TECHNICAL WHITE PAPER

How It Works: Cloud-Native Protection for Microsoft Azure SQL with Rubrik Security Cloud

Brian Mislavsky & Chris Lumnah

RWP-0618

Table of Contents

INTRODUCTION

Welcome to *How It Works: Cloud-Native Protection for Microsoft Azure SQL*. This guide introduces the features, architecture, and workflows for protecting Azure SQL databases with Rubrik Security Cloud (RSC). This information helps you evaluate, design, or implement the technologies discussed here.

AUDIENCE

We designed this guide for architects, DBAs, and engineers who need a deep dive into Rubrik's Azure SQL protection. If you're responsible for data security, compliance, or governance, you'll find the technical underpinnings you need right here.

OBJECTIVES

This guide provides the technical underpinnings of Rubrik Security Cloud (RSC) for Microsoft Azure SQL. By the end of this reference, you will understand:

- How RSC solves the persistence and frequency gaps in native Azure backups.
- The architectural trust relationship between RSC and your Azure tenant.
- How immutable snapshots provide a "clean room" for ransomware recovery.

AZURE SQL AUTOMATED BACKUPS

Microsoft's Azure SQL Database and Azure SQL Managed Instance include a basic backup offering encompassing two fundamental types of protection: Point-in-Time Retention (PiTR) backups and Long-term Retention (LTR) backups. Both options are available at different levels of redundancy selected by the customer to meet business needs, and each has a unique set of features and limitations.

This section provides a brief overview of core concepts related to automated backups in Azure SQL. For additional information, refer to the Microsoft Azure SQL documentation.

PITR BACKUPS

PiTR backups are automatically taken & stored on the same Server / Managed Instance based on the service tier and according to the following schedule, and are used to restore a database to a point in time within the configured retention period:

	Frequency	Max Retention
Full Backups	Weekly	7 Days Basic | 35 Days Other
Differential Backups	12 or 24 hours	7 Days Basic | 35 Days Other
Transaction Log Backups	Approx. every 10 minutes	7 Days Basic | 35 Days Other

For Azure SQL Database, databases are restored by creating a new database instance on the same server as the original database.

For Azure SQL Managed Instances, databases are restored by creating a new database on a target instance, whether it's the same as the source or a different one.

LONG-TERM RETENTION BACKUPS

LTR backups leverage the full backups taken for PiTR and can be stored in Azure Blob Storage for up to 10 years in a Microsoft Azure-managed storage account. Once LTR backups are configured, full backups are copied to the storage account weekly, monthly, or yearly, based on the policy.[1]

CHALLENGES

Cloud Native Protection limitations

Digital enterprises increasingly use multiple clouds, private and public, to deploy applications, mitigate vendor lock-in, and use best-of-breed solutions. However, this fragments data across hybrid and multi-cloud infrastructures, fracturing IT's ability to protect, manage, and secure data and operations.

Public cloud providers handle the cloud's protection and availability. But customers are ultimately responsible for protecting their resources **within** the cloud, their applications and data, regardless of the provider. Microsoft's Shared Responsibility Model clarifies these concepts.

This leaves customers with a critical question: How do I efficiently and reliably protect my assets in the cloud? While that question seems simple, choosing the right solution is challenging.

Customers may try to lift-and-shift legacy tools into hybrid or multi-cloud environments. Unfortunately, this approach often hinders the agility and elasticity enterprises seek in a cloud strategy.

The alternative, using platform-native tools, can be similarly flawed. It segments data protection operations between public cloud providers and public / on-premises environments. Moreover, this creates significant headwinds for compliance, visibility, and operational efficiency.

Azure SQL Backup & Recovery Considerations

While Azure SQL offers solid built-in protection, you must account for specific platform behaviors when designing your strategy:

Backup Persistence: PiTR backups are tied to the logical server. If you delete the server, Azure purges the backups. To mitigate this risk, enable Long-Term Retention (LTR); these backups live independently and survive server deletion.

Frequency & Retention: Azure manages the backup cadence. While you can choose 12- or 24-hour differential intervals, the platform controls the full backup schedule. PITR retention caps at 35 days for most tiers, so use LTR for anything longer than that. Azure SQL has a maximum retention of 10 years.

On-Demand Capabilities: Azure SQL Database doesn't offer "one-click" portal backups to standard SQL files. You'll have to use the "Export" function to create .bacpac files, which can strain resources on larger databases.

1 https://learn.microsoft.com / en-us / azure / azure-sql / datcloud's protection and availabilityabase / long-term-retention-overview?view=azuresql

Recovery Scopes & Subscription Boundaries
While Microsoft has introduced support for Cross-Subscription Restore, specific recovery boundaries remain:

- **Azure SQL Database:** PiTR backups are generally restored to a new database on the original server or a different server within the same region. Cross-region recovery is available via Geo-Restore but involves different performance profiles.
- **Azure SQL Managed Instance:** Restores must be performed to an instance with compatible hardware configurations and software generations.

Geo-Restore & Redundancy Constraints
The default storage redundancy for backups is typically Geo-Redundant Storage (GRS). While cost-effective, Geo-Restore is a "best-effort" recovery mechanism intended for regional disasters. It carries a Recovery Point Objective (RPO) of up to 1 hour and a Recovery Time Objective (RTO) of up to 12 hours. Additionally, during a widespread regional outage, the failover region may experience high resource demand, slowing recovery. For mission-critical applications requiring near-instant recovery, Failover Groups or Active Geo-Replication should be implemented to supplement standard backups.

THE RUBRIK APPROACH

MULTI-CLOUD PROTECTION

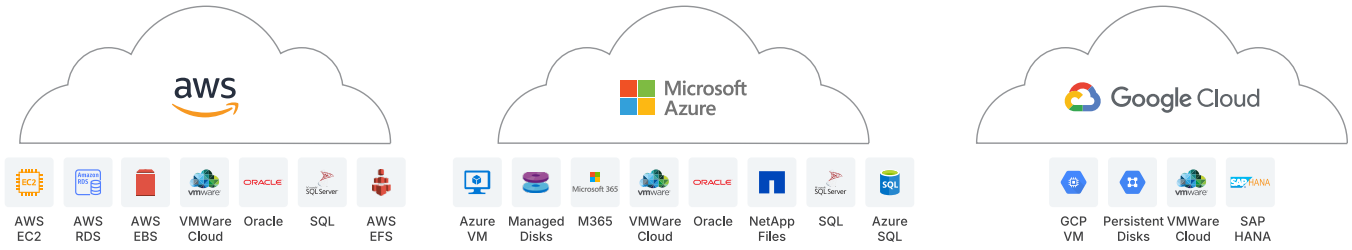

Figure 1 – Rubrik Security Cloud Multi-cloud Protection

Rubrik's goals are to simplify and automate data protection and security against events such as accidental data deletion and ransomware attacks, using policy-based protection and frictionless operations. Rubrik Security Cloud is a Software-as-a-service (SaaS) data protection platform that provides automated backup, recovery, and replication schedules across regions and clouds, all managed by a single global policy engine. This solution enables Rubrik customers to harness the benefits of rapid innovation and simplified management complexity, with data security delivered as a service.

Protecting Azure SQL workloads with Rubrik Security Cloud involves three steps.

Step	Detail
Authorization	Authorize Rubrik Security Cloud to access the Azure Subscription(s) that require protection via an OAuth integrated workflow that aligns with Azure security best practices.
Configuration	Use a single, declarative SLA policy engine to automatically create Azure SQL DB and Managed Instance database snapshots to suit backup and retention requirements.
Protection	Recover and export databases rapidly through Rubrik Security Cloud's SaaS UI. Security Cloud acts as a single pane of glass for hybrid and multi-cloud deployments.

ENHANCING AZURE SQL PROTECTION WITH RUBRIK IMMUTABLE BACKUPS

Rubrik Security Cloud (RSC) builds on Azure SQL's native backups to provide more flexibility and recovery options. It allows you to overcome platform limitations across various Azure Tenants, Subscriptions, and Regions.

Key advantages include:

- **Unified Management:** Manage data across regions and tenants from a single point without needing persistent compute resources in your environment.
- **Automated Protection:** Use global SLA Domains — a declarative, policy-driven framework — to automate protection at the subscription or resource group level.
- **Persistent Backups:** Unlike native PiTR backups, which vanish if a server is deleted, Rubrik's backups persist in your storage account or Rubrik Cloud Vault (RCV).
- **Granular Frequency:** Choose backup intervals more frequent than Azure's standard schedules.

ARCHITECTURE

HIGH-LEVEL ARCHITECTURE

Rubrik Security Cloud (RSC) uses a policy-driven engine to protect Azure SQL Databases and Managed Instances within your subscriptions. You can choose between two primary protection modes based on your recovery requirements: **Azure SQL Automated Backup Orchestration** or **Rubrik Immutable Backups**.

Security & Authentication

Regardless of the mode you choose, RSC uses a secure, unified authentication framework:

- **Service Principal:** RSC creates a service principal when you enable Cloud-Native Protection for a subscription.
- **Least Privilege:** This principle uses a custom role with only the minimum permissions needed to protect and restore your databases.
- **Secure Storage:** We store the application credentials in an encrypted format within a dedicated, customer-specific RSC database.

Mode 1: Orchestrated Azure Backups

This mode manages the native backup functionality already built into Azure.

1. **SLA Assignment:** You assign Rubrik SLAs to your discovered databases.
2. **Synchronization:** RSC uses Azure Resource Manager (ARM) APIs to sync native backup policies with your Rubrik SLAs.
3. **Recovery:** RSC triggers and monitors database recovery through these same native APIs.

Mode 2: Rubrik Backups Orchestration

This mode provides persistent, decoupled protection that survives the deletion of the source server.

1. **Trigger:** When an SLA triggers a backup, RSC requests ephemeral resources from Rubrik Exocompute in your subscription.
2. **Extraction:**
    - **Azure SQL DB:** RSC creates a transactionally consistent copy, then uses SQLPackage and BCP to extract the schema (DACPAC) and data to a temporary disk.
    - **Managed Instance:** RSC uses the native T-SQL `BACKUP DATABASE` command to stream backups directly to Azure Blob storage URLs in `.bak` format.
3. **Persistence:** RSC securely stores snapshot metadata and terminates the Exocompute resources once the task is complete to save costs.

ONBOARDING

As stated in this document, protecting Microsoft Azure SQL DBs and Managed Instances consists of 3 steps: Authorize, Configure, and Protect. Customers should refer to the Rubrik Security Cloud User Guide for specific instructions on configuring Rubrik Security Cloud to protect Azure SQL workloads. After reading this document, the reader should have a clear understanding of how Microsoft Azure SQL protection with Rubrik Security Cloud is architected, configured, and utilized.

AUTHORIZATION

Authorizing Rubrik Security Cloud to protect Microsoft Azure SQL DBs and Managed Instances is a straightforward process:

1. From the **Azure** section of the **Cloud Accounts** settings page, click **Add Azure Subscription** to launch the configuration wizard

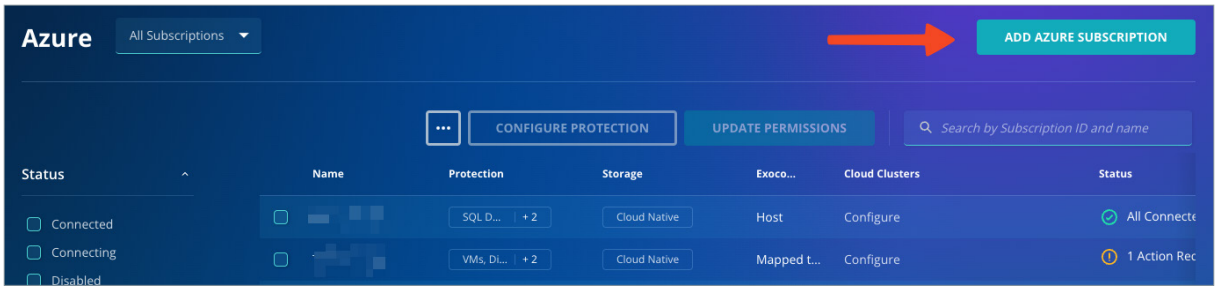

Figure 2 – Adding a Microsoft Azure Subscription – Launch Add Cloud Account Wizard

2. After selecting Azure as the Cloud Provider, select Azure SQL databases and / or Azure SQL managed instances as the use case.

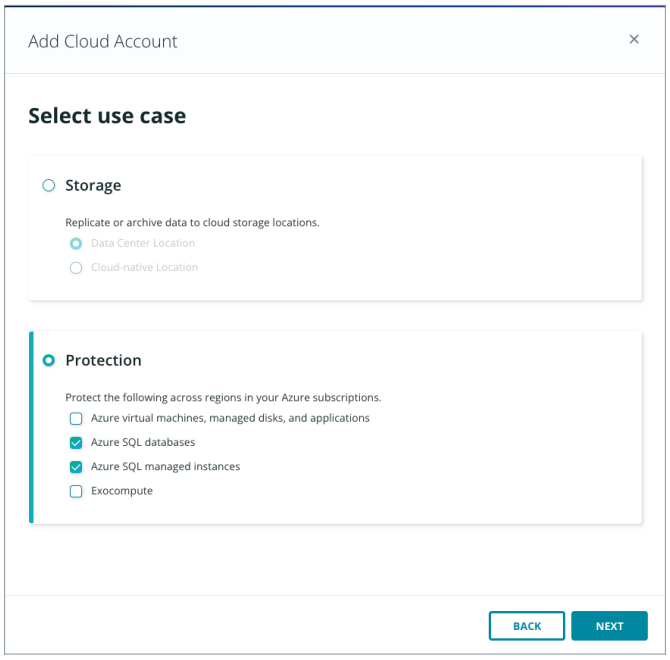

Figure 3 – Adding a Microsoft Azure Subscription – Select Azure SQL Use Case

The wizard then guides the customer in logging in to the specified Microsoft Entra ID with a user who can read, create, and update application registrations, roles, and role assignments. Once authenticated, the user selects the appropriate Subscription(s) and region(s) in scope for protection, a resource group, and then clicks Submit. The process will then create roles in the chosen Subscriptions and assign these roles to a Service Principal created by Rubrik Security Cloud. Additionally, it will ask the customer to configure Exocompute if required (this can be configured later and is detailed in the next section).

To protect Microsoft Azure SQL, Rubrik Security Cloud requires interacting with the customer's Azure subscription(s). Rubrik Security Cloud leverages the Microsoft Azure SQL Database and Managed Instance APIs, whose access is controlled by Microsoft Entra ID.

Microsoft Entra ID (formerly Azure Active Directory) is quite powerful and supports a variety of identities, including users, groups, federated users and groups, and service principals. Permissions are delegated to or revoked from these identities through roles, which define the actions a specific identity can and cannot perform. The scope at which the role is assigned determines which resources the identity can access. Common scopes for role assignment include subscriptions and Resource Groups.

Rubrik Security Cloud leverages a service principal and a custom role (with minimum required permissions) assigned to the subscription(s) the customer chooses to protect. These objects and trusts are created when the customer's subscription(s) are added to Rubrik Security Cloud and are assigned only the permissions necessary to protect the customer's Azure SQL Databases and Managed Instances. The user credentials enabling protection are only used when initially adding subscriptions to Rubrik Security Cloud.

During this process, a user with Global Admin permissions must create the required service principals in the customer's Entra ID tenant. All subsequent operations (backup and restore) utilize the enterprise app registration and custom roles created.

From an Entra ID perspective, the figure below depicts how the workflow interacts with a customer's Azure subscription(s).

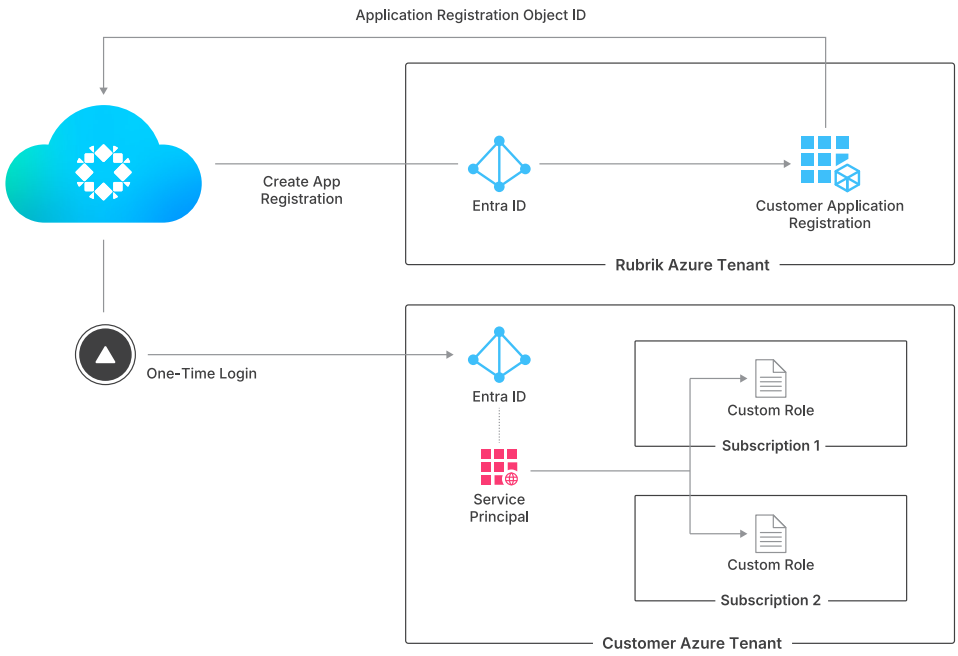

Figure 4 – Rubrik Security Cloud Application Registration Workflow

In Microsoft Entra ID, there are two representations of applications: application objects, also known as Application Registrations, and Service Principals, also known as Enterprise Application Registrations. Application objects describe an application to Microsoft Entra ID. They can be considered the application's definition, allowing Azure to issue tokens to the application based on the app registration's settings. This application object exists only in its home tenant, even though it is a multi-tenant application that supports Service Principals in other directories. Service Principals govern an application that connects to Microsoft Entra ID, which can be considered an instance of the application within the customer's directory. Any given application can have at most one application object and one or more Service Principals representing instances of the application in every directory in which it acts.

As depicted in Figure 4, a customer-specific application object is created in a Rubrik-owned and managed Azure Tenant. The customer-specific application accesses the customer's subscriptions by utilizing corresponding service principals in the customer's directories. These service principals are created when customers add their subscriptions to Rubrik Security Cloud for protection. The permissions delegated to this service principal are controlled by the custom roles assigned to it in each subscription. This architecture establishes a trust relationship between the customer's Microsoft Entra ID and Rubrik's, enabling Rubrik Security Cloud to interact with Azure SQL APIs after authenticating with Rubrik's Entra ID. A significant benefit of this approach is that it does not require the customer to share long-lived Azure credentials with Rubrik when enabling protection for Azure resources with Rubrik Security Cloud.

Once this process is complete, Rubrik confirms that the necessary permissions are in place and has all the required information to begin protecting the Azure subscription(s). Examples of the custom roles created during this process are available for reference on GitHub.

Another benefit of this method is that if these permissions need to be modified in the future, Rubrik Security Cloud can prompt the user to update the role via OAuth. The user initiates the workflow in Rubrik Security Cloud, logs in, and authorizes the role changes when prompted.

Alternative methods are also available to add Microsoft Azure subscriptions to Rubrik Security Cloud. If one of these approaches is necessary in your environment, please contact Rubrik Support for enablement. These include:

- Addition of a subscription without leveraging OAuth and a cross-tenant app registration
- Manually entering the subscription details when adding an Azure subscription
- Programmatically creating and adding subscriptions to Rubrik Security Cloud

CONFIGURATION

Detailed configuration steps for protecting Microsoft Azure SQL databases with Rubrik Security Cloud are available here.

As mentioned previously, Rubrik provides two modes of protection: orchestration of Azure SQL's basic automated backups and RSC Backup Orchestration for Azure SQL. The following table is included in the above-mentioned product documentation and is included here because familiarity with the different modes is key to understanding how Rubrik Security Cloud protects Azure SQL workloads.

Backup Type	Description	Recovery
Immutable Backups	When Immutable backups are enabled, RSC performs and manages short- and long-term retention backups according to the SLA Domain configuration.	When immutable backups are enabled, RSC supports exporting databases to a SQL Server in any location. Recovery can span different subscriptions or a cloud or on-premises setup, independent of Azure.
Long-term retention (LTR) backups	When Immutable backups are configured, Rubrik performs LTR backups instead of Azure SQL native backups. Otherwise, RSC supports the management of backups taken natively by Azure based on the non-daily frequencies defined in the SLA Domain configuration. Azure Native LTR backups have a retention period of up to 10 years.	When immutable backups are not configured, RSC supports exporting databases from LTR backups across different Azure regions within the same subscription as the source database server or managed instance.
Point-in-time (PiTR) backups	PITR retention backups are kept for up to 35 days, depending on the Azure SQL workload's service tier.	RSC supports point-in-time restore (PiTR), which creates a new database from backups taken at any point in the specified retention period. PiTR is limited to recovery within the Azure region of the source database or managed instance.

Orchestrated Azure Backups

For customers who want to leverage the included basic automated backup functionality in Azure SQL, Rubrik Security Cloud uses Microsoft Azure APIs for Azure SQL DB and Azure SQL Managed Instances to configure automated backup settings for the protected databases. The customer requires no additional configuration at this point, and an SLA can be configured according to the limits defined by the Azure SQL purchasing and deployment model in use.

PITR & LTR backups taken via Rubrik Security Cloud using native Azure SQL orchestration will be subject to the limitations outlined in a previous section of this document.

RSC Backup Orchestration

Adding an Azure Subscription is required for both Azure SQL Database and Managed Instances. The wizard will walk you through the process and explain what happens. While there are similarities, we will call out the differences below.

Once Immutable Backups are configured for an Azure Subscription, PiTR backups rely on Azure SQL's native backups. In contrast, LTR and On-Demand backups are now protected using RSC Backup Orchestration

Azure SQL Database

The Azure Subscription onboarding wizard will prompt you to select a resource group. When performing Azure SQL DB copy backups, we create a new Rubrik-managed SQL server and temporary database copies within the customer's environment. These resources will be placed in the resource group specified by the customer as part of their Azure SQL DB cloud account mapping.

To effectively manage these resources, Rubrik requires elevated permissions, including delete permissions. Having these resources in a dedicated Rubrik-managed resource group allows Rubrik to inherit the higher permissions at the resource group level, which the customer has explicitly authorized. This approach enables Rubrik to perform backup tasks using only minimal permissions at the subscription level. To understand the prerequisites for protecting Azure SQL Database, see Prerequisites for Azure SQL DB Rubrik backup.

Elastic Pool Support
Rubrik Security Cloud discovers and protects databases within Azure SQL elastic pools. Elastic pools provide a cost-effective solution for managing multiple databases with varying and unpredictable resource demands.

When Azure SQL databases are part of an elastic pool, Rubrik Security Cloud:

- Automatically discovers the elastic pool membership during the subscription discovery process
- Displays the elastic pool name as metadata for each member database
- Supports all backup and recovery operations (PiTR, LTR, and Immutable Backups) for databases within elastic pools
- Allows SLA assignment at the individual database level, regardless of pool membership

Networking Automation
In addition to the default approach of allowing all Azure services to connect to the Rubrik-managed SQL server (via IP whitelisting 0.0.0.0), Rubrik Security Cloud can now automate the configuration of more secure networking options:

- **Service Endpoint Automation:** Rubrik can automatically configure VNet service endpoints for the Rubrik-managed SQL server, restricting network access to traffic originating from specified virtual network subnets.
- **Private Endpoint Automation:** Rubrik can automate the creation of Azure Private Endpoints for the Rubrik-managed SQL server, providing private connectivity from the Exocompute VNet to the SQL server over the Microsoft backbone network. This eliminates public internet exposure for backup traffic.

> **Note:** Configuring private endpoints may increase data transfer costs compared to service endpoints or public access.

These networking options provide customers with enhanced security postures for their backup infrastructure, aligning with zero-trust network architectures.

Azure SQL Managed Instance
For customers to leverage Rubrik's Immutable Backups for Microsoft Azure SQL Managed Instances, additional configuration is required after connecting Rubrik Security Cloud to their Microsoft Azure Subscription(s). While some steps can be completed manually, Rubrik Security Cloud includes a wizard that coordinates the entire process.

> **NOTE:** The wizard must be run for each subscription, depending on the Azure SQL type in use: Database or Managed Instance. For example, if a subscription has Azure SQL Databases and Managed Instances, the wizard will need to be run twice — once for Azure SQL DB protection and again for Managed Instances.

Encryption: TDE with Customer-Managed Keys (CMK)
Rubrik Security Cloud supports protecting Azure SQL databases that use Transparent Data Encryption (TDE) with customer-managed keys (CMK). When this capability is enabled, Rubrik can perform database copy operations and backups on CMK-encrypted databases, preserving the encryption configuration throughout the backup and recovery lifecycle.

This ensures that organizations using CMK-based TDE for regulatory compliance can leverage Rubrik immutable backups without modifying their encryption strategy.

PROTECTION

Protection is prioritized according to the level at which an SLA is assigned. The protection hierarchy is listed below. In both cases, the lower level takes precedence over the higher level.

- Azure SQL DB
  - Subscription
  - Resource Groups
  - Servers
  - Tag Rules
  - Databases
- Azure SQL MI
  - Subscription
  - Resource Groups
  - Tag Rules
  - Managed Instances
  - Databases

Rubrik Security Cloud batches all snapshot jobs to prevent overrunning Azure API limits with a single large batch of snapshot or replication activities. By default, Rubrik Security Cloud runs up to 20 protection jobs in parallel for Azure SQL Database. For Azure SQL Managed Instance, backup concurrency is limited to 1 job per logical server at a time to ensure optimal performance and resource utilization.

PROTECTION USING AZURE SQL AUTOMATED BACKUP ORCHESTRATION

Once subscriptions are added and SLA Domains are assigned, Rubrik Security Cloud will begin protecting workloads in Azure. The Rubrik Security Cloud job framework will automatically begin scheduling and snapshotting Azure SQL Databases and Managed Instances based on the SLA Domains you create and assign, eliminating the need to schedule jobs manually.

PiTR & LTR backups

When not using Immutable Backups, customers configure SLAs in Rubrik Security Cloud that are then used to enforce corresponding policies in Microsoft Azure.

Reminder: Because this protection method leverages native Microsoft Azure SQL backups, backup policies must adhere to all applicable guidelines. If a different retention period, recovery location, or backup frequency is desired, customers should consider protecting Azure SQL using Rubrik's Immutable Backups.

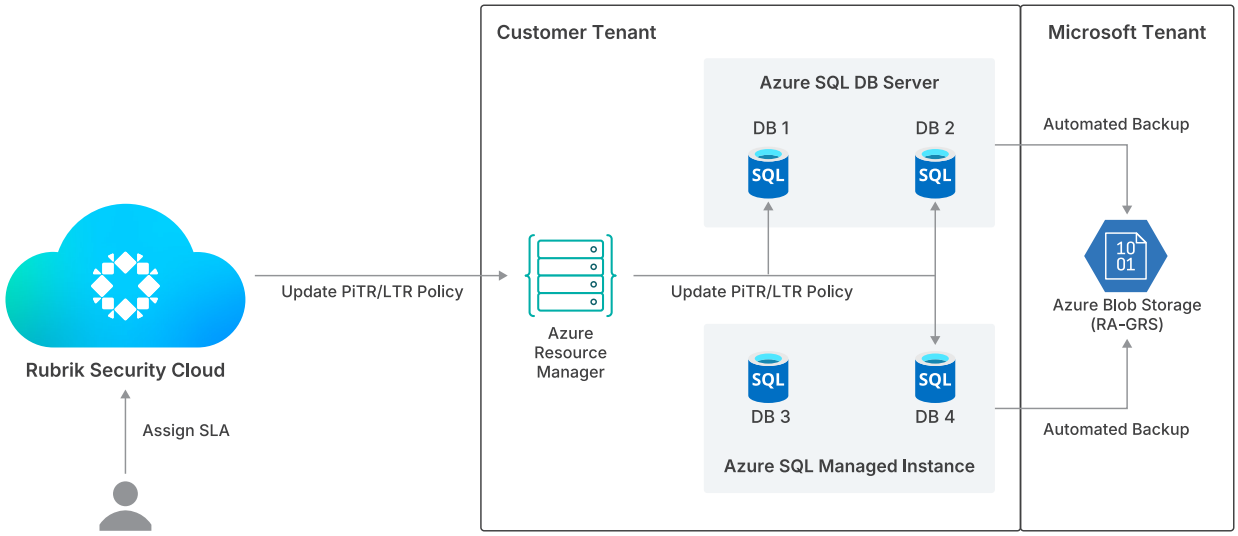

Figure 5 – Protection Using Azure SQL Automated Backup Orchestration

When a user assigns an SLA corresponding to Short-Term Retention to an Azure SQL object, RSC uses the Azure Resource Manager APIs to set the corresponding Backup Policies.

PiTR & LTR restores

Restoring a database from either a short- or long-term backup with RSC follows a process similar to configuring backups.

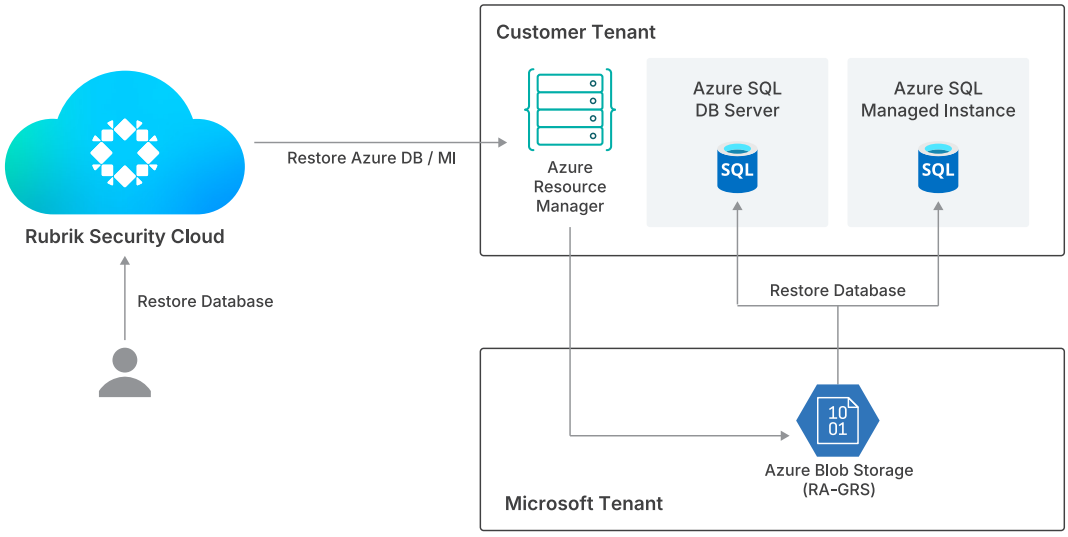

Figure 6 – Azure SQL recovery without Immutable Backups

When not using Azure SQL Automated Backup Orchestration in RSC, customers leverage RSC to view taken backups and initiate a recovery from them. Behind the scenes, while RSC leverages the native Azure Resource Manager APIs for Azure SQL to perform restores, there is a minor difference in how PiTR & LTR restores are performed.

When restoring from a PiTR backup, RSC retrieves all pertinent information about the available backups directly from Microsoft Azure. This is primarily to reduce the number of API queries to Azure that RSC would ultimately need if it kept metadata about PiTR backups. Since the result of the `earliestRestoreDate` API query (see below for a list of APIs used) is relatively static for a given day, there is no additional benefit to keeping this data in RSC. The PiTR database recovery workflow is depicted in Figure 8 below.

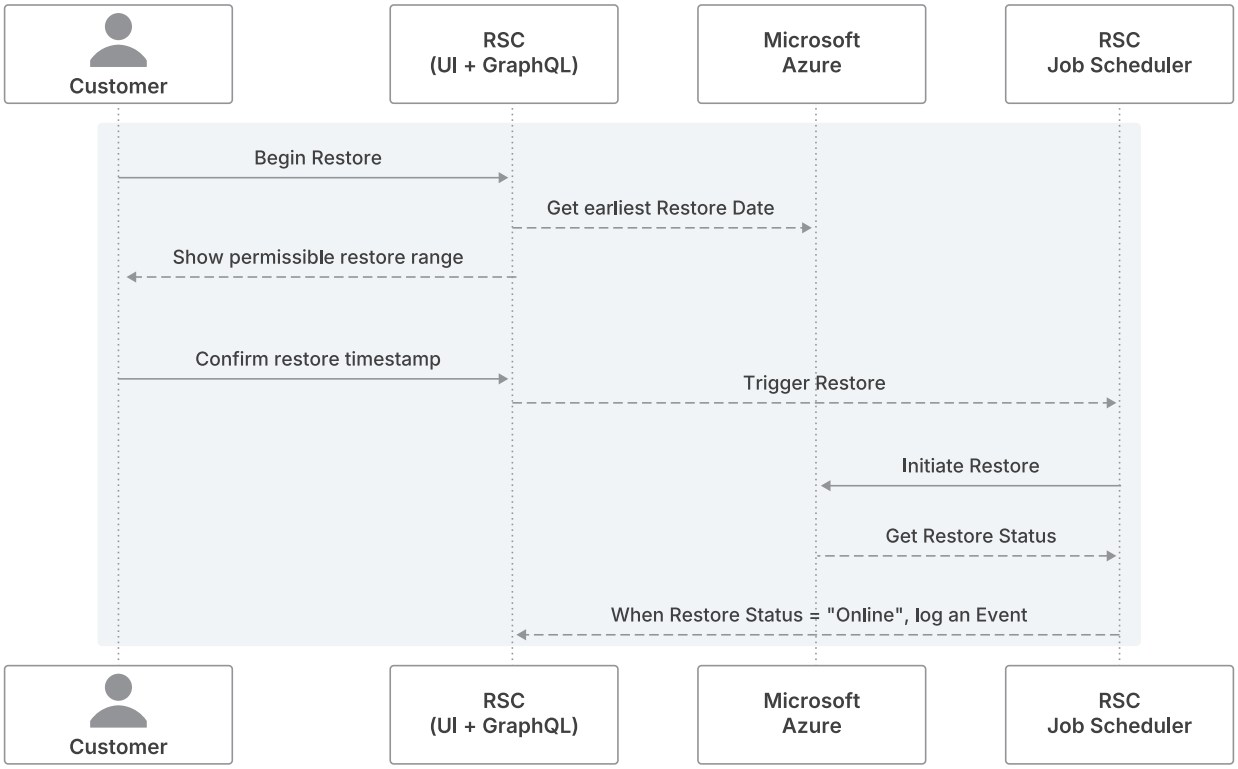

Figure 7 – Azure SQL database recovery flow - PiTR restore

Unlike PiTR backups, Rubrik Security Cloud stores metadata about LTR backups in its database and regularly pulls the list of available backups from Azure to stay in sync. As the number of LTR backups increases over time and the required API does not support pagination, RSC querying Azure directly for the list of LTR backups, as it does with PiTR backups, could increase UI latency for users and degrade the overall experience.

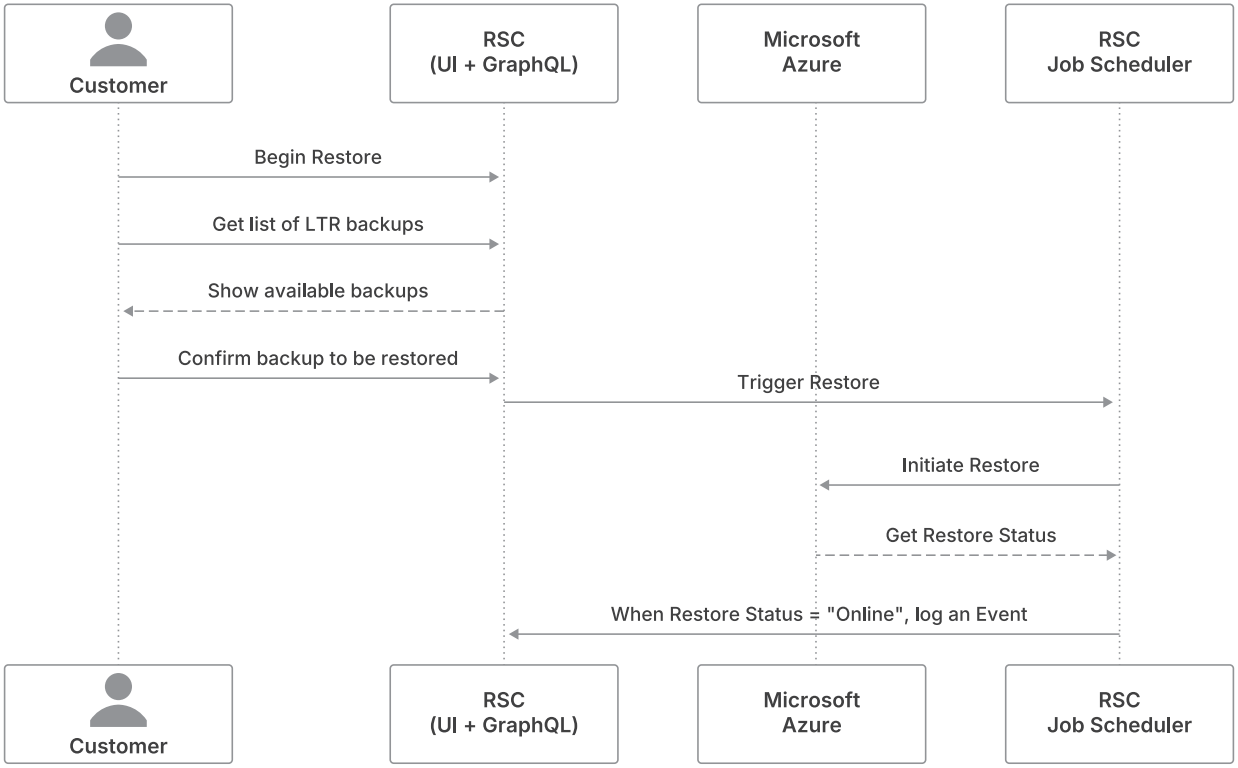

Figure 8 – Azure SQL recovery flow - Long-term retention recovery

Azure SQL Database APIs used - PiTR Recovery

Azure SQL DB	Azure SQL Managed Instance
Create a Database	Create Managed Instance Database
Get Restore status	Get Restore status
Get earliestRestoreDate	Get earliestRestorePoint

Azure SQL Database APIs used - LTR Recovery

Azure SQL DB	Azure SQL Managed Instance
LTR Backups–List by Database	LTR MI Backups–List by Database
Databases–Create or Update (with `CreateMode` set to `RestoreLongTermRetentionBackup`)	Managed Databases–Create or Update (with `ManagedDatabaseCreateMode` set to `RestoreLongTermRetentionBackup`)
LTR Backups–Copy	Managed Database Restore Details–Get

PROTECTING AZURE SQL USING RSC BACKUP ORCHESTRATION

In addition to orchestrating the native automated protection of Azure SQL, Rubrik has introduced the concept of Immutable Backups for Azure SQL protection. This functionality places Rubrik Security Cloud in the data path to protect Azure SQL and allows customers to address the limitations listed in the previous sections.

Components

To provide persistence for Azure SQL backups, Rubrik leverages additional features / technology outlined in this section.

Rubrik Exocompute

Rubrik Exocompute is an ephemeral container-based framework that Rubrik leverages to process data. For Microsoft Azure, Rubrik leverages Azure Kubernetes Service for compute resources. Exocompute nodes are deployed in customer-owned Azure Subscriptions and are created and destroyed as needed to minimize costs. Exocompute can be deployed to each customer subscription or a centralized subscription, which can be leveraged across multiple subscriptions in the same region, further decreasing costs and complexity.

Azure Database Copy

For Azure SQL Databases, Rubrik uses the Microsoft Azure Database Copy process to copy the live database to a separate Azure SQL Server instance in a Rubrik-managed resource group defined when adding the subscription to RSC. When performing Azure SQL DB Copy backups, we create a new Rubrik-managed SQL server and temporary database copies within the customer's environment. The Rubrik-managed SQL Server will have a name like "**rbrk-<sub_id>-<random_char>**" (i.e., **rbrk-x524a5c2-g8f8-4h8d-9dk7-9239nbdc156d-9k1qiECN2M**). These resources will be placed in the customer-specified resource group during onboarding for their Azure SQL Database cloud account.

To effectively manage these resources, Rubrik requires elevated permissions, including delete permissions. Having these resources in a dedicated Rubrik-managed resource group allows Rubrik to inherit these higher permissions at the resource group level, permissions that the customer explicitly authorizes. This approach enables Rubrik to perform backup tasks using only minimal permissions at the subscription level.

Additionally, Rubrik requires the following permissions for configuring the firewall settings on the Rubrik-managed SQL server:

- `Microsoft.Sql/servers/firewallRules/read`
- `Microsoft.Sql/servers/firewallRules/write`

These permissions are set at the resource group level.

By default, Rubrik allows all Azure services to connect to the Rubrik-managed SQL server (0.0.0.0). However, when service endpoint or private endpoint automation is enabled, Rubrik can configure more restrictive network access policies. See the Networking Automation section above for details.

We are not setting up a service or private endpoint on the SQL server by default. These Rubrik-managed SQL servers are established during the first backup job for each subscription and region.

If the customer wishes to configure a service or private endpoint, they can do so after the SQL server is created. Please note that configuring the private endpoint for the SQL server will increase data transfer costs.

BCP (Bulk Copy Program) for SQL Database
For Azure SQL Databases, Rubrik supports an alternative backup format using the BCP (Bulk Copy Program) utility. BCP separates schema and data extraction, enabling optimized parallel data handling for large databases.

Key characteristics of BCP backups:

- **Schema extraction:** Database schema is extracted as a DACPAC file using SqlPackage's extract mode, capturing all schema objects (tables, views, stored procedures, etc)
- **Data extraction:** Table data is exported using the BCP utility with row delimiters, enabling efficient bulk data transfer
- **Temporal table support:** Handles temporal tables with hidden columns, ensuring system-time period columns are correctly captured and restored
- **Cloud Storage Layer:** BCP data is ingested into Rubrik's Cloud Storage Layer (CSL), which provides deduplication and efficient data packing
- **TABLOCK hint control:** The TABLOCK hint during BCP operations is configurable via application-specific tuning (AST), allowing optimization for concurrent workload scenarios

BAK (Native SQL Server Backup) for Managed Instances
For Azure SQL Managed Instances, Rubrik supports native SQL Server backups using the T-SQL BACKUP DATABASE command. This approach writes backup data directly to Azure Blob storage URLs, bypassing the need for CDC and the BACPAC export process.

Key characteristics of BAK backups:

- **Native SQL Server format:** Uses the standard BACKUP DATABASE ... TO URL command with the COPY_ONLY flag to produce a backup without affecting the database's backup chain.
- **Multi-file striping:** Supports writing to multiple blob URLs simultaneously (up to 64 stripe files), enabling parallel I / O and faster backup throughput.
- **Compression:** Automatically selects the optimal compression algorithm based on SQL Server version:
- **SQL Server 2025+ (version 17):** Uses ZSTD_LOW compression, which offers approximately 50% CPU utilization compared to MS_XPRESS
- **Older SQL Server versions:** Compression is disabled by default to avoid vCPU saturation
- **Additional compression algorithms are supported:** ZSTD (various levels), MS_XPRESS (various levels), and QAT_DEFLATE
- **Direct blob storage:** BAK files are written directly to Azure Blob storage URLs in the customer's storage account, with the URL format: https://{storageAccount}.blob.{endpoint}/{container}/{snappableID}/{snapshotID}/{prefix}N.bak
- **Credential management:** Rubrik sets up SQL Server credentials for blob storage access using either access keys or AAD (Entra ID) authentication

- **No CDC dependency:** Transactional consistency is inherent to the native BACKUP DATABASE command, eliminating the need for Change Data Capture configuration
- **Immutability:** RSC applies Azure Blob-level immutability locks to all stripe files after the backup completes
- **Concurrency:** Backup concurrency is limited to 1 job at a time when BAK is enabled, ensuring optimal performance

Cloud Storage Layer (CSL)
Rubrik developed a unified snapshot-aware storage layer for Cloud Native workloads. The CSL enables Rubrik Security Cloud to store data ranging from a few bytes to multiple GBs in cloud object storage, while allowing various Rubrik services to use a single API for consumption. It provides deduplication, data packing, and garbage collection, offering significant storage and API cost savings for customers, since data is stored in their accounts. In Microsoft Azure, the CSL is built on Azure Blob Storage and Azure Table Storage.

Snapshot Immutability
Rubrik leverages Azure Blob level immutability to provide an additional layer of security for Persistent Backups. When an Azure Storage Account is created using the Configure Immutable Backups wizard, it is created with versioning & immutability enabled on the storage container via the Set Blob Service Properties & Blob Containers – Create APIs.

If a customer uses an existing storage account in the wizard, RSC will ensure versioning is enabled as part of a periodic job that checks for immutability.

Taking an Immutable Backup
When Rubrik Security Cloud triggers an Immutable Backup to be taken of an Azure SQL Database or Managed Instance, the following high-level steps take place:

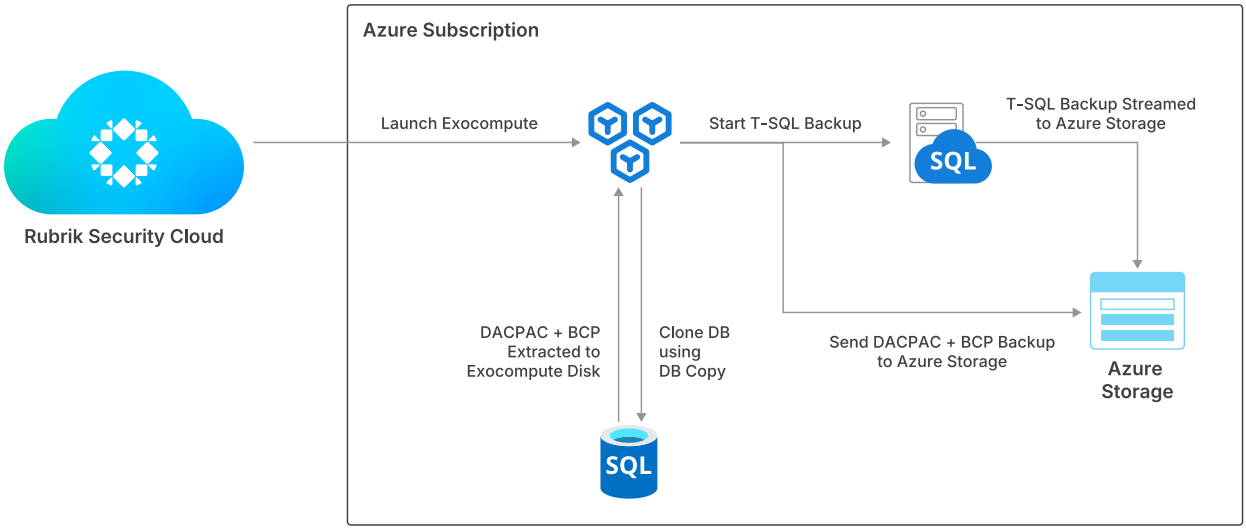

Figure 9 – Immutable Backups – high-level steps

- Backup is triggered based on the defined SLA or on demand, and a request for Rubrik Exocompute resources in the same region as the Azure SQL resource.

- For Azure SQL DB, a database copy is made on the Rubrik-managed SQL Server instance.
    a. Once Exocompute is available, launch a task to extract the database schema via a DACPAC file and the data via BCP processing. Data is written to the empty disk to store the backup while being processed
- For Azure SQL Managed Instances, issue the `BACKUP DATABASE` command to write the backup to the Azure Storage Account.
- Database backup metadata is read and stored in the RSC database.
- Database backup data is ingested into Azure Blob Storage and made immutable according to the SLA.

There are two primary components of this process that we will further explain in detail: the Backup layer & the Ingestion Layer.

BACKUP

When utilizing Azure SQL Backup Orchestration SLA, all backup operations are managed directly by the Azure platform. These processes leverage native Azure platform APIs specifically designed for Azure SQL Database and Managed Instance (MI) backups, ensuring seamless data protection. The following details outline how Rubrik Backup Orchestration integrates with these native capabilities to provide automated management and monitoring of your backup environment.

AZURE SQL DATABASE

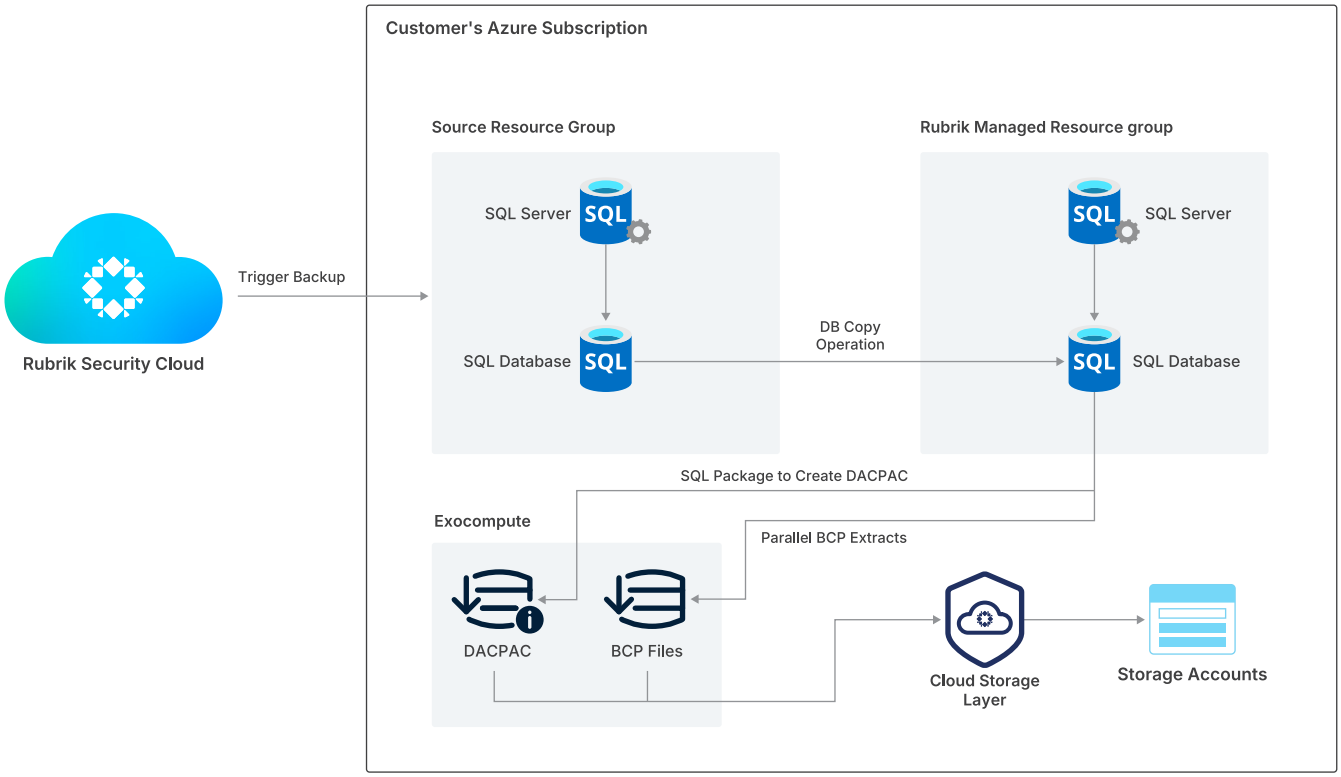

Figure 10 – Azure SQL DB Backup

1. Trigger a database copy to the Rubrik-managed SQL server instance.
2. Schema extraction as a DACPAC file using SQLPackage extract mode.
3. Table data extraction using the BCP utility with row delimiters.
4. The CSL will ingest the DAPCAC and BCP output from the Exocompute. The CSL is used to reduce storage, then moves the "snapshot" to the Storage Account.
5. Make the snapshot immutable in the Storage Account
6. Remove the database from the Rubrik-managed SQL server instance.
7. Metadata is sent to RSC
8. Exocompute is deleted

AZURE SQL MANAGED INSTANCE

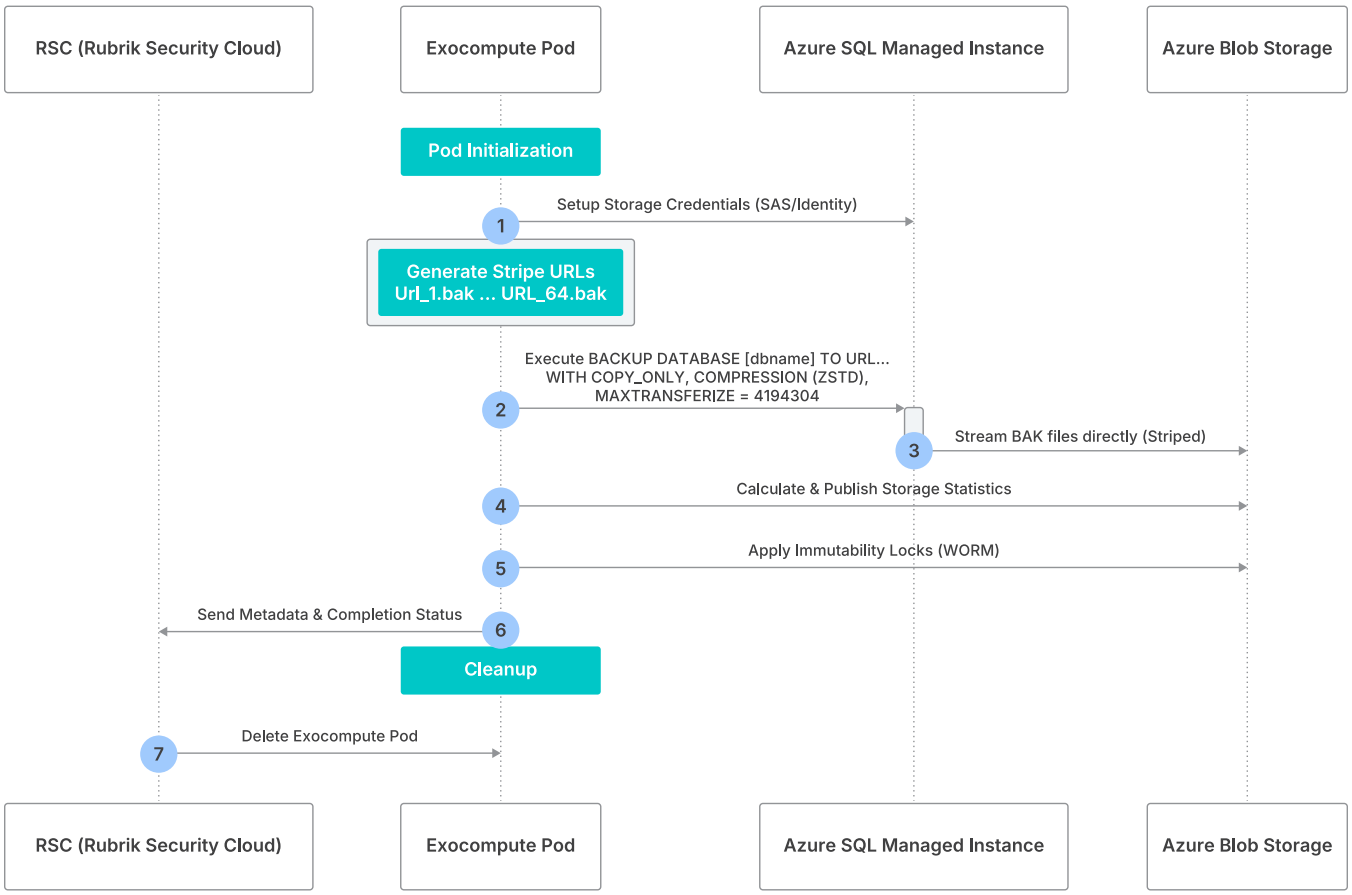

Figure 11 – Azure SQL MI Backup

1. Exocompute pod starts and sets up storage credentials on the Managed Instance.
2. Generate T-SQL Backup Command using striped Azure Blob storage URLs (up to 64 files). Example below:

```
SQL
BACKUP DATABASE [DatabaseName] TO
URL = 'https://account.blob.core.windows.net/container/.../prefix_1.bak',
URL = 'https://account.blob.core.windows.net/container/.../prefix_2.bak'
WITH COPY_ONLY, COMPRESSION (ALGORITHM = ZSTD_LOW, LEVEL = LOW),
MAXTRANSFERSIZE = 4194304, STATS = 5;
```

3. Execute T-SQL Backup Command
4. BAK files are written directly to Azure Blob storage.
5. Calculate and publish storage statistics.
6. Apply immutability locks to all stripe files.
7. Metadata is sent to RSC
8. Exocompute is deleted

RECOVERY

ENTRA ID (AAD) AUTHENTICATION FOR SQL OPERATIONS

In addition to the service principal-based authorization for Azure resource management, Rubrik Security Cloud supports Entra ID (Azure Active Directory) authentication for direct SQL database operations during backup and recovery workflows.

When enabled, the recovery handler factory selects between SQL authentication and Entra ID (AAD) authentication based on the connection details configured for the target database. AAD authentication enables passwordless connectivity using Azure AD service principals, eliminating the need to manage SQL login credentials for recovery operations.

Key capabilities of AAD authentication for recovery:

- **Passwordless recovery:** Uses Azure AD tokens instead of SQL Server credentials for database import operations
- **AAD principal repair:** After importing a database backup, the recovery handler automatically identifies and repairs AAD login-based users. Invalid AAD users are converted to contained users or disabled to ensure the recovered database functions correctly.
- **Temporary admin management:** For SQL Database, a temporary admin with the dbmanager role is created; for Managed Instances, a sysadmin role is used. These temporary credentials are used only during the import process and are cleaned up afterward.

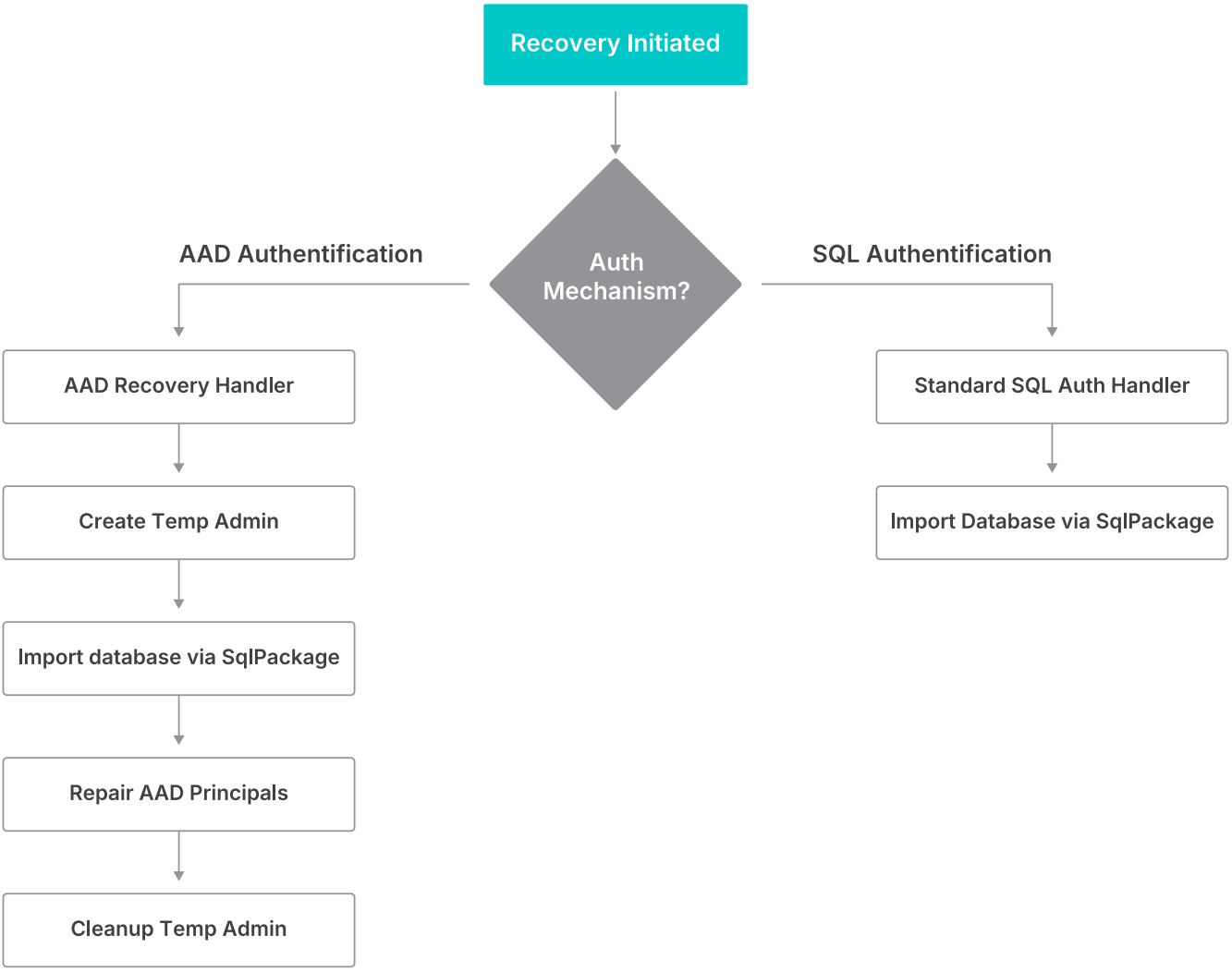

Figure 12 – Entra ID Authentication

AZURE SQL DATABASE

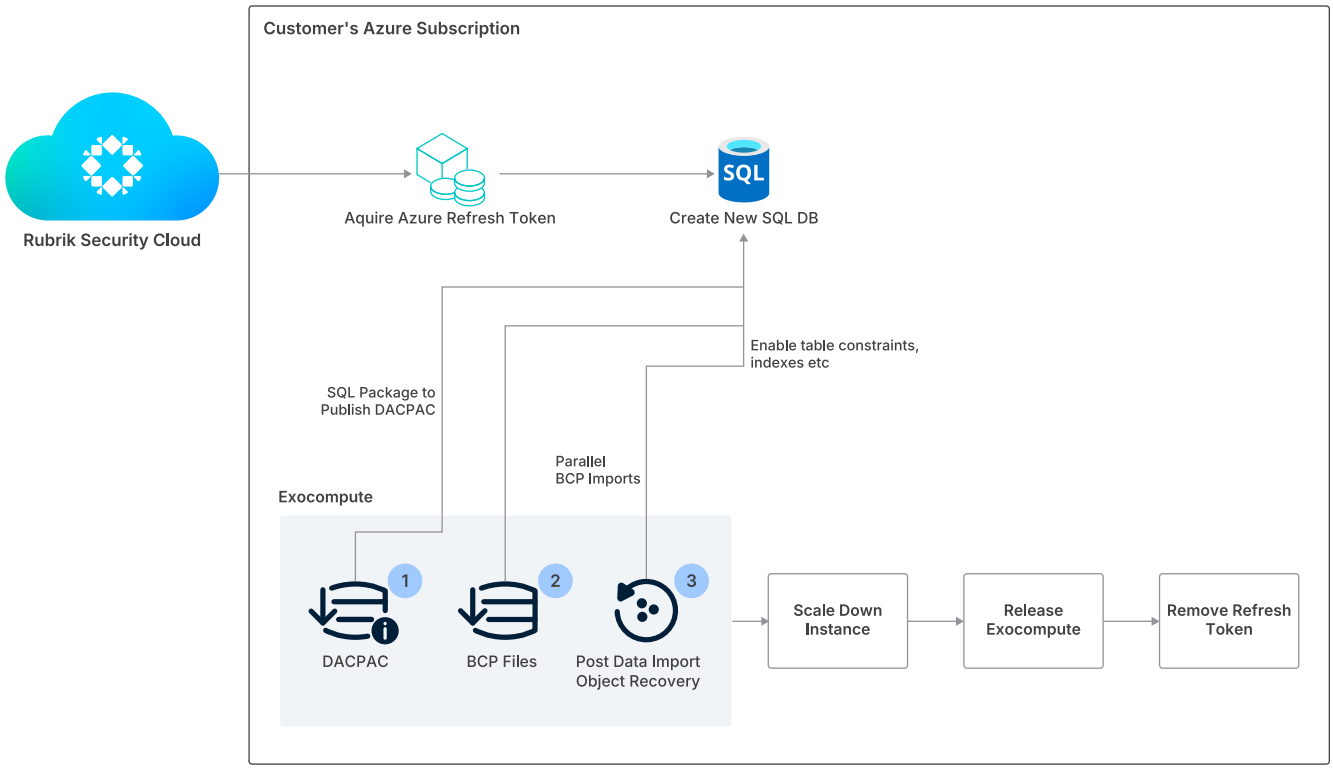

Figure 13 – Azure SQL DB Immutable Backup Restore

1. User initiates a restore from RSC
2. RSC will acquire a reference to the Azure refresh token for the destination subscription with a 7-day TTL.
3. RSC will create a new database based on SQL DB-specific logic
    a. For Native (PiTR / LTR) exports:
        i. Calls Azure ARM API with the appropriate restore mode:
            01. `PointInTimeRestoreMod:` passes `RestoreTime` and `SourceDatabaseID`
            02. `LTRBackupRestoreMod:` passes `LtrBackupID` constructed from subscription / resource-group / server / database / snapshot
        ii. Configures `SqlDbSpecificCreateParams` with backup redundancy, elastic pool, and tags.
    b. For Rubrik (persistent backup) exports:
        i. Creates an empty database with SKU configuration
        ii. **Recovery size scaling:** the database is created 1.7x larger than the original to provide buffer space for index rebuilding during BCP import.
        iii. **Hyperscale auto-scaling:** Backups of databases >100 GiB, RSC will auto-scale to Hyperscale for faster recovery performance.

01. Initially provisions a General Purpose Serverless Gen5 database utilizing 32 vCores, as databases originally established as Hyperscale cannot be subsequently downscaled.

02. Subsequently, upgrade the database to Hyperscale Serverless Gen5, incorporating configurable vCores and Serverless properties, with a minimum of 10 cores and auto-pause functionality enabled.

03. Upon completion of the recovery process, the service tier is reverted to the original configuration.

iv. **Elastic pool support:** If elasticPoolName is specified, constructs the elastic pool ID and sets it on the create params.

v. **Tag export:** Copies tags from the source database (PiTR) or source snapshot (LTR) to the new database.

4. **Request Exocompute Cluster:** Requests an AKS exocompute cluster in the destination region. A Premium SSD managed disk for the exocompute pod as scratch space is attached to the Exocompute cluster.

5. **Recover Data Task:** This task runs through a 7-stage pipeline that deploys the database schema via DACPAC, then imports data via BCP bulk copy.

   a. **Stage 1: Initialize Config:** Sets up the CSL snapshot reader to extract schema item IDs from snapshot metadata and creates the recovery scratch directory on the mounted disk.

   b. **Stage 2: Import Database Schema from the DACPAC:**

      i. Deploys the schema via sqlpackage to publish against the target database with a 10-hour timeout.

      ii. After the schema is deployed, query the database to build a map of all tables organized by schema. This map drives the subsequent BCP data import

   c. **Stage 3: Prepare Database For Recovery: Optimizes the target database for bulk import:**

      i. Enables `SET QUOTED_IDENTIFIER ON` for compatibility

      ii. Identifies memory-optimized tables as these require special BCP flags (no TABLOCK hint, no SORT_IN_TEMPDB)

      iii. Identifies sparse-columned tables, as these skip DATA_COMPRESSION during index rebuild

      iv. Build a Table Metadata Registry tracking per-table constraints, triggers, and indexes

      v. **Disables constraints:** drops foreign key checks so rows can be loaded in any order

      vi. **Disables triggers:** prevents trigger execution during bulk load

      vii. **Drops non-clustered indexes:** they will be rebuilt after data import for better performance

   d. **Stage 4: Import Database Table Data via BCP:** RSC will use up to 64 processes to read BCP files in 500MB chunks and import the data into the tables.

e. **Stage 5: Perform Post Data Import Tasks:**

   i. Re-enables constraints

   ii. Re-enables triggers

   iii. Rebuilds clustered indexes

   iv. Rebuilds non-clustered indexes

f. **Stage 6: Validate Data Integrity:** Compares actual row counts in the target database against expected counts from BCP metadata. Fails the recovery if any table has a row count mismatch.

g. **Stage 7: Perform Post Recovery Tasks:** Remove temporary database users created during safe mode recovery.

6. **Post-Data Recovery Task:**

   a. **Exports tags:** copies source database tags to the recovered database

   b. **Scales down the database:**

      i. Hyperscale scaling back to the original service tier (e.g., General Purpose, Standard)

      ii. Non-Hyperscale server scaling back to the original max size

      iii. Elastic pool re-association if needed

7. **Release Cluster & Delete Empty Disks:** Releases the exocompute AKS cluster and deletes the temporary managed disk.

8. **Remove Refresh Token Reference:** Releases the refresh token reference.

Upload BACPAC

When selecting the *Upload BACPAC* method, a customer can either select an existing Azure Storage Account or create a new one. The Recovery Layer then uploads the BACPAC file to the identified Storage Account and provides the customer with a download link. The specific steps for this are as follows:

1. The user initiates the recovery workflow

2. Rubrik Exocompute resources are requested in the same region as the Azure SQL resource, and a disk is launched to store backup data.

3. Retrieve the desired snapshot data from the CSL (Azure Storage)

4. Reconstruct a BACPAC file on the local disk. The recovery layer retrieves the schema (DACPAC) and table data (BCP) separately from the CSL and assembles them into a single standard BACPAC file.

5. Upload the BACPAC file to Azure Blob Storage.

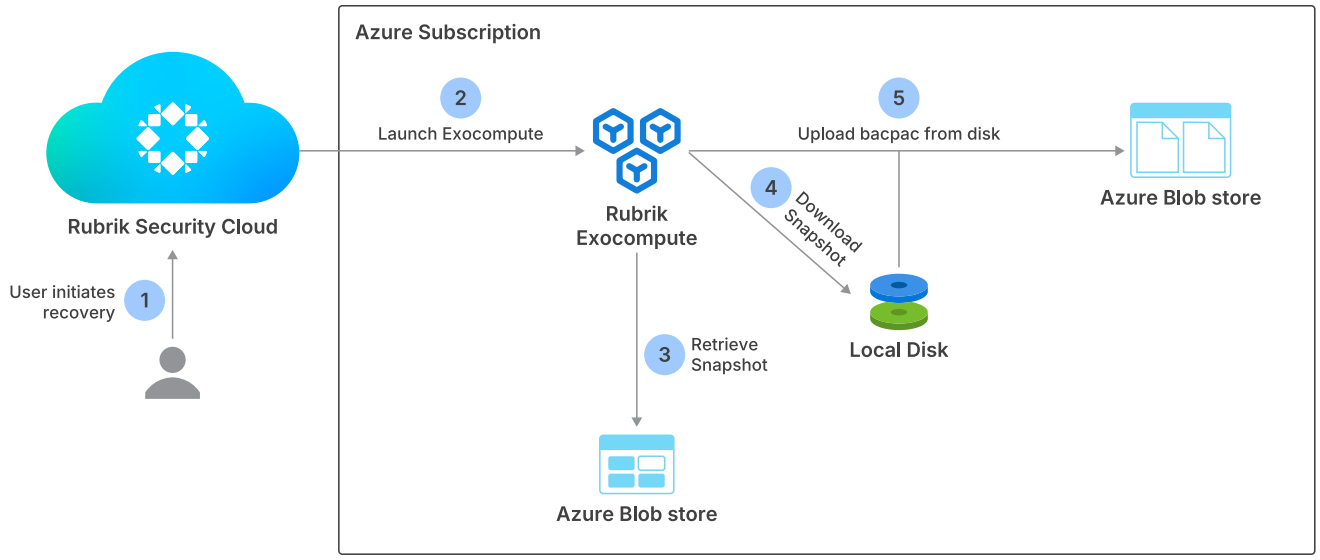

Figure 14 – Azure SQL DB Immutable Backup – Upload BACPAC

AZURE SQL MANAGED INSTANCE

1. User initiates a restore from RSC
2. RSC will acquire a reference to the Azure refresh token for the destination subscription with a 7-day TTL.
3. RSC will create a new database based on SQL DB-specific logic
    a. **For Native (PiTR / LTR) exports:** Uses the same ARM API restore modes as SQL Database, but with MI-specific LTR backup ID format.
    b. **For Rubrik persistent backups (BAK files):** BAK-based backups skip this task entirely
4. **Request Exocompute Cluster:** Requests an AKS exocompute cluster in the destination region.
5. **Recover Data Task:**
    a. Generate T-SQL Restore Command using striped Azure Blob storage URLs (up to 64 files).
    b. **Get AAD credentials:** BAK recovery requires AAD credentials on the storage account to generate SAS tokens. RSC populates these credentials, handling both standard cloud accounts and RCV (Rubrik Cloud Vault) archival locations
    c. Execute T-SQL Restore Command
6. **Release Cluster & Delete Empty Disks:** Releases the exocompute AKS cluster.
7. **Remove Refresh Token Reference:** Releases the refresh token reference.

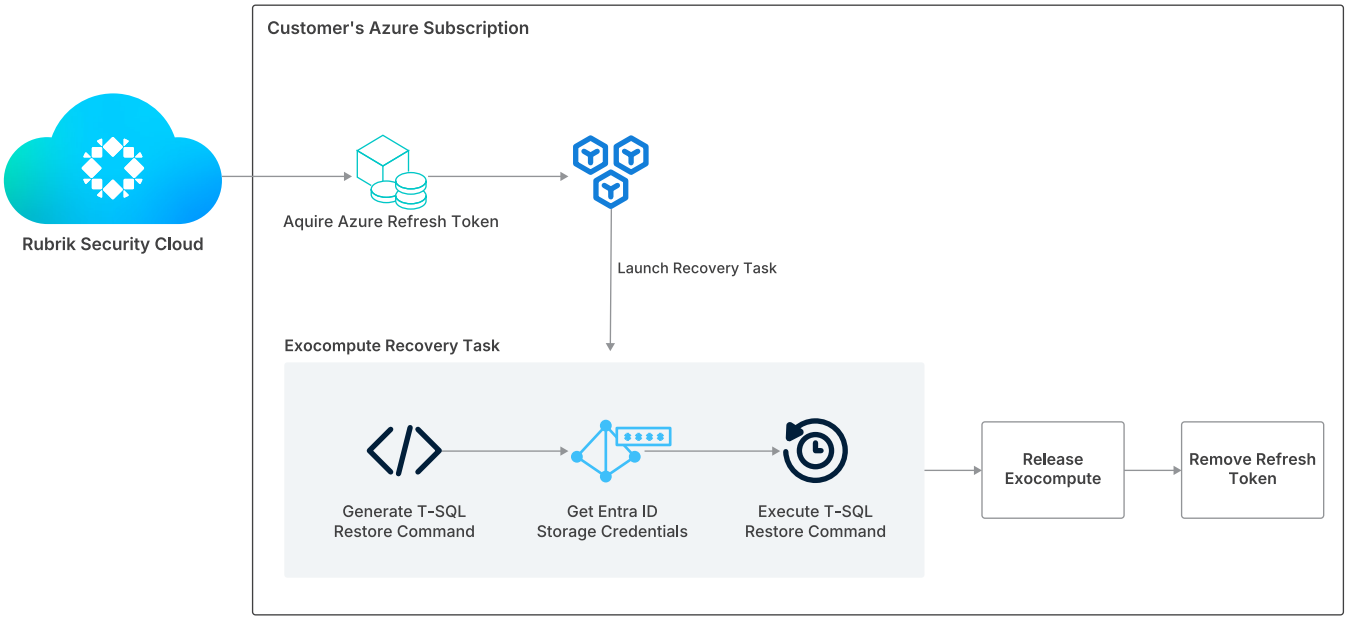

Figure 15 – Azure SQL MI Immutable Backup Restore

SUMMARY

This concludes How it Works: Cloud-Native Protection for Microsoft Azure SQL. This document explains the core components, architecture, and value proposition of Rubrik Security Cloud's protection of Microsoft Azure SQL.

For additional information, please visit https://www.rubrik.com or contact your Rubrik Account Team.

VERSION HISTORY

Version	Date	Summary of Changes
1.0	April 2023	Initial Release
2.0	May 2025	Added new functionality for Standard AGs, updated terminology
3.0	May 2026	Updated Azure SQL DB and MI backup and recovery functionality

rubrik

Global HQ
3495 Deer Creek Road
Palo Alto, CA 94304
United States

1-844-4RUBRIK
inquiries@rubrik.com
www.rubrik.com

rwp-hiw-cloud-native-protection-for-microsoft-azure-sql-with-rsc / 20260505