnome | mate
REGISTER_SERVICE=Y | N
ADD_FIREWALL_RULES=Y | N
VDI_MODE=Y | N
START_SERVICE=Y | N
```

(可选) 步骤 **3j**: 写入或更新 **MCS** 的注册表值

在模板计算机上, 将命令行添加到 `/etc/xdl/mcs/mcs_local_setting.reg` 文件以根据需要写入或更新注册表值。此操作可防止数据和设置在 MCS 预配的计算机每次重新启动时丢失。

`/etc/xdl/mcs/mcs_local_setting.reg` 文件中的每一行就是用于设置或更新注册表值的一个命令。

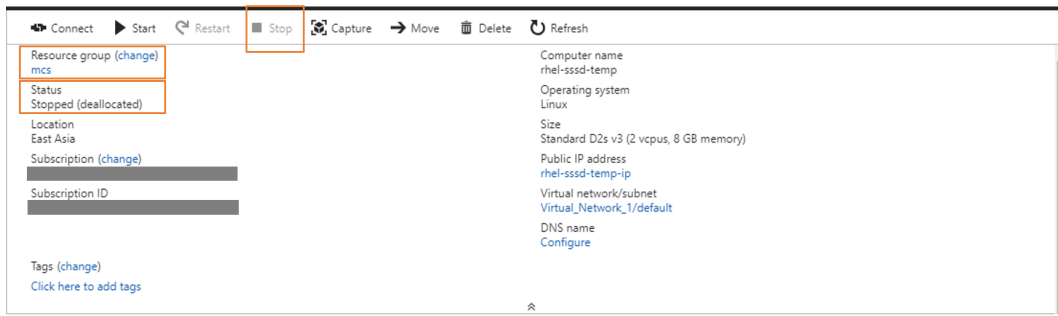
例如, 您可以将以下命令行添加到 `/etc/xdl/mcs/mcs_local_setting.reg` 文件中, 以分别写入或更新注册表值:

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

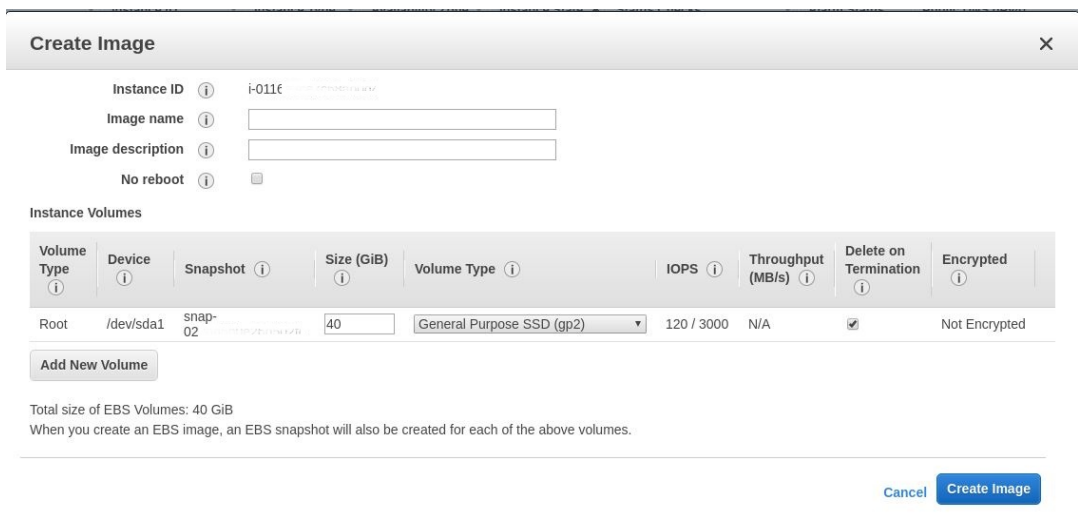
```
1 update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels
  \Clipboard\ClipboardSelection" -v "Flags" -d "0x00000003"
```

步骤 **3k**: 创建主映像

1. 如果您通过编辑 `/etc/xdl/mcs/mcs.conf` 配置 MCS 变量, 请运行 `/opt/Citrix/VDA/sbin/deploymcs.sh`。如果您使用 GUI 配置 MCS 变量, 请单击部署。在 GUI 上单击部署后, 在 GUI 上设置的变量将覆盖您在 `/etc/xdl/mcs/mcs.conf` 文件中设置的变量。
2. 根据您使用的公有云创建主映像的快照并命名。
 - 【适用于 **XenServer** (以前称为 **Citrix Hypervisor**)、**GCP** 和 **VMware vSphere**] 在模板 VM 上安装应用程序并关闭模板 VM。创建并命名主映像的快照。
 - (适用于 **Azure**) 在模板 VM 上安装应用程序后, 从 Azure 门户关闭模板 VM。请确保模板 VM 的电源状态显示 **Stopped (deallocated)** (已停止 (已取消分配))。记住此处的资源组名称。在 Azure 上查找您的主映像时需要该名称。



- (适用于 **AWS**) 在模板 VM 上安装应用程序后，从 AWS EC2 门户关闭模板 VM。请确保模板 VM 的实例状态显示 **Stopped**（已停止）。右键单击模板 VM，然后选择映像 > 创建映像。键入信息并根据需要进行设置。单击创建映像。



- (适用于 **Nutanix**) 在 Nutanix AHV 上，关闭模板 VM。创建并命名主映像的快照。

注意：

必须在 Acropolis 快照名称前加上前缀 **XD_** 才能在 Citrix Virtual Apps and Desktops 中使用。根据需要使用 Acropolis 控制台重命名快照。重命名快照后，重新启动创建目录向导以获取刷新后的列表。

步骤 4：创建计算机目录

1. 登录到 [Citrix Cloud](#)。
2. 在左上角的菜单中，选择我的服务 > **DaaS**。
3. 在管理 > 完整配置中，选择计算机目录。
4. 该向导将引导您创建计算机目录。

在 Nutanix 独有的 **Container**（容器）页面上，选择您之前为模板 VM 指定的容器。

在主映像页面上，选择映像快照。

在虚拟机页面上，检查虚拟 CPU 的数量以及每个 vCPU 的核心数。选择 MCS 作为计算机部署方法，然后选择未加入域作为要在目录中创建的计算机的标识。

根据需要执行其他配置任务。有关详细信息，请参阅[创建计算机目录](#)。

注意：

如果 Delivery Controller 上的计算机目录创建过程需要大量时间，请转至 Nutanix Prism 并手动打开前缀为 **Preparation** 的计算机的电源。这种方法有助于继续执行创建过程。

步骤 5：创建交付组

交付组是从一个或多个计算机目录中选择的计算机的集合。它指定哪些用户可以使用这些计算机，以及可供这些用户使用的应用程序和桌面。有关详细信息，请参阅[创建交付组](#)。

使用轻松安装创建未加入域的 **Linux VDA**（预览版）

August 20, 2024

本文将引导您使用轻松安装方法在 Citrix DAAS 或 Citrix Virtual Apps and Desktops 2407 及更高版本中使用计算机目录创建和注册未加入域的 Linux VDA。

重要：

- 对于 Citrix DaaS 客户：
 - 可以在公有云或本地数据中心部署未加入域的 VDA。未加入域的 VDA 由 Citrix DaaS 中的控制平面进行管理。
 - 要创建未加入域的 VDA，使用 Citrix Gateway 服务的客户必须确保启用 [Rendezvous V2](#)。仅当您计划在本地虚拟机管理程序上预配计算机或者想要在 Workspace 中使用 Active Directory 作为身份提供程序时，才需要使用 Cloud Connector。
- 要创建未加入域的 VDA，还可以使用 MCS。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)。
 - MCS 不支持裸机服务器。
- 以下功能适用于未加入域的 Linux VDA：
 - [在未加入域的 VDA 上创建具有指定属性的本地用户](#)
 - [非 SSO 身份验证](#)
 - [使用 Azure Active Directory 进行身份验证](#)
 - [Rendezvous V2](#)

步骤 1: 创建一个空的计算机目录

登录到 Citrix Web Studio 并创建一个不包含任何计算机的空计算机目录。请注意，Linux VDA 不支持使用令牌文件注册电源受管理的计算机目录。

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明，请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制，使得该过程不同于为 Windows VDA 计算机创建计算机目录：

- 对于操作系统，请选择：
 - 多会话操作系统选项（适用于托管共享桌面交付模型）。
 - 单会话操作系统选项（适用于 VDI 专用桌面交付模型）。
- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。
- 请勿在同一个计算机目录中混合加入了域和未加入域的计算机。
- 只有加入了域的计算机支持 Remote PC Access 计算机目录，未加入域的计算机不支持该目录。

注意：

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是，选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

步骤 2: 创建一个注册令牌

要使用轻松安装创建未加入域的 VDA，您需要一个令牌文件以将 VDA 注册到计算机目录中，并向云或本地 Delivery Controller 进行身份验证。Linux VDA 不支持使用令牌文件注册电源受管理的计算机目录。

要创建注册令牌，请在 Citrix Web Studio 中完成以下步骤：

提示：

在 Citrix DaaS 中，Web Studio 称为“完整配置”。

1. 选择您之前创建的空计算机目录，然后在操作栏中选择 **管理注册令牌**。
2. 在管理注册令牌页面上，单击生成以创建注册令牌。或者，您可以选择处于有效状态的现有令牌。
3. 按照向导完成设置。
4. 下载成功创建的令牌，将其保存在一个安全位置并使其具有最低权限。稍后，当您运行轻松安装脚本时，`CTX_XDL_NDJ_ENROLMENT_TOKEN_FILE` 变量允许您指定用于注册 VDA 的令牌文件的路径。

步骤 3: 安装.NET

除.NET Runtime 之外, 在安装或升级 Linux VDA 之前, 还必须在所有受支持的 Linux 发行版中安装.ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

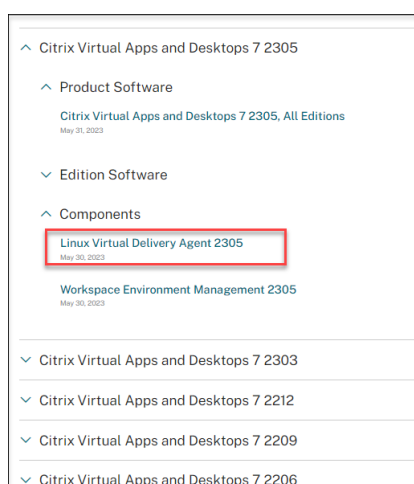
如果您的 Linux 发行版包含所需的.NET 版本, 请从内置源中进行安装。否则, 请从 Microsoft 软件包源中安装.NET。有关详细信息, 请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装.NET 后, 请运行 **which dotnet** 命令查找您的运行时路径。

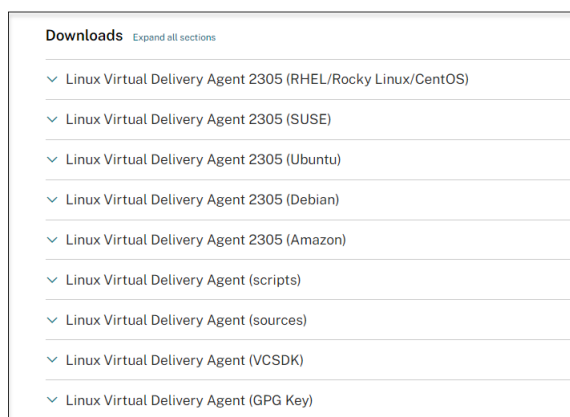
根据命令输出, 设置.NET Runtime 二进制文件路径。例如, 如果命令输出为 /aa/bb/dotnet, 请使用 /aa/bb 作为.NET 二进制文件路径。

步骤 4: 下载 Linux VDA 软件包

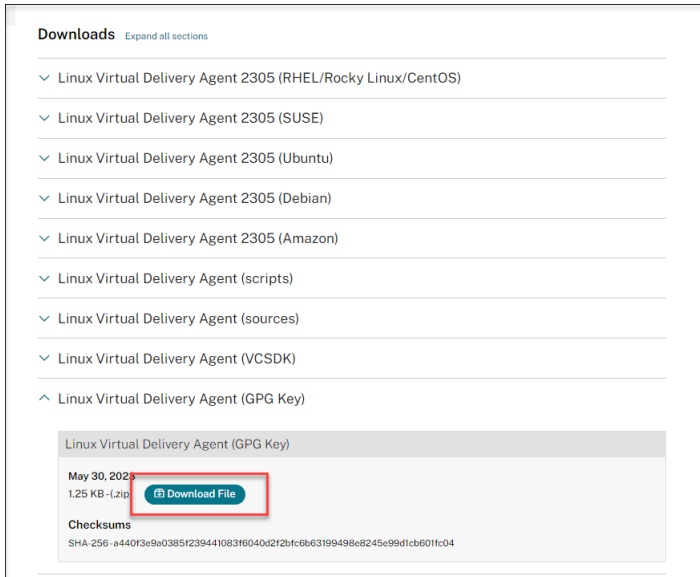
1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。
3. 展开组件以查找 Linux VDA。例如:



4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。
6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要使用公钥验证 Linux VDA 软件包的完整性，请执行以下操作：

- 对于 RPM 软件包，请运行以下命令以将公钥导入到 RPM 数据库中以及检查软件包的完整性：

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

- 对于 DEB 软件包，请运行以下命令以将公钥导入到 DEB 数据库中以及检查软件包的完整性：

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

步骤 5：安装 Linux VDA 软件包

要为 Linux VDA 设置环境，请运行以下命令。

对于 Amazon Linux 2、CentOS、RHEL 和 Rocky Linux 发行版：

注意：

- 对于 RHEL 和 CentOS，请先安装 EPEL 存储库，然后才能成功安装 Linux VDA。有关如何安装 EPEL 的信息，请参阅 <https://docs.fedoraproject.org/en-US/epel/> 上提供的说明。
- 在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上安装 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

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

注意：

在 GCP 上托管的 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 上安装 Linux VDA 后，以太网连接可能会断开，VM 重新启动后可能无法访问 Linux VDA。要解决此问题，请在首次登录 VM 时设置 root 用户密码，并确保您能够以 root 用户身份登录 VM。然后，在重新启动 VM 后，在控制台中运行以下命令：

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

对于 Ubuntu/Debian 发行版：

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

注意：

- 要为 Debian 11 发行版安装必要的依赖项，请将 `deb http://deb.debian.org/debian/bullseye main` 行添加到 `/etc/apt/sources.list` 文件。
- 对于 GCP 上的 Ubuntu 20.04，请禁用 RDNS。为此，请在 `/etc/krb5.conf` 的 `[libdefaults]` 下添加 `rdns = false` 行。

对于 SUSE 发行版：

1. 对于 AWS、Azure 和 GCP 上的 SUSE 15.5，请确保：

- 您使用的是 **libstdc++6** 版本 12 或更高版本。
- `/etc/sysconfig/windowmanager` 中的 **Default_WM** 参数设置为 “**gnome**”。

2. 请运行以下命令来安装 Linux VDA：

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

步骤 6：安装 NVIDIA GRID 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器（主机驱动程序），请参阅以下指南：

- [Citrix Hypervisor](#)
- [VMware ESX](#)
- [Nutanix AHV](#)

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序，请执行下面的常规步骤：

1. 确保来宾 VM 已关闭。
2. 在虚拟机管理程序控制面板中，为 VM 分配一个 GPU。

3. 启动 VM。
4. 在 VM 上安装来宾 VM 驱动程序。

步骤 7: 指定要使用的数据库

安装 Linux VDA 软件包后, 可以通过编辑 **/etc/xdl/db.conf** 来指定要使用的 SQLite 或 PostgreSQL。

为此, 请在运行 **sudo /opt/Citrix/VDA/sbin/ctxinstall.sh** 或 **/opt/Citrix/VDA/bin/easyinstall** 之前编辑 **/etc/xdl/db.conf**。

注意:

- 我们建议您仅对 VDI 模式使用 SQLite。
- 对于轻松安装和 MCS, 您可以指定使用 SQLite 或 PostgreSQL, 而不必手动安装。除非通过 **/etc/xdl/db.conf** 另行指定, 否则 Linux VDA 默认使用 PostgreSQL。
- 也可以使用 **/etc/xdl/db.conf** 配置 PostgreSQL 的端口号。

步骤 8: 运行轻松安装来配置环境和 VDA 以完成安装

安装 Linux VDA 软件包后, 使用 **ctxinstall.sh** 脚本来配置运行环境。

注意:

在设置运行时环境之前, 请确保已在您的操作系统中安装了 **en_US.UTF-8** 区域设置。如果该区域设置在您的操作系统中不可用, 请运行 **sudo locale-gen en_US.UTF-8** 命令。对于 Debian, 通过取消注释 **# en_US.UTF-8 UTF-8** 行来编辑 **/etc/locale.gen** 文件, 然后运行 **sudo locale-gen** 命令。

ctxinstall.sh

ctxinstall.sh 为轻松安装脚本, 用于进行一些预配置和设置 VDA 运行环境变量。

- 只有 root 用户可以运行此脚本。
- 轻松安装使用 **/opt/Citrix/VDA/sbin/ctxinstall.conf** 作为其配置文件来设置、保存和同步使用的所有环境变量的值。我们建议您仔细阅读模板 (**ctxinstall.conf.tpl**), 然后自定义自己的 **ctxinstall.conf**。首次创建配置文件时, 请使用以下任一方法:
 - 通过复制 **/opt/Citrix/VDA/sbin/ctxinstall.conf.tpl** 模板文件并将其另存为 **/opt/Citrix/VDA/sbin/ctxinstall.conf**。
 - 通过运行 **ctxinstall.sh**。每次运行 **ctxinstall.sh** 时, 您的输入都会保存在 **/opt/Citrix/VDA/sbin/ctxinstall.conf** 中。
- 轻松安装功能支持模块化运行。模块包括预检查、安装、域配置、设置和验证。

- 可以在 `/var/log/xdl/ctxinstall.log` 中找到此脚本的调试详细信息。

有关详细信息，请使用帮助命令 **ctxinstall.sh -h**。

注意：

- 遵循最小权限原则，确保只有 root 用户才能读取 `/opt/Citrix/VDA/sbin/ctxinstall.conf`，因为文件中可能会设置域加入密码。
- 卸载 Linux VDA 会删除 `/opt/Citrix/VDA` 下的文件。我们建议您在卸载 VDA 之前备份 `/opt/Citrix/VDA/sbin/ctxinstall.conf`。

可以在交互模式或无提示模式下运行 `ctxinstall.sh`。在运行该脚本之前，请设置以下环境变量：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于未加入域的场景，请将其设置为 "y"。
- **CTX_XDL_NDJ_ENROLLMENT_TOKEN_FILE=' <path-to-token-file-on-vda-machine>'** - 要使用轻松安装创建未加入域的 VDA，您需要一个令牌文件以将 VDA 注册到 Delivery Controller 的计算机目录中。将令牌保存在一个安全位置并使其具有最低权限。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境，将该值设置为 'y'。
- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro，这是一组 GPU 加速技术，旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro，则要为 VDI 桌面（单会话）模式配置 VDA（即 `CTX_XDL_VDI_MODE=' y'`）。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口（默认端口 80 和 1494）。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate** - 指定要在会话中使用的 GNOME、GNOME Classic 或 MATE 桌面环境。如果不指定变量，则使用在 VDA 上配置的默认桌面。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/kde/mate/xfce/ <none>** - 指定要在会话中使用的 GNOME、GNOME Classic、KDE、MATE 或 Xfce 桌面环境。如果将其设置为 '<none>'，则将使用在 VDA 上配置的默认桌面。可以使用 `CTX_XDL_DESKTOP_ENVIRONMENT` 变量指定要在会话中使用的桌面环境。您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息，请参阅[桌面切换命令](#)和[系统托盘](#)。
- **CTX_XDL_DOTNET_RUNTIME_PATH=' <path-to-install-dotnet-runtime>'** - 安装 .NET 以支持新的 Broker 代理服务 (ctxvda) 的路径。默认路径为 '/usr/bin'。
- **CTX_XDL_VDA_PORT=' <port-number>'** - Linux VDA 将通过 TCP/IP 端口与 Delivery Controller 通信。

交互式模式 要在交互模式下运行 `ctxinstall.sh` 脚本，请使用不带 `-S` 选项的 `sudo /opt/Citrix/VDA/sbin/ctxinstall.sh` 命令。在命令行界面的每个提示符处键入相关的变量值。如果已经设置了变量，`ctxinstall.sh` 会要求您确认，以防您想要对其进行更改。

无提示模式 在无提示模式下，必须使用 `/opt/Citrix/VDA/sbin/ctxinstall.conf` 或导出命令设置前面的变量。之后，请运行 `ctxinstall.sh -S`（请注意，此处字母 **S** 为大写字母）。如果未设置所有必需变量或者某些值无效，则除非有默认值，否则 `ctxinstall.sh` 将中止执行。

如果对其进行设置，每个变量的导出值将覆盖 `/Citrix/VDA/sbin/ctxinstall.conf` 中的值。更新后的所有值都保存在 `/Citrix/VDA/sbin/ctxinstall.conf` 中。

```

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

```

运行 `sudo` 命令时，键入 `-E` 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 `#!/bin/bash` 作为第一行来创建 shell 脚本文件。

或者，您可以使用单个命令指定所有变量。

要设置 VDA 运行环境变量（以 `CTX_XDL_` 开头的变量），您可以运行 `ctxinstall.sh -s`（请注意，此处字母 **s** 为小写字母）。

步骤 9: 运行 XDPing

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅 [XDPing](#)。

步骤 10: 运行 Linux VDA

启动 **Linux VDA**:

启动 Linux VDA 服务:

```

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

```

停止 **Linux VDA**:

要停止 Linux VDA 服务，请执行以下操作：

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

注意：

在停止 **ctxvda** 和 **ctxhdx** 服务之前，请运行 **systemctl stop ctxmonitord** 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

重新启动 **Linux VDA**:

重新启动 Linux VDA 服务：

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

检查 **Linux VDA** 的状态：

要检查 Linux VDA 服务的运行状态，请执行以下操作：

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

步骤 **11**：创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明，请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制：

- 确保所选 AD 用户和组已正确配置，能够登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的（匿名）用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

重要：

Linux VDA 1.4 及更高版本支持发布应用程序。但是，Linux VDA 不支持将桌面和应用程序交付给相同的计算机。

有关如何创建计算机目录和交付组的信息，请参阅 [Citrix Virtual Apps and Desktops 7 2407](#)。

使用 **Machine Creation Services (MCS)** 创建 **Linux VDA**

August 21, 2024

可以使用 MCS 创建加入了域和未加入域的 VDA。如果您想在 Citrix DaaS 中创建未加入域的 Linux VDA，还可以参考专门的文章[使用 MCS 创建未加入域的 Linux VDA](#)。

重要：

下面是自 2212 版起所做的重要更改：

- /etc/xdl/mcs/mcs.conf 文件或轻松安装 GUI 上的这一 **AD_INTEGRATION** 变量不再有默认值。必须根据需要设置一个值。有关详细信息，请参阅本文中的[步骤 3h: 配置 MCS 变量](#)部分。
- /etc/xdl/mcs/mcs.conf 中的 **UPDATE_MACHINE_PW** 条目的有效值不再设置为 **enabled** 或 **disabled**，而是设置为 **Y** 或 **N**。有关详细信息，请参阅本文中的[自动更新计算机帐户密码](#)部分。

支持的发行版

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

- Citrix 使用以下 Centrify 版本在相关 Linux 发行版上进行初始功能验证：

Linux 发行版	Centrify 版本
RHEL 7/8	5.8.0
SUSE	5.7.1
Debian、Ubuntu	5.6.1

使用其他版本的 Centrify 可能会导致出现错误。请勿使用 Centrify 将模板计算机加入到域中。

- 如果您使用 PBIS 或 Centrify 将 MCS 创建的计算机加入到 Windows 域，请完成以下任务：
 - 在模板计算机上，在 `/etc/xdl/mcs/mcs.conf` 文件中配置 PBIS 或 Centrify 软件包下载路径或直接安装 PBIS 或 Centrify 软件包。
 - 在运行 `/opt/Citrix/VDA/sbin/deploymcs.sh` 之前，请创建一个组织单位 (OU)，该组织单位对其所有从属 MCS 创建的计算机具有写入和密码重置权限。
 - 在 `/opt/Citrix/VDA/sbin/deploymcs.sh` 完成运行后重新启动 MCS 创建的计算机之前，请根据您的部署在 Delivery Controller 或 Citrix Cloud Connector 上运行 `klist -li 0x3e4 purge`。
- 要使用当前正在运行的使用 SSSD 连接到域的 RHEL 8.x/9.x 或 Rocky Linux 8.x/9.x VDA 作为 MCS 的模板 VM，请确保：
 - VDA 是手动安装的，而不是通过轻松安装来安装的。轻松安装使用适用于 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 的 **Adcli**，而 MCS 不支持 SSSD 和 **Adcli** 的组合。
 - Samba 服务器配置为使用 SSSD 进行 AD 身份验证。有关详细信息，请参阅 Red Hat 文章，网址为 <https://access.redhat.com/solutions/3802321>。

受支持的虚拟机管理程序

- AWS
- XenServer (以前称为 Citrix Hypervisor)
- GCP
- Microsoft Azure
- Nutanix AHV
- VMware vSphere

如果您尝试在受支持的虚拟机管理程序以外的其他虚拟机管理程序上准备主映像，可能会出现意外结果。

使用 **MCS** 创建 **Linux VM**

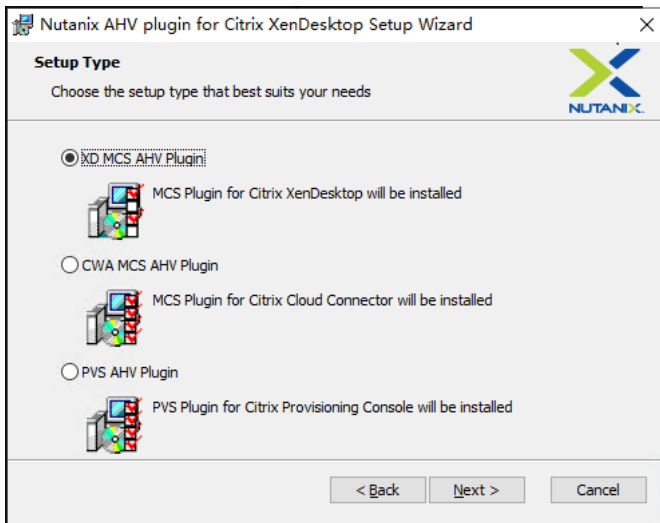
注意事项

- 自 2203 版本起，您可以在 Microsoft Azure、AWS 和 GCP 上为 Citrix Virtual Apps and Desktops 以及 Citrix DaaS (以前称为 Citrix Virtual Apps and Desktops 服务) 托管 Linux VDA。要将这些公有云主机连接添加到您的 Citrix Virtual Apps and Desktops 部署中，您需要 Citrix Universal 混合多云 (HMC, Citrix Universal Hybrid Multi-Cloud) 许可证。
- 不支持将裸机服务器与 MCS 结合使用以创建虚拟机。

(仅限 **Nutanix**) 步骤 **1**: 安装并注册 **Nutanix AHV** 插件

从 Nutanix 中获取 Nutanix AHV 插件包。在您的 Citrix Virtual Apps and Desktops 环境中安装并注册该插件。有关详细信息，请参阅 [Nutanix 支持门户](#) 中提供的《Nutanix Acropolis MCS Plugin Installation Guide》(Nutanix Acropolis MCS 插件安装指南)。

步骤 **1a**: 为本地 **Delivery Controller** 安装并注册 **Nutanix AHV** 插件 安装 Citrix Virtual Apps and Desktops 后，选择并在 Delivery Controller 上安装 **XD MCS AHV** 插件。



步骤 **1b**: 为云端 **Delivery Controller** 安装并注册 **Nutanix AHV** 插件 选择并安装适用于 Citrix Cloud Connector 的 **CWA MCS AHV** 插件。在向 Citrix Cloud 租户注册的所有 Citrix Cloud Connector 上安装插件。即使 Citrix Cloud Connector 在没有 AHV 的情况下提供资源位置，也必须注册 Citrix Cloud Connector。

步骤 **1c**: 安装插件后完成以下步骤

- 验证是否在 `C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\1.0.0.0` 中创建了 Nutanix Acropolis 文件夹。
- 运行 `"C:\Program Files\Common Files\Citrix\HCLPlugins\RegisterPlugins.exe"-PluginsRoot "C:\Program Files\Common Files\Citrix\HCLPlugins\CitrixMachineCreation\1.0.0.0"` 命令。
- 在本地 Delivery Controller 上重新启动 Citrix Host Service、Citrix Broker Service 和 Citrix Machine Creation Service，或者在 Citrix Cloud Connector 上重新启动 Citrix RemoteHCLServer Service。

提示：

我们建议您在安装或更新 Nutanix AHV 插件时停止并重新启动 Citrix Host Service、Citrix Broker Service 和 Machine Creation Service。

步骤 2: 创建主机连接

本部分内容举例说明了如何创建与 Azure、AWS、XenServer（以前称为 Citrix Hypervisor）、GCP、Nutanix AHV 和 VMware vSphere 的主机连接。

注意:

对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。

对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。

有关详细信息，请参阅 Citrix Virtual Apps and Desktops 文档中的[创建和管理连接和资源](#)以及 Citrix DaaS 文档中的[创建和管理连接](#)。

- [在 Citrix Studio 中创建与 Azure 的主机连接](#)
- [在 Citrix Studio 中创建与 AWS 的主机连接](#)
- [在 Citrix Studio 中创建与 XenServer 的主机连接](#)
- [在 Citrix Studio 中创建与 GCP 的主机连接](#)
- [在 Citrix Studio 中创建与 Nutanix 的主机连接](#)
- [在 Citrix Studio 中创建与 VMware 的主机连接](#)

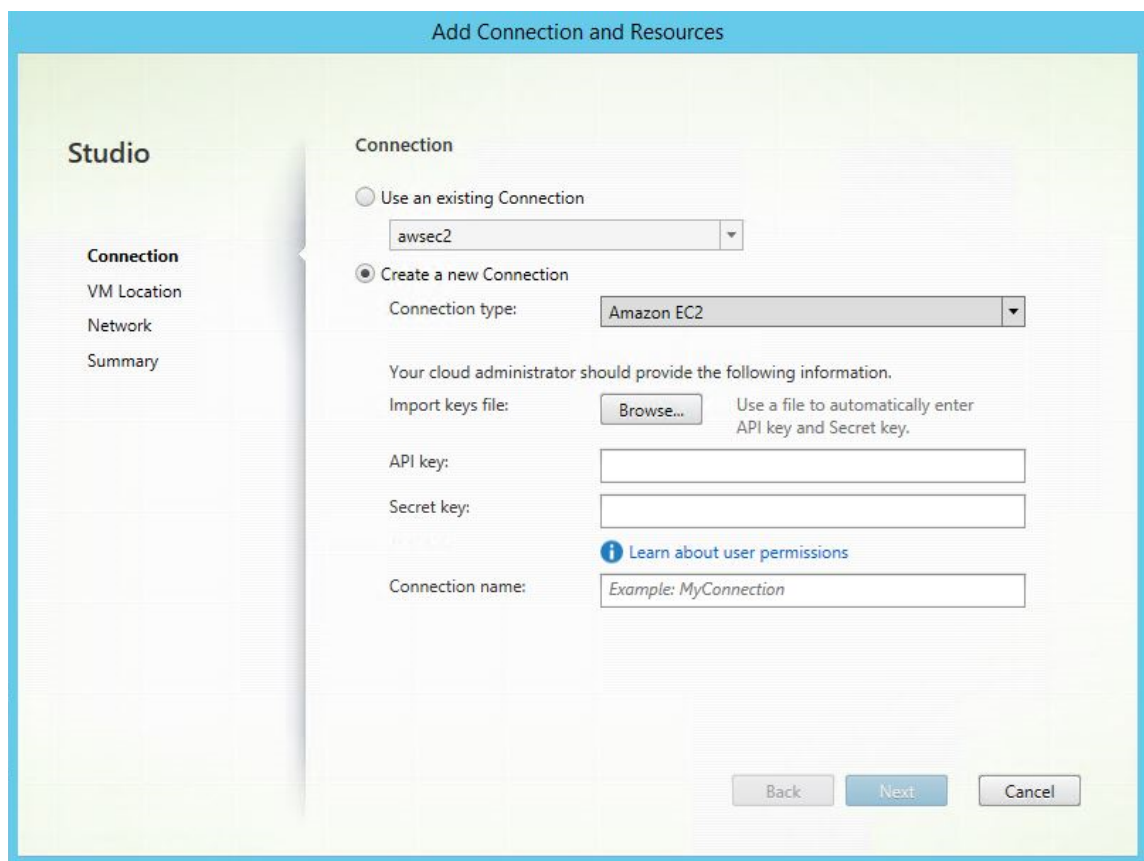
在 **Citrix Studio** 中创建与 **Azure** 的主机连接

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
2. 在添加连接和资源向导中，选择 Microsoft Azure 作为连接类型。
3. 选择“Microsoft Azure”作为连接类型。
4. 向导将引导您完成各个页面。特定页面内容取决于所选连接类型。完成每一页之后，请选择下一步，直到到达摘要页为止。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)一文中的步骤 2: 创建主机连接。

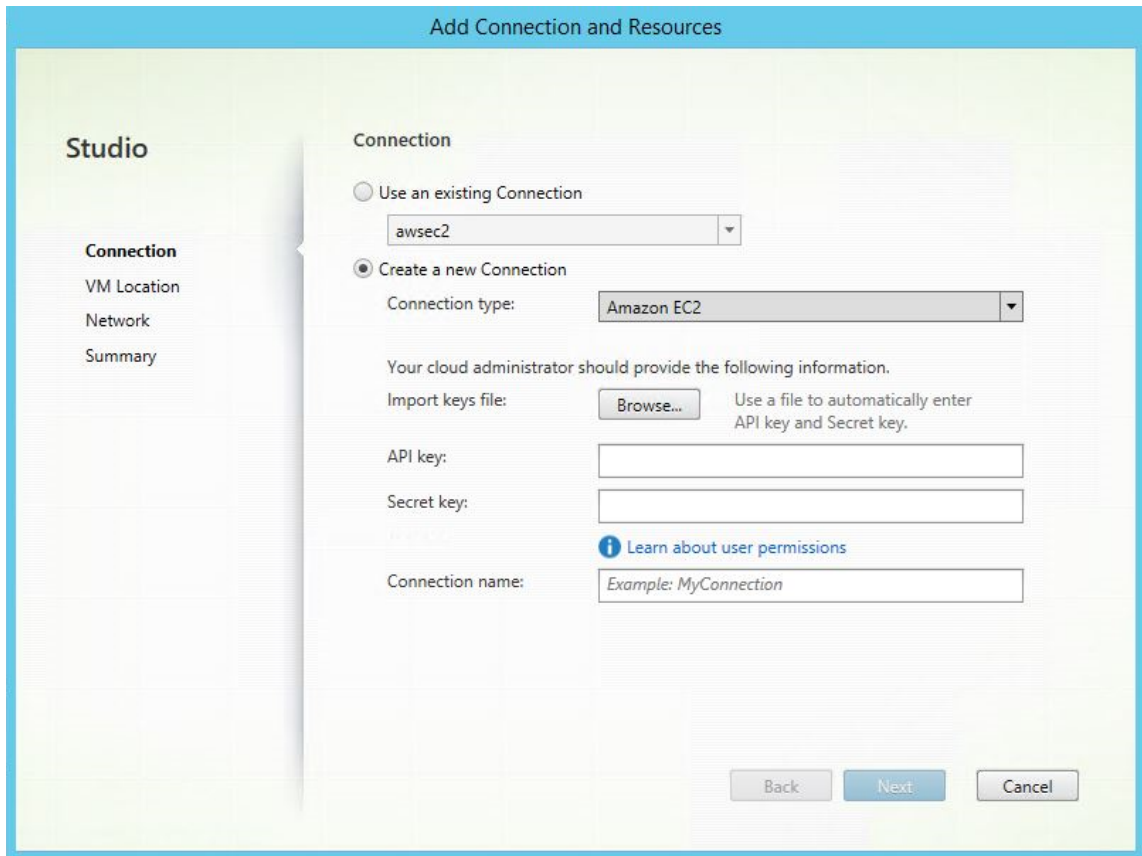
在 **Citrix Studio** 中创建与 **AWS** 的主机连接

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
2. 在添加连接和资源向导中，选择 **Amazon EC2** 作为连接类型。

例如，在本地 Citrix Studio 中：



3. 键入 AWS 帐户的 API 密钥和密钥，然后键入您的连接名称。



API 密钥是您的访问密钥 ID，密钥是您的秘密访问密钥。这些密钥被视为访问密钥对。如果您丢失了秘密访问密钥，则可以删除该访问密钥并创建另一个访问密钥。要创建访问密钥，请执行以下操作：

- a) 登录 AWS 服务。
 - b) 导航到身份和访问管理 (IAM) 控制台。
 - 1 在左侧导航窗格中，选择用户。
 - c) 选择目标用户并向下滚动以选择安全凭据选项卡。
 - d) 向下滚动并单击创建访问密钥。此时将显示一个新窗口。
 - e) 单击下载 **.csv** 文件并将访问密钥保存到一个安全的位置。
4. 向导将引导您完成各个页面。特定页面内容取决于所选连接类型。完成每一页之后，请选择下一步，直到到达摘要页为止。

在 **Citrix Studio** 中创建与 **XenServer** 的主机连接

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
2. 在添加连接和资源向导中，在连接类型字段中选择 XenServer（以前称为 Citrix Hypervisor）。
3. 键入连接地址 (XenServer URL) 和凭据。

4. 输入连接名称。

在 **Citrix Studio** 中创建与 **GCP** 的主机连接 根据 [Google 云端平台虚拟化环境](#) 设置您的 GCP 环境，然后完成以下步骤以创建与 GCP 的主机连接。

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
2. 在添加连接和资源向导中，选择 **Google** 云端平台作为连接类型。

例如，在 Citrix Cloud 上的基于 Web 的 Studio 控制台中：

The screenshot shows the 'Add Connection and Resources' wizard. The left sidebar has five steps: 1. Connection (selected), 2. Region, 3. Network, 4. Scopes, and 5. Summary. The main content area is titled 'Connection' and has two radio buttons: 'Use an existing connection' (unselected) and 'Create a new connection' (selected). Under 'Use an existing connection', there is a dropdown menu with 'Bing Test' selected. Under 'Create a new connection', there are several fields: 'Zone name' (dropdown), 'Connection type' (dropdown with 'Google Cloud Platform' selected), 'Service account key' (with an 'Import key...' button), 'Service account ID' (text input), and 'Connection name' (text input). At the bottom, there are three buttons: 'Next', 'Cancel', and a circular icon with the number '7'.

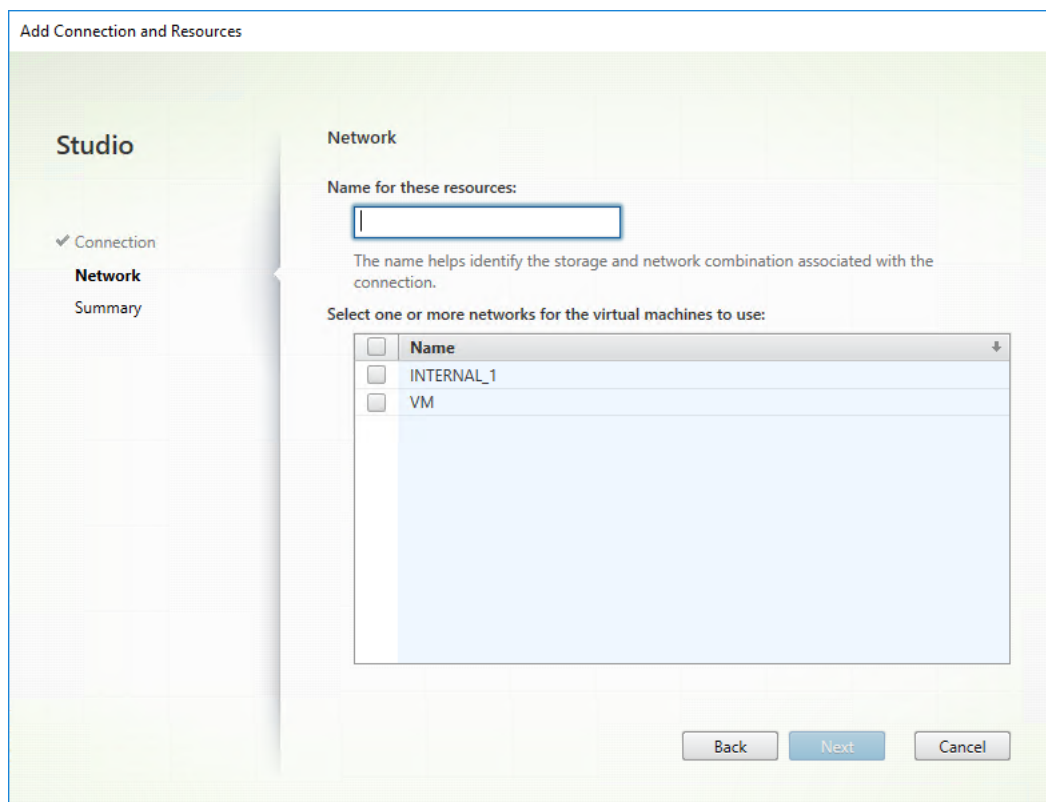
3. 导入 GCP 帐户的服务帐号密钥并键入连接名称。
4. 向导将引导您完成各个页面。特定页面内容取决于所选连接类型。完成每一页之后，请选择下一步，直到到达摘要页为止。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)一文中的步骤 **2**：创建主机连接。

在 **Citrix Studio** 中创建与 **Nutanix** 的主机连接

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。

2. 在添加连接和资源向导中的连接页面上选择 Nutanix AHV 作为连接类型，然后指定虚拟机管理程序地址、凭据以及连接名称。在网络页面上，选择用于单元的网络。

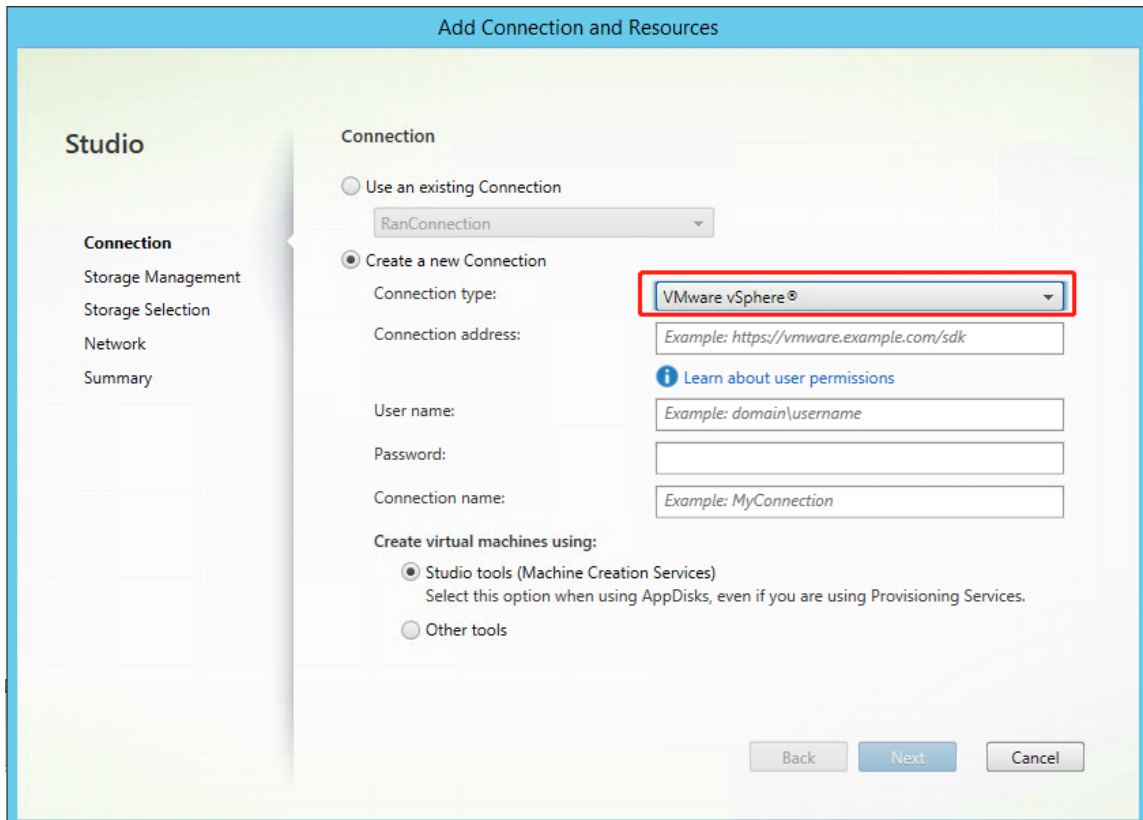
例如，在本地 Citrix Studio 中：



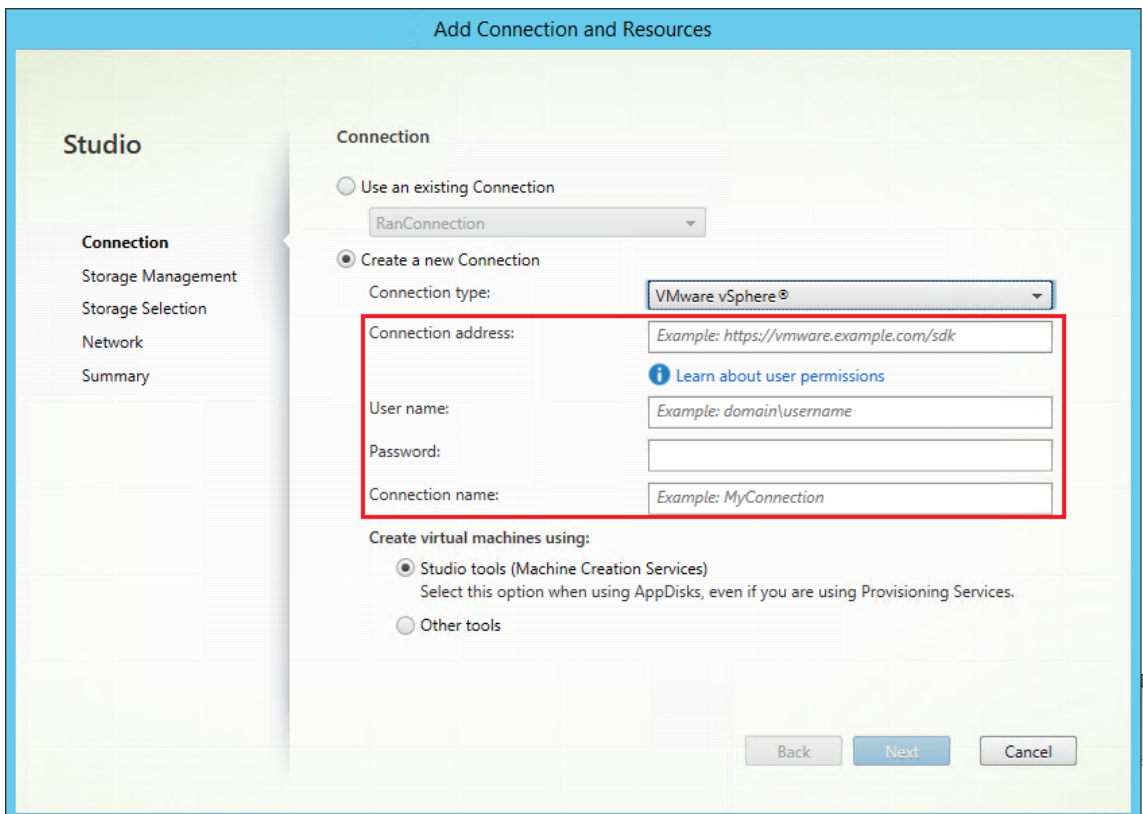
在 **Citrix Studio** 中创建与 **VMware** 的主机连接

1. 在 vSphere 环境中安装 vCenter Server。有关详细信息，请参阅 [VMware vSphere](#)。
2. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
3. 选择 VMware vSphere 作为连接类型。

例如，在本地 Citrix Studio 中：



4. 键入 VMware 帐户的连接地址 (vCenter Server URL)、您的凭据和连接名称。



步骤 3：准备主映像

(仅适用于 **XenServer**) **步骤 3a：安装 XenServer VM Tools** 在模板 VM 上安装 XenServer VM Tools，以便每个 VM 都使用 xe CLI 或 XenCenter。除非安装这些工具，否则 VM 的性能可能会很慢。如果没有这些工具，无法执行以下任何操作：

- 彻底关闭、重新启动或挂起 VM。
 - 在 XenCenter 中查看 VM 性能数据。
 - 迁移正在运行的 VM (通过 [XenMotion](#))。
 - 创建快照或带有内存 (检查点) 的快照，以及还原到快照。
 - 在正在运行的 Linux VM 上调整 vCPU 数。
1. 根据正在使用的虚拟机管理程序版本，从 [XenServer 下载页面](#) 或 [Citrix Hypervisor 下载页面](#) 下载适用于 Linux 的 XenServer VM Tools 文件。
 2. 将 `LinuxGuestTools-xxx.tar.gz` 文件复制到 Linux VM 或 Linux VM 可以访问的共享驱动器。
 3. 解压 tar 文件的内容：`tar -xzf LinuxGuestTools-xxx.tar.gz`
 4. 根据您的 Linux 发行版，运行以下命令安装 `xe-guest-utilities` 软件包。

对于 **RHEL/CentOS/Rocky Linux/SUSE**：

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

对于 **Ubuntu/Debian**：

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

5. 在 XenCenter 中的常规选项卡上检查模板 VM 的虚拟化状态。如果正确安装了 XenServer VM Tools，则虚拟化状态显示已优化。

步骤 3b：在 **AWS**、**Azure** 和 **GCP** 上验证 **SUSE 15.5** 的配置 对于 AWS、Azure 和 GCP 上的 SUSE 15.5，请确保：

- 您使用的是 **libstdc++6** 版本 12 或更高版本。
- `/etc/sysconfig/windowmanager` 中的 **Default_WM** 参数设置为 “**gnome**”。

步骤 3c：在 **GCP** 上为 **Ubuntu 20.04** 禁用 **RDNS** 在模板 VM 上，在 `/etc/krb5.conf` 的 **[libdefaults]** 下添加 **rdns = false** 行。

步骤 3d: 在模板 VM 上安装 .NET 和 Linux VDA 软件包

注意:

要使用当前正在运行的 VDA 作为模板 VM，请跳过此步骤。要使用当前正在运行的使用 SSSD 连接到域的 RHEL 8.x/9.x 或 Rocky Linux 8.x/9.x VDA 作为模板 VM，请确保：

- VDA 是手动安装的，而不是通过轻松安装来安装的。轻松安装使用适用于 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 的 **Adcli**，而 MCS 不支持 SSSD 和 **Adcli** 的组合。
- Samba 服务器配置为使用 SSSD 进行 AD 身份验证。有关详细信息，请参阅 Red Hat 文章，网址为 <https://access.redhat.com/solutions/3802321>。

在安装 Linux VDA 软件包之前，请在模板 VM 上安装 .NET 并注意以下事项：

- 除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。
- 如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装 .NET 后，根据您的 Linux 发行版运行以下命令以安装 Linux VDA：

对于 RHEL/CentOS/Rocky Linux:

注意:

在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上安装 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

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

对于 Ubuntu/Debian:

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

对于 SUSE:

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

(仅限 **RHEL 7**) 步骤 **3e**: 启用存储库以安装 **tdb-tools** 软件包 对于 **RHEL 7** 服务器:

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

对于 RHEL 7 工作站:

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

步骤 3f: (仅限 **RHEL**) 安装可以提供 **ntfs-3g** 的 **EPEL** 存储库 在 RHEL 8 上安装 EPEL 存储库。有关如何安装 EPEL 的信息, 请参阅 <https://docs.fedoraproject.org/en-US/epel/> 上提供的说明。

步骤 3g: (仅适用于 **SUSE**) 手动安装 **ntfs-3g** 在 SUSE 平台上, 所有存储库都不提供 **ntfs-3g**。下载源代码, 编译并手动安装 **ntfs-3g**:

1. 安装 GNU Compiler Collection (GCC) 编译器系统并将软件包设置为:

```
1 sudo zypper install gcc
2 sudo zypper install make
```

2. 下载 **ntfs-3g** 软件包。

3. 解压缩 **ntfs-3g** 软件包:

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

4. 输入 **ntfs-3g** 软件包的路径:

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

5. 安装 **ntfs-3g**:

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

步骤 3h: 指定要使用的数据库 安装 Linux VDA 软件包后, 您可以在 SQLite 与 PostgreSQL 之间切换。为此, 请完成以下步骤:

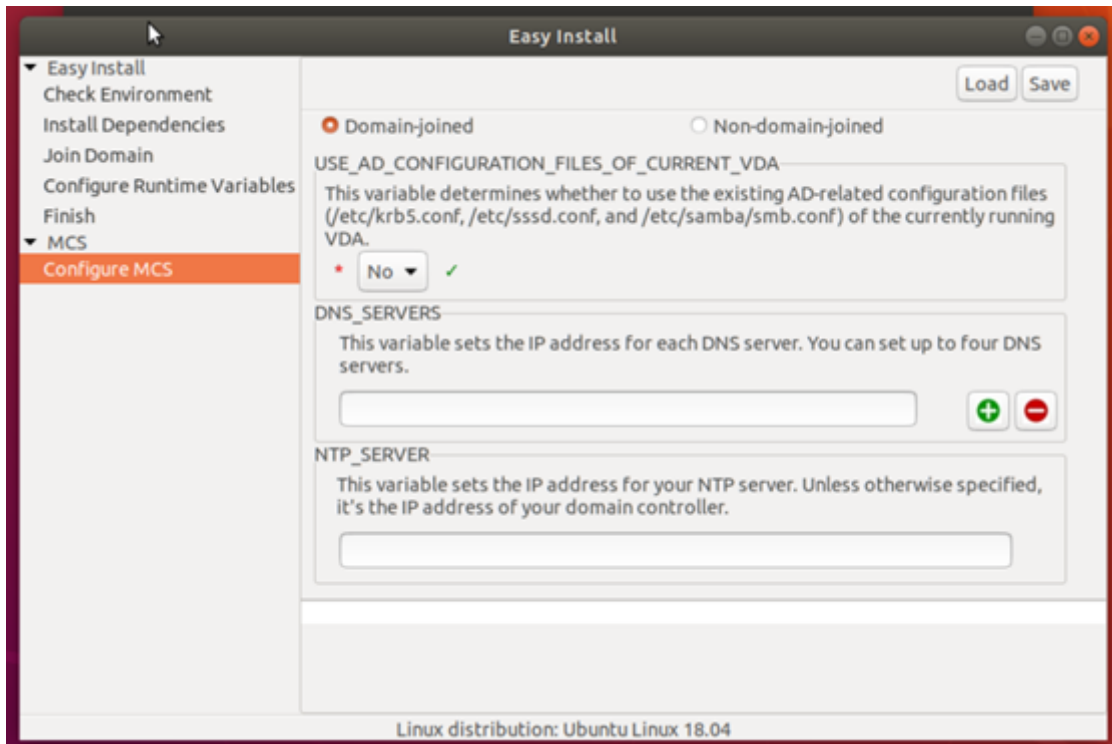
注意:

- 我们建议您仅将 SQLite 用于 VDI 模式, 将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS, 您可以指定使用 SQLite 或 PostgreSQL, 而不必手动安装。除非通过 **/etc/xdl/db.conf** 另行指定, 否则 Linux VDA 默认使用 PostgreSQL。
- 也可以使用 **/etc/xdl/db.conf** 配置 PostgreSQL 的端口号。

1. 运行 **/opt/Citrix/VDA/sbin/ctxcleanup.sh**。如果是全新安装, 则省略此步骤。
2. 请在运行 **deploymcs.sh** 前编辑 **/etc/xdl/db.conf**。

步骤 3i: 配置 **MCS** 变量 配置 MCS 变量有两种方法:

- 编辑 **/etc/xdl/mcs/mcs.conf** 文件。
- 使用轻松安装 GUI。要打开轻松安装 GUI, 请在您的 Linux VDA 的桌面环境中运行 **/opt/Citrix/VDA/bin/easyinstall** 命令。



提示：

单击保存将变量设置保存到指定路径下的本地文件中。单击加载从您指定的文件加载变量设置。

下面是您可以为未加入域和已加入域的场景配置的 MCS 变量：

- 适用于未加入域的场景

可以使用默认变量值或者根据需要自定义变量（可选）：

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

`DESKTOP_ENVIRONMENT=gnome | mate`

`REGISTER_SERVICE=Y | N`

`ADD_FIREWALL_RULES=Y | N`

`VDI_MODE=Y | N`

`START_SERVICE=Y | N`

- 适用于已加入域的场景

- `Use_AD_Configuration_Files_Of_Current_VDA`：确定是否使用当前正在运行的 VDA 的现有 AD 相关配置文件（`/etc/krb5.conf`、`/etc/sss.conf` 和 `/etc/samba/smb.conf`）。如果设置为 Y，MCS 创建的计算机上的配置文件将与当前正在运行的 VDA 上的等效文件相同。但是，您仍然必须配置 `dns` 和 `AD_INTEGRATION` 变量。默认值为 N，这意味着主映像上的配置模板负责确定 MCS 创建的计算机上的配置文件。要使用当前正在运行的 VDA 作为模板 VM，请将该值设置为 Y。否则，请将其设置为 N。

- **dns**: 设置每个 DNS 服务器的 IP 地址。您最多可以设置四个 DNS 服务器。
- **NTP_SERVER**: 设置 NTP 服务器的 IP 地址。除非另行说明, 否则该地址是您的域控制器的 IP 地址。
- **WORKGROUP**: 将工作组名称设置为您在 AD 中配置的 NetBIOS 名称(区分大小写)。否则, MCS 将使用域名中紧随计算机主机名之后的部分作为工作组名称。例如, 如果计算机帐户为 **user1.lvda.citrix.com**, MCS 将使用 **lvda** 作为工作组名称, 而 **citrix** 是正确的选择。请务必正确设置工作组名称。
- **AD_INTEGRATION**: 设置 SSSD、Winbind、PBIS 或 Centrify。有关 MSC 支持的 Linux 发行版和域加入方法的列表, 请参阅本文中的支持¹的发行版。
- **CENTRIFY_DOWNLOAD_PATH**: 设置下载 Server Suite Free (以前称为 Centrify Express) 软件包的路径。只有将 **AD_INTEGRATION** 变量设置为 Centrify 时, 该值才会生效。
- **CENTRIFY_SAMBA_DOWNLOAD_PATH**: 设置下载 Centrify Samba 软件包的路径。只有将 **AD_INTEGRATION** 变量设置为 Centrify 时, 该值才会生效。
- **PBIS_DOWNLOAD_PATH**: 设置下载 PBIS 软件包的路径。只有将 **AD_INTEGRATION** 变量设置为 PBIS 时, 该值才会生效。
- **UPDATE_MACHINE_PW**: 启用或禁用自动更新计算机帐户密码。有关详细信息, 请参阅[自动更新计算机帐户密码](#)。

- Linux VDA 配置变量:

```
DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime
DESKTOP_ENVIRONMENT=gnome | mate
SUPPORT_DDC_AS_CNAME=Y | N
VDA_PORT=port-number
REGISTER_SERVICE=Y | N
ADD_FIREWALL_RULES=Y | N
HDX_3D_PRO=Y | N
VDI_MODE=Y | N
SITE_NAME=dns-site-name | '<none>'
LDAP_LIST=' list-ldap-servers' | '<none>'
SEARCH_BASE=search-base-set | '<none>'
FAS_LIST=' list-fas-servers' | '<none>'
START_SERVICE=Y | N
TELEMETRY_SOCKET_PORT=port-number
TELEMETRY_PORT=port-number
```

(可选) 步骤 **3j**: 写入或更新 **MCS** 的注册表值 在模板计算机上, 将命令行添加到 `/etc/xdm/mcs/mcs_local_setting.reg` 文件以根据需要写入或更新注册表值。此操作可防止数据和设置在 MCS 预配的计算机每次重新启动时丢失。

`/etc/xdl/mcs/mcs_local_setting.reg` 文件中的每一行就是用于设置或更新注册表值的一个命令。

例如，您可以将以下命令行添加到 `/etc/xdl/mcs/mcs_local_setting.reg` 文件中，以分别写入或更新注册表值：

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

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

注意

要修改 MCS 的设置，您可以编辑 `/etc/xdl/ad_join` 和 `/etc/xdl/mcs/` 下的文件，但禁止编辑 `/var/xdl/mcs` 下的任何文件。

步骤 3j: 创建主映像

1. (仅适用于 SSSD + RHEL 8.x/9.x 或 Rocky Linux 8.x/9.x) 运行 `update-crypto-policies --set DEFAULT:AD-SUPPORT` 命令，然后重新启动模板 VM。
2. 如果您通过编辑 `/etc/xdl/mcs/mcs.conf` 配置 MCS 变量，请运行 `/opt/Citrix/VDA/sbin/deploymcs.sh`。如果您使用 GUI 配置 MCS 变量，请单击部署。在 GUI 上单击部署后，在 GUI 上设置的变量将覆盖您在 `/etc/xdl/mcs/mcs.conf` 文件中设置的变量。
3. (如果使用当前正在运行的 VDA 作为模板 VM，或者如果是未加入域的场景，请跳过此步骤。) 在模板 VM 上，更新配置模板以自定义创建的所有 VM 上的相关 `/etc/krb5.conf`、`/etc/samba/smb.conf` 和 `/etc/sss/sss.conf` 文件。

对于 Winbind 用户，请更新 `/etc/xdl/ad_join/winbind_krb5.conf.tpl` 和 `/etc/xdl/ad_join/winbind_smb.conf.tpl` 模板。

对于 SSSD 用户，请更新 `/etc/xdl/ad_join/sss.conf.tpl`、`/etc/xdl/ad_join/sss_krb5.conf.tpl` 和 `/etc/xdl/ad_join/sss_smb.conf.tpl` 模板。

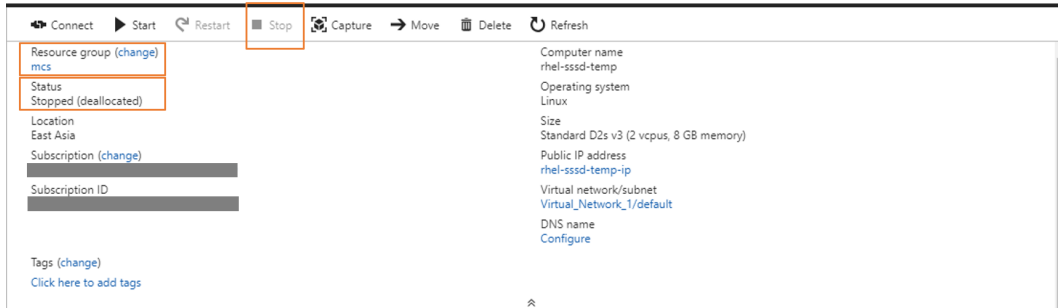
对于 Centrify 用户，请更新 `/etc/xdl/ad_join/centrify_krb5.conf.tpl` 和 `/etc/xdl/ad_join/centrify_smb.conf.tpl` 模板。

注意：

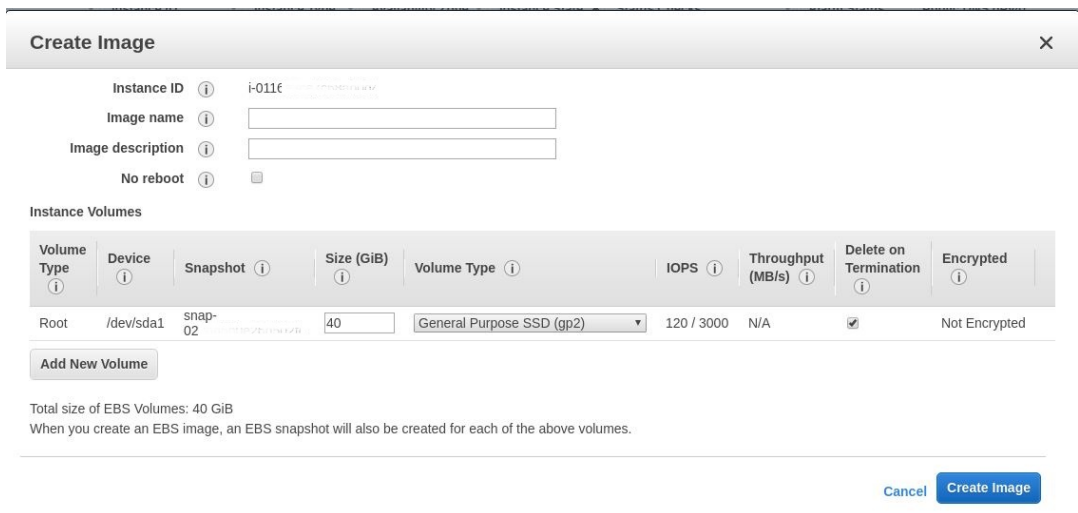
请保留模板文件中使用的现有格式并使用变量，例如 `$WORKGROUP`、`$REALM`、`$realm`、`${new_hostname}` 和 `$AD_FQDN`。

4. 根据您使用的公有云创建主映像的快照并命名。
 - (适用于 **XenServer**、**GCP** 和 **VMware vSphere**) 在模板 VM 上安装应用程序并关闭模板 VM。创建并命名主映像的快照。

- (适用于 **Azure**) 在模板 VM 上安装应用程序后，从 Azure 门户关闭模板 VM。确保模板 VM 的电源状态为 **Stopped (deallocated)** (已停止 (已取消分配))。记住此处的资源组名称。在 Azure 上查找您的主映像时需要该名称。



- (适用于 **AWS**) 在模板 VM 上安装应用程序后，从 AWS EC2 门户关闭模板 VM。确保模板 VM 的电源状态为已停止。右键单击模板 VM，然后选择映像 > 创建映像。键入信息并根据需要进行设置。单击创建映像。



- (适用于 **Nutanix**) 在 Nutanix AHV 上，关闭模板 VM。创建并命名主映像的快照。

注意：

必须在 Acropolis 快照名称前加上前缀 **XD_** 才能在 Citrix Virtual Apps and Desktops 中使用。根据需要使用 Acropolis 控制台重命名快照。重命名快照后，重新启动创建目录向导以获取刷新后的列表。

(对于 **GCP**) 步骤 **3I**: 在 **RHEL 8.x/9.x** 和 **Rocky Linux 8.x/9.x** 上配置以太网连接 在 GCP 上托管的 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 上安装 Linux VDA 后，以太网连接可能会断开，VM 重新启动后可能无法访问 Linux VDA。要解决此问题，请在首次登录 VM 时设置 root 用户密码，并确保您能够以 root 用户身份登录 VM。然后，在重新启动 VM 后，在控制台中运行以下命令：

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

步骤 4: 创建计算机目录

在 Citrix Studio 或 Web Studio 中，创建计算机目录并指定要在目录中创建的 VM 数量。创建计算机目录时，请选择您的主映像并注意以下事项：

- 在 Nutanix 独有的 **Container**（容器）页面上，选择您之前为模板 VM 指定的容器。
- 创建包含单会话操作系统计算机的目录时，会出现桌面体验页面，在该页面中，您可以决定用户每次登录时会发生什么。

The screenshot shows the 'Machine Catalog Setup' wizard. The left sidebar lists steps 1 through 16, with 'Desktop Experience' (step 4) highlighted. The main content area is titled 'Desktop Experience' and asks 'Which desktop experience do you want users to have?'. It offers two radio button options: 'I want users to connect to a new (random) desktop each time they log on.' (selected) and 'I want users to connect to the same (static) desktop each time they log on.'. Below this, it asks 'Do you want to save any changes that the user makes to the desktop?' with two radio button options: 'Yes, create a dedicated virtual machine and save changes on the local disk.' (selected) and 'No, discard all changes and clear virtual desktops when the user logs off.'. At the bottom of the wizard are three buttons: 'Back', 'Next', and 'Cancel'.

在桌面体验页面上，选择以下选项之一：

- 用户在每次登录时均会连接至一个新的（随机的）桌面。
- 用户每次登录时连接至同一个（静态）桌面。

如果选择第一个选项，用户对桌面所做的更改将被丢弃（非永久）。

如果您选择第二项并使用 MCS 来预配计算机，您可以对如何处理用户对桌面所做的更改进行配置：

- 在本地磁盘上保存用户对桌面的更改（永久）。
 - 在用户注销时放弃用户更改并清除虚拟桌面（非永久）。如果您使用的是用户个性化层，请选择此选项。
- 更新包含永久性计算机的 MCS 目录的主映像时，添加到该目录的任何新计算机都将使用更新后的映像。先前存在的计算机将继续使用原始主映像。

有关详细信息，请参阅 [Citrix Virtual Apps and Desktops](#) 文档和 [Citrix DaaS](#) 文档中的“计算机目录创建”。

注意：

对于 Nutanix 环境，如果 Delivery Controller 上的计算机目录创建过程需要大量时间，请转至 Nutanix Prism 并手动打开前缀为 **Preparation** 的计算机的电源。这种方法有助于继续执行创建过程。

步骤 5：创建交付组

交付组是从一个或多个计算机目录中选择的计算机的集合。它指定哪些用户可以使用这些计算机，以及可供这些用户使用的应用程序和桌面。

有关详细信息，请参阅 [Citrix Virtual Apps and Desktops](#) 文档和 [Citrix DaaS](#) 文档中的“交付组创建”。

注意：

使用 MCS 创建的 VM 可能无法在 Citrix Cloud Connector 中注册并显示为未注册。当您在 Azure 上托管 VM 并使用 Samba Winbind 加入 AD 域时，就会出现此问题。要解决此问题，请完成以下步骤：

1. 转至 ADSI 编辑控制台，选择未注册的 VM，然后编辑其计算机帐户的 **msDS-SupportedEncryptionTypes** 属性。
2. 在 VM 上重新启动 **ctxjproxy** 和 **ctxvda** 服务。如果 VM 的状态更改为已注册，请继续执行步骤 3 到 5。
3. 在模板 VM 上打开 **/var/xdl/mcs/ad_join.sh** 文件。
4. 在 **/var/xdl/mcs/ad_join.sh** 文件中的以下几行之后添加一行 **net ads entypes set \$NEW_HOSTNAME\$ < 加密类型属性的十进制值，例如 28> -U \$NEW_HOSTNAME\$ -P password:**

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

5. 创建新快照并使用新模板创建 VM。

使用 MCS 升级您的 Linux VDA

要使用 MCS 来升级 Linux VDA，请执行以下操作：

1. 在将您的 Linux VDA 升级到当前版本之前，请确保已安装.NET。

- 在除 Amazon Linux 2 之外的所有受支持的 Linux 发行版中安装.NET Runtime 8.0。
- 对于 Amazon Linux 2，请继续安装.NET Runtime 6.0。

如果您的 Linux 发行版包含所需的.NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装.NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

2. 在模板计算机上升级 Linux VDA:

注意:

- 此外，您还可以使用[通过 Azure 进行 Linux VDA 自助更新](#)功能来计划软件自动更新。要实现此目标，请在模板计算机上的 `etc/xdl/mcs/mcs_local_setting.reg` 文件中添加命令行。例如，可以添加以下命令行:

```
1 create -k "HKLM\System\CurrentControlSet\Control\Citrix\
   SelfUpdate" -t "REG_DWORD" -v "fEnabled" -d "0x00000001"
   --force
2
3 create -k "HKLM\System\CurrentControlSet\Control\Citrix\
   SelfUpdate" -t "REG_SZ" -v "ScheduledTime" -d "
   Immediately" --force
4
5 create -k "HKLM\System\CurrentControlSet\Control\Citrix\
   SelfUpdate" -t "REG_SZ" -v "Url" -d "<Your-Azure-
   Container-Url>" - force
6
7 create -k "HKLM\System\CurrentControlSet\Control\Citrix\
   SelfUpdate" -t "REG_SZ" -v "CaCertificate" -d "<Local-
   Certificate-Path-of-PortalAzureCom>" --force
```

- 从 2407 版本开始，Linux VDA 委派包管理器 **rpm** 或 **dpkg** 在升级期间处理配置文件。下面介绍了 rpm 和 dpkg 如何与配置文件更改交互:
 - **rpm**: 默认保留本地版本并使用 `rpmnew` 扩展名保存软件包中的新版本。
 - **dpkg**: 以交互方式提示您选择如何继续。要以静默方式升级 Linux VDA，同时保留本地配置文件并将新的软件包版本另存为 **.dpkg-new** 或 **.dpkg-dist**，请使用以下命令:

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

对于 **RHEL** 和 **Rocky Linux** 发行版:

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

注意：

在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 升级 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

对于 **SUSE** 发行版：

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

对于 **Ubuntu/Debian** 发行版：

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

3. 编辑 `/etc/xdl/mcs/mcs.conf` 和 `/etc/xdl/mcs/mcs_local_setting.reg`。
4. 生成新快照。
5. 在 Citrix Studio 中，选择用于更新计算机目录的新快照。在每台计算机重新启动之前，等待一段时间。请勿手动重新启动计算机。

自动更新计算机帐户密码

默认情况下，计算机帐户密码在创建计算机目录后 30 天过期。要防止密码过期以及自动更新计算机帐户密码，请执行以下操作：

1. 在运行 `/opt/Citrix/VDA/sbin/deploymcs.sh` 之前，将以下条目添加到 `/etc/xdl/mcs/mcs.conf` 中。

```
UPDATE_MACHINE_PW="Y"
```

2. 运行 `/opt/Citrix/VDA/sbin/deploymcs.sh` 后，打开 `/etc/cron.d/mcs_update_password_cronjob` 以设置更新时间和频率。默认设置每周星期日凌晨 2:30 更新计算机帐户密码。

每次更新计算机帐户密码后，Delivery Controller 上的票证缓存将变为无效，并且在 `/var/log/xdl/jproxy.log` 中可能会出现以下错误：

```
[ERROR] - AgentKerberosServiceAction.Run: GSSException occurred.
Error: Failure unspecified at GSS-API level (Mechanism level:
Checksum failed)
```

要消除错误，请定期清除票证缓存。可以在所有 Delivery Controller 或域控制器上安排缓存清理任务。

在 **MCS** 创建的 **VM** 上启用 **FAS**

可以在以下发行版上运行的 MCS 创建的虚拟机上启用 FAS：

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

在模板 VM 上准备主映像时启用 FAS

1. 导入根 CA 证书。

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

2. 运行轻松安装脚本 (ctxinstall.sh)，并确保模板 VM 已正确加入域。
3. 运行 `ctxfascfg.sh`。
4. 在 `/etc/xdl/mcs/mcs.conf` 中设置变量。

注意：

必须在 `/etc/xdl/mcs/mcs.conf` 中设置所有必需的变量，因为这些变量是在 VM 启动时调用的。

- a) 将 `Use_AD_Configuration_Files_Of_Current_VDA` 的值设置为 Y。
 - b) 根据需要设置其他变量，例如 `VDI_MODE`。
5. 将以下命令行添加到 `/etc/xdl/mcs/mcs_local_setting.reg` 文件中，用于设置 FAS 服务器地址：

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

6. 运行脚本 `/opt/Citrix/VDA/sbin/deploymcs.sh`。

在 MCS 创建的 VM 上启用 FAS

如果未如上文所述在模板计算机上启用 FAS，则可以在每个 MCS 创建的 VM 上启用 FAS。

要在 MCS 创建的 VM 上启用 FAS，请执行以下操作：

1. 在 `/etc/xdl/mcs/mcs.conf` 中设置变量。

注意：

必须在 `/etc/xdl/mcs/mcs.conf` 中设置所有必需的变量，因为这些变量是在 VM 启动时调用的。

- a) 将 `Use_AD_Configuration_Files_Of_Current_VDA` 的值设置为 Y。
 - b) 根据需要设置其他变量，例如 `VDI_MODE`。
2. 将以下命令行添加到 `/etc/xdl/mcs/mcs_local_setting.reg` 文件中，用于设置 FAS 服务器地址：

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

3. 导入根 CA 证书。

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

4. 运行脚本 `/opt/Citrix/VDA/sbin/ctxfascfg.sh`。

使用 Citrix Provisioning 创建 Linux VDA

August 20, 2024

可以使用 Citrix Provisioning 创建加入域的 VDA。

本文提供有关通过流技术推送 Linux 目标设备的信息。使用此功能，您可以直接在 Citrix Virtual Apps and Desktops 环境中预配 Linux 虚拟桌面。

支持以下 Linux 发行版：

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

重要：

- 我们建议您使用 Citrix Provisioning 的最新安装包。请根据您的 Linux 发行版使用该软件包。要使用 Linux 流技术推送代理 2109 及更高版本，需要安装 Citrix Provisioning 服务器 2109 或更高版本。
- 使用 Citrix Provisioning 通过流技术推送 Linux 目标设备时，请对单个共享磁盘映像创建单独的启动分

区，以便预配的设备能够正常启动。

- 请避免使用 **btrfs** 格式化任何分区。GRUB2 在查找 **btrfs** 分区时出现内部问题。**GRUB** 代表 **GRand Unified Bootloader**。

有关详细信息，请参阅 Citrix Provisioning 文档中的[通过流技术推送 Linux 目标设备](#)。

在 **Citrix DaaS Standard for Azure** 中创建 **Linux VDA**

August 21, 2024

可以在 Citrix DaaS Standard for Azure（以前称为“适用于 Azure 的 Citrix Virtual Apps and Desktops Standard”）中创建已加入域和未加入域的 Linux VDA，以将虚拟应用程序和桌面从 Microsoft Azure 交付到任何设备。有关详细信息，请参阅 [Citrix DaaS Standard for Azure](#)。

支持的 **Linux** 发行版

以下 Linux 发行版支持此功能：

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

步骤 1：在 **Azure** 中准备主映像

注意：

此外，您还可以使用 [Linux VDA 自助更新](#) 功能来计划软件自动更新。要实现此目标，请在主映像上的

etc/xdl/mcs/mcs_local_setting.reg 文件中添加命令行。

例如，可以添加以下命令行：

```
1 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
   -t "REG_DWORD" -v "fEnabled" -d "0x00000001" --force
2
3 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
   -t "REG_SZ" -v "ScheduledTime" -d "Immediately" --force
4
5 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
   -t "REG_SZ" -v "Url" -d "<Your-Azure-Container-Url>" --force
6
7 create -k "HKLM\System\CurrentControlSet\Control\Citrix\SelfUpdate"
   -t "REG_SZ" -v "CaCertificate" -d "<Local-Certificate-Path-of-
   PortalAzureCom>" --force
```

1. 在 Azure 中，创建支持的发行版的 Linux VM。
2. 如有必要，请在 Linux VM 上安装桌面环境。
3. 在 VM 上，根据您的 Linux 发行版安装.NET。

除.NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装.ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的.NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装.NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

4. (仅限 Ubuntu) 在 `/etc/network/interfaces` 文件中添加 `source /etc/network/interfaces.d/*` 行。
5. (仅限 Ubuntu) 请将 `/etc/resolv.conf` 指向 `/run/systemd/resolve/resolv.conf`，而非将其指向 `/run/systemd/resolve/stub-resolv.conf`：

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

6. 安装 Linux VDA 软件包。
7. 指定要使用的数据库。

作为一项实验性功能，除了 PostgreSQL 外，您还可以使用 SQLite。安装 Linux VDA 软件包后，您还可以在 SQLite 与 PostgreSQL 之间切换。为此，请完成以下步骤：

- a) 运行 `/opt/Citrix/VDA/sbin/ctxcleanup.sh`。如果是全新安装，则省略此步骤。
- b) 请在运行 `deploymcs.sh` 前编辑 `/etc/xdl/db.conf`。

注意：

- 我们建议您仅对 VDI 模式使用 SQLite。
- 对于轻松安装和 MCS，您可以在 SQLite 与 PostgreSQL 之间切换，而不必手动安装。除非通过 `/etc/xdl/db.conf` 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 也可以使用 `/etc/xdl/db.conf` 来配置 PostgreSQL 的端口号。

8. 更改 MCS 变量。

配置 MCS 变量有两种方法：

- 编辑 `/etc/xdl/mcs/mcs.conf` 文件。
- 使用轻松安装 GUI。要打开轻松安装 GUI，请在您的 Linux VDA 的桌面环境中运行 `/opt/Citrix/VDA/bin/easyinstall` 命令。

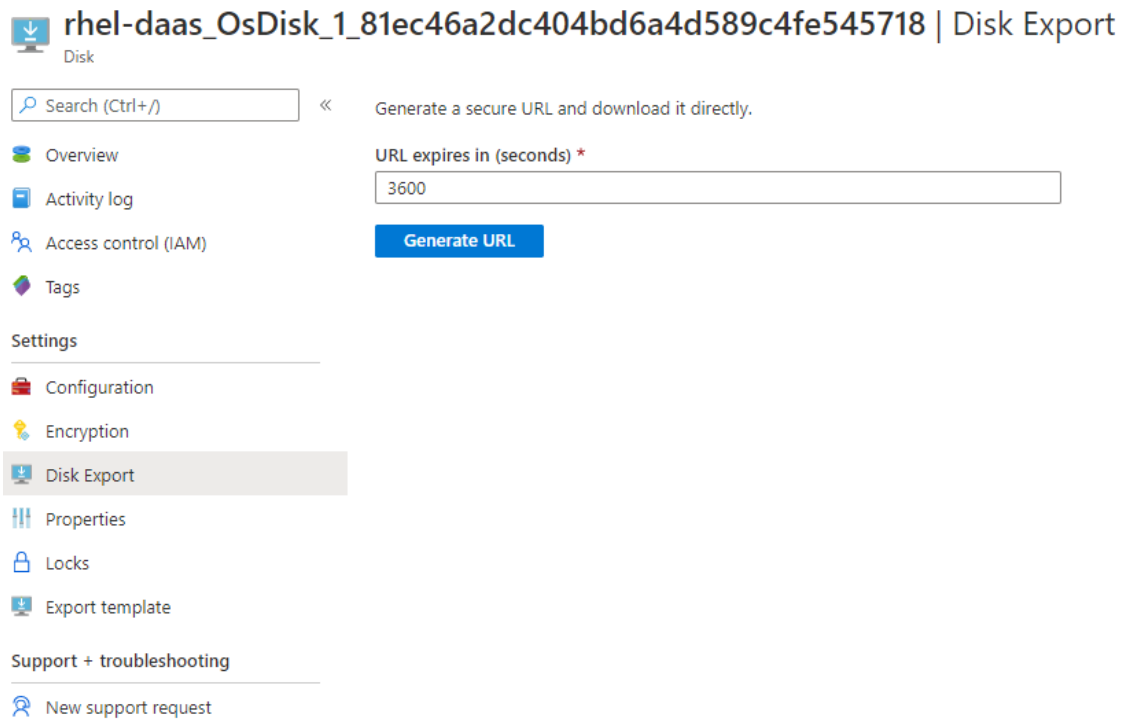
注意：

请将 `dns` 变量保留为未指定。

如果在创建计算机目录时选择静态或随机类型，请设置 `VDI_MODE=Y`。

如果您通过编辑 `/etc/xdl/mcs/mcs.conf` 配置 MCS 变量，请运行 `/opt/Citrix/VDA/sbin/deploymcs.sh`。如果您使用 GUI 配置 MCS 变量，请单击部署。在 GUI 上单击部署后，在 GUI 上设置的变量将覆盖您在 `/etc/xdl/mcs/mcs.conf` 文件中设置的变量。

9. 在 Azure 中，停止（或取消分配）VM。单击磁盘导出为虚拟硬盘 (VHD) 文件生成 SAS URL，您可以将该文件用作主映像来创建其他 VM。



10. (可选) 在主映像上配置组策略设置。您可以使用 `ctxreg` 工具来配置组策略设置。例如，以下命令将为 PDF 打印启用自动创建 **PDF** 通用打印机策略。

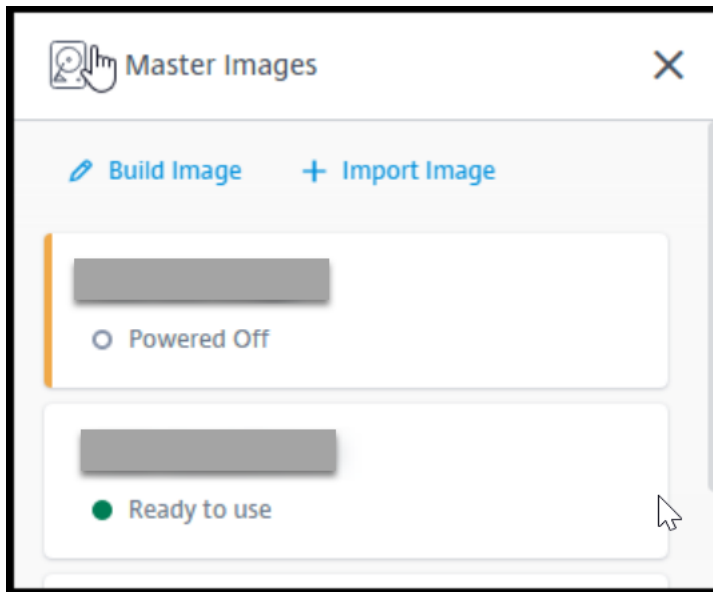
```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\GroupPolicy\Defaults\PrintingPolicies" -t "REG_DWORD" -v "AutoCreatePDFPrinter" -d "0x00000001" --force
```

步骤 2: 从 **Azure** 导入主映像

1. 在**管理**控制板中，展开右侧的主映像。显示内容将列出 Citrix 提供的主映像以及您创建和导入的映像。

提示：

此服务的大多数管理员活动都通过管理和监视控制板进行管理。创建第一个目录后，管理控制板将在登录到 Citrix Cloud 并选择 **Managed Desktops**（托管桌面）服务后自动启动。



2. 单击导入映像。
3. 输入您在 Azure 中生成的 VHD 文件的 SAS URL。选择 **Linux** 作为主映像类型。

Import Image from Azure

Enter the Azure-generated URL for the Virtual Hard Disk ?

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

Master image type

- Windows
 Linux

Name The New Master Image

E.g. "Windows 10 + My Apps"

4. 按照向导中的说明完成导入主映像的操作。

步骤 3：创建计算机目录

访问[管理](#)控制板，然后单击创建目录。创建计算机目录时，请选择之前创建的主映像。

注意：

用作主映像的 VM 无法通过 SSH 或 RDP 访问。要访问 VM，请使用 Azure 门户中的串行控制台。

手动安装 **Linux VDA**

August 20, 2024

可以在以下 Linux 发行版上手动安装 Linux VDA:

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

在 **Amazon Linux 2**、**RHEL** 和 **Rocky Linux** 上手动安装 **Linux VDA**

August 20, 2024

重要:

对于全新安装，我们建议您使用[轻松安装](#)进行快速安装。轻松安装省时又省力，与本文中详述的手动安装相比，更不易于出错。

步骤 **1**: 准备配置信息和 **Linux** 计算机

步骤 **1a**: 验证网络配置

确保已连接并正确配置网络。例如，必须在 Linux VDA 上配置 DNS 服务器。

步骤 **1b**: 设置主机名

为确保正确报告计算机的主机名，请更改 `/etc/hostname` 文件，使其仅包含计算机的主机名。

`hostname`

步骤 **1c**: 为主机名分配环回地址

为确保正确报告计算机的 DNS 域名和完全限定域名 (FQDN)，请更改 `/etc/hosts` 文件中的以下行，使其前两个条目为 FQDN 和主机名：

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

例如：

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

从文件中的其他条目中删除对 **hostname-fqdn** 或 **hostname** 的任何其他引用。

注意：

Linux VDA 当前不支持 NetBIOS 名称截断。主机名不得超过 15 个字符。

提示：

只能使用字符 a-z、A-Z、0-9 和连字符 (-)。请避免使用下划线 (_)、空格和其他符号。主机名不得以数字开头和以连字符结尾。此规则也适用于 Delivery Controller 主机名。

步骤 1d: 检查主机名

验证主机名设置是否正确无误：

```
1 hostname
```

此命令仅返回计算机的主机名，而不返回其完全限定域名 (FQDN)。

验证 FQDN 设置是否正确无误：

```
1 hostname -f
```

此命令返回计算机的 FQDN。

步骤 1e: 检查名称解析和服务可访问性

确认可以解析 FQDN 并对域控制器和 Delivery Controller 执行 ping 操作：

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

如果无法解析 FQDN 或 Ping 不通上述任一计算机，请先检查相关步骤，然后再继续。

步骤 1f: 配置时钟同步

确保 VDA、Delivery Controller 和域控制器之间的时钟始终精确同步至关重要。将 Linux VDA 托管为虚拟机 (VM) 可能会导致时钟偏差问题。出于此原因，最好使用远程时间服务来同步时间。

RHEL 默认环境使用 Chrony 守护程序 (`chronyd`) 进行时钟同步。

配置 **Chrony** 服务 以 root 用户身份，编辑 `/etc/chrony.conf` 并为每个远程时间服务器添加一个服务器条目：

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

在典型部署中，时间从本地域控制器同步，而不是直接从公共 NTP 池服务器同步。为域中的每个 Active Directory 域控制器添加一个服务器条目。

删除列出的任何其他服务器条目，包括环回 IP 地址、localhost 以及公共服务器 `*.pool.ntp.org` 条目。

保存更改并重新启动 Chrony 守护程序：

```
1 sudo systemctl restart chronyd
```

步骤 1g: 安装 OpenJDK 11

Linux VDA 要求存在 OpenJDK 11。

- 如果使用的是 RHEL，则当您安装 Linux VDA 时，OpenJDK 11 会自动作为依赖项进行安装。
- 如果使用的是 Amazon Linux 2 或 Rocky Linux，请运行以下命令以启用并安装 OpenJDK 11：

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

确认版本是否正确：

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

预先封装的 OpenJDK 可能为早期版本。更新到 OpenJDK 11：

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

步骤 1h: 安装并指定要使用的数据库

注意：

- 我们建议您仅将 SQLite 用于 VDI 模式，将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS，您可以指定使用 SQLite 或 PostgreSQL，而不必手动安装。除非通过 `/etc/xdl/db.conf` 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 对于手动安装，您必须手动安装 SQLite、PostgreSQL 或者两者。如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 `/etc/xdl/db.conf` 来指定使用其中一个。

本部分内容介绍如何安装 PostgreSQL 和 SQLite 以及如何指定要使用的两者中的任意一个。

安装 **PostgreSQL** Linux VDA 需要 PostgreSQL:

- 适用于 Amazon Linux 2 的 PostgreSQL 9
- 适用于 RHEL 8.x 和 Rocky Linux 8.x 的 PostgreSQL 10
- 适用于 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 的 PostgreSQL 13

请运行以下命令以安装 PostgreSQL:

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

对于 RHEL 8.x 和 RHEL 9.4/9.3/9.2, 请运行以下命令以安装适用于 PostgreSQL 的 **libpq**:

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

请运行以下命令以初始化数据库。此操作会在 **/var/lib/pgsql/data** 下创建数据库文件。

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

请分别运行以下命令以在计算机启动时启动 PostgreSQL 或者立即启动:

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl start postgresql
```

请使用以下命令检查 PostgreSQL 版本:

```
1 psql --version
```

(仅限 Amazon Linux 2) 使用 **psql** 命令行实用程序确认数据目录已设置:

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

安装 **SQLite** 请运行以下命令以安装 SQLite:

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

指定要使用的数据库 如果您同时安装了 SQLite 和 PostgreSQL, 则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

1. 运行 **/opt/Citrix/VDA/sbin/ctxcleanup.sh**。如果是全新安装, 则省略此步骤。
2. 编辑 **/etc/xdl/db.conf** 以指定要使用的数据库。
3. 运行 **ctxsetup.sh**。

注意：

也可以使用 `/etc/xdl/db.conf` 配置 PostgreSQL 的端口号。

步骤 2：准备虚拟机管理程序

在支持的虚拟机管理程序上将 Linux VDA 当作 VM 运行时，需要作出一些更改。根据正在使用的虚拟机管理程序平台做出以下更改。如果正在裸机硬件上运行 Linux 计算机，则无需作出任何更改。

修复 XenServer（以前称为 Citrix Hypervisor）上的时间同步问题

启用了 XenServer 时间同步功能时，在每个半虚拟化 Linux VM 中，您会遇到 NTP 和 XenServer 出现的问题。两者都试图管理系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。无需在 HVM 模式下进行任何更改。

如果您正在运行半虚拟化 Linux 内核，并且安装了 XenServer Tools，则可以检查 XenServer 时间同步功能是否存在以及是否已从 Linux VM 中启用：

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

此命令返回 0 或 1：

- 0 - 时间同步功能已启用，且必须禁用。
- 1 - 时间同步功能已禁用，无需采取任何操作。

如果 `/proc/sys/xen/independent_wallclock` 文件不存在，则不需要执行以下步骤。

如果已启用，请通过向该文件写入 1 来禁用时间同步功能：

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

要使此更改成为永久更改，并在重新启动后仍然有效，请编辑 `/etc/sysctl.conf` 文件并添加以下行：

```
xen.independent_wallclock = 1
```

要验证这些更改，请重新启动系统：

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

此命令返回值 1。

修复 **Microsoft Hyper-V** 上的时间同步问题

安装了 Hyper-V Linux 集成服务的 Linux VM 可应用 Hyper-V 时间同步功能来使用主机操作系统的时间。为确保系统时钟始终精确可靠，必须一同启用此功能与 NTP 服务。

从管理操作系统中：

1. 打开 Hyper-V 管理器控制台。
2. 对于 Linux VM 的设置，请选择 **Integration Services**（集成服务）。
3. 确保已选择 **Time synchronization**（时间同步）。

注意：

此方法与 VMware 和 XenServer（以前称为 Citrix Hypervisor）不同，这两款产品会禁用主机时间同步功能以避免与 NTP 发生冲突。Hyper-V 时间同步可以与 NTP 时间同步共存并互补。

修复 **ESX** 和 **ESXi** 上的时间同步问题

启用 VMware 时间同步功能后，在每个半虚拟化的 Linux VM 中，您都会遇到 NTP 和虚拟机管理程序的问题。两者都试图同步系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。

如果正在运行安装了 VMware Tools 的半虚拟化 Linux 内核：

1. 打开 vSphere Client。
2. 编辑 Linux VM 设置。
3. 在 **Virtual Machine Properties**（虚拟机属性）对话框中，打开 **Options**（选项）选项卡。
4. 选择 **VMware Tools**。
5. 在 **Advanced**（高级）框中，取消选中 **Synchronize guest time with host**（与主机同步客户机时间）。

步骤 3：将 **Linux VM** 添加到 **Windows** 域中

以下方法可用于将 Linux 计算机添加到 Active Directory (AD) 域：

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

根据所选的方法，按说明执行操作。

注意：

为 Linux VDA 中的本地帐户和 AD 中的帐户使用相同的用户名时，会话启动可能会失败。

Samba Winbind

对于 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2，请运行以下命令以防止 **pam_winbind** 更改根目录的所有权：

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

安装或更新所需软件包：

对于 RHEL 9.4/9.3/9.2/8.x 和 Rocky Linux 9.4/9.3/9.2/8.x：

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

对于 Amazon Linux 2：

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

在计算机启动时启用要启动的 **Winbind** 守护程序 Winbind 守护程序必须配置为在计算机启动时启动：

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

配置 **Winbind** 身份验证 通过使用 Winbind 将计算机配置为执行 Kerberos 身份验证：

1. 请运行以下命令：

对于 RHEL 9.4/9.3/9.2/8.x 和 Rocky Linux 9.4/9.3/9.2/8.x：

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

对于 Amazon Linux 2：

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

其中，**REALM** 是大写的 Kerberos 领域名称，而 **domain** 是域的 NetBIOS 名称。

如果需要通过 DNS 查找 KDC 服务器和领域名称，请将以下两个选项添加至前面的命令：

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

请忽略 `authconfig` 命令返回的有关 `winbind` 服务无法启动的任何错误。`authconfig` 尝试在计算机尚未加入域的情况下启动 `winbind` 服务时，可能会出现这些错误。

2. 打开 `/etc/samba/smb.conf` 并将以下条目添加到 `[Global]` 部分下方，但要放在 `authconfig` 工具生成的部分后面：

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

3. (仅限 RHEL 9.4/9.3/9.2/8.x 和 Rocky Linux 9.4/9.3/9.2/8.x) 打开 `/etc/krb5.conf` 并在 `[libdefaults]`、`[realms]` 和 `[domain_realm]` 部分下添加条目：

在 `[libdefaults]` 部分下：

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

在 `[realms]` 部分下：

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

在 `[domain_realm]` 部分下：

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

Linux VDA 需要使用系统 keytab 文件 `/etc/krb5.keytab` 以执行身份验证并向 Delivery Controller 注册。计算机首次加入域后，前面的 `kerberos method` 设置将强制 Winbind 创建系统 keytab 文件。

加入 **Windows** 域 您的域控制器必须可访问，而且您必须具有有权将计算机添加到域的 Active Directory 用户帐户。

要将 Linux VM 添加到 Windows 域中，请运行以下命令：

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

提示：

对于在 Amazon Linux 2 上运行的 Linux VM，您也可以使用以下命令将其添加到 Windows 域中：

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

REALM 是大写的 Kerberos 领域名称，**user** 是有权将计算机添加到域的域用户。

为 **Winbind** 配置 **PAM** 默认情况下，Winbind PAM 模块 (pam_winbind) 的配置不启用 Kerberos 票据缓存和主目录的创建。打开 **/etc/security/pam_winbind.conf**，并在 [Global] 部分下添加或更改以下条目：

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

确保删除每个设置中的任何前置分号。这些更改要求重新启动 Winbind 守护程序：

```
1 sudo systemctl restart winbind
```

提示：

仅当计算机加入域后，winbind 守护程序才会始终保持运行。

打开 **/etc/krb5.conf** 并将 [libdefaults] 部分下方的以下设置从 KEYRING 更改为 FILE 类型：

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

对于 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2，请运行以下命令来解决 Winbind 上的 SELinux 问题：

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

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 **Active Directory** 中有一个计算机对象。

运行 **Samba** 的 **net ads** 命令以验证计算机是否已加入域：

```
1 sudo net ads testjoin
```

运行以下命令验证额外的域和计算机对象信息：

```
1 sudo net ads info
```

验证 Kerberos 配置 为了确保 Kerberos 已正确配置为可与 Linux VDA 配合使用，请验证系统 keytab 文件是否已创建并且包含有效密钥：

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令，使用这些密钥向域控制器验证计算机的身份：

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义，以免发生 shell 替换。在某些环境中，DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行，则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存：

```
1 sudo klist
```

请使用以下命令检查计算机的帐户详细信息：

```
1 sudo net ads status
```

验证用户身份验证 使用 **wbinfo** 工具验证是否可向域验证域用户的身份：

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

这里指定的域为 AD 域名，而不是 Kerberos 领域名称。对于 bash shell，必须使用另一个反斜杠对反斜杠 (\) 字符进行转义。此命令返回一条成功或失败消息。

要验证 Winbind PAM 模块是否已正确配置，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证 Kerberos 凭据缓存中的票据是否有效且未过期：

```
1 klist
```

退出会话。

```
1 exit
```

直接登录 GNOME 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6：安装 Linux VDA](#)。

Quest Authentication Services

在域控制器上配置 **Quest** 假定您已在 Active Directory 域控制器上安装并配置了 Quest 软件，而且已获得管理权限，有权在 Active Directory 中创建计算机对象。

允许域用户登录 **Linux VDA** 计算机 为了让域用户能够在 Linux VDA 计算机上建立 HDX 会话：

1. 在 Active Directory 用户和计算机管理控制台中，为该用户帐户打开 Active Directory 用户属性。
2. 选择 **Unix Account** (Unix 帐户) 选项卡。
3. 选中 **Unix-enabled** (已启用 Unix)。
4. 将 **Primary GID Number** (首选 GID 编号) 设置为实际域用户组的组 ID。

注意：

这些说明相当于设置域用户，以便他们可以使用控制台、RDP、SSH 或任何其他远程协议进行登录。

在 Linux VDA 上配置 Quest

SELinux 策略强制实施解决方法 默认 RHEL 环境会强制实施 SELinux。此强制功能会影响 Quest 使用的 Unix 域套接字 IPC 机制，并阻止域用户登录。

解决此问题的最便捷的方法是禁用 SELinux。以 root 用户身份，编辑 `/etc/selinux/config` 并更改 **SELinux** 设置：

`SELINUX=permissive`

此更改要求重新启动计算机：

```
1 reboot
```

重要：

请谨慎使用此设置。禁用后重新启用 SELinux 策略强制实施会导致完全锁定，即便是对 root 用户和其他本地用户也是如此。

配置 VAS 守护程序 必须启用并断开自动续订 Kerberos 票据功能。必须禁用身份验证（脱机登录）功能。

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

此命令将续订间隔设为 9 小时（32400 秒），即比默认的 10 小时票据生命周期短 1 小时。请在票据生命周期较短的系统上设置较低的值。

配置 PAM 和 NSS 要启用通过 HDX 进行的域用户登录以及其他服务（例如 su、ssh 和 RDP），请运行以下命令以手动配置 PAM 和 NSS：

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

加入 Windows 域 使用 Quest **vastool** 命令将 Linux 计算机加入到 Active Directory 域中：

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

`user` 为有权将计算机加入 Active Directory 域的任何域用户。**domain-name** 为域的 DNS 名称，例如 `example.com`。

加入域后重新启动 Linux 计算机。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 **Active Directory** 中有一个计算机对象。要验证加入了 Quest 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入域，此命令会返回域名。如果计算机未加入任何域，则会显示以下错误：

```
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
```

验证用户身份验证 要验证 Quest 是否能够通过 PAM 对域用户进行身份验证，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：

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

验证 Kerberos 凭据缓存中的票据是否有效且未过期：

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

退出会话。

```
1 exit
```

直接登录 GNOME 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

Centrify DirectControl

加入 Windows 域 安装 Centrify DirectControl Agent 后，请使用 Centrify `adjoin` 命令将 Linux 计算机加入 Active Directory 域：

```
1 su -
2 adjoin -w -V -u user domain-name
```

`user` 参数为有权将计算机加入 Active Directory 域的任何 Active Directory 域用户。**domain-name** 是将 Linux 计算机加入到的域的名称。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 Active Directory 中有一个计算机对象。要验证加入了 Centrify 的 Linux 计算机是否位于域中, 请执行以下操作:

```
1 su -
2 adinfo
```

验证 `Joined to domain` 值是否有效以及 `CentrifyDC mode` 是否返回了 `connected`。如果模式仍然卡在正在启动状态, 则表明 Centrify 客户端遇到了服务器连接或身份验证问题。

请使用以下命令可获得更全面的系统和诊断信息:

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

测试与各种 Active Directory 和 Kerberos 服务的连接。

```
1 adinfo --test
```

在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

SSSD

如果您使用的是 SSSD, 请按照此部分中的说明进行操作。此部分包含有关如何将 Linux VDA 计算机加入 Windows 域的说明以及如何配置 Kerberos 身份验证的指导。

要在 RHEL 上设置 SSSD, 请执行以下操作:

1. 加入域并创建主机 `keytab`
2. 设置 SSSD
3. 启用 SSSD
4. 验证 Kerberos 配置
5. 验证用户身份验证

加入域并创建主机 **keytab** SSSD 并不提供用于加入域和管理系统 `keytab` 文件的 Active Directory 客户端功能。可以改为使用 **adcli**、**realmd** 或 **Samba**。

本部分内容介绍了适用于 Amazon Linux 2 的 **Samba** 方法以及适用于 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 的 **adcli** 方法。对于 **realmd**, 请参阅 RHEL 文档。必须在配置 SSSD 之前执行这些步骤。

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

安装或更新所需软件包:

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

在正确配置了以下文件的 Linux 客户端上:

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

将计算机配置为进行 **Samba** 和 Kerberos 身份验证:

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

其中, **REALM** 是大写的 Kerberos 领域名称, **domain** 是 Active Directory 域的简短 NetBIOS 名称。

注意:

本文中的设置适用于单域、单林模型。根据您的 AD 基础结构配置 Kerberos。

如果需要通过 DNS 查找 KDC 服务器和领域名称, 请将以下两个选项添加至前面的命令:

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

打开 **/etc/samba/smb.conf** 并将以下条目添加到 **[Global]** 部分下方, 但要放在 **authconfig** 工具生成的部分后面:

```
kerberos method = secrets and keytab
winbind offline logon = no
```

加入 Windows 域。请确保域控制器可访问, 并且您具有有权将计算机添加到域的 Active Directory 用户帐户:

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

REALM 是大写的 Kerberos 领域名称, **user** 是有权将计算机添加到域的域用户。

- **Adcli (RHEL 9.4/9.3/9.2/8.x 和 Rocky Linux 9.4/9.3/9.2/8.x):**

安装或更新所需软件包:

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

将计算机配置为进行 **Samba** 和 Kerberos 身份验证:

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

打开 **/etc/krb5.conf** 并在 [realms] 和 [domain_realm] 部分下添加条目。

在 [realms] 部分下:

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

在 [domain_realm] 部分下:


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

加入 Windows 域。请确保域控制器可访问，并且您具有有权将计算机添加到域的 Active Directory 用户帐户：

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

REALM 是大写的 Kerberos 领域名称，**user** 是有权将计算机添加到域的域用户。

设置 **SSSD** 设置 SSSD 的步骤如下：

- 通过运行 `sudo yum -y install sssd` 命令在 Linux VDA 上安装 **sssd-ad** 软件包。
- 对各种文件（例如 `sssd.conf`）进行配置更改。
- 启动 **sssd** 服务。

(仅限 RHEL 9.4/9.3/9.2/8.x 和 Rocky Linux 9.4/9.3/9.2/8.x)

打开 `/etc/sss/sss.conf` 并在 `[domain/ad.example.com]` 部分下添加以下条目：

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

将 **ad.example.com**、**server.ad.example.com** 替换为相应的值。有关详细信息，请参阅 [sssd-ad\(5\) - Linux 手册页](#)。

对 `sssd.conf` 设置文件所有权和权限：

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

启用 **SSSD** 对于 **RHEL 9.4/9.3/9.2/8.x** 和 **Rocky Linux 9.4/9.3/9.2/8.x**：

请运行以下命令以启用 SSSD：

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

对于 **Amazon Linux 2**：

使用 **authconfig** 启用 SSSD。安装 **oddjob mkhomedir** 以确保主目录创建与 SELinux 兼容：

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

验证 **Kerberos** 配置 验证系统 **keytab** 文件是否已创建并且包含有效密钥:

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令，以使用这些密钥向域控制器验证计算机的身份:

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (****) 进行转义，以免发生 shell 替换。在某些环境中，DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行，则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存:

```
1 sudo klist
```

验证用户身份验证 使用 **getent** 命令确认支持的登录格式以及 NSS 是否工作:

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

DOMAIN 参数指示简短形式的域名。如果需要使用另一种登录格式，请先使用 **getent** 命令进行验证。

支持的登录格式如下:

- 低级别登录名称: **DOMAIN\username**
- UPN: **username@domain.com**
- NetBIOS 前缀格式: **username@DOMAIN**

要验证 SSSD PAM 模块是否已正确配置，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为以下命令返回的 **uid** 创建了对应的 Kerberos 凭据缓存文件:

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

验证用户的 Kerberos 凭据缓存中的票据是否有效且未过期。

```
1 klist
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

PBIS

下载所需的 **PBIS** 软件包

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

使 **PBIS** 安装脚本可执行

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

运行 **PBIS** 安装脚本

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

加入 **Windows** 域 您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户：

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

user 为有权将计算机添加到 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称，例如 example.com。

注意：要将 Bash 设置为默认 shell，请运行 **/opt/pbis/bin/config LoginShellTemplate/bin/bash** 命令。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 Active Directory 中有一个计算机对象。要验证加入了 PBIS 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入某个域，此命令将返回有关当前加入的 AD 域和 OU 的信息。否则，仅显示主机名。

验证用户身份验证 要验证 PBIS 是否能够通过 PAM 对域用户进行身份验证，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

步骤 4: 安装 .NET

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

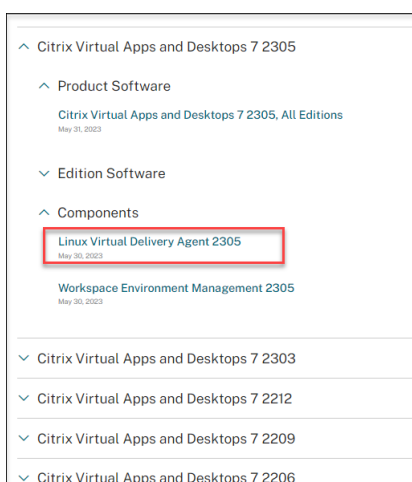
如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装 .NET 后，请运行 **which dotnet** 命令查找您的运行时路径。

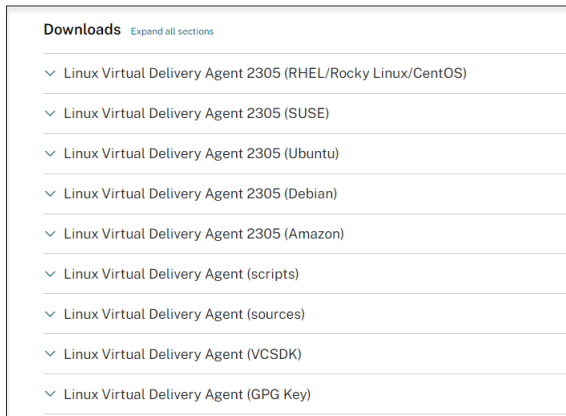
根据命令输出，设置 .NET Runtime 二进制文件路径。例如，如果命令输出为 /aa/bb/dotnet，请使用 /aa/bb 作为 .NET 二进制文件路径。

步骤 5: 下载 Linux VDA 软件包

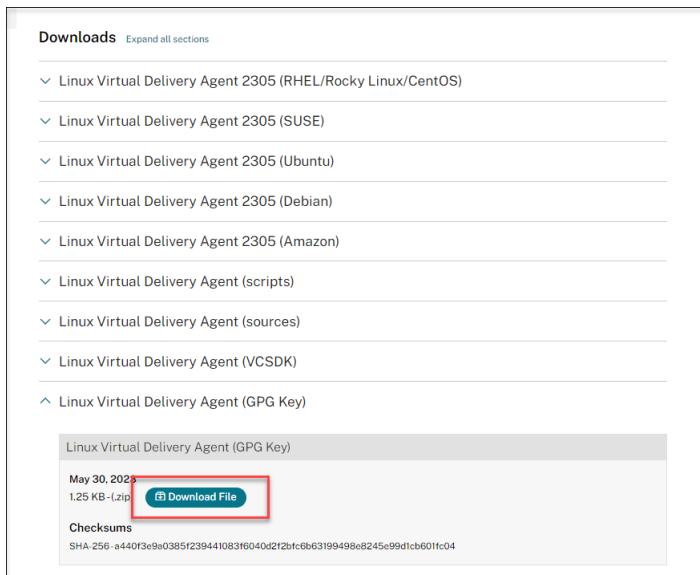
1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。
3. 展开组件以查找 Linux VDA。例如：



4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。
6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要验证 Linux VDA 软件包的完整性，请运行以下命令以将公钥导入到 RPM 数据库中以及检查软件包的完整性：

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

步骤 6: 安装 Linux VDA

您可以进行全新安装，也可以升级现有安装。Linux VDA 支持从最新版本进行升级。例如，您可以将 Linux VDA 从 2308 升级到 2311，从 1912 LTSR 升级到 2203 LTSR。

步骤 6a: 执行全新安装

1. (可选) 卸载旧版本

如果安装了除之前的两个版本和 LTSR 版本之外的早期版本，请在安装新版本之前将其卸载。

a) 停止 Linux VDA 服务：

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

注意：

在停止 **ctxvda** 和 **ctxhdx** 服务之前，请运行 **systemctl stop ctxmonitord** 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

b) 卸载软件包：

```
1 sudo rpm -e XenDesktopVDA
```

注意：

要运行命令，需要提供完整路径；或者，也可以将 **/opt/Citrix/VDA/sbin** 和 **/opt/Citrix/VDA/bin** 添加到系统路径。

2. 下载 Linux VDA 软件包

转至 [Citrix Virtual Apps and Desktops 下载页面](#)。展开相应版本的 Citrix Virtual Apps and Desktops，然后单击组件以下载与 Linux 发行版匹配的 Linux VDA 包。

3. 安装 Linux VDA

注意：

- 对于 RHEL 和 Rocky Linux，请先安装 EPEL 存储库，然后才能成功安装 Linux VDA。有关如何安装 EPEL 的信息，请参阅 <https://docs.fedoraproject.org/en-US/epel/> 上提供的说明。
- 在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上安装 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。

- 使用 **Yum** 安装 Linux VDA 软件：

对于 **Amazon Linux 2**：

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

对于 **RHEL 9.4/9.3/9.2** 和 **Rocky Linux 9.4/9.3/9.2**：

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

对于 **RHEL 8.x** 和 **Rocky Linux 8.x**：

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

- 使用 RPM 软件包管理器安装 Linux VDA 软件。在此之前，您必须解决以下依赖项：

对于 **Amazon Linux 2**：

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

RHEL 9.4/9.3/9.2 和 **Rocky Linux 9.4/9.3/9.2**：

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

对于 **RHEL 8.x** 和 **Rocky Linux 8.x**：

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

RHEL 9.4/9.3/9.2 和 **Rocky Linux 9.4/9.3/9.2** 的的 RPM 依赖项列表：

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

```
39 util-linux-user >= 2.32.1
40
41 xorg-x11-utils >= 7.5
42
43 bash >= 4.3
44
45 findutils >= 4.6
46
47 gawk >= 4.2
48
49 sed >= 4.5
50
51 cups >= 1.6.0
52
53 cups-filters >= 1.20.0
54
55 ghostscript >= 9.25
56
57 libxml2 >= 2.9
58
59 libmspack >= 0.7
60
61 ibus >= 1.5
62
63 nss-tools >= 3.44.0
64
65 chkconfig >= 1.20
66
67 cyrus-sasl-gssapi >= 2.1
68
69 python3 >= 3.6~
70
71 qt5-qtbase >= 5.5~
72
73 qt5-qtbase-gui >= 5.5~
74
75 qrencode-libs >= 3.4.4
76
77 imlib2 >= 1.4.9
78
79 fuse-libs >= 2.9
80
81 pulseaudio-utils >= 15.0
```

RHEL 8.x 和 Rocky Linux 8.x 的 RPM 依赖项列表:

```
1 java-11-openjdk >= 11
2
3 icoutils >= 0.32
4
5 firewalld >= 0.6.3
6
7 polycoreutils-python >= 2.8.9
```



```
8
9  polycoreutils-python-utils >= 2.8
10
11 python3-polycoreutils >= 2.8
12
13 dbus >= 1.12.8
14
15 dbus-common >= 1.12.8
16
17 dbus-daemon >= 1.12.8
18
19 dbus-tools >= 1.12.8
20
21 dbus-x11 >= 1.12.8
22
23 xorg-x11-server-utils >= 7.7
24
25 xorg-x11-xinit >= 1.3.4
26
27 libXpm >= 3.5.12
28
29 libXrandr >= 1.5.1
30
31 libXtst >= 1.2.3
32
33 pam >= 1.3.1
34
35 util-linux >= 2.32.1
36
37 util-linux-user >= 2.32.1
38
39 xorg-x11-utils >= 7.5
40
41 bash >= 4.3
42
43 findutils >= 4.6
44
45 gawk >= 4.2
46
47 sed >= 4.5
48
49 cups >= 1.6.0
50
51 foomatic-filters >= 4.0.9
52
53 cups-filters >= 1.20.0
54
55 ghostscript >= 9.25
56
57 libxml2 >= 2.9
58
59 libmspack >= 0.7
60
```

```
61 krb5-workstation >= 1.13
62
63 ibus >= 1.5
64
65 nss-tools >= 3.44.0
66
67 gperftools-libs >= 2.4
68
69 cyrus-sasl-gssapi >= 2.1
70
71 python3 >= 3.6~
72
73 qt5-qtbase >= 5.5~
74
75 qt5-qtbase-gui >= 5.5~
76
77 qrencode-libs >= 3.4.4
78
79 imlib2 >= 1.4.9
```

Amazon Linux 2 的 RPM 依赖项列表:

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

```
32
33 gawk >= 4.0
34
35 sed >= 4.2
36
37 cups >= 1.6.0
38
39 foomatic-filters >= 4.0.9
40
41 libxml2 >= 2.9
42
43 libmspack >= 0.5
44
45 ibus >= 1.5
46
47 cyrus-sasl-gssapi >= 2.1
48
49 gperftools-libs >= 2.4
50
51 nss-tools >= 3.44.0
52
53 qt5-qtbase >= 5.5~
54
55 qrencode-libs >= 3.4.1
56
57 imlib2 >= 1.4.5
```

注意：

有关此版本的 Linux VDA 支持的 Linux 发行版和 Xorg 版本的列表，请参阅[系统要求](#)。

步骤 6b: 升级现有安装 (可选)

Linux VDA 支持从最新版本进行升级。例如，您可以将 Linux VDA 从 2308 升级到 2311，从 1912 LTSR 升级到 2203 LTSR。

对于 **Amazon Linux 2**、**RHEL** 和 **Rocky Linux** 发行版：

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

注意：

- 升级现有安装将覆盖 /etc/xdl 下的配置文件。在进行升级之前，请务必备份这些文件。
- 在 RHEL 9.4/9.3/9.2 和 Rocky Linux 9.4/9.3/9.2 上升级 Linux VDA 之前，请将 **libsepol** 软件包更新到版本 3.4 或更高版本。
- 从 2407 版本开始，Linux VDA 委派包管理器 rpm 或 dpkg 在升级期间处理配置文件。下面介绍了 rpm 和 dpkg 如何与配置文件更改交互：

- **rpm**: 默认保留本地版本并使用.rpmpnew 扩展名保存软件包中的新版本。
- **dpkg**: 以交互方式提示您选择如何继续。要以静默方式升级 Linux VDA, 同时保留本地配置文件并将新的软件包版本另存为 **.dpkg-new** 或 **.dpkg-dist**, 请使用以下命令:

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

- 升级软件后重新启动 Linux VDA 计算机。

步骤 7: 安装 NVIDIA GRID 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

注意:

要对 Amazon Linux 2 使用 HDX 3D Pro, 我们建议您安装 NVIDIA 驱动程序 470。有关详细信息, 请参阅[系统要求](#)。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器 (主机驱动程序), 请参阅以下指南:

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

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序, 请执行以下步骤:

1. 确保来宾 VM 已关闭。
2. 在 XenCenter 中, 为 VM 分配一个 GPU。
3. 启动 VM。
4. 为 NVIDIA GRID 驱动程序准备 VM:

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

5. 请按照 [Red Hat Enterprise Linux 文档](#)中的步骤安装 NVIDIA GRID 驱动程序。

注意:

安装 GPU 驱动程序期间, 为每个问题选择默认答案 (no)。

重要：

在启用 GPU 直通后，无法再通过 XenCenter 访问 Linux VM。使用 SSH 进行连接。

```
nvidia-smi
```

```

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

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

```

为显卡设置正确的配置：

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

要利用高分辨率和多监视器功能，您需要有效的 NVIDIA 许可证。要申请许可证，请按照“GRID Licensing Guide.pdf - DU-07757-001 September 2015”产品文档执行操作。

步骤 8：配置 Linux VDA**注意：**

在设置运行时环境之前，请确保已在您的操作系统中安装了 **en_US.UTF-8** 区域设置。如果该区域设置在您的操作系统中不可用，请运行 **sudo locale-gen en_US.UTF-8** 命令。对于 Debian，通过取消注释 **# en_US.UTF-8 UTF-8** 行来编辑 **/etc/locale.gen** 文件，然后运行 **sudo locale-gen** 命令。

安装软件包后，必须运行 **ctxsetup.sh** 脚本来配置 Linux VDA。执行任何更改之前，该脚本都会验证环境，确保所有依赖项都已安装。如有必要，可以随时重新运行该脚本以更改设置。

可以按照提示手动运行脚本，也可以采用预先配置的响应自动运行脚本。继续操作前，请查看该脚本的帮助信息：

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

提示配置

运行会提示各种问题的手动配置：

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

自动配置

自动安装时，通过环境变量提供设置脚本所需的选项。如果所需的所有变量都存在，脚本不会提示您提供任何信息。

支持的环境变量包括：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于已加入域的场景，请将其设置为 'n'。
- **CTX_XDL_AD_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA 要求使用 Kerberos 配置设置向 Delivery Controller 进行身份验证。Kerberos 配置根据系统上已安装和已配置的 Active Directory 集成工具确定。
- **CTX_XDL_DDC_LIST=' <list-ddc-fqdns>'** - Linux VDA 要求提供由空格分隔的 Delivery Controller 完全限定域名 (FQDN) 列表以用于向 Delivery Controller 进行注册。必须至少指定一个 FQDN 或 CNAME。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境，将该值设置为 'y'。
- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro，这是一组 GPU 加速技术，旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro，则要为 VDI 桌面（单会话）模式配置 VDA（即 CTX_XDL_VDI_MODE=' y'）。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口（默认端口 80 和 1494）。
- **CTX_XDL_DESKTOP_ENVIRONMENT='gnome/gnome-classic/kde/mate/xfce/' <none> '-** 指定要在会话中使用的 GNOME、GNOME Classic、KDE、MATE 或 Xfce 桌面环境。如果将其设置为 '<none>'，则将使用在 VDA 上配置的默认桌面。
您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息，请参阅[桌面切换命令](#)和[系统托盘](#)。
- **CTX_XDL_DOTNET_RUNTIME_PATH='path-to-install-dotnet-runtime'** - 安装 .NET 以支持新的 Broker 代理服务 (ctxvda) 的路径。默认路径为 '/usr/bin'。
- **CTX_XDL_VDA_PORT='port-number'** - Linux VDA 通过 TCP/IP 端口与 Delivery Controller 通信。
- **CTX_XDL_SITE_NAME='<dns-name>'** - Linux VDA 通过 DNS 发现 LDAP 服务器。要将 DNS 搜索结果限制为本地站点，应指定 DNS 站点名称。如果不需要，请将其设置为 '<none>'。
- **CTX_XDL_LDAP_LIST=' <list-ldap-servers>'** - Linux VDA 查询 DNS 来发现 LDAP 服务器。如果 DNS 无法提供 LDAP 服务记录，您可以提供以空格分隔的 LDAP FQDN（带有 LDAP 端口）列表。例如，

ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。要在 Active Directory 林中启用更快的 LDAP 查询，请在域控制器上启用全局目录，并将相关的 LDAP 端口号指定为 3268。默认情况下，此变量设置为 '**<none>**'。

- **CTX_XDL_SEARCH_BASE=search-base-set** - Linux VDA 通过设置为 Active Directory 域根的搜索库来查询 LDAP（例如，DC=mycompany,DC=com）。为提高搜索性能，可以指定搜索基础（例如 OU=VDI,DC=mycompany,DC=com）。如果不需要，请将其设置为 '**<none>**'。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'** - Linux VDA 支持使用 DNS CNAME 记录指定 Delivery Controller 名称。

设置环境变量并运行配置脚本：

```

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

```

运行 `sudo` 命令时，键入 **-E** 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 **#!/bin/bash** 作为第一行来创建 shell 脚本文件。

另外，您可以使用单个命令指定所有参数：

```

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

```

```
16 /opt/Citrix/VDA/sbin/ctxsetup.sh --silent
```

删除配置更改

在某些情形下，您可能需要删除 **ctxsetup.sh** 脚本对配置所做的更改，但不卸载 Linux VDA 软件包。

继续操作前，请查看此脚本的帮助信息：

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

删除配置更改：

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

重要：

此脚本会从数据库删除所有配置数据，从而使 Linux VDA 无法使用。

配置日志

ctxsetup.sh 和 **ctxcleanup.sh** 脚本会在控制台上显示错误，并将其他信息写入配置日志文件 **/tmp/xdl.configure.log**。

重新启动 Linux VDA 服务，确保更改生效。

步骤 9：运行 **XDPing**

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅 [XDPing](#)。

步骤 10：运行 **Linux VDA**

使用 **ctxsetup.sh** 脚本配置 Linux VDA 后，可以运行以下命令来控制 Linux VDA。

启动 **Linux VDA**：

启动 Linux VDA 服务：

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

停止 **Linux VDA**：

要停止 Linux VDA 服务，请执行以下操作：


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

注意：

在停止 **ctxvda** 和 **ctxhdx** 服务之前，请运行 **systemctl stop ctxmonitord** 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

重新启动 **Linux VDA**：

重新启动 Linux VDA 服务：

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

检查 **Linux VDA** 的状态：

要检查 Linux VDA 服务的运行状态，请执行以下操作：

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

步骤 11：创建计算机目录

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明，请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制，使得该过程不同于为 Windows VDA 计算机创建计算机目录：

- 对于操作系统，请选择：
 - 多会话操作系统选项（对于托管共享桌面交付模型）。
 - 单会话操作系统选项（适用于 VDI 专用桌面交付模型）。
- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。

注意：

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是，选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

提示：

当您已将删除的计算机重新加入到 Active Directory 域时，请从中删除该计算机并将其重新添加到其计算机目录中。

步骤 12：创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明，请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制：

- 确保所选 AD 用户和组已正确配置，能够登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的（匿名）用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

重要：

Linux VDA 1.4 及更高版本支持发布应用程序。但是，Linux VDA 不支持将桌面和应用程序交付给相同的计算机。

有关如何创建计算机目录和交付组的信息，请参阅 [Citrix Virtual Apps and Desktops 7 2407](#)。

在 SUSE 上手动安装 Linux VDA

August 20, 2024

重要：

对于全新安装，我们建议您使用[轻松安装](#)进行快速安装。轻松安装省时又省力，与本文中详述的手动安装相比，更不易于出错。

步骤 1：准备配置信息和 Linux 计算机

步骤 1a：启动 YaST 工具

SUSE Linux Enterprise YaST 工具用于对操作系统执行方方面面的配置。

启动基于文本的 YaST 工具：

```
1 su -
2
3 yast
```

要启动基于 UI 的 YaST 工具，请执行以下操作：

```
1 su -
2
3 yast2 &
```

步骤 **1b**: 配置网络连接

以下各部分介绍了如何配置 Linux VDA 使用的各种网络设置和服务。网络配置通过 YaST 工具执行，而不得使用其他方法，例如 Network Manager。以下说明介绍的是使用基于 UI 的 YaST 工具的情形。也可以使用基于文本的 YaST 工具，但导航方法稍有不同，对此本文未作介绍。

配置主机名和域名系统 (DNS)

1. 启动基于 UI 的 YaST 工具。
2. 选择系统，然后选择网络设置。
3. 打开 **Hostname/DNS** (主机名/DNS) 选项卡。
4. 为 **Set Hostname via DHCP** (通过 DHCP 设置主机名) 选择 **no** (否) 选项。
5. 为修改 **DNS** 配置选择使用自定义策略选项。
6. 编辑以下内容，以反映所作的网络设置：
 - 静态主机名 - 添加计算机的 DNS 主机名。
 - 名称服务器 - 添加 DNS 服务器的 IP 地址。通常是 AD 域控制器的 IP 地址。
 - 域搜索列表 - 添加 DNS 域名。

7. 更改 `/etc/hosts` 文件的以下行以包含 FQDN 和主机名作为前两个条目：

```
127.0.0.1 <FQDN of the VDA> <hostname of the VDA> localhost
```

注意：

Linux VDA 当前不支持 NetBIOS 名称截断。因此，主机名不得超过 15 个字符。

提示：

只能使用字符 a-z、A-Z、0-9 和连字符 (-)。请避免使用下划线 (_)、空格和其他符号。主机名不得以数字开头和以连字符结尾。此规则也适用于 Delivery Controller 主机名。

检查主机名 验证主机名设置是否正确无误：

```
1 hostname
```

此命令仅返回计算机的主机名，而不返回其完全限定域名 (FQDN)。

验证 FQDN 设置是否正确无误：

```
1 hostname -f
```

此命令返回计算机的 FQDN。

检查名称解析和服务可访问性 确认可以解析 FQDN 并对域控制器和 Delivery Controller 执行 ping 操作：

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

如果无法解析 FQDN 或 Ping 不通上述任一计算机，请先检查相关步骤，然后再继续。

步骤 1c: 配置 NTP 服务

维护 VDA、Delivery Controller 和域控制器之间的时钟始终精确同步至关重要。将 Linux VDA 托管为虚拟机 (VM) 可能会导致时钟偏差问题。出于此原因，最好使用远程 NTP 服务来保持时间同步。默认 NTP 设置可能需要作一些更改。

对于 **SUSE 15.5**：

1. 启动基于 UI 的 YaST 工具。
2. 选择网络服务，然后选择 **NTP** 配置。
3. 在 **Start NTP Daemon**（启动 NTP 守护程序）部分，选择 **Now and on Boot**（现在及引导时）。
4. 为配置源选择动态。
5. 根据需要添加 NTP 服务器。NTP 服务通常托管在 Active Directory 域控制器上。
6. 在 `/etc/chrony.conf` 中删除或注释以下行（如果存在）。

```
include /etc/chrony.d/*.conf
```

编辑 `chrony.conf` 后，重新启动 `chronyd` 服务。

```
1 sudo systemctl restart chronyd.service
```

步骤 1d: 安装 Linux VDA 依赖软件包

适用于 SUSE Linux Enterprise 的 Linux VDA 软件依赖于以下软件包：

- OpenJDK 11

- Open Motif Runtime Environment 2.3.1 或更高版本
- Cups 1.6.0 或更高版本
- ImageMagick 6.8 或更高版本

添加存储库 除了 ImageMagick 之外，您还可以从官方存储库中获取大多数必需的软件包。要获取 ImageMagick 软件包，请使用 YaST 或以下命令启用 `sle-module-desktop-applications` 存储库：

```
SUSEConnect -p sle-module-desktop-applications/<version number>/  
x86_64
```

安装 Kerberos 客户端 安装 Kerberos 客户端，在 Linux VDA 与 Delivery Controller 之间实现双向身份验证：

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

Kerberos 客户端配置依赖于所使用的 Active Directory 集成方法。请参阅下面的说明。

安装 OpenJDK 11 Linux VDA 要求存在 OpenJDK 11。

要安装 OpenJDK 11，请运行以下命令：

```
1 sudo zypper install java-11-openjdk
```

安装并指定要使用的数据库

注意：

- 我们建议您仅将 SQLite 用于 VDI 模式，将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS，您可以指定使用 SQLite 或 PostgreSQL，而不必手动安装。除非通过 `/etc/xdl/db.conf` 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 对于手动安装，您必须手动安装 SQLite、PostgreSQL 或者两者。如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 `/etc/xdl/db.conf` 来指定使用其中一个。

本部分内容介绍如何安装 PostgreSQL 和 SQLite 以及如何指定要使用的两者中的任意一个。

安装 PostgreSQL 要安装 PostgreSQL，请运行以下命令：

```
1 sudo zypper install postgresql-server  
2  
3 sudo zypper install postgresql-jdbc
```

请分别运行以下命令以在计算机启动时启动 PostgreSQL 或者立即启动：

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl restart postgresql
```

安装 **SQLite** 对于 SUSE，请运行以下命令以安装 SQLite:

```
1 sudo zypper install sqlite3
```

指定要使用的数据库 如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

1. 运行 **/opt/Citrix/VDA/sbin/ctxcleanup.sh**。如果是全新安装，则省略此步骤。
2. 编辑 **/etc/xdl/db.conf** 以指定要使用的数据库。
3. 运行 **ctxsetup.sh**。

注意:

也可以使用 **/etc/xdl/db.conf** 配置 PostgreSQL 的端口号。

步骤 2: 准备虚拟机管理程序

在支持的虚拟机管理程序上将 Linux VDA 当作 VM 运行时，需要作出一些更改。根据正在使用的虚拟机管理程序平台做出以下更改。如果正在裸机硬件上运行 Linux 计算机，则无需作出任何更改。

修复 **XenServer** (以前称为 **Citrix Hypervisor**) 上的时间同步问题

如果启用了 XenServer 时间同步功能，则在每个半虚拟化 Linux VM 中，您会遇到 NTP 和 XenServer 出现的问题。两者都试图管理系统时钟。为避免时钟与其他服务器不同步，请将每个 Linux 来宾中的系统时钟与 NTP 同步。这种情况要求禁用主机时间同步。无需在 HVM 模式下进行任何更改。

如果您正在运行半虚拟化 Linux 内核，并且安装了 XenServer Tools，则可以检查 XenServer 时间同步功能是否存在以及是否已从 Linux VM 中启用:

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

此命令返回 0 或 1:

- 0 - 时间同步功能已启用，且必须禁用。
- 1 - 时间同步功能已禁用，无需采取任何操作。

如果 `/proc/sys/xen/independent_wallclock` 文件不存在，则不需要执行以下步骤。

如果已启用，请通过向该文件写入 `1` 来禁用时间同步功能：

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

要使此更改成为永久更改，并在重新启动后仍然有效，请编辑 `/etc/sysctl.conf` 文件并添加以下行：

```
xen.independent_wallclock = 1
```

要验证这些更改，请重新启动系统：

```
1 reboot
```

重新启动后，验证此设置是否正确：

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

此命令返回值 `1`。

修复 **Microsoft Hyper-V** 上的时间同步问题

安装了 Hyper-V Linux 集成服务的 Linux VM 可应用 Hyper-V 时间同步功能来使用主机操作系统的时间。为确保系统时钟始终精确可靠，请一同启用此功能与 NTP 服务。

从管理操作系统中：

1. 打开 Hyper-V 管理器控制台。
2. 对于 Linux VM 的设置，请选择 **Integration Services**（集成服务）。
3. 确保已选择 **Time synchronization**（时间同步）。

注意：

此方法与 VMware 和 XenServer（以前称为 Citrix Hypervisor）不同，这两款产品会禁用主机时间同步功能以避免与 NTP 发生冲突。Hyper-V 时间同步可以与 NTP 时间同步共存并互补。

修复 **ESX** 和 **ESXi** 上的时间同步问题

如果启用了 VMware 时间同步功能，在每个半虚拟化的 Linux VM 中，您都会遇到 NTP 和虚拟机管理程序的问题。两者都试图同步系统时钟。为避免时钟与其他服务器不同步，请将每个 Linux 来宾中的系统时钟与 NTP 同步。这种情况要求禁用主机时间同步。

如果正在运行安装了 VMware Tools 的半虚拟化 Linux 内核：

1. 打开 vSphere Client。
2. 编辑 Linux VM 设置。

3. 在 **Virtual Machine Properties** (虚拟机属性) 对话框中, 打开 **Options** (选项) 选项卡。
4. 选择 **VMware Tools**。
5. 在 **Advanced** (高级) 框中, 取消选中 **Synchronize guest time with host** (与主机同步客户机时间)。

步骤 3: 将 **Linux VM** 添加到 **Windows** 域中

以下方法可用于将 Linux 计算机添加到 Active Directory (AD) 域:

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

根据所选的方法, 按说明执行操作。

注意:

为 Linux VDA 中的本地帐户和 AD 中的帐户使用相同的用户名时, 会话启动可能会失败。

Samba Winbind

加入 **Windows** 域 您的域控制器必须可访问, 并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户:

1. 启动 YaST, 选择网络服务, 然后选择 **Windows** 域成员身份。
2. 进行以下更改:
 - 将域或工作组设为 Active Directory 域的名称或域控制器的 IP 地址。确保域名为大写。
 - 选中 **Use SMB information for Linux Authentication** (为 Linux 身份验证使用 SMB 信息)。
 - 选中 **Create Home Directory on Login** (在登录时创建主目录)。
 - 选中 **Single Sign-on for SSH** (为 SSH 使用单点登录)。
 - 确保未选中 **Offline Authentication** (脱机身份验证)。此选项与 Linux VDA 不兼容。
3. 单击确定。如果系统提示您安装某些软件包, 请单击 **Install** (安装)。
4. 如果找到域控制器, 则会询问您是否要加入域。单击是。
5. 出现提示时, 键入有权将计算机添加到域的域用户的凭据, 然后单击 **OK** (确定)。
6. 手动重新启动服务或重新启动计算机。我们建议您重新启动计算机:

```
1 su -
2 reboot
```


验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 Active Directory 中有一个计算机对象。

运行 **Samba** 的 **net ads** 命令以验证计算机是否已加入域:

```
1 sudo net ads testjoin
```

运行以下命令验证额外的域和计算机对象信息:

```
1 sudo net ads info
```

验证 **Kerberos** 配置 确保系统 **keytab** 文件是否已创建并且包含有效密钥:

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令, 使用这些密钥向域控制器验证计算机的身份:

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义, 以免发生 shell 替换。在某些环境中, DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行, 则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存:

```
1 sudo klist
```

请使用以下命令检查计算机帐户详细信息:

```
1 sudo net ads status
```

验证用户身份验证 使用 **wbinfo** 工具验证是否可向域验证域用户的身份:

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

这里指定的域为 AD 域名, 而不是 Kerberos 领域名称。对于 bash shell, 必须使用另一个反斜杠对反斜杠 (\) 字符进行转义。此命令返回一条成功或失败消息。

验证 Winbind PAM 模块的配置是否正确。为此, 请使用以前未使用过的域用户帐户登录到 Linux VDA。

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

验证是否为 **id -u** 命令返回的 **uid** 创建了对应的 Kerberos 凭据缓存文件:

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

验证用户的 Kerberos 凭据缓存中的票据是否有效且未过期:

```
1 klist
```

退出会话。

```
1 exit
```

直接登录 Gnome 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

Quest Authentication Service

在域控制器上配置 **Quest** 假定您已在域控制器上安装并配置了 Quest 软件，而且已获得管理权限，有权在 **Active Directory** 中创建计算机对象。

允许域用户登录 **Linux VDA** 计算机 为了让域用户能够在 Linux VDA 计算机上建立 HDX 会话：

1. 在 Active Directory 用户和计算机管理控制台中，为该用户帐户打开 Active Directory 用户属性。
2. 选择 **Unix Account** (Unix 帐户) 选项卡。
3. 选中 **Unix-enabled** (已启用 Unix)。
4. 将 **Primary GID Number** (首选 GID 编号) 设置为实际域用户组的组 ID。

注意：

这些说明相当于设置域用户，以便他们可以使用控制台、RDP、SSH 或任何其他远程协议进行登录。

在 Linux VDA 上配置 Quest

配置 **VAS** 守护程序 必须启用并断开自动续订 Kerberos 票据功能。必须禁用身份验证 (脱机登录) 功能：

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

此命令将续订间隔设为 9 小时 (32400 秒)，即比默认的 10 小时票据生命周期短 1 小时。请在票据生命周期较短的系统上设置较低的值。

配置 **PAM** 和 **NSS** 要启用通过 HDX 进行的域用户登录以及其他服务 (例如 su、ssh 和 RDP)，请手动配置 PAM 和 NSS：

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

加入 **Windows** 域 使用 Quest `vastool` 命令将 Linux 计算机加入到 Active Directory 域中:

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

user 为有权将计算机加入 Active Directory 域的任何域用户。**domain-name** 为域的 DNS 名称, 例如 `example.com`。

加入域后重新启动 Linux 计算机。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 **Active Directory** 中有一个计算机对象。要验证加入了 Quest 的 Linux 计算机是否位于域中, 请执行以下操作:

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

如果计算机已加入域, 此命令会返回域名。如果计算机未加入任何域, 则会显示以下错误:

```
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
```

验证用户身份验证 验证 Quest 是否可以通过 PAM 对域用户进行身份验证。为此, 请使用以前未使用过的域用户帐户登录到 Linux VDA。

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

验证是否为 `id -u` 命令返回的 `uid` 创建了对应的 Kerberos 凭据缓存文件:

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

验证 Kerberos 凭据缓存中的票据是否有效且未过期:

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

退出会话。

```
1 exit
```

直接登录 Gnome 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

Centrify DirectControl

加入 **Windows** 域 安装 Centrify DirectControl Agent 后, 请使用 Centrify **adjoin** 命令将 Linux 计算机加入 Active Directory 域:

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

user 为有权将计算机加入 Active Directory 域的任何 Active Directory 域用户。**domain-name** 是将 Linux 计算机加入到的域的名称。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (Windows 和 Linux VDA) 都要在 Active Directory 中有一个计算机对象。要验证加入了 Centrify 的 Linux 计算机是否位于域中, 请执行以下操作:

```
1 sudo adinfo
```

验证 **Joined to domain** 值是否有效以及 **CentrifyDC mode** 是否返回了 **connected**。如果模式仍然卡在正在启动状态, 则表明 Centrify 客户端遇到了服务器连接或身份验证问题。

请使用以下命令可获得更全面的系统和诊断信息:

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

测试与各种 Active Directory 和 Kerberos 服务的连接。

```
1 adinfo --test
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

SSSD

如果您在 SUSE 上使用 SSSD, 请按照此部分中的说明进行操作。此部分包含有关如何将 Linux VDA 计算机加入 Windows 域的说明以及如何配置 Kerberos 身份验证的指导。

要在 SUSE 上设置 SSSD, 请完成以下步骤:

1. 加入域并创建主机 keytab
2. 为 SSSD 配置 PAM
3. 设置 SSSD
4. 启用 SSSD
5. 验证域成员身份
6. 验证 Kerberos 配置
7. 验证用户身份验证

加入域并创建主机 **keytab** SSSD 并不提供用于加入域和管理系统 keytab 文件的 Active Directory 客户端功能。可以改为使用 **Samba** 方法。在配置 SSSD 之前, 请完成以下步骤。

1. 停止并禁用 Name Service Cache Daemon (NSCD) 守护进程。

```
1 sudo systemctl stop nscd
2 sudo systemctl disable nscd
```

2. 检查主机名和 Chrony 时间同步。

```
1 hostname
2 hostname -f
3 chronyc traking
```

3. 安装或更新所需软件包：

```
1 sudo zypper install samba-client sssd-ad
```

4. 以 root 用户身份编辑 `/etc/krb5.conf` 文件，以允许 **kinit** 实用程序与目标域进行通信。在 **[libdefaults]**、**[realms]** 和 **[domain_realm]** 部分下添加以下条目：

注意：

根据您的 AD 基础结构配置 Kerberos。以下设置适用于单域、单林模型。

```
1 [libdefaults]
2
3     dns_canonicalize_hostname = false
4
5     rdns = false
6
7     default_realm = REALM
8
9     forwardable = true
10
11 [realms]
12
13     REALM = {
14
15         kdc = fqdn-of-domain-controller
16
17         default_domain = realm
18
19         admin_server = fqdn-of-domain-controller
20     }
21
22
23 [domain_realm]
24
25     .realm = REALM
```

realm 是 Kerberos 领域的名称，例如 `example.com`。**REALM** 是大写的 Kerberos 领域名称，例如 `EXAMPLE.COM`。

5. 以 root 用户身份编辑 `/etc/samba/smb.conf`，以允许 **net** 实用程序与目标域进行通信。将以下条目添加到 **[global]** 部分：

```
1 [global]
2     workgroup = domain
3
4     client signing = yes
5
6     client use spnego = yes
7
8     kerberos method = secrets and keytab
9
10    realm = REALM
11
12    security = ADS
```

domain 是 Active Directory 域的简短 NetBIOS 名称，例如 EXAMPLE。

6. 修改 `/etc/nsswitch.conf` 文件中的 **passwd** 和 **group** 条目以在解析用户和组时引用 SSSD。

```
1 passwd: compat sss
2
3 group: compat sss
```

7. 使用已配置的 Kerberos 客户端以管理员身份向目标域进行身份验证。

```
1 kinit administrator
```

8. 使用 **net** 实用程序将系统加入域并生成系统 Keytab 文件。

```
1 net ads join osname="SUSE Linux Enterprise Server" osVersion=15 -U
   administrator
```

为 **SSSD** 配置 **PAM** 在为 SSSD 配置 PAM 之前，请安装或更新所需的软件包：

```
1 sudo zypper install sssd sssd-ad
```

将 PAM 模块配置为通过 SSSD 进行用户身份验证，并为用户登录创建主目录。

```
1 sudo pam-config --add --sss
2 sudo pam-config --add--mkhomedir
```

设置 SSSD

1. 以 root 用户身份编辑 `/etc/sss/sss.conf`，以允许 SSSD 守护程序与目标域进行通信。 `sss.conf` 配置示例（可以根据需要添加额外的选项）：

```
1 [sss]
2     config_file_version = 2
3     services = nss,pam
4     domains = domain-dns-name
5
```

```

6 [domain/domain-dns-name]
7     id_provider = ad
8     auth_provider = ad
9     access_provider = ad
10    ad_domain = domain-dns-name
11    ad_server = fqdn-of-domain-controller
12    ldap_id_mapping = true
13    ldap_schema = ad
14
15 # Kerberos settings
16     krb5_ccachedir = /tmp
17     krb5_ccname_template = FILE:%d/krb5cc_%U
18
19 # Comment out if the users have the shell and home dir set on the
    AD side
20
21     fallback_homedir = /home/%d/%u
22     default_shell = /bin/bash
23
24 # Uncomment and adjust if the default principal SHORTNAME$@REALM
    is not available
25
26 # ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
27
28     ad_gpo_access_control = permissive

```

domain-dns-name 是 DNS 域名，例如 example.com。

注意：

ldap_id_mapping 设置为 true，以便 SSSD 本身负责将 Windows SID 映射到 Unix UID。否则，Active Directory 必须能够提供 POSIX 扩展程序。将 **ad_gpo_access_control** 设置为 **permissive** 以防止 Linux 会话出现无效登录错误。请参阅 [sssd.conf](#) 和 [sssd-ad](#) 的手册页。

2. 对 `sssd.conf` 设置文件所有权和权限：

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

启用 **SSSD** 请运行以下命令以在系统启动时启用并启动 SSSD 守护程序：

```

1 sudo systemctl enable sssd
2 sudo systemctl start sssd

```

验证域成员身份

1. 运行 **Samba** 的 `net ads` 命令以验证计算机是否已加入域：

```
1 sudo net ads testjoin
```

2. 运行以下命令验证额外的域和计算机对象信息：

```
1 sudo net ads info
```

验证 **Kerberos** 配置 确保系统 keytab 文件是否已创建并且包含有效密钥:

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。

运行 Kerberos **kinit** 命令, 以使用这些密钥向域控制器验证计算机的身份:

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (****) 进行转义, 以免发生 shell 替换。在某些环境中, DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行, 则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存:

```
1 sudo klist
```

验证用户身份验证 SSSD 不直接通过守护程序提供用于测试身份验证的命令行工具, 只能通过 PAM 完成。

要验证 SSSD PAM 模块是否已正确配置, 请使用以前未使用的域用户帐户登录 Linux VDA。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

验证 **klist** 命令返回的 Kerberos 票据是否适用于该用户并且尚未过期。

以 root 用户身份, 验证是否已为前面的 **id -u** 命令返回的 uid 创建相应的票据缓存文件:

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

直接登录 Gnome 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

PBIS

下载所需的 **PBIS** 软件包 例如:

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


使 **PBIS** 安装脚本可执行 例如：

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

运行 **PBIS** 安装脚本 例如：

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

加入 **Windows** 域 您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户：

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

user 为有权将计算机添加到 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称，例如 example.com。

注意：要将 Bash 设置为默认 shell，请运行 **/opt/pbis/bin/config LoginShellTemplate/bin/bash** 命令。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 **Active Directory** 中有一个计算机对象。要验证加入了 PBIS 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入某个域，此命令将返回有关当前加入的 AD 域和 OU 的信息。否则，仅显示主机名。

验证用户身份验证 验证 PBIS 是否能够通过 PAM 对域用户进行身份验证。为此，请使用以前未使用过的域用户帐户登录到 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

步骤 4: 安装.NET

除.NET Runtime 之外, 在安装或升级 Linux VDA 之前, 还必须在所有受支持的 Linux 发行版中安装.ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

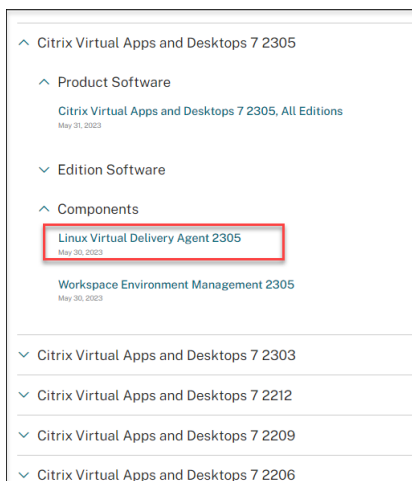
如果您的 Linux 发行版包含所需的.NET 版本, 请从内置源中进行安装。否则, 请从 Microsoft 软件包源中安装.NET。有关详细信息, 请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装.NET 后, 请运行 **which dotnet** 命令查找您的运行时路径。

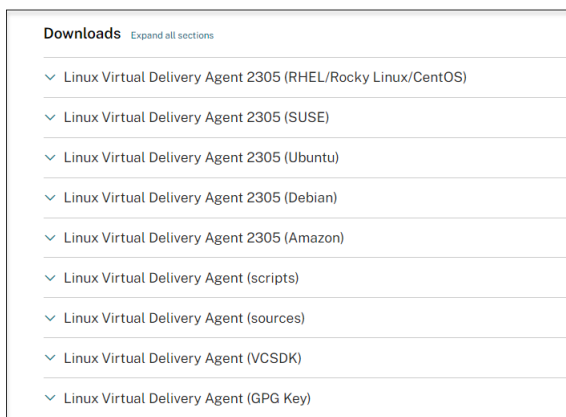
根据命令输出, 设置.NET Runtime 二进制文件路径。例如, 如果命令输出为 /aa/bb/dotnet, 请使用 /aa/bb 作为.NET 二进制文件路径。

步骤 5: 下载 Linux VDA 软件包

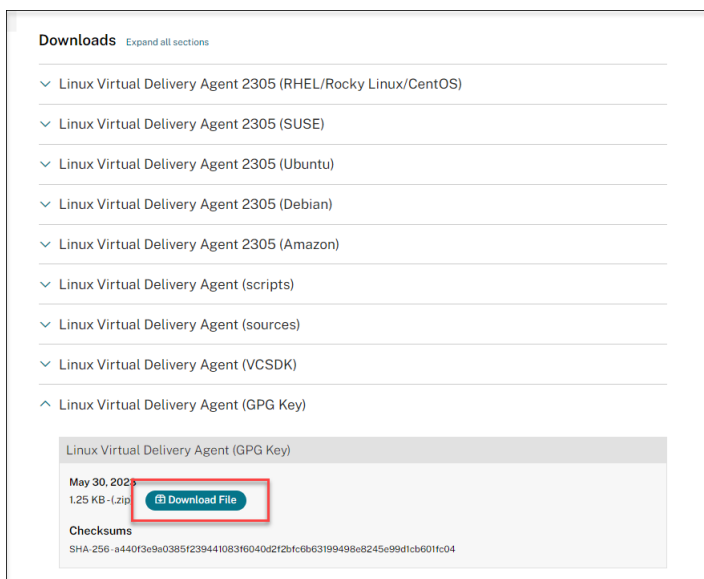
1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。
3. 展开组件以查找 Linux VDA。例如:



4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。
6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要使用公钥验证 Linux VDA 软件包的完整性，请运行以下命令以将公钥导入到 RPM 数据库中以及检查软件包的完整性：

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

步骤 6：安装 Linux VDA

步骤 6a：卸载旧版本

如果安装了除先前的两个版本和 LTSR 版本之外的早期版本，请在安装新版本之前将其卸载。

1. 停止 Linux VDA 服务：

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

注意：

在停止 **ctxvda** 和 **ctxhdx** 服务之前，请运行 **systemctl stop ctxmonitord** 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

2. 卸载软件包：

```
1 sudo rpm -e XenDesktopVDA
```

重要:

支持从最新的两个版本进行升级。

注意:

您可以在 `/opt/Citrix/vda/` 下找到已安装的组件。

要运行命令，需要提供完整路径；或者，也可以将 `/opt/Citrix/VDA/sbin` 和 `/opt/Citrix/VDA/bin` 添加到系统路径。

步骤 6b: 安装 Linux VDA

使用 Zypper 安装 Linux VDA 软件:

```
1 sudo zypper install XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

使用 RPM 软件包管理器安装 Linux VDA 软件:

```
1 sudo rpm -i XenDesktopVDA-<version>.sle15_x.x86_64.rpm
```

步骤 6c: 升级 Linux VDA (可选)

Linux VDA 支持从最新版本进行升级。例如，您可以将 Linux VDA 从 2308 升级到 2311，从 1912 LTSR 升级到 2203 LTSR。

注意:

升级现有安装将覆盖 `/etc/xdm` 下的配置文件。在进行升级之前，请务必备份这些文件。

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

SUSE 15 的 RPM 依赖项列表:

```
1 java-11-openjdk >= 11
2
3 ImageMagick >= 7.0
4
5 dbus-1 >= 1.12.2
6
7 dbus-1-x11 >= 1.12.2
8
9 xorg-x11 >= 7.6_1
10
11 libXpm4 >= 3.5.12
12
13 libXrandr2 >= 1.5.1
14
15 libXtst6 >= 1.2.3
```

```
16
17 pam >= 1.3.0
18
19 bash >= 4.4
20
21 findutils >= 4.6
22
23 gawk >= 4.2
24
25 sed >= 4.4
26
27 cups >= 2.2
28
29 cups-filters >= 1.25
30
31 libxml2-2 >= 2.9
32
33 libmspack0 >= 0.6
34
35 ibus >= 1.5
36
37 libtcmalloc4 >= 2.5
38
39 libcap-progs >= 2.26
40
41 mozilla-nss-tools >= 3.53.1
42
43 libpython3_6m1_0 >= 3.6~
44
45 libQt5Widgets5 >= 5.12
46
47 libqrencode4 >= 4.0.0
48
49 libImlib2-1 >= 1.4.10
```

重要：

在升级后重新启动 Linux VDA 计算机。

步骤 7：安装 **NVIDIA GRID** 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器（主机驱动程序），请参阅以下指南：

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

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序，请执行下面的常规步骤：

1. 确保来宾 VM 已关闭。
2. 在虚拟机管理程序控制面板中，为 VM 分配一个 GPU。
3. 启动 VM。
4. 在 VM 上安装来宾 VM 驱动程序。

步骤 8: 配置 Linux VDA

注意:

在设置运行时环境之前，请确保已在您的操作系统中安装了 **en_US.UTF-8** 区域设置。如果该区域设置在您的操作系统中不可用，请运行 **sudo locale-gen en_US.UTF-8** 命令。对于 Debian，通过取消注释 **# en_US.UTF-8 UTF-8** 行来编辑 **/etc/locale.gen** 文件，然后运行 **sudo locale-gen** 命令。

安装软件包后，必须运行 **ctxsetup.sh** 脚本来配置 Linux VDA。脚本做出任何更改之前，都会验证环境，并确保所有依赖项都已安装。如有必要，可以随时重新运行该脚本以更改设置。

可以按照提示手动运行脚本，也可以采用预先配置的响应自动运行脚本。继续操作前，请查看该脚本的帮助信息：

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

提示配置

运行会提示各种问题的手动配置：

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

自动配置

自动安装时，通过环境变量提供设置脚本所需的选项。如果所需的所有变量都存在，脚本不会提示您提供任何信息。

支持的环境变量包括：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于已加入域的场景，请将其设置为 'y'。
- **CTX_XDL_AD_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA 要求使用 Kerberos 配置设置向 Delivery Controller 进行身份验证。Kerberos 配置根据系统上已安装和已配置的 Active Directory 集成工具确定。
- **CTX_XDL_DDC_LIST=' <list-ddc-fqdns>'** - Linux VDA 要求提供由空格分隔的 Delivery Controller 完全限定域名 (FQDN) 列表以用于向 Delivery Controller 进行注册。必须至少指定一个 FQDN 或 CNAME。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境，将该值设置为 'y'。

- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro，这是一组 GPU 加速技术，旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro，则要为 VDI 桌面（单会话）模式配置 VDA（即 CTX_XDL_VDI_MODE= 'y'）。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口（默认端口 80 和 1494）。
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/gnome-classic/mate/ <none>** - 指定要在会话中使用的 GNOME、GNOME Classic 或 MATE 桌面环境。如果将其设置为 '**<none>**'；则将使用在 VDA 上配置的默认桌面。
您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息，请参阅[桌面切换命令](#)和[系统托盘](#)。
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** - 安装 .NET 以支持新的 Broker 代理服务 (ctxvda) 的路径。默认路径为 '**/usr/bin**'。
- **CTX_XDL_VDA_PORT=port-number** - Linux VDA 通过 TCP/IP 端口与 Delivery Controller 通信。
- **CTX_XDL_SITE_NAME=<dns-name>** - Linux VDA 通过 DNS 发现 LDAP 服务器。要将 DNS 搜索结果限制为本地站点，应指定 DNS 站点名称。如果不需要，请将其设置为 '**<none>**'。
- **CTX_XDL_LDAP_LIST=' <list-ldap-servers>'** - Linux VDA 查询 DNS 来发现 LDAP 服务器。如果 DNS 无法提供 LDAP 服务记录，您可以提供以空格分隔的 LDAP FQDN（带有 LDAP 端口）列表。例如，ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。要在 Active Directory 林中启用更快的 LDAP 查询，请在域控制器上启用全局目录，并将相关的 LDAP 端口号指定为 3268。默认情况下，此变量设置为 '**<none>**'。
- **CTX_XDL_SEARCH_BASE=search-base-set** - Linux VDA 通过设置为 Active Directory 域根的搜索库来查询 LDAP（例如，DC=mycompany,DC=com）。为提高搜索性能，可以指定搜索基础（例如 OU=VDI,DC=mycompany,DC=com）。如果不需要，请将其设置为 '**<none>**'。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME=' y|n'** - Linux VDA 支持使用 DNS CNAME 记录指定 Delivery Controller 名称。

设置环境变量并运行配置脚本：

```

1 export CTX_XDL_NON_DOMAIN_JOINED='n'
2 export CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest
3 export CTX_XDL_DDC_LIST='<list-ddc-fqdns>'
4 export CTX_XDL_VDI_MODE='y|n'
5 export CTX_XDL_HDX_3D_PRO='y|n'
6 export CTX_XDL_START_SERVICE='y|n'
7 export CTX_XDL_REGISTER_SERVICE='y|n'
8 export CTX_XDL_ADD_FIREWALL_RULES='y|n'
9 export CTX_XDL_DESKTOP_ENVIRONMENT= gnome|gnome-classic|mate|'<none>'
10 export CTX_XDL_DOTNET_RUNTIME_PATH='<path-to-install-dotnet-runtime>'

```

```

11 export CTX_XDL_VDA_PORT='<port-number>'
12 export CTX_XDL_SITE_NAME='<dns-site-name>' | '<none>'
13 export CTX_XDL_LDAP_LIST='<list-ldap-servers>' | '<none>'
14 export CTX_XDL_SEARCH_BASE='<search-base-set>' | '<none>'
15 export CTX_XDL_SUPPORT_DDC_AS_CNAME='y|n'
16 sudo -E /opt/Citrix/VDA/sbin/ctxsetup.sh --silent

```

运行 `sudo` 命令时，键入 **-E** 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 **#!/bin/bash** 作为第一行来创建 shell 脚本文件。

另外，您可以使用单个命令指定所有参数：

```

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

```

删除配置更改

在某些情形下，您可能需要删除 **ctxsetup.sh** 脚本对配置所做的更改，但不卸载 Linux VDA 软件包。

继续操作前，请查看此脚本的帮助信息：

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

删除配置更改：

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

重要：

此脚本会从数据库删除所有配置数据，从而使 Linux VDA 无法使用。

配置日志

ctxsetup.sh 和 **ctxcleanup.sh** 脚本会在控制台上显示错误，并将其他信息写入配置日志文件：

```
/tmp/xdl.configure.log
```


重新启动 Linux VDA 服务，确保更改生效。

步骤 9: 运行 **XDPing**

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅 [XDPing](#)。

步骤 10: 运行 **Linux VDA**

使用 `ctxsetup.sh` 脚本配置 Linux VDA 后，可以运行以下命令来控制 Linux VDA。

启动 **Linux VDA**:

启动 Linux VDA 服务:

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

停止 **Linux VDA**:

要停止 Linux VDA 服务，请执行以下操作:

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

注意:

在停止 `ctxvda` 和 `ctxhdx` 服务之前，请运行 `systemctl stop ctxmonitord` 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

重新启动 **Linux VDA**:

重新启动 Linux VDA 服务:

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

检查 **Linux VDA** 状态:

要检查 Linux VDA 服务的运行状态，请执行以下操作:

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

步骤 11：创建计算机目录

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明，请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制，使得该过程不同于为 Windows VDA 计算机创建计算机目录：

- 对于操作系统，请选择：
 - 多会话操作系统选项（对于托管共享桌面交付模型）。
 - 单会话操作系统选项（适用于 VDI 专用桌面交付模型）。
- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。

注意：

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是，选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

提示：

如果删除计算机后将其重新加入 Active Directory 域，则必须删除计算机，然后将其重新添加到计算机目录。

步骤 12：创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明，请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制：

- 确保所选的 AD 用户和组已正确配置，可以登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的（匿名）用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

重要：

Linux VDA 1.4 及更高版本支持发布应用程序。但是，Linux VDA 不支持将桌面和应用程序交付给相同的计算机。

有关如何创建计算机目录和交付组的信息，请参阅 [Citrix Virtual Apps and Desktops 7 2407](#)。

在 Ubuntu 上手动安装 Linux VDA

August 20, 2024

重要:

对于全新安装，我们建议您使用[轻松安装](#)进行快速安装。轻松安装省时又省力，与本文中详述的手动安装相比，更不易于出错。

步骤 1: 准备配置信息和 **Linux** 计算机

步骤 1a: 验证网络配置

确保已连接并正确配置网络。例如，必须在 Linux VDA 上配置 DNS 服务器。

如果您使用的是 Ubuntu Live Server，请在设置主机名之前在 `/etc/cloud/cloud.cfg` 配置文件中做以下更改：

```
preserve_hostname: true
```

步骤 1b: 设置主机名

为确保正确报告计算机的主机名，请更改 `/etc/hostname` 文件，使其仅包含计算机的主机名。

```
hostname
```

步骤 1c: 为主机名分配环回地址

确保计算机的 DNS 域名和完全限定域名 (FQDN) 正确地返回报告。方法是更改 `/etc/hosts` 文件的以下行以包含 FQDN 和主机名作为前两个条目：

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

例如：

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

从文件中的其他条目中删除对 `hostname-fqdn` 或 `hostname` 的任何其他引用。

注意:

Linux VDA 当前不支持 NetBIOS 名称截断。因此，主机名不得超过 15 个字符。

提示:

只能使用字符 a-z、A-Z、0-9 和连字符 (-)。请避免使用下划线 (_)、空格和其他符号。主机名不得以数字开头和以连字符结尾。此规则也适用于 Delivery Controller 主机名。

步骤 1d: 检查主机名

验证主机名设置是否正确无误：

```
1 hostname
```

此命令仅返回计算机的主机名，而不返回其 FQDN。

验证 FQDN 设置是否正确无误：

```
1 hostname -f
```

此命令返回计算机的 FQDN。

步骤 1e: 禁用多播 DNS

默认设置启用了多播 DNS (**mDNS**)，而这会导致名称解析结果不一致。

要禁用 **mDNS**，请编辑 **/etc/nsswitch.conf** 并更改包含以下内容的行：

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

更改为：

```
hosts: files dns
```

步骤 1f: 检查名称解析和服务可访问性

确认可以解析 FQDN 并对域控制器和 Delivery Controller 执行 ping 操作：

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

如果无法解析 FQDN 或 Ping 不通上述任一计算机，请先检查相关步骤，然后再继续。

步骤 1g: 配置时钟同步 (**chrony**)

确保 VDA、Delivery Controller 和域控制器之间的时钟始终精确同步至关重要。将 Linux VDA 托管为虚拟机 (VM) 可能会导致时钟偏差问题。出于此原因，最好使用远程时间服务来同步时间。

安装 chrony：

```
1 apt-get install chrony
```

以 root 用户身份，编辑 **/etc/chrony/chrony.conf** 并为每个远程时间服务器添加一个服务器条目：

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

在典型部署中，时间从本地域控制器同步，而不是直接从公共 NTP 池服务器同步。为域中的每个 Active Directory 域控制器添加一个服务器条目。

删除列出的任何其他 **server** 或 **pool** 条目，包括环回 IP 地址、localhost 和公共服务器 ***.pool.ntp.org** 条目。

保存更改并重新启动 Chrony 守护程序：

```
1 sudo systemctl restart chrony
```

步骤 1h: 安装 OpenJDK 11

Linux VDA 要求存在 OpenJDK 11。

Ubuntu 22.04 包括 OpenJDK 11。

要在 Ubuntu 20.04 上安装 OpenJDK 11，请运行以下命令：

```
1 sudo apt-get install -y openjdk-11-jdk
```

步骤 1i: 安装并指定要使用的数据库

注意：

- 我们建议您仅将 SQLite 用于 VDI 模式，将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS，您可以指定使用 SQLite 或 PostgreSQL，而不必手动安装。除非通过 **/etc/xdl/db.conf** 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 对于手动安装，您必须手动安装 SQLite、PostgreSQL 或者两者。如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

本部分内容介绍如何安装 PostgreSQL 和 SQLite 以及如何指定要使用的两者中的任意一个。

安装 PostgreSQL 请运行以下命令以安装 PostgreSQL：

```
1 sudo apt-get install -y postgresql
2
3 sudo apt-get install -y libpostgresql-jdbc-java
```

请分别运行以下命令以在计算机启动时启动 PostgreSQL 或者立即启动：

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl restart postgresql
```

安装 **SQLite** 对于 Ubuntu，请运行以下命令来安装 SQLite：

```
1 sudo apt-get install -y sqlite3
```

指定要使用的数据库 如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

1. 运行 **/opt/Citrix/VDA/sbin/ctxcleanup.sh**。如果是全新安装，则省略此步骤。
2. 编辑 **/etc/xdl/db.conf** 以指定要使用的数据库。
3. 运行 **ctxsetup.sh**。

注意：

也可以使用 **/etc/xdl/db.conf** 配置 PostgreSQL 的端口号。

步骤 1j: 安装 **Motif**

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

步骤 1k: 安装其他软件包

对于 Ubuntu 22.04:

```
1 sudo apt-get install -y libsasl2-2
2 sudo apt-get install -y libsasl2-modules-gssapi-mit
3 sudo apt-get install -y libldap-2.5-0
4 sudo apt-get install -y krb5-user
5 sudo apt-get install -y libgtk2.0-0
```

对于 Ubuntu 20.04:

```
1 sudo apt-get install -y libsasl2-2
2 sudo apt-get install -y libsasl2-modules-gssapi-mit
3 sudo apt-get install -y libldap-2.4-2
4 sudo apt-get install -y krb5-user
5 sudo apt-get install -y libgtk2.0-0
```

步骤 2: 准备虚拟机管理程序

在支持的虚拟机管理程序上将 Linux VDA 当作 VM 运行时，需要作出一些更改。根据正在使用的虚拟机管理程序平台做出以下更改。如果正在裸机硬件上运行 Linux 计算机，则无需作出任何更改。

修复 **XenServer**（以前称为 **Citrix Hypervisor**）上的时间同步问题

启用了 XenServer 时间同步功能时，在每个半虚拟化 Linux VM 中，您会遇到 NTP 和 XenServer 出现的问题。两者都试图管理系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。无需在 HVM 模式下进行任何更改。

如果您正在运行半虚拟化 Linux 内核，并且安装了 XenServer Tools，则可以检查 XenServer 时间同步功能是否存在以及是否已从 Linux VM 中启用：

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

此命令返回 0 或 1：

- 0 - 时间同步功能已启用，且必须禁用。
- 1 - 时间同步功能已禁用，无需采取任何操作。

如果 `/proc/sys/xen/independent_wallclock` 文件不存在，则不需要执行以下步骤。

如果已启用，请通过向该文件写入 1 来禁用时间同步功能：

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

要使此更改成为永久更改，并在重新启动后仍然有效，请编辑 `/etc/sysctl.conf` 文件并添加以下行：

```
xen.independent_wallclock = 1
```

要验证这些更改，请重新启动系统：

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

此命令返回值 1。

修复 **Microsoft Hyper-V** 上的时间同步问题

安装了 Hyper-V Linux 集成服务的 Linux VM 可以使用 Hyper-V 时间同步功能来使用主机操作系统的的时间。为确保系统时钟始终精确可靠，请一同启用此功能与 NTP 服务。

从管理操作系统中：

1. 打开 Hyper-V 管理器控制台。
2. 对于 Linux VM 的设置，请选择 **Integration Services**（集成服务）。
3. 确保已选择 **Time synchronization**（时间同步）。

注意：

此方法与 VMware 和 XenServer（以前称为 Citrix Hypervisor）不同，这两款产品会禁用主机时间同步功能以避免与 NTP 发生冲突。Hyper-V 时间同步可以与 NTP 时间同步共存并互补。

修复 ESX 和 ESXi 上的时间同步问题

启用 VMware 时间同步功能后，在每个半虚拟化的 Linux VM 中，您都会遇到 NTP 和虚拟机管理程序的问题。两者都试图同步系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。

如果正在运行安装了 VMware Tools 的半虚拟化 Linux 内核：

1. 打开 vSphere Client。
2. 编辑 Linux VM 设置。
3. 在 **Virtual Machine Properties**（虚拟机属性）对话框中，打开 **Options**（选项）选项卡。
4. 选择 **VMware Tools**。
5. 在 **Advanced**（高级）框中，取消选中 **Synchronize guest time with host**（与主机同步客户机时间）。

步骤 3：将 Linux VM 添加到 Windows 域中

以下方法可用于将 Linux 计算机添加到 Active Directory (AD) 域：

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

根据所选的方法，按说明执行操作。

注意：

为 Linux VDA 中的本地帐户和 AD 中的帐户使用相同的用户名时，会话启动可能会失败。

Samba Winbind

安装或更新所需软件包

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

在计算机启动时启用要启动的 **Winbind** 守护程序 Winbind 守护程序必须配置为在计算机启动时启动：

```
1 sudo systemctl enable winbind
```


注意：

确保 `winbind` 脚本位于 `/etc/init.d` 下。

配置 **Kerberos** 以 root 用户身份，打开 `/etc/krb5.conf` 并配置以下设置：

注意：

根据您的 AD 基础结构配置 Kerberos。以下设置适用于单域、单林模型。

```
[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
```

此上下文中的 **domain-dns-name** 参数为 DNS 域名，例如 **example.com**。**REALM** 是大写的 Kerberos 领域名称，例如 **EXAMPLE.COM**。

配置 **Winbind** 身份验证 手动配置 Winbind，因为 Ubuntu 没有诸如 RHEL 中的 **authconfig** 和 SUSE 中的 **yast2** 这类工具。

运行 **vim /etc/samba/smb.conf** 命令打开 `/etc/samba/smb.conf`，然后配置以下设置：

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
encrypt passwords = yes
idmap config *:range = 16777216-33554431
kerberos method = secrets and keytab
```

```
winbind refresh tickets = yes
```

```
template shell = /bin/bash
```

WORKGROUP 是 **REALM** 中的第一个字段，**REALM** 是大写的 Kerberos 领域名称。

配置 **nsswitch** 打开 **/etc/nsswitch.conf** 并将 **winbind** 附加到以下行：

```
passwd: compat winbind
```

```
group: compat winbind
```

加入 **Windows** 域 您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户：

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

其中，**REALM** 是大写的 Kerberos 领域名称，**user** 是有权将计算机添加到域的域用户。

重新启动 **winbind**

```
1 sudo systemctl restart winbind
```

为 **Winbind** 配置 **PAM** 运行以下命令，确保选中 **Winbind NT/Active Directory authentication** (Winbind NT/Active Directory 身份验证) 和 **Create home directory on login** (在登录时创建主目录) 选项：

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

提示：

仅当计算机加入域后，**winbind** 守护程序才会始终保持运行。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（不论是 Windows 还是 Linux）都要在 Active Directory 中有一个计算机对象。

运行 **Samba** 的 **net ads** 命令以验证计算机是否已加入域：

```
1 sudo net ads testjoin
```

运行以下命令验证额外的域和计算机对象信息：

```
1 sudo net ads info
```

验证 **Kerberos** 配置 要验证 Kerberos 是否已正确配置为可与 Linux VDA 配合使用，请验证系统 **keytab** 文件是否已创建并且包含有效密钥：

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令，以使用这些密钥向域控制器验证计算机的身份：

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义，以免发生 shell 替换。在某些环境中，DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行，则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存：

```
1 sudo klist
```

请使用以下命令检查计算机的帐户详细信息：

```
1 sudo net ads status
```

验证用户身份验证 使用 **wbinfo** 工具验证是否可向域验证域用户的身份：

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

这里指定的域为 AD 域名，而不是 Kerberos 领域名称。对于 bash shell，必须使用另一个反斜杠对反斜杠 (\) 字符进行转义。此命令返回一条成功或失败消息。

要验证 Winbind PAM 模块是否已正确配置，请使用以前未使用的域用户帐户登录 Linux VDA。

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

注意：

要成功运行 SSH 命令，请确保 SSH 已启用并正常运行。

验证是否为 **id -u** 命令返回的 uid 创建了对应的 Kerberos 凭据缓存文件：

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

验证用户的 Kerberos 凭据缓存中的票据是否有效且未过期：

```
1 klist
```

退出会话。

```
1 exit
```

直接登录 GNOME 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6：安装 Linux VDA](#)。

提示：

如果使用域帐户登录时成功执行用户身份验证，但无法显示您的桌面，请重新启动计算机并重试。

Quest Authentication Service

在域控制器上配置 **Quest** 假定您已在 Active Directory 域控制器上安装并配置了 Quest 软件，而且已获得管理权限，有权在 **Active Directory** 中创建计算机对象。

允许域用户登录 **Linux VDA** 计算机 为了让域用户能够在 Linux VDA 计算机上建立 HDX 会话：

1. 在 Active Directory 用户和计算机管理控制台中，为该用户帐户打开 Active Directory 用户属性。
2. 选择 **Unix Account** (Unix 帐户) 选项卡。
3. 选中 **Unix-enabled** (已启用 Unix)。
4. 将 **Primary GID Number** (首选 GID 编号) 设置为实际域用户组的组 ID。

注意：

这些说明相当于设置域用户，以便他们可以使用控制台、RDP、SSH 或任何其他远程协议进行登录。

在 Linux VDA 上配置 Quest

SELinux 策略强制实施解决方法 默认 RHEL 环境会强制实施 SELinux。此强制功能会影响 Quest 使用的 Unix 域套接字 IPC 机制，并阻止域用户登录。

解决此问题的最便捷的方法是禁用 SELinux。以 root 用户身份，编辑 **/etc/selinux/config** 并更改 **SELinux** 设置：

```
SELINUX=disabled
```

此更改要求重新启动计算机：

```
1 reboot
```

重要：

请谨慎使用此设置。禁用后重新启用 SELinux 策略强制实施会导致完全锁定，即便是对 root 用户和其他本地用户也是如此。

配置 **VAS** 守护程序 必须启用并断开自动续订 Kerberos 票据功能。必须禁用身份验证（脱机登录）功能：

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

此命令将续订间隔设为 9 小时 (32400 秒)，即比默认的 10 小时票据生命周期短 1 小时。请在票据生命周期较短的系统上设置较低的值。

配置 PAM 和 NSS 要启用通过 HDX 进行的域用户登录以及其他服务 (例如 su、ssh 和 RDP)，请运行以下命令以手动配置 PAM 和 NSS:

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

加入 Windows 域 使用 Quest **vastool** 命令将 Linux 计算机加入到 Active Directory 域中:

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

user 为有权将计算机加入 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称,例如 example.com。

加入域后重新启动 Linux 计算机。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (不论是 Windows 还是 Linux) 都要在 Active Directory 中有一个计算机对象。要验证加入了 Quest 的 Linux 计算机是否位于域中,请执行以下操作:

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

如果计算机已加入域,此命令会返回域名。如果计算机未加入任何域,则会显示以下错误:

```
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
```

验证用户身份验证 要验证 Quest 是否能够通过 PAM 对域用户进行身份验证,请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件:

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

验证 Kerberos 凭据缓存中的票据是否有效且未过期:

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

Centrify DirectControl

加入 **Windows** 域 安装 Centrify DirectControl Agent 后, 请使用 Centrify **adjoin** 命令将 Linux 计算机加入 Active Directory 域:

```
1 su -
2 adjoin -w -V -u user domain-name
```

user 参数为有权将计算机加入到 **Active Directory** 域的任何 Active Directory 域用户。**domain-name** 参数是将 Linux 计算机加入到的域的名称。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机 (不论是 Windows 还是 Linux) 都要在 **Active Directory** 中有一个计算机对象。要验证加入了 Centrify 的 Linux 计算机是否位于域中, 请执行以下操作:

```
1 su -
2
3 adinfo
```

验证 **Joined to domain** 值是否有效以及 **CentrifyDC mode** 是否返回了 **connected**。如果模式仍然卡在正在启动状态, 则表明 Centrify 客户端遇到了服务器连接或身份验证问题。

请使用以下命令可获得更全面的系统和诊断信息:

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

测试与各种 Active Directory 和 Kerberos 服务的连接。

```
1 adinfo --test
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

SSSD

配置 **Kerberos** 请运行以下命令以安装 Kerberos:

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

要配置 Kerberos, 请以 root 用户身份打开 **/etc/krb5.conf** 并配置以下参数:

注意：

根据您的 AD 基础结构配置 Kerberos。以下设置适用于单域、单林模型。

```
[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
```

此上下文中的 `domain-dns-name` 参数为 DNS 域名，例如 `example.com`。`REALM` 是大写的 Kerberos 领域名称，例如 `EXAMPLE.COM`。

加入域 必须将 SSSD 配置为使用 Active Directory 作为其身份提供程序并且使用 Kerberos 进行身份验证。但是，SSSD 并不提供用于加入域和管理系统 `keytab` 文件的 AD 客户端功能。可以改为使用 **adcli**、**realmd** 或 **Samba**。

注意：

本部分内容仅提供 **adcli** 和 **Samba** 的信息。

- 如果您使用 **adcli** 加入域，请完成以下步骤：

1. 安装 **adcli**。

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

2. 通过 **adcli** 加入域。

请使用以下命令删除旧系统 `keytab` 文件并加入域：

```
1 su -
2
3 rm -rf /etc/krb5.keytab
4
5 adcli join domain-dns-name -U user -H hostname-fqdn
```

user 是有权将计算机添加到域的域用户。**hostname-fqdn** 是计算机的 FQDN 格式的主机名。

需要 **-H** 选项，**adcli** 才能生成格式为 `host/hostname-fqdn@REALM` 的 SPN (Linux VDA 要求使用此格式)。

3. 验证域成员身份。

对于 Ubuntu 22.04 和 Ubuntu 20.04 计算机，请运行 `adcli testjoin` 命令来测试这些计算机是否已加入域。

- 如果使用 **Samba** 加入域，请完成以下步骤：

1. 安装软件包。

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

2. 配置 **Samba**。

打开 `/etc/samba/smb.conf` 并配置以下设置：

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
client signing = yes
client use spnego = yes
kerberos method = secrets and keytab
```

WORKGROUP 是 **REALM** 中的第一个字段，**REALM** 是大写的 Kerberos 领域名称。

3. 使用 **Samba** 加入域。

您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Windows 帐户。

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

其中，**REALM** 是大写的 Kerberos 领域名称，**user** 是有权将计算机添加到域的域用户。

设置 **SSSD** 安装或更新所需软件包：

如果尚未安装，请安装所需的 SSSD 和配置软件包：

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

如果已安装软件包，则建议进行更新：

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


注意:

默认情况下, Ubuntu 中的安装过程将自动配置 **nsswitch.conf** 和 PAM 登录模块。

配置 SSSD 启动 SSSD 守护程序之前, 需要更改 SSSD 配置。对于某些版本的 SSSD, 默认不安装 **/etc/sss-**
d/sss.conf 配置文件, 必须手动创建。以 root 用户身份创建或打开 **/etc/sss/sss.conf** 并配置以下设置:

```
[sss]
```

```
services = nss, pam
```

```
config_file_version = 2
```

```
domains = domain-dns-name
```

```
[domain/domain-dns-name]
```

```
id_provider = ad
```

```
access_provider = ad
```

```
auth_provider = krb5
```

```
krb5_realm = REALM
```

```
# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer
```

```
than 14 days
```

```
krb5_renewable_lifetime = 14d
```

```
# Set krb5_renew_interval to lower value if TGT ticket lifetime is
```

```
shorter than 2 hours
```

```
krb5_renew_interval = 1h
```

```
krb5_ccachedir = /tmp
```

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

```
# This ldap_id_mapping setting is also the default value
```

```
ldap_id_mapping = true
```

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

```
default_shell = /bin/bash
```

```
ad_gpo_map_remote_interactive = +ctxhdx
```

注意:

ldap_id_mapping 设置为 **true**, 以便 SSSD 本身负责将 Windows SID 映射到 Unix UID。否则, Active Directory 必须能够提供 POSIX 扩展程序。PAM 服务 **ctxhdx** 已添加到 **ad_gpo_map_remote_interactive**。

此上下文中的 **domain-dns-name** 参数为 DNS 域名，例如 example.com。**REALM** 是大写的 Kerberos 领域名称，例如 EXAMPLE.COM。不需要配置 NetBIOS 域名。

有关配置设置的信息，请参阅 `sssd.conf` 和 `sssd-ad` 的手册页。

SSSD 守护程序要求配置文件必须仅具有所有者读取权限：

```
1 sudo chmod 0600 /etc/sss/sss.conf
```

启动 **SSSD** 守护程序 运行以下命令立即启动 SSSD 守护程序，以及使守护程序在计算机启动时启动：

```
1 sudo systemctl start sssd
2
3 sudo systemctl enable sssd
```

PAM 配置 运行以下命令，确保选中 **SSS authentication**（SSS 身份验证）和 **Create home directory on login**（在登录时创建主目录）选项：

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

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 **Active Directory** 中有一个计算机对象。

- 如果使用 **adcli** 验证域成员身份，请运行 `sudo adcli info domain-dns-name` 命令以显示域信息。
- 如果使用 **Samba** 验证域成员身份，请运行 `sudo net ads testjoin` 命令验证计算机是否已加入到域，运行 `sudo net ads info` 命令验证额外的域和计算机对象信息。

验证 **Kerberos** 配置 要验证 Kerberos 是否已正确配置为可与 Linux VDA 配合使用，请验证系统 `keytab` 文件是否已创建并且包含有效密钥：

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos `kinit` 命令，使用这些密钥向域控制器验证计算机的身份：

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义，以免发生 shell 替换。在某些环境中，DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行，则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 已缓存：

```
1 sudo klist
```

验证用户身份验证 SSSD 不直接通过守护程序提供用于测试身份验证的命令行工具，只能通过 PAM 完成。

要验证 SSSD PAM 模块是否已正确配置，请使用以前未使用的域用户帐户登录 Linux VDA。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

验证 **klist** 命令返回的 Kerberos 票据是否适用于该用户并且尚未过期。

以 root 用户身份，验证是否已为前面的 **id -u** 命令返回的 uid 创建相应的票据缓存文件：

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

可以通过登录 KDE 或 Gnome Display Manager 执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

PBIS

下载所需的 **PBIS** 软件包

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

使 **PBIS** 安装脚本可执行

```
1 sudo chmod +x pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

运行 **PBIS** 安装脚本

```
1 sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

加入 **Windows** 域 您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户：

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

user 为有权将计算机添加到 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称，例如 example.com。

注意：要将 Bash 设置为默认 shell，请运行 **sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** 命令。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 [Active Directory](#) 中有一个计算机对象。要验证加入了 PBIS 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入某个域，此命令将返回有关当前加入的 AD 域和 OU 的信息。否则，仅显示主机名。

验证用户身份验证 要验证 PBIS 是否能够通过 PAM 对域用户进行身份验证，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

步骤 4: 安装.NET

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

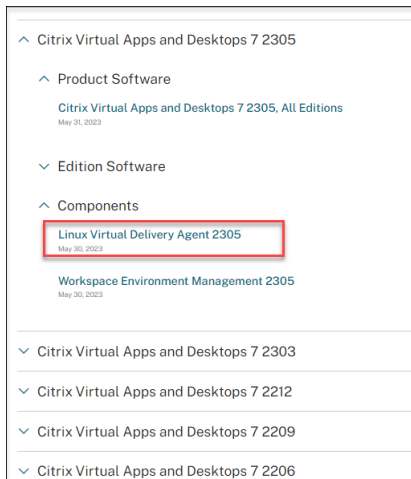
如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

安装 .NET 后，请运行 **which dotnet** 命令查找您的运行时路径。

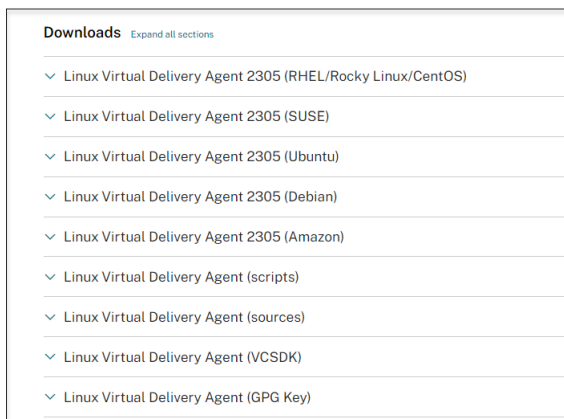
根据命令输出，设置 .NET Runtime 二进制文件路径。例如，如果命令输出为 /aa/bb/dotnet，请使用 /aa/bb 作为 .NET 二进制文件路径。

步骤 5: 下载 Linux VDA 软件包

1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。
3. 展开组件以查找 Linux VDA。例如：

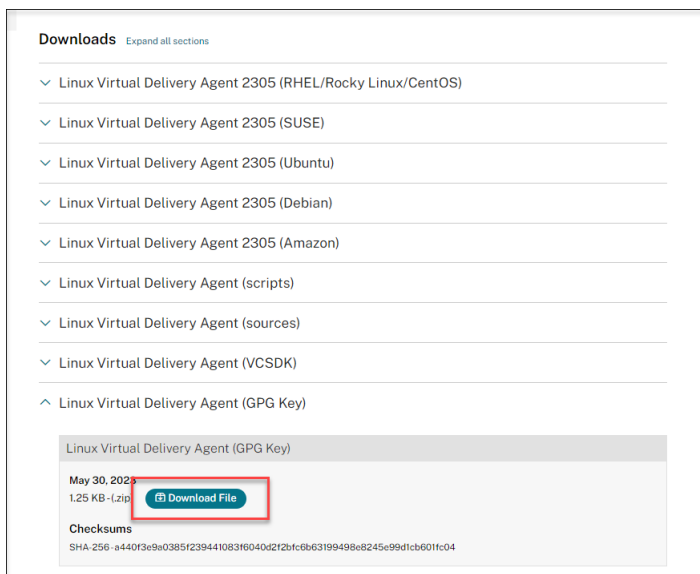


4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。

6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要验证 Linux VDA 软件包的完整性，请运行以下命令以将公钥导入到 DEB 数据库中以及检查软件包的完整性：

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

步骤 6：安装 Linux VDA

步骤 6a：安装 Linux VDA

使用 Debian 软件包管理器安装 Linux VDA 软件：

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

注意：

对于 GCP 上的 Ubuntu 20.04，请禁用 RDNS。为此，请在 /etc/krb5.conf 的 **[libdefaults]** 下添加 **rdns = false** 行。

Ubuntu 22.04 的 **Debian** 依赖项列表：

```
1 openjdk-11-jdk >= 11
2
3 imagemagick >= 8:6.9.11
4
5 libgtkmm-3.0-1v5 >= 3.24.5
6
7 ufw >= 0.36
8
9 ubuntu-desktop >= 1.481
10
11 libxrandr2 >= 2:1.5.2
12
13 libxtst6 >= 2:1.2.3
14
15 libxm4 >= 2.3.8
16
17 util-linux >= 2.37
18
19 gtk3-nocsd >= 3
20
21 bash >= 5.1
22
23 findutils >= 4.8.0
24
25 sed >= 4.8
26
27 cups >= 2.4
28
```

```
29 libmspack0 >= 0.10
30
31 ibus >= 1.5
32
33 libgoogle-perftools4 >= 2.9~
34
35 libpython3.10 >= 3.10~
36
37 libsasl2-modules-gssapi-mit >= 2.1.~
38
39 libnss3-tools >= 2:3.68
40
41 libqt5widgets5 >= 5.15~
42
43 libqrencode4 >= 4.1.1
44
45 libimlib2 >= 1.7.4
```

Ubuntu 20.04 的 Debian 依赖项列表:

```
1 openjdk-11-jdk >= 11
2
3 imagemagick >= 8:6.9.10
4
5 libgtkmm-3.0-1v5 >= 3.24.2
6
7 ufw >= 0.36
8
9 ubuntu-desktop >= 1.450
10
11 libxrandr2 >= 2:1.5.2
12
13 libxtst6 >= 2:1.2.3
14
15 libxm4 >= 2.3.8
16
17 util-linux >= 2.34
18
19 gtk3-nocsd >= 3
20
21 bash >= 5.0
22
23 findutils >= 4.7.0
24
25 sed >= 4.7
26
27 cups >= 2.3
28
29 libmspack0 >= 0.10
30
31 ibus >= 1.5
32
33 libgoogle-perftools4 >= 2.7~
```

```
34
35 libpython3.8 >= 3.8~
36
37 libsasl2-modules-gssapi-mit >= 2.1.~
38
39 libnss3-tools >= 2:3.49
40
41 libqt5widgets5 >= 5.7~
42
43 libqrencode4 >= 4.0.0
44
45 libimlib2 >= 1.6.1
```

注意：

有关此版本的 Linux VDA 支持的 Linux 发行版和 Xorg 版本的列表，请参阅[系统要求](#)。

步骤 6b: 升级 Linux VDA (可选)

Linux VDA 支持从最新版本进行升级。例如，您可以将 Linux VDA 从 2308 升级到 2311，从 1912 LTSR 升级到 2203 LTSR。

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

注意：

- 升级现有安装将覆盖 /etc/xdl 下的配置文件。在进行升级之前，请务必备份这些文件。
- 从 2407 版本开始，Linux VDA 委派包管理器 rpm 或 dpkg 在升级期间处理配置文件。下面介绍了 rpm 和 dpkg 如何与配置文件更改交互：
 - **rpm**: by default keeps the local version and saves the new version from the package with a .rpmnew extension.
 - **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as **.dpkg-new** or **.dpkg-dist**, use the following command:

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

步骤 7: 安装 NVIDIA GRID 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器（主机驱动程序），请参阅以下指南：

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

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序，请执行下面的常规步骤：

1. 确保来宾 VM 已关闭。
2. 在虚拟机管理程序控制面板中，为 VM 分配一个 GPU。
3. 启动 VM。
4. 在 VM 上安装来宾 VM 驱动程序。

步骤 8：配置 Linux VDA

注意：

在设置运行时环境之前，请确保已在您的操作系统中安装了 **en_US.UTF-8** 区域设置。如果该区域设置在您的操作系统中不可用，请运行 **sudo locale-gen en_US.UTF-8** 命令。对于 Debian，通过取消注释 **# en_US.UTF-8 UTF-8** 行来编辑 **/etc/locale.gen** 文件，然后运行 **sudo locale-gen** 命令。

安装软件包后，必须运行 **ctxsetup.sh** 脚本来配置 Linux VDA。执行任何更改之前，该脚本都会验证环境，确保所有依赖项都已安装。如有必要，可以随时重新运行该脚本以更改设置。

可以按照提示手动运行脚本，也可以采用预先配置的响应自动运行脚本。继续操作前，请查看该脚本的帮助信息：

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

提示配置

运行会提示各种问题的手动配置：

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

自动配置

自动安装时，设置脚本所需的选项可由环境变量提供。如果所需的所有变量都存在，则脚本不会提示用户提供任何信息，从而允许通过脚本完成安装过程。

支持的环境变量包括：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于已加入域的场景，请将其设置为 'n'。

- **CTX_XDL_AD_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA 要求使用 Kerberos 配置设置向 Delivery Controller 进行身份验证。Kerberos 配置根据系统上已安装和已配置的 Active Directory 集成工具确定。
- **CTX_XDL_DDC_LIST=' <list-ddc-fqdns>'** - Linux VDA 要求提供由空格分隔的 Delivery Controller 完全限定域名 (FQDN) 列表以用于向 Delivery Controller 进行注册。必须至少指定一个 FQDN 或 CNAME。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境, 将该值设置为 'y'。
- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro, 这是一组 GPU 加速技术, 旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro, 则要为 VDI 桌面 (单会话) 模式配置 VDA (即 CTX_XDL_VDI_MODE=' y')。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口 (默认端口 80 和 1494)。
- **CTX_XDL_DESKTOP_ENVIRONMENT='gnome/gnome-classic/kde/mate/xfce/' <none> '-** 指定要在会话中使用的 GNOME、GNOME Classic、KDE、MATE 或 Xfce 桌面环境。如果将其设置为 '<none>' ; 则将使用在 VDA 上配置的默认桌面。
您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息, 请参阅[桌面切换命令](#)和[系统托盘](#)。
- **CTX_XDL_DOTNET_RUNTIME_PATH='path-to-install-dotnet-runtime'** - 安装 .NET 以支持新的 Broker 代理服务 (ctxvda) 的路径。默认路径为 '/usr/bin'。
- **CTX_XDL_VDA_PORT='port-number'** - Linux VDA 通过 TCP/IP 端口与 Delivery Controller 通信。
- **CTX_XDL_SITE_NAME='<dns-name>'** - Linux VDA 通过 DNS 发现 LDAP 服务器。要将 DNS 搜索结果限制为本地站点, 应指定 DNS 站点名称。如果不需要, 请将其设置为 '<none>' 。
- **CTX_XDL_LDAP_LIST=' <list-ldap-servers>'** - Linux VDA 查询 DNS 来发现 LDAP 服务器。如果 DNS 无法提供 LDAP 服务记录, 您可以提供以空格分隔的 LDAP FQDN (带有 LDAP 端口) 列表。例如, ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268。要在 Active Directory 林中启用更快的 LDAP 查询, 请在域控制器上启用全局目录, 并将相关的 LDAP 端口号指定为 3268。默认情况下, 此变量设置为 '<none>' 。
- **CTX_XDL_SEARCH_BASE='search-base-set'** - Linux VDA 通过设置为 Active Directory 域根的搜索库来查询 LDAP (例如, DC=mycompany,DC=com)。为提高搜索性能, 可以指定搜索基础 (例如 OU=VDI,DC=mycompany,DC=com)。如果不需要, 请将其设置为 '<none>' 。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME=' y|n'** - Linux VDA 支持使用 DNS CNAME 记录指定 Delivery Controller 名称。

设置环境变量并运行配置脚本:

```

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

```

运行 `sudo` 命令时，键入 `-E` 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 `#!/bin/bash` 作为第一行来创建 shell 脚本文件。

另外，您可以使用单个命令指定所有参数：

```

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

```

删除配置更改

在某些情形下，您可能需要删除 `ctxsetup.sh` 脚本对配置所做的更改，但不卸载 Linux VDA 软件包。

继续操作前，请查看此脚本的帮助信息：

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

删除配置更改：

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

重要：

此脚本会从数据库删除所有配置数据，从而使 Linux VDA 无法使用。

配置日志

ctxsetup.sh 和 **ctxcleanup.sh** 脚本会在控制台上显示错误，并将其他信息写入配置日志文件 **/tmp/xdl.configure.log**。

重新启动 Linux VDA 服务，确保更改生效。

卸载 Linux VDA 软件

要检查 Linux VDA 是否已安装并查看已安装软件包的版本，请运行以下命令：

```
1 dpkg -l xendesktopvda
```

查看更多详细信息：

```
1 apt-cache show xendesktopvda
```

要卸载 Linux VDA 软件，请执行以下操作：

```
1 dpkg -r xendesktopvda
```

注意：

卸载 Linux VDA 软件会删除关联的 PostgreSQL 和其他配置数据。但是，不会删除在安装 Linux VDA 之前设置的 PostgreSQL 软件包和其他依赖软件包。

提示：

本节中的信息未介绍包括 PostgreSQL 在内的依赖软件包的删除操作。

步骤 9：运行 XDPing

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅 [XDPing](#)。

步骤 10：运行 Linux VDA

使用 **ctxsetup.sh** 脚本配置 Linux VDA 后，请使用以下命令控制 Linux VDA。

启动 **Linux VDA**:

启动 Linux VDA 服务:

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

停止 **Linux VDA**:

要停止 Linux VDA 服务, 请执行以下操作:

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

注意:

在停止 **ctxvda** 和 **ctxhdx** 服务之前, 请运行 **systemctl stop ctxmonitord** 命令以停止监视服务守护程序。否则, 监视服务守护程序将重新启动您停止的服务。

重新启动 **Linux VDA**:

重新启动 Linux VDA 服务:

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

检查 **Linux VDA** 状态:

要检查 Linux VDA 服务的运行状态, 请执行以下操作:

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

步骤 **11**: 创建计算机目录

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明, 请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制, 使得该过程不同于为 Windows VDA 计算机创建计算机目录:

- 对于操作系统, 请选择:
 - 多会话操作系统选项 (对于托管共享桌面交付模型)。

- 单会话操作系统选项（适用于 VDI 专用桌面交付模型）。

- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。

注意：

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是，选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

提示：

如果删除计算机后将其重新加入 Active Directory 域，则必须删除计算机，然后将其重新添加到计算机目录。

步骤 12：创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明，请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制：

- 确保所选 AD 用户和组已正确配置，能够登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的（匿名）用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

有关如何创建计算机目录和交付组的信息，请参阅[Citrix Virtual Apps and Desktops 7 2407](#)。

在 Debian 上手动安装 Linux VDA

August 20, 2024

重要：

对于全新安装，我们建议您使用[轻松安装](#)进行快速安装。轻松安装省时又省力，与本文中详述的手动安装相比，更不易于出错。

步骤 1：准备配置信息和 Linux 计算机

步骤 1a：设置主机名

为确保正确报告计算机的主机名，请更改 `/etc/hostname` 文件，使其仅包含计算机的主机名。

`hostname`

步骤 **1b**: 为主机名分配环回地址

确保计算机的 DNS 域名和完全限定域名 (FQDN) 正确地返回报告。方法是更改 `/etc/hosts` 文件的以下行以包含 FQDN 和主机名作为前两个条目:

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

例如:

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

从文件中的其他条目中删除对 `hostname-fqdn` 或 `hostname` 的任何其他引用。

注意:

Linux VDA 当前不支持 NetBIOS 名称截断。主机名不得超过 15 个字符。

提示:

只能使用字符 a-z、A-Z、0-9 和连字符 (-)。请避免使用下划线 (_)、空格和其他符号。主机名不得以数字开头和以连字符结尾。此规则也适用于 Delivery Controller 主机名。

步骤 **1c**: 检查主机名

重新启动计算机并验证主机名设置是否正确:

```
1 hostname
```

此命令仅返回计算机的主机名, 而不返回其 FQDN。

验证 FQDN 设置是否正确无误:

```
1 hostname -f
```

此命令返回计算机的 FQDN。

步骤 **1d**: 禁用多播 DNS

默认设置启用了多播 DNS (**mDNS**), 而这会导致名称解析结果不一致。

要禁用 **mDNS**, 请编辑 `/etc/nsswitch.conf` 并更改以下行:

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

更改为:

```
hosts: files dns
```

步骤 1e: 检查名称解析和服务可访问性

确认可以解析 FQDN 并对域控制器和 Delivery Controller 执行 ping 操作:

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

如果无法解析 FQDN 或 Ping 不通上述任一计算机, 请先检查相关步骤, 然后再继续。

步骤 1f: 配置时钟同步 (**chrony**)

确保 VDA、Delivery Controller 和域控制器之间的时钟始终精确同步至关重要。将 Linux VDA 托管为虚拟机 (VM) 可能会导致时钟偏差问题。出于此原因, 最好使用远程时间服务来同步时间。

安装 chrony:

```
1 apt-get install chrony
```

以 root 用户身份, 编辑 **/etc/chrony/chrony.conf** 并为每个远程时间服务器添加一个服务器条目:

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

在典型部署中, 时间从本地域控制器同步, 而不是直接从公共 NTP 池服务器同步。为域中的每个 Active Directory 域控制器添加一个服务器条目。

删除列出的任何其他 **server** 或 **pool** 条目, 包括环回 IP 地址、localhost 和公共服务器 ***.pool.ntp.org** 条目。

保存更改并重新启动 Chrony 守护程序:

```
1 sudo systemctl restart chrony
```

步骤 1g: 安装软件包

```
1 sudo apt-get install -y libsasl2-2
2
3 sudo apt-get install -y libgtk2.0-0
```

步骤 1h: 添加存储库以安装必要的依赖项

对于 Debian 11, 请将 **deb http://deb.debian.org/debian/ bullseye main** 行添加到 **/etc/apt/sources.list** 文件中。

步骤 1i: 安装并指定要使用的数据库

注意:

- 我们建议您仅将 SQLite 用于 VDI 模式，将 PostgreSQL 用于托管共享桌面交付模型。
- 对于轻松安装和 MCS，您可以指定使用 SQLite 或 PostgreSQL，而不必手动安装。除非通过 **/etc/xdl/db.conf** 另行指定，否则 Linux VDA 默认使用 PostgreSQL。
- 对于手动安装，您必须手动安装 SQLite、PostgreSQL 或者两者。如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

本部分内容介绍如何安装 PostgreSQL 和 SQLite 以及如何指定要使用的两者中的任意一个。

安装 PostgreSQL 请运行以下命令以安装 PostgreSQL:

```
1 sudo apt-get update
2
3 sudo apt-get install -y postgresql
4
5 sudo apt-get install -y libpostgresql-jdbc-java
```

请分别运行以下命令以在计算机启动时启动 PostgreSQL 或者立即启动:

```
1 sudo systemctl enable postgresql
2
3 sudo systemctl start postgresql
```

安装 SQLite 对于 Debian，请运行以下命令以安装 SQLite:

```
1 sudo apt-get install -y sqlite3
```

指定要使用的数据库 如果您同时安装了 SQLite 和 PostgreSQL，则可以在安装 Linux VDA 软件包后通过编辑 **/etc/xdl/db.conf** 来指定使用其中一个。

1. 运行 **/opt/Citrix/VDA/sbin/ctxcleanup.sh**。如果是全新安装，则省略此步骤。
2. 编辑 **/etc/xdl/db.conf** 以指定要使用的数据库。
3. 运行 **ctxsetup.sh**。

注意:

也可以使用 **/etc/xdl/db.conf** 配置 PostgreSQL 的端口号。

步骤 2: 准备虚拟机管理程序

在支持的虚拟机管理程序上将 Linux VDA 当作 VM 运行时，需要作出一些更改。根据正在使用的虚拟机管理程序平台做出以下更改。如果正在裸机硬件上运行 Linux 计算机，则无需作出任何更改。

修复 **XenServer**（以前称为 **Citrix Hypervisor**）上的时间同步问题

启用了 XenServer 时间同步功能时，在每个半虚拟化 Linux VM 中，您会遇到 NTP 和 XenServer 出现的问题。两者都试图管理系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。无需在 HVM 模式下进行任何更改。

如果您正在运行半虚拟化 Linux 内核，并且安装了 XenServer Tools，则可以检查 XenServer 时间同步功能是否存在以及是否已从 Linux VM 中启用：

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

此命令返回 0 或 1：

- 0 - 时间同步功能已启用，且必须禁用。
- 1 - 时间同步功能已禁用，无需采取任何操作。

如果 `/proc/sys/xen/independent_wallclock` 文件不存在，则不需要执行以下步骤。

如果已启用，请通过向该文件写入 1 来禁用时间同步功能：

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

要使此更改成为永久更改，并在重新启动后仍然有效，请编辑 `/etc/sysctl.conf` 文件并添加以下行：

```
xen.independent_wallclock = 1
```

要验证这些更改，请重新启动系统：

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

此命令返回值 1。

修复 **Microsoft Hyper-V** 上的时间同步问题

安装了 Hyper-V Linux 集成服务的 Linux VM 可以使用 Hyper-V 时间同步功能来使用主机操作系统的时间。为确保系统时钟始终精确可靠，请一同启用此功能与 NTP 服务。

从管理操作系统中：

1. 打开 Hyper-V 管理器控制台。

2. 对于 Linux VM 的设置，请选择 **Integration Services** (集成服务)。
3. 确保已选择 **Time synchronization** (时间同步)。

注意：

此方法与 VMware 和 XenServer (以前称为 Citrix Hypervisor) 不同，这两款产品会禁用主机时间同步功能以避免与 NTP 发生冲突。Hyper-V 时间同步可以与 NTP 时间同步共存并互补。

修复 ESX 和 ESXi 上的时间同步问题

启用 VMware 时间同步功能后，在每个半虚拟化的 Linux VM 中，您都会遇到 NTP 和虚拟机管理程序的问题。两者都试图同步系统时钟。为避免时钟与其他服务器不同步，请确保每个 Linux 客户机中的系统时钟都与 NTP 同步。这种情况要求禁用主机时间同步。

如果正在运行安装了 VMware Tools 的半虚拟化 Linux 内核：

1. 打开 vSphere Client。
2. 编辑 Linux VM 设置。
3. 在 **Virtual Machine Properties** (虚拟机属性) 对话框中，打开 **Options** (选项) 选项卡。
4. 选择 **VMware Tools**。
5. 在 **Advanced** (高级) 框中，取消选中 **Synchronize guest time with host** (与主机同步客户机时间)。

步骤 3：将 Linux VM 添加到 Windows 域中

以下方法可用于将 Linux 计算机添加到 Active Directory (AD) 域：

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

根据所选的方法，按说明执行操作。

注意：

为 Linux VDA 中的本地帐户和 AD 中的帐户使用相同的用户名时，会话启动可能会失败。

Samba Winbind

安装或更新所需软件包

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

在计算机启动时启用要启动的 **Winbind** 守护程序 Winbind 守护程序必须配置为在计算机启动时启动：

```
1 sudo systemctl enable winbind
```

注意：

确保 **winbind** 脚本位于 `/etc/init.d` 下。

配置 **Kerberos** 以 root 用户身份，打开 `/etc/krb5.conf` 并配置以下设置：

注意：

根据您的 AD 基础结构配置 Kerberos。以下设置适用于单域、单林模型。

```
[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
```

此上下文中的 **domain-dns-name** 参数为 DNS 域名，例如 **example.com**。**REALM** 是大写的 Kerberos 领域名称，例如 **EXAMPLE.COM**。

配置 **Winbind** 身份验证 运行 `vim /etc/samba/smb.conf` 命令打开 `/etc/samba/smb.conf`，然后配置以下设置：

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
encrypt passwords = yes
idmap config *:range = 16777216-33554431
```

```
kerberos method = secrets and keytab
winbind refresh tickets = yes
template shell = /bin/bash
```

WORKGROUP 是 **REALM** 中的第一个字段，**REALM** 是大写的 Kerberos 领域名称。

配置 **nsswitch** 打开 **/etc/nsswitch.conf** 并将 **winbind** 附加到以下行：

```
passwd: files systemd winbind
group: files systemd winbind
```

加入 **Windows** 域 您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户：

```
1 sudo net ads join <Kerberos realm name in uppercase> -U <domain user
  with permission to add computers to the domain>
```

重新启动 **Winbind**

```
1 sudo systemctl restart winbind
```

为 **Winbind** 配置 **PAM** 运行以下命令，确保选中 **Winbind NT/Active Directory authentication** (Winbind NT/Active Directory 身份验证) 和 **Create home directory on login** (在登录时创建主目录) 选项：

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

提示：

仅当计算机加入域后，**winbind** 守护程序才会始终保持运行。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（不论是 Windows 还是 Linux）都要在 **Active Directory** 中有一个计算机对象。

运行 **Samba** 的 **net ads** 命令以验证计算机是否已加入域：

```
1 sudo net ads testjoin
```

运行以下命令验证额外的域和计算机对象信息：

```
1 sudo net ads info
```

验证 **Kerberos** 配置 要验证 Kerberos 是否已正确配置为可与 Linux VDA 配合使用，请验证系统 **keytab** 文件是否已创建并且包含有效密钥：

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令，以使用这些密钥向域控制器验证计算机的身份：

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义，以免发生 shell 替换。在某些环境中，DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行，则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存：

```
1 sudo klist
```

请使用以下命令检查计算机的帐户详细信息：

```
1 sudo net ads status
```

验证用户身份验证 使用 **wbinfo** 工具验证是否可向域验证域用户的身份：

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

这里指定的域为 AD 域名，而不是 Kerberos 领域名称。对于 bash shell，必须使用另一个反斜杠对反斜杠 (\) 字符进行转义。此命令返回一条成功或失败消息。

要验证 Winbind PAM 模块是否已正确配置，请使用以前未使用的域用户帐户登录 Linux VDA。

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

注意：

要成功运行 SSH 命令，请确保 SSH 已启用并正常运行。

验证是否为 **id -u** 命令返回的 uid 创建了对应的 Kerberos 凭据缓存文件：

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

验证用户的 Kerberos 凭据缓存中的票据是否有效且未过期：

```
1 klist
```

退出会话。

```
1 exit
```

直接登录 GNOME 或 KDE 控制台也可以执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

提示:

如果使用域帐户登录时成功执行用户身份验证，但无法显示您的桌面，请重新启动计算机，然后重试。

Quest Authentication Service

在域控制器上配置 **Quest** 假定您已在 Active Directory 域控制器上安装并配置了 Quest 软件，而且已获得管理权限，有权在 [Active Directory](#) 中创建计算机对象。

允许域用户登录 **Linux VDA** 计算机 为了让域用户能够在 Linux VDA 计算机上建立 HDX 会话:

1. 在 Active Directory 用户和计算机管理控制台中，为该用户帐户打开 Active Directory 用户属性。
2. 选择 **Unix Account** (Unix 帐户) 选项卡。
3. 选中 **Unix-enabled** (已启用 Unix)。
4. 将 **Primary GID Number** (首选 GID 编号) 设置为实际域用户组的组 ID。

注意:

这些说明相当于设置域用户，以便他们可以使用控制台、RDP、SSH 或任何其他远程协议进行登录。

在 Linux VDA 上配置 Quest

SELinux 策略强制实施解决方法 默认 RHEL 环境会强制实施 SELinux。此强制功能会影响 Quest 使用的 Unix 域套接字 IPC 机制，并阻止域用户登录。

解决此问题的最便捷的方法是禁用 SELinux。以 root 用户身份，编辑 `/etc/selinux/config` 并更改 **SELinux** 设置:

```
SELINUX=disabled
```

此更改要求重新启动计算机:

```
1 reboot
```

重要:

请谨慎使用此设置。禁用后重新启用 SELinux 策略强制实施会导致完全锁定，即便是对 root 用户和其他本地用户也是如此。

配置 **VAS** 守护程序 必须启用并断开自动续订 Kerberos 票据功能。必须禁用身份验证 (脱机登录) 功能:

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

此命令将续订间隔设为 9 小时（32400 秒），即比默认的 10 小时票据生命周期短 1 小时。请在票据生命周期较短的系统上设置较低的值。

配置 PAM 和 NSS 要启用通过 HDX 进行的域用户登录以及其他服务（例如 su、ssh 和 RDP），请运行以下命令以手动配置 PAM 和 NSS：

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

加入 Windows 域 使用 Quest **vastool** 命令将 Linux 计算机加入到 Active Directory 域中：

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

user 为有权将计算机加入 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称，例如 example.com。

加入域后重新启动 Linux 计算机。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（不论是 Windows 还是 Linux）都要在 Active Directory 中有一个计算机对象。要验证加入了 Quest 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入域，此命令会返回域名。如果计算机未加入任何域，则会显示以下错误：

```
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
```

验证用户身份验证 要验证 Quest 是否能够通过 PAM 对域用户进行身份验证，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：


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

验证 Kerberos 凭据缓存中的票据是否有效且未过期：

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

Centrify DirectControl

加入 **Windows** 域 安装 Centrify DirectControl Agent 后，请使用 Centrify `adjoin` 命令将 Linux 计算机加入 Active Directory 域：

```
1 su -
2 adjoin -w -V -u user domain-name
```

user 参数为有权将计算机加入到 Active Directory 域的任何 Active Directory 域用户。**domain-name** 参数是将 Linux 计算机加入到的域的名称。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（不论是 Windows 还是 Linux）都要在 Active Directory 中有一个计算机对象。要验证加入了 Centrify 的 Linux 计算机是否位于域中，请执行以下操作：

```
1 su -
2
3 adinfo
```

验证 **Joined to domain** 值是否有效以及 **CentrifyDC mode** 是否返回了 **connected**。如果模式仍然卡在正在启动状态，则表明 Centrify 客户端遇到了服务器连接或身份验证问题。

请使用以下命令可获得更全面的系统和诊断信息：

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

测试与各种 Active Directory 和 Kerberos 服务的连接。

```
1 adinfo --test
```

在进行域加入验证后继续执行[步骤 6: 安装 Linux VDA](#)。

SSSD

配置 **Kerberos** 请运行以下命令以安装 Kerberos:

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

要配置 Kerberos, 请以 root 用户身份打开 **/etc/krb5.conf** 并配置以下参数:

注意:

根据您的 AD 基础结构配置 Kerberos。以下设置适用于单域、单林模型。

```
[libdefaults]
default_realm = REALM
dns_lookup_kdc = false
rdns = false

[realms]
REALM = {
admin_server = domain-controller-fqdn
kdc = domain-controller-fqdn
}

[domain_realm]
domain-dns-name = REALM
.domain-dns-name = REALM
```

此上下文中的 `domain-dns-name` 参数为 DNS 域名, 例如 `example.com`。REALM 是大写的 Kerberos 领域名称, 例如 `EXAMPLE.COM`。

加入域 必须将 SSSD 配置为使用 Active Directory 作为其身份提供程序并且使用 Kerberos 进行身份验证。但是, SSSD 并不提供用于加入域和管理系统 keytab 文件的 AD 客户端功能。可以改为使用 **adcli**、**realmd** 或 **Samba**。

注意:

本部分内容仅提供 **adcli** 和 **Samba** 的信息。

- 如果您使用 **adcli** 加入域, 请完成以下步骤:

1. 安装 **adcli**。

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

2. 通过 **adcli** 加入域。

请使用以下命令删除旧系统 keytab 文件并加入域：

```
1 su -
2
3 rm -rf /etc/krb5.keytab
4
5 adcli join domain-dns-name -U user -H hostname-fqdn
```

user 是拥有将计算机添加到域的域用户。**hostname-fqdn** 是计算机的 FQDN 格式的主机名。

需要 **-H** 选项，**adcli** 才能生成格式为 host/hostname-fqdn@REALM 的 SPN（Linux VDA 要求使用此格式）。

3. 验证系统 keytab。

运行 `sudo klist -ket` 命令以确保系统 keytab 文件已创建。

验证每个键的时间戳是否与将计算机加入域的时间相匹配。

- 如果使用 **Samba** 加入域，请完成以下步骤：

1. 安装软件包。

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

2. 配置 **Samba**。

打开 `/etc/samba/smb.conf` 并配置以下设置：

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
client signing = yes
client use spnego = yes
kerberos method = secrets and keytab
```

`WORKGROUP` 是 `REALM` 中的第一个字段，`REALM` 是大写的 Kerberos 领域名称。

3. 使用 **Samba** 加入域。

您的域控制器必须可访问，并且您必须具有有权将计算机添加到域的 Windows 帐户。

```
1 sudo net ads join <the Kerberos realm name in uppercase> -U <
   domain user with permission to add computers to the domain>
```

设置 **SSSD** 安装或更新所需软件包:

如果尚未安装, 请安装所需的 SSSD 和配置软件包:

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

如果已安装软件包, 则建议进行更新:

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

配置 **SSSD** 启动 SSSD 守护程序之前, 需要更改 SSSD 配置。对于某些版本的 SSSD, 默认不安装 **/etc/sss-d/sss.conf** 配置文件, 必须手动创建。以 root 用户身份创建或打开 **/etc/sss-d/sss.conf** 并配置以下设置:

```
[sss-d]
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
```

注意：

`ldap_id_mapping` 设置为 **true**，以便 SSSD 本身负责将 Windows SID 映射到 Unix UID。否则，**Active Directory** 必须能够提供 POSIX 扩展程序。PAM 服务 `ctxhdx` 已添加到 `ad_gpo_map_remote_interactive`。

此上下文中的 **domain-dns-name** 参数为 DNS 域名，例如 `example.com`。**REALM** 是大写的 Kerberos 领域名称，例如 `EXAMPLE.COM`。不需要配置 NetBIOS 域名。

有关配置设置的信息，请参阅 `sssd.conf` 和 `sssd-ad` 的手册页。

SSSD 守护程序要求配置文件必须仅具有所有者读取权限：

```
1 sudo chmod 0600 /etc/sss/sss.conf
```

启动 **SSSD** 守护程序 运行以下命令立即启动 SSSD 守护程序，以及使守护程序在计算机启动时启动：

```
1 sudo systemctl start sssd
2
3 sudo systemctl enable sssd
```

PAM 配置 运行以下命令，确保选中 **SSS authentication**（SSS 身份验证）和 **Create home directory on login**（在登录时创建主目录）选项：

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

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 **Active Directory** 中有一个计算机对象。

- 如果使用 **adcli** 验证域成员身份，请运行 `sudo adcli info domain-dns-name` 命令以显示域信息。
- 如果使用 **Samba** 验证域成员身份，请运行 `sudo net ads testjoin` 命令验证计算机是否已加入到域，运行 `sudo net ads info` 命令验证额外的域和计算机对象信息。

验证 **Kerberos** 配置 要验证 Kerberos 是否已正确配置为可与 Linux VDA 配合使用，请验证系统 `keytab` 文件是否已创建并且包含有效密钥：

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos `kinit` 命令，使用这些密钥向域控制器验证计算机的身份：

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义, 以免发生 shell 替换。在某些环境中, DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行, 则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 已缓存:

```
1 sudo klist
```

验证用户身份验证 SSSD 不直接通过守护程序提供用于测试身份验证的命令行工具, 只能通过 PAM 完成。

要验证 SSSD PAM 模块是否已正确配置, 请使用以前未使用的域用户帐户登录 Linux VDA。

```
1 ssh localhost -l domain\username
2
3 id -u
4
5 klist
6
7 exit
```

验证 **klist** 命令返回的 Kerberos 票据是否适用于该用户并且尚未过期。

以 root 用户身份, 验证是否已为前面的 **id -u** 命令返回的 uid 创建相应的票据缓存文件:

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

可以通过登录 KDE 或 Gnome Display Manager 执行类似的测试。在进行域加入验证后继续执行 [步骤 6: 安装 Linux VDA](#)。

PBIS

下载所需的 **PBIS** 软件包

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

使 **PBIS** 安装脚本可执行

```
1 sudo chmod +x pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

运行 **PBIS** 安装脚本

```
1 sudo sh pbis-open-9.1.0.551.linux.x86_64.deb.sh
```

加入 **Windows** 域 您的域控制器必须可访问, 并且您必须具有有权将计算机添加到域的 Active Directory 用户帐户:

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

user 为有权将计算机添加到 Active Directory 域的域用户。**domain-name** 为域的 DNS 名称, 例如 example.com。

注意：要将 Bash 设置为默认 shell，请运行 **sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** 命令。

验证域成员身份 Delivery Controller 要求所有 VDA 计算机（Windows 和 Linux VDA）都要在 **Active Directory** 中有一个计算机对象。要验证加入了 PBIS 的 Linux 计算机是否位于域中，请执行以下操作：

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

如果计算机已加入某个域，此命令将返回有关当前加入的 AD 域和 OU 的信息。否则，仅显示主机名。

验证用户身份验证 要验证 PBIS 是否能够通过 PAM 对域用户进行身份验证，请使用以前未使用的域用户帐户登录 Linux VDA。

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

验证是否为 **id -u** 命令返回的 UID 创建了对应的 Kerberos 凭据缓存文件：

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

退出会话。

```
1 exit
```

在进行域加入验证后继续执行 **步骤 6：安装 Linux VDA**。

步骤 4：安装 .NET

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .ASP.NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

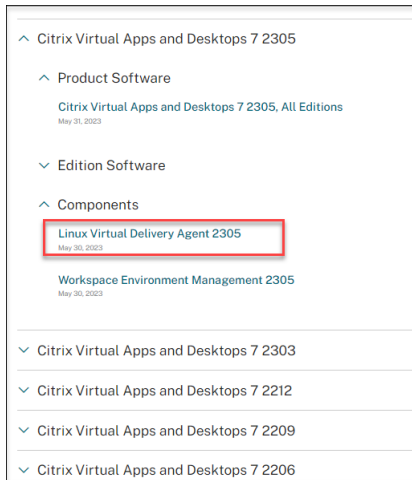
安装 .NET 后，请运行 **which dotnet** 命令查找您的运行时路径。

根据命令输出，设置 .NET Runtime 二进制文件路径。例如，如果命令输出为 `/aa/bb/dotnet`，请使用 `/aa/bb` 作为 .NET 二进制文件路径。

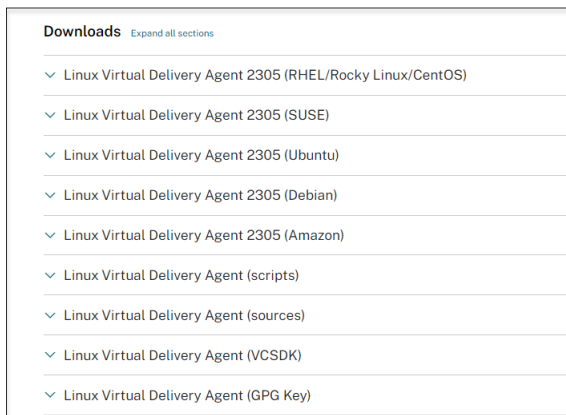
步骤 5：下载 Linux VDA 软件包

1. 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
2. 展开适当版本的 Citrix Virtual Apps and Desktops。

3. 展开组件以查找 Linux VDA。例如：

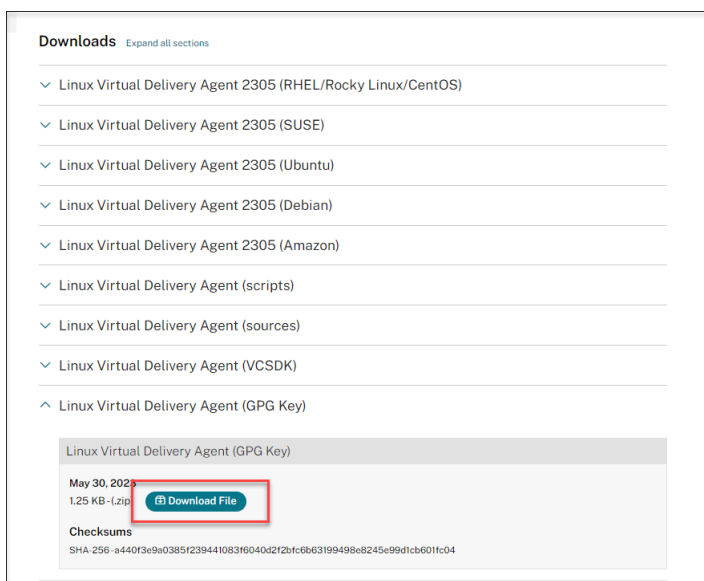


4. 单击 Linux VDA 链接可访问 Linux VDA 下载内容。



5. 下载与您的 Linux 发行版匹配的 Linux VDA 软件包。

6. 下载可用于验证 Linux VDA 软件包完整性的 GPG 公钥。例如：



要验证 Linux VDA 软件包的完整性，请运行以下命令以将公钥导入到 DEB 数据库中以及检查软件包的完整性：

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

步骤 6: 安装 Linux VDA

步骤 6a: 安装 Linux VDA

使用 Debian 软件包管理器安装 Linux VDA 软件：

对于 **Debian 12.5**：

```
1 sudo dpkg -i xendesktopvda_<version>.debian12_amd64.deb
```

对于 **Debian 11**：

```
1 sudo dpkg -i xendesktopvda_<version>.debian11_amd64.deb
```

Debian 12.5 的依赖项列表：

```
1 openjdk-17-jdk >= 17
2
3 imagemagick >= 8:6.9.11
4
5 ufw >= 0.36
6
7 desktop-base >= 12.0.6
8
9 libxrandr2 >= 2:1.5.2
10
```

```
11 libxtst6 >= 2:1.2.3
12
13 libxm4 >= 2.3.8
14
15 util-linux >= 2.38
16
17 gtk3-nocsd >= 3
18
19 bash >= 5.2
20
21 findutils >= 4.9.0
22
23 sed >= 4.9
24
25 cups >= 2.4
26
27 ghostscript >= 10.0.0~
28
29 libmspack0 >= 0.11
30
31 ibus >= 1.5
32
33 libgoogle-perftools4 >= 2.10~
34
35 libpython3.11 >= 3.11~
36
37 libsasl2-modules-gssapi-mit >= 2.1.~
38
39 libnss3-tools >= 2:3.87
40
41 libqt5widgets5 >= 5.15~
42
43 mutter >= 43.8
44
45 libqrencode4 >= 4.1.1
46
47 libimlib2 >= 1.10.0
48
49 libfuse2 >= 2.9.9
50
51 pulseaudio-utils >= 16.1
```

Debian 11 的依赖项列表:

```
1 libnss3-tools >= 2:3.61
2
3 libfuse2 >= 2.9
4
5 openjdk-11-jdk >= 11
6
7 imagemagick >= 8:6.9.10
8
9 ufw >= 0.36
```

```
10
11 desktop-base >= 10.0.2
12
13 libxrandr2 >= 2:1.5.1
14
15 libxtst6 >= 2:1.2.3
16
17 libxm4 >= 2.3.8
18
19 util-linux >= 2.33
20
21 gtk3-nocsd >= 3
22
23 bash >= 5.0
24
25 findutils >= 4.6.0
26
27 sed >= 4.7
28
29 cups >= 2.2
30
31 ghostscript >= 9.53~
32
33 libmspack0 >= 0.10
34
35 ibus >= 1.5
36
37 libgoogle-perftools4 >= 2.7~
38
39 libpython3.9 >= 3.9~
40
41 libsasl2-modules-gssapi-mit >= 2.1.~
42
43 libqt5widgets5 >= 5.5~
44
45 mutter >= 3.38.6~
46
47 libqrencode4 >= 4.0.0
48
49 libimlib2 >= 1.5.1
```

注意：

有关此版本的 Linux VDA 支持的 Linux 发行版和 Xorg 版本的列表，请参阅[系统要求](#)。

步骤 6b：升级 Linux VDA（可选）

Linux VDA 支持从最新版本进行升级。例如，您可以将 Linux VDA 从 2308 升级到 2311，从 1912 LTSR 升级到 2203 LTSR。

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

```
2
3 sudo apt-get install -f
```

注意：

- 升级现有安装将覆盖 /etc/xdl 下的配置文件。在进行升级之前，请务必备份这些文件。
- 从 2407 版本开始，Linux VDA 委派包管理器 **rpm** 或 **dpkg** 在升级期间处理配置文件。下面介绍了 rpm 和 dpkg 如何与配置文件更改交互：
 - **rpm**: by default keeps the local version and saves the new version from the package with a .rpmnew extension.
 - **dpkg**: interactively prompts you with a choice on how to proceed. To silently upgrade the Linux VDA while retaining your local configuration file and saving the new package version as **.dpkg-new** or **.dpkg-dist**, use the following command:

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

步骤 7：安装 NVIDIA GRID 驱动程序

启用 HDX 3D Pro 要求您在虚拟机管理程序和 VDA 计算机上安装 NVIDIA GRID 驱动程序。

要在特定虚拟机管理程序上安装和配置 NVIDIA GRID 虚拟 GPU 管理器（主机驱动程序），请参阅以下指南：

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

要安装和配置 NVIDIA GRID 来宾 VM 驱动程序，请执行下面的常规步骤：

1. 确保来宾 VM 已关闭。
2. 在虚拟机管理程序控制面板中，为 VM 分配一个 GPU。
3. 启动 VM。
4. 在 VM 上安装来宾 VM 驱动程序。

步骤 8：配置 Linux VDA

注意：

在设置运行时环境之前，请确保已在您的操作系统中安装了 **en_US.UTF-8** 区域设置。如果该区域设置在您的操作系统中不可用，请运行 **sudo locale-gen en_US.UTF-8** 命令。对于 Debian，通过取消注释 **# en_US.UTF-8 UTF-8** 行来编辑 **/etc/locale.gen** 文件，然后运行 **sudo locale-gen** 命令。

安装软件包后，必须运行 `ctxsetup.sh` 脚本来配置 Linux VDA。执行任何更改之前，该脚本都会验证环境，确保所有依赖项都已安装。如有必要，可以随时重新运行该脚本以更改设置。

可以按照提示手动运行脚本，也可以采用预先配置响应自动运行脚本。继续操作前，请查看该脚本的帮助信息：

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

提示配置

运行会提示各种问题的手动配置：

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

自动配置

自动安装时，设置脚本所需的选项可由环境变量提供。如果所需的所有变量都存在，则脚本不会提示用户提供任何信息，从而允许通过脚本完成安装过程。

支持的环境变量包括：

- **CTX_XDL_NON_DOMAIN_JOINED=' y|n'** - 是否将计算机加入域。默认值为 'n'。对于已加入域的场景，请将其设置为 'y'。
- **CTX_XDL_AD_INTEGRATION=' winbind|sssd|centrify|pbis|quest'** - Linux VDA 要求使用 Kerberos 配置设置向 Delivery Controller 进行身份验证。Kerberos 配置根据系统上已安装和已配置的 Active Directory 集成工具确定。
- **CTX_XDL_DDC_LIST=' <list-ddc-fqdns>'** - Linux VDA 要求提供由空格分隔的 Delivery Controller 完全限定域名 (FQDN) 列表以用于向 Delivery Controller 进行注册。必须至少指定一个 FQDN 或 CNAME。
- **CTX_XDL_VDI_MODE=' y|n'** - 将计算机配置为专用桌面交付模型 (VDI) 还是托管共享桌面交付模型。对于 HDX 3D Pro 环境，将该值设置为 'y'。
- **CTX_XDL_HDX_3D_PRO=' y|n'** - Linux VDA 支持 HDX 3D Pro，这是一组 GPU 加速技术，旨在优化富图形应用程序的虚拟化水平。如果选择了 HDX 3D Pro，则要为 VDI 桌面（单会话）模式配置 VDA（即 `CTX_XDL_VDI_MODE=' y'`）。
- **CTX_XDL_START_SERVICE=' y|n'** - 确定是否在完成配置后启动 Linux VDA 服务。
- **CTX_XDL_REGISTER_SERVICE=' y|n'** - 在启动计算机后启动 Linux Virtual Desktop 服务。
- **CTX_XDL_ADD_FIREWALL_RULES=' y|n'** - Linux VDA 服务要求允许传入网络连接通过系统防火墙。您可以在系统防火墙中自动为 Linux Virtual Desktop 打开所需端口（默认端口 80 和 1494）。
- **CTX_XDL_DESKTOP_ENVIRONMENT='gnome/gnome-classic/kde/mate/' <none>'** - 指定要在会话中使用的 GNOME、GNOME Classic 或 MATE 桌面环境。如果将其设置为 '<none>'；则将使用在 VDA 上配置的默认桌面。

您还可以通过运行命令或使用系统托盘在桌面环境之间切换。有关详细信息，请参阅[桌面切换命令](#)和[系统托盘](#)。

- **CTX_XDL_DOTNET_RUNTIME_PATH=***path-to-install-dotnet-runtime* - 安装.NET 以支持新的 Broker 代理服务 (ctxvda) 的路径。默认路径为 `‘/usr/bin’`。
- **CTX_XDL_VDA_PORT=***port-number* - Linux VDA 通过 TCP/IP 端口与 Delivery Controller 通信。
- **CTX_XDL_SITE_NAME=***<dns-name>* - Linux VDA 通过 DNS 发现 LDAP 服务器。要将 DNS 搜索结果限制为本地站点，应指定 DNS 站点名称。如果不需要，请将其设置为 `‘<none>’`。
- **CTX_XDL_LDAP_LIST=**`‘<list-ldap-servers>’` - Linux VDA 查询 DNS 来发现 LDAP 服务器。如果 DNS 无法提供 LDAP 服务记录，您可以提供以空格分隔的 LDAP FQDN（带有 LDAP 端口）列表。例如，`ad1.mycompany.com:389 ad2.mycompany.com:3268 ad3.mycompany.com:3268`。要在 Active Directory 林中启用更快的 LDAP 查询，请在域控制器上启用全局目录，并将相关的 LDAP 端口号指定为 3268。默认情况下，此变量设置为 `‘<none>’`。
- **CTX_XDL_SEARCH_BASE=***search-base-set* - Linux VDA 通过设置为 Active Directory 域根的搜索库来查询 LDAP（例如，`DC=mycompany,DC=com`）。为提高搜索性能，可以指定搜索基础（例如 `OU=VDI,DC=mycompany,DC=com`）。如果不需要，请将其设置为 `‘<none>’`。
- **CTX_XDL_SUPPORT_DDC_AS_CNAME=**`‘y|n’` - Linux VDA 支持使用 DNS CNAME 记录指定 Delivery Controller 名称。

设置环境变量并运行配置脚本：

```

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

```

运行 `sudo` 命令时，键入 `-E` 选项以将现有环境变量传递给其创建的新 shell。我们建议您使用前面的命令并加上 `#!/bin/bash` 作为第一行来创建 shell 脚本文件。

另外，您可以使用单个命令指定所有参数：

```

1 sudo CTX_XDL_NON_DOMAIN_JOINED='n' \
2 CTX_XDL_AD_INTEGRATION=sssd|winbind|centrify|pbis|quest \
3 CTX_XDL_DDC_LIST='<list-ddc-fqdns>' \

```

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

删除配置更改

在某些情形下，您可能需要删除 **ctxsetup.sh** 脚本对配置所做的更改，但不卸载 Linux VDA 软件包。

继续操作前，请查看此脚本的帮助信息：

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

删除配置更改：

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

重要：

此脚本会从数据库删除所有配置数据，从而使 Linux VDA 无法使用。

配置日志

ctxsetup.sh 和 **ctxcleanup.sh** 脚本会在控制台上显示错误，并将其他信息写入配置日志文件 **/tmp/xdl.configure.log**。

重新启动 Linux VDA 服务，确保更改生效。

卸载 Linux VDA 软件

要检查 Linux VDA 是否已安装并查看已安装软件包的版本，请运行以下命令：

```
1 dpkg -l xendesktopvda
```

查看更多详细信息：

```
1 apt-cache show xendesktopvda
```

要卸载 Linux VDA 软件，请执行以下操作：

```
1 dpkg -r xendesktopvda
```

注意：

卸载 Linux VDA 软件会删除关联的 PostgreSQL 和其他配置数据。但是，不会删除在安装 Linux VDA 之前设置的 PostgreSQL 软件包和其他依赖软件包。

提示：

本节中的信息未介绍包括 PostgreSQL 在内的依赖软件包的删除操作。

步骤 9：运行 XDPing

请运行 `sudo /opt/Citrix/VDA/bin/xdping` 以检查 Linux VDA 环境存在的常见配置问题。有关详细信息，请参阅 [XDPing](#)。

步骤 10：运行 Linux VDA

使用 `ctxsetup.sh` 脚本配置 Linux VDA 后，请使用以下命令控制 Linux VDA。

启动 **Linux VDA**：

启动 Linux VDA 服务：

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

停止 **Linux VDA**：

要停止 Linux VDA 服务，请执行以下操作：

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

注意：

在停止 `ctxvda` 和 `ctxhdx` 服务之前，请运行 `systemctl stop ctxmonitord` 命令以停止监视服务守护程序。否则，监视服务守护程序将重新启动您停止的服务。

重新启动 **Linux VDA**：

重新启动 Linux VDA 服务：

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



```
4
5 sudo systemctl restart ctxvda
```

检查 **Linux VDA** 状态：

要检查 Linux VDA 服务的运行状态，请执行以下操作：

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

步骤 11：创建计算机目录

创建计算机目录和添加 Linux VDA 计算机的过程与传统的 Windows VDA 方法类似。有关如何完成这些任务的更加详细的说明，请参阅[创建计算机目录](#)和[管理计算机目录](#)。

创建包含 Linux VDA 计算机的计算机目录时会面临一些限制，使得该过程不同于为 Windows VDA 计算机创建计算机目录：

- 对于操作系统，请选择：
 - 多会话操作系统选项（对于托管共享桌面交付模型）。
 - 单会话操作系统选项（适用于 VDI 专用桌面交付模型）。
- 请勿在同一个计算机目录中混合使用 Linux 和 Windows VDA 计算机。

注意：

早期版本的 Citrix Studio 不支持“Linux 操作系统”的概念。但是，选择 **Windows Server** 操作系统或服务器操作系统选项等同于使用托管共享桌面交付模型。选择 **Windows** 桌面操作系统或桌面操作系统选项等同于使用每计算机一个用户交付模型。

提示：

如果删除计算机后将其重新加入 Active Directory 域，则必须删除计算机，然后将其重新添加到计算机目录。

步骤 12：创建交付组

创建交付组和添加包含 Linux VDA 计算机的计算机目录的过程与 Windows VDA 计算机几乎相同。有关如何完成这些任务的更加详细的说明，请参阅[创建交付组](#)。

创建含有 Linux VDA 计算机目录的交付组时会面临以下限制：

- 确保所选 AD 用户和组已正确配置，能够登录到 Linux VDA 计算机。
- 请勿允许未经身份验证的（匿名）用户登录。
- 请勿在交付组中混入含有 Windows 计算机的计算机目录。

有关如何创建计算机目录和交付组的信息，请参阅 [Citrix Virtual Apps and Desktops 7 2407](#)。

配置

July 15, 2024

本部分详细介绍了 Linux VDA 的功能，包括功能说明、配置和故障排除。

管理

July 15, 2024

本部分内容包含以下主题：

- [Linux VDA 数据收集计划](#)
- [HDX Insight](#)
- [与 Citrix Telemetry Service 集成](#)
- [面向 Citrix DaaS Standard for Azure 的 Linux VDA 自我更新](#)
- [Linux VM 和 Linux 会话指标](#)
- [日志收集](#)
- [会话影子处理](#)
- [监视服务守护程序](#)
- [故障排除](#)
- [其他](#)
 - [适用于 HTML5 的 Citrix Workspace 应用程序支持](#)
 - [创建 **Python3** 虚拟环境](#)
 - [将 NIS 与 Active Directory 集成](#)
 - [IPv6](#)
 - [LDAPS](#)
 - [Xauthority](#)

Linux VDA 数据收集计划

August 20, 2024

安装 Linux VDA 后，您将自动参与数据收集计划。数据收集计划收集统计数据和使用情况数据并将数据发送到 Citrix Analytics，以帮助提高 Citrix 产品的质量和性能。数据上载发生在您启动会话时。

注册表设置

默认情况下，Linux VDA 数据收集计划处于启用状态。可以通过以下注册表设置启用或禁用该计划：

位置：HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

名称：**CEIPSwitch**

值：1 = 已禁用，0 = 已启用（默认值）

如果未指定，则启用 Linux VDA 数据收集计划。

可以运行以下命令来禁用 Linux VDA 数据收集计划：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP" -v "CEIPSwitch" -d "1"
```

从 Linux VDA 收集的数据

下表提供了收集的数据类型的示例。

数据点	注册表项名称	说明
计算机 GUID	machine_guid	用于标识从其获取数据的计算机的 GUID 字符串
Linux 操作系统名称和版本	os_name_version	字符串，表示计算机的 Linux 操作系统名称和版本
AD 解决方案	ad_solution	字符串，表示计算机的域加入方法
Linux 内核版本	kernel_version	字符串，表示计算机的内核版本
GPU 类型	gpu_type	表示 VDA 计算机的 GPU 类型
CPU 内核	cpu_cores	整数，表示计算机的 CPU 内核数
CPU 频率	cpu_frequency	表示 CPU 频率的浮点数（以 MHz 为单位）
物理内存大小	memory_size	整数，表示物理内存大小（以 KB 为单位）

数据点	注册表项名称	说明
LVDA 版本	vda_version	字符串，表示安装的 Linux VDA 版本。
LVDA 更新或全新安装	update_or_fresh_install	字符串，表示正在全新安装或更新当前 Linux VDA 软件包
是否启用 VDI 模式	vdi_mode	字符串，表示是否启用了 VDI 模式
LVDA 安装方法	install_method	字符串，表示通过使用 MCS、PVS、轻松安装或手动安装安装了当前 Linux VDA 软件包。
是否启用 HDX 3D Pro	hdx_3d_pro	字符串，表示是否在计算机上启用了 HDX 3D Pro
是否启用了 BCR	bcr	字符串，表示是否在此计算机上启用了浏览器内容重定向 (BCR)
系统区域设置	system_locale	字符串，表示此计算机的区域设置
场 ID	farm_id	字符串，表示场 ID。
VDA 虚拟化类型	vda_virtualization	字符串，表示创建了 VDA 的虚拟机管理程序
会话密钥	session_key	标识从其获取数据的会话
资源类型	resource_type	字符串，表示启动的会话的资源类型：桌面或 <appname>
Receiver 客户端类型	receiver_type	整数，表示用于启动会话的 Citrix Workspace 应用程序类型
Receiver 客户端版本	receiver_version	字符串，表示用于启动会话的 Citrix Workspace 应用程序版本
打印计数	printing_count	整数，表示会话使用打印功能的次数
USB 重定向计数	usb_redirecting_count	整数，表示会话使用 USB 设备的次数
Gfx 提供程序类型	gfx_provider_type	字符串，表示会话的图形提供程序类型
重影计数	shadow_count	整数，表示会话被重影的次数
用户选择的语言	ctxism_select	包含用户选择的所有语言的组合长字符串
智能卡重定向计数	scard_redirecting_count	整数，表示智能卡重定向用于会话中应用程序的会话登录和用户身份验证的次数
视频编解码器类型	graphic_video_codec_type	字符串，表示哪个视频编解码器用于 Thinwire。

数据点	注册表项名称	说明
登录凭据类型	credentials_type	整数，表示用于登录 Linux VDA 的凭据类型
水印	watermark	字符串，表示是否启用了会话水印
水印透明度	watermark_transparency	整数，表示水印透明度
水印自定义文本长度	watermark_custom_text_len	整数，表示水印自定义文本长度
MTU	mtu	字符串，表示是否在此会话中使用最大传输单位 (MTU)
MTU MSS	mtu_mss	整数，表示最大分段大小 (MSS)
文件传输	filetrans	整数，表示文件传输策略设置
文件传输上传次数	filetrans_upload_count	整数，表示会话中使用“上载”图标进行文件传输的次数
文件传输下载次数	filetrans_download_count	整数，表示会话中使用“下载”图标进行文件传输的次数

HDX Insight

August 20, 2024

概述

Linux VDA 部分支持 [HDX Insight](#) 功能。

安装

没有依赖软件包需要安装。

使用情况

HDX Insight 将分析通过 Citrix Workspace 应用程序和 Linux VDA 之间的 Citrix ADC 传递的 ICA 消息。所有 HDX Insight 数据均来自 NSAP 虚拟通道并以未压缩方式发送。NSAP 虚拟通道默认处于启用状态。

以下命令分别禁用和启用 NSAP 虚拟通道：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\  
VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP" -d "0x00000000"  
--force
```

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\  
VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP" -d "0x00000001"  
--force
```

故障排除

不显示任何数据点

可能有两个原因：

- HDX Insight 未正确配置。
例如，未在 Citrix ADC 上启用 AppFlow，或者在 Citrix ADM 上配置了不正确的 Citrix ADC 实例。
- 在 Linux VDA 上未启动 ICA 控制虚拟通道。

```
ps aux | grep -i ctxctl
```

如果 `ctxctl` 未运行，请与管理员联系以向 Citrix 报告缺陷。

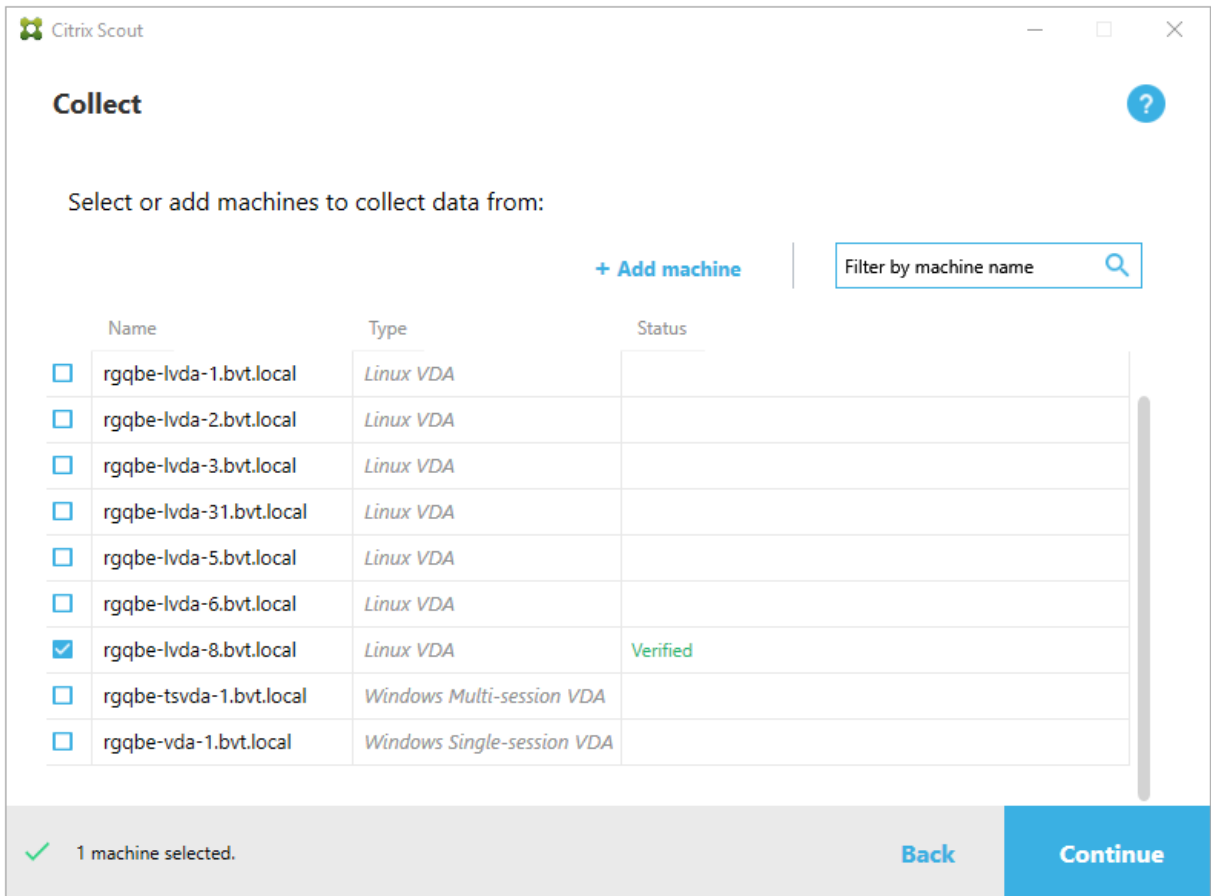
不显示任何应用程序数据点

请确认是否启用了无缝虚拟通道且无缝应用程序正在运行。

与 Citrix Telemetry Service 集成

August 20, 2024

通过与 Linux VDA 软件集成的 Citrix Telemetry Service (`ctxtelemetry`)，您可以运行 Citrix Scout，然后使用 `/opt/Citrix/VDA/bin/xdlcollect.sh` 脚本来收集有关 Linux VDA 的日志。



启用和禁用 Citrix Telemetry Service

- 要启用该服务，请运行 **sudo systemctl enable ctxtelemetry.socket** 命令。
- 要禁用该服务，请运行 **sudo systemctl disable ctxtelemetry.socket**。

端口

默认情况下，Citrix Telemetry Service (`ctxtelemetry`) 使用 TCP/IP 端口 7503 侦听 Citrix Scout。它使用 Delivery Controller 上的 TCP/IP 端口 7502 与 Citrix Scout 进行通信。

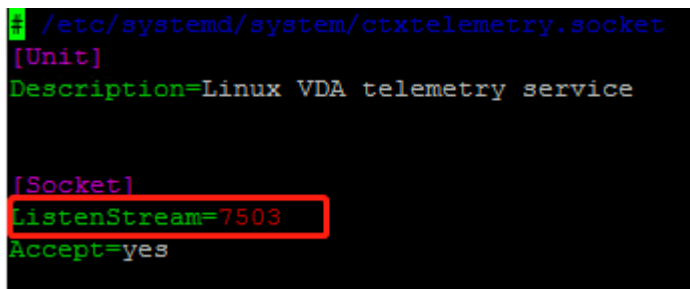
要在安装 VDA 后更改端口，请执行以下操作：

1. 要更改用于与 Scout 通信的端口，请运行以下命令。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t REG_DWORD -v "TelemetryServicePort" -d <port number> --force
```

2. 要更改侦听 Scout 的套接字端口，请运行以下命令以打开并编辑 `ctxtelemetry.socket` 文件。

```
1 sudo vi /etc/systemd/system/ctxtelemetry.socket
```



```
#!/bin/sh
[Unit]
Description=Linux VDA telemetry service

[Socket]
ListenStream=7503
Accept=yes
```

3. 请运行以下命令以重新启动套接字端口。

```
1 sudo systemctl daemon-reload
2 sudo systemctl stop ctxtelemetry.socket
3 sudo systemctl start ctxtelemetry.socket
```

4. 在防火墙配置中启用新端口。

例如，如果您使用的是 Ubuntu，请运行 **sudo ufw allow 7503** 命令以启用端口 7503。

注意：

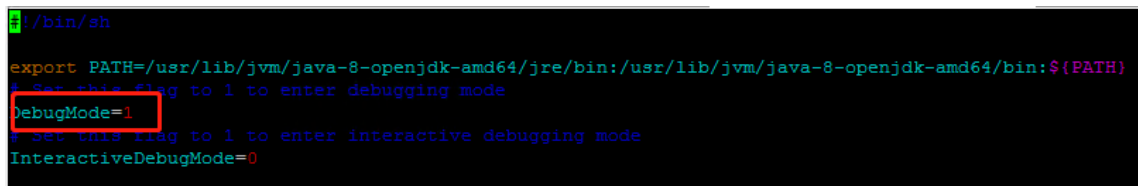
也可以改为运行 `ctxsetup.sh` 来自动执行前面的步骤 3 和 4。

调试模式

如果 Citrix Telemetry Service 无法按预期方式运行，则可以启用调试模式来确定原因。

1. 要启用调试模式，请运行以下命令以打开 `ctxtelemetry` 文件，然后将 `DebugMode` 值更改为 1。

```
1 sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
```



```
#!/bin/sh
export PATH=/usr/lib/jvm/java-8-openjdk-amd64/jre/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin:${PATH}
# Set this flag to 1 to enter debugging mode
DebugMode=1
# Set this flag to 1 to enter interactive debugging mode
InteractiveDebugMode=0
```

2. 手动停止 Citrix Telemetry Service，或者等待 15 分钟以使服务自动停止。


```

administrator@RGQBE-LVDA-3:~$ sudo netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:139             0.0.0.0:*                LISTEN      1447/smbd
tcp        0      0 127.0.0.53:53          0.0.0.0:*                LISTEN      971/systemd-resolve
tcp        0      0 0.0.0.0:22             0.0.0.0:*                LISTEN      1309/sshd
tcp        0      0 127.0.0.1:631          0.0.0.0:*                LISTEN      25158/cupsd
tcp        0      0 127.0.0.1:5432         0.0.0.0:*                LISTEN      998/postgres
tcp        0      0 0.0.0.0:445            0.0.0.0:*                LISTEN      1447/smbd
tcp6       0      0 :::2598                :::*                    LISTEN      28100/ctxhdx
tcp6       0      0 :::139                 :::*                    LISTEN      1447/smbd
tcp6       0      0 :::7502                :::*                    LISTEN      1958/java
tcp6       0      0 :::7303                :::*                    LISTEN      17/init
tcp6       0      0 :::80                  :::*                    LISTEN      1610/java
tcp6       0      0 :::1494                :::*                    LISTEN      28100/ctxhdx
tcp6       0      0 :::22                  :::*                    LISTEN      1309/sshd
tcp6       0      0 :::1:631               :::*                    LISTEN      25158/cupsd
tcp6       0      0 :::445                 :::*                    LISTEN      1447/smbd
administrator@RGQBE-LVDA-3:~$

```

在此示例中，可以运行以下命令来停止 Citrix Telemetry Service。

```

1 sudo netstat -ntlp
2 kill -9 1958

```

3. 要重新启动 Citrix Telemetry Service，请在 Scout 上选择您的 Linux VDA 并在 `/var/log/xdl/` 中查找 `telemetry-debug.log`。

服务等待时间

打开套接字端口的 `systemd` 守护程序默认启动并使用少量资源。Citrix Telemetry Service 默认处于停止状态，并仅在从 Delivery Controller 发出日志收集请求时启动。日志收集完成后，服务会等待新的收集请求，持续时间为 15 分钟，如果没有任何收集请求，则会再次停止。可以通过以下命令配置等待时间。最小值为 10 分钟。如果设置的值小于 10 分钟，则最小值 10 分钟将生效。设置等待时间后，停止并重新启动服务。

```

1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
  VirtualDesktopAgent" -v "TelemetryServiceIdleTimeoutInMinutes" -d <
  number> -t REG_DWORD

```

验证测试

在开始收集信息之前，验证测试将针对选定的每台计算机自动运行。这些测试将确保满足这些要求。如果某台计算机的测试失败，Scout 将显示一条消息，提供建议的更正措施。有关验证测试的详细信息，请参阅 Citrix Scout 文档中的[验证测试](#)部分。

通过 Azure 进行 Linux VDA 自助更新

August 20, 2024

此功能有助于立即或在计划的时间自动更新 Linux VDA 软件。当您在 Citrix DaaS Standard for Azure（以前称为“适用于 Azure 的 Citrix Virtual Apps and Desktops Standard”）中创建 Linux VDA 时非常有用。您在 Azure 中不需要 VM 的任何管理员权限。有关详细信息，请参阅在 [Citrix DaaS Standard for Azure 中创建 Linux VDA](#)。

配置

要使用此功能，请完成以下步骤：

步骤 1: 将更新信息和新 **VDA** 包上载到 **Azure** 容器

步骤 1a: 在 Azure 存储帐户下创建一个容器并将容器访问级别设置为 **Blob (Anonymous read access for blobs only)** (Blob (仅限 blob 的匿名读取访问权限))。

注意：

Azure 容器和 blob 专门由客户持有和管理。Citrix 对其任何安全问题不承担责任。为了确保数据安全性和成本效益，请在每次自助更新后将容器访问级别设置为 **Private (no anonymous access)** (私密 (无匿名访问权限))。

步骤 1b: 将 VDA 更新信息合并到名为 UpdateInfo.json 的 JSON 文件中。有关文件格式的示例，请参阅以下块：

```
1 {
2
3   "Version": "21.04.200.4",
4   "Distributions": [
5     {
6
7     "TargetOS": "RHEL7_9",
8     "PackageName": "",
9     "PackageHash": ""
10    }
11   ,
12   {
13
14   "TargetOS": "UBUNTU20_04",
15   "PackageName": "",
16   "PackageHash": ""
17   }
18
19 ]
20 }
```

其中，“**Version**”表示新的 VDA 版本，“**Distributions**”是一组更新对象。每个对象包含三个项目：

- “**TargetOS**”：必须是“RHEL7_9”（适用于 RHEL 7、CentOS 7 和 Amazon Linux 2）或“UBUNTU20_04”。**ctxmonitord** 无法识别任何其他发行版。
- “**PackageName**”：指定版本的 VDA 软件包的全名。
- “**PackageHash**”：使用 `shasum -a 256 <pkgname>` 命令计算的 SHA-256 值。

步骤 1c: 将 JSON 文件和 Linux VDA 软件包的新版本上载到您的 Azure 容器。

步骤 2: 在主映像或每个 VDA 上启用自助更新功能

默认情况下,自我更新处于禁用状态。如果在 Citrix DaaS Standard for Azure 中创建 Linux VDA,必须在主映像上执行功能启用。否则,请直接在每个目标 VDA 上启用该功能。

要启用自我更新,请运行类似于以下内容的命令,以编辑 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix 下的注册表项。

```

1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_DWORD" -v "fEnabled" -d "0
   x00000001" --force
2
3 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "ScheduledTime" -d "
   Immediately" --force
4
5 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "Url" -d "<Your-Azure-
   Container-Url>" --force
6
7 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
   Control\Citrix\SelfUpdate" -t "REG_SZ" -v "CaCertificate" -d "<Local
   -Certificate-Path-of-PortalAzureCom>" --force

```

下表介绍了注册表设置。

注册表设置	说明
fEnabled	此设置为必填项。默认情况下,该值为 0,表示自我更新已禁用。可以将其设置为 1 以启用自我更新。
Url	此设置为必填项。它设置 Azure 容器的 URL 以获取更新信息和新的 VDA 软件包。
ScheduledTime	此设置为必填项。可以将其设置为 Immediately 或 NextStart 。 Immediately 意味着下载 VDA 软件包后立即运行更新。当下载速度很高且您的更新非常紧急时,此选项适用。但是,如果下载软件包时有任何实时会话,则可能会中断用户体验。 NextStart 意味着在下次启动 ctxmonitord 时运行更新。当下载速度不高且您的更新不紧急时,此选项适用。

注册表设置	说明
CaCertificate	此设置为可选设置。它设置 PEM 证书的完整路径以验证 Azure 容器的 URL。对于 Azure blob，它可以是从浏览器中检索并转换为 PEM 的 portal.azure.com 的证书。为了安全起见，我们建议您添加此注册表设置，但是 Ubuntu 或 Debian 不支持该设置，因为在 Ubuntu 或 Debian 上使用指定的证书文件验证对等方时 <code>curl</code> 命令无法按预期运行。确保设置证书的最低权限。

重新启动 **ctxmonitord** 时，它首先查询 **Url** 以获取 UpdateInfo.json 文件，然后从 JSON 文件中检索更新版本。然后 **ctxmonitord** 会将更新版本与当前版本进行比较。如果当前版本较早，该服务将从 Azure 下载新版本的 VDA 软件包并将其保存在本地。之后，它根据 **ScheduledTime** 的设置运行更新。对于本地部署，您可以直接重新启动 **ctxmonitord** 以触发更新。但是，在 Citrix DaaS Standard for Azure 中，您对 VM 没有管理员权限，只能在重新启动 VDA 计算机后重新启动 **ctxmonitord**。如果更新失败，您的 VDA 将回滚到现有版本。

注意：

- 无法更改在主映像上配置的注册表设置。
- 如果环境中的所有 VM 同时下载软件包，本地网络可能会拥塞。
- 要从提供的 URL 下载，必须在目标 VM 上安装 `curl`。
- 如果更新和回滚均失败，用户数据将丢失。
- 如果更新失败但回滚成功，同一网络中的用户可能拥有不同版本的 Linux VDA。这种情况不是最佳情况。
- 更新通常需要几分钟时间才能完成。Citrix Studio 中没有状态指示器。

Linux VM 和 Linux 会话指标

August 20, 2024

下表列出了适用于 Linux VM 和 Linux 会话的指标。

指标	所需的最低 VDA 版本	说明	备注
登录持续时间	2402	Linux VDA 2109 及更高版本中提供了总登录时间。它是登录过程的一种度量，该过程是指用户从 Citrix Workspace 应用程序开始连接到能够使用会话的这段时间。Linux VDA 2402 及更高版本中提供了登录过程的每个阶段（例如 HDX 连接、身份验证和 GPO ）所花费的时间。要查看登录持续时间指标，请在趋势视图上选择登录性能选项卡。	在 Citrix Director 和 Monitor 中均可用。
ICA 延迟	2311	ICA 延迟是指网络延迟，表示网络是否缓慢运行。打开会话详细信息视图以访问此指标。	在 Citrix Director 和 Monitor 中均可用。
策略	2311	对当前会话有效的所有策略都显示在会话详细信息视图中的策略选项卡上。	在 Citrix Director 和 Monitor 中均可用。
会话自动重新连接计数	2109	要查看会话中的自动重新连接次数，请访问趋势视图。设置条件并单击应用以缩小搜索结果范围。会话自动重新连接计数列将显示会话的自动重新连接次数。会话可靠性或客户端自动重新连接策略生效时，将启用自动重新连接。有关会话重新连接和相关策略的详细信息，请参阅以下文章。 Sessions 客户端自动重新连接策略设置 会话可靠性策略设置	在 Citrix Director 和 Monitor 中均可用。

指标	所需的最低 VDA 版本	说明	备注
空闲时间	2103	要访问此指标，请通过选择过滤器 > 会话 > 所有会话打开所有会话页面。	在 Citrix Director 和 Monitor 中均可用。
Linux VM 的衡量指标	2103	Linux VM 的衡量指标如下：CPU 核心数、内存大小、硬盘容量以及当前和历史 CPU 和内存利用率	在 Citrix Director 和 Monitor 中均可用。
协议	1909	Linux 会话的传输协议将在会话详细信息视图中显示为 UDP 或 TCP。	在 Citrix Director 和 Monitor 中均可用。
ICA RTT	1903	ICA 往返时间 (RTT) 是您按下某个键直到响应出现在终端上所经过的时间。要获取 ICA RTT 指标，请在 Citrix Studio 中创建 ICA 往返行程计算和 ICA 往返行程计算时间间隔策略。	在 Citrix Director 和 Monitor 中均可用。

日志收集

August 20, 2024

日志记录机制概述

下表概述了 Linux VDA 的日志记录机制。

日志记录模块	日志文件名格式	日志文件名示例	日志记录作用域	最大单个文件大小	轮换阈值 (默认值)	配置
HDX	hdx.log[.n]	hdx.log, hdx.log.1	图形、登录、音频、键盘、鼠标	200 MiB	1 个当前文件， 2 个旧文件	可通过 setlog 实用程序进行配置

日志记录模块	日志文件名格式	日志文件名示例	日志记录作用域	最大单个文件大小	轮换阈值 (默认值)	配置
Jproxy	jproxy.log[.n.log]	jproxy.log、 jproxy.log.1.log	VDA 注册、用 户身份验证	20 MiB	1 个当前文件， 10 个旧文件	可通过 setlog 实用 程序或 /etc/xdl/log4j2.xml 进行配置
VDA	vda.YYYY- MM- DD.hh.mm.ss.l	vda.2024- 05- 06.20.18.40.log	VDA 注册	50 MiB	1 个当前文件， 1 个旧文件	可通过 setlog 实用 程序或 /etc/xdl/bro- kera- gent.conf 进 行配置

注意：

- 第一个日志文件的名称中没有数字，后续文件以 “.n” 编号，其中 “n” 表示文件数量。例如，“hdx.log” 是第一个 HDX 日志文件，“hdx.log.1” 是第二个 HDX 日志文件。
- 单个日志文件的最大大小以兆字节 (MiB) 为单位进行测量。
- 正在生成但尚未达到单个文件的最大大小的日志文件称为 “当前” 日志文件。当 “当前” 日志文件达到单个文件的最大大小时，它会滚动并变成 “旧” 日志文件。
- 轮换阈值可配置为限制可以保留的 “旧” 日志文件的数量。达到限制时，最旧的日志文件将被删除。

日志记录配置

本部分内容提供有关日志记录配置的其他信息，对上述表格中概述的详细信息进行了补充。

Linux VDA 的日志记录默认处于启用状态

ctxlogd 守护程序和 setlog 实用程序包括在 Linux VDA 版本软件包中。默认情况下，ctxlogd 守护程序在您安装并配置 Linux VDA 后启动。跟踪的所有其他服务都基于 ctxlogd 守护程序。如果不希望跟踪 Linux VDA，可以停止 ctxlogd 守护程序。

通过 /etc/xdl/brokeragent.conf 配置 VDA 日志记录**注意：**

如果您只想为 VDA 日志记录配置日志级别，而不深入研究其他日志记录参数（例如单个日志文件的最大大小），则

可以使用本文后面介绍的 setlog 实用程序。否则，请使用 /etc/xdl/brokeragent.conf。

VDA 上的 /etc/xdl/brokeragent.conf 文件可用于配置 VDA 日志记录。例如：

```
<!-- VDA trace settings -->
<add key="TraceProvider.Type" value="QueuedLogFileProvider"/>
<add key="TraceProvider.BaseFilename" value="/var/log/xdl/vda.log"/>
<add key="TraceProvider.AppendMode" value="true"/>
<add key="TraceProvider.SizeLimit" value="50"/>
<add key="TraceProvider.TrailCount" value="2"/>
<add key="TraceProvider.ProcessSafe" value="true"/>
<add key="TraceProvider.CategoryFilter" value="All" />
```

通过 /etc/xdl/log4j2.xml 配置 Jproxy 日志记录

注意：

如果您只想为 Jproxy 日志记录配置日志级别，而不深入研究其他日志记录参数（例如单个日志文件的最大大小），则可以使用本文后面介绍的 setlog 实用程序。否则，请使用 /etc/xdl/log4j2.xml。

下面是通过 /etc/xdl/log4j2.xml 配置 Jproxy 日志记录的示例，其中 **SizeBasedTriggeringPolicy** 参数指定单个 Jproxy 日志文件的最大大小，而 **DefaultRollerStrategy** 参数设置可以保留的 Jproxy 日志文件总数。

```
<!-- Logging Properties -->
<Properties>
  <Property name="LOG_PATTERN">%d{yyyy-MM-dd HH:mm:ss.SSS}{GMT} [%-5p] [%tid] - %m%n</Property>
  <Property name="logfile">/var/log/xdl/jproxy.log</Property>
  <Property name="perfflogfile">/var/log/xdl/jproxyperf.log</Property>
</Properties>

<Appenders>

  <Console name="CONSOLE" target="SYSTEM_OUT">
    <PatternLayout pattern="%{LOG_PATTERN}"/>
  </Console>

  <RollingFile name="logfile" fileName="${sys:logfile}"
    filePattern="${sys:logfile}.%i.log">
    <PatternLayout pattern="%{LOG_PATTERN}"/>
    <Policies>
      <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRollerStrategy max="10"/>
  </RollingFile>

  <RollingFile name="perfflogfile" fileName="${sys:perfflogfile}"
    filePattern="${sys:perfflogfile}.%i.log">
    <PatternLayout pattern="%{LOG_PATTERN}"/>
    <Policies>
      <SizeBasedTriggeringPolicy size="19500KB" />
    </Policies>
    <DefaultRollerStrategy max="10"/>
  </RollingFile>
```


通过 **setlog** 实用程序配置 **HDX** 日志记录

setlog 实用程序位于 `/opt/Citrix/VDA/bin/` 路径下。只有 root 用户有权运行该程序。可以使用 GUI 或运行命令来查看和更改配置选项，包括值（日志文件路径、单个文件大小限制和轮换阈值）和日志级别。请运行以下命令以获取使用 setlog 实用程序的帮助信息：

```
1 setlog help
```

值 默认情况下，HDX 日志保存在 `/var/log/xdl/hdx.log` 下，单个 HDX 日志文件的大小限制为 200 MiB，在 `/var/log/xdl/hdx.log` 下最多可以保存两个“旧” HDX 日志文件。

要查看当前的所有 setlog 值，请运行以下命令：

```
1 setlog values
2
3 log_path (Log Output Path) = /var/log/xdl/hdx.log
4
5 log_size (Max Log Size (MiB)) = 200
6
7 log_count (Max Old Log Files) = 2
```

要查看或设置单个 setlog 值，请运行以下命令：

```
1 setlog value <name> [<value>]
```

例如：

```
1 setlog value log_size 100
```

级别 默认情况下，日志级别将设置为 **info**（不区分大小写）。

当您遇到需要进行故障排除的问题时，我们建议您在大多数情况下将日志级别设置为 **verbose**。然后，请重现该问题并收集日志。

要设置日志级别（包括“已禁用”、“已继承”、“跟踪”、“详细”、“信息”、“警告”、“错误”和“致命错误”），请运行以下命令：

```
1 setlog level <class> [<level>]
```

日志级别	命令参数（不区分大小写）
已禁用	none
已继承	inherit
跟踪	trace

日志级别	命令参数（不区分大小写）
详细	verbose
信息	info
警告	warning
错误	error
致命错误	fatal

<class> 变量指定 Linux VDA 的一个组件。要涵盖所有组件，请将其设置为“全部”。例如：

```
1 setlog level all error
```

要查看支持的所有类或组件，请运行以下命令：

```
1 setlog levels
```

还原默认值 请将所有级别和值还原到默认设置：

```
1 setlog default
```

重要：

ctxlogd 服务是使用 `/var/xdl/.ctxlog` 文件配置的，只有 root 用户能够创建该文件。其他用户对该文件没有写入权限。我们建议 root 用户不要向其他用户授予写入权限。否则会导致对 **ctxlogd** 进行任意配置或恶意配置，这样会影响服务器性能，进而影响用户体验。

日志收集

可以运行 **bash /opt/Citrix/VDA/bin/xdlcollect.sh** 命令来收集日志。用于收集日志的 **xdlcollect** Bash 脚本将集成到 Linux VDA 软件中并位于 `/opt/Citrix/VDA/bin` 下。

日志收集完成后，将在 VDA 上的 `/tmp/xdlcollect` 下生成一个 ZIP 文件。

故障排除

`/var/xdl/.ctxlog` 文件丢失（例如，意外删除）时，**ctxlogd** 守护程序失败，您将无法重新启动 **ctxlogd** 服务。

`/var/log/messages`:

```
1 Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
  configuration file.
2
```

```
3 Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited, code
   =exited, status=1/FAILURE
4
5 Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.
6
7 Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
```

要解决此问题，请以 root 用户身份运行 `setlog` 以重新创建 `/var/xdl.ctxlog` 文件。然后重新启动其他服务所基于的 **ctxlogd** 服务。

会话影子处理

August 20, 2024

通过重影会话功能，域管理员可以在 Intranet 中查看用户的 ICA 会话。该功能使用 noVNC 连接到 ICA 会话。

注意：

要使用该功能，请使用 **Citrix Director 7.16** 或更高版本。

安装和配置

依赖项

会话重影需要两个新的依赖关系 `python-websockify` 和 `x11vnc`。在安装 Linux VDA 后手动安装 `python-websockify` 和 `x11vnc`。

对于 **Amazon Linux2**:

请运行以下命令以安装 `python-websockify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本)：

```
1 sudo pip3 install websockify
2 sudo yum install x11vnc
```

对于 **RHEL 9.4/9.3/9.2/8.x** 和 **Rocky Linux 9.4/9.3/9.2/8.x**:

请运行以下命令以安装 `python-websockify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本)。

```
1 sudo pip3 install websockify
2 sudo yum install x11vnc
```

通过启用 EPEL 和 CodeReady Linux Builder 存储库解析 `x11vnc`：

```
1 dnf install -y --nogpgcheck https://dl.fedoraproject.org/pub/epel/epel-
   release-latest-8.noarch.rpm
2
```

```
3 subscription-manager repos --enable "codeready-builder -for-rhel-8-x86_64-rpms"
```

对于 **Ubuntu**:

请运行以下命令以安装 `python-websocketify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本):

```
1 sudo pip3 install websocketify
2 sudo apt-get install x11vnc
```

对于 **SUSE**:

请运行以下命令以安装 `python-websocketify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本):

```
1 sudo pip3 install websocketify
2 sudo zypper install x11vnc
```

对于 **Debian 12.5**:

请运行以下命令以安装 `python-websocketify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本):

```
1 apt install python3-websocketify
2 sudo apt-get install x11vnc
```

对于 **Debian 11**:

请运行以下命令以安装 `python-websocketify` 和 `x11vnc` (`x11vnc` 版本 0.9.13 或更高版本):

```
1 sudo pip3 install websocketify
2 sudo apt-get install x11vnc
```

端口

会话重影功能会自动选择 6001-6099 内的可用端口来建立从 Linux VDA 到 Citrix Director 的连接。因此，可以并行重影的 ICA 会话数不应超过 99 个。请确保有足够的端口可用来满足您的要求，尤其是多会话重影。

注册表

下表列出了相关注册表:

注册表	说明	默认值
EnableSessionShadowing	启用或禁用会话重影功能	1 (启用)
ShadowingUseSSL	确定是否对 Linux VDA 和 Citrix Director 之间的连接加密	0 (禁用)

可在 Linux VDA 上运行 `ctxreg` 命令来更改注册表值。例如，要禁用会话影子处理，请运行以下命令:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\  
VirtualDesktopAgent" -v "EnableSessionShadowing" -d 0x00000000
```

SSL

Linux VDA 和 Citrix Director 之间的 noVNC 连接使用 WebSocket 协议。对于会话重影，将根据前面提及的“ShadowingUseSSL”注册表来选择 `ws://` 或 `wss://`。默认情况下，选择 `ws://`。但是，出于安全原因，我们建议您使用 `wss://`，并在每个 Citrix Director 客户端和每台 Linux VDA 服务器上安装证书。Citrix 拒绝承担使用 `ws://` 进行 Linux VDA 会话重影的任何安全责任。

获取服务器证书和根 **SSL** 证书 证书必须由可信证书颁发机构 (CA) 签发。

对于要在其中配置 SSL 的每台 Linux VDA 服务器来说，都需要一份单独的服务器证书（包括密钥）。服务器证书标识特定的计算机，因此您必须知道每台服务器的完全限定的域名 (FQDN)。您可以对整个域使用通配符证书。在这种情况下，您必须至少知道域名。

与 Linux VDA 通信的每个 Citrix Director 客户端也需要根证书。根证书可从颁发服务器证书的另一 CA 获得。

可以从以下 CA 安装服务器和客户端证书：

- 与您的操作系统捆绑在一起的 CA
- 企业 CA（贵组织允许您访问的 CA）
- 未与操作系统捆绑在一起的 CA

请咨询您组织的安全团队，了解他们需要哪种方法来获取证书。

重要：

- 服务器证书的公用名必须是 Linux VDA 的确切 FQDN 或至少是正确的通配符加域字符。例如 `vda1.basedomain.com` 或 `*.basedomain.com`。
- 哈希算法（包括 SHA1 和 MD5）对于数字证书中的签名而言太弱，某些浏览器不支持。因此，SHA-256 指定为最低标准。

在每个 **Citrix Director** 客户端上安装根证书 会话重影与 IIS 使用相同的基于注册表的证书存储，因此您可以使用 IIS 或 Microsoft 管理控制台 (MMC) 证书管理单元安装根证书。收到 CA 的证书后，可在 IIS 中重新启动 Web 服务器证书向导，然后此向导将安装该证书。另外，您可以在使用 MMC 并将证书作为独立的管理单元添加的计算机上，查看并导入证书。默认情况下，Internet Explorer 和 Google Chrome 会导入安装在操作系统上的证书。对于 Mozilla Firefox，必须在证书管理器的 **Authorities**（颁发机构）选项卡上导入您的根 CA 证书。

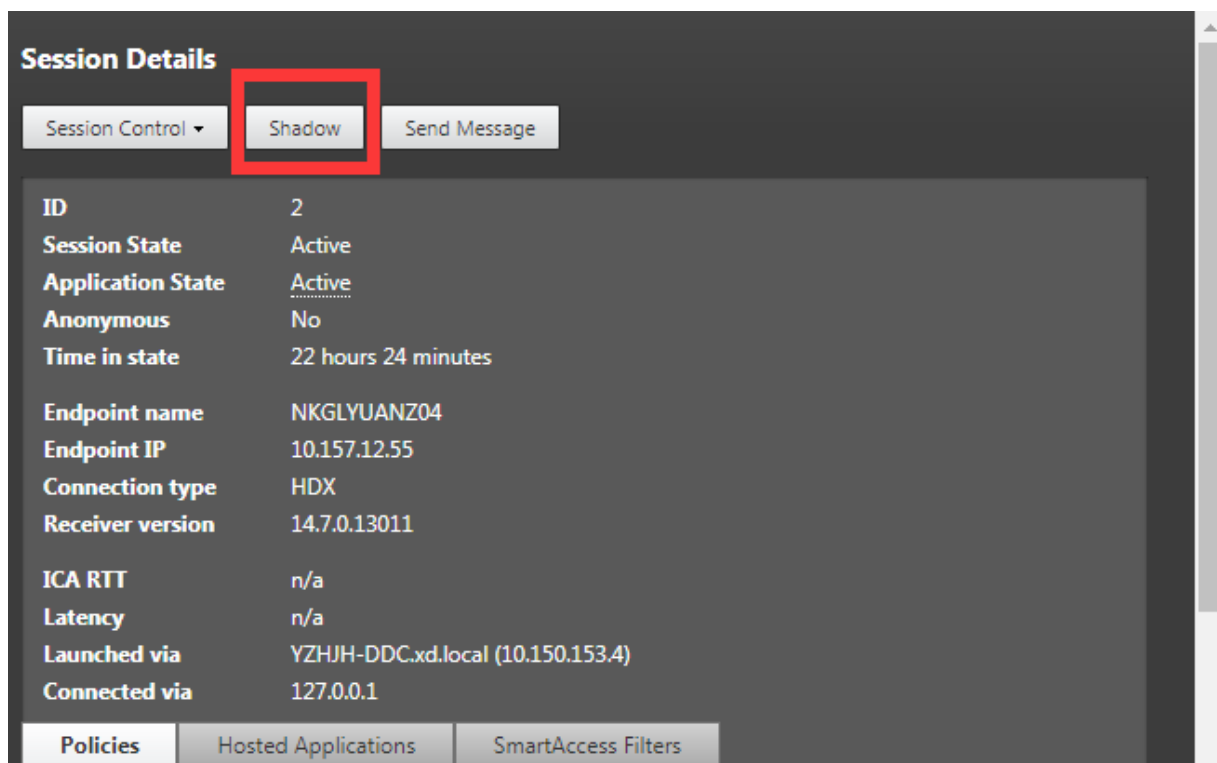
在每台 **Linux VDA** 服务器上安装服务器证书及其密钥 将服务器证书命名为“`shadowingcert.*`”，并将密钥文件命名为“`shadowingkey.*`”（* 指示格式，例如 `shadowingcert.pem` 和 `shadowingkey.key`）。将服务器证书和密钥文件放在路径 `/etc/xdm/shadowingssl` 下，并使用受限权限适当地对其进行保护，仅允许 `ctxsrvr` 具有读取访问

权限。如果名称或路径不正确，Linux VDA 将无法找到特定的证书或密钥文件，从而导致与 Citrix Director 的连接失败。命令如下所示：

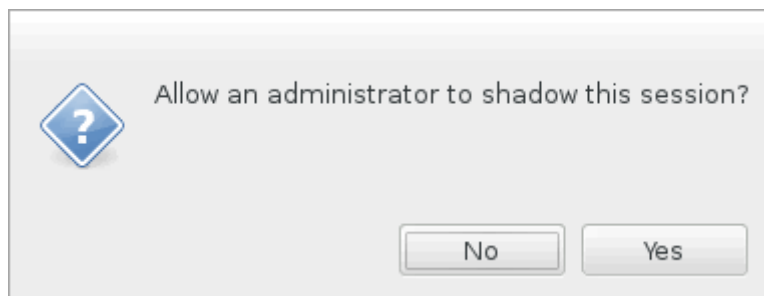
```
1 cp <vda's-public-key> /etc/xdl/shadowingssl/shadowingcert.pem
2 cp <vda's-server-private-key> /etc/xdl/shadowingssl/shadowingkey.key
3 sudo chown ctxsrvr:ctxadm /etc/xdl/shadowingssl/shadowingcert.pem
4 sudo chown ctxsrvr:ctxadm /etc/xdl/shadowingssl/shadowingkey.key
```

使用情况

在 Citrix Director 中，找到目标会话，然后单击会话详细信息视图中的重影以向 Linux VDA 发送重影请求。



连接初始化后，ICA 会话客户端（不是 Citrix Director 客户端）上会显示一条确认消息，要求用户允许对会话进行重影。



如果用户单击是，则 Citrix Director 端将显示一个窗口，指示正在重影 ICA 会话。

有关用法的详细信息，请参阅 [Citrix Director 文档](#)。

限制

- 会话重影仅适用于 Intranet。它不适用于外部网络，即使通过 Citrix Gateway 连接也是如此。Citrix 拒绝承担在外部网络中进行 Linux VDA 会话重影的任何责任。
- 启用了会话重影后，域管理员只能查看 ICA 会话，但无权对其写入或控制。
- 管理员在 Citrix Director 中单击重影后，系统会显示一条确认消息，要求用户允许对会话进行重影。只有在会话用户允许后才能对会话进行重影。
- 前面提及的确认消息有超时限制，即 20 秒。超时后，重影请求将失败。
- 一个会话只能由一位管理员进行重影。例如，如果管理员 B 向会话管理员 A 发送重影请求，获取用户权限的确认信息会重新出现在用户设备上。如果用户同意，管理员 A 的重影连接将停止，并为管理员 B 构建新的重影连接。如果管理员为同一会话发送另一个重影请求，还可以构建新的重影连接。
- 要使用会话重影功能，请安装 Citrix Director 7.16 或更高版本。
- Citrix Director 客户端使用 FQDN 而不是 IP 地址来连接到目标 Linux VDA 服务器。因此，Citrix Director 客户端必须能够解析 Linux VDA 服务器的 FQDN。

故障排除

如果会话重影失败，请在 Citrix Director 客户端和 Linux VDA 上进行调试。

在 Citrix Director 客户端上

通过浏览器的开发人员工具，在控制台选项卡上检查输出日志。或者在网络选项卡上检查 ShadowLinuxSession API 的响应。如果显示获取用户权限的确认信息，但连接建立失败，请手动 ping VDA 的 FQDN 以验证 Citrix Director 是否能够解析 FQDN。如果 `wss://` 连接出现问题，请检查您的证书。

在 Linux VDA 上

确认在出现重影请求时，是否显示要求用户允许的确认消息。如果没有，请检查 `vda.log` 和 `hdx.log` 文件以获取相关线索。要获取 `vda.log` 文件，请执行以下操作：

1. 找到 `/etc/xdl/ctx-vda.conf` 文件。取消注释以下行以启用 `vda.log` 配置：

```
Log4jConfig="/etc/xdl/log4j.xml"
```

2. 打开 `/etc/xdl/log4j.xml`，找到 `com.citrix.dmc` 部分，并将“info”更改为“trace”，如下所示：

```
1 <!-- Broker Agent Plugin - Director VDA plugin Logger -->
2
3 <logger name="com.citrix.dmc">
4
```

```
5     <level value="trace"/>
6
7     </logger>
```

3. 运行 `systemctl restart ctxvda` 命令以重新启动 `ctxvda` 服务。

如果在建立连接期间出现错误：

1. 检查是否有任何防火墙限制阻止会话重影打开端口。
2. 在使用 SSL 的情况下，确认您已正确命名证书和密钥文件，并将其置于正确的路径下。
3. 确认 6001-6099 之间是否有足够的端口用于新的重影请求。

监视服务守护程序

August 20, 2024

监视服务 **ctxmonitord** 是一个守护程序，用于监视 Linux VDA 的运行状况。它包含三个模块：

- 主要进程监视模块：此模块委派 Systemd 在出现意外故障时自动重新启动关键服务，定期扫描和记录目标进程的状态，并确保及时清理 Xorg 残留项。日志存储在 `/var/log/xdl/ms.log` 中。
- XDPing 集成模块：此模块定期执行 xDPing 分析和备份任务。这些任务的所有输出都可以在 `/var/log/xdl/m-sxdping.log` 中找到。
- Linux VDA 自我更新模块：有关详细信息，请参阅[通过 Azure 进行的 Linux VDA 自我更新](#)。

配置

默认情况下，当您启动 VDA 时，监视服务 **ctxmonitord** 会自动启动。使用管理员权限，您可以通过 `/etc/xdl/ctxmonitord.conf` 和 `/etc/xdl/whitelist.conf` 对其进行配置。

ctxmonitord.conf

此配置文件指定了监视服务 `ctxmonitord` 及其模块的行为。默认情况下，它配置如下：

```
1     ; This is the configuration file for ctxmonitord service
2
3
4     ; Section 'service' configures the key process monitor module
5     [service]
6     MonitorEnable=true           ; true or false
7     DetectInterval=300           ; in seconds, minimum is 60
8
9     ; Section 'xdping' configures the XDPing integration module
10    [xdping]
```



```
11 XdpingEnable=true      ; true or false
12 AnalysisInterval=600   ; in seconds, minimum is 60, 0 means disable
13 BackupInterval=1       ; in days, 0 means disable
14
15 ; Section 'rules' configures the rules about how to monitor each key
   process
16 ; Each rule should be named as rules.<ProcessName> where <ProcessName>
   is the name of the process to be monitored
17 ; 'ProcessName' should be the name of the target process that is picked
   up from whitelist.conf
18 ; 'MonitorType' should be 'HealthCheck' or 'ResidueCleanup'. All '
   MonitorType' should be 'HealthCheck' except for 'Xorg'
19 [rules]
20 [rules.ctxhdx]
21 ProcessName=ctxhdx
22 MonitorType=HealthCheck
23
24 [rules.ctxvda]
25 ProcessName=ctxvda
26 MonitorType=HealthCheck
27
28 [rules.ctxjproxy]
29 ProcessName=ctxjproxy
30 MonitorType=HealthCheck
31
32 [rules.ctxpolicyd]
33 ProcessName=ctxjproxy
34 MonitorType=HealthCheck
35
36 [rules.ctxlogd]
37 ProcessName=ctxlogd
38 MonitorType=HealthCheck
39
40 [rules.xorg]
41 ProcessName=Xorg
42 MonitorType=ResidueCleanup
```

whitelist.conf

此配置文件指定要监视的目标进程的白名单。whitelist.conf 文件中还必须列出 ctxmonitord.conf 文件中指定的目标服务。默认情况下，它配置如下：

```
1 ctxhdx
2 ctxvda
3 ctxjproxy
4 ctxlogd
5 ctxcdm
6 ctxcups
7 ctxpolicyd
8 ctxusbsd
9 ctxceip
```

```
10 ctxsdcd
11 ctxrunatboot
12 ctxgdtd
13 ctxbcrd
14 ctxfidod
15 ctxpfdw
16 ctxwcamsd
17 ctxcertmgr
18 ctxscardsd # Except for SLES
19 ctxfedsd   # Only for RHEL 8/9
20 Xorg
```

注意：

- 要使您对 `ctxmonitord.conf` 和 `whitelist.conf` 文件所做的更改生效，请运行 **systemctl restart ctxmonitord** 命令来重新启动监视服务守护程序。
- 要配置服务在出现意外故障时如何重新启动，请编辑 `/usr/lib/systemd/system/ctx*.service` 文件。例如，下面是默认的重新启动选项：

```
1 Restart=on-failure
2 RestartSec=5
3 StartLimitInterval=60
4 StartLimitBurst=3
```

故障排除

August 20, 2024

本文介绍了如何使用 **XDPing** 进行故障排除以及如何使用 **ctxsdcutil** 实用程序查询会话衡量指标。

XDPing

Linux **XDPing** 工具是一个命令行应用程序。它可以自动执行检查 Linux VDA 环境中的常见配置问题的过程。

安装 Linux XDPing 工具

运行 `ctxsetup.sh` 不会安装 **XDPing**。要安装 **XDPing**，请运行 `sudo /opt/Citrix/VDA/bin/xdping`。此命令还会创建一个 **XDPing** 所需的 **Python3** 虚拟环境。如果此命令无法创建 **Python3** 虚拟环境，请按照[创建 Python3 虚拟环境](#)中的说明手动创建该环境。

要解决在使用 `pip` 工具时可能会遇到的 SSL 连接错误，请考虑将以下可信主机添加到 `/etc/pip.conf` 文件中：

```
[global]
trusted-host =
pypi.org
files.pythonhosted.org
```

可以使用 **XDPing** 完成的任务

XDPing 附带从命令 shell 运行的名为 **XDPing** 的单个可执行文件。

下表介绍了可以使用相应的 **XDPing** 命令完成的各种任务：

任务	XDPing 命令	备注
显示命令行选项	sudo /opt/Citrix/VDA/bin/xdping -h	不适用
运行全套测试	sudo /opt/Citrix/VDA/bin/xdping (在没有任何命令行选项的情况下运行 XDPing)	Linux XDPing 工具在系统中执行了 150 多项单项测试。有关详细信息，请参阅本文后面的单项测试。
运行 VDA 注册状态检查	sudo /opt/Citrix/VDA/bin/xdping -a	有关详细信息，请参阅本文后面的注册状态检查的范围。
备份 VDA 的关键数据	sudo /opt/Citrix/VDA/bin/xdping -b	有关详细信息，请参阅本文后面的 VDA 数据的备份和比较。
比较 VDA 备份数据的两个最新副本	sudo /opt/Citrix/VDA/bin/xdping -diff	有关详细信息，请参阅本文后面的 VDA 数据的备份和比较。
比较 VDA 备份数据的两个特定副本	**sudo /opt/Citrix/VDA/bin/xdping -diff=< 一个备份数据目录 >:< 另一个备份数据目录 >** 另一个备份数据目录 > 一个备份数据目录 > 另一个备份数据目录 > 一个备份数据目录 >	有关详细信息，请参阅本文后面的 VDA 数据的备份和比较。
在安装 Linux VDA 软件包之前检查环境	sudo /opt/Citrix/VDA/bin/xdping -preflight	不适用
仅运行特定的测试类别（例如，time、Kerberos 和 database 测试）	sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos,database	不适用

任务	XDPing 命令	备注
探测特定的 Delivery Controller	**sudo /opt/Citrix/VDA/bin/xdping -d **	不适用
显示 VDA 上安装了的依赖项	sudo /opt/Citrix/VDA/bin/xdping - query-pkgs 或 sudo /opt/Citrix/VDA/bin/xdping -q	不适用
显示 VDA 上安装的依赖项并将查询结果保存在特定路径中	**sudo /opt/Citrix/VDA/bin/xdping -f < 用于保存查询结果的特定路径 > * * 或 * *sudo /opt/Citrix/VDA/bin/xdping --query-pkgs-to < 用于保存查询结果 的特定路径 > * * 用于保存查询结果 的特定路径 > 用于保存查询结果的 特定路径 > 用于保存查询结果的特定 路径 > 用于保存查询结果的特定路径 >	不适用

单项测试 Linux **XDPing** 工具在系统中执行 150 多个单项测试，这些测试大致分类如下：

- 检查是否满足 Linux VDA 系统要求。
- 识别和显示计算机信息，包括 Linux 发行版。
- 检查 Linux 内核的兼容性。
- 检查是否存在任何可能影响 Linux VDA 操作的已知 Linux 发行版问题。
- 检查安全增强型 Linux (SELinux) 模式和兼容性。
- 识别网络接口并检查网络设置。
- 检查存储分区和可用的磁盘空间。
- 检查计算机主机和域名配置。
- 检查 DNS 配置并执行查找测试。
- 识别基本虚拟机管理程序并检查虚拟机配置。支持：
 - XenServer (以前称为 Citrix Hypervisor)
 - Microsoft HyperV
 - VMware vSphere
- 检查时间设置并检查网络时间同步是否可以运行。
- 检查 PostgreSQL 服务是否正确配置和运行。

- 检查 SQLite 的配置和运行是否正确。
- 检查防火墙是否已启用且所需端口是否已打开。
- 检查 Kerberos 配置并执行身份验证测试。
- 检查 LDAP 搜索环境中的组策略服务引擎。
- 检查 Active Directory 集成是否已正确设置，以及当前计算机是否已加入域。支持：
 - Samba Winbind
 - Dell Quest Authentication Services
 - Centrify DirectControl
 - SSSD
- 检查 Active Directory 中 Linux 计算机对象的完整性。
- 检查可插拔身份验证模块 (PAM) 配置。
- 检查核心转储模式。
- 检查是否安装了 Linux VDA 所需的软件包。
- 识别 Linux VDA 软件包并检查安装的完整性。
- 检查 PostgreSQL 注册表数据库的完整性。
- 检查 Linux VDA 服务是否已正确配置和运行。
- 检查 VDA 和 HDX 配置的完整性。
- 探测配置的每个 Delivery Controller，以测试 Broker Service 是否可访问、运行且响应迅速。
- 检查计算机是否已在 Delivery Controller 场中注册。
- 检查每个活动或断开连接的 HDX 会话的状态。
- 扫描日志文件中是否存在与 Linux VDA 相关的错误和警告。
- 检查 Xorg 的版本是否适合。
- 检查是否安装了所需的依赖项。

示例输出 下面是运行 Kerberos 测试的示例输出：

```
sudo xdping -T kerberos

Root User -----
User:          root
EUID:          0
Verify user is root                                [Pass]

Kerberos -----
Kerberos version: 5
Verify Kerberos available                          [Pass]
Verify Kerberos version 5                          [Pass]
KRB5CCNAME:    [Not set]
               Distro default FILE:/tmp/krb5cc_%.uid}
KRB5CCNAME type: [Supported]
KRB5CCNAME format: [Default]
Verify KRB5CCNAME cache type                       [Pass]
Verify KRB5CCNAME format                           [Pass]
Configuration file: /etc/krb5.conf [Exists]
```

```

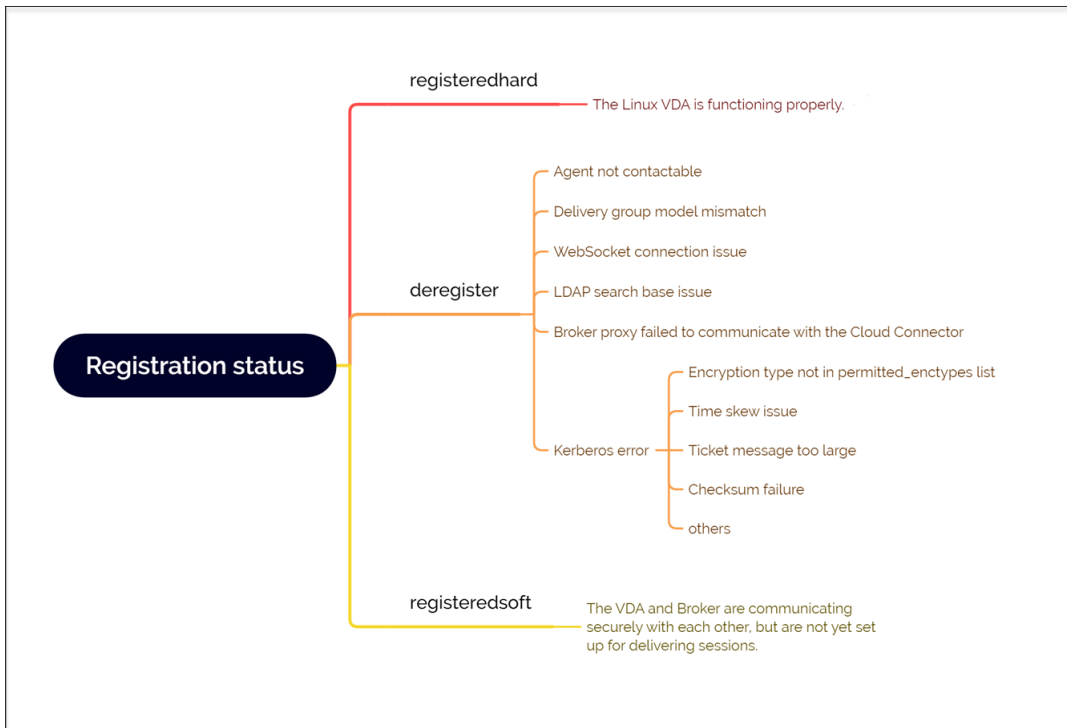
    Verify Kerberos configuration file found [Pass]
Keytab file: /etc/krb5.keytab [Exists]
Default realm: XD2.LOCAL
Default realm KDCs: [NONE SPECIFIED]
Default realm domains: [NONE SPECIFIED]
DNS lookup realm: [Enabled]
DNS lookup KDC: [Enabled]
Weak crypto: [Disabled]
Clock skew limit: 300 s
    Verify system keytab file exists [Pass]
    Verify default realm set [Pass]
    Verify default realm in upper-case [Pass]
    Verify default realm not EXAMPLE.COM [Pass]
    Verify default realm domain mappings [Pass]
    Verify default realm master KDC configured [Pass]
    Verify Kerberos weak crypto disabled [Pass]
    Verify Kerberos clock skew setting [Pass]
Default ccache: [Not set]
    Distro default FILE:/tmp/krb5cc_%{uid}
Default ccache type: [Supported]
Default ccache format: [Default]
    Verify default credential cache cache type [Pass]
    Verify default credential cache format [Pass]
UPN system key [MYVDA1$@██████████]: [MISSING]
SPN system key [host/██████████@██████████]: [Exists]
    Verify Kerberos system keys for UPN exist [ERROR]
    No system keys were found for the user principal name (UPN) of
    the machine account. For the Linux VDA to mutually authenticate
    with the Delivery Controller, the system keytab file must
    contain keys for both the UPN and host-based SPN of the machine
    account.

    Verify Kerberos system keys for SPN exist [Pass]
Kerberos login: [FAILED AUTHENTICATION]
    Keytab contains no suitable keys for MYVDA1$@██████████
    while getting initial credentials
    Verify KDC authentication [ERROR]
    Failed to authenticate and obtain a Ticket Granting Ticket (TGT)
    from the KDC authentication service for the machine account UPN
    MYVDA1$@██████████. Check that the Kerberos configuration is
    valid and the keys in the system keytab are current.

Summary -----
The following tests did not pass:
    Verify Kerberos system keys for UPN exist [ERROR]
    Verify KDC authentication [ERROR]

```

VDA 注册状态检查的范围 Linux **XDPing** 工具还提供了一个分析模块，可帮助您检查和分析 VDA 注册状态。有关注册状态检查的范围，请参见以下屏幕截图：



VDA 数据的备份和比较 自 Linux VDA 2305 起，**XDPing** 工具引入了 VDA 备份模块。此模块允许您随时备份 VDA 的关键数据，例如配置、数据库和二进制权限数据。当 VDA 正常运行时，您可以备份 VDA 的关键数据。如果 VDA 以后出现故障，请备份另一个数据副本并比较两个数据副本以方便故障排除。下表介绍了 VDA 数据备份以及与相应的 **XDPing** 命令的比较：

任务	XDPing 命令	备注
备份 VDA 的关键数据	sudo /opt/Citrix/VDA/bin/xdping -b	每次运行备份命令时，都会生成备份数据的副本并将其保存在 /var/ctxbackup 下的目录中。备份数据目录以 yyyy-mm-dd-hh_mm_ss 格式的当前日期和时间命名，例如 2023-02-27-16_31_27 。默认情况下，备份数据目录的最大数量为 30，超过该数量时 XDPing 工具会轮换或删除旧的备份数据目录。要自定义目录轮换的次数，请运行以下命令： <pre>sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\Backup"-t "REG_DWORD"-v "MaxDirRotationCount"-d "0x00000005"--force</pre>
比较 VDA 备份数据的两个最新副本	sudo /opt/Citrix/VDA/bin/xdping -diff	不适用
比较 VDA 备份数据的两个特定副本	**sudo /opt/Citrix/VDA/bin/xdping -diff=< 一个备份数据目录 >:< 另一个备份数据目录 > * * 另一个备份数据目录 > 一个备份数据目录 > 另一个备份数据目录 > 一个备份数据目录 >	不适用

会话指标查询实用程序

ctxsession

此实用程序提供 Windows 用户体验。使用此实用程序，您可以按以下屏幕截图所示访问会话指标：


```
root@u22:~# /opt/Citrix/VDA/bin/ctxsession -v

      Session Id:      4
      Protocols:      TCP-TLS-CGP-ICA
      User Name:      st\wcx
      Client Name:    BLR3-EWS-WPD030
      Client Address: 10.109.143.143:58535
      Local Address:  10.158.180.61:443
      Remote Address: 10.109.143.143:58535
      Security Protocol: TLSv1.2
      Security Cipher: ECDHE-RSA-AES256-GCM-SHA384
      Cipher Strength: 256 bits
      ICA Encryption: Basic
      Rendezvous Version: None
      Reducer Version: 4

      ICA Statistics:
      ICA RTT: 0 ms
      Sent Bandwidth: 112 bps
      Received Bandwidth: 0 bps
```

ctxqsession、ctxquser、ctxqfull、ctxquery

有关如何使用这些实用程序的说明，请运行帮助命令，例如：

```
root@ubuntu2204-33:~# /opt/Citrix/VDA/bin/ctxqsession --help
Usage: ctxqsession [-f short_format_options | -o long_format_options] [-m] [user user_name]
Display information about ICA connections to the local server.

Options:
  -f          Change the displayed information specified by short format options
  -o          Change the displayed information specified by long format options
  -m          Alter the column headers to remove spaces
  user       Display information about the specified user

Short format options:
  d          Client device name (client name)
  i          Session ID
  n          Session name
  p          Published application name
  P          Protocols
  S          Session state
  u          User name
  x          X display number

Long format options(comma separated):
  app       Published application name
  dev       Client device name (client name)
  id        Session ID
  proto     Protocols
  sess      Session name
  state     Session state
  user      User name
  xdp       X display number
```

用户只能查询自己的会话指标。只有 root 用户和 **ctxadm** 组用户有权查询其他用户的会话指标。

ctxsdcutil

此实用程序有助于查询 VDA 上托管的所有会话或特定会话的以下指标。为此，请运行 `/opt/Citrix/VDA/bin/ctxsdcutil -q <all | SessionID> [-c]` 命令。[-c] 参数意味着每秒查询一次指标。

- 输入会话带宽
- 输出会话带宽、
- 输出会话线速度
- 延迟 - 上次记录
- 往返行程时间
- 输出 **ThinWire** 带宽
- 输出音频带宽
- 输出打印机带宽
- 输入驱动器带宽
- 输出驱动器带宽

其他

July 15, 2024

本部分内容包含以下主题：

- [适用于 HTML5 的 Citrix Workspace 应用程序支持](#)
- [创建 Python3 虚拟环境](#)
- [将 NIS 与 Active Directory 集成](#)
- [IPv6](#)
- [LDAPS](#)
- [Xauthority](#)

适用于 **HTML5** 的 **Citrix Workspace** 应用程序支持

July 15, 2024

可以使用适用于 HTML5 的 Citrix Workspace 应用程序来直接访问 Linux 虚拟应用程序和桌面，而无需将客户端连接到 Citrix Gateway。有关适用于 HTML5 的 Citrix Workspace 应用程序的信息，请参阅 [Citrix 文档](#)。

启用此功能

默认情况下，此功能处于禁用状态。要启用该功能，请执行以下操作：

1. 在 Citrix StoreFront 中，启用适用于 HTML5 的 Citrix Workspace 应用程序。

有关详细过程，请参阅知识中心文章 [CTX208163](#) 的“步骤 1”。

2. 启用 WebSocket 连接。

- a) 在 Citrix Studio 中，将 **WebSocket** 连接策略设置为允许。

您还可以设置其他 WebSocket 策略。有关 WebSocket 策略的完整列表，请参阅 [WebSockets 策略设置](#)。

- b) 在 VDA 上，请按此顺序重新启动 `ctxvda` 服务和 `ctxhdx` 服务，以使设置生效。

- c) 在 VDA 上，运行以下命令来检查 WebSocket 侦听器是否正在运行。

```
netstat -an | grep 8008
```

WebSocket 侦听器处于运行状态时，命令输出将如下所示：

```
tcp 0 0 :::8008 :::* LISTEN
```

注意：您还可以启用 TLS 加密以保护 WebSocket 连接的安全。有关启用 TLS 加密的信息，请参阅[使用 TLS 保护用户会话安全](#)。

创建 Python3 虚拟环境

August 20, 2024

如果要连接到网络，运行 `sudo /opt/Citrix/VDA/bin/xdping` 或 `/opt/Citrix/VDA/sbin/enable_ldaps.sh` 命令可以创建 Python3 虚拟环境。但是，如果这些命令无法创建 Python3 虚拟环境，则即使没有网络连接，也可以手动创建。本文详细介绍了创建没有网络连接的 Python3 虚拟环境的必备条件和步骤。

必备条件

- 必须具有管理权限才能访问 `/opt/Citrix/VDA/sbin/ctxpython3` 目录。
- Python3 软件包的滚轮文件已准备就绪。可以从 <https://pypi.org/> 下载滚轮文件。

创建 Python3 虚拟环境

请完成以下步骤以创建 Python3 虚拟环境：

1. 安装 Python3 依赖项。

对于 **Amazon Linux 2**：

```
1 yum -y install python3 python3-devel krb5-devel gcc
```

对于 **RHEL** 和 **Rock Linux**：

```
1 yum -y install python3-devel krb5-devel gcc
```

注意：

您可能必须启用特定存储库才能安装某些依赖项。对于 RHEL 7，请运行 `subscription-manager repos --enable rhel-7-server-optional-rpms` 命令。对于 RHEL 8，请运行 `subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms` 命令。

对于 **Debian** 来说，**Ubuntu**：

```
1 apt-get -y install python3-dev python3-pip python3-venv libkrb5-dev
```

对于 **SUSE**:

```
1 zypper -n install lsb-release python3-devel python3-setuptools
   krb5-devel gcc libffi-devel libopenssl-devel
```

2. 创建 Python3 虚拟环境。

注意:

要解决在使用 pip 工具时可能会遇到的 SSL 连接错误, 请考虑将以下可信主机添加到 /etc/pip.conf 文件中:

```
[global]
trusted-host =
pypi.org
files.pythonhosted.org
```

对于 **Amazon Linux 2、Debian、RHEL、Rocky Linux、Ubuntu**:

```
1 sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

对于 **SUSE**:

```
1 sudo ln -s /usr/lib/mit/bin/krb5-config /usr/bin/krb5-config
2
3 export PATH=$PATH:/usr/lib/mit/bin:/usr/lib/mit/sbin
4
5 sudo mkdir -p /usr/lib/mit/include/gssapi/
6
7 sudo ln -s /usr/include/gssapi/gssapi_ext.h/usr/lib/mit/include/
   gssapi/gssapi_ext.h
8
9 sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
```

3. 安装 LDAPS 依赖项。

```
1 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
   upgrade pip==21.3.1
2
3 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
   cffi==1.15.0 cryptography==36.0.2 decorator==5.1.1 gssapi
   ==1.7.3 ldap3==2.9.1 pyasn1==0.4.8 pycparser==2.21 six==1.16.0
```

4. 安装 XDPing 依赖项。

```
1 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install --
   upgrade pip==21.3.1
2
3 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
   asn1crypto==1.5.1 cffi==1.15.0 cryptography==36.0.2 decorator
   ==5.1.1 gssapi==1.7.3 ldap3==2.9.1 netifaces==0.11.0 packaging
   ==21.3 pg8000==1.26.0 psutil==5.9.0 pyasn1==0.4.8 pycparser
```

```
==2.21 pyparsing==3.0.8 scramp==1.4.1 six==1.16.0 termcolor
==1.1.0
4
5 sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /
opt/Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
```

将 NIS 与 Active Directory 集成

July 15, 2024

本文介绍如何在 Linux VDA 中使用 SSSD 将 NIS 与 Windows Active Directory (AD) 集成。Linux VDA 被视为 Citrix Virtual Apps and Desktops 的一个组件。因此，它与 Windows AD 环境紧密结合。

使用 NIS（而非 AD）作为 UID 和 GID 提供程序要求帐户信息（用户名和密码组合）在 AD 和 NIS 中相同。

注意：

仍由 AD 服务器执行身份验证。不支持 NIS+。如果使用 NIS 作为 UID 和 GID 提供程序，则不再使用来自 Windows 服务器的 POSIX 属性。

提示：

此方法代表已弃用的 Linux VDA 部署方法，这种方法仅用于特殊用例。对于 RHEL/CentOS 发行版，请按照在 [Amazon Linux 2、CentOS、RHEL 和 Rocky Linux 上手动安装 Linux VDA](#) 中的说明进行操作。对于 Ubuntu 发行版，请按照在 [Ubuntu 上手动安装 Linux VDA](#) 中的说明进行操作。

SSSD 是什么？

SSSD 是系统守护程序。其主要功能是为了实现通过可以为系统提供缓存和脱机支持的通用框架来识别远程资源并对其进行身份验证。它提供 PAM 和 NSS 两种模块，将来可以为扩展用户信息支持基于 D-BUS 的接口。此外它还提供更好的数据库来存储本地用户帐户和扩展用户数据。

将 NIS 与 AD 相集成

要将 NIS 与 AD 集成，请完成以下步骤：

步骤 1： 将 **Linux VDA** 添加为 **NIS** 客户端

配置 NIS 客户端：

```
1 yum -y install ybind rpcbind oddjob-mkhomedir
```

设置 NIS 域：

```
1 ypdomainname nis.domain
2 echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
```

在 `/etc/hosts` 中添加 NIS 服务器和客户端的 IP 地址：

```
{ NIS server IP address }    server.nis.domain nis.domain
```

通过 `authconfig` 配置 NIS：

```
1 sudo authconfig --enablenis --nisdomain=nis.domain --nisserver=server.
  nis.domain --enablemkhomedir --update
```

`nis.domain` 表示 NIS 服务器的域名。`server.nis.domain` 是 NIS 服务器的主机名，也可以是 NIS 服务器的 IP 地址。

配置 NIS 服务：

```
1 sudo systemctl start rpcbind ypbind
2
3 sudo systemctl enable rpcbind ypbind
```

确保 NIS 配置正确：

```
1 ypwhich
```

验证可从 NIS 服务器获得帐户信息：

```
1 getent passwd nisaccount
```

注意：

`nisaccount` 表示 NIS 服务器上的实际 NIS 帐户。确保 UID、GID、主目录和登录 shell 都已正确配置。

步骤 2：使用 Samba 加入域并创建主机 keytab

SSSD 并不提供用于加入域和管理系统 keytab 文件的 AD 客户端功能。实现这些功能可以采用几种方法，包括：

- `adcli`
- `realmd`
- `Winbind`
- `Samba`

本节信息只介绍 Samba 方法。对于 `realmd`，请参阅 RHEL 或 CentOS 供应商的文档。必须在配置 SSSD 之前执行这些步骤。

使用 **Samba** 加入域并创建主机 **keytab**：

在正确配置了以下文件的 Linux 客户端上：

- `/etc/krb5.conf`

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

将计算机配置为进行 Samba 和 Kerberos 身份验证:

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

其中, **REALM** 是大写的 Kerberos 领域名称, 而 **domain** 是域的 NetBIOS 名称。

如果需要通过 DNS 查找 KDC 服务器和领域名称, 请将以下两个选项添加至前面的命令:

```
--enablekrb5kcdns --enablekrb5realmdns
```

打开 `/etc/samba/smb.conf` 并将以下条目添加到 **[Global]** 部分下方, 但要放在 **authconfig** 工具生成的部分后面:

```
kerberos method = secrets and keytab
winbind offline logon = no
```

加入 Windows 域要求您的域控制器可访问, 而且您具有有权将计算机添加到域的 AD 用户帐户:

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

REALM 是大写的 Kerberos 领域名称, **user** 是有权将计算机添加到域的域用户。

步骤 3: 设置 SSSD

注意

将 SSSD 与 Name Service Cache Daemon (NSCD) 一起使用可能会导致出现意外行为。有关详细信息, 请参阅 [Using NSCD with SSSD](#) (将 NSCD 与 SSSD 结合使用)。

设置 SSSD 的步骤如下:

- 在 Linux 客户端计算机上安装 **sssd-ad** 和 **sssd-proxy** 软件包。
- 对各种文件 (例如 **sssd.conf**) 进行配置更改。
- 启动 **sssd** 服务。

`/etc/sss/sss.conf` **sss.conf** 配置示例 (可以根据需要添加更多选项):

```
1 [sss]
2 config_file_version = 2
3 domains = EXAMPLE
4 services = nss, pam
5
6 [domain/EXAMPLE]
7 # Uncomment if you need offline logins
8 # cache_credentials = true
9 re_expression = (((?P<domain>[^\]+)\ (?P<name>.+)) | ((?P<name>[^\]+)@(?P
   <domain>.+)) | (^(?P<name>[^\]+)$))
```



```
10 id_provider = proxy
11 proxy_lib_name = nis
12 auth_provider = ad
13 access_provider = ad
14
15 # Should be specified as the long version of the Active Directory
    domain.
16 ad_domain = EXAMPLE.COM
17
18 # Kerberos settings
19 krb5_ccachedir = /tmp
20 krb5_ccname_template = FILE:%d/krb5cc_%U
21
22 # Uncomment if service discovery is not working
23 # ad_server = server.ad.example.com
24
25 # Comment out if the users have the shell and home dir set on the AD
    side
26 default_shell = /bin/bash
27 fallback_homedir = /home/%d/%u
28
29 # Uncomment and adjust if the default principal SHORTNAME$@REALM is not
    available
30 # ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
```

将 **ad.domain.com**、**server.ad.example.com** 替换为相应的值。有关详细信息，请参阅 [sssd-ad\(5\) - Linux 手册页](#)。

对 **sssd.conf** 设置文件所有权和权限：

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

步骤 4：配置 NSS/PAM

RHEL/CentOS:

使用 **authconfig** 启用 SSSD。安装 **oddjob mkhomedir** 以确保主目录创建与 SELinux 兼容：

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

提示：

配置 Linux VDA 设置时，要为 SSSD 考虑上述操作，而对 Linux VDA 客户端无需特殊设置。对于 **ctxsetup.sh** 脚本中的额外解决方案，请使用默认值。

步骤 5: 验证 Kerberos 配置

为了确保 Kerberos 已正确配置为可与 Linux VDA 配合使用, 请检查系统 **keytab** 文件是否已创建并且包含有效密钥:

```
1 sudo klist -ke
```

此命令显示各种主体名称与密码套件组合可用的密钥列表。运行 Kerberos **kinit** 命令, 以使用这些密钥向域控制器验证计算机的身份:

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

计算机和领域名称必须指定为大写。美元符号 (\$) 必须使用反斜杠 (\) 进行转义, 以免发生 shell 替换。在某些环境中, DNS 域名与 Kerberos 领域名称不同。请确保使用领域名称。如果此命令成功运行, 则不会显示任何输出。

请使用以下命令验证计算机帐户的 TGT 票据已缓存:

```
1 sudo klist -ke
```

步骤 6: 验证用户身份验证

使用 **getent** 命令确认支持的登录格式以及 NSS 是否工作:

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

DOMAIN 参数指示简短形式的域名。如果需要使用另一种登录格式, 请先使用 **getent** 命令进行验证。

支持的登录格式如下:

- 低级别登录名称: **DOMAIN\username**
- UPN: **username@domain.com**
- NetBIOS 前缀格式: **username@DOMAIN**

要验证 SSSD PAM 模块是否已正确配置, 请使用域用户帐户登录 Linux VDA。该域用户帐户以前未曾使用过。

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

检查是否为以下命令返回的 **uid** 创建了对应的 Kerberos 凭据缓存文件:

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

检查用户 Kerberos 凭据缓存中的票据是否有效且未过期:

```
1 klist
```

IPv6

July 15, 2024

Linux VDA 支持 IPv6 以便与 Citrix Virtual Apps and Desktops 保持一致。使用此功能时，请注意以下事项：

- 对于双堆栈环境，除非显式启用 IPv6，否则使用 IPv4。
- 如果在 IPv4 环境中启用了 IPv6，Linux VDA 将无法运行。

重要：

- 整个网络环境必须为 IPv6，而非仅针对 Linux VDA。
- Centrify 不支持纯 IPv6。

安装 Linux VDA 时，不需要对 IPv6 执行任何特殊设置任务。

为 Linux VDA 配置 IPv6

更改 Linux VDA 的配置之前，请确保您的 Linux 虚拟机以前在 IPv6 网络中运行。有两个与 IPv6 配置有关的注册表项：

```
1 "HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
  OnlyUseIPv6ControllerRegistration"
2 "HKLM\Software\Policies\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
  ControllerRegistrationIPv6Netmask"
```

必须将 **OnlyUseIPv6ControllerRegistration** 设置为 1 才能在 Linux VDA 上启用 IPv6：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
  Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
  OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
```

如果 Linux VDA 有多个网络接口，则可以使用 **ControllerRegistrationIPv6Netmask** 指定用于 Linux VDA 注册的网络接口：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
  Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "
  ControllerRegistrationIPv6Netmask " -d "{
2   IPv6 netmask }
3 " --force
```

请将 **{IPv6 netmask}** 替换为真实的网络掩码（例如 2000::/64）。

有关 Citrix Virtual Apps and Desktops 中的 IPv6 部署的详细信息，请参阅 [IPv4/IPv6 支持](#)。

故障排除

检查基础 IPv6 网络环境并使用 ping6 检查 AD 和 Delivery Controller 是否可访问。

LDAPS

July 15, 2024

LDAPS 是轻型目录访问协议 (LDAP) 的安全版本，其中 LDAP 通信使用 TLS/SSL 进行加密。

默认不加密客户端与服务器应用程序之间的 LDAP 通信。通过 LDAPS，您可以保护 Linux VDA 与 LDAP 服务器之间的 LDAP 查询内容。

以下 Linux VDA 组件在 LDAPS 上都具有依赖项：

- Broker 代理：Linux VDA 注册到 Delivery Controller 中
- 策略服务：策略评估

配置 LDAPS 涉及以下过程：

- 在 Active Directory (AD)/LDAP 服务器上启用 LDAPS
- 导出根 CA 以供客户端使用
- 在 Linux VDA 上启用/禁用 LDAPS
- 为第三方平台配置 LDAPS
- 配置 SSSD
- 配置 Winbind
- 配置 Centrify
- 配置 Quest

注意：

可以运行以下命令为 LDAP 服务器设置监视周期。默认值为 15 分钟。请至少将其设置为 10 分钟。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -v "ListOfLDAPServersMonitorPeroid" -t "REG_DWORD" -d "0x0000000f" --force
```

在 AD/LDAP 服务器上启用 LDAPS

可以通过安装 Microsoft 证书颁发机构 (CA) 或非 Microsoft CA 提供的格式正确的证书来启用通过 SSL 的 LDAP (LDAPS)。

提示：

在域控制器上安装企业根 CA 时将自动启用 LDAPS。

有关如何安装证书并验证 LDAPS 连接的详细信息，请参阅 [How to enable LDAP over SSL with a third-party certification authority](#)（如何借助第三方证书颁发机构启用通过 SSL 的 LDAP）。

当您有一个多层证书颁发机构层次结构时，您在域控制器上不会自动拥有用于 LDAPS 身份验证的合适证书。

有关如何使用多层证书颁发机构层次结构为域控制器启用 LDAPS 的信息，请参阅 [LDAP over SSL \(LDAPS\) Certificate](#)（通过 SSL 的 LDAP (LDAPS) 证书）一文。

启用根证书颁发机构以供客户端使用

客户端必须使用 LDAP 服务器信任的 CA 颁发的证书。要为客户端启用 LDAPS 身份验证，请将根 CA 证书导入到受信任的密钥库。

有关如何导出根 CA 的详细信息，请参阅 Microsoft 支持 Web 站点上的 [How to export Root Certification Authority Certificate](#)（如何导出根证书颁发机构）。

在 Linux VDA 上启用或禁用 LDAPS

要在 Linux VDA 上启用或禁用 LDAPS，请运行 **enable_ldaps.sh** 脚本（在以管理员身份登录时）。要在非交互模式下运行 **enable_ldaps.sh** 脚本，请将以下变量导出到您的环境中：

```
1 #CTX_LDAPS_KEYSTORE_PASSWORD=  
2  
3 #CTX_LDAPS_LDAP_SERVERS=
```

- 要通过提供的根 CA 证书启用通过 SSL/TLS 的 LDAP，请执行以下操作：

```
1 /opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
```

- 要使用通道绑定通过 SSL/TLS 启用 LDAP，请执行以下操作：

```
1 /opt/Citrix/VDA/sbin/enable_ldaps.sh -Enablecb pathToRootCA
```

注意：

用于通道绑定的根 CA 证书必须采用 PEM 格式。如果启用 LDAPS 无法成功创建 Python3 虚拟环境，请按照 [创建 Python3 虚拟环境](#) 中的说明手动创建该环境。

要解决在使用 pip 工具时可能会遇到的 SSL 连接错误，请考虑将以下可信主机添加到 `/etc/pip.conf` 文件中：

```
[global]  
trusted-host =
```

pypi.org
files.pythonhosted.org

- 回退到未启用 SSL/TLS 的 LDAP

```
“  
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Disable  
“
```

The Java keystore dedicated for LDAPS resides in **/etc/xdm/.keystore**. Affected registry keys include:

```
“  
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServers  
HKLM\Software\Citrix\VirtualDesktopAgent\ListOfLDAPServersForPolicy  
HKLM\Software\Citrix\VirtualDesktopAgent\UseLDAPS  
HKLM\Software\Policies\Citrix\VirtualDesktopAgent\Keystore  
HKLM\Software\Citrix\VirtualDesktopAgent\EnableChannelBinding  
“
```

为第三方平台配置 **LDAPS**

除 Linux VDA 组件外，还有多个附着于可能也需要安全 LDAP 的 VDA 的第三方软件组件，例如 SSSD、Winbind、Centrify 和 Quest。以下各部分介绍了如何通过 LDAPS、STARTTLS 或 SASL 签名和封装配置安全 LDAP。

提示：

并非所有这些软件组件都优先使用 SSL 端口 636 来确保安全 LDAP。并且大多数时间 LDAPS（端口 636 上通过 SSL 的 LDAP）不能与端口 389 上的 STARTTLS 共存。

SSSD

根据选项在端口 636 或端口 389 上配置 SSSD 安全 LDAP 流量。有关详细信息，请参阅 [SSSD LDAP Linux 手册页](#)。

Winbind

Winbind LDAP 查询使用 ADS 方法。Winbind 在端口 389 上仅支持 StartTLS 方法。受影响的配置文件为 **/etc/samba/smb.conf** 和 **/etc/openldap/ldap.conf**（对于 Amazon Linux 2、RHEL、Rocky Linux、CentOS 和 SUSE）或 **/etc/ldap/ldap.conf**（对于 Debian 和 Ubuntu）。按如下所示更改这些文件：

- `smb.conf`

```
ldap ssl = start tls
ldap ssl ads = yes
client ldap sasl wrapping = plain
```
- `ldap.conf`

```
TLS_REQCERT never
```

或者，可以通过 SASL GSSAPI 签名和封装配置安全 LDAP，但不能与 TLS/SSL 共存。要使用 SASL 加密，请更改 **smb.conf** 配置：

```
ldap ssl = off
ldap ssl ads = no
client ldap sasl wrapping = seal
```

Centrify

Centrify 在端口 636 上不支持 LDAPS。但它在端口 389 上提供安全加密。有关详细信息，请参阅 [Centrify 站点](#)。

Quest

Quest Authentication Service 在端口 636 上支持 LDAPS，但在端口 389 上使用其他方法提供安全加密。

故障排除

使用此功能时可能会引发以下问题：

- **LDAPS 服务可用性**

请确认 LDAPS 连接是否在 AD/LDAP 服务器上可用。默认情况下，端口为 636。
- 启用了 **LDAPS** 时 **Linux VDA** 注册失败

验证 LDAP 服务器和端口是否已正确配置。请先检查根 CA 证书，确保其与 AD/LDAP 服务器匹配。
- 意外错误地更改注册表项

如果在未使用 **enable_ldaps.sh** 的情况下意外更新了 LDAPS 相关的注册表项，则可能会破坏 LDAPS 组件的依赖项。
- **LDAP** 流量不通过 **SSL/TLS** 从 **Wireshark** 或任何其他网络监视工具加密

默认禁用 LDAPS。请运行 `/opt/Citrix/VDA/sbin/enable_ldaps.sh` 强制执行。

- 没有来自 **Wireshark** 或任何其他网络连接监视工具的 **LDAPS** 流量
发生 Linux VDA 注册和组策略评估时会出现 LDAP/LDAPS 流量。
- 无法通过在 **AD** 服务器上运行 **ldp connect** 验证 **LDAPS** 可用性
使用 AD FQDN 而非 IP 地址。
- 无法通过运行 **/opt/Citrix/VDA/sbin/enable_ldaps.sh** 脚本导入根 **CA** 证书
请提供 CA 证书的完整路径，并确认根 CA 证书的类型是否正确。它应与受支持的大多数 Java Keytool 类型兼容。如果它未在支持列表中列出，您可以先转换类型。如果遇到证书格式问题，我们建议您使用 base64 编码的 PEM 格式。
- 无法通过 **Keytool -list** 显示根 **CA** 证书
通过运行 **/opt/Citrix/VDA/sbin/enable_ldaps.sh** 启用 LDAPS 时，证书将被导入到 **/etc/xdm/.keystore** 中，并设置密码以保护密钥库。如果忘记了密码，可以重新运行该脚本以创建密钥库。

Xauthority

July 15, 2024

Linux VDA 支持使用 X11 显示功能（包括 **xterm** 和 **gvim**）进行交互式远程处理的环境。此功能提供必需的安全机制以确保 XClient 与 XServer 之间的通信安全。

可以通过两种方法确保此安全通信的权限安全：

- **Xhost**。默认情况下，Xhost 仅允许本地主机 XClient 与 XServer 进行通信。如果选择允许远程 XClient 访问 XServer，则必须运行 Xhost 命令授予对特定计算机的权限。或者，也可以使用 **xhost +** 以允许任意 XClient 连接到 XServer。
- **Xauthority**。可以在每个用户的主目录中找到 **.Xauthority** 文件。它用于将凭据存储在 xauth 使用的 cookie 中以用于对 XServer 进行身份验证。启动 XServer 实例 (Xorg) 后，该 cookie 将用于对该特定显示的连接进行身份验证。

工作原理

Xorg 启动时，**.Xauthority** 文件将被传递到 Xorg。此 **.Xauthority** 文件包含以下元素：

- 显示数量
- 远程请求协议
- cookie 数量

可以使用 `xauth` 命令浏览此文件。例如：

```
1 # xauth -f ~/.Xauthority
2
3 # > list
4
5 # > us01msip06:107 MIT-MAGIC-COOKIE-1
   fb228d1b695729242616c5908f11624b
```

如果 **XClient** 远程连接到 Xorg，则必须满足两个必备条件：

- 设置远程 XServer 的 **DISPLAY** 环境变量。
- 获取包含 Xorg 中的其中一个 cookie 数量的 `.Xauthority` 文件。

配置 Xauthority

要在 Linux VDA 上启用 **Xauthority** 以进行远程 X11 显示，必须创建下面两个注册表项：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
   CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "
   XauthEnabled" -d "0x00000001" --force
2
3 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
   CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "ListenTCP"
   -d "0x00000001" --force
```

启用 **Xauthority** 后，手动或通过挂载共享主目录将 `.Xauthority` 文件传递到 **XClient**：

- 手动将 `.Xauthority` 文件传递给 XClient

启动 ICA 会话后，Linux VDA 将为 XClient 生成 `.Xauthority` 文件，并将该文件存储在登录用户的主目录中。可以将此 `.Xauthority` 文件复制到远程 XClient 计算机，并设置 **DISPLAY** 和 **XAUTHORITY** 环境变量。**DISPLAY** 是存储在 `.Xauthority` 文件中的显示编号，**XAUTHORITY** 是 **Xauthority** 的文件路径。例如，请查看以下命令：

```
1 export DISPLAY={
2   Display number stored in the Xauthority file }
3
4
5 export XAUTHORITY={
6   the file path of .Xauthority }
```

注意：

如果未设置 **XAUTHORITY** 环境变量，则默认使用 `~/.Xauthority` 文件。

- 通过挂载共享主目录将 `.Xauthority` 文件传递到 XClient

最便捷的方式是为登录用户装载共享主目录。当 Linux VDA 启动 ICA 会话时，将在登录用户的主目录下创建 `.Xauthority` 文件。如果此主目录与 XClient 共享，用户不需要手动将此 `.Xauthority` 文件传输到 XClient。正确设置 **DISPLAY** 和 **XAUTHORITY** 环境变量后，将自动在 XServer 桌面中显示 GUI。

故障排除

如果 **Xauthority** 无法正常运行，请按照故障排除步骤进行操作：

1. 以具有 root 权限的管理员身份获取所有 Xorg cookie：

```
1 ps aux | grep -i xorg
```

此命令将显示启动过程中传递到 Xorg 的 Xorg 进程和参数。另一个参数显示使用的 `.Xauthority` 文件。例如：

```
1 /var/xdl/xauth/.Xauthority110
```

使用 **Xauth** 命令显示 cookie：

```
1 Xauth -f /var/xdl/xauth/.Xauthority110
```

2. 使用 **Xauth** 命令显示 `~/.Xauthority` 中包含的 cookie。如果显示编号相同，则显示的 cookie 必须与 Xorg 和 XClient 的 `.Xauthority` 文件中的 cookie 相同。
3. 如果 cookie 相同，请检查是否能够使用 Linux VDA 的 IP 地址访问远程显示端口，并检查已发布的桌面的显示数量。

例如，在 XClient 计算机上运行以下命令：

```
1 telnet 10.158.11.11 6160
```

端口号为 `6000 + \< 显示数量\>` 的总和。

如果此 Telnet 操作失败，防火墙可能在阻止请求。

身份验证

July 15, 2024

本部分内容包含以下主题：

- [使用 Azure Active Directory 进行身份验证](#)
- [双跃点单点登录身份验证](#)
- [联合身份验证服务](#)

- [非 SSO 身份验证](#)
- [智能卡](#)
- [未经身份验证的（匿名）用户进行访问](#)

使用 **Azure Active Directory** 进行身份验证

August 20, 2024

注意：

此功能仅适用于 Azure 托管的 VDA。

根据您的需求，可以在 Azure 中部署两种类型的 Linux VDA：

- 加入了 Azure AD DS 的 VM。VM 已加入到 Azure Active Directory (AAD) 域服务 (DS) 托管的域。用户使用其域凭据登录 VM。
- 未加入域的 VM。VM 与 AAD 标识服务集成以提供用户身份验证。用户使用其 AAD 凭据登录 VM。

有关 AAD DS 和 AAD 的详细信息，请参阅这篇 [Microsoft 文章](#)。

本文介绍了如何在未加入域的 VDA 上启用和配置 AAD 标识服务。

支持的发行版

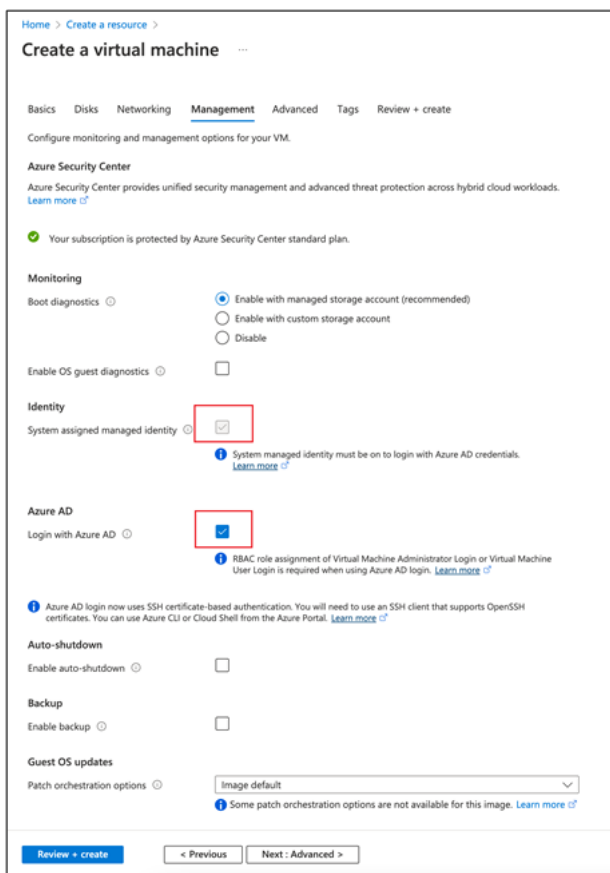
- Ubuntu 22.04、20.04
- RHEL 8.10、8.9、8.8
- SUSE 15.5

有关详细信息，请参阅这篇 [Microsoft 文章](#)。

步骤 1：在 **Azure** 门户上创建模板 VM

创建模板 VM 并在 VM 上安装 Azure CLI。

1. 在 Azure 门户上，创建模板 VM。在单击检查 + 创建之前，请务必在管理选项卡上选择 **Login with Azure AD**（使用 Azure AD 登录）。



2. 在模板 VM 上安装 Azure CLI。
有关详细信息，请参阅这篇 [Microsoft 文章](#)。

步骤 2: 在模板 VM 上准备主映像

要准备主映像，请按照[使用 MCS 在 Azure 上创建 Linux VM](#) 中的步骤 3: 准备主映像进行操作。

步骤 3: 将模板 VM 设置为未加入域的模式

创建主映像后，请按照以下步骤将 VM 设置为未加入域的模式：

1. 从命令提示符运行以下脚本。

```
1 Modify /var/xdl/mcs/mcs_util.sh
```

2. 找到 `function read_non_domain_joined_info()`，然后将 `NonDomainJoined` 的值更改为 2。有关示例，请参阅以下代码块。

```
1 function read_non_domain_joined_info()  
2 {  
3
```

```

4 log "Debug: Enter read_non_domain_joined_info"
5 # check if websocket enabled
6 TrustIdentity=`cat ${
7   id_disk_mnt_point }
8   ${
9     ad_info_file_path }
10  | grep '[TrustIdentity]' | sed 's/\s//g'`
11 if [ "$TrustIdentity" == "[TrustIdentity]" ]; then
12   NonDomainJoined=2
13 fi
14 ...
15 }

```

3. 保存更改。
4. 关闭模板 VM。

步骤 4: 从模板 VM 创建 Linux VM

准备好未加入域的模板 VM 后，请按照以下步骤创建 VM：

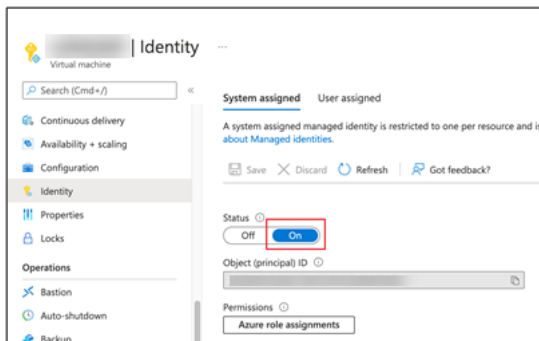
1. 登录 Citrix Cloud。
2. 双击 Citrix DaaS，然后访问完整配置管理控制台。
3. 在计算机目录中，选择使用 Machine Creation Services 从模板 VM 创建 Linux VM。有关详细信息，请参阅 Citrix DaaS 文档中的[未加入域的 VDA](#)。

步骤 5: 将 AAD 用户帐户分配给 Linux VM

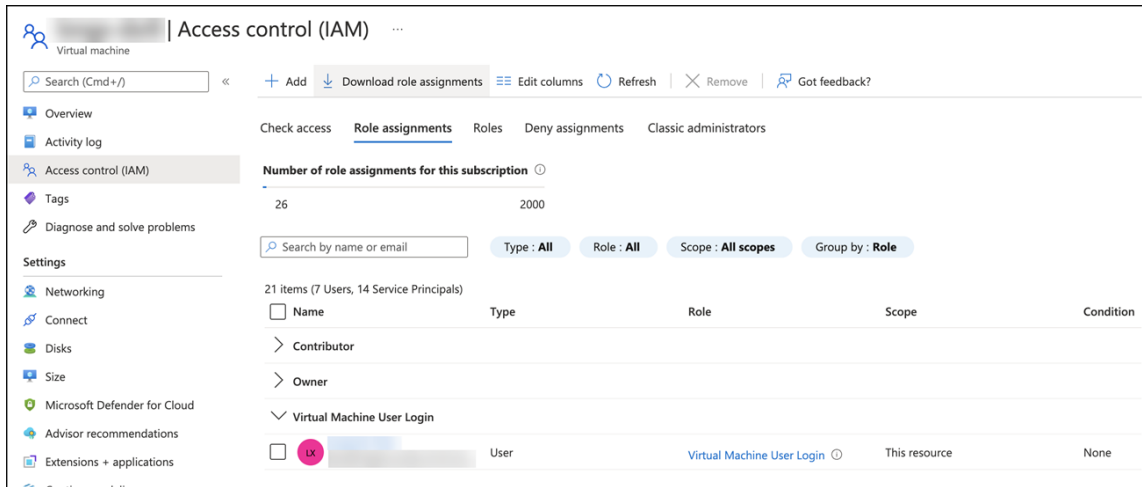
创建未加入域的 VM 后，为其分配 AAD 用户帐户。

要将 AAD 用户帐户分配给 VM，请执行以下步骤：

1. 使用管理员帐户访问 VM。
2. 在 **Identify**（识别）> **System assigned**（系统分配）选项卡上，启用 **System Identity**（系统标识）。



3. 在 **Access control (IAM)** (访问控制 (IAM)) > **Role assignments** (角色分配) 选项卡上, 找到 **Virtual Machine User Login** (虚拟机用户登录) 区域, 然后根据需要添加 AAD 用户帐户。

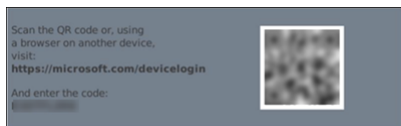


登录到未加入域的 VDA

贵组织中的最终用户可以通过两种方式登录未加入域的 VDA。详细步骤如下所述：

1. 启动 Workspace 应用程序, 然后通过输入 AAD 用户名和密码登录到 Workspace。此时将显示 Workspace 页面。
2. 双击未加入域的桌面。此时将显示 AAD 登录页面。

该页面因 VDA 上设置的登录模式而异：设备代码或 AAD 帐户/密码。默认情况下, Linux VDA 使用设备代码登录模式对 AAD 用户进行身份验证, 如下所示。作为管理员, 您可以根据需要将登录模式更改为 AAD 帐户/密码。有关详细步骤, 请参阅以下部分。



3. 根据屏幕上的说明, 通过以下方式之一登录到桌面会话：
 - 扫描 QR 代码并输入该代码。
 - 输入 AAD 用户名和密码。

更改为 AAD 帐户/密码登录模式

默认情况下, Linux VDA 使用设备代码对 AAD 用户进行身份验证。有关详细信息, 请参阅这篇 [Microsoft 文章](#)。要将登录模式更改为 AAD 帐户/密码, 请执行以下步骤：

在 VDA 上运行以下命令, 找到注册表项 `AADAcctPwdAuthEnable`, 然后将其值更改为 `0x00000001`。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\
  Services\CitrixBrokerAgent\WebSocket" -t "REG_DWORD" -v "
  AADAcctPwdAuthEnable" -d "0x00000001" --force
```

注意：

此方法不适用于 Microsoft 帐户或启用了双重身份验证的帐户。

双跃点单点登录身份验证

August 20, 2024

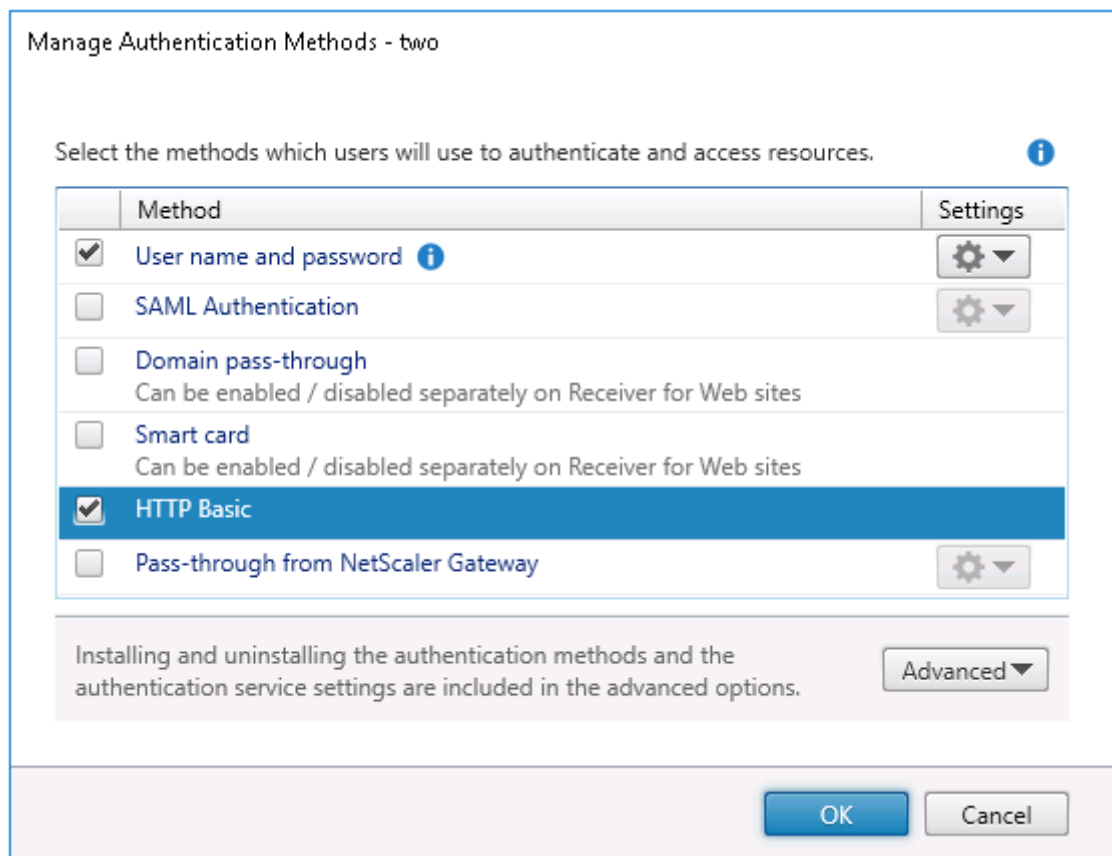
可以将用于访问 StoreFront 应用商店的用户凭据注入到适用于 Linux 的 Citrix Workspace 应用程序和 Citrix Receiver for Linux 13.10 的 AuthManager 模块中。注入后，您可以使用该客户端访问 Linux 虚拟桌面会话中的虚拟桌面和应用程序，而无需再次输入用户凭据。

注意：

适用于 Linux 的 Citrix Workspace 应用程序和 Citrix Receiver for Linux 13.10 支持此功能。

要启用此功能，请执行以下操作：

1. 在 Linux VDA 上，安装适用于 Linux 的 Citrix Workspace 应用程序或 Citrix Receiver for Linux 13.10。
从 Citrix Workspace 应用程序或 Citrix Receiver 的 [Citrix 下载页面](#) 下载相应应用程序。
默认安装路径为 /opt/Citrix/ICAClient/。如果要将在应用程序安装在其他路径中，请将 ICAROOT 环境变量设置为指向实际安装路径。
2. 在 Citrix StoreFront 管理控制台中，为目标应用商店添加 **HTTP** 基本身份验证方法。



3. 将以下注册表项添加到 AuthManager 配置文件 (\$ICAROOT/config/AuthManConfig.xml) 以允许进行 HTTP 基本身份验证:

```

1 <Protocols>
2   <HTTPBasic>
3     <Enabled>True</Enabled>
4   </HTTPBasic>
5 </Protocols>

```

4. 请运行以下命令以在指定目录中安装根证书。

```

1 cp rootcert.pem $ICAROOT/keystore/cacerts/
2 $ICAROOT/util/ctx_rehash $ICAROOT/keystore/cacerts/

```

5. 请运行以下命令以启用该功能:

```

1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d "0
  x00000001"

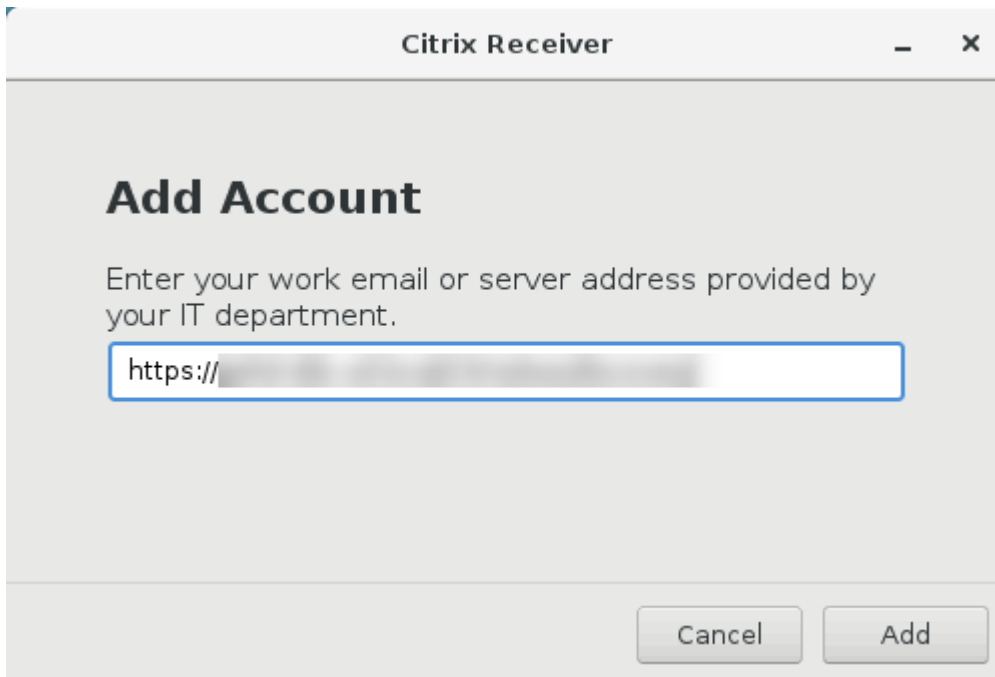
```

6. 启动 Linux 虚拟桌面会话和该会话中的适用于 Linux 的 Citrix Workspace 应用程序或 Citrix Receiver for Linux 13.10。

首次启动 Citrix Workspace 应用程序时, 系统会提示您输入应用商店帐户。以后您可以自动登录到之前指定的应用商店。

注意：

输入 HTTPS URL 作为您的应用商店帐户。



联合身份验证服务

August 20, 2024

可以使用联合身份验证服务 (FAS) 对登录到 Linux VDA 的用户进行身份验证。对于 FAS 登录功能，Linux VDA 将使用与 Windows VDA 相同的 Windows 环境。有关为 FAS 配置 Windows 环境的信息，请参阅[联合身份验证服务](#)。本文提供了特定于 Linux VDA 的额外信息。

注意：

- Linux VDA 不支持会话中行为策略。
- Linux VDA 使用短连接与 FAS 服务器传输数据。

支持的发行版

FAS 支持有限的 Linux 发行版和域加入方法。请参阅以下列表：

	Winbind	SSSD	Centrify	PBIS	Quest
Amazon Linux 2	是	是	是	是	否
Debian 12.5/11.9	是	是	是	是	否
RHEL 9.4/9.3/9.2	是	是	否	否	是
RHEL 8.10/8.9/8.8	是	是	是	是	是
Rocky Linux 9.4/9.3/9.2	是	是	否	否	否
Rocky Linux 8.10/8.9/8.8	是	是	否	否	否
SUSE 15.5	是	是	是	否	否
Ubuntu 22.04/20.04	是	是	是	是	是

在 Linux VDA 上配置 FAS

安装证书

要验证用户的证书，请在 VDA 上安装根 CA 证书以及所有中间证书。例如，要安装根 CA 证书，请从前面的从 **Microsoft CA**（在 **AD** 中）检索 **CA** 证书步骤中获取 AD 根证书，或者从根 CA 服务器 <http://CA-SERVER/certsrv> 下载该证书。

注意：

以下命令也适用于配置中间证书。

例如，要将 DER 文件（.crt、.cer、.der）转换为 PEM，请运行类似于以下内容的命令：

```
1 sudo openssl x509 -inform der -in root.cer -out root.pem
```

然后，运行类似如下的命令将根 CA 证书安装到 `openssl` 目录：

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

注意：

请勿将根 CA 证书置于 `/root` 路径下。否则，FAS 无权读取根 CA 证书。

运行 `ctxfascfg.sh`

运行 `ctxfascfg.sh` 脚本以配置 FAS:

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

您可以在物体时模式下运行 `ctxfascfg.sh`。在无提示模式下运行该脚本之前，请设置以下环境变量：

- **CTX_FAS_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis | quest:** 表示 Active Directory 集成方法，该方法在指定了 `CTX_EASYINSTALL_ADINTEGRATIONWAY` 时为 `CTX_EASYINSTALL_ADINTEGRATIONWAY`。如果未指定 `CTX_EASYINSTALL_ADINTEGRATIONWAY`，`CTX_FAS_ADINTEGRATIONWAY` 将使用自己的值设置。
- **CTX_FAS_CERT_PATH =<certificate path>:** 指定存储根证书和所有中间证书的完整路径。
- **CTX_FAS_KDC_HOSTNAME:** 在选择 PBIS 和 Quest 时指定密钥发行中心 (KDC) 的主机名。
- **CTX_FAS_PKINIT_KDC_HOSTNAME:** 指定 PKINIT KDC 主机名，除非另有说明，否则该名称为 `CTX_FAS_KDC_HOSTNAME`。如果您有多个 Delivery Controller，请将该域的所有 KDC 的主机名添加到 `/etc/krb5.conf` 文件中的 `pkinit_kdc_hostname` 中。有关详细信息，请参阅知识中心文章 [CTX322129](#)。
- **CTX_FAS_SERVER_LIST=' list-fas-servers'** - 联合身份验证服务 (FAS) 服务器是通过 AD 组策略配置的。有关域 GPO 上的 FAS 策略设置的信息，请参阅 [配置组策略](#)。Linux VDA 不支持 AD 组策略，但您可以改为提供以分号分隔的 FAS 服务器的列表。顺序必须与在 AD 组策略中配置的顺序相同。如果删除了任何服务器地址，请使用 **<none>** 文本字符串填充其空白，并且不要修改服务器地址的顺序。要与 FAS 服务器正确通信，请确保附加的端口号与在 FAS 服务器上指定的端口号一致，例如 `CTX_XDL_FAS_LIST=' fas_server_1_url:port_number; fas_server_2_url: port_number; fas_server_3_url: port_number'`。

要升级现有 Linux VDA 安装，您可以运行以下命令来设置 FAS 服务器并重新启动 `ctxvda` 服务以使您的设置生效。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
  VirtualDesktopAgent\Authentication\UserCredentialService" -t "
  REG_SZ" -v "Addresses" -d "<Your-FAS-Server-List>" --force
2
3 systemctl restart ctxjproxy
4
5 systemctl restart ctxvda
```

要通过 `ctxreg` 更新 FAS 服务器，请运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
  VirtualDesktopAgent\Authentication\UserCredentialService" -v "
  Addresses" -d "<Your-FAS-Server-List>"
2
3 systemctl restart ctxjproxy
4
5 systemctl restart ctxvda
```

选择正确的 Active Directory 集成方法，然后键入正确的证书路径（例如 `/etc/pki/CA/certs/`）。

脚本随后安装 `krb5-pkinit` 和 `pam_krb5` 软件包，并设置相关的配置文件。对于 RHEL 8 及更高版本，将 `PAM_KRB5` 移至 EPEL 存储库时，脚本会尝试在这些发行版中启用 EPEL。

禁用 FAS

要在 Linux VDA 上禁用 FAS，请使用以下命令从 ConfDB 中删除所有 FAS 服务器：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\  
VirtualDesktopAgent\Authentication\UserCredentialService" -t "REG_SZ"  
" -v "Addresses" -d "" --force  
2  
3 systemctl restart ctxjproxy  
4  
5 systemctl restart ctxvda
```

限制

- FAS 尚不支持锁屏界面。如果在会话中单击锁定按钮，则无法使用 FAS 重新登录该会话。
- 此版本仅支持[联合身份验证服务体系结构概述](#)一文中概括的常见 FAS 部署，不包括 **Windows 10 Azure AD** 联接。

故障排除

在对 FAS 进行故障排除之前，请确保：

- Linux VDA 的安装和配置正确无误。
- 使用密码身份验证可以在公用存储中成功启动非 FAS 会话。

如果非 FAS 会话正常运行，请将登录类别的 HDX 日志级别设置为“VERBOSE”，将 VDA 日志级别设置为“TRACE”。有关为 Linux VDA 启用跟踪日志记录的信息，请参阅知识中心文章 [CTX220130](#)。

也可以使用 Linux [XDPing](#) 工具检查您的 Linux VDA 环境中可能存在的常见配置问题。

FAS 服务器配置错误

从 FAS 应用商店启动会话失败。

检查 `/var/log/xdl/hdx.log` 并查找类似如下内容的错误日志：

```
1 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [  
Logon Type] Federated Authentication Logon.  
2
```

```
3 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas:
  entry
4
5 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start
  connect to server 0
6
7 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0:
  failed to connect: Connection refused.
8
9 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas:
  failed to connect to server [0], please confirm if fas service list
  is well configured in condb
10
11 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit
  , 43
12
13 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user:
  failed to validate fas credential
14
15 2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate:
  failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER
```

解决方案 请运行以下命令以确认 Citrix 注册表值“HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent\Auth”是否设置为 <Your-FAS-Server-List>。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
```

如果现有设置不正确，请按照前面的[设置 FAS 服务器](#)步骤操作以重新设置。

CA 证书配置错误

从 FAS 应用商店启动会话失败。将显示一个灰色窗口，并且几秒钟后消失。



检查 `/var/log/xdl/hdx.log` 并查找类似如下内容的错误日志：

```
1 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
  get_logon_certificate: entry
2
```

```
3 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller:
   current process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]
4
5 2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
   get_public_certificate: entry
6
7 2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
   for response...
8
9 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query
   to server success
10
11 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
   get_public_certificate: exit
12
13 2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
   input size 1888
14
15 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
   output size 1415
16
17 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
   get_logon_certificate: get logon certificate success
18
19 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
   cache certificate success
20
21 2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
   get_logon_certificate: exit, 0
22
23 2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
   pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
```

解决方案 验证您是否已在 `/etc/krb5.conf` 中正确设置存储根 CA 证书和所有中间证书的完整路径。完整路径如下所示：

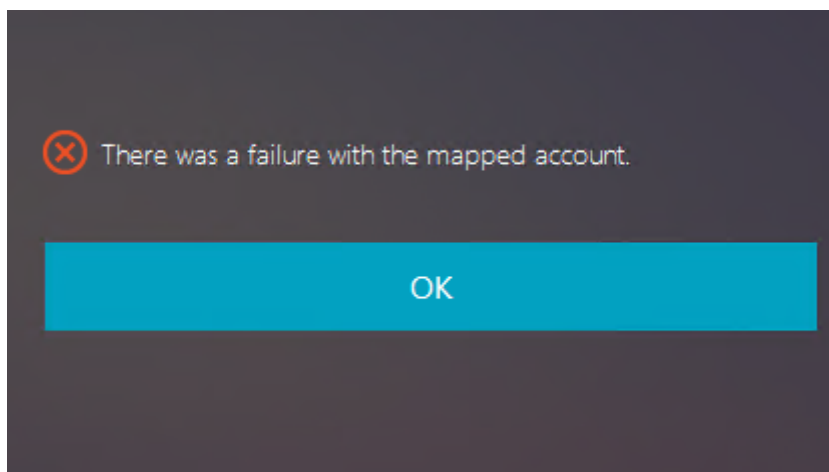
```
1  [realms]
2
3  EXAMPLE.COM = {
4
5
6      .....
7
8      pkinit_anchors = DIR:/etc/pki/CA/certs/
9
10     .....
11
12 }
```

如果现有设置不正确，请按照前面的[安装证书](#)步骤重新对其进行设置。

或者，检查根 CA 证书是否有效。

重影帐户映射错误

FAS 配置了 SAML 身份验证。ADFS 用户在 ADFS 登录页面上输入用户名和密码后，可能会出现以下错误。



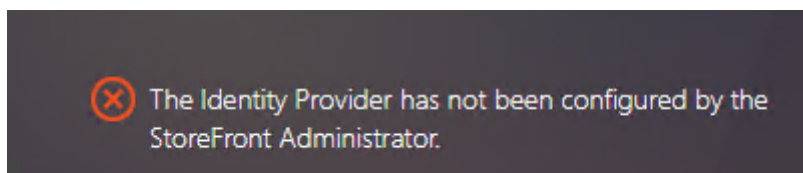
该错误表明：

- 已成功验证 ADFS 用户，但在 AD 上没有配置任何重影用户。

解决方案 在 AD 上设置重影帐户。

未配置 ADFS

尝试登录 FAS 应用商店过程中出现以下错误：



当您将 FAS 应用商店配置为使用 SAML 身份验证，但缺少 ADFS 部署时，会出现此问题。

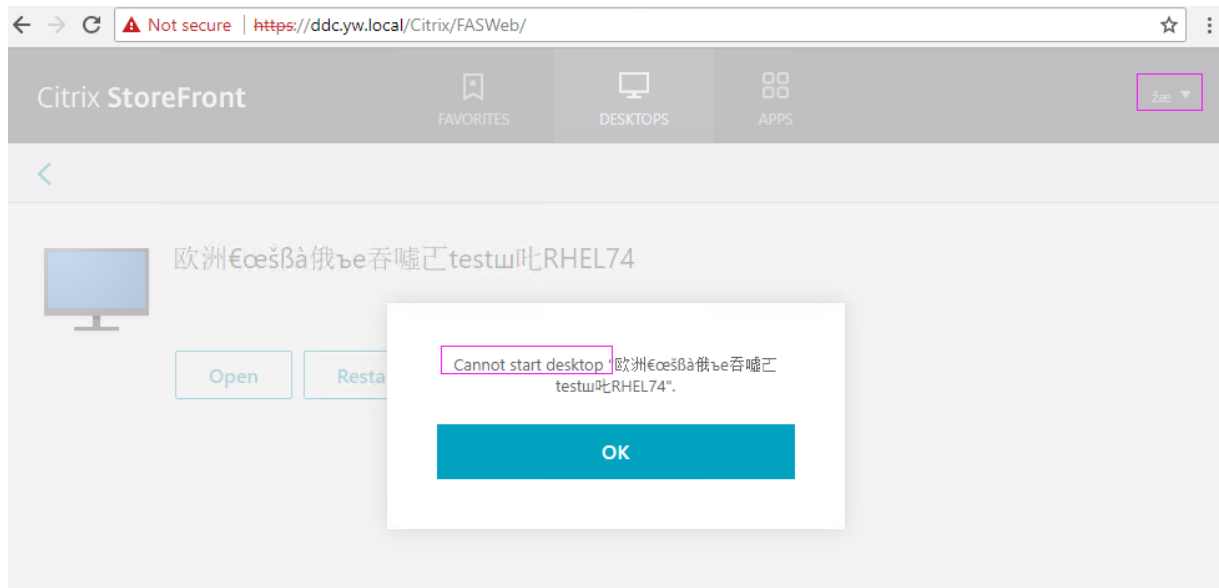
解决方案 为联合身份验证服务部署 ADFS IdP。有关详细信息，请参阅[联合身份验证服务 ADFS 部署](#)。

相关信息

- 通用 FAS 部署在[联合身份验证服务体系结构概述](#)一文中加以概括。
- [联合身份验证服务高级配置](#)一章中引入了“操作方法”文章。

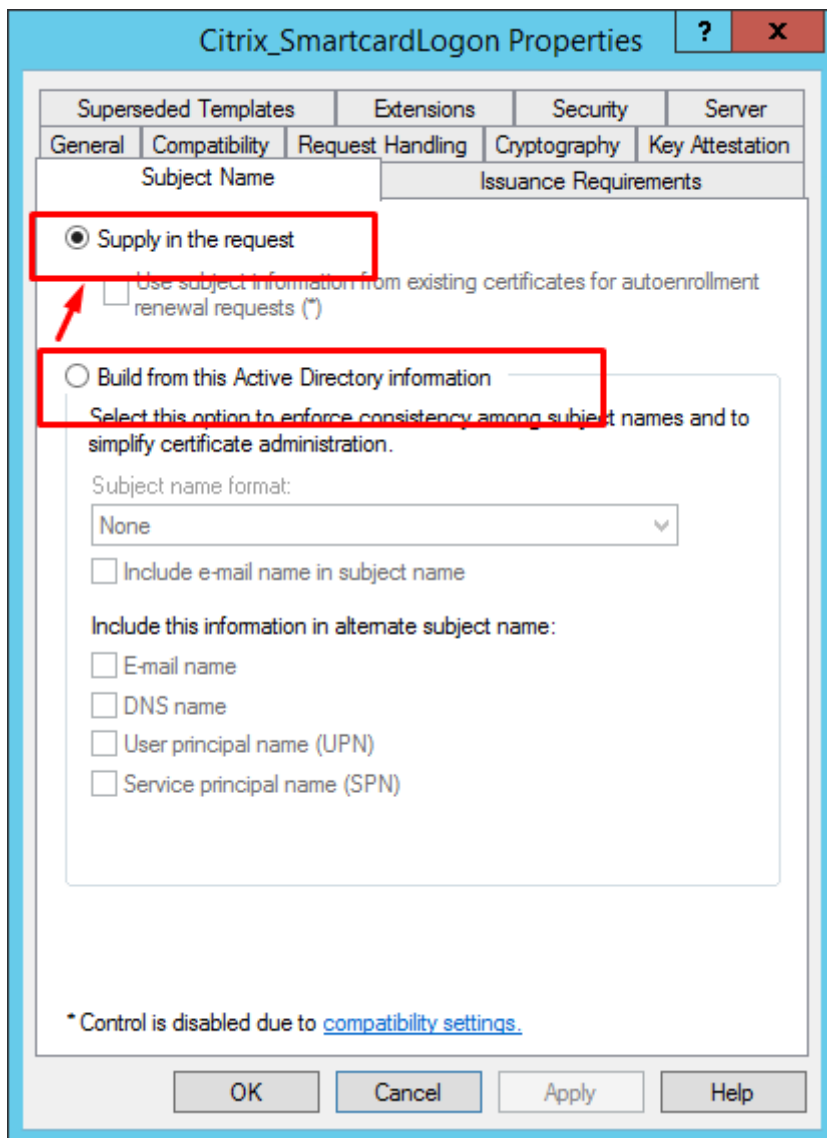
已知问题

如果正在使用 FAS，则尝试启动包含非英语字符的已发布桌面或应用程序会话时，可能会失败。



解决方法

在 CA 工具中右键单击 **Manage Templates**（管理模板）以将 **Citrix_SmartcardLogon** 模板从 **Build from this Active Directory information**（基于此 Active Directory 信息构建）更改为 **Supply in the request**（在请求中提供）：



FIDO2 (预览版)

July 15, 2024

可以设置 FIDO2 身份验证，以使用 Linux VDA 上托管的 Google Chrome 访问 Web 站点。

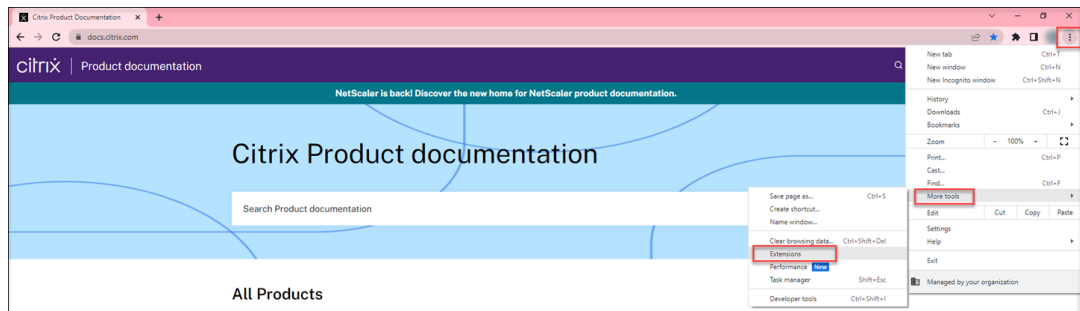
注意：

此功能在预览版中提供。预览功能可能未完全本地化，建议在非生产环境中使用。Citrix 技术支持不对在预览功能中发现的问题提供支持。

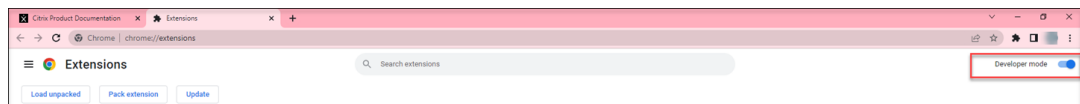
Linux VDA 仅支持 FIDO2 和 Google Chrome 的组合。

要设置 FIDO2 身份验证，请完成以下步骤：

1. 下载 Citrix FIDO2 扩展程序软件包。
 - a) 转至 [Citrix Virtual Apps and Desktops 下载页面](#)。
 - b) 展开适当版本的 Citrix Virtual Apps and Desktops。
 - c) 单击组件找到 Linux VDA。
 - d) 单击 Linux VDA 打开其下载页面。
 - e) 下载源软件包。
 - f) 解压源软件包即可找到 **FIDO2-JavaScript-Extensions.zip**。
 - g) 解压 FIDO2 扩展程序软件包。可以在扩展程序 > **chrome** > **fido2** 中找到 FIDO2 扩展程序目录。
2. 在 Google Chrome 中添加 Citrix FIDO2 扩展程序：
 - a) 打开 Linux VDA 上托管的 Google Chrome。
 - b) 单击地址栏右侧的三点菜单，然后选择更多工具 > 扩展程序。



- c) 开启开发人员模式。



- d) 单击 **Load unpacked** (加载解压内容)，然后在扩展程序 > **chrome** > **fido2** 中选择扩展程序目录。
3. 在您想要对其使用 FIDO2 身份验证的 Web 站点中，注册 FIDO2 安全密钥以使用 FIDO2 身份验证。
 - a) 将 FIDO2 安全密钥插入安装了 Citrix Workspace 应用程序的客户端。
 - b) 启用多重身份验证并添加 FIDO2 作为身份验证方法。

设置 FIDO2 身份验证后，系统将提示您触摸安全密钥以成功访问该 Web 站点。

非 SSO 身份验证

July 15, 2024

本文提供有关如何在 Linux VDA 上启用非 SSO 身份验证的指导。

概述

默认情况下，Linux VDA 已启用单点登录 (SSO)。用户使用一组凭据登录 Citrix Workspace 应用程序和 VDA 会话。

要让用户使用一组不同的凭据登录 VDA 会话，请在 Linux VDA 上禁用 SSO。下表列出了非 SSO 场景中支持的用户身份验证方法的组合。

Citrix Workspace 应用程序	VDA 会话
用户名	用户名
智能卡	用户名
用户名	智能卡
FAS	用户名
FAS	智能卡

禁用 SSO

在您的 Linux VDA 上运行以下命令：

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

智能卡

August 20, 2024

登录 Linux Virtual Desktop 会话时，可以使用连接到客户端设备的智能卡进行身份验证。此功能是通过 ICA 智能卡虚拟通道进行的智能卡重定向实现的。还可以在会话中使用智能卡。用例包括：

- 向文档添加数字签名
- 加密或解密电子邮件
- 对 Web 站点进行身份验证

对于此功能，Linux VDA 使用与 Windows VDA 相同的配置。有关详细信息，请参阅本文中的[配置智能卡环境](#)部分。

注意：

不支持在 Linux VDA 会话中使用映射的智能卡登录 Citrix Gateway。

必备条件

智能卡直通身份验证的可用性取决于以下条件：

- 您的 Linux VDA 安装在下面其中一个发行版中：
 - RHEL 9.4/ 9.3/9.2
 - RHEL 8.10/8.9/8.8
 - Rocky Linux 9.4/ 9.3/9.2
 - Rocky Linux 8.10/8.9/8.8
 - Ubuntu 22.04
 - Ubuntu 20.04
 - Debian 12.5/11.9

完成安装 VDA 后，请验证您的 VDA 是否能够在 Delivery Controller 中注册，以及是否能够使用 Windows 凭据打开已发布的 Linux 桌面会话。

- 使用 OpenSC 支持的智能卡。有关详细信息，请参阅[确保 OpenSC 支持智能卡](#)。

确保 OpenSC 支持智能卡

OpenSC 是 RHEL 7.4+ 上广泛使用的智能卡驱动程序。作为 CoolKey 的完全兼容替代品，OpenSC 支持多种类型的智能卡（请参阅 [Red Hat Enterprise Linux 中的智能卡](#)）。

在本文中，YubiKey 智能卡用作阐明配置的示例。YubiKey 是一个通用 USB CCID PIV 设备，可以很容易地从 Amazon 或其他零售供应商处购买。OpenSC 驱动程序支持 YubiKey。

如果贵组织需要某些其他更高级的智能卡，请准备一台安装了 Linux 发行版或 OpenSC 软件包的物理机。有关 OpenSC 安装的信息，请参阅[安装智能卡驱动程序](#)。插入智能卡，然后运行以下命令以确认 OpenSC 是否支持智能卡：

```
1 pkcs11-tool --module opensc-pkcs11.so --list-slots
```

配置

准备根证书

根证书用于验证智能卡上的证书。完成以下步骤以下载并安装根证书。

1. 获取 PEM 格式的根证书，通常从 CA 服务器中获取。

可以运行类似如下的命令以将 DER 文件 (*.crt、*.cer、*.der) 转换为 PEM。在下面的命令示例中，**certnew.cer** 为 DER 文件。

```
1 openssl x509 -inform der -in certnew.cer -out certnew.pem
```

2. 在 `openssl` 目录中安装根证书。 `certnew.pem` 文件用作示例。

```
1 cp certnew.pem <path where you install the root certificate>
```

要创建安装根证书的路径，请运行 `sudo mkdir -p <path where you install the root certificate>`。

配置智能卡环境

可以使用 `ctxsmartlogon.sh` 脚本配置智能卡环境或手动完成配置。

- (选项 1) 使用 `ctxsmartlogon.sh` 脚本配置智能卡环境

注意：

`ctxsmartlogon.sh` 脚本将 PKINIT 信息添加到默认领域。可以通过 `/etc/krb5.conf` 配置文件更改此设置。

首次使用智能卡之前，请运行 `ctxsmartlogon.sh` 脚本以配置智能卡环境。

提示：

如果已使用 SSSD 加入域，请在运行 `ctxsmartlogon.sh` 后重新启动该 SSSD 服务。

对于 RHEL 8 及更高版本，将 `PAM_KRB5` 移至 EPEL 存储库时，脚本将尝试在这些发行版中启用 EPEL。

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

结果类似于以下内容：

```
*****
# ctxsmartlogon.sh sets up smart card logon for the linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
*****
Step 1:Enable smart card logon.
Do you want enable smart card logon? (y/n)[y] y
Step 2:Select the Active Directory integration method.
Please select which Active Directory integration method to use:
  1: Winbind
  2: SSSD
  3: Centrify
Select one of the above options (1-3)[1] 1
Step 3:Install dependent packages.
Installing, please wait...
[krb5-pkinit][pam_krb5] already installed.
[pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][coolkey] already installed.
Packages installed. [Success]
Step 4:Configure krb5.conf.
Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
/etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/libcoolkeypk11.so):/usr/lib64/pkcs11/libcoolkeypk11.so
/etc/pam.d/ctxfsc configure successfully.
/etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

注意：

如果您使用 Quest 作为域加入方法，则需要您指定密钥发行中心 (KDC) 的主机名。例如：

```
[root@jx-rh79qu110 ~]# /opt/Citrix/VDA/sbin/ctxsmartlogon.sh
#*****
# ctxsmartlogon.sh sets up smart card logon for the Linux VDA, which
# includes automatic installation of the necessary packages and changes to
# the configuration files.
#
# Note:
# The ctxsmartlogon.sh adds pkinit information to the default realm. You can
# change this setting through the /etc/krb5.conf configuration file.
#*****
Step 1:Enable smart card logon.
Do you want enable smart card logon? (y/n)[y]
Step 2:Select the Active Directory integration method.
Please select which Active Directory integration method to use:
  1: Winbind
  2: SSSD
  3: Centrifify
  4: Quest
Select one of the above options (1-4)[1] 4
Step 3:Install dependent packages.
Installing, please wait...
[krb5-pkinit][pam krb5] already installed.
[pcsc-lite][pcsc-lite-ccid][pcsc-lite-libs][opensc] already installed.
Packages installed. [Success]
Step 4:Configure krb5.conf.
Specify the path to the root CA (e.g., /etc/pki/CA/certs/root.pem):/etc/pki/CA/certs/root.pem
Specify the KDC hostname:kdc.test.local
/etc/krb5.conf configure successfully.
Step 5:Configure PAM files.
Specify the path to the smart card PKCS11 driver (e.g., /usr/lib64/pkcs11/opensc-pkcs11.so):/usr/lib64/pkcs11/opensc-pkcs11.so
/etc/pam.d/ctxfsc configure successfully.
/etc/pam.d/smartcard-auth configure successfully.
ctxsmartlogon.sh executed successfully. SmartCard is ready.
```

此外，您还可以通过运行 ctxsmartlogon.sh 脚本来禁用智能卡：

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

结果类似于以下内容：

```
#*****
# 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.
```

(选项 2) 手动配置智能卡环境 Linux VDA 使用与 Windows VDA 相同的智能卡环境。在该环境中，必须配置多个组件，包括域控制器、Microsoft 证书颁发机构 (CA)、Internet Information Services、Citrix StoreFront 和 Citrix Workspace 应用程序。有关基于 YubiKey 智能卡的配置信息，请参阅知识中心文章 [CTX206156](#)。

在进入下一步之前，请确保：

- 您已正确配置了所有组件。
- 您已将私钥和用户证书下载到智能卡。
- 您可以使用智能卡成功登录 VDA。

安装 **PC/SC Lite** 软件包 PCSC Lite 是个人计算机/智能卡 (PC/SC) 规范在 Linux 中的实现。它提供与智能卡和读卡器进行通信的 Windows 智能卡接口。Linux VDA 中的智能卡重定向是在 PC/SC 级别实现的。

运行以下命令可安装 PC/SC Lite 软件包：

RHEL 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：

```
1 yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
```

Ubuntu 22.04、Ubuntu 20.04、Debian 12.5、Debian 11.9：

```
1 apt-get install -y libpcsclite1 libccid
```

安装智能卡读卡器 OpenSC 是广泛使用的智能卡驱动程序。如果未安装 OpenSC，请运行以下命令以进行安装：

RHEL 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：

```
1 yum install opensc
```

Ubuntu 22.04、Ubuntu 20.04、Debian 12.5、Debian 11.9：

```
1 apt-get install -y opensc
```

安装用于智能卡身份验证的 **PAM** 模块 运行以下命令可安装 pam_krb5 和 krb5-pkinit 模块。

RHEL 9.4/9.3/9.2/8.x、Rocky Linux 9.4/9.3/9.2/8.x：

```
1 yum install krb5-pkinit
```

Ubuntu 22.04、Ubuntu 20.04：

```
1 apt-get install libpam-krb5 krb5-pkinit
```

Debian 12.5、Debian 11.9：

```
1 apt-get install -y libpam-krb5 krb5-pkinit
```

pam_krb5 模块是一个可插拔的身份验证模块。PAM 感知应用程序可以使用 pam_krb5 检查密码并从密钥发行中心 (KDC) 获取票证授予票证。krb5-pkinit 模块包含允许客户端使用私钥和证书从 KDC 获取初始凭据的 PKINIT 插件。

配置 **pam_krb5** 模块 pam_krb5 模块与 KDC 交互以使用智能卡中的证书获取 Kerberos 票据。要在 PAM 中启用 pam_krb5 身份验证，请运行以下命令：

```
1 authconfig --enablekrb5 --update
```

在 **/etc/krb5.conf** 配置文件中，根据实际的领域添加 PKINIT 信息。

注意：

pkinit_cert_match 选项指定客户端证书在用于尝试 PKINIT 身份验证之前必须匹配的规则。匹配规则的语法是：

[关系运算符] 组件规则…

其中，关系运算符可以是 **&&**，表示所有组件规则都必须匹配，也可以是 **||**，表示只有一个组件规则必须匹配。

下面是一个通用 krb5.conf 文件的示例：

```

1 EXAMPLE.COM = {
2
3
4     kdc = KDC.EXAMPLE.COM
5
6     auth_to_local = RULE:[1:$1@$0]
7
8     pkinit_anchors = FILE:<path where you install the root certificate
9                       >/certnew.pem
10
11    pkinit_kdc_hostname = KDC.EXAMPLE.COM
12
13    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
14
15    pkinit_eku_checking = kpServerAuth
16 }
```

添加 PKINIT 信息后，该配置文件将如下所示。

```

CTXDEV.LOCAL = {
    kdc = ctx-ad.ctxdev.local
    auth_to_local = RULE:[1:$1@$0]
    pkinit_kdc_hostname = ctx-ad.ctxdev.local
    pkinit_anchors = FILE:/etc/pki/CA/certs/certnew.pem
    pkinit_eku_checking = kpServerAuth
    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature
}
```

配置 PAM 身份验证 PAM 配置文件指示用于 PAM 身份验证的模块。要添加 pam_krb5 作为身份验证模块，请向 **/etc/pam.d/smartcard-auth** 文件中添加以下行：

```

auth [success=done ignore=ignore default=die] pam_krb5.so preauth_options
=X509_user_identity=PKCS11:<path to the pkcs11 driver>/opensc-pkcs11.
so
```

如果使用 SSSD，则修改后该配置文件将如下所示。


```

#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        [success=done ignore=ignore default=die] pam_krb5.so preauth_opt=X509_user_identity=PKCS11:/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so
auth        sufficient    pam_permit.so
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     [default=bad success=ok auth_err=ignore user_unknown=ignore ignore=ignore] pam_krb5.so
account     required      pam_permit.so

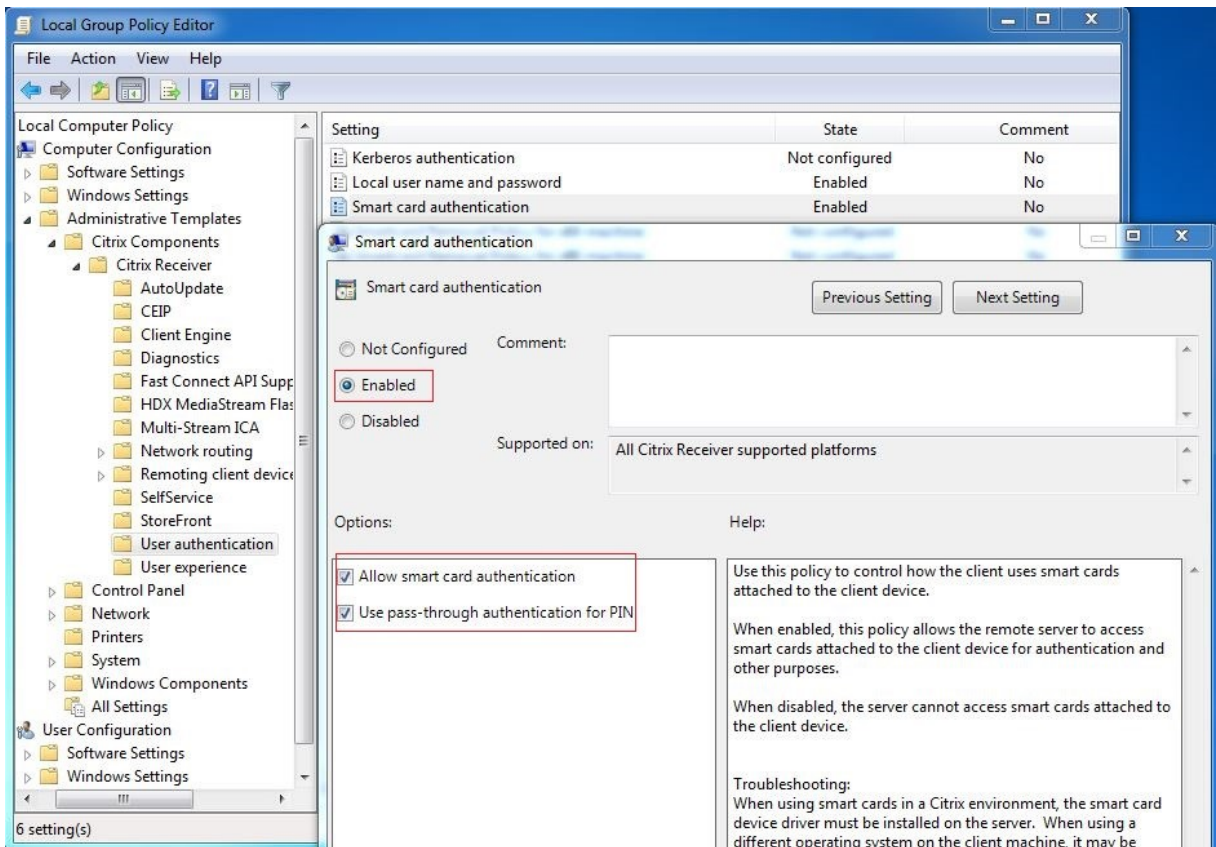
session    optional      pam_keyinit.so revoke
session    required      pam_limits.so
-session   optional      pam_systemd.so
#session   optional      pam_oddjob_mkhomedir.so umask=0077
session    optional      pam_mkhomedir.so umask=0077
session    [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session    required      pam_unix.so
session    optional      pam_sss.so
session    optional      pam_krb5.so

```

(可选) 使用智能卡进行单点登录

单点登录 (SSO) 是一项 Citrix 功能，用于实现对虚拟桌面和应用程序启动的直通身份验证。此功能降低了用户键入其 PIN 码的次数。要将 SSO 与 Linux VDA 结合使用，请配置 Citrix Workspace 应用程序。该配置与 Windows VDA 相同。有关详细信息，请参阅知识中心文章 [CTX133982](#)。

在 Citrix Workspace 应用程序中配置组策略时，请按如下所示启用智能卡身份验证。



快速智能卡登录

快速智能卡是对现有基于 HDX PC/SC 的智能卡重定向的改进。在高延迟 WAN 环境中使用智能卡时，可以提高性能。有关详细信息，请参阅[智能卡](#)。

Linux VDA 支持在以下版本的 Citrix Workspace 应用程序中使用快速智能卡：

- Citrix Receiver for Windows 4.12
- 适用于 Windows 的 Citrix Workspace 应用程序 1808 及更高版本

在客户端上启用快速智能卡登录 默认情况下，快速智能卡登录功能在 VDA 上处于启用状态，在客户端上处于禁用状态。在客户端，要启用快速智能卡登录，请将以下参数包含在关联 StoreFront 站点的 default.ica 文件中：

```
1 [WFClient]
2 SmartCardCryptographicRedirection=On
```

在客户端上禁用快速智能卡登录 要在客户端上禁用快速智能卡登录，请从关联的 StoreFront 站点的 default.ica 文件中删除 **SmartCardCryptographicRedirection** 参数。

运行 XDPing

完成上述步骤后，可以使用 Linux [XDPing](#) 工具检查 Linux VDA 环境中可能存在的常见配置问题。

使用情况

使用智能卡登录 Linux VDA

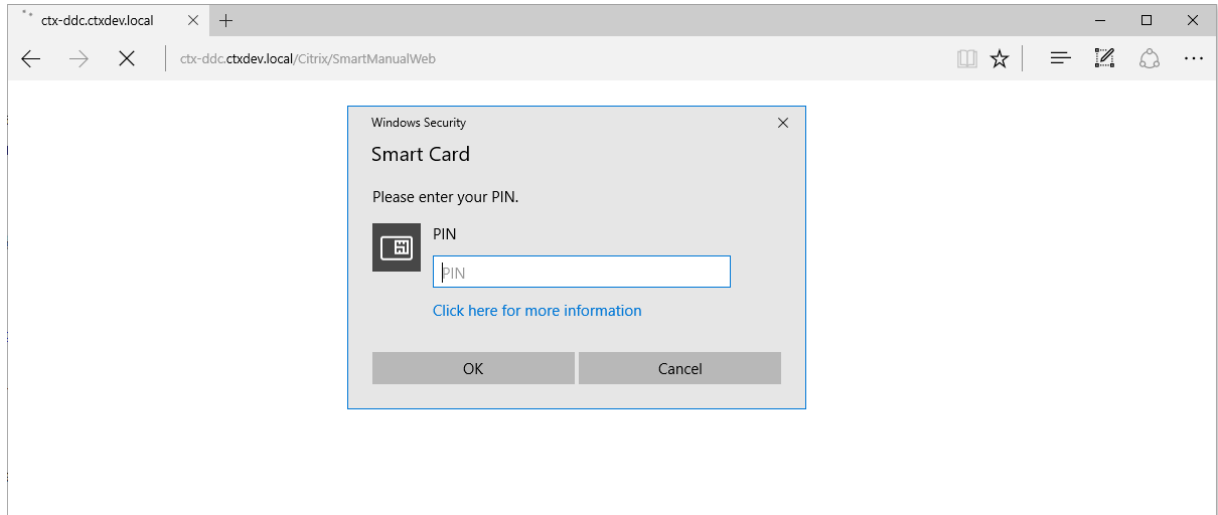
可以在 SSO 和非 SSO 场景中使用智能卡登录 Linux VDA。

- 在 SSO 场景中，您将使用缓存的智能卡证书和 PIN 码自动登录 StoreFront。在 StoreFront 中启动 Linux Virtual Desktop 会话时，PIN 码将传递到 Linux VDA 以进行智能卡身份验证。
- 在非 SSO 场景中，系统将提示您选择证书并键入 PIN 码以登录 StoreFront。



在 StoreFront 中启动 Linux Virtual Desktop 会话时，将显示 Linux VDA 登录对话框，如下所示。用户名是从智能卡中的证书中提取的，必须重新键入 PIN 码以进行登录身份验证。

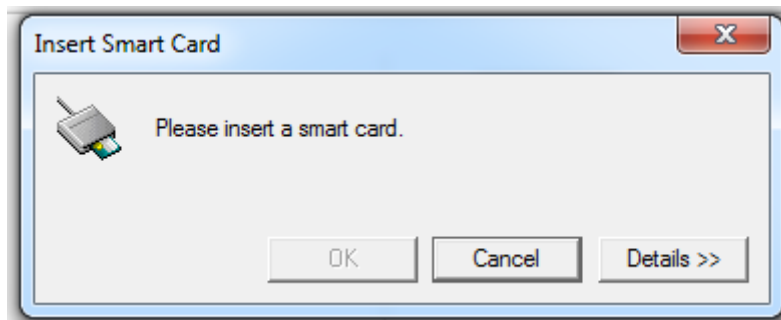
此行为与 Windows VDA 相同。



使用智能卡重新连接到会话

请务必将智能卡连接到客户端设备，才能重新连接到会话。否则，Linux VDA 端将显示一个灰色缓存窗口，并快速退出，因为重新身份验证在未连接智能卡的情况下失败。在这种情况下，系统不提供任何其他提示以提醒您连接智能卡。

但是，在 StoreFront 端，如果在您重新连接到会话时未连接智能卡，StoreFront Web 可能会按如下所示提供警报：



限制

支持有限的 **Linux** 发行版和 **AD** 集成方法

- 智能卡直通身份验证支持有限的 Linux 发行版和 AD 集成方法。请参阅以下列表：

	Winbind	SSSD	Centrify	Quest
Debian 12.5/11.9	是	是	是	否
RHEL 9.4/9.3/9.2	是	是	否	是
RHEL 8.10/8.9/8.8	是	是	是	是
Rocky Linux 9.4/9.3/9.2	是	是	否	否
Rocky Linux 8.10/8.9/8.8	是	是	否	否
Ubuntu 22.04/20.04	是	是	是	是

智能卡移除设置

通过编辑 Linux VDA 上的以下注册表项，您可以决定在会话期间将已登录用户的智能卡从智能卡读卡器中移除时将发生什么：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LocalPolicies\SecurityOptions

值名称	类型	默认	说明
SCardRemoveOption	REG_DWORD	0x00000000	此设置决定了在会话期间将已登录用户的智能卡从智能卡读卡器中移除时将发生什么。选项如下所示。 0 不采取任何行动 1 强制注销：在移除智能卡后自动注销用户。 2 断开会话连接：移除智能卡后，将断开会话连接但不注销用户。用户稍后可以重新插入智能卡以重新连接会话。

值名称	类型	默认	说明
SCardRemoveActionDelaySecs	REG_DWORD	15	如果将移除选项设置为强制注销或断开会话连接，则可以进一步指定在采取行动之前延迟几秒钟。使用此设置时，移除智能卡后，用户会话中会出现一个消息框，指明该会话将在指定的秒数后被强制注销/断开连接。如果您在该时间之前重新插入智能卡，会话将不间断地继续进行

限制 Linux VDA 一次仅支持一个智能卡读卡器。

支持其他智能卡和 **PKCS#11** 库

Citrix 提供通用智能卡重定向解决方案。虽然我们的支持列表中仅列出了 OpenSC 智能卡，但是您可以尝试使用其他智能卡和 PKCS#11 库。要切换到特定的智能卡或 PKCS#11 库，请执行以下操作：

1. 使用 PKCS#11 库替换所有 `opensc-pkcs11.so` 实例。
2. 要将 PKCS#11 库的路径设置为注册表，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v "
PKCS11LibPath" -d "PATH"
```

其中 **PATH** 指向您的 PKCS#11 库，例如 `/usr/lib64/pkcs11/opensc-pkcs11.so`

3. 在客户端上禁用快速智能卡登录。

未经身份验证的（匿名）用户进行访问

July 15, 2024

您可以允许用户访问应用程序和桌面，而不需要向 StoreFront 或 Citrix Workspace 应用程序提供凭据。要向未经身份验证的用户授予访问权限，您必须拥有未经身份验证的 StoreFront 应用商店，并为交付组中未经身份验证的用户启用访问权限。

注意：

只有加入了域的 VDA 才支持未经身份验证的用户进行访问。

未经身份验证的用户不支持会话预启动。会话预启动在适用于 Android 的 Citrix Workspace 应用程序上也不受支持。

创建未经身份验证的 **StoreFront** 应用商店

1. 在 Citrix StoreFront 管理控制台的左侧窗格中选择应用商店节点，然后在操作窗格中单击创建应用商店。
2. 在应用商店名称页面上，指定应用商店的名称，选择仅允许未经身份验证的 (匿名) 用户访问此应用商店，然后单击下一步。

有关详细信息，请参阅[创建未经身份验证的应用商店](#)。

在交付组中为未经身份验证的用户启用访问权限

交付组是从一个或多个计算机目录中选择的计算机的集合。指定可以使用交付组中的应用程序和桌面的用户时，您可以向未经身份验证的用户授予访问权限。有关详细信息，请参阅[创建交付组](#)。

为未经身份验证的用户会话设置空闲超时时间

未经身份验证的用户会话的默认空闲超时为 10 分钟，当客户端断开连接时，这些会话将自动注销。可以通过注册表设置 **AnonymousUserIdleTime** 配置自定义空闲超时时间。例如，要将自定义空闲超时时间设置为五分钟，请运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix" -v AnonymousUserIdleTime -d 0
  x00000005
```

设置未经身份验证的用户的最大数量

要设置未经身份验证的用户的最大数量，请使用注册表项 **MaxAnonymousUserNumber**。此设置限制单个 Linux VDA 上同时运行的未经身份验证的用户会话数。可以使用 **ctxreg** 工具配置此注册表设置。例如，要将值设置为 32，请运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix" -v MaxAnonymousUserNumber -d 0
  x00000020
```

重要:

限制未经身份验证的用户会话数。如果同时启动太多会话，VDA 可能会出现問題，其中包括耗尽可用内存。

故障排除

配置未经身份验证的用户会话时，请注意以下事项：

- 无法登录到未经身份验证的用户会话。

确认注册表是否已更新包含了以下内容（设置为 0）：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg read -k "HKLM\System\CurrentControlSet
   \Control\Citrix" -v MaxAnonymousUserNumber
```

确认 **ncsd** 服务是否正在运行，且已配置为启用 **passwd** 缓存：

```
1 ps uax | grep nscd
2 cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
```

如果已启用，请将 **Passwd** 缓存变量设置为 **no**，然后重新启动 **ncsd** 服务。更改此配置后您可能需要重新安装 Linux VDA。

- 使用 **KDE** 时未经身份验证的用户会话中显示锁屏按钮。

默认情况下未经身份验证的用户会话中禁用锁屏按钮和菜单。但是，它们仍可显示在 KDE 中。在 KDE 中，要对特定用户禁用锁屏按钮和菜单，请将以下行添加到配置文件 **\$HOME/.kde/share/config/kdeglobals** 中。例如：

```
1 [KDE Action Restrictions]
2 action/lock_screen=false
```

但是，如果在全局范围的 **kdeglobals** 文件（如 **/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals**）中将 **KDE Action Restrictions** 参数配置为不可变，用户配置将不起作用。

要解决此问题，请修改系统范围的 **kdeglobals** 文件以删除 **[KDE Action Restrictions]** 部分中的 **[\$i]** 标记，或直接使用系统范围的配置来禁用锁屏按钮和菜单。有关 KDE 配置的详细信息，请参阅 [KDE System Administration/Kiosk/Keys](#) 页面。

File

July 15, 2024

本部分内容包含以下主题：

- [文件复制和粘贴](#)
- [文件传输](#)

文件复制和粘贴

July 15, 2024

用户可以使用右键单击菜单或键盘快捷键在会话与本地客户端之间复制和粘贴文件。此功能需要 Citrix Virtual Apps and Desktops 2006 或更高版本以及适用于 Windows 的 Citrix Workspace 应用程序 1903 或更高版本。

要成功复制和粘贴文件，请确保：

- 文件的最大数量不超过 20。
- 最大文件大小不超过 200 MB。
- Nautilus 文件管理器在安装了 Linux VDA 的计算机上可用。
- 文件名仅包含 ASCII 字符，不包含特殊字符。

支持的 **Linux** 发行版

文件复制和粘贴功能适用于 Linux VDA 支持的所有 Linux 发行版。

相关策略

以下剪贴板策略与配置功能相关。有关剪贴板策略的详细信息，请参阅[策略支持列表](#)。

- 客户端剪贴板重定向
- 剪贴板选择更新模式
- 主选定内容更新模式

注意：

要禁用文件复制和粘贴功能，请在 Citrix Studio 中将客户端剪贴板重定向策略设置为禁止。

限制

- 不支持剪切。剪切文件请求被视为复制操作。
- 不支持拖放。
- 不支持复制目录。
- 文件复制和粘贴必须按顺序执行。只有成功复制和粘贴上一个文件后，才能复制下一个文件。

文件传输

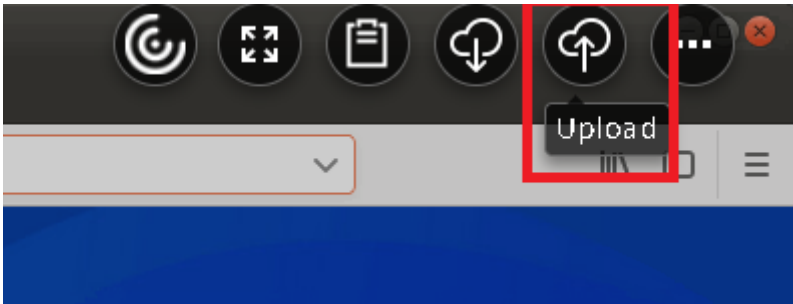
July 15, 2024

支持在 Linux VDA 和客户端设备之间进行文件传输。当客户端设备运行支持 HTML5 沙盒属性的 Web 浏览器时，此功能可用。HTML5 沙盒属性允许用户使用适用于 HTML5 和 Chrome 的 Citrix Workspace 应用程序访问虚拟桌面和应用程序。

注意：

文件传输对适用于 HTML5 和 Chrome 的 Citrix Workspace 应用程序可用。

在已发布的应用程序和桌面会话中，文件传输允许在 Linux VDA 与客户端设备之间上传和下载文件。要将文件从客户端设备上载到 Linux VDA，请单击 Citrix Workspace 应用程序工具栏上的上传图标，然后从文件对话框中选择所需的文件。要将文件从 Linux VDA 下载到客户端设备，请单击下载图标。可以在上传或下载过程中添加文件。一次最多可以传输 10 个文件。



注意：

要在 Linux VDA 与客户端设备之间上传和下载文件，请启用 Citrix Workspace 应用程序的工具栏。可以使用允许拖放文件的 Citrix Workspace 应用程序版本。

自动下载是文件传输的增强功能。下载或移动到 VDA 上的保存到我的设备目录的文件将自动传输到客户端设备。

注意：

自动下载要求将 **File transfer for Citrix Workspace app for ChromeOS/HTML5**（适用于 ChromeOS/HTML5 的 Citrix Workspace 应用程序的文件传输）和 **Download file for Citrix Workspace app for ChromeOS/HTML5**（适用于 ChromeOS/HTML5 的 Citrix Workspace 应用程序的文件下载）策略设置为允许。

下面是自动下载的一些用例：

- 下载文件以保存到我的设备

在已发布的桌面和 Web 浏览器应用程序会话中，从 Web 站点下载的文件可以保存到 VDA 上的保存到我的设备目录中，以便自动传输到客户端设备。要实现自动下载，请将会话中 Web 浏览器的默认下载目录设置为保存到

我的设备，并在运行适用于 HTML5 或 Chrome 的 Citrix Workspace 应用程序的 Web 浏览器中设置本地下载目录。

- 将文件移动或复制到保存到我的设备

在已发布的桌面会话中，选择目标文件，然后将其移动或复制到保存到我的设备目录中，以便在客户端设备上可用。

文件传输策略

默认启用文件传输。使用 Citrix Studio 可更改这些策略（位于用户设置 > ICA\文件重定向下）。使用文件传输策略时，请注意以下事项：

File transfer for Citrix Workspace app for ChromeOS/HTML5（适用于 ChromeOS/HTML5 的 Citrix Workspace 应用程序的文件传输）：允许或阻止最终用户在 Citrix Virtual Apps and Desktops 会话或 Citrix DaaS 会话与最终用户的设备之间传输文件。

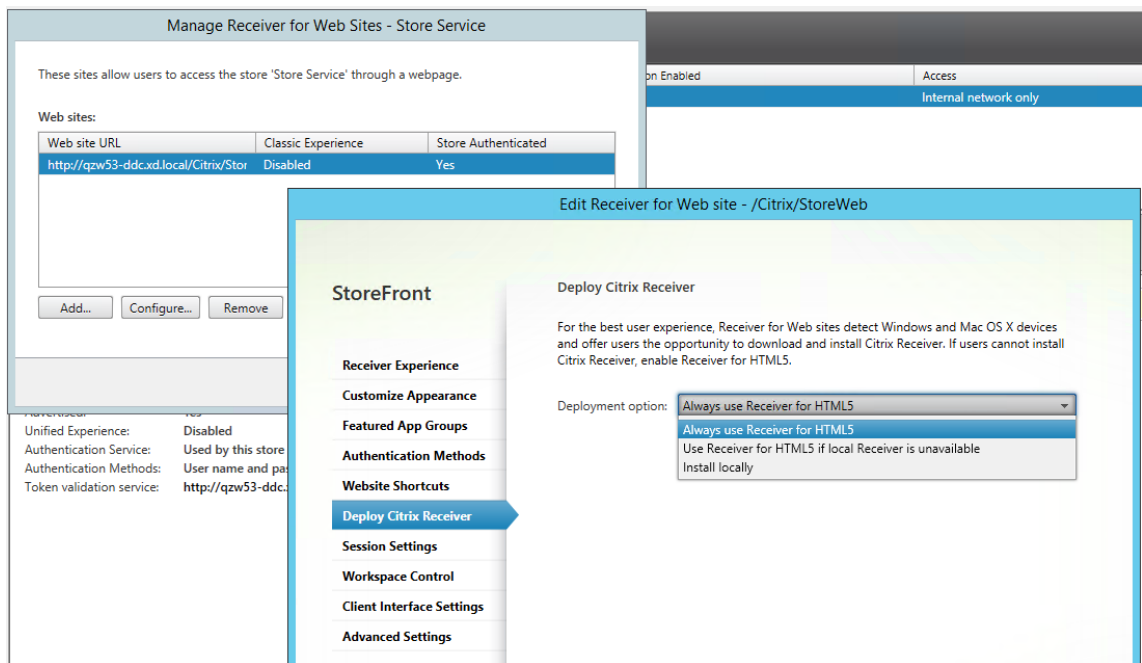
Upload file for Citrix Workspace app for ChromeOS/HTML5（适用于 ChromeOS/HTML5 的 Citrix Workspace 应用程序的文件上传）：允许或阻止用户上传文件。文件可以从用户的设备上载到 Citrix Virtual Apps and Desktops 会话或 Citrix DaaS 会话。

Download file for Citrix Workspace app for ChromeOS/HTML5（适用于 ChromeOS/HTML5 的 Citrix Workspace 应用程序的文件下载）：允许或阻止用户下载文件。文件可以从 Citrix Virtual Apps and Desktops 会话或 Citrix DaaS 会话下载到用户的设备。

使用情况

要通过适用于 **HTML5** 的 **Citrix Workspace** 应用程序使用文件传输功能，请执行以下操作：

1. 在 Citrix Studio 中，将 **WebSocket** 连接策略设置为允许。
2. 在 Citrix Studio 中，通过上述文件传输策略启用文件传输功能。
3. 在 Citrix StoreFront 管理控制台中，单击应用商店，选择管理 **Receiver for Web** 站点节点，然后通过选择始终使用 **Receiver for HTML5** 选项来启用 Citrix Receiver for HTML5。



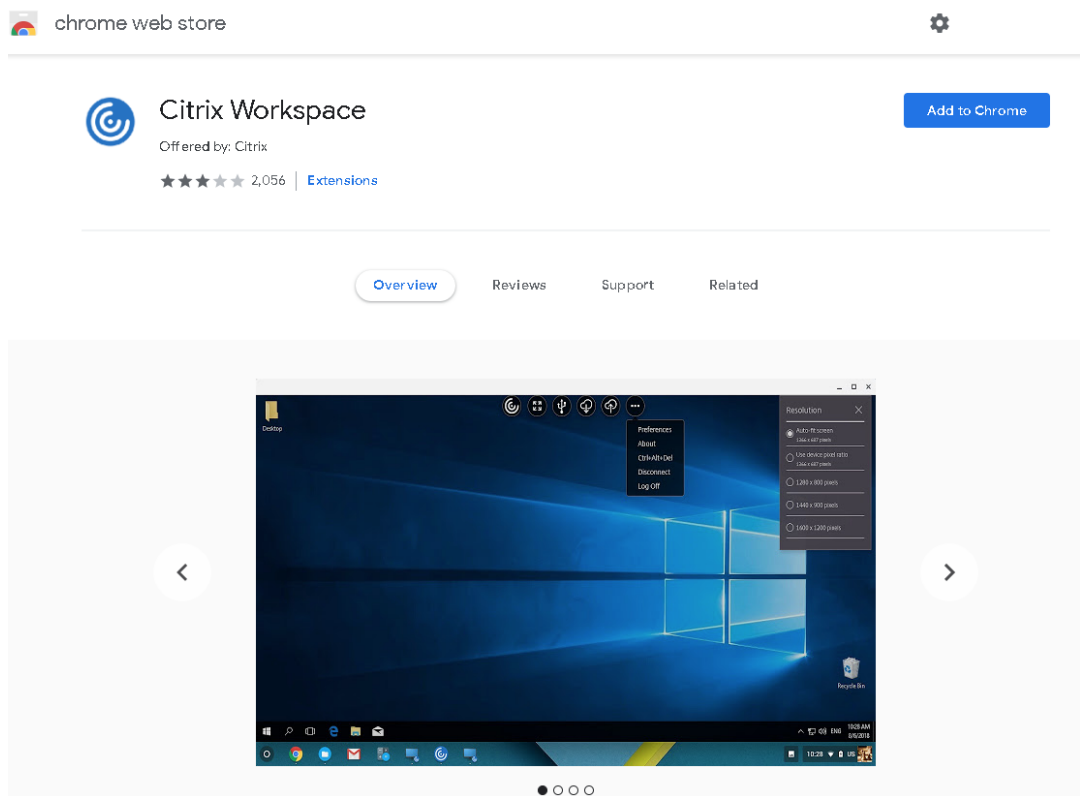
4. 启动虚拟桌面或 Web 浏览器应用程序会话。在 Linux VDA 与客户端设备之间执行一个或多个文件传输。

要通过适用于 **Chrome** 的 **Citrix Workspace** 应用程序使用文件传输功能，请执行以下操作：

1. 通过上述文件传输策略启用文件传输功能。
2. 从 Chrome 网上应用店中获取 Citrix Workspace 应用程序。

如果您已将适用于 Chrome 的 Citrix Workspace 应用程序添加到“Chrome 应用程序”页面，请跳过此步骤。

- a) 在 Google Chrome 的搜索框中键入适用于 **Chrome** 的 **Citrix Workspace**。单击搜索图标。
- b) 在搜索结果中，单击指向 Chrome 网上应用店（提供 Citrix Workspace 应用程序）的 URL。



- c) 单击添加到 **Chrome** 以将 Citrix Workspace 应用程序添加到 Google Chrome 中。
3. 在“Chrome 应用程序”页面上单击“适用于 Chrome 的 Citrix Workspace 应用程序”。
4. 键入要连接的 StoreFront 应用商店的 URL。
如果您之前键入了该 URL，请跳过此步骤。
5. 启动虚拟桌面或应用程序会话。在 Linux VDA 与客户端设备之间执行一个或多个文件传输。

图形

July 15, 2024

本部分内容包含以下主题：

- [自动 DPI 缩放](#)
- [客户端电池状态显示](#)
- [图形配置和微调](#)
- [HDX 屏幕共享](#)
- [未虚拟化的 GPU](#)

- [会话水印](#)
- [Thinwire 渐进式显示](#)

自动 DPI 缩放

August 20, 2024

Linux VDA 支持自动 DPI 缩放。当用户打开虚拟桌面或应用程序会话时，会话中的 DPI 值会自动更改以匹配客户端上的 DPI 设置。

下面是与此功能相关的注意事项：

- 该功能要求您为 Citrix Workspace 启用 DPI 匹配。对于适用于 Windows 的 Citrix Workspace 应用程序，请确保选中否，使用本机分辨率选项。有关为适用于 Windows 的 Citrix Workspace 应用程序配置 DPI 缩放的详细信息，请参阅 [DPI 缩放](#)。
- 要使用该功能在多显示器方案中起作用，必须使用相同的 DPI 设置对每台显示器进行配置。不支持混合 DPI 方案。如果使用不同的 DPI 设置配置显示器，Linux VDA 会对所有屏幕应用最小 DPI 值。
- 该功能已对 MATE、GNOME、GNOME Classic 和 KDE 启用。使用 KDE 或 MATE 时，请注意以下事项：
 - 对于在 Mate 桌面环境中运行的 KDE 虚拟桌面：
 - * 我们建议使用 KDE Plasma 5 或更高版本。
 - * 在会话运行时更改客户端上的 DPI 设置要求用户注销并重新登录。
 - 对于在 Mate 桌面环境中运行的 MATE 虚拟桌面：
 - * 仅支持缩放比例 1 和 2。
 - * 在会话运行时更改客户端上的 DPI 设置要求用户注销并重新登录。
- 虚拟会话中的 DPI 值会根据客户端上的 DPI 设置自动更改。目前，该功能仅支持整数类型的缩放比例，例如 100% 和 200%。如果在客户端配置的缩放比例类型为分数，虚拟会话 DPI 将根据下表更改为整数缩放比例。示例：如果缩放比例为 125%，DPI 值将更改为 100%。

客户端缩放比例	远程会话 DPI	显示缩放
小于或等于 174%	96 (1 x 96)	100%
175%–274%	192 (2 x 96)	200%
275%–399%	288 (3 x 96)	300%
大于或等于 400%	384 (4 x 96)	400%

注意：

Linux VDA 支持高达 200% 的 MATE 桌面扩展。

客户端电池状态显示

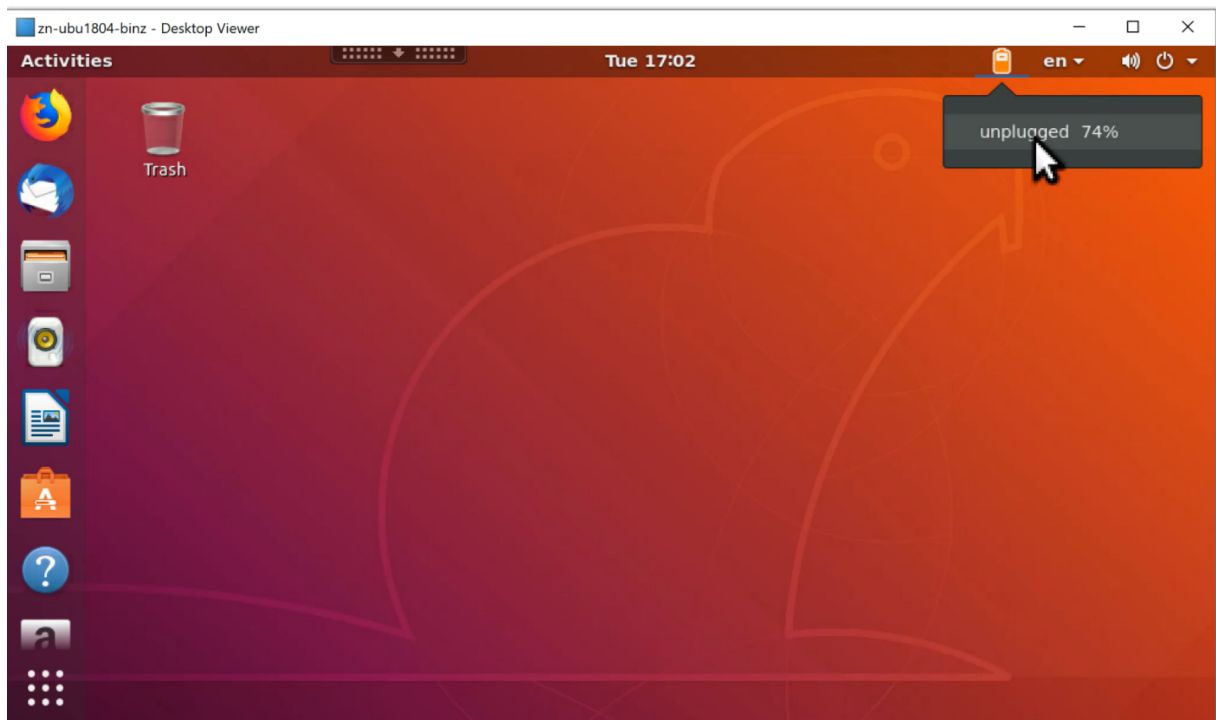
July 15, 2024

Linux VDA 可以重定向并显示虚拟桌面中客户端设备的电池状态。默认情况下，此功能处于启用状态，适用于以下版本的 Citrix Workspace 应用程序：








- 适用于 iOS 的 Citrix Workspace 应用程序
- 适用于 Linux 的 Citrix Workspace 应用程序
- 适用于 Mac 的 Citrix Workspace 应用程序（不支持版本 2204.1）
- 适用于 Windows 的 Citrix Workspace 应用程序（不支持版本 2204.1）

概述

当用户打开虚拟桌面时，他们可以在 Linux 系统托盘中看到电池图标。电池图标表示其客户端设备的电池状态。要查看剩余电池寿命的百分比，请单击电池图标。有关示例，请参见以下屏幕截图：



不同的电池图标表示不同的电池状态。有关概述，请参见下表：

电池图标	充电状态	剩余电池寿命级别	剩余电池寿命的百分比
	正在充电, 用 “+” 符号表示	高, 用绿色表示	=80%
	正在充电, 用 “+” 符号表示	中, 用琥珀色表示	= 20% 和 < 80%
	正在充电, 用 “+” 符号表示	低, 用红色表示	< 20%
	未充电, 用 “-” 符号表示	高, 用绿色表示	=80%
	未充电, 用 “-” 符号表示	中, 用琥珀色表示	= 20% 和 < 80%
	未充电, 用 “-” 符号表示	低, 用红色表示	< 20%
	未知	未知	未知

配置

默认情况下, 客户端电池状态显示处于启用状态。

要禁用该功能, 请运行以下命令:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc" -v "Enabled" -d "0x00000000"
```

要启用此功能, 请运行以下命令:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc" -v "Enabled" -d "0x00000001"
```

注意:

上述命令会影响软键盘功能, 该功能与客户端电池状态显示共享 Mobile Receiver 虚拟通道 (MRVC)。

根据您的发行版, 完成额外的步骤以启用该功能。有关详细信息, 请参阅[启用系统托盘](#)部分。

图形配置和微调

August 20, 2024

本文介绍了 Linux VDA 图形配置和微调。

有关详细信息，请参阅[系统要求](#)和[安装概述](#)部分。

配置

针对 **3D** 图形工作负载优化

此设置配置最适合图形密集型工作负载的相应默认值。可为工作负载集中于图形密集型应用程序的用户启用此设置。仅在会话可以使用 GPU 的情况下应用此策略。显式覆盖由此策略设置的默认设置的其他任何设置具有更高的优先级。

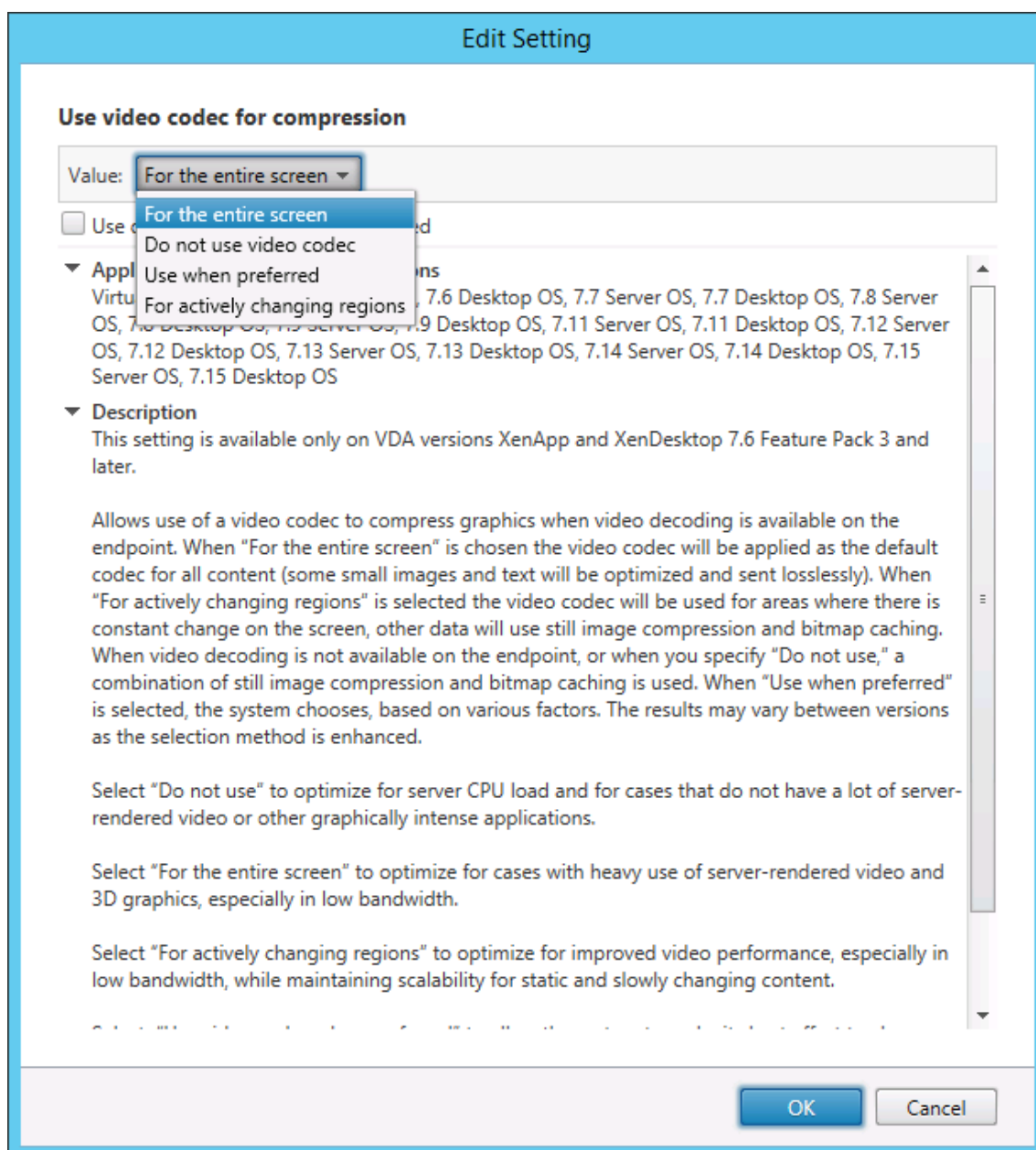
默认情况下，禁止优化 **3D** 图形工作负载。

视频编解码器进行压缩

Thinwire 是 Linux VDA 中使用的显示内容远程处理技术。该技术允许一台计算机上生成的图形传输（通常跨网络）到另一台计算机上进行显示。

使用[视频编解码器进行压缩](#)图形策略设置默认图形模式，并针对不同的用例提供以下选项：

- 偏好时使用。这是默认设置。无需执行其他配置。它可确保为所有 Citrix 连接选择 Thinwire，且 Thinwire 已针对典型桌面工作负载在可扩展性、带宽和卓越图像质量方面经过优化。
- 针对整个屏幕。为 Thinwire 提供全屏 H.264 或 H.265，以针对改进用户体验和带宽使用情况进行优化，尤其是在大量使用 3D 图形的情况下。选择了针对整个屏幕，或者选择了偏好时使用并启用了[针对 3D 图形工作负载优化](#)时，支持[会话水印](#)。
- 针对主动变化的区域。Thinwire 中的自适应显示技术识别移动图像（视频、动态 3D）。它仅在图像正在移动的屏幕部分中使用 AV1、H.265 或 H.264。选择性使用 AV1、H.265 或 H.264 视频编解码器支持 HDX Thinwire 检测屏幕的频繁更新部分并对其进行编码。对于屏幕的其余部分（包括文本和摄影图片），仍继续使用图像压缩（JPEG、RLE）和位图缓存。用户可获得以下益处：占用更低的带宽、提高视频内容质量以及在其他位置获得无损文本或高质量图像。当视觉质量策略设置为始终无损或无损构建时，不支持选择性使用 AV1 和 H.265。



一些其他策略设置（包括以下视觉显示策略设置）可以用于对显示远程处理技术的性能进行完善：

- [简单图形的首选颜色深度](#)
- [目标帧速率](#)
- [视觉质量](#)

AV1/H.265/H.264 硬件编码

使用视频编解码器的硬件编码策略允许使用 GPU 硬件加速（如果可用）通过视频编解码器压缩屏幕元素。GPU 硬件加速可优化硬件资源利用率，极大地提高了每秒帧数 (FPS) 的性能。

GPU 硬件加速涵盖使用视频编解码器进行压缩策略设置的所有图形模式：

- 偏好时使用
- 针对整个屏幕
- 针对主动变化的区域

要启用硬件视频压缩，请完成以下步骤：

1. 将使用视频编解码器的硬件编码策略设置为已启用。
2. 将使用视频编解码器进行压缩设置为偏好时使用、针对整个屏幕或针对主动变化的区域。请确保未将其设置为不使用视频编解码器。

VDA 和 Citrix Workspace 应用程序必须支持和启用 AV1 或 H.265 视频编解码器才能使用。在编解码器协商期间，AV1 优先于 H.265 和 H.264。不支持 AV1 时，将协商使用 H.265。如果不支持 AV1 和 H.265，会话将回退到使用 H.264 视频编解码器。如果 GPU 硬件不可用，VDA 会回退到使用软件视频编解码器的基于 CPU 的编码。

AV1 硬件编码的要求

VDA

- VDA: 2311 或更高版本
- GPU: **NVIDIA Ada Lovelace** 或更高版本 (有关 NVIDIA GPU 支持的编解码器列表，请参阅 NVIDIA 文档，网址为 <https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new>)
- NVIDIA 显卡驱动程序 522.25 或更高版本 (Video Codec SDK v12.0)

客户端

- 适用于 Windows 的 Citrix Workspace 应用程序 2305 或更高版本
- 支持 AV1 解码的客户端 GPU：
 - NVIDIA Ampere 或更高版本
 - Intel 第 11 代/Arc 或更高版本
 - AMD Radeon RX 6000/Radeon Pro W6000 系列 (RDNA2) 或更高版本

H.265 硬件编码的要求

客户端

- Citrix Receiver for Windows 4.10 到 4.12
- 适用于 Windows 的 Citrix Workspace 应用程序 1808 及更高版本

要在客户端上启用 H.265 硬件编码，请参阅 [H.265 视频编码](#)。

H.265/H.264 无损压缩

H.265/H.264 无损压缩适用于通过 NVIDIA GPU 进行的 HDX 3D PRO 硬件加速。H.265 无损压缩需要适用于 Windows 的 Citrix Workspace 应用程序 2305 及更高版本。H.264 无损压缩需要以下客户端：

- 适用于 Windows 的 Citrix Workspace 应用程序 2303 及更高版本
- 配备 Apple M1 芯片的适用于 Mac 的 Citrix Workspace 应用程序 2301 及更高版本

要启用 H.265/H.264 无损压缩，请完成以下步骤：

1. 将使用视频编解码器的硬件编码策略设置为已启用。
2. 将[使用视频编解码器进行压缩](#)策略设置为针对整个屏幕。
3. 将视觉质量策略设置为 **Always losses**（始终有损）或无损构建。

允许视觉无损压缩

允许视觉无损压缩策略允许对图形使用视觉无损压缩，而不是真正的无损压缩。相比于真正无损的压缩，视觉无损功能可提高性能，但会产生视觉上不易察觉的轻微损失。此设置可更改视觉质量设置的值的使用方式。

默认情况下，允许视觉无损压缩策略处于禁用状态。要启用视觉无损压缩，请将允许视觉无损压缩设置为已启用，将视觉质量策略设置为无损构建。

如果将使用视频编解码器进行压缩策略设置为不使用视频编解码器，则视觉无损压缩将应用到静态图像编码。如果将使用视频编解码器进行压缩策略设置为不使用视频编解码器以外的图形模式，则视觉无损压缩将应用到 H.264 编码。

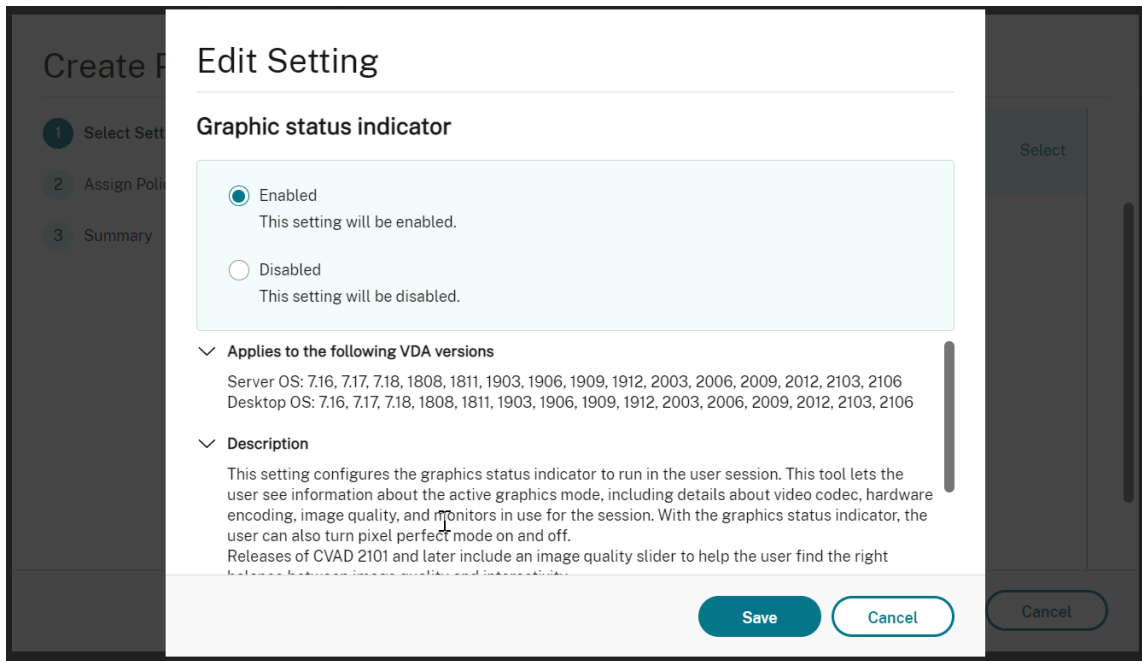
有关视觉质量和使用视频编解码器进行压缩策略设置的详细信息，请参阅[视觉显示策略设置](#)和[图形策略设置](#)。

图形质量滑块

我们在虚拟 Linux 会话中运行的图形状态指示器工具中包含了图形质量滑块。该滑块有助于在图像质量与交互性之间找到正确的平衡。

要使用滑块，请完成以下步骤：

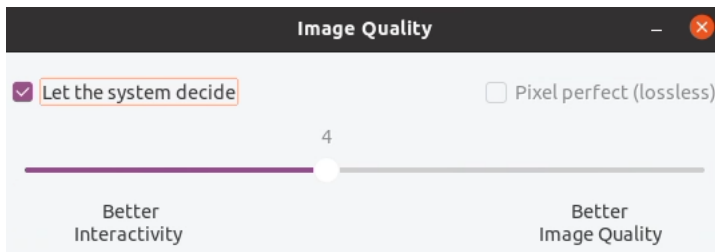
1. 在 Citrix Studio 中启用图形状态指示器策略。



2. 打开终端并运行 `ctxslider` 命令。此时将显示滑块 UI。

注意：

- 如果已将视觉质量策略设置为始终无损或无损构建，则不会显示滑块 UI。
- 您可以从终端和系统托盘启动滑块用户界面。



现在可以提供以下选项：

- 要更改图像质量，请移动滑块。滑块支持的范围为 0-9。
- 要使用系统定义的设置，请选择让系统决定。
- 要切换到无损模式，请选择像素完美。

根据带宽估算值调整平均位速率

通过根据带宽估算值调整平均位速率，Citrix 增强了 HDX 3D Pro 硬件编码。

使用 HDX 3D Pro 硬件编码时，VDA 可以间歇性地估计网络的带宽并相应地调整已编码的帧的位速率。这一新增功能提供了一种平衡清晰度和流畅度的机制。

默认情况下启用此功能。要禁用此功能，请运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  DisableReconfigureEncoder" -d "0x00000001" --force
```

除了使用此功能之外，您还可以运行以下命令来调整清晰度和流畅度。**AverageBitRatePercent** 和 **MaxBitRatePercent** 参数将设置带宽使用情况的百分比。设置的值越高，图形就越清晰，但流畅度越低。建议的设置范围为 50 到 100。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  AverageBitRatePercent" -d "90" --force
2
3 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  MaxBitRatePercent" -d "100" --force
```

在调整平均位速率时，如果您的屏幕画面保持不变，则最近的帧将处于低质量状态，因为未发送任何新的帧。通过重新配置并即时发送最高质量的最新帧，锐化支持可以解决此问题。

有关 Linux VDA Thinwire 支持的策略的完整列表，请参阅[策略支持列表](#)。

有关 Linux VDA 上的多监视器支持配置的信息，请参阅 [CTX220128](#)。

并行处理

Thinwire 可以通过并行处理某些任务来提高每秒帧数 (FPS)，但总体 CPU 占用量的开销略高。默认情况下，此功能处于禁用状态。要启用此功能，请在您的 VDA 上运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  ParallelProcessing" -d "0x00000001" --force
```

故障排除

检查正在使用哪种图形模式

运行以下命令来检查正在使用哪种图形模式 (**0** 表示 TW+。**1** 表示全屏视频编解码器)：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
```

结果类似于：

```
create -k "HKLM\System\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "GraphicsMode"-d "0x00000000"--force
```

验证 **AV1** 是否正在使用中

注意：

要验证当前会话正在使用哪个视频编解码器，请运行下面提供的命令或者通过[系统托盘](#)检查显卡状态。

请运行以下命令来验证是否正在使用 AV1 (**0** 表示未在使用。**1** 表示正在使用)：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep AV1
```

例如，结果可能类似于：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"  
-v "AV1"-d "0x00000000"--force
```

验证是否正在使用 **H.265**

运行以下命令来验证是否正在使用全屏 H.265 (**0** 表示未使用。**1** 表示正在使用)：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
```

例如，结果可能类似于：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"  
-v "H265"-d "0x00000000"--force
```

验证是否正在使用 **H.264**

运行以下命令来验证是否正在使用 H.264 (**0** 表示未在使用。**1** 表示正在使用)：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
```

例如，结果可能类似于：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"  
-v "H264"-d "0x00000000"--force
```

检查正在使用哪种 **YUV** 编码方案

运行以下命令来检查正在使用哪种 YUV 编码方案 (**0** 表示 YUV420；**1** 表示 YUV422；**2** 表示 YUV444)：

注意：

仅当正在使用视频编解码器时，**YUVFormat** 的值才有意义。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
```

例如，结果可能类似于：

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"  
-v "YUVFormat"-d "0x00000000"--force
```

验证是否正在使用 **YUV444** 软件编码

请运行以下命令以验证是否正在使用 YUV444 软件编码：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
```

当 YUV444 处于运行状态时，结果类似于：

```
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
```

验证 **HDX 3D Pro** 是否已启用

请运行以下命令以验证 HDX 3D Pro 是否已启用：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep ProductEdition  
2  
3 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep StackSessionMode  
4  
5 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep 3DPro
```

启用了 HDX 3D Pro 时，结果类似于：

```
create -k "HKLM\Software\Citrix\VirtualDesktopAgent\State"-t "REG_SZ"  
-v "ProductEdition"-d "<PLT or ENT>"--force  
create -k "HKLM\System\CurrentControlSet\Control\Citrix\WinStations\  
tcp"-t "REG_DWORD"-v "StackSessionMode"-d "0x00000000"--force  
create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"  
-v "3DPro"-d "0x00000000"--force
```

要验证 HDX 3D Pro 所需的 NVIDIA 库是否已加载，请在 Linux VDA 上运行 **nvidia-smi** 命令。结果类似于：

```

1 Tue Apr 12 10:42:03 2016
2 +-----+
3 | NVIDIA-SMI 361.28      Driver Version: 361.28      |
4 |-----+-----+
5 | GPU Name          Persistence-M| Bus-Id        Disp.A | Volatile
6 |   Uncorr. ECC     |              |            |     |
7 | Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util
8 | Compute M.       |              |            |     |
9 |=====+=====+=====+=====+
10 |    0  GRID K1              Off | 0000:00:05.0   Off |
11 | N/A   42C    P0             N/A | 14W / 31W     | 207MiB / 4095MiB |    8%
12 | Default |
13 |-----+-----+
14 | Processes:
15 |   Memory |
16 | GPU      PID  Type  Process name
17 | Usage    |
18 |=====+=====+
19 |    0      2164  C+G   /usr/local/bin/ctxgfx
20 | 106MiB |
21 |    0      2187    G     Xorg
22 | 85MiB |
23 +-----+

```

验证是否正在对 **3D Pro** 使用硬件编码

运行以下命令（**0** 表示未使用。**1** 表示正在使用）：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
```

使用 3D Pro 时，结果类似于：

```
create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics"-t "REG_DWORD"
-v "HardwareEncoding"-d "0x00000001"--force
```

确认 **NVIDIA GRID** 图形驱动程序是否已正确安装

要确认 NVIDIA GRID 图形驱动程序是否已正确安装，请运行 **nvidia-smi**。结果类似于：

```
1 +-----+
```



```

2 | NVIDIA-SMI 352.70      Driver Version: 352.70      |
3 | -----+-----+-----+-----+
4 | GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile
   Uncorr. ECC |
5 | Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util
   Compute M. |
6 | =====+=====+=====+=====+
7 |    0   Tesla M60                Off  | 0000:00:05.0    Off  |
   Off  |
8 | N/A   20C    P0      37W / 150W |      19MiB /  8191MiB |      0%
   Default |
9 | -----+-----+-----+-----+
10 |
11 | -----+-----+-----+-----+
12 | Processes:                                                       GPU
   Memory |
13 | GPU       PID  Type  Process name
   Usage   |
14 | =====+=====+=====+=====+
15 | No running processes found
16 | -----+-----+-----+-----+

```

为显卡设置正确的配置：

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

HDX 3D Pro 多监视器重绘问题

如果在非主监视器屏幕上发生重绘问题，请检查 NVIDIA GRID 许可证是否可用。

检查 **Xorg** 错误日志

Xorg 的日志文件命名类似于 **Xorg.{DISPLAY}.log**，位于 **/var/log/** 文件夹中。

已知问题及限制

对于 **vGPU**，**XenServer**（以前称为 **Citrix Hypervisor**）本地控制台显示 **ICA** 桌面会话屏幕

解决方法：通过运行以下命令禁用 VM 的本地 VGA 控制台：

```
1 [root@xenserver ~]# xe vgpu-param-set uuid=vgpu-uuid extra_args=
  disable_vnc=1
```

登录时 **Gnome 3** 桌面显示很慢

这是 Gnome 3 桌面会话启动的限制。

在调整 **Citrix Workspace** 应用程序窗口大小时，有些 **OpenGL/WebGL** 应用程序无法很好地呈现

调整 Citrix Workspace 应用程序窗口大小会改变屏幕分辨率。NVIDIA 专用驱动程序会更改某些内部状态，并可能要求应用程序做出相应的响应。例如，WebGL 库元素 **lightgl.js** 可能生成错误 **Rendering to this texture is not supported (incomplete frame buffer)**（不支持呈现为此纹理（帧缓冲区不完整））。

HDX 屏幕共享

July 15, 2024

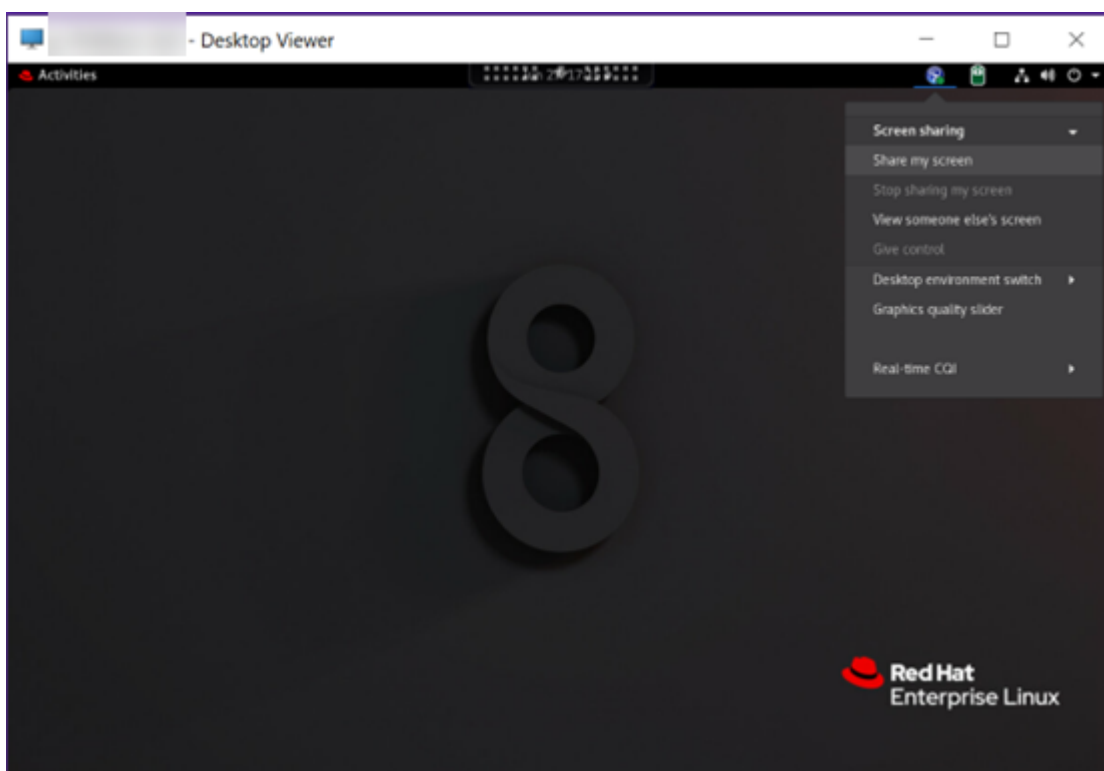
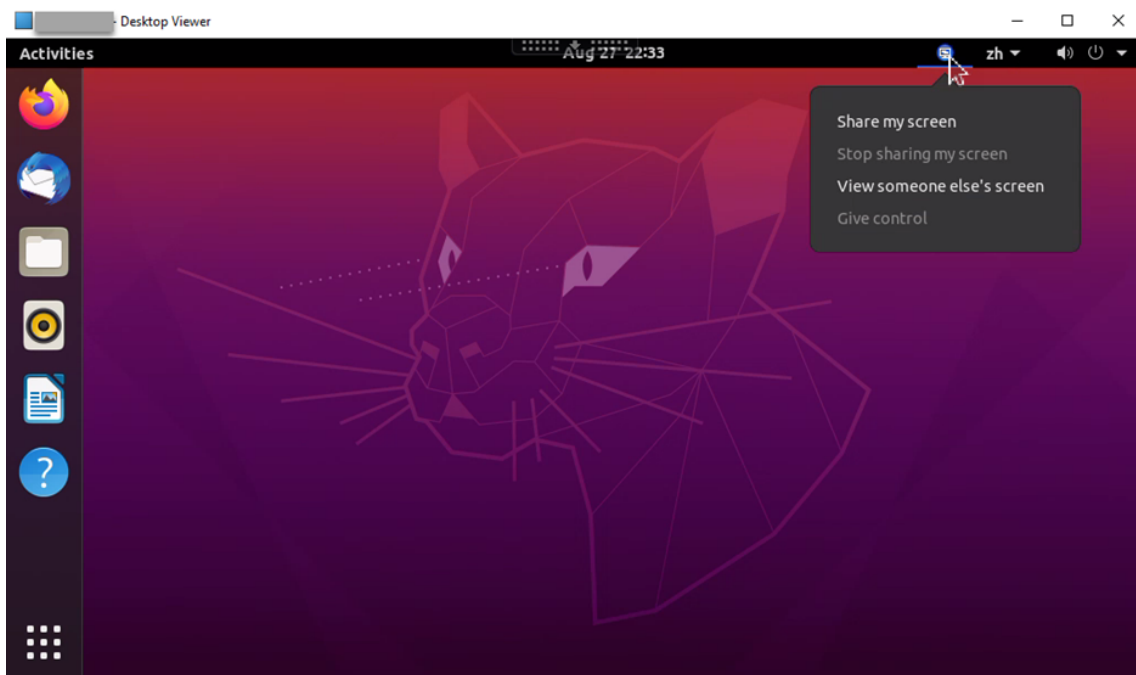
概述

Linux VDA 允许您与其他虚拟桌面上的会话用户共享自己的虚拟桌面屏幕。

以下示例将引导您完成共享屏幕和查看其他人屏幕的过程。

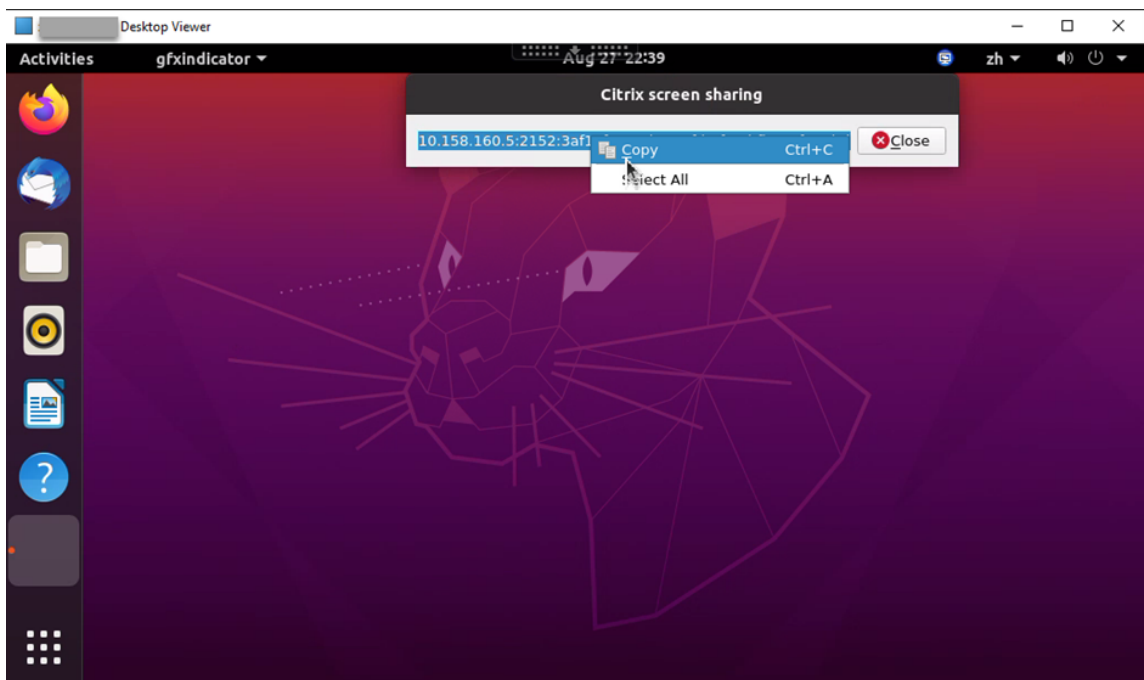
要共享屏幕，请执行以下操作：

1. 在虚拟桌面的通知区域中，单击下面的系统托盘图标并选择屏幕共享 > 共享我的屏幕。



2. 单击复制并关闭。

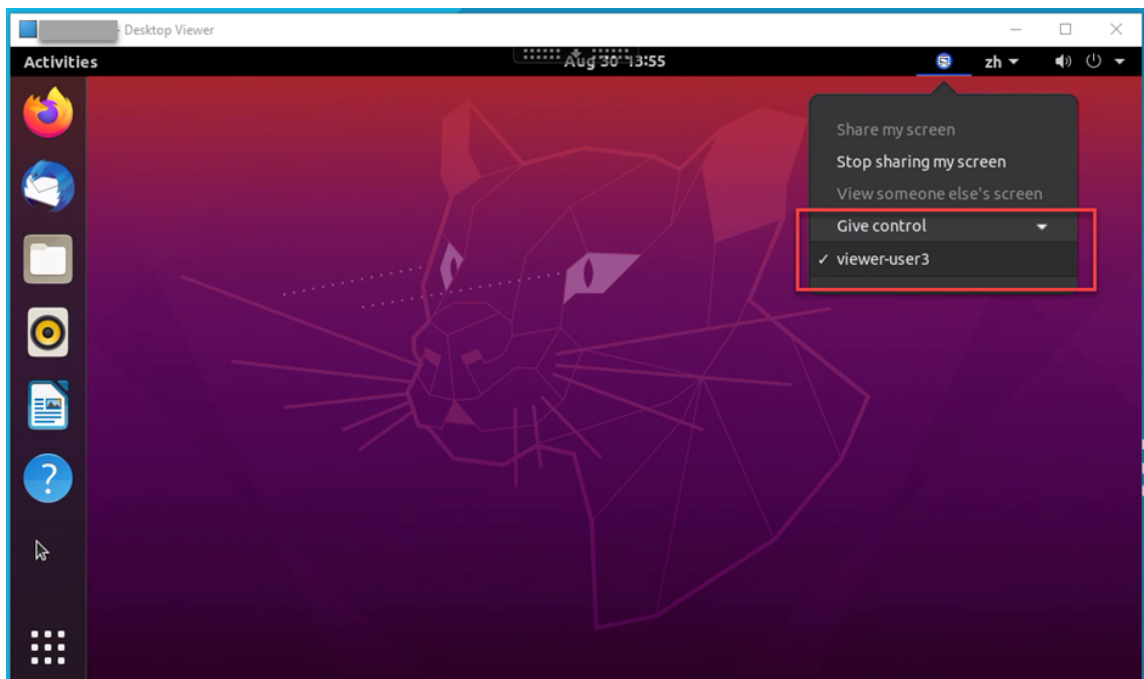
当前的屏幕共享代码将一直保留，直到您停止并重新开始共享屏幕。



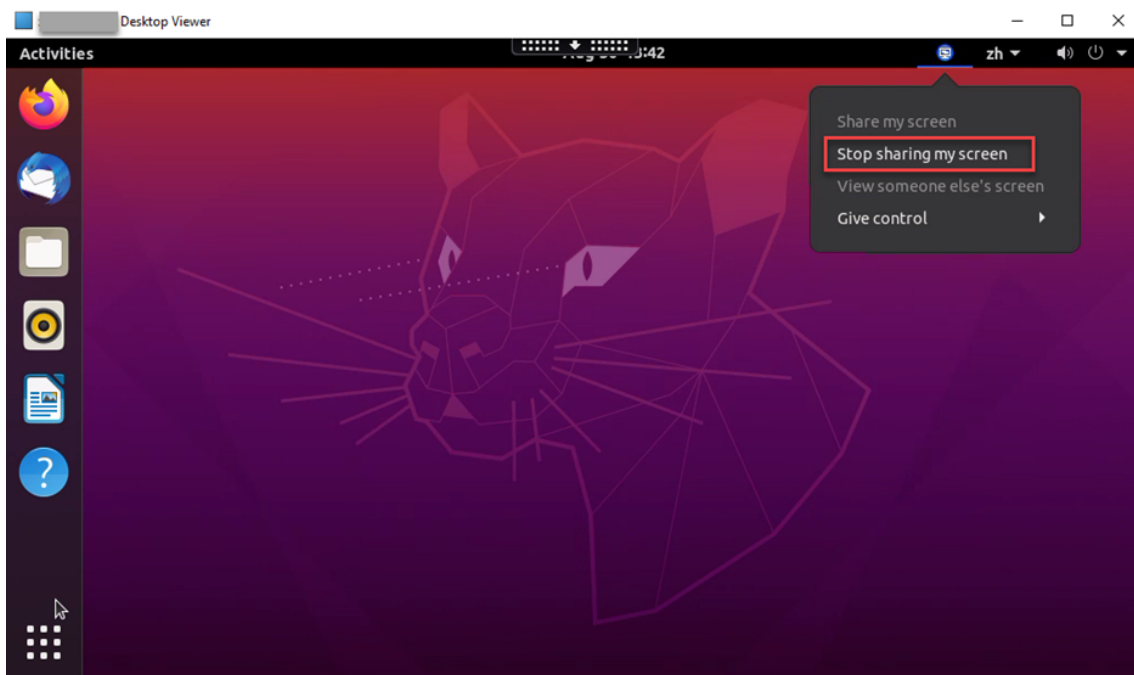
提示：

共享您的屏幕时，屏幕周围有一个红色边框，表示正在进行共享。

3. 与其他虚拟桌面（要与其共享您的屏幕）上的会话用户共享复制的代码。
4. 要允许某个查看器控制您的屏幕，请选择授予控制权，然后选择查看者的名称。要停止授予控制权，请清除该查看器的名称。

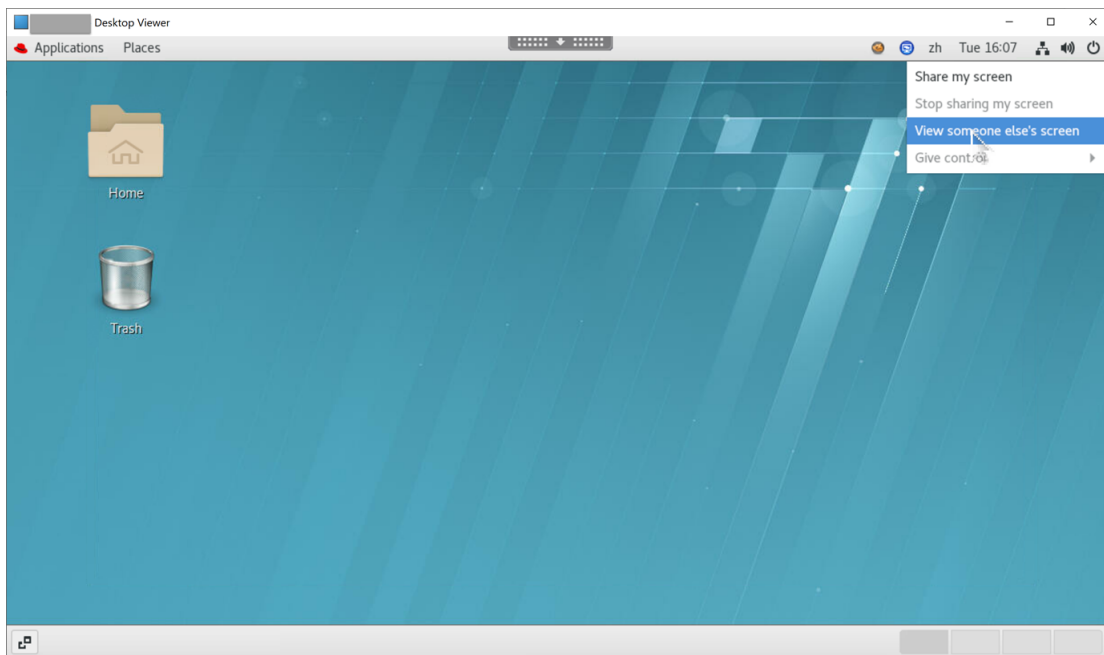


5. 要停止共享屏幕，请选择停止共享我的屏幕。

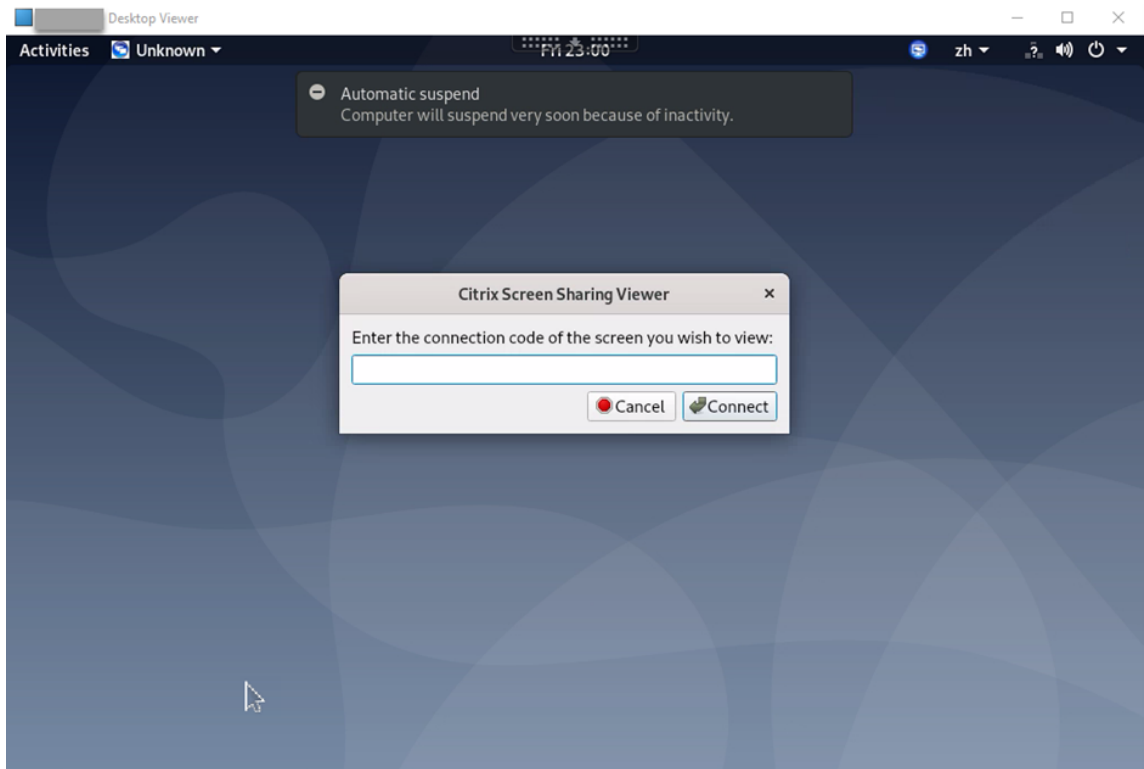


要查看其他人的屏幕，请执行以下操作：

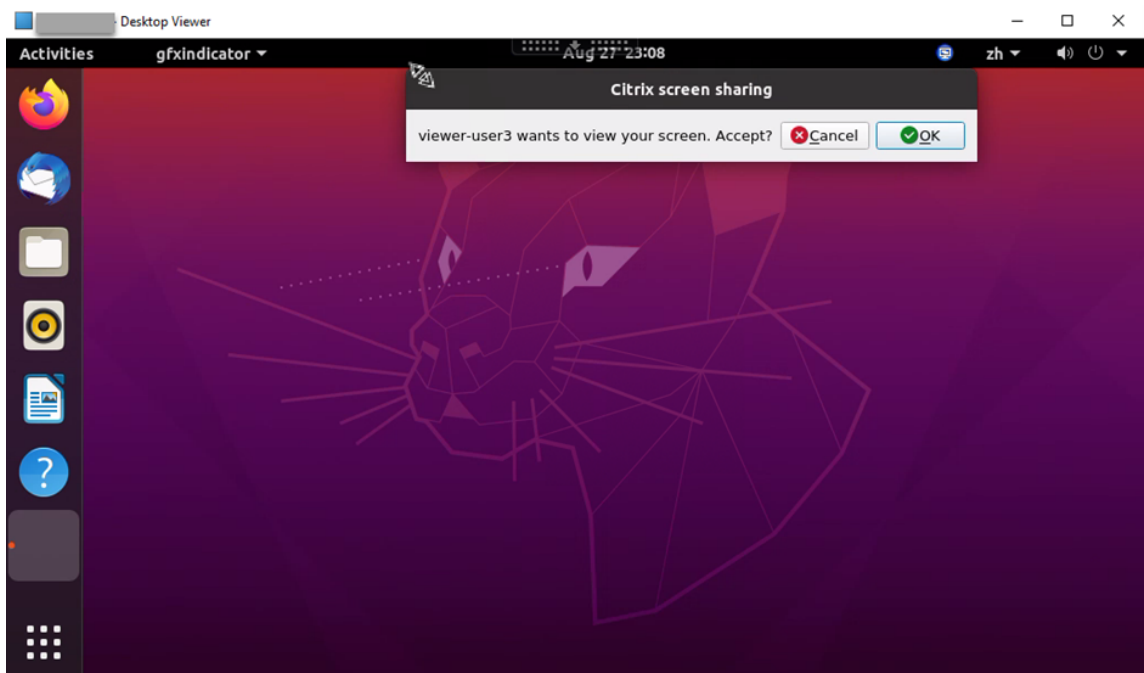
1. 在虚拟桌面的通知区域中，单击屏幕共享图标并选择查看其他人的屏幕。



2. 输入要查看的屏幕的连接代码，然后单击连接。



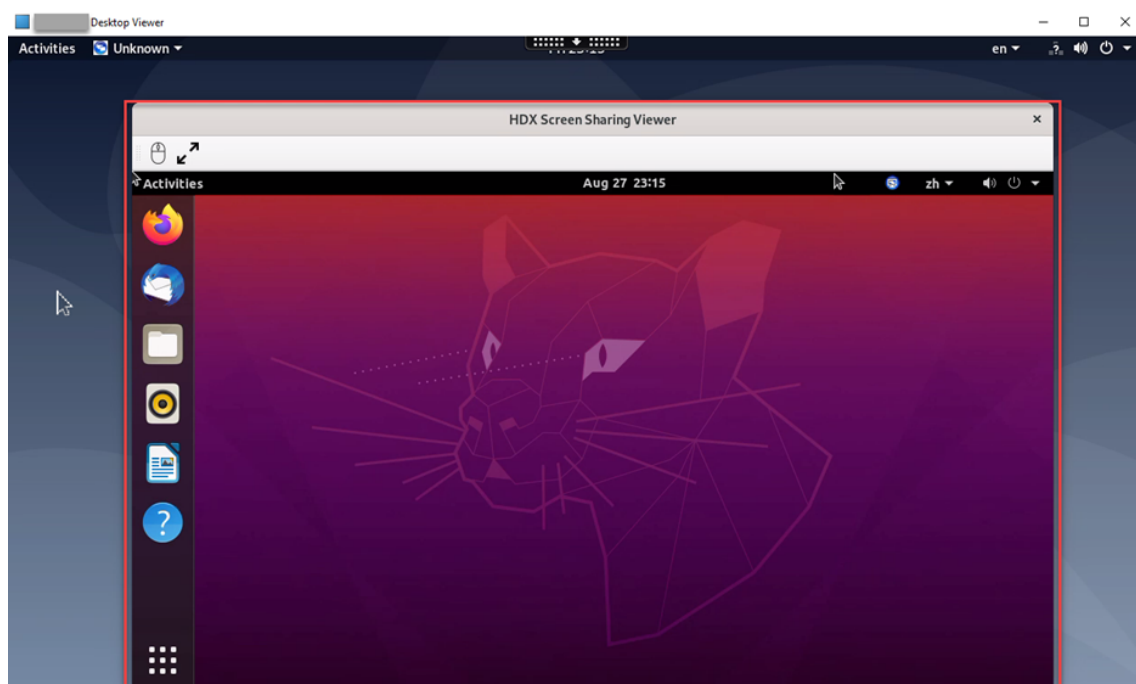
3. 等待屏幕共享者接受您的请求。例如：



提示：

- 在共享者方面，Linux 系统会针对您的请求发出通知。
- 如果共享者未在 30 秒内接受您的请求，您的请求将过期并显示提示。

4. 屏幕共享者通过单击确定接受您的请求后，共享的屏幕将显示在 Desktop Viewer 中。您以查看者的身份与自动分配的用户名连接。

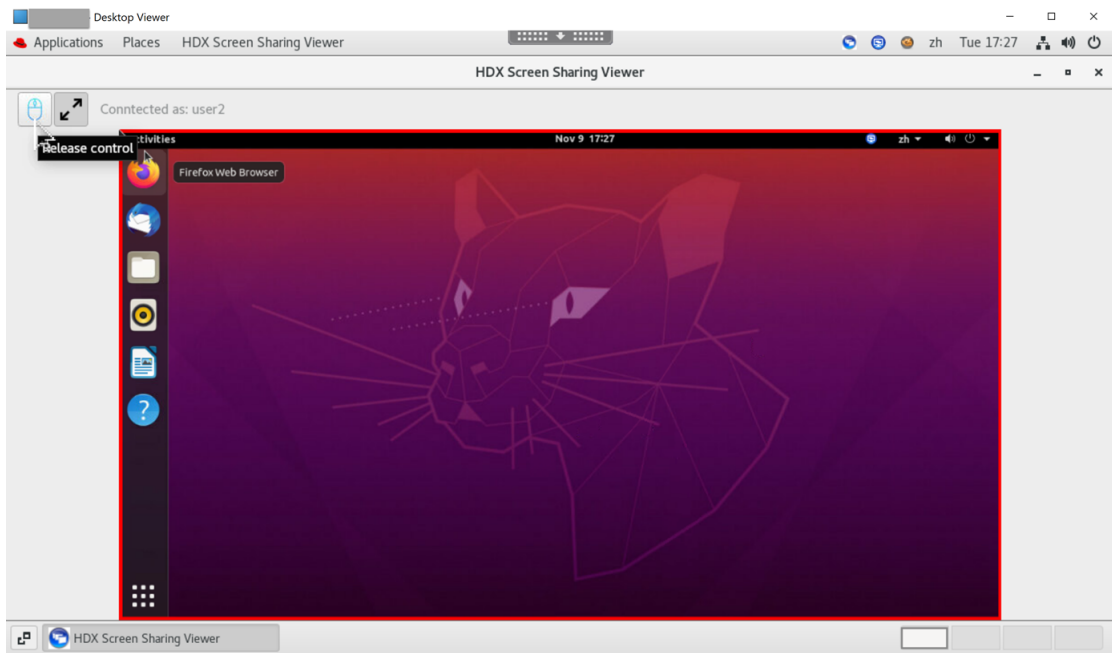


5. 要请求控制共享的屏幕，请单击左上角的鼠标图标。

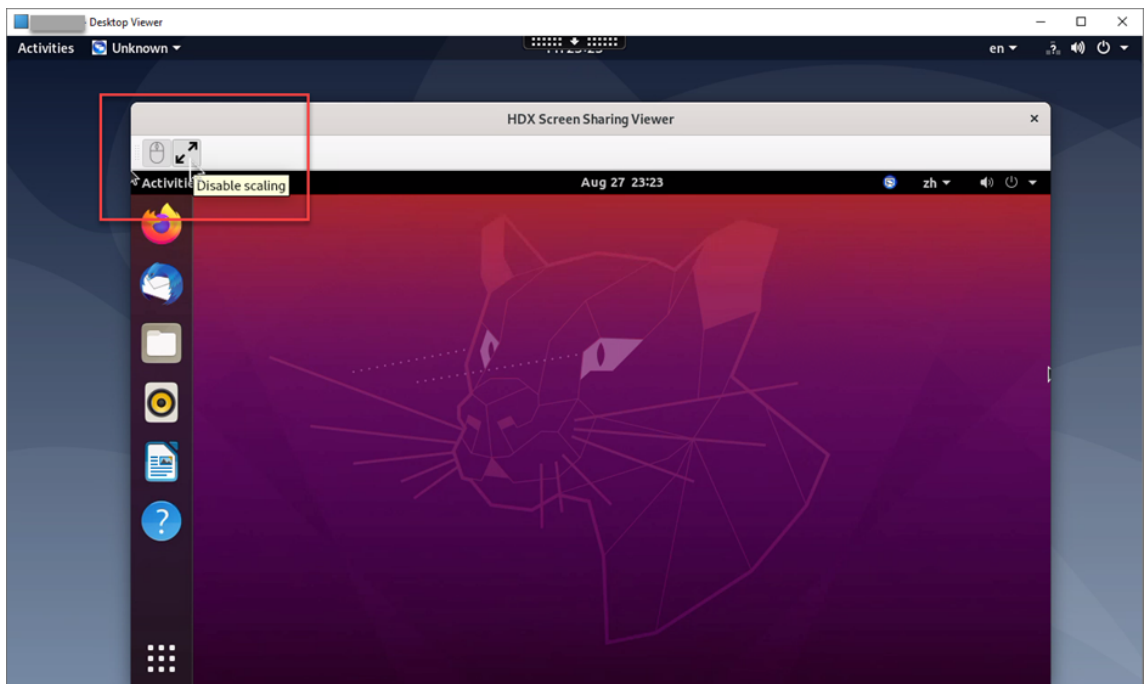
提示：

- 如果共享者未在 30 秒内接受您的请求，您的请求将过期。
- 一次只允许一个查看者控制共享屏幕。

再次单击鼠标图标可释放对共享屏幕的控制权。



6. 要禁用显示缩放或缩放到窗口大小，请单击鼠标图标旁边的图标。



配置

屏幕共享功能默认处于禁用状态。要启用该功能，请完成以下设置：

1. 启用系统托盘。
2. 对于 Citrix Virtual Apps and Desktops 2112 及更高版本，请在 Citrix Studio 中启用屏幕共享策略。

3. (可选) 对于 Citrix Virtual Apps and Desktops 2109 及更早版本, 请运行以下命令在 Linux VDA 上启用屏幕共享:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
   CurrentControlSet\Control\Citrix\Thinwire" -v "
   EnableScreenSharing" -d "0x00000001"
```

4. 在您的防火墙中允许使用端口 52525–52625。

注意事项

- 屏幕共享功能不支持 H.265 视频编解码器。
- 屏幕共享功能对应用程序会话不可用。
- 默认情况下, 桌面会话的用户最多可以与 10 个查看者共享其会话屏幕。查看者的最大数量可通过 `ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire"-v "ScreenSharingViewerMaxNum"-d <hex_value>` 进行配置。当达到最大数量时, 用户尝试接受额外的连接请求时会出现提示。

多显示器支持

August 20, 2024

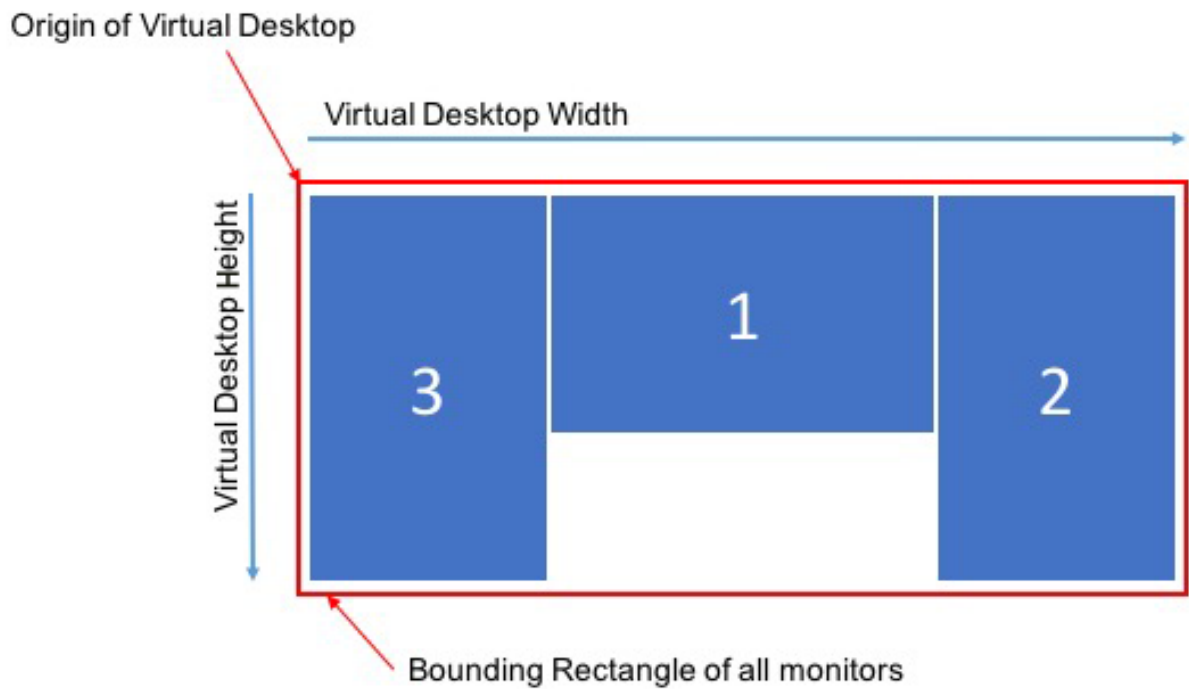
概述

Linux VDA 提供开箱即用的多显示器支持, 每个显示器的默认分辨率为 2560×1600。标准 VDA 最多支持九个显示器, HDX 3D Pro VDA 最多支持四个显示器。

本文介绍了如何针对不同的显示器分辨率和布局配置 Linux VDA。

虚拟会话桌面

与 Windows VDA 一样, Linux VDA 具有多显示器虚拟桌面的概念。它基于所有显示器的边框, 而非显示器的实际布局。因此, 从理论上讲, 虚拟桌面的区域可以大于客户端显示器所覆盖的区域。



虚拟会话桌面大小

虚拟会话桌面的原点是根据所有显示器的边框的左上角计算得出的。该点位于 $X = 0, Y = 0$ 位置，其中 X 和 Y 分别是横轴和竖轴。

虚拟会话桌面的宽度是指从所有显示器的边框的原点到右上角的水平距离（以像素为单位）。

同样，虚拟会话桌面的高度是指从所有显示器的边框的原点到左下角的垂直距离（以像素为单位）。

此计算非常重要，原因如下：

- 允许不同的客户端显示器布局
- 了解 Linux VDA 上的内存使用情况

允许不同的客户端显示器配置

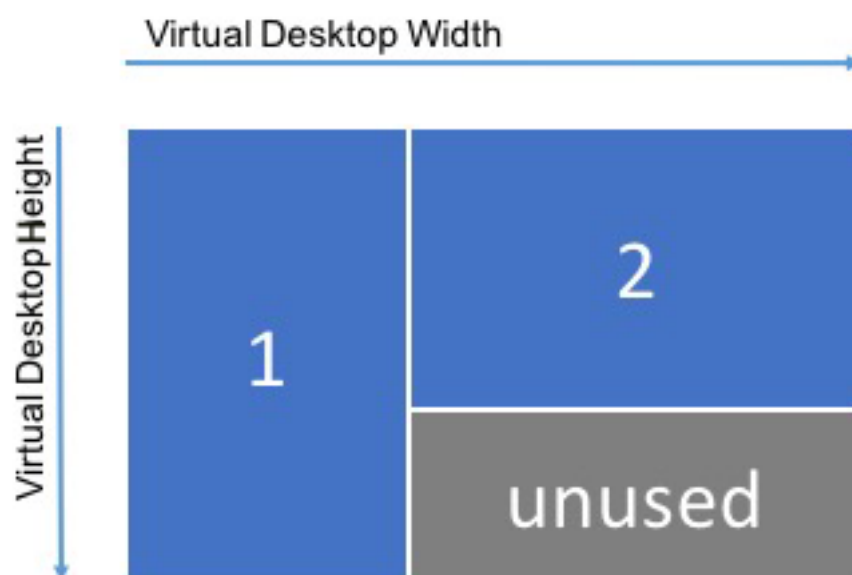
通过了解各种客户端显示器配置的最大虚拟桌面大小，您可以配置 Linux VDA，使其在客户端显示器配置方面更加灵活。

请注意以下客户端监视器配置：



上图显示了开箱即用的多显示器配置，其中包含两个显示器，每个显示器的分辨率为 2560×1600。

现在，请考虑使用以下客户端显示器配置连接到同一个 Linux VDA：



如果上图中每个显示器的分辨率均为 2560×1600，开箱即用的多显示器配置参数将不足。最大高度太小，无法容纳此显示器布局的虚拟会话桌面。要适应本示例中的客户端显示器配置，必须将 Linux VDA 虚拟桌面的大小设置为 4160×2560。

要在多显示器配置中获得最大的灵活性，请在要支持的所有显示器布局中找到最小的边框。对于包含两个 2560×1600 显示器的配置，可能的布局包括：

- 显示器 **1** 2560×1600 和显示器 **2** 2560×1600
- 显示器 **1** 1600×2560 和显示器 **2** 2560×1600
- 显示器 **1** 2560×1600 和显示器 **2** 1600×2560
- 显示器 **1** 1600×2560 和显示器 **2** 1600×2560

要适应上述所有布局，您需要一个 5120×2560 的虚拟会话桌面。它是可以包含所有所需布局的最小边框。

如果所有用户在典型的横向布局中只有一个显示器，请将最大虚拟桌面大小设置为显示器的最高分辨率。



在此示例中，需要将虚拟桌面的大小设置为 2560×1600。由于默认配置为 5120×1600 和 2 个显示器，因此需要更改配置以优化单显示器部署的内存使用情况。

注意：

如果在多显示器设置中桌面显示分辨率不正确，请在 Citrix Workspace 应用程序中调整每英寸点数 (DPI) 设置。有关详细信息，请参阅知识中心文章 [CTX230017](#)。

了解 Linux VDA 上的内存使用情况

通过了解虚拟桌面的大小，您可以计算每个 HDX 会话使用的内存量。此内存是指会话开始时分配给每个会话用来保存图形数据的内存。在会话的生命周期中，此内存的大小不变。虽然此内存不是会话使用的内存总量，但它是计算每个会话内存使用量的最简单的方法。

要计算分配给每个 HDX 会话的内存量，请使用以下公式：

$$M = X \times Y \times Z,$$

其中：

- **M** 为用于会话图形的内存量。
- **X** 为虚拟会话桌面的宽度。
- **Y** 为虚拟会话桌面的高度。
- **Z** 为 HDX 会话窗口的颜色深度。该值以字节为单位，而不是以位为单位，因此使用 4 表示 32 位颜色。

注意：

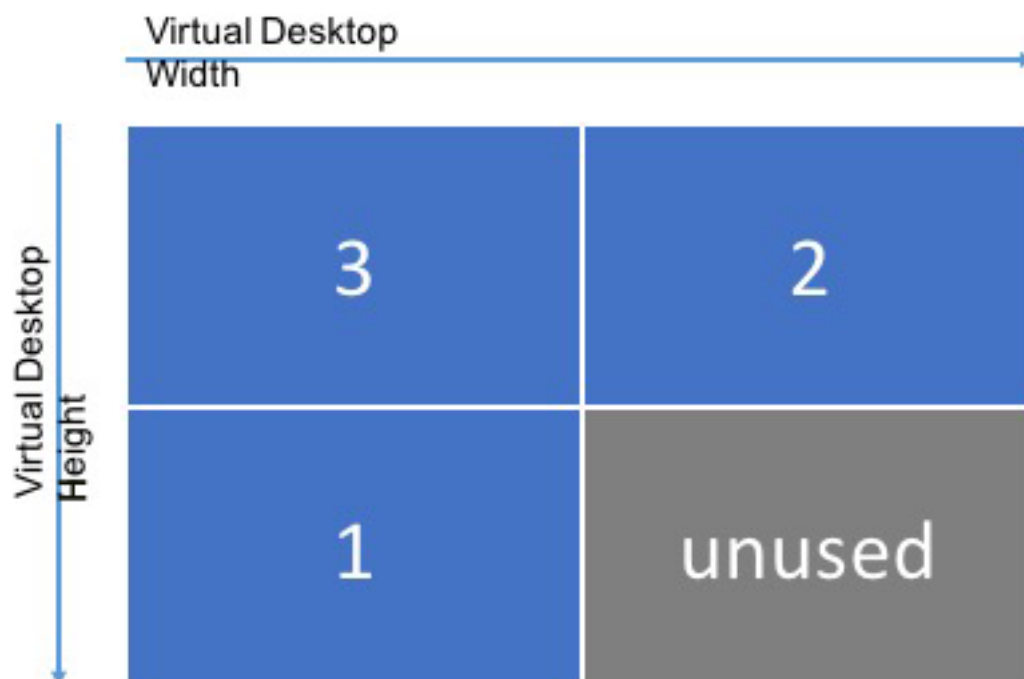
X 服务器的颜色深度会启动，但不会随着会话的生命周期而改变（从登录到断开连接/重新连接，直至注销）。因此，Linux VDA 始终将虚拟会话桌面分配为 32 位，并根据会话请求的颜色深度向下采样。

例如，对于 1024×768 会话，使用的内存为：

$$1024 \times 768 \times 4 / 2^{20} \text{ MB} = 3 \text{ MB}$$

了解内存使用情况对于提高每个 Linux VDA 上的会话密度非常重要。

请注意以下客户端监视器配置：



如果每个显示器的分辨率为 2560×1600，为了适应此客户端显示器配置，虚拟会话桌面大小必须为 5120×3200。请注意，灰色区域未使用，相当于 16384000（即 2560 x 1600 x 4）字节的内存浪费量。

Citrix 多显示器配置参数

可以使用以下配置参数控制 Linux VDA 的多显示器功能：

- **MaxScreenNum**

参数: HKEY_LOCAL_MACHINE/System/CurrentControlSet/Control/Citrix/Thinwire/MaxScreenNum

说明: 要支持的显示器数量

类型: DWORD

默认值: 4

最大值: 标准 VDA 为 9, HDX 3D Pro VDA 为 4

- **MaxFbWidth**

参 数: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFb-Width

说明: 虚拟会话桌面的最大宽度

类型: DWORD

默认值: 5120

最大值: 16384 (8192 x 2)

- **MaxFbHeight**

参 数: HKEY_LOCAL_MACHINE /System/CurrentControlSet/Control/Citrix/Thinwire/MaxFb-Height

说明: 虚拟会话桌面的最大高度

类型: DWORD

默认值: 1600

最大值: 16384 (8192 x 2)

更改 Linux VDA 多显示器配置

以下部分概述了如何在 Linux VDA 上启用、配置和禁用多显示器功能。

请使用以下方法设置显示器的最大数量:

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxScreenNum" -d " NumMons" --force
```

其中 **NumMons** 为介于 1 到 9 (适用于标准 VDA) 或 1 和 4 (适用于 HDX 3D Pro VD) 之间的值。

请使用以下方法设置虚拟会话桌面的最大宽度:

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxFbWidth" -d " MaxWidth" --force
```

其中 **MaxWidth** 为介于 **1,024** 到 **16,384** 之间的值。

请使用以下方法设置虚拟会话桌面的最大高度:

```
1 sudo ctxreg create -k " HKEY_LOCAL_MACHINE \System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "MaxFbHeight" -d " MaxHeight" --force
```

其中 **MaxHeight** 为介于 **1,024** 到 **16,384** 之间的值。

未虚拟化的 GPU

August 20, 2024

在 Linux VDA 文档中，未虚拟化的 GPU 是指：

- Remote PC Access 场景中使用的 GPU
- 从虚拟机管理程序传递的 GPU

本文提供了有关支持非虚拟化 GPU 的信息。

为支持适用于 Linux 的 NVIDIA Capture SDK 的 NVIDIA GPU 启用 HDX 3D Pro

对于支持适用于 Linux 的 NVIDIA Capture SDK 的 NVIDIA GPU，只需在安装 Linux VDA 时将 **CTX_XDL_HDX_3D_PRO** 设置为 **Y** 即可启用 HDX 3D Pro。无需执行其他配置。启用 HDX 3D Pro 后，硬件加速默认处于启用状态。

与不支持适用于 Linux 的 NVIDIA Capture SDK 的 NVIDIA GPU 以及 AMD 和 Intel 等其他制造商提供的 GPU 兼容

注意：

在这种情况下，仅支持软件编码。

步骤 1： 安装 Linux VDA 时将 **CTX_XDL_HDX_3D_PRO** 设置为 **Y**

有关环境变量的信息，请参阅 [步骤 8：设置运行时环境以完成安装](#)。

步骤 2： 安装 **Xdamage**

例如，您可以运行 **sudo apt-get install -y libxdamage1** 在 Ubuntu 20.04 上安装 XDamage。通常情况下，XDamage 是作为 XServer 的扩展程序。

步骤 3： 通过运行以下命令启用 **XDamage**

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\XDamage" -t "REG_DWORD" -v "XDamageEnabled" -d "0x00000001" --force
```

步骤 4: 修改 Xorg 配置文件

可以在 `/etc/X11` 下找到下面四个模板配置文件。根据连接的显示器的数量，修改其中一个名称中带相应数量的模板配置文件。例如，如果仅连接了一台显示器，请修改名称中带数字 1 的模板配置文件，即 `ctx-driver_name-1.conf`。如果连接了两台显示器，请修改名称中带数字 2 的模板配置文件，即 `ctx-driver_name-2.conf`。

- `ctx-driver_name-1.conf`
- `ctx-driver_name-2.conf`
- `ctx-driver_name-3.conf`
- `ctx-driver_name-4.conf`

以 **`ctx-driver_name-1.conf`** 为例，执行以下操作来修改模板配置文件：

1. 将 **`driver_name`** 替换为实际的驱动程序名称。

例如，如果您的驱动程序名称为 `intel`，可以将配置文件名称更改为 `ctx-intel-1.conf`。

2. 添加视频驱动程序信息。

每个模板配置文件都一个包含名为“Device”的部分，这部分被注释掉。本节介绍视频驱动程序信息。请在添加您的视频驱动程序信息之前先完成本节内容。要启用本部分内容，请执行以下操作：

- a) 请参阅 GPU 制造商提供的指南以了解配置信息。可以生成本机配置文件。使用本机配置文件验证您的 GPU 是否可以在本地环境中运行。
- b) 将本机配置文件的“Device”部分复制到 **`ctx-driver_name-1.conf`**。

3. 运行以下命令来设置注册表项，以使 Linux VDA 能够识别在步骤 1 中设置的配置文件名称。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
  CurrentControlSet\Control\Citrix\XDamage" -t "REG_SZ" -v "
  DriverName" -d "intel" --force
```

监视 Remote PC Access VDA 的空白

对于使用未虚拟化的 GPU 的 Remote PC Access VDA，Linux VDA 支持物理显示器遮蔽。

支持该功能的经过全面测试的 Linux 发行版包括 Ubuntu 20.04 和 Debian 11。

默认情况下，该功能处于禁用状态。要将其启用，请完成下面两个步骤：

1. 根据您的 Linux 发行版安装 `evdi-dkms` 软件包：

```
1 sudo apt install evdi-dkms
```

2. 启用图形显示卸载到 EVDI：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  EVDI" -d "0x00000001" --force
```


3. 如果您使用的是 Intel GPU，请禁用显示管理器。否则，显示管理器会占用 Intel GPU 并使其无法用于 Citrix 远程会话。

```
1 sudo systemctl disable --now gdm
```

故障排除

无图形输出或图形输出为乱码

如果您可以在本地运行 3D 应用程序，并且所有配置均正确，则丢失图形输出或图形输出为乱码缺陷导致的。请使用 `/opt/Citrix/VDA/bin/setlog` 并将 `GFX_X11` 设置为 “verbose” 来收集跟踪信息以进行调试。

会话水印

July 15, 2024

会话水印有助于威慑和启用数据窃取的跟踪功能。跟踪信息在会话桌面上显示，对使用相机和屏幕截图窃取数据的用户具有威慑作用。您可以将水印指定为文本层或带有 alpha 通道的 PNG 图像。该水印在整个会话屏幕上显示，但不会改变原始文档的内容。

重要：

会话水印不是一项安全功能。它不能完全阻止数据盗窃，但可以提供一定级别的威慑作用和可跟踪性。使用此功能时，我们不保证完整的信息可追溯性。相反，我们建议您将此功能与其他安全解决方案（如果适用）结合使用。

会话水印中包含用于跟踪数据盗窃的信息。最重要的数据是创建屏幕图像的会话的用户的身份（通过其登录凭据进行跟踪）。为了更有效地跟踪数据泄漏，请包括服务器或客户端 Internet 协议地址以及连接时间等其他信息。

要调整用户体验，请使用以下会话水印策略设置配置屏幕上的放置位置和水印外观：

会话水印策略设置

启用会话水印

启用了此设置时，会话显示屏幕上将覆盖一层显示会话特定信息的不透明水印。其他水印设置取决于此设置的启用。

默认情况下，会话水印处于禁用状态。

包括客户端 IP 地址

启用了此设置时，会话将显示当前的客户端 IP 地址作为水印。

默认情况下，包括客户端 IP 地址处于禁用状态。

包括连接时间

启用了此设置时，会话水印将显示连接时间。格式为 yyyy/mm/dd hh:mm。显示的时间取决于系统时钟和时区。

默认情况下，包括连接时间处于禁用状态。

包括登录用户名

启用了此设置时，会话将显示当前的登录用户名作为水印。显示格式为 USERNAME@DOMAINNAME。我们建议用户名最多包含 20 个字符。当用户名超过 20 个字符时，可能会出现较小的字体大小或截断，这降低了水印的有效性。

默认情况下，包括登录用户名处于启用状态。

包括 VDA 主机名

启用了此设置时，会话将显示当前 ICA 会话的 VDA 主机名作为水印。

默认情况下，包括 VDA 主机名处于启用状态。

包括 VDA IP 地址

启用了此设置时，会话将显示当前 ICA 会话的 VDA IP 地址作为水印。

默认情况下，包括 VDA IP 地址处于禁用状态。

会话水印样式

此设置控制您显示单个水印文本标签还是多个标签。请从值下拉菜单中选择多个或单个。有关其他样式选项，请参阅本文中的水印自定义文本部分。

多个将在会话中显示 5 个水印标签。其中 1 个标签在中心显示，另外 4 个在边角显示。

单个将在会话中心显示 1 个水印标签。

默认情况下，会话水印样式设置为多个。

水印透明度

可以指定介于 0 到 100 之间的水印透明度。指定的值越大，水印越不透明。

默认情况下，值为 17。

水印自定义文本

默认情况下，该值为空。可以键入非空字符串，设置语法以形成字符串，或者使用组合形式以在会话水印中显示。非空字符串每行最多支持 25 个 Unicode 字符。较长的字符串将被截断为 25 个字符。

例如，您可以将策略设置为以下值：

```
<date> <time><newline><username><style=single><fontsize=40><font=
Ubuntu><position=center><rotation=0><newline><serverip><newline><
clientip><newline>Citrix Linux VDA<newline>Version 2207
```

有关所有语法选项的说明，请参见下表：

语法选项	说明	有效设置（区分大小写）	默认值	备注
<style>	水印布局样式	xstyle, single, tile, horizontal	xstyle	-
<position>	水印位置	center、topleft、topright、bottomleft、bottomright	center	仅当布局样式设置为单一时才有效。
<rotation>	水印旋转到一定角度	-180-180	0	-
<transparency>	水印不透明度	0-100	17	-
	-	系统支持的字体	Sans	-
<fontsize>	-	20-50	0（自动计算）	-
<fontzoom>	通过 <fontsize> 和 <image> 设置的字体和图像大小的百分比	0 -	100	-

语法选项	说明	有效设置（区分大小写）	默认值	备注
<image>	PNG 水印	VDA 上的 PNG 图像的路径	不适用	此语法配置 PNG 水印。仅支持带有 alpha 通道的 PNG。使用 PNG 水印时，只有 <style>、<position>、<rotation>、<transparency> 和 <fontzoom> 语法选项才有效。
<date>	会话连接日期的占位符 (YYYY/MM/DD)	不适用	不适用	-
<time>	会话连接时间的占位符 (HH:MM)	不适用	不适用	-
<domain>	用户帐户域的占位符	不适用	不适用	-
<username>	当前登录用户名的占位符（不包括用户帐户域）	不适用	不适用	-
<hostname>	VDA 主机名的占位符	不适用	不适用	-
<clientip>	客户端 IP 地址的占位符	不适用	不适用	-
<serverip>	VDA 的 IP 地址的占位符	不适用	不适用	-

注意：

如果至少通过一个有效的语法选项指定水印自定义文本，则会忽略除启用会话水印之外的所有其他会话水印策略。

如果未指定语法选项或者将其设置为不支持的值，则使用其默认值。

限制

- 在以下任一情况下都支持会话水印：
 - 当使用视频编解码器进行压缩设置为针对整个屏幕时。
 - 当使用视频编解码器进行压缩设置为偏好时使用，并且启用了[针对 3D 图形工作负载优化](#)时。
- 在使用浏览器内容重定向的会话中不支持会话水印。要使用会话水印功能，请务必禁用浏览器内容重定向。

- 如果会话在使用旧版 NVIDIA 驱动程序的全屏硬件加速 H.264 或 H.265 编码模式下运行，则不支持会话水印，并且不显示会话水印。（在这种情况下，注册表中的 NvCaptureType 设置为 2。）
- 水印对会话重影功能不可见。
- 如果您按 Print Screen 键捕获屏幕，在 VDA 端捕获的屏幕将不包括水印。我们建议您采取措施避免复制屏幕捕获。

多会话 Linux VDA 上的共享 GPU 加速

July 15, 2024

HDX 3D PRO 仅支持为 VDI 桌面配置的 Linux VDA（单会话模式）。对于多会话 Linux VDA，您可以在其上启用共享 GPU 加速功能以加快 OpenGL 3D 应用程序的运行速度。

注意：

共享 GPU 加速功能不支持 Wayland 显示服务器。

配置

要在多会话 Linux VDA 中启用共享 GPU 加速功能以加快 OpenGL 3D 应用程序的运行速度，请完成以下配置步骤：

步骤 1：安装 VirtualGL

从 <https://sourceforge.net/projects/virtualgl/files> 中下载并安装 **VirtualGL**。下载适用于基于 Debian 的 Linux 发行版的 **.deb** 软件包和基于 RHEL 的 Linux 发行版的 **.rpm** 软件包。

步骤 2：配置 VirtualGL

1. 停止 Linux 显示管理器，例如 LightDM 或 GNOME Display Manager (GDM)。
2. 请通过运行以下命令来执行 VirtualGL 配置脚本：

```
1 #/opt/VirtualGL/bin/vglserver_config
```

我们建议您在脚本执行过程中做以下选择：

- 选择 “1” 以 “将服务器配置为与 VirtualGL (GLX + EGL 后端) 结合使用”
- 选择 “n” 以 “限制 3D X 服务器对 **vglusers** 组的访问”
- 选择 “n” 以 “限制帧缓冲设备对 **vglusers** 组的访问”
- 选择 “n” 以 “禁用 XTEST 扩展程序”

3. 退出配置脚本并重新启动 Linux 显示管理器。

步骤 3: 运行启用了 GPU 加速功能的 OpenGL 3D 应用程序

有两种方法可以在 Linux VDA 会话中运行启用了 GPU 加速功能的 OpenGL 3D 应用程序:

- 方法 1: 为所有 OpenGL 3D 应用程序启用共享 GPU 加速

为此, 请在 Linux VDA 中打开 bash 终端, 运行以下命令, 然后重新启动 bash 终端。这种方法为从 bash 终端启动的所有 OpenGL 3D 应用程序启用共享 GPU 加速。

```
1 #/opt/Citrix/VDA/sbin/ctxgpushare.sh enable
```

- 方法 2: 为特定的 OpenGL 3D 应用程序启用共享 GPU 加速:

为此, 请在 Linux VDA 中打开终端, 然后使用指定的应用程序名称运行以下命令:

```
1 #vglrun <AppName>
```

限制

- 共享 GPU 加速与 Linux VDA 上的显示管理器密切配合。建议使用 LightDM 作为共享 GPU 加速的显示管理器, 以实现预期功能和性能。
- 只有 Ubuntu 和 Debian 上的 Firefox 支持 WebGL 硬件加速。

可扩展性

可以共享 GPU 的最大并发会话数量取决于 CPU 和系统内存。它还在很大程度上取决于 GPU 的最大视频内存。

例如:

如果

那么

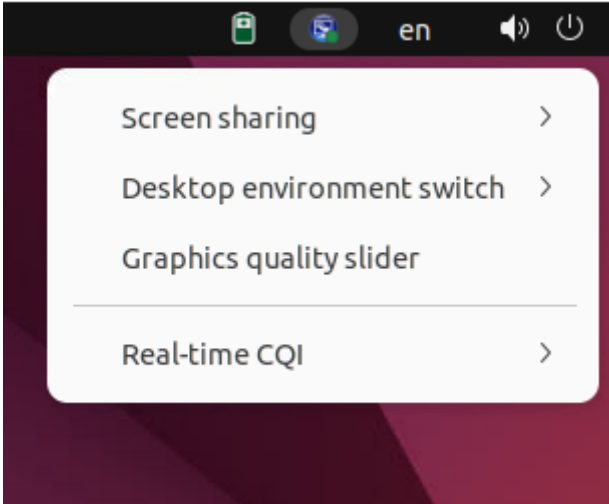
NVIDIA **M10-2B** vGPU 的视频内存为 2048 MB, 诸如 VariCAD 查看器之类的 OpenGL 应用程序在每个会话中使用 100 MB 的视频内存来处理工作负载,

理论上, 支持的最大并发会话数不能超过 20。

系统托盘

August 20, 2024

会话用户可以单击以下系统托盘图标来执行操作或查看指示器：



系统托盘中的项目简介

每个项目对应一个带切换开关的功能。禁用与某个项目对应的功能时，该项目将隐藏且不显示。

- 屏幕共享

有关此功能的详细信息，请参阅 [HDX 屏幕共享](#)。

- 桌面环境切换

此项目是 **ctxdesktopswitch.sh** 的 GUI。有关详细信息，请参阅 [按会话用户划分的自定义桌面环境](#)。

默认情况下，会话用户自定义桌面环境功能处于启用状态。要禁用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "EnableDesktopSwitch" -d "0x00000000" --force
```

- 图形质量滑块

有关详细信息，请参阅图形配置文章中的 [图形质量滑块](#) 部分。

- 实时 CQI

当前显示的是 ICA 往返时间 (RTT) 和延迟数据。有关详细信息，请参阅 [会话数据查询实用程序](#)。

系统托盘图标的显示方式会有所差别，具体取决于实时 CQI 中的延迟：



存在用于控制图标何时更改显示方式的阈值。默认情况下，这些阈值的设置如下：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  HighLatencyThreshold" -d "0x000000dc" --force
```

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  LowLatencyThreshold" -d "0x00000078" --force
```

当实际延迟小于或等于 **LowLatencyThreshold** 时，该图标将标记为绿色。当实际延迟大于 **HighLatencyThreshold** 时，该图标将标记为红色。在其他情况下，图标标记为黄色。如果禁用了实时 CQI，该图标将没有颜色标记。

默认情况下，实时 CQI 已启用并显示。要将其禁用并隐藏，并使托盘图标没有颜色标记，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  EnableCqiShow" -d "0x00000000" --force
```

- 显卡状态

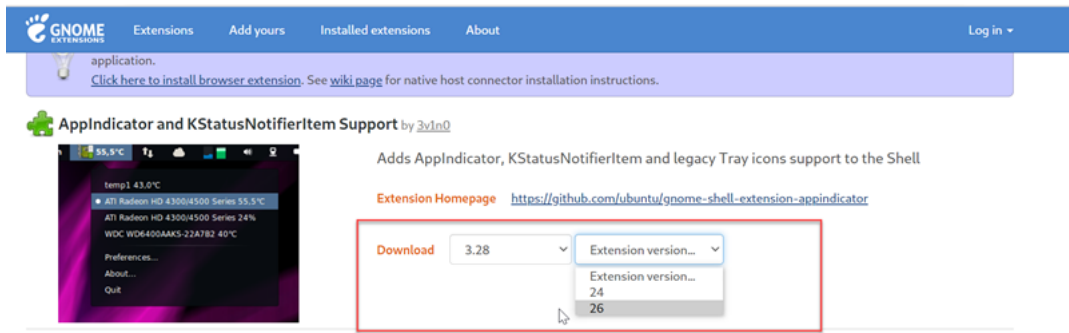
此指示器显示当前会话的显卡设置。默认启用此功能。要禁用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
  EnableGfxInfo" -d "0x00000000" --force
```

启用系统托盘

默认情况下，系统托盘和客户端电池状态显示处于启用状态。但是，在某些情况下，必须进行额外配置才能启用系统托盘和客户端电池状态显示。详细信息如下：

1. 在 Citrix Studio 中启用图形状态指示器策略。
2. (仅当您使用随 **GNOME** 一起安装的 **RHEL 8.x/9.x**、**Rocky Linux 8.x/9.x**、**Debian 11** 或 **SUSE 15.x** 时，才需要执行此步骤。) 为您的 GNOME shell 安装兼容的扩展程序以启用 AppIndicator 支持。
 - a) 运行命令 `gnome-shell --version` 以检查 GNOME shell 版本。
 - b) 从 <https://extensions.gnome.org/extension/615/appindicator-support> 下载 GNOME shell 的兼容扩展程序。例如，如果您的 shell 版本为 3.28，则可以选择扩展程序版本 24 或 26。



- c) 解压缩下载的软件包,并将解压缩后的目录重命名为 **appindicatorssupport@rgcjonas.gmail.com**。确认软件包的 **metadata.json** 文件中的 **uuid** 值已设置为 **appindicatorssupport@rgcjonas.gmail.com**。
- d) 运行 `mv` 命令以将 **appindicatorssupport@rgcjonas.gmail.com** 目录移动到 `/usr/share/gnome-shell/extensions/` 下方的位置。
- e) 运行 `chmod a+r metadata.json` 命令以使 **metadata.json** 文件可供其他用户读取。

提示：

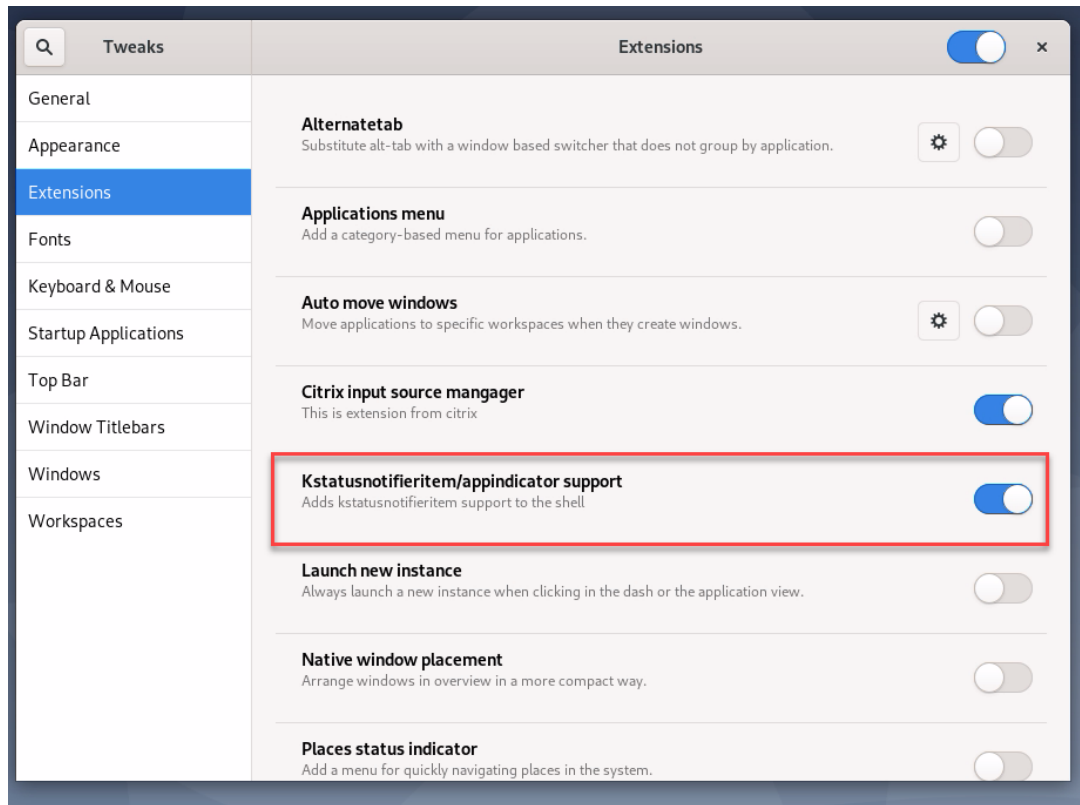
默认情况下,只有根用户可以读取 **appindicatorssupport@rgcjonas.gmail.com** 目录中的 **metadata.json** 文件。要支持屏幕共享,请确保其他用户也能够读取 **metadata.json** 文件。

- f) 安装 GNOME Tweaks。
- g) 在桌面环境中,通过按顺序按 `Alt+F2`、`r` 和 `Enter` 键或通过运行 `killall -SIGQUIT gnome-shell` 命令来重新加载 GNOME shell。
- h) 在桌面环境中,运行 GNOME Tweaks,然后在 Tweaks 或“扩展程序”工具中启用 **KStatusNotifierItem/AppIndicator** 支持。

注意：

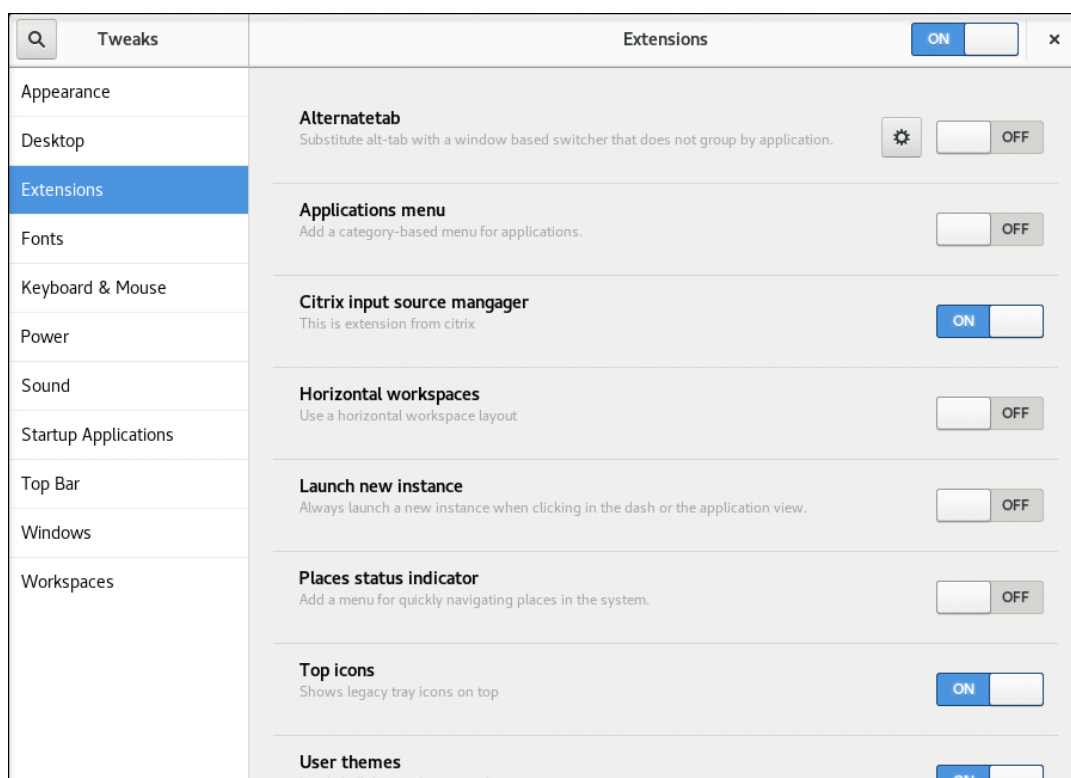
对于与 GNOME 一起安装的 RHEL 9.x、Rocky Linux 9.x 和 SUSE 15.x,扩展程序管理已从 Tweaks 中移出。我们建议您从 GNOME Software 应用商店下载 GNOME 扩展程序,然后在 GNOME 扩展程序中启用 AppIndicator 支持。

3. (仅当您使用随 **GNOME** 一起安装的 **Debian 11** 时,才需要执行此步骤。) 请完成以下步骤以安装和启用 GNOME 系统托盘图标。
 - a) 运行 `sudo apt install gnome-shell-extension-appindicator` 命令。为使 GNOME 能够看到该扩展程序,您可能必须注销后再重新登录。
 - b) 在活动屏幕中搜索 Tweaks。
 - c) 在 Tweaks 工具中选择扩展程序。
 - d) 启用 **Kstatusnotifieritem/appindicator** 支持。



4. (如果您使用的是随 **GNOME** 一起安装的 **RHEL 7.9**，则必须执行此步骤。) 请完成以下步骤以安装和启用 GNOME 系统托盘图标。

- a) 在活动屏幕中搜索 Tweaks。
- b) 在 Tweaks 工具中选择扩展程序。
- c) 启用顶部图标。



d) 注销并重新登录会话。

Thinwire 渐进式显示

July 15, 2024

会话交互性在低带宽或高延迟连接中会降级。例如，在 Web 页面上滚动可能会变慢、无响应或断断续续。键盘和鼠标操作可能会滞后于图形更新。

在 7.17 版中，能够使用策略设置来降低带宽占用量，方法是将会话配置为低视觉质量，或者设置较低的颜色深度（16 位或 8 位图形）。但是，您必须知道用户是在使用弱连接。HDX Thinwire 不会根据网络状况动态调整静态图像质量。

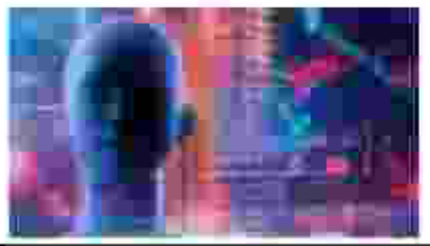
自版本 7.18 起，在以下任一情况下，HDX Thinwire 默认会切换到渐进式更新模式：

- 可用带宽低于 2 Mbps。
- 网络延迟超过 200 毫秒。

在此模式下：

例如，在下图中，渐进式更新模式处于活动状态，字母 **F** 和 **e** 具有蓝色质像，图像已深度压缩。此方法可显著降低带宽占用量，从而提高图像和文本的接收速度，并改进会话交互性。

Features



用户停止与会话交互后，降级的图像和文本将逐渐锐化到无损效果。例如，在下图中，字母不再包含蓝色虚像，图像以源质量显示。

Features



对于图像，锐化使用随机块状方法。对于文本，锐化各个字母或单词的各个部分。锐化过程是在多个帧上进行。此方法可避免在处理单个大型锐化帧时出现延迟。

瞬变影像（视频）仍通过自适应显示或选择性 H.264 进行管理。

如何使用渐进式模式

默认情况下，渐进式模式对视觉质量策略设置高、中（默认设置）和低而言处于随时准备使用状态。

在以下情况下强制关闭（不使用）渐进式模式：

- 视觉质量 = 始终无损或无损构建
- 简单图形的首选颜色深度 = 8 位
- 使用视频编解码器进行压缩 = 针对整个屏幕（需要全屏 H.264 时）

渐进式模式处于随时准备使用状态时，如果出现以下情况之一，则默认启用此模式：

- 可用带宽低于 2 Mbps
- 网络延迟超过 200 毫秒

发生模式切换后，在该模式中至少将花费 10 秒钟时间，即使网络条件暂时不利亦如此。

更改渐进式模式行为

可以通过运行以下命令更改渐进式模式的行为：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplay" -d "<value>" --force
```

其中，<value>：

0 = 始终关闭（在任何情况下都不使用）

1 = 自动（根据网络状况切换，默认值）

2 = 始终打开

处于自动模式 (1) 时，可以运行以下命令之一更改切换渐进式模式时的阈值：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
```

其中 <value> 为 < 阈值，以 Kbps 为单位 >（默认值 = 2048）

示例：4096 = 在带宽低于 4 Mbps 时开启渐进式模式

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "ProgressiveDisplayLatencyThreshold" -d "<value>" --force
```

其中 <value> 为 < 阈值，以毫秒为单位 >（默认值 = 200）

示例：100 = 在网络延迟低于 100 毫秒时开启渐进式模式。

常规内容重定向

July 15, 2024

客户端驱动器映射和客户端文件夹重定向

如果

那么

您仅在主机 (VDA) 上启用了客户端驱动器映射，

客户端完整卷会自动映射到主目录中的 **ctxmnt** 子目录下的会话。

您在主机 (VDA) 上启用了客户端文件夹重定向，用户在用户设备（客户端）上对其进行配置，

将重定向用户指定的本地卷部分。

USB 设备重定向

USB 设备在 Citrix Workspace 应用程序与 Linux VDA 桌面之间共享。将 USB 设备重定向到桌面后，您可以像使用本地连接的 USB 设备那样使用该设备。

客户端驱动器映射

July 15, 2024

可以使用客户端驱动器映射和客户端文件夹重定向使客户端文件在主机端会话中可访问。客户端驱动器映射与客户端文件夹重定向之间的比较如下：

如果	那么
您仅在主机 (VDA) 上启用了客户端驱动器映射，	客户端完整卷会自动映射到主目录中的 ctxmnt 子目录下的会话。
您在主机 (VDA) 上启用了客户端文件夹重定向，用户在用户设备 (客户端) 上对其进行配置，	将重定向用户指定的本地卷部分。

启用客户端驱动器映射

要启用客户端驱动器映射，请在 Citrix Studio 中将客户端驱动器重定向策略设置为允许。有关该策略的详细信息，请参阅[文件重定向策略设置](#)。

启用客户端文件夹重定向并指定要重定向的文件夹

要启用客户端文件夹重定向，请在 VDA 上运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\Client Folder Redirection" -t "REG_DWORD" -v "CFROnlyModeAvailable" -d "0x00000001" --force
```

要指定要将哪些文件夹从客户端重定向到主机端会话，请在用户设备上完成以下步骤：

1. 确保安装了最新版本的 Citrix Workspace 应用程序。
2. 从 Citrix Workspace 应用程序安装目录启动 **CtxCFRUI.exe**。
3. 选择自定义单选按钮，然后添加、编辑或删除文件夹。
4. 断开连接然后重新连接会话，以使设置生效。

USB 设备重定向

August 20, 2024

USB 设备在 Citrix Workspace 应用程序与 Linux VDA 桌面之间共享。将 USB 设备重定向到桌面后，您可以像使用本地连接的 USB 设备那样使用该设备。

提示：

当网络延迟低于 100 毫秒时，我们建议使用 USB 设备重定向。当网络延迟超过 200 毫秒时，请勿使用 USB 设备重定向。

USB 设备重定向包含三个主要方面的功能：

- 开源 USB/IP 项目
- Citrix USB 会话模块
- Citrix USB 服务模块

开源 **USB/IP** 项目：

USB/IP 项目由 Linux 内核驱动程序和一些用户模式库组成，这些库允许您与内核驱动程序通信以获取所有 USB 数据。

Linux VDA 根据开源 USB/IP 项目实现 USB 设备重定向，并重复使用 USB/IP 的内核驱动程序和用户模式库。但是，Linux VDA 与 Citrix Workspace 应用程序之间的所有 USB 数据传输都封装在 Citrix ICA USB 协议中。

Citrix USB 会话模块：

Citrix USB 会话模块充当 USB/IP 内核模块与 Citrix Workspace 应用程序之间的通信桥梁。

Citrix USB 服务模块：

Citrix USB 服务模块管理 USB 设备上的所有操作，例如，连接或断开 USB 设备。

USB 设备重定向的工作原理

通常情况下，如果 USB 设备成功重定向至 Linux VDA，会在系统 /dev 路径中创建一个或多个设备节点。但是，重定向的设备有时不可用于活动的 Linux VDA 会话。USB 设备依赖于驱动程序才能正常使用，且有些设备需要特殊的驱动程序。如果并未提供驱动程序，活动 Linux VDA 会话无法使用重定向的 USB 设备。为确保 USB 设备的连接性，请安装驱动程序并正确配置系统。

Linux VDA 支持从客户端成功重定向的 USB 设备列表。

支持的 **USB** 设备

提示:

我们增加了对 USB 3.0 端口的支持。可以将 USB 3.0 设备插入客户端设备上的 USB 3.0 端口。

下列设备已确认支持此 Linux VDA 版本。可以随意使用其他设备，但会有意外结果:

USB 大容量存储设备	VID:PID	文件系统
Netac Technology Co., Ltd	0dd8:173c	FAT32、NTFS
Kingston Datatraveler 101 II	0951:1625	FAT32、NTFS
Kingston Datatraveler GT101 G2	1567:8902	FAT32、NTFS
SanDisk SDCZ80 flash drive	0781:5580	FAT32、NTFS
WD HDD	1058:10B8	FAT32、NTFS
Toshiba Kingston DataTraveler 3.0 USB device	0930:6545	FAT32、NTFS
Taiwan OEM - OBSOLETE VendorCo ProductCode Disk 2.0	FFFF:5678	FAT32、NTFS
TD-RDF5A Transcend USB device	8564:4000	FAT32、NTFS

注意:

要在 Amazon Linux 2、CentOS、RHEL、Rocky Linux 和 SUSE 上使用 NTFS，请先在这些发行版上启用 NTFS 支持。

USB 3D 鼠标	VID:PID
3DConnexion SpaceMouse Pro	046d: c62b

USB 扫描仪	VID:PID
Epson Perfection V330 photo	04B8: 0142

Yubico USB	VID:PID
Yubico YubiKey OTP+FIDO+CCID - Keyboard, HID	1050:0407

网络摄像机 USB	VID:PID
Logitech composite USB device - WebCam, Audio	0460:0825

配置 **USB** 设备重定向

(仅适用于 **CentOS**、**RHEL** 和 **Rocky Linux**) 安装或编译 **USB/IP** 内核模块

Linux VDA 使用 USB/IP 作为 USB 设备重定向的虚拟主机控制器。因为在大多数情况下，USB/IP 内核模块是随 Linux 内核版本 3.17 及更高版本发布的，因此，默认情况下您不必构建内核模块。但是，USB/IP 内核模块不适用于 CentOS、RHEL 和 Rocky Linux。要在这些 Linux 发行版中使用 USB 设备重定向，必须安装或编译 USB/IP 内核模块。根据您的 Linux 发行版从 <https://pkgs.org/download/kmod-usbip> 下载并安装 USB/IP。

设置 **USB** 设备重定向策略

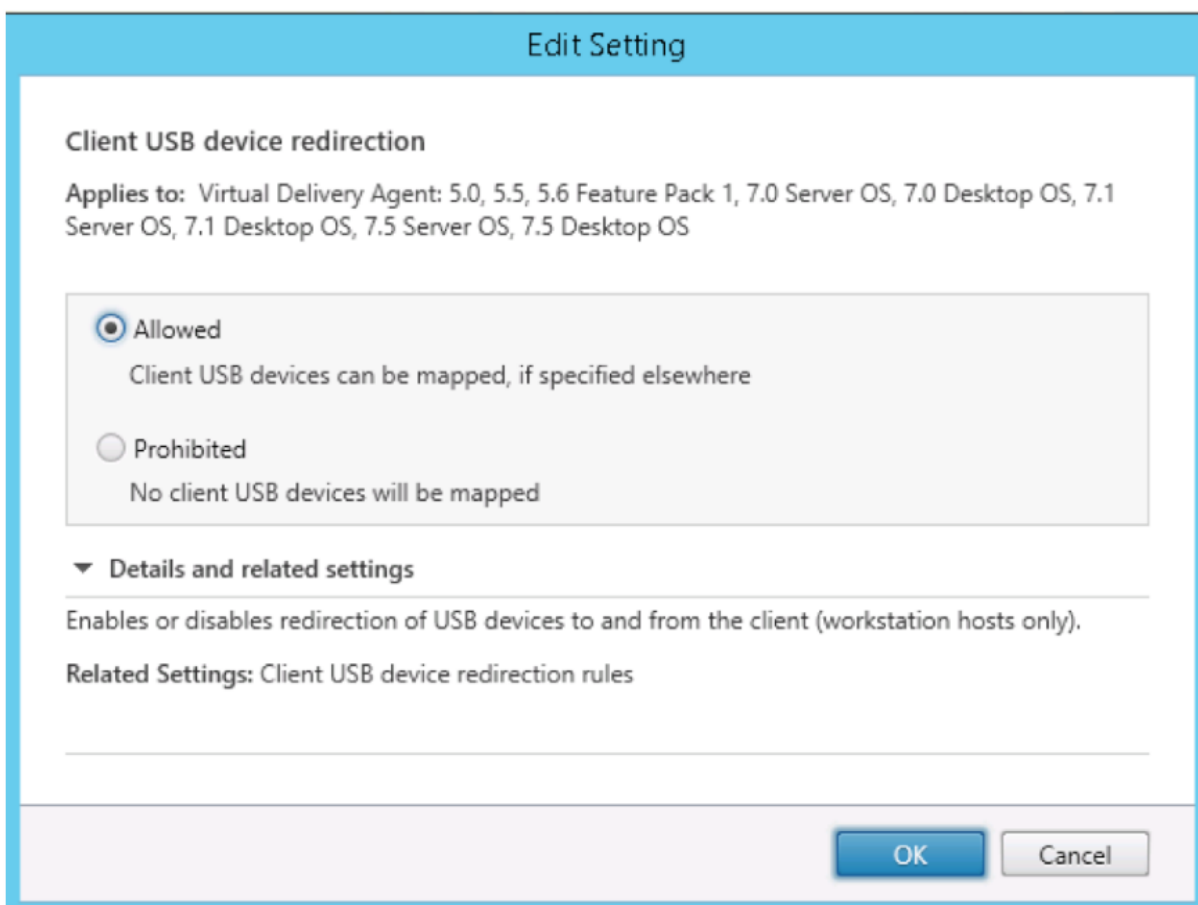
有一个 Citrix 策略控制是否启用或禁用 USB 设备重定向。还可以使用 Delivery Controller 策略指定设备类型。为 Linux VDA 配置 USB 设备重定向时，请配置以下策略和规则：

- 客户端 USB 设备重定向策略
- 客户端 USB 设备重定向规则

启用 **USB** 设备重定向 在 Citrix Studio 中，启用（或禁用）从客户端进行的 USB 设备重定向（仅限工作站主机）。

在编辑设置对话框中：

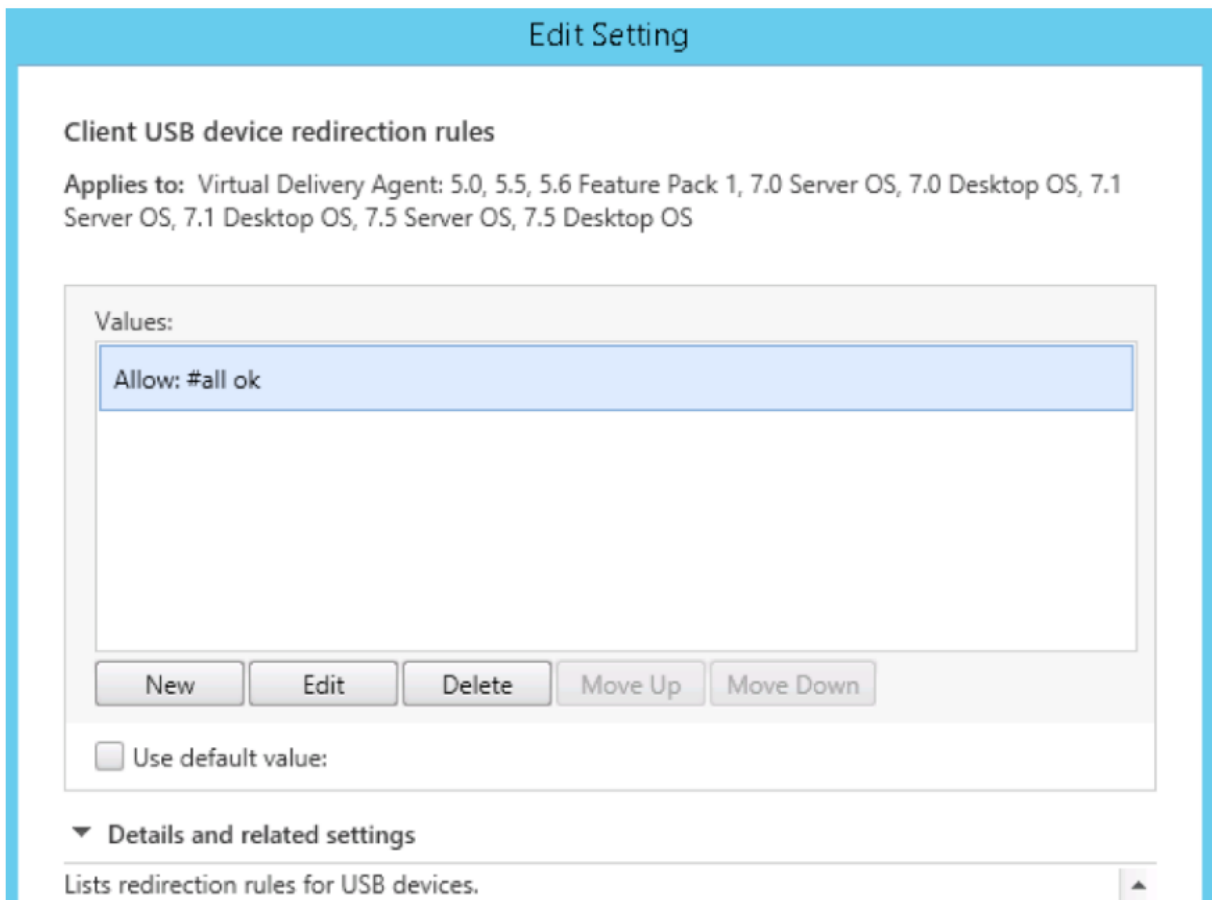
1. 选择允许。
2. 单击确定。



设置 **USB** 设备重定向规则 启用 USB 重定向策略后，使用 Citrix Studio 设置重定向规则，方法是指定允许（或拒绝）在 Linux VDA 上使用哪些设备。

在客户端 **USB** 设备重定向规则对话框中：

1. 单击新建添加重定向规则，或单击编辑检查现有规则。
2. 创建（或编辑）规则后，单击确定。



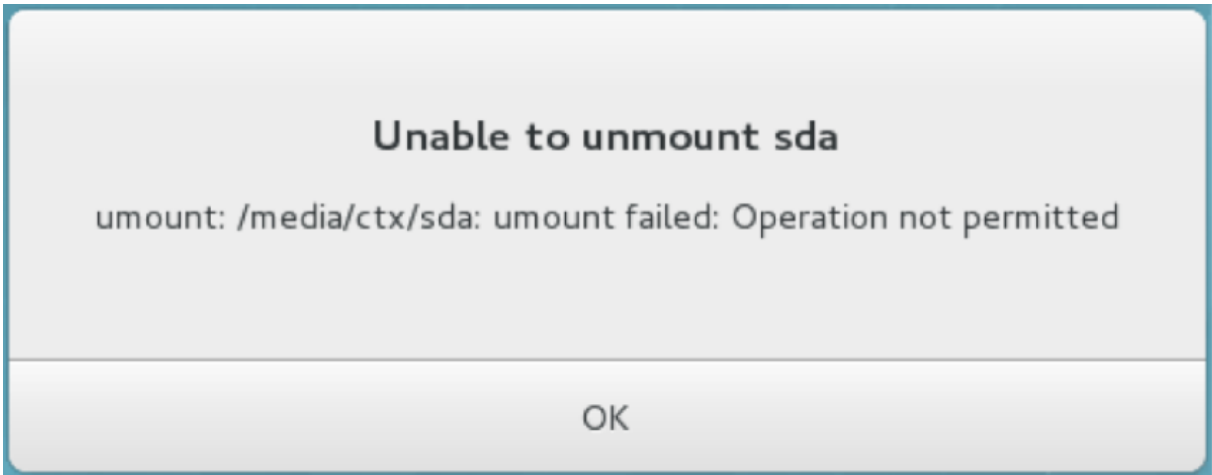
有关如何配置通用 USB 设备重定向的详细信息，请参阅 [Citrix Generic USB Redirection Configuration Guide](#) (《Citrix 通用 USB 重定向配置指南》)。

解决 **USB** 设备重定向问题

请根据本节中的信息解决您在使用 Linux VDA 时可能遇到的各种问题。

无法卸载重定向的 **USB** 磁盘

Linux VDA 使用管理权限管理从 Citrix Workspace 应用程序重定向的所有 USB 磁盘，以确保只有所有者能够访问重定向的设备。因此，您只能使用管理权限卸载设备。



停止重定向 **USB** 磁盘时文件丢失

如果使用 Citrix Workspace 应用程序的工具栏立即停止重定向 USB 磁盘，您在磁盘上修改或创建的文件可能会丢失。出现此问题是因为您将数据写入文件系统时，系统在文件系统中装载内存缓存。数据并未写入磁盘本身。如果使用 Citrix Workspace 应用程序的工具栏停止重定向，则没有时间将数据刷新至磁盘，从而导致数据丢失。

为了解决此问题，请先在终端使用 **sync** 命令将数据刷新至磁盘，然后再停止 USB 重定向。

Citrix Workspace 应用程序的工具栏中无设备

有时，您可能无法看到 Citrix Workspace 应用程序的工具栏中没有列出设备，这表示没有进行 USB 重定向。



如果遇到问题，请验证以下各项：

- 策略已配置为允许 USB 设备重定向。
- Citrix USB 服务模块正在运行。

如果策略未正确设置，请参考本文中的[设置 USB 设备重定向策略](#)部分进行更正。

如果 Citrix USB 服务模块未运行，请完成以下步骤：

1. 请使用以下命令检查 USB/IP 内核模块在您的 Linux 发行版中是否可用：

```
1 modinfo usbip-core
```

2. 如果输出如下所示，请根据您的 Linux 发行版安装或编译 USB/IP 内核模块：

```
1 modinfo: ERROR: Module usbip-core not found.
```

- 对于 Amazon Linux 2、CentOS、RHEL 和 Rocky Linux，请参阅本文中的[安装或编译 USB/IP 内核模块](#)部分。
- 对于 SUSE，请从 <https://software.opensuse.org/package/usbip> 下载并安装 USB/IP 软件包。
- 对于 Ubuntu/Debian，请完成以下步骤以编译和安装 USB/IP 内核模块：

a) 下载 USB/IP 内核模块源代码。

转至位于 <https://github.com/torvalds/linux/tree/master/drivers/usb/usbip> 的 Linux 内核存储库，选择目标 Linux 内核版本（v4.15 或更高版本）标记，然后获取链接，例如 <https://github.com/torvalds/linux/tree/v4.15/drivers/usb/usbip>。

转至 [DownGit](#) 并输入之前的链接，创建用于下载 USB/IP 源代码的下载链接。

b) 请使用以下命令解压源文件：

```
1 unzip ${
2   USBIP_SRC }
3   .zip
4
5 cd usbip
```

c) 按如下所示修改 **Makefile** 文件：

```
1 # SPDX-License-Identifier: GPL-2.0
2
3 ccflags-$(CONFIG_USBIP_DEBUG) := -DDEBUG
4
5 obj-$(CONFIG_USBIP_CORE) += usbip-core.o
6
7 usbip-core-y := usbip_common.o usbip_event.o
8
9 obj-$(CONFIG_USBIP_VHCI_HCD) += vhci-hcd.o
10
11 vhci-hcd-y := vhci_sysfs.o vhci_tx.o vhci_rx.o vhci_hcd.o
12
13 #obj-$(CONFIG_USBIP_HOST) += usbip-host.o
14
15 #usbip-host-y := stub_dev.o stub_main.o stub_rx.o stub_tx.o
16
17 #obj-$(CONFIG_USBIP_VUDC) += usbip-vudc.o
18
19 #usbip-vudc-y := vudc_dev.o vudc_sysfs.o vudc_tx.o vudc_rx.
   o vudc_transfer.o vudc_main.o
```

d) 编译源代码：

```
1 apt-get install linux-headers-`uname -r`
```

```
2  
3 make -C /lib/modules/`uname -r`/build M=$PWD
```

e) 安装 USB/IP 内核模块:

```
1 cp usbip-core.ko vhci-hcd.ko /opt/Citrix/VDA/lib64/
```

f) 重新启动 **ctxusbsd** 服务以加载 USB/IP 内核模块:

```
1 systemctl restart ctxusbsd
```

当 **USB** 设备在 **Citrix Workspace** 应用程序的工具栏中可见, 但被标记为“受策略限制”时, 重定向失败

出现此问题时, 请执行以下操作:

- 配置 Linux VDA 策略以启用重定向。
- 检查是否在 Citrix Workspace 应用程序的注册表中配置了任何其他策略限制。请检查注册表路径中的 **DeviceRules**, 以确保此设置未拒绝访问该设备:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA Client\GenericUSB
```

有关详细信息, 请参阅知识中心

文章 [How to Configure Automatic Redirection of USB Devices](#) (如何配置 USB 设备的自动重定向)。

USB 设备已成功重定向, 但无法在会话中使用

通常情况下, 只能重定向受支持的 **USB 设备**。其他设备可能也会重定向到活动 Linux VDA 会话。对于每个重定向的设备, 都会在系统 **/dev** 路径中创建用户拥有的节点。但是, 用户是否可以成功使用设备由驱动程序和配置决定。如果您发现拥有 (已插入) 的某个设备无法访问, 请将该设备添加到不受限制策略。

注意:

对于 USB 驱动器, Linux VDA 会配置和装载磁盘。用户 (且仅限安装它的所有者) 无需执行任何其他配置即可访问该磁盘。未包含在受支持的设备列表中的设备可能不是这种情况。

剪贴板重定向

July 15, 2024

剪贴板重定向允许您在 VDA 会话中运行的应用程序与在客户端设备上运行的应用程序之间复制和粘贴数据。

本文介绍了允许您实现剪贴板重定向的可用 Citrix 策略。

面向剪贴板重定向的 **Citrix** 策略

客户端剪贴板重定向

此设置允许或阻止将用户设备上的剪贴板映射到 VDA 上的剪贴板。

默认情况下，剪贴板重定向设置为允许。

要阻止剪贴数据在会话与本地剪贴板之间传输，请选择禁止。用户仍可以在会话中运行的应用程序之间复制和粘贴数据。

剪贴板重定向带宽限制

此设置指定在会话与本地剪贴板之间传输数据时允许使用的最大带宽 (kbps)。

剪贴板重定向带宽限制百分比

此设置指定在会话与本地剪贴板之间传输数据时允许使用的最大带宽 (占总会话带宽的百分比)。

将剪贴板客户端限制为会话传输大小

此设置指定单个复制粘贴操作可以从客户端设备传输到虚拟会话的剪贴板数据的最大大小。

要限制剪贴板传输大小，请启用将剪贴板客户端限制为会话传输大小设置。然后，在大小限制字段中，输入一个以千字节为单位的值，以定义本地剪贴板与会话之间的数据传输大小。

默认情况下，此设置处于禁用状态，并且对客户端到会话的传输没有限制。

将剪贴板会话限制为客户端传输大小

此设置指定单个复制粘贴操作可以从虚拟会话传输到客户端设备的剪贴板数据的最大大小。

要限制剪贴板传输大小，请启用将剪贴板会话限制为客户端传输大小设置。然后，在大小限制字段中，输入一个以千字节为单位的值，以定义会话与本地剪贴板之间的数据传输大小。

默认情况下，此设置处于禁用状态，并且对到客户端的会话传输没有限制。

“限制客户端剪贴板写入”和“客户端剪贴板写入允许的格式”

启用这两个设置可以允许您将特定的数据格式从会话复制并粘贴到客户端 (写入到客户端)。

以下剪贴板格式是系统定义的格式：

- CF_TEXT

- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

“限制会话剪贴板写入”和“会话剪贴板写入允许的格式”

启用这两个设置可以允许您将特定的数据格式从客户端复制并粘贴到会话（写入到会话）。

以下剪贴板格式是系统定义的格式：

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_UNICODETEXT
- CF_LOCALE
- CF_DIBV5
- CF_HDROP

键盘

July 15, 2024

本部分内容包含以下主题：

- [客户端 IME](#)
- [客户端 IME 用户界面同步](#)
- [动态键盘布局同步](#)
- [软键盘](#)

- [支持多语言输入](#)

X 键盘扩展 (XKB) 配置

August 21, 2024

从 2407 版本开始，Linux VDA 将 evdev 指定为键盘配置的默认 XKB 规则。此决策有以下几个理由：

- 大多数现代 Linux 发行版默认使用 evdev XKB 规则。
- 几乎所有应用程序都遵循 evdev XKB 规则的按键代码模式。
- evdev XKB 规则提供了扩展的支持，除了基本字母数字字符和典型功能按键之外，还支持大量的其他按键符号，例如“取消”、“重做”、“撤消”、“xf86Copy”、“XF86Open”和“XF86Paste”
- Linux 中的 evdev XKB 规则解决了各种与键盘相关的问题，尤其是那些与对按键输入的错误响应有关的问题。

如果您想切换回 XFree86 规则，您可以进行以下注册表设置：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "XkbRuleName" -d "xorg"
```

在此命令中，只有区分大小写的精确文本“xorg”被识别为“xfree86”规则集；任何其他输入值都无效。

客户端输入法编辑器 (IME)

July 15, 2024

概述

必须通过 IME 输入双字节字符，例如中文、日语和朝鲜语字符。通过与客户端上的 Citrix Workspace 应用程序兼容的任何 IME（例如 Windows 本机 CJK IME）键入此类字符。

安装

在安装 Linux VDA 时，会自动安装此功能。

使用情况

照常打开 Citrix Virtual Apps 或 Citrix Virtual Desktops 会话。

根据需要在客户端上更改您的输入法以开始使用客户端 IME 功能。

已知问题

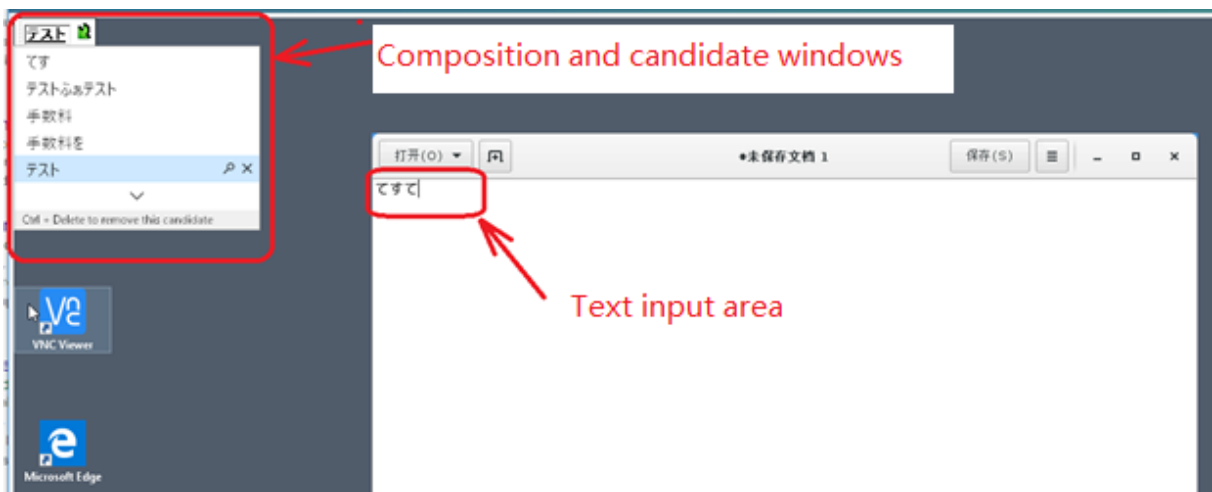
- 必须先双击 Google 电子表格中的单元格，才能使用客户端 IME 功能在单元格中键入字符。
- 不会在“密码”字段中自动禁用客户端 IME 功能。
- IME 用户界面不在输入区域中跟随光标。

客户端 IME 用户界面同步

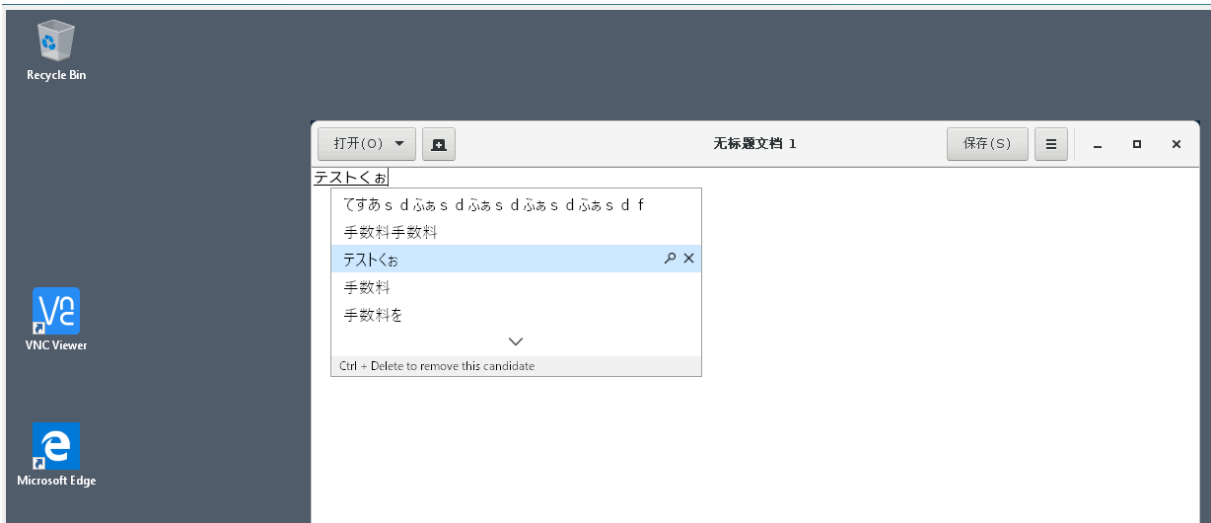
July 15, 2024

概述

迄今为止，客户端 IME 用户界面（包括撰写窗口和候选窗口）置于屏幕左上角。以前不跟随光标，有时距离文本输入区域中的光标较远：



Citrix 增强了可用性，并进一步提高了客户端 IME 的用户体验，如下所示：



使用此功能需满足的必备条件

1. 在 Linux VDA 上启用 Intelligent Input Bus (IBus)。有关如何在 Linux 操作系统上启用 IBus 的信息，请参阅操作系统供应商提供的文档。例如：
 - Ubuntu: <https://help.ubuntu.com/community/ibus>
 - CentOS、RHEL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/7.0_release_notes/sect-red_hat_enterprise_linux-7.0_release_notes-internationalization-input_methods
 - Debian: <https://wiki.debian.org/I18n/ibus>
 - SUSE: <https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-gnome-settings.html#sec-gnome-settings-lang>
2. 该功能将自动安装，但您必须先启用该功能才能使用它。

启用和禁用该功能

默认情况下禁用客户端 IME 用户界面同步功能。要启用或禁用此功能，请设置客户端键盘布局同步和 **IME** 改进功能策略，或通过 `ctxreg` 实用程序编辑注册表。

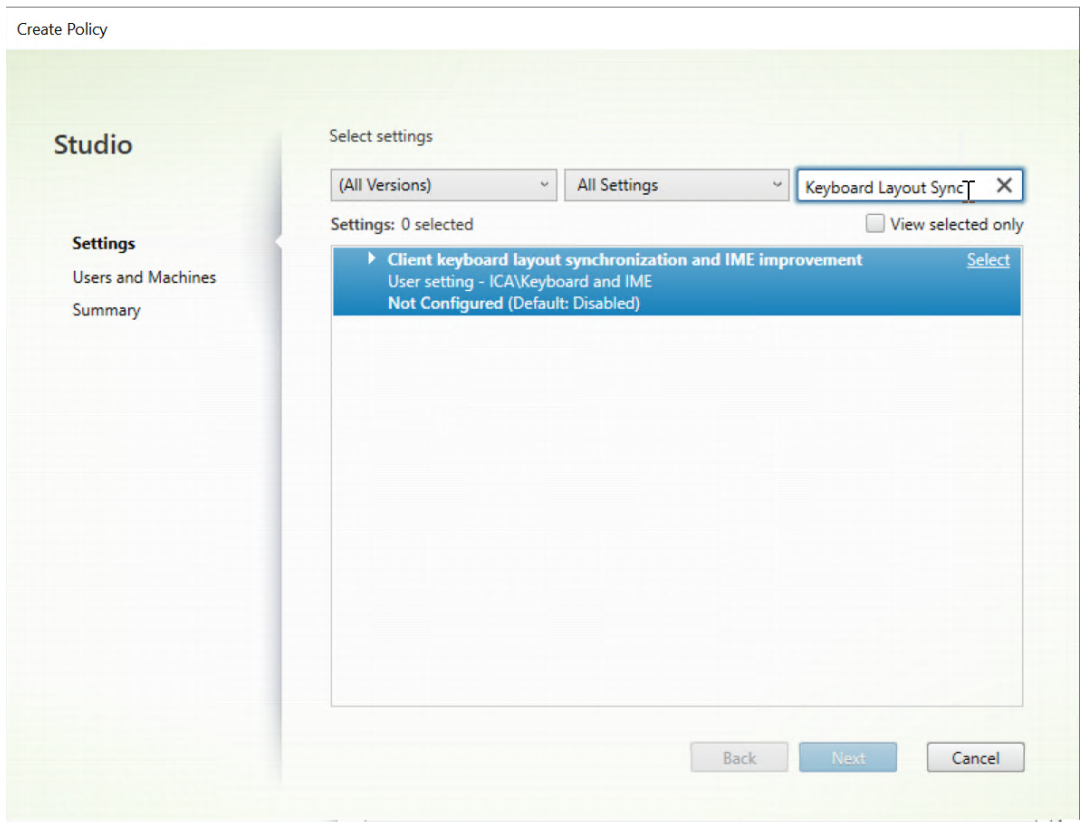
注意：

客户端键盘布局同步和 **IME** 改进功能策略的优先级高于注册表设置，可以应用到您指定的用户和计算机对象或站点中的所有对象。给定 Linux VDA 上的注册表设置应用到该 VDA 上的所有会话。

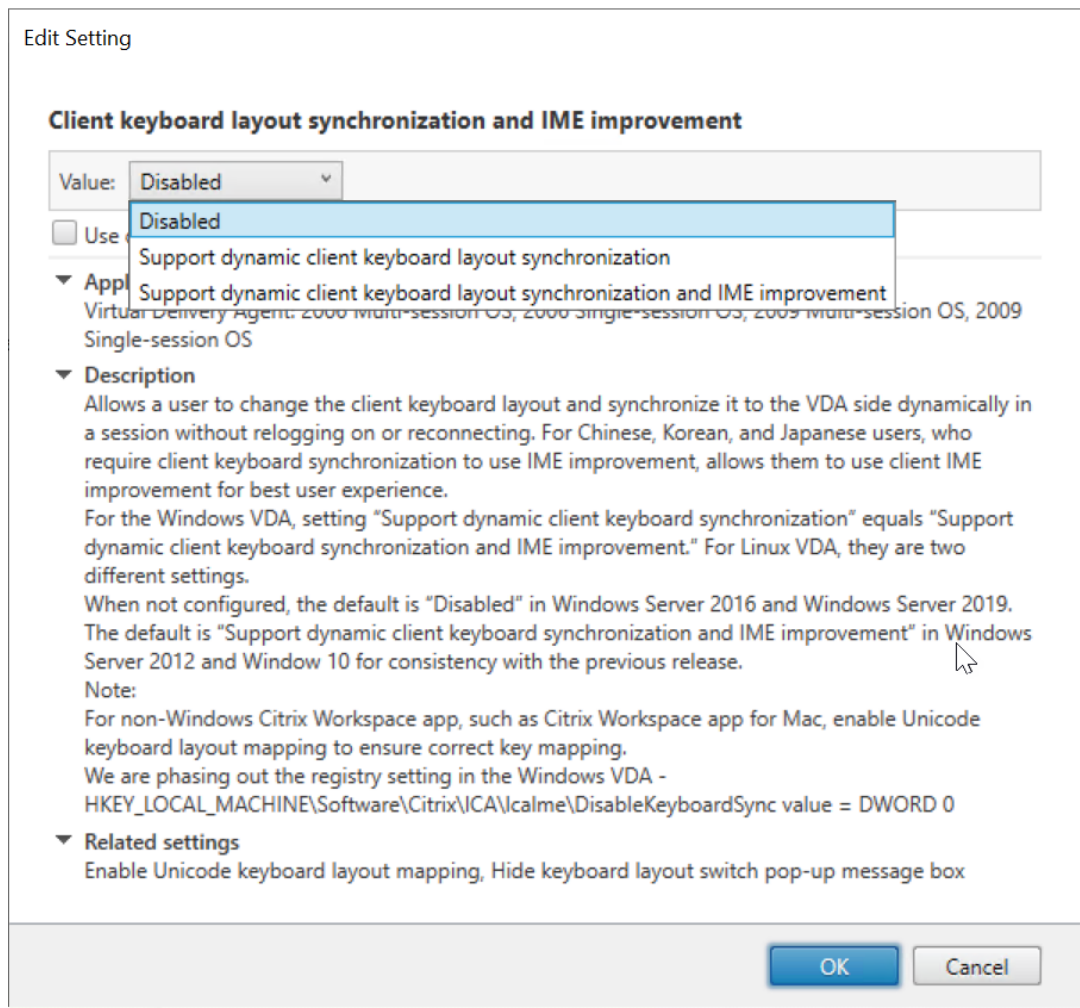
- 设置客户端键盘布局同步和 **IME** 改进功能策略以启用或禁用客户端 IME 用户界面同步功能：

1. 在 Studio 中，右键单击策略，然后选择创建策略。

2. 搜索客户端键盘布局同步和 **IME** 改进功能策略。



3. 单击策略名称旁边的选择。
4. 设置策略。



共有三个可用选项：

- 已禁用：禁用动态键盘布局同步和客户端 IME 用户界面同步。
 - 支持动态客户端键盘布局同步：启用动态键盘布局同步，而不考虑位于 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` 的 **SyncKeyboardLayout** 注册表项的 DWORD 值。
 - 支持动态客户端键盘布局同步和 **IME** 改进功能：启用动态键盘布局同步和客户端 IME 用户界面同步，而不考虑位于 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` 的 **SyncKeyboardLayout** 和 **SyncClientIME** 注册表项的 DWORD 值。
- 通过 `ctxreg` 实用程序编辑注册表以启用或禁用客户端 IME 用户界面同步功能：

要启用该功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncClientIME" -d "0x00000001"
```

要禁用该功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncClientIME" -d "0x00000000"
```

动态键盘布局同步

July 15, 2024

以前，Linux VDA 和客户端设备上的键盘布局必须相同。例如，当客户端设备上的键盘布局从“英语”更改为“法语”，但在 VDA 上未更改时，可能会出现按键映射问题。

Citrix 通过自动将 VDA 的键盘布局与客户端设备的键盘布局同步来解决此问题。无论何时客户端设备上的键盘布局发生变化，VDA 随后都会做出恰当的调整。

注意：

适用于 HTML5 的 Citrix Workspace 应用程序不支持动态键盘布局同步功能。

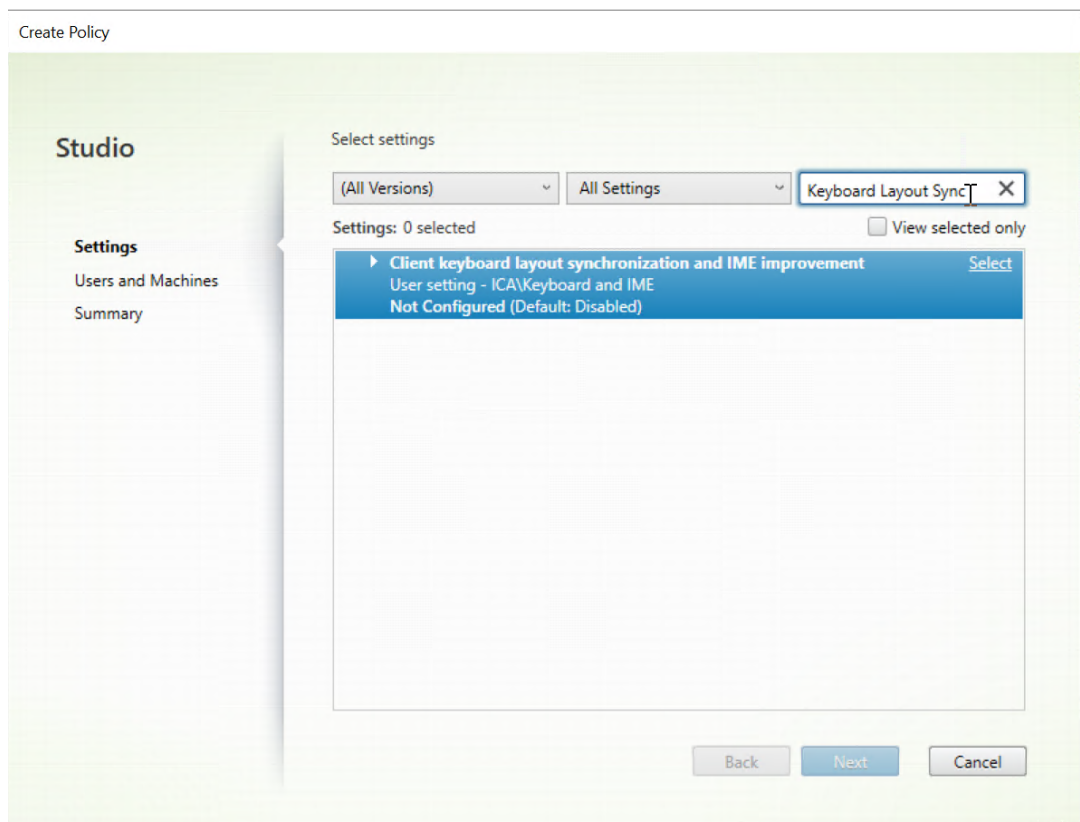
配置

默认情况下，动态键盘布局同步功能处于禁用状态。要启用或禁用此功能，请设置客户端键盘布局同步和 **IME** 改进功能策略，或通过 `ctxreg` 实用程序编辑注册表。

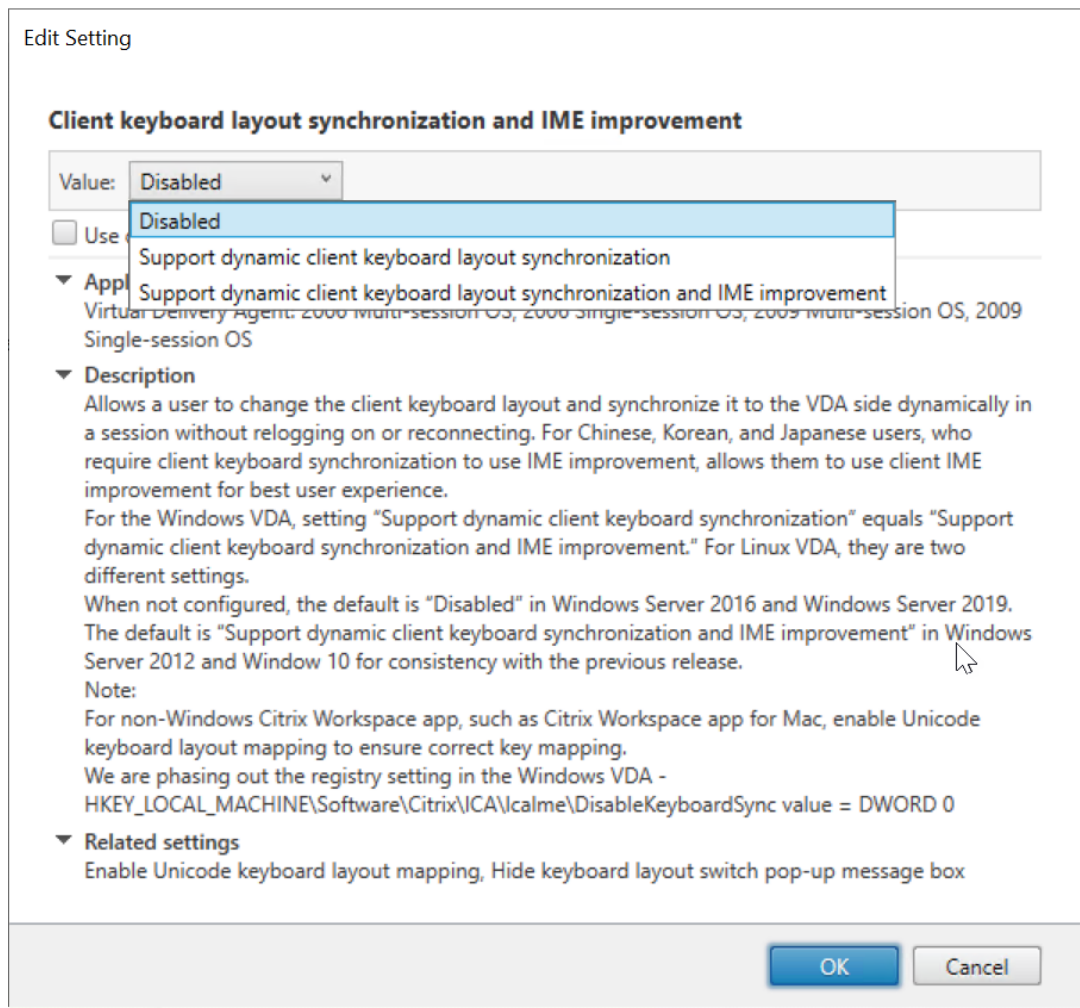
注意：

客户端键盘布局同步和 **IME** 改进功能策略的优先级高于注册表设置，可以应用到您指定的用户和计算机对象或站点中的所有对象。给定 Linux VDA 上的注册表设置应用到该 VDA 上的所有会话。

- 设置客户端键盘布局同步和 **IME** 改进功能策略以启用或禁用动态键盘布局同步功能：
 1. 在 Studio 中，右键单击策略，然后选择创建策略。
 2. 搜索客户端键盘布局同步和 **IME** 改进功能策略。



3. 单击策略名称旁边的选择。
4. 设置策略。



共有三个可用选项：

- 已禁用：禁用动态键盘布局同步和客户端 IME 用户界面同步。
 - 支持动态客户端键盘布局同步：启用动态键盘布局同步，而不考虑位于 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` 的 **SyncKeyboardLayout** 注册表项的 DWORD 值。
 - 支持动态客户端键盘布局同步和 **IME** 改进功能：启用动态键盘布局同步和客户端 IME 用户界面同步，而不考虑位于 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar` 的 **SyncKeyboardLayout** 和 **SyncClientIME** 注册表项的 DWORD 值。
- 通过 `ctxreg` 实用程序编辑注册表以启用或禁用动态键盘布局同步功能：

要启用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncKeyboardLayout" -d "0x00000001"
```


要禁用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Citrix\LanguageBar" -v "SyncKeyboardLayout" -d "0x00000000"
```

使用情况

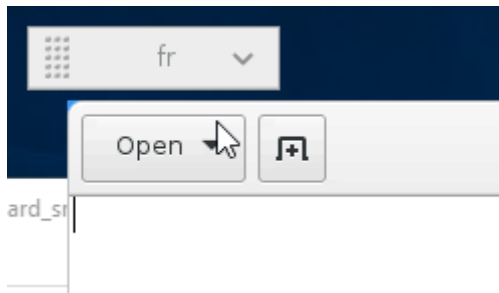
启用此功能后，如果会话过程中客户端设备上的键盘布局发生变化，会话的键盘布局也将相应地发生变化。

例如，如果将客户端设备上的键盘布局更改为“法语 (FR)”：

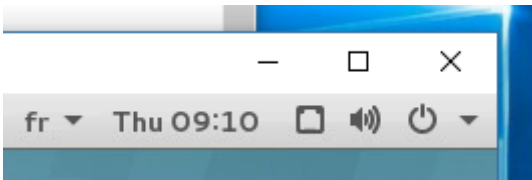


Linux VDA 会话的键盘布局随后也将更改为“fr”。

在应用程序会话中，如果启用了语言栏，则可以看到这一自动变化情形：



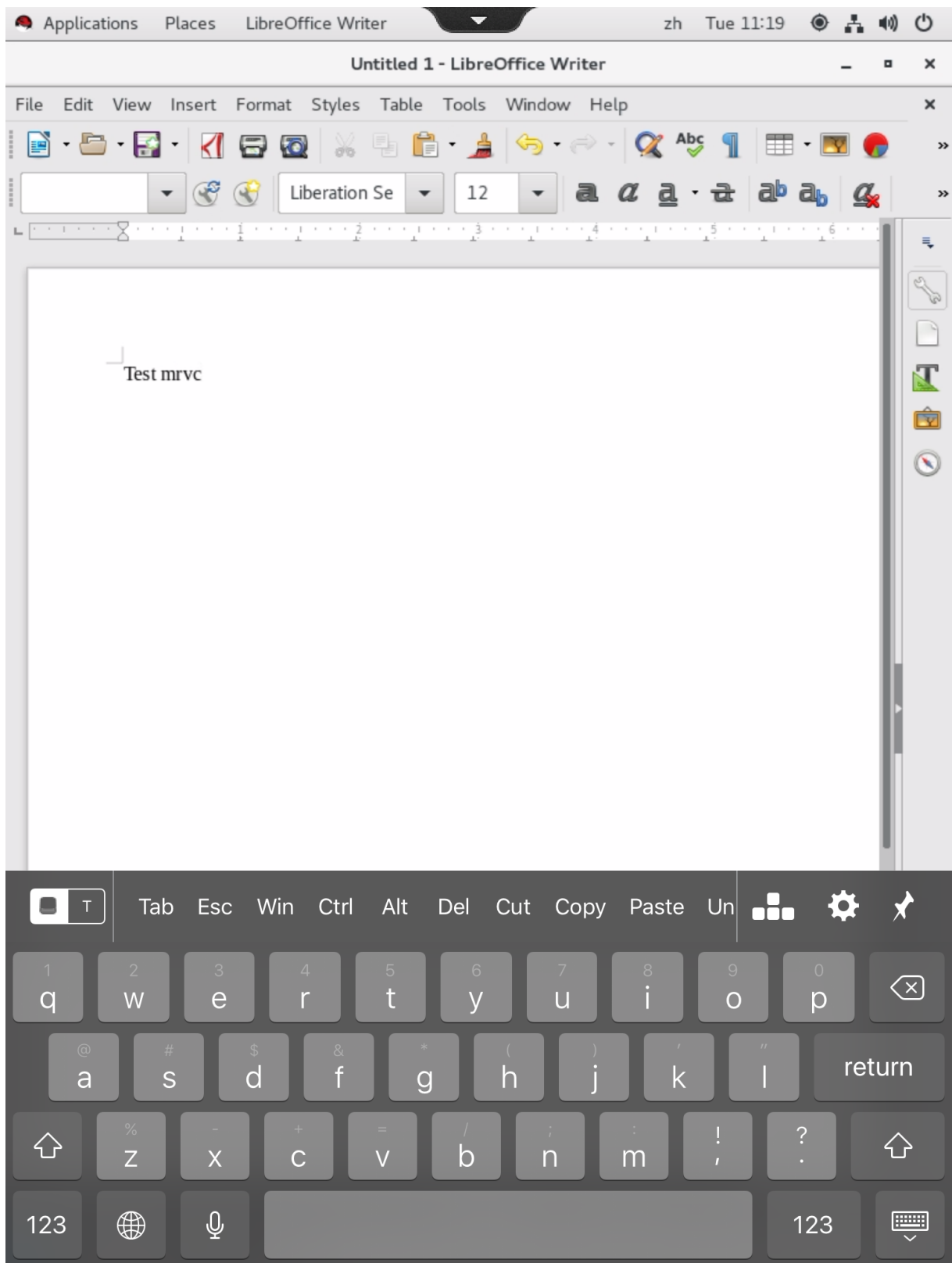
在桌面会话中，可以在任务栏中看到这一自动发生的变化：



软键盘

July 15, 2024

可以在 Linux 虚拟桌面或应用程序会话中使用软键盘功能。软键盘会在您输入或离开输入字段时显示或隐藏。



注意：

适用于 iOS 和 Android 的 Citrix Workspace 应用程序支持该功能。

启用和禁用该功能

默认情况下，该功能处于禁用状态。可以使用 **ctxreg** 实用程序启用或禁用该功能。给定 Linux VDA 上的功能配置应用于该 VDA 上发布的所有会话。

要启用此功能，请执行以下操作：

1. 请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc" -v "Enabled" -d "0x00000001"
```

2. 在 Citrix Studio 中，将自动显示键盘策略设置为允许。
3. (可选) 对于 RHEL 7 和 CentOS 7，运行以下命令将智能输入总线 (IBus) 配置为默认 IM 服务：

```
1 echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
```

要禁用该功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\Control\Citrix\VirtualChannels\MrVc" -v "Enabled" -d "0x00000000"
```

注意：

之前的设置将在您登录到新会话或注销后返回到当前会话时生效。

限制

- 该功能在 Google Chrome、LibreOffice 和其他应用程序中可能无法正常运行。
- 要在手动隐藏软键盘后再次显示它，请单击非输入字段，然后再次单击当前输入字段。
- 在 Web 浏览器中从一个输入字段切换到单击另一个输入字段时，软键盘可能不显示。要解决此问题，请单击非输入字段，然后单击目标输入字段。
- 此功能不支持 Unicode 字符和双字节字符（例如，中文、日语和韩语字符）。
- 软键盘不适用于密码输入字段。
- 软键盘可能会与当前输入字段重叠。在这种情况下，请移动应用程序窗口或向上滚动屏幕以将输入字段移到可访问的位置。
- 由于 Citrix Workspace 应用程序和 Huawei 平板电脑之间存在兼容性问题，因此即使连接了物理键盘，软键盘也会显示在 Huawei 平板电脑上。

支持多语言输入

August 20, 2024

自 Linux VDA 版本 1.4 起，Citrix 添加了对已发布的应用程序的支持。用户可以在没有 Linux 桌面环境的情况下访问所需的 Linux 应用程序。

但是，由于语言栏与 Linux 桌面环境高度集成，因此，Linux VDA 上的本地语言栏对于已发布的应用程序不可用。因此，用户无法以需要 IME 的语言（例如，中文、日语或韩语）输入文本。在应用程序会话期间，用户也不能在键盘布局之间切换。

为了解决这些问题，此功能为接受文本输入的已发布应用程序提供语言栏。通过语言栏，用户可以在应用程序会话期间选择服务器端 IME 以及在键盘布局之间切换。

配置

您可以使用 **ctxreg** 实用程序启用或禁用此功能（默认情况下禁用）。给定 Linux VDA 服务器上的功能配置应用于在该 VDA 上发布的所有应用程序。

配置注册表项为“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar”，类型为 DWORD。

要启用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
   CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
   x00000001"
```

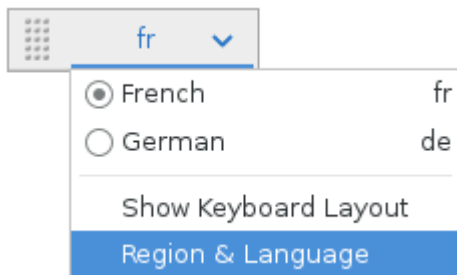
要禁用此功能，请运行以下命令：

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
   CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
   x00000000"
```

使用情况

用法很简单。

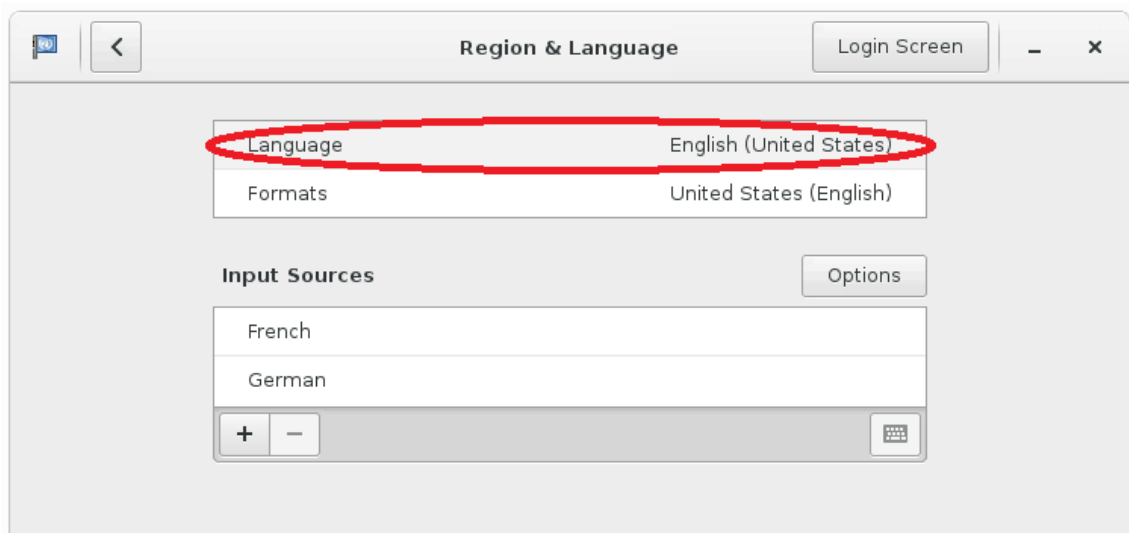
1. 启用功能。
2. 访问可以接受文本输入的已发布的应用程序。在会话中应用程序旁边会显示语言栏。
3. 从下拉菜单中选择区域和语言以添加所需的语言（输入源）。



4. 从下拉菜单中选择 IME 或键盘布局。
5. 使用所选的 IME 或键盘布局键入语言。

注意：

- 在 VDA 端语言栏上更改键盘布局时，请确保在运行 Citrix Workspace 应用程序的客户端上使用相同的键盘布局。
- 必须将 **accountsservice** 软件包升级到版本 0.6.37 或更高版本，才能在区域和语言对话框中执行设置。



多媒体

July 15, 2024

本部分内容包含以下主题：

- [音频功能](#)
- [浏览器内容重定向](#)

- [HDX 网络摄像机视频压缩](#)

音频功能

August 20, 2024

自适应音频

自适应音频默认处于启用状态。它支持以下 Citrix Workspace 应用程序客户端：

- 适用于 Windows 的 Citrix Workspace 应用程序 - 2109 及更高版本
- 适用于 Linux 的 Citrix Workspace 应用程序 - 2109 及更高版本
- 适用于 Mac 的 Citrix Workspace 应用程序 - 2109 及更高版本

当您使用列表中未包括的客户端时，自适应音频会回退到旧版音频。

自适应音频会根据网络条件动态调整音频采样比特率以增强音频体验，因而无需在 VDA 上手动配置 [音频质量策略](#)。

下表显示了自适应音频与旧版音频之间的比较结果：

自适应音频	旧版音频
最大音频采样率：48 kHz	最大音频采样率：8 kHz
立体声声道	单声道

音频服务注意事项

从版本 2407 开始，Linux VDA 在 Debian 12.5、RHEL 9.x 和 Rocky Linux 9.x 中支持 PipeWire，在这些版本中，它是默认的音频服务。对于其他发行版，请继续使用 PulseAudio。

PulseAudio

- 在 RHEL 8.x 和 Rocky Linux 8.x 上使用 PulseAudio 13.99 或更高版本。

PipeWire

Linux VDA 仅在 Debian 12.5、RHEL 9.x 和 Rocky Linux 9.x 中支持 PipeWire。下面是需要考虑的一些注意事项：

- 如果您在 Debian 12.5、RHEL 9.x 或 Rocky Linux 9.x 上安装了 Linux VDA 版本 2407 或更高版本，那么您使用的是 PipeWire。
- 如果您在 RHEL 9.x 或 Rocky Linux 9.x 上安装了 2407 之前的 Linux VDA 版本，那么您使用的是 PulseAudio。要继续使用 PulseAudio，请什么都不要做。要切换到 PipeWire，请完成以下步骤：

1. 将 PulseAudio 软件包替换为 pipewire-pulseaudio 软件包：

```
1 dnf swap --allowerase pulseaudio pipewire-pulseaudio
```

2. 在全局范围内启用 Pipewire 相关服务：

```
1 systemctl --global --now enable pipewire pipewire-pulse
   pipewire-pulse.socket wireplumber
```

3. 重新启动 Linux VDA。

音频的容损模式（技术预览版）

容损模式支持音频。当用户通过延迟和数据包丢失率均较高的网络进行连接时，此功能改进了实时流技术推送的用户体验并提高了通过 EDT 传输的音频质量。

有关容损模式和 EDT 的详细信息，请参阅 Citrix Virtual Apps and Desktops 文档中的[其他信息](#)。

为音频功能启用容损模式

默认情况下，音频的容损模式处于禁用状态。要启用它，请完成下面的两个步骤：

1. 通过设置 [HDX 自适应传输](#) 策略启用自适应传输。默认情况下启用自适应传输。
2. 通过设置 [自适应音频](#) 策略启用自适应音频。自适应音频默认处于启用状态。
3. 通过以下注册表设置启用容损模式（EDT 不可靠传输）：

```
1 ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\
   VirtualChannels\Audio" -t "REG_DWORD" -v "fEdtUnreliableAllowed"
   -d "0x00000001" --force
```

客户端要求和设置

要对音频使用容损模式，请确保您的 Citrix Workspace 应用程序支持并配置为启用此功能；否则，音频默认为 EDT 可靠传输。

下面是支持容损模式的最低 Citrix Workspace 应用程序版本：

- 适用于 Windows 的 Citrix Workspace 应用程序最低版本 2309

- 适用于 Linux 的 Citrix Workspace 应用程序最低版本 2311
- 适用于 Mac 的 Citrix Workspace 应用程序最低版本 2311

用于改善音频性能的数据包丢失隐藏（技术预览版）

自 2407 版本起，抖动缓冲机制得到了改进，并为自适应音频编解码器添加了数据包丢失隐藏 (PLC) 功能。PLC 有助于重新构建丢失的数据包。此增强功能有助于改进数据包容损和抖动容限，因而提高了音频的容损模式（EDT 有损）的音频性能。

默认情况下，此功能处于禁用状态。要启用此功能，请完成以下步骤：

1. 为音频功能启用容损模式。
2. 启用允许支持多个音频设备的音频重定向功能。

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -t "
  REG_DWORD" -v "fEnableAudioRedirectionV4" -d "0x00000001" --
  force
```

3. 要增强容损模式下的音频性能，请为音频启用“读取/写入 V2”：

```
1 ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\
  VirtualChannels\Audio" -t "REG_DWORD" -v "fEnableReadWriteV2" -
  d "0x00000001" --force
```

支持多个音频设备（预览版）

概述

自版本 2311 起，Linux VDA 引入了音频重定向功能。该功能允许将安装了 Citrix Workspace 应用程序的客户端计算机上的多个音频设备重定向到远程 Linux VDA 会话。

启用此功能后：

- 客户端计算机上的所有本地音频设备都显示在会话中。随音频设备显示的不是 CitrixAudioSink（音频输出）或 CitrixAudioSource（音频输入），而是各自的设备名称。可以在会话中的应用程序中选择音频设备，也可以在会话期间使用默认音频设备，该设备也是客户端计算机的默认音频设备。如有必要，您可以从客户端计算机的系统设置中更改默认音频设备。更新客户端计算机的默认音频设备后，新设备将作为默认音频设备出现在会话中。
- 当您插入或移除会话中的音频设备时，这些音频设备会动态更新。

配置

要使用该功能，请在 Linux VDA 上将其启用，然后选择支持的 Citrix Workspace 应用程序。

在 **Linux VDA** 上启用该功能 默认情况下，允许多个音频设备支持的音频重定向功能处于禁用状态。要将其启用，请在 Linux VDA 上运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "
  fEnableAudioRedirectionV4" -t BIN -d "1"
```

要禁用或重新启用该功能，请分别运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "
  fEnableAudioRedirectionV4" -d "0"
```

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
  CurrentControlSet\Control\Citrix\VirtualChannels\Audio" -v "
  fEnableAudioRedirectionV4" -d "1"
```

客户端要求和设置 该功能仅支持以下客户端：

- 适用于 Windows 的 Citrix Workspace 应用程序
- 适用于 Linux 的 Citrix Workspace 应用程序最低版本 2212
- 适用于 HTML5 的 Citrix Workspace 应用程序最低版本 2306
- 适用于 Chrome 的 Citrix Workspace 应用程序最低版本 2306
- 适用于 Mac 的 Citrix Workspace 应用程序最低版本 2311

需要在 Citrix Workspace 应用程序中进行正确的设置才能使该功能按预期运行。有关详细信息，请参阅 [Citrix Workspace 应用程序](#) 文档。

已知问题

由于 PulseAudio 存在 [问题](#)，因此，在 Ubuntu 22.04 会话中，尝试在音频设备之间切换可能会失败。要解决此问题，请从 VDA 中删除当前会话用户的 PulseAudio 配置，然后重新打开会话。要删除 PulseAudio 配置，请运行 `$ rm -r ~/.config/pulse` 命令。

浏览器内容重定向

July 15, 2024

概述

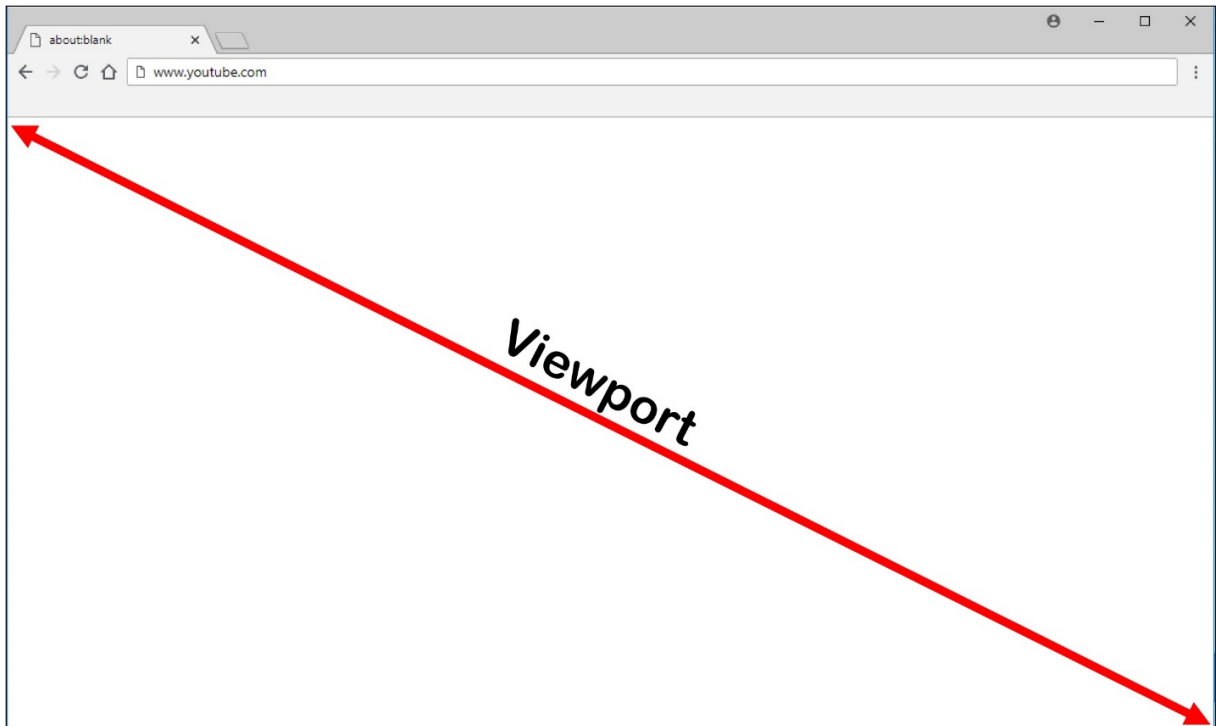
浏览器内容重定向提供了在客户端呈现允许列表中的 Web 页面的功能。此功能会使用 Citrix Workspace 应用程序在客户端实例化相应的呈现引擎，该引擎会从 URL 提取 HTTP 和 HTTPS 内容。

注意：

Linux VDA 在 Google Chrome 中支持浏览器内容重定向。

此叠加 Web 布局引擎在客户端上运行，而非在 VDA 上运行，并且使用客户端 CPU、GPU、RAM 和网络。

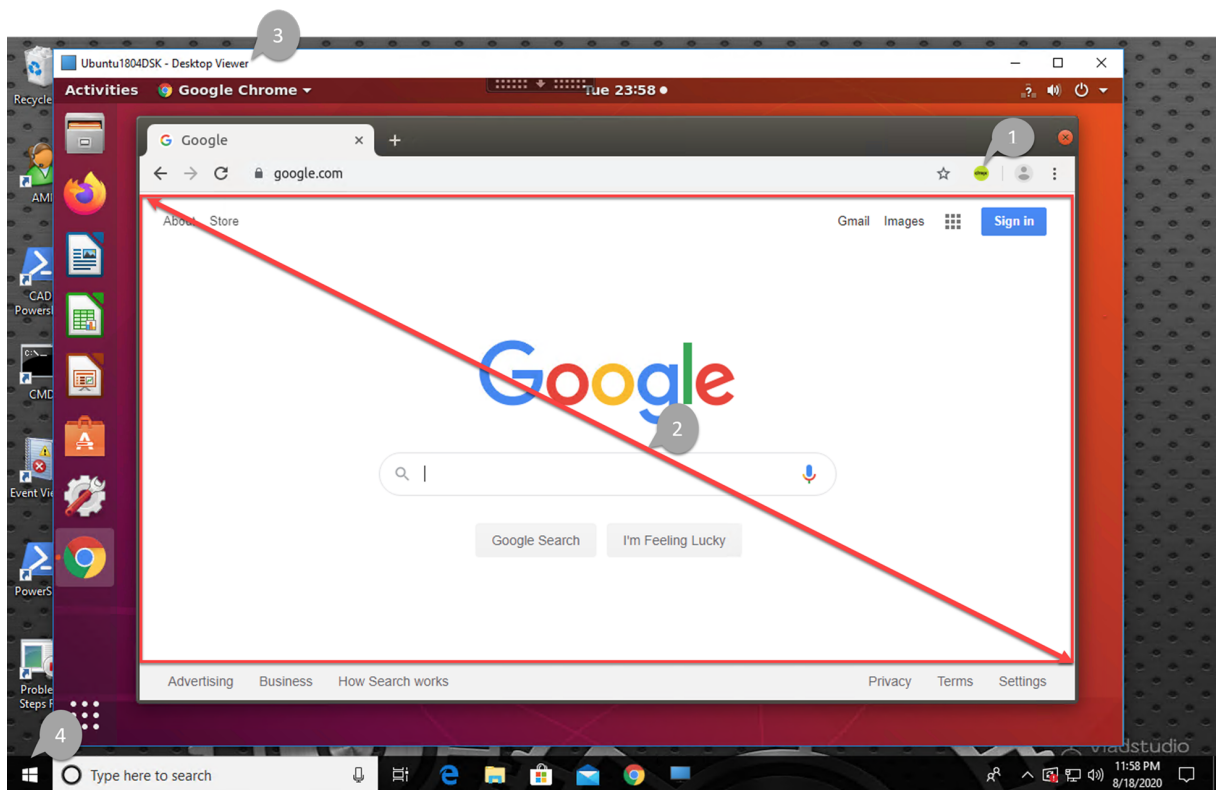
只有浏览器视口会进行重定向。视口是浏览器中显示内容的矩形区域。视口不包括地址栏、收藏夹栏和状态栏等项目。这些项目仍在 VDA 上的浏览器中运行。



配置 Studio 策略，用于指定包含可重定向的允许列表中的 URL 的访问控制列表。配置用于禁用特定 URL 重定向的阻止列表。

如果在允许列表中找到与 URL 匹配的 URL，但在任何阻止列表中均未找到，虚拟通道 (CTXCSB) 会指示 Citrix Workspace 应用程序需要重定向并中继 URL。然后，Citrix Workspace 应用程序会实例化一个本地呈现引擎并显示此 Web 站点。

之后，Citrix Workspace 应用程序会将此 Web 站点无缝融入虚拟桌面浏览器内容区域中。



1. Citrix 浏览器内容重定向扩展程序的图标

扩展程序图标的颜色指定 Chrome 扩展程序的状态。其颜色为以下三种颜色之一：

- 绿色：活动并连接
- 灰色：在当前选项卡上不活动/空闲
- 红色：已损坏/不运行

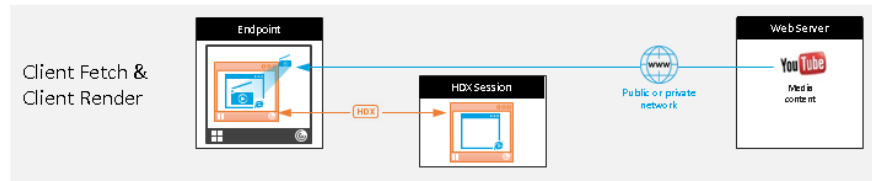
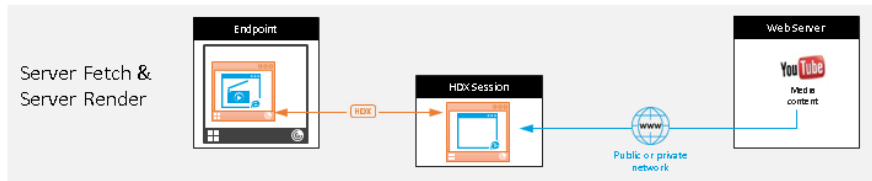
2. 视口在客户端上呈现或混合回虚拟桌面

3. Linux VDA

4. Windows 客户端

下面是 Citrix Workspace 应用程序提取内容的方式的几种情况：

Redirection scenarios



Benefits:

- Better end user experience (Adaptive Bit Rate (ABR))
- Reduced VDA resource usage (CPU/RAM/IO)
- Reduced bandwidth consumption

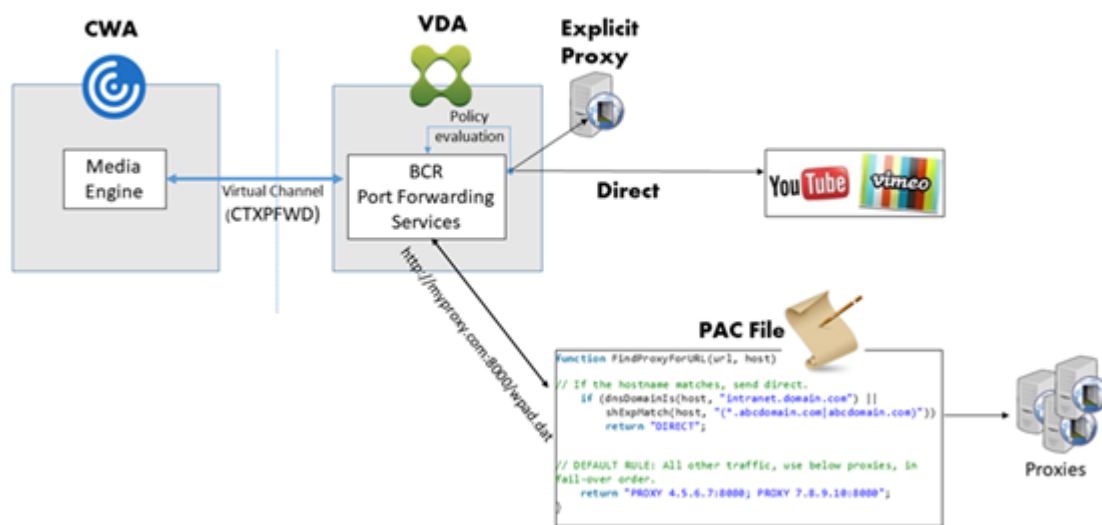
- 服务器提取和服务器呈现：由于没有将站点添加到允许列表或重定向失败，因此没有重定向。我们将回退到在 VDA 上呈现 Web 页面，并使用 Thinwire 来远程显示图形。使用策略来控制回退行为。这种情况会导致 VDA 上的 CPU、RAM 和带宽消耗较高。
- 服务器提取和客户端呈现：Citrix Workspace 应用程序使用虚拟通道 (CTXPFW) 通过 VDA 连接 Web 服务器并从中提取内容。当客户端无权访问 Web 服务器（例如，瘦客户端）时，此选项非常有用。它会降低 VDA 上的 CPU 和 RAM 占用量，但在 ICA 虚拟通道中占用带宽。

此方案有三种操作模式。CTXPFW 将数据转发到 VDA 访问的代理设备以获得 Web 服务器的访问权限。

可选择的策略选项：

- 显式代理 - 如果您的数据中心中有单个显式代理。
- 直接或透明 - 如果没有代理或者如果使用透明代理。
- PAC 文件 - 如果您依赖 PAC 文件，以便 VDA 中的浏览器可以自动选择适当的代理服务器来获取指定的 URL。

有关详细信息，请参阅本文后面的浏览器内容重定向代理配置设置。



- 客户端提取和客户端呈现：由于 Citrix Workspace 应用程序直接连接 Web 服务器，因此需要访问 Internet。在这种情况下，会从 Citrix Virtual Apps and Desktops 站点卸载所有网络、CPU 和 RAM 使用量。

优势：

- 更出色的最终用户体验（自适应比特率 (ABR)）
- 降低了 VDA 资源使用量 (CPU/RAM/IO)
- 降低了带宽占用量

系统要求

Windows 客户端：

- 适用于 Windows 的 Citrix Workspace 应用程序 1809 或更高版本

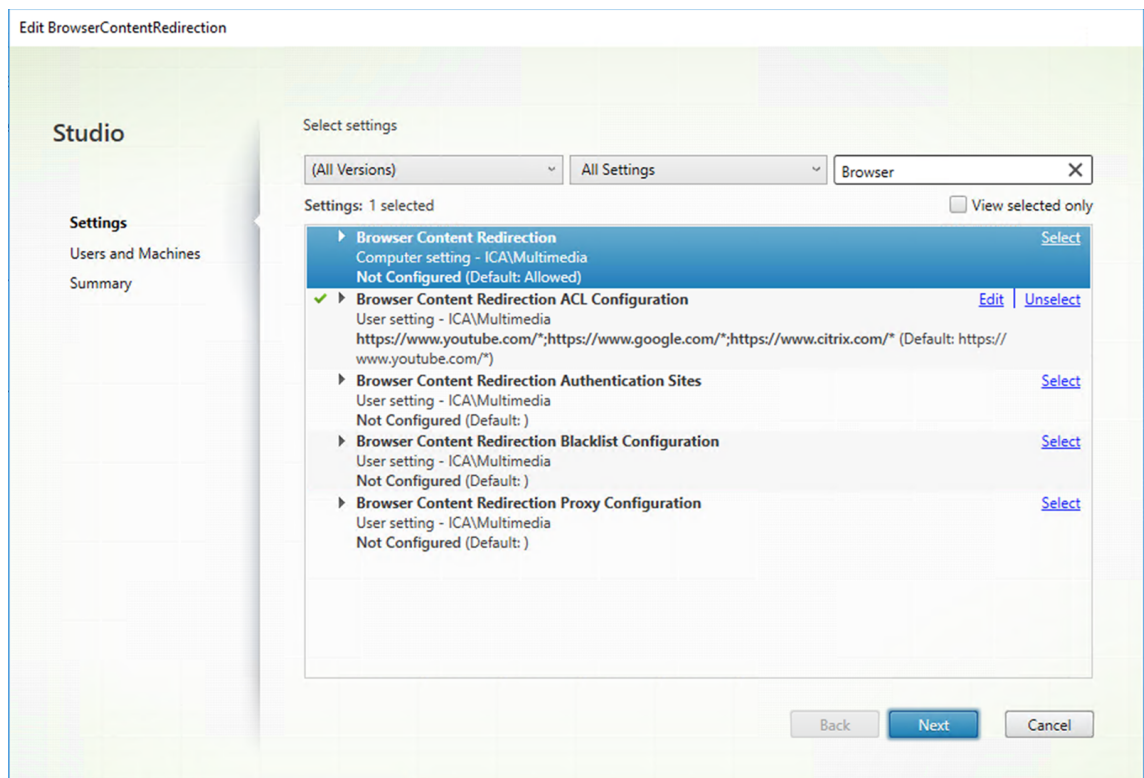
Linux VDA：

- VDA 上的浏览器：添加了 Citrix 浏览器内容重定向扩展程序的 Google Chrome v66 或更高版本

配置浏览器内容重定向

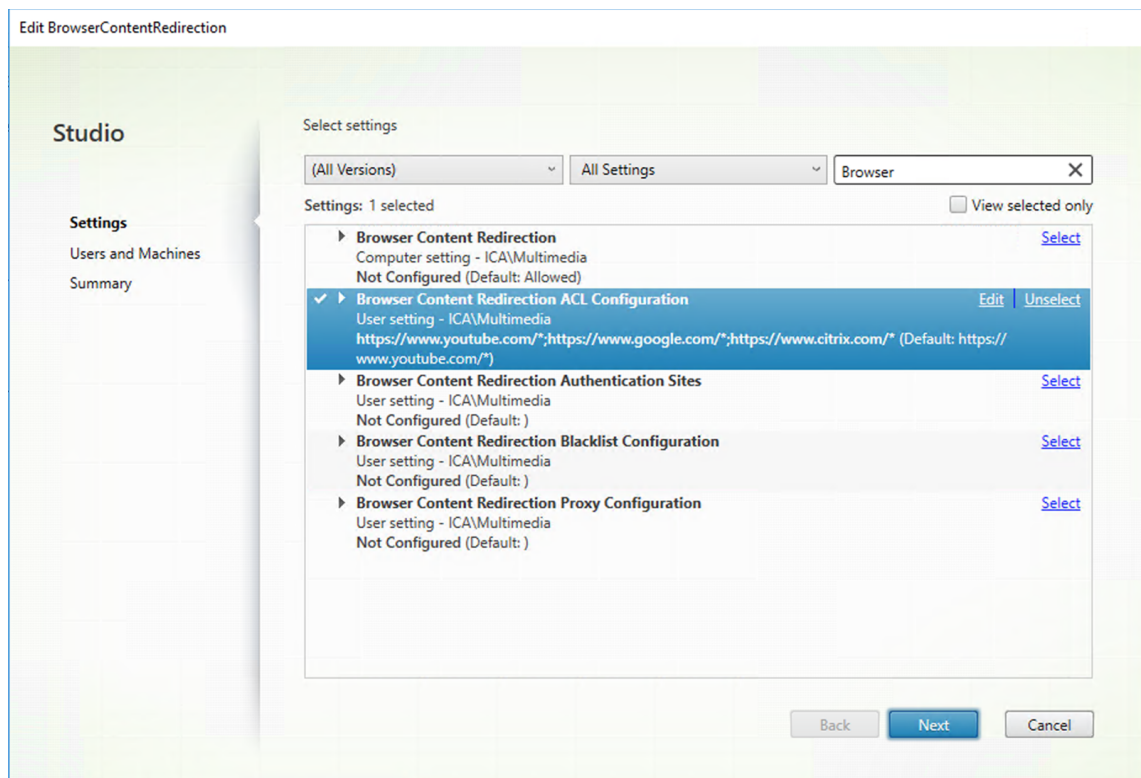
要使用浏览器内容重定向，请配置相关策略并在 Google Chrome 中安装浏览器内容重定向扩展程序。为此，请完成以下步骤：

1. 在 Citrix Studio 中，将浏览器内容重定向设置为允许以启用浏览器内容重定向。



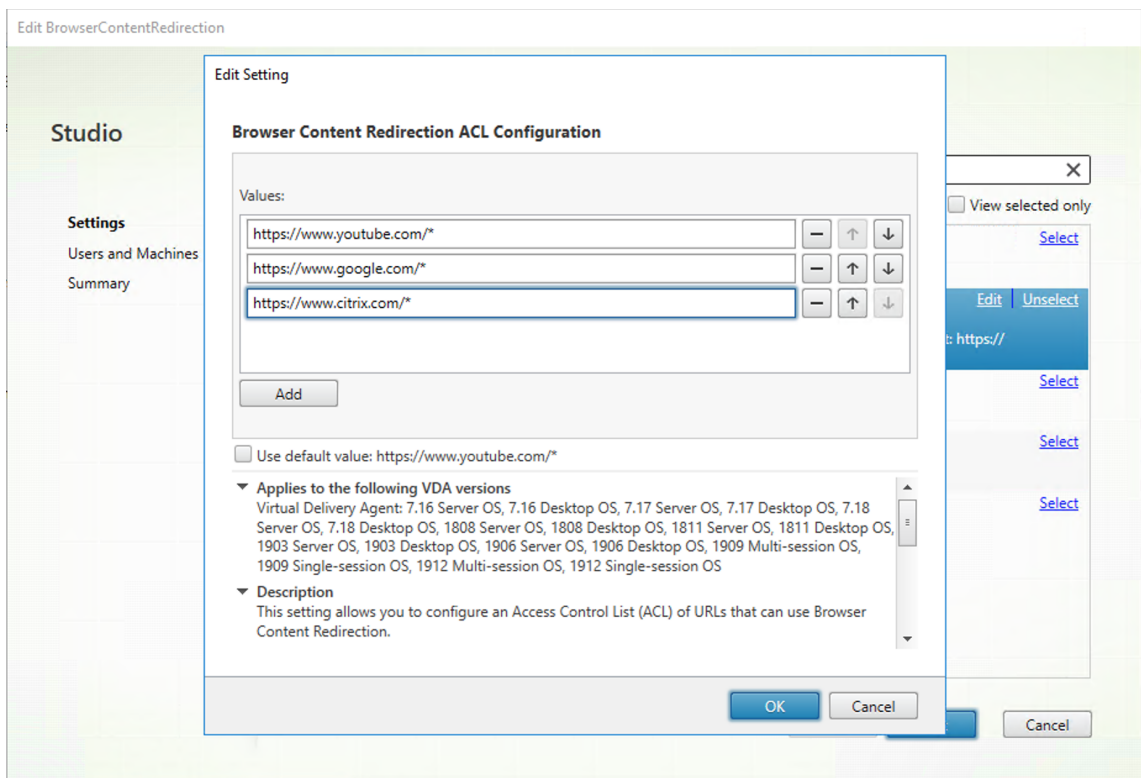
2. 指定其内容能够重定向到客户端的 URL 的允许列表和用于禁用特定 URL 重定向的阻止列表。配置阻止列表属于可选操作。

浏览器内容重定向 **ACL** 配置设置指定其内容能够重定向到客户端的 URL 的允许列表。指定 URL 时，可以使用 * 通配符表示除协议之外的所有 URL 组件。

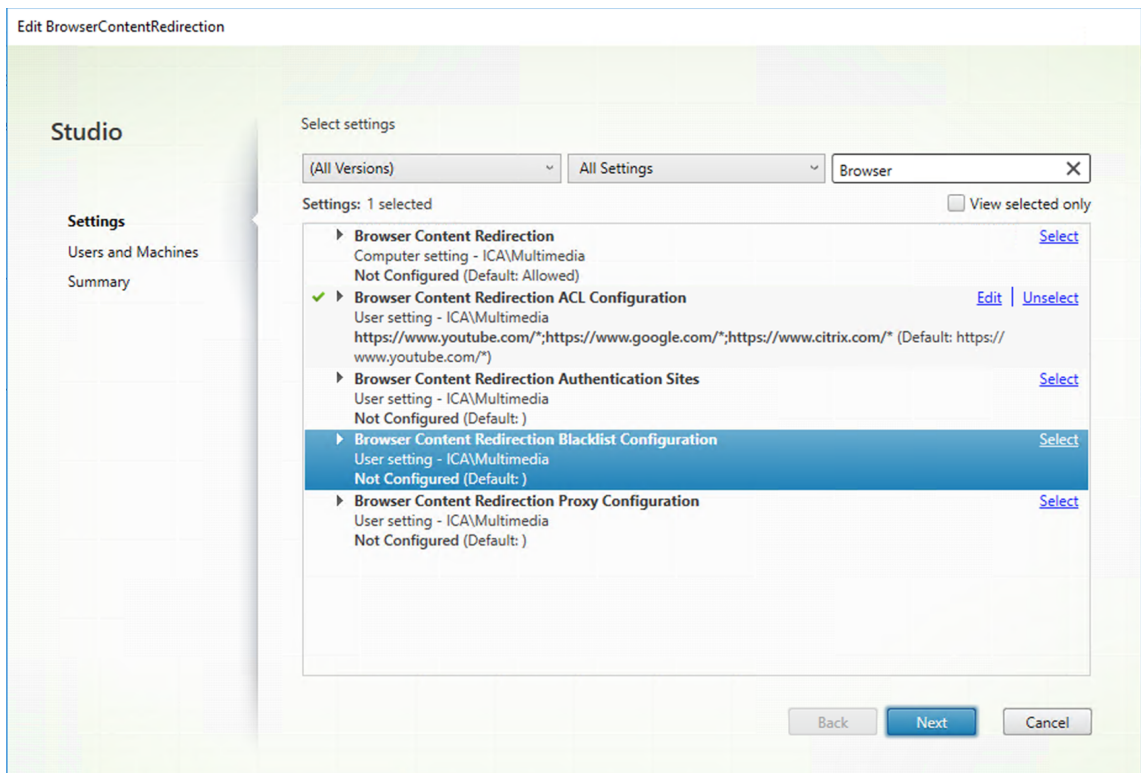


下面是允许的示例：

- `http://www.xyz.com/index.html` (可以通过在 URL 中指定路径来实现更好的粒度。例如，如果指定 `https://www.xyz.com/sports/index.html`，则只重定向 `index.html` 页面。)
- `https://www.xyz.com/*`
- `http://www.xyz.com/*videos*`
- `http://*.xyz.com/`
- `http://*.*.com/`



浏览器内容重定向黑名单配置指定禁用特定 URL 重定向的阻止列表。



3. 要启用服务器提取和客户端呈现，请配置浏览器内容重定向代理配置设置。

此设置用于为 VDA 上的浏览器内容重定向的代理设置提供配置选项。如果已使用有效的代理地址和端口号、PAC/WPAD URL 或直接/透明设置启用此设置，Citrix Workspace 应用程序将始终先尝试进行服务器提取和客户端呈现。有关详细信息，请参阅回退机制。

如果已禁用或未配置此设置并使用默认值，则 Citrix Workspace 应用程序将尝试进行客户端提取和客户端呈现。

默认情况下，此设置为禁止。

显式代理允许的模式：

`http://\<hostname/ip address>:\<port\>`

示例：

`http://proxy.example.citrix.com:80 http://10.10.10.10:8080`

PAC/WPAD 文件允许的模式：

`http://<hostname/ip address>:\<port>/<path>/<Proxy.pac>`

示例：`http://wpad.myproxy.com:30/configuration/pac/Proxy.pac`

`https://<hostname/ip address>:\<port>/<path>/<wpad.dat>`

示例：`http://10.10.10.10/configuration/pac/wpad.dat`

直接或透明代理允许的模式：

在策略文本框中键入单词 **DIRECT**。

注意：

还可以通过编辑注册表值 `HKLM\Software\Citrix\HdxMediastream\WebBrowserRedirection` 来设置代理。此外，注册表值 `HKLM\Software\Citrix\HdxMediastream\AllowNonTlsPacUri` 允许您决定是否允许通过 HTTP 下载 PAC 文件。默认值为 0，这意味着不允许使用 HTTP。

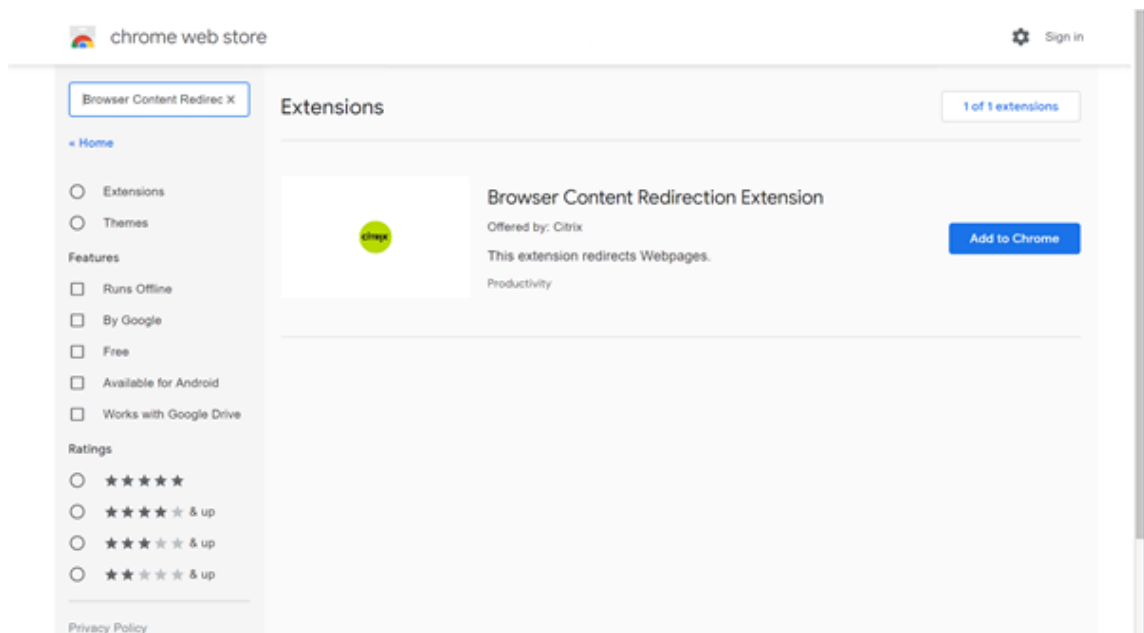
用于策略设置的注册表覆盖选项。有关相关注册表项的列表，请参阅浏览器内容重定向注册表项覆盖。

4. 单击 VDA 上的 **Add to Chrome**（添加到 Chrome），从 Chrome 网上应用店添加 Citrix 浏览器内容重定向扩展程序。这样做有助于 VDA 上的浏览器检测（正在导航到的）URL 是否与允许列表或阻止列表匹配。

重要：

客户端上不需要此扩展程序。仅在 VDA 上添加。

Chrome 扩展程序基于每个用户安装。不需要更新黄金映像即可添加或删除扩展程序。



回退机制

如果您启用了浏览器内容重定向代理配置策略，则 Citrix Workspace 应用程序将尝试进行服务器提取和客户端呈现。如果服务器提取和客户端呈现失败，则会回退到客户端提取和客户端呈现。如果客户端计算机无法访问 Web 服务器，VDA 上的浏览器随后可以在服务器上重新加载和呈现页面（服务器提取服务器呈现）。

浏览器内容重定向注册表项覆盖

警告：

注册表编辑不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因注册表编辑器使用不当导致出现的问题能够得以解决。使用注册表编辑器需自担风险。在编辑注册表之前，请务必进行备份。

HKLM\Software\Citrix\HdxMediastream

名称	类型	值
WebBrowserRedirection	DWORD	1 = 允许，0 = 禁止
WebBrowserRedirectionAcl	REG_MULTI_SZ	/

名称	类型	值
WebBrowserRedirectionProxyAddress	REG_SZ	如果将其设置为以下任何一种模式，则会启用服务器提取客户端呈现：显式代理 - 如果您的数据中心的中心中只有一个显式代理。直接或透明 - 如果没有代理或者如果使用透明代理。 PAC 文件 - 如果您依赖 PAC 文件，以便 VDA 中的浏览器可以自动选择适当的代理服务器来获取指定的 URL。
WebBrowserRedirectionBlacklist	REG_MULTI_SZ	/
AllowNonTlsPacUri	DWORD	确定是否允许通过 HTTP 下载 PAC 文件。默认值为 0，这意味着不允许使用 HTTP。如果将其设置为 1， HDXWebProxy.exe 可以通过 HTTP 下载 PAC 文件（而非严格通过 HTTPS）。

HDX 网络摄像机视频压缩

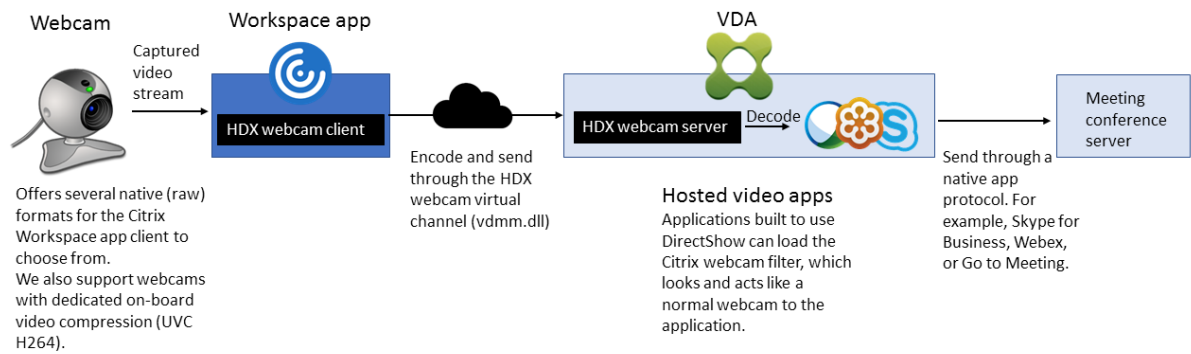
August 20, 2024

概述

在 Linux VDA 会话中运行的视频会议应用程序的用户现在可以使用 HDX 网络摄像机视频压缩功能的网络摄像机。默认情况下启用该功能。我们建议您始终尽可能使用 HDX 网络摄像机视频压缩功能。

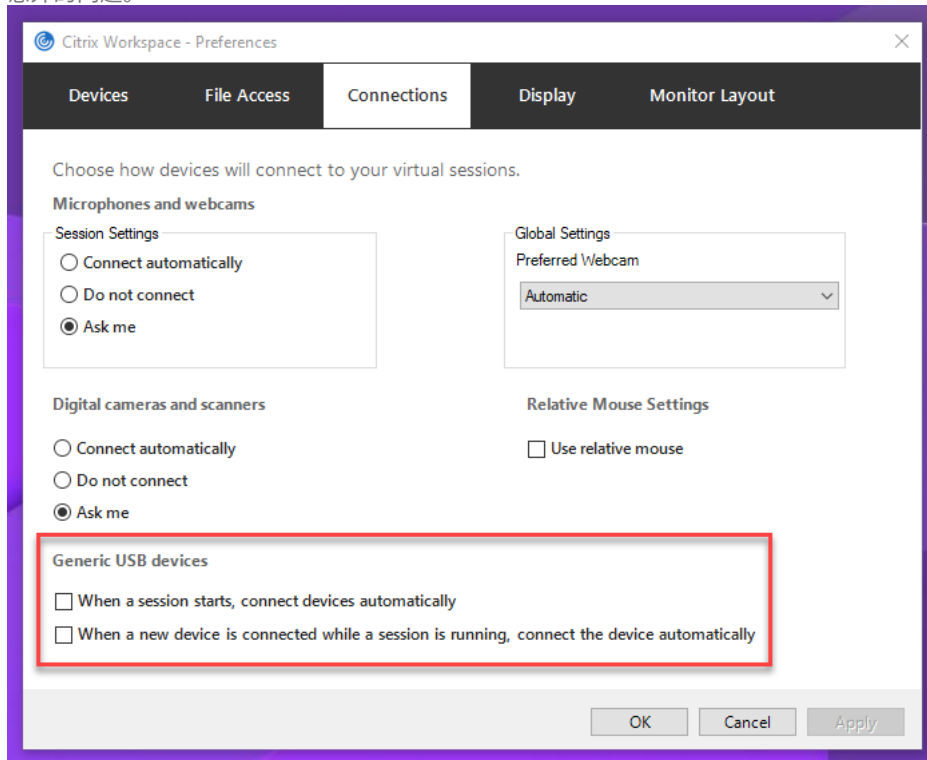
HDX 网络摄像机视频压缩也称为优化网络摄像机模式。这种类型的网络摄像机视频压缩将 H.264 视频直接发送到在虚拟会话中运行的视频会议应用程序。HDX 网络摄像机视频压缩使用属于客户端操作系统的多媒体框架技术捕获来自捕捉设备的视频，并对其进行转换代码和压缩。捕获设备的制造商提供插入操作系统内核流技术推送体系结构的驱动程序。

客户端处理与网络摄像机的通信。之后，客户端仅将视频发送到可以正确显示它的服务器。服务器不能直接通过网络摄像机通信，但其集成可在您的桌面中为您提供相同的体验。Workspace 应用程序会压缩视频以节省带宽，并在 WAN 场景中提高恢复能力。



注意：

- 该功能不适用于 Azure 计算机，因为 Azure 计算机上缺少该功能所依赖的 **videodev** 内核模块。
- 此功能仅支持来自 Citrix Workspace 应用程序客户端的 H.264 视频。
- 支持的网络摄像机分辨率在 48x32 到 1920x1080 之间。
- 使用网络摄像机时，请勿从 Citrix Workspace 应用程序工具栏中选择通用 **USB** 设备。否则，可能会出现意外的问题。

支持 **Citrix Workspace** 应用程序

HDX 网络摄像机视频压缩支持以下版本的 Citrix Workspace 应用程序：

Platform	处理器
适用于 Windows 的 Citrix Workspace 应用程序	适用于 Windows 的 Citrix Workspace 应用程序支持 XenApp 和 XenDesktop 7.17 及更高版本上面向 32 位和 64 位应用程序的网络摄像机视频压缩。在早期版本中，适用于 Windows 的 Citrix Workspace 应用程序仅支持 32 位应用程序。
适用于 Chrome 的 Citrix Workspace 应用程序	由于某些 ARM Chromebook 不支持 H.264 编码，因此，只有 32 位应用程序可以使用优化的 HDX 网络摄像机视频压缩。

完全测试的网络摄像机

不同的网络摄像机提供不同的帧速率，并具有不同级别的亮度和对比度。Citrix 使用以下网络摄像机进行初始功能验证：

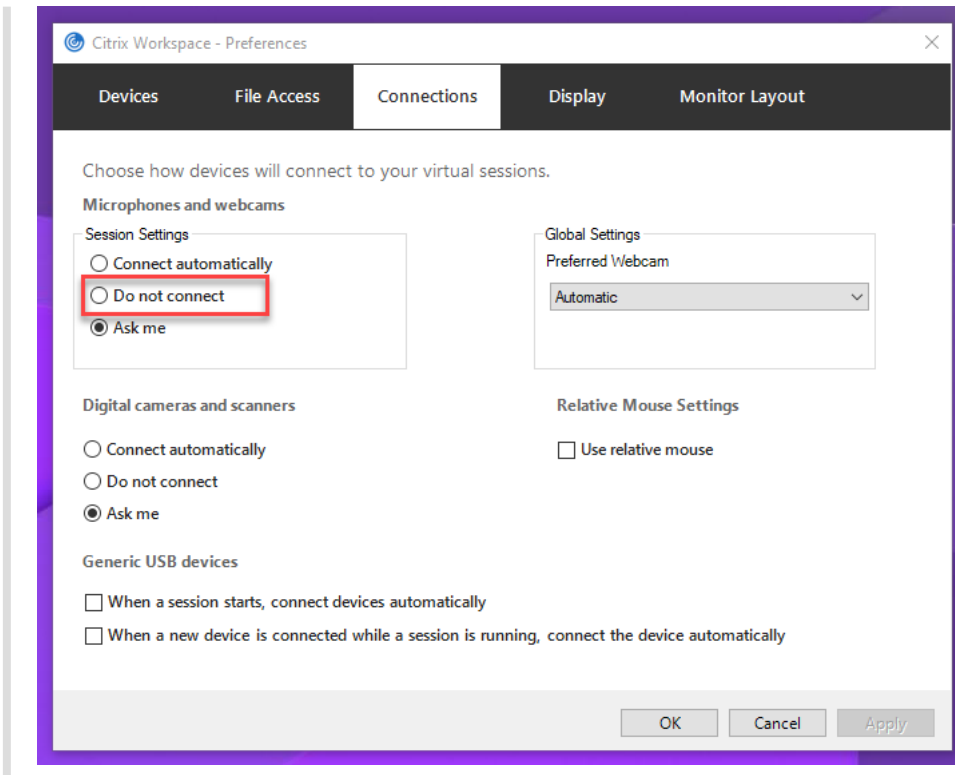
- Logitech HD Webcam C270
- Logitech Webcam C930e
- Microsoft-LifeCam-HD3000

配置

默认情况下启用此功能。要使用，请完成以下验证和配置：

提示：

Citrix Workspace 应用程序用户可以通过选择 Desktop Viewer 麦克风和网络摄像机设置的不连接来覆盖默认设置。



1. VDA 安装完成后，验证 VDA 是否能够在 Delivery Controller 中注册，以及已发布的 Linux 桌面会话是否能够使用 Windows 凭据成功启动。
2. 确保您的 VDA 可以访问 Internet，然后运行 `sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh` 命令以完成网络摄像机配置。如果您的 VDA 无法访问 Internet，请转到步骤 3。

注意：

`uname -r` 与内核标头之间可能会发生内核不匹配。不匹配会导致 `ctxwcamcfg.sh` 脚本失败。要正确使用 HDX 网络摄像机视频压缩，请运行 **`sudo apt-get dist-upgrade`**，重新启动 VDA，然后重新运行 `ctxwcamcfg.sh` 脚本。

如果您的 VDA 部署在 Debian 上，请确保其在最新的内核版本上运行。否则，请运行以下命令以更新到最新的内核版本：

```
1 sudo apt-get update
2 sudo apt-get dist-upgrade
3 sudo reboot
```

如果 VDA 部署在 SUSE 15 上，请运行以下命令以更新到最新的内核版本并重新启动：

```
1 zypper up kernel-default
2 reboot
```

`ctxwcamcfg.sh` 脚本有助于：

- a) 在 VDA 上安装 `kernel-devel` 和动态内核模块支持 (DKMS) 程序。

- `kernel-devel` 用于构建相应版本的虚拟网络摄像机内核模块。
- DKMS 用于动态管理虚拟网络摄像机内核模块。

注意：

在 RHEL、Rocky Linux 和 CentOS 上安装上述程序时，`ctxwcamcfg.sh` 脚本会在您的 VDA 上安装并启用以下存储库：

- 适用于 Enterprise Linux (EPEL) 的额外软件包
- RPM Fusion

- 从 <https://github.com/umlaeute/v4l2loopback> 中下载 `v4l2loopback` 开源代码并使用 DKMS 管理 `v4l2loopback`。
`v4l2loopback` 是一个允许您创建 V4L2 环回设备的内核模块。
 - 运行 `sudo systemctl restart ctxwcamsd` 命令。Linux VDA 的网络摄像机服务 - `ctxwcamsd` - 重新启动并加载 HDX 网络摄像机视频压缩功能的 `v4l2loopback` 内核模块。
3. 如果您的 VDA 无法访问 Internet，请在另一台计算机上构建 `v4l2loopback` 内核模块，然后再将其复制到您的 VDA。
- 准备一台能够访问 Internet 且内核版本与 VDA 相同的计算机。`uname -r` 命令可帮助查找内核版本。
 - 在计算机上，运行 `sudo mkdir -p /var/xdl` 命令。
 - 将 `/var/xdl/configure_*` 从您的 VDA 复制到 `/var/xdl/` 下的计算机。
 - 在计算机上，运行 `sudo /opt/Citrix/VDA/sbin/ctxwcamcfg.sh` 命令以生成内核模块。如果该命令成功运行，则会在 `/var/lib/dkms/v4l2loopback/1.81b8df79107d1fbf392fdcbaa051bd227a9c94c1/$(uname -r)/x86_64/module/` 路径下创建 `v4l2loopback.ko` 文件。忽略在运行 `ctxwcamcfg.sh` 脚本时可能会出现错误。
 - 将 `v4l2loopback.ko` 从计算机复制到您的 VDA 并将其置于 `/opt/Citrix/VDA/lib64/` 下。
 - 在您的 VDA 上，运行 `sudo systemctl restart ctxwcamsd` 命令以重新启动网络摄像机服务并加载 `v4l2loopback` 内核模块。

未加入域的 Linux VDA

July 15, 2024

概述

未加入域的 VDA 无需将 VDA 加入 Active Directory 域进行 VDA 和用户身份验证。创建未加入域的 VDA 时，您需要生成公私密钥对，用于将 VDA 注册到云控制平面。因此，不再需要加入 Active Directory 域。当用户从未加入域的 VDA 启动会话时，VDA 会使用用户用于登录 Citrix Workspace 应用程序的用户名创建本地映射帐户。VDA 分配一个随机密码，本地映射帐户使用该密码进行 SSO 和会话重新连接。如果您更改了随机密码，SSO 和会话重新连接将失败。要禁用 SSO，请参阅[非 SSO 身份验证](#)。

重要：

- Citrix DaaS 支持未加入域的 VDA。
 - 您的控制平面必须在 Citrix DaaS 上部署。
 - 可以在公有云或本地数据中心部署未加入域的 VDA。Citrix DaaS 中的控制平面管理未加入域的 VDA。
 - 要创建未加入域的 VDA，必须启用 [Rendezvous V2](#)。仅当您计划在本地虚拟机管理程序上预配计算机或者想要在 Workspace 中使用 Active Directory 作为身份提供程序时，才需要使用 Cloud Connector。
- 要创建未加入域的 VDA，您可以同时使用 Machine Creation Services (MCS) 和轻松安装。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)和[使用轻松安装创建未加入域的 Linux VDA](#)。
 - MCS 不支持裸机服务器。

适用于未加入域的 Linux VDA 的功能

在未加入域的 VDA 上创建具有指定属性的本地用户

打开未加入域的 VDA 上托管的会话时，VDA 会自动创建具有默认属性的本地用户。VDA 将根据您用于登录 Citrix Workspace 应用程序的用户名创建本地用户。还可以指定用户属性，包括用户的用户标识符 (UID)、组 ID (GID)、主目录和登录 shell。要使用此功能，请完成以下步骤：

1. 请运行以下命令以启用该功能：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent\LocalMappedAccount" -t "REG_DWORD" -v "CreateWithUidGid" -d "0x00000001" --force
```

2. 在 VDA 的安装路径下的 `/var/xdl/getuidgid.sh` 脚本中指定以下属性：

属性	必需或可选	说明
<code>uid</code>	必需	用户标识符 (UID) 是由 Linux 分配给系统中的每个用户的编号。它决定用户可以访问哪些系统资源。
<code>gid</code>	必需	组标识符 (GID) 是用于表示特定组的编号。

属性	必需或可选	说明
homedir	可选	Linux 主目录是特定用户的目录。
shell	可选	登录 shell 是在用户登录其用户帐户时提供给用户的 shell。

下面是 `getuidgid.sh` 脚本的示例：

注意：

请确保您在脚本中指定的属性有效。

```
1 #!/bin/bash
2
3 #####
4 #
5 # Citrix Virtual Apps & Desktops For Linux Script: Get uid and gid
6 #   for the user
7 # Copyright (c) Citrix Systems, Inc. All Rights Reserved.
8 #
9
10 export LC_ALL="en_US.UTF-8"
11
12 function get_uid_gid_for_user()
13 {
14
15     echo "uid:12345"
16     echo "gid:1003"
17     echo "homedir:/home/$1"
18     echo "shell:/bin/sh"
19 }
20
21
22 get_uid_gid_for_user $1
```

非 SSO 身份验证

默认情况下，Linux VDA 已启用单点登录 (SSO)。用户使用一组凭据登录 Citrix Workspace 应用程序和 VDA 会话。要让用户使用一组不同的凭据登录 VDA 会话，请在 Linux VDA 上禁用 SSO。有关详细信息，请参阅[非 SSO 身份验证](#)。

使用 **Azure Active Directory** 进行身份验证

您在 Azure 中部署的未加入域的 VDA 与 AAD 标识服务集成以提供用户身份验证。有关详细信息，请参阅[使用 Azure Active Directory 进行身份验证](#)。

Rendezvous V2

支持未加入域的 VDA 使用 Rendezvous V2 绕过 Citrix Cloud Connector。有关详细信息，请参阅[Rendezvous V2](#)。

策略支持列表

August 20, 2024

Linux VDA 策略支持列表

Studio 策略	注册表项名称	类型	模块	默认值
使用客户端本地时间	UseLocalTimeOfClient	布尔	ICA 时区控制	使用服务器时区
ICA 往返行程计算	IcaRoundtripCheckEnabled	布尔	ICA 最终用户监视	已启用 (1)
ICA 往返行程计算间隔	IcaRoundtripCheckPeriod	整数	ICA 最终用户监视	15
空闲连接的 ICA 往返行程计算	IcaRoundtripCheckWhileIdle	布尔	ICA 最终用户监视	已禁用 (0)

Studio 策略	注册表项名称	类型	模块	默认值
总会话带宽限制	LimitOverallBw	DWORD	ICA\带宽	0
音频重定向带宽限制	LimitAudioBw	DWORD	ICA\带宽	0
音频重定向带宽限制百分比	LimitAudioBwPercent	DWORD	ICA\带宽	0
客户端USB设备重定向带宽限制	LimitUSBDeviceBw	DWORD	ICA\带宽	0
客户端USB设备重定向带宽百分比	LimitUSBDeviceBwPercent	DWORD	ICA\带宽	0
文件重定向带宽限制	LimitCdmBw	DWORD	ICA\带宽	0
文件重定向带宽限制百分比	LimitCdmBwPercent	DWORD	ICA\带宽	0
打印机重定向带宽限制	LimitPrinterBw	DWORD	ICA\带宽	0

策略	项名称	类型	模块	默认值
打印机	LimitPrinterBwPercentage	百分比	ICA	0
重定向			带宽	
带宽限			制百分	
制百分			比	
比				
WebSockets	AcceptWebSockets	计算机	ICA\WebSockets	允许
连接				
WebSockets	WebSocketsPort	计算机	ICA\WebSockets	8080
端口号				
WebSockets	TrustedOriginServers	计算机	ICA\WebSockets	
可信源				
服务器				
列表				
ICA 保	SendICAKeepAlive	计算机	ICA 保	不发送
持活动			持活动	ICA 保
状态			状态	持活动
				状态消
				息 (0)
ICA 保	ICAKeepAliveTime	计算机	ICA 保	60 秒
持活动			持活动	
状态超			状态	
时				
ICA 侦	IcaListenerPort	计算机	ICA	1494
听器端				
口号				
HDX	HDXoverUDP	计算机	ICA	Preferred(2)
自适应				
传输				
Rendezvous	RendezvousProtocol	计算机	ICA	禁止
协议				
会话可	AcceptSessionReliableConnections	计算机	ICA	允许
靠性连			话可靠	(1)
接			性	

策略	项名称	类型	模块	默认值
重新连接 UI 透明度级别	Reconnect	计算机	ITracex	30%
会话可靠性口号	SessionReliability	计算机	ICA\会话可靠	2598
会话可靠性超时	SessionReliabilityTimeout	计算机	ICA\会话可靠	180 s
客户端自动重新连接	AllowAutoReconnect	客户端	ICA\客户端	允许 (1)
客户端音频重定向	AllowAudioRedirection	客户端	ICA\音频	允许 (1)
客户端打印机重定向	AllowPrinterRedirection	客户端	ICA\打印	允许 (1)
自动创建 PDF 通用打印机	AutoCreatePDFPrinter	客户端	ICA\打印	已禁用 (0)

策略	注册表项名称	类型	模块	默认值
打印机驱动程序映射和兼容性	DriverMappingList	字符串	打印	"Microsoft XPS Document Writer *, Deny ; Send to Microsoft OneNote *, Deny"
客户端剪贴板重定向	AllowClipboardRedirection	布尔值	剪贴板	允许 (1)
将剪贴板客户端限制为会话传输大小	LimitClipboardTransferC2H	整数	剪贴板	已禁用 (0)
将剪贴板会话限制为客户端传输大小	LimitClipboardTransferH2C	整数	剪贴板	已禁用 (0)

策略	项名称	类型	模块	默认值
剪贴板重定向带宽限制	LimitClipboardBW	数字	ICA\带宽	0
剪贴板重定向带宽限制百分比	LimitClipboardBWPct	数字	ICA\带宽	0
限制客户端剪贴板写入	RestrictClipboardWrite	布尔	ICA\带宽	已禁用 (0)
客户端剪贴板写入允许的格式	ClientClipboardAllowedFormats	字符串	ICA\带宽	禁用
限制会话剪贴板写入	RestrictSessionClipboardWrite	布尔	ICA\带宽	已禁用 (0)
会话剪贴板写入允许的格式	SessionClipboardAllowedFormats	字符串	ICA\带宽	禁用
客户端USB设备重定向	AllowUSBRedir	布尔	USB	禁止 (0)
客户端USB设备重定向规则	USBDeviceRules	字符串	USB	”
客户端USB设备重定向规则	USBDeviceRules	字符串	USB	” “\0” ”

策略	项名称	类型	模块	默认值
移动图像压缩	MovingImageCompressionConfiguration	用户	Thinwire	禁用 (1)
额外颜色压缩	ExtraColorCompression	用户	Thinwire	已禁用 (0)
目标最低帧速率	TargetedMinimumFramesPerSecond	用户	Thinwire	15 fps
目标帧速率	FramesPerSecond	用户	Thinwire	30 fps
视觉质量	VisualQuality	用户	Thinwire	中 (3)
使用视频编解码器进行压缩	VideoCodec	用户	Thinwire	首选时使用 (3)
使用视频编解码器的硬件编码	UseHardwareEncoding	用户	Thinwire	禁用 (1)
允许视觉无损压缩	AllowVisuallyLosslessCompression	用户	Thinwire	已禁用 (0)
针对 3D 图形工作负载优化	OptimizeFor3DWorkload	用户	Thinwire	已禁用 (0)
简单图形的首选颜色深度	PreferredColorDepth	用户	Thinwire	24 位/像素 (1)

策略	项名称	类型	模块	默认值
图形状 态指示 器	DisplayLossIndicators	注册表	Tratowire	已禁用 (0)
音频质 量	SoundQuality	音频	音频	高 - 高 清晰度 音频 (2)
客户端 麦克风 重定向	AllowMicrophoneRedirection	音频	音频	允许 (1)
最大会 话数	MaximumNumberofSessions	计算机 管理	负载管 理	250
并发登 录数容 差	ConcurrentSessionsPerDevice	计算机 管理	负载管 理	2
启用控 制器自 动更新	EnableAutomaticControllerUpdates	计算机 管理	OffControl Deliv- ery Agent 设置	(1)
剪贴板 选择更 新模式	ClipboardSelectionMode	注册表	剪贴板	ClipboardMode
主选定 内容更 新模式	PrimarySelectionUpdateMode	注册表	剪贴板	ClipboardMode
最高 speex 质量	MaxSpeexQuality	音频	音频	5
自动连 接客户 端驱动 器	AutoConnectDrives	文件重 定向	文件重 定向\CDM	已启用 (1)

策略	项名称	类型	模块	默认值
客户端 光盘驱 动器	AllowCdr	客户 端	文件重 定 向\CDM	允许 (1)
客户端 固定驱 动器	AllowFixed	客户 端	文件重 定 向\CDM	允许 (1)
客户端 软盘驱 动器	AllowFloppy	客户 端	文件重 定 向\CDM	允许 (1)
客户端 网络驱 动器	AllowNetwork	客户 端	文件重 定 向\CDM	允许 (1)
客户端 可移动 驱动器	AllowRemovable	客户 端	文件重 定 向\CDM	允许 (1)
客户端 驱动器 重定向	AllowDriveRedir	客户 端	文件重 定 向\CDM	允许 (1)
只读客 户端驱 动器访 问	ReadOnlyMapped	客户 端	文件重 定 向\CDM	已禁用 (0)
自动显 示键盘	AllowAutoKeyboard	客户 端	文件重 定 向\CDM	已禁用 (0)
允许在 桌面与 客户端 之间传 输文件	AllowFileTransfer	客户 端	文件传 输	允许
从桌面 下载文 件	AllowFileDownload	客户 端	文件传 输	允许

策略	项名称	类型	模块	默认值
将文件 上载到 桌面	AllowFileUpload	布尔	文件传 输	允许
会话空 闲计时 器	EnableSessionIdleTimer	布尔	会话计 时器	已启用 (1)
会话空 闲计时 器间隔	SessionIdleTimerInterval	数字	会话计 时器	1440 分钟
断开会 话计时 器	EnableSessionDisconnectTimer	布尔	会话断 开计时器	已禁用 (0)
断开会 话计时 器间隔	SessionDisconnectTimerPeriod	数字	会话断 开计时器	1440 分钟
浏览器 内容重 定向	WebBrowserRedirect	布尔	计算 机媒体	允许
浏览器 内容重 定向 ACL 配 置	WebBrowserRedirectACL	字符串	计算 机媒体	https :// www. youtube .com /*
浏览器 内容重 定向身 份验证 站点	WebBrowserRedirectAuthenticationSites	字符串	计算 机媒体	空值
浏览器 内容重 定向黑 名单配 置	WebBrowserRedirectBlacklist	字符串	计算 机媒体	空值

策略	项名称	类型	模块	默认值
浏览器内容重定向代理配置	WebBrowserRedirectCA	字符串	ICA\媒体	空值
启用会话水印	EnableSessionWatermark	布尔	ICA\会话	已禁用
包括客户端 IP 地址	IncludeClientIPAddress	布尔	ICA\会话	已禁用
包括连接时间	IncludeConnectTime	布尔	ICA\会话	已禁用
包括登录用户名	IncludeLoginUserName	布尔	ICA\会话	已启用
包括 VDA 主机名	IncludeVDAMainHostName	布尔	ICA\会话	已启用
包括 VDA IP 地址	IncludeVDAMIPAddress	布尔	ICA\会话	已禁用
会话水印样式	WatermarkStyle	字符串	ICA\会话	多个
水印透明度	WatermarkTransparency	字符串	ICA\会话	17

Studio 注册表

策略	项名称	类型	模块	默认值
----	-----	----	----	-----

水印自定义	Watermark	Text	CustomTEA会话水	有效的自定义
本			印\水印内容	文本配置可能

```

如下所示:
<
font
=
Arial

Unicode
MS
><
domain
><
newline
><
fontsize
=47><
username
><
newline
><
fontzoom
=120>
Citrix
<
newline
><
style
=
tile
>
Najing
<
clientip
><
newline
><

```

Studio 注册表

策略	项名称	类型	模块	默认值
----	-----	----	----	-----

会话指 标收集	UsageDataCollectionThroughClient	布尔	计算机	禁用 (1)
------------	----------------------------------	----	-----	-----------

策略	项名称	类型	模块	默认值
动态窗口预览	EnableDynamicPreview	布尔型	计算机预览	此设置启用或禁用无缝窗口的显示。启用此设置后，您可以在已发布的应用程序的窗口中预览屏幕上的内容。例如，您可以使用用于 Windows 的 Citrix Workspace 应用程序打开 Google Chrome 会话的多个窗口，然后将鼠标悬停在任务栏上的 Chrome 图标上以预览

注意：

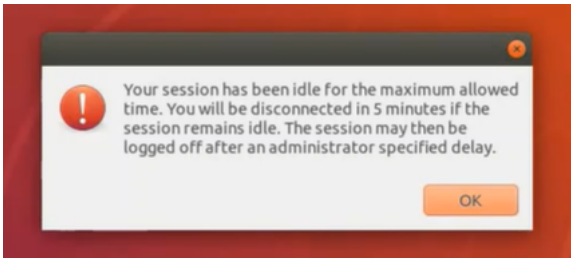
只有 Windows Virtual Delivery Agent (VDA) 支持通过用户数据报协议 (UDP) 传输音频。Linux VDA 不支持。有关详细信息，请参阅[通过用户数据报协议 \(UDP\) 实时传输音频](#)。

可以使用以下 Citrix 策略设置在 Citrix Studio 中配置会话连接计时器：

- 会话空闲计时器：确定是否对空闲会话强制实施时间限制。
- 会话空闲计时器间隔：设置空闲会话的时间限制。如果会话空闲计时器设置为已启用，并且活动会话在设置的时间内未收到用户输入，会话将断开连接。
- 断开连接的会话计时器：确定是否对断开连接的会话强制实施时间限制。
- 断开连接的会话计时器间隔：设置断开连接的会话注销之前的时间间隔。

更新任何策略设置时，请确保这些设置在部署中保持一致。

空闲会话的时间限制到期时，将显示一条警告消息。有关示例，请参见以下屏幕截图。按确定关闭警告消息，但无法使会话保持活动状态。要使会话保持活动状态，请提供用户输入以重置空闲计时器。



可以在 Citrix Studio 7.12 版及更高版本中配置以下策略。

- MaxSpeexQuality

值（整数）：[0-10]

默认值：5

详细信息：

音频质量为中低时，音频重定向采用 Speex 编解码器对音频数据进行编码（请参阅音频质量策略）。Speex 是一种有损编解码器，这意味着它会牺牲输入语音信号保真度来进行压缩。与其他一些语音编解码器不同，这可以控制在质量和比特率之间所做的权衡。大多数时候 Speex 编码过程通过范围在 0 到 10 之间的质量参数控制。质量越高，比特率越高。

最高 Speex 质量根据音频质量和带宽限制（请参阅音频重定向带宽限制策略）选择最佳 Speex 质量对音频数据编码。如果音频质量为中，编码器将处于宽带模式，这意味着采样率较高。如果音频质量为低，编码器将处于窄带模式，这意味着采样率较低。相同的 Speex 质量在不同的模式下有不同的比特率。最佳 Speex 质量出现在最大值满足以下条件时：

- 不高于最高 Speex 质量。
- 其比特率等于或小于带宽限制。

相关设置：音频质量、音频重定向带宽限制

- PrimarySelectionUpdateMode

值（枚举）：[0, 1, 2, 3]

默认值：3

详细信息：

选择数据并通过按下鼠标中键粘贴数据时，将使用主选定内容。

此设置控制 Linux VDA 和客户端上的主选定内容变更是否可以在对方的剪贴板上更新。有四个值选项：

Primary selection update mode

Value: Selection changes are not updated on neither client nor host

Use

▼ Description
This setting is supported only by Linux VDA version 1.4 onwards.

PRIMARY selection is used for explicit copy/paste actions such as mouse selection and middle mouse button paste. This setting controls whether PRIMARY selection changes on the Linux VDA can be updated on the client's clipboard (and vice versa). It can include one of the following selection changes:

Selection changes are not updated on the client or the host. PRIMARY selection changes do not update a client's clipboard. Client clipboard changes do not update PRIMARY selection.

Host selection changes are not updated on the client. PRIMARY selection changes do not update a client's clipboard. Client clipboard changes update the PRIMARY selection.

Client selection changes are not updated on the host. PRIMARY selection changes update the client's clipboard. Client clipboard changes do not update the PRIMARY selection.

Selection changes are updated on both the client and host. PRIMARY selection change updates the client's clipboard. Client clipboard changes update the PRIMARY selection.

▼ Related settings
Clipboard selection update mode

- 选定内容变更既不在客户端上也不在主机上更新

Linux VDA 上的主选定内容变更不更新客户端上的剪贴板。客户端上的主选定内容变更不更新 Linux VDA 上的剪贴板。

- 主机选定内容变更不更新到客户端
Linux VDA 上的主选定内容变更不更新客户端上的剪贴板。客户端上的主选定内容变更更新 Linux VDA 上的剪贴板。
- 客户端选定内容变更不更新到主机
Linux VDA 上的主选定内容变更更新客户端上的剪贴板。客户端上的主选定内容变更不更新 Linux VDA 上的剪贴板。
- 选定内容变更同时在客户端和主机上更新
Linux VDA 上的主选定内容变更更新客户端上的剪贴板。客户端上的主选定内容变更更新 Linux VDA 上的剪贴板。此选项为默认值。

相关设置：剪贴板选定内容更新模式

- ClipboardSelectionUpdateMode

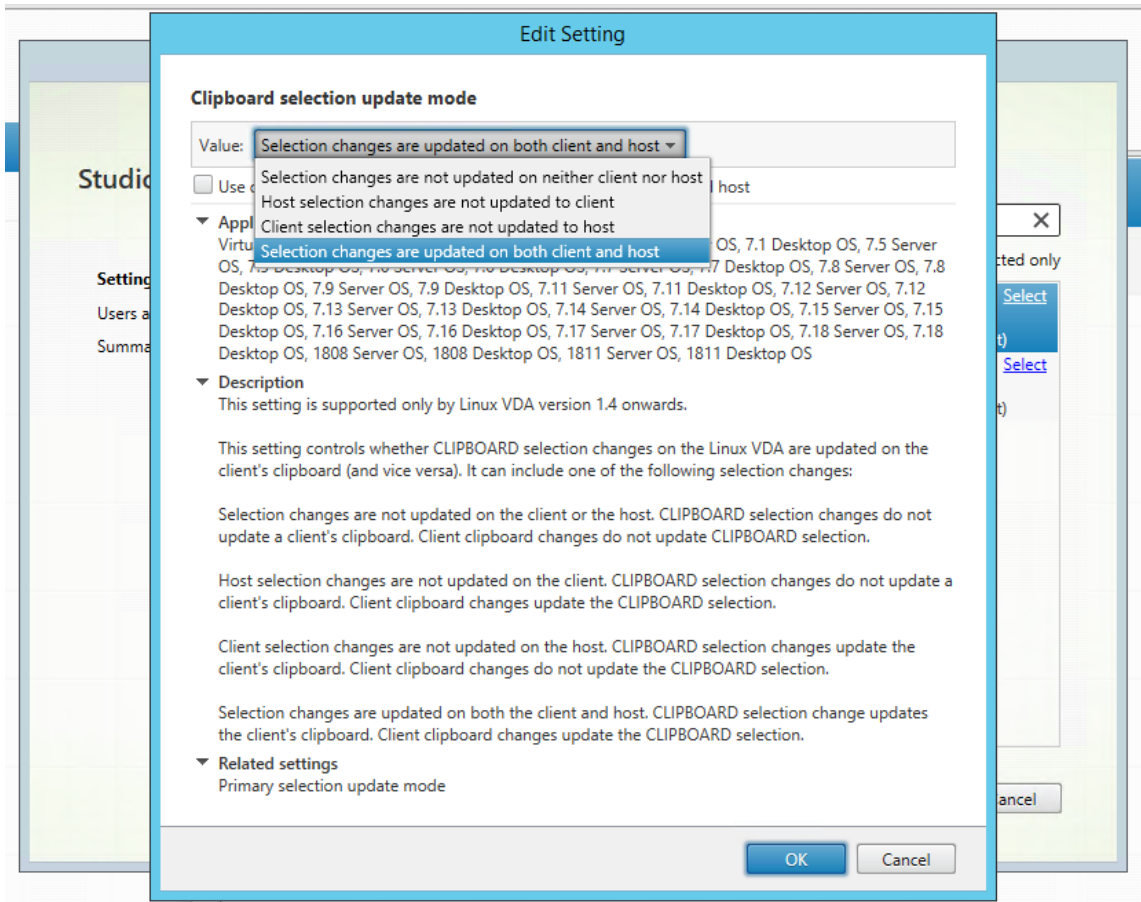
值（枚举）：[0, 1, 2, 3]

默认值：3

详细信息：

当您选择某些数据并明确请求将其“复制”到剪贴板时使用剪贴板选定内容，例如通过从快捷菜单中选择“复制”。剪贴板选定内容主要用于 Microsoft Windows 剪贴板操作，而主选定内容对 Linux 而言是唯一的。

此策略控制 Linux VDA 和客户端上的剪贴板选定内容变更是否可以在对方的剪贴板上更新。有四个值选项：



- 选定内容变更既不在客户端上也不在主机上更新
Linux VDA 上的剪贴板选定内容变更不更新客户端上的剪贴板。客户端上的剪贴板选定内容变更不更新 Linux VDA 上的剪贴板。
- 主机选定内容变更不更新到客户端
Linux VDA 上的剪贴板选定内容变更不更新客户端上的剪贴板。客户端上的剪贴板选定内容变更更新 Linux VDA 上的剪贴板。
- 客户端选定内容变更不更新到主机
Linux VDA 上的剪贴板选定内容变更更新客户端上的剪贴板。客户端上的剪贴板选定内容变更不更新 Linux VDA 上的剪贴板。
- 选定内容变更同时在客户端和主机上更新
Linux VDA 上的剪贴板选定内容变更更新客户端上的剪贴板。客户端上的剪贴板选定内容变更更新 Linux VDA 上的剪贴板。此选项为默认值。

相关设置：主选定内容更新模式

注意：

Linux VDA 支持同时剪贴板选定内容和主选定内容。要控制 Linux VDA 与客户端之间的复制和粘贴行为，我们

建议您同时将剪贴板选定内容更新模式和主选定内容更新模式设置为相同的值。

打印

July 15, 2024

本部分内容包含以下主题：

- [打印最佳做法](#)
- [PDF 打印](#)

打印最佳做法

August 20, 2024

本文提供有关打印最佳实践的信息。

安装

Linux VDA 要求同时启用 **cups** 和 **foomatic** 过滤器。在安装 VDA 时安装过滤器。还可以根据分发情况手动安装过滤器。

打印策略设置

客户端打印机重定向

此设置允许您决定客户端打印机是否映射到 VDA 会话。默认情况下，允许客户端打印机映射。

自动创建客户端打印机

此设置指定可以映射到 VDA 会话的客户端打印机。默认情况下，此设置设为自动创建所有客户端打印机，这意味着所有客户端打印机都映射到 VDA 会话。有关此设置的详细信息，请参阅 Citrix Virtual Apps and Desktops 文档中的[自动创建客户端打印机](#)。

自动创建 **PDF** 通用打印机

要使用 [PDF 打印](#) 功能，请将此策略设置为已启用。

打印机驱动程序映射和兼容性

Citrix 提供三种类型的通用打印机驱动程序（postscript、pcl5 和 pcl6）。但是，通用打印机驱动程序可能与您的客户端打印机不兼容。在这种情况下，早期版本中的唯一选择为编辑 `~/CitrixProfile$CLIENT_NAME` 配置文件。自版本 1906 起，可以选择改为在 Citrix Studio 中配置打印机驱动程序映射和兼容性策略。

要在 Citrix Studio 中配置打印机驱动程序映射和兼容性策略，请执行以下操作：

1. 选择打印机驱动程序映射和兼容性策略。
2. 单击添加。
3. 使用客户端打印机的驱动程序名称填写驱动程序名称。如果使用适用于 Linux 的 Citrix Workspace 应用程序，请改为填写打印机名称。
4. 根据需要执行以下操作：
 - 要允许将客户端打印机重定向到 VDA 会话并仅使用通用打印驱动程序，请选择允许或仅使用通用驱动程序创建。
 - 要阻止具有指定驱动程序名称的客户端打印机重定向到 VDA 会话，请选择不创建。
 - 要允许服务器应用程序访问与服务器具有相同驱动程序但驱动程序名称不同的客户端打印机，请选择替换为，然后在 VDA 上键入驱动程序文件的绝对路径。

Edit Setting

Printer driver mapping and compatibility

Driver Name	Action	Settings	Se
Microsoft XPS Document Writer *	Do not create	Unmodified	
Send to Microsoft OneNote *	Do not create	Unmodified	
FX FPS Color Class Driver	Replace with	/u	

Use default v

Applies to the

Virtual Delivery Agent OS, 7.5
 Server OS, 7.5
 Server OS, 7.5
 Server OS, 7.5
 7.15 Server OS, 7.15
 OS, 7.18 Server OS, 7.18 Desktop OS

Description
 Lists driver substitution rules for auto-created client printers. When you define these rules, you can allow or prevent printers to be created with the specified driver. Additionally, you can allow created printers to use only universal printer drivers.
 Driver substitution overrides (or maps) printer driver names the client provides, substituting an

Driver Name:

Allow
 Do not create
 Create with universal driver only
 Replace with

注意：

仅支持 PPD 驱动程序文件。

使用情况

可以从已发布的桌面和已发布的应用程序打印。所有客户端打印机都可以映射到 VDA 会话。对于桌面和应用程序，打印机名称不同：

- 对于发布的桌面：
`<client printer name>:$CLIENT_NAME:dsk$SESSION_ID`
- 对于发布的应用程序：
`<client printer name>:$CLIENT_NAME:app$SESSION_ID`

注意：

如果同一用户同时打开了已发布的桌面和已发布的应用程序，会话可以访问两种打印机。无法在已发布的应用程序会话中的桌面打印机上打印，也无法在已发布的桌面会话中的应用程序打印机上打印。

故障排除

无法打印

打印无法正常工作时，请检查打印守护程序 **ctxlpmngt** 和 **CUPS** 框架。

打印守护程序 **ctxlpmngt** 是一个按会话进程，必须在会话期间内运行。请运行以下命令以确认打印守护程序是否正在运行。如果 **ctxlpmngt** 未运行，请从命令行中手动启动 **ctxlpmngt**。

```
1 ps -ef | grep ctxlpmngt
```

如果仍无法打印，请检查 **CUPS** 框架。**ctxcups** 服务用于打印机管理，并与 Linux CUPS 框架通信。此进程在每个计算机上有一个，可通过运行以下命令进行检查：

```
1 systemctl status ctxcups
```

收集 **CUPS** 日志的额外步骤

要收集 CUPS 日志，请运行以下命令以配置 CUPS 服务文件。否则，CUPS 日志无法记录在 **hdx.log** 中：

```
1 sudo systemctl stop cups
2
3 sudo vi /etc/systemd/system/printer.target.wants/cups.service
4
5 PrivateTmp=false
6
7 sudo systemctl start cups
8
9 sudo systemctl daemon-reload
```

注意：

此配置仅在出现问题时收集完整的打印日志时设置。在正常情况下，不建议做此配置，因为这会损害 CUPS 安全性。

打印输出为乱码

打印机驱动程序不兼容可能会导致输出乱码。系统中为每个用户提供了驱动程序配置，该配置可通过编辑 **~/.CtxlpProfile\$CLIENT_NAME** 配置文件进行配置：

```
1 [DEFAULT_PRINTER]
2
3 printername=
4
5 model=
6
7 ppdpath=
8
9 drivertype=
```

重要:

printername 字段包含的是当前客户端默认打印机的名称。它是一个只读值。请勿编辑。

不能同时设置字段 **ppdpath**、**model** 和 **drivertype**，因为对于映射的打印机而言，只有一个字段生效。

- 如果通用打印机驱动程序与客户端打印机不兼容，请使用 **model=** 选项配置本机打印机驱动程序的型号。可以使用 **lpinfo** 命令查找打印机的当前型号名称：

```
1 lpinfo - m
2
3 ...
4
5 xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
6
7 xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
8 xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
```

然后可以设置型号以与打印机匹配：

```
1 model=xerox/ph3115.ppd.gz
```

- 如果通用打印机驱动程序与客户端打印机不兼容，请配置本机打印机驱动程序的 PPD 文件路径。**ppdpath** 值是本机打印机驱动程序文件的绝对路径。

例如，`/home/tester/NATIVE_PRINTER_DRIVER.ppd` 下存在一个 **ppd** 驱动程序：

```
1 ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
```

- Citrix 提供了三种类型的通用打印机驱动程序（postscript、pcl5 和 pcl6）。可以根据打印机属性配置驱动程序类型。

例如，如果客户端默认打印机驱动程序类型为 PCL5，请将 **drivertype** 设置为：

```
1 drivertype=pcl5
```

输出大小为零

尝试使用其他类型的打印机。并尝试使用 CutePDF 和 PDFCreator 之类的虚拟打印机以确定此问题是否与打印机驱动程序有关。

打印作业取决于客户端默认打印机的打印机驱动程序。务必确定当前活动驱动程序类型。如果客户端打印机使用的是 PCL5 驱动程序，而 Linux VDA 选择的是 Postscript 驱动程序，则会出现问题。

如果打印机驱动程序类型正确，可以执行以下步骤来确定问题：

1. 登录到已发布的桌面会话。
2. 运行 `vi ~/.CtxlpProfile$CLIENT_NAME` 命令。
3. 添加以下字段以在 Linux VDA 上保存后台打印文件：

```
1 deletespoolfile=no
```

4. 注销并重新登录以加载配置更改。
5. 打印文档以重现问题。打印后，后台打印文件保存在 `/var/spool/cups-ctx/$logon_user/$spool_file` 下。
6. 检查后台打印是否为空。如果后台打印文件大小为零，表示有问题。请联系 Citrix 支持（并提供打印日志）以获取更多指导。
7. 如果 spool 大小不为零，则将该文件复制到客户端。后台打印文件内容取决于客户端默认打印机的打印机驱动程序类型。如果映射的打印机（本机）驱动程序是 `postscript`，则可以直接在 Linux 操作系统上打开后台打印文件。检查内容是否正确。

如果后台打印文件是 PCL，或者客户端操作系统是 Windows，则将后台打印文件复制到客户端，并使用不同的打印机驱动程序在客户端打印机上打印该文件。

8. 将映射的打印机更改为使用不同的打印机驱动程序。下例以 PostScript 客户端打印机为例：

- a) 登录活动会话，在客户端桌面上打开浏览器。
- b) 打开打印管理门户：

```
1 localhost:631
```

- c) 选择映射的打印机 `CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID` 和修改打印机。此操作要求使用管理员权限。
- d) 保持 `cups-ctx` 连接，然后单击 **Continue**（继续）以更改打印机驱动程序。
- e) 在 **Make**（制造商）和 **Model**（型号）字段中，从 Citrix UPD 驱动程序中选择不同的打印机驱动程序。例如，如果安装了 CUPS-PDF 虚拟打印机，可以选择“Generic CUPS-PDF Printer”（通用 CUPS-PDF 打印机）驱动程序。保存更改。
- f) 如果此过程成功，则会在 `.CtxlpProfile$CLIENT_NAME` 中配置驱动程序的 PPD 文件路径，以允许映射的打印机使用新选择的驱动程序。

已知问题

下面是已确定的在 Linux VDA 上打印时存在的问题：

CTXPS 驱动程序与部分 PLC 打印机不兼容

如果发生打印输出损坏，请将打印机驱动程序设置为制造商提供的本机打印机驱动程序。

打印大文档时打印速度较慢

在本地客户端打印机上打印大文档时，文档会通过服务器连接进行传输。如果连接的速度很慢，传输可能需要很长时间。

在其他会话中看到打印机和打印作业通知

Linux 的会话概念与 Windows 操作系统不同。因此，所有用户都会获得系统范围的通知。您可以禁用这些通知，方法是更改 CUPS 配置文件：**`/etc/cups/cupsd.conf`**。

找到文件中配置的当前策略名称：

DefaultPolicy **default**

如果策略名称为 *default*，则将以下行添加到默认策略 XML 块中：

```
1 <Policy default>
2
3     # Job/subscription privacy...
4
5     JobPrivateAccess default
6
7     JobPrivateValues default
8
9     SubscriptionPrivateAccess default
10
11    SubscriptionPrivateValues default
12
13    ... ..
14
15    <Limit Create-Printer-Subscription>
16
17        Require user @OWNER
18
19        Order deny,allow
20
21    </Limit>
22
23    <Limit All>
24
25        Order deny,allow
26
27    </Limit>
28
29 </Policy>
```

PDF 打印

July 15, 2024

使用支持 PDF 打印的 Citrix Workspace 应用程序版本，可以打印从 Linux VDA 会话内部转换的 PDF。会话打印作业将被发送到安装了 Citrix Workspace 应用程序的本地计算机。在本地计算机上，可以使用所选 PDF 查看器打开 PDF，并在所选打印机上进行打印。

Linux VDA 支持在以下版本的 Citrix Workspace 应用程序上进行 PDF 打印：

- Citrix Receiver for HTML5 版本 2.4 到 2.6.9、适用于 HTML5 的 Citrix Workspace 应用程序 1808 及更高版本
- Citrix Receiver for Chrome 版本 2.4 到 2.6.9、适用于 Chrome 的 Citrix Workspace 应用程序 1808 及更高版本
- 适用于 Windows 的 Citrix Workspace 应用程序 1905 及更高版本

配置

除了使用支持 PDF 打印的 Citrix Workspace 应用程序版本外，还要在 Citrix Studio 中设置以下策略：

- 将客户端打印机重定向设置为允许（默认设置为允许）
- 将自动创建 **PDF** 通用打印机设置为已启用（默认设置为已禁用）
- 将自动创建客户端打印机设置为自动创建所有客户端打印机。

启用这些策略后，当您在已启动的会话中单击打印时，打印预览将显示在本地计算机上，以便您选择打印机。有关设置默认打印机的信息，请参阅 [Citrix Workspace 应用程序文档](#)。

Remote PC Access

August 20, 2024

概述

Remote PC Access 是 Citrix Virtual Apps and Desktops 的扩展程序。它使组织能够轻松地允许员工以安全的方式远程访问其物理办公室 PC。如果用户可以访问其办公室 PC，他们可以访问完成工作所需的所有应用程序、数据和资源。

Remote PC Access 使用交付虚拟桌面和应用程序的相同 Citrix Virtual Apps and Desktops 组件。部署和配置 Remote PC Access 的要求和流程与部署 Citrix Virtual Apps and Desktops 所需的要求和流程相同。这种统一性提供了一致且统一的管理体验。用户通过使用 Citrix HDX 交付其远程办公室 PC 会话，获得最佳用户体验。

有关详细信息，请参阅 Citrix Virtual Apps and Desktops 文档中的 [Remote PC Access](#)。

注意事项

这些注意事项是 Linux VDA 特有的：

- 在物理机上，请仅在非 3D 模式下使用 Linux VDA。由于 NVIDIA 的驱动程序限制，启用了 HDX 3D 模式时，PC 的本地屏幕无法停止。显示此屏幕存在潜在的安全风险。
- 请对物理 Linux 计算机使用单会话操作系统类型的计算机目录。
- 自动用户分配不适用于 Linux 计算机。通过自动用户分配，用户在本地登录 PC 时会自动将其分配给各自的计算机。在没有管理员干预的情况下执行此登录。客户端上的 Citrix Workspace 应用程序可以帮助用户在 Remote PC Access 桌面会话中访问办公室 PC 上的应用程序和数据。
- 如果用户已在本地登录到其 PC，尝试从 StoreFront 启动 PC 将失败。
- 节能选项不适用于 Linux 计算机。

配置

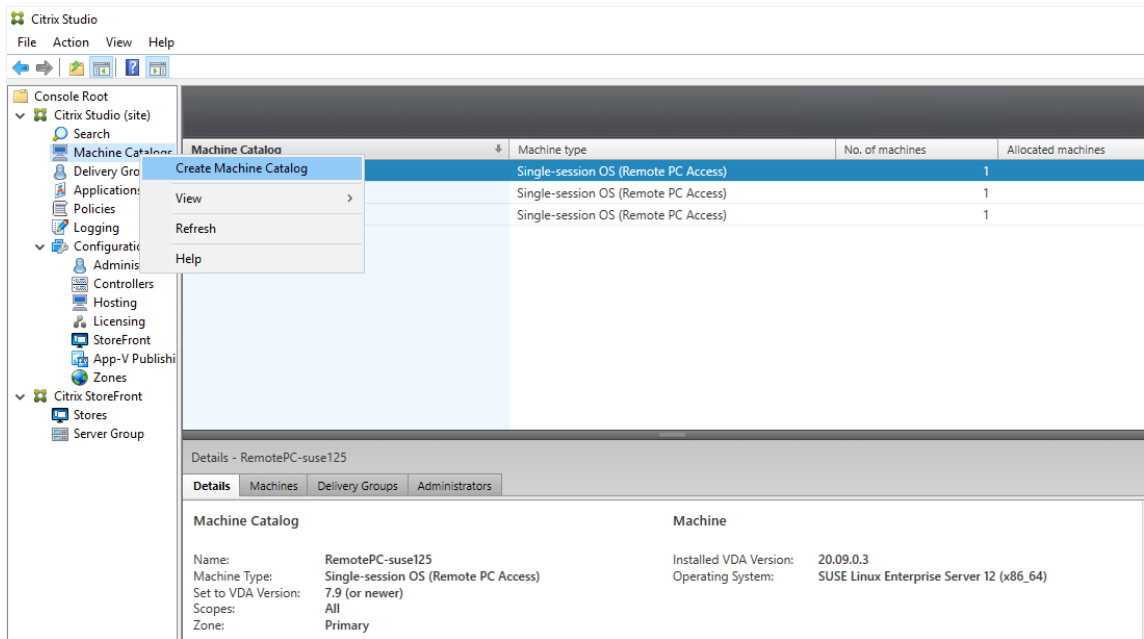
要交付 Linux PC 会话，请在目标 PC 上安装 Linux VDA，创建 **Remote PC Access** 类型的计算机目录，然后创建交付组，以使计算机目录中的 PC 可供请求访问的用户使用。以下部分详细介绍了该过程：

步骤 1 - 在目标 PC 上安装 Linux VDA

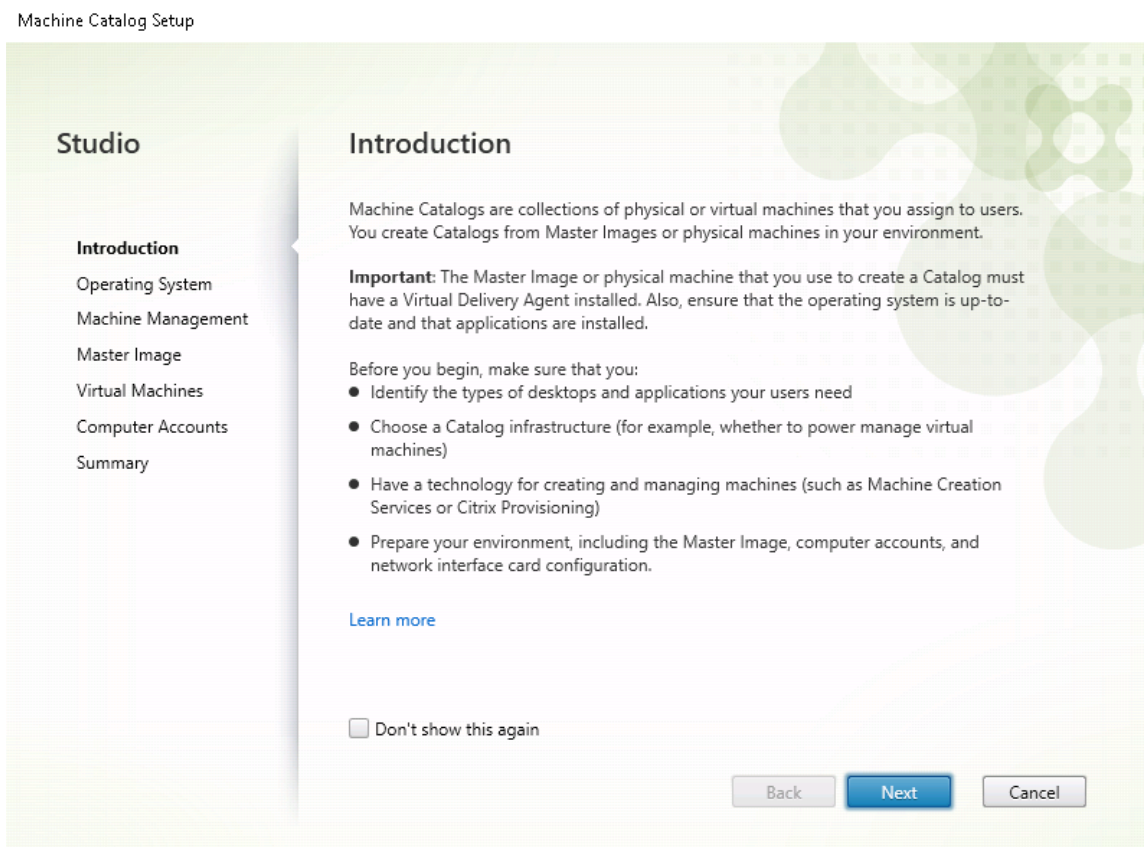
建议您使用[轻松安装](#)来安装 Linux VDA。在安装过程中，将 `CTX_XDL_VDI_MODE` 变量的值设置为 Y。

步骤 2 - 创建 **Remote PC Access** 类型的计算机目录

1. 在 Citrix Studio 中，右键单击计算机目录，然后从快捷菜单中选择创建计算机目录。

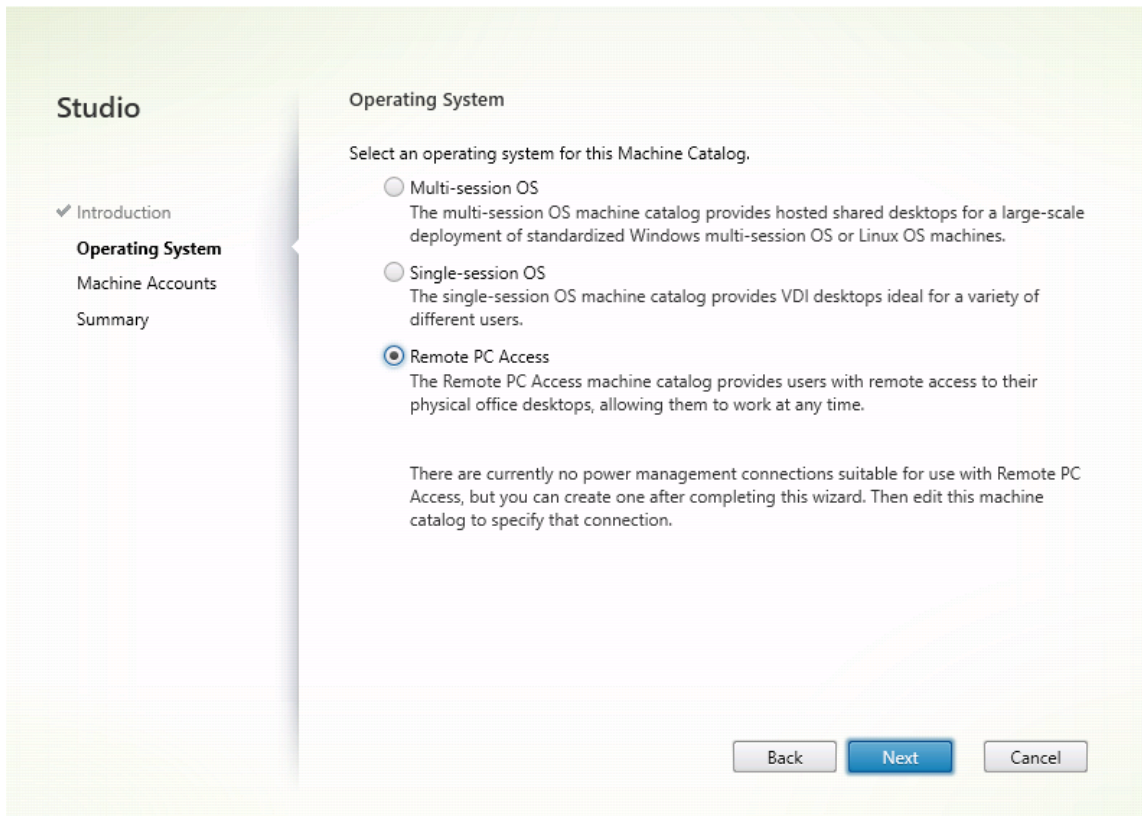


2. 单击简介页面上的下一步。



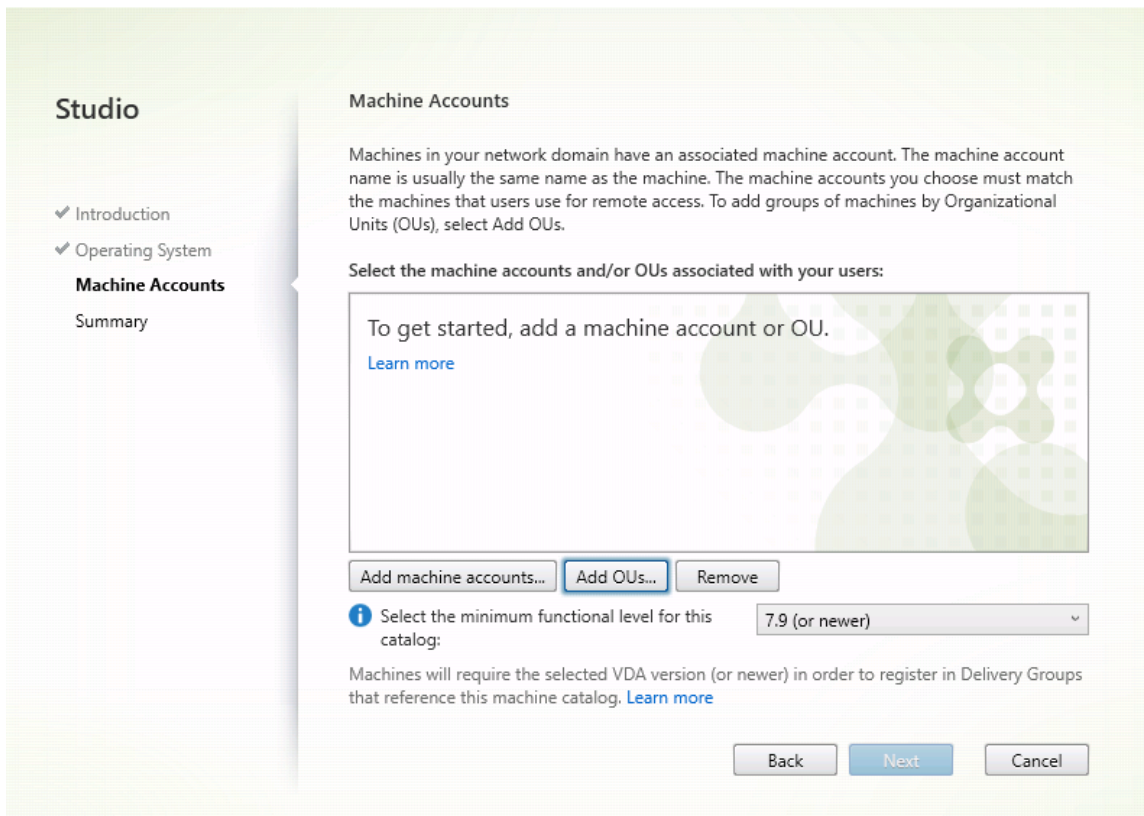
3. 在操作系统页面上选择 **Remote PC Access**。

Machine Catalog Setup



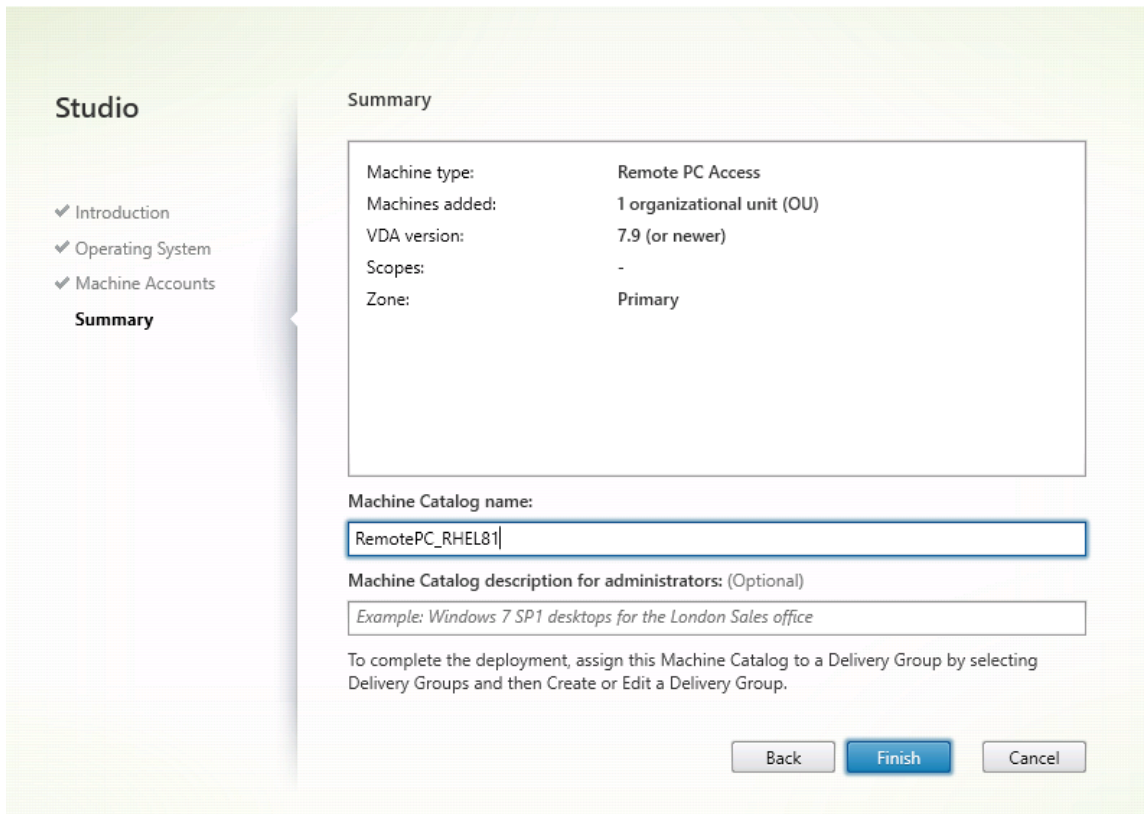
4. 单击添加 **OU** 以选择包含目标 PC 的 OU，或者单击添加计算机帐户将单个计算机添加到计算机目录中。

Machine Catalog Setup

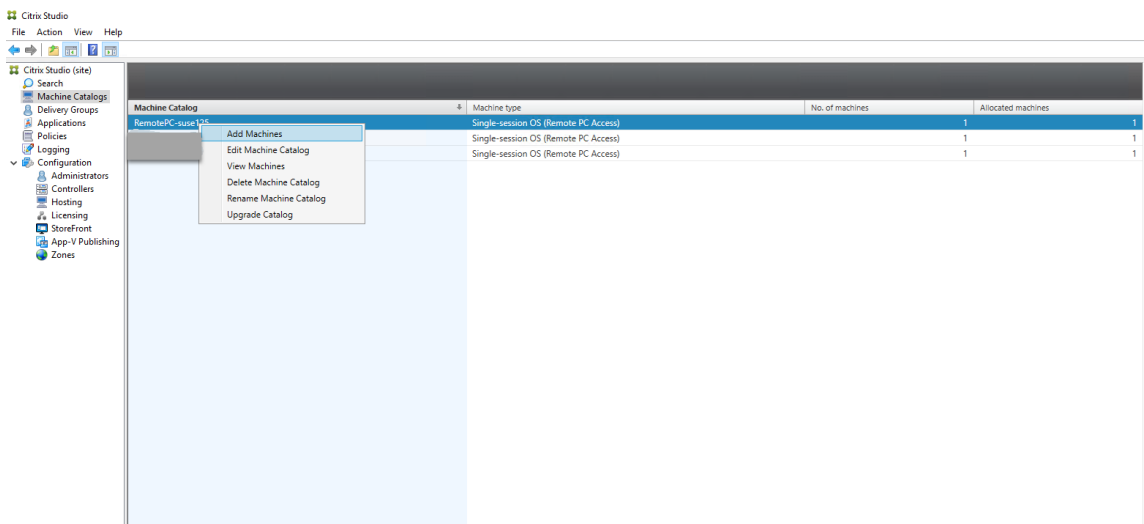


5. 命名计算机目录。

Machine Catalog Setup

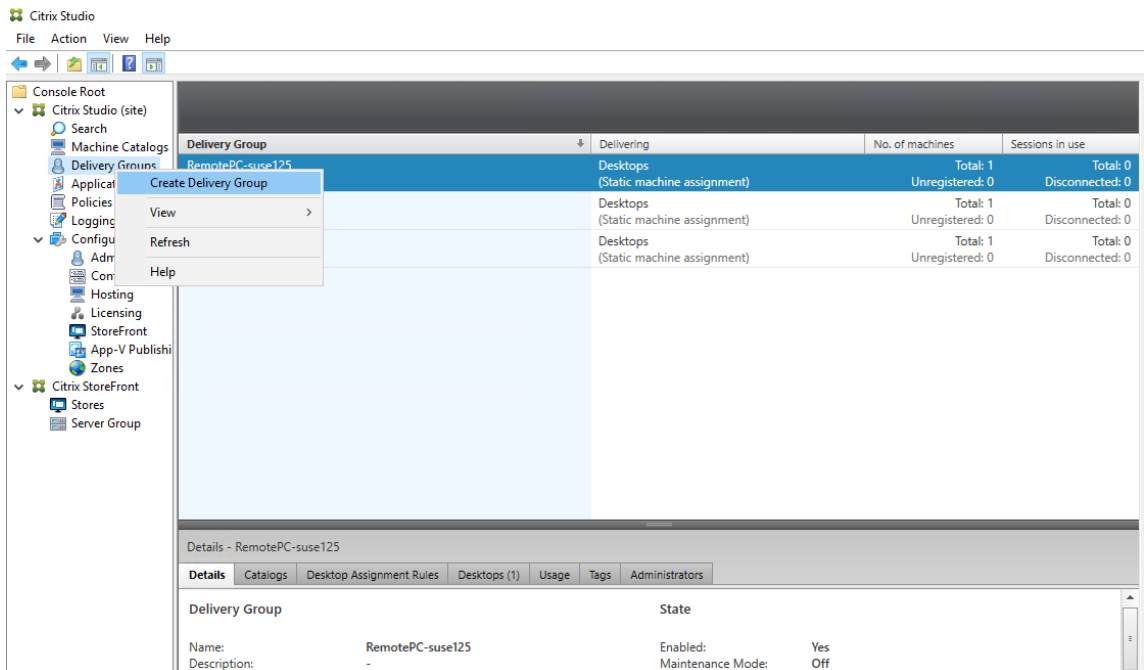


6. (可选) 右键单击计算机目录以执行相关操作。

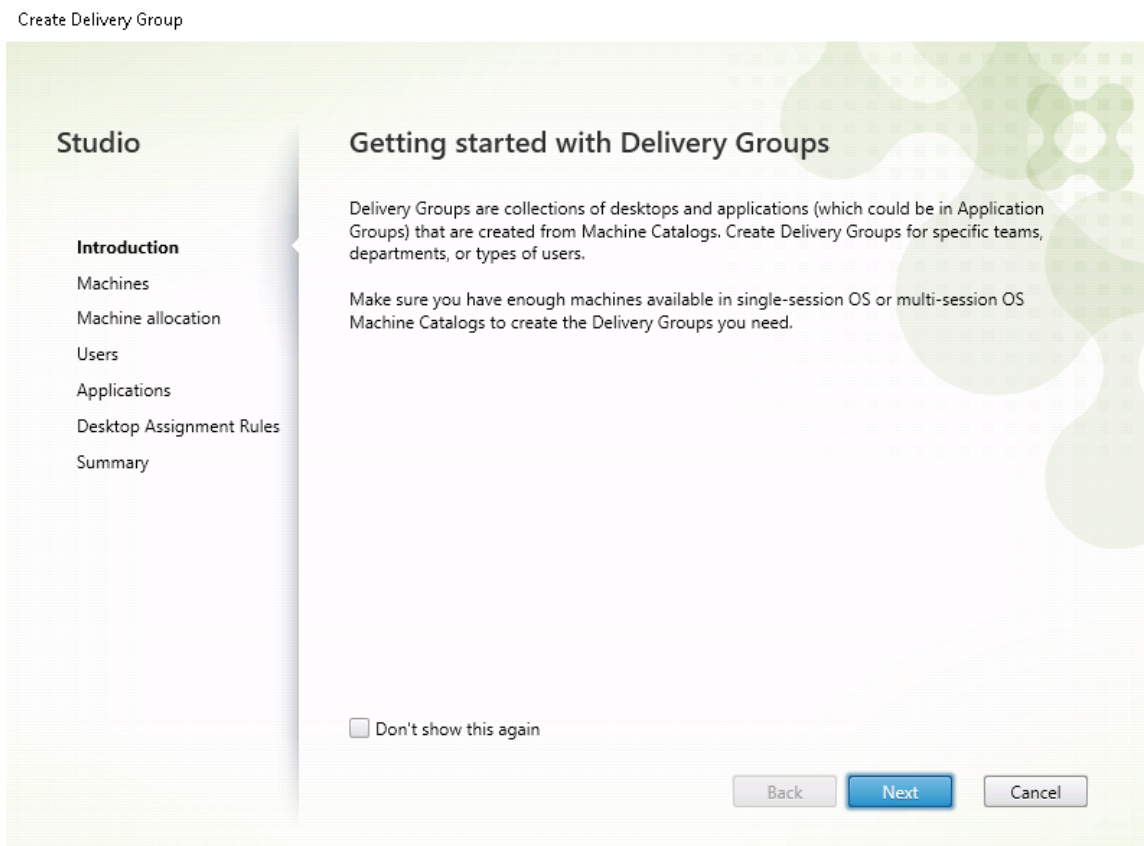


步骤 3 - 创建交付组，以使计算机目录中的 PC 可供请求访问权限的用户使用

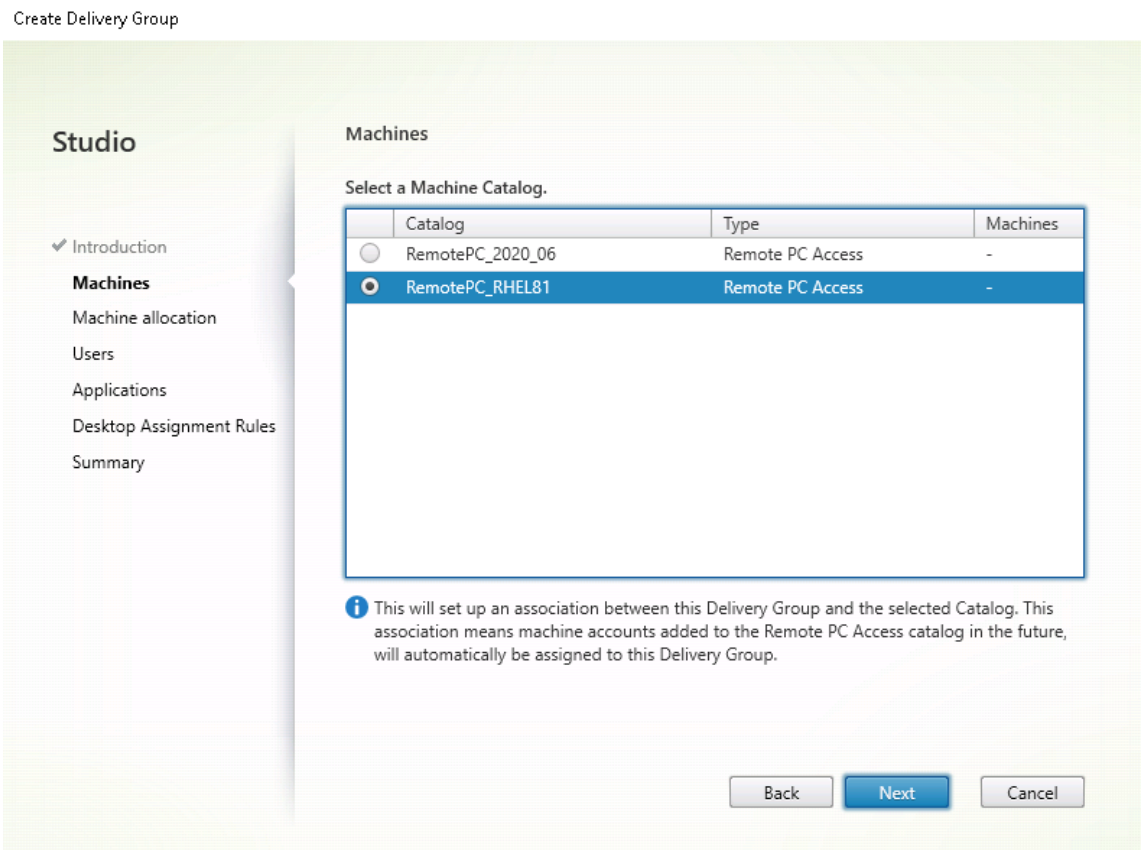
1. 在 Citrix Studio 中，右键单击交付组，然后从快捷菜单中选择创建交付组。



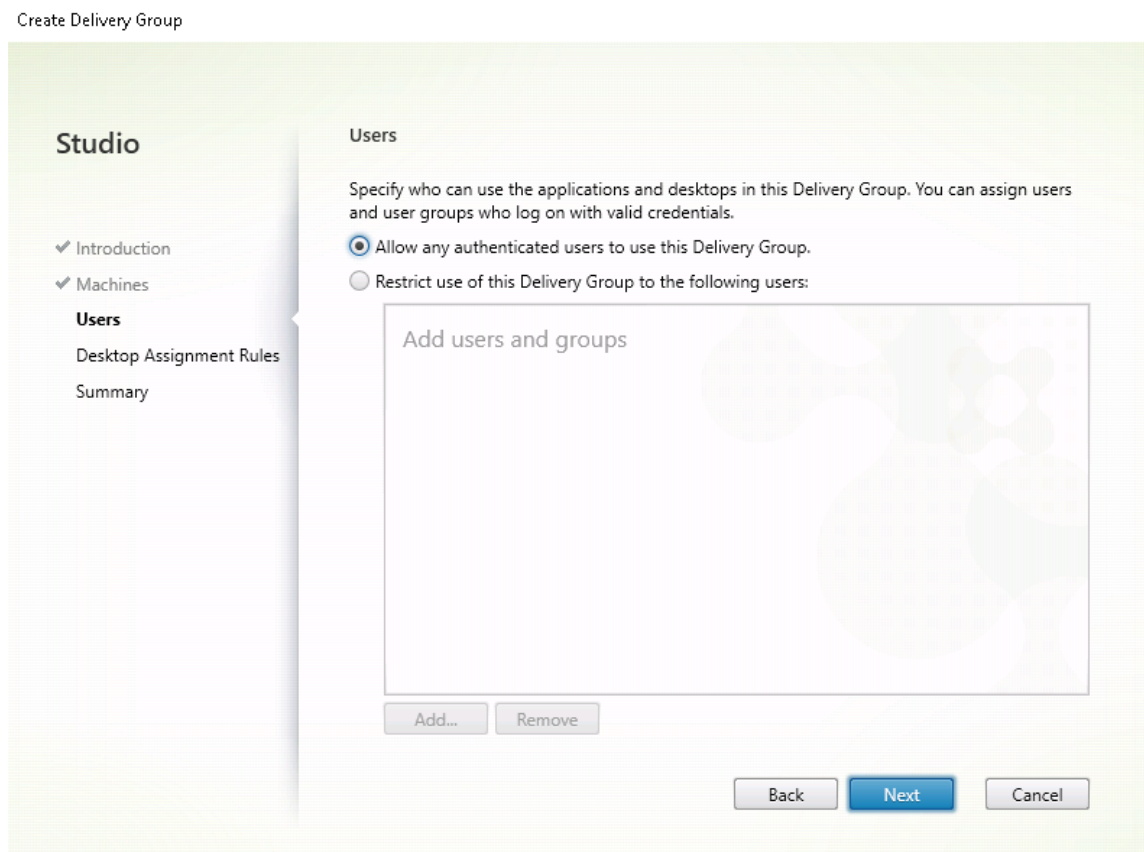
2. 单击交付组入门页面上的下一步。



3. 选择在步骤 2 中创建的计算机目录，以将其与交付组关联。



4. 添加可以访问计算机目录中 PC 的用户。添加的用户可以使用客户端设备上的 Citrix Workspace 应用程序远程访问 PC。



局域网唤醒

Remote PC Access 支持局域网唤醒功能，用户可以使用此功能远程开启物理 PC。借助此功能，用户可以在办公室 PC 不使用时将其关闭，以节约能源成本。用户还可以在计算机意外关闭时进行远程访问。

借助局域网唤醒功能，幻数据包会在 Delivery Controller 指示时直接从 PC 上运行的 VDA 发送到 PC 所在的子网。这允许此功能在不依赖额外的基础结构组件或第三方解决方案的情况下运行，以便传送幻数据包。

局域网唤醒功能不同于传统的基于 SCCM 的局域网唤醒功能。有关基于 SCCM 的局域网唤醒的信息，请参阅 [局域网唤醒-SCCM 集成](#)。

系统要求

下面是使用局域网唤醒功能的系统要求：

- 控制平面：
 - Citrix DaaS（以前称为 Citrix Virtual Apps and Desktops 服务）
 - Citrix Virtual Apps and Desktops 2012 或更高版本
- 物理 PC：

- VDA 版本 2012 或更高版本
- 在 BIOS 中和 NIC 上启用了局域网唤醒

配置局域网唤醒

目前，仅在使用 PowerShell 时支持集成的局域网唤醒的配置。

要配置局域网唤醒，请执行以下操作：

1. 如果您还没有 Remote PC Access 计算机目录，请创建一个目录。
2. 如果您还没有局域网唤醒主机连接，请创建一个连接。

注意：

要使用局域网唤醒功能，如果您具有“Microsoft 配置管理器局域网唤醒”类型的主机连接，请创建一个主机连接。

3. 检索局域网唤醒主机连接的唯一标识符。
4. 将局域网唤醒主机连接与计算机目录相关联。

要创建局域网唤醒主机连接，请执行以下操作：

```
1 # Load Citrix SnapIns
2 Add-PSSnapIn -Name "*citrix*"
3
4 # Provide the name of the Wake on LAN host connection
5 [string]$connectionName = "Remote PC Access Wake on LAN"
6
7 # Create the hypervisor connection
8 $hypHc = New-Item -Path xdhyp:\Connections `
9     -Name $connectionName `
10    -HypervisorAddress "N/A" `
11    -UserName "woluser" `
12    -Password "wolpwd" `
13    -ConnectionType Custom `
14    -PluginId VdaWOLMachineManagerFactory `
15    -CustomProperties "<CustomProperties>/"
16    -Persist
17
18 $bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
19     $hypHc.HypervisorConnectionUid
20 # Wait for the connection to be ready before trying to use it
21 while (-not $bhc.IsReady)
22 {
23
24     Start-Sleep -s 5
25     $bhc = Get-BrokerHypervisorConnection -
26         HypHypervisorConnectionUid $hypHc.HypervisorConnectionUid
```

```
26 }
```

主机连接准备就绪后，运行以下命令以检索主机连接的唯一标识符：

```
1 $bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>"
2 $hypUId = $bhc.Uid
```

检索连接的唯一标识符后，运行以下命令以将连接与 Remote PC Access 计算机目录相关联：

```
1 Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog -
   RemotePCHypervisorConnectionUId $hypUId
```

5. 请在计算机目录中的每个 VM 上的 BIOS 中和 NIC 上启用局域网唤醒。

注意：启用局域网唤醒的方法因计算机配置的不同而异。

- 要在 BIOS 中启用局域网唤醒，请执行以下操作：
 - a) 进入 BIOS 并启用局域网唤醒功能。
访问 BIOS 的方法取决于主板的制造商和制造商选择的 BIOS 供应商。
 - b) 保存您的设置并重新启动计算机。
- 要在 NIC 上启用局域网唤醒，请执行以下操作：
 - a) 运行 `sudo ethtool <NIC>` 命令以检查您的 NIC 是否支持幻数据包。
<NIC> 是您的 NIC 的设备名称，例如 `eth0`。`sudo ethtool <NIC>` 命令提供有关 NIC 功能的输出：
 - 如果输出中包含一个类似于 `Supports Wake-on: <letters>` 的行（其中 <letters> 包含字母 `g`），您的 NIC 将支持局域网唤醒幻数据包方法。
 - 如果输出中包含一个类似于 `Wake-on: <letters>` 的行（其中 <letters> 包含字母 `g`，但不包含字母 `d`），则局域网唤醒幻数据包方法已启用。但是，如果 <letters> 包含字母 `d`，则表示局域网唤醒功能已禁用。在这种情况下，请通过运行 `sudo ethtool -s <NIC> wol g` 命令启用局域网唤醒。
 - b) 在大多数发行版中，每次启动后都需要运行 `sudo ethtool -s <NIC> wol g` 命令。要永久设置此选项，请根据您的发行版完成以下步骤：

Ubuntu:

将 `up ethtool -s <NIC> wol g` 行添加到接口配置文件 `/etc/network/interfaces` 中。例如：

```
1 # ifupdown has been replaced by netplan(5) on this system.
   See
2 # /etc/netplan for current configuration.
3 # To re-enable ifupdown on this system, you can run:
4 # sudo apt install ifupdown
5 auto eth0
6 iface eth0 inet static
7     address 10.0.0.1
8     netmask 255.255.240.0
```

```
9 gateway 10.0.0.1
10 up ethtool -s eth0 wol g
```

RHEL/SUSE:

将以下 `ETHTOOL_OPTS` 参数添加到接口配置文件 `/etc/sysconfig/network-scripts/ifcfg-<NIC>` 中:

```
1 ETHTOOL_OPTS="-s ${
2 DEVICE }
3 wol g"
```

设计注意事项

当您计划在 Remote PC Access 中使用局域网唤醒时，请注意以下事项：

- 多个计算机目录可以使用相同的局域网唤醒主机连接。
- 要使一台 PC 唤醒另一台 PC，两台 PC 必须位于同一子网中，并使用相同的局域网唤醒主机连接。这些 PC 是在相同还是不同的计算机目录中并不重要。
- 将主机连接分配给特定区域。如果您的部署中包含多个区域，则需要每个区域中使用局域网唤醒主机连接。这同样适用于计算机目录。
- 幻数据包使用全局广播地址 255.255.255.255 进行广播。确保该地址未被阻止。
- 子网中必须至少打开一台 PC（对于每个局域网唤醒连接），才能唤醒该子网中的计算机。

操作注意事项

下面是使用局域网唤醒功能的注意事项：

- VDA 必须至少注册一次，才能使用集成的局域网唤醒功能唤醒 PC。
- 局域网唤醒功能只能用于唤醒 PC。该功能不支持其他电源操作，例如重新启动或关闭。
- 创建局域网唤醒连接后，该功能在 Studio 中可见。但是，不支持在 Studio 中编辑其属性。
- 幻数据包通过以下两种方式之一发送：
 - 当用户尝试启动到其 PC 的会话并且 VDA 未注册时
 - 当管理员从 Studio 或 PowerShell 手动发送打开电源命令时
- 由于 Delivery Controller 不知道 PC 的电源状态，因此，Studio 在电源状态下显示不支持。Delivery Controller 使用 VDA 注册状态来确定 PC 是打开还是关闭。

更多资源

下面是 Remote PC Access 的其他资源：

- 解决方案设计指南：[Remote PC Access 设计决策](#)。
- Remote PC Access 体系结构的示例：[Citrix Remote PC Access 解决方案的参考体系结构](#)。

会话

July 15, 2024

本部分内容包含以下主题：

- [自适应传输](#)
- [使用临时主目录登录](#)
- [发布应用程序](#)
- [会话可靠性](#)
- [Rendezvous V1](#)
- [Rendezvous V2](#)
- [使用 TLS 保护用户会话安全](#)
- [使用 DTLS 保护用户会话安全](#)

自适应传输

July 15, 2024

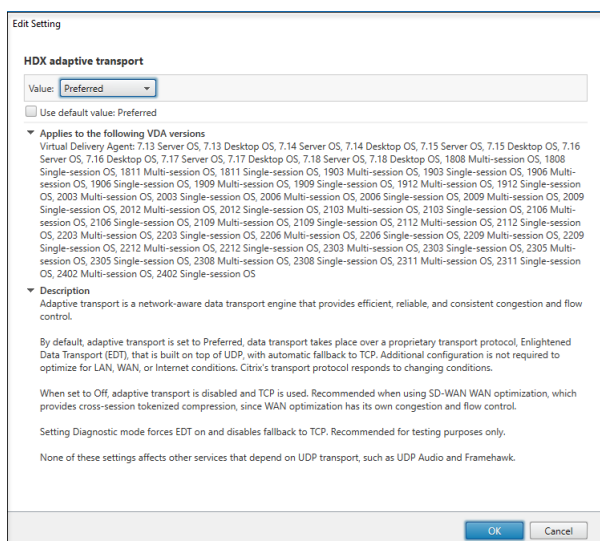
自适应传输是 Citrix Virtual Apps and Desktops 中可以使用 Enlightened Data Transport (EDT) 作为 ICA 连接的传输协议的机制。当 EDT 不可用时，自适应传输将切换到 TCP。

EDT 是基于用户数据报协议 (UDP) 构建的 Citrix 专有传输协议。它在保持服务器可扩展性的同时，在具有挑战性的长途连接方面提供了出色的用户体验。EDT 提高了不可靠网络中所有 ICA 虚拟通道的数据吞吐量，从而提供更出色、更一致的用户体验。

有关详细信息，请参阅 Citrix Virtual Apps and Desktops 文档中的[自适应传输](#)。

启用或禁用自适应传输

默认情况下启用自适应传输。可以使用 **HDX** 自适应传输策略设置配置以下选项：



- 首选：自适应传输处于启用状态，它使用 Enlightened Data Transport (EDT) 作为首选传输协议，并回退到 TCP。
- 诊断模式：自适应传输已启用，并强制使用 EDT。已禁用回退到 TCP。仅建议对测试和故障排除使用此设置。
- 关。自适应传输已禁用，只有 TCP 用于传输。

检查是否正在使用自适应传输

要检查是否将 EDT 用作当前会话的传输协议，请运行以下命令。

```
1 /opt/Citrix/VDA/bin/ctxquery -f iP
```

如果使用 EDT，则显示的传输协议包括 UDP，例如：

```
root@j1-u20: ~/Desktop$ ctxquery -f iP
SESSION:ID      TRANSPORT PROTOCOLS  RENDEZVOUS
j1-u20:0        -                    -
j1-u20:1        -                    -
j1-u20:2        -                    -
j1-u20:12      UDP-CGP-ICA         NONE
root@j1-u20: ~/Desktop$
```

EDT MTU 发现

MTU 发现允许 EDT 在建立会话时自动确定最大传输单位 (MTU)。这样做可以防止出现可能会导致性能下降或无法建立会话的 EDT 数据包碎片。

系统要求：

- Linux VDA 最低版本 2012

- Citrix Workspace 应用程序：
 - Windows: 1911 或更高版本
- Citrix ADC：
 - 13.0.52.24 或更高版本
 - 12.1.56.22 或更高版本
- 必须启用会话可靠性

如果您使用的是不支持此功能的客户端平台或版本，请参阅 [CTX231821](#) 以了解有关配置适合您的环境的自定义 EDT MTU 的详细信息。

控制 VDA 上的 EDT MTU 发现

EDT MTU 发现在 VDA 上默认处于启用状态。可以通过设置 `MtuDiscovery` 注册表项将其启用或禁用，如下所示：

- 要启用 EDT MTU 发现，请使用以下命令设置 `MtuDiscovery` 注册表项，重新启动 VDA，然后等待 VDA 注册：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t "REG_DWORD" -v "MtuDiscovery" -d "0x00000001" --force
```

- 要禁用 EDT MTU 发现，请删除 `MtuDiscovery` 注册表值。

警告：

注册表编辑不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因注册表编辑器使用不当导致出现的问题能够得以解决。使用注册表编辑器需自担风险。在编辑注册表之前，请务必进行备份。

控制客户端上的 EDT MTU 发现

可以通过在 ICA 文件中添加 `MtuDiscovery` 参数，在客户端上有选择地控制 EDT MTU 发现。要禁用此功能，请在 `Application` 部分下设置以下策略：

`MtuDiscovery=Off`

要重新启用此功能，请从 ICA 文件中删除 `MtuDiscovery` 参数。

重要：

要使此 ICA 文件参数起作用，请在 VDA 上启用 EDT MTU 发现。如果未在 VDA 上启用 EDT MTU 发现，则 ICA 文件参数无效。

增强的 EDT 拥塞控制

引入了一种拥塞控制算法来优化 EDT 协议。这种实现允许 EDT 实现更高的吞吐量并减少延迟，从而增强用户体验。

默认情况下启用此功能。要将其禁用并重新启用，请分别运行以下命令，然后重新启动 `ctxhdx` 服务：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d "0x00000000" --force
```

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd\Tds\udp\UDPStackParameters" -t "REG_DWORD" -v "edtBBR" -d "0x00000001" --force
```

HDX 自适应吞吐量

August 20, 2024

HDX 自适应吞吐量通过调整输出缓冲区智能地微调 ICA 会话的高峰吞吐量。输出缓冲区的数量最初设置为较高的值。这一较高的值允许更快更高效地将数据传输到客户端，尤其是在高延迟网络中。此功能带来了增强的用户体验。它提供了更好的交互性、更快的文件传输、更流畅的视频播放以及更高的帧速率和分辨率。将持续测量会话交互性以确定 ICA 会话中的任何数据流是否会对交互性产生不利影响。如果出现这种情况，吞吐量将减少，以降低大量数据流对会话产生的影响并允许恢复交互性。

重要：

HDX 自适应吞吐量通过将此机制从客户端移到 VDA 改变了设置输出缓冲区的方式。无需手动配置。

该功能要求 VDA 最低版本 2311。默认启用此功能。如果它被禁用，则可以通过在 VDA 上运行以下命令来启用它：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd" -t "REG_DWORD" -v "AdaptiveScalingEnabled" -d "0x00000001" --force
```

该功能仅应用到启用该功能后启动的会话。

会话登录屏幕上的自定义背景和横幅消息

July 15, 2024

向会话登录屏幕添加自定义背景或横幅消息

提示:

要在 SUSE 15.5 上使用此功能, 请从 <http://download.opensuse.org/distribution/leap/15.3/repo/oss/> 安装 `imlib2`。

可以使用以下命令将自定义背景消息或横幅消息添加到会话登录屏幕。要向会话登录屏幕添加背景消息和横幅消息, 可以将横幅消息嵌入到背景图像中。打开会话后, 首先显示横幅消息页面, 然后出现身份验证对话框。

要设置自定义横幅消息的标题, 请运行:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "LogonDisplayStringTitle" -d "<Banner message title>" --force
```

横幅消息标题的最大长度为 **64** 字节。

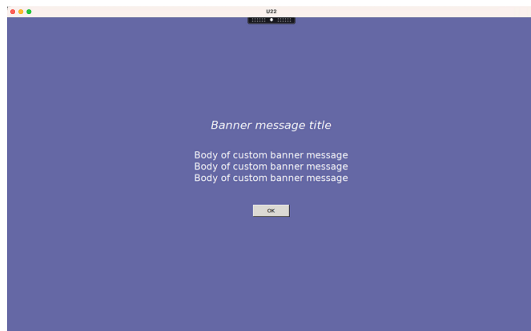
要设置自定义横幅消息的正文文本, 请运行:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "LogonDisplayString" -d "Body of custom banner message\nBody of custom banner message\nBody of custom banner message\n" --force
```

横幅消息正文的最大长度为 **1024** 字节。

提示:

`\n` 元素创建换行符。在此示例中, 横幅消息屏幕如下所示:



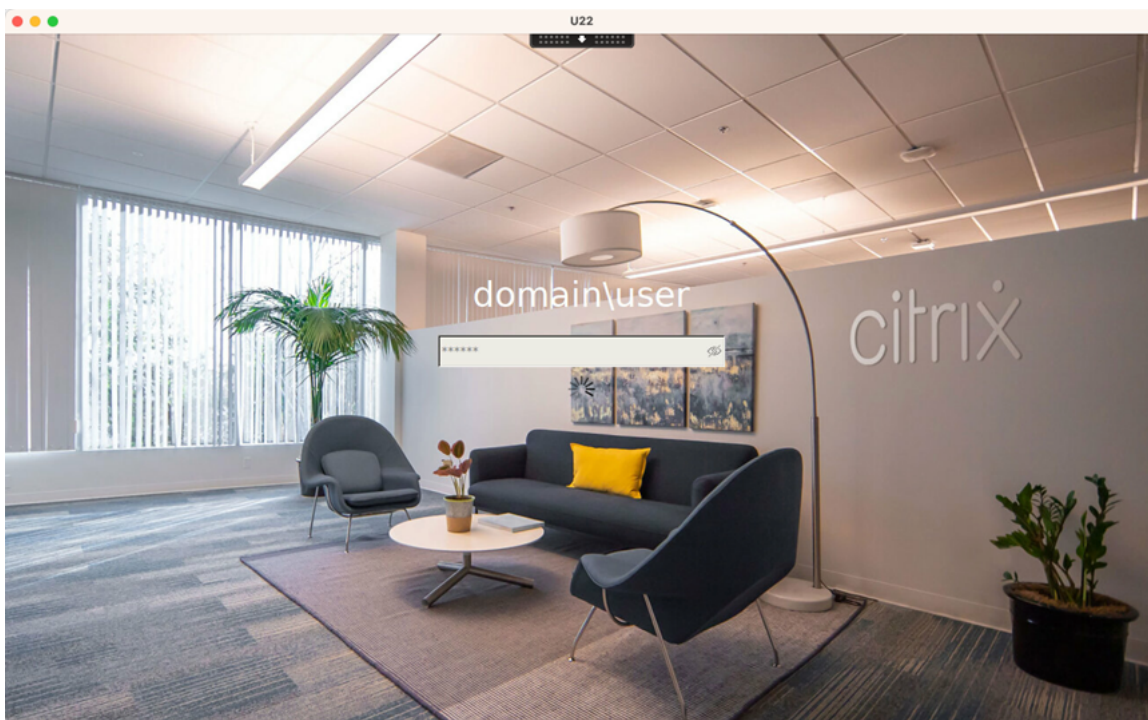
为了确保您的自定义横幅消息按预期显示, 您可能需要使用转义字符。例如, 请在 HTML 中使用 `&` 表示 `&`, 并在字符 `!` 前面放置一个反斜杠 (`****`) 以在 Bash 命令中对其进行转义。

要向会话登录屏幕中添加自定义背景, 请运行:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix" -t "REG_SZ" -v "BackgroundImagePath" -d "<path to the background image>" --force
```

要查看自定义背景, 会话用户必须有权访问背景图像路径。

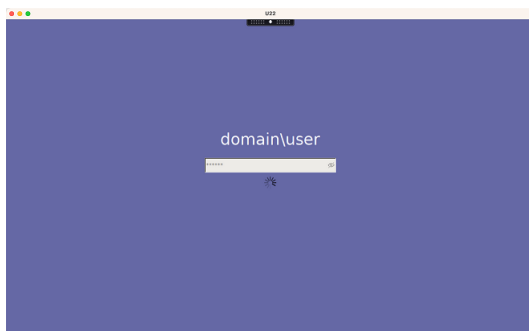
例如:



会话登录屏幕示例

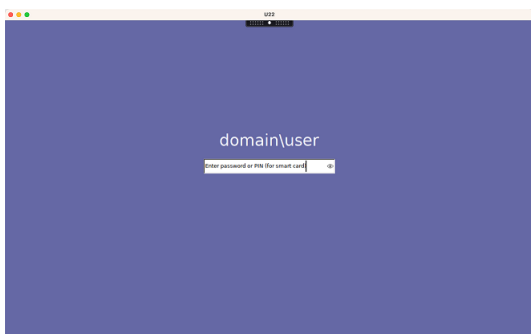
下面是不同场景中的会话登录屏幕示例：

- 单点登录 (SSO) 场景中的会话登录：



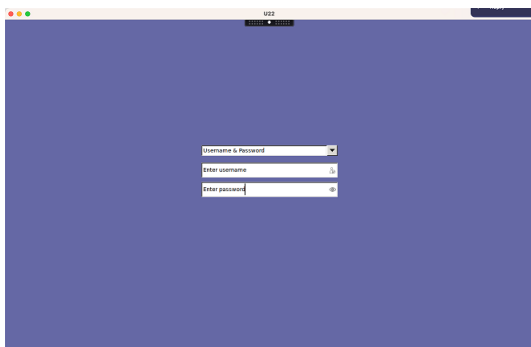
显示登录过程。

- 典型的非 SSO 场景中的会话登录：

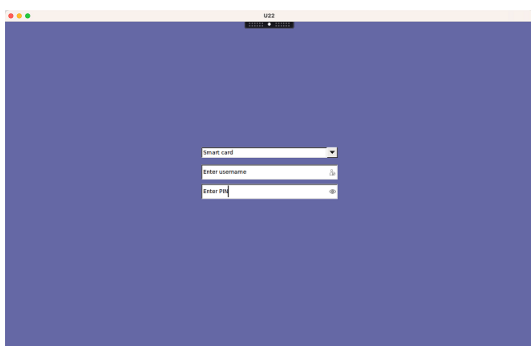


- 在非 SSO 场景中需要密码或 PIN 码。
 - 用户可以切换密码和 PIN 码的可见性，使用户能够轻松找出错误的输入。
- 当用户使用与登录到 Citrix Workspace 应用程序时使用的凭据不同的凭据登录 VDA 会话时，非 SSO 场景中的会话登录：

用于会话登录的用户名和密码：



用于会话登录的智能卡：



有关非 SSO 场景中支持的用户身份验证方法的组合，请参阅[非 SSO 身份验证](#)。

按会话用户划分的自定义桌面环境

August 20, 2024

可以使用 **CTX_XDL_DESKTOP_ENVIRONMENT** 变量为会话用户指定桌面环境。自 2209 版起，会话用户可以自定义自己的桌面环境。必须事先在 VDA 上安装桌面环境，才能让会话用户使用此功能。

下表显示了支持按会话用户划分的自定义桌面环境的 Linux 发行版和桌面环境的列表。

Linux 发行版	支持的桌面
Debian 12.5	MATE、GNOME、GNOME-Classic、KDE、 Xfce
Debian 11.9	MATE、GNOME、GNOME-Classic、KDE
RHEL 8.10/8.9/8.8	MATE、GNOME、GNOME-Classic、 Xfce
Rocky Linux 8.10/8.9/8.8	MATE、GNOME、GNOME-Classic、KDE、 Xfce
SUSE 15.5	MATE、GNOME、GNOME-Classic
Ubuntu 22.04/20.04	MATE、GNOME、GNOME-Classic、KDE、 Xfce

桌面切换命令

注意：

您可以同时从终端设备和[系统托盘](#)在桌面环境之间切换。

要从终端设备切换到目标桌面环境，请在会话中运行相应的命令：

如果目标桌面环境为：	请运行以下命令：
GNOME	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh GNOME</code>
GNOME Classic	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh GNOME-CLASSIC</code>
MATE	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh MATE</code>
KDE	<code>/opt/Citrix/VDA/bin/ ctxdesktopswitch.sh KDE</code>

KDE 提示

- Magnus 可能会在 KDE 中启动时加载。作为解决方法，您可以通过运行 `sudo apt remove magnus` 来删除 Magnus 软件包。

- 要禁用 KDE 启动期间出现的 QT 警告，请以 root 用户身份通过添加以下条目来编辑 `/usr/share/qt5/qtlogging.ini`:

```
1 qt.qpa.xcb.xcberror.error=false
2 qt.qpa.xcb.warning=false
3 qt.qpa.xcb.error=false
```

- KDE 的屏幕解锁可能会失败。作为解决方法，我们建议您禁用桌面的自动锁定功能。

使用临时主目录登录

July 15, 2024

在 Linux VDA 上的装载点出现故障的情况下，可以指定临时主目录。如果指定了临时主目录，则将在会话登录期间装载点失败时显示一条提示。用户数据之后将存储在临时主目录下。

下表介绍了有助于设置您的主目录的注册表项。

注册表项	说明	命令
LogNoHome	控制用户是否可以在没有主目录的情况下登录会话。默认值为 1，表示是。如果将该值设置为 0，则将禁用不适用主目录的会话登录。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "LogNoHome"-d "0x00000001"--force</pre>
HomeMountPoint	在 Linux VDA 上设置本地装载点。例如，如果 <code>/mnt/home</code> 是装载点，则用户的主目录为 <code>/mnt/home/domain/<user_name></code> 。确保装载点与您的环境中的用户主目录相同。仅当 <code>CheckUserHomeMountPoint</code> 设置为 0 时，此设置才会生效。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_SZ"-v "HomeMountPoint"-d "<A directory where the NFS share is to be mounted>"--force</pre>

注册表项	说明	命令
<code>CheckUserHomeMountPoint</code>	控制是否将用户特定的主目录设置为 Linux VDA 上的装载点。如果要用户特定的主目录设置为装载点，请将该值设置为 1 。默认值为 0 。	<pre>ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "CheckUserHomeMountPoint"-d "0x00000001"--force</pre>
<code>TempHomeDirectoryPath</code>	在 Linux VDA 上设置临时主目录，以防装载点出现故障。默认值为 <code>/tmp</code> 。仅当装载点由确定 <code>HomeMountPoint</code> 且 <code>CheckUserHomeMountPoint</code> 不可用时，临时主目录设置才会生效。用户的临时主目录为 <code>/tmp/CTXSmf_user_id</code> 。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_SZ"-v "TempHomeDirectoryPath"-d "</tmp by default >"--force</pre>
<code>CheckMountPointRetryTime</code>	设置检查安装是否成功的次数，频率为每秒一次。默认值为 5。	<pre>ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "CheckMountPointRetryTime"-d "0x00000010"--force</pre>
<code>RemoveHomeOnLogoff</code>	控制是否在用户注销时删除临时主目录。1 表示是。0 表示否。	<pre>create -k "HKLM\System\CurrentControlSet\Control\Citrix"-t "REG_DWORD"-v "RemoveHomeOnLogoff"-d "0x00000000"--force</pre>

发布应用程序

July 15, 2024

借助 Linux VDA 7.13 版，Citrix 向所有受支持的 Linux 平台中添加了无缝应用程序功能。不需要执行任何特定安装过程即可使用此功能。

提示：

在 Linux VDA 1.4 中，Citrix 增加了对已发布的非无缝应用程序和会话共享功能的支持。

使用 **Citrix Studio** 发布应用程序

您可以在创建交付组或将应用程序添加到现有交付组时发布 Linux VDA 上安装的应用程序。该过程与发布 Windows VDA 上安装的应用程序类似。有关详细信息，请参阅 [Citrix Virtual Apps and Desktops 文档](#)（根据使用的 Citrix Virtual Apps and Desktops 版本）。

注意：

- 配置交付组时，请确保交付类型设置为桌面和应用程序或应用程序。
- 我们建议您为应用程序和桌面交付创建单独的 VDA 和交付组。
- 要使用无缝应用程序，请勿在 StoreFront 上禁用无缝模式。无缝模式默认处于启用状态。如果您已通过设置“TWIMode=Off”禁用该模式，请删除此设置，而非将其更改为“TWIMode=On”。否则可能无法启动已发布的桌面。

限制

Linux VDA 不支持单个用户启动同一应用程序的多个并发实例。

在应用程序会话中，只有特定于应用程序的快捷方式才能按预期运行。

已知问题

在发布应用程序期间发现了以下已知问题：

- 不支持非长方形窗口。窗口的边角可能会显示服务器端背景。
- 不支持从已发布的应用程序预览窗口的内容。
- 运行多个 LibreOffice 应用程序时，只有第一个启动的应用程序显示在 Citrix Studio 上，因为这些应用程序共享进程。
- “Dolphin”之类的基于 Qt5 的已发布应用程序可能不显示图标。要解决此问题，请参阅网址为 <https://wiki.archlinux.org/title/Qt> 的文章。

- Linux 应用程序通常有一个“关于”窗口，其中包含有关正在使用的应用程序的信息，而获取更多信息的 Web 链接通常位于这些窗口中。单击“关于”窗口中的 Web 链接可以从 **calc**、**gedit**、日历和 **LibreOffice** 套件等已发布的应用程序中启动 Web 浏览器。无意中启动 Web 浏览器会绕过应用程序隔离，并且可能会危及安全性。要解决此问题，请通过完成以下步骤来更改默认 Web 浏览器：

1. 在自定义位置中创建 `none.sh` 文件，例如：

```
1 sudo mkdir /home/none
2
3 sudo touch /home/none/none.sh
4
5 sudo chmod +x /home/none/none.sh
```

2. 将以下行添加到 `none.sh` 文件中：

```
1 #!/bin/bash
2
3 echo "NONE"
```

3. 使用 `sudo` 创建 `/etc/xdg/mimeapps.list` 文件，然后将以下行添加到 `mimeapps.list` 文件中：

```
1 [Default Applications]
2
3 text/html=none.desktop
4
5 x-scheme-handler/http=none.desktop
6
7 x-scheme-handler/https=none.desktop
8
9 x-scheme-handler/about=none.desktop
10
11 x-scheme-handler/unknown=none.desktop
```

4. 使用 `sudo` 创建 `/usr/share/applications/none.desktop` 文件，然后将以下行添加到 `none.desktop` 文件中：

```
1 [Desktop Entry]
2
3 Encoding=UTF-8
4
5 Version=1.0
6
7 Type=Application
8
9 Terminal=false
10
11 Exec=/home/none/none.sh
12
13 Name=None
14
15 Icon=/home/none/none.sh
```

由于您可以将 `none.sh` 文件置于自定义位置，因此请确保 `none.desktop` 可以正确引用 `none.sh` 文件。

Rendezvous V1

July 15, 2024

使用 Citrix Gateway 服务时，Rendezvous 协议允许流量绕过 Citrix Cloud Connector，以直接安全地连接到 Citrix Cloud 控制平面。

有两种类型的流量需要考虑：1) 控制 VDA 注册和会话代理的流量；2) HDX 会话流量。

Rendezvous V1 允许 HDX 会话流量绕过 Cloud Connector，但它仍要求 Cloud Connector 代理所有控制流量以进行 VDA 注册和会话代理。

要求

- 使用 Citrix Workspace 和 Citrix Gateway 服务访问环境。
- 控制平面：Citrix DaaS（以前称为 Citrix Virtual Apps and Desktops 服务）。
- Linux VDA 版本 2112 或更高版本。
 - 版本 2112 是非透明 HTTP 代理所需的最低版本。
 - 版本 2204 是透明代理和 SOCKS5 代理所需的最低版本。
- 在 Citrix 策略中启用 Rendezvous 协议。有关详细信息，请参阅 [Rendezvous 协议策略设置](#)。
- VDA 必须对 `https://*.nssvc.net` 具有访问权限，包括所有子域。如果无法通过该方式将所有子域添加到允许列表中，请改为使用 `https://*.c.nssvc.net` 和 `https://*.g.nssvc.net`。有关详细信息，请参阅 Citrix Cloud 文档（在 Virtual Apps and Desktops 服务下方）的 [Internet 连接要求](#) 部分和知识中心文章 [CTX270584](#)。
- 在代理会话时，Cloud Connector 必须获取 VDA 的 FQDN。通过以下两种方式之一完成此任务：
 - 为站点启用 **DNS** 解析。导航到完整配置 > 设置并打开启用 **DNS** 解析设置。或者，使用 Citrix Virtual Apps and Desktops 远程 PowerShell SDK 并运行命令 `Set-BrokerSite -DnsResolutionEnabled $true`。有关 Citrix Virtual Apps and Desktops 远程 PowerShell SDK 的详细信息，请参阅 [SDK 和 API](#)。
 - 具有 **VDA** 的 **PTR** 记录的 **DNS** 反向查找区域。如果选择此选项，我们建议您将 VDA 配置为始终尝试注册 PTR 记录。为此，请使用组策略编辑器或组策略对象，导航到计算机配置 > 管理模板 > 网络 > **DNS** 客户端，然后将注册 **PTR** 记录设置为已启用并注册。如果连接的 DNS 后缀与域的 DNS 后缀不匹配，您还必须配置 **Connection-specific DNS suffix**（连接特有的 DNS 后缀）设置，以便计算机能够成功注册 PTR 记录。

注意：

如果使用 DNS 解析选项，则 Cloud Connector 必须能够解析 VDA 计算机的完全限定域名 (FQDN)。在

内部用户直接连接到 VDA 计算机的情况下，客户端设备还必须能够解析 VDA 计算机的 FQDN。

如果使用 DNS 反向查找区域，则 PTR 记录中的 FQDN 必须与 VDA 计算机的 FQDN 相匹配。如果 PTR 记录包含不同的 FQDN，则汇聚连接将失败。例如，如果计算机的 FQDN 是 `vda01.domain.net`，则 PTR 记录必须包含 `vda01.domain.net`。使用其他 FQDN（例如 `vda01.sub.domain.net`）将不起作用。

代理配置

VDA 支持通过 HTTP 和 SOCKS5 代理建立 Rendezvous 连接。

代理注意事项

在 Rendezvous 中使用代理时，请注意以下事项：

- 支持非透明 HTTP 代理和 SOCKS5 代理。
- 不支持数据包解密和检查。配置异常，以便 VDA 与 Gateway 服务之间的 ICA 流量不会被拦截、解密或检查。否则，连接会中断。
- HTTP 代理通过使用协商和 Kerberos 身份验证协议支持基于计算机的身份验证。连接到代理服务器时，协商身份验证方案会自动选择 Kerberos 协议。Kerberos 是 Linux VDA 支持的唯一方案。

注意：

要使用 Kerberos，必须为代理服务器创建服务主体名称 (SPN)，并将其与代理的 Active Directory 帐户关联。VDA 在建立会话时生成格式为 `HTTP/<proxyURL>` 的 SPN，其中代理 URL 是从 **Rendezvous** 代理策略设置中检索的。如果不创建 SPN，身份验证将失败。

- 当前不支持使用 SOCKS5 代理进行身份验证。如果使用 SOCKS5 代理，必须配置例外，以便发往 Gateway 服务地址的流量（在要求中指定）可以绕过身份验证。
- 只有 SOCKS5 代理支持通过 EDT 进行数据传输。对于 HTTP 代理，请使用 TCP 作为 ICA 的传输协议。

透明代理

Rendezvous 支持透明 HTTP 代理。如果在网络中使用透明代理，则不需要在 VDA 上进行其他配置。

非透明代理

在网络中使用非透明代理时，请配置 **Rendezvous 代理配置** 设置。启用该设置后，为 VDA 指定 HTTP 或 SOCKS5 代理地址，以了解要使用哪个代理。例如：

- 代理地址：`http://<URL or IP>:<port>` 或 `socks5://<URL or IP>:<port>`

8. Citrix Gateway 服务将连接信息发送到 Cloud Connector。Cloud Connector 确定连接是否为 Rendezvous 连接，并将信息发送到 VDA。
9. VDA 建立与 Citrix Gateway 服务的直接连接。
10. 如果无法在 VDA 与 Citrix Gateway 服务之间建立直接连接，VDA 将通过 Cloud Connector 代理其连接。
11. Citrix Gateway 服务在端点与 VDA 之间建立连接。
12. VDA 通过 Cloud Connector 借助 Citrix DaaS 验证其许可证。
13. Citrix DaaS 通过 Cloud Connector 向 VDA 发送会话策略。这些策略已应用。

Rendezvous V2

July 15, 2024

使用 Citrix Gateway 服务时，Rendezvous 协议允许流量绕过 Citrix Cloud Connector，以直接安全地连接到 Citrix Cloud 控制平面。

有两种类型的流量需要考虑：1) 控制 VDA 注册和会话代理的流量；2) HDX 会话流量。

Rendezvous V1 允许 HDX 会话流量绕过 Cloud Connector，但它仍要求 Cloud Connector 代理所有控制流量以进行 VDA 注册和会话代理。

支持加入了标准 AD 域的计算机和未加入域的计算机将 Rendezvous V2 与单会话和多会话 Linux VDA 配合使用。对于未加入域的计算机，Rendezvous V2 允许 HDX 流量和控制流量绕过 Cloud Connector。

要求

使用 Rendezvous V2 的要求是：

- 使用 Citrix Workspace 和 Citrix Gateway 服务访问环境。
- 控制平面：Citrix DaaS（以前称为 Citrix Virtual Apps and Desktops 服务）。
- VDA 版本 2201 或更高版本。
 - 版本 2204 是 HTTP 和 SOCKS5 代理所需的最低版本。
- 在 Citrix 策略中启用 Rendezvous 协议。有关详细信息，请参阅 [Rendezvous 协议策略设置](#)。
- VDA 必须有权访问：
 - TCP 443 上的 https://*.xendesktop.net。如果您无法以该方式允许所有子域，则可以使用 https://<customer_ID>.xendesktop.net，其中 <customer_ID> 为 Citrix Cloud 客户 ID，如 Citrix Cloud 管理员门户中所示。

- https://*.nssvc.net, 包括所有子域。如果您无法通过该方式将所有子域列入白名单, 请改为使用 https://*.c.nssvc.net 和 https://*.g.nssvc.net。有关详细信息, 请参阅 Citrix Cloud 文档 (在 Virtual Apps and Desktops 服务下方) 的 [Internet 连接要求](#) 部分和知识中心文章 [CTX270584](#)。
- VDA 必须能够连接到上文提到的地址:
 - 在 TCP 443 上, 用于 TCP Rendezvous。
 - 在 UDP 443 上, 用于 EDT Rendezvous。

代理配置

使用 Rendezvous 时, VDA 支持通过代理连接控制流量和 HDX 会话流量。两种类型的流量的要求和注意事项不同, 因此请仔细查看。

控制流量代理注意事项

- 仅支持 HTTP 代理。
- 不支持数据包解密和检查。配置例外, 以便 VDA 和 Citrix Cloud 控制平面之间的控制流量不会被拦截、解密或检查。否则, 连接将失败。
- 不支持代理身份验证。
- 要为控制流量配置代理, 请按如下方式编辑注册表:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "ProxySettings" -d "http://<URL or IP>:<port>" --force
```

HDX 流量代理注意事项

- 支持 HTTP 和 SOCKS5 代理。
- EDT 只能与 SOCKS5 代理一起使用。
- 要为 HDX 流量配置代理, 请使用 [Rendezvous 代理配置策略](#) 设置。
- 不支持数据包解密和检查。配置例外, 以便 VDA 和 Citrix Cloud 控制平面之间的 HDX 流量不会被拦截、解密或检查。否则, 连接将失败。
- HTTP 代理通过使用协商和 Kerberos 身份验证协议支持基于计算机的身份验证。连接到代理服务器时, 协商身份验证方案会自动选择 Kerberos 协议。Kerberos 是 Linux VDA 支持的唯一方案。

注意：

要使用 Kerberos，必须为代理服务器创建服务主体名称 (SPN)，并将其与代理的 Active Directory 帐户关联。VDA 在建立会话时生成格式为 `HTTP/<proxyURL>` 的 SPN，其中代理 URL 是从 **Rendezvous** 代理策略设置中检索的。如果不创建 SPN，身份验证将失败。

- 当前不支持使用 SOCKS5 代理进行身份验证。如果使用 SOCKS5 代理，必须配置例外，以便发往 Gateway 服务地址的流量（在要求中指定）可以绕过身份验证。
- 只有 SOCKS5 代理支持通过 EDT 进行数据传输。对于 HTTP 代理，请使用 TCP 作为 ICA 的传输协议。

透明代理

Rendezvous 支持透明 HTTP 代理。如果在网络中使用透明代理，则不需要在 VDA 上进行其他配置。

如何配置 Rendezvous V2

下面是在您的环境中配置 Rendezvous 的步骤：

1. 请确保满足[所有要求](#)。
2. 安装 VDA 后，运行以下命令以设置所需的注册表项：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "GctRegistration" -d "0x00000001" --force
```

3. 重新启动 VDA 计算机。
4. 创建 Citrix 策略或者编辑现有策略：
 - 将 Rendezvous 协议设置设置为允许。默认情况下，Rendezvous 协议处于禁用状态。启用（允许）Rendezvous 协议后，Rendezvous V2 而非 V1 将生效。
 - 请务必正确设置 Citrix 策略过滤器。该策略适用于需要启用 Rendezvous 的计算机。
 - 请确保 Citrix 策略具有正确的优先级，以免覆盖其他策略。

Rendezvous 验证

要检查会话是否正在使用 Rendezvous 协议，请在终端中运行 `/opt/Citrix/VDA/bin/ctxquery -f iP` 命令。

显示的传输协议指明连接类型：

- TCP Rendezvous: TCP - TLS - CGP - ICA
- EDT Rendezvous: UDP - DTLS - CGP - ICA

- 通过 Cloud Connector 代理: TCP - PROXY - SSL - CGP - ICA 或 UDP - PROXY - DTLS - CGP - ICA

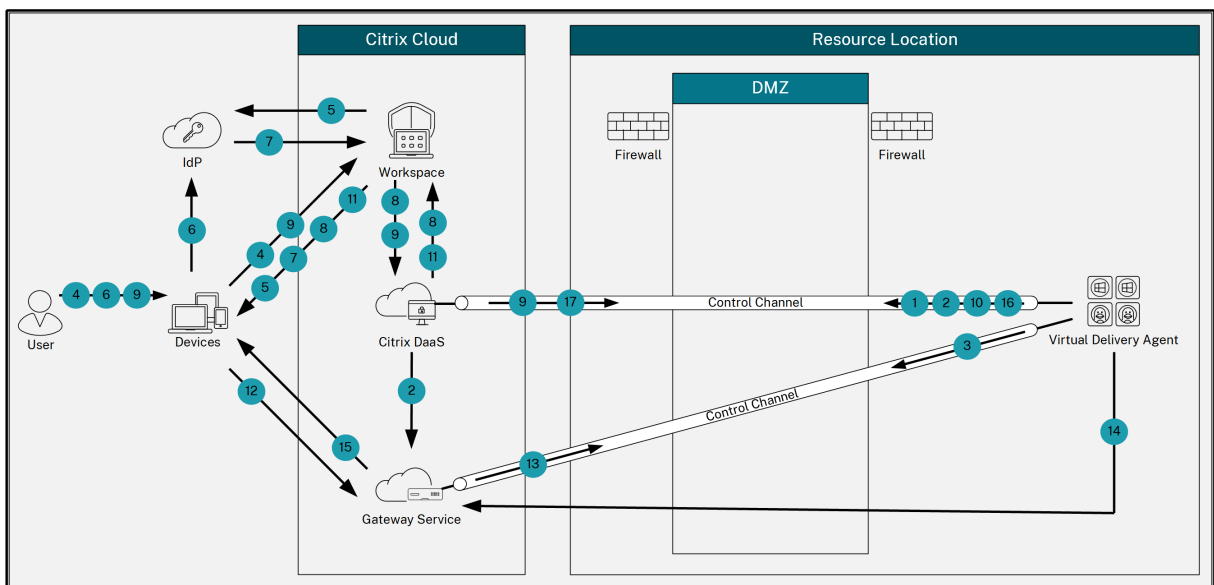
如果正在使用 Rendezvous V2, 协议版本将显示 2.0。

提示:

如果启用了 Rendezvous 时 VDA 无法直接访问 Citrix Gateway 服务, VDA 将回退到通过 Cloud Connector 代理 HDX 会话。

Rendezvous 通信流

下图说明了有关 Rendezvous 通信流的步骤顺序。



1. VDA 与 Citrix Cloud 建立 WebSocket 连接并注册。
2. VDA 在 Citrix Gateway 服务中注册并获得专用令牌。
3. VDA 与 Gateway 服务建立永久控制连接。
4. 用户导航到 Citrix Workspace。
5. Workspace 评估身份验证配置并将用户重定向到相应的 IdP 进行身份验证。
6. 用户输入自己的证书。
7. 成功验证用户凭据后, 用户将被重定向到 Workspace。
8. Workspace 为用户计算资源的数量并显示这些资源。
9. 用户从 Workspace 中选择桌面或应用程序。Workspace 向 Citrix DaaS 发送请求, 后者代理连接并指示 VDA 为会话做好准备。
10. VDA 使用 Rendezvous 功能及其标识进行响应。
11. Citrix DaaS 生成启动票据并通过 Workspace 将其发送到用户设备。
12. 用户的端点连接到 Gateway 服务并提供启动票据以进行身份验证和识别要连接到的资源。
13. Gateway 服务将连接信息发送到 VDA。
14. VDA 为与 Gateway 服务的会话建立直接连接。

15. Gateway 服务完成端点与 VDA 之间的连接。
16. VDA 验证会话的许可。
17. Citrix DaaS 向 VDA 发送适用的策略。

Secure HDX (技术预览版)

August 20, 2024

可以端到端加密 Citrix Workspace 应用程序 (客户端) 与 VDA (会话主机) 之间的 ICA 会话。

端到端加密 (E2EE) 功能不允许包括 Citrix Gateway 在内的任何中间网络元素解密 ICA 流量。它可以提升环境的安全状况, 并且易于配置和管理。

系统要求

- Linux VDA 最低版本 2311
- Delivery Controller 最低版本 2308
- StoreFront 最低版本 2308
- 适用于 Windows 的 Citrix Workspace 应用程序最低版本 2308

配置

启用端到端加密

默认情况下, 端到端加密 (E2EE) 功能处于禁用状态。要将其启用, 请在 Citrix Studio 中将 **Secure HDX** 策略设置为已启用。

安排证书续订

端到端加密 (E2EE) 功能需要由 Linux VDA 上的 **ctxcertmgr** 服务进行管理的自签名证书及其私钥。

ctxcertmgr 服务启动或重新启动时会创建新的自签名证书。默认情况下, **ctxcertmgr** 服务每 7 天在凌晨 2:00 更新一次证书 (包括其私钥)。还可以使用类似于以下内容的注册表设置来安排证书续订:

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -
   t "REG_SZ" -v "CaRotationStartDate" -d "2023-10-19" --force
2
3 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -
   t "REG_SZ" -v "CaRotationTime" -d "00:45:30" --force
4
5 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\SecureHDX" -
   t "REG_DWORD" -v "CaRotationPeriod" -d "0x00000005" --force
```

在上面的示例中，第一次证书续订时间设置为 2023-10-19 的 00:45:30。之后，**ctxcermgr** 服务每 5 天在 00:45:30 续订一次证书。计划的日期和时间是指 Linux VDA 上的日期和时间。

使用 **DTLS** 保护用户会话安全

July 15, 2024

从 7.18 版本起，DTLS 加密是一项完全受支持的功能。默认情况下，此功能在 VDA 上处于启用状态。有关详细信息，请参阅[传输层安全性](#)。

启用 **DTLS** 加密

验证自适应传输是否已启用

在 Citrix Studio 中，验证 **HDX** 自适应传输策略设置为首选还是诊断模式。

在 **Linux VDA** 上启用 **SSL** 加密

在 Linux VDA 上，使用 `/opt/Citrix/VDA/sbin` 上的 **enable_vdassl.sh** 工具启用（或禁用）SSL 加密。有关工具中可用的选项的信息，请运行 `/opt/Citrix/VDA/sbin/enable_vdassl.sh -h` 命令。

注意：

Linux VDA 支持 DTLS 1.0、DTLS 1.2 和 DTLS 1.3，默认情况下使用 DTLS 1.2。检查您的 Citrix Workspace 应用程序中正在使用哪个版本的 DTLS。请务必在 Linux VDA 和您的 Citrix Workspace 应用程序中使用相同版本的 DTLS。如果您的 Citrix Workspace 应用程序仅支持 DTLS 1.0（例如，Citrix Receiver for Windows 4.11），请使用 **enable_vdassl.sh** 工具将 **SSLMinVersion** 设置为 **TLS_1.0**，将 **SSLCipherSuite** 设置为 **COM** 或 **ALL**。

使用 **TLS** 保护用户会话安全

July 15, 2024

自版本 7.16 起，Linux VDA 支持用于保护用户会话安全的 TLS 加密。默认情况下，TLS 加密处于禁用状态。

启用 **TLS** 加密

要启用用于保护用户会话安全的 TLS 加密，请在 Linux VDA 和 Delivery Controller (Controller) 上安装证书并启用 TLS 加密。

在 **Linux VDA** 上安装证书

获取 PEM 格式的服务器证书和 CRT 格式的根证书。服务器证书包含以下部分：

- 证书
- 未加密的私钥
- 中间证书（可选）

服务器证书示例：

启用 TLS 加密

在 **Linux VDA** 上启用 **TLS** 加密 在 Linux VDA 上，使用 `/opt/Citrix/VDA/sbin` 目录中的 `enable_vdassl.sh` 脚本启用（或禁用）TLS 加密。有关脚本中可用的选项的信息，请运行 `/opt/Citrix/VDA/sbin/enable_vdassl.sh -help` 命令。

```
root@lxu1804:~# /opt/Citrix/VDA/sbin/enable_vdassl.sh
==Enable/Disable SSL on Linux VDA==
To enable SSL, a certificate file must be specified, otherwise the local certificate file under
/etc/xdl/.sslkeystore/ is used, If the local certificate file does not exist, the command
fails. You can specify the SSL port number, version and cipher suite, otherwise, their default
values are used!

Usage: enable_vdassl.sh -Disable
       Disable Linux VDA SSL.

Usage: enable_vdassl.sh -Enable [-Certificate <CERT-FILE>] [-SSLPort <SSL-PORT-NUMBER>]
       [-SSLMinVersion <SSL-MIN-VERSION>] [-SSLCipherSuite <SSL-CIPHER-SUITE>]
       Enable Linux VDA SSL.

Options:
-Certificate <CERT-FILE>
  Specify a certificate file, where <CERT-FILE> must include the full file path. Only one format
  is currently supported, that is PEM.

-RootCertificate <ROOT-CERT-FILE>
  Specify a root certificate file, where <ROOT-CERT-FILE> must include the full file path, The root certificate will be put in the local keystore(under /etc/xdl/.sslkeystore/cacerts).

-SSLPort <SSL-PORT-NUMBER>
  Specify an SSL port number. Unless otherwise specified, the default port 443 used.

-SSLMinVersion <TLS_1.0|TLS_1.1|TLS_1.2|TLS_1.3>
  Specify SSL version. Unless otherwise specified, the default value TLS_1.2 is used.

-SSLCipherSuite <GOV|COM|ALL>
  Specify an SSL Cipher suite. Unless otherwise specified, the default value GOV is used.

Examples:
enable_vdassl.sh -Enable -Certificate "/home/cert001.pem"
Enable Linux VDA SSL using Certificate cert001.pem.

enable_vdassl.sh -Enable -RootCertificate "/home/rootCR.cer"
Enable Linux VDA SSL using Root Certificate rootCR.cer with local certificate(under /etc/xdl/.sslkeystore).

enable_vdassl.sh -Enable -SSLPort 445
Enable Linux VDA SSL on port 445 using local certificate(under /etc/xdl/.sslkeystore).

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445
Enable Linux VDA SSL using Certificate cert001.pem on port 445, with default SSLMinVersion and SSLCipherSuite.

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445 -SSLMinVersion "TLS_1.2"
Enable Linux VDA SSL using Certificate cert001.pem on port 445 with SSLMinVersion TLS_1.2 and default SSLCipherSuite..

enable_vdassl.sh -Enable -Certificate "/home/cert001.pem" -SSLPort 445 -SSLMinVersion "TLS_1.2" -SSLCipherSuite "GOV"
Enable Linux VDA SSL using Certificate cert001.pem on port 445 with SSLMinVersion TLS_1.2 and SSLCipherSuite GOV.
```

提示：必须在每台 Linux VDA 服务器上安装服务器证书，以及必须在每台 Linux VDA 服务器和客户机上安装根证书。

在 **Controller** 上启用 **TLS** 加密

注意：

只能对整个交付组启用 TLS 加密。不能为特定应用程序启用 TLS 加密。

在 Controller 上的 PowerShell 窗口中，按顺序运行以下命令以对目标交付组启用 TLS 加密。

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule -HdxSslEnabled $true`

注意：

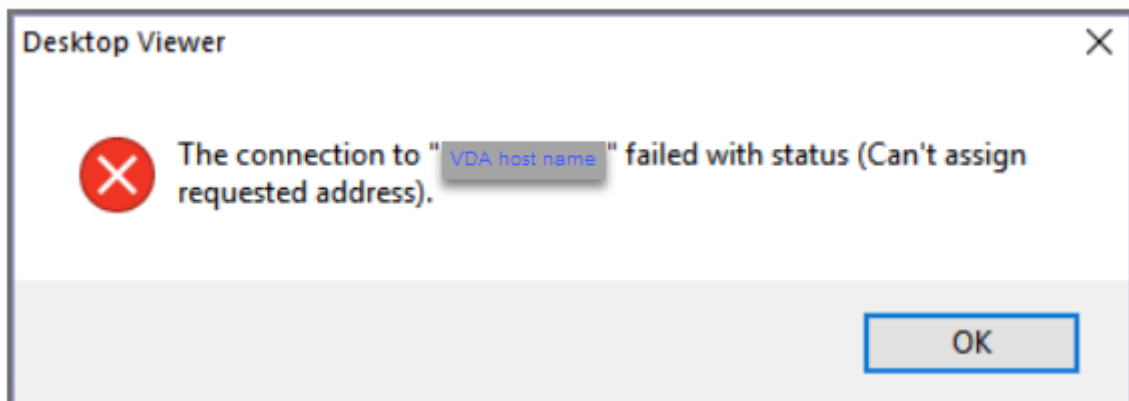
要确保 ICA 会话文件中仅包含 VDA FQDN，还可以运行 `Set-BrokerSite -DnsResolutionEnabled $true` 命令。该命令将启用 DNS 解析。如果禁用了 DNS 解析，ICA 会话文件会显示 VDA IP 地址，并仅为与 TLS 相关的项目（例如 `SSLProxyHost` 和 `UDPDTLSPort`）提供 FQDN。

要在 Controller 上禁用 TLS 加密，请按顺序运行以下命令：

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule -DesktopGroupName 'GROUPNAME' | Set-BrokerAccessPolicyRule -HdxSslEnabled $false`
3. `Set-BrokerSite -DnsResolutionEnabled $false`

故障排除

尝试访问已发布的桌面会话时，适用于 Windows 的 Citrix Workspace 应用程序中可能会出现下面的“无法分配请求的地址”错误：



解决方法：向 **hosts** 文件添加一个条目，类似于：

```
<IP address of the Linux VDA> <FQDN of the Linux VDA>
```

在 Windows 计算机上，**hosts** 文件通常位于 `C:\Windows\System32\drivers\etc\hosts` 中。

会话可靠性

August 20, 2024

Citrix 将会话可靠性功能引入所有支持的 Linux 平台。会话可靠性在默认情况下处于启用状态。

会话可靠性将在网络中断时无缝重新连接 ICA 会话。有关会话可靠性的详细信息，请参阅[客户端自动重新连接和会话可靠性](#)。

注意：

默认情况下，通过会话可靠性连接传输的数据是纯文本数据。出于安全考虑，我们建议您启用 TLS 加密。有关 TLS 加密的详细信息，请参阅[使用 TLS 保护用户会话安全](#)。

配置

Citrix Studio 中的策略设置

可以在 Citrix Studio 中为会话可靠性设置以下策略：

- 会话可靠性连接
- 会话可靠性超时
- 会话可靠性端口号
- 重新连接 UI 透明度级别

有关详细信息，请参阅[会话可靠性策略设置](#)和[客户端自动重新连接策略设置](#)。

注意：

设置会话可靠性连接或会话可靠性端口号策略后，请按此顺序重新启动 VDA 服务和 HDX 服务，以使设置生效。

Linux VDA 上的设置

- 启用/禁用会话可靠性 **TCP** 侦听器

默认情况下，在端口 2598 上启用和侦听会话可靠性 TCP 侦听器。要禁用该侦听器，请运行以下命令。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
   CurrentControlSet\Control\Citrix\WinStations\cgp" -v "
   fEnableWinStation" -d "0x00000000"
```

注意：

请重新启动 HDX 服务以使您的设置生效。禁用 TCP 侦听器不会禁用会话可靠性。仍可通过其他侦听器（例如 SSL）获得会话可靠性，前提是通过会话可靠性连接策略启用了该功能。

- 会话可靠性端口号

还可以使用以下命令设置会话可靠性端口号（以端口号 2599 为例）。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
   CurrentControlSet\Control\Citrix\WinStations\cgp" -v "PortNumber" -d
   "2599"
```

注意：

请重新启动 HDX 服务以使您的设置生效。如果在 **Citrix Studio** 中通过策略设置设置了端口号，则将忽略 Linux VDA 上的设置。确保 VDA 上的防火墙配置为不禁止通过设置端口传输网络流量。

- 服务器到客户端保持活动状态时间间隔

当会话中没有任何活动（例如，没有鼠标移动或屏幕更新）时，将在 Linux VDA 与客户端之间发送保持活动状态消息。保持活动状态消息用于检测客户端是否仍可响应。如果客户端没有响应，则挂起会话，直到客户端重新连接。此设置指

定相邻保持活动状态消息之间间隔的秒数。默认情况下未配置此设置。要对其进行配置，请运行以下命令（以 10 秒为例）。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig" -t "REG_DWORD" -v "CgpServerToClientKeepAlive" -d "10" --force
```

- 客户端到服务器保持活动状态时间间隔

此设置指定从 ICA 客户端发送到 Linux VDA 的相邻保持活动状态消息之间间隔的秒数。默认情况下未配置此设置。要对其进行配置，请运行以下命令（以 10 秒为例）。

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XTEConfig" -t "REG_DWORD" -v "CgpClientToServerKeepAlive" -d "10" --force
```

故障排除

通过策略设置启用会话可靠性后，无法启动会话。

要解决此问题，请执行以下操作：

1. 在 Citrix Studio 中通过策略设置启用会话可靠性后，请务必按此顺序重新启动 VDA 服务和 HDX 服务。
2. 在 VDA 上，运行以下命令以验证会话可靠性 TCP 侦听器是否正在运行（以端口 2598 为例）。

```
1 netstat -an | grep 2598
```

如果会话可靠性端口上没有 TCP 侦听器，请运行以下命令启用该侦听器。

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\WinStations\cgp" -v "fEnableWinStation" -d "0x00000001"
```

会话录制（预览版）

July 15, 2024

可以录制和重播 Linux VDA 上托管的会话。

注意：

此功能在预览版中提供。预览功能可能未完全本地化，建议在非生产环境中使用。Citrix 技术支持部门不对在预览版功能中发现的问题提供支持。

启用或禁用会话录制

要启用或禁用 Linux VDA 的会话录制，请分别将 **SmAudAllowed** 设置为 **1** 或 **0**。可以使用以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
  SmAudAllowed" -d "0x00000001" --force
```

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
  SmAudAllowed" -d "0x00000000" --force
```

注意：

在 Linux VDA 上启用会话录制后，用户登录其会话时会收到关于正在录制其会话的通知。

指定录制文件的文件大小

录制文件大小越大，下载录制文件所需的时间越长，在播放过程中使用搜寻滑块进行导航时文件的响应速度越慢。要控制文件的大小，请为文件指定阈值限制。当录制文件达到此限制时，将关闭当前文件，并创建一个额外的文件以继续录制。此操作称为滚动。

请使用以下命令可以为滚动指定两个阈值：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
  RolloverFileSizeInMB" -d "0x00000032" --force
2
3 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_DWORD" -v "
  RolloverTimeInHours" -d "0x0000000c" --force
```

- **RolloverFileSizeInMB**。达到规定的大小时，当前文件将关闭并打开一个新文件。默认情况下，当大小超过 50 MB 时会发生滚动。支持的值：10–300。
- **RolloverTimeInHours**。达到持续时间后，当前文件将关闭并打开一个新文件。默认情况下，会话录制了 12 小时时会滚动更新。支持的值：1–24。

满足上述两个条件中的第一个条件时会滚动更新。例如，您指定文件大小为 17 MB，指定持续时间为 6 小时。当您的录制文件在 3 小时内达到 17 MB 时，Session Recording 会关闭该文件并打开一个新文件。

为防止创建许多小文件，滚动会在至少一小时之后才进行，而不考虑为文件大小指定的值。文件大小超过 300 MB 属此规则的例外情况。

指定录制文件的存储位置

默认情况下，录制文件存储在 `/var/xdl/session_recordings` 下。要指定不同的路径，请运行以下命令：

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
  CurrentControlSet\Control\Citrix\SmartAuditor" -t "REG_SZ" -v "Path"
  -d "<your custom storage path>" --force
```

可以将录制文件存储在本地驱动器或指向网络路径的装载点上。配置对您设置的存储路径的正确访问权限，并授予用户 `ctxsrvr` 对该路径的写入权限。

查看录制

要查看录制文件，请完成以下步骤以安装 Session Recording Player 或 Session Recording Web Player:

1. 使用您的 Citrix 帐户凭据访问 [Citrix Virtual Apps and Desktops 下载页面](#) 并下载产品文件。解压文件。
2. 双击 `SessionRecordingPlayer.msi` 和 `SessionRecordingWebPlayer.msi` 并按照说明完成安装。

提示:

要使用 Session Recording Web Player，请仅将其安装在 Session Recording Server 上，并确保录制文件在 Session Recording Server 上可用。有关详细信息，请参阅 [Citrix Session Recording 文档](#)。

限制

- 对于虚拟应用程序会话，录制通知可能不居中。

虚拟通道 SDK (预览版)

July 15, 2024

借助适用于 Linux VDA 的虚拟通道软件开发工具包 (SDK)，您可以编写要在 VDA 上运行的服务器端应用程序。有关详细信息，请参阅 [Citrix Virtual Channel SDK for the Linux VDA](#) (适用于 Linux VDA 的 Citrix 虚拟通道 SDK) 文档。

适用于 Linux VDA 的 Citrix 虚拟通道 SDK 可从 [Citrix Virtual Apps and Desktops 下载页面](#) 进行下载。展开相应版本的 VDA Citrix Virtual Apps and Desktops 并单击组件以选择要下载的 Linux VDA。

注意:

此功能在预览版中提供。预览功能可能未完全本地化，建议在非生产环境中使用。Citrix 技术支持不对在预览功能中发现的问题提供支持。

Wayland (预览版)

August 20, 2024

Linux VDA 在 RHEL 9.4/9.3/9.2、Rocky Linux 9.4/9.3/9.2 和 Ubuntu 22.04 上支持 GNOME 中的 Wayland。

以下功能已在 Wayland 中经过全面测试：

- 音频
- 剪贴板
- 客户端驱动器映射 (CDM)
- 打印
- USB 设备重定向

注意：

- 此功能在预览版中提供。预览功能可能未完全本地化，建议在非生产环境中使用。Citrix 技术支持部门不对在预览版功能中发现的问题提供支持。
- 不支持 HDX 3D Pro。
- 不支持 Linux 虚拟应用程序会话。

启用 Wayland

要使用 Wayland，请运行以下命令将注册表项 **EnableWayland** 设置为 **1**：

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet\Control\Citrix\Wayland" -t "REG_DWORD" -v "EnableWayland" -d "0x00000001" --force
```

默认情况下，注册表项 **EnableWayland** 设置为 **0**，这意味着使用 X11。

检查 Wayland 是否在使用中

1. 在 Linux 中打开终端窗口。
2. 运行 **echo \$XDG_SESSION_TYPE** 命令。

如果使用 Wayland，则输出中会出现 **'wayland'**。

限制

使用 Wayland 时会发现以下限制：

- 客户端设备的键盘布局与 VDA 的键盘布局不同步。

最佳做法

July 15, 2024

此部分将指导您完成以下过程：

- 使用 [Machine Creation Services \(MCS\)](#) 在 Google 云端平台 (GCP) 上创建 Linux VDA

使用 **Machine Creation Services (MCS)** 在 **Google** 云端平台 (**GCP**) 上创建 **Linux VDA**

August 20, 2024

要使用 MCS 在 GCP 上创建 Linux VDA，请完成以下步骤：

步骤 1: 在 [GCP](#) 上创建 Linux 虚拟机 (VM)

步骤 2: 创建 [GCP 服务帐户](#)

步骤 3: 在 [Citrix Studio](#) 中创建与 GCP 的主机连接

步骤 4: 准备 Linux VDA 主映像

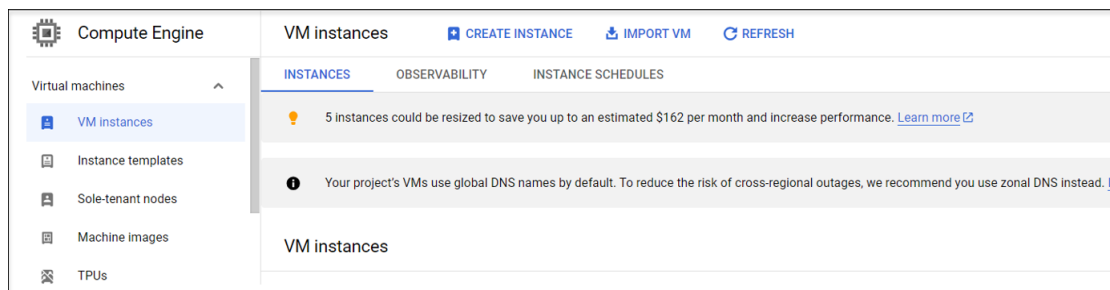
步骤 5: 创建计算机目录

步骤 6: 创建交付组

步骤 1: 在 **GCP** 上创建 **Linux VM**

步骤 1a: 在 Google Cloud 控制台中，导航到 **Compute Engine > VM instances** (VM 实例)。

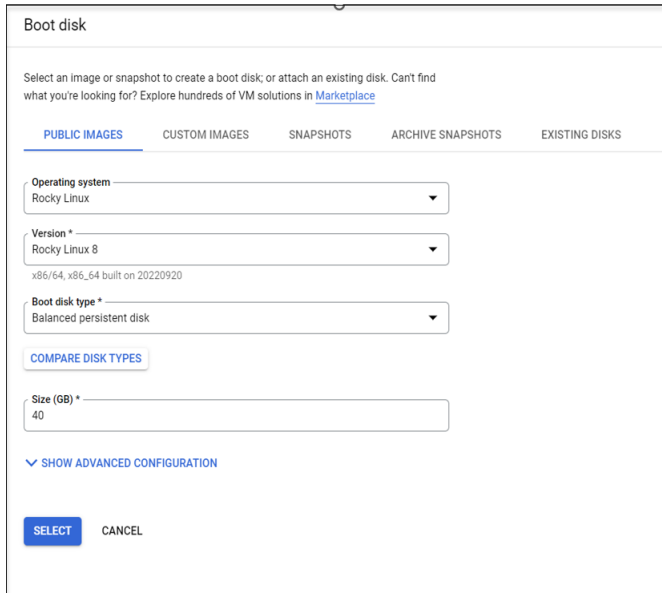
步骤 1b: 在“VM instances” (VM 实例) 页面上，单击 **CREATE INSTANCE** (创建实例) 以创建 VM 实例。



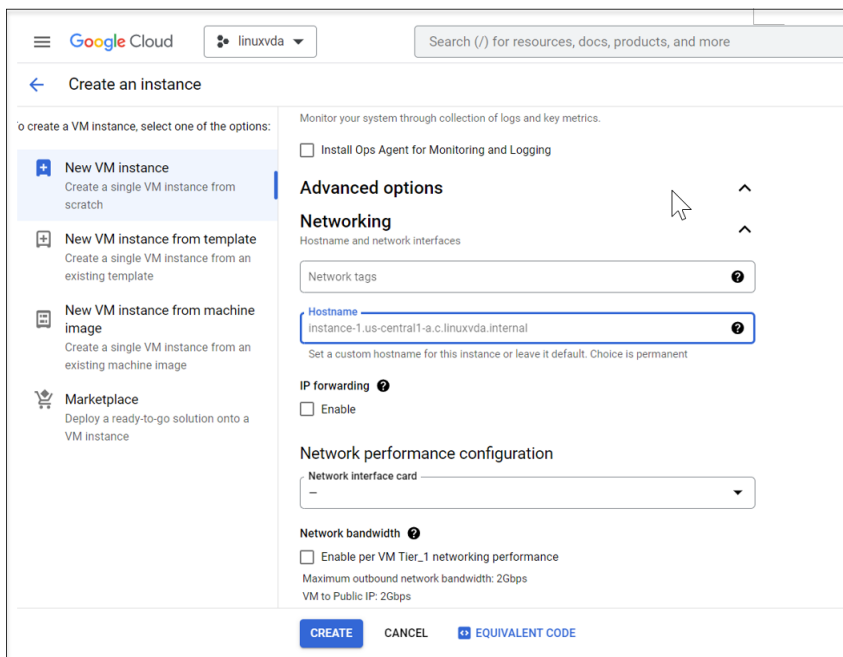
步骤 1c: 进行以下配置并将其他配置保留为默认配置：

- 输入您的 VM 实例的名称。

- 选择一个区域和资源域来托管您的 VM。
- (可选) 向您的 VM 中添加 GPU。有关详细信息，请参阅本文中的步骤 **4c**。
- 在 **Boot disk** (引导磁盘) 部分中，选择您的 VM 的操作系统和磁盘大小。例如：

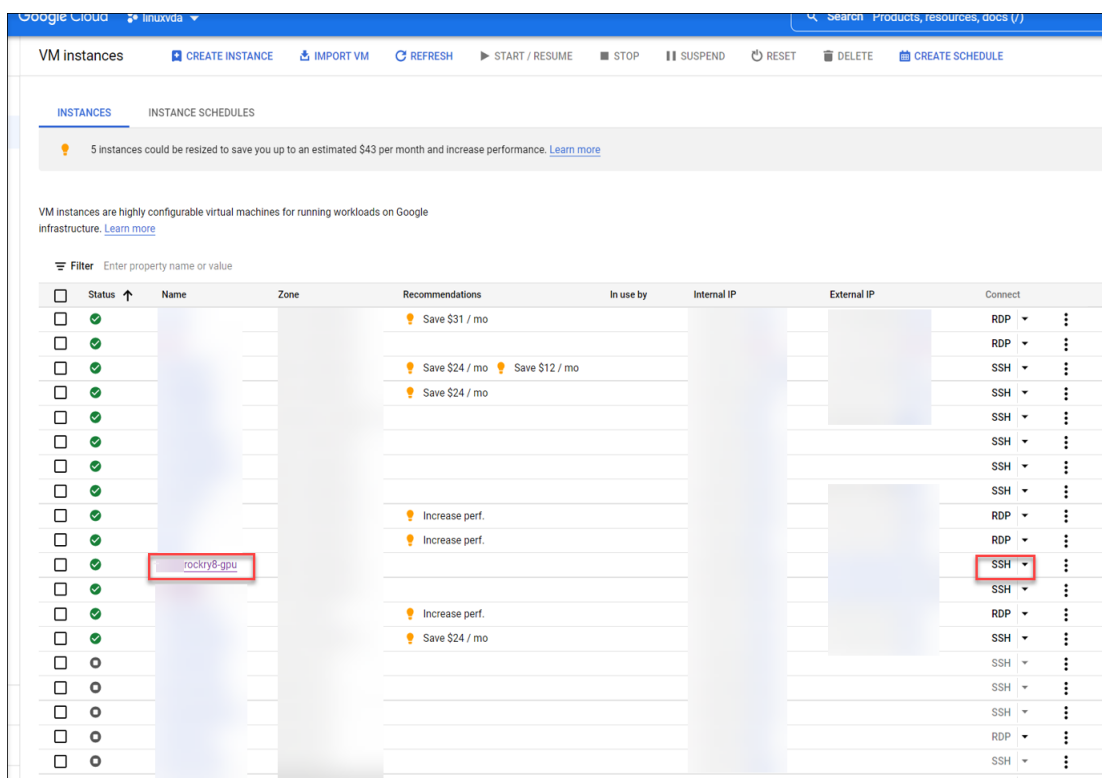


- 导航到高级选项 > 网络连接，然后将主机名字段设置为 FQDN。

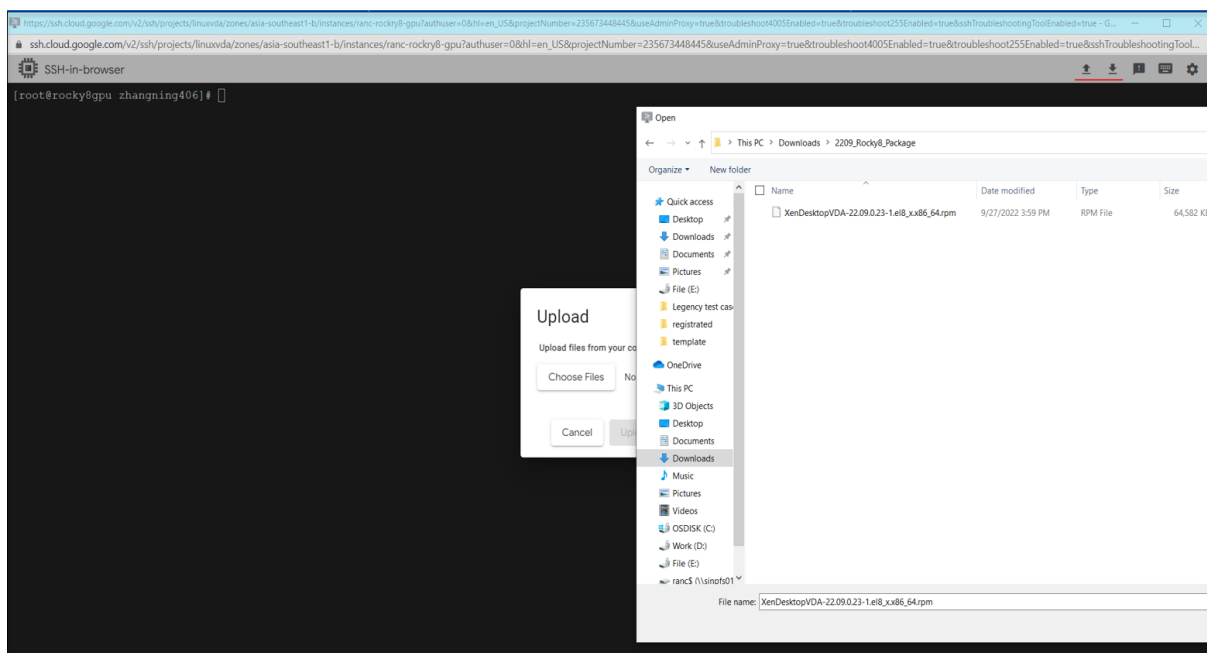


步骤 **1d**: 单击创建创建 VM 实例。

步骤 **1e**: 创建 VM 后，返回到 **Compute Engine** 控制板，在列表中找到您的 VM 实例，然后单击“SSH”按钮连接到您的 VM。



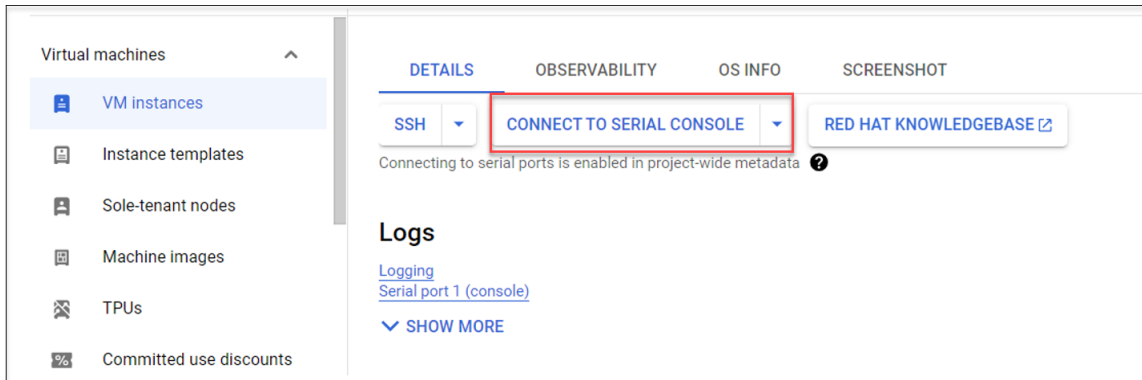
步骤 1f: 通过基于 Web 的 SSH 客户端将 Linux VDA 软件包上传到您的 VM。



步骤 1g: 防止无法使用 SSH 访问 VM。

重新启动后可能无法访问 VM。要解决此问题，请在首次登录 VM 时设置 root 用户密码，并确保您能够以 root 用户身份登录 VM。然后，在重新启动 VM 后，在控制台中运行以下命令：

```
1 nmcli dev connect eth0
```

2 `systemctl restart NetworkManager`

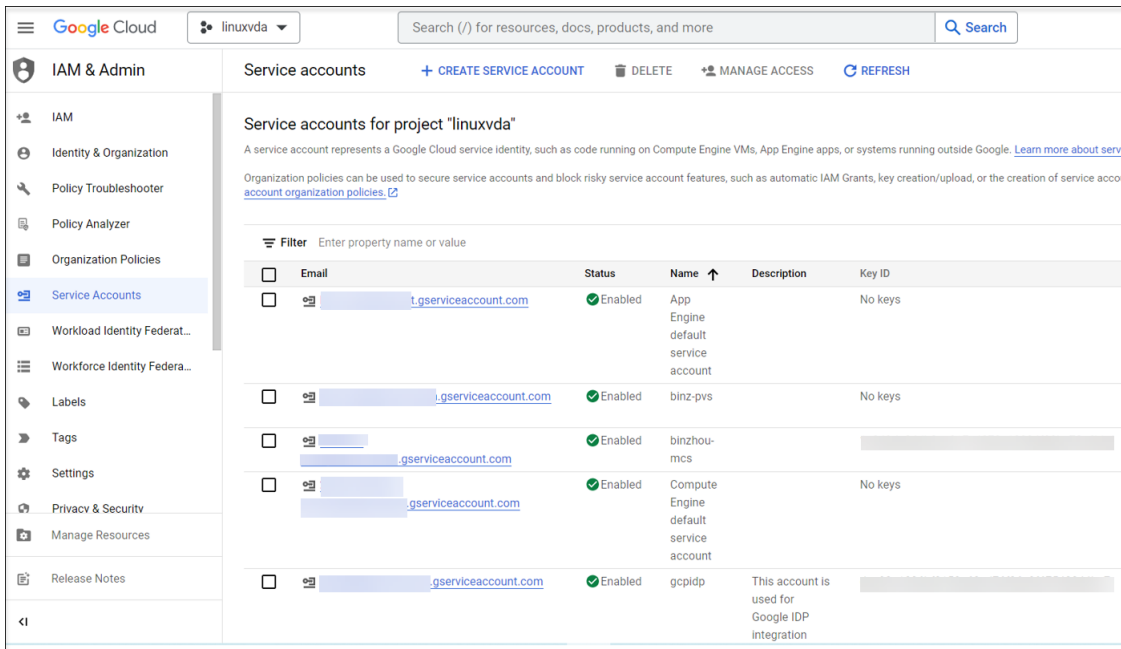
步骤 2：创建 GCP 服务帐户

本部分内容将指导您完成创建 GCP 服务帐户的过程，包括创建服务帐户密钥和向服务帐户授予所需的角色。

注意：

要创建 GCP 服务帐户，请确保您被授予了 [服务帐户管理员](#) (roles/iam.serviceAccountAdmin) IAM 角色。

步骤 2a：在 Google Cloud 控制台中，导航到 **IAM & admin (IAM 和管理员) > Service Accounts (服务帐户)**，然后单击 **Create Service Account (创建服务帐户)** 选项卡。



步骤 2b：在 **Service account details (服务帐户详细信息)** 步骤中，在字段中输入值，然后单击 **Create and continue (创建并继续)**。

Google Cloud | linuxvda | Search (/) for resources, docs, products, and more

IAM & Admin | Create service account

1 Service account details

Service account name
Display name for this service account

Service account ID *
x C

Email address: <id>@linuxvda.iam.gserviceaccount.com

Service account description
Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to project (optional)

3 Grant users access to this service account (optional)

DONE CANCEL

步骤 2c: 跳过可选步骤，然后单击底部的完成。

IAM & Admin | Create service account

1 Service account details

2 Grant this service account access to project (optional)

Grant this service account access to linuxvda so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Select a role

IAM condition (optional) ?
+ ADD IAM CONDITION

+ ADD ANOTHER ROLE

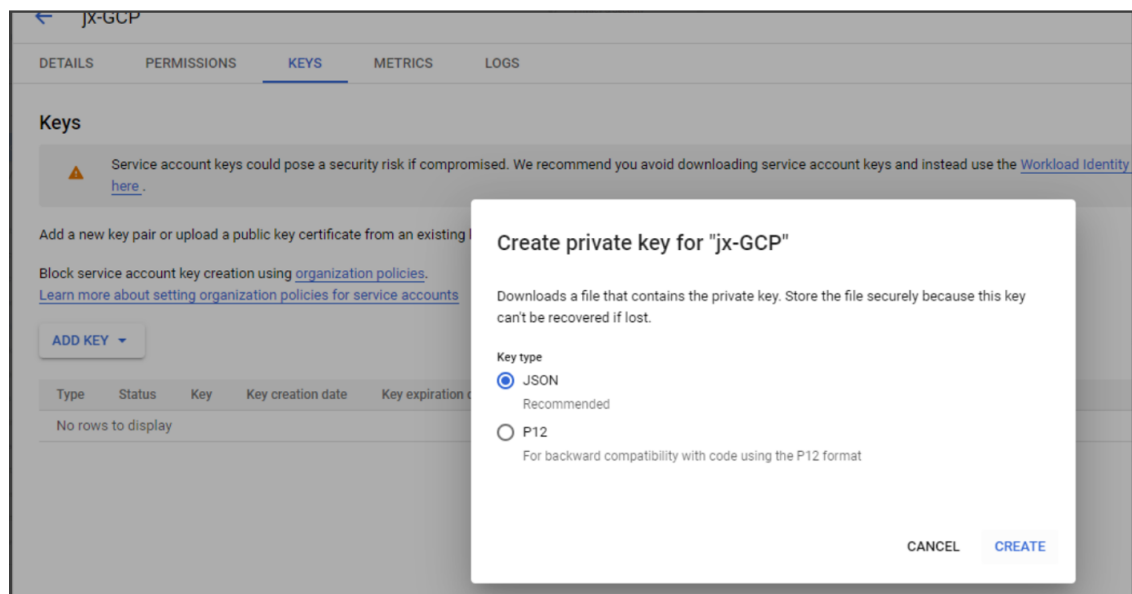
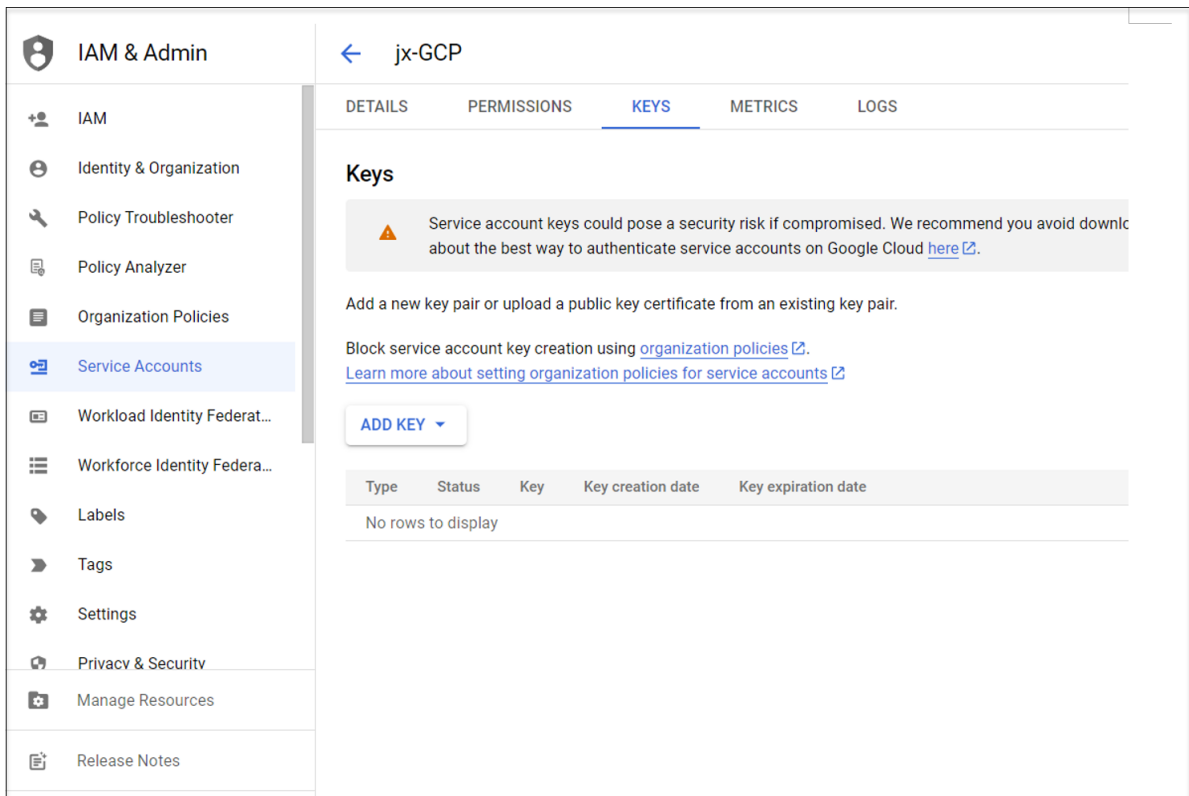
CONTINUE

3 Grant users access to this service account (optional)

DONE CANCEL

步骤 2d: 再次导航到 **IAM & admin (IAM 和管理员) > Service Accounts (服务帐户)**，然后单击 **Service accounts (服务帐户)** 选项卡。找到新创建的服务帐户，导航到 **Keys (密钥)** 选项卡，然后单击 **Add key (添加密钥)**

钥) > **Create new key** (创建新密钥) > **JSON** > **Create** (创建)。



注意：

下载密钥文件后，无法再次下载。

步骤 2e: 在 Google Cloud 控制台中，导航到 **IAM & Admin** (IAM 和管理员) > **IAM**，然后单击 **Add** (添加)。在 **New members** (新成员) 字段中搜索并选择新创建的服务帐户，然后为该服务帐户选择一个角色以授予其访问您的

资源的权限。通过单击 **Add another role**（添加其他角色）继续授予角色，以确保将以下所有角色授予新创建的服务帐户。

- 计算管理员
- 存储管理员
- **Cloud Build** 编辑者
- 服务帐户用户
- 云数据存储用户
- 计算实例管理员 (**Beta**)
- 所有者

例如：



Cloud Build Editor	4/7 excess permissions
Cloud Datastore User	15/16 excess permissions
Compute Admin	524/576 excess permissions
Compute Instance Admin (beta)	121/162 excess permissions
Owner	4804/4871 excess permissions
Service Account User	3/5 excess permissions
Storage Admin	13/24 excess permissions

步骤 3：在 **Citrix Studio** 中创建与 **GCP** 的主机连接

根据 [Google 云端平台虚拟化环境](#) 设置您的 GCP 环境，然后完成以下步骤以创建与 GCP 的主机连接。

1. 对于本地 Delivery Controller，请在本地 Citrix Studio 中选择配置 > 托管 > 添加连接和资源以创建主机连接。对于云 Delivery Controller，请在 Citrix Cloud 上的基于 Web 的 Studio 控制台中选择管理 > 托管 > 添加连接和资源以创建主机连接。
2. 在添加连接和资源向导中，选择 **Google** 云端平台作为连接类型。

例如，在 Citrix Cloud 上的基于 Web 的 Studio 控制台中：

Add Connection and Resources

1 Connection
2 Region
3 Network
4 Scopes
5 Summary

Connection

Use an existing connection
BingTest

Create a new connection

Zone name:
[Redacted]

Connection type:
Google Cloud Platform

Service account key:
Import key...

Service account ID:
[Redacted]

Connection name:
[Redacted]

Create virtual machines using:
 Citrix provisioning tools (Machine Creation Services or Citrix Provisioning)
 Other tools

Next Cancel 7

3. 导入 GCP 帐户的服务帐号密钥并键入连接名称。

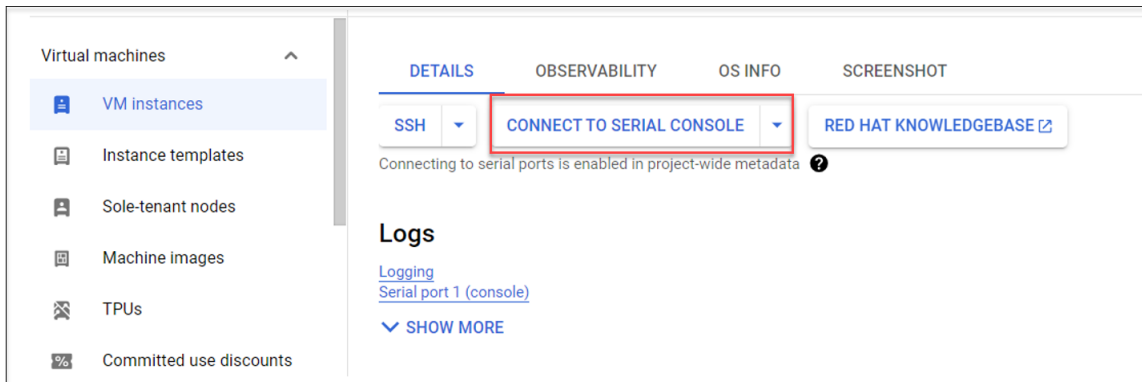
4. 向导将引导您完成各个页面。特定页面内容取决于所选连接类型。完成每一页之后，请选择下一步，直到到达摘要页为止。有关详细信息，请参阅[使用 MCS 创建未加入域的 Linux VDA](#)一文中的[步骤 2：创建主机连接](#)。

步骤 4：准备 **Linux VDA** 主映像

步骤 **4a**：（适用于 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x）配置以太网连接。

在 GCP 上托管的 RHEL 8.x/9.x 和 Rocky Linux 8.x/9.x 上安装 Linux VDA 后，以太网连接可能会断开，VM 重新启动后可能无法访问 Linux VDA。要解决此问题，请在首次登录 VM 时设置 root 用户密码，并确保您能够以 root 用户身份登录 VM。然后，在重新启动 VM 后，在控制台中运行以下命令：

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



步骤 **4b**: 在模板 VM 上安装 Linux VDA 软件包。

在模板 VM 上，执行以下操作来安装 Linux VDA 软件包：

1. 安装 .NET。

除 .NET Runtime 之外，在安装或升级 Linux VDA 之前，还必须在所有受支持的 Linux 发行版中安装 .NET Core Runtime。Amazon Linux 2 需要版本 6。其他发行版需要版本 8。

如果您的 Linux 发行版包含所需的 .NET 版本，请从内置源中进行安装。否则，请从 Microsoft 软件包源中安装 .NET。有关详细信息，请参阅 <https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers>。

2. 安装 Linux VDA 软件包：

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

3. 启用 EPEL 存储库：

```
1 sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

步骤 **4c**: 向 GCP 上的模板 VM 中添加图形处理器 (GPU) (可选)。

1. 在 Google Cloud 控制台中，向模板 VM 中添加一个或多个 GPU。有关在 GCP 上添加和删除 GPU 的信息，请参阅 <https://cloud.google.com/compute/docs/gpus/add-remove-gpus>。

2. 在模板 VM 上安装相应的 GPU 驱动程序。有关详细信息，请参阅 <https://cloud.google.com/compute/docs/gpus/install-drivers-gpu>。

所需的 **NVIDIA** 驱动程序版本：

在 Compute Engine 上运行的 NVIDIA GPU 必须使用以下 NVIDIA 驱动程序版本：

- 对于 L4 GPU：
 - Linux: 525.60.13 或更高版本
- 对于 A100 GPU：
 - Linux: 450.80.02 或更高版本
- 对于 T4、P4、P100 和 V100 GPU：
 - Linux: 410.79 或更高版本
- 对于 K80 GPU (**生命周期已结束**)：
 - Linux: 410.79 - 最新的 R470 版本

对于 K80 GPU，NVIDIA 宣布 **R470 驱动程序** 分支是获得调试支持的最终驱动程序版本。要查看此更新，请参阅 [NVIDIA Software Support Matrix](#) (NVIDIA 软件支持列表)。

安装脚本：

可以使用以下脚本来自动执行安装过程：

```
1 https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

支持的操作系统：

安装脚本已在以下 Linux 发行版上进行了测试：

- Debian 11
- Red Hat Enterprise Linux (RHEL) 8
- Rocky Linux 8
- Ubuntu 20/22

如果您在其他 Linux 发行版上使用此脚本，安装将失败。对于 Linux VM，此脚本仅安装 NVIDIA 驱动程序。

a) 下载安装脚本。

```
1 curl https://raw.githubusercontent.com/GoogleCloudPlatform/compute-gpu-installation/main/linux/install_gpu_driver.py --output install_gpu_driver.py
```

b) 授予对脚本的完全访问权限。

```
1 chmod 777 install_gpu_driver.py
```

c) 运行安装脚本。

```
1 python3 install_gpu_driver.py
```

d) 在 gdm3 中禁用 Wayland。

- 在以下位置之一找到您的发行版的 Wayland 配置文件：
 - /etc/gdm3/custom.conf (Ubuntu)
 - /etc/gdm/custom.conf (CentOS、RHEL、Rocky Linux)
- 使用 `sudo/root` 权限打开文件。
- 通过删除该行开头的 `#` 取消注释 **WaylandEnable=false**。
- 重新启动 VM。

e) 如果您安装了 NVIDIA 510 或更高版本的驱动程序，请禁用 GSP 固件。

如果启用了 GSP 固件，请通过将 NVIDIA 模块参数 **NVreg_EnableGpuFirmware** 设置为 0 来将其禁用。

请通过在 `/etc/modprobe.d/nvidia.conf` 文件中添加以下条目来设置此参数：

- `options nvidia NVreg_EnableGpuFirmware=0`
- 如果 `/etc/modprobe.d/nvidia.conf` 文件不存在，则请创建。

完成此步骤时，请记住以下几点：

- 使用 `sudo` 运行命令来创建和更新配置文件。
- 要重新启动 VM，您可以在 Linux 终端中使用 **sudo reboot**，也可以停止并启动 VM。

步骤 **4d**：配置 MCS 变量。

通过编辑 `/etc/xdl/mcs/mcs.conf` 文件来配置 MCS 变量。下面是您可以为未加入域和已加入域的场景配置的 MCS 变量：

- 适用于未加入域的场景

可以使用默认变量值或者根据需要自定义变量（可选）：

```
DOTNET_RUNTIME_PATH=**path-to-install-dotnet-runtime \**
DESKTOP_ENVIRONMENT=**gnome | mate \**
REGISTER_SERVICE=Y | N
ADD_FIREWALL_RULES=Y | N
VDI_MODE=Y | N
START_SERVICE=Y | N
```

```
#####Linux VDA Configuration#####
# Provide Linux VDA configuration information.
# Please refer to Linux VDA Documentation for these settings.
DOTNET_RUNTIME_PATH=/usr/bin
DESKTOP_ENVIRONMENT=gnome
SUPPORT_DDC_AS_CNAME=N
VDA_PORT=80
REGISTER_SERVICE=Y
ADD_FIREWALL_RULES=Y
HDX_3D_PRO=N
VDI_MODE=N
SITE_NAME='<none>'
LDAP_LIST='<none>'
SEARCH_BASE='<none>'
START_SERVICE=Y
```

- 适用于已加入域的场景

- `Use_AD_Configuration_Files_Of_Current_VDA`: 确定是否使用当前正在运行的 VDA 的现有 AD 相关配置文件 (`/etc/krb5.conf`、`/etc/sss.conf` 和 `/etc/samba/smb.conf`)。如果设置为 Y，MCS 创建的计算机上的配置文件将与当前正在运行的 VDA 上的等效文件相同。但是，您仍然必须配置 `dns` 和 `AD_INTEGRATION` 变量。默认值为 N，这意味着主映像上的配置模板负责确定 MCS 创建的计算机上的配置文件。要使用当前正在运行的 VDA 作为模板 VM，请将该值设置为 Y。否则，请将其设置为 N。
- `dns`: 设置每个 DNS 服务器的 IP 地址。您最多可以设置四个 DNS 服务器。
- `NTP_SERVER`: 设置 NTP 服务器的 IP 地址。除非另行说明，否则该地址是您的域控制器的 IP 地址。
- `WORKGROUP`: 将工作组名称设置为您在 AD 中配置的 NetBIOS 名称(区分大小写)。否则，MCS 将使用域名中紧随计算机主机名之后的部分作为工作组名称。例如，如果计算机帐户为 **user1.lvda.citrix.com**，MCS 将使用 **lvda** 作为工作组名称，而 **citrix** 是正确的选择。请务必正确设置工作组名称。
- `AD_INTEGRATION`: 设置 Winbind、SSSD、PBIS 或 Centrify。有关 MSC 支持的 Linux 发行版和域加入方法的列表，请参阅[支持的发行版](#)。
- `CENTRIFY_DOWNLOAD_PATH`: 设置下载 Server Suite Free (以前称为 Centrify Express) 软件包的路径。只有将 `AD_INTEGRATION` 变量设置为 Centrify 时，该值才会生效。

- `CENTRIFY_SAMBA_DOWNLOAD_PATH`: 设置下载 Centrify Samba 软件包的路径。只有将 `AD_INTEGRATION` 变量设置为 Centrify 时, 该值才会生效。
- `PBIS_DOWNLOAD_PATH`: 设置下载 PBIS 软件包的路径。只有将 `AD_INTEGRATION` 变量设置为 PBIS 时, 该值才会生效。
- `UPDATE_MACHINE_PW`: 启用或禁用自动更新计算机帐户密码。有关详细信息, 请参阅[自动更新计算机帐户密码](#)。
- Linux VDA 配置变量:

```
DOTNET_RUNTIME_PATH=**path-to-install-dotnet-runtime \**
DESKTOP_ENVIRONMENT= **gnome | mate \**
SUPPORT_DDC_AS_CNAME=Y | N
VDA_PORT=port-number
REGISTER_SERVICE=Y | N
ADD_FIREWALL_RULES=Y | N
HDX_3D_PRO=Y | N
VDI_MODE=Y | N
SITE_NAME=dns-site-name | <none> '
LDAP_LIST= 'list-ldap-servers' | <none> ' '
SEARCH_BASE=search-base-set | <none> ' '
FAS_LIST= 'list-fas-servers' | <none> ' '
START_SERVICE=Y | N
TELEMETRY_SOCKET_PORT=port-number
TELEMETRY_PORT=port-number
```

步骤 4e: 创建主映像

1. (仅适用于 SSSD + RHEL 8.x/9.x 或 Rocky Linux 8.x/9.x) 运行 `update-crypto-policies --set DEFAULT:AD-SUPPORT` 命令, 然后重新启动模板 VM。
2. 运行 `/opt/Citrix/VDA/sbin/deploymcs.sh`。

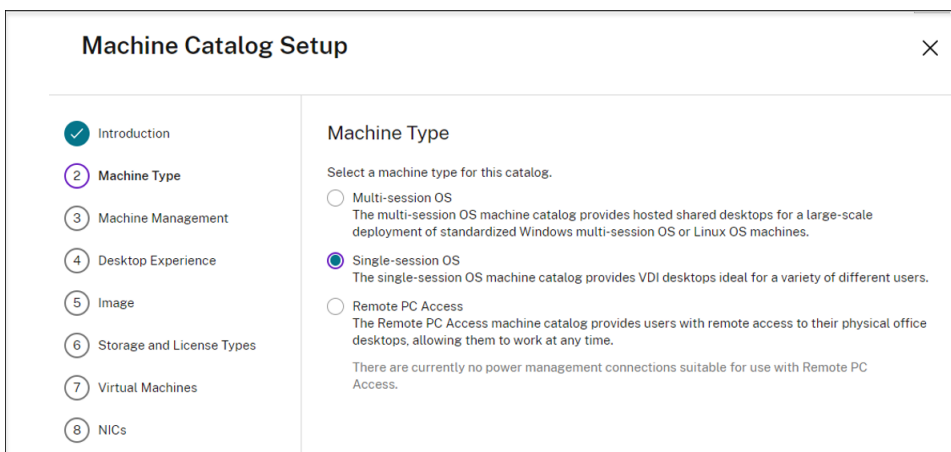
对于未加入域的场景, 以下错误是正常的, 不会阻止您继续操作。

```
[root@localhost ~]#
[root@localhost ~]# /opt/Citrix/VDA/sbin/deploymcs.sh
Installing Linux VDA dependency packages ...
Installing package redhat-lsb-core
Installing package nautilus
Installing package totem-nautilus
Installing package brasero-nautilus
Installing package pulseaudio
Installing package pulseaudio-module-x11
Installing package pulseaudio-gdm-hooks
Installing package pulseaudio-module-bluetooth
Installing package alsa-plugins-pulseaudio
Installing package pciutils
Installing package openssh
Installing package openssh-clients
Installing package java-11-openjdk
Installing package chrony
Installing package krb5-workstation
Installing package oddjob-mkhomedir
Starting PostgreSQL database ...
Installing package tdb-tools
Installing package ntfs-3g
/opt/Citrix/VDA/lib64/mcs ~
installing mcs systemd unit file: ad_join.service ...
~
/opt/Citrix/VDA/lib64/mcs ~
~
Installing winbind dependency packages ...
Installing package samba-winbind
Installing package samba-winbind-clients
ERROR: Exit funtion conf_dns, dns not configured, please do it manually.
[root@localhost ~]#
```

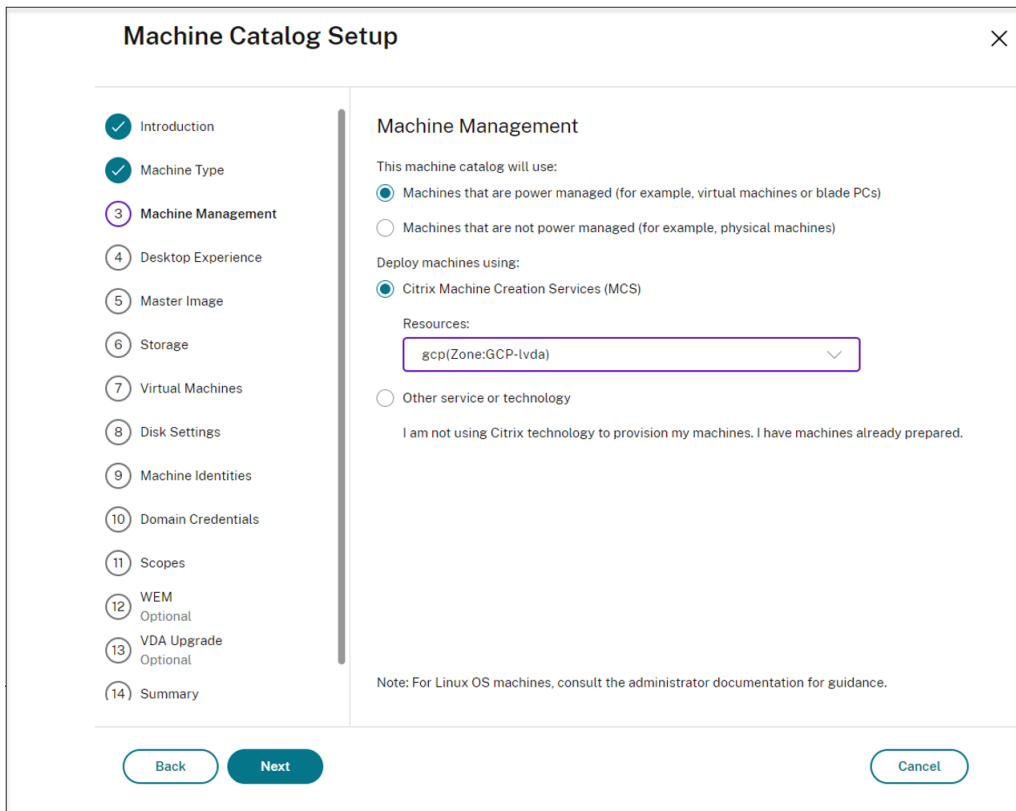
3. 在模板 VM 上安装应用程序并关闭模板 VM。创建并命名主映像的快照。

步骤 5: 创建计算机目录

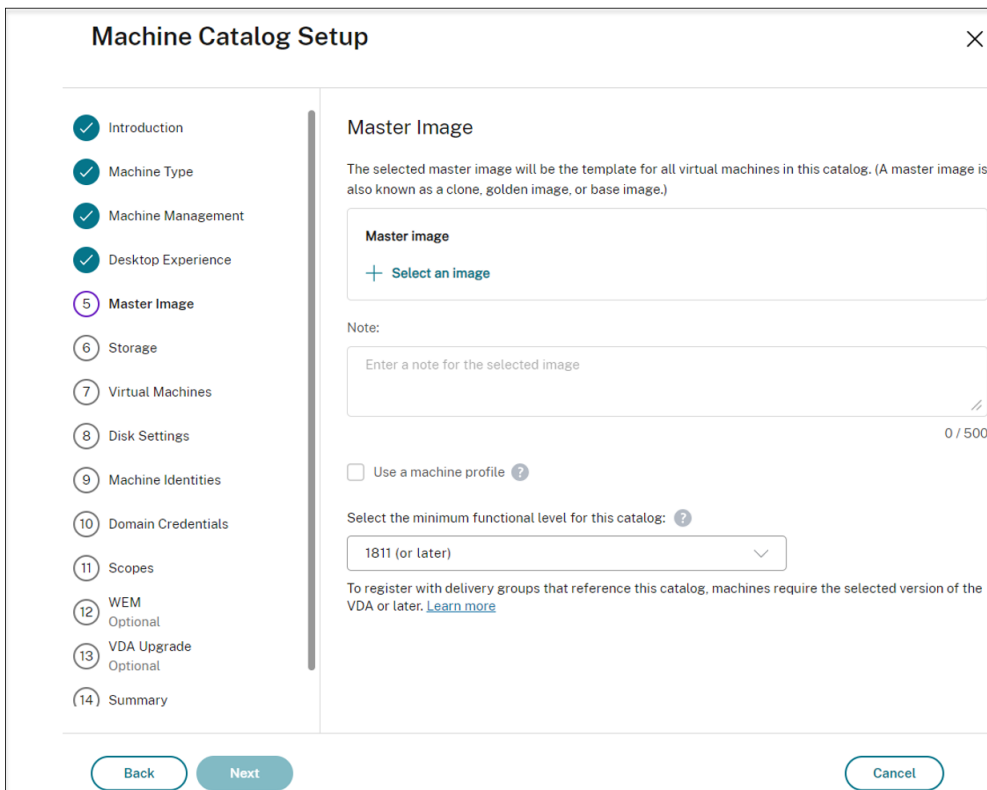
1. 登录到 [Citrix Cloud](#)。在左上角的菜单中，选择我的服务 > **DaaS**。
2. 在管理 > 完整配置中，选择左窗格中的计算机目录。
3. 然后在操作栏中选择创建计算机目录。
4. 在计算机类型页面上，选择多会话操作系统或单会话操作系统，然后选择下一步。

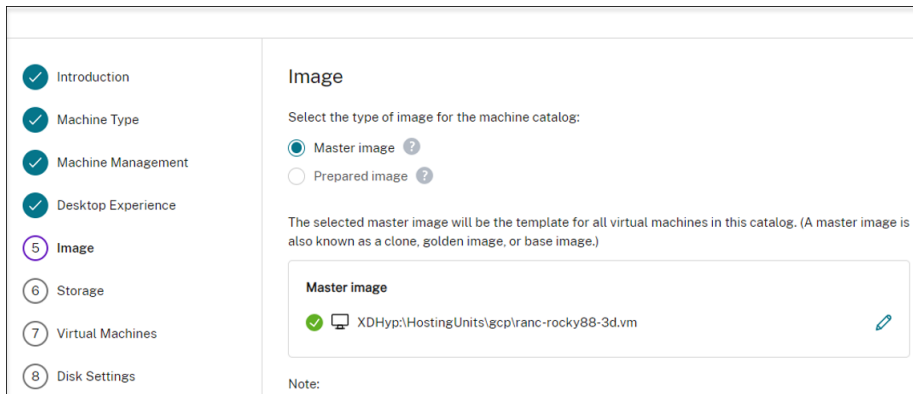


5. 在计算机管理页面上，选择进行电源管理的计算机和 **Citrix Machine Creation Services** 选项，然后选择下一步。如果有多种资源，请从菜单中选择一种资源。



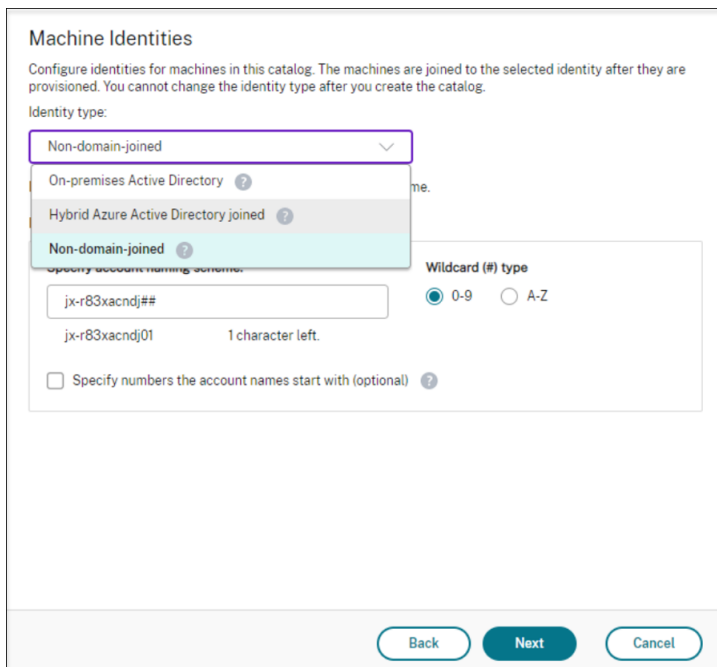
6. 在主映像页面上，选择之前创建的主映像。





7. 在计算机标识页面上，如果主映像未加入任何域，请选择未加入域；如果您将主映像加入某个域，请选择 Active Directory 帐户。

对于未加入域的场景：



对于已加入域的场景：

accounts or use existing accounts.

Select an Active Directory account option:

Create new Active Directory accounts

Use existing Active Directory accounts

Use an existing identity pool to create new accounts ?

Location (domain) for those accounts:

gcp.local

Default OU

- > Computers
- > Domain Controllers
- > ForeignSecurityPrincipals
- > Managed Service Accounts
- > PBIS
- > Program Data
- > System
- > Users

Selected location:

Default OU

Machine account naming scheme ?

Specify account naming scheme:

Wildcard (#) type

0-9 A-Z

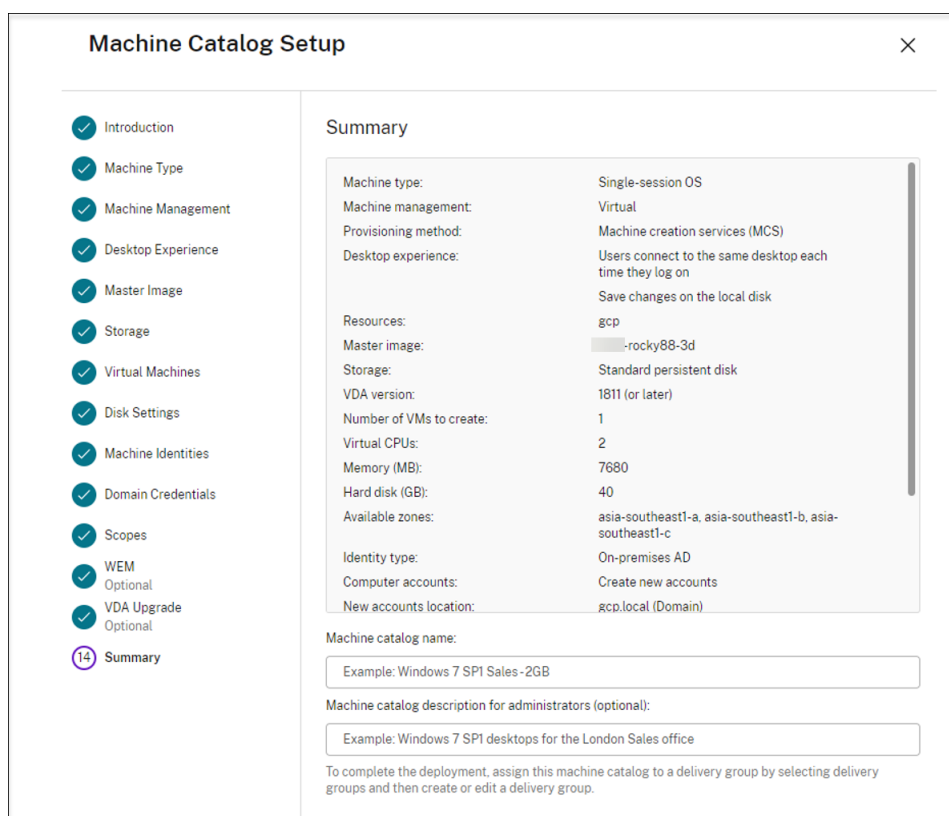
Specify numbers the account names start with (optional) ?

Back Next Cancel

- 如果选择创建新的 **Active Directory** 帐户，请选择一个域，然后输入表示在 Active Directory 中创建的已预配的 VM 计算机帐户的命名方案的字符序列。帐户命名方案可以包含 1-64 个字符，不能包含空格，也不能包含非 ASCII 字符或特殊字符。
- 如果选择使用现有的 **Active Directory** 帐户，请选择浏览以导航到所选计算机的现有 Active Directory 计算机帐户。
- 在域凭据页面上，选择输入凭据，键入用户名和密码，选择保存，然后选择下一步。键入的凭据必须具有执行 Active Directory 帐户操作的权限。

8. 在其他页面上配置其他设置。有关详细信息，请参阅[创建 Google 云端平台目录](#)。

9. 在摘要页面上，确认信息，指定目录的名称，然后选择完成。



完成计算机目录创建可能需要很长时间。完成后，系统将列出目录。可以验证计算机是否在 Google Cloud 控制台中的目标节点组上创建。

步骤 6：创建交付组

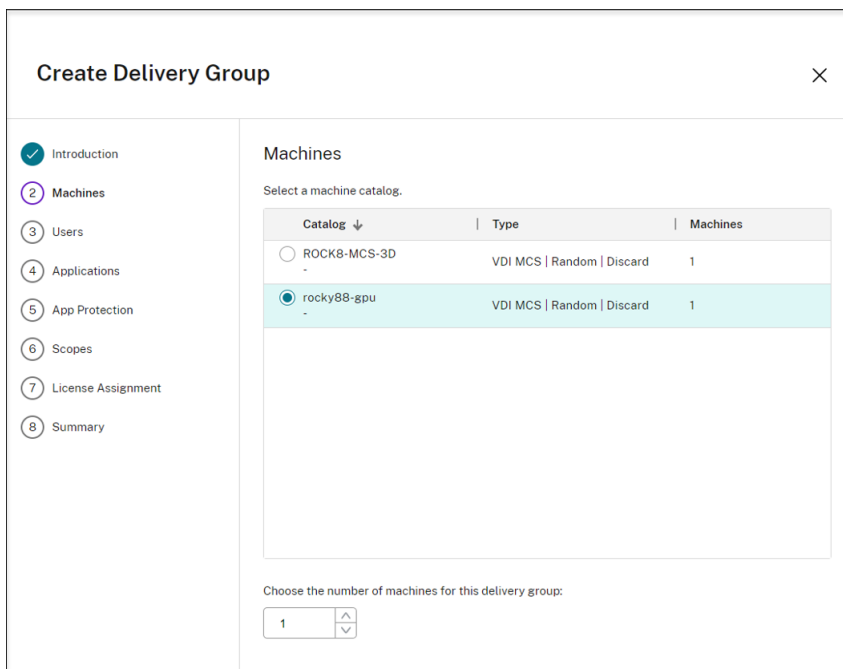
交付组是从一个或多个计算机目录中选择的计算机的集合。它指定哪些用户可以使用这些计算机，以及可供这些用户使用的应用程序和桌面。

有关详细信息，请参阅 [Citrix DaaS](#) 文档中的“交付组创建”。

1. 登录到 [Citrix Cloud](#)。在左上角的菜单中，选择我的服务 > **DaaS**。
2. 在管理 > 完整配置中，选择左窗格中的交付组。
3. 在操作栏中选择创建交付组。交付组创建向导将打开。

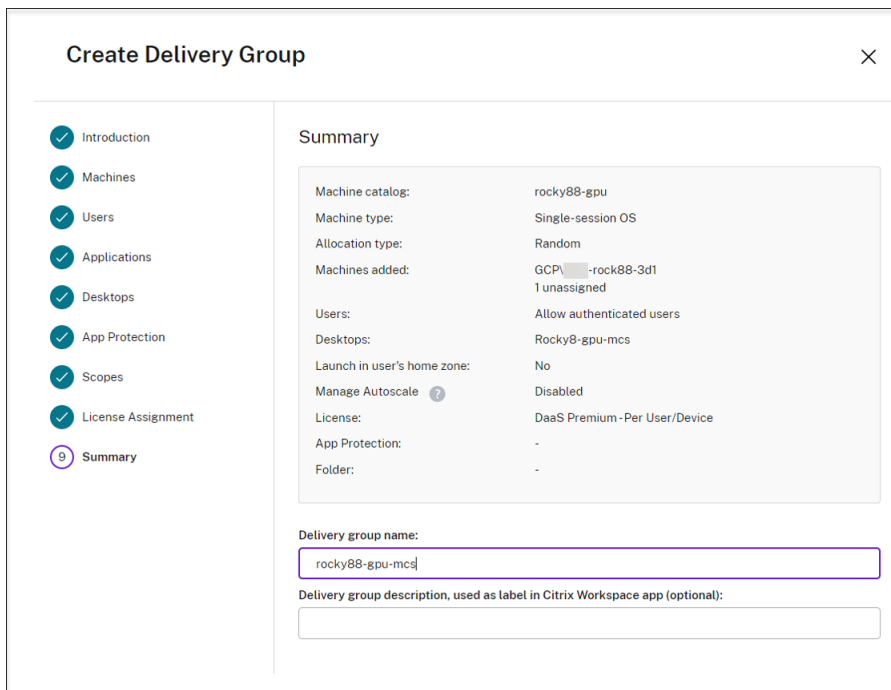
您看到的向导页面可能会有所差别，具体取决于您所做的选择。

4. 在计算机页面上，选择一个计算机目录并选择要从该目录中使用的计算机数。

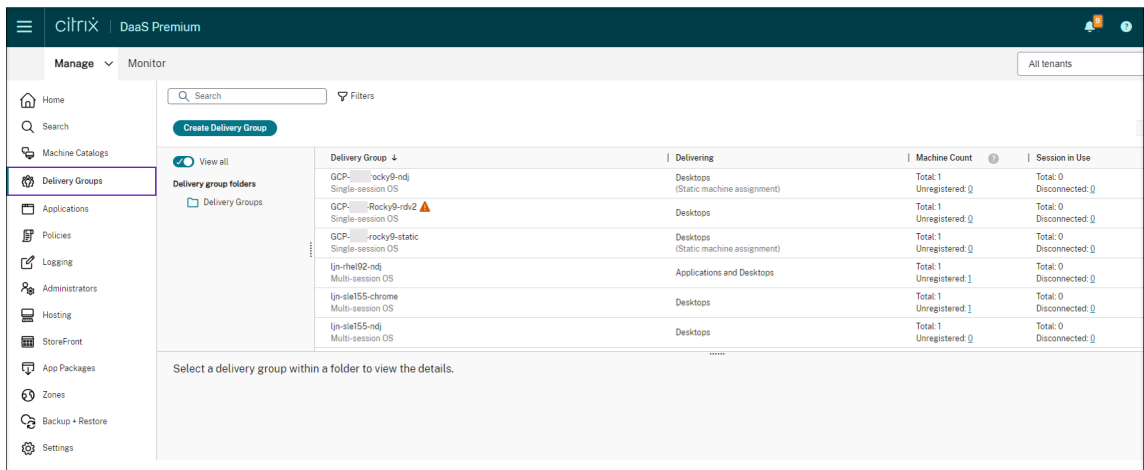


5. 在其他页面上配置其他设置。有关详细信息，请参阅[创建交付组](#)。
6. 在摘要页面上，输入交付组的名称。您也可以（选择性）输入说明，该说明显示在 Citrix Workspace 应用程序和完整配置管理界面中。例如：

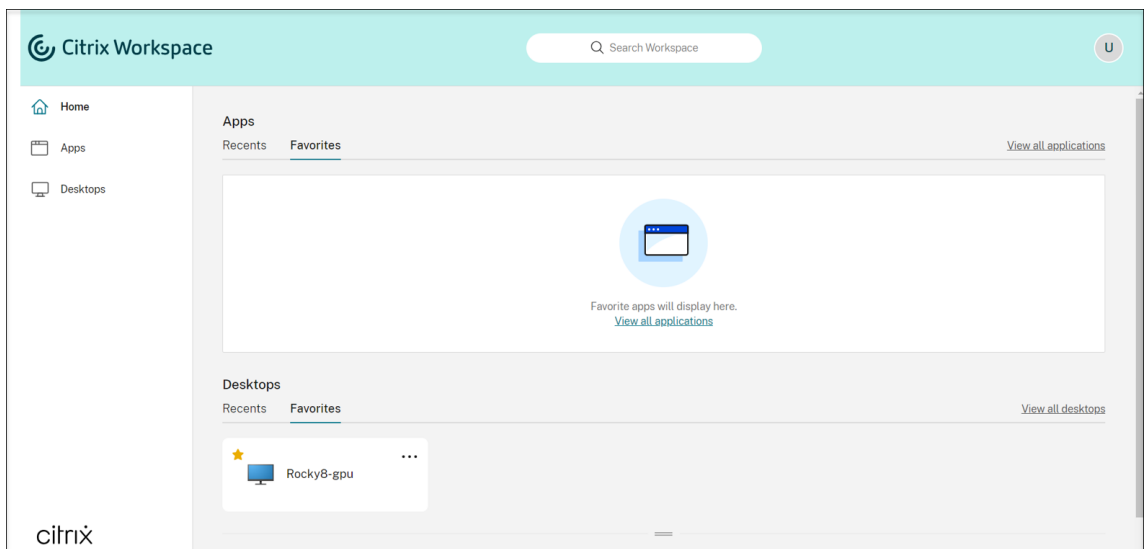
输入交付组的名称：



完整配置管理界面中的交付组列表：



Citrix Workspace 应用程序中已交付的计算机



使用 **Ansible** 管理您的部署

August 20, 2024

Ansible 有助于在整个部署中自动执行部署应用程序以及进行配置和更新的过程。本文提供了有关使用 Ansible 高效管理您的部署的分步说明。

步骤 **1**: 确定要部署的内容

在开始之前，您确定需要部署的内容，例如应用程序、服务、配置和环境变量。

第 2 步：设置您的 **Ansible** 项目

为您的 Ansible 项目创建目录结构。用来组织操作手册内容的一种关键方法是 Ansible 的“角色”功能。有关详细信息，请参阅 Ansible 文档中的[角色](#)。

下面是两个供您参考的示例目录结构：

示例目录结构 #1

```
1 production # inventory file for production servers
2 staging # inventory file for staging environment
3
4 group_vars/
5     group1.yml # here we assign variables to particular
6     group2.yml # groups
7 host_vars/
8     hostname1.yml # here we assign variables to particular
9     hostname2.yml # systems
10
11 library/ # if any custom modules, put them here (
12     optional)
13 module_utils/ # if any custom module_utils to support
14     modules, put them here (optional)
15 filter_plugins/ # if any custom filter plugins, put them here
16     (optional)
17
18 site.yml # master playbook
19 webserver.yml # playbook for webserver tier
20 dbserver.yml # playbook for dbserver tier
21
22 roles/
23     common/ # this hierarchy represents a "role"
24         tasks/ #
25             main.yml # <-- tasks file can include smaller files
26                 if warranted
27         handlers/ #
28             main.yml # <-- handlers file
29         templates/ # <-- files for use with the template
30             resource
31             ntp.conf.j2 # <----- templates end in .j2
32         files/ #
33             bar.txt # <-- files for use with the copy resource
34             foo.sh # <-- script files for use with the script
35                 resource
36         vars/ #
37             main.yml # <-- variables associated with this role
38         defaults/ #
39             main.yml # <-- default lower priority variables for
40                 this role
```

```
34     meta/                #
35         main.yml         # <-- role dependencies
36     library/             # roles can also include custom modules
37     module_utils/       # roles can also include custom module_utils
38     lookup_plugins/     # or other types of plugins, like lookup in
                        this case
39
40     webtier/             # same kind of structure as "common" was
                        above, done for the webtier role
41     monitoring/         # ""
42     fooapp/              # ""
```

示例目录结构 #2

```
1  inventories/
2    production/
3      hosts                # inventory file for production servers
4      group_vars/
5        group1.yml         # here we assign variables to particular
6          groups
7        group2.yml
8      host_vars/
9        hostname1.yml     # here we assign variables to particular
10         systems
11       hostname2.yml
12
13    staging/
14      hosts                # inventory file for staging environment
15      group_vars/
16        group1.yml         # here we assign variables to particular
17          groups
18        group2.yml
19      host_vars/
20        stagehost1.yml    # here we assign variables to particular
21         systems
22       stagehost2.yml
23
24    library/
25    module_utils/
26    filter_plugins/
27
28    site.yml
29    webservers.yml
30    dbservers.yml
31
32    roles/
33      common/
34      webtier/
35      monitoring/
36      fooapp/
```

第 3 步：配置您的清单

定义您的清单文件 (inventory.ini)。清单文件通常列出您要使用 Ansible 管理的主机，以及必要的详细信息，例如主机名、IP 地址和组成员身份。例如：

```
1 # Hostname and ip address
2 [UBUNTU2004]
3 <ip address>
4 [UBUNTU2204]
5 <ip address>
6 [RHEL8]
7 <ip address>
8 [RHEL9]
9 <ip address>
10 [DEBIAN11]
11 <ip address>
12 [DEBIAN12]
13 <ip address>
14 [SUSE15]
15 <ip address>
16
17 [all:children]
18 UBUNTU2004
19 UBUNTU2204
20 RHEL8
21 RHEL9
22 DEBIAN11
23 DEBIAN12
24 SUSE15
25
26 [all:vars]
27 ansible_user=<ansible execute user e.g root>
28 ansible_password=<>
29 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

第 4 步：创建 Ansible 操作手册

创建操作手册 (.yaml 文件) 以自动执行部署任务。本部分提供了用于自动执行各种部署任务的示例操作手册。

用于修补 Linux 发行版的示例操作手册

要使用 Ansible 修补各种 Linux 发行版，您可以创建一个名为 patch-for-different-distribution.yaml 的 YAML 操作手册文件，并在该文件中填充与以下内容类似的任务。hosts 指令定义清单中（在此上下文中为 inventory.ini）将用来执行操作手册的目标主机。

```
1 - name: Upgrade and Reboot RHEL & Debian family Linux distros
2   hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
```

```
3 vars:
4   reboot_connect_timeout: 5
5   reboot_post_reboot_delay: 15
6   reboot_timeout: 600
7 tasks:
8   # Upgrade RHEL family OS packages
9   - name: Upgrade RHEL Family OS packages
10     ansible.builtin.yum:
11       name: '*'
12       state: latest
13     when:
14       - ansible_facts['distribution'] == "RedHat"
15       - ansible_facts['distribution_major_version'] == "7"
16
17   # Upgrade RHEL family OS packages
18   - name: Upgrade RHEL Family OS packages
19     ansible.builtin.yum:
20       name: '*'
21       state: latest
22     when:
23       - ansible_facts['distribution'] == "RedHat"
24       - ansible_facts['distribution_major_version'] == "8"
25
26   # Upgrade RHEL family OS packages
27   - name: Upgrade RHEL Family OS packages
28     ansible.builtin.yum:
29       name: '*'
30       state: latest
31     when:
32       - ansible_facts['distribution'] == "RedHat"
33       - ansible_facts['distribution_major_version'] == "9"
34
35   # Ubuntu Family upgrade
36   - name: Update repositories cache
37     apt:
38       update_cache: yes
39     when:
40       - ansible_facts['distribution'] == "Ubuntu"
41       - ansible_facts['distribution_major_version'] == "20"
42
43   - name: Update all packages to their latest version
44     apt:
45       name: "*"
46       state: latest
47     when:
48       - ansible_facts['distribution'] == "Ubuntu"
49       - ansible_facts['distribution_major_version'] == "22"
50
51   # Debian Family upgrade
52   - name: Upgrade the OS (apt-get dist-upgrade)
53     apt:
54       upgrade: dist
55     when:
```

```

56     - ansible_facts['distribution'] == "Debian"
57     - ansible_facts['distribution_major_version'] == "11"
58
59     - name: Upgrade the OS (apt-get dist-upgrade)
60       apt:
61         upgrade: dist
62       when:
63         - ansible_facts['distribution'] == "Debian"
64         - ansible_facts['distribution_major_version'] == "12"
65
66     # Reboot after upgrade
67     - name: Reboot host
68       reboot:
69         connect_timeout: ""
70         post_reboot_delay: ""
71         reboot_timeout: ""

```

用于安装.**Net** 环境的示例操作手册

以下示例操作手册在特定的 Linux 发行版上安装各种版本的.NET 环境。

```

1  - name: Install dotnet runtime environment on Linux distros
2  hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
3  tasks:
4    # Install dotnet runtime environment on RHEL7
5    - name: Enable the rhel-7-server-dotnet-rpms repository
6      command: subscription-manager repos --enable=rhel-7-server-dotnet
   -rpms
7    when:
8      - ansible_facts['distribution'] == "RedHat"
9      - ansible_facts['distribution_major_version'] == "7"
10
11   - name: Install dotnet runtime environment on RHEL7
12     ansible.builtin.yum:
13       name: rh-dotnet60-aspnetcore-runtime-6.0
14       state: present
15     when:
16       - ansible_facts['distribution'] == "RedHat"
17       - ansible_facts['distribution_major_version'] == "7"
18
19   - name: Remove /usr/bin/dotnet if it exists
20     file:
21       path: /usr/bin/dotnet
22       state: absent
23
24   - name: Create a symbolic link
25     file:
26       src: /opt/rh/rh-dotnet60/root/usr/lib64/dotnet/dotnet
27       dest: /usr/bin/dotnet
28       state: link
29

```

```
30 # RHEL8 linux vda install dotnet runtime environment
31 - name: Install dotnet-runtime-8.0
32   ansible.builtin.dnf:
33     name: dotnet-runtime-8.0
34     state: present
35   when:
36     - ansible_facts['distribution'] == "RedHat"
37     - ansible_facts['distribution_major_version'] == "8"
38
39 - name: Install aspnetcore-runtime-8.0
40   ansible.builtin.dnf:
41     name: aspnetcore-runtime-8.0
42     state: present
43   when:
44     - ansible_facts['distribution'] == "RedHat"
45     - ansible_facts['distribution_major_version'] == "8"
46
47 # RHEL9 linux vda install dotnet runtime environment
48 - name: Install dotnet-runtime-8.0
49   ansible.builtin.dnf:
50     name: dotnet-runtime-8.0
51     state: present
52   when:
53     - ansible_facts['distribution'] == "RedHat"
54     - ansible_facts['distribution_major_version'] == "9"
55
56 - name: Install aspnetcore-runtime-8.0
57   ansible.builtin.dnf:
58     name: aspnetcore-runtime-8.0
59     state: present
60   when:
61     - ansible_facts['distribution'] == "RedHat"
62     - ansible_facts['distribution_major_version'] == "9"
63
64 # Ubuntu20.04 linux vda install dotnet runtime environment
65 - name: Register Microsoft key and feed
66   shell: |
67     wget https://packages.microsoft.com/config/ubuntu/20.04/
68           packages-microsoft-prod.deb -O packages-microsoft-prod.deb
69     dpkg -i packages-microsoft-prod.deb
70     rm packages-microsoft-prod.deb
71   when:
72     - ansible_facts['distribution'] == "Ubuntu"
73     - ansible_facts['distribution_major_version'] == "20"
74
75 - name: Install dotnet-runtime-8.0
76   ansible.builtin.apt:
77     name: dotnet-runtime-8.0
78     state: present
79     update_cache: yes
80   when:
81     - ansible_facts['distribution'] == "Ubuntu"
82     - ansible_facts['distribution_major_version'] == "20"
```

```
82
83   - name: Install aspnetcore-runtime-8.0
84     ansible.builtin.apt:
85       name: aspnetcore-runtime-8.0
86       state: present
87       update_cache: yes
88     when:
89       - ansible_facts['distribution'] == "Ubuntu"
90       - ansible_facts['distribution_major_version'] == "20"
91
92   # Ubuntu22.04 linux vda install dotnet runtime environment
93   - name: Install dotnet-runtime-8.0
94     ansible.builtin.apt:
95       name: dotnet-runtime-8.0
96       state: present
97       update_cache: yes
98     when:
99       - ansible_facts['distribution'] == "Ubuntu"
100      - ansible_facts['distribution_major_version'] == "22"
101
102   - name: Install aspnetcore-runtime-8.0
103     ansible.builtin.apt:
104       name: aspnetcore-runtime-8.0
105       state: present
106       update_cache: yes
107     when:
108       - ansible_facts['distribution'] == "Ubuntu"
109       - ansible_facts['distribution_major_version'] == "22"
110
111   # Debian11 linux vda install dotnet runtime environment
112   - name: Register Microsoft key and feed
113     shell: |
114       wget https://packages.microsoft.com/config/debian/11/packages-
115         microsoft-prod.deb -O packages-microsoft-prod.deb
116       dpkg -i packages-microsoft-prod.deb
117       rm packages-microsoft-prod.deb
118     when:
119       - ansible_facts['distribution'] == "Debian"
120       - ansible_facts['distribution_major_version'] == "11"
121
122   - name: Install dotnet-runtime-8.0
123     ansible.builtin.apt:
124       name: dotnet-runtime-8.0
125       state: present
126       update_cache: yes
127     when:
128       - ansible_facts['distribution'] == "Debian"
129       - ansible_facts['distribution_major_version'] == "11"
130
131   - name: Install aspnetcore-runtime-8.0
132     ansible.builtin.apt:
133       name: aspnetcore-runtime-8.0
134       state: present
```

```
134     update_cache: yes
135     when:
136         - ansible_facts['distribution'] == "Debian"
137         - ansible_facts['distribution_major_version'] == "11"
138
139     # Debian12 linux vda install dotnet runtime environment
140     - name: Register Microsoft key and feed
141       shell: |
142         wget https://packages.microsoft.com/config/debian/12/packages-
143             microsoft-prod.deb -O packages-microsoft-prod.deb
144         dpkg -i packages-microsoft-prod.deb
145         rm packages-microsoft-prod.deb
146     when:
147         - ansible_facts['distribution'] == "Debian"
148         - ansible_facts['distribution_major_version'] == "12"
149
150     - name: Install dotnet-runtime-8.0
151       ansible.builtin.apt:
152         name: dotnet-runtime-8.0
153         state: present
154         update_cache: yes
155     when:
156         - ansible_facts['distribution'] == "Debian"
157         - ansible_facts['distribution_major_version'] == "12"
158
159     - name: Install aspnetcore-runtime-8.0
160       ansible.builtin.apt:
161         name: aspnetcore-runtime-8.0
162         state: present
163         update_cache: yes
164     when:
165         - ansible_facts['distribution'] == "Debian"
166         - ansible_facts['distribution_major_version'] == "12"
167
168     # Sles15 linux vda install dotnet runtime environment
169     - name: Register Microsoft key and feed
170       shell: |
171         sudo rpm -Uvh https://packages.microsoft.com/config/sles/15/
172             packages-microsoft-prod.rpm
173         sudo ln -s /etc/yum.repos.d/microsoft-prod.repo /etc/zypp/repos
174             .d/microsoft-prod.repo
175     when:
176         - ansible_facts['distribution'] == "SLES"
177         - ansible_facts['distribution_major_version'] == "15"
178
179     - name: Install dotnet-runtime-8.0
180       community.general.zypper:
181         name: dotnet-runtime-8.0
182         state: present
183         update_cache: yes
184     when:
185         - ansible_facts['distribution'] == "SLES"
186         - ansible_facts['distribution_major_version'] == "15"
```



```

184
185     - name: Install aspnetcore-runtime-8.0
186       community.general.zypper:
187         name: aspnetcore-runtime-8.0
188         state: present
189         update_cache: yes
190       when:
191         - ansible_facts['distribution'] == "SLES"
192         - ansible_facts['distribution_major_version'] == "15"
193
194     # Amazon2 linux vda install dotnet runtime environment
195     - name: Install dotnet-runtime-8.0
196       ansible.builtin.yum:
197         name: dotnet-runtime-8.0
198         state: present
199       when:
200         - ansible_facts['distribution'] == "Amazon"
201         - ansible_facts['distribution_major_version'] == "2"
202
203     - name: Install aspnetcore-runtime-8.0
204       ansible.builtin.yum:
205         name: aspnetcore-runtime-8.0
206         state: present
207       when:
208         - ansible_facts['distribution'] == "Amazon"
209         - ansible_facts['distribution_major_version'] == "2"

```

用于升级 **Linux VDA** 的示例操作手册

要使用 Ansible 自动升级 Linux VDA，您可以创建两个单独的操作手册。一个操作手册，例如 `get_the_build.yml`，专门用于下载 Linux VDA 软件包并将其传输到目标计算机（主机）。另一个操作手册，例如 `linux_upgrade.yml`，包含用于使用之前下载的软件包在目标计算机上升级 Linux VDA 的任务。

示例操作手册 `get_the_build.yml`

```

1 - hosts: localhost
2   name: Get the latest release build to local
3   vars:
4     build_url: <linux vda download link> # replace with your actual
5       value.
6     local_tmp: "/tmp/" # replace with your actual value.
7     remote_tmp: "/tmp/" # replace with your actual value.
8     linuxvda_file_name : "linux vda rpm/deb file name" # replace with
9       your actual value.
10  tasks:
11  - name: Download the file
12    get_url:
13      url: ""
14      dest: ""
15  tags:
16  - get

```

```
15
16
17 - hosts: <host1,host2,host3> # replace with your actual hosts in the
    inventory file.
18   name: Copy a file to remote location
19   tasks:
20     - name: Copy vda to the remote machine
21       ansible.builtin.copy:
22         src: ""
23         dest: ""
24         remote_src: no
25     tags:
26     - copy
```

示例操作手册 **linux_upgrade.yml**

```
1 - name: Upgrade Linux VDA and Reboot RHEL & Debian Linux distros
2   hosts: <host1,host2,host3> # replace with your actual hosts in the
    inventory file.
3   vars:
4     remote_tmp: "/path/to/remote/tmp" # replace with your actual path
5     rhel7_file_name: "rhel7_file.rpm" # replace with your actual file
    name
6     rhel8_file_name: "rhel8_file.rpm" # replace with your actual file
    name
7     rhel9_file_name: "rhel9_file.rpm" # replace with your actual file
    name
8     ubuntu2004_file_name: "ubuntu2004_file.deb" # replace with your
    actual file name
9     ubuntu2204_file_name: "ubuntu2204_file.deb" # replace with your
    actual file name
10    debian11_file_name: "debian11_file.deb" # replace with your actual
    file name
11    debian12_file_name: "debian12_file.deb" # replace with your actual
    file name
12    suse15_file_name: "suse15_file.deb" # replace with your actual file
    name
13    amazon2_file_name: "amazon2_file.rpm" # replace with your actual
    file name
14   tasks:
15     # Upgrade RHEL linux vda packages
16     - name: Upgrade RHEL7 linux vda packages
17       ansible.builtin.yum:
18         name: ""
19         state: present
20       when:
21         - ansible_facts['distribution'] == "RedHat"
22         - ansible_facts['distribution_major_version'] == "7"
23
24     # Upgrade RHEL linux vda packages
25     - name: Upgrade RHEL8 linux vda packages
26       ansible.builtin.yum:
27         name: ""
28         state: present
```

```
29     when:
30         - ansible_facts['distribution'] == "RedHat"
31         - ansible_facts['distribution_major_version'] == "8"
32
33     # Upgrade RHEL linux vda packages
34     - name: Upgrade RHEL9 linux vda packages
35       ansible.builtin.yum:
36         name: ""
37         state: present
38       when:
39         - ansible_facts['distribution'] == "RedHat"
40         - ansible_facts['distribution_major_version'] == "9"
41
42     # Ubuntu20.04 linux vda upgrade
43     - name: Ubuntu20.04 linux vda upgrade
44       ansible.builtin.apt:
45         deb: ""
46       when:
47         - ansible_facts['distribution'] == "Ubuntu"
48         - ansible_facts['distribution_major_version'] == "20"
49
50     - name: Ubuntu22.04 linux vda upgrade
51       ansible.builtin.apt:
52         deb: ""
53       when:
54         - ansible_facts['distribution'] == "Ubuntu"
55         - ansible_facts['distribution_major_version'] == "22"
56
57     # Debian Linux VDA upgrade
58     - name: Debian11 Linux VDA upgrade
59       ansible.builtin.apt:
60         deb: ""
61       when:
62         - ansible_facts['distribution'] == "Debian"
63         - ansible_facts['distribution_major_version'] == "11"
64
65     - name: Debian12 Linux VDA upgrade
66       ansible.builtin.apt:
67         deb: ""
68       when:
69         - ansible_facts['distribution'] == "Debian"
70         - ansible_facts['distribution_major_version'] == "12"
71
72     # Sles15 Linux VDA upgrade
73     - name: Sles15 Linux VDA upgrade
74       community.general.zypper:
75         name: ""
76         state: present
77       when:
78         - ansible_facts['distribution'] == "SLES"
79         - ansible_facts['distribution_major_version'] == "15"
80
81     # Amazon2 Linux VDA upgrade
```

```
82 - name: Amazon2 Linux VDA upgrade
83   ansible.builtin.yum:
84     name: ""
85   when:
86     - ansible_facts['distribution'] == "Amazon"
87     - ansible_facts['distribution_major_version'] == "2"
88   # Reboot after upgrade
89   - name: Reboot host
90     reboot:
91       connect_timeout: ""
92       post_reboot_delay: ""
93       reboot_timeout: ""
```

用于将网络文件系统 (NFS) 服务器装载为主目录的示例操作手册

以下示例操作手册将 NFS 服务器作为主目录装载到目标主机上:

```
1 - hosts: <host1,host2,host3> # replace with your actual hosts in the
   inventory file.
2 vars:
3   nfs_server = <nfsserver ip address> # replace with your actual
   values
4   mount_points = /home/<domain realm>/user1,/home/<domain realm>user2
   # replace with your actual values
5   nfs_shares = user1,user2 # replace with your actual values
6   owners = user1,user2 # replace with your actual values
7   groups = group1,group2 # replace with your actual values
8 tasks:
9   - name: Enable NFS as home directory
10     ansible.builtin.command:
11       cmd: "/opt/Citrix/VDA/bin/ctxreg create -k 'HKLM\System\
   CurrentControlSet\Control\Citrix' -t 'REG_DWORD' -v '
   CheckUserHomeMountPoint' -d '0x00000001' --force"
12     register: result
13     failed_when: result.rc != 0
14     check_mode: no
15
16   - name: Mount NFS shares
17     ansible.builtin.mount:
18       path: ""
19       src: ":"
20       fstype: nfs
21       opts: rw,noexec
22       state: mounted
23     loop: ""
24
25   - name: Set owner, group and mode for NFS client paths
26     ansible.builtin.file:
27       path: ""
28       owner: ""
29       group: ""
30       mode: ""
```

用于远程命令执行的示例操作手册

用于修改注册表设置的示例操作手册

```
1 - hosts: <host1,host2,host3> # replace with your actual hosts in the
  inventory file.
2 vars:
3   registry_key: "your_registry_key" # E.g. registry_key = HKLM\
  System\CurrentControlSet\Control\Terminal Server\Wds\icawd
4   registry_type: "your_registry_type" # E.g. registry_type =
  REG_DWORD
5   registry_value: "your_registry_value" # E.g. registry_value =
  AdaptiveScalingEnabled
6   registry_data: "your_registry_data" # E.g. registry_data = 0
  x00000000
7 tasks:
8 - name: Execute AdaptiveScaling redirection script
  ansible.builtin.command:
9   cmd: "/opt/Citrix/VDA/bin/ctxreg create -k "" -t "" -v "" -d ""
  --force"
11  register: result
12  failed_when: result.rc != 0
13  check_mode: no
```

用于锁定 RHEL 次要版本的示例操作手册

```
1 - hosts: <host1,host2,host3> # replace with your actual hosts in the
  inventory file.
2 vars:
3   rhel_minor_version: "9.3" # replace with your actual minor version
  such as 9.3, 8.8
4 tasks:
5 - name: Lock system to a specific minor version
  ansible.builtin.command:
6   cmd: "subscription-manager release --set="
7   register: result
8   failed_when: "'Error' in result.stderr"
```



© 2024 Cloud Software Group, Inc. All rights reserved. Cloud Software Group, the Cloud Software Group logo, and other marks appearing herein are property of Cloud Software Group, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner(s).