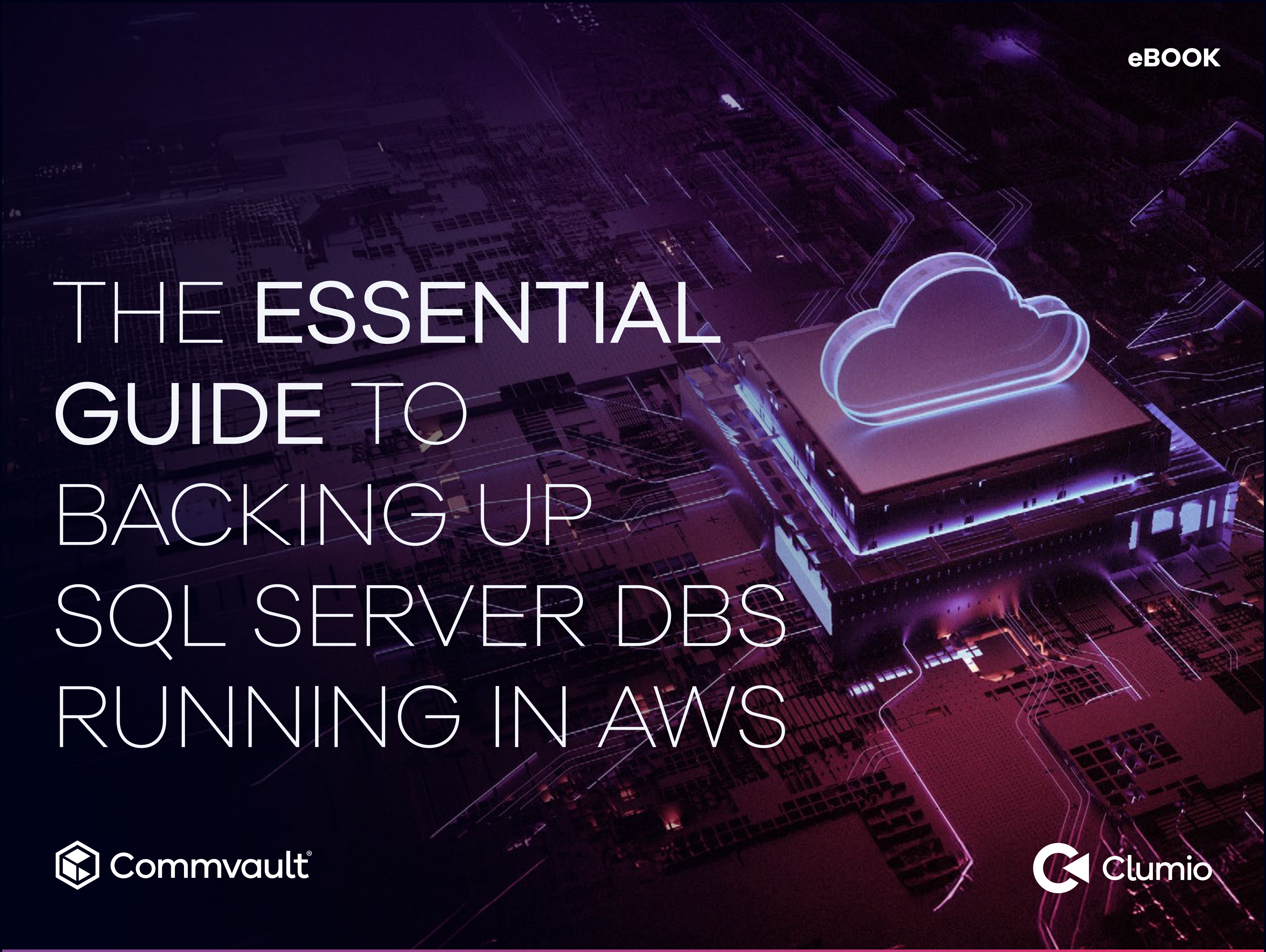


eBOOK

THE ESSENTIAL GUIDE TO BACKING UP SQL SERVER DBS RUNNING IN AWS

Commvault®

Clumio

TABLE OF CONTENTS

Commvault®

01

Introduction

As cloud services reach maturity, architects, engineers, and operators of cloud storage and databases are now designing systems with a foundational focus on data integrity and resilience.

SQL Server, trusted for over 30 years by database and IT professionals, is still a popular and reliable database serving industrial, healthcare, manufacturing, education, and several other critical industries. For both cloud-native and cloud-migration use cases, SQL Server on Amazon Web Services (AWS) has emerged as a strong, flexible database solution. However, understanding the responsibility model, architecture, and complexities of backing up data on SQL Server in AWS requires a nuanced level of understanding in the context of the cloud.

This Essential Guide aims to help database administrators (DBAs) and cloud infrastructure / IT teams understand the different architectures of SQL Server on AWS. Beginning with an overview of the shared responsibility model governing SQL Server on AWS, it goes on to explore various server architectures including Amazon RDS for SQL Server, SQL Server on Amazon EC2 Instances, and SQL Server on VMware Cloud (VMC).

This book then unpacks SQL Server Backup concepts, introducing key terms like VSS agent, snapshots, T-logs, and point-in-time recovery. We then explore various SQL Server backup options in AWS, including in-built backups, snapshots, AWS Backup, third-party tools, and Clumio, an AWS backup services leader. We delve into Clumio's architecture and advantages, and its application to SQL Server on EC2 and RDS. We also guide you through setting up your backup policy, performing your first backup, and initiating point in time recoveries with Clumio. To contextualize these concepts, we present real-world case studies demonstrating Clumio's application in SQL Server Backup. The guide concludes by reflecting on how SQL Server backups in AWS can be made easier on an ongoing basis, giving businesses the confidence to secure their data effectively and efficiently.

Whether you are new to AWS and SQL Server backups or a seasoned professional looking to optimize your database environment, this guide offers insights, practical steps, and inspiration to make the most of the opportunities that the integration of SQL Server on AWS and Clumio offers.

Understanding the shared responsibility model governing SQL Server on AWS

As organizations transition their SQL Server workloads to AWS, it's crucial to understand the shared responsibility model, a pivotal aspect of working with data in AWS. This model delineates the roles of the cloud provider and the customer as it relates to the protection, resilience, and security of the data.

In AWS, the shared responsibility model can be thought of as two parts: "Security of the Cloud" and "Security in the Cloud."

Security of the Cloud pertains to AWS's responsibility. AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud. This includes physical security of the buildings, power supply and durability of the server, storage, and networking hardware in its data centers, and the uptime and availability of AWS-provided software.

Customer is responsible for Security in the Cloud	**DATA** Durability, Resilience, Integrity, Availability, Access Control, Encryption

Security in the Cloud, on the other hand, is the responsibility of the customer. Customers are responsible for the security and resilience of any data they put on the cloud. In the context of SQL Server on AWS, this includes managing access controls, encrypting the data as necessary, and maintaining resilience and recoverability against operational disruptions such as accidental or malicious deletions.

Whether SQL Server is deployed within Amazon RDS, directly on Amazon EC2 instances, or as part of VMC, understanding who is responsible for what is key to making sure there are no gaps in data resilience and application availability.

Cloud provider is responsible for Security of the Cloud	**INFRASTRUCTURE DURABILITY** Hardware and infrastructure resilience	**SERVICE UPTIME** Computer, Storage, Network, Database service avaiability

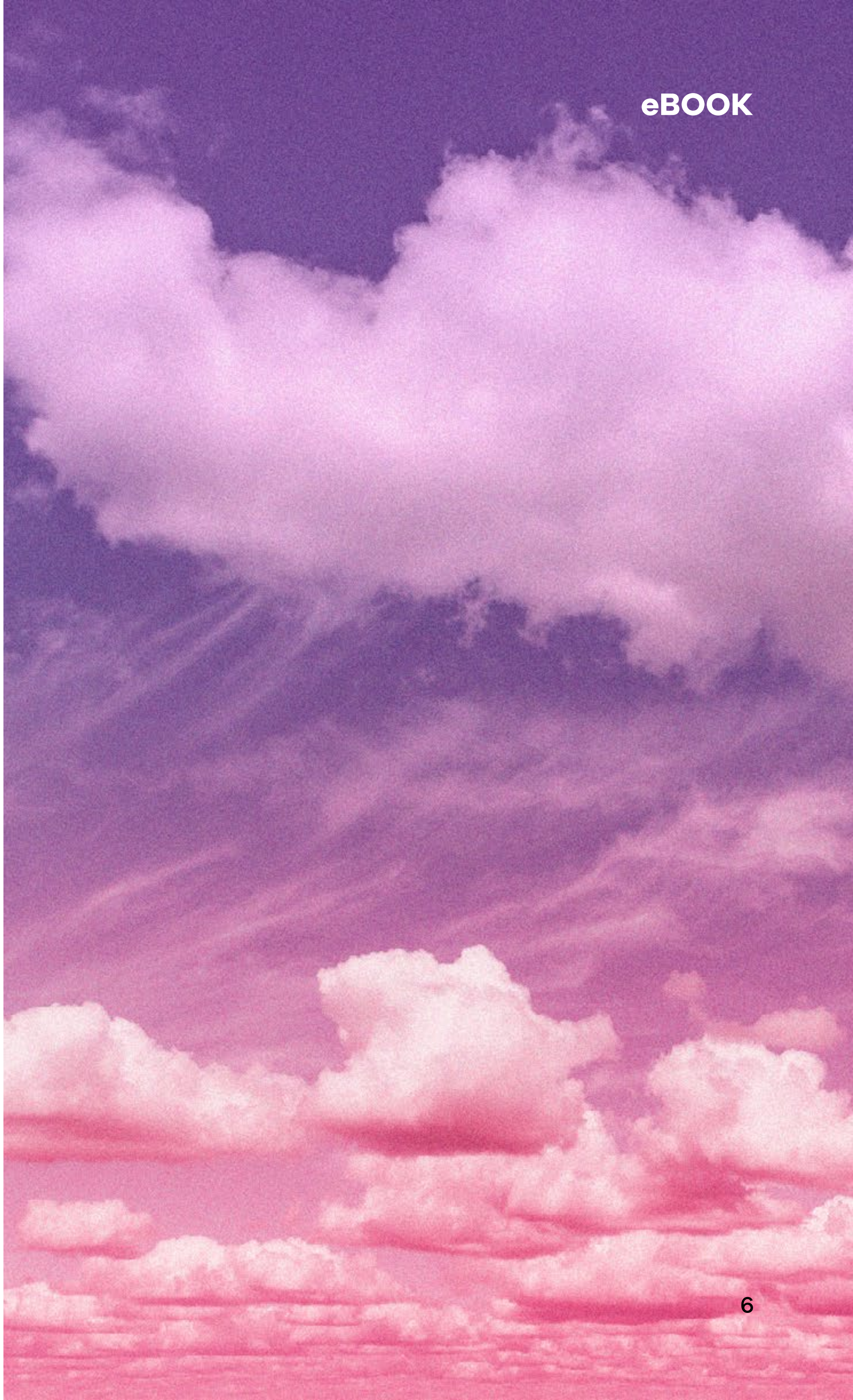
You are responsible for maintaining the security & resilience of any data you put on the cloud.

When running SQL Server on Amazon RDS, AWS handles much of the underlying infrastructure security and database patching. AWS also provides options for the customer to back up their data on Amazon RDS using natively available tools such as snapshots or automated backups. However, the customer still has responsibilities, such as managing the policies of these backup configurations, air gapping backups for additional protection, maintaining data privacy, and securing applications connected to the database server.

For SQL Server on Amazon EC2 instances or VMC, customers get more control over the upgrades, patching, and performance tuning of the databases, along with implementing the right backup and recovery strategies for the customers' particular use case. AWS's role is primarily focused on the uptime and availability of the underlying compute and storage instances.

Understanding this shared responsibility model is crucial for implementing effective backup strategies for SQL Server on AWS. It is necessary to know what part of the backup process AWS manages and what parts need to be handled by the customer or a third-party solution like Clumio.

The next sections will delve into the different architectures of SQL Server on AWS, further exploring the nuances of the shared responsibility model.

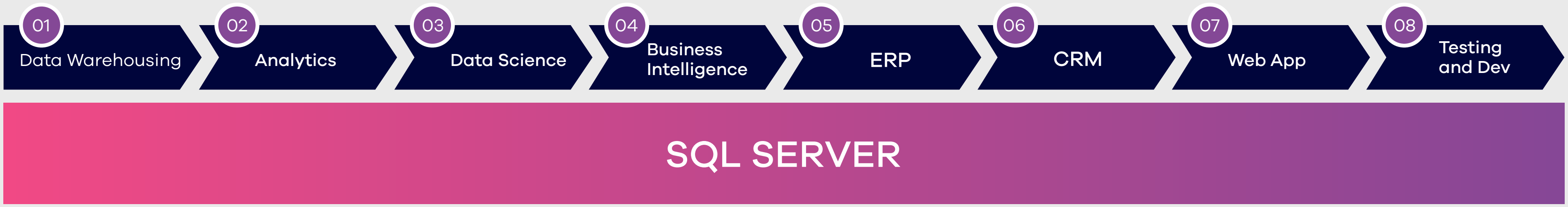


Overview of SQL Server on AWS

When it comes to relational databases, Microsoft SQL Server remains a popular choice due to its robust features, reliable performance, and wide compatibility. As businesses move toward cloud-based solutions, deploying SQL Server on AWS has become a common practice.

SQL Server's versatile deployment options cater to a wide variety of businesses that are running applications on AWS, warehousing data for business intelligence purposes, or even feeding data into data lakes for data science and machine learning. Ultimately, the choice of SQL Server deployment on AWS depends on several factors, including the company's business needs, the level of control needed over the database, the available IT resources, and the company's long-term data strategy.

AWS offers multiple ways to deploy SQL Server, each with its unique benefits and considerations.



01 Data Warehousing

02 Analytics

03 Data Science

04 Business Intelligence

05 ERP

06 CRM

07 Web App

08 Testing and Dev

SQL SERVER

HOSTED ON AMAZON RDS

Amazon RDS is a managed database service that simplifies much of the administrative overhead. It takes care of routine database tasks such as provisioning, setup, patching, and upgrades. This is an excellent option for businesses that prefer to focus on their application and leave most of the database maintenance tasks to AWS. Amazon RDS is among the most popular databases for developers and DevOps teams. Of course, given it is a managed service, it is also the most expensive of the three SQL Server deployment models in AWS.

DEPLOYED ON AMAZON EC2

This choice offers maximum control over the database environment. Deploying SQL Server on Amazon EC2 instances is especially beneficial when businesses have specific needs that are unmet by Amazon RDS, such as the need for more direct control over the performance characteristics of their database. These deployments are usually staffed by DBAs. In most cases, deploying SQL Server on Amazon EC2 costs the least in terms of infrastructure, since you are paying only for compute and storage resources.

DEPLOYED ON VMWARE CLOUD (VMC) ON AWS

With this service, businesses can seamlessly migrate their VMware-based SQL Server workloads to the AWS Cloud without needing to re-platform their applications. This offers the advantage of familiarity with VMware's software-defined data center technologies while leveraging AWS's scalability and service.

Let's take a deeper look into the details of these deployment options and considerations for backing up SQL Server in each of these environments.

Amazon RDS is among the most popular databases, but is also the most expensive of the three deployment models.

02

SQL Server Backup Concepts

When implementing a robust backup strategy for SQL Server on AWS, it's crucial to understand the key concepts involved.

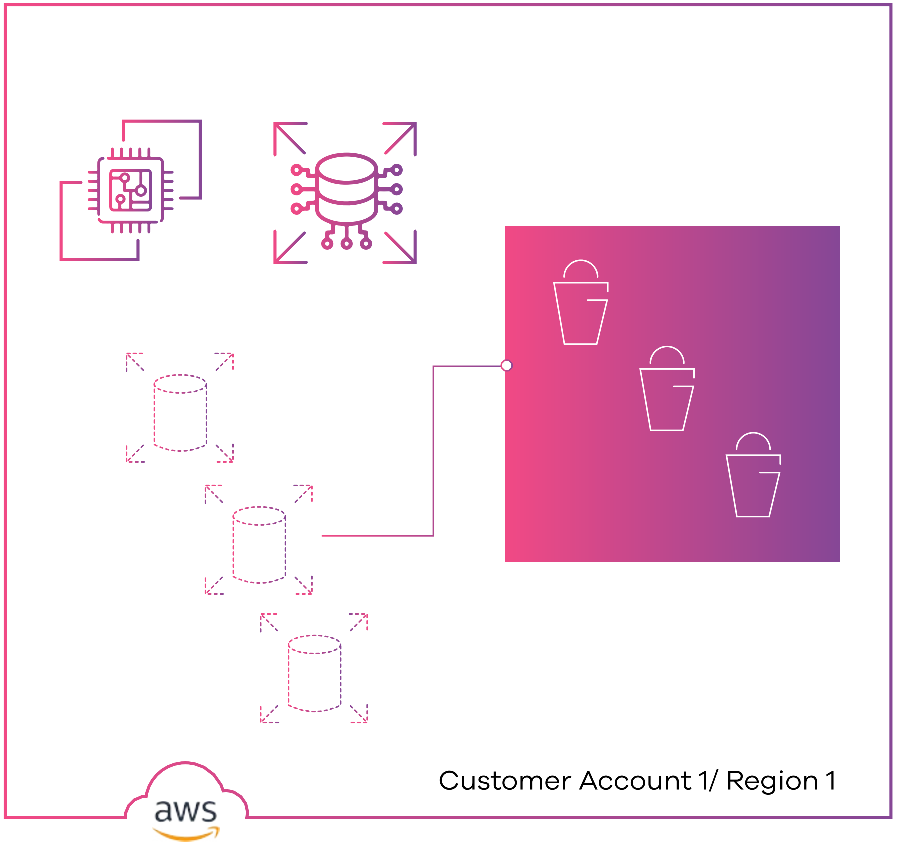
These concepts form the foundation of backup operations and directly influence how you maintain data integrity, achieve business continuity, and enable the quick recovery of your data after an operational disruption. This chapter defines those key concepts, and will better equip you to design and implement a backup strategy that meets your business needs, optimizes resources, and mitigates the risks associated with data loss.

Full, differential, and incremental backups

In SQL Server, the first backup is always a full backup— a complete backup of all your data and any additional components required to restore the database.

It serves as the base for both differential and incremental backups and typically takes the most time and storage space. Differential backups contain all the changes made since the last full backup. The Differential Change Map, or DCM, keeps track of these changes. While these backups typically take less time and storage space than full backups, they may become larger over time as they always include all changes since the last full backup. This can result in duplicate data across multiple differential backups.

While not natively supported by SQL Server, incremental backups involve backing up only the changes since the last successful backup. This results in small and fast change capture. For SQL Server instances running in AWS, incremental backups can be implemented in the form of incremental snapshots, as explained in the section below.



In the realm of data storage, backups, and disaster recovery, 'snapshots' are a key concept.

A snapshot is essentially a copy of the state of a storage volume at a particular point in time. It serves as a detailed table of contents, pointing to where data blocks are stored at that specific time. Snapshots occupy disk space, which is why most database backup strategies employ taking snapshots on a daily, weekly, or monthly basis and retaining them for several weeks or months.

Snapshots are an effective component of a backup strategy for SQL Server databases because they enable you to quickly save the state of your database at any given moment, and then return to that state if needed. This is especially useful in scenarios where a faulty update or a cyber-attack causes data loss or corruption.

On AWS, snapshots can be created for Amazon RDS instances, Amazon EBS volumes attached to Amazon EC2 instances, or the VMware cluster in which the SQL servers are running. Creating a snapshot of an Amazon RDS instance or an Amazon EBS volume results in a copy of the data at that point in time, allowing the user to return to that point if required. This may tempt you to think that snapshots can also be used for point in time recovery, but that is not the case. Point in time recovery (covered in more detail below) lets you recover to any point in time, down to the minute or second. Snapshots, however, let you return to only that point when the snapshot was taken.

Transaction logs

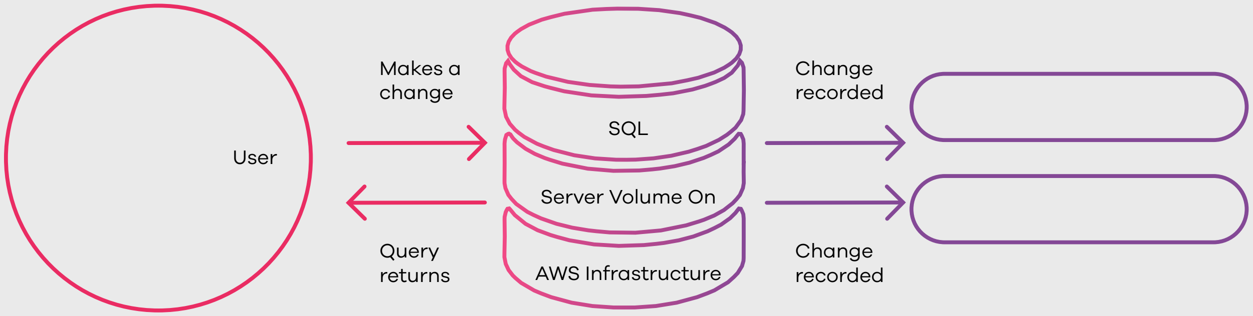
In the context of SQL Server, transaction logs (T-logs) are vital for maintaining data integrity and enabling point-in-time recovery. Every database has a transaction log that records all modifications to the database. In addition to user transactions, it records the details of modifications made by system procedures and functions to maintain data integrity.

T-logs contain a sequential record of all actions that have occurred in the database. When a user or application issues an INSERT, DELETE, UPDATE, or any other SQL command that modifies data, the change is recorded in the log before it is committed to the database. This way, data modifications are not lost in the event of a system failure.

Regular backups of the transaction log enable point-in-time recovery for the database, down to the exact second, rather than periodic intervals as with snapshots. This can be especially helpful in scenarios where data corruption or loss has occurred in between snapshots and you need to revert to a state right before the incident. SQL Server can use the transaction log to recover the database by rolling forward committed transactions and rolling back incomplete transactions.

In an AWS environment, Amazon RDS for SQL Server automatically manages the transaction log backups if you've enabled automated backups. However, for SQL Server on Amazon EC2 instances, the DBA or IT team will need to handle transaction log backups separately or automate it with Clumio.

Understanding the role of transaction logs is essential for a comprehensive SQL Server backup strategy that leaves no gaps.



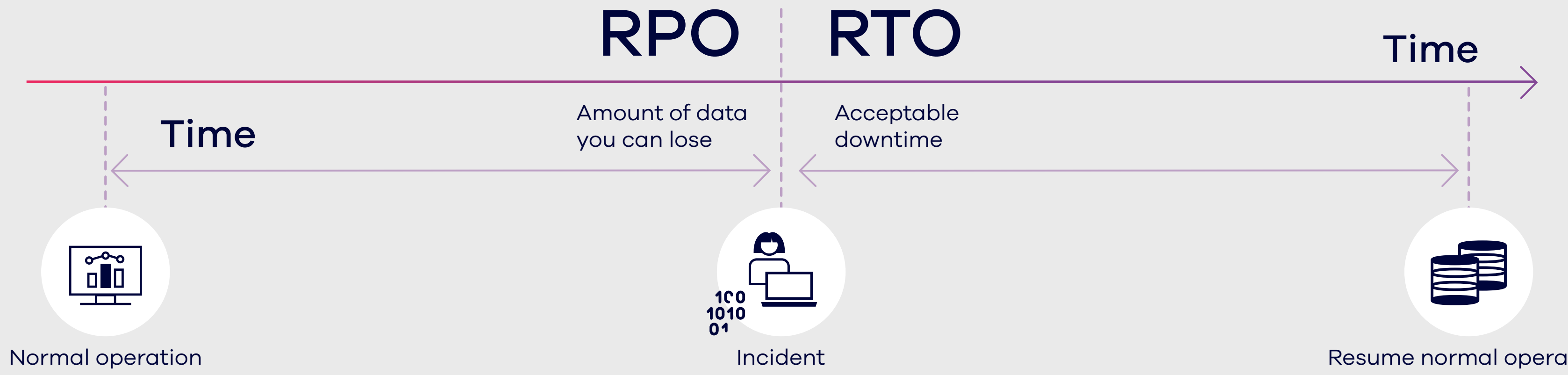
Regular backups of the transaction log enable point-in-time recovery, down to the exact second.

Recovery Point Objective and Recovery Time Objective

In the realm of data backup and recovery, perhaps the most important metrics are Recovery Point Objective (RPO) and Recovery Time Objective (RTO). They describe the tolerance of an application, SLA, or business towards data loss (RPO) and downtime (RTO).

RPO describes the maximum amount of data loss a business can tolerate in the event of a disaster. RPO is measured in time and refers to the age of the last backup that will be used to recover the data. For example, if your RPO is 15 minutes, then in the event of a disruption, you should be prepared to lose no more than the last 15 minutes' worth of data.

RTO, on the other hand, is the duration within which systems and applications must be restored after an outage to avoid unacceptable consequences associated with a break in business continuity.



In the context of SQL Server on AWS, understanding and setting the appropriate RPO and RTO is crucial.

For instance, if you're running SQL Server on Amazon RDS, AWS provides the option for automated backups that allow for point-in-time recovery to within seconds. However, the RTO can vary based on the size of the database and the nature of the incident. In contrast, for SQL Server on Amazon EC2 or VMware Cloud instances, the RTO and RPO will depend on how you've configured your backups, and the capabilities of the tools you are using.

Clumio delivers low RPOs and RTOs, in addition to capabilities like SQL-queried record retrieval. More on that later.

RPO describes the tolerance of an application, SLA, or business towards data loss.

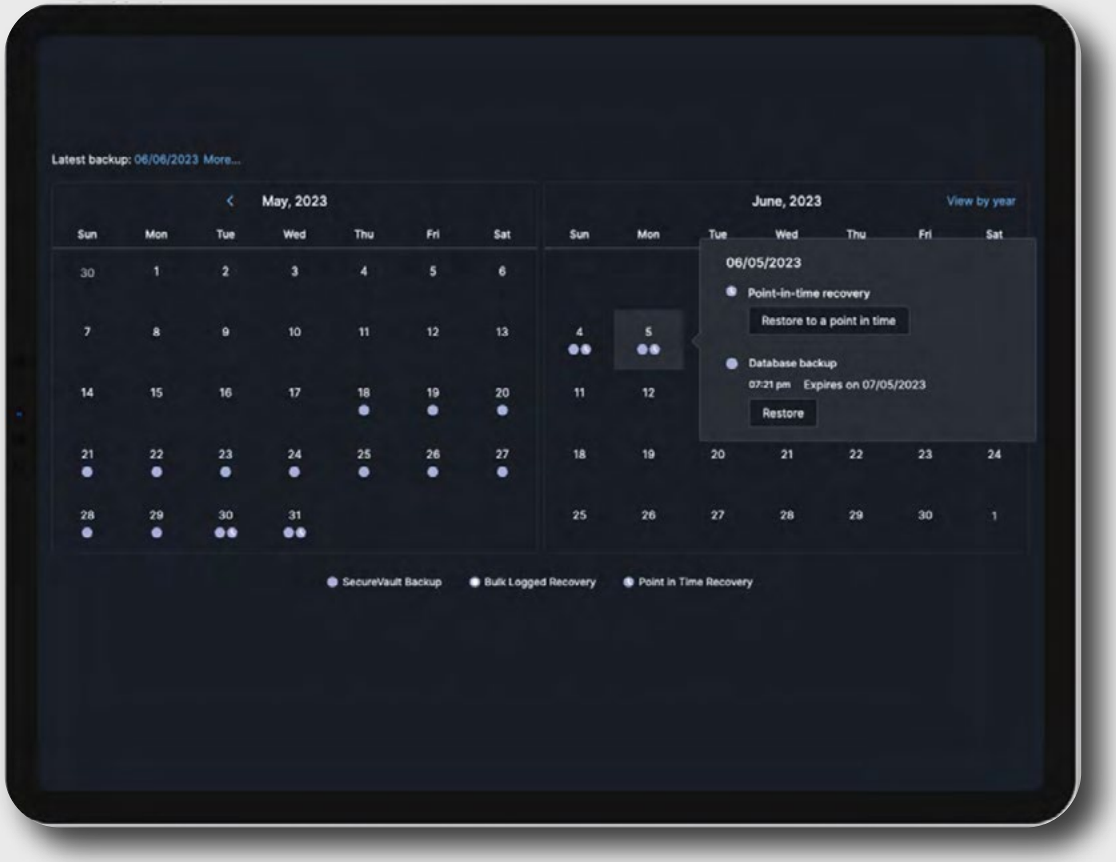
RTO describes the tolerance towards downtime.

Point-in-time recovery

As discussed in the section on transaction logs, point-in-time recovery (PITR) refers to the recovery of data to a specific point in time.

In other words, it is the process of restoring the data from a backup to a specific desired moment. PITR is essential in busy databases when data corruption or loss has occurred and there's a need to revert the data to its state to an exact point in time to debug or recover from the underlying issue.

When a database is restored using a full database backup (usually a snapshot) and subsequent transaction log backups, a SQL Server instance can be restored to a specific point in time. The combination of these two provides a much more complete strategy than using snapshots alone.



Latest backup: 06/06/2023 More...

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18 ●	19 ●	20 ●
21 ●	22 ●	23 ●	24 ●	25 ●	26 ●	27 ●
28 ●	29 ●	30 ●●	31 ●●			

Sun	Mon	Tue	Wed	Thu	Fri	Sat
4 ●●	5 ●●					
11	12					
18	19	20	21	22	23	24
25	26	27	28	29	30	1

06/05/2023

● Point-in-time recovery

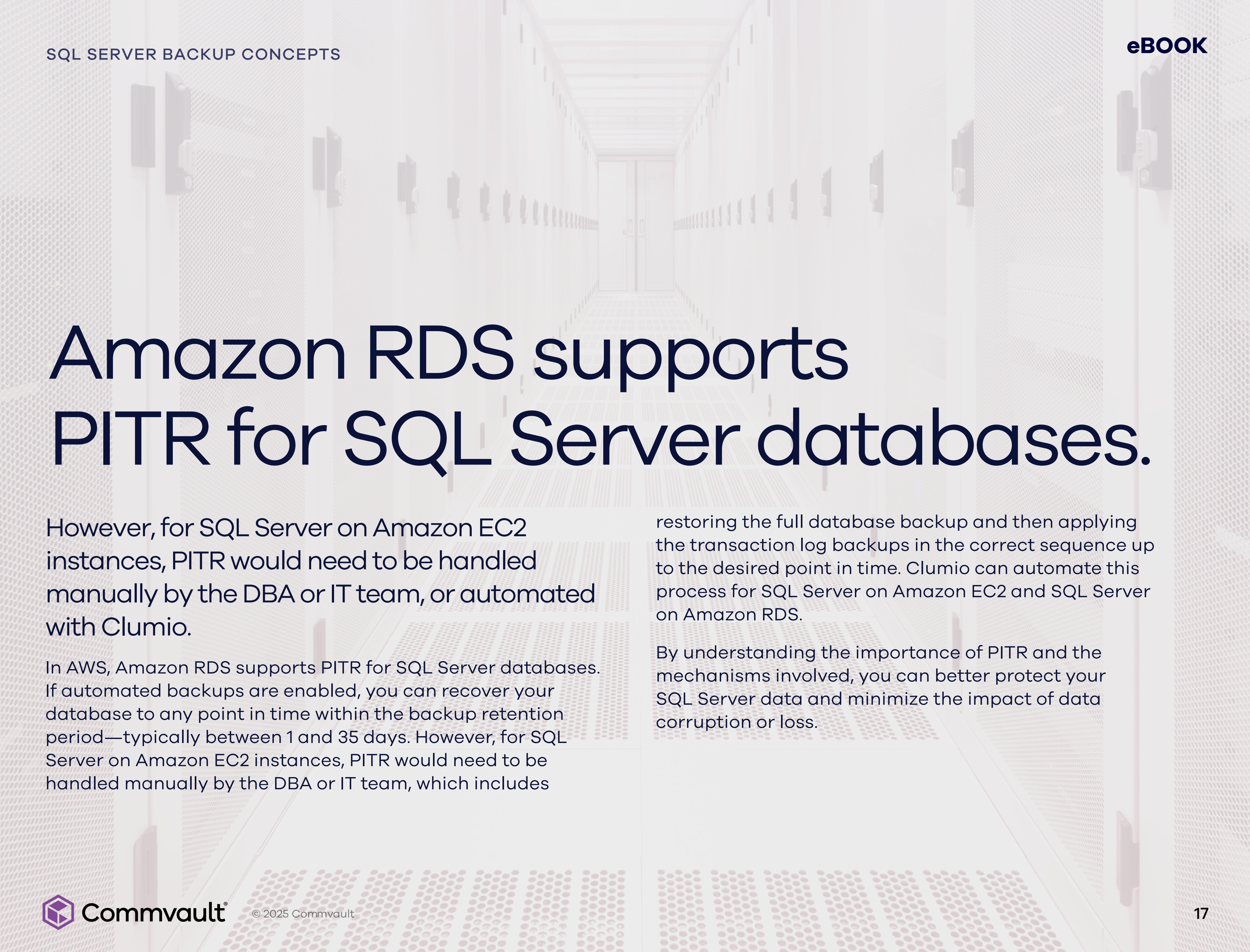
Restore to a point in time

● Database backup

07:21 pm Expires on 07/05/2023

Restore

● SecureVault Backup ● Bulk Logged Recovery ● Point in Time Recovery



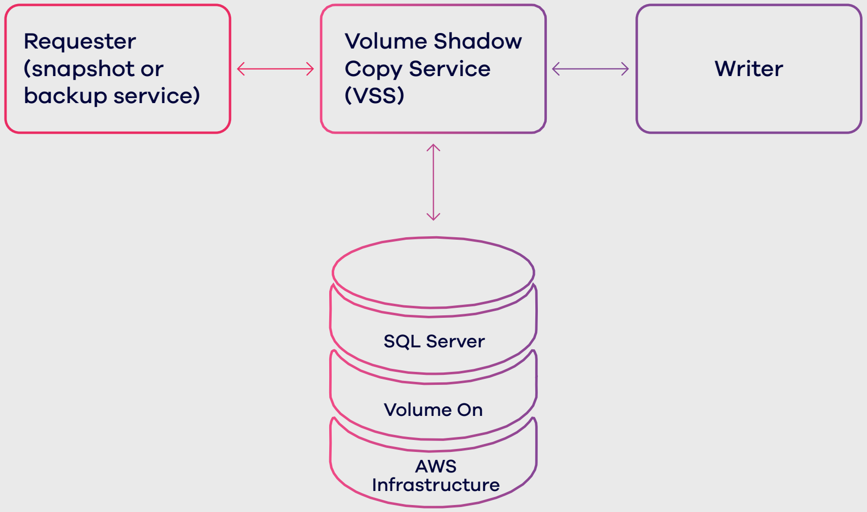
Amazon RDS supports PITR for SQL Server databases.

However, for SQL Server on Amazon EC2 instances, PITR would need to be handled manually by the DBA or IT team, or automated with Clumio.

In AWS, Amazon RDS supports PITR for SQL Server databases. If automated backups are enabled, you can recover your database to any point in time within the backup retention period—typically between 1 and 35 days. However, for SQL Server on Amazon EC2 instances, PITR would need to be handled manually by the DBA or IT team, which includes restoring the full database backup and then applying the transaction log backups in the correct sequence up to the desired point in time. Clumio can automate this process for SQL Server on Amazon EC2 and SQL Server on Amazon RDS.

By understanding the importance of PITR and the mechanisms involved, you can better protect your SQL Server data and minimize the impact of data corruption or loss.

Volume Shadow Copy Service


Requester (snapshot or backup service)
Volume Shadow Copy Service (VSS)
Writer
SQL Server
Volume On
AWS Infrastructure

Microsoft Volume Shadow Copy Service, commonly known as VSS, is a crucial component of a comprehensive backup strategy for SQL Server on AWS.

VSS is a set of Component Object Model (COM) APIs that provide a framework to enable storage volume backups to be performed while applications on a system continue to write to the volumes. VSS provides a mechanism for creating 'shadow copies'—consistent point-in-time copies of data in which all transactions are complete and no data is left in an intermediate state.

Since SQL Server is a live, transactional system, creating backups during ongoing operations can be challenging. VSS helps manage this complexity by working with SQL Server to quiesce, or temporarily pause, database operations, so that all database transactions are at a known, consistent state before the backup is taken.

VSS's role is fundamental regardless of your SQL Server deployment option. Whether you're running SQL Server on Amazon RDS, on Amazon EC2 instances, or on VMC, understanding how VSS works aids in formulating a backup strategy that helps minimize data loss in the event of an operational disruption.

In the next section, we'll delve into the backup strategies for SQL Server on AWS.

03

Your Choices for SQL Server Backup In AWS

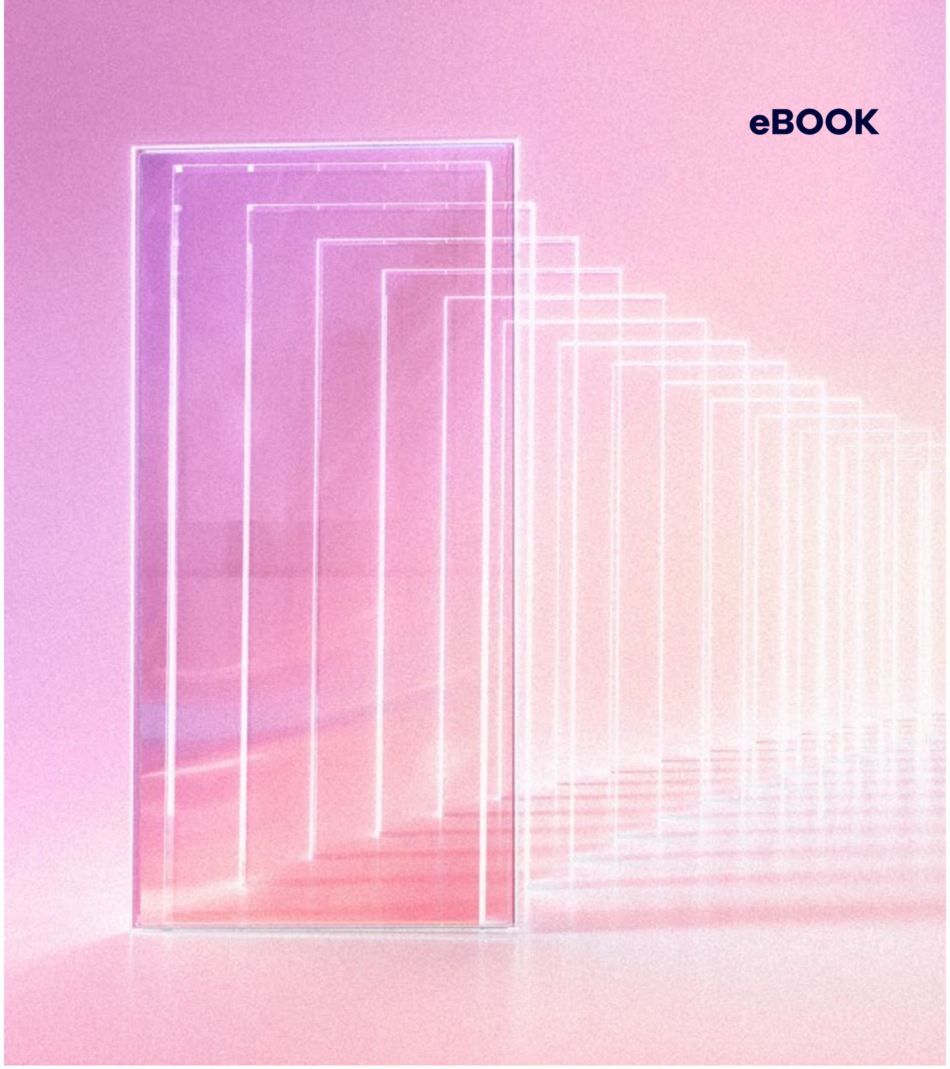
There are several backup strategies for SQL Server on AWS, and trade-offs among the different deployment options—SQL Server on Amazon RDS, on Amazon EC2 instances, and on VMC on AWS.

Whether you are a DBA, IT professional, or a decision-maker responsible for data protection in your organization, this chapter aims to provide you with a comprehensive understanding of your choices for SQL Server backups in AWS. We'll also discuss how third-party backup solutions like Clumio can help streamline this process and provide additional layers of data protection.

Native SQL Server backups

Whether you're running your SQL Server on Amazon EC2, Amazon RDS, or VMC on AWS, you can back up SQL Server databases using native differential or full backups. This method provides portability of SQL Server databases across Amazon RDS, Amazon EC2, and VMware instances, and backups can be stored in Amazon EBS, Amazon S3, or Amazon FSx. However, it's important to understand the nuances and considerations.

Since SQL Server backups support only full or differential backups (and not incremental backups), the size of these backups can get pretty large, especially for fast-changing databases. Moreover, storing native SQL backups in Amazon EBS or Amazon FSx can get expensive and cause storage and throughput bottlenecks. To get around this, customers sometimes manually export their SQL Server backups to Amazon S3. However, since Amazon S3 has a file size limit of 5TB, this method is not recommended for larger databases. In addition, customers cannot back up to or restore from more than 10 backup files at the same time, and cannot run more than two concurrent backup / restore tasks.



From a usability standpoint, native SQL Server backups come with some limitations. You cannot take a native backup during a snapshot or automated backup. And for differential backups to work, you cannot take a snapshot between the last full backup and your differential backup task. To simplify backups for SQL Server, customers therefore choose either built-in backups in AWS, AWS Backup, or Clumio.

Built-in backups in AWS

AWS provides some built-in options for backing up SQL Server depending on the deployment model. These methods provide foundational data protection capabilities and are a good starting point for your backup strategy.

FOR SQL SERVER ON AMAZON RDS: AWS offers automated backups up to the size of the provisioned disk capacity, which should satisfy most of your operational backup requirements. Amazon RDS does this by taking a full daily snapshot in the user-specified time window, along with transaction logs that enable point-in-time recovery. You can specify a backup window, and AWS will take care of the rest. It provides a backup retention period of up to 35 days, enabling point-in-time recovery.

However, these built-in backups do have certain limitations. For example, they are tied to the life of the instance. If the Amazon RDS instance is deleted, all associated automated backups are also deleted. Therefore, they may not serve as a comprehensive ransomware or accidental deletion recovery solution. Additionally, the retention period of 35 days may not be enough for businesses with longer data retention requirements.

FOR SQL SERVER ON AMAZON EC2 INSTANCES: AWS provides the capability to create Amazon Machine Images (AMIs) and snapshots of Amazon EBS volumes. The combination of these is colloquially known as 'Amazon EC2 snapshots', and can be operated manually with AWS Backup. They are volume level copies and do not allow database-level backup and restore, and typically do not provide point-in-time recovery.

FOR SQL SERVER ON VMC ON AWS: Data protection and backup options are typically as native VMware snapshots, or by third-party solutions that can be found on the VMware Marketplace. VMware snapshots can impact VM performance, and third-party applications can be costly and require additional storage.

Regardless of your SQL Server deployment model in AWS, it's important to be aware of these built-in backup options, their benefits, and limitations. Fortunately, you can augment these with other backup methods to create a comprehensive backup strategy.

Snapshots

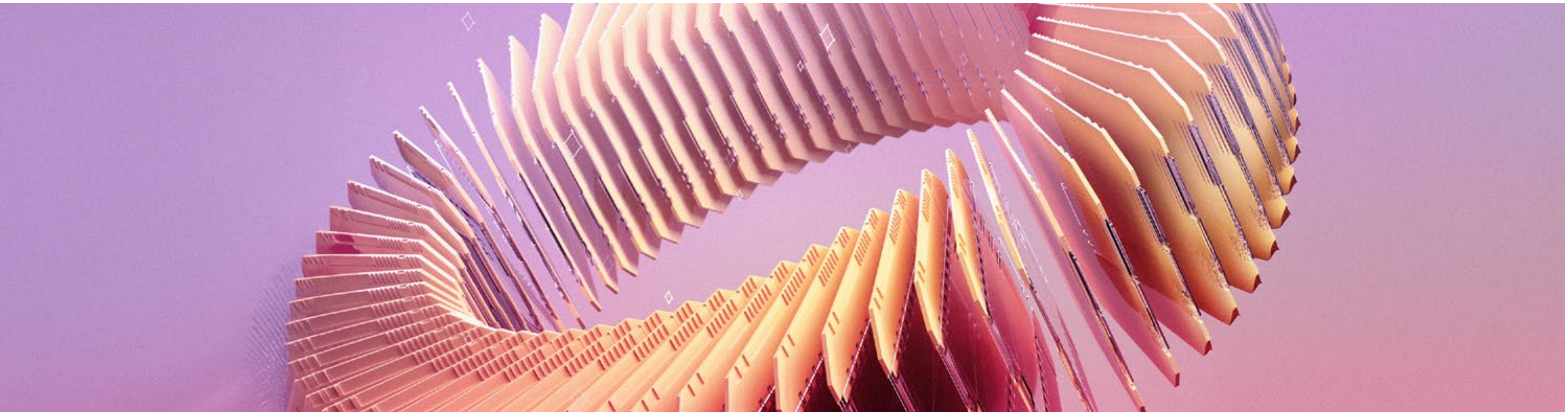
Manual snapshots are often used in combination with automated backups to provide additional coverage for specific business needs, such as backup before a significant change, for archival purposes, or to meet specific compliance requirements.

FOR SQL SERVER ON AMAZON RDS: AWS offers the capability to manually create database snapshots at any time. These user-initiated full snapshots enable you to periodically back up your SQL Server instance in a known state, and then restore to that state at any time. Unlike automated backups, manual DB snapshots are kept until explicitly deleted, providing a longer-term, durable backup solution beyond the maximum 35-day retention period of automated backups. However, these snapshots are not point-in-time recoverable, and incur additional storage fees.

FOR SQL SERVER ON AMAZON EC2: You can manually create Amazon EBS snapshots, as stated in the previous section. These snapshots are incremental, which means only the changed blocks since your last snapshot are stored. While these can be space-efficient for static or slow-changing databases, it is not recommended to use incremental snapshots for databases with high change rates, since it can raise costs unexpectedly.

FOR VMC ON AWS: You can leverage the features of your VMware environment to perform manual backups. This could involve creating snapshots of your VMs, or manually exporting your databases.

Keep in mind that manual backup methods such as those laid out above require diligent monitoring and management. Also, in the event of a disaster, piecing together data from different manual backups can be time-consuming and complex.



AWS Backup

Some AWS customers use AWS Backup to manage backups and snapshots of their SQL Server databases.

FOR SQL SERVER ON AMAZON RDS: AWS Backup can take full snapshots of Amazon RDS SQL Server DB instances, and keep them for a user-specified retention period. AWS Backup allows you to define policies, including when to take backups, how long to retain them, and when to transition the backups to a cheaper storage class or delete them. The key limitation is that AWS Backup does not offer point-in-time recovery for RDS.

FOR SQL SERVER ON AMAZON EC2: AWS Backup creates a VSS-enabled snapshot of the Amazon EBS volume attached to your Amazon EC2 instance running the database, resulting in an application-consistent backup. AWS Backup also allows you to manage backups across multiple accounts and regions. However, as in the case with Amazon RDS, AWS Backup does not offer point-in-time recoveries for SQL Server on Amazon EC2.

FOR VMC ON AWS: AWS Backup also supports backing up virtual machines in VMC on AWS that may be running SQL Server by creating an AWS gateway to connect to vSphere. There is no point-in-time recovery capability out of the box.

Clumio

Clumio is a serverless backup solution for data in AWS, including SQL Server. Clumio is architected with features that are uniquely advantageous for backing up SQL Server databases running on AWS, regardless of deployment model.

AIR GAP: Air gapping is a means of securing critical backed up data outside of the primary access control domain, so that data is unreachable by malicious actors even if the enterprise security domain is breached. Clumio is air gapped so that your SQL Server backups remain secure and recoverable even if your primary environment is compromised.

CONTINUOUS BACKUPS AND POINT IN TIME RECOVERY: By continuously backing up transaction logs, along with a combination of full and incremental VSS-enabled snapshots, Clumio can enable granular point-in-time recovery for both Amazon RDS and self-managed distributions of SQL Server.

COMPLIANCE EFFICIENCY: In addition to operational backups, Clumio also offers an ultra low-cost long-term retention mode for SQL Server backups by automatically storing this data in Amazon S3. This is particularly useful for customers that need to adhere to compliance requirements in regulated industries.

AVAILABILITY GROUP AWARENESS: Clumio backups are Availability Group aware, so that whenever specified by the user, backups are taken from a secondary replica, rather than the primary database serving the application, to minimize performance impact on the production environment. The user may specify the primary replica as their preferred choice for taking backups.

NO MAINTENANCE: Clumio is a simple SaaS solution with no custom engineering requirements, and 24X7 human support built in. This helps customers free up their engineering staff to focus on revenue drivers.

NO PERFORMANCE IMPACT: With no proprietary agents to install and the ability to back up from a secondary replica, Clumio allows backups to be performed without impacting your database's performance.

GRANULAR RECORD RETRIEVAL: Uniquely, Clumio offers Amazon RDS users the ability to simply query backups to retrieve granular records or datasets using SQL queries, rather than having to restore full databases and then look for the right data. This helps simplify data retrieval, minimize downtime, and help ease compliance audits.

INDUSTRY LEADING RPO / RTO: Clumio's serverless architecture enables low RPO and RTO by efficiently scaling resources and parallelizing operations. The platform uses Lambdas orchestrated into large workflows, enabling predictable performance even at scale.

04

Exploring Clumio for SQL Server Backup In AWS

We've covered several methods for backing up SQL Servers running on AWS, the inherent limitations with each of these methods, and the resources required to manage backups across different SQL Server deployments.

Now let's further explore how Clumio can be used by DBAs / DBEs as well as cloud infra / IT teams to gain robust, reliable, and hassle-free backups for SQL Server on AWS. Whether you are running SQL Server on Amazon RDS or Amazon EC2, Clumio can provide a secure, unified, and simple solution that overcomes many of the limitations of the other backup methods.

Introduction to Clumio

Clumio is a backup and recovery solution that helps AWS customers build resilience into their applications, databases, data lakes, and sensitive information.

Air-gapped and immutable by design, Clumio helps organizations recover quickly from ransomware and operational disruptions, automate data compliance, and optimize storage costs. Customers choose Clumio for 4 key reasons:

- ✓ **SIMPLICITY:** With no hardware or software to deploy, along with an intuitive UI and a rich API, Clumio makes backing up even highly complex and interwoven data services simple.
- ✓ **SECURITY:** Clumio securely air gaps customer data in an immutable vault that is encrypted and MFA'd. Clumio's data storage helps meet ISO, medical, and financial compliance standards.
- ✓ **ARCHITECTURAL EFFICIENCIES:** Clumio is architected as a stateless data processing pipeline, leveraging a proprietary serverless orchestration engine that gives it a scalability of exabytes per customer, change tracking of thousands of API operations per second, and incredibly fast recovery.
- ✓ **COST PERFORMANCE:** Clumio is not only optimized to reduce backup footprint, it also includes 24/7 engineering support in its consumption based pricing.

Let's explore the architecture that makes all this possible.

Architecture

As shown in the figure, all you need to start using Clumio is an AWS account with SQL Server instances.

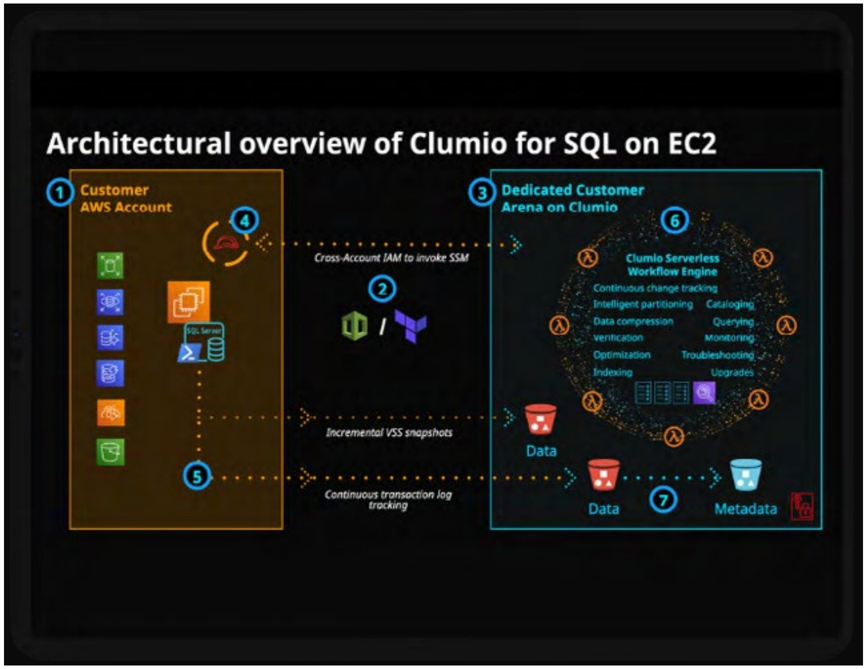
Simply login to your AWS account and invoke Clumio through a CloudFormation Template or Terraform.

Clumio sets up a dedicated, immutable, and air gapped environment on its portal to store your backups.

Clumio does not deploy any external agents, and once granted IAM permissions, works with the Amazon EC2 Systems Manager (SSM). For Amazon RDS deployments, Clumio uses direct APIs of the respective services.

Clumio then takes incremental, application-consistent VSS snapshots from the SSM agent periodically and combines it with continuous transaction log tracking to capture each change.

Clumio's serverless workflow engine provides consistent, queryable, and efficient storage of both data and metadata from backed-up databases on scale-out object storage.



Architectural overview of Clumio for SQL on EC2

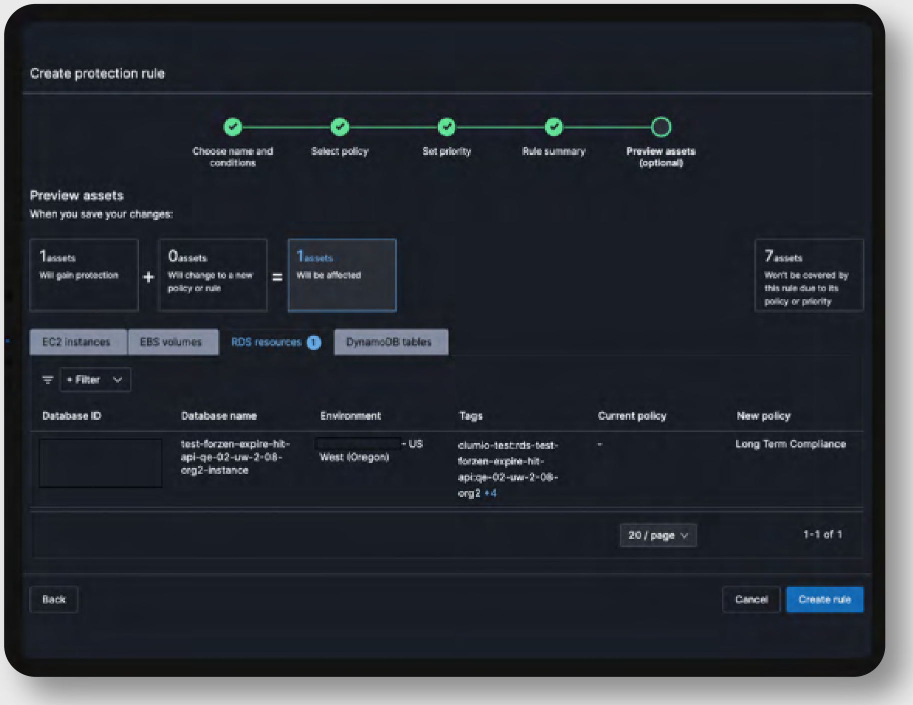
1 Customer AWS Account
4
Cross-Account IAM to invoke SSM
2
/
SQL Server
3 Dedicated Customer Arena on Clumio
6
Clumio Serverless Workflow Engine
Continuous change tracking
Intelligent partitioning
Cataloging
Data compression
Querying
Verification
Monitoring
Optimization
Troubleshooting
Indexing
Upgrades
Incremental VSS snapshots
Data
5
Continuous transaction log tracking
Data
7
Metadata

Clumio secures the backup vault with its own rotating keys, and optionally, customer-provided keys as well.

Clumio backups are incremental forever, supporting fast backup and restores while helping reduce overall cost.

Let's explore how to implement Clumio backups of SQL Servers in AWS.

How to backup Amazon RDS SQL Server



Create protection rule

Preview assets

When you save your changes:

1 assets Will gain protection + 0 assets Will change to a new policy or rule = 1 assets Will be affected

7 assets Won't be covered by this rule due to its policy or priority

Database ID	Database name	Environment	Tags	Current policy	New policy
[redacted]	test-forzen-expire-hit-api-qe-02-uw-2-08-org2-instance	[redacted] - US West (Oregon)	clumio-test:rds-test-forzen-expire-hit-api:qe-02-uw-2-08-org2 +4	-	Long Term Compliance

The first few steps are common across the three deployment models of SQL Server.

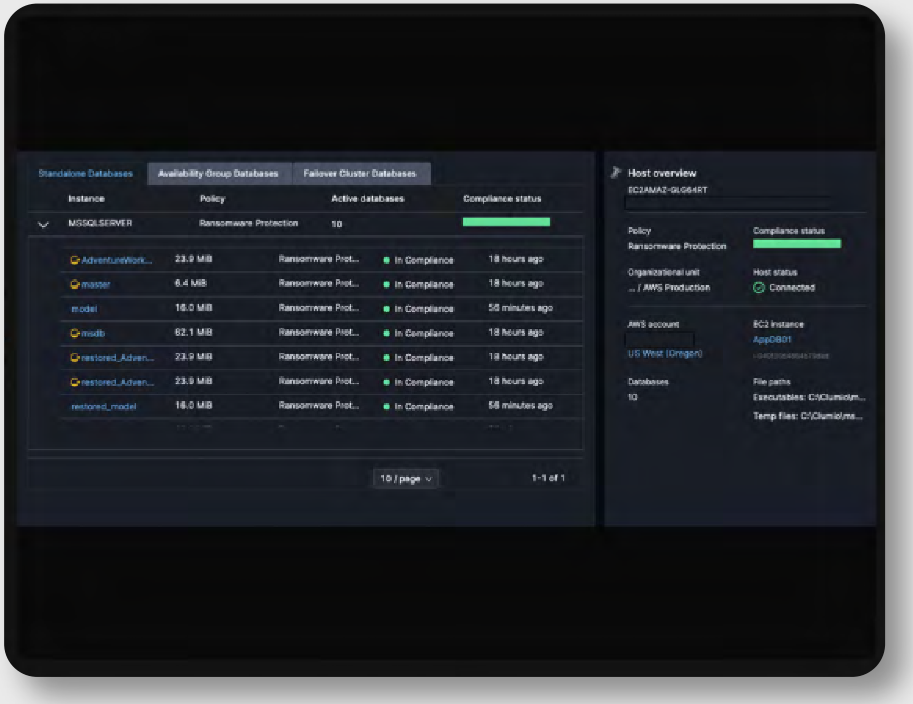
- Create a Clumio account and login.
- Verify IAM permissions required to deploy Clumio on support.clumio.com.
- Login to your AWS account.
- Within Clumio, add your AWS account details.
- Kickoff creation of the CloudFormation stack or initialize the Clumio Terraform provider in the Terraform registry.
- Confirm connection status on Clumio shows 'Connected'.

For detailed instructions or troubleshooting, hop over to support.clumio.com. Next, back up your Amazon RDS environment to Clumio using these simple steps.

- Once your account containing Amazon RDS instances is connected, define a Clumio RDS policy indicating when snapshots or backups of your Amazon RDS instances should be generated and the retention period for this data.
- Create a Protection Rule to apply this policy to your Amazon RDS resources based on tags, and across accounts and regions. This automates your backups. Clumio will retain the latest snapshot in your environment for incremental backups. The snapshot will then be updated according to your policy definition.

As mentioned earlier, Clumio offers snapshot recovery, point-in-time recovery, as well as SQL query-based granular retrieval for Amazon RDS.

How to backup SQL Server on EC2



Instance	Policy	Active databases	Compliance status
MSSQLSERVER	Ransomware Protection	10	🟩

AdventureWork...	23.9 MiB	Ransomware Prot...	● In Compliance	18 hours ago
master	6.4 MiB	Ransomware Prot...	● In Compliance	18 hours ago
model	16.0 MiB	Ransomware Prot...	● In Compliance	56 minutes ago
msdb	62.1 MiB	Ransomware Prot...	● In Compliance	18 hours ago
restored_Adven...	23.9 MiB	Ransomware Prot...	● In Compliance	18 hours ago
restored_Adven...	23.9 MiB	Ransomware Prot...	● In Compliance	18 hours ago
restored_model	16.0 MiB	Ransomware Prot...	● In Compliance	56 minutes ago

Host overview
EC2AMAZ-GLG64RT
[redacted]

Policy: Ransomware Protection
Compliance status: 🟩
Organizational unit: ... / AWS Production
Host status: ✓ Connected
AWS account: [redacted] US West (Oregon)
EC2 instance: AppDB01 [illegible]
Databases: 10
File paths: Executables: C:\Clumio\m... Temp files: C:\Clumio\ms...

From the Clumio AWS Environment page, add SQL server hosts and assign IAM permissions using the CloudShell command.

Install Clumio executables which will connect the SQL server host to Clumio.

Add all nodes in your Always On Availability Groups to Clumio so backups can be performed from a secondary replica, which avoids impacting the performance of the primary database.

Apply backup policies at the Availability Group level for Always On Availability Groups, selecting the preferred replica for backups. Policies can also be set at the host or individual database level, allowing flexibility in how and where backups are run.

Clumio backs up SQL Server on EC2 databases whether they are in full or simple recovery mode. If the database is in full recovery mode, there are two ways to recover: Point-in-time recovery or full database recovery. Simple recovery mode allows for full database recovery only. Clumio also supports SQL Server Failover Cluster Instance (FCI) backups.

05

Customer Success Stories

BIOPLUS

BioPlus Specialty Pharmacy, a subsidiary of healthcare giant Elevance (previously Anthem) is the United States' largest non-payer specialty pharmacy. BioPlus innovated a 2-hour acceptance guarantee, 2 day ready to ship guarantee, and 2 click online refills, which means a lot of transactions hit up their SQL Server databases.

Moreover, BioPlus's IT and DBA teams manage data in a highly regulated environment, subject to HIPAA, Medicare, and PCI compliance standards. In addition, given the frequency of ransomware attacks in the healthcare and medical space, they had to prioritize backing up their AWS environment, with particular focus on SQL Server.

The team at BioPlus evaluated multiple vendors but settled on Clumio for its scalability and air gapped security. Additionally, Clumio's cost effective long-term retention requirements for patient records helped them simplify HIPAA and CMS compliance (with policies to store data for up to 10 years).

BioPlus was able to have all their backups up and running in less than a week from sign up, and have since tested several modalities of restores, including granular record-level recoveries. Overall, Clumio saved the team ~20% vs. other solutions considered.

A PUBLICLY LISTED LUXURY TIMEPIECE MANUFACTURER

A US-based luxury watch manufacturer founded in the 1800s, with over $750M in revenue today, chose Clumio to protect its SQL Server databases running in AWS.

While migrating their infrastructure from on-premises to AWS, they were considering using Rubrik to protect their SQL Server databases in the cloud, but eventually chose Clumio for its cloud-native, SaaS approach to backups. Today, they are leveraging Clumio to back up their SQL Server environment in AWS, including failover cluster instances.

06

The Future of SQL Server Backup In AWS

Today's growing need for increased resilience, faster recovery times, and compliance with strict data regulation is driving rapid evolution in backup and recovery strategies. From automatic and manual backups to the centralization offered by AWS Backup, AWS offers a range of options to meet diverse business needs. Each method comes with its unique advantages and potential drawbacks. However, the nuances and complexities around managing backups across different SQL Server deployments, limitations on point-in-time recovery, and other constraints, call for a more comprehensive solution.

Clumio addresses these challenges with an air-gapped and immutable architecture, keeping your backups protected even in worst-case scenarios. Clumio's continuous change tracking, point-in-time recovery, help with compliance efficiency, granular record retrieval, availability group awareness, and low maintenance approach make it a very popular solution for many SQL Server customers. Clumio showcases the power of a cloud-native, SaaS approach to backups, focusing on simplicity, security, architectural efficiencies, and cost performance. As seen in the customer stories, Clumio's solutions have enabled businesses to achieve significant savings, improve efficiency, and bolster their data protection strategy.

Looking ahead, as SQL Server continues to migrate to the cloud, the need for robust, reliable, and scalable backup solutions will only increase. Services like Clumio will help you stay at the forefront of this shift, providing a secure and unified solution for SQL Server backups in AWS. As you evolve your business continuity and disaster recovery strategies to protect your SQL Server databases in AWS, we hope that this book has provided you with a comprehensive understanding of your options and a clear direction for the future. While the landscape of backup and recovery is complex and constantly evolving, the principles remain the same: secure your data, maintain its recoverability, and do so in the most efficient and cost-effective manner possible. The future of SQL Server backups in AWS is bright, and we look forward to seeing what innovations will come next. Equipped with the knowledge and insights from this book, you are well-prepared to navigate this future successfully.

Commvault®

REQUEST A 1:1 DEMO

commvault.com/clumio/request-demo

Clumio