APPSYNC INTEGRATION WITH MICROSOFT SQL SERVER

ABSTRACT
This white paper discusses how DELL EMC® AppSync integrates with Microsoft® SQL Server to provide a solution for continuous availability of critical user data enterprise-grade protection solutions for critical SQL Server databases with a negligible performance impact to SQL workloads, enabling faster recovery of databases.

January, 2018
The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2018 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA. 01/18 White Paper H12583.1

Dell EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.
TABLE OF CONTENTS

EXECUTIVE SUMMARY .................................................................................................................. 5
  AUDIENCE.................................................................................................................................... 5

CREATING SQL SERVER DATABASE COPIES .................................................................................. 5
  Prerequisites and Restrictions ....................................................................................................... 5
  SQL Server backup types .............................................................................................................. 5
  Efficiency ..................................................................................................................................... 5
  Salient features ........................................................................................................................... 6
  Truncating the transaction log ..................................................................................................... 6

SQL SERVER TRANSACTION LOG BACKUP .................................................................................. 7
  Restrictions .................................................................................................................................. 7
  Log backup expiration .................................................................................................................. 8

RESTORING SQL SERVER DATABASE COPIES ............................................................................. 8
  Restoring from the copy ................................................................................................................ 8
  Affected entities during restore .................................................................................................... 9
  Backup of transaction logs .......................................................................................................... 10
  Restoring damaged databases .................................................................................................... 11
  Restore restrictions ..................................................................................................................... 11

MOUNTING SQL SERVER DATABASE COPIES WITH RECOVERY ............................................. 11
  Mount and production host version compatibility ....................................................................... 11
  Mount SQL Server database copy as a clustered resource .......................................................... 12
  Special considerations for mount to production cluster .............................................................. 12
  Supported mount recovery types ............................................................................................... 12
  Mounting copies using Path Mapping ......................................................................................... 13
  Mounting copies from multiple production hosts to a single mount host .................................. 13
  Mounting multiple database copies existing on same production LUNs or consistency groups ... 14
  Partial recovery of mounted databases ....................................................................................... 14

UNMOUNTING SQL SERVER DATABASE COPIES ..................................................................... 14
  Custom shutdown script prior to unmounts .................................................................................. 14

SQL SERVER ALWAYSON INTEGRATION WITH APPSYNC .......................................................... 15
  AlwaysOn Failover Cluster Instances ......................................................................................... 15
  Protecting databases in clustered instances .............................................................................. 15
  AlwaysOn Availability Groups ................................................................................................... 16
EXECUTIVE SUMMARY

DELL EMC AppSync provides an efficient and simple backup and recovery solution for Microsoft SQL Server 2008, 2008 R2, 2012, 2014, 2016 and 2017 versions. It facilitates and automates the creation of disk-based copies of Microsoft SQL Server databases on various supported DELL EMC storage which can be used for protection and repurposing.

AUDIENCE

This white paper has been written for the following audience:

- Microsoft SQL Server administrators responsible for implementing AppSync in a SQL Server environment
- DELL EMC internal and field personnel who assist customers with implementing AppSync in Microsoft SQL Server environments

CREATING SQL SERVER DATABASE COPIES

PREREQUISITES AND RESTRICTIONS

1. The SQL Server database must be online during replication. If one or more databases subscribed to a service plan are offline during the service plan run, protection does not fail and copies are created for ONLY the databases that are online. AppSync dynamically checks for the database status during the service plan run and protects all the databases that are online at the time of the run. The protection will fail if all the databases subscribed to the service plan are offline during service plan run. This also applies for successful completion of SQL repurposing workflows.

2. System databases cannot be protected and are not listed post discovery. The system databases must not reside on the same storage volume as user databases. They might inadvertently get overwritten if a user database copy is restored.

3. SQL Server database and its transaction logs must be located on disks in the same storage array. AppSync can truncate transaction logs using log backups with log truncation enabled in SQL Server service plans. AppSync must have Full SQL Server backup type selected, under the Create local copy menu, in order to truncate transaction logs.

4. Full-text catalogs associated with a file group are included as part of the copy. If the full-text catalogs are not located on supported storage, protection will fail. When using full-text catalogs, make sure that the storage device where the catalog is located does includes data related to the database only.

5. If attempting to create a copy of a database mirror, the copy fails with an error indicating that the database is currently not in a valid state.

6. In Hyper-V environments, AppSync requires the storage for SQL databases and log files to be on iSCSI direct attached devices, Virtual Fiber Channel (NPIV), or SCSI pass-through devices. SCSI Command Descriptor Block (CDB) filtering must be turned off in the parent partition for SCSI pass-through. It is turned on by default. This is also applicable for SQL cluster servers.

SQL SERVER BACKUP TYPES

The following backup types are supported: Full, Copy, Non-VDI and Crash Consistent.

1. **Full**: Protects the database and the active part of the transaction log. This copy type is typically used when the copy will be considered a backup of the database or when the copy will be mounted in order to use a third-party product to create a backup of the database. This type of copy allows restoring of transaction logs to bring the database forward to a point in time that is newer than the copy, assuming those transaction logs have been backed up. AppSync uses Microsoft SQL Server’s VDI snapshot feature to create this type of copy.

   NOTE: SINCE “FULL” BACKUP TYPE CHANGES THE TRANSACTION LOG SEQUENCE, IT IS NECESSARY THAT USERS HAVE ONLY ONE SERVICE PLAN CREATING FULL COPIES OF A SQL DATABASE. IT IS ALSO NOTEWORTHY THAT NO OTHER 3RD PARTY BACKUP APPLICATIONS SHOULD BE CREATING “FULL” COPIES OF THE SAME SQL DATABASE.

2. **Copy**: Protects the database and the active part of the transaction log without affecting the sequence of backups. This provides DBAs with a way to create a copy without interfering with third-party backup applications that may be creating full and/or differential backups of the SQL Server databases. Similar to Full, AppSync uses Microsoft SQL Server’s VDI snapshot feature to create this type of copy.

3. **Non-VDI**: Protects the database without using Microsoft SQL Server’s VDI snapshot feature. This creates crash consistent copies of SQL using the Microsoft VSS freeze/thaw framework against the file system.

4. **Crash Consistent**: Protects the database without any agent involvement. Neither Microsoft VSS Framework nor Microsoft VDI snapshot features are used. This backup type creates true crash consistent copies of SQL databases using array level features only.

EFFICIENCY
Subscribing multiple SQL Server databases across instances to the same SQL Server service plan, to create copies of multiple databases in parallel, is supported. There is a multithreaded approach in the AppSync host plugin that enables application consistent backups of multiple databases in parallel. This provides a performance boost. AppSync divides the subscribed databases into efficiency groups. Each group will not have more than 35 databases per instance on a host, by default. This grouping avoids flooding the host plugin with requests and ensures load balancing during the copy creation process.

If there are two instances on the same host, then a group can have a maximum of 70 databases altogether, subject to a maximum of 35 databases per instance. While the first group is being protected, the other groups that belong to the same host wait until the protection of the former group is completed. The number of databases that can be protected together, per instance, is restricted by setting the “Maximum number of databases to be protected together” option in the “Advanced Plan Settings” tab of the SQL Server service plan.

**SALIENT FEATURES**

1. **Dynamic discovery of databases**: AppSync can dynamically add or remove databases from a protection plan as and when they are added or removed from the instance on the SQL Server. This can be achieved by subscribing the User Databases folder of a particular SQL Server instance to a service plan.

2. **Dynamic discovery of database components**: AppSync dynamically discovers database components and creates copies accordingly. For example, a user may add or delete some file groups to a SQL Server database on the production environment. AppSync will dynamically accommodate such changes and will create a copy that is up-to-date with the production database.

3. **Operations for deleted databases**: A database that has been removed from a production SQL Server instance but has a copy in AppSync, is marked with a “database could not be found” flag in the AppSync console. Choose to mount or restore such database copies is supported. These databases, however, are no longer considered during subsequent protection cycles. The “database could not be found” flag does not apply to database components like log files and file groups. If intending to add or remove individual database components, select the copy for restore carefully. Optionally, mount a copy to check whether all the desired data is available in that backup before performing a restore.

4. **Pre-copy script**: To perform preparatory steps before creating a copy, specify a pre-copy script and parameters in the service plan’s Settings tab. The pre-copy script runs as per the scheduled set in the Plan Startup phase. Valid script formats are .bat, .exe, and .ps1 (PowerShell scripts). Optionally enter credentials to run the script as a specific user. The script runs as Local System by default. The default location of the script is \%ProgramData\%EMC\AppSync\scripts\ on the application host. Exact parameters depend upon the specific script. Parameters with spaces must be enclosed in double quotes. This phase can be enabled or disabled (default). This operation requires the Service Plan Administrator role in AppSync.

5. **Post-copy script**: To perform cleanup or other post-copy steps after creating a copy, specify a post-copy script and parameters in the service plan’s Settings tab. The script runs on successful completion of the Create copy phase. Valid script formats are .bat, .exe, and .ps1 (PowerShell scripts). Optionally enter credentials to run the script as a specific user. The script runs as Local System by default. The default location of the script is \%ProgramData\%EMC\AppSync\scripts\ on the application host. Exact parameters depend upon the specific script. Parameters with spaces must be enclosed in double quotes. This phase can be enabled or disabled (default). This operation requires the Service Plan Administrator role in AppSync.

**TRUNCATING THE TRANSACTION LOG**

In addition to scheduling online backups, create a maintenance plan for the transaction logs so they are truncated on a regular basis. If log records are never deleted from the transaction log, the logical log would grow until it fills all the available space on the disks holding the physical log files. At some point, old log records that are no longer necessary for recovering or restoring a database must be deleted to make way for new log records. The process of deleting these log records to reduce the size of the logical log is referred to as truncating the log.

Create a maintenance plan using the AppSync Transaction Log backup feature to occur at regular intervals that keep the transaction log from growing past the desired size. This means that the sequence of log backups must contain every log record that was written since the database backup. When maintaining a sequence of transaction log backups, no log record can be truncated until after it has been written to a log backup. Active logs (.ldf) and transaction log backups must not be confused. The latter is a production of backing up the former. Only the transaction log backup can be used in the roll-forward procedure.
SQL SERVER TRANSACTION LOG BACKUP

Every SQL Server database has a transaction log. AppSync supports SQL Server transaction log backup.

AppSync writes the log backups to the SQL Server’s default backup directory, by default. This can be customized in the service plan’s menu. If backing up logs for databases in a failover cluster environment, use shared storage or a network share so the log backups are written to the same location.

Use transaction log backups during the recovery of a production database, or when making a copy of a production database. Depending on the database recovery model, the transaction log volume can become full. To prevent the accumulation of logs, regularly run transaction log backups with the “Truncate the logs” option enabled.

AppSync can also backup transaction logs in AlwaysOn Availability Group (AAG) environments. AppSync can backup the primary or secondary database copy. When truncation is enabled, initiate the truncation process by backing up either the primary or secondary database transaction log.

Transaction log backups are supported using streaming backup only - they are not supported using VSS hardware snapshot technology. AppSync backs up transaction logs to a file. The file can be written to a local volume or network share using an UNC path; specified in the Backup path option.

NOTE: APPSYNC SUPPORTS UNC PATHS ON A NETWORK SHARE ONLY IF BOTH MACHINES ARE IN THE SAME DOMAIN.

Configure AppSync to enable transaction log backups for an SQL Server service plan by selecting the “Enable log backup” checkbox on the Create Copy options page. After selecting this checkbox, the “Transaction Log Backup Options” dialog box is enabled as depicted below in **Figure 1**, where the “Truncate the logs” option is selected. There exists an option to customize when, and how, to run log backups, and where to write the log backup files. Transaction log backups run sequentially.

![Transaction Log Backup Options](image)

**Figure 1 - Transaction Log Backup Options**

**RESTRICTIONS**

- To back up a transaction log, the database recovery model must be either “Full” or “Bulk-logged.” AppSync skips backing up the log for any database with the simple recovery model.
- To create any log backups with log truncation, first create at least one full database backup.
- To truncate transaction logs, AppSync must have a Full database backup copy type selected.
- Subscribe a database to only one service plan with log backup enabled.
- To truncate logs in an AAG environment, subscribe only one copy of a database to a service plan that is configured for Full database backups and transaction log backups with log truncation.
To back up transaction logs for databases that belong to an availability group, alter the schedule so that different copies of the database are not backed up at the same time.

LOG BACKUP EXPIRATION

AppSync expires log backups when the service plan runs to create a new log backup. During expiration, AppSync deletes the log backup file and removes information about the backup from the AppSync database. Log backups are eligible for expiration when the following conditions occur:

- The log backup is older than the service plan's "Minimum Retention Hours" setting.
- All older database backups are expired. The database backups included in this check depends on the SQL Server Backup Type.
  - If the log backup service plan has SQL Server Backup Type set to Full, then Full database backups created by any service plan are considered.
  - If the log backup service plan has SQL Server Backup Type set to Copy, only database backups created by that service plan are considered when looking for older database backups.

RESTORING SQL SERVER DATABASE COPIES

This section describes the methods for restoring a SQL Server database. The restore granularity for SQL Server database copies created by AppSync is based at the LUN level or at the consistency group level. Careful planning of the database layout is desirable to avoid any affected entities. See section titled **Affected entities during restore** for more detailed information.

RESTORING FROM THE COPY

Before restoring a database, it is highly recommended to back up the current transaction logs first. This minimizes the amount of data that is lost. AppSync can back up the transaction logs before the restore by utilizing the SQL Server Restore wizard. When selecting “Yes” to whether you wish AppSync to initiate a transaction log backup, you will be prompted to enter a location to copy the logs, and also a number of other options. You need to use one of the SQL Server management tools to restore it after the database has been restored, in order to recover the specific transaction logs to be replayed.

The **Recovery**, **No Recovery**, and **Standby** options are available when restoring the full database copy model. You may also restore from an AppSync SQL database copy when:

- The production database has been lost or deleted
- The production database has been corrupted
- The production database needs to be rolled back

Acknowledge the restore of affected databases that may come up due to the storage layout of the primary restore candidate. See section titled **Affected entities during restore** for more detailed information.

Depending upon the recovery type selected, either **Standby** or **Recovery**, AppSync performs database recovery post completion of a storage level restore. In case **No Recovery** is selected, the database is attached to the production instance but left in a non-operational mode. Subsequently, manual intervention is needed to fully recover the database.

AppSync only supports the **No Recover** mode during a restore of a SQL databases if the copy was taken using Non-VDI or Crash Consistent backup types. Restoring with **recovery** or **standby** when the copy was taken using Non-VDI or Crash Consistent copy types is not supported.

The following table depicts different restore scenarios and their preferred recovery types.
<table>
<thead>
<tr>
<th>Recovery Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recovery</strong></td>
<td>Recovers the database after restoring. Additional backups cannot be restored post recovery i.e. transaction logs cannot be rolled forward</td>
</tr>
<tr>
<td><strong>Standby</strong></td>
<td>Leaves the database in a standby state, where the database is available for limited read-only access. It rolls back uncommitted transactions, but saves the undo actions in a standby file so that recovery results can be reverted.</td>
</tr>
<tr>
<td><strong>NoRecovery</strong></td>
<td>Leaves the database in the restoring state and does not roll back the uncommitted transactions. This allows the transaction log backups to be restored in the current recovery path.</td>
</tr>
</tbody>
</table>

Table 1 - SQL Server Recovery Types

**AFFECTED ENTITIES DURING RESTORE**

An affected entity is data that resides along with data to be protected, on the production host, which becomes part of the copy. When restoring data items from a copy, a prompt will appear asking to acknowledge the other items which will be restored; in addition to the ones selected to restore.

Therefore, users have to be careful while planning their data layout for databases and must ensure that only those databases which are closely related and are required to be restored together by the user are protected together within the same copy.

If there are affected entities in your underlying storage configuration, the Restore Wizard notifies you of these items. Restore cannot proceed further without you acknowledging the listed of affected entities. If required, take necessary steps to detach affected databases and then proceed with restore.

The following scenarios produce affected entities that require you to acknowledge that additional items will be restored:

- If the databases are in the same consistency group, they become affected entities when the other database is protected.
- If the databases are on the same LUN (no consistency group involved), they become affected entities when one of the database from the same LUN is protected.

If the affected entity was protected along with a database that is selected for restore, it will be restored by AppSync. Any other database that was not protected, but is an affected entity, will also be overwritten.

AppSync calculates affected entities for consistency groups and LUNs of databases which are selected for restore. If the affected database(s) in turn, partially reside on other consistency groups or LUNs, AppSync does not calculate affected entities on those consistency groups or LUNs. Depending on the type of affected entity, the affected databases are detached by AppSync if they were protected in the same service plan as the database to be restored. For databases that were not protected in the same service plan run as the database to be restored, you must manually detach them from the SQL Server instance.

**Figure 2**, below, depicts the two types of affected entities seen during restore.
Affected entities are calculated only for the SQL Server instances where the credentials are configured. AppSync does a fresh database discovery for all these instances before calculating the affected entities. It is recommended not to share storage across databases that belong to different SQL Server Instances.

## BACKUP OF TRANSACTION LOGS

It is highly recommended to back up transaction logs before performing a restore. Every SQL Server database has a transaction log that records all transactions, and the database modifications made by each transaction. Optionally choose to backup transaction logs prior to restore in AppSync by specifying a location, including specifying a file prefix and extension. This backup can be used to recover the database in case the database is restored with either the NORECOVERY or STANDBY options. If the database is damaged, select the Database is damaged option to facilitate a tail log backup. See the Restoring damaged databases section.

Optionally choose to truncate transaction logs and overwrite existing backup files. These options are not available if the database is damaged.

Table 2 explains which user interface (UI) combination options can be selected to achieve the correct BACKUP LOG command as per the needs for taking transaction log backups during a restore.

<table>
<thead>
<tr>
<th>Database is damaged</th>
<th>Truncate the transaction logs</th>
<th>Overwrite existing backup files</th>
<th>Sample BACKUP LOG command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checked</td>
<td>N/A - Not Selectable</td>
<td>N/A - Not Selectable</td>
<td>WITH NO_TRUNCATE / WITH CONTINUE_AFTER_ERROR</td>
</tr>
<tr>
<td>N/A - Not Selectable</td>
<td>Checked</td>
<td>Unchecked</td>
<td>None</td>
</tr>
<tr>
<td>N/A - Not Selectable</td>
<td>Checked</td>
<td>Checked</td>
<td>WITH INIT, SKIP</td>
</tr>
<tr>
<td>N/A - Not Selectable</td>
<td>Unchecked</td>
<td>Checked</td>
<td>WITH NO_TRUNCATE, INIT, SKIP</td>
</tr>
</tbody>
</table>

Table 2 - AppSync UI options with Backup Transaction Logs
If choosing not to back up the transaction logs before the for restore operation, additional recovery options like Leave databases ready to use (RESTORE WITH RECOVERY) and Overwrite the existing databases (WITH REPLACE) appear in the user interface. The latter should be selected if intending to erase and replace the existing database with the backed up/restored data. This must be done carefully, otherwise, it may result in data loss or corruption.

**RESTORING DAMAGED DATABASES**

Damaged databases may have data files missing or damaged with their log files intact. Tail log backups capture the tail of the log even if the database is offline, damaged, or has missing data files. AppSync can take tail log backups for damaged databases. A damaged database must not contain bulk-logged changes and it must not be in an OFFLINE state.

If the production database is damaged and the Database is damaged check box is selected, AppSync backs up the tail log of the damaged database before proceeding with the restore. A damaged database can be in a RECOVERY_PENDING or SUSPECT state. AppSync first tries to detach the database by setting EMERGENCY mode. If AppSync fails to set the EMERGENCY mode on the database, it drops the database, and then proceeds with the restore. Once the restore operation has successful completed, continue to recover the database manually using the tail log backup.

For this case, the BACKUP LOG statement is executed with either NO_TRUNCATE or with CONTINUE_AFTER_ERROR option. The Truncate the transaction logs and Overwrite existing backup files options are not available when you select the Database is damaged option.

In cases where there are other ONLINE affected databases that were protected along with the damaged database, AppSync does not apply the damaged database scenario on any of them, rather, they are taken offline, detached, and then restored. The BACKUP LOG statement is executed with the NO_TRUNCATE option for all such databases.

**RESTORE RESTRICTIONS**

1. The database to be restored must not be a snapshot database.

2. A database snapshot is a read-only, static view, of a SQL Server database (the source database). The database snapshot is consistent transaction-wise with the source database as of the moment of the snapshot's creation. A database snapshot always resides on the same server instance as its source database. As the source database is updated, the database snapshot is updated.

3. The database to be restored must not have snapshot databases.

4. The database to be restored must not be mirrored. This means it should not have an active mirroring session.

5. A database cannot be restored if it is part of an Availability Group. AppSync does remove the database from the Availability Group as part of the restore process, but does not put the database back into the Availability Group after the restore process completes.

**MOUNTING SQL SERVER DATABASE COPIES WITH RECOVERY**

The mount host must have Microsoft® SQL Server installed if recovering databases from the mounted copy. If the mount host is a virtual machine, the vCenter server must be registered within AppSync. This is needed to mount RDMs as well as VMware’s virtual disks.

**MOUNT AND PRODUCTION HOST VERSION COMPATIBILITY**

- If the major version of the SQL Server instance on the production host is newer than that of the mount host, recovery fails for all databases belonging to that instance.

- If the major version of the SQL Server instance on the production host is older than that of the mount host, recovery succeeds only if the recovery type is either RECOVERY or NORECOVERY. Recovery fails if recovery type is STANDBY.

- If the major version of the SQL Server instance on the production host is the same as that of the mount host, but the minor version is newer, recovery fails for all databases belonging to that instance.

- If the major version of the SQL Server instance on the production host is the same as that of the mount host, but the minor version is older, recovery succeeds only if the recovery type is either RECOVERY or NORECOVERY. Recovery fails if recovery type is STANDBY.

- If the version of SQL Server instance on the mount host is newer than that of the production host, then the database version gets upgraded on recovery. However, in case of Dell EMC RecoverPoint, the changes are discarded when the bookmark is dismounted.
MOUNT SQL SERVER DATABASE COPY AS A CLUSTERED RESOURCE

- To mount a copy from a production cluster to an alternate cluster, as a clustered resource, select Mount and recovery copy and then select a clustered SQL server instance of the alternate cluster under the Recovery Instance drop down.

- Mount to a mount point (example: I:\mountpoint, where I:\ is a clustered disk) on the clustered mount host is supported. Specify the alternate mount path in the Mount path options, or specify different mount paths for each filesystem using the Mapped Path > Path Mapping Settings option. The root disk for the alternate mount point must be a clustered disk, and the SQL Server must have a dependency on that clustered disk.

- Multiple copies of the same database can be mounted to an alternate cluster at the same time.

- All recovery types are supported.

- Manually disable automount, by running “diskpart” at a command prompt, then entering “automount disable” at the DISKPART> prompt.

- Mount and recover copy as a clustered resource to a clustered production server is supported under the below considerations. The clustered SQL instance requires briefly being stopped and started along with any other dependent service. AppSync adds the newly introduced clustered disks as dependencies for the clustered SQL service while mounting, which requires to offline the clustered SQL service, and any dependent service. This is required to ensure the clustered disks are moved to the new node, along with clustered services, and are brought online before the services are brought online, in case failing over to another node.

SPECIAL CONSIDERATIONS FOR MOUNT TO PRODUCTION CLUSTER

- Mounting to a standalone instance on any production cluster node using the original path is not supported.

- For Mount and recover copy as a clustered resource to a production virtual server, consider the following:
  - Mounting to a different clustered SQL server instance is supported
  - Mounting to the same clustered SQL server instance is not supported
  - Mounting to an alternate mount point is supported, so long as the root disk for the alternate mount point is a clustered disk (must be added as a dependency to SQL server)
  - Mounting as Same as original path is not supported

- Performing a RecoverPoint mounted restore while the copy is mounted to a production cluster is not supported

- Mounting a Crash-Consistent or RecoverPoint API copy of a clustered SQL database, as a clustered resource, to another SQL instance residing on a production cluster, or any participant node of that production cluster, is not supported. This is due to fact that there is no VSS metadata generated for these type of backup types, and so the first mount itself is a VDS mount which is unable to perform a disk re-signature, therefore leading to mount failure.

SUPPORTED MOUNT RECOVERY TYPES

- **RECOVERY**: Instructs the recovery operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.

- **NORECOVERY**: Instructs the recovery operation not to roll back any uncommitted transactions. When in NORECOVERY mode, the database is unusable. This option is useful when the Database Administrator needs to restore one or more transaction log backups. The database is attached to the instance selected for recovery, and is then left in the RESTORING state.

- **STANDBY**: Recovers files and opens the database in a read-only mode. Subsequently, the Database Administrator can manually apply additional transaction log backups.

**RECOVERY, NO RECOVERY, AND STANDBY MODES ARE NOT SUPPORTED FOR NON-VDI AND CRASH CONSISTENT COPIES.**
IF RECOVERING A DATABASE FROM AN OLDER VERSION OF SQL SERVER ONTO A NEWER SQL SERVER VERSION, DO NOT USE STANDBY MODE. USING STANDBY MODE, THE UPGRADE TO THE NEWER VERSION CANNOT BE SUPPORTED AND WILL RESULT IN A FAILURE OF THE OPERATION.

- **ATTACH DATABASE**: Mounts the file system on which the database files are located and attaches the database to the SQL Server. The Attach Database Recovery Type option is only available for Non-VDI and Crash Consistent copies. Additional steps for full recovery of the database may be required.

**ATTACH DATABASE IS NOT SUPPORTED FOR FULL OR COPY SQL COPIES**

**MOUNTING COPIES USING PATH MAPPING**

The Mount Path > Mapped Path > Path Mapping Settings option mounts the copy to a host using a path mapping table set to user-defined locations. When using a path mapping table, there is more control over where the data will be located. Configure path mapping where the source file system and the target mount point is specified.

The following, seen in **Figure 3**, is a sample path mapping table. The first two target paths, G: and H: drives will be used rather than E: and F: from the source host, typically because a mount host is used for several source hosts using the same source drives. The I: drive must already be available on the mount host - the root drive for the mount path must pre-exist before attempting a mount.

![Path Mapping Settings](image)

**Notes**

- If a target path is not provided for a source path, then it is mounted as the same as the source path.
- Ensure that the absolute mount path on the target host is specified, otherwise, if the path is invalid, the mount fails.
- **Mount Copy Overrides** becomes unavailable if selecting the Mapped Path option.
- If one of the entered paths is invalid, the VSS import fails, therefore the entire mount fails. Partial failed scenarios are not supported for Windows mount operations.
- Nested target mount points are not supported.
- Path Mapping is not applicable to metadata paths.

**MOUNTING COPIES FROM MULTIPLE PRODUCTION HOSTS TO A SINGLE MOUNT HOST**

Multiple SQL server instances across hosts can be subscribed to the same service plan. When the service plan runs, it groups the databases-instance by host, and creates copies for all such groups – the database copies are grouped by hosts. These groups may further get divided on the basis of the load on each instance (see the Efficiency section for more details). A single mount host may be chosen for mounting, and optionally recovering, all such copies. Be sure and avoid any conflicting mount points or database names across hosts.
To avoid conflicting mount paths, choose the **Mapped Path** as the mount option, and provide different mount paths for each filesystem by clicking on **Path Mapping Settings**. **Mount Copy Overrides** are not available with the **Mapped Path** option. Alternatively choose the host level mount overrides to specify a different mount path for all databases that will be protected on that host.

To avoid conflicting database names on the mount host in case production database names are the same across hosts, choose to rename the suffix for all databases that will be protected on a particular production host. A database name conflict can only occur when intending to recover the databases post mount.

Failure to avoid such conflicts may result in partially mounted/recovered copies. Choosing separate mount hosts, one for each of the involved production hosts, helps to avoid any conflicts. This can be achieved by overriding the “mount host” / “recovery SQL Server Instance” for each of the production hosts involved. For details on configuring mount overrides, refer the *AppSync User and Administration Guide*, section **Overriding mount settings in a service plan**.

**MOUNTING MULTIPLE DATABASE COPIES EXISTING ON SAME PRODUCTION LUNS OR CONSISTENCY GROUPS**

This scenario may occur when the filesystems that are being mounted were already mounted for another set of databases in the same service plan run. If several databases share the same storage LUNs or Consistency Group, and the databases get split due to the restriction on the number of databases per run (see the *Efficiency* section for more details), only one of the resulting groups will be mounted successfully. The group that gets mounted first will mount successfully and the next ones will fail and a warning explaining the reason will be displayed.

The advanced option **Maximum number of SQL databases**, in the service plan’s **Advanced Plan Settings** menu, can be set to a higher value (default recommended value of 35) in order to force all databases in a single group to be copied together, however, it is not recommended from a performance perspective. Careful planning of the storage layout of databases in advance may prevent this issue from occurring.

**PARTIAL RECOVERY OF MOUNTED DATABASES**

AppSync is capable of partially recovering databases during a mount operation. This means that the entire mount operation does not fail if one or more databases fail to recover. For example, if trying to mount 10 SQL Server databases, three having name collisions with databases already present on the target SQL Server instance, then AppSync fails recovery for only those three databases. The remaining seven will be recovered successfully.

**UNMOUNTING SQL SERVER DATABASE COPIES**

In case of clustered unmount from a Clustered SQL instance, AppSync requires to briefly stop the clustered SQL services and any dependent services on it in order to properly remove the clustered disks from the clustered instance.

**CUSTOM SHUTDOWN SCRIPT PRIOR TO UNMOUNTS**

Prior to unmount, if you wish to perform a customized shut down of the database, you can place a script at the following location: `%ProgramData%\EMC\AppSync\script`. The script name must be in this format:

```<ServicePlanName>_<host_ProductionInstanceName OR ProductionInstanceName>_ShutdownSQL.bat```

Where:

- **ServicePlanName** is the name of the service plan to which the database is subscribed
- **host_ProductionInstanceName** OR **ProductionInstanceName**:
  - In **host_ProductionInstanceName**, replace **host** by another name
  - The **ProductionInstanceName** is needed irrespective of whether there are different SQL instances or not.
  - Use **ProductionInstanceName** in case of default production instance being equal to the host name

- Using the `_` as a separator in the script file name is mandatory.
It is recommended to run the script as a Windows user. To run the script as a SQL Server user in an SQL Server 2012 environment, the Local System user must have the **sysadmin** role.

The same customized Shutdown script must be present on all the participant nodes of an SQL clustered instance in case of a clustered mount.

In the absence of a customized script, AppSync performs a shutdown of the databases prior to unmount. This is not a configurable option in the UI, and AppSync looks for this script during the unmount operation. If the script is present and does not have proper access, or executable permissions, AppSync displays a warning and continues with the shutdown and unmount operation.

### SQL SERVER ALWAYSON INTEGRATION WITH APPSYNC

AppSync provides protection of databases residing in both **AlwaysOn Failover Cluster** and **AlwaysOn Availability Groups** environments. Failover cluster instances are only supported on single subnets.

### ALWAYSON FAILOVER CLUSTER INSTANCES

SQL Server can be configured with Microsoft AlwaysOn Failover Cluster Instances, wherein a clustered SQL Server instance is associated with Virtual Server. If one of the owner nodes in the cluster shuts down, then the clustered SQL Server instance fails over to the other available node.

AppSync can create and restore copies of clustered databases in a failover clustered instance. AppSync also automatically recognizes the node on which the clustered SQL Server instance is available; in case the instance has failed over to another node. It can also:

- Mount and recover copies of clustered databases as a clustered resource an alternate clustered instance of a production cluster or an alternate cluster.
- Mount and recover copies of clustered databases as standalone databases to alternate non-clustered instances of any node in the production or alternate cluster.
- Mount and recover copies of clustered databases to standalone servers.
- Mount and recover copies from standalone servers as standalone databases to cluster nodes or as clustered resource to clustered instances.

Refer to the AppSync User and Administration Guide for more information.

### PROTECTING DATABASES IN CLUSTERED INSTANCES

AppSync communicates with the SQL Server virtual server rather than individual nodes. The AppSync Host Plug-in is required on each of the cluster nodes first, then the virtual server must be registered.

To begin protecting databases in clustered instances, follow these steps:

1. Install the AppSync Host Plug-in by adding each node of the cluster to AppSync, or by first manually installing the software on each node then registering the host within AppSync. This must be completed before moving to the next step.

2. Add the SQL Server virtual server to AppSync. AppSync will discover only the Clustered SQL Server instances and clustered filesystems if applicable.
   - The SQL Server virtual server is the IP address or FQDN name that is registered as an IP address or network name in the cluster registry, respectively, and is directly associated with the SQL Server Role.

3. Enter the SQL Server connection settings by clicking on the SQL Server instance on the Microsoft SQL Server protection page. Use either SQL Server or Windows authentication as applicable. AppSync will discover the user databases that belong to the instance.

4. Subscribe databases to a service plan to begin protecting them, or subscribe the User Databases folder to a service plan to dynamically protect user databases. AppSync will protect any new user databases and stop protecting any deleted or offline databases using this method.

Manually delete any AppSync copies of deleted databases.
ALWAYSON AVAILABILITY GROUPS

The Availability Groups can be part of clustered, as well as non-clustered, SQL Server instances. An Availability Group is a set of SQL Server databases that all fail over together. Each availability group supports a set of primary databases and up to four secondary databases, each with a replica on a different node in failover cluster.

AppSync can create and restore copies of databases in availability groups by:

- Protecting databases in availability groups that are part of clustered or non-clustered instances
- Protecting primary or secondary databases
- Creating Full or Copy backups of primary databases
- Creating Copy backups of readable secondary databases. Readable secondary databases must be set to “Yes” in the Availability Replica Properties dialog box.

PROTECTING DATABASES IN AVAILABILITY GROUPS

1. Install the AppSync Host Plug-in by adding the node to AppSync for which to protect, or manually installing the software on the node and then registering it within AppSync.

   It is unnecessary to install the AppSync Host Plug-in on all of the nodes that are part of the availability group. The plug-in only needs to be installed on the node that host the databases which AppSync protects.

   If an instance is clustered, however, the plug-in must be installed on all of the nodes for which the instance can failover. In addition, the virtual server must also be added to AppSync as described in the Protecting databases in clustered instances section.

   If an instance is not clustered, ensure the node on which the instance is installed, is registered to AppSync.

   Figure 4 is an example of how to protect databases in a 3-node cluster with one clustered instance of SQL Server and one non-clustered instance. Lrmq123 and Lrmq124 are the possible owners of the clustered instance. Lrmq096 is the virtual server network name of the clustered instance. Lrmq125 hosts a non-clustered instance of SQL Server. To protect databases on any node in the cluster, the cluster nodes are added to AppSync and the host plug-in are installed. Then Lrmq096, the virtual server, is added.

2. From the Microsoft SQL Server protection page, select the SQL Server instance that owns the database replica that is to be protected and enter the SQL Server connection settings, using either a SQL Server login or Windows authentication as applicable. AppSync will discover the user databases that belong to the instance and display information about the availability groups that they belong to.

   Figure 5 shows the clustered instance LRMQ096\SQL2012CS1 and the non-clustered instance LRMQ125\SQL2012CS_SA1
3. Choose to either subscribe a primary or secondary database to a service plan so that AppSync will protect that database replica. If the role changes from primary to secondary, or vice versa, AppSync will continue to protect the same database replica. AppSync protects databases by SQL Server instance, not by role.

Selecting the Auto Switch to Copy is enabled ensures that AppSync will automatically switch to the Copy backup type, when the service plan is configured to do a Full backup, and the database is a secondary replica.

Figure 6 shows the information that AppSync gathers for databases in availability groups, if the database role is primary or secondary, and if it is secondary, the type of commit that is configured. The “Database Status” column shows if the database is online, and if AppSync can protect it. When “Online, readable” is displayed, AppSync protects the secondary database. When “Online, not readable” is displayed, and the secondary database’s “Readable Secondary” option is not set to “Yes” within SQL Management Studio, the secondary database is not able to be protected by AppSync. “Read-intent only” is not supported. See Figure 7 for an example of the “Readable Secondary” options.
RESTORING DATABASES IN AVAILABILITY GROUPS

AppSync can restore a database from a copy taken from the primary or secondary replica. These general guidelines apply:

- The database being restored must be removed from the availability group before it is restored. AppSync will do this automatically, however, it will need to add back to the availability group once the recovery has completed.
- In most cases, when restoring the primary replica, the secondary replica needs to be reseeded.
- If restoring to a secondary replica which is to be reseeded to the other replicas, first use the option to failover the primary replica. Once recovery is complete, use the restored replica to reseed the other database replicas.
- AppSync restores to the server (or SQL Server virtual server) that was used to create the copy.
- Restores are at the volume level, so all databases on the volume will be impacted.
  - If the databases were protected together, an option is presented for restoring and recovering the other database(s).
  - If the databases were not protected together, the other databases will be overwritten, but not recovered. Dethatching them from SQL Server before running the restore is required.

DETAILS OF RESTORING TO A PRIMARY REPLICA

The following explains the steps that AppSync performs when restoring to a primary replica, as well as the steps that you should perform after the restore is complete.

When restoring to a primary replica, AppSync will:

1. Suspend data movement for all replicas.
2. Remove the database and all of its replicas from the availability group.
3. If the replica being restored is currently a secondary replica and the **Failover the Availability Group if the current role is Secondary** option was selected, AppSync will failover the availability group.

THE OTHER DATABASE REPLICAS IN THE AVAILABILITY GROUP MUST BE HEALTHY OR THE RESTORE WILL NOT PROCEED

4. Take the database offline and detaches it

5. Restore the volume(s) to the recovery state selected from the Restore wizard – RECOVERY, NORECOVERY, or STANDBY. If there are transaction log backups that need to be restored, use NORECOVERY or STANDBY.

---

**Figure 8 - SQL Server Restore Options**

After AppSync recovers the database, manually perform the following steps:

1. Restore any log backups that are required, and recover the database

2. To reseed secondary databases using the “Add Database to Availability Group” wizard’s full synchronization option, delete the secondary databases.
3. Add the primary database back into the availability group. One way to do this is to use the SQL Server Management Studio:

3.1 Connect to the server instance hosting the primary replica

3.2 Expand the AlwaysOn High Availability node and the Availability Groups node

3.3 Right-click the availability group and select Add Database to launch the Add Database to Availability Group wizard

3.4 Select the database on the Select Databases page. If a database does not meet all of the prerequisites, the Status hyperlink provides a brief explanation of why the database is ineligible. For more information, click the hyperlink.
3.5 Select the data synchronization preference on the **Select Initial Data Synchronization** page

![Select Initial Data Synchronization](image1.png)

3.6 Connect to secondary SQL Server instances on the next page and click **Next**

3.7 If the validation is successful, click **Next**, otherwise correct the problem and click **Re-run Validation**

3.8 Review the summary and click **Finish**

![Add Database to Availability Group Summary](image2.png)

---

Figure 11 - Select Data Synchronization

Figure 12 - Add Database to Availability Group Summary
3.9 If selecting the Full or Join only synchronization options, the primary and secondary databases should be added again to the availability group. If selecting to Skip initial data synchronization, refer to Books Online for SQL Server for details on how to start data movement to the secondary database.

DETAILS OF RESTORING TO A SECONDARY REPLICA

The following explains the steps that AppSync performs when restoring to a secondary replica, as well as the steps that should be performed after the restore is complete. Restoring a secondary replica may be useful if the replica is behind reseeding it is desired.

If the replica being restored is currently a secondary replica, and the Failover the Availability Group if the current role is Secondary option is selected, refer to the Details of RESTORING TO A PRIMARY REPLICA section.

When restoring to a secondary replica, AppSync will:

1. Suspend data movement of just the selected secondary replica.
2. Remove the selected secondary database from the availability group.
3. Restore the volume(s) to the recovery state selected from the Restore wizard. For secondary replicas, use the NORECOVERY. You want to leave the database in this state so that it can rejoin the availability group.

After AppSync restores the database, perform the following steps manually:

1. Restore any log backups that are required, and leave the database in the NORECOVERY state
2. Join the secondary database back to the availability group using the following T-SQL statement:

```sql
ALTER DATABASE [AAGName] SET HADR AVAILABILITY GROUP = [DBName];
```
SQL SERVER DATABASES ON VMWARE VIRTUAL DISKS

AppSync provides the ability to protect, mount, and restore SQL Server databases residing on VMware virtual disks. The SQL Server instance can be clustered or non-clustered.

PREREQUISITES AND RESTRICTIONS

- For successful mapping, the Virtual Center must be added to the AppSync server and discovery must be performed.
- For successful protection, both the log and database files must reside on virtual disks. There cannot be a combination of physical and virtual storage.
- Non-persistent virtual disks are not supported.
- NFS datastores are not supported.
- Protection of SQL Server databases across virtual machines sharing the same datastore are not supported.
- If the ESX version is less than 5.0:
  - Mounting a copy of a non-clustered SQL database to the production server, or cluster, is not supported.
  - Mounting a copy of a clustered SQL database to the production server is supported only from second mount onwards, that is, the copy has been previously mounted outside of production clustered.
- When mounting virtual disks in ESXi 5.x and RecoverPoint 4.0 environments, disable hardware acceleration to ensure successful virtual access type mounts. For more details, refer the VMware Knowledge Base article 2006858.

ADDITIONAL INFORMATION

The following is additional information about creating, mounting, and restoring AppSync copies of virtual disks.

When VSS used

Although SQL Server’s VDI API is used to create application consistent copies, VSS is also normally used to flush I/O’s and create shadow copies of the volumes. If the ESX version is less than 5.0, VSS is not used.

Restore

All the databases residing on the same datastore are displayed as affected entities. Restores of virtual disks will restore the entire datastore. If multiple databases are protected alongside the database being restored, they are display as affected entities and can optionally recover them. Otherwise, manually detach the databases so the restore can overwrite them. They can be attached at a crash consistent recovery.

If RecoverPoint is used, restore is at the consistency group level. If there is another datastore in the same consistency group, it will also be restored. If AppSync detects any databases on that datastore, they are also displayed as affected entities.

If restoring in a clustered environment, ensure all nodes of the cluster are added to AppSync. This is so AppSync can remove the source virtual disk from all the VMs in the cluster.

REPURPOSING SQL SERVER DATABASE

AppSync allows you to create copies of your SQL Server database for application testing and validation, test and development, reporting, data masking, and data analytics. AppSync identifies copies that are created from a repurpose action as first generation and second generation copies. The source of a second generation copy is a first generation copy. You can create multiple second generation copies from a first generation copy.
Two types of repurposing

- Native array repurposing - The first generation copy is a copy of the source database. For example, an array based snapshot of the source is the first generation copy

- RecoverPoint bookmark repurposing - The first generation copy is a copy of the LUN at the local or remote replication site in the RecoverPoint consistency group – the first generation copy is the array LUN taken from the LUN which is used by RecoverPoint as an “intermediary step”

Additional information about repurposing

- The first generation copy can be used as source for multiple second generation copies

- Both first and second generation repurposed copies can be either on-demand or scheduled

- Restoring second generation copies is not supported – only first generation copies can be restored

- Restoring a first generation copy is not supported in the case of RecoverPoint bookmark repurposing

- Restoring a first generation copy is not supported in the case of remote VMAX operations – remote VMAX copies cannot be restored

- The first generation copy of a database is generally an application consistent copy. It includes application discovery, mapping, and database freeze/thaw – with options to use Non VDI or Crash-Consistent as seen in Figure 14

![Figure 14 - Repurpose SQL-Specific Copy Options](image)

- **VSS Retry Options** can be configured by clicking the Advanced Plan Settings. This includes a VSS retry count and retry interval in seconds for freeze/thaw operations using the repurposing workflow. VSS retry options are not applicable when creating Crash-Consistent SQL copies

- Second generation copies are created using the first generation copy as the source, without impacting the application. They do not include application discovery, mapping, nor application freeze/thaw. If a first generation copy is mounted with recovery when the second generation copy is refreshed, the second generation copy might not be recoverable after the mount

- Refreshing mounted Crash-Consistent or Recover Point APIT copies of clustered SQL databases, which are mounted to a production cluster, is not supported. VDS mounts will fail due to disk signature conflicting with production disks.

- Log backups are not supported as part of repurposing

- Repurposing multiple SQL databases together is not supported
DIFFERENCES BETWEEN APPSYNC AND REPLICATION MANAGER

- AppSync supports multiple virtual disks shared on the same datastore. Databases on different virtual disks can reside in the same datastore, however, when restoring a database, all other databases sharing the same datastore will also be restored. This is because AppSync restores at the datastore level.
- AppSync does not require manual re-signaturing. AppSync takes care of the re-signaturing.
- AppSync supports production mounts of virtual disks.

CONCLUSION

This whitepaper discusses utilizing Dell EMC AppSync to support backup, recovery, and repurposing workflows for Microsoft SQL Server standalone and clusters. This white paper discusses such features as dynamic discovery of databases during the protection phase and the usage of pre and post scripts. It also explains various options and scenarios for mounting, restoring, and repurposing of SQL Server database copies.

LISTING OF FIGURES

Figure 1 - Transaction Log Backup Options..........................................................7
Figure 2 - Effected Entities Restore Warning .......................................................10
Figure 3 - Path Mapping Settings ........................................................................13
Figure 4 - Protecting Databases in Availability Groups ........................................16
Figure 5 – SQL Server Protection Page.................................................................17
Figure 6 – User Databases ....................................................................................17
Figure 7 - Readable Secondary Options ...............................................................18
Figure 8 - SQL Server Restore Options ................................................................19
Figure 9 - SQL Management Studio Restoring ....................................................20
Figure 10 - Select Databases for Availability Group .............................................20
Figure 11 - Select Data Synchronization ................................................................21
Figure 12 - Add Database to Availability Group Summary .................................21
Figure 13 - Add Database to Availability Group Results .....................................22
Figure 14 - Repurpose SQL-Specific Copy Options .............................................24

LISTING OF TABLES

Table 1 - SQL Server Recovery Types .................................................................9
Table 2 - AppSync UI options with Backup Transaction Logs ...............................10

REFERENCES