TECHNICAL WHITE PAPER

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

Rubrik Security Cloud

Brian Mislavsky
April 2023
RWP-0618

Table of Contents

INTRODUCTION

Welcome to *How It Works: Cloud-Native Protection for Microsoft Azure SQL*. The purpose of this document is to aid the reader in familiarizing themselves with the features, architecture, and workflows of protecting Microsoft Azure SQL with Rubrik Security Cloud. Such information will prove valuable while evaluating, designing, or implementing the technologies described herein.

AUDIENCE

This guide is for anyone who wants to better understand the capabilities of Rubrik Security Cloud's Microsoft Azure SQL protection and the technical architectures that underpin those capabilities. This includes architects, engineers, DBAs, and administrators responsible for Microsoft Azure and Microsoft Azure SQL infrastructure and data protection operations as well as individuals with a vested interest in security, compliance, or governance.

OBJECTIVES

The goal of this guide is to provide the reader with a clear and concise point of technical reference regarding the architecture and workflows utilized by Rubrik Security Cloud to protect Microsoft Azure SQL. After reading this document, the reader should be able to answer the following questions regarding the protection of Microsoft Azure SQL with Rubrik Security Cloud:

- What does Rubrik Security Cloud do?
- What problem(s) does protecting Microsoft Azure SQL with Rubrik Security Cloud solve?
- How does one configure and utilize protection of Microsoft Azure SQL?
- How is protection of Microsoft Azure SQL architected and why?
- How does protection of Microsoft Azure SQL protect against Ransomware and assist with Incident Response?
- How does protection of Microsoft Azure SQL with Rubrik Security Cloud compare to alternate solutions?

AZURE SQL AUTOMATED BACKUPS

Microsoft's Azure SQL database & Azure SQL Managed Instance include a basic backup offering that includes two fundamental types of protection—Point-in-time Retention (PiTR) backups and Long-term Retention (LTR) backups. Both of these are offered across various levels of redundancy that can be chosen by the customer to meet business needs, and each having a different set of features and limitations.

This section will briefly go over some core concepts of Azure SQL automated backups, and additional information can be found in 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 STR and can be stored as Azure blobs for up to 10 years in a Microsoft Azure managed storage account. Once LTR backups are configured, full backups can be copied to the storage account on a weekly, monthly, and yearly basis depending on the backup policy configured.

CHALLENGES

Cloud Native Protection Limitations

Digital enterprises are increasingly using multiple private and public clouds to deploy applications, avoid vendor lock-in, and exploit best-of-breed solutions. However, this fragments data within clouds, as well as across hybrid and multi-cloud infrastructures, fracturing IT's ability to protect, manage, and secure their data, operations, and business.

Public cloud providers themselves are responsible for the protection and availability of the cloud; however, it is still the customer's responsibility to protect resources in the cloud. What this means, practically speaking, is that it is ultimately the customer's responsibility to protect their applications and data running in a public cloud, regardless of provider. The Shared Responsibility Model published by Microsoft is a great point of reference for these concepts.

This leaves the customer at a critical decision point—*How do I efficiently and reliably protect my assets that reside in the cloud?* While the question seems simple on its face, selecting the appropriate solution is actually quite difficult.

In hybrid or multi-cloud environments, customers might be inclined to lift and shift legacy tooling into the cloud. Unfortunately, this approach often hampers the agility and elasticity that enterprises are seeking when adopting a cloud strategy.

The alternative, leveraging platform native tooling from the cloud provider themselves, can be similarly flawed as this segments data protection operations between public cloud providers as well as between public and on-premises environments. Such an approach leads to significant headwinds in terms of compliance, visibility, and operational efficiency.

Azure SQL Backup Limitations

While Microsoft Azure SQL includes some basic backup features, the service has some limitations when it comes to data protection:

- **Backup Persistence –** When an Azure SQL server is deleted, so are any of its PiTR backups. If LTR backups have been enabled for a database, they can be used to restore to a different server in the same subscription in the event of deletion.
- **Limited backup frequencies & retention limits –** Microsoft Azure SQL only supports built-in backup frequencies that cannot be customized. Additionally, there is a maximum retention of 35 days for PiTR backups (except for databases on the Basic or Hyperscale tier) and up to 10 years with LTR backups.
- **On-demand backups –** No ability to take an on-demand backup if needed. Automated backups can only be taken according to the predefined schedule.
- **Recovery limitations –** Both Azure SQL DB & Azure SQL Managed Instances have limitations around where they can be recovered to.
  - **Azure SQL DB –** PiTR backups can only be recovered to the original server in the same region & subscription. LTR backups can only be recovered to the same subscription.
  - **Azure SQL Managed Instance –** PiTR backups can only be recovered to the original region and LTR backups can only be recovered to the original subscription.
- **Limitations with Geo-Restore –** The default redundancy for both Azure SQL DB & Azure SQL Managed Instances is RA-GRS. While this can be changed to LRS or ZRS to reduce costs, RA-GRS replicates data only to a Microsoft determined paired region. Due to this, there could be a Recovery Point Objective (RPO) of 1 hour and a Recovery Time Objective (RTO) of up to 12 hours to recover data from the paired region. Additionally, Geo-Restore only comes into play in the event of a regional outage in the source region. Given the higher than normal demand during a regional outage, the failover region may not have enough resources to support every recovery right away.

THE RUBRIK APPROACH

Multi-Cloud Protection

Figure 1 – Rubrik Security Cloud Multi-cloud Protection

Rubrik's goals are to simplify and automate the ability to secure and protect data from events such as accidental deletion of data or ransomware using policy-based protection and frictionless operations. Rubrik Security Cloud is a Software as a Service (SaaS) based data protection platform providing automated backup, recovery, and replication schedules across regions and across clouds with a single global policy engine. This solution allows Rubrik customers to reap the benefits of rapid innovation and reduced management complexity with data security delivered as a service.

Protecting Azure SQL workloads with Rubrik Security Cloud consists of 3 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 builds on Azure SQL's built-in backup offering and provides users greater flexibility, protection granularity, and recoverability options. Rubrik customers can now overcome the limitations of Azure SQL's built-in data protection and do so across Azure Tenants, Subscriptions, and Regions.The key features of Rubrik's Immutable Backups for Azure SQL are:

- Unified data management across regions, subscriptions, tenants, and services
- Automated Global Data Protection with Rubrik Security Cloud SLA Domains
- Enhanced security with native immutability
- Backup persistence of databases beyond life of Azure SQL Server
- Increased frequency and granularity options for backups
- Recovery of databases to the user's choice of Azure Subscriptions/Regions
- Long-term retention of backups in a customer managed Storage Account

UNIFIED DATA MANAGEMENT ACROSS SUBSCRIPTIONS AND CLOUD PLATFORMS

Single Point of Management and Automation via Rubrik Security Cloud – Rubrik's Security Cloud SaaS platform is a single point of management and automation for hybrid and multi-cloud environments. It requires no persistently running compute resources in the customer's Azure environment. Rubrik Security Cloud provides customers with a simple, homogenous data management experience across platforms that reduces the drag associated with legacy tooling and point solutions.

Consolidated Reporting – Easily track SLA Domain assignment, protection and recovery activity, and SLA policy compliance across subscriptions, tenants, platforms, and clouds from a single, easy to use reporting engine.

AUTOMATED GLOBAL DATA PROTECTION WITH RUBRIK SECURITY CLOUD SLA DOMAINS

In the data protection world, Service Level Agreements (SLAs) define protection levels for workloads, availability targets, and objectives that are crucial to a company. Collecting this information, implementing it, and staying compliant with the SLA is usually a tedious and difficult process. Rubrik Security Cloud uses global SLA Domains, a declarative, policy-driven framework, to make achieving your SLAs easier.

Global Protection – Rubrik Security Cloud SLA Domains can be assigned across object types even if those objects are spread across clouds, subscriptions, or on-premises. This allows one set of policies to be used to manage data wherever it may be in the environment.

Subscription and Resource Group Level Auto-Protection – Assign SLA Domains to entire Azure subscriptions or resource groups and ensure that every Azure SQL Database provisioned receives the required level of data protection without the need for explicit SLA assignment. Subscription and resource group level SLA Domain assignments can be overridden using tag-based assignment or by directly assigning SLA Domains to Azure SQL or Managed Instance Databases.

Tag-Based Auto-Protection – Allows for the assignment of SLA Domains to Azure SQL or Managed Instance Databases whenever a specific tag key or key value pair is found. This includes any Azure SQL or Managed Instance Database in any Azure subscription in scope. These tag rules allow customers to leverage existing provisioning and governance logic to apply the appropriate SLA Domains across Azure tenants, subscriptions, and regions without the need for manual intervention.

ENHANCED SECURITY WITH NATIVE IMMUTABILITY

When configured, Rubrik Immutable Backups for Microsoft Azure SQL provides additional resilience to Azure Databases and Managed Instances by making snapshot data immutable.

BACKUP PERSISTENCE OF DATABASES BEYOND LIFE OF AZURE SQL SERVER

Once an Azure SQL server is deleted, any databases on the server are also deleted, preventing recovery using PiTR backups as they are stored on the server. If optional LTR backups have been configured via Rubrik Security Cloud, this is a non-issue. Rubrik Security Cloud leverages an Azure Storage Account to provide persistent storage for use with its protection of Microsoft Azure SQL databases. This allows backups taken by Rubrik Security Cloud to persist even if the server the database was housed on has been deleted.

INCREASED FREQUENCY AND GRANULARITY OPTIONS FOR BACKUPS

When Storage Persistence for Azure SQL is enabled in Rubrik Security Cloud, customers can take backups at more granular frequencies than Azure SQL's automated backups as well as on-demand backups. Once enabled, Rubrik Security Cloud takes database backups in the Microsoft SQL BACPAC format and stores them in a Storage Account defined by the customer.

RECOVERY FOR MICROSOFT AZURE SQL ACROSS SUBSCRIPTIONS OR REGIONS

As Rubrik Security Cloud creates Microsoft SQL compatible backups that are decoupled from the Azure SQL service, they can be used to recreate or recover a SQL database regardless of its location. This includes Azure SQL servers in subscriptions and regions other than the source.

LONG TERM RETENTION OF BACKUPS IN A CUSTOMER MANAGED STORAGE ACCOUNT

Since backups taken by Rubrik Security Cloud can be sent to an Azure Storage Account with retention defined by a customer's defined SLAs. This is a simplified and more cost-effective way to handle long-term retention, since the behavior is the same regardless of the tier of database that is deployed. While Azure SQL's automated LTR backups are stored in a Microsoft managed Storage Account, Rubrik Security Cloud stores LTR backups in a Storage Account owned and managed by the customer.

ARCHITECTURE AND COMPONENTS

HIGH-LEVEL ARCHITECTURE

Cloud-Native Protection for Microsoft Azure SQL allows customers to take advantage of the power of Rubrik's SLA policy engine via Rubrik Security Cloud to protect Azure SQL Databases and Managed Instances inside of their Azure subscriptions. Rubrik does so by allowing customers to choose from one of two data protection types: orchestration of Azure SQL automated backups or Rubrik's Persistent Backup for Azure SQL.

Azure SQL Automated Backup Orchestration

The following workflow outlines the high-level list of steps involved with protecting an Azure SQL Database or Managed Instance by is used when protecting Microsoft Azure SQL resources in a protected Azure subscription using Rubrik Security Cloud.

1. Rubrik Security Cloud authenticates into the customer's Azure tenant using a service principal. This service principal is created by Rubrik Security Cloud when the customer enables Cloud-Native Protection for their Azure subscription(s). The role assigned to this service principal grants it the necessary permissions to protect and restore Azure SQL Databases in the customer's subscription(s). The credentials for the corresponding application object are stored in an encrypted format within a customer-specific database in Rubrik Security Cloud.

2. Customers leverage Rubrik Security Cloud to configure & assign SLAs for discovered Azure SQL Databases or Managed Instances. RSC then uses Microsoft Azure Resource Manager APIs to synchronize the native backup policies with Rubrik SLAs, as well as perform database recovery.

Rubrik Immutable Backups

The following workflow outlines the steps involved with protecting an Azure SQL Database or Managed Instance using Rubrik's Immutable Backups for Azure SQL:

1. Rubrik Security Cloud authenticates into the customer's Azure tenant using a service principal. This service principal is created by Rubrik Security Cloud when the customer enables Cloud-Native Protection for Azure SQL/Managed Instances in their Azure subscription(s). The role assigned to this service principal grants it the necessary permissions to protect and restore Azure SQL Databases/Managed Instances in the customer's subscription(s). The credentials for the corresponding application object are stored in an encrypted format within a customer specific database in Rubrik Security Cloud.

2. Rubrik Security Cloud Immutable Backups are configured per Azure Subscription and used to protect any Azure SQL DBs/Managed within it. Database credentials (OAuth with Azure Active Directory or Local Database Admin) are assigned to each Database and used to create a local database user that Rubrik will leverage in order to enable Change Data Capture (CDC) on the databases requiring protection.

   Alternatively, customers can run a script (detailed in later sections) in order to enable CDC and create a dedicated backup user on their databases. The Persistent Storage is a Microsoft Azure Storage Account specified for storing Persistent Backups.

3. Customers then leverage Rubrik Security Cloud to configure & assign SLAs for discovered Azure SQL Databases or Managed Instances.

4. When a backup is triggered based on an SLA, Rubrik Security Cloud makes a request for Rubrik Exocompute resources. Once Exocompute resources are available, a temporary disk is attached to the Exocompute node

5. An Exocompute task is launched to take a backup of the database and store it in *BACPAC* format on the attached local disk.

6. When the backup finishes, another exotask is launched to fetch CDC data from the database *BACPAC* file and CDC data are uploaded to Azure Blob storage located in the Storage Account defined by the customer and then merged

7. If no longer needed for other tasks, Exocompute and disk resources are spun down and terminated.

8. Snapshot metadata is securely stored in Rubrik Security Cloud.

HOW IT WORKS

As stated previously in this document, protecting Microsoft Azure SQL DBs and Managed Instances consists of 3 steps: *Authorize*, *Configure*, and *Protect*. This document dives deeper into the *Authorize* and *Protect* steps. 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 an understanding of how protecting Microsoft Azure SQL 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 very simple process:

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

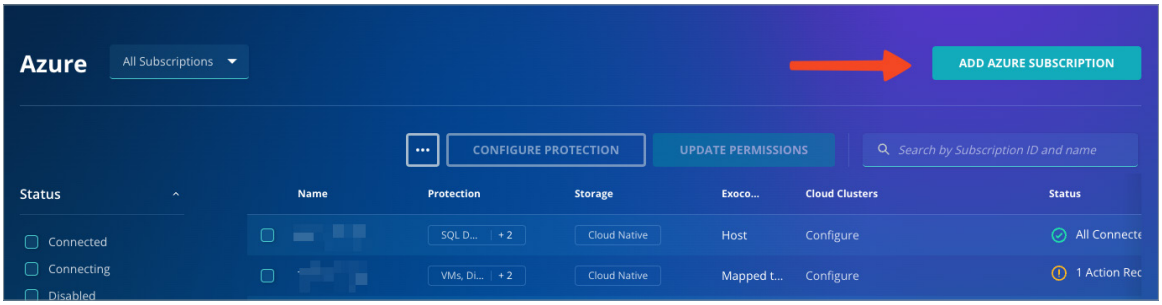

Figure 2 – Adding a Microsoft Azure Subscription - Launch Add Cloud Account Wizard

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

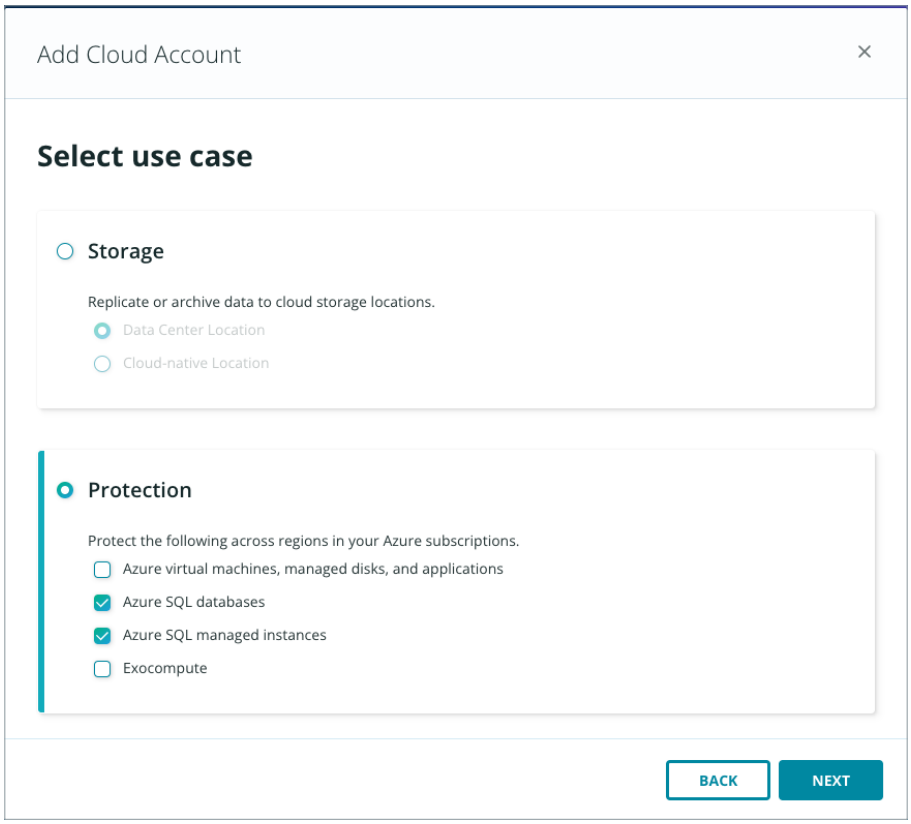

Figure 3 – Adding a Microsoft Azure Subscription - Select Azure SQL Use Case

The wizard then guides the customer with logging into the specified Azure Active Directory with a user that has the ability to 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 and then clicks Submit. The process will then create roles in the selected Subscriptions and assign the roles to a Service Principal created by Rubrik Security Cloud as well as ask the customer to configure Exocompute if required (this can be configured later and is detailed in the next section).

In order to protect Microsoft Azure SQL, Rubrik Security Cloud needs a means by which to interact 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 Azure Active Directory.

Azure Active Directory itself is quite powerful and supports a variety of identities such as users, groups, federated users and groups, as well as service principals. Permissions are delegated to or revoked from these identities via roles, which define what actions a specific identity can and and cannot take. 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 credentials of the user enabling protection is only used when initially adding subscriptions to Rubrik Security Cloud.

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

The figure below depicts how the workflow interacts with a customer's Azure subscription(s) from an Azure AD perspective.

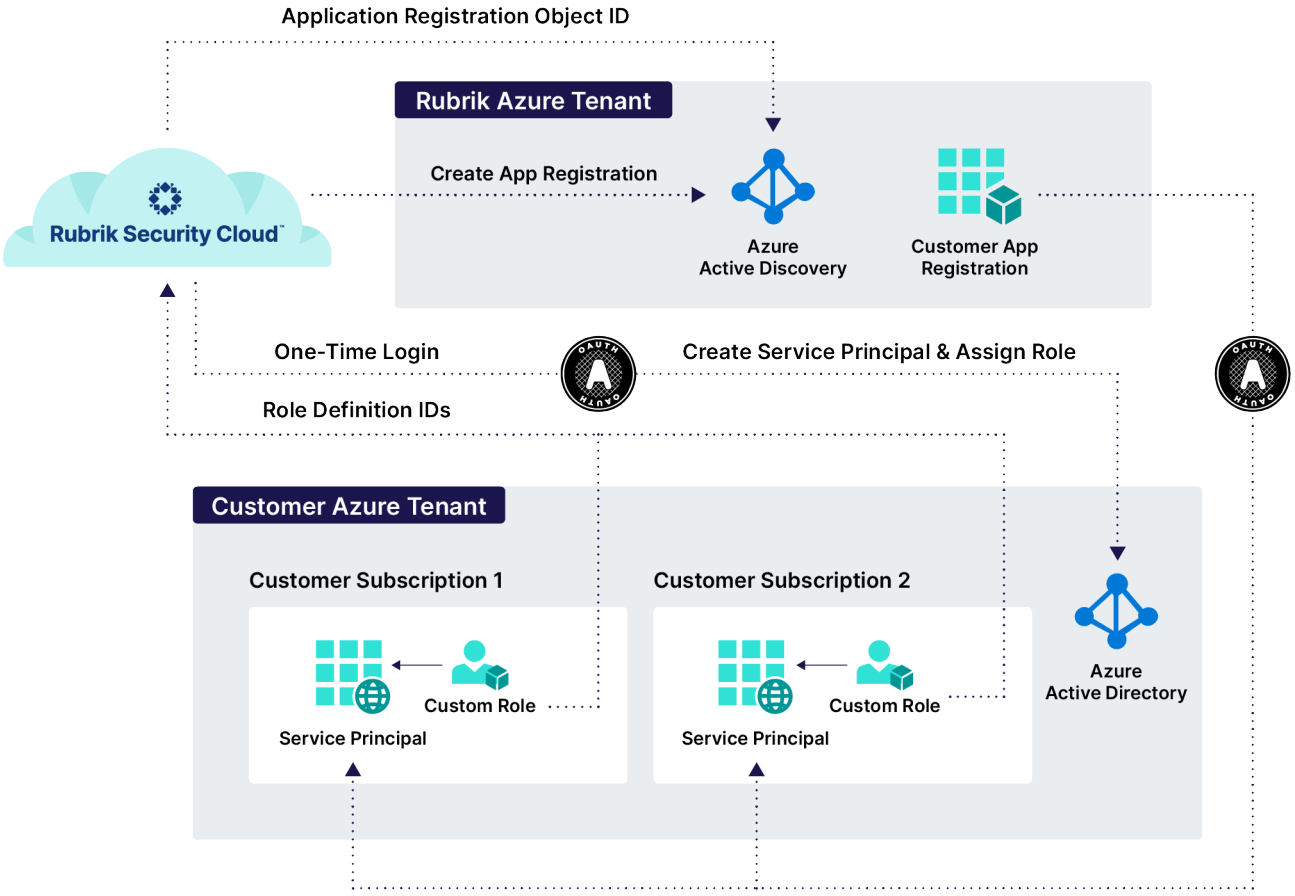

Figure 4 – Rubrik Security Cloud Application Registration Workflow

In Azure Active Directory, there are two representations of applications—application objects, also known as Application Registrations, and Service Principals, also referred to as Enterprise Application Registrations. Application objects describe an application to Azure Active Directory and can be considered the definition of the application, allowing Azure to know how to issue tokens to the application based on the app registration's settings. This application object only exists in its home tenant, even if it is a multi-tenant application supporting Service Principals in other directories. Service Principals are what govern an application connecting to Azure Active Directory and can be considered the instance of the application in the customer's directory. For any given application, it 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 the customer

initially adds their subscriptions to Rubrik Security Cloud for protection. The permissions delegated to this service principal are controlled via the custom roles assigned to the service principal in each subscription. This architecture essentially builds a type of trust between the customer's Azure Active Directory and Rubrik's, allowing Rubrik Security Cloud to interact with the Azure SQL APIs once it authenticates into Rubrik's Azure Active Directory. A major 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 is aware that the necessary permissions are in place and has all the information needed 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 walk through updating the role via OAuth. The user simply initiates the workflow in Rubrik Security Cloud, logs in, and authorizes the role changes when prompted

There are also alternative methods available to add Microsoft Azure subscriptions into Rubrik Security Cloud. If one of these approaches is necessary in your environment, please reach out to 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 with Rubrik Security Cloud can be found here.

As mentioned previously, Rubrik provides two modes of protection - orchestration of Azure SQL's basic automated backups as well as Rubrik Immutable Backups for Azure SQL. The following table can be found in the above mentioned product documentation and is included here as 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 takes and manages the short-term and long-term retention backups according to the SLA Domain configuration	When persistent backups are enabled, RSC supports exporting databases to a SQL Server in any location. Recovery can be across different subscriptions or to a cloud or on-premises setup that is independent of Azure.
Long-term retention (LTR) backups	When Immutable backups are not configured, RSC supports the management of backups taken natively by Azure based on the non-daily frequencies defined in the SLA Domain configuration. LTR backups have a retention period of up to 10 years.	When persistent backups are not configured, RSC supports exporting databases from LTR backups across different Azure regions within the same subscription as that of the source database server or managed instance.

Backup Type	Description	Recovery
Point-in-time (PiTR) backups	Short-term retention backups are retained for a maximum of 35 days depending on the service tier of the Azure SQL workload.	RSC supports point-in-time restore (PiTR) which creates a new database from backups taken at any point in time in the specified retention period. PiTR is limited to recovery within the Azure region of the source database or managed instance.

Azure SQL automated backup orchestration

For customers wanting to leverage the included basic automated backup functionality of Azure SQL, Rubrik Security Cloud uses Microsoft Azure APIs for Azure SQL DBs and Azure SQL Managed Instances to set the automated backup configuration of the protected databases. No additional configuration needs to be done by the customer at this point and an SLA can be configured according to the limits defined by the Azure SQL purchasing & 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.

Rubrik Immutable Backups

In order for customers to leverage Rubrik's Immutable Backups for Microsoft Azure SQL, some additional configuration needs to take place after connecting Rubrik Security Cloud to their Microsoft Azure Subscription(s). While some of these steps can be completed manually, Rubrik Security Cloud includes a wizard that coordinates the entire process.

Once Immutable Backups are configured for an Azure Subscription, PiTR backups continue to be based on Azure SQL's native backups while LTR & On Demand backups are now protected using Rubrik Immutable Backups.

NOTE: At this time Azure SQL Service tiers Basic, S0, S1, S2 do not support CDC and therefore do not support Immutable Backups with Rubrik Security Cloud.

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

PROTECTION

Protection is prioritized according to the level at which an SLA is assigned. The protection hierarchy for Azure SQL Databases can be found here and the Azure SQL Managed Instance protection hierarchy can be found here.

Rubrik Security Cloud handles batching all snapshot jobs so as not to overrun the API limits on Azure with one big batch of snapshot or replication activities. By default, Rubrik Security Cloud will run a maximum of 20 protection jobs in parallel per protected object type (e.g. - Azure SQL Database). Let's dig into the specifics of how Rubrik protects Azure SQL databases and Managed Instances.

Protecting Azure SQL without Immutable Backups

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 begin automatically scheduling and snapshotting Azure SQL Databases and Managed Instances in accordance with the SLA Domains that have been created and assigned without needing to schedule any jobs.

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: As this method of protection leverages native Microsoft Azure SQL backups to protect data, backup policies adhere to any/all guidelines. If a different retention period, recovery location, or backup frequency is desired, customers should consider protecting Azure SQL using Rubrik's Immutable Backups.

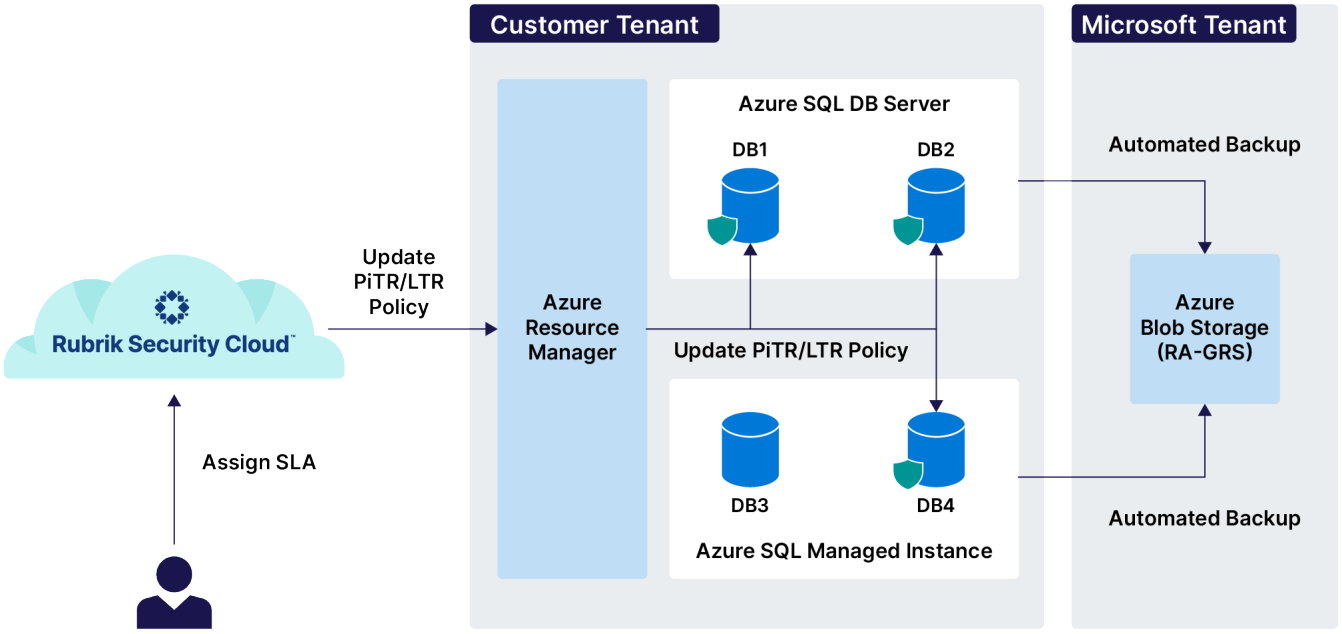

Figure 5 – Azure SQL backups without Immutable Backups

When an SLA that corresponds to a Short Term Retention is assigned to an Azure SQL Database by a user, RSC uses the following 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 similar process as configuring backups.

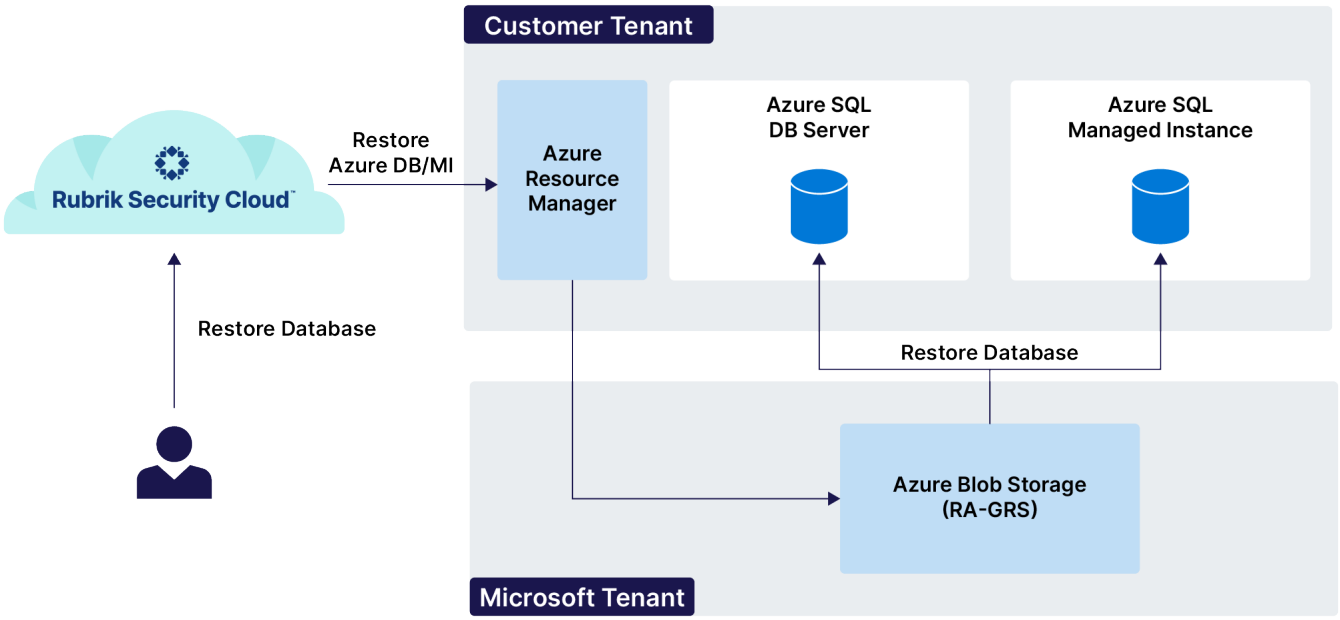

Figure 6 – Azure SQL recovery without Immutable Backups

When not using Immutable Backups, customers leverage RSC to view backups that have been taken and kick off a recovery from them. Behind the scenes, while RSC leverages the native Azure Resource Manager APIs for Azure SQL to perform the restores, there is a minor difference in how PiTR & LTR restores take place.

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 would ultimately be needed by RSC if it kept metadata about PiTR backups. Since the result of the `earliestRestoreDate` APi query (see below for list of APIs used) would be relatively static in nature for a given day, there is no additional benefit in keeping this data in RSC. The STR database recovery workflow is depicted in Figure 7 below.

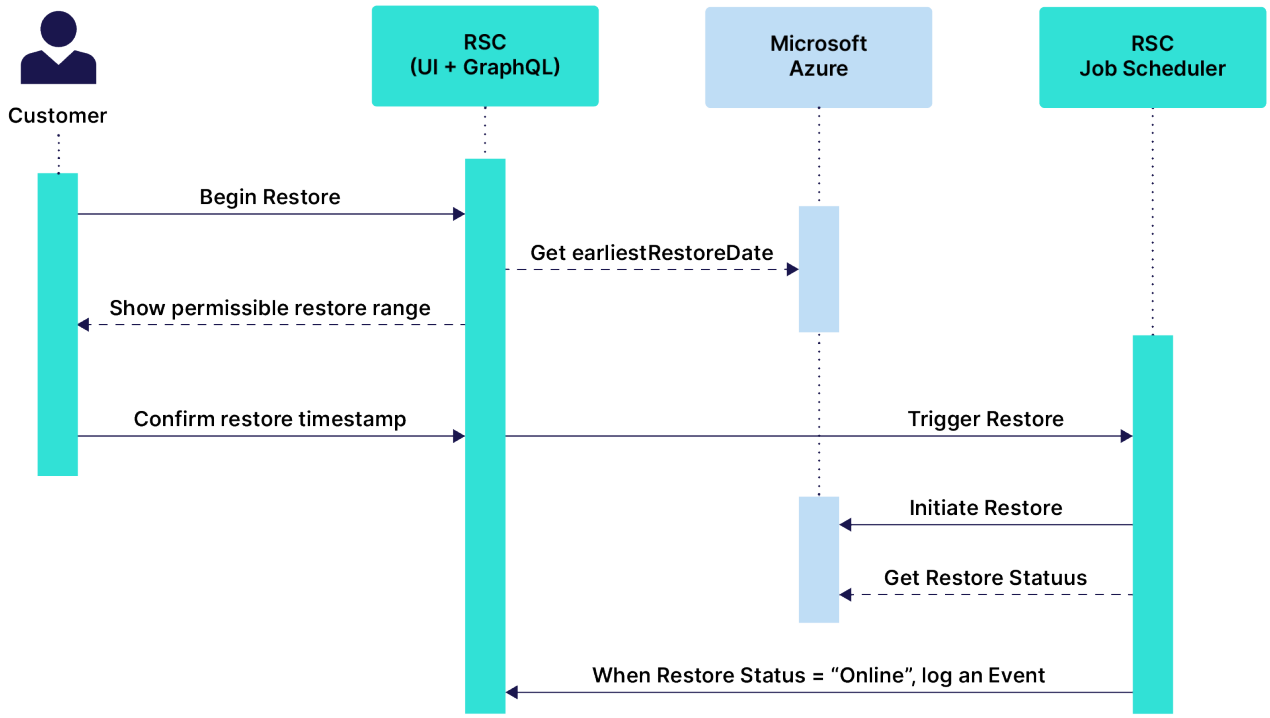

Figure 7 – Azure SQL database recovery flow - PiTR restore

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

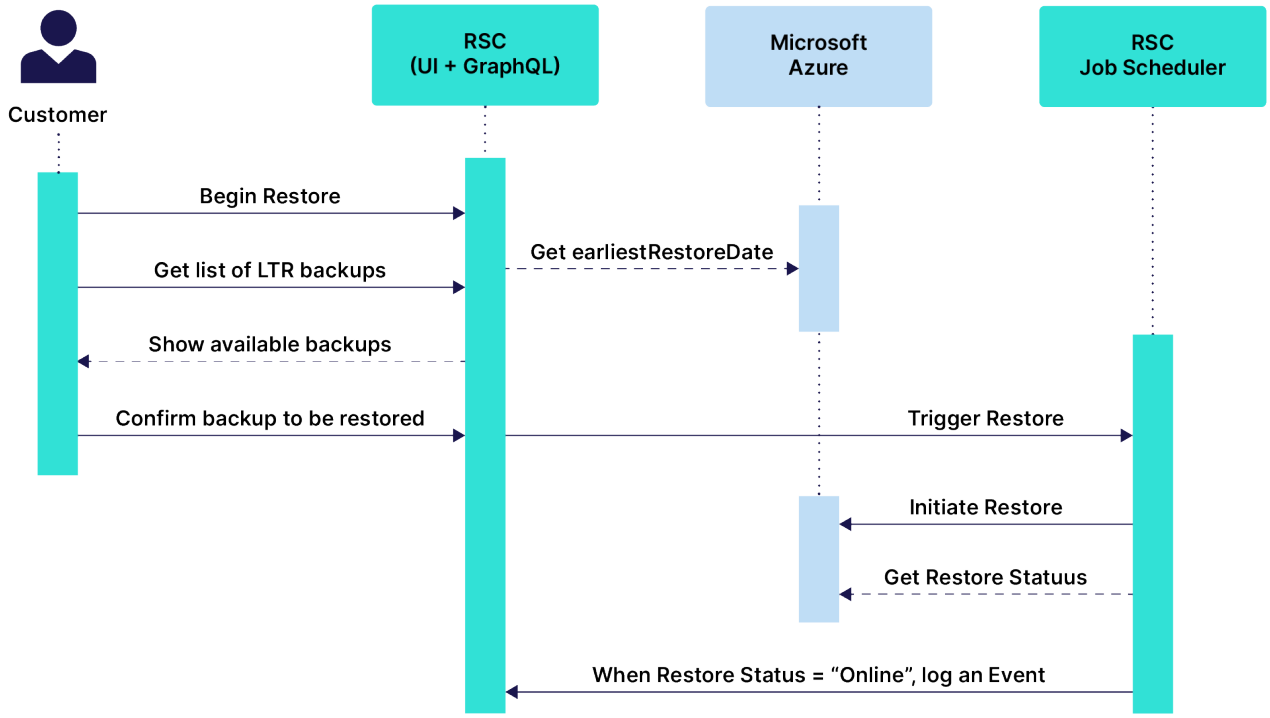

Figure 8 – Azure SQL recovery flow - Long-term retention recovery

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 with Immutable Backups

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 for protecting Azure SQL and allows customers to address the limitations listed in previous sections.

COMPONENTS

In order to provide persistence to backups of Azure SQL, Rubrik leverages some additional features/technology that are 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 in order 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.

For Azure SQL Immutable Backups, Rubrik leverages Exocompute resources to perform database exports using the SQLPackage utility.

Change Data Capture (CDC) & Local Database User

RSC leverages a local database user in order to perform backups. Additionally, in order to ensure transactional consistency of the backups, Rubrik requires that CDC be enabled for every database being protected with Immutable Backups. Enabling CDC is a one time action done prior to enabling Immutable Backups.

Rubrik provides three options to create a local user and enable CDC. Instructions on selecting the appropriate option are outlined in the Product Documentation here.

1. **OAuth –** This option relies on an Azure SQL administrator authorizing RSC to create backup credentials and enable CDC by using OAuth to sign-in to a specific Azure Active Directory domain.

2. **Database administrator credentials –** This option allows RSC to create backup credentials and enable CDC by connecting to the database using a local database administrator credential. RSC does not store these credentials anywhere except in the cache, and removes them from the cache after 30 minutes. The database administrator can also manually clear the credentials from RSC.

3. **Manually created backup credentials –** RSC allows database administrators to download and configure a script to manually perform the prerequisite tasks required for taking persistent database backups. When run on the Azure SQL workload, the script creates the credentials specified in the script and enables CDC for taking backups using those credentials. RSC allows the database administrator to manually clear the credentials after taking database backups.

For Options 1 & 2, RSC creates the local database user and stores its credentials as follows:

- **Database credentials**
  - Username – The local database user that Rubrik creates will use the following format:

    ```
    rubrik_login_<random_alphanumeric_string_upto_20_chars>
    ```

  - Password – The local database user will have a randomly generated password set according to the guidelines set forth here.

- **Encryption at rest –** Database credentials will be encrypted before being stored in a centrally managed and isolated Cloud KMS

- **Encryption in flight –** Communication between RSC and the Exocompute utilizing the credentials in the customer subscription is secured using TLS 1.2, with each task using short-loved TLS certificates and keys.

- **Logging –** Rubrik leverages a custom code function to handle secrets that ensures that any/all credentials used throughout Rubrik Security Cloud are never logged, preventing accidental revealing of secrets.

Cloud Storage Layer (CSL)

Rubrik's in-house developed unified snapshot aware storage layer for Cloud Native workloads. The CSL enables Rubrik Security Cloud to store data ranging in size 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 functionality, which offer significant storage and API cost savings to customers as 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 by default with versioning & immutability enabled on the storage container using the Set Blob Service Properties & Blob Containers - Create APIs.

If a customer uses an already 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 a Immutable Backup to be taken of an Azure SQL Database or Managed Instance, the following high level steps take place:

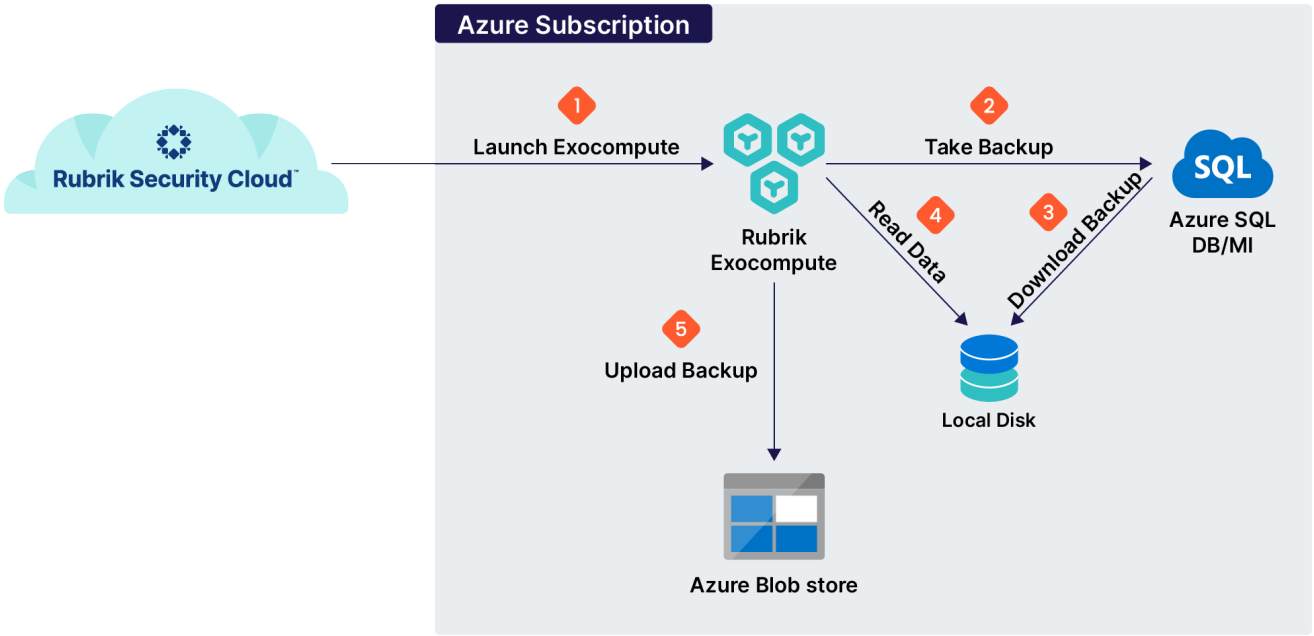

Figure 9 – Immutable Backups - high level steps

1. 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.
2. Once Exocompute is available, launch a task to take the backup as well as launch an empty disk to store the backup
3. Database backup is taken in BACPAC format and stored on local disk. CDC changes are collected from the database.
4. Database backup metadata is read and stored in RSC database
5. Database backup data (BACPAC file + CDC changes) is ingested into Azure Blob Storage

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

Backup Layer

The backup layer is responsible for taking the snapshot of the Azure SQL database and involves the following steps:

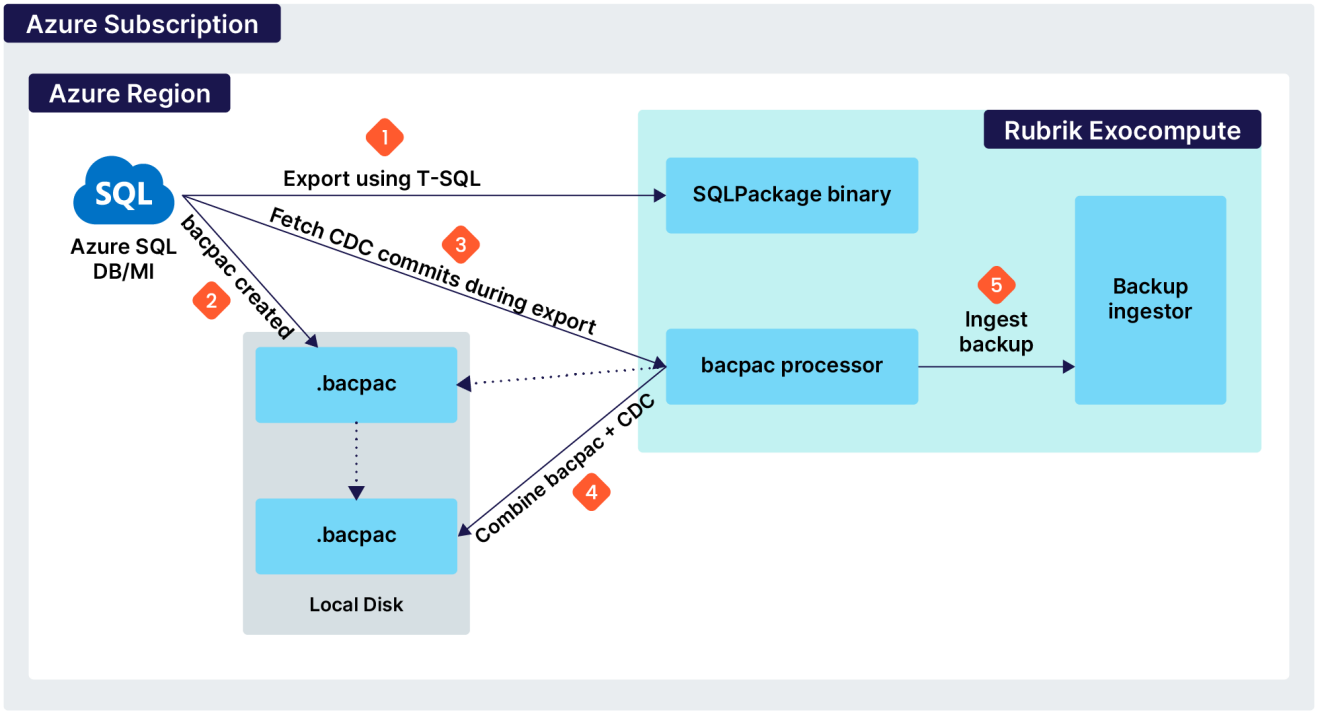

Figure 10 – Backup layer steps

1. Database export as a BACPAC file starts using the SQLPackage binary
2. BACPAC file is stored on local disk
3. Fetch CDC changes that occurred during database export process
4. Combine BACPAC file and CDC changes into a new BACPAC file.
5. Call ingestion layer to ingest the backups to Azure storage.

Additionally, there is a periodic task that automatically runs that reads the BACPAC file and CDC data and "repairs" the backup in order to make the backup transactionally consistent. The Repair process is made up of the following steps:

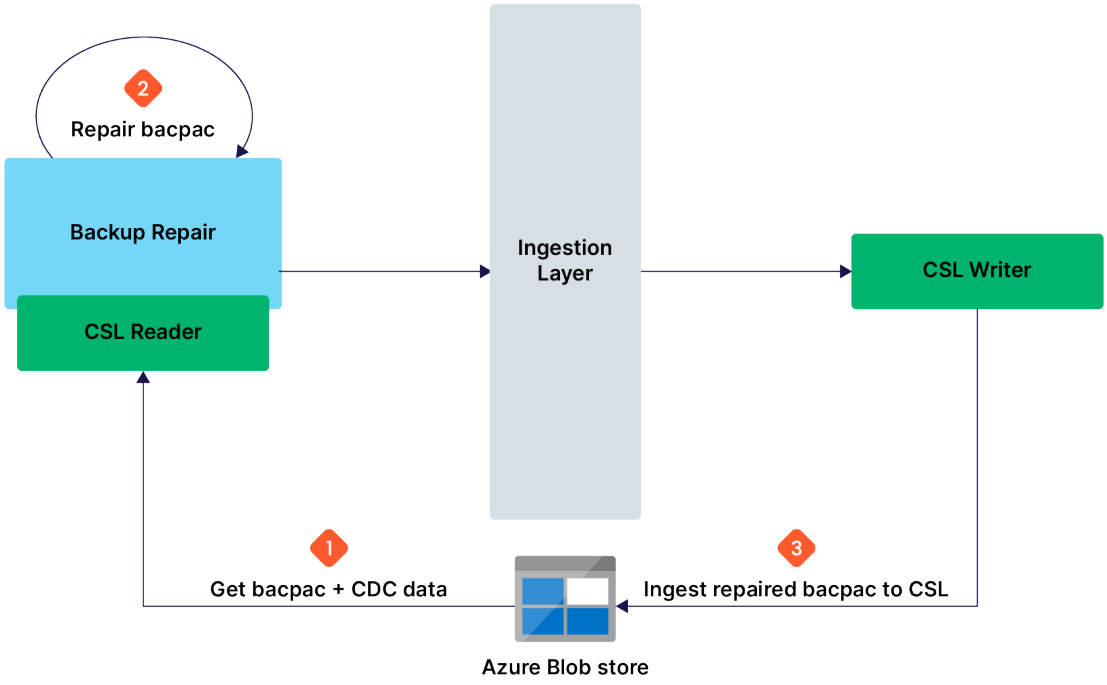

Figure 11 – Azure SQL Immutable Backup - BACPAC repair process

1. Determine list of snapshots that require repair
2. Launch an Exocompute task for each snapshot to repair the backup
3. Once the snapshot is repaired, mark the backup as recoverable.

Ingestion Layer

The ingestion layer of the backup process is responsible for ingesting the backup as well as storing it. As previously mentioned, Rubrik leverages its CSL for persisting Azure SQL backups.

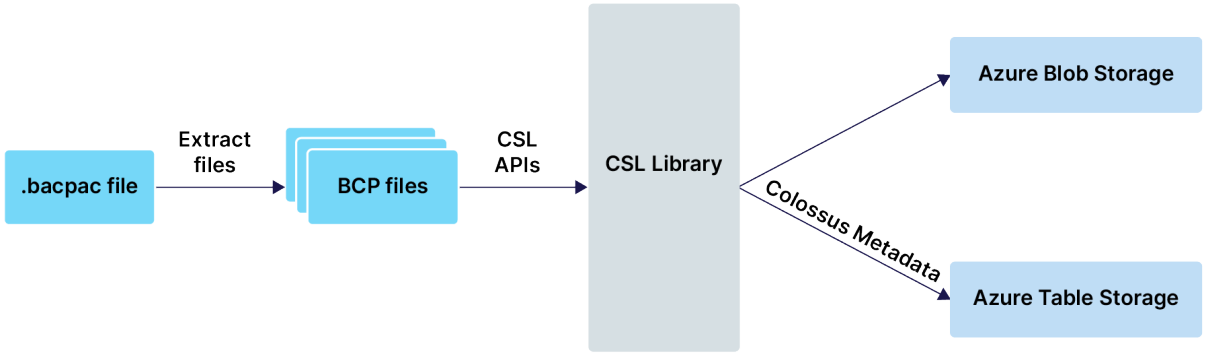

Figure 12 – Azure SQL Immutable Backup - Ingestion Layer

The CSL works at a file level, therefore, once the BACPAC file is created Exocompute uncompresses it and the individual files are extracted and written to Azure Blob storage. This process maximizes the deduplication of Azure SQL backups, which in turn leads to additional cost savings for Rubrik customers.

IMMUTABLE BACKUP RECOVERY

Recovering from an Azure SQL Immutable Backup is always a user driven action. When a user selects a snapshot to recover from, they are presented with a choice to either download the database BACPAC file from a user defined Azure Storage Account or recover the database to a server.

Similar to how taking an Azure SQL Immutable backup works, recovery involves two main components: The Retriever Layer and the Recovery Layer.

Retriever Layer

The retriever layer is responsible for retrieving the requested snapshot from Azure Blob storage. As shown in Figure 13 below, the CSL reader process will fetch the data corresponding to the requested snapshot and write it to the local disk as a BACPAC.

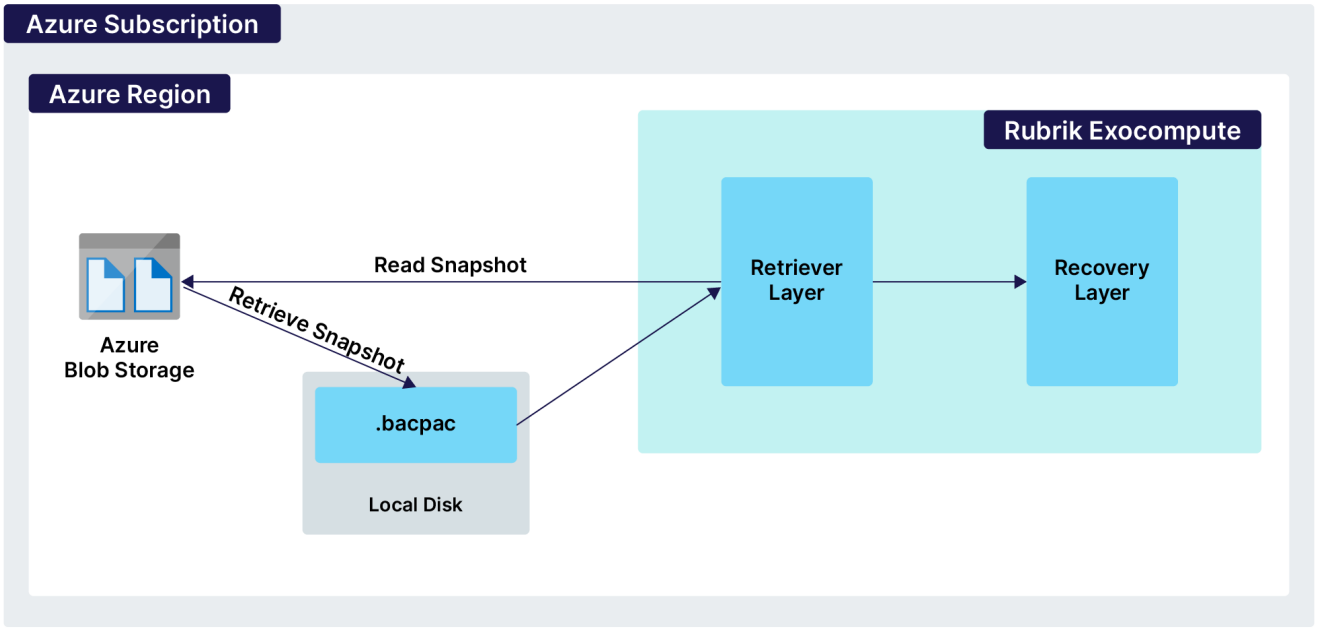

Figure 13 – Azure SQL Immutable Backup - Recovery - Retriever Layer

Recovery Layer

The purpose of the recovery layer of the restore process is to take the BACPAC file created by the retriever layer and recover the snapshot data from it. As mentioned, this is performed in one of two ways, depending on the method of recovery selected by the customer.

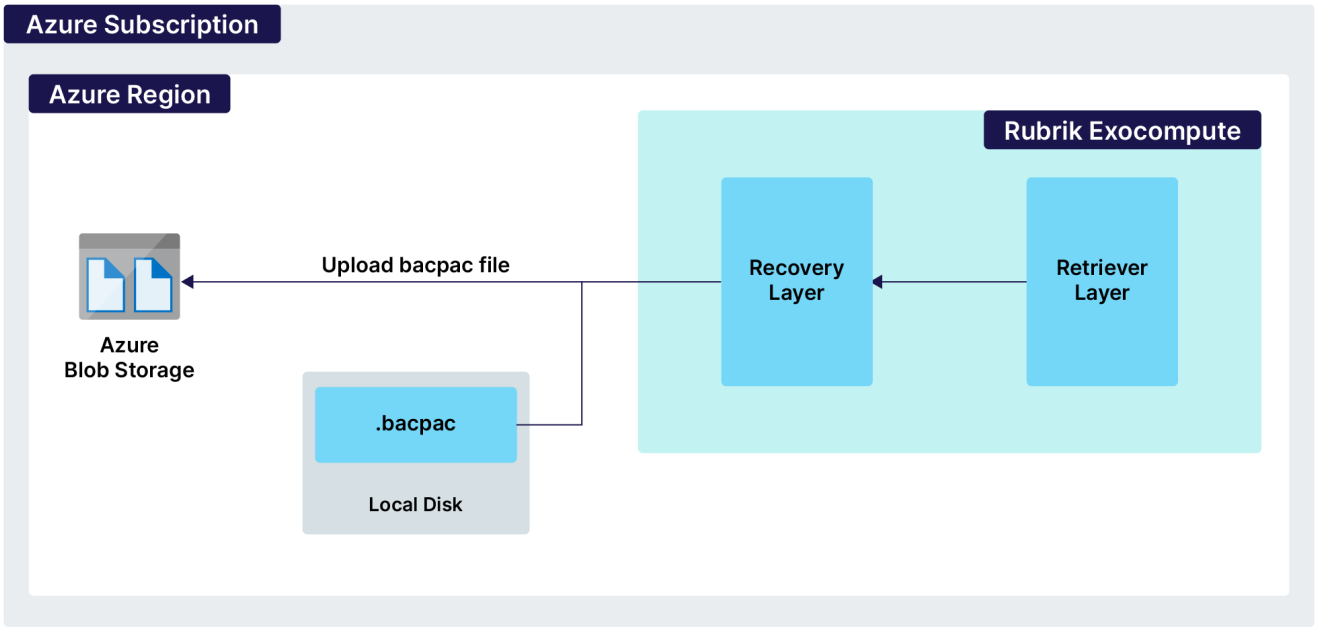

Figure 14 – Azure SQL Immutable Backup - Recovery Layer

Upload BACPAC

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

1. Recovery workflow is initiated by user
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. Read the snapshot data and construct a BACPAC file from it on the local disk
5. Upload the BACPAC file to Azure Blob Storage and then provides the user with a link to the downloadable BACPAC file

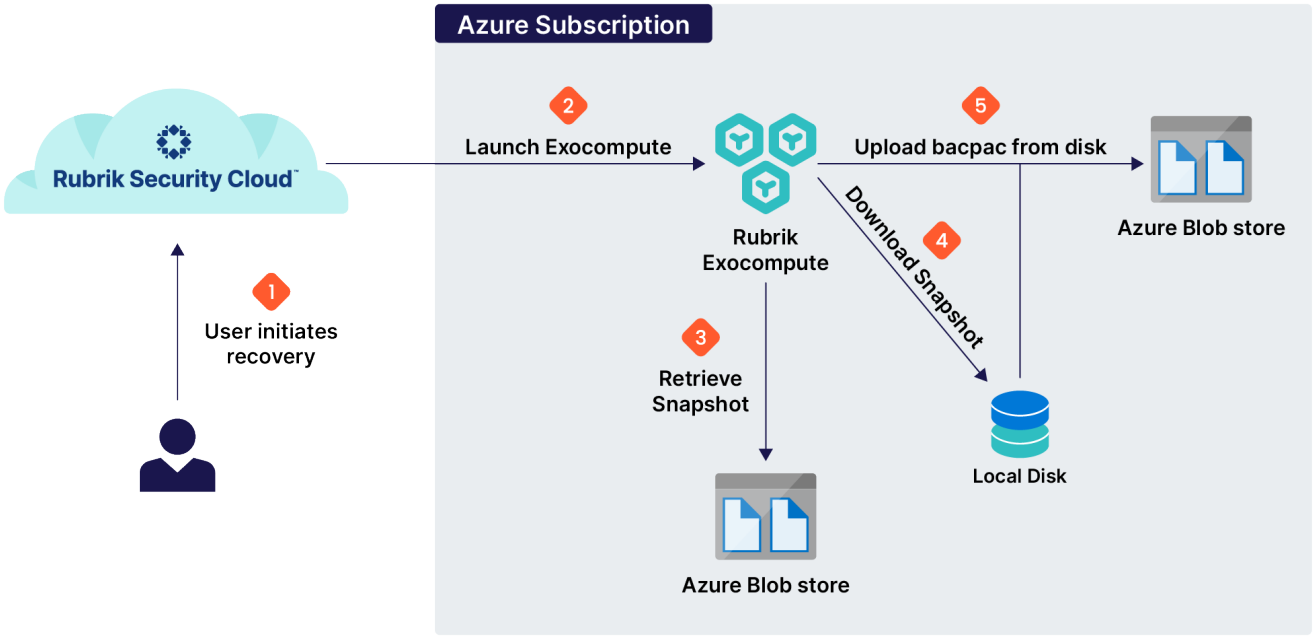

Figure 15 – Azure SQL Immutable Backup - Upload BACPAC

Recover to Database

When selecting the Recover to Database method, a customer uses a wizard to provide credentials (to create a database) and database information for a target server to recover the database to.

The Recovery Layer then either creates a new database (using the specifications of the source, taken at time of backup) or uses a preexisting one and exports the backup data into the identified database according to the following steps:

1. Recovery workflow is initiated by user
2. A new empty database is created (if needed)
3. Rubrik Exocompute resources are requested in the same region as the Azure SQL resource
4. A disk is launched to store backup data.
5. Retrieve the desired snapshot data from the CSL (Azure Storage)
6. Read the snapshot data and construct a BACPAC file from it on the local disk
7. Use SqlPackage to import the data into the new database

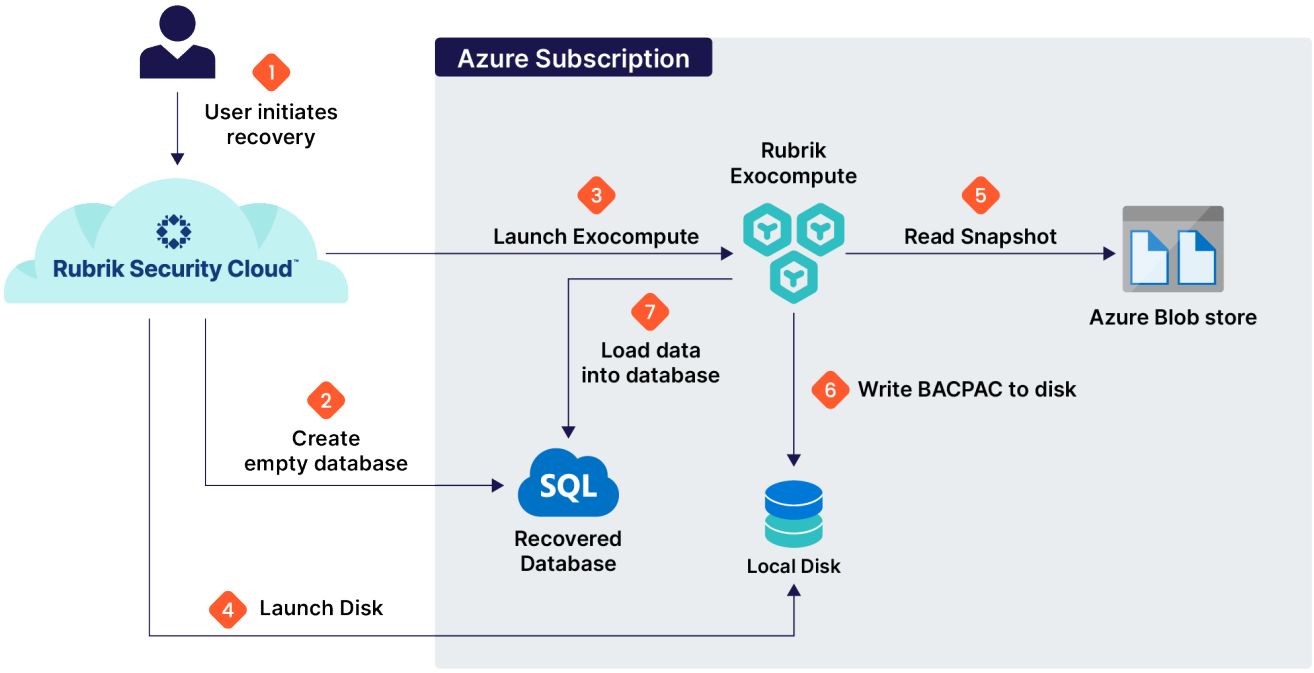

Figure 16 – Azure SQL Immutable Backup - Upload BACPAC

SUMMARY

This concludes How it Works: Cloud-Native Protection for Microsoft Azure SQL. This document explained the core components, architecture, and value proposition of how Rubrik Security Cloud protects Microsoft Azure SQL.

For additional information, please go to https://www.rubrik.com or reach out to your Rubrik Account Team.

VERSION HISTORY

Version	Date	Summary of Changes
1.0	April 2023	Initial Release

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 / 20230406