TECHNICAL WHITE PAPER

How It Works: Cloud Native Protection for Amazon Relational Database Service (RDS)

Bill Gurling
September 2023
RWP-0552

Table of Contents

INTRODUCTION

Welcome to *How It Works: Cloud-Native Protection for Amazon Relational Database Service (RDS)*. The purpose of this document is to aid the reader in familiarizing themselves with the features, architecture, and workflows of Rubrik's Cloud-Native Protection for Amazon RDS. 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 Cloud-Native Protection for Amazon RDS on the Rubrik Security Cloud platform and the technical architectures that underpin those capabilities. This includes architects, engineers, and administrators responsible for AWS infrastructure, databases, 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 architecture and workflows utilized by Cloud-Native Protection for Amazon RDS on Rubrik Security Cloud. After reading this document, the reader should be able to answer the following questions regarding Cloud-Native Protection for Amazon RDS:

- What does Cloud-Native Protection for Amazon RDS do?
- What problem(s) does the Cloud-Native Protection for Amazon RDS solve?
- How does one configure and utilize Cloud-Native Protection for Amazon RDS?
- How is Cloud-Native Protection for Amazon RDS architected? Why?
- How does Cloud-Native Protection for Amazon RDS operate?
- How does Cloud-Native Protection for Amazon RDS compare to alternate solutions?

CHALLENGES

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 AWS 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, it is in fact quite complex.

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.

The need for an alternate approach is clear.

THE RUBRIK APPROACH

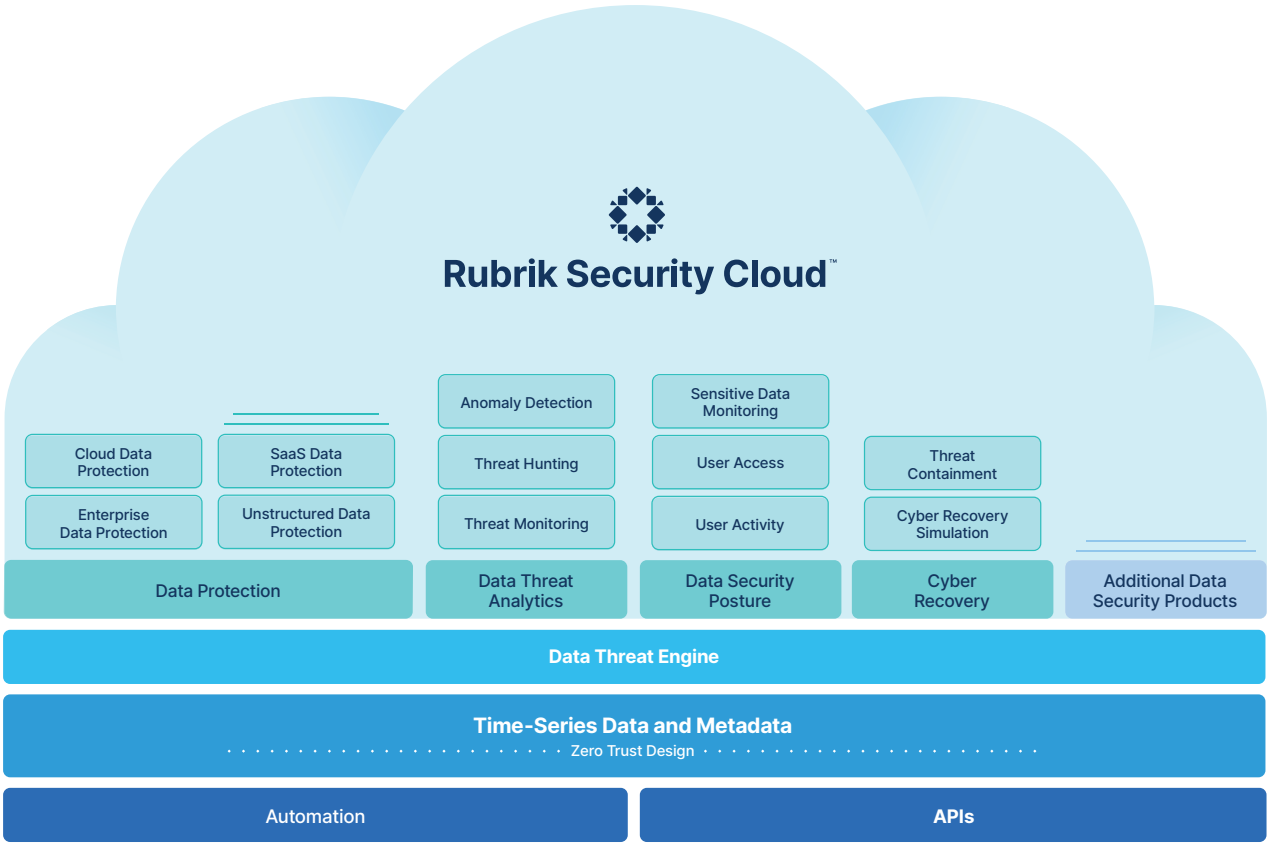

Figure 1 – Rubrik Security Cloud Multi-cloud Protection

Rubrik's goals are to collapse the footprint of legacy data protection solutions, simplify and automate backup and recovery using policy-based protection, minimize data loss, and to streamline operations. Rubrik's Cloud-Native Protection for Amazon RDS reduces painful manual job scheduling. Cloud-Native Protection for Amazon RDS is a Software as a Service (SaaS) based data protection platform that provides automated backup and recovery schedules across regions, and even across clouds with a single global policy engine. It also allows the customer to quickly find and recover snapshots with predictive search. This solution allows Rubrik customers to reap the benefits of rapid innovation and reduced management complexity with data protection delivered as a service.

Protecting RDS workloads with Rubrik Security Cloud consists of 3 primary steps.

Step	Detail
Authorize	Authorize Rubrik Security Cloud to access the AWS Account(s) that require protection via an AWS CloudFormation integrated workflow that aligns with AWS security best practices.
Configure	Use a single, declarative SLA policy engine to automatically configure point-in-time recovery, as well as create and expire RDS snapshots to suit backup and replication requirements.
Protect	Assign SLA policies to the instances that require protection. Automatic protection for newly created databases in AWS accounts ensures workloads are protected when they are provisioned. Recover instances rapidly through the Rubrik Security Cloud UI or API. Rubrik Security Cloud acts as a single pane of glass for hybrid or multi-cloud deployments.

KEY FEATURES

The key features of Rubrik's Cloud-Native Protection for Amazon RDS include:

- Unified data management across regions, accounts
- Automated global data protection via Rubrik Security Cloud SLA policies
- Rapid recovery for RDS instances or clusters within or across regions
- Replication and logical air gapping for increased security and data resiliency

Unified Data Management Across Accounts and Cloud Platforms

Single Point of Management and Automation via Rubrik Security Cloud – Rubrik's Cloud SaaS platform is a single point of management and automation for hybrid and multi-cloud environments. It requires no persistently running instances or compute in the customer's AWS environment. Rubrik Security Cloud provides a simple, homogenous end-user data management experience across platforms and reduces the drag associated with legacy tooling and point solutions. For Amazon RDS, this includes unified administration for point recovery and long term retention.

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

Automated Global Data Protection via Rubrik Security Cloud SLA Domains

SLA Domains – In the data protection world, Service Level Agreements (SLAs) define protection levels for workloads, availability targets, and objects 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 uses global SLA Domains, a declarative, policy-driven framework, to make achieving your SLAs easier. When using an SLA Domain for Cloud-Native Protection for Amazon RDS, it is comprised of four components:

- Snapshot Frequency
- Snapshot Retention
- Replication Target
- Log Backup Retention

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

Account Level Auto-Protection – Assign SLA policies to entire AWS accounts and ensure that every instance provisioned into the protected regions in those accounts receives the required level of data protection without the need for explicit SLA assignment. Account level SLA Domains can be overridden using tag-based assignment or by directly assigning SLA Domains to instances.

Tag-Based Auto-Protection – Allows for the assignment of SLA policies to instances whenever a specific tag key, or key value pair is found on any instance in any AWS account in scope. These tag rules allow customers to leverage existing provisioning and governance logic to apply the appropriate SLA Domains across AWS accounts and regions without the need for manual intervention.

Policy Driven Long Term Retention and Point-in-time Recovery – Cloud Native Protection for Amazon RDS allows customers to configure long term snapshot retention as well as RDS log backup retention within a single SLA Domain Policy; eliminating the need for multi-service and per-instance data protection configurations.

Rapid Recovery for Instances Within or Across Regions

RDS Instance Export – Create a new RDS instance from the selected snapshot or point-in-time. This workflow allows the user to define the database instance identifier, class, storage type, VPC, subnet, security groups as well as other database and network parameters. In addition, if you use a selected snapshot, you can also define the KMS key and region.Tags that were on the source instance at the time of the snapshot can also be included or excluded from the export process.

Replication and logical air gapping for increased security and data resiliency

Rubrik's Cloud Native Protection allows for automated replication of snapshots to a different account and/or region. This provides data resiliency for snapshots in case there is an incident.

Continuous backups coupled with **scheduled** snapshots taken by AWS are stored with the database instance. If an instance is deleted either accidentally or by a bad actor, the backups are also deleted. Once the instance is deleted, there is no way to get the database back. With Rubrik Cloud Native Protection, **manual** snapshots are taken and stored in a separate storage location protected from accidental deletion.

ARCHITECTURE AND COMPONENTS

HIGH LEVEL ARCHITECTURE

Cloud-Native Protection for Amazon RDS allows Rubrik customers to take advantage of the power of the Rubrik's SLA policy engine via the Rubrik Security Cloud SaaS platform to protect RDS workloads inside of their AWS accounts. This allows customers to experience the power of Rubrik without the need to deploy, manage, or patch any long running compute instances in the customer's AWS environment nor on-premises. In order to accomplish this, Rubrik Security Cloud leverages the native snapshot, image creation, and recovery APIs provided by AWS to backup and recover RDS instances.

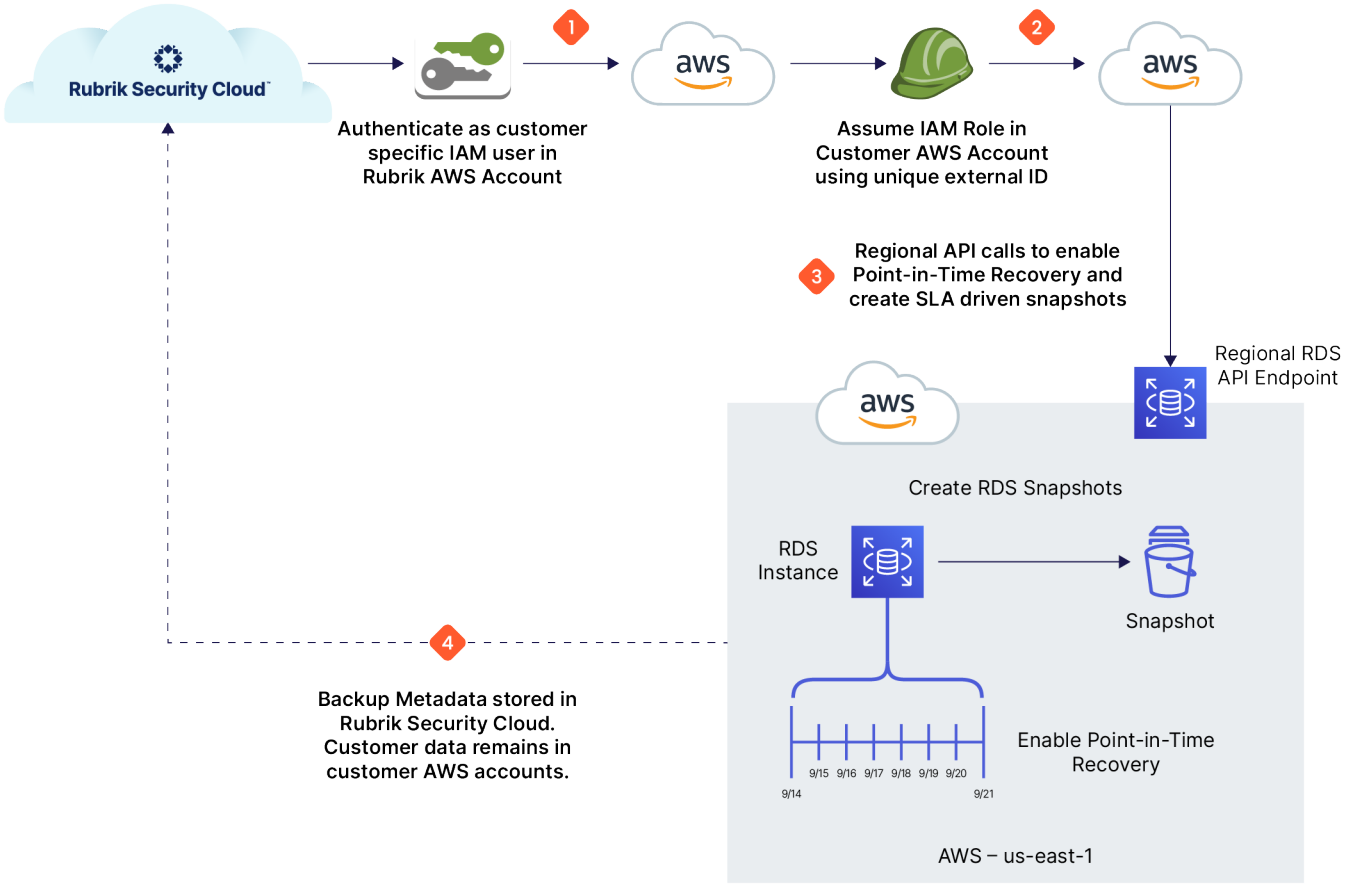

Figure 2 – High Level Architecture: Cloud-Native Protection for Amazon RDS

At a high level, the following workflow is used when protecting Amazon RDS resources in a customer owned AWS Account using Rubrik Security Cloud.

1. Rubrik Security Cloud authenticates into a Rubrik owned AWS account using a customer specific AWS Identity and Access Management (IAM) user. The credentials for this IAM user are stored in an encrypted format within a customer specific database.

2. Rubrik Security Cloud assumes an IAM role that is pre-created by CloudFormation when the customer enables Cloud-Native Protection for their AWS account. This role has the necessary permissions to protect and restore Amazon RDS assets in the customer's environment.

a. In order to utilize this role, AWS requires that API calls come from the trusted Rubrik AWS account.

b. Additionally, these API calls must include an external ID that is specific to the AWS account being protected. This external ID is encrypted, and securely stored in Rubrik Security Cloud, outside of the trusted Rubrik owned AWS account.

3. Rubrik Security Cloud calls the regional API endpoints in the customer's AWS account to enable point-in-time recovery and create RDS snapshots in accordance with the frequency and retention defined in the SLA Domain policies assigned to these assets in Rubrik Security Cloud.

4. Snapshot and point-in-time recovery metadata is securely stored in Rubrik Security Cloud.

COMPONENTS

The solution depicted in *Figure 2* consists of many AWS and Rubrik components. Let's discuss each in more detail as well as describe its role within Cloud-Native Protection for Amazon RDS prior to delving deeper into the architectures and associated workflows.

Cloud-Native Protection for Amazon RDS is configured and managed via Rubrik's Rubrik Security Cloud SaaS platform. When protecting AWS assets, Rubrik Security Cloud assumes an IAM Role from a Rubrik owned AWS account into the customer owned AWS account being protected. This grants Rubrik Security Cloud the permissions required to protect RDS instances in the customer's account. Rubrik Security Cloud leverages the AWS Backup API endpoints in each of the protected AWS region(s) to create RDS Snapshots and manage Point-in-Time Recovery in accordance with the SLA Domain policy applied. These snapshots are automatically created, and expired by Rubrik Security Cloud. Only metadata is sent to Rubrik Security Cloud; the RDS snapshots that contain customer data reside solely in the customer owned AWS account.

HOW CLOUD-NATIVE PROTECTION FOR AMAZON RDS WORKS

As stated previously in this document, protecting AWS workloads with Rubrik Security Cloud consists of 3 primary steps: *Authorize*, *Configure*, and *Protect*. This section of the document dives deeper into each of these steps, both operationally and architecturally. This should leave the reader with a basic familiarity of how Cloud-Native Protection for Amazon RDS on Rubrik Security Cloud is architected, configured, and utilized.

Authorize

Authorizing Rubrik Security Cloud to protect workloads in AWS is a simple process. The customer clicks the **Add AWS Account** button in the **AWS Cloud Accounts** section of **Settings** in Rubrik Security Cloud. The customer selects **Protection** and enters the AWS **Account ID** for the account requiring protection and an **Account Name** to be displayed in the Rubrik Security Cloud console. The customer selects the AWS Regions that should be protected. A wizard in AWS then walks the customer through launching a CloudFormation Stack to create the service principles and permissions required to protect this account with Cloud-Native Protection for Amazon RDS. Let's explore how this workflow operates under the covers. A list of the required permissions are listed in the Rubrik Security Cloud User Guide.

In order to protect workloads running in AWS, Rubrik Security Cloud needs a means by which to interact with the customer's AWS account(s). As stated prior, Rubrik Security Cloud leverages the native snapshotting and image creation capabilities of AWS in order to backup, replicate, and restore the assets it is protecting. These capabilities are available via the AWS API to which access is controlled by the AWS Identity and Access Management (IAM) service. The IAM service itself is quite powerful and supports a variety of service principals

such as users, groups, federated users and groups, as well as roles. Permissions are delegated to or revoked from these service principals via associated IAM Policies. Rubrik Security Cloud leverages IAM roles for RDS native protection.

The figure below depicts how the workflow interacts with a customer's account from an AWS IAM perspective.

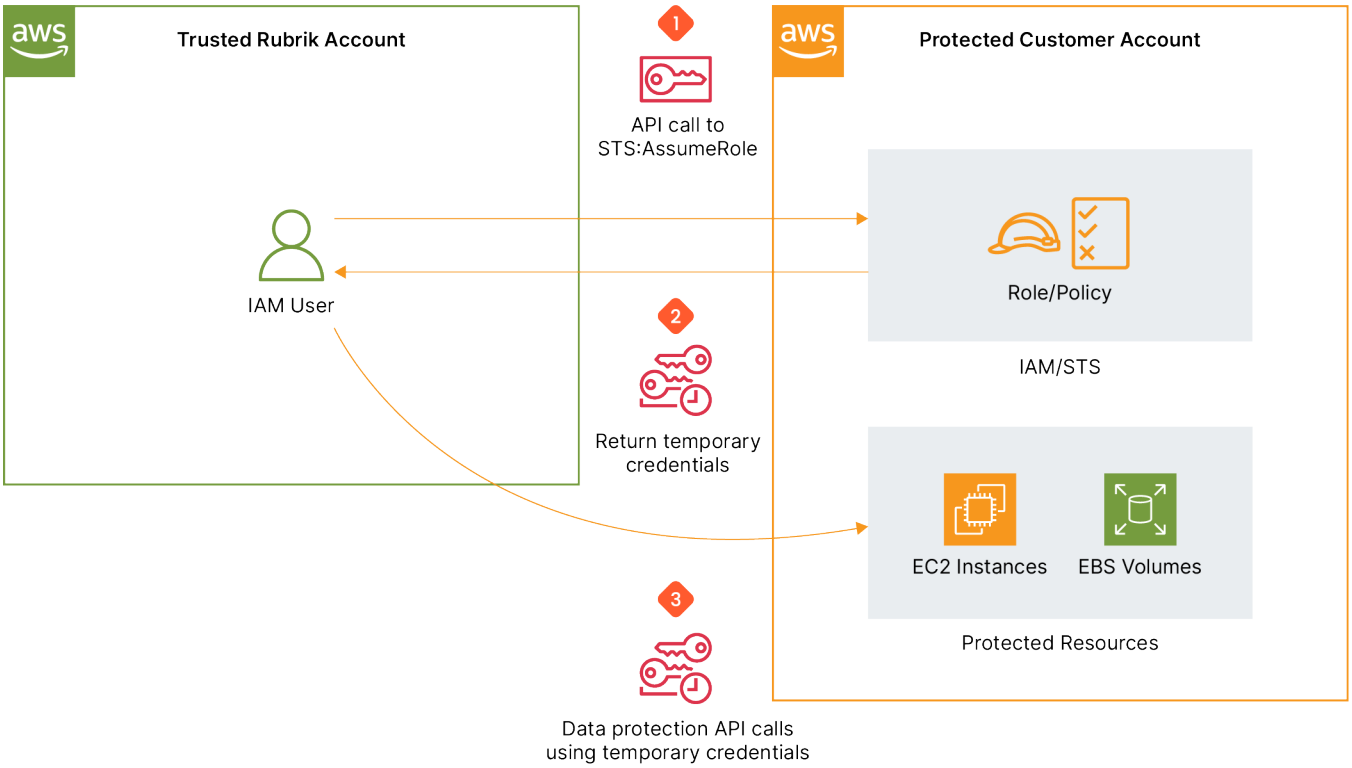

Figure 3 – Rubrik Security Cloud AssumeRole Workflow

Roles are an identity that AWS customers can create in AWS and assign permissions to. Roles can be utilized only by trusted AWS services or accounts. These roles are assumed from a trusted Rubrik owned AWS account via the AWS Security Token Service (STS) API and allow Rubrik to interact with resources in the customer's environment without the need for long lived static credentials. Leveraging roles for this type of workflow is in line with AWS best practices. The alternative, leveraging IAM Access keys assigned to a user, is a significantly less desirable approach which unfortunately, is still frequently employed by many. This is due to the fact that the Access keys are long lived credentials that are not restricted to a trusted entity. If these keys leak, they can be used by anyone, from anywhere.

The role and policy indicated above have to be created and maintained. To accomplish this, Rubrik Security Cloud leverages AWS CloudFormation. AWS CloudFormation allows AWS customers to model infrastructure and application resources with templates written in formats like JSON or YAML. These templates can be submitted to the CloudFormation service on AWS to create a collection of resources known as a stack. These stacks and their life cycles are managed via CloudFormation itself rather than individually in AWS. Update the template in CloudFormation and the stack will be updated accordingly, delete the stack and all resources provisioned as part of the stack will be deleted as well.

Once the role is built in the customer account, Rubrik Security Cloud needs its Amazon Resource Name (ARN) in order to utilize it when interacting with the newly added AWS account. This value is securely transmitted back to Rubrik using Amazon's Simple Notification Service (SNS) thus completing the workflow and allowing Rubrik secure access to the customer's AWS account for data protection purposes. The diagram below depicts this provisioning process.

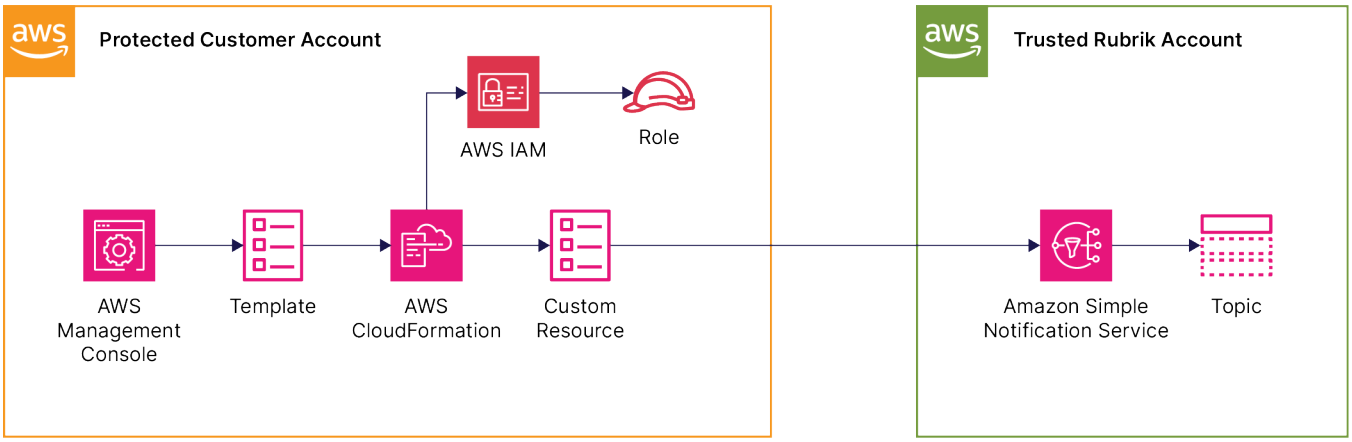

Figure 4 – Rubrik Security Cloud IAM Role Provisioning Process

Once this process is complete Rubrik Security Cloud knows the necessary permissions are in place and has all the information needed to begin protecting the AWS account. Examples of the CloudFormation templates and IAM policies used 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 resources via CloudFormation. As an example, this update process would be used if a customer configured Amazon EC2 protection for an AWS account that already had Amazon RDS protection enabled in Rubrik Security Cloud. The update workflow would merely add the principals and permissions necessary via an update to the existing CloudFormation stack.

Configure

SLA DOMAINS AND CLOUD-NATIVE PROTECTION FOR AMAZON RDS

Once an AWS account is added to Rubrik Security Cloud, the next step is to create one or more SLA Domains in order to begin protecting workloads in AWS. These SLA Domains can then be applied to RDS instances at which time Rubrik Security Cloud will configure point-in-time recovery and begin creating and expiring RDS snapshots to protect the relevant workloads according to the parameters defined in the SLA Domain. SLA Domains are a powerful replacement to the job scheduling approach used by many traditional data protection solutions largely due to their declarative nature which generally maps directly to the RPOs and RTOs required by businesses.

Building an SLA Domain for use with Cloud-Native Protection for Amazon RDS is a straightforward and simple process that is outside of the scope of this document. Please reference the Rubrik Security Cloud User Guide for details on SLA creation within Rubrik Security Cloud. That said, there are a few specific elements of SLA domains relative to Cloud-Native Protection for Amazon RDS that will be highlighted.

Replication can be set up to a different region inside the same account, or cross account and cross region. This will allow a backup to be copied to a secondary region and/or account. Additionally, you can decide how long the replicated backup should live in the secondary location.

When creating a SLA domain within Rubrik Security Cloud, the customer can define the **Log backup for AWS RDS** duration. This parameter is used to define the number of days for which point-in-time recovery will be available on any RDS instance protected with the corresponding SLA domain. This duration can be configured up to 35 days, as allowed by Amazon RDS.

At this point in time the **Archiving** section does not apply to Cloud-Native Protection for Amazon RDS.

Protect

Once all the required AWS accounts have been added to Rubrik Security Cloud and the desired SLA Domains have been created, it's time to begin protecting RDS workloads in AWS. In order to get started, the appropriate SLA Domains need to be assigned to the proper RDS instances. As of this writing, Cloud-Native Protection for Amazon RDS supports protecting all RDS database engines including Aurora, and has a few different options with regards to assigning SLA Domains to RDS instances. Each of them provides a unique business benefit when employed properly.

ACCOUNT LEVEL SLA ASSIGNMENT

Upon navigating to the **Inventory** screen in Rubrik Security Cloud and selecting **AWS – RDS**, users are presented with an inventory of all RDS instances that Rubrik Security Cloud is currently capable of protecting. Selecting the **AWS Accounts** tab will present a view with a list of all AWS accounts added to Rubrik Security Cloud and some relevant information on each account.

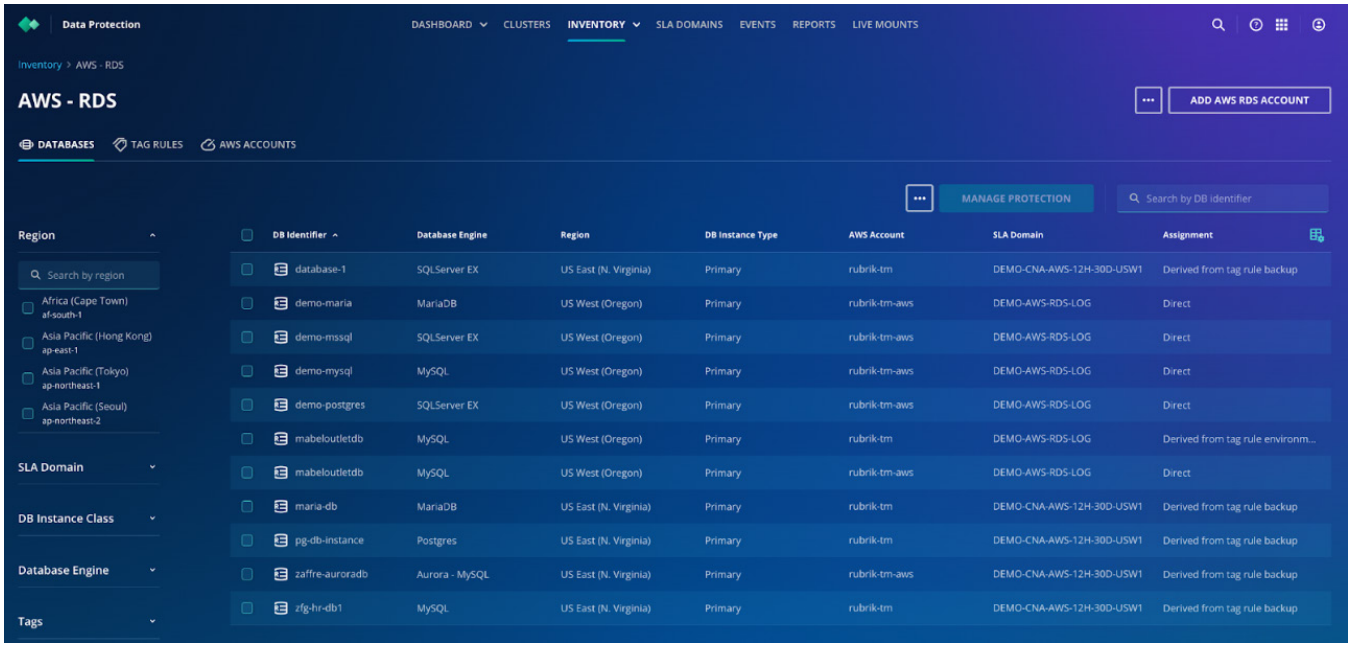

Figure 5 – Cloud-Native Protection for Amazon RDS Inventory

From this view the customer can select one or more accounts using the corresponding checkboxes and use the **Manage Protection** button to modify the SLA Domain assigned to that account. SLA Domains assigned at the account level will automatically inherit down to all RDS instances in all protected regions unless another SLA Domain is assigned via a tag rule or directly to an instance itself. This is a useful technique to automatically protect all instances provisioned into an AWS account with the desired SLA Domain. As an example, one might assign a fairly lightweight SLA Domain to a developer account so that all assets in that account are recoverable regardless of how they were provisioned.

SLA Domains with a yellow exclamation point over them in the SLA Domain selection window have some configuration that is incompatible with Cloud-Native Protection for Amazon RDS for the account(s) that have been selected. Hover over the warning icon to receive a description of the warning.

In addition to assigning an SLA Domain to the account, this dialogue box also allows the customer to select **Clear Existing Assignment** or **Do Not Protect**. Both of these configurations produce a similar result when utilized at the AWS account level. The selected accounts and instances within them would be unprotected unless an SLA Domain was assigned to them using another means.

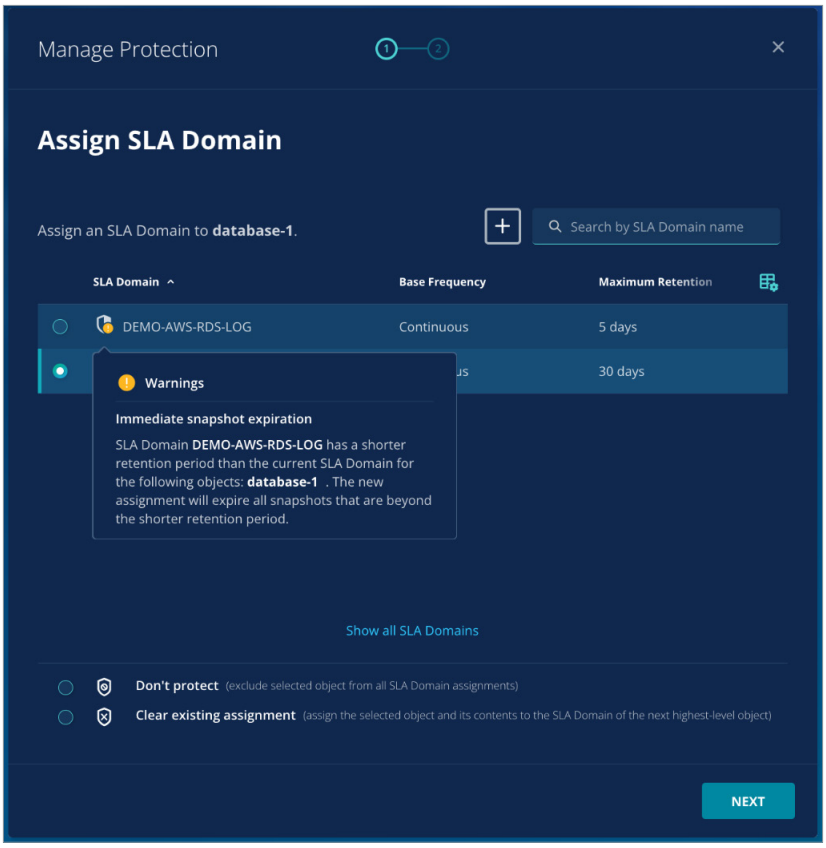

Figure 6 – Cloud-Native SLA Assignment Warning

TAG RULES

Tag Rules are a powerful construct that allow Cloud-Native Protection for Amazon RDS to leverage existing business logic and provisioning workflows to assign SLA Domains to the appropriate resources. This is accomplished by mapping an SLA Domain to an AWS Tag or to an AWS Tag/Value Pair. These rules can then be applied to RDS instances across any number of AWS regions and accounts as desired. This is an extremely simple, powerful, and lightweight mechanism for protecting RDS instances in large multi-account AWS environments.

When creating a tag rule, selecting (**All tag values**) in the value field will cause the rule to match all instances with the specified **Tag Key** regardless of the value. Similarly, selecting (**No tag value**) will cause the rule to apply only when a matching **Tag Key** is found without a **Tag Value**.

Tag rules also require that the customer selects the SLA Domain to assign when the tag rule has a match. This SLA Domain will be assigned to all matching objects, unless they have a specific SLA Domain assigned directly to them. Rubrik Security Cloud will periodically poll the customer's AWS Account(s) and update the assigned SLA Domains in accordance with these tag rules. This includes updated tag keys or values, newly provisioned RDS instances that match the rule, or modifications to the tag rule itself.

Lastly, leveraging **Do Not Protect** within a tag rule can oftentimes be useful for excluding specific workloads from account level Auto-Protection.

Since RDS instances can have multiple tags, Rubrik Security Cloud may apply multiple tag rules to the same instance. If the tag rules specify different SLA Domains, Rubrik Security Cloud selects one of them based upon the following order of precedence, highest to lowest:

- Do Not Protect
- The SLA Domain with the most frequent snapshots
- The SLA Domain with the longest retention

DIRECT SLA DOMAIN ASSIGNMENT

Users can also directly assign SLA Domains to RDS instances from the **AWS – RDS Inventory** screen within Rubrik Security Cloud. The filter options on the left-hand side of this view are particularly useful for tailoring the view to the desired resources, as is the instance search box at the top right of the view. In order to determine the current SLA Domain assigned to an instance, simply reference the **SLA Domain** column in this view. The source of that SLA Domain can be viewed in the **Assignment** column of the same view.

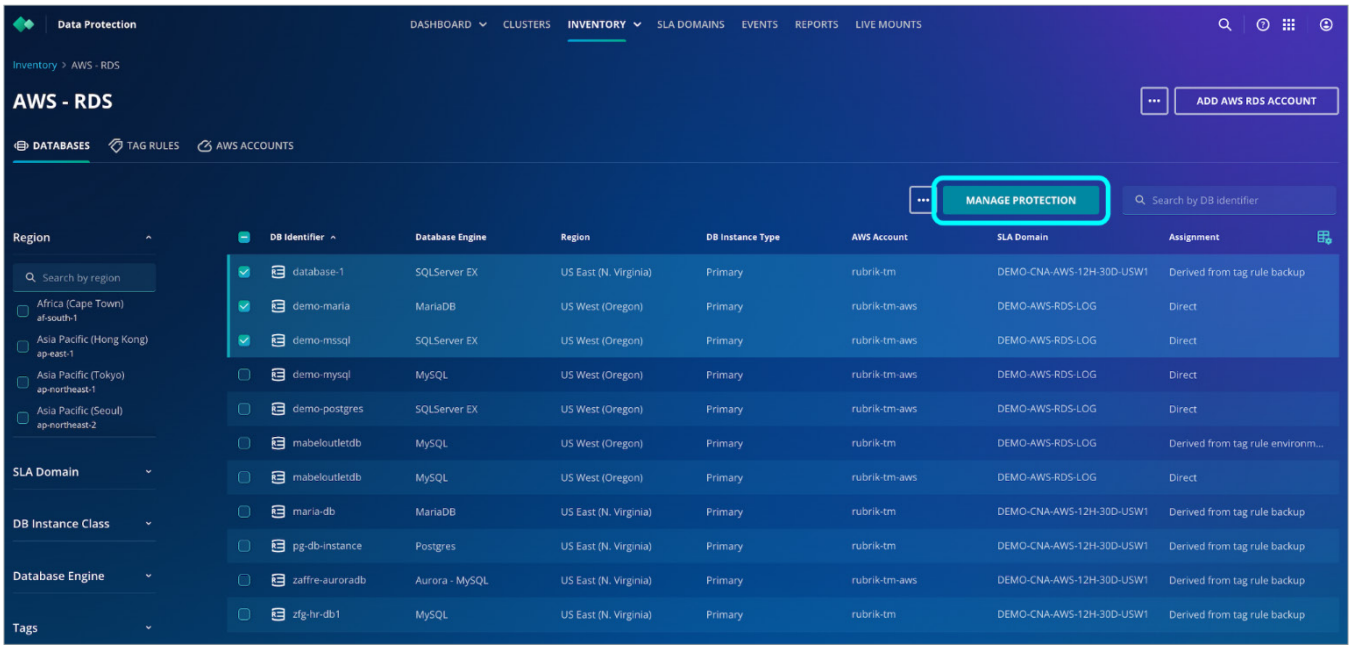

Figure 7 – Direct SLA Assignment

Ticking the checkboxes next to the desired instances and clicking **Manage Protection** will bring up the SLA Domain assignment dialogue box. This can also be accomplished by drilling down into an individual RDS

instance and clicking the same button. This view provides the option to select an existing SLA Domain as well as **Clear existing assignment**, which will remove any existing SLAs directly assigned to the object, and **Do Not Protect**, which forces Rubrik Security Cloud to stop protecting the object.

The main difference between these two options is the fact that **Clear existing assignment** will allow SLA Domains assigned at the account level or via tag rules to inherit down to the selected object. **Do Not Protect**, on the other hand, will force Rubrik not to protect the object regardless of any account level assignments or tag rules that might be in play. In general, direct SLA Domain assignment tends to be a last resort for overriding the SLA Domains inherited from account level SLA Domain assignment or tag rules as opposed to the primary means of assigning SLA Domains to RDS instances.

ON DEMAND SNAPSHOTS

On Demand snapshots are snapshots that are manually created and differ from SLA Domain created snapshots in a few significant ways. An On Demand snapshot is created when a user clicks the **Take On Demand Snapshot** button on the view for an RDS instance. The user will also select an SLA to manage the life cycle of that on demand snapshot. This means that Rubrik will automatically delete the on demand snapshot based on the retention value in the SLA. On Demand snapshots can also be deleted manually by first changing the retention to **Forever**. This will allow you to then manually delete the On Demand snapshot. Only snapshots with a type of **On Demand** can be deleted by users. On Demand snapshots are a useful tool when employed prior to making any change inside of an instance that cannot be easily recreated through other means. Simply take an On Demand snapshot, perform the required task, and then delete the On Demand snapshot after confirming the change is non-breaking. On Demand snapshots can also be used in conjunction with scripting to pause applications prior to snapshotting an instance.

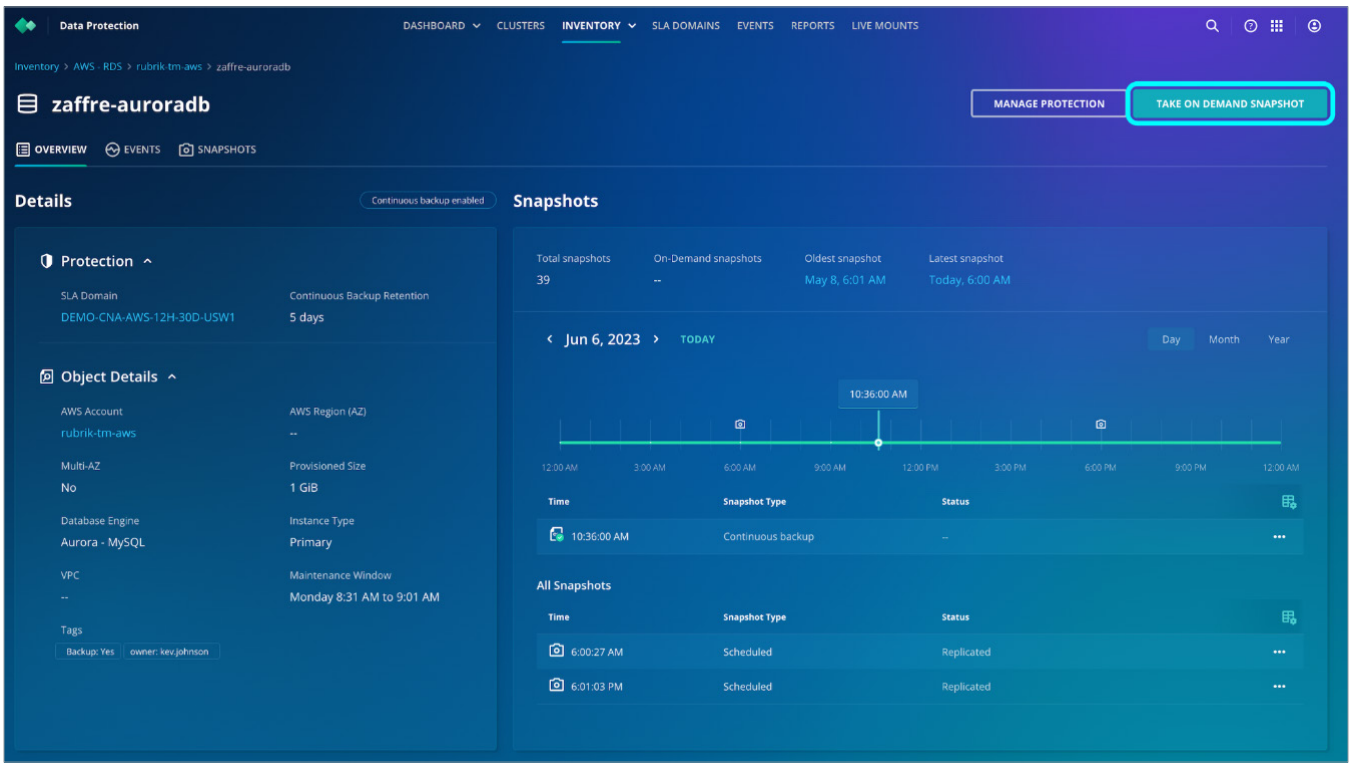

Figure 8 – On Demand Snapshot

HOW IT WORKS

TYPES OF BACKUPS

Rubrik Security Cloud provides two types of backups. Continuous and Scheduled backups.

Continuous (log) Backups: Provides extremely granular protection, by recording all changes as they occur near real time and enables point-in-time recovery. This is based on AWS automated snapshots.

Scheduled Backups: Based on the frequency set inside the SLA Domain. This is required for cross account or region replication. RDS Schedule d snapshots are incremental with the exception of Aurora, which are always full snapshots.

With AWS Aurora, snapshots are not incremental and will create full copies. This exponentially increases the cost of backing up the data. E.g. If you snapshot your database everyday for 30 days, and the database is on average 10GB large, you will be billed for 30×10GB = 300GB of storage.

ASSIGNING SLA

Once accounts are added and SLA Domains are assigned, it's time to start protecting RDS workloads in AWS. After SLAs are assigned to RDS workloads in Rubrik Security Cloud, Cloud-Native Protection for RDS will enable point-in-time recovery, and begin automatically scheduling and creating RDS Snapshots in the customers AWS account(s) as required. These snapshots are automatically created and retained in accordance with the frequency and retention defined in the assigned SLA Domain, there is no need for the customer to schedule any jobs. This process is identical for all database engines in RDS.

Enabling Point-in-Time Recovery

The number of days for which point-in-time recovery is available for an RDS instance protected by Cloud-Native Protection for Amazon RDS is defined by the **Log backup for AWS RDS** parameter within the RDS instance's assigned SLA Domain. In order to perform point-in-time recovery, automatic RDS backups must be enabled on the RDS instance being recovered. Per AWS:

AWS RDS (NON-AURORA)

"When automated backups are turned on for your DB instance, Amazon RDS automatically performs a daily snapshot of your data. The snapshot occurs during your preferred backup window. It also captures transaction logs to Amazon S3 every 5 minutes (as updates to your DB instance are made)."[1]

AWS RDS (AURORA)

"Aurora backs up your cluster volume automatically and retains recovery data for the length of the backup retention period. Aurora backups are continuous and incremental so you can quickly restore to any point within the backup retention period."[2]

When a customer defines the Log Backup for AWS RDS duration in an SLA Domain and assigns that SLA Domain to an RDS instance, Rubrik Security Cloud automatically configures the retention period for RDS automated backups on that RDS instance to match. This is what allows Rubrik Security Cloud to perform point-in-time recovery. As of this writing, AWS supports retention periods up to 35 days[3]. This change takes effect during the next maintenance window for that instance. Changes to the retention duration for instances that already have Log Backup for AWS RDS enabled are immediate. The diagram below depicts this workflow at a logical level.

1 https://aws.amazon.com/blogs/database/implementing-a-disaster-recovery-strategy-with-amazon-rds/
2 https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html
3 https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html

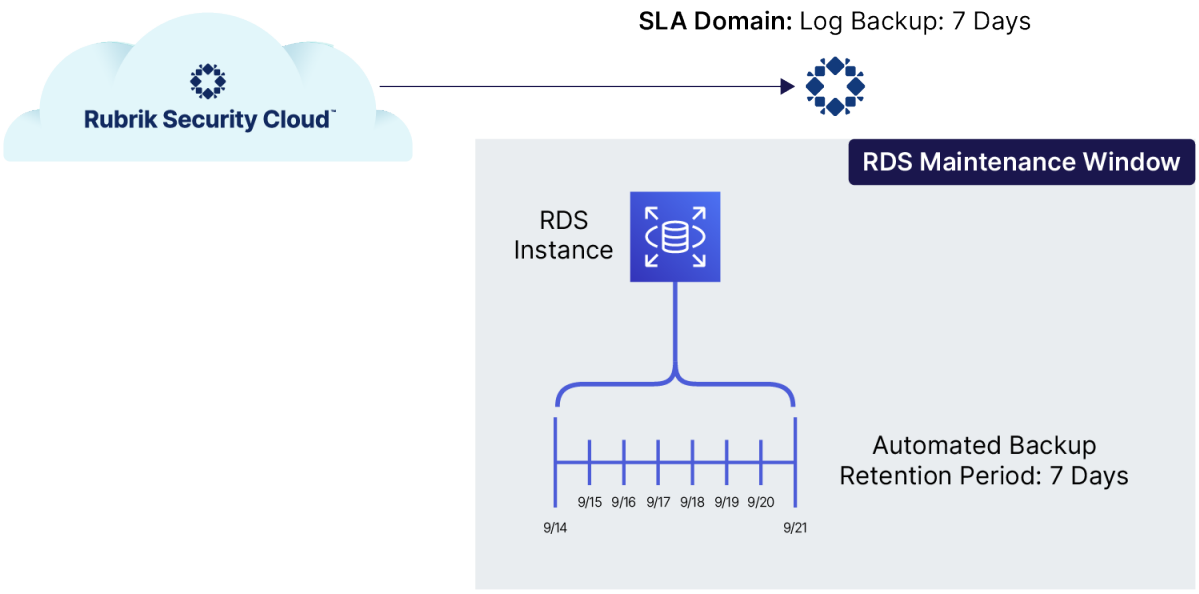

Figure 9 – Configuring Point-In-Time Recovery for Amazon RDS

Snapshot Based Protection

For recovery outside of the point-in-time recovery window described above, Rubrik Security Cloud leverages on-demand RDS Snapshots. The API call is made to initiate the RDS snapshot process via the CreateDBSnapshot for Non-Aurora instances and CreateDBClusterSnapshot for Aurora clusters API call. Finally, the Rubrik specific tags are added, and any non-conflicting tags are copied from the source instance to the snapshot. Snapshots that are taken of instances with encrypted volumes are automatically encrypted.

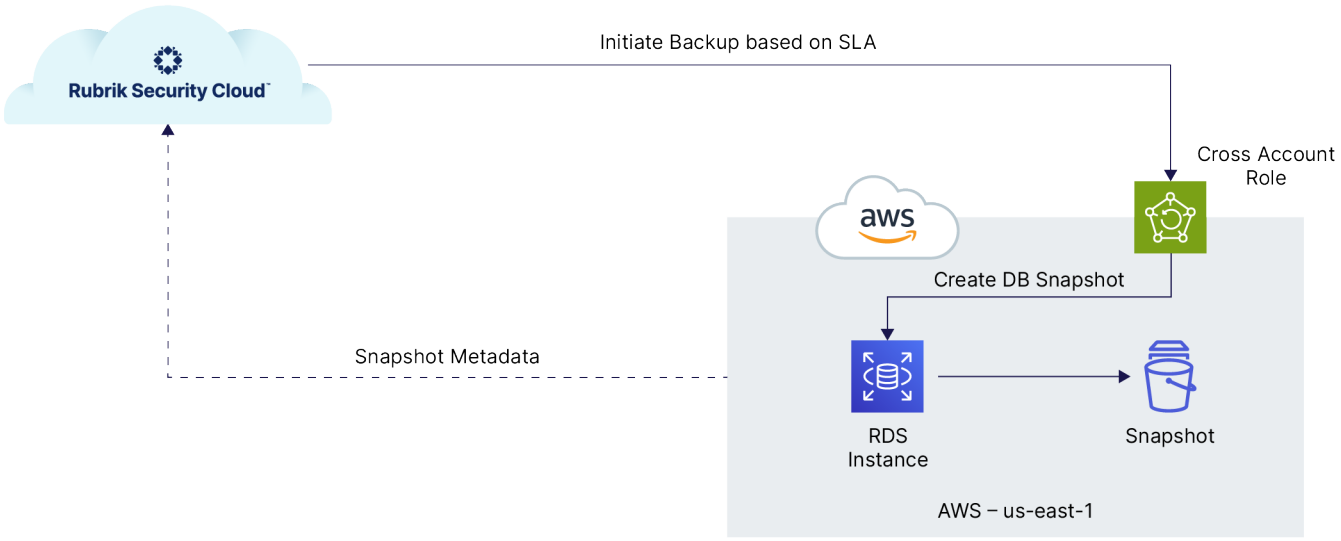

Figure 10 – RDS Instance Snapshot via AWS Backup

RECOVERY

RDS instance exports allow the customer to create a new RDS instance from a snapshot of their choosing. This workflow allows the user to define the database instance identifier, class, storage type, KMS key, VPC, subnet, security groups as well as other database and network parameters. Tags that were on the source instance at the time of the snapshot can also be included or excluded from the export process.

Point-in-Time Exports

Customers can initiate a point-in-time export by drilling down into the calendar view for the instance they want to recover. Point-in-time recovery is available for at least a portion of any day where a green bar is displayed at the bottom of the calendar tile for that day. As stated prior, the time period for which point-in-time recovery is available is dictated by the retention period for RDS log backups in an instance's assigned SLA domain.

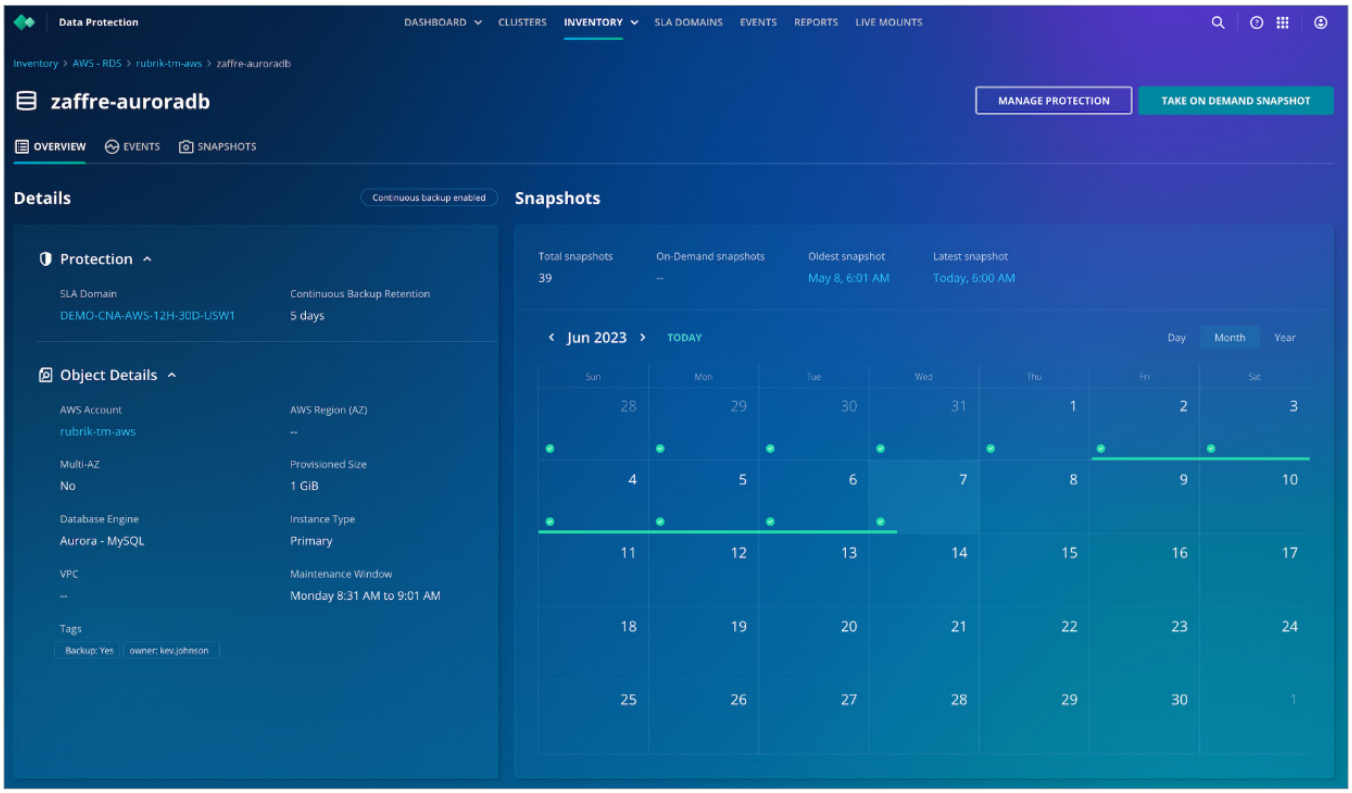

Figure 11 – Point-in-Time Recovery Window for Amazon RDS

Upon drilling down into the calendar view for the day containing the desired recovery point, customers can utilize the slider to select any portion of the green bar for point-in-time recovery. Once the desired time is selected, customers can click the ellipses next to the Log backup entry and select **Export** to begin the recovery process. Log backups can only be recovered within the same region as their source instance.

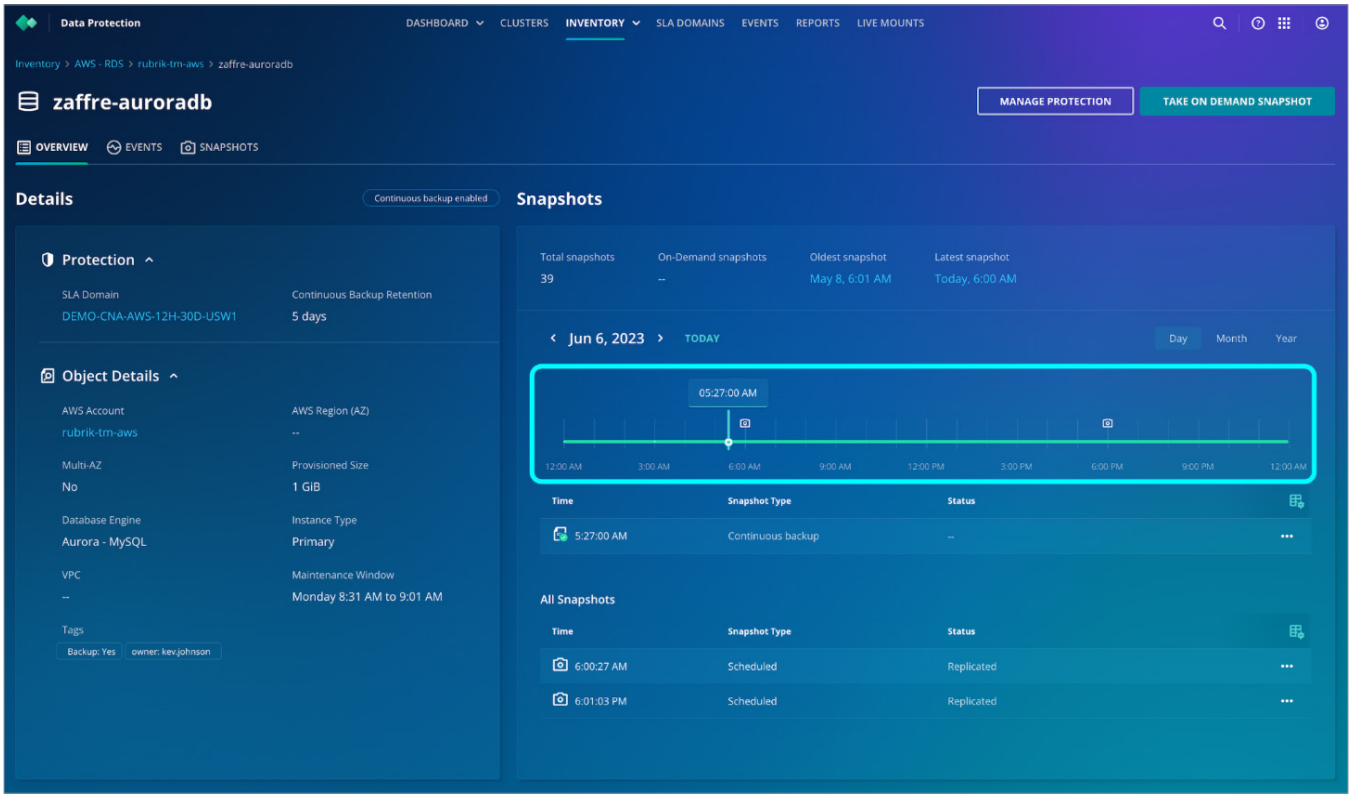

Figure 12 – Point-in-Time Recovery Selection for Amazon RDS

Once the export is initiated, Rubrik Security Cloud uses the RDS RestoreDBInstanceToPointInTime API for Non-Aurora instances and RestoreDBClusterToPointInTime for Aurora clusters to initiate the creation of a new instance or cluster. This instance will be transactionally consistent to the point in time selected by the customer when initiating the export operation. The diagram below depicts this process:

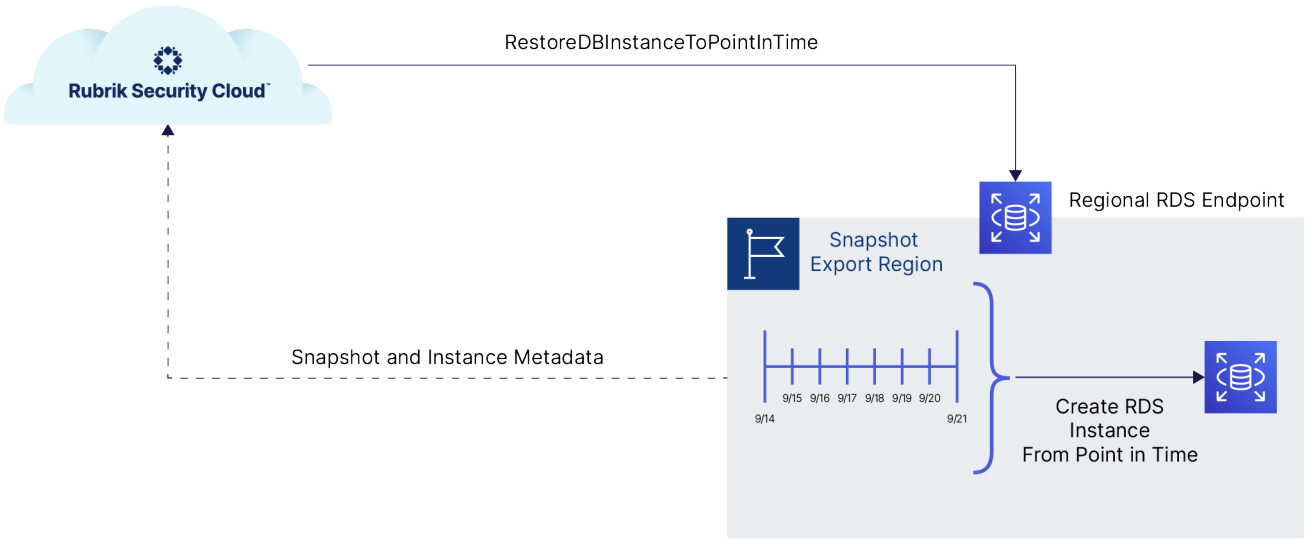

Figure 13 – Point-in-Time Recovery for Amazon RDS

Snapshot Exports

Customers can also export RDS instances using any of the on demand or scheduled snapshots created by an assigned SLA domain. Snapshot based recovery is available for any day containing a checkmark on the calendar view for the instance being recovered. Rubrik Security Cloud accomplishes this by first copying the snapshot to the export region via the RDS CopyDBSnapshot for non-Aurora instances or CopyDBClusterSnapshot API for Aurora clusters if necessary, and then by launching a new RDS instance using the RDS RestoreDBInstanceFromDBSnapshot API for non-Aurora instances or RestoreDBClusterFromSnapshot for Aurora clusters. Lastly, if a copy was required, the copied snapshot is removed via the DeleteDBSnapshot for non-Aurora instances or DeleteDBClusterSnapshot API. To begin the export process, customers click on the ellipses next to the desired snapshot and select **Export**. Snapshot based exports have the added benefit of offering the customer the option to export to regions other than that of the source instance. This is accomplished by copying the snapshot to the destination region prior to executing the export. The diagram below depicts this process.

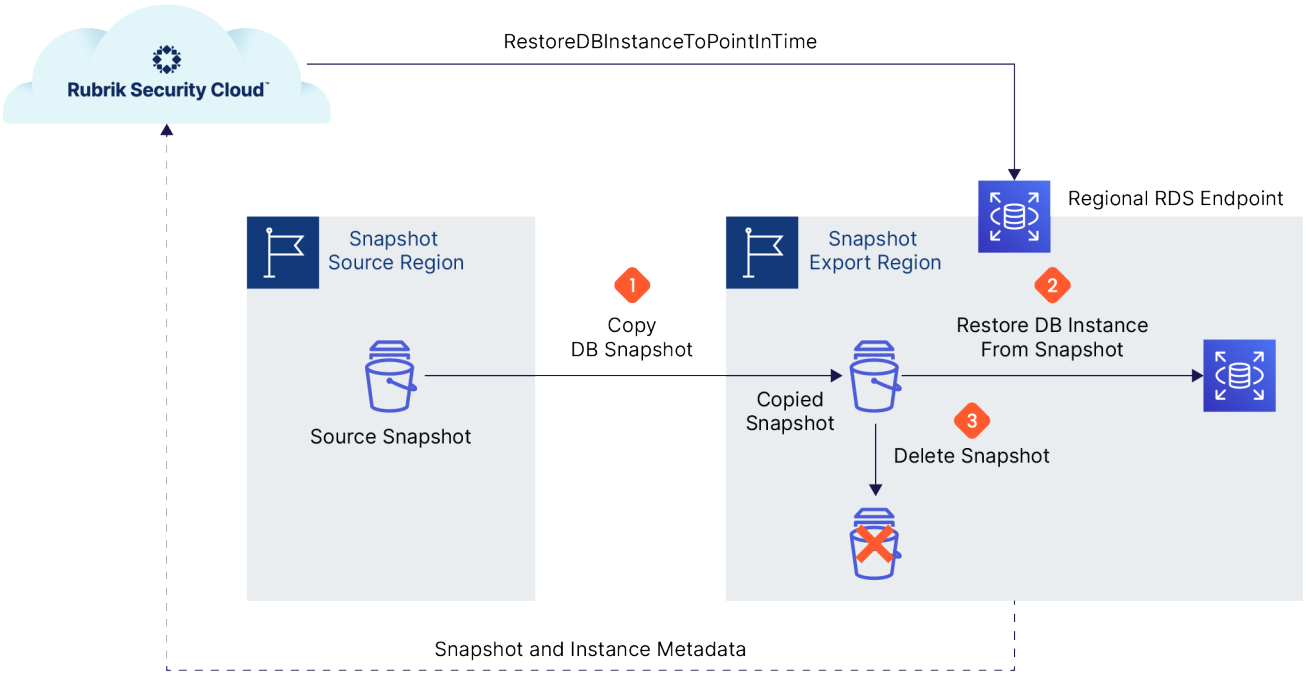

Figure 14 – Cross Region Snapshot Export for Amazon RDS

SUMMARY

This concludes the How it Works guide on Cloud-Native Protection for Amazon RDS on Rubrik Security Cloud. The guide explained the architecture and the value proposition of Cloud-Native Protection for Amazon RDS as well as educated the reader on the nuances of each major workflow within the product. Additionally, the guide equipped the reader with some common techniques and best practices for using the product efficiently from both an operations and cost perspective, allowing them to fully understand and unlock the potential of Cloud-Native Protection for Amazon RDS.

GLOSSARY

AMAZON RELATIONAL DATABASE SERVICE (RDS) INSTANCE

Amazon RDS provides a selection of instance types optimized to fit different relational database use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your database. Each instance type includes several instance sizes, allowing you to scale your database to the requirements of your target workload.

AMAZON RELATIONAL DATABASE SERVICE (RDS) SNAPSHOT

When creating a snapshot, Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. Creating this DB snapshot on a Single-AZ DB instance results in a brief I/O suspension that can last from a few seconds to a few minutes, depending on the size and class of your DB instance. For MariaDB, MySQL, Oracle, and PostgreSQL, I/O activity is not suspended on your primary during backup for Multi-AZ deployments, because the backup is taken from the standby. For SQL Server, I/O activity is suspended briefly during backup for Multi-AZ deployments.

AMAZON RESOURCE NAME (ARN)

Amazon Resource Names (ARNs) uniquely identify AWS resources. We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

AMAZON SIMPLE NOTIFICATION SERVICE (SNS)

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud. It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

AMAZON WEB SERVICES (AWS)

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

AWS AVAILABILITY ZONES (AZ)

An AWS Availability Zone is a subset of a region, except in the case where a Region only has one AZ, which is rare. AZs are discrete data centers within a region and have redundant power, networking, and connectivity within the Region Network performance across AZs is typically high throughput and low latency. AZs are typically many miles away from other AZs but are generally within 60 miles of one another within the region. Certain entities such as EC2 instances and EBS volumes exist within a single AZ, although the latter is redundant within the AZ but not across AZs or regions.

AWS CLOUDFORMATION

AWS CloudFormation allows AWS customers to model infrastructure and application resources with templates written in formats like JSON or YAML. These templates can be submitted to the CloudFormation service on AWS to create a collection of resources known as a stack. These stacks and their life cycles are managed via CloudFormation itself rather than individually in AWS. Update the template in CloudFormation and the stack will be updated accordingly, delete the stack and all resources provisioned as part of the stack will be deleted as well.

AWS IDENTITY AND ACCESS MANAGEMENT (IAM) POLICY

AWS customers manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied.

AWS IDENTITY AND ACCESS MANAGEMENT (IAM) ROLE

Roles are an identity that AWS customers can create in AWS and assign permissions to. Roles can be utilized only by trusted AWS services or accounts. IAM Policies attached to these roles dictate the actions that an entity using the role can take.

AWS KEY MANAGEMENT SERVICE (KMS)

AWS KMS is a managed service that enables AWS customers to easily create and control the keys used for cryptographic operations. The service provides a highly available key generation, storage, management, and auditing solution for AWS customers to encrypt or digitally sign data within their own applications or control the encryption of data across AWS services.

AWS REGIONS

AWS is made up of regions spread across the globe, each region is completely independent of the other regions in terms of location, power, cooling, water supply, etc. Each region consists of one or more (typically two or more) availability zones. Many AWS services (such as EC2) are managed at the regional level, leveraging separate consoles and API endpoints when interacting with different regions within an AWS account.

CUSTOMER AWS ACCOUNT

The customer owned AWS account houses the resources protected by Cloud-Native Protection for Amazon RDS. Rubrik Security Cloud assumes an IAM Role in the form of the Rubrik owned AWS account into the customer owned account in order to make the API calls necessary to protect EC2 instances and EBS volumes. The images and snapshots created by this process continue to reside in the customer owned AWS account, only metadata is stored in Rubrik Security Cloud.

POINT-IN-TIME RECOVERY FOR AMAZON RDS

When automatic backups are enabled for an Amazon RDS instance, AWS automatically creates and saves automated backups of your DB instance during its backup window. RDS creates both storage volume snapshots of your DB instance, as well as log backups of databases running on the Instance. RDS saves the automated backups of your DB instance according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period.

RELIC

In terms of Cloud-Native Protection for Amazon RDS, a relic is an EC2 instance or EBS volume that has been deleted while there are still existing snapshots for the object. These snapshots are retained according to the SLA assigned to the object at the time of deletion. On demand snapshots can be deleted as usual.

RUBRIK AWS ACCOUNT

The Rubrik owned AWS account is used as an identity source and as a bastion for interacting with the customer owned AWS accounts that are under protection via Rubrik Security Cloud. Making AWS API calls from this account allows Rubrik to leverage IAM Roles to interact with customer owned AWS accounts.

SERVICE LEVEL AGREEMENT (SLA) DOMAIN

Rubrik SLA Domains are data protection policies built within Rubrik Security Cloud and then assigned to assets requiring protection. SLA Domains are comprised of these three components when utilized by Cloud-Native Protection for Amazon RDS:

- Snapshot Frequency
- Snapshot Retention
- Replica Region and Duration

APPENDIX A – AWS TAGS

Instance Snapshot Tags:

Tag Key	Tag Value
rk_aws_native_account_id	Rubrik Security Cloud UUID for source AWS Account
rk_component	"Cloud Native Protection"
rk_gc_leaked_resource	true/false
rk_object	"RDS Backup"
rk_snapshot_type	Primary/Replica
rk_source_db_instance_name	Name of source RDS instance
rk_taskchain_id	Rubrik Security Cloud UUID for job taskchain

Exported Instance Tags:

Tag Key	Tag Value
Name	The name assigned to the exported instance
rk_aws_native_account_id	Rubrik Security Cloud UUID for source AWS Account
rk_component	"Cloud Native Protection"
rk_export_source_region	source region for export
rk_export_timestamp	Export timestamp in Zulu time
rk_snapshot_type	Primary/Replica
rk_source_db_instance_name	Name of source RDS instance
rk_source_db_instance_restore_time	Point in time for export snapshot
rk_user	User who initiated the snapshot

APPENDIX B – METADATA

Metadata	Use Case
Account ID Account Name IAM Role (created by Rubrik) Stack ARN (created by Rubrik)	For setup/upgrade/disable operations for each account added on Rubrik Security Cloud.
Following properties of EC2 instances in the accounts added on Rubrik Security Cloud: • Instance ID • Instance Name • Instance Type • Region and Availability Zone • VPC ID, name • Private/Public IP • OS Type • Tags • Is VM Marketplace or not. • SSH key pair name • Indexing status of the instance	Display instances list on Inventory and protect/recover them. Only applies to protected regions.
Following properties of EBS volumes in the accounts added on Rubrik Security Cloud: • Volume ID • Volume name • Volume Type • Region and Availability Zone • Size, IOPS • Is encrypted or not. • Is Marketplace or not. • Tags • Indexing status of the volume • ID of snapshot from which volume was created (if applicable) • Device path (if the volume is attached to an EC2 instance) • Is root volume (if the volume is attached to an EC2 instance)	Display EBS volumes on Inventory and protect/recover them. Only applies to protected regions.
Following properties of VPCs in the accounts added on Rubrik Security Cloud: • VPC ID • VPC Name • Region	To allow filtering by VPCs in the EC2 instance/EBS volume list pages on Inventory. Only applies to protected regions.

Following properties of snapshots (AMI, EBS Volume Snapshots) taken by the Rubrik: • Snapshot ID, name • Metadata of corresponding EC2 instance/EBS Volume	Backup and Recovery.
For File Recovery feature: • VPC, 2 Subnets in which Exocompute is launched. • Cluster Control Plane Security Group and Worker Node Security Group (Created by Rubrik) • Cluster Role ARN, Worker Node Role ARN, Node Instance Profile (Created by Rubrik) • Index files (which includes filesystem metadata: file paths, stat information etc.).	To be able to launch EKS clusters in provided VPC and run indexing/file download.
Following properties of RDS instances in the accounts added on Rubrik Security Cloud: • DB instance identifier, DbiResourceId, instance ARN • DB engine and engine version • DB instance class • Allocated storage • Region, Primary Availability zone • Is instance multi-az or not • Status of DB instance • Maintenance window • Option group, parameter group, subnet group • Backup retention period • Pending backup retention period modification • Storage type and IOPS • Kms Key • Port • VPC ID	Display RDS instance list on inventory and protect/recover them. Only applies to protected regions.

VERSION HISTORY

Version	Date	Summary of Changes
1.0	October 2020	Initial Release
2.0	July 2023	Updated for Aurora Support
2.1	September 2023	Updated Figure 1 and boilerplate

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-amazon-rds / 20230913