Linux Virtual Delivery Agent 2104

Contents

What's new

June 24, 2022

What's new in 2104

Version 2104 of the Linux VDA includes the following new features and enhancements:

Support for non-domain-joined Linux VDAs in the Citrix Virtual Apps and Desktops service

You can now use MCS to create non-domain-joined Linux VDAs in the Citrix Virtual Apps and Desktops service. For more information, see Non-domain-joined.

OpenJDK 11 being required

The Linux VDA now requires the presence of OpenJDK 11. Among the Linux distributions that the Linux VDA supports, only Ubuntu 16.04 requires you to install OpenJDK 11 manually. On the other supported distributions, OpenJDK 11 is installed automatically as a dependency when you install the Linux VDA.

XDPing changes

Running `ctxsetup.sh` no longer installs XDPing. You can run `sudo /opt/Citrix/VDA/bin/xdping` to install XDPing. This command also creates a Python3 virtual environment that is required for XDPing. For more information, see XDPing

Smart card support for Ubuntu

Users can use a smart card connected to the client device for authentication when logging on to a Linux virtual desktop session. This release further lets you use smart card pass-through authentication in Ubuntu 20.04, Ubuntu 18.04, and Ubuntu 16.04 sessions. For more information, see Pass-through authentication by using smart cards.

Support for the MATE desktop

We have added support for the lightweight MATE desktop on CentOS, RHEL, Ubuntu, and Debian. You can now specify the MATE or GNOME desktop through a new variable available in the ctxinstall.sh,

ctxsetup.sh, and deploymcs.sh scripts. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set this variable value to mate. For more information, see Easy install and the manual installation articles such as Install Linux Virtual Delivery Agent for RHEL/CentOS.

PBIS support for RHEL 8, CentOS 8, and SUSE 12.5

We have added PBIS support for joining RHEL 8, CentOS 8, and SUSE 12.5 machines to Windows domains.

Fixed issues

December 21, 2021

The following issues have been fixed since Linux Virtual Delivery Agent 2103:

- When you press the Shift+Tab keys in a Citrix Workspace app for Mac or Linux session, the keys might not register as expected. [LNXVDA-9744]

Known issues

July 3, 2023

The following issues have been identified in this release:

- In a GNOME desktop session, attempts to change the keyboard layout might fail. [CVADHELP-15639]
- Non-seamless published applications might exit shortly after launch. The issue occurs after a Mutter upgrade to a version later than mutter-3.28.3-4. To work around the issue, use mutter-3.28.3-4 or earlier. [LNXVDA-6967]
- The Linux VDA does not work as expected when you use NVIDIA GRID 3D cards without enabling HDX 3D Pro. The issue occurs on RHEL 7.7 and earlier, SUSE 12.5 and earlier, and Ubuntu 16.04. The reason is that multiple OpenGL libraries cannot coexist in the graphics systems of these Linux distributions.
- An unexpected window appears during file download. The window does not affect the file download functionality and it disappears automatically after a while. [LNXVDA-5646]

- The default settings of PulseAudio cause the sound server program to exit after 20 seconds of inactivity. When PulseAudio exits, audio does not work. To work around this issue, set exit-idle-time=-1 in the /etc/pulse/daemon.conf file. [LNXVDA-5464]
- `libtcmalloc 4.3.0` in SUSE 12.5 might cause processes to exit unexpectedly.
- The `ctxhdx` service might exit unexpectedly on the Ubuntu 16.04 and SUSE 12.5 VDAs. The issue occurs with the GNU C Library (`glibc`) Versions 2.22 through 2.24. The issue is fixed in `glibc 2.25`. If you are using the SUSE 12.5 distribution, you can install the patch that SUSE provides for fixing the issue. No fix is available for Ubuntu 16.04 at the time the Linux VDA is released. [LNXVDA-4481]
- Sessions cannot be launched in Citrix Workspace app for Linux when SSL encryption is enabled and session reliability is disabled. [RFLNX-1557]
- The `indicator-datetime-service` process does not consume the $TZ environment variable. When the client and session locate in different time zones, the unity panel on Ubuntu 16.04 Unity Desktop does not show the time of the client. [LNXVDA-2128]
- Ubuntu graphics: In HDX 3D Pro, a black frame might appear around applications after resizing the Desktop Viewer, or sometimes, the background can appear black.
- Printers created by the Linux VDA printing redirection might not be removed after logging out of a session.
- CDM files are missing when a directory contains numerous files and subdirectories. This issue might occur if the client side has too many files or directories.
- In this release, only UTF-8 encoding is supported for non-English languages.
- Citrix Workspace app for Android CAPS LOCK state might be reversed during session roaming. The CAPS LOCK state can be lost when roaming an existing connection to Citrix Workspace app for Android. As a workaround, use the Shift key on the extended keyboard to switch between upper case and lower case.
- Shortcut keys with ALT do not always work when you connect to the Linux VDA using Citrix Workspace app for Mac. Citrix Workspace app for Mac sends AltGr for both left and right Options/Alt keys by default. You can modify this behavior within the Citrix Workspace app settings but the results vary with different applications.
- Registration fails when the Linux VDA is rejoined to the domain. The rejoining generates a fresh set of Kerberos keys. But, the Broker might use a cached out-of-date VDA service ticket based on the previous set of Kerberos keys. When the VDA tries to connect to the Broker, the Broker might not be able to establish a return security context to the VDA. The usual symptom is that the VDA registration fails.

  This problem can eventually resolve itself when the VDA service ticket expires and is renewed. But because service tickets are long-lived, it can take a long time.

As a workaround, clear the Broker's ticket cache. Restart the Broker or run the following command on the Broker from a command prompt as Administrator:

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

This command purges all service tickets in the LSA cache held by the Network Service principal under which the Citrix Broker Service runs. It removes service tickets for other VDAs and potentially other services. However, it is harmless –these service tickets can be reacquired from the KDC when needed again.

- Audio plug-n-play is not supported. You can connect an audio capture device to the client machine before starting to record audio in the ICA session. If a capture device is attached after the audio recording application has started, the application might become unresponsive and you must restart it. If a capture device is unplugged while recording, a similar issue might occur.
- Citrix Workspace app for Windows might experience audio distortion during audio recording.

Third party notices

October 20, 2021

Linux Virtual Delivery Agent Version 2104 (PDF Download)

This release of the Linux VDA can include third party software licensed under the terms defined in the document.

Deprecation

June 11, 2021

The announcements in this article are intended to give you advanced notice of platforms, Citrix products, and features that are being phased out so that you can make timely business decisions. Citrix monitors customer use and feedback to determine when they are withdrawn. Announcements can change in subsequent releases and might not include every deprecated feature or functionality.
For details about product lifecycle support, see the Product Lifecycle Support Policy article.

Deprecations and removals

The following table shows the platforms, Citrix products, and features that are deprecated or removed.

Deprecated items are not removed immediately. Citrix continues to support them in this release but they will be removed in a future Current Release.

Removed items are either removed, or are no longer supported, in the Linux VDA.

Item	Deprecation announced in	Removed in
Support for RHEL 7.7, CentOS 7.7	2006	2009
Support for SUSE 12.3	2006	2006
Support for RHEL 6.10, CentOS 6.10	2003	2003
Support for RHEL 6.9, CentOS 6.9	1909	1909
Support for RHEL 7.5, CentOS 7.5	1903	1903
Support for RHEL 7.4, CentOS 7.4	1811	1811
Support for RHEL 6.8, CentOS 6.8	1811	1811
Support for RHEL 7.3, CentOS 7.3	7.18	7.18
Support for RHEL 6.6, CentOS 6.6	7.16	7.16
SUSE 11.4	7.16	7.16

System requirements

October 22, 2021

Linux distributions

> **Note:**
>
> System requirements for components not covered here (such as Citrix Workspace app) are described in their respective documentation sets.
>
> Before installing the Linux VDA, install .NET Core Runtime 3.1 according to the instructions at https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers.

For more information about using this Current Release (CR) in a Long Term Service (LTSR) environment and other FAQs, see Knowledge Center article.

The Linux VDA supports the following Linux distributions:

- SUSE Linux Enterprise:
  - Server 12 Service Pack 5 + SUSE Linux Enterprise Workstation Extension 12 SP5
  - Server 12 Service Pack 5
- Red Hat Enterprise Linux
  - Workstation 8.3
  - Workstation 8.2
  - Workstation 8.1
  - Workstation 7.9
  - Workstation 7.8
  - Server 8.3
  - Server 8.2
  - Server 8.1
  - Server 7.9
  - Server 7.8
- CentOS Linux
  - CentOS 8.3
  - CentOS 8.2
  - CentOS 8.1
  - CentOS 7.9
  - CentOS 7.8
- Ubuntu Linux
  - Ubuntu Desktop 20.04
  - Ubuntu Server 20.04
  - Ubuntu Desktop 18.04
  - Ubuntu Server 18.04
  - Ubuntu Live Server 18.04

  - Ubuntu Desktop 16.04
  - Ubuntu Server 16.04

- Debian Linux
  - Debian 10.7

> **Note:**
>
> CentOS project shifts focus to CentOS Stream. CentOS Linux 8, as a rebuild of RHEL 8, ends at the end of 2021. CentOS Stream continues after that date, serving as the upstream (development) branch of Red Hat Enterprise Linux. For more information, see https://www.redhat.com/en/blog/centos-stream-building-innovative-future-enterprise-linux.

The Linux VDA supports several methods for integrating Linux machines with Microsoft Active Directory (AD):

	Winbind	SSSD	Centrify	PBIS	Quest
RHEL 8.3	Yes	Yes	Yes	Yes	No
CentOS 8.3	Yes	Yes	Yes	Yes	No
RHEL 8.2	Yes	Yes	Yes	Yes	No
CentOS 8.2	Yes	Yes	Yes	Yes	No
RHEL 8.1	Yes	Yes	Yes	Yes	No
CentOS 8.1	Yes	Yes	Yes	Yes	No
RHEL 7.9	Yes	Yes	Yes	Yes	Yes
CentOS 7.9	Yes	Yes	Yes	Yes	Yes
RHEL 7.8	Yes	Yes	Yes	Yes	Yes
CentOS 7.8	Yes	Yes	Yes	Yes	Yes
Ubuntu 20.04	Yes	Yes	Yes	Yes	Yes
Ubuntu 18.04	Yes	Yes	Yes	Yes	Yes
Ubuntu 16.04	Yes	Yes	Yes	Yes	Yes
Debian 10.7	Yes	Yes	Yes	Yes	No
SUSE 12.5	Yes	Yes	Yes	Yes	Yes

For a matrix of the Linux distributions and the Xorg versions that this version of the Linux VDA supports, see the following table. For more information, see XorgModuleABIVersions.

Linux distribution	Xorg version
RHEL 8.3, CentOS 8.3	1.20.8
RHEL 8.2, CentOS 8.2	1.20.8
RHEL 8.1, CentOS 8.1	1.20.8
RHEL 7.9, CentOS 7.9	1.20
RHEL 7.8, CentOS 7.8	1.20
Ubuntu 20.04	1.20
Ubuntu 18.04	1.19
Ubuntu 16.04	1.18
Debian 10.7	1.20
SUSE 12.5	1.19

Do not use HWE kernel or HWE Xorg on Ubuntu.

Use PulseAudio 13.99 on RHEL 8.x and CentOS 8.x.

In all cases, the supported processor architecture is x86-64.

> **Note:**
>
> - Citrix's support for a Linux OS platform and version expires when the support from the OS vendor expires.
> - GNOME and KDE desktops are supported on SUSE 12, RHEL 7, CentOS 7, RHEL 8, and CentOS 8. Unity desktop is supported on Ubuntu 16.04. GNOME desktop is supported on Ubuntu 20.04, Ubuntu 18.04, and Debian 10.7. MATE desktop is supported on all Linux distributions that the Linux VDA supports, except SUSE 12.5. At least one desktop must be installed.

Citrix Virtual Desktops

The Linux VDA is compatible with all currently supported versions of Citrix Virtual Apps and Desktops. For information about the Citrix product lifecycle, and to find out when Citrix stops supporting specific versions of products, see the Citrix Product Lifecycle Matrix.

The configuration process for Linux VDAs differs slightly from Windows VDAs. However, any Delivery Controller farm is able to broker both Windows and Linux desktops.

Supported host platforms and virtualization environments

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

> **Tip:**
>
> See the vendor's documentation for the list of supported platforms.
>
> **Note:**
>
> Azure, AWS, and GCP are compatible only with the Citrix Virtual Apps and Desktops service. Bare metal servers are not supported when MCS is used to create virtual machines.

Active Directory integration packages

The Linux VDA supports the following Active Directory integration packages or products:

- Samba Winbind
- Quest Authentication Services v4.1 or later
- Centrify DirectControl
- SSSD
- PBIS (compatible with RHEL 7, Ubuntu, and Debian)

> **Tip:**
>
> For the list of supported platforms, see the documentation from the vendors of the Active Directory integration packages.

HDX 3D Pro

The following hypervisors and NVIDIA GRID™ GPU are required to support HDX 3D Pro.

Hypervisors

- Citrix Hypervisor

- VMware ESX and ESXi
- Nutanix AHV

> **Note:**
>
> The hypervisors are compatible with certain Linux distributions.

GPU

The Linux VDA supports the following GPUs for GPU pass-through:

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

The Linux VDA supports the following GPUs for vGPU:

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

Installation overview

July 13, 2021

There are options for you to install the Linux VDA. You can do a fresh installation or upgrade an existing installation from the previous two versions and from an LTSR release.

- Easy install. After installing the Linux VDA package on a machine, you can configure the running environment by using the ctxinstall.sh script. For more information, see Easy install.
- Create Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure: You can create both domain-joined and non-domain-joined Linux VDAs in Citrix Virtual Apps and Desktops Standard

for Azure to deliver virtual apps and desktops to any device from Microsoft Azure. For more information, see Create Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure.

- MCS. You can use MCS to create Linux VMs in batches where the Linux VDA package is also installed. For more information, see Use MCS to create Linux VMs.
- Use Citrix Provisioning to create Linux VMs: You can provision Linux virtual desktops directly in the Citrix Virtual Apps and Desktops environment. For more information, see Streaming Linux target devices in the Citrix Provisioning documentation.
- Manual installation. You can use the following general steps to install the Linux VDA. Variations and specific commands are documented by distribution. For more information, see Install Linux Virtual Delivery Agent for RHEL/CentOS, Install Linux Virtual Delivery Agent for SUSE, Install Linux Virtual Delivery Agent for Ubuntu, and Install Linux Virtual Delivery Agent for Debian.

  1. Prepare for installation.
  2. Prepare the hypervisor.
  3. Add the Linux virtual machine (VM) to the Windows domain.
  4. Install the Linux VDA.
  5. Configure the Linux VDA.
  6. Create the machine catalog in Citrix Virtual Apps or Citrix Virtual Desktops.
  7. Create the delivery group in Citrix Virtual Apps or Citrix Virtual Desktops.

XDPing

You can use the Linux XDPing tool to check for common configuration issues with a Linux VDA environment. For more information, see XDPing.

Install .NET Core Runtime 3.1 as a prerequisite

Before installing the Linux VDA, install .NET Core Runtime 3.1 according to the instructions at https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers.

After installing .NET Core Runtime 3.1, run the **which dotnet** command to find your runtime path.

Based on the command output, set the .NET Core runtime binary path. For example, if the command output is /aa/bb/dotnet, use /aa/bb as the .NET binary path.

Configure Delivery Controllers

March 22, 2022

XenDesktop 7.6 and earlier versions require changes to support the Linux VDA. For those versions, a hotfix or update script is required. The installation and verification instructions are provided in this article.

Update Delivery Controller configuration

For XenDesktop 7.6 SP2, apply Hotfix Update 2 to update the Broker for Linux Virtual Desktops. Hotfix Update 2 is available here:

CTX142438: Hotfix Update 2 - For Delivery Controller 7.6 (32-bit) –English

For versions earlier than XenDesktop 7.6 SP2, you can use the PowerShell script named **Update-BrokerServiceConfig.ps1** to update the Broker Service configuration. This script is available in the following package:

- citrix-linuxvda-scripts.zip

Repeat the following steps on every Delivery Controller in the farm:

1. Copy the **Update-BrokerServiceConfig.ps1** script to the Delivery Controller machine.
2. Open a Windows PowerShell console in the context of the local administrator.
3. Browse to the folder containing the **Update-BrokerServiceConfig.ps1** script.
4. Run the **Update-BrokerServiceConfig.ps1** script:

```
.\Update-BrokerServiceConfig.ps1
<!--NeedCopy-->
```

> **Tip:**
>
> By default, PowerShell is configured to prevent the execution of PowerShell scripts. If the script fails to run, change the PowerShell execution policy before trying again:
>
> ```
> Set-ExecutionPolicy Unrestricted
> <!--NeedCopy-->
> ```

The **Update-BrokerServiceConfig.ps1** script updates the Broker Service configuration file by using new WCF endpoints required by the Linux VDA and restarts the Broker Service. The script determines the location of the Broker Service configuration file automatically. A backup of the original configuration file is created in the same directory, with **.prelinux** appended to the file name.

These changes have no impact on the brokering of Windows VDAs configured to use the same Delivery Controller farm. A single Controller farm can manage and broker sessions for both Windows and Linux VDAs seamlessly.

> **Note:**
>
> The Linux VDA does not support Secure ICA for encryption. Enabling Secure ICA on the Linux VDA causes session launch failure.

Verify Delivery Controller configuration

When the required configuration changes have been applied to a Delivery Controller, the **EndpointLinux** string appears five times in the **%PROGRAMFILES%\Citrix\Broker\Service\BrokerService.exe.confi** file.

From the Windows command prompt, log on as a local administrator to check:

```
cd "%PROGRAMFILES%"\Citrix\Broker\Service\
findstr EndpointLinux BrokerService.exe.config
<!--NeedCopy-->
```

Easy install

June 10, 2022

Easy install is supported as of Version 7.13 of the Linux VDA. This feature helps you set up a running environment of the Linux VDA by installing the necessary packages and customizing the configuration files automatically.

Use easy install

To use this feature, do the following:

1. Prepare configuration information and the Linux machine.
2. Install the Linux VDA package.
   Go to the Citrix Virtual Apps and Desktops download page. Expand the appropriate version of Citrix Virtual Apps and Desktops and click **Components** to download the Linux VDA package that matches your Linux distribution.
3. Set up the runtime environment to complete the Linux VDA installation.

Step 1: Prepare configuration information and the Linux machine

Collect the following configuration information needed for easy install:

- Host name - Host name of the machine on which the Linux VDA is to be installed
- IP address of Domain Name Server
- IP address or string name of NTP Server
- Domain Name - The NetBIOS name of the domain
- Realm Name - The Kerberos realm name
- FQDN of Active Domain - Fully qualified domain name

> **Important:**
>
> - To install the Linux VDA, verify that the repositories are added correctly on the Linux machine.
> - To launch a session, verify that the X Window system and desktop environments are installed.

Considerations

- The workgroup name, by default, is the domain name. To customize the workgroup in your environment, do the following:

  a. Create the /tmp/ctxinstall.conf file on the Linux VDA machine.
  b. Add the workgroup=<your workgroup> line to the file and save your changes.

- Centrify does not support pure IPv6 DNS configuration. At least one DNS server using IPv4 is required in /etc/resolv.conf for `adclient` to find AD services properly.

  Log:

  ```
  ADSITE    : Check that this machine's subnet is in a site known by
      AD    : Failed
            : This machine's subnet is not known by AD.
            : We guess you should be in the site Site1.
  <!--NeedCopy-->
  ```

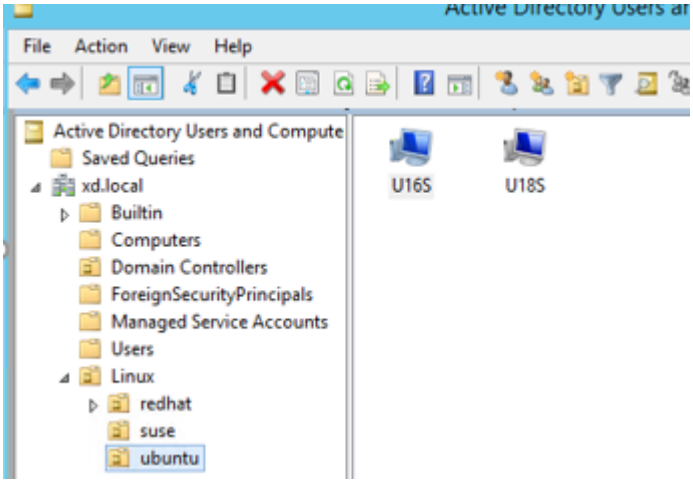
  This issue is unique to Centrify and its configuration. To resolve this issue, do the following:

  a. Open **Administrative Tools** on the domain controller.
  b. Select **Active Directory Sites and Services**.
  c. Add a proper subnet address for **Subnets**.

- To join your VDA to a specific OU, do the following:

  1. Ensure that the specific OU exists on the domain controller.

     For an example OU, see the following screen capture



2. Create the /tmp/ctxinstall.conf file on the VDA.
3. Add the ou=<your ou> line to the /tmp/ctxinstall.conf file.

   OU values vary with different AD methods. See the following table.

OS	Winbind	SSSD	Centrify	PBIS
RHEL 8	ou=”OU=redhat,OU=Linux”	ou=”OU=redhat,OU=Linux”	ou=”XD.LOCAL/Linux/redhat”	ou=”Linux/redhat”
RHEL 7	ou=”Linux/redhat”	ou=”Linux/redhat”	ou=”XD.LOCAL/Linux/redhat”	ou=”Linux/redhat”
Ubuntu	ou=”Linux/ubuntu”	ou=”Linux/ubuntu”	ou=”XD.LOCAL/Linux/ubuntu”	ou=”Linux/ubuntu”
SUSE 12.5	ou=”Linux/suse”	ou=”Linux/suse”	ou=”XD.LOCAL/Linux/suse”	ou=”Linux/suse”
Debian	ou=”Linux/debian”	ou=”Linux/debian”	ou=”XD.LOCAL/Linux/debian”	ou=”Linux/debian”

- Easy install supports pure IPv6 as of Linux VDA 7.16. The following preconditions and limitations apply:
  - Your Linux repository must be configured to ensure that your machine can download required packages over pure IPv6 networks.
  - Centrify is not supported on pure IPv6 networks.

> **Note:**
>
> If your network is pure IPv6 and all your input is in proper IPv6 format, the VDA registers with the Delivery Controller through IPv6. If your network has a hybrid IPv4 and IPv6 con-

> figuration, the type of the first DNS IP address determines whether IPv4 or IPv6 is used for registration.

- If you choose Centrify as the method to join a domain, the ctxinstall.sh script requires the Centrify package. There are two ways for ctxinstall.sh to get the Centrify package:
  - Easy install helps download the Centrify package from the Internet automatically. The following are the URLs for each distribution:

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

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

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

    Ubuntu/Debian: wget https://downloads.centrify.com/products/infrastructure-services/19.9/centrify-infrastructure-services-19.9-deb8-x86_64.tgz?_ga=2.151462329.1042350071.1592881996-604509155.1572850145
  - Fetch the Centrify package from a local directory. To designate the directory of the Centrify package, do the following:

    a. Create the /tmp/ctxinstall.conf file on the Linux VDA server if it does not exist.
    b. Add the “centrifypkgpath=<path name>”line to the file.

    For example:

```
cat /tmp/ctxinstall.conf
set “centrifypkgpath=/home/mydir”
ls -ls /home/mydir
    9548 -r-xr-xr-x. 1 root root  9776688 May 13  2016
       adcheck-rhel4-x86_64
    4140 -r--r--r--. 1 root root  4236714 Apr 21  2016
   centrifyda-3.3.1-rhel4-x86_64.rpm
    33492 -r--r--r--. 1 root root 34292673 May 13  2016
   centrifydc-5.3.1-rhel4-x86_64.rpm
    4 -rw-rw-r--. 1 root root     1168 Dec  1  2015
   centrifydc-install.cfg
    756 -r--r--r--. 1 root root   770991 May 13  2016
   centrifydc-ldapproxy-5.3.1-rhel4-x86_64.rpm
    268 -r--r--r--. 1 root root   271296 May 13  2016
   centrifydc-nis-5.3.1-rhel4-x86_64.rpm
    1888 -r--r--r--. 1 root root  1930084 Apr 12  2016
   centrifydc-openssh-7.2p2-5.3.1-rhel4-x86_64.rpm
    124 -rw-rw-r--. 1 root root   124543 Apr 19  2016
   centrify-suite.cfg
    0 lrwxrwxrwx. 1 root root       10 Jul  9  2012 install-
   express.sh -> install.sh
```



```
13       332 -r-xr-xr--. 1 root root   338292 Apr 10  2016 install
         .sh
14        12 -r--r--r--. 1 root root    11166 Apr  9  2015 release-
         notes-agent-rhel4-x86_64.txt
15         4 -r--r--r--. 1 root root     3732 Aug 24  2015 release-
         notes-da-rhel4-x86_64.txt
16         4 -r--r--r--. 1 root root     2749 Apr  7  2015 release-
         notes-nis-rhel4-x86_64.txt
17        12 -r--r--r--. 1 root root     9133 Mar 21  2016 release-
         notes-openssh-rhel4-x86_64.txt
18   <!--NeedCopy-->
```

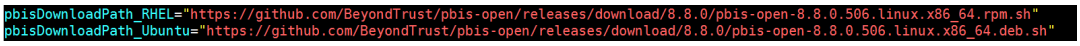
- If you choose PBIS as the method to join a domain, the ctxinstall.sh script requires the PBIS package. There are two ways for ctxinstall.sh to get the PBIS package:
  - Easy install helps download the PBIS package from the Internet automatically. The following are the URLs for each distribution:

    RHEL 7 / CentOS 7 / SUSE 12.5: `wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.rpm.sh`

    RHEL 8 / CentOS 8: `wget https://github.com/BeyondTrust/pbis-open/releases/download/9.1.0/pbis-open-9.1.0.551.linux.x86_64.rpm.sh`

    Ubuntu/Debian: `wget https://github.com/BeyondTrust/pbis-open/releases/download/8.8.0/pbis-open-8.8.0.506.linux.x86_64.deb.sh`
  - Fetch a specific version of the PBIS package from the Internet. To do so, change the "pbisDownloadPath"line in the /opt/Citrix/VDA/sbin/ctxinstall.sh file to designate the URL of the PBIS package.

    For an example, see the following screen capture:



Step 2: (For Ubuntu 16.04 only) Install OpenJDK 11

On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.

4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 3: Install the Linux VDA package

To set up the environment for the Linux VDA, run the following commands.

For RHEL and CentOS distributions:

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

For Ubuntu/Debian distributions:

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

> **Note:**
>
> To install the necessary dependencies for a Debian distribution, add the `deb http://deb.debian.org/debian/ oldstable main` line to the /etc/apt/sources.list file.

For SUSE distributions:

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

Step 4: Set up the runtime environment to complete the installation

> **Note:**
>
> Before setting up the runtime environment, ensure that the `en_US.UTF-8` locale has been installed in your OS. If the locale is not available in your OS, run the `sudo locale-gen en_US.UTF-8` command. For Debian, edit the `/etc/locale.gen` file by uncommenting the `# en_US.UTF-8 UTF-8` line and then run the `sudo locale-gen` command.

After installing the Linux VDA package, configure the running environment by using the ctxinstall.sh script. You can run the script in interactive mode or silent mode.

> **Note:**
>
> Easy install might seem unresponsive while it downloads .NET Core Runtime that is over 27 MB in size. For the downloading progress, check /var/log/ctxinstall.log.

Interactive mode:

To do a manual configuration, run the following command and type the relevant parameter at each prompt.

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

Silent mode:

To use easy install in silent mode, set the following environment variables before running ctxinstall.sh.

- **CTX_EASYINSTALL_HOSTNAME=host-name** –Denotes the host name of the Linux VDA server.
- **CTX_EASYINSTALL_DNS=ip-address-of-dns** –IP address of DNS.
- **CTX_EASYINSTALL_NTPS=address-of-ntps** –IP address or string name of the NTP server.
- **CTX_EASYINSTALL_DOMAIN=domain-name** –The NetBIOS name of the domain.
- **CTX_EASYINSTALL_REALM=realm-name** –The Kerberos realm name.
- **CTX_EASYINSTALL_FQDN=ad-fqdn-name**
- **CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind | sssd | centrify | pbis** –Denotes the Active Directory integration method.
- **CTX_EASYINSTALL_USERNAME=domain-user-name** –Denotes the name of the domain user; used to join the domain.
- **CTX_EASYINSTALL_PASSWORD=password** –Specifies the password of the domain user; used to join the domain.

The ctxsetup.sh script uses the following variables:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N** –The Linux VDA supports specifying a Delivery Controller name using a DNS CNAME record.
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'**–The Linux VDA requires a space-separated list of Delivery Controller Fully Qualified Domain Names (FQDNs) to use for registering with a Delivery Controller. At least one FQDN or CNAME must be specified.
- **CTX_XDL_VDA_PORT=port-number** –The Linux VDA communicates with Delivery Controllers through a TCP/IP port.
- **CTX_XDL_REGISTER_SERVICE=Y | N** –The Linux Virtual Desktop services are started after machine startup.
- **CTX_XDL_ADD_FIREWALL_RULES=Y | N** –The Linux Virtual Desktop services require incoming network connections to be allowed through the system firewall. You can open the required ports (by default ports 80 and 1494) automatically in the system firewall for the Linux Virtual Desktop.

- **CTX_XDL_HDX_3D_PRO=Y | N** –The Linux VDA supports HDX 3D Pro, a set of GPU acceleration technologies designed to optimize the virtualization of rich graphics applications. If HDX 3D Pro is selected, the VDA is configured for VDI desktops (single-session) mode - (that is, CTX_XDL_VDI_MODE=Y).
- **CTX_XDL_VDI_MODE=Y | N** –Whether to configure the machine as a dedicated desktop delivery model (VDI) or hosted shared desktop delivery model. For HDX 3D Pro environments, set the value to Y.
- **CTX_XDL_SITE_NAME=dns-name** –The Linux VDA discovers LDAP servers through DNS. To limit the DNS search results to a local Site, specify a DNS Site name. If unnecessary, set to **<none>**.
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** –The Linux VDA queries DNS to discover LDAP servers. If DNS cannot provide LDAP service records, you can provide a space-separated list of LDAP FQDNs with LDAP ports. For example, ad1.mycompany.com:389. If unnecessary, set to **<none>**.
- **CTX_XDL_SEARCH_BASE=search-base-set** –The Linux VDA queries LDAP through a search base set to the root of the Active Directory Domain (for example, DC=mycompany,DC=com). To improve search performance, you can specify a search base (for example, OU=VDI,DC=mycompany,DC=com). If unnecessary, set to **<none>**.
- **CTX_XDL_FAS_LIST='list-fas-servers'**–The Federated Authentication Service (FAS) servers are configured through AD Group Policy. The Linux VDA does not support AD Group Policy, but you can provide a semicolon-separated list of FAS servers instead. The sequence must be the same as configured in AD Group Policy. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.
- **CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime** –The path to install .NET Core Runtime 3.1 for supporting the new broker agent service (`ctxvda`). The default path is /usr/bin.
- **CTX_XDL_DESKTOP_ENVIRONMENT=gnome/mate** –Specifies the GNOME or MATE desktop environment to use in sessions. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set the variable value to **mate**.

  > **Note:**
  >
  > You can also change the desktop environment for a target session user by completing the following steps:
  >
  > 1. Create a `.xsession` file under the **$HOME/<username>** directory on the VDA.
  > 2. Edit the `.xsession` file to specify a desktop environment based on distributions.

For MATE desktop on CentOS, Ubuntu, and Debian

MSESSION="$(type -p mate-session)"
if [-n "$MSESSION"]; then
exec mate-session
fi

For GNOME desktop on CentOS

GSESSION="$(type -p gnome-session)"
if [-n "$GSESSION"]; then

```
    export GNOME_SHELL_SESSION_MODE=classic
    exec gnome-session --session=gnome-classic      fi
         **For GNOME desktop on Ubuntu and Debian**
```

GSESSION="$(type -p gnome-session)"
if [-n "$GSESSION"]; then

```
    exec gnome-session      fi
```

3. Share the 700 file permission with the target session user.

- **CTX_XDL_START_SERVICE=Y | N** –Whether or not the Linux VDA services are started when the configuration is complete.
- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The default port is 7503.
- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

If any parameters are not set, the installation rolls back to interactive mode, with a prompt for user input. When all parameters are already set through the environment variables, the ctxinstall.sh script still prompts for user input for the path to install .NET Core Runtime 3.1.

In silent mode, you must run the following commands to set environment variables and then run the ctxinstall.sh script.

```
export CTX_EASYINSTALL_HOSTNAME=host-name

export CTX_EASYINSTALL_DNS=ip-address-of-dns

export CTX_EASYINSTALL_NTPS=address-of-ntps

export CTX_EASYINSTALL_DOMAIN=domain-name

export CTX_EASYINSTALL_REALM=realm-name
```



```
10
11 export CTX_EASYINSTALL_FQDN=ad-fqdn-name
12
13 export CTX_EASYINSTALL_ADINTEGRATIONWAY=winbind | sssd | centrify |
      pbis
14
15 export CTX_EASYINSTALL_USERNAME=domain-user-name
16
17 export CTX_EASYINSTALL_PASSWORD=password
18
19 export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N
20
21 export CTX_XDL_DDC_LIST= ‘ list-ddc-fqdns ’
22
23 export CTX_XDL_VDA_PORT=port-number
24
25 export CTX_XDL_REGISTER_SERVICE=Y | N
26
27 export CTX_XDL_ADD_FIREWALL_RULES=Y | N
28
29 export CTX_XDL_HDX_3D_PRO=Y | N
30
31 export CTX_XDL_VDI_MODE=Y | N
32
33 export CTX_XDL_SITE_NAME=dns-site-name | '<none>'
34
35 export CTX_XDL_LDAP_LIST= ‘ list-ldap-servers ’  | '<none>'
36
37 export CTX_XDL_SEARCH_BASE=search-base-set | '<none>'
38
39 export CTX_XDL_FAS_LIST= ‘ list-fas-servers ’  | '<none>'
40
41 export CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime
42
43 export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'
44
45 export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number
46
47 export CTX_XDL_TELEMETRY_PORT=port-number
48
49 export CTX_XDL_START_SERVICE=Y | N
50
51 sudo -E /opt/Citrix/VDA/sbin/ctxinstall.sh
52 <!--NeedCopy-->
```

When running the sudo command, type the -E option to pass the existing environment variables to the new shell it creates. We recommend that you create a shell script file from the preceding commands with **#!/bin/bash** as the first line.

Alternatively, you can specify all parameters by using a single command:

```
1 sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \
2
```



```
CTX_XDL_DDC_LIST= ‘ list-ddc-fqdns ’  \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST= ‘ list-ldap-servers ’  \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST= ‘ list-fas-servers ’  \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

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

Troubleshooting

Use the information in this section to troubleshoot issues that can arise from using this feature.

Joining a domain by using SSSD fails

An error might occur when you attempt to join a domain, with the output similar to the following (verify logs for screen printing):

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

/var/log/xdl/vda.log:


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

/var/log/messages:

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

To resolve this issue:

1. Run the `rm -f /etc/krb5.keytab` command.
2. Run the `net ads leave $REALM -U $domain-administrator` command.
3. Remove the machine catalog and delivery group on the Delivery Controller.
4. Run /opt/Citrix/VDA/sbin/ctxinstall.sh.

5. Create the machine catalog and delivery group on the Delivery Controller.

Ubuntu desktop sessions show a gray screen

This issue occurs when you launch a session that is then blocked in a blank desktop. In addition, the console of the machine also shows a gray screen when you log on by using a local user account.

To resolve this issue:

1. Run the `sudo apt-get update` command.
2. Run the `sudo apt-get install unity lightdm` command.
3. Add the following line to `/etc/lightdm/lightdm.conf`:
   `greeter-show-manual-login=true`

Attempts to launch Ubuntu desktop sessions fail due to a missing home directory

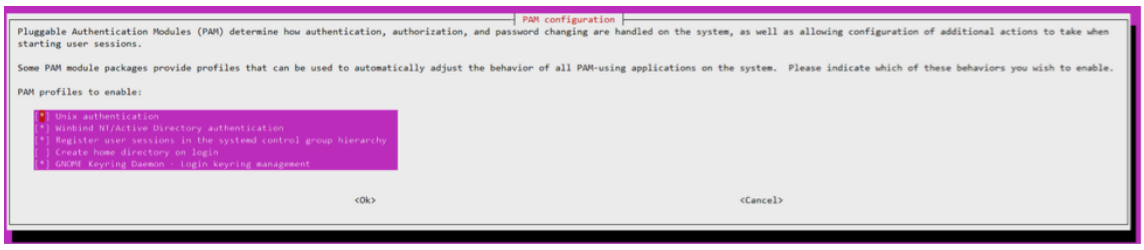
/var/log/xdl/hdx.log:

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

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

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

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

> **Tip:**
>
> The root cause of this issue is that the home directory is not created for the domain administrator.

To resolve this issue:

1. From a command line, type **pam-auth-update**.
2. In the resulting dialog, verify that **Create home directory login** is selected.



Session does not launch or ends quickly with dbus error

/var/log/messages (for RHEL or CentOS):

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

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

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

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

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

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

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

Or, alternately for Ubuntu distributions, use the log /var/log/syslog:

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

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

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

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

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



```
      blocked the reply, the reply timeout expired, or the network
      connection was broken.]
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
12 <!--NeedCopy-->
```

Some groups or modules do not take effect until a restart. If the **dbus** error messages appear in the log, we recommend that you restart the system and retry.

SELinux prevents SSHD from accessing the home directory

The user can launch a session but cannot log on.

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

To resolve this issue:

1. Disable SELinux by making the following change to /etc/selinux/config.

   SELINUX=disabled

2. Restart the VDA.

Create Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure

May 16, 2023

You can create both domain-joined and non-domain-joined Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure to deliver virtual apps and desktops to any device from Microsoft Azure. For more information, see Citrix Virtual Apps and Desktops Standard for Azure.

Supported Linux distributions

The following Linux distributions support this feature:

- RHEL 8.3
- RHEL 8.2
- RHEL 7.8
- Ubuntu 20.04
- Ubuntu 18.04
- Ubuntu 16.04

Steps

To create Linux VDAs in Citrix Virtual Apps and Desktops Standard for Azure, complete the following steps:

1. Prepare a master image in Azure:

   a) In Azure, create a Linux VM of a supported distribution.

b) Install a desktop environment on the Linux VM if necessary.

c) On the VM, install .NET Core Runtime 3.1 according to the instructions at https://docs.microsoft.com/en-us/dotnet/core/install/linux-package-managers.

d) (For Ubuntu only) Add the `source /etc/network/interfaces.d/*` line to the `/etc/network/interfaces` file.

e) (For Ubuntu only) Point `/etc/resolv.conf` to `/run/systemd/resolve/resolv.conf` instead of pointing it to `/run/systemd/resolve/stub-resolv.conf`:

```
unlink /etc/resolv.conf

ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
<!--NeedCopy-->
```

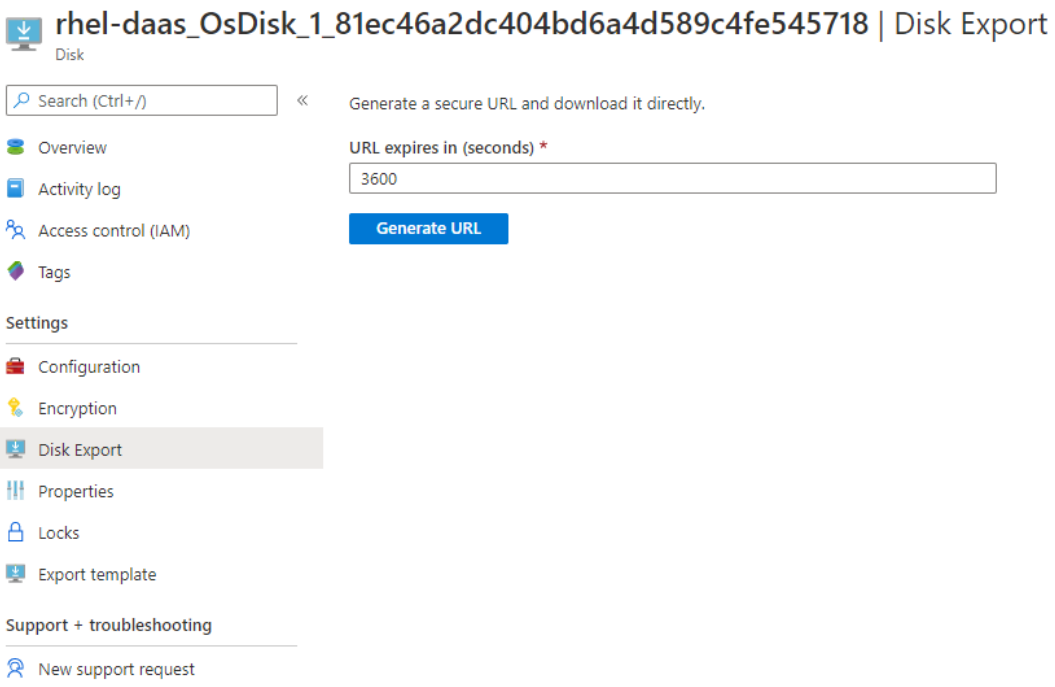
f) Install the Linux VDA package.

g) Change variables in /etc/xdl/mcs/mcs.conf. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA.

> **Note:**
>
> Leave the dns variable unspecified.
>
> If you select the **Static** or **Random** type when creating a machine catalog, set `VDI_MODE=Y`.

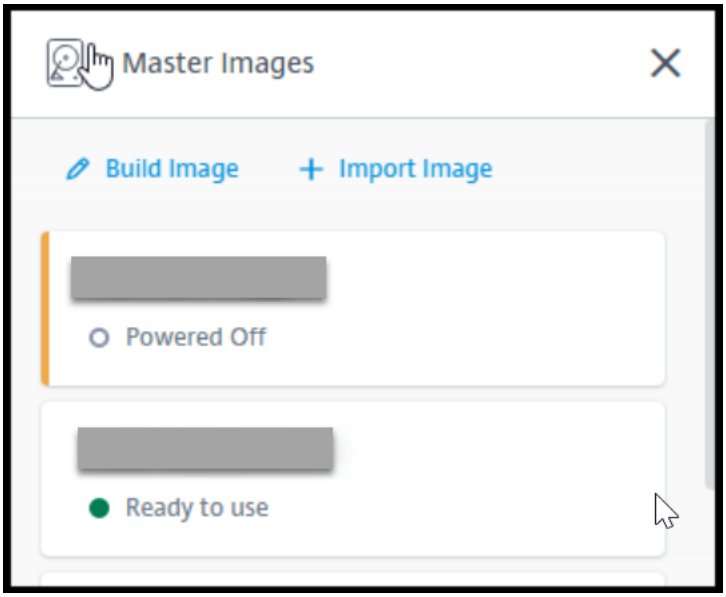
h) Run `/opt/Citrix/VDA/sbin/deploymcs.sh`.

i) In Azure, stop (or deallocate) the VM. Click **Disk Export** to generate a SAS URL for the Virtual Hard Disk (VHD) file that you can use as a master image to create other VMs.



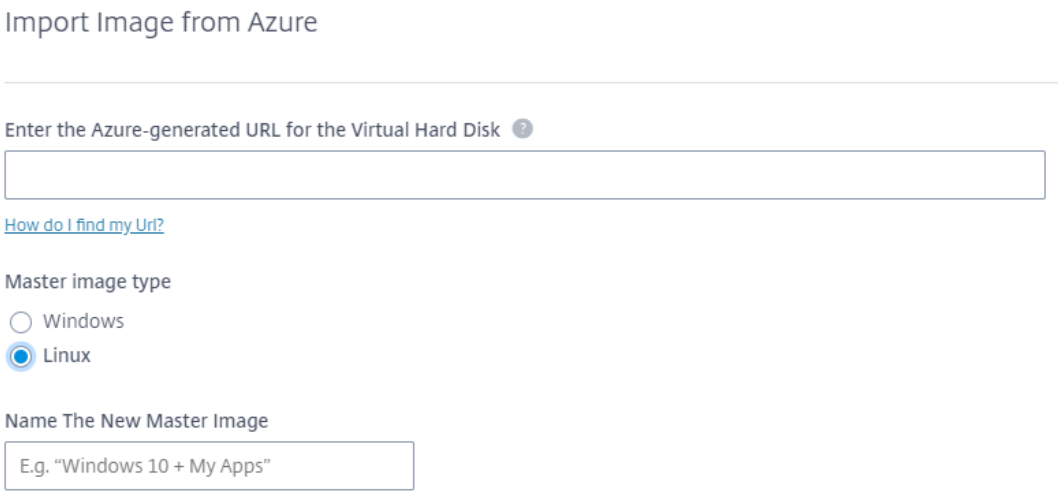
j) (Optional) Make group policy settings on the master image.

```
You can use the `ctxreg` tool to make group policy settings. For example, the following command enables the **Auto-create PDF Universal Printer** policy for PDF printing.

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

2. Import the master image from Azure.

a) From the Manage dashboard, expand **Master Images** on the right. The display lists the master images that Citrix provides, and images that you created and imported.

Tip: Most of the administrator activities for this service are managed through the **Manage** and **Monitor** dashboards. After you create your first catalog, the **Manage** dashboard launches automatically after you sign in to Citrix Cloud and select the **Managed Desktops** service.


```




b) Click **Import Image**.

c) Enter the SAS URL for the VHD file you generated in Azure. Select **Linux** for the master image type.




d) Follow the instructions in the wizard to complete importing the master image.

3. Create a Machine Catalog.

   Access the Manage dashboard and click **Create Catalog**. When creating the Machine Catalog, choose the master image you created earlier.

   **Note:** You can create non-domain-joined Linux Machine Catalogs in Citrix-managed Azure subscription only.

# Use Machine Creation Services (MCS) to create Linux VMs

April 11, 2023

To use MCS to create Linux VMs, prepare a master image on your hypervisor. This process entails installing the VDA on the template VM, creating a Machine Catalog in Citrix Studio, creating a Delivery Group, and performing certain configuration tasks.

> **Note:**
>
> Unexpected results can occur if you try to prepare a master image on hypervisors other than Citrix Hypervisor, Microsoft Azure, VMware vSphere, AWS, GCP, or Nutanix AHV.
>
> Microsoft Azure, AWS, and GCP are not supported as of Citrix Virtual Apps and Desktops 7 2003. But you can continue using the hosts in the Citrix Virtual Apps and Desktops service.

## Supported distributions

| | Winbind | SSSD | Centrify | PBIS |
|---|---|---|---|---|
| RHEL 8.3 | Yes | No | No | No |
| CentOS 8.3 | Yes | No | No | No |
| RHEL 8.2 | Yes | No | No | No |
| CentOS 8.2 | Yes | No | No | No |
| RHEL 8.1 | Yes | No | No | No |
| CentOS 8.1 | Yes | No | No | No |
| RHEL 7.9 | Yes | Yes | No | No |
| CentOS 7.9 | Yes | Yes | No | No |
| RHEL 7.8 | Yes | Yes | No | No |
| CentOS 7.8 | Yes | Yes | No | No |
| Ubuntu 20.04 | Yes | Yes | No | No |
| Ubuntu 18.04 | Yes | Yes | No | No |
| Ubuntu 16.04 | Yes | Yes | No | No |
| Debian 10.7 | Yes | Yes | No | No |
| SUSE 12.5 | Yes | Yes | No | No |

### Use MCS to create Linux VMs on Citrix Hypervisor

#### Step 1: Prepare a master image

A master image contains the operating system, non-virtualized applications, VDA, and other software. To prepare a master image, do the following:

**Step 1a: Install Citrix VM Tools** Citrix VM Tools must be installed on the template VM for each VM to be able to use the xe CLI or XenCenter. VM performance can be slow unless the tools are installed. Without the tools, you cannot do any of the following:

- Cleanly shut down, restart, or suspend a VM.
- View the VM performance data in XenCenter.
- Migrate a running VM (through XenMotion).
- Create snapshots or snapshots with memory (checkpoints), and revert to snapshots.
- Adjust the number of vCPUs on a running Linux VM.

1. Run the following command to mount Citrix VM Tools named guest-tools.iso.

```
sudo mount /dev/cdrom /mnt
<!--NeedCopy-->
```

2. Run the following command to install the xe-guest-utilities package based on your Linux distribution.

   For RHEL/CentOS:

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

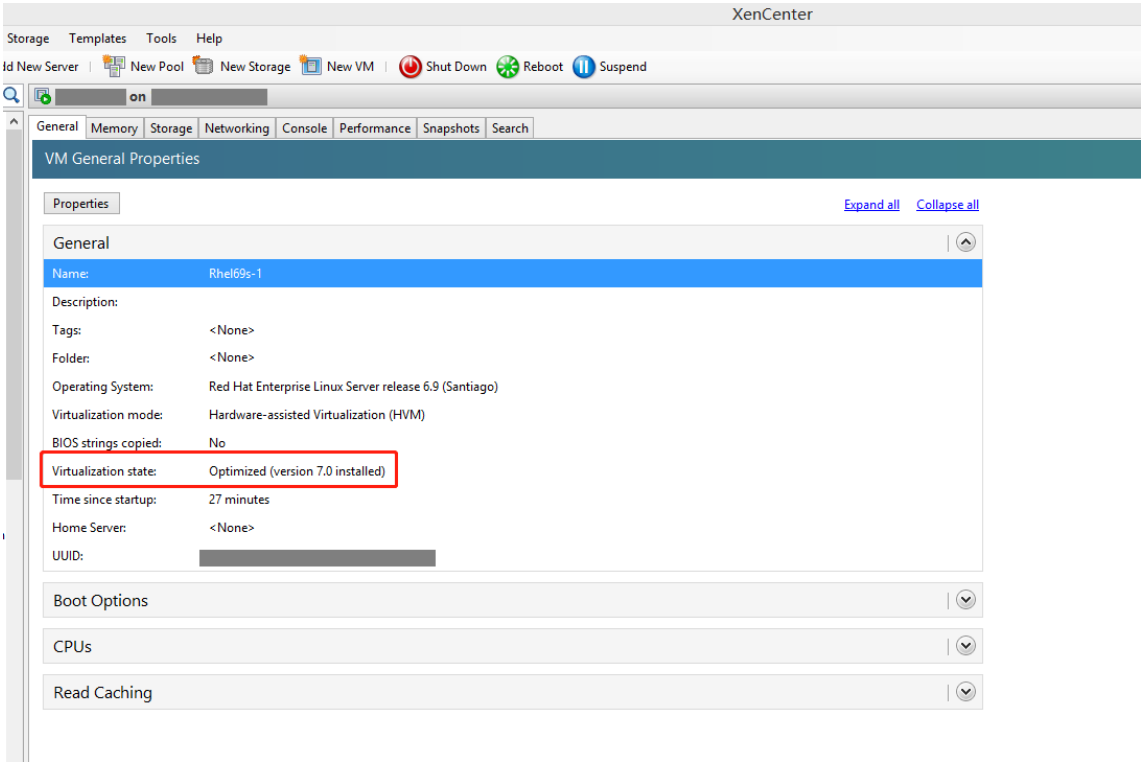
   For Ubuntu/Debian:

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

   For SUSE 12:

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

3. Check the virtualization state of the template VM on the **General** tab in XenCenter. If Citrix VM Tools are installed correctly, the virtualization state is **Optimized**:



Step 1b: (For Ubuntu 16.04 only) Install OpenJDK 11 On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 1c: Install the Linux VDA package on the template VM

> **Note:**
>
> To use a currently running VDA as the template VM, omit this step.
>
> Before installing the Linux VDA package on the template VM, install .NET Core Runtime 3.1. For more information, see Installation overview.

Based on your Linux distribution, run the following command to set up the environment for the Linux VDA:

For RHEL/CentOS:

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

For Ubuntu/Debian:

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

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

For SUSE 12:

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

Step 1d: Enable repositories to install the tdb-tools package **For RHEL 7 server:**

```
subscription-manager repos --enable=rhel-7-server-optional-rpms
<!--NeedCopy-->
```

For RHEL 7 workstation:

```
subscription-manager repos --enable=rhel-7-workstation-optional-rpms
<!--NeedCopy-->
```

Step 1e: Install the EPEL repository that contains ntfs-3g Install the EPEL repository on RHEL 8/CentOS 8, RHEL 7/CentOS 7 so that running deploymcs.sh later installs the ntfs-3g package contained in it.

Step 1f: Manually install ntfs-3g on SUSE 12 On the SUSE 12 platform, there is no repository providing ntfs-3g. Download the source code, compile, and install ntfs-3g manually:

1. Install the GNU Compiler Collection (GCC) compiler system and the make package:

   ```
   sudo zypper install gcc
   sudo zypper install make
   <!--NeedCopy-->
   ```

2. Download the ntfs-3g package.
3. Decompress the ntfs-3g package:

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

4. Enter the path to the ntfs-3g package:


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

5. Install ntfs-3g:

```
./configure
make
make install
<!--NeedCopy-->
```

Step 1g: Set up the runtime environment Before running `deploymcs.sh`, do the following:

- Change variables in `/etc/xdl/mcs/mcs.conf`. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA. The following are variables you can set as required:
  - `Use_Existing_Configurations_Of_Current_VDA`: Determines whether to use the existing configurations of the currently running VDA. If set to Y, configuration files on MCS-created machines are the same as the equivalents on the currently running VDA. However, you still must configure the `dns` and `AD_INTEGRATION` variables. The default value is N, which means configuration files on MCS-created machines are determined by configuration templates on the master image.
  - `dns`: Sets the DNS IP address.
  - `AD_INTEGRATION`: Sets Winbind or SSSD. For a matrix of the Linux distributions and domain joining methods that MSC supports, see Supported distributions in this article.
  - `WORKGROUP`: Sets the workgroup name (case-sensitive) if it is configured in AD.
- On the template machine, add command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file for writing or updating registry values as required. This action prevents the loss of data and settings every time an MCS-provisioned machine restarts.

  Each line in the `/etc/xdl/mcs/mcs_local_setting.reg` file is a command for setting or updating a registry value.

  For example, you can add the following command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file to write or update a registry value respectively:

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

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


Step 1h: Create a master image

1. Run `/opt/Citrix/VDA/sbin/deploymcs.sh`.
2. (Optional) On the template VM, update the configuration templates to customize the relevant `/etc/krb5.conf`, `/etc/samba/smb.conf`, and `/etc/sssd/sssd.conf` files on all created VMs.

   For Winbind users, update the `/etc/xdl/mcs/winbind_krb5.conf.tmpl` and `/etc/xdl/mcs/winbind_smb.conf.tmpl` templates.

   For SSSD users, update the `/etc/xdl/mcs/sssd.conf.tmpl`, `/etc/xdl/mcs/sssd_krb5.conf.tmpl`, and `/etc/xdl/mcs/sssd_smb.conf.tmpl` templates.

   > **Note:**
   >
   > Keep the existing format used in the template files and use variables such as $WORKGROUP, $REALM, $realm, and $AD_FQDN.

3. On Citrix Hypervisor, shut down the template VM. Create and name a snapshot of your master image.

Step 2: Create a Machine Catalog

In Citrix Studio, create a Machine Catalog and specify the number of VMs to create in the catalog. Do other configuration tasks as needed. For more information, see Create a machine catalog using Studio.

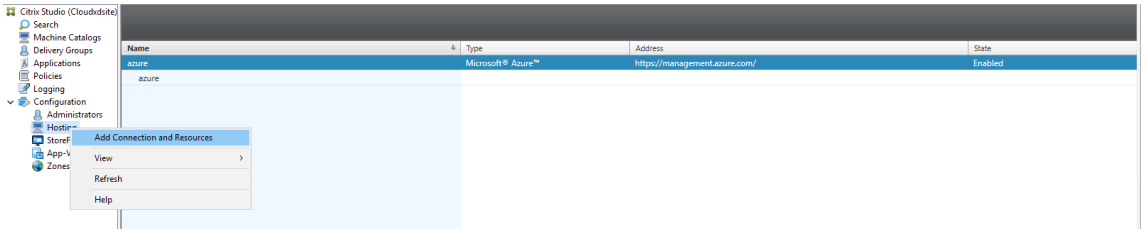
Step 3: Create a Delivery Group

A Delivery Group is a collection of machines selected from one or more Machine Catalogs. The Delivery Group specifies which users can use those machines, and the applications and desktops available to those users. For more information, see Create Delivery Groups.

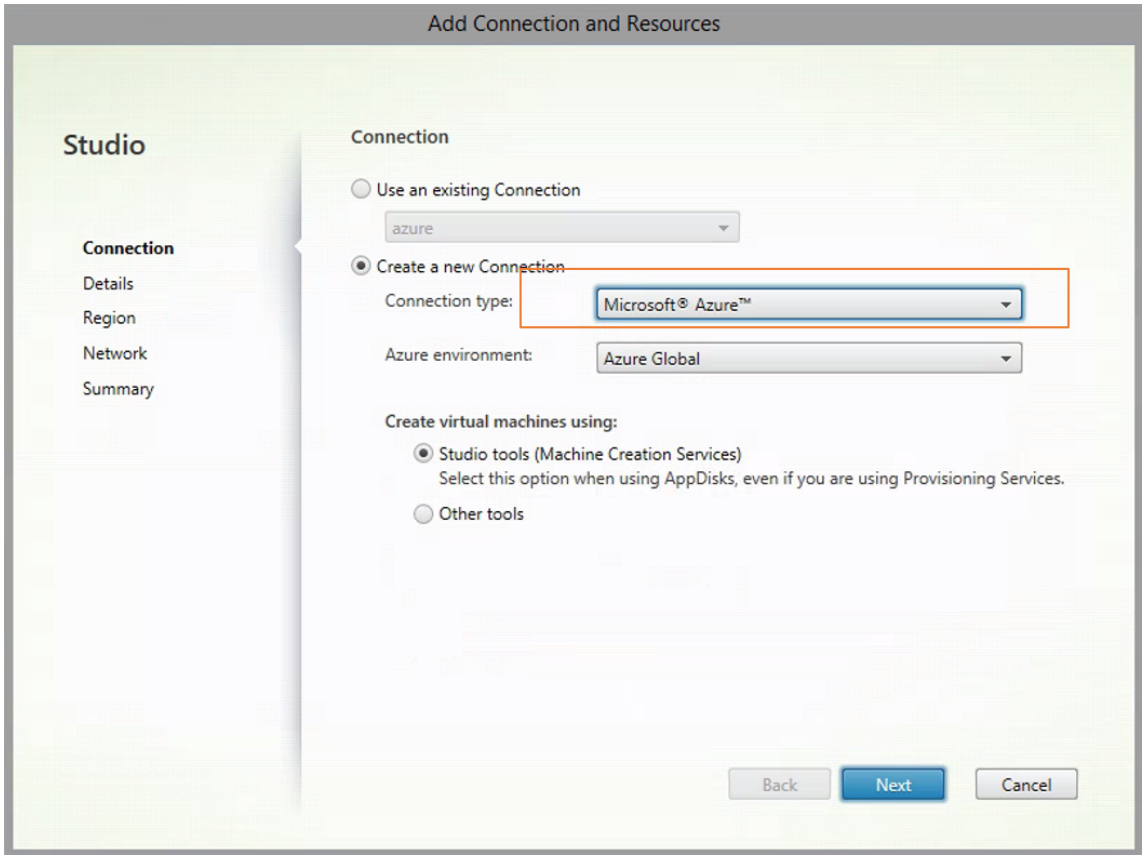
Use MCS to create Linux VMs on Azure

Step 1: Create a hosting connection to Azure in Citrix Studio

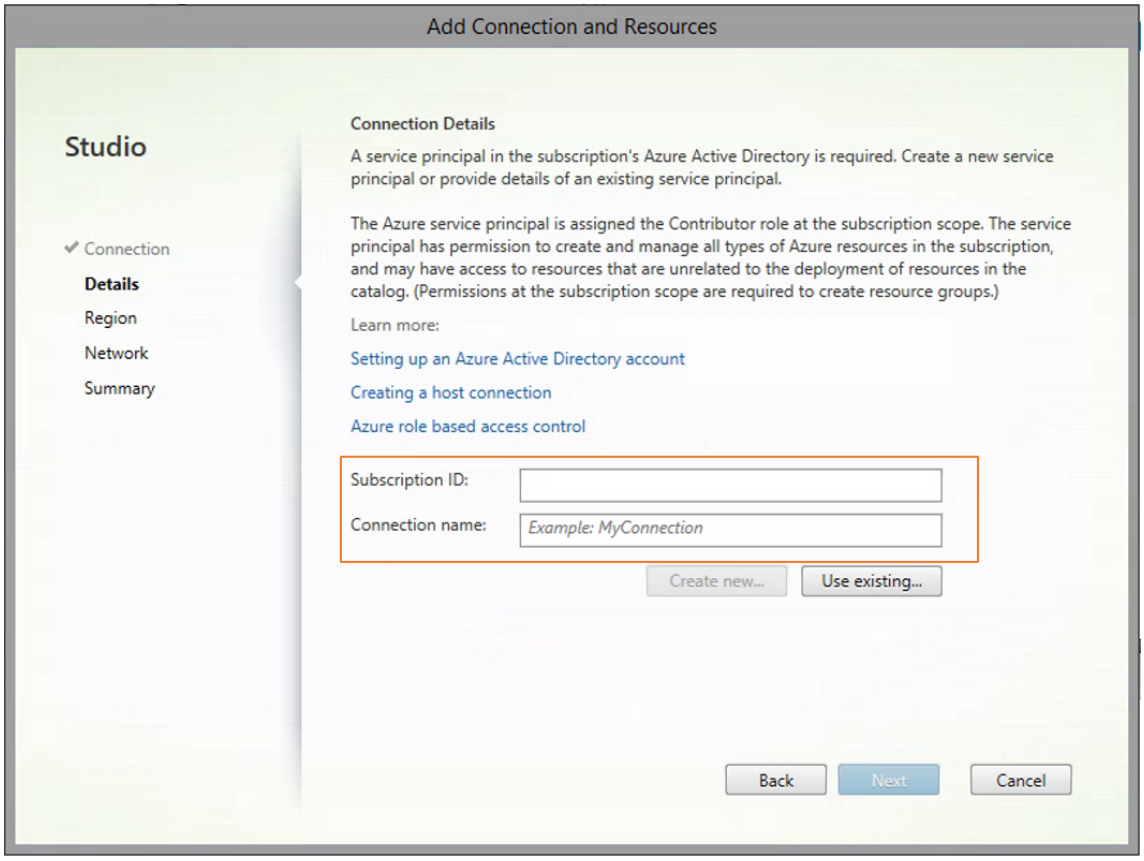
1. In Citrix Studio on Citrix Cloud, choose **Configuration > Hosting > Add Connection and Resources** to create a connection to Azure.



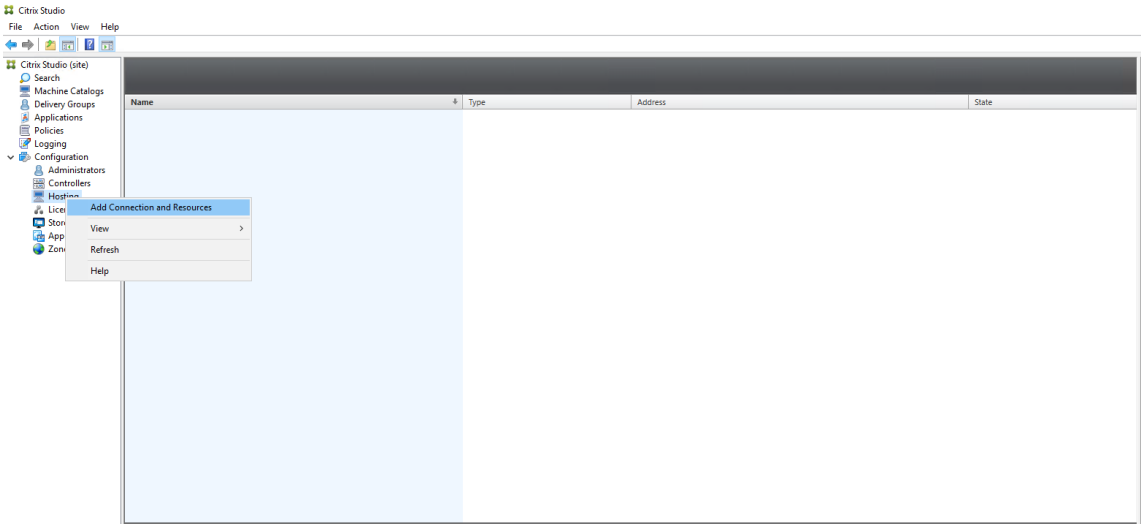
2. Choose Microsoft Azure as the connection type.



3. Type the subscription ID of your Azure account and your connection name.



A new connection appears in the hosting pane.



Step 2: Prepare a master image on the template VM

A master image contains the operating system, non-virtualized applications, VDA, and other software. To prepare a master image, do the following:

Step 2a: Configure cloud-init for Ubuntu 18.04 To ensure that a VDA host name persists when a VM is restarted or stopped, run the following command.

```
1  echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.
       cfg
2  <!--NeedCopy-->
```

Ensure that the following lines are present under the **system_info** section in the /etc/cloud/cloud.cfg file:

```
1  system_info:
2     network:
3         renderers: ['netplan', 'eni', 'sysconfig']
4  <!--NeedCopy-->
```

Step 2b: (For Ubuntu 16.04 only) Install OpenJDK 11 On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 2c: Install the Linux VDA package on the template VM

> **Note:**
>
> To use a currently running VDA as the template VM, omit this step.
>
> Before installing the Linux VDA package on the template VM, install .NET Core Runtime 3.1. For more information, see Installation overview.

Based on your Linux distribution, run the following command to set up the environment for the Linux VDA:

For RHEL/CentOS:

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

For Ubuntu/Debian:

```
1  sudo dpkg -i  <PATH>/<Linux VDA DEB>
```



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

For SUSE 12:

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

Step 2d: Install the EPEL repository that contains ntfs-3g Install the EPEL repository on RHEL 8/CentOS 8, RHEL 7/CentOS 7 so that running deploymcs.sh later installs the ntfs-3g package contained in it.

Step 2e: Manually install ntfs-3g on SUSE 12 On the SUSE 12 platform, there is no repository providing ntfs-3g. Download the source code, compile, and install ntfs-3g manually:

1. Install the GNU Compiler Collection (GCC) compiler system and the make package:

   ```
   sudo zypper install gcc
   sudo zypper install make
   <!--NeedCopy-->
   ```

2. Download the ntfs-3g package.
3. Decompress the ntfs-3g package:

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

4. Enter the path to the ntfs-3g package:

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

5. Install ntfs-3g:

   ```
   ./configure
   make
   make install
   <!--NeedCopy-->
   ```

Step 2f: Set up the runtime environment Before running `deploymcs.sh`, do the following:

- Change variables in `/etc/xdl/mcs/mcs.conf`. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA. The following are some of the variables, of which `dns` and `AD_INTEGRATION` must be set:

Note: If a variable can be set with multiple values, put the values inside single quotes and separate them with spaces. For example, LDAP_LIST='aaa.lab:389 bbb.lab:389.'

  - `Use_Existing_Configurations_Of_Current_VDA`: Determines whether to use the existing configurations of the currently running VDA. If set to Y, configuration files on MCS-created machines are the same as the equivalents on the currently running VDA. However, you still must configure the `dns` and `AD_INTEGRATION` variables. The default value is N, which means configuration files on MCS-created machines are determined by configuration templates on the master image.
  - `dns`: Sets the DNS IP address.
  - `AD_INTEGRATION`: Sets Winbind or SSSD (SSSD is not supported on SUSE).
  - `WORKGROUP`: Sets the workgroup name (case-sensitive) if it is configured in AD.

- On the template machine, add command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file for writing or updating registry values as required. This action prevents the loss of data and settings every time an MCS-provisioned machine restarts.

  Each line in the `/etc/xdl/mcs/mcs_local_setting.reg` file is a command for setting or updating a registry value.

  For example, you can add the following command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file to write or update a registry value respectively:

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

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

Step 2g: Create a master image

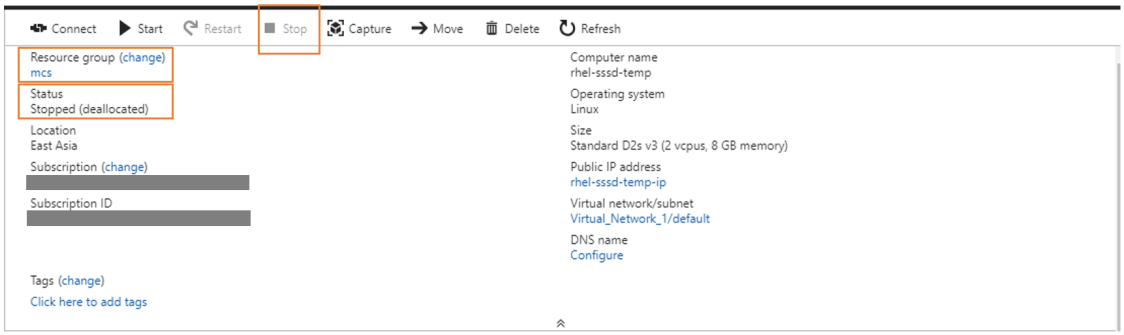
1. Run `/opt/Citrix/VDA/sbin/deploymcs.sh`.
2. (Optional) On the template VM, update the configuration templates to customize the relevant `/etc/krb5.conf`, `/etc/samba/smb.conf`, and `/etc/sssd/sssd.conf` files on all created VMs.

   For Winbind users, update the `/etc/xdl/mcs/winbind_krb5.conf.tmpl` and `/etc/xdl/mcs/winbind_smb.conf.tmpl` templates.

   For SSSD users, update the `/etc/xdl/mcs/sssd.conf.tmpl`, `/etc/xdl/mcs/sssd_krb5.conf.tmpl`, and `/etc/xdl/mcs/sssd_smb.conf.tmpl` templates.

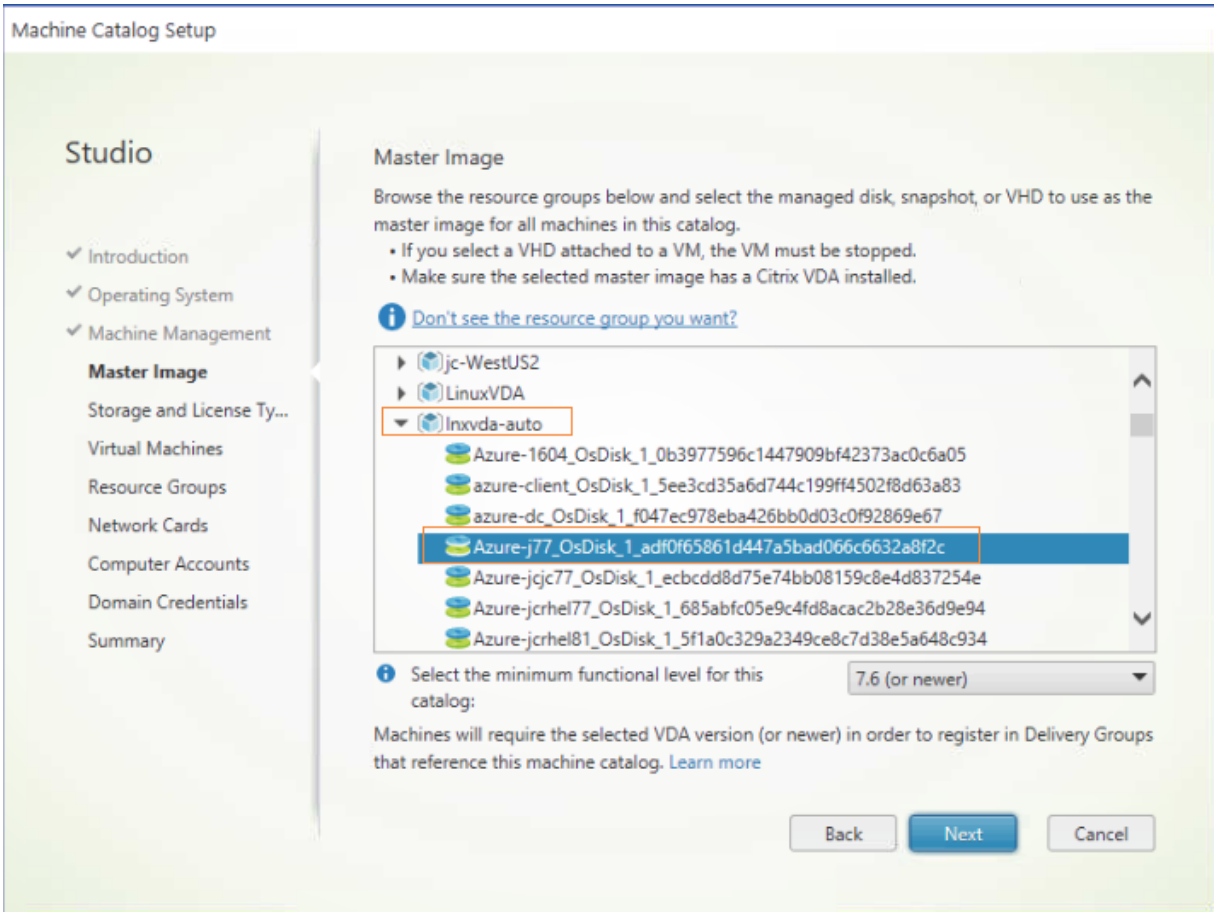
Note: Keep the existing format used in the template files and use variables such as $WORKGROUP, $REALM, $realm, and $AD_FQDN.

3. Install applications on the template VM and shut down the template VM from the Azure portal. Ensure that the power status of the template VM is **Stopped (deallocated)**. Remember the name of the resource group here. You need the name to locate your master image on Azure.



Step 3: Create a Machine Catalog

In Citrix Studio, create a Machine Catalog and specify the number of VMs to create in the catalog. When creating the Machine Catalog, choose your master image from the resource group where the template VM belongs and find the VHD of the template VM.



Do other configuration tasks as needed. For more information, see Create a machine catalog using Studio.

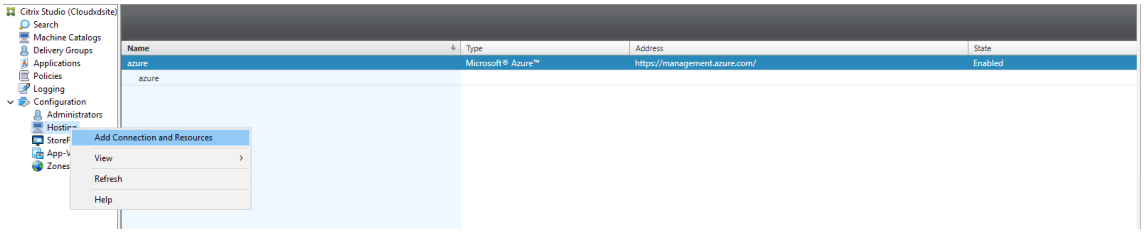
Step 4: Create a Delivery Group

A Delivery Group is a collection of machines selected from one or more Machine Catalogs. The Delivery Group specifies which users can use those machines, and the applications and desktops available to those users. For more information, see Create Delivery Groups.

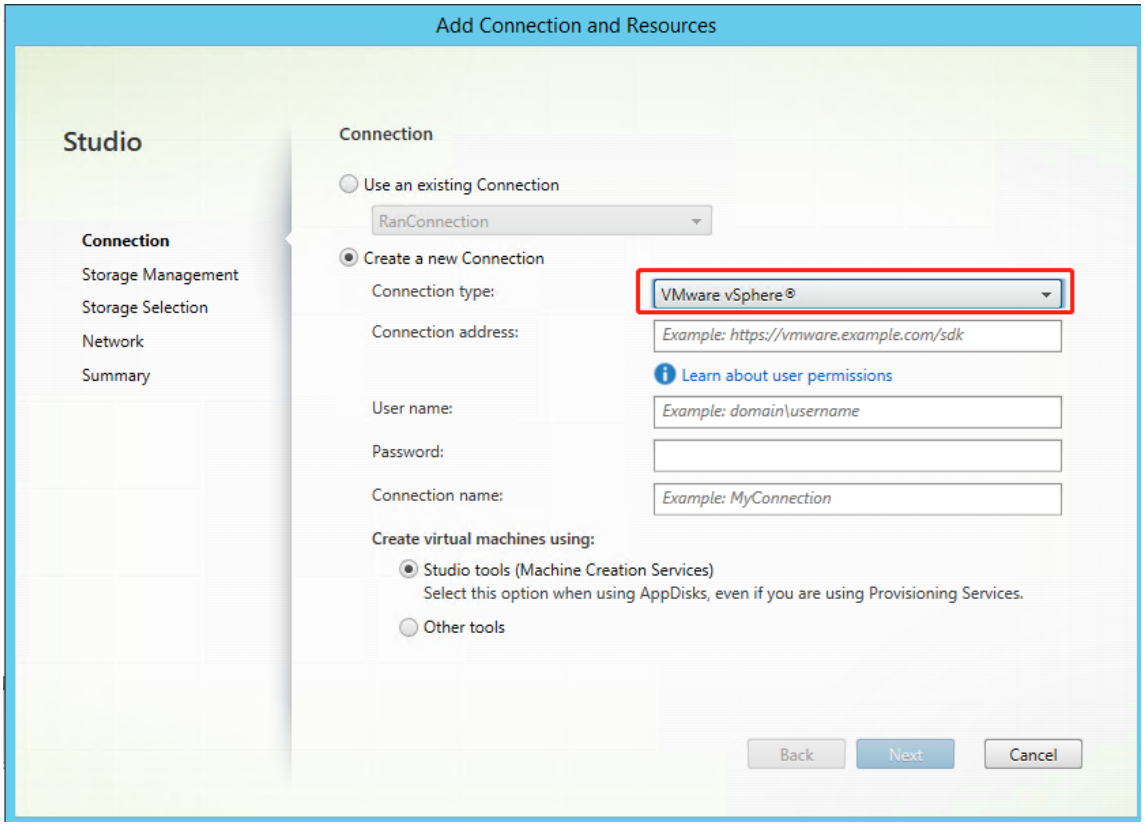
Use MCS to create Linux VMs on VMware vSphere

Step 1: Create a hosting connection to VMware in Citrix Studio

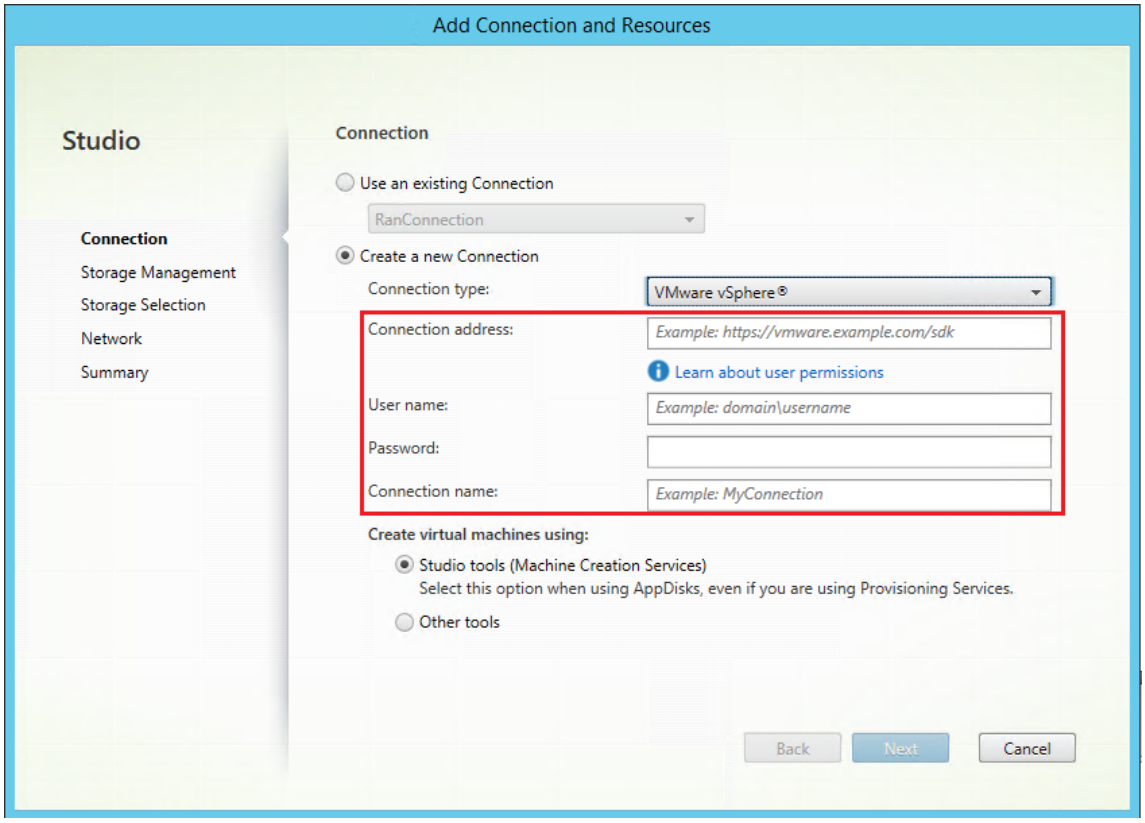
1. Install vCenter Server in the vSphere environment. For more information, see VMware vSphere.
2. In Citrix Studio, choose **Configuration > Hosting > Add Connection and Resources** to create a connection to VMware vSphere.



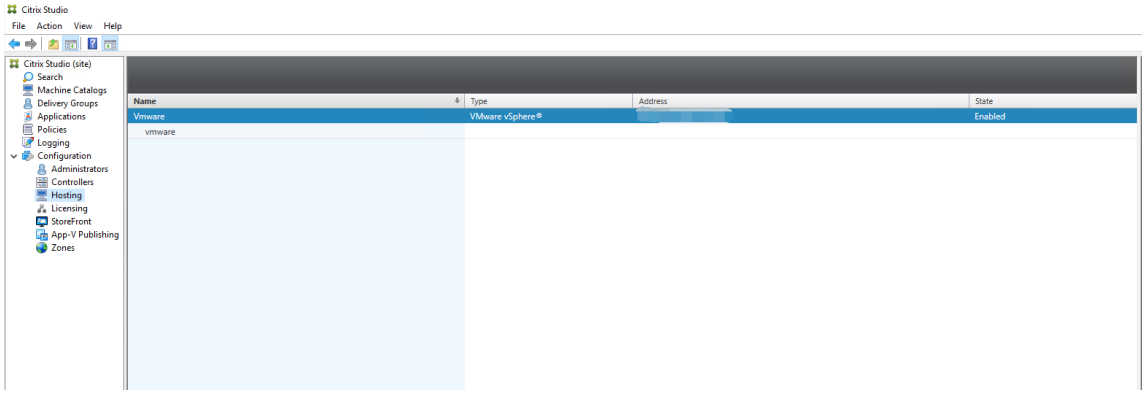
3. Choose VMware vSphere as the connection type.



4. Type the connection address (the vCenter Server URL) of your VMware account, your user name and password, and your connection name.



A new connection appears in the hosting pane.



Step 2: Prepare a master image

A master image contains the operating system, non-virtualized applications, VDA, and other software. To prepare a master image, do the following:

Step 2a: (For Ubuntu 16.04 only) Install OpenJDK 11 On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.

2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 2b: Install the Linux VDA package on the template VM

> **Note:**
>
> To use a currently running VDA as the template VM, omit this step.
>
> Before installing the Linux VDA package on the template VM, install .NET Core Runtime 3.1. For more information, see Installation overview.

Based on your Linux distribution, run the following command to set up the environment for the Linux VDA:

For RHEL/CentOS:

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

For Ubuntu/Debian:

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

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

For SUSE 12:

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

Step 2c: Install the EPEL repository that contains ntfs-3g Install the EPEL repository on RHEL 8/CentOS 8, RHEL 7/CentOS 7 so that running deploymcs.sh later installs the ntfs-3g package contained in it.

Step 2d: Manually install ntfs-3g on SUSE 12 On the SUSE 12 platform, there is no repository providing ntfs-3g. Download the source code, compile, and install ntfs-3g manually:

1. Install the GNU Compiler Collection (GCC) compiler system and the make package:


```
sudo zypper install gcc
sudo zypper install make
<!--NeedCopy-->
```

2. Download the ntfs-3g package.
3. Decompress the ntfs-3g package:

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

4. Enter the path to the ntfs-3g package:

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

5. Install ntfs-3g:

```
./configure
make
make install
<!--NeedCopy-->
```

Step 2e: Set up the runtime environment Before running `deploymcs.sh`, do the following:

- Change variables in `/etc/xdl/mcs/mcs.conf`. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA. The following are some of the variables, of which `dns` and `AD_INTEGRATION` must be set:

  Note: If a variable can be set with multiple values, put the values inside single quotes and separate them with spaces. For example, LDAP_LIST='aaa.lab:389 bbb.lab:389.'

  - `Use_Existing_Configurations_Of_Current_VDA`: Determines whether to use the existing configurations of the currently running VDA. If set to Y, the configuration files on MCS-created machines are the same as the equivalents on the currently running VDA. However, you still must configure the `dns` and `AD_INTEGRATION` variables. The default value is N, which means configuration files on MCS-created machines are determined by configuration templates on the master image.
  - `dns`: Sets the DNS IP address.
  - `AD_INTEGRATION`: Sets Winbind or SSSD (SSSD is not supported on SUSE).
  - `WORKGROUP`: Sets the workgroup name (case-sensitive) if it is configured in AD.

- On the template machine, add command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file for writing or updating registry values as required. This action prevents the loss of data and settings every time an MCS-provisioned machine restarts.

Each line in the /etc/xdl/mcs/mcs_local_setting.reg file is a command for setting or updating a registry value.

For example, you can add the following command lines to the /etc/xdl/mcs/mcs_local_setting.reg file to write or update a registry value respectively:

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

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

Step 2f: Create a master image

1. Run /opt/Citrix/VDA/sbin/deploymcs.sh.
2. (Optional) On the template VM, update the configuration templates to customize the relevant /etc/krb5.conf, /etc/samba/smb.conf, and /etc/sssd/sssd.conf files on all created VMs.

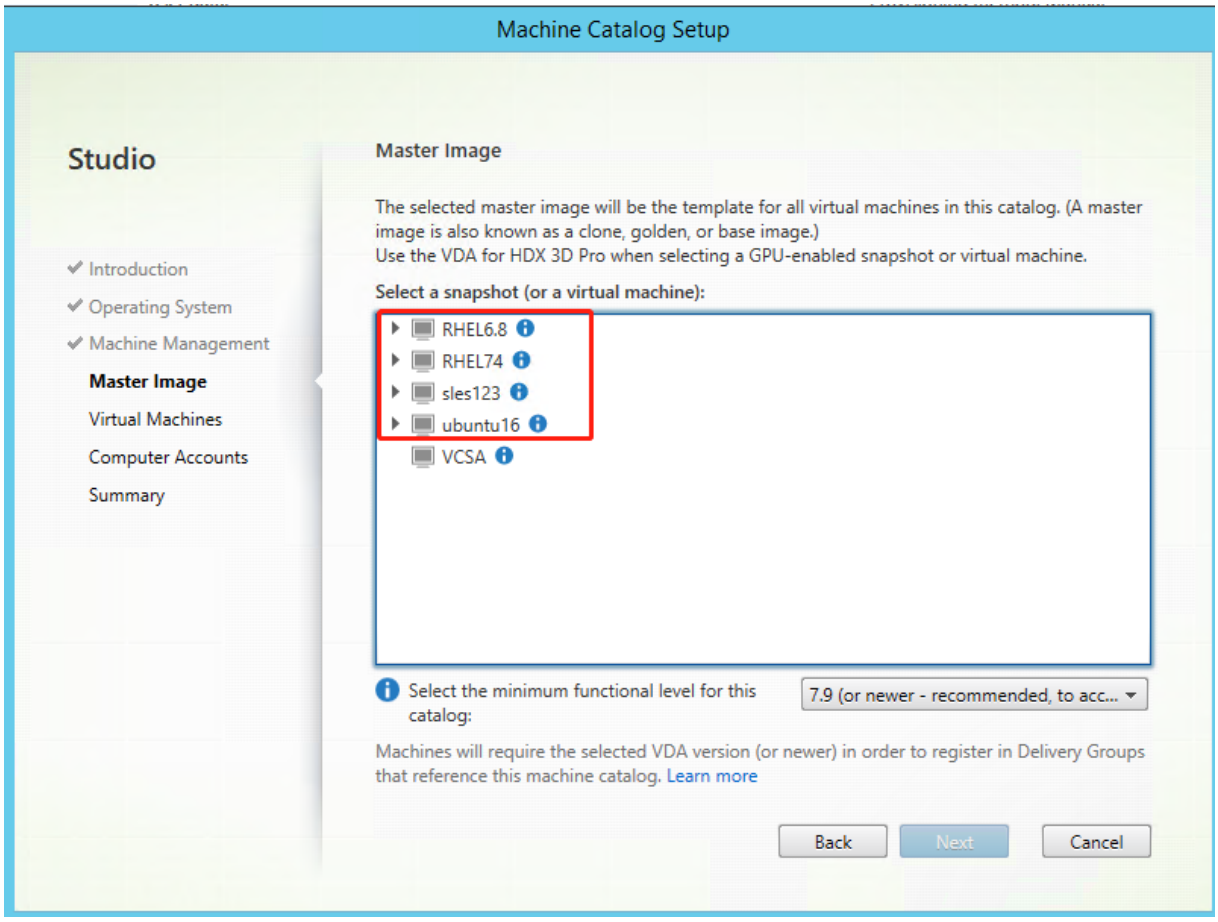
   For Winbind users, update the /etc/xdl/mcs/winbind_krb5.conf.tmpl and /etc/xdl/mcs/winbind_smb.conf.tmpl templates.

   For SSSD users, update the /etc/xdl/mcs/sssd.conf.tmpl, /etc/xdl/mcs/sssd_krb5.conf.tmpl, and /etc/xdl/mcs/sssd_smb.conf.tmpl templates.

   Note: Keep the existing format used in the template files and use variables such as $WORKGROUP, $REALM, $realm, and $AD_FQDN.
3. After you finish installing applications on the template VM, shut down the template VM from the VMware. Take a snapshot of the template VM.

Step 3: Create a Machine Catalog

In Citrix Studio, create a Machine Catalog and specify the number of VMs to create in the catalog. When creating the Machine Catalog, choose your master image from the snapshot list.



Do other configuration tasks as needed. For more information, see Create a machine catalog using Studio.

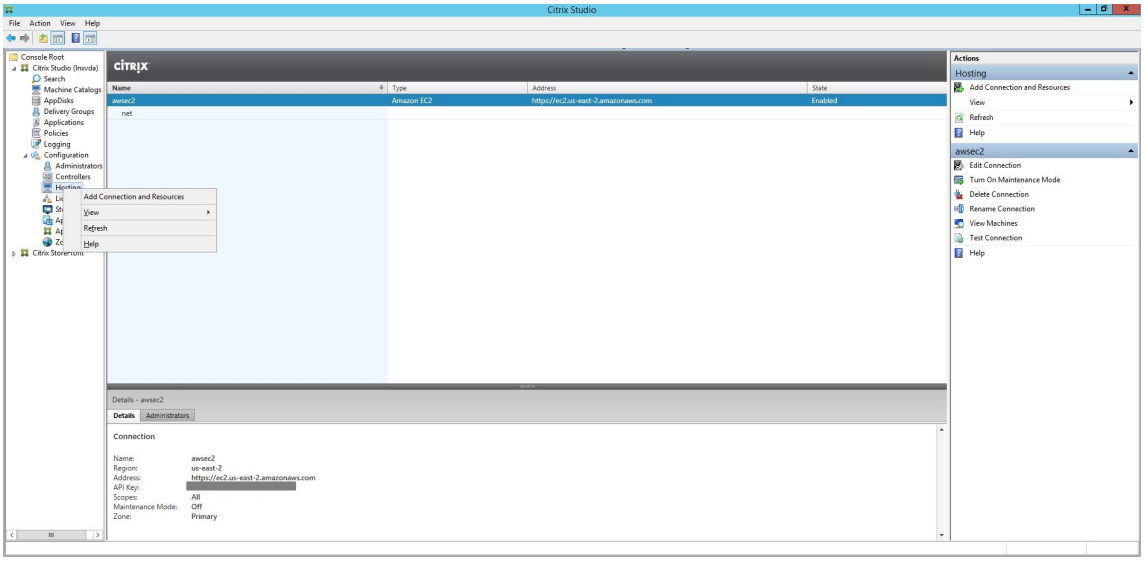
Step 4: Create a Delivery Group

A Delivery Group is a collection of machines selected from one or more Machine Catalogs. The Delivery Group specifies which users can use those machines, and the applications and desktops available to those users. For more information, see Create Delivery Groups.

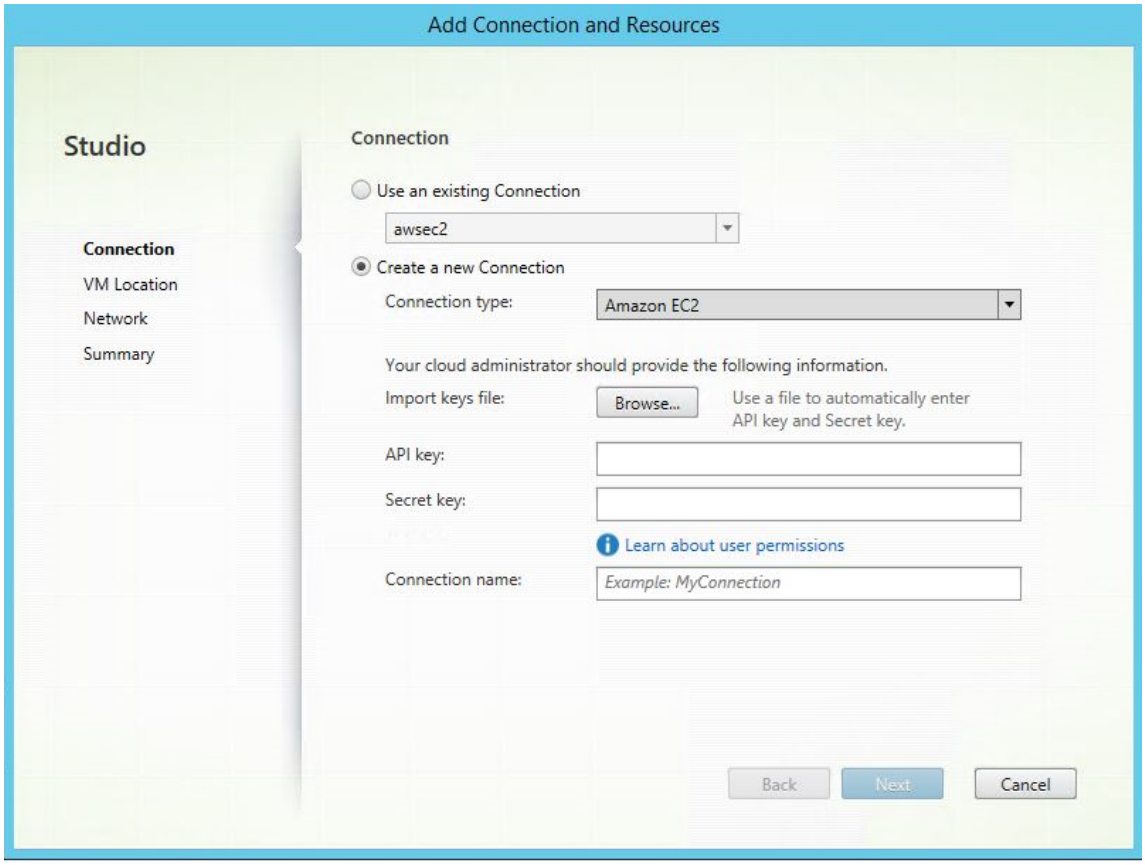
Use MCS to create Linux VMs on AWS

Step 1: Create a hosting connection to AWS in Citrix Studio

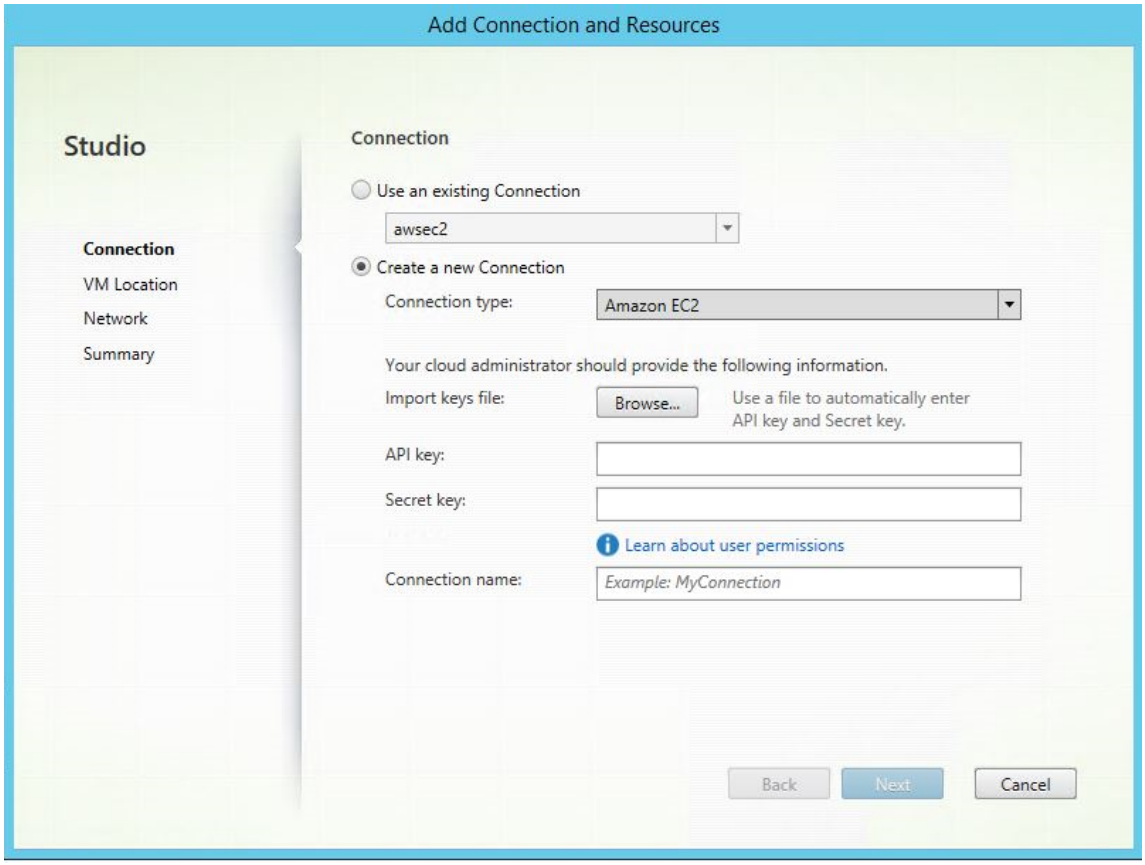
1. In Citrix Studio on Citrix Cloud, choose **Configuration** > **Hosting** > **Add Connection and Resources** to create a connection to AWS.



2. Choose **Amazon EC2** as the connection type.



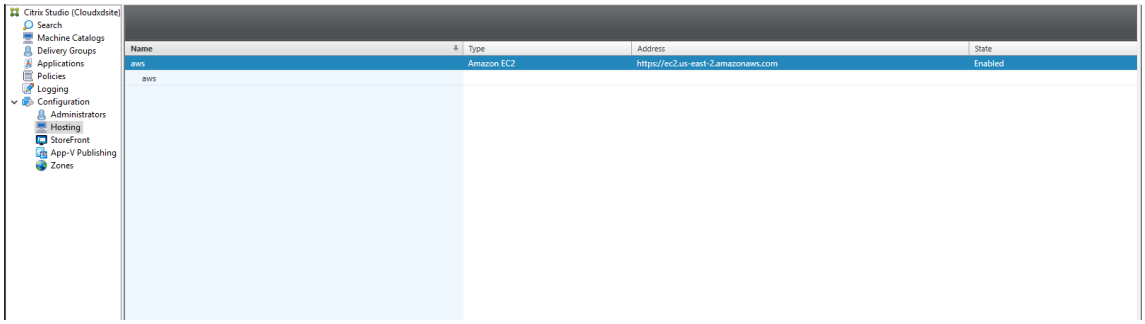
3. Type the API key and secret key of your AWS account and type your connection name.



The **API key** is your access key ID and the **Secret key** is your secret access key. They are considered as an access key pair. If you lose your secret access key, you can delete the access key and create another one. To create an access key, do the following:

a) Sign in to the AWS services.
b) Navigate to the Identity and Access Management (IAM) console.
c) On the left navigation pane, choose **Users**.
d) Select the target user and scroll down to select the **Security credentials** tab.
e) Scroll down and click **Create access key**. A new window appears.
f) Click **Download .csv file** and save the access key to a secure location.

A new connection appears in the hosting pane.



Step 2: Prepare a master image

A master image contains the operating system, non-virtualized applications, VDA, and other software. To prepare a master image, do the following:

Step 2a: Configure cloud-init

1. To ensure that a VDA host name persists when an EC2 instance is restarted or stopped, run the following command to preserve the VDA host name.

   ```
   echo "preserve_hostname: true" > /etc/cloud/cloud.cfg.d/99_hostname.cfg
   <!--NeedCopy-->
   ```

   For Ubuntu 18.04, ensure that the following lines are present under the system_info section in the /etc/cloud/cloud.cfg file:

   ```
   system_info:
       network:
           renderers: ['netplan', 'eni', 'sysconfig']
   <!--NeedCopy-->
   ```

2. To use SSH for remotely accessing MCS-created VMs on AWS, enable password authentication because no key name is attached to those VMs. Do the following as needed.

   - Edit the `cloud-init` configuration file, /etc/cloud/cloud.cfg. Ensure that the **ssh_pwauth: true** line is present. Remove or comment the **set-password** line and the following lines if they exist.

     ```
     users:
     - default
     <!--NeedCopy-->
     ```

   - If you plan to use the default user `ec2-user` or `ubuntu` created by `cloud-init`, you can change the user password by using the `passwd` command. Keep the new password in mind for later use to log in to the MCS-created VMs.

   - Edit the `/etc/ssh/sshd_config` file to ensure that the following line is present:

     ```
     PasswordAuthentication yes
     <!--NeedCopy-->
     ```

     Save the file and run the `sudo service sshd restart` command.

Step 2b: (For Ubuntu 16.04 only) Install OpenJDK 11 On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 2c: Install the Linux VDA package on the template VM

> **Note:**
>
> To use a currently running VDA as the template VM, omit this step.
>
> Before installing the Linux VDA package on the template VM, install .NET Core Runtime 3.1. For more information, see Installation overview.

Based on your Linux distribution, run the following command to set up the environment for the Linux VDA:

For RHEL/CentOS:

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

For Ubuntu/Debian:

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

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

For SUSE 12:

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

Step 2d: Install the EPEL repository that contains ntfs-3g Install the EPEL repository on RHEL 8/CentOS 8, RHEL 7/CentOS 7 so that running deploymcs.sh later installs the ntfs-3g package contained in it.

Step 2e: Manually install ntfs-3g on SUSE 12 On the SUSE 12 platform, there is no repository providing ntfs-3g. Download the source code, compile, and install ntfs-3g manually:

1. Install the GNU Compiler Collection (GCC) compiler system and the make package:


```
sudo zypper install gcc
sudo zypper install make
<!--NeedCopy-->
```

2. Download the ntfs-3g package.
3. Decompress the ntfs-3g package:

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

4. Enter the path to the ntfs-3g package:

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

5. Install ntfs-3g:

```
./configure
make
make install
<!--NeedCopy-->
```

Step 2f: Set up the runtime environment Before running `deploymcs.sh`, do the following:

- Change variables in `/etc/xdl/mcs/mcs.conf`. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA. The following are some of the variables, of which `dns` and `AD_INTEGRATION` must be set:

  Note: If a variable can be set with multiple values, put the values inside single quotes and separate them with spaces. For example, LDAP_LIST='aaa.lab:389 bbb.lab:389.'

  - `Use_Existing_Configurations_Of_Current_VDA`: Determines whether to use the existing configurations of the currently running VDA. If set to Y, the configuration files on MCS-created machines are the same as the equivalents on the currently running VDA. However, you still must configure the `dns` and `AD_INTEGRATION` variables. The default value is N, which means configuration files on MCS-created machines are determined by configuration templates on the master image.
  - `dns`: Sets the DNS IP address.
  - `AD_INTEGRATION`: Sets Winbind or SSSD (SSSD is not supported on SUSE).
  - `WORKGROUP`: Sets the workgroup name (case-sensitive) if it is configured in AD.

- On the template machine, add command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file for writing or updating registry values as required. This action prevents the loss of data and settings every time an MCS-provisioned machine restarts.

Each line in the /etc/xdl/mcs/mcs_local_setting.reg file is a command for setting or updating a registry value.

For example, you can add the following command lines to the /etc/xdl/mcs/mcs_local_setting.reg file to write or update a registry value respectively:

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

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

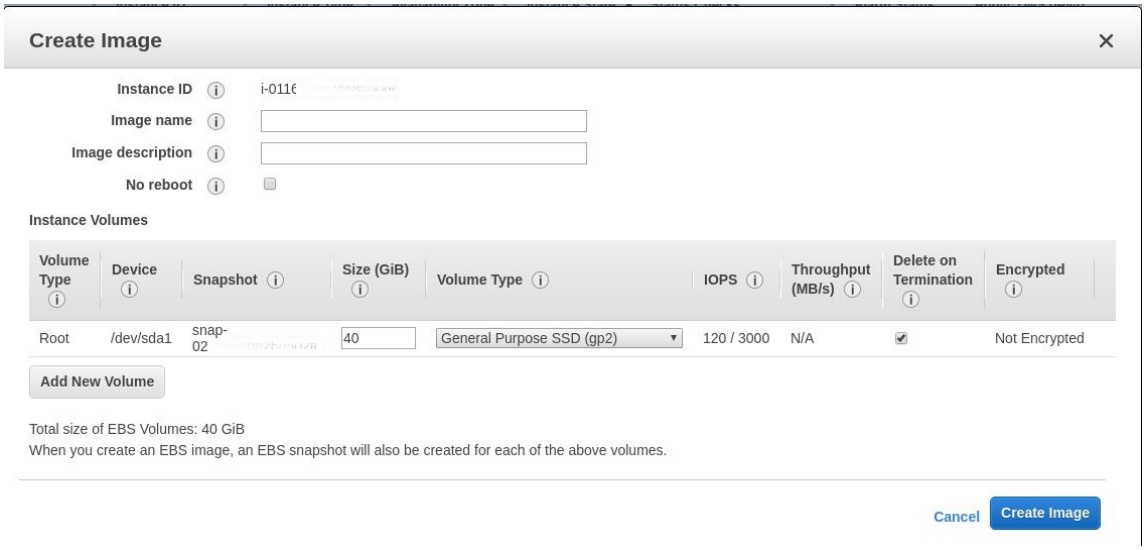
Step 2g: Create a master image

1. Run /opt/Citrix/VDA/sbin/deploymcs.sh.
2. (Optional) On the template VM, update the configuration templates to customize the relevant /etc/krb5.conf, /etc/samba/smb.conf, and /etc/sssd/sssd.conf files on all created VMs.

   For Winbind users, update the /etc/xdl/mcs/winbind_krb5.conf.tmpl and /etc/xdl/mcs/winbind_smb.conf.tmpl templates.

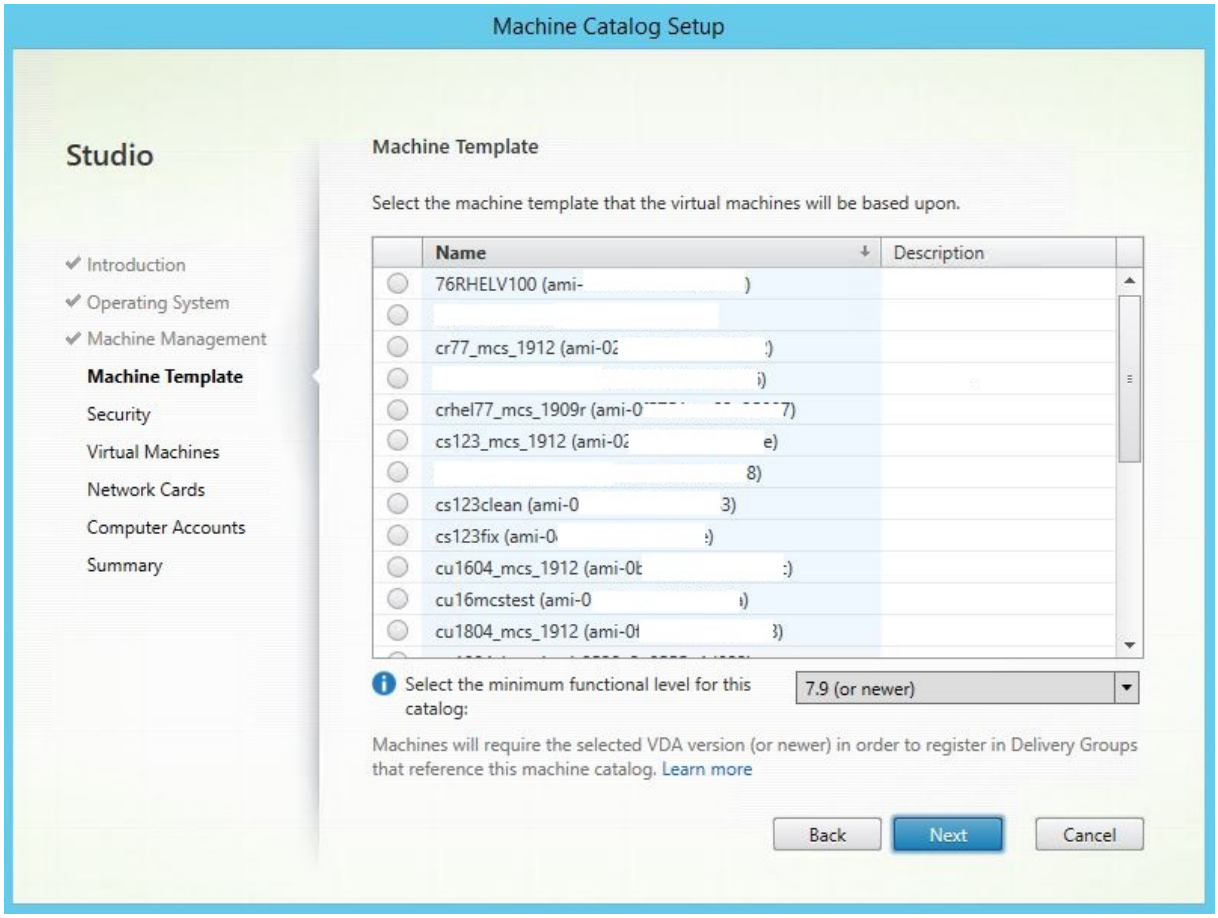
   For SSSD users, update the /etc/xdl/mcs/sssd.conf.tmpl, /etc/xdl/mcs/sssd_krb5.conf.tmpl, and /etc/xdl/mcs/sssd_smb.conf.tmpl templates.

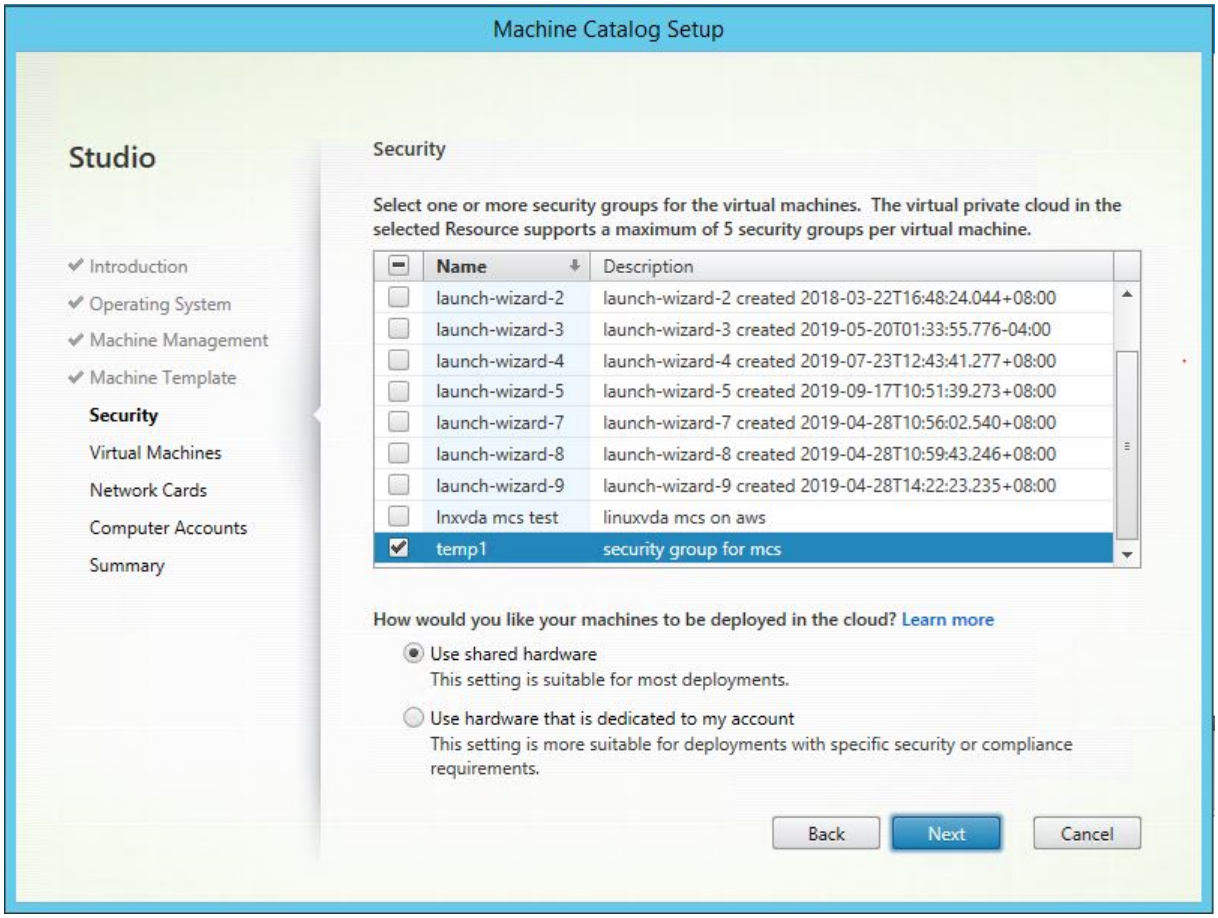
   Note: Keep the existing format used in the template files and use variables such as $WORKGROUP, $REALM, $realm, and $AD_FQDN.
3. Install applications on the template VM and shut down the template VM from the AWS EC2 portal. Ensure that the instance state of the template VM is **Stopped**.
4. Right-click the template VM and select **Image > Create Image**. Type information and make settings as needed. Click **Create Image**.



Step 3: Create a Machine Catalog

In Citrix Studio, create a Machine Catalog and specify the number of VMs to create in the catalog. When creating the Machine Catalog, choose your machine template (the master image you created earlier) and select one or more security groups.





Do other configuration tasks as needed. For more information, see Create a machine catalog using Studio.

Step 4: Create a Delivery Group

A Delivery Group is a collection of machines selected from one or more Machine Catalogs. The Delivery Group specifies which users can use those machines, and the applications and desktops available to those users. For more information, see Create Delivery Groups.

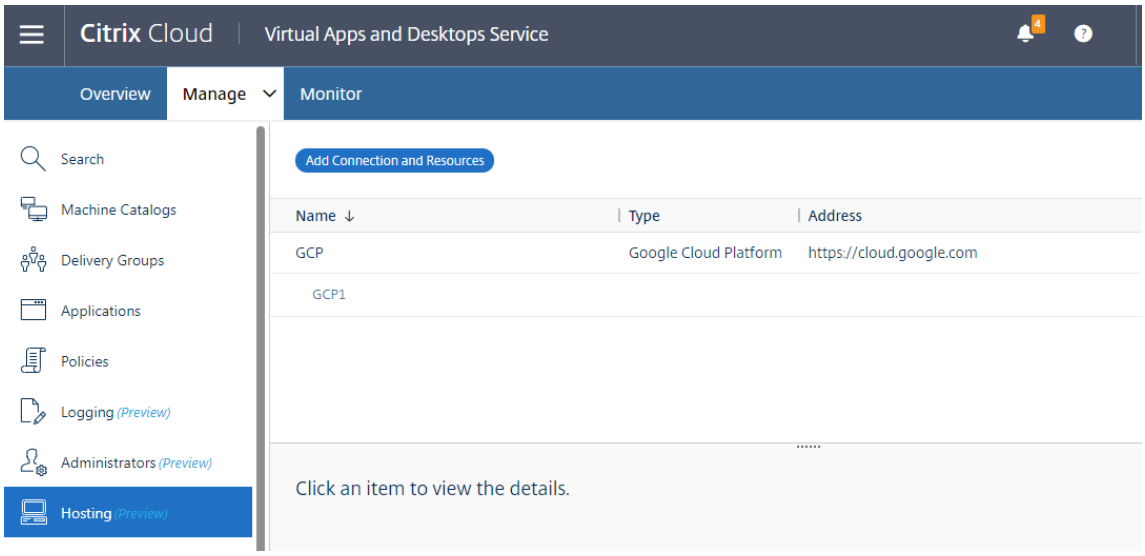
Use MCS to create Linux VMs on GCP

Step 1: Set up your GCP environment

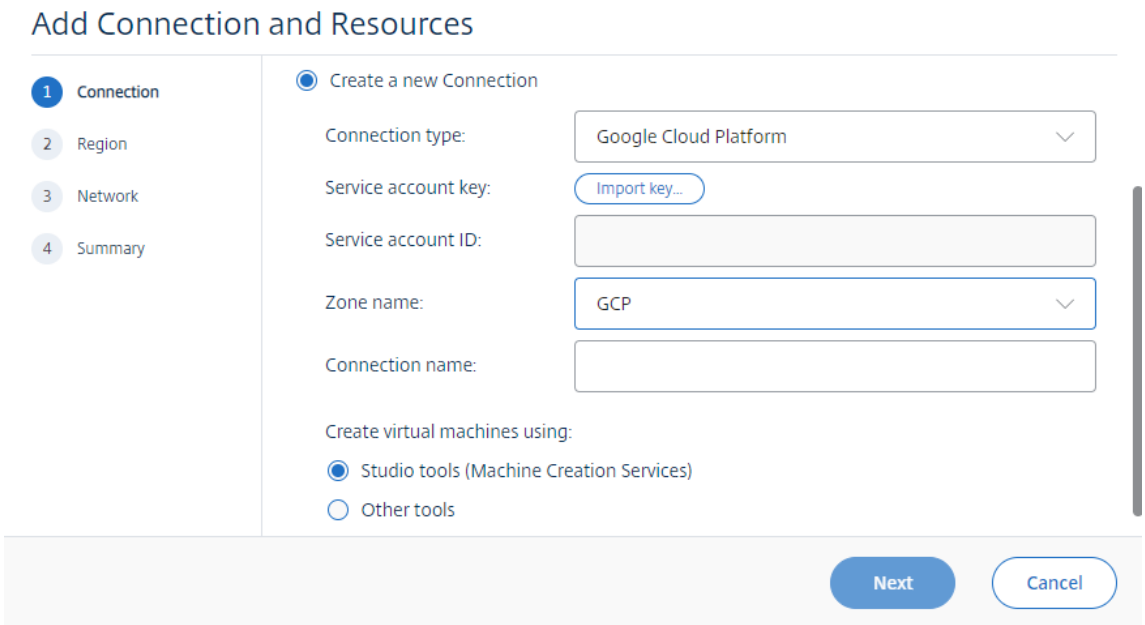
For more information, see Google Cloud Platform virtualization environments.

Step 2: Create a hosting connection to GCP in Citrix Studio

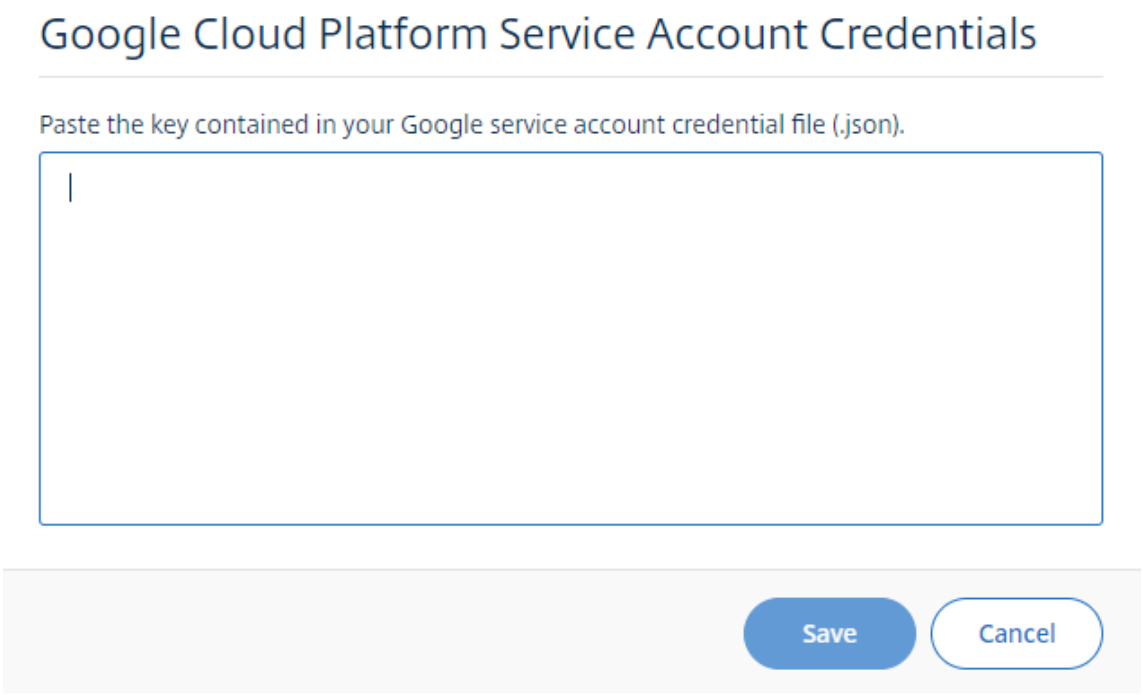
1. In Citrix Studio on Citrix Cloud, choose **Configuration** > **Hosting** > **Add Connection and Resources** to create a connection to GCP.



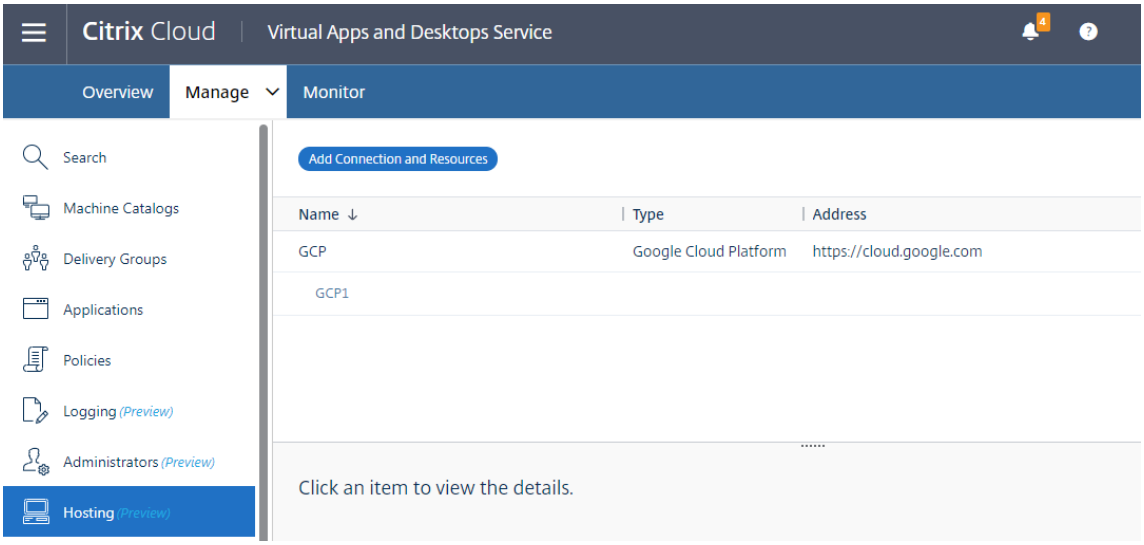
2. Choose **Google Cloud Platform** as the connection type.



3. Import the service account key of your GCP account and type your connection name.



A new connection appears in the hosting pane.



Step 3: Prepare a master image

A master image contains the operating system, non-virtualized applications, VDA, and other software. To prepare a master image, do the following:

Step 3a: (For Ubuntu 16.04 only) Install OpenJDK 11 On Ubuntu 16.04, install OpenJDK 11 by completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

Step 3b: Install the Linux VDA package on the template VM

> **Note:**
>
> To use a currently running VDA as the template VM, omit this step.
>
> Before installing the Linux VDA package on the template VM, install .NET Core Runtime 3.1. For more information, see Installation overview.

Based on your Linux distribution, run the following command to set up the environment for the Linux VDA:

For RHEL/CentOS:

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

For Ubuntu/Debian:

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

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

For SUSE 12:

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

Step 3c: Install the EPEL repository that contains ntfs-3g Install the EPEL repository on RHEL 8/CentOS 8, RHEL 7/CentOS 7 so that running deploymcs.sh later installs the ntfs-3g package contained in it.

Step 3d: Manually install ntfs-3g on SUSE 12 On the SUSE 12 platform, there is no repository providing ntfs-3g. Download the source code, compile, and install ntfs-3g manually:

1. Install the GNU Compiler Collection (GCC) compiler system and the make package:


```
sudo zypper install gcc
sudo zypper install make
<!--NeedCopy-->
```

2. Download the ntfs-3g package.
3. Decompress the ntfs-3g package:

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

4. Enter the path to the ntfs-3g package:

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

5. Install ntfs-3g:

```
./configure
make
make install
<!--NeedCopy-->
```

Step 3e: Set up the runtime environment Before running `deploymcs.sh`, do the following:

- Change variables in `/etc/xdl/mcs/mcs.conf`. The `mcs.conf` configuration file contains variables for setting MCS and the Linux VDA. The following are some of the variables, of which `dns` and `AD_INTEGRATION` must be set:

  Note: If a variable can be set with multiple values, put the values inside single quotes and separate them with spaces. For example, LDAP_LIST='aaa.lab:389 bbb.lab:389.'

  - `Use_Existing_Configurations_Of_Current_VDA`: Determines whether to use the existing configurations of the currently running VDA. If set to Y, the configuration files on MCS-created machines are the same as the equivalents on the currently running VDA. However, you still must configure the `dns` and `AD_INTEGRATION` variables. The default value is N, which means configuration files on MCS-created machines are determined by configuration templates on the master image.
  - `dns`: Sets the DNS IP address.
  - `AD_INTEGRATION`: Sets Winbind or SSSD (SSSD is not supported on SUSE).
  - `WORKGROUP`: Sets the workgroup name (case-sensitive) if it is configured in AD.

- On the template machine, add command lines to the `/etc/xdl/mcs/mcs_local_setting.reg` file for writing or updating registry values as required. This action prevents the loss of data and settings every time an MCS-provisioned machine restarts.

Each line in the /etc/xdl/mcs/mcs_local_setting.reg file is a command for setting or updating a registry value.

For example, you can add the following command lines to the /etc/xdl/mcs/mcs_local_setting.reg file to write or update a registry value respectively:

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

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

Step 3f: Create a master image

1. Run /opt/Citrix/VDA/sbin/deploymcs.sh.
2. (Optional) On the template VM, update the configuration templates to customize the relevant /etc/krb5.conf, /etc/samba/smb.conf, and /etc/sssd/sssd.conf files on all created VMs.

   For Winbind users, update the /etc/xdl/mcs/winbind_krb5.conf.tmpl and /etc/xdl/mcs/winbind_smb.conf.tmpl templates.

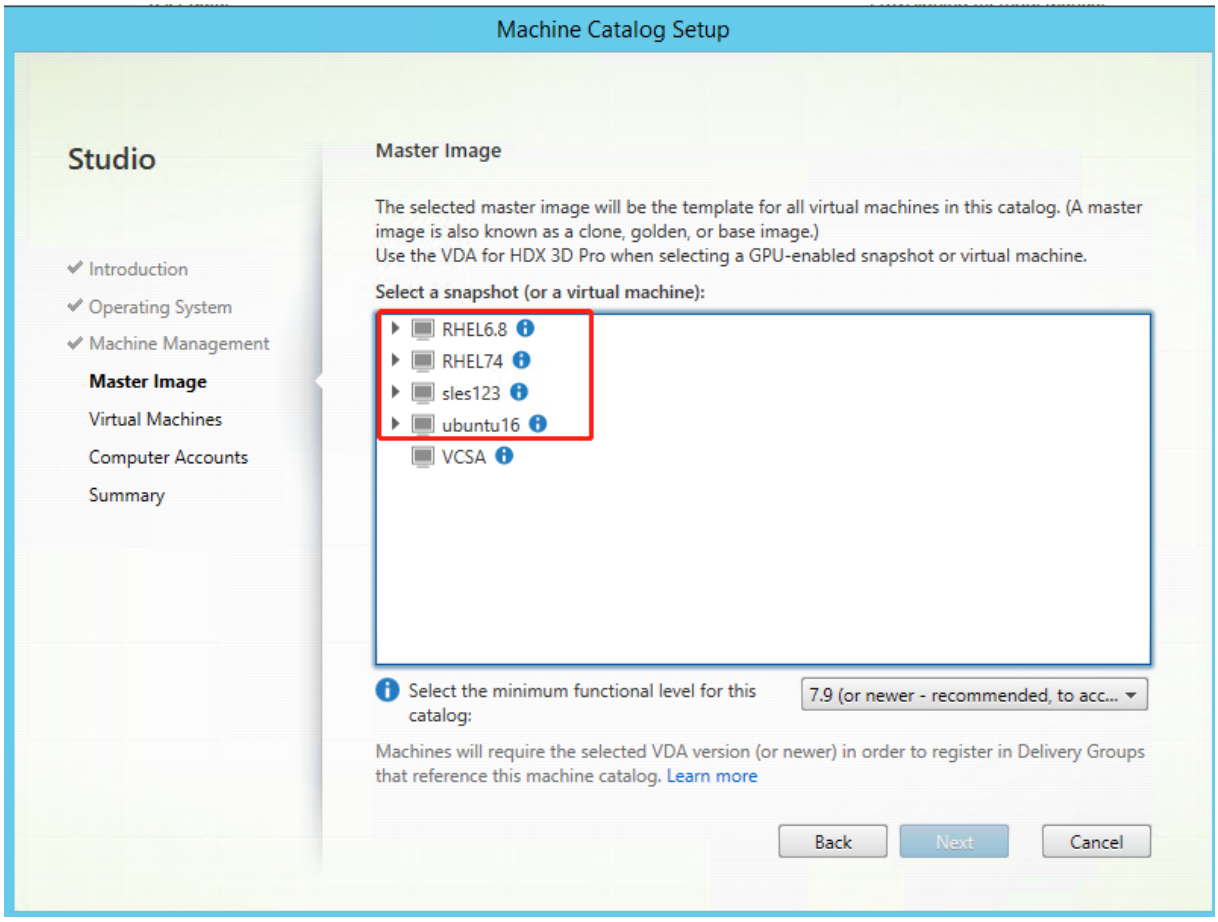
   For SSSD users, update the /etc/xdl/mcs/sssd.conf.tmpl, /etc/xdl/mcs/sssd_krb5.conf.tmpl, and /etc/xdl/mcs/sssd_smb.conf.tmpl templates.

   > **Note:**
   >
   > Keep the existing format used in the template files and use variables such as $WORKGROUP, $REALM, $realm, and $AD_FQDN.

3. After you finish installing applications on the template VM, shut down the template VM from the VMware. Take a snapshot of the template VM.

Step 4: Create a Machine Catalog

In Citrix Studio, create a Machine Catalog and specify the number of VMs to create in the catalog. When creating the Machine Catalog, choose your master image from the snapshot list.



Do other configuration tasks as needed. For more information, see Create a machine catalog using Studio.

Step 5: Create a Delivery Group

A Delivery Group is a collection of machines selected from one or more Machine Catalogs. The Delivery Group specifies which users can use those machines, and the applications and desktops available to those users. For more information, see Create Delivery Groups.

Use MCS to upgrade your Linux VDA

To use MCS to upgrade your Linux VDA, do the following:

1. Ensure that you installed .NET Core Runtime 3.1 before you upgrade your Linux VDA to the current release.
2. Upgrade your Linux VDA on the template machine:

   For RHEL 7 and CentOS 7:


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

For RHEL 8 and CentOS 8:

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

For SUSE 12:

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

For Ubuntu 16.04:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu16.04_amd64.deb
<!--NeedCopy-->
```

For Ubuntu 18.04:

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

For Ubuntu 20.04:

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

3. Edit `/etc/xdl/mcs/mcs.conf` and `/etc/xdl/mcs/mcs_local_setting.reg`.
4. Take a new snapshot.
5. In Citrix Studio, select the new snapshot to update your Machine Catalog. Wait before each machine restarts. Do not restart a machine manually.

Automate machine account password updates

Machine account passwords, by default, expire 30 days after the machine catalog is created. To prevent password expiration and to automate machine account password updates, do the following:

1. Add the following entry to /etc/xdl/mcs/mcs.conf before running /opt/Citrix/VDA/sbin/deploymcs.sh.

   ```
   UPDATE_MACHINE_PW="enabled"
   ```

2. After running /opt/Citrix/VDA/sbin/deploymcs.sh, open /etc/cron.d/mcs_update_password_cronjob to set the update time and frequency. The default setting updates machine account passwords weekly at 2:30AM, Sunday.

After each machine account password update, the ticket cache on the Delivery Controller becomes invalid and the following error might appear in /var/log/xdl/jproxy.log:

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

To eliminate the error, clear the ticket cache regularly. You can schedule a cache cleanup task on all Delivery Controllers or on the domain controller.

Enable FAS on MCS-created VMs

You can enable FAS on MCS-created VMs that run on the following distributions:

	Winbind	SSSD	Centrify
RHEL 8, CentOS 8	Yes	No	No
RHEL 7, CentOS 7	Yes	Yes	No
Ubuntu 20.04	Yes	No	No
Ubuntu 18.04	Yes	No	No
Ubuntu 16.04	Yes	No	No
Debian 10.7	Yes	No	No
SUSE 12.5	Yes	No	No

Enable FAS when you are preparing a master image on the template VM

1. Run the script `opt/Citrix/VDA/sbin/ctxinstall.sh` and set all environment variables such as the list of FAS servers. For more information about the environment variables, see Easy install.

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

2. Import the root CA certificate.

   ```
   1 sudo cp root.pem /etc/pki/CA/certs/
   2 <!--NeedCopy-->
   ```

3. Run ctxfascfg.sh.
4. Set variables in `/etc/xdl/mcs/mcs.conf`.

a) Set the value of `Use_Existing_Configurations_Of_Current_VDA` to Y.
b) Set the `FAS_LIST` variable to your FAS server address or multiple FAS server addresses that are separated by semicolons and enclosed by double quotes, for example, `FAS_LIST="<FAS_SERVER_FQDN>;<FAS_SERVER_FQDN>"`.
c) Set the other variables as required, such as `VDI_MODE`.

5. Run the script `/opt/Citrix/VDA/sbin/deploymcs.sh`.

Enable FAS on an MCS-created VM

If FAS is not enabled on the template machine as described earlier, you can enable FAS on each MCS-created VM.

To enable FAS on an MCS-created VM, do the following:

1. Set variables in /etc/xdl/mcs/mcs.conf.

   a) Set the value of `Use_Existing_Configurations_Of_Current_VDA` to Y.
   b) Set the `FAS_LIST` variable to your FAS server address.
   c) Set the other variables as required, such as `VDI_MODE`.

2. Import the root CA certificate.

```
sudo cp root.pem /etc/pki/CA/certs/
<!--NeedCopy-->
```

3. Run the `/opt/Citrix/VDA/sbin/ctxfascfg.sh` script.

> **Note:**
>
> You must set all necessary variables in `/etc/xdl/mcs/mcs.conf` because these variables are called upon VM startup.

Use Citrix Provisioning to create Linux VMs

June 11, 2021

This article provides information about the Citrix Provisioning Linux streaming feature. Using this feature, you can provision Linux virtual desktops directly in the Citrix Virtual Apps and Desktops environment. For more information, see the Citrix Provisioning documentation.

The following Linux distributions are supported:

- Ubuntu 16.04

- Ubuntu 18.04.5 (experimental)
- RHEL 8.3 (experimental)

> **Important:**
>
> - To use this feature for Ubuntu 18.04.5 and RHEL 8.3, use the **PVS Linux Streaming Agent (Ubuntu 18.04)-Experimental** package and the **PVS Linux Streaming Agent (RHEL8.3)-Experimental** package, respectively. The installation packages are available on the Linux VDA download page.
> - To use this feature for Ubuntu 16.04, download the latest Citrix Provisioning ISO and locate the target software for Ubuntu 16.04. For more information, see Configure Linux Streaming in the Citrix Provisioning documentation.

Consider the following when provisioning Linux target devices:

- Sometimes, the client drive cannot be mapped to a provisioned Linux VM session. To resolve this issue, halt the CDM service using `service ctxcdm stop`, before installing the Citrix Provisioning target device, then run the `pvs-imager` command to convert it.
- Linux streaming only supports Winbind as the tool for joining a Windows domain.
- When you enable RAM cache for the Linux device, set the cache size to 8 MB (the minimum value). Linux uses as much RAM as necessary, including all available memory, for the write cache. The amount specified in the console is the amount reserved up front. Citrix recommends that you reserve as little as possible, which effectively allows Linux to manage memory usage.
- The target device name in the Citrix Provisioning imager UI typically defaults to `im\\_localhost`. This value must be changed when you create more than one vDisk. Using the same target device name causes the imager command to fail.
- Installation (and subsequent updates) must be done in super user mode. There are two ways to install as a super user:
  - Enter user mode in a terminal using the `su` command.
  - Enter `sudo` before the command. For example, `sudo yum install tdb-tools`; enter `sudo` for every command.
- The Linux client's system clock must be synchronized by using the active directory controller.
- UEFI is not supported.
- VMM is not supported.
- The write cache drive must have the label `PVS_Cache` for it to be used as a write cache. The entire partition is used.
- English localizations are displayed on non-English installations.
- SE Linux is not supported.
- Targets running on XenServer must run in HVM mode.
- After booting a Linux target device, a warning message might display indicating a SE Linux Alert Browser.

- Two streamed Ubuntu 18.04 VMs hosted on ESXi get the same IP address through DHCP. To resolve this issue, configure the VM to use the MAC address as a unique ID to retrieve an IP address through DHCP.
- For Ubuntu 18.04.5 and RHEL 8.3, machine account passwords do not update in Active Directory automatically. When a password expires and the streamed VM fails to join the domain, try to reset the password through the Citrix Provisioning Console.
- For Ubuntu 16.04, only Winbind provided by Samba 4.4 and earlier releases is supported when you provision Linux target devices using Citrix Provisioning.

Installation options

To install the Linux streaming component, you must be logged in as an administrator. When installing, consider that the following commands must be issued in a root shell, or by using `sudo` privileges.

> **Note:**
>
> A self-signed certificate must be created if streaming Citrix Provisioning Linux target devices. The Soap server uses an SSL connection requiring you to configure an X.509 certificate on the Soap server.
>
> The certificate's CA must also be present on the Provisioning Server and the Linux target device. For information on creating a self-signed certificate, see Creating self-signed certificates for Linux streaming.

For Ubuntu 16.04 distributions:

```
sudo dpkg -i pvs-<version>.deb

sudo apt-get -yf install
<!--NeedCopy-->
```

For Ubuntu 18.04 distributions:

```
sudo apt-get -y install dracut dracut-network tdb-tools python3 python3
   -distutils
sudo dpkg -i pvs_<version>_ubuntu18.04_amd64.deb
<!--NeedCopy-->
```

For RHEL 8.3 distributions:

```
yum –nogpgcheck localinstall pvs_<version>_rhel8.3_x86_64.deb
<!--NeedCopy-->
```

Using the GUI to create a Linux golden image

To invoke the GUI to install this feature:

1. Log in while an administrator.
2. Run the `pvs-imager` command:

> **Tip:**
>
> When the `pvs-imager` command fails due to a host name issue, verify that your network configuration is correct. Do not set the system's host name to `localhost`. On RHEL8.3, log on with a X11 display server instead of Wayland to use the GUI.

After running the command, the UI page displays:

Using the command line interface to install the Linux streaming feature

To invoke the command line to install this feature:

1. Log in while as an administrator.
2. Run the following command:

   ```
   pvs-imager -C
   ```

The command-line installation includes two options:

- \-C allows you to create a vDisk
- \-U allows you to update an existing vDisk

The following information illustrates non-GUI related installation options for the Linux Streaming feature:

```
Usage: ./pvs-imager \[-hCU] \[-a|--address=<IPaddr>] \[-u|--username=<
    username>] \[-p|--password=<password>] \[-P|--port=<port>] \[-d|--
    domain=<domain] \[-S|--store=<store>] \[-v|--vdisk=<vdisk name>] \[-
    s|--size=<vdisk size] \[-D|--device=<sourceDevice>] \[-c|--
    collection=<collection>] \[-n|--name=<name>]
 Non-GUI Modes:
  -C      -  Create a new vDisk
   ---OR----
  -U      -  Update an existing vDisk

 General Options:
  -a <server IP> -  Address or hostname of PVS server
  -u <username>  -  Username for API login
  -p <password>  -  Password for API login
  -d <domain>   -  AD domain for API login
  -P <port>     -  Base port for API login (default: 54321)
  -S <store>    -  Store containing vDisk
  -c <collection> -  Collection to store imaging device in
  -n <name>     -  Device name for imaging device
  -v <name>     -  vDisk name
  -s <size>     -  vDisk size (Create Mode only, default: sourceDevice
     size)
  -D <sourceDev> -  devnode to clone
  -V       -  increment debug verbosity (up to 5 times)
  -g <grubMode>  -  Supported Grub settings ( 'debug' )
```

Supported file systems for imaging are ext4, xfs, or btrfs.

> **Tip:**
>
> Debugging logs for pvs-imager, created using -VVVVV switch, are created in the folder that executed the pvs-imager tool. The name of the log file is pvs-imager.log.

About disk caching

For hard disk caching or hard disk overflow caching without the Citrix Virtual Apps and Desktops Setup Wizard, format the target device disk using a formatted partition. Include the label PVS_Cache. This

object can be created with the `mkfs -L PVS_Cache` command on the target device. Any case-sensitive file system can be used for the cache, but XFS is recommended.

> **Tip:**
>
> An administrator can create any cache disk selection logic for their environment by writing a bash script that runs at launch time. The script would look for a cache device candidate by whatever mechanism is best suited to the environment, running `mkfs` on it, and rebooting.

When configuring disk caching:

- Citrix recommends using the Citrix Virtual Apps and Desktops Setup Wizard to create the Linux target device.
- Manually creating the label requires adherence to case sensitivity to avoid configuration conflicts.
- Alternately, consider using the manual method for creating the write cache.

Manually creating the write cache for a target device

By default, the Citrix Virtual Apps and Desktops Setup Wizard ignores drives that are attached to the current template. The wizard creates a write cache based on the parameters you provide. Sometimes, the write cache drive encounters problems during automatic creation using the wizard. Or, when the target device continuously falls back to the server side cache as a result of a problem with the created drive. To resolve these issues, create the object manually using the `mkfs -L PVS_Cache` command on the target device.

The Citrix Virtual Apps and Desktops Setup Wizard recognizes manually created write cache changes for the target device by default when you use the `UseTemplatecache` parameter. On the provisioning server running the Citrix Virtual Apps and Desktops Setup Wizard, or where the remote provisioning console points, change the registry setting:

Create the following registry key on the provisioning console machine to disable the template cache:

`HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ProvisioningServices`

`Name: UseTemplateCache`

`Type: DWORD`

`Value: 0`

Run the Citrix Virtual Apps and Desktops Setup Wizard. On the **Virtual machines** page change the local write cache disk size to 0 GB (default is 6 GB).

Install Linux Virtual Delivery Agent for RHEL/CentOS

June 10, 2022

You can choose to follow the steps in this article for manual installation or use easy install for automatic installation and configuration. Easy install saves time and labor and is less error-prone than the manual installation.

> **Note:**
>
> Use easy install only for fresh installations. Do not use easy install to update an existing installation.

Step 1: Prepare RHEL 8/CentOS 8, RHEL 7/CentOS 7 for VDA installation

Step 1a: Verify the network configuration

We recommend that the network is connected and configured correctly before proceeding.

Step 1b: Set the host name

To ensure that the host name of the machine is reported correctly, change the **/etc/hostname** file to contain only the host name of the machine.

`hostname`

Step 1c: Assign a loopback address to the host name

To ensure that the DNS domain name and Fully Qualified Domain Name (FQDN) of the machine are reported back correctly, change the following line of the `/etc/hosts` file to include the FQDN and host name as the first two entries:

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

For example:

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

Remove any other references to **hostname-fqdn** or **hostname** from other entries in the file.

> **Note:**
>
> The Linux VDA currently does not support NetBIOS name truncation. Therefore, the host name must not exceed 15 characters.

> **Tip:**
>
> Use a–z, A–Z, 0–9, and hyphen (-) characters only. Avoid underscores (_), spaces, and other symbols. Do not start a host name with a number and do not end with a hyphen. This rule also applies to Delivery Controller host names.

Step 1d: Check the host name

Verify that the host name is set correctly:

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

This command returns only the machine's host name and not its fully qualified domain name (FQDN).

Verify that the FQDN is set correctly:

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

This command returns the FQDN of the machine.

Step 1e: Check name resolution and service reachability

Verify that you can resolve the FQDN and ping the domain controller and Delivery Controller:

```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

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

If you cannot resolve the FQDN or ping either of these machines, review the steps before proceeding.

Step 1f: Configure clock synchronization

Maintaining accurate clock synchronization between the VDAs, Delivery Controllers, and domain controllers is crucial. Hosting the Linux VDA as a virtual machine can cause clock skew problems. For this reason, synchronizing time with a remote time service is preferred.

An RHEL 8/RHEL 7 default environment uses the Chrony daemon (`chronyd`) for clock synchronization.

Configure the Chrony service As a root user, edit **/etc/chrony.conf** and add a server entry for each remote time server:

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

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

In a typical deployment, synchronize time from the local domain controllers and not directly from public NTP pool servers. Add a server entry for each Active Directory domain controller in the domain.

Remove any other server entries listed including loopback IP address, localhost, and public server ***.pool.ntp.org** entries.

Save changes and restart the Chrony daemon:

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

Step 1g: Install OpenJDK 11

The Linux VDA requires the presence of OpenJDK 11. The runtime environment is automatically installed as a dependency when you install the Linux VDA.

Confirm the correct version:

```
sudo yum info java-11-openjdk
<!--NeedCopy-->
```

The prepackaged OpenJDK might be an earlier version. Update to OpenJDK 11:

```
sudo yum -y update java-11-openjdk
<!--NeedCopy-->
```

Open a new shell and verify the version of Java:

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

> **Tip:**
>
> To avoid registration failure with the Delivery Controller, ensure that you installed only OpenJDK 11. Remove all other versions of Java from your system.

Step 1h: Install PostgreSQL

The Linux VDA requires either PostgreSQL 10.5 or later on RHEL 8 or PostgreSQL 9.2 or later on RHEL 7.

Install the following packages:

```
sudo yum -y install postgresql-server

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

The following post-installation step is required to initialize the database and to ensure that the service starts upon machine startup. This action creates database files under **/var/lib/pgsql/data**. The command differs between PostgreSQL 10 and 9:

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

Step 1i: Start PostgreSQL

Start the service upon machine startup and start the service immediately:

```
sudo systemctl enable postgresql

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

Check the version of PostgreSQL by using:

```
psql --version
<!--NeedCopy-->
```

(RHEL 7 only) Verify that the data directory is set by using the **psql** command-line utility:

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

Step 2: Prepare the hypervisor

Some changes are required when running the Linux VDA as a virtual machine on a supported hypervisor. Make the following changes according to the hypervisor platform in use. No changes are required if you are running the Linux machine on bare metal hardware.

Fix time synchronization on Citrix Hypervisor

When the Citrix Hypervisor Time Sync feature is enabled, within each paravirtualized Linux VM you experience issues with NTP and Citrix Hypervisor, both of which try to manage the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each

Linux guest is synchronized with the NTP. This case requires disabling host time synchronization. No changes are required in HVM mode.

On some Linux distributions, if you are running a paravirtualized Linux kernel with Citrix VM Tools installed, you can check whether the Citrix Hypervisor Time Sync feature is present and enabled from within the Linux VM:

```
su -

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

This command returns 0 or 1:

- 0 - The time sync feature is enabled, and must be disabled.
- 1 - The time sync feature is disabled, and no further action is required.

If the /proc/sys/xen/independent_wallclock file is not present, the following steps are not required.

If enabled, disable the time sync feature by writing 1 to the file:

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

To make this change permanent and persistent after restart, edit the **/etc/sysctl.conf** file and add the line:

```
xen.independent_wallclock = 1
```

To verify these changes, restart the system:

```
su -

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

This command returns the value 1.

Fix time synchronization on Microsoft Hyper-V

The Linux VMs with Hyper-V Linux Integration Services installed can apply the Hyper-V time synchronization feature to use the time of the host operating system. To ensure that the system clock remains accurate, you must enable this feature alongside the NTP services.

From the management operating system:

1. Open the Hyper-V Manager console.
2. For the settings of a Linux VM, select **Integration Services**.
3. Ensure that **Time synchronization** is selected.

> **Note:**
>
> This approach is different from VMware and Citrix Hypervisor, where host time synchronization is disabled to avoid conflicts with NTP. Hyper-V time synchronization can coexist and supplement NTP time synchronization.

Fix time synchronization on ESX and ESXi

When the VMware Time Synchronization feature is enabled, within each paravirtualized Linux VM you experience issues with the NTP and the hypervisor, both of which try to synchronize the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each Linux guest is synchronized with the NTP. This case requires disabling host time synchronization.

If you are running a paravirtualized Linux kernel with VMware Tools installed:

1. Open the vSphere Client.
2. Edit settings for the Linux VM.
3. In the **Virtual Machine Properties** dialog, open the **Options** tab.
4. Select **VMware Tools**.
5. In the **Advanced** box, clear **Synchronize guest time with host**.

Step 3: Add the Linux virtual machine (VM) to the Windows domain

The Linux VDA supports several methods for adding Linux machines to the Active Directory (AD) domain:

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

Follow instructions based on your chosen method.

> **Note:**
>
> Session launches might fail when the same user name is used for the local account in the Linux VDA and the account in AD.

Samba Winbind

Install or update the required packages:

For RHEL 8/CentOS 8:


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

For RHEL 7/CentOS 7:

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

Enable Winbind daemon to start upon machine startup The Winbind daemon must be configured to start upon machine startup:

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

Configure Winbind Authentication Configure the machine for Kerberos authentication by using Winbind:

1. Run the following command.

   For RHEL 8:

   ```
   sudo authselect select winbind with-mkhomedir --force
   <!--NeedCopy-->
   ```

   For RHEL 7:

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

   Where **REALM** is the Kerberos realm name in uppercase and **domain** is the NetBIOS name of the domain.

   If DNS-based lookup of the KDC server and realm name is required, add the following two options to the previous command:

   `--enablekrb5kdcdns --enablekrb5realmdns`

   Ignore any errors returned from the `authconfig` command about the `winbind` service failing to start. The errors can occur when `authconfig` tries to start the `winbind` service without the machine yet being joined to the domain.

2. Open **/etc/samba/smb.conf** and add the following entries under the [Global] section, but after the section generated by the `authconfig` tool:

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

3. (RHEL 8 only) Open **/etc/krb5.conf** and add entries under the [libdefaults], [realms], and [domain_realm] sections:

   Under the [libdefaults] section:

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

   Under the [realms] section:

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

   Under the [domain_realm] section:

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

The Linux VDA requires the system keytab file /etc/krb5.keytab to authenticate and register with the Delivery Controller. The previous kerberos method setting forces Winbind to create the system keytab file when the machine is first joined to the domain.

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

RHEL 8:

```
sudo realm join -U user --client-software=winbind REALM
<!--NeedCopy-->
```

RHEL 7:

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

REALM is the Kerberos realm name in uppercase, and **user** is a domain user who has permissions to add computers to the domain.

Configure PAM for Winbind By default, the configuration for the Winbind PAM module (pam_winbind) does not enable Kerberos ticket caching and home directory creation. Open **/etc/security/pam_winbind.conf** and add or change the following entries under the [Global] section:

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

Ensure that any leading semi-colons from each setting are removed. These changes require restarting the Winbind daemon:

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

> **Tip:**
>
> The `winbind` daemon stays running only if the machine is joined to a domain.

Open **/etc/krb5.conf** and change the following setting under the [libdefaults] section from KEYRING to FILE type:

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

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory.

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos configuration To ensure that Kerberos is configured correctly for use with the Linux VDA, verify that the system keytab file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:


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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Examine the account details of the machine using:

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

Verify user authentication Use the **wbinfo** tool to verify that domain users can authenticate with the domain:

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

The domain specified here is the AD domain name, not the Kerberos realm name. For the bash shell, the backslash (\) character must be escaped with another backslash. This command returns a message indicating success or failure.

To verify that the Winbind PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

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

Verify that the tickets in the Kerberos credential cache are valid and not expired:

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

Exit the session.

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Quest Authentication Services

Configure Quest on domain controller Assume that you have installed and configured the Quest software on the Active Directory domain controllers, and have been granted administrative privileges to create computer objects in Active Directory.

Enable domain users to log on to Linux VDA machines To enable domain users to establish HDX sessions on a Linux VDA machine:

1. In the Active Directory Users and Computers management console, open Active Directory user properties for that user account.
2. Select the **Unix Account** tab.
3. Check **Unix-enabled**.
4. Set the **Primary GID Number** to the group ID of an actual domain user group.

> **Note:**
>
> These instructions are equivalent for setting up domain users for logon using the console, RDP, SSH, or any other remoting protocol.

Configure Quest on Linux VDA

Work around SELinux policy enforcement The default RHEL environment has SELinux fully enforced. This enforcement interferes with the Unix domain socket IPC mechanisms used by Quest, and prevents domain users from logging on.

The convenient way to work around this issue is to disable SELinux. As a root user, edit **/etc/selinux/-config** and change the **SELinux** setting:

```
SELINUX=permissive
```

This change requires a machine restart:

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

> **Important:**
>
> Use this setting carefully. Reenabling SELinux policy enforcement after disabling can cause a complete lockout, even for the root user and other local users.

Configure VAS daemon Auto-renewal of Kerberos tickets must be enabled and disconnected. Authentication (offline logon) must be disabled.


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

sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-
    auth false
<!--NeedCopy-->
```

This command sets the renewal interval to nine hours (32,400 seconds) which is one hour less than the default 10-hour ticket lifetime. Set this parameter to a lower value on systems with a shorter ticket lifetime.

Configure PAM and NSS To enable domain user logon through HDX and other services such as su, ssh, and RDP, run the following commands to configure PAM and NSS manually:

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

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

Join Windows domain Join the Linux machine to the Active Directory domain using the Quest **vastool** command:

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

The user is any domain user who has permissions to join computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a Quest-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the domain name. If the machine is not joined to any domain, the following error appears:

```
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
```


Verify user authentication To verify that Quest can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

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

Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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

Verify that the tickets in the Kerberos credential cache are valid and not expired:

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

Exit the session.

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Centrify DirectControl

Join Windows domain With the Centrify DirectControl Agent installed, join the Linux machine to the Active Directory domain using the Centrify `adjoin` command:

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

The user parameter is any Active Directory domain user who has permissions to join computers to the Active Directory domain. The **domain-name** is the name of the domain to join the Linux machine to.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a Centrify-joined Linux machine is on the domain:

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


Verify that the Joined to domain value is valid and the CentrifyDC mode returns connected. If the mode remains stuck in the starting state, then the Centrify client is experiencing server connection or authentication problems.

More comprehensive system and diagnostic information is available using:

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

Test connectivity to the various Active Directory and Kerberos services.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

SSSD

If you are using SSSD, follow the instructions in this section. This section includes instructions for joining a Linux VDA machine to a Windows domain and provides guidance for configuring Kerberos authentication.

To set up SSSD on RHEL and CentOS, do the following:

1. Join the domain and create host keytab
2. Set up SSSD
3. Enable SSSD
4. Verify the Kerberos configuration
5. Verify user authentication

Join the domain and create host keytab SSSD does not provide Active Directory client functions for joining the domain and managing the system keytab file. You can use `adcli`, `realmd`, or `Samba` instead.

This section describes the `Samba` and the `adcli` approaches for RHEL 7 and RHEL 8 respectively. For `realmd`, see the RHEL or CentOS documentation. These steps must be followed before configuring SSSD.

- **Samba (RHEL 7):**

  Install or update the required packages:

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


On the Linux client with properly configured files:

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

Configure the machine for Samba and Kerberos authentication:

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

Where **REALM** is the Kerberos realm name in uppercase and **domain** is the short NetBIOS name of the Active Directory domain.

> **Note:**
>
> Settings in this article are meant for the single-domain, single-forest model. Configure Kerberos based on your AD infrastructure.

If DNS-based lookup of the KDC server and realm name is required, add the following two options to the preceding command:

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

Open **/etc/samba/smb.conf** and add the following entries under the **[Global]** section, but after the section generated by the **authconfig** tool:

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

Join the Windows domain. Ensure that your domain controller is reachable and you have an Active Directory user account with permissions to add computers to the domain:

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

REALM is the Kerberos realm name in uppercase and **user** is a domain user who has permissions to add computers to the domain.

- **Adcli (RHEL 8):**

  Install or update the required packages:

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

  Configure the machine for Samba and Kerberos authentication:


```
sudo authselect select sssd with-mkhomedir --force
<!--NeedCopy-->
```

Open **/etc/krb5.conf** and add the entries under the [realms] and [domain_realm] sections.

Under the [realms] section:

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

Under the [domain_realm] section:

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

Join the Windows domain. Ensure that your domain controller is reachable and you have an Active Directory user account with permissions to add computers to the domain:

```
sudo realm join REALM -U user
<!--NeedCopy-->
```

REALM is the Kerberos realm name in uppercase and **user** is a domain user who has permissions to add computers to the domain.

Set up SSSD Setting up SSSD consists of the following steps:

- Install the **sssd-ad** package on the Linux VDA by running the `sudo yum -y install sssd` command.
- Make configuration changes to various files (for example, sssd.conf).
- Start the **sssd** service.

An example **sssd.conf** configuration for RHEL 7 (extra options can be added as needed):

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

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

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

```



```
# Should be specified as the lower-case version of the long version of
    the Active Directory domain.
ad_domain = ad.example.com

# Kerberos settings
krb5_ccachedir = /tmp
krb5_ccname_template = FILE:%d/krb5cc_%U

# Uncomment if service discovery is not working
# ad_server = server.ad.example.com

# Comment out if the users have the shell and home dir set on the AD
    side
default_shell = /bin/bash
fallback_homedir = /home/%d/%u

# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
    available
# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
<!--NeedCopy-->
```

Replace **ad.example.com**, **server.ad.example.com** with the corresponding values. For more details, see sssd-ad(5) - Linux man page.

(RHEL 8 only)
Open **/etc/sssd/sssd.conf** and add the following entries under the [domain/ad.example.com] section:

```
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
```

Set the file ownership and permissions on sssd.conf:

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

Enable SSSD RHEL 8:

Run the following commands to enable SSSD:

```
sudo systemctl restart sssd
sudo systemctl enable sssd.service
sudo chkconfig sssd on
<!--NeedCopy-->
```


RHEL 7/CentOS 7:

Use `authconfig` to enable SSSD. Install **oddjob-mkhomedir** to ensure that the home directory creation is compatible with SELinux:

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

sudo service sssd start

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

Verify Kerberos configuration Verify that the system **keytab** file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos **kinit** command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign (**$**) must be escaped with a backslash (****) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Verify user authentication Use the **getent** command to verify that the logon format is supported and the NSS works:

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

The **DOMAIN** parameter indicates the short version domain name. If another logon format is needed, verify by using the **getent** command first.

The supported logon formats are:

- Down-level logon name: `DOMAIN\username`

- UPN: `username@domain.com`
- NetBIOS Suffix format: `username@DOMAIN`

To verify that the SSSD PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

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

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

Verify that a corresponding Kerberos credential cache file was created for the **uid** returned by the command:

```
ls /tmp/krb5cc_{
 uid }

<!--NeedCopy-->
```

Verify that the tickets in the user's Kerberos credential cache are valid and not expired.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

PBIS

Download the required PBIS package For RHEL 7/CentOS 7, for example:

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

For RHEL 8/CentOS8, for example:

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

Make the PBIS installation script executable For RHEL 7/CentOS 7, for example:

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

For RHEL 8/CentOS 8, for example:

```
chmod +x pbis-open-9.1.0.551.linux.x86_64.rpm.sh
<!--NeedCopy-->
```


Run the PBIS installation script For RHEL 7/CentOS 7, for example:

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

For RHEL 8/CentOS 8, for example:

```
sh pbis-open-9.1.0.551.linux.x86_64.rpm.sh
<!--NeedCopy-->
```

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

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

The **user** is a domain user who has permissions to add computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Note: To set Bash as the default shell, run the **/opt/pbis/bin/config LoginShellTemplate/bin/bash** command.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a PBIS-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the information about the currently joined AD domain and OU. Otherwise, only the host name appears.

Verify user authentication To verify that PBIS can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\user

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

Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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


Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Step 4: Install the Linux VDA

You can do a fresh installation or upgrade an existing installation from the previous two versions and from an LTSR release.

To do a fresh installation

1. (Optional) Uninstall the old version

   If you installed an earlier version other than the previous two and an LTSR release, uninstall it before installing the new version.

   a) Stop the Linux VDA services:

   ```
   sudo /sbin/service ctxvda stop

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

   > **Note:**
   >
   > Before you stop the `ctxvda` and `ctxhdx` services, run the **service ctxmonitorservice stop** command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

   b) Uninstall the package:

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

   > **Note:**
   >
   > To run a command, the full path is needed; alternately, you can add **/opt/Citrix/VDA/sbin** and **/opt/Citrix/VDA/bin** to the system path.

2. Download the Linux VDA package

   Go to the Citrix Virtual Apps and Desktops download page. Expand the appropriate version of Citrix Virtual Apps and Desktops and click **Components** to download the Linux VDA package that matches your Linux distribution.

3. Install the Linux VDA

   - Install the Linux VDA software using Yum:

     For RHEL 8/CentOS 8:

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

     For RHEL 7/CentOS 7:

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

   - Install the Linux VDA software using the RPM package manager. Before doing so, you must resolve the following dependencies:

     For RHEL 8/CentOS 8:

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

     For RHEL 7/CentOS 7:

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

     RPM dependency list for RHEL 8.2/CentOS 8.2:

     ```
     postgresql-jdbc >= 42.2.3

     postgresql-server >= 10.6

     java-11-openjdk >= 11

     icoutils >= 0.32

     firewalld  >= 0.8.0

     policycoreutils-python-utils >= 2.9

     python3-policycoreutils >= 2.9

     dbus >= 1.12.8

     dbus-common  >= 1.12.8

     dbus-daemon   >= 1.12.8

     dbus-tools  >= 1.12.8

     dbus-x11 >= 1.12.8

     ```



```
xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.4

libXpm >= 3.5.12

libXrandr >= 1.5.1

libXtst >= 1.2.3

motif >= 2.3.4

pam >= 1.3.1

util-linux >= 2.32.1

util-linux-user  >= 2.32.1

xorg-x11-utils  >= 7.5

bash >= 4.4

findutils >= 4.6

gawk >= 4.2

sed >= 4.5

cups >= 2.2

foomatic-filters >= 4.0.9

cups-filters >= 1.20.0

ghostscript  >= 9.25

libxml2 >= 2.9

libmspack >= 0.7
<!--NeedCopy-->
```

RPM dependency list for RHEL 7/CentOS 7:

```
postgresql-server >= 9.2

postgresql-jdbc >= 9.2

java-11-openjdk >= 11

ImageMagick >= 6.7.8.9

firewalld >= 0.3.9

```



```
policycoreutils-python >= 2.0.83

dbus >= 1.6.12

dbus-x11 >= 1.6.12

xorg-x11-server-utils >= 7.7

xorg-x11-xinit >= 1.3.2

libXpm >= 3.5.10

libXrandr >= 1.4.1

libXtst >= 1.2.2

motif >= 2.3.4

pam >= 1.1.8

util-linux >= 2.23.2

bash >= 4.2

findutils >= 4.5

gawk >= 4.0

sed >= 4.2

cups >= 1.6.0

foomatic-filters >= 4.0.9

openldap >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.6.0

gperftools-libs >= 2.4

rpmlib(FileDigests) <= 4.6.0-1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

pmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsXz) <= 5.2-1
```



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

> **Note:**
>
> For a matrix of the Linux distributions and the Xorg versions that this version of the Linux VDA supports, see System requirements.

> **Note:**
>
> After installing the Linux VDA on RHEL 7.x, run the `sudo yum install -y python-websockify x11vnc` command. The purpose is to install `python-websockify` and `x11vnc` manually for using the session shadowing feature. For more information, see Shadow sessions.

To upgrade an existing installation

You can upgrade an existing installation from the previous two versions and from an LTSR release.

- To upgrade your software using Yum:

  For RHEL 7/CentOS 7:

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

- To upgrade your software using the RPM package manager:

  For RHEL 7/CentOS 7:

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

> **Note:**
>
> If you are using RHEL 7, make sure to complete the following steps after you run the preceding upgrade commands:
>
> 1. run `/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\VirtualDesktopAgent"-t "REG_SZ"-v "DotNetRuntimePath"-d "/opt/rh/rh-dotnet31/root/usr/bin/"--force` to set the right .NET runtime path.
> 2. Restart the `ctxvda` service.
>
> **Important:**
>
> Restart the Linux VDA machine after upgrading the software.

Step 5: Install NVIDIA GRID drivers

Enabling HDX 3D Pro requires extra installation steps to install the requisite graphics drivers on the hypervisor and on the VDA machines.

Configure the following:

1. Citrix Hypervisor
2. VMware ESX

Follow the instructions for your chosen hypervisor.

Citrix Hypervisor:

This detailed section walks through the install and configuration of the NVIDIA GRID drivers on Citrix Hypervisor.

VMware ESX:

Follow the information contained in this guide to install and configure the NVIDIA GRID drivers for VMware ESX.

VDA machines:

Follow these steps to install and configure the drivers for each of the Linux VM guests:

1. Before starting, ensure that the Linux VM is shut down.
2. In XenCenter, add a GPU in GPU pass-through mode to the VM.
3. Start the RHEL VM.

To prepare the machine for the NVIDIA GRID drivers, run the following commands:

```
yum install gcc

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

systemctl set-default multi-user.target
<!--NeedCopy-->
```

Follow the steps in the Red Hat Enterprise Linux document to install the NVIDIA GRID driver.

> **Note:**
>
> During the GPU driver install, select the default (‘no’) for each question.
>
> **Important:**
>
> After GPU pass-through is enabled, the Linux VM is no longer accessible through XenCenter. Use SSH to connect.


```
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                                                 |
+-----------------------------------------------------------------------------+
```

Set the correct configuration for the card:

`etc/X11/ctx-nvidia.sh`

To take advantage of large resolutions and multi-monitor capabilities, you need a valid NVIDIA license. To apply for the license, follow the product documentation from "GRID Licensing Guide.pdf - DU-07757-001 September 2015."

Step 6: Configure the Linux VDA

After installing the package, you must configure the Linux VDA by running the ctxsetup.sh script. Before making any changes, the script verifies the environment and ensures that all dependencies are installed. If necessary, you can rerun the script at any time to change settings.

You can run the script manually with prompting, or automatically with preconfigured responses. Review Help about the script before proceeding:

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

Prompted configuration

Run a manual configuration with prompted questions:

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


Automated configuration

For an automated install, provide the options required by the setup script with environment variables. If all required variables are present, the script does not prompt for any information.

Supported environment variables include:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N** –The Linux VDA supports specifying a Delivery Controller name using a DNS CNAME record. Set to N by default.
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'**–The Linux VDA requires a space-separated list of Delivery Controller Fully Qualified Domain Names (FQDNs) to use for registering with a Delivery Controller. At least one FQDN or CNAME alias must be specified.
- **CTX_XDL_VDA_PORT=port-number** –The Linux VDA communicates with Delivery Controllers through a TCP/IP port, which is port 80 by default.
- **CTX_XDL_REGISTER_SERVICE=Y | N** - The Linux Virtual Desktop services are started after machine startup. The value is set to Y by default.
- **CTX_XDL_ADD_FIREWALL_RULES=Y | N** –The Linux Virtual Desktop services require incoming network connections to be allowed through the system firewall. You can open the required ports (ports 80 and 1494 by default) automatically in the system firewall for the Linux Virtual Desktop. Set to Y by default.
- **CTX_XDL_AD_INTEGRATION=1 | 2 | 3 | 4 | 5** –The Linux VDA requires Kerberos configuration settings to authenticate with the Delivery Controllers. The Kerberos configuration is determined from the installed and configured Active Directory integration tool on the system. Specify the supported Active Directory integration method to use:
  - 1 –Samba Winbind
  - 2 –Quest Authentication Services
  - 3 –Centrify DirectControl
  - 4 –SSSD
  - 5 –PBIS
- **CTX_XDL_HDX_3D_PRO=Y | N** –The Linux VDA supports HDX 3D Pro, a set of GPU acceleration technologies designed to optimize the virtualization of rich graphics applications. If HDX 3D Pro is selected, the VDA is configured for VDI desktops (single-session) mode - (that is, CTX_XDL_VDI_MODE=Y).
- **CTX_XDL_VDI_MODE=Y | N** –Whether to configure the machine as a dedicated desktop delivery model (VDI) or hosted shared desktop delivery model. For HDX 3D Pro environments, set this variable to Y. This variable is set to N by default.

- **CTX_XDL_SITE_NAME=dns-name** –The Linux VDA discovers LDAP servers through DNS. To limit the DNS search results to a local site, specify a DNS site name. This variable is set to **<none>** by default.
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** –The Linux VDA queries DNS to discover LDAP servers. If DNS cannot provide LDAP service records, you can provide a space-separated list of LDAP FQDNs with LDAP ports. For example, ad1.mycompany.com:389. This variable is set to **<none>** by default.
- **CTX_XDL_SEARCH_BASE=search-base-set** –The Linux VDA queries LDAP through a search base set to the root of the Active Directory Domain (for example, DC=mycompany,DC=com). To improve search performance, you can specify a search base (for example, OU=VDI,DC=mycompany,DC=com). This variable is set to **<none>** by default.
- **CTX_XDL_FAS_LIST='list-fas-servers'**–The Federated Authentication Service (FAS) servers are configured through the AD Group Policy. The Linux VDA does not support AD Group Policy, but you can provide a semicolon-separated list of FAS servers instead. The sequence must be the same as configured in the AD Group Policy. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** –The path to install .NET Core Runtime 3.1 for supporting the new broker agent service (`ctxvda`). The default path is /usr/bin.
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** –Specifies the GNOME or MATE desktop environment to use in sessions. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set the variable value to **mate**.

  Note:

  You can also change the desktop environment for a target session user by completing the following steps:

  1. Create a `.xsession` file under the **$HOME/<username>** directory on the VDA.
  2. Edit the `.xsession` file to specify a desktop environment based on distributions.

     For MATE desktop on CentOS, Ubuntu, and Debian

     ```
     MSESSION="$(type -p mate-session)"
     if [ -n "$MSESSION"]; then
     exec mate-session
     fi
     ```

     For GNOME desktop on CentOS

GSESSION=”$(type -p gnome-session)”
if [-n “$GSESSION”]; then

```
    export GNOME_SHELL_SESSION_MODE=classic
    exec gnome-session --session=gnome-classic      fi
         **For GNOME desktop on Ubuntu and Debian**
```

GSESSION=”$(type -p gnome-session)”
if [-n “$GSESSION”]; then

```
    exec gnome-session      fi
```

3. Share the 700 file permission with the target session user.

- **CTX_XDL_START_SERVICE=Y | N** –Whether or not the Linux VDA services are started when the Linux VDA configuration is complete. Set to Y by default.
- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The default port is 7503.
- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

Set the environment variable and run the configure script:

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST= ‘ list-ddc-fqdns ’

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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



```
export CTX_XDL_DESKTOP_ENVIRONMENT= gnome | mate | '<none>'

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

When running the sudo command, type the **-E** option to pass the existing environment variables to the new shell it creates. We recommend that you create a shell script file from the preceding commands with **#!/bin/bash** as the first line.

Alternatively, you can specify all parameters by using a single command:

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST= ' list-ddc-fqdns '  \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST= ' list-ldap-servers '  \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST= ' list-fas-servers '  \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

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



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

Remove configuration changes

In some scenarios, you might have to remove the configuration changes made by the **ctxsetup.sh** script without uninstalling the Linux VDA package.

Review Help about this script before proceeding:

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

To remove configuration changes:

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

> **Important:**
>
> This script deletes all configuration data from the database and renders the Linux VDA inoperable.

Configuration logs

The **ctxsetup.sh** and **ctxcleanup.sh** scripts display errors on the console, with additional information written to the configuration log file **/tmp/xdl.configure.log**.

Restart the Linux VDA services to have the changes take effect.

Step 7: Run the Linux VDA

After configuring the Linux VDA by using the **ctxsetup.sh** script, you can run the following commands to control the Linux VDA.

Start the Linux VDA:

To start the Linux VDA services:

```
sudo /sbin/service ctxhdx start

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

Stop the Linux VDA:

To stop the Linux VDA services:


```
sudo /sbin/service ctxvda stop

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

> **Note:**
>
> Before you stop the `ctxvda` and `ctxhdx` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

Restart the Linux VDA:

To restart the Linux VDA services:

```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx restart

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

Check the status of the Linux VDA:

To check the running status of the Linux VDA services:

```
sudo /sbin/service ctxvda status

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

Step 8: Create the machine catalog in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating machine catalogs and adding Linux VDA machines is similar to the traditional Windows VDA approach. For a more detailed description of how to complete these tasks, see Create machine catalogs and Manage machine catalogs.

For creating machine catalogs that contain Linux VDA machines, there are a few restrictions that differentiate the process from creating machine catalogs for Windows VDA machines:

- For the operating system, select:
  - The **Multi-session OS** option for a hosted shared desktops delivery model.
  - The **Single-session OS** option for a VDI dedicated desktop delivery model.
- Do not mix Linux and Windows VDA machines in the same machine catalog.

> **Note:**
>
> Early versions of Citrix Studio did not support the notion of a "Linux OS."However, selecting the **Windows Server OS** or **Server OS** option implies an equivalent hosted shared desktops delivery model. Selecting the **Windows Desktop OS** or **Desktop OS** option implies a single user per machine delivery model.
>
> **Tip:**
>
> If you remove and rejoin a machine to the Active Directory domain, you must remove and add the machine to the machine catalog again.

Step 9: Create the delivery group in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating a delivery group and adding machine catalogs containing Linux VDA machines is almost identical to Windows VDA machines. For a more detailed description of how to complete these tasks, see Create Delivery Groups.

For creating delivery groups that contain Linux VDA machine catalogs, the following restrictions apply:

- Ensure that the AD users and groups you select have been properly configured to log on to the Linux VDA machines.
- Do not allow logon of unauthenticated (anonymous) users.
- Do not mix the delivery group with machine catalogs that contain Windows machines.

> **Important:**
>
> Publishing applications is supported with Linux VDA Version 1.4 and later. However, the Linux VDA does not support the delivery of desktops and apps to the same machine.

For information about how to create machine catalogs and delivery groups, see Citrix Virtual Apps and Desktops 7 2103.

Install Linux Virtual Delivery Agent for SUSE

June 10, 2022

You can choose to follow the steps in this article for manual installation or use easy install for automatic installation and configuration. Easy install saves time and labor and is less error-prone than the manual installation.

> **Note:**
>
> Use easy install only for fresh installations. Do not use easy install to update an existing installation.

Step 1: Prepare for installation

Step 1a: Launch the YaST tool

The SUSE Linux Enterprise YaST tool is used for configuring all aspects of the operating system.

To launch the text-based YaST tool:

```
su -

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

Alternatively, launch the UI-based YaST tool:

```
su -

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

Step 1b: Configure networking

The following sections provide information on configuring the various networking settings and services used by the Linux VDA. Configuring networking is carried out via the YaST tool, not via other methods such as Network Manager. These instructions are based on using the UI-based YaST tool. The text-based YaST tool can be used but has a different method of navigation that is not documented here.

Configure host name and DNS

1. Open YaST Network Settings.
2. SLED 12 Only: On the **Global Options** tab, change the **Network Setup Method** to **Wicked Service**.
3. Open the **Hostname/DNS** tab.
4. Clear **Change hostname via DHCP**.
5. Check **Assign Hostname to Loopback IP**.
6. Edit the following to reflect your networking setup:

    - Host name –Add the DNS host name of the machine.

- Domain name –Add the DNS domain name of the machine.
- Name server –Add the IP address of the DNS server. It is typically the IP address of the AD Domain Controller.
- Domain search list –Add the DNS domain name.

> **Note:**
>
> The Linux VDA currently does not support NetBIOS name truncation. Therefore, the host name must not exceed 15 characters.
>
> **Tip:**
>
> Use a–z, A–Z, 0–9, and hyphen (-) characters only. Avoid underscores (_), spaces, and other symbols. Do not start a host name with a number and do not end with a hyphen. This rule also applies to Delivery Controller host names.

Disable multicast DNS On SLED only, the default settings have multicast DNS (mDNS) enabled, which can lead to inconsistent name resolution results. mDNS is not enabled on SLES by default, so no action is required.

To disable mDNS, edit **/etc/nsswitch.conf** and change the line containing:

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

To:

```
hosts: files dns
```

Check the host name Verify that the host name is set correctly:

```
1 hostname
2 <!--NeedCopy-->
```

This command returns only the machine's host name and not its fully qualified domain name (FQDN).

Verify that the FQDN is set correctly:

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

This command returns the machine's FQDN.

Check name resolution and service reachability Verify that you can resolve the FQDN and ping the domain controller and Delivery Controller:


```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

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

If you cannot resolve the FQDN or ping either of these machines, review the steps before proceeding.

Step 1c: Configure the NTP service

It is crucial to maintain accurate clock synchronization between the VDAs, Delivery Controllers, and domain controllers. Hosting the Linux VDA as a virtual machine can cause clock skew problems. For this reason, maintaining time using a remote NTP service is preferred. Some changes might be required to the default NTP settings:

1. Open YaST NTP Configuration and select the **General Settings** tab.
2. In the Start NTP Daemon section, check **Now and on Boot**.
3. If present, select the **Undisciplined Local Clock (LOCAL)** item and click **Delete**.
4. Add an entry for an NTP server by clicking **Add**.
5. Select the **Server Type** and click **Next**.
6. Type the DNS name of the NTP server in the Address field. This service is normally hosted on the Active Directory domain controller.
7. Leave the Options field unchanged.
8. Click **Test** to verify that the NTP service is reachable.
9. Click **OK** through the set of windows to save the changes.

> **Note:**
>
> For SLES 12 implementations, the NTP daemon might fail to start due to a known SUSE issue with AppArmor policies. Follow the resolution for additional information.

Step 1d: Install Linux VDA dependent packages

The Linux VDA software for SUSE Linux Enterprise depends on the following packages:

- Postgresql10-server 10.12 or later
- OpenJDK 11
- OpenMotif Runtime Environment 2.3.1 or later
- Cups 1.6.0 or later

- Foomatic filters 3.0.0 or later
- ImageMagick 6.8 or later

Add repositories You can obtain some required packages, such as PostgreSQL and ImageMagick, from the SUSE Linux Enterprise Software Development Kit (SDK). To obtain the packages, add the SDK repository by using YaST or download the SDK image file and then mount it locally by using the following commands:

```
sudo mkdir -p /mnt/sdk

sudo mount -t iso9660 path-to-iso/SLE-12-SP5-SDK-DVD-x86_64-GM-DVD1.iso
    /mnt/sdk

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

Install the Kerberos client Install the Kerberos client for mutual authentication between the Linux VDA and the Delivery Controllers:

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

The Kerberos client configuration depends on which Active Directory integration approach is used. See the following description.

Install OpenJDK 11 The Linux VDA requires the presence of OpenJDK 11.

> **Tip:**
>
> To avoid registration failure with the Delivery Controller, ensure that you installed only OpenJDK 11. Remove all other versions of Java from your system.

- **SLED:**

1. On SLED, check whether OpenJDK 11 has been installed:

   ```
   sudo zypper info java-11-openjdk
   <!--NeedCopy-->
   ```

2. Update to OpenJDK 11 if the status is reported as out-of-date:

   ```
   sudo zypper update java-11-openjdk
   <!--NeedCopy-->
   ```

3. Check the Java version:


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

- **SLES:**

1. On SLES, install OpenJDK 11:

```
sudo zypper install java-11-openjdk
<!--NeedCopy-->
```

2. Check the Java version:

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

Install PostgreSQL On SLED/SLES 12, install the packages:

```
sudo zypper install postgresql-init

sudo zypper install postgresql10-server

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

Post-installation steps are required to initialize the database service and to ensure that PostgreSQL is started upon machine startup:

```
sudo systemctl enable postgresql

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

Database files locate at /var/lib/pgsql/data.

Remove repositories With dependent packages installed, run the following commands to remove the SDK repository that was set up earlier and the media mounted:

```
sudo zypper rr sdk

sudo umount /mnt/sdk

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


Step 2: Prepare Linux VM for Hypervisor

Some changes are required when running the Linux VDA as a virtual machine on a supported hypervisor. Make the following changes according to the hypervisor platform in use. No changes are required if you are running the Linux machine on bare metal hardware.

Fix time synchronization on Citrix Hypervisor

If the Citrix Hypervisor Time Sync feature is enabled, within each paravirtualized Linux VM you experience issues with NTP and Citrix Hypervisor both trying to manage the system clock. To avoid the clock becoming out of sync with other servers, the system clock within each Linux guest must be synchronized with NTP. This case requires disabling host time synchronization. No changes are required in HVM mode.

On some Linux distributions, if you are running a paravirtualized Linux kernel with Citrix VM Tools installed, you can check whether the Citrix Hypervisor Time Sync feature is present and enabled from within the Linux VM:

```
su -

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

This command returns 0 or 1:

- 0 - The time sync feature is enabled, and must be disabled.
- 1 - The time sync feature is disabled, and no further action is required.

If the **/proc/sys/xen/independent_wallclock** file is not present, the following steps are not required.

If enabled, disable the time sync feature by writing **1** to the file:

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

To make this change permanent and persistent after restart, edit the **/etc/sysctl.conf** file and add the line:

```
xen.independent_wallclock = 1
```

To verify these changes, restart the system:

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

After restart, verify that the setting is correct:


```
su -

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

This command returns the value 1.

Fix time synchronization on Microsoft Hyper-V

Linux VMs with Hyper-V Linux Integration Services installed can apply the Hyper-V time synchronization feature to use the host operating system's time. To ensure that the system clock remains accurate, enable this feature alongside the NTP services.

From the management operating system:

1. Open the Hyper-V Manager console.
2. For the settings of a Linux VM, select **Integration Services**.
3. Ensure that **Time synchronization** is selected.

> **Note:**
>
> This approach is different from VMware and Citrix Hypervisor, where host time synchronization is disabled to avoid conflicts with NTP. Hyper-V time synchronization can coexist and supplement NTP time synchronization.

Fix time synchronization on ESX and ESXi

If the VMware Time Synchronization feature is enabled, within each paravirtualized Linux VM you experience issues with NTP and the hypervisor both trying to synchronize the system clock. To avoid the clock becoming out of sync with other servers, synchronize the system clock within each Linux guest with NTP. This case requires disabling host time synchronization.

If you are running a paravirtualized Linux kernel with VMware Tools installed:

1. Open the vSphere Client.
2. Edit settings for the Linux VM.
3. In the **Virtual Machine Properties** dialog, open the **Options** tab.
4. Select **VMware Tools**.
5. In the **Advanced** box, clear **Synchronize guest time with host**.

Step 3: Add the Linux virtual machine (VM) to the Windows domain

The Linux VDA supports several methods for adding Linux machines to the Active Directory (AD) domain:

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

Follow instructions based on your chosen method.

> **Note:**
>
> Session launches might fail when the same user name is used for the local account in the Linux VDA and the account in AD.

Samba Winbind

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add machines to the domain:

1. Open YaST Windows Domain Membership.
2. Make the following changes:
   - Set the **Domain or Workgroup** to the name of your Active Directory domain or the IP address of the domain controller. Ensure that the domain name is in uppercase.
   - Check **Also Use SMB information for Linux Authentication**.
   - Check **Create Home Directory on Login**.
   - Check **Single Sign-on for SSH**.
   - Ensure that **Offline Authentication** is not checked. This option is not compatible with the Linux VDA.
3. Click **OK**. If prompted to install some packages, click **Install**.
4. If a domain controller is found, it asks whether you want to join the domain. Click **Yes**.
5. When prompted, type the credentials of a domain user with permission to add computers to the domain and click **OK**.
6. A message indicating success is displayed.
7. If prompted to install some samba and krb5 packages, click **Install**.

YaST might have indicated that these changes require some services or the machine to be restarted. We recommend you restart the machine:

```
su -

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

SUSE 12 Only: Patch Kerberos credential cache name SUSE 12 has changed the default Kerberos credential cache name specification from the usual **FILE:/tmp/krb5cc_%{uid}** to **DIR:/run/user/%{uid}/krb5cc**. This new DIR caching method is not compatible with the Linux VDA and must be changed manually. As a root user, edit **/etc/krb5.conf** and add the following setting under the **[libdefaults]** section if not set:

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

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory.

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos configuration To ensure that Kerberos is configured correctly for use with the Linux VDA, verify that the system keytab file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is

different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Examine the machine account details using:

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

Verify user authentication Use the `wbinfo` tool to verify that domain users can authenticate with the domain:

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

The domain specified here is the AD domain name, not the Kerberos realm name. For the bash shell, the backslash (\) character must be escaped with another backslash. This command returns a message indicating success or failure.

To verify that the Winbind PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

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

Verify that a corresponding Kerberos credential cache file was created for the uid returned by the **id -u** command:

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

Verify that the tickets in the user's Kerberos credential cache are valid and not expired:

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

Exit the session.

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Quest authentication service

Configure Quest on domain controller Assume that you have installed and configured the Quest software on the Active Directory domain controllers, and have been granted administrative privileges to create computer objects in Active Directory.

Enable domain users to log on to Linux VDA machines To enable domain users to establish HDX sessions on a Linux VDA machine:

1. In the Active Directory Users and Computers management console, open Active Directory user properties for that user account.
2. Select the **Unix Account** tab.
3. Check **Unix-enabled**.
4. Set the **Primary GID Number** to the group ID of an actual domain user group.

> **Note:**
>
> These instructions are equivalent for setting up domain users for logon using the console, RDP, SSH, or any other remoting protocol.

Configure Quest on Linux VDA

Configure VAS daemon Auto-renewal of Kerberos tickets must be enabled and disconnected. Authentication (offline logon) must be disabled:

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

sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-
   auth false
<!--NeedCopy-->
```

This command sets the renewal interval to nine hours (32,400 seconds) which is one hour less than the default 10-hour ticket lifetime. Set this parameter to a lower value on systems with a shorter ticket lifetime.

Configure PAM and NSS To enable domain user logon through HDX and other services such as su, ssh, and RDP, run the following commands to configure PAM and NSS manually:

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

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


Join Windows domain Join the Linux machine to the Active Directory domain using the Quest `vastool` command:

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

The **user** is any domain user who has permissions to join computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a Quest-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the domain name. If the machine is not joined to any domain, the following error appears:

```
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
```

Verify user authentication To verify that Quest can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

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

Verify that a corresponding Kerberos credential cache file was created for the uid returned by the **id -u** command:

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

Verify that the tickets in the Kerberos credential cache are valid and not expired:

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

Exit the session.

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


A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Centrify DirectControl

Join Windows domain With the Centrify DirectControl Agent installed, join the Linux machine to the Active Directory domain using the Centrify **adjoin** command:

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

The **user** is any Active Directory domain user who has permissions to join computers to the Active Directory domain. The **domain-name** is the name of the domain to join the Linux machine to.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a Centrify-joined Linux machine is on the domain:

```
su -

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

Verify that the **Joined to domain** value is valid and the **CentrifyDC mode** returns **connected**. If the mode remains stuck in the starting state, then the Centrify client is experiencing server connection or authentication problems.

More comprehensive system and diagnostic information is available using:

```
adinfo --sysinfo all

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

Test connectivity to the various Active Directory and Kerberos services.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

SSSD

If you are using SSSD on SUSE, follow the instructions in this section. This section includes instructions for joining a Linux VDA machine to a Windows domain and provides guidance for configuring Kerberos

authentication.

To set up SSSD on SUSE, complete the following steps:

1. Join the domain and create host keytabs
2. Configure PAM for SSSD
3. Set up SSSD
4. Enable SSSD
5. Verify domain membership
6. Verify the Kerberos configuration
7. Verify user authentication

Join the domain and create host keytab SSSD does not provide Active Directory client functions for joining the domain and managing the system keytab file. You can use the Samba approach instead. Complete the following steps before configuring SSSD.

1. Stop and disable the Name Service Cache Daemon (NSCD) daemon.

```
sudo systemctl stop nscd

sudo systemctl disable nscd
<!--NeedCopy-->
```

2. Install or update the required packages:

```
sudo zypper install krb5-client

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

3. Edit /etc/krb5.conf file as a root user to permit the `kinit` utility to communicate with the target domain. Add the following entries under the **[libdefaults]**, **[realms]**, and **[domain_realm]** sections:

> **Note:**
>
> Configure Kerberos based on your AD infrastructure. The following settings are meant for the single-domain, single-forest model.

```
[libdefaults]

    dns_canonicalize_hostname = false

    rdns = false

    default_realm = REALM

    forwardable = true
```



```
10
11 [realms]
12
13     REALM = {
14
15
16          kdc = fqdn-of-domain-controller
17
18         default_domain = realm
19
20         admin_server =     fqdn-of-domain-controller
21       }
22
23 [domain_realm]
24
25     .realm = REALM
26
27     realm = REALM
28 <!--NeedCopy-->
```

realm is the Kerberos realm name, such as example.com. **REALM** is the Kerberos realm name in uppercase, such as EXAMPLE.COM. **fqdn-of-domain-controller** is the FQDN of the domain controller.

4. Edit /etc/samba/smb.conf as a root user to permit the **net** utility to communicate with the target domain. Add the following entries under the **[global]** section:

```
1  [global]
2      workgroup = domain
3
4      realm = REALM
5
6      security = ADS
7
8      kerberos method = secrets and keytab
9
10     client signing = yes
11
12     client use spnego = yes
13 <!--NeedCopy-->
```

domain is the short NetBIOS name of an Active Directory domain, such as EXAMPLE.

5. Modify the **passwd** and **group** entries in the /etc/nsswitch.conf file to reference SSSD when resolving users and groups.

```
1 passwd: compat sss
2
3 group: compat sss
4 <!--NeedCopy-->
```

6. Join the Windows domain. Ensure that your domain controller is reachable and you have an

Active Directory user account with permissions to add computers to the domain:

```
sudo realm join REALM -U user
<!--NeedCopy-->
```

user is a domain user who has permissions to add computers to the domain.

Configure PAM for SSSD Before configuring PAM for SSSD, install or update the required packages:

```
sudo zypper install sssd sssd-ad
<!--NeedCopy-->
```

Configure the PAM module for user authentication through SSSD and create home directories for user logons.

```
sudo pam-config --add  --sss
sudo pam-config --add --mkhomedir
<!--NeedCopy-->
```

Set up SSSD

1. Edit /etc/sssd/sssd.conf as a root user to permit the SSSD daemon to communicate with the target domain. An example sssd.conf configuration (extra options can be added as needed):

```
[sssd]
    config_file_version = 2
    services = nss,pam
    domains = domain-dns-name

[domain/domain-dns-name]
    id_provider = ad
    auth_provider = ad
    access_provider = ad
    ad_domain = domain-dns-name
    ad_server = fqdn-of-domain-controller
    ldap_id_mapping = true
    ldap_schema = ad

# Kerberos settings
    krb5_ccachedir = /tmp
    krb5_ccname_template = FILE:%d/krb5cc_%U

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

    fallback_homedir = /home/%d/%u
    default_shell = /bin/bash

```



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

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

    ad_gpo_access_control = permissive

<!--NeedCopy-->
```

domain-dns-name is the DNS domain name, such as example.com.

> **Note:**
>
> **ldap_id_mapping** is set to true so that SSSD itself takes care of mapping Windows SIDs to Unix UIDs. Otherwise, the Active Directory must be able to provide POSIX extensions. **ad_gpo_access_control** is set to **permissive** to prevent an invalid logon error for Linux sessions. See the man pages for sssd.conf and sssd-ad.

2. Set the file ownership and permissions on sssd.conf:

```
sudo chmod 0600 /etc/sssd/sssd.conf
<!--NeedCopy-->
```

Enable SSSD Run the following commands to enable and start the SSSD daemon at system startup:

```
sudo systemctl enable sssd
sudo systemctl start sssd
<!--NeedCopy-->
```

Verify domain membership

1. Run the net ads command of Samba to verify that the machine is joined to a domain:

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

2. Run the following command to verify extra domain and computer object information:

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

Verify Kerberos configuration To verify that Kerberos is configured correctly for use with the Linux VDA, verify that the system keytab file has been created and contains valid keys:

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


This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign (**$**) must be escaped with a backslash (****) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Verify user authentication SSSD does not provide a command-line tool for testing authentication directly with the daemon, and can only be done via PAM.

To verify that the SSSD PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

id -u

klist

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

Verify that the Kerberos tickets returned by the `klist` command are correct for that user and have not expired.

As a root user, verify that a corresponding ticket cache file was created for the uid returned by the previous `id -u` command:

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

PBIS

Download the required PBIS package For example:


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

Make the PBIS installation script executable For example:

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

Run the PBIS installation script For example:

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

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

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

The **user** is a domain user who has permissions to add computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Note: To set Bash as the default shell, run the **/opt/pbis/bin/config LoginShellTemplate/bin/bash** command.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a PBIS-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the information about the currently joined AD domain and OU. Otherwise, only the host name appears.

Verify user authentication To verify that PBIS can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\user

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


Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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

Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Step 4: Install the Linux VDA

Step 4a: Uninstall the old version

If you installed an earlier version other than the previous two and an LTSR release, uninstall it before installing the new version.

1. Stop the Linux VDA services:

   ```
   sudo /sbin/service ctxvda stop

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

   > **Note:**
   >
   > Before you stop the `ctxvda` and `ctxhdx` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

2. Uninstall the package:

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

> **Important:**
>
> Upgrading from the latest two versions is supported.
>
> **Note:**
>
> Installation components are located in **/opt/Citrix/VDA/**.
>
> To run a command, the full path is needed; alternatively, you can add **/opt/Citrix/VDA/sbin** and **/opt/Citrix/VDA/bin** to the system path.

Step 4b: Download the Linux VDA package

Go to the Citrix Virtual Apps and Desktops download page. Expand the appropriate version of Citrix Virtual Apps and Desktops and click **Components** to download the Linux VDA package that matches your Linux distribution.

Step 4c: Install the Linux VDA

Install the Linux VDA software using Zypper:

For SUSE 12:

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

Install the Linux VDA software using the RPM package manager. Before doing so, resolve the following dependencies:

For SUSE 12:

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

Step 4d: Upgrade the Linux VDA (optional)

You can upgrade an existing installation from the previous two versions and from an LTSR release.

For SUSE 12:

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

RPM Dependency list for SUSE 12:

```
postgresql-server >= 9.3

postgresql-jdbc >= 9.2

java-11-openjdk >= 11

ImageMagick >= 6.8

dbus-1 >= 1.8.8

dbus-1-x11 >= 1.8.8

libXpm4 >= 3.5.11

```



```
libXrandr2 >= 1.4.2

libXtst6 >= 1.2.2

motif >= 2.3

pam >= 1.1.8

bash >= 4.2

findutils >= 4.5

gawk >= 4.1

sed >= 4.2

cups >= 1.6.0

cups-filters-foomatic-rip >= 1.0.0

openldap2 >= 2.4

cyrus-sasl >= 2.1

cyrus-sasl-gssapi >= 2.1

libxml2 >= 2.9

python-requests >= 2.8.1

rpmlib(PayloadFilesHavePrefix) <= 4.0-1

rpmlib(CompressedFileNames) <= 3.0.4-1

rpmlib(PayloadIsLzma) <= 4.4.6-1

libtcmalloc4 >= 2.5

libcap-progs >= 2.22

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

ibus >= 1.5

xorg- x11-server = 7.6_1.19.6

xorg-x11 = 7.6_1

postgresql10-server >= 10.12

libgtk-2_0-0 >= 2.24

libgthread-2_0-0 >= 2.48
```



```
68
69 pulseaudio-utils >= 5.0
70
71 lsb-release >= 2.0
72 <!--NeedCopy-->
```

> **Important:**
>
> Restart the Linux VDA machine after upgrading.

Step 5: Configure the Linux VDA

After installing the package, you must configure the Linux VDA by running the ctxsetup.sh script. Before making any changes, the script verifies the environment and ensures that all dependencies are installed. If necessary, you can rerun the script at any time to change settings.

You can run the script manually with prompting, or automatically with preconfigured responses. Review Help about the script before proceeding:

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

Prompted configuration

Run a manual configuration with prompted questions:

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

Automated configuration

For an automated installation, provide the options required by the setup script with environment variables. If all required variables are present, the script does not prompt for any information.

Supported environment variables include:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N** –The Linux VDA supports specifying a Delivery Controller name using a DNS CNAME record. Set to N by default.
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'**–The Linux VDA requires a space-separated list of Delivery Controller Fully Qualified Domain Names (FQDNs) to use for registering with a Delivery Controller. At least one FQDN or CNAME alias must be specified.
- **CTX_XDL_VDA_PORT=port-number** –The Linux VDA communicates with Delivery Controllers through a TCP/IP port, which is port 80 by default.

- **CTX_XDL_REGISTER_SERVICE=Y | N** - The Linux Virtual Desktop services are started after machine startup. The value is set to Y by default.
- **CTX_XDL_ADD_FIREWALL_RULES=Y | N** –The Linux Virtual Desktop services require incoming network connections to be allowed through the system firewall. You can open the required ports (ports 80 and 1494 by default) automatically in the system firewall for the Linux Virtual Desktop. Set to Y by default.
- **CTX_XDL_AD_INTEGRATION=1 | 2 | 3 | 4** –The Linux VDA requires Kerberos configuration settings to authenticate with the Delivery Controllers. The Kerberos configuration is determined from the installed and configured Active Directory integration tool on the system. Specify the supported Active Directory integration method to use:
  - 1 –Samba Winbind
  - 2 –Quest Authentication Service
  - 3 –Centrify DirectControl
  - 4 –SSSD
- **CTX_XDL_HDX_3D_PRO=Y | N** –The Linux VDA supports HDX 3D Pro, a set of GPU acceleration technologies designed to optimize the virtualization of rich graphics applications. If HDX 3D Pro is selected, the VDA is configured for VDI desktops (single-session) mode - (that is, CTX_XDL_VDI_MODE=Y).
- **CTX_XDL_VDI_MODE=Y | N** –Whether to configure the machine as a dedicated desktop delivery model (VDI) or hosted shared desktop delivery model. For HDX 3D Pro environments, set this variable to Y. This variable is set to N by default.
- **CTX_XDL_SITE_NAME=dns-name** –The Linux VDA discovers LDAP servers through DNS. To limit the DNS search results to a local site, specify a DNS site name. This variable is set to **<none>** by default.
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** –The Linux VDA queries DNS to discover LDAP servers. If DNS cannot provide LDAP service records, you can provide a space-separated list of LDAP FQDNs with LDAP ports. For example, ad1.mycompany.com:389. This variable is set to **<none>** by default.
- **CTX_XDL_SEARCH_BASE=search-base-set** –The Linux VDA queries LDAP through a search base set to the root of the Active Directory Domain (for example, DC=mycompany,DC=com). To improve search performance, you can specify a search base (for example, OU=VDI,DC=mycompany,DC=com). This variable is set to **<none>** by default.
- **CTX_XDL_FAS_LIST='list-fas-servers'**–The Federated Authentication Service (FAS) servers are configured through AD Group Policy. The Linux VDA does not support AD Group Policy, but you can provide a semicolon-separated list of FAS servers instead. The sequence must be the same

as configured in AD Group Policy. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.

- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** –The path to install .NET Core Runtime 3.1 for supporting the new broker agent service (`ctxvda`). The default path is /usr/bin.
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** –Specifies the GNOME or MATE desktop environment to use in sessions. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set the variable value to **mate**.

  > **Note:**
  >
  > You can also change the desktop environment for a target session user by completing the following steps:
  >
  > 1. Create a `.xsession` file under the **$HOME/<username>** directory on the VDA.
  > 2. Edit the `.xsession` file to specify a desktop environment based on distributions.
  >
  > **For MATE desktop on CentOS, Ubuntu, and Debian**
  >
  > MSESSION="$(type -p mate-session)"
  > if [-n "$MSESSION"]; then
  > exec mate-session
  > fi
  >
  > **For GNOME desktop on CentOS**
  >
  > GSESSION="$(type -p gnome-session)"
  > if [-n "$GSESSION"]; then
  >
  > ```
  > 1     export GNOME_SHELL_SESSION_MODE=classic
  > 2     exec gnome-session --session=gnome-classic      fi
  >           **For GNOME desktop on Ubuntu and Debian**
  > ```
  >
  > GSESSION="$(type -p gnome-session)"
  > if [-n "$GSESSION"]; then
  >
  > ```
  > 1     exec gnome-session      fi
  > ```
  >
  > 3. Share the 700 file permission with the target session user.

- **CTX_XDL_START_SERVICE=Y | N** –Whether or not the Linux VDA services are started when the Linux VDA configuration is complete. Set to Y by default.
- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The de-

fault port is 7503.

- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

Set the environment variable and run the configure script:

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST=‘list-ddc-fqdns’

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

When running the sudo command, type the **-E** option to pass the existing environment variables to the new shell it creates. We recommend that you create a shell script file from the preceding commands with **#!/bin/bash** as the first line.

Alternatively, you can specify all parameters by using a single command:

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST=‘list-ddc-fqdns’ \
```



```
4
5  CTX_XDL_VDA_PORT=port-number \
6
7  CTX_XDL_REGISTER_SERVICE=Y|N \
8
9  CTX_XDL_ADD_FIREWALL_RULES=Y|N \
10
11 CTX_XDL_AD_INTEGRATION=1|2|3|4 \
12
13 CTX_XDL_HDX_3D_PRO=Y|N \
14
15 CTX_XDL_VDI_MODE=Y|N \
16
17 CTX_XDL_SITE_NAME=dns-name \
18
19 CTX_XDL_LDAP_LIST=‘list-ldap-servers’ \
20
21 CTX_XDL_SEARCH_BASE=search-base-set \
22
23 CTX_XDL_FAS_LIST=‘list-fas-servers’ \
24
25 CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \
26
27 CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \
28
29 CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \
30
31 CTX_XDL_TELEMETRY_PORT=port-number \
32
33 CTX_XDL_START_SERVICE=Y|N \
34
35 /opt/Citrix/VDA/sbin/ctxsetup.sh
36 <!--NeedCopy-->
```

Remove configuration changes

In some scenarios, you might have to remove the configuration changes made by the **ctxsetup.sh** script without uninstalling the Linux VDA package.

Review Help about this script before proceeding:

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

To remove configuration changes:

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


> **Important:**
>
> This script deletes all configuration data from the database and renders the Linux VDA inoperable.

Configuration logs

The **ctxsetup.sh** and **ctxcleanup.sh** scripts display errors on the console, with additional information written to a configuration log file:

`/tmp/xdl.configure.log`

Restart the Linux VDA services to have the changes take effect.

Step 6: Run the Linux VDA

After configuring the Linux VDA by using the **ctxsetup.sh** script, you can run the following commands to control the Linux VDA.

Start the Linux VDA:

To start the Linux VDA services:

```
sudo /sbin/service ctxhdx start

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

Stop the Linux VDA:

To stop the Linux VDA services:

```
sudo /sbin/service ctxvda stop

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

> **Note:**
>
> Before you stop the `ctxvda` and `ctxhdx` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

Restart the Linux VDA:

To restart the Linux VDA services:


```
sudo /sbin/service ctxvda stop

sudo /sbin/service ctxhdx restart

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

Check the Linux VDA status:

To check the running status of the Linux VDA services:

```
sudo /sbin/service ctxvda status

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

Step 7: Create the machine catalog in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating machine catalogs and adding Linux VDA machines is similar to the traditional Windows VDA approach. For a more detailed description of how to complete these tasks, see Create machine catalogs and Manage machine catalogs.

For creating machine catalogs that contain Linux VDA machines, there are a few restrictions that differentiate the process from creating machine catalogs for Windows VDA machines:

- For the operating system, select:
  - The **Multi-session OS** option for a hosted shared desktops delivery model.
  - The **Single-session OS** option for a VDI dedicated desktop delivery model.
- Do not mix Linux and Windows VDA machines in the same machine catalog.

> **Note:**
>
> Early versions of Citrix Studio did not support the notion of a "Linux OS."However, selecting the **Windows Server OS** or **Server OS** option implies an equivalent hosted shared desktops delivery model. Selecting the **Windows Desktop OS** or **Desktop OS** option implies a single user per machine delivery model.
>
> **Tip:**
>
> If you remove and rejoin a machine to the Active Directory domain, you must remove and add the machine to the machine catalog again.

Step 8: Create the delivery group in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating a delivery group and adding machine catalogs containing Linux VDA machines is almost identical to Windows VDA machines. For a more detailed description of how to complete these tasks, see Create Delivery Groups.

For creating delivery groups that contain Linux VDA machine catalogs, the following restrictions apply:

- Ensure that the AD users and groups you select have been properly configured to log on to the Linux VDA machines.
- Do not allow logon of unauthenticated (anonymous) users.
- Do not mix the delivery group with machine catalogs that contain Windows machines.

> **Important:**
>
> Publishing applications is supported with Linux VDA Version 1.4 and later. However, the Linux VDA does not support the delivery of desktops and apps to the same machine.

For information about how to create machine catalogs and delivery groups, see Citrix Virtual Apps and Desktops 7 2103.

Install Linux Virtual Delivery Agent for Ubuntu

June 10, 2022

You can choose to follow the steps in this article for manual installation or use easy install for automatic installation and configuration. Easy install saves time and labor and is less error-prone than the manual installation.

> **Note:**
>
> Use easy install only for fresh installations. Do not use easy install to update an existing installation.

Step 1: Prepare Ubuntu for VDA installation

Step 1a: Verify the network configuration

We recommend that the network is connected and configured correctly before proceeding.

If you are using a Ubuntu 18.04 Live Server, make the following change in the **/etc/cloud/cloud.cfg** configuration file before setting the host name:


```
preserve_hostname: true
```

Step 1b: Set the host name

To ensure that the host name of the machine is reported correctly, change the **/etc/hostname** file to contain only the host name of the machine.

```
hostname
```

Step 1c: Assign a loopback address to the host name

Ensure that the DNS domain name and Fully Qualified Domain Name (FQDN) of the machine are reported back correctly. The way is to change the following line of the **/etc/hosts** file to include the FQDN and host name as the first two entries:

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

For example:

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

Remove any other references to **hostname-fqdn** or **hostname** from other entries in the file.

> **Note:**
>
> The Linux VDA currently does not support NetBIOS name truncation. Therefore, the host name must not exceed 15 characters.
>
> **Tip:**
>
> Use a–z, A–Z, 0–9, and hyphen (-) characters only. Avoid underscores (_), spaces, and other symbols. Do not start a host name with a number and do not end with a hyphen. This rule also applies to Delivery Controller host names.

Step 1d: Check the host name

Verify that the host name is set correctly:

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

This command returns only the host name of the machine and not its FQDN.

Verify that the FQDN is set correctly:

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


This command returns the FQDN of the machine.

Step 1e: Disable multicast DNS

The default settings have multicast DNS (**mDNS**) enabled, which can lead to inconsistent name resolution results.

To disable **mDNS**, edit **/etc/nsswitch.conf** and change the line containing:

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

To:

```
hosts: files dns
```

Step 1f: Check name resolution and service reachability

Verify that you can resolve the FQDN and ping the domain controller and Delivery Controller:

```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

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

If you cannot resolve the FQDN or ping either of these machines, review the steps before proceeding.

Step 1g: Configure clock synchronization (chrony)

Maintaining accurate clock synchronization between the VDAs, Delivery Controllers and domain controllers is crucial. Hosting the Linux VDA as a virtual machine can cause clock skew problems. For this reason, synchronizing time with a remote time service is preferred.

Install chrony:

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

As a root user, edit **/etc/chrony/chrony.conf** and add a server entry for each remote time server:

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


In a typical deployment, synchronize time from the local domain controllers and not directly from public NTP pool servers. Add a server entry for each Active Directory domain controller in the domain.

Remove any other **server** or **pool** entries listed including loopback IP address, localhost, and public server ***.pool.ntp.org** entries.

Save changes and restart the Chrony daemon:

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

Step 1h: Install OpenJDK 11

The Linux VDA requires the presence of OpenJDK 11.

On Ubuntu 16.04, install OpenJDK 11 y completing the following steps:

1. Download the latest OpenJDK 11 from https://jdk.java.net/archive/.
2. Run the `tar zxf openjdk-11.0.2_linux-x64_bin.tar.gz` command to unzip the downloaded package.
3. (Optional) Run the `mv jdk-11.0.2/ <target directory>` command to save OpenJDK in a target directory.
4. Run the `update-alternatives --install /usr/bin/java java <custom directory>/bin/java 2000` command to set up the Java runtime.
5. Run the `java -version` command to verify the version of Java.

On Ubuntu 20.04 and Ubuntu 18.04, install OpenJDK 11 by using:

```
sudo apt-get install -y openjdk-11-jdk
<!--NeedCopy-->
```

Step 1i: Install PostgreSQL

The Linux VDA requires PostgreSQL Version 9.x on Ubuntu:

```
sudo apt-get install -y postgresql

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

Step 1j: Install Motif

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


Step 1k: Install other packages

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

sudo apt-get install -y libsasl2-modules-gssapi-mit

sudo apt-get install -y libldap-2.4-2

sudo apt-get install -y krb5-user

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

Step 2: Prepare the hypervisor

Some changes are required when running the Linux VDA as a virtual machine on a supported hypervisor. Make the following changes according to the hypervisor platform in use. No changes are required if you are running the Linux machine on bare metal hardware.

Fix time synchronization on Citrix Hypervisor

When the Citrix Hypervisor Time Sync feature is enabled, within each paravirtualized Linux VM you experience issues with NTP and Citrix Hypervisor, both of which try to manage the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each Linux guest is synchronized with the NTP. This case requires disabling host time synchronization. No changes are required in HVM mode.

On some Linux distributions, if you are running a paravirtualized Linux kernel with Citrix VM Tools installed, you can check whether the Citrix Hypervisor Time Sync feature is present and enabled from within the Linux VM:

```
su -

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

This command returns 0 or 1:

- 0 - The time sync feature is enabled, and must be disabled.
- 1 - The time sync feature is disabled, and no further action is required.

If the /proc/sys/xen/independent_wallclock file is not present, the following steps are not required.

If enabled, disable the time sync feature by writing 1 to the file:


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

To make this change permanent and persistent after restart, edit the **/etc/sysctl.conf file** and add the line:

```
xen.independent_wallclock = 1
```

To verify these changes, restart the system:

```
su -

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

This command returns the value 1.

Fix time synchronization on Microsoft Hyper-V

Linux VMs with Hyper-V Linux Integration Services installed can use the Hyper-V time synchronization feature to use the host operating system's time. To ensure that the system clock remains accurate, enable this feature alongside NTP services.

From the management operating system:

1. Open the Hyper-V Manager console.
2. For the settings of a Linux VM, select **Integration Services**.
3. Ensure that **Time synchronization** is selected.

> **Note:**
>
> This approach is different from VMware and Citrix Hypervisor, where host time synchronization is disabled to avoid conflicts with NTP. Hyper-V time synchronization can coexist and supplement NTP time synchronization.

Fix time synchronization on ESX and ESXi

When the VMware Time Synchronization feature is enabled, within each paravirtualized Linux VM you experience issues with the NTP and the hypervisor, both of which try to synchronize the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each Linux guest is synchronized with the NTP. This case requires disabling host time synchronization.

If you are running a paravirtualized Linux kernel with VMware Tools installed:

1. Open the vSphere Client.

2. Edit settings for the Linux VM.
3. In the **Virtual Machine Properties** dialog, open the **Options** tab.
4. Select **VMware Tools**.
5. In the **Advanced** box, clear **Synchronize guest time with host**.

Step 3: Add the Linux virtual machine (VM) to the Windows domain

The Linux VDA supports several methods for adding Linux machines to the Active Directory (AD) domain:

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

Follow instructions based on your chosen method.

> **Note:**
>
> Session launches might fail when the same user name is used for the local account in the Linux VDA and the account in AD.

Samba Winbind

Install or update the required packages

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

Enable the Winbind daemon to start on machine startup The Winbind daemon must be configured to start on machine startup:

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

> **Note:**
>
> Ensure that the `winbind` script is located under `/etc/init.d`.

Configure Kerberos Open **/etc/krb5.conf** as a root user, and make the following settings:

> **Note:**
>
> Configure Kerberos based on your AD infrastructure. The following settings are meant for the single-domain, single-forest model.

```
[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
```

The **domain-dns-name** parameter in this context is the DNS domain name, such as **example.com**. The **REALM** is the Kerberos realm name in uppercase, such as **EXAMPLE.COM**.

Configure Winbind Authentication Configure Winbind manually because Ubuntu does not have a tool like `authconfig` in RHEL and yast2 in SUSE.

Open **/etc/samba/smb.conf**, and make the following settings:

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
encrypt passwords = yes
idmap config *:range = 16777216-33554431
winbind trusted domains only = no
kerberos method = secrets and keytab
winbind refresh tickets = yes
template shell = /bin/bash
```

WORKGROUP is the first field in **REALM**, and **REALM** is the Kerberos realm name in uppercase.

Configure nsswitch Open **/etc/nsswitch.conf**, and append `winbind` to the following lines:

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

Join Windows Domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

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

Where **REALM** is the Kerberos realm name in uppercase, and **user** is a domain user with permissions to add computers to the domain.

Restart winbind

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

Configure PAM for Winbind Run the following command and ensure that the **Winbind NT/Active Directory authentication** and **Create home directory on login** options are selected:

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

> **Tip:**
>
> The `winbind` daemon stays running only if the machine is joined to a domain.

Verify Domain Membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory.

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos Configuration To verify that Kerberos is configured correctly for use with the Linux VDA, verify that the system **keytab** file has been created and contains valid keys:


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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Examine the account details of the machine using:

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

Verify user authentication Use the **wbinfo** tool to verify that domain users can authenticate with the domain:

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

The domain specified here is the AD domain name, not the Kerberos realm name. For the bash shell, the backslash (\) character must be escaped with another backslash. This command returns a message indicating success or failure.

To verify that the Winbind PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

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

> **Note:**
>
> To run an SSH command successfully, ensure that SSH is enabled and working properly.

Verify that a corresponding Kerberos credential cache file was created for the uid returned by the **id -u** command:

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

Verify that the tickets in the user's Kerberos credential cache are valid and not expired:

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

Exit the session.

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

> **Tip:**
>
> If you succeed in user authentication but cannot show your desktop when logging on with a domain account, restart the machine and then try again.

Quest authentication service

Configure Quest on domain controller Assume that you have installed and configured the Quest software on the Active Directory domain controllers, and have been granted administrative privileges to create computer objects in Active Directory.

Enable domain users to log on to Linux VDA machines To enable domain users to establish HDX sessions on a Linux VDA machine:

1. In the Active Directory Users and Computers management console, open Active Directory user properties for that user account.
2. Select the **Unix Account** tab.
3. Check **Unix-enabled**.
4. Set the **Primary GID Number** to the group ID of an actual domain user group.

> **Note:**
>
> These instructions are equivalent for setting up domain users for logon using the console, RDP, SSH, or any other remoting protocol.

Configure Quest on Linux VDA

Work around SELinux policy enforcement The default RHEL environment has SELinux fully enforced. This enforcement interferes with the Unix domain socket IPC mechanisms used by Quest, and prevents domain users from logging on.

The convenient way to work around this issue is to disable SELinux. As a root user, edit **/etc/selinux/-config** and change the **SELinux** setting:

```
SELINUX=disabled
```

This change requires a machine restart:

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

> **Important:**
>
> Use this setting carefully. Reenabling SELinux policy enforcement after disabling can cause a complete lockout, even for the root user and other local users.

Configure VAS daemon Auto-renewal of Kerberos tickets must be enabled and disconnected. Authentication (offline logon) must be disabled:

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

sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-
   auth false
<!--NeedCopy-->
```

This command sets the renewal interval to nine hours (32,400 seconds) which is one hour less than the default 10-hour ticket lifetime. Set this parameter to a lower value on systems with a shorter ticket lifetime.

Configure PAM and NSS To enable domain user logon through HDX and other services such as su, ssh, and RDP, run the following commands to configure PAM and NSS manually:

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

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

Join Windows domain Join the Linux machine to the Active Directory domain using the Quest `vastool` command:

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


The user is any domain user with permissions to join computers to the Active Directory domain. The domain-name is the DNS name of the domain, for example, example.com.

Verify domain membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory. To verify that a Quest-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the domain name. If the machine is not joined to any domain, the following error appears:

```
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
```

Verify user authentication To verify that Quest can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

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

Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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

Verify that the tickets in the Kerberos credential cache are valid and not expired:

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

Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Centrify DirectControl

Join Windows domain With the Centrify DirectControl Agent installed, join the Linux machine to the Active Directory domain using the Centrify `adjoin` command:

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

The **user** parameter is any Active Directory domain user with permissions to join computers to the Active Directory domain. The **domain-name** parameter is the name of the domain to join the Linux machine to.

Verify domain membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory. To verify that a Centrify-joined Linux machine is on the domain:

```
su -

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

Verify that the **Joined to domain** value is valid and the **CentrifyDC mode** returns **connected**. If the mode remains stuck in the starting state, then the Centrify client is experiencing server connection or authentication problems.

More comprehensive system and diagnostic information is available using:

```
adinfo --sysinfo all

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

Test connectivity to the various Active Directory and Kerberos services.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

SSSD

Configure Kerberos Run the following command to install Kerberos:

```
sudo apt-get install krb5-user
<!--NeedCopy-->
```


To configure Kerberos, open **/etc/krb5.conf** as root and set the parameters:

> **Note:**
>
> Configure Kerberos based on your AD infrastructure. The following settings are meant for the single-domain, single-forest model.

```
[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
```

The `domain-dns-name` parameter in this context is the DNS domain name, such as example.com. The *REALM* is the Kerberos realm name in uppercase, such as EXAMPLE.COM.

Join the domain SSSD must be configured to use Active Directory as its identity provider and Kerberos for authentication. However, SSSD does not provide AD client functions for joining the domain and managing the system keytab file. You can use `adcli`, `realmd`, or `Samba` instead.

> **Note:**
>
> This section only provides information for `adcli` and `Samba`.

Use adcli to join the domain:

Install adcli:

Install the required package:

```
sudo apt-get install adcli
<!--NeedCopy-->
```

Join the domain with adcli:

Remove the old system keytab file and join the domain using:


```
su -

rm -rf /etc/krb5.keytab

adcli join domain-dns-name -U user -H hostname-fqdn
<!--NeedCopy-->
```

The **user** is a domain user with permissions to add machines to the domain. The **hostname-fqdn** is the host name in FQDN format for the machine.

The **-H** option is necessary for `adcli` to generate SPN in the format of host/hostname-fqdn@REALM, which the Linux VDA requires.

Verify system keytab:

For a Ubuntu 20.04 machine, run the `adcli testjoin` command to test whether it is joined to the domain.

For a Ubuntu 18.04 or Ubuntu 16.04 machine, run the `sudo klist -ket` command to ensure that the system keytab file has been created.

Verify that the timestamp for each key matches the time the machine was joined to the domain.

Use Samba to join the domain:

Install the package:

```
sudo apt-get install samba krb5-user
<!--NeedCopy-->
```

Configure Samba:

Open **/etc/samba/smb.conf**, and make the following settings:

```
[global]

workgroup = WORKGROUP

security = ADS

realm = REALM

client signing = yes

client use spnego = yes

kerberos method = secrets and keytab
```

WORKGROUP is the first field in **REALM**, and **REALM** is the Kerberos realm name in uppercase.

Join the domain with Samba:

Your domain controller must be reachable and you must have a Windows account with permissions to add computers to the domain.


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

Where **REALM** is the Kerberos realm name in uppercase, and **user** is a domain user with permissions to add computers to the domain.

Set up SSSD **Install or update required packages:**

Install the required SSSD and configuration packages if not already installed:

```
sudo apt-get install sssd
<!--NeedCopy-->
```

If the packages are already installed, an update is recommended:

```
sudo apt-get install --only-upgrade sssd
<!--NeedCopy-->
```

> **Note:**
>
> By default, the install process in Ubuntu configures **nsswitch.conf** and the PAM login module automatically.

Configure SSSD SSSD configuration changes are required before starting the SSSD daemon. For some versions of SSSD, the **/etc/sssd/sssd.conf** configuration file is not installed by default and must be created manually. As root, either create or open **/etc/sssd/sssd.conf** and make the following settings:

```
[sssd]
services = nss, pam
config_file_version = 2
domains = domain-dns-name
[domain/domain-dns-name]
id_provider = ad
access_provider = ad
auth_provider = krb5
krb5_realm = REALM
# Set krb5_renewable_lifetime higher if TGT renew lifetime is longer
than 14 days
krb5_renewable_lifetime = 14d
```



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

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

> **Note:**
>
> ldap_id_mapping is set to **true** so that SSSD itself takes care of mapping Windows SIDs to Unix UIDs. Otherwise, the Active Directory must be able to provide POSIX extensions. PAM service `ctxhdx` is added to ad_gpo_map_remote_interactive.
>
> The **domain-dns-name** parameter in this context is the DNS domain name, such as example.com. The **REALM** is the Kerberos realm name in uppercase, such as EXAMPLE.COM. There is no requirement to configure the NetBIOS domain name.
>
> For information about the configuration settings, see the man pages for sssd.conf and `sssd-ad`.

The SSSD daemon requires that the configuration file must have owner read permission only:

```
sudo chmod 0600 /etc/sssd/sssd.conf
<!--NeedCopy-->
```

Start SSSD daemon Run the following commands to start the SSSD daemon now and to enable the daemon to start upon machine startup:

```
sudo systemctl start sssd

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

PAM configuration Run the following command and ensure that the **SSS authentication** and **Create home directory on login** options are selected:

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


Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory.

Use adcli to verify domain membership:

Show the domain information by running the following command:

```
sudo adcli info domain-dns-name
<!--NeedCopy-->
```

Use Samba to verify domain membership:

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos configuration To verify that Kerberos is configured correctly for use with the Linux VDA, verify that the system keytab file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT for the machine account has been cached using:

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

Verify user authentication SSSD does not provide a command-line tool for testing authentication directly with the daemon, and can only be done via PAM.

To verify that the SSSD PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

id -u

klist

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

Verify that the Kerberos tickets returned by the **klist** command are correct for that user and have not expired.

As a root user, verify that a corresponding ticket cache file was created for the uid returned by the previous **id -u** command:

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

A similar test can be performed by logging on to KDE or Gnome Display Manager. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

PBIS

Download the required PBIS package For example:

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

Make the PBIS installation script executable For example:

```
sudo chmod +x pbis-open-8.8.0.506.linux.x86_64.deb.sh
<!--NeedCopy-->
```

Run the PBIS installation script For example:

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

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:


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

The **user** is a domain user who has permissions to add computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Note: To set Bash as the default shell, run the **sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** command.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a PBIS-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the information about the currently joined AD domain and OU. Otherwise, only the host name appears.

Verify user authentication To verify that PBIS can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
sudo ssh localhost -l domain\\user

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

Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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

Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Step 4: Install the Linux VDA

Step 4a: Download the Linux VDA package

Go to the Citrix Virtual Apps and Desktops download page. Expand the appropriate version of Citrix Virtual Apps and Desktops and click **Components** to download the Linux VDA package that matches your Linux distribution.

Step 4b: Install the Linux VDA

Install the Linux VDA software using the Debian package manager:

For Ubuntu 20.04:

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

For Ubuntu 18.04:

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

For Ubuntu 16.04:

```
sudo dpkg -i xendesktopvda_<version>.ubuntu16.04_amd64.deb
<!--NeedCopy-->
```

Debian dependency list for Ubuntu 20.04:

```
postgresql >= 12

libpostgresql-jdbc-java >= 42.2

openjdk-11-jdk >= 11

imagemagick >= 8:6.9.10

ufw >= 0.36

ubuntu-desktop >= 1.450

libxrandr2 >= 2:1.5.2

libxtst6 >= 2:1.2.3

libxm4 >= 2.3.8

util-linux >= 2.34

gtk3-nocsd >= 3
```



```
22
23 bash >= 5.0
24
25 findutils >= 4.7.0
26
27 sed >= 4.7
28
29 cups >= 2.3
30
31 libmspack0 >= 0.10
32
33 libgoogle-perftools4 >= 2.7~
34
35 libpython2.7 >= 2.7~
36 <!--NeedCopy-->
```

Debian dependency list for Ubuntu 18.04:

```
1  postgresql >= 9.5
2
3  libpostgresql-jdbc-java >= 9.2
4
5  openjdk-11-jdk >= 11
6
7  gtk3-nocsd >=3
8
9  imagemagick >= 8:6.8.9.9
10
11 ufw >= 0.35
12
13 ubuntu-desktop >= 1.361
14
15 libxrandr2 >= 2:1.5.0
16
17 libxtst6 >= 2:1.2.2
18
19 libxm4 >= 2.3.4
20
21 util-linux >= 2.27.1
22
23 bash >= 4.3
24
25 findutils >= 4.6.0
26
27 sed >= 4.2.2
28
29 cups >= 2.1
30
31 libldap-2.4-2 >= 2.4.42
32
33 libsasl2-modules-gssapi-mit >= 2.1.~
34
35 python-requests >= 2.9.1
```



```
libgoogle-perftools4 >= 2.4~

xserver-xorg-core >= 2:1.18

xserver-xorg-core << 2:1.19

x11vnc>=0.9.13

python-websockify >= 0.6.1
<!--NeedCopy-->
```

Debian dependency list for Ubuntu 16.04:

```
postgresql >= 9.5

libpostgresql-jdbc-java >= 9.2

imagemagick >= 8:6.8.9.9

ufw >= 0.35

ubuntu-desktop >= 1.361

libxrandr2 >= 2:1.5.0

libxtst6 >= 2:1.2.2

libxm4 >= 2.3.4

util-linux >= 2.27.1

bash >= 4.3

findutils >= 4.6.0

sed >= 4.2.2

cups >= 2.1

libldap-2.4-2 >= 2.4.42

libsasl2-modules-gssapi-mit >= 2.1.~

python-requests >= 2.9.1

libgoogle-perftools4 >= 2.4~

xserver-xorg-core >= 2:1.18

xserver-xorg-core << 2:1.19

x11vnc>=0.9.13
```



```
40
41 python-websockify >= 0.6.1
42 <!--NeedCopy-->
```

> **Note:**
>
> For a matrix of the Linux distributions and the Xorg versions that this version of the Linux VDA supports, see System requirements.

Step 4c: Upgrade the Linux VDA (optional)

You can upgrade an existing installation from the previous two versions and from an LTSR release.

```
1 sudo dpkg -i <PATH>/<Linux VDA deb>
2 <!--NeedCopy-->
```

Step 4d: Configure the Linux VDA

After installing the package, you must configure the Linux VDA by running the ctxsetup.sh script. Before making any changes, the script verifies the environment and ensures that all dependencies are installed. If necessary, you can rerun the script at any time to change settings.

You can run the script manually with prompting, or automatically with preconfigured responses. Review Help about the script before proceeding:

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

Prompted configuration Run a manual configuration with prompted questions:

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

Automated configuration For an automated install, the options required by the setup script can be provided with environment variables. If all required variables are present, the script does not prompt the user for any information, allowing for a scripted installation process.

Supported environment variables include:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N** –The Linux VDA supports specifying a Delivery Controller name using a DNS CNAME record. Set to N by default.

- **CTX_XDL_DDC_LIST='list-ddc-fqdns'**–The Linux VDA requires a space-separated list of Delivery Controller Fully Qualified Domain Names (FQDNs) to use for registering with a Delivery Controller. At least one FQDN or CNAME alias must be specified.
- **CTX_XDL_VDA_PORT=port-number** –The Linux VDA communicates with Delivery Controllers through a TCP/IP port, which is port 80 by default.
- **CTX_XDL_REGISTER_SERVICE=Y | N** –The Linux Virtual Desktop services are started after machine startup. Set to Y by default.
- **CTX_XDL_ADD_FIREWALL_RULES=Y | N** –The Linux Virtual Desktop services require incoming network connections to be allowed through the system firewall. You can open the required ports (ports 80 and 1494 by default) automatically in the system firewall for the Linux Virtual Desktop. Set to Y by default.
- **CTX_XDL_AD_INTEGRATION=1 | 2 | 3 | 4 |5** –The Linux VDA requires Kerberos configuration settings to authenticate with the Delivery Controllers. The Kerberos configuration is determined from the installed and configured Active Directory integration tool on the system. Specify the supported Active Directory integration method to use:
  - 1 –Samba Winbind
  - 2 –Quest Authentication Service
  - 3 –Centrify DirectControl
  - 4 –SSSD
  - 5 –PBIS
- **CTX_XDL_HDX_3D_PRO=Y | N** –The Linux VDA supports HDX 3D Pro, a set of GPU acceleration technologies designed to optimize the virtualization of rich graphics applications. If HDX 3D Pro is selected, the VDA is configured for VDI desktops (single-session) mode - (that is, CTX_XDL_VDI_MODE=Y).
- **CTX_XDL_VDI_MODE=Y | N** –Whether to configure the machine as a dedicated desktop delivery model (VDI) or hosted shared desktop delivery model. For HDX 3D Pro environments, set this variable to Y. This variable is set to N by default.
- **CTX_XDL_SITE_NAME=dns-name** –The Linux VDA discovers LDAP servers through DNS. To limit the DNS search results to a local site, specify a DNS site name. This variable is set to **<none>** by default.
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** –The Linux VDA queries DNS to discover LDAP servers. If DNS cannot provide LDAP service records, you can provide a space-separated list of LDAP FQDNs with LDAP ports. For example, ad1.mycompany.com:389. This variable is set to **<none>** by default.
- **CTX_XDL_SEARCH_BASE=search-base-set** –The Linux VDA queries LDAP through a search base set to the root of the Active Directory Domain (for example, DC=mycompany,DC=com).

However, to improve search performance, you can specify a search base (for example, OU=VDI,DC=mycompany,DC=com). This variable is set to **<none>** by default.

- **CTX_XDL_FAS_LIST='list-fas-servers'**–The Federated Authentication Service (FAS) servers are configured through AD Group Policy. The Linux VDA does not support AD Group Policy, but you can provide a semicolon-separated list of FAS servers instead. The sequence must be the same as configured in AD Group Policy. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.
- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** –The path to install .NET Core Runtime 3.1 for supporting the new broker agent service (`ctxvda`). The default path is /usr/bin.
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** –Specifies the GNOME or MATE desktop environment to use in sessions. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set the variable value to **mate**.

  Note:

  You can also change the desktop environment for a target session user by completing the following steps:

  1. Create a `.xsession` file under the **$HOME/<username>** directory on the VDA.
  2. Edit the `.xsession` file to specify a desktop environment based on distributions.

     For MATE desktop on CentOS, Ubuntu, and Debian

     MSESSION="$(type -p mate-session)"
     if [-n "$MSESSION"]; then
     exec mate-session
     fi

     For GNOME desktop on CentOS

     GSESSION="$(type -p gnome-session)"
     if [-n "$GSESSION"]; then

     ```
     1     export GNOME_SHELL_SESSION_MODE=classic
     2     exec gnome-session --session=gnome-classic      fi
                **For GNOME desktop on Ubuntu and Debian**
     ```

     GSESSION="$(type -p gnome-session)"
     if [-n "$GSESSION"]; then

     ```
     1     exec gnome-session      fi
     ```


> 3. Share the 700 file permission with the target session user.

- **CTX_XDL_START_SERVICE=Y | N** –Whether or not the Linux VDA services are started when the Linux VDA configuration is complete. Set to Y by default.
- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The default port is 7503.
- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

Set the environment variable and run the configure script:

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST= ‘ list-ddc-fqdns ’

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

When running the sudo command, type the **-E** option to pass the existing environment variables to the new shell it creates. We recommend that you create a shell script file from the preceding commands

with **#!/bin/bash** as the first line.

Alternatively, you can specify all parameters by using a single command:

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST= ' list-ddc-fqdns '  \

CTX_XDL_VDA_PORT=port-number \

CTX_XDL_REGISTER_SERVICE=Y|N \

CTX_XDL_ADD_FIREWALL_RULES=Y|N \

CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \

CTX_XDL_HDX_3D_PRO=Y|N \

CTX_XDL_VDI_MODE=Y|N \

CTX_XDL_SITE_NAME=dns-name \

CTX_XDL_LDAP_LIST= ' list-ldap-servers '  \

CTX_XDL_SEARCH_BASE=search-base-set \

CTX_XDL_FAS_LIST= ' list-fas-servers '  \

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

CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \

CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \

CTX_XDL_TELEMETRY_PORT=port-number \

CTX_XDL_START_SERVICE=Y|N \

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

Remove configuration changes In some scenarios, you might have to remove the configuration changes made by the **ctxsetup.sh** script without uninstalling the Linux VDA package.

Review Help about this script before proceeding:

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

To remove configuration changes:

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



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

> **Important:**
>
> This script deletes all configuration data from the database and renders the Linux VDA inoperable.

Configuration logs The **ctxsetup.sh** and **ctxcleanup.sh** scripts display errors on the console, with additional information written to the configuration log file **/tmp/xdl.configure.log**.

Restart the Linux VDA services to have the changes take effect.

Uninstall the Linux VDA software To check whether the Linux VDA is installed and to view the version of the installed package:

```
dpkg -l xendesktopvda
<!--NeedCopy-->
```

To view more detailed information:

```
apt-cache show xendesktopvda
<!--NeedCopy-->
```

To uninstall the Linux VDA software:

```
dpkg -r xendesktopvda
<!--NeedCopy-->
```

> **Note:**
>
> Uninstalling the Linux VDA software deletes the associated PostgreSQL and other configuration data. However, the PostgreSQL package and other dependent packages that were set up before the installation of the Linux VDA are not deleted.
>
> **Tip:**
>
> The information in this section does not cover the removal of dependent packages including PostgreSQL.

Step 5: Run the Linux VDA

Once you have configured the Linux VDA using the **ctxsetup.sh** script, you use the following commands to control the Linux VDA.

Start the Linux VDA:

To start the Linux VDA services:


```
sudo systemctl start ctxhdx

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

Stop the Linux VDA:

To stop the Linux VDA services:

```
sudo systemctl stop ctxvda

sudo systemctl stop ctxhdx
<!--NeedCopy-->
```

> **Note:**
>
> Before you stop the `ctxvda` and `ctxhdx` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

Restart the Linux VDA:

To restart the Linux VDA services:

```
sudo systemctl stop ctxvda

sudo systemctl restart ctxhdx

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

Check the Linux VDA status:

To check the running status of the Linux VDA services:

```
sudo systemctl status ctxvda

sudo systemctl status ctxhdx
<!--NeedCopy-->
```

Step 6: Create the machine catalog in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating machine catalogs and adding Linux VDA machines is similar to the traditional Windows VDA approach. For a more detailed description of how to complete these tasks, see Create machine catalogs and Manage machine catalogs.

For creating machine catalogs that contain Linux VDA machines, there are a few restrictions that differentiate the process from creating machine catalogs for Windows VDA machines:

- For the operating system, select:
  - The **Multi-session OS** option for a hosted shared desktops delivery model.
  - The **Single-session OS** option for a VDI dedicated desktop delivery model.
- Do not mix Linux and Windows VDA machines in the same machine catalog.

> **Note:**
>
> Early versions of Citrix Studio did not support the notion of a "Linux OS."However, selecting the **Windows Server OS** or **Server OS** option implies an equivalent hosted shared desktops delivery model. Selecting the **Windows Desktop OS** or **Desktop OS** option implies a single user per machine delivery model.
>
> **Tip:**
>
> If you remove and rejoin a machine to the Active Directory domain, you must remove and add the machine to the machine catalog again.

Step 7: Create the delivery group in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating a delivery group and adding machine catalogs containing Linux VDA machines is almost identical to Windows VDA machines. For a more detailed description of how to complete these tasks, see Create Delivery Groups.

For creating delivery groups that contain Linux VDA machine catalogs, the following restrictions apply:

- Ensure that the AD users and groups you select have been properly configured to log on to the Linux VDA machines.
- Do not allow logon of unauthenticated (anonymous) users.
- Do not mix the delivery group with machine catalogs that contain Windows machines.

For information about how to create machine catalogs and delivery groups, see Citrix Virtual Apps and Desktops 7 2106.

Install Linux Virtual Delivery Agent for Debian

June 10, 2022

You can choose to follow the steps in this article for manual installation or use easy install for automatic installation and configuration. Easy install saves time and labor and is less error-prone than the manual installation.

> **Note:**
>
> Use easy install only for fresh installations. Do not use easy install to update an existing installation.

Step 1: Prepare Debian for VDA installation

Step 1a: Verify the network configuration

We recommend that the network is connected and configured correctly before proceeding.

Step 1b: Set the host name

To ensure that the host name of the machine is reported correctly, change the **/etc/hostname** file to contain only the host name of the machine.

```
hostname
```

Step 1c: Assign a loopback address to the host name

Ensure that the DNS domain name and Fully Qualified Domain Name (FQDN) of the machine are reported back correctly. The way is to change the following line of the **/etc/hosts** file to include the FQDN and host name as the first two entries:

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

For example:

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

Remove any other references to **hostname-fqdn** or **hostname** from other entries in the file.

> **Note:**
>
> The Linux VDA currently does not support NetBIOS name truncation. Therefore, the host name must not exceed 15 characters.
>
> **Tip:**
>
> Use a–z, A–Z, 0–9, and hyphen (-) characters only. Avoid underscores (_), spaces, and other symbols. Do not start a host name with a number and do not end with a hyphen. This rule also applies to Delivery Controller host names.

Step 1d: Check the host name

Verify that the host name is set correctly:

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

This command returns only the host name of the machine and not its FQDN.

Verify that the FQDN is set correctly:

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

This command returns the FQDN of the machine.

Step 1e: Disable multicast DNS

The default settings have multicast DNS (**mDNS**) enabled, which can lead to inconsistent name resolution results.

To disable **mDNS**, edit **/etc/nsswitch.conf** and change the line containing:

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

To:

```
hosts: files dns
```

Step 1f: Check name resolution and service reachability

Verify that you can resolve the FQDN and ping the domain controller and Delivery Controller:

```
nslookup domain-controller-fqdn

ping domain-controller-fqdn

nslookup delivery-controller-fqdn

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

If you cannot resolve the FQDN or ping either of these machines, review the steps before proceeding.

Step 1g: Configure clock synchronization (chrony)

Maintaining accurate clock synchronization between the VDAs, Delivery Controllers and domain controllers is crucial. Hosting the Linux VDA as a virtual machine can cause clock skew problems. For this reason, synchronizing time with a remote time service is preferred.

Install chrony:

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

As a root user, edit **/etc/chrony/chrony.conf** and add a server entry for each remote time server:

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

In a typical deployment, synchronize time from the local domain controllers and not directly from public NTP pool servers. Add a server entry for each Active Directory domain controller in the domain.

Remove any other **server** or **pool** entries listed including loopback IP address, localhost, and public server ***.pool.ntp.org** entries.

Save changes and restart the Chrony daemon:

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

Step 1h: Install packages

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

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

Step 1i: Add the oldstable repository

To install the necessary dependencies for a Debian distribution, add the `deb http://deb.debian.org/debian/ oldstable main` line to the `/etc/apt/sources.list` file.

Step 2: Prepare the hypervisor

Some changes are required when running the Linux VDA as a virtual machine on a supported hypervisor. Make the following changes according to the hypervisor platform in use. No changes are required if you are running the Linux machine on bare metal hardware.

Fix time synchronization on Citrix Hypervisor

When the Citrix Hypervisor Time Sync feature is enabled, within each paravirtualized Linux VM you experience issues with NTP and Citrix Hypervisor, both of which try to manage the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each Linux guest is synchronized with the NTP. This case requires disabling host time synchronization. No changes are required in HVM mode.

On some Linux distributions, if you are running a paravirtualized Linux kernel with Citrix VM Tools installed, you can check whether the Citrix Hypervisor Time Sync feature is present and enabled from within the Linux VM:

```
su -

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

This command returns 0 or 1:

- 0 - The time sync feature is enabled, and must be disabled.
- 1 - The time sync feature is disabled, and no further action is required.

If the /proc/sys/xen/independent_wallclock file is not present, the following steps are not required.

If enabled, disable the time sync feature by writing 1 to the file:

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

To make this change permanent and persistent after restart, edit the **/etc/sysctl.conf file** and add the line:

```
xen.independent_wallclock = 1
```

To verify these changes, restart the system:

```
su -

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

This command returns the value 1.

Fix time synchronization on Microsoft Hyper-V

Linux VMs with Hyper-V Linux Integration Services installed can use the Hyper-V time synchronization feature to use the host operating system's time. To ensure that the system clock remains accurate, enable this feature alongside NTP services.

From the management operating system:

1. Open the Hyper-V Manager console.
2. For the settings of a Linux VM, select **Integration Services**.
3. Ensure that **Time synchronization** is selected.

> **Note:**
>
> This approach is different from VMware and Citrix Hypervisor, where host time synchronization is disabled to avoid conflicts with NTP. Hyper-V time synchronization can coexist and supplement NTP time synchronization.

Fix time synchronization on ESX and ESXi

When the VMware Time Synchronization feature is enabled, within each paravirtualized Linux VM you experience issues with the NTP and the hypervisor, both of which try to synchronize the system clock. To avoid the clock becoming out of sync with other servers, ensure that the system clock within each Linux guest is synchronized with the NTP. This case requires disabling host time synchronization.

If you are running a paravirtualized Linux kernel with VMware Tools installed:

1. Open the vSphere Client.
2. Edit settings for the Linux VM.
3. In the **Virtual Machine Properties** dialog, open the **Options** tab.
4. Select **VMware Tools**.
5. In the **Advanced** box, clear **Synchronize guest time with host**.

Step 3: Add the Linux virtual machine (VM) to the Windows domain

The Linux VDA supports several methods for adding Linux machines to the Active Directory (AD) domain:

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

Follow instructions based on your chosen method.

> **Note:**

> Session launches might fail when the same user name is used for the local account in the Linux VDA and the account in AD.

Samba Winbind

Install or update the required packages

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

Enable the Winbind daemon to start on machine startup The Winbind daemon must be configured to start on machine startup:

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

> **Note:**
>
> Ensure that the winbind script is located under `/etc/init.d`.

Configure Kerberos Open **/etc/krb5.conf** as a root user, and make the following settings:

> **Note:**
>
> Configure Kerberos based on your AD infrastructure. The following settings are meant for the single-domain, single-forest model.

```
[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
```

The **domain-dns-name** parameter in this context is the DNS domain name, such as **example.com**. The **REALM** is the Kerberos realm name in uppercase, such as **EXAMPLE.COM**.

Configure Winbind Authentication Open **/etc/samba/smb.conf**, and make the following settings:

```
[global]
workgroup = WORKGROUP
security = ADS
realm = REALM
encrypt passwords = yes
idmap config *:range = 16777216-33554431
winbind trusted domains only = no
kerberos method = secrets and keytab
winbind refresh tickets = yes
template shell = /bin/bash
```

WORKGROUP is the first field in **REALM**, and **REALM** is the Kerberos realm name in uppercase.

Configure nsswitch Open **/etc/nsswitch.conf**, and append `winbind` to the following lines:

```
passwd: systemd winbind
group: systemd winbind
```

Join Windows Domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

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

Where **REALM** is the Kerberos realm name in uppercase, and **user** is a domain user with permissions to add computers to the domain.

Restart Winbind

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

Configure PAM for Winbind Run the following command and ensure that the **Winbind NT/Active Directory authentication** and **Create home directory on login** options are selected:

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


> **Tip:**
>
> The `winbind` daemon stays running only if the machine is joined to a domain.

Verify Domain Membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory.

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos Configuration To verify that Kerberos is configured correctly for use with the Linux VDA, verify that the system **keytab** file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:

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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Examine the account details of the machine using:

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

Verify user authentication Use the **wbinfo** tool to verify that domain users can authenticate with the domain:


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

The domain specified here is the AD domain name, not the Kerberos realm name. For the bash shell, the backslash (\) character must be escaped with another backslash. This command returns a message indicating success or failure.

To verify that the Winbind PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

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

> **Note:**
>
> To run an SSH command successfully, ensure that SSH is enabled and working properly.

Verify that a corresponding Kerberos credential cache file was created for the uid returned by the **id -u** command:

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

Verify that the tickets in the user's Kerberos credential cache are valid and not expired:

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

Exit the session.

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

A similar test can be performed by logging on to the Gnome or KDE console directly. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

> **Tip:**
>
> If you succeed in user authentication but cannot show your desktop when logging on with a domain account, restart the machine and then try again.

Quest authentication service

Configure Quest on domain controller Assume that you have installed and configured the Quest software on the Active Directory domain controllers, and have been granted administrative privileges to create computer objects in Active Directory.

Enable domain users to log on to Linux VDA machines To enable domain users to establish HDX sessions on a Linux VDA machine:

1. In the Active Directory Users and Computers management console, open Active Directory user properties for that user account.
2. Select the **Unix Account** tab.
3. Check **Unix-enabled**.
4. Set the **Primary GID Number** to the group ID of an actual domain user group.

> **Note:**
>
> These instructions are equivalent for setting up domain users for logon using the console, RDP, SSH, or any other remoting protocol.

Configure Quest on Linux VDA

Work around SELinux policy enforcement The default RHEL environment has SELinux fully enforced. This enforcement interferes with the Unix domain socket IPC mechanisms used by Quest, and prevents domain users from logging on.

The convenient way to work around this issue is to disable SELinux. As a root user, edit **/etc/selinux/-config** and change the **SELinux** setting:

```
SELINUX=disabled
```

This change requires a machine restart:

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

> **Important:**
>
> Use this setting carefully. Reenabling SELinux policy enforcement after disabling can cause a complete lockout, even for the root user and other local users.

Configure VAS daemon Auto-renewal of Kerberos tickets must be enabled and disconnected. Authentication (offline logon) must be disabled:

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

sudo /opt/quest/bin/vastool configure vas vas_auth allow-disconnected-
    auth false
<!--NeedCopy-->
```


This command sets the renewal interval to nine hours (32,400 seconds) which is one hour less than the default 10-hour ticket lifetime. Set this parameter to a lower value on systems with a shorter ticket lifetime.

Configure PAM and NSS To enable domain user logon through HDX and other services such as su, ssh, and RDP, run the following commands to configure PAM and NSS manually:

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

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

Join Windows domain Join the Linux machine to the Active Directory domain using the Quest `vastool` command:

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

The user is any domain user with permissions to join computers to the Active Directory domain. The domain-name is the DNS name of the domain, for example, example.com.

Verify domain membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory. To verify that a Quest-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the domain name. If the machine is not joined to any domain, the following error appears:

```
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
```

Verify user authentication To verify that Quest can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

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


Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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

Verify that the tickets in the Kerberos credential cache are valid and not expired:

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

Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Centrify DirectControl

Join Windows domain With the Centrify DirectControl Agent installed, join the Linux machine to the Active Directory domain using the Centrify `adjoin` command:

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

The **user** parameter is any Active Directory domain user with permissions to join computers to the Active Directory domain. The **domain-name** parameter is the name of the domain to join the Linux machine to.

Verify domain membership The Delivery Controller requires that all VDA machines, whether Windows or Linux, have a computer object in Active Directory. To verify that a Centrify-joined Linux machine is on the domain:

```
su -

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

Verify that the **Joined to domain** value is valid and the **CentrifyDC mode** returns **connected**. If the mode remains stuck in the starting state, then the Centrify client is experiencing server connection or authentication problems.

More comprehensive system and diagnostic information is available using:

```
adinfo --sysinfo all
```



```
2
3 adinfo --diag
4 <!--NeedCopy-->
```

Test connectivity to the various Active Directory and Kerberos services.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

SSSD

Configure Kerberos Run the following command to install Kerberos:

```
1 sudo apt-get install krb5-user
2 <!--NeedCopy-->
```

To configure Kerberos, open **/etc/krb5.conf** as root and set the parameters:

> **Note:**
>
> Configure Kerberos based on your AD infrastructure. The following settings are meant for the single-domain, single-forest model.

```
[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
```

The domain-dns-name parameter in this context is the DNS domain name, such as example.com. The *REALM* is the Kerberos realm name in uppercase, such as EXAMPLE.COM.

Join the domain SSSD must be configured to use Active Directory as its identity provider and Kerberos for authentication. However, SSSD does not provide AD client functions for joining the domain and managing the system keytab file. You can use `adcli`, `realmd`, or `Samba` instead.

> **Note:**
>
> This section only provides information for `adcli` and `Samba`.

Use adcli to join the domain:

Install adcli:

Install the required package:

```
sudo apt-get install adcli
<!--NeedCopy-->
```

Join the domain with adcli:

Remove the old system keytab file and join the domain using:

```
su -

rm -rf /etc/krb5.keytab

adcli join domain-dns-name -U user -H hostname-fqdn
<!--NeedCopy-->
```

The **user** is a domain user with permissions to add machines to the domain. The **hostname-fqdn** is the host name in FQDN format for the machine.

The **-H** option is necessary for `adcli` to generate SPN in the format of host/hostname-fqdn@REALM, which the Linux VDA requires.

Verify system keytab:

Run the `sudo klist -ket` command to ensure that the system keytab file has been created.

Verify that the timestamp for each key matches the time the machine was joined to the domain.

Use Samba to join the domain:

Install the package:

```
sudo apt-get install samba krb5-user
<!--NeedCopy-->
```

Configure Samba:

Open **/etc/samba/smb.conf**, and make the following settings:

```
[global]

workgroup = WORKGROUP
```



```
security = ADS

realm = REALM

client signing = yes

client use spnego = yes

kerberos method = secrets and keytab
```

WORKGROUP is the first field in **REALM**, and **REALM** is the Kerberos realm name in uppercase.

Join the domain with Samba:

Your domain controller must be reachable and you must have a Windows account with permissions to add computers to the domain.

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

Where **REALM** is the Kerberos realm name in uppercase, and **user** is a domain user with permissions to add computers to the domain.

Set up SSSD Install or update required packages:

Install the required SSSD and configuration packages if not already installed:

```
sudo apt-get install sssd
<!--NeedCopy-->
```

If the packages are already installed, an update is recommended:

```
sudo apt-get install --only-upgrade sssd
<!--NeedCopy-->
```

> **Note:**
>
> By default, the install process in Ubuntu automatically configures **nsswitch.conf** and the PAM login module.

Configure SSSD SSSD configuration changes are required before starting the SSSD daemon. For some versions of SSSD, the **/etc/sssd/sssd.conf** configuration file is not installed by default and must be created manually. As root, either create or open **/etc/sssd/sssd.conf** and make the following settings:

```
[sssd]

services = nss, pam

config_file_version = 2
```



```
domains = domain-dns-name

[domain/domain-dns-name]

id_provider = ad

access_provider = ad

auth_provider = krb5

krb5_realm = REALM

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

krb5_renewable_lifetime = 14d

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

krb5_renew_interval = 1h

krb5_ccachedir = /tmp

krb5_ccname_template = FILE:%d/krb5cc_%U

# This ldap_id_mapping setting is also the default value

ldap_id_mapping = true

override_homedir = /home/%d/%u

default_shell = /bin/bash

ad_gpo_map_remote_interactive = +ctxhdx
```

> **Note:**
>
> ldap_id_mapping is set to **true** so that SSSD itself takes care of mapping Windows SIDs to Unix UIDs. Otherwise, the Active Directory must be able to provide POSIX extensions. PAM service `ctxhdx` is added to ad_gpo_map_remote_interactive.
>
> The **domain-dns-name** parameter in this context is the DNS domain name, such as example.com. The **REALM** is the Kerberos realm name in uppercase, such as EXAMPLE.COM. There is no requirement to configure the NetBIOS domain name.
>
> For information about the configuration settings, see the man pages for sssd.conf and `sssd-ad`.

The SSSD daemon requires that the configuration file must have owner read permission only:

```
sudo chmod 0600 /etc/sssd/sssd.conf
<!--NeedCopy-->
```


Start SSSD daemon Run the following commands to start the SSSD daemon now and to enable the daemon to start upon machine startup:

```
sudo systemctl start sssd

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

PAM configuration Run the following command and ensure that the **SSS authentication** and **Create home directory on login** options are selected:

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

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory.

Use adcli to verify domain membership:

Show the domain information by running the following command:

```
sudo adcli info domain-dns-name
<!--NeedCopy-->
```

Use Samba to verify domain membership:

Run the **net ads** command of Samba to verify that the machine is joined to a domain:

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

Run the following command to verify extra domain and computer object information:

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

Verify Kerberos configuration To verify that Kerberos is configured correctly for use with the Linux VDA, verify that the system keytab file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos `kinit` command to authenticate the machine with the domain controller using these keys:


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

The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT for the machine account has been cached using:

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

Verify user authentication SSSD does not provide a command-line tool for testing authentication directly with the daemon, and can only be done via PAM.

To verify that the SSSD PAM module is configured correctly, log on to the Linux VDA using a domain user account that has not been used before.

```
ssh localhost -l domain\\username

id -u

klist

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

Verify that the Kerberos tickets returned by the **klist** command are correct for that user and have not expired.

As a root user, verify that a corresponding ticket cache file was created for the uid returned by the previous **id -u** command:

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

A similar test can be performed by logging on to KDE or Gnome Display Manager. Proceed to Step 4: Install the Linux VDA after the domain joining verification.

PBIS

Download the required PBIS package For example:

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


Make the PBIS installation script executable For example:

```
sudo chmod +x pbis-open-8.8.0.506.linux.x86_64.deb.sh
<!--NeedCopy-->
```

Run the PBIS installation script For example:

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

Join Windows domain Your domain controller must be reachable and you must have an Active Directory user account with permissions to add computers to the domain:

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

The **user** is a domain user who has permissions to add computers to the Active Directory domain. The **domain-name** is the DNS name of the domain, for example, example.com.

Note: To set Bash as the default shell, run the **sudo /opt/pbis/bin/config LoginShellTemplate/bin/bash** command.

Verify domain membership The Delivery Controller requires that all VDA machines (Windows and Linux VDAs) have a computer object in Active Directory. To verify that a PBIS-joined Linux machine is on the domain:

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

If the machine is joined to a domain, this command returns the information about the currently joined AD domain and OU. Otherwise, only the host name appears.

Verify user authentication To verify that PBIS can authenticate domain users through PAM, log on to the Linux VDA using a domain user account that has not been used before.

```
sudo ssh localhost -l domain\\user

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

Verify that a corresponding Kerberos credential cache file was created for the UID returned by the **id -u** command:

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


Exit the session.

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

Proceed to Step 4: Install the Linux VDA after the domain joining verification.

Step 4: Install the Linux VDA

Step 4a: Download the Linux VDA package

Go to the Citrix Virtual Apps and Desktops download page. Expand the appropriate version of Citrix Virtual Apps and Desktops and click **Components** to download the Linux VDA package that matches your Linux distribution.

Step 4b: Install the Linux VDA

Install the Linux VDA software using the Debian package manager:

```
sudo dpkg -i xendesktopvda_<version>.debian10_amd64.deb
<!--NeedCopy-->
```

Debian dependency list for Debian 10.7:

```
postgresql              >= 11
libpostgresql-jdbc-java >= 42.2
openjdk-8-jdk           >= 8u252
imagemagick             >= 8:6.9.10
ufw                     >= 0.36
desktop-base            >= 10.0.2
libxrandr2              >= 2:1.5.1
libxtst6                >= 2:1.2.3
libxm4                  >= 2.3.8
util-linux              >= 2.33
gtk3-nocsd              >= 3
bash                    >= 5.0
findutils               >= 4.6.0
sed                     >= 4.7
cups                    >= 2.2
ghostscript             >= 9.27~
libmspack0              >= 0.10
libgoogle-perftools4    >= 2.7~
libpython2.7            >= 2.7~
libsasl2-modules-gssapi-mit >= 2.1.~

<!--NeedCopy-->
```


> **Note:**
>
> For a matrix of the Linux distributions and the Xorg versions that this version of the Linux VDA supports, see System requirements.

Step 4c: Upgrade the Linux VDA (optional)

You can upgrade an existing installation from the previous two versions and from an LTSR release.

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

Step 4d: Configure the Linux VDA

After installing the package, you must configure the Linux VDA by running the ctxsetup.sh script. Before making any changes, the script verifies the environment and ensures that all dependencies are installed. If necessary, you can rerun the script at any time to change settings.

You can run the script manually with prompting, or automatically with preconfigured responses. Review Help about the script before proceeding:

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

Prompted configuration Run a manual configuration with prompted questions:

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

Automated configuration For an automated install, the options required by the setup script can be provided with environment variables. If all required variables are present, the script does not prompt the user for any information, allowing for a scripted installation process.

Supported environment variables include:

- **CTX_XDL_SUPPORT_DDC_AS_CNAME=Y | N** –The Linux VDA supports specifying a Delivery Controller name using a DNS CNAME record. Set to N by default.
- **CTX_XDL_DDC_LIST='list-ddc-fqdns'**–The Linux VDA requires a space-separated list of Delivery Controller Fully Qualified Domain Names (FQDNs) to use for registering with a Delivery Controller. At least one FQDN or CNAME alias must be specified.
- **CTX_XDL_VDA_PORT=port-number** –The Linux VDA communicates with Delivery Controllers through a TCP/IP port, which is port 80 by default.

- **CTX_XDL_REGISTER_SERVICE=Y | N** –The Linux Virtual Desktop services are started after machine startup. Set to Y by default.
- **CTX_XDL_ADD_FIREWALL_RULES=Y | N** –The Linux Virtual Desktop services require incoming network connections to be allowed through the system firewall. You can open the required ports (ports 80 and 1494 by default) automatically in the system firewall for the Linux Virtual Desktop. Set to Y by default.
- **CTX_XDL_AD_INTEGRATION=1 | 2 | 3 | 4 |5** –The Linux VDA requires Kerberos configuration settings to authenticate with the Delivery Controllers. The Kerberos configuration is determined from the installed and configured Active Directory integration tool on the system. Specify the supported Active Directory integration method to use:
  - 1 –Samba Winbind
  - 2 –Quest Authentication Service
  - 3 –Centrify DirectControl
  - 4 –SSSD
  - 5 –PBIS
- **CTX_XDL_HDX_3D_PRO=Y | N** –The Linux VDA supports HDX 3D Pro, a set of GPU acceleration technologies designed to optimize the virtualization of rich graphics applications. If HDX 3D Pro is selected, the VDA is configured for VDI desktops (single-session) mode - (that is, CTX_XDL_VDI_MODE=Y).
- **CTX_XDL_VDI_MODE=Y | N** –Whether to configure the machine as a dedicated desktop delivery model (VDI) or hosted shared desktop delivery model. For HDX 3D Pro environments, set this variable to Y. This variable is set to N by default.
- **CTX_XDL_SITE_NAME=dns-name** –The Linux VDA discovers LDAP servers through DNS. To limit the DNS search results to a local site, specify a DNS site name. This variable is set to **<none>** by default.
- **CTX_XDL_LDAP_LIST='list-ldap-servers'** –The Linux VDA queries DNS to discover LDAP servers. If DNS cannot provide LDAP service records, you can provide a space-separated list of LDAP FQDNs with LDAP ports. For example, ad1.mycompany.com:389. This variable is set to **<none>** by default.
- **CTX_XDL_SEARCH_BASE=search-base-set** –The Linux VDA queries LDAP through a search base set to the root of the Active Directory Domain (for example, DC=mycompany,DC=com). However, to improve search performance, you can specify a search base (for example, OU=VDI,DC=mycompany,DC=com). This variable is set to **<none>** by default.
- **CTX_XDL_FAS_LIST='list-fas-servers'**–The Federated Authentication Service (FAS) servers are configured through AD Group Policy. The Linux VDA does not support AD Group Policy, but you can provide a semicolon-separated list of FAS servers instead. The sequence must be the same

as configured in AD Group Policy. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.

- **CTX_XDL_DOTNET_ RUNTIME_PATH=path-to-install-dotnet-runtime** –The path to install .NET Core Runtime 3.1 for supporting the new broker agent service (`ctxvda`). The default path is /usr/bin.
- **CTX_XDL_DESKTOP _ENVIRONMENT=gnome/mate** –Specifies the GNOME or MATE desktop environment to use in sessions. If you leave the variable unspecified, the desktop currently installed on the VDA is used. However, if the currently installed desktop is MATE, you must set the variable value to **mate**.

  > **Note:**
  >
  > You can also change the desktop environment for a target session user by completing the following steps:
  >
  > 1. Create a `.xsession` file under the **$HOME/<username>** directory on the VDA.
  > 2. Edit the `.xsession` file to specify a desktop environment based on distributions.
  >
  > **For MATE desktop on CentOS, Ubuntu, and Debian**
  >
  > MSESSION=”$(type -p mate-session)”
  > if [-n “$MSESSION”]; then
  > exec mate-session
  > fi
  >
  > **For GNOME desktop on CentOS**
  >
  > GSESSION=”$(type -p gnome-session)”
  > if [-n “$GSESSION”]; then
  >
  > ```
  > 1     export GNOME_SHELL_SESSION_MODE=classic
  > 2     exec gnome-session --session=gnome-classic      fi
  >           **For GNOME desktop on Ubuntu and Debian**
  > ```
  >
  > GSESSION=”$(type -p gnome-session)”
  > if [-n “$GSESSION”]; then
  >
  > ```
  > 1     exec gnome-session      fi
  > ```
  >
  > 3. Share the 700 file permission with the target session user.

- **CTX_XDL_START_SERVICE=Y | N** –Whether or not the Linux VDA services are started when the Linux VDA configuration is complete. Set to Y by default.
- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The de-

fault port is 7503.

- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

Set the environment variable and run the configure script:

```
export CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N

export CTX_XDL_DDC_LIST=‘list-ddc-fqdns’

export CTX_XDL_VDA_PORT=port-number

export CTX_XDL_REGISTER_SERVICE=Y|N

export CTX_XDL_ADD_FIREWALL_RULES=Y|N

export CTX_XDL_AD_INTEGRATION=1|2|3|4|5

export CTX_XDL_HDX_3D_PRO=Y|N

export CTX_XDL_VDI_MODE=Y|N

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

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

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

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

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

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

export CTX_XDL_TELEMETRY_SOCKET_PORT=port-number

export CTX_XDL_TELEMETRY_PORT=port-number

export CTX_XDL_START_SERVICE=Y|N

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

When running the sudo command, type the **-E** option to pass the existing environment variables to the new shell it creates. We recommend that you create a shell script file from the preceding commands with **#!/bin/bash** as the first line.

Alternatively, you can specify all parameters by using a single command:

```
sudo CTX_XDL_SUPPORT_DDC_AS_CNAME=Y|N \

CTX_XDL_DDC_LIST=‘list-ddc-fqdns’ \
```



```
4
5  CTX_XDL_VDA_PORT=port-number \
6
7  CTX_XDL_REGISTER_SERVICE=Y|N \
8
9  CTX_XDL_ADD_FIREWALL_RULES=Y|N \
10
11 CTX_XDL_AD_INTEGRATION=1|2|3|4|5 \
12
13 CTX_XDL_HDX_3D_PRO=Y|N \
14
15 CTX_XDL_VDI_MODE=Y|N \
16
17 CTX_XDL_SITE_NAME=dns-name \
18
19 CTX_XDL_LDAP_LIST= ‘ list-ldap-servers ’  \
20
21 CTX_XDL_SEARCH_BASE=search-base-set \
22
23 CTX_XDL_FAS_LIST= ‘ list-fas-servers ’  \
24
25 CTX_XDL_DOTNET_RUNTIME_PATH=path-to-install-dotnet-runtime \
26
27 CTX_XDL_DESKTOP_ENVIRONMENT=gnome | mate \
28
29 CTX_XDL_TELEMETRY_SOCKET_PORT=port-number \
30
31 CTX_XDL_TELEMETRY_PORT=port-number \
32
33 CTX_XDL_START_SERVICE=Y|N \
34
35 /opt/Citrix/VDA/sbin/ctxsetup.sh
36 <!--NeedCopy-->
```

Remove configuration changes In some scenarios, you might have to remove the configuration changes made by the **ctxsetup.sh** script without uninstalling the Linux VDA package.

Review Help about this script before proceeding:

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

To remove configuration changes:

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

> **Important:**
>
> This script deletes all configuration data from the database and renders the Linux VDA inoperable.

Configuration logs The **ctxsetup.sh** and **ctxcleanup.sh** scripts display errors on the console, with additional information written to the configuration log file **/tmp/xdl.configure.log**.

Restart the Linux VDA services to have the changes take effect.

Uninstall the Linux VDA software To check whether the Linux VDA is installed and to view the version of the installed package:

```
dpkg -l xendesktopvda
<!--NeedCopy-->
```

To view more detailed information:

```
apt-cache show xendesktopvda
<!--NeedCopy-->
```

To uninstall the Linux VDA software:

```
dpkg -r xendesktopvda
<!--NeedCopy-->
```

> **Note:**
>
> Uninstalling the Linux VDA software deletes the associated PostgreSQL and other configuration data. However, the PostgreSQL package and other dependent packages that were set up before the installation of the Linux VDA are not deleted.
>
> **Tip:**
>
> The information in this section does not cover the removal of dependent packages including PostgreSQL.

Step 5: Run the Linux VDA

Once you have configured the Linux VDA using the **ctxsetup.sh** script, you use the following commands to control the Linux VDA.

Start the Linux VDA:

To start the Linux VDA services:

```
sudo systemctl start ctxhdx

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

Stop the Linux VDA:

To stop the Linux VDA services:


```
sudo systemctl stop ctxvda

sudo systemctl stop ctxhdx
<!--NeedCopy-->
```

> **Note:**
>
> Before you stop the `ctxvda` and `ctxhdx` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

Restart the Linux VDA:

To restart the Linux VDA services:

```
sudo systemctl stop ctxvda

sudo systemctl restart ctxhdx

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

Check the Linux VDA status:

To check the running status of the Linux VDA services:

```
sudo systemctl status ctxvda

sudo systemctl status ctxhdx
<!--NeedCopy-->
```

Step 6: Create the machine catalog in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating machine catalogs and adding Linux VDA machines is similar to the traditional Windows VDA approach. For a more detailed description of how to complete these tasks, see Create machine catalogs and Manage machine catalogs.

For creating machine catalogs that contain Linux VDA machines, there are a few restrictions that differentiate the process from creating machine catalogs for Windows VDA machines:

- For the operating system, select:
  - The **Multi-session OS** option for a hosted shared desktops delivery model.
  - The **Single-session OS** option for a VDI dedicated desktop delivery model.
- Do not mix Linux and Windows VDA machines in the same machine catalog.

> **Note:**
>
> Early versions of Citrix Studio did not support the notion of a "Linux OS."However, selecting the **Windows Server OS** or **Server OS** option implies an equivalent hosted shared desktops delivery model. Selecting the **Windows Desktop OS** or **Desktop OS** option implies a single user per machine delivery model.
>
> **Tip:**
>
> If you remove and rejoin a machine to the Active Directory domain, you must remove and add the machine to the machine catalog again.

Step 7: Create the delivery group in Citrix Virtual Apps or Citrix Virtual Desktops

The process for creating a delivery group and adding machine catalogs containing Linux VDA machines is almost identical to Windows VDA machines. For a more detailed description of how to complete these tasks, see Create Delivery Groups.

For creating delivery groups that contain Linux VDA machine catalogs, the following restrictions apply:

- Ensure that the AD users and groups you select have been properly configured to log on to the Linux VDA machines.
- Do not allow logon of unauthenticated (anonymous) users.
- Do not mix the delivery group with machine catalogs that contain Windows machines.

For information about how to create machine catalogs and delivery groups, see Citrix Virtual Apps and Desktops 7 2103.

Configure the Linux VDA

June 11, 2021

This section details the features of the Linux VDA, including feature description, configuration, and troubleshooting.

> **Tip:**
>
> The `xdlcollect` Bash script used to collect logs is integrated into the Linux VDA software and located under /opt/Citrix/VDA/bin. After you install the Linux VDA, you can run the `bash /opt/Citrix/VDA/bin/xdlcollect.sh` command to collect logs.
>
> After log collection completes, a compressed log file is generated in the same folder as the script.

> `xdlcollect` can ask you whether or not to upload the compressed log file to Citrix Insight Services (CIS). If you agree, `xdlcollect` returns an upload_ID after the upload completes. The upload does not remove the compressed log file from your local machine. Other users can use the upload_ID to access the log file in CIS.

Integrate NIS with Active Directory

June 11, 2021

This article describes how to integrate NIS with Windows Active Directory (AD) on the Linux VDA by using SSSD. The Linux VDA is considered a component of Citrix Virtual Apps and Desktops. As a result, it fits tightly into the Windows AD environment.

Using NIS as a UID and GID provider instead of using AD requires that the account information (user name and password combinations) is the same in both AD and NIS.

> **Note:**
>
> Authentication is still performed by the AD server. NIS+ is not supported. If you use NIS as the UID and GID provider, the POSIX attributes from the Windows server are no longer used.
>
> **Tip:**
>
> This method represents a deprecated way to deploy the Linux VDA, which is used only for special use cases. For an RHEL/CentOS distribution, follow the instructions in Install Linux Virtual Delivery Agent for RHEL/CentOS. For an Ubuntu distribution, follow the instructions in Install Linux Virtual Delivery Agent for Ubuntu.

What is SSSD?

SSSD is a system daemon. Its primary function is to provide access to identify and authenticate remote resources through a common framework that can provide caching and offline support for the system. It provides both PAM and NSS modules, and in the future can support D-BUS based interfaces for extended user information. It also provides a better database to store local user accounts and extended user data.

Integrate NIS with AD

To integrate NIS with AD, do the following:

1. Add the Linux VDA as a NIS client
2. Join the domain and create a host keytab using Samba

3. Set up SSSD
4. Configure NSS/PAM
5. Verify the Kerberos configuration
6. Verify user authentication

Add the Linux VDA as a NIS client

Configure the NIS client:

```
yum –y install ypbind rpcbind oddjob-mkhomedir
<!--NeedCopy-->
```

Set the NIS domain:

```
ypdomainname nis.domain
echo "NISDOMAIN=nis.domain" >> /etc/sysconfig/network
<!--NeedCopy-->
```

Add the IP address for the NIS server and client in **/etc/hosts**:

```
{ NIS server IP address }    server.nis.domain nis.domain
```

Configure NIS by `authconfig`:

```
sudo authconfig --enablenis --nisdomain=nis.domain --nisserver=server.
    nis.domain --enablemkhomedir --update
<!--NeedCopy-->
```

The **nis.domain** represents the domain name of the NIS server. The **server.nis.domain** is the host name of the NIS server, which can also be the IP address of the NIS server.

Configure the NIS services:

```
sudo systemctl start rpcbind ypbind

sudo systemctl enable rpcbind ypbind
<!--NeedCopy-->
```

Ensure that the NIS configuration is correct:

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

Validate that the account information is available from the NIS server:

```
getent passwd nisaccount
<!--NeedCopy-->
```


> **Note:**
>
> The **nisaccount** represents the real NIS account on the NIS server. Ensure that the UID, GID, home directory, and login shell are configured correctly.

Join the domain and create a host keytab using Samba

SSSD does not provide AD client functions for joining the domain and managing the system keytab file. There are a few methods for achieving the functions, including:

- `adcli`
- `realmd`
- `Winbind`
- `Samba`

The information in this section describes the Samba approach only. For `realmd`, see the RHEL or CentOS vendor's documentation. These steps must be followed before configuring SSSD.

Join the domain and create host keytab using Samba:

On the Linux client with properly configured files:

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

Configure the machine for Samba and Kerberos authentication:

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

Where **REALM** is the Kerberos realm name in uppercase and **domain** is the NetBIOS name of the domain.

If DNS-based lookup of the KDC server and realm name is required, add the following two options to the preceding command:

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

Open **/etc/samba/smb.conf** and add the following entries under the **[Global]** section, but after the section generated by the **authconfig** tool:

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

Joining the Windows domain requires that your domain controller is reachable and you have an AD user account with permissions to add computers to the domain:


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

`REALM` is the Kerberos realm name in uppercase and `user` is a domain user who has permissions to add computers to the domain.

Set up SSSD

Setting up SSSD consists of the following steps:

- Install the **sssd-ad** and **sssd-proxy** packages on the Linux client machine.
- Make configuration changes to various files (for example, **sssd.conf**).
- Start the **sssd service**.

/etc/sssd/sssd.conf An example **sssd.conf** configuration (more options can be added as needed):

```
[sssd]
config_file_version = 2
domains = EXAMPLE
services = nss, pam

[domain/EXAMPLE]
# Uncomment if you need offline logins
# cache_credentials = true
re_expression = (((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?P<domain>.+$))|(^(?P<name>[^@\\]+)$))
id_provider = proxy
proxy_lib_name = nis
auth_provider = ad
access_provider = ad

# Should be specified as the long version of the Active Directory domain.
ad_domain = EXAMPLE.COM

# Kerberos settings
krb5_ccachedir = /tmp
krb5_ccname_template = FILE:%d/krb5cc_%U

# Uncomment if service discovery is not working
# ad_server = server.ad.example.com

# Comment out if the users have the shell and home dir set on the AD side
default_shell = /bin/bash
fallback_homedir = /home/%d/%u

```



```
# Uncomment and adjust if the default principal SHORTNAME$@REALM is not
    available
# ldap_sasl_authid = host/client.ad.example.com@AD.EXAMPLE.COM
<!--NeedCopy-->
```

Replace **ad.domain.com**, **server.ad.example.com** with the corresponding value. For more details, see the sssd-ad(5) - Linux man page.

Set the file ownership and permissions on **sssd.conf**:

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

Configure NSS/PAM

RHEL/CentOS:

Use **authconfig** to enable SSSD. Install **oddjob-mkhomedir** to ensure that the home directory creation is compatible with SELinux:

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

sudo systemctl start sssd

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

> **Tip:**
>
> When configuring Linux VDA settings, consider that for SSSD, there has no special settings for the Linux VDA client. For extra solutions in the **ctxsetup.sh** script, use the default value.

Verify the Kerberos configuration

To ensure that Kerberos is configured correctly for use with the Linux VDA, check that the system **keytab** file has been created and contains valid keys:

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

This command displays the list of keys available for the various combinations of principal names and cipher suites. Run the Kerberos **kinit** command to authenticate the machine with the domain controller using these keys:

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


The machine and realm names must be specified in uppercase. The dollar sign ($) must be escaped with a backslash (\) to prevent shell substitution. In some environments, the DNS domain name is different from the Kerberos realm name. Ensure that the realm name is used. If this command is successful, no output is displayed.

Verify that the TGT ticket for the machine account has been cached using:

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

Verify user authentication

Use the **getent** command to verify that the logon format is supported and whether the NSS works:

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

The **DOMAIN** parameter indicates the short version domain name. If another logon format is needed, verify by using the **getent** command first.

The supported logon formats are:

- Down-level logon name: `DOMAIN\username`
- UPN: `username@domain.com`
- NetBIOS Suffix format: `username@DOMAIN`

To verify that the SSSD PAM module is configured correctly, use a domain user account to log on to the Linux VDA. The domain user account has not been used before.

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

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

Check that a corresponding Kerberos credential cache file was created for the **uid** returned by the command:

```
ls /tmp/krb5cc_{
 uid }

<!--NeedCopy-->
```

Check that the tickets in the user's Kerberos credential cache are valid and not expired:

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


Publish applications

June 22, 2022

With Linux VDA Version 7.13, Citrix added the seamless applications feature to all the supported Linux platforms. No specific installation procedures are required to use this feature.

> **Tip:**
>
> With Linux VDA version 1.4, Citrix added support for non-seamless published applications and session sharing.

Publish applications using Citrix Studio

You can publish applications installed on a Linux VDA when you create a delivery group or add applications to an existing delivery group. The process is similar to publishing applications installed on a Windows VDA. For more information, see the Citrix Virtual Apps and Desktops documentation (based on the version of Citrix Virtual Apps and Desktops being used).

> **Tip:**
>
> When configuring delivery groups, ensure that the delivery type is set to **Desktop and applications** or **Applications**.
>
> **Important:**
>
> Publishing applications is supported with Linux VDA Version 1.4 and later. However, the Linux VDA does not support the delivery of desktops and apps to the same machine. To address this issue, we recommend that you create separate delivery groups for app and desktop deliveries.
>
> **Note:**
>
> To use seamless applications, do not disable the seamless mode on StoreFront. The seamless mode is enabled by default. If you have already disabled it by setting "TWIMode=Off,"remove this setting instead of changing it to "TWIMode=On."Otherwise you might not be able to launch a published desktop.

Limitation

The Linux VDA does not support the launch of multiple concurrent instances of the same application by a single user.

Known issues

The following known issues are identified during publishing applications:

- Non-rectangular windows are not supported. The corners of a window might show the server-side background.
- Preview of the content of a window from a published application is not supported.
- Currently, the seamless mode supports the following Window Managers: Mutter, Metacity, and Compiz (Ubuntu 16.04). Kwin and other window managers are not supported. Ensure that your window manager is set a supported one.
- When you run multiple LibreOffice applications, only the one launched first shows on Citrix Studio because these applications share the process.
- Published Qt5-based applications like "Dolphin"might not show icons. To resolve the issue, see the article at https://wiki.archlinux.org/title/Qt.
- All the taskbar buttons of published applications running in the same ICA session are combined in the same group. To resolve this issue, set the taskbar property not to combine taskbar buttons.

Remote PC Access

June 11, 2021

Overview

Remote PC Access is an extension of Citrix Virtual Apps and Desktops. It enables organizations to easily allow employees to access their physical office PCs remotely in a secure manner. If users can access their office PCs, they can access all the applications, data, and resources they need to do their work.

Remote PC Access uses the same Citrix Virtual Apps and Desktops components that deliver virtual desktops and applications. The requirements and process of deploying and configuring Remote PC Access are the same as the requirements and process required for deploying Citrix Virtual Apps and Desktops for the delivery of virtual resources. This uniformity provides a consistent and unified administrative experience. Users receive the best user experience by using Citrix HDX to deliver their remote office PC sessions.

For more information, see Remote PC Access in the Citrix Virtual Apps and Desktops documentation.

Considerations

These considerations are specific to the Linux VDA:

- On physical machines, use the Linux VDA only in non-3D mode. Due to limitations on NVIDIA's driver, the local screen of the PC cannot be blacked out and displays the activities of the session when HDX 3D mode is enabled. Showing this screen is a potential security risk.
- Use machine catalogs of type single-session OS for physical Linux machines.
- Automatic user assignment is not available for Linux machines. With automatic user assignment, users are assigned to their machines automatically when they log on locally to the PCs. This logon occurs without administrator intervention. The Citrix Workspace app running on the client device gives users access to the applications and data on the office PC within the Remote PC Access desktop session.
- If users are already logged on to their PCs locally, attempts to launch the PCs from StoreFront fail.
- Power saving options are not available for Linux machines.

Configuration

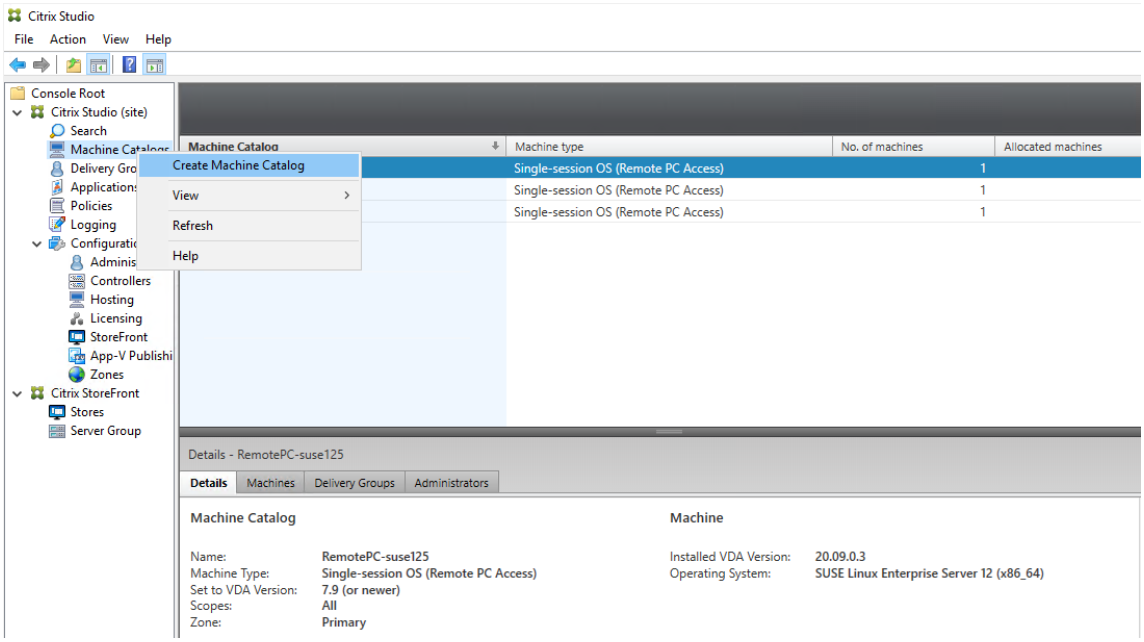
To deliver Linux PC sessions, install the Linux VDA on target PCs, create a machine catalog of the **Remote PC Access** type, and create a Delivery Group to make the PCs in the machine catalog available for users who request access. The following section details the procedure:

Step 1 - Install the Linux VDA on target PCs

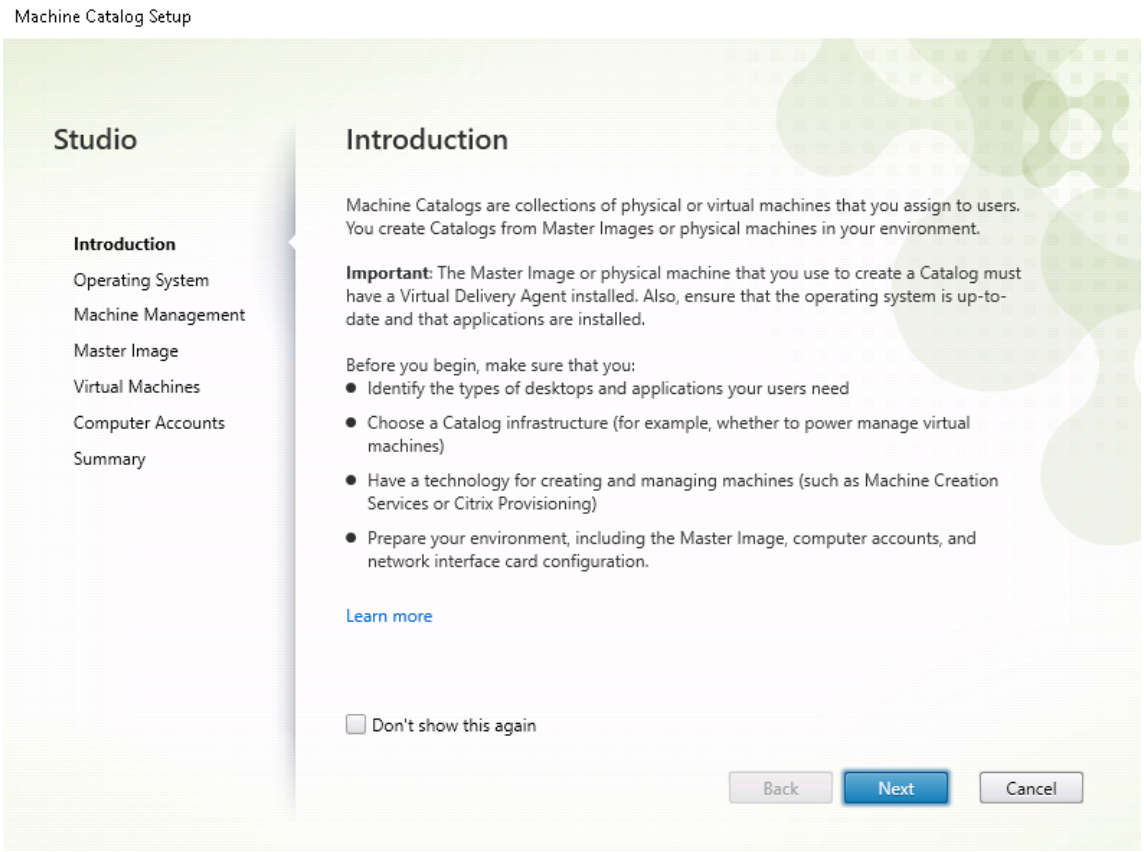
We recommend you use easy install to install the Linux VDA. During the installation, set the value of the `CTX_XDL_VDI_MODE` variable to `Y`.

Step 2 - Create a machine catalog of the Remote PC Access type

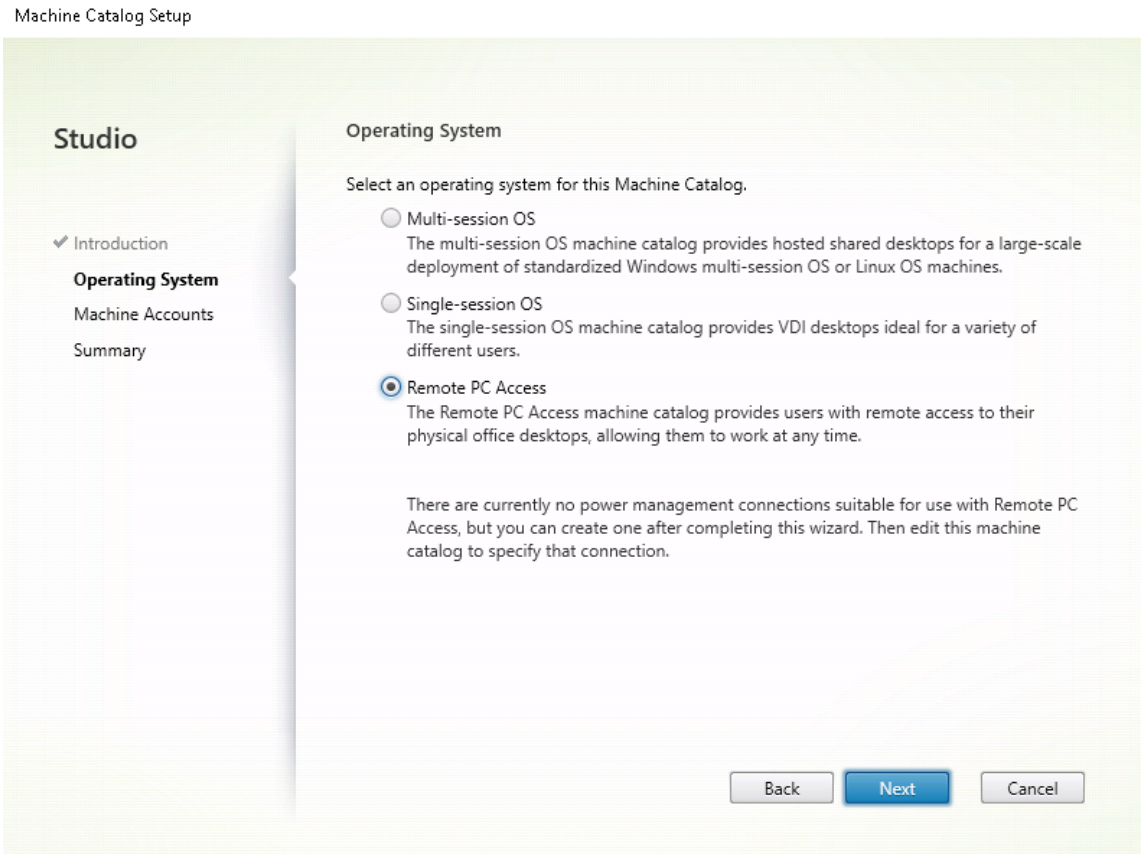
1. In Citrix Studio, right-click **Machine Catalogs** and select **Create Machine Catalog** from the shortcut menu.



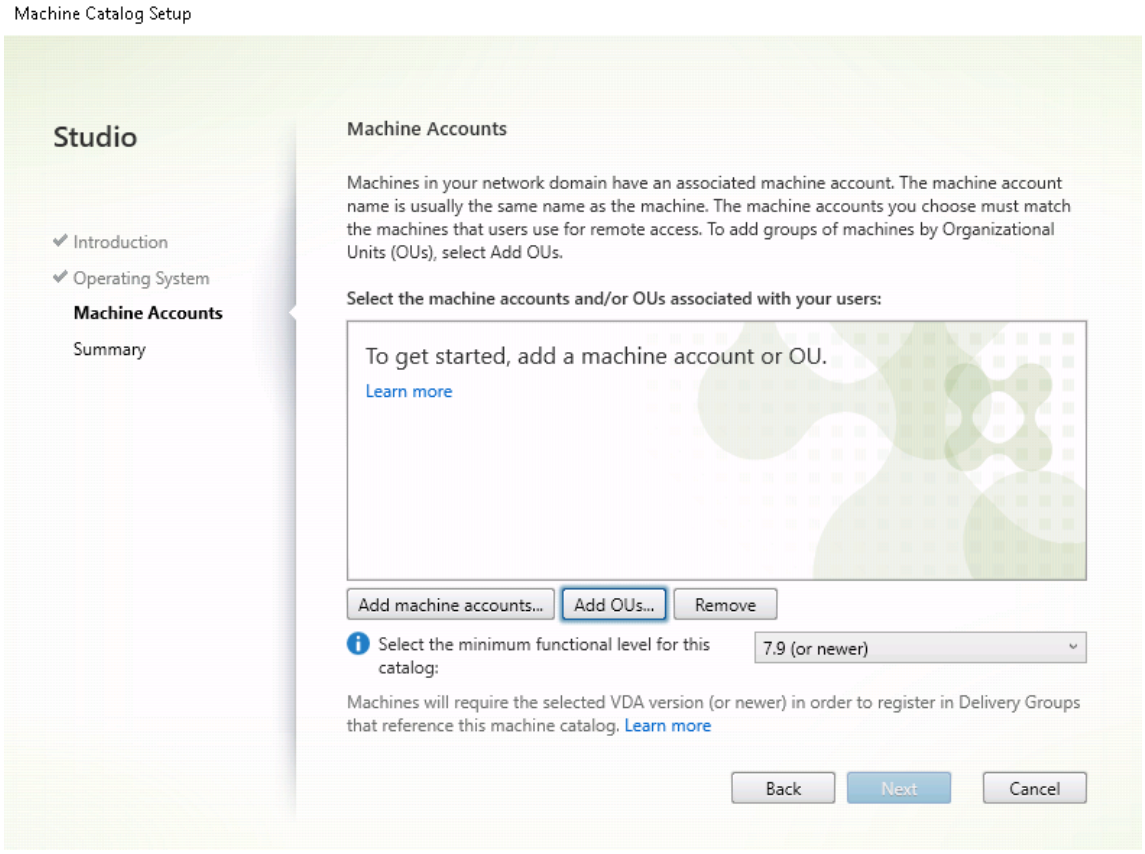
2. Click **Next** on the **Introduction** page.



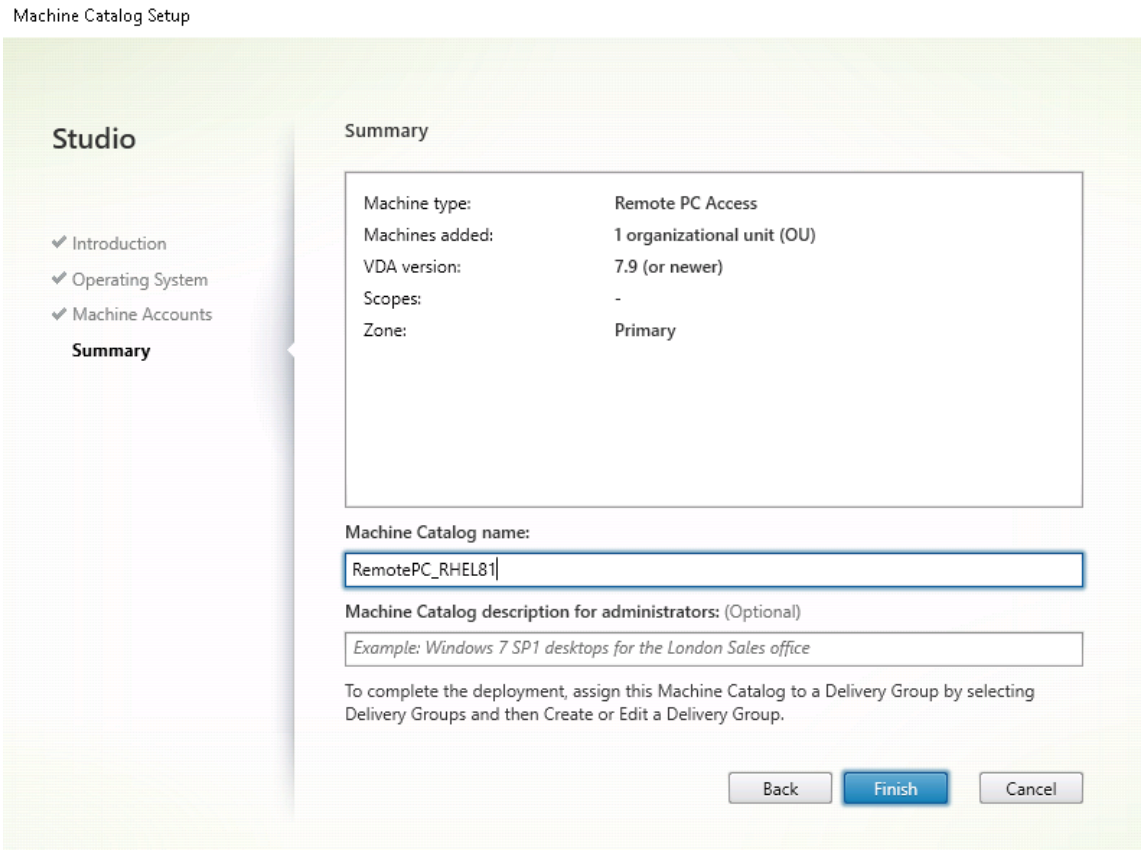
3. Select **Remote PC Access** on the **Operating System** page.



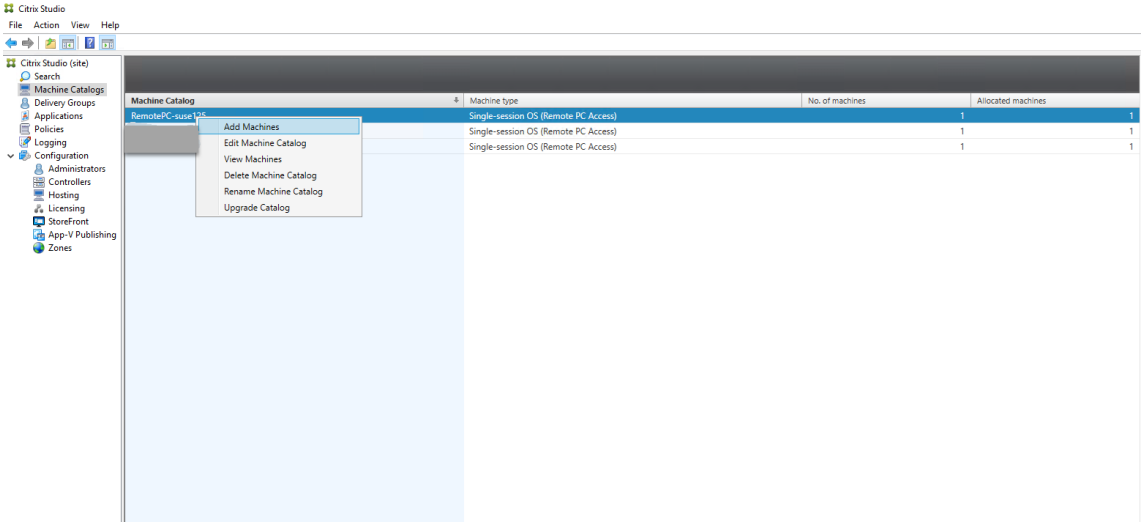
4. Click **Add OUs** to select OUs that contain the target PCs, or click **Add machine accounts** to add individual machines to the machine catalog.



5. Name the machine catalog.

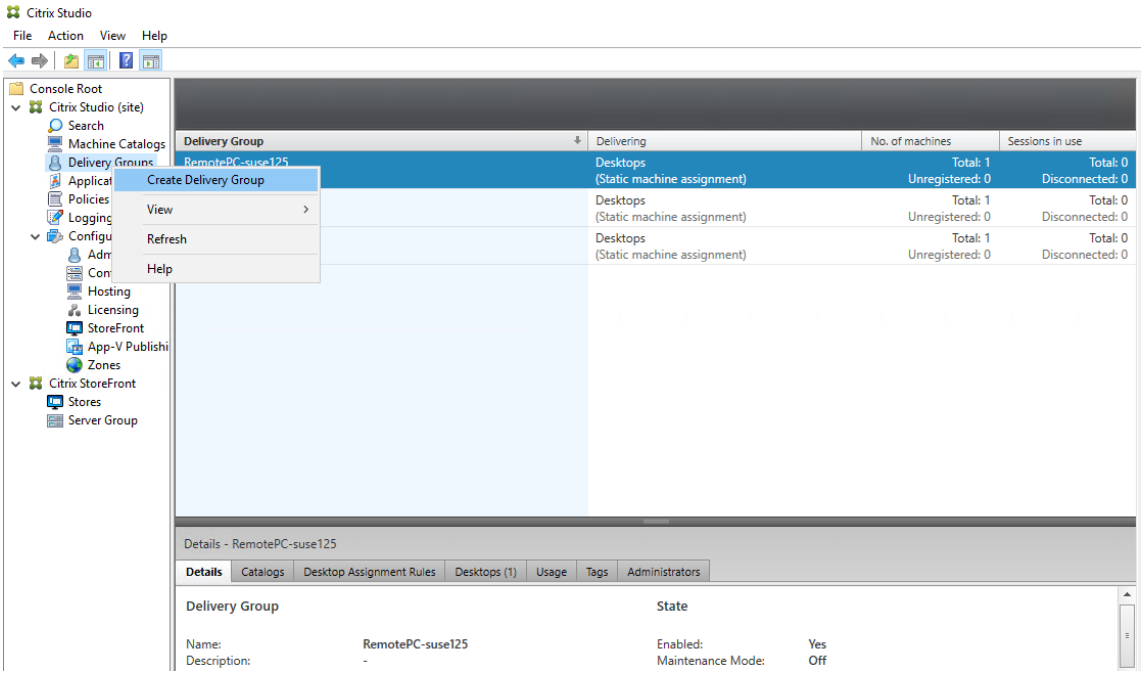


6. (Optional) Right-click the machine catalog to perform relevant operations.

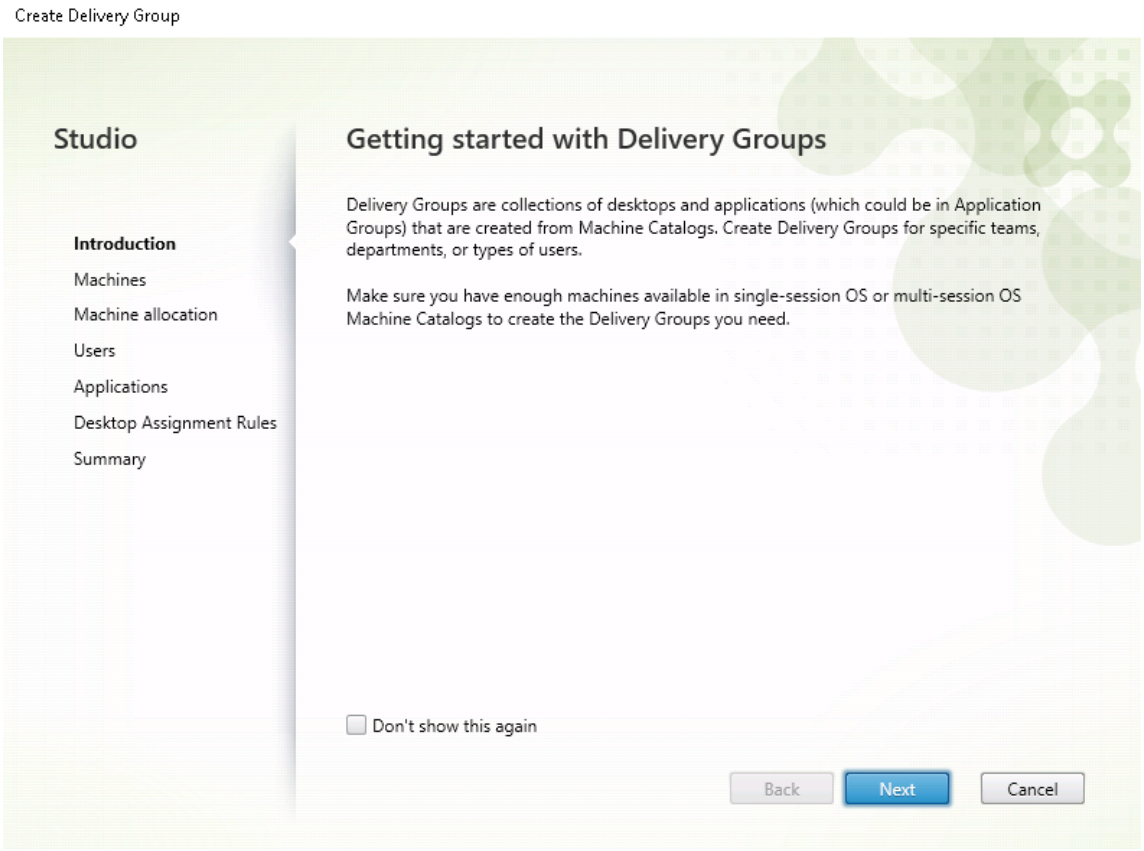


Step 3 - Create a Delivery Group to make the PCs in the machine catalog available for users who request access

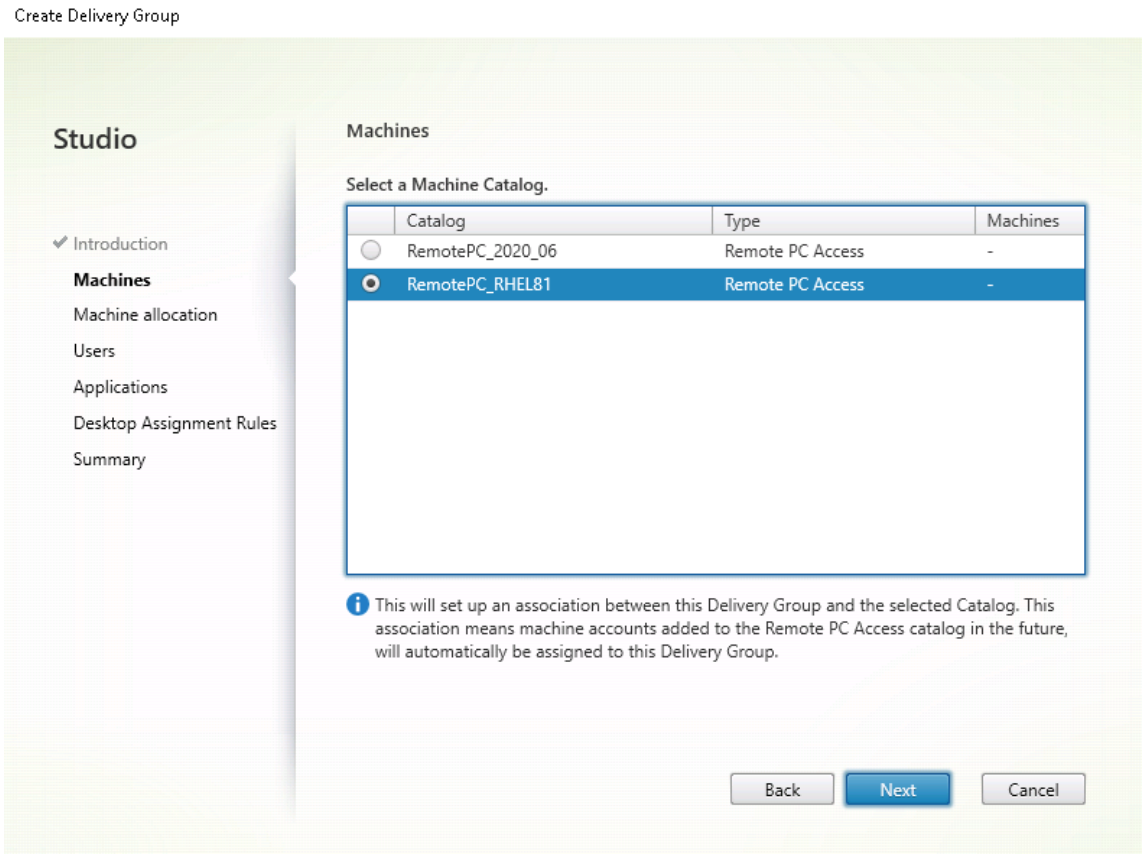
1. In Citrix Studio, right-click **Delivery Groups** and select **Create Delivery Group** from the shortcut menu.



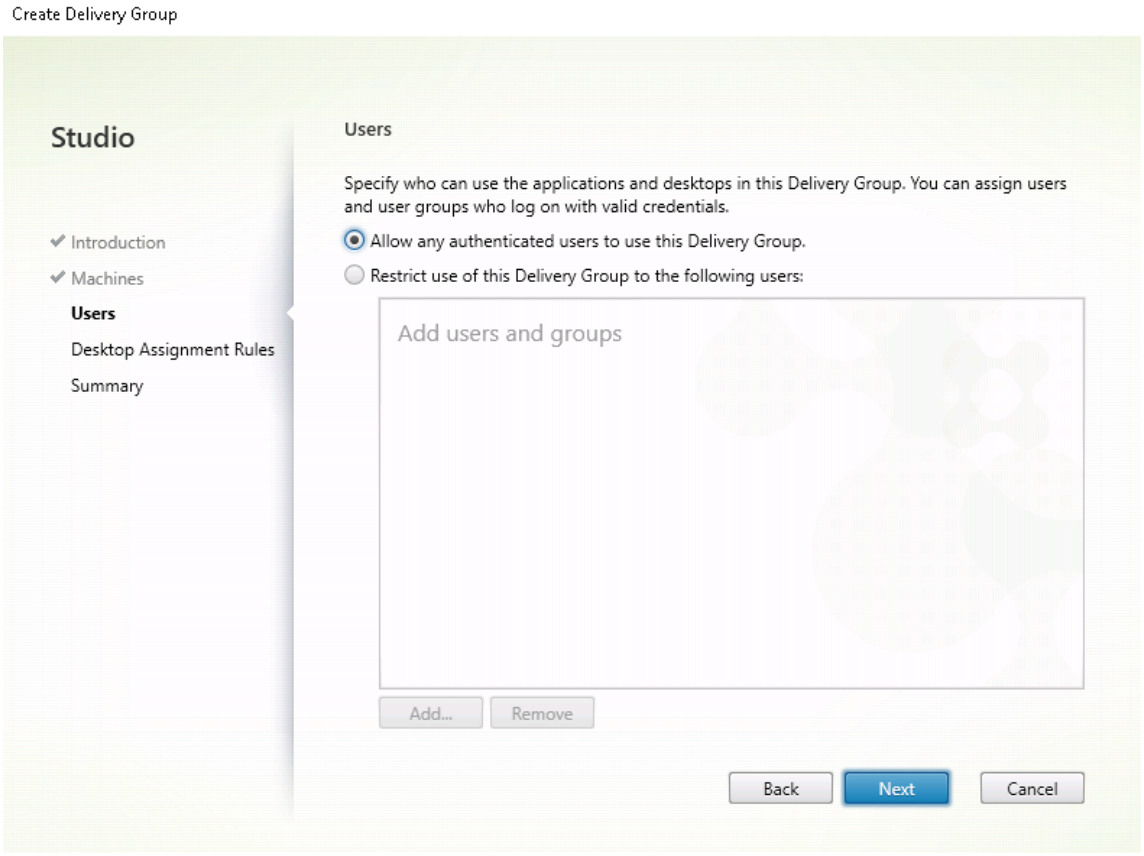
2. Click **Next** on the **Getting started with Delivery Groups** page.



3. Select the machine catalog created in Step 2 to associate it with the Delivery Group.



4. Add users who can access the PCs in the machine catalog. The users you add can use Citrix Workspace app on a client device to access the PCs remotely.



Wake on LAN

Remote PC Access supports Wake on LAN, which gives users the ability to turn on physical PCs remotely. This feature enables users to keep their office PCs turned off when not in use to save energy costs. It also enables remote access when a machine has been turned off inadvertently.

With the Wake on LAN feature, the magic packets are sent directly from the VDA running on the PC to the subnet in which the PC resides when instructed by the delivery controller. This allows the feature to work without dependencies on extra infrastructure components or third-party solutions for delivery of magic packets.

The Wake on LAN feature differs from the legacy SCCM-based Wake on LAN feature. For information on the SCCM-based Wake on LAN, see Wake on LAN –SCCM-integrated.

System requirements

The following are the system requirements for using the Wake on LAN feature:

- Control plane:

  - Citrix Virtual Apps and Desktops Service
  - Citrix Virtual Apps and Desktops 2012 or later

- Physical PCs:
  - VDA version 2012 or later
  - Wake on LAN enabled in BIOS and on the NIC

Configure Wake on LAN

Currently, the configuration of integrated Wake on LAN is only supported using PowerShell.

To configure Wake on LAN:

1. Create the Remote PC Access machine catalog if you do not have one already.
2. Create the Wake on LAN host connection if you do not have one already.

   > **Note:**
   >
   > To use the Wake on LAN feature, if you have a host connection of the "Microsoft Configuration Manager Wake on LAN"type, create a host connection.

3. Retrieve the Wake on LAN host connection's unique identifier.
4. Associate the Wake on LAN host connection with a machine catalog.

   To create the Wake on LAN host connection:

```
# Load Citrix SnapIns
Add-PSSnapIn -Name "*citrix*"

# Provide the name of the Wake on LAN host connection
[string]$connectionName = "Remote PC Access Wake on LAN"

# Create the hypervisor connection
$hypHc = New-Item -Path xdhyp:\Connections `
            -Name $connectionName `
            -HypervisorAddress "N/A" `
            -UserName "woluser" `
            -Password "wolpwd" `
            -ConnectionType Custom `
            -PluginId VdaWOLMachineManagerFactory `
            -CustomProperties "<CustomProperties></
               CustomProperties>" `
            -Persist

$bhc = New-BrokerHypervisorConnection -HypHypervisorConnectionUid
   $hypHc.HypervisorConnectionUid

# Wait for the connection to be ready before trying to use it
```



```
while (-not $bhc.IsReady)
{

    Start-Sleep -s 5
    $bhc = Get-BrokerHypervisorConnection -
       HypHypervisorConnectionUid $hypHc.HypervisorConnectionUid
 }

<!--NeedCopy-->
```

When the host connection is ready, run the following commands to retrieve the host connection's unique identifier:

```
$bhc = Get-BrokerHypervisorConnection -Name "<WoL Connection Name>
    "
$hypUid = $bhc.Uid
<!--NeedCopy-->
```

After you retrieve the connection's unique identifier, run the following commands to associate the connection with the Remote PC Access machine catalog:

```
Get-BrokerCatalog -Name "<Catalog Name>" | Set-BrokerCatalog -
    RemotePCHypervisorConnectionUid $hypUid
<!--NeedCopy-->
```

5. Enable Wake on LAN in BIOS and on the NIC on each VM in the machine catalog.

   Note: The method for enabling Wake on LAN varies with different machine configurations.

   - To enable Wake on LAN in BIOS:
     a) Enter BIOS and enable the Wake on LAN feature.
        The method for accessing BIOS depends on the manufacturer of your motherboard and the BIOS vendor the manufacturer has selected.
     b) Save your settings and restart the machine.
   - To enable Wake on LAN on the NIC:
     a) Run the `sudo ethtool <NIC>` command to check whether your NIC supports magic packets.
        `<NIC>` is the device name of your NIC, for example, `eth0`. The `sudo ethtool <NIC>` command provides output about the capabilities of your NIC:
        - If the output contains a line similar to `Supports Wake-on: <letters>` where `<letters>` contains the letter `g`, your NIC supports the Wake on LAN magic packet method.
        - If the output contains a line similar to `Wake-on: <letters>` where `<letters>` contains the letter `g` and does not contain the letter `d`, the Wake on LAN magic packet method is enabled. However, if `<letters>` contains the

letter `d`, it indicates that the Wake on LAN feature is disabled. In this case, enable Wake on LAN by running the `sudo ethtool -s <NIC> wol g` command.

b) On most distributions, the `sudo ethtool -s <NIC> wol g` command is required after each startup. To persistently set this option, complete the following steps based on your distributions:

Ubuntu:

Add the `up ethtool -s <NIC> wol g` line to the interface configuration file `/etc/network/interfaces`. For example:

```
# ifupdown has been replaced by netplan(5) on this system.
    See
# /etc/netplan for current configuration.
# To re-enable ifupdown on this system, you can run:
# sudo apt install ifupdown
auto eth0
iface eth0 inet static
           address 10.0.0.1
           netmask 255.255.240.0
           gateway 10.0.0.1
           up ethtool -s eth0 wol g
<!--NeedCopy-->
```

RHEL/SUSE:

Add the following `ETHTOOL_OPTS` parameter to the interface configuration file `/etc/sysconfig/network-scripts/ifcfg-<NIC>`:

```
ETHTOOL_OPTS="-s ${
 DEVICE }
  wol g"
<!--NeedCopy-->
```

Design considerations

When you are planning to use Wake on LAN with Remote PC Access, consider the following:

- Multiple machine catalogs can use the same Wake on LAN host connection.
- For a PC to wake up another PC, both PCs must be in the same subnet and use the same Wake on LAN host connection. It does not matter if the PCs are in the same or different machine catalogs.
- Host connections are assigned to specific zones. If your deployment contains more than one zone, you need a Wake on LAN host connection in each zone. The same applies to machine catalogs.
- Magic packets are broadcasted using the global broadcast address 255.255.255.255. Ensure that the address is not blocked.
- There must be at least one PC turned on in the subnet - for every Wake on LAN connection - to be able to wake up machines in that subnet.

Operational considerations

The following are considerations for using the Wake on LAN feature:

- The VDA must register at least once before the PC can be woken up using the integrated Wake on LAN feature.
- Wake on LAN can only be used to wake up PCs. It does not support other power actions, such as restart or shut down.
- After the Wake on LAN connection is created, it is visible in Studio. However, editing its properties within Studio is not supported.
- Magic packets are sent in one of the two ways:
  - When a user tries to launch a session to their PC and the VDA is unregistered
  - When an administrator sends a power on command manually from Studio or PowerShell
- Because the delivery controller is unaware of a PC's power state, Studio displays **Not Supported** under power state. The delivery controller uses the VDA registration state to determine whether a PC is on or off.

More resources

The following are other resources for Remote PC Access:

- Solution design guidance: Remote PC Access Design Decisions.
- Examples of Remote PC Access architectures: Reference Architecture for Citrix Remote PC Access Solution.

Print

August 18, 2022

This article provides information about printing best practices.

Installation

The Linux VDA requires both **cups** and **foomatic** filters. The filters are installed when you install the VDA. You can also install the filters manually based on the distribution. For example:

On RHEL 7:

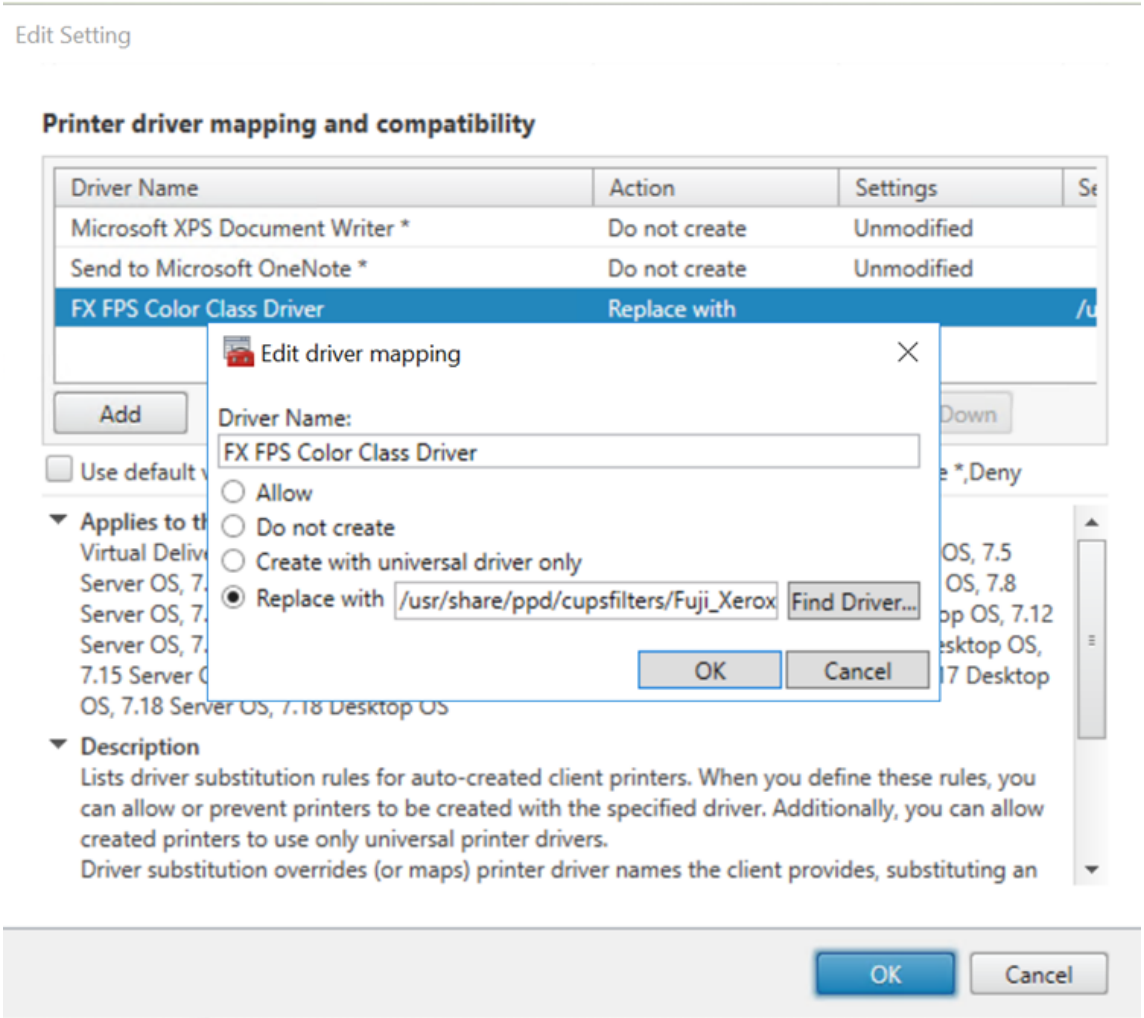

```
sudo yum  – y install cups

sudo yum -y install foomatic-filters
<!--NeedCopy-->
```

Configuration

There are three types of Universal Printer Driver supplied by Citrix (postscript, pcl5, and pcl6). However, the Universal Printer Driver might not be compatible with your client printer. In this case, your only option in earlier releases was to edit the ~/.CtxlpProfile$CLIENT_NAME configuration file. Starting with Version 1906, you can choose to configure the **Printer driver mapping and compatibility** policy in Citrix Studio instead.

To configure the **Printer driver mapping and compatibility** policy in Citrix Studio:

1. Select the **Printer driver mapping and compatibility** policy.
2. Click **Add**.
3. Fill in **Driver name** with the driver name of the client printer. If you are using Citrix Workspace app for Linux, fill in the printer name instead.
4. Choose **Replace with** and type in the absolute path of the driver file on the VDA.



Note:

- Only PPD driver files are supported.
- Other options of the **Printer driver mapping and compatibility** policy are not supported. Only **Replace with** takes effect.

Usage

You can print from both published desktops and published applications. Only the client-side default printer is mapped into a Linux VDA session. The printer names are different for desktops and applications:

- For published desktops:
  `CitrixUniversalPrinter:$CLIENT_NAME:dsk$SESSION_ID`
- For published applications:
  `CitrixUniversalPrinter:$CLIENT_NAME:app$SESSION_ID`

> **Note:**
>
> If the same user opens both a published desktop and a published application, both printers are available to the session. Printing on a desktop printer in a published application session, or printing on an application printer in a published desktop fails.

Troubleshooting

Unable to print

When printing is not working correctly, check the print daemon, **ctxlpmngt**, and the CUPs framework.

The print daemon, **ctxlpmngt**, is a per-session process and must be running for the length of the session. Run the following command to verify that the printing daemon is running. If **ctxlpmngt** is not running, start **ctxlpmngt** manually from a command line.

```
ps –ef | grep ctxlpmngt
<!--NeedCopy-->
```

If printing is still not working, check the CUPS framework. The **ctxcups** service is used for printer management and communicates with the Linux CUPS framework. It is a single process per machine and can be checked by running the following command:

```
service ctxcups status
<!--NeedCopy-->
```

Extra steps for collecting CUPS logs

To collect CUPS logs, run the following commands to configure the CUPS service file. Otherwise, CUPS logs cannot be recorded in **hdx.log**:

```
sudo service cups stop

sudo vi /etc/systemd/system/printer.target.wants/cups.service

PrivateTmp=false

sudo service cups start

sudo systemctl daemon-reload
<!--NeedCopy-->
```


> **Note:**
>
> This configuration is made only for collecting the full printing log when an issue arises. Under normal circumstances, this configuration is not recommended because it breaks CUPS security.

Print output is garbled

An incompatible printer driver can cause garbled output. A per-user driver configuration is available and can be configured by editing the **~/.CtxlpProfile$CLIENT_NAME** configuration file:

```
[DEFAULT_PRINTER]

printername=

model=

ppdpath=

drivertype=
<!--NeedCopy-->
```

> **Important:**
>
> The **printername** is a field containing the name of the current client-side default printer. It is a read-only value. Do not edit it.
>
> The fields **ppdpath**, **model**, and **drivertype** cannot be set at the same time because only one takes effect for the mapped printer.

- If the Universal Printer driver is not compatible with the client printer, configure the model of the native printer driver using the **model**= option. You can find the current model name of the printer by using the **lpinfo** command:

  ```
  lpinfo  – m

  …

  xerox/ph3115.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

  xerox/ph3115fr.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0
  xerox/ph3115pt.ppd.gz Xerox Phaser 3115, SpliX V. 2.0.0

  <!--NeedCopy-->
  ```

  You can then set the model to match the printer:

  ```
  model=xerox/ph3115.ppd.gz
  <!--NeedCopy-->
  ```


- If the Universal Printer driver is not compatible with the client printer, configure the PPD file path of the native printer driver. The value of **ppdpath** is the absolute path of the native printer driver file.

  For example, there is a **ppd driver** under /home/tester/NATIVE_PRINTER_DRIVER.ppd:

  ```
  ppdpath=/home/tester/NATIVE_PRINTER_DRIVER.ppd
  <!--NeedCopy-->
  ```

- There are three types of Universal Printer Driver supplied by Citrix (postscript, pcl5, and pcl6). You can configure the driver type based on your printer properties.

  For example, if the client default printer driver type is PCL5, set **drivertype** to:

  ```
  drivertype=pcl5
  <!--NeedCopy-->
  ```

Output size is zero

Try different types of printers. And try a virtual printer like CutePDF and PDFCreator to find out whether this issue is related to the printer driver.

The print job depends on the printer driver of the client default printer. It's important to identify the type of the current active driver type. If the client printer is using a PCL5 driver but the Linux VDA chooses a Postscript driver, an issue can occur.

If the printer driver type is correct, you can identify the problem by performing the following steps:

1. Log on to a published desktop session.
2. Run the **vi ~/.CtxlpProfile$CLIENT_NAME** command.
3. Add the following field to save the spool file on the Linux VDA:

   ```
   deletespoolfile=no
   <!--NeedCopy-->
   ```

4. Log off and back on to load the configuration changes.
5. Print the document to reproduce the issue. After printing, a spool file is saved under **/var/spool/cups-ctx/$logon_user/$spool_file**.
6. Check whether the spool is empty. If the spool file is zero, it represents an issue. Contact Citrix Support (and provide the printing log) for more guidance.
7. If the spool size is not zero, copy the file to the client. The spool file content depends on the printer driver type of the client default printer. If the mapped printer (native) driver is postscript, the spool file can be opened in the Linux OS directly. Check whether the content is correct.

If the spool file is PCL, or if the client OS is Windows, copy the spool file to the client and print it on the client-side printer by using a different printer driver.

8. Change the mapped printer to use a different printer driver. The following example uses the postscript client printer as an example:

   a) Log on to an active session and open a browser on the client desktop.

   b) Open the printing management portal:

   ```
   localhost:631
   <!--NeedCopy-->
   ```

   c) Choose the mapped printer **CitrixUniversalPrinter:$ClientName:app/dsk$SESSION_ID** and **Modify Printer**. This operation requires administrator privileges.

   d) Retain the cups-ctx connection, then click Continue to change the printer driver.

   e) In the **Make** and **Model** fields, choose a different printer driver from the Citrix UPD driver. For example, if the CUPS-PDF virtual printer is installed, select the Generic CUPS-PDF Printer driver. Save the change.

   f) If this process succeeds, configure the PPD file path of the driver in **.CtxlpProfile$CLIENT_NAME** to allow the mapped printer to use the newly selected driver.

Known issues

The following issues have been identified during printing on the Linux VDA:

CTXPS driver is not compatible with some PLC printers

If you encounter printing output corruption, set the printer driver to the native one provided by the manufacturer.

Slow printing performance for large documents

When you print a large document on a local client printer, the document is transferred over the server connection. On slow connections, the transfer can take a long time.

Printer and print job notifications seen from other sessions

Linux does not have the same session concept as the Windows operating system. Therefore, all users get system-wide notifications. You can disable these notifications by changing the CUPS configuration file: **/etc/cups/cupsd.conf**.

Locate the current policy name configured in the file:

```
DefaultPolicy default
```

If the policy name is *default*, add the following lines to the default policy XML block:

```
<Policy default>

     # Job/subscription privacy...

     JobPrivateAccess default

     JobPrivateValues default

     SubscriptionPrivateAccess default

     SubscriptionPrivateValues default

     ... ...

     <Limit Create-Printer-Subscription>

          Require user @OWNER

          Order deny,allow

     </Limit>

     <Limit All>

          Order deny,allow

     </Limit>

</Policy>
<!--NeedCopy-->
```

File copy and paste

June 11, 2021

Users can copy and paste files between a session and a local client. This feature requires Citrix Virtual Apps and Desktops 2006 or later and Citrix Workspace app 1903 or later for Windows. Copy and paste functions by using the right-click menu or keyboard shortcuts.

To successfully copy and paste files, ensure that:

- The maximum number of files does not exceed 20.
- The maximum file size does not exceed 200 MB.

Supported platforms

The file copy and paste feature is available for:

- RHEL 7.8
- SLES 12.5
- Ubuntu 16.04
- Ubuntu 18.04
- Debian 10

Relevant policies

The following clipboard policies are relevant to configuring the feature. For more information about the clipboard policies, see Policy support list.

- Client clipboard redirection
- Clipboard selection update mode
- Primary selection update mode

> **Note:**
>
> To disable the file copy and paste feature, set the **Client clipboard redirection** policy to **Prohibited** in Citrix Studio.

Limitations

- Cut is not supported. Requests to cut a file are treated as copy.
- Drag and drop is not supported.
- Copying directories is not supported.

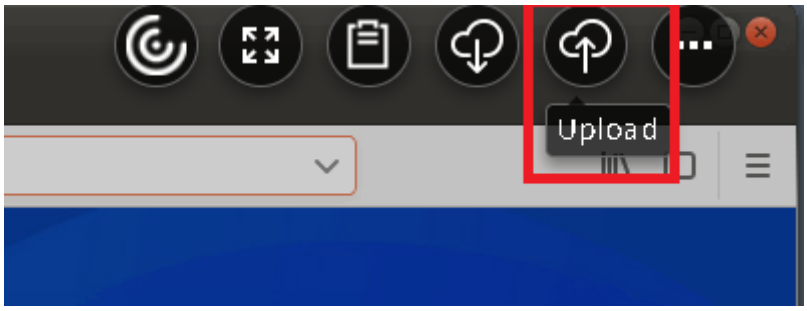
File transfer

June 11, 2021

File transfer is supported between the Linux VDA and the client device. This feature is available when the client device runs a web browser that supports the HTML5 sandbox attribute. The HTML5 sandbox attribute allows users to access virtual desktops and apps using Citrix Workspace app for HTML5 and for Chrome.

> **Note:**
>
> File transfer is available for Citrix Workspace app for HTML5 and for Chrome.

Within published app and desktop sessions, file transfer allows file uploads and downloads between the Linux VDA and the client device. To upload files from the client device to the Linux VDA, click the **Upload** icon on the toolbar of Citrix Workspace app and select the file you want from the file dialogs. To download files from the Linux VDA to the client device, click the **Download** icon. You can add files during uploading or downloading. You can transfer up to 100 files at any one time.



> **Note:**
>
> To upload and download files between the Linux VDA and the client device, ensure that the toolbar of Citrix Workspace app is enabled.
> You can use a version of Citrix Workspace app that lets you drag and drop files.

Auto-download is an enhancement for file transfer. Files you download or move to the **Save to My Device** directory on the VDA are transferred to the client device automatically.

> **Note:**
>
> Auto-download requires the **Allow file transfer between desktop and client** and **Download file from desktop** policies to be set to **Allowed**.

Here are some use cases for auto-download:

- Download files to **Save to My Device**

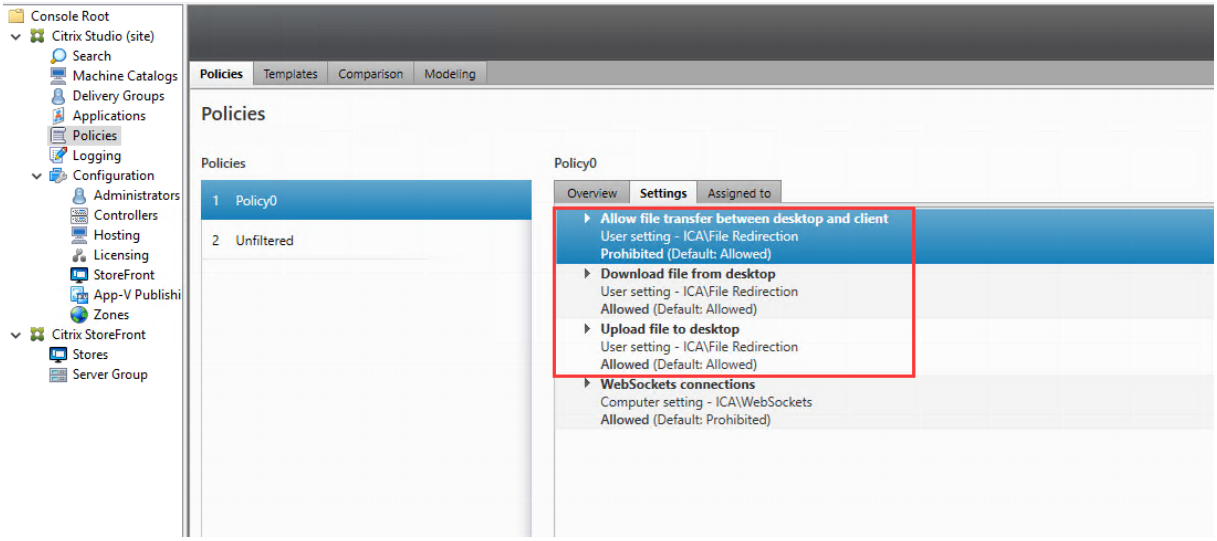
  Within published desktop and web browser app sessions, files you download from websites can be saved to the **Save to My Device** directory on the VDA for automatic transfer to the client device. To achieve auto-download, set the default download directory of the in-session web browser to **Save to My Device** and set a local download directory in the web browser that runs your Citrix Workspace app for HTML5 or for Chrome.

- Move or copy files to **Save to My Device**

  Within published desktop sessions, choose the target files and move or copy them to the **Save to My Device** directory for availability on the client device.

File transfer policies

You can use Citrix Studio to set the file transfer policies. By default, file transfer is enabled.



Policy descriptions:

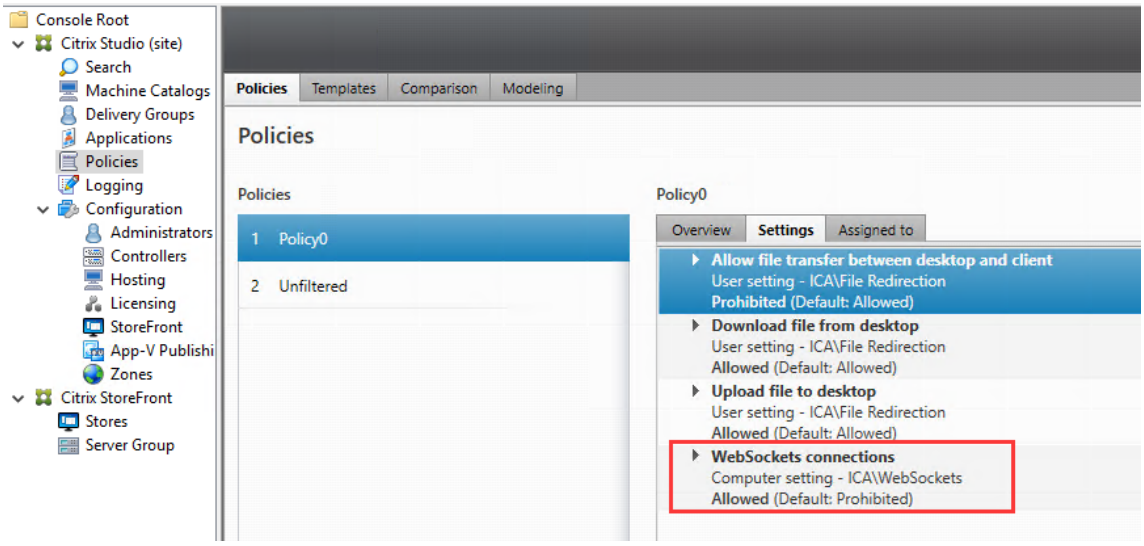
- **Allow file transfer between desktop and client.** Allows or prevents users from transferring files between a Citrix Virtual Apps and Desktops session and their devices.
- **Download file from desktop.** Allows or prevents users from downloading files from a Citrix Virtual Apps and Desktops session to their device.
- **Upload file to desktop.** Allows or prevents users from uploading files from their device to a Citrix Virtual Apps and Desktops session.

> **Note:**
>
> To ensure that the **Download file from desktop** and **Upload file to desktop** policies take effect, set the **Allow file transfer between desktop and client** policy to **Allowed**.

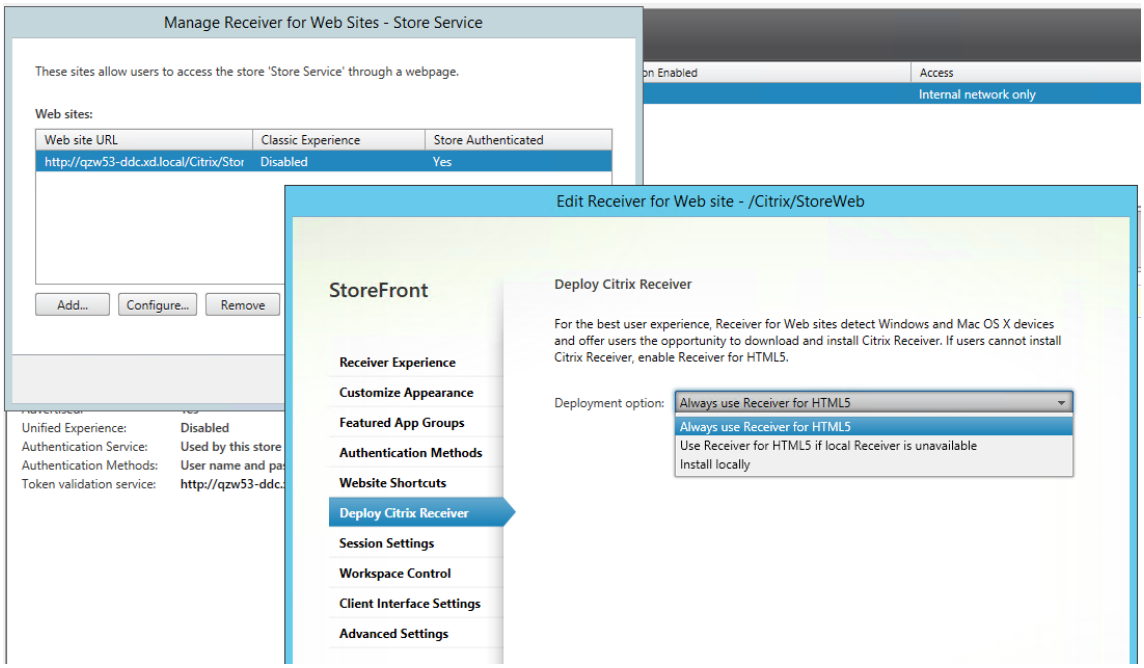
Usage

To use the file transfer feature through Citrix Workspace app for HTML5:

1. In Citrix Studio, set the **WebSockets connections** policy to **Allowed**.



2. In Citrix Studio, enable file transfer through the file transfer policies described earlier.
3. In the Citrix StoreFront management console, click **Stores**, select the **Manage Receiver for Web Sites** node, and enable Citrix Receiver for HTML5 by selecting the **Always use Receiver for HTML5** option.



4. Launch a virtual desktop or web browser app session. Perform one or more file transfers between the Linux VDA and your client device.

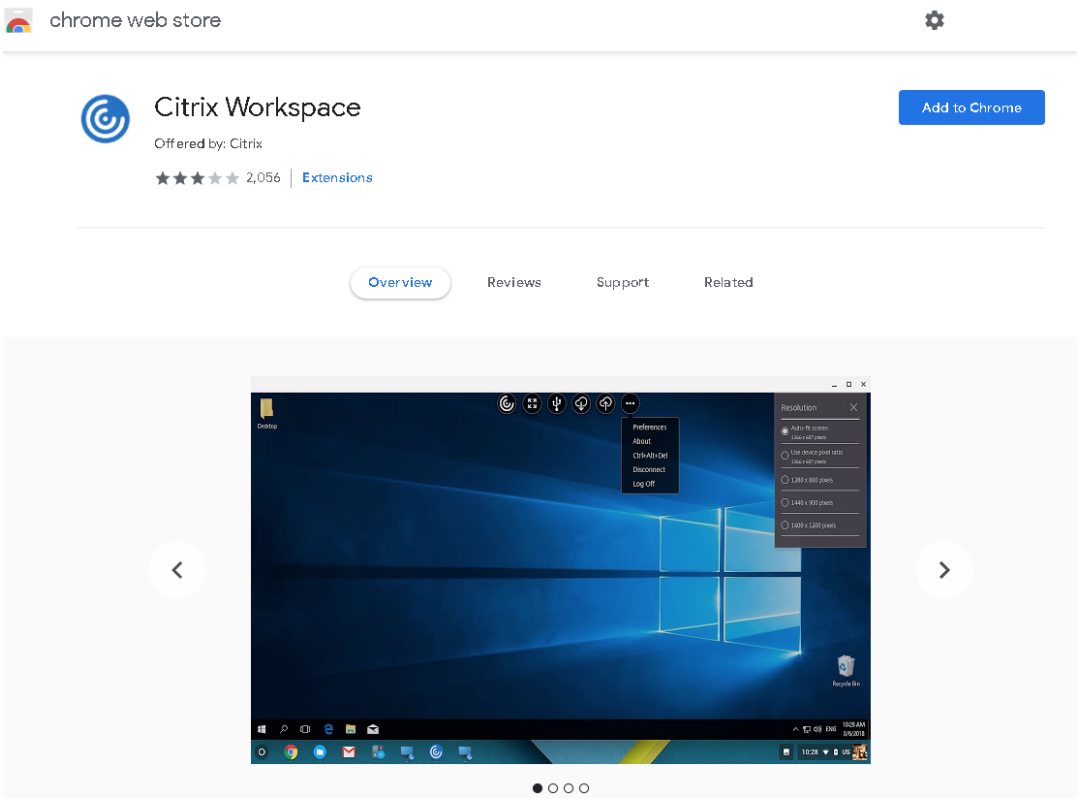
To use the file transfer feature through Citrix Workspace app for Chrome:

1. Enable file transfer through the file transfer policies described earlier.
2. Obtain Citrix Workspace app from the Chrome Web Store.

Skip this step if you already added Citrix Workspace app for Chrome to the Chrome Apps page.

a) Type **Citrix Workspace for Chrome** in the search box of Google Chrome. Click the search icon.

b) Among the search results, click the URL to the Chrome Web Store where Citrix Workspace app is available.



c) Click **Add to Chrome** to add Citrix Workspace app to Google Chrome.

3. Click Citrix Workspace app for Chrome on the Chrome Apps page.
4. Type the URL of your StoreFront store to connect.

   Skip this step if you typed the URL before.
5. Launch a virtual desktop or app session. Perform one or more file transfers between the Linux VDA and your client device.

PDF printing

June 11, 2021

Using a version of Citrix Workspace app that supports PDF printing, you can print PDFs converted from within the Linux VDA sessions. Session print jobs are sent to the local machine where Citrix Workspace app is installed. On the local machine, you can open PDFs using your PDF viewer of choice and print them on your printer of choice.

The Linux VDA supports PDF printing on the following versions of Citrix Workspace app:

- Citrix Receiver for HTML5 Versions 2.4 through 2.6.9, Citrix Workspace app 1808 for HTML5 and later
- Citrix Receiver for Chrome Versions 2.4 through 2.6.9, Citrix Workspace app 1808 for Chrome and later
- Citrix Workspace app 1905 for Windows and later

Configuration

Apart from using a version of Citrix Workspace app that supports PDF printing, enable the following policies in Citrix Studio:

- **Client Printer Redirection** (enabled by default)
- **Auto-create PDF Universal Printer** (disabled by default)

With these policies enabled, a print preview appears on the local machine for you to select a printer when you click **Print** within your launched session. See the Citrix Workspace app documentation for information about setting default printers.

Configure graphics

February 21, 2023

This article provides guidance for the Linux VDA graphics configuration and fine-tuning.

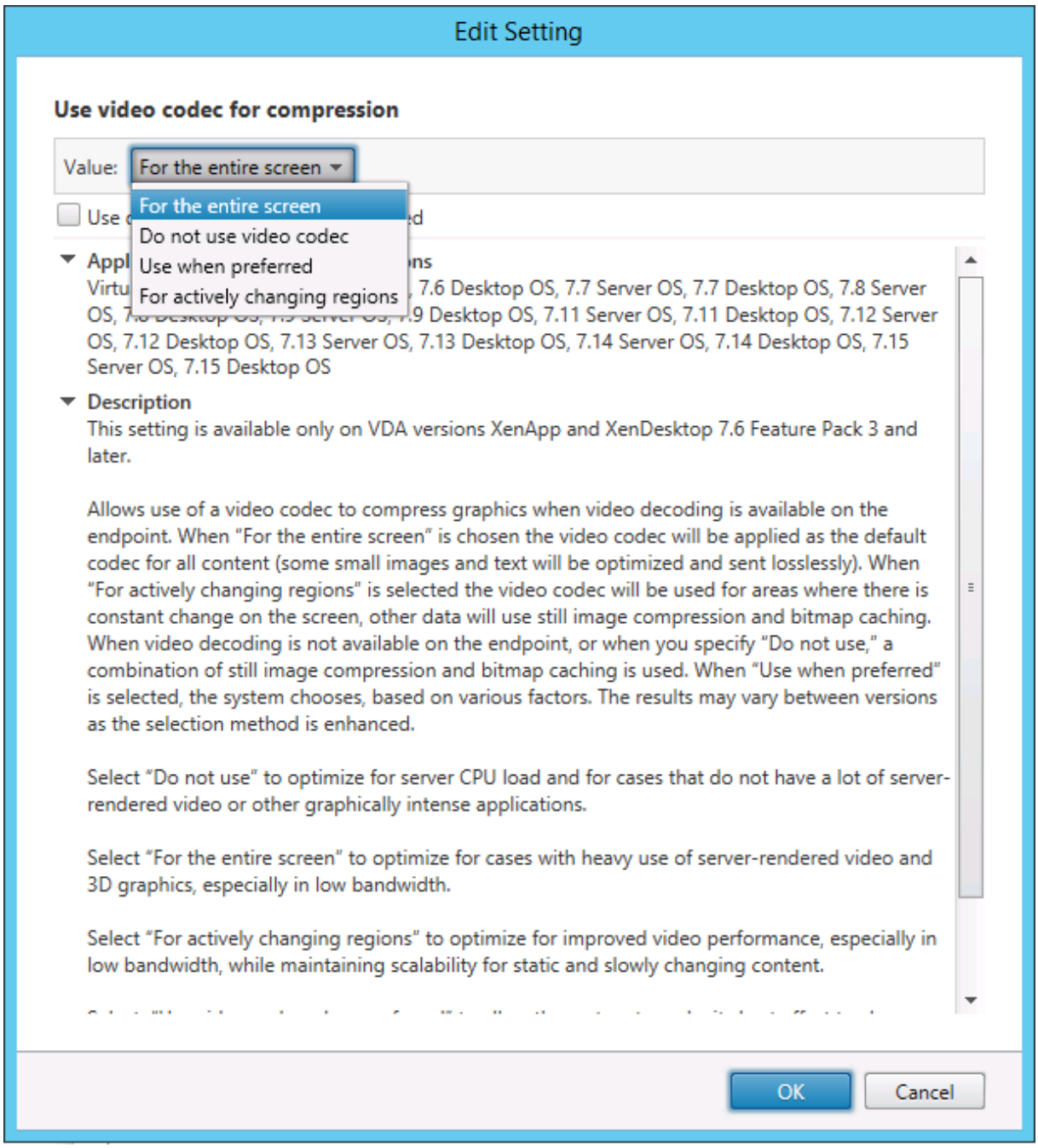
For more information, see System requirements and the Installation overview section.

Configuration

Thinwire is the display remoting technology used in the Linux VDA. The technology allows graphics generated on one machine to be transmitted, typically across a network, to another machine for display.

The **Use video codec for compression** graphics policy sets the default graphics mode and provides the following options for different use cases:

- **Use when preferred**. This setting is the default. No additional configuration is required. Keeping this setting ensures that Thinwire is selected for all Citrix connections, and optimized for scalability, bandwidth, and superior image quality for typical desktop workloads.
- **For the entire screen**. Delivers Thinwire with full-screen H.264 or H.265 to optimize for improved user experience and bandwidth, especially in cases with heavy use of 3D graphics.
- **For actively changing regions**. The adaptive display technology in Thinwire identifies moving images (video, 3D in motion), and uses H.264 only in the part of the screen where the image is moving. The ***selective use of the H.264 video codec*** enables HDX Thinwire to detect and encode parts of the screen that are frequently updated using the H.264 video codec, for example, video content. Still image compression (JPEG, RLE) and bitmap caching continue to be used for the rest of the screen, including text and photographic imagery. Users get the benefit of lower bandwidth and better quality for video content combined with lossless text or high quality imagery elsewhere. To enable this feature, change the policy setting **Use video codec for compression** to **Use when preferred** (default) or **For actively changing regions**. For more information, see Graphics policy settings.



Some other policy settings, including the following visual display policy settings can be used to fine-tune the performance of display remoting:

- **Preferred color depth for simple graphics**
- **Target frame rate**
- **Visual quality**

Use H.264 for Build to Lossless in Thinwire

By default, the **Build to Lossless** preference of the **Visual quality** policy setting is now H.264 instead of JPEG for moving images.

H.264 encoding offers superior image quality. The **Use video codec for compression** policy controls that preference, with the default being **Use when preferred**. To force **Build to Lossless** to use JPEG, set the **Use video codec for compression** policy to **Do not use video codec**. If your client does not support Selective H.264, **Build to Lossless** falls back to JPEG regardless of the policy settings. Citrix Receiver for Windows 4.9 through 4.12, Citrix Receiver for Linux 13.5 through 13.10, Citrix Workspace app 1808 for Windows and later, and Citrix Workspace app 1808 for Linux and later support Selective H.264. For more information about the **Visual quality** and **Use video codec for compression** policy settings, see Visual display policy settings and Graphics policy settings.

Support for H.265 video codec

Starting with the 7.18 release, the Linux VDA supports the H.265 video codec for hardware acceleration of remote graphics and videos. You can use this feature on Citrix Receiver for Windows 4.10 through 4.12 and on Citrix Workspace app 1808 for Windows and later. To benefit from this feature, enable it on both the Linux VDA and on your client. If the GPU of your client does not support H.265 decoding using the DXVA interface, the H.265 Decoding for graphics policy setting is ignored and the session falls back to using the H.264 video codec. For more information, see H.265 video encoding.

To enable H.265 hardware encoding on the VDA:

1. Enable the **Use hardware encoding for video codec** policy.
2. Enable the **Optimize for 3D graphics workload** policy
3. Ensure that the **Use video codec for compression** policy is default or set to **For the entire screen**.
4. Ensure that the **Visual quality** policy is **NOT** set to **Build to Lossless** or **Always Lossless**.

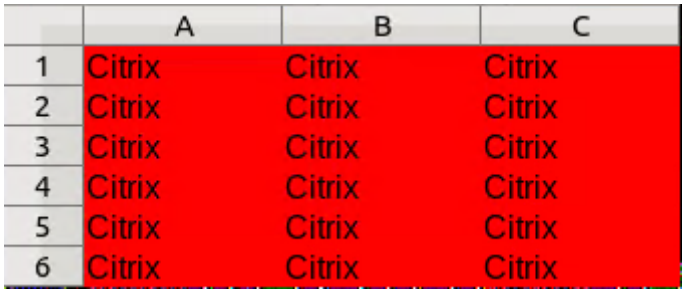
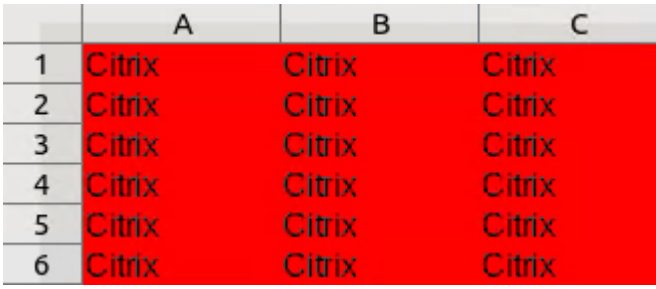
To enable H.265 hardware encoding on your client, see H.265 video encoding.

Support for YUV444 software encoding

The Linux VDA supports YUV444 software encoding. The YUV encoding scheme assigns both brightness and color values to each pixel. In YUV, 'Y'represents the brightness, or 'luma'value, and 'UV' represents the color, or 'chroma'values. You can use this feature of the Linux VDA on Citrix Receiver for Windows 4.10 through 4.12 and on Citrix Workspace app 1808 for Windows and later.

Each unique Y, U, and V value comprises 8 bits, or one byte, of data. The YUV444 data format transmits 24 bits per pixel. The YUV422 data format shares U and V values between two pixels, which results in

an average transmission rate of 16 bits per pixel. The following table shows an intuitive comparison between YUV444 and YUV420.

YUV444	YUV420
	

To enable YUV444 software encoding on the VDA:

1. Ensure that the **Use video codec for compression** policy is set to **For the entire screen**.
2. Ensure that the **Visual quality** policy is set to **Always Lossless** or **Build to Lossless**.

Adjust average bit rates based on bandwidth estimates

Citrix enhances HDX 3D Pro hardware encoding by adjusting average bit rates based on bandwidth estimates.

When the HDX 3D Pro hardware encoding is in use, the VDA can intermittently estimate the bandwidth of the network and adjust the bit rates of encoded frames based on the bandwidth estimates. This new feature provides a mechanism to balance between sharpness and fluency.

This feature is enabled by default. To disable it, run the following command:

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    DisableReconfigureEncoder" -d "0x00000001" --force
<!--NeedCopy-->
```

In addition to using this feature, you can also run the following commands to adjust between sharpness and fluency. The **AverageBitRatePercent** and **MaxBitRatePercent** parameters set the percentage of bandwidth usage. The higher values you set, the sharper graphics and lower fluency you get. The recommended setting range is 50–100.

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    AverageBitRatePercent" -d "90" --force

sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    MaxBitRatePercent" -d "100" --force
```



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

In the average bit rate adjustment, when your screen holds still, the most recent frame stays in a low-quality state because no new frames are sent. Sharpening support can address this issue by reconfiguring and immediately sending the most recent frame at the highest quality.

For a full list of the policies supported by the Linux VDA Thinwire, see Policy support list.

For information on the configuration of multi-monitor support on the Linux VDA, see CTX220128.

Troubleshooting

Check which graphics mode is in use

Run the following command to check which graphics mode is in use (**0** means TW+; **1** means full-screen video codec):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep GraphicsMode
<!--NeedCopy-->
```

The result resembles:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "GraphicsMode"-d "0x00000000"--force
```

Check whether H.264 is in use

Run the following command to check whether H.264 is in use (**0** means not in use; **1** means in use):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H264
<!--NeedCopy-->
```

The result resembles:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H264"-d "0x00000000"--force
```

Check whether H.265 is in use

Run the following command to check whether full-screen H.265 is in use (**0** means not in use; **1** means in use):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep H265
<!--NeedCopy-->
```


The result resembles:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "H265"-d "0x00000000"--force
```

Check which YUV encoding scheme is in use

Run the following command to check which YUV encoding scheme is in use (**0** means YUV420. **1** means YUV422. **2** means YUV444):

Note: The value of YUVFormat is meaningful only when a video codec is in use.

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep YUVFormat
<!--NeedCopy-->
```

The result resembles:

```
create -k "HKLM\Software\Citrix\Ica\Session\4\Graphics"-t "REG_DWORD"
-v "YUVFormat"-d "0x00000000"--force
```

Check whether YUV444 software encoding is in use

Run the following command to check whether YUV444 software encoding is in use:

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep Graphics
<!--NeedCopy-->
```

When YUV444 is in use, the result resembles:

```
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
```

Check whether hardware encoding is in use for 3D Pro

Run the following command (**0** means not in use; **1** means in use):

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep HardwareEncoding
<!--NeedCopy-->
```


The results resemble:

```
create -k "HKLM\Software\Citrix\Ica\Session\1\Graphics"-t "REG_DWORD"
-v "HardwareEncoding"-d "0x00000001"--force
```

Another way is to use the **nvidia-smi** command. The outputs resemble the following if hardware encoding is in use:

```
Tue Apr 12 10:42:03 2016
+-----------------------------------------------------+
| NVIDIA-SMI 361.28     Driver Version: 361.28        |
|-------------------------------+----------------------+----------------------+

| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile
   Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util
   Compute M. |
|===============================+======================+======================|

|   0  GRID K1             Off  | 0000:00:05.0     Off |
                      N/A |
| N/A   42C    P0    14W /  31W |    207MiB /  4095MiB |      8%
   Default |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+

| Processes:                                                       GPU
   Memory |
|  GPU       PID  Type  Process name
   Usage      |
|=============================================================================|

|    0      2164  C+G   /usr/local/bin/ctxgfx
   106MiB |
|    0      2187    G   Xorg
    85MiB |
+-----------------------------------------------------------------------------+

<!--NeedCopy-->
```

Verify that the NVIDIA GRID graphics driver is installed correctly

To verify that the NVIDIA GRID graphics driver is installed correctly, run **nvidia-smi**. The results resemble:

```
+-----------------------------------------------------+
| NVIDIA-SMI 352.70     Driver Version: 352.70        |
|-------------------------------+----------------------+----------------------+
```



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


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

<!--NeedCopy-->
```

Set the correct configuration for the card:

`etc/X11/ctx-nvidia.sh`

HDX 3D Pro multi-monitor redraw issues

If you are seeing redraw issues on screens other than the primary monitor, check that the NVIDIA GRID license is available.

Check Xorg error logs

The log file of Xorg is named similar to **Xorg.{DISPLAY}.log** in the **/var/log/** folder.

Known issues and limitations

For vGPU, the Citrix Hypervisor local console shows the ICA desktop session screen

Workaround: Disable the VM's local VGA console by running the following command:

```
xe vm-param-set uuid=<vm-uuid> platform:vgpu_extra_args="disable_vnc=1"
<!--NeedCopy-->
```


NVIDIA K2 graphics cards do not support YUV444 hardware encoding in pass-through mode

With **Build to Lossless** enabled through the policy setting, a black or gray screen appears when users are launching an app/desktop session with an NVIDIA K2 graphics card. The issue occurs because NVIDIA K2 graphics cards do not support YUV444 hardware encoding in pass-through mode. For more information, see Video Encode and Decode GPU Support Matrix.

Gnome 3 desktop popups slow when logging on

It is a limitation of Gnome 3 desktop session startup.

Some OpenGL/WebGL applications do not render well upon resizing the Citrix Workspace app window

Resizing the window of Citrix Workspace app changes the screen resolution. The NVIDIA proprietary driver changes some internal states and might require applications to respond accordingly. For example, the WebGL library element **lightgl.js** might spawn an error saying that '`Rendering to this texture is not supported (incomplete frame buffer)`'.

Thinwire progressive display

June 11, 2021

Session interactivity can degrade on low bandwidth or high latency connections. For example, on connections with less than 2 Mbps bandwidth or latency of more than 200 ms, scrolling on a webpage can become slow, unresponsive, or choppy. Keyboard and mouse operations can lag behind graphics updates.

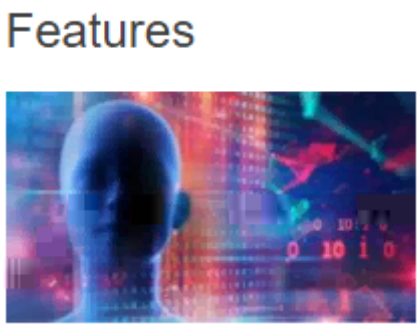
Through version 7.17, you were able to use policy settings to reduce bandwidth consumption by configuring the session to **Low** visual quality, or setting a lower color depth (16-bit or 8-bit graphics). However, you had to know that a user was on a weak connection. HDX Thinwire did not dynamically adjust static image quality based on network conditions.

Starting with Version 7.18, HDX Thinwire, by default, switches to a progressive update mode when available bandwidth falls below 2 Mbps, or network latency exceeds 200 ms. In this mode:

- All static images are heavily compressed.
- Text quality is reduced.

For example, in the following graphic where progressive update mode is active, the letters **F** and **e** have blue artifacts, and the image is heavily compressed. This approach significantly reduces bandwidth consumption, which allows images and text to be received more quickly, and session interactivity improves.

When you stop interacting with the session, the degraded images and text are progressively sharpened to lossless. For example, in the following graphic, the letters no longer contain blue artifacts, and the image appears at source quality.



For images, sharpening uses a random block-like method. For text, individual letters or parts of words are sharpened. The sharpening process occurs over several frames. This approach avoids introducing a delay with a single large sharpening frame.

Transient imagery (video) is still managed with adaptive display or Selective H.264.

How progressive mode is used

By default, progressive mode is on standby for the **Visual quality** policy settings: **High**, **Medium** (default), and **Low**.

Progressive mode is forced off (not used) when:

- **Visual quality** = **Always Lossless** or **Build to Lossless**
- **Preferred color depth for simple graphics** = 8-bit
- **Use video codec for compression** = **For the entire screen** (when full-screen H.264 is desired)

When progressive mode is on standby, by default it is enabled when either of the following conditions occurs:

- Available bandwidth drops below 2 Mbps
- Network latency increases above 200 ms

After a mode switch occurs, a minimum of 10 s is spent in that mode, even if the adverse network conditions are momentary.

Change progressive mode behavior

You can change the progressive mode behavior by running the following command:

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplay" -d "<value>" --force
<!--NeedCopy-->
```

where <value>:

0 = Always off (do not use under any circumstances)

1 = Automatic (toggle based on network conditions, default value)

2 = Always on

When in automatic mode (1), you can run either of the following commands to change the thresholds at which progressive mode is toggled:

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k "HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplayBandwidthThreshold" -d "<value>" --force
<!--NeedCopy-->
```

where <value> is <threshold in Kbps> (default = 2,048)

Example: 4096 = toggle progressive mode on if bandwidth falls below 4 Mbps

```
sudo /opt/Citrix/VDA/bin/ctxreg  create -k "HKEY_LOCAL_MACHINE\SOFTWARE
    \CurrentControlSet\Control\Citrix\Thinwire" -t "REG_DWORD" -v "
    ProgressiveDisplayLatencyThreshold" -d "<value>" --force
<!--NeedCopy-->
```

where <value> is <threshold in ms> (default = 200)

Example: 100 = toggle progressive mode on if network latency drops below 100 ms.

Non-GRID 3D graphics

February 20, 2024

Overview

With this feature enhancement, the Linux VDA supports not only NVIDIA GRID 3D cards but also non-GRID 3D cards.

Installation

To use the non-GRID 3D graphics feature, you must:

- Install XDamage as a prerequisite. Typically, XDamage exists as an extension of XServer.
- Set `CTX_XDL_HDX_3D_PRO` to `Y` when installing the Linux VDA. For information about environment variables, see Step 4: Set up the runtime environment to complete the installation.

Configuration

Xorg configuration files

If your 3D card driver is NVIDIA, the configuration files are installed and set automatically.

Other types of 3D cards

If your 3D card driver is NOT NVIDIA, you must modify the four template configuration files installed under `/etc/X11/`:

- ctx-driver_name-1.conf
- ctx-driver_name-2.conf
- ctx-driver_name-3.conf
- ctx-driver_name-4.conf

Using **ctx-driver_name-1.conf** as an example, do the following to modify the template configuration files:

1. Replace ***driver_name*** with your actual driver name.

   For example, if your driver name is `intel`, you can change the configuration file name to `ctx-intel-1.conf`.

2. Add the video driver information.

   Each template configuration file contains a section named "Device,"which is commented out. This section describes the video driver information. Enable this section before adding your video driver information. To enable this section:

   a) See the 3D card guide provided by the manufacturer for configuration information. A native configuration file can be generated. Verify that your 3D card can work in a local environment with the native configuration file when you are not using a Linux VDA ICA session.

   b) Copy the "Device"section of the native configuration file to **ctx-driver_name-1.conf**.

3. Run the following command to set the registry key so that the Linux VDA can recognize the configuration file name set in Step 1.

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
     CurrentControlSet\Control\Citrix\XDamage" -t "REG_SZ" -v "
     DriverName" -d "intel" --force
2 <!--NeedCopy-->
```

Enable the non-GRID 3D graphics feature

The non-GRID 3D graphics feature is disabled by default. You can run the following command to enable it by setting XDamageEnabled to 1.

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\System\
     CurrentControlSet\Control\Citrix\XDamage" -t "REG_DWORD" -v "
     XDamageEnabled" -d "0x00000001" --force
2 <!--NeedCopy-->
```

Troubleshooting

No or garbled graphic output

If you can run 3D applications locally and all configurations are correct, missing or garbled graphic output is the result of a bug. Use /opt/Citrix/VDA/bin/setlog and set GFX_X11 to verbose to collect the trace information for debugging.

Hardware encoding does not work

This feature supports only software encoding.

Configure policies

June 11, 2021

Installation

Follow the installation articles to prepare the Linux VDA.

Dependencies

Ensure that you install these dependencies before installing the Linux VDA package.

RHEL/CentOS:

```
sudo yum -y install openldap

sudo yum -y install libxml2

sudo yum -y install cyrus-sasl

sudo yum -y install cyrus-sasl-gssapi
<!--NeedCopy-->
```

SLES/SELD:

```
sudo zypper install openldap2

sudo zypper install libxml2

sudo zypper install cyrus-sasl

sudo zypper install cyrus-sasl-gssapi
<!--NeedCopy-->
```

Ubuntu:

```
sudo apt-get install -y libldap-2.4-2

sudo apt-get install -y libsasl2-2

sudo apt-get install -y libsasl2-modules-gssapi-mit
<!--NeedCopy-->
```


Configuration

Policy settings in Citrix Studio

To set policies in Citrix Studio, do the following:

1. Open **Citrix Studio**.
2. Select the **Policies** panel.
3. Click **Create Policy**.
4. Set the policy according to the Policy support list.

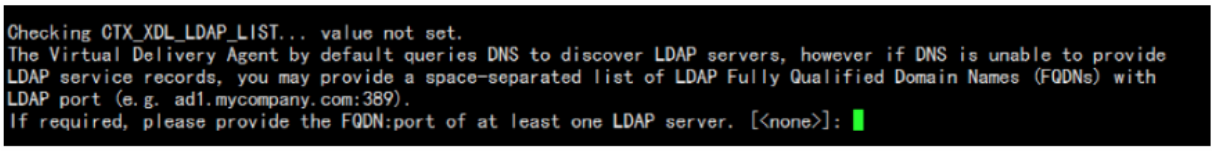
LDAP server setting on the VDA

The LDAP server setting on Linux VDA is optional for single domain environments but mandatory for multiple domain and multiple forest environments. The setting is necessary for the policy service to perform an LDAP search in these environments.

After installing the Linux VDA package, run the command:

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

Type all the LDAP servers in the suggested format: space-separated list of LDAP Fully Qualified Domain Names (FQDNs) with the LDAP port (for example, ad1.mycompany.com:389 ad2.mycomany.com:389).



You can also run the **ctxreg** command to write this setting to the registry directly:

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
   VirtualDesktopAgent" -t "REG_SZ" -v "ListOfLDAPServers" -d "ad1.
   mycompany.com:389 ad2.mycomany.com:389" --force
<!--NeedCopy-->
```

Policy support list

June 11, 2021

Linux VDA policy support list

Studio Policy	Key Name	Type	Module	Default Value
Use local time of client	UseLocalTimeOfClient	User	ICA\Time Zone Control	Use server time zone
ICA round trip calculation	IcaRoundTripCheckEnabled	Computer	ICA\End User Monitoring	Enabled (1)
ICA round trip calculation interval	IcaRoundTripCheckPeriod	Computer	ICA\End User Monitoring	15
ICA round trip calculations for idle connections	IcaRoundTripCheckWhenIdle	Computer	ICA\End User Monitoring	Disabled (0)
Overall session bandwidth limit	LimitOverallBw	User	ICA\Bandwidth	0
Audio redirection bandwidth limit	LimitAudioBw	User	ICA\Bandwidth	0
Audio redirection bandwidth limit percent	LimitAudioBwPercent	User	ICA\Bandwidth	0
Client USB device redirection bandwidth limit	LimitUSBBw	User	ICA\Bandwidth	0
Client USB device redirection bandwidth percent	LimitUSBBwPercent	User	ICA\Bandwidth	0
Clipboard redirection bandwidth limit	LimitClipbdBW	User	ICA\Bandwidth	0
Clipboard redirection bandwidth limit percent	LimitClipbdBWPercent	User	ICA\Bandwidth	0
File redirection bandwidth limit	LimitCdmBw	User	ICA\Bandwidth	0
File redirection bandwidth limit percent	LimitCdmBwPercent	User	ICA\Bandwidth	0

Studio Policy	Key Name	Type	Module	Default Value
Printer redirection bandwidth limit	LimitPrinterBw	User	ICA\Bandwidth	0
Printer redirection bandwidth limit percent	LimitPrinterBwPercent	User	ICA\Bandwidth	0
WebSockets connections	AcceptWebSocketsConnections	Computer	ICA\WebSockets	Prohibited
WebSockets port number	WebSocketsPort	Computer	ICA\WebSockets	8008
WebSockets trusted origin server list	WSTrustedOriginServerList	Computer	ICA\WebSockets	*
ICA keep alives	SendICAKeepAlives	Computer	ICA keep alive	Do not send ICA keep alive messages (0)
ICA keep alive timeout	ICAKeepAliveTimeout	Computer	ICA keep alive	60 seconds
ICA listener port number	IcaListenerPortNumber	Computer	ICA	1494
HDX adaptive transport	HDXoverUDP	Computer	ICA	Preferred(2)
Session reliability connections	AcceptSessionReliabilityConnections	Computer	ICA\Session Reliability	Allowed(1)
Reconnection UI transparency level	ReconnectionUiTransparencyLevel	Computer	ICA\Auto Client Reconnect	80%
Session reliability port number	SessionReliabilityPort	Computer	ICA\Session Reliability	2598
Session reliability timeout	SessionReliabilityTimeout	Computer	ICA\Session Reliability	180 s
Auto Client Reconnect	AllowAutoClientReconnect	User	ICA\Auto Client Reconnect	Allowed (1)
Client audio redirection	AllowAudioRedirection	User	Audio	Allowed (1)
Client printer redirection	AllowPrinterRedir	User	Printing	Allowed (1)

Studio Policy	Key Name	Type	Module	Default Value
Auto-create PDF Universal Printer	AutoCreatePDFPrinter	User	Printing	Disabled (0)
Printer driver mapping and compatibility	DriverMappingList	User	Printing	`"Microsoft XPS Document Writer *, Deny;Send to Microsoft OneNote *, Deny"`
Client clipboard redirection	AllowClipboardRedir	User	Clipboard	Allowed (1)
Client USB device redirection	AllowUSBRedir	User	USB	Prohibited (0)
Client USB device redirection rules	USBDeviceRules	User	USB	“\0”
Moving image compression	MovingImageCompressionConfiguration	User	Thinwire	Enabled (1)
Extra color compression	ExtraColorCompression	User	Thinwire	Disabled (0)
Target minimum frame rate	TargetedMinimumFramesPerSecond	User	Thinwire	10 fps
Target frame rate	FramesPerSecond	User	Thinwire	30 fps
Visual quality	VisualQuality	User	Thinwire	Medium (3)
Use video codec for compression	VideoCodec	User	Thinwire	Use when preferred (3)
Use hardware encoding for video codec	UseHardwareEncodingForVideoCodec	User	Thinwire	Enabled (1)
Allow visually lossless compression	AllowVisuallyLosslessCompression	User	Thinwire	Disabled (0)
Optimize for 3D graphics workload	OptimizeFor3dWorkload	User	Thinwire	Disabled (0)
Preferred color depth for simple graphics	PreferredColorDepth	User	Thinwire	24 bits per pixel(1)

Studio Policy	Key Name	Type	Module	Default Value
Audio quality	SoundQuality	User	Audio	High –high definition audio (2)
Client microphone redirection	AllowMicrophoneRedir	User	Audio	Allowed (1)
Maximum number of sessions	MaximumNumberOfSessions	Computer	Load Management	250
Concurrent logons tolerance	ConcurrentLogonsTolerance	Computer	Load Management	2
Enable auto update of Controllers	EnableAutoUpdateOfControllers	Computer	Virtual Delivery Agent Settings	Allowed (1)
Clipboard selection update mode	ClipboardSelectionUpdateMode	User	Clipboard	3
Primary selection update mode	PrimarySelectionUpdateMode	User	Clipboard	3
Max speex quality	MaxSpeexQuality	User	Audio	5
Auto connect client drives	AutoConnectDrives	User	File redirection/CDM	Enabled (1)
Client optical drives	AllowCdromDrives	User	File redirection/CDM	Allowed (1)
Client fixed drives	AllowFixedDrives	User	File redirection/CDM	Allowed (1)
Client floppy drives	AllowFloppyDrives	User	File redirection/CDM	Allowed (1)
Client network drives	AllowNetworkDrives	User	File redirection/CDM	Allowed (1)
Client drive redirection	AllowDriveRedir	User	File redirection/CDM	Allowed (1)
Read-only client drive access	ReadOnlyMappedDrive	User	File redirection/CDM	Disabled (0)
Automatic keyboard display	AllowAutoKeyboardPopup	User	MRVC	Disabled (0)

Studio Policy	Key Name	Type	Module	Default Value
Allow file transfer between desktop and client	AllowFileTransfer	User	File Transfer	Allowed
Download file from desktop	AllowFileDownload	User	File Transfer	Allowed
Upload file to desktop	AllowFileUpload	User	File Transfer	Allowed
Session idle timer	EnableSessionIdleTimer	User	Session Timers	Enabled (1)
Session idle timer interval	SessionIdleTimerInterval	User	Session Timers	1440 minutes
Disconnected session timer	EnableSessionDisconnectTimer	User	Session Timers	Disabled (0)
Disconnected session timer interval	SessionDisconnectTimerPeriod	User	Session Timers	1440 minutes

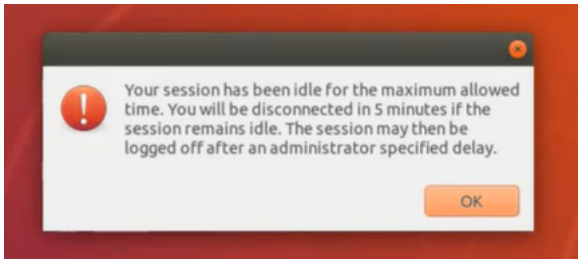
> **Note:**
>
> Only the Windows Virtual Delivery Agent (VDA) supports audio over User Datagram Protocol (UDP). The Linux VDA does not. For more information, see Audio over User Datagram Protocol (UDP) Real-time Transport.

You can use the following Citrix policy settings to configure session connection timers in Citrix Studio:

- **Session idle timer**: Determines whether to enforce a time limit for idle sessions.
- **Session idle timer interval**: Sets a time limit for idle sessions. If Session idle timer is **Enabled** and an active session has not received user input during the set time, the session disconnects.
- **Disconnected session timer**: Determines whether to enforce a time limit for disconnected sessions.
- **Disconnected session timer interval**: Sets an interval before a disconnected session is logged off.

When you update any of the policy settings, ensure that they are consistent across your deployment.

A warning message appears when your time limit for idle sessions expires. See the following screen capture for an example. Pressing **OK** closes the warning message but cannot keep your session active. To keep your session active, provide user input to reset the idle timer.



The following policies can be configured in Citrix Studio Version 7.12 and later.

- MaxSpeexQuality

  Value (integer): [0–10]

  Default value: 5

  Details:

  Audio redirection encodes audio data with the Speex codec when audio quality is medium or low (see the policy Audio quality). Speex is a lossy codec, which means that it achieves compression at the expense of fidelity of the input speech signal. Unlike some other speech codecs, it is possible to control the tradeoff made between quality and bit rate. The Speex encoding process is controlled most of the time by a quality parameter that ranges from 0 to 10. The higher the quality is, the higher the bit rate.

  The max Speex quality chooses the best Speex quality to encode audio data according to audio quality and bandwidth limit (see the policy Audio redirection bandwidth limit). If the audio quality is medium, the encoder is in wide band mode, which means a higher sampling rate. If the audio quality is low, the encoder is in narrow band mode, which means a lower sampling rate. The same Speex quality has different bit rates in different modes. The best Speex quality is when the largest value meets the following conditions:

  - It is equal to or less than the max Speex quality.
  - Its bit rate is equal to or less than the bandwidth limit.

  Related Settings: Audio quality, Audio redirection bandwidth limit

- PrimarySelectionUpdateMode

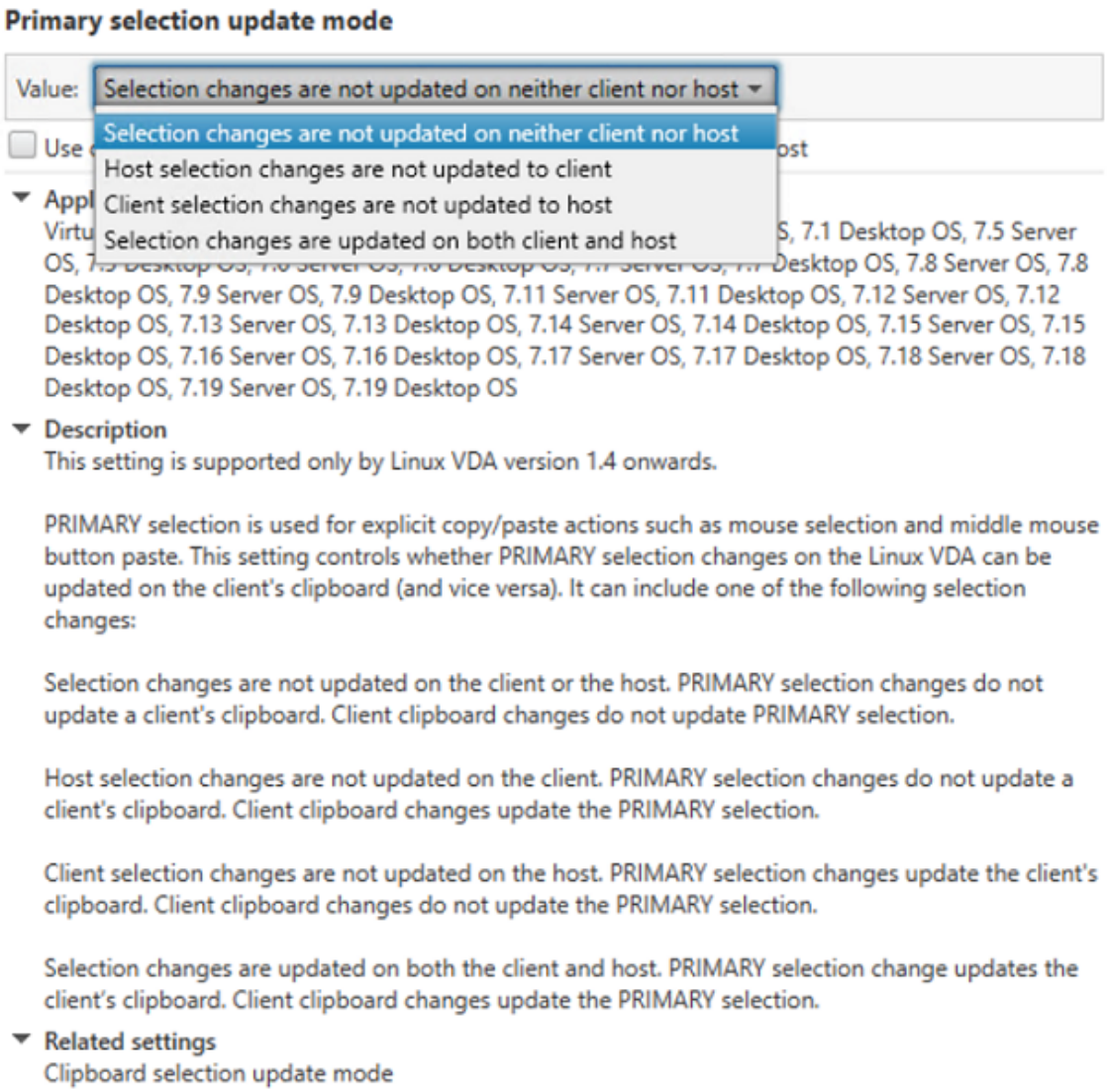
  Value (enum): [0, 1, 2, 3]

  Default value: 3

  Details:

  Primary selection is used when you select data and paste it by pressing the middle mouse button.

  This policy controls whether primary selection changes on the Linux VDA and client can update the clipboard on each other. There are four value options:



- **Selection changes are not updated on neither client nor host**
  Primary selection changes on the Linux VDA do not update the clipboard on the client. Primary selection changes on the client do not update the clipboard on the Linux VDA.
- **Host selection changes are not updated to client**
  Primary selection changes on the Linux VDA do not update the clipboard on the client. Primary selection changes on the client update the clipboard on the Linux VDA.
- **Client selection changes are not updated to host**
  Primary selection changes on the Linux VDA update the clipboard on the client. Primary selection changes on the client do not update the clipboard on the Linux VDA.
- **Selection changes are updated on both client and host**
  Primary selection changes on the Linux VDA update the clipboard on the client. Primary selection changes on the client update the clipboard on the Linux VDA. This option is the

default value.

Related Setting: Clipboard selection update mode

- ClipboardSelectionUpdateMode

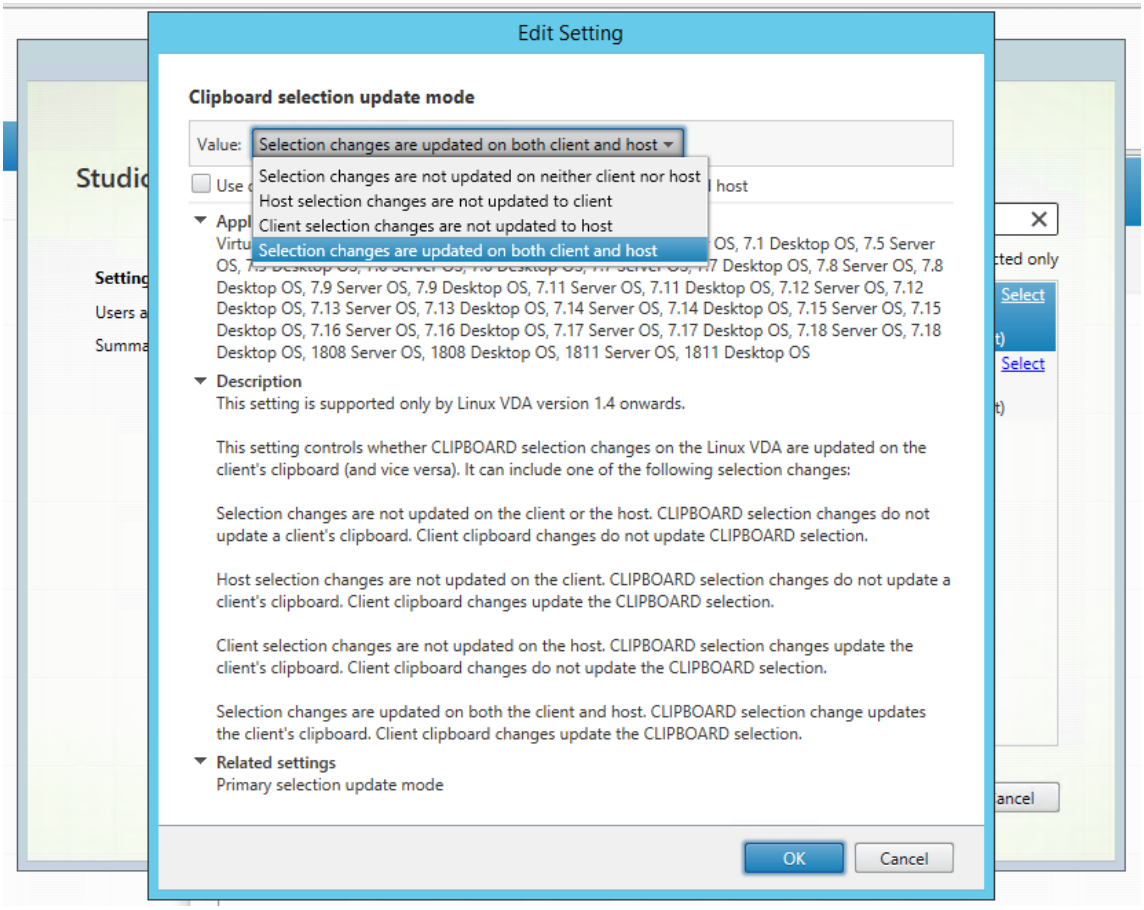
  Value (enum): [0, 1, 2, 3]

  Default value: 3

  Details:

  Clipboard selection is used when you select some data and explicitly request it to be "copied" to the clipboard, such as by selecting "Copy"from the shortcut menu. Clipboard selection is primarily used in connection with Microsoft Windows clipboard operations while primary selection is unique to Linux.

  This policy controls whether clipboard selection changes on the Linux VDA and client can update the clipboard on each other. There are four value options:



  - **Selection changes are not updated on neither client nor host**
    Clipboard selection changes on the Linux VDA do not update the clipboard on the client. Clipboard selection changes on the client do not update the clipboard on the Linux VDA.

- **Host selection changes are not updated to client**
  Clipboard selection changes on the Linux VDA do not update the clipboard on the client. Clipboard selection changes on the client update the clipboard on the Linux VDA.
- **Client selection changes are not updated to host**
  Clipboard selection changes on the Linux VDA update the clipboard on the client. Clipboard selection changes on the client do not update the clipboard on the Linux VDA.
- **Selection changes are updated on both client and host**
  Clipboard selection changes on the Linux VDA update the clipboard on the client. Clipboard selection changes on the client update the clipboard on the Linux VDA. This option is the default value.

Related Setting: Primary selection update mode

> **Note:**
>
> The Linux VDA supports both clipboard selection and primary selection. To control the copy and paste behaviors between the Linux VDA and the client, we recommend that you set both clipboard selection update mode and primary selection update mode to the same value.

Configure IPv6

June 11, 2021

The Linux VDA supports IPv6 to align with Citrix Virtual Apps and Desktops. When using this feature, consider the following:

- For dual stack environments, IPv4 is used unless IPv6 is explicitly enabled.
- If IPv6 is enabled in an IPv4 environment, the Linux VDA fails to function.

> **Important:**
>
> - The whole network environment must be IPv6, not only for the Linux VDA.
> - Centrify does not support pure IPv6.

No special setup tasks are required for IPv6 when you install the Linux VDA.

Configure IPv6 for the Linux VDA

Before changing the configuration for the Linux VDA, ensure that your Linux virtual machine has previously worked in an IPv6 network. There are two registry keys related to IPv6 configuration:


```
1 “HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD”
    -v “OnlyUseIPv6ControllerRegistration”
2 “HKLM\Software\Policies\Citrix\VirtualDesktopAgent” -t “REG_DWORD”
    -v “ControllerRegistrationIPv6Netmask”
3 <!--NeedCopy-->
```

OnlyUseIPv6ControllerRegistration must be set to 1 to enable IPv6 on the Linux VDA:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
    Citrix\VirtualDesktopAgent" -t "REG_DWORD" -v "
    OnlyUseIPv6ControllerRegistration" -d "0x00000001" --force
2 <!--NeedCopy-->
```

If the Linux VDA has more than one network interfaces, **ControllerRegistrationIPv6Netmask** can be used to specify which one is used for the Linux VDA registration:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Policies\
    Citrix\VirtualDesktopAgent" -t "REG_SZ" -v "
    ControllerRegistrationIPv6Netmask " -d "{
2  IPv6 netmask }
3  " --force
4 <!--NeedCopy-->
```

Replace **{IPv6 netmask}** with the real netmask (for example, 2000::/64).

For more information about IPv6 deployment in Citrix Virtual Apps and Desktops, see IPv4/IPv6 support.

Troubleshooting

Check the basic IPv6 network environment and use ping6 to check whether AD and Delivery Controller are reachable.

Configure Citrix Customer Experience Improvement Program (CEIP)

June 11, 2021

When you participate in the CEIP, anonymous statistics and usage information are sent to Citrix to help improve the quality and performance of Citrix products. In addition, a copy of the anonymous data is sent to Google Analytics (GA) for fast and efficient analysis.

Registry settings

By default, you automatically participate in the CEIP when you install the Linux VDA. The first upload of data occurs approximately seven days after you install the Linux VDA. You can change this default setting in the registry.

- **CEIPSwitch**

Registry setting that enables or disables the CEIP (default = 0):

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

Name: CEIPSwitch

Value: 1 = disabled, 0 = enabled

When unspecified, the CEIP is enabled.

You can run the following command on a client to disable the CEIP:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\
     Citrix\CEIP" -v "CEIPSwitch" -d "1"
2 <!--NeedCopy-->
```

- **GASwitch**

Registry setting that enables or disables GA (default = 0):

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

Name: GASwitch

Value: 1 = disabled, 0 = enabled

When unspecified, GA is enabled.

You can run the following command on a client to disable GA:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\
     Citrix\CEIP" -v "GASwitch" -d "1"
2 <!--NeedCopy-->
```

- **DataPersistPath**

Registry setting that controls the data persisting path (default = /var/xdl/ceip):

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CEIP

Name: DataPersistPath

Value: String

You can run the following command to set this path:


```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SOFTWARE\
   Citrix\CEIP" -v "DataPersistPath" -d "your_path"
<!--NeedCopy-->
```

If the configured path does not exist or cannot be accessed, data is saved in the default path.

CEIP data collected from the Linux VDA

The following table gives an example of the types of anonymous information collected. The data does not contain any details that identify you as a customer.

Data Point	Key Name	Description
Machine GUID	machine_guid	Identifying the machine where the data originates
AD solution	ad_solution	Text string denoting the machine's domain joining method
Linux kernel version	kernel_version	Text string denoting the machine's kernel version
LVDA version	vda_version	Text string denoting the installed version of the Linux VDA.
LVDA update or fresh install	update_or_fresh_install	Text string denoting the current Linux VDA package is being freshly installed or updated
LVDA installed method	install_method	Text string denoting that the current Linux VDA package is installed by using MCS, PVS, easy install, or manual installation.
HDX 3D pro enabled or not	hdx_3d_pro	Text string denoting whether HDX 3D Pro is enabled on the machine
VDI mode enabled or not	vdi_mode	Text string denoting whether VDI mode is enabled
System Locale	system_locale	Text string denoting the locale of this machine

Data Point	Key Name	Description
LVDA key services last restart time	ctxhdx ctxvda	The last restart time of the `ctxhdx` and `ctxvda` services, in the format of dd-hh:mm:ss, for example, 10-17:22:19
GPU type	gpu_type	Denoting the GPU type of the machine
CPU cores	cpu_cores	Integer denoting the number of CPU cores of the machine
CPU frequency	cpu_frequency	Float denoting the CPU frequency in MHz
Physical memory size	memory_size	Integer denoting the physical memory size in KB
Launched session number	session_launch	Integer denoting the number of sessions launched (logged on or reconnected) on the machine at the time we collect this data point
Linux OS name and version	os_name_version	Text string denoting the Linux OS name and version of the machine
Session key	session_key	Identifying the session where the data originates
Resource type	resource_type	Text string denoting the resource type of the launched session: desktop or `<appname>`
Active session time	active_session_time	Used to save the session's active times. One session can have multiple active times because the session can disconnect/reconnect
Session duration time	session_duration_time	Used to save the session's duration from logon to logoff
Receiver client type	receiver_type	Integer denoting the type of Citrix Workspace app used to launch the session

Data Point	Key Name	Description
Receiver client version	receiver_version	Text string denoting the version of Citrix Workspace app used to launch the session
Printing count	printing_count	Integer denoting the number of times the session uses the printing function
USB redirection count	usb_redirecting_count	Integer denoting the number of times the session uses a USB device
Gfx Provider type	gfx_provider_type	Text string denoting the graphics provider type of the session
Shadowing count	shadow_count	Integer denoting the number of times the session has been shadowed
User selected Language	ctxism_select	Composed long string that contains all languages that users have selected
Smartcard redirecting count	scard_redirecting_count	Integer denoting the number of times smart card redirection is used for session logons and user authentication for in-session apps

Configure USB redirection

June 11, 2021

USB devices are shared between Citrix Workspace app and the Linux VDA desktop. When a USB device is redirected to the desktop, the user can use the USB device as if it were locally connected.

> **Tip:**
>
> We recommend using USB redirection when the network latency is lower than 100 milliseconds. Do not use USB redirection when the network latency is higher than 200 milliseconds.

USB redirection includes three main areas of functionality:

- Open-source project implementation (VHCI)
- VHCI service
- USB service

Open-source VHCI:

This portion of the USB redirection feature develops a general USB device sharing system over an IP network. It consists of a Linux kernel driver and some user mode libraries that allow you to communicate with the kernel driver to get all the USB data. In the Linux VDA implementation, Citrix reuses the kernel driver of VHCI. However, all the USB data transfers between the Linux VDA and Citrix Workspace app are encapsulated in the Citrix ICA protocol package.

VHCI service:

The VHCI service is an open-source service provided by Citrix to communicate with the VHCI kernel module. This service works as a gateway between VHCI and the Citrix USB service.

USB service:

The USB service represents a Citrix module that manages all the virtualization and data transfers on the USB device.

How USB redirection works

Typically, if a USB device is redirected successfully to the Linux VDA, one or more device nodes are created in the system /dev path. Sometimes, however, the redirected device is not usable for an active Linux VDA session. USB devices rely on drivers to function properly and some devices require special drivers. If drivers are not provided, the redirected USB devices are inaccessible to the active Linux VDA session. To ensure USB device connectivity, install the drivers and configure the system properly.

The Linux VDA supports a list of USB devices that are successfully redirected to and from the client. In addition, the device is properly mounted, especially the USB disk, allowing the user to access the disk without any additional configuration.

Supported USB devices

The following devices have been verified to support this version of the Linux VDA. Other devices might be freely used, with unexpected results:

> **Note:**
>
> The Linux VDA supports only USB 2.0 protocols.

USB mass storage device	VID:PID	File system
Netac Technology Co., Ltd	0dd8:173c	FAT32
Kingston Datatraveler 101 II	0951:1625	FAT32
Kingston Datatraveler GT101 G2	1567:8902	FAT32
SanDisk SDCZ80 flash drive	0781:5580	FAT32
WD HDD	1058:10B8	FAT32

USB 3D mouse	VID:PID
3DConnexion SpaceMouse Pro	046d: c62b

USB scanner	VID:PID
Epson Perfection V330 photo	04B8: 0142

Configure USB redirection

A Citrix policy controls whether USB device redirection is enabled or disabled. In addition, the type of device can also be specified using a Delivery Controller policy. When configuring USB redirection for the Linux VDA, configure the following policy and rules:

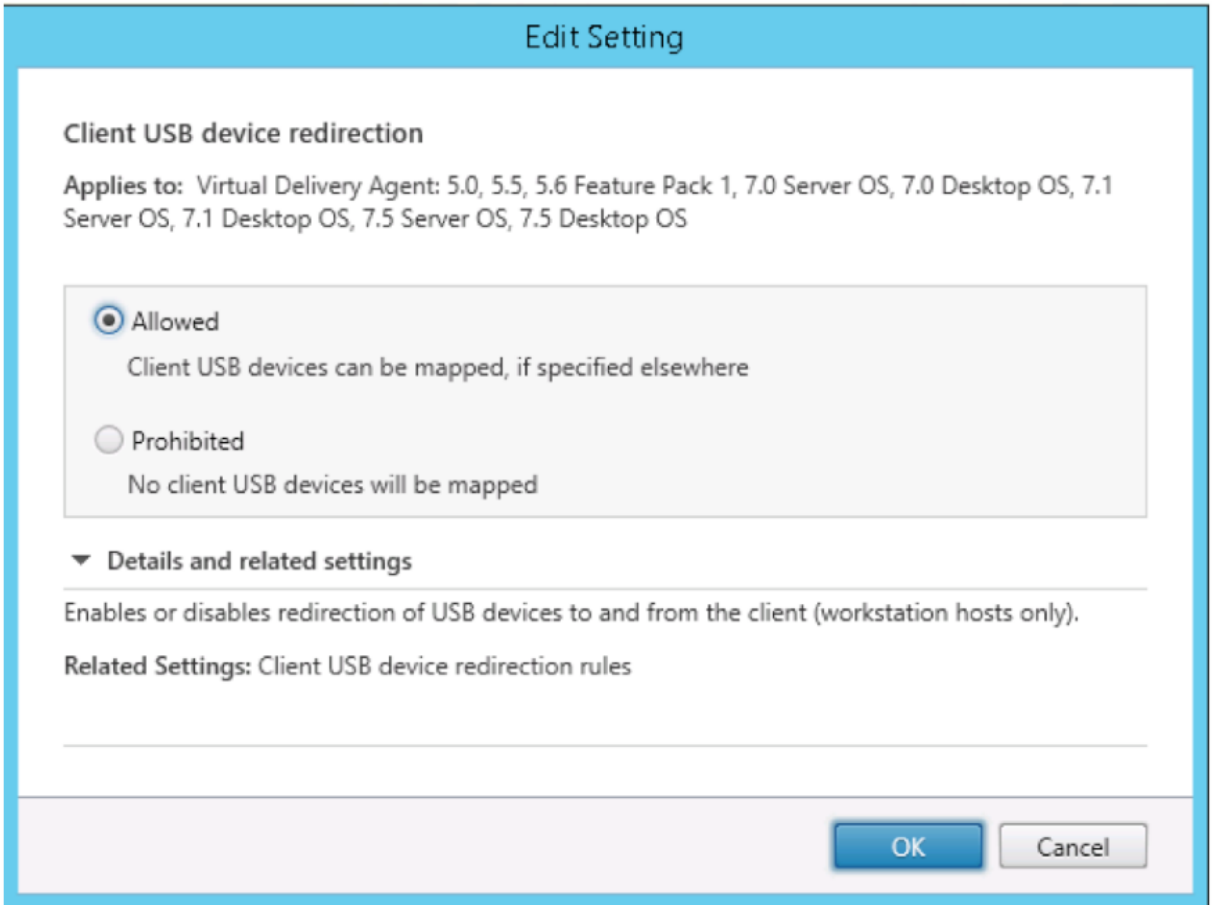
- Client USB device redirection policy
- Client USB device redirection rules

Enable USB redirection policy

In Citrix Studio, enable (or disable) USB device redirection to and from the client (for workstation hosts only).

In the **Edit Setting** dialog:

1. Select **Allowed**.
2. Click **OK**.

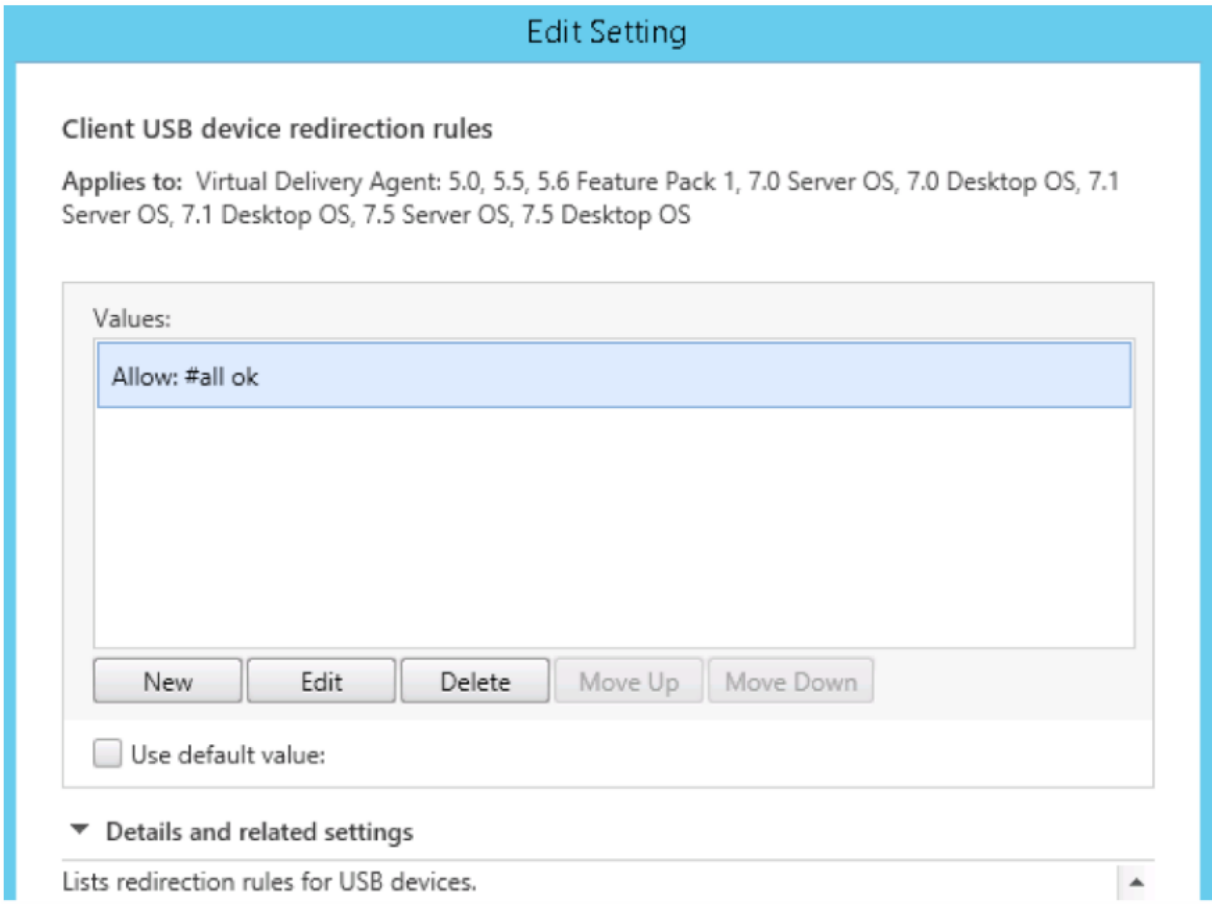


Set USB redirection rules

After enabling the USB redirection policy, set the redirection rules using Citrix Studio by specifying which devices are allowed (or denied) on the Linux VDA.

In the Client USB device redirection rules dialog:

1. Click **New** to add a redirection rule, or click **Edit** to review an existing rule.
2. After creating (or editing) a rule, click **OK**.



For more information about configuring generic USB redirection, see Citrix Generic USB Redirection Configuration Guide.

Build the VHCI kernel module

USB redirection depends on the VHCI kernel modules (`usb-vhci-hcd.ko` and `usb-vhci-iocif.ko`). These modules are part of the Linux VDA distribution (as part of the RPM package). They are compiled based on the official Linux distribution kernels and are noted in the following table:

Supported Linux distribution	Kernel version
RHEL 8.3	4.18.0-240.15.1
RHEL 8.2	4.18.0-240.15.1
RHEL 8.1	4.18.0-240.15.1
RHEL 7.9	3.10.0-1160.21.1
RHEL 7.8	3.10.0-1160.21.1

Supported Linux distribution	Kernel version
SUSE 12.5	4.12.14-122.63.1
Ubuntu 20.04	5.4.0-70
Ubuntu 18.04	4.15.0-140
Ubuntu 16.04	4.4.0-206
Debian 10	4.19.0-16

> **Important:**
>
> If the kernel of your machine is not compatible with the driver built for the Linux VDA, the USB service might fail to start. In this case, you can use the USB redirection feature only if you build your own VHCI kernel modules.

Verify whether your kernel is consistent with the modules built by Citrix

On the command line, run the following command to verify whether the kernel is consistent:

```
insmod /opt/Citrix/VDA/lib64/usb-vhci-hcd.ko
<!--NeedCopy-->
```

If the command runs successfully, the kernel module has loaded successfully and the version is consistent with the one installed by Citrix.

If the command runs with errors, the kernel is inconsistent with the Citrix module and must be rebuilt.

Rebuild the VHCI kernel module

If your kernel module is inconsistent with the Citrix version, do the following:

1. Download the LVDA source code from the Citrix download site. Select the file contained in the section "**Linux Virtual Delivery Agent (sources)**."
2. Unzip the **citrix-linux-vda-sources.zip** file. Navigate to **linux-vda-souces/vhci-hcd-1.15.tar.bz2** and unzip the VHCI source files by using **tar xvf vhci-hcd-1.15.tar.bz2**.
3. Build the kernel module based on the header files and the **Module.symvers** file. Use the following steps to install the kernel header files and create **Module.symvers** based on the appropriate Linux distribution:

   RHEL/CentOS:


```
yum install kernel-devel
<!--NeedCopy-->
```

SUSE 12:

```
zypper install kernel-devel

zypper install kernel-source
<!--NeedCopy-->
```

Ubuntu:

```
apt-get install linux-headers
<!--NeedCopy-->
```

> **Tip:**
>
> If the installation is successful, there is a kernel folder resembling:
>
> /usr/src/kernels/3.10.0-327.10.1.el7.x86_64

4. In the /usr/src/kernels/3.10.0-327.10.1.el7.x86_64 folder, verify that the **Module.symvers** file is present. If this file is not in the folder, build the kernel (by running the following commands in sequence: `make oldconfig`; `make prepare`; `make modules`; `make`) to get this file or copy it from **/usr/src/kernels/3.10.0-327.10.1.el7.x86_64-obj/x86_64/defaults/module.***
5. Run the following commands to install the development tools.

RHEL 8, CentOS 8:

```
yum groupinstall 'Development Tools'

yum install elfutils-libelf-devel
<!--NeedCopy-->
```

RHEL 7, CentOS 7:

```
yum groupinstall 'Development Tools'
<!--NeedCopy-->
```

Ubuntu 20.04, Ubuntu 18.04, Debian 10:

```
apt install build-essential flex bison libelf-dev
<!--NeedCopy-->
```

Ubuntu 16.04:

```
apt install build-essential flex bison

<!--NeedCopy-->
```


6. In the `vhci-hcd-1.15/Makefile` file, change the Makefile of VCHI and set KDIR to the kernel directory:

```
#KDIR = $(BUILD_PREFIX)/lib/modules/$(KVERSION)/build

KDIR = /usr/src/kernels/3.10.0-327.10.1.el7.x86_64
<!--NeedCopy-->
```

7. In the `vhci-hcd-1.15/` folder, run `make` to build the VHCI kernel.

> **Note:**
>
> If the build was successful, `usb-vhci-hcd.ko` and `usb-vhci-iocifc.ko` are created in the `vhci-hcd-1.15/` folder.

8. Replace the kernel module with the newly built one: **cp -f usb-vhci-*.ko /opt/Citrix/VDA/lib64/**
9. Restart the USB service:

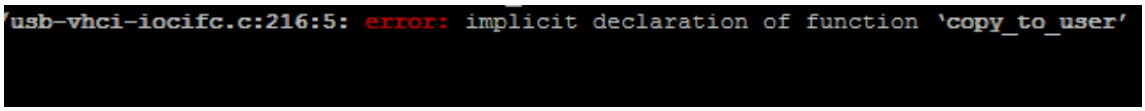
```
service ctxusbsd restart
<!--NeedCopy-->
```

10. Log off and back on to the session again. Check whether USB redirection is functioning.

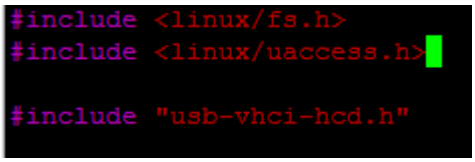
Troubleshoot kernel building issues

The following errors might occur when you build the VHCI module with specific kernels:

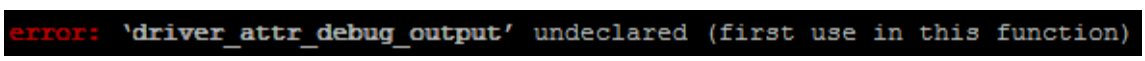
- The `implicit declaration of function ‘copy\\_to\\_user’` error might occur, see the following screen capture:



  The error occurs due to header file changes in the kernels. As a workaround, add the **#include <linux/uaccess.h>** line to the `vhci-hcd-1.15/usb-vhci-iocifc.c` file.



- The `‘driver\\_attr\\_debug_output’ undeclared` error might occur, see the following screen capture:



The error occurs when symbols are missing on the kernel. As a workaround, disable the macro definition for DEBUG in the `vhci-hcd-1.15/usb-vhci-iocifc.c` and `vhci-hcd-1.15/usb-vhci-hcd.c` files.

```
22
23 //#define DEBUG
24
25 #include <linux/module.h>
```

- The `‘make[3]: *** No rule to make target 'arch/x86/tools/relocs_32.c', needed by 'arch/x86/tools/relocs_32.o'. Stop.` error might occur, see the following screen capture:

```
scripts/kconfig/conf  --syncconfig Kconfig
make[3]: *** No rule to make target 'arch/x86/tools/relocs_32.c', needed by 'arch/x86/tools/relocs_32.o'.  Stop.
arch/x86/Makefile:232: recipe for target 'archscripts' failed
make[2]: *** [archscripts] Error 2
make[2]: Leaving directory '/usr/src/linux-headers-5.4.0-1031-azure'
Makefile:102: recipe for target 'testcc' failed
make[1]: *** [testcc] Error 2
make[1]: Leaving directory '/home/administrator1/linuxvda-vhci'
Makefile:97: recipe for target 'conf/usb-vhci.config.h' failed
make: *** [conf/usb-vhci.config.h] Error 2
```

As a workaround, replace `SUBDIRS=$(PWD)` with `M=$(shell pwd)` by using the following commands under the `vhci-hcd-1.15/` path:

```
1  sed -i 's/SUBDIRS=$(PWD)/M=$(shell pwd)/g' Makefile
2
3  sed -i 's/SUBDIRS=$(PWD)/M=$(shell pwd)/g' test/Makefile
4  <!--NeedCopy-->
```

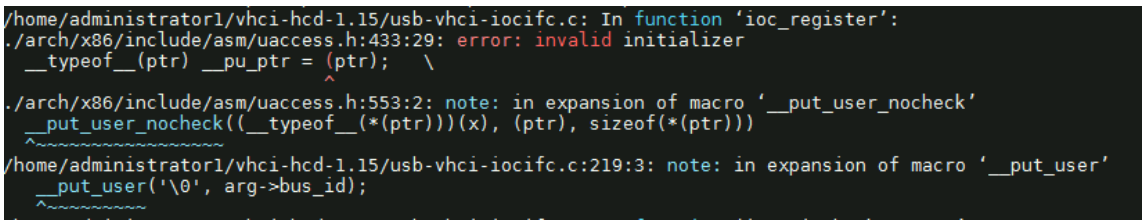
- The `./include/uapi/linux/stat.h:30:17: error: expected ‘)’ before numeric constant #define S_IRUSR 00400` error might occur, see the following screen capture:

```
In file included from ./include/linux/stat.h:7:0,
                 from ./include/linux/module.h:10,
                 from /home/administrator1/vhci-hcd-1.15/usb-vhci-hcd.c:24:
./include/uapi/linux/stat.h:30:17: error: expected ‘)’ before numeric constant
 #define S_IRUSR 00400
                 ^
/home/administrator1/vhci-hcd-1.15/usb-vhci-hcd.c:1312:34: note: in expansion of macro ‘S_IRUSR’
 static DRIVER_ATTR(debug_output, S_IRUSR | S_IWUSR, show_debug_output, store_debug_output);
                                  ^~~~~~~
```

Run the following commands to work around the issue:

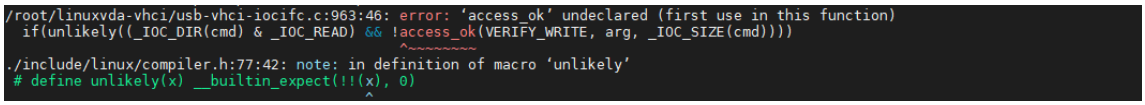
```
1  sed -i 's/show_debug_output/debug_output_show/g' usb-vhci-iocifc.
       c usb-vhci-hcd.c
2
3  sed -i 's/store_debug_output/debug_output_store/g' usb-vhci-
       iocifc.c usb-vhci-hcd.c
4
5  sed -i 's/static DRIVER_ATTR(debug_output, S_IRUSR | S_IWUSR,
       debug_output_show, debug_output_store);/static DRIVER_ATTR_RW(
       debug_output);/g' usb-vhci-iocifc.c usb-vhci-hcd.c
6  <!--NeedCopy-->
```


- The `./arch/x86/include/asm/uaccess.h:433:29: error: invalid initializer __typeof__(ptr)__pu_ptr = (ptr); \` error might occur, see the following screen capture:



  As a workaround, change the 219 line of the `usb-vhci-iocifc.c` file from `__put_user('\0', arg->bus_id);` to `__put_user('\0', arg->bus_id + 0);`.

- The `error: 'access_ok' undeclared (first use in this function) if(unlikely((_IOC_DIR(cmd)& _IOC_READ)&& !access_ok(VERIFY_WRITE, arg, _IOC_SIZE(cmd))))` error might occur, see the following screen capture:



  Run the following commands to work around the issue:

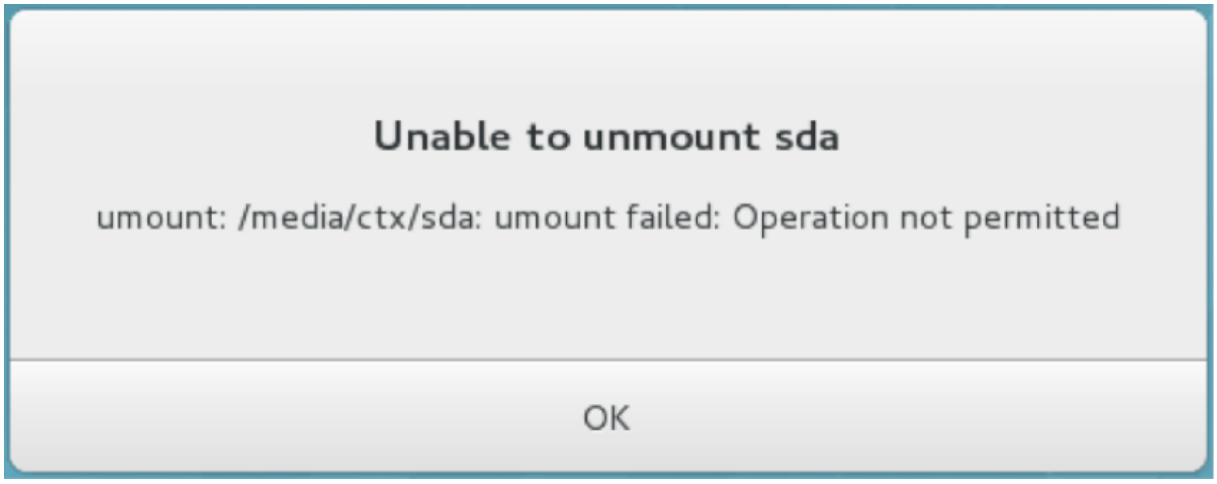
```
sed -i 's/VERIFY_READ, //g' usb-vhci-iocifc.c
sed -i 's/VERIFY_WRITE, //g' usb-vhci-iocifc.c
<!--NeedCopy-->
```

Troubleshoot USB redirection issues

Use the information in this section to troubleshoot various issues that you might encounter when using the Linux VDA.

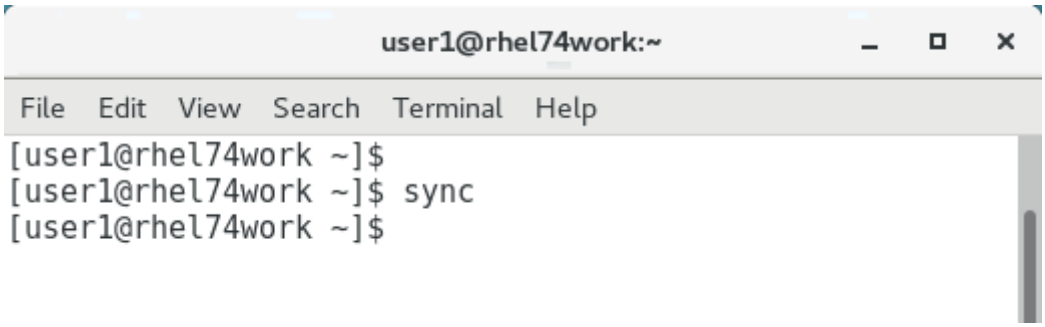
Unable to unmount the redirected USB disk

For the access control of all USB disks redirected from Citrix Workspace app, the Linux VDA manages all these devices under administrative privilege to ensure that only the owner can access the redirected device. As a result, the user cannot unmount the device without the administrative privilege.



File lost when you stop redirecting a USB disk

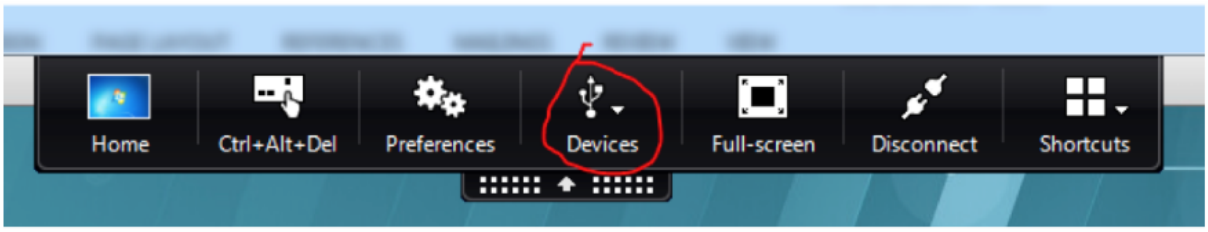
If you redirect a USB disk to a session and try to modify it (for example, create some files on the disk), then stop redirecting it immediately using the toolbar of Citrix Workspace app, the file you modified or created can be lost. This issue occurs because when you write data to a file system, the system mounts the memory cache in the file system. The data is not written to the disk itself. If you stop redirecting using the toolbar of Citrix Workspace app, there is no time remaining for the data being flushed to the disk, which results in lost data. To resolve this issue, use the sync command in a terminal to flush data to the disk before stopping USB redirection.



No devices in the toolbar of Citrix Workspace app

Sometimes, you might not be able to see devices listed in the toolbar of Citrix Workspace app, which indicates that no USB redirection is taking place. If you encounter the issue, verify the following:

- The policy is configured to allow USB redirection
- The Kernel module is compatible with your kernel



> **Note:**
>
> The **Devices** tab is not available in Citrix Workspace app for Linux.

Failed redirection when USB devices can be seen in the toolbar of Citrix Workspace app, but are labeled *policy restricted*

When the issue occurs, do the following:

- Configure the Linux VDA policy to enable redirection.
- Check whether any additional policy restrictions are configured in the registry of Citrix Workspace app. Check **DeviceRules** in the registry path to ensure that the device is not denied access by this setting:

  HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA Client\GenericUSB

For more information, see the Knowledge Center article How to Configure Automatic Redirection of USB Devices.

A USB device is redirected successfully, but I cannot use it in my session

Typically, only supported USB devices can be redirected. Other devices might be redirected to an active Linux VDA session too. For every redirected device, a node owned by the user is created in the system **/dev** path. However, it is the drivers and the configuration that determine whether the user can use the device successfully. If you find a device owned (plugged in) but inaccessible, add the device to an unrestricted policy.

> **Note:**
>
> In the case of USB drives, the Linux VDA configures and mounts the disk. The user (and only the owner who installed it) can access the disk without any additional configuration. This might not be the case for devices that are not in the supported device list.

Configure session reliability

June 11, 2021

Citrix introduces the session reliability feature to all supported Linux platforms. Session reliability is enabled by default.

Session reliability reconnects ICA sessions seamlessly across network interruptions. For more information about session reliability, see Auto client reconnect and session reliability.

Note: Data transmitted through a session reliability connection is in plain text by default. For security purposes, We recommend that you enable TLS encryption. For more information about TLS encryption, see Secure user sessions using TLS.

Configuration

Policy settings in Citrix Studio

You can set the following policies for session reliability in Citrix Studio:

- Session reliability connections
- Session reliability timeout
- Session reliability port number
- Reconnection UI transparency level

For more information, see Session reliability policy settings and Auto client reconnect policy settings.

Note: After setting the **Session reliability connections** or **Session reliability port number** policy, restart the VDA service and the HDX service, in this order, for your settings to take effect.

Settings on the Linux VDA

- **Enable/disable the session reliability TCP listener**

By default, the session reliability TCP listener is enabled and listening on port 2598. To disable the listener, run the following command.

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
     CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "
     fEnableWinStation" -d "0x00000000"
2 <!--NeedCopy-->
```


Note: Restart the HDX service for your settings to take effect. Disabling the TCP listener does not disable session reliability. Session reliability is still available through other listeners (for example, SSL) if the feature is enabled through the **Session reliability connections** policy.

- **Session reliability port number**

You can also set the session reliability port number by using the following command (using port number 2599 as an example).

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\SYSTEM\
     CurrentControlSet\Control\Citrix\WinStations\cgp" -v "PortNumber"
      -d "2599"
2 <!--NeedCopy-->
```

Note: Restart the HDX service for your setting to take effect. If the port number has been set through the policy setting in Citrix Studio, your setting on the Linux VDA is ignored. Ensure that the firewall on the VDA is configured not to prohibit network traffic through the set port.

- **Server-to-client keep-alive interval**

Session reliability keep-alive messages are sent between the Linux VDA and the ICA client when there is no activity in the session (for example, no mouse movement, no screen update). The keep-alive messages are used to detect whether the client is still responsive. If there is no response from the client, the session is suspended until the client reconnects. This setting specifies the number of seconds between successive keep-alive messages. By default, this setting is not configured. To configure it, run the following command (using 10 seconds as an example).

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
     Citrix\XTEConfig" -t "REG_DWORD" -v "CgpServerToClientKeepAlive"
      -d "10" --force
```

- **Client-to-server keep-alive interval**

This setting specifies the number of seconds between successive keep-alive messages sent from the ICA client to the Linux VDA. By default, this setting is not configured. To configure it, run the following command (using 10 seconds as an example).

```
1 /opt/Citrix/VDA/bin/ctxreg create -k "HKEY_LOCAL_MACHINE\SOFTWARE\
     Citrix\XTEConfig" -t "REG_DWORD" -v "CgpClientToServerKeepAlive"
      -d "10" --force
2 <!--NeedCopy-->
```

Troubleshooting

Unable to launch sessions after enabling session reliability through the policy setting.

To work around this issue, do the following:

1. Ensure that the VDA service and HDX service are restarted, in this order, after you enable session reliability through the policy setting in Citrix Studio.
2. On the VDA, run the following command to verify that the session reliability TCP listener is running (using port 2598 as an example).

```
netstat -an | grep 2598
<!--NeedCopy-->
```

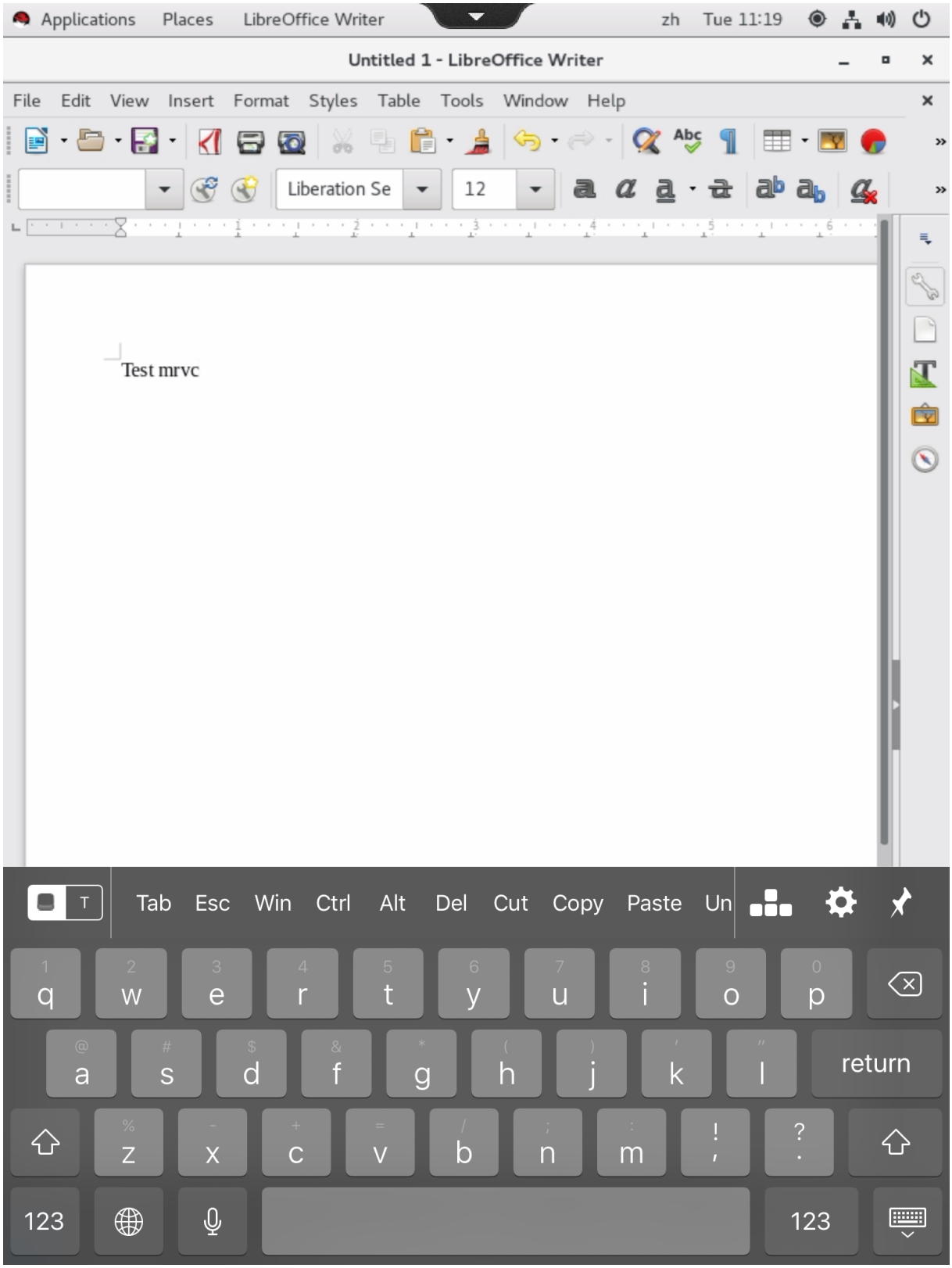
If there is no TCP listener on the session reliability port, enable the listener by running the following command.

```
/opt/Citrix/VDA/bin/ctxreg update -k  "HKEY_LOCAL_MACHINE\SYSTEM\
   CurrentControlSet\Control\Citrix\WinStations\cgp"   -v  "
   fEnableWinStation" -d "0x00000001"
<!--NeedCopy-->
```

Soft keyboard

August 18, 2022

The soft keyboard feature is available in a Linux virtual desktop or application session. The soft keyboard shows or hides automatically when you enter or leave an input field.



> **Note:**
>
> The feature is available for RHEL 7.8, RHEL 7.9, RHEL 8.1 RHEL 8.2, RHEL 8.3, SUSE 12.5, Ubuntu 16.04, Ubuntu 18.04, and Ubuntu 20.04. It is supported on Citrix Workspace app for iOS and for Android.

Enable and disable the feature

The feature is disabled by default. Use the **ctxreg** utility to enable or disable the feature. The feature configuration on a given Linux VDA applies to all sessions on that VDA.

To enable the feature:

1. Run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
   CurrentControlSet\Control\Citrix\VirtualChannels\MrVc"  -v "
   Enabled" -d "0x00000001"
<!--NeedCopy-->
```

2. In Citrix Studio, set the **Automatic keyboard display** policy to **Allowed**.
3. (Optional) For RHEL 7 and CentOS 7, run the following command to configure Intelligent Input Bus (IBus) as the default IM service:

```
echo "GTK_IM_MODULE=ibus" >>/etc/bashrc
<!--NeedCopy-->
```

To disable the feature, run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\CurrentControlSet\
   Control\Citrix\VirtualChannels\MrVc"  -v "Enabled" -d "0x00000000"
<!--NeedCopy-->
```

> **Note:**
>
> The preceding settings take effect when you log on to a new session or log off and back on to the current session.

Limitations

- The feature might not work as expected with Google Chrome, LibreOffice, and other apps.
- To display the soft keyboard again after hiding it manually, click a non-input field and then the current input field again.
- The soft keyboard might not appear when you click from one input field to another in a web browser. To work around this issue, click a non-input field and then the target input field.

- The feature does not support Unicode characters and double-byte characters (such as Chinese, Japanese, and Korean characters).
- The soft keyboard is not available for password input fields.
- The soft keyboard might overlap the current input field. In this case, move the app window or scroll up your screen to move the input field to an accessible position.
- Due to compatibility issues between Citrix Workspace app and Huawei tablets, the soft keyboard appears on Huawei tablets even with a physical keyboard connected.

Client Input Method Editor (IME)

June 11, 2021

Overview

Double-byte characters such as Chinese, Japanese, and Korean characters must be typed through an IME. Type such characters with any IME that is compatible with Citrix Workspace app on the client side, such as the Windows native CJK IME.

Installation

This feature is installed automatically when you install the Linux VDA.

Usage

Open a Citrix Virtual Apps or Citrix Virtual Desktops session as per usual.

Change your input method as required on the client side to start using the client IME feature.

Known issues

- Double-clicking a cell in a Google spreadsheet is a must before you can use the client IME feature to type characters in the cell.
- The client IME feature is not disabled automatically in Password fields.
- The IME user interface does not follow the cursor in the input area.

Support for multiple language inputs

June 11, 2021

As of the Linux VDA Version 1.4, Citrix has added support for published applications. Users can access a desired Linux application without the Linux desktop environment.

However, the native language bar on the Linux VDA was unavailable to the published application because the language bar is highly integrated with the Linux desktop environment. As a result, users were unable to input text in a language that requires IME such as Chinese, Japanese, or Korean. It was also not possible for users to switch between keyboard layouts during an application session.

To address those issues, this feature provides a language bar for published applications that accept text input. The language bar enables users to select a server-side IME and to switch between keyboard layouts during an application session.

Configuration

You can use the **ctxreg** utility to enable or disable this feature (disabled by default). The feature configuration on a given Linux VDA server applies to all applications published on that VDA.

The configuration key is "HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Control\Citrix\LanguageBar" and the type is DWORD.

To enable this feature, run the command:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE \SYSTEM\
     CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
     x00000001"
2 <!--NeedCopy-->
```

To disable this feature, run the command:

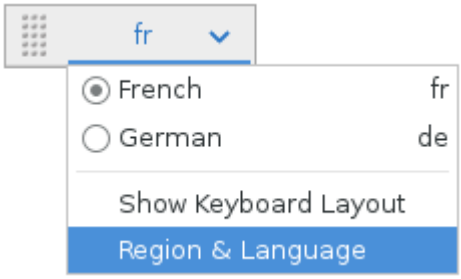
```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE \SYSTEM\
     CurrentControlSet\Control\Citrix\LanguageBar" -v "Enabled" -d "0
     x00000000"
2 <!--NeedCopy-->
```

Usage

The usage is straightforward.

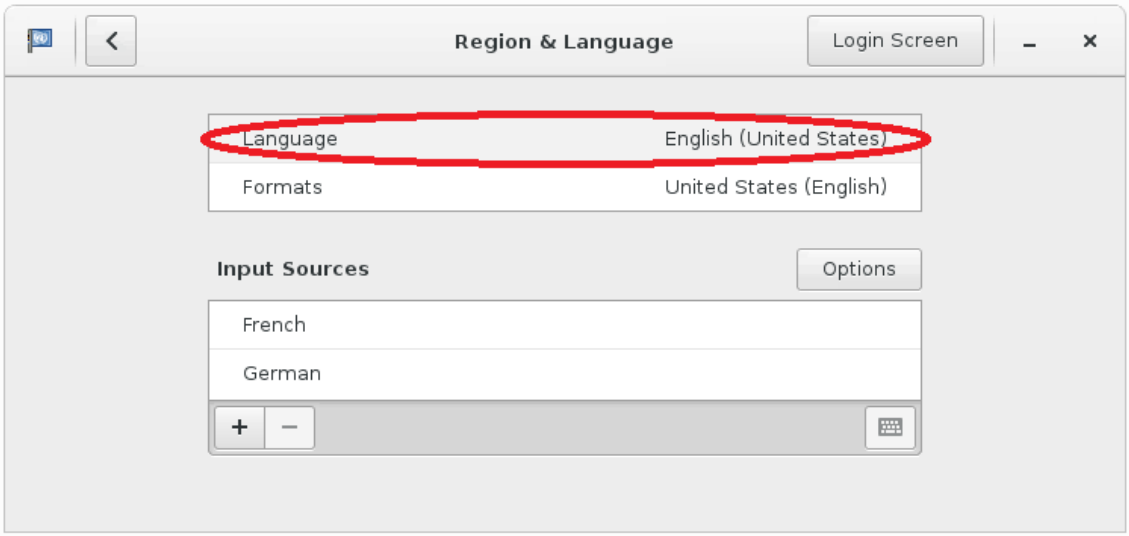
1. Enable the feature.

2. Access a published application that can accept text input. A language bar appears in the session, alongside the application.
3. From the drop-down menu, select **Region & Language** to add the desired language (input source).



4. Select the IME or keyboard layout from the drop-down menu.
5. Type a language using the selected IME or keyboard layout.

> **Note:**
>
> - When you change a keyboard layout on the VDA-side language bar, ensure that the same keyboard layout is used on the client side (running Citrix Workspace app).
> - The **accountsservice** package must be upgraded to Version 0.6.37 or later before you can perform settings in the **Region & Language** dialog box.



Dynamic keyboard layout synchronization

June 11, 2021

Previously, the keyboard layouts on the Linux VDA and on the client device had to be the same. For example, when the keyboard layout changed from English to French on the client device but not on the VDA, key mapping issues might occur and persist until the VDA changed to French too.

Citrix addresses the issue by synchronizing the keyboard layout of the VDA with the keyboard layout of the client device automatically. Anytime the keyboard layout on the client device changes, the layout on the VDA follows suit.

> **Note:**
>
> Citrix Workspace app for HTML5 does not support the dynamic keyboard layout synchronization feature.

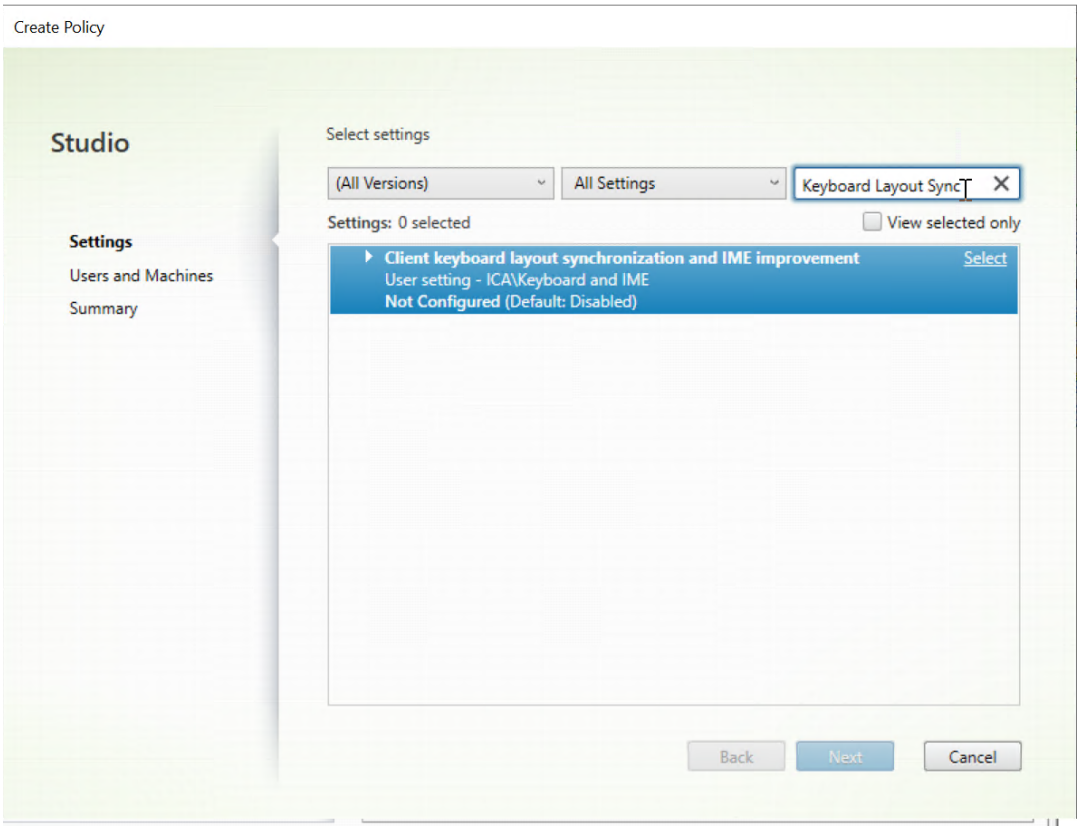
Configuration

The dynamic keyboard layout synchronization feature is disabled by default. To enable or disable the feature, set the **Client Keyboard Layout Sync and IME Improvement** policy or edit the registry through the `ctxreg` utility.

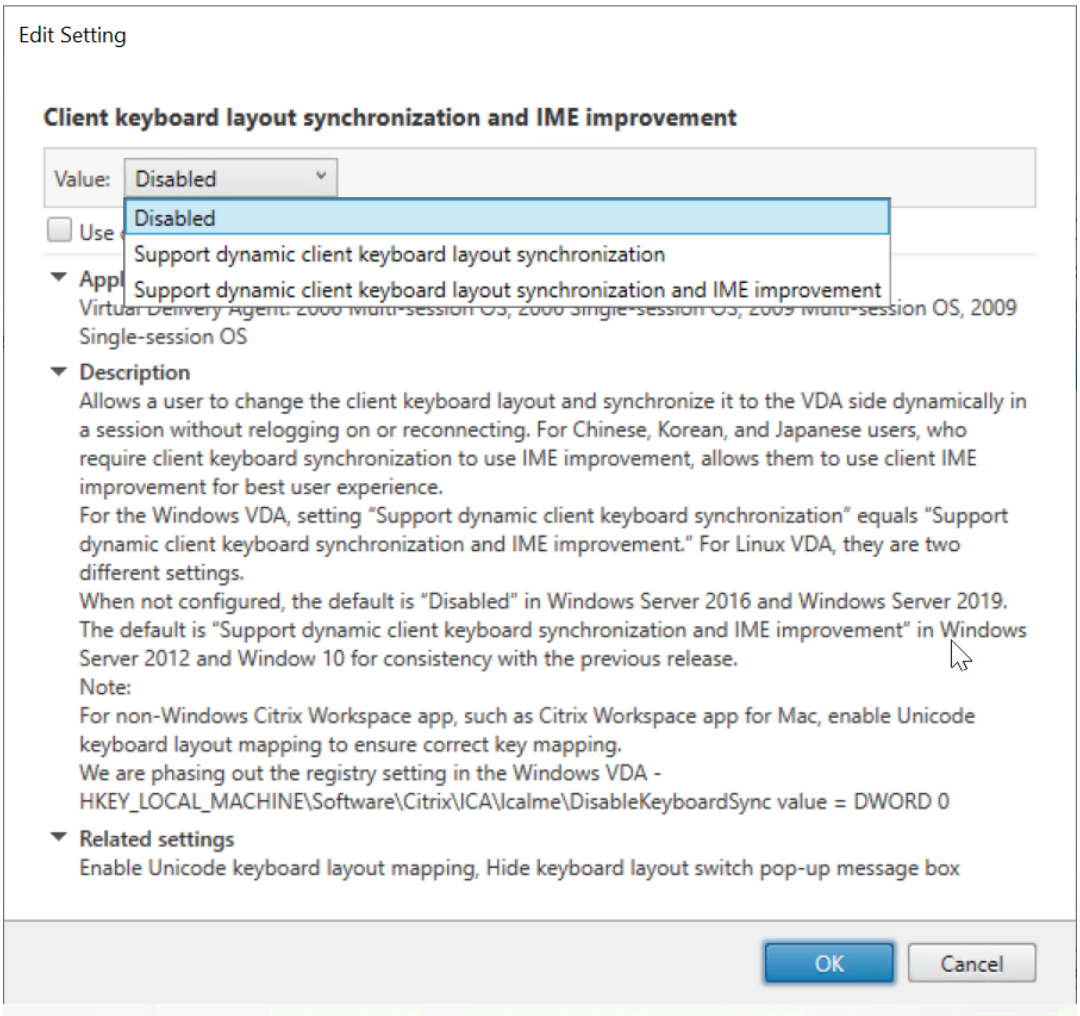
> **Note:**
>
> The **Client Keyboard Layout Sync and IME Improvement** policy takes priority over registry settings and can be applied to user and machine objects you specify or all objects in your site. Registry settings on a given Linux VDA apply to all sessions on that VDA.

- Set the **Client Keyboard Layout Sync and IME Improvement** policy to enable or disable the dynamic keyboard layout synchronization feature:

  1. In Studio, right-click **Policies** and select **Create Policy**.
  2. Search for the **Client Keyboard Layout Sync and IME Improvement** policy.



3. Click **Select** next to the policy name.
4. Set the policy.



There are three options available:

- **Disabled**: disables dynamic keyboard layout synchronization and client IME user interface synchronization.
- **Support dynamic client keyboard layout synchronization**: enables dynamic keyboard layout synchronization regardless of the DWORD value of the **SyncKeyboardLayout** registry key at `HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control\Citrix\LanguageBar`.
- **Support dynamic client keyboard layout synchronization and IME improvement**: enables both dynamic keyboard layout synchronization and client IME user interface synchronization regardless of the DWORD values of the **SyncKeyboardLayout** and **SyncClientIME** registry keys at `HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control\Citrix\LanguageBar`.

- Edit the registry through the `ctxreg` utility to enable or disable the dynamic keyboard layout synchronization feature:

To enable this feature, run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Control\Citrix\LanguageBar" -v "
   SyncKeyboardLayout" -d "0x00000001"
<!--NeedCopy-->
```

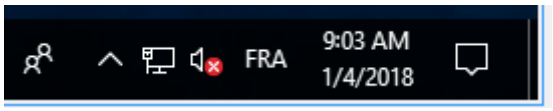
To disable this feature, run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Control\Citrix\LanguageBar" -v "
   SyncKeyboardLayout" -d "0x00000000"
<!--NeedCopy-->
```

Usage

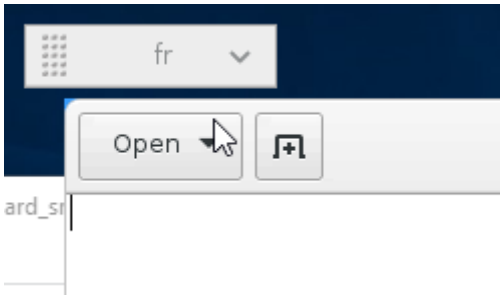
With this feature enabled, when the keyboard layout changes on the client device during a session, the keyboard layout of the session changes accordingly.

For example, if you change the keyboard layout on a client device to French (FR):

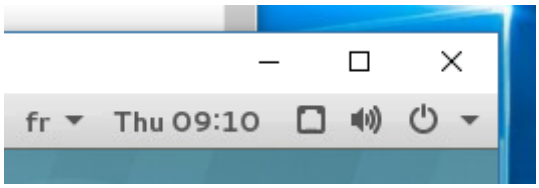


Then the keyboard layout of the Linux VDA session also changes to “fr.”

In an application session, you can see this automatic change if you have enabled the language bar:



In a desktop session, you can see this automatic change in the task bar:

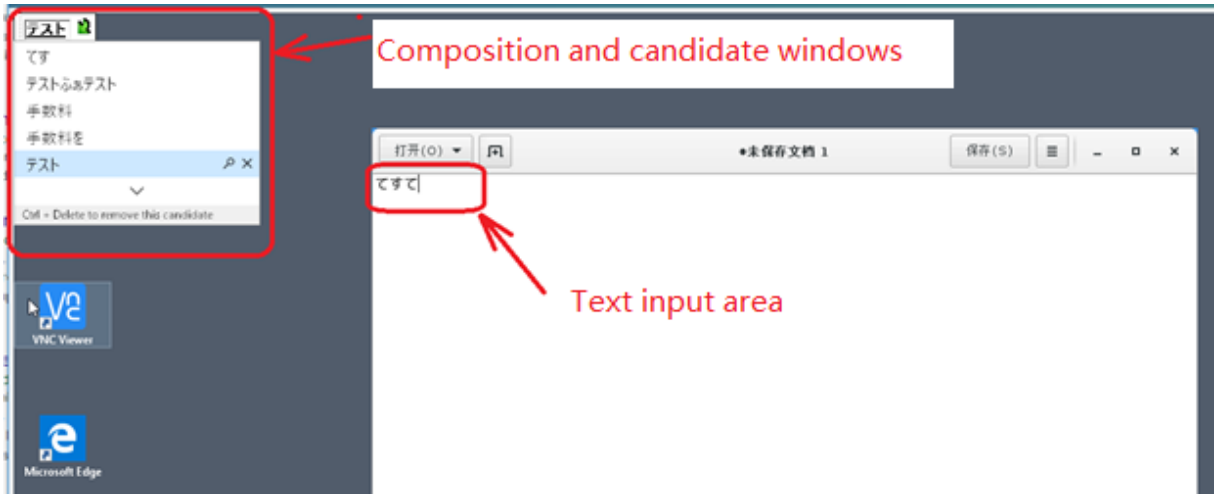


Client IME user interface synchronization

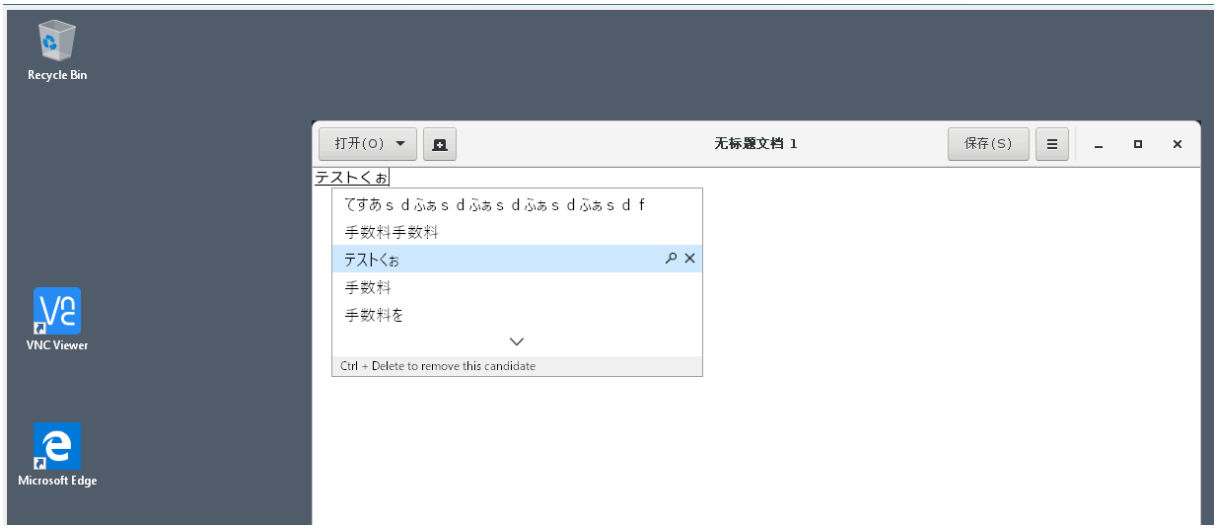
June 11, 2021

Overview

To date, the client IME user interface (including the composition window and candidate window) was positioned in the upper left corner of the screen. It did not follow the cursor and sometimes was located far from the cursor in the text input area:



Citrix enhances usability and further improves the user experience with the client IME as follows:



> **Note:**
>
> The feature is available for RHEL 7.x, CentOS 7.x, Ubuntu 16.04, Ubuntu 18.04, and SUSE 12.x. It

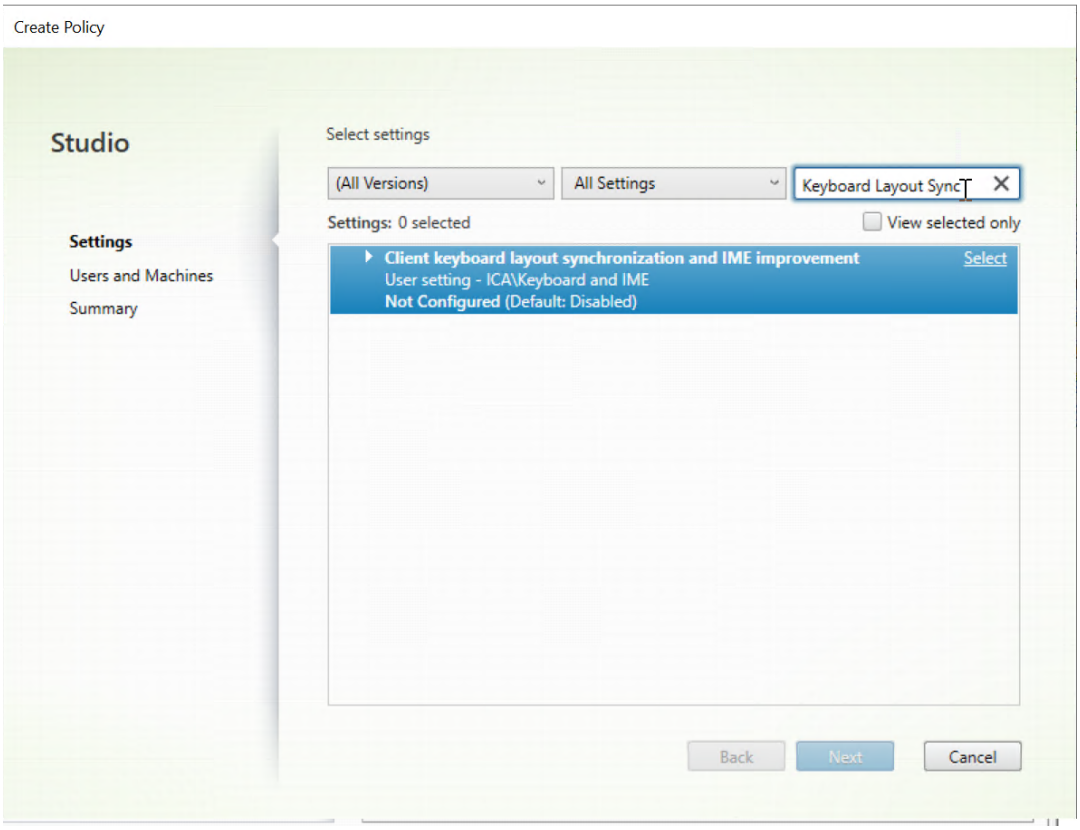
> is supported on Citrix Workspace app for Windows and for Mac.
>
> To use the feature in RHEL 7.x desktop sessions, you must enable `IBus`. For example, set the user interface language to one that requires an IME to input, or add **GTK_IM_MODULE=ibus** to the **${HOME}/.config/imsettings/xinputrc** file.
>
> The feature installs automatically, but you must enable the feature before you can use it.

Enable and disable the feature

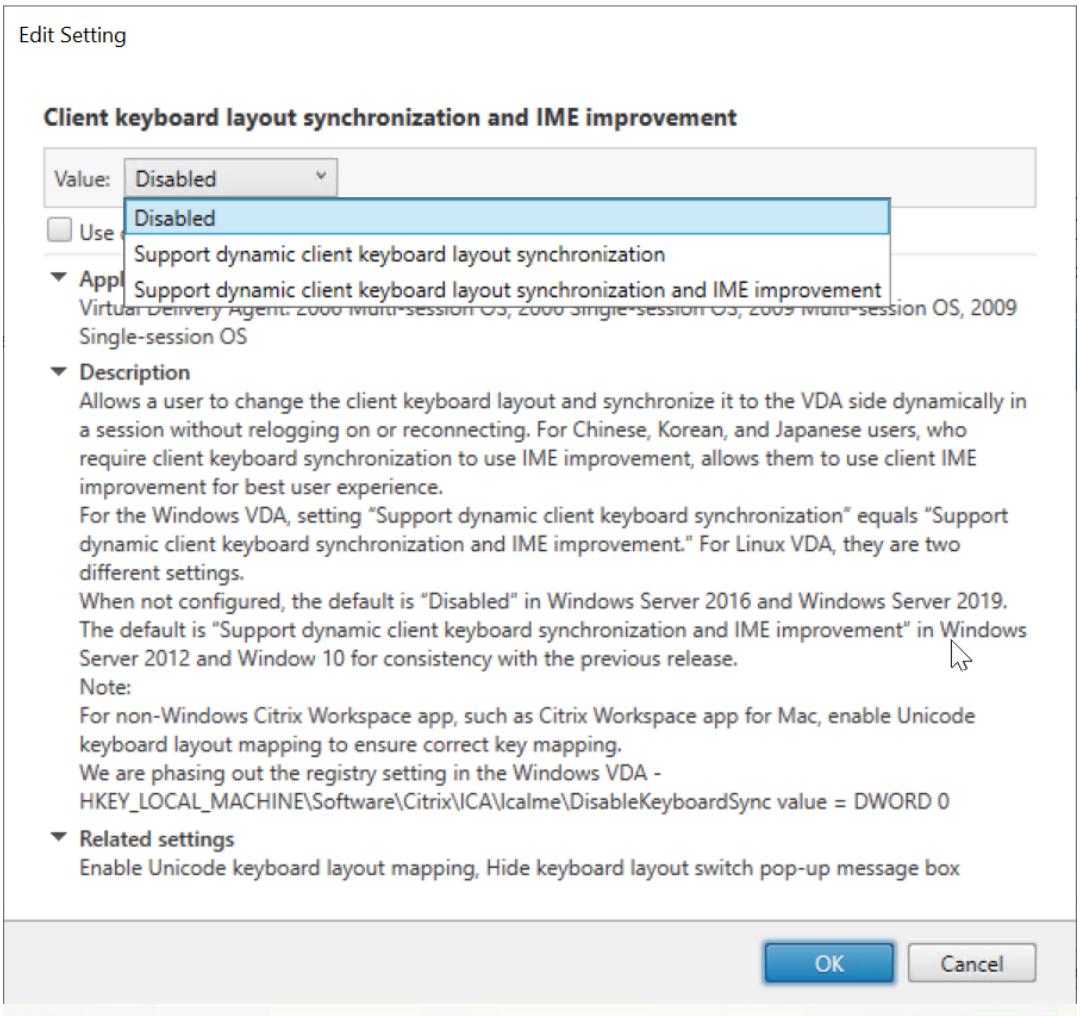
The client IME user interface synchronization feature is disabled by default. To enable or disable the feature, set the **Client Keyboard Layout Sync and IME Improvement** policy or edit the registry through the `ctxreg` utility.

> **Note:**
>
> The **Client Keyboard Layout Sync and IME Improvement** policy takes priority over registry settings and can be applied to user and machine objects you specify or all objects in your site. Registry settings on a given Linux VDA apply to all sessions on that VDA.

- Set the **Client Keyboard Layout Sync and IME Improvement** policy to enable or disable the client IME user interface synchronization feature:
  1. In Studio, right-click **Policies** and select **Create Policy**.
  2. Search for the **Client Keyboard Layout Sync and IME Improvement** policy.



3. Click **Select** next to the policy name.
4. Set the policy.



There are three options available:

- **Disabled**: disables dynamic keyboard layout synchronization and client IME user interface synchronization.
- **Support dynamic client keyboard layout synchronization**: enables dynamic keyboard layout synchronization regardless of the DWORD value of the **SyncKeyboardLayout** registry key at `HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control\Citrix\LanguageBar`.
- **Support dynamic client keyboard layout synchronization and IME improvement**: enables both dynamic keyboard layout synchronization and client IME user interface synchronization regardless of the DWORD values of the **SyncKeyboardLayout** and **SyncClientIME** registry keys at `HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control\Citrix\LanguageBar`.

- Edit the registry through the `ctxreg` utility to enable or disable the client IME user interface synchronization feature:

To enable the feature, run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Control\Citrix\LanguageBar" -v  "
   SyncClientIME" -d "0x00000001"
<!--NeedCopy-->
```

To disable the feature, run the command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Control\Citrix\LanguageBar" -v "
   SyncClientIME" -d "0x00000000"
<!--NeedCopy-->
```

HDX Insight

June 11, 2021

Overview

The Linux VDA partially supports the HDX Insight feature. HDX Insight is part of the Citrix Application Delivery Management (ADM) and is based on the popular industry standard AppFlow. It enables IT to deliver an exceptional user experience by providing unprecedented end-to-end visibility into the Citrix ICA traffic that passes through the Citrix ADC or Citrix SD-WAN application networking fabric. For more information, see HDX Insight

Installation

No dependent packages need installation.

Usage

HDX Insight analyzes the ICA messages passed through the Citrix ADC between Citrix Workspace app and the Linux VDA. All HDX Insight data is sourced from the NSAP virtual channel and sent uncompressed. The NSAP virtual channel is enabled by default.

The following commands disable and enable the NSAP virtual channel, respectively:

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
   VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP" -d "0x00000000"
   --force
<!--NeedCopy-->
```



```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
    VirtualDesktopAgent" -t "REG_DWORD" -v "EnableNSAP" -d "0x00000001"
    --force
<!--NeedCopy-->
```

Troubleshooting

No data points are displayed

There might be two causes:

- HDX Insight is not configured correctly.

  For example, AppFlow is not enabled on the Citrix ADC or an incorrect Citrix ADC instance is configured on the Citrix ADM.

- The ICA Control Virtual Channel is not started on the Linux VDA.

  ```
  ps aux | grep -i ctxctl
  ```

  If `ctxctl` is not running, contact your administrator to report a bug to Citrix.

No application data points are displayed

Verify that the seamless virtual channel is enabled and a seamless application is launched for a while.

Rendezvous protocol

October 8, 2021

In environments that use the Citrix Gateway service, the Rendezvous protocol allows HDX sessions to bypass the Citrix Cloud Connector and connect directly and securely to the Citrix Gateway service.

Requirements:

- Access to environment using Citrix Workspace and Citrix Gateway service.
- Control Plane: Citrix Virtual Apps and Desktops Service (Citrix Cloud).
- Linux VDA Version 2012 or later.
- Enable the Rendezvous protocol in the Citrix policy. For more information, see Rendezvous protocol policy setting.

- The VDAs must have access to `https://*.nssvc.net`, including all subdomains. If you cannot whitelist all subdomains in that manner, use `https://*.c.nssvc.net` and `https://*.g.nssvc.net` instead. For more information, see the Internet Connectivity Requirements section of the Citrix Cloud documentation (under Virtual Apps and Desktop service) and the Knowledge Center article CTX270584.
- Cloud Connectors must obtain the VDAs' FQDNs when brokering a session. To achieve this goal, enable DNS resolution for the site: Using the Citrix Virtual Apps and Desktops Remote PowerShell SDK, run the command `Set-BrokerSite -DnsResolutionEnabled $true`. For more information about the Citrix Virtual Apps and Desktops Remote PowerShell SDK, see SDKs and APIs.

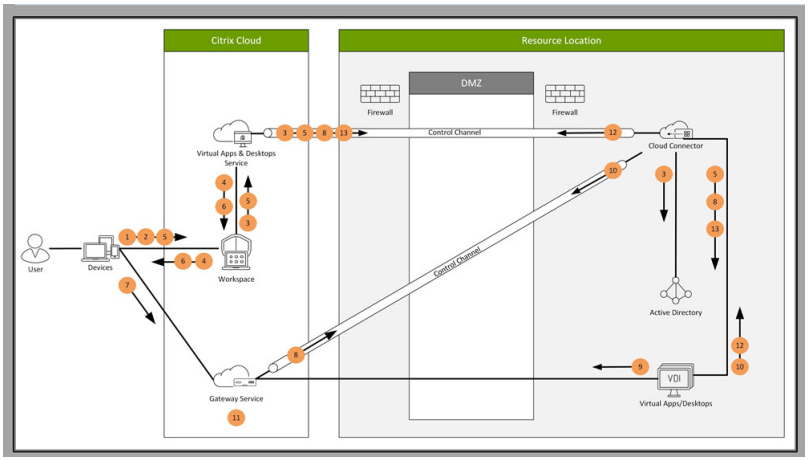
> **Important:**
>
> The Rendezvous protocol doesn't support transparent or explicit proxies. To use proxies, continue to use the Cloud Connector for ICA traffic.

If you enable Rendezvous and the VDA cannot reach the Citrix Gateway service directly, the VDA falls back to proxy the HDX session through the Cloud Connector.

If you meet all requirements, follow these steps to validate if Rendezvous is in use:

1. Launch a terminal on the VDA.
2. Run `su root -c "/opt/Citrix/VDA/bin/ctxquery -f iuStdP"`.
3. The TRANSPORT PROTOCOLS indicates the type of connection:
    - TCP Rendezvous: TCP - SSL - CGP - ICA
    - EDT Rendezvous: UDP - DTLS - CGP - ICA
    - Proxy through Cloud Connector: TCP - CGP - ICA

This diagram is an overview of the Rendezvous connection flow. Follow the steps to understand the flow.



1. Navigate to Citrix Workspace.

2. Enter credentials in Citrix Workspace.
3. If using on-premises Active Directory, the Citrix Virtual Apps and Desktops service authenticates credentials with Active Directory using the Cloud Connector channel.
4. Citrix Workspace displays enumerated resources from the Citrix Virtual Apps and Desktops service.
5. Select resources from Citrix Workspace. The Citrix Virtual Apps and Desktops service sends a message to the VDA to prepare for an incoming session.
6. Citrix Workspace sends an ICA file to the endpoint that contains an STA ticket generated by Citrix Cloud.
7. The endpoint connects to the Citrix Gateway service, provides the ticket to connect to the VDA, and Citrix Cloud validates the ticket.
8. The Citrix Gateway service sends connection information to the Cloud Connector. The Cloud Connector determines if the connection is supposed to be a Rendezvous connection and sends the information to the VDA.
9. The VDA establishes a direct connection to the Citrix Gateway service.
10. If a direct connection between the VDA and the Citrix Gateway service isn't possible, the VDA proxies its connection over the Cloud Connector.
11. The Citrix Gateway service establishes a connection between the endpoint device and the VDA.
12. The VDA verifies its license with the Citrix Virtual Apps and Desktops service through the Cloud Connector.
13. The Citrix Virtual Apps and Desktops service sends session policies to the VDA through the Cloud Connector. Those policies are applied.

Adaptive transport

June 11, 2021

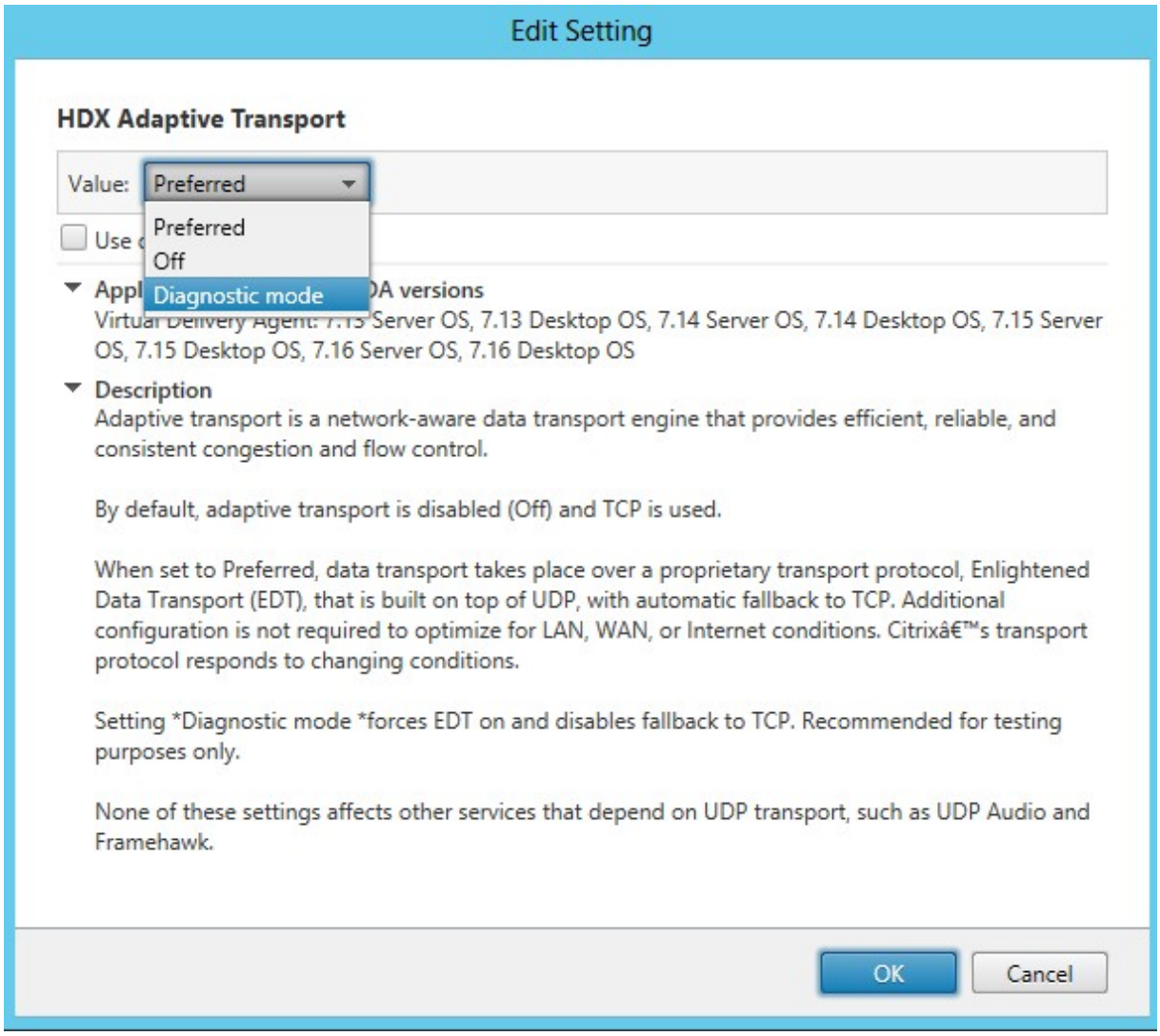
Adaptive transport is a data transport mechanism for Citrix Virtual Apps and Desktops. It is faster, more scalable, improves application interactivity, and is more interactive on challenging long-haul WAN and internet connections. For more information about adaptive transport, see Adaptive transport.

Enable adaptive transport

In Citrix Studio, verify that the **HDX Adaptive Transport** policy is set to **Preferred** or **Diagnostic mode**. **Preferred** is selected by default.

- **Preferred**: Adaptive transport over Enlightened Data Transport (EDT) is used when possible, with fallback to TCP.

- **Diagnostic mode**: EDT is forced on and fallback to TCP is disabled.



Disable adaptive transport

To disable adaptive transport, set the **HDX Adaptive Transport** policy to **Off** in Citrix Studio.

Check whether adaptive transport is enabled

To check whether UDP listeners are running, run the following command.

```
netstat -an | grep "1494\|2598"
<!--NeedCopy-->
```

In normal circumstances, the output is similar to the following.


```
udp        0      0 0.0.0.0:2598            0.0.0.0:*

udp       0      0 :::1494                 :::*
<!--NeedCopy-->
```

EDT MTU discovery

EDT automatically determines the Maximum Transmission Unit (MTU) when establishing a session. Doing so prevents EDT packet fragmentation that might result in performance degradation or failure to establish a session.

Minimum requirements:

- Linux VDA 2012
- Citrix Workspace app 1911 for Windows
- Citrix ADC:
  - 13.0.52.24
  - 12.1.56.22
- Session reliability must be enabled

If using client platforms or versions that do not support this feature, see Knowledge Center article CTX231821 for details about how to configure a custom EDT MTU that is appropriate for your environment.

> **WARNING:**
>
> Editing the registry incorrectly can cause serious problems that might require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

Enable or disable EDT MTU discovery on the VDA

EDT MTU discovery is disabled by default.

- To enable EDT MTU discovery, set the `MtuDiscovery` registry key by using the following command, restart the VDA, and wait for the VDA to register:

  ```
  /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\CurrentControlSet
  \Control\Terminal Server\Wds\icawd"-t "REG_DWORD"-v "MtuDiscovery
  "-d "0x00000001"--force
  ```

- To disable EDT MTU discovery, delete the `MtuDiscovery` registry value.

This setting is machine-wide and affects all sessions connecting from a supported client.

Control EDT MTU discovery on the client

You can control EDT MTU discovery selectively on the client by adding the `MtuDiscovery` parameter in the ICA file. To disable the feature, set the following under the `Application` section:

```
MtuDiscovery=Off
```

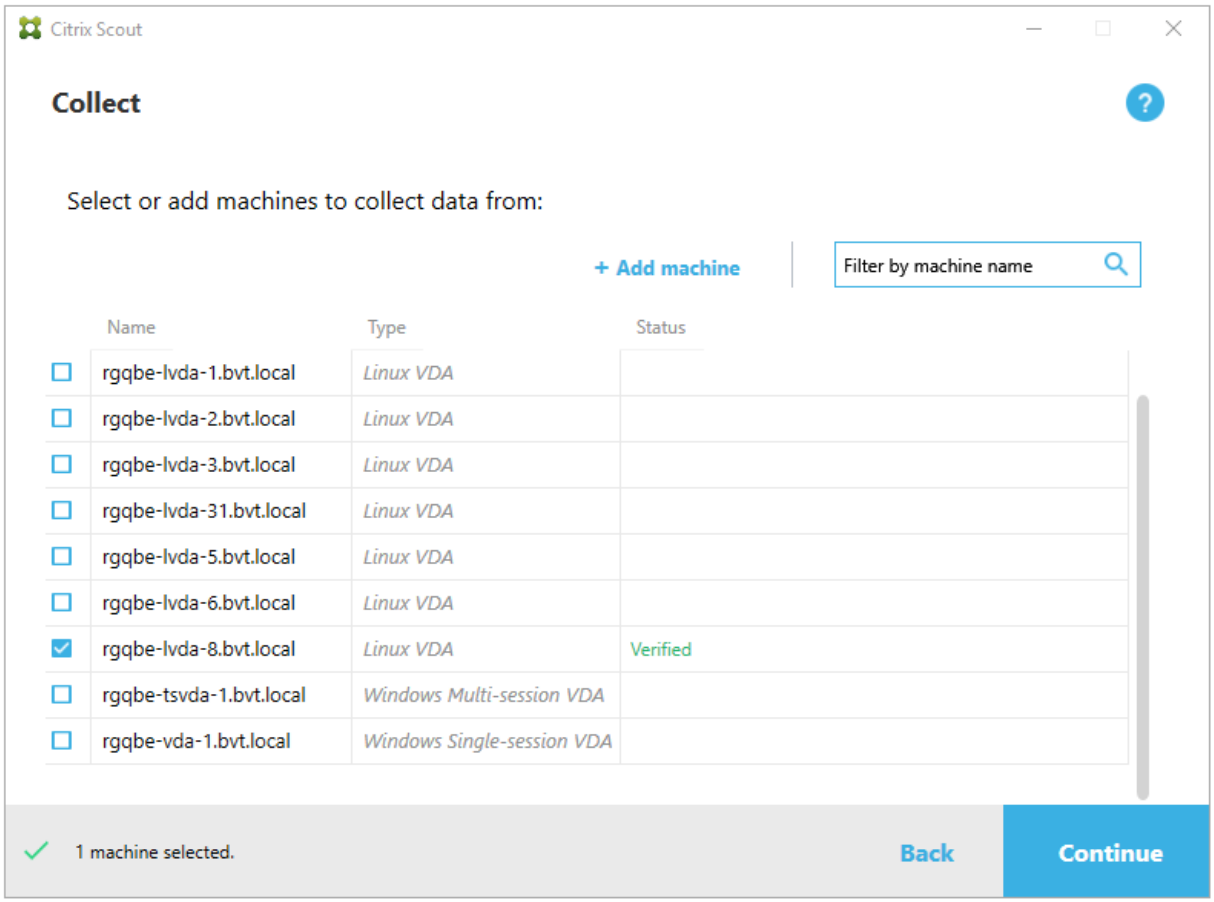
To re-enable the feature, remove the `MtuDiscovery` parameter from the ICA file.

> **IMPORTANT:**
>
> For this ICA file parameter to work, enable EDT MTU discovery on the VDA. If EDT MTU discovery is not enabled on the VDA, the ICA file parameter has no effect.

Integrate with the Citrix Telemetry Service

June 11, 2021

With the Citrix Telemetry Service (ctxtelemetry) integrated with the Linux VDA software, you can run Citrix Scout, which then uses the /opt/Citrix/VDA/bin/xdlcollect.sh script, to collect logs about the Linux VDA.



> **Note:**
>
> After upgrading from Linux VDA 1912 and earlier versions, you must rerun /opt/Citrix/VDA/sbin/c-txsetup.sh to configure the variables for the Citrix Telemetry Service (ctxtelemetry). For more information about the variables, see Easy install.

Enable and disable the Citrix Telemetry Service

- To enable the service, run the **sudo systemctl enable ctxtelemetry.socket** command.
- To disable the service, run **sudo systemctl disable ctxtelemetry.socket**.

Ports

The Citrix Telemetry Service (ctxtelemetry), by default, uses TCP/IP port 7503 to listen for Citrix Scout. It uses TCP/IP port 7502 on the Delivery Controller to communicate with Citrix Scout.

You can use the default ports or change ports through the following variables when you install the Linux VDA.

- **CTX_XDL_TELEMETRY_SOCKET_PORT** –The socket port for listening for Citrix Scout. The default port is 7503.
- **CTX_XDL_TELEMETRY_PORT** –The port for communicating with Citrix Scout. The default port is 7502.

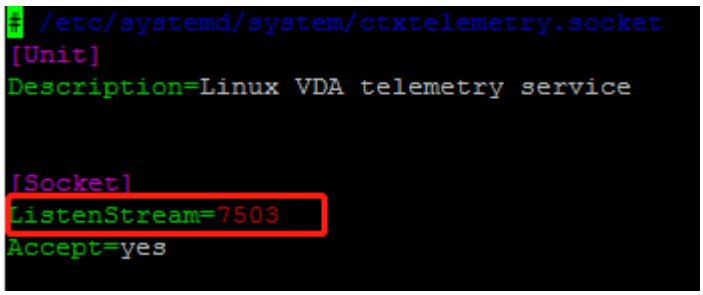
To change ports after you have your VDA installed, do the following:

1. To change a port for communicating with Scout, run the following command.

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
   VirtualDesktopAgent" -v "TelemetryServicePort" -d <port number>
    -t REG_DWORD
<!--NeedCopy-->
```

2. To change the socket port for listening for Scout, run the following command to open and edit the ctxtelemetry.socket file.

```
sudo vi /etc/systemd/system/ctxtelemetry.socket
<!--NeedCopy-->
```



3. Run the following commands to restart the socket port.

```
sudo systemctl daemon-reload
sudo systemctl stop ctxtelemetry.socket
sudo systemctl start ctxtelemetry.socket
<!--NeedCopy-->
```

4. Enable the new ports in your firewall configuration.

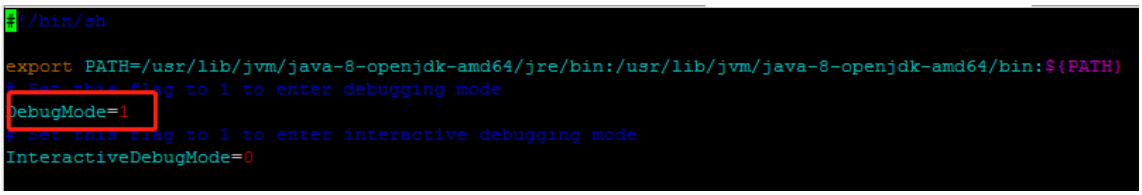
   If you are using a Ubuntu distribution, for example, run the **sudo ufw allow 7503** command to enable port 7503.

Debug mode

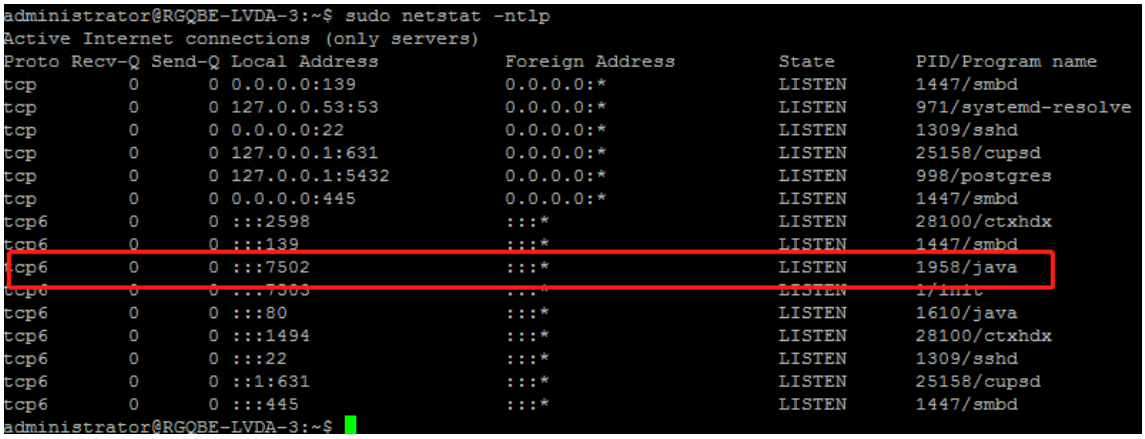
If the Citrix Telemetry Service does not work as expected, you can enable debug mode to determine the causes.

1. To enable debug mode, run the following command to open the ctxtelemetry file and then change the DebugMode value to 1.


```
sudo vi /opt/Citrix/VDA/sbin/ctxtelemetry
<!--NeedCopy-->
```



2. Manually stop the Citrix Telemetry Service, or wait 15 minutes for the service to stop automatically.



   In this example, you can run the following commands to stop the Citrix Telemetry Service.

```
sudo netstat -ntlp
Kill -9 1958
<!--NeedCopy-->
```

3. To restart the Citrix Telemetry Service, select your Linux VDA on Scout and find telemetry-debug.log in /var/log/xdl/.

Service wait time

The `systemd` daemon that opens the socket port starts by default and uses few resources. The Citrix Telemetry Service stops by default and starts only when there is a log collection request from the Delivery Controller. After log collection completes, the service awaits new collection requests for a duration of 15 minutes and stops again if there are not any. You can configure the wait time through the following command. The minimum value is 10 minutes. If you set a value less than 10 minutes, the minimum value, 10 minutes, takes effect. After setting the wait time, stop and restart the service.

```
/opt/Citrix/VDA/bin/ctxreg create -k "HKLM\Software\Citrix\
   VirtualDesktopAgent" -v "TelemetryServiceIdleTimeoutInMinutes" -d <
   number> -t REG_DWORD
<!--NeedCopy-->
```


Verification tests

Before a collection starts, verification tests run automatically for each selected machine. These tests ensure that the requirements are met. If a test fails for a machine, Scout displays a message, with suggested corrective actions. For more information about verification tests, see the Verification tests section in the Citrix Scout documentation.

Tracing On

June 11, 2021

Overview

Collecting logs and reproducing issues slow down the diagnostics and degrade the user experience. The Tracing On feature eases such efforts. Tracing is enabled for the Linux VDA by default.

Configuration

The `ctxlogd` daemon and the `setlog` utility are now included in the Linux VDA release package. By default, the `ctxlogd` daemon starts after you install and configure the Linux VDA.

ctxlogd daemon

All the other services that are traced depend on the `ctxlogd` daemon. You can stop the `ctxlogd` daemon if you do not want to keep the Linux VDA traced.

setlog utility

Tracing On is configured using the `setlog` utility, which is under the **/opt/Citrix/VDA/bin/** path. Only the root user has the privilege to run it. You can use the GUI or run commands to view and change the configurations. Run the following command for help with the `setlog` utility:

```
setlog help
<!--NeedCopy-->
```


Values By default, **Log Output Path** is set to **/var/log/xdl/hdx.log**, **Max Log Size** is set to 200 MB, and you can save up to two old log files under **Log Output Path**.

View the current `setlog` values:

```
setlog values

log_path (Log Output Path) = /var/log/xdl/hdx.log

log_size (Max Log Size (MiB)) = 200

log_count (Max Old Log Files) = 2
<!--NeedCopy-->
```

View or set a single `setlog` value:

```
setlog value <name> [<value>]
<!--NeedCopy-->
```

For example:

```
setlog value log_size 100
<!--NeedCopy-->
```

Levels By default, the log level is set to **Warnings**.

View the log levels set for different components:

```
setlog levels
<!--NeedCopy-->
```

You can set all log levels (including Disable, Inherited, Verbose, Information, Warnings, Errors, and Fatal Errors) by using the following command:

```
setlog level <class> [<level>]
<!--NeedCopy-->
```

The **<class>** variable specifies one component of the Linux VDA. To cover all components, set it to all:

```
setlog level all error

Setting log class ALL to ERROR.
<!--NeedCopy-->
```

Flags By default, the flags are set as follows:

```
setlog flags

```



```
DATE = true

TIME = true

NAME = true

PID = true

TID = false

SID = true

UID = false

GID = false

CLASS = false

LEVEL = false

FUNC = true

FILE = false
<!--NeedCopy-->
```

View the current flags:

```
setlog flags
<!--NeedCopy-->
```

View or set a single log flag:

```
setlog flag <flag> [<state>]
<!--NeedCopy-->
```

Restore Defaults Revert all levels, flags, and values to the default settings:

```
setlog default
<!--NeedCopy-->
```

> **Important:**
>
> The `ctxlogd` service is configured using the **/var/xdl/.ctxlog** file, which only root users can create. Other users do not have write permission to this file. We recommend that root users not give write permission to other users. Failure to comply can cause the arbitrary or malicious configuration to `ctxlogd`, which can affect server performance and therefore the user experience.

Troubleshooting

The `ctxlogd` daemon fails and you cannot restart the `ctxlogd` service when the **/var/xdl/.ctxlog** file is missing (for example, accidentally deleted).

/var/log/messages:

```
Apr 1 02:28:21 RH72 citrix-ctxlogd[17881]: Failed to open logging
   configuration file.

Apr 1 02:28:21 RH72 systemd: ctxlogd.service: main process exited, code
   =exited, status=1/FAILURE

Apr 1 02:28:21 RH72 systemd: Unit ctxlogd.service entered failed state.

Apr 1 02:28:21 RH72 systemd: ctxlogd.service failed.
<!--NeedCopy-->
```

To solve this issue, run `setlog` as a root user to recreate the **/var/xdl/.ctxlog** file. Then restart the `ctxlogd` service on which other services depend.

Shadow sessions

June 11, 2021

The session shadowing feature allows domain administrators to view users' ICA sessions in an intranet. The feature uses noVNC to connect to the ICA sessions and is supported only with RHEL 7.x and Ubuntu 16.04.

> **Note:**
>
> To use the session shadowing feature, the version of Citrix Director must be 7.16 or later.

Installation and configuration

Dependencies

Two new dependencies, `python-websockify` and `x11vnc`, are required for session shadowing. The `python-websockify` and `x11vnc` dependencies are installed automatically when you install the Linux VDA on Ubuntu 16.04. On RHEL 7.x, you must install `python-websockify` and `x11vnc` manually after you install the Linux VDA.

Run the following command on RHEL 7.x to install `python-websockify` and `x11vnc` (`x11vnc` version 0.9.13 or later).


```
sudo yum install -y python-websockify x11vnc
<!--NeedCopy-->
```

To resolve `python-websockify` and `x11vnc`, enable the following repositories on RHEL 7.x:

- Extra Packages for Enterprise Linux (EPEL)

  The EPEL repository is required for both `python-websockify` and `x11vnc`. Run the following command to enable the EPEL repository:

  ```
  sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(rpm -E '%{
  rhel }
  ').noarch.rpm
  <!--NeedCopy-->
  ```

- Optional RPMs

  Run either of the following commands to enable the optional RPMs repository for installing some dependency packages of `x11vnc`:

  For workstation:

  ```
  subscription-manager repos --enable=rhel-7-workstation-optional-rpms
  <!--NeedCopy-->
  ```

  For server:

  ```
  subscription-manager repos --enable=rhel-7-server-optional-rpms
  <!--NeedCopy-->
  ```

Port

The session shadowing feature automatically selects available ports from within 6001-6099 to build up connections from the Linux VDA to Citrix Director. Therefore, the number of ICA sessions that you can shadow concurrently is limited to 99. Ensure that enough ports are available to meet your requirements, especially for multi-session shadowing.

Registry

The following table lists related registries:

Registry	Description	Default Value
EnableSessionShadowing	Enables or disables the session shadowing feature	1 (Enabled)
ShadowingUseSSL	Determines whether to encrypt the connection between the Linux VDA and Citrix Director	0 (Disabled)

Run the `ctxreg` command on the Linux VDA to change the registry values. For example, to disable session shadowing, run the following command:

```
1 /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
     VirtualDesktopAgent" -v "EnableSessionShadowing" -d 0x00000000
```

SSL

The noVNC connection between the Linux VDA and Citrix Director uses the WebSocket protocol. For session shadowing, whether `ws://` or `wss://` is chosen is determined by the previously mentioned "ShadowingUseSSL"registry. By default, `ws://` is chosen. However, for security reasons, we recommend that you use `wss://` and install certificates on each Citrix Director client and on each Linux VDA server. Citrix disclaims any security responsibility for the Linux VDA session shadowing by using `ws://`.

Obtain server and root SSL certificates Certificates must be signed by a trusted Certificate Authority (CA).

A separate server certificate (including the key) is required for each Linux VDA server on which you want to configure SSL. A server certificate identifies a specific computer, so you must know the Fully Qualified Domain Name (FQDN) of each server. For convenience, you can use a wildcard certificate for the whole domain instead. In this case, you must know at least the domain name.

In addition to installing a server certificate on each server, you must install a root certificate from the same CA on each Citrix Director client that communicates with the Linux VDA server. Root certificates are available from the same CAs that issue the server certificates. You can install server and client certificates from a CA that is bundled with your operating system, from an enterprise CA (a CA that your organization makes accessible to you), or from a CA not bundled with your operating system. Consult the security team of your organization to find out which of the methods they require for obtaining certificates.

Important:

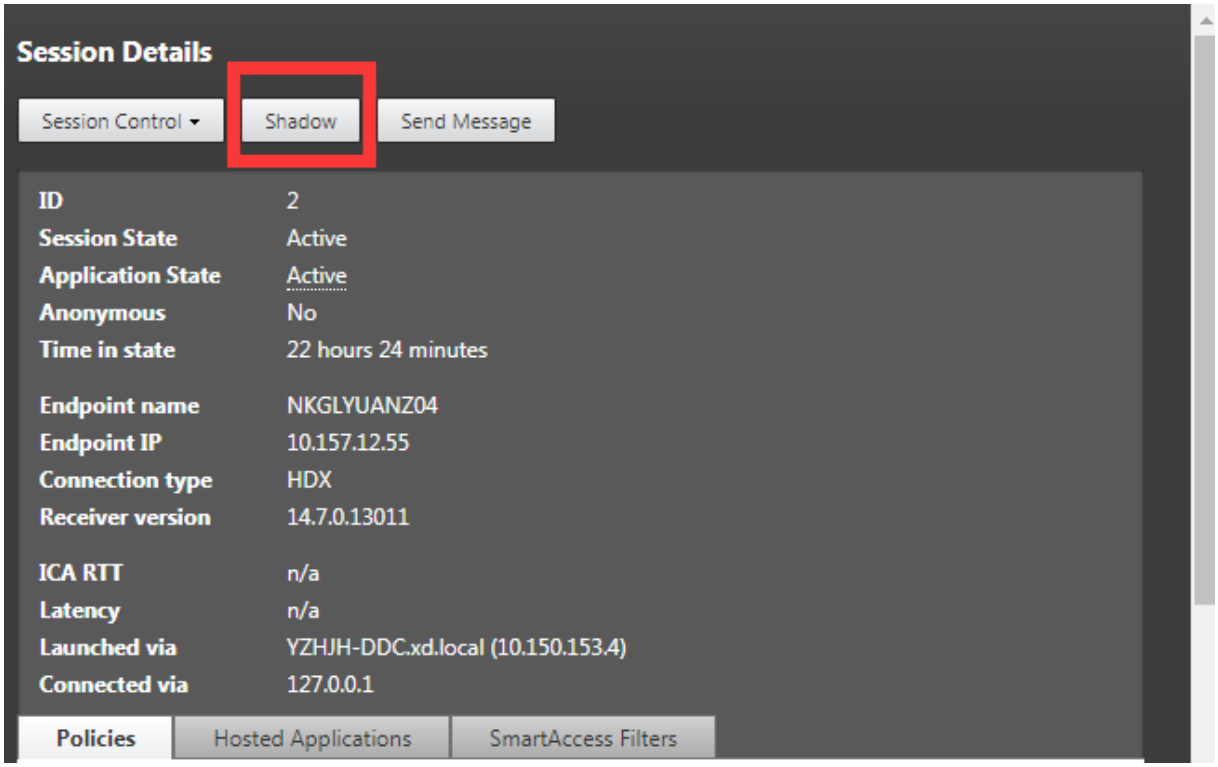
- The Common Name for a server certificate must be the exact FQDN of the Linux VDA server or at least the correct wildcard plus domain characters. For example, vda1.basedomain.com or *.basedomain.com.
- Hashing algorithms including the SHA1 and MD5 are too weak for signatures in digital certificates for some browsers to support. So SHA-256 is specified as the minimum standard.

Install a root certificate on each Citrix Director client Session shadowing uses the same registry-based certificate store as IIS, so you can install root certificates using IIS or the Microsoft Management Console (MMC) Certificates snap-in. When you receive a certificate from a CA, you can restart the Web Server Certificate Wizard in IIS and the wizard installs the certificate. Alternatively, you can view and import certificates on the computer using the MMC and add the certificate as a standalone snap-in. Internet Explorer and Google Chrome import the certificates installed on your operation system by default. For Mozilla Firefox, you must import your root SSL certificates on the **Authorities** tab of Certificate Manager.

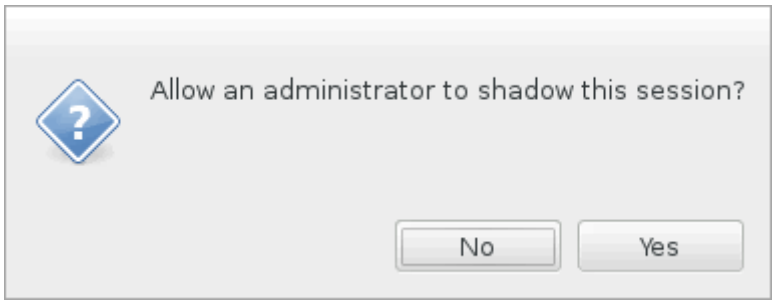
Install a server certificate and its key on each Linux VDA server Name the server certificates "shadowingcert.*"and the key file "shadowingkey.*"(* can indicate the format, for example, shadowingcert.csr and shadowingkey.key). Put server certificates and key files under the path **/etc/xdl/shadowingssl** and protect them properly with restricted permissions. An incorrect name or path makes the Linux VDA unable to find a specific certificate or key file and therefore causes connection failure with Citrix Director.

Usage

From Citrix Director, find the target session and click **Shadow** in the **Session Details** view to send a shadowing request to the Linux VDA.



After the connection initializes, a confirmation appears on the ICA session client (not the Citrix Director client) to ask the user for permission to shadow the session.



If the user clicks **Yes**, a window appears on the Citrix Director side, indicating that the ICA session is being shadowed.

For more information about the usage, see the Citrix Director Documentation.

Limitations

- Session shadowing is designed for use in an Intranet only. It does not work for external networks even connecting through Citrix Gateway. Citrix disclaims any responsibility for the Linux VDA session shadowing in an external network.
- With session shadowing enabled, a domain administrator can only view the ICA sessions, but has no permission to write or control it.

- After an administrator clicks **Shadow** from Citrix Director, a confirmation appears to ask the user for permission to shadow the session. A session can be shadowed only when the session user gives the permission.
- The previously mentioned confirmation has a timeout limitation, which is 20s. A shadowing request fails when the time runs out.
- One ICA session can be shadowed by only one administrator in one Citrix Director window. If an ICA session has been shadowed by administrator A and meanwhile, administrator B sends a shadowing request, the confirmation for getting the user permission reappears on the user device. If the user agrees, the shadowing connection for administrator A stops and a new shadowing connection is built for administrator B. It is the same if another shadowing request for the same ICA session is sent by the same administrator.
- To use session shadowing, install Citrix Director 7.16 or later.
- A Citrix Director client uses an FQDN rather than an IP address to connect to the target Linux VDA server. Therefore, the Citrix Director client must be able to resolve the FQDN of the Linux VDA server.

Troubleshooting

If session shadowing fails, perform debugging on both the Citrix Director client and the Linux VDA.

On the Citrix Director client

Through the developer tools of the browser, check the output logs on the **Console** tab. Or, check the response of the ShadowLinuxSession API on the **Network** tab. If the confirmation for getting the user permission appears but the connection fails to be built, ping the FQDN of the Linux VDA manually to verify that Citrix Director can resolve the FQDN. If there is an issue with the `wss://` connection, check your certificates.

On the Linux VDA

Verify that the confirmation for getting the user permission appears in response to a shadowing request. If it does not, check the vda.log and hdx.log files for clues. To obtain the vda.log file, do the following:

1. Find the /etc/xdl/ctx-vda.conf file. Uncomment the following line to enable the vda.log configuration:

   ```
   Log4jConfig="/etc/xdl/log4j.xml"
   ```

2. Open /etc/xdl/log4j.xml, locate the com.citrix.dmc part, and change “info”to “trace”as follows:


```
<!-- Broker Agent Plugin - Director VDA plugin Logger -->

 <logger name="com.citrix.dmc">

   <level value="trace"/>

 </logger>
<!--NeedCopy-->
```

3. Run the `service ctxvda restart` command to restart the `ctxvda` service.

If there is an error during connection build-up:

1. Check for any firewall limitation that stops session shadowing from opening the port.
2. Verify that certificates and key files are named properly and put under the correct path if it is the SSL scenario.
3. Verify that there are enough ports left between 6001-6099 for new shadowing requests.

Browser content redirection

December 21, 2021

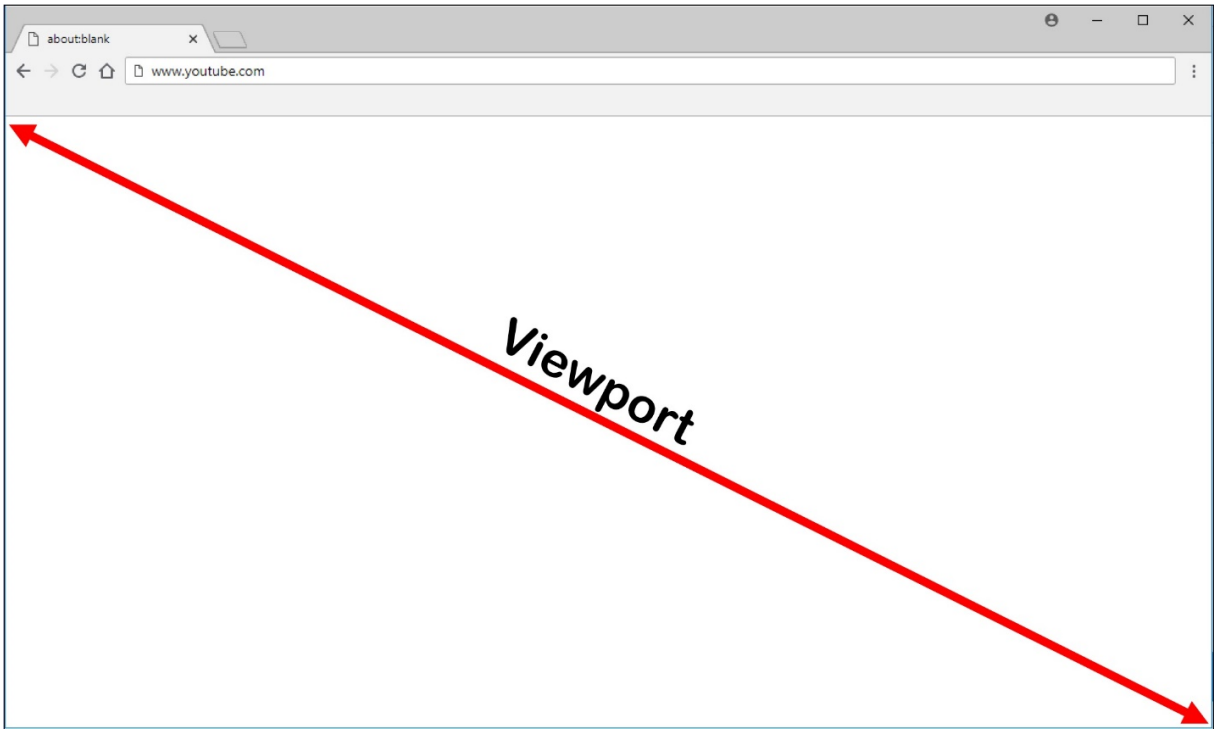
Overview

The Linux VDA supports browser content redirection in Google Chrome. Browser content redirection provides the ability of rendering webpages in the allow list on the client side. This feature uses Citrix Workspace app to instantiate a corresponding rendering engine on the client side, which fetches the HTTP and HTTPS content from the URL.

> **Note:**
>
> You can specify which webpages are redirected to the client side by using an allow list. Conversely, you can specify which webpages are not redirected to the client side by using a block list.

This overlay web layout engine runs on the client instead of on the VDA and uses the client CPU, GPU, RAM, and network.

Only the browser viewport is redirected. The viewport is the rectangular area in your browser where content displays. The viewport does not include items such as the address bar, favorites bar, and status bar. Those items are still running in the browser on the VDA.



System requirements

Windows client:

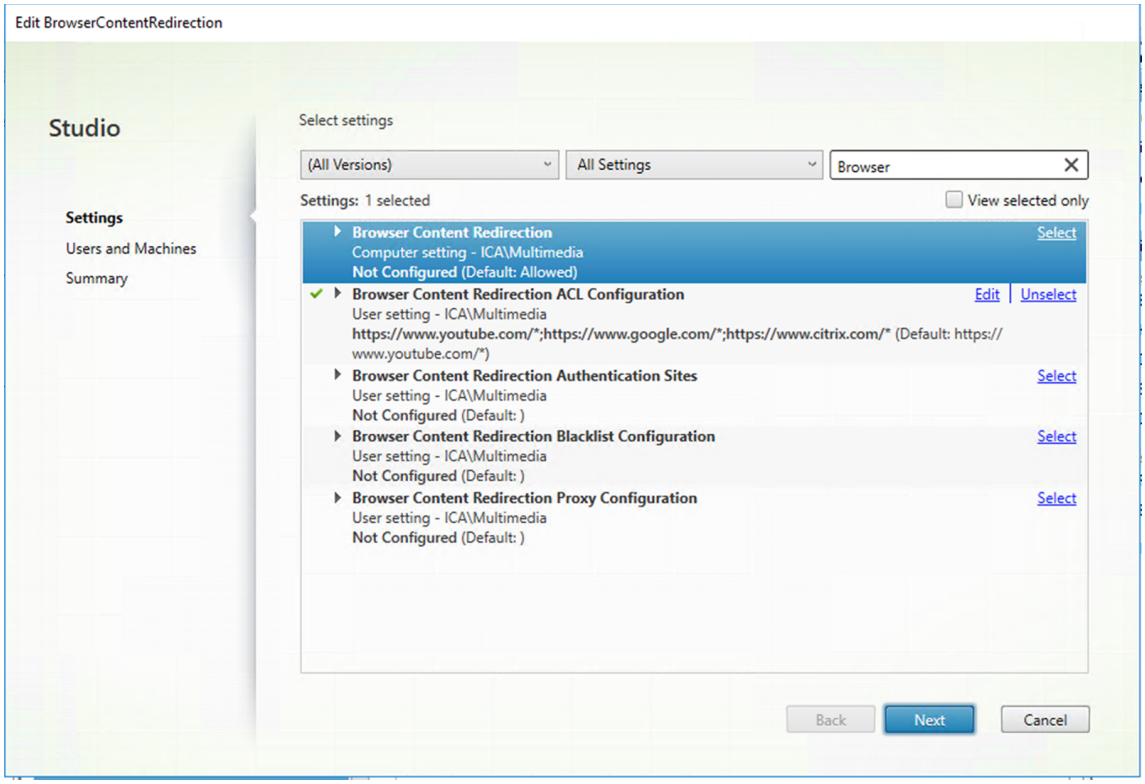
- Citrix Workspace app 1809 for Windows or later

Linux VDA:

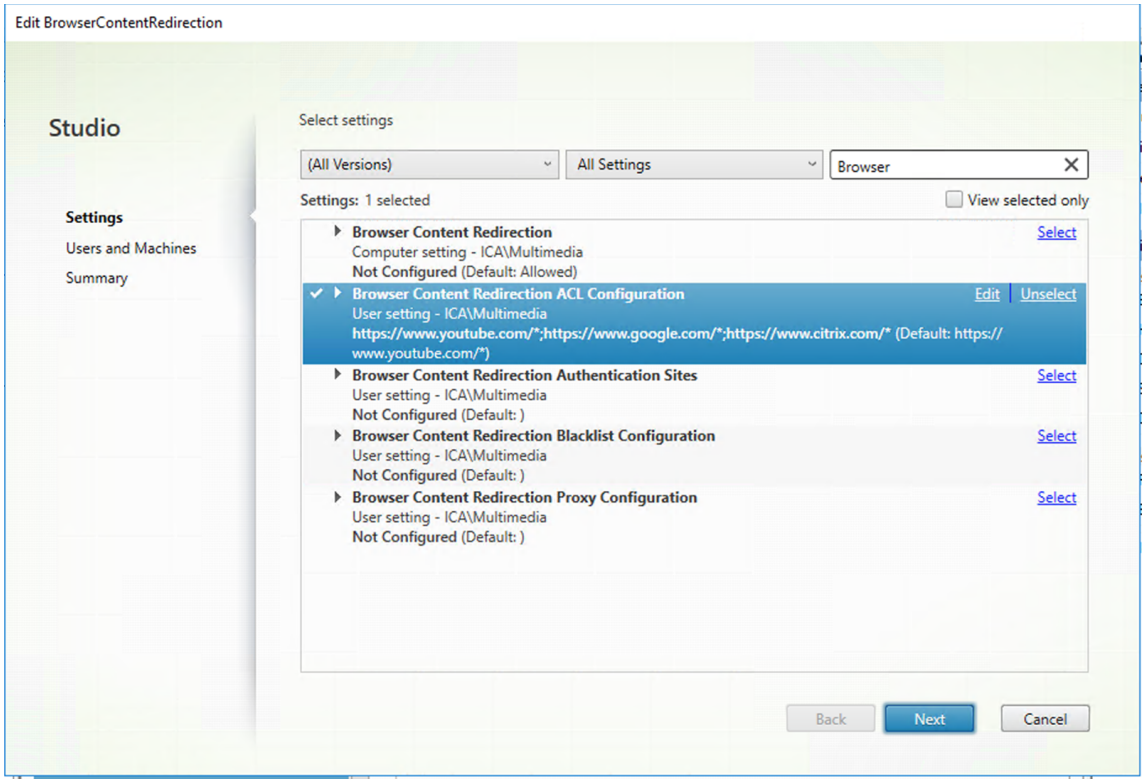
- VDA operating system: Ubuntu 16.04, Ubuntu 18.04, RHEL 7.8, RHEL 8.2, RHEL 8.1, SLES 12.5
- Browser on the VDA: Google Chrome v66 or later with the Citrix browser content redirection extension added

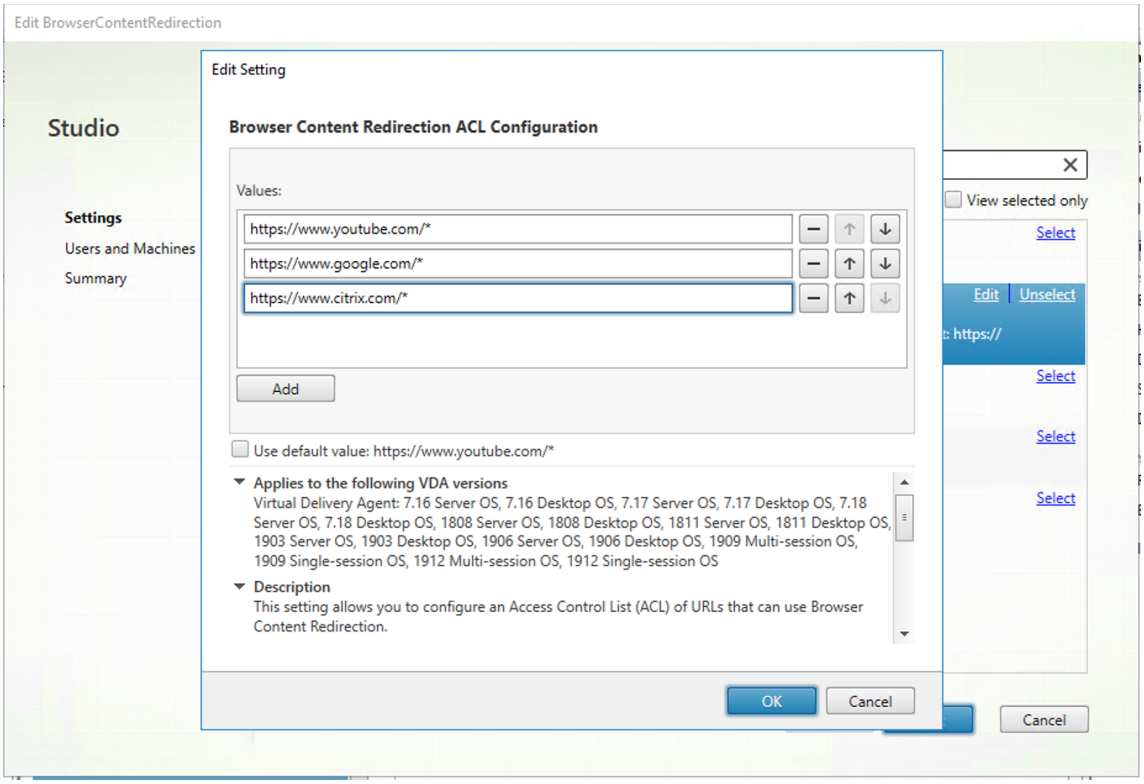
Configure browser content redirection

1. In Citrix Studio, configure a policy that specifies an allow list of URLs that can use browser content redirection and a block list of URLs that cannot. Browser content redirection is set to **Allowed** by default.

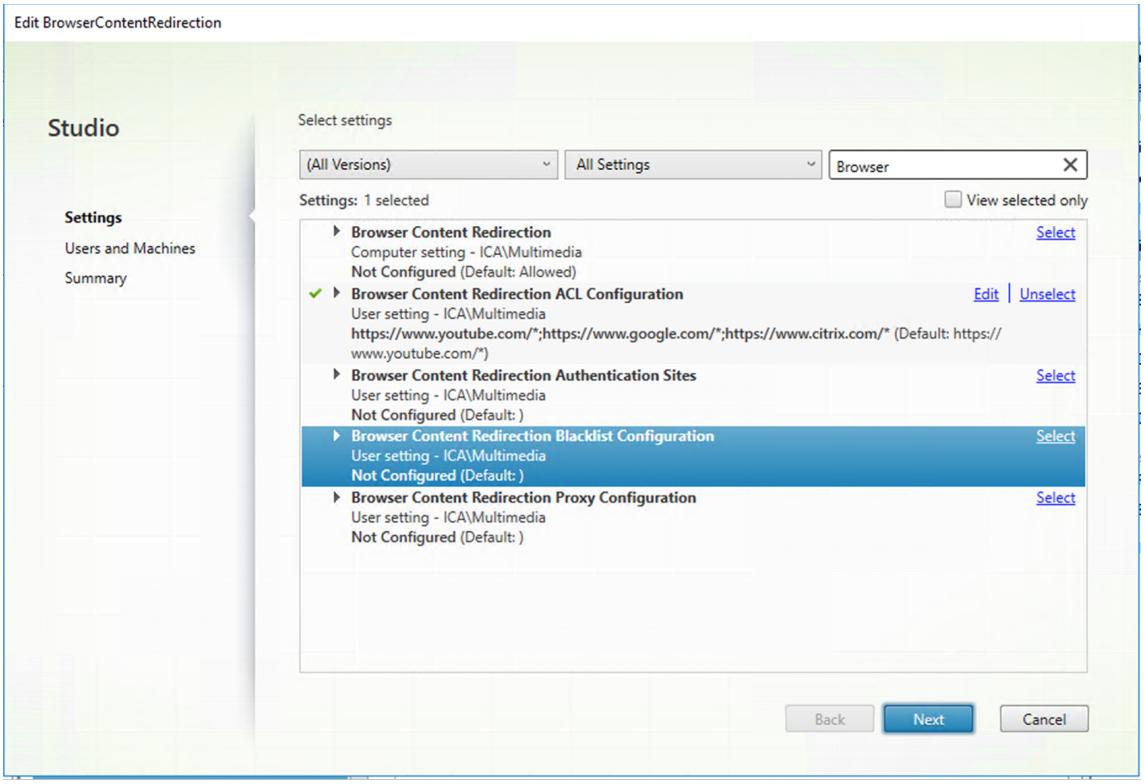


The **Browser Content Redirection ACL Configuration** setting specifies an allow list of URLs that can use browser content redirection.





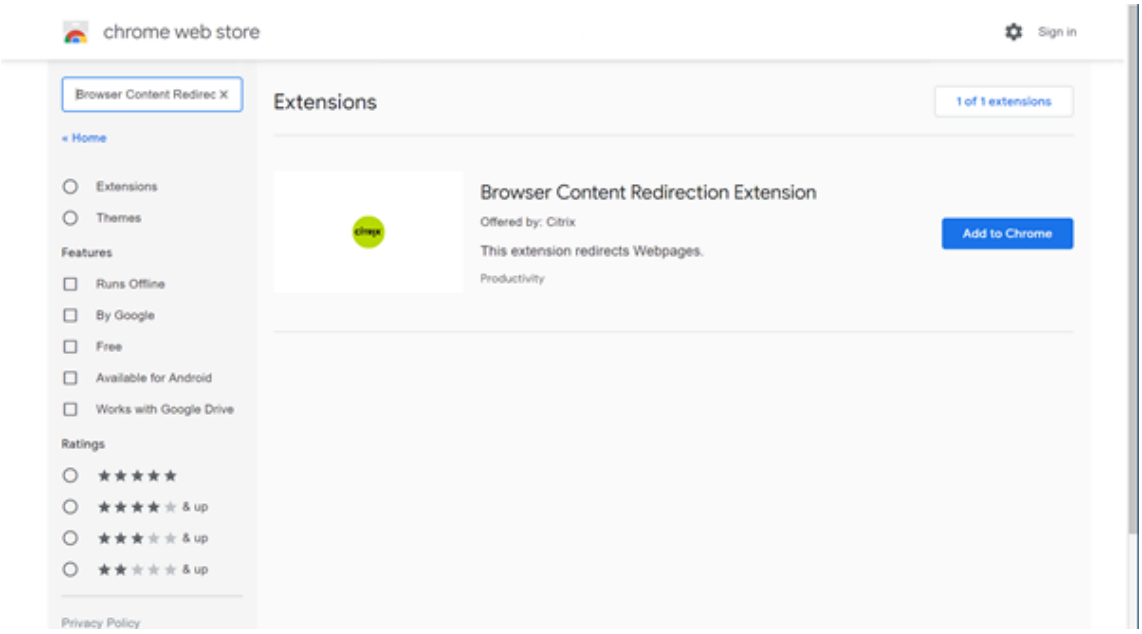
The **Browser Content Redirection Blacklist Configuration** setting specifies a block list of URLs that cannot use browser content redirection.



> **Note:**
>
> The Linux VDA currently does not support the **Browser Content Redirection Proxy Configuration** setting.

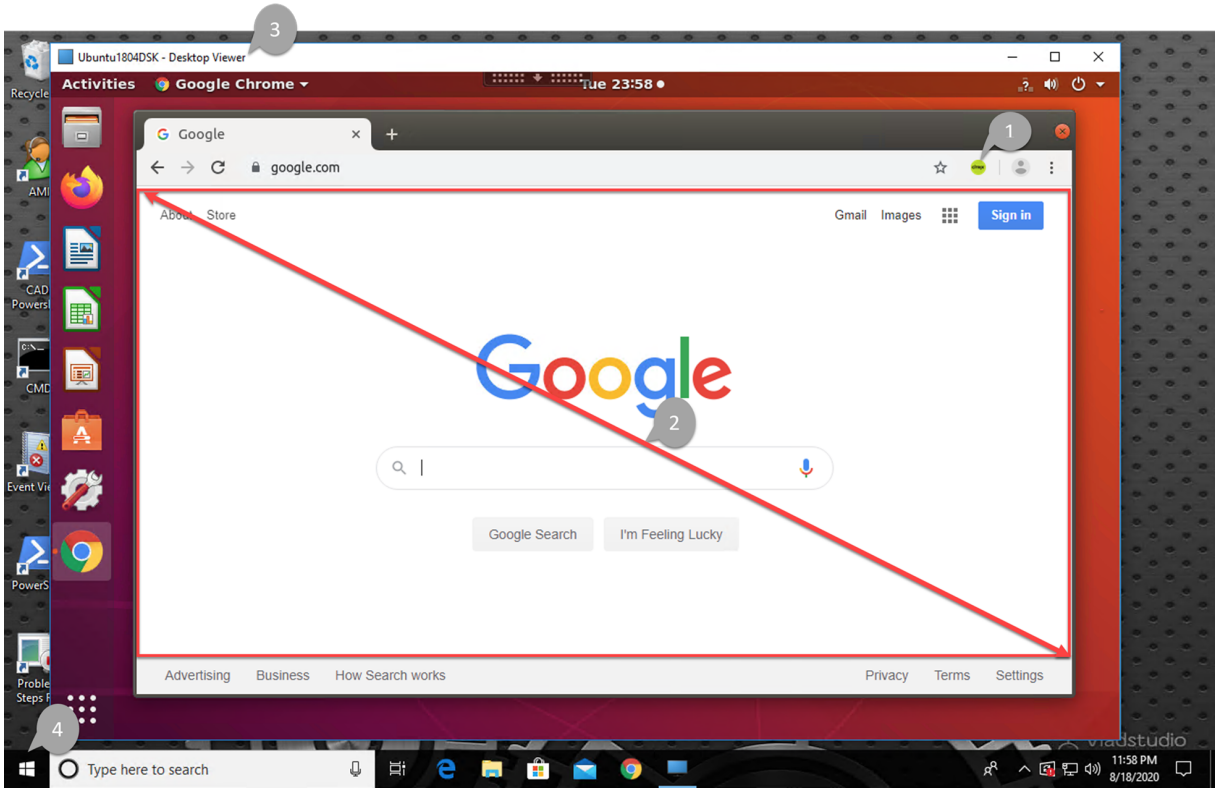
2. For the browser on the VDA to detect whether a URL (being navigated to) matches an allow list or a block list, add the Citrix browser content redirection extension from the Chrome Web Store. Click **Add to Chrome** on the VDA.

   > **Important:**
   >
   > The extension is not required on the client. Add it only on the VDA.
   >
   > Chrome extensions are installed on a per-user basis. Updating a golden image to add or remove an extension is not required.



If a match to a URL is found in an allow list (for example, `https://www.mycompany.com/`) but not in any block list, a virtual channel (CTXCSB) instructs the Citrix Workspace app that a redirection is required and relays the URL. Citrix Workspace app then instantiates a local rendering engine and displays the website.

Citrix Workspace app then blends back the website into the virtual desktop browser content area seamlessly.



1. Icon of the Citrix browser content redirection extension

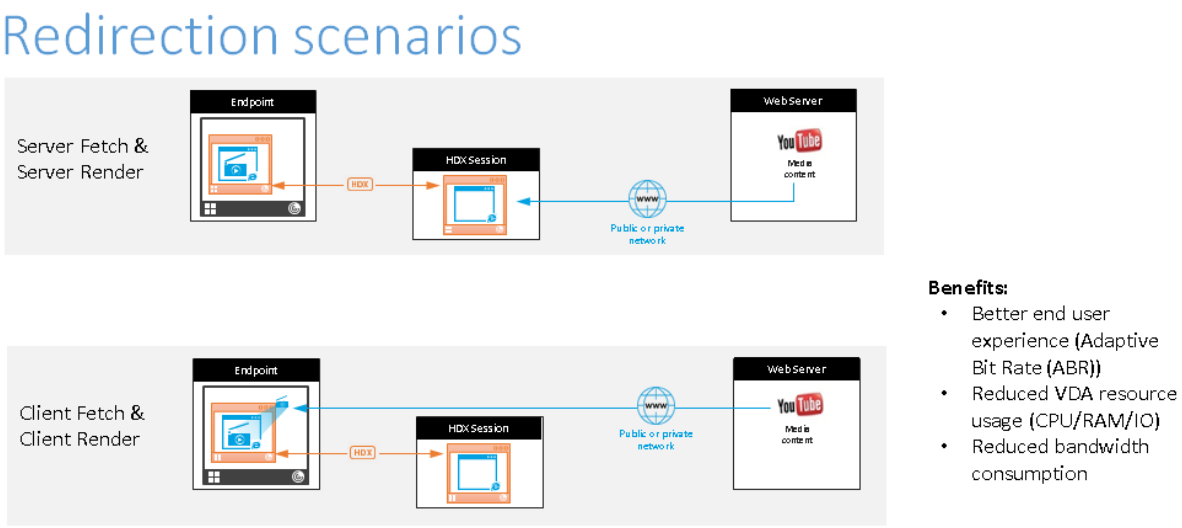
   The color of the extension icon specifies the status of the Chrome extension. It is one of the three colors:

   - Green: Active and connected
   - Gray: Not active/idle on the current tab
   - Red: Broken/Not working

2. Viewport rendered on the client or blended back to the virtual desktop
3. Linux VDA
4. Windows client

Redirection scenarios

Here are scenarios of how the Citrix Workspace app fetches content:



- **Server fetch and server render:** There is no redirection because you did not add the site to the allow list or the redirection failed. We fall back to rendering the webpage on the VDA and use Thinwire to remote the graphics. Use policies to control the fallback behavior. This scenario causes high CPU, RAM, and bandwidth consumption on the VDA.
- **Client fetch and client render:** Because the Citrix Workspace app contacts the web server directly, it requires Internet access. This scenario offloads all the network, CPU, and RAM usage from your Citrix Virtual Apps and Desktops site.

Fallback mechanism

There might be times when client redirection fails. For example, if the client machine does not have direct Internet access, an error response might go back to the VDA. In such cases, the browser on the VDA can then reload and render the page on the server.

Support Citrix Workspace app for HTML5

June 11, 2021

Starting with this release, you can use Citrix Workspace app for HTML5 to access Linux virtual apps and desktops directly without connecting your client to Citrix Gateway. For information about Citrix Workspace app for HTML5, see the Citrix documentation.

Enable this feature

This feature is disabled by default. To enable it, do the following:

1. In Citrix StoreFront, enable Citrix Workspace app for HTML5.

   For the detailed procedure, see Step 1 of Knowledge Center article CTX208163.

2. Enable WebSocket connections.

   a) In Citrix Studio, set the **WebSockets connections** policy to **Allowed**.

      You can also set the other WebSocket policies. For a full list of the WebSocket policies, see WebSockets policy settings.

   b) On the VDA, restart the `ctxvda` service and the `ctxhdx` service, in this order, for your setting to take effect.

   c) On the VDA, run the following command to check whether the WebSocket listener is running.

      ```
      netstat -an | grep 8008
      ```

      When the WebSocket listener is running, the command output is similar to the following:

      ```
      tcp 0 0 :::8008 :::* LISTEN
      ```

   Note: You can also enable TLS encryption to secure WebSocket connections. For information about enabling TLS encryption, see Secure user sessions using TLS.

Monitor Linux VMs and Linux sessions in Citrix Director

June 1, 2023

This article lists some of the metrics that are available for Linux VMs and Linux sessions in Citrix Director.

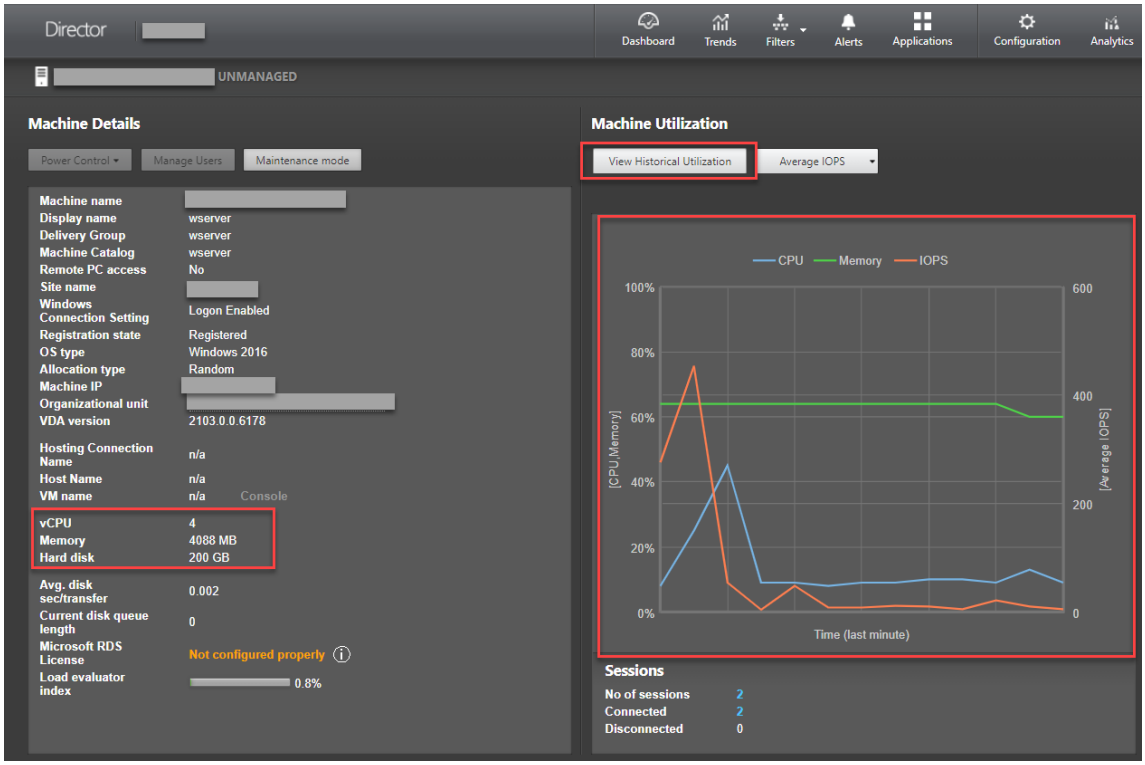
Metrics for Linux VMs

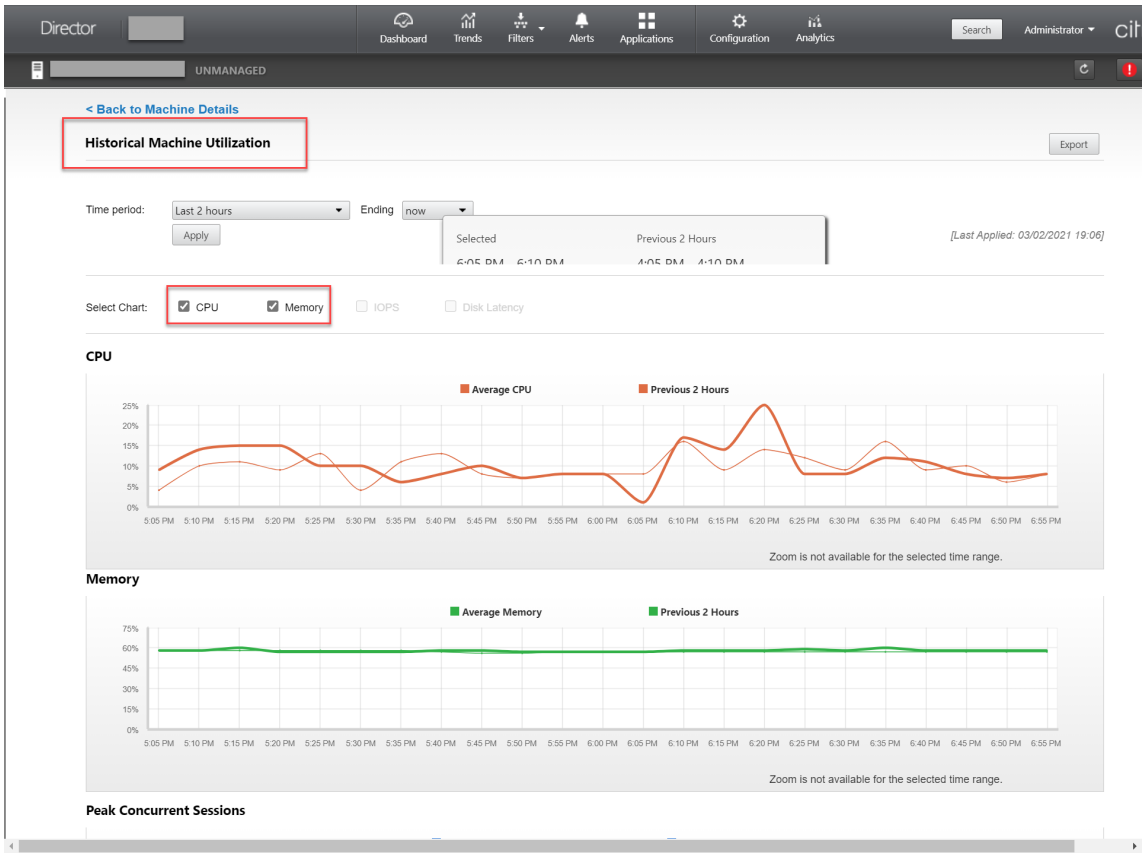
To access the metrics of a Linux VM, find the VM in Citrix Director and check the **Machine Details** panel.

Starting with Linux VDA Version 2103, the following metrics for Linux VMs are available in Citrix Director.

- The number of CPU cores
- Memory size
- Hard disk capacity

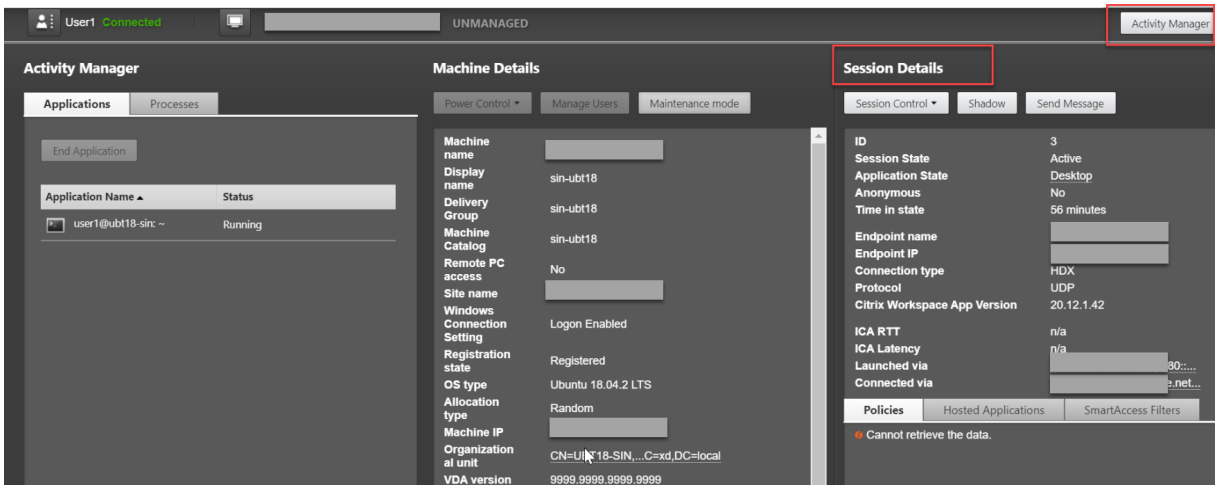
- Current and historical CPU and memory utilization





Metrics for Linux sessions

To view the metrics of a Linux session, open the **All Sessions** page by selecting **Filters > Sessions > All Sessions**, or access the **Session Details** panel. To access the **Session Details** panel, open the **All Sessions** page and click a target session to access its **Activity Manager** view. For example:



- ICA RTT

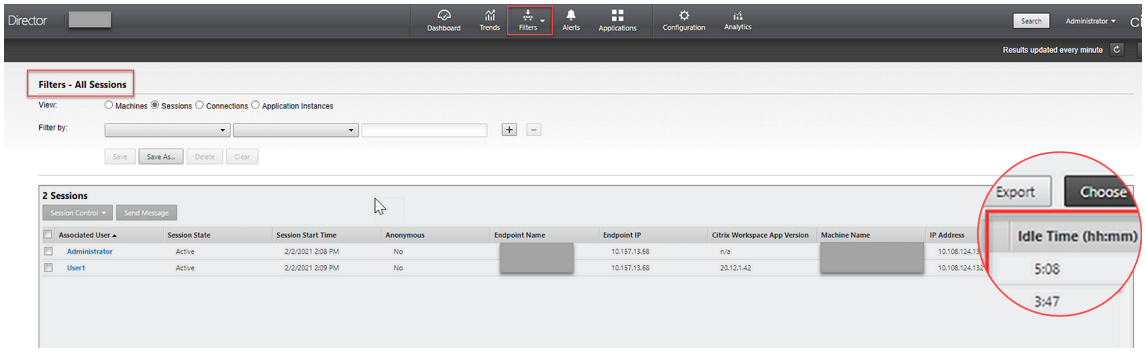
  Starting with Linux VDA Version 1903, ICA RTT metrics are available. To view ICA RTT metrics, use Citrix Director 1903 or later and create the **ICA round trip calculation** and **ICA round trip calculation interval** policies in Citrix Studio. For information about policy creation, see Create a policy using Studio.

- Protocol

  Starting with Linux VDA Version 1909, protocol information is available. The transport protocol of a Linux session appears as **UDP** or **TCP** in the **Session Details** panel.

- Idle time

  Starting with Linux VDA Version 2103, the idle time metric is available for Linux sessions. To access this metric, open the **All Sessions** page by selecting **Filters > Sessions > All Sessions**.



Monitor service daemon

June 11, 2021

The monitor service daemon monitors key services by performing periodical scans. When detecting exceptions, the daemon restarts or stops service processes and cleans up process residuals for releasing resources. The detected exceptions are recorded in the **/var/log/xdl/ms.log** file.

Configuration

The monitor service daemon starts automatically when you start the VDA.

You can configure the feature through the **scanningpolicy.conf**, **rulesets.conf**, and **whitelist.conf** files with administrator privileges. The configuration files are located at **/opt/Citrix/VDA/sbin**.

To make your changes in the **scanningpolicy.conf**, **rulesets.conf**, and **whitelist.conf** files take effect, run the following command to restart the monitor service daemon.

```
service ctxmonitorservice restart
<!--NeedCopy-->
```

- **scanningpolicy.conf**

  This configuration file enables or disables the monitor service daemon. It sets the service detection interval and specifies whether to repair detected exceptions.

  - MonitorEnable: true/false (true by default)
  - DetectTime: 20 (unit: seconds, default value: 20, minimum value: 5)
  - AutoRepair: true/false (true by default)
  - MultBalance: false
  - ReportAlarm: false

- **rulesets.conf**

  This configuration file specifies the target services to monitor. There are four monitored services by default as shown in the following screen capture.


```
MonitorUser: all
MonitorType: 3
ProcessName: ctxhdx
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: ctxvda
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: ctxpolicyd
Operation: 4
DBRecord: false

MonitorUser: all
MonitorType: 3
ProcessName: Xorg
Operation: 8
DBRecord: false
```

To configure each service to monitor, set the following fields.

- MonitorUser: all
- MonitorType: 3
- ProcessName: <> (The process name cannot be left blank and must be exactly matched.)
- Operation: 1/2/4/8 (1 = stop the service when exceptions are detected. 2 = kill the service when exceptions are detected. 4 = restart the service. 8 = clean up the Xorg process residuals.)
- DBRecord: false

- **whitelist.conf**

  The target services specified in the **rulesets.conf** file must also be configured in **whitelist.conf** file. The white list configuration is a secondary filter for security.

  To configure the white list, include only the process names (which must be match exactly) in the **whitelist.conf** file. For an example, see the following screen capture.


```
ctxcdmd
ctxcdmmount
ctxcdmstat
ctxceip
ctxclipboard
ctxconnect
ctxcredentialctl
ctxctl
ctxcupsd
ctxdisconnect
ctxeuem
ctxfiletransfer
ctxgfx
ctxhdx
ctxism
ctxlogd
ctxlogin
ctxmonitorservice
ctxmrvc
ctxpolicyd
ctxscardsd
ctxvhcid
ctxvda
Xorg
```

> **Note:**
>
> Before you stop the `ctxvda`, `ctxhdx`, and `ctxpolicyd` services, run the `service ctxmonitorservice stop` command to stop the monitor service daemon. Otherwise, the monitor service daemon restarts the services you stopped.

Secure user sessions using TLS

February 18, 2022

As of Version 7.16, the Linux VDA supports TLS encryption for secure user sessions. TLS encryption is disabled by default.

Enable TLS encryption

To enable TLS encryption for secure user sessions, obtain certificates and enable TLS encryption on both the Linux VDA and the Delivery Controller (the Controller).

Obtain certificates

Obtain server certificates in PEM format and root certificates in CRT format from a trusted Certificate Authority (CA). A server certificate contains the following sections:

- Certificate
- Unencrypted private key
- Intermediate certificates (optional)

An example of a server certificate:


```
-----BEGIN CERTIFICATE-----
MIIDTTCCAragAwIBAgIJALluncpiqGXCMA0GCSqGSIb3DQEBBQUAMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
MB4XDTA4MDkzMDEwNTk1M1oXDTI4MDkyNTEwNTk1M1owgYoxCzAJBgNVBAYTAlVL
MRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEUMBIGA1UE
ChMLQ2l0cml4IFRlc3QxGzAZBgNVBAsTElNlcnZlciBDZXJ0aWZpY2F0ZTEgMB4G
A1UEAxMXY2EwMDEtc2MwMDEuY2l0cml0ZS5uZXQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALCTTOdxcivbI0LOF66xgO5gkNeIGKVP+37pSKV8B66lWCVzr6p9
t72Fa+9oCcf2x/ue274NXFcg4fqGRDsrEwl3YxM6COyBf7L6psrsCDNnBPlq8TJH
4xoPIXUeaW4MVk/3PVyfhHKs4fz8yy1I4VDnXVHhw+OFQ2Bq3NhwsRhnAgMBAAGj
gdwwgdkwCQYDVR0TBAIwADAdBgNVHQ4EFgQUrLidzYot+CUXSh9xMfp1M+/O8yOw
gZkGA1UdIwSBkTCBjoAU85kN1EPJOcVhcOss1slseDQwGsKha6RpMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
ggkAy8nC8dcB32EwEQYJYIZIAYb4QgEBBAQDAgVgMA0GCSqGSIb3DQEBBQUAA4GB
AD5axBYHwIxJCJzNt2zdXnbp2O0yUToWElBwQe/9cGaP6CpjoxJ7FJa2/8IpaT68
Vel8u1SEYY1GKCGCw93pc7sPKqb8pGBRI5/dygb+geFk1Q7KyVbuOIjOtr3pkxAe
b6CFJtNLudHUrwF6lOrB72zbyz3PiIx+HEwt1jOj8z4K
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCwk0zncXIr2yNC9BeusYDuYJDXiBilT/t+6UilfAeupVglc6+q
fbe9hWvvaAnH9sf7ntu+DVxXIOH6hkQ7KxMNd2MTOgjsgX+y+qbK7AgzZwT9avEy
R+MaDyF1HmluDFZP9z1cn4RyrOH8/MstSOFQ511R4cPtBUNgatzYcLEYZwIDAQAB
AoGBAKwBgZu/bkl8edgB8YPyU7diiBX89IOs4b/aPjM+JDmjxb8N96RsPO24p9Ea
FtUc9+iL8mEroLUbSicCXjsJFc+cxg9vVaNa6EEkkBj735oCUERqSxOYb/lAdck/
FXzU0tqytUe/KHgcSgjtjrSeqLJqMm+yxzBAatVRTTzGdwAhAkEA3l1kRZjIN5uz
Enmi2RTI3ngBhBP/S3GEbvJfKsD5n2Ri9O+OoEPxclvvp5ne8QOzUpshbjFEPbOC
ykZ6UassFwJBAMtI5yPnV9ewPzJoaNjZIJcMtNXDchSlxXiJiyzv+Qmr8RuQz9Pv
fIenmTrfZ+Wo4DaKg+8ar2OvOnKFOHFAmDECQQDEWRlH6cE3WyCfN1u942M9XkhR
GvSpR7+b///vL6Nwwv3CwPV9n8DTpL+wuDkJZ9nCvRteil9MlaMTYjs3alNvAkEA
qy5JzZcbBnrYzMbVO32jju7ZPISnhTGO1xDjzMSLLpTGpNLN34b0k3sTclr8L42E
uQjtTqRm+wdsrVF3lFazkQJANudmsUVv3gZKhMGaV2hzIdXIfHyOIYv+3leZhQY6
h5eEmxSZS5OTvyNGt2e6m2ZgaZmjTagH59TCBHvR5nof2g==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIDGTCCAoKgAwIBAgIJAMvJwvHXAd9hMA0GCSqGSIb3DQEBBQUAMGcxCzAJBgNV
BAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNVBAcTCUNhbWJvdXJuZTEU
MBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNhMDAxLmNpdHJpdGUubmV0
MB4XDTA4MDkzMDEwNDExMVoXDTI4MDkyNTEwNDExMVowZzELMAkGA1UEBhMCVUsx
EjAQBgNVBAgTCUNhbWJyaWRnZTESMBAGA1UEBxMJQ2FtYm9lcm5lMRQwEgYDVQQK
EwtDaXRyaXggVGVzdDEaMBgGA1UEAxMRY2EwMDEuY2l0cml0ZS5uZXQwgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAKVZmF7Uj7uOnvO3Qwdfi0nr3QkNH2DXpWrZ
Zh8cI9Vv+UFRUiC6oB7izLtBMFn3fOUP7i2CfkHN3ZGJ17p89pdyjket1MslVeJw
acOqrYvD+fNNSvJjunTbaCywVtALjmFSfMHeZJXVSckrpEhnk0nkMS16tcrya/K/
osSlzvI3AgMBAAGjgcwwgckwDAYDVR0TBAUwAwEB/zAdBgNVHQ4EFgQU85kN1EPJ
OcVhcOss1slseDQwGsIwgZkGA1UdIwSBkTCBjoAU85kN1EPJOcVhcOss1slseDQw
GsKha6RpMGcxCzAJBgNVBAYTAlVLMRIwEAYDVQQIEwlDYW1icmlkZ2UxEjAQBgNV
BAcTCUNhbWJvdXJuZTEUMBIGA1UEChMLQ2l0cml4IFRlc3QxGjAYBgNVBAMTEWNh
MDAxLmNpdHJpdGUubmV0ggkAy8nC8dcB32EwDQYJKoZIhvcNAQEFBQADgYEAIZ4Z
gXLLXf12RNqh/awtSbd41Ugv8BIKAsg5zhNAiTiXbzz8Cl3ec53Fb6nigMWc5Tli
UNCLXwnxRUiD4O0tESLX9ACUNH3I94yxOgujkSOSBni21jjZTvfBB32Rmr5DByJg
UmKORn/hdqMlcqpe5wO6as6+HN4WUOi+hEtUMME=
-----END CERTIFICATE-----
```


Enable TLS encryption

Enable TLS encryption on the Linux VDA On the Linux VDA, use the `enable_vdassl.sh` tool to enable (or disable) TLS encryption. The tool is located in the **/opt/Citrix/VDA/sbin** directory. For information about the options available in the tool, run the `/opt/Citrix/VDA/sbin/enable_vdassl.sh -help` command.

Tip: A server certificate must be installed on each Linux VDA server and root certificates must be installed on each Linux VDA server and client.

Enable TLS encryption on the Controller

> **Note:**
>
> You can enable TLS encryption only for entire delivery groups. You cannot enable TLS encryption for specific applications.

In a PowerShell window on the Controller, run the following commands in sequence to enable TLS encryption for the target delivery group.

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME'| Set-BrokerAccessPolicyRule –HdxSslEnabled $true`

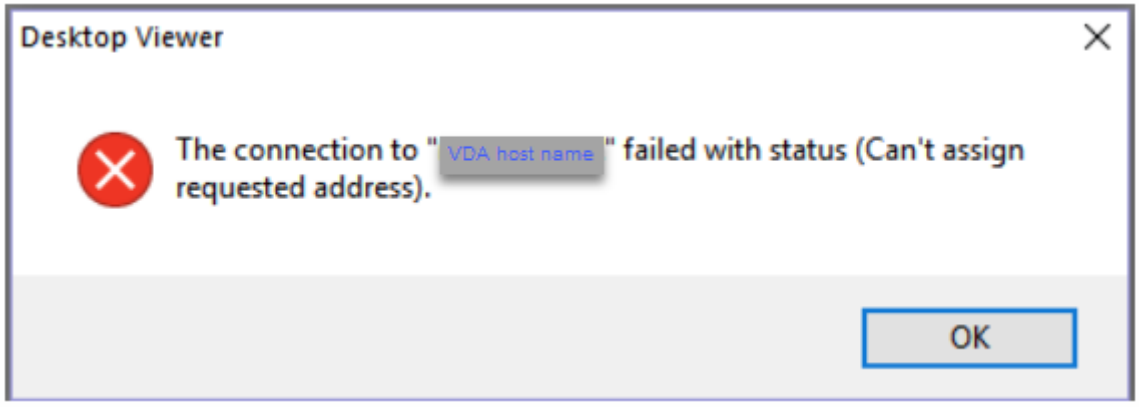
> **Note:**
>
> To ensure that only VDA FQDNs are contained in an ICA session file, you can also run the `Set-BrokerSite –DnsResolutionEnabled $true` command. The command enables DNS resolution. If you disable DNS resolution, an ICA session file discloses VDA IP addresses and provides FQDNs only for the TLS-related items such as SSLProxyHost and UDPDTLSPort.

To disable TLS encryption on the Controller, run the following commands in sequence:

1. `Add-PSSnapin citrix.*`
2. `Get-BrokerAccessPolicyRule –DesktopGroupName 'GROUPNAME'| Set-BrokerAccessPolicyRule –HdxSslEnabled $false`
3. `Set-BrokerSite –DnsResolutionEnabled $false`

Troubleshooting

The following "Can't assign requested address" error might occur in Citrix Workspace app for Windows when you try to access a published desktop session:



As a workaround, add an entry to the **hosts** file, which is similar to:

```
<IP address of the Linux VDA>      <FQDN of the Linux VDA>
```

On Windows machines, the **hosts** file typically locates at `C:\Windows\System32\drivers\etc\hosts`.

Secure user sessions using DTLS

June 11, 2021

DTLS encryption is a fully supported feature starting with the 7.18 release. By default, this feature is enabled on the Linux VDA. For more information, see Transport Layer Security.

Enable DTLS encryption

Verify that adaptive transport is enabled

In Citrix Studio, verify that the **HDX Adaptive Transport** policy is set to **Preferred** or **Diagnostic mode**.

Enable SSL encryption on the Linux VDA

On the Linux VDA, use the `enable_vdassl.sh` tool to enable (or disable) SSL encryption. The tool is located at **/opt/Citrix/VDA/sbin**. For information about the options available in the tool, run the `/opt/Citrix/VDA/sbin/enable_vdassl.sh –h` command.

> **Note:**
>
> Currently, the Linux VDA supports both DTLS 1.0 and DTLS 1.2. DTLS 1.2 requires Citrix Receiver

> for Windows 4.12, or Citrix Workspace app 1808 for Windows or later. If your client supports only DTLS 1.0 (for example, Citrix Receiver for Windows 4.11), set `SSLMinVersion` to `TLS_1.0` and `SSLCipherSuite` to `COM` or `ALL` using the `enable_vdassl.sh` tool.

Text-based session watermark

June 11, 2021

Text-based session watermarks help to deter and enable tracking data theft. This traceable information appears on the session desktop as a deterrent to those using photographs and screen captures to steal data. You can specify a watermark that is a layer of text, which displays over the entire session screen without changing the content of the original document.

> **Important:**
>
> Text-based session watermarking is not a security feature. The solution does not prevent data theft completely, but it provides some level of deterrent and traceability. We do not guarantee complete information traceability when using this feature. However, we recommend that you combine this feature with other security solutions as applicable.

The session watermark is text and is applied to the session that is delivered to the user. The session watermark carries information for tracking data theft. The most important data is the identity of the logon user of the current session in which the screen image was taken. To trace the data leakage more effectively, include other information such as server or client internet protocol address and a connect time.

To adjust the user experience, use the Session Watermark policy settings to configure the placement and watermark appearance on the screen.

Limitations

- Session watermarks are not supported in sessions where browser content redirection is used. To use the session watermark feature, ensure that browser content redirection is disabled.
- Session watermark is not supported and does not appear if the session is running in full-screen hardware accelerated H.264 or H.265 encoding mode with legacy NVIDIA drivers (in this case, NvCaptureType is set to 2 in the registry).
- Watermarks are not visible for session shadowing.
- If you press the Print Screen key to capture a screen, the screen captured at the VDA side does not include the watermarks. We recommend that you take measures to avoid screen captures being copied.

Pass-through authentication by using smart cards

June 11, 2021

Users can use a smart card connected to the client device for authentication when logging on to a Linux virtual desktop session. This feature is implemented through smart card redirection over the ICA smart card virtual channel. Users can also use the smart card within the session. Use cases include adding a digital signature to a document, encrypting or decrypting an email, and authenticating to a website that requires smart card authentication.

The Linux VDA uses the same configuration as the Windows VDA for this feature. For more information, see the Configure the smart card environment section in this article.

> **Note:**
>
> Using a mapped smart card within a Linux VDA session to sign on to Citrix Gateway is not officially supported.

Prerequisites

The availability of smart card pass-through authentication is contingent on the following conditions:

- Your Linux VDA is installed on one of the following distributions:
  - RHEL 8/CentOS 8
  - RHEL 7/CentOS 7
  - Ubuntu 20.04
  - Ubuntu 18.04
  - Ubuntu 16.04

  After the VDA installation completes, verify that your VDA can register with the Delivery Controller and the published Linux desktop sessions can be launched successfully using Windows credentials.
- Smart cards supported by OpenSC are used. For more information, see Ensure that OpenSC supports your smart card.
- Citrix Workspace app for Windows is used.

Ensure that OpenSC supports your smart card

OpenSC is a widely used smart card driver on RHEL 7.4+. As a fully compatible replacement of CoolKey□OpenSC supports many types of smart cards (see Smart Card Support in Red Hat

Enterprise Linux).

In this article, the YubiKey 4 smart card is used as an example to illustrate the configuration. YubiKey 4 is an all-in-one USB CCID PIV device that can easily be purchased from Amazon or other retail vendors. The OpenSC driver supports YubiKey 4.

If your organization requires some other more advanced smart card, prepare a physical machine with RHEL 7 or RHEL 8 and the OpenSC package installed. For information about the OpenSC installation, see Install the smart card driver. Insert your smart card, and run the following command to verify that OpenSC supports your smart card:

```
pkcs11-tool --module opensc-pkcs11.so --list-slots
```

Configuration

Prepare a root certificate

A root certificate is used to verify the certificate on the smart card. Do the following to download and install a root certificate.

1. Obtain a root certificate in PEM format, typically from your CA server.

   You can run a command similar to the following to convert a DER file (*.crt, *.cer, *.der) to PEM. In the following command example, **certnew.cer** is a DER file.

   ```
   openssl x509 -inform der -in certnew.cer -out certnew.pem
   <!--NeedCopy-->
   ```

2. Install the root certificate to the `openssl` directory. The **certnew.pem** file is used as an example.


```
cp certnew.pem <path where you install the root certificate>
<!--NeedCopy-->
```

To create a path for installing the root certificate, run `sudo mdkir -p <path where you install the root certificate>`.

Build the pam_krb5 module on RHEL 8/CentOS 8

Smart card authentication depends on the pam_krb5 module, which is deprecated on RHEL 8/CentOS 8. To use smart card authentication on RHEL 8/CentOS 8, build the pam_krb5 module as follows:

1. Download the pam_krb5-2.4.8-6 source code from https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html.
2. Build and install the pam_krb5 module on RHEL 8/CentOS 8.

```
yum install -y opensc pcsc-lite pcsc-lite-libs pcsc-lite-ccid nss-
    tools
yum install gcc krb5-devel pam-devel autoconf libtool
rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio  – div
tar xvzf pam_krb5-2.4.8.tar.gz
cd pam_krb5-2.4.8
./configure --prefix=/usr
make
make install
<!--NeedCopy-->
```

3. Verify that pam_krb5.so exists under /usr/lib64/security/.

```
ls -l /usr/lib64/security | grep pam_krb5
<!--NeedCopy-->
```

Configure the smart card environment

You can use the ctxsmartlogon.sh script to configure the smart card environment or complete the configuration manually.

(Option 1) Use the ctxsmartlogon.sh script to configure the smart card environment

> **Note:**
>
> The ctxsmartlogon.sh script adds PKINIT information to the default realm. You can change this setting through the **/etc/krb5.conf** configuration file.

Before using smart cards for the first time, run the ctxsmartlogon.sh script to configure the smart card environment.


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

The results resemble the following:

```
#*****************************************************************************
# 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.
```

To disable smart cards:

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

The results resemble the following:

```
#*****************************************************************************
# 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.
```

(Option 2) configure the smart card environment manually The Linux VDA uses the same smart card environment with the Windows VDA. In the environment, multiple components must be configured, including the Domain Controller, Microsoft Certificate Authority (CA), Internet Information Services, Citrix StoreFront, and Citrix Workspace app. For information about the configuration based on the YubiKey 4 smart card, see Knowledge Center article CTX206156.

Before proceeding to the next step, ensure that all components are correctly configured, the private key and user certificate are downloaded to the smart card, and you can successfully log on to the Windows VDA using the smart card.

Install the PC/SC Lite packages PCSC Lite is an implementation of the Personal Computer/Smart Card (PC/SC) specification in Linux. It provides a Windows smart card interface for communicating to smart cards and readers. Smart card redirection in the Linux VDA is implemented on the PC/SC level.

Run the following command to install the PC/SC Lite packages.

```
yum install pcsc-lite pcsc-lite-ccid pcsc-lite-libs
<!--NeedCopy-->
```

Install the smart card driver OpenSC is a widely used smart card driver on RHEL. If OpenSC is not installed, run the following command to install it.

```
yum install opensc
<!--NeedCopy-->
```

Install the PAM modules for smart card authentication Run the following command to install the pam_krb5 and krb5-pkinit modules.

RHEL 7/CentOS 7:

```
yum install pam_krb5 krb5-pkinit
<!--NeedCopy-->
```

RHEL 8/CentOS 8:

```
yum install krb5-pkinit
<!--NeedCopy-->
```

Ubuntu 20.04, Ubuntu 18.04, Ubuntu 16.04:

```
apt-get install libpam-krb5 krb5-pkinit
<!--NeedCopy-->
```

The pam_krb5 module is a pluggable authentication module that PAM-aware applications can use to check passwords and obtain ticket-granting tickets from the Key Distribution Center (KDC). The krb5-pkinit module contains the PKINIT plug-in that allows clients to obtain initial credentials from the KDC using a private key and a certificate.

Configure the pam_krb5 module The pam_krb5 module interacts with the KDC to get Kerberos tickets using certificates in the smart card. To enable pam_krb5 authentication in PAM, run the following command:

```
authconfig --enablekrb5 --update
<!--NeedCopy-->
```

In the **/etc/krb5.conf** configuration file, add PKINIT information according to the actual realm.

> **Note:**
>
> The **pkinit_cert_match** option specifies matching rules that the client certificate must match before it is used to attempt PKINIT authentication. The syntax of the matching rules is:
>
> *[relation-operator] component-rule ...*
>
> where `relation-operator` can be either &&, meaning all component rules must match, or ||, meaning only one component rule must match.

Here is an example of a generic krb5.conf file:

```
EXAMPLE.COM = {


    kdc = KDC.EXAMPLE.COM

    auth_to_local = RULE:[1:$1@$0]

    pkinit_anchors = FILE:<path where you install the root certificate
       >/certnew.pem

    pkinit_kdc_hostname = KDC.EXAMPLE.COM

    pkinit_cert_match = ||<EKU>msScLogin,<KU>digitalSignature

    pkinit_eku_checking = kpServerAuth

  }

<!--NeedCopy-->
```

The configuration file resembles the following after you add the PKINIT information.


```
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
}
```

Configure PAM authentication PAM configuration files tell what modules are used for PAM authentication. To add pam_krb5 as an authentication module, add the following line to the **/etc/pam.d/smartcard-auth** file:

```
auth [success=done ignore=ignore default=die] pam_krb5.so preauth_options
=X509_user_identity=PKCS11:<path to the pkcs11 driver>/opensc-pkcs11.
so
```

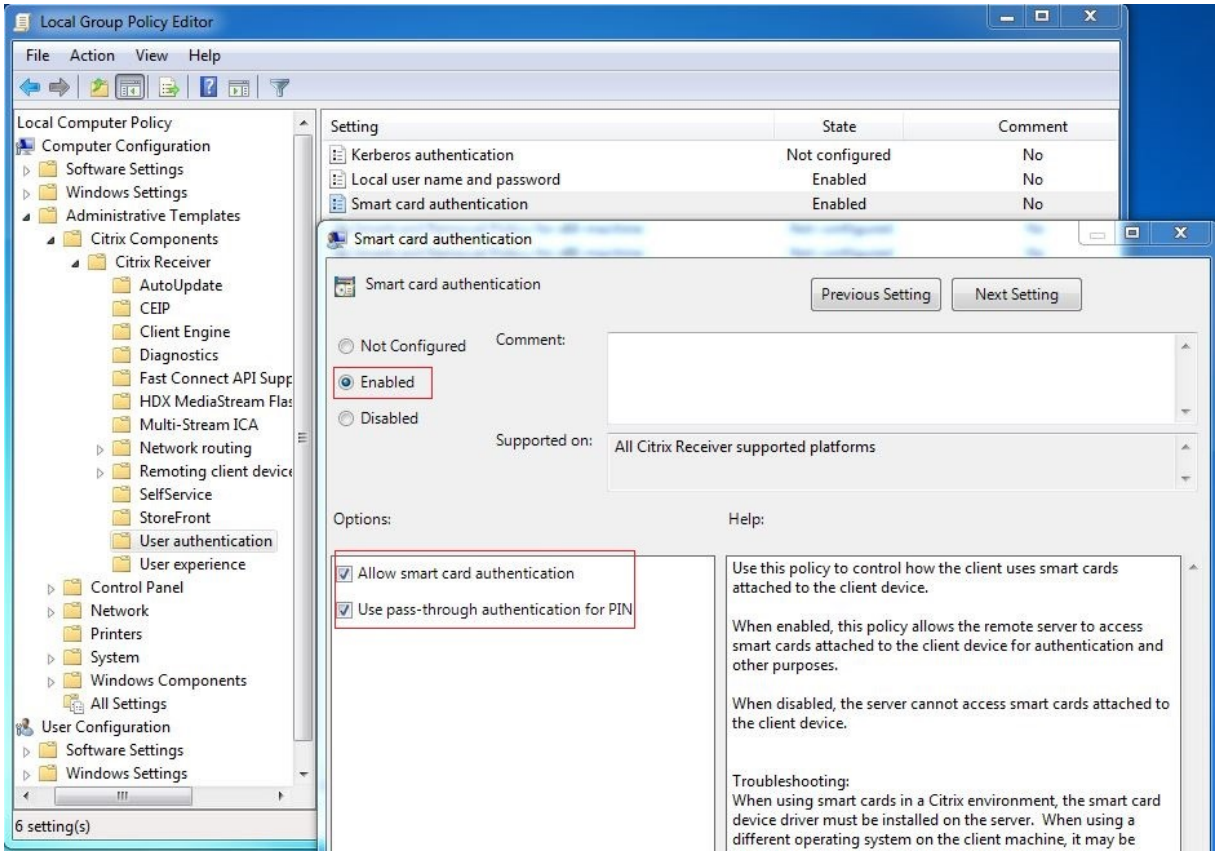
The configuration file resembles the following after modification if SSSD is used.

```
#%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
```

(Optional) Single sign-on by using smart cards

Single sign-on (SSO) is a Citrix feature that implements pass-through authentication with virtual desktop and application launches. This feature reduces the number of times that users type their PIN. To use SSO with the Linux VDA, configure Citrix Workspace app. The configuration is the same with the Windows VDA. For more information, see Knowledge Center article CTX133982.

Enable the smart card authentication as follows when configuring the group policy in Citrix Workspace app.



Fast smart card logon

Fast smart card is an improvement over the existing HDX PC/SC-based smart card redirection. It improves performance when smart cards are used in high-latency WAN environments. For more information, see Smart cards.

The Linux VDA supports fast smart card on the following versions of Citrix Workspace app:

- Citrix Receiver for Windows 4.12
- Citrix Workspace app 1808 for Windows and later

Enable fast smart card logon on the client Fast smart card logon is enabled by default on the VDA and disabled by default on the client. On the client, to enable fast smart card logon, include the following parameter in the default.ica file of the associated StoreFront site:

```
1 [WFClient]
2 SmartCardCryptographicRedirection=On
3 <!--NeedCopy-->
```

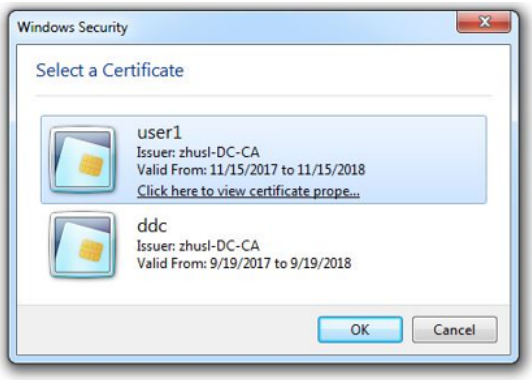

Disable fast smart card logon on the client To disable fast smart card logon on the client, remove the **SmartCardCryptographicRedirection** parameter from the default.ica file of the associated StoreFront site.

Usage

Log on to the Linux VDA by using a smart card

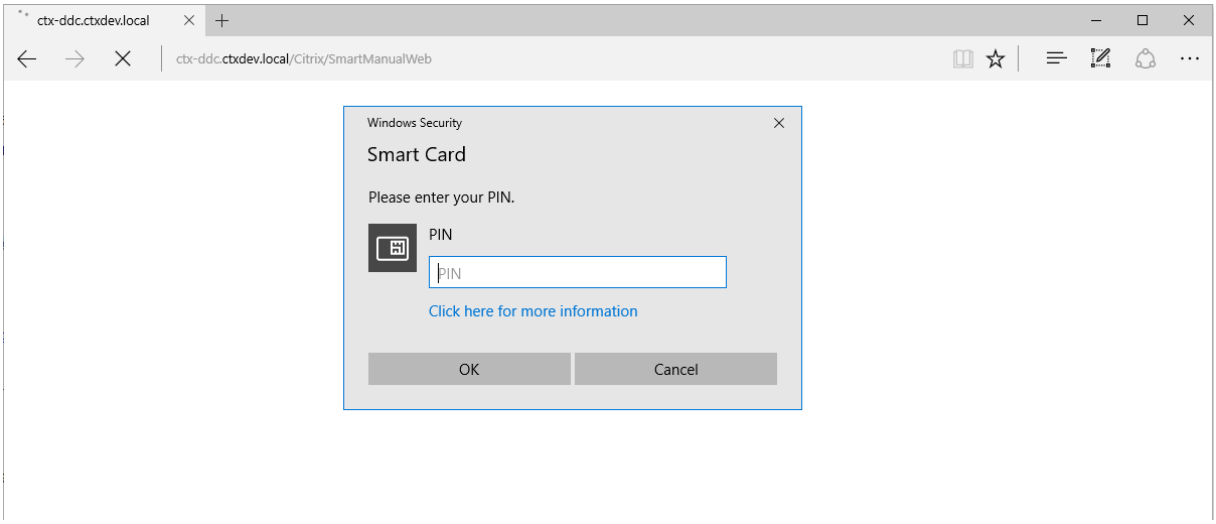
Users can use a smart card to log on to the Linux VDA in both SSO and non-SSO scenarios.

- In the SSO scenario, users are logged on to StoreFront automatically by using the cached smart card certificate and PIN. When users launch a Linux virtual desktop session in StoreFront, the PIN is passed to the Linux VDA for smart card authentication.
- In the non-SSO scenario, users are prompted to select a certificate and type a PIN to log on to StoreFront.



When users launch a Linux virtual desktop session in StoreFront, a dialog box for logon to the Linux VDA appears as follows. The user name is extracted from the certificate in the smart card and users must type the PIN again for logon authentication.

This behavior is the same with the Windows VDA.



Reconnect to a session by using a smart card

To reconnect to a session, ensure that the smart card is connected to the client device. Otherwise, a gray caching window appears on the Linux VDA side and exits quickly because reauthentication fails without the smart card connected. No other prompt is provided in this case to remind you to connect the smart card.

On the StoreFront side, however, if a smart card is not connected when you try to reconnect to a session, the StoreFront web might give an alert as follows.

Limitation

Smart card removal policy

Now, the Linux VDA uses only the default behavior for smart card removal. When you remove the smart card after logging on to the Linux VDA successfully, the session still keeps connected and the session screen is not locked.

Support for other smartcards and the PKCS#11 library

Although only the OpenSC smart card is listed on our support list, you can try using other smart cards and the PKCS#11 library because Citrix is providing a generic smart card redirection solution. To switch to your specific smart card or the PKCS#11 library:

1. Replace all the `opensc-pkcs11.so` instances with your PKCS#11 library.
2. To set the path of your PKCS#11 library to the registry, run the following command:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
   CurrentControlSet\Control\Citrix\VirtualChannels\Scard" -v "
   PKCS11LibPath" -d "PATH"
<!--NeedCopy-->
```

   where **PATH** points to your PKCS#11 library such as /usr/lib64/pkcs11/opensc-pkcs11.so
3. Disable fast smart card logon on the client.

Double-hop single sign-on authentication

June 11, 2021

The feature injects user credentials entered for accessing a StoreFront store to the AuthManager module of Citrix Workspace app for Linux and Citrix Receiver for Linux 13.10. After injection, you can use the client to access virtual desktops and applications from within a Linux virtual desktop session, without entering user credentials for a second time.

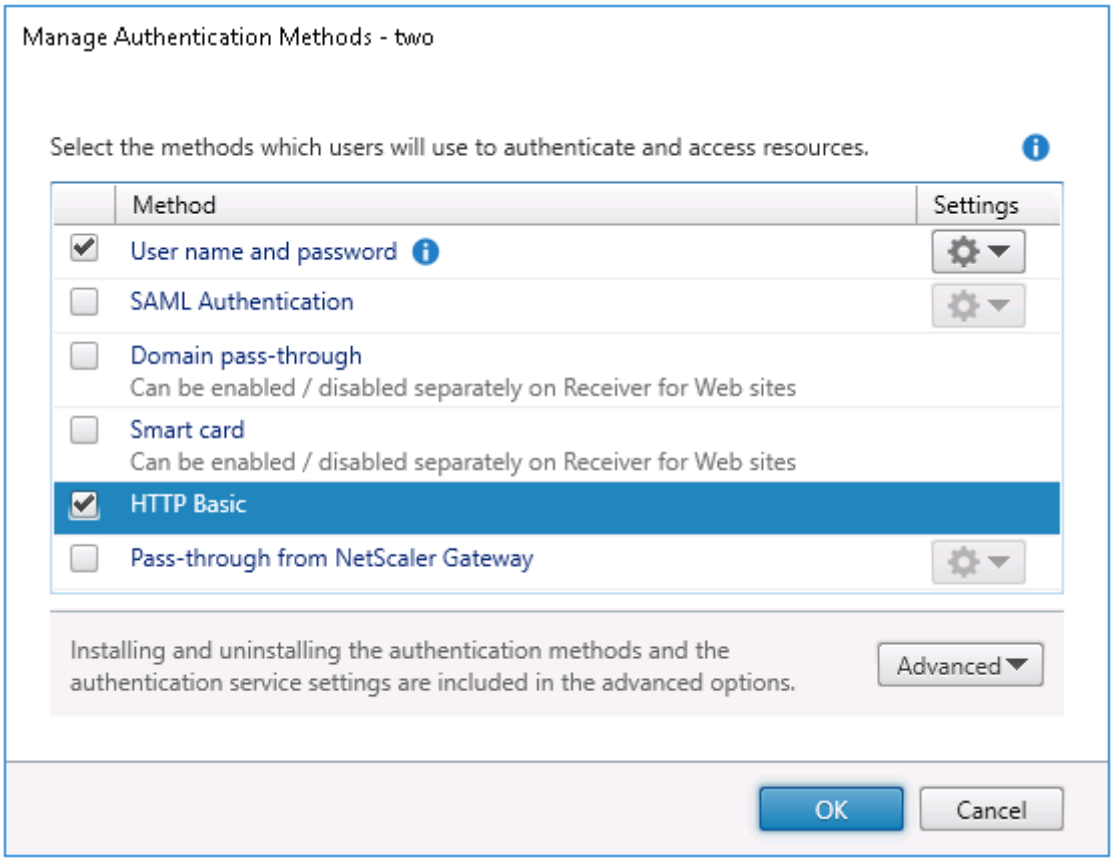
> **Note:**
>
> This feature is supported on Citrix Workspace app for Linux and Citrix Receiver for Linux 13.10.

To enable the feature:

1. On the Linux VDA, install Citrix Workspace app for Linux or Citrix Receiver for Linux 13.10.

   Download the app from the Citrix download page for Citrix Workspace app or for Citrix Receiver.

   The default installation path is /opt/Citrix/ICAClient/. If you install the app to a different path, set the ICAROOT environment variable to point to the actual installation path.
2. In the Citrix StoreFront management console, add the **HTTP Basic** authentication method for the target store.



3. Add the following key to the AuthManager configuration file ($ICAROOT/config/AuthManConfig.xml) for allowing the HTTP Basic authentication:

```
<Protocols>
     <HTTPBasic>
         <Enabled>True</Enabled>
    </HTTPBasic>
</Protocols>
<!--NeedCopy-->
```

4. Run the following commands to install the root certificate in the specified directory.

```
cp rootcert.pem $ICAROOT/keystore/cacerts/
$ICAROOT/util/ctx_rehash  $ICAROOT/keystore/cacerts/
<!--NeedCopy-->
```

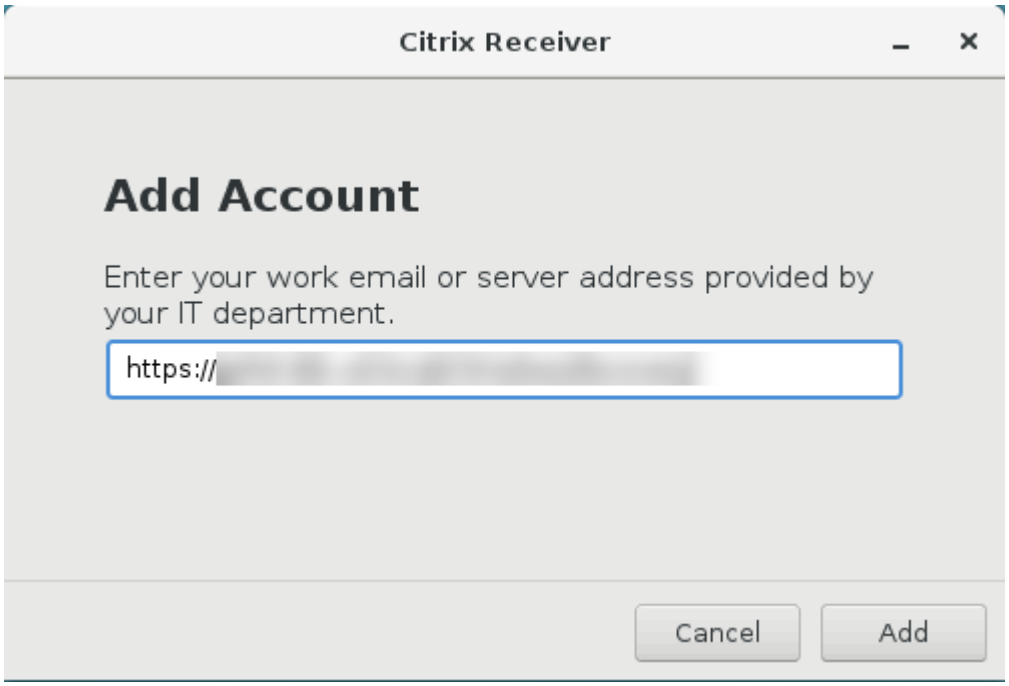
5. Run the following command to enable the feature:

```
/opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
   CurrentControlSet\Control\Citrix" -v "LurSsonEnabled" -d "0
   x00000001"
<!--NeedCopy-->
```

6. Launch a Linux virtual desktop session and start Citrix Workspace app for Linux or Citrix Receiver for Linux 13.10 within that session.

You are prompted for a store account the first time you start Citrix Workspace app for Linux or Citrix Receiver for Linux 13.10 within a Linux virtual desktop session. Later on, you are logged on to the store you specified earlier automatically.

> **Note:**
>
> Enter an HTTPS URL as your store account.



Configure unauthenticated sessions

June 11, 2021

Use the information in this article to configure unauthenticated sessions. No special settings are required when installing the Linux VDA to use this feature.

> **Note:**
>
> When configuring unauthenticated sessions, consider that session prelaunch is not supported. Session prelaunch is also not supported on Citrix Workspace app for Android.

Create an unauthenticated store

To support an unauthenticated session on the Linux VDA, create an unauthenticated store using StoreFront.

Enable unauthenticated users in a Delivery Group

After creating an unauthenticated store, enable unauthenticated users in a Delivery Group to support an unauthenticated session. To enable unauthenticated users in a Delivery Group, follow the instructions in the Citrix Virtual Apps and Desktops documentation.

Set the unauthenticated session idle time

An unauthenticated session has a default idle timeout of 10 minutes. This value is configured through the registry setting **AnonymousUserIdleTime**. Use the **ctxreg** tool to change this value. For example, to set this registry setting to five minutes:

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
    CurrentControlSet\Control\Citrix" -v AnonymousUserIdleTime -d 0
    x00000005
<!--NeedCopy-->
```

Set the maximum number of unauthenticated users

To set the maximum number of unauthenticated users, use the registry key **MaxAnonymousUserNumber**. This setting limits the number of unauthenticated sessions running on a single Linux VDA concurrently. Use the **ctxreg** tool to configure this registry setting. For example, to set the value to 32:

```
sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\System\
    CurrentControlSet\Control\Citrix" -v MaxAnonymousUserNumber -d 0
    x00000020
<!--NeedCopy-->
```

> **Important:**
>
> Limit the number of unauthenticated sessions. Too many sessions being launched concurrently can cause problems on the VDA, including running out of available memory.

Troubleshooting

Consider the following when configuring unauthenticated sessions:

- **Failed to log on to an unauthenticated session**.

Verify that the registry was updated to include the following (set to 0):

```
sudo /opt/Citrix/VDA/bin/ctxreg read -k "HKLM\System\CurrentControlSet
    \Control\Citrix" -v MaxAnonymousUserNumber
<!--NeedCopy-->
```


Verify that the **ncsd** service is running and configured to enable **passwd** cache:

```
ps  uax | grep nscd
cat /etc/nscd.conf | grep 'passwd' | grep 'enable-cache'
<!--NeedCopy-->
```

Set the **passwd** cache variable to **no** if it is enabled, then restart the **ncsd** service. You might need to reinstall the Linux VDA after changing this configuration.

- **The lock screen button is displayed in an unauthenticated session with KDE**.

The lock screen button and menu are disabled by default in an unauthenticated session. However, they can still be displayed in KDE. In KDE, to disable the lock screen button and menu for a particular user, add the following lines to the configuration file **$Home/.kde/share/config/kdeglobals**. For example:

```
[KDE Action Restrictions]
action/lock_screen=false
<!--NeedCopy-->
```

However, if the `KDE Action Restrictions` parameter is configured as immutable in a global wide `kdeglobals` file such as `/usr/share/kde-settings/kde-profile/default/share/config/kdeglobals`, the user configuration has no effect.

To resolve this issue, modify the system-wide `kdeglobals` file to remove the `[$i]` tag at the `[KDE Action Restrictions]` section, or directly use the system-wide configuration to disable the lock screen button and menu. For details about the KDE configuration, see the KDE System Administration/Kiosk/Keys page.

Configure LDAPS

October 7, 2021

Secure LDAP (LDAPS) allows you to enable the Secure Lightweight Directory Access Protocol for your Active Directory managed domains to provide communications over SSL (Secure Socket Layer)/TLS (Transport Layer Security).

By default, LDAP communications between client and server applications are not encrypted. LDAP using SSL/TLS (LDAPS) enables you to protect the LDAP query content between the Linux VDA and the LDAP servers.

The following Linux VDA components have dependencies on LDAPS:

- Broker agent: Linux VDA registration with a Delivery Controller
- Policy service: Policy evaluation

Configuring LDAPS involves:

- Enable LDAPS on the Active Directory (AD)/LDAP server
- Export the root CA for client use
- Enable/disable LDAPS on the Linux VDA
- Configure LDAPS for third-party platforms
- Configure SSSD
- Configure Winbind
- Configure Centrify
- Configure Quest

Enable LDAPS on the AD/LDAP server

You can enable LDAP over SSL (LDAPS) by installing a properly formatted certificate from either a Microsoft certification authority (CA) or a non-Microsoft CA.

> **Tip:**
>
> LDAP over SSL/TLS (LDAPS) is enabled automatically when you install an Enterprise Root CA on a domain controller.

For more information about how to install the certificate and verify the LDAPS connection, see How to enable LDAP over SSL with a third-party certification authority on the Microsoft Support site.

When you have a multi-tier (such as a two-tier or three-tier) certificate authority hierarchy, you do not automatically have the appropriate certificate for LDAPS authentication on the domain controller.

For information about how to enable LDAPS for domain controllers using a multi-tier certificate authority hierarchy, see the LDAP over SSL (LDAPS) Certificate article on the Microsoft TechNet site.

Enable root certificate authority for client use

The client must be using a certificate from a CA that the LDAP server trusts. To enable LDAPS authentication for the client, import the root CA certificate to a trusted keystore.

For more information about how to export Root CA, see How to export Root Certification Authority Certificate on the Microsoft Support website.

Enable or disable LDAPS on the Linux VDA

To enable or disable LDAPS on the Linux VDA, run the following script (while logged on as an administrator):

The syntax for this command includes the following:

- Enable LDAP over SSL/TLS with the root CA certificate provided:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enable pathToRootCA
<!--NeedCopy-->
```

- Enable LDAP over SSL/TLS with channel binding:

```
/opt/Citrix/VDA/sbin/enable_ldaps.sh -Enablecb pathToRootCA
<!--NeedCopy-->
```

> **Note:**
>
> The root CA certificate for channel binding must be in PEM format. If enabling LDAPS does not create a Python3 virtual environment successfully, create it manually following the instructions at Create a Python3 virtual environment.

- Fall back to LDAP without SSL/TLS

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

The Java keystore dedicated for LDAPS is located in **/etc/xdl/.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
<!--NeedCopy-->
```

Configure LDAPS for third-party platform

Besides the Linux VDA components, several third-party software components that adhere to the VDA might also require secure LDAP, such as SSSD, Winbind, Centrify, and Quest. The following sections describe how to configure secure LDAP with LDAPS, STARTTLS, or SASL sign and seal.

> **Tip:**
>
> Not all of these software components prefer to use SSL port 636 to ensure secure LDAP. And most of the time, LDAPS (LDAP over SSL on port 636) cannot coexist with STARTTLS on port 389.

SSSD

Configure the SSSD secure LDAP traffic on port 636 or port 389 as per the options. For more information, see the SSSD LDAP Linux man page.

Winbind

The Winbind LDAP query uses the ADS method. Winbind supports only the StartTLS method on port 389. Affected configuration files are **ldap.conf** at **/etc/openldap/ldap.conf** and **smb.conf** at **/etc/samba/smb.conf**. Change the files as follows:

```
ldap.conf:

TLS_REQCERT never

smb.conf:

ldap ssl = start tls
ldap ssl ads = yes
client ldap sasl wrapping = plain
<!--NeedCopy-->
```

Alternately, secure LDAP can be configured by SASL GSSAPI sign and seal, but it cannot coexist with TLS/SSL. To use SASL encryption, change the **smb.conf** configuration:

```
smb.conf:

ldap ssl = off
ldap ssl ads = no
client ldap sasl wrapping = seal
<!--NeedCopy-->
```

Centrify

Centrify does not support LDAPS on port 636. However, it does provide secure encryption on port 389. For more information, see the Centrify site.

Quest

Quest Authentication Service does not support LDAPS on port 636, but it provides secure encryption on port 389 using a different method.

Troubleshooting

The following issues might arise when you use this feature:

- **LDAPS service availability**

  Verify that the LDAPS connection is available on the AD/LDAP server. The port is on 636 by default.

- **Linux VDA registration failed when LDAPS is enabled**

  Verify that the LDAP server and ports are configured correctly. Check the Root CA Certificate first and ensure that it matches the AD/LDAP server.

- **Incorrect registry change by accident**

  If the LDAPS related keys were updated by accident without using **enable_ldaps.sh**, it might break the dependency of LDAPS components.

- **LDAP traffic is not encrypted through SSL/TLS from Wireshark or any other network monitoring tools**

  By default, LDAPS is disabled. Run **/opt/Citrix/VDA/sbin/enable_ldaps.sh** to force it.

- **There is no LDAPS traffic from Wireshark or any other networking monitoring tool**

  LDAP/LDAPS traffic occurs when Linux VDA registration and Group Policy evaluation occur.

- **Failed to verify LDAPS availability by running ldp connect on the AD server**

  Use the AD FQDN instead of the IP Address.

- **Failed to import Root CA certificate by running the /opt/Citrix/VDA/sbin/enable_ldaps.sh script**

  Provide the full path of the CA certificate, and verify that the Root CA Certificate is the correct type. It is supposed to be compatible with most of the Java Keytool types supported. If it is not listed in the support list, you can convert the type first. We recommend the base64 encoded PEM format if you encounter a certificate format problem.

- **Failed to show the Root CA certificate with Keytool -list**

  When you enable LDAPS by running `/opt/Citrix/VDA/sbin/enable_ldaps.sh`, the certificate is imported to /etc/xdl/.keystore, and the password is set to protect the keystore. If you forget the password, you can rerun the script to create a keystore.

Create a Python3 virtual environment

October 18, 2023

If you are connecting to the network, running the `sudo /opt/Citrix/VDA/bin/xdping` or `/opt/Citrix/VDA/sbin/enable_ldaps.sh` command can create a Python3 virtual environment. However, if the commands fail to create a Python3 virtual environment, you can create it manually even without a network connection. This article details the prerequisites and steps to create a Python3 virtual environment without a network connection.

Prerequisites

- You must have administrative privileges to access the `/opt/Citrix/VDA/sbin/ctxpython3` directory.
- The wheel files of `Python3` packages are in place. You can download the wheel files from https://pypi.org/.

Create a Python3 virtual environment

Complete the following steps to create a Python3 virtual environment:

1. Install Python3 dependencies.

   For RHEL:

   ```
   yum -y install python36-devel krb5-devel gcc
   <!--NeedCopy-->
   ```

   > **Note:**
   >
   > You might have to enable a particular repository to install some dependencies. For RHEL 7, run the `subscription-manager repos --enable rhel-7-server-optional-rpms` command. For RHEL 8, run the `subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms` command.

   For Ubuntu\Debian:

   ```
   apt-get -y install python3-dev python3-pip python3-venv libkrb5-dev
   <!--NeedCopy-->
   ```

   For SUSE:

   ```
   zypper -i -n install python3-devel python3-setuptools krb5-devel gcc libffi48-devel
   <!--NeedCopy-->
   ```


> **Note:**
>
> You might have to enable the SUSE_Linux_Enterprise_Software_Development_Kit_12_SP repository to install some dependencies.

2. Create a Python3 virtual environment.

   For RHEL, Ubuntu, Debian:

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3
<!--NeedCopy-->
```

   For SUSE:

```
sudo python3 -m venv /opt/Citrix/VDA/sbin/ctxpython3

sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
    setuptools==40.6.2
<!--NeedCopy-->
```

3. Install LDAPS dependencies.

   For RHEL, Ubuntu, Debian:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
    cffi == 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi
    ==1.6.2 ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0
    psutil == 5.8.0 pyasn1 == 0.4.8 pycparser == 2.20 scramp ==
    1.2.0 six == 1.15.0 termcolor == 1.1.0
<!--NeedCopy-->
```

   For SUSE:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install
    cffi == 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi
    ==1.6.2 ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0
    psutil == 5.8.0 pyasn1 == 0.4.8 pycparser == 2.20 scramp ==
    1.2.0 six == 1.15.0 termcolor == 1.1.0
<!--NeedCopy-->
```

4. Install XDPing dependencies.

   For RHEL, Ubuntu, Debian:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install
    cffi == 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi
    ==1.6.2 ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0
    psutil == 5.8.0 pyasn1 == 0.4.8 pycparser == 2.20 scramp ==
    1.2.0 six == 1.15.0 termcolor == 1.1.0

sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m pip install /
    opt/Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
<!--NeedCopy-->
```


For SUSE:

```
sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install
    cffi == 1.14.2 cryptography == 3.1 decorator == 4.4.2 gssapi
    ==1.6.2 ldap3==2.8.1 netifaces == 0.10.9 pg8000 == 1.17.0
    psutil == 5.8.0 pyasn1 == 0.4.8 pycparser == 2.20 scramp ==
    1.2.0 six == 1.15.0 termcolor == 1.1.0

sudo /opt/Citrix/VDA/sbin/ctxpython3/bin/python3 -m easy_install /
    opt/Citrix/VDA/sbin/ctxpython3/packages/xdping-*.whl
<!--NeedCopy-->
```

XDPing

June 11, 2021

Description

The Linux XDPing tool is a command-line based application that automates the process of checking for common configuration issues with a Linux VDA environment.

The Linux XDPing tool performs over 150 individual tests on the system, which are broadly categorized as follows:

- Check whether Linux VDA system requirements are met
- Identify and display machine information including the Linux distributions
- Check the Linux kernel compatibility
- Check for any known Linux distribution issues that can impact the Linux VDA operation
- Check the Security-Enhanced Linux (SELinux) mode and compatibility
- Identify network interfaces and check network settings
- Check storage partitioning and available disk space
- Check machine host and domain name configuration
- Check DNS configuration and perform lookup tests
- Identify underlying hypervisors and check virtual machine configuration. Support for:
  - Citrix Hypervisor
  - Microsoft HyperV
  - VMware vSphere
- Check time settings and check whether network time synchronization is operational
- Check whether PostgreSQL service is properly configured and operational
- Check whether the firewall is enabled and required ports are open

- Check Kerberos configuration and perform authentication tests
- Check the LDAP search environment for the group policy service engine
- Check whether Active Directory integration is set up properly and the current machine is joined to the domain. Support for:
  - Samba Winbind
  - Dell Quest Authentication Services
  - Centrify DirectControl
  - SSSD
- Check the integrity of the Linux computer object in Active Directory
- Check Pluggable Authentication Module (PAM) configuration
- Check the core dump pattern
- Check whether packages required by the Linux VDA are installed
- Identify the Linux VDA package and check the integrity of the installation
- Check the integrity of the PostgreSQL registry database
- Check whether the Linux VDA services are properly configured and operational
- Check the integrity of the VDA and HDX configuration
- Probe each configured Delivery Controller to test that the Broker Service is reachable, operational, and responsive
- Check whether the machine is registered with the Delivery Controller farm
- Check the state of each active or disconnected HDX session
- Scan log files for the Linux VDA related errors and warnings
- Check whether the version of Xorg is suitable

Use the Linux XDPing tool

> **Note:**
>
> Running ctxsetup.sh does not install XDPing. You can run `sudo /opt/Citrix/VDA/bin/xdping` to install XDPing.
> This command also creates a Python3 virtual environment that is required for XDPing. If this command fails to create a Python3 virtual environment, create it manually following the instructions at Create a Python3 virtual environment.

XDPing comes with the single executable named `xdping` that is run from the command shell.

To display the command-line options, use the `--help` option:

```
sudo /opt/Citrix/VDA/bin/xdping --help
<!--NeedCopy-->
```

To run the full suite of tests, run `xdping` without any command-line options:


```
sudo /opt/Citrix/VDA/bin/xdping
<!--NeedCopy-->
```

To check the environment before installing the Linux VDA package, run the `pre-flight` tests:

```
sudo /opt/Citrix/VDA/bin/xdping --preflight
<!--NeedCopy-->
```

To run specific test categories only, for example, the time and Kerberos tests, use the `-T` option:

```
sudo /opt/Citrix/VDA/bin/xdping -T time,kerberos
<!--NeedCopy-->
```

To probe a particular XenDesktop Controller:

```
sudo /opt/Citrix/VDA/bin/xdping -d myddc.domain.net
<!--NeedCopy-->
```

Sample Output

The following is a sample output from running the Kerberos test:

```
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/m                  l]: [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]
```


Configure Xauthority

June 11, 2021

The Linux VDA supports environments that use X11 display functionality (including `xterm` and `gvim`) for interactive remoting. This feature provides a security mechanism necessary to ensure secure communication between XClient and XServer.

There are two methods to secure permission for this secure communication:

- **Xhost**. By default, Xhost allows only the localhost XClient to communicate with XServer. If you choose to allow a remote XClient to access XServer, the Xhost command must be run to grant permission on the specific machine. Or, you can alternately use **xhost +** to allow any XClient to connect to XServer.
- **Xauthority**. The `.Xauthority` file can be found in each user's home directory. It is used to store credentials in cookies used by xauth for authentication of XServer. When an XServer instance (Xorg) is started, the cookie is used to authenticate connections to that specific display.

How it works

When Xorg starts up, a `.Xauthority` file is passed to the Xorg. This `.Xauthority` file contains the following elements:

- Display number
- Remote request protocol
- Cookie number

You can browse this file using the `xauth` command. For example:

```
# xauth -f ~/.Xauthority

# > list

# > us01msip06:107  MIT-MAGIC-COOKIE-1
     fb228d1b695729242616c5908f11624b
<!--NeedCopy-->
```

If XClient connects to the Xorg remotely, two prerequisites must be met:

- Set the **DISPLAY** environment variable to the remote XServer.
- Get the `.Xauthority` file which contains one of the cookie numbers in Xorg.

Configure Xauthority

To enable Xauthority on the Linux VDA for remote X11 display, you must create the following two registry keys:

```
sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
    CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "
    XauthEnabled" -d "0x00000001" --force

sudo /opt/Citrix/VDA/bin/ctxreg create -k "HKLM\System\
    CurrentControlSet\Control\Citrix\Xorg" -t "REG_DWORD" -v "ListenTCP"
     -d "0x00000001" --force
<!--NeedCopy-->
```

After enabling Xauthority, pass the `.Xauthority` file to the XClient manually or by mounting a shared home directory:

- Pass the `.Xauthority` file to the XClient manually

  After launching an ICA session, the Linux VDA generates the `.Xauthority` file for the XClient and stores the file in the logon user's home directory. You can copy this `.Xauthority` file to the remote XClient machine, and set the `DISPLAY` and `XAUTHORITY` environment variables. `DISPLAY` is the display number stored in the `.Xauthority` file and `XAUTHORITY` is the file path of Xauthority. For an example, see the following command:

  ```
  export DISPLAY={
  Display number stored in the Xauthority file }


  export XAUTHORITY={
  the file path of .Xauthority }

  <!--NeedCopy-->
  ```

  > **Note:**
  >
  > If the `XAUTHORITY` environment variable is not set, the `~/.Xauthority` file is used by default.

- Pass the `.Xauthority` file to the XClient by mounting a shared home directory

  The convenient way is to mount a shared home directory for the logon user. When the Linux VDA starts an ICA session, the `.Xauthority` file is created under the logon user's home directory. If this home directory is shared with the XClient, the user does not need to transmit this `.Xauthority` file to the XClient manually. After the `DISPLAY` and `XAUTHORITY` environment variables are set correctly, the GUI is displayed in the XServer desktop automatically.

Troubleshooting

If Xauthority does not work, follow the troubleshooting steps:

1. As an administrator with root privilege, retrieve all Xorg cookies:

   ```
   ps aux | grep -i xorg
   <!--NeedCopy-->
   ```

   This command displays the Xorg process and the parameters passed to Xorg while starting. Another parameter displays which `.Xauthority` file is used. For example:

   ```
   /var/xdl/xauth/.Xauthority110
   <!--NeedCopy-->
   ```

   Display the cookies using the **Xauth** command:

   ```
   Xauth -f /var/xdl/xauth/.Xauthority110
   <!--NeedCopy-->
   ```

2. Use the `Xauth` command to show the cookies contained in `~/.Xauthority`. For the same display number, the displayed cookies must be the same in the `.Xauthority` files of Xorg and XClient.

3. If the cookies are the same, check the remote display port accessibility by using the IP address of the Linux VDA (for example, 10.158.11.11) and the published desktop display number (for example, 160).

   Run the following command on the XClient machine:

   ```
   telnet 10.158.11.11 6160
   <!--NeedCopy-->
   ```

   The port number is the sum of 6000 + <display number>.

   If this telnet operation fails, the firewall might be blocking the request.

Configure Federated Authentication Service

March 28, 2023

The Linux VDA supports using FAS to log on to your Citrix Virtual Apps and Desktops environment. It uses the same Windows environment as the Windows VDA for the FAS logon feature. For information about configuring the Windows environment for FAS, see Federated Authentication Service. This article provides extra information specific to the Linux VDA.

> **Note**
>
> The Linux VDA does not support the **In-session Behavior** policy.
>
> The Linux VDA uses short connections to transmit data with FAS servers.

Configure FAS on the Linux VDA

FAS support on RHEL 8/CentOS 8

FAS depends on the pam_krb5 module, which is deprecated on RHEL 8/CentOS 8. To use FAS on RHEL 8/CentOS 8, build the pam_krb5 module as follows:

1. Download the pam_krb5-2.4.8-6 source code from the following website:

   https://centos.pkgs.org/7/centos-x86_64/pam_krb5-2.4.8-6.el7.x86_64.rpm.html.

2. Build and install the pam_krb5 module on RHEL 8/CentOS 8.

```
yum install make gcc krb5-devel pam-devel autoconf libtool
rpm2cpio pam_krb5-2.4.8-6.el7.src.rpm | cpio -div
tar xvzf pam_krb5-2.4.8.tar.gz
cd pam_krb5-2.4.8
./configure --prefix=/usr
make
make install
<!--NeedCopy-->
```

3. Verify that pam_krb5.so exists under /usr/lib64/security/.

```
ls -l /usr/lib64/security | grep pam_krb5
<!--NeedCopy-->
```

Set FAS servers

For fresh Linux VDA installation, to use FAS, type the FQDN of each FAS server when you are asked for CTX_XDL_FAS_LIST during the execution of ctxinstall.sh or ctxsetup.sh. Because the Linux VDA does not support AD Group Policy, you can provide a semicolon-separated list of FAS servers instead. If any server address is removed, fill its blank with the **<none>** text string and do not modify the order of server addresses.

For upgrading an existing Linux VDA installation, you can rerun `ctxsetup.sh` to set the FAS servers. Or you can run the following commands to set the FAS servers and to restart the `ctxvda` service to make your setting take effect.

```
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 service ctxjproxy restart
4
5 service ctxvda restart
6 <!--NeedCopy-->
```

To update the FAS servers through `ctxreg`, run the following commands:

```
1 sudo /opt/Citrix/VDA/bin/ctxreg update -k "HKLM\Software\Citrix\
     VirtualDesktopAgent\Authentication\UserCredentialService" -v "
     Addresses" -d "<Your-FAS-Server-List>"
2
3 service ctxjproxy restart
4
5 service ctxvda restart
6 <!--NeedCopy-->
```

Install certificates

For the verification of users' certificates, install the root CA certificate and all intermediate certificates on the VDA. For example, to install the root CA certificate, obtain the AD root certificate from the preceding **Retrieve the CA Certificate from the Microsoft CA (on AD)** step, or download its DER format from the root CA server `http://CA-SERVER/certsrv`.

> **Note:**
>
> The following commands also apply to configuring an intermediate certificate.

Convert a DER file (.crt, .cer, .der) to PEM by running the command similar to the following:

```
1 sudo openssl x509 -inform der -in root.cer -out root.pem
2 <!--NeedCopy-->
```

Then, install the root CA certificate to the `openssl` directory by running the command similar to the following:

```
1 sudo cp root.pem /etc/pki/CA/certs/
2 <!--NeedCopy-->
```

> **Note**
>
> Do not put the root CA certificate under the **/root** path. Otherwise, FAS does not have the read permission to the root CA certificate.

Run ctxfascfg.sh

Run the ctxfascfg.sh script to configure FAS parameters:


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

Two environment variables are added so that `ctxfascfg.sh` can be run in silent mode:

- **CTX_FAS_ADINTEGRATIONWAY=winbind | sssd | centrify** –Denotes the Active Directory integration method, which equals to `CTX_EASYINSTALL_ADINTEGRATIONWAY` when `CTX_EASYINSTALL_ADINTEGRATIONWAY` is specified. If `CTX_EASYINSTALL_ADINTEGRATIONWAY` is not specified, `CTX_FAS_ADINTEGRATIONWAY` uses its own value setting.
- **CTX_FAS_CERT_PATH =<certificate path>** –Specifies the full path that stores the root certificate and all intermediate certificates.

Choose the correct Active Directory integration method and then type the correct path of certificates (for example, `/etc/pki/CA/certs/`).

The script then installs the krb5-pkinit and pam_krb5 packages and sets the relevant configuration files.

Limitation

- FAS supports limited Linux platforms and AD integration methods. See the following matrix:

	Winbind	SSSD	Centrify
RHEL 8.3 / CentOS 8.3	Yes	Yes	Yes
RHEL 8.2 / CentOS 8.2	Yes	Yes	Yes
RHEL 8.1 / CentOS 8.1	Yes	Yes	Yes
RHEL 7.9 / CentOS 7.9	Yes	Yes	Yes
RHEL 7.8 / CentOS 7.8	Yes	Yes	Yes
Ubuntu 20.04	Yes	No	Yes
Ubuntu 18.04	Yes	No	Yes
Ubuntu 16.04	Yes	No	Yes
SLES 12.5	Yes	No	Yes

- FAS does not support lock screen yet. If you click the lock button in a session, you cannot log back on to the session again by using FAS.
- This release supports only the common FAS deployments summarized in the Federated Authentication Service architectural overview article and does not include **Windows 10 Azure AD Join**.

Troubleshooting

Before troubleshooting FAS, ensure that the Linux VDA is installed and configured correctly so that a non-FAS session can be launched successfully on the common store by using password authentication.

If non-FAS sessions work properly, set the HDX log level of the **Login** class to VERBOSE and the VDA log level to TRACE. For information on how to enable trace logging for the Linux VDA, see Knowledge Center article CTX220130.

FAS server configuration error

Launching a session from the FAS store fails.

Check **/var/log/xdl/hdx.log** and find the error log similar to the following:

```
2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: [Logon Type] Federated Authentication Logon.

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: entry

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas: start connect to server 0

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: connect_fas0: failed to connect: Connection refused.

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: failed to connect to server [0], please confirm if fas service list is well configurated in condb

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_fas: exit, 43

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: validate_user: failed to validate fas credential

2021-01-28 01:42:16.164 <P26422:S4> citrix-ctxlogin: LoginBoxValidate: failed validation of user 'user1@CTXDEV.LOCAL', INVALID_PARAMETER

<!--NeedCopy-->
```

Solution Run the following command to verify that the Citrix registry value "HKEY_LOCAL_MACHINE\SOFTWARE
is set to <Your-FAS-Server-List>.

```
sudo /opt/Citrix/VDA/bin/ctxreg dump | grep "UserCredentialService"
<!--NeedCopy-->
```


If the existing setting is incorrect, follow the preceding Set FAS servers step to set it again.

Incorrect CA certificate configuration

Launching a session from the FAS store fails. A gray window appears and disappears several seconds later.

Check **/var/log/xdl/hdx.log** and find the error log similar to the following:

```
2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
   get_logon_certificate: entry

2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin: check_caller:
   current process: pid [30656], name [/opt/Citrix/VDA/bin/ctxlogin]

2021-01-28 01:47:46.210 <P30656:S5> citrix-ctxlogin:
   get_public_certificate: entry

2021-01-28 01:47:46.211 <P30656:S5> citrix-ctxlogin: query_fas: waiting
    for response...

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: query_fas: query
   to server success

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin:
   get_public_certificate: exit

2021-01-28 01:47:46.270 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
    input size 1888

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: fas_base64_decode:
    output size 1415

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
   get_logon_certificate: get logon certificate success

2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin: cache_certificate:
    cache certificate success

```



```
2021-01-28 01:47:46.271 <P30656:S5> citrix-ctxlogin:
   get_logon_certificate: exit, 0

2021-01-28 01:47:48.060 <P30656:S5> citrix-ctxlogin: validate_user:
   pam_authenticate err,can retry for user user1@CTXDEV.LOCAL
<!--NeedCopy-->
```

Solution Verify that the full path that stores the root CA certificate and all intermediate certificates is set correctly in /etc/krb5.conf. The full path is similar to the following:

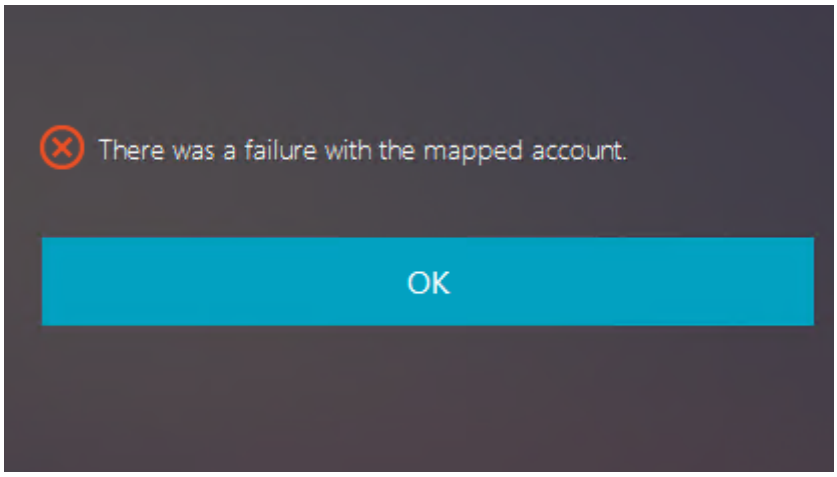
```
 [realms]

EXAMPLE.COM = {


    ......

    pkinit_anchors = DIR:/etc/pki/CA/certs/

    ......

 }

<!--NeedCopy-->
```

If the existing setting is incorrect, follow the preceding Install certificates step to set it again.

Alternatively, check whether the root CA certificate is valid.

Shadow account mapping error

FAS is configured by SAML authentication. The following error might occur after an ADFS user enters the user name and password on the ADFS logon page.

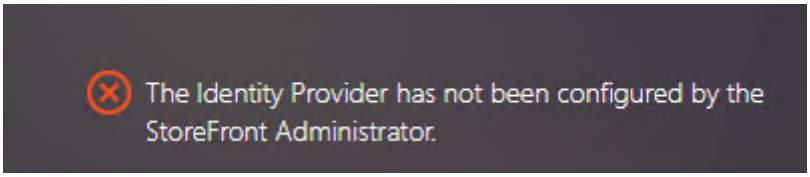


This error indicates that the ADFS user has been verified successfully, but there is no shadow user configured on AD.

Solution Set the Shadow Account on AD.

ADFS not configured

The following error occurs during a logon attempt to the FAS store:



The issue occurs when the FAS store is configured to use SAML authentication but the ADFS deployment is missing.

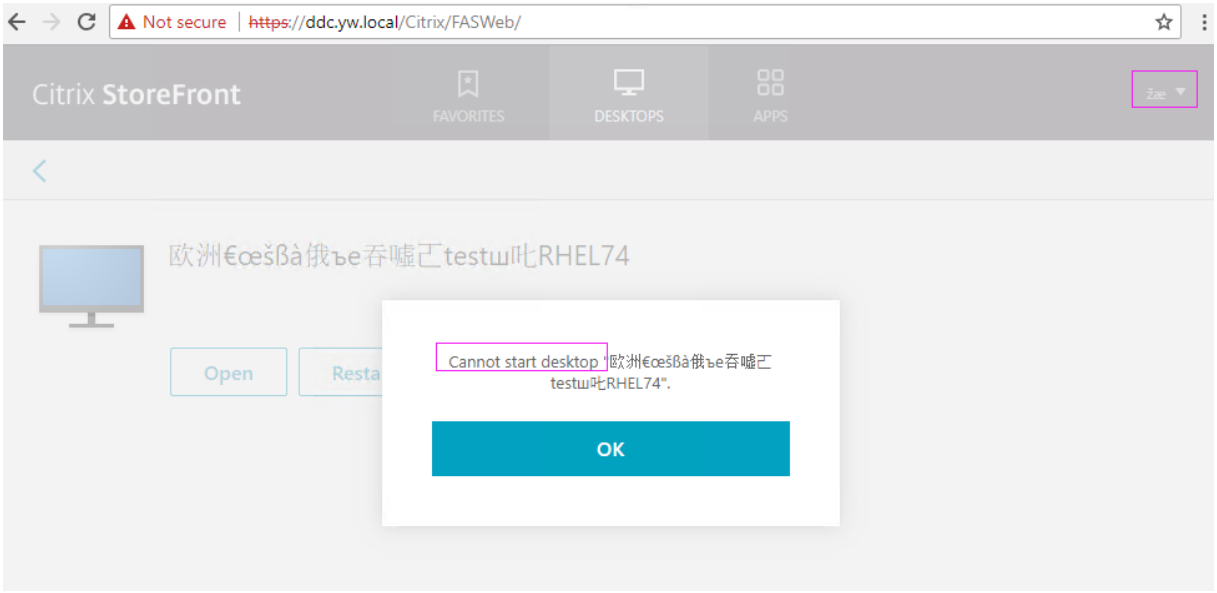
Solution Deploy the ADFS IdP for Federated Authentication Service. For more information, see Federated Authentication Service ADFS deployment.

Related information

- The common FAS deployments are summarized in the Federated Authentication Service architectural overview article.
- "How-to"articles are introduced in the Federated Authentication Service advanced configuration chapter.

Known issue

When FAS is in use, you can fail when trying to launch a published desktop or app session with non-English characters.



Workaround

Right-click **Manage Templates** in the CA tool to change the **Citrix_SmartcardLogon** template from **Build from this Active Directory information** to **Supply in the request**: