Zerto

a Hewlett Packard Enterprise company

Protecting Microsoft SQL Server with Zerto

Best Practices Guide

Version 1.0

Zerto
a Hewlett Packard Enterprise company

Table of Contents

Introduction

Overview

Microsoft SQL Server is one of the most common database servers utilized today. As the size and number of Microsoft SQL server instances in an environment increases, so too does the complexity and requirements when it comes to disaster recovery and replication.

This guide is intended to cover everything related to protecting Microsoft SQL Servers and their various configurations with Zerto. It is presumed the reader has a good understanding of VMware vCenter, Microsoft Windows, Microsoft SQL Server, and Zerto.

VM Configuration

Virtual Disk Configuration

As per Microsoft SQL Best Practices it is recommended to ensure the Windows VM has sufficient disks for segregating the different types of data inside a Microsoft SQL VM.

These are typically:

1. Operating System Disk
2. Paging File Disk
3. SQL Server Database Disk
4. SQL Server Logging Disk
5. SQL Server TempDB Disk

This is the recommended minimum number of disks for a Microsoft SQL VM.

It is recommended that the Windows Paging file be placed on a separate disk to ensure bandwidth and journal space are not wasted replicating changes to this file by utilizing the feature covered in the Zerto Temp Data Configuration section (on next page).

VM Size and Resources

Zerto can replicate any VM hardware version, specification, or size supported by the target platform. It is not possible to configure automatic reconfiguration of VM resources such as CPU and RAM as part of a failover to platforms other than the public cloud. However, it is possible to select a target Resource Pool for the VM to limit its resource usage in a move or failover scenario. This method allows dynamic management of resource allocation after a move or failover event as opposed to a static reconfiguration of VM specification and resources.

Zerto Temp Data Configuration

Zerto has the unique ability to replicate a VMDK/RDM once but then not replicate any subsequent changes; this feature is known as Temp Data Disk. This means that if TempDB and a Windows Paging file, for example, are placed on separate VMDKs/RDMs and set as Temp Data in Zerto, an initial copy of the disk will be replicated to the recovery site, but subsequent changes will not be replicated. This is configured within the individual disk settings on a protected VM as per figure 1 on the following page.

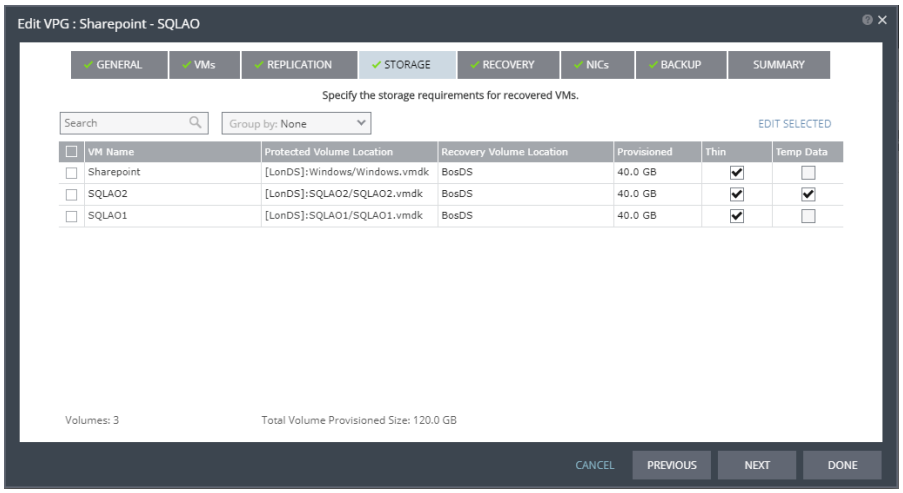

Figure 1

When Windows boots after a failover or failover test, Zerto will present the original replica of the page file and TempDB disk. Because Windows clears its page file and SQL clears its TempDB on every reboot this will have no impact on the ability to recover.

This feature is unique and provides the following significant benefits when protecting Microsoft SQL Servers:

- Can result in up to 50% reduction in replication traffic (results can vary depending on the type of Microsoft SQL Workloads).
- Reduces replication traffic even if TempDBs are relatively small in relation to the production databases.
- Reduces journal usage in the recovery site as changes to TempDB or the Paging File are not stored or needed.
- Write I/O to the Temp Data disks are not copied to the VRA, reducing load significantly in high I/O VMs.
- Removes complexity in having to store VMDKs in a separate datastore so as not to replicate them with legacy storage-based replication technologies.
- Removes the need to exclude the disk from replication as with VM-level replication technologies.
- Journal promotion on recovery can be performed quicker due to there being no data to promote on these disks.

The Temp Data disk option can be enabled on any disks in a protected VM to allow local, subsequently created SQL Backups and log-shipping shares to be kept on the protected VM without wasting bandwidth replicating this duplicate data. It is recommended to keep the protected VM as lean as possible to ensure replication traffic is kept to a minimum along with reduced journal size and storage usage. Examples are not creating large files on non-Temp Data disks such as database backups, ISOs, etc.

Microsoft SQL Server Configuration

Backups

It is not recommended to perform a SQL backup of databases to a local disk on a Zerto-protected SQL VM. Creating a local SQL .BAK file can significantly increase the RPO of the VPG until the entire .BAK file has been replicated to the recovery site. It will also significantly increase journal space usage as the entire database .BAK file will be held in the journal for the period specified on the VPG SLAs. If local SQL .BAK files are required, then it is recommended for these to be created on a separate VMDK indicated as a Temp Data disk in Zerto before the SQL .BAK files are created to ensure the data is not replicated by Zerto. Any other backups, such as snapshot-based backups of the protected VMs or agent-based backups to a remote server, have no impact on Zerto.

Log Shipping

Log shipping can be enabled on Microsoft SQL VMs protected by Zerto. It is, however, important to note that with the journaling capability within Zerto, which allows failover to points in time every few seconds while maintaining write-order fidelity, this can often remove the use case of running SQL log shipping in many environments.

Microsoft SQL Server Consistency

Crash Consistency

Through Virtual Protection Groups (VPGs) Zerto can provide consistency between VMs during replication; however, all the checkpoints created are crash consistent, essentially the same as if the power is pulled on a VM. This is deliberate because obtaining application-consistent checkpoints, which we cover in the next section, requires everything in memory to be flushed to disk, thus stunning the application. This is not something you would want to do every 5 seconds and so your RPO will therefore have to increase; realistically you probably wouldn't want to do this more than 2-4 times a day = RPO of 6 hours. When it comes to a recovery scenario data loss is everything and that is why customers will always recover to the latest point in time possible; with Zerto this is typically seconds before. Zerto also ensures that write order fidelity is maintained on all replicated data which, when tied with the fact that MSSQL itself is ACID-compliant, meaning SQL will guarantee the Atomicity, Consistency, Isolation & Durability properties to ensure database reliability even in the event of errors, power failures etc., then having application consistency with large data loss scenarios in any disaster is no longer a requirement.

Application Consistency

Zerto replicates writes occurring to the protected SQL VM disks; it does not replicate the memory of the protected VM. To ensure any transactions held in the memory of the VM are replicated by Zerto, as required for application consistency, they need to be committed to disk, typically achieved using the Microsoft VSS SQL Writer service. It is not necessary to configure application consistency in Zerto to maintain a consistent database in recovery, it is only required for transactions held in memory and for a failsafe point in time for recovery. Application-consistent points in time will be indicated as a checkpoint in the Zerto journal of changes to ensure visibility when performing a move, failover, or failover test in Zerto. The checkpoints can be created manually, scripted as part of an existing database quiesce operation, or automatically by the Zerto VSS agent.

Implementation of the VSS components is covered in a separate VSS deployment guide on MyZerto.

NOTE: Use of the VSS components requires a separate license and installation media.

Microsoft SQL Server High Availability

Microsoft SQL Server Failover Cluster

When protecting a MSSQL Failover Cluster with Zerto the key consideration is consistency of the database RDMs and cluster itself.

NOTE: This section does not apply to Always-On Availability Group clusters.

Only the Primary Active Node in an Active/Passive Cluster should be protected by Zerto. Protecting both nodes with Zerto is not recommended for the following reasons:

1. This will require double the number of RDMs in the target site as there is no ability to replicate 2 VMs to the same target RDM.
2. If the active node is not changed often then a large amount of data will require replication after switching to the passive node.
3. A failover operation will be complex as the nodes will be using separate disks in the target site and this will break the cluster.

If the Active Node role is switched to a non-Zerto-protected node then any changes made to the RDMs are not replicated. Once the Active Node role is moved back to the Zerto-protected node the cluster will be in an inconsistent state, as the target RDMs contain data Zerto did not replicate. Performing a Force-Sync operation on the cluster VPG will return the cluster to a protected and consistent state.

This operation scans both the source and target RDMs then replicates any changes and inconsistencies found. The Force-Sync operation can be initiated manually to maintain cluster consistency during maintenance etc. For example, during cluster

maintenance, when the administrator changes the Active Node role back to the Zerto protected node, their final action should be to select Force-Sync in the Zerto GUI. Note: Performing a Force-Sync operation will attempt to preserve the journal in some capacity, however it may reset the journal on the VPG, removing the ability to recover to previous point in time before the operation was performed. It is therefore recommended to only perform this operation out of working hours if the journal still contains consistent checkpoints that are required.

Automating Active Node Change by Script

An alternative to manually performing Force-Sync operations for maintaining consistency in shared disk clusters is to automate this operation using the Zerto PowerShell SQL Cluster script found in the Zerto Tech Marketing GitHub https://github.com/Zerto-Tech-Marketing/MSSQL_MSCS_Failover.

Note: No script is required for MSSQL Clustering support; it can simply be used to automate the manual process for maintaining consistency after cluster failover/failback operations.

The first step to utilizing this script is to create two VPGs: one protecting the current active node only and a second protecting the current passive node only.

The script should then be scheduled to be run directly on both SQL nodes every 1 minute. Its purpose is to check the active SQL node is the node protected by Zerto and automatically unpause the relevant VPG and perform a Force-Sync if this is ever changed. The script will also pause the formerly active SQL nodes VPG to clearly indicate that the passive SQL node is not being replicated. Further information is provided in the comments section at the beginning of the script.

Post-Failover Configuration

Only the active node VPG should be recovered in a failover scenario; the passive node should then be rebuilt or moved, assuming production is still available, after recovery. If you choose to move the passive node the following process should be used:

1. Verify that the active node is fully functional at the new site.
2. Modify the passive node at the original site to remove the shared disks and force it to boot to BIOS.
3. Move the passive node with Zerto, using the existing VPG, to the new site and verify networking to Domain and the Active node.
4. Shut down the passive node and add the shared disks or RDMs to it.
5. Power on the passive VM and verify that cluster services can failover.

Zerto can automatically change the IP address of VMs as part of a failover or failover test operation. However, if a MSSQL cluster requires a new IP address on the target site, this feature should not be used. This is due to issues with clusters and IP changes that can require manual intervention as part of a failover operation that significantly increase the RTO and complexity.

It is therefore recommended for an MSSQL cluster to have listeners pre-configured on both the source and target IP ranges, each node with a dedicated heartbeat NIC, so that only a simple DNS update to the new listener IP is required as part of a failover operation. If IP changes are required, this can easily be tested as part of a failover test operation as covered later in this guide.

Limitations

Zerto MSSQL Failover Cluster support is not compatible with the following:

1. Active / Active cluster – All SQL instances must run on the same node.
2. Protecting both nodes in a shared disk cluster – Only the active node can be protected.
3. Replicating 2 VMs to the same target RDMs.
4. Protecting Cluster VMs using multi-writer VMDKs as shared cluster disks.
5. Protecting Cluster VMs using iSCSI in-guest initiators to access shared cluster disks.

6. When a ZCC is in use. RDM to RDM replication cannot be used because Zerto does not support RDMs as a configurable ZORG resource in neither VCD nor VC back-end cloud sites.
7. Replicating on-premise to a cloud through a ZCC. Only RDM replication to VMDK can be used; however, RDM replication to VMDK is not advised as recovery will not be seamless.
    - For more information on this please see the following Zerto KB article: Zerto KB

Always-On Availability Groups

When protecting Always-On Availability Group-based clusters, only the IP address is shared between the cluster nodes. It is therefore possible to protect one or both nodes with Zerto simultaneously.

Zerto recommends that all nodes are protected together within a single VPG to ensure the entire availability group can be recovered in sync.

Post Failover Configuration

The biggest consideration here is once again the re-IP aspect, should it be required. As mentioned in the MSSQL Failover Cluster section, Zerto can automatically change the IP address of VMs as part of a failover or failover test operation. That being said, the Always-On Availability Group itself will need some changes too. While this could be scripted using a post-recovery failover script, it is recommended to perform any re-IP of MSSQL Always On nodes manually post-failover to reduce complexity.

It is also recommended for an MSSQL Always-On Availability Group to have listeners pre-configured on both the source and target IP ranges too, so that again, only a simple DNS update to the new listener IP is required as part of a failover operation. If IP changes are required, this can easily be tested as part of a failover test operation as covered later in this guide.

Failover Testing

To successfully perform a non-disruptive failover test of an Microsoft SQL virtual machine configured in one of the above high-availability configurations, Active Directory and DNS services are required to be online in the failover test isolated network. Therefore, Zerto recommends protecting an Active Directory Domain Controller, configured as a global catalog and the primary or secondary DNS server for the SQL Server virtual machine. Zerto is used to bring an up-to-date copy of Active Directory online with ease for failover testing.

The Active Directory virtual machine should never be recovered to previous points in time in a production/live failover. Therefore, Zerto recommends placing the Active Directory virtual machine in its own VPG and assigning both failover and failover test Network Adapters in the virtual machine to connect to an isolated test network. Zerto recommends adhering Microsoft best practices for Active Directory for production/live failovers.

Note: When booting Active Directory in an isolated test network, a minimum five-minute window is required for Active Directory services to come fully online to allow the cluster services to start.

Useful References

- https://www.zerto.com/myzerto/technical-documentation/
- https://www.zerto.com/myzerto/knowledge-base/
- https://www.zerto.com/myzerto/forums/