Virtualizing Active Directory Domain Services On VMware vSphere®

Release 2.0

June 2014

vmware®

VMware, Inc.
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com

Contents

List of Figures

List of Tables

1.   Introduction

As the prominent directory service and authentication store, Active Directory Domain Services (AD DS) is in the majority of network infrastructures. In some environments AD DS is viewed as another required service, but it does not attract much attention. In other environments AD DS is treated as the business critical application (BCA) that it is. Considering that the ability to access network resources and the Internet, look up user information, and use email often requires AD DS, it is worth understanding the importance of this service and the stability of its underlying infrastructure.

In much the same way that the criticality of AD DS differs from organization to organization, so does the acceptance of virtualizing this service. More conservative organizations choose to virtualize a portion of the AD DS environment and retain a portion on physical hardware. The cause is typically misinformation, lack of experience in virtualization, or fear of the unknown.

With the release of Windows Server 2012, new features alleviate many of the legitimate concerns that administrators have about virtualizing AD DS. These new features, the latest versions of VMware® vSphere®, and recommended practices help achieve 100 percent virtualization of AD DS.

1.1   Purpose

This guide provides best practice guidelines for deploying AD DS on vSphere. The recommendations in this guide are not specific to a particular set of hardware or to the size and scope of a specific AD DS implementation. The examples and considerations in this document provide guidance, but do not represent strict design requirements.

1.2   Target Audience

This guide assumes a basic knowledge and understanding of vSphere and AD DS.

- Architectural staff can use this document to understand the design considerations for deploying a virtualized AD DS environment on vSphere.

- Engineers and administrators can use this document as a catalog of technical capabilities.

- Management staff and process owners can use this document to help model business processes that take advantage of the savings and operational efficiencies achieved with virtualization.

1.3   Scope

The scope of this document is limited to the following topics:

- Drivers for virtualizing AD DS – overview of AD DS and the reason that the vSphere platform is ideal for the virtualization of AD DS.

- Historical inhibitors and recent facilitators for virtualization – the historical reasoning for maintaining a physical presence in AD DS environments and how impediments have been removed through technical advancements and real-world experience.

- Technical considerations and best practices – features and design considerations of AD DS and vSphere to achieve successful virtualization.

2.   Why Virtualize Active Directory?

AD DS is a multi-master, hierarchical directory service with the following features:

- A database schema that governs the objects and attributes held in the database

- A global catalog of all data within the entire directory structure

- A replication service

- A set of role masters singularly responsible for critical services within the forest and domains, such as schema updates and distribution of security principle relative identifiers (RIDs).

The primary use of this directory service is user and computer authentication within a domain, a set of domains, a forest or a set of forests. However, Active Directory has evolved to more than an authentication service. In many organizations, it is a central repository for not only user and computer data, but also for application configuration information, network resource location services, and name resolution, and so on. It also acts as the authentication source for external systems. It is clear that AD DS is a critical piece of infrastructure and must be designed with the same diligence as any other BCA.

The criticality of AD DS should not be a deterrent to virtualizing domain controllers. Domain controllers are computers that run AD DS and keep full copies of the Active Directory database for their domain. The characteristics of domain controllers make them ideal for virtualization.

2.1   Workload Characteristics

Although domain controllers are a central part of the infrastructure that almost every user and computer interacts with on a daily basis, the workload characteristics of domain controllers are not as significant. Domain controllers handle hundreds (and in very active environments, thousands) of queries per minute. During a spike in activity, the load can be hundreds of queries per second. However, much of this data is accessible in memory, reducing the overhead associated with writing and reading from disk. If an organization has standardized on Windows 2008 R2, a 64-bit operating system, it is possible that the entire Active Directory database is stored in memory.

The distributed nature of Active Directory enables out-of-the-box load balancing for client communication. This feature is dependent on how the organization has chosen to scale its domain controller infrastructure. However, virtualization removes the need to limit the deployment of domain controllers due to hardware availability concerns. Multiple smaller domain controller virtual machines can provide the same performance as fewer larger domain controllers, while providing increased high availability by scaling the workload horizontally.

Many domain controllers are implemented only as a physical or virtual server with an installation of Windows, an anti-virus program, a monitoring agent, and a backup utility. This setup leads to low resource consumption. On physical servers with multicore processors, compute resources are wasted when they are dedicated to a single application, such as Active Directory. Domain controllers are easily deployed on small virtual machines, allowing consolidation with other applications and services.

2.2   Virtualization is Mainstream

Many organizations have developed *virtualization first* policies. Typically, this phrase means that any new servers to be provisioned in the data center are deployed in the virtualized environment. In some more advanced organizations, physical servers are the exception, requiring upper-level management approval before a system other than VMware vSphere Hypervisor can be deployed. The reason is not only that virtualization drives efficiency, reduces capital and operational expenditures, and increases productivity, but also because it is robust enough for the most aggressive workloads. Otherwise, the largest companies in the world would not be running their most critical applications on vSphere.

There was a time when the virtualization of production workloads was met with resistance and skepticism. The hundreds of independent performance studies, customer references, and real-world scenarios described on the Internet prove that a shift has occurred over the past few years. Most organizations have re-evaluated and are using virtualization for their most critical and aggressive workloads. Physical servers running vSphere are no longer the minority in data centers. Very soon, when the last remaining physical domain controller is out of warranty, the policy of most organizations will no longer allow the deployment of a physical Windows operating system. At this point, there is no technical reason why the entire Active Directory environment cannot be virtualized.

2.3   Availability

When domain controllers are distributed according to best practices, the loss of a single domain controller does not impact the availability of the directory service. Each domain controller maintains a copy of the entire directory for its respective domain. Users and applications locate domain controllers and advertised services using the Domain Name Service (DNS), which is often hosted on domain controllers. In the case of an unresponsive domain controller, requests are retried against a responsive domain controller in round-robin fashion.

Certain domain controllers perform specific Operations Master roles that make them unique in an Active Directory infrastructure because these roles can only be performed by the designated operations masters. All requests requiring such roles are sent to the designated domain controllers. When the domain controller responsible for the applicable role is unavailable to service that request (such as a schema update or creating a new domain), the request will fail. However, these failures are usually local to the requestor and these types of requests are infrequent in the normal course of an Active Directory operation. Additionally, the condition is easily correctable—an operations master role can be transferred or seized from a role-holder if remediating the role-holder is determined to be more administratively expensive.

In a physical environment, the issue with providing a highly-available Active Directory infrastructure is the need for more servers. Knowing the workload characteristics of most domain controllers, it is not practical to dedicate two or three of today's powerful servers when they are utilized at only 5 to 10 percent of their capacity. Building domain controllers on an existing vSphere infrastructure, or building a new vSphere infrastructure on the two or three physical servers is more efficient.

3. Understanding Domain Controller Virtualization

The historical inhibitors to virtualizing domain controllers are not many, but due to the nature of AD DS, they are compelling. Many of these inhibitors are viewed as legitimate blockers for virtualizing domain controllers. Fortunately, VMware can solve most of the issues; some through technical solutions and others through operational discipline, corporate policies, and common sense.

3.1 Securing Virtualized Domain Controllers

The recommendation for physical domain controllers to be protected from unauthorized physical access has been in existence for a long time. This physical security is in addition to the access control that is inherently provided using native Active Directory security groups to protect the logical access to Active Directory data. The focus of this section is the physical security of the data on the domain controller.

With physical domain controllers, the common scenario is to deploy the Active Directory database on direct-attached storage, creating one physical unit of containment. This unit is then protected by isolating the physical servers in a cage within a data center, placing them in a locked closet, or using some other way to satisfy the requirement. Realistically, domain controllers are protected using the same level of physical protection as any other physical server in a data center (a badge outside the data center door and, at most, a locked rack).

The virtualization of domain controllers comes under scrutiny because of the perceived ease with which a virtual domain controller can be moved or migrated and its virtual hard disk copied. The ability to gain access to a vCenter Server or a VMware ESXi™ host to the level that allows a malicious user to control a virtualized domain controller exposes a larger issue with infrastructure security. This is a problem better dealt with through policy and operational processes as opposed to technology.

In the most restrictive of environments, technical solutions might be required. If that is the case, there are recommendations for hardening virtual machines and vCenter Servers that range from restricting network access to setting up management networks to limiting the interaction between virtual machines and host memory. For a small percentage of environments, security can be taken to the extreme, causing management overhead. For most environments, commonly followed security practices are sufficient.

For more information on securing virtual machines and vCenter Server, refer to *vSphere Security ESXi 5.1 vCenter Server 5.1* at http://pubs.vmware.com/vsphere-51/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-51-security-guide.pdf.

To mitigate the effects of a possible theft of domain controller disks (and, consequently, the data contained within), organizations can leverage industry-standard encryption technologies. Encryption can be done at the array, host, disk, or file-level. Each of these options have their advantages and disadvantages, and each organization can determine the suitability of each option based on the organization's overall goals, policies, and budget.

For example, all the major storage hardware vendors have encryption solutions that encrypt all data at the physical disk level. The administrative overhead for ensuring data security and fidelity is relatively low; however, these solutions are generally more expensive because of the specialized disks that are required to achieve the objectives.

Bitlocker and other options, such as TrueCrypt, provide cheaper alternatives to the array-based options. These solutions enable in-guest, whole-disk, partition, or file-level encryption. Administrative maintenance and the possible loss of recovery keys are some of the drawbacks to these options.

Although VMware vSphere does not currently support host-level encryption (encryption at the ESXi level), all other encryption options can be used to secure the disks of a domain controller virtualized on the vSphere platform. The considerations and requirements for the use of these encryption options on a physical domain controller remain same when the domain controller is virtualized. The costs and benefits of the solutions are identical regardless of the physical or virtual nature of the domain controller.

3.2   Protecting AD DS against Virtual Infrastructure Failures

The most common deployment scenarios found in production environments consist of a mix of physical and virtual domain controllers. These hybrid deployments are typically in place because of legacy physical systems, a sense of comfort in having one or more physical domain controllers in the environment, or fear of a failure in the virtual infrastructure that might disrupt the directory services. The first two reasons are valid. Migrations take time and legacy applications are often hard coded to a specific piece of infrastructure. Many organizations have virtualized the majority of domain controllers, but one or two physical domain controllers remain as a safety measure against legitimate concerns around a completely virtualized Active Directory. The last reason, however, is typically attributed to users who are not familiar with vSphere.

Active Directory is meant to be deployed in a distributed architecture to support a robust and resilient service. vSphere can also be deployed using vSphere clusters to distribute workloads among hosts. A vSphere cluster is a unit of management. That is, a group of ESXi hosts that can take over for each other in the case of a failure. Typically, a vSphere cluster shares storage targets, networking, and possibly server chassis or racks. With redundancy built into the various levels (storage, network, and physical placement), a single vSphere cluster can be made resilient by isolating compute resources into multiple failure domains. This is the recommended approach when the virtual infrastructure is limited to a single vSphere cluster. In a design with multiple clusters, failure domain isolation is implemented more easily.

When designed with availability and resiliency in mind, a virtual infrastructure can be fault tolerant. The argument to not virtualize every domain controller because of a fear of a failure at the virtual infrastructure level that might result in a loss of directory services has little merit in a well-designed environment.

3.3   Time Synchronization

Proper and accurate timekeeping is a critical service used by Active Directory and its clients for authentication and replication arbitration. Due to the critical nature of having accurate time across the environment, there is a perception that virtualization might exacerbate any flaws in the time synchronization mechanisms used throughout the environment. Virtual machines are known to experience some level of time drift due to the sharing of physical resources that can occur as a result of multiple virtual machines running on a single physical host. This concern can be mitigated by following VMware and Microsoft best practices for time keeping.

Operating systems track the passage of time using one of two methods: tick counting or tickless timekeeping. In tick counting, the operating system uses a hardware device to pass interrupts at a predetermined interval, such as 100 times per second. The operating system keeps track of these interrupts, or ticks, to determine how much time has passed. In tickless timekeeping, a hardware device keeps track of the number of time units that have passed since the system booted. When needed, the operating system reads the counter. Windows-based operating systems use ticks to keep time. Windows-based systems with multiprocessor HALs and some ACPI uniprocessor HALs use the CMOS periodic timer to deliver interrupts, or ticks.

In addition to tracking the passage of time, operating systems must keep track of absolute time, or wall-clock time. During startup, the operating system reads wall-clock time from the CMOS, to the nearest second, or from a network time source. In virtual machines, VMware Tools™ provides an additional mechanism to synchronize the guest operating system's clock with the host's clock.

Timer devices in physical machines tend to drift as do those in virtual machines. As a result, timekeeping software is generally required to keep time accurate over the long term. For Windows-based operating systems, this is provided natively using the Windows Time Service (W32Time).

In an Active Directory forest, the Windows Time Service keeps time following a hierarchy, starting at the forest root PDC emulator to the lowest-level domain clients. Because of Active Directory's dependence on the Windows Time Service, it is the natural choice for time-keeping within Windows-based guests.

By default, domain-joined Windows systems use the domain hierarchy for time keeping. In virtual machines, this setup might conflict with VMware Tools periodic clock synchronization. The result is two time services, one pointing to a reliable time source, the other pointing to a non-reliable time source, thereby updating the time to inaccurate values. For this reason, the recommendation is to standardize on one tool. For domain-joined Windows machines, the Windows Time Service is the preferred method.

VMware Tools time synchronization is disabled by default. To verify that time synchronization is disabled, check the VMware Tools user interface or virtual machine advanced configuration (`.vmx` file). Even with VMware Tools time synchronization disabled, time synchronization is performed during guest operating system startup and during some virtual machine maintenance operations on the vSphere platform.

For more information on disabling VMware time synchronization for virtual machines, see the VMware KB article *Disabling Time Synchronization* at http://kb.vmware.com/kb/1189.

3.4   USN Rollback
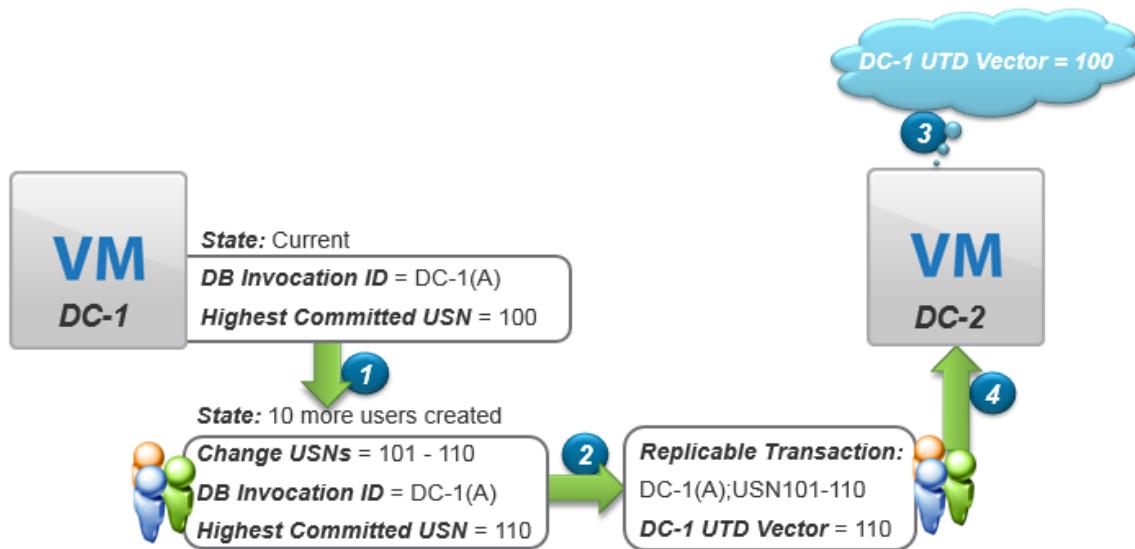
The distributed, multi-master design of Active Directory makes it robust and scalable. However, there must be mechanisms in place for detecting and correcting simultaneous edits to objects and ensuring proper convergence of changes within the directory. Active Directory replication can be a complicated topic. However, this section focuses on the cause and effect of Update Sequence Number (USN) rollback and the steps that have been taken to address this issue.

Every write transaction performed by a domain controller against an object in the directory must be uniquely traceable back to the originating domain controller. Every transaction is stamped with a globally unique identifier, the Update Sequence Number (USN), and the identity (InvocationID) of the Active Directory database where the write operation occurred. Together, these processes provide a replicable transaction.

When domain controllers create new objects or write changes to existing objects in the database, new USNs are continuously created. Domain controllers keep track of replication using the Up-to-Dateness (UTD) Vector table, which tracks the Globally Unique Identifier for each domain controller (DSA GUID), the USN UTD vector for the local domain controller and its replication partners, and the time of the last successful replication from each replication partner. This helps to ensure that domain controllers are aware that new changes are available for replication.

In the following illustration, domain controller 1 (DC-1) creates 10 new users. These users are stamped with USN 101 through USN 110. The InvocationID of the originating domain controller (DC-1) is DC-1(A). USNs 101 – 110 plus InvocationID DC-1(A) represent the replicable transactions. The change is replicated to domain controller 2 (DC-2).
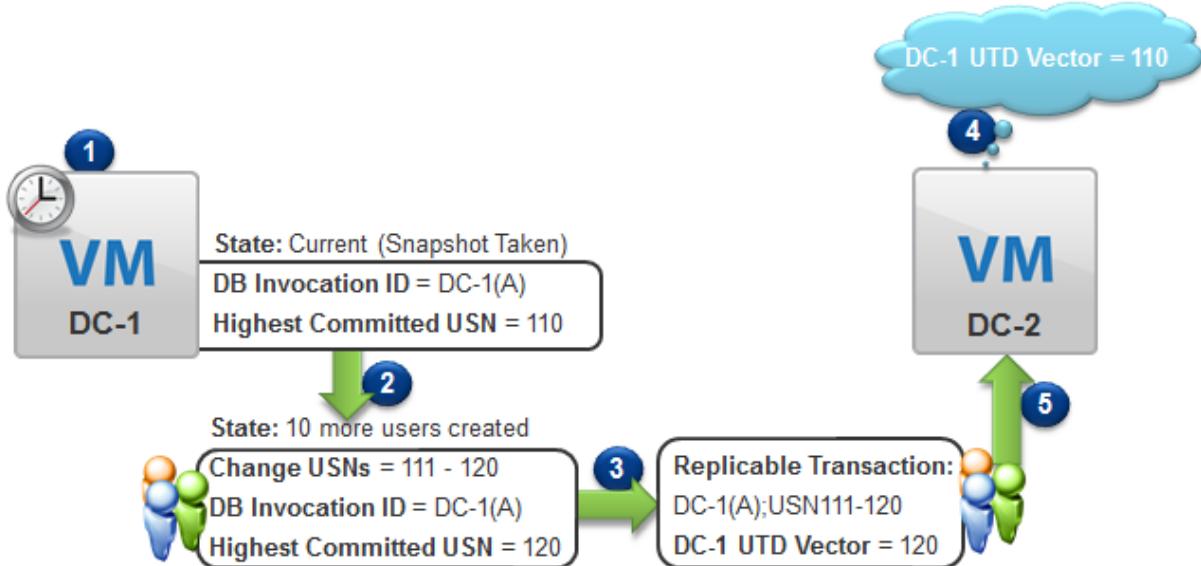
Figure 1. Active Directory Replication

The virtualization of domain controllers allows administrators to take advantage of a new feature set not available to physical domain controllers. Among those features, cloning and snapshots are two powerful tools that enable rapid deployment, testing, and recovery. However, because of the way in which Active Directory tracks the replication of changes, the use of snapshots (to revert to a point in time) or the cloning of domain controllers causes a divergence in replication tables. This discrepancy can lead to data corruption within the Active Directory database, resulting in a USN rollback. A USN rollback is a condition in which a domain controller accepts new write transactions, but is unable to replicate them to its replication partners because the domain controller's local UTD vector value is lower than the value expected by the replication partners.

In the following illustration, domain controller 1 (DC-1) has a highest committed USN value of 110, which is in line with DC-2's expectation. An administrator takes a snapshot of the virtualized domain controller and then creates 10 more users. Because the USNs of these changes are higher than the UTD vector of DC-1, the changes are replicated to domain controller 2 (DC-2).

Figure 2. Users Created After a Virtual Machine Snapshot
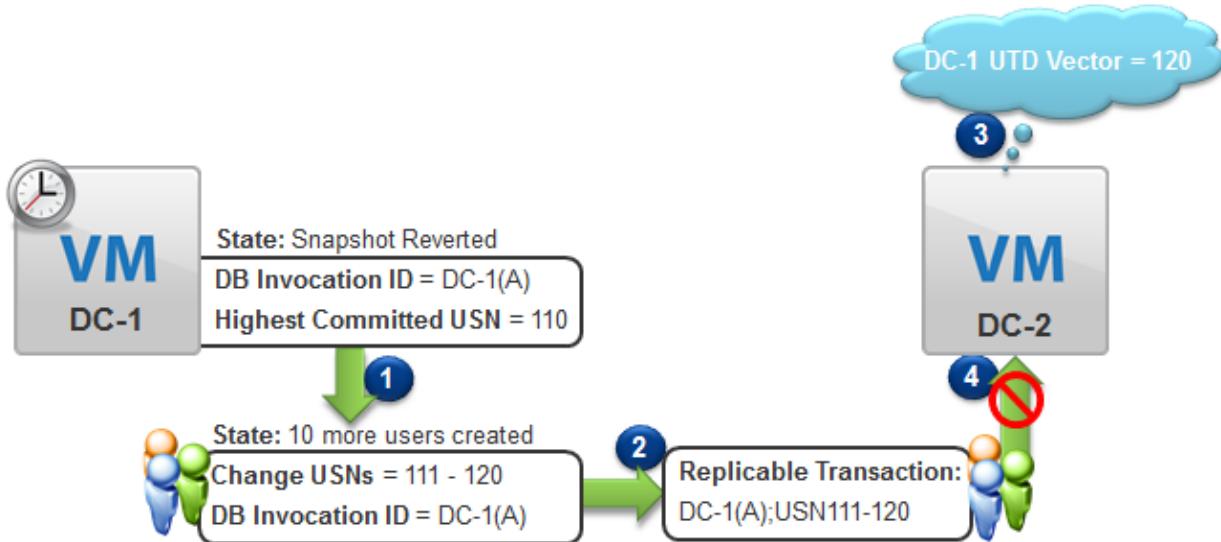
After the replication of the new users has converged, the administrator reverts to the most recent snapshot taken of domain controller 1 (DC-1), rolling it back to the point before the last batch of users was created (Highest Committed USN = 110). However, the users remain in the active directory database on other domain controllers, which are not aware of the snapshot reversion that occurred on DC-1. They also still remember the last UTD vector they have for DC-1 based on the last successful replication.

Figure 3. Domain Controller Reverted to Previous Snapshot

As a result of DC-1 being rolled back, its highest committed USN is 110. However, all other domain controllers have a UTD vector of 120 for DC-1. The administrator creates 10 new users on DC-1, which are assigned USNs = 111 – 120 and InvocationID = DC-1(A). Because DC-2 has a UTD vector of 120 for DC-1 and an InvocationID of DC-1(A), DC-2 does not inbound replicate the latest updates made on DC-1.

Figure 4. USN Rollback Effect after Reverting DC-1 Snapshot

Because of the UTD vector, DC-2 assumes that DC-1 has knowledge of the previously-created 10 users, and as a result, the changes corresponding to USN 111 through 120 are not replicated to DC-1. DC-1 has 10 users that it assumes are valid, but which do not exist on any other domain controller. Additionally, DC-2 has 10 users that it assumes are valid, but which do not exist on DC-1. This is an example of how USN rollback can disrupt replication throughout the environment.
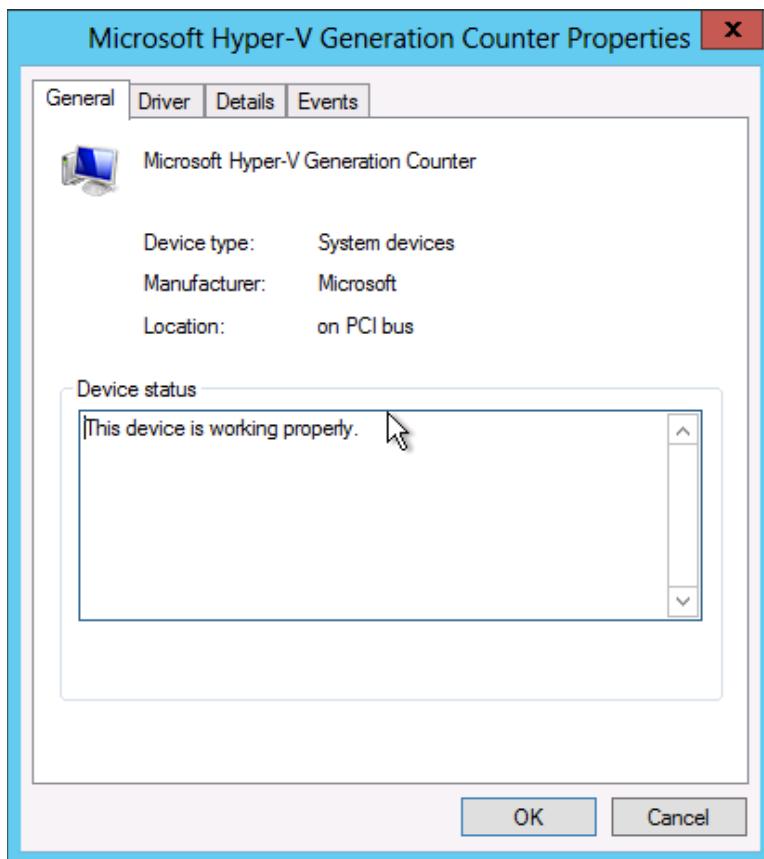
Among the technical reasons for not virtualizing domain controllers, the fear of a USN rollback is perhaps the most compelling and technically valid. However, mitigating the risk of a USN rollback is not difficult. The best way to avoid a USN rollback is to implement a technical solution as opposed to a policy-based solution. Microsoft has achieved this in Windows Server 2012 with its implementation of VM-Generation ID.

3.5   Windows Server 2012 Virtualization Safeguards

With Windows Server 2012, Microsoft introduced the VM-Generation ID, a 128-bit counter exposed by the hypervisor to the virtual machine guest operating system. The VM-Generation ID provides the virtual machine guest operating system with knowledge of the state of the virtual machine. For example, whether or not the virtual machine has been restored to a previous point in time or has been cloned.
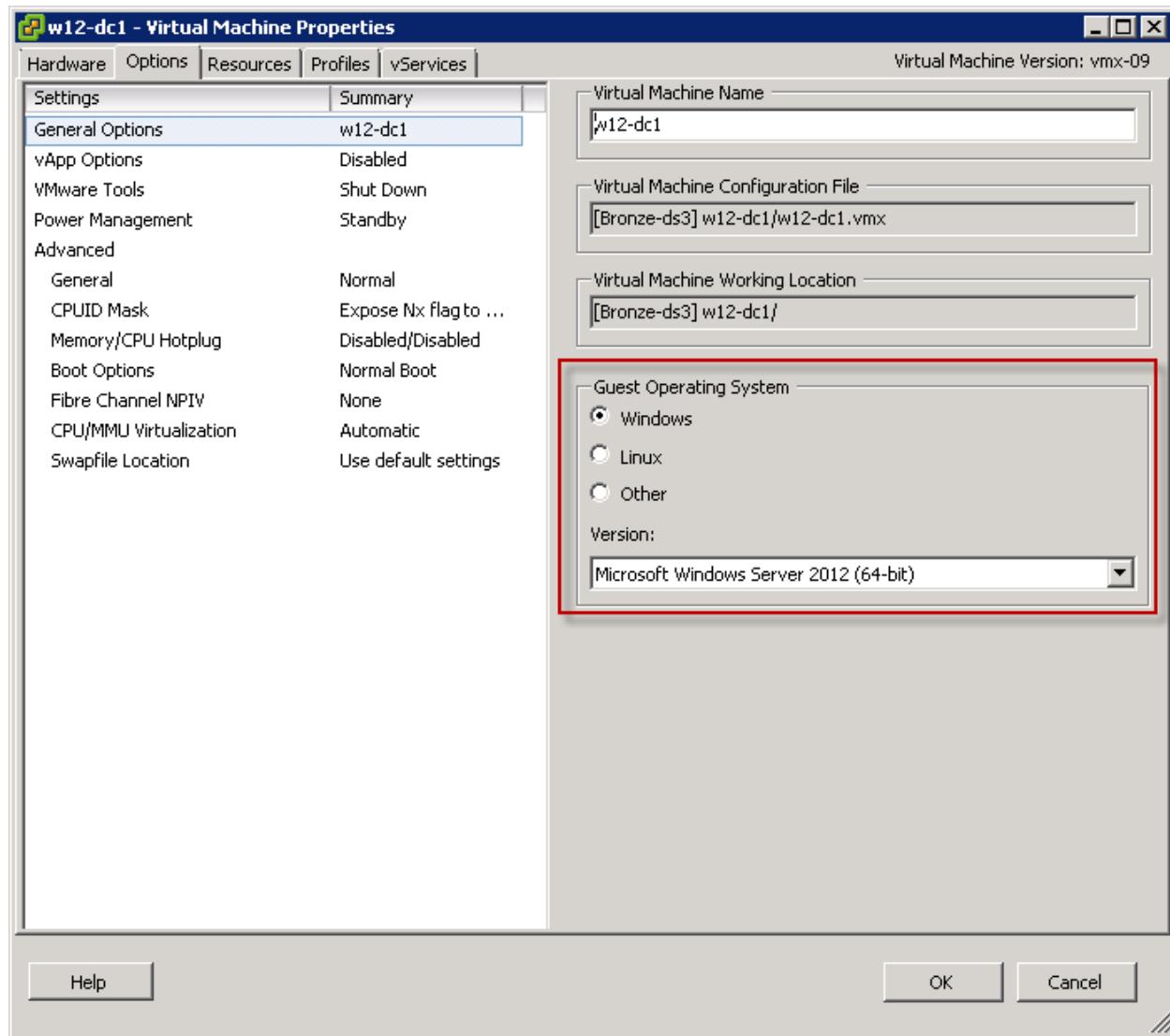
The VM-Generation ID is exposed to the guest operating system through the Microsoft driver, `Microsoft Hyper-V Generation Counter`. Contrary to the name, the driver is a generic communication path between the guest operating system and the hypervisor provided by Microsoft and is not specific to Hyper-V. The following screen shows the properties page for the Microsoft Hyper-V Generation Counter device.

Figure 5. Microsoft Hyper-V Generation Counter Device

VM-Generation ID is available in both the Microsoft Windows 8 and Windows Server 2012 operating systems. However, to take advantage of VM-Generation ID on the vSphere platform, vCenter Server and vSphere hosts must be running version 5.0 update 2 or later, and the virtual machine guest operating system must be configured as Microsoft Windows Server 2012 (64-bit) or later. See the following screen.

Figure 6. Virtual Machine Guest Operating System Version
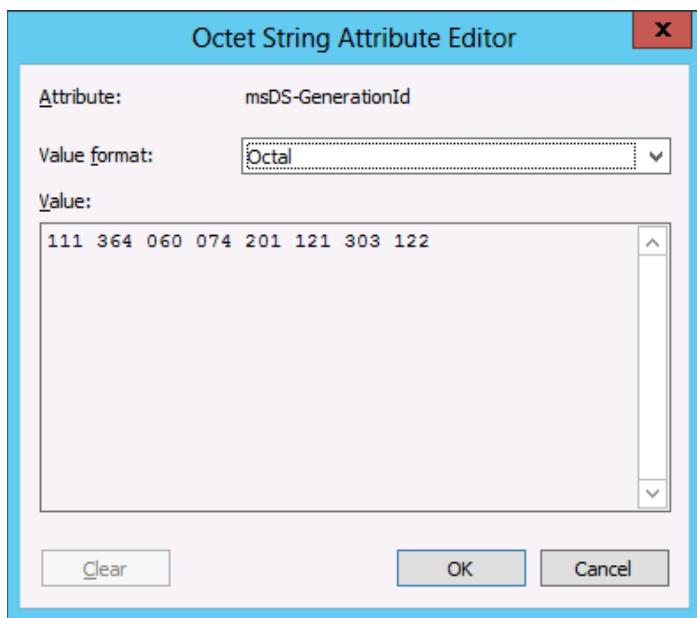
When deployed on a supported hypervisor and supported guest operating system, the VM-Generation ID is created during the creation of the virtual machine. Over the lifecycle of the virtual machine, the VM-Generation ID is updated as a result of various state changes of the virtual machine. The following table lists the common changes in the lifecycle of a virtual machine and whether or not these changes cause a VM-Generation ID change.

Table 1. Virtual Machine State Changes

Scenario	*VM-Generation ID Change*
VMware vSphere vMotion®/VMware vSphere Storage vMotion	No
Virtual machine pause/resume	No
Virtual machine reboot	No
vSphere host reboot	No
Delete VM Snapshot	No
Import virtual machine	Yes
Cold clone	Yes
Hot clone *Note* Hot cloning of virtual domain controllers is not supported by either Microsoft or VMware. Do not attempt hot cloning under any circumstances.	Yes
New virtual machine from VMware Virtual Disk Development Kit (VMDK) copy	Yes
Cold snapshot revert (while powered off or while running and not taking a memory snapshot)	Yes
Hot snapshot revert (while powered on with a memory snapshot)	Yes
Restore from virtual machine level backup	Yes
Virtual machine replication (using both host-based and array-level replication)	Yes

Active Directory takes advantage of VM-Generation ID to understand the state of the virtual machine. This is accomplished through the new msDS-GenerationId attribute that is now part of domain controller computer objects in Windows Server 2012. See the following figure.

Figure 7. msDS-GenerationId Attribute

Active Directory Domain Services, running on every Windows Server 2012 domain controller, and hosted on a hypervisor supporting VM-Generation ID, evaluates the VM-Generation ID during startup and before every write operation to ascertain that the present value is not different from the one stored in the msDS-GenerationId attribute within the local active directory database. A domain controller evaluates its local database for the value of the msDS-GenerationId in the following instances:

- When a new domain controller is created, the msDS-GenerationId attribute is evaluated and populated for the first time.

- During startup of an existing domain controller, the VM-Generation ID is evaluated. If the attribute is not already populated, it is then populated for the first time. This is the case when a virtual machine has been powered on for the first time on a supported hypervisor (for instance, during a virtual machine import or hypervisor upgrade).

- During startup of an existing domain controller with a populated msDS-GenerationId attribute, the VM-Generation ID is evaluated to verify whether or not the state of the virtual machine has changed while the virtual machine was powered off.

- On a running domain controller, before a write operation to the Active Directory database is committed, the VM-Generation ID is evaluated to verify whether or not the state of the virtual machine has changed during runtime.
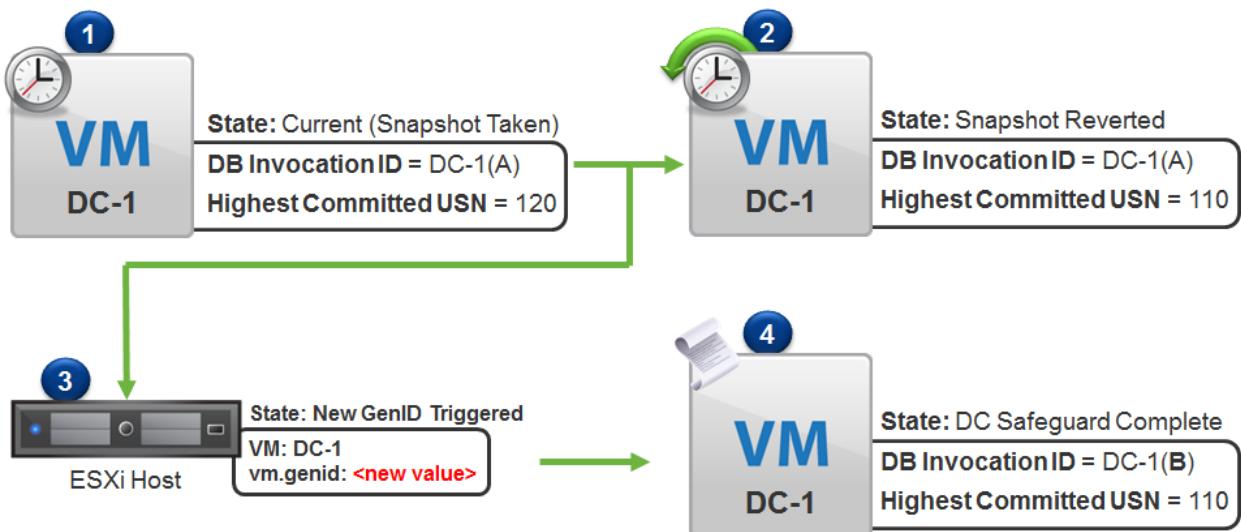
The implementation of VM-Generation ID support in Active Directory provides two new features to virtualized domain controller implementations – domain controller safeguard and domain controller cloning.

3.5.1 Domain Controller Safeguard

The domain controller safeguard feature allows a domain controller that has been reverted from a snapshot, restored from a virtual machine-level backup, or replicated for disaster recovery purposes, to continue to function as a member of the Active Directory. During startup of the directory service and prior to committing writes to the database, the VM-Generation ID is evaluated. If the VM-Generation ID has changed, the domain controller performs a set of immediate actions (virtualization safeguards) to guarantee that the Active Directory does not become corrupt and that the domain controller does not become isolated.
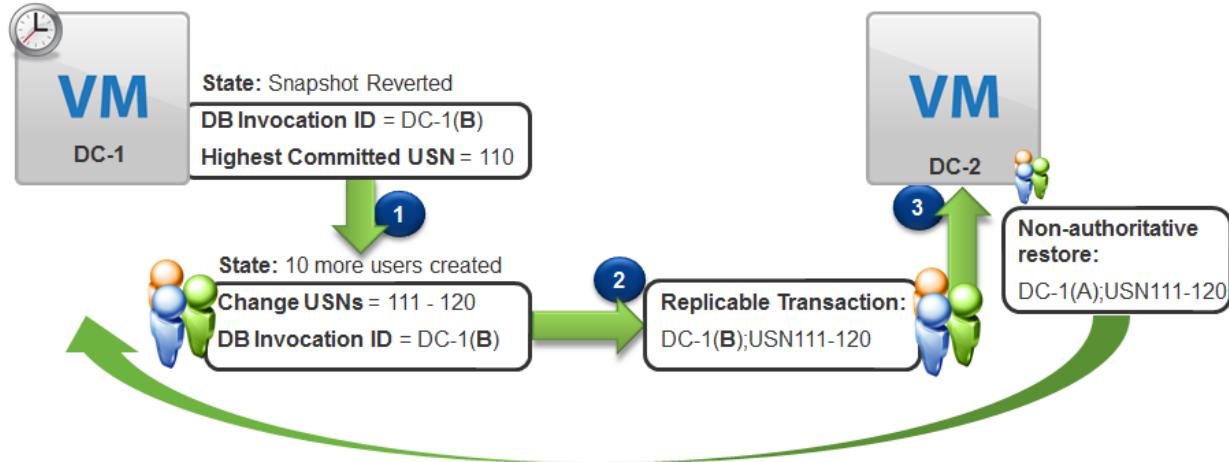
First, the domain controller's local RID pool is invalidated to avoid duplicate RIDs from being assigned to new security principals, such as new users, security groups and computers. Second, the domain controller changes the InvocationID of the local Active Directory database. Changing the InvocationID of the database allows the domain controller to continue committing updates to Active Directory and ensures that all replication partners regard the new changes as valid, replicable transactions.

Figure 8. Domain Controller Snapshot Reversion with Safeguard

In the case of a snapshot restore, a virtual machine-level restore, or a replicated domain controller being powered on due to a disaster recovery, after the safeguards have kicked in, new changes can continue to be committed to the database. Changes that previously originated from the recovered domain controller, but were lost due to the recovery, are replicated.

Figure 9. Replication After Safeguard

Virtual machines that were cloned or created from a virtual machine copy and not prepared correctly can perform safeguard operations but are automatically rebooted into directory restore mode (DSRM) to protect the source domain controller from becoming isolated. The behavior in this scenario is different from a snapshot rollback situation and will be discussed next.

3.5.2 Domain Controller Cloning

VM-Generation ID supports domain controller cloning. This is the ability to clone a virtual machine, which is already a domain controller, to create a new domain controller. This provides a fast and safe way to create new domain controllers based on an existing domain controller virtual machine which has been properly prepared. Preparing a reference domain controller for cloning involves a number of specific administrative tasks which must be successfully completed before the cloning process is initiated.

To prepare an existing domain controller for cloning, first verify that the software installed in Windows can be safely cloned. Some applications, such as monitoring, antivirus, and backup agents, use unique identifiers to communicate with the parent system. When these applications are cloned, the identity of the agents might be duplicated so that the applications no longer function as intended. Windows has a default list of applications that can be cloned. To view the list of applications installed on a system but not in the default inclusion list, run the following PowerShell command:

```
Get-ADDCCloningExcludedApplicationList
```
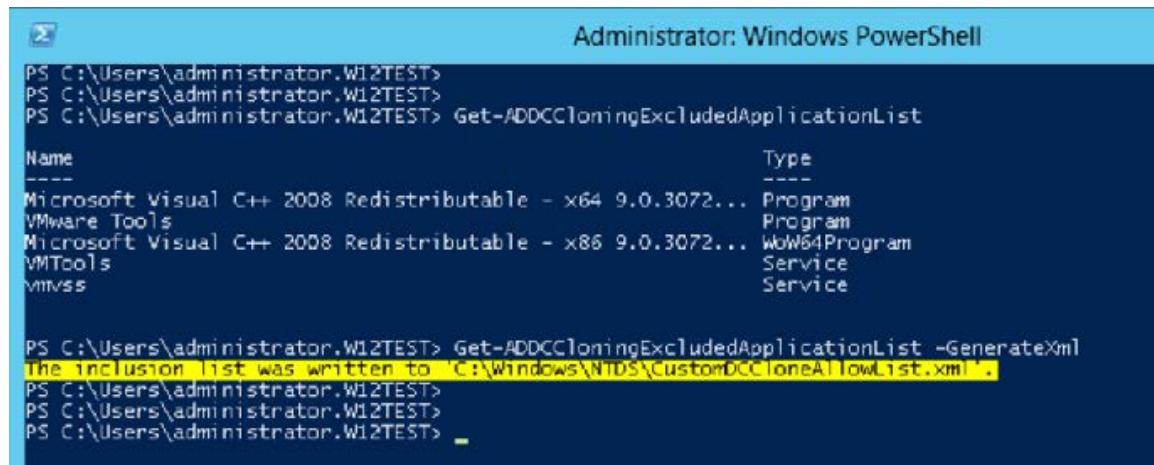
Note   VMware Tools is safe to clone.

Before commencing a domain controller cloning operation, verify with the software vendor whether or not the additional software that is installed is safe to clone. If not, remove the applications before continuing. Installed applications that are safe for cloning must be added to the `CustomDCCloneAllowList.xml` file in order to continue with the preparation of the DC for cloning. After it is determined that the installed applications are safe to clone, the applications must be added to the CustomDCCloneAllowList.xml file by running the following PowerShell command.

`Get-ADDCCloningExcludedApplicationList –GenerateXml`

The output from both commands is shown in the following screen.

Figure 10. Generating a Custom Application Allow List

The next step to prepare the source domain controller for cloning is to add it to the Active Directory global security group named *Cloneable Domain Controllers*. Membership in this security group is what flags the Domain Controller as being available for cloning. This is also a role - separation (RBAC) mechanism to ensure that the Directory Service administrator (or someone with such equivalent access) is, at least, involved in the decision-making process of whether or not a domain controller can be cloned in the infrastructure.

The final step is to run the `New-ADDCCloneConfigFile` PowerShell command to perform the following functions:

- Verify that any additional software installed on the domain controller is included in the applications list.

- Verify that the source domain controller computer object is a member of the Cloneable Domain Controllers security group.

- Create the `DCCloneConfig.xml` file, which contains the identity information, such as host name and IP address, for the new domain controller.

When either of the first two verifications fails, the `DCCloneConfig.xml` file is not created.

The `New-ADDCCloneConfigFile` command accepts an array of parameters that specify the identity of the new domain controller. The following figure shows a subset of the available parameters. For a complete list of parameters, refer to the Microsoft TechNet article, *New-ADDCCloneConfigFile* at http://technet.microsoft.com/en-us/library/jj158947.aspx.

Figure 11. New-ADDCCloneConfigFile Command

After the `DCCloneConfig.xml` file has been created, the source domain controller can be shut down to prepare for cloning.

Note   Do not use hot cloning of the source domain controller virtual machine to create a new domain controller for production use. Hot cloning places the Active Directory database in a state that is incompatible for domain controller cloning and is not supported.

After the source domain controller has been shut down, the virtual machine can be safely cloned to create a new domain controller. Create the new domain controller using the clone feature in the vSphere client or by copying the VMDK of the source domain controller and creating a new virtual machine. The preferred approach is to use the vSphere client.

Note   Do not power on the source domain controller until you have successfully completed the cloning operation in the vSphere client (or have copied the `.vmdk` file).

When cloning the source domain controller virtual machine using the vSphere client:

- Do not customize the guest operating system.

- Edit the hardware before deploying the new clone to make sure that the network settings are correct.

- Before powering on the new clone, verify that its vmnic is connected to a network that has connectivity to the Active Directory infrastructure. This new domain controller must be able to communicate with the Active Directory to successfully complete the cloning process. When the new clone is unable to communicate with other domain controllers, the cloning operation fails. You might need to perform the cloning operation again.

For more information, refer to the Microsoft KB article, *DC cloning fails with no DSRM, duplicate source and control computer* at http://support.microsoft.com/kb/2742970.

You can power on the source domain controller immediately after you have completed the virtual machine cloning operation.

After the new virtual machine is powered on, the change in VM-Generation ID triggers the domain controller safeguard to be initiated, discarding the RID pool and changing the InvocationID of the local Active Directory database. The presence of the DCCloneConfig.xml file confirms the intention to clone a domain controller and the cloning process begins. After the cloning process is complete, a new domain controller is added to the Active Directory.

Note Because of the Active Directory tombstone lifetime of 180 days, verify that domain controllers used solely as sources for cloning are powered on and allowed to replicate at least once every 180 days.

If you use Windows Backup to back up the source domain controller's system state, delete the backup catalog on the new domain controller to avoid restoring a point-in-time snapshot from the source domain controller.

4. Best Practices for Virtualizing Domain Controllers

The virtualized Active Directory Domain Services environment requires the same level of planning and maintenance as a physical environment, with additional considerations about the virtualized infrastructure.
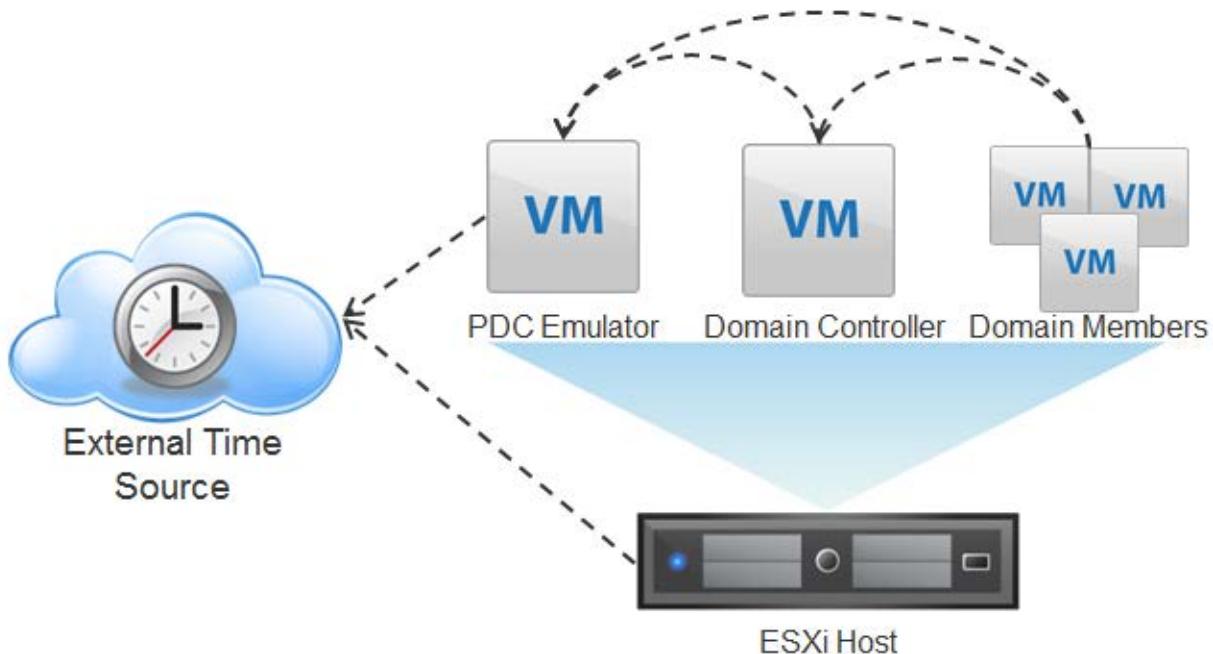
4.1 Timekeeping

A key principle of Active Directory design is providing a reliable time source for domain controllers. A domain controller relies on accurate time for Kerberos authentication and replication arbitration. Although Kerberos does not require highly accurate time synchronization across the Active Directory forest (the default maximum time deviation is five minutes), the use of time as an arbitration mechanism for replication requires that time across domain controllers should have as little divergence as possible.

The first step to proper timekeeping is to make sure that all ESXi hosts are configured to synchronize with a reliable time source. The time source can be a stratum 1 time source (GPS or hardware clock) or a stratum 2 or 3 time source (such as pool.ntp.org). When multiple sources are used, they must be at the same stratum level.
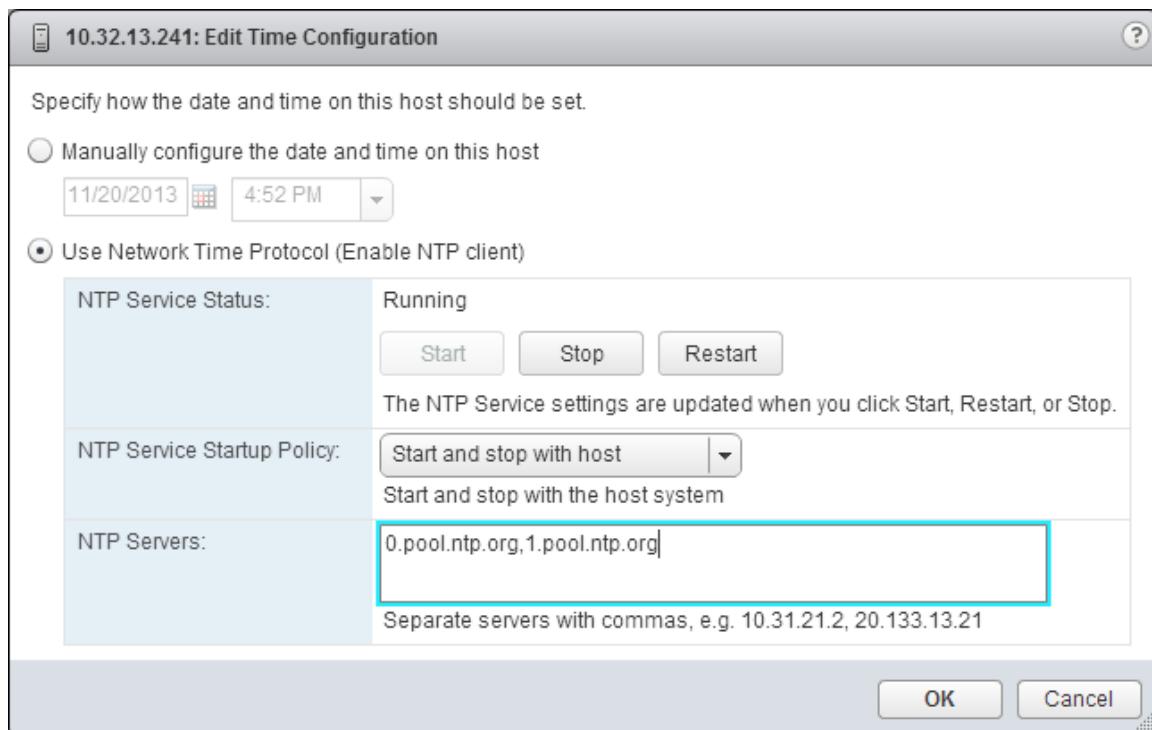
In addition, as shown in the following figure, the Active Directory forest should use the same time sources as the ESXi hosts. Do not synchronize ESXi hosts with a virtualized domain controller. During startup, VMware Tools performs a one-time synchronization of the guest operating system time based on the ESXi host time (even when VMware Tools time synchronization is disabled). Using the same time sources helps maintain an accurate ESXi host time, which is especially useful when the ESXi host is configured for Active Directory integration. Using the same time sources also maintains accurate virtual domain controller time during boot cycles.

Figure 12. Time Synchronization for ESXi Host and PDC Emulator

To configure Network Time Protocol (NTP) on an ESXi host, use the vSphere Web Client.
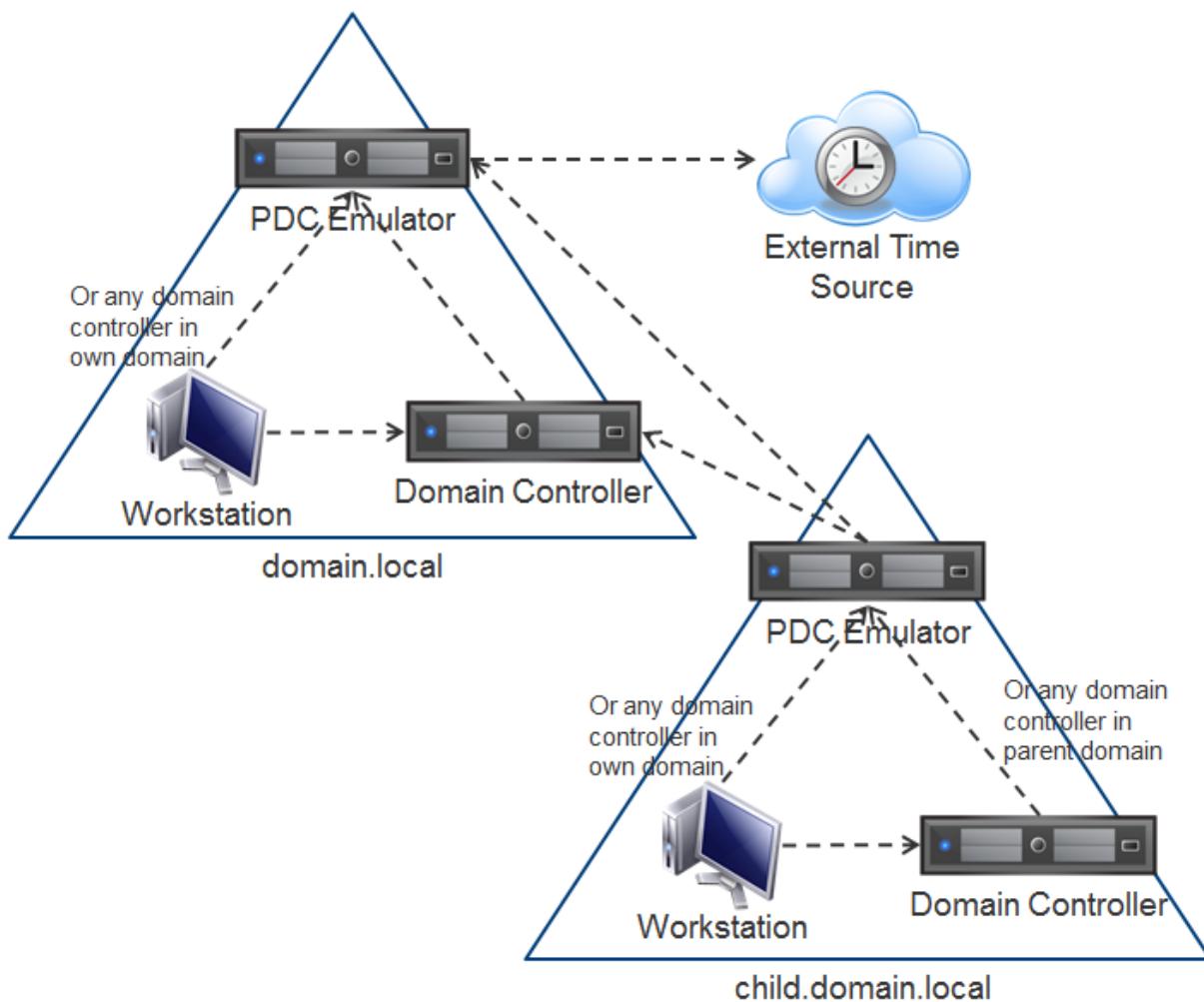
1. Select the host from within the vSphere Web Client and navigate to the **Manage** tab.

2. From the menu, select the **Settings** pane and **Time Configuration** option.

3. Click **Edit** to open the **Edit Time Configuration** window.

4. Select **Use Network Time Protocol (Enable NTP client)**.

5. Click **Start** to start the NTP service.

6. Set the **NTP Service Startup Policy** to **Start and stop with host**.

7. Enter a comma-separated list of NTP servers in the **NTP Servers** text field as shown in the following screen.

For more information, see *Configuring Network Time Protocol (NTP) on ESX/ESXi hosts using the vSphere Client* http://kb.vmware.com/kb/2012069.

The Primary Domain Controller emulator (PDCe) operations master role holder in the forest root domain is responsible for keeping accurate time for the entire forest. The following illustration shows how time synchronization works in a multi-domain Active Directory forest.

Figure 13. Time Synchronization Using a Domain Hierarchy

The PDCe in the forest root domain synchronizes time with a reliable time source that is external to the Active Directory forest. Peer domain controllers in the forest root domain synchronize time from the forest root PDCe. If the forest root contains workstations, the workstations can synchronize time from any domain controller in the forest root domain. All domain controllers in any subdomains within the forest can synchronize time from any domain controller in a parent domain. Workstations in subdomains synchronize time from domain controllers in their local domain.

The domain controller holding the PDCe operations master role must be configured to synchronize time with a reliable time source, preferably the same time source that the ESXi hosts are using. To synchronize time, domain controllers (and all Windows-based systems) use the Windows Time Service (W32Time), which by default is set to the hardware clock or BIOS on the forest root PDCe.

To configure the domain controller holding the PDCe operations master role so that it synchronizes with a reliable time source, run the following command:

```
w32tm /config /manualpeerlist:"0.pool.ntp.org,1.pool.ntp.org"
/syncfromflags:manual /reliable:yes /update
```

The command arguments are as follows:

- `config` – Use `w32tm` in configuration mode.

- `manualpeerlist` – List of reliable time sources from which to synchronize time in DNS or IP format. When defining multiple sources, the list must be space delimited and enclosed in quotation marks.

- `syncfromflags` – Source with which the NTP client is synchronized. The `manual` option configures the NTP client to use entries in the manual peer list (required for the PDCe). The `domhier` option (the default setting) configures the NTP client to use the domain hierarchy for time synchronization.

- `reliable` – Identifies a domain controller as a reliable time source.

- `update` – Notifies the Windows-based time service that a change has been made and causes the change to take effect.

When the PDCe role is transferred to (or seized on) another domain controller, you must run the previous command on the new domain controller. Run the following command on the domain controller that previously had the PDCe role to make sure it is following the domain hierarchy for time synchronization:

```
w32tm /config /syncfromflags:domhier /reliable:no /update
```

For more information about managing the Windows Time Service, see *Managing the Windows Time Service* at http://technet.microsoft.com/en-us/library/cc737124(v=ws.10).aspx.
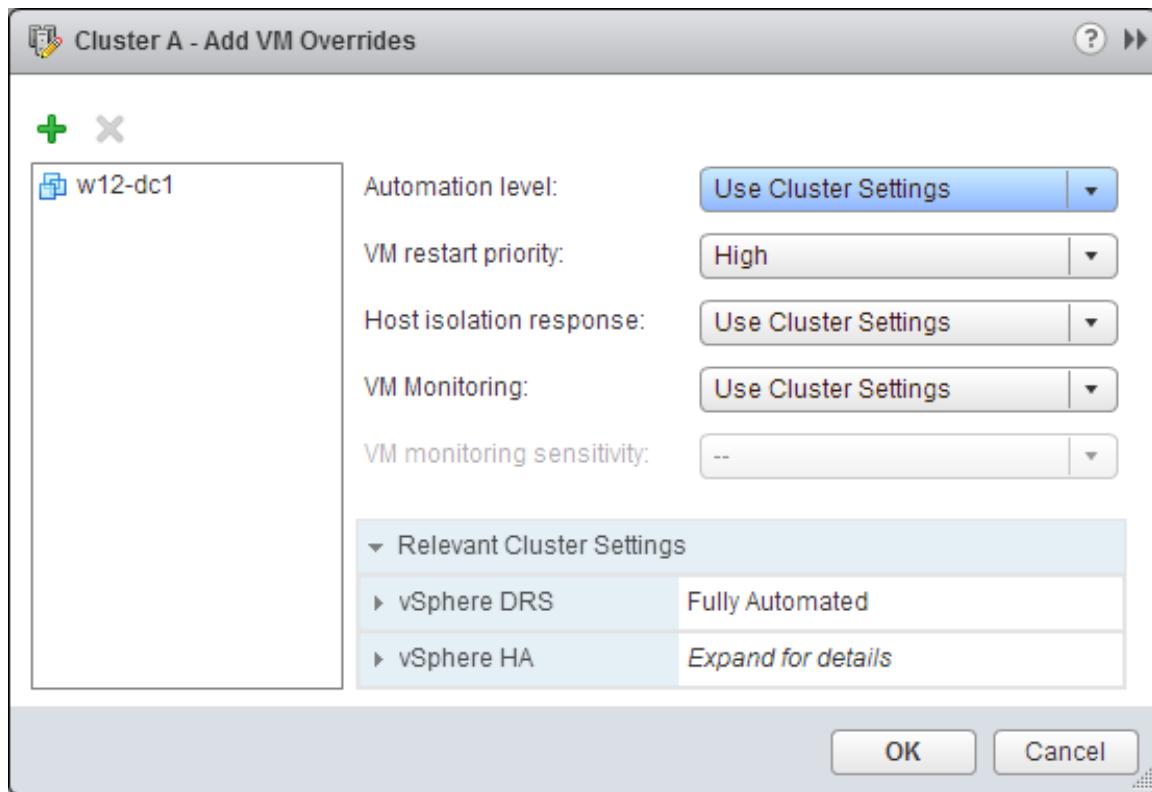
4.2   vSphere HA and vSphere DRS

VMware recommends that domain controllers use VMware vSphere High Availability (HA) and VMware vSphere Distributed Resource Scheduler™ (DRS). With vSphere HA, in the event of an ESXi host failure, all virtual machines that were running on the host at the time of the failure will experience a hard shutdown, but they are automatically restarted on other hosts that are running within the vSphere cluster. vSphere HA reduces recovery time to several minutes, as opposed to the length of time it takes an administrator to receive notification and respond to the outage.

vSphere HA provides the following options that are helpful to virtualized domain controllers:
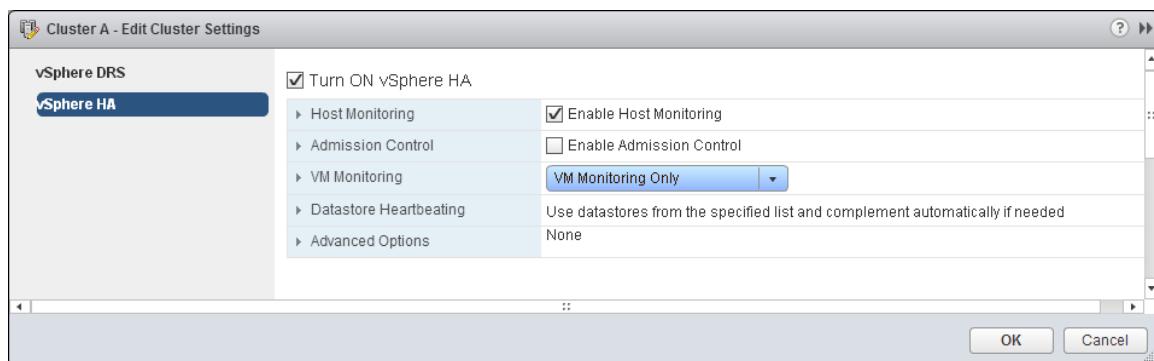
- VM restart priority – By default, vSphere HA assigns no preference to the order in which virtual machines are restarted after a failure. In most cases, a scaled-out deployment of domain controllers can support a brief outage. Domain controller virtual machines can be assigned a higher restart priority if necessary. Enable the individual virtual machine for high-priority restart as shown in the following screen:

Figure 14. VM Restart Priority

- VM Monitoring – When enabled, a heartbeat is established between VM Tools installed in the guest operating system and the ESXi host. If this heartbeat is lost, vSphere HA examines the network and the storage I/O for the virtual machine to determine if the guest operating system has stopped functioning or if the problem is isolated to heartbeat communication. If the storage I/O is not available, the vSphere HA initiates a restart of the virtual machine. Virtual machine monitoring can help recover a Windows virtual machine.

Enabling virtual machine monitoring is shown in the following figure.

Figure 15. Enabling Virtual Machine Monitoring in vSphere HA
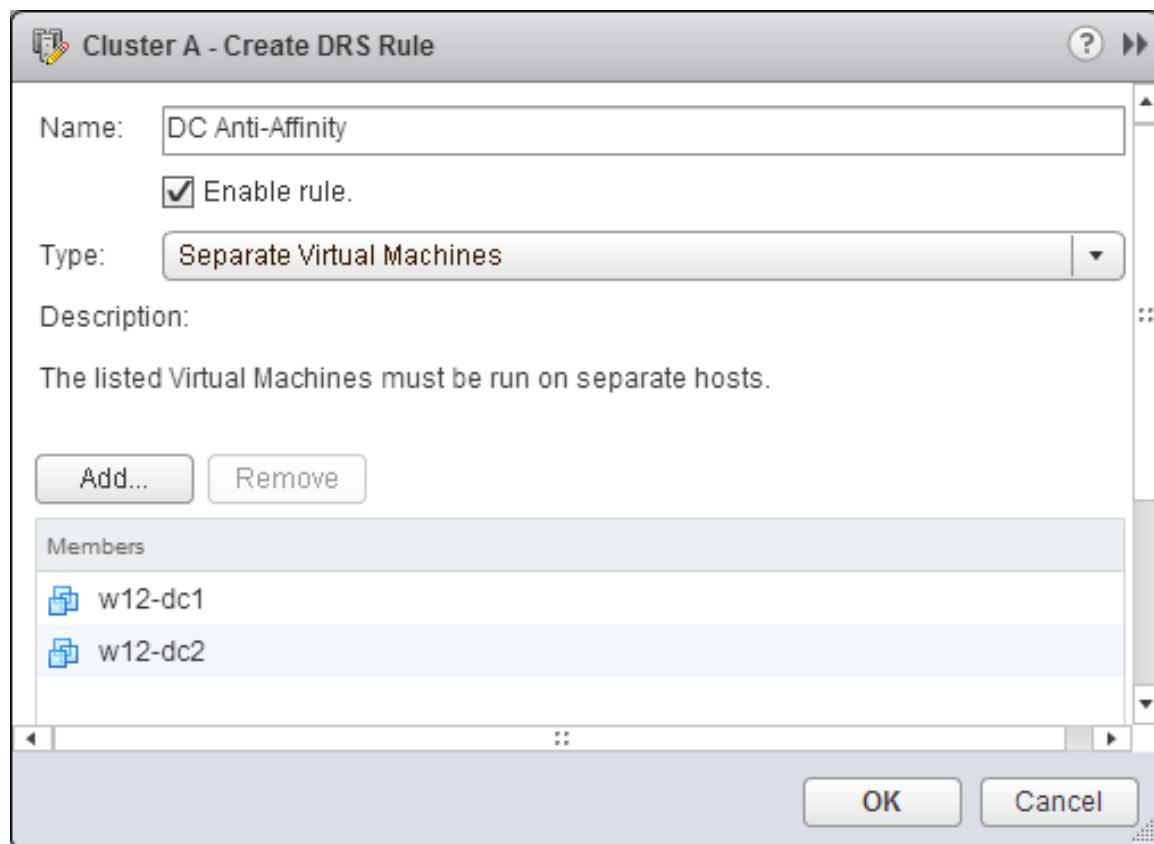
vSphere DRS manages compute resources within a vSphere cluster to balance workloads among available resources based on each virtual machine's utilization and entitlement. When available resources within a vSphere cluster change, due either to host maintenance or additional capacity coming online, DRS determines on which hosts workloads can best obtain required resources. In fully automated mode, DRS uses vSphere vMotion to migrate virtual machines between ESXi hosts with no downtime to the virtual machines or running applications.

An added feature available in DRS is the ability to create rules as to the placement of virtual machines. With DRS rules and in conjunction with virtual machine and host groups, virtual machine placement can be controlled at a very granular level. For instance, a DRS anti-affinity rule can be created to keep domain controller virtual machines from running on the same host. Using virtual machine and host groups, an administrator can configure half the domain controllers to prefer to run on ESXi hosts in one blade chassis or rack, while the other half prefers to run on another, reducing the impact of a localized failure.

Consider the following recommendations when using vSphere DRS on clusters hosting virtualized domain controllers:

- Enable vSphere DRS in fully-automated mode – When configured for fully-automated mode, DRS makes and enforces its recommendations automatically to provide the best balance of resources within a vSphere cluster. This also allows for any DRS rule violations to be remedied during the next DRS evaluation cycle (every 5 minutes).

- Create DRS anti-affinity rules – Running two domain controllers on a single ESXi host is not ideal, but the risk is minimal if more than two domain controllers exist within the site. VMware suggests that you keep them as separated as possible to avoid impacting multiple domain controllers at the same time. Do this by creating DRS anti-affinity rules and placing a few or all of the domain controller virtual machines into the rule.

Figure 16. DRS Anti-Affinity Rule

- Create DRS host and virtual machine groups – You can create logical groups of hosts and virtual machines within a vSphere cluster. Groupings of hosts and virtual machines can then have affinity or anti-affinity rules applied to them. If a vSphere cluster spans multiple failure domains, such as a blade chassis, consider creating a host group per blade chassis and splitting domain controllers into two virtual machine groups. Virtual machine-to-host rules can then be used to keep domain controllers split between the hosts.

Figure 17. Virtual Machine and Host DRS Groups
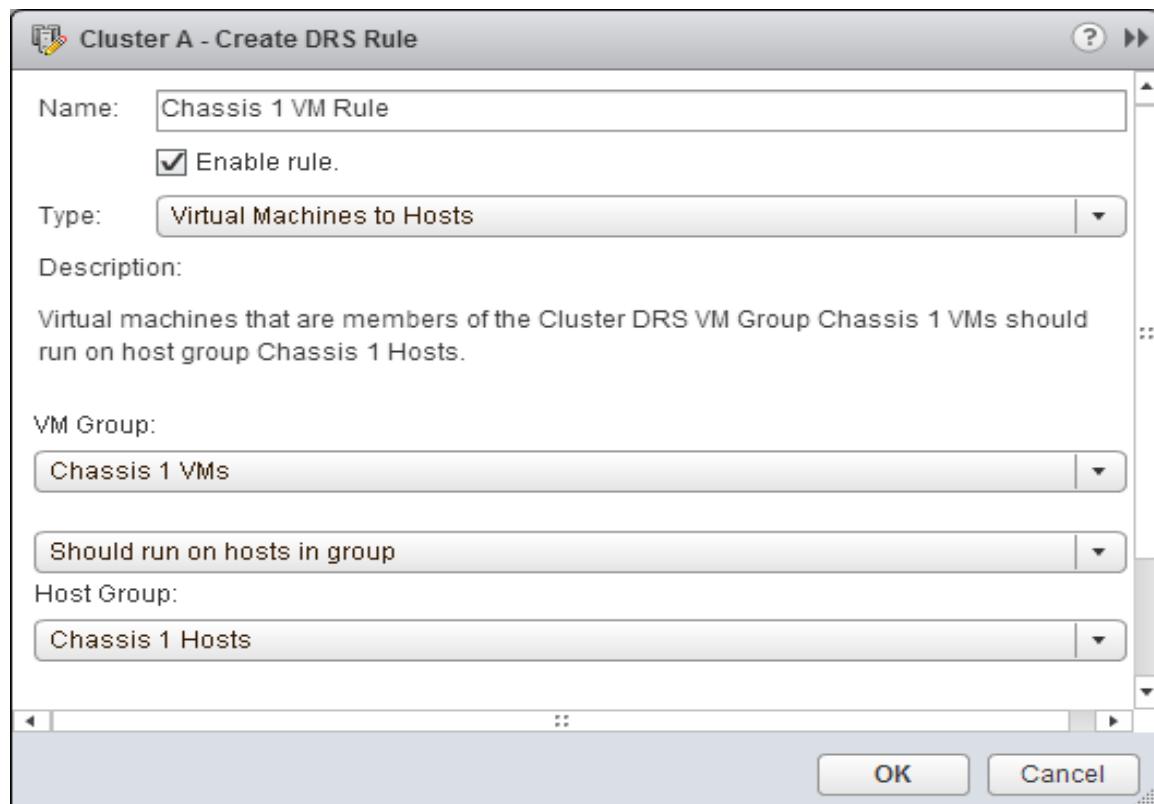
- Create soft virtual machine-to-host rules – You can use DRS rules to keep (or separate) virtual machines in a group on hosts in a group. These rules have two modes; soft (should run on) and hard (must run on). A soft rule can be violated if no other option exists. For example, when a chassis fails and its virtual machines have a soft rule that retains them on the failed set of hosts, the virtual machines are recovered on any remaining hosts, even in violation of the rule. A hard rule cannot be violated without administrator intervention. Hard rules are typically used for licensing purposes. Soft anti-affinity rules are preferred for virtualized domain controllers.

Figure 18. Should-Run-On DRS Rule

4.3   Domain Controller Golden Templates

Rapid deployment of domain controllers has been a major incentive for embracing the virtualization of domain controllers. Until recently, administrators were required to prepare the base operating system image to become a domain controller, provision the new domain controller using `dcpromo`, install any supporting software, such as monitoring or backup agents, then wait for replication to complete before the new domain controller was ready for production. Deploying multiple domain controllers could take hours or days.

With the introduction of the domain controller cloning feature, vSphere and Windows Server 2012 eliminate most of the manual tasks previously required to deploy domain controllers. Consider the following recommendations when choosing to use this method to deploy domain controllers:

- vCenter Server and vSphere must both be at a minimum of version 5.0 update 2.

- All software running on a source domain controller must be validated as able to be cloned and added to the custom exclusion list. Confirm with the software vendor whether or not the software can be installed in the template prior to cloning. If unknown, consider removing the software prior to cloning. VM Tools is safe to be installed prior to cloning.

- For security purposes, the Cloneable Domain Controllers security group must remain empty in-between cloning of domain controllers.

- Any domain controller considered to be a reference for cloning must be powered on and running on the network when no cloning operations are being conducted. Keeping a domain controller off of the network longer than the tombstone lifetime (180 days by default) results in replication problems that have to be manually corrected before replication will resume.

- If you use Windows Backup to back up the system state of the domain controller that is acting as a template, delete any backup history on the new domain controller after cloning and run a new backup immediately afterward.

- Shut down the source domain controller before initiating the virtual machine clone operation. Hot cloning is a mature and very useful feature that has been available with vSphere for many years. However, the Active Directory database must be shut down properly and cleanly before being cloned. The shutdown is required due to the way in which Active Directory determines whether or not domain controller safeguards are initiated. Hot cloning a domain controller for the purpose of provisioning a new domain controller for production is not supported by VMware or Microsoft.

4.4 Disaster Recovery of Domain Controllers

When properly designed, the same multimaster replication and service location mechanisms that make Active Directory highly available within a site, also make the service site resilient and disaster recovery ready, when properly designed. Proper design means that domain controllers have been deployed in geographically dispersed locations.

The portability of virtual machines is an effective enabler for providing disaster recovery for applications by replicating the contents of a virtual machine to another data center. In the event of a disaster, the replica can be powered on and (with minimal configuration) the virtual machine is back in service. Traditionally, the concern with this approach was how to keep the virtual machine files up to date on a regular basis. The latest versions of vSphere support two methods for keeping virtual machines up to date – VMware vSphere Replication and array-based replication. Both solutions can be paired with the VMware vCenter Site Recovery Manager™ to automate the recovery and configuration of virtual machines in the case of a disaster, planned failover, or planned migration. How can this solution be used with virtualized domain controllers?

The common answer is that it is typically best to use Active Directory's native replication capabilities to provide protection for domain controllers using a minimal number of domain controllers in each site. Given the low maintenance and performance requirements of most domain controllers, this setup does not incur significant overhead. However, there are scenarios where using tools such as vCenter Site Recovery Manager for the recovery of domain controllers can prove helpful.
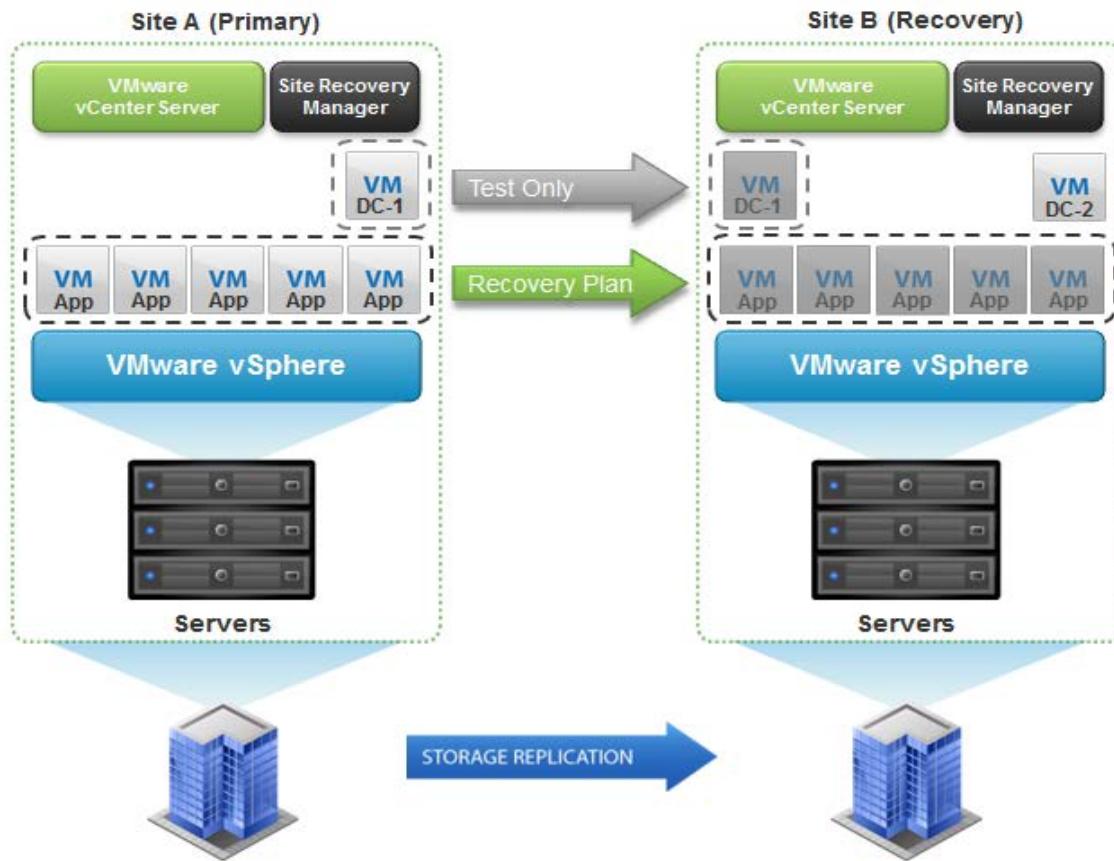
4.4.1 Using Domain Controllers During Disaster Recovery Testing

One of the features of vCenter Site Recovery Manager is the ability to test a disaster recovery plan without impacting the production environment. This is accomplished by using point-in-time copies of the virtual machines (either array-based or virtual machine snapshots) and an isolated network. When running in test mode, virtual machines are recovered based on the recovery plan. The plan might involve recovering virtual machines in a particular order and reconfiguring network settings. By executing these non-disruptive test recovery plans, administrators can test their configuration to verify that applications recover as expected.

Some applications depend on the availability of Active Directory Domain Services (or DNS). In this situation, you can add a domain controller from the protected site to the recovery plan whereby the domain controller is used only when the recovery plan runs in test mode. Or, you can clone a domain controller to the isolated network from the recovery site each time a test recovery plan is initiated.

The first option is automated and self-contained. As shown in the following figure, the application virtual machines are part of two recovery plans, the production recovery plan and a test recovery plan. To be brought online, the application requires Active Directory. During a test failover, the application virtual machines are recovered to an isolated network with no access to Active Directory. To satisfy the Active Directory requirement, the test recovery plan includes DC-1, which is replicated to Site B strictly for disaster recovery testing purposes. When initiated, the test recovery plan powers on all application servers and DC-1, enabling the application to come online. In a real failover, only the application servers fail over and (through the DC Locator process) discover DC-2.
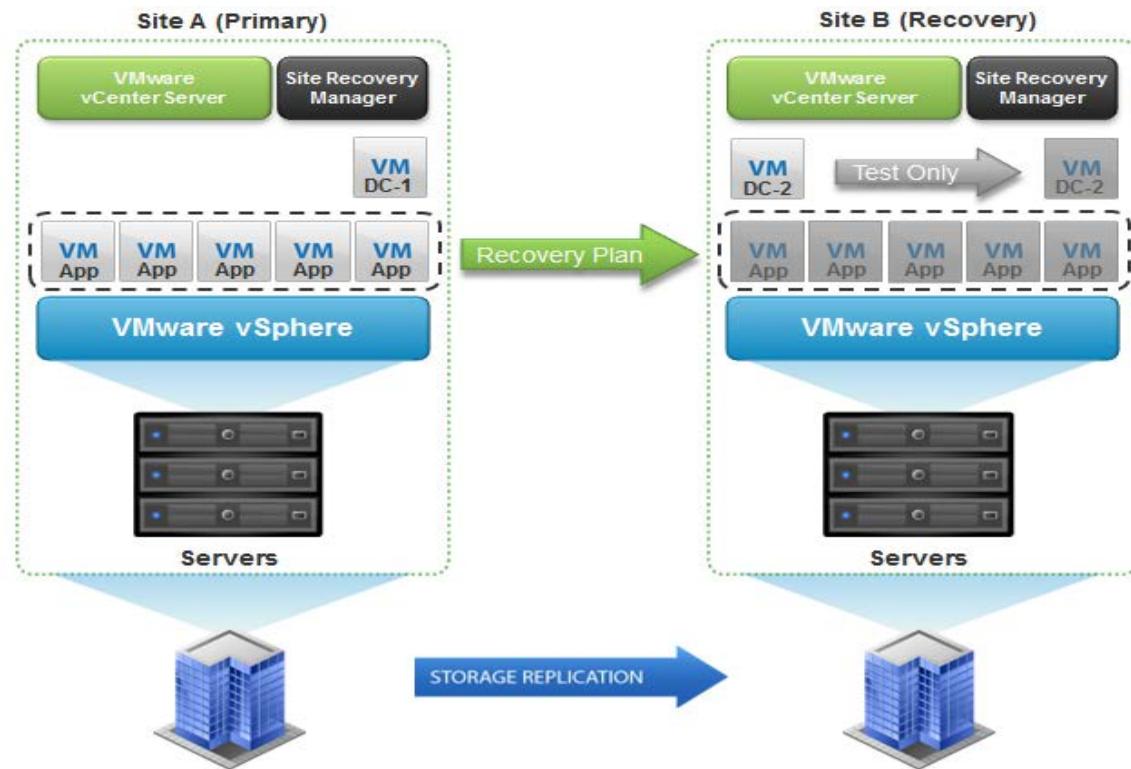
Figure 19. Using Primary Site Domain Controller During Recovery Plan Testing

The second option is to clone a local domain controller during the execution of the recovery plan. Depending on the OS version of the source domain controller, the clone operation can be online (hot clone, which is not supported for a Windows Server 2012 domain controller), offline and scripted, or manual.

During a test recovery plan execution, the clone is initiated and after completion, the application servers are brought online in isolation. Through the DC Locator process, the application servers discover DC-2, enabling them to be brought online. After the test, the cloned domain controller is deleted. Although it closely resembles what would happen during an actual disaster; this option requires extra scripting or manual intervention to complete the cloning and cleanup.

Figure 20. Cloning Recovery Site Domain Controller During Recovery Plan Testing
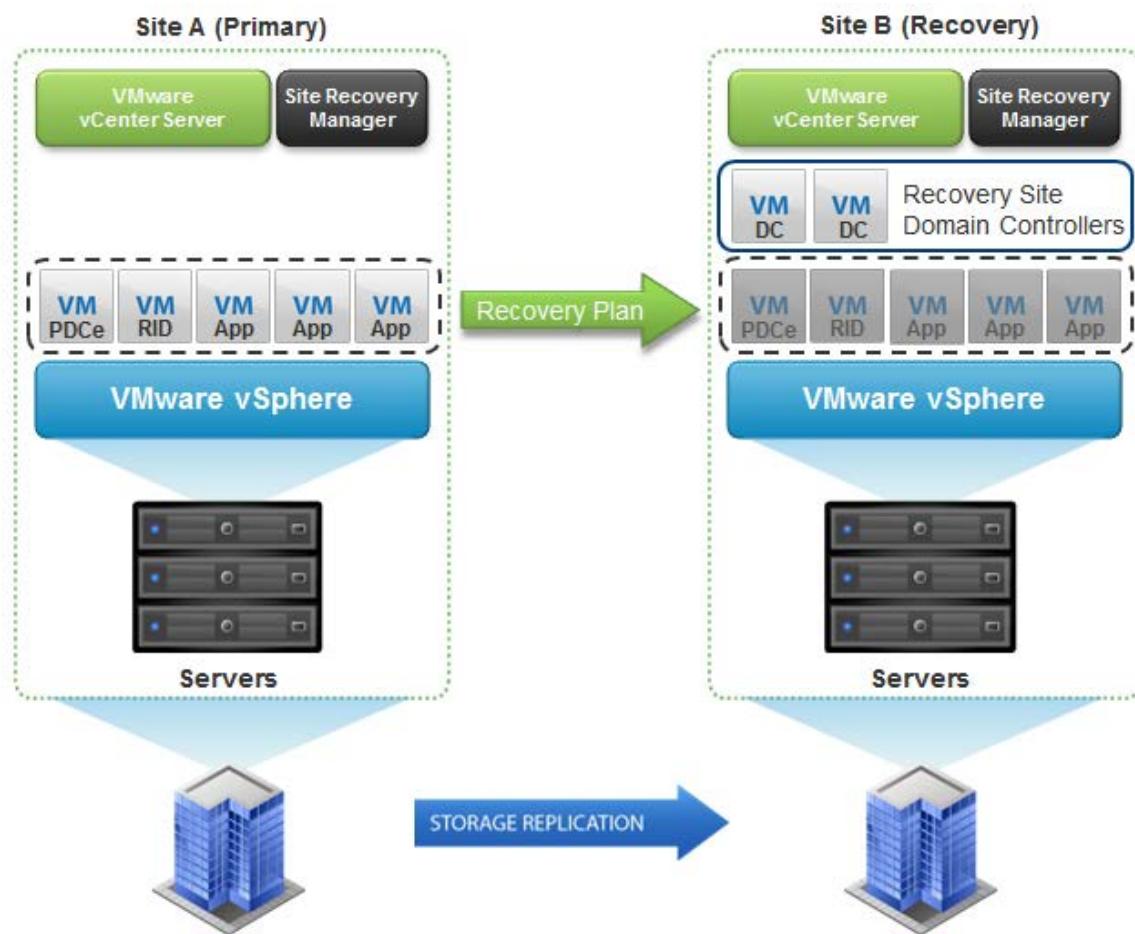
4.4.2 Protecting Operations Master Role Holders

The *Operations Master* role holders (also known as the FSMO role holders) provide services that can only be performed by a single domain controller. For instance, the Schema master role holder is the only domain controller on which a Schema change can occur. If the Schema master is offline during an attempted Schema change, the change fails. These roles can be transferred in the case of a failure, but doing so during an unplanned recovery event might not be the desired approach. There are five operations master roles, two forest-wide roles, and three domain-wide roles. By default, the operations master roles are located on the first domain controller in the forest (all five roles) and on the first domain controller in the domain (three domain-specific roles). Best practice recommends distributing the roles across two or more domain controllers within a domain.

In this scenario, the recommendation is to maintain a set of domain controllers on the recovery site and protect the operations master role holders using replication. Whether or not all five roles should be protected depends on the environment, but (at a minimum) consider protecting the RID and PDCe master roles. The PDCe is the default domain controller for many tasks, such as password reset and verification, group policy changes, and authoritative time. If you are planning to leverage VM-Generation ID and virtualization safeguards, you must protect the RID master.

As part of virtualization safeguards (activated during the recovery of a domain controller using replication), a recovered domain controller discards its local RID pool and contacts the RID master to obtain a new pool. If the RID master for the domain is not available, the domain controller is unable to create any new security principals. Often these two roles are co-located on a single domain controller making it straight forward to protect both roles. The other roles are less likely to impact any recovery effort and can be seized if the outage is an extended one.

Figure 21. Protecting Operations Master Role Holders

5.   Conclusion

Many organizations are striving to complete virtualization of their data centers, including business critical and infrastructure applications. In the past, there were challenges with being able to confidently virtualize these applications, but many of the challenges have been resolved by improvements in the technology and access to information about how to virtualize critical applications.

Active Directory is the core directory and authentication source for many organizations. Although some virtualization of Active Directory domain controllers is common, organizations remain cautious about virtualizing 100 percent of their Active Directory infrastructure. With the release of Windows Server 2012, virtual machine snapshots and cloning of virtualized domain controllers is now possible, relatively safe and supported. Complete virtualization of an Active Directory infrastructure is achievable when best practices are followed.

Appendix A: Testing Domain Controller Cloning

This exercise provides a process for testing and verifying the successful cloning of a virtualized domain controller in a vSphere environment. For detailed information on the cloning process, please refer *to Virtualized Domain Controller Deployment and Configuration* at http://technet.microsoft.com/en-us/library/jj574223.aspx.

In this scenario, the following tasks have already been completed.

- vSphere 5.5 (ESXi and vCenter Server) installed and operational

- Windows 2012 Active Directory domain created (w12test.local)

- All operations master roles on a single domain controller (w12-dc100)

- A second domain controller (w12-dc102) deployed in the domain

Prepare the Source Domain Controller
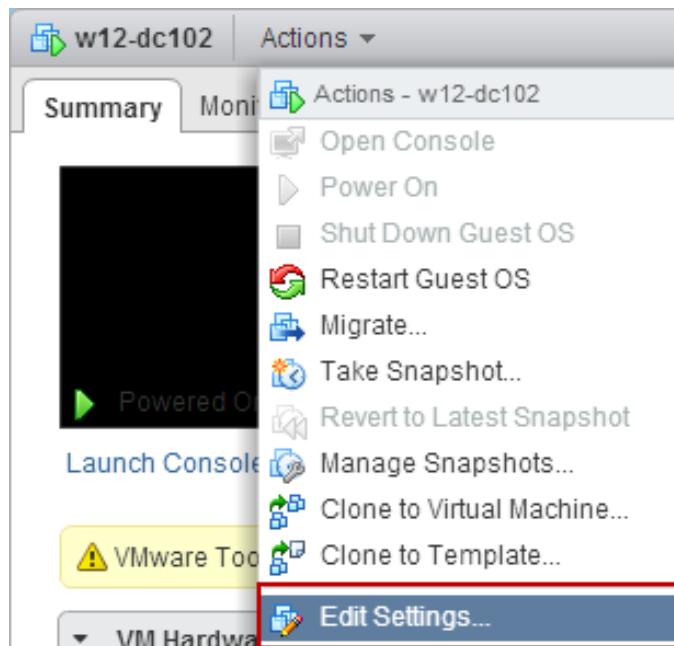
Preparing the source domain controller requires you to validate that the hypervisor supports VM-Generation ID, verify the set of applications that are installed, and enable the domain controller to be cloned through security group membership.

Validate the Hypervisor Supports VM-Generation ID
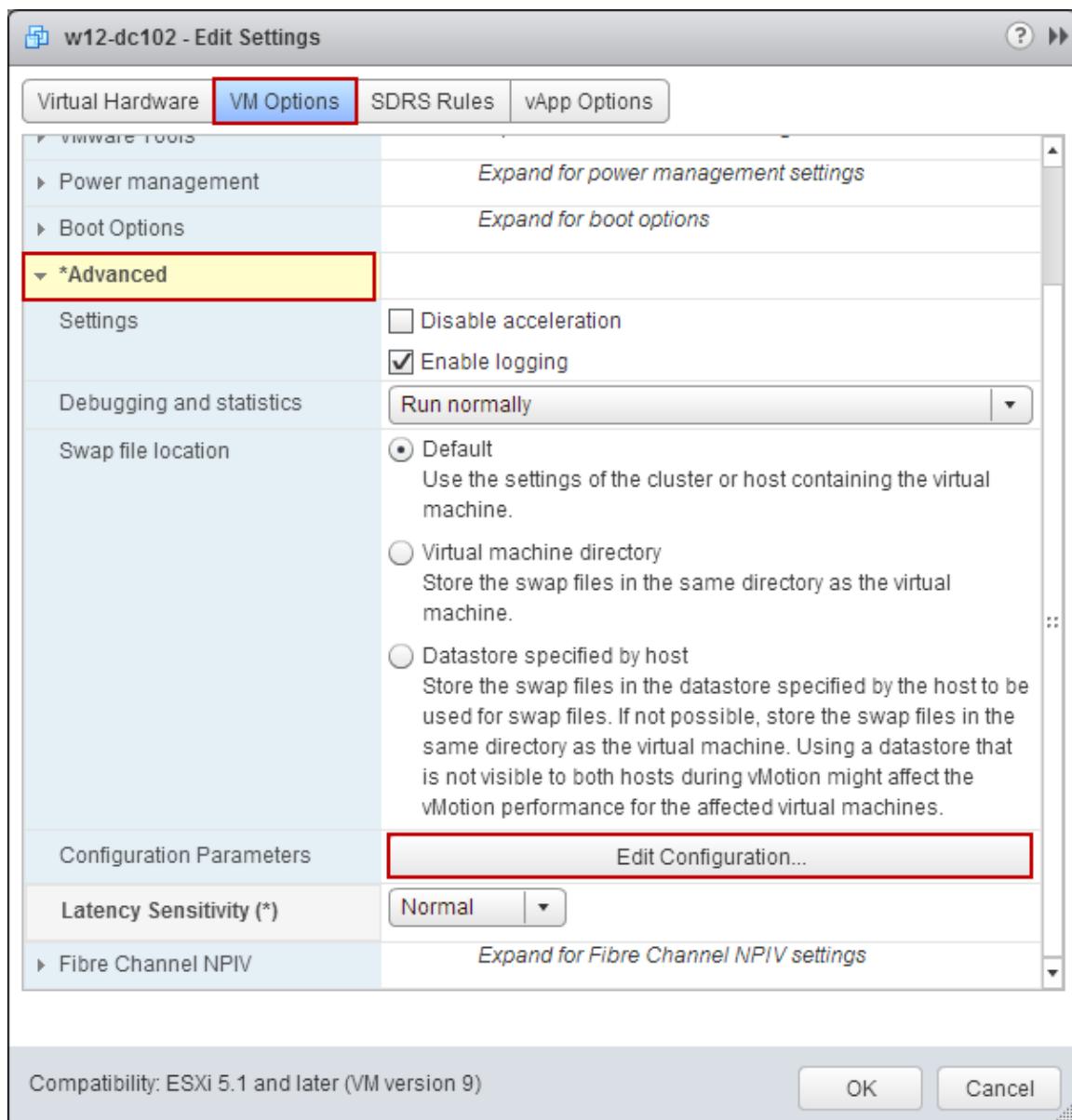
There are different options available to validate that the hypervisor supports the VM-Generation ID feature. If the source domain controller virtual machine is powered off, the value of the VM-Generation ID can be checked in the advanced options of the virtual machine.

vSphere uses the configuration parameter `vm.genidX` or `vm.genid` (depending on the vSphere version) to store the generation ID of the virtual machine.
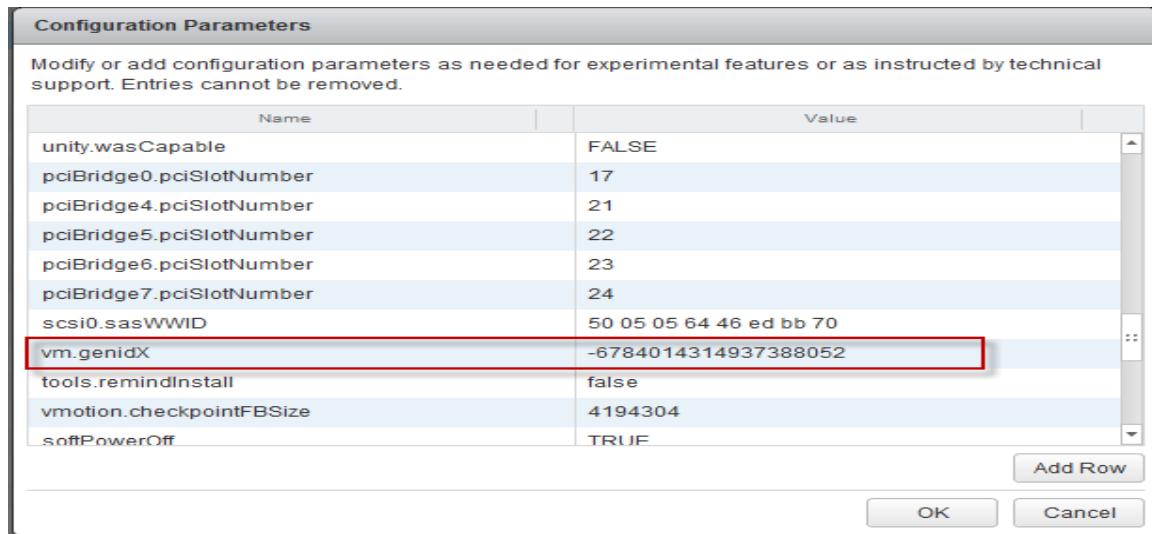
1. Log in to the VMware vSphere Web Client and navigate to the source domain controller virtual machine object. From the **Actions** menu, select **Edit Settings**.
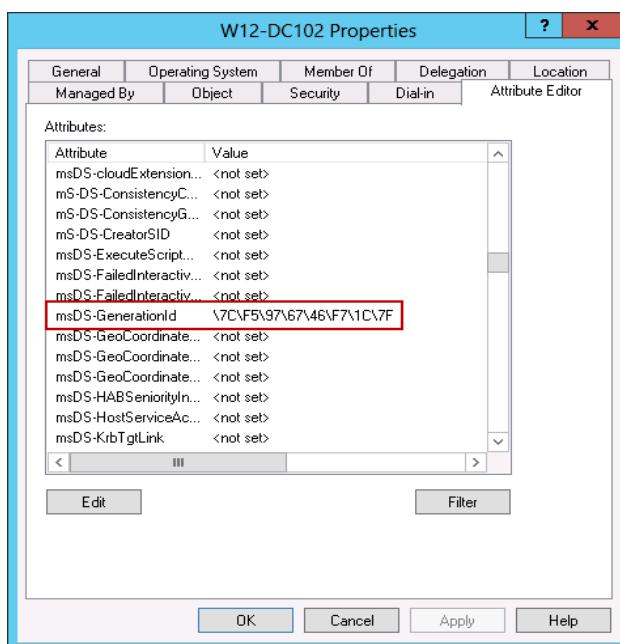
2. From the Edit Settings dialog box, select **VM Options**.

3. Expand the **Advanced** menu and select **Edit Configuration**.
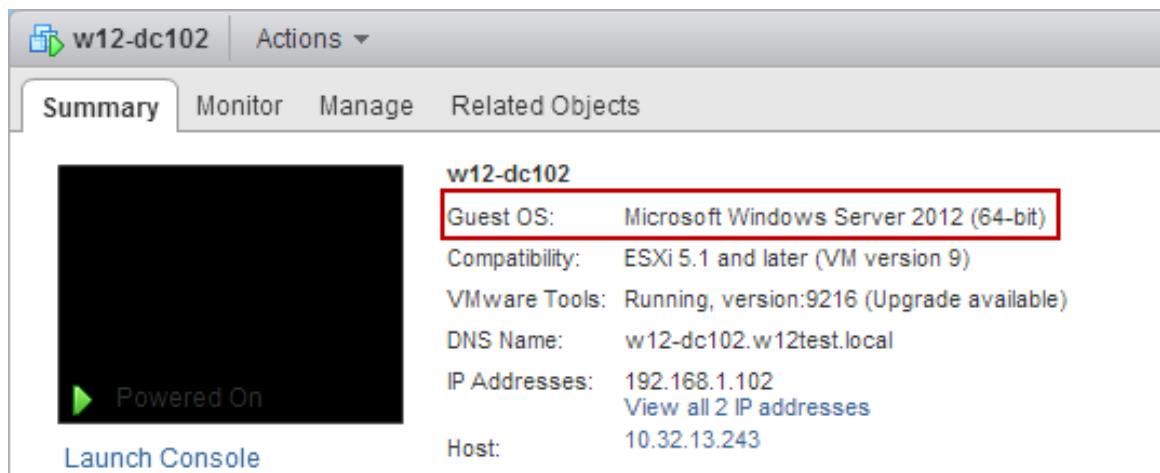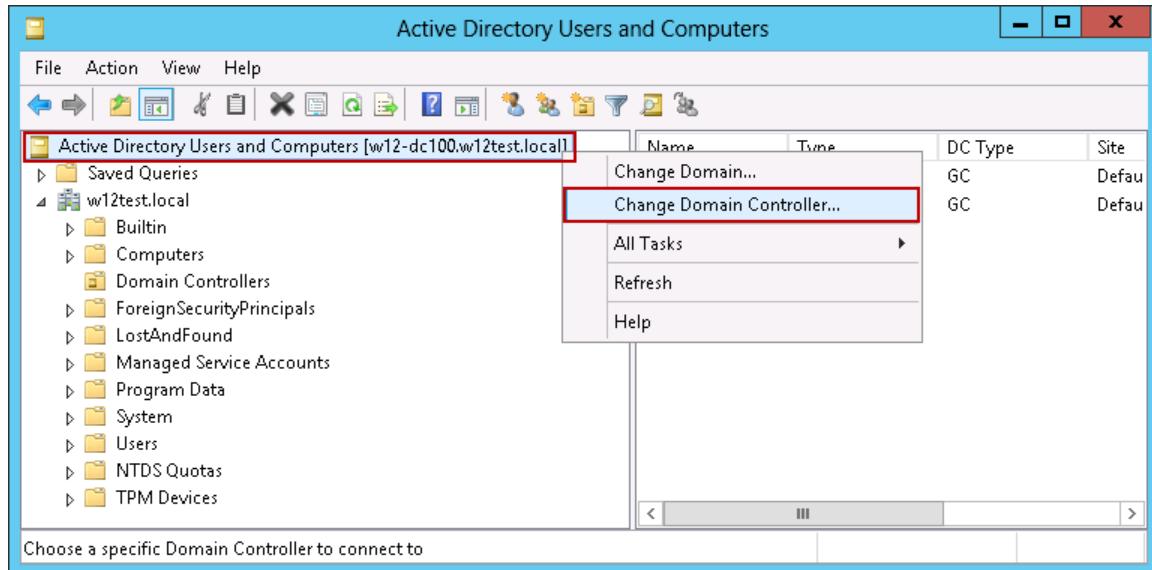
4. Scroll to locate the `vm.genidX` parameter and verify that it is populated.

5. If the source domain controller is powered on, you can check the VM-Generation ID by querying the value of the msDS-GenerationId attribute on the domain controller. The msDS-GenerationId attribute is not a replicated value and must be checked locally on the source domain controller.

   a. Log in to the source domain controller and launch **Active Directory Users and Computers**.

   b. From the menu bar select **View, Advanced Features**. The **Attribute Editor** tab is an advanced feature not viewable by default.

   c. Navigate to the **Domain Controllers Organizational Unit (OU)** and open the properties for the source domain controller.

   d. In the **Attribute Editor** tab, scroll to find the msDS-GenerationId attribute.
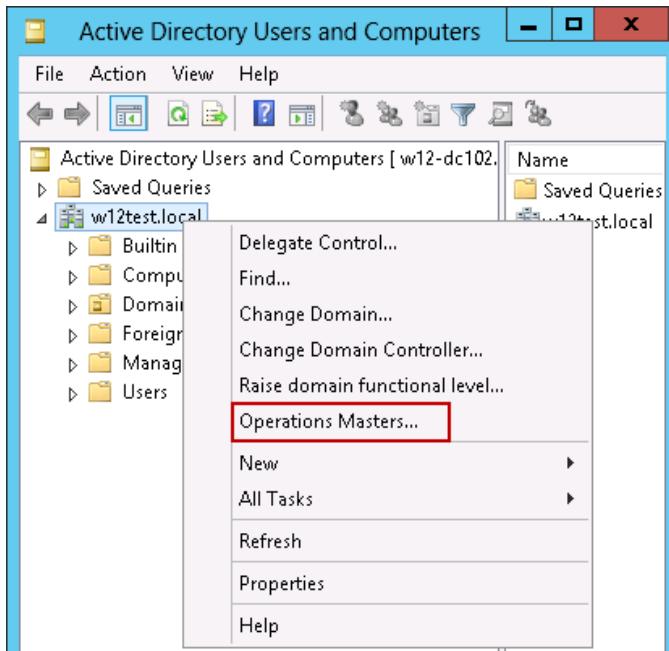
If the attribute appears as **<not set>**, verify that you have connected **Active Directory Users and Computers** to the local domain controller (and *not* a remote domain controller) and that the virtual machine is configured with a guest operating system such as Windows 2012 or later. Both are shown in the following figures.
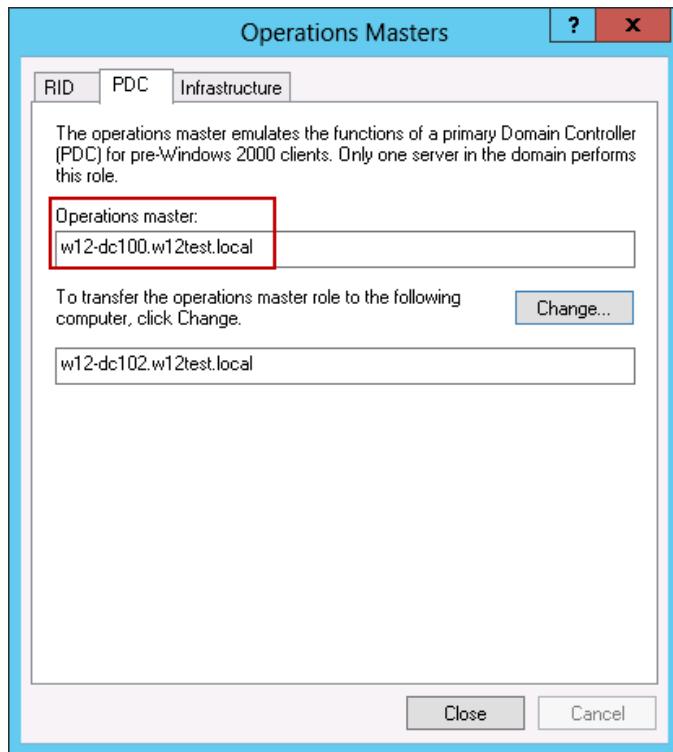
Verify that the PDC Emulator is Running on Windows 2012 or Later

Virtual domain controller cloning requires the operations master role holder for the PDC Emulator role to run on Windows Server 2012. To verify, perform the following steps.

1. Open **Active Directory Users and Computers**, right-click the domain object within the directory tree, and select **Operations Masters** as shown in the following figure.

2. Open the **PDC** tab and identify the host name of the current PDC Emulator as shown in the following figure.

Alternatively, you can use the PDC Emulator to determine the operating system version.

1. Log in to the PDC Emulator.

2. Use the following PowerShell command to find the PDC Emulator of the domain and the Windows version.

```
Get-AdComputer (Get-ADDomainController –Discover –Service
"PrimaryDC").name –property * | select name,operatingsystem
```

Validate the Applications Installed on the Source Domain Controller

The applications installed on the source must be listed in the default DCclone Allow List (`%systemroot%\system32\DefaultDCCloneAllowList.xml`) or in the custom allow list to allow cloning to proceed.

1. To identify whether or not any installed applications require removal, or can be cloned, but must be included in a custom allow list, run the following PowerShell command.

    `Get-ADDCCloningExcludedApplicationList`

    The output displays applications or services that are not in the default clone list. VMware services (VMTools and vmvss) and applications (VMware Tools) can be cloned. Any other applications should be verified with the vendor.

2. Remove any application that is not safe for cloning, such as monitoring or backup agents. Then verify the list again.

3. To add applications to the custom allow list, run the following command.
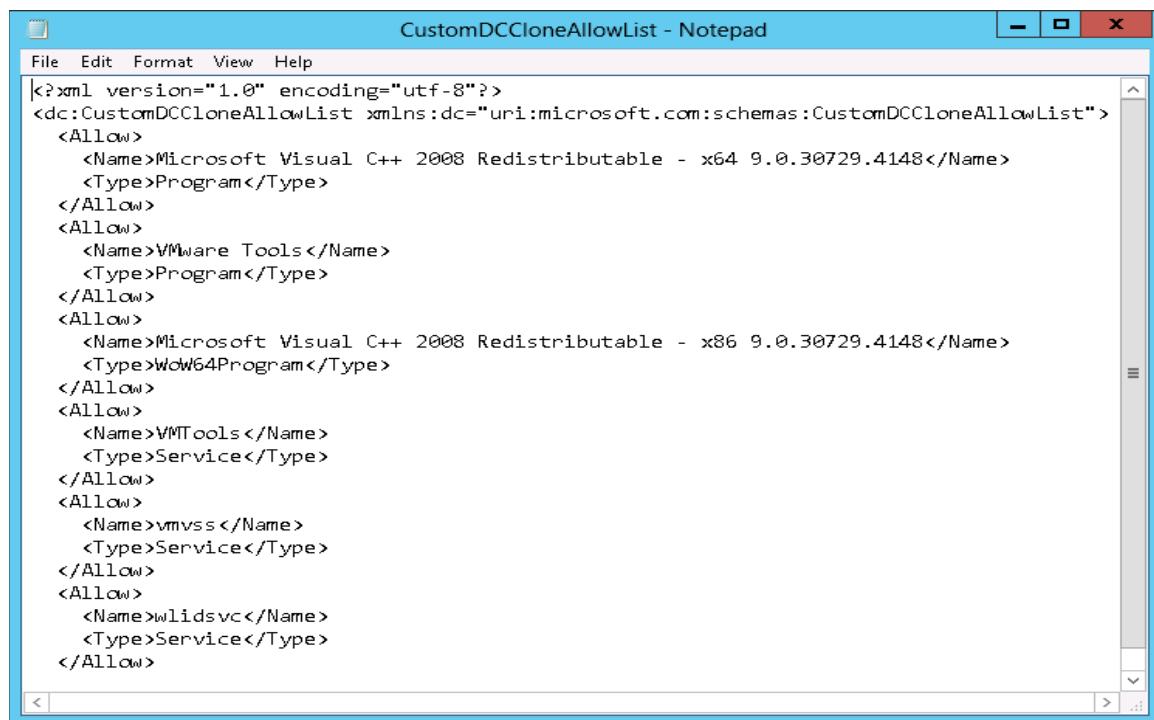
    `Get-ADDCCloningExcludedApplicationList –GenerateXml`

    As shown in the following screen, the custom allow list is written to `CustomDCCloneAllowList.xml` within the NTDS folder.

The custom allow list is shown in the following figure.

Figure 22. Custom Allow XML File

Allow Domain Controller Cloning

Domain controller cloning is protected by a new right in Windows 2012 Active Directory. This right is granted to the Active Directory Global Security group, Cloneable Domain Controllers. When the PDCe role is transferred to a Windows 2012 domain controller, this group is created but contains no members. To allow the source domain controller to be cloned, it must be added to the Cloneable Domain Controllers group.

1. Open **Active Directory Users and Computers**.

2. Navigate to the **Users** container, and open the **Cloneable Domain Controllers** group as shown in the following figure.

3. Add the source domain controller to the group as shown in the following figure.

Create the Domain Controller Configuration File
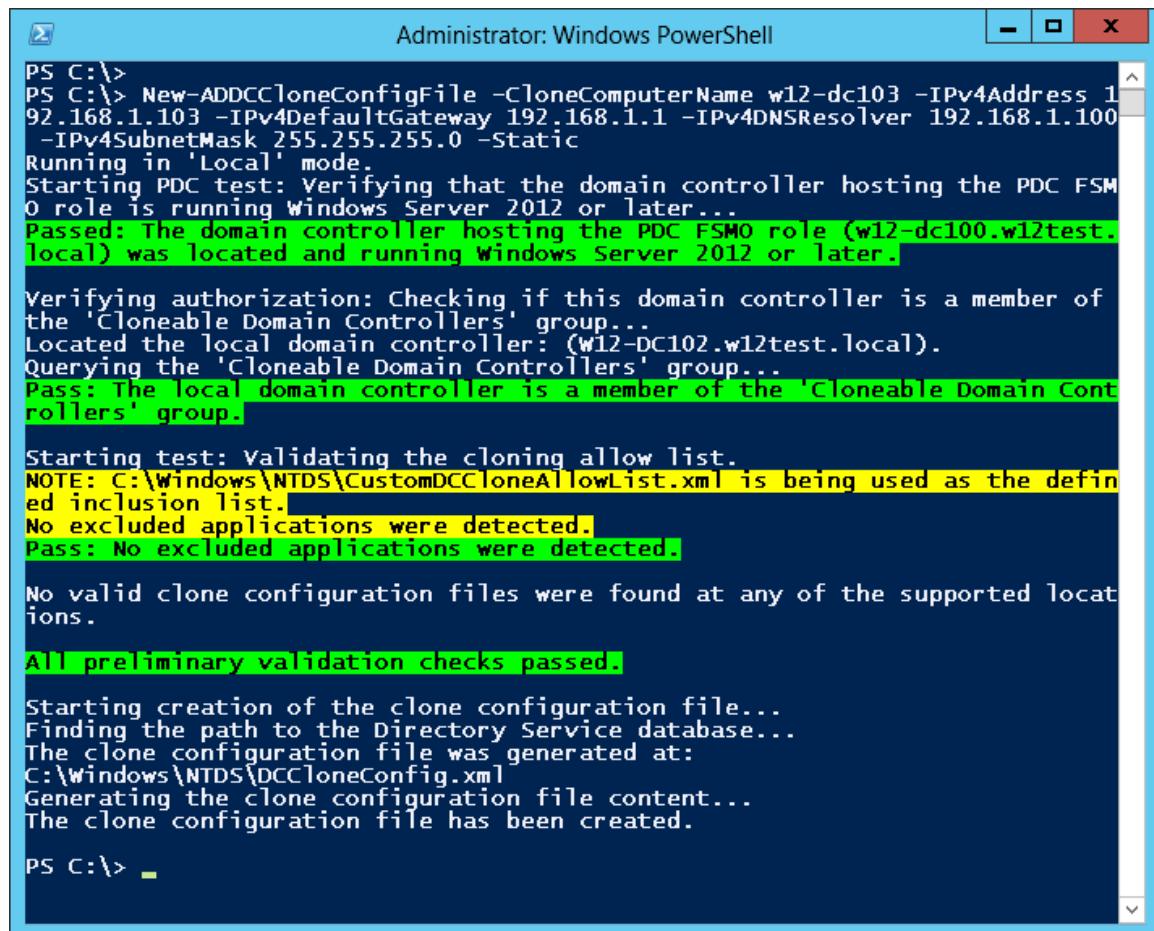
The domain controller configuration file (`DCCloneConfig.xml`) is an XML file that contains the identity information for the new domain controller. Before shutting down the source domain controller, the configuration file must be created.

To create the Domain Controller Configuration File, use the PowerShell command `New-ADDCCloneConfigFile`, which has various parameters. The most common parameters are as follows:

```
New
-ADDCCloneConfigFile
-CloneComputerName <New DC Name>
-IPv4Address <IP Address>
-IPv4DefaultGateway <Gateway>
-IPv4DNSResolver <DNS Server>
-IPv4SubnetMask <Subnet Mask>
-Static
```

The following screen shows the output of this command, including the pre-checks that occur during execution. The pre-checks include verifying that the PDCe runs on a Windows Server 2012 domain controller, checking whether or not the source domain controller is a member of the Cloneable Domain Controllers group, verifying the clone allow list, and checking for existing clone configuration files.
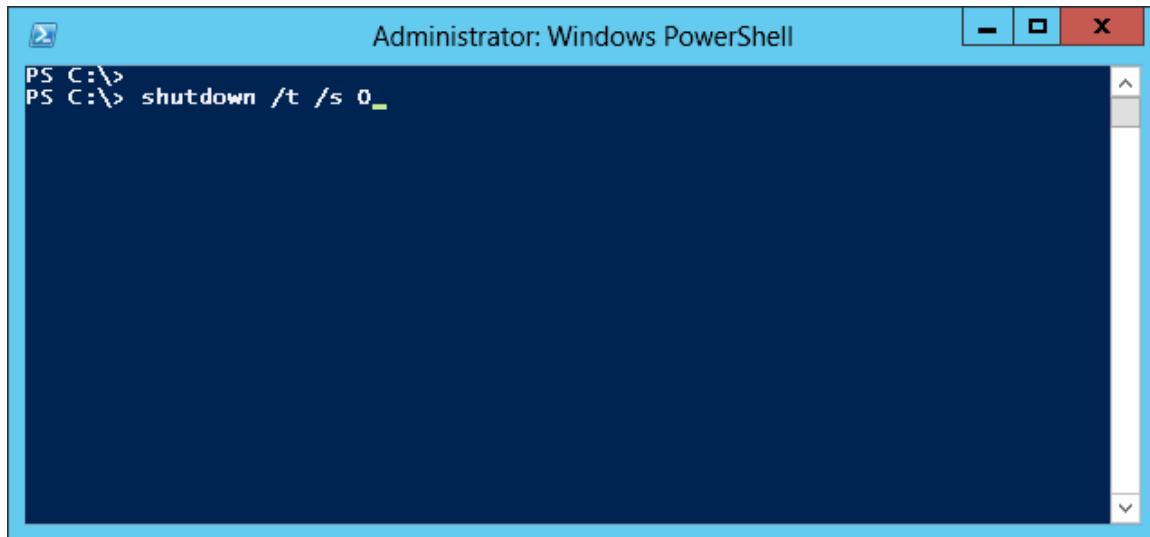
Figure 23. Generating DC Clone Configuration File

Shut Down the Source Domain Controller

The last step before you can clone a domain controller is to shut it down as shown in the following screen. Hot-cloning a domain controller for the purpose of creating a domain controller clone is not supported in a production environment.

Figure 24. Shutting Down the Domain Controller
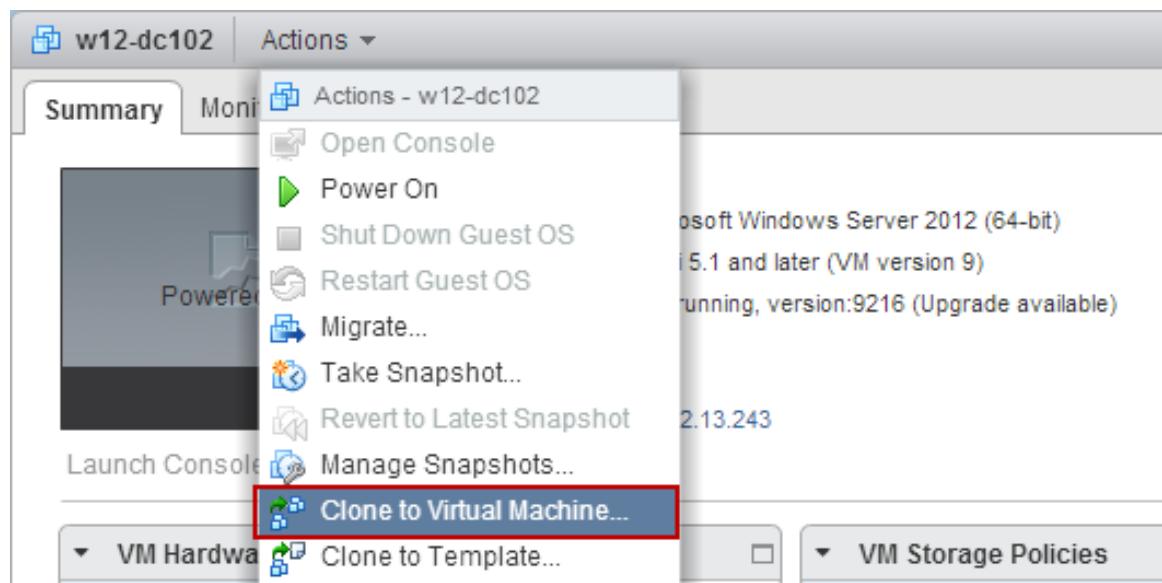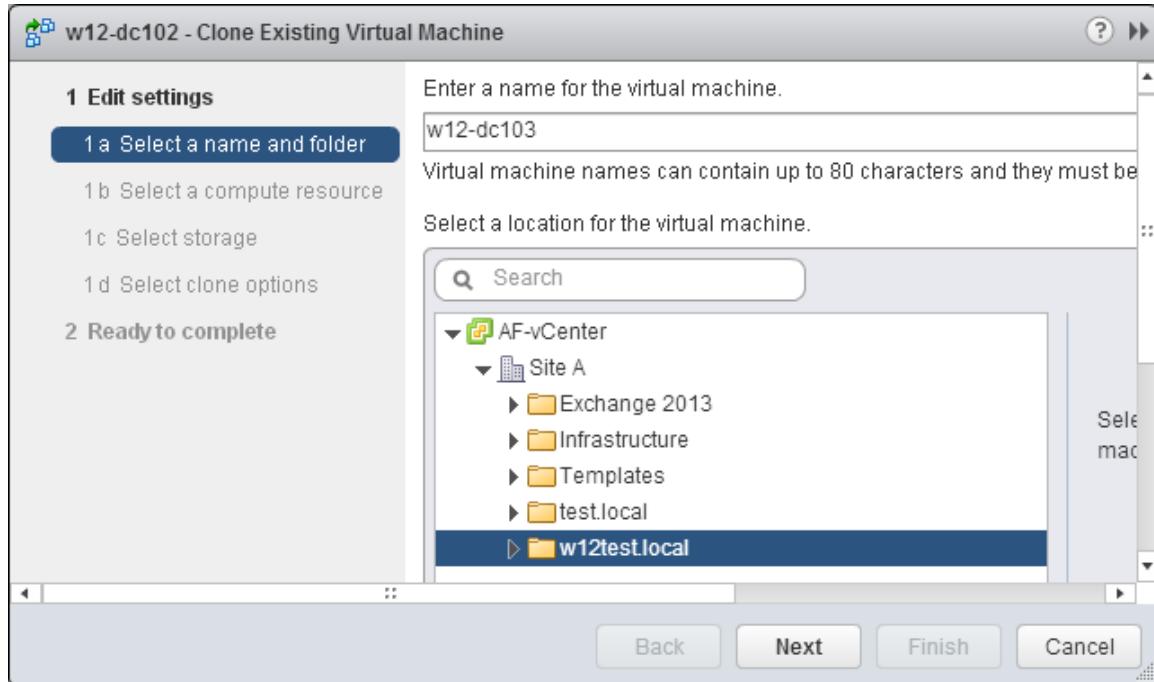
Clone the Source Domain Controller Virtual Machine

With the source domain controller virtual machine powered off, the virtual machine can be cloned through any VMware supported process, including standard cloning and copying of a virtual hard disk. In this example, the standard cloning process is used.
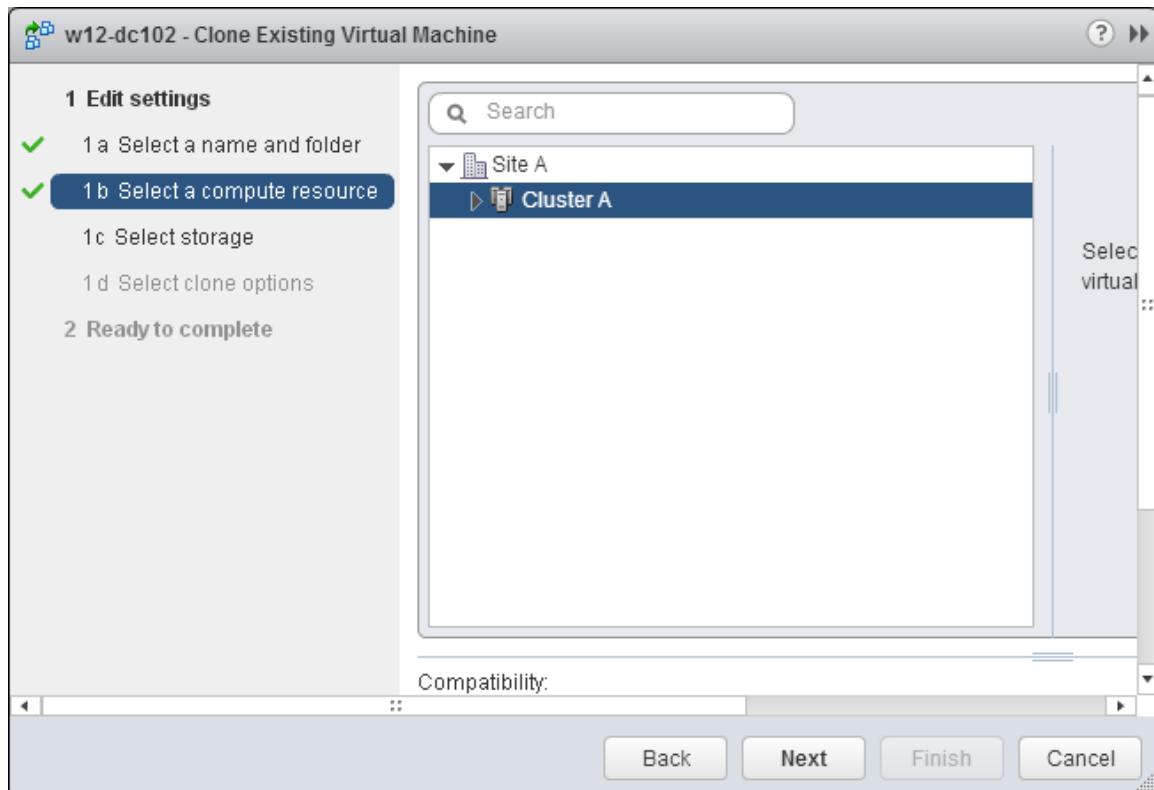
1. Within the vSphere Web Client, locate the source domain controller virtual machine and select **Clone to Virtual Machine** from the **Actions** menu.
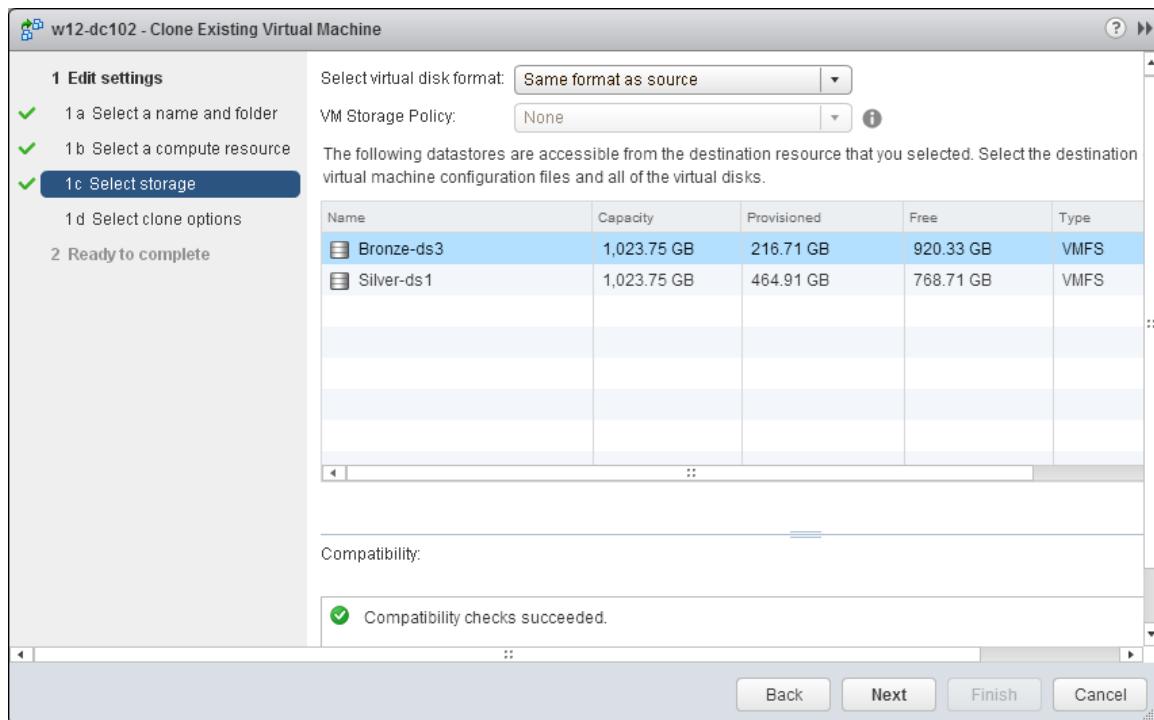
2. Using the **Clone Existing Virtual Machine** wizard, provide a name and location for the new virtual machine.
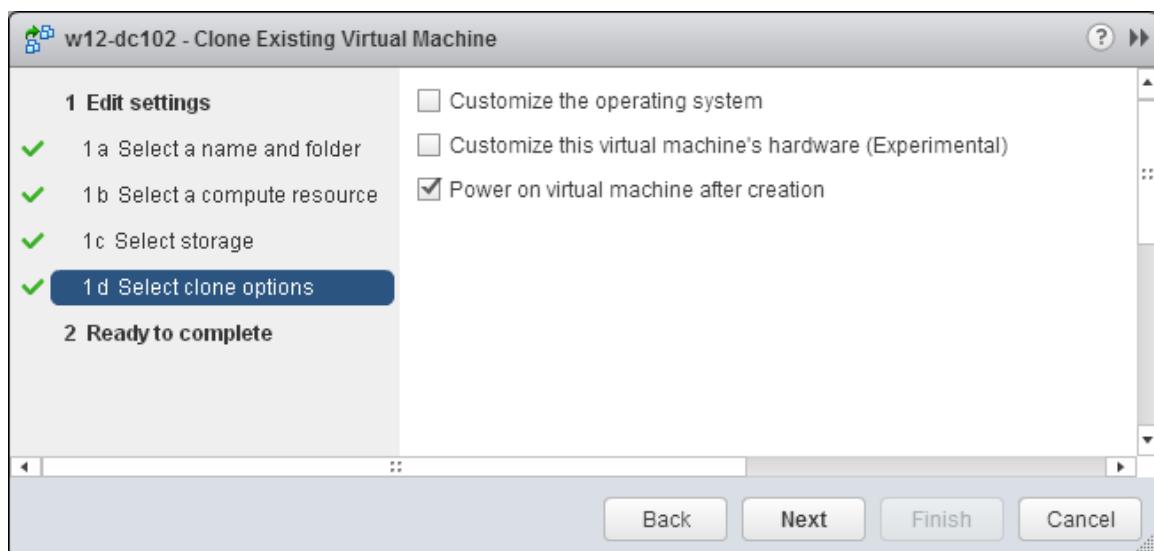
3. Select the compute resource for the new virtual machine.
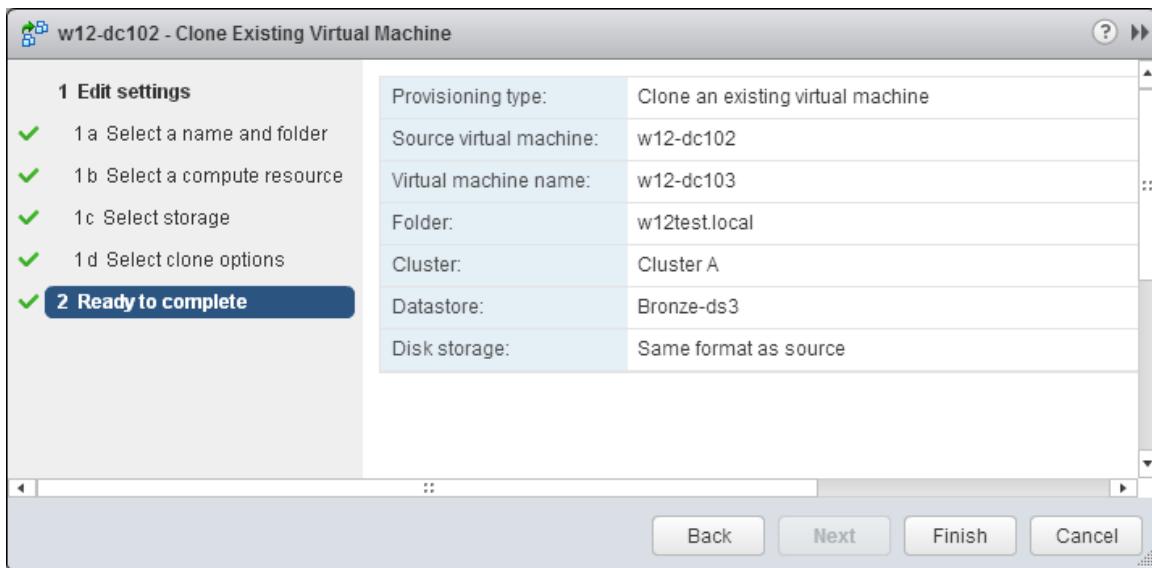
4. Select the storage format and location for the new virtual machine.
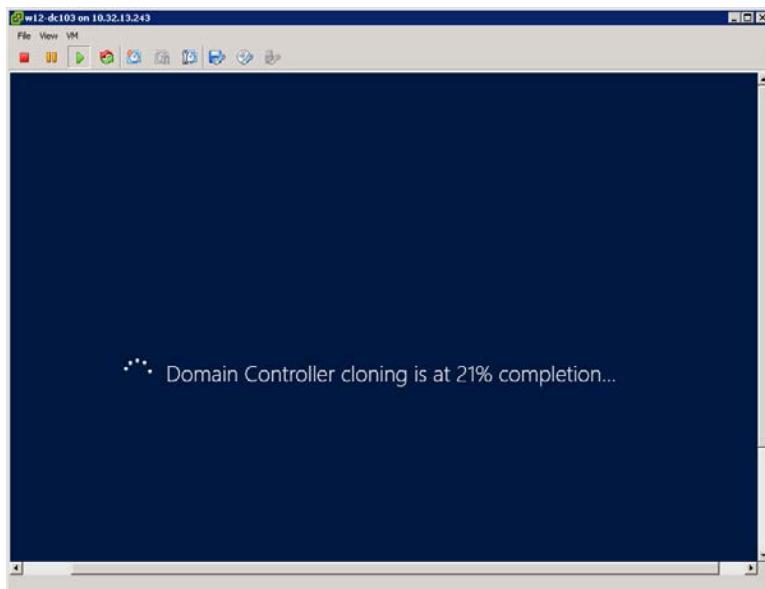
5. If the new domain controller is to reside on a new network segment, either leave all check boxes deselected and modify the network settings prior to powering on the virtual machine, or select the check box to edit the virtual hardware and adjust the network settings for the virtual NIC. The virtual machine must be able to communicate on the network during the first boot cycle to initiate the cloning process. In this example, the virtual machine is deployed on the same network segment as the source. Do not select the *Customize the operating system* option because that will interfere with the safe cloning operation.

6. Confirm the settings and proceed with the cloning.
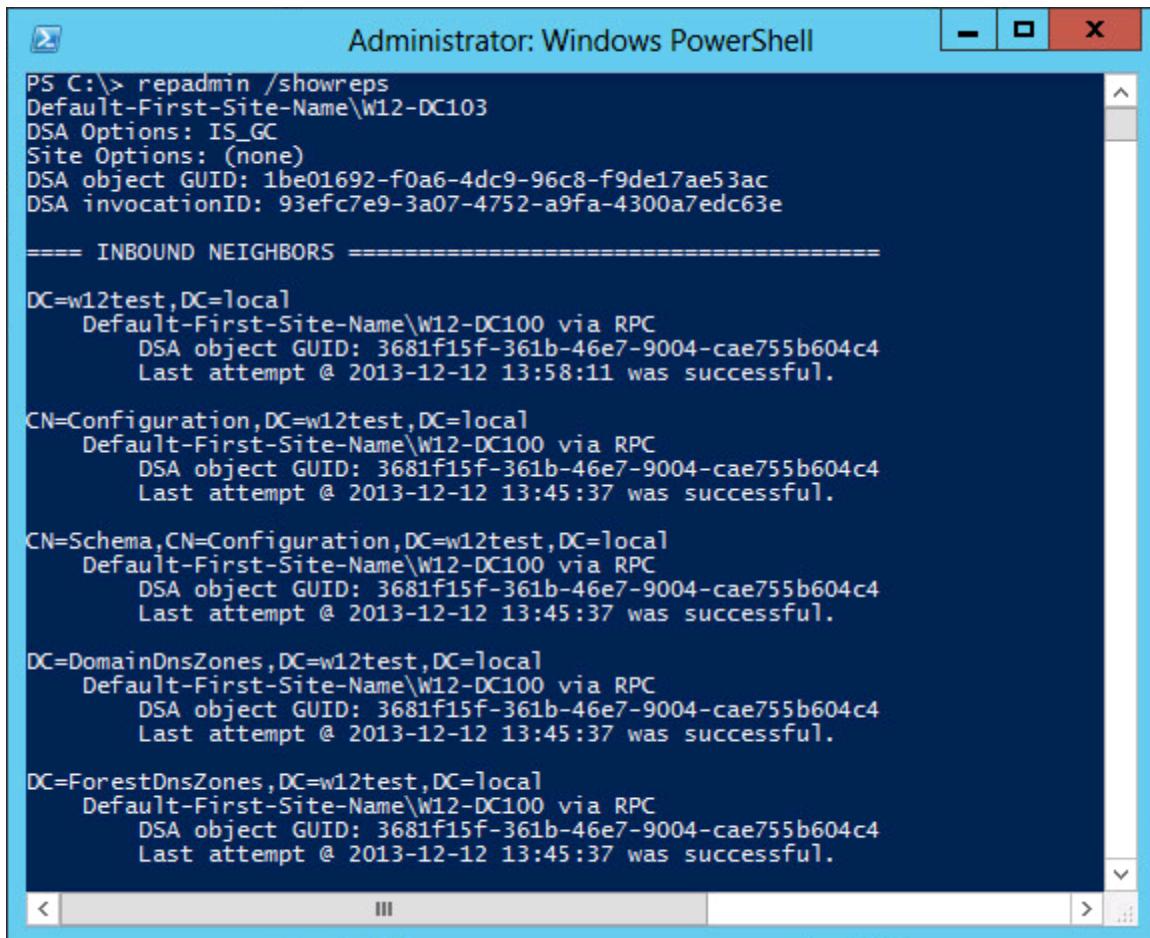
7. When the cloning process is completed and the virtual machine boots for the first time, the domain controller virtualization safeguards initiate the cloning process.

8. Verify the domain controller identity and replication status.

In the following figure, the domain controller name and invocation ID are unique, and replication is completing successfully with w12-dc100.

The directory service event log tracks notable events during the domain controller cloning process as described in the following table.

Table 2. Domain Controller Cloning Events

Event ID	Details
2160	The local Active Directory Domain Services has found a virtual domain controller cloning configuration file.
2170	A Generation ID change has been detected.
1109	The InvocationID attribute for this directory server has been changed.
2163	DsRoleSvc service was started to clone the local virtual domain controller.
29219	Virtual domain controller cloning succeeded.
2179	The msDS-GenerationId attribute of the Domain Controller's computer object has been set to the following parameters:

Appendix B: Testing Domain Controller Safeguard

The example described in this section takes a snapshot of a domain controller, creates a group of users, and rolls back the domain controller to its prior state (before users were created). The purpose of this exercise is to provide a process for testing and verifying the successful invocation of domain controller safeguard in Windows Server 2012 and vSphere.

For detailed information on the safeguard process, refer to *Virtualized Domain Controller Deployment and Configuration* at http://technet.microsoft.com/en-us/library/jj574223.aspx.

For this exercise, the following tasks have already been completed.

- vSphere 5.5 (ESXi and vCenter Server) installed and operational

- Windows 2012 Active Directory domain created (w12test.local)

- All operations master roles on a single domain controller (w12-dc100)

- A second domain controller available to test safeguard

Verify Replication is Functional

Before taking a snapshot of the test domain controller, verify that replication is functioning. Run the following command.

```
Repadmin /showreps
```

This command provides details about the local domain controller and provides replication information about Active Directory partitions. In the following screen, the *DSA invocationID* is the current version of the Active Directory database. The Active Directory database version and USNs are used to create transactions that can be replicated between domain controllers.

Figure 25. Verifying Replication

Create a Virtual Machine Snapshot

After verifying that replication has successfully completed, create a snapshot of the current domain controller. For testing purposes, create the snapshot to include the virtual machine's memory so that the virtual machine can revert while running. Quiescing the file system is optional.

Figure 26. Taking a Virtual Machine Snapshot
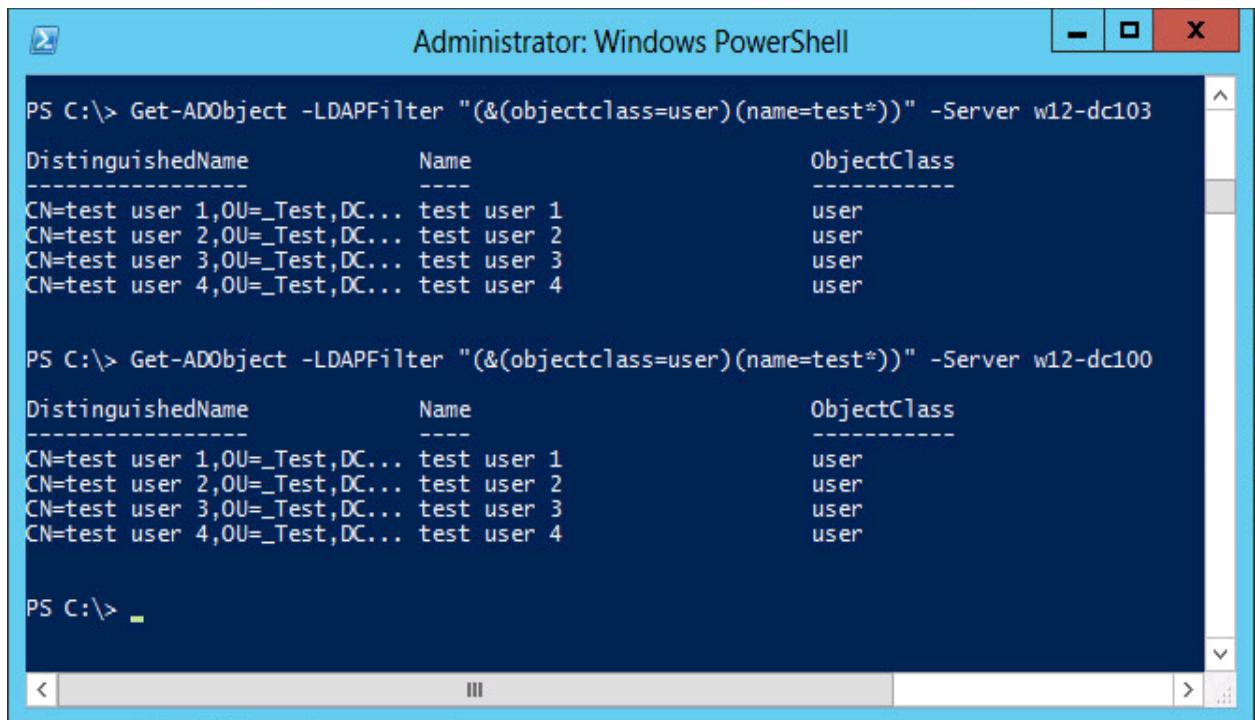
Create a Set of Test Users

Prior to domain controller safeguards, changes that occurred on a domain controller between the time of the snapshot and the time of the reversion are not replicated back to the originating domain controller. This result is due to neighboring domain controllers assuming that the originator is up to date.

To verify that changes lost due to a snapshot reversion are replicated back to the originator, create one or more users. In this example, four users are created. Each domain controller is queried to verify that it has knowledge of the new users.

Figure 27. Verifying that Users Exist on All Domain Controllers
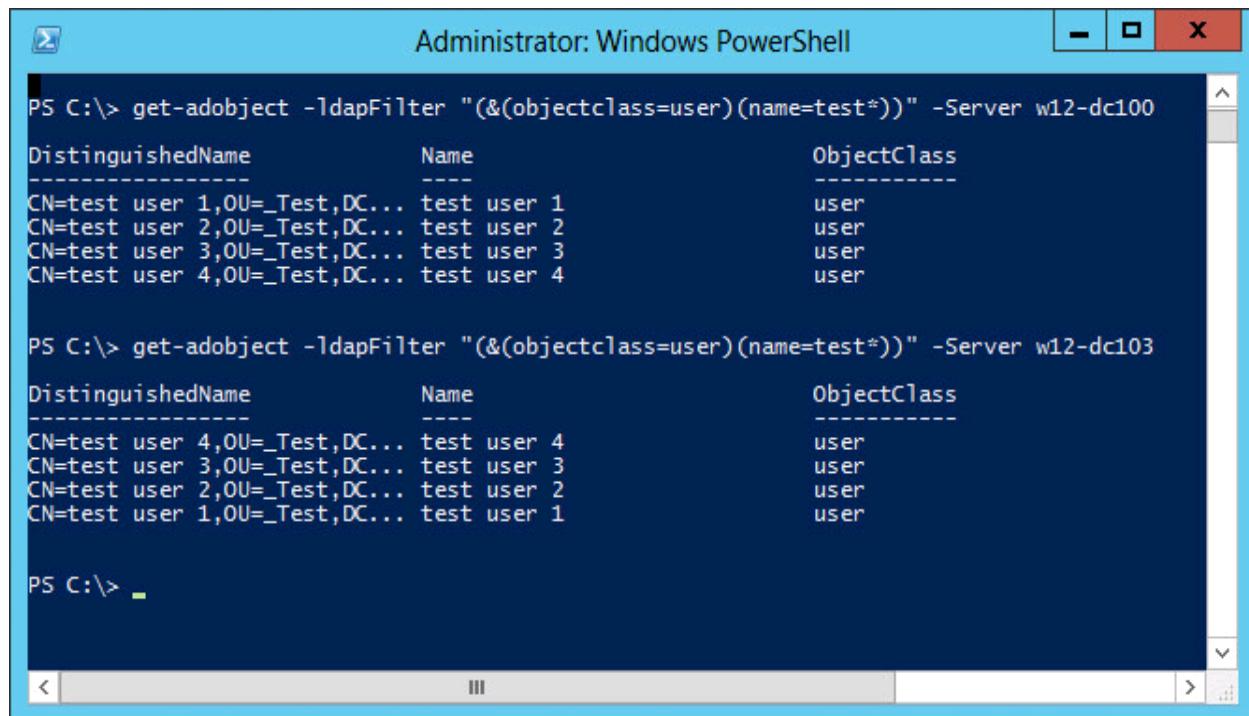
Revert the Snapshot

Revert to the previous snapshot. After the snapshot has been rolled back, validate that both domain controllers have knowledge of the newly-created users. In this example, the domain controller that has been reverted has replicated back the changes that were lost due to the rollback.

Figure 28. Verifying Users After Reversion
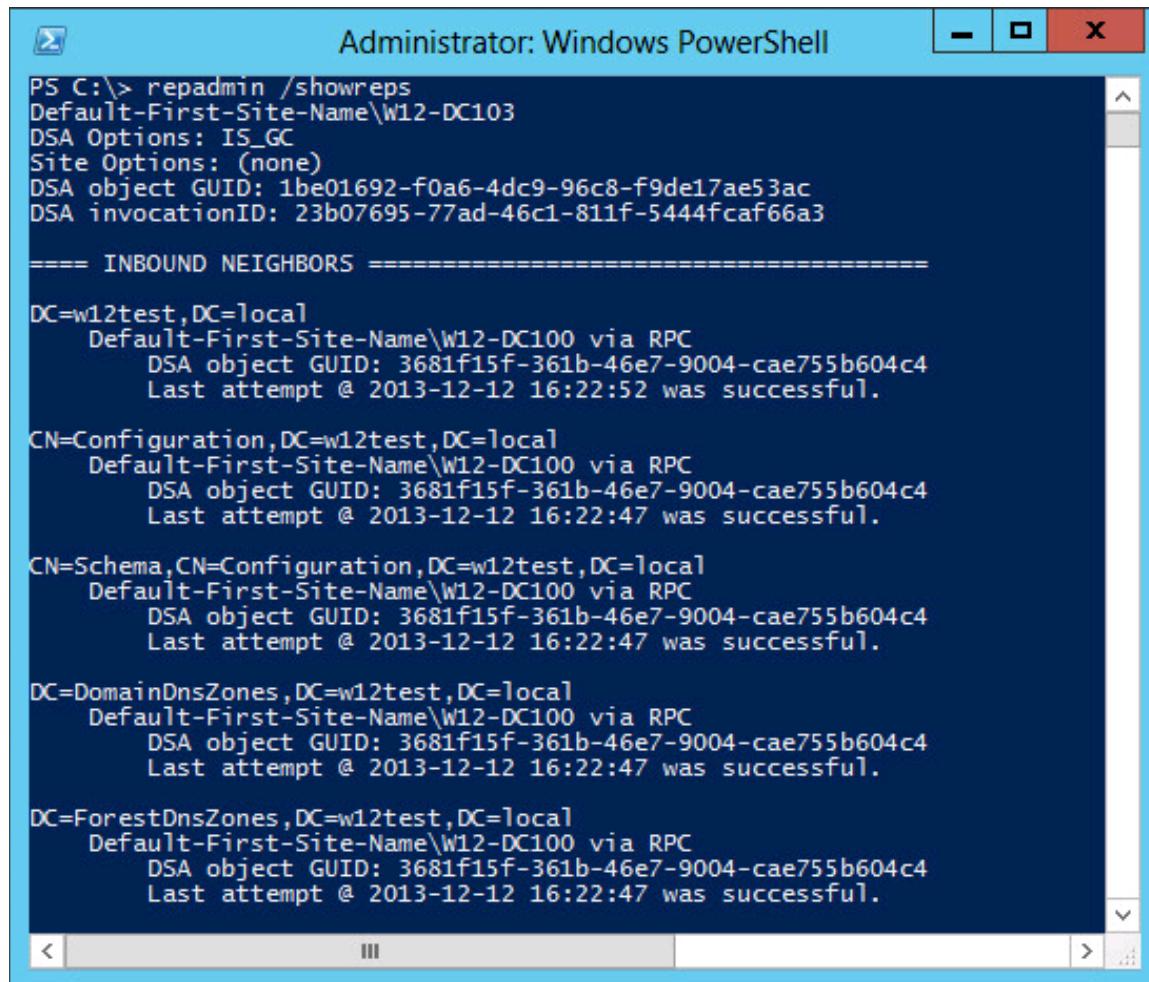
Validate Safeguard Features

To validate that the safeguard features were activated during the snapshot reversion, run the repadmin command, then check the InvocationID. This step allows you to verify that replication is functioning. The DSA InvocationID has been changed, informing other domain controllers that this is a new version of the Active Directory database and that new changes originating from this domain controller are valid.

Figure 29. Validating Replication and InvocationID

```
PS C:\> repadmin /showreps
Default-First-Site-Name\W12-DC103
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: 1be01692-f0a6-4dc9-96c8-f9de17ae53ac
DSA invocationID: 23b07695-77ad-46c1-811f-5444fcaf66a3

==== INBOUND NEIGHBORS ======================================

DC=w12test,DC=local
    Default-First-Site-Name\W12-DC100 via RPC
        DSA object GUID: 3681f15f-361b-46e7-9004-cae755b604c4
        Last attempt @ 2013-12-12 16:22:52 was successful.

CN=Configuration,DC=w12test,DC=local
    Default-First-Site-Name\W12-DC100 via RPC
        DSA object GUID: 3681f15f-361b-46e7-9004-cae755b604c4
        Last attempt @ 2013-12-12 16:22:47 was successful.

CN=Schema,CN=Configuration,DC=w12test,DC=local
    Default-First-Site-Name\W12-DC100 via RPC
        DSA object GUID: 3681f15f-361b-46e7-9004-cae755b604c4
        Last attempt @ 2013-12-12 16:22:47 was successful.

DC=DomainDnsZones,DC=w12test,DC=local
    Default-First-Site-Name\W12-DC100 via RPC
        DSA object GUID: 3681f15f-361b-46e7-9004-cae755b604c4
        Last attempt @ 2013-12-12 16:22:47 was successful.

DC=ForestDnsZones,DC=w12test,DC=local
    Default-First-Site-Name\W12-DC100 via RPC
        DSA object GUID: 3681f15f-361b-46e7-9004-cae755b604c4
        Last attempt @ 2013-12-12 16:22:47 was successful.
```

Additionally, the directory service event log tracks the events during domain controller safeguard as described in the following table.

Table 3. Domain Controller Safeguard Events

Event ID	Details
2170	A Generation ID change has been detected.
2181	The transaction was aborted due to the virtual machine reverting to a previous state. This result occurs after the application of a virtual machine snapshot, after a virtual machine import operation, or after a live migration operation.
2204	Active Directory Domain Services has detected a change of VM-Generation ID. The change means that the virtual domain controller has been reverted to a previous state. Active Directory Domain Services will perform the following operations to protect the reverted domain controller against data divergence and to protect creation of security principals with duplicate SIDs: create a new invocation ID and invalidate current RID pool.
1109	The invocation ID attribute for this directory server has been changed.