



TECHNICAL WHITE PAPER

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

Rubrik Security Cloud

Brian Mislavsky & Chris Lumnah
RWP-0618

Table of Contents

- INTRODUCTION 3**
- Audience 3
- Objectives 3
- Azure SQL automated backups 3
 - PiTR backups 4
 - Long-term Retention Backups 4
- Challenges 4
 - Cloud Native Protection Limitations 4
 - Azure SQL Backup Limitations 5
- The Rubrik Approach 6
 - Multi-Cloud Protection 6
 - Enhancing Azure SQL protection with Rubrik Immutable Backups 7
- ARCHITECTURE AND COMPONENTS 8**
- High-Level Architecture 8
 - Azure SQL Automated Backup Orchestration 8
 - Rubrik Immutable Backups 9
- HOW IT WORKS 10**
- Authorization 10
- Configuration 13
 - Azure SQL Automated Backup Orchestration 14
 - Rubrik Immutable Backups 14
 - Azure SQL Database 14
 - Azure SQL Managed Instance 15
- Protection 15
 - Protecting Azure SQL without Immutable Backups 15
 - PiTR & LTR backups 15
 - PiTR & LTR restores 16
 - Protecting Azure SQL with Immutable Backups 18
 - Components 19
 - Taking an Immutable Backup 21
 - Immutable Backup Recovery 25
- SUMMARY 28**
- VERSION HISTORY 28**

INTRODUCTION

Welcome to *How It Works: Cloud-Native Protection for Microsoft Azure SQL*. The purpose of this document is to help readers familiarize themselves with the features, architecture, and workflows for 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 designed for anyone seeking to gain a deeper understanding of Rubrik Security Cloud's Microsoft Azure SQL protection and the technical architectures that underpin these capabilities. This includes architects, engineers, database administrators (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 technical reference regarding the architecture and workflows used 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 the protection of Microsoft Azure SQL?
- How is the protection of Microsoft Azure SQL architected, and why?
- How does the protection of Microsoft Azure SQL protect against Ransomware and assist with Incident Response?
- How does the protection of Microsoft Azure SQL with Rubrik Security Cloud compare to alternate solutions?

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 across different levels of redundancy chosen by the customer to meet business needs, and each comes with a unique set of features and limitations.

This section provides a brief overview of core concepts related to Azure SQL automated backups. 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 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 weekly, monthly, and yearly, depending on the backup policy configured.¹

CHALLENGES

Cloud Native Protection Limitations

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

Public cloud providers are responsible for the protection and availability of the cloud; however, it remains the customer's responsibility to protect their resources **within** the cloud. This means, practically speaking, that it is ultimately the customer's responsibility to protect their applications and data running in a public cloud, regardless of the provider. The [Shared Responsibility Model](#), published by Microsoft, is a valuable reference point 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 may seem simple, selecting the appropriate solution is challenging.

Customers might be inclined to lift and shift legacy tooling into the cloud in hybrid or multi-cloud environments. Unfortunately, this approach often hinders the agility and elasticity that enterprises seek when adopting a cloud strategy.

¹ <https://learn.microsoft.com/en-us/azure/azure-sql/database/long-term-retention-overview?view=azuresql>

The alternative, leveraging platform-native tooling from the cloud provider itself, can be similarly flawed, as this segments data protection operations between public cloud providers and public and on-premises environments. Such an approach leads to significant headwinds regarding 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, any of its PiTR backups are also deleted. If LTR backups have been enabled for a database, they can be restored to a different server within 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 Database and Azure SQL Managed Instances have limitations regarding 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, a Recovery Point Objective (RPO) of 1 hour and a Recovery Time Objective (RTO) of up to 12 hours may be required to recover data from the paired region. Additionally, Geo-Restore only comes into play during a regional outage in the source region. Given the higher-than-normal demand during a regional outage, the failover region may not have sufficient resources to support every recovery immediately.

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 data deletion or ransomware, 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 builds on Azure SQL's built-in backup offering, providing users with greater flexibility, protection granularity, and recovery options. Rubrik customers can now overcome the limitations of Azure SQL's built-in data protection 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 the 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 or to Rubrik Cloud Vault (RCV)

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, homogeneous data management experience across platforms, reducing the drag associated with legacy tooling and point solutions.

Consolidated Reporting – Easily track SLA Domain assignment, protection, 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 crucial to a company. Collecting this information, implementing it, and staying compliant with the SLA are usually tedious and challenging processes. Rubrik Security Cloud utilizes global SLA Domains, a declarative, policy-driven framework, to simplify achieving your SLAs.

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 environments. This enables a single set of policies to be applied across all data locations within 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 enable customers to utilize existing provisioning and governance logic to apply the appropriate SLA Domains across Azure tenants, subscriptions, and regions without requiring 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 are also deleted, preventing recovery using Point-in-Time Recovery (PiTR) backups, as they are stored on the server. This is not an issue if optional LTR backups have been configured via Rubrik Security Cloud. Rubrik Security Cloud utilizes an Azure Storage Account to provide persistent storage for protecting Microsoft Azure SQL databases. This allows backups taken by Rubrik Security Cloud to persist even if the server on which the database was housed 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 and 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 decoupled from the Azure SQL service, they can recreate or recover an SQL database regardless of 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 or RCV, with retention defined by a customer's defined Service Level Agreements (SLAs), this is a simplified and more cost-effective way to handle long-term retention, as the behavior remains the same regardless of the database tier 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 or Rubrik Cloud Vault (RCV).

ARCHITECTURE AND COMPONENTS

HIGH-LEVEL ARCHITECTURE

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

Azure SQL Automated Backup Orchestration

The following workflow outlines the high-level steps in protecting an Azure SQL Database or Managed Instance, which 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 principle 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 or 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. Customers leverage Rubrik Security Cloud to configure and assign Service Level Agreements (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 and 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 databases or 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 and 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.
 - a. Azure SQL DB uses the Azure Database Copy process to create a transactionally consistent database copy.
 - b. Azure SQL Managed Instance leverages Change Data Capture (CDC) to create a transactionally consistent database backup.
3. Customers then utilize Rubrik Security Cloud to configure and assign SLAs for discovered Azure SQL Databases or Managed Instances.
4. When a backup is triggered based on an SLA, Rubrik Security Cloud requests resources from Rubrik Exocompute. Once Exocompute resources are available, a temporary disk is attached to the Exocompute node.
5. An Exocompute task is launched to take a database backup and store it in BACPAC format on the attached local disk.
6. For Azure SQL Managed Instances, when the backup finishes, another exotask is launched to fetch CDC data from the database BACPAC file, and the 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 in this document, protecting Microsoft Azure SQL DBs and Managed Instances consists of 3 steps: *Authorize*, *Configure*, and *Protect*. This document provides a deeper dive into the Authorize & 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 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

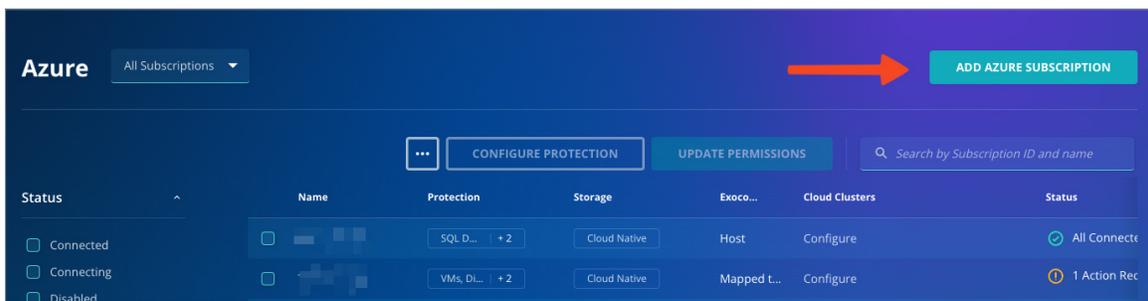


Figure 2 – Adding a Microsoft Azure Subscription - Launch Add Cloud Account Wizard

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

The screenshot shows a window titled "Add Cloud Account" with a close button (X) in the top right corner. Below the title bar, the heading "Select use case" is displayed. There are two main sections:

- Storage:** This section is currently unselected. It contains the text "Replicate or archive data to cloud storage locations." and two radio button options: "Data Center Location" (selected) and "Cloud-native Location" (unselected).
- Protection:** This section is selected, indicated by a blue vertical bar on the left and a blue circle next to the heading. It contains the text "Protect the following across regions in your Azure subscriptions." and four checkbox options:
 - Azure virtual machines, managed disks, and applications
 - Azure SQL databases
 - Azure SQL managed instances
 - Exoccompute

At the bottom right of the window, there are two buttons: "BACK" and "NEXT".

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

The wizard then guides the customer with logging into 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 Exoccompute 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, as well as 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).

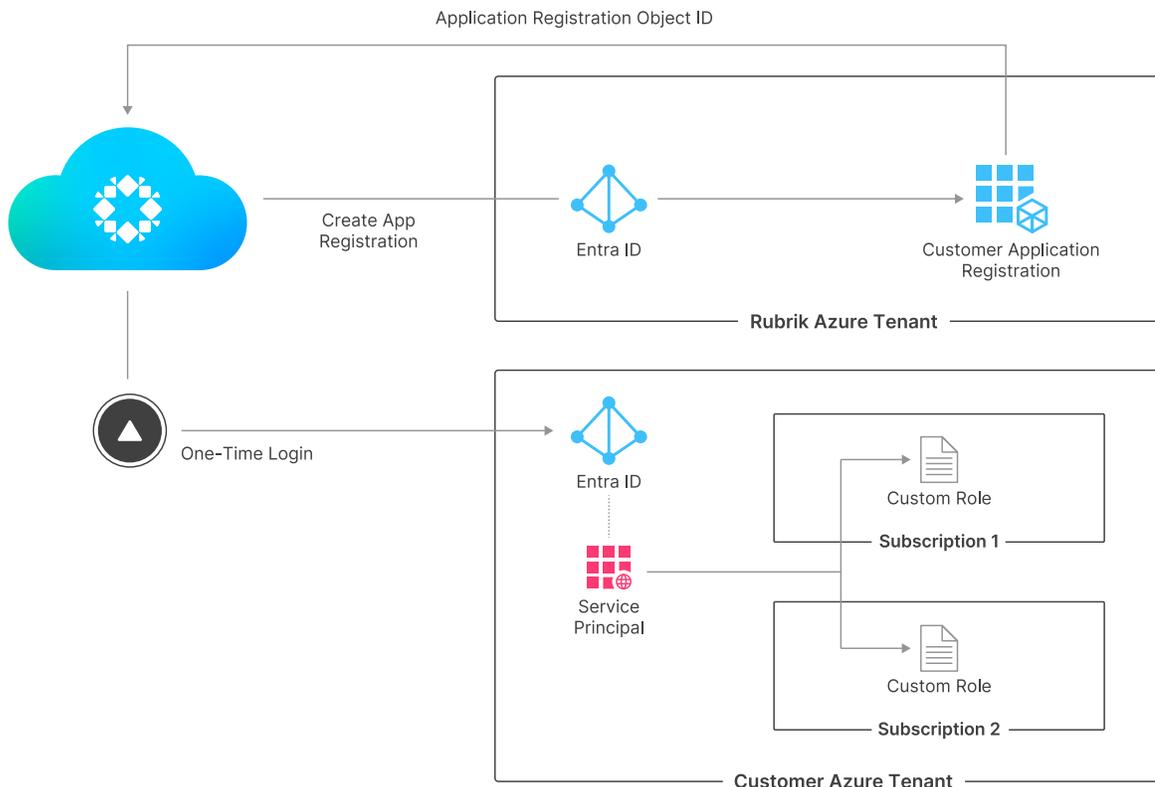


Figure 4 – Rubrik Security Cloud Application Registration Workflow

In Microsoft Entra ID, there are two representations of applications – application objects, also known as Application Registrations, and Service Principals, also called Enterprise Application Registrations. Application objects describe an application to Microsoft Entra ID. They 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 govern an application connecting to Microsoft Entra ID, which can be considered the 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 via the custom roles assigned to the service principal in each subscription. This architecture essentially establishes a type of trust between the customer's Microsoft Entra ID and Rubrik's,

enabling Rubrik Security Cloud to interact with Azure SQL APIs once it authenticates into 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 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 and 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 immutable 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 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.

Backup Type	Description	Recovery
Point-in-time (PiTR) backups	PiTR 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 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 who want to leverage the included basic automated backup functionality of Azure SQL, Rubrik Security Cloud utilizes Microsoft Azure APIs for Azure SQL DB and Azure SQL Managed Instances to configure the 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.

Rubrik Immutable Backups

Adding an Azure Subscription is required for both Azure SQL Database and Managed Instances. The wizard will walk you through the process and 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 Rubrik Immutable Backups.

AZURE SQL DATABASE

The Azure Subscription onboarding wizard will ask for 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 immutable Azure SQL backups](#).

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 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 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.

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 protection hierarchy for Azure SQL Managed Instances can be found [here](#).

Rubrik Security Cloud handles batching all snapshot jobs to prevent overrunning the API limits on Azure with a single large batch of snapshot or replication activities. By default, Rubrik Security Cloud runs a maximum of 20 protection jobs in parallel per protected object type (e.g. Azure SQL Database). Let's dig into 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 automatically begin scheduling and snapshotting Azure SQL Databases and Managed Instances per the SLA Domains created and assigned, eliminating the need to schedule any 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: As this protection method leverages native Microsoft Azure SQL backups to protect data, 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.

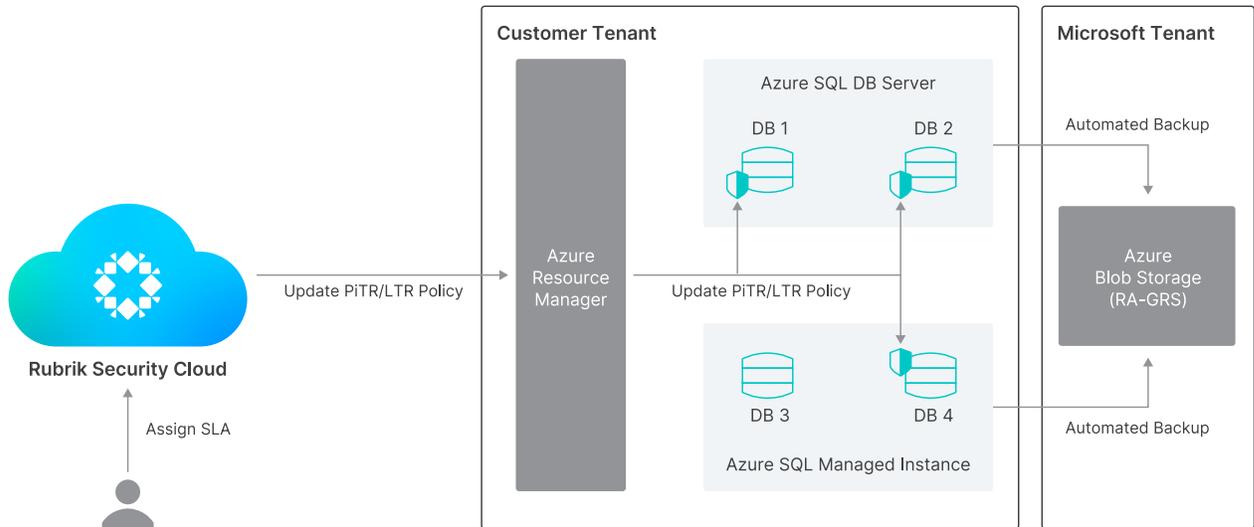


Figure 5 – Azure SQL backups without Immutable Backups

When a user assigns an SLA corresponding to a Short Term Retention to an Azure SQL Database, 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 similar process to configuring backups.

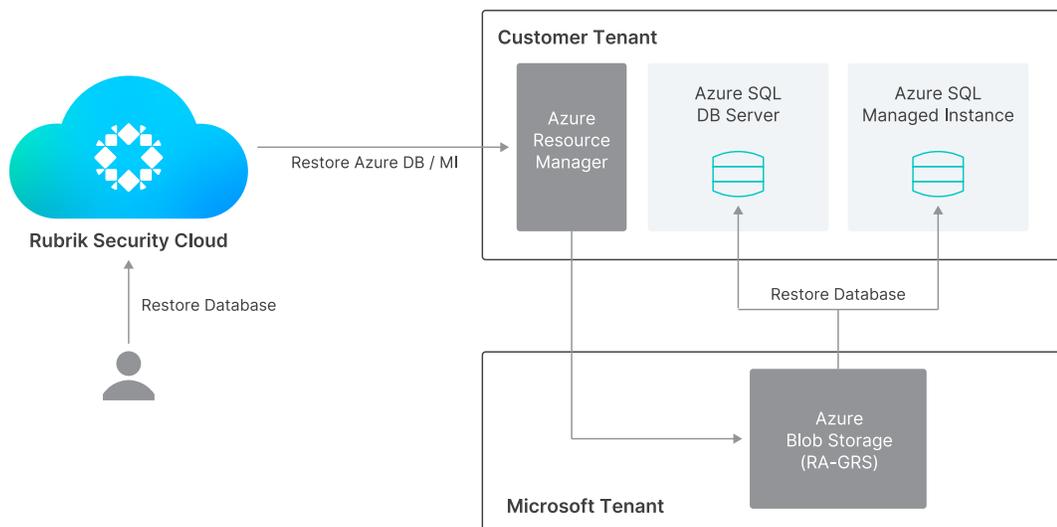


Figure 6 – Azure SQL recovery without Immutable Backups

When not using Immutable Backups, customers leverage RSC to view backups 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 occur.

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 a list of APIs used) would be relatively static for a given day, there is no additional benefit in keeping this data in RSC. The PiTR database recovery workflow is depicted in Figure 7 below.

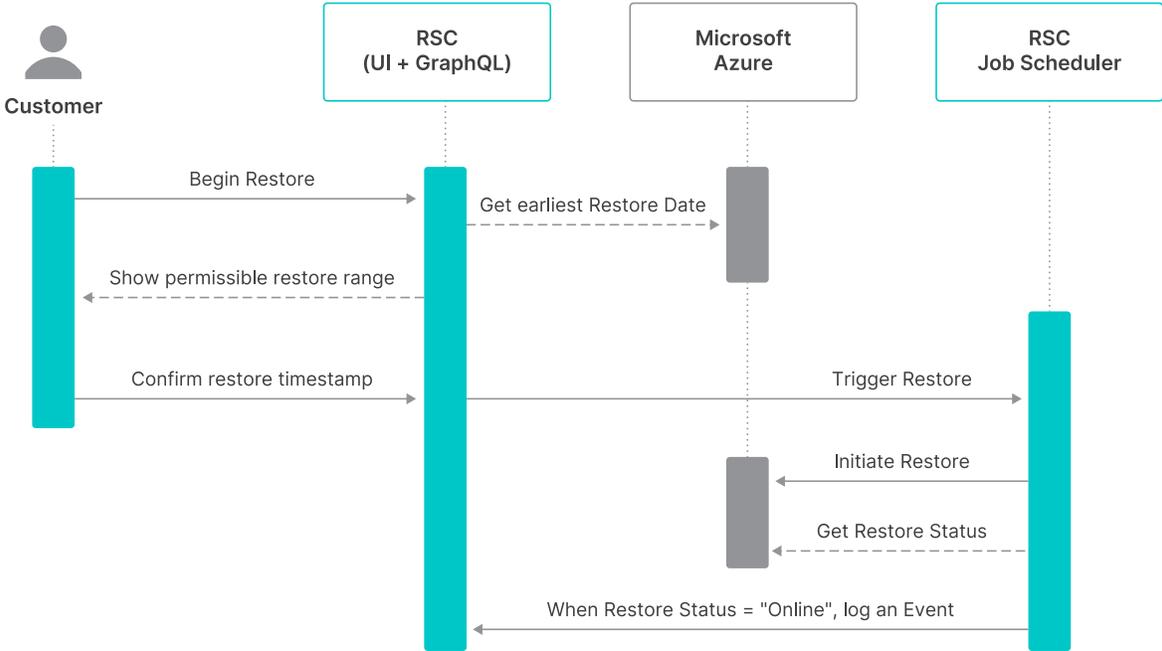


Figure 7 – Azure SQL database recovery flow - PiTR restore

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 will increase 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.

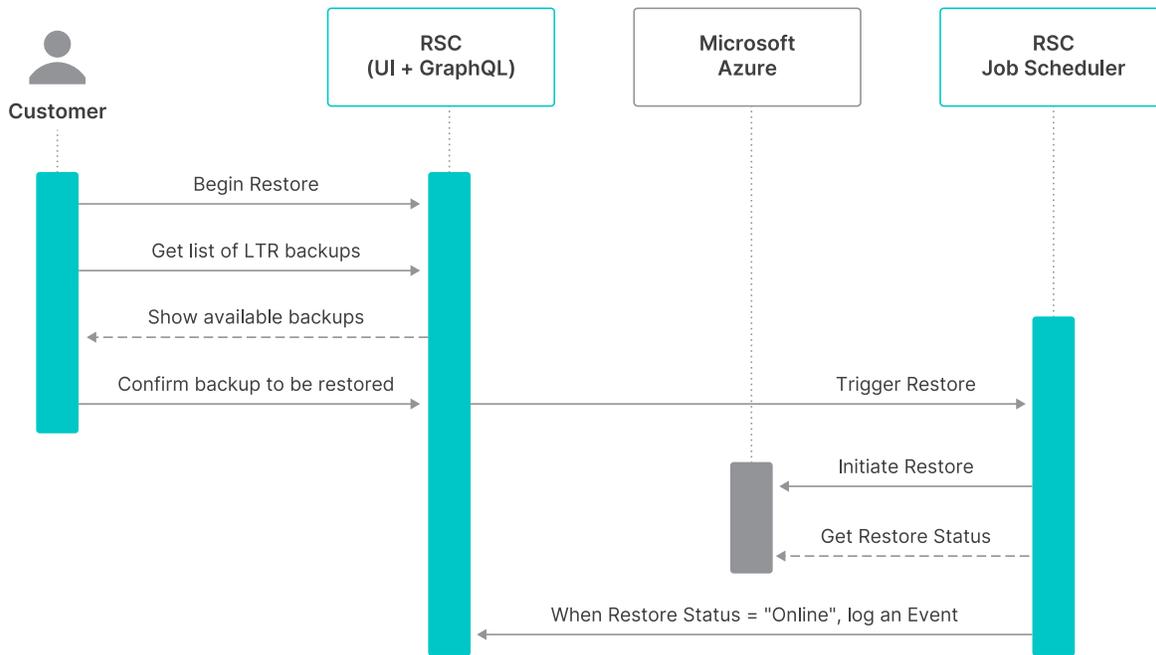


Figure 8 – Azure SQL recovery flow - Long-term retention recovery

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 <code>CreateMode</code> set to <code>RestoreLongTermRetentionBackup</code>)	Managed Databases – Create or Update (with <code>ManagedDatabaseCreateMode</code> set to <code>RestoreLongTermRetentionBackup</code>)
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

To provide persistence to backups of Azure SQL, Rubrik leverages some 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.

For Azure SQL Immutable Backups, Rubrik leverages Exocompute resources to perform database exports using the [SQLPackage](#) utility.

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 as part of their Azure SQL Database cloud account onboarding.

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.

Currently, we whitelist the IP (0.0.0.0), which enables all Azure services to connect to the SQL server.

We are not setting up a service or private endpoint on the SQL server. These Rubrik-managed SQL servers are established during the first backup job per subscription per 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 lead to increased data transfer costs.

Change Data Capture (CDC) & Local Database User

For Azure SQL Managed Instances, RSC leverages a local database user to perform backups. Additionally, to ensure transactional consistency of the backups, Rubrik requires that CDC be enabled for every database protected with Immutable Backups. Enabling CDC is a one-time action that is done before enabling immutable backups.

Rubrik provides three options for creating a local user and enabling 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 Microsoft Entra ID 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 to take 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 Exocompute, utilizing the credentials in the customer subscription, is secured using TLS 1.2, with each task using short-lived TLS certificates and keys.
- **Logging** – Rubrik leverages a custom code function to handle secrets that ensure that credentials used throughout Rubrik Security Cloud are not logged, preventing accidental revealing of secrets.

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 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 made 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 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:

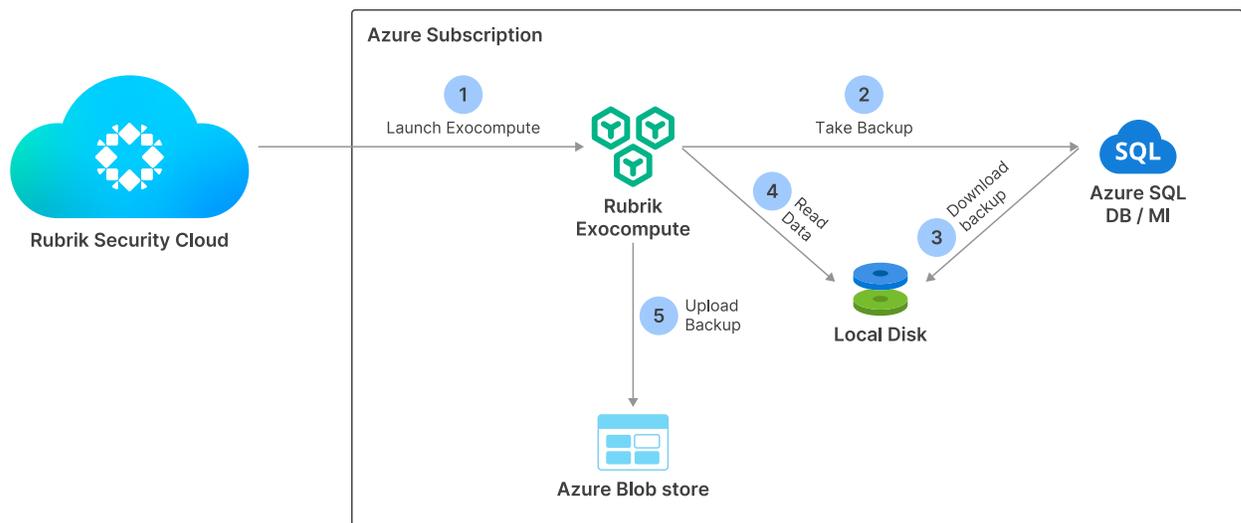


Figure 9 – Immutable Backups - high level steps

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. For Azure SQL DB, a database copy is made on the Rubrik-managed SQL Server instance.
3. Once Exocompute is available, launch a task to take the backup and launch an empty disk to store the backup.
4. Database backup is taken in BACPAC format and stored on a local disk.
5. For Azure SQL Managed Instances, CDC changes are collected from the database.
6. Database backup metadata is read and stored in the RSC database.
7. Database backup data is ingested into Azure Blob Storage and made immutable according to the SLA.
 - a. Azure SQL DB: BACPAC file
 - b. Azure SQL Managed Instance: BACPAC file + CDC changes

There are two primary 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 snapshots of the databases and involves the following steps:

Azure SQL Database

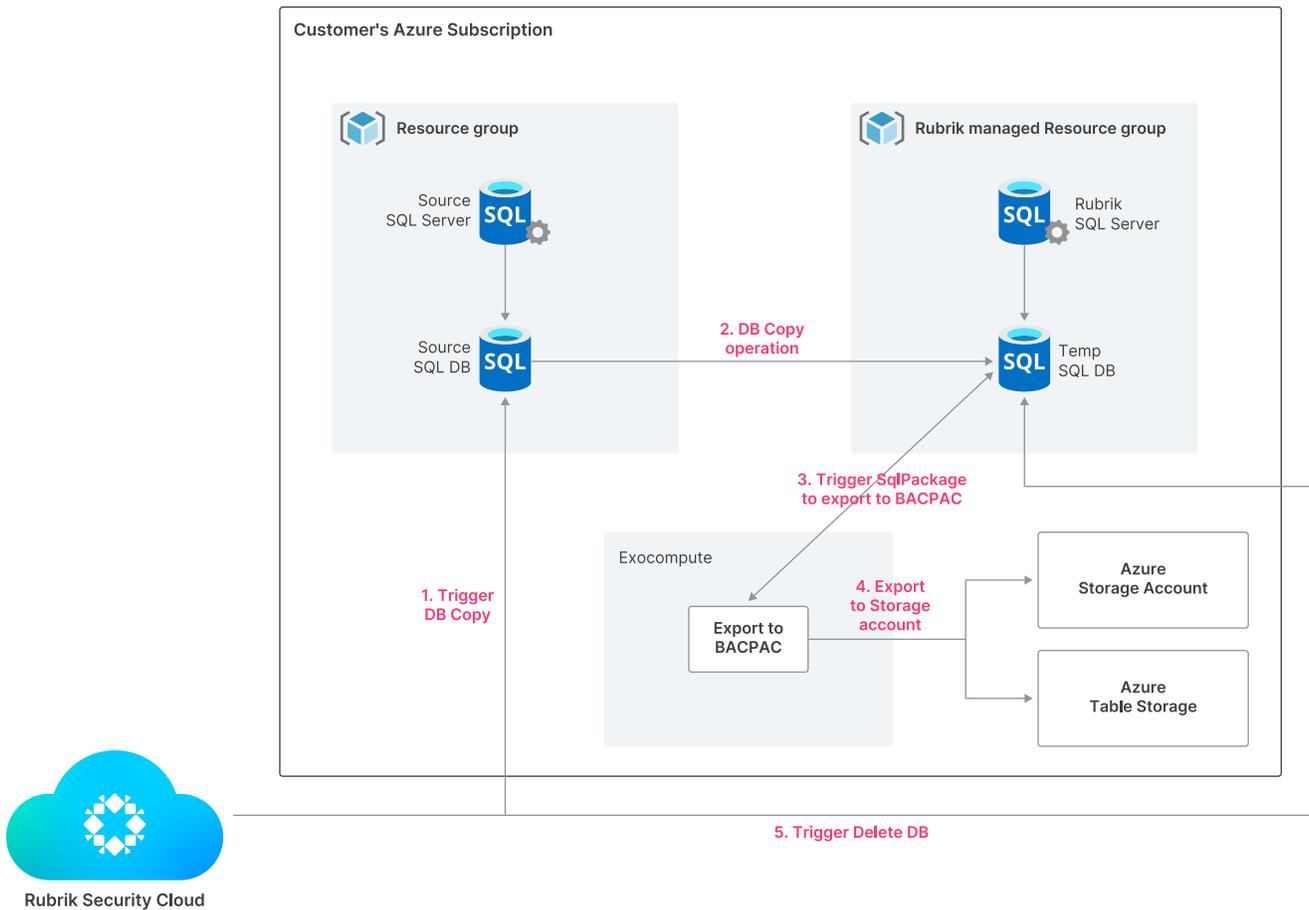


Figure 10 – Backup layer steps

1. Trigger a database copy to the Rubrik-managed SQL server instance.
2. Database export as a BACPAC file starts using the SQLPackage binary
3. Call the ingestion layer to ingest the backups to Azure storage.
4. Remove the database from the Rubrik-managed SQL server instance.

Azure SQL Managed Instance

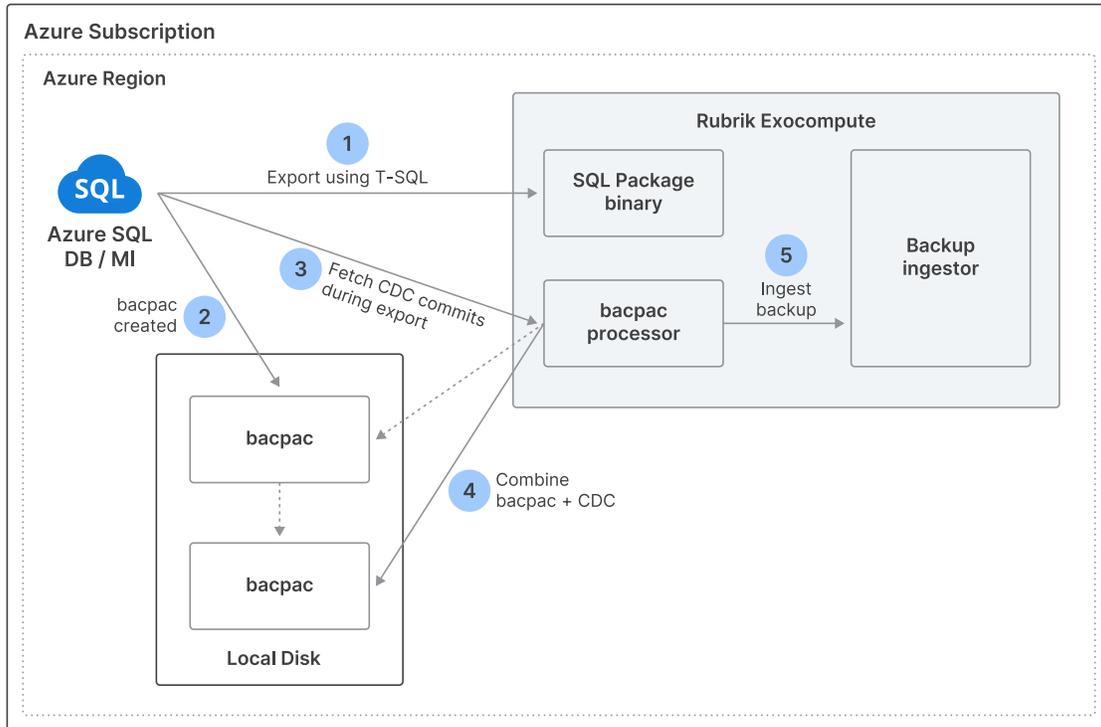


Figure 11 – Backup layer steps

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

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



Figure 12 – Azure SQL Immutable Backup - BACPAC repair process

1. Determine the 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 and storing the backup. As [previously mentioned](#), Rubrik leverages its [CSL](#) for persisting Azure SQL backups.

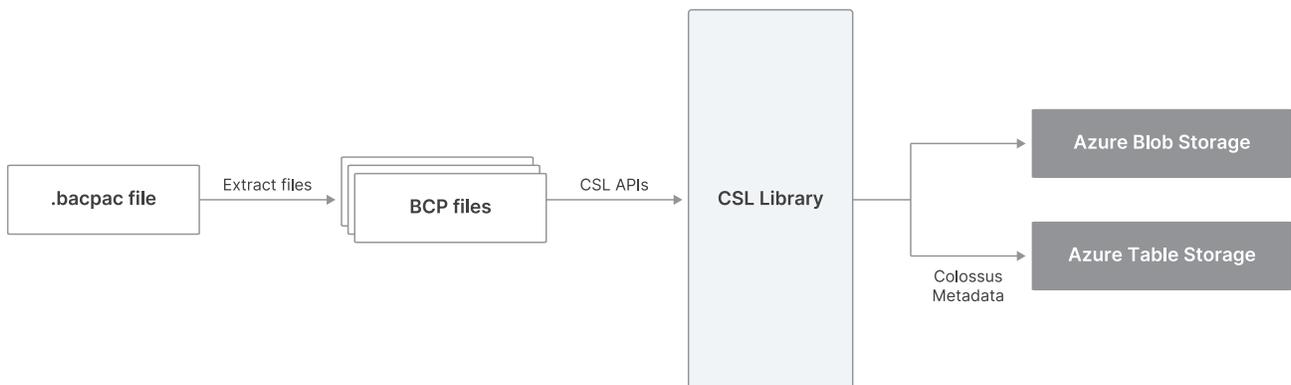


Figure 13 – Azure SQL Immutable Backup - Ingestion Layer

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, leading 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.

Like taking an Azure SQL Immutable backup works, recovery involves the Retriever Layer and the Recovery Layer.

Retriever Layer

The retriever layer retrieves the requested snapshot from Azure Blob storage. As shown in *Figure 14* below, the CSL reader process will fetch the data corresponding to the requested snapshot and write it to the local disk as a BACPAC.

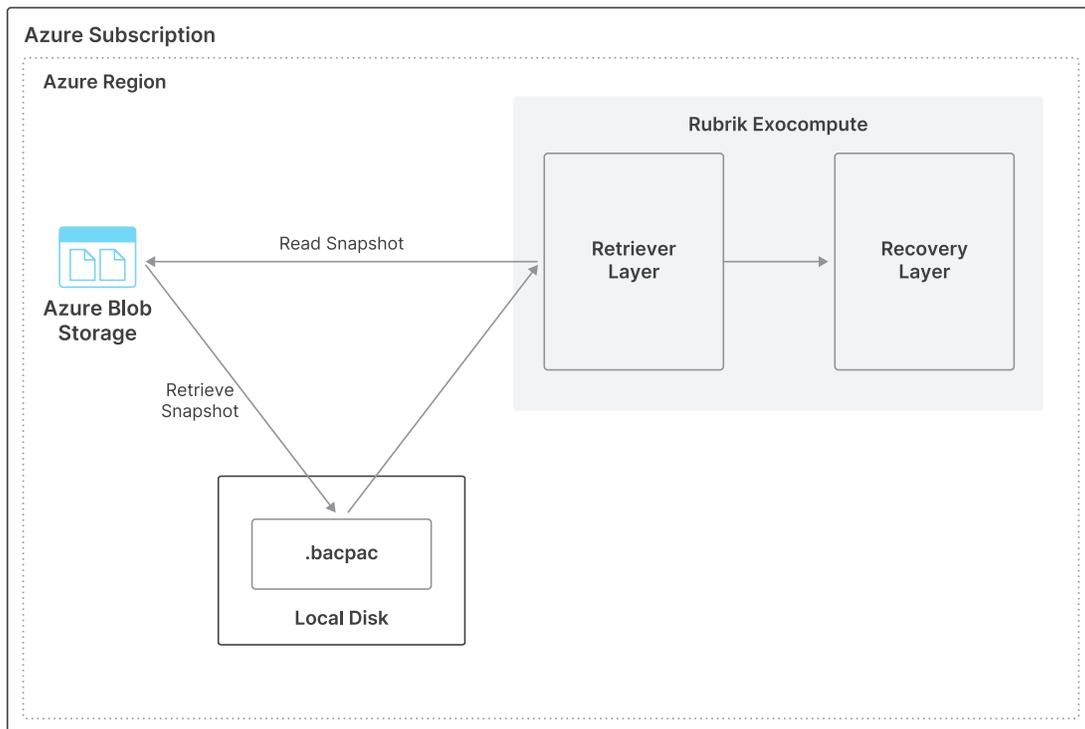


Figure 14 – Azure SQL Immutable Backup - Recovery - Retriever Layer

Recovery Layer

The recovery layer of the restore process takes the BACPAC file created by the retriever layer and recovers its snapshot data. As mentioned, this is performed in one of two ways, depending on the method of recovery selected by the customer.

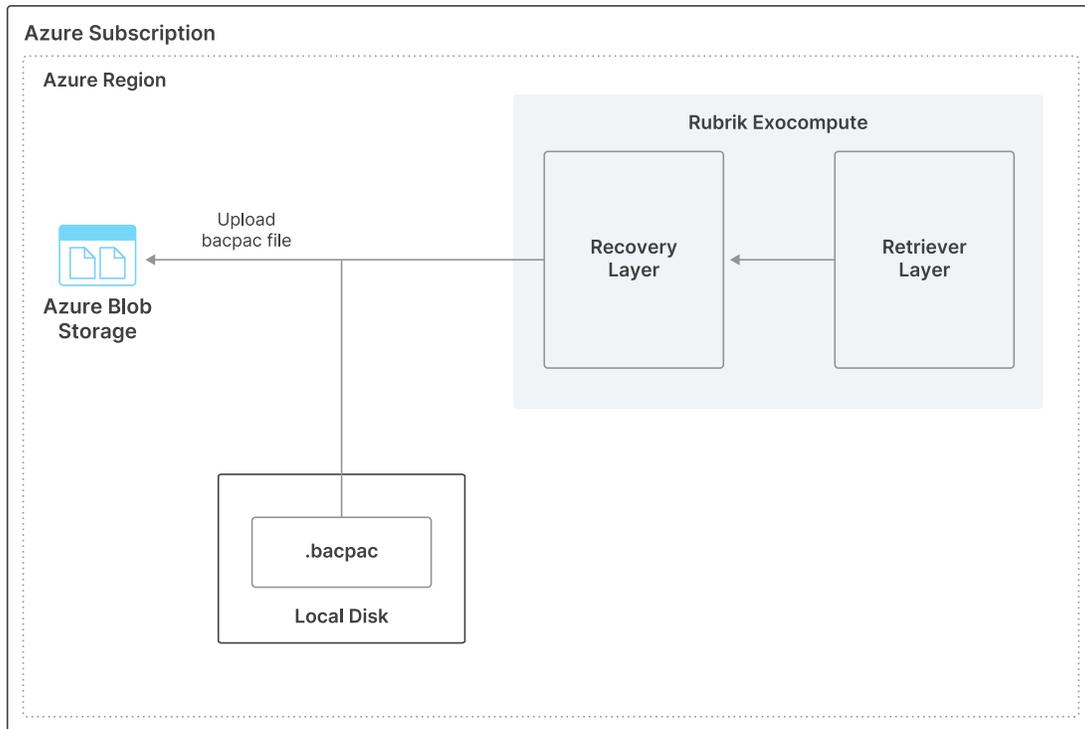


Figure 15 – Azure SQL Immutable Backup - Recovery Layer

Upload BACPAC

When selecting the Upload BACPAC method, a customer is presented with 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 gives the customer a link to download the file. The specific steps for this are as follows:

1. The user initiates the recovery workflow
2. Rubrik Exoccompute 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 made immutable

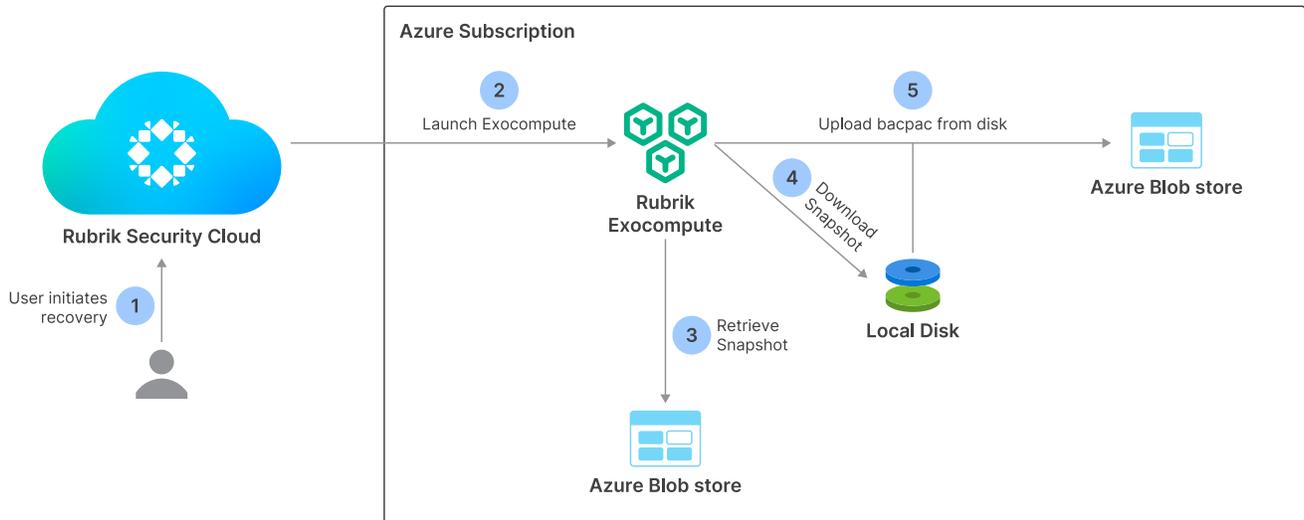


Figure 16 – Azure SQL Immutable Backup - Upload BACPAC

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.

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

1. The user initiates the recovery workflow
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

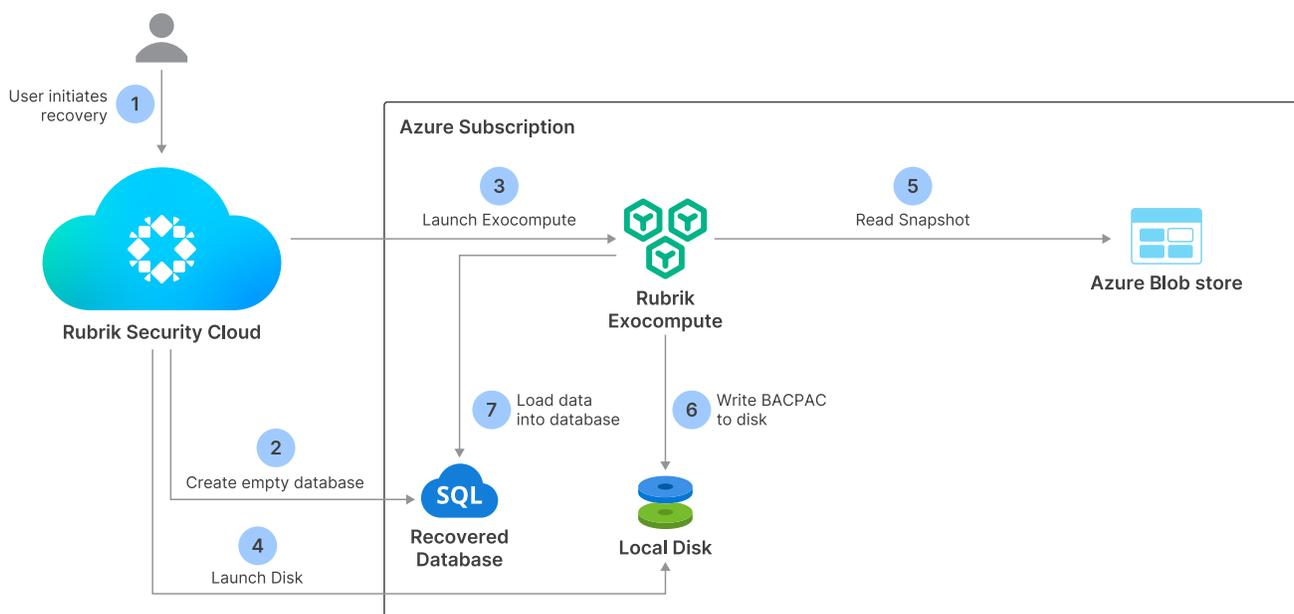


Figure 17 – Azure SQL Immutable Backup - Recover to Database

SUMMARY

This concludes *How it Works: Cloud-Native Protection for Microsoft Azure SQL*. This document explained 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



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

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

Rubrik (NYSE: RBRK) is on a mission to secure the world's data. With Zero Trust Data Security™, we help organizations achieve business resilience against cyberattacks, malicious insiders, and operational disruptions. Rubrik Security Cloud, powered by machine learning, secures data across enterprise, cloud, and SaaS applications. We help organizations uphold data integrity, deliver data availability that withstands adverse conditions, continuously monitor data risks and threats, and restore businesses with their data when infrastructure is attacked.

For more information please visit www.rubrik.com and follow @rubrikinc on X (formerly Twitter) and Rubrik on LinkedIn. Rubrik is a registered trademark of Rubrik, Inc. All company names, product names, and other such names in this document are registered trademarks or trademarks of the relevant company.