



---

TECHNICAL WHITE PAPER

# Exploring the Depth of Simplicity: Protecting Microsoft SQL Server with Rubrik

# TABLE OF CONTENTS

<b>AUDIENCE .....</b>	<b>3</b>
<b>EXECUTIVE SUMMARY .....</b>	<b>3</b>
<b>COMMON CUSTOMER CHALLENGES .....</b>	<b>3</b>
<b>OVERVIEWS .....</b>	<b>4</b>
Capability Overview .....	4
Setup Overview .....	4
SLA Domain Policy Overview.....	5
Configuration & Restore Overview.....	7
<b>DEEP DIVE .....</b>	<b>10</b>
Incremental Forever Snapshots.....	10
Granular Database Protection.....	10
Point-In-Time Restores.....	11
Restore User - Role Based Access Control.....	11
Simple Configuration.....	12
Transparent Connector Upgrades .....	12
Auto-Discovery .....	12
Centralized Management .....	12
Encrypted Database Support .....	13
Flash Optimized Parallel Ingest.....	13
Flexibility to protect SQL at a VMware level.....	13
<b>SQL DATABASE RECOVERY OPTIONS.....</b>	<b>13</b>
<b>ADVANCED SQL FUNCTIONALITY SUPPORT.....</b>	<b>13</b>
Always On .....	13
Windows Server Failover Clustering (WSFC) with SQL Server.....	13
Cross Version Restore .....	14
Recovery to a Specific LSN via the API .....	14
<b>UNDER THE HOOD .....</b>	<b>15</b>
SQL Backup Flow.....	15
SQL Restore Flow .....	15
<b>CONCLUSION.....</b>	<b>15</b>

## AUDIENCE

This white paper is intended for backup administrators and DBA's to provide a deep dive around the implementation and benefits of Rubrik's support for Microsoft SQL Server backups.

Note: for the remainder of this white paper, we will simply refer to "SQL Server" to be concise rather than "Microsoft SQL Server", "SQL", or "MS SQL".

## EXECUTIVE SUMMARY

Many organizations use Microsoft SQL Server for their critical applications. Throughout the years SQL Server has steadily improved to become a critical component of modern datacenters. Despite these improvements, SQL Server backups have often been a question of trade-offs between cost, efficiency, and simplicity.

Rubrik's SQL Server backup functionality builds on top of Rubrik's policy-driven architecture extending it to SQL backups. Although SQL Server environments can be complex, Rubrik's support for SQL Server backups aligns with the core architectural and operational simplicity of the Rubrik platform.

For a walkthrough of Rubrik architecture, see the "[Technology Overview & How It Works](#)" white paper.

## COMMON CUSTOMER CHALLENGES

In creating Rubrik's SQL support, we discussed with our customers what challenges they were seeing and incorporated those into our design goals. We heard...

"Taking backups of our 5TB database does not fit into our backup window."

"Managing agents is very painful. DBAs or Sysadmins own the primary servers and upgrading agents creates a lot of tension between teams."

"With our current solution, the agent does not list out the databases on the host. Manually identifying databases on the host and protecting them is painful."

"We have to store two daily full-snapshots on the primary server to provide a reasonable RTO. This consumes a significant amount of space on the primary."

## OVERVIEWS

### CAPABILITY OVERVIEW

Aligning with the challenges we heard from customers, the key benefits of our approach are:

- [Auto-discovery](#) of all instances, databases, and clusters on each SQL Server lowering operational overhead during configuration
- [Centralized management](#) via visibility in the Rubrik UI of all databases both being backed up as well as excluded from backup
- [“Incremental-forever” backups](#) via block mapping and intelligent transaction log handling to dramatically reduce local storage requirements, provide much faster backups, and reduce network usage
- [Granular database protection](#) via Rubrik SLA policies - ability to have different policies at the server and instance level as well per database on the same SQL Server or instance
- [Seamless point-in-time restore](#) via a very simple, efficient interface providing full-backup restores + transaction log replay via a single operation
- [Log truncation and log management](#) providing further operational time savings
- [Copy-only mode](#) for seamless transition or coexistence with existing backup product
- [Full application awareness](#) in high availability deployments like Always On Availability Groups and Windows Failover Cluster.

We'll explore these and more in the feature Deep Dive below.

### SETUP OVERVIEW

Rubrik supports all versions of SQL supported by Microsoft via Extended Support - Windows Server 2008 R2 and newer with SQL Server 2008 and newer. Please see the [Rubrik Compatibility Matrix](#) for supported version details. Per Microsoft, Extended Support for SQL Server 2005 [ended in April 2016](#).

To provide integrated SQL functionality, Rubrik uses a lightweight SQL Connector which uses minimal CPU and Storage. The connector encrypts all backup traffic. As an MSI, it can be installed manually or easily deployed via standard provisioning tools. The connector does not require changing existing maintenance plans or rebuilding them from scratch.

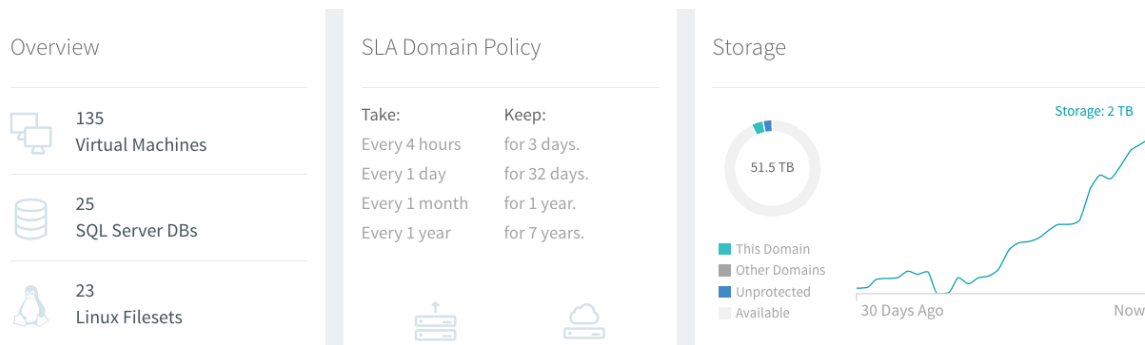
To reduce operational overhead, connector upgrades after installation are automatic and completely transparent to the user.

Once the connector is installed, SQL databases are automatically discovered. The user can assign policies to the individual databases – these policies are core to Rubrik and can be leveraged across multiple data types.

## SLA DOMAIN POLICY OVERVIEW

A Rubrik SLA Domain Policy is a declarative policy encompassing the core items needed for backup and recovery replacing the need to individually configure jobs, tasks, and other items. SLA Domain Policies are a core part of Rubrik's architecture which extend across all data types as shown here.

Figure 1: Rubrik Policy Overview



Let's walk through the pieces needed to configure an SLA Domain Policy that can apply to all data types - we'll look at SQL specific items in the next section.

1. **Backup Frequency:** this is also known as Recovery Point Objective (RPO). Simply put, how often are backups are taken?
  - For databases, this determines how often a database restore point is synthesized from incrementally transmitted block maps. For databases in Full Recovery mode, RPO is also impacted by the frequency of transaction log backups.
2. **Availability Duration:** this is also known as retention. Simply put, how long are backups retained?
  - For databases, retention may often be shorter than other data types unless needed for regulatory or compliance reasons.

Figure 2: Rubrik Policy Components - Frequency and Duration

Service Level Agreement  
Choose how often we take Snapshots, and the length of time we keep them.

Backup Frequency	Take Snapshots:	Keep Snapshots:	Availability Duration
	Every (Hours) <b>4</b>	For (Days) <b>3</b>	
	Every (Days) <b>1</b>	For (Days) <b>32</b>	
	Every (Months) <b>1</b>	For (Years) <b>1</b>	
	Every (Years) <b>1</b>	For (Years) <b>2</b>	

3. **Archival Policy:** this is also known Recovery Time Objective (RTO) or “When and Where to Archive”. Archive targets can be public cloud (AWS or Azure) or private cloud (S3 compatible object stores or NFS). This dictates which cloud target is used for archive and when archives are maintained solely in the cloud and not on the local Rubrik cluster. If archives are maintained solely in the cloud (past 30 days for instance), RTO is longer due to the time required to retrieve back to the Rubrik cluster.

  - For databases, long-term archive required for regulatory or compliance reasons can be stored in a cloud archive.

Figure 3: Rubrik Policy Components - Archival

### Archival Policy


- Enable Archiving
- Enable Instant Archive ⓘ

Archival Location

Azure:se3demo ▾

---

Move the slider to choose how long data is kept on the local Rubrik cluster before archiving.



66 days

Snapshots will be stored on the local Rubrik cluster for 66 days . Data will then be moved to your archival location and kept there for 6 years 299 days. Snapshots older than 7 years will no longer be available.

4. **Replication Policy:** this relates to Disaster Recovery. Simply put, how much replicated data should be maintained at a DR site?


  - For databases, this will often be a shorter value. In a Disaster Recovery situation, recovery of the most recent state of a database is most common. This policy section allows cost savings by storing a recent subset of data at a DR site.

Figure 4: Rubrik Policy Components - Replication

### Replication Retention Policy

- Enable Replication

Move the slider to choose how long data is kept on the replication target. Leftmost means only the most recent replicated snapshot will be maintained.



30 days

Snapshots will be stored on the replication target for 30 days.

The policy architecture is intentionally straightforward to configure yet powerful - the screenshots above illustrate the concepts well. Please see Rubrik documentation and the core architecture white paper for a more thorough walkthrough of SLA Domain details.

## CONFIGURATION & RESTORE OVERVIEW

As noted above, SLA Domain Policies can be applied at the SQL server, instance, database, or cluster level. A visual walkthrough illustrates the concepts well along with illustrating the notably few SQL specific configuration options.

*A list of auto-discovered SQL inventory at the server and database level - instance level is also available.*

Figure 5: SQL Inventory - Server Level

Name	Cluster	Instances	Status	SLA Domains
DEMO-MEGASQL	--	1	Connected	Gold, Unprotected
DEMO-SQL12-WFC1	DEMO-WFCluster1	0	Connected	--
DEMO-SQL12-WFC2	DEMO-WFCluster1	0	Connected	--

Figure 6: SQL Inventory - Database Level

Name	Availability Replica	Log Backup	Copy Only	SLA Domain
BIGDB	No	N/A	No	Unprotected
master	No	N/A	No	Unprotected
model	No	--	No	Unprotected
msdb	No	N/A	No	Unprotected
smalldb	No	--	No	Unprotected
SSDdb	No	5 min	No	Gold

After selecting a server, instance, or database, clicking the “Manage Protection” button brings up the policy assignment screen where can assign a policy as well as set options around copy only backups, log backup frequency, and log backup retention.

Figure 7: SQL Policy Management

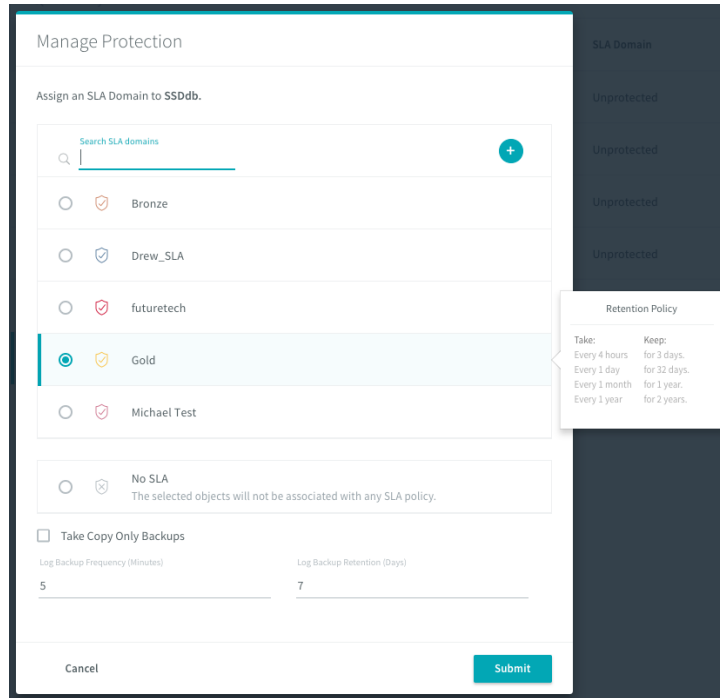
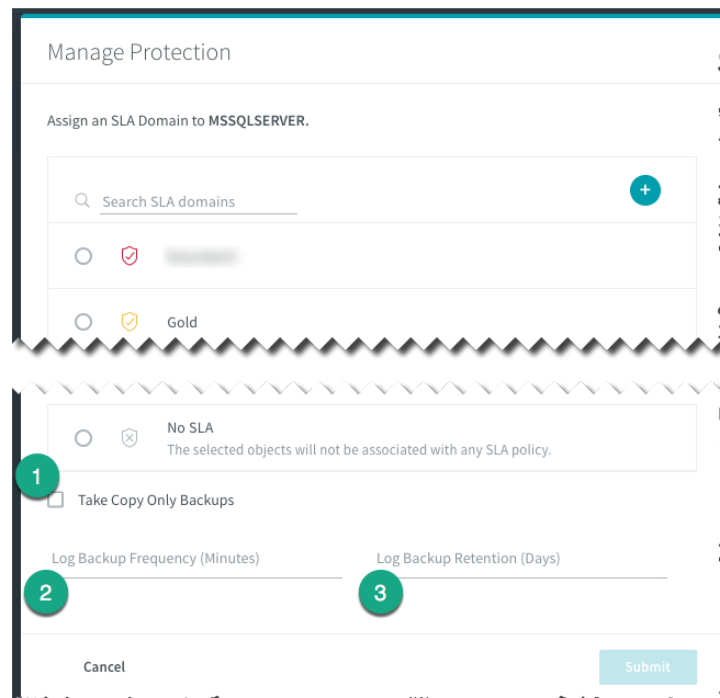


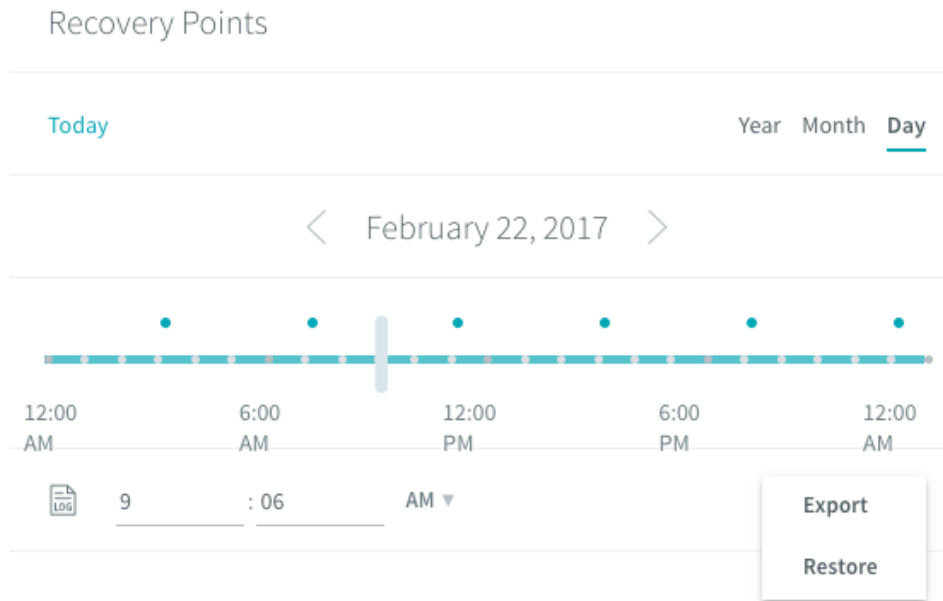
Figure 8: SQL Policy Management





Once protected, the restore process is a simple slider for point-in-time restore.

Figure 9: Point in Time Restore Control



## DEEP DIVE

Let's explore in depth at a feature by feature level the items discussed briefly in the overview above. Many of the features explored below are intentionally transparent during day to day operations.

### INCREMENTAL FOREVER SNAPSHOTS

Incremental Forever snapshots dramatically reduce storage usage as well as network traffic both inside the datacenter as well as over replication links. Although SQL Server does not natively support incremental-forever backups, Rubrik can provide this capability via block mapping.

Databases using Full recovery model can be protected through policy driven snapshots and backups of the transaction log, or through policy driven snapshots only. The Rubrik cluster performs an initial full database backup followed by periodic block mapping to detect and transmit changed data based on the assigned policy. Additionally, there are frequent interim backups of the transaction log with ability to specify the default frequency transaction log backups as well as retention of transaction log backups.

Figure 10: SQL Specific Policy Options

The screenshot shows a configuration interface for SQL Specific Policy Options. It includes a checkbox labeled "Take Copy Only Backups" which is currently unchecked. Below this, there are two input fields: "Log Backup Frequency (Minutes)" with a value of "5" and "Log Backup Retention (Days)" with a value of "7".

The combination of database snapshots and transaction log backups permits granular restore of a database to a specified recovery point. See the “Advanced SQL Functionality” section below for details on the “Recovery to a Specific LSN via the API” feature. During backups, transaction logs can be either truncated or left untouched via a “Copy Only” mode - a checkbox option available during SLA assignment.

For databases using Simple recovery model, the Rubrik cluster performs policy driven snapshots of the database similar to the approach outlined above.

### GRANULAR DATABASE PROTECTION

Protection policies can be assigned at the Windows host level, an entire SQL instance, any individual database, and even multiple overlapping levels - the most granular assignment has priority. Derived assignment provides a way to uniformly manage and protect those databases however only applies to the databases that exist at the time of the assignment. This provides flexibility on SQL servers hosting many databases for different purposes - some of which may have more stringent RPO and RTO requirements than others.

Rubrik can easily assign unique policies to individual databases.

See the “SLA Domains” column in the screenshot below as an example.

Figure 11: SLA Domains

Name	Location	Availability Replica	Log Backup	Copy Only	SLA Domain
AdventureWorks2012	se-dhutton-win/MSSQLSERVER	No	15 min	No	Gold
AdventureWorks2012_...	se-dhutton-win/MSSQLSERVER	No	--	No	Unprotected
AdventureWorks2014	SE-JBURRELL-WIN/MSSQLSE...	No	10 min	No	Bronze

## POINT-IN-TIME RESTORES

SQL Databases often support critical workloads which require an RPO in minutes. Rubrik achieves this by backing up the transaction logs in addition to the databases. During the restore process, a user specifies a desired restore time simply by choosing a day and dragging a slider to the desired time. Alternately, a specific time can be typed in. The system then performs the following steps:

- Restores the full snapshot closest to the user specified time
- Replays and applies transactions starting from the point of snapshot to the time specified by the user. This is often known as “rolling transaction logs”.

While the amount of time needed to restore is dependent on multiple variables (network speed, primary storage, and more), the time for replaying transaction logs can be reduced by specifying more full snapshots - something configurable via Rubrik policy.

Figure 12: Point in Time Restore Controls

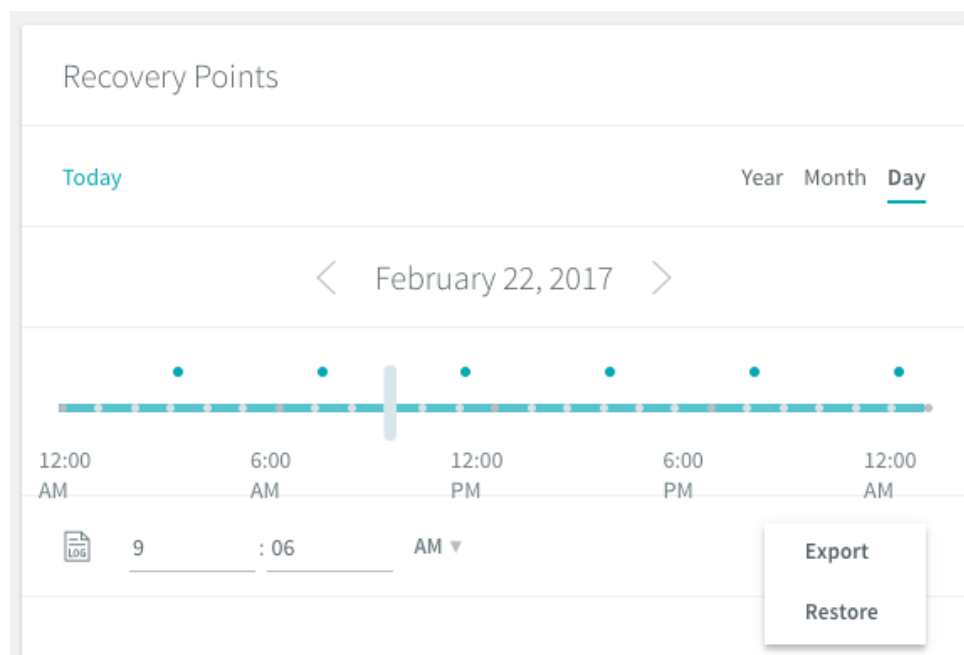
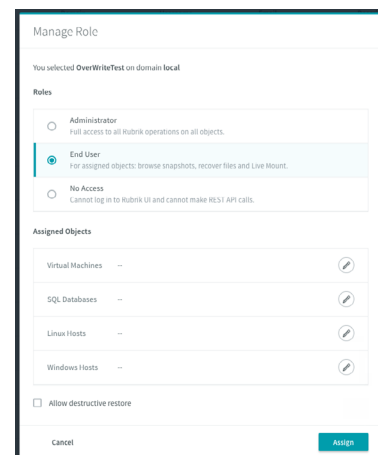


Figure 13: Role Based Access Control

## RESTORE USER - ROLE BASED ACCESS CONTROL

Restores can be performed either by a Rubrik administrator or via the “End User” role - a role that is assigned granular per object permissions on creation to perform restores. Users with this role can specifically be granted permission to perform in-place “Destructive Restores” which will overwrite existing data. Alternatively, the End User role can Export to a new location as described below.



## SIMPLE CONFIGURATION

There is no job configuration and no requirement for a staging server - simply the Rubrik cluster and SQL servers with the connector installed.

## TRANSPARENT CONNECTOR UPGRADES

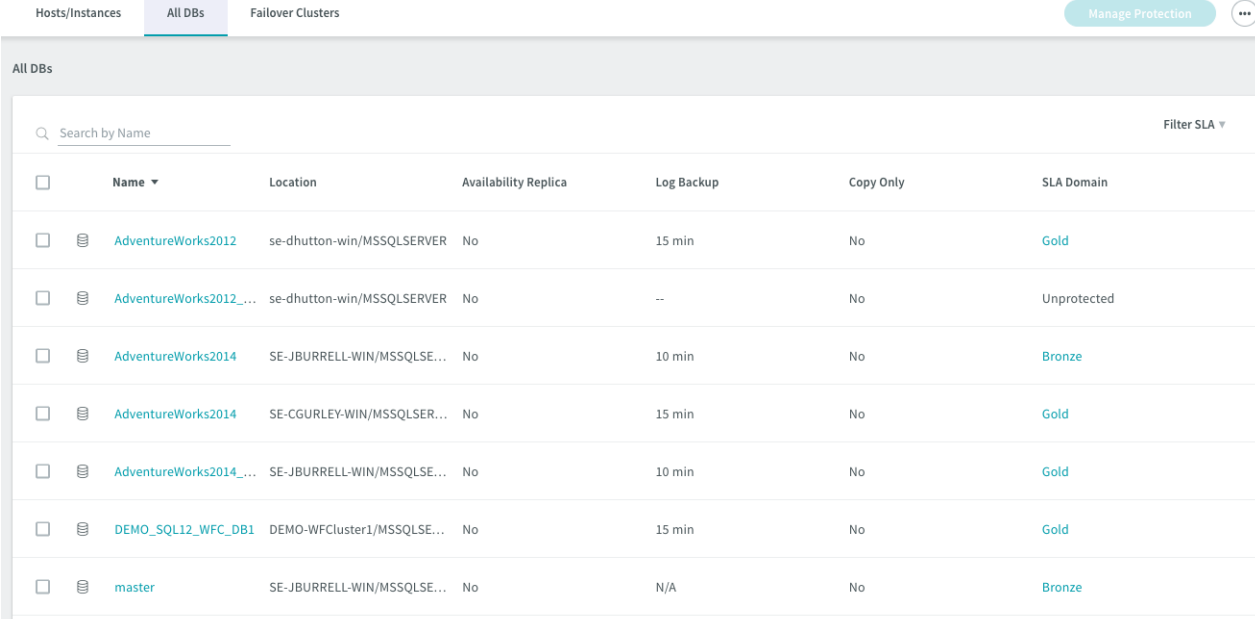
In larger environments, upgrading backup agents with new versions can be time-consuming and often delay backup environment upgrades due to time required for agent updates.

Rubrik's ability to automatically and transparently upgrade its SQL connector removes the time needed for manual agent updates. We have done this via an inner and outer core design - the outer core detects software upgrades and upgrades the inner core connector. The connector upgrade does not require a restart of SQL Server or the underlying Windows host.

## AUTO-DISCOVERY

Rubrik auto-discovers all instances and databases on each SQL server where the connector is installed. Multiple views are then provided which are easily sortable and searchable - Hosts/Instances, All DBs, and Failover Clusters.

Figure 14: Auto-Discovery of all databases



<input type="checkbox"/>	Name	Location	Availability Replica	Log Backup	Copy Only	SLA Domain
<input type="checkbox"/>	AdventureWorks2012	se-dhutton-win/MSSQLSERVER	No	15 min	No	Gold
<input type="checkbox"/>	AdventureWorks2012_...	se-dhutton-win/MSSQLSERVER	No	--	No	Unprotected
<input type="checkbox"/>	AdventureWorks2014	SE-JBURRELL-WIN/MSSQLSE...	No	10 min	No	Bronze
<input type="checkbox"/>	AdventureWorks2014	SE-CGURLEY-WIN/MSSQLSER...	No	15 min	No	Gold
<input type="checkbox"/>	AdventureWorks2014_...	SE-JBURRELL-WIN/MSSQLSE...	No	10 min	No	Gold
<input type="checkbox"/>	DEMO_SQL12_WFC_DB1	DEMO-WFCcluster1/MSSQLSE...	No	15 min	No	Gold
<input type="checkbox"/>	master	SE-JBURRELL-WIN/MSSQLSE...	No	N/A	No	Bronze

## CENTRALIZED MANAGEMENT

Although an overused term, Rubrik does provide a “single pane of glass” for all supported backup workloads. For SQL specifically, Rubrik customers can see all backed up SQL servers, instances, and databases in a single interface. The same UI and the same policy engine is used whether for SQL, VMware, Linux, or Windows.

This does not preclude the ability for DBA's to verify backup success from SQL Server Management Studio via querying the “[backup set](#)” table in the “msdb” database which records all successful backups. See [this link](#) on MSDN for further details. As well, a full walkthrough of this process is available as a KnowledgeBase article in the Rubrik Support Portal.

## ENCRYPTED DATABASE SUPPORT

Rubrik will backup encrypted databases and fingerprint-based compression will also work on encrypted databases. For restore, the workflows are the same with one additional step - users must manage keys manually. Steps required to move keys are detailed in [this Microsoft article](#). Once this is done, the intended database can be exported from the Rubrik UI.

## FLASH OPTIMIZED PARALLEL INGEST

Although a core Rubrik capability, flash optimized parallel ingest is particularly relevant for large database backups. Backups are taken in parallel across multiple nodes due to Rubrik's distributed job scheduling and land on flash before destaging to disk. This removes bottlenecks during large initial backups and allows faster protection.

## FLEXIBILITY TO PROTECT SQL AT A VMWARE LEVEL

In a virtualized environment using Simple Recovery Mode, Rubrik's enhanced VMware backup capabilities may be sufficient with even lower operational overhead. While this does not include many of the specific benefits listed in this paper, it does provide backup consistency via Rubrik's VSS implementation, Instant Recovery via Live Mount, and Object Level Recovery using Kroll. For databases in simple recovery mode, this may meet or exceed business requirements. This approach notably does not provide Point In Time Restore, Granular Database Protection, and Log Management.

## SQL DATABASE RECOVERY OPTIONS

There are two recovery options for SQL Databases - Restore and Export.

Choosing Restore drops the original database and creates a new database on the same instance with the same name and file structure. A common use case for this option is corruption of the original database where a restore to a previous "point in time" is desired.

Choosing Export creates a new database. If restoring to the same SQL instance, a different database name is used with file structure reflecting that name. If to a different SQL instance, the same or different database name can be used. A common use case for this option is use a production database snapshot as the source to spin up a dev/test clone.

Given SQL Server's capability for a database to have multiple data and log files, customers can now specify at an individual file level the restore location during an Export operation.

## ADVANCED SQL FUNCTIONALITY SUPPORT

### ALWAYS ON

Always On is a high availability solution provided by SQL Server. Additional details are available from Microsoft in [Overview of Always On Availability Groups \(SQL Server\)](#).

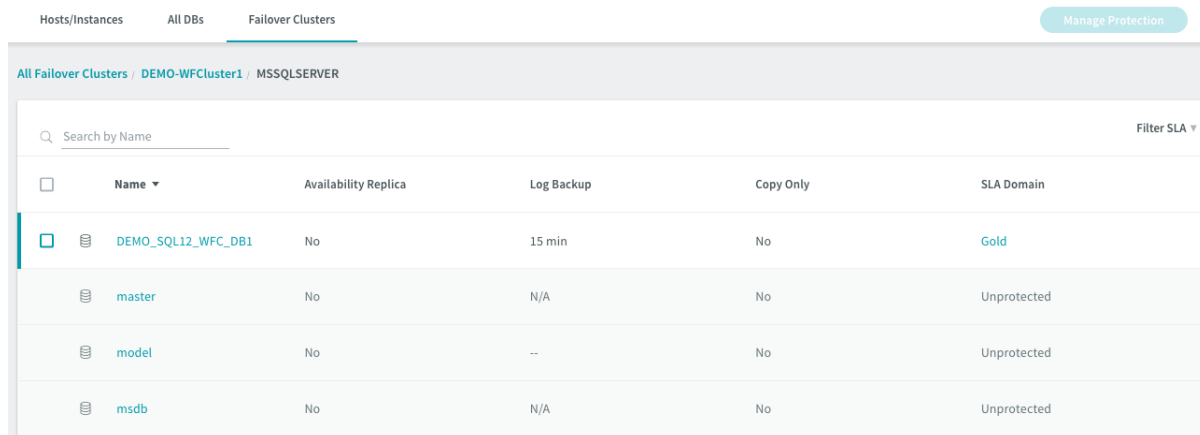
Rubrik will automatically detect if databases are part of an Availability Group and not "double backup" the same database located on multiple servers in the same Availability Group. In the case of Always On failover, simply switch protection to the new database with no loss of history and no need for a new full backup.

## WINDOWS SERVER FAILOVER CLUSTERING (WSFC) WITH SQL SERVER

Windows Server Failover Clustering (WSFC) with SQL Server is a popular option for providing SQL high availability due to its lower storage and SQL licensing requirements. Please see [this Microsoft article](#) for more details.

Rubrik supports WSFC with SQL Server for backups. After installing connectors on Node A & Node B, Rubrik will recognize an entity called “Failover Cluster” with its underlying instances and database. Use the same methods in the Rubrik UI for SLA assignment as any other SQL databases. In case of failover, no manual steps are required - Rubrik will automatically protect the database on Node B with no new full backup required.

Figure 15: Failover Clusters



<input type="checkbox"/>	Name ▾	Availability Replica	Log Backup	Copy Only	SLA Domain
<input checked="" type="checkbox"/>	DEMO_SQL12_WFC_DB1	No	15 min	No	Gold
<input type="checkbox"/>	master	No	N/A	No	Unprotected
<input type="checkbox"/>	model	No	--	No	Unprotected
<input type="checkbox"/>	msdb	No	N/A	No	Unprotected

## CROSS VERSION RESTORE

SQL DBA's often need to restore SQL backups to different versions of SQL Server whether for testing or restoring old backups where only newer SQL Server versions remain in the customer environment. Rubrik supports restoring databases to the same or newer SQL Server version. For precise details, please consult the Rubrik Compatibility Matrix for a current list of supported Source and Target SQL Server versions.

## RECOVERY TO A SPECIFIC LSN VIA THE API

Rubrik provides the SQL DBA the ability to recover to a specific Log Sequence Number (LSN) via the Rubrik API for full recovery model databases. This is especially useful when trying to recover to a specific transaction or set of transactions. A LSN recovery is different than a standard Transaction Log recovery as it allows you to recover to a specific state that is anywhere within the Transaction Log instead of the end of the Transaction Log.

```
{
  "recoveryPoint": {
    "lsnPoint": {
      "lsn": "LSN Number",
    }
  }
}
```

## UNDER THE HOOD

### SQL BACKUP FLOW

1. The Rubrik cluster initiates a snapshot based on SLA policy. For security reasons, all backups are initiated by the Rubrik cluster rather than any client or connector.
2. The connector takes a snapshot of the SQL Server database.
3. The Rubrik cluster sends the previous snapshot metadata (if any) to the connector.
4. The connector scans the current snapshot and compares metadata to the previous snapshot's metadata.
5. For specific data blocks where metadata doesn't match, the connector sends the data to Rubrik.
6. After all the database files are scanned, the connector deletes the snapshot.

### SQL RESTORE FLOW

1. Rubrik finds the snapshot and log backups needed to recover to a certain point in time as specified by the user.
2. The connector verifies file permissions via a temporary database.
3. Rubrik transfers row files (mdf) and log files (ldf) to their destinations.
4. The connector uses a native SQL restore method to restore the database.
5. Rubrik transfers log backups to a temporary directory.
6. The connector issues SQL commands to replay log backups to the requested point in time.
7. After replay of logs is complete, the connector deletes log backups from the temporary directory.
8. The connector opens the database to be externally accessible. Restore/Export is done.

For further details, please see the "SQL Server Databases" chapter of the Rubrik User Guide or reach out to your Rubrik sales team.

## CONCLUSION

While robust and full-featured, Rubrik's support for SQL backups extends the Rubrik focus on simplicity via understanding customers' true operational requirements - who says backups can't be fun?