Inside Citrix chapter sixteen – The one with Provisioning Services

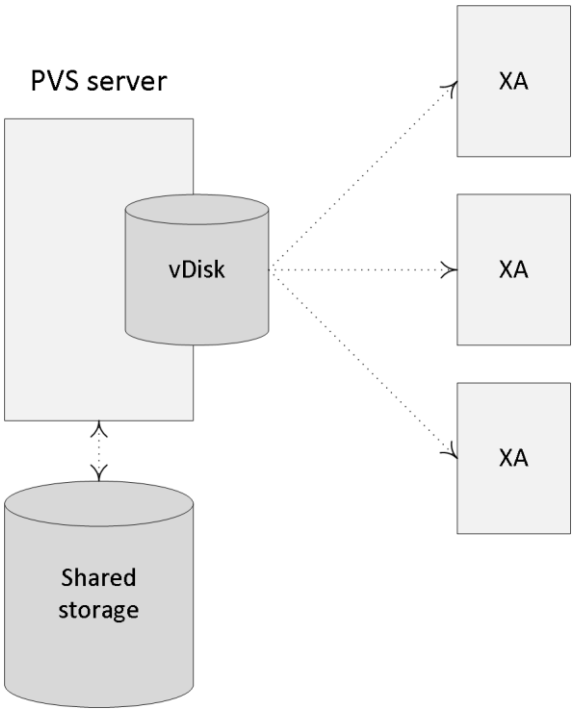
When it comes to delivering the base Operating System within Citrix-orientated environments we have a couple of options to choose from. For one, we can decide to install and manage everything manually, like we are used to doing on our home PCs. Second, you can use an automation tool of some sort (or script it yourself) to install and update your XenApp servers and/or XenDesktop VDI machines, which isn't that uncommon. However, ongoing maintenance will always be a challenge.

And third, we can leverage single image management in the form of either Citrix Machine Creation Services (MCS) and/or Citrix Provisioning Services (PVS). MCS will be discussed in the next chapter, for now I would to spend a minute or two talking about PVS: what it is, how it works, and what some of the advantages are, or can be. Note that this chapter is primarily meant to give you a high-level overview of Citrix Provisioning Services, as I'm not going to cover all the ins and outs that come with setting up, managing and maintaining a PVS Farm; some details will be included, though.

An overview

Provisioning services is based on software streaming technology. Simply put, a single read-only (hence the single image management remark earlier) vDisk or virtual disk will be streamed over the network to multiple so-called target devices, which can be XenApp servers or XenDesktop VDI-based VMs. You will always have at least two provisioning servers for HA purposes, or more depending on the size of your deployment and the number of target devices that need to be serviced.

You could easily provision several hundreds or thousands perhaps of physical and/or virtual machines from 'just' two provisioning servers, although more would be preferred.



PVS overview

Virtual Disk (vDisk)

It all starts with the actual vDisk, or the creation of it. This is the process of configuring and installing all desired software (including applications) and additional components onto a physical and/or virtual machine of choice, like we would normally do when creating a master image; in fact it is not uncommon to use Microsoft SCCM, or a similar solution for this. This machine will be referred to as the Master Target Device.

Next, using specific PVS client software, which will need to be installed on the Master Target Device as well, a vDisk will be created (exported) from the device's local hard drive. vDisks can be stored locally on the provisioning server, on a network file share, or on a shared storage platform accessible by all provisioning servers. Although a great deal of reads will probably be read from cache once the first few machines have booted (read-only vDisk, remember) when dealing with potentially hundreds of machines all trying to access the same vDisk, some consideration regarding the type of storage used to store the vDisk will be needed.

As mentioned vDisks can be assigned to multiple target devices in read-only mode, also referred to as Standard Image mode, meaning that the vDisk will be shared by multiple physical and/virtual devices at the same time. Each machine will have a Write Cache location assigned to it where all the writes to the (read-only) vDisk will be stored, more on these in a minute. However, vDisks can also be assigned in a one-to-one fashion, which is referred to as Private Image Mode, allowing the user to read and write to the vDisk. All changes made will be saved.

vDisk life cycle, Versioning

The life cycle of a vDisk is pretty basic, at least in theory. After creating a vDisk and assigning it to multiple target devices, which is basically step one, it will need to be maintained and updated from time to time. Finally, when no longer in use, a vDisk might need to be retired and taken out of production. While each of these phases have their own specific steps to consider, I would like to briefly highlight the built-in Provisioning Services Versioning technology used for updating and maintenance purposes.

When updating and maintaining a vDisk (in read-only standard mode) using PVS versioning, it will involve creating a new version of the current vDisk, also known as a differencing disk, which will then be linked to the original vDisk. When Versioning is used, this process can be automated, but it can be done manually as well. Next the newly provisioned (differencing) vDisk needs to be assigned to the Master Target Device mentioned earlier and booted in maintenance mode, something which is also easily done from the Versioning console. Once booted, you are free to make your changes and shut down the Master Target Device after you are done.

> **FMA fact**: vDisk updates can be automated and scheduled. This feature supports updates detected and delivered from WSUS and SCCM Electronic Software Delivery servers.

As a final step, the updated vDisk will need to be promoted to either Test or Production. This step can also be automated and scheduled when needed. Beware that when you create multiple 'new' versions, these will all be differencing disks pointing back to the original vDisk. Citrix advises merging the differencing disks back into the base image whenever you have created 3 to a maximum of 5 differencing disks a.k.a. as a chain of differencing disks. This will not only save you some disk space, but will also positively impact performance. Again, you will use the Versioning console to do this. To summarize:

1. From your Provisioning Service management console right-click one of your vDisks in Standard Image Mode and select 'Versions'.
2. In the 'Versions' menu select 'New'.
3. You will see a new version being created with access set to 'Maintenance'.
4. Next you go to the Master Target device, which is used for updating vDisks and change its 'Type' to 'Maintenance'.
5. Assign the vDisk you want to update to the Master Target device.
6. Power on the Master Target device and select the proper vDisk from the boot menu.
7. Login to the Master Target device and make your changes / updates.
8. Run any disk sealing tasks and power off the Master Target device.
9. Again, from your Provisioning Service management console right-click the vDisk in standard image mode and select 'Versions'.
10. Highlight the vDisk you have just updated and promote it to test first (this step is optional, you can also promote it to production right away).
    a. Here you either select 'Immediate' or 'Scheduled'. With 'Immediate' the Target devices will need to be rebooted first before they will be able to use the new updated vDisk. With 'Scheduled' they will need to be rebooted after the scheduled time and/or date.

11. Think about replicating your vDisk files to your other PVS servers. This is something that PVS can help you with as well; it has a built-in mechanism for this.
12. Select one of your Target devices and configure it to boot from the 'Test' vDisk.
13. When testing is done and successful, promote the vDisk to production as per the steps mentioned above.
14. After multiple updates have taken place, also meaning that multiple vDisks / differencing disks have been created it is time to merge.
15. As before, from the PVS management console go into 'Versions' and this time select 'Merge'.
    a. From the 'Merge' menu you have the option to either choose 'Merged Updates', which will merge multiple differencing disks into a new file linked to the original base file, or 'Merge Base'. The latter will merge the original file and all of the differencing disk files into a single new file without any linked files attached.
    b. Also, you need to select if you are merging it into: Maintenance, Test or Production.
16. Again, think about replicating everything to your other PVS servers.
17. If the merged vDisk is still in maintenance or test, promote it to production.
18. Once you are done you can manually delete the older versions. However, do note that once they are deleted you will not be able to revert back to one of these versions, which is also possible using PVS Versioning.

It is not uncommon to see some of the more 'old school' system administrators using the manual (copying and editing multiple vDisks) approach. Also, in larger environments a hybrid deployment is often used as well. Here you would use versioning for your test and development environments and for production you would apply the manual (full vDisk copy) approach, something I would recommend doing as well. In both cases it is important to first get yourself familiarized with the process and steps involved, one at the time.

> **FMA fact**: Be aware that while promoting the version, PVS will actually open up the vDisk and write to it. This it can lead to inconsistencies if you are storing vDisks locally and replication can be complicated.

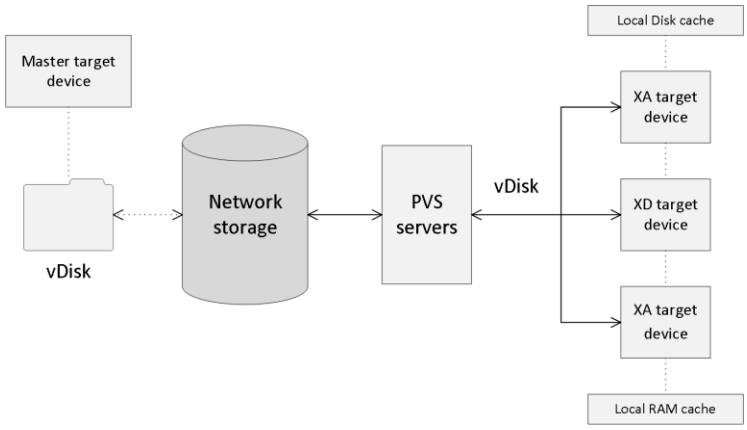
Write cache

When working with 'normal' PCs, or server systems for that matter, we normally don't think about this too much, but when a machine boots and also after it has entered its steady state, it will need to write certain information to base Operating System.

The same applies to applications and other software being used. If the earlier mentioned vDisk is in read-only mode, where will it store those writes? For this they created something that is referred to as the target device's write cache. As with MCS, where all the writes to the base OS are stored on a differencing disk, all writes to a read-only vDisk will be stored in its accompanying write cache.

Each target device will have its own write cache and, depending on the OS used and the type of deployment (RDSH or VDI), its size will vary. What is also worth noting is that when a Standard

Image mode machine gets rebooted, its write cache is cleared; it will start out fresh again. Logically this does not apply to machines in Private Image mode where all changes made during the session will be persistent.



Provisioning Services architecture

Adding write cache to a target device can be done in multiple ways and during the last couple of years Citrix has made some impressive improvement in this area. The following section is meant to provide you with an overview on the various write cache options available.

Cache on device hard drive

The write cache file is stored on the local hard drive of the target device itself; this can be a virtual or physical machine and thus a virtual or physical hard drive as well. No additional software is needed to enable this feature; it is configurable from the PVS management console. As mentioned, the write cache file is only temporary, as it gets refreshed when the machine reboots unless it is set to Private Image Mode. A common and recommended approach when cache in RAM is not possible.

Cache on device hard drive persisted

Here we are basically talking about the same set-up as cache on device hard drive, but this time no changes will be lost when the machine reboots. Note that a different bootstrap file (will be discussed shortly) will be needed for this method. If you want use PVS for persistent desktops, I would recommend to also have a look at the 'Personal vDisks' section over at page 241.

Cache on a server

The write cache file can also be placed on the provisioning server itself, meaning that all writes will be handled by the PVS server. This can increase the overall disk IO and network bandwidth needed, something to consider. It also means that all writes will need to traverse the network back to the PVS server as opposed to writing them on the local hard drive of the target device itself. Personally I wouldn't recommend this setup.

Cache on server persistent

If a vDisk is set to cache on server persistent, each target device that accesses the vDisk automatically has a device-specific, writable disk file created. All changes will be persistent, so after a reboot no changes will be lost. You can also assign a target device to multiple vDisks. To be honest, persistent desktops based on PVS, or MCS (PvD) for that matter isn't a common approach. While personal vDisk might be optional, I would recommend using full VMs instead.

Cache in device RAM

Writes will be temporarily stored in the RAM of the target device. As you can probably imagine, this is by far the fastest method out there today. Besides cache in device RAM with overflow on HDD, this is the preferred way to implement PVS write cache.

Cache in device RAM with overflow on HDD

As soon as the target device runs out of free memory space to store any writes the system will switch to the local hard drive (cache on device hard drive). This method uses the VHDX differencing format. When enough RAM is available, the target device will write to RAM first. When RAM is full, the least recently used blocks of data will be written to the local disk to accommodate newer data on RAM. The amount of RAM specified is the non-paged kernel memory that the target device will consume. The more RAM you assign, the more writes can be stored in RAM and, as a result, the better the overall user experience will be. This has been a very popular approach throughout the past year or so. The more RAM you have available for caching purposes, the better of you will be.

The boot mechanism

When a target device starts it needs to somehow be able to find and contact a provisioning server to eventually stream down the appropriate vDisk. This information is stored in a so-called Bootstrap file named ARDBP32.BIN.

It contains everything the target device needs to know to contact a PVS server so that the streaming process can be initialized. For one, it will contain information about the login servers (with a maximum of four IP addresses), which will be used by the target device to logon to PVS. However, that doesn't mean that one of these (potentially) four servers will also be used for streaming purposes. That is decided in the next phase where the login server will notify the target device about actual streaming server. You will find all the steps involved a few paragraphs down.

The Bootstrap file is delivered through a TFTP server, this also (partly) applies to the alternative BDM (Boot Device Manager) approach, which will be discussed in more detail as we progress. There are some distinct differences between the two.

TFTP

When using TFTP the target device needs to know how and where it can find the TFTP server to download the Bootstrap file before being able to contact a PVS server. Secondly, since this is a critical step you want to make sure that the TFTP server isn't a single point of failure, meaning that you would like to implement some form of high availability. Throughout the next section I'll list some of the options you have in achieving this (delivering the Bootstrap and HA).

> **FMA fact**: Provisioning Services has its own built-in TFTP server. However, you are free to use whatever you prefer.

One of the most popular approaches in delivering the TFTP server address to your target devices is through DHCP, but there are other options as well:

- You can use DHCP option 66 to enter the TFTP's server address. However, note that you will only be able to fill in one server address: no HA here.
- You can use DHCP option 66 combined with DNS round robin. Here you put in a hostname that has multiple A (host) records (PVS servers) in DNS, which will then be rotated in a round robin fashion. Has its flaws (not all PXE clients are able to handle multiple host entries for example) but works.
- Using PXE (broadcast) services. PVS also offers the ability to configure a PXE service when initially configuring Provisioning Services. When a target device boots it will send out a broadcast message to find a suitable PXE server. This way, if one of the two PVS servers (also hosting the PXE services) is down, the other one will respond and deliver the Bootstrap file. Instant HA without needing to touch DHCP: a very popular approach. Again, note that there are various PXE clients out there and that they do not all offer the exact same functionalities.
- And finally the Citrix NetScaler. Using the NetScaler you have multiple options in making your TFTP server highly available.

BDM

There are actually two different methods to make use of the Boot Device Manager. Let's start with Provisioning Services. PVS offers a quick wizard, which will generate a relatively small .ISO file (around 300KB). Next you configure your target devices to boot from this .ISO file, using their CDROM/DVD players, for example, by specifying its (shared) network location.

This method uses a two-stage boot process where the PVS location will be hardcoded into the bootstrap generated by the BDM. The rest of the information (like the PVS device drivers) is downloaded from the PVS server using a proprietary download protocol based on TFTP (UDP, port Nr. 6969). Here, TFTP will still be used. This works with virtual as well as physical machines.

As of XenDesktop version 7.0, when using the XenDesktop setup wizard we can create and assign a small BDM hard disk partition, which will be attached to the virtual machine as a separate virtual disk. Using this method the above-mentioned two-stage approach is no longer needed because the partition already contains all the PVS drivers.

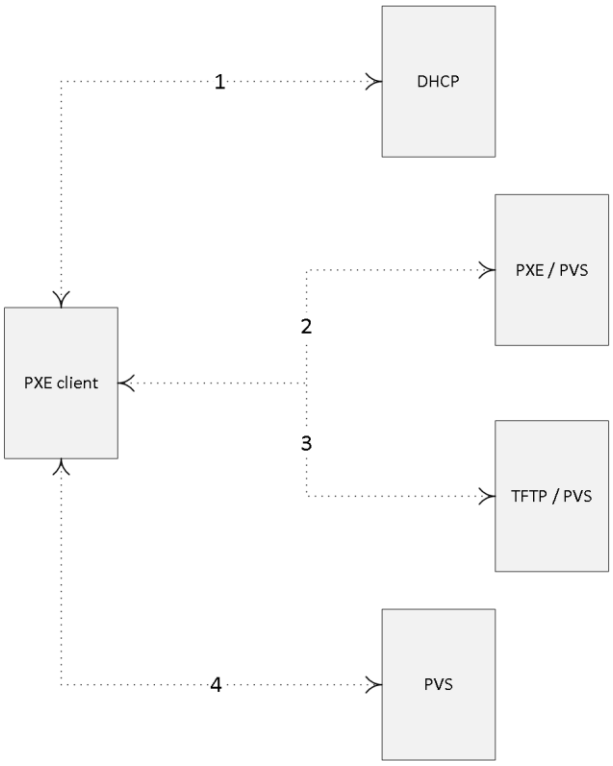
This way all the information needed, as discussed previously, will be directly available without the need for PXE, TFTP and/or DHCP. Advised for virtual machines only.

> **FMA Fact**: As an added advantage, using the BDM method will also decrease boot times by around 5 to 10 seconds since we don't have to wait for PXE and TFTP.

PVS logon and boot steps summarized

When we sum up the whole process from start to finish it comes down to this:

- First the target devices boots and acquires an IP address.
- The target device first identifies a TFTP server.
- Next the Bootstrap file will be downloaded and the target device will boot from it.
- The target device will contact and log onto one of the PVS servers.
- The logon server will notify the target device about the streaming server.
- Target device starts streaming the vDisk from the PVS server.



PVS logon and boot process

High Availability

When dealing with highly available PVS environments, there are multiple components to consider. First of all, PVS needs an SQL database, which somehow will need to be made highly available, secondly, you will at least install and configure two PVS servers and configure / enable HA within the PVS software itself, which is done from the management console, and last but not least, the storage where your vDisks reside also needs to be shared, or made highly available. And of course when making use of TFTP for Bootstrap delivery, that will need to be made HA as well, as we have seen in the previous section.

As a side note, Multiple PVS servers can also be used for load-balancing purposes with regard to the streaming traffic they handle – active / active.

> **FMA fact**: When vDisks are stored locally on the Provisioning Servers, you will need to implement some sort of replication mechanism so that all PVS servers will be able to offer the exact same vDisks. This can also be done manually from the PVS management console. Recommended automation methods include both DFS-R and Robocopy.

When it comes to the SQL database you have multiple options:

- PVS offers a built-in offline database support mechanism. It uses a snapshot of the database when the database becomes unavailable. Note that this feature is disabled by default.
- Database mirroring, SQL Clustering and SQL Always On are all valid options as well.

To conclude, here is a shortlist of components to consider when implementing HA for PVS:

- The Provisioning Services SQL database.
- Your physical or virtual Provisioning Servers.
- Storage used to store your vDisk files, PVS servers and/or target devices.
- The TFTP servers for Bootstrap delivery.

Provisioning wizards

Once you have set up your base Provisioning Services infrastructure, your Master Target Device, or multiple, and configured a vDisk, the time has finally come to (auto-) provision multiple virtual target devices using the streamed VM set-up wizard or the XenDesktop set-up wizard. Ultimately, the devices, which end up using one or multiple of your vDisks, are referred to as device collections: also see the next section for more information with regard to some of the basic terminology used with Provisioning Services. Of course PVS can be used with physical machines just as easily; the way this is set up is just slightly different.

The streamed VM set-up wizard

This wizard is accessed directly from the PVS management console. With the streamed wizard we can create multiple virtual machines, or target devices, to where the vDisks will be steamed (a preconfigured template VM will be needed for this): it will create the target device objects within

the appropriate device collection (a collection needs to present) and finally it will assign a vDisk in Standard Image Mode to the virtual machines. It goes without saying that the vDisk will have to be made beforehand as well.

> **FMA fact**: The streamed wizard supports the following Hypervisors: XenServer, Hyper-V through SCVMM and ESX through vCenter.

During the wizard session you will be able to select the Hypervisor to connect to the VM template on which the newly provisioned VMs will be based, the device collection to where the VMs will be added during creation, choose the number of VMs to create, the number of CPUs and the amount of memory each VM will be assigned, and finally you have the option to let the system create new or add-in existing Active Directory computer accounts.

Make sure to read through all of the prerequisites before you start using the wizard. I would suggest the Citrix E-docs website as a good starting point.

The XenDesktop set-up wizard

To be able to use the XenDesktop wizard you need to make sure that your Delivery Controllers and Provisioning Servers are both on the same version. The XenDesktop set-up wizard, just as with the streamed VM set-up wizard, creates VMs on a XenDesktop hosted Hypervisor (Host Connection) making use of a preconfigured template VM machine.

It will also create target devices within a new or existing device collection, which will take on the name of the corresponding XenDesktop Catalog. It will assign a Standard Image vDisk to the VMs within the device collection, and at the same time it will add all virtual desktops to the XenDesktop Machine Catalog within Studio. Once you have gone through all the steps in the wizard, this in a nutshell is what happens next:

- First, if needed, it will create a new XenDesktop Machine Catalog.
- Create VMs on a host's Hypervisor using the preconfigured machine template.
- Create BDM partitions, if specified.
- If using a Streamed with personal vDisk Catalog, create a personal vDisk, and then attach the personal vDisk to the VM. This will be discussed in the next session; personal vDisks are somewhat special.
- Create a write cache disk of the specified size (make sure to check out all prerequisites and write cache considerations beforehand).
- Create Provisioning Services target devices, and then assign the selected vDisk to those devices.
- Add the target devices to the selected Provisioning Services Collection.
- Add the VMs to the XenDesktop Catalog.
- Boot each VM to format the newly created write cache disk.

Personal vDisks

While using Provisioning Services (or Machine Creation Services) to configure persistent desktops, either through persistent write cache or by attaching a Personal vDisk, isn't a very popular approach within larger enterprises, there are tons of smaller companies who are interested in this type of setup. If you talk to Citrix, they would normally advise you to use PvDs for no more then a 100 to max 125 machines, primarily to prevent things from getting too complicated. If more are needed, fully cloned VMs are preferably used instead. Also, on my website I get a lot of questions around PvDs and related technology. That's why I decided to include the subject in here as well.

A Personal vDisk, or PvD in short, offers a way for users to store their changes when working on a virtual, pooled static machine (the PvD is assigned or attached to a virtual machine and then the user is assigned a virtual machine on first use). When PVS is used, streaming is only possible to virtual machines; streaming to physical is not supported. PvD technology can be used with PVS as well as MCS.

> **FMA fact**: Personal vDisks can only be assigned to an desktop Operating System; server OSs are not supported at this time.

An overview

With Personal vDisks we still use one base (master) image just like before but we now get an extra Personal vDisk attached to our VM on which all our 'personal' changes, will be stored. These will include all file level and registry changes including but not limited to: installed or streamed applications provisioned by SCCM, App-V (think cache) or XenApp but also things like: desktop wallpapers, Start menu settings, favorites etc. It basically stores all changes made under C:\Users as far as the user profile is concerned and everything else when it comes to applications that get installed / updated.

Split in two

The PvD VHDs, by default, are split in two when it comes to storage allocation for personal (profile-related) changes and application installs and/or updates. If your PvD is 10 GB in size it will allocate 5 GB for profile / personal storage and the other 5 GB for application installs / updates etc. This can be changed (Registry setting) into 70 / 30, 90 / 10, or 99 / 1 even; give this some thought and adjust accordingly. When used in combination with user profile management and/or folder redirection, there's not that much left and it would be a waste of space.

When an administrator needs or wants to edit the underlying base (master) image: no problem! He or she simply updates or installs an application, service pack, security update or whatever needs fixing and the Personal vDisk will take it from there. The user will see all the changes he or she made (stored on the PvD) in conjunction with the base (master) image even if it's being updated live, although the administrator must still roll out these changes to the end-users from Studio like before, so the VM will still need a reboot. It allows all of the user changes to persist over the base image changes.

The Personal vDisk communicates with the XenDesktop Personal vDisk agent, which is installed on the base (master) image during (Catalog) creation. This agent tracks what's installed and available on the base (master) image versus what's installed and changed on the Personal vDisk and will blend these two together once the base (master) image has been updated and rolled out, or applied, to the end-user. This way we get the persistence of dedicated desktops together with the management advantages of pooled desktops.

Some characteristics

If a conflict exists, for example, when a user installs the same application on his PvD as the Administrator does on the base (master) image, then the system will make a note of this and remove the software installed by the user, keeping the PvD as small in size as possible. Note that this is a default setting and is customisable the way you see fit. PvDs can be resized afterwards. Their default size and location are selected during the Catalog creation wizard (selecting MCS as the provisioning mechanism) from XenDesktop Studio or the PVS set-up wizard when PVS is applied.

They end up smaller in size than the 'normal' provisioned differencing disks created with dedicated desktops.

Thin provisioning is supported, so PvDs can be attached to any storage target as defined within your Hypervisor: this means that PvDs can be on different (storage) locations than your actual VMs, enabling you to spread out the IOPS load.

A PvD can be used as a simple profile management solution for small-sized environments, although Citrix recommends using a separate profile management solution alongside. It's compatible with almost all profile management solutions out there.

PvDs allow for easier management but with the flexibility of dedicated desktops. They are 100% persistent with pooled VDI storage and management. As mentioned earlier, PvDs are compatible with most PC life cycle management systems like SCCM, and application virtualisation solutions like Citrix XenApp. And just so you know, PvDs are also available in VDI-in-a-Box, which is still supported by Citrix.

Basic Provisioning Services terminology

Below I will go over some of the terminology that goes with configuring, managing and maintaining Citrix Provisioning Services, some of which you have already come across during the previous sections.

- Farm: This represents the top level of a PVS infrastructure. All sites within a Farm share that farms Microsoft SQL database. A Farm also includes a Citrix License Server, local or network shared storage, and collections of target devices.
- Site: A Site represents a logical grouping of all Provisioning Servers, Device Collections, target devices and storage.

- Stores: This is where you physically store your vDisk files. This can be on either local storage, on the PVS server (s) itself or shared storage in the form of a SAN. When a vDisk is created from the PVS management console you will need to assign it to a store.
- Device Collections: They enable you to manage a large number of devices as a logical group. They simplify administration since administrative tasks can be executed at Device Collection level instead of on a per device basis. A Collection can also represent a physical location or a specific subnet range.
- Target Devices: All devices, both virtual as well as physical, that get a vDisk streamed over the network. The device used to create and maintain the actual vDisk is referred to as a Master Target Device.

Key takeaways

- Provisioning Services streams a base image over the network down to either virtual or physical machines.
- It works for both desktop as well as server Operating Systems.
- A device using a vDisk is also referred to as a Target Device.
- The machine used to create and maintain the vDisk is referred to as the Master Target Device.
- Target Devices are managed using Device Collections.
- Dozens, hundreds or thousands of Target Devices can share a single vDisk.
- The life cycle of a vDisk consists of creation, deployment, maintenance and finally retirement. For this we can leverage the built-in PVS Versioning mechanism.
- Give your write cache sizing and location some consideration: you will be glad that you did.
- Although PVS vDisks can also be streamed in Private Mode, where any changes made to the vDisk will be saved, this isn't a very popular approach.
- Provisioning Services can seem complicated and challenging at first. Take your time and take it step by step, you will be fine. There are many excellent resources out there to help you on your way.
- Make sure to make your PVS infrastructure highly available.
- While PvDs have their use, apply them wisely: it's not for everyone. And while this may be somewhat off topic, in many cases where VDI is being considered, RDSH might make more sense.
- Make sure to check out CTX117372 for some best practices around PVS networking.
- While in the past it was always considered a best practice to use physical machines for your Provisioning Servers, today virtual machines are almost always recommend by Citrix. This has a lot to do with the enhancements around standard networking.
- The same applies to isolating your PVS traffic, again mainly due to the advancements that have been made on the networking and virtualisation side of things during the last couple of years. Keep it simple. One of the main reasons why isolation might still make sense is because of security considerations.
- CTX131611 lists a bunch of Known Hardware Related Provisioning Services Issues.
- Check out CTX124185 for best practices around antivirus on PVS vDisks.