Inside Citrix chapter seventeen – The one with Machine Creation Services

Machine Creation Services is simple to operate: it is integrated right into XenDesktop and you don't have to build and maintain a separate infrastructure like with PVS. Also, while MCS was originally developed and meant for the provisioning of VDI-based VMs, as of XenDesktop 7 it now also supports the provisioning of server Operating Systems. Never mind if it's on-premises or in the cloud, since MCS can do both.

> **FMA fact**: While I use the term 'provisioning' do not confuse the provisioning of machines with MCS with that of PVS (see previous chapter). In general, provisioning means providing or making something available. A term widely used in a variety of concepts within IT.

Breakdown

Next to Provisioning Services, MCS is the second option that we have regarding (automated) desktop image delivery and single image management within XenDesktop. MCS is designed in such a way that it will communicate directly with the Hypervisors Application Program Interface, or API to take care of things like VM creation, the starting and stopping of VMs (power management), delete VMs and so on.

It supports Microsoft's Hyper-V using SCVMM, VMware's vSphere through vCenter and Citrix's XenServer by directly communicating with the Pool master and/or XenCenter.

As with PVS, it all starts with a master VM, template machine, golden image or whatever you would like to call it. A master VM is nothing more than a virtual machine with everything installed; applications, antivirus, patches etc. and configured exactly the way you want to present it to your users.

This will also include the amount of compute (vCPU, Memory, Disk space etc.) assigned to the VM. Next, when you create a new Machine Catalog from Studio (based on MCS technology) you will be asked to select this master VM, which will then serve as the base image from where all other VMs will be (automatically) provisioned.

During these steps you will also have the option to change some of the earlier made configuration choices with regard to the amount of compute assigned to VM (a more detailed overview regarding the steps involved can be found on page 251). MCS is based on Differencing Disk technology, which is similar to that of Linked Clones, used by VMware for example. Using MCS we can provision three different types of desktops (Catalog), see the table on the next page for an overview.

MCS desktop types

Desktop type	Description
Pooled-Random	A Desktop will be assigned randomly when a user logs on. When they logoff the desktop will become free for use and any changes made will be lost completely
Pooled-Static	A desktop will be permanently assigned to a user at logon. It will stay with the user even after logoff. Any changes made will be discarded during reboot / logoff
Dedicated	A desktop will be permanently assigned to a user at logon. It will stay with the user even after logoff. Any changes made will be saved to the VM no matter how many times it gets rebooted / refreshed

When provisioning new machines using the Machine Catalog wizard from Studio, from a technical point of view it will work like this: MCS will first take a snapshot of your master VM (template machine) or you can take one manually, which has the added advantage that you can name it yourself. Next, this snapshot is consolidated (merged) and a temporary virtual machine will be created.

This virtual machine will then boot so that certain tasks can be taken care of (DHCP, KMS etc…) before it will be copied over to all datastores known as part of your VM deployment. All VMs created as part of the provisioning process will consist of a Differencing Disk and an Identity Disk: both will be attached to the VM (how this is handled technically slightly differs per Hypervisor).

The Differencing Disk is meant to store all changes made to the VM (it functions as a write cache) while the Identify Disk is meant to give the VM its own identity used within Active Directory. During this phase MCS will also take care of machine account creation in Active Directory. And finally the actual VMs, based on the information provided during the wizard will be created / provisioned on your underlying Host Connection.

1. Create / configure master VM:
    a. Install Operating System.
    b. Install VM tools.
    c. Join Domain.
    d. Install VDA.
    e. Install applications.
    f. Anti-virus scan.
    g. Other…
2. Create Machine Catalog in Studio:
    a. Select Operating System type (server, desktop, remote PC).
    b. Select machine management (PVS, MCS, other service or technology.
    c. Desktop experience (random or static assigned desktops).
    d. Select master VM.
    e. VM config (compute: vCPU, RAM, Disk space), # of machines.
    f. AD location for machine accounts plus the account-naming scheme.
    g. Summary, hit finish.
3. Automated (or manual) snapshot creation.
4. Temporary VM is created. This will boot so that certain tasks like DHCP, KMS etc. can be taken care of first.
5. Full copy of snapshot.
6. Snapshot is copied to accompanying data stores.
7. AD identities are generated.
8. Desktops are added to Active Directory.
9. VMs are provisioned and disks get attached.
10. Depending on the amount of machines and the type of storage platform used this can take up to several hours. Done!

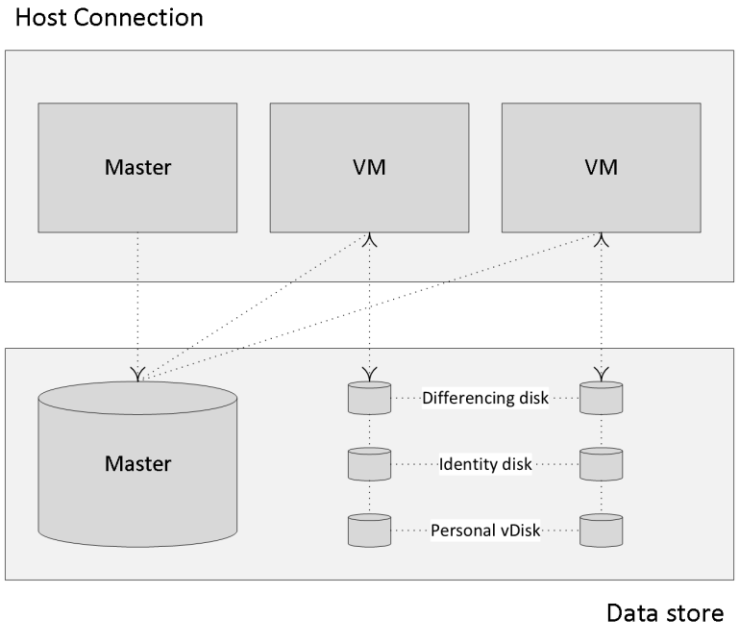
Why this matters

Each time you create a new master VM, or update an existing one, for that matter, this process repeats itself, meaning that if you have multiple master VMs, let's say two for VDI (Windows 7 and 8) and two for RDSH (Windows Server 2008 R2 and 2012 R2), the system will create a snapshot of every master VM, four in this example, which will then need to be copied over to all of the datastores as part of your VDI / RDSH deployment.

As mentioned, when updating an existing master image, the same process is repeated. The system will basically treat the updated master VM as a new master VM and, as such, a snapshot will again be created and copied over to the accompanying datastores accordingly. For obvious reasons, a few questions go with this.

- How many master virtual machines will you be managing – meaning the different types of virtual machines, Windows 7, 8 and so on?
- How many datastores will you be using?
- How many times, per week, month, year will your master VM(s) need to be updated? However, this will always be a hard question to answer.
- Does your storage platform support thin provisioning? If not, then every provisioned VM will be as big as the master VM it is based on.

By answering these types of questions you will have at least an indication of the amount of storage needed and the administrative overhead that comes with maintaining your virtual infrastructure based on MCS.



MCS Differencing Disks

Your workloads

As we all know, virtualising 100% of your application workload is not going to happen anytime soon. And even when a large part is virtualised, using App-V, for example, applications might be pre-cached, meaning you will still need to (potentially) 'break open' the base image they reside on if the application itself needs to be updated, patched etc. in some way. Of course the same applies when applications and/or plug-ins are installed in the base image.

Now imagine managing 4 (or more) different base images (master VMs), all with a relatively intensive maintenance cycle perhaps caused by a bunch of 'home-made' applications that need to be updated on a weekly basis.

Every week these images, or master VMs, need to be updated at least once, meaning a new snapshot per master image that will need to be copied down to all the participating datastores for each image type, taking up CPU resources, causing a peak in IO and network resource usage. This alone can be a time-consuming process depending on the number and size of your images and the number of datastores involved.

> **FMA fact**: Today technologies like application layering and containerisation can help us overcome most of these application-related issues; however, the general adoption of these kinds of technologies and products will still take some time.

Rollbacks

Rollbacks are treated the same way (there is a rollback option available in Studio specific to MCS) in that they are seen as yet another image that is different from the one your VMs currently rely on. This means that the whole process as described previously will repeat itself.

Some more considerations

When a master VM / image is updated, copied over to all datastores etc. and assigned to the appropriate VMs, these VMs will first need to be rebooted to be able to make use of the new or updated master VM. While this may sound as a relatively simple process there are some things you need to take into consideration.

- When rebooting a couple of hundred virtual machines (or more) this can have a potential negative impact on your underlying storage platform. Also referred to as a boot storm. This is not something you want to risk during work hours.
- Some companies have very strict policies when it comes to idle and disconnect sessions. There might still be some user sessions in a disconnect state at the time you want to reboot certain VMs, but when the company policy states that users may not be logged off (forced) unless they do so themselves or automatically when the configured disconnect policy kicks in, you will have to reschedule or reconsider your planned approach. Not that uncommon, trust me.
- All this might also interfere with other processes that might be active during the night.
- And while not on topic, all this applies to Provisioning Services as well.

Storage implications

I already mentioned a couple of things to consider with regard to your underlying storage platform when using MCS, but I am pretty sure I can come up with few more, so let's have another look.

- MCS is, or at least can be, storage-intensive with regard to the (read) IOPS needed. On average it will need around 1.6 times more IOPS when compared to PVS: again, mainly read traffic from the master VM as mentioned earlier. However… and this is often where the confusion starts, the 1.6 number is based on the overall average, meaning that it also takes into account boot and logon storms (that's also why they are mainly read IOPS). If we primarily focus on the so-called steady state IOPS, then it's closer to 1.2: a big difference, right?
- While the above shouldn't be too big an issue with all modern storage technologies available today, it's still something you need to consider.
- The ability to cache reads is a very welcome storage feature! Don't be surprised if you end up hitting a read cache ratio of 75% of higher, boosting overall performance.
- Consider having a look at IntelliCache, a technology built into XenServer, when deploying non-persistent VMs on NFS-based storage.
- Clustered Shared Volumes Cache (Hyper-V) is also (very) helpful. In fact, it would

probably outperform Provisioning Services all together, with the exception of cache in RAM that is.

- When looking up IOPS recommendations based on Citrix best practices, be aware that these are (almost) exclusively based on steady state operations.
- A medium workload (based on the medium login VSI workload) running on a Windows 7 or 8 virtual machine (provisioned with MCS) will need around 12 IOPS during its steady state with a read/write ratio of 20:80. During boot, these numbers will be the other way around: 80:20.
- Running the same workload on a Server 2012 R2 virtual machine will lead to 9 IOPS during its steady state with a read/write ratio of 20:80. Note that this is on a per user basis. And again, during boot this will probably be closer to 80:20 read/write.
- Collect as many (IOPS / workload-specific) data as possible and consult with one or multiple storage administrators. See what they think.
- Remember that there is more to IOPS than just the read/write ratios. Also consider the different types of IOPS like random and sequential, the configured storage block sizes used, and the actual throughput available.
- Often multiple datastores are created to spread the overall IOPS load on the underlying storage platform. Think this through. Remember that each time a new master VM is created or one is updated, it will need to be copied over to all datastores as part of your VM deployment.
- Load testing can provide us with some useful and helpful numbers with regard to performance and scale. However, don't lose track of any potential resource-intensive applications that might not be included during some of the standard baseline tests. You don't want to run into any surprises when going live, do you? Include them.
- Whenever possible, try to scale for peaks, meaning boot, logon and logoff storms.
- If needed, schedule and pre-boot multiple VDI VMs, as well as any virtual XenApp servers, before the majority of your users start their day.
- Try to avoid overcommitting your host's compute resources as much as possible. And while overcommitting your physical CPUs / cores is fine, never overcommit memory: there is no need.
- Failing over VMs while using MCS isn't a problem, but if you would like to move the accompanying virtual machine disks / files as well, using Storage vMotion, Live Storage Migration and/or Storage XenMotion, you are out of luck. This is not supported.
- The renaming of network connections and Data Stores is also not possible.
- There is a strong dependency between the VM and the Disk ID / datastore it resides on (information stored in the Central Site database): once broken or interrupted, your VM will not be able to boot properly.
- Consider implementing storage technologies and hyper-converged solutions like Nutanix, Atlantis, VSAN, VPLEX (there are plenty more out there), enabling you to move around your MCS-based VMs, including their disks, either automated or manually, while maintaining the VM Disk ID dependency using and combining features like Shadow Clones, Data Locality, Data synchronisations and so on. SDS is key.
- The ability to thin provisioning the earlier discussed differencing disks would be preferred. This will initially save you a lot of disk space, and if your environment isn't that big (< 1000), re-provisioning your VMs once or twice a month might be optional as

well. You might be able to get up to a few hundred machines per hour. This is something that your storage platform will need to support.

- The same applies to compression and de-duplication.
- I/O offload for writes and a caching mechanism of some sort for reads will greatly enhance the overall user experience.
- Monitor the growth of your differencing disks and size your storage platform accordingly. Do not forget to include the duplicate images of your master VM copied over to each datastore when it comes to free GBs needed. When using Hyper-V a differencing disk cannot be marked as exposable. As a result, when the VM is rebooted, the disk will be detached, a new disk will be attached and the old one will be deleted.
- Also, when an image is updated, at least temporarily, there will be two full images / master VMs residing in each datastore: do not forget to include these into your GB calculations as well.
- As soon as all VMs within the datastore are using the new or updated image, the 'old' one will be deleted automatically. By default, this will happen after a time period of six hours, but this is configurable through PowerShell.
- Think about a reboot schedule for your XenApp servers. Each reboot will 'refresh' the differencing disk, making them start from zero. Reboot your VDI VMs on a daily basis or make them automatically reboot once a user logs off.
- Consider your underlying and supported Hypervisor when using MCS. You can use NFS for XenServer and ESXi or Clustered Shared Volumes for Hyper-V. However, at this time no thin provisioning on XenServer with block-based storage. Do note that some interesting new developments surrounding block-based storage in the latest XenServer Tech Preview, a.k.a. Dundee are on their way.
- MCS combined with Hyper-V local storage is also optional. Simply configure a cluster without shared storage. A copy of your master image will be placed on one of your local drives, C:\, D:\ etc.
- Citrix used to recommend using NFS exclusively to go with MCS, but that was more geared towards the inability of XenServer to thin provision disks based on block-based storage than anything else. While NFS will be more straightforward to configure and maintain, block-based storage with MCS is also supported and used in many production environments as well.

Updating

When you update the master VM/image of a persistent virtual machine, or multiple, only newly provisioned persistent VMs will be able to use the updated master image. All of the existing persistent virtual machines will continue to rely on the 'old' master VM. When dealing with non-persistent VMs this works different. Once the new or updated image is assigned and the VMs reboot, the old image will be discarded (and eventually deleted once it is no longer in use) and the new one will be used from then on.

Key takeaways

- MCS is considered to be easy. It is managed and configured directly from Studio and you do not need any additional infrastructural components as you do with PVS.
- MCS is based on differencing disks technology.
- Your base or golden image will be copied over to all datastores, which are part of the virtual machine deployment. Take this into account when thinking about your storage needs.
- When application virtualisation is not an option, often forcing you to install applications into your base image, think about using application layering as an alternative.
- When using MCS, rollbacks are treated the same way as a new or updated base image: they will again need to be copied over to all datastores involved. Note that in some cases the previous image might still be in use by some machines. If so, than no full copy will be needed.
- Give your Idle and Disconnect session policies some thought. This will make it easier to reboot your machines during night-time, depending on company policy, of course.
- Go over the earlier mentioned list of storage implications a couple of times: there is a lot to consider.