High Availability for

Citrix XenServer

Enhancing XenServer Fault Tolerance with High Availability

www.citrix.com

Contents

Introduction

High Availability (HA) is a designed approach to ensure that a prearranged optimal level of operational performance occurs within a specific environment.

For server virtualization, and XenServer in particular, HA ensures that mission and business critical virtual machines (VMs) are continuously operating within a resource pool. To accomplish this XenServer's HA feature:

- Reliably detects a host failure
- Computes a failure plan to deal with rapid recovery

One of the most significant challenges in implementing HA is to reliably detect when a server has failed, instead of just being temporarily unreachable. If HA mistakenly assumes that a master has failed, and elects a new master, unpredictable behavior would occur when the first master rejoins the pool.

Another significant challenge is to detect and handle situations in which network problems may cause hosts in a resource pool to lose communication with each other but remain in contact with their shared storage device. In this scenario, we need to ensure that only one host is able to access and write to that shared storage device in order to avoid simultaneous writes.

When HA is enabled, XenServer continually monitors the health of the hosts in a pool. The HA mechanism automatically moves protected VMs to a healthy host if the current VM host fails. Additionally, if the host that fails is the master, HA selects another host to take over the master role automatically, so that you can continue to manage the XenServer pool.

To absolutely guarantee that a host is unreachable, a resource pool configured for high-availability uses several heartbeat mechanisms to regularly check up on hosts. These heartbeats go through both the storage interfaces (to the Heartbeat SR) and the networking interfaces (over the management interfaces). Both of these heartbeat routes can be multi-homed for additional resilience to prevent false positives.

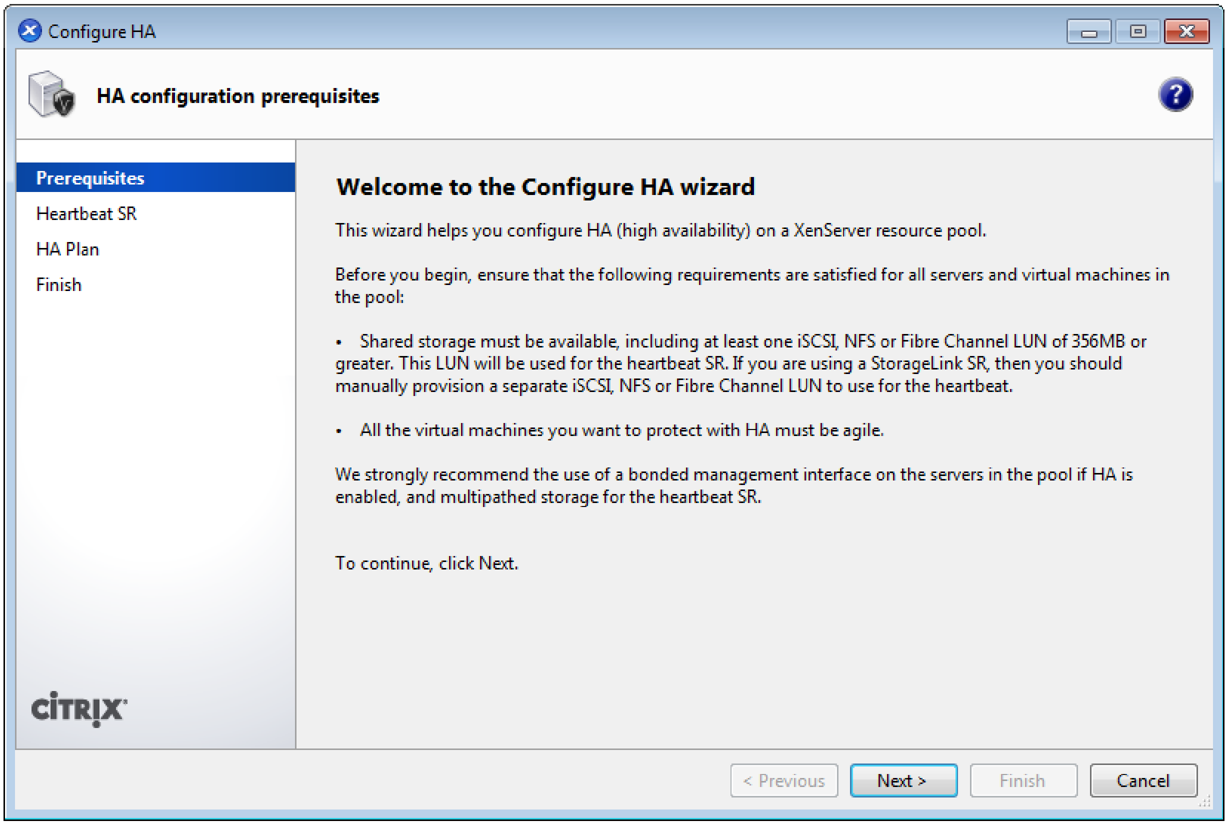
XenServer dynamically maintains a failover plan which details what to do if a set of hosts in a pool fail at any given time. An important concept to understand is the host failures to tolerate value, which is defined as part of HA configuration. This determines the number of failures that is allowed without any loss of service. For example, if a resource pool consisted of 16 hosts, and the tolerated failures is set to 3, the pool calculates a failover plan that allows for any 3 hosts to fail and still be able to restart VMs on other hosts. If a plan cannot be found, then the pool is considered to be overcommitted. The plan is dynamically recalculated based on VM lifecycle operations and movement. Alerts are sent through XenCenter or e-mail if changes such as addition of new VMs to the pool cause your pool to become overcommitted.

Heartbeating for availability

XenServer solves the problem of reliably detect when a server has failed, instead of just being temporarily unreachable and detecting handle situations in which network problems may cause hosts in a resource pool to lose communication with each other but remain in contact with their shared storage device by using the following two mechanisms:

1. storage heartbeat
2. network heartbeat

To enable HA in a pool an iSCSI, Fibre Channel or NFS storage repository (SR) must be used as the heartbeat SR. XenServer automatically creates small virtual disks in this SR. The first disk is used by every physical host in the resource pool as a shared quorum disk. Each host allocates itself a unique block in the shared disk and regularly writes to the block to indicate that it is alive.



Welcome to the Configure HA wizard

This wizard helps you configure HA (high availability) on a XenServer resource pool.

Before you begin, ensure that the following requirements are satisfied for all servers and virtual machines in the pool:

- Shared storage must be available, including at least one iSCSI, NFS or Fibre Channel LUN of 356MB or greater. This LUN will be used for the heartbeat SR. If you are using a StorageLink SR, then you should manually provision a separate iSCSI, NFS or Fibre Channel LUN to use for the heartbeat.
- All the virtual machines you want to protect with HA must be agile.

We strongly recommend the use of a bonded management interface on the servers in the pool if HA is enabled, and multipathed storage for the heartbeat SR.

To continue, click Next.

Figure 1: Configure HA Wizard

When the HA mechanism is enabled, all hosts exchange data over both network and storage channels, indicating which hosts they can see over both channels, which indicates which I/O paths are working and which are not. This information is exchanged until consensus is reached about the hosts and storage in the pool. When this happens, the HA functionality is ready to protect the pool.

This process may take a few minutes for larger pools, but is only required when HA is first enabled.

Once HA is active, each host regularly writes storage updates to the heartbeat virtual disk, and network packets over the management interface. It is vital to ensure that network adapters are bonded for resilience, and that storage interfaces are using dynamic multipathing where supported. This ensures that any single adapter or wiring failure does not result in any availability issues.

The worst-case scenario for HA is the situation where a host is thought to be off-line but is actually still writing to the shared storage, because this can result in corruption of persistent data. To prevent this situation without requiring active power strip controls, XenServer employs hypervisor-level fencing. This is a Xen modification which hard-powers off the host at a very low-level if it does not hear regularly from a watchdog process running in the control domain. Because it is implemented at a very low-level, this also protects the storage in the case where the control domain becomes unresponsive for some reason.

Hosts will self-fence (power off and restart) in the event of any heartbeat failure except in the following cases:

- When the storage heartbeat is present for all hosts but the network has partitioned (so that there are now two groups of hosts). In this case, all of the hosts which are members of the largest network partition stay running, and the hosts in the smaller network partition self-fence. The assumption here is that the network outage has isolated the VMs, and that they must be restarted on a host with working networking. If the network partitions are exactly the same size, then only one of them will self-fence.
- When the storage heartbeat goes away but the network heartbeat remains. In this case, the hosts check to see if they can contact all other hosts over the network. If they can, the hosts remain running on the assumption that the storage heartbeat server has gone down. This does not compromise VM safety, but any network problems will result in fencing because that would mean both heartbeats have disappeared.

Planning for failure

The heartbeat system gives XenServer reliable notification of host failure, which allows XenServer HA to perform another important role -- planning for failure.

A resource pool consists of multiple physical hosts, each with potentially different amounts of host memory and a different number of running VMs. To ensure that no single host failure results in the VMs on that host becoming unrestartable (for example, because of insufficient memory on any other host), the XenServer pool dynamically computes a failure plan which calculates the actions that would be taken on any host failure.

However, a single host failure plan does not cover more advanced cases, such as network partitions, where multiple hosts may become unreachable. This is why XenServer dynamically calculates the maximum number of server in a pool that can fail (the "number of host failures to tolerate" or nhtol) before the pool can no longer guarantee that all protected VMs in the pool will continue to run.

A brute force search of all possible failures across all hosts across all VMs would take exponentially increasing time to complete on larger and larger pools. XenServer applies the following heuristics to ensure the plan is computed in a reasonably small time:

- For up to three host failures, XenServer performs a comprehensive search which tries almost all permutations.

  This covers many unlikely scenarios, for example, the situation where hosts or VMs have very different amounts of memory (for example, 4GB on one, 128GB on another). This produces a very robust plan.

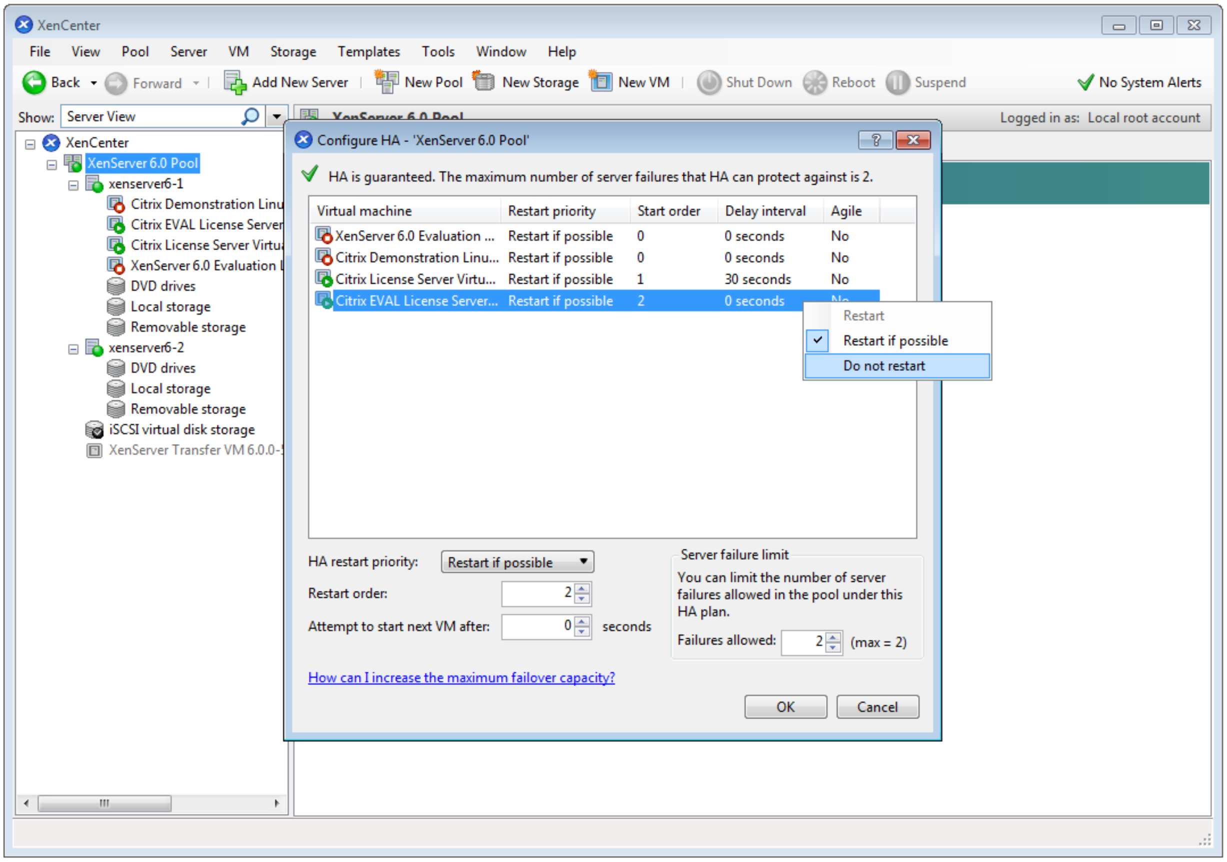
- For more than three host failures, XenServer takes a more conservative approach, assuming every VM to be as large as the largest, and considering each host to have the same number of VMs as the most densely packed host. XenServer does not approximate the host memory, so planning for pools with uneven amounts of host memory is robust and accurate.

Warning: In this planning mode a single very large VM in a pool will result in a low *nhtol* value. If this is a problem, then try to reduce the *nhtol* or try to have a more even spread of VM memory sizes.

Because planning algorithms are designed for unexpected host failures, we only consider absolutely essential resource reservations which would prevent the VM from starting on the alternative host (storage is visible and enough memory is present). We do not perform CPU reservation on the basis that it can be optimized at a later stage using live relocation once the VM is back up and running.

Overcommit protection

The XenServer dynamically calculates a new failover plan in response to any changes in the pool that might affect it (for example, when a new VM is started). If a new plan cannot be calculated because of insufficient resources across the pool, the XenServer will return an over-commitment error message to the client which blocks the operation.


XenCenter
File View Pool Server VM Storage Templates Tools Window Help
Back Forward Add New Server New Pool New Storage New VM Shut Down Reboot Suspend
No System Alerts
Show: Server View
XenServer 6.0 Pool
Logged in as: Local root account
XenCenter
XenServer 6.0 Pool
xenserver6-1
Citrix Demonstration Linu
Citrix EVAL License Server
Citrix License Server Virtu
XenServer 6.0 Evaluation
DVD drives
Local storage
Removable storage
xenserver6-2
DVD drives
Local storage
Removable storage
iSCSI virtual disk storage
XenServer Transfer VM 6.0.0-
Restart
Restart if possible
Do not restart

Configure HA - 'XenServer 6.0 Pool'

✓ HA is guaranteed. The maximum number of server failures that HA can protect against is 2.

Virtual machine	Restart priority	Start order	Delay interval	Agile
XenServer 6.0 Evaluation ...	Restart if possible	0	0 seconds	No
Citrix Demonstration Linu...	Restart if possible	0	0 seconds	No
Citrix License Server Virtu...	Restart if possible	1	30 seconds	No
Citrix EVAL License Server...	Restart if possible	2	0 seconds	No

HA restart priority: Restart if possible

Restart order: 2

Attempt to start next VM after: 0 seconds

Server failure limit

You can limit the number of server failures allowed in the pool under this HA plan.

Failures allowed: 2 (max = 2)

How can I increase the maximum failover capacity?

OK Cancel

Figure 2: Setting HA restart priority

When reconfiguring HA using XenCenter, you can supply a hypothetical series of VM priorities, and XenServer will return a number of host failures which would be tolerated under this scheme. This lets you try various combinations of VM protections, depending on your business needs, and see if the number of host failures is appropriate to the level of paranoia you desire.

You can also use the xe pool-ha-compute-max-host-failures-to-tolerate command to do this.

XenServer HA is done at the XenAPI level, and so any of the standard clients (such as the xe CLI or XenCenter) or any third-party clients which use the XenAPI will all interoperate just fine. The

Finally, HA makes master election completely invisible. Any host in a pool can be a master host, and the pool database is constantly replicated across all nodes and also backed up to shared storage on the heartbeat SR for additional safety. Any XenAPI client can connect to any host, and a redirect is issued to the current master host.

HA Restart priorities

Virtual machines can assigned a restart priority and a flag to indicates whether or not they should be protected by HA. When HA is enabled, every effort is made to keep protected virtual machines live. If a restart priority is specified, any protected VM that is halted will be started automatically. If a server fails then the running VMs will be started on another server.

The restart priorities determine the order in which XenServer attempts to start VMs when a failure occurs. In a given configuration where a number of server failures greater than zero can be tolerated (as indicated in the HA panel in the GUI, or by the ha-plan-exists-for field on the pool object on the CLI), the VMs that have restart priorities 0, 1, 2 or 3 are guaranteed to be restarted given the stated number of server failures. VMs with a best-effort priority setting are not part of the failover plan and are not guaranteed to be kept running, since capacity is not reserved for them. If the pool experiences server failures and enters a state where the number of tolerable failures drops to zero, the protected VMs will no longer be guaranteed to be restarted. If this condition is reached, a system alert will be generated. In this case, should an additional failure occur, all VMs that have a restart priority set will behave according to the best-effort behavior.

If a protected VM cannot be restarted at the time of a server failure (for example, if the pool was overcommitted when the failure occurred), further attempts to start this VM will be made as the state of the pool changes. This means that if extra capacity becomes available in a pool (if you shut down a non-essential VM, or add an additional server, for example), a fresh attempt to restart the protected VMs will be made, which may now succeed.

An explanation of the restart priorities is shown below:

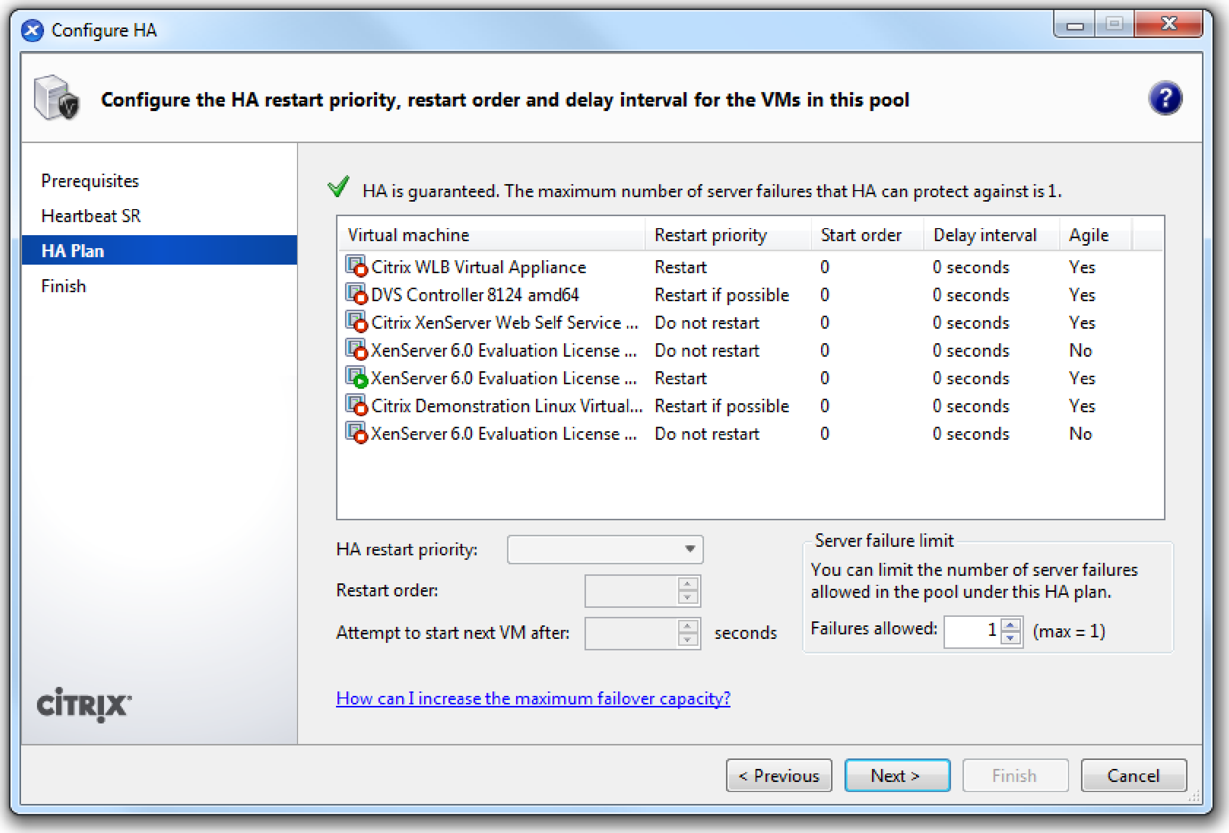
HA Restart Priority	**Restart Explanation**
0	attempt to start VMs with this priority first
1	attempt to start VMs with this priority, only after having attempted to restart all VMs with priority 0
2	attempt to start VMs with this priority, only after having attempted to restart all VMs with priority 1
3	attempt to start VMs with this priority, only after having attempted to restart all VMs with priority 2
best-effort	attempt to start VMs with this priority, only after having attempted to restart all VMs with priority 3

HA Always Run	**Explanation**
True	VMs with this setting are included in the restart plan
False	VMs with this setting are NOT included in the restart plan

Planning for failure

VMs in an HA pool can be either fully protected, best-effort or unprotected. VMs which are protected are all included in the failover planning, and if no plan exists for which they can all be reliably restarted, then the pool is considered to be overcommitted.

Best-effort VMs are not considered when calculating a failover plan, but the pool will still try to start them as a one-off, if a host that is running them fails. This restart is attempted after all protected VMs are restarted, and if the attempt to start them fails, then it will not be retried.



Configure the HA restart priority, restart order and delay interval for the VMs in this pool

✓ HA is guaranteed. The maximum number of server failures that HA can protect against is 1.

Virtual machine	Restart priority	Start order	Delay interval	Agile
Citrix WLB Virtual Appliance	Restart	0	0 seconds	Yes
DVS Controller 8124 amd64	Restart if possible	0	0 seconds	Yes
Citrix XenServer Web Self Service ...	Do not restart	0	0 seconds	Yes
XenServer 6.0 Evaluation License ...	Do not restart	0	0 seconds	No
XenServer 6.0 Evaluation License ...	Restart	0	0 seconds	Yes
Citrix Demonstration Linux Virtual...	Restart if possible	0	0 seconds	Yes
XenServer 6.0 Evaluation License ...	Do not restart	0	0 seconds	No

HA restart priority:

Restart order:

Attempt to start next VM after: seconds

Server failure limit

You can limit the number of server failures allowed in the pool under this HA plan.

Failures allowed: 1 (max = 1)

How can I increase the maximum failover capacity?

Figure 1: Configuring HA Plan

Advanced HA features are available using the CLI. Each protected VM in an HA pool can be assigned a numeric ha-restart-priority. If a pool is well-resourced with a high nhtol, then these restart priorities are not relevant and the VMs are all guaranteed to be started.

If more hosts fail than have been planned for, then the priorities are used to determine the order in which VMs are restarted. This ensures that in over-committed pools, the most important VMs are restarted first. Although the pool will start priority 1 VMs first, they might not finish booting before the priority 2 VMs, and so this should not be used as the basis for service ordering.

It is very important to ensure that a VM is agile when protecting it by HA. If the VM is not agile (for example, if it has a physical CD drive mapped in from a host), then it can only be assigned Best Effort restart because it is tied to one host.

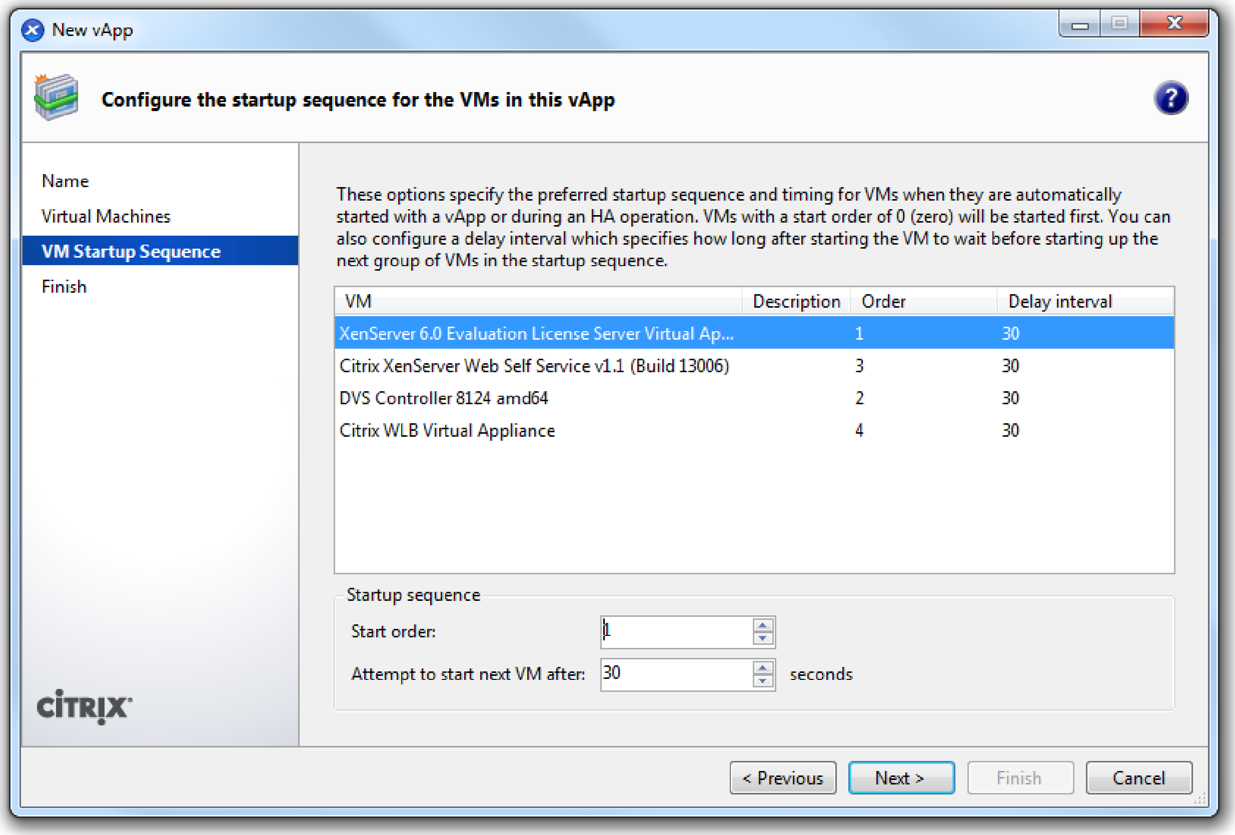
vApp integration with HA

The vApp functionality is fully integrated with our HA functionality. Even during HA operations the preferred start-up sequence and timing for vApp VMs are automatically retained and followed.

A *vApp* is a logical group of one or more related VMs that can be started simultaneously, as a single entity. When a vApp starts, the VMs contained within it will start in a user-defined order, so VMs that depend upon each another start in the correct sequence.

For software updates and other tasks that require reboots, creating vApps can free administrators from manually sequencing the startup of dependent VMs. The VMs within the vApp do not have to reside on one host and will be distributed within a pool using the normal rules and workload balancing.

The vApp functionality is particularly useful in a Disaster Recovery situation in which an administrator might group all VMs that reside on the same Storage Repository or that relate to the same Service Level Agreement (SLA).



Configure the startup sequence for the VMs in this vApp

These options specify the preferred startup sequence and timing for VMs when they are automatically started with a vApp or during an HA operation. VMs with a start order of 0 (zero) will be started first. You can also configure a delay interval which specifies how long after starting the VM to wait before starting up the next group of VMs in the startup sequence.

VM	Description	Order	Delay interval
XenServer 6.0 Evaluation License Server Virtual Ap...		1	30
Citrix XenServer Web Self Service v1.1 (Build 13006)		3	30
DVS Controller 8124 amd64		2	30
Citrix WLB Virtual Appliance		4	30

Startup sequence

Start order: 1

Attempt to start next VM after: 30 seconds

Figure 3: Configure vApp VM boot sequence during HA operation

XenCenter support for HA

The best practice for HA is not to make configuration changes while it is enabled. If you are actively making configuration changes, such as applying patches, disable HA while making the changes.

XenCenter makes some common changes to make HA much more user-friendly:

- Protected VMs can be shut down using the CLI or from within the guest. If you try to shut down from XenCenter, it will give you the option of unprotecting the VM and then shutting it down first. Accidental in-guest shutdowns do not result in downtime, but administrators will still have the option to stop a protected guest if desired.
- If you want to reboot a host when HA is enabled, XenCenter automatically uses the hypothetical planning calculation to determine if this would invalidate the failover plan. If it does not affect it, then the host is shut down normally. If the plan would be violated, but the nhtol is greater than 1, XenCenter will give the administrator the option of lowering the nhtol value by 1. This reduces the overall resilience of the pool, but always ensures that at least one host failure will be tolerated. When the host comes back up, the plan is automatically recalculated and the original nhtol value restored if appropriate.
- If you try to apply a hotfix, XenCenter will disable HA for the duration of the pool patching wizard. It is important to manually keep an eye on hotfix application to ensure that host failures do not disrupt the operation of the pool.

HA best practices

Hosts per HA-enabled resource pool requirements

The HA maintains the state of all three hosts. If one loses the connection or its heartbeat (host 3) to others, HA determines that Hosts 1 and 2 can still communicate (this is the larger host group; larger network partition). Therefore, Host 3 is fenced (smaller host group; smaller network partition).

From the example above, there is one problem with having only two servers in the pool. The network partitions are the same size after losing connection between Host 1 and 2. In this case, the host with the higher host-UUID is fenced and rebooted.

In pool of less than 3 hosts HA may not function as expected. This is because HA does not check the link state of the interfaces. It rather works on the bases of HA Host Groups and Network Partitions. The HA Daemon chooses the larger of the network partitions to take over and fences servers on the smaller partition.

When there are three hosts in the pool and one losses connection, HA Daemon keeps the larger network partition with two hosts and the smaller network partition with one host. The host in the smaller partition is fenced and rebooted.

When there are only two hosts in a pool and one of them loses connection to the other, there is no larger partition to choose. When there are only two hosts in an HA pool, there is no way for the HA daemon to know which of the two hosts has had its network cable pulled. It ends up defaulting to the host with the lowest UUID. For more details read Citrix Knowledge Center Article CTX129721 - High Availability Behavior When the Heartbeat is Lost Between Two Hosts in a Pool.

Migrating VMs (XenMotion) with HA enabled

HA needs to have enough memory available to be able to guarantee a restart of all protected VMs in the event of the number of host failures specified occurs. When a migration using XenMotion occurs, HA works out a plan on how to react to each host’s failing. When migrating, you are using an extra amount of RAM on the target host, so for HA to cope with either the source or target failing, it needs to have that extra amount of RAM available on both the host and destination server. This follows the restrictions that are put in place so that HA can provide guarantees on VM uptime and availability.

When HA is enabled and a VM HA Protection level is set to Protected, in addition, the destination server must have the available memory needed to migrate the VM using XenMotion. The host server that is still holding the memory of the VM also must have that same amount of memory available (in reserve) to perform the migration. In other words, the host server must have available the same amount of memory that the VM is about to release.

If HA is not able to satisfy these requirements, you receive the following error message: “Error: This operation cannot be performed because HA would no longer be guaranteed for this pool. To perform this operation anyway, you must disable or reconfigure HA.” For more details read Citrix Knowledge Center Article CTX120499 - XenMotion Fails when HA is Enabled.

Conclusion

As server virtualization environments become the standard platform for cloud, desktop and IaaS environments running business and mission critical applications, high availability and disaster recover becomes even more critical. Providing high availability at different levels of the server virtualization infrastructure mitigates the risk of significant downtime and user impact for integrated solutions that require significant and reliable business continuity.

About Citrix

Citrix Systems, Inc. (NASDAQ:CTXS) is the leading provider of virtualization, networking and software as a service technologies for more than 230,000 organizations worldwide. Its Citrix Delivery Center, Citrix Cloud Center (C3) and Citrix Online Services product families radically simplify computing for millions of users, delivering applications as an on-demand service to any user, in any location on any device. Citrix customers include the world’s largest Internet companies, 99 percent of Fortune Global 500 enterprises, and hundreds of thousands of small businesses and prosumers worldwide. Citrix partners with over 10,000 companies worldwide in more than 100 countries. Founded in 1989, annual revenue in 2008 was $1.6 billion.